IEEE 8th Workshop on RTL and High Level Testing (WRTLT'07), pp. 17-22, October 2007.

An Extended Class of Acyclically Testable Circuits

Nobuya Oka[†], Chia Yee Ooi[‡], Hideyuki Ichihara[†], Tomoo Inoue[†], and Hideo Fujiwara[*]

[†]*Graduate School of Information Sciences, Hiroshima City University*
3-4-1, Ozuka-higashi, Asaminami-ku, Hiroshima, 731-3194 Japan

[‡]*Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Skudai Johor MALAYSIA*

[*]*Graduate School of Information Science, Nara Institute of Science and Technology, Kansai Science City 630-0192. Japan*

oka@dsgn.im.hiroshima-cu.ac.jp, {ichihara, tomoo}@hiroshima-cu.ac.jp, ooichiayee@fke.utm.my, fujiwara@is.naist.jp

Abstract

The class of acyclically testable sequential circuits [7] includes that of acyclic sequential circuits, and furthermore these classes are equivalent in test generation complexity. In this work, we consider an extension of the class of acyclically testable sequential circuits. To extend the class, we introduce a pair of justification and propagation thru trees, instead of the thru trees presented in [7]. Based on the thru tree pair, we propose an extended class of acyclically testable sequential circuits. While the new class properly includes the previous class, it is conjectured that the test generation complexity of the new class is equivalent to that of the previous one. Experimental results show that the DFT method based on the proposed class requires smaller overhead than the previous one with smaller test generation time.

Key words : test generation, acyclic testability, design-for-testability, combinational test generation complexity, τ^k notation.

1 Introduction

The test generation problem even for combinational circuits, was shown to be NP-complete [1], but empirical observations tell us that the test generation complexity of practically encountered combinational circuits seems to be polynomial[2]. On the other hand, the problem of test generation for sequential circuits is hard to be solved in practicable time. In many cases, the problem is converted into that for combinational circuits by full scan design. However, there exist some sequential circuits whose test generation is practically tractable as well as the test generation complexity of combinational circuits. Ooi *et al*. introduced that any sequential circuits can be classified by denoting $\tau = \Theta(n^r)$ (n is the size of the combinational circuit, r is some constant larger than 2) as the test generation complexity of combinational circuits[3, 4]. For example, the test generation complexity for the sequential circuits that have balanced structure or internally balanced structure is said to be τ-equivalent because it is equivalent to combinational test generation complexity[5, 6]. The test generation complexity of general acyclic sequential circuits is denoted by τ^2-bounded ($O(\tau^2(n))$). This means general acyclic sequential circuits are comparatively easily testable though they are more hardly testable than combinational circuits and the sequential circuits with balanced structure and internally balanced structure. In addition, the authors introduced a wider class called acyclically testable sequential circuits whose test generation complexity is τ^2-bounded. Note that this class is wider than the class of acyclic sequential circuits but its test generation complexity is equivalent to that of the acyclic sequential circuits [7]. They defined a thru tree whose its root is a primary output, its leaves are primary inputs and each of its edges represents a thru function. A sufficient condition of acyclical testability were defined based on the thru trees.

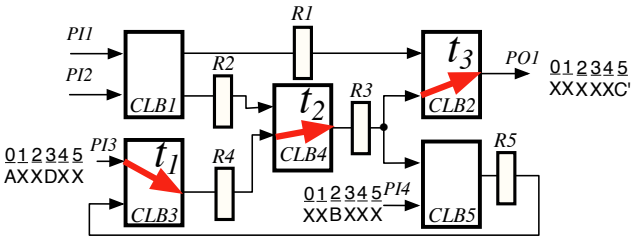

Figure 1. Sequential circuit S_1

In this work, we discuss an extension of the class of acyclically testable sequential circuits. We propose a pair of justification and propagation thru trees instead of the thru trees presented in [7]. Based on the thru tree pair, we introduce a new class of sequential circuits named *extended acyclically testable sequential circuits*. The class of extended acyclically testable sequential circuits properly includes that of acyclically testable sequential circuits. This implies that the DFT based on the extended acyclical testability requires smaller hardware overhead than the DFT based on the acyclical testability. Experimental results show that the DFT overhead based on the extended acyclical testability is smaller than that based on the previous class, while complete fault efficiency can be obtained.

2 Acyclically testable circuits

2.1 R-graph for sequential circuits

An example sequential circuit considered in this work is illustrated in Fig. 1. A sequential circuit is composed of combinational logic blocks(CLBs), registers and connections between CLBs and registers. Some CLBs have *thru* functions. Here we consider two types of thru functions as shown in Fig. 2. One is a *simple* thru: it transfers the data of an input of the CLB to the output independent of other inputs. The other is a *merge* thru: it binds the data of a certain number of inputs and transfers them to the output without modification.

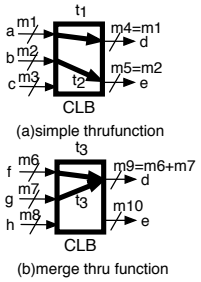

Figure 2. Thru function

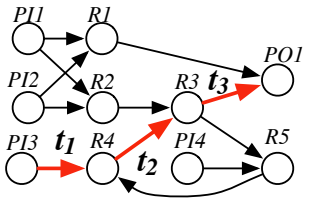

Figure 3. R-graph of sequential circuit S_1

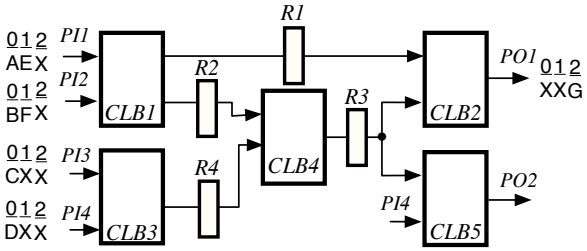

Figure 4. Sequential circuit S_2

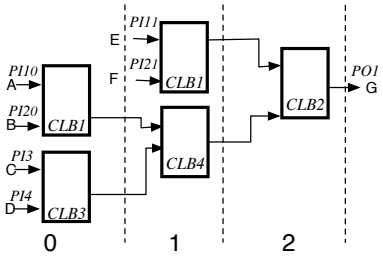

Figure 5. Time expansion model $C(S_2, CLB2)$ for $CLB2$ of S_2

A sequential circuit is represented by an R-graph defined as follows.

Definition 1 (R-graph): An R-graph is a directed graph $G_R = (V, A, w, r, t)$ that has the following properties.

1. A vertex v ($\in V$) corresponds to a register, a primary input or a primary output. Sets V_R, V_I and V_O are a set of registers, primary inputs and primary outputs, respectively. Then, the set V represents the union of V_R, V_I and V_O, i.e., $V = V_R \cup V_I \cup V_O$.
2. An arc (u,v) ($\in A$) denotes a connection via combinational logic blocks or directly from u corresponding to a register or a primary input to v corresponding to a register or a primary output.
3. The mapping $w(v)$ ($w: V \rightarrow Z^+$) denotes the bit width of v corresponding to a register, a primary input or a primary output, where Z^+ denotes a set of non-negative integers.
4. The mapping $r(v)$ ($r: v \rightarrow \{h, \phi\}$) defines the type of the vertex. If $r(v) = h$, v corresponds to a hold register. Otherwise if $r(v)$ is empty, i.e., $r(v) = \phi$, v corresponds to a register with no hold function, a primary input or a primary output.
5. For any arc (u,v), if (u,v) has a thru function t_i, the mapping $t(u,v)$ is denoted by $t(u,v) = t_i$. If there exists no thru function for (u,v), $t(u,v)$ is empty ($t(u,v) = \phi$), i.e., $t: V^2 \rightarrow F \cup \{\phi\}$, where $F = \{t_1, t_2, ..., t_m\}$ denotes a set of thru functions.

For example, Fig. 3 shows an R-graph for sequential circuit S_1.

If a thru function t_i is active according to the values of a subset V' of primary inputs and registers(i.e., $V' \subseteq V_I \cup V_R$), t_i is said to be *activated* by V', and expressed as $\alpha(t_i) = V_t$ ($\alpha: F \rightarrow 2^V$). If $\alpha(t_i)$ is empty (i.e., $\alpha(t_i) = \phi$), t_i is always activated.

2.2 The test generation complexity of acyclic sequential circuits

In [3, 4], the test generation complexity of combinational circuits is denoted by $\tau(n)$ ($= \Theta(n^r)$) (n is the size of the combinational circuit, and r is some constant larger than 2), and that of a sequential circuit is expressed by means of τ^k.

Based on this notation, the test generation complexity of acyclic sequential circuits is denoted by $O(\tau^2(n))$ (i.e., τ^2-bounded). This implies that the class of acyclic sequential circuits is comparatively easily testable. For example, consider an acyclic sequential circuit S_2 illustrated in Fig. 4. The test generation for S_2 can be performed on a time expansion model (TEM) as shown in Fig. 5. A TEM consists of CLBs and their connections, and represents the time when the output of each CLB is determined by other CLBs. Note that a TEM has no registers and thereby is a combinational circuit.

When a test pattern ($PI1_0$, $PI2_0$, $PI3$, $PI4$, $PI1_1$, $PI2_1$: $PO1$) = (A, B, C, D, E, F : G) is obtained from TEM $C(S_2, CLB2)$ for $CLB2$ of S_2 illustrated in Fig. 5, the corresponding test sequence for S_2 is ($PI1$, $PI2$, $PI3$, $PI4$, $PI5$: $PO1$, $PO2$) = $\langle$ $(A, B, C, D, X : X, X)$, $(E, F, X, X, X : X, X)$, $(X, X, X, X, X : G, X)$ $\rangle$. Here, the length of a TEM refers to the number of time frames (i.e., the difference between the maximum and minimum numbers labeled for CLBs plus 1) in the CLB. For example, the length of TEM $C(S_2, CLB2)$ in Fig. 5 is 3.

2.3 Acyclical testability

The previous work [7] introduced a class of *acyclically testable* sequential circuits whose test complexity is equivalent to that of acyclic sequential circuits, i.e., the complexity is τ^2-bounded. For example, sequential circuit S_1 in Fig. 1 is acyclically testable. R-graph of S_1 is illustrated in Fig. 3. Note that the R-graph has a cycle consisting of vertices $R3$, $R4$ and $R5$, i.e., it is not acyclic. In this R-graph, any of vertices $R3$, $R4$ and $R5$ can compose a feedback vertex set (FVS)[1]. Sequential circuit S_1 has a path from primary input $PI3$ to primary output $PO1$, consisting only of simple thrus

[1] For a direction graph $G = (V, A)$, a subset of the vertex set V' ($\subseteq V$) is a feedback vertex set if the subgraph of G induced by $V - V'$ is acyclic.

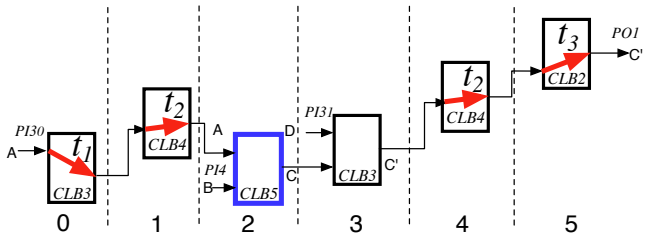

Figure 6. Time expansion model $C(S_1, CLB5)$ for $CLB5$ of S_1

t_1, t_2 and t_3. Accordingly, $R3$ and $R4$ can be justified from $PI3$, and can be observed at $PO1$. In other words, the path with thrus can be considered to be a scan path of $R3$ and $R4$, and consequently the test generation for S_1 can be achieved in the same way as acyclic sequential circuits. Such a thru path is referred to as a thru tree [7]. In [7], the sufficient conditions of acyclically testable is denoted with the concept of thru tree. One of the conditions is that an FVS is covered by the thru trees, e.g., in the R-graph in Fig. 3, the thru tree with thrus t_1, t_2 and t_3 covers an FVS $\{R3\}$. If a sequential circuit satisfies the sufficient conditions, a TEM for any CLB in the sequential circuit can be constructed. For example, a TEM $C(S_1, CLB5)$ for $CLB5$ of S_1 is shown in Fig. 6.

3 Extended class of acyclically testable sequential circuits

Let's consider sequential circuit S_3 in Fig. 7. Note that, unlike sequential circuit S_1 in Fig. 1, S_3 has no thru function in $CLB4$. Fig. 8 shows the R-graph for sequential circuit S_3. As we can see from Fig. 8, S_3 has no thru tree from $PI3$ to $PO1$. Accordingly it is not acyclically testable. However, we can obtain a TEM for any CLB. A TEM for $CLB5$ $C(S3, CLB5)$ is shown in Fig. 9.

For a given TEM, a test sequence generated for a fault in $C(S,B)$ must correspond to a test sequence for the corresponding fault in S. A test pattern generated on $C(S_3, CLB5)$ can be transformed to a test sequence. When a test pattern $(PI1_0, PI2_0, PI3_0, PI4, PI1_1, PI2_1, PI3_1 : PO1) = (E, F, A', B, G, H, D : C'')$ is obtained on TEM $C(S_3, CLB4)$ for $CLB5$ of S_3 illustrated in Fig. 9, the corresponding test sequence for S_3 is $(PI1, PI2, PI3, PI4 : PO1) = \langle (E, F, A', X : X), (X, X, X, B : X), (G, H, D, X : X), (X, X, X, X : X), (X, X, X, X : C'') \rangle$.

This example implies that an FVS in an R-graph does not always need to be covered by thru trees whose root and leaves are a primary output and primary inputs, respectively. In the R-graph in Fig. 8, two FVSs $R4$ and $R3$ are covered by the two *partial* thru trees, one contains just thru functions t_1 and the other contains t_3. Hence, any value of register $R4$ can be justified with thru function t_1, and any value of register $R3$ can be propagated to primary output $PO1$ with thru function t_3. Such justification and propagation are sufficient for testing CLBs.

In the following discussion, we present justification thru trees and propagation thru trees in place of the thru trees defined in [7], and propose an extended class of acyclically testable sequential circuits.

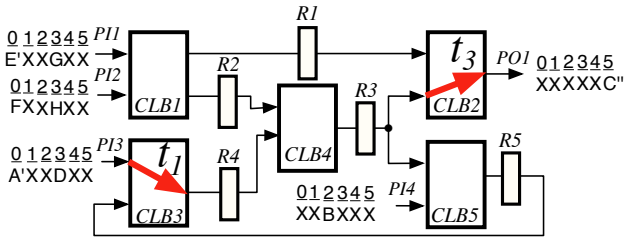

Figure 7. Sequential circuit S_3

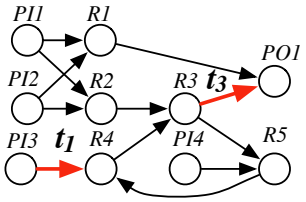

Figure 8. R-graph of S_3

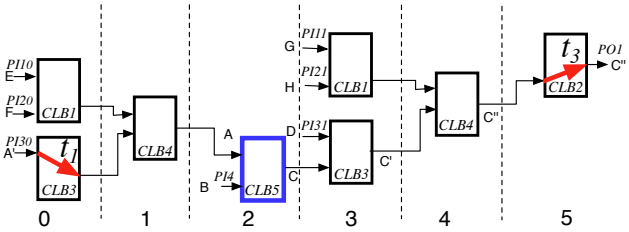

Figure 9. Time expansion model $C(S_3, CLB5)$ for $CLB5$ of S_3

3.1 Thru trees for justification and propagation

For any cycle, if there exists a thru path for justification to any register on the cycle and a thru path for propagation from any register on the cycle, a TEM for testing any CLB can be derived. Here, a pair of thru trees are defined as follows.

Definition 2 (Justification thru tree and propagation thru tree): Let $G_R = (V, A, w, h, t)$ be an R-graph. A justification thru tree $T^J = (V^J, A^J), (V^J \subseteq V, A^J \subseteq A)$ is a subgraph of G_R that satisfies the following conditions.

1. Any leaf v $(\in V^J)$ corresponds to a primary input, i.e., $v \in V_{PI}$.
2. Any arc (u, v) $(\in A^J)$ has a thru function, i.e., $t(u, v) \neq \phi$.
3. All inputs of a merge thru function are included in T^J or no input is includes in T^J, i.e., $\forall t_i \subseteq F$, $(t^{-1}(t_i) \cap A^J = t^{-1}(t_i)) \vee (t^{-1}(t_i) \cap A^J = \phi)$.

A propagation thru tree $T^P = (V^P, A^P), (V^P \subseteq V, A^P \subseteq A)$ is a subgraph of G_R that satisfies the following conditions.

1. The root v $(\in V^P)$ corresponds to a primary output, i.e., $v \in V_{PO}$.
2. Any arc (u, v) $(\in A^P)$ has a thru function , i.e., $t(u, v) \neq \phi$.

Definition 2 implies that any vertex (or a register) in a justification thru tree T^J can be justified from primary inputs, and any vertex (or a register) in a propagation thru tree T^P can be observed at a primary output. Note that justification and propagation with thru trees can be regarded as scan-in and scan-out operation, respectively. Therefore, if justification (resp. propagation) thru trees cover an FVS for an R-graph, we can justify (resp. observe) at least one register on every cycle in the sequential circuit corresponding to the R-graph.

In the R-graph of sequential circuit S_3 (Fig. 8), justification and propagation thru trees are $T^J = (V^J = \{PI1, R4\}, A^J = \{(PI1, R4)\})$ and $T^P = (V^P = \{R3, PO1\}, A^P = \{(R3, PO1)\})$, respectively. Note that the FVS for the R-graph is corresponding to that for R-graph of S_1. Vertices $R4$ and $R3$ are included in thru trees T^J and T^P, respectively. Accordingly registers $R3$ and $R4$ in the FVS can be justified from primary input and the registers can be observed from a primary output. Based on this property, TEM $C(S_3, CLB5)$ for $CLB5$ of S_3 can be obtained as shown in Fig. 9. Note that if arc $(PI4, R5)$ has a thru function, the subgraph $(\{PI4, R5\}, \{(PI4, R5)\})$ can be another justification tree, and hence a different TEM can be available.

When a justification thru tree is identical to a propagation one, i.e., $T^J = T^P$, the thru tree corresponds to the thru tree defined in [7]. Thus, a class of pairs of justification and propagation thru trees includes that of the thru trees defined in [7].

3.2 Extended acyclically testable sequential circuits

A thru function for a sequential circuit is activated according to values in registers or primary inputs. These thru tree relationship is denoted as follows.

Definition 3 (Dependency between thru trees): Let $T_i = (V_i, A_i)$ and $T_j = (V_j, A_j)$ be thru trees of G_R. Let thru tree $T_i = (V_i, A_i)$, $D(T_i)$ denote a set of registers or primary inputs that activate a thru function t ($\in T_i$), i.e., $D(T_i) = \cup_{a \in A_i} \alpha(t(a))$. If there exists a vertex v that activates a thru function $t (\in T_i)$, i.e., $v \in D(T_i)$, T_i depends on vertex v. And, for two thru trees T_i and T_j, if the union of $D(T_i)$ and V_j is not empty, i.e., $D(T_i) \cap V_j \neq \phi$, T_i depends on T_j and this relation is represented as $T_i \prec T_j$.

For a sequential circuit with justification and propagation thru trees, if the sequential circuit has a dependency between thru trees, there is a possibility that thru functions cannot be activated in some TEMs.

Let us consider sequential circuit S_3 illustrated in Fig. 7. Suppose that thru function t_1 is activated by the value of $PI2$ ($\alpha(t_1) = PI2$), that is, the justification thru tree $T^J = (V^J = \{PI1, R4\}, A^J = \{(PI1, R4)\})$ depends on primary input $PI2$. Here, let us consider that registers $R2$ and $R4$ are justified simultaneously. Because of $\alpha(t_1) = PI2$, primary input $PI2$ needs a particular value to justify $R4$ with thru function t_1. On the other hand, to justify $R2$, another particular value may be needed on primary input $PI2$. Thus, these values on primary input $PI2$ may be conflict, and consequently some test patterns for $CLB4$ may not be justified.

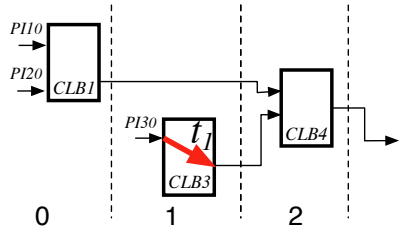

Figure 10. Time expansion model $C(S_3, CLB4)$ for $CLB4$ of S_3

Some dependencies of thru trees can be solved with a hold function of a register. For example, in S_3 shown in Fig. 7, if registers $R2$ or $R4$ have a hold function, input to $CLB1$ and activation for thru function t_1 can be implemented in separate time frames. As a result, TEM $C(S_3, CLB4)$ can be obtained as shown in Fig. 10. Whether the value confliction should be adjusted with hold functions depends on not only the dependency of thru trees but also the dependency of paths. The definition of path dependency is as follows.

Definition 4 (Dependency between paths): Let $p = (u_1, u_2, ..., u_{n_p})$ and $q = (v_1, v_2, ..., v_{n_q})$ be paths in an R-graph. If the following conditions are satisfied, p depends on q.

1. The lengths of paths p and q are equal, i.e., $n_p = n_q (= n)$.
2. The end vertices on paths p and q are the same, i.e., $u_{n_p} = v_{n_q}$.
3. The first arc (u_1, u_2) of paths p has a thru function, i.e., $t(u_1, u_2) \neq \phi$.
4. The start vertices on paths p and q are the same or the thru function of arc (u_1, u_2) is activated by vertex v_1 on q, i.e., $v_1 \in \alpha(t(u_1, u_2))$.

Based on the above definitions, a sufficient conditions of extended acyclically testable sequential circuits is as follows.

Definition 5 (a set of thru tree pair in sequential circuit): For an R-graph $G_R = (V, A, w, r, t)$ of a sequential circuit S, if S has a set of justification thru trees $\boldsymbol{T}^J = \{T_i^J = (V_i^J, A_i^J), i = 1, 2, ..., n^J\}$ and propagation thru trees $\boldsymbol{T}^P = \{T_i^P = (V_i^P, A_i^P), i = 1, 2, ..., n^P\}$, and hold registers such that satisfies the following conditions, S is said that S has a set of thru tree pair $(\boldsymbol{T}^J, \boldsymbol{T}^P)$. Here, $\boldsymbol{T} = \boldsymbol{T}^J \cup \boldsymbol{T}^P$, $V_T^J = \cup_{i=1}^{n^J} V_i^J$, $V_T^P = \cup_{i=1}^{n^P} V_i^P$.

1. The set of justification thru trees $\boldsymbol{T}^J$ satisfies the following conditions.
   (a) All thru trees in $\boldsymbol{T}^J$ are disjoint, i.e., $\forall i, j (i \neq j), V_i^J \cap V_j^J = \phi$.
   (b) A vertex set V_T^J includes all vertices composing at least one FVS, i.e., for any FVS $V' (\subseteq V)$, $V' \subseteq V_T^J$.
   (c) Any vertex $v (\in V_T^J)$ corresponds to a hold register, i.e., $\forall v \in V_T^J, r(v) = h$.

2. The set of propagation thru trees $\boldsymbol{T}^P$ satisfies the following conditions.
   (a) All thru trees in $\boldsymbol{T}^P$ are disjoint i.e., $\forall i, j(i \neq j), V_i^P \cap V_j^P = \phi$.
   (b) Vertex set V_T^P includes all the vertices composing at least one FVS, i.e., for any FVS, $V'(\subseteq V)$, $V' \subseteq V_T^P$.
   (c) For any vertex $v(\in V_T^P), v$ corresponds to a hold register, i.e., $\forall v \in V_T^P, r(v) = h$.
3. Let $T_k = (V_k, A_k)$ be a justification thru tree included in $\boldsymbol{T}^J$. Thru tree T_k satisfies the following conditions.
   (a) A thru tree T_k doesn't depend on itself($T_k \not\prec T_k$).
   (b) Any vertex $v \in D(T_k)$ on which T_k depends is included in other justification thru trees or V_{PI}, i.e., $\forall v \in D(T_k), v \in (V_T^J - V_k) \cup V_{PI}$.
4. For any justification thru tree $T^J(\in \boldsymbol{T}^J)$ and $T \neq T^J(\in \boldsymbol{T})$ satisfy the following conditions.
   (a) Thru tree T^J depends on any vertex that doesn't depend on T, i.e., $D(T^J) \cap D(T) = \phi$.
   (b) If T depends on T^J, T^J doesn't depend on T, i.e., $T \prec T^J \Rightarrow T^J \not\prec T$.
5. Let $p = (u_1, u_2, ..., u_{n_p})$ and $q = (v_1, v_2, ..., v_{n_q})$ be any pair of paths that satisfy the following conditions.
   - In any path of p and q, any cycle is simple [2].
   - Path p depends on q.
   - Let u_{i_p} $(1 < i_p < n_p)$ be a vertex. The path $(u_1, ..., u_{i_p})$ is included in a justification thru tree, and the subsequent arc $(u_{i_p}, u_{i_{p+1}})$ is not included in any justification thru tree. Path p includes no subpath $(u_{j_p}, ..., u_{k_p})$ $(i_p < j_p \leq k_p < n_p)$ such that vertex u_{j_p} is included in a justification thru tree and u_{k_p} is included in a propagation thru tree, i.e., $u_{j_p} \in V_T^J$ and $u_{k_p} \in V_T^P$.
   - Path q includes no subpath $(v_{j_q}, ..., v_{k_q})$ $(1 < j_q \leq k_q < n_q)$ such that vertex v_{j_q} is included in a justification thru tree and v_{k_q} is included in a propagation thru tree, i.e., $v_{j_q} \in V_T^J$ and $v_{k_q} \in V_T^P$.

   Either p or q, denoted by $\hat{p} = (\hat{u_1}, \hat{u_2}, ..., \hat{u_n})$, includes a vertex $\hat{u_k}$ such that :
   (a) for a subpath $p' = (\hat{u_k}, \hat{u_{k+1}}, ..., \hat{u_n})$ of $\hat{p}$, the length $|p'|$ of p' is equal to or longer than the length $|p''|$ of a distinct path $p'' = (\hat{u_k}, ..., \hat{u_{n_p}})$, and
   (b) a vertex $\hat{u_k}$ has a hold function , i.e., $r(\hat{u_k}) = h$.

[2] A cycle (v_1, v_2,..., v_k), where $v_1 = v_k$, is simple if v_2, v_3,..., v_k are distinct.

Table 1. Circuit information

name	#FF	area of register	area of CLB
ex1	40	680	1984
ex2	96	1632	2357
lwf	48	816	3966

If a sequential circuit satisfies Definition 5, the sequential circuit is said to be extended acyclically testable sequential circuit. If the product of the maximum height of thru trees in $\boldsymbol{T}^J$ and $\boldsymbol{T}^P$ and the depth of dependency in the set of thru trees $\boldsymbol{T} = \boldsymbol{T}^J \cup \boldsymbol{T}^P$ is a constant, Definition 5 corresponds to the definition of acyclical testability except in regard to thru trees. Furthermore, the thru tree defined in [7] is also any of justification and propagation thru trees. Therefore, we have the following theorem.

Theorem 1: The class of extended acyclically testable sequential circuits is a super set of the class of acyclic sequential circuits[7].

Based on the above, it is possible to conjecture as follows.

Conjecture : Test generation complexity of extended acyclically testable sequential circuits is τ^2-bounded.

This proof is a future work.

4 Case studies

We show the effectiveness of the proposed class by our experiments, using SUN blade2000 workstation (UltraSPARC-III+, CPU 1.02GHz, Memory 2GB). For three sequential circuits shown in Table 1, we attempted three types of DFTs: full-scan design (FS), acyclically testable design (AT) and extended acyclically testable design (EAT). In FS, all registers are replaced with scaned-FFs. In AT and EAT, some thru functions are added to CLBs so that a given circuit becomes acyclically testable and extended acyclically testable, respectively. After the DFTs, we generated a TEM for each CLB in each sequential circuit. Note that, in FS, a TEM for a CLB corresponds to the CLB. Target faults of each circuit were faults in each CLB and wires between elements in the circuits. In a TEM, some CLBs appear more than once, e.g., $CLB1$ appears at time frames 0 and 1 in the TEM as shown in Fig. 5. Therefore, a single fault in a CLB of a sequential circuit is mapped to a multiple fault in the corresponding TEM. In our experiments, we converted such a multiple fault into a single fault by the method of [8] and employ TetraMax (Synopsys, Inc.), which is a combinational TPG for single stuck-at faults. The circuit size was estimated provided that the area size of an inverter is one. We achieved 100% fault efficiency for all the circuits with three DFTs.

Table 2 shows area overhead, test application cycle, number of test patterns, test generation time and total size of TEMs. TEMs are performed by the definition [7], for AT and EAT. Column area overhead shows the difference of the circuit size before and after the DFTs. The test application cycle shown in the fourth column for FS is calculated by $(number\ of\ test\ patterns) \times ((number\ of\ FFs) + 1) + (number\ of\ FFs))$, and that for AT and EAT is calculated by $(number\ of\ test\ patterns) \times (length\ of\ TEM)$. Column

Table 2. Case studies

name	method	area overhead	test application cycle	number of test pattern	test generation time(s)	total size of TEMs
ex1	FS	280	2336	56	0.02	1984
	AT	73	336	56	0.06	1984
	EAT	49	348	58	0.22	2700
ex2	FS	672	4073	41	0.06	2357
	AT	96	280	56	0.12	2357
	EAT	48	288	48	1.02	4343
lwf	FS	336	3527	71	0.20	3966
	AT	145	492	82	0.37	3966
	EAT	97	426	71	1.23	5484

TEM area denotes the sum of the size of TEM for each CLB in a circuit. Note that the size of a TEM does not include that of the CLBs on thru trees because the root (a leaf) of a justification (propagation) thru tree can be regarded as a primary input (output).

From Table 2, we can see the followings. The area overhead for AT and EAT is smaller than that for FS. Moreover, our method, EAT, needs smaller area overhead than AT. This is because the number of thru functions required by EAT is less than that of AT. The test application cycle for AT and EAT is smaller than that for FS since AT and EAT do not need scan operation. The test application cycle for EAT is comparable to AT.

Three DFTs are comparable in the number of test patterns. For *ex2* and *lwf*, the number of test patterns of EAT can be small as compared with AT. The reason is that TEMs of EAT is often implemented justification and propagation of values without using thru functions. Because the number of thru functions required for EAT is less than that of AT. Therefore, the one test pattern of EAT can find many faults compared with that of AT.

For ex2, although the number of test patterns for EAT is smaller than AT, the test application cycle for EAT is not. This is because the sum of the length of TEMs for EAT is greater than that for AT. The total size of TEMs for EAT is larger than that for AT. In EAT, instead of thru functions, the logic of CLBs is used for justification and propagation as frequently as possible. As a result, the size of TEMs for EAT is larger than that for AT, so that the test generation time for EAT increases. However, since a TEM is a combinational circuit, we can achieve complete fault efficiency with practical time.

5 Conclusion

In this work, we proposed the extension of the class of acyclically testable sequential circuits whose the test generation complexity is τ^2-bounded. In the new class, we defined justification and propagation thru trees instead of the thru trees presented in [7] and proposed the sufficient conditions of the class. Case study show that the area overhead of extended acyclically testable sequential circuits is smaller than that of the acyclically testable sequential circuits, and complete fault efficiency is achieved as well as the acyclically testable sequential circuits.

Our future work is evaluation and experiment of larger circuits, analysis of test generation complexity of extended acyclically testable sequential circuits. We are now making further experiments for large circuits. We can expect that the experimental results also show the effectiveness of our improvement. The final manuscript will include the experimental results.

Acknowledgments

The authors would like to thank the members of Computer Design and Test Laboratory, Nara Institute of Science and Technology, Japan and Dr. Yuki Yoshikawa and other members of Computer Design Laboratory, Hiroshima City University, Japan for their valuable comments.

References

[1] P. Goel, "Test generation costs analysis and projections," Proc. 17th DAC, pp. 77-84, June 1980.

[2] M.R. Prasad, P. Chong, and K. Keutzer, "Why is ATPG easy?" Proc. 36th DAC, pp. 22-28, June 1999.

[3] C. Y. Ooi and H. Fujiwara, "Classification of sequential circuits based on τ^k notation," IEEE Proc. 13th Asian Test Symp., pp.348-353, Nov. 2004.

[4] C. Y. Ooi, T. Clouqueur and H. Fujiwara, "Classification of sequential circuits based on τ^k notation and its applications", IEICE Trans. on Info. and syst., pp.2738-2747, Dec. 2005.

[5] R. Gupta and M. A. Breuer, "The BALLAST methodology for structured partial scan design," IEEE Trans. Comput., vol. 39, no. 4, pp. 538-544, April 1990.

[6] H. Fujiwara, "A new class of sequential circuits with combinational test generation complexity," IEEE Trans. Comput., vol. 49, no. 9, pp. 895-905, Sept. 2000.

[7] C. Y. Ooi and H. Fujiwara, "A new class of sequential circuits with acyclic test generation complexity," IEEE Proc. ICCD, pp.425-431, Oct. 2006.

[8] H. Ichihara, and T. Inoue, "A method of test generation for acyclic sequential circuits using single stuck-at fault combinational ATPG," IEICE Trans. Fundamentals, Vol. E86-A, No. 12, pp. 3072-3078, Dec. 2003.