Nonscan Design for Testability for Synchronous Sequential Circuits Based on Conflict Resolution

Dong Xiang, *Member*, *IEEE*, Yi Xu, and Hideo Fujiwara, *Fellow*, *IEEE*

Abstract—A testability measure called *conflict* based on conflict analysis in the process of sequential circuit test generation is introduced to guide nonscan design for testability. The testability measure indicates the number of potential conflicts to occur or the number of clock cycles required to detect a fault. A new testability structure is proposed to insert control points by switching the extra inputs to primary inputs, using whichever extra inputs of all control points can be controlled by independent signals. The proposed design for testability approach is economical in delay, area, and pin overheads. The nonscan design for testability method based on the *conflict* measure can reduce many potential backtracks and make many hard-to-detect faults easy-to-detect; therefore, it can enhance actual testability of the circuit greatly. Extensive experimental results are presented to demonstrate the effectiveness of the method.

Index Terms—Conflict, inversion parity, nonscan design for testability, partial scan design, sequential depth for testability, testability measure.

1 Introduction

Design for testability for sequential circuits is very essential. Full scan design arranges all flip-flops in a chain when the circuit is being tested and values of state lines are scanned in before each test and scanned out after each test, which reduces the test generation problem to the combinational circuit test generation problem. An attractive alternative to full scan design is partial scan design, in which only a subset of the flip-flops is placed in a scan chain. Delay, area overheads, and test application time can be reduced. Scan design has the following disadvantages: 1) Test vectors cannot be applied at the speed of the operational clock and test application time is higher than that in a nonscan designed circuit due to shifting tests and test responses through scan chains; 2) scan design can only insert control and observation points into state lines and outputs of flip-flops. Greater testability improvement can be obtained when test points are inserted into other internal lines.

1.1 Previous Work

Nonscan design can provide at-speed test, low test application cost, and, above all, effectively enhance testability. Test point insertion has been extensively used in various issues of design for testability. Hayes and Friedman [13] and Saluja and Reddy [27] proposed insertion of test points in a combinational circuit as a means to make the circuit fully testable by a test set of small cardinality. Fujiwara et al. [8] and Pradhan [23] proposed the use of extra inputs to simplify testing by augmenting a machine so that it contains the synchronizing sequence and the distinguishing sequence, through which an easily testable sequential machine can be designed. Saluja and Dandapani [26] presented a method of modifying a multiple-output sequential machine by adding extra inputs. A machine so modified can be tested by a checking experiment and determined to be faulty or fault-free by observing one output value only. Motohara and Fujiwara [17] utilized a couple of effective heuristics to place test points for combinational circuits based on testability analysis and the FAN test generation algorithm [9] in order to obtain complete test efficiency. Rudnick et al. proposed a hard-fault-oriented observation point insertion method to enhance testability and provide at-speed test by combining an aliasing minimization technique [24]. Recently, papers [6], [19] presented techniques to achieve complete test efficiency by modifying the state transition table of a sequential machine, which can make all hard-to-reach states easily reachable.

Recent literature [4], [7], [11], [16], [21], [25], [29], [31], [32] tend to place test points based on testability analyzers. Rudnick et al. [25] presented a greedy procedure to load flip-flops at data inputs of flip-flops and place observation points at any internal nodes using SCOAP [12]. Control points at data inputs of flip-flops can make the loaded flip-flops combinational elements like scan design. Tamarapalli and Rajski [29] presented a multiphase test point insertion method in a scan-based environment using a probabilistic estimation of testability gain of the remaining hard fault set of the previous phase, which can obtain complete or near-complete fault coverage for built-in self test (BIST). Cheng and Lin [4] proposed a timing-driven test point placement method based on the COP measure and the gradient approach.

- D. Xiang and Y. Xu are with the School of Software, Tsinghua University, Beijing 100084, P.R. China. E-mail: dxiang@tsinghua.edu.cn.
- H. Fujiwara is with the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma, Nara 630-0101, Japan.

Manuscript received 16 May 2001; revised 14 Nov. 2001; accepted 28 July 2002.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 114161.

0018-9340/03/$17.00 © 2003 IEEE Published by the IEEE Computer Society

Most of the above testability-analysis-based test point insertion methods used the classic testability measures just like SCOAP [12] or COP. The measures did not cope with the influences of reconvergent fanouts. However, reconvergent fanouts have great effects on testability. Additionally, the COP measure is unable to handle general sequential circuits. It is essential to present a good testability measure which can reflect the actual testability of a sequential circuit in the process of test generation. Dey and Potkonjak [7] introduced a new testability measure called k-level controllability/observability to break cycles of the EXU s-graph for RTL circuits. Test multiplexers were inserted to avoid equal weight reconvergent fanouts. Ghosh et al. [11] proposed a nonscan design for testability of RTL circuits using a testability measure independent of data path widths. Complete or near complete test efficiency was obtained for almost all circuits with very low area, delay, and power overheads. Parikh and Abramovici [20], [21] proposed a new testability measure to guide partial scan design and partial reset successfully, which represents the number of clock cycles required to activate, propagate, and detect a fault. However, all testability measures in [7], [11], [16], [20], [21] still did not consider influences of reconvergent fanouts well. Xiang and Patel [30] introduced a testability measure called TIP (testability improvement potential) based on valid circuit state information via logic simulation, which effectively includes influences of reconvergent fanouts into circuit states. Partial scan design based on the testability measure can reduce as many as possible backtracks and make many hard-to-detect faults easy-to-detect. However, the TIP measure can only handle partial scan design.

1.2 Objectives of This Paper

We shall propose a conflict-analysis-based measure for synchronous sequential circuits. A conflict is caused by reconvergent fanouts with nonuniform inversion parity and the same sequential depth for testability paths. A couple of techniques are utilized to estimate the testability measure in order to emulate the actual testability of a sequential circuit during test generation: Inversion parity and *sequential depth for testability* are used to analyze potential conflicts during test generation of a synchronous sequential circuit. Potential conflicts between fault effect activation and fault effect propagation signal assignments are checked intensively because fault effect activation and fault effect propagation are closely related [28]. Stem segment partitioning is utilized to simplify observability analyses. Test points according to the conflict-analysis-based measures are placed in order to resolve as many as possible potential conflicts or make as many as possible hard-to-detect faults easy-to-detect.

In the rest of this paper, definitions and notation of the paper are presented in Section 2. Procedures to calculate inversion parity and sequential depth for testability are introduced in Section 3. A new testability measure, called *conflict*, is presented in Section 4. The test point selection procedure is proposed in Section 5. A new test point structure is also introduced in Section 5, which makes the design for testability method economical in area, delay, and pin overheads. Experimental results are presented in Section 6. The paper is concluded in Section 7.

2 Definitions and Notation

We introduce some definitions and notation of the paper first. A *signal requirement* is a 2-tuple (A, v), which means a node A is required to be assigned a value v, where $v \in \{1, 0, \times\}$. The *noncontrolling 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. The controlling and the noncontrolling values of an AND gate are 0 and 1, respectively. As for XOR gate, both 0 and 1 are noncontrolling values because both values on one of its inputs cannot determine the value of its output. Consider a 2-input multiplexer, its output can be determined when the control signals are assigned specified values and the selected input is assigned a determined value. Thereby, all inputs should be regarded as noncontrolling because they do not dictate the output. Only pairs of inputs can dictate the output.

Definition 1. *A conflict is defined as follows: A line l has been assigned value v, it also needs to be assigned value v' in the same clock cycle. 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.*

Usually, a conflict occurs at fanout stems. When all assignments are necessary, a conflict indicates the fault under consideration may quite possibly be redundant; otherwise, it can be resolved by backtracking. The main cause of conflicts is still reconvergent fanouts with nonuniform inversion parities. In the rest of this section, *the easiest way to justify a signal requirement* is determined by the *conflict* measure, which means the minimum controllability measure.

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

Inversion parity $inv_v(B, A)$ from node A to B 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),
2. The easiest way to justify (B, v) passes only a path of odd inversion parity from A to B,
3. The easiest way to justify (B, v) passes only a path of even inversion parity from A to B,
4. The easiest way to justify (B, v) 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 $seq_v(l, s)$ $(v \in \{0, 1\})$ from a fanout stem s to a line l is defined as the least number of clock cycles required to justify a signal requirement (l, v) at the line l to the fanout stem s.*

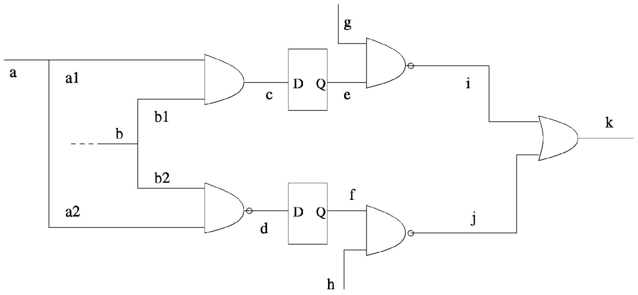


Fig. 1. Example for inversion parity and sequential depth for testability.

When $seq_v(l, s) = 0$, it indicates the easiest way to justify the signal requirement (l, v) has no signal requirement on the fanout stem s or the easiest way to justify that the signal requirement 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. As shown in Fig. 1, a, g, and h are primary inputs and b is not as easy as a primary input to control value 0. $inv_0(k, a) = 11$ because the easiest way to control value 0 on k includes paths a-c-e-i-k and a-d-f-j-k and inversion parities of them are odd and even, respectively. $inv_1(k, a) = 00$ because signal requirement $(k, 1)$ can be met by assigning value 0 on g or h. $inv_0(k, b) = 01$ because signal requirement $(k, 0)$ can be satisfied by assigning value 1 on b via b-c-e-i-k, whose inversion parity is odd. $inv_1(k, b) = 00$ because the easiest way to meet $(k, 0)$ has no signal requirement on b. $seq_0(k, a) = seq_0(k, b) = 1$ and $seq_1(k, a) = seq_1(k, b) = 0$.

i-controllability $C_l(i)$ of node l should reflect the potential number of conflicts (or possibility of causing conflicts) and the number of clock cycles required in order to justify a signal requirement (l, i), where $i \in \{\times, 0, 1\}$. The easiest fault effect propagation (*EFEP*) path of a fault is the easiest path to propagate the fault effect on the node to a primary output. In this case, the easiest path indicates the path with minimum observability. We define different observabilities for different fault effects D and $\overline{D}$. Lines outside of the EFEP path that feed the gates in the EFEP path are called sensitization lines. Assume observabilities of successors of a node have been calculated. The EFEP path of the node can be obtained as follows: If the node has only one successor, add the node into the EFEP path; otherwise, add the fanout branch with the least observability measure into the EFEP path. The above process should continue until a primary output is reached, which forms the EFEP path of the fault. Observability measures of the successors are available because observability is calculated backward from primary outputs to primary inputs. Therefore, the above discussion presents a complete procedure to calculate EFEP path for each fault. *v-Observability* $O_A(v)$ $(v \in \{D, \overline{D}\})$ reflects the number of conflicts (or possibility of causing conflicts) or the number of clock cycles required to propagate a fault effect v along the EFEP path. The EFEP path can be partitioned into stem segments, where a stem segment is the path segment between two fanout stems.

3 Calculations of Inversion Parity and Sequential Depth for Testability

Sequential depth for testability and *inversion parity* are calculated from fanout stems that reach the line under consideration. Calculation of inversion parity includes testability consideration. Assume "$\overline{\quad\quad}$" is the bitwise NOT operator.

Procedure 1 (inversion parity)

1. If line l is a fanout branch steming from s (or s'), where s' is a fanout stem succeeding to s,
$$inv_v(l, s) = \begin{cases} 10 & \text{if } l \text{ stems from } s; \\ inv_v(s', s) & \text{if } l \text{ stems from } s'. \end{cases}$$
2. If line l is the output of an inverter with input i, let $v \in \{0, 1\}$
$$inv_v(l, s) = \begin{cases} \overline{inv_v(i, s)} & \text{if } inv_v(i, s) = 10 \text{ or } 01; \\ inv_v(i, s) & \text{if } inv_v(i, s) = 00 \text{ or } 11. \end{cases}$$
3. If line l is the output of a D flip-flop with input i, for $v \in \{0, 1\}$
$$inv_v(l, s) = inv_v(i, s).$$
4. Let line l be the output of an AND or OR gate with inputs $i_1, i_2, \ldots, i_n$, where v_1 and v_2 are the value output when all inputs are assigned noncontrolling value and one of the input is assigned controlling value, respectively.
$$inv_{v_1}(l, s) = inv_{v_1}(i_1, s) \vee \ldots \vee inv_{v_1}(i_n, s),$$
where "$\vee$" is the bitwise OR operator of the binary numbers.
$$inv_{v_2}(l, s) = inv_{v_2}(i, s),$$
where i is the easiest input of gate l to be controlled to value v_2.
5. Let line l be the output of a NAND or NOR gate with inputs $i_1, i_2, \ldots, i_n$, $v_1, v_2 \in \{0, 1\}$ are defined as above, we have
$$tem = inv_{\overline{v_1}}(i_1, s) \vee \ldots \vee inv_{\overline{v_1}}(i_n, s),$$
$$inv_{v_1}(l, s) = \begin{cases} \overline{tem} & \text{if } tem = 01 \text{ or } 10; \\ tem & \text{if } tem = 00 \text{ or } 11. \end{cases}$$
$$inv_{v_2}(l, s) = inv_{\overline{v_2}}(i, s),$$
where i is the easiest input to be controlled to the controlling value.

Procedure 2 is utilized to calculate the *sequential depth for testability* from a fanout stem s for a line l which is a predecessor of l. We have $seq_v(l, s) = 0$ if l is unreachable from fanout stem s.

Procedure 2 (sequential depth for testability)

1. If line l is a fanout branch steming from s (or s'), where s' is a fanout stem succeeding to s,

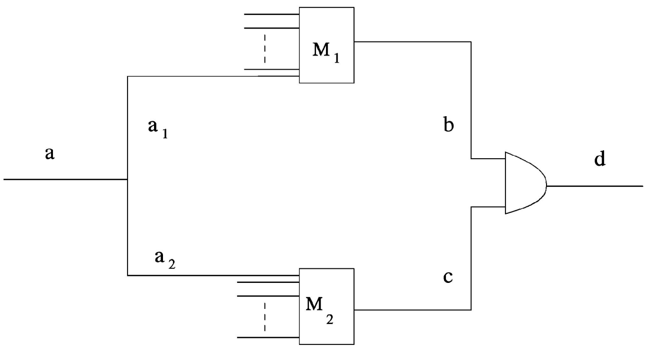


Fig. 2. Sequential depth for testability.

$$seq_v(l,s) = \begin{cases} 0 & \text{if } l \text{ stems from } s; \\ seq_v(l,s') & \text{if } l \text{ stems from } s^{'}. \end{cases}$$

2. If line l is the output of an inverter with input i,

$$seq_v(l,s) = seq_{\overline{v}}(i,s).$$

3. If line l is the output of a D flip-flop with input i,

$$seq_v(l,s) = seq_v(i,s) + 1.$$

4. Let line l be the output of an AND, OR, NAND, or NOR gate with inputs $i_1, i_2, \ldots, i_n$, v_1 and v_2 be the values of l when all inputs are assigned noncontrolling values v_3 and one of the inputs is assigned the controlling value v_4. Input i is the easiest input to be controlled as the controlling value v_4.

$$seq_{v_1}(l,s) = max(seq_{v_3}(i_1,s), \ldots, seq_{v_3}(i_n,s)),$$
$$seq_{v_2}(l,s) = seq_{v_4}(i,s).$$

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. When a cycle is met, iterative calculation of the sequential depth for testability may be necessary. Assume $i_1, i_2, \ldots, i_n$ are inputs of an AND gate with output l. Let i be the easiest input to be justified to the controlling value,

$$seq_1(l,s) = max(seq_1(i_1,s), \ldots, seq_1(i_n,s)),$$
$$seq_0(l,s) = seq_0(i,s).$$

Lemma 1. *Assume the line d as shown in Fig. 2 can be assigned value 1 without any conflict, where the signal requirements $(b,1)$ and $(c,1)$ must set specified values on a_1 and a_2. Let the corresponding numbers of clock cycles required to justify both $(b,1)$ and $(c,1)$ to a be M_1 and M_2. The number of clock cycles T_1 required to assign value 1 on d is no more than,*

$$T = max(M_1, M_2). \qquad (1)$$

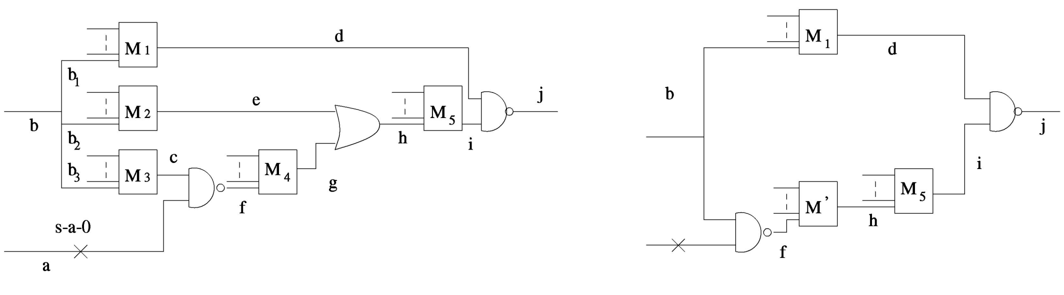
Proof. Let justifications of signal requirements $(b,1)$ and $(c,1)$ assign some specified values on the fanout stem s. The boxes in Fig. 2 can be two sequential machines. Therefore, $T = max(M_1, M_2)$ clock cycles is required in order to set the line d as value 1. The lines b and/or c can be set as the noncontrolling value via other primary inputs if one of the signal requirements $(b,1)$ and $(c,1)$ can be met via other primary inputs, which causes no signal requirement on a. The signal requirement $(d,1)$ can be justified through a with less than T clock cycles in this case. □

The *sequential depth for testability* starting from primary inputs is similar to the measure in [20], [21], which has been used to guide partial scan design and partial reset successfully. The sequential 1-controllability measure is $M_1 + M_2$ for the example in Fig. 2 according to SCOAP [12]. It is shown that the *sequential depth for testability* represents more actual testability than the sequential controllability of SCOAP.

Lemma 2. *Assume the fault effect of the line a can be propagated to j successfully, as shown in Fig. 3a. Let justifications of signal requirements $(d,1)$, $(e,0)$, and $(c,1)$ need M_1, M_2, and M_3 clock cycles, respectively. M_4 and M_5 clock cycles are required in order to propagate the fault effect from f to g and from h to i, respectively. The number of clock cycles to propagate the fault effect from a to j is no more than,*

$$T = max(max(M_3 + M_4, M_2) + M_5, M_1). \qquad (2)$$

Proof. The worst case is that there exists no easy-to-control node in the paths b-d, b-e, and b-c. That is, the sensitization values on c, e, and d should have some signal requirement on b. The number of clock cycles T_1



(a)

(b)

Fig. 3. Fault effect propagation with sequential depth for testability.

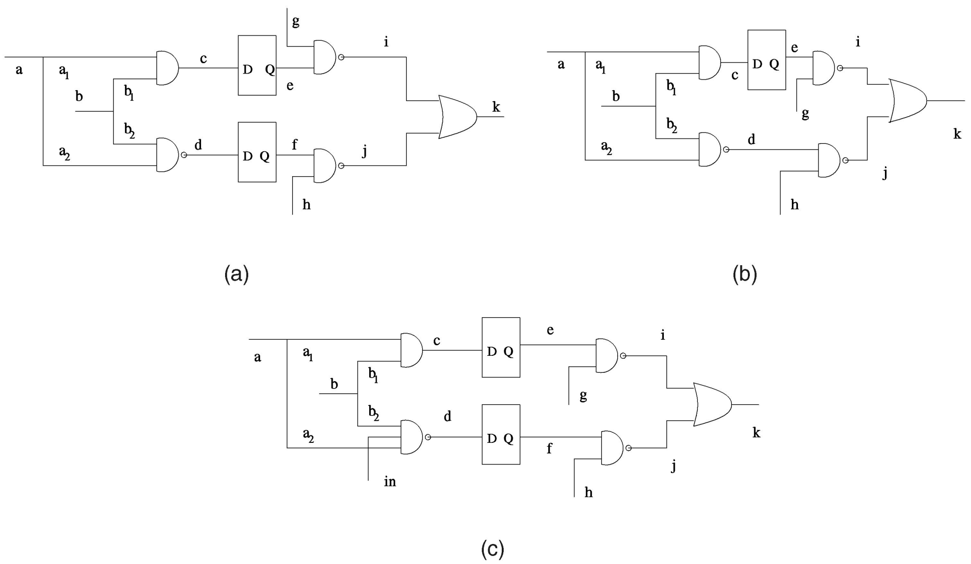


Fig. 4. Conflict analysis by signal requirement justification.

required to propagate the fault effect on the line a to h is thus,

$$T_1 = max(M_3 + M_4, M_2). \quad (3)$$

Up to now, the subcircuits included in M_2, M_3, the gate f and M_4 can be reduced to a new machine M', as shown in Fig. 3b, through which the propagation of the fault effect needs no more than T_1 clock cycles. It does not indicate that the logic in Fig. 3a is the same as that in Fig. 3b. The reduction is only utilized to estimate the number of clock cycles T required to propagate the effect on the line a to the line j. Therefore, the number of clock cycles required to propagate the fault effect from a to j is no more than $T = max(T_1 + M_5, M_1)$. □

The number of clock cycles required to propagate a fault effect from a to j in Fig. 3a should be $M_1 + M_2 + M_3 + M_4 + M_5$ according to SCOAP [12]. It is clear that SCOAP presented too pessimistic an estimation.

4 The Conflict-Analysis-Based Measure *conflict*

4.1 Controllability of the conflict Measure

The *conflict* measure penalizes controllability at the reconvergent points of fanouts with nonuniform inversion parity and equal sequential depth for testability. In order to get a more accurate analysis, we need to calculate inversion parity from a fanout stem s to lines which are reachable from s before its final reconvergent point as introduced in the above section.

Sequential depth for testability of different paths corresponding to a reconvergent fanout should be considered. A reconvergent fanout causes no potential conflict if sequential depths for testability of the reconvergent paths are unequal although inversion parities of both paths are unequal. Let us consider the example in Fig. 4a. We have $seq_0(i, a) = seq_0(j, a) = 1$. The inversion parities of the above two paths are nonuniform, that is, $inv_0(i, a) = 01$ while $inv_0(j, a) = 10$. The signal requirement $(k, 0)$ at node k needs to assign both i and j as value 0. A conflict should occur at a or b. Therefore, 0-controllability $C_k(0)$ of line k should be penalized. Let us consider the circuit presented in Fig. 4b. In this case, $seq_0(i, a) = 1$ and $seq_0(j, a) = 0$. Hence, i and j can be justified by setting a in different clock cycles. Line k can be assigned value 0 without any conflict.

When there is an easy-to-control node in one of the reconvergent fanout paths, that path seems to be cut. The signal requirement of the gate will not cause any conflict at the fanout stem. As for the circuit presented in Fig. 4c, there is an easy-to-control input *in* feeding the gate d. It looks like the path a-d-f-j is being cut. The signal requirement $(k, 0)$ at line k can be justified without any conflict. However, we do not need to check whether there exists one or more easy-to-control node in a path, which has been included in inversion parity and sequential depth for testability.

We would like to use the circuit as shown in Fig. 5 to illustrate how inversion parity and *sequential depth for testability* have great effects on controllability. We would like to show there still exists no conflict even though inversion parities of two reconvergent fanout branches are different if the sequential depths for testability of them are different. Let us consider activation of the fault $15/0$. Lines 14 and 5 must be assigned value 0 in order to activate the fault. The easier way to set 14 as value 0 is to set 13 as value 0. The easier way to set 10 as value 0 is to set value 0 on line 8. It is necessary to set value 0 on line 17 in order to set value 0 on line 5, to meet which line 8 must be assigned value 1. It seems a conflict on line 8 occurs because $inv_0(14, 8) \neq inv_0(5, 8)$. Actually, there is no conflict on fanout stem 8 because $seq_0(5, 8) \neq seq_0(14, 8)$. We can easily set value 1 on line 15 in the following way: Set value 0 on primary input 1 in the first clock cycle; set value 1 on primary input 1 and value 0 on primary input 2 in the second clock cycle. The fault $15/0$ can be activated successfully after two clock cycles.

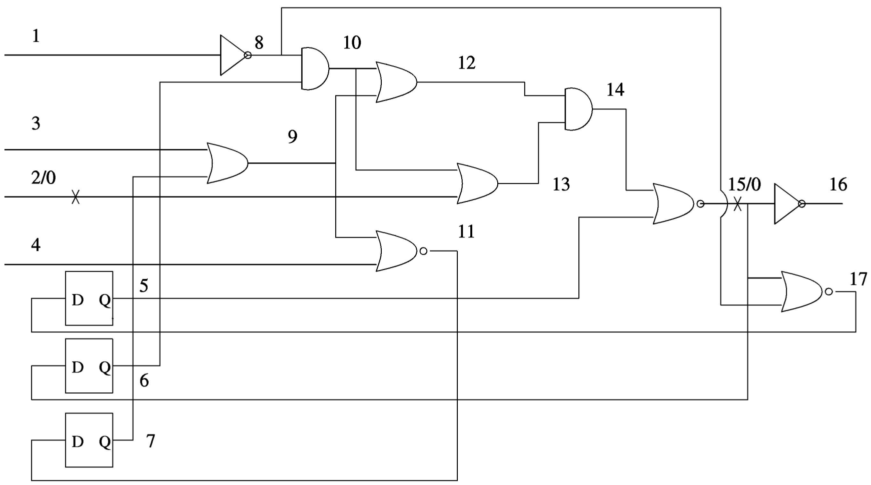


Fig. 5. Different delays cause no conflict.

We can calculate controllability measures of the *conflict* measure as follows: Consider a 2-input AND gate with inputs A, B, and an output y,

$$C_y(0) = min(C_A(0), C_B(0)), \tag{4}$$

$$C_y(1) = C_A(1) + C_B(1) + p, \tag{5}$$

where $p = 10 \cdot n$, n is the number of fanouts s with $inv_1(A, s) \neq inv_1(B, s)$, and none of them is 00, also $seq_1(A, s) = seq_1(B, s)$. Let y be the output of an OR gate with inputs A and B, we have

$$C_y(0) = C_A(0) + C_B(0) + p, \tag{6}$$

$$C_y(1) = min(C_A(1), C_B(1)), \tag{7}$$

where p in (6)-(9) can be obtained like that of an AND gate. Let y be the output of an exclusive-or gate with inputs A and B,

$$C_y(0) = min(C_A(0) + C_B(0) + p, C_A(1) + C_B(1) + p), \tag{8}$$

$$C_y(1) = min(C_A(1) + C_B(0) + p, C_A(0) + C_B(1) + p). \tag{9}$$

If A and B should be assigned v_1 and v_2, p in (8) and (9) can be determined as follows: $p = 10 \cdot n$, n is the number of fanouts s with $inv_{v_1}(A, s) \neq inv_{v_2}(B, s)$, and none of them is 00, also $seq_{v_1}(A, s) = seq_{v_2}(B, s)$. Let i be the input of an inverter with output y,

$$C_y(v) = C_i(\overline{v}), \tag{10}$$

where $\overline{1} = 0$, $\overline{0} = 1$, and $v \in \{0, 1\}$. Consider a D flip-flop with an input i and an output y, our method gives an additional penalty set as the same as a conflict for the output controllability measures,

$$C_y(v) = C_i(v) + 10 \quad (v \in \{0, 1\}). \tag{11}$$

Calculations of other types of gates are similar. Only the typical gates, such as AND, OR, NOT, NAND, and NOR, are considered in this paper. Other gates or functional units can be extended easily. The gate-based multiplexers can be dealt with like other gates. It should be noted that *conflict* penalizes the controllability measure of the value that needs to assign all of its the input as noncontrolling value. When a sequential loop is met, iterative calculation should be necessary like SCOAP [12].

4.2 Observability of the conflict Measure

Observabilities are calculated assuming a fault effect is propagated along the easiest fault effect propagation path. We shall still use inversion parity to calculate observability. *conflict* considers interdependences among signal requirements on the sensitization lines between two fanout stems along the EFEP path. Inversion parity and sequential depth for testability are two important factors of potential conflicts.

Consider the circuit shown in Fig. 6a; there exists no conflict at b in order to propagate the fault effect of the fault $a/0$ because the sequential depths for testability of the paths $b - b_2$, $b - c$, and $b_2 - d$ are 0, 1, and 0, respectively. If we want to propagate the fault effect of the fault $d/0$ in Fig. 6b to h, c_2 and f should be assigned value 1 and 0, respectively. The sequential depths for testability of the paths $c - e$, $e - h$, and $c - f$ are 0, 1, and 1, respectively. Therefore, a conflict should occur at c in order to propagate the fault effect from d to h. When there exists an easy-to-control node in the path from one of the sensitization lines to a fanout stem, a conflict can be avoided. As shown in Fig. 6c, e and f should be assigned 1 and 0, respectively, in order to propagate the fault effect from a to h. The signal requirement $(f, 0)$ can be met by controlling line c as value 0. Therefore, there should be no conflict at b when propagating the fault effect from node a to node h. The circuit shown in Fig. 6d is another conflict example. The sequential depths for testability of the paths $b - e$, $e - g$, and $b - d - f$ are 1, 0, and 1, respectively. $inv_1(e, b) \neq inv_0(f, b)$ and a conflict must occur at b when propagating the fault effect from a to h.

Let us consider fault effect propagation of the fault $2/0$ along the EFEP path 2-13-14-15-16 in the circuit, as shown in Fig. 5, again. Line 10 must be assigned value 0 in order to propagate the fault effect from node 2 to 13. The easier way to set value 0 on line 10 is to set value 0 on line 8. Line 12

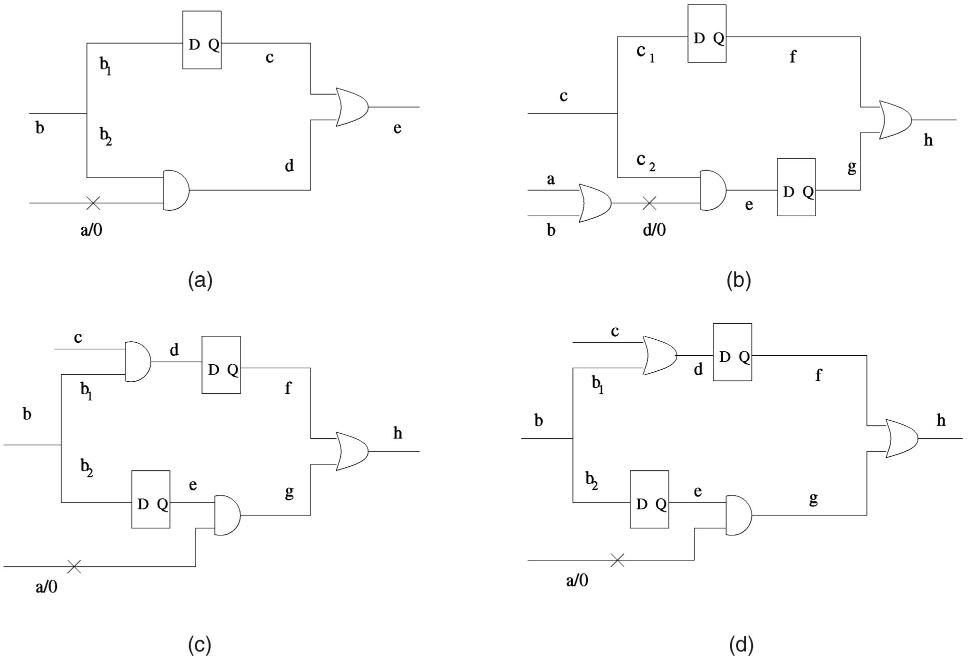


Fig. 6. Conflict analysis for fault effect propagation.

must be assigned value 1 in order to propagate the fault effect from line 13 to 14, which can be met by assigning value 1 on primary input 3. Line 5 must be controlled to value 0 in order to propagate the fault effect from line 14 to 15, which can be satisfied by assigning value 1 to the fanout stem 8. Line 8 must be controlled to value 0 and value 1 in order to propagate the fault effect of the fault $2/0$ to the primary output. It seems there should be a conflict at line 8 because $inv_0(10,8) \neq inv_0(5,8)$. Actually, there is no conflict on line 8 because $seq_0(10,8) \neq seq_0(13,5)$. We can propagate the fault effect of the single stuck-at fault $2/0$ to the primary output by using the following scheme: Primary input 1 is set as value 0 at the first clock cycle; primary inputs 1 and 3 are both controlled to value 1 at the second clock cycle. Therefore, the fault effect of the fault $2/0$ can be propagated to the primary output successfully without any conflict.

We must check the potential conflicts between the fault effect activation signal requirements and the fault effect propagation signal requirements. According to the conventional testability measures, fault effect activation and fault effect propagation are considered as two separate events. Savir pointed out good controllability and good observability do not always guarantee good testability [28] using previous measures. However, the fault effect activation problem and the fault effect propagation problem are closely related. Observability is calculated in *conflict* considering the fault effect is propagated along the EFEP path. Conflicts between the signal requirements of fault activation and signal requirements of sensitization lines should also be included. As shown in Fig. 7a, lines a and b should be assigned 1 in order to activate the single stuck-at fault $c/0$. Lines d, f, and h should be assigned 1, 0, and 1, respectively, in order to propagate the fault effect from c to i. Concurrent justification of the signal requirement $(a,1)$ and one or more of the signal requirements $(d,1)$, $(f,0)$, and $(h,1)$ may cause conflicts at a fanout stem s. It should be noted that observability estimation based on the above scheme does not include potential conflicts between signal requirements $(a,1)$ and $(b,1)$. Controllability estimation as stated in the above subsection only considers potential conflicts between signal requirements $(a,1)$ and $(b,1)$. When $inv_1(a,s) \neq inv_1(h,s)$, or $inv_1(a,s) \neq inv_1(f,s)$, or $inv_1(a,s) \neq inv_0(d,s)$, and sequential depths for testability of the corresponding paths are equal, a conflict occurs. We consider potential conflicts between fault effect activation and the sensitization signal requirements corresponding to the first stem segment in the EFEP path.

The observability measure of the *conflict* measure is calculated as follows: Let l be a primary output of the circuit, $O_l(v) = 0$, $v \in \{D, \overline{D}\}$. Consider the fault effect is propagated along the EFEP path. Potential conflicts are checked between two neighboring fanouts in the EFEP path. It should be noted that the EFEP path with respect to *conflict* is available because observability of *conflict* is calculated from primary outputs to inputs step by step. As shown in Fig. 7b, consider the fault effect is propagated along s_1-d-e-f-s_2, justification of signal requirements $(a,1)$, $(b,0)$, and $(c,1)$ may cause a conflict at a fanout stem s. We can get the number of potential conflicts of fault effect propagation as follows: First, we check whether the signal requirement $(a,1)$ causes conflicts with any signal requirements $(b,1)$ and $(c,1)$ according to inversion parities of the sensitization lines. If so, the observability measure is penalized. We then check whether justifications of the signal requirements $(d,1)$, $(f,0)$, and $(h,1)$ cause conflicts. The above comparisons are not so complex as they only utilize the calculated inversion parities and sequential depths for testability.

$$O_{s_1}(v) = O_{s_2}(v) + C_a(1) + C_b(0) + C_c(1) + p, \qquad (12)$$

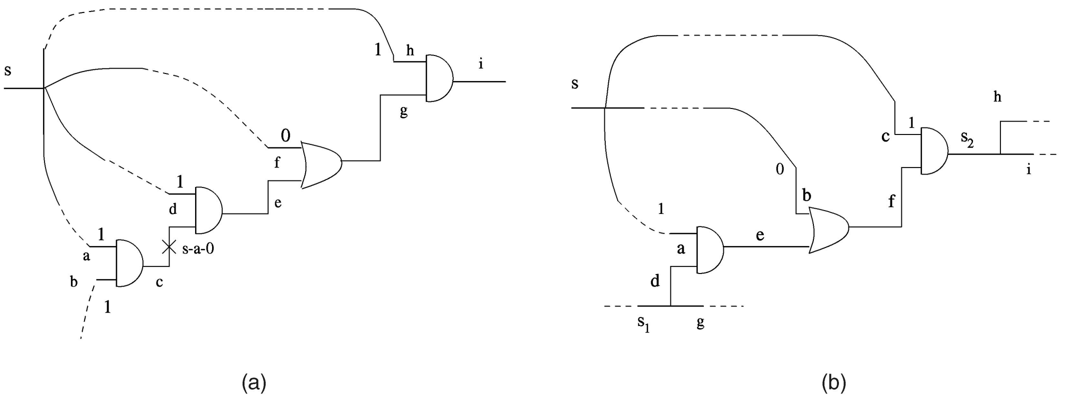


Fig. 7. Conflict analysis for observability: (a) conflicts between fault effect activation and propagation, (b) stem segment partitioning for observability.

where $p = n \cdot 10$, n is the number of potential conflicts when propagating the fault effect from s_1 to s_2, $\triangle p = n_1 \cdot 10$, and n_1 is the number of potential conflicts between b and c, and conflicts between a and one of b and c. Let l be the output of a D flip-flop with an input i, the same penalty as that in (11) is utilized to calculate observability measure of the data input,

$$O_i(v) = O_l(v) + 10. \qquad (13)$$

Our method still set observabilities of primary outputs as 0 like [12]. It should be noted that calculation of the *conflict* measure can be finished in $O(F \cdot N)$ time (F and N represent the number of lines and the number of fanouts, respectively).

5 Test Point Insertion

The proposed nonscan design for testability method utilizes only the *conflict* measure, which is independent of any test pattern generator and fault simulator. Test point insertion based on the *conflict* measure tries to reduce as many as possible potential conflicts in the process of test generation, which can make many hard-to-detect faults easily testable. Therefore, we can say the proposed test point insertion scheme can effectively enhance fault coverage.

5.1 Test Point Selection

Three separate classes of test points: 1-control (an OR gate with an extra input), 0-control (an AND gate with an extra input), and observation points are inserted into the circuit based on the *conflict* measure. Conflicts can be avoided by inserting test points. Test points are selected based on the *conflict* measure and the following testability gain function:

$$TG = \sum_{l/i \in F} (\triangle C_l(\bar{i}) + \triangle O_l(v)), \qquad (14)$$

where $\bar{i} = 0$ if $i = 1$, $\bar{i} = 1$ if $i = 0$; $v = \bar{D}$ if $i = 1$, $v = D$ if $i = 0$. $\triangle C_l(\bar{i})$ and $\triangle O_l(v)$ represent reduction of $\bar{i}$-controllability and v-observability, respectively. Potential conflict reduction between fault effect activation and fault effect propagation has been included in $\triangle O_l(v)$.

Procedure 3 (Test Point Selection)

1. Calculate the *conflict* measure of the circuit as stated in Section 3;
2. Choose the lines with the hard faults and their immediate successors and predecessors as test point candidates (TPC) based on *conflict*;
3. Use the selective tracing scheme to calculate testability gains when inserting three separate classes of test points into all nodes in the TPC set;
4. Select the best place and the best type of test point according to the results obtained in Step 3; insert the corresponding test point into the selected node; update testability of the circuit using the selective tracing scheme;
5. If all test points have been inserted, end the procedure. Otherwise, update TPC set, go to Step 3.

The selective tracing scheme [30], [31], [32] adopted in Steps 3 and 4 can be illustrated as follows: When controllability of a line changes, controllability of the immediate successor(s) of the line should be updated. When observability of a line changes, observability of the immediate predecessor(s) of the line should be updated. When controllability of an input of a gate changes, observability for other inputs of the gate should be updated.

5.2 New Test Point Structure

Test multiplexers are not inserted into the circuit directly, unlike [7], [11], [25], which are connected with the control input of the conventional test point. Fig. 8a presents the original circuit. One input of the multiplexer is connected with a PI, another input of the multiplexer is connected with a constant (1 for 0-control test point, 0 for 1-control test point), as shown in Fig. 8b. The extra inputs of test points at nodes A, B, and E are connected with constants 0, 1, and 0, respectively, when $ntest = 1$ (normal operation). The reason why test points are not inserted into the circuit directly like the previous methods [7], [11], [25] is that signals of the subcircuit connected with the test points in the original circuit cannot be blocked during ATPG and testing. The control input of all test multiplexers can also be thought of as a regular PI, which may cause a lot of conflicts during ATPG at that line using the previous methods because all test multiplexers are controlled by the same test input.

Dey and Potkonjak [7] proposed a nonscan design for testability based on k-level controllability/observability for RTL circuits, in which a scheme avoids generating equal weight reconvergent fanout regions when connecting extra

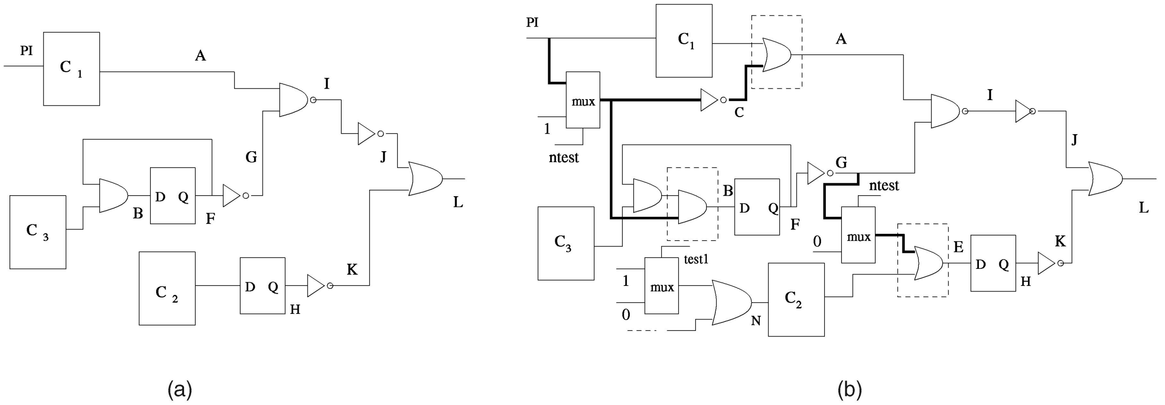


Fig. 8. DFT circuit with a single extra control input: (a) the original circuit, (b) mux-based DFT by switching the extra inputs to PIs.

inputs of test multiplexers with the same PI port. It is more possible for different control points to share the same PI for gate-level circuits. We have shown that conflicts can still be avoided even though a reconvergent fanout is an equal weight one in Section 3. Let a test point be inserted into node l. The extra control input i of a control test point is connected with a primary input in order to avoid conflicts. Generally, two different types of conflicts should be avoided: 1) conflicts at the primary input connected with the extra inputs of the control point generated by justifying signal requirement of the node l; 2) conflicts at the primary input connected with the extra inputs of the control point generated by justifying signal requirement of the reconvergent nodes succeeding to the PI and l. In our method, the control input of a test point is connected with a PI, which generates no reconvergent fanouts if possible. Otherwise, our method tries to connect control inputs of test points with PIs, which generates reconvergent fanouts of unequal sequential depth for testability if possible. Finally, our method connects extra inputs of test points with PIs, which generates reconvergent fanouts of equal sequential depth for testability and uniform inversion parity.

When the number of control points is greater than the number of PIs, more than one control point can be connected with the same PI. As shown in Fig. 8, two control points are inserted into nodes A and B, respectively. Justification of signal requirement should not generate conflicts at PI because $seq_1(G, PI) \neq seq_1(A, PI)$. Detailed techniques can be found in [32].

Each PI can be shared by the test points inserted into A and B, which introduces a new reconvergent fanout. Signal requirement $(I, 0)$ does not cause any conflict at PI because $seq_1(G, PI) \neq seq_1(A, PI)$. The extra input of the 1-control point at node E is connected with node G, which generates a new reconvergent region at node G. However, signal requirement $(L, 0)$ does not cause any conflict at node F because $seq_0(J, F) \neq seq_0(K, F)$ although $inv_0(J, F) \neq inv_0(K, F)$.

Constants can also be inserted like the constant multiplexer inserted into node N, as shown in Fig. 8b if testability of the circuit is still not good enough after all test points are inserted. It is not a good way to multiplex a constant with an internal line in the circuit directly, which may make fault effects of faults preceding to the node unobservable. Two constants 0 and 1 can be multiplexed whose output is connected with an extra OR (or AND) gate. The other input of the extra OR gate (or AND) gate is connected with the predecessor line of N in the original circuit. The extra input of the constant inserted at node N can be 1 or 0 by controlling the control input test1 of the constant multiplexer. Judicious grouping of multiple constants inserted makes the same control input $test1$ controlling multiple constant multiplexers be shared by more than one constants with similar schemes stated earlier in this section. Fig. 9 presents a simplified version of the DFT circuit as shown in Fig. 8b. The extra input of the test point is connected with another extra gate, which replaces the multiplexer, as shown in Fig. 9. One input of the extra gate is connected with a PI, the other input is the test mode line *ntest*. The circuit is set as the test mode when $ntest = 0$, while it is set as the normal mode when $ntest = 0$. The control input *test1* can be connected with the extra input of a gate like that inserted into node N, as shown in Fig. 8b, which reduces to inserting extra control points into the circuit. Up to now, it is unnecessary to use any constants and test multiplexers in the DFT circuit. It should be noted that the DFT circuits shown in Figs. 8 and 9 need only one extra input to switch all extra inputs of test points into PIs. The DFT circuit presented in Fig. 9 is economical in delay, pin, and area overheads. The test circuit of Figs. 8 and 9 is presented in Fig. 10.

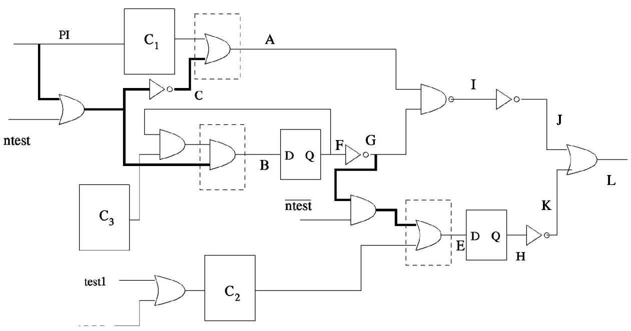


Fig. 9. Simplified DFT circuit: Test mode when ntest = 0 and normal mode when ntest = 1.

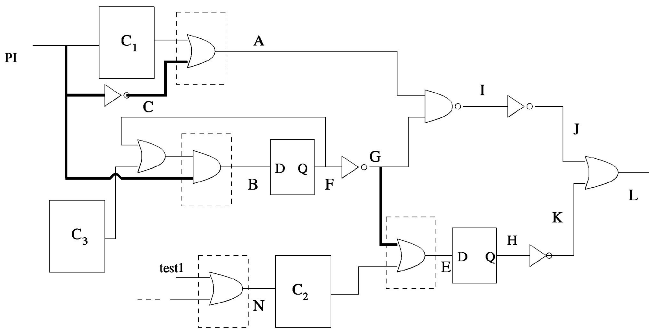


Fig. 10. Test circuit with conflict-analysis-based DFT.

Consider a signal requirement $(I, 0)$. Both A and G should be assigned value 1 in order to meet $(I, 0)$. $(A, 1)$ can be satisfied by $(PI, 0)$, while $(G, 1)$ can be met by assigning value 0 on PI. Therefore, the newly generated reconvergent fanouts cause no conflict in order to meet signal requirement $(I, 0)$. Consider another signal requirement $(L, 0)$. Lines J and K should be assigned 0. Line G should be assigned 0 in order to meet $(J, 0)$. Line G should be assigned 1 in order to meet $(K, 0)$. Line G can be assigned 1 and 0 in two clock cycles, respectively. No conflict occurs.

The exclusive-or chain scheme is adopted in all experiments of this paper. There may exist some aliasing when the number of observation points is large and a single exclusive-or chain is utilized [10], [24]. It is found that one or two exclusive-or chains are sufficient to avoid aliasing.

6 Experimental Results

A system called *nscan* has been completed to implement the nonscan design for testability method on E3000 server using C language. Table 1 shows the HITEC [18] ATPG results on the DFT method for almost all iscas89 and iscas93 circuits. In Table 1, ao, FC, TE, *cpu*, *vec*, *tp*, and *po* represent area overhead (percentage), fault coverage (percentage), test efficiency (percentage), ATPG time (seconds), the number of test vectors generated by HITEC, the number of test points, and the number of extra pins utilized, respectively.

The system *nscan* obtains 100 percent or near 100 percent test efficiency for almost all ISCAS and the synthesized circuits, except s38417, by inserting a reasonable number of test points. As for s344, s641, and s713, the system reaches fault coverages 98.3 percent, 99.4 percent, and 93.1 percent, respectively, and 100 percent test efficiency after inserting only one test point. *Nscan* gets 100 percent fault coverage for circuits s820, s1488, s967, s991, and s1512 after inserting 2, 3, 3, 3, and 12 test points, respectively.

The system *nscan* gets good fault coverage and test efficiency for hard-to-test circuits s526, s526n, s9234, s13207, s15850, s15850.1, s38417, s38584, and s38584.1, as shown in Table 1. HITEC gets 80.5 percent fault coverage and 82.7 percent test efficiency after 580 test points have been inserted into s38417. DFT results of the synthesized circuits am2910, div16, and mult16 are also given. The system *nscan* derives 93.1 percent (98.6 percent), 92.7 percent (98.5 percent), and 99.5 percent (100 percent) fault coverage (test efficiency) for circuits am2910, div16, and mult16 after 6, 35, and 30 test points are inserted, respectively.

Paper [25] presented results of quite a few circuits. Therefore, a system called *opus-ns* has been implemented according to the method proposed in Rudnick et al. [25]. As for *opus-ns*, the number of loaded flip-flops is equal to the number of PIs and other test points are observation points. It is shown that *nscan* gets better fault coverage and test efficiency than *opus-ns* except circuits s386, s510, and s4863. The system *nscan* obtains 96.7 percent, 97.9 percent, and 98.5 percent fault coverage for s386, s510, and s4863, while *opus-ns* gets 97.2 percent, 97.9 percent, and 99.3 percent fault coverage for the circuits. *nscan* and *opus-ns* obtain the same fault coverage and test efficiency for circuits s641, s820, and s832. The system *nscan* reaches much better fault coverage and test efficiency than *opus-ns* for circuits s526, s526n, s1423, s9234, s13207, s15850, s15850.1, s38417, s1512, s3330, and s3384, as shown in Table 1. The system *nscan* obtains better fault coverage and test efficiency than *opus-ns* for all remaining circuits.

Pin overhead for all DFT circuits in experiments of this paper is no more than 3 for all circuits with large enough size, which includes one extra control input for all test points and one or two extra outputs for outputs of the exclusive-or chains. No constant multiplexer is inserted in experiments of this paper. No node is switched to an easy-to-control node, as shown in Figs. 7, 8, and 9. However, only control test points in this approach contribute to delay overhead. The system *nscan* inserts few control test points for the largest circuits. Only one extra gate for each control test point is inserted into functional paths, as shown in Figs. 7, 8, and 9, which makes the method economical in delay overhead. Test points can be inserted away from the critical paths [4] if necessary.

It has been shown that *nscan* needs more test vectors for a number of circuits than *opus-ns*. The most important reason is that *nscan* gets more fault coverage than *opus-ns* for most of the circuits. Usually, HITEC needs a long test sequence to detect a hard-to-detect fault like other test generators.

We also compare *nscan* with two effective partial scan design tools opus [5] and CoPs [20]. Results in [20] were presented based on the GENTEST algorithm. CoPs is implemented and run on HITEC in this paper. The corresponding test generation results are shown in Table 2. It is shown that test generation results of CoPs after partial scan design are not completely compatible with those presented in [20] because different test generators are used. As shown in Table 2, *tap* is the number of test vectors for *nscan*. As for opus and CoPs, we have

$$tap = (sff + 1) \cdot vec + sff, \qquad (15)$$

where *sff* is the number of scan flip-flops and *vec* is the number of vectors generated by HITEC. As shown in Table 2, $a(b)$ stands for $a \cdot 10^b$. The nonscan design for testability method *nscan* gets no worse fault coverage for all circuits except s510, s526, s526n, s953, s1196, and mult16 than opus. *nscan* derives much better fault coverage than opus for circuits s991, s1269, s1512, and s3384. The system *nscan* gets worse fault coverage for circuits s386, s526, s526n, and s953 than CoPs. *nscan* obtains the same fault coverage as CoPs for circuits s641, s820, s832, s1488, s1494, s967, and s991. *nscan* reaches better fault coverage than CoPs for all

TABLE 1
Performance of *nscan* on the ISCAS and Synthesized Circuits

circuit	orig.		nscan					opus-ns				
	PI	FC	tp/po	ao	FC/TE	vec	cpu	tp/po	ao	FC/TE	vec	cpu
s298	3	86.0	3/3	2.0	98.8/100	551	9	3/3	2.0	95.3/99.4	449	50
s344	9	95.3	1/1	0.6	98.3/100	115	64	1/1	0.6	93.9/96.8	70	314
s349	9	95.4	2/2	1.1	98.6/100	218	64	2/2	1.1	98.0/99.7	112	392
s382	3	90.9	4/4	1.9	97.8/100	609	26.2	4/2	1.9	90.8/94.5	552	572
s386	7	81.8	3/3	1.7	96.7/100	308	1	3/3	1.7	97.2/100	278	6.5
s400	3	75.1	5/5	2.4	96.8/99.8	399	97.3	3/3	2.4	91.0/97.0	482	607
s444	3	78.7	5/5	2.3	95.7/99.6	326	62.8	3/3	2.3	91.3/95.4	590	551
s510	19	0.0	6/1	4.2	97.9/100	433	8.4	6/1	1.7	99.7/100	584	9
s526	3	9.2	5/5	1.9	94.3/96.8	8536	4738	6/2	1.9	77.6/85.6	890	2678
s526n	3	9.95	5/5	1.9	94.5/96.8	8022	4836	6/2	1.9	79.0/84.4	504	2192
s641	35	86.5	1/1	0.3	99.4/100	283	3.45	1/1	0.3	99.4/100	282	6.47
s713	35	81.9	1/1	0.3	93.1/100	276	5.82	1/1	0.3	93.0/100	258	8.6
s820	18	95.7	2/2	0.7	100/100	717	10.9	2/2	0.7	100/100	631	17.3
s832	18	93.9	2/2	1.0	98.4/100	745	19.6	2/2	1.0	98.4/100	716	20.5
s953	16	8.3	3/3	0.8	99.4/100	385	9.6	3/3	0.8	98.1/99.9	539	304
s1196	14	99.8	1/1	0.2	99.8/100	450	3.25	1/1	0.4	99.8/100	448	3.3
s1238	14	94.7	2/2	0.4	96.9/100	461	4.53	2/2	0.4	94.7/100	450	5.2
s1423	17	38.2	40/2	8.0	93.6/94.6	607	2132	40/2	5.0	83.5/84.9	161	5812
s1488	8	97.2	3/3	0.5	100/100	693	50	3/3	0.5	99.9/100	626	86
s1494	8	96.5	3/3	0.5	99.2/100	689	47	3/3	0.5	98.3/100	517	167
s5378	35	68.4	60/2	4.6	97.3/99.5	1337	6584	60/2	2.2	96.9/99.3	1187	8987
s9234	36	9.3	160/3	4.6	92.8/95.7	3685	8045	160/3	4.5	39.7/42.4	611	9424
s13207	62	8.9	240/3	4.8	91.8/94.9	3927	13488	240/3	3.7	52.7/58.8	1816	11101
s15850	77	6.6	240/3	4.5	94.2/97.6	8583	8441	240/3	2.1	77.9/82.3	2484	5978
s15850.1	77	38.0	240/3	4.5	94.0/97.5	5151	9934	240/3	2.1	79.5/83.8	2770	5528
s35932	35	89.2	200/3	1.7	90.9/100	318	1694	200/3	1.4	89.9/100	316	2847
s38417	28	3.57	580/3	3.9	80.5/82.7	1531	36.5h	580/3	4.0	9.6/10.9	597	68406
s38584	38	60.9	400/3	2.4	91.6/94.5	8908	59757	400/3	2.7	87.8/93.0	6321	76172
s38584.1	38	61.5	390/3	2.8	91.5/94.2	8297	58254	390/3	2.6	87.6/92.7	6503	79420
s967	16	7.1	3/3	0.8	100/100	415	17	3/3	0.8	98.4/99.7	637	107
s991	65	2.2	3/3	0.7	100/100	97	0.08	3/3	0.7	99.9/100	76	1.0
s1269	18	17.9	14/2	3.9	99.4/99.7	204	113	14/2	1.2	96.5/99.9	245	82
s1512	29	4.9	12/1	2.8	100/100	3224	6705	12/1	0.8	16.5/94.5	112	1704
s3271	26	98.7	9/2	1.6	99.6/99.6	688	4552	9/2	0.3	98.8/99.2	802	732
s3330	40	73.3	40/2	3.8	92.5/96.1	782	3144	40/1	1.0	81.9/84.4	571	10611
s3384	43	88.9	40/2	4.1	98.3/98.5	180	1355	40/1	0.9	89.6/89.7	236	8160
s4863	49	95.2	9/2	0.8	98.5/98.5	391	1805	9/1	0.2	99.3/100	371	504
s6669	83	99.0	9/2	0.6	99.9/99.9	327	2659	9/1	0.2	99.6/99.6	286	850
am2910	20	90.8	6/2	0.5	93.1/98.6	1610	1520	6/1	0.2	90.6/97.7	1016	2123
div16	33	78.3	35/2	0.7	92.7/98.5	287	102	35/2	2.1	89.2/96.2	241	2076
mult16	18	89.9	30/2	0.8	99.5/100	309	7.2	30/2	4.4	98.0/98.6	222	890

remaining circuits and much better fault coverage for circuits s13207, s38417, s1269, s3330, and s3384. The nonscan design for testability method *nscan* needs much fewer test cycles than opus and CoPs for almost all circuits.

7 Conclusions

A conflict-analysis-based testability measure *conflict* was proposed to guide nonscan design for testability. The system is called *nscan*. Reconvergent fanouts with nonuniform inversion parity is still one of the main causes of conflicts in the process of sequential circuit test generation. The testability measure implies the number of potential conflicts to occur when generating a test for a specific fault. A couple of schemes were adopted in the above measure to emulate the actual testability of a sequential circuit during test generation:

1. Inversion parity in sequential circuits was used to analyze potential conflicts.
2. Interdependence between fault effect activation and fault effect propagation signal assignments was checked intensively.
3. *Sequential depth for testability* was introduced to enhance testability of the circuit which calculates the *conflict* measure.
4. Different fault effects have different propagation conditions; we define different observabilities for them.
5. Stem segment partitioning is introduced to handle observability calculation.

A new test point structure is introduced to enhance testability of the circuits, which makes the method economical in pin, area, and delay overheads. Test points are inserted based on *conflict* in order to reduce as many as possible potential conflicts in the process of test generation and therefore make many hard-to-detect faults easy-to-detect and enhance fault coverage greatly. Extensive

TABLE 2
Comparison of *nscan* with the Previous Partial Scan Methods

circuit	nscan				opus				CoPs			
	tp/po	FC	tap	cpu	sff	FC	tap	cpu	sff	FC	tap	cpu
s298	3/3	98.8	551	9	2	98.1	722	0.6	2	98.4	1022	1.8
s344	3/3	99.7	146	0.5	3	96.5	375	13.8	3	96.8	363	31.4
s349	2/2	98.6	218	64	3	96.0	303	392	3	96.3	343	30.7
s382	4/4	97.8	609	26.2	4	95.5	1794	67	3	96.0	1055	51.1
s386	3/3	96.7	308	1	3	95.3	875	1.6	3	99.0	751	0.9
s400	5/5	96.8	399	97.3	4	94.4	1564	105	3	96.0	1771	37.5
s444	5/5	95.7	326	62.8	3	93.3	1579	50.8	3	94.7	2415	59.1
s510	6/1	97.9	433	8.4	5	100	893	0.03	5	100	1307	1.1
s526	5/5	94.3	8536	4738	3	95.9	16127	4400	4	94.4	17094	2947
s526n	5/5	94.5	8022	4836	3	95.8	17227	3677	4	94.6	11359	2693
s641	1/1	99.4	283	3.45	7	94.6	1197	0.68	4	99.4	924	1.6
s713	1/1	93.1	276	5.82	7	88.5	1087	2.5	5	92.9	1097	4.8
s820	2/2	100	717	10.9	4	100	1969	0.03	4	100	1649	1.8
s832	2/2	98.4	745	19.6	4	98.4	2204	0.2	4	98.4	1649	2.0
s953	3/3	99.4	385	9.6	3	100	1831	2.0	3	99.9	1055	1.8
s1196	1/1	99.8	450	3.25	1	100	887	0.08	1	99.8	935	3.4
s1238	2/2	96.9	461	4.53	2	94.9	1337	1.1	2	94.9	1316	5.2
s1423	40/2	93.6	607	2132	28	89.0	10497	3441	28	92.5	11193	2283
s1488	3/3	100	693	50	5	100	2075	0.8	5	100	2021	3.8
s1494	3/3	99.2	689	47	5	99.2	1853	0.7	5	99.2	1985	4.0
s5378	60/3	97.3	1337	6584	50	93.9	52019	1266	50	97.0	41768	5645
s9234	160/3	92.8	3685	8045	97	89.9	1.28(6)	9424	97	90.2	3.95(5)	8348
s13207	240/3	91.8	3927	13488	160	90.4	3.93(5)	8619	240	58.2	2.47(5)	8742
s15850	240/3	94.2	8583	8441	170	93.4	1.43(6)	4960	200	89.0	9.70(5)	22319
s15850.1	210/3	93.8	4205	8077	160	90.5	1.39(6)	13694	160	87.0	7.04(5)	21750
s35932	200/3	90.9	318	1694	306	89.8	151658	1381	306	89.8	75376	1779
s38417	650/3	82.9	2328	31.1h	400	79.7	438292	14887	500	60.5	1.92(6)	30758
s38584	450/3	92.8	12382	51838	N/A	N/A	N/A	N/A	300	88.0	3.95(6)	59016
s38584.1	450/3	92.9	13012	50777	N/A	N/A	N/A	N/A	300	88.0	3.95(6)	58957
s967	3/3	100	415	17	5	100	2477	0.1	5	100	1590	1.9
s991	3/3	100	97	0.08	7	3.2	87	0.4	7	100	599	0.7
s1269	12/2	97.6	188	2621	8	79.8	1890	1462	8	81.9	2060	1294
s1512	12/2	100	3375	700	12	82.6	23685	1704	8	95.4	81143	3245
s3271	9/2	99.6	688	4552	6	99.6	6194	170	6	99.4	5718	5.8
s3330	40/2	92.0	722	15538	30	89.0	15592	6366	30	82.8	11004	9910
s3384	40/2	98.3	180	1355	30	89.2	4680	8359	30	89.4	4153	8007
s4863	9/2	98.5	391	1805	9	97.0	4139	3263	9	97.6	3829	3292
s6669	9/2	99.9	327	2659	9	99.4	2689	850	9	99.5	2969	1191
am2910	6/2	93.1	1610	1520	22	91.1	11245	939	22	92.7	53957	662
div16	35/2	92.7	287	102	21	91.5	5499	1304	21	87.0	19601	2888
mult16	30/2	99.5	309	7.2	22	99.8	3748	18.5	22	97.1	3817	754

experimental results were presented to demonstrate the effectiveness of the method by comparing with the previous nonscan design for testability method opus-ns and two effective partial scan design tools opus and CoPs. Nscan obtains better fault coverage than opus-ns for almost all benchmark circuits, while it gets even better fault coverage than both partial scan design tools for almost all circuits.

Acknowledgments

Dong Xiang would like to express his thanks to Professor Janak Patel of the University of Illinois at Urbana Champaign for presentation of software packages HITEC and opus, synthesized circuits, and, above all, kindly encouragement. The authors would like to thank all of the anonymous reviewers for their constructive comments. This work was supported in part by the National Science Foundation of China under grant 69773030 and, in part, by the Semiconductor Technology Academic Research Center (STARC) of Japan. An earlier version of this paper was published in the *Proceedings of the IEEE International Test Conference*, pp. 520-529, 2000.

References

[1] M. Abramovici, M.A. Breuer, and A.D. Friedman, *Digital Systems Testing and Testable Design.* Computer Science Press, 1990.

[2] M.L. Bushnell and V.D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits.* Kluwer Academic, 2000.

[3] K.T. Cheng and V.D. Agrawal, "Partial Scan Method for Sequential Circuits with Feedback," *IEEE Trans. Computers,* vol. 39, pp. 544-548, 1990.

[4] K.T. Cheng and C.J. Lin, "Timing-Driven Test Point Insertion for Full Scan and Partial Scan BIST," *Proc. IEEE Int'l Test Conf.,* pp. 506-514, 1995.

[5] V. Chickermane and J.H. Patel, "An Optimization Based Approach to the Partial Scan Design Problem," *Proc. IEEE Int'l Test Conf.,* pp. 377-386, 1990.

[6] D.K. Das, S. Ohtake, and H. Fujiwara, "New DFT Techniques of Non-Scan Sequential Circuits with Complete Fault Efficiency," *Proc. Eighth IEEE Asian Test Symp.,* pp. 263-268, 1999.

[7] S. Dey and M. Potkonjak, "Non-Scan Design for Testability Techniques Using RTL Design Information," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 16, no. 12, pp. 1488-1506, 1997.

[8] H. Fujiwara, Y. Nagao, T. Sasao, and K. Kinoshita, "Easily Testable Sequential Machines with Extra Inputs," *IEEE Trans. Computers,* vol. 24, no. 8, pp. 821-826, Aug. 1975.

[9] H. Fujiwara, *Logic Testing and Design for Testability.* The MIT Press, 1985.

[10] H. Fujiwara and A. Yamamoto, "Parity Scan Design to Reduce the Cost of Test Application," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 12, no. 10, pp. 1604-1611, 1993.

[11] I. Ghosh, A. Raghunathan, and N.K. Jha, "Design for Hierarchical Testability of RTL Circuits Obtained by Behavioral Synthesis," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 16, no. 9, pp. 1001-1014, 1997.

[12] L.H. Goldstein, "Controllability/Observability Analysis of Digital Circuits," *IEEE Trans. Circuits and Systems,* vol. 26, pp. 685-693, 1979.

[13] J.P. Hayes and A.D. Friedman, "Test Point Placement to Simplify Fault Detection," *IEEE Trans. Computers,* vol. 23, no. 7, pp. 727-735, July 1974

[14] H.H.S. Gundlack and K.D. Müller-Glaser, "On Automatic Testpoint Insertion in Sequential Circuits," *Proc. IEEE Int'l Test Conf.,* pp. 1072-1079, 1990.

[15] C.C. Lin, M. Marek-Sadowska, K.T. Cheng, and M.T.C. Lee, "Test Point Insertion: Scan Path through Functional Logic," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 17, no. 9, pp. 838-851, 1998.

[16] C.J. Lin, Y. Zorian, and S. Bhawmik, "PSBIST: A Partial Scan Based Built-In Self Test Scheme," *Proc. IEEE Int'l Test Conf.,* pp. 507-516, 1993.

[17] A. Motohara and H. Fujiwara, "Design for Testability for Complete Test Coverage," *IEEE Design and Test of Computers,* vol. 1, no. 4, pp. 25-32, Nov. 1984.

[18] T. Niermann and J. Patel, "HITEC: A Test Generation Package for Sequential Circuits," *Proc. European Conf. Design Automation,* pp. 214-218, 1991.

[19] S. Ohtake, T. Masuzawa, and H. Fujiwara, "A Non-Scan DFT Method for Controllers to Achieve Complete Fault Efficiency," *Proc. Seventh IEEE Asian Test Symp.,* pp. 204-211, 1998.

[20] P.S. Parikh and M. Abramovici, "Testability-Based Partial Scan Analysis," *J. Electronic Testing: Theory and Applications,* vol. 7, pp. 61-70, 1995.

[21] P.S. Parikh and M. Abramovici, "On Combining Design for Testability Techniques," *Proc. IEEE Int'l Test Conf.,* pp. 423-429, 1995.

[22] I. Pomeranz and S.M. Reddy, "Design-for-Testability for Path Delay Faults in Large Combinational Circuits Using Test Points," *Proc. ACM/IEEE Design Automation Conf.,* pp. 358-364, 1994.

[23] D.K. Pradhan, "Sequential Network Design Using Extra Inputs for Fault Detection," *IEEE Trans. Computers,* vol. 32, no. 3, pp. 319-323, Mar. 1983.

[24] E.M. Rudnick, V. Chickermane, and J.H. Patel, "An Observability Enhancement Approach for Improved Testability and At-Speed Test," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 13, no. 8, pp. 1051-1056, 1994.

[25] E.M. Rudnick, V. Chickermane, P. Banerjee, and J.H. Patel, "Sequential Circuit Testability Enhancement Using a Non-Scan Approach," *IEEE Trans. VLSI Systems,* vol. 3, no. 2, pp. 333-338, 1995.

[26] K.K. Saluja and R. Dandapani, "An Alternative to Scan Design Methods for Sequential Machines," *IEEE Trans. Computers,* vol. 35, no. 4, pp. 384-388, 1986.

[27] K.K. Saluja and S.M. Reddy, "On Minimally Testable Logic Networks," *IEEE Trans. Computers,* vol. 23, no. 11, pp. 1204-1207, Nov. 1974.

[28] J. Savir, "Good Controllability and Good Observability Do Not Guarantee Testability," *IEEE Trans. Computers,* vol. 32, no. 12, pp. 1198-1200, Dec. 1983.

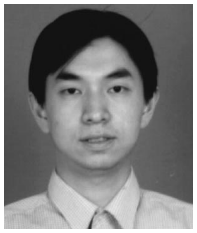
[29] N. Tamaramalli and J. Rajski, "Constructive Multi-Phase Test Point Insertion for Scan Based BIST," *Proc. IEEE Int'l Test Conf.,* pp. 649-658, 1996.

[30] D. Xiang and J.H. Patel, "A Global Algorithm for the Partial Scan Design Problem Using Circuit State Information," *Proc. IEEE Int'l Test Conf.,* pp. 548-557, Oct. 1996.

[31] D. Xiang and Y. Xu, "Partial Reset for Synchronous Sequential Circuits Using Almost Independent Reset Signals," *Proc. 19th IEEE VLSI Test Symp.,* pp. 82-87, Apr. 2001.

[32] D. Xiang and H. Fujiwara, "Handling the Pin Overhead Problem of DFTs for High-Quality and At-Speed Test," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 21, no. 9, pp. 1105-1113, Sept. 2002.

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, Tsinghua University from October 1996 to February 2003. He is now with the School of Software, Tsinghua University. His research interests include design and test of digital systems, including design for testability, testability analysis, and BIST, fault-tolerant computing, distributed computing, and computer networking. He authored *Digital Systems Testing and Design for Testability* (in Chinese, Science Press, 1997). He is a member of the IEEE and IEEE Computer Society.



Yi Xu received the BS degree and the MS degree, both in electronic engineering, in 1998 and 2001, respectively. His research interests include VLSI design and design for testability. He is now with Zhongxing Corporation working on CPU design as a senior engineer.

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 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 systems 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 IECE Young Engineer Award in 1977, IEEE Computer Society Certificate of Appreciation Award in 1991, 2000, and 2001, Okawa Prize for publication in 1994, IEEE Computer Society Meritorious Award in 2001. He is an advisory member of the *IEICE Transactions on Information and Systems* and an editor of the *IEEE Transactions on Computers*, *Journal of Electronic Testing*, *Journal Circuits, Systems, and Computers*, *Journal of VLSI Design*, and others. Dr. Fujiwara is a fellow of the IEEE, a Golden Core member of the IEEE Computer Society, a fellow of the IEICE (the Institute of Electronics, Information Processing and Communication Engineers of Japan), and a member of the Information Processing Society of Japan.