Constructive multi-phase test point insertion for scan-based BIST

Conference Paper *in* IEEE International Test Conference (TC) · November 1996

DOI: 10.1109/TEST.1996.557122 · Source: IEEE Xplore

2 authors, including:

Janusz Rajski
Mentor, A Siemens Business
324 PUBLICATIONS **8,663** CITATIONS

SEE PROFILE

Constructive Multi-Phase Test Point Insertion for Scan-Based BIST

Nagesh Tamarapalli

MACS Laboratory, Dept. of Electrical Eng.,
McGill University,
Montreal. CANADA H3A 2A7

Janusz Rajski

Mentor Graphics Corp.,
8005 S.W. Boeckman Rd.,
Wilsonville. OR 97070-7777

Abstract

This paper presents a novel test point insertion technique which, unlike the previous ones, is based on a constructive methodology. A divide and conquer approach is used to partition the entire test into multiple phases. In each phase a group of test points targeting a specific set of faults is selected. Control points within a particular phase are enabled by fixed values, resulting in a simple and natural sharing of the logic driving them. Experimental results demonstrate that complete or near-complete stuck-at fault coverage can be achieved by the proposed technique with the insertion of a few test points and a minimum number of phases.

1. Introduction

With the ever increasing complexity of present day integrated circuits, the costs of testing have become a significant part of the overall chip costs. Automatic test pattern generation (ATPG) for these complex circuits consumes considerable amounts of resources. In addition test vectors have to be ported, resulting in difficulty of testing of the embedded components. Due to these reasons Built-In-Self-Test (BIST) is finding increasing acceptance. BIST entails a significant departure from the conventional testing paradigm allowing for at-speed testing, tester cost reduction and test hardware reuse.

Of the various BIST schemes, complete scan path in combination with pseudo-random patterns is widely adopted due to its ease of implementation and attractive cost trade-off. Unfortunately, the presence of random pattern resistant faults in many practical circuits poses a serious limitation to its success. For these circuits acceptable test quality can be achieved only by applying inordinate number of random patterns, elevating the cost of test application and fault simulation beyond acceptable levels.

The solutions proposed to tackle the random pattern resistance problem can be broadly classified as those that modify the input patterns or those that modify the circuit-under-test. The first category consists of reseeding [11], [14], [26], weighted random [16], [25], and the more recent pattern mapping techniques [6], [22]. In reseeding, ATPG is used to generate test cubes for random pattern resistant faults. These cubes are then compressed and stored as seeds of an LFSR. The weighted random approach reduces test length by appropriately assigning signal probabilities to inputs of a circuit. Finally, pattern mapping techniques transform ineffective random patterns to make them compatible with the deterministic patterns by employing mapping logic.

In this paper we are interested in the second class of solution, circuit modification, that introduces test points to improve the random pattern testability of a circuit. Optimal test point insertion in circuits with reconvergent fanout has been proven to be NP-complete [15]. Numerous approximate techniques have been proposed in the literature for control and observation point insertion. The underlying philosophy of these techniques is to identify, using either exact fault simulation [5], [12] or approximate testability measures, locations in a circuit to introduce control and observation points [7], [18], [19]. Typically, control points are driven by independent, equi-probable signals and test application is performed in a single session by enabling all control and observation points simultaneously.

Most practical circuits require a large number of control and observation points in order to meet the coverage requirements. Fault simulation based selection of these large number of points suffers from excessive CPU-time requirements. Approximate testability measures, although reduce the time, do not properly capture the interaction among the control points. In fact, of the 2^K possible combinations of values at K equi-probable control points, many may be detrimental due to conflicting values. Since no attempt is made to consider this destructive correlation during the selection of control points, it is not unusual to obtain reduced coverage as increasing number of control points are inserted. This results in divergence of the solution and selection of a large number of futile control points. In addition sharing of the logic driving the control points, essential to limiting the area overhead, is not straightforward. These disadvantages render the traditional circuit modifications less applicable to the current complex designs.

Proceedings of the International Test Conference (ITC'96)
1089-3539/96 $ 10.00 © 1996 IEEE

This paper presents a novel constructive test point insertion technique based on a new probabilistic fault simulation technique [21]. The key idea is to use a divide and conquer technique to partition the test into multiple phases. As shown in Figure 1, each phase contributes to the results achieved so far, moving the solution closer to complete fault coverage. The design of each phase i.e., the selection of control/observation points is guided by the progressively reducing set of undetected faults. Within each phase, a set of control points maximally contributing to the fault coverage achieved so far is identified using a probabilistic fault simulation technique, which accurately computes the impact of a new control point in the presence of the control points selected so far. In this manner, in each phase a group of control points, driven by fixed values and operating synergistically, is enabled. In addition, observation points maximally enhancing the fault coverage are selected by a covering technique that utilizes the probabilistic fault simulation information.

The advantages of the proposed method are its ability to converge, due to partitioning, and increased accuracy in predicting the impact of multiple control points, due to the usage of fixed values. In addition, the number of ineffective control points is reduced, since conflicting control points are enabled in different phases. These features, combined with inherent sharing of logic driving the control points, lead to a technique with low area overhead. Finally, power dissipation during test mode is potentially reduced due to the usage of fixed values to drive control points and due to the distribution of control points across multiple phases.

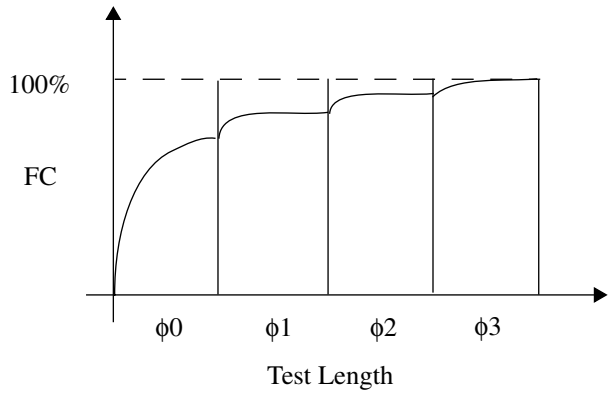


Fig. 1 Fault coverage plot of a circuit incorporating the proposed BIST scheme with 4 test phases

This paper is organized as follows. In Section 2 the proposed BIST scheme and the methodology for arriving at such a scheme are outlined. Section 3 discusses probabilistic fault simulation, a key piece in the methodology. The selection of control and observation points is explained in Section 4. Section 5 presents the experimental results. Finally, Section 6 concludes the paper.

2. The Constructive BIST Scheme and Methodology

Figure 2 illustrates main components of the proposed scan-based BIST scheme. The phase decoder block in the figure plays a key role in the scheme. The inputs of the phase decoder are driven by a pattern counter which is part of the BIST controller. The outputs of the phase decoder, which indicate the current phase in progress, drive control points inserted in the combinational logic. Given the number of phases N, the phase decoder block can be synthesized with the number of outputs ranging from $\lceil \log_2 N \rceil$ to $(N-1)$ based on the constraints on routing and area-overhead.

For the example in Figure 2a, the entire test is divided into four phases, Φ0-Φ3. The three non-overlapping outputs of the phase decoder are depicted in Figure 2b. In each phase, a set of control points is enabled by the phase decoder outputs and a specific number of patterns are applied. Each phase thus facilitates detection of a specific set of faults and contributes to the fault coverage obtained so far.

C1 and C2, in Figure 2a, illustrate the implementation of an AND and an OR type control point respectively. Only additional logic necessary for the implementation of C1 and C2 is shown within the combinational logic block. An AND type control point is realized at a node *f,* as shown in the figure, by ANDing *f* with the test signal *f_test*. The test signal is obtained by NORing the enable signals of the phases during which the control point is intended to be active, phases Φ1 and Φ2 in the case of C1. Similarly, as shown in the figure, an OR type control point is implemented by ORing the normal signal *g*, with the test signal *g_test*. OR of the enabling phases, Φ0 and Φ1 for C2 in the figure, serves as the test signal.

Table 1 lists the output values of C1 and C2 for phases Φ0-Φ3. The value of *f_test* is zero during phases Φ1 and Φ2, forcing the control point C1 to be zero irrespective of the value of *f*. However, during phases Φ0 and Φ3, *f_test*'s value is 1 and the normal signal *f* is passed to C1. Similarly, *g_test*'s value is one during phases Φ2 and Φ3 forcing the output of C2, regardless of *g*'s value, to one. During Φ0 and Φ1, *g_test* is zero, allowing the normal signal *g* to reach the output of C2.

Control points that are enabled in the same phase share the enable signal corresponding to that phase. This is illustrated in Figure 2a by the sharing of Φ2 output of the phase decoder between control points C1 and C2. Hence sharing of the logic driving the control points is inherent to this scheme. In the traditional circuit modification, this sharing is not straightforward and is attempted as a post-process after completing the selection of all control points.

The placement of an observation point is much less involved compared to that of a control point. An observation point is implemented at a node by means of an additional fanout from that node. This fanout is connected to an

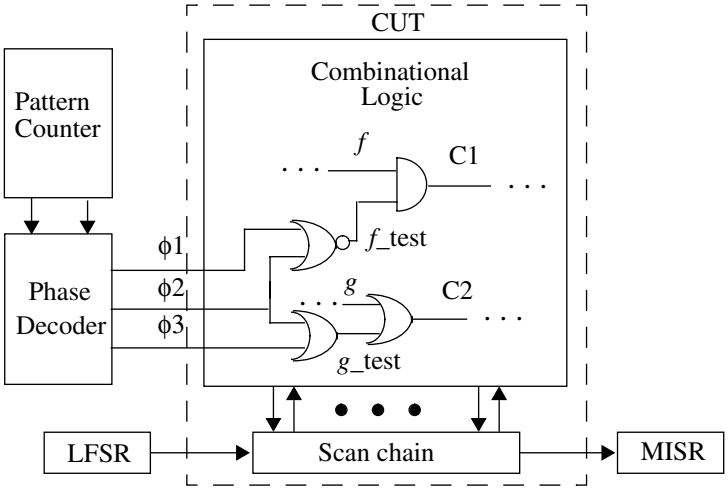


Fig. 2a Circuit incorporating the proposed BIST scheme

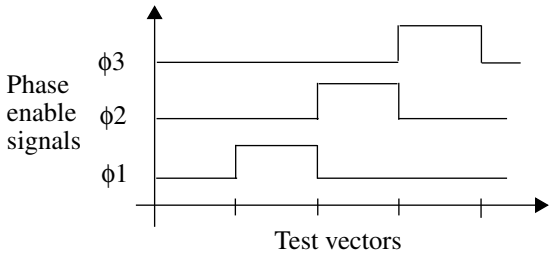


Fig. 2b Phase decoder outputs

	C1	C2
ϕ0	*f*	*g*
ϕ1	*0*	*g*
ϕ2	*0*	*1*
ϕ3	*f*	*1*

Table 1 Control point synthesis table

output response analyzer either directly or through a space compactor [8]. Observation points in the proposed scheme are kept enabled for the entire duration of the test.

Next, the constructive methodology used to design the proposed scheme is outlined. The steps involved in the methodology are depicted in Algorithm 1. Given a circuit along with parameters such as test length, target fault coverage and maximum number of control/observation points, the objective is to perform circuit modification such that target fault coverage is met within the specified test length, while satisfying the constraints on the number of control/observation points, with a minimum number of phases.

The above objective is achieved by partitioning the entire test into multiple phases and designing each phase considering the set of undetected faults. Therefore the phases are geared towards detecting a specific group of faults. In each phase a collection of control and observation points are inserted iteratively by assessing their impact on the fault coverage. The consequences of inserting a control/observation point at a circuit node are determined by means of a new technique called probabilistic fault simulation (PFS) described in Section 3. The PFS technique computes detection probability for the set of undetected faults at different nodes in the circuit using analytical methods. This detection probability information, along with a detection threshold, is then used to guide the selection of control/observation points. The value of the detection threshold needs to be determined by considering factors like the phase duration and the desired detection confidence. Notice that the detection threshold can be varied across different phases.

```
Inputs: CUT, # Random Patterns, TargetFC,
        MaxContolPoints, MaxObservationPoints
Output: Modified CUT which achieves TargetFC
Perform fault simulation;
for Phase 0 to (N-1) {
  Get the list of undetected faults;
  /* DetectionThreshold, MBPC, CPLimit, OPLimit */
  Set the design parameters;
  /* control points selection */
  for ControlPoint 0 to (CPLimit-1) {
    Perform probabilistic fault simulation;
    List the set of control point candidates;
    Evaluate the set of candidates;
    Select the best candidate;
    if(best candidate meets MBPC criterion)
      Insert the control point;
    else break;
  }
  /* observation points selection */
  Perform probabilistic fault simulation;
  for ObservationPoint 0 to (OPLimit-1) {
    List the set of observation point candidates;
    Select the best candidate;
    if(best candidate meets MBPC criterion)
      Insert the observation point;
    else break;
  }
  Perform fault simulation
  if(FC >= TargetFC) exit;
}
```

Algorithm 1 The proposed BIST methodology

The design of a particular phase is concluded when it is determined that the expected benefit of a control/observation point does not justify the associated cost of implementation. A user specified threshold minimum-benefit-per-cost (MBPC) is used for this purpose. The final step is to fault simulate the phase so that the list of undetected faults required for the design of subsequent phase is determined.

3. Probabilistic Fault Simulation

The objective of probabilistic fault simulation is to determine, for each node in the circuit, the list of faults that propagate to it, along with the associated detection probabilities. Since this information, called propagation profile, is utilized in the iterative control/observation points selection, a fast and accurate technique is necessary for its computation.

Several techniques have been suggested in the literature to compute the detection probability, defined as the probability that a fault f is detected at some primary output by a random vector [1], [2], [13], [20], [23], [24]. However, none of these techniques is directly applicable for our purpose of reliably estimating the detection probability of undetected faults at various nodes in the circuit. Hence a new method for propagation profile computation based on analytical fault propagation is proposed. The probability of detecting a fault f at a node is computed by propagating the fault forward using analytical equations. During forward propagation, information is maintained about the polarity of propagation, i.e., $D/\bar{D}$. This reflects the scenario in a circuit with reconvergent fanout where a fault can propagate to a node as a D or a $\bar{D}$. In addition, keeping the polarity information is helpful in properly capturing the reconvergence of fault effects. The probability of detecting a fault at a node is then obtained by adding two components, the probability of detecting it as a D and a $\bar{D}$ at that node.

The underlying probabilities necessary for forward fault propagation are obtained by a prior logic simulation of a pre-specified number of random vectors. For every gate, input pattern counts are updated during the logic simulation. Multiple input gates are factorized into equivalent 2-input gates so as to reduce the memory requirement. The advantage of relying on simulation to obtain probabilities is its natural ability to capture correlation.

The essential task in forward fault propagation consists of determining the probability of detecting a fault f as a D and a $\bar{D}$ at a gate output, given the respective probabilities at the inputs of the gate. The equations for detection probability computation for a 2-input AND gate, a 2-input OR gate and an inverter are illustrated in Table 2. The notation used is as follows. P_{xy}, where $xy \in \{00, 01, 10, 11\}$ denotes the probability of occurrence of a pattern xy at the inputs a and b in the good circuit. Let n_g/n_f, where $n_g, n_f \in \{0, 1\}$, for a node $n \in \{a, b, c\}$ denote its values in the good and the faulty circuits respectively. ${P_D}^n$ (${P_{\bar{D}}}^n$), where $D = 1/0$ ($\bar{D} = 0/1$), denotes the probability of occurrence of a value D ($\bar{D}$) at node n. ${P_1}^n$ (${P_0}^n$) denotes the probability of obtaining a value 1 (0) at node $n \in \{a, b, c\}$ in the good circuit. Notice that ${P_1}^n$, usually referred to as the signal probability of node n, is equal to $({P_{1/1}}^n + {P_{1/0}}^n)$. The signal probabilities of nodes a, b, and c can also be computed from appropriate pattern probabilities.

In what follows the detection probability computation at the output c of a 2-input AND gate is explained. Similar arguments apply for other gates as well. A fault f can be detected as a D (1/0) at output c of the 2-input AND gate when it is detected as D (1/0) at either input a or input b and the other input carries a value D (1/0) or 1/1. Now the probability of the event of fault detection as D at the output due to $a = D$ and $b = D$ is considered, to illustrate the method used to arrive at the final equations in Table 2.

$$P(a = D \cap b = D) = P(a_g = 1 \cap a_f = 0 \cap b_g = 1 \cap b_f = 0)$$

Applying the Bayes' rule, assuming that the fault effect at inputs a and b are independent, and noting that the good value at input b is independent of the faulty value at input a, reduces the above equation to

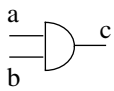
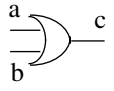
$$P(a = D \cap b = D) = \frac{P(a_g = 1 \cap b_g = 1)}{P(a_g = 1)P(b_g = 1)} {P_D}^a {P_D}^b$$

The above equation can be rewritten as follows, by observing that $P(a_g = 1)$ ($P(b_g = 1)$) is same as ${P_1}^a$ (${P_1}^b$), and that $P(a_g = 1 \cap b_g = 1)$ is same as P_{11}

$$P(a = D \cap b = D) = \frac{P_{11}}{{P_1}^a {P_1}^b} {P_D}^a {P_D}^b$$

Utilizing a similar technique to determine the probabilities of other components of ${P_D}^c$ results in the equation given in Table 2. ${P_D}^a {P_1}^b$ (${P_1}^a {P_D}^b$) in this equation gives the probability of detecting the fault as D at output c, due to the possibility of detecting it as a D at input a (b) and input b (a) carrying value D or 1/1. ${P_D}^a {P_D}^b$, is a correction term since both ${P_D}^a {P_1}^b$ and ${P_1}^a {P_D}^b$ account for the event of detecting the fault as D at output c due to simultaneous detection of fault as D at both the inputs. Finally, the multiplication factor, $\frac{P_{11}}{{P_1}^a {P_1}^b}$, serves as a correction factor to account for correlation between inputs a and b.

Table 2 Equations for detection probability computation

	$P_1^a = P_{10} + P_{11}; \quad P_1^b = P_{01} + P_{11}$ $P_D^c = \frac{P_{11}}{P_1^a P_1^b}\left[P_D^a P_1^b + P_1^a P_D^b - P_D^a P_D^b\right]$ $P_{\overline{D}}^c = \frac{P_{01}}{P_0^a P_1^b} P_{\overline{D}}^a [P_1^b - P_D^b] + \frac{P_{10}}{P_1^a P_0^b}[P_1^a - P_D^a] P_{\overline{D}}^b + \frac{P_{00}}{P_0^a P_0^b} P_{\overline{D}}^a P_{\overline{D}}^b$
	$P_0^a = P_{00} + P_{01}; \quad P_0^b = P_{00} + P_{10}$ $P_D^c = \frac{P_{01}}{P_0^a P_1^b}[P_0^a - P_{DB}^a] P_D^b + \frac{P_{10}}{P_1^a P_0^b} P_D^a [P_0^b - P_{DB}^b] + \frac{P_{11}}{P_1^a P_1^b} P_D^a P_D^b$ $P_{\overline{D}}^c = \frac{P_{00}}{P_0^a P_0^b}\left[P_{\overline{D}}^a P_0^b + P_0^a P_{\overline{D}}^b - P_{\overline{D}}^a P_{\overline{D}}^b\right]$
	$P_D^b = P_{\overline{D}}^a$ $P_{\overline{D}}^b = P_D^a$

Similarly, a fault f can be detected as a $\overline{D}$ at the output c when it is detected as a $\overline{D}$ at input a (b) and input b (a) carries a value 1/1, or it is detected as a $\overline{D}$ at inputs a and b simultaneously. The three terms in equation for $P_{\overline{D}}^c$ in Table 2, along with their respective correction factors, correspond to the probabilities of above three possibilities.

The probability of detecting a fault f at its location is equal to its excitation probability. The detection probability for the set of undetected faults is thus initialized to their respective excitation probability. Forward propagation of these faults is then performed utilizing the given equations, in a levelized, event-driven fashion. For each gate, the list of faults at its output is computed from the corresponding list of faults at its inputs. In order to reduce the memory requirement, only faults that are detected with a probability greater than a certain minimum threshold are inserted into the list. This list computation is performed until all events subside or primary outputs are reached.

The memory requirement and the accuracy of probabilistic fault simulation can be further improved by adopting a two-level method of detection probability computation. In this technique, first all undetected faults are propagated up to the output of their corresponding fanout free region. Surrogate faults are then associated with active stems, i.e., stems corresponding to the outputs of fanout-free regions with non-empty fault list. Observability of these active stems at various nodes in the circuit is computed through forward propagation of the associated surrogate faults. Hence, the list at a node may now be composed of two kinds of faults, surrogate faults and faults within the local fanout free region. The complete list of faults at a node is then obtained by appropriately expanding the surrogate faults.

4. Test Point Insertion

The selection of test points in the proposed scheme is driven by probabilistic fault simulation described in the previous section. Observation points facilitate the propagation of faults, where as control points change the signal probabilities of gates in their fanout cone and hence change the excitation and propagation of faults. The observation and control point selection method is explained in the following.

4.1 Observation Point Selection

The purpose of this step is to identify a pre-specified number of nodes that enable the detection of a maximum number of faults. This problem, which was first proposed in [10], has been addressed previously [15], [17]. Our objective, unlike that of earlier approaches, is to select the nodes such that detection probability of a maximum number of faults meets a user specified threshold DTh. A three step process, explained below, is followed to achieve this objective.

First, probabilistic fault simulation is performed to determine the propagation profile. This information is represented internally as a sparse two dimensional matrix $T_{M \times N}$, with the collected nodes as rows, undetected faults as columns, and the probability of detecting a fault j, at a node i as entry $T[i,j]$. In order to reduce the memory requirement, faults that propagate to a node with probability less than certain minimum threshold are dropped. In addition, nodes that carry less than a minimum number of faults, as determined by the minimum-benefit-per-cost criterion, are eliminated. The problem of selecting a pre-specified number of observation points, now becomes equivalent to that of selecting the set of rows which maxi-

mizes the number of columns satisfying the detection threshold DTh. Since this problem is NP-complete [9], a greedy heuristic selection method, outlined in the following, is utilized.

The following terms need to be defined for the explanation of selection process. Let $R = \{r_1, r_2, ..., r_k\}$ denote the set of k rows selected so far. The partial covering of a column j, PC_j is defined as $PC_j = \sum_{i = 1 \text{ to k}} T[r_i, j]$, where $r_i \in R$. The partial covering of a column j, represents approximate cumulative detection probability of the corresponding fault at nodes corresponding to the selected rows. A row i is said to cover a column j if $PC_j < DTh$ and $PC_j + T[i, j] \geq DTh$. Let $W_i = \sum_{j = 1 \text{ to N}} max(0, min(\ (DTh - PC_j), T[i, j]\ ))$, where $i = 1 \text{ to M}$, denote the weight of a row i.

The partial cover of all columns is initialized to zero and the selection of rows is performed iteratively, in a greedy fashion. At each iteration a row that covers a maximum number of columns is selected. When multiple rows covering the same maximum number of columns are present, their weights are used to select one among them. Subsequently partial cover of the effected columns is updated to reflect the recent selection. This selection process continues until a pre-specified number of observation points are selected or no observation point meets the minimum-benefit-per-cost criterion.

Following the completion of the selection process, an improvement of the set of selected rows is performed. Since the selection of a row at iteration *i* does not consider the effect of rows selected in subsequent iterations, the final number of columns covered by such a row could be less than the number of columns covered by it at the point of selection. Hence, the selected row that covers least number of columns at the end of selection process is returned and the partial cover of effected columns is changed accordingly. The best unselected row is then determined. The returned selected row is replaced by the best unselected row, if it covers less number of columns than the later.

4.2 Control Point Selection

It is well known that random pattern resistance cannot be entirely solved by inserting observation points alone, control points are needed as well. Control points in the proposed scheme are driven by fixed values and are aimed at improving the excitation and propagation of specific faults. A three step method is presented to insert control points. First, probabilistic fault simulation is performed to determine the propagation profile. Second, the set of candidate locations for control point insertion is determined. Finally, the best candidate is selected, by determining the benefit of each candidate through incremental probabilistic fault simulation. The steps of control point selection are explained below.

4.2.1 Candidate Signals Enumeration

The aim of this step is to quickly identify the set of control point candidate nodes. It is necessary to eliminate ineffective nodes early on so that time spent in the subsequent selection step is reduced. However, care must be taken not to miss good candidate nodes. The set of candidates are determined by an estimation technique, that computes two estimates E_0 and E_1, for various nodes in the circuit. These measures give an indication of the number of faults that could potentially be detected by placing an AND/OR type control point respectively. Nodes for which E_0 or E_1 exceeds a minimum acceptable value are retained for subsequent evaluation.

The computation of E_0 and E_1 utilizes the propagation profile information, and is driven by three pre-specified parameters viz. minimum probability threshold MPTh, low threshold LTh, and high threshold HTh. MPTh specifies the minimum acceptable value for 0 (1) probability at a node in the circuit. A node for which 0-probability (1-probability) falls below MPTh is called a 0-failing (1-failing) node. For each 0-failing (1-failing) node in the circuit, a constant 0-value (1-value) is injected at that node and the consequent signal probability changes are determined by an analytical technique described below. The parameters LTh and HTh are used to record changes in signal probability of nodes due to the insertion of a control point. A node for which 0 or 1 probability changes from a value below LTh before the insertion of a control point, to a value above HTh after the insertion of a control point is said to facilitate the excitation and propagation of certain faults. For the purpose of estimation, such faults are considered to be detected. Hence, estimation procedure predicts the number of faults that could be detected by a control point by reasoning about the impact of signal probability changes on the faults.

The effect of control point insertion at a node, on the signal probabilities, is determined by an analytical technique. This technique computes changes to the signal probabilities obtained by a prior logic simulation of few random vectors. The advantage of relying on such a technique, over logic simulation of the modified circuit with control point, is its ability to speedily analyze multiple control points. The placement of an AND (OR) type control point is simulated by injecting a constant 0 (1) as a decrement (increment) to the existing signal probability. This signal probability change is then propagated forward analytically.

The basic task in forward propagation consists of computing the signal probability change of a gate, given the changes in the signal probabilities of its inputs. Table 3 presents the equations for computation of modified pattern probabilities. Signal probability change of a gate can be obtained, depending on the gate function, from the appropriate modified pattern probability. The notation used in this table is as follows. P_{xy} (P_{xy}^*) where $xy \in \{00, 01, 10, 11\}$ denotes the probability of occurrence of

a pattern xy at the inputs a and b of a 2-input gate in the circuit without (with) control point. Δ_{xy}^{+} (Δ_{xy}^{-}) denotes the increase (decrease) in the probability of occurrence of a pattern xy. Let $n_o, n_m \in \{0, 1\}$, denote the value of an input $n \in \{a, b\}$ in the circuit without and with the control point respectively. δ_n^{+} (δ_n^{-}), denotes $P(n_m = 1 | n_o = 0)$ ($P(n_m = 0 | n_o = 1)$), the probability that a 0 (1) on input n in the original circuit changes to a 1 (0) in the modified circuit. Finally, let $rs \rightarrow xy$, where $rs, xy \in \{00, 01, 10, 11\}$, denote the event of a pattern rs at the inputs a and b in the original circuit without the control point changing to a pattern xy in the circuit with the control point.

Table 3 Equations for pattern probability change computation

$P_{xy}^{*} = P_{xy} + \Delta_{xy}^{+} - \Delta_{xy}^{-}$; $xy \in \{00, 01, 10, 11\}$
Pattern 00
$\Delta_{00}^{-} = P_{00}(\delta_a^{+} + \delta_b^{+} - \delta_a^{+}\delta_b^{+})$
$\Delta_{00}^{+} = P_{01}(1 - \delta_a^{+})\delta_b^{-} + P_{10}\delta_a^{-}(1 - \delta_b^{+}) + P_{11}\delta_a^{-}\delta_b^{-}$
Pattern 01
$\Delta_{01}^{-} = P_{01}(\delta_a^{+} + \delta_b^{-} - \delta_a^{+}\delta_b^{-})$
$\Delta_{01}^{+} = P_{00}(1 - \delta_a^{+})\delta_b^{+} + P_{10}\delta_a^{-}\delta_b^{+} + P_{11}\delta_a^{-}(1 - \delta_b^{-})$
Pattern 10
$\Delta_{10}^{-} = P_{10}(\delta_a^{-} + \delta_b^{+} - \delta_a^{-}\delta_b^{+})$
$\Delta_{10}^{+} = P_{00}\delta_a^{+}(1 - \delta_b^{+}) + P_{01}\delta_a^{+}\delta_b^{-} + P_{11}(1 - \delta_a^{-})\delta_b^{-}$
Pattern 11
$\Delta_{11}^{-} = P_{11}(\delta_a^{-} + \delta_b^{-} - \delta_a^{-}\delta_b^{-})$
$\Delta_{11}^{+} = P_{00}\delta_a^{+}\delta_b^{+} + P_{01}\delta_a^{+}(1 - \delta_b^{-}) + P_{10}(1 - \delta_a^{-})\delta_b^{+}$

The change computation for 00 pattern probability is explained in the following. Similar arguments apply for computation of other pattern probability changes as well. The probability of occurrence of a 00 pattern, P_{00}, at a gate with inputs a and b, decreases due to one of the following events $00 \rightarrow 01$, $00 \rightarrow 10$, and $00 \rightarrow 11$. Next, computation of $P(00 \rightarrow 01)$ is outlined to illustrate the method followed in arriving at the final equations in Table 3.

$$P(00 \rightarrow 01) = P(a_m = 0 \cap b_m = 1 \cap a_o = 0 \cap b_o = 0)$$

Assuming that the signal probability changes at the inputs a and b are independent, and noting that $P(a_o = 0 \cap b_o = 0) = P_{00}$, we have

$$P(00 \rightarrow 01) = P(a_m = 0 | a_o = 0) P(b_m = 1 | b_o = 0) P_{00}$$

Now, noting that $P(a_m = 0 | a_o = 0) = 1 - P(a_m = 1 | a_o = 0)$ and $P(n_m = 1 | n_o = 0) = \delta_n^{+}$, we arrive at the following final result

$$P(00 \rightarrow 01) = (1 - \delta_a^{+})\delta_b^{+} P_{00}$$

The first term in Δ_{00}^{-}, $P_{00}\delta_a^{+}$, accounts for the possibility of 00 patterns changing to 10 or 11. Similarly, the second term, $P_{00}\delta_b^{+}$, accounts for the possibility of 00 patterns becoming 01 or 11. The third term, $P_{00}\delta_a^{+}\delta_b^{+}$, is a correction term necessitated due to the fact that both first and second terms account for the possibility of a 00 pattern converting to 11.

Similarly, the increase in P_{00} is caused by one the following three events $01 \rightarrow 00$, $10 \rightarrow 00$, $11 \rightarrow 00$. The first, second and third terms in Δ_{00}^{+} account for these three possibilities respectively.

Starting from the control point location, probability changes are computed utilizing the equations illustrated in Table 3, in a levelized, selective-trace manner. For each gate that is effected new signal probability is obtained from appropriate modified pattern probabilities. Fanouts of the effected gate are scheduled for evaluation only if the new signal probability deviates from the original signal probability by more than a pre-specified threshold. The signal probability change along with the threshold LTh and HTh are then used to compute the estimates.

The rationale behind the computation of estimates is as follows. The nodes that are suitable for inserting a control point are those that have extreme probability of a 0 or a 1, since in general, such extreme probability propagates through several levels of a circuit, rendering portions of the circuit non-switching. Such non-switching signals impact both excitation and propagation probabilities of faults. Hence the estimation procedure fixes failing nodes to a particular value, and computes the consequent changes in signal probability. Based on these changes, propagation profile information and pre-specified thresholds, the number of faults that could potentially be detected is determined. This procedure thus quickly eliminates a vast majority of nodes that do not impact the faults from being considered by the subsequent more rigorous evaluation step.

4.2.2 Candidate Signals Evaluation

The estimates of candidate signals computed in the prior step are not precise enough for the selection to be based on them. Hence a more refined analysis of candidate signals is performed. For each candidate node, a 0 or 1 value, depending on whether an AND or an OR type control point is being simulated, is temporarily injected as an increment or a decrement and the resulting change in the propagation profile is determined as explained below.

Insertion of control point at a node perturbs the list of faults associated with itself or nodes in its fanout cone. Therefore it is necessary to recalculate the list for this set only. Accordingly, an incremental probabilistic fault simulation, using modified probabilities, is performed to determine the new list of faults at various nodes in the fanout cone of the candidate signal. Starting from the control point location, new fault lists are computed utilizing the equations given in Table 2, in a levelized, event driven fashion. For each gate that is effected new fault list is obtained from modified lists associated with its inputs. The necessary new underlying probabilities are obtained by the analytical technique for pattern probability change computation presented in the previous section.

The rank of a candidate is defined as the number of additional faults that propagate to primary outputs or observation points. The candidate with highest rank is then selected for control point insertion.

5. Experimental Results

The proposed algorithms have been implemented as a prototype tool comprising flat probabilistic fault simulation, greedy observation point selection, and iterative control point selection. Experiments are then conducted on several highly random pattern resistant ISCAS85 benchmarks [3] and full-scan versions of ISCAS89 benchmarks [4].

For each benchmark circuit, redundant faults are first eliminated using an efficient ATPG tool. Test point insertion is then performed with a pre-specified number of control and observation points. The objective of the test point insertion is to achieve complete or near-complete fault coverage with 32000 random patterns, and minimum number of phases.

Before proceeding with the insertion of test points, the number of phases N, and the design parameters of each phase, i.e., the duration and the number of control and observation points in each phase, have to be determined. We have chosen N to be an arbitrary value, and evenly distributed the entire test length among these phases. Fault simulation for phase Φ_0 is then performed using a fast fault simulator. The set of undetected faults at the end of Φ_0 is then used to insert observation points. No control points are inserted during Φ_0 in order to preserve the initial fault coverage. The design of each of the subsequent phases Φ_1 to $\Phi_{(N-1)}$ is then carried out iteratively. In each phase a fixed number of control points targeting the set of undetected faults inherited from the previous phase, have been inserted with the knowledge of already inserted test points. All observation points are inserted during Φ_0 and the total number of control points are distributed equally among the phases Φ_1 to $\Phi_{(N-1)}$.

The selection of test points within a particular phase is controlled by a set of parameters such as the detection probability threshold DTh, the minimum-benefit-per-cost MBPC, and the implementation cost factors of control and observation points. The cost factors reflect the relative complexity of implementing a particular type of test point and provide the user with a means to obtain a solution with the preferred type of test points. A detection probability threshold of $4/D_{\Phi I}$ where $D_{\Phi I}$ is the duration of phase Φ_I, and a minimum-benefit-per-cost of 10 has been used. In addition, the cost of realizing a control point is assumed to be one unit, and that of an observation point to be 0.5. When multiple test phases are used, it is possible to re-enable a control point in phases subsequent to the phase in which it is selected. A cost factor of 0.25 is assigned for the re-enabled control points.

Since the objective is to obtain complete or near-complete fault coverage with a minimum number of phases, first a 2-phase experiment is performed for all the circuits. Table 4 presents the results obtained for this experiment. In this table, fault simulation results of the original circuit without any modifications are given under the column Original-Circuit, while those of the 2-phase experiment are given under the column 2-phase. In order to reduce any random effect, the average and maximum fault coverage of ten fault simulation experiments are presented. In addition, for the 2-phase experiment the number of control and observation points inserted in each circuit are reported.

The following important observations can be made from the 2-phase experiment results. First, for all the circuits an average fault coverage of greater than 99% is achieved with the insertion of very few test points. For circuits c2670, s3330 and s3384 complete coverage of stuck-at faults is obtained. This is significant considering the simplicity of control point implementation in the 2-phase scheme. All control points are essentially disabled during $\Phi 0$ and enabled during $\Phi 1$ by a single phase decoder output.

Second, it is interesting to note that for circuits c2670 and c7552, the proposed method achieves complete fault coverage with fewer or same number of test points compared to the best reported results in [19]. For the rest of the benchmark circuits, to the best of our knowledge, results for their full-scan versions are being reported for the first time in the open literature. The proposed method achieves complete coverage for c2670 with one control point and five observation points compared to three control points and 7 observation points reported in [19]. This suggests that it is indeed not only feasible to use fixed values for control points, but it might be better than equi-probable control points. However, as is evident from the results of c7552, it may be necessary to increase the number of phases in order to achieve complete fault coverage. For this circuit complete fault coverage is obtained with 6 phases and with the number of control/observation points same as that reported in [19].

Hence, for circuits for which complete coverage is not achieved with the 2-phase scheme, multi-phase experiments are conducted. The results for this experiment are presented in under the column multi-phase in Table 4, where the number of phases are reported in addition to the

number of control/observation points, average and maximum fault coverage of ten fault simulation trials.

From the multi-phase results it can be observed that the number of phases that are necessary to achieve complete/ near-complete fault coverage is small, with a maximum of 6 for c7552. This confirms the premise that only few combinations of values are necessary at the control points. In fact for the circuit c7552, the number of different combinations that occur at the 18 control points is only 6, compared to the maximum possible 2^{18} combinations when equiprobable values are used. In addition, the insertion of very few control/observation points resulted in complete or near-complete fault coverage. This is due to the fact that conflicting control points are enabled in different phases, reducing the number of futile control points. Figure 3, illustrates the fault coverage plot of the c7552 for the 6-phase experiment. This plot clearly demonstrates the necessity as well as the constructive nature of the multi-phase solution.

Although in most cases it is possible to obtain complete fault coverage using the multi-phase test point insertion, for some circuits it may be necessary to insert too many test points in order to detect the final few hard-to-test faults. This is the result of these faults lying distributed in the circuit. At this stage, an alternative deterministic way of detecting the remaining faults, using reseeding seems to provide an attractive trade-off [14]. In this technique, first ATPG is used to generate test cubes for the final hard faults. These cubes are then compressed and stored as seeds of an LFSR in order to reduce the memory requirement. Decompressing these seeds generates test vectors that match the test cubes in all specified positions. Efficient compression/ decompression techniques are presented in [11] and [26]. For s15850 and s38417 a smart ATPG tool is used to generate deterministic test cubes for the remaining undetected faults. The number of cubes and hence the number of seeds necessary to achieve 100% fault coverage for circuits s15850 and s38417 is 8 and 10 respectively.

6. Conclusions

In this paper a novel, constructive test point insertion technique for scan-based BIST has been presented. A divide and conquer technique is used to partition the entire test into multiple phases. With in each phase a group of control and observation points are inserted targeting the set of undetected faults. A new technique called probabilistic fault simulation is used as a vehicle to select the control/ observation points. Experimental results indicate that it is possible to achieve complete or near-complete fault coverage by using the simple 2-phase scheme with the insertion of very few test points. This is a significant result, since control points in this scheme are very simple and are enabled by true or complemented form of a single signal. In addition it is shown that fault coverage achieved can be further enhanced by adopting multi-phase scheme. Experimental results indicate that the number of phases needed is small and demonstrate the constructive nature of these phases. For the circuits for which complete coverage has not been achieved, test cubes are generated to detect the remaining undetected faults. It is shown that very few cubes are necessary to detect these faults.

Acknowledgment

We would like to acknowledge the fruitful discussions we had with John Waicukauski of ATTI and the interaction we had with Pieter Trouborst of Nortel.

References

[1] P.H. Bardell, W.H. McKenney, and J. Savir,"Built-In Test for VLSI: Pseudorandom Techniques", *John Wiley and Sons*, New York, 1987.

[2] F. Brglez,"On Testability Analysis of Combinational Networks", *Proc. of Int'l. Symp. on Circuits and Systems*, pp. 221-225, 1984.

[3] F. Brglez, and H. Fujiwara,"A Neural Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran", *Proc. of Int'l. Symp. on Circuits and Systems*, pp. 663-698, 1985.

Table 4 Test point insertion results

Circuit Name	Original-Circuit (32k random patterns)		2-phase (32k random patterns)			Multi-phase (32k random patterns)			
	Average FC	Max. FC	#CPs/ #OPs	Average FC	Max. FC	#Phases	#CPs/ #OPs	Average FC	Max. FC
c2670	88.12	88.82	1/5	100.0	**100.0**	-	-	-	-
c7552	96.73	97.06	18/2	99.49	99.60	6	18/2	100.0	**100.0**
s3330	87.95	88.99	1/4	99.97	**100.0**	-	-	-	-
s3384	96.06	96.30	0/5	100.0	**100.0**	-	-	-	-
s4863	97.66	98.12	6/4	99.78	99.83	4	6/4	99.96	**100.0**
s9234	93.13	93.79	8/10	99.80	99.85	3	8/10	99.97	**100.0**
s15850	95.72	95.99	15/17	99.16	99.21	4	15/17	99.67	**99.71**
s38417	95.40	95.64	18/30	99.79	99.82	3	18/30	99.81	**99.85**

[4] F. Brglez, D. Bryan, and K. Kozminski,"Combinational Profiles of Sequential Benchmark Cicuits", *Proc. of Int'l. Symp. on Circuits and Systems*, pp. 1929-1934, 1989.

[5] A.J. Briers, and K.A.E. Totton,"Random Pattern Testability by Fast Fault Simulation", *Proc. of Int'l Test Conf.*, pp. 274-281, 1986.

[6] M. Chatterjee, and D.K. Pradhan,"A Novel Pattern Generator for Near-Perfect Fault-Coverage", *Proc. of VLSI Test Symp.*, pp. 417-425, 1995.

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

[8] J.R. Fox,"Test-Point Condensation in the Diagnosis of Digital Circuits", *Proc. of IEE*, Vol. 124, pp. 89-94, Feb. 1977.

[9] M.R. Garey and D.S. Johnson,"Computers and Intractability: A Guide to the Theory of NP-Completeness", *Freeman & Co.*, San Francisco, 1979.

[10] J.P. Hayes and A.D. Friedman,"Test point placement to simplify fault detection", *IEEE Trans. Comp.*, Vol. C-23, No. 7, pp. 727-735, 1974.

[11] S. Hellebrand, J. Rajski, S. Tarnick, S. Venkataraman, and B. Courtois,"Built-In Test for Circuits with Scan Based on Reseeding of Multiple-Polynomial Linear Feedback Shift Registers", *IEEE Trans. on Comp.*, Vol. C-44, pp. 223-233, Feb. 1995.

[12] V.S. Iyengar and D. Brand,"Synthesis of Pseudo-Random Pattern Testable Designs", *Proc. of Int'l Test Conf.*, pp. 501-508, 1989.

[13] S.K. Jain and V.D. Agrawal,"Statistical Fault Analysis", *IEEE Design & Test of Comp.*, Vol. 2, No. 1, pp. 38-44, 1985.

[14] B. Koenemann,"LFSR-Coded Test Patterns for Scan Designs", *Proc. of Euro. Test Conf.*, pp. 237-242, 1991.

[15] B. Krishnamurthy,"A Dynamic Programming Approach to the Test Point Insertion Problem", *Proc. of Design Auto. Conf.*, pp. 695-705, 1987.

[16] S. Pateras, and J. Rajski,"Cube-Contained Random Patterns and their Application to the Complete Testing of Synthesized Multi-Level Circuits", *Proc. of Int'l. Test Conf.*, pp. 473-482, 1991.

[17] E.M. Rudnick, V. Chickermane, and J.H. Patel,"An Observability Enhancement Approach for Improved Testability and At-Speed Test", *IEEE Trans. on CAD*, Vol. 13, No. 8, pp. 1051-1056, 1994.

[18] Y. Savaria, M. Yousef, B. Kaminska, and M. Koudil,"Automatic Test Point Insertion for Pseudo-Random Testing", *Proc. of Int'l Symp. on Circuits and Systems*, pp. 1960-1963, 1991.

[19] B.H. Seiss, P.M. Trouborst, M.H. Schulz,"Test Point Insertion for Scan-Based BIST", *Proc. of 2nd Euro. Test Conf.*, pp. 253-262, 1991.

[20] S.C. Seth, L. Pan, and V.D. Agrawal,"PREDICT - Probabilistic Estimation of Digital Circuit Testability", *Proc. of Fault-Tolerant Computing Symp.*, pp. 220-225, 1985.

[21] N. Tamarapalli, and J. Rajski,"Multi-Phase Test Point Insertion for Integrated Circuits", *US patent pending.*

[22] N.A. Touba, and E.J. McCluskey,"Transformed Pseudo-Random Patterns for BIST", *Proc. of VLSI Test Symp.*, pp. 410-416, 1995.

[23] J.A. Waicukauski, E.B. Eichelberger, D.O. Forlenza, E. Lindbloom, and T.M. McCarthy,"A Statistical Calculation of Fault Detection Probabilities by Fast Fault Simulation", *Proc. of Int'l Test Conf.*, pp. 779-784, 1985.

[24] H.-J. Wunderlich,"PROTEST: A Tool for Probabilistic Testability Analysis", *Proc. of Design Auto. Conf.*, pp. 204-211, 1985.

[25] H.-J. Wunderlich,"Multiple Distributions for Biased Random Test Patterns", *Proc. Int'l Test Conf.*, pp. 236-244, 1988.

[26] N. Zacharia, J. Rajski, J. Tyszer, and J.A. Waicukauski, "Two-Dimentional Test Data Decompressor for Multiple Scan Designs", *in this proceedings*.

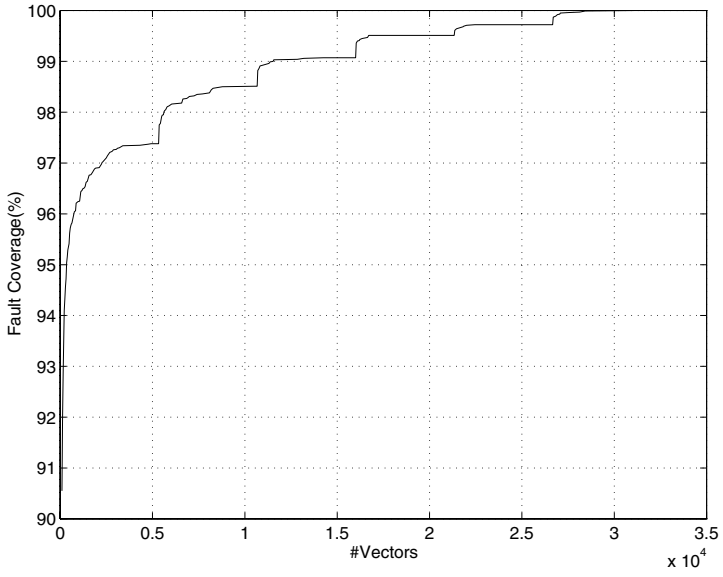


Fig. 3 Fault coverage plot of c7552 with 6 test phases