LOW OVERHEAD TEST POINT INSERTION FOR SCAN-BASED BIST

Michinobu Nakao*, Seiji Kobayashi*, Kazumi Hatayama*, Kazuhiko Iijima**, and Seiji Terada***

* Hitachi Research Laboratory, Hitachi, Ltd., Hitachi-shi, 319-1292 Japan
** Enterprise Server Division, Hitachi, Ltd., Hadano-shi, 259-1392 Japan
*** Hitachi Business Solution, Ltd., Yokohama-shi, 231-0015 Japan

Abstract
This paper presents a practical test point insertion method for scan-based BIST. To apply test point insertion in actual LSIs, especially high performance LSIs, it is important to reduce the delay penalty and the area overhead of the inserted test points. Here, efficient test point selection algorithms, which are suitable for utilizing overhead reduction approaches such as restricted cell replacement, test point flip-flops sharing, are proposed to meet the above requirements. The effectiveness of the algorithms is demonstrated by some experiments.

1. Introduction
Increasing gate count of logic LSIs is causing many difficulties in designing a high quality test. One of the difficulties is the increase of test data size (or test pattern length). The built-in self-test (BIST) [1] can solve this problem. However, the BIST requires a very large number of patterns to achieve high fault coverage due to the nature of pseudo-random pattern test. The test point Insertion (TPI) [2]-[11] method has been introduced to overcome this weak point. This method improves random pattern testability efficiently by inserting control points and observation points, and so it is effective in the reduction of pattern length and the improvement of fault coverage.

Recently, several approaches for test point selection have been proposed: a method using fault simulation [2],[3]; a method based on cost minimization reflecting random pattern testability [4]; and a method based on path tracing [5]. Furthermore, the cost minimization approach has been enhanced to reduce computational complexity allowing large-scale circuits to be handled [6][7].

The TPI method has two problems for practical application. One is the problem of delay penalty. From the viewpoint of performance, it is hardly acceptable to insert an AND or an OR gate as a control point on timing critical paths, because test points cause additional path delays. The other problem is area overhead due to circuits and wires being inserted for test points. The area overhead should be reduced as much as possible from the viewpoint of manufacturing cost. Both problems have a trade-off with the fault coverage by the BIST. Several studies have considered these problems and proposed solutions: a timing-driven method that avoids inserting test points on critical paths by computing delay information in test point selection [8]; a method to reduce delay and area overheads by improving test point circuits and sharing some test points [9]; and a method to reduce the number of test points by enhancing fault coverage by switching the effective test points in each test phase [10].

Here we focus on circuits designed by a full-scan based BIST scheme, and we set the following policies for TPI from the viewpoint of practical use.
(1) Reduce the delay penalty to a negligible level.
(2) Restrict the area overhead for test points.
(3) Do not increase the number of LSI pins.
(4) Restrict the random pattern length in BIST.

The importance of policies (1) and (2) has been stated before. Policy (3) reflects a limitation on the number of LSI pins. Therefore we principally insert a scan flip-flop, which we call test point flip-flop (TP-FF), for a control point or an observation point. Policy (4) originates from a limitation of the testing time. Our goal is to maximize the fault coverage for BIST by TPI, while keeping the four policies.

Therefore we adopt two approaches to low overhead TPI: restricted cell replacement and TP-FF sharing. The restricted cell replacement is an approach to reduce delay penalty, where each control point is implemented by replacing a cell of restricted types to a certain cell. The TP-FF sharing is introduced to reduce area overhead,
where a TP-FF is shared for some test points under some conditions.

Moreover, new test point selection algorithms are proposed in which the key points are an optimization based on the cost function reflecting the fault coverage, an accelerated iterative improvement and considering the TP-FF sharing.

The rest of the paper is organized as follows. In Section 2, an overview of TPI for the scan-based BIST is given and the background to test point selection based on random pattern testability is outlined. In Section 3, the restricted cell replacement approach and the TP-FF sharing approach are described. The new test point selection algorithms are presented in Section 4. Finally some experimental results are shown to demonstrate the effectiveness of our approach in Section 5.

2. Background and Definitions

We suppose a circuit under test is full-scan designed. Fig. 1 shows the structure of our scan-based BIST. The flip-flops in the figure include scan flip-flops for normal function (FF) and TP-FFs. Two kinds of tests can be executed with this structure. One is a pseudo-random test using the LFSR and the MISR as a pattern generator and a signature analyzer, respectively. The other is an ordinary scan test, which can be used for detecting random pattern resistant faults.

The TPI method is a technique that improves testability for a circuit under test by inserting test points. Example test points in a circuit and their effects are illustrated in Fig. 2. Fig. 2(a) shows a 0-control point which improves 0-controllability for signal A. The control point inserts an AND gate (an OR gate in the case of a 1-control point) and a TP-FF. So as not to change the original logic in the normal operation mode, the signal from the TP-FF to the OR/AND gate has to be set to the non-controlling value, that is 1 for the AND gate and 0 for the OR gate. The control point improves the controllability of the region where the inserted signal propagates toward outputs (Ctl-up region), and changes the observability of the region where the signals changing controllability propagate toward inputs (Obs-chg region). Fig. 2(b) shows the observation point which improves observability for signal B. The observation point inserts a TP-FF. The observation point improves the observability of the region where the inserted signal propagates toward inputs (Obs-up region).

We can define relationships between two test points. The relationship between control points is called a strong one if their Ctl-up regions overlap, a weak one if their Ctl-up regions do not overlap but their Obs-chg regions do, and an independent one if their Obs-chg regions are distinct from each other. The relationship between observation points is called a strong one if their Obs-up regions overlap, and an independent one if their Obs-up regions are distinct from each other. No weak relationship is defined in the case of the observation points.

Like the approach of [4], our approach is based on the well-known probabilistic random pattern testability measure called COP (Controllability Observability Procedure) [12]. \( C_i \) represents 1-controllability for signal \( i \), that is the probability for which signal \( i \) has a logic value 1. \( O_i \) represents observability for signal \( i \), that is the probability for which a logic value at signal \( i \) can be observed in at least one of the primary outputs or the scan flip-flops. Table 1 shows examples of the COP calculation rules. For simplicity the measure of COP is based on the calculation rule of the probability for the circuit without reconvergent fanout. \( P_{ii} \) represents the probability of detecting the stuck-at-0 fault at signal \( i \) by a random pattern, and it is defined as

\[
P_{ii} = C_i(s) O_i,
\]

where \( C_i(0) = C_i, C_i(1) = 1 - C_i \).

Then the cost function \( V \) in [13] is defined as

\[
V = \frac{1}{|E|} \sum_{f=0}^{1} \frac{1}{P_f},
\]
Table 1 COP calculation rules

<table>
<thead>
<tr>
<th>k = NOT(i)</th>
<th>C_k</th>
<th>O_k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - C_i</td>
<td></td>
<td>O_i</td>
</tr>
</tbody>
</table>

where F represents the set of target faults. The cost function V can be viewed as the average expected pattern length of all the faults in F for random pattern tests because 1/P_{int} is the expected test length for the stuck-at-fault at signal i.

In [4], a test point selection algorithm based on minimizing of the cost function V was proposed, and it was reported that the test points selected by this algorithm improve the fault coverage for random pattern tests greatly. In the algorithm, selection of the test point is repeated such that the cost V after insertion is a minimum until the given number of test points or the desired cost is achieved. Since calculation of the cost V after inserting test point candidates for every test point selection wastes CPU time, the CRF (Cost Reduction Factor) was introduced to approximate the reduction of V due to insertion of a test point candidate. The test point candidate with a large CRF value is regarded as one of the good candidates in terms of the cost reduction.

3. Approaches to Low Overhead TPI

3.1 Restricted cell replacement approach

In order to reduce the delay overhead by TPI, we utilize restricted cell replacement approach.

Implementation of control points by replacing an original cell by a corresponding cell has been presented in [9][11]. The increase of gate delay by the cell replacement can be smaller than by the insertion of an AND/OR cell. Fig. 3(b) is an example of a 1-control point at the output of an inverter cell, which is implemented by replacing the inverter cell with a NAND cell and adding a TP-FF for which output connects to the NAND cell. Fig. 3(b) is an example of a 0-control point at the output of an inverter cell, which is implemented by replacing the inverter cell with a NOR cell in the same way as 1-control point.

The key of our approach is that replaceable cell types are restricted to amplifiers with small drivability. The corresponding cell may be some special cells only for control points in order to have the almost same gate delay and drivability as the replaceable cell. Then, the delay penalty by control points is expected to be reduced to a negligible level in our approach if we assume that wiring delays are more dominant than gate delays. This is because the influence on the wiring delay of the normal path through a control point is small if the drivability of the replaceable cell is small, and amplifiers with small drivability tend to avoid being used on the critical paths. Furthermore, our approach ensures capability for fault coverage improvement since amplifiers are frequently used in whole circuit.

On the other hand, the insertion of an observation point has little influence on the delay of the path in the normal logic of an amplifier is placed soon after the fanout of the observation point. Therefore, positions for inserting an observation point are not restricted.

3.2 TP-FF sharing approach

In order to reduce the area overhead by TPI, we utilize techniques for sharing a TP-FF for plural test points.

The sharing of test points has been discussed in [9][11]. Fig. 4 shows examples of TP-FF sharing we adopt. The TP-FF sharing for control points is implemented by distributing the output of one TP-FF to control inputs of plural control points as shown in Figs. 4(a)-(c). The TP-FF sharing for observation points is implemented by connecting the signal lines from the fanouts in the observation points to one TP-FF through a combinational circuit with multiple inputs and one output, which is called a compactor, as shown in Fig. 4(d).

The TP-FF sharing for control points in the strong relationship is shown in Figs. 4(a), (b). Fig. 4(a) shows the case where plural control points which have the same sink gate share a TP-FF. The effect of TPI by 'TP-FF shared' is the same as that by inserting 1-control point on the output of the AND gate. Fig. 4(b) shows the case where plural control points which have the same source gate or plural control points which are at the equivalent
signals share a TP-FF. The effect of TPI by 'TP-FF shared' is the same as that by inserting 1-control point on the output of the AND gate. These facts about the effect of TPI must be taken into consideration in test point selection process. We note that there are cases which insert inverters so that a non-controlling value (NCV) for each combinational gate inserted for the control points which share a TP-FF is fed from the TP-FF at the same time. The reason why we adopt these sharing is that degradation of the testability due to sharing is expected to be small, and such circuit structures are expected to appear frequently for control points inserted by the restricted cell replacement. Fig. 4(c) shows the TP-FF sharing for two control points in the independent or in the weak relationship. Testability is not affected by the TP-FF sharing for independent control points. Fig. 4(d) shows the TP-FF sharing for two observation points in the independent relationship.

A compactor must be inserted in sharing the TP-FFs for observation points. In order to use the smallest size compactor possible with little loss of testability, compactor selection is decided according to the controllability at each observation point. Fig. 5 summarizes usage priority of compactors and selection rules. The priority is decided by considering the compactor size. In Fig. 5, the number x is a threshold for deciding the compactor and it is given a small value (e.g. 0.1). The observability after sharing is also given to show that the TP-FF sharing has little effect on testability. For example, when a NAND gate is used as a compactor, the observability at the observation point after the TP-FF sharing is equal to the controllability which becomes NCV (1 in this case) at the other observation point, that is, \( O_a = C_x \) and \( O_b = C_x \). The explanation is similar for a NOR and a NAND-INV compactor. For the case of the EOR compactor, the testability of the circuit is not affected at all by the TP-FF sharing. This technique can also be extended to three or more observation points, but sharing a TP-FF for more observation points means a further decrease in observability except for the EOR compactor.

The process for deciding TP-FF sharing of independent observation points is given as follows. We find a group of independent observation points so that

(d1) the 1-controllability at each observation point is greater than \(1-x\), and a TP-FF is shared by using a NAND compactor;

(d2) the 1-controllability at each observation point is less than \(x\), and a TP-FF is shared by using a NOR compactor;

### Table 1: Compactor and Selection Rules

<table>
<thead>
<tr>
<th>Priority</th>
<th>Compactor</th>
<th>Controllability condition</th>
<th>Observability after sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A-B -B</td>
<td>( C_A &gt; 1 - x ) ( C_B &gt; 1 - x )</td>
<td>( O_a = C_B &gt; 1 - x ) ( O_B = C_A &gt; 1 - x )</td>
</tr>
<tr>
<td>2</td>
<td>A-B -B</td>
<td>( C_A &lt; x ) ( C_B &lt; x )</td>
<td>( O_a = 1 - C_B &gt; 1 - x ) ( O_B = 1 - C_A &gt; 1 - x )</td>
</tr>
<tr>
<td>3</td>
<td>A-B -B</td>
<td>( C_A &gt; 1 - x ) ( C_B &gt; x )</td>
<td>( O_a = 1 - C_B &gt; 1 - x ) ( O_B = C_A &gt; 1 - x )</td>
</tr>
<tr>
<td>4</td>
<td>A-B -B</td>
<td>EOR</td>
<td>( O_a = 1 ) ( O_b = 1 )</td>
</tr>
</tbody>
</table>

Fig. 5 Compactors and selection rules.
(d3) the 1-controllability at each observation point is greater than 1 - x or less than x, and a TP-FF is shared by using a NAND-INV compactor; and

(d4) the 1-controllability of at least one observation point is greater than x and less than 1 - x, and TP-FFs are shared by using an EOR compactor.

The key to this process is to select the compactor according to the controllability at each observation point. We note that to limit any adverse effect on TPI by sharing a TP-FF in (d1)-(d3), it is necessary to limit the number of the observation points in the sharing group and to set the threshold x to be properly small.

For further reduction of area overhead for TP-FFs, it is possible to share TP-FFs with a scan flip-flop for normal function, which is called an original flip-flop (O-FF). The case of sharing independent control points with a primary input was presented in [9]. Fig. 6 shows how to share the TP-FFs for observation points with an O-FF. Plural observation points in the independent relationship are shared with the O-FF as the primary output, if the fan-in cone of the O-FF does not intersect the Obs-up region of each observation point. The compacted signal value of the observation points and the data input of the O-FF are merged by the compactor such as an EOR gate, and the output of the compactor fed to the O-FF. In the normal mode, signal values of the observation points must be masked using the test mode signal, so that it is not propagated to the O-FF. We can say that the effect of TPI is retained even if TP-FFs are shared with the O-FF. But we do not adopt sharing TP-FFs with an O-FF because there are two problems: the overhead for wiring the test mode signal and the delay penalty of normal paths.

4. Test Point Selection Algorithms

New test point selection algorithms for low overhead TPI are proposed. The algorithms are suitable for utilizing the overhead reduction approaches described in Section 3 because fault coverage is maximized when signals to insert a test point, the number of TP-FFs and the random pattern length are limited. In the following, first we formalize the test point selection problem in an actual situation and discuss its treatment by a conventional approach. Then, an optimization technique based on the cost reflecting the fault coverage and an accelerated iterative improvement technique are proposed. Also, a test point selection algorithm considering the TP-FF sharing technique is presented.

4.1 Test point selection under an actual situation

In an actual situation, the following restrictions are set to reduce the delay penalty, the area overhead and the testing time.

(1) There are some regions and lines where TPI is prohibited.

(2) The number of test points or the number of TP-FFs is bounded.

(3) The number of random patterns is predetermined.

Our purpose is to select the set of the test points which achieve as high fault coverage as possible under the given restrictions.

There are two problems in applying the algorithm proposed in [4] which was outlined in Section 2. One is validity of the cost function V. Minimization of V is based on the position that the number of random patterns for complete fault coverage should be minimized and fault coverage is not always improved at the specific number of patterns, since detection probability P, with a minimum amount often influences V and the detection probabilities for the majority of faults are not always reflected. In particular, if the position for test point insertion is restricted and the minimum detection probability can not be improved, differences between the test points to decrease V and the test points to improve the fault coverage become remarkable. The second problem is the optimization capability of this algorithm. This algorithm is a kind of greedy technique where the optimal element is selected in each selection process. The set of resulting test points is not always the optimal solution though the selection of each test point must have been the best under the temporary circuit condition. Therefore an iterative improvement technique is generally used for combinational optimization problems in order to find a optimal or nearly optimal solution.
We present a new test point selection algorithm which can solve the above problem in the following. Its main ideas are an optimization technique based on the cost reflecting the fault coverage and an accelerated iterative improvement technique.

4.2 Optimization based on expected fault coverage

We define a new cost function which reflects the fault coverage of the random pattern test, and we explain the technique to optimize it.

The set of target faults is represented by \( F \), the detection probability of a fault \( f \) is represented by \( P_f \), and the number of the patterns is represented by \( t \). Then the expected value of the fault coverage \( EFC(t) \) which is called as the expected fault coverage, is given as follows:

\[
EFC(t) = \frac{1}{|F|} \sum_{f \in F} \left[ 1 - (1 - P_f)^t \right].
\]

Then a new cost function is defined as follows:

\[
U(t) = \sum_{f \in F} (1 - P_f)^t.
\]

This function \( U(t) \) means the expected value of the number of undetected faults in \( t \) random patterns. Minimization of \( U(t) \) is equivalent to maximization of the expected fault coverage. For calculation efficiency, an enhanced CRF is introduced as the approximation of the reduction of \( U(t) \) by inserting a test point. The enhanced CRFs for a 1-control point, a 0-control point, and an observation point on signal line \( a \), which are represented by \( CRF_{1CON}^o, CRF_{0CON}^o \), and \( CRF_{OBS}^o \), respectively, are defined as follows:

\[
CRF_{1CON}^o = \frac{2 - C_a}{C_a + 1} \frac{C_a - 1}{2} \frac{\partial U(t)}{\partial C_a},
\]

\[
CRF_{0CON}^o = \frac{2 - 2 \cdot C_a}{2 - C_a} \frac{C_a}{2} \frac{\partial U(t)}{\partial C_a},
\]

\[
CRF_{OBS}^o = O_a \frac{\partial U(t)}{\partial O_a},
\]

where \( k \) in the equations is a parameter such that \( 0 \leq k \leq 1 \). \( \frac{\partial U(t)}{\partial C_a} \) and \( \frac{\partial U(t)}{\partial O_a} \) represent gradients of the function \( U(t) \) for 1-controllability and observability at signal line \( a \), respectively. The gradients of all signals can be computed in linear time order of gate counts in the same way as described in [13].

\[
\frac{\partial U(t)}{\partial O_a} = -t \sum_{a \in \Sigma} (1 - P_{a/s})^{-1} \cdot C_a(s)
\]

\[
\frac{\partial U(t)}{\partial C_a} = -t \sum_{a \in \Sigma} (1 - P_{a/s})^{-1} \cdot C_a(s)
\]

The enhanced CRF for \( k=0 \) means the first degree term in the Taylor expansion of the actual cost reduction (ACR) because it is a product of the gradient and the change of controllability or observability. The enhanced CRF for \( k=1 \) corresponds to the CRF equations described in [4], except for the shape of the cost function. This parameter \( k \) is introduced because the enhanced CRF for each case of \( k=0 \) and \( k=1 \) can be as a poor approximation of ACR for some circuits. In our test point selection process, the parameter \( k \) takes five values, 0, 1/4, 1/2, 3/4 and 1, in turn. Furthermore, in order to cover a few efficient test points missed by using the enhanced CRF, the usual CRF for the cost function \( V \) is used together as a guide for test point selection.

4.3 Optimization with accelerated iterative improvement

In this subsection, the test point selection problem under restriction in the number of test points is formalized as follows:

Given the number of random patterns \( t \) and the number of test points \( N_o \), we select the set of test points to maximize the expected fault coverage \( EFC(t) \), that is, to minimize the cost function \( U(t) \).

Our proposed algorithm derives a local optimum solution for this problem by introducing the iterative improvement technique (the local search technique) which searches in the neighborhood of the current solution and improves the quality of the solution iteratively. First, the usual test point selection process computing the set of \( N_o \) test points is executed for generating an initial solution. Next, the quality of the solution \( U(t) \) is improved by replacing a part of the current set of test points by the new test points. This test point exchange process is repeated until the current solution becomes a local optimum solution which is the best solution in the neighborhood of the current solution.

In order to introduce the iterative improvement, two procedures, a procedure that selects and removes some test points with a small contribution for minimization of the cost function, and a procedure that selects new test points which can reduce the cost function the most, are required.
For the latter, the conventional test point selection procedure can be appropriated and we omit the details. Therefore only the former is described below.

As a guide to the contribution for each test point in the procedure removing test points, an actual cost addition (ACA), which is the addition in the function $U(t)$ when only the target test point is removed from the set of test points, is computed. The test point with a large ACA in the set of test points can be said to contribute to the cost function minimization. Therefore, the target of the removal can be the test point with the smallest ACA.

There is a problem that the processing time of the test point selection increases though the quality of the solution is improved when the iterative improvement technique is introduced. In order to accelerate the process without losing the capability of the optimization, some independent test points are removed simultaneously. As for the test point selection procedure, the simultaneous selection of plural independent test points for acceleration has been proposed in [7]. The removal of plural test points can be executed for a test point which has smaller ACA than a certain threshold, (a) by selecting independent test points in ascending order of ACA and (b) removing these test points simultaneously.

Algorithm 1 shows the enhanced test point selection algorithm including the iterative improvement. Given the number of test points ($N_w$), the number of random patterns ($t$) and the initial number of simultaneously exchanged test points in the iterative improvement ($N_s$), the algorithm first generates an initial solution and then improves the solution based on the iterative improvement. In the iterative improvement procedure, first $n$, the maximum number of simultaneously exchanged test points, is set to $N_w$, then at most $n$ test points are removed and replaced by new test points. If there are some test points included both in the set of removed test points and the set of inserted test points, $n$ is decreased by the number of such test points. This procedure is repeated until $n$ becomes 0, in other words until it is judged that a local optimal solution is derived.

4.4 A test point selection in consideration of the TP-FF sharing

The area overhead by TPI is expected to be almost proportional to the number of TP-FFs. So, we consider the following problem:

```plaintext
Enhanced_Test_Point_Selection (N_w, t, N_s) |
/* n : Number of simultaneous exchanges */
/* S, S_r, S_i : Set of test points */
/* #S, #S_r, #S_i : Number of test points */
/* CUT : Current circuit under test */
S = f;
while (#S < N_w) |
    S_r=Insert_Test_Points (t);
    S = S ∪ S_r;
}
N = N_w;
while (n > 0) |
    S_r=Remove_Test_Points (t, S, n);
    S = S - S_r;
    while (#S_i < #S_r) |
        S_i=Insert_Test_Points (t);
        S = S ∪ S_i;
    }
    n = n - (#S_i ∩ S_r);
}
return S; /* Enhanced_Test_Point_Selection */

Insert_Test_Points (t) |
Compute U(t) and CRFs for each signal;
Compute ACAs for test points with large CRF;
Select independent test points with large ACA;
Insert the selected test points to CUT;
return (the selected test points);
/* Insert_Test_Points */

Remove_Test_Points (t, S, n) |
Compute U(t) and ACAs for test points in S;
Select n independent test points with large ACA;
Remove the selected test points to CUT;
return (the selected test points);
/* Remove_Test_Points */
```

Algorithm 1. Enhanced test point selection

Given the number of random patterns $t$ and the number of TP-FFs $N_{FF}$, we select the set of the test points to maximize the expected fault coverage $EFC(t)$ in consideration of the TP-FF sharing.

In our approach, the number of test points $N_w$ in the test point selection process is set to the expected number from the given number of TP-FFs, and the solution is generated by selecting efficient test points from them based on the priority of each test point and sharing TP-FFs. This procedure prevents test points with a small effect from being inserted, so that the area overhead by TPI can be reduced.
Test_Point_Selection_with_Sharing ( N_p, t, N_w, mfc, R )
{
    /* N_p : Number of test points */
    /* S, S', S' ; Set of test points */
    S = f;
    N_p = R * N_p;
    S = Enhanced_Test_Point_Selection ( N_p, t, N_w );
    Compute ACAs for test points in S ;
    while ( #S < N_p and mfc < Fc(t) )
    {
        Select tp as test point with the largest ACA in S ;
        Compute S_p as test points group including tp
        based on ACAs and TP-FF sharing in Figs. 4,5 ;
        S = S ∪ S_p ;
        S = S - S_p ;
    }
    return S ;
} /* Test_Point_Selection_with_Sharing */

Algorithm 2. Test point selection with TP-FF sharing

Algorithm 2 shows the test point selection algorithm in consideration of the TP-FF sharing. The number of TP-FFs (N_p), the number of random patterns (t), the initial number of simultaneously exchanged test points in the iterative improvement (N_w), the desired value of EFC(t) (mfc) and the ratio of the number of test points in test point selection to N_p (R) are given. After the enhanced test point selection process, the ACAs defined in the previous section are computed for all the resulting test points, and the test points are sorted in the descending order of ACAs. In order to optimize the priority of test points, they are sorted again in the descending order of new ACAs, which are computed as reduction of the cost U(t) in inserting the test points sequentially in the current order. A test point with the largest ACA is selected and the type of TP-FF sharing is decided according to selection rules in Fig. 4, 5. The groups of test points shared by a TP-FF are decided in the order based on the priority computed above. This group selection process is repeated until the number of TP-FFs or the desired expected fault coverage is achieved.

5. Experimental results

We have implemented the program based on the proposed algorithms and executed experiments using several full-scan designed industrial circuits. Table 2 shows circuit characteristics. The restricted cell replacement approach is adopted, so that a control point is restricted to the replacement of the cells which are equivalent to an amplifier with small drivability, and an observation point can be inserted into any signal lines.

The parameters for the proposed algorithms are given as follows. The number of the random patterns t is 255k. The maximum number of the simultaneous exchanges N_w is 15. The pseudo-random patterns in the BIST is generated by the LFSR and the fault coverage is computed by a fault simulator. All experiments are executed on the HITACHI9000V / VR260-EG workstation.

Table 3 shows the experimental results related to the effect of Algorithm 1 and influence on the fault coverage by adopting the restricted cell replacement approach. The number of the test points N_p is 0.2% of the gates count. The condition for TPI is shown in column 2, where 'No' means there are no test points, 'Free' means that control points or observation points can be inserted in all signal lines, and 'Rest' means the restricted cell replacement approach is adopted. The cost function in the test point selection algorithm is shown in column 3. It consists of the cost reflecting pattern length (Ptn) in the conventional algorithm and the cost reflecting the expected fault coverage (Fc) in the proposed algorithm. Whether the accelerated iterative improvement technique is adopted (Y) or not (N) is shown in column 4. The results for the above condition are shown in columns 5-7, which are the expected fault coverage (EFC), the fault coverage by BIST (FC), and the processing time for test point selection (Time).

These results can be summarized as follows. We considered only the expected fault coverage because it is regard as a good guide of the fault coverage from columns 5-6.

- The restricted cell replacement approach causes a degradation in the expected fault coverage. In particular the results for circuits A and D are remarkable. We consider that some of the efficient control points cannot be inserted because the signal lines for the control point are restricted to 3-5% of the number of all signal lines.
Table 3 Experimental results on Algorithm 1

<table>
<thead>
<tr>
<th>Circ</th>
<th>TPI</th>
<th>Cost</th>
<th>Imp</th>
<th>EFC (%)</th>
<th>FC (%)</th>
<th>Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>No</td>
<td>94.46</td>
<td>94.34</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Free</td>
<td>Ptn</td>
<td>N</td>
<td>99.07</td>
<td>99.22</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fc</td>
<td>N</td>
<td>99.80</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fc</td>
<td>Y</td>
<td>99.84</td>
<td>99.50</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Rest</td>
<td>Ptn</td>
<td>N</td>
<td>96.01</td>
<td>95.64</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fc</td>
<td>N</td>
<td>98.62</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fc</td>
<td>Y</td>
<td>98.85</td>
<td>98.82</td>
<td>6</td>
</tr>
<tr>
<td>B</td>
<td>No</td>
<td>94.18</td>
<td>93.93</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Free</td>
<td>Ptn</td>
<td>N</td>
<td>99.76</td>
<td>99.30</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fc</td>
<td>N</td>
<td>99.77</td>
<td>-</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fc</td>
<td>Y</td>
<td>99.92</td>
<td>99.73</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Rest</td>
<td>Ptn</td>
<td>N</td>
<td>98.50</td>
<td>98.84</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fc</td>
<td>N</td>
<td>99.28</td>
<td>-</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fc</td>
<td>Y</td>
<td>99.44</td>
<td>99.32</td>
<td>6</td>
</tr>
<tr>
<td>C</td>
<td>No</td>
<td>95.18</td>
<td>94.63</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Free</td>
<td>Ptn</td>
<td>N</td>
<td>99.80</td>
<td>99.15</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fc</td>
<td>N</td>
<td>99.84</td>
<td>-</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fc</td>
<td>Y</td>
<td>99.87</td>
<td>99.21</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Rest</td>
<td>Ptn</td>
<td>N</td>
<td>98.11</td>
<td>97.49</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fc</td>
<td>N</td>
<td>98.36</td>
<td>-</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fc</td>
<td>Y</td>
<td>98.55</td>
<td>97.87</td>
<td>10</td>
</tr>
<tr>
<td>D</td>
<td>No</td>
<td>81.91</td>
<td>84.47</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Free</td>
<td>Ptn</td>
<td>N</td>
<td>97.20</td>
<td>97.85</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fc</td>
<td>N</td>
<td>98.73</td>
<td>-</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fc</td>
<td>Y</td>
<td>99.09</td>
<td>99.36</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Rest</td>
<td>Ptn</td>
<td>N</td>
<td>86.52</td>
<td>92.33</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fc</td>
<td>N</td>
<td>93.59</td>
<td>-</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fc</td>
<td>Y</td>
<td>94.45</td>
<td>97.21</td>
<td>13</td>
</tr>
</tbody>
</table>

- The change in the cost largely contributes to the improvement of the expected fault coverage. From the results of circuits A and D, we consider that the optimization using the proposed cost function is more effective than the conventional cost function reflecting pattern length, when signals to insert a test point, the number of test points and random pattern length are restricted.
- The accelerated iterative improvement technique contributes to the improvement of the expected fault coverage, while the processing time for the test point selection increases about 1.5 times using this technique.

Table 4 shows the experimental results on Algorithm 2 and the TP-FF sharing. For each circuit, there are three conditions: two cases that the TP-FF sharing process is executed after Algorithm 1 (where $N_p$ is 0.1% of the gates count and $R$ is 4) in row 'A12'. In the TP-FF sharing for observation points, the threshold $x$ is set to 0.1 and the number of observation points in each sharing group is limited to two. Furthermore, test points to be shared are limited to the same logic block (5-10k gates) considering the wiring overhead. The results for the above condition are shown in columns 3-7, which are the expected fault coverage (EFC), the number of test points, control points, observation points, TP-FFs. Columns 8-14 show the number of TP-FFs which can be eliminated by the TP-FF sharing in Figs. 4, 5, where (a) corresponds to the case of Fig. 4(a), and so on. The summation of these numbers means the total number of test points reduction.

The results can be summarized as follows.
- Algorithm 2 reduces the number of TP-FFs computed in Algorithm 1 drastically with a small degradation of the expected fault coverage, which demonstrates the efficiency of the proposed algorithm. From the results that the expected fault coverage in Algorithm 2 is higher than the TP-FF sharing after Algorithm 1 (where $\#TP/#gate = 0.2\%$) in the circuits B and D, the procedure of selection from test points whose number is more than the given number of test points is efficient.
- The TP-FF reduction rate in using the TP-FF sharing is about 50% on the average, which demonstrates the efficiency of our approach. The TP-FF sharing for control points with the same source gate is most effective for the control point. This is because the cells which can be replaced for a control point often exist for the purpose of amplifier after fanout in the circuits used for these experiments. The TP-FF sharing for the independent control points and the independent observation points are also effective. The sharing of the independent observation points with NAND, NOR, NAND-NOT gates as the compactor is not as effective as the EOR compactor because the limit of one group with two observation points.

Experimental results for the effect on ordering the test points are shown in Fig. 7. The expected fault coverage is plotted according to the number of test points. Cases using the cost function reflecting pattern length (Ptn), the cost function reflecting the expected fault coverage and the iterative improvement technique (Fc+Imp), and sorting in a descending order on ACA (Fc+Imp+Sort) are shown. A high expected fault coverage is achieved in a smaller number of test points by optimizing the ordering of test points. If about 100 test points
are inserted in the case Fc+Imp+Sort, the expected fault coverage achieved is almost the same as that inserting about 500 test points in the case Fc+Imp.

5. Conclusion
In this paper, we presented an approach to reduce delay penalty and area overhead in the TPI method for scan-based BIST. We adopted two overhead reduction approaches, the restricted cell replacement and the TP-FF sharing. New test point selection algorithms which are suitable for the approaches are proposed. The keys of the enhanced test point selection algorithm are an optimization technique based on the cost function reflecting the fault coverage and an accelerated iterative improvement technique. This algorithm is more effective when the signals to insert a test point, the number of test points and the random pattern length are limited. Furthermore, the algorithm combining the test point selection and the TP-FF sharing was also proposed.

The efficiency of the proposed algorithms was evaluated by experimental results. The first algorithm achieved a higher fault coverage than previous approaches in reasonable CPU-time. The second algorithm reduces the number of TP-FFs drastically with a small degradation of fault coverage since efficient test points are selected by optimization of test point priority.

References