International Test Conference, pp. 520-529, October 2000.

Non-Scan Design for Testability for Synchronous Sequential Circuits Based on Conflict Analysis

Dong Xiang   Yi Xu
Institute of Microelectronics
Tsinghua University
Beijing, 100084, P. R. China
xd@dns.ime.tsinghua.edu.cn

Hideo Fujiwara
Grad. Sch. of Inform. Sci.
Nara Inst. of Sci. and Techn.
Ikoma, Nara 630-0101, Japan
fujiwara@is.aist-nara.ac.jp

Abstract

A non-scan design for testability method is presented for synchronous sequential circuits. A testability measure called *conflict* based on conflict analysis in the process of synchronous sequential circuit test generation is introduced. Reconvergent fanouts with nonuniform inversion parity is still one of the main causes of redundancy and backtracking in the process of sequential circuit test generation. A new concept called *sequential depth for testability* is introduced to calculate the conflict-analysis-based testability measure. Potential conflicts between fault effect activation and fault effect propagation are also checked because they are closely related. The testability measure implies the number of potential conflicts to occur or the number of clock cycles required to detect a fault. The non-scan design for testability method based on the *conflict* measure can reduce many potential backtracks, make many hard-to-detect faults easy-to-detect and many redundant faults testable, therefore, can enhance fault coverage of the circuit greatly. It is believed that non-scan design for testability using the *conflict* measure can improve the actual testability of a circuit. Extensive experimental results are presented to demonstrate the effectiveness of the method.

1 Introduction

Test generation for highly sequential circuits is quite complex. 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. Values of the state lines can be controlled and observed completely by using full scan. Therefore, full scan 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. Test application time is higher than that in a non-scan design due to shifting tests and responses through scan chains. However, test application can be reduced by using parallel scan chains with the cost of more extra pins. 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.

Non-scan design can provide at-speed test, low test application cost, and effectively enhance testability. Test point insertion have been extensively used in various issues of design for testability. Hayes and Friedman [12] and Saluja and Reddy [19] proposed insertion of test points in a combinational circuit as means to make the circuit fully testable by a test set of small cardinality. Fujiwara et al. [8] and Pradhan [16] 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. Recently, papers [6,14] presented techniques to achieve complete fault efficiency by modifying the state transition table of a sequential machine, which can make all hard-to-reach states easily reachable.

Recent liturature [4,5,10,17,18,21,24] tend to place test points based on testability analysers. Chicker-

Supported in part by the national science foundation of China under grant 69773030, in part by Japan Society for the Promotion of Science (JSPS) under Grant-in-Aid for Scientific Research B(2) (No. 09480054) and in part by Semiconductor Technology Academic Research Center (STARC) under the Research Project (No. 973)."

mane, Rudnick, Banerjee and Patel [18] presented a greedy procedure to load flip-flops at data inputs of flip-flops and place observation points at any internal nodes using SCOAP [11]. Control points at data inputs of flip-flops can make the loaded flip-flops to be combinational elements like scan design. Rudnick, Chickermane and Patel proposed a hard-fault-oriented observation point insertion method to enhance testability and provide at-speed test by combining an aliasing minimization technique [17]. Tamarapalli and Rajski [21] presented a multi-phase 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. Cheng and Lin [4] proposed a timing-driven test point placement method based on the COP measure and a special testability gain estimation method called gradient approach.

Most of the above methods used the classic testability measures just like SCOAP [11] or COP [3]. The measures did not cope with the influences of reconvergent fanouts well. However, reconvergent fanouts have great effects on testability. Additionally, the COP measure [3] is unable to handle sequential circuits. It is essential to present a good testability measure which can reflect the actual testability of a 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, Raghunathan, and Jha [10] proposed a non-scan design for testability of RTL circuits using a testability measure independent of data path widths. Xiang [23] proposed a testability measure called SCTM for combinational circuits by intensively checking potential conflicts of justification and fault effect propagation. SCOAP was replaced by SCTM to guide test pattern generation in the FAN algorithm [9]. Experimental results showed that SCTM outperforms SCOAP for most of the iscas85 circuits.

We shall propose a conflict-analysis-based measure for synchronous sequential circuits. Test points according to the conflict-analysis-based measures are placed in order to reduce as many as possible potential conflicts, or make as many as possible redundant faults testable. 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 analyse potential conflicts during test generation of a synchronous sequential circuit; potential conflicts between fault effect activation and fault effect propagation signal assignments are checked because fault effect activation and fault effect propagation are closely related [20].

2 Preliminaries

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 *non-controlling value* v of inputs of a gate with an output y is that the value of y can be determined only when all inputs are set v; the output y of the gate can be determined if only one of its inputs is set the *controlling* value. We define the *selecting-all-value* of the output of a gate as the value which needs to set all inputs of the gate as the non-controlling value; the *selecting-one-value* of the output of a gate is the value which needs to set only one of the inputs of the gate as the controlling value.

Definition 1 *A conflict is defined as follows: A line l is assigned value v, in the previous process of test generation, l needs to be assigned value v'. If intersection of v and v' produces a new covered value, the line l is assigned $v \cap v'$; otherwise, a conflict occurs on l.*

When all assignments are necessary, a conflict indicates the fault under consideration is redundant; otherwise, it can be resolved by backtracking. The main cause of conflicts is still reconvergent fanouts with nonuniform inversion parities.

Definition 2 *Inversion parity of a path is defined as the number of inversions in the path modula 2. Inversion parity $inv_v(B, A)$ ($v \in \{0, 1\}$) from node A to B is defined as a two binary bit number: (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 number of clock cycles required to justify a signal requirement (l, v) at the line l to the fanout stem s in the easiest way.*

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 the signal requirement passing no flip-flop. It should be noted that sequential depth for testability is quite different from sequential depth that considers only the circuit structure. *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 for the conflict-analysis-based measure is the easiest path to propagate a fault effect D or $\overline{D}$ on a 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. The EFEP path can be partitioned into stem segments, where a stem segment is the path segment between two fanout stems. *$v-$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. We would like to calculate *sequential depth for testability* and inversion parity from fanout stems which can reach the line under consideration. Calculation of inversion parity includes testability consideration. Therefore, we define $inv_v(l, s)$ as the inversion parity between l and s in the easiest way in order to set value v on l. Assume “———” is the bitwise NOT operator in Procedure 1.

Procedure 1 (inversion parity)

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

$$inv_v(l, s) = \begin{cases} 10 & \text{if } s' \text{ is } s; \\ inv_v(s', s) & \text{otherwise} \end{cases}$$

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 and v_1 are the selecting-all-value and the selecting-one-value of the gate, respectively.

$$inv_v(l, s) = inv_v(i_1, s) \vee \ldots \vee inv_v(i_n, s)$$

where “$\vee$” is the bitwise OR operator of the binary numbers.

$$inv_{v_1}(l, s) = inv_{v_1}(i_k, s)$$

where i_k is the easiest input of gate l to be controlled to value v_1.

5. Let line l be the output of a NAND or NOR gate with inputs $i_1, i_2, \ldots, i_n$, $v \in \{0, 1\}$ is the selecting-all-value of the gate, and v_1 is the selecting-one-value, we have

$$tem = inv_{\overline{v}}(i_1, s) \vee \ldots \vee inv_{\overline{v}}(i_n, s)$$

$$inv_v(l, s) = \begin{cases} \overline{tem} & \text{if } tem = 01 \text{ or } 10 \\ tem & \text{if } tem = 00 \text{ or } 11 \end{cases}$$

$$inv_{v_1}(l, s) = inv_{\overline{v_1}}(i_k, s)$$

where i_k is the easiest input to be controlled to the controlling value.

We use the following procedure to calculate the *sequential depth for testability* from a fanout stem 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 a line l is a fanout branch, for $v \in \{0, 1\}$,

$$seq_v(l, s) = \begin{cases} 0 & \text{if } s' \text{ is } s \\ seq_v(l, s') & \text{otherwise} \end{cases}$$

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 selecting-all-value and the selecting-one-value of the line l, and v_3, v_4 be the non-controlling and controlling values of the gate, respectively. Let input i be the easiest input to be controlled as the controlling value,

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

Sequential depth for testability is different from the sequential depth only according to structural analysis. 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 controlled as the controlling value,

$$\begin{aligned} seq_1(l, s) &= max(seq_1(i_1, s), \ldots, seq_1(i_n, s)) \\ seq_0(l, s) &= seq_0(i, s) \end{aligned}$$

3 The Conflict-Analysis-Based Measure *conflict*

Influences of inversion parity, and sequential depth for testability on testability are illustrated in this section. Calculation of *conflict* is also introduced.

3.1 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. In sequential circuits, sequential depth for testability of different paths should be considered.

Let us consider the example in Fig. 1(a). 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. The signal requirement $(i, 0)$ needs to meet signal requirements $(g, 1)$ and $(e, 1)$. The signal requirement $(e, 1)$ needs to meet the signal requirements $(c, 1)$, which must assign value 1 on both a and b. Simultaneously, the signal requirement $(j, 0)$ equals to $(f, 1)$ and $(h, 1)$. The signal requirement $(f, 1)$ needs to assign value 0 at line a or b. 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. 1(b). Line k can be assigned value 0 without any conflict. We can do that by the following steps: lines a and b are assigned value 1 in the first clock cycle; line a is assigned 0 while lines g and h are assigned 1 in the second clock cycle.

When there is an easy-to-control node in one of the reconvergent fanout paths, that path seems to be cut. The selecting-all-value signal requirement will not cause any conflict at the fanout stem. As for the circuit presented in Fig. 1(c), there is an easy-to-control input *in* feeding the gate *d*. It looks like the path $a-d-f-j$ being cut. We have $inv_0(j, b) = inv_0(j, a) = 00$. 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 calculation of inversion parity and sequential depth for testability.

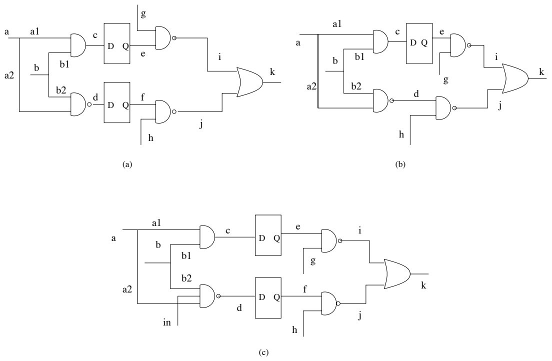

Figure 1: Conflict analysis by signal requirement justification

We would like to use the circuit as shown in Fig. 2 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.

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

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

$$C_y(1) = C_A(1) + C_B(1) + p \qquad (2)$$

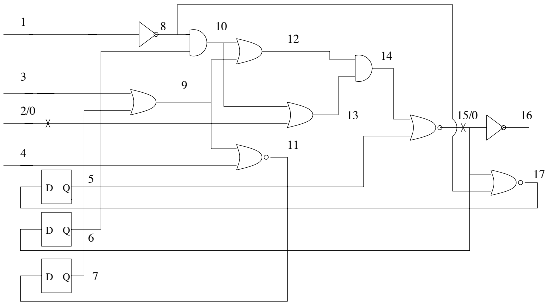

Figure 2: Different delays cause no conflict

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,

$$
\begin{aligned}
C_y(0) &= C_A(0) + C_B(0) + p & (3)\\
C_y(1) &= min(C_A(1), C_B(1)) & (4)
\end{aligned}
$$

where p can be obtained like that of an AND gate. Let i be the input of an inverter with output y,

$$C_y(v) = C_i(\overline{v}) \tag{5}$$

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,

$$C_y(v) = C_i(v) + 10 \quad (v \in \{0, 1\}) \tag{6}$$

Calculations of other types of gates are similar. It should be noted that *conflict* penalizes the selecting-all-value controllability. When a sequential loop is met, iterative calculation should be invoked like SCOAP [11].

3.2 Observability of the Conflict Measure

Observabilities are calculated assume 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. 3(a), 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. 3(b) to h, c_1 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. 3(c), 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. 3(d) 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.

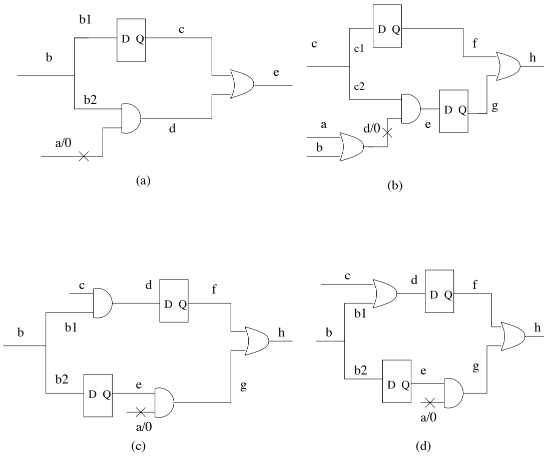

Figure 3: Conflict analysis for fault effect propagation

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. 2 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 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 [20] using previous measures. However, the fault effect activation problem and the fault effect propagation problem are closely related. Observability is calculated in *conflict* consider 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. 4, 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 of the fault.

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. 4, consider the fault effect of c is propagated along $c-e-g-s_1$, justification of signal requirements $(a,1)$, $(d,1)$, $(f,0)$ and $(h,1)$ may cause a conflict at a fanout stem s. We can get the number of potential conflicts of fault effect propagation as follows. Firstly, we check whether the signal requirement $(a,1)$ and $(d,1)$ cause conflicts according to inversion parities of the sensitization lines. If so, the observability measure is penalized. We then check whether the justification of the signal requirement $(f,0)$ and $(h,1)$ cause conflicts with the previous processed signal requirements.

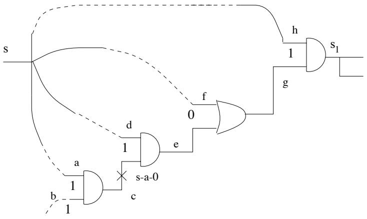

Figure 4: Conflict between fault effect activation and propagation

$$O_c(v) = O_{s_1}(v) + C_d(1) + C_f(0) + C_h(1) + p \qquad (7)$$

where $p = n \cdot 10$, n is the number of potential conflicts when propagating the fault effect from c to s_1, $\triangle p = n_1 \cdot 10$, and n_1 is the number of potential conflicts between two of d, f and h, and conflicts between a and one of d, f, and h. Let l be the output of a D flip-flop with an input i,

$$O_i(v) = O_l(v) + 10 \qquad (8)$$

4 Test Point Insertion

We have presented the conflict-analysis-based measure *conflict* in the above sections. 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). The proposed non-scan 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 or redundant faults easily testable. Therefore, we can say the proposed test point insertion scheme can enhance fault coverage directly. A circuit-state-information-based testability measure was also introduced to select scan flip-flops effectively [23], in which information of

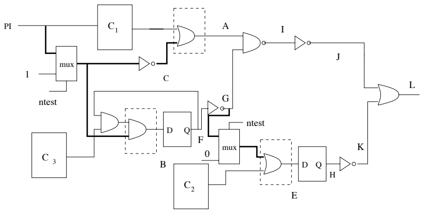

Figure 5: Mux-based DFT circuit

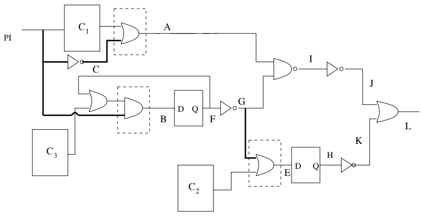

Figure 6: Test circuit with conflict-analysis-based DFT

potential conflicts is included in the circuit states obtained by logic simulation. Scan flip-flop selection tries to reduce as many as possible potential conflicts and make as many as possible hard-to-reach states easy-to-reach.

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

$$TG = \sum_{l/i \in F} (\triangle C_l(\bar{i}) + \triangle O_l(v)) \tag{9}$$

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-analysis-based measure *conflict* 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) on the basis of *conflict*;
3. using the selective tracing scheme to recalculate 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, generate tests of the DFT circuit, end the procedure. Otherwise, update TPC set, go to step 3.

The selective tracing scheme 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.

Test multiplexers are not inserted into the test point directly, which are connected with the control input of the conventional test point. 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. 5. The reason 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 at that line because all test multiplexers are controlled by the same test input. New reconvergent fanouts may be generated by the control points inserted.

Dey and Potkonjak [7] proposed a non-scan design for testability based on $k-$level controllability/observability for RTL circuits, in which a scheme to avoid generating equal weight reconvergent fanout regions when inserting test multiplexers and connecting them 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. 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. To avoid conflicts generated by the newly introduced reconvergent fanouts, a couple of effective techniques are utilized to connect the extra input of a control test point with a PI, and make a PI shared by more than one control test points if necessary.

A control test point can also be connected with an easy-to-control node in the circuit as stated in [5,18]. As shown in Fig. 5, $C_{1\sim3}$ are three subcircuits. Three test points are inserted into A, B, and E, respectively.

An $1-$control test point inserted into E is connected with node G, which can be easily controlled to 1.

Each PI can be shared by test points inserted into A and B, which generates 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 $inv_0(J, F) \neq inv_0(K, F)$.

Consider a signal requirement $(I, 0)$. Both of 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, no conflict occurs on PI because of the newly generated reconvergent fanout. 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 sequential clocks, respectively. Therefore, no conflict occurs at G.

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 [17]. It is found one or two exclusive-or chains are sufficient to avoid aliasing.

5 Experimental Results

A system called *nscan* has been completed to implement the non-scan design for testability method on E3000 server using C language. Table 1 shows the HITEC [13] ATPG results on the DFT method for almost all iscas89 and iscas93 circuits. In table 1, *tp* and *op* represent the number of test points inserted and the number of introduced extra pins. FC and TE represent fault coverage and test efficiency of a circuit, respectively. *cpu* and *vec* represent ATPG time (seconds) and the number of test vectors generated by HITEC, respectively.

The system *nscan* obtains 100% or near 100% test efficiency for almost all ISCAS and the synthesized circuits except s38417 by inserting reasonable number of test points. More than 90% fault coverage has been obtained for all circuits except s38417.

The system *nscan* gets good fault coverage and test efficiency for hard-to-test circuits s526, s526n, s9234, s13207, s15850, s15850.1, s38584, and s38584.1 as shown in Table 1. HITEC gets 80.5% fault coverage and 82.7% test efficiency after 580 test points have been inserted into s38417 by nscan.

Papers [5,18] presented results of quite a few circuits. Therefore, a system called *opus-ns* has been implemented according to the method proposed in Chickermane, Rudnick, Banerjee and Patel [5,18]. As for *opus-ns*, *npi* (the number of PIs) flip-flops of the circuit are loaded at the data inputs of the flip-flops, and $ntp - npi$ observation points are then inserted. It is shown that *nscan* consistently gets better fault coverage and test efficiency than *opus-ns* except circuits s386, and s4863.

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. Opus-ns and nscan get the same fault coverage for circuits s641, s820, s832 and s1196. The system *nscan* obtains better fault coverage and test efficiency than *opus-ns* for all remaining circuits.

Like other sequential test generators, HITEC needs a long sequence of test vectors to cover a hard-to-detect fault. Nscan makes a couple of hard faults testable. This is the most important reason why nscan generates more test vectors than opus-ns for a number of benchmark circuits. Pin overhead for all DFT circuits in experiments of this paper is no more than 3 for all circuits, which includes one extra control input for all test multiplexers and one or two extra outputs for outputs of the exclusive-or chains. Constants 1 and 0 related to all test multiplexers can be connected with *set* or *reset* inside a chip or a system-on-a-chip. No constant multiplexer is inserted in experiments of this paper. Test points can be inserted to avoid critical paths if necessary. However, only control test points in this approach contribute to delay overhead. The system *nscan* inserts fewer control points than observation points for the largest circuits.

6 Conclusions

A non-scan design for testability for synchronous sequential circuits was proposed using a conflict-analysis-based testability measure. A testability measure called *conflict* based on conflict analysis during test generation was introduced. 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 to generate 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 analyse potential conflicts; (2) interdependence between fault effect activation and fault effect propagation signal assignments were checked intensively; (3) a new concept called *sequential depth for testability* was introduced to calculate the *conflict* measure. A new test point structure is introduced to enhance testability of the circuits. Test points are inserted on the basis of *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 experimental results were presented to demonstrate the effectiveness of the method by comparison with the previous method. Dong Xiang would like to express his thanks to Prof. Janak Patel of Univ. of Illinois at Urbana Champaign for presentation of HITEC, synthesized circuits and above all kindly encouragements.

References

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

[2] V. D. Agrawal and M. R. Mercer, "Testability Measures–What Do They Tell Us ?" Proc. of *IEEE Int Test Conf.*, pp. 401-406, 1982.

[3] F. Brglez, "On Testability of Combinational Networks," Proc. of *IEEE Int. Symp. on Cir. & Sys.*, pp. 221-225, 1984.

[4] K. T. Cheng and C. J. Lin, "Timing-Driven Test Point Insertion for Full Scan and Partial Scan BIST," Proc. of *IEEE Int. Test Conf.*, pp. 506-514, 1995.

[5] V. Chickermane, E. M. Rudnick, P. Banerjee, and J. H. Patel, "Non-Scan Design for Testability Techniques for Sequential Circuits," Proc. of *ACM/IEEE Design Automation Conf.*, pp. 236-241, 1993.

[6] D. K. Das, S. Ohtake, and H. Fujiwara,"New DFT Techniques of Non-Scan Sequential Circuits with Complete Fault Efficiency," Proc. of 8-th *IEEE Asian Test Symposium*, pp. 263-268, 1999.

[7] S. Dey and M. Potkonjak, "Non-Scan Design for Testability Techniques Using RT-Level Design Information," *IEEE Trans. CAD*, 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, 1975.

[9] H. Fujiwara, Logic Testing and Design for Testability, the MIT press, 1985.

[10] I. Ghosh, A. Raghunathan, and N. K. Jha, "Design for Hierarchical Testability of RTL Circuits Obtained by Behavioral Synthesis," *IEEE Trans. CAD*, Vol. 16, No. 9, pp. 1001-1014, 1997.

[11] L. H. Goldstein, "Controllability/Observability Analysis of Digital Circuits," *IEEE Trans. Cir. and Sys.*, Vol. 26, pp. 685-693, 1979.

[12] J. P. Hayes and A. D. Friedman, "Test Point Placement to Simplify Fault Detection," Proc. of *IEEE Int. Symp. on Fault-Tolerant Computing*, pp. 73-78, 1973.

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

[14] S. Ohtake, T. Masuzawa, and H. Fujiwara, "A Non-Scan DFT Method for Controllers to Achieve Complete Fault Efficiency," Proc. of 7-th *IEEE Asian Test Symposium*, pp. 204-211, 1998.

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

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

[17] 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, No. 8, pp. 1051-1056, 1994.

[18] E. M. Rudnick, V. Chickermane, P. Banerjee, and J. H. Patel, "Sequential Circuit Testability Enhancement Using a Nonscan Approach," *IEEE Trans. VLSI Sys.*, Vol. 3, No. 2, pp. 333-338, 1995.

[19] K. K. Saluja and S. M. Reddy, "On Minimally Testable Logic Networks," *IEEE Trans. Computers*, Vol. 23, No. 11, pp. 1204-1207, 1974.

[20] J. Savir, "Good Controllability and Good Observability Do Not Guarantee Testability," *IEEE Trans. Computers*, Vol. 32, No. 12, pp. 1198-1200, 1983.

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

[22] D. Xiang, "SCTM: A Signal Conflict Oriented Testability Measure," *Chinese Journal of Computers*, Vol. 16, No. 4, pp. 273-280, 1993 (in Chinese).

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

[24] D. Xiang, D. Z. Wei, and S. S. Chen, "A Global Test Point Placement Algorithm for Combinational Circuits," Proc. of 5-th *Int. VLSI Design Conf.*, pp. 227-232, 1992.

Table 1: Performance of *nscan* on the ISCAS and synthesized circuits

circuit	tp/po	**orig.**			**nscan**			**opus-ns**		
		PI	FFs	FC/TE	FC/TE	vec	cpu	FC/TE	vec	cpu
s298	2/2	3	14	86.0/94.5	98.1/100	551	9	95.9/99.7	340	229
s344	1/1	3	14	95.3/98.5	98.3/100	115	64	93.9/96.8	70	314
s349	2/2	9	15	95.4/99.1	98.6/100	218	64	98.0/99.7	112	392
s382	4/1	3	21	90.9/92.9	96.6/100	1720	955	90.8/94.5	552	572
s386	3/3	7	6	81.8/100	96.7/100	308	1.7	97.2/100	278	6.5
s400	3/3	3	21	75.1/77	95.1/100	532	4721	91.0/97.0	482	607
s444	3/3	3	21	78.7/83.1	93.8/100	958	470	91.3/95.4	590	551
s510	6/1	19	6	0.0/100	97.9/100	433	8.4	99.7/100	584	9
s526	6/2	3	21	9.2/11.7	91.7/97.9	2013	5148	77.6/85.6	890	2678
s526n	6/2	3	21	9.95/12.3	92.0/97.9	2824	5525	79.0/84.4	504	2192
s641	1/1	35	19	86.5/100	99.4/100	283	3.45	99.4/100	282	6.47
s713	1/1	35	19	81.9/100	93.1/100	276	5.82	93.0/100	258	8.6
s820	3/3	18	5	95.7/100	100/100	717	10.9	100/100	631	17.3
s832	3/3	18	5	93.9/100	98.4/100	745	19.6	98.4/100	716	20.5
s953	3/3	16	29	8.3/100	99.4/100	385	9.6	98.1/99.9	539	304
s1196	1/1	14	18	99.8/100	99.8/100	450	3.25	99.8/100	448	3.3
s1238	2/2	14	18	94.7/100	96.9/100	461	4.53	94.7/100	450	5.2
s1423	40/2	17	74	38.2/38.9	93.6/94.6	607	2132	83.5/84.9	161	5812
s1488	3/3	8	6	97.2/100	100/100	691	103	99.9/100	626	86
s1494	3/3	8	6	96.5/100	99.2/100	716	63	98.3/100	517	167
s5378	60/3	35	179	68.4/72.9	97.3/99.5	1337	6584	96.9/99.3	1187	8987
s9234	160/3	36	211	9.3/13.9	92.8/95.7	3685	8045	39.7/42.4	611	9424
s13207	240/3	62	638	8.9/88.0	91.8/94.9	3927	13488	52.7/58.8	1816	11101
s15850	240/3	77	534	6.6/17.4	94.2/97.6	8583	8441	77.9/82.3	2484	5978
s15850.1	210/3	77	534	38.0/48.5	93.8/97.7	4205	8077	79.5/83.8	2770	5528
s35932	160/3	35	1728	89.2/99.4	90.9/100	318	1694	89.9/100	316	2847
s38417	580/3	28	1636	3.57/4.90	80.5/82.7	1531	36.5h	9.6/10.9	597	68406
s38584	400/3	38	1426	60.9/66.4	91.6/94.5	8908	59757	87.8/93.0	6321	76172
s38584.1	390/3	38	1426	61.5/67.0	91.5/94.2	8297	58254	87.6/92.7	6503	79420
s967	3/3	16	29	7.1/100	100/100	415	17	98.4/99.7	637	107
s991	3/3	65	19	2.2/100	100/100	97	0.08	99.9/100	76	1.0
s1269	12/2	18	37	17.9/58.4	97.6/99.4	188	2621	96.5/99.9	245	82
s1512	12/2	29	57	4.9/96.2	100/100	3375	700	16.5/94.5	112	1704
s3271	9/2	26	116	98.7/98.8	99.2/99.4	692	6724	98.8/99.2	802	732
s3330	40/2	40	132	73.3/77.6	92.0/97.9	722	15538	81.9/84.4	571	10611
s3384	40/2	43	183	88.9/88.9	98.3/98.5	180	1355	89.6/89.7	236	8160
s4863	9/2	49	104	95.2/95.2	98.5/98.5	391	1805	99.3/100	371	504
s6669	9/2	83	239	99.0/99.0	99.9/99.9	327	2659	99.6/99.6	286	850
am2910	6/2	20	87	90.8/98.1	93.1/99.9	1610	1520	90.6/97.7	1016	2123
div16	35/2	33	50	78.3/84.7	92.7/98.5	287	102	89.2/96.2	241	2076
mult16	30/2	18	55	89.9/90.8	99.5/100	309	7.2	98.0/98.6	222	890