A New Diagnostic Test Generation Algorithm and a Coverage Metric[*]

Yu Zhang and Vishwani D. Agrawal
Auburn University, Department of Electrical and Computer Engineering, Auburn, AL 36849, USA
yzz0009@auburn.edu, *vagrawal@eng.auburn.edu*

Abstract–New algorithms for exclusive test generation and diagnostic fault simulation serve as core for a proposed diagnostic ATPG system. As defined in the literature, an exclusive test detects any one fault from a target fault pair but not the other. Hence it distinguishes between the two faults. In our algorithm, two or three ATPG modeling gates are added to a single copy of the circuit netlist to insert a single fault whose test is guaranteed to detect exactly one fault from the given pair. This exclusive test is valid for a multiple output circuit as well. If the inserted single fault has no test then the given fault pair is equivalent. The diagnostic fault simulator, that uses any available single fault simulation algorithm, partitions the fault list into groups. Fault pairs that the simulated vector can distinguish between are split among separate groups. Faults that form a single fault group are dropped from further simulation with subsequent vectors. We define a new diagnostic coverage (DC) metric as the ratio of the number of fault groups to the number of total faults. The proposed diagnostic ATPG system starts by first generating conventional fault coverage vectors. Those vectors are then simulated to determine the DC, followed by repeated applications of exclusive test generation and diagnostic fault simulation to enhance DC close to 100%.

1 Introduction

The objective of test generation is to find sufficient tests to detect all or most modeled faults. Although fault coverage (percentage) has a somewhat nonlinear relationship with the tested product quality or defect level (parts per million), for practical reasons fault coverage continues to be a measure of the test quality. Most test generation systems are built around *core ATPG algorithms* [3] for (1) finding a test vector for a target fault and (2) simulating faults to find how many have been detected by a given vector. The system then attempts to find tests of *high* fault coverage because the primary objective is fault detection, i.e., presence or absence of faults.

Some test scenarios go beyond fault detection. Here we must *diagnose* or identify the fault making the test intent different from the original detection coverage. In practice, detection tests with high coverage may not have adequate diagnostic capability. One often uses tests for multiple fault models [11, 13] or multiple tests for the same model [2]. Basically, we generate tests that are redundant for fault detection and then hope that they will provide better diagnostic capability. To reduce the excess tests we may resort to optimization or removal of unnecessary tests [4, 10].

A contribution of this paper is in providing basic core algorithms for a diagnostic test generation system. Given a pair of faults, we should either find a distinguishing test or prove that the faults are equivalent in a diagnostic sense [9]. The new exclusive test algorithm of Sections 4 and 5 represents a non-trivial improvement in computational efficiency over previously published work [1]. Next, we provide a diagnostic coverage metric in Section 6.1 similar to the detection fault coverage, such that a 100% diagnostic coverage means that each modeled fault is distinguished from all other faults. Diagnostic fault simulation of Section 6.3 is another core algorithm presented. Finally, a diagnostic test generation system in Section 6 combines the core algorithms and the new coverage metric into an ATPG system.

2 Definition

Exclusive test for a pair of faults has been defined as a test that detects one fault but not the other [1]. Primary intent is to distinguish between the fault pair. For a multiple output circuit, this definition is applied separately to each output. An exclusive test can detect both faults as long as they are not being detected at the same outputs. Perhaps a more appropriate term would be *distinguishing test.* However, following existing usage of the term, we will continue with *exclusive test.*

3 Previous Work

This background is reproduced from a previous publication [1]. Consider a pair of faults, f_1 and f_2.

---

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

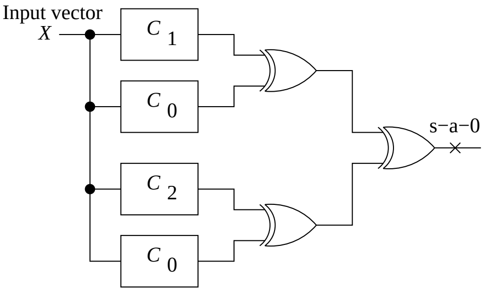


Figure 1. The general exclusive test problem.

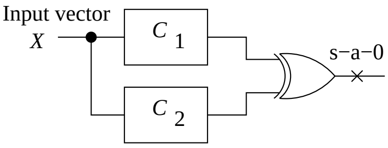


Figure 2. Simplified exclusive test.

An exclusive test must detect exactly one of these faults. Figure 1 illustrates generation of an exclusive test. A fault-free digital circuit under test (CUT) is shown as C. Blocks C_1 and C_2 are the same circuit with faults f_1 and f_2, respectively. The circuit is assumed to be combinational and can have any number of inputs. For clarity, we will only consider single output functions. Any input vector that produces a 1 output in Figure 1, i.e., detects a s-a-0 fault at the primary output is an exclusive test for the fault pair (f_1, f_2). This is the Boolean satisfiability version of the exclusive test problem,

$$(C \oplus C_1) \oplus (C \oplus C_2) = 1 \tag{1}$$

that can be simplified as:

$$C_1 \oplus C_2 = 1 \tag{2}$$

This simplification, shown in Figure 2, implies that the two faulty circuit outputs differ for the exclusive test. In the previous work, this problem was solved as that of test generation for a multiple fault in the combined C_1 and C_2 circuit. The multiple fault was modeled as a single fault [5] thus reducing the problem to that of single-fault ATPG in a twice as large circuit. In the new algorithm of the next section, we solve it as a single-fault ATPG problem without doubling the circuit size.

4 New Exclusive Test Algorithm

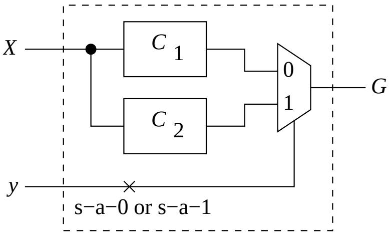


Figure 3. An ATPG circuit for exclusive test.

To further simplify the solution shown in Figure 2 and equation 2, we first transform it into an equivalent single-fault ATPG problem shown in Figure 3. Here we have introduced a new primary input variable y. The function G in Figure 3 can be expressed as Shannon's expansion [3] about y with cofactors C_1 and C_2:

$$G(X, y) = \bar{y}C_1 + yC_2 \tag{3}$$

The condition for detecting s-a-0 or s-a-1 fault on y, using Boolean difference [3], is

$$\frac{\partial G}{\partial y} = G(X, 0) \oplus G(X, 1) = C_1 \oplus C_2 = 1 \tag{4}$$

This is identical to the exclusive test condition of equation 2. Thus, we establish that a vector X that detects either s-a-0 or s-a-1 fault on y in the circuit $G(X, y)$ of Figure 3 will also detect the output s-a-0 fault in the circuit of Figure 2. We call G as *ATPG circuit.* It maps the given complex ATPG problem as a single fault ATPG problem. Next, we synthesize a compact circuit for $G(X, y)$.

Suppose fault f_1 is line x_1 s-a-a and fault f_2 is line x_2 s-a-b, where x_1 and x_2 are two signal lines in the circuit C. Fault variables a and b can assume any value 0 or 1. The primary input of C is a vector X that may or may not contain the two fault lines. We express the fault-free function using Shannon's expansion [3] as,

$$\begin{aligned} C(X, x_1, x_2) &= \bar{x}_1\bar{x}_2C(X, 0, 0) + \bar{x}_1x_2C(X, 0, 1) \\ &+ x_1\bar{x}_2C(X, 1, 0) + x_1x_2C(X, 1, 1) \end{aligned} \tag{5}$$

Therefore, the cofactors of $G(X, y)$ are given by,

$$\begin{aligned} C_1 = C(X, a, x_2) &= \bar{a}\bar{x}_2C(X, 0, 0) + \bar{a}x_2C(X, 0, 1) \\ &+ a\bar{x}_2C(X, 1, 0) + ax_2C(X, 1, 1) \\ C_2 = C(X, x_1, b) &= \bar{x}_1\bar{b}C(X, 0, 0) + \bar{x}_1bC(X, 0, 1) \\ &+ x_1\bar{b}C(X, 1, 0) + x_1bC(X, 1, 1) \end{aligned} \tag{6}$$

Let us define the following variables:

$$x'_1 = \bar{y}a + yx_1 \quad \text{and} \quad x'_2 = \bar{y}x_2 + yb \tag{7}$$

Using the rules of Boolean algebra, such as absorption and consensus theorems [8], we obtain

$$\bar{x}'_1\bar{x}'_2 = \bar{y}\bar{a}\bar{x}_2 + y\bar{x}_1\bar{b} \tag{8}$$

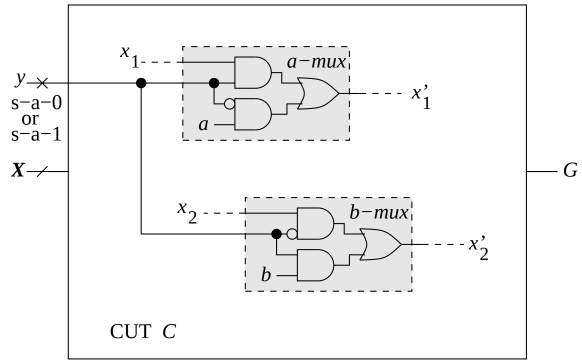


Figure 4. ATPG circuit with multiplexers inserted in CUT such that a test for s-a-0 or s-a-1 fault on y is an exclusive test for faults x_1 s-a-a and x_2 s-a-b.

$$\bar{x}'_1 x'_2 = \bar{y}\bar{a}x_2 + y\bar{x}_1 b \tag{9}$$

$$x'_1 \bar{x}'_2 = \bar{y}a\bar{x}_2 + yx_1\bar{b} \tag{10}$$

$$x'_1 x'_2 = \bar{y}ax_2 + yx_1 b \tag{11}$$

First we substitute C_1 and C_2 from equations 6 into equation 3 and then make use of equations 8 through 11, to obtain

$$\begin{aligned} G(X,y) &= \bar{x}'_1\bar{x}'_2 C(X,0,0) + \bar{x}'_1 x'_2 C(X,0,1) \\ &+ x'_1\bar{x}'_2 C(X,1,0) + x'_1 x'_2 C(X,1,1) \\ &= C(X, x'_1, x'_2) \end{aligned} \tag{12}$$

where the last result follows from the Shannon's expansion of the original circuit function C, given by equation 5, in which new variables x'_1 and x'_2 defined in equations 7 replace x_1 and x_2, respectively.

The synthesized circuit for $G(X,y)$, shown in Figure 4, is obtained by inserting two multiplexers, $a - mux$ and $b - mux$ controlled by a new primary input y, in the original circuit $C(X)$. For any 0 or 1 value of variables a and b each multiplexers simplifies either to a single gate or a gate with an inverter.

Consider the example circuit of Figure 5 from a previous paper [1]. We seek an exclusive test for two faults shown. The ATPG circuit for this problem is given in Figure 6. The logic shown with shading is obtained by simplifying the multiplexers of Figure 4 upon setting $a = 1$ and $b = 0$. The exclusive test found by a single fault ATPG is $a = 0,\ b = 1,\ c = 1,\ d = 0$. The signal values shown on lines are from five-valued D-algebra [3].

5 Multiple Output Circuits

An exclusive test for two faults in a single output circuit must detect only one of those faults. In a multiple output circuit, that same condition must be satisfied at least on one output. Suppose, C_1 and C_2 blocks in Figure 2 each has n outputs. We will then have n two-input XOR gates such that the ith XOR gate receives ith outputs from C_1 and C_2. All XORs feed into an n-input OR gate whose output contains the single s-a-0 faults to be detected.

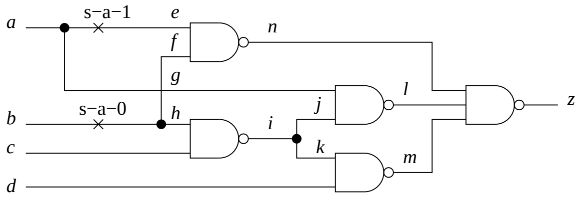


Figure 5. A circuit with exclusive test required for e s-a-1 and b s-a-0 [1].

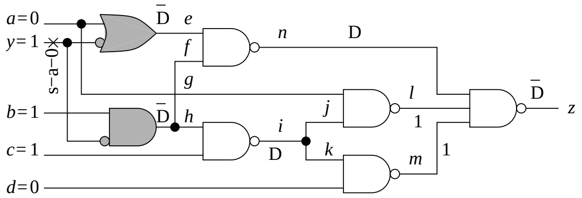


Figure 6. ATPG circuit for the exclusive test problem of Figure 5.

In general, an exclusive test for a multiple output circuit can detect both targeted faults as long as they are not being detected on exactly the same outputs. The ATPG circuit derived in the previous section remains valid for multiple outputs. Figure 7 shows the construction for two outputs.

Consider the problem of finding an exclusive test for two s-a-1 faults in the C17 circuit shown in Figure 8(a). The fault on signal y in the ATPG circuit of Figure 8(b) provides a test $X0011$. We can verify that this test detects both faults but at different outputs, hence it will distinguish between them.

6 Diagnostic ATPG System

6.1 Diagnostic Coverage Metric

To generate diagnostic tests we need a coverage criterion. This would be similar to the fault coverage (FC) used in conventional ATPG systems where fault detection is the objective. Since the core algorithm of previous sections generates distinguishing tests for fault pairs, our first inclination was to define coverage as the fraction of distinguishable fault pairs. Consider an example. For N faults, there will be $N(N-1)/2$ fault pairs. For a moderate size circuit, suppose $N = 10,000$ then there are 49,995,000 fault pairs. If our tests do not distinguish 5,000 fault pairs, then the coverage would be $(49,995,000 - 5,000)/49,995,000 =$

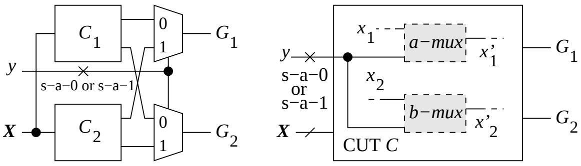
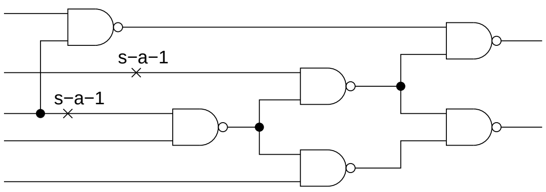
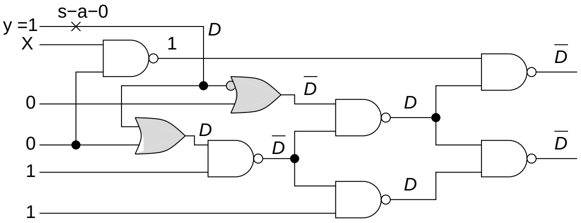


Figure 7. Exclusive test ATPG circuit for a two-output CUT.



(a) C17: Exclusive test problem for two s−a−1 faults.



(b) ATPG circuit and an exclusive test.

Figure 8. Exclusive test example for multi output circuit C17.

0.999, which turns out to be highly optimistic. There is additional problem of high complexity; for $N = 10^6$ the number of fault pairs is approximately half a billion. We, therefore, propose an alternative metric.

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. 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

$$DC = \frac{Number\ of\ detected\ fault\ groups}{Total\ number\ of\ faults} = \frac{n}{N} \quad (13)$$

Initially, without any tests, $DC = 0$, and when all

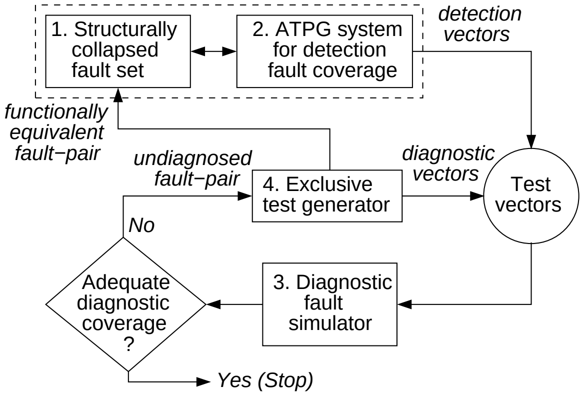


Figure 9. Diagnostic test generation system.

faults are detected and pair-wise distinguished, $DC = 1$. Also, the numerator in equation 13 is the number of fault dictionary syndromes [3] and the reciprocal of DC is the *diagnostic resolution* (DR) [1]. For completeness of this discussion, detection fault coverage (FC) is,

$$FC = \frac{Number\ of\ detected\ faults}{Total\ number\ of\ faults} = \frac{N - |g_0|}{N} \quad (14)$$

6.2 ATPG System

Figure 9 shows the flowchart of a diagnostic ATPG system implemented in the *Python* programming language [12]. Main functions are Blocks 1 through 4. Blocks 1 and 2 form a conventional detection coverage ATPG system. In our system, these functions are provided by Atalanta [6] and Hope [7] acquired from Virginia Tech. Block 4 is an exclusive test generator program that implements the core algorithm of Sections 4 and 5. Internally, it also uses Atalanta for detecting a fault on line y in the ATPG circuit constructed for the given fault pair (Figures 4 and 7). Block 3 is a diagnostic fault simulator described next.

6.3 Diagnostic Fault Simulator

We explain the simulation algorithm using a hypothetical example given in Figure 10. Suppose a circuit has eight faults ($N = 8$), a through h. Assume that the circuit has two outputs. The grey shading, which identifies the undetected fault group g_0, indicates that all faults are undetected in the initial fault list. Also, fault coverage (FC) and diagnostic coverage (DC) are both initially 0. Next, we assume that three vectors are generated in the detection phase (Blocks 1 and 2 in Figure 9) for a 100% FC.

The diagnostic phase begins with the three vectors supplied to Block 3. The first vector is simulated for all eight faults and is found to detect a, e and g. Suppose, faults a and e are detected only on the first output and

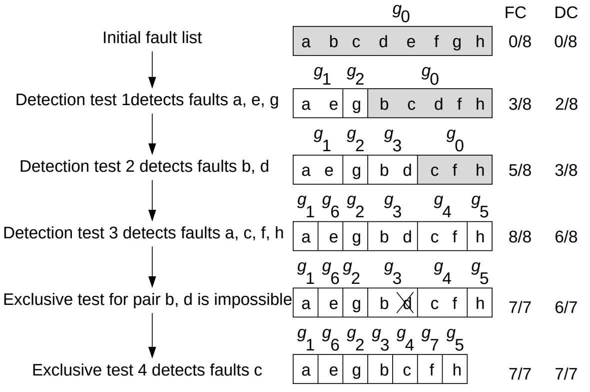


Figure 10. Diagnostic fault simulation.

g is detected on both outputs. Thus, fault pairs (a, g) and (e, g) are distinguishable, while the pair (a, e) is not distinguishable. The result is shown in the second list in Figure 10. The fault list is partitioned into three groups. The first two groups, g_1 and g_2, shown without shading contain detected faults. Group g_0 now has 5 faults. Each group contains faults that are not distinguished from others within that group, but are distinguished from those in other groups. The fault coverage is 3/8 and diagnostic coverage is 2/8.

Fault g, which is in a single fault group is dropped from further simulation. Because this fault has been uniquely distinguished from all other faults, its distinguishability status will not change by other vectors. Note that pair-wise distinguishability provided by future vectors can only subdivide the groups and a subdivision of a group with just one fault will be impossible. *The fact that faults can be dropped in diagnostic fault simulation is not always recognized.* However, fault dropping is possible here only because our interest is in coverage and not in minimizing the vectors. Seven faults are now simulated for the second vector, which detects faults b, d and g. Suppose, b and d are detected at the same set of outputs and hence are placed within same partition g_3. Thus, $FC = 5/8$ and $DC = 3/8$. No fault can be dropped at this stage.

Vector 3 detects faults a, c, f and h increasing the fault coverage to 100%. This time c and f are detected at the same set of outputs and are placed together in group g_4. Detection at different outputs distinguishes h from these two and hence h is placed in a separate group g_5. Also, noting that this test distinguishes between a and e, group g_1 is split into g_1 and g_6. Now, $FC = 8/8 = 1.0$ and $DC = 6/8$. Faults in fault groups with single fault are dropped.

Having exhausted the detection vectors, we find that two pairs, (b, d) and (c, f), are not distinguished. We supply target fault pair (b, d) to Block 4 in the ATPG system of Figure 9. Suppose we find that an exclusive test is impossible, i.e., two faults are equivalent. We remove one of these faults, say d, from g_3 and from the fault list as well. This does not change fault coverage since $FC = 7/7$, but improves the diagnostic coverage to $DC = 6/7$. All faults except c and f are now dropped from further simulation.

The only remaining fault pair (c, f) is targeted and an exclusive test is found. Thus, g_4 is partitioned to create group g_7 with fault f. The new partitioning has just one fault per group, $FC = 7/7$, and $DC = 7/7$.

6.4 Redundant and Aborted Faults

In the ATPG system of Figure 9 when a single fault ATPG program is run it can produce three possible outcomes, (1) test found, (2) no test possible, or (3) run aborted due to CPU time or search limit in the program. In (1) a new detection or exclusive test is found. In (2), if it is detection test then the fault is redundant and is removed from the fault list. If it is an exclusive test then the two faults are equivalent (perhaps functionally) and any one is removed from the fault list. In (3), for detection phase the fault remains in the set g_0 and results in less than 100% FC and DC. For an aborted exclusive test the target fault pair remains indistiguishable causing reduced DC and and worse diagnostic resolution (DR).

7 Results

We used the ATPG system of Figure 9 and Subsection 6.2. Internally, it employes the ATPG program Atalanta [6] and fault simulator Hope [7]. The circuit modeling for exclusive test and fault grouping for diagnostic fault simulation were implemented in the Python language [12]. The system runs on a PC based on Intel Core-2 duo 2.66GHz processor with 3GB memory. The results for c432 were as follows:

- Fault detection phase:
  Number of structurally collapsed faults: 524
  Number of detection vectors generated: 51
  Faults: detected 520, aborted 3, redundant 1
  Fault coverage, FC: 99.43%
- Diagnostic phase:
  Initial DC of 51 vectors: 91.985%
  Number of exclusive tests generated: 18
  Number of undistinguished groups: 13
  Largest size of undistinguished group: 2
  Final diagnostic coverage DC: 97.506%

Fault coverage (FC) and diagnostic coverage (DC) as functions of number of vectors are shown in Figure 11. First 51 vectors were generated in the fault detection phase, which identified only one of the known four redundant faults in this circuit. Diagnostic fault simulation computed the diagnostic coverage of 51 vectors as 91.985%. The diagnostic phase produced 18 exclusive tests while aborting on 13 fault pairs. Diagnostic coverage of combined 69 vectors was

Table 1. Diagnostic test generation for ISCAS'85 benchmark circuits.

Circuit	No. of faults	Detection test generation				Diagnostic test generation					
		Det. vect.	FC %	CPU s	DC %	Exclu. vect.	Abort. pairs	Equv. pairs	Max. undiag. group size	DC %	CPU s
c17	22	7	100.0	0.031	95.45	1	0	0	1	100.0	0.33
c432	524	51	99.24	0.032	91.99	18	13	0	2	97.51	1.75
c499	758	53	100.0	0.032	97.36	0	12	0	2	98.40	0.39
c880	942	50	100.0	0.047	92.57	10	55	0	2	94.16	2.77
c1355	1574	85	100.0	0.046	58.90	2	740	0	3	59.38	26.06
c1908	1879	114	99.89	0.047	84.73	20	300	1	8	86.46	10.84
c2670	2747	107	98.84	0.110	79.10	43	494	1	11	86.42	26.70
c3540	3428	145	100.0	0.125	85.18	29	541	3	8	89.69	22.03

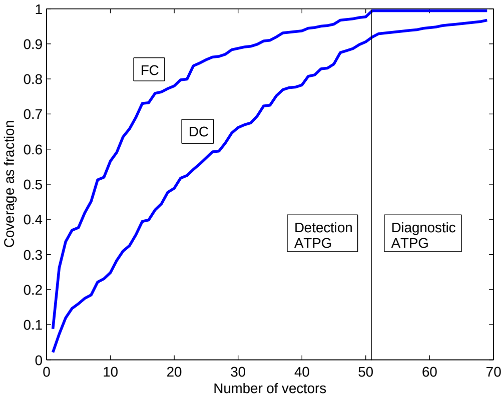


Figure 11. ATPG for c432, fault coverage (FC) and diagnostic coverage (DC).

97.506%. No undiagnosed group had more than 2 faults. Results for ISCAS'85 circuits, shown in Table 1, are analyzed in Appendix.

8 Conclusion

The core algorithms for exclusive test generation and diagnostic fault simulation should find effective use in the test generation systems of the future. Key features of these algorithms are: (1) exclusive test generation is no more complex than test generation for a single fault, and (2) diagnostic fault simulation has similar complexity as conventional simulation with fault dropping. The new definition of diagnostic coverage (DC) is no more complex than the conventional fault detection coverage and it directly relates to the *diagnostic resolution*, which is reciprocal of DC. Future extensions of this work would be on generating compact diagnostic tests and organizing fault dictionaries of test syndromes. Because our fault simulation is done with fault dropping, the syndromes will contain 0, 1, and X (don't care). However, these don't cares do not reduce the diagnosability of a fault. Although, reordering or compaction of vectors will be affected. This needs further investigation.

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] B. Benware, C. Schuermyer, S. Ranganathan, R. Madge, N. Tamarpalli, K.-H. Tsai, and J. Rajski, "Impact of Multiple-Detect Test Patterns on Product Quality," in *Proc. International Test Conf.*, 2003, pp. 1031–1040.

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

[4] Y. Higami, K. K. Saluja, H. Takahashi, S. Kobayashi, and Y. Takamatsu, "Compaction of Pass/Fail-based Diagnostic Test Vectors for Combinational and Sequential Circuits," in *Proc. ASPDAC*, 2006, pp. 75–80.

[5] Y. C. Kim, V. D. Agrawal, and K. K. Saluja, "Multiple Faults: Modeling, Simulation and Test," in *Proc. 15th International Conf. VLSI Design*, Jan. 2002, pp. 592–597.

[6] H. K. Lee and D. S. Ha, *On the Generation of Test Patterns for Combinational Circuits.* Tech. Report 12-93, Dept. of Elec. Eng., Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1993.

[7] H. K. Lee and D. S. Ha, "HOPE: An Effcient Parallel Fault Simulator for Synchronous Sequential Circuits," *IEEE Trans. Computer-Aided Design*, vol. 15, no. 9, pp. 1048–1058, Sept. 1996.

[8] V. P. Nelson, H. T. Nagle, B. D. Carroll, and J. D. Irwin, *Digital Logic Circuit Analysis & Design.* Englewood Cliffs, New Jersey: Prentice Hall, 1995.

[9] 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.

[10] 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.

[11] Y. Takamatsu, H. Takahashi, Y. Higami, T. Aikyo, and K. Yamazaki, "Fault Diagnosis on Multiple Fault Models by Using Pass/Fail Information," *IEICE Transactions on Information and Systems*, vol. E91-D, no. 3, pp. 675–682, 2008.

[12] G. van Rossum and F. L. Drake, Jr., editors, *Python Tutorial Release 2.6.3.* docs@python.org: Python Software Foundation, Oct. 2009.

[13] N. Yogi and V. D. Agrawal, "N-Model Tests for VLSI Circuits," in *Proc. 40th Southeastern Symp. System Theory*, 2008, pp. 242–246.

9 Appendix

We draw several inferences from the results of Section 7. For circuit c432, three ATPG runs in the detection phase and 13 in the diagnostic phase were aborted. In general, such behavior is expected because ATPG is NP-complete. This circuit is known to have four redundant faults and all three abotrted detection runs correspond to redundant faults. A closer examination of 13 aborted exclusive test ATPG runs showed all to be redundant. That means that all 13 fault pairs are *functionally* equivalent. Updating the fault list by removing one fault from each equivalent pair will increase DC to 100% and reduce the size of the largest undiagnosed fault group to 1.

Table 1 indicates a dropping DC as circuit size increases. Once again, this is expected because of larger numbers of aborted pairs. Notice 740 aborted pairs for c1355. This circuit is functionally equivalent to c499, which has a large number of XOR gates. In c1355, each XOR gate is expanded as four NAND gates. This implementation of XOR function is known to have several functionally equivalent faults. See V. D. Agrawal, A. V. S. S. Prasad and M. V. Atre, "Fault Collapsing via Functional Dominance," *Proc. International Test Conference*, 2003, pp. 274-280. That paper shows that the structurally collapsed set of 1,574 faults reduces to 950 faults when functional collapsing is used. If we use the set of 950 faults, the same $85 + 2 = 87$ vectors of Table 1 will show a significantly higher DC. Thus, the advantage of functional fault collapsing, though only marginal in detection ATPG, can be significant in diagnostic test generation.

Similarly, the size of the undiagnosed fault group tends to increase for larger circuits. It is 11 for c2670. This is related to the lower DC, whose reciprocal is the diagnostic resolution (DR). $DR > 1$ indicates poor diagnosis; the *ideal* resolution $DR = 1$ requires that each undistinguished fault group is no larger than 1.

The above discussion points to the need for improved fault collapsing and efficient redundancy identification algorithms. Much work on these has been reported, which we are exploring for suitable approaches. Interestingly, we find that most redundancies or functional fault equivalences are localized within relatively small subcircuits irrespective of how large the entire circuit is. Even though both problems have exponential complexities, this observation provides hope for finding efficient algorithms.

A comparison of two CPU time columns of Table 1 shows that diagnostic ATPG takes significantly more. This is because our implementation uses an existing ATPG program without changes. For detection ATPG circuit data structure is built once and then every vector generation run uses the same. Even though the data structure building time is higher (can be ten times) than that of a single vector generation, the total CPU time is dominated by the combined vector generation times. In diagnostic ATPG, because we modify the netlist, the ATPG program rebuilds the data structure prior to each vector generation run. An improved program will incrementally update the data structure instead of rebuilding it from scratch. That will bring the diagnostic ATPG time in line with the detection ATPG time.