Handling the Pin Overhead Problem of DFTs for High-Quality and At-Speed Tests

Dong Xiang and Hideo Fujiwara

***Abstract*—The pin overhead problem of nonscan design for testability (DFT) and built—in self-test design has been an unsolved problem for a long time. A new algorithm is proposed to connect extra pins of control test points with primary inputs. An economical test point structure is introduced, in which only one gate delay is added to the corresponding functional paths inserted into a control test point. Unlike almost all of the previous nonscan DFT methods which do not handle pin overhead well, this method allows at most three extra pins. Techniques are presented to connect an extra input of a control test point to a primary input in order to avoid conflicts produced by the newly generated reconvergent fanouts. Similar techniques are proposed to connect more than one control input with the same PI. Sufficient experimental results are presented to demonstrate the effectiveness of the method.**

***Index Terms*—Conflict, delay overhead, inversion parity, nonscan design for testability, pin overhead, sequential depth for testability.**

I. Introduction

Scan design makes the scanned flip—flops controllable and observable directly [5], [23], which reduces the test generation problem to that of a combinational circuit. Test application time of scanned circuits is more than that in a nonscan design environment due to shifting tests and responses through scan chains. Greater testability improvement can be obtained when control points and observation points are inserted into different points and places away from the inputs and outputs of flip—flops unlike scan design. Nonscan design can provide at-speed test, low test application cost, and effectively enhance testability at the expense of more complex automatic test pattern generation (ATPG) compared with ATPG of full scan designed circuits. Also, the ATPG cost for well-designed circuits should be acceptable. It was shown that one can have more confidence in the stuck-at fault coverage metric when used with at-speed tests rather than with scan design [14]. The key to the test point insertion problem should be: 1) how to place test points, and 2) the way to handle the extra pins of test points. We focus on the scheme to handle extra pins of control test points.

A. Test Point Insertion

Test point insertion for testability has been studied extensively during the past decades [3], [6], [9], [10], [15], [16], [18]–[24]. It has been utilized in various designs for testability (DFT) topics, such as nonscan DFT, scan-based built-in self-test (BIST) [13], [21], establishment of cost-free scan paths [12], BIST design and nonscan DFT for RTL circuits [4], [8], and nonscan DFT and BIST for SOCs. Hayes and Friedman [9] and [10] proposed insertion of test points in a combinational circuit as a means to make the circuit fully testable and diagnosable by a test set of small cardinality. In [9], techniques were proposed to reduce pin overhead by connecting two observation points with a NAND gate. A scheme was also proposed to reduce the number of extra pins of control points by controlling all extra inputs via a single pin in [10], where each control point was implemented by an exclusive-or gate. The first scheme may still introduce some aliasing, while the second scheme can still produce many conflicts at the extra pin during signal requirement justification. Fujiwara *et al.* [6] proposed the use of a reasonable number 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.

B. The Pin Overhead Problem

The pin overhead problem was not handled well by almost all of the previous methods. Previous methods addressed this problem in several ways. 1) Many methods, such as [18], did not address the extra pin overhead reduction problem. 2) Connect the extra pins with an extra register [3], [8], [13], [21]. As for nonscan DFT, the scheme needs to shift in values of a test at extra control inputs and shift out the responses at the observation points [3] like scan design, which requires more test cycles. Test points are inserted into scan designed circuits in [13] and [21], where extra pins of control points are connected with the pseudorandom test generator. These methods have to control the test point number in order to limit the test input number. 3) Control all extra inputs by a single extra pin [11], [12]. This technique cannot improve testability of a circuit effectively when the number of test points is large enough because numerous contradictory signal requirements may occur at the extra pin during ATPG or testing. 4) Control all extra control inputs as controlling values (1 for 1-control points and 0 for 0-control points) during testing and noncontrolling values in operational mode [3], [12], [15]. This technique can cause bad fault coverage because all extra control inputs are assigned fixed values during ATPG or testing. 5) Instead of inserting test points into the circuit, test multiplexers are utilized to improve testability [4], [20], where all test multiplexers are controlled by the same control input. This scheme makes all subcircuits, preceding the places inserted test multiplexers, unobservable. 6) Multiplex extra observation points with primary outputs or boundary scanned connections for embedded systems in a boundary scan environment [4], [7], [23]. This technique makes the primary outputs in the original circuit or boundary scan connections unobservable during testing. 7) Muradali and Rajski [16] proposed a self-driven test point insertion method for nonscan designed circuits. The method [16] drove the control inputs of control test points via internal controllability points. It also adopted observability cells by switching the observation points into some observability points inside the circuit. However, they did not consider possible negative effects of reconvergences of the predecessors with the test points when driving control points by some controllable internal nodes. The observability cells may make the observability points in the original circuit hard to observe.

The following methods handled pin overhead well. Williams and Angell [23] should be a good example without considering the scan-out. They inserted a control point at the data input of a flip–flop, where the output of each flip–flop was connected with the control input of the test multiplexer inserted into the data input of the next flip–flop. All flip–flops can be controlled and observed via one extra control input and one extra output. However, it is not good to multiplex the scan-out with a primary output, which makes the primary output unobservable in test mode. Rudnick *et al.* proposed a hard-fault-oriented observation point insertion method to enhance testability and provide

Manuscript received December 4, 2001; revised February 21, 2002 and April 8, 2002. This work was supported in part by the National Science Foundation of China under Grant 69773030 and Grant 985, by the Fundamental Research Foundation of the Education Ministry of China, by the Japan Society for the Promotion of Science (JSPS) under a Grant-in-Aid for Scientific Research B(2) (09480054), and by Semiconductor Technology Academic Research Center (STARC) under Research Project 973. This paper was recommended by Associate Editor K. Chakrabarty.

D. Xiang is with the Institute of Microelectronics, Tsinghua University, Beijing 100084, 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 (e-mail: fujiwara@is.aist-nara.ac.jp).

Publisher Item Identifier 10.1109/TCAD.2002.801099.

at-speed testing by combining an aliasing minimization technique [19]. It is found that aliasing seems to be trivial when two exclusive-or chains are utilized to connect observation points. Touba and McCluskey [22] selected test points by path tracing instead of probabilistic testability analysis. The extra input of each control point was driven by the output of the AND gate, which is connected with two or more primary inputs. Xiang *et al.* [24] proposed a nonscan DFT method based on a conflict-analysis-based testability measure called *conflict*. A new test point structure was utilized, which makes the proposed method economical in area, delay, and pin overheads by connecting extra pins with primary inputs.

C. Organization of This Paper

Techniques are proposed to connect extra inputs of control points with the primary inputs in order to avoid potential conflicts generated by the new reconvergent fanouts, where the potential conflicts may generate new redundant faults. Sequential depth for testability and inversion parity are utilized to connect extra inputs of control points with primary inputs. Our method tries to avoid new reconvergent fanouts which may cause new untestable faults, if possible. More than one extra pin can be connected with the same primary input, which makes the proposed method able to obtain even better testability improvement than scan design.

In the rest of this paper, definitions and summary of *nscan* [24] are presented in Section II. A procedure to select test points is presented in Section III. Techniques to avoid negative effects when connecting extra inputs of control points for nonscan DFT are introduced in Section IV. Good applications of the proposed method and experimental results are presented in Section V.

II. Preliminaries

A. Definitions and Notation

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 noncontrolling values of an AND gate are 0 and 1, respectively. The main cause of conflict is reconvergent fanouts with nonuniform inversion parities.

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

The easiest way mentioned in Definition 1 and later in this paper is determined by the *conflict* measure [24]. A simplified metric is utilized to estimate $\mathrm{inv}_v(A,\, B)$. Inversion parity $\mathrm{inv}_v(A,\, B)$ 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 is only a path of odd inversion parity from A to B; 3) the easiest way to justify $(B,\, v)$ passes is only a path of even inversion parity from A to B; 4) the easiest way to justify $(B,\, v)$ passes is at least one path of even inversion parity and one path of odd inversion parity from A to B, respectively.

Definition 2: Sequential depth for testability $\mathrm{seq}_v(l,\, s)$ $(v \in \{0,\, 1\})$ from a fanout stem s to a line l is defined as the number of clock cycles required to justify a signal requirement $(l,\, v)$ at line l to the fanout stem s in the easiest way.

B. Summary of nscan

The i *controllability* $C_l(i)$ of node l should reflect the potential number of conflicts (or possibility to cause 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. 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. v *Observability* $O_A(v)$ $(v \in \{D,\, \overline{D}\})$ reflects the number of conflicts (or possibility to cause 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. We can calculate the controllability measures as follows. Consider a 2-input AND gate with inputs A, B, and an output y,

$$C_y(0) = \min(C_A(0),\, C_B(0))$$
$$C_y(1) = C_A(1) + C_B(1) + p$$

where $p = 10 \cdot n$, n is the number of reconvergent fanouts s in the circuit with $\mathrm{inv}_1(A,\, s) \neq \mathrm{inv}_1(B,\, s)$, and none of them is 00; also $\mathrm{seq}_1(A,\, s) = \mathrm{seq}_1(B,\, s)$. Here 10 is an empirical constant. Calculations of other types of gates are similar. Calculation of observability includes interdependence between fault effect activation and fault effect propagation. More details of the *conflict* measure can be found in [24].

The nonscan DFT method *nscan* selects test points based on the *conflict* measure and the selective tracing algorithm. The selective tracing scheme can be illustrated as follows: controllability of the output of a gate should be updated if the controllability of one input of the gate gets changed. Observability of an input of a gate should be updated if observability of its output is changed or another input of the same gate gets changed controllability with respect to the noncontrolling value. More than one control point can be connected with the same primary input, which makes the nonscan design for testability method get even better fault coverage than scan design. However, [24] did not present the scheme to connect control points with primary inputs.

III. Test Point Selection

The following procedure is utilized to select test points. In the procedure, *ncp* and *nop* are the number of control points selected and the number of observation points selected up to now. Calculation of the *conflict* measure can be completed in less than a half hour for all iscas89 and iscas93 benchmark circuits using an Ultra 10 workstation. The proposed method updates testability of the corresponding part using the selective tracing scheme after each test point has been selected. Equation (1) is the gain function used to select test points. In (1), F is the fault set with changed controllability if a test point is inserted into a node, $\triangle C_{\overline{i}}(A)$ is the controllability improvement of fault A/i, $\triangle O_A(v)$ is the observability improvement of the fault (v is D for fault $A/0$, and $\overline{D}$ for fault $A/1$)

$$TG = \sum_{A/i \in F} (\triangle C_{\overline{i}}(A) + \triangle O_A(v)). \tag{1}$$

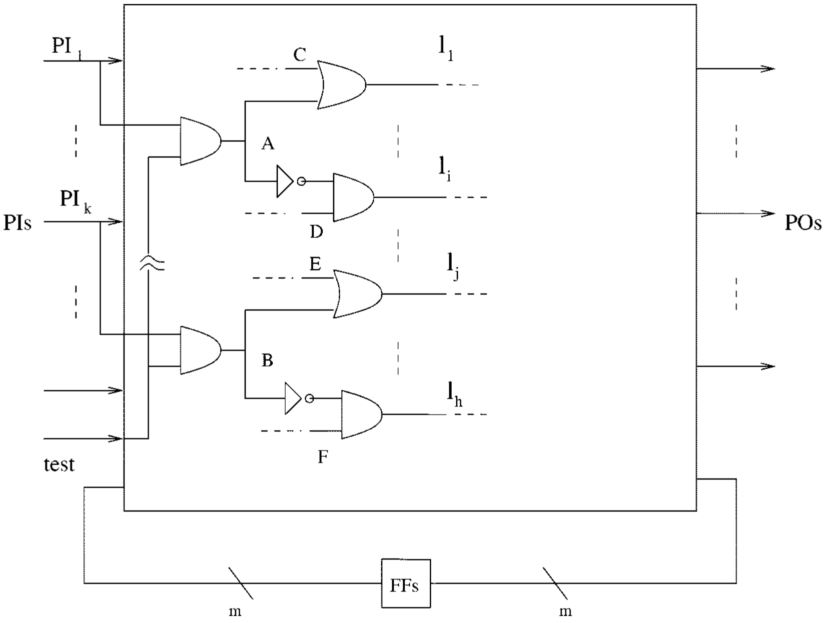

Fig. 1. Low-overhead nonscan DFT.

In the following procedure, TIC is the test point insertion candidate set.

Procedure Test-Point-Selection$()$:

1) Calculate the *conflict* measure as illustrated in [24].
2) While test point selection still has not been completed, do:
    a) for each element $v \in$ TIC, calculate testability gain according to (1) and the selective tracing procedure as stated above;
    b) select the node with the most testability gain, and insert the selected test point into the circuit;
    c) update the *conflict* measure with respect to the test point selected, update sequential depth for testability and inversion parity, update the TIC set.
3) Connect extra pins of control points with PIs using techniques introduced later in the paper and randomly place observation points into the exclusive-or trees.

It should be noted that the process for this method to select test points is different from *nscan*. The selective tracing method updates sequential depth for testability, inversion parity, and testability measures after each test point is selected. The *conflict* measure is calculated only once during the whole process of DFT.

IV. Test Point Connection for Low-Overhead Nonscan DFT

Ghosh and Jha [8] connected control ports of test multiplexers with PI ports in order to reduce pin overhead of nonscan DFT for RTL circuits. Dey and Potkonjak [4] presented techniques to avoid generating equal weight reconvergent fanout regions when inserting test multiplexers and connecting them with the same PI port. This technique is a little pessimistic because not all equal weight reconvergent fanouts cause conflicts during ATPG. It is possible for a large number of different control points to share the same PI in gate-level circuits. Fig. 1 presents the general DFT structure of the proposed method. Nodes $l_1, \ldots, l_i, \ldots, l_j, \ldots, l_h$ are inserted into a control test point, respectively. Control points $l_1, \ldots, l_i$ share the same primary input $PI_1, \ldots$, and $l_j, \ldots, l_h$ share the same primary input PI_k, respectively. An AND gate is used as a switching logic, which is connected with all control points sharing the same PI. Similarly, an OR gate or other types of gates can also be used as the switching logic. The extra input of an 1-control point is connected with the AND gate directly, while the extra input of a 0-control point is connected with the AND gate via an inverter. All switching gates are connected with a single extra input *test*. The circuit is set to the test mode when $test = 1$, while it is set to the operational mode if $test = 0$. Techniques are proposed to avoid new untestable faults when connecting extra pins of control points with primary inputs.

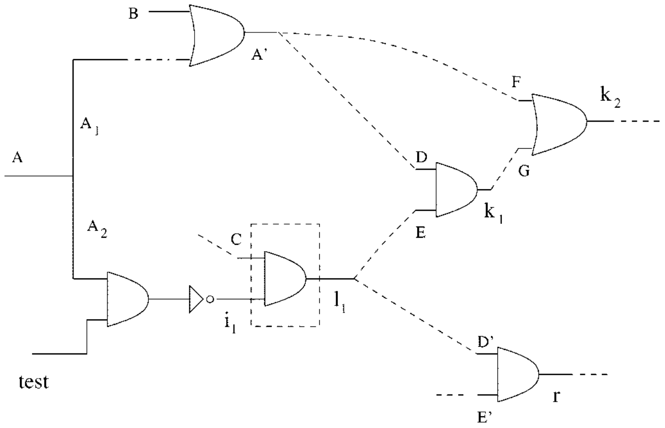

Fig. 2. Avoidance of conflicts generated by fault effect propagation.

A. Avoidance of Conflicts Generated by Fault Effect Propagation

Assume a 0-control test point is inserted into node l_1 as shown in Fig. 2, whose extra input i_1 is connected with a primary input A through a switching gate. A new fanout is generated at A. The new reconvergent fanout may cause problems to some testable faults during fault effect propagation. It is necessary to check whether fault effect propagation of faults at the fanout branches A_1 and A_2 and fanout stem A generates conflicts. Faults on all reconvergent fanout stems and branches along the EFEP paths between k_1 or k_2 and A are also checked. Potential conflicts caused by fault effect propagation along the EFEP path are checked. The EFEP path is determined by the *conflict* measure as stated earlier. Let the EFEP path of a fault at $A_2/0$ be A_2–l_1–E–G–k_2.

First, fault effect of $A_2/0$ should be activated. That is to say, A should be assigned 1. Lines C, D, and F should be assigned 1, 1, and 0, respectively. *Sequential depth for testability* is utilized to check potential conflicts. No conflict occurs if the easiest way to justify signal requirements $(C, 1)$, $(D, 1)$, and $(F, 0)$ has no signal requirements on A_2 and $\mathrm{seq}_1(C, S)$, $\mathrm{seq}_1(D, S)$, and $\mathrm{seq}_0(F, S)$ are not equal to the numbers of flip–flops in the path segments A_2–i_1, A_2–l_1–E, and A_2–l_1–E–G, respectively. Inversion parity is used to check potential conflicts if one of the above pairs has equal value as illustrated as follows.

It is clear that $\mathrm{inv}_1(A_2, A) = 10$. Therefore, the fault effect propagation condition of fault $A_2/0$ is not met if one of $\mathrm{inv}_1(C, A)$, $\mathrm{inv}_1(D, A)$, and $\mathrm{inv}_0(F, A)$ is not 10 or 00 and the corresponding sequential depth for testability conditions are not satisfied. Different faults at the same line may have different EFEP paths. Let the EFEP path of fault $A_2/1$ be A–l_1–D'–r. It is necessary to check whether $\mathrm{inv}_0(A_2, A)$ and $\mathrm{inv}_1(E', A)$ are compatible. It should be noted that $\mathrm{inv}_0(A_2, A) = 01$. To meet the fault effect propagation condition of fault $A_2/1$, $\mathrm{inv}_1(E', A)$ should be 00 or 01 if $\mathrm{seq}_1(E', A)$ is equal to the number of flip–flops in the EFEP path between A and D' in this case. Similar schemes can be adopted to check potential conflicts when propagating fault effects on line A_1.

The possible conflicts when propagating fault effects of faults on line A should be checked as follows. Let the EFEP path of fault $A/1$ $(i \in \{0, 1\})$ be A–A_2–l_1–E–G–k_2. The fault effect propagation condition has been met if all $\mathrm{inv}_1(C, A)$, $\mathrm{inv}_1(D, A)$, and $\mathrm{inv}_0(F, A)$ are equal to 00. Otherwise, the following conditions should be checked if any one of them is unequal to 00. Let $\mathrm{inv}_1(C, A) \neq 00$, then no conflict occurs if all paths from A to C have uniform inversion parity as that of the path segment A–i_1 in the EFEP path of $A/1$. Let the EFEP path of $A/0$ still be A–A_2–l_1–E–G–k_2 and $\mathrm{inv}_1(C, A) \neq 00$. No conflict occurs if all paths from A to C have uniform inversion parity as that of the path segment A–i_1.

B. Avoidance of Conflicts Generated by Signal Requirement Justification

Techniques are used to check whether potential conflicts occur when justifying the signal requirements that need to assign noncontrolling values to all inputs at the reconvergent points of the newly generated fanouts after the above fault effect propagation conditions have been met. When the extra input i_1 of a 0-control point l_1 is connected with a primary input A (it is similar for an 1-control point), two different classes of conflicts should be avoided: a) meeting the signal requirement that needs to assign noncontrolling values on all its inputs at l_1 should generate no conflict at the fanout stem A and b) signal requirements that need to assign noncontrolling values on all inputs of the convergent points such as k_1 and k_2 between l_1 and A should generate no conflict.

Let a 0-control test point be inserted into node l_1; two different classes of conflicts should be avoided in order to connect the extra input i_1 of l_1 with a primary input A (or a pseudoprimary input) as shown in Fig. 2. First, signal requirements $(i_1, 1)$ and $(C, 1)$ should generate no conflict at A if l_1 is reachable from A in the original circuit. Second, assume l_1 converges with A at an AND gate k_1 or an OR gate k_2, a signal requirement $(k_2, 0)$ or $(k_1, 1)$ should generate no conflict at A.

The idea of finding a matching primary input for the extra input of a control point can be illustrated as follows: a) First, our method tries to find a PI which is not convergent with the node l inserted a control point, and l is unreachable from the PI in the original circuit. b) Try to find a PI, where the sequential depth for testability from PI to l is unequal to 0 if l is reachable from PI. c) Try to find a PI, where l converges with PI and all the reconvergences are unequal weight ones. d) Try to find a PI, where the sequential depth for testability from PI to l is equal to 0 and the reconvergence is of uniform inversion parity. e) Try to find a PI, where l converges with PI and all the reconvergences are of equal weight and uniform inversion parity.

Procedure *connect-control-input* selects a primary input to connect the extra input of a control test point, which generates unequal weight reconvergences. It selects a primary input to connect the extra input of a control point, which generates reconvergences with uniform inversion parity if the above condition is not met.

Procedure Connect-Control-Input$()$:

1) Connect the control input i of the control test point l with a primary input A, where l does not converge with A and is not reachable from A if possible.
2) Select the primary input A as a candidate to connect the extra control input i of control point l, where the fault effect propagation conditions as illustrated in Fig. 2 are met. If the fault effect propagation conditions are met, check the signal requirement justification conditions as follows.
3) Connect the extra input of a control test point with a primary input A to avoid newly generated conflicts called *connect-for-different-weight-reconvergence$()$*.
4) When the above schemes cannot select a matching primary input to connect the extra input of the control point, call *connect-for-uniform-inversion-parity$()$*.

In the rest of this section, we shall illustrate the procedures based on Fig. 3. A 1-control test point with an extra input i_2 and 0-control test point with an extra input i_1 are inserted into l_2 and l_1, respectively.

Procedure Connect-for-Different-Weight-Reconvergence$()$:

1) Connect the control input i_1 (or i_2) of the 0-(or 1-)control test point l_1 (or l_2) with a primary input A, where A does not converge with l_1 (or l_2) and l_1 (or l_2) is reachable from A with $\mathrm{seq}_1(C, A) \neq 0$ [or $\mathrm{seq}_0(C', A) \neq 0$] if possible.
2) Connect the control input i_2 of a 1-control point l_2 with a primary input A, where l_2 is unreachable from A, but does converge with A at an AND or NAND gate k_1' (or OR or NOR gate k_2') with $\mathrm{seq}_1(F', A) \neq \mathrm{seq}_1(D', A)$ [or $\mathrm{seq}_0(E', A) \neq \mathrm{seq}_0(G', A)$] if possible.
3) Connect control input i_1 of a 0-control point l_1 with a primary input A, where l_1 is unreachable from A, but does converge with A at an AND or NAND gate k_1 (or OR or NOR gate k_2) with $\mathrm{seq}_1(D, A) \neq \mathrm{seq}_1(F, A)$ [or $\mathrm{seq}_0(E, A) \neq \mathrm{seq}_0(G, A)$] if possible.
4) Connect the control input i_2 of the 1-control point l_2 with a primary input A, where l_2 is reachable from A with $\mathrm{seq}_0(C', A) \neq 0$, and l_2 converges with A at an AND or NAND gate k_1' (or OR or NOR gate k_2') with $\mathrm{seq}_1(D', A) \neq \mathrm{seq}_1(F', A)$ [or $\mathrm{seq}_0(E', A) \neq \mathrm{seq}_0(G', A)$] if possible.
5) Connect the control input i_1 of a 0-control point l_1 with a primary input A with $\mathrm{seq}_1(C, A) \neq 0$, and C converges with A at an AND or NAND gate k_1 (or OR or NOR gate k_2) with $\mathrm{seq}_1(D, A) \neq \mathrm{seq}_1(F, A)$ [or $\mathrm{seq}_0(G, A) \neq \mathrm{seq}_0(E, A)$] if possible.

The procedure *connect-for-different-weight-reconvergence$()$* connects extra inputs of control test points with primary inputs, which does not generate equal weight reconvergences. The procedure *connect-for-uniform-inversion-reconvergence$()$* is adopted to connect the extra input i of a control point l with a primary input, which generates only reconvergent fanouts with uniform inversion parity. The following techniques are utilized. a) Connect the extra input i

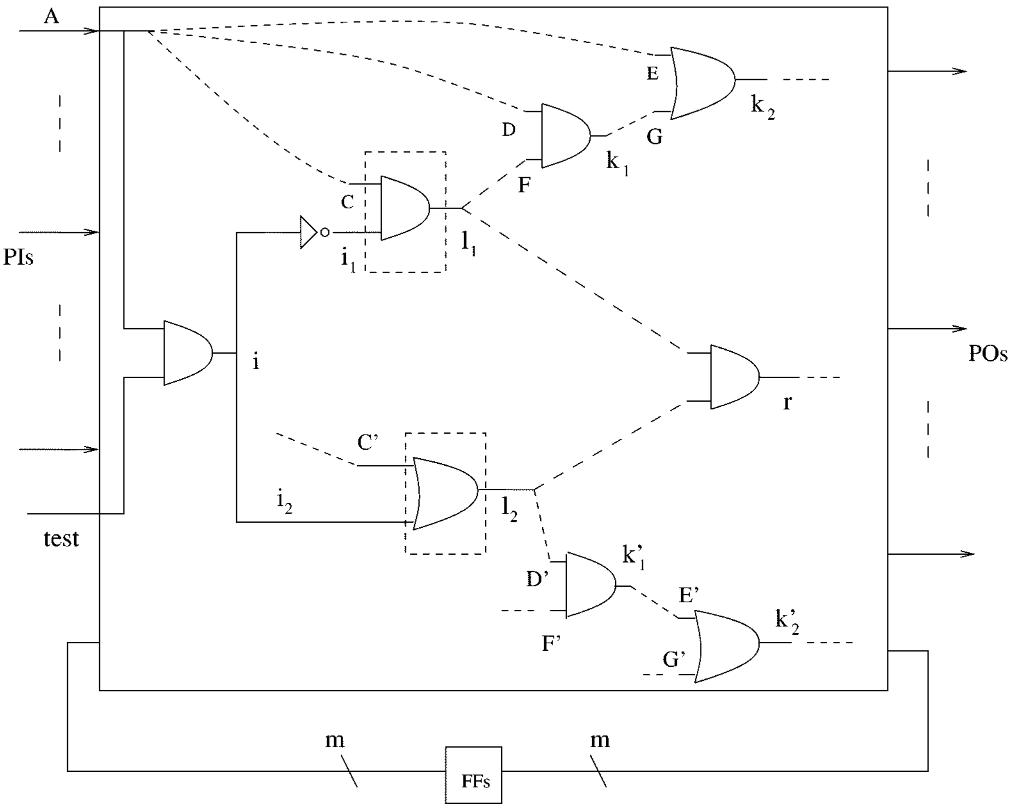

Fig. 3. Avoidance of conflicts generated by signal requirement justification.

with a primary input A, where l is unreachable from A. b) Connect the extra input i of l with a primary input A, where l is reachable from A but does not converge with A. c) Connect the extra input i of l with a primary input A, where l is reachable from A and converges with A via uniform inversion parity paths.

Procedure Connect-for-Uniform-Inversion-Parity$()$*:*

1) Connect the control input i_2 of the 1-control test point l_2 with a primary input A, where l_2 is unreachable from A, but does converge with A at an AND or NAND gate k'_1 (or OR or NOR gate k'_2) with $\mathrm{seq}_1(D', A) = \mathrm{seq}_1(F', A)$ [or $\mathrm{seq}_0(D', A) = \mathrm{seq}_0(F', A)$] and $\mathrm{inv}_1(F', A) = 0$ [or $\mathrm{inv}_0(G', A) = 0$], if possible.
2) Connect the control input i_1 of a 0-control point l_1 with a primary input A, where l_1 is unreachable from A, but does converge with A at an AND or NAND k_1 (or OR or NOR k_2) with $\mathrm{seq}_1(D, A) = \mathrm{seq}_1(F, A)$ [or $\mathrm{seq}_1(E, A) = \mathrm{seq}_1(G, A)$] and one of $\mathrm{inv}_1(D, A)$ and $\mathrm{inv}_1(F, A)$ [or $\mathrm{inv}_0(E, A)$ and $\mathrm{inv}_0(G, A)$] is 0, if possible.
3) Connect the extra control input i_1 (or i_2) of the 0-control (or 1-control) test point l_1 (or l_2) with a primary input A, where l_1 (or l_2) does not converge with A and l_1 (or l_2) is reachable from A with $\mathrm{seq}_1(C, A) = 0$ [or $\mathrm{seq}_0(C', A) = 0$] and $\mathrm{inv}_1(C, A) = 0$ [or $\mathrm{inv}_0(C', A) = 0$] for a 0-control (or an 1-control) test point if possible.
4) Connect the control input i_2 of the 1-control point l_2 with a primary input A, where l_2 is reachable from A with $\mathrm{seq}_0(C', A) = 0$ and $\mathrm{inv}_0(C', A) = 0$, and l_2 converges with A at an AND or NAND gate k'_1 (or OR or NOR gate k'_2) with $\mathrm{seq}_1(D', A) = \mathrm{seq}_1(F', A)$ [or $\mathrm{seq}_0(E', A) = \mathrm{seq}_0(G', A)$] and one of $\mathrm{inv}_0(F', A)$ or $\mathrm{inv}_0(D', A)$ [or $\mathrm{inv}_0(E', A)$ and $\mathrm{inv}_0(G', A)$] is 0, if possible.
5) Connect the control input i_1 of the 0-control point l_1 with a primary input A, where l_1 is reachable from A with $\mathrm{seq}_1(C, A) = 0$ and $\mathrm{inv}_1(C, A) = 0$, and l_1 converges with A at an AND or NAND k_1 (or OR or NOR gate k_2) with $\mathrm{seq}_1(D, A) = \mathrm{seq}_1(F, A)$ [or $\mathrm{seq}_0(G, A) = \mathrm{seq}_0(E, A)$] and one of $\mathrm{inv}_1(D, A)$ and $\mathrm{inv}_1(F, A)$ [or $\mathrm{inv}_0(G, A)$ and $\mathrm{inv}_0(E, A)$] is 0, if possible.

C. Sharing Primary Inputs

When the number of control points is greater than the number of primary inputs, more than one control point can be connected with the same PI. Let two control points with extra inputs i_1 and i_2 be inserted into l_1 and l_2, respectively, and l_1 and l_2 converge at gate r with inputs a and b (a and b are reachable from l_1 and l_2, respectively) as shown in Fig. 3. The following schemes can be adopted to reduce as many as possible negative effects of newly generated reconvergent fanouts. a) Two control points are connected with the same primary input if they do not converge in the original circuit. b) Two control points can be connected with the same primary input if both points converge with unequal sequential depth for testability. c) Two control points are connected with the same primary input if they converge at a gate with one of the reconvergent fanout branches of 0 inversion parity.

Procedure Share-Primary-Input$()$*:*

1) Connect extra inputs i_1 and i_2 of control points l_1 and l_2 with the same primary input A, where l_1 and l_2 do not converge in the original circuit if possible.
2) Select the primary input A as a candidate to connect the extra control inputs i_1 and i_2 of control points l_1 and l_2, where the fault effect propagation conditions as illustrated in Fig. 2 are met. If the fault effect propagation conditions are met, check the signal requirement justification conditions as follows.

TABLE I
COMPARISON WITH *nscan* [24] ON THE ISCAS CIRCUITS

circuit	nscan				lcdft				
	tp/po	FC/TE	vec	cpu	tp/po	ao	FC/TE	vec	cpu
s1423	40/2	93.6/94.6	607	2132	40/2	5.3	94.1/95.0	274	2078
s5378	60/2	97.3/99.5	1337	6584	60/2	3.0	97.5/99.5	2599	6695
s9234	160/3	92.8/95.7	3685	8045	120/1	3.2	94.8/97.2	1884	2703
s9234.1	160/3	90.9/94.8	2946	9832	120/1	3.2	94.8/97.2	1884	2702
s13207	240/3	91.8/94.9	3927	13488	240/1	3.6	96.3/99.4	5044	16165
s13207.1	240/3	91.2/94.5	4023	15720	240/1	3.7	96.7/99.3	5059	20795
s15850	240/3	94.2/97.6	8583	8441	240/3	3.3	93.2/97.5	4007	7976
s15850.1	240/3	94.0/97.5	5151	9934	240/3	3.5	92.4/96.6	3097	10403
s35932	200/3	90.9/100	318	1694	200/3	1.6	90.9/100	257	3325
s38417	580/3	80.5/82.7	1531	36.5h	580/3	3.3	85.7/87.6	9452	98317
s38584	400/3	91.6/94.5	8908	59757	400/3	2.6	92.6/94.9	8820	52178
s38584.1	400/3	91.4/93.8	10043	63268	400/3	2.6	93.7/96.0	11205	40439

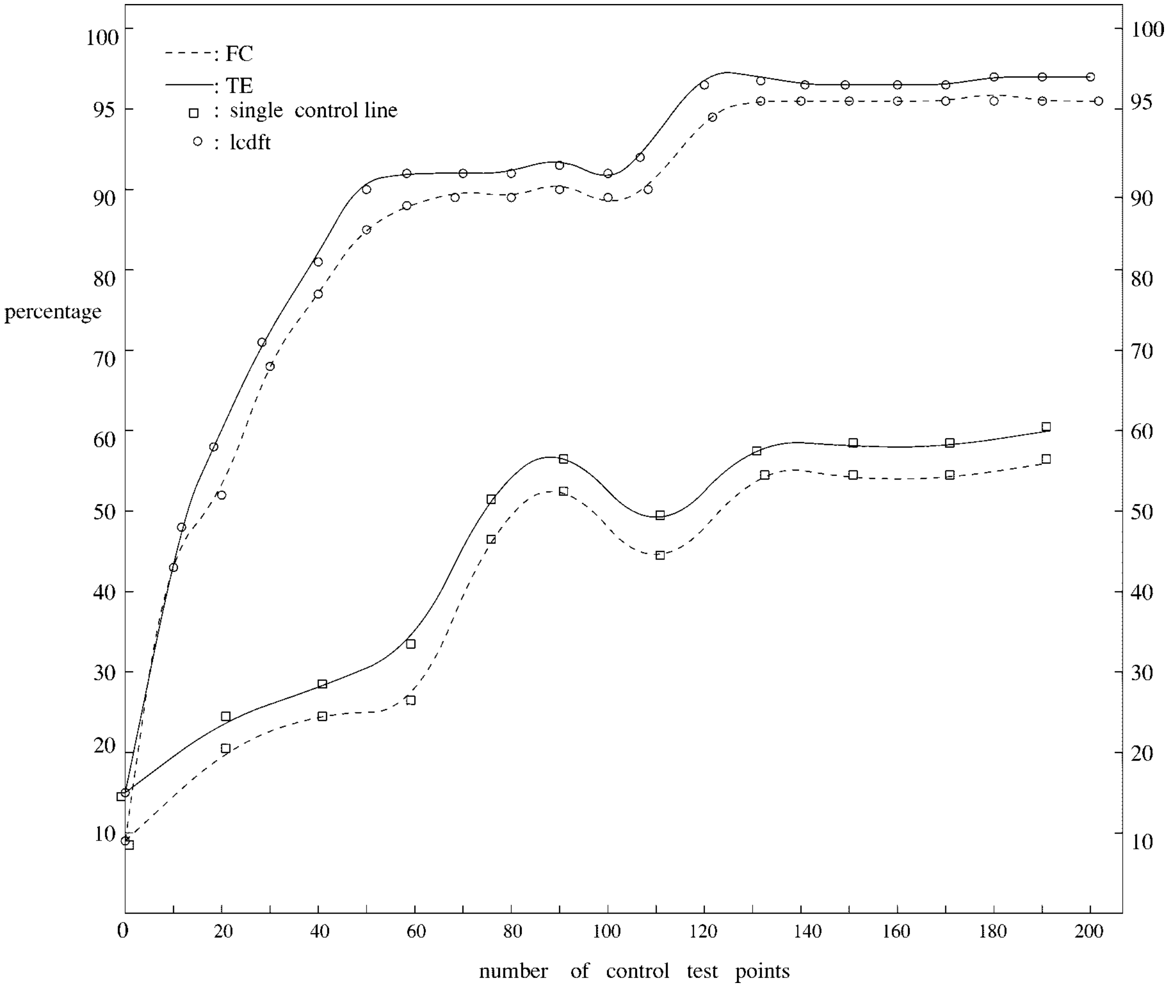

Fig. 4. Testability improvement for s9234.1 with various number of test points.

3) When l_1 and l_2 converge at gate r (with inputs a and b, respectively) in the original circuit, connect the extra inputs i_1 and i_2 with a primary input A using the following schemes:
   - $\mathrm{seq}_1(a, A) \neq \mathrm{seq}_1(b, A)$ if r is an AND or NAND gate.
   - $\mathrm{seq}_0(a, A) \neq \mathrm{seq}_0(b, A)$ if r is an OR or NOR gate.
4) When a primary input meeting the conditions in 2) and 3) is unavailable, l_1 and l_2 converge at r with inputs a and b; connect both extra inputs i_1 and i_2 with a primary input using the following schemes.
   - One of $\mathrm{inv}_1(a, l_1)$ and $\mathrm{inv}_1(b, l_2)$ is 0 if r is AND or NAND.
   - One of $\mathrm{inv}_0(a, l_1)$ and $\mathrm{inv}_0(b, l_2)$ is 0 if r is OR or NOR.

The exclusive-or chain scheme is adopted to connect all observation points in all experiments of this paper. There may exist some aliasing when the number of observation points is large and a single exclu-

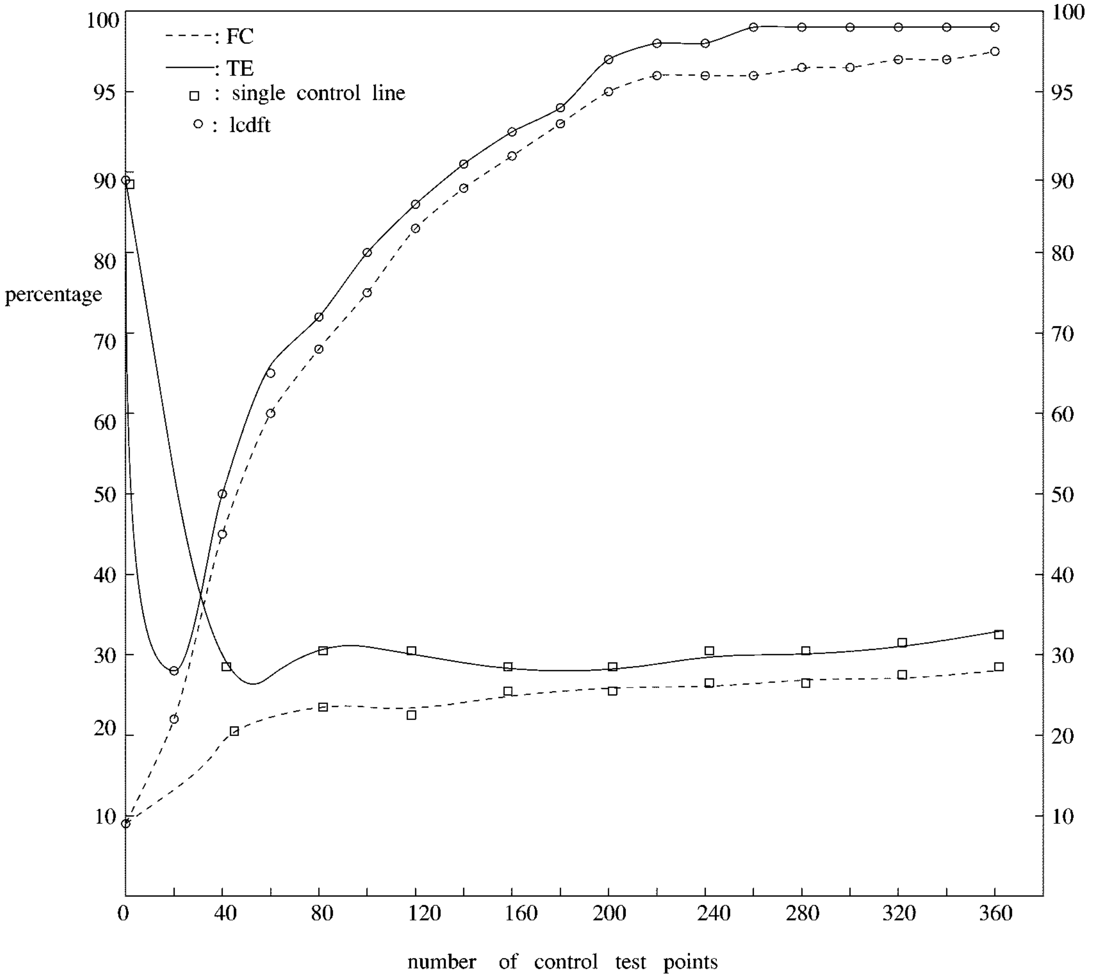

Fig. 5. Testability improvement for s13207.1 with various number of test points.

sive-or chain is utilized [19]. It is found that one or two or a little more exclusive-or trees are sufficient to avoid aliasing.

V. Experimental Results and Applications

The seven cases, as mentioned in Section I, for previous methods to handle extra pins cannot get good enough results. Case 1) causes unacceptable pin overhead when the number of test points is large enough; therefore, their method cannot insert enough number of test points for good testability. Case 2) can cause test application and hardware overhead problems. Case 6) may cause problems during ATPG and testing because some lines in the original circuit become unobservable in test mode. Cases 3)–5) generate similar results, which are unable to obtain good enough testability. The most important reasons why our method outperforms almost all the above ones are: i) our method drives all control points via primary inputs based on a new test point structure that can control all control test points by almost independent signals; ii) a very good testability measure called *conflict* is adopted to select test points; iii) techniques are adopted to connect extra pins of the control test points with PIs to avoid negative effects; and iv) more than one control point can be connected with the same PI, which makes our method obtain even better fault coverage than scan design.

A. Experimental Results

A system called low-cost nonscan design for testability (*lcdft*) has been implemented based on the method presented in this paper running on an Ultra 10 workstation. Table I shows the HITEC [17] test generation results of the large iscas89 benchmark circuits compared with the recent nonscan design method *nscan* [24]. The system *lcdft* gets even better fault coverage for most circuits than *nscan*, especially the ones that need a large number of control test points to get satisfactory fault coverage. As shown in Table I, *tp, po, FC, TE, vec., cpu,* and *ao* represent the number of test points, the number of extra pins, fault coverage (%), test efficiency (%), the number of test vectors, cpu time (seconds), and area overhead (%). Area estimation is presented based on cell library *class.lib* of SYNOPSYS. Routing complexity is still not included because of resource constraint. The number of test points can be reduced when a better test generator is utilized.

The proposed method works very well for large circuits, which need a couple of control points to obtain good enough testability improvement. The system *lcdft* gets better fault coverage for circuits s1423, s5378, s9234, s9234.1, s13207, s13207.1, s38417, s38584, and s38584.1. The system reaches much better fault coverage results for five hard-to-test sequential circuits s9234, s9234.1, s13207, s13207.1, and s38417. The system *lcdft* obtains a little worse fault coverage than *nscan* for circuits s15850 and s15850.1. Experimental results for nonscan designed s9234.1 and s13207.1 with various numbers of test points are presented in Figs. 4 and 5 in order to show the potential of PIs as testability improvement resources. In Table II, *ncp*, *nop*, *ao*, and *po* represent the number of control points, the number of observation points, area overhead, and pin overhead, respectively.

TABLE II
DESIGN FOR TESTABILITY RESULTS WITH *lcdft*

cir.	FFs	PIs	ncp	nop	pin	FC	TE	vec	cpu	ao
s9234	211	36	160	0	1	95.43	97.75	1760	2565	4.2
s9234.1	211	36	160	0	1	95.43	97.75	1760	2635	4.2
s13207	638	62	240	0	1	96.32	99.38	5044	16165	3.6
s13207.1	638	62	320	0	1	97.27	98.82	6636	3721	4.9
s15850	534	77	112	128	3	93.21	97.46	4007	7976	3.3
s15850.1	534	77	110	130	3	92.37	96.63	3097	10403	3.5
s38417	1638	28	400	180	3	85.68	87.62	9452	98317	3.3
s38417	1638	28	500	200	3	89.80	92.06	8170	62911	4.1
s38417	1638	28	600	200	3	91.67	93.89	5988	48829	4.6
s3271	116	26	18	0	1	99.82	100	298	984	1.5
s3330	132	40	50	0	1	93.49	93.95	805	4107	3.4
s3384	183	43	40	0	1	96.59	96.71	137	2603	2.3
s4863	104	49	30	0	1	99.50	99.90	647	2521	1.6
s6669	239	83	18	0	1	99.85	99.85	413	335	0.6

B. Applications of the Proposed Method

Scan-Based BIST: Random testability may still be not good enough even for fully scanned circuits. Test points are inserted in order to get good enough testability. However, almost all previous methods do not handle the extra pins well after test points have been inserted. Most of the current methods connect the extra pins with the pseudorandom pattern test generator. It is not good if the number of test points is large, which can make the number of test input unacceptable. The proposed method can be adopted to connect extra pins of control test points in fully or partially scanned circuits. All pseudoprimary inputs and primary inputs can be connected with extra inputs of test points.

Pin Overhead Reduction for Partial Reset: Most of the previous partial reset methods control all partial reset signals via a single extra pin, which cannot effectively improve testability of a circuit. The proposed method can also be applied to extra pin overhead reduction for partial reset with a little modification. Partial reset signals are inserted into data inputs (or outputs) of flip–flops and connected with primary inputs using the proposed method. The partial reset technique has an attractive property, that is, at most one gate is inserted into any functional path. The partial reset signals can be inserted away from critical paths if necessary. It is found that inserting partial reset signals into outputs or inputs of flip–flops gets quite close results.

Test Point Insertion in Partially Scanned Circuits: The proposed method can be extended to test point insertion in partially scanned circuits for deterministic testability. Extra pins of control test points can be connected with primary inputs and pseudoprimary inputs as stated above. Test application time and test power cannot be reduced greatly via scan cell ordering and test vector ordering. Test application time and test power can be effectively reduced compared to the scanning more flip–flops while fault coverage and test efficiency can be comparable or even better.

Input Reduction for BIST: The size of the pseudo-random pattern test generator may be very large if the number of inputs of the circuit is large enough. Techniques are proposed to reduce the number of test inputs. These techniques try to combine more than one input together, which can greatly reduce test cost without any test efficiency degradation. Most of the previous methods are time consuming and highly dependent on the deterministic test generator. The compatibility of inputs can be determined by the algorithms proposed in this paper. Two primary inputs should be compatible if no new reconvergent fanout generated by combining the inputs together, which causes potential conflicts.

VI. CONCLUSION

A new method was proposed to handle the pin overhead problem for nonscan DFT, which connects extra pins of control points with primary inputs by using an economical (in pin, delay, and area overheads) test point structure. Extra inputs were connected with PIs in order to avoid new untestable faults. The control test points contribute to delay overhead of the method, which were inserted away from the critical paths if necessary. It was shown that the negative effects caused by the new reconvergent fanouts generated by connecting control points with PIs are trivial and use of PIs as testability improvement resources is definitely a good choice. The nonscan DFT method generates even better fault coverage than scan design and presents at-speed testing.

ACKNOWLEDGMENT

D. Xiang would like to express his thanks to Prof. J. Patel of the University of Illinois at Urbana Champaign for his kind encouragement. The authors would like to thank Y. Xu for implementation of the method.

REFERENCES

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

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

[3] T. H. Chen and M. A. Breuer, "Automatic design for testability via testability measures," *IEEE Trans. Computer-Aided Design*, vol. 4, pp. 3–11, Jan. 1985.

[4] S. Dey and M. Potkonjak, "Non-scan design for testability techniques using RTL design information," *IEEE Trans. Computer-Aided Design of ICAS*, vol. 16, pp. 1488–1506, Dec. 1997.

[5] E. B. Eichelberger and T. W. Williams, "A logic design structure for LSI testability," in *Proc. ACM/IEEE Design Automation Conf.*, 1977, pp. 462–468.

[6] H. Fujiwara, Y. Nagao, T. Sasao, and K. Kinoshita, "Easily testable sequential machines with extra inputs," *IEEE Trans. Comput.*, vol. 24, pp. 821–826, Aug. 1975.

[7] H. Fujiwara, *Logic Testing and Design for Testability*. Cambridge, MA: MIT Press, 1985.

[8] I. Ghosh, A. Raghunathan, and N. K. Jha, "Design for hierarchical testability of RTL circuits obtained by behavioral synthesis," *IEEE Trans. Computer-Aided Design*, vol. 16, pp. 1001–1014, Sept. 1997.

[9] J. P. Hayes and A. D. Friedman, "Test point placement to simplify fault detection," *IEEE Trans. Comput.*, vol. 23, pp. 727–735, July 1974.

[10] J. P. Hayes, "On modifying logic networks to improve diagnosability," *IEEE Trans. Comput.*, vol. 23, pp. 56–72, Jan. 1974.

[11] H. C. Liang and C. L. Lee, "Effective methodology for mixed scan and reset design based on test generation and structure of sequential circuits," in *Proc. 8th IEEE Asian Test Symp.*, Nov. 1999, pp. 173–178.

[12] 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*, vol. 17, pp. 838–851, Sept. 1998.

[13] C. J. Lin, Y. Zorian, and S. Bhawmik, "Integration of partial scan scan and built-in self test," *J. Electron. Testing: Theory and Applicat.*, vol. 7, pp. 125–137, Aug.–Oct. 1995.

[14] 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%?," in *Proc. IEEE Int. Test Conf.*, 1991, pp. 358–364.

[15] A. Motohara and H. Fujiwara, "Design for testability for complete test coverage," *IEEE Design Test Comput.*, vol. 1, pp. 25–32, Nov. 1984.

[16] F. Muradali and J. Rajski, "A self-driven test structure for pseudorandom testing of nonscan sequential circuits," in *Proc. 14th IEEE VLSI Test Symp.*, 1996, pp. 17–25.

[17] T. Niermann and J. Patel, "HITEC: A test generation package for sequential circuits," in *Proc. Eur. Conf. Design Automation*, 1991, pp. 214–218.

[18] I. Pomeranz and S. M. Reddy, "Design-for-testability for path delay faults in large combinational circuits using test points," *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 333–343, Apr. 1998.

[19] 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*, vol. 13, pp. 1051–1056, Aug. 1994.

[20] E. M. Rudnick, V. Chickermane, P. Banerjee, and J. H. Patel, "Sequential circuit testability enhancement using a nonscan approach," *IEEE Trans. VLSI Syst.*, vol. 3, pp. 333–338, Apr. 1995.

[21] H. C. Tsai, K. T. Cheng, and S. Bhawmik, "On improving test quality of scan-based BIST," *IEEE Trans. Computer-Aided Design*, vol. 19, pp. 928–938, Aug. 2000.

[22] N. A. Touba and E. J. McCluskey, "Test point insertion based on path tracing," in *Proc. 14th IEEE VLSI Test Symp.*, 1996, pp. 2–8.

[23] 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, Jan. 1973.

[24] D. Xiang, Y. Xu, and H. Fujiwara, "Non-scan design for testability for synchronous sequential circuits based on conflict analysis," in *Proc. IEEE Int. Test Conf.*, 2000, pp. 520–529.