At-Speed Test for Path Delay Faults Using Practical Techniques

Wangqi Qiu Jing Wang D. M. H. Walker

Dept. of Computer Science
Texas A&M University
College Station TX 77843-3112
Tel: (979) 862-4387
Fax: (979) 847-8578
Email: {wangqiq, j0w2111, walker}@cs.tamu.edu

Abstract

To detect the smallest delay faults at a fault site, the longest path(s) through it must be tested at full speed. Existing test generation tools are either inefficient in automatically identifying the longest testable paths due to the high computational complexity or do not support at-speed test using existing practical design-for-testability structures, such as scan design. In this work a fast test generation methodology for scan-based synchronous sequential circuits is presented, under two at-speed test strategies used in industry. The two strategies are compared and the test generation efficiency is evaluated on the ISCAS89 benchmark circuits.

1. Introduction

Delay test has been researched for many years. At-speed test, which significantly increases the delay fault coverage, is used in industrial applications. The transition fault model, which is the simplest delay fault model, is usually used in these applications. However, the transition fault model targets the large delay faults which cause all the sensitizable paths through the fault site to be slow. Recent research shows that resistive opens are one of the major defect types which cause delay faults [1], and that small delay faults cannot be neglected [2]. To detect the smallest delay fault at a fault site, the longest sensitizable path through it must be tested. But (longest) path delay fault test generation is much more expensive than transition fault test generation, because a transition test can be composed by pairing stuck-at-0 and stuck-at-1 vectors [3] and the transition fault test generation for sequential circuits has been extensively researched [4].

Recently some research significantly decreased the cost of path delay fault test generation [5][6] and these methodologies are able to integrate some path selection criteria, such as the longest path through each line. However, they assume the circuits are combinational, i.e. there is no dependence between the two test vectors or between two bits within a vector. Thus, if these methodologies are applied to sequential circuits, it must be assumed that the circuits are full scan and the first vector is stored in the flip-flops when the second vector is scanned in. Because such a scan design requires more silicon area and introduces extra delay, compared to a muxed scan design, it is rarely used in industry. Therefore, a new automatic test pattern generation (ATPG) tool for path delay faults in sequential circuits has to be developed, to target commonly-used design-for-testability (DFT) structures, such as muxed scan.

The ATPG was developed by extending a path generation algorithm for combinational circuits [6] to handle scan-based synchronous sequential circuits. The K longest paths through each line for both slow-to-rise and slow-to-fall faults on the line are generated. Two at-speed test approaches, "launch-on-shift" and "launch-on-capture", are used as the constraints from the scan design. This results in sequential false paths [7] which are combinational testable. The delays of the longest combinational and sequential testable paths through each line are compared in the experiments.

The remainder of the paper is organized as follows. Section 2 introduces the two practical test approaches to apply at-speed test in a scan-based circuit. Section 3 describes the test generation algorithm using the two approaches. Section 4 includes experimental results on the ISCAS89 benchmark circuits and Section 5 concludes with directions for future research.

2. Scan-Based At-Speed Test Approaches

In low-cost automatic test equipment (ATE), the test speed is usually much slower than the functional speed of the circuit under test. This is not a problem to detect stuck-at and large delay faults, but small delay faults may escape. Therefore at-speed test is preferred to increase the realistic delay fault coverage.

However, due to the low-cost ATE speed limitation, the at-speed tests primarily in use in industry are built-in self-test (BIST) and AC scan. Evidence has shown that BIST can achieve very high fault coverage for stuck-at and transition faults [8], but it has low probability to sensitize enough critical paths, e.g. the longest path through each line. On the other hand, functional tests running at full speed are becoming unattractive due to the high cost of development and application [9]. Therefore, this paper focuses on high-quality delay test generation using existing scan designs.

In this paper the muxed data scan design is assumed, with a scan enable signal selecting either serial scan data or circuit data. The flip-flops are clocked with the system clock. Two scan-based at-speed test methodologies, which have found increasing usage in industry, will be briefly introduced in the next two sections.

2.1. Launch-on-Shift (Skewed Load)

The procedure of the launch-on-shift (or skewed load) test approach is:

1. The circuit is set to scan mode. The first test vector is scanned into the scan chains using the slow scan clock, and the values are set on primary inputs (PIs).
2. The second test vector is obtained by shifting the scan chain by one bit. Usually the PIs do not change values due to the constraints from low-cost ATEs.
3. The circuit is set to the functional mode by flipping the scan-enable signals and pulsing the system clock to capture the circuit values in the flip-flops. The values on primary outputs (POs) are captured if necessary.
4. The circuit is set to the scan mode and the values in the scan chains are scanned out using the slow scan clock. This step can be overlapped with step 1.

The advantage of this approach is that fast test generation methodologies for combinational circuits can be applied without many modifications. Scanned flip-flops are considered primary inputs in the ATPG for combinational circuits, so the new constraints on these "primary inputs" must be added to the existing ATPG.

The disadvantage of this approach is that the scan enable signals must operate at full speed. In addition, a large percentage of the sensitizable paths under the launch-on-shift constraints are sequential false paths, i.e. these paths are not sensitizable in functional mode, so some redundant faults would be detected.

2.2. Launch-on-Capture (Functional Justification)

The procedure of the launch-on-capture (or functional justification, broadside) test approach is:

1. Same as the launch-on-shift approach step 1.
2. The circuit is set to functional mode. A dummy cycle is inserted if the scan-enable signal cannot operate at full speed or the system clock frequency is very high, so that the launch clock pulse width is too large. Figure 1(a) shows the clock waveform. For comparison, Figure 1(b) shows the clock waveform if the time is sufficient for the scan enable signal to propagate. In this approach, the launch cycle is kept identical to the shift cycle with respect to period, rising edge, and pulse width.
3. The system clock is pulsed twice. At the first clock pulse, the second test vector is derived from the first test vector. At the second clock pulse, the at-speed test is performed and the output values are captured in the scanned flip-flops. The values on POs are captured if necessary.
4. Same as the launch-on-shift approach step 4.

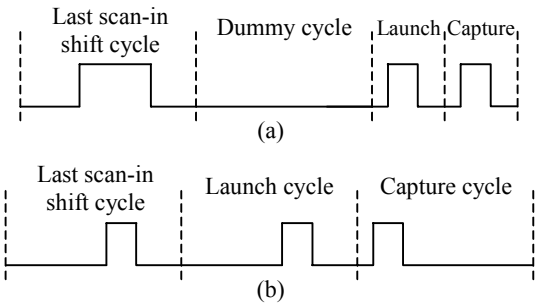

Figure 1. Launch-on-capture clock waveforms.

The advantage of this approach is that it does not require the scan enable signal to operate at full speed. And the sensitizable paths under the launch-on-capture constraints are also sensitizable in functional mode, unless the first vector represents an illegal state.

Though the launch-on-capture approach is more promising and practical for industrial use [9], the launch-on-shift approach is also included in this work because it may have lower data volume, it may detect some delay faults that are not functional, and test generation only requires combinational test.

3. Test Generation

The test generation algorithm was developed from a fast ATPG for combinational circuits [6]. In this section, the path generation engine is introduced, and the constraints from the launch-on-shift/capture approach are applied to eliminate sequential false paths from the combinational testable path set, and the time frame expansion method is used for the launch-on-capture approach.

3.1. Path Generation Engine

Figure 2 is the algorithm used in the path generation engine [6]. In this paper, a *launch point* (of a path) is a primary input or scanned flip-flop, and a *capture point* is a primary output or a scanned flip-flop. In the preprocessing phase, the maximum structural distance from each gate to capture points is computed, without considering any logic constraint. This value is termed the *PERT delay*. In the path generation phase, *partial paths* are initialized from launch points. A partial path is a path which originates from a launch point but has not reached any capture point. A value called *esperance* [10] is associated with a partial path. The esperance is the sum of the length of the partial path and the PERT delay from its last node to a capture point. In other words, the esperance of a partial path is the upper bound of its delay when it becomes a *complete path*, which reaches a capture point.

In each iteration of the path generation phase, the partial path with the maximum esperance value is extended by adding one gate. If the last gate of the partial path has more than one fanout, the partial path splits. Then the constraints to propagate the transition on the added gate, such as non-controlling side input values, are applied.

Direct implications [10] are then used to propagate the constraints throughout the circuit. A direct implication on a gate is one where an input or output of that gate can be directly determined from the other values assigned to that gate. Figure 3 shows some examples of direct implications on an AND gate. The values in boxes are implied from the existing values. If there are any conflicts, the whole search space which contains the partial path is trimmed off. If the partial path does not reach a capture point, some false path elimination techniques [6] are applied to prevent it from growing to a false path. Then its esperance value is updated and it is inserted back into the partial path store. If a partial path becomes a complete path, final justification is performed to find a vector. Details of final justification are provided in Section 3.3. This process repeats until enough longest testable paths are generated.

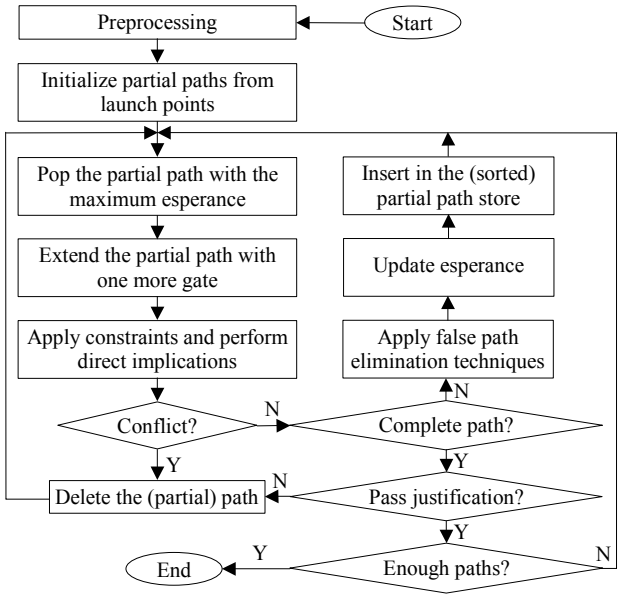

Figure 2. Path generation algorithm.

In this work, the goal is to generate the K longest path through each line for both slow-to-rise and slow-to-fall faults. So the search space is limited to the fan-in and fan-out cones of the fault site and the test generation does not stop until both faults are detected.

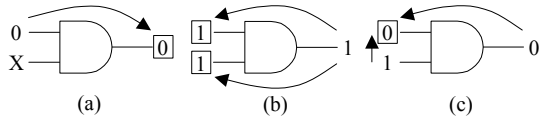

Figure 3. Examples of direct implications [6].

3.2. Implications on Scanned Flip-Flops

Direct implications can be performed on scanned flip-flops as well as regular gates to detect most local conflicts and eliminate sequential false paths. Since local conflicts are the fundamental reason for false paths in most circuits [10], performing direct implications as much as possible can identify most false paths and significantly speed up the test generation process.

If the launch-on-shift approach is used, the logic values on neighboring scanned flip-flops are dependent on each other. For example, in Figure 4, the logic value of cell *A* in the first vector is as same as that of cell *B* in the second vector. The relation between cell *B* and *C* is same. Therefore, if there is a rising transition assigned to cell *B*, direct implications would try to assign a logic 1 to cell *A* in the first vector and a logic 0 to cell *C* in the second vector, and propagate the new assignments throughout the circuit. If there are any conflicts, the partial path is a sequential false path under the launch-on-shift constraints. It is assumed that the scan chain design cannot be modified to reduce the dependence such as inserting dummy cells between the scanned flip-flops.

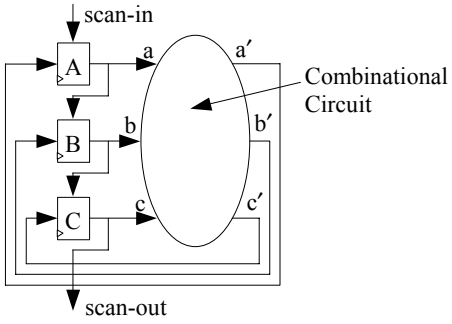

Figure 4. Implications on scanned flip-flops.

If the launch-on-capture approach is used, dependence exists between the two vectors. Even if the circuit has a pipeline structure, in which the two vectors can be independent, the structure can also be seen as the general structure shown in Figure 4. The conversion is shown is Figure 5. Thus the second vector is the output of the combinational circuit, derived from the first vector, excluding the primary input and output bits. In other words, $V_2=C(V_1)$, where V_1 and V_2 are the two vectors and C is the logic of the combinational circuit. For example, if it is assumed that a testable path has a rising transition launching from cell *A* and a rising transition captured on cell *B*, in Figure 4, then for the first vector, output *a´* must be a logic 1 (then it becomes the value for input *a* in the second vector); and for the second vector, input *b* must be a logic 0 because it is derived from the first vector. Then more direct implications can be performed from *a´* and *b*.

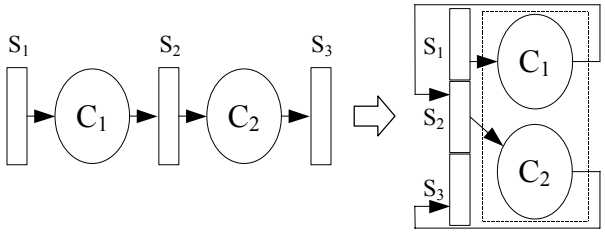

Figure 5. A pipeline structure.

3.3. Final Justification

A PODEM [11] based justification process is performed to find a vector pair when a complete path is found. Because most conflicts are eliminated by direct implications, this process is likely to succeed. Since the two vectors are dependent, whenever a decision (a logic value on any bit in either vector) is made at a primary input or scanned flip-flop, direct implications have to be performed to trim the search space. For the launch-on-shift approach, both vectors can be justified in this way.

For the launch-on-capture approach, because the second vector is derived as the circuit response to the first vector, one time frame expansion is used. In Figure 6, both the circuit and scan chains are duplicated. The first vector V_1 can be generated within one time frame, but since the second vector $V_2=C(V_1')$, the goal is to find a satisfying V_1'. Because V_1 and V_1' are identical excluding the "don't care" bits, in the justification process there must be no conflicts between V_1 and V_1', i.e. a bit is logic 1 in V_1 but 0 in V_1' (it is consistent if one of them is a "don't care"). Similarly, whenever a decision is made on any bit in either vector, direct implications must be performed to keep the logic assignments on any line in the two identical circuits consistent.

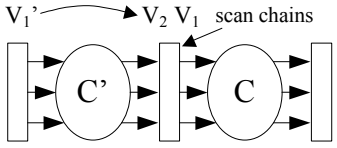

Figure 6. Time frame expansion for final justification using launch-on-capture.

4. Experimental Results

The proposed ATPG has been implemented in Visual C++ and run on Windows 2000 with a 450 MHz Pentium III processor and 128 MB memory. The unit delay model is used for simplicity. Experiments are performed on the full scan versions of the largest ISCAS89 benchmark circuits.

Table 1 shows the results for generating the longest robustly-testable path for each fault, under the launch-on-capture and launch-on-shift constraints. It is assumed that on each line there are slow-to-rise and slow-to-fall delay faults. The number of faults is twice the number of lines in a circuit, and the same as the number of transition faults. Column 3 shows the upper bound of detectable faults. This number is less than the total number of faults, because it is also assumed that the primary inputs cannot change their logic values from the first vector to the second vector in a test pattern, and the primary outputs are masked (not observed), due to the constraints from low-cost ATEs. Therefore no transition can happen on some of the lines and some transitions are not observable. Columns 4 and 5 show the number of primary inputs and flip-flops for each circuit. It is assumed that the circuits are full scan and there is only one scan chain for each circuit, in random order. Columns 6-8 show the results for the launch-on-capture approach and columns 9-11 for the launch-on-shift approach. Columns 6 and 9 show the number of paths generated by the ATPG. Before test compaction, each generated path has a test pattern, which contains two test vectors. The number of patterns after compaction is shown in column 7 and 10. The test patterns are compacted by a simple greedy static compaction algorithm, in which each new pattern is combined with the first compatible existing pattern. Columns 8 and 11 show the CPU time.

Figure 7 shows the transition fault coverage using the launch-on-capture or launch-on-shift approach only, or both. The data in column 3 in Table 1 is used as the total number of detectable faults. It can be seen that for most circuits, the launch-on-shift approach can detect more transition faults than the launch-on-capture approach, except for circuit s5378. The fault coverage assuming combinational enhanced-scan is shown for comparison. In combinational enhanced-scan, two independent vectors can be stored in the scan chain, so the fault coverage is an upper bound. Again in this mode it is assumed that the primary inputs hold their logic values from the first vector to the second vector, and the primary outputs are masked. Thus the coverage loss is purely due to the launch-on-capture and launch-on-shift constraints. Although the faults that the launch-on-capture approach cannot detect must be sequentially redundant in functional mode, the test patterns are still useful because these sequentially redundant faults may cause reliability problems.

Figure 8 is the comparison for path length using the launch-on-capture and launch-on-shift approaches, for circuit s15850. The faults are indexed so that the length of the longest testable path for each fault, under the launch-on-capture constraints, is in increasing order. The longest path for each fault assuming combinational enhanced-scan is also generated for comparison. Because the primary inputs hold and the primary outputs are masked, some faults have no coverage even if the circuit uses combinational enhanced-scan. It can be seen that for most faults, the maximum path length using the launch-on-shift approach is close to the upper bound, but this is not true for the launch-on-capture approach. All the other circuits have similar plots except for circuit s5378. This phenomenon indicates that the constraints from the launch-on-capture approach are stronger than the constraints from the launch-on-shift approach for most circuits.

Table 1. Test generation summary.

Circuit	# Lines	UB # Detectable Faults	# Primary Inputs	# Flip-Flops	Launch-on-Capture			Launch-on-Shift		
					# Paths Generated	# Test Patterns	CPU Time (m:s)	# Paths Generated	# Test Patterns	CPU Time (m:s)
s1423	1 423	2 420	17	74	369	199	1:07	666	191	0:46
s1488	1 488	1 310	8	6	187	85	0:18	206	81	0:16
s1494	1 494	1 324	8	6	188	84	0:18	204	79	0:17
s5378	5 378	7 564	35	179	1 799	406	1:09	1 110	94	1:12
s9234	9 234	16 166	36	211	2 326	762	5:53	3 608	681	5:12
s13207	13 207	22 886	62	638	3 180	892	10:55	6 469	1 635	7:02
s15850	15 850	24 338	77	534	2 625	470	8:00	5 828	645	4:50
s35932	35 932	59 246	35	1 728	9 762	36	78:27	12 194	44	52:18
s38417	38 417	74 926	28	1 636	14 531	890	34:59	17 554	655	19:29
s38584	38 584	59 454	38	1 426	9 683	527	31:36	21 047	679	23:08

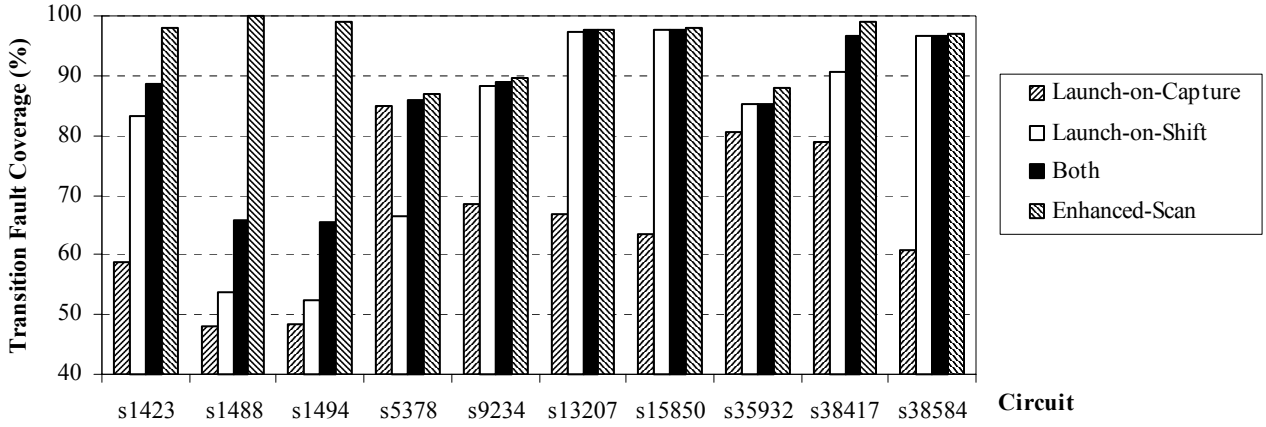

Figure 7. Transition fault coverage comparison.

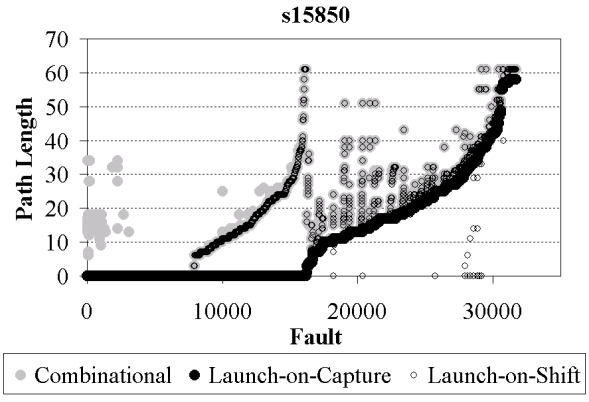

Figure 8. Path length comparison.

5. Conclusions and Future Work

We have proposed a fast automatic test pattern generation tool for path delay faults in sequential circuits, using the launch-on-shift and launch-on-capture at-speed test approaches. The generated test patterns can be applied to the commonly-used scan designs, and at-speed test can be performed using low-cost automatic test equipment.

Experiments have shown that for most circuits, the launch-on-capture approach results in stronger constraints and tighter dependence between the two vectors in a test pattern, than the launch-on-shift approach. The test quality using the launch-on-shift approach is close to the upper bound, in terms of the maximum path delay through each fault site. However, the launch-on-capture approach can eliminate most of the sequentially redundant faults in functional mode.

This work is being extended to generate test patterns for industrial designs. To achieve this goal, the tool must be able to handle circuits which have more complicated features, such as clock gating and multi-cycle paths.

Acknowledgements

This research was funded by the Semiconductor Research Corporation under contract 2000-TJ-844 and the National Science Foundation under contract CCR-1109413.

References

[1] B. R. Benware, R. Madge, C. Lu and R. Daasch, "Effectiveness Comparisons of Outlier Screening Methods for Frequency Dependent Defects on Complex ASICs," *IEEE VLSI Test Symp.*, Apr. 2003, pp. 39-46.

[2] R. R. Montanes and J. P. Gyvez, "Resistance Characterization for Weak Open Defects," *IEEE Design & Test of Computers*, vol. 19, no. 5, Sept.-Oct. 2002, pp. 18-25.

[3] J. Waicukauski, E. Lindbloom, B. K. Rosen and V. S. Iyengar, "Transition Fault Simulation," *IEEE Design & Test of Computers*, vol. 4, no. 5, April 1987, pp. 32-38.

[4] K. T. Cheng, "Test Generation for Delay Faults in Non-Scan and Partial Scan Sequential Circuits," *IEEE/ACM Int'l Conf. on Computer Aided Design*, Santa Clara, CA, Nov. 1992, pp. 554-559.

[5] M. Sharma and J. H. Patel, "Finding a Small Set of Longest Testable Paths that Cover Every Gate," *IEEE Int'l Test Conf.*, Baltimore, MD, Oct. 2002, pp. 974-982.

[6] W. Qiu and D. M. H. Walker, "An Efficient Algorithm for Finding the K Longest Testable Paths Through Each Gate in a Combinational Circuit," *IEEE Int'l Test Conf.*, Charlotte, NC, Sept. 2003, pp. 592-601.

[7] T. J. Chakraborty and V. D. Agrawal, "Effective Path Selection for Delay Fault Testing of Sequential Circuits," *IEEE Int'l Test Conf.*, Washington, DC, Nov. 1997, pp. 998-1003.

[8] M. P. Kusko, B. J. Robbins, T. J. Koprowski and W. V. Huott, "99% AC Test Coverage Using Only LBIST on the 1GHz IBM S/390 zSeries 900 Microprocessor," *IEEE Int'l Test Conf.*, Baltimore, MD, Oct. 2001, pp. 586-592.

[9] J. Saxena, K. M. Butler, J. Gatt, R. Raghuraman, S. P. Kumar, S. Basu, D. J. Campbell and J. Berech, "Scan-Based Transition Testing – Implementation and Low Cost Test Challenges," *IEEE Int'l Test Conf.*, Baltimore, MD, Oct. 2002, pp. 1120-1129.

[10] J. Benkoski, E. V. Meersch, L. J. M. Claesen and H. D. Man, "Timing Verification Using Statically Sensitizable Paths," *IEEE Trans. on Computer-Aided Design*, vol. 9, no. 10, Oct. 1990, pp. 1073-1084.

[11] P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits," *IEEE Trans. on Computers*, vol. C-30, no. 3, Mar. 1981, pp.215-222.