2009 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems

Reducing Test Point Area for BIST through Greater Use of Functional Flip-Flops to Drive Control Points

Joon-Sung Yang[1], Benoit Nadeau-Dostie[2] and Nur A. Touba[1]

[1]Computer Engineering Research Center
Dept. of Electrical & Computer Eng.
University of Texas, Austin, TX 78712
{jsyang, touba}@ece.utexas.edu

[2]Logic Vision, Inc.
25 Metro Drive, Third Floor
San Jose, CA, 95110
benoit@logicvision.com

Abstract

Recently, a new test point insertion method for pseudo-random built-in self-test (BIST) was proposed in [Yang 09] which tries to use functional flip-flops to drive control test points instead of adding extra dedicated flip-flops for driving the control points. This paper investigates methods to further reduce the area overhead by replacing dedicated flip-flops which could not be replaced in [Yang 09]. A new algorithm (alternative selection algorithm) is proposed to find candidate flip-flops out of the fan-in cone of a test point. Experimental results indicate that most of the not-replaced flip-flops in [Yang 09] can be replaced and hence even more significant area reduction can be achieved with minimizing the loss of testability.

1. Introduction

Built-in self-test (BIST) embeds test pattern generation and output response analysis on-chip. It provides numerous advantages in terms of reducing test generation costs, reducing tester storage requirements, allowing the rapid application of many patterns to target non-modeled faults, test reuse, and in-field testing where there is no access to a tester. The most efficient BIST techniques are based on pseudo-random testing where the test patterns can be generated using a linear feedback shift register (LFSR) which has a very compact structure. However, the presence of *random-pattern-resistant (r.p.r.)* faults which have low detection probabilities may prevent pseudo-random BIST from achieving sufficiently high fault coverage. There are two general approaches for improving the fault coverage for pseudo-random BIST: (1) modifying the pattern generator so it generates patterns to detect r.p.r faults, and (2) modifying the circuit-under-test (CUT) to eliminate the r.p.r. faults by increasing their detection probability.

In pattern generator modification, a number of techniques have been proposed including weighted pattern testing [Pomeranz 92], [Bershteyn 93], [Kapur 94], [Jas 01], pattern mapping [Chatterjee 95], [Touba 95a, 95b], LSFR reseeding [Konemann 91, 01], [Hellebrand 92, 95], [Krishna 01], [Rajski 02] and others.

In CUT modification, test points are inserted [Eichelberger 83] to improve the fault coverage. Observability is enhanced by adding observation points and controllability on a particular node is enhanced by adding a control point. Control points insert AND or OR gates at a node and are activated by pseudo-random values from a dedicated flip-flop during BIST operation. Since test point insertion (TPI) adds extra gates in a design, area and performance overhead are issues. [Krishnamurthy 87] proved that finding optimal locations in circuits with reconvergent fanouts is NP-complete and hence there has been a lot of research on test point insertion techniques. Test point insertion

1550-5774/09 $26.00 © 2009 IEEE
DOI 10.1109/DFT.2009.33

methods based on fault simulation [Briers 86], [Iyengar 89], path tracing [Touba 96], and testability measures [Seiss 91] have been proposed.

In [Yang 09], a method was presented for reducing the area impact of TPI by removing dedicated flip-flops used for driving control points. Test points are inserted with any TPI algorithms and, in a post-processing step, dedicated flip-flops are replaced by functional flip-flops in a design. The replacement technique in [Yang 09] is constrained so that, by construction, it will not introduce any additional timing constraints and achieves basically the same fault coverage as conventional test point insertion using dedicated flip-flops. However, a drawback in [Yang 09] is that some dedicated flip-flops get marked as non-replaceable and thus limit the area overhead reduction that is achieved.

In this paper, a new functional flip-flop selection replacement technique is proposed which provides a new selection algorithm to remove the not-replaced dedicated flip-flops in [Yang 09]. Dedicated flip-flops cannot be replaced when no functional flip-flops satisfy the reconvergence rule and the path inversion parity rule and when control points requires controllability greater than 0.5. In the proposed method, an alternative selection algorithm tries to replace them by functional flip-flops without introducing new timing constraints.

Sec. 2 describes the overview of test point insertion method using functional flip-flops. Sec. 3 describes an alternative selection algorithm. Experimental results are shown in Sec. 4 and conclusions are given in Sec. 5.

2. Overview of Test Point Insertion Using Functional Flip-Flops

This section gives a brief overview of the procedure for test point insertion using functional flip-flops to drive control points that was described in [Yang 09]. Area is generally the main issue for test point insertion. [Yang 09] showed a method for significant area reduction without losing testability and without introducing timing constraints. Dedicated flip-flops for control points are replaced by the functional flip-flops using logic cone analysis that considers the path inversion parity and distance information. To avoid additional timing constraints, the functional flip-flops in the fan-in of the control points are considered as candidate flip-flops for driving the control point because the new paths that are introduced cannot be longer than the existing functional paths by construction. Note, however, that when the dedicated flip-flops are replaced, primitive gates are added to a design and hence the testability may be affected by the circuit modification. The following rules were proposed in [Yang 09] to prevent the testability loss.

1. Illegal reconverence from the candidate flip-flop must be checked - Reconvergence coming from the candidate functional flip-flop in the fanout of a control point can block fault propagation. This needs to be considered and avoided.
2. Opposite path inversion parity must exist along the paths from the functional flip-flop to the control point - Having opposite inversion parity along two paths (the existing functional path from the functional flip-flop to the control point and the new path created through the added test point enable, "TP_Enable", primitive gate for the control point) ensures that a path is testable.

To take the two rules mentioned above into consideration, logic cone analysis is performed starting from the control point and searching the functional flip-flops in its fan-in. Both logical distance and path inversion information are checked. Distance information is used so as not to introduce any delay paths that impact timing. Therefore, the nearest functional flip-

flop from the control point is considered first. As stated in rule 2, the testability can be maintained by strictly having opposite inversion parity along the existing functional path from the flip-flop and along the path through the TP_Enable gate. In the parity inversion analysis, if a flip-flop is found to have multiple existing functional paths with both inversion parities (i.e., one or more with an odd number of inversions, and one or more with an even number of inversions) then that functional flip-flop is discarded from the candidate list for driving the control point. The only functional flip-flops that can be considered as candidates are those for which all existing functional paths to the control point have the same inversion parity. Only for those flip-flops is it possible to satisfy rule 2 by making the newly created path through the TP_Enable gate have opposite inversion parity from the existing functional paths.

The purpose of the TP_Enable gate is that if a functional flip-flop is being used to activate a control point, it may affect the system function during normal operation. So the TP_Enable gate is needed to deactivate the control point at all times during normal operation. The TP_Enable gate can deactivate the control point by always driving one input of the control point gate with a non-controlling value (‘*1*’ for AND gate control point and ‘*0*’ for OR gate control point). This modifies the CUT, and four types of control point structures are used in [Yang 09] to maximize the random pattern testability. Fig. 1 shows the new control point structures which include both control-0 points and control-1 points with both even and odd inversion parity through the TP_Enable gate.

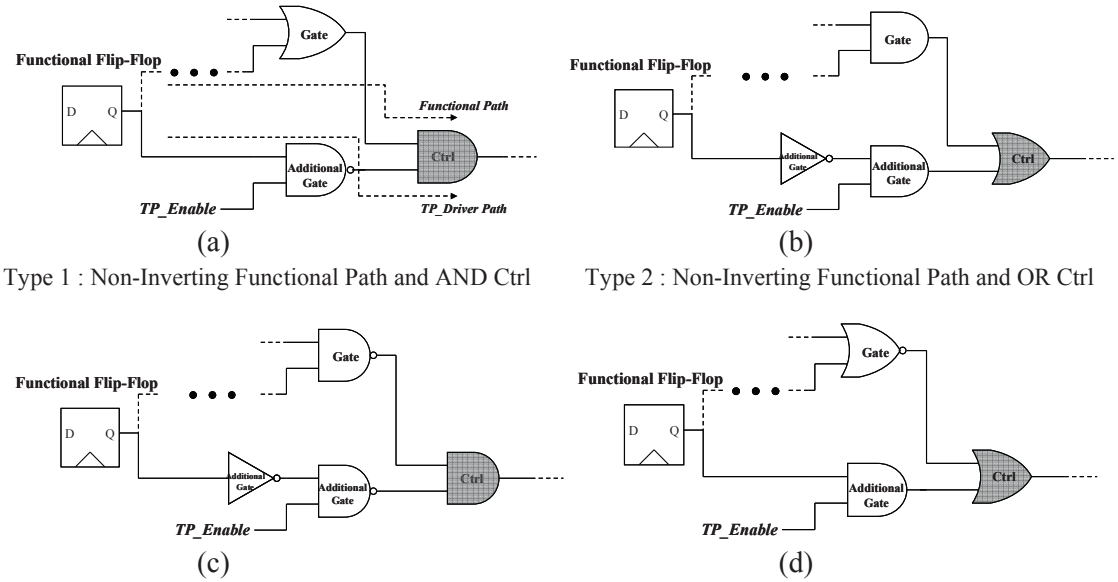

Figure 1. New Types of Control Point Structure for Different Path Inversion Parity

As mentioned, there are two paths existing from a functional flip-flop to a control point. Henceforth, in this paper, the original functional path will be referred as the “*Functional Path*” and the new path created from a functional flip-flop to the control point will be referred as the “*TP_Driver Path*”. In a conventional control point structure, propagation is only enabled when the control point is driven by a dedicated flip-flop with a non-controlling value. However, the random testability is enhanced by the new control point structures in Fig. 1. Having opposite inversion parity between the two paths enables better propagation and helps to reduce the number of test patterns needed to achieve the random testability.

Dedicated flip-flops for the control points cannot be replaced in [Yang 09] when no functional flip-flop meet the two rules listed above. In particular, the opposite path inversion parity rule can be one of the main limiting reasons why some dedicated flip-flops are not replaced. Also note that some test points have only a single functional flip-flop in its fan-in. This happens when the controllability of a certain node needs to be greater than 0.5. In this scenario, a dedicated flip-flop cannot be replaced. These not-replaced flip-flops could still have a large area impact for some designs.

In this paper, a new algorithm is proposed to replace the not-replaced flip-flops in [Yang 09]. The proposed alternative selection algorithm reduces the area impact of test point insertion without introducing any additional delays.

3. Alternative Selection Algorithm

This section describes the new algorithm for addressing the limitations mentioned in the previous section. Depending on the design, it may not be possible for some control points to find flip-flops that have functional paths to the control point that are either all even or all odd parity. Moreover, control points that are placed at nodes in AND and OR trees are more likely to have a skewed value on the functional path, either '*0*' or '*1*', so they have skewed controllability. The algorithm in [Yang 09] cannot replace the dedicated flip-flops in the above cases. The goal with the alternative selection algorithm described here is to relax the rules mentioned in Sec. 2, and be able to find more candidate functional flip-flops that can be used to replace the dedicated flip-flops to achieve the more area reduction.

A dedicated flip-flop for a control point with a single functional flip-flop in its fan-in cannot be replaced in [Yang 09]. Fig. 2 is an example of a small circuit with one control point. This circuit has flip-flops (denoted *A* to *F*) and combinational logic (denoted *G1* to *G6* and *Ctrl*). The control point highlighted in gray (*Ctrl*) is driven by two flip-flops (*A* and *B*) where *A* is a dedicated flip-flop for activating the control point. Because there is only one functional flip-flop in the fan-in of the control point, the dedicated flip-flop *A* cannot be replaced by a functional flip-flop in [Yang 09].

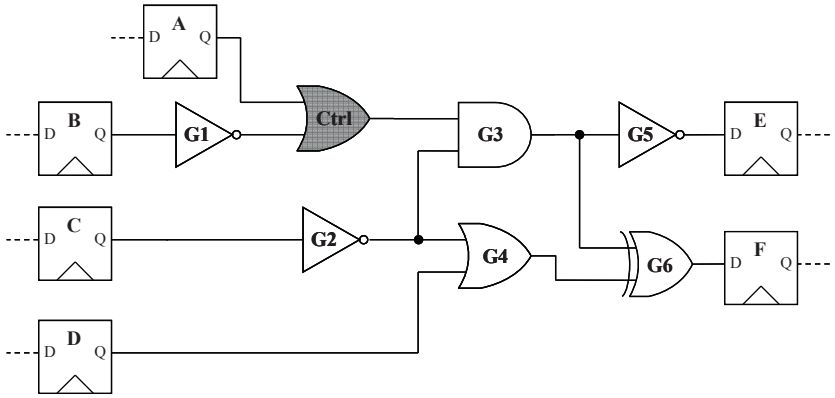

Figure 2. Example of a Circuit with One Control Point (*Ctrl*)

To replace the dedicated flip-flop in Fig.2 where no functional flip-flops are available in the fan-in of test point, an alternative selection algorithm searches additional candidates from outside of test point's fan-in cone. Because the candidate flip-flops are searched outside of the fan-in of the control point, they need to be carefully chosen to avoid introducing new timing constraints.

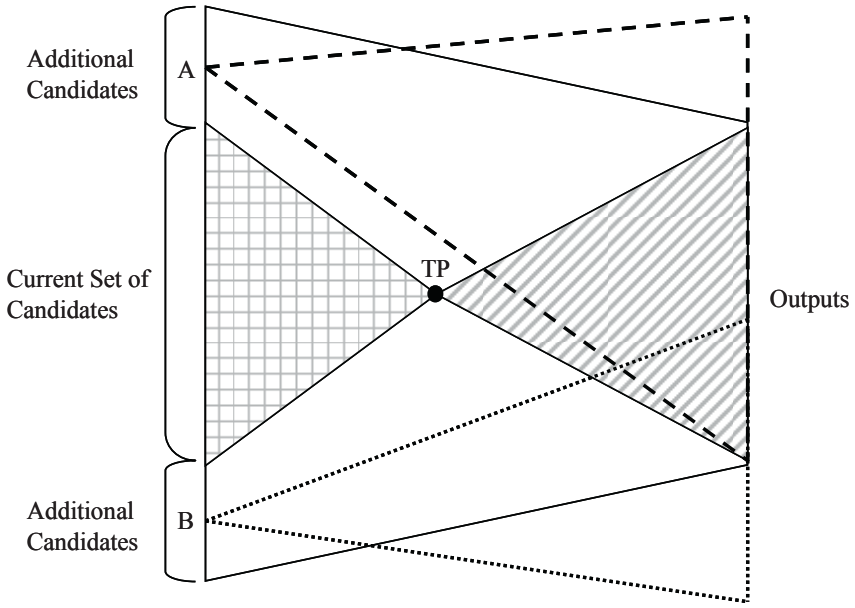

Figure 3. Alternative Selection Algorithm

Fig. 3 shows logic cones in a circuit. Through logic cone analysis starting from the test point, the fan-in cone of the test point can be determined. Functional flip-flops belonging to the fan-in (cross striped logic cone) are the candidates (*Current Set of Candidates*) for replacing a dedicated flip-flop. Since *Current Set of Candidates* belongs to a test point, functional flip-flops in *Current Set of Candidates* will be first considered as candidates. Some dedicated flip-flops may not satisfy the rules in Sec. 2 and are not replaced in [Yang 09]. If rules are relaxed and more candidates are found, there will be higher probability of having more number of dedicated flip-flops replaced. Hence, as long as no redundant faults are introduced, a functional flip-flop which does not meet the rules can be considered for replacing a dedicated flip-flop.

Assume that there are no functional flip-flops available in the current set of candidates for performing the replacement. In this case, the alternative selection algorithm can be used to find possible functional flip-flops that can replace the dedicated flip-flop for the test point. The alternative selection algorithm finds a functional flip-flop that is not in the fan-in of the test point which may add additional timing constraints. Therefore, the selection algorithm needs to guarantee that there is no performance penalty by the proposed method.

The alternative selection algorithm starts from the test point. The nodes are traversed from the test point and the fan-out cone is generated. The fan-out cone is highlighted with dashed lines in Fig. 3. The proposed algorithm tries to find functional flip-flops that are related to the outputs in the test point logic cone. From each output of the fan-out cone, the logic cone analysis generates a fan-in cone. Once all the output fan-in cones are generated, it is possible to list all the inputs which are associated with outputs in the test point's fan-out cone. Some inputs are also found in the test point's fan-in cone while other inputs do not belong to the test point's fan-in cone. In Fig. 3, we can find a parallelogram which is the logic associated with the test point. *Current Set of Candidates* denotes the inputs of the test point's fan-in cone and *Additional Candidates* are the inputs which are not in the test point's fan-in cone, but found from the logic cone analysis from the outputs.

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, functional flip-flop candidates need to be considered in a logic parallelogram. In the example in Fig. 3, since only one functional flip-flop is found in the *Current Set of Candidates* and it is not replaceable, functional flip-flop candidates need to be selected from the set of *Additional Candidates*. There are two types of inputs in *Additional 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* in Fig. 3 has its logic cone drawn with a dashed line and it includes all the outputs of TP fan-out cone. The fan-out logic cone of *B* is marked with a dotted line, and some portion of the outputs belongs to it. Since candidate flip-flops are found outside of the test point's fan-in cone, the main concern is the additional timing constraints in the alternative selection algorithm. 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. Therefore, candidate *B* is not acceptable. Inputs that have all of the test point's fan-out cone's outputs as their outputs can be considered as candidates for the alternative selection and hence candidate *A* is acceptable. However, if there is no functional flip-flop found which covers all the outputs, a dedicated flip-flop cannot be replaced.

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. Instead of tracing back from the test point as in [Yang 09], for each acceptable candidate, propagation from the test point is performed until the overlapped gate element with the test point's fan-out logic cone is first found. 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 [Yang 09].

In Fig. 2, flip-flops *E* and *F* are found in the test point's fan-out cone. First, back tracing from *E* generates a logic cone and inputs *B* and *C* are found. Secondly, *F* fan-in cone has *B*, *C* and *D* as its inputs. Since *B* directly belongs to the control point's logic cone, *Current Set of Candidates* is *B*. *Additional Candidates* are *C* and *D*. The fan-out cones of *C* and *D* are generated and need to be checked whether they cover all the outputs of the control point, *E,* and *F*. *E* is not included by fan-out cone of *D* and hence it is not guaranteed to avoid new timing constraints. However, *C* covers all outputs *E* and *F*, and *G3* is the nearest overlapped element from the test point between *C* and the test point's fan-out cones. Therefore, *C* can be used to replace *A*. If there are multiple candidates found, the flip-flop which has the nearest overlapped element to the test point is used. Fig. 4 shows a circuit when the dedicated flip-flop *A* for the control point in Fig. 2 has been replaced with functional flip-flop *C*.

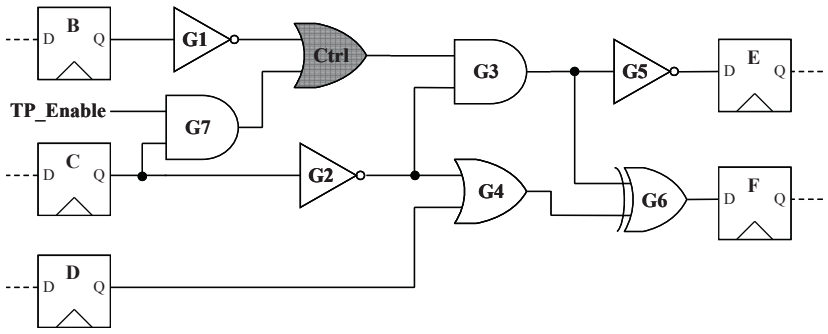

Figure 4. Dedicated Flip-Flop in Fig. 2 Replacement by a Functional Flip-Flop

4. Experimental Results

In this section, experimental results are presented for industrial designs to evaluate the improvements that are obtained through the algorithm proposed here. The LogicVision testpointAnalyze tool [LogicVision] was used to determine the locations of test points in each design. Dedicated flip-flops for driving inserted control points are first replaced by the method in [Yang 09]. Then the proposed algorithm is used to try to increase the number of dedicated flip-flops that can be reduced without impacting delay and with minimizing the loss of testability.

Table 1 shows the number of dedicated flip-flops that are required for driving control points for each of the methods. The first column gives the design name. The number of observation points and control points inserted by the LogicVision tool [LogicVision] are shown in the second and third column. The sum of observation points and control points is the total number of test points inserted in a design. The fourth and fifth columns show the results in [Yang 09]. The number of dedicated flip-flops shows the number of dedicated flip-flops for driving control points that could not be replaced and the number of functional flip-flops denotes the number of dedicated flip-flops that could be replaced. The last two columns show the results using the proposed method. The proposed method is applied on top of [Yang 09] and replaces dedicated flip-flops that are not replaced by [Yang 09].

Table 1. Dedicated Flip-Flop Replacement Comparisons

Design	Conventional TP Insertion		Replacement in [Yang 09]		Proposed Method	
	Observation Points	Control Points	Dedicated Flip-Flops	Functional Flip-flops	Dedicated Flip-Flops	Functional Flip-flops
Design A	70	179	39	140	2	177
Design B	129	371	13	358	8	363

Table 2 compares the 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. For test point area reduction, the following equation is used:

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

As shown with 130nm TSMC technology in [Yang 09], each of the new control points sourced 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 following equation can be used to extrapolate the area reduction with 0.25 as a *k* factor.

Table 2. Improvement Comparisons

Design	Conventional TP Insertion		Replacement in [Yang 09]		Proposed Method	
	Reduction Ratio (%)	Test Point Area Reduction	Reduction Ratio (%)	Test Point Area Reduction	Reduction Ratio (%)	Test Point Area Reduction
Design A	0	0	78.2%	42.2%	98.9%	53.3%
Design B	0	0	96.5%	53.7%	97.8%	54.5%

Table 3 shows a testability comparison with other test point insertion techniques (conventional method and a method in [Yang 09]). The fault coverage is shown when 100000 random patterns are applied. In Design A, most of the dedicated flip-flops are replaced, however, the testability that is achieved is a little lower than the conventional methods. Top-up patterns can be applied to achieve higher testability.

Table 3. Testability Comparison with Faults Information

	Design Method	Design A	Design B
Fault Coverage	Conventional	98.71%	98.25%
	[Yang 09]	98.61%	97.86%
	Proposed	98.09%	97.85%

5. Conclusion

In this paper, a new technique which helps to reduce the area impact of test point insertion is proposed. An alternative selection algorithm replaces the dedicated flip-flops with skewed controllability. Experimental results indicate that significant numbers of dedicated flip-flops are replaced by the proposed method and hence the area impact by a test point insertion can be noticeably reduced. Note that the proposed method can be incorporated with the existing test point insertion algorithms to minimize the area impact with minimizing the loss of random pattern testability.

References

[Bershteyn 93] Bershteyn, M., "Calculation of multiple sets of weights for weighted random testing," *Proc. of International Test Conference*, pp. 1031-1040, 1993.

[Briers 86] Briers, A. J., and K.A.E. Totton, "Random Pattern Testability by Fast Fault Simulation," *Proc. of International Test Conference*, pp. 274-281, 1986.

[Chatterjee 95] Chatterjee, M., and D.K. Pardhan, "A Novel Pattern Generator for Near-Perfect Fault-Coverage," *Proc. of VLSI Test Symposium*, pp. 417-425, 1995.

[Eichelberger 83] 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.

[Hellebrand 92] Hellebrand, S., S. Tarnick, J. Rajski, and B. Courtois, "Generation of Vector Patterns Through Reseeding of Multiple-Polynomial Linear Feedback Shift Register," *Proc. of International Test Conference*, pp. 120-129, 1992.

[Hellebrand 95] 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.

[Iyengar 89] Iyengar, V.S., and D. Brand, "Synthesis of Pseudo-Random Pattern Testable Designs," *Proc. International Test Conference*, pp. 501-508, 1989.

[Jas 01] 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.

[Kapur 94] Kapur, R., S. Patil, T.J. Snethen, and T.W. Williams, "Design of an efficient weighted random pattern generation system," *Proc. of International Test Conference*, pp. 491-500, 1994.

[Konemann 91] Koenemann, B., "LFSR-Coded Test Patterns for Scan Designs," *Proc. of European Test Conference*, pp. 237-242, 1991.

[Konemann 01] Koenemann, B., C. Barnhart, B. Keller, T. Snethen, O. Farnsworth, and D. Wheater, "A SmartBIST variant with guaranteed encoding," *Proc. of VLSI Test Symposium*, pp. 325-330, 2001.

[Krishna 01] Krishna, C.V., A. Jas, and N.A. Touba, "Test Vector Encoding Using Partial LFSR Reseeding," *Proc. of International Test Conference*, pp. 885 – 893, 2001.

[Krishnamurthy 87] Krishnamurthy, B., "A Dynamic Programming Approach to the Test Point Insertion Problem," Proc. of the 24th Design Automation Conference, pp. 695-704, 19877.

[LogicVision] ETAnalysis Tools Reference Manual, software version 6.0a, 2007

[Pomeranz 92] Pomeranz, I., and S.M. Reddy, "3-Weight Pseudo-Random Test Generation Based on a Deterministic Test Set for Combinational and Sequential Circuits," IEEE Transactions on Computer-Aided Design, Vol. 12, No. 7, pp. 1050-1058, Jul. 1993.

[Rajski 02] 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.

[Seiss 91] Seiss, B.H., P. Trouborst, and M. Schulz, "Test Point Insertion for Scan-Based BIST," *Proc. European Test Conference*, pp. 253-262, 1991.

[Tamarapalli 96] Tamarapalli, N., and J. Rajski "Constructive Multi-Phase Test Point Insertion for Scan-Based BIST," *Proc. of International Test Conference*, pp. 649-658, 1996.

[Touba 95a] Touba, N.A., and E.J. McCluskey, "Transformed Pseudo-Random Patterns for BIST," *Proc. of VLSI Test Symposium*, pp. 410-416, 1995.

[Touba 95b] Touba, N.A., and E.J. McCluskey, "Synthesis of Mapping Logic for Generating Transformed Pseeudo-Random Patterns for BIST," *Proc. International Test Conference*, pp. 674-682, 1995.

[Touba 96] Touba, N.A., and E.J. McCluskey, "Test Point Insertion Based on Path Tracing," *Proc. of VLSI Test Symposium*, pp. 2-8, 1996.

[Yang 09] Yang, J.-S., B. Nadeau-Dostie, and N.A. Touba, "Test Point Insertion Using Functional Flip-Flops to Drive Control Points," *Proc. of International Test Conference*, 2009.