

DELAY FAULT MODELS AND METRICS

Eduardas Bareiša, Vacius Jusas, Kęstutis Motiejūnas, Rimantas Šeinauskas

*Software Engineering Department, Kaunas University of Technology
Studentų 50-406, LT-51368 Kaunas, Lithuania*

Abstract. The delay fault testing has become an important part of the overall test development process. But delay fault testing is not so mature as stuck-at fault testing. The paper surveys various delay fault models, their advantages and limitations. The current trends in test pattern generation for delay faults are analyzed, too. The test pattern generation is directly related to the coverage metrics. The coverage metrics mainly tend to evaluate the quality of path delay fault patterns. The focus is made on two groups of the metrics: non-enumerative methods and statistical methods. The non-enumerative methods rely on the traditional counting of the paths, which are tested under robust, non-robust, or functional sensitization criteria. The statistical methods relate their computations to the parameters under process variation. The basis of the methods is their weakness. There is still no accepted common metric to evaluate the quality of the delay test patterns. On that base, the paper suggests a new approach to the evaluation of the quality of the delay test patterns. This approach stands on the new term – test confidence level (the ground level, the first level, the second level and etc.). The procedure of the evaluation of test quality for delay faults is presented, too.

1. Introduction

The main objective of traditional test development has been the achievement of high stuck-at fault coverage. The increasing design complexity and reduced error margins in semiconductor manufacturing are forcing design and test engineers to take a new look at fault models. Rapidly shrinking feature sizes raise the specter of new types of defects, and increasing gate counts have increased the number of locations where such defects can occur. The presence of some random defects does not affect a circuit's operation at slow speed while it may cause circuit malfunction at rated speed. This kind of defect is called the delay defect. With ever shrinking geometries, growing density and increasing clock rate of chips, delay testing is gaining more and more industry attention to maintain test quality for speed-related failures. The purpose of a delay test is to verify that the circuit operates correctly at a desired clock speed. Although application of stuck-at fault tests can detect some delay defects, it is no longer sufficient to test the circuit for the stuck-at faults alone. Therefore, delay testing is becoming a necessity for today's integrated circuits.

Delay testing techniques are either based on a variable clock scheme or a rated clock scheme. The most commonly used is the variable clock scheme. Consider the combinational circuit shown in 0(a). In the variable clock scheme, two clocks are required to separately strobe the primary inputs and outputs. In a two-pattern test (V_1, V_2), the first pattern V_1 is applied to the primary inputs at time t_1 and the second

pattern V_2 at time t_2 (0(b)). The shaded area represents the amount of time required for the faulty circuit to stabilize after the application of V_1 . The circuit response is observed at time t_3 . This two-pattern test is used to determine if the propagation delay of a path being tested exceeds the time interval $(t_3 - t_2)$. The time interval $(t_3 - t_2)$ is the maximum allowable path delay for the rated frequency of operation. If we assume that no path delay in a faulty circuit exceeds twice the clock period, then $(t_2 - t_1)$ should be at least twice the interval $(t_3 - t_2)$. Such a delay testing methodology increases the test application time, but it makes a test generation and fault simulation easier.

In the rated clock scheme, all input patterns are applied at the rated circuit speed using the strobe for the primary inputs and outputs (0(c)). All the path delays in the fault-free circuit are assumed to be less than the interval $(t_2 - t_1)$. However, the path delays in faulty circuit may exceed this interval. Therefore, the logic transitions that arise at time t_1 due to the pattern pair (V_0, V_1) may still be propagating through the circuit during the time interval $(t_3 - t_2)$. In addition, other transitions may originate at time t_2 due to the pattern pair (V_1, V_2). If we assume as before that no path delay in the faulty circuit exceeds twice the clock period, then signal values during interval $(t_3 - t_2)$ depend on three patterns (V_0, V_1, V_2). That makes test generation and fault simulation more complicated. Nevertheless, the rated clock scheme is quite often used in the industry. But, all the path delay faults that can affect the rated-clock operation of the circuit are testable by the variable-clock method. Also, all path

delay faults that are untestable by the variable-clock method are, in fact, untestable by the rated-clock method. However, some faults tested by the variable-

clock method may be incapable of affecting the rated-clock operation. Therefore, test generation and fault simulation use a variable clock scheme.

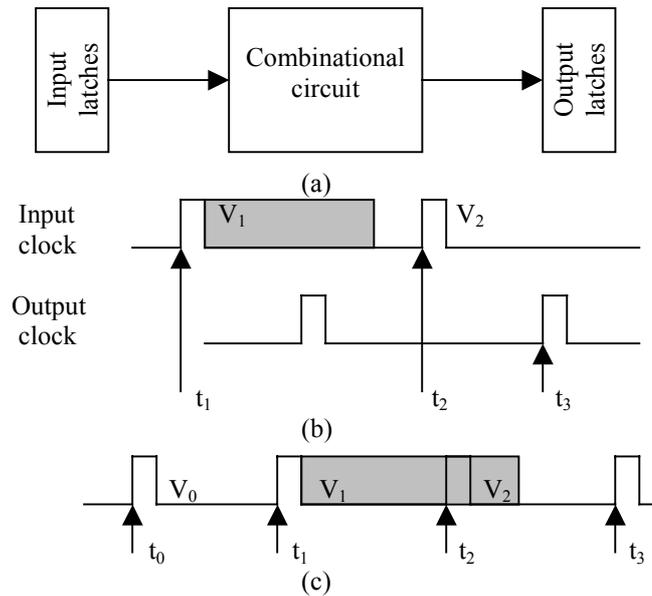


Figure 1. The clocking schemes for combinational circuit testing

This paper is organized as follows. In Section 2, various fault models used in delay fault testing are presented. In Section 3, we analyze the current trends in test pattern generation for delay faults. In Section 4, we discuss the coverage metrics of delay faults. In Section 5, we propose a new metric for delay faults. Section 6 summarizes the paper.

2. Delay Fault Models

Three classical fault models are developed to represent delay defects (transition fault, gate delay fault, path delay fault) [0]. Sometimes there is used the fourth type of delay fault model – segment delay fault, which is intermediate to the gate delay and path delay faults. Each of these fault models has their own limitations and advantages that are discussed in some detail in this paper.

2.1. Transition Fault Model

The transition fault model [0] associates every line of the circuit with two transition faults: a *slow-to-rise fault* (rising fault) and a *slow-to-fall fault* (falling fault). The slow-to-rise (fall) transition fault temporarily behaves like a 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 stuck-at fault. Tests that meet these conditions are referred to as *conventional transition fault tests* [0]. Transition faults

model defects for which the delay is large enough to cause a logical failure when a signal propagates along any path through the site of the fault. The gross delay assumption simplifies a test generation for transition faults. The primary weakness of the transition fault model is that the minimum achievable delay fault size is difficult to determine. The longer the timing of the paths used to detect the faults, the smaller the sizes of defects they can capture. The second weakness of transition fault is that many small gate delay faults, each undetectable as a transition fault, can give rise to a large path delay fault. Without explicitly targeting specific paths, employing multiple-detections becomes an alternative for enhancing the quality of transition test set effectiveness [0].

2.2. Gate Delay Fault Model

The gate delay fault model [0] is a more general case of the transition fault model, because it considers the fault size. The gate delay fault model assumes 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 in the propagation of a rising or falling transition from the gate input to output. The set of faults considered includes numerical delay information. The gate delay fault model to separate from the transition fault model is called a *small gate delay fault model* [0, 0]. In general, a test for a given gate delay fault f can only detect delay defects on f with sufficiently large sizes. A commonly used metric called the *detection threshold* [0, 0] is used to measure the capability of a test for detecting a gate

delay fault f . The *detection threshold* is the minimum size ϵ such that if the size of f is greater than ϵ , f is guaranteed to be detected by a pair of two test patterns. But this metric is not practical for test pattern generation based on gate delay fault model 0.

2.3. Path Delay Fault Model

The path delay fault model was introduced by Smith 0. This model has received greater attention than the transition fault and gate delay fault models. A lot of works are devoted to various aspects of test generation and fault simulation of path delay faults. It is not possible to enumerate all of them.

In the path delay fault model, any path with a total delay exceeding the system clock interval is said to have a path delay fault. This model 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 (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. A path delay fault is represented as a sequence of falling ($1 \rightarrow 0$) or rising ($0 \rightarrow 1$) transitions along a physical path from a primary input to a primary output in a circuit. To observe path delay defects, it is necessary to create and propagate transitions in the circuit. This requires application of a pair of patterns. The first pattern stabilizes all signals in the circuit, while the second one causes the desired transitions. The main drawback of path delay fault model is the huge number of paths, which is not possible to enumerate.

Various classifications are used for path delay faults 0, 0, 0, 0. We will present the classification 0, which was introduced firstly and is used most frequently 0. Under this classification, path delay faults are classified as robustly testable, non-robustly testable, functionally sensitizable, and redundant. The classification in 0 is based on the signal values that the applied test (pair of patterns) induces on the inputs of the gates along the path where the fault is encountered. The *on-input* of a gate is defined as the one along which a rising or falling transition is to be propagated to the output of the gate. The remaining inputs of a gate are defined as the *off-inputs*. We denote the controlling and non-controlling values of a gate by cv and ncv , respectively. A path delay fault is tested robustly under some test if at every gate along the path the off-inputs assume either stable values ncv when the on-input has an $ncv \rightarrow cv$ transition, or $x \rightarrow ncv$ values when the on-input has a $cv \rightarrow ncv$ transition (x denotes don't care). A path delay fault is tested non-robustly if the on-input is allowed to have an $ncv \rightarrow cv$ transition while the off-inputs have early arriving $cv \rightarrow ncv$ transitions. Functional sensitization includes an additional relaxation when several gate inputs, including the on-input, have an $ncv \rightarrow cv$ transition. In this case, the inputs that have the $ncv \rightarrow cv$

transition are multiply sensitized. A path delay fault is redundant if there is no pair of patterns that satisfies at least one of the above conditions along some gate on the path. Redundant path delay faults do not need to be tested since they never affect the timing performance of a circuit under any possible delay assignment.

2.4. Segment delay fault model

The *segment delay fault model* proposed in 0 is an extension of the gate delay fault model. A segment delay fault models the delay defects on a segment, which is a subpath of L consecutive lines, where $L \geq 1$. L can be chosen based on available statistics about the types of manufacturing defects. Once L is chosen, the fault list comprises of all segments of length L and all the paths whose entire length is less than L . It is assumed that every path containing a segment delay fault has a delay greater than the maximum allowed value. Tests to detect segment delay faults should propagate a transition launched at the beginning of a segment through the segment to an observed output, possibly through a multi-path. When L is selected to be 1, segment delay faults are essentially transition faults. Although segment delay faults alleviate part of the capability to detect small size defects of transition faults by testing longer segments instead of lines, a segment delay fault can still be activated and propagated through any path passing through it and hence a segment fault test may not detect small delay faults. Additionally, the number of segment delay faults increases with increasing segment lengths.

3. Current trends in test pattern generation for delay faults

All the presented fault models have their advantages and weaknesses. The most different models among them are transition fault model and path delay model. On one hand, transition fault patterns are considered to be more effective for large size delay defects, which can happen randomly at any site of a circuit. On the other hand, path delay test patterns aim to detect small size delay defects on a selected set of long timing paths. It is then hoped that the combination of these two orthogonal strategies can capture most of the delay defects and ensure the circuit performance. This direction was recognized very soon after introduction transition and path delay fault models. Pramanick and Reddy 0 proposed to generate a robust or non-robust test for a longest path passing through the target fault. This method does not take testability into consideration when selecting a longest path containing the fault site. For many transition faults no test can be found for a selected longest structural path passing through the fault site. The fault coverage achieved was typically low. Therefore, some years passed until the next attempt in this direction, which was the combination of gate delay fault and path delay fault models

0. Mahlstedt [0] presented delay test pattern generator (DELTEST), which tried to generate an optimal delay test. An optimal delay test for a gate delay fault is a test that sensitizes the longest functional path through the fault site. It is an NP-hard problem to generate optimal delay tests. Therefore, there was used a fairly simple procedure: generate a test for gate delay fault and then try to generate a better test. The algorithm is complete in the sense that it will generate an optimal delay test or prove the delay fault as redundant if given enough time. So, as we understand, it may take long hours or days.

The third attempt in this direction was the combination of the transition and path delay faults to define a line delay test [0]. Majhi et al. proposed a coverage metric and a two-pass test generation method for path delay faults in combinational logic circuits. The coverage is measured for each line with a rising and a falling transition. The test, called “line delay test”, is a path delay test for the longest sensitizable path producing a given transition on the target line. The maximum number of tests (and faults) is limited to twice the number of lines. Two-pass test generation procedure begins with a minimal set of longest paths covering all lines and generates tests for them. Fault simulation is used to determine the coverage metric. For uncovered lines, in the second pass, several paths of decreasing length are targeted. The main drawback of the method in [0] is that only single robustly testable paths are used to cover lines in the circuit. Consequently, the transition fault coverage is much lower than that of the conventional transition fault tests, which allow the use of multi-paths and non-robust test to detect transition faults.

The blend of transition fault and path delay tests was finally recognized not long ago [0, 0, 0, 0, 0, 0, 0, 0, 0]. [0, 0, 0] constitute a group of works that ensure the selection of the longest potentially testable path through every line in the circuit.

Shao et al. [0] suggested for obtaining a high quality transition fault test set using reasonable run times to generate initially a conventional transition fault test set and then to augment it by tests from a path-oriented test generator POTENT, which seeks to propagate transition faults along paths with maximal delays. The transition fault test set is simulated and all the faults that are detected with sufficient quality can be dropped. The remaining faults are targeted using POTENT. The overall flow of POTENT includes three phases. For each fault targeted in Phase I, POTENT attempts to find a longest D-value sensitizable single path through the fault to test the target fault. In the second phase, POTENT searches for a longest D-value sensitizable single (partial) path to propagate the fault effect. All the remaining faults are targeted in the third phase, where multi-path activation and propagation are used to detect each fault.

Gupta and Hsiao [0] proposed quite opposite approach for delay tests generation than that described in [0]. The employed approach is clearly expressed in

the objective of the paper [0] – in order to have high quality delay test, it needs to have high robust path coverage, high non-robust path coverage and high transition fault coverage. Test generation consists of three phases: robust test generation, non-robust test generation and transition test generation. The first step towards the test generation for robust paths is to enumerate the robustly testable paths for which tests are to be generated. In this phase, test patterns are generated that will only launch the targeted transitions. An implication-based approach similar to that used to enumerate robust paths is used to first drop all the paths that cannot be tested non-robustly. But the number of paths dropped depends on the number of robust paths detected. Higher robust path coverage generally translates to more non-robust paths dropped, because non-robust cannot overlap with robust paths. In the second phase, test pattern generated is such that it excites as many undetected faults F as possible and the corresponding next vector excites as many opposite of F as possible and also propagates them to k levels ahead from the fault site; where k is the iteration number within phase II. Detected transition faults are dropped after two phases. The third phase targets the remaining hard-to-detect transition faults and is a fault dependent phase. Unlike the other two phases it adds a pattern pair for each detected fault.

Qiu and Walker [0] stated that testing the K longest paths through each gate (KLPG) in a circuit detects the smallest local delay faults under process variation. Process variation occurs everywhere in an integrated circuit, even for a single gate, it is hard to determine which path is the actual longest path passing through it. Therefore, testing only one path through each gate cannot guarantee the detection of smallest local delay defects. This work defines a test pattern generation methodology to find the K longest testable paths through each gate in combinational circuit. Many techniques are used to significantly reduce the search space. In the presented experiments, K varies from 1 to 20.

Yang et al [0] provided a test generation tool targeting on a path-oriented transition fault model. Under this model, a transition fault is detected through the longest sensitizable path. False-path pruning technique is used to identify the longest sensitizable path through each fault site. The paper differs from all the other works in this area in the two following ways. First, test generation is based on a circuit Boolean Satisfiability solver. Second, a new false-path pruning technique employs the features of the solver.

Lu et al. [0] moved further than in [0] - to detect the smallest delay fault, it is necessary to test all the longest paths through the fault site. The paper presents a method to generate the longest path set for delay test under process variation. To capture both structural and systematic process correlation, linear delay functions are used. The paper considers only inter-die systematic process variation, such as variations on gate length, metal width, metal thickness, and inter-layer-dielectric

thickness. A path-pruning technique is proposed to discard paths that are not longest under process variation, resulting in a significant reduction in the number of paths. The proposed method can be applied to any process variation as long as its impact on delay is linear.

Gupta and Hsiao [10] introduced a new transition fault model (ALAPTF), which combined the features of the transition delay fault and segment delay fault models. The objective of this model is to accumulate the effect of small delays on a particular node, which can then be tested by the traditional transition fault model. This accumulation of small delays can be modeled by launching each transition fault as late as possible at the fault site. The notion of ALAPTF was implemented by making sure that a transition fault is launched via one of the longest robust *segment* ending at the fault site. Note that the segment needs not start from a primary input. The fault is propagated to a primary output by one of the longest paths starting from the fault site.

4. Coverage Metrics of Delay Test Patterns

The previous section presented the current trends in the delay test pattern generation. As we can conclude from that section, the usual way to generate the delay test patterns is to propagate transition faults through the longest sensitizable path of the circuit. Such an approach has the following two advantages. First, the transition faults cover all the circuit and enable the detection of large delay defects. Second, sensitization of the longest testable path enables the detection of small delay defects. The transition faults have their "golden" coverage metric, while the path delay faults have no such a "golden" metric. The transition faults coverage metric cannot evaluate the quality of the blend of the transition faults tests and path delay faults tests, because it considers only transition faults. There is a need to have one common metric, because everyone claims when presents a new method that combines transition fault tests and path delays tests that his method is the best. A lot of works are devoted to propose their own delay tests quality evaluation metric. They can be broadly divided into four categories: enumerative methods, non-enumerative methods, statistical methods and miscellaneous methods. Let's analyze them. We do not include the enumerative simulators into our analysis because they are inapplicable for circuits with a large number of faults [10]. The non-enumerative methods can be further divided into two groups: approximate and exact. We leave out of our scope the approximate methods, because the computed estimate is pessimistic, i.e., the true fault coverage is not smaller than the estimated one [10].

4.1. The non-enumerative exact methods

The method in [10] can be considered as the first non-enumerative exact method. The method computes the exact fault coverage by implicitly numbering every path-delay fault, and takes advantage of consecutively numbered faults. It achieves this by representing a set of consecutively numbered faults as a closed interval. This technique is useful as long as large number of detected faults is consecutive. The method has the drawback that if a single pattern (or a set of patterns) detects a very large number of nonconsecutive faults, then the method ends up enumerating every fault. Clearly, this is impractical for circuits with very large numbers of paths.

The method in [10] is an exact fault simulator that uses a path-status graph (PSG) to capture information pertaining to detected faults. A vertex in PSG corresponds to a primary input, fan-out node or primary output of the circuit being simulated. Each edge between two nodes in PSG represents a unique sub-path in the circuit. Structurally, there is a one-to-one correspondence between a path in the circuit and a path in PSG. As test vectors are simulated, the exact fault coverage is maintained by splitting appropriately chosen vertices in PSG. The space requirements of PSG can grow exponentially with the number of test vectors. To alleviate this problem, the method incorporates a procedure for recombining the split nodes. As the simulation proceeds, the PSG expands and shrinks. In addition, all tested and untested path delay faults can be identified since there is a one-to-one correspondence between paths in the PSG and in the original circuit. These path delay faults are, however, left implicit in the PSG, and may be extracted if and when desired. Thus, the PSG is a compact representation of the tested and untested fault lists, which can be very large. The size of the PSG cannot be expressed precisely for any given circuit and any iteration in the fault simulation. In general, this size depends on the size of the original circuit, the number of paths, the number of tested faults, and the size and ordering of the test set. Even though the maximum size of PSG can be exponential in the number of circuit nodes, PSG technique has theoretically an exponential time complexity, too.

The method in [10] proposes a technique for computing exact fault coverage for any fault model. It consists of appropriate formation and counting of colors. Each color represents a set of faults, and the definition of colors varies according to the fault model. A color is essentially a set of faults if the sample space of faults is exponential then a color represents a set of faults in an implicit manner, without enlisting the faults. The technique utilizes the two aspects on which the fault coverage for any model depends, the circuit lines and the patterns applied to the circuit. The whole technique consists of one single forward traversal of the underlying circuit. The colors are formulated on demand at each circuit line (node) during this

topological traversal. The main difference of this method from the method in 0 is in the modification of information on the fan-outs. In the color counting technique, the colors on the fan-out stems are divided into subsets and these new colors are copied onto the fan-outs. This increases the color count of the color list. The color data structure eats up a lot of storage space and also has a considerable share in the time complexity, too.

The method in 0 formulates the path delay fault coverage problem as a combinatorial problem that amounts to storing and manipulating sets using a special type of binary decision diagrams, called zero-suppressed binary decision diagrams (ZBDD). ZBDDs are BDDs based on a new reduction rule, which eliminates all the nodes whose one-edge points to the zero-terminal node. The basic path delay faults grading algorithm is a two-phase process for every test in the test-set. The test is simulated during the first phase to find the path delay faults tested under a single test. The second phase consists of storing the path delay faults identified in the previous phase in a ZBDD format and updating the set of total path delay faults tested with the new faults. The simulation procedure requires two topological traversals of the input circuit. A forward traversal performs the actual simulation for the test patterns and a backward traversal marks the lines and nodes (gates and primary inputs) along the sensitized paths. After all of the test patterns are processed, the procedure finds the total number of tested path delay faults by computing the cardinality of the final ZBDD. An extension to the basic scheme is presented to handle the case when the ZBDD representing all the tested path delay faults cannot be stored. The proposed solution uses the notion of independent cuts to perform a virtual circuit partition while ensuring a minimal loss in the fault coverage accuracy. The main advantage of this method against the methods presented above is the possibility to handle quite large test sets (673 286 number of patterns).

The method in 0 focuses on combinational circuits that contain very large number of path delay faults and are checked by very large test sets. A directed graph p -graph is constructed in which every segment where robust propagation occurs is a directed arc and primary inputs, gate outputs and fan-out branches are nodes. A separate p -graph is constructed for every pattern pair. Paths in a p -graph can be counted starting from the primary outputs and back-tracing to primary inputs. However, counting the number of tested paths for a test set T is complicated because a path can be tested by multiple test patterns. The principle of inclusion and exclusion is used to obtain the exact fault coverage. While the worst-case complexity of this approach is exponential in the number of patterns, several empirical characteristics of path delay coverage are exploited to develop an approach with significantly lower average case complexity. Experimental results demonstrate that the proposed approach can be used to perform path delay fault simulation for circuits

with large numbers of paths for 50,000 patterns (non-robust) and 1,000,000 patterns (robust). But a quite large amount of time is used (from 45 minutes for c880 to 19 hours for c7552 – 50, 000 non-robust single input transition vectors).

The existing non-enumerative methods have several drawbacks. These methods rely on the traditional metric of fault coverage that counts the number of the paths, which are tested under robust, non-robust, or functional sensitization criteria. Based on this metric, testing p long paths has the same fault coverage as testing p short paths, which does not reflect the real test quality. In addition, since the total number of paths is exponential in the number of gates, the methods have high requirements for memory and use a large amount of time for computations.

4.2. Statistical methods

The above described non-enumerative test quality evaluation methods do not take into account delay fault sizes, which are a consequence of the fabrication process. The likelihood of each path exceeding the timing constraint is different. These fabrication-process effects are incorporated into the statistical delay faults coverage metrics. Sivaraman and Strojwas 0 proposed very general statistical coverage metric, which is defined as

$$\text{Statistical coverage (test)} = \frac{100 \times \text{Probability (test detects a delay fault | fabricated chip is faulty)}}{1} \quad (1)$$

The metric defines the test coverage in terms of the probability that if the fabricated chip is faulty, the test set will detect a delay fault in the chip. The proposed metric for the delay fault coverage of a test set is general enough to be applicable to any delay fault model. However, the definition by itself does not suggest any scheme to determine the coverage of a given test set under a given delay fault model. The methods that propose statistical delay fault coverage metrics differ only in the mapping this abstract metric (1) to concrete delay fault model.

Sivaraman and Strojwas 0 presented distributed path delay fault model, in which path delay faults are a result of excess component delays caused by variations in fabrication-process parameters such as transistor channel widths, transistor channel lengths, oxide thickness, doping concentration, etc. The excess delay is assumed not to be localized at any one component – component delays vary in a correlated manner due to across-the-chip fabrication-process parameter variations. This cumulative sum of excess component delays along one or more paths results in delay faults. This situation could arise in spite of the fact that each circuit component meets its individual delay specifications – because of correlation between component delays due to similar fabrication-process effects, the sum of these delays along paths might violate a timing constraint. These fabrication-process

parameter variations are global, i.e., the fabrication-process parameters vary by the same amount all across the chip. A single variable is sufficient to represent each fabrication-process parameter variation in all the components of the chip. The correlated component-delay model models path delay faults caused by global distributed fabrication-process parameter variations. Authors of the approach in 0 recognized that the statistical coverage estimates obtained are obviously dependent on the kind of delay faults modeled and the distribution of the fault sizes. Thus, one would expect different results for localized delay defects arising from local fabrication-process disturbances, e.g., a local defect on a metal interconnect resulting in a narrow highly resistive interconnect segment, which retains component functionality, but slows it down. Different results can also be expected for distributed path delay faults as a result of different fabrication-process parameter variations for each component in a chip, e.g., the channel widths of different devices in the same chip varying by different amounts.

The method in 0 first prunes the set of all paths by the false-path-aware statistical timing analysis. In the process, those paths that are unlikely to cause a delay problem are removed. Paths that cannot be sensitized by any test are also removed. The output of the false-path-aware statistical timing analysis tool is an universal path candidate set (U). The size of U depends on the cutoff period T, and so does the quality of the path coverage metric developed based upon U. T selection is based upon results from transition fault testing. T essentially denotes the *shortest* timing length among the paths. The higher quality the transition patterns are, the larger the T will be and the smaller the size of U is. Only a subset of U (say S) is selected for testing. The focus is shifted on the quality of selected paths, instead of the quality of tests generated based upon those paths. Metric involves only static analysis and is pattern-independent. The complete procedure of static quality evaluation scheme is as follows. In each Monte Carlo sampling run, first a circuit instance with cell/interconnect delays is generated according to the delay distributions characterized through Monte Carlo SPICE earlier. Two analysis steps will then evaluate this instance: "statistical analysis of S" and "statistical analysis of U-S". The "statistical analysis of S" is to check if there is any path in S (on the given instance) longer than the testing clock C. If so, then this instance is said to be faulty and covered by S (Covered). The "statistical analysis of U-S" performs a similar analysis on the set of U-S and reports the number of faulty instances not covered by S (Noncovered). At the end, scheme will calculate the probability of a faulty path captured by S based upon all the instances statistically produced. The proposed metric has a very narrow application. Its objective is the critical path selection for delay test patterns generation.

The method in 0 presents a coverage metric for estimating the test quality with respect to timing

defects under process variations. The timing defect model in pattern-selection framework is similar to a gate delay fault model and is called a *pin-to-pin timing defect* model. The base of the proposed metric is the critical probability of a pin-to-pin segment s under pattern i , $CP_i(s)$: the probability that the pattern i can generate a circuit delay exceeding the clock period if there is a timing defect on the pin-to-pin segment s . To consider the topological overlap of detecting timing defects for a set of patterns, the critical probability $CPp(s)$ of a pin-to-pin segment s under a set of patterns P is defined. The $CPp(s)$ is obtained by the following equation:

$$CP_p(s) = 1 - \prod_{i \in P} (1 - CP_i(s))$$

The premise is assumed that timing defects may locate uniformly over all pin-to-pin segments. Therefore, the critical probability of each pin-to-pin segment is treated with equal importance. The coverage metric of a set of patterns P is formulated as the summation of each $CPp(s)$. The most crucial part of computing the coverage metric is to obtain the $CP_i(s)$ of each pin-to-pin segment s and each pattern i . For each pattern i , a Monte-Carlo-based timing analyzer is applied to generate timing configuration samples based on the statistical timing model. Most of computation comes from the statistical dynamic timing analysis. The computation time varies from 77 s for C880 to 7084 s for S35932. The proposed metric is used to select test patterns from a larger set of test patterns. In this methodology, the job of ATPG is not to produce the final test but to produce a superset of test patterns such that they can be further processed in the pattern selection step.

The method in 0 uses a delay fault model, which assumes that delay faults are caused by global delay faults only or the combination of local and global delay faults. Using this delay fault model, the general metric introduced in 0 is translated into formula to compute the detection probability (DP) for fault site i with local extra delay Δ (the size of the local delay fault): $DP_{i,\Delta}$, $DP_{i,\Delta}(t) = P(\text{at least one tested path through } i \text{ is slow} \mid \text{at least one path through } i \text{ is slow})$. For fault site i with an arbitrary Δ , the DP for site i is computed as:

$$DP_i(t) = \int_{\Delta > \Delta_{0,i}} DP_{i,\Delta}(t) \cdot p_i(\Delta) d\Delta$$

where $\Delta_{0,i}$ is the value of local extra delay below which there is no delay fault, $p_i(\Delta)$ is the probability density function of Δ at fault site i , and is computed using the probability density function of delay caused by physical defects, such as resistive opens. Then the overall fault coverage for test set t is:

$$FC(t) = \sum_i DP_i(t) \cdot w_i \times 100\%$$

where w_i is the weight for fault site i ($\sum_i w_i = 1$). w_i depends on the location of the fault. The fault sites with many long paths through them are more likely to cause delay faults than the fault sites, which have only short paths through them. w_i is also sensitive to the ratio of local/global delay faults. If the ratio is high, the weights are almost equal for all fault sites. If it is low, the fault sites with only short paths passing them through can have weights close to 0. In this work, equal weights are used for simplicity. If the path delays are not independent variables (and in reality, they are not), the DP computation is dependent on the delay space. Therefore, if accurate correlation information is not known, the DP computation is not easy. To solve this problem, two extremities are assumed. If no correlation is assumed, path delays are independent variables. This assumption results in the lower bound of fault coverage. If 100% intra-die process correlation is assumed (only inter-die process variation is considered), the upper bound of fault coverage is computed. The fault efficiency is computed because the fault sites with no transition fault test are not included in the computation and the false paths are eliminated by the KLPG test generator. The presented method has the following two limitations. Intra-die process variation and multi-path sensitization must be considered because some local delay faults are only detectable through a set of paths. A local delay fault can be caused by resistive shorts as well as opens, which are only assumed in this work. The fault coverage metric for resistive shorts is more complicated.

The use of statistical methods has several deficiencies. The first deficiency is their dependency on the parameters considered under process variation. The other parameters would be used, the other results would be obtained. There is no universal set of parameters defined. The second deficiency is the use of statistical timing analyzer, which is slow.

4.3. Miscellaneous methods

The method in 0 estimates fault efficiency for path delay faults based on untestable path analysis rather than fault coverage. Fault efficiency is defined as the percentage of detected faults for all the testable faults. The proposed method first identifies a subset of the untestable paths by using the partial path sensitization. The paths not identified as untestable are potentially testable paths. Although the potentially testable paths may contain untestable ones, they are too many to check whether testable or not one by one. In order to obtain the total number of the untestable paths more quickly, a part of potentially testable paths is sampled. The samples only are checked whether they are testable or not. The total number of the untestable paths is presumed from the derived percentage of the untestable paths in the samples, while the paths detected by the generated test patterns are computed with fault simulation. The proposed method could estimate fault efficiency only approximately. An

advantage of the proposed method is that fault efficiency can be computed in a reasonable computation time.

A quite large team of authors from different universities and several industrial companies searched for a new metric of delay faults to resolve the mismatch between the transition faults and the timing defects 0. The main finding was that transition fault coverage could be used to predict the coverage of statistical timing defects, if the right subset of the transition faults is selected to calculate the fault coverage. The hard-to-detect faults constitute the right subset of the transition faults. However, this finding was not defined as new metrics of delay faults because it was valid only for two circuits from three circuits, which were used in the experiments. The paper finally recognized that it requires further investigation in the future.

5. New Coverage Metric of Delay Test Patterns

The presented analysis of coverage metrics of delay test patterns showed that there still are no single unique metrics. From Section 3, we can conclude that the usual way of test pattern generation for delay faults is the transition fault testing, and to complement transition fault testing, usually a set of timing critical paths are selected for explicit testing. Transition fault tests are the most effective when applied to a highly optimized circuit 0, 0. As a result, the quality of multiple transition test sets combined is able to catch up with the quality of an ideal path delay test set. The problem of using multiple transition test sets lies in its high cost.

In transition fault testing, transition fault path does not always contain the longest path passing through the site of the fault. To improve transition test quality, it is intuitive to improve the quality of transition fault paths by sensitizing the path with longest propagation time passing through each site. However, the quality of the paths will be highly dependent on the accuracy of the timing model. In the deep sub-micron domain, the accuracy of the timing model is not easy to define, and traditional timing analysis will often fail to provide a good delay prediction. Moreover, because process variation occurs everywhere in an integrated circuit, even for a single gate, it is hard to determine which path is the actual longest path passing through it. Therefore, testing only one path through each gate cannot guarantee the detection of the smallest local delay faults. Testing the K longest paths through a fault site increases the fault detection probability under process variation, because it increases the probability that the actual longest path is tested 0.

The strength of transition fault patterns is their better ability to capture large-size defects and full the topological circuit coverage. The strength of path delay patterns is their better ability to capture small size defects, which are distributed along the circuit. Therefore, the coverage metric of delay test patterns has to evaluate the contribution of transition fault

patterns and the contribution of path delay patterns. The only metric in 0 from all the analyzed metrics that consider path delay fault coverage deals with local delay faults, which are modeled as transition faults. The authors of this metric also suggested a test generation strategy 0:

1. Apply transition fault tests to detect large local delay faults (from industrial experience most local delay faults are large).
2. Apply at-speed test to one of the longest paths through each gate or line, because this is the second largest coverage loss factor.
3. Test more possible longest paths.

But the coverage metric in 0 has the following two weaknesses. First, it is dependent on the parameters considered under process variation. The only resistive opens are considered as a kind of possible defects. Second, this metric does not count explicitly how many times the longest path passes through each site.

Based on the above analysis we suggest introducing a new term for definition of delay test patterns coverage – test confidence level. The delay test patterns have the ground level of test confidence if it is defined their transition fault coverage. The delay test patterns have the first level of test confidence if the longest path passes through each site at least once. The delay test patterns have the second level of test confidence if the longest path passes through each site at least two times and so on.

Let's analyze the proposed metric. According to the metric the transition fault coverage has to be determined firstly. If transition faults are fully covered (coverage 100%), there is a meaning to consider does the test pass the first level of test confidence. The coverage 100% does not mean that every transition fault has a path passing through the fault site. Some transition faults may be checked through multi-paths. If the test passes the first level of test confidence, then it has to be checked for the second level of test confidence and so on.

The coverage metric usually has two purposes. Firstly, it is a guide for test pattern generation. Metric defines test pattern generation strategy. Secondly, the coverage metric is a mean of the evaluation of the test quality. The test quality defines which test is better. The proposed metric could very well serve for the first purpose – to be the test generation guide, but it needs to have little bit relaxed coverage metric for the second purpose. The relaxation is related to the definition of the longest path passing through the fault site. This relaxation is based on the following two reasons. Firstly, the estimation of the longest path passing through the fault site requires additional software, which is not straightforward and which is not always available. Secondly, the test patterns are generated in different ways: deterministic, random, and functional. Some test generation techniques do not target the generation for the longest path, but the quality of test

patterns still has to be evaluated as precise as possible. Therefore we suggest that the notion of "the longest path" would be changed by the notion "the relative longest path" in the coverage metric that is targeted for the evaluation of the test quality. The notion "the relative longest path" means that the longest path is chosen only among the paths that are checked by test patterns. Such a path has to be chosen for every transition fault. Such a chosen path may be far away (much shorter) from the longest structural path passing through the fault site. To show that a length of the path is still important, we suggest calculating the sum (denoted by S_{long}) of the lengths of the relative longest path for every transition fault. The test is better if its sum S_{long} is bigger.

We have mentioned before that the coverage 100% of transition faults does not mean that every transition fault has a path passing through the fault site. If a transition fault is checked but it does not have a path passing through the fault site, it indicates the difficulties of the construction of such a path. It may be the worst case that there is no possibility at all to construct a path passing through the fault site. The fault may be checked only through multi-path. Therefore, such a fault should not be rejected as the fault that does not have a path passing through the fault site. We suggest assigning the length to such a path to one. It is the second relaxation in the metric of test evaluation quality.

When test passes some level of test confidence there always can be additional test patterns that check some transition faults above predefined level. For example, suppose, the test has the first level of test confidence, this means that every transition fault is checked by one path passing through the fault site. But there could be the test patterns that check some transition faults by two or more paths. To take into account the income of these additional test patterns, we propose calculating the sum (denoted by S_{other}) of the lengths of paths checked by these test patterns.

The final estimation of the test quality at some level of test confidence is defined as follows:

$$\text{Quality} = S_{long} + K S_{other},$$

where K is a coefficient (≤ 1).

So far, we have made two relaxations in the metric of test evaluation quality for delay faults. Firstly, the relative longest structural path changes the longest structural path. The relative longest structural path means that the longest path is chosen among the paths that are checked by the test. Secondly, the transition faults that are checked by the test but they do not have the path passing through the fault site are not rejected as such ones. They are considered as having the path of length one.

Finally, the evaluation of the test quality could be defined as a procedure consisting of the following steps:

1. Determine the transition fault coverage.

2. If the test has the transition fault coverage less than 100%, set the level of test confidence to 0 (ground level), and go to the end of the procedure.
3. Determine the level of test confidence.
4. Calculate the final estimation of the test quality at some level of test confidence by the formula:
Quality = $S_{\text{long}} + K S_{\text{other}}$, where
 S_{long} – the sum of the lengths of the longest paths passing through each transition fault and checked by test patterns;
 S_{other} – the sum of the lengths of the paths checked by remained test patterns;
K - coefficient (≤ 1).
5. Stop.

We suppose that such a metric would well suit for the evaluation of the quality of the delay test patterns. The test is better if it has a higher level of test confidence. The test is better at the same level of test confidence if it has a higher estimate of test quality.

6. Conclusions

We have surveyed various delay fault models, their advantages and limitations. We have analyzed the current trends of the test pattern generation for delay faults. It is generally accepted that the delay test patterns should consist of transition fault patterns and path delay patterns where the longest path passes through each site at least once. We have presented the coverage metrics of delay test patterns. They mainly tend to evaluate the quality of path delay fault patterns. We have focused on two groups of the metrics: non-enumerative methods and statistical methods. The non-enumerative methods rely on the traditional counting of the paths, which are tested under robust, non-robust, or functional sensitization criteria. The statistical methods relate their computations to the parameters under process variation. The basis of the methods is their weakness. There is still no accepted single metric to evaluate the quality of the delay test patterns. On that base, we suggested the new approach to the evaluation of the quality of the delay test patterns. This approach stands on the new term – test confidence level (the ground level, the first level, the second level etc.). The transition fault coverage constitutes the ground level of test confidence of delay test patterns. The longest path passes through each site at least once that denotes the first level of test confidence of delay test patterns. The longest path passes through each site at least two times that denotes the second level of test confidence of delay test patterns. The proposed approach allows expressing the quality of the delay test patterns rightly.

References

- [1] R. Garcia. Rethink fault models for submicron-IC test. *Test & Measurement World*, October 2001, 35-38.
- [2] S. Bose, P.Agrawal, V.D.Agrawal. A Rated-Clock Test Method for Path Delay Faults. *IEEE Transactions on Very Large Scale Integration Systems*, No.6(2), 1998, 323-342.
- [3] N. Jha, S. Gupta, Testing of Digital Systems. *Cambridge University Press*, 2003, 1000.
- [4] S. Majumder, M.L. Bushnell, V.D. Agrawal. Path Delay Testing: Variable-Clock Versus Rated-Clock. *Proceedings of the 11th International Conference on VLSI Design*, 1998, 470-475.
- [5] A.K. Majhi, V.D. Agrawal. Tutorial: Delay Fault Models and Coverage. *Proceedings of 11th International Conference on VLSI Design*, 1998, 364-369.
- [6] J.A. Waicukauski, E. Lindbloom, B.K. Rosen, V.S. Iyengar. Transition Fault Simulation. *IEEE Design and Test of Computers*, April 1987, 32-38.
- [7] Y. Shao, I. Pomeranz, S. M. Reddy. On Generating High Quality Tests for Transition Faults. *Proceedings of the 11th Asian Test Symposium (ATS'02)*, 2002, 1-8.
- [8] I. Pomeranz, S.M. Reddy. On n-detection test sets and variable n-detection test sets for transition faults. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol.19, Issue: 3, March 2000, 372 -383.
- [9] J.L. Carter, V.S. Iyengar, B.K. Rosen. Efficient Test Coverage Determination for Delay Faults. *IEEE Int'l Test Conf.*, Washington, DC, Sept. 1987, 418-427.
- [10] V.S. Iyengar, B.K. Rosen, J.A. Waicukauski. On computing the sizes of detected delay faults. *IEEE TCAD*, March 1990, 299-312.
- [11] A.K. Pramanick, S.M. Reddy. On the fault coverage of delay fault detecting tests. *Proc. EDAC*, March 1990, 334-338.
- [12] G.L. Smith. Model for Delay Faults Based Upon Paths. *IEEE Int'l Test Conf.*, Philadelphia, PA, Oct. 1985, 342-349.
- [13] K.-T. Cheng, H.-C. Chen. Delay testing for nonrobust untestable circuits. *Proc. Int. Test Conf.*, 1993, 954-961.
- [14] W.K. Lam, A. Saldanha, R.K. Brayton, A.L. Sangiovanni-Vincentelli. Delay fault coverage, test set size, and performance trade-offs. *IEEE Transactions on Computer-Aided Design*, No.14(1), 1995, 32-44.
- [15] M.A. Gharaybeh, M.L.Bushnell, V.D.Agrawal. Classification and test generation for path-delay faults using stuck-at faults. *Proceedings of International Test Conference*, 1995, 139-148.
- [16] M. Sivaraman, A.J.Strojwas. Primitive path delay fault identification. *Proceedings of International Conference on VLSI Design*, 1997, 95-100.
- [17] S. Padmanaban, M.K. Michael, S. Tragoudas. Exact Path Delay Fault Coverage with Fundamental ZBDD Operations. *IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems*, Vol.22, No.3, March 2003, 305-316.
- [18] K. Heragu, J.H. Patel, V.D. Agrawal. Segment Delay Faults: A New Fault Model. *Proc. 14th IEEE VTS*, April 1996, 32-39.
- [19] A.K. Pramanick, S.M. Reddy. On the Detection of Delay Faults. *Proc. ITC*, September 1988, 845-856.
- [20] U. Mahlstedt. DELTEST: Deterministic Test Generation for Gate Delay Faults. *Proceedings of International Test Conference*, October 1993, 972-980.

- [21] **A.K. Majhi, J. Jacob, L.M. Patnaik, V.D. Agrawal.** On Test Coverage of Path Delay Faults. *Proceedings of 9th International Conference on VLSI Design, January 1996*, 418-421.
- [22] **A. Murakami, S. Kajihara, T. Sasao, I. Pomeranz, S.M. Reddy.** Selection of Potentially Testable Path Delay Faults for Test Generation. *IEEE Int'l Test Conf., Atlantic City, NJ, Oct. 2000*, 376-384.
- [23] **J.J. Liou, L.C. Wang, K.T. Cheng.** On Theoretical and Practical Considerations of Path Selection for Delay Fault Testing. *IEEE/ACM Int'l Conf. on Computer Aided Design, San Jose, CA, Nov. 2002*, 94-100.
- [24] **M. Sharma, J.H. Patel.** Finding a Small Set of Longest Testable Paths that Cover Every Gate. *IEEE Int'l Test Conf., Baltimore, MD, Oct. 2002*, 974-982.
- [25] **P. Gupta, M.S. Hsiao.** High Quality ATPG for Delay Defects. *Proceedings of the IEEE International Test Conference, September 2003*, 584-591.
- [26] **W. Qiu, 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*, 592-601.
- [27] **K. Yang, K.-T. Cheng, L.-C. Wang.** TranGen: A SAT-Based ATPG for Path-Oriented Transition Faults. *Proceedings of the ASP-DAC, 27-30 January 2004*, 92-97.
- [28] **X. Lu, Z. Li, W. Qiu, D.M.H. Walker, W. Shi.** Longest Path Selection for Delay Test Under Process Variation. *Proceedings of the 2004 Conference on Asia South Pacific Design Automation, 2004*, 98-103.
- [29] **P. Gupta, M.S. Hsiao.** ALAPTF: A New Transition Fault Model and the ATPG Algorithm. *Proceedings of the IEEE International Test Conference, October 2004*, 820-829.
- [30] **M.A. Gharaybeh, M.L. Bushnell, V.D. Agrawal.** The Path-Status Graph with Application to Delay Fault Simulation. *IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems, Vol.17, No.4, April 1998*, 324-332.
- [31] **B. Kapoor.** An efficient method for computing exact path delay fault coverage. *Proc. European Design Automation Conf., Mar. 1995*, 516-520.
- [32] **J. Deodhar, S. Tragoudas.** Color Counting and its Application to Path Delay Fault Coverage. *Proceedings of the International Symposium on Quality Electronic Design (ISQED'01), 2001*, 378-383.
- [33] **N.M. Abdulrazzaq, S.K. Gupta.** Path-Delay Fault Simulation for Circuits with Large Numbers of Paths for Very Large Test Sets. *Proceedings of the 21st IEEE VLSI Test Symposium (VTS'03), 2003*, 650-657.
- [34] **M. Sivaraman, A.J. Strojwas.** Path Delay Fault Diagnosis and Coverage – a Metric and an Estimation Technique. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol.20, No. 3, March 2001*, 440-457.
- [35] **J.-J. Liou, L.-C. Wang, A. Krstic, K.-T. Cheng.** Critical Path Selection For Deep Sub-Micron Delay Test and Timing Validation. *IEICE Trans. Fundamentals, Vol.E86-A, No.12, December 2003*, 1-11.
- [36] **Mango, C.-T. Chao, L.-C. Wang, K.-T. Cheng.** Pattern Selection for Testing of Deep Sub-Micron Timing Defects. *Design, Automation and Test in Europe Conference and Exhibition Vol. II (DATE'04), Paris, France, February 16 - 20, 2004*, 2160-2165.
- [37] **W. Qiu, X. Lu, J. Wang, Z. Li, D.M.H. Walker, W. Shi.** A Statistical Fault Coverage Metrics for Realistic Path Delay Faults. *Proceedings of the 22nd IEEE VLSI Test Symposium (VTS'04), 2004*, 37-42.
- [38] **M. Fukunaga, S. Kajihara, S. Takeoka.** On Estimation of Fault Efficiency for Path Delay Faults. *Proceedings of the 12th Asian Test Symposium (ATS'03), 2003*, 64-67.
- [39] **L.-C. Wang, A. Krstic, L. Lee, K.-T. Cheng, M.R. Mercer, T.W. Williams, M.S. Abadir.** Using Logic Models To Predict The Detection Behavior of Statistical Timing Defects. *Proceedings of the IEEE International Test Conference, 2003*, 1041-1050.
- [40] **J.-J. Liou, L.-C. Wang, K.-T. Cheng, J. Dworak, M.R. Mercer, R. Kapur, T.W. Williams.** Analysis of Delay Test Effectiveness with a Multiple-Clock Scheme. *Proceedings of International Test Conference, 2002*, 11, 407- 416.
- [41] **E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas.** Transition Fault Coverage for Different Implementations of the Circuit. *Electronics and electrical engineering, ISSN 1392-1215, Kaunas, Technologija, 2005, No.3(59), 79 – 83.*

DOI: 10.5755/j01.itc.34.4.12024