The article has been accepted for inclusion in a future issue of ETRI Journal, but has not been fully edited. Content may change prior to final publication.
http://dx.doi.org/10.4218/etrij.14.0113.1121

Test Point Insertion with Control Point by Greater Use of Existing Functional Flip-Flops

Joon-Sung Yang, and Nur A. Touba

This paper presents a novel test point insertion method for pseudo-random built-in self-test (BIST) to reduce the area overhead. Recently, a new test point insertion method for BIST was proposed which tries to use functional flip-flops to drive control test points instead of adding extra dedicated flip-flops for driving the control points. Replacement rule used in the previous work has limitations preventing some dedicated flip-flops from being replaced by functional flip-flops. This paper proposes a logic cone analysis based test point insertion approach to overcome the limitations. Logic cone analysis is performed to find candidate functional flop-flops for replacing dedicated flip-flops. Experimental results indicate that the proposed method reduces test point area overhead significantly with minimal loss of testability by replacing the dedicated flip-flops.

Keywords: Test Point Insertion, BIST, Control Point, Logic Cone Analysis, Functional Flip-Flop

I. Introduction

Testers have limitations in terms of I/O channels, bandwidth, speed, etc., which becomes a bottleneck for relying on conventional external testing. BIST (Built-in self-test) supports to reduce the dependency on external testers by reducing test time, tester investment cost, test data bandwidth and test data storage requirements [1]-[3].

With BIST, circuits that generate test patterns and analyze the output responses of the logic are embedded on-chip. The test pattern generator automatically generates the patterns for application to the inputs of the circuit-under-test (CUT). The output response analyzer compacts the output response of the CUT into a signature. This provides a variety of benefits including the ability to apply a large number of test patterns in a short time (i.e., shorter test time), at-speed testing, minimal automatic test equipment (ATE) storage requirements, test application in the field over the lifetime of the part, and a reusable test solution for embedded cores. In particular, BIST is crucial for applications such as aerospace, defense, automotive, computer, etc., for the reliability of the entire system.

The most economical logic BIST techniques are based on pseudo-random pattern testing. On-chip input pattern generator constructed from a linear feedback shift register (LFSR) is most commonly used to generate pseudo-random patterns with its compact structure. And on-chip output response analyzer compacts the output responses into a signature and this allows significant compaction of test data. Pseudo-random pattern testing also can achieve high coverage of non-modeled 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 pseudo-random patterns. There have been many research efforts to overcome the fault coverage limitation by r.p.r faults. Mainly two directions are given to enhance the fault coverage. One is to modify the pattern generator in order to generate patterns that detect hard faults. Various methods have been proposed such as weighted pattern generation [2], [4]-[8], pattern mapping [9]-[11], bit-fixing [12], bit-flipping [13], and LFSR reseeding [14]-[19].

The other approach is to make the CUT random testable by

Manuscript received Nov. 21, 2013; revised Jan. 29, 2014; accepted Nov. 6, 2014.

Joon-Sung Yang (corresponding author, js.yang@skku.edu) is with the department of Semiconductor Systems Engineering, Sungkyunkwan University, Suwon, Rep. of Korea.
Nur Touba (touba@ece.utexas.edu) is with the department of Electrical and Computer Engineering, University of Texas at Austin, Austin, USA.

inserting test points [20] and this enhances the r.p.r faults detection probability by modifying the CUT. Test point insertion (TPI) involves adding control and observation points to the CUT. Observation points make a node observable by adding an extra flip-flop or sampling it through a scan-chain. Control points involve AND or OR operations for a node with an activation signal. The activation signal is driven by a test dedicated flip-flop (this flip-flop is only used to drive an internal node only for test purposes and is referred to as a dedicated flip-flop or a test dedicated flip-flop) which receives pseudo-random values during BIST and is set to a non-controlling value during normal operation (also referred as functional mode). Additional hardware is needed to form the test points which add area and performance overhead to a design. Since optimal test point placement is NP-complete [16], a number of TPI methods have been proposed using fault simulation [21], [22] and testability measures [18]. TPI for minimizing performance overhead [23], [24] and TPI for minimizing area overhead [25]-[28] are two general strategies for TPI methods.

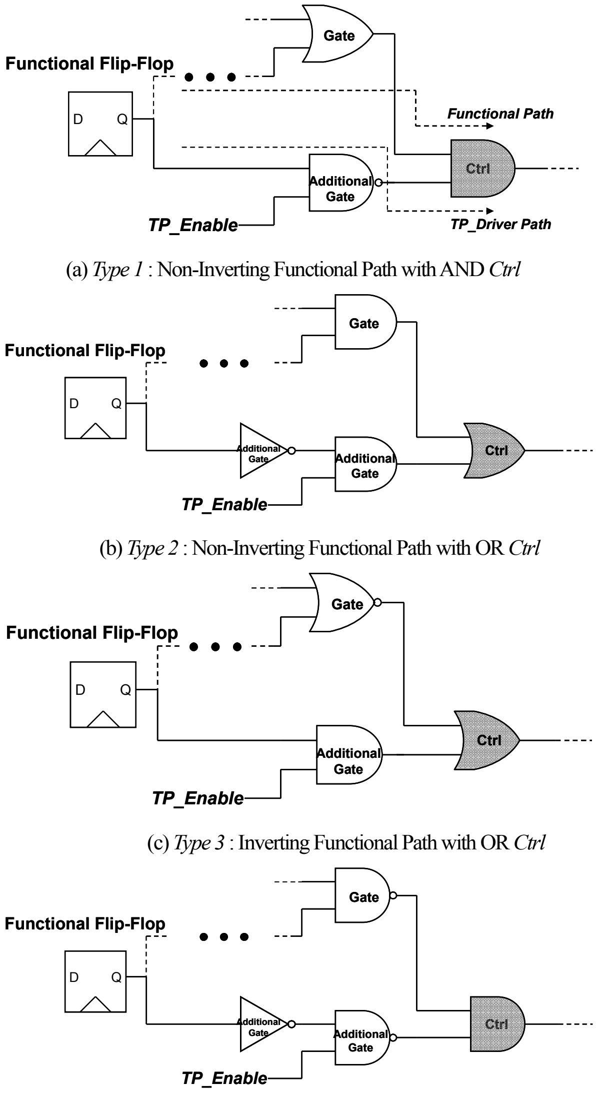
A new test point insertion method was proposed in [29]. It replaces test dedicated flip-flops for driving control points by existing functional flip-flops via conservative replacement rules. It was shown that a significant test point area reduction can be achieved. However, the method used in [29] limits the candidate search space, leaves some test dedicated flip-flops as non-replaceable and thus may limits the area overhead reduction that is achieved.

In this paper, we propose a novel test point insertion method that replaces dedicated flip-flops for control points with functional flip-flops to further reduce the area overhead. [30] introduces a way to replace non-replaceable flip-flops by relaxing the replacement rules in [29]. This paper proposed a different test point insertion method. Unlike in [30], the proposed method replaces dedicated flip-flops via logic cone analysis without relaxing any replacement rules. It is able to replace more test dedicated flip-flops and achieves significant area reduction. Preliminary results were presented in [31]. This paper shows an in-depth analysis with benchmark circuits including industrial circuits. A key feature of the proposed approach is the greater effort to reduce the test point area overhead by removing the dedicated flip-flops used for driving the control points.

II. Overview of Functional Flip-Flop Driving Test Point Insertion Method [29] and Its Drawback

This section gives an overview of the test point insertion method proposed in [29]. Usually, when test points are inserted, dedicated flip-flops for test purpose are added to drive control points and capture the observation points to increase fault coverage. Extra dedicated flip-flops for control and observation points add the area overhead. [29] proposed a test point insertion method that replaces the dedicated flip-flops by existing functional flip-flop for control points. This helps to minimize the area overhead by reducing the number of dedicated flip-flops for driving control points.

Since the test point insertion method in [29] replaces the dedicated flip-flops for control points, during the test point decision stage, functional flip-flops are identified that are suitable to drive the control point. There are guidelines required for a new test point insertion approach - 1) no new timing constraints and 2) no loss of the testability. Based on



(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. 1. Four New Control Point Structures Driven by Functional Flip-Flops [29, 30]

the above guidelines, the four new control point structures which do not require test dedicated flip-flops are proposed in Fig. 1. As can be seen from *Type 1 – 4*, there are paths from a functional flip-flop to the control point (*Ctrl*). One is an existing functional path from a functional flip-flop to a control point (*Functional Path*) and the other is the newly introduced path which is ANDed/NANDed with the *TP_Enable* signal (*TP_Driver Path*). Because *Functional Path* and *TP_Driver Path* converge, this naturally introduces reconvergent paths driven by the functional flip-flop. Hence, there is a possibility to introduce redundant faults. To avoid redundancies, the opposite path inversion parity is considered. Path inversion parity means the number of inversions by inverters, nand or nor gates. 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*'. This is a first replacement rule used in [29]. [30] relaxed this rule to replace more dedicated flip-flops. Sec.III.2.1 gives a detailed explanation with an example. Test points are only activated in test mode and this should not change any functional operation during the system operation. To hold this transparency property, test points are deactivated while the system operates by setting the test point activation signals to its non-controlling value so that the functional logic value can arrive to the control gate. A global signal, "*TP_Enable*", is introduced to enable and to disable test points for different modes of the system. *TP_Enable* signal can also help to change the *TP_Driver Path* value if a functional flip-flop has a very skewed signal probability.

As a second replacement rule, [29] and [30] check the illegal reconvergence. Illegal reconvergence is defined as a path reconvergence blocking the fault propagation. Reconvergence from the 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 fanout of a test point, it may prevent from detecting hard to test faults.

TPI in [29] is performed based on two replacement rules - 1) the opposite path inversion and 2) the illegal reconvergence check. This identifies candidate functional flip-flops for driving control points. To keep the testability, the functional flip-flops with the same inversion polarity, even or odd polarity, are only chosen as candidates. For example, if there are multiple paths with different inversion polarity from the functional flip-flop to the control point, the functional flip-flop is discarded from the candidate list for dedicated flip-flop replacement. This may leave many control points not replaced by functional flip-flops. One limitation in [29] is that, depending on the design, there may be some control points that do not have either all even or all odd inversion parity on *Functional Paths* to the control point. The second limitation is that dedicated flip-flops cannot be replaced if there are no functional flip-flops satisfying the reconvergence check to avoid testability loss. The other limitation could be the single functional flip-flop in the control point fan-in cone. There may be cases when only one functional flip-flop is found as a candidate. This occurs when a test point has a single functional flip-flop in its fan-in. This happens when the controllability to a certain value (*'0'* or *'1'*) needs to be higher than 0.5. For example, AND or OR trees are more likely to have a skewed controllability on the functional path, either *'0'* or *'1'*, so they may have a single functional flip-flop in order to have a skewed value.

III. Enhanced Test Point Insertion Details

This section describes the enhanced test point insertion algorithm for overcoming limitations - 1) no inversion parity path found and 2) illegal reconvergence path found - addressed in the previous section. The proposed method uses a logic cone analysis to perform test point insertion which is different from [29] and [30]. It always performs better TPI than the conventional methods because the proposed method is not restricted by the replacement rules. In this paper, the difference between the conventional methods [29], [30] and the proposed method is highlighted. To efficiently explain the proposed method and highlight the difference, we will show an example describing how the proposed method can replace more dedicated flip-flops which are not replaced by [29] and

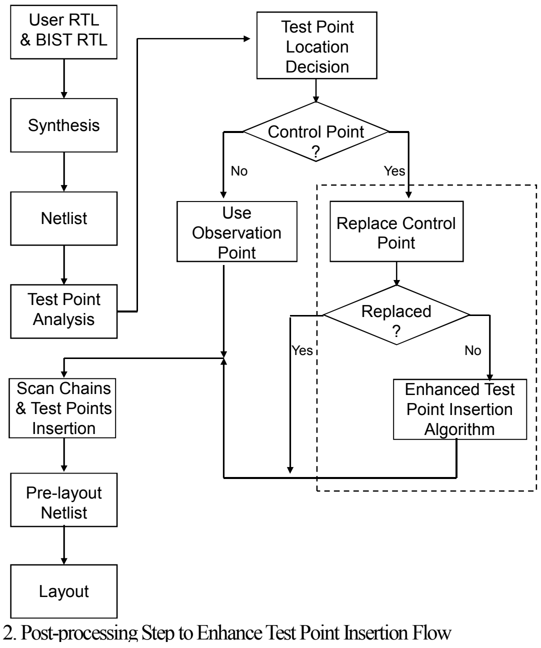


Fig. 2. Post-processing Step to Enhance Test Point Insertion Flow

[30].

1. Enhanced Test Point Insertion Flow

This paper describes the enhanced test point insertion flow. Fig. 2 shows a design synthesis flow that incorporates scan, BIST, and test point insertion. During the test point insertion stage, functional flip-flop candidates for each control point are identified. The proposed method performs a post-processing step in which the control points are identified for not being replaced by the functional flip-flops. The dashed line in Fig. 2 indicates the post-processing flow that runs the enhanced test point insertion flow. The post-processing performs additional dedicated flip-flop replacement for those which are not replaced by TPI in [29], [30]. Note that the proposed method is a different method than [29], [30], however, this paper explains the flow as if the proposed method works on top of the method in [29], [30]. This is only to illustrate how the proposed method replaces more dedicated flip-flops which are not replaced by conventional approaches. But, it should be noted that this is only for an illustrative purpose and all test dedicated flip-flops are replaced by the proposed flow.

2. Enhanced Test Point Insertion Details

2.1 Test Point Insertion in [29]

An example of logic with test dedicated flip-flops by a conventional test point insertion is given in Fig. 3. There are 11 flip-flops (denoted *A* to *K*) and combinational elements (dented *G1* to *G11* and *Ctrl*). It has one control point highlighted in gray (*Ctrl*) and a dedicated flip-flop *E* drives the control point in test mode.

Based on the analysis [29], the following inversion and logical distance (or logical depth) information can be generated from functional flip-flops to *Ctrl*. The number of inversions is checked through paths from functional flip-flop to Ctrl and the logical distance from Ctrl is measured.

Candidate Flip-Flop	*Inversions*	*Logical Distance*
A	*2 & 3*	*2 & 3 & 4*
B	*2 & 3*	*2 & 3 & 4*
D	*2*	*3*

As explained in the previous section, *A* and *B* violate the inversion parity check. Because they have even and odd inversion parity through multiple paths to *Ctrl*, the dedicated flip-flop *E* for a control point cannot be replaced by *A* or *B*. Flip-flop *D* has an even parity to *Ctrl*, however, it does not satisfy the reconvergence check because it reconverges with a *Ctrl* output path. This may block the propagation of hard to detect faults. *D* cannot be used to replace the dedicated flip-flop *E*, hence, there are no functional flip-flops that can replace *E* in Fig 3. This may leave many dedicated flip-flops not replaced since they are limited by the rules in [29]. This limits the area reduction.

2.2 Identifying Test Point Fan-out Cone

Unlike the method shown in [29], [30], the goal with the enhanced test point insertion flow described here is to find more candidate functional flip-flops that can be used to replace the dedicated flip-flops to achieve more area reduction without relaxing the replacement rules – inversion parity check and illegal convergence check.

Assume that there are no functional flip-flops available in the current set of candidates for performing the replacement. In this case, the enhanced test point insertion flow can be used to find possible functional flip-flops that can replace the dedicated flip-flop for the test point. This flow finds a functional flip-flop that does not belong to the test point fan-in that may cause additional timing concerns. Hence, the selection needs to confirm that the proposed method does not induce a performance overhead.

The enhanced test point insertion flow initiates from the test point. The nodes are visited from the test point and the test point fan-out cone is found. Each node is traversed from the control point and this gives functional flip-flops that are related to the outputs in the test point logic cone. In Fig. 3, via fan-out cone analysis, *G6*, *G8*, *G10* and *G11* are visited and *I* and *J* are found as flip-flops that belong to the fan-out cone of control point. Therefore, we choose *I* and *J* as candidates for the dedicated flip-flop (*E*) replacement.

Visited Gates	*Flip-flops in Fan-out cone*
G6, G8, G10, G11	*I & J*

Once the functional flip-flops in the fan-out cone of control point are identified, the timing related information is obtained because the fan-in cone and fan-out cone analysis find all logic related to the control point. The other flip-flops outside of the fan-out cone of the control point do not guarantee the timing requirements, hence, they will not be considered.

2.3 Analyzing Fan-in Cones from Candidate Functional Flip-Flops

Once all the inputs which are related to the test point are found, a functional flip-flop needs to be selected to replace the dedicated flip-flop. To avoid introducing additional timing constraints, only functional flip-flops in fan-out cone need to be considered. In the example in Fig. 3, flip-flops *I* and *J*, are found in *Ctrl* fan-out cone by the analysis described in the previous section.

From each output of the logic cone, the logic cone analysis can generate a fan-in cone. Fan-in cone analysis for *I* and *J* can be performed and this gives all flip-flops belonging to *I* and *J* fan-in cone. Based on the analysis, the following flip-flops are found in the fan-in cone of *I* and *J*.

Fan-in Cone	*Flip-flops belonging to Fan-in cone*
I	*A, B, C, D, E, F*
J	*A, B, C, D, E, F, G*

In Sec.III.2.1, *A*, *B*, *D* and *E* are initially considered, however, they are found to be not-suitable for replacing a dedicated flip-flop because they do not meet the replacement rules. Hence, we consider the possible candidates to replace the dedicated flip-flops as *C*, *F* and *G*. They are the inputs which are not in the test point's fan-in cone, but found from the logic cone analysis from the outputs.

2.4 Replacing Dedicated Flip-Flops

Once the fan-in cone analysis finds all inputs related to the test point, a functional flip-flop needs to be selected to replace the dedicated flip-flop. There are two types of candidates. One is an input whose fan-out cone includes all the outputs of the test point's fan-out cone. The other type of input partially covers the outputs of the test point's fan-out cone. *A* type of candidates is found by generating a fan-out cone of each functional flip-flop. In Fig. 3, since *C*, *F* and *G* are candidates, we generate fan-out cones respectively. From *C*, the logic cone analysis finds the fan-out cone which includes outputs *I* and *J*. Fan-out of *F* finds output *I* and *J*, and *J* and *K* are found by the logic cone analysis from *G*.

Fan-out Cone Start	*Outputs of Fan-out Cone*
C	*I & J*
F	*I & J*
G	*K & J*

In the enhanced test point insertion flow, the additional timing constraint is the main concern. To guarantee no performance penalty, it is necessary to find functional flip-flops that cover all the outputs of the test point's fan-out cone. Inputs that have all of the test point's fan-out cone's outputs as their outputs can be considered as candidates. In Sec.III.2.2, *I* and *J* are found by fan-out cone analysis of the control point. As can be seen from the above, the *C* and *F* fan-out cones cover *I* and *J*. Therefore, *G* is not acceptable and *C* and *F* can be considered as candidates for replacing a dedicated flip-flop, *E*.

It is also noted that the random pattern testability should not be degraded by the proposed TPI method. For testability, it is necessary to check whether the fault propagation is blocked since there may be many paths from the functional flip-flops to other nodes in a circuit. Hence, the illegal reconvergence is also checked when the candidate function flip-flops are determined by fan-in and fan-out cone analysis. 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. For example, in Fig. 3, *C* and *F* are found as candidates to replace a dedicated flip-flop for the control point. For these, the illegal reconvergence check needs to be performed for the testability. *C* drives *G6* and it also has a reconvergence path to *G10*, however, *F* is only fed *G6*. *C* violates an illegal reconvergence rule because it can block the fault propagation. Hence, *F* is chosen to replace

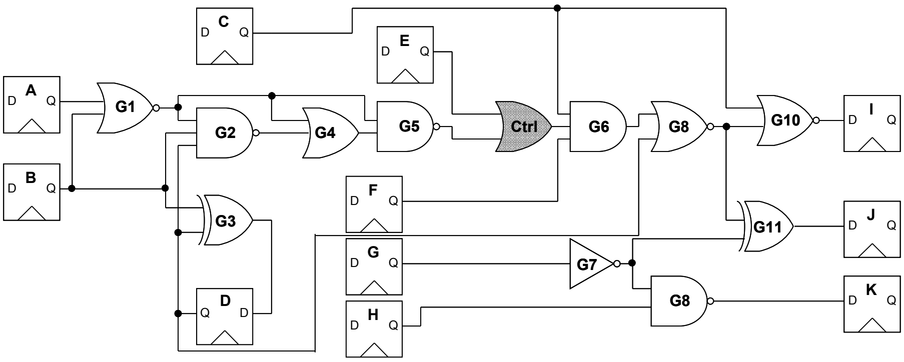


Fig. 3. Example of a Circuit by Conventional Control Point (*Ctrl*) Insertion with Dedicated Flip-Flop (*E*)

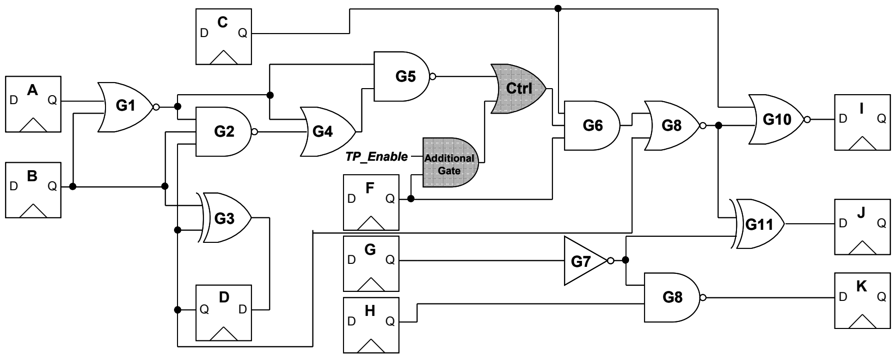


Fig. 4. Example of a Circuit by Proposed Test Point Insertion with Functional Flip-Flip (*F*)

the dedicated flip-flop. If there are many candidates which satisfy the fan-in and fan-out cone analysis and the reconvergence check, the flip-flop driving a gate closest to the control point needs to be chosen. A functional flip-flop that is logically close to the control point is chosen to replace a dedicated flip-flop. Acceptable candidates' logic cones partly share the logic with the test point's fan-out cone. The propagation from the test point is performed for each acceptable candidate until the overlapped gate element with the test point's fan-out logic cone is first found. Due to the performance overhead, it is required to minimize the length of the newly created test path from the candidate flip-flop to the control point. The input which has the closest overlapped element is selected for replacing the not-replaced dedicated flip-flops that remain after using the method in [29].

Once a functional flip-flop is determined for TPI, the new type of control point needs to be chosen. As can be seen in Fig. 1, the control point is driven by the combination of a functional flip-flop and a *TP_Enable* signal. In this manner, the control point is driven with *0's* and *1's*. And the functional flip-flop drives a gate which is placed on the fan-out path from the control point. In Fig. 3, *F* will be used to replace the dedicated flip-flop, hence, not only does it drive the control point but also it is fed into *G6*. *G6* is placed in the fan-out path from the control point. *G6* can propagate the control point output when the value at *F* is *1*, a non-controlling value. And F needs to drive the new control point with a non-controlling value in order to force *0's* and *1's* to the control point.

Fig. 4 shows the control point insertion by the proposed method based on the circuit in Fig. 3. The dedicated flip-flop, *E*, is replaced by the function flip-flop *F* with one additional AND gate. *Type 3* control point structure is used with a functional flip-flop.

Note that although this paper explains how not-replaced dedicated flip-flops can be replaced by the proposed method to show the difference with conventional approaches, in reality, all dedicated flip-flops are replaced by the logic cone analysis described in Sec.III.2.

IV. Experimental Results

In this section, experimental results are presented with the proposed test insertion method to evaluate the improvements that are obtained through the flow proposed. Four industrial designs, OR1200 (OpenRisc Processor) [32], a NOC (network-on-chip) design A (NOC-A) [33] and a NOC design B (NOC-B) [34] are used and test points are inserted. The Mentor Graphics Tessent tool [35] was used to determine the location of test points in each design. Dedicated flip-flops for driving the control points are replaced by the proposed flow to reduce the number of dedicated flip-flops without impacting delay and without increasing the loss of testability.

The proposed method determines the functional flip-flop that can be used to drive the control point. 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

Table 1. Control Point Dedicated Flip-Flop Replacement Improvement Comparison

Design	Conventional TP Insertion		Replacement in [30]		Replacement by Proposed Method		Replacement Improvement Ratio over [30]
	Observation Point	Control Point	Dedicated Flip-Flop	Functional Flip-Flop	Dedicated Flip-Flop	Functional Flip-Flop	
Design A	1451	3771	573	3198	517	3254	9.8%
Design B	129	371	13	358	8	363	38.5%
Design C	3	24	0	24	0	24	N/A
Design D	70	179	15	164	2	177	86.7%
OR 1200	5	27	5	22	0	27	100%
NOC-A	9	35	5	30	0	35	100%
NOC-B	12	58	21	37	0	58	100%

Table 2. Synthesized Area Results for OR 1200 [30] and NOC-A Designs

		Original	Scan Insertion	Relative Increase	Dedicated Flip-Flop	Relative Increase	Functional Flip-Flop	Relative Increase
OR1200	Combinational	178072	229417	28.83%	229745	0.18%	229951	0.30%
	Sequential	127543	155934	22.26%	157866	1.51%	156236	0.24%
	Total	305615	385351	26.09%	387611	0.74%	386187	0.27%
NOC-A	Combinational	83450	102860	23.26%	103236	0.45%	103549	0.83%
	Sequential	126377	165923	31.29%	168578	2.10%	166466	0.43%
	Total	209827	268783	28.10%	271814	1.44%	270015	0.59%

Design A – D. Second and third column show the number of observation points and control points respectively with a conventional test point insertion method. The total number of test points is the summation of two column values. A dedicated flip-flop is inserted to each test point. For example, *Design A* has 5222 test points - 1451 observation points and 3771 control points, respectively. A conventional test point insertion method inserts 5222 dedicated flip-flops. Since the test point insertion method in [30] ([30] is a replacement rule relaxed version of [29] and [30] shows better results than [29]) and this paper replaces the dedicated flip-flops for control points, the number of observation points remains the same as the conventional method. The fourth and fifth columns show the number of dedicated flip-flops and the number of functional flip-flop used to drive control points by TPI in [30]. Dedicated flip-flops in fourth column give the number of dedicated flip-flops for control points which fails be replaced by the method in [30]. And the fifth column shows the number of functional flip-flops used to replace the dedicated flip-flops for control points in the third column. In *Design A*, 3198 out of 3771 dedicated flip-flops for control points are replaced by the method in [30]. For dedicated flip-flops in the fourth column, the proposed method in this paper tries to replace them by the functional flip-flops and the results are shown in the sixth and seventh column. The proposed method replaces 3254 out of 3771 dedicated flip-flops for control in *Design A* and 363 out of 371 dedicated flip-flops in *Design B*. Both the proposed method and [30] replace all control points with a dedicated flip-flop. Both the proposed method and [30] replace all control points with a dedicated flip-flop in *Design C*. For OR1200 and NOC, designs are manipulated to generate non-replaceable dedicated flip-flops by [30] (all dedicated flip-flops were initially replaceable). The proposed method was able to find the functional flip-flop and replace dedicated flip-flops in OR1200 and NOC designs. Last column shows the replacement ratio. In *Design D*, OR1200, NOC-A and NOC-B, the proposed method replaces most of not-replaced dedicated flip-flops by [30]. The conventional method may work well for some circuits, however, the proposed method always enhances the performance of test point insertion than conventional methods. For example, a conventional TPI performs very well in *Design C*. However, regardless of characteristics of benchmark circuits, the proposed TPI significantly outperforms conventional methods shown in the replacement ratio. The proposed method in *Design D* achieves a very high replacement ratio while the conventional method does not. This is particularly evident for NOC-B where the conventional method replaces around 60% of the dedicated flip-flops using replacement rules, whereas the proposed method replaces 100% using logic cone analysis.

Design A and *B* shows a relatively low replacement ratio. They have a number of AND or OR tree structures and the control points used in those trees require skewed controllability. In this case, the controllability to a certain value (*'0'* or *'1'*) needs to be higher than 0.5, hence, dedicated flip-flops cannot be replaced. It should be noted that the low replacement ratio in *Design A* and *B* is caused not by the proposed TPI but by their design characteristics. They need dedicated flip-flops to have skewed controllability in testing.

As shown in Table 1, the proposed method further replaces dedicated flip-flops for control points. For the test point area reduction evaluation, two designs (OR1200 and NOC designs) are synthesized with 130nm TSMC technology [36]. The synthesized results are shown for combinational, sequential logic and the summation of two in Table 2. The second column shows the original design area and the third column provides the area when logic BIST and scan chains with TAP 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 synthesized area and the increase over the original design are shown in the seventh and eighth columns respectively. Because dedicated flip-flops are added by conventional test point insertion methods, the sequential logic part has a significant increase compared to the proposed method. And the proposed method adds extra primitive gates (*Additional Gate* in Fig. 4) and this gives little area overhead than the conventional method, however, the total area reduction is significant because dedicated flip-flops are removed.

In OR1200 and NOC-A, the new control point requires ¼ of

Table 3. Improvement Comparisons for Dedicated Flip-Flop Reduction Ratio and Test Point Area Reduction Ratio

Design	Conventional TP Insertion		Replacement in [30]		Replacement by Proposed Method		Test Point Area Reduction Improvement Ratio over [30]
	Reduction Ratio	Test Point Area Reduction	Reduction Ratio	Test Point Area Reduction	Reduction Ratio	Test Point Area Reduction	
Design A	N/A	N/A	84.8%	45.9%	86.2%	46.7%	1.7%
Design B	N/A	N/A	96.4%	53.7%	97.8%	54.4%	1.3%
Design C	N/A	N/A	100%	66.7%	100%	66.6%	N/A
Design D	N/A	N/A	91.6%	49.4%	98.8%	53.3%	7.5%
OR 1200	N/A	N/A	81.4%	51.5%	100%	63.2%	18.5%
NOC-A	N/A	N/A	85.7%	51.1%	100%	59.6%	14.3%
NOC-B	N/A	N/A	63.7%	39.6%	100%	62.1%	36.2%

the area of the original control point driven with a dedicated flip-flop. Therefore, the extrapolated area for *Design A – D* and NOC-B can be calculated based on the following equation [30].

$$\frac{New\ Area}{Old\ \ Area} = \frac{Nobs + Ndedicated + k * Nfunctional}{Nobs + Ndedicated + Nfunctional}$$
$$= 1 - Area\_reduction$$

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 variable *k*, *k* factor, is approximately 0.25 for OR1200 and NOC, the area reduction for *Design A – D* can be calculated.

Table 3 compares the improvement ratios of dedicated flip-flop reduction and test point area reduction. The reduction ratio is computed as the number of functional flip-flops used to replace dedicated flip-flops divided by the number of total control points. In *Design D*, [30] replaces 164 dedicated flip-flops out of 179 control point dedicated flip-flops and this achieves 91.6% improvement. 177 dedicated flip-flops are replaced by the proposed method in this paper and this gives 98.8% improvement. In terms of test point area reduction, the proposed method achieves 53.3% area reduction compared to 49.4% by [30] in *Design D*. For NOC-B, the reduction ratio is considerably enhanced from 63.7% to 100% and the test area reduction is significantly improved from 39.6% to 62.1%. The area reduction is calculated by the proposed equation shown above. As Table 1 shows, the proposed method replaces more number of dedicated flip-flops and this further reduces the area overhead in test point insertion. Table 3 shows the further area reduction results.

Table 4. Fault Coverage Comparisons

	NO Test Points	Conventional TPI	Method in [30]	Proposed Method
Design A	79.79%	96.82%	96.47%	96.45%
Design B	87.16%	98.25%	97.86%	97.85%
Design C	88.19%	93.35%	93.30%	93.30%
Design D	95.05%	98.71%	98.63%	98.25%
OR 1200	93.18%	98.96%	99.76%	99.76%
NOC-A	97.78%	99.41%	99.81%	99.71%
NOC-B	96.84%	99.07%	98.92%	98.92%

Fault coverage results are shown in Table 4 with no test point insertion and other test point insertion techniques (a conventional test point insertion using dedicated flip-flops, a method in [30] and the proposed method). The fault coverage is measured when 100000 random patterns are applied for *Design A – D* and 16000 random patterns are applied for OR1200 and NOC designs. The first column shows the designs. Second column shows the fault coverage when no test points are inserted. Testability results with a conventional test point insertion method using dedicated flip-flops, with a test point insertion method in [30] and with a proposed method are shown in third, fourth and fifth columns, respectively. The coverage results show that the proposed method achieves almost the same fault coverage as the conventional test point insertion method does. Some circuits show that the coverage goes down a little more than other circuits, for example *Design D*. This may happen because of the noise related to vectors. Noise is caused by the fact that not exactly the same vectors are applied and the slight differences in coverage are similar to the effect of changing the polynomial, seed or chain ordering. The coverage variations tend to reduce with an increasing number of vectors. This might result in the lower test coverage in the benchmark circuits. If a minimal loss of fault coverage difference is considered as an issue, the fault coverage loss can be compensated by a combination of three options – 1) applying more random patterns, 2) calculating more top up patterns or 3) adding more test points. However, it should be noted that the fault simulator does not model internal faults of flip-flops. Hence, it seems that the proposed method has more faults than the conventional approaches even though there are actually less number of faults with less number of dedicated flip-flops.

V. Conclusions

In this paper, a new test point insertion technique performs the logic cone analysis and finds more functional flip-flop candidates without a having rule based search that prunes out the functional flip-flop candidates [30]. This significantly increases the number of functional flip-flops can be used to replace the control points with a dedicated flip-flop. The experimental results indicate that a number of dedicated flip-flops are replaced by functional flip-flops using the proposed method in this paper. The proposed method reduces the area overhead in test point insertion and this helps to reduce the number of possible faults added by the additional logic. By considering the testability issues, considerable area savings are achieved while the random pattern testability of the circuit is preserved and the new timing constraints that would result in a performance overhead are not introduced. Overall, the test point area was reduced by about more than half while the fault coverage loss during the random pattern phase was kept very close to other test point insertion methods. There may be some ways such as adding more test points, applying top-up patterns or trying more random patterns to compensate a slightly higher coverage loss.

The proposed method can be easily adopted to exiting test point insertion algorithms to remove test dedicated flip-flops. The proposed method only involves a static fan-in and fan-out cone analysis regarding control points. 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 test point replacement flow considering the physical layout. Layout aware control point replacement flow might provide more number of functional flip-flop candidates from outside of the logic cone. This would help to further increase the number of control points, thereby, reduces the area overhead by the control points.

Acknowledgement

This paper was supported by Samsung Research Fund, Sungkyunkwan University, 2013.

References

[1] Barnhart, C., V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller, and B. Koenemann, "OPMISR: the Foundation for Compressed ATPG Vectors," *Proc. of International Test Conference*, pp. 748-757, 2001.

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

[3] Ren, H., M. Kusko, V. Kravets, and R. Yaari, "Low cost test point insertion without using extra registers for high performance design," *Proc. of International Test Conference*, pp. 1-8, 2009.

[4] Bershteyn, M., "Calculation of multiple sets of weights for weighted random testing," *Proc. of International Test Conference*, pp. 1031-1040, 1993.

[5] Lai, L., J.H. Patel, T. Rinderknecht, and W.-T. Cheng, "Hardware efficient LBIST with complementary weights," *Proc. of International Conference on Computer Design*, pp. 479-481, 2005.

[6] Paschalis, A., I. Voyiatzis, and D. Gizopoulos, "Accumulator Based 3-Weight Pattern Generation," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, Vol. 20, No. 2, pp. 357-361, 2012.

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

[8] Schnurmann, H.D., E. Lindbloom, and R.G. Carpenter, "Weighted test generator in built-in self-test design based on genetic algorithm and cellular automata," *International Conference on Electronic Measurement & Instruments* , pp. 134-138, 2011.

[9] Gherman, V., H.-J. Wunderlich, H. Vranken, F. Hapke, M. Wittke, and M. Garbers, "Efficient pattern mapping for deterministic logic BIST," *Proc. of International Test Conference*, pp. 48-56, 2004.

[10] Touba, N. A., and E. J. McCluskey, "Synthesis of Mapping Logic for Generating Transformed Pseudo-Random Patterns for BIST," *Proc. of International Test Conference*, pp. 674-682, 1995.

[11] Wunderlich, H.-J., H. Vranken, F. Hapke, M. Wittke, and M. Garbers, "Efficient pattern mapping for deterministic logic BIST," *Proc. of VLSI Test Symposium*, pp. 48-56, 2004.

[12] Li, W., C. Yu, S. M. Reddy, and I. Pomeranz, "A scan BIST generation method using a Markov source and partial bit-fixing," *Proc. of Design Automation Conference*, pp. 554-559, 2003.

[13] Wunderlich, H.-J., and G. Kiefer, "Bit-Flipping BIST," *Proc. IEEE/ACM International Conference on Computer-Aided Design*, pp. 337-343, 1996.

[14] Hellebrand, S., J. Rajski, S. Tarnick, S. Venkataraman, and B. Courtois, "Built-in Test for Circuits with Scan Based on Reseeding of Multiple-Polynomial Linear Feedback Shift Registers," *IEEE Trans. on Computers*, Vol. 44, No. 2, pp. 223-233, Feb. 1995.

[15] Kim, H.-S., and S. Kang, "Increasing encoding efficiency of LFSR reseeding-based test compression," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 25, No. 5, pp. 913-917, 2006.

[16] Kavousianos, X., V. Tenentes, K. Chakrabarty, and E.

Kalligeros, “Defect-Oriented LFSR Reseeding to Target Unmodeled Defects Using Stuck-at Test Sets,” *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, Vol. 19, No. 12, pp. 2330 – 2335, 2011.

[17] Lee, J. and N. A. Touba, “LFSR-Reseeding Scheme Achieving Low-Power Dissipation During Test,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 26, No. 2, pp. 396-401, 2006.

[18] Rajski, J., 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. of International Test Conference*, pp. 301-310, 2002.

[19] Wang, Z., H. Fang, K. Chakrabarty, and M. Bienek, “Deviation-Based LFSR Reseeding for Test-Data Compression,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 28, No. 2, pp. 259-271, 2009.

[20] Eichelberger, E. B., and E. Lindbloom, “Random-Pattern Coverage Enhancement and Diagnosis for LSSD Logic Self Test,” *IBM Journal of Research and Development*, Vol. 27, No. 3, pp. 265-272, May 1983.

[21] Basturkmen, N.Z., S. M. Reddy, and J. Rajski, “Improved algorithms for constructive multi-phase test point insertion for scan based BIST,” *Proc. of Asia and South Pacific Design Automation Conference*, pp. 604-611, 2002.

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

[23] Cheng, K.-T., and C.J. Lin, “Timing –Driven Test Point Insertion for Full-Scan and Partial-Scan BIST,” *Proc. of International Test Conference*, pp. 506-514, 1995.

[24] Tsai, H.-C., K.-T. Cheng, L.-J. Lin and S. Bhawmik, “Efficient Test Point Selection for Scan-based BIST,” *IEEE Trans. VLSI Systems*, Vol. 6, No. 4, pp. 667-676, 1998.

[25] Haoxing, R., M. Kusko, V. Kravets, and R. Yaari, “Low cost test point insertion without using extra registers for high performance design,” *Proc. of International Test Conference*, pp. 1-8, 2009.

[26] Nakao, M., S. Kobayashi, K. Hatayama, and K. Iijima, “Low Overhead Test Point Insertion for Scan-Based BIST,” *Proc. of International Test Conference*, pp. 384-357, 1999.

[27] Sethuram, R., 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. of Asian Test Symposium*, 2006.

[28] Tamarapalli, N., and J. Rajski “Constructive Multi-Phase Test Point Insertion for Scan-Based BIST,” *Proc. of International Test Conference*, pp. 649-658, 1996.

[29] Yang, J.-S., N. A. Touba, and B. Nadeau-Dostie, “Test Point Insertion Using Functional Flip-Flops to Drive Control Points,” *Proc. of International Test Conference*, pp. 1-10, 2009.

[30] Yang, J.-S., N.A. Touba, and B. Nadeau-Dostie, “Test Point Insertion with Control Points Driven by Existing Functional Flip-Flops”, *IEEE Transactions on Computers*, Vol. 10, No. 10, pp. 1473-1483, 2012.

[31] Yang, J.-S., B. Nadeau-Dostie, and N.A. Touba, "Reducing Test Point Area for BIST through Greater Use of Functional Flip-Flops to Drive Control Points", *Proc. of IEEE Symposium on Defect and Fault Tolerance*, pp. 20-28, 2009.

[32] OPENCORES, *http://www.opencores.org*

[33] Yang, J.-S., and N.A. Touba, “Automated Selection of Signals to Observe for Efficient silicon Debug,” *Proc. of VLSI Test Symposium*, 2009.

[34] Jang. W., and D. Z. Pan, “Application-Aware NoC Design for Efficient SDRAM Access,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 30, No. 10, pp. 1521-1533, 2011.

[35] Mentor Graphics Tessent Tools Reference Manual, 2010.

[36] Synopsys Design Compiler, A-2007.12-SP4