1997 IEEE Asian Test Symposium, pp. 62-67, November 1997.

Sequential Test Generation Based on Circuit Pseudo-Transformation

Satoshi Ohtake Tomoo Inoue Hideo Fujiwara
Graduate School of Information Science, Nara Institute of Science and Technology
Ikoma, Nara 630-01, JAPAN

Abstract

The test generation problem for a sequential circuit capable of generating tests with combinational test generation complexity *can be reduced to that for the combinational circuit formed by replacing each FF in the sequential circuit by a wire. In this paper, we consider an application of this approach to general sequential circuits. We propose a test generation method using* circuit pseudo-transformation *technique: given a sequential circuit, we extract a subcircuit with* balanced structure *which is capable of generating tests with combinational test generation complexity, replace each FF in the subcircuit by wire, generate test sequences for the transformed sequential circuit, and finally obtain test sequences for the original sequential circuit. We also estimate the effectiveness of the proposed method by experiment with ISCAS'89 benchmark circuits.*

1 Introduction

Test generation for sequential circuits is a well-known hard problem, and is considerably difficult compared to that for combinational circuits[1]. The search space of test generation for a sequential circuit depends on the state space of the sequential circuit and hence on the number of FFs of the circuit. If the number of FFs can be reduced, the sequential test generation time will be reduced. In order to reduce the number of FFs in a sequential circuit tentatively during test generation for the sequential circuit, we shall propose *circuit pseudo-transformation*(CPT for short). CPT changes a circuit under consideration into a different circuit whose test generation is easier than the original one, where the circuit is not changed physically but is just transformed tentatively only during test generation. A test generation method for sequential circuits based on CPT consists of the following three steps: given a circuit, transform the circuit by CPT, generate a test sequence for the transformed circuit, and make a test sequence for the original circuit from the test sequence obtained for the transformed circuit. The *software transformation*, presented by Balakrishnan and Chakradhar[2], is a CPT based on retiming technique[3]. Though the software transformation can reduce the number of FFs, transformed circuits are not sufficiently easy to generate test sequences.

The test generation problem for a sequential circuit *capable of generating tests with combinational test generation complexity* can be reduced to that for the combinational circuit formed by replacing each FF in the sequential circuit by a wire. *Balanced structures*[4] and *internally balanced structures*[5] are known as sequential circuits capable of generating tests with combinational test generation complexity. In this paper, we shall present a test generation method using a CPT called *combinational circuit pseudo-transformation*(CCPT for short). CCPT consists of two steps: find a balanced subcircuit in a given circuit, and replace each FF in the subcircuit by a wire. The test generation method using this CCPT has the following three steps: given a sequential circuit, transform the circuit by CCPT, generate a test sequence for the transformed circuit, and make the sequence for the original circuit from the generated sequence. Since the number of FFs in the transformed circuit is smaller than that in the original circuit, it will be expected to reduce the test generation time and to increase the fault coverage by this transformation. Although the test generation method requires a circuit (physical) modification, which adds a hold mode to some FFs, the hardware overhead is negligible and the performance degradation dose not occur.

This paper is organized as follows: Section 2.1 proposes the definition of CCPT. *K-clock hold transformation*, *k-clock hold sequence transformation* and *k-clock hold testing* are proposed in Section 2.2, 2.3 and 2.4, respectively, to use the test sequence of the transformed circuit by CCPT as a test sequence of the original circuit. Section 3 presents testability preservation of both CCPT and that of k-clock hold transformation, and describes the reducibilities of test generation problems. Section 4 presents a test generation method based on CCPT and estimates the effectiveness of the method by experiment with ISCAS'89 benchmark circuits. The experimental results show that the proposed method can reduce test generation time for most circuits and can increase the fault coverage for several circuits.

2 Circuit Pseudo-Transformations

If a transformation modifies both the hardware design of a circuit and the circuit model for test generation, the transformation is said to be *circuit transformation*(CT). On the other hand, a transformation which only modifies the circuit model for test generation is called *circuit pseudo-transformation* and defined as follows.

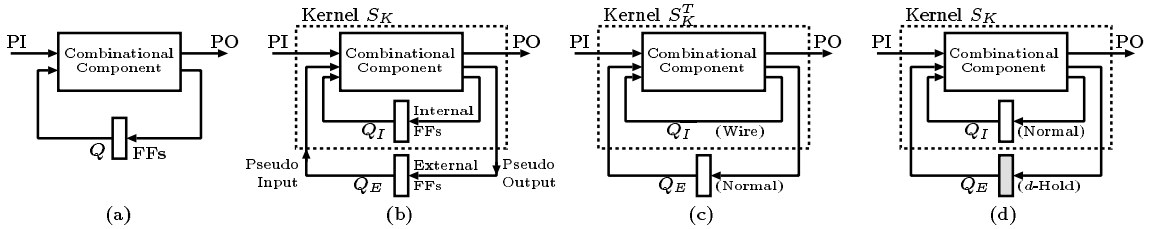

Figure 1: (a) Sequential circuit S, (b) kernel circuit S^K and set of external FFs Q_E, (c) sequential circuit S^T, and (d) sequential circuit S^H.

Definition 1 A transformation which transforms "tentatively" a circuit into another circuit is said to be *circuit pseudo-transformation*(CPT). Here, "tentatively" means that the circuit model for test generation is modified tentatively during test generation but the hardware design of the circuit is not changed. □

As one of the CPTs, we shall propose *combinational circuit pseudo-transformation*(CCPT) in Section 2.1. The CCPT can reduce the number of FFs in a sequential circuit. We shall consider a test generation method based on CCPT. We expect that the test generation time for a circuit which is obtained by CCPT is less than the original circuit. We shall present such a test generation method using CCPT in Section 4. A test sequence which is obtained by the test generation method using CCPT can not be used as a test sequence for the original (before CCPT) circuit. In order to use the generated sequence as a test sequence for the original circuit, we shall propose a CT (called *d-clock hold transformation*) in Section 2.2 and a sequence transformation for the obtained test sequence (called *k-clock hold sequence transformation*) in Section 2.3.

2.1 Combinational Circuit Pseudo-Transformation

Let S be a sequential circuit(see Figure 1(a)). Let Q be the set of all FFs in S and let Q_E be a subset of Q. An FF in Q_E is said to be an *external FF* and an FF in $Q_I = Q - Q_E$ is said to be an *internal FF*. Then, we can define a *kernel circuit* as follows.

Definition 2 A subcircuit S_K, which is formed by replacing each external FFs in S by primary input/output, in S is said to be a *kernel circuit* of S (see Figure 1(b)). The inputs to the kernel circuit from external FFs is said to be *pseudo-inputs*. Similarly, the outputs from the kernel circuit to external FFs is said to be *pseudo-outputs*. The inputs of the kernel circuit are primary inputs of S and pseudo-inputs from external FFs. Similarly, the outputs of the kernel circuit are primary outputs of S and pseudo-outputs to external FFs. □

The process to determine external FFs and a kernel circuit for a circuit is called *partitioning*. If a sequential circuit has no feedback loops, the circuit is said to be an *acyclic structure*. In the rest of this paper, we assume that the kernel circuit is an acyclic structure. Suppose a path P from an input to an output of a circuit. The number of FFs in P is said to be the *sequential depth* of P. The largest sequential depth in the kernel circuit is said to be the *sequential depth of the kernel circuit*.

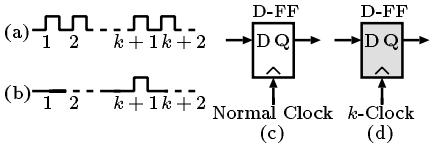

Figure 2: (a) Normal clock, (b) k-clock, (c) normal FF, and (d) k-clock hold FF.

Suppose that a sequential circuit S is partitioned into a kernel circuit and several external FFs (see Figure 1(b)). Then, we can define *combinational circuit pseudo-transformation* as follows.

Definition 3 A circuit pseudo-transformation T that transforms S into a circuit S^T by replacing each internal FF by a wire is said to be *combinational circuit pseudo-transformation*(CCPT) (see Figure 1(c)). □

2.2 *d*-Clock Hold Transformation

We suppose that a sequential circuit is synchronized by a single system clock, called a *normal clock* (see Figure 2(a)). An FF to which the normal clock is supplied is said to be a *normal FF* (see Figure 2(c)). We define *k-clock* and *k-clock hold FFs* as follows.

Definition 4 Let k be a positive integer. A clock that generates pulses at every $(k+1)$-th cycle of the normal clock is said to be a *k-clock* (see Figure 2(b)). □

Definition 5 Let k be a positive integer. An FF to which the k-clock is supplied is said to be a *k-clock hold FF* (see Figure 2(d)). □

The k-clock hold FF holds a current data during k cycles and loads a new data at the next cycle.

Let S be a sequential circuit (see Figure 1(a)). Suppose that circuit S is partitioned into a kernel circuit and external FFs (see Figure 1(b)). Let d be the sequential depth of the kernel circuit. Then, we can define *d-clock hold transformation* as follows.

Definition 6 A circuit transformation H that transforms S into a sequential circuit S^H by replacing each external FF by the d-clock hold FF is said to be *d-clock hold transformation* (see Figure 1(d)). □

Note that the d-clock hold transformation requires a hardware design modification. The hardware design modification will be described in Section 2.4.

2.3 *k*-Clock Hold Sequence Transformation

Suppose a sequence t whose length is a multiple of $k+1$, where k is a positive integer. Let $t_i(i =$

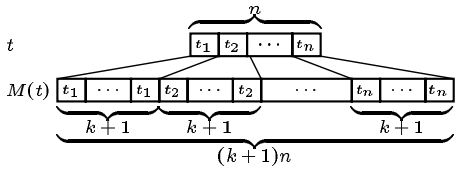

Figure 3: k-clock hold sequence transformation.

$1, 2, \ldots, n$) be the ith vector in t, where n is the length of t. Then, we can define a *k-clock hold sequence* as follows.

Definition 7 For any integer i, j and m such that $1 \leq m \leq n/(k+1)$ and $(m-1)(k+1) < i \leq m(k+1) \wedge (m-1)(k+1) < j \leq m(k+1) \wedge i \neq j$, if $t_i = t_j$, sequence t is said to be a *k-clock hold sequence*. □

Let t be a sequence of vectors. Let $t_i (i = 1, 2, \ldots, n)$ be the ith vector of t, where n is the number of vectors in t. Let k be a positive integer. Let t^k be a k-clock hold sequence and let $t^k_l (l = 1, 2, \ldots, n(k+1))$ be the lth vector in t^k, where $n(k+1)$ is the number of vector in t^k. Then, we can define *k-clock hold sequence transformation* as follows.

Definition 8 Let M be a sequence transformation. For any integer i, j such that $(i-1)(k+1) < j \leq i(k+1)$, if $t^k_j = t_i$, M is said to be *k-clock hold sequence transformation* (see Figure 3). □

2.4 d-Clock Hold Testing

Let S be a sequential circuit (see Figure 1(a)). Suppose that a sequential circuit S is partitioned into a kernel circuit and external FFs (see Figure 1(b)). Let d be the sequential depth of the kernel circuit. Let S^H be a sequential circuit which is obtained from S by the d-clock hold transformation (see Figure 1(d)). In order to realize S^H, the original circuit S is augmented to S' so that S' has two configurations of a normal mode and a test mode. That is, the normal mode of S' is equivalent to S and the test mode of S' is equivalent to S^H. To implement such S', we need to add an extra clock, d-hold clock, which is applied to external FFs in test mode. Although the hardware modification is required for the test generation method based on CCPT, the hardware overhead is negligible and the performance degradation does not occur. When S' is configured to the normal mode S, the normal clock is supplied to external FFs. A test application method for the normal mode S is said to be *normal testing*. On the other hand, when S' is configured to the test mode S^H, the d-clock is supplied to external FFs. A test application method for the test mode S^H, called *d-clock hold testing*, is defined as follows.

Definition 9 Let t be a d-clock hold sequence. A test application method for S^H which applies t as an input sequence is said to be *d-clock hold testing*. □

3 Testability Preservation

Here, we assume that when a circuit is given the test application method for the circuit is also given. Let C be a circuit and let τ be a CT either or a CPT. Let C^τ be a circuit obtained from S by τ. Let F and F^τ be the sets of all faults in C and C^τ, respectively. Let f be a fault in C and let C_f be the faulty circuit of C caused by f. When an input sequence t of C is applied to C and C_f, if the response of C is different from that of C_f, t is said to be a *test sequence* for f and f is said to be *testable* by t.

Definition 10 Transformation τ is said to be a *testability preserving transformation* if the following three conditions are satisfied.

(i) There exists a mapping $\varphi_\tau : F \mapsto F^\tau$.

(ii) For any f in F, if f is testable by the test application method of a circuit C, $\varphi_\tau(f)$ is also testable by the test application method of C^τ.

(iii) For any f in F, if f is not testable by the test application method of C, $\varphi_\tau(f)$ is also not testable by the test application method of C^τ. □

Let $\mathcal{T}(f)$ denote the set of all test sequences for f. Then, we define the property of the test generation problem reduction as follows.

Definition 11 For C and C^τ, *the test generation problem of C can be reduced to the test generation problem of C^τ*, if the following two conditions are satisfied.

(i) Transformation τ is a testability preserving transformation.

(ii) Let φ_τ be a mapping from F to F^τ and let F' ($\subseteq F^\tau$) be the set of testable faults of C^τ by the test application method of C^τ. For all f in F', there exists transformation σ such that $\bigcup_{t \in \mathcal{T}(f)} \{\sigma(t)\} \subseteq \bigcap_{g \in \varphi_\tau^{-1}(f)} \mathcal{T}(g)$. □

Note that the condition (ii) means that any test sequence for a testable fault f in C^τ can be transformed into that for a testable fault g, which corresponds to f, in C by σ.

Suppose that a sequential circuit S (see Figure 1(a)) is partitioned into a kernel circuit S_K and a set of external FFs Q_E (see Figure 1(b)). Let Q_I be the set of internal FFs and let d be the sequential depth of the kernel circuit. Let H be the d-clock hold transformation and let S^H be the sequential circuit obtained from S by H (see Figure 1(d)). Let T be the CCPT and let S^T be a circuit obtained from S by T (see Figure 1(c)). The test application method of S^H is the d-clock hold testing and that of S^T is the normal testing. We consider a new transformation HT which transforms S^H into S^T. The transformation HT changes each normal FF into a wire and also changes each d-clock hold FF into a normal FF. Let τ be any of transformations of H, T and HT. Let S be a sequential circuit (before transformation τ) and let S^τ be the circuit obtained from S by τ. We consider the fault mapping from the set of faults in S to that in S^τ.

Lemma 1 *Transformation τ maintains a one-to-one correspondence between a fault on each line in S and the fault on the same line in S^τ.*

Proof : Transformation τ changes some normal FFs into wires, changes normal FFs into d-clock hold FFs, or changes d-clock hold FFs into normal FFs. Let $\mathcal{F}$ be the set of all FFs in S such that any FF in $\mathcal{F}$ is changed into a wire by τ. Let F be the set of all faults in S and let F^τ be the set of all faults in $S\tau$. Let F_E be the subset of F such that any fault in F_E is neither

the input line fault nor the output line fault of FFs in $\mathcal{F}$. Therefore, each fault in F_E is mapped to the fault on the same line in S^τ. On the other hand, both the input fault and the output fault of an FF which is changed into a wire are represented by a fault of the wire. □

Let φ_τ be a fault mapping from the set of all faults in S to that in S^τ. Let φ'_τ be the one-to-one mapping from a fault on each line in S to the fault on the same line in S^τ. For transformation τ, we assume that φ_τ is restricted to φ'_τ.

Definition 12 Let S be a sequential circuit which is an acyclic structure. With regard to S, for any pair of a primary input and a primary output, if all the paths from the primary input to the primary output are of equal sequential depth, S is said to be a *balanced structure*[4]. □

If a sequential circuit is a balanced structure, the test generation problem for the circuit is reduced to the combinational circuit which is replaced each FF in the sequential circuit by a wire. We consider an application of this approach to general sequential circuits. Section 4 presents a test generation method based on CCPT.

Given a sequential circuit S, a test sequence t obtained by the test generation method based on CCPT is an input sequence for the circuit S^H which is transformed from the given circuit S by transformation H. Therefore, the obtained sequence t can be used as a test sequence for the circuit S^H. In order to clarify the reducibility between the test generation problem for a general circuit and that for the circuit transformed by H, we consider the reducibility between the test generation problem for a given circuit and that for each circuit transformed by each of transformations HT and T. The following theorems show the reducibility for these transformations in two cases: when a circuit whose kernel circuit is a balanced structure and when that is an acyclic structure.

Theorem 1 *Let S be a sequential circuit. If kernel circuit S_K in S is a sequential circuit which is a balanced structure, the test generation problem of S^H, which is obtained from S by transformation H, is reduced to the test generation problem of S^T, which is obtained from S by the transformation T.*

Proof : We show that transformation HT satisfies the two conditions (i) and (ii) of Definition 11.

(Definition 11-(i)) In order to show that the transformation HT is a testability preserving transformation, we show that the three conditions (i), (ii) and (iii) of Definition 10 are satisfied.

(Definition 10-(i)) From Lemma 1, it is obvious that there exists a one-to-one mapping from faults in S^H to those in S^T.

(Definition 10-(ii)) Let f^H be any fault of S^H and let f^T be a fault of S^T which corresponds to f^T. We show that if f^H is testable by the test application method of S^H, f^T is also testable by the test application method of S^T. If f^H is testable, there exists a test sequence t^H. The test sequence t^H is a d-clock hold sequence, because the test application method of S^H is d-clock hold testing. Let $(d+1)n$ be the number of vectors of t^H and let $O^H = o^H_1, o^H_2, \ldots, o^H_{(d+1)n}$ be an output sequence of S^H for t^H. Let $O^{H\prime} = o^H_{d+1}, o^H_{(d+1)2}, \ldots, o^H_{(d+1)n}$ be a sequence extracted from O^H at every $(d+1)$-th cycle. Let M be the d-clock hold sequence transformation and let M^{-1} be the inverse transformation of M. Note that the number of vectors in M^{-1} is n. Let $O^T = o^T_1, o^T_2, \ldots, o^T_n$ be an output sequence of S^T for $M^{-1}(t^H)$. Then, the equation $O^T = O^{H\prime}$ holds because the kernel circuit is a balanced structure. The error caused by f^T is observed in O^T because the error caused by f^H is observed in $O^{H\prime}$. Furthermore, there exists a one-to-one mapping between f^H and f^T (from Lemma 1). Thus f^T of S^T is testable by $M^{-1}(t^H)$ if the fault f^H of S^H is testable by t^H.

(Definition 10-(iii)) Let f^T be any fault of S^T and let f^H be a fault of S^H corresponding to f^H. We show that if f^T is testable by the test application method of S^T, f^H is also testable by the test application method of S^H. If f^T is testable, there exists a test sequence t^T. Let n be the number of vectors of t^H and let $O^T = o^T_1, o^T_2, \ldots, o^T_n$ be an output sequence of S^T if t^T is applied to S^T. The test sequence t^H must be a d-clock hold sequence, because the application method of S^H is d-clock hold testing. Let M be the d-clock hold sequence transformation and let $M(t^T)$ be the sequence obtained from t^T by M. Let $O^H = o^H_1, o^H_2, \ldots, o^H_{(d+1)n}$ be an output sequence of S^H for $M(t^T)$ and let $O^{H\prime} = o^H_{d+1}, o^H_{(d+1)2}, \ldots, o^H_{(d+1)n}$ be a sequence extracted from O^H at every $(d+1)$-th cycle. Then, the equation $O^{H\prime} = O^T$ holds because the kernel circuit is a balanced structure. The error caused by f^H is observed in $O^{H\prime}$ because the error caused by f^T is observed in O^T. Furthermore, there exists a one-to-one mapping between f^T and f^H from Lemma 1. Thus f^H of S^H is testable by $M(t^T)$ if f^T of S^T is testable by t^T.

(Definition 11-(ii)) Let f^H be any testable fault in S^H and let f^T be the testable fault corresponding to f^H in S^T. Let M be the d-clock hold sequence transformation. From Lemma 1, there exists a one-to-one mapping between f^H and f^T. Furthermore, from (Definition 10-(iii)), for $\mathcal{T}(f^T)$ which is the set of all test sequences of f^T by the test application method of S^T, $\bigcup_{t\in\mathcal{T}(f^T)}\{M(t)\}$ is a set of test sequences of f^H for the test application method of S^H. Thus there exists a sequence transformation M such that $\bigcup_{t\in\mathcal{T}(f^T)}\{M(t)\} \subseteq \mathcal{T}(f^H)$.

Hence the theorem is proved. □

Theorem 2 *Let S be a sequential circuit. If kernel circuit S_K in S is a sequential circuit which is an acyclic structure, the test generation problem of S^H,*

which is obtained from S by transformation H, is not always reduced to the test generation problem of S^T, which is obtained from S by transformation T.

Theorem 3 *If S_K is a sequential circuit which is a balanced structure, the test generation problem of S is not always reduced to the test generation problem of S^T.*

The proof of Theorem 2 and 3 are omitted here due to limitations of space. For further details, refer to [6].

Corollary 1 *If S_K is a sequential circuit which is an acyclic structure, the test generation problem of S is not always reduced to the test generation problem of S^T.*

4 Test Generation Method

In this section, we propose a test generation method using CCPT. We assume that a circuit structure of a kernel circuit is a balanced structure. The reason of this assumption is that if a kernel circuit is a balanced structure, the test generation problem of S^H can be reduced to the test generation problem of S^T (from Theorem 1). However, even if the kernel circuit is a balanced structure, the test generation problem of S can not always be reduced to the test generation problem of S^T (from Theorem 3). That is, the set of faults detected in S^T with a test sequence obtained by test generation for S^T differ from that in S by a test sequence obtained by test generation for S. We evaluate the difference between faults detected in S and those in S^T by experiments in Section 4.2.

4.1 Processes of Test Generation and Application

Let S be a given sequential circuit and let S_K be a balanced kernel circuit of S. Let Q_E be a set of external FFs and let Q_I be a set of internal FFs. Let d be the sequential depth of S_K. The test generation method using CCPT consists of the following four steps:

1. Partition S (see Figure 1(a)) into S_K and Q_E (see Figure 1(b)).
2. Transform S into a circuit S^T (see Figure 1(c)) by CCPT.
3. Generate a test sequence t^T for S^T by applying a test generation algorithm to S^T.
4. Transform the test sequence t^T generated at step 3 into a d-clock hold sequence t^H by d-clock hold sequence transformation.

The test application method corresponding to the test generation method using CCPT consists of two steps:

1. Transform S into a sequential circuit S^H (see Figure 1(d)) by the d-clock hold transformation.
2. Apply the sequence t^H to S^H and observe the response.

4.2 Experimental Results

In order to estimate the effectiveness of the proposed test generation method, we implemented it and experimented on test generation with ISCAS'89 benchmark circuits. In our experimentation, we used the FASTEST test generation algorithm[7] on the S-4/20 model 712 (Fujitsu) workstation.

Table 1: Circuit characteristics of ISCAS'89 benchmark circuits.

circuit	original				CCPT			
	#gate	#pi	#po	#ff	#i-ff	#e-ff	d	cpu(sec)
s382	99	3	6	21	6	15	1	0.5
s400	106	3	6	21	6	15	1	0.4
s444	119	3	6	21	6	15	1	0.5
s641	107	35	24	19	4	15	1	1.7
s713	139	35	23	19	4	15	1	2.2
s953	311	16	23	29	23	6	1	2.1
s1196	388	14	14	18	2	16	1	2.1
s1238	428	14	14	18	2	16	1	2.0
s1423	490	17	5	74	2	72	1	4.7
s5378	1004	35	49	179	55	124	2	44.7
s9234.1	2027	36	39	211	18	193	4	162.0
s9234	2027	19	22	228	18	210	4	161.5
s13207.1	2573	62	152	638	197	441	8	429.6
s13207	2573	31	121	669	198	471	8	418.7

A procedure of CCPT was implemented in a C language program. The program consists of the following three steps: finding a subcircuit which is an acyclic structure in a original benchmark circuit S as to find a MFVS (Minimum Feedback Vertex Set)[8] in S, finding a subcircuit which is a balanced structure in the acyclic subcircuit as a kernel circuit, and transforming S into S^H by replacing each internal FF of S with a wire. The second step of finding a balanced subcircuit is described in [4]. In our experiments, the second step of the program finds more simply a balanced subcircuit.

Table 1 shows circuit characteristics of ISCAS'89 benchmark circuits. The first column denoted by "circuit" shows circuit names. The second column denoted by "original" shows the numbers of gates, primary inputs and outputs and FFs of circuits before transformation. The third column denoted by "CCPT" shows the numbers of internal FFs and external FFs, sequential depths of kernel circuits (denoted by d) and CPU time (in seconds) required to CCPT. Note that the internal FFs are replaced with wires by CCPT.

Table 2 shows the experimental results of the FASTEST. Column #fault shows the number of faults. Column S shows the results of test generation for S and column S^T shows the results of test generation for S^T. For both cases, the fault coverage of generated test sequences is shown in column %f.cov.. Column cpu(sec) shows CPU time (in seconds) required to generate the test sequence. Column #vec. in column S shows the number of vectors (or length) of the generated test sequence and column #vec. in column S^T shows the number of vectors of the d-clock hold sequence which is transformed from the test sequence generated for S^T by the d-clock hold sequence transformation, i.e., [the number of vectors of generated test sequence for S^T] $\times$ [d+1]. From this experiment, with regard to the fault coverage and the test generation time, the result of S^T is better than the result of S for five circuits s382, s400, s444, s713 and s1423. The test generation time of S^T is less than

Table 2: Experimental results of FASTEST.

circuit	#fault	S			S^T			difference			
		#vec.	%f.cov.	cpu(sec)	#vec.	%f.cov.	cpu(sec)	$\#S_D$	$\#S_D^T$	$\#S_D S_{UD}^T$	$\#S_{UD} S_D^T$
s382	764	51	55.63	630	172	82.98	399	425	634	3	212
s400	800	51	54.63	680	140	81.00	388	437	648	3	214
s444	888	47	18.92	1680	174	79.73	410	168	708	0	540
s641	1278	146	87.32	48	318	87.25	53	168	708	0	540
s713	1426	137	83.10	234	282	83.24	233	1185	1187	0	2
s953	1906	12	7.92	20	14	7.82	14	151	149	3	1
s1196	2392	347	99.87	210	768	99.87	204	2389	2389	0	0
s1238	2476	368	96.65	555	668	99.65	515	2393	2393	0	0
s1423	2846	550	87.16	7298	954	88.97	5627	2481	2532	14	65
s5378	10590	856	78.57	154397	1605	77.75	168018	8821	8234	182	95
s9234.1	18468	51	10.02	236206	245	10.02	212121	1851	1851	0	0
s9234	18468	4	0.38	3170	20	0.38	3059	70	70	0	0
s13207.1	26358	117	11.64	226916	513	11.53	188256	3069	3041	40	12
s13207	26358	161	7.23	250668	702	6.84	153687	1906	1803	103	0

that of S for four circuits s1196, s1238, s9234.1 and s9234. In this case, the difference between the fault coverage of S and that of S^T is not observed. The test generation time of S^T is less than of S at two circuits s13207.1 and s13207. Their fault coverages are degraded a little. For the other circuits, the difference between the fault coverage of S and that of S^T is small.

Next, we estimate the difference between the set of faults detected in S and that in S^T. The last column denoted by "difference" in Table 2 shows the difference. The number of faults detected in S and in S^T are shown in column $\#S_D$ and $\#S_D^T$, respectively. Column $\#S_D S_{UD}^T$ shows the number of faults detected in S which is not detected in S^T. Inversely, column $\#S_{UD} S_D^T$ shows the number of faults not detect in S which is detected in S^T. $\#S_{UD} S_D^T$ is superior to $\#S_D S_{UD}^T$ for five circuits s382, s400, s444, s713 and s1423 that are improved by the proposed method for both the fault coverage and the test generation time. As an example, consider the circuits s400 and s713. For these circuits, all faults detected in S are also detected in S^T and furthermore several faults not detected in S are detected in S^T. For circuits s1196, s1238, s9234.1 and s9234, $\#S_{UD} S_D^T$ and $\#S_D S_{UD}^T$ are 0, i.e., detected faults are the same for both S and S^T. Therefore their fault coverage are equal.

As seen in the above observation, the proposed method increases the number of test vectors. However, in return for this disadvantage, the proposed method can reduce test generation time for most circuits and can increase fault coverage for several circuits.

5 Conclusions

In this paper, we have proposed a test generation method based on combinational circuit pseudo-transformation and have presented k-clock hold testing which is a test application method using a test sequence generated by the proposed method. We have considered test generation problems of an original circuit and its transformed circuits by combinational circuit pseudo-transformation and d-clock hold transformation, and also clarified the reducibility of those test generation problems. Furthermore, we estimated the effectiveness of the proposed method by experiments using benchmark circuits. The proposed method could reduce test generation time for most circuits and could increase fault coverage for several circuits.

Acknowledgments

We would like to thank Profs. Toshimitsu Masuzawa and Michiko Inoue for their valuable discussions.

References

[1] M. Abramovici, M. A. Breuer and A. D. Friedman: *Digital System Testing and Testable Design*, IEEE Press, 1995.

[2] A. Balakrishnan and S. T. Chakradhar: Software Transformations for Sequential Test Generation, *IEEE 4th Asian Test Symposium*, pp. 266–272, November 1995.

[3] G. D. Micheli: *Synthesis and Optimization of Digital Circuits*, McGraw-Hill, Inc., 1994.

[4] R. Gupta, R. Gupta and M. A. Breuer: The BALLAST Methodology for Structured Partial Scan Design, *IEEE Transactions on Computers*, Vol. 39, No. 4, pp. 538–544, April 1990.

[5] H. Fujiwara, S. Ohtake and T. Takasaki: Sequential Circuit Structure with Combinational Test Generation Complexity and Its Application (in Japanese), *IEICE*, Vol. J80-D-I, No. 2, pp. 155–163, February 1997.

[6] S. Ohtake, T. Inoue and H. Fujiwara: Sequential Test Generation Based on Circuit Pseudo-Transformation, Technical Report NAIST-IS-TR97014, Nara Institute of Science and Technology, July 1997.

[7] T. P. Kelsey, K. K. Saluja and S. Y. Lee: An Efficient Algorithm for Sequential Circuit Test Generation, *IEEE Transactions on Computers*, Vol. 42, No. 11, pp. 1361–1371, November 1993.

[8] S. T. Chakradhar, A. Balakrishnan and V. D. Agrawal: An Exact Algorithm for Selecting Partial Scan Flip–Flops, *Proceedings of 31th ACM/IEEE Design Automation Conference*, pp. 81–86, June 1994.