Open access • Proceedings Article • DOI:10.1109/VTEST.1999.766661

Adaptive techniques for improving delay fault diagnosis — Source link

J. Ghosh-Dastidar, Nur A. Touba

Institutions: University of Texas at Austin

Published on: 26 Apr 1999 - VLSI Test Symposium

Topics: Stuck-at fault

Adaptive Techniques for Improving Delay Fault Diagnosis

Jayabrata Ghosh-Dastidar and Nur A. Touba

Computer Engineering Research Center
Department of Electrical and Computer Engineering
University of Texas, Austin, TX 78712-1084
E-mail: {dghosh, touba}@cat.ece.utexas.edu

Abstract

This paper presents adaptive techniques for improving delay fault diagnosis. These techniques reduce the search space for direct probing which can save a lot of time during failure analysis. Given a set of two-pattern tests that resulted in faulty output responses, a procedure for deriving additional two-pattern tests that will improve the diagnostic resolution of delay faults is described. Two new techniques based on adjacency testing and delay-size bounding are presented. These techniques can be used to greatly reduce the number of suspect lines and thereby provide a more precise diagnosis that is valid for either single or multiple delay faults. Experimental results are shown indicating that the number of suspects can be reduced dramatically for both single and multiple delay faults.

1. Introduction

With the advent of deep submicron technology and more aggressive clocking strategies, delay faults are becoming more prevalent. With ever increasing clock frequencies, small delay defects that were previously tolerable, are now starting to cause timing failures. Diagnosing delay faults is essential for improving the yield and quality of integrated circuits. Some direct probing mechanisms like E-Beam probing exist, but their effectiveness is limited by such factors as multiple layers of metals, CMP (chemical mechanical polishing), long test lengths, and new package types like flip-chip. Most importantly, with shrinking device sizes, the search space for any direct probing technique has increased tremendously. Automated tools are needed for failure analysis to significantly prune down the search space for direct probing.

Most of the work in diagnosis has been based on the classical single stuck-at fault model. Very little work has been done on diagnosis of delay faults. Girard, *et al.* [Girard 92] proposed an efficient procedure based on critical path tracing [Abramovici 83] from a 6-valued simulation. It is a post-test diagnostic procedure where given the faulty output responses from a set of test vectors, it identifies a set of *suspects* which are lines where the existence of a single point delay defect could explain all of the faulty test responses. This technique is very useful in reducing the search space for a delay defect. However, there are two drawbacks. One is that the number of suspects can still be very large, and the other is that the diagnosis may be invalid in the presence of multiple delay defects.

One way to improve the diagnostic resolution (i.e., reduce the number of suspects) for a given test set is to use a special diagnostic ATPG procedure described by Girard, *et al.*, in [Girard 96]. This is a simulation-based ATPG procedure that uses genetic algorithms to arrive at a final test set. By using this test set instead of one derived by other means, when a delay fault occurs, a better diagnostic resolution can be obtained during diagnosis.

This paper takes a different approach for improving diagnostic resolution. Adaptive techniques are presented for improving diagnostic resolution for both single and multiple delay faults. Given a set of two-pattern tests that resulted in faulty output responses, a procedure for generating additional two-pattern tests that will improve the diagnostic resolution is described. Two new techniques based on adjacency testing [Craig 85] and delay-size bounding are presented. These techniques can be used to greatly reduce the number of suspects and thereby provide a more precise diagnosis that is valid for either single or multiple delay faults. This is very important for providing better guidance for direct probing and can save a lot of time during failure analysis.

Note that while the adaptive techniques in this paper are described for gate delay fault diagnosis, they can also be used for path delay fault diagnosis. Hsu and Gupta [Hsu 98] recently proposed a path-oriented diagnosis procedure. The adaptive techniques described here can be used to generate additional two-pattern tests that will increase the diagnostic resolution for the procedure described in [Hsu 98].

The paper is organized as follows: Section 2 describes how path tracing from a faulty output can be used to identify a set of suspects. Section 3 presents an approach for using adjacency tests to reduce the size of the suspect set. Section 4 explains how robust tests that give fault-free responses can be used to eliminate suspects from the suspect set. Experimental results for benchmark circuits are shown in Sec. 5. Section 6 is a conclusion.

2. Critical Path Tracing

The idea of performing critical path tracing using a 6-valued algebra to identify a set of gate delay fault suspects that may explain an observed faulty output was proposed in [Girard 92]. For a test sequence, each two-pattern test for which the circuit-under-test produced a faulty output is simulated using a 6-valued algebra based on the H6 algebra [Hayes 86]. The symbols used are the following: *S0* for static zero, *S1* for static one, *R1* for a rising transition, *F0* for a falling transition, *X0* for static-0 hazard, and *X1* for a static-1 hazard. The advantage of using this 6-valued algebra is that it does not depend on any gate propagation delay or delay fault size. From each faulty output, critical path tracing is performed to identify the suspects (i.e., critical lines) that may have caused the faulty value. For each two-pattern test, *t*, which gives a faulty output response at output *j*, the set of suspects will be denoted *SUSPECTS(t,j)*.

The set of prime suspects where a single point fault may explain all of the observed faulty behavior will be denoted as *PRIME_SUSPECTS*. It is obtained by taking the intersection of all suspects for all two-pattern tests and all outputs:

PRIME_SUSPECTS =
$\cap$ *SUSPECTS(t,j)* for all *t* and *j*

If the set *PRIME_SUSPECTS* is empty, then it can immediately be concluded that there are multiple faults.

The set of multiple suspects, whose members are suspects for multiple faults, will be denoted as *MULTIPLE_SUSPECTS*. It is obtained by taking the union of all suspects minus the set *PRIME_SUSPECTS*.

MULTIPLE_SUSPECTS =
$\cup$ *SUSPECTS(t,j) – PRIME_SUSPECTS* for all *t* and *j*

Note that each member of the set *MULTIPLE_SUSPECTS* can explain some of the observed faulty response, but cannot individually explain all of the observed faulty response.

In the example in Fig. 1, gates *G1* and *G3* have delay faults. The primary outputs O_1 and O_2 have faulty values on application of the test vector pair *(01001,11000)*. Using critical path tracing starting from O_1, we have *SUSPECTS(t,1) = {*O_1 , L_3 , I_1 *}*. Critical path tracing starting from O_2 results in *SUSPECTS(t,2) = {*O_2, L_{10}, I_5*}*. So the intersection of sets *SUSPECTS(t,1)* and *SUSPECTS(t,2)* gives *PRIME_SUSPECTS* = { }. The set union of *SUSPECTS(t,1)* and *SUSPECTS(t,2)* gives *MULTIPLE_SUSPECTS = {*O_1, L_3, I_1, O_2, L_{10}, I_5*}*. Note that *MULTIPLE_SUSPECTS* contains the actual faulty lines L_3 and L_{10} whereas *PRIME_SUSPECTS* is empty signifying the presence of multiple faults.

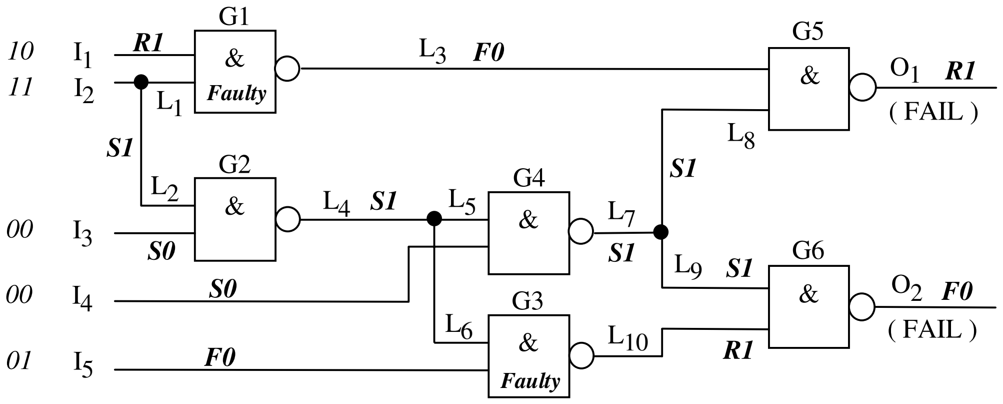

Figure 1. Signal values at Different Lines after Application of Vector Pair *(01001,11000)*

Faulty Gates: G1 and G3
SUSPECTS(t,O_1*) = {*O_1, L_3, I_1*}*
SUSPECTS(t,O_2*)= {*O_2, L_{10}, I_5*}*
PRIME_SUSPECTS = {}
MULTIPLE_SUSPECTS = {O_1, L_3, I_1, O_2, L_{10}, I_5*}*

The size of the suspect set depends on which two-pattern tests detect the delay faults. The fewer the suspects, the better the diagnostic resolution. The focus of this paper is to present techniques to further reduce the set of suspects by performing additional tests.

3. Using Adjacency Tests For Diagnosis

One of the approaches proposed here for reducing the set of suspects is to use adjacency tests that are derived from a failing two-pattern test in the original test sequence. An adjacency test is one in which the V_1 and V_2 vectors of a two-pattern test differ in only one input bit (i.e., only one input value makes a transition while the others are held constant). The advantage of using an adjacency test during diagnosis is that if the adjacency test results in a faulty output value, then the set of suspects derived by critical path tracing will be very small. This is because there is a transition on only one input, so the number of lines in the circuit which are tested by an adjacency test is relatively small which makes diagnosis much easier. The key is to find an adjacency test that results in a faulty output value. When performing post-test diagnosis where one or more two-pattern tests in the original test sequence produced a faulty output response, the idea is to derive adjacency tests from the failing two-pattern tests. In this section, a procedure is described for making use of information extracted from adjacency tests derived from the failing two-pattern tests.

3.1 Deriving Adjacency Tests From Failing Two-Pattern Tests

Given a two-pattern test $(V_{1,orig}, V_{2,orig})$ in the original sequdence that failed, let $DIFF\_INPUTS(V_{1,orig}, V_{2,orig})$ be the set of inputs whose values differ in $V_{1,orig}$ and $V_{2,orig}$ (i.e., the set of inputs on which there are transitions. Then a two-pattern adjacency test $(V_{1,adj}, V_{2,adj})$ can be derived by setting both $V_{1,adj}$ and $V_{2,adj}$ equal to $V_{2,orig}$ and then complementing one of the input bits in $V_{1,adj}$ corresponding to one of the inputs in the set $DIFF\_INPUTS(V_{1,orig}, V_{2,orig})$. If there are n inputs in the set $DIFF\_INPUTS(V_{1,orig}, V_{2,orig})$, then there are n different adjacency tests can be derived from $(V_{1,orig}, V_{2,orig})$. For example, if $(V_{1,orig}, V_{2,orig}) = (10110, 00100)$ then the set $DIFF\_INPUTS(V_{1,orig}, V_{2,orig})$ would include the first and fourth inputs. Two adjacency tests could be derived: *(10100, 00100)* and *(00110, 00100).*

If the adjacency tests derived in the manner described above are applied to the circuit-under-test, then there are three cases:

1. Exactly one adjacency test produces a faulty output response - In this case, the set of suspects is narrowed down considerably because the number of critical lines in the critical path tracing for the adjacency test will generally be much less than that for the original two-pattern test that produced a faulty output response.
2. More than one adjacency test produces a faulty output response - In this case, the set of prime suspects is the intersection of the critical lines for each adjacency test. If the intersection of the critical lines is empty, then this implies a multiple point fault (diagnosing multiple point faults will be describe in detail later).
3. None of the adjacency tests produces a faulty output response - This implies that multiple input transitions in the original two-pattern test are needed to detect the delay fault. This may occur if a hazard was originally propagated to an output due to the delay fault. This information can be used to prune the suspect list down further as will be explained later.

In cases 1 and 2, a much better diagnostic resolution can be obtained compared with using only the original two-pattern test that produced the faulty output. In case 3, additional steps need to be taken to reduce the number of suspects.

3.2 Deriving Minimum Input Transition Tests From Failing Two-Pattern Tests

The strategy for case 3 is to begin with the original two-pattern test and systematically reduce the number of transitions as much as possible while still detecting the fault. If there are n inputs in the set $DIFF\_INPUTS(V_{1,orig}, V_{2,orig})$, then the first step is to derive n two-pattern tests by simply removing one of the input transitions in the original two-pattern test. This is done by setting the corresponding input bit value in the V_1 pattern equal to that in V_2 pattern. One of the resulting two-pattern tests that still produces a faulty output is then chosen arbitrarily and the process repeats recursively until a point is reached where none of the derived two-pattern tests produces a faulty output. Obviously this point is guaranteed to be reached once the number of input transitions gets down to one since none of the adjacency tests produced a faulty output. Thus, the maximum number of two-pattern tests that are applied is $(n-1)(n+2)/2$. The two-pattern test with the fewest number of input transitions, $(V_{1,min\_tran1}, V_{2,min\_tran1})$, that still produces a faulty output can then be used for critical path tracing to generate the suspect set.

To further reduce the size of the suspect set, the procedure described above can be repeated with one modification. Instead of arbitrarily selecting which two-pattern test (that produces a faulty output) is used to derive the next batch of new two-pattern tests, the following criteria is used. The two-pattern test that has the fewest number of input transitions in common with *(*V_{1,min_tran1}*,* V_{2,min_tran1}*)* is selected at each step in the recursion. The purpose of this heuristic is to try to find a new two-pattern test *(*V_{1,min_tran2}*,* V_{2,min_tran2}*)* that still produces a faulty output, but has a different set of input transitions than *(*V_{1,min_tran1}*,* V_{2,min_tran1}*)*. The reason for this is so that the set of prime suspect derived by taking the intersection of the set of suspects for *(*V_{1,min_tran1}*,* V_{2,min_tran1}*)* and *(*V_{1,min_tran2}*,* V_{2,min_tran2}*)* will be small. If the set of prime suspects is empty, then this implies a multiple point fault. The procedure describe here for generating additional two-pattern tests can be repeated as desired to try to further reduce the intersection of the suspect sets for the failing two-pattern tests.

3.3 Performing Diagnosis for Multiple Point Faults

If the set of prime suspects become empty, then no single point fault can explain all of the observed faulty output responses, hence there must be a multiple point fault. In the presence of a multiple point fault, the diagnosis strategy proposed here is to determine for all two-pattern tests *t* and all faulty outputs *j* which set of suspects *SUSPECTS(t,j)* is the smallest. At least one of the multiple point faults must be in this set of suspects. By focusing the diagnosis on the smallest set of suspects, the direct probing process can be aided. Once one of the defects is located, then that information can be used to deduce where the location of the other defects could be.

4. Delay-Size Bounding

Another approach for reducing the set of suspects is to find two-pattern tests that place an upper bound on the amount of delay that a defect could add on a suspect line. If the upper bound is tight enough, then the suspect can be removed from the suspect list. Consider a delay defect on a line which adds additional delay on every path that goes through it. The amount of delay added by the defect will be referred to as the "size" of the defect. The size of a delay defect can range from small to large (gross) amounts of delay. Depending on the size of a delay defect and the amount of delay slack on each path going through it, a delay defect may cause a fault along *all* of the paths going through it (corresponding to a gross defect), along *some* of the paths going through it, or along *none* of the paths going through it (i.e., the delay defect size is less than delay slack).

The longest robustly tested [Lin 87] path through a suspect that is found to be fault-free places an upper bound on the delay defect size that could be present at the suspect. For example, if the longest robustly tested path through a suspect is 9ns, and the clock period is 10ns, then if the output response is fault-free, there cannot be a delay defect with size greater than 1ns present at the suspect and still be consistent with the observed fault-free response. Note that a robustly tested path is one in which the test cannot be invalidated by the presence of hazards or other delay faults in the circuit.

Consider the case where the shortest path through a suspect that produces a faulty response is l_{faulty}, and the length of a robustly tested path through a suspect that produces a fault-free response is $l_{fault\text{-}free}$. If $l_{fault\text{-}free}$ is greater than l_{faulty}, then the suspect can be removed from consideration because there is no delay defect size which would be consistent with the observed faulty and fault-free responses. It can be concluded that the faulty responses must be due to a delay defect at another suspect [Ghosh-Dastidar 98].

So one strategy for reducing the number of suspects is to apply a two-pattern test that robustly tests a path through a suspect that is longer than the shortest path through the suspect in which a faulty response was observed. If such a two-pattern test produces a fault-free response, then the suspect can be eliminated from further consideration. Given the set of suspects for a circuit-under-test, the shortest faulty path through each suspect can be computed. Then an attempt can be made to find a robust test for a path through the suspect that is longer than the shortest faulty path. The resulting two-pattern test can be used to try to reduce the number of suspects.

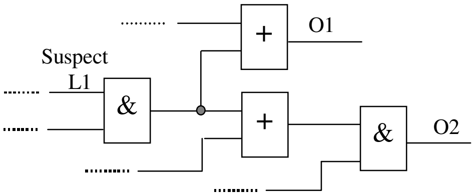

Figure 2. Example of Delay-Size Bounding

Consider the example in Fig. 2. A faulty value is observed at the output O1 for one of the original two-pattern tests. Suppose that from critical path tracing, the line L1 is found to be a suspect. If the path through L1 to output O2 is longer than the path through L1 to output O1, then the strategy proposed here is to find a two-pattern test that robustly tests the path through L1 to output O2. If that test results in a fault-free value at O2, then L1 can be removed from the suspect list. This is because if there was a delay defect at L1, it should have also caused a faulty response at output O2.

5. Experimental Results

Experiments using the adaptive techniques described in this paper were performed for some of the ISCAS 85 benchmark circuits [Brglez 85]. Table 1 shows results for the case where a single fault was injected in the circuit-under-test. The number of suspects obtained using the critical path tracing method described in [Girard 92] is shown. By using the adaptive techniques described in this paper, the resulting number of suspects is shown in the last column. As can be seen, the number of suspects can be greatly reduced providing a much better diagnostic resolution. Table 2 shows results for the case where multiple faults were injected in the circuit-under-test. In this case also, the number of suspects can be significantly reduced using the adaptive method.

Table 1 . Experimental Results for Fault Diagnosis of Single Point Faults

Circuit	Single Point Fault		Num. of Suspects	
	Fault Location	Defect Size	[Girard 92]	Adaptive Method
C432	Gate: [209]	2	47	31
C432	Gate: 341	3	82	31
C432	Gate: [342]	3	16	5
C880	Gate: [63]	2	47	8
C880	Gate: 541	2	74	40
C880	Gate: 832	3	47	8
C1908	Gate: 137	4	155	12
C1908	Gate: 425	4	184	37
C1908	Gate: 1116	4	155	12

Table 2. Experimental Results for Fault Diagnosis of Multiple Point Faults

Circuit	Multiple Point Faults		Num. of Suspects	
	Fault Location	Defect size	Critical Path Tracing	Adaptive Method
C432	Gate: 350 Gate: [209]	3	82	53
C432	Gate: 154 Gate: [209]	3	127	47
C432	Gate: 154 Gate: [342]	3	127	44
C880	Gate: 832 Gate: 773	3	118	45
C880	Gate: 1 Gate: 541	3	74	19
C880	Gate: 819 Gate: 541	3	118	45
C1908	Gate: 1921 Gate: 1104	4	89	21
C1908	Gate: 1357 Gate: 1116	4	184	39
C1908	Gate: 1921 Gate: 1647	4	89	21

6. Conclusions

This paper presented adaptive techniques for generating additional two-pattern tests to improve the diagnostic resolution for delay faults. These techniques can be used during post-test diagnosis to significantly reduce the number of suspects and therefore provide better guidance for direct probing. This speeds up the failure analysis process and can save a lot of time.

Acknowledgements

This material is based on work supported in part by the Defense Advanced Research Projects Agency (DARPA) under Contract No. DABT63-97-C-0024, and in part by the National Science Foundation under Grant No. MIP-9702236.

References

[Abramovici 83] Abromovici, M., P. R. Menon, and D. T. Miller, “Critical Path Tracing - An Alternative to Fault Simulation”, *Proc. 20th Design Automation Conference*, pp. 214-220, 1983.

[Abramovici 90] Abramovici, M., M. A. Breuer, and A. D. Friedman, *Digital System Testing and Testable Design,* Computer Science Press, New York, 1990.

[Brglez 85] Brglez, F., and H. Fujiwara, “A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortan,” *Proc. International Symposium on Circuits and Systems*, pp. 663-698, 1985.

[Craig 85] Craig, G.L., and C.R. Kime, “Pseudo-Exhaustive Adjacency Testing: A BIST Approach for Stuck-Open Faults,” *Proc. International Test Conference*, pp. 126-137, 1985.

[Girard 92] Girard, P., C. Landrault, S. Pravossoudovitch, “A Novel Approach to Delay-Fault Diagnosis”, *Proc. 29th Design Automation Conference,* pp. 357-360, 1992.

[Girard 96] Girard, P., C. Landrault, S. Pravossoudovitch, and B. Rodriguez, “A Diagnostic ATPG for Delay Faults Based on Genetic Algorithms”, *Proc. International Test Conference*, pp. 286-293, 1996.

[Ghosh-Dastidar 98] Ghosh-Dastidar, J., and N.A. Touba, "A Systematic Approach for Diagnosing Multiple Delay Faults", *Proc. Symposium on Defect and Fault Tolerance*, pp. 211-216, 1998.

[Hayes 86] Hayes, J.P., “Digital Simulation with Multiple Logic Values”, *IEEE Trans. Computer-Aided Design*, vol. 5, no. 2, pp. 274 -283, Apr. 1986.

[Hsu 98] Hsu, Y.-C., and S.K. Gupta, “A New Path-Oriented Effect-Cause Methodology to Diagnose Delay Failures,” *Proc. International Test Conference*, pp. 758-767, 1998.

[Lin 87] Lin, C.J., and S. M. Reddy, “On delay fault testing in logic circuits”, *IEEE Trans. Computer-Aided Design,* Vol. CAD-6, pp. 694-703, Sept. 1987.

[Smith 85] Smith, G.L., “Model for delay faults based upon paths”, *Proc. Int. Test Conference*, pp. 342-349, 1985.