IEEE the 11th Asian Test Symposium (ATS'02), pp. 86-91, Nov. 2002.

Non-Scan Design for Testability Based on Fault Oriented Conflict Analysis

Dong Xiang Shan Gu
Institute of Microelectronics
Tsinghua University
Beijing, 100084
P. R. China

Hideo Fujiwara
Graduate School of Inform. Sci.
Nara Institute of Sci. and Techn.
Ikoma, Nara 630-0101
Japan

Abstract

A two stage non-scan design for testability method is proposed. The first stage selects test points based on an earlier testability measure *conflict*. A new testability measure *conflict+* based on conflict analysis of hard-faults in the process of test generation is introduced, which emulates most general features of sequential ATPG. A new design for testability algorithm is proposed to select test points by using *conflict+*. Test points are selected in the second stage based on the hard faults after the initial ATPG run of the design for testability circuit in the preliminary stage. Effective approximate schemes are adopted to get reasonable estimation of the testability measure. Several effective techniques are adopted to accelerate the process of the proposed design for testability algorithm.

1 Introduction

Scan design places scan flip-flops into one or more scan chains. Much more test application time is necessary due to shifting tests and test responses through scan chains. Also tests cannot be applied at the speed of operational clock. Test efficiency and fault coverage parameters of at-speed test should be more dependable than those of scan design circuits [10].

It is believed that an effective testability measure is necessary to select test points for non-scan design for testability. Fujiwara [4] found that the computing complexity of testability estimation is $NP-$complete even for $3-$level monotune or unate combinational circuits. SCOAP [6] has been widely used for more than two decades, which is ineffective to analyse testability of hard-to-test circuits with complex reconvergent fanout structures. The $k-$level controllability/observability measure for RTL circuits [3] indicates the number of clock cycles required to control or observe a data paths. The $k-$level controllability/observability measure still did not consider influences of reconvergent fanouts. Ghosh and Jha [5] extracted testability from the CDFG (control/data flow graph), which was not influenced by the width of data paths. Drivability has found to be an effective fault-oriented measure to guide fault effect propagation path selection in Gentest [2]. The drivability measure is actually an extension of the SCOAP testability measure, therefore, it still did not include influences of reconvergent fanouts. Chakravarty *et al.* [1] proposed a testability measure to estimate testability of a faulty circuit with multiple faults based on conditional probabilities. Testability of a fault is the $D-$ (or $\bar{D}$) controllability of the fault at the primary outputs. Two of the recent best sequential test generators utilized conflict oriented search [7,9].

The *conflict* measure [12] intensively checks influences of reconvergent fanouts on testability of a sequential circuit. The *nscan* design for testability method inserts test points based on the *conflict* testability measure, which can obtains even better fault coverage than previous scan design methods [12]. The *lcdft* proposed a new algorithm to connect the extra pins of the control test points with primary inputs after test points have been selected, which is economical in delay, pin, and area overheads. A new testability measure is proposed with respect to hard faults. The measure is proposed based on the popular $9-$valued logic system, such as, HITEC [11], ATOMS [7], and FASTEST [8]. The proposed measure can be extended to any other multiple valued system. Intensive conflict analysis of the reconvergent fanouts is presented. Testability of a fault is D- or $\bar{D}$-controllability on primary outputs. The proposed measure *conflict+* should be more accurate than the drivability measure [2] because it includes influences of reconvergent fanouts. It should also be more accurate than the *conflict* mea-

Proceedings of the 11 th Asian Test Symposium (ATS'02)
1081-7735/02 $17.00 © 2002 IEEE

IEEE COMPUTER SOCIETY

sure because it naturally evaluate testability of a fault by a single metric but not controllability and observability measures like most of the previous testability measures. Fault effects are allowed to be propagated along multiple paths.

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 $v \in \{1, 0\}$. Many sequential test generators [7,8,11] that adopt the 9−valued logic system, can relax the fault effect propagation conditions and obtain more actual fault coverage. Values of a line are 2−tuples, where the first element represents the fault-free value and the second element the faulty value. The 9 values of the logic system are $(\times, \times)$ (U), $(1,1)$ (I), $(0,0)$ (O), $(0,\times)$ $(0u)$, $(1,\times)(1u)$, $(\times,0)(u0)$, $(\times,1)(u1)$, $(1,0)(D)$, and $(0,1)(\overline{D})$. $v-$Controllability (v is one of the 9 values) of line l indicates the number of potential conflicts occur or the number of clock cycles required to justify a signal requirement (l, v). No observability is necessary in the *conflict+* measure because testability of a fault is the D or $\bar{D}$ controllability on the primary outputs in the faulty circuit.

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

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 path set from B to A in order to justify the signal requirement (B, v).*

The main cause of conflicts is still reconvergent fanouts with nonuniform inversion parities. The easiest way mentioned in Definition 1 and later in this paper is determined by the *conflict* measure [12]. It is impossible to enumerate all those paths between A and B in a very large sequential circuit, therefore, a simplified metric is utilized to do that [12].

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 that the easiest way to justify the signal requirement (l, v) has no signal requirement (l may be unreachable from s) on the fanout stem s or pass no flip-flop. It should be noted that sequential depth for testability is quite different from sequential depth that considers only the circuit structure. It should be noted that $seq_0(l, s)$ and $seq_1(l, s)$ are not always the same, and $seq_0(l, s)$ and $seq_1(l, s)$ are both set as 0 when l is unreachable from s. $Seq_v(l, s)$ can also be 0 if signal requirement (l, v) can be met in an easier way without having any signal requirement on the fanout s.

Details to calculate the *conflict* measure, inversion parity and sequential depth for testability can be found in [13]. For a pair of values A and B, we call A dominates B if $A \cap B = A$. We can also call B contains A. It indicates that $C_l(A) \geq C_l(B)$ for a specific line l. For example, for a specific line l, we have,

$$C_l(I) \geq C_l(1u) \geq C_l(U). \quad (1)$$

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

We always have $C_l(v_1) \leq C_l(v_2)$ for a line l and any pair of values v_1 and v_2 if v_1 contains v_2 based on the 9−valued logic system. Therefore, we can only consider dominating assignments when calculating testability measures.

3 Calculations of the conflict+

The intersection table for lines which are reachable from the fault line is presented in Fig. 1 based on the 9−valued logic system. As for lines that are unreachable from the fault point, they are unable to be assigned values D, $\overline{D}$, $u1$, $u0$, $1u$ and $0u$. According to the intersection table in Fig. 1, $u0 \cap 1u = D$, there is no conflict. For a line that is unreachable from the fault point, $0 \cap 1$ generates a conflict.

Let us consider propagation of the fault effect of the fault $s/0$ as presented in Fig. 2 along the EFEP path *s-d-e-f-h-i-j*. The lines a, c, b and g should be assigned values $1u$, $u0$, $u1$ and $0u$ respectively. No conflict occurs at the line a because $seq_1(a_2, a) \neq seq_0(c, a)$. No conflict occurs at the line b because b is reachable from the fault point s, therefore, b can be assigned

IEEE COMPUTER SOCIETY

$\cap$ / B	U	u0	u1	0u	1u	D	$\overline{D}$	C	
U	U	u0	u1	0u	1u	D	$\overline{D}$	O	I
u0	u0	u0	c	O	D	D	c	O	c
u1	u1	c	u1	$\overline{D}$	I	c	$\overline{D}$	c	I
0u	0u	O	$\overline{D}$	0u	c	c	$\overline{D}$	O	c
1u	1u	D	I	c	1u	D	c	c	I
D	D	D	c	c	D	D	c	c	c
$\overline{D}$	$\overline{D}$	c	$\overline{D}$	$\overline{D}$	c	c	$\overline{D}$	c	c
O	O	O	c	O	c	c	c	O	c
I	I	c	I	c	I	c	c	c	I

Figure 1: The intersection table.

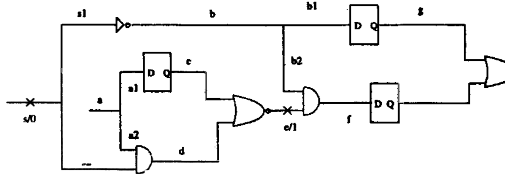

Figure 2: Example of the intersection table.

$u1 \cap 0u = \overline{D}$. Consider propagating the fault effect of the fault $e/1$ along the path e-f-h-i-j. The fault can be activated via the primary input a. The fanout stem b is unreachable from the fault point e. The intersection of 1 (b_2) and 0 (b_1) is a conflict for lines that are unreachable from the fault line.

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

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

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

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

Assume A and B are inputs of an AND gate with an output l. One of A and B can be assigned value O in order to assign O to l. Other 8 assignments can also control l as value O. They are $(\overline{D}, u0)$, $(0u, u0)$, $(\overline{D}, D)$, $(0u, D)$, $(u0, \overline{D})$, $(u0, 0u)$, $(D, \overline{D})$, $(D, 0u)$. There are only 4 containing assignments $(0u, u0)$, $(u0, 0u)$, (O, U) and (U, O) in order to control l as value O. There will be no conflict while justifying the above 4 assignments. We do not need to penalize O-controllability at the output of a 2−input AND gate. In order to control value I to the output of the AND gate, A and B should be assigned value I. While justifying the assignment, potential conflicts occur.

Assignments (D, I), (D, D), $(D, 1u)$, (I, D) and $(1u, D)$ can set the output l of a 2−input AND gate as value D. Because (D, I) and (D, D) dominate $(D, 1u)$, and (I, D) dominates $(1u, D)$, we have containing assignments $(1u, D)$ and $(D, 1u)$ for D−controllability of line l. Line l can be controlled as value $\overline{D}$ by assignments $(\overline{D}, I)$, $(\overline{D}, u1)$, $(\overline{D}, \overline{D})$, $(I, \overline{D})$ and $(u1, \overline{D})$. Assignments $(\overline{D}, I)$ and $(\overline{D}, \overline{D})$ dominate $(\overline{D}, u1)$, and assignment $(I, \overline{D})$ dominates $(u1, \overline{D})$, we can only consider assignments $(u1, \overline{D})$ and $(\overline{D}, u1)$ for $\overline{D}$−controllability of the line l. The D−controllability and $\overline{D}$−controllability are quite similar to the drivability adopted by the Gentest algorithm [2]. Equations (2)∼(9) are presented to calculate controllability of lines reachable from the fault point. D−(or $\overline{D}$)controllability of lines unreachable from the fault point are ∞. Let l be a primary input, $C_l(v) = 0$ for $v \in \{O, I, u0, u1, D, \overline{D}, 0u, 1u\}$. If l is a fanout branch steming from s, we have,

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

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

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

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

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

There is one containing assignment (I, I) for value I,

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

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

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

The expressions for other values are,

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

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

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

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

IEEE COMPUTER SOCIETY

Testability estimation of other gates should be similar. Let l be the output of an inverter with input I,

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

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

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

Calculations of other types of gates are similar. The *conflict+* measure is a hard-fault-oriented testability measure. $D-$ and $\overline{D}-$controllability measures of the lines which are unreachable from the fault point are 0. The *conflict+* measure penalizes the value of a gate which needs to control all inputs as non-controlling values (1 for AND or NAND gate, 0 for OR or NOR gate) consider potential conflicts.

Consider a line l is unreachable from the fault line, we have, $C_l(u0) = C_l(0u) = C_l(O)$, $C_l(1u) = C_l(u1) = C_l(I)$, and $C_l(D) = C_l(\bar{D}) = \infty$. Calculations of $C_l(O)$ and $C_l(I)$ are similar to those of *conflict* [12]. We can use controllability measures to represent testability of a fault. Let l/i be the fault line, we have,

$$det(l/i) = \min\left(C_{po_1}(D), C_{po_1}(\bar{D}), \ldots, C_{po_m}(\bar{D})\right), \quad (10)$$

where $po_1, po_2, \ldots, po_m$ are primary outputs, and $det(l/i)$ is the testability of fault l/i. The proposed testability measure is a fault-oriented one, calculation of which should be time-consuming if it is calculated based on separate faults. Effective approximate techniques are utilized to estimate the testability measure. First, conflict information of the *conflict* measure [12] is adopted to calculate the *conflict+* measure. The controllability measures on the values $I, O, u1, 1u$ and $0u$ are the same for all faulty circuits. The *conflict+* testability measure corresponding to one fault only handles lines that are reachable from the fault, which needs less time than that of SCOAP.

4 New Design for Testability Algorithm

As shown in Fig. 3, let a $0-$control test point be inserted into node a. The bold-faced lines are those that get changed controllability based on the selective tracing scheme and the conflict [12,13] measure. A new scheme is adopted to estimate testability gain. The ***active fault set*** is defined as faults with changed testability (with respect to *conflict+*). (i) Initially, all hard faults that reach line a should be included in the active fault set. For each b successor of a, check all faults that reach b. If the fault gets changed $D-$ or $\bar{D}-$controllability measure at line b, put the fault into the active fault set of the line b. Continue the above

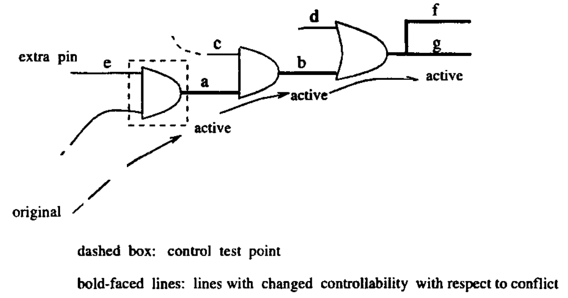

Figure 3: Illustration of the design for testability scheme.

process until out of the bold-faced range. (ii) Drive all active faults of the nodes just outside of bold-faced range until the active fault set of the line is empty or a primary output is reached. No active faults are added during the second phase.

First, all hard faults that reach the node should be considered as active fault candidates. All active faults of its predecessors should be active fault candidates of the node. An active fault candidate should be excluded if its $D-$controllability and $\bar{D}-$controllability at that node are unchanged. An active candidate should be deleted if its $D-$controllability and $\bar{D}-$controllability at the line are greater than its testability, where fault effect propagation of the fault from the node cannot obtain better testability. Testability gain is estimated according to testability of all active faults F at all primary outputs or extra observation points. Let po_1, po_2, ..., po_k be reachable from the fault f. The updated testability of a hard fault f after a control test point has been inserted is,

$$det'(f) = min(C_D(po_1), C_{\bar{D}}(po_1), \ldots, C_{\bar{D}}(po_k)) \quad (11)$$

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

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

$$det(f, l) = min(C_D(l), C_{\bar{D}}(l)), \quad (13)$$

where $C_D(l)$ and $C_{\bar{D}}(l)$ are D and $\bar{D}$-controllability measures of fault f on line l. Let $det(f, l) < det(f)$, testability gain after an observation point is inserted into l can be obtained as follows,

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

IEEE COMPUTER SOCIETY

Here F are hard faults that have a path from the faulty site to a primary output.

It should be time-consuming if testability gains of all test point candidates are recalculated after a test point is inserted. It is also unnecessary to estimate testability gains again for all test point candidates after a control test point has been inserted because a test point only makes a limited range of lines get changed testability. The following scheme is adopted to handle the problem. Testability gains of all test point candidates should be estimated for the first control test point. Our method selects the node with the greatest testability gain to insert the corresponding test point. After the test point has been inserted, testability of a limited range in the circuit gets changed testability. Testability gains of lines that get changed testability should be updated for the second test point while testability gains of the test point candidates not influenced by the inserted test point are not updated. The above process should continue until all control points are inserted. The above technique can save *very much* cpu time for very large circuits compared with the procedure that updates testability improvement potentials of all test point candidates after a control test point has been inserted. Similar technique is adopted to select observation points. After an observation point has been inserted, testability gains of only lines that are reachable from the observation point should be updated.

Procedure *test-point-selection*()

1. Calculate the *conflict+* measure based on the hard fault set of the initial stage DFT circuit after the initial run of HITEC. Select test point candidates for control points based on the *conflict+* measure.
2. While (control point selection not completed)
   (a) for each test point candidate c, obtain the region R that gets changed *conflict* measure with the selective tracing scheme when a control point is inserted into c.
   (b) Drive the active fault set from c based on techniques introduced above until out of the testability changed region R.
   (c) Drive the active faults until a primary output reaches or active fault set becomes empty based on techniques introduced above.
   (d) Get the testability improvement potentials according to equations (11) and (12). Insert a control point with the most testability gain.
   (e) Update testability gains of nodes influenced by the inserted test point.
3. Select observation points based equations (13) and (14). Place observation points into exclusive-or trees and connect extra pins of control points using techniques in paper [13].

5 Experimental Results

The fault-oriented non-scan design for testability method has been implemented and run on an ultra10 workstation. The design for testability system is called *econ*. The preliminary design for testability selects stage test points based on *lcdft* [13] and the *conflict* measure. The number of test points inserted in the initial phase is mainly determined empirically for good enough fault coverage in order to make testability analysis cost in the second stage acceptable. The HITEC test generator does an initial run on the design for testability circuit after the preliminary DFT phase has been completed. The final phase design for testability is based on the hard faults obtained from the initial run results of HITEC.

Table 1 presents comparison of *econ* with *lcdft* [13] and *nscan* [12]. The proposed method *econ* generates better results on fault coverage than *lcdft* for all circuits except s38417, s3271, and s6669. The system *econ* generates a little worse results for circuits s38417, s3271, and s6669 than *lcdft*, and the same results for circuits s5378, s13207, and s13207.1 as it. The proposed method *econ* generates better results for all circuits than *nscan* except s15850, s1512, s3384, and s6669. The new method econ gets slightly worse fault coverages on the above four circuits. The proposed system gets apparently better fault coverages than *nscan* on circuits s9234, s9234.1, s13207, s13207.1, s38417, s35932, s38584, s38584.1, and s3330. The new method econ obtains a little better fault coverage on circuits s1423, s5378, s1269, and s4863 than nscan.

6 Conclusions

A two-stage non-scan design for testability algorithm was proposed based on fault-oriented conflict analysis. In the initial phase, test points were selected based on the *conflict* [13] measure and the selective tracing algorithm. Test points were selected using the new testability measure *conflict+*, and a new design for testability algorithm based on the hard fault set in the design for testability circuit of the preliminary stage. Unlike conventional testability measures, *conflict+* does not need observability measure any more, according to which fault effects are allowed to be propagated along multiple paths. Several good techniques

IEEE COMPUTER SOCIETY

Table 1: Comparison of the proposed method with *nscan* [12] and *lcdft* [13]

circuit	econ			lcdft			nscan		
	tp/po	FC/TE(%)	vec	tp/po	FC/TE(%)	vec	tp/po	FC/TE(%)	vec
s1423	40/2	94.1/95.0	274	-	-	-	40/2	93.6/94.6	607
s5378	60/2	97.5/99.5	2599	-	-	-	60/2	97.3/99.5	1337
s9234	158/1	96.0/95.7	2539	160/1	95.3/97.8	1760	160/3	92.8/95.7	3685
s9234.1	154/1	95.5/97.8	2304	160/1	95.4/97.8	1760	160/3	90.9/94.8	2946
s13207	240/1	96.3/99.4	5044	240/1	96.3/99.4	5044	240/3	91.8/94.9	3927
s13207.1	240/1	96.7/99.3	5059	240/1	96.7/99.3	5059	240/3	91.2/94.5	4023
s15850	280/2	94.1/98.3	3571	280/2	92.7/96.9	7008	240/3	94.2/97.6	8583
s15850.1	280/2	94.2/98.7	3584	280/2	92.7/96.5	2835	240/3	94.0/97.5	5151
s35932	235/3	93.8/100	504	235/3	90.1/100	269	235/3	91.3/100	248
s38417	800/3	91.2/93.3	6839	800/3	91.7/93.9	5988	800/3	82.8/85.2	2271
s38584	499/3	94.9/97.2	12367	500/3	92.9/95.3	10600	500/3	92.0/94.4	8207
s38584.1	450/3	94.8/97.0	12038	450/3	92.8/95.1	12510	450/3	93.0/95.5	10338
s1269	12/1	99.7/100	326	12/2	98.2/99.8	352	12/2	99.4/99.7	204
s1512	9/1	99.9/99.9	4938	12/1	99.0/99.0	3391	12/1	100/100	3224
s3271	15/1	99.6/99.8	263	15/1	99.8/100	298	9/1	99.6/99.6	688
s3330	60/2	95.3/96.3	751	60/2	94.3/94.7	857	60/2	92.5/92.8	817
s3384	40/1	97.9/97.9	166	40/1	96.6/96.7	137	40/2	98.3/98.5	180
s4863	9/1	98.6/99.1	419	9/2	98.6/99.0	439	9/2	98.5/98.5	391
s6669	9/1	99.8/99.8	252	9/1	99.9/99.9	257	9/2	99.9/99.9	327

were introduced to accelerate the design for testability procedure. It was shown that the proposed DFT method outperforms two recent good non-scan design for testability methods.

References

[1] S. Chakravarty and H. B. Hunt III, "On computing signal probability and detection probability of stuck-at faults," *IEEE Trans. Computers*, vol. 39, no. 11, pp. 1369-1377, 1990.

[2] W. T. Cheng and T. J. Chakraborty, "Gentest: An automatic test generation algorithm for sequential circuits," *IEEE Computer*, vol. 22, no. 4, pp. 43-49, 1989.

[3] S. Dey and M. Potkonjak, "Non-scan design for testability techniques using RTL design information," *IEEE Trans. Computer-Aided Design of ICAS*, vol. 16, no. 12, pp. 1488-1506, 1997.

[4] H. Fujiwara, "Computational complexity of controllability/observability problems for combinational circuits," *IEEE Trans. Computers*, vol. 39, no. 6, pp. 762-767, 1990.

[5] I. Ghosh and N. K. Jha, "Design for hierarchical testability of RTL circuits obtained by behavioral synthesis," *IEEE Trans. Computer-Aided Design of ICAS*, vol. 16, no. 9, pp. 1001-1014, 1997.

[6] L. H. Goldstein, "Controllability/observability analysis of digital circuits," *IEEE Trans. Cir. and Sys.*, vol. 26, pp. 685-693, 1979.

[7] I. Hamzaoglu and J. Patel, "Deterministic test pattern generation techniques for sequential circuits," Proc. of *IEEE/ACM Int. Conf. Computer-Aided Design*, pp. 538-543, 2000.

[8] T. P. Kelsey, K. K. Saluja, and S. Y. Lee, "An efficient algorithm for sequential circuit test generation," *IEEE Trans. Computers*, vol. 42, no. 11, pp. 1361-1371, 1993.

[9] X. Lin, I. Pomeranz, and S. M. Reddy, "Techniques for improving the efficiency of sequential circuit test generation," Proc. of *IEEE/ACM Int. Conf. on Computer-Aided Design*, pp. 147-151, 1999.

[10] P. C. Maxwell, R. C. Aitken, V. Johansen, and I. Chiang, "The effect of different test sets on quality level prediction: When is 80% better than 90% ?" Proc. of *IEEE Int. Test Conf.*, pp. 358-364, 1991.

[11] T. Niermann and J. Patel, "HITEC: A test generation package for sequential circuits," Proc. of *European Conf. on Design Automation*, pp. 214-218, 1991.

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

[13] D. Xiang and Y. Xu, "Cost-effective non-scan design for testability for actual testability improvement," Proc. of 19-th *IEEE Int. Conf. on Computer Design*, pp. 154-159, 2001.

1081-7735/02 $17.00 © 2002 IEEE

IEEE COMPUTER SOCIETY