Tutorial: Delay Fault Models and Coverage

Ananta K. Majhi*
Mentor Graphics (India) Ltd., Hyderabad 500034, India
anant_majhi@mentorg.com

Vishwani D. Agrawal
Bell Labs, Murray Hill, NJ 07974, USA
va@research.bell-labs.com

Abstract

Failures that cause logic circuits to malfunction at the desired clock rate and thus violate timing specifications are currently receiving much attention. Such failures are modeled as delay faults. They facilitate delay testing. The use of delay fault models in VLSI test generation is very likely to gain industry acceptance in the near future. In this paper, we review delay fault models, discuss their classifications and examine fault coverage metrics that have been proposed in the recent literature. A comparison between delay fault models, namely, gate delay, transition, path delay, line delay and segment delay faults, shows their benefits and limitations. Various classifications of the path delay fault model, that have received the most attention in recent years, are reviewed. We believe an understanding of delay fault models is essential in today's VLSI design and test environment.

1 Introduction

Digital system designers have traditionally maximized the frequency of system clocks in order to obtain the highest performance from the hardware. The maximum allowable clock rate is determined by the propagation delays of the combinational logic block between latches. Consider the circuit under test shown in Figure 1. During the normal operation of the circuit, the input clock C_1 is the same as the output clock C_2 and the period (T_c) of C_1 and C_2 corresponds to the system clock. This period should be greater than the maximum propagation delay DP_{max} for the circuit. However, during testing for delay faults, we use two separate test clocks, C_1 and C_2, running at a frequency that is slower than the normal system clock. Thus, the period of test clocks, T_t, is longer than T_c. The two test clocks are skewed by the amount T_c. The activation of the output clock C_2 must follow the activation of the input clock C_1 by at least DP_{max} time units (i.e., $T_c = t_2 - t_1 \geq DP_{max}$). If the output clock is activated sooner, then unstabilized and possibly incorrect logic values may be latched in output latches. Since

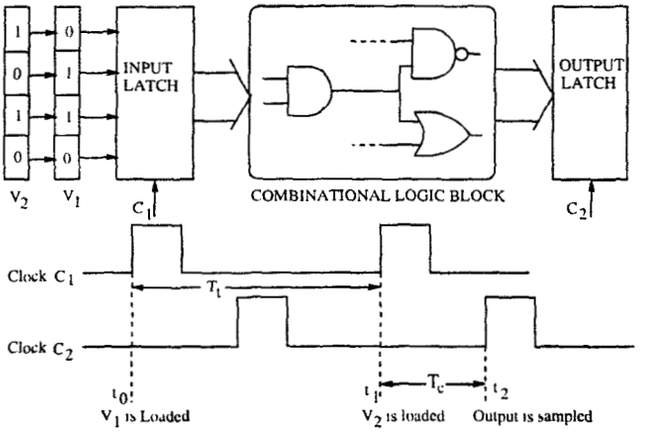

Figure 1: Hardware model and clock timings

delay faults do not alter the logic function realized by a circuit and since the tests for stuck-at faults are normally applied at a slow clock rate (due to the ATE limitation or other reasons), they are inadequate for detecting delay faults. Special two-pattern test vectors are required for detecting delay faults.

The hardware model used in delay fault testing is shown in Figure 1. Here the vector pair $< V_1, V_2 >$ constitutes a delay test and signals C_1 and C_2 are used to clock the input and output latches, respectively. At time t_0, an initializing input vector V_1 is applied, and the circuit is allowed to stabilize under input V_1. At time t_1, the propagation vector V_2 is applied, and the outputs are sampled at time t_2, where $(t_2 - t_1)$ is the intended time interval between the input and output clocks, called the rated clock interval T_c.

Exhaustive testing is quite impractical for delay faults since the total number of pattern-pairs required will be $(2^n)(2^n - 1)$, which is of the order 2^{2n}, for a circuit having n inputs. One must derive suitable and reasonable delay fault models and devise algorithms that can generate tests for the modeled faults. Various fault models used in delay fault testing, their classifications, and coverage metrics are discussed in the following sections.

*Formerly with Texas Instruments, Bangalore 560017, India

2 Delay Fault Models

There are three classical fault models that have been developed in recent past to represent delay defects (i.e., transition fault, gate delay fault and path delay fault). In recent years, two more fault models (i.e., line delay fault and segment delay fault) have been developed, which are basically derived from the classical ones. Each of these fault models have their own limitations and advantages in various aspects that have been discussed in some detail in this paper.

2.1 Transition Fault Model

The transition fault model [6, 23, 29, 34] is considered as a logical model for a defect that delays a rising or falling transition at inputs and outputs of logic gates. There are two kinds of transition faults, i.e., *slow-to-rise* and *slow-to-fall*. The slow-to-rise (fall) transition fault temporarily behaves like a DC stuck-at-0 (1) fault. A test for a transition fault is a pair of input patterns, one *(initialization pattern)* to set up the initial state for the transition and another *(propagation pattern)* to cause the appropriate transition and observe its effect at a primary output. The propagation pattern is identical to a pattern that detects the corresponding DC stuck-at fault. The transition fault coverage is a measure of the effectiveness of the delay test in detecting *large* delay variations. Transition faults model defects for which the delay is large enough to cause a logical failure when the signal propagates along any path through the site of the fault. The main drawback of this model is the assumption of a large gate delay defect [33]. Also, it is difficult to tell how small a delay fault can be, before it is not detectable. In practice, delay variations tend to be distributed over many circuit elements. Thus, many small gate delay faults, each undetectable as a transition fault, can give rise to a large path delay fault.

2.2 Gate Delay Fault Model

Carter *et al.* [4] introduced a *quantitative* model for delay faults, known as the *gate delay fault*. They assume that delays through logic gates are known with some precision. The characteristics (size and location) of likely delay faults are also known. The delays through a gate are represented by intervals in this model. A fault is an added delay of certain size (magnitude), say δ, in the propagation of a rising or falling transition from the gate input to output. The set of faults considered includes numerical delay information. An excessive delay of 3 nanoseconds at a point is not the same fault as an excessive delay of 5 nanoseconds at that point.

Most of the recent research in this area has concentrated on the determination of fault sizes detected by a given test [4, 27, 28]. Given a particular fault of a fixed known size, Carter *et al.* [4] provide a method to determine whether a test T detects that fault. This is clearly a painstaking and inefficient method, and it would be more desirable to find a certain *minimum* fault size at a fault site such that given a test T for a fault at the above fault site, T is guaranteed to detect any fault at that site with a magnitude greater than the determined minimum size.

2.3 Path Delay Fault Model

The path delay fault model was first proposed by Smith [33]. This model has received greater attention than the gate delay and transition fault models, and has been quite extensively studied [1, 2, 3, 5, 9, 10, 17, 19, 21, 25, 26, 30, 31]. A considerable amount of research has already been reported on various aspects of test generation and fault simulation of path delay faults.

In path delay fault model, any path with a total delay exceeding the system clock interval is said to have a path delay fault. This models distributed defects that affect an entire path. For each physical path P, connecting a primary input to a primary output of the circuit, there are two corresponding delay paths. The *rising path (falling path)* is the path traversed by a transition that is initiated as a rising (falling) transition at the input of path P and changes the direction of transition whenever it passes through an inverting gate. We present the following definitions that are frequently used in path delay fault testing [19].

Definition 1: Let G be a gate on path P in a logic circuit, and let r be an input to gate G; r is called an *off-path sensitizing input* if r is not on path P.

Definition 2: A two-pattern test $< V_1, V_2 >$ is called a *robust test* for a delay fault on path P, if the test detects that fault *independently* of all *other delays* in the circuit [30].

Definition 3: A two-pattern test $< V_1, V_2 >$ is called a *nonrobust test* for a delay fault on path P, if it detects the fault under the assumption that no other path in the circuit involving the off-path inputs of gates on P has a delay fault [12].

2.4 Line Delay Fault Model

We can combine the relevant features of transition [34] and path delay [33] fault models to define a *line delay test* [20, 22]. A rising (falling) line delay test will test the longest sensitizable path passing through a target line producing a rising (falling) transition on it. With this model, the coverage is measured for all lines with two possible transitions. Thus, the maximum number of faults (or tests) is twice the number of lines. (For example, in c6288, we will consider only 12576 line delay

faults, whereas the total number of possible path faults is $\approx 1.98 \times 10^{20}$.) Yet, the test criterion is similar to path delay fault, and not like gate or transition delay fault. In general, a test will cover several lines.

Conventional path delay test generators attempt to derive robust tests for a subset of paths in the circuit, based on some path selection criterion such as the worst-case path selection [18] or a threshold-based path selection [14, 16, 35]. However, a large number of these paths may not be robustly testable and hence the test coverage of the targeted paths can be very low [10]. The new coverage metric seeks to remove this deficiency by attempting to derive a pair of line delay tests for each line in the circuit.

The basic idea of an iterative approach for generating a robust test was first proposed by Park and Mercer [24]. They have followed an *approximate method* where the search space for test generation is biased to find a test along a path whose propagation delay is greater than or equal to a predefined threshold value. Bose [2] preselects paths in a given range of lengths and shows that despite a low path coverage, a high gate (or line) coverage can be obtained. In that case, however, lines are not tested through longest sensitizable paths. The method of Majhi *et al.* [22], on the other hand, is an *exact method* for generating a robust test for the longest testable path through each line. To facilitate the simultaneous consideration of robust and nonrobust tests, they use a 9-value logic system described in [21].

2.4.1 Line Delay Coverage Metric

The motivation of defining the line delay test is to robustly detect the smallest incremental delay defect associated with a rising or falling transition at any line. Suppose, Δ_L is the incremental delay of a rising or falling transition through line L. Then, for detection of this delay fault,

$$\Delta_L + T_P > T_C \qquad \text{or} \qquad \Delta_L > T_C - T_P \tag{1}$$

where T_C = system clock period and T_P = nominal delay of the path P through which L is tested. From relation (1), we determine that the smallest incremental delay fault on L can be detected via the path through L having the longest nominal delay $(T_P)_{max}$, i.e.,

$$(\Delta_L)_{min} = T_C - (T_P)_{max} \tag{2}$$

By sensitizing the longest path through L, we are able to detect the delay fault of the smallest size. However, simultaneous delay variations are possible for other gates on P due to correlation with L. Suppose, delays of other gates increase. Then the line delay test for L will detect a delay fault of even smaller size. If the delay of other gates reduce while that of L increases, the sensitivity of the test reduces. Considering correlation of delays, this later case is less probable. The basic assumption associated with the line delay fault model is that the delays of all gates are not reduced below their nominal values.

Main advantages of this fault model are that the number of faults is limited to twice the number of lines in the circuit and almost all lines can be tested. Since the fault is tested along the longest propagation path, the system timing failures caused by the smallest localized delay defects or the accumulation of distributed delay defects can be detected. In transition fault model, a delay test is obtained along any arbitrary path because the size of delay fault is assumed to be large enough to be tested via any path through the fault site. For the gate delay fault model one must specify exact sizes of delay defects. Difficulties arise when accurate information on delays is not available. Transition and gate delay faults do not model the distributed delay defects along a target path. The line delay model, on the other hand, retains many advantages of the transition and gate delay fault models, while alleviating the major drawback of the path delay model (viz., too many paths to be tested and the low fault coverage).

2.4.2 Limitations of Line Delay Fault Model

In order to derive a line delay test for a given line, we first target the longest structural path. If that path is not robustly testable then we go for the next longest path and so on, until we find a test for the longest robustly testable path. The limitation in this approach is that in addition to this robustly testable path there could exist a nonrobust test for some longer path through the target line. In such a case, the robust test would miss a line delay fault that only causes the longer nonrobustly testable path to fail. To overcome this limitation one may include any possible nonrobust tests for all paths that are longer than the longest robustly testable path.

Another limitation of this model is that in case of certain distributed delay defects the derived tests will fail to detect some of the delay faults that are not targeted. We consider only one path through any given line for determining a line delay test. However, there may be some other paths of the same length (or shorter) through the target line, with distributed delay defects exceeding the permissible propagation delay. Consider the three paths shown in Figure 2. Suppose, all three paths have the same delay. Let us assume that paths 1 and 2 are the longest structural paths selected for testing lines A and B, respectively. Let us further assume that the smallest incremental delay that is detectable for each path is Δ (i.e., $\Delta = T_C - T_P$), where T_P is the nominal delay of each of the paths and T_C is the clock period. If the incremental delay of nodes A and B are $\Delta - \varepsilon$, where ε is small, then paths 1 and 2 will pass their tests. However, path 3 has a fault that is not detected by the test vectors though it is a detectable delay fault. The basic assump-

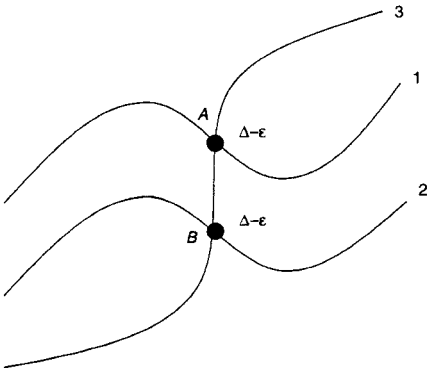

Figure 2: Limitation of the fault model

tion associated with the line delay fault model is that the delays of all gates are not reduced below their nominal values. More than one faulty gate can occur in practice due to correlation between delays of gates. In that case many gates in paths 1 and 2 will have increased delays and it is more likely that the tests for those faults will show failures.

There can be several ways of dealing with the situation depicted in Figure 2. When there are several longest paths of equal length through a target line, we can consider all such paths to increase the confidence level for the tests obtained. However, this can lead to a potentially large number of paths to be tested in some circuits.

2.5 Segment Delay Fault Model

Heragu *et al.* [13] have proposed a model that considers slow-to-rise and slow-to-fall defects on segments, whose length L, can be chosen from available statistics about the types of manufacturing defects. L can be as small as 1 (transition faults) or as large as the maximum logic depth (path faults). Once L is chosen, the fault list will comprise of all segments of length L and all paths whose entire length is less than L.

The segment delay fault model is an attempt to combine the advantages of the classical delay fault models while avoiding their limitations. Unlike the path delay fault model, this model can prevent an explosion of the number of faults to be considered. At the same time, a defect over a segment may be large enough to affect any path through it. This assumption seems more realistic than the transition delay fault model that requires the defect on a single line to be large enough to affect any path passing through it. Due to process variations, every gate in the circuit is affected and their delays increase only by a small amount. Again, randomly occurring defects are of very small sizes. Defects of this nature may produce delay faults only on longest paths. The segment delay fault tests may not be able to detect some of these defects.

FAULT MODELS	ADVANTAGES	LIMITATIONS
Gate Delay (Carter et al., ITC'87)	All gates can be modeled	* Distributed failures not considered * Exact defect size not possible
Transition (Waicukauski et al.'87)	Easy to model all gates	* Distributed failures not considered
Path Delay (Smith, ITC'85)	Distributed failures are considered	Impossible to enumerate all possible paths
Line Delay (Majhi et al., VLSI-D'96)	* All gates are modeled * Distributed failures considered * Better coverage metric * Additional fault coverage by using multi-pass technique	* Existence of non-robust test * May fail for some shorter paths (hard constraint)
Segment Delay (Heragu et al., VTS'96)	Considers general delay defect from spot to distributed failures	* Longest delay path may not be tested

Figure 3: Comparison of different fault models

In this scenario, the line delay fault model described in the previous section will be effective as it considers the longest sensitizable path through each line. However, a modification of the model that considers longest sensitizable paths through each segment will do well. But, when locations of defects are such that they affect only short paths, the tests may not detect some faults. Hence, in the ideal case, there is no substitute for a test having a 100% path delay coverage. Figure 3 summarizes the advantages and limitations of various delay fault models.

3 Path Delay Fault Classification

Although one can obtain high coverage of transition, line delay or segment delay faults, the effectiveness of these fault models is not proven. The gate delay model has remained largely unused due to its strong dependence on nominal delays, whose values may not be accurately known. The path delay model could be the ideal choice. However, typical circuits contain too many paths, many of which have no test. Recent research has produced several classifications of path delay faults in the hope of finding a small subset of paths that must be tested.

3.1 Lam et al.'s Classification

Lam *et al.* [17] show that path delay faults in a circuit can be partitioned into two sets. The occurrence of one or more delay faults from the first set can cause an increase in the circuit delay. This set must be tested for delay faults to certify correct operation. It includes all robust testable path delay faults (RD). Delay faults in the second set are termed as *robust dependent* (RD) delay faults. The occurrence of such a delay fault cannot increase the circuit delay unless some non-RD delay fault from the first set

also occurs. Thus, RD delay faults need not be tested to ensure that a manufactured circuit operates at the desired speed, *provided all RD faults have been tested.*

3.2 Cheng and Chen's Classification

Cheng and Chen [7, 8] classify path delay faults into four categories: (1) robust testable, (2) non-robust testable, (3) functional sensitizable and (4) redundant faults. For non-robust testable (equivalent to static sensitizable) faults and functional sensitizable faults, no single vector pair can detect such faults under *arbitrary* delay assignments. A functional sensitizable but static unsensitizable fault is a *false path* if any of the controlling-valued side-inputs of such a path is earlier than the corresponding on-input. If side-inputs of such a path are late, then the path may become *true* and affect circuit's timing.

Redundant path delay faults: A path delay fault is called *redundant* if the path is a false path (i.e., paths which cannot be sensitized by any vector pair) under all delay assignments. Otherwise, it is called *irredundant.*

3.3 Gharaybeh et al.'s Classification

Gharaybeh *et al.* [12] classify path delay faults into three categories: (1) singly-testable, (2) multiply-testable and (3) singly-testable dependent.

Singly-Testable: A path-delay fault is singly-testable (ST) iff there exists a delay test that guarantees its detection when it is the only path-delay fault in the circuit. This test is either robust or validatable non-robust or non-robust.

Singly-Testable Dependent: A path-delay fault is singly-testable dependent (ST dependent) iff it cannot propagate a late transition given the absence of certain ST faults. The ST dependent path delay fault set is a subset of the robust dependent (RD) set [17], but may include some functional sensitizable faults.

Multiply-Testable: A path-delay fault is multiply-testable (MT) iff it is neither ST nor ST-dependent. A test is possible only when several paths have simultaneous delay faults.

3.4 Sivaraman and Strojwas' Classification

Sivaraman and Strojwas [32] classify path delay faults into three categories: (1) primitive SPDF (single path delay fault), (2) primitive MPDF (multiple path delay fault) and (3) primitive-dependent. Primitive PDFs were first defined by Ke and Menon [15]. Informally, singly non-robustly testable paths belong to the set of primitive PDFs. Also, the set of paths, that is jointly non-robustly testable and minimal, belongs to the set of primitive PDFs. The circuit will exhibit fault free timing behavior for all input vectors and for all timing constraints

<table>
<tr><td>Lam et al (IEEE-TCAD'95)</td><td colspan="2">non-RD</td><td colspan="2">RD</td></tr>
<tr><td>Cheng and Chen (IEEE-TCAD'96)</td><td>robust testable</td><td>non-robust testable</td><td>functional sensitizable</td><td>redundant</td></tr>
<tr><td>Gharaybeh et al. (JETTA'97)</td><td colspan="2">singly-testable</td><td>multiply-testable</td><td>singly-testable dependent</td></tr>
<tr><td>Sivaraman and Strojwas (VLSI Design'97)</td><td colspan="2">primitive SPDF</td><td>primitive MPDF</td><td>primitive-dependent</td></tr>
</table>

Figure 4: Path delay fault classifications

larger than some given value (say, clock period T) iff it can be guaranteed that all primitive PDFs are absent for T. Moreover, the number of primitive PDFs in a circuit is considerably smaller than all PDFs. Sivaraman and Strojwas [32] refined the classification of paths proposed by Gharaybeh *et al.* [12]. Their primitive SPDFs are the same as ST faults. However, the primitive-dependent class is larger than the ST-dependent class. The primitive-dependent faults need not be tested if the circuit is guaranteed to be free from all primitive SPDFs and primitive MPDFs. The authors have exploited the fact that non-robustly testable SPDFs and MPDFs determine the stabilization time of the circuit outputs to extract primitive PDFs in an iterative fashion with maximal pruning and minimal computational overhead.

A comparative representation of the various classifications is given in Figure 4.

4 Conclusion

We have surveyed various delay fault models and their advantages and limitations. An efficient fault model that will result in a high fault coverage and low computational complexity still remains elusive. It is generally accepted that the path delay model is most comprehensive. We have presented various path delay fault classifications that are central to their real application. Understanding of delay fault coverage is still evolving. At this time, there is no agreement on using a fault model. Tests for selected critical paths are popular. Tools emphasize path timing analysis and delay fault simulation. Probably the next generation of tools will identify *false paths*, for which tests do not exist.

We have discussed delay fault models for combinational circuits. Test considerations for scan circuits are similar. However, an additional *hold* latch is required for every scan flip-flop to hold the first vector while the second vector is scanned in. The same fault models also apply to non-scan sequential circuits, with one difference. Additional robustness considerations are needed at the boundaries between time-frames. Two test methods have been proposed. In the *variable-clock* method [5], a slow clock is used except one rated-clock application. Thus robustness

conditions apply to just one time-frame boundary. In the *rated-clock* method [2], robustness conditions apply to all time-frames.

References

[1] P. Agrawal, V. D. Agrawal, and S. C. Seth. Generating Tests for Delay Faults in Nonscan Sequential Circuits. *IEEE Design and Test of Computers*. 10:20–28, March 1993.

[2] S. Bose. *Testing for Path Delay Faults in Synchronous Digital Circuits*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, December 1995.

[3] S. Bose, P. Agrawal, and V. D. Agrawal. Generation of Compact Delay Tests by Multiple Path Activation. In *Proc. International Test Conference*, pages 714–723, October 1993.

[4] J. L. Carter, V. S. Iyengar, and B. K. Rosen. Efficient Test Coverage Determination for Delay Faults. In *Proc. International Test Conference*, pages 418–427, September 1987.

[5] T. J. Chakraborty, V. D. Agrawal, and M. L. Bushnell. Delay Fault Models and Test Generation for Random Logic Sequential Circuits. In *Proc. 29th Design Automation Conference*, pages 165–172, June 1992.

[6] K. T. Cheng. Transition Fault Simulation for Sequential Circuits. In *Proc. International Test Conference*, pages 723–731, October 1992.

[7] K. T. Cheng and H. Chen. Delay Testing for Non-robust Untestable Circuits. In *Proc. International Test Conference*, pages 954–961, October 1993.

[8] K. T. Cheng and H. C. Chen. Classification and Identification of Nonrobust Untestable Path Delay Faults. *IEEE Transactions on Computer Aided Design*, 15:845–853, August 1996.

[9] K. Fuchs, F. Fink, and M. H. Schulz. DYNAMITE: An Efficient Automatic Test Pattern Generation System for Path Delay Faults. *IEEE Transactions on Computer Aided Design*, 10:1323–1335, October 1991.

[10] K. Fuchs, M. Pabst, and T. Rossel. RESIST: A Recursive Test Pattern Generation Algorithm for Path Delay Faults Considering Various Test Classes. *IEEE Transactions on Computer Aided Design*, 13:1550–1562, December 1994.

[11] M. A. Gharaybeh, M. L. Bushnell, and V. D. Agrawal. Parallel Concurrent Path-Delay Fault Simulation Using Single Input Change Patterns. In *Proc. 9th International Conference on VLSI Design*, pages 426–431, January 1996.

[12] M. A. Gharaybeh, M. L. Bushnell, and V. D. Agrawal. Classification and Test Generation for Path-Delay Faults using Single Stuck-Fault Tests. *Jour. Electronic Testing: Theory and Applications*, 11(1):55–67, August 1997.

[13] K. Heragu, J. H. Patel, and V. D. Agrawal. Segment Delay Faults: A New Fault Model. In *Proc. 14th IEEE VLSI Test Symposium*, pages 32–39, April 1996.

[14] Y. Ju and R. Saleh. Incremental Techniques for Identification of Statically Sensitizable Critical Paths. In *Proc. 28th Design Automation Conf.*, pages 541–546, June 1991.

[15] W. Ke and P. R. Menon. Synthesis of Delay-Verifiable Combinational Circuits. *IEEE Transactions on Computers*, 44(2):213–222, February 1995.

[16] S. Kundu. An Incremental Algorithm for Identification of Longest (Shortest) Paths. *Integration, the VLSI Journal*, 17:25–31, 1994.

[17] W. K. Lam, A. Saldhana, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. Delay Fault Coverage, Test Set Size, and Performance Tradeoffs. *IEEE Transactions on Computer Aided Design*, 14(1):32–44, January 1995.

[18] W. N. Li, S. M. Reddy, and S. K. Sahni. On Path Selection in Combinational Logic Circuits. *IEEE Transactions on Computer Aided Design*, 8:56–63, January 1989.

[19] C. J. Lin and S. M. Reddy. On Delay Testing in Logic Circuits. *IEEE Transactions on Computer-Aided Design*, 6:694–703, September 1987.

[20] A. K. Majhi, V. D. Agrawal, J. Jacob, and L. M. Patnaik. Line Coverage of Path Delay Faults. *IEEE Trans. on VLSI Systems*, under review 1997.

[21] A. K. Majhi, J. Jacob, L. M. Patnaik, and V. D. Agrawal. An Efficient Automatic Test Generation System for Path Delay Faults in Combinational Circuits. In *Proc. 8th International Conference on VLSI Design*, pages 161–165, January 1995.

[22] A. K. Majhi, J. Jacob, L. M. Patnaik, and V. D. Agrawal. On Test Coverage of Path Delay Faults. In *Proc. 9th International Conference on VLSI Design*, pages 418–421, January 1996.

[23] Y. K. Malaiya and R. Narayanaswamy. Modeling and Testing for Timing Faults in Synchronous Sequential Circuits. *IEEE Design & Test of Computers*, 1(4):62–74, November 1984.

[24] E. S. Park and M. R. Mercer. An Efficient Delay Test Generation System for Combinational Logic Circuits. *IEEE Transactions on Computer Aided Design*, 11:926–938, July 1992.

[25] I. Pomeranz and S. M. Reddy. An Efficient Nonenumerative Method to Estimate the Path Delay Fault Coverage in Combinational Circuits. *IEEE Transactions on Computer Aided Design*, 13:240–250, February 1994.

[26] I. Pomeranz, S. M. Reddy, and P. Uppaluri. NEST: A Non-Enumerative Test Generation Method for Path Delay Faults in Combinational Circuits. In *Proc. Design Automation Conference*, pages 439–445, June 1993.

[27] A. K. Pramanick and S. M. Reddy. On the Computation of the Ranges of Detected Delay Fault Sizes. In *Proc. Int'l Conference on CAD*, pages 126–129, November 1989.

[28] A. K. Pramanick and S. M. Reddy. On the Fault Coverage of Gate Delay Fault Detecting Tests. *IEEE Transactions on Computer Aided Design*, 16(1):78–94, January 1997.

[29] M. H. Schulz and F. Brglez. Accelerated transition fault simulation. In *Proc. Design Automation Conference*, pages 237–243, June 1987.

[30] M. H. Schulz, F. Fink, and K. Fuchs. Parallel Pattern Fault Simulation of Path Delay Faults. In *Proc. 26th Design Automation Conference*, pages 357–363, 1989.

[31] M. H. Schulz, K. Fuchs, and F. Fink. Advanced Automatic Test Pattern Generation Techniques for Path Delay Faults. In *Proc. 19th Fault Tolerant Computing Symposium*, pages 44–51, July 1989.

[32] M. Sivaraman and A. J. Strojwas. Primitive Path Delay Fault Identification. In *Proc. 10th International Conference on VLSI Design*, pages 95–100, January 1997.

[33] G. L. Smith. Model for Delay Faults Based upon Paths. In *Proc. International Test Conference*, pages 342–349, September 1985.

[34] J. A. Waicukauski, E. Lindbloom, B. K. Rosen, and V. S. Iyengar. Transition Fault Simulation. *IEEE Design & Test of Computers*, 4:32–38, April 1987.

[35] S. H. Yen, D. H. Du, and S. Ghanta. Efficient Algorithms for Extracting the K Most Critical Paths in Timing Analysis. In *Proc. 26th Design Automation Conf.*, pages 649–654, July 1989.