Test Point Insertion with Control Points Driven by Existing Functional Flip-Flops

Joon-Sung Yang, *Member*, *IEEE*, Nur A. Touba, *Fellow*, *IEEE*, and
Benoit Nadeau-Dostie, *Senior Member*, *IEEE*

Abstract—This paper presents a novel test point insertion method for pseudorandom built-in self-test (BIST) to reduce the area overhead. The proposed method replaces dedicated flip-flops for driving control points by existing functional flip-flops. For each control point, candidate functional flip-flops are identified by using logic cone analysis that investigates the path inversion parity, logical distance, and reconvergence from each control point. Four types of new control point structures are introduced based on the logic cone analysis results to avoid degrading the testability. Experimental results indicate that the proposed method significantly reduces test point area overhead by replacing the dedicated flip-flops and achieves essentially the same fault coverage as conventional test point implementations using dedicated flip-flops driving the control points.

Index Terms—Dedicated flip-flop, functional flip-flop, logic cone analysis, test point insertion.

1 Introduction

Test cost for complex designs has increased significantly. The amount of test data volume required is growing rapidly. Testers have limited I/O channels and speed, and hence pose a major bottleneck for conventional external testing.

Built-in self-test (BIST) helps to reduce test data bandwidth requirements and test storage requirements by orders of magnitude [1], [13]. It involves the use of on-chip test pattern generation and output response analysis. BIST provides a number of important advantages including the ability to apply a large number of test patterns in a short period of time, minimal tester storage requirements, at-speed testing, application of tests out in the field over the lifetime of the part, and a reusable test solution for embedded cores. The most economical logic BIST techniques are based on pseudorandom pattern testing. One of the attractions to pseudorandom pattern testing is the simple logic structures as a part of circuit under test (CUT) which provide the input stimuli and the circuit response compression. This allows significant compaction of test data. Pseudorandom pattern testing also can achieve high coverage of nonmodeled faults which are not explicitly targeted during deterministic test generation. However, a major challenge is the presence of random-pattern-resistant (r.p.r.) faults which have low detection probabilities and hence may limit the fault coverage that can be achieved with pseudorandom patterns. Various methods have been proposed to overcome the problem. One approach is to modify the pattern generator using methods such as weighted pattern testing [2], [13], [14], [19], [23], [25], pattern mapping [4], [30], [31], bit-fixing [32], bit-flipping [34], and LFSR reseeding [8], [9], [15], [16], [17], [24].

The other approach to make the CUT random pattern testable is to modify it by inserting test points [6]. Test point insertion (TPI) involves adding control and observation points to the CUT. Observation points make a node observable by adding an extra primary output or sampling it in a scan cell. Control points involve ANDing or ORing a node with an activation signal where the activation signal is driven by a dedicated flip-flop which receives pseudorandom values during BIST and is set to a noncontrolling value during normal operation. Additional hardware is needed to form the test points which adds area and performance overhead to a design. Since optimal test point placement is NP-complete [18], a number of TPI methods have been proposed using fault simulation [3], [12] and testability measures [26]. Two general strategies for TPI have been widely studied to overcome the overhead issues: 1) TPI for minimizing performance overhead, and 2) TPI for minimizing area overhead.

To minimize the performance overhead for TPI, Cheng and Lin [5] and Tsai et al. [33] proposed timing driven test point insertion techniques which avoid TPI on critical timing paths. They showed that by avoiding control point insertion on critical timing paths, high fault coverage can be achieved without performance degradation. Reducing the number of test points to minimize the area overhead, TPI techniques like path tracing [32] and multiphase TPI [29] were introduced. Tamarapalli and Rajski [29] partition the entire test into multiple phases by divide and conquer method and control points are activated only during certain phases and deactivated during other phases. This provides greater control over the interaction of the control points with each other which can help reduce the total number of test points required. Nakao et al. [21] and Youssef et al. [37] propose methods for having

- *J.-S. Yang is with SungKyunKwan University, Suwon-Si, Gyeonggi-Do 440-746, Korea. E-mail:js.yang@skku.edu.*
- *N.A. Touba is with the University of Texas at Austin, 1 University Station, #C0803, Austin, TX 78712-1084. E-mail: touba@ece.utexas.edu.*
- *B. Nadeau-Dostie is with Mentor Graphics, 8005 Southwest Boeckman Road, Wilsonville, OR 97070.*
E-mail: Benoit_Nadeau-Dostie@mentor.com.

Manuscript received 22 Oct. 2010; revised 8 May 2011; accepted 10 Aug. 2011; published online 30 Sept. 2011.
Recommended for acceptance by C. Metra.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-2010-10-0582.
Digital Object Identifier no. 10.1109/TC.2011.189.

0018-9340/12/$31.00 © 2012 IEEE Published by the IEEE Computer Society

one dedicated flip-flop drive the activation signal for multiple control points, i.e., sharing the dedicated flip-flops among the control points to reduce the total number of dedicated flip-flops that are required. Sethuram et al. [27] introduced a test point insertion technique using the unused scan flip-flops which are available in their specific design macros and reduced both test volume and automatic test pattern generation (ATPG) CPU time. Haoxing et al. [7] proposed a low cost test point insertion method using gates with a controllability near 0.5 to drive control points and avoided the reconvergent fan-out issue by using gates outside of test point fan-in or fan-out logic.

In this paper, we present a new test point insertion method that reduces the area impact. Preliminary results were presented in [36]. A key feature of the proposed approach is that it significantly reduces the test point area overhead by removing the dedicated flip-flops used for driving the control points.

In Section 2, we give the motivation and the overview of the proposed work. Sections 3 and 4 describe the proposed test point insertion flow detail with examples. Testability with different signal probabilities is discussed in Section 5. Experimental results are shown in Section 6 and conclusions are given in Section 7.

2 Motivation and Overview of Proposed Method

As shown in Section 1, a number of techniques have been proposed to reduce the overhead for test point insertion. In spite of these efforts, the International Technology Roadmap for Semiconductors (ITRS) [11] predicts that logic BIST for random patterns will still take about 3.1 percent of chip area whereas the area for test compression will vary from 1.1 to 1.7 percent. Hetherington et al. [10] indicate that logic BIST area takes 1.58 percent of the chip area and 0.78 percent of chip area is used for test points (49.3 percent of logic BIST area). One unpublished industrial design evaluation shows that logic BIST adds 1.34 percent to the chip area of which about 30 percent is related to the test points (0.4 percent to the chip area), and another shows that 2.68 percent chip area is increased by logic BIST and test points take 1.16 percent chip area. This suggests that test points correspond to 43 percent of the area increase in logic BIST. Test point area may vary depending on the circuit characteristics, the number of pseudorandom patterns used, and the fault coverage required. However, a considerable portion of BIST area is usually related to test points, so it is important to find new techniques that can reduce the area overhead. In this paper, focusing on this significant area overhead for BIST, a new method for reducing the area impact of test point insertion is proposed by removing the dedicated flip-flops used for driving the control points.

Fig. 1 shows a design synthesis flow that incorporates scan, BIST, and test point insertion. In the conventional approach, when test points are inserted, dedicated flip-flops are assigned to drive the control points and capture the observation points to achieve higher fault coverage. The idea proposed here is to minimize the area overhead by replacing the dedicated flip-flops for driving control points with existing functional flip-flops in the design. The test points are first inserted with any TPI algorithm [3], [5], [12], [21], [25], [32], [37]. Then, the proposed method performs a postprocessing step in which functional flip-flops are identified for driving the control points via logic cone analysis. Control points are only considered and observation points are not modified. The dashed box in Fig. 1 indicates the postprocessing flow that finds and replaces the control points to generate the netlist.

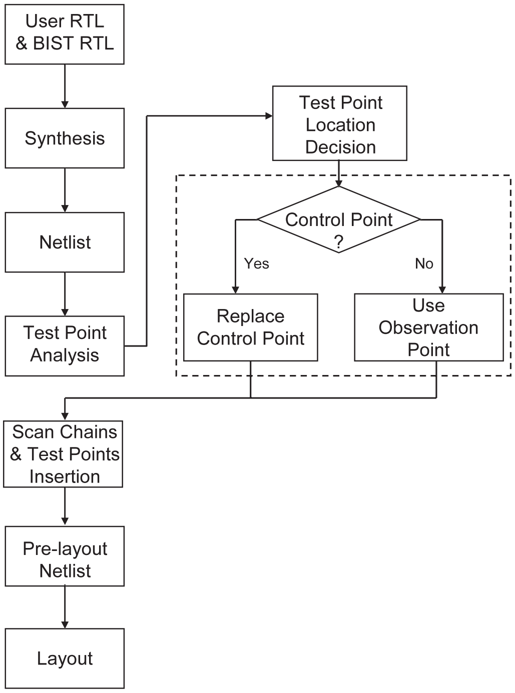
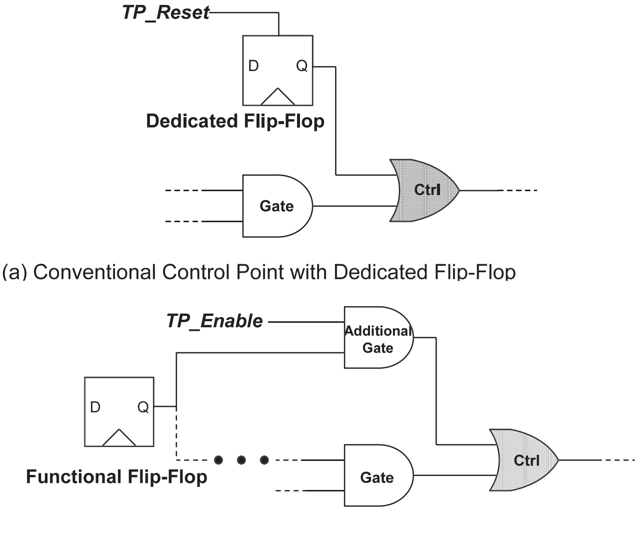


Fig. 1. New design synthesis flow proposed with a postprocessing step for testability and area overhead minimized test point insertion.

Since the proposed test point insertion method replaces the dedicated flip-flops for control points, a postprocessing in a dashed box identifies functional flip-flops which are suitable to drive the control point. First, functional flip-flops in the fan-in of the control point are only considered as candidates to ensure that no new timing constraints are introduced between any two flip-flops. The method inherently introduces reconvergent paths sourced by the candidate flip-flops and has the potential to introduce redundant faults. Redundancies are avoided by taking into account the path inversion parity of the reconvergent paths. The proposed method essentially achieves the same fault coverage as an implementation based on dedicated flip-flops, but with lower area cost. The functional flip-flops which are “logically” near the control point are chosen as candidates to replace a dedicated control point flip-flop for two reasons. The first reason is to minimize the length of the newly created test path from the candidate flip-flop to the control point. The second reason is that the transitions through the control point will have roughly the same delay as those along the functional path from the selected functional flip-flop. As will be explained in detail in Section 3.2, the proposed method does not create any relationships between control points and unrelated registers, and hence no new timing constraints are introduced.



(a) Conventional Control Point with Dedicated Flip-Flop

(b) Example of Proposed Control Point with Functional Flip-Flop

Fig. 2. Conventional and proposed control point.

Since the proposed method modifies the CUT, it needs to make sure that it does not reduce the random pattern testability achieved. 1) Test point activation/deactivation—Test points are activated in test mode and the activation signal is controlled by a flip-flop scanned in with a pseudorandom value for each scan vector in a conventional BIST application. And test points are deactivated while the system operates. During system operation, the activation signal is set to its noncontrolling value so that the functional logic value can pass through the control gate. For this purpose, we introduce a global signal, called, "*TP_Enable*" and it enables and disables the control points depending on the system mode. 2) New timing path—Newly introduced signal creates a new timing path from the functional flip-flop to the control point; however, it guarantees that no performance penalty is introduced. It will be explained in detail in Section 3.1.

Fig. 2a shows an example of a conventional control point that uses a dedicated flip-flop and Fig. 2b illustrates an example of the proposed control point that is driven by a functional flip-flop. The functional flip-flop not only drives the AND gate at the bottom but also operates as a test point driver via the additional gate path in Fig. 2b.

3 Proposed Test Point Insertion Details and Testability Consideration 1

To reduce area overhead, the proposed method introduces a few primitive gates and adds a new global signal. This may impact the testability of the design; hence, we propose one guideline to minimize the number of redundant or untested faults introduced by circuit modification.

- Opposite Path Inversion Parity—There are paths from a functional flip-flop to the control point. One is an existing functional path from a functional flip-flop to a control point and the other is the newly introduced path which is ANDed with the *TP_Enable* signal. Opposite path inversion parity along the paths from a functional flip-flop to a control needs to be maintained. Having opposite inversion parity along these two paths makes a path testable by appropriately applying either "*0*" or "*1*."

The following sections describe each of the required steps in the proposed test point insertion method for replacing the dedicated flip-flops with functional flip-flops to drive the control points.

3.1 Logic Cone Analysis to Find Candidate Functional Flip-Flops

Fig. 3 shows an example of logic with a conventional test point insertion which uses a dedicated flip-flop. This circuit has flip-flops (denoted *A* to *I*) and combinational elements (denoted *G1* to *G17* and *Ctrl*). It has one control point highlighted in gray color (*Ctrl*) and a dedicated flip-flop *I* drives the control point in test mode. Pseudorandom patterns are shifted in to drive the control point (*Ctrl*) in test mode. The test point is activated when the output of gate *Ctrl* is fixed to a "*1*" (i.e., *control-1 point*). And if an *AND* gate is used as *Ctrl*, the output of *Ctrl* is fixed to a "*0*" when it is activated (i.e., *control-0 point*). During the system operation, *Ctrl* is made transparent by having a noncontrolling value in flip-flop *I* so that the value in *G10* can be transferred to the one input of *G12* without any change.

To find the functional flip-flop for replacing the dedicated flip-flop, logic cone analysis needs to be performed. As mentioned above, the path inversion parity rule needs to be taken into consideration in performing logic cone analysis. Logic cone analysis starts from the control point (*Ctrl*) and traces back to the flip-flops. In a depth-first

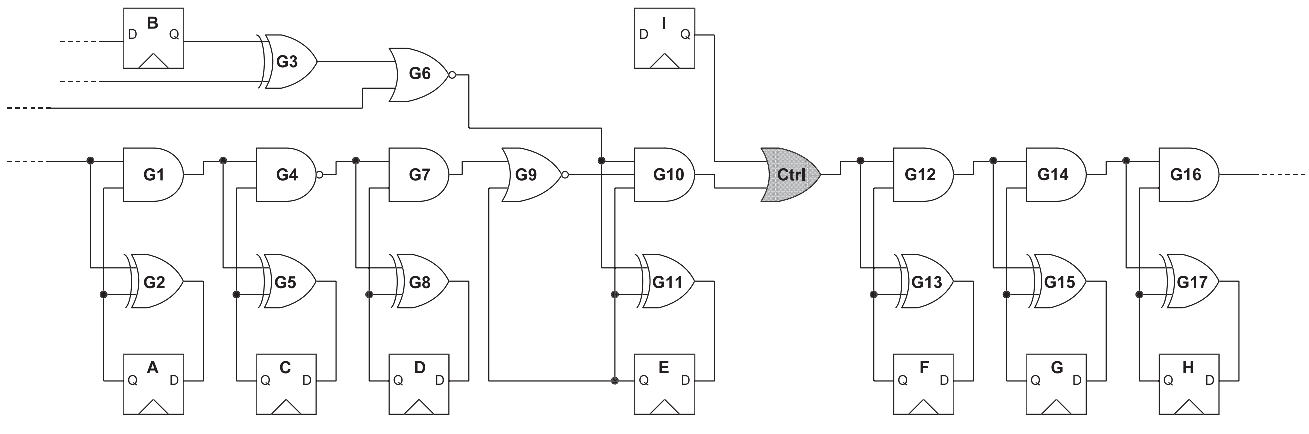


Fig. 3. Example of a circuit by conventional control point (*Ctrl*) insertion with dedicated flip-flop (*I*).

search manner (a breadth-first search manner can also be used), all primitive gates and flip-flops belonging to the logic cone of *Ctrl* are visited. Flip-flop I and *G10* are directly connected to *Ctrl*. However, since it is dedicated for the control point, I needs to be dropped from the search space. *G6* and *G3* are visited in a depth-first search and B is a second found flip-flop in the *Ctrl* fan-in cone. While visiting gates, the logical distance from *Ctrl* is counted. In this way, flip-flops A, B, C, D, and E are found as candidates for replacing the control point dedicated flip-flop I. As shown in Section 2, a new timing path is a matter of concern in selecting a functional flip-flop to replace a dedicated flip-flop. Therefore, logical distance information needs to be considered so as not to introduce any delay paths that add performance overhead. The functional flip-flop distance to a control point can be measured based on the number of levels of logic. The logical distance is used to maximize the probability for the test point driver to be relatively close to the test point to minimize the length of the wires.

To maintain the opposite path inversion parity, the inversion parity information is also checked while the nodes are traversed when searching. In Fig. 3, gates *G4*, *G6*, and *G9* introduce the inverse path parity. There is one inverting gate, *G6*, from *Ctrl* to the flip-flop *B*; hence, *B* has odd inversion parity. In the same manner, logic cone analysis with parity information consideration finds flip-flops A and C with even inversion parity from *Ctrl*. B and D have odd parity along their paths to the control point and E has even and odd parity paths from *Ctrl*. In this analysis, gates with dual polarity are considered as noninverting gates. Some gates such as XOR gates and MUXes have both noninverting and inverting paths and they introduce dual polarity. We assume those gates as noninverting sources. Further analysis on dual polarity will be discussed in Section 6.

The following shows the results of logic cone analysis for Fig. 3. There exist two multiple paths from *Ctrl* to a flip-flop E; hence, the logic cone analysis shows the even and odd inversion parity and two logical distances. And other flip-flops have a single path from *Ctrl*.

CandidateFlip-Flop	*Inversions*	*LogicalDistance*
A	2	5
B	1	3
C	2	4
D	1	3
E	0 & 1	1 & 2

There may be cases when only one functional flip-flop is found as a candidate by logic cone analysis. This occurs when a test point has a single functional flip-flop in its fan-in. This happens when the biased controllability (to a certain value either "0" or "1") is required higher than 0.5 such as in an OR or AND tree. In this case, the proposed test point insertion method does not replace a dedicated flip-flop.

3.2 Proposed Control Point Structure

Fig. 1 shows an example of a conventional control point that uses a dedicated flip-flop and Fig. 3 illustrates an example with a dedicated flip-flop I. Assume that it is replaced by one of the functional flip-flops among A to E. If a functional flip-flop directly drives the control point, it affects the system function. As mentioned in Section 2, test points need to be deactivated while the system operates. To hold this transparency property during system operation, one global signal called "TP_Enable" is introduced and it is deactivated during system operation.

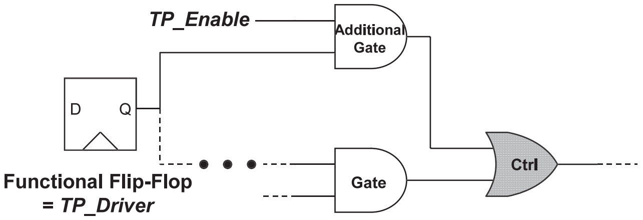


Fig. 4. Proposed control point structure with functional flip-flop.

Fig. 4 shows the proposed control point structure which is driven by a functional flip-flop. The functional flip-flop is also named as test point driver (*TP_Driver*) which sources *Ctrl* through the existing path. And it also drives *Ctrl* via the newly added path by TP_Enable with an additional gate in Fig. 4. The TP_Enable signal can block the signal propagation by setting its value to "*0*." This places a noncontrolling value at the input of the control point. When TP_Enable is "*1*," *Ctrl* can have a value determined by a functional flip-flop.

3.3 Path Inversion and Control Point Structures

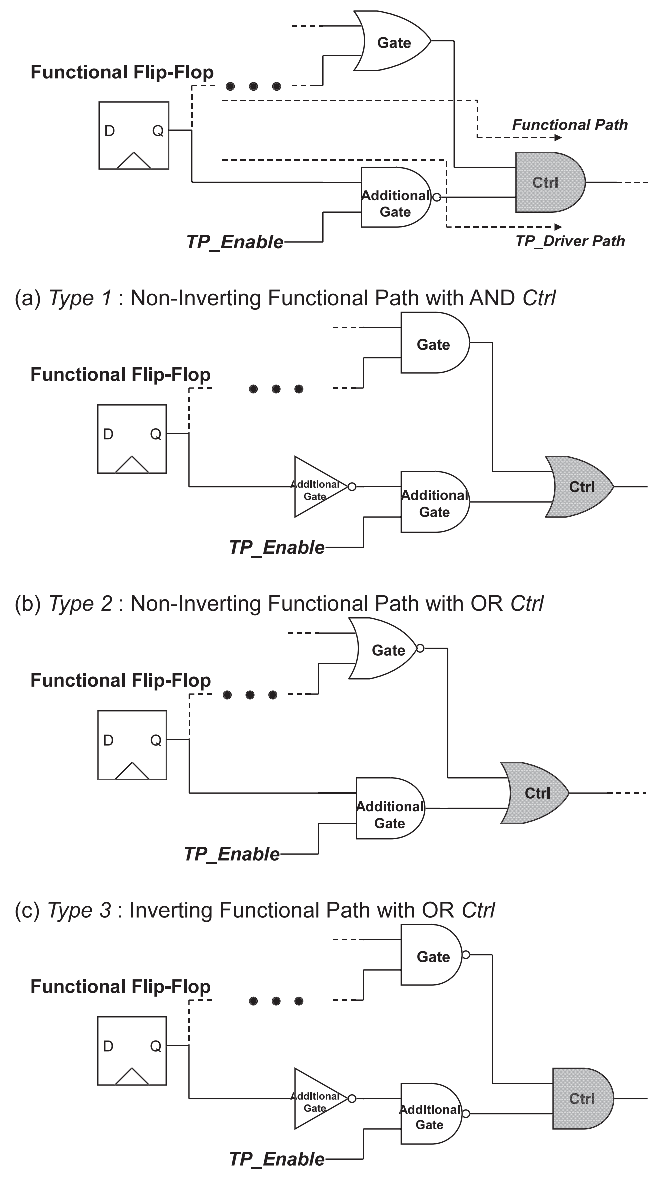
The proposed method modifies the CUT to try to maximize the random pattern testability. Four different types of control point structures are proposed to improve testability depending on the path inversion.

If a functional flip-flop is chosen to replace a dedicated flip-flop for the control point, a new path is created from the functional flip-flop to the control point. This new path will be referred to as the "*TP_Driver Path*." The original functional path will be referred to as the "*Functional Path*." A value in *Functional Path* can only propagate when TP_Enable is disabled in Fig. 4. However, the opposite inversion parity between the *TP_Driver Path* and *Functional Path* can enable propagation through *Functional Path* without disabling the test point. This increases the random pattern testability and helps to reduce the number of test patterns needed compared to having the same inversion parity along the two paths. Considering that either an AND or OR gate can be used for creating a control point, there are four types of control points that satisfy the inversion parity as shown in Fig. 5.

In Figs. 5a and 5b, the *TP_Driver Path* needs to have inversion because *Functional Path* has a noninverting path. Figs. 5c and 5d show the control point with an inverting path on *Functional Path*. When an inverter is added in the *TP_Driver Path* as in Figs. 5b and 5d, either an inverter can be used or the flip-flop's *Q_bar* can be connected to the additional gate.

3.4 Testability Enhancement Example

Fig. 6 shows an AND tree example with the proposed control point structure. Assume the path inversion is not considered, no inversion would be made along the *TP_Driver Path* when a dedicated control point driver flip-flop is replaced, as in Fig. 4. In this case, hard to test faults have a small probability of being able to propagate through



(a) *Type 1* : Non-Inverting Functional Path with AND *Ctrl*

(b) *Type 2* : Non-Inverting Functional Path with OR *Ctrl*

(c) *Type 3* : Inverting Functional Path with OR *Ctrl*

(d) *Type 4* : Inverting Functional Path with AND *Ctrl*

Fig. 5. Four control point structure types for different path inversion parity.

the circuit if all test points are disabled ($TP\_Enable = 0$). For example, the stuck-at-0 fault (S-A-0) at the input of TP2 is one of the hard to detect faults, and it would preferably need TP1 to be active in order to propagate S-A-0 through the AND gate which is located after TP1 and TP2, and then to propagate through TP3 and the remainder of the circuit. However, when path inversion is considered, the control point structure will solve this problem. S-A-0 at the input of TP2 could propagate even when TP_Enable is "1." All inputs of the AND gate should be "1" to provoke S-A-0 at TP2, and it automatically sets the AND gate with a controlling value (0). This makes the control point disabled without setting TP_Enable to be "0." TP_Enable reconvergence could be an issue in the proposed method; however, the path inversion analysis solves this problem so that the testability is not degraded. Hence, they are detected relatively easier than not having the path inversion information. The other hard to detect faults in Fig. 6 are the S-A-1 faults on the TP_Enable input of the AND gates disabling the test points. Since it is unlikely to randomly detect the faults on TP_Enable branches, ATPG patterns may be used for detecting most of these faults.

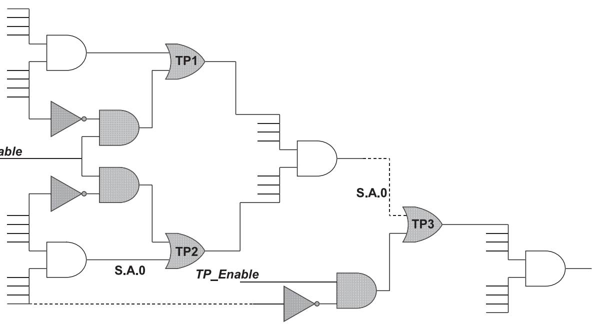


Fig. 6. AND tree example with proposed control point insertion.

4 Proposed Test Point Insertion Details and Testability Consideration 2

Previous section describes the logic cone analysis with logical distance and path inversion polarity. Here, we propose another guideline to maximize the testability by the proposed TPI method.

- Illegal Reconvergence Check—Reconvergence from the candidate functional flip-flop needs to be checked to avoid the case that blocks the propagation of hard to test faults. If the functional flip-flop used to drive the test point drives some gate in a fan-out of a test point, it may prevent from detecting hard to test faults. Details are explained with an example circuit.

Using the information achieved in Section 3, this section explains two ways to replace dedicated flip-flops by functional flip-flops.

4.1 Test Point Replacement Rule

4.1.1 Conservative Replacement Rule

As mentioned in Section 2, the functional flip-flop which is logically closest to the control point is chosen to replace a dedicated flip-flop for a control point. With a conservative replacement rule [36], if both inversion parities are found along paths from the control point to one flip-flop, that flip-flop is discarded from the candidate list. This guarantees that paths are always testable by having the opposite parity along the *Functional Path* and *TP_Driver Path*. For example, if during logic cone analysis, a flip-flop is found to have one path with one inversion, and another path with two inversions, it is not considered as a candidate. In Section 3.1, E is found to be the closest flip-flop to the control point location. However, there exist two paths with different polarity from the functional flip-flop E to *Ctrl* in Fig. 3. Therefore, E is not considered as a candidate functional flip-flop and the next nearest flip-flop to *Ctrl*, B or D, needs to be used to replace I. In this example, B is chosen to replace. Since B has an inverting functional path and OR gate is used as a control point gate, *Type 3* structure in

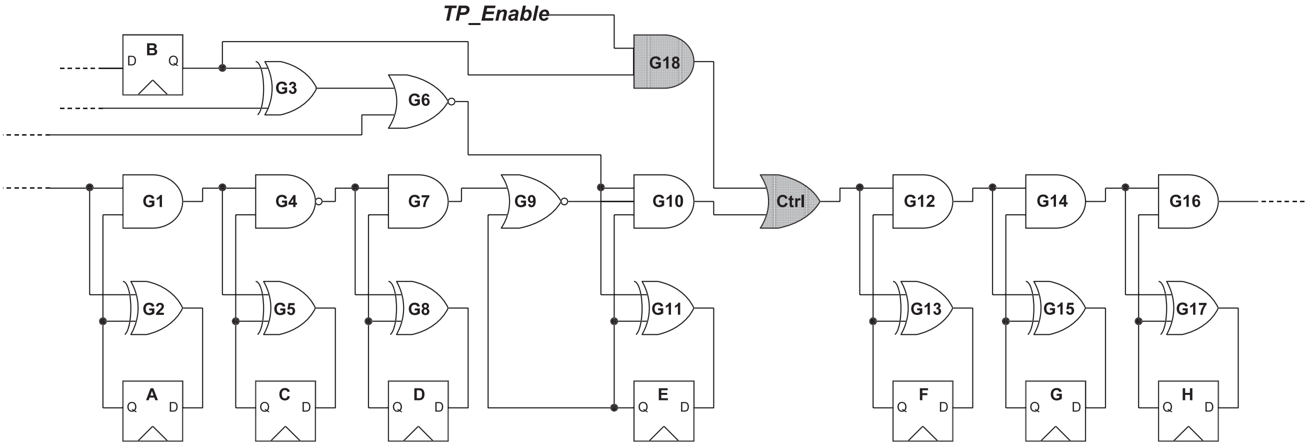


Fig. 7. Modified circuit example in Fig. 3 by proposed test insertion technique with functional flip-flop (B) using conservative replacement rule.

Fig. 5 is used as a new test point structure. Fig. 7 shows the control point insertion by the proposed TPI method based on the circuit in Fig. 3.

4.1.2 Relaxed Replacement Rule

Depending on circuits, there may be functional flip-flops in the fan-in of the control point which have multiple paths to Ctrl. If they have even and odd path inversion parity, they cannot be replaced by the conservative replacement rule. The relaxed replacement rule is applied to enhance the dedicated flip-flop replacement. In this rule, the functional flip-flops which have both inversions are also considered as candidates. Testability is still achievable by making opposite path inversion between *Functional Path* and *TP_Driver Path*. Assume that *Functional Path 1* has an even parity and *Functional Path 2* has an odd parity from the functional flip-flop to a control point. If *Functional Path 1* is chosen to determine the type of control point structure in Fig. 5, depending on the control point gate, *Type 1* or *Type 2* structure can be selected to follow the opposite path inversion parity guidance in Section 3. However, *Functional Path 2* and *TP_Driver Path* have the same path inversion parity. Since TP_Enable signal also can drive the control point with controlling and noncontrolling value, the testability does not get harmed. Hence, in Fig. 3, the relaxed replacement rule selects E to replace the dedicated flip-flop and *Type 2* structure is used as a new test point structure. Fig. 8 illustrates the CUT modification by the proposed TPI with a relaxed replacement rule. Note that if the same logical distance is found with different parity in functional paths, the control point type with less number of gates introduced in Fig. 5 needs to be selected to minimize the possibility of additional faults.

4.2 Illegal Reconvergence

Logic cone analysis determines the functional flip-flop candidates for driving control test points. For testability, since there may be many connections from the functional flip-flops to other nodes in a circuit, it is necessary to check whether the fault propagation is blocked. Reconvergence from *TP_Driver* in the fan-out of a control point can block the fault propagation. If any gate in the fan-out of a control point is sourced by *TP_Driver*, it can obstruct fault propagation and it may result in the loss of testability.

To illustrate the illegal reconvergence issue, a test point inserted circuit by the proposed method in Fig. 8 is modified; however, it is almost the same as the original circuit in Fig. 3 with the exception of OR gate *G20*. The relaxed replacement rule selects E as a *TP_Driver* based on the distance and *Type 2* control point is inserted to satisfy

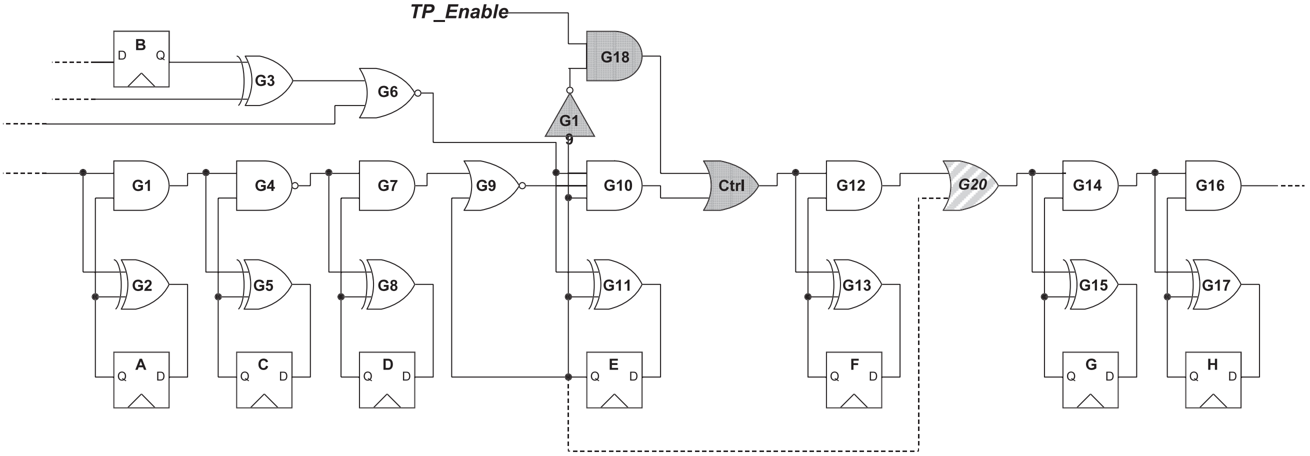


Fig. 8. Modified circuit example in Fig. 3 by proposed test insertion technique with functional flip-flop (E) using relaxed replacement rule and example of a circuit with illegal reconvergence.

TABLE 1
Number of Dedicated Flip-Flops for Test Points by Conventional and Proposed Test Point Insertion Techniques

Design	Conventional Method		Proposed Control Point Replacement Method						
	Num. Observation Point	Num. Control Points	Conservative Replacement Rule			Relaxed Replacement Rule			Improvement Ratio Between Rules(%)
			Num. Functional Flip-Flops	Num. Dedicated Flip-Flops	Reduction Ratio(%)	Num. Functional Flip-Flops	Num. Dedicated Flip-Flops	Reduction Ratio(%)	
A	3	24	22	2	91.7%	24	0	100%	100%
B	2	85	83	2	97.6%	84	1	98.8%	50%
C	100	733	659	74	89.9%	700	33	95.5%	55.4%
D	70	179	140	39	78.2%	164	15	91.6%	61.5%
OR1200	5	27	27	0	100%	27	0	100%	N/A
NOC	9	35	35	0	100%	35	0	100%	N/A

the inversion parity requirement. Assume E has a branch to *G20* illustrated as a dashed line from E to *G20*. This forms reconvergence from E to a fan-out of a control point. Due to the reconvergence, whenever E has "*1*," it drives *G20* with a controlling value and this may block the fault propagation.

As mentioned above, reconvergence from the candidate functional flip-flop needs to be checked and another functional flip-flop that satisfies the opposite path inversion rule and the illegal reconvergence check rule is found to drive control. Hence, the illegal reconvergence analysis removes E from the candidate list, and the next closet flip-flop is chosen. In Section 3, B and D have a distance 3 and they do not introduce a longer timing path than the existing longest path. There is no reconvergence from B or D to the fan-out of the control point, so they do not violate the illegal reconvergence condition. Since B and D have the odd inversion parity and an OR control point is used, a *Type 3* control point needs to be applied when replacing the dedicated flip-flop. If there is no flip-flop that satisfies the conditions for replacement, a dedicated flip-flop cannot be replaced.

5 Proposed Test Point Insertion Details and Testability Consideration 3

In conventional test point insertion techniques, test points are driven by the dedicated flip-flop and pseudorandom patterns are shifted in. Hence, in general, test points are generally assumed to be always enabled with a controllability of 0.5. In this context, in the proposed method, TP_Enable signal generally has a value of "1" so as to have 0.5 controllability at the control point. However, the stuck-at-1 fault on TP_Enable can only be detected when the TP_Enable signal is set to "0." To detect this fault, TP_Enable needs to take on a value of "0" some times. Experiments were performed to find the optimal signal probability for TP_Enable that would maximize the testability. Different input size OR gates are used to change the TP_Enable signal probability. For example, if two equiprobable pseudorandom signals are ORed together, the signal probability is increased to 0.75. In the general case, driving the TP_Enable signal by a k input OR gate achieves a $(2^{k-1})/(2^k)$ signal probability.

Different TP_Enable signal probabilities change the controllability on the control points and detectability of the stuck-at-1 fault on the TP_Enable signal. In Section 6, experimental results are shown for different signal probabilities for the TP_Enable signal.

6 Experimental Results

In this section, experimental results are presented with the proposed test insertion method. Four industrial designs, OR1200 (OpenRisc Processor) [22], and a network-on-chip (NOC) design [35] are used and test points are inserted. The LogicVision testpointAnalyze tool [20] was used to determine the location of test points in each design.

Based on the testability considerations in Sections 3, 4, and 5, the proposed method determines the functional flip-flop that can be used as *TP_Drivers*. In Table 1, the number of dedicated flip-flops that are replaced by functional flip-flops using the proposed method is shown. The first column gives the design name and four industrial designs are named as *Designs A-D*. The second and third columns show the number of observation points and control points, respectively, with a conventional test point insertion method. The summation of the two column values is the total number of test points. Each test point requires a dedicated flip-flop. For example, *Design A* has three observation points and 24 control points. In total, the conventional test point insertion method adds 27 dedicated flip-flops. Since the proposed test point insertion method replaces the dedicated flip-flops for control points, the number of observation points is the same as the conventional method. The fourth and fifth columns show the proposed test point insertion results with a conservative replacement rule. The fourth column shows the number of functional flip-flops used to replace the dedicated flip-flops for control points in the third column. The fifth column gives the number of dedicated flip-flops for control points which are unable to be replaced by the proposed method. In *Design A*, 22 out of 24 dedicated flip-flops for control point are replaced by the proposed method with a conservative replacement rule and two dedicated flip-flops are not replaced. As explained in Sections 3 and 4, if there is only one functional flip-flop in the candidate list or no candidate meets the rules, a dedicated flip-flop cannot be replaced. The sixth column shows the reduction ratio which is computed as the number of functional flip-flops used to replace dedicated

TABLE 2
Synthesized Area Result for OR1200

	Original	Scan Insertion	Relative Increase	Dedicated F/F	Relative Increase	Functional F/F	Relative Increase
Combinational	178072	229417	28.83%	229745	0.18%	229951	0.30%
Sequential	127543	155937	22.26%	157866	1.51%	156236	0.24%
Total	305615	385351	26.09%	387611	0.74%	386187	0.27%

flip-flops (the fourth column) divided by the number of total control points (the third column). Results for the relaxed replacement rule are shown in the seventh and eight columns. The ninth column shows the reduction ratio for the relaxed replacement rule. Since the relaxed rule enables more replacement by loosening the path inversion restriction, two nonreplaced control points by the conservative replacement rule in *Design A* are all replaced by the functional flip-flops. These results using both rules show a significant area reduction by replacing dedicated flip-flops using the proposed method. The last column shows the improvement ratio for the relaxed replacement rule over the conservative replacement rule. As the data show, more than 50 percent of nonreplaced flip-flop can be removed using the relaxed replacement rule. In the relaxed replacement rule, the only nonreplaced flip-flops are found when there is only one flip-flop existing in the fan-in of the control point and this cannot be replaced due to the skewed controllability requirement as explained in Section 3.

To better understand the test point area reduction, OR1200 and NOC designs are synthesized with 130 nm TSMC technology. Since OR1200 is an open source design, the synthesis results are provided in Table 2. Synthesized results are shown for combinational, sequential logic, and the summation of them. The second column shows the area of the original design and third column describes the area when scan chains with TAP and logic BIST are inserted in the original design. And the increase rate is shown in the fourth column. Area for conventional test point insertion is in the fifth column and the sixth column shows the relative increase over the original design. The seventh and eighth columns show the synthesized area and the increase over the original design, respectively. As expected, since conventional test point insertion techniques involve adding dedicated flip-flops, there is a considerable increase in the sequential logic part compared to the proposed method which replaces them by existing functional flip-flops. And the proposed method introduces extra primitive gates and gives higher area overhead than the conventional method; however, the total area reduction is significant. (NOC synthesis result also shows the similar result.)

In OR1200 and NOC, each of the new control points driven by a functional flip-flop takes approximately $1/4$ of the area of the original control points driven with a dedicated flip-flop. Therefore, the extrapolated area for *Designs A-D* can be calculated based on the following equation:

$$\frac{NewArea}{OldArea} = \frac{Nobs + Ndedicated + k * Nfunctional}{Nobs + Ndedicated + Nfunctional}$$
$$= 1 - Area\_reduction,$$

where *Nobs* denotes the number of flip-flops for observation points, and *Ndedicated* and *Nfunctional* indicate the number of dedicated flip-flops and functional flip-flops used for control point, respectively. Since the k factor is approximately 0.25 for both OR1200 and NOC, we calculate the area reduction of *Designs A-D*. Table 3 shows the area reduction results by the proposed method. Since the relaxed replacement rule shows the better performance, the higher area reduction is achieved.

Fault coverage results are shown in Table 4 with proposed test point implementation techniques and the standard LogicVision implementation. The first column of upper and lower tables in Table 4 gives the design name. There are four different cases for which results were generated. These were when no test points are inserted (*NO TP*), test points are inserted with dedicated flip-flops (*Dedicated F/F*), when the proposed method with a conservative replacement rule is used to replace dedicated flip-flops with functional flip-flops (*Conservative Rule*) and when dedicated flip-flops are replaced by the proposed relaxed replacement rule (*Relaxed Rule*). Since the testability varies with the TP_Enable signal probabilities in the proposed methods, four different probabilities (1/2, 15/16, 63/64, and 255/256) are used to evaluate the random pattern testability. Because *Designs A-D* are random-pattern-resistant circuits, the fault coverage is checked with 16,000 and 100,000 random patterns. Since OR1200 and NOC designs are found to be relatively random pattern testable circuits, 2,048 and 16,000 random test patterns are applied. With different TP_Enable signal probabilities, the fault coverage by *NO TP* is shown in the second and seventh columns and the third and eight columns give the fault coverage when test points are inserted with dedicated flip-flops (*Dedicated F/F*) for 16,000 and 100,000 patterns (2,048 and 16,000 patterns for OR1200 and NOC), respectively. The coverage

TABLE 3
Test Point Area Reduction Result

Design	Test Point Area Reduction	
	Conservative Replacement Rule	Relaxed Replacement Rule
A	61.6%	66.7%
B	71.6%	72.4%
C	59.3%	63.0%
D	42.2%	49.4%
OR1200	63.0%	63.0%
NOC	59.4%	59.4%

TABLE 4
Testability Comparison of Proposed Methods with Conventional Implementation

Design	No TP	Dedicated F/F	TP_Enable Probability	Conservative Rule	Relaxed Rule	No TP	Dedicated F/F	TP_Enable Probability	Conservative Rule	Relaxed Rule
	16000 Patterns					100000Patterns				
A	85.86%	88.99%	1/2	88.07%	88.07%	88.19%	93.35%	1/2	92.06%	92.06%
			15/16	89.06%	89.13%			15/16	93.16%	93.16%
			63/64	89.17%	89.17%			63/64	93.30%	93.30%
			255/256	89.17%	89.17%			255/256	93.30%	93.30%
B	85.80%	92.04%	1/2	91.05%	91.05%	89.08%	94.37%	1/2	94.36%	94.36%
			15/16	92.26%	92.26%			15/16	94.52%	94.52%
			63/64	92.15%	92.15%			63/64	94.80%	94.80%
			255/256	92.13%	92.13%			255/256	94.77%	94.77%
C	84.45%	88.10%	1/2	87.51%	87.56%	87.72%	91.67%	1/2	90.42%	90.41%
			15/16	88.34%	88.40%			15/16	91.64%	91.64%
			63/64	88.16%	88.19%			63/64	91.21%	91.34%
			255/256	87.64%	87.54%			255/256	90.37%	90.39%
D	91.61%	97.66%	1/2	97.56%	97.62%	95.05%	98.71%	1/2	98.61%	98.61%
			15/16	97.66%	97.69%			15/16	98.63%	98.71%
			63/64	97.54%	97.57%			63/64	98.61%	98.65%
			255/256	97.34%	97.34%			255/256	98.46%	98.57%
	2048 Patterns					16000 Patterns				
OR1200	93.18%	98.96%	1/2	98.44%	99.44%	95.32%	99.74%	1/2	99.70%	99.70%
			15/16	98.86%	98.86%			15/16	99.72%	99.72%
			63/64	98.86%	98.86%			63/64	99.76%	99.76%
			255/256	98.65%	98.65%			255/256	99.72%	99.72%
NOC	97.78%	99.41%	1/2	99.40%	99.40%	98.84%	99.87%	1/2	99.73%	99.73%
			15/16	99.42%	99.42%			15/16	99.81%	99.81%
			63/64	99.42%	99.42%			63/64	99.81%	99.81%
			255/256	99.42%	99.42%			255/256	99.75%	99.75%

results show that the proposed method achieves almost the same fault coverage as the conventional TPI method does. The small fault coverage difference, about 0.05-0.1 percent from most of the benchmark circuits, between *Dedicated F/F* and *Conservative/Relaxed Rule* with 15/16 signal probability essentially corresponds to the number of faults added by the new test points that can only be detected when *TP_Enable* is "0." Those faults, the faults on the *TP_Enable* branches, are very difficult to detect randomly and will require ATPG patterns. The fault coverage loss can be compensated by a combination of three options—applying more random patterns, calculating more top-up patterns, or adding more test points. However, note that since the fault simulator does not consider internal faults of flip-flops, it appears that the proposed method has more faults than the standard implementation even though there are actually less.

As mentioned in Section 3, in the proposed test point insertion method, the gates with dual polarity are considered as noninverting gates. However, an analysis of the dual polarity gates revealed that considering MUX primitives as noninverting gates is not optimal. This is because paths going through the select input have an implied dual polarity. For example, in Fig. 5a illustrating a *Type 1* control point, suppose that *OR Gate ("Gate")* is a MUX primitive and its select input is directly connected to the candidate functional flip-flop output, then the stuck-at-0 fault on the select input becomes impossible to detect without setting *TP_Enable* to *0*. Changing the control point for a *Type 4* control point does not improve the situation as it makes the stuck-at-1 the hard to detect fault instead. Therefore, candidate flip-flops with a path going through the select input of a MUX primitive should be discarded. However, this nonoptimal MUX primitives management had no impact on the experimental results of three of the six circuits since MUXes were implemented or modeled as AND/OR structures in those circuits. *Design D* has approximately 38,161 instances of MUXes modeled as MUX primitives; however, the result shows that the testability seems similar to that of other design. Hence, in the experiments, the testability is not significantly affected by this issue.

As shown in Table 4, different *TP_Enable* signal probabilities are used to study how sensitive the fault coverage is by *TP_Enable*. Fig. 9 shows the fault coverage of

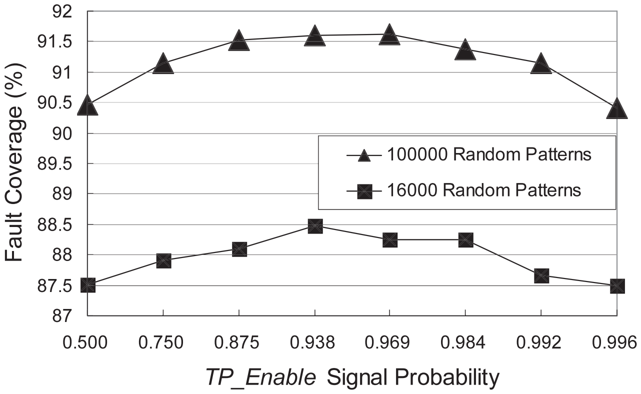


Fig. 9. Testability versus TP_Enable signal probability.

Design C with different signal probabilities of $(2^{k-1})/(2^k)$ for $k = 1$ to 6. 16,000 and 100,000 random patterns are applied to achieve random fault coverage. Graph shows the different fault coverage with different TP_Enable probabilities. Based on the simulation results, since fault coverage varies about 1.2 percent only by changing the signal probability, TP_Enable signal probability can give a significant improvement in the fault coverage. This is to be expected since the LogicVision testpointAnalyze tool assumes that the TP_Enable probability is exactly *1*. Both cases illustrate that there is saturation of the coverage. Therefore, the probability of TP_Enable needs to be kept high, say 15/16 or 31/32, so that the efficiency of the original test point insertion method is not affected. In the proposed method, the maximum fault coverage is obtained in this way. Some circuits show that the coverage goes down a little when TP_Enable has a signal probability 15/16 compared with 63/64. This may happen because of the noise related to vectors. When different vectors are applied to CUT, they can introduce the noise. This might result in the lower test coverage in the benchmark circuits.

OR1200 was used in order to verify that no performance penalty is introduced. For both the conventional method and the proposed method, the same critical path (from *startpoint : or1200_ctrl/ex_insn_reg[1]* to *the endpoint: or1200_genpc/pcreg_reg[29]*) is found by *Synopsys Design-Compiler* timing report [28]. This shows that the proposed method reduces significant area overhead without performance overhead.

7 Conclusions

In this paper, a new test point insertion technique which replaces the dedicated flip-flops for control points with existing functional flip-flops is proposed. The experimental results indicate that the methodology proposed in this paper can significantly reduce the number of dedicated flip-flops by replacing them with functional flip-flops. By considering the testability issues, significant area savings are achieved while preserving the random pattern testability of the circuit and without introducing new timing constraints that would complicate timing closure. The test point area was typically reduced by about more than half while the fault coverage loss during the random pattern phase was limited to less than 0.1 percent for most circuits. Several options such as applying more random patterns, calculating more top-up patterns, or adding more test points can be applied to compensate for a slightly higher coverage loss.

The proposed method can be used with any existing test point insertion procedures without having to modify their algorithms. The method is therefore neutral with respect to the handling of unknowns in the circuit and test power as it does not deal with the selection of the test points, only their implementation. Dedicated flip-flop replacement reduces the dynamic power on the clock network compared with conventional test point insertion methods [7]. The proposed method can be easily used on top of existing conventional test point insertion algorithms by replacing the added dedicated flip-flops. The proposed method only involves static tracing of the fan-in and fan-out of gates which are related to control points and very efficient algorithms are available for performing these tasks which are less complex than the algorithms used for test point selection itself. It should also be noted that the proposed new test point implementation method gives the flexibility of adding more test points to achieve even higher coverage or reduce test time.

Future work includes a new implementation for observation points. This will be useful for circuits for certain designs with a significant number of observation points.

Acknowledgments

This research was supported in part by the US National Science Foundation (NSF) under Grant No. CCF-0916837.

References

[1] C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller, and B. Koenemann, “OPMISR: The Foundation for Compressed ATPG Vectors,” *Proc. Int’l Test Conf.*, pp. 748-757, 2001.

[2] M. Bershteyn, “Calculation of Multiple Sets of Weights for Weighted Random Testing,” *Proc. Int’l Test Conf.*, pp. 1031-1040, 1993.

[3] A.J. Briers and K.A.E. Totton, “Random Pattern Testability by Fast Fault Simulation,” *Proc. Int’l Test Conf.*, pp. 274-281, 1986.

[4] M. Chatterjee and D.K. Pardhan, “A Novel Pattern Generator for Near-Perfect Fault-Coverage,” *Proc. IEEE VLSI Test Symp.*, pp. 417-425, 1995.

[5] K.-T. Cheng and C.J. Lin, “Timing -Driven Test Point Insertion for Full-Scan and Partial-Scan BIST,” *Proc. Int’l Test Conf.*, pp. 506-514, 1995.

[6] E.B. Eichelberger and E. Lindbloom, “Random-Pattern Coverage Enhancement and Diagnosis for LSSD Logic Self Test,” *IBM J. Research and Development*, vol. 27, no. 3, pp. 265-272, May 1983.

[7] R. Haoxing, M. Kusko, V. Kravets, and R. Yaari, “Low Cost Test Point Insertion without Using Extra Registers for High Performance Design,” *Proc. Int’l Test Conf.*, 2009.

[8] S. Hellebrand, S. Tarnick, J. Rajski, and B. Courtois, “Generation of Vector Patterns through Reseeding of Multiple-Polynomial Linear Feedback Shift Register,” *Proc. Int’l Test Conf.*, pp. 120-129, 1992.

[9] 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. Computers*, vol. 44, no. 2, pp. 223-233, Feb. 1995.

[10] G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassab, A. Hassan, and J. Rajski, “Logic BIST for Large Industrial Designs: Real Issues and Case Studies,” *Proc. Int’l Test Conf.*, pp. 358-367, 1999.

[11] “The International Technology Roadmap for Semiconductors,” *Semiconductor Industry Assoc.*, 2007.

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

[13] A. Jas and C.V. Krishna, and N.A. Touba, "Hybrid BIST Based on Weighted Pseudo-Random Testing: A New Test Resource Partitioning," *Proc. IEEE VLSI Test Symp.*, pp. 2-8, 2001.

[14] R. Kapur, S. Patil, T.J. Snethen, and T.W. Williams, "Design of an Efficient Weighted Random Pattern Generation System," *Proc. Int'l Test Conf.*, pp. 491-500, 1994.

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

[16] B. Koenemann, C. Barnhart, B. Keller, T. Snethen, O. Farnsworth, and D. Wheater, "A SmartBIST Variant with Guaranteed Encoding," *Proc. Asian Test Symp.*, pp. 325-330, 2001.

[17] C.V. Krishna, A. Jas, and N.A. Touba, "Test Vector Encoding Using Partial LFSR Reseeding," *Proc. Int'l Test Conf.*, pp. 885-893, 2001.

[18] B. Krishnamurthy, "A Dynamic Programming Approach to the Test Point Insertion Problem," *Proc. Design Automation Conf.*, pp. 695-704, 1987.

[19] L. Lai, J.H. Patel, T. Rinderknecht, and W.-T. Cheng, "Hardware Efficient LBIST with Complementary Weights," *Proc. Int'l Conf. Computer Design*, pp. 479-481, 2005.

[20] *ETAnalysis Tools Reference Manual, Software Version 6.0a*, 2007.

[21] M. Nakao, S. Kobayashi, K. Hatayama, and K. Iijima, "Low Overhead Test Point Insertion for Scan-Based BIST," *Proc. Int'l Test Conf.*, pp. 384-357, 1999.

[22] OPENCORES, http://www.opencores.org, 2011.

[23] I. Pomeranz and S.M. Reddy, "3-Weight Pseudo Random Test Generation Based on a Deterministic Test Set for Combinational and Sequential Circuits," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 7, pp. 1050-1058, July 1993.

[24] J. Rajski, J. Tyszer, M. Kassab, N. Mukherjee, R. Thompson, T. Kun-Han, A. Hertwig, N. Tamarapalli, G. Mrugalski, G. Eider, and Q. Jun, "Embedded Deterministic Test for Low Cost Manufacturing Test," *Proc. Int'l Test Conf.*, pp. 301-310, 2002.

[25] H.D. Schnurmann, E. Lindbloom, and R.G. Carpenter, "The Weighted Random Test-Pattern Generator," *IEEE Trans. Computers*, vol. 24, no. 7, pp. 695-700, July 1975.

[26] B.H. Seiss, P. Trouborst, and M. Schulz, "Test Point Insertion for Scan-Based BIST," *Proc. European Test Conf.*, pp. 253-262, 1991.

[27] R. Sethuram, S. Wang, S.T. Chakradhar, and M.L. Bushnell, "Zero Cost Test Point Insertion Technique to Reduce Test Set Size and Test Generation Time for Structured ASICs," *Proc. Asian Test Symp.*, 2006.

[28] *Synopsys Design Compiler*, A-2007.12-SP4, http://www.synopsys.com/tools/implementation/rtlsynthesis/Pages/default.aspx.

[29] N. Tamarapalli and J. Rajski, "Constructive Multi-Phase Test Point Insertion for Scan-Based BIST," *Proc. Int'l Test Conf.*, pp. 649-658, 1996.

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

[31] N.A. Touba and E.J. McCluskey, "Synthesis of Mapping Logic for Generating Transformed Pseeudo-Random Patterns for BIST," *Proc. Int'l Test Conf.*, pp. 674-682, 1995.

[32] N.A. Touba and E.J. McCluskey, "Test Point Insertion Based on Path Tracing," *Proc. IEEE VLSI Test Symp.*, pp. 2-8, 1996.

[33] H.-C. Tsai, K.-T. Cheng, L.-J. Lin, and S. Bhawmik, "Efficient Test Point Selection for Scan-Based BIST," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 6, no. 4, pp. 667-676, Dec. 1998.

[34] H.-J. Wunderlich and G. Kiefer, "Bit-Flipping BIST," *Proc. IEEE/ACM Int'l Conf. Computer-Aided Design*, pp. 337-343, 1996.

[35] J.-S. Yang and N.A. Touba, "Automated Selection of Signals to Observe for Efficient Silicon Debug," *Proc. IEEE VLSI Test Symp.*, 2009.

[36] J.-S. Yang, N.A. Touba, and B. Nadeau-Dostie, "Test Point Insertion Using Functional Flip-Flops to Drive Control Points," *Proc. Int'l Test Conf.*, 2009.

[37] M. Youssef, Y. Savaria, and B. Kaminska, "Methodology for Efficiently Inserting and Condensing Test Points," *IEEE Proc. Computers and Digital Techniques*, vol. 140, no. 3, pp. 154-160, May 1993.

Joon-Sung Yang (S'05-M'09) received the BS degree from Yonsei University, Seoul, Korea, in 2003, and the MS and PhD degrees from the University of Texas at Austin, in 2007 and 2009, respectively, all in electrical and computer engineering. After graduation, he worked for Intel Corporation for three years. He is currently with SungKyunKwan University in Korea. His research interests are VLSI testing, silicon debug, and nanometer scale test and design methodologies. He was a recipient of Korea Science and Engineering Foundation (KOSEF) Scholarship in 2005 and the Best Paper Award at the 2008 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems. He is a member of the IEEE.

Nur A. Touba (SM'05-F'09) received the BS degree from the University of Minnesota, Minneapolis, in 1990, and the MS and PhD degrees from Stanford University, California, in 1991 and 1996, respectively, all in electrical engineering. He is currently a professor with the Department of Electrical and Computer Engineering, University of Texas at Austin. He was a recipient of the US National Science Foundation (NSF) Early Faulty CAREER Award in 1997, the Best Paper Award at the 2001 VLSI Test Symposium, and the 2008 Defect and Fault Tolerance Symposium. He served as a program chair for the 2008 International Test Conference and the general chair for the 2007 Defect and Fault Tolerance Symposium. He currently serves on the program committee for the Design Automation and Test in Europe Conference, International On-Line Test Symposium, European Test Symposium, Asian Test Symposium, Defect and Fault Tolerance Symposium, and International Test Synthesis Workshop. He became a fellow of the IEEE in 2009.

Benoit Nadeau-Dostie (S'80-M'85) received the PhD degree in electrical engineering from the Université de Sherbrooke, Québec, Canada, in 1985. He has been the chief architect at Mentor Graphics since 2009. He is defining test strategies and algorithms with a focus on embedded test. Prior to this, he was LogicVision's chief scientist for 15 years. From 1986 to 1994, he was an advisory engineer at Bell-Northern Research (BNR). He was the main architect of BNR's Design-for-Testability (DFT) strategy. From 1985 to 1986, he was with the Department of Electrical Engineering of the Université Laval (Vision and Digital Systems Lab). His contribution was an auditory prosthesis based on a microelectronics neural stimulator. He edited one book, and published several articles in scientific journals and international conferences. He also holds 40 US patents related to memory, logic, and board testing. He is a senior member of the IEEE and was involved in many of its activities over the last 25 years of his career including program committees of international conferences, working group of the 1149.1 standard, and editorial board member of the *Design and Test of Computers* magazine.