

PAPER

Non-scan Design for Testability for Synchronous Sequential Circuits Based on Fault-Oriented Conflict Analysis*

Dong XIANG[†], Shan GU[†], *Nonmembers*, and Hideo FUJIWARA^{††}, *Fellow*

SUMMARY A two stage non-scan design for testability method is proposed. The first stage selects test points based on an earlier testability measure *conflict*. A new design for testability algorithm is proposed to select test points by a fault-oriented testability measure *conflict+* in the second stage. Test points are selected in the second stage based on the hard faults after the initial ATPG run of the design for testability circuit in the preliminary stage. The new testability measure *conflict+* based on conflict analysis of hard-faults in the process of test generation is introduced, which emulates most general features of sequential ATPG. The new testability measure reduces testability of a fault to the minimum D or \bar{D} controllability of the primary outputs, and therefore, does not need observability measure any more. Effective approximate schemes are adopted to get reasonable estimation of the testability measure. A couple of effective techniques are also adopted to accelerate the process of the proposed design for testability algorithm. Experimental results show that the proposed method gets even better results than two of the recent non-scan design for testability methods *nscan* and *lcdft*.

key words: *at-speed test, conflict, containing assignment, non-scan design for testability, sequential depth for testability*

1. Introduction

Scan design places scan flip-flops into one or more scan chains. Much more test application time is necessary due to shifting tests and test responses through scan chains. Also tests cannot be applied at the speed of operational clock. Test efficiency and fault coverage parameters of at-speed test should be more dependable than those of scan design circuits [1].

1.1 Previous Work

Design for testability only based on structure information cannot obtain satisfactory results. It is believed that an effective testability measure is necessary to select test points for non-scan design for testability. SCOAP [6] has been widely used for more than

two decades, which has been proved to be a success. However, it is found to be ineffective to analyze testability of hard-to-test circuits with complex reconvergent fanout structures. The k -level controllability/observability measure for RTL circuits [3] indicates the number of clock cycles required to control or observe a data path. The k -level controllability/observability measure still did not consider influences of reconvergent fanouts. Test multiplexers were selected based on the testability improvement potential of the k -level controllability and observability. Extra control lines of the test multiplexers were connected with PI ports to avoid equal weight reconvergent fanouts. Ghosh and Jha [5] extracted testability from the CDFG (control data flow graph), which was not influenced by the width of data paths. Test multiplexers were placed into uncontrollable places. Chakravarty et al. [1] proposed a testability measure to estimate testability of a faulty circuit with multiple faults based on conditional probabilities. The proposed method is a refinement of an earlier measure called PREDICT. Consistent assignments were obtained to reduce computing effort of the method. Detectability of a fault is the D - (or \bar{D}) controllability of the fault at the primary outputs. Williams and Angell [14] considered the use of test points in conjunction with additional logic to provide an easy means to control or observe the state of a sequential circuit. For a circuit modified in this way, test generation reduces to that of a combinational circuit with only a single extra pin. Fujiwara [4] showed that computing complexity of exact testability estimation for even 3-level monotone or unate combinational circuits is NP-complete. Therefore, it is impossible to get accurate testability estimation for general sequential circuits in reasonable time.

1.2 Motivations

Almost all previous testability measures partition testability parameters into two separate parts: (1) controllability and (2) observability. However, they are closely interdependent because the assignments for controllability and assignments for observability are specified on the same circuit simultaneously. Savir pointed out that interdependence of controllability and observability

Manuscript received January 6, 2003.

Manuscript revised June 10, 2003.

[†]The authors are with the School of Software, Tsinghua University, Beijing 100084, P. R. China.

^{††}The author is with the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630-0101 Japan.

*This work is in part supported by JSPS under grant L03540 and NSF of China under grant 69773030.

ity should be considered early in [13]. A testability measure can present more accurate testability estimation if only one testability parameter is utilized. This is one of the most important motivations of the paper. Drivability has found to be an effective fault-oriented measure to guide fault effect propagation path selection in Gentest [2]. The drivability measure is actually an extension of the SCOAP testability measure, therefore, it still did not include influences of reconvergent fanouts. Almost all of previous testability measures estimate testability in fault-free circuits for simplicity. The drivability measure is a good example to evaluate testability of the faulty circuit. Chakravarty et. al. [1] is another good example to estimate testability of faulty circuits. Testability analysis in faulty circuits presents more accurate and reasonable testability if the computing amount is acceptable. Design for testability based on this kind of testability measures can present better testability improvement. This is another one of the most important motivations of the paper. Two of the recent best sequential test generators ATOMs [7] and MIX+ [10] utilized conflict oriented search.

The *conflict* measure [15], [18] intensively checks influences of reconvergent fanouts on testability of a sequential circuit. A couple of techniques, such as, inversion parity, sequential depth for testability, and fanout stem partitioning, are utilized to estimate testability of a circuit. Interdependence between fault effect activation and propagation is included in the *conflict* measure. Anyway, the *conflict* measure still evaluates controllability and observability measures separately. Xiang and Xu [17] utilized a single-parameter testability measure called TIP based on valid state information to guide scan flip-flop selection, where the information of reconvergent fanouts are included in the valid state information effectively. However, the TIP measure can only be used to select scan flip-flops.

A new testability measure is proposed with respect to hard faults. The measure is calculated based on the popular 9-valued logic system, which was widely utilized by the important test generators, such as, HITEC [12], FASTEST [8], ATOMs [7], MIX [9], and MIX+ [10]. The 9-valued logic system is $\{O, I, u0, u1, 1u, 0u, D, \bar{D}, U\}$, where each element represents a 2-tuple (a, b) , where a is the fault-free value, and b is the faulty value. All 9 values represent $(0,0)$, $(1,1)$, $(x,0)$, $(x,1)$, $(1,x)$, $(0,x)$, $(1,0)$, $(0,1)$, and (x,x) , respectively. The 9-valued system can always present more accurate testability information than the usual 3-valued logic system. Intensive conflict analysis of the reconvergent fanouts is presented. The new measure *conflict+* is a fault-oriented one, which uses only controllability measures. Testability of a fault is D - or \bar{D} -controllability on primary outputs. The proposed measure *conflict+* should be more accurate than the drivability measure [2] because it includes influences of reconvergent fanouts and more accurate logic system is

used. However, the *cpu* consumption of the new measure should be comparable to that of the drivability [2]. It should also be more accurate than *conflict* measure because it naturally evaluates testability of a fault by a single metric but not controllability and observability measures separately like most of the previous testability measures. A new design for testability algorithm based on the new measure is introduced, which is completely different from the conventional ones and can obtain more accurate testability improvement.

In the rest of this paper, definitions and notation are introduced in Sect. 2. Conflicts during sequential ATPG are studied in Sect. 3. The new testability measure is presented in Sect. 4. The controllability domination relation is illustrated in Sect. 5. A new design for testability algorithm is proposed in Sect. 6. Sufficient experimental results are presented in Sect. 7. The paper is concluded in Sect. 8.

2. Preliminaries

We present some definitions and notation of the paper first, and then a simple introduction of the *conflict* measure and the framework of the design for testability method.

2.1 Notation and Definitions

A *signal requirement* is a 2-tuple (A, v) , which means a node A is required to be assigned a value v , where $v \in \{1, 0\}$. The *non-controlling value* v of inputs of a gate with an output y is that the value of y can be determined only when all inputs are set v ; the output y of the gate can be determined if only one of its inputs is set the *controlling* value. v -Controllability (v is one of the 9 values) of line l indicates the number of potential conflicts may occur or the number of clock cycles required to justify a signal requirement (l, v) . No observability is necessary in the *conflict+* measure because testability of a fault is the D or \bar{D} controllability on the primary outputs in the faulty circuit.

Definition 1: A *conflict* is defined as follows: A line l in a faulty circuit is assigned value v , in the previous process of test generation, l needs to be assigned value v' . If intersection of v and v' produces a new covered value, the line l is assigned $v \cap v'$; otherwise, a conflict occurs on l , where $v, v' \in \{O, I, u0, u1, 1u, 0u, 1u, D, \bar{D}, U\}$.

Definition 2: Inversion parity of a path is defined as the number of inversions in the path modulo 2. Inversion parity $inv_{v_1}(A, B)$ ($v_1 \in \{0, 1\}$) between two nodes is defined as inversion parity information of the path set from B to A in order to justify the signal requirement (B, v_1) .

The main cause of conflicts is still reconvergent

fanouts with nonuniform inversion parities. It is impossible to enumerate all those paths between A and B in a very large sequential circuit, therefore, a simplified metric is utilized to do that. Inversion parity $inv_{v_1}(A, B)$ from node B to A is represented by a two binary bit number in this paper: (1) 00, (2) 01, (3) 10, (4) 11, which means: (1) there is no path from A to B or no signal requirement on node A in order to meet signal requirement (B, v_1) , (2) justify (B, v_1) passes only a path of odd inversion parity from A to B , (3) justify (B, v_1) passes only a path of even inversion parity from A to B , (4) justify (B, v_1) passes at least one path of even inversion parity and one path of odd inversion parity from A to B , respectively.

Definition 3: Sequential depth for testability $seq_{v_1}(l, s)$ ($v_1 \in \{0, 1\}$) from a fanout stem s to a line l is defined as the number of clock cycles required to justify a signal requirement (l, v_1) at the line l to the fanout stem s .

It should be noted that calculations of inversion parity and sequential depth for testability are completely the same as those in *nscan*[15], [18] and *lcdft*[16]. When $seq_{v_1}(l, s) = 0$, it indicates that justification of the signal requirement (l, v_1) has no signal requirement on the fanout stem s or passes no flip-flop. It should be noted that sequential depth for testability is quite different from sequential depth that considers only the circuit structure. Calculation of inversion parity includes testability consideration. Therefore, we define $inv_{v_1}(l, s)$ as the inversion parity between l and s in order to set value v_1 on l .

It should be noted that $seq_0(l, s)$ and $seq_1(l, s)$ are not always the same, and $seq_0(l, s)$ and $seq_1(l, s)$ are both set as 0 when l is unreachable from s . $Seq_{v_1}(l, s)$ can also be 0 if signal requirement (l, v_1) can be met in an easier way without having any signal requirement on the fanout s . When a cycle is met, iterative calculation of the sequential depth for testability may be necessary.

Definition 4: We call an *assignment* (a_1, a_2, \dots, a_n) for inputs of a block (a gate or a functional unit) is dominated by another assignment (b_1, b_2, \dots, b_n) if a_i is dominated by b_i for $i = 1, 2, \dots, n$. An assignment (a_1, a_2, \dots, a_n) is a *containing assignment* if there is no assignment (b_1, b_2, \dots, b_n) such that (b_1, b_2, \dots, b_n) is dominated by (a_1, a_2, \dots, a_n) , and both assignments set output of the block into the same value v .

Definition 5: We call the *controllability domination relation* holds for a line l only if for any pair of values v and v' , and v contains v' , we have $C_l(v) \leq C_l(v')$.

3. Conflicts during Test Generation

The *conflict* measure in [15], [18] is enhanced to a fault-oriented measure called *conflict+* based on the pop-

Table 1 The intersection table.

A \ B		U	u0	u1	0u	1u	D	\overline{D}	O	I	
		U	u0	u1	0u	1u	D	\overline{D}	O	I	
		u0	u0	c	O	D	D	c	O	c	
		u1	c	u1	\overline{D}	I	c	\overline{D}	c	I	
		0u	0u	O	\overline{D}	0u	c	c	\overline{D}	O	c
		1u	1u	D	I	c	1u	D	c	c	I
		D	D	D	c	c	D	D	c	c	c
		\overline{D}	\overline{D}	c	\overline{D}	\overline{D}	c	c	\overline{D}	c	c
		O	O	O	c	O	c	c	c	O	c
		I	I	c	I	c	I	c	c	c	I

ular 9-valued logic system. Several important sequential test generators, such as, HITEC [12], FASTEST [8], ATOMs [7], MIX [9] and MIX+ [10] adopt the 9-valued logic system as illustrated in Sect. 2, which can relax the fault effect propagation conditions and obtain more actual fault coverage. We use two separate intersection tables to deal with the conflict analysis problem. The intersection table for lines which are reachable from the fault line is presented in Table 1 based on the 9-valued logic system. The intersection table based on the 3-valued logic system ($\{0, 1, \times\}$) should be used for lines that are unreachable from the fault point. As for lines that are unreachable from the fault point, they are unable to be assigned values $D, \overline{D}, u1, u0, 1u$ and $0u$. According to the intersection table for the 9-valued logic, $u0 \cap 1u = D$, there is no conflict. For a line that is unreachable from the fault point, $O \cap I$ generates a conflict.

We would like to show the necessity to adopt different logic systems for lines that are reachable from the fault point or unreachable from the fault point. Let us consider propagation of the fault effect of the fault $s/0$ as presented in Fig. 1 along the EFEP (easiest fault effect propagation) path $s-d-e-f-h-i$. The lines a, c, b and g should be assigned values $1u, u0, u1$ and $0u$, respectively. No conflict occurs at the line a because $seq_1(a_2, a) \neq seq_0(c, a)$. No conflict occurs at the line b because b is reachable from the fault point s , therefore, b can be assigned $u1 \cap 0u = \overline{D}$. Consider propagating the fault effect of the fault $e/1$ along the path $e-f-h-i$. The fault can be activated via the primary input a . The fanout stem b is unreachable from the fault point e . The intersection of 1 and 0 is a conflict according to the 3-valued logic system. Actually, the fault $e/1$ is redundant.

Assume B_1 and B_2 are two branches stemming from the same node, and B_2 is assigned value I , and B_1 is assigned $u0$. The intersection should produce a conflict,

$$(1, 1) \cap (\times, 0) = (1, \phi),$$

where “ \cap ” is an intersection operator and ϕ indicates a contradictory assignment. When a node has been assigned I , and then it is required to be assigned O ,

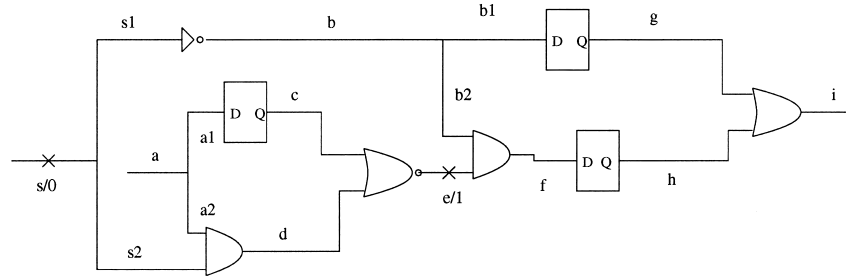


Fig. 1 Necessity for separate intersection tables.

$$I \cap O = (1, 1) \cap (0, 0) = (\phi, \phi).$$

The intersection generates a conflict. Assume a node has been assigned $1u$, and then it is required to assign O .

$$1u \cap O = (1, \times) \cap (0, 0) = (\phi, 0),$$

a conflict occurs. For a pair of values A and B , we call A dominates B if $A \cap B = A$. It indicates that $C_l(A) \geq C_l(B)$ for a specific line l . For example, for a specific line l , we have,

$$C_l(I) \geq C_l(1u) \geq C_l(U). \tag{1}$$

4. Calculation of *conflict+*

Theorem 1: The containing assignments for a specific value v are enough in order to calculate v -controllability of a block.

Proof: Because the *conflict+* measure is a SCOAP-like testability measure, testability estimation for each line considers only the easiest assignments. For each assignment (c_1, c_2, \dots, c_n) of value v , suppose (c_1, c_2, \dots, c_n) is not a containing assignment. There is a containing assignment (a_1, a_2, \dots, a_n) of value v , which is dominated by (c_1, c_2, \dots, c_n) . That is, a_1 contains c_1 , a_2 contains c_2 , ..., a_n contains c_n , respectively. Therefore, (a_1, a_2, \dots, a_n) is easier to be justified than (c_1, c_2, \dots, c_n) because a_1, a_2, \dots, a_n are easier to be justified than c_1, c_2, \dots, c_n , respectively. It is unnecessary to include the assignment (c_1, c_2, \dots, c_n) in the testability calculation expression of the block. So, to control the node to value v should only consider the containing assignments. \square

For example, while we calculate $0u$ -controllability of the output of an AND gate, we can only consider the containing assignments $(U, 0u)$ and $(0u, U)$. The details to obtain containing assignments for any value and any types of gates or functional units will not be presented in this paper for simplicity.

Theorem 2: The *conflict+* measures for different hard faults in the hard fault set have the following properties,

- assume $v \neq D$ or \bar{D} , v -controllability of a specific line l corresponding to different faults in the hard fault set are the same;
- let faults f_1 and f_2 be on the lines l_1 and l_2 , respectively, and a line l be unreachable from l_1 and l_2 , l has the same v -controllabilities (v is any one of the 9 values) corresponding to the faults f_1 and f_2 .

Proof: The expressions for v -controllabilities do not include D or \bar{D} if $v \neq D$ or \bar{D} . Therefore, v -controllabilities for a specific line are always the same corresponding to different hard faults, $v \in \{U, u1, u0, 1u, 0u, O, I\}$.

Because l is unreachable from both of l_1 and l_2 , the D - (or \bar{D}) controllabilities of the line l corresponding to f_1 and f_2 are ∞ . And $C_l(v)$'s are the same for any $v \neq D, \bar{D}$ according to the above paragraph. \square

Consider a line l is unreachable from the fault line, we have, $C_l(u0) = C_l(0u) = C_l(O)$, $C_l(1u) = C_l(u1) = C_l(I)$, and $C_l(D) = C_l(\bar{D}) = \infty$. Calculation of $C_l(O)$ and $C_l(I)$ are similar to those of *conflict* [15], [18]. We only consider lines that are reachable from the fault line when calculating testability measures corresponding to a fault. This technique can save a lot of *cpu* time.

Lemma 1: Let l be the fault line with a fault $l/1$, we have $C_l(D) = C_l(O) = C_l(u0) = \infty$, $C_l(I) = C_l(1u)$, $C_l(\bar{D}) = C_l(0u)$ and $C_l(u1) = 0$.

Lemma 2: Let l be the fault line with a fault $l/0$, we have $C_l(D) = C_l(1u)$, $C_l(O) = C_l(0u)$, $C_l(\bar{D}) = C_l(u1) = C_l(I) = \infty$ and $C_l(u0) = 0$.

The implication tables based on the 9-valued logic system for 2-input AND, 2-input OR gates and the INVERTER are presented as shown in Table 2. Assume A and B are inputs of an AND gate with an output l . A or B should be assigned value O in order to assign O to l , that is, assignments (O, v) and (v, O) , where v in any one of the 9 values. Other 8 assignments can also control l as value O . They are $(\bar{D}, u0)$, $(0u, u0)$, (\bar{D}, D) , $(0u, D)$, $(u0, \bar{D})$, $(u0, 0u)$, (D, \bar{D}) , $(D, 0u)$ as shown in Table 2. There are only 4 containing assignments $(0u, u0)$, $(u0, 0u)$, (O, U) and (U, O) in order to control l as value O . There will be no conflict while justifying the above 4 assignments. We do not need

Table 2 Implication tables based on the 9-valued system: (a) AND gate, (b) OR gate, (c) INVERTER.

		A								
		I								
B		U	u0	u1	0u	1u	D	\bar{D}	O	I
U	U	u0	U	0u	U	u0	0u	O	U	
u0	u0	u0	u0	O	u0	u0	O	O	u0	
u1	U	u0	u1	0u	U	u0	\bar{D}	O	u1	
0u	0u	O	0u	0u	0u	O	0u	O	0u	
1u	U	u0	U	0u	1u	D	0u	O	1u	
D	u0	u0	u0	O	D	D	O	O	D	
\bar{D}	0u	O	\bar{D}	0u	O	\bar{D}	O	\bar{D}		
O	O	O	O	O	O	O	O	O	O	
I	U	u0	u1	0u	1u	D	\bar{D}	O	I	

(a)

		A								
		I								
B		U	u0	u1	0u	1u	D	\bar{D}	O	I
U	U	U	u1	U	1u	1u	u1	U	I	
u0	U	u0	u1	U	1u	D	u1	u0	I	
u1	u1	u1	u1	u1	I	I	u1	u1	I	
0u	U	U	u1	0u	1u	1u	\bar{D}	0u	I	
1u	1u	1u	I	1u	1u	1u	I	1u	I	
D	1u	D	I	1u	1u	D	I	D	I	
\bar{D}	u1	u1	u1	\bar{D}	I	I	\bar{D}	\bar{D}	I	
O	U	u0	u1	0u	1u	D	\bar{D}	O	I	
I	I	I	I	I	I	I	I	I	I	

(b)

		A	
		I	
B		U	I
U	U	U	
u0	u0	u1	
u1	u1	u0	
0u	0u	1u	
1u	1u	0u	
D	D	\bar{D}	
\bar{D}	\bar{D}	D	
O	O	I	
I	I	O	

(c)

to penalize O -controllability at the output of a 2-input AND gate.

In order to control value I to the output of the AND gate, A and B should be assigned value I . While justifying the assignment, potential conflicts may occur. Assignments (D, I) , (D, D) , $(D, 1u)$, (I, D) and $(1u, D)$ can set the output l of a 2-input AND gate as value D according to Table 2. Because (D, I) and (D, D) dominate $(D, 1u)$, and (I, D) dominates $(1u, D)$, we have containing assignments $(1u, D)$ and $(D, 1u)$ for D -controllability of line l . Line l can be controlled as value \bar{D} by assignments (\bar{D}, I) , $(\bar{D}, u1)$, (\bar{D}, \bar{D}) , (I, \bar{D}) and $(u1, \bar{D})$. Assignments (\bar{D}, I) and (\bar{D}, \bar{D}) dominate $(\bar{D}, u1)$, and assignment (I, \bar{D}) dominates $(u1, \bar{D})$, we can only consider assignments $(u1, \bar{D})$ and $(\bar{D}, u1)$ for \bar{D} -controllability of the line l . The D -controllability and \bar{D} -controllability are quite similar to the drivability adopted by the Gentest algorithm [2]. However, the drivability is an extension of the SCOAP testability measure, which did not include any influences of reconvergent fanouts. The following expressions are used to calculate controllability of lines reachable from the fault point. D -(or \bar{D})controllability of lines unreachable from the fault point are ∞ . Let $v \in \{O, I, u0, u1, D, \bar{D}, 0u, 1u\}$ in the rest of this subsection. If l is a primary input, $C_l(v) = 0$. Assume l is a fanout branch stemming from s , we have,

$$C_l(v) = C_s(v).$$

Let l be the output of an AND gate with inputs A and B . There are four different containing assignments (O, U) , (U, O) , $(0u, u0)$, and $(u0, 0u)$ that set l to value O , we have,

$$C_l(O) = \min(C_A(O), C_B(O), C_A(0u) + C_B(u0), C_A(u0) + C_B(0u)). \quad (2)$$

There exist two containing assignments $(u0, U)$ and $(U, u0)$ that sets l to value $u0$,

$$C_l(u0) = \min(C_A(u0), C_B(u0)). \quad (3)$$

There is only one containing assignment (I, I) which sets l to value I ,

$$C_l(I) = C_A(I) + C_B(I) + p, \quad (4)$$

where $p = n \cdot 10$, n is the number of reconvergent fanouts s with $inv_1(A, s) \neq inv_1(B, s)$ and $seq_1(A, s) = seq_1(B, s)$. There are two containing assignments $(1u, D)$ and $(D, 1u)$ that set l to value D , we have,

$$C_l(D) = \min(C_A(1u) + C_B(D), C_A(D) + C_B(1u)) + p. \quad (5)$$

Testability estimation for other values are presented as follows.

$$C_l(u1) = C_A(u1) + C_B(u1) + p. \quad (6)$$

$$C_l(\bar{D}) = \min(C_A(u1) + C_B(\bar{D}), C_A(\bar{D}) + C_B(u1)) + p. \quad (7)$$

$$C_l(0u) = \min(C_A(0u), C_B(0u)). \quad (8)$$

$$C_l(1u) = C_A(1u) + C_B(1u) + p. \quad (9)$$

Let l be the output of an OR gate with inputs A and B , containing assignments of all assignments can be obtained similarly from Table 2. We have,

$$C_l(O) = C_A(O) + C_B(O) + p,$$

$$C_l(I) = \min(C_A(I), C_B(I), C_A(1u) + C_B(u1), C_A(u1) + C_B(1u)),$$

$$C_l(u0) = C_A(u0) + C_B(u0) + p,$$

$$C_l(u1) = \min(C_A(u1), C_B(u1)),$$

$$C_l(D) = \min(C_A(D) + C_B(u0), C_A(u0) + C_B(D)) + p,$$

$$C_l(\bar{D}) = \min(C_A(0u) + C_B(\bar{D}), C_A(\bar{D}) + C_B(0u)) + p,$$

$$C_l(0u) = C_A(0u) + C_B(0u) + p,$$

$$C_l(1u) = \min(C_A(1u), C_B(1u)).$$

Let l be the output of an INVERTER with input I ,

$$C_l(v) = C_l(\overline{v}),$$

where $\overline{0} = I$, $\overline{u0} = u1$, $\overline{D} = \overline{D}$, $\overline{0u} = 1u$. If l is the output of a D flip-flop with input i ,

$$C_l(v) = C_i(v) + 10.$$

Calculations of other types of gates are similar. The *conflict+* measure is a hard-fault-oriented testability measure. D - and \overline{D} -controllability measures of the lines which are unreachable from the fault point are ∞ , whose observability is also 0. It should be noted that the *conflict+* measure still considers potential conflicts for the value of a gate that needs to assign controlling values on the inputs of the gate.

We do not need observability in the *conflict+* measure any more. We can use controllability measures to represent testability of a fault. Let l/v ($v \in \{0, 1\}$) be the fault line, we have,

$$\begin{aligned} test(l/v) = \min & (C_{po_1}(D), C_{po_1}(\overline{D}), \dots, \\ & C_{po_m}(D), C_{po_m}(\overline{D})), \end{aligned} \quad (10)$$

where po_1, po_2, \dots, po_m are primary outputs, and $test(l/v)$ is the testability of fault l/v .

The proposed testability measure is a fault-oriented one, calculation of which should be time-consuming if it is calculated based on separate faults. Effective approximate techniques are utilized to estimate the testability measure. First, conflict information of the *conflict* measure [15], [18] is adopted to calculate the *conflict+* measure. It is shown that calculation of the *conflict* measure can be completed in no more than half an hour for all the iscas89 circuits. The conflict information can be used for testability measures corresponding to all faults although the faulty circuits with respect to different faults are different. It should be noted that the expressions for controllability measures of a line with respect to values I , O , $u1$, $1u$, $u0$, and $0u$ have nothing to do with D and \overline{D} , therefore, the controllability measures on the values are the same for all faulty circuits. The *conflict+* testability measure corresponding to one fault only handles lines that are reachable from the fault, which needs less time than that of SCOAP.

The above approximate schemes may still get inaccurate estimation. As shown in Fig. 2, $seq_0(h, s) = 0$ and $seq_1(g, s) = 1$, $inv_0(h, s) = 00$ and $inv_0(g, s) = 01$. No conflict occurs at the fanout stem s in order to justify signal requirement $(i, 0)$ in the fault-free circuit. Consider the faulty circuit with a fault $b/1$. $seq_0(h, s) = 1$ and $seq_0(g, s) = 1$, $inv_0(h, s) = 10$ and $inv_0(g, s) = 01$ in this case. Actually, a conflict occurs when justifying a signal requirement $(i, 0)$. Therefore, using calculated inversion parity and sequential depth for testability in the fault-free circuit may still get inaccurate estimation. However, the above approximate schemes are really effective and can get accurate estimation in most cases.

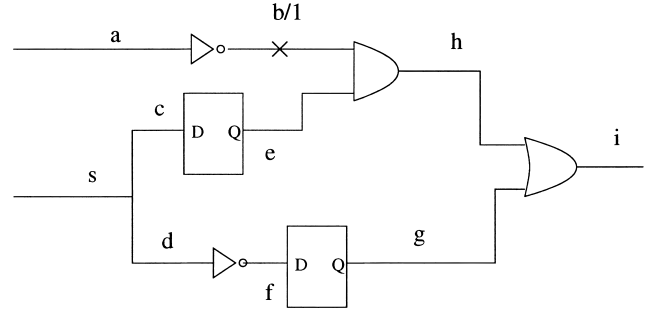


Fig. 2 Inaccuracy of the approximate schemes in faulty circuits.

5. The Controllability Domination Relation

Recall that we call the *controllability domination relation* holds for a line l only if for any pair of values v_1 and v_2 , and v_1 contains v_2 , we have $C_l(v_1) \leq C_l(v_2)$.

Theorem 3: The controllability domination relation of an AND gate holds for the output y according to the controllability calculation as illustrated in Sect. 4 if the relation holds for all inputs of the gate. That is to say, let v_1 dominate v_2 , if $C_{in}(v_1) \geq C_{in}(v_2)$ for each input, we have $C_y(v_1) \geq C_y(v_2)$.

Proof: Lines in a faulty circuit can be classified into 2 separate types: (1) lines reachable from the fault point, (2) lines unreachable from the fault point. We would like to prove the theorem by induction. We need to prove a 2-input AND gate first. Assume A and B are two inputs of an AND gate with output y , we want to prove the following statement: let v_1 dominate v_2 , $C_y(v_1) \geq C_y(v_2)$ if $C_A(v_1) \geq C_A(v_2)$ and $C_B(v_1) \geq C_B(v_2)$. We should check 2-tuples $(O, u0)$, $(O, 0u)$, $(I, u1)$, $(I, 1u)$, $(D, u0)$, $(D, 1u)$, $(\overline{D}, u1)$, and $(\overline{D}, 0u)$ for lines that are reachable from the fault line. As for lines unreachable from the fault point, only the first 4 2-tuples should be checked because the D -controllability and \overline{D} -controllability of these lines are ∞ . It should be noted that $p_1 \leq p_2 \leq p_3$. Consider the 2-tuple $(\overline{D}, 0u)$, we have,

$$\begin{aligned} C_y(0u) &= \min(C_A(0u), C_B(0u)) \\ C_y(\overline{D}) &= \min(C_A(u1) + C_B(\overline{D}), C_A(\overline{D}) + C_B(u1)) \\ &\quad + p_2, \end{aligned}$$

according to Eqs. (5) and (8).

- If $C_y(\overline{D}) = C_A(u1) + C_B(\overline{D}) + p_1$ and $C_y(0u) = C_A(0u)$,

$$C_y(\overline{D}) = C_A(u1) + C_B(\overline{D}) + p_1 \geq C_A(u1) + C_B(0u) \geq C_A(u1) + C_A(0u) \geq C_y(0u).$$
- If $C_y(\overline{D}) = C_A(u1) + C_B(\overline{D}) + p_1$ and $C_y(0u) = C_B(0u)$, we have,

$$C_y(\overline{D}) = C_A(u1) + C_B(\overline{D}) + p_1 \geq C_B(\overline{D}) \\ \geq C_B(0u) = C_y(0u).$$

- If $C_y(\overline{D}) = C_A(\overline{D}) + C_B(u1) + p_1$ and $C_y(0u) = C_A(0u)$,

$$C_y(\overline{D}) = C_A(\overline{D}) + C_B(u1) + p_1 \geq C_A(0u) \\ + C_B(u1) \geq C_B(0u) + C_B(u1) \geq C_B(0u) \\ = C_y(0u).$$

- If $C_y(\overline{D}) = C_A(\overline{D}) + C_B(u1) + p_1$ and $C_y(0u) = C_A(0u)$,

$$C_y(\overline{D}) = C_A(\overline{D}) + C_B(u1) + p_1 \geq C_A(\overline{D}) \\ \geq C_A(0u) = C_y(0u).$$

The controllability domination relation for other 2-tuples can be proved similarly. Up to now, we have proved that for a 2-input AND gate with inputs A, B and output y reachable from the fault point, for any value pair v_1 and v_2 , v_1 contains v_2 , $C_y(v_1) \leq C_y(v_2)$ if $C_A(v_1) \leq C_A(v_2)$ and $C_B(v_1) \leq C_B(v_2)$. Actually, we have also proved Theorem 2 holds true for an output of an AND gate unreachable from the fault point.

Assume the controllability domination relation holds for an n -input AND gate, we need to prove the controllability domination relation also holds for an $(n+1)$ -input AND gate. We can do the following transformation for an $(n+1)$ -input AND gate with inputs $1, 2, \dots, n, n+1$ and output y . The $(n+1)$ -input AND gate is transformed into an n -input AND gate with inputs $1, 2, \dots, n$ and an output A , where A and $n+1$ are inputs of a 2-input AND gate with output y . According to the premises, the controllability domination relation holds for the output A of the n -input AND gate. The controllability domination relation also holds for the output y of the two input AND gate with inputs A and $n+1$. \square

Corollary 1: When controllability domination relation of inputs of an OR, NOR, NAND, INVERTER or a D flip-flop holds, the controllability domination relation also holds for the output of any one of them.

Theorem 4: The controllability domination relation holds for all lines of the circuit.

Proof: It is easy to know that the domination relation holds for all the primary inputs. It is clear that the controllability domination relation also holds for outputs of any types of gates or D flip-flops if it does for all of its inputs. Controllability calculation for a cycle is similar in each iteration. And the controllability domination relation holds for all lines in the cycle. Then the controllability domination relation can be extended gate by gate and step by step till the primary outputs. \square

6. A New Design for Testability Algorithm

As shown in Fig. 3, let a 0-control test point be inserted into node a . The bold-faced lines are those that

get changed controllability based on the selective tracing scheme and the *conflict* measure in [15], [16]. A new scheme is adopted to estimate testability gain. The *active fault set* is defined as faults with changed testability (with respect to *conflict+*). (i) Initially, all hard faults that reach line a should be included in the active fault set. For each successor b of a , check all faults that reach b . If the fault gets changed D - or \overline{D} -controllability measure at line b , put the fault into the active fault set of the line b . Continue the above process until out of the bold-faced range. (ii) Drive all active faults of the nodes just outside of bold-faced range until the active fault set of the line is empty or a primary output is reached. No active faults are added during the second phase.

Hard-to-detect faults and their predecessors and successors are considered as test point candidates. The following heuristics are used to check active faults for a node:

- Active faults of its predecessors are active fault candidates of the node.
- All faults that reach the node should be candidates of active faults.
- All faults with unchanged D and \overline{D} controllability should be deleted from the active fault set.
- All faults with both D and \overline{D} -controllability at the node greater than the testability of the fault should be deleted from the active fault set.

First, all hard faults that reach a successor of the line should be considered as active fault candidates. All active faults of its predecessors should be active fault candidates. An active fault candidate should be excluded if its D -controllability and \overline{D} -controllability at that line are unchanged. An active fault candidate should be deleted if its D -controllability and \overline{D} -controllability at the line are greater than its testability. Testability gain is estimated according to testability of all active faults F at all primary outputs. The updated testability $test'(f)$ of a hard fault f after a control test point has been inserted is,

$$test'(f) = \min(C_D(po_1), C_{\overline{D}}(po_1), \dots, C_D(po_k), \\ C_{\overline{D}}(po_k)), \quad (11)$$

$$gain(l) = \sum_{f \in F} [test(f) - test'(f)]. \quad (12)$$

The testability gain after a control test point has been inserted into the line is the summation of testability improvement of all hard faults. In Eq. (12), $test(f)$ is the testability of fault f in the original circuit. It is quite interesting to estimate testability gain when an observation point is inserted into a line. The testability gain can be estimated according to testabilities of the set of all faults F_1 that reach the node,

$$test(f, l) = \min(C_D(l), C_{\overline{D}}(l)). \quad (13)$$

Let $test(f, l) < test(f)$, testability gain after an observation point is inserted into l can be obtained as

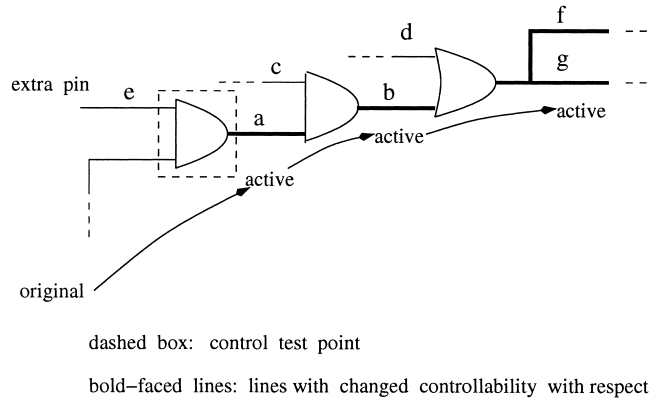


Fig. 3 Illustration of the design for testability scheme.

follows,

$$gain(l) = \sum_{f \in F_1} [test(f) - test(f, l)]. \quad (14)$$

It is quite easy to estimate testability gain of an observation point, which can be obtained from the local information and does not need any algorithm to calculate testability improvement like that of a control point. An observation point never changes testability of a fault with respect to any other nodes. An observation point can change testability of a fault with respect to the whole circuit.

It should be time-consuming if testability gains of all test point candidates are recalculated after a test point is inserted. It is also unnecessary to estimate testability gain again for all lines after a control test point has been inserted based on the *conflict+* measure because a test point only makes a limited range of lines get changed testability. The following scheme is adopted to handle the problem. Testability gain of each test point candidate should be estimated for the first control test point. Our method selects the node with the greatest testability gain to insert the corresponding test point. After the test point has been inserted, testability of a limited range in the circuit gets changed testability. Testability gains of lines get changed testability should be updated for the second test point while testability gains of the test point candidates not influenced by the inserted test point are not updated. It is found that the above technique can obtain good enough testability improvement. It should be noted that all bold-faced nodes as shown in Fig. 3 are nodes with changed testability after a control test point has been inserted into node *a*. The procedure to calculate testability of test point candidates get changed testability exactly the same as stated earlier. The above process should continue until all control points are inserted. The above technique can save *very much* cpu time for very large circuits compared with the procedure that updates testability improvement potentials of all test point candidates (with respect to the *conflict*

measure) after a control test point has been inserted.

Similar technique is adopted to select observation points. After an observation point has been inserted, testability gains of nodes that reach the observation point should be updated. The above techniques can also reduce cpu time drastically although testability gain for an observation point can be obtained directly from testability measures of the node with respect to the hard faults that reach it. It should be noted that inversion parity and sequential depth for testability for the related nodes should be updated after a control point has been inserted. Therefore, the testability measure *conflict+* calculates only once for the whole design for testability process. Our method calculates only the hard faults after the initial ATPG run of the preliminary phase design for testability circuit. The *conflict+* measure can thus be calculated in reasonable time. The following procedure presents the whole test point selection process.

Procedure *test-point-selection()*

1. Calculate the *conflict+* measure based on the hard fault set of the DFT circuit after the initial run of HITEC. Select test point candidates for control points based on the *conflict+* measure.
2. While (control point selection not completed)
 - a. For each test point candidate *c*, obtain the region *R* that gets changed *conflict* measure with the selective tracing scheme when a control point is inserted into *c*.
 - b. Drive the active fault set from *c* based on techniques introduced above until out of the region *R* with changed testability.
 - c. Drive the active faults until a primary output reaches or active fault set becomes empty based on techniques introduced above.
 - d. Get the testability gains according to Eqs. (11) and (12). Insert a control point with the most testability gain.
 - e. Update testability gains of nodes influenced by the inserted test point.

Table 3 Comparison of the proposed method with *nscan* [15], [18] and *lcdft* [16].

circuits	econ				lcdft			nscan		
	tp/po	FC/TE(%)	vec	ao(%)	tp/po	FC/TE(%)	vec	tp/po	FC/TE(%)	vec
s1423	40/2	94.1/95.0	274	5.3	–	–	–	40/2	93.6/94.6	607
s5378	60/2	97.5/99.5	2599	3.0	–	–	–	60/2	97.3/99.5	1337
s9234	158/1	96.0/97.7	2539	4.2	160/1	95.3/97.8	1760	160/3	92.8/95.7	3685
s9234.1	154/1	95.5/97.8	2304	4.1	160/1	95.4/97.8	1760	160/3	90.9/94.8	2946
s13207	240/1	96.3/99.4	5044	3.6	240/1	96.3/99.4	5044	240/3	91.8/94.9	3927
s13207.1	240/1	96.7/99.3	5059	3.7	240/1	96.7/99.3	5059	240/3	91.2/94.5	4023
s15850	280/2	94.1/98.3	3571	4.3	280/2	92.7/96.9	7008	240/3	94.2/97.6	8583
s15850.1	280/2	94.2/98.7	3584	4.3	280/2	92.7/96.5	2835	240/3	94.0/97.5	5151
s35932	235/3	93.8/100	504	1.9	235/3	90.1/100	269	235/3	91.3/100	248
s38417	800/3	91.2/93.3	6839	4.6	800/3	91.7/93.9	5988	800/3	82.8/85.2	2271
s38584	499/3	94.9/97.2	12367	3.3	500/3	92.7/96.9	10600	500/3	92.0/94.4	8207
s38584.1	450/3	94.8/97.0	12038	2.9	450/3	92.8/95.1	12510	450/3	93.0/95.5	10338
s1269	12/1	99.7/100	326	2.2	12/2	98.2/99.8	352	12/2	99.4/99.7	204
s1512	9/1	99.9/100	4938	1.2	12/1	99.0/99.0	3391	12/1	100/100	3224
s3271	15/1	99.6/99.8	263	0.9	15/1	99.8/100	298	9/1	99.6/99.6	688
s3330	60/2	95.3/96.3	751	4.0	60/2	94.3/94.7	857	60/2	92.5/92.8	817
s3384	40/1	97.9/97.9	166	2.2	40/1	96.6/96.7	137	40/2	98.3/98.5	180
s4863	9/1	98.6/99.1	419	0.3	9/2	98.6/99.0	439	9/2	98.5/98.5	391
s6669	9/1	99.8/99.8	252	0.2	9/1	99.9/99.9	257	9/2	99.9/99.9	327

3. Select observation points based Eqs. (13) and (14). Place observation points into exclusive-or trees and connect extra pins of control points using techniques in paper [16].

7. Experimental Results

The fault-oriented non-scan design for testability method has been implemented and run on an ultra10 workstation. The design for testability system is called *econ*. The proposed method is compared with *nscan* [15], [18] and *lcdft* [16], which are good non-scan design for testability methods proposed recently. The *nscan* [15], [18] illustrates a good measure called *conflict*, and cost-effective test point structures. Extra pins of the control test points are driven by the primary inputs based the test point structures. The *lcdft* emphasizes the techniques to connect extra pins of the control test points with the proper primary inputs in order to avoid the negative effects of the new reconvergent fanouts. More than one control points can be connected with the same primary inputs, which makes *nscan* and *lcdft* even outperforms the previous partial scan design tools on fault coverage. Certainly, much less test application cost and test power consumption are required than the scan design tools.

The preliminary design for testability selects test points based on *lcdft* [16] and the *conflict* measure. The number of test points inserted in the initial phase is mainly determined empirically for good enough fault coverage in order to make testability analysis cost in the second stage acceptable. The HITEC test generator does an initial run on the design for testability circuit after the preliminary DFT phase has been completed.

The initial run of the HITEC on the DFT circuit in the first stage should be unimportant compared with the cpu time for the original circuit or the final ATPG run of the DFT circuit. The final phase design for testability is based on the hard faults obtained from the initial run results of HITEC.

Table 3 presents the ATPG results of HITEC on the iscas circuits. The parameters *tp*, *po*, *ao*, *FC*, *TE*, and *vec* represent the number of test points, pin overhead, area overhead, fault coverage, test efficiency, and the number of test vectors generated using the HITEC test generator. It is shown that the proposed algorithm can effectively place observation points.

Table 3 presents comparison of *econ* with *lcdft* [16]. The proposed method *econ* generates better results on fault coverage than *lcdft* for all circuits except s38417, s3271, and s6669. The system *econ* generates a little worse results for circuits s38417, s3271, and s6669 than *lcdft*, and the same results for circuits s13207 and s13207.1. Both systems generate 91.2% and 91.7% fault coverage for s38417, respectively. The proposed method *econ* works very well on the above circuits because all the circuits need a couple of observation points to obtain good enough fault coverage. The *econ* is good at selecting observation points. Especially, *econ* and *lcdft* generate 94.1% and 92.7%, 94.2% and 92.7% fault coverage after 280 test points have been inserted into both circuits s15850 and s15850.1. After 235, 500, 450 test points are inserted into circuit s35932, s38584, and s38584.1, methods *econ* and *lcdft* generate 93.8% and 90.1%, 94.9% and 92.9%, 94.8% and 92.8% fault coverage, respectively. The proposed method and *lcdft* generate 99.7% and 98.2%, 99.9% and 99.0%, 95.3% and 94.3%, 97.9% and 96.6% fault coverages for cir-

circuits s1269, s1512, s3330, s3384, and s4863 after 12, 9, 60, 40 and 9 test points are inserted, respectively.

Performance comparison between *econ* and *nscan* [15], [18] is presented in Table 3. The proposed method *econ* generates better results for all circuits than *nscan* except s15850, s1512, s3384, and s6669. The new method *econ* gets slightly worse fault coverages on the above four circuits. The proposed system gets apparently better fault coverages than *nscan* on circuits s9234, s9234.1, s13207, s13207.1, s38417, s35932, s38584, s38584.1, and s3330. The proposed method *econ* obtains a little better fault coverage on circuits s1423, s5378, s1269, and s4863 than *nscan*. It is clear that *econ* works well on the hard-to-test circuits, where the hard-to-test circuits indicate the ones that performance (especially for fault coverage) of HITEC [12] and other deterministic test generators [2], [7]–[10] is very bad. The proposed method obtain much better results for the hard-to-test circuits s9234, s9234.1, s13207, s13207.1, s38417, and s3330 than *nscan*. The most important reason why *econ* works better than *nscan* on these circuits is that the above circuits need many control points to obtain good fault coverage. The proposed method and *lcdft* select control points until they are unable to be connected with the primary inputs while *nscan* selects test points exactly according to the *conflict* measure.

The proposed testability measure *conflict+* can be utilized to select the best sensitivity path in a circuit or in the backtrace procedure instead of the drivability measure in a sequential test generator like Gentest [2]. First, a simple testability measure, such as, the *conflict* measure [15], [18] or the SCOAP [6] can be used to guide test generation in the initial run. After that, the *conflict+* measures are calculated for the aborted faults after the initial ATPG run. The *conflict+* measure should work better for the hard faults than the previous measures or the drivability measure. Better fault coverage or test efficiency can be obtained. The *conflict+* measure can also be applied to other sequential test generators to select the fault effect propagation path, which can improve performance of sequential ATPG in most cases. This should be a direct application of the proposed *conflict+* measure.

8. Conclusions

A two-stage non-scan design for testability method was proposed based on a fault-oriented conflict analysis. In the initial phase, test points were selected based on the *conflict* measure [15], [18] and the selective tracing algorithm. Test points were selected using the new testability measure *conflict+*, and a new design for testability algorithm based on the hard fault set generated by the initial run of HITEC on the design for testability circuit of the preliminary stage. The following techniques were adopted, which make the proposed testability measure

demonstrates actual testability of a circuit: (i) The 9-valued logic system is utilized to calculate the testability measure, which is commonly adopted in several important sequential test generators, such as, HITEC, ATOMs, FASTEST, MIX and MIX+. Containing assignment was used to estimate testability, using which concise formulae were presented based on the 9-valued logic system. (ii) Fault effects were allowed to be naturally propagated along multiple paths unlike previous testability measures. (iii) Unlike conventional testability measures, *conflict+* does not need observability measure any more, where testability of a fault is the minimum D or \bar{D} -controllability measure of the fault at all primary outputs. Several good techniques were introduced to accelerate the design for testability procedure and testability estimation. Sufficient experimental results showed that the proposed design for testability method outperforms two recent good non-scan design for testability methods [15], [16].

References

- [1] S. Chakravarty and H.B. Hunt III, "On computing signal probability and detection probability of stuck-at faults," IEEE Trans. Comput., vol.39, no.11, pp.1369–1377, 1990.
- [2] W.T. Cheng and T.J. Chakraborty, "Gentest: An automatic test generation algorithm for sequential circuits," Computer, vol.22, no.4, pp.43–49, 1989.
- [3] S. Dey and M. Potkonjak, "Non-scan design for testability techniques using RTL design information," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.16, no.12, pp.1488–1506, 1997.
- [4] H. Fujiwara, "Computational complexity of controllability/observability problems for combinational circuits," IEEE Trans. Comput., vol.39, no.6, pp.762–767, 1990.
- [5] I. Ghosh and N.K. Jha, "Design for hierarchical testability of RTL circuits obtained by behavioral synthesis," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.16, no.9, pp.1001–1014, 1997.
- [6] L.H. Goldstein, "Controllability/observability analysis of digital circuits," IEEE Trans. Circuits Syst., vol.26, pp.685–693, 1979.
- [7] I. Hamzaoglu and J. Patel, "Deterministic test pattern generation techniques for sequential circuits," Proc. IEEE/ACM Int. Conf. Computer-Aided Design, pp.538–543, 2000.
- [8] T.P. Kelsey, K.K. Saluja, and S.Y. Lee, "An efficient algorithm for sequential circuit test generation," IEEE Trans. Comput., vol.42, no.11, pp.1361–1371, 1993.
- [9] X. Lin, I. Pomeranz, and S.M. Reddy, "MIX: A test generation system for synchronous sequential circuits," Proc. 11th Int. VLSI design Conf., pp.456–463, 1998.
- [10] X. Lin, I. Pomeranz, and S.M. Reddy, "Techniques for improving the efficiency of sequential circuit test generation," Proc. IEEE/ACM Int. Conf. on Computer-Aided Design, pp.147–151, 1999.
- [11] P.C. Maxwell, R.C. Aitken, V. Johansen, and I. Chiang, "The effect of different test sets on quality level prediction: When is 80% better than 90%?" Proc. IEEE Int. Test Conf., pp.358–364, 1991.
- [12] T. Niermann and J. Patel, "HITEC: A test generation package for sequential circuits," Proc. European Conf. on Design Automation, pp.214–218, 1991.

- [13] J. Savir, "Good controllability and good observability do not guarantee testability," *IEEE Trans. Comput.*, vol.32, no.12, pp.1198–1200, 1983.
- [14] M.J.Y. Williams and J.B. Angell, "Enhancing testability of large-scale integrated circuits via test points and additional logic," *IEEE Trans. Comput.*, vol.22, pp.46–60, no.1, 1973.
- [15] D. Xiang, Y. Xu, and H. Fujiwara, "Non-scan design for testability for synchronous sequential circuits based on conflict analysis," *Proc. IEEE Int. Test Conf.*, pp.520–529, 2000.
- [16] D. Xiang and H. Fujiwara, "Handling the pin overhead problem of DFTs for high-quality and at-speed test," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol.21, no.9, pp.1105–1113, 2002.
- [17] D. Xiang and Y. Xu, "A multiple phase partial scan design method," *Proc. 10th IEEE Asian Test Symposium*, pp.17–22, 2001.
- [18] D. Xiang, Y. Xu, and H. Fujiwara, "Non-scan design for testability for synchronous sequential circuits based on conflict resolution," *IEEE Trans. Comput.*, vol.51, no.8, pp.1063–1075, 2003.
- [19] D. Xiang and Y. Xu, "Partial reset for synchronous sequential circuits using almost independent reset signals," *Proc. IEEE VLSI Test Symposium*, pp.82–87, 2001.



Hideo Fujiwara received the BE, ME, and PhD degrees in Electronic Engineering from Osaka University, Osaka, Japan, in 1969, 1971, and 1974, respectively. He was with Osaka University from 1974 to 1985 and Meiji University from 1985 to 1993 and joined the Nara Institute of Science and Technology in 1993. In 1981, he was a visiting research assistant professor at the University of Waterloo, and, in 1984, he was a visiting research associate professor at McGill University, Canada. Presently, he is a professor in the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan. His research interests are logic design, digital system design and test, VLSI CAD, and fault-tolerant computing, including high level/logic synthesis, design for testability, built-in self-test, test pattern generation, parallel processing, and computational complexity. He is the author of *Logic Testing and Design for Testability* (MIT press, 1985). He received the IEICE Young Engineer Award in 1977, IEEE Computer Society Certificate of Appreciation Award in 1991, 2000, and 2001, Okawa Prize for publication in 2001. He is an advisory member of the IEICE Transactions on Information and Systems, Journal of Electronic Testing, Journal of Circuits, Systems, and Computers, Journal of VLSI Design, and others. He also served IEEE Transactions on Computers as an editor. Dr. Fujiwara is a fellow of the IEEE, a Golden Core member of the IEEE Computer Society, and a member of the Information Processing Society of Japan.



Dong Xiang received the BS degree and the MS degree in Computer Science from Chongqing University in 1987 and 1990, respectively. He received the PhD degree in Computer Engineering from the Institute of Computing Technology, the Chinese Academy of Sciences in 1993. He visited Concordia University, Montreal, Canada, as a postdoctoral researcher from 1994 to 1995 and the University of Illinois, Urbana Champaign from 1995 to 1996.

He was with the Institute of Microelectronics from Oct., 1996 to March 2003 as an Associate Professor. He is with the School of Software, Tsinghua University. He is now on leave and visiting Nara Institute of Science and Technology as a JSPS invitation fellow. His research interests include design and test of digital systems, including design for testability, testing, testability analysis, and BIST, fault-tolerant computing, distributed computing, and computer networking. He authored *Digital System Testing and Design for Testability* (Science Press, 1997). He is a member of the IEEE and IEEE Computer Society.



Shan Gu received the BS degree and the MS degree in Microelectronics, Tsinghua University in 2000 and 2003, respectively. Her research interests include design for testability and testing.