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