Reduced Complexity Test Generation Algorithms for Transition Fault Diagnosis*

Yu Zhang and Vishwani D. Agrawal
Auburn University, Department of Electrical and Computer Engineering, Auburn, AL 36849, USA
yuzhang@auburn.edu, vagrawal@eng.auburn.edu

Abstract–To distinguish between a pair of transition faults, we need to find a test vector pair (LOC or LOS type) that produces different output responses for the two faults. By adding a few logic gates and one modeling flip-flop to the circuit under test (CUT), we create a diagnostic ATPG model usable by a conventional single stuck-at fault test pattern generator. Given a transition fault pair, this ATPG model either finds a distinguishing test or proves the faults to be equivalent. An efficient diagnostic fault simulator is devised to find undistinguishable fault pairs from a fault list by a test vector set. The number of fault pairs that needs to be targeted by the ATPG is greatly reduced after diagnostic fault simulation. We use a previously proposed diagnostic coverage (DC) metric to determine the distinguishability (diagnosability) of a test vector set. Experimental results show improved DC for benchmark circuits after applying the proposed diagnostic ATPG algorithms.

1 Introduction

A recent paper [16] describes a system for generating diagnostic tests using the single stuck-at fault model. Main ideas introduced there were a definition of diagnostic coverage and algorithms for diagnostic simulation and exclusive test [1] generation. In that work emphasis was placed on using the existing tools that were originally designed for fault detection only.

The present work provides a similar capability for the diagnosis of transition faults. A reader will find these extensions to be non-trivial. In this work we emphasize the use of existing tools and techniques to allow implementation of the new algorithms. The basic tools used are the simulation and test generation programs for detection of single stuck-at faults. Scan test environment is assumed in which both launch-off-capture (LOC) and launch-off-shift (LOS) types of tests can be conducted. It is well known that a transition fault test contains a pair of vectors applied to the combinational part of the circuit [4]. In scan testing, as the first of these vectors is scanned in it appears at the input of the combinational logic. The second vector is then either produced by clocking the circuit in the normal mode (known as launch-off-capture or LOC test) or in the scan mode (launch-off-shift or LOS test), following which the response is captured in scan register always in the normal mode and scanned out in the scan mode.

Our principal objective in this work is not just to test for transition faults as is normally done, but to distinguish the exact fault that may have caused a failure. We borrow a diagnostic coverage (DC) metric from a previous paper [16]. The new modeling techniques and algorithms for transition faults are more efficient than those published before [7]. Our implementation and results support the practicality of the presented approach.

2 Background and Motivation

The usefulness of the transition fault model stems from the fact that modern VLSI devices must be tested for performance. Transition faults are not perfect and in fact may not represent many of the actual defects. Their acceptability, like that of stuck-at faults, is due to several practical reasons. To name a few, their number grows only linearly with circuit size, they require two-pattern tests that are essential for detecting delay and other non-classical faults, and the scan methodology can be adapted to test them.

Why is the diagnosis of transition faults important? The same technology advances that give us lower cost and higher performance make it necessary that we diagnose delay defects. Presently, we must rely on ad-hoc measures like at-speed testing, N-detect vectors, etc. N-detect tests have shown promising ability to detect/diagnose non-classical faults [11]. By increasing N, we expect that more faults, including some timing related faults, can be captured. But the number of test patterns also grows with increasing N, which leads to longer test time and/or higher cost. Methods have been proposed [8, 9, 14] for compaction of N-detect vector

*This research is supported in part by the National Science Foundation Grant CNS-0708962.

set. These require computational effort. The present work is aimed at providing a similar diagnostic capability for transition faults as is available for stuck-at faults [1, 5, 15, 16] without increasing the ATPG effort while also keeping test pattern count down.

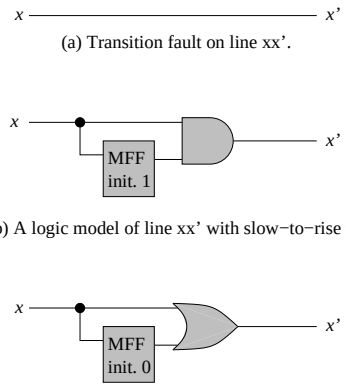
A previous attempt [7] at generating diagnostic tests for transition faults used a complex model requiring up to four copies of the combinational circuit. This increases the complexity of the ATPG, which is already known to be an NP-complete problem [4]. Besides, that method is demonstrated to work only for LOC tests. The ATPG model presented here requires only one copy of the circuit although the model is mildly sequential. The sequential behavior comes from one or two modeling flip-flops that are added to the circuit. These flip-flops are pre-initialized and hence do not increase the ATPG complexity. The new procedure is equally suitable for both LOC and LOS tests. It is also more flexible since both sequential and combinational ATPG can be used. For combinational ATPG the CUT needs to be unrolled into two time frames. It is still more efficient compared to [7] because only two, and not four, copies of CUT are required. SAT based diagnostic test generation procedures have been proposed [3] for the transition fault model. A SAT formulation may be too complex for large designs.

3 Modeling a Transition Fault

Because we wish to use the existing methods that are based on the logic description of the fault condition, we will use a synchronous model for the transition fault. Figure 1 shows a method of modeling a single slow-to-rise or slow-to-fall transition fault on a line xx' in the combinational logic with a synchronous sequential circuit. The shaded elements are inserted for modeling the fault and do not belong to the actual circuit. The modeling flip-flop (MFF) is initialized to the value shown. Consider the slow-to-rise fault in Figure 1(b). Flip-flop initialization to 1 ensures that the output x' on the line will be the correct logic value on the first vector. Of the four two pattern sequences on x, 00, 01, 10 and 11, all except 01 will produce the correct output at x'. The sequence 01 at x will appear as 00 at x', correctly representing a slow-to-rise transition fault on line xx'. Figure 1(c) shows a similar model for a slow-to-fall transition fault on line xx'.

4 An ATPG Model

An ATPG (automatic test pattern generation) model is a netlist of the circuit under test (CUT), modified to represent the target fault as a stuck-at fault.



(a) Transition fault on line xx'.

(b) A logic model of line xx' with slow−to−rise fault.

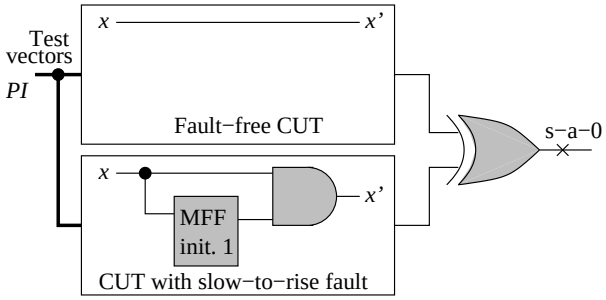
(c) A logic model of line xx' with slow−to−fall fault.

Figure 1. Modeling a faulty circuit with a single transition fault on line xx'.

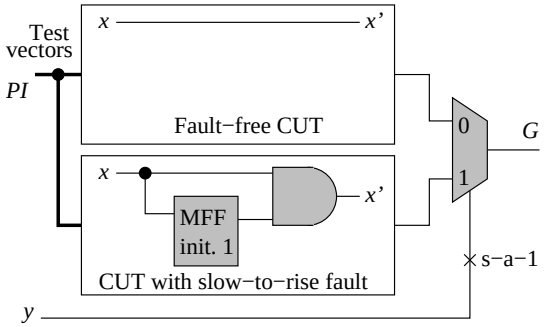
The modification amounts to insertion of a few logic elements for modeling only. For a transition fault, we construct a single stuck-at fault ATPG model as shown in Figure 2. The ATPG model of Figure 2(a) gives the conventional Boolean satisfiability formulation. Note that a 1 output from the EXOR gate cannot be obtained by a single vector. Because the modeling flip-flop MFF is initialized to 1, initially, $x' = x$. To produce a different output from the faulty circuit, the first vector must set $x = 0$ and then a second vector should set $x = 1$, besides sensitizing a path from x' to the primary output (PO) or an observable point.

The ATPG model of Figure 2(b) can be used in the same way [16]. Any test sequence for either s-a-0 or s-a-1 fault on y must produce different outputs from the fault-free and faulty circuits. The advantage of this model is that it can be simplified to use a single copy of the circuit which is an improvement over the previous work [7]. The analysis that leads to the ATPG model of Figure 3 is the same as given in a recent paper [16]. There a single-copy ATPG model was obtained for finding an exclusive test for a pair of stuck-at faults. Thus, the fault-free CUT in Figures 2 (a) and (b) was replaced by the CUT containing one of the faults.

The main idea that allows us to collapse the two copies of the circuit in Figure 2(b) into a single copy is the realization that the two circuits are almost identical. The only difference is at the faulty line. It can be shown [16] that the multiplexer at PO can be moved to the fault site. The procedure is as follows: Suppose a transition fault is to be detected on a signal interconnect from x (source) to x' (destination). In a single copy of the circuit, the source signal x is made to fan out as two signals $x1$ and $x2$. Fanout $x1$ is left as fault-free signal. The other fanout $x2$ is modified according



(a) An ATPG model: test for output s−a−0 fault detects the slow−to−rise fault on xx'.



(b) An alternative ATPG model: test for stuck−at fault on y detects the slow−to−rise fault on xx'.

Figure 2. ATPG model with a stuck-at fault for detecting a slow-to-rise fault on line xx'.

to Figure 1 to produce the faulty value. These two signals $x1$ and $x2$ are applied to the two data inputs of a multiplexer whose output x' now feeds into all destinations of the original x', and whose control input is a new PI y. The target fault now is any stuck-at fault (s-a-0 or s-a-1) on y. Any test for this target fault must produce different values at fault-free $x1$ and the faulty $x2$ while propagating the value of x' to a PO, and hence must detect the fault modeled by $x2$. This ATPG model for a slow-to-rise fault on xx' is shown in Figure 3. Any test for y s-a-0 or for y s-a-1 in the ATPG model of Figure 3 will always contain two vectors. The model for a slow-to-fall transition fault is obtained by replacing the AND gate by an OR gate and changing the initialization of the flip-flop to 0, as shown in Figure 1(c). The gate and multiplexer combination can be further simplified to an equivalent ATPG model given in Figure 4, which shows the ATPG models for both slow-to-rise and slow-to-fall transition faults.

5 Combinational and Scan Circuits

The above fault modeling procedure is valid for both combinational and scanned sequential circuits. For a combinational circuit under test (CUT), the modeling

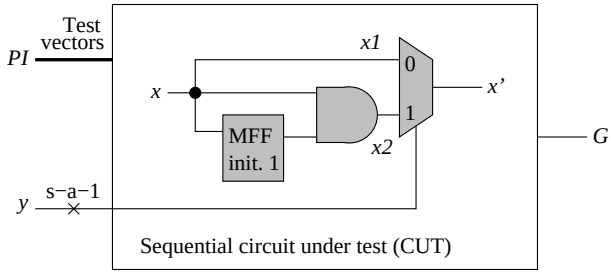
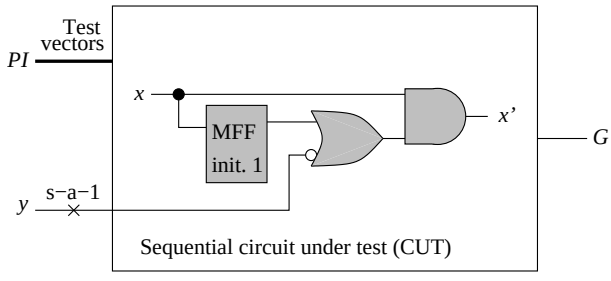
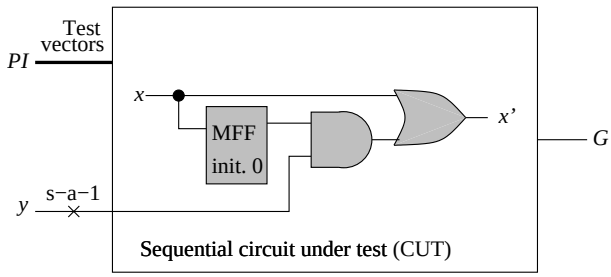


Figure 3. A single circuit copy ATPG model in which a test for a stuck-at fault on y detects the slow-to-rise fault on line xx'.



(a) Slow−to−rise transition fault on line xx'.



(b) Slow−to−fall transition fault on line xx'.

Figure 4. Simplified single circuit copy ATPG models in which a test for a stuck-at fault on y detects a transition fault on line xx'.

flip-flop (MFF) serves two purposes. First, it requires a two-vector test. Second, the initial state of the flip-flop makes it impossible to activate the fault effect at x' in the first vector. This model can be used to generate a two-vector test either by a sequential ATPG program or by a combinational ATPG program applied to a two time-frame expansion of the circuit.

For a scanned sequential circuit under test (CUT) the ATPG models of the previous section will also generate two-vector tests. The vectors can be generated either by a scan ATPG program in the partial scan mode to accommodate the modeling flip-flop (MFF) or by a combinational ATPG program. The second vector is generated either as a launch-off-capture (LOC) sequence or as a launch-off-shift (LOS) sequence.

Table 1. Transition fault diagnostic test generation for ISCAS'89 benchmark circuits. Circuits have full scan and tests are generated for application in LOC mode.

Circuit	No. of faults	Detection test generation					Diagnostic test generation			
		Detect tests	FC %	DC %	Undiag. fault groups	Largest group	Exclusive tests	DC %	Undiag. fault groups	Largest group
s27	46	11	100.0	52.2	12	7	18	97.8	1	2
s298	482	44	79.9	62.4	62	5	34	70.1	39	4
s382	616	51	80.8	64.1	82	4	24	68.5	58	4
s1423	2364	102	92.9	79.3	280	5	106	84.1	182	5
s5378	6589	205	91.2	82.0	400	9	472	90.0	85	7
s9234	10416	377	92.8	75.8	1219	11	597	82.1	754	8
s13207	14600	480	89.1	70.0	1707	20	543	74.1	1392	11
s15850	17517	306	87.6	71.2	1961	9	486	74.3	1565	7
s35932	52988	75	99.0	88.3	3737	6	725	90.2	2867	4
s38417	47888	244	98.4	87.5	4090	9	1336	91.0	2883	8
s38584	56226	395	95.7	86.7	4042	8	1793	90.3	2440	7

6 Diagnostic Test Generation

The main contribution of previous sections is in modeling of a transition fault as a single stuck-at fault. The benefit of this model is that we can use the tools and techniques available for single stuck-at faults. We now illustrate the use of the following techniques discussed in a recent paper [16] for transition faults:

1. Diagnostic coverage (DC) of tests that provides a quantitative measure for their ability to distinguish between any pair of faults.

2. Diagnostic fault simulator that determines DC for any given set of vectors and identifies undistinguished fault pairs. This diagnostic fault simulator internally uses any conventional single stuck-at fault simulator.

3. Exclusive test generator that derives an exclusive test for a fault pair such that the two faults in the pair can be distinguished from each other. If an exclusive test is found to be impossible then the two faults are equivalent and one of them can be removed from the fault list to further collapse it. This exclusive test generator internally uses a conventional single stuck-at fault test generator.

4. A complete diagnostic test generation system that first generates the conventional tests for fault detection coverage, determines the DC of those tests, and then generates more vectors if necessary to enhance DC.

The results of these procedures when applied to transition faults are shown in Table 1, which gives two types of coverages [16]. For a set of vectors we group faults such that all faults within a group are not distinguishable from each other by those vectors, while each fault in a group is pair-wise distinguishable from all faults in every other group. This grouping is similar to equivalence collapsing except here grouping is conditional to the vectors. If we generate a new vector that detects a subset of faults in a group then that group is partitioned into two groups, one containing the detected subset and the other containing the rest. For multi-output circuit, the targeted group may be divided into more than two sub groups. Suppose, we have sufficient vectors to distinguish between every fault pair, then there will be as many groups as faults and every group will have just one fault. Prior to test generation all faults are in a single group we will call g_0. As tests are generated, detected faults leave g_0 and start forming new groups, g_1, g_2, . . . g_n, where n is the number of distinguishable fault groups. For perfect detection tests g_0 will be a null set and for perfect diagnostic tests, $n = N$, where N is the total number of faults. We define *diagnostic coverage, DC*, as

$$DC = \frac{Number\ of\ detected\ fault\ groups}{Total\ number\ of\ faults} = \frac{n}{N} \qquad (1)$$

Initially, without any tests, $DC = 0$, and when all faults are detected and pair-wise distinguished, $DC = 1$. Also, the numerator in equation 1 is the number of fault dictionary syndromes [4] and the reciprocal of DC is the *diagnostic resolution* (DR) [1]. The detection fault coverage (FC) is given by,

$$FC = \frac{Number\ of\ detected\ faults}{Total\ number\ of\ faults} = \frac{N - |g_0|}{N} \qquad (2)$$

We used Fastscan [12] to generate fault detection tests for transition faults. Fastscan can generate transition fault tests for full-scan circuits in either of the

two (LOC and LOS) modes. The results of Table 1 are for LOC mode only. The equivalent circuits of Figure 4 provide an alternative method. Here the target transition fault is represented as a single stuck-at fault. The modeling flip-flop MFF starts with a specified initial state and is not scanned. Thus, Fastscan generates a test for a single stuck-at fault y s-a-1 in the partial scan mode; all normal flip-flops of the circuit are scanned and the modeling flip-flop MFF is not scanned. All flip-flops including MFF are assumed to have the same clock. Because of the initial state of the unscanned MFF, the fault cannot be detected by the first vector, which merely initializes the circuit. The test consists of two combinational vectors, i.e., a scan-in sequence, followed by one clock in normal mode (LOC) or in scan mode (LOS), capture, and a scan-out sequence.

The second column of Table 1 lists the number of transition faults. Faults on same fanout free interconnect and the input and output of a not gate are collapsed [12]. Also some of the redundant transition faults are identified during ATPG and they are removed. The third column lists the number of LOC tests. Note that Fastscan can perform test pattern compaction. Since in this work our focus is on the ATPG algorithm, we did not perform compaction on diagnostic test patterns. Each test consists of a scan-in, capture and scan-out sequence. The detection fault coverage (FC) of transition faults is given in column 4. Reasons of less than 100% FC are (a) aborted ATPG runs, (b) LOS mode not used, and (c) redundancy or untestability not identified. Because Fastscan for transition faults operates in sequential mode it often fails to identify redundancies. In our ongoing work we are developing a combinational two time-frame model mentioned in Section 5 to improve the fault efficiency. Based on observations made from several small ISCAS'89 circuits through detailed structural analysis we find that most aborted pairs are actually functionally equivalent. If all fault equivalences are identified, similar to fault efficiency, "diagnostic efficiency" would be much higher than diagnositic coverage. Note that FC can be considered as an upper bound on DC and fault efficiency is then an upper bound on "diagnostic efficiency". The experimental results from [7] show that all targeted fault pairs are either distinguished or identified as equivalent pairs. Using two-time-frame expasion we should get similar results. This needs further investigation.

Column 5 of Table 1 gives the diagnostic coverage (DC) obtained from diagnostic fault simulation [16], which divides faults into groups. Group g_0 contains undetected faults. Groups with more than one fault contain the faults that are mutually undistinguished (or undiagnosed). Thus, circuit s27 has 12 such groups and the largest of those groups has 7 faults (see columns 6 and 7). Similarly, s5378 has 400 multi-fault undiagnosed groups, largest one containing 9 faults. In [7] instead of targeting all transition faults they randomly choose 1000 faults and extract those pairs that cannot be distinguished by a detection test set. These pairs are then serves as the starting set for their diagnostic test generation flow. For s38584 the CPU time reported in [7] is 174649 seconds whereas ours is 14841 seconds for the same circuit. Note that CPU times are used here only for rough comparison (hardware configuration is not reported in [7]).

The purpose of diagnostic test generation is to derive exclusive tests that will provide pair-wise diagnosis of faults within groups. This is done by modeling a pair of transition faults as two stuck-at faults using the technique of Figure 4 and then using a single stuck-at fault representation for those two faults [16]. Additional tests obtained for fault pairs formed within each multi-fault group are listed in column 8 of Table 1. The corresponding diagnostic coverage (DC) is given in column 9. For example, 18 tests were generated for s27 raising DC to 97.8%. There was only one undiagnosed fault group left (column 10) and it contained two faults (column 11).

The two undiagnosed fault of s27 are shown in Figure 5. Using a two time-frame combinational model, we determined that these two faults cannot be distinguished by any LOC test. Because the functional operation of the circuit is constained to a subset of conditions allowed during the LOC testing, these two faults can be considered functionally equivalent. That will make $DC = 100\%$ in column 9. At the present stage of our investigation such fault equivalence checking is not automated. Once we have enhanced such capability, we hope to analyze the undiagnosed fault groups in column 10 for all circuits. We have verified that with an added LOS test all faults in s27 can be diagnosed.

7 Conclusion

The stuck-at fault models of transition faults presented here are significantly more compact than those previously published [7]. Combined with diagnostic fault simulation and test generation algorithms similar to those for stuck-at faults [16], the new transition fault models provide potentially very effective ATPG methodology. Delay fault diagnosis is important in the characterization of modern VLSI devices and a high diagnostic coverage of tests is desirable. Whether or not the tests have an adequate diagnostic coverage cannot be ascertained unless we have an effective tool for identifying fault equivalence [2, 6, 13]. The present work

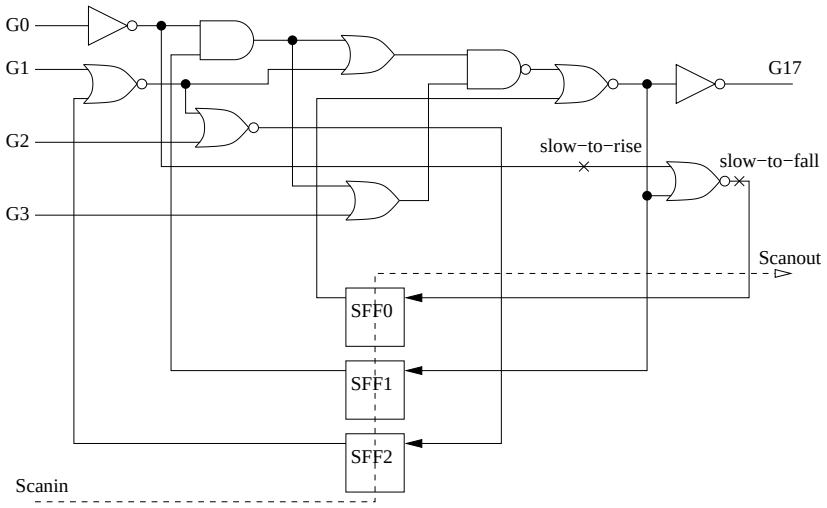


Figure 5. Circuit s27 showing the fault pair left undiagnosed after simulation of 11+18 LOC tests of Table 1. These faults are functionally equivalent though they are diagnosable in the LOS mode.

provides the possibility of doing so entirely by combinational ATPG programs. Our ongoing research is exploring this aspect. That will give transition fault testing the same level of maturity as enjoyed by stuck-at faults. Another direction for the future is to make the diagnostic tests specific to small delay defects (SDD), i.e., derive tests to detect transition faults each through the longest sensitizable path through its site [10].

References

[1] V. D. Agrawal, D. H. Baik, Y. C. Kim, and K. K. Saluja, "Exclusive Test and its Applications to Fault Diagnosis," in *Proc. 16th International Conf. VLSI Design*, Jan. 2003, pp. 143–148.

[2] M. E. Amyeen, W. K. Fuchs, I. Pomeranz, and V. Boppana, "Fault Equivalence Identification in Combinational Circuits Using Implication and Evaluation Techniques," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 7, July 2003.

[3] N. K. Bhatti and R. D. Blanton, "Diagnostic Test Generation for Arbitrary Faults," in *Proc. International Test Conf.*, 2006. Paper 19.2.

[4] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits*. Boston: Springer, 2000.

[5] T. Gruning, U. Mahlstedt, and H. Koopmeiners, "DIATEST: A Fast Diagnostic Test Pattern Generator for Combinational Circuits," in *Proc. IEEE International Conf. Computer-Aided Design*, 1991, pp. 194–197.

[6] I. Hartanto, V. Boppana, and W. K. Fuchs, "Diagnostic Fault Equivalence Identification Using Redundancy Information & Structural Analysis," in *Proc. International Test Conf.*, 1996, pp. 20–25.

[7] Y. Higami, Y. Kurose, S. Ohno, H. Yamaoka, H. Takahashi, Y. Takamatsu, Y. Shimizu, and T. Aikyo, "Diagnostic Test Generation for Transition Faults Using a Stuck-at ATPG Tool," in *Proc. International Test Conf.*, 2009. Paper 16.3.

[8] K. R. Kantipudi and V. D. Agrawal, "A Reduced Complexity Algorithm for Minimizing N-Detect Tests," in *Proc. 20th International Conf. VLSI Design*, Jan. 2007, pp. 492–497.

[9] X. Kavousianos and K. Chakrabarty, "Generation of Compact Stuck-At Test Sets Targeting Unmodeled Defects," *IEEE Trans. Computer-Aided Design*, vol. 30, no. 5, pp. 787–791, May 2011.

[10] A. K. Majhi, V. D. Agrawal, J. Jacob, and L. M. Patnaik, "Line Coverage of Path Delay Faults," *IEEE Trans. VLSI Systems*, vol. 8, pp. 610–614, Oct. 2000.

[11] E. J. McCluskey and C.-W. Tseng, "Stuck-Fault Tests vs. Actual Defects," in *Proc. International Test Conf.*, 2000, pp. 336–342.

[12] Mentor Graphics, *FastScan and FlexTest Reference Manual*, 2004.

[13] R. K. K. R. Sandireddy and V. D. Agrawal, "Diagnostic and Detection Fault Collapsing for Multiple Output Circuits," in *Proc. Design, Automation and Test in Europe (DATE'05)*, Mar. 2005, pp. 1014–1019.

[14] M. A. Shukoor and V. D. Agrawal, "A Two Phase Approach for Minimal Diagnostic Test Set Generation," in *Proc. 14th IEEE European Test Symp.*, May 2009, pp. 115–120.

[15] X. Yu, M. E. Amyeen, S. Venkataraman, R. Guo, and I. Pomeranz, "Concurrent Execution of Diagnostic Fault Simulation and Equivalence Identification During Test Generation," in *Proc. 21st IEEE VLSI Test Symp.*, 2003, pp. 351–356.

[16] Y. Zhang and V. D. Agrawal, "A Diagnostic Test Generation System," in *Proc. International Test Conf.*, 2010. Paper 12.3.