# Test Generation for Timing-Critical Transition Faults

Xijiang Lin Mark Kassab Janusz Rajski

*Mentor Graphics Corp.*
*8005 SW Boeckman Rd*
*Wilsonvill, OR 97068*
*{xijiang_lin, mark_kassab, janusz_rajski}@mentor.com*

## Abstract

*Timing-aware ATPG [1] has been shown to be an effective method for generating high-quality test sets that detect small delay defects through the longest paths. However, this method usually results in a much higher test pattern count than the traditional transition fault test generation. In this paper, we propose a new criterion that identifies a subset of transition faults to be targeted by the timing-aware ATPG in order to reduce test pattern count while minimizing the impact on the overall delay test quality. The new criterion utilizes the minimal static slack to classify certain transition faults as timing-critical. The test pattern count reduction is achieved by restricting the timing-aware ATPG to targeting the timing-critical transition faults while using traditional transition fault test generation for the remaining transition faults. The experimental results for the industrial circuits show the effectiveness of the proposed method.*

## 1. Introduction

Devices manufactured using deep-submicron technologies are more vulnerable to timing-related defects that change the timing of the circuit instead of its function. Timing-related defects typically originate from process variations and random defects [2]. Process variations are caused by physical parameter variations during the manufacturing process and cannot be eliminated completely. Tightly controlling the process variations and adding extra delay margins during the design phase are two common methods to tolerate the process variations [2][3]. Random delay defects, caused by resistive shorts and resistive opens, etc., introduce additional delays and may cause circuits to malfunction during normal operation. Delay fault testing has become a critical requirement to ensure test quality.

Various fault models have been proposed to target delay defects. The transition fault model [4] considers a gross delay at every gate terminal in the circuit and assumes that the additional delay at the fault site is large enough to cause a logic failure. Due to the limited fault population in the transition fault model, it has been widely used in the industry. However, transition fault test generation ignores the actual delays through the fault activation and propagation paths, and is more likely to detect a fault through a shorter path. As a result, the generated test set may not be capable of detecting small delay defects. The path delay fault model [5] focuses on the testing of a set of predefined structural paths in order to detect the accumulated delays along these paths. Since the number of structural paths in the circuit increases exponentially as the circuit size grows, it is infeasible to consider every path in the circuit explicitly. Instead, a subset of critical paths identified by static timing analysis tools is typically considered during test generation. In [6], a statistical approach is proposed to select the critical paths for testing the path delay faults. To achieve better delay fault coverage, the path selection approach in [7] tries to cover every gate with the longest sensitized path through it. However, this approach suffers from high computational complexity for large designs and it may not always be possible to test every gate based on robust or non-robust conditions. As a compromise between the transition fault model and the path delay fault model, the segment delay fault model was proposed in [8] to robustly test the longest segments in the circuit. In [9], the As-Late-As-Possible Transition Fault (ALAPTF) model was proposed to address the launch condition for the transition fault in order to catch small delay defects accumulated along the transition launch segment. This approach does not consider the fault propagation path from the fault site to the observation point. Hence, the generated test pattern may still detect the faults through shorter paths and the small delay defects may not be detected. Moreover, finding the longest robustly-tested launch segment is a complex task.

To simplify the test generation process for small delay defects, several approaches [1] [10][11] have been proposed to integrate the delay information from standard delay format (SDF) [19] files into the transition fault test generation and/or test application. The True-Time delay test proposed in [10][11] adjusts the test clock frequencies based on the path delay information from SDF in order to

1081-7735/07 $25.00 © 2007 IEEE
DOI 10.1109/ATS.2007.42

achieve the desired launch to capture timing and improve the ability to detect small delay defects. A given test pattern may be applied at a frequency that is higher than the system frequency. This approach may lead to yield loss since a small delay variation may be within the available slack and is therefore redundant during system operation. In addition, when the frequency is increased, paths with longer delays become multi-cycle paths and must be masked. The increase in the number of X states typically reduces the efficiency of on-chip scan compression methods and increases the scan data volume.

In [1], the timing-aware ATPG for the transition faults is proposed. The proposed approach uses the path delay information from SDF to guide the fault effect propagation and the fault site activation in order to detect the target fault through the longest path. Two delay test quality metrics, delay test coverage [1] and statistical delay quality level (SDQL) [12][13], are used to evaluate the quality of the transition fault test set at detecting small delay defects. The experimental results show that the higher the number of transition faults explicitly targeted by the timing-aware test generator, the better the generated test set at detecting small delay defects. However, compared with traditional transition fault test generation, the timing-aware ATPG may generate 10 times more test patterns when targeting every transition fault explicitly. To reduce the test pattern count, a fault dropping criterion, named Dropping based on Slack Margin (DSM), is proposed. In the DSM criterion, a fault $f$ is dropped from the target fault list if a generated test pattern $t_i$ can detect $f$ through a path that satisfies the following condition:

$$\frac{PD_f^s - PD_f^a}{T_{TC} - PD_f^a} < \delta \qquad (1)$$

, where $PD_f^s$ is the longest static path delay through $f$ based on structural analysis; $PD_f^a$ is the longest actual path delay detecting $f$ when applying $t_i$; $T_{TC}$ is the test clock period and $\delta$ is a real number between 0 and 1. When choosing $\delta$ to be 0.5, the experimental results in [1] show that the test pattern count is reduced significantly while sacrificing little test quality. However, the test pattern count is still significantly larger than the test pattern count generated by traditional transition fault test generation.

As the number of test patterns is proportional to the test application cost, it is desirable to reduce the test pattern count further while minimizing the reduction in the effectiveness of the test set at detecting small delay defects. To achieve this goal, we propose a test generation method that restricts timing-aware ATPG to target timing-critical transition faults.

The remainder of the paper is organized as follows. Section 2 gives the motivation of this work. Section 3 describes the criterion for identifying timing-critical transition faults. Section 4 presents the test generation flow for the timing-critical transition faults and a procedure that orders the generated test set based on the detection of the timing-critical transition faults in order to catch the chips with small delay defects as early as possible. Section 5 reports the test generation results for several industrial circuits and Section 6 concludes the paper.

## 2. Motivation

When a delay defect is present in a circuit, the defect's impact on the normal operation of the circuit depends on the defect's location and delay fault size:

- If the delay fault size is greater than the slack of the shortest functional path, $S_{min}$, the circuit will malfunction independent of the delay defect's location along the functional path. Test patterns generated based on the traditional transition fault can ordinarily detect this type of delay defect during at-speed testing.
- If the delay fault size is less than the slack of the longest functional path, $S_{max}$, the circuit will operate correctly even if the delay defects are present in the functional paths.
- When the delay fault size is between $S_{min}$ and $S_{max}$, the circuit will malfunction during normal operation if the sum of the longest path delay passing through the defect and the extra delay introduced by the defect is greater than the system clock period; otherwise, the circuit will still operate correctly. This scenario is shown in Figure 1, where the delay defect does not impact system operation if it is at location 1, but makes the circuit malfunction if it is at location 2.

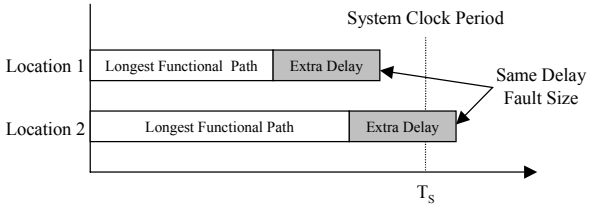


**Figure 1: The impact of the delay defect at different locations on normal operation**

Although it is important to test every functional path thoroughly to avoid delay defects escaping the test, this test strategy is impractical from test cost point of view. Typically, small delay defects have a higher probability of being introduced during manufacturing than large delay defects [17]. The paths with tight slack margins are more vulnerable to violating timing constraints in the presence of the small delay defects. As a result, a trade-off between test cost and test quality is to spend more test resources on testing the delay defects at functional critical paths. Based on this observation, we propose following transition fault testing strategy:

- Identify the transition faults located at the timing-critical paths, i.e., the paths with tight slack margin,

and target them explicitly by using timing information in order to detect the faults through longest paths.
- For the faults not at the timing-critical paths, use traditional transition test generation strategy to detect them.

## 3. Timing Critical Transition Faults

A transition fault is a timing-critical transition fault if it lies along any timing-critical path. All the transition faults along the timing-critical paths are timing-critical transition faults. Unlike path delay testing, a timing-critical transition fault $f$ may be testable even if the timing-critical path including $f$ is untestable.

Since the delay information from SDF is available for the timing-aware ATPG, we use it to identify the timing-critical transition faults based on the condition listed below:

$$\frac{T_{TC} - PD_f^s}{T_{TC}} < \lambda \qquad (2)$$

, where $PD_f^s$ is the longest static path delay through $f$ based on structural analysis; $T_{TC}$ is the test clock period and $\lambda$ is a real number between 0 and 1. Equation (2) means that a fault is in a timing-critical functional path when the minimum slack margin is less than some predefined limit.

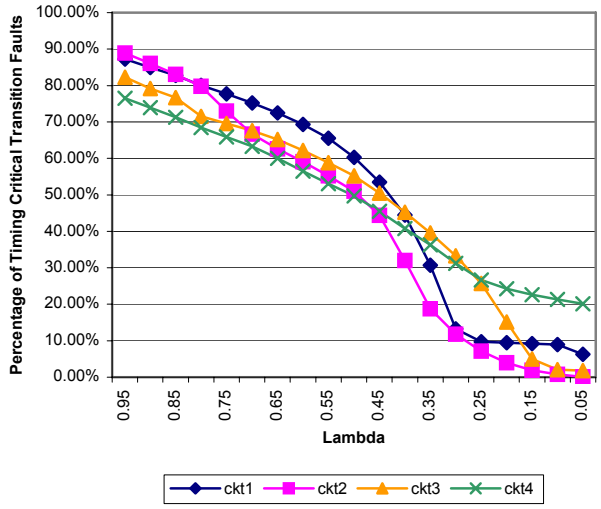


**Figure 2: Percentages of timing-critical transition faults for different λ**

To identify the timing-critical transition fault based on Equation (2), the longest static path delay through each transition fault site is computed. We use the sum of the transition arrival time and the fault effect propagation time as an approximation for the path delay.

At a fault site $f$, the following definitions are used to calculate the delay information associated with $f$ [1]:

- Transition arrival time to $f$, $AT_f$: The time to launch a transition at $f$ from primary inputs or pseudo primary inputs through combinational logic.
- Fault effect propagation time from $f$, $PT_f$: The time to propagate the fault effect at $f$ to a primary output or pseudo primary output through combinational logic.
- Path delay through $f$, $PD_f$: The sum of $AT_f$ and $PT_f$.
- Slack at $f$, $S_f$: The difference between the clock period and $PD_f$.
- Static path delay, $PD_f^s$: The longest structural path delay passing through $f$.
- Actual path delay, $PD_f^a$: The delay is associated with a test pattern $t_i$ that detects $f$ and it is defined as:

$$PD_f^a(t_i) = AT_f(t_i) + \underset{p \in P_s}{MAX}(PT_f^p(t_i)) \qquad (3)$$

, where $P_s$ includes all the sensitization paths starting from $f$. For a given test set $T$, the actual path delay is defined as:

$$PD_f^a = \underset{t_i \in T_D}{MAX}(PD_f^a(t_i)) \qquad (4)$$

, where $T_D$ is the set of test patterns in $T$ that detect $f$. When $T_D$ is empty, $PD_f^a$ is equal to 0.

In Figure 2, we show the percentage of timing-critical transition faults over the total number of transition faults in four industrial circuits by using Equation (2). It can be seen that the number of timing-critical transition faults is reduced as λ decreases. For the same λ, the percentage of timing-critical transition faults is circuit-dependent. For example, when λ=0.25, less than 10% of transition faults are timing-critical in *ckt1* and *ckt2*, but more than 25% are in *ckt3* and *ckt4*. Moreover, in *ckt1* and *ckt4* the number of timing-critical transition faults reduces slowly as λ decreases to 0.

## 4. Test Generation and Test Pattern Ordering

Since the timing-critical transition faults should be detected through the longest paths, the traditional transition test generator needs to be modified in order to maximize the sum of fault propagation delay and fault activation delay. The fault propagation delay is the delay from the fault site to the primary outputs or the pseudo primary outputs and the fault activation delay is the delay from the primary inputs or the pseudo primary inputs to the fault site. In this paper, we apply the timing-aware test generation method proposed in [1] that maximizes the fault propagation delay and fault activation delay, and uses weighted random decisions to cover a wide range of paths in the design.

In order to reduce the test pattern count, traditional transition fault test generation is used to target the non-timing-critical transition faults, and a fault is dropped from fault list once it is detected by the fault simulator.

**Table 1: Characteristics of the circuits**

| Ckt | # Gates (Millions) | # Faults (Millions) | Percentage of Timing-Critical Transition Faults | | | |
|---|---|---|---|---|---|---|
| | | | λ=0.25 | λ=0.2 | λ=0.15 | λ=0.1 |
| ckt1 | 0.24 | 0.6 | 9.69 | 9.45 | 9.23 | 8.94 |
| ckt2 | 0.6 | 1.69 | 7.14 | 3.98 | 1.92 | 0.74 |
| ckt3 | 2.43 | 3.37 | 25.71 | 15.11 | 4.99 | 2.09 |
| ckt4 | 5.06 | 4.55 | 26.67 | 24.24 | 22.64 | 21.33 |

Generating two test sets for timing-critical and non-timing-critical transition faults independently will increase the overall test pattern count. Hence, we propose the test generation flow listed below:

- During test generation, timing information is always used to guide the test generator that detects the target fault through longest path.
- During fault simulation, a non-timing-critical transition faults is dropped from the fault list once it is detected. A timing-critical transition fault is dropped from the fault list only when the test generator has targeted it explicitly or the dropping based on slack margin criterion is met.

Since the majority of non-timing-critical transition faults are detected and dropped during fault simulation and only a small percentage of non-timing-critical transition faults are typically targeted explicitly by the test generator, the test generation flow proposed above should not increase test pattern count for detecting the non-timing-critical faults even if the timing information is used during test generation. This assumption is supported by the experimental results shown in [1].

Test pattern ordering [16] is an effective method to reduce the test application time for identifying the chips with defects and to minimize the fault coverage loss due to test pattern truncation. However, applying test patterns that detect most transition faults first may not be a good strategy to detect small delay defects since the small delay defects are more likely to cause functional paths with tight slack margins to fail. We propose a test pattern ordering procedure listed below in order to apply the test patterns that detect most of the timing-critical transition faults first during test application.

**Procedure** *Order_Test_Patterns*(*T*)

/* *T* is the test set to be ordered. */

1. For each timing-critical transition fault *f*, apply path-based method [1][13] to identify the test pattern in *T* that detects *f* with largest actual path delay.
2. Count the number of timing-critical transition faults detected by each test pattern.
3. Let $T_c$ be the test set including all the test patterns that detect the timing-critical transition faults.
4. Order $T_c$ in decreasing order of the number of timing-critical transition faults detected by each test pattern.
5. Fault simulate all non-timing-critical transition faults by applying $T_c$.
6. Let $T_n$ include all the test patterns not in $T_c$ and fault simulate $T_n$ for all the undetected non-timing-critical faults.
7. Order $T_n$ by decreasing order of the number of transition faults detected by each test pattern and append the ordered test patterns to the end of $T_c$.

## 5. Experimental Results

Four industrial circuits were used to evaluate the test generation results for the timing-critical transition faults. The characteristics of the circuits are shown in Table 1. The distributions of the timing-critical transition faults for each circuit with different λ are shown in Figure 2. The exact percentages of the timing-critical faults for λ=0.25, λ=0.2, λ=0.15, and λ=0.1 are listed in Table 1.

To measure the test quality of the transition fault test set, the delay test coverage (DTC) proposed in [1] is used here. DTC is defined as below:

$$W_f = \frac{PD_f^a}{PD_f^s} \tag{5}$$

$$DTC = \frac{\sum_{f \in F} W_f}{N} \tag{6}$$

, where *F* is the set of transition faults and *N* is number of faults in *F*.

For each circuit, three types of experiments were performed to generate broadside transition fault test sets:

- *Ex*p. 1 – Traditional ATPG: A fault is dropped from the fault list once it is detected.
- *Exp*. 2 – Timing-aware ATPG with DSM δ=0.5 alone: The delay information from SDF is used to guide the test generation. A fault is dropped from the fault list once the actual path delay to detect the transition fault satisfies the dropping based on slack margin criterion with δ=0.5.
- *Exp. 3* – Timing-aware ATPG with DSM δ=0.5 and different λ for identifying the timing-critical transition faults: The test setup is the same as *Exp*. 2. A non-timing-critical fault is dropped from the fault list once it is detected. A detected timing-critical fault is dropped from the target fault list based on the DSM criterion with δ=0.5. λ is set to be 0.25, 0.2, 0.15, and 0.1 in this experiment.

**Table 2: Test generation results for transition faults**

| Ckt | TC (%) | Trad. | | DSM δ=0.5 | | λ=0.25 | | λ=0.2 | | λ=0.15 | | λ=0.1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Pat. | DTC | Pat. | DTC | Pat. | DTC | Pat. | DTC | Pat. | DTC | Pat. | DTC |
| ckt1 | 94.66 | 2847 | 88.02 | 4181 | 89.82 | 3966 | 89.72 | 3918 | 89.71 | 3849 | 89.74 | 3809 | 89.75 |
| ckt2 | 89.96 | 19424 | 72.72 | 30858 | 75.00 | 27479 | 74.61 | 26067 | 74.52 | 23021 | 74.62 | 21282 | 74.13 |
| ckt3 | 90.56 | 3882 | 67.99 | 6073 | 75.45 | 5719 | 75.04 | 5484 | 74.62 | 4885 | 73.66 | 3932 | 73.33 |
| ckt4 | 91.38 | 33048 | 78.36 | 79172 | 81.23 | 75838 | 81.17 | 73986 | 81.08 | 74846 | 81.10 | 72125 | 81.04 |

**Table 3: DTC for the timing-critical transition faults with λ=0.25 by applying different test sets**

| Ckt | Trad. | DSM δ=0.5 | λ=0.25 | λ=0.2 | λ=0.15 | λ=0.1 |
|---|---|---|---|---|---|---|
| ckt1 | 90.42 | 91.48 | 91.37 | 91.47 | 91.41 | 91.37 |
| ckt2 | 60.20 | 65.98 | 65.60 | 64.94 | 63.71 | 62.99 |
| ckt3 | 62.92 | 72.24 | 71.97 | 71.29 | 69.69 | 69.18 |
| ckt4 | 69.19 | 74.05 | 74.00 | 73.93 | 73.86 | 73.77 |

**Table 4: DTC for the timing-critical transition faults with λ=0.2 by applying different test sets**

| Ckt | Trad. | DSM δ=0.5 | λ=0.25 | λ=0.2 | λ=0.15 | λ=0.1 |
|---|---|---|---|---|---|---|
| ckt1 | 90.58 | 91.58 | 91.46 | 91.56 | 91.49 | 91.46 |
| ckt2 | 59.67 | 66.07 | 65.74 | 65.35 | 63.89 | 62.97 |
| ckt3 | 62.83 | 71.35 | 71.18 | 70.79 | 69.09 | 68.43 |
| ckt4 | 68.08 | 73.08 | 73.05 | 73.02 | 72.96 | 72.88 |

**Table 5: DTC for the timing-critical transition faults with λ=0.15 by applying different test sets**

| Ckt | Trad. | DSM δ=0.5 | λ=0.25 | λ=0.2 | λ=0.15 | λ=0.1 |
|---|---|---|---|---|---|---|
| ckt1 | 90.70 | 91.66 | 91.54 | 91.64 | 91.57 | 91.55 |
| ckt2 | 57.06 | 64.55 | 64.08 | 64.03 | 63.13 | 61.88 |
| ckt3 | 62.16 | 66.26 | 66.23 | 66.02 | 65.07 | 64.72 |
| ckt4 | 67.16 | 72.22 | 72.20 | 72.17 | 72.15 | 72.07 |

**Table 6: DTC for the timing-critical transition faults with λ=0.1 by applying different test sets**

| Ckt | Trad. | DSM δ=0.5 | λ=0.25 | λ=0.2 | λ=0.15 | λ=0.1 |
|---|---|---|---|---|---|---|
| ckt1 | 90.87 | 91.82 | 91.70 | 91.80 | 91.73 | 91.71 |
| ckt2 | 57.32 | 64.96 | 64.63 | 64.51 | 63.78 | 63.08 |
| ckt3 | 63.04 | 63.83 | 63.81 | 63.76 | 63.75 | 63.54 |
| ckt4 | 66.39 | 71.54 | 71.52 | 71.50 | 71.47 | 71.42 |

The test generation results are shown in Table 2. After the circuit name, the transition fault test coverage is given under the column *TC*. The test generation results for *Exp*. 1 and *Exp*. 2 are shown under the columns *Trad*. and *DSM δ=0.5*, respectively. Under the columns *λ=0.25*, *λ=0.2*, *λ=0.15*, and *λ=0.1*, we give the test generation results for *Exp*. 3. The test pattern count and delay test coverage for each experiment are shown under the sub-columns *Pat.* and *DTC*, respectively. It can be seen that the timing-aware ATPG with DSM δ=0.5 generates 46.85% to 139.57% more test patterns than the traditional transition fault ATPG. However, the delay test coverage is improved by 1.8% to 7.46%. By using timing-aware ATPG to target timing-critical faults only, the number of generated test patterns is reduced when compared with using DSM δ=0.5 alone. In *ckt2* and *ckt3*, the test pattern count with *λ=0.1* is close to the test pattern count by using traditional ATPG. Although the delay test coverage drops by 0.87% and 2.12% when comparing with DSM δ=0.5, delay test coverage improvements of 1.41% and 5.34% are achieved when compared with traditional ATPG. In *ckt1* and *ckt4*, the test pattern count reduction is not sensitive to *λ* since the percentages of timing-critical transition faults do not reduce much as *λ* decreases. However, the test pattern count with *λ=0.1* is still 8.9% less than the test pattern count with DSM δ=0.5 alone and the delay test coverage drops by only 0.07% and 0.19%, respectively.

From the experimental results shown in Table 2, it can be seen that targeting the timing-critical transition faults by the timing-aware ATPG helps to reduce the test pattern count while having moderate impact on the overall delay test coverage. It is also observed that the test pattern increase is mainly caused by the timing-critical transition faults when applying timing-aware ATPG. This can be seen clearly from the test pattern count comparison between DSM δ=0.5 and λ=0.25.

Since small delay defects are more likely to make timing-critical paths fails and they have a higher probability of being introduced during manufacturing than the large delay defects, next we study the impact of different test sets on detecting the timing-critical transition faults. The study is done by fault grading the same set of timing-critical transition faults by applying the six test sets from Table 2. The delay test coverage for the timing-critical transition faults with λ=0.25, λ=0.2, λ=0.15, and λ=0.1 are shown in Tables 3 to 6. As expected, all the test sets generated by the timing-aware ATPG always achieve higher delay test coverage than that generated by the traditional ATPG. However, except the *ckt1*, the delay test coverage for the timing-critical transition faults is significant lower than the delay fault coverage for all the transition faults even if timing-aware ATPG is used. For example, in *ckt4* the test set generated by using DSM δ=0.5 alone achieves 81.23%

delay test coverage for all the transition faults, but it only achieves 74.05% delay test coverage for the timing-critical transition faults with λ=0.25. The low delay test coverage may be caused by the following reasons:

1. The structural longest paths passing through the timing-critical faults are not sensitizable and the timing-critical transition faults can be tested only through the shorter paths.
2. The DSM criterion drops too many timing-critical faults before targeting them explicitly by ATPG.
3. Timing-aware ATPG maximizes the fault propagation path and fault activation path separately and uses weighted random decision.

To investigate the second reason, we force the timing-aware ATPG to target every timing-critical transition fault. However, the delay test coverage difference is still significant. As reported in the literature [18], the majority of timing-critical paths extracted by timing analysis tools are untestable. We believe the main reason that causes the lower delay test coverage for the timing-critical transition faults is the first reason listed above. To estimate the delay test coverage more accurately, it is better to identify the non-sensitizable paths first and utilize the analysis results to correct the calculation of the static longest path delay through each transition fault. This will be our future research work.

From Tables 3 to 6, it is also noticeable that for the same set of timing-critical transition faults, the test set generated with larger λ achieves better delay fault coverage. This is due to the second and third reasons listed above. However, if every timing-critical transition fault is targeted explicitly, the test pattern count will increase dramatically, especially for the circuits with a large percentage of timing-critical transition faults. Applying the DSM criterion for dropping the timing-critical faults achieves a good tradeoff between the test pattern count increase and the delay test coverage loss.

## 6. Conclusions

Test sets for delay defects have become essential for ensuring test quality of deep-submicron designs. To detect small delay defects, the timing-aware ATPG utilizes delay information from SDF to detect the transition faults through the longest paths. However, it suffers from much higher test pattern count than the traditional transition fault test generation. As the test pattern count is proportional to the test cost, we proposed a method to reduce the test pattern count while minimizing the impact on the ability of the test to detect small delay defects. This is achieved by using timing-aware ATPG to target the timing-critical transition faults while using traditional transition fault ATPG to target the remaining faults.

Since small delay defects have a higher probability to be introduced during manufacturing than larger delay defects, we also proposed a test pattern ordering procedure that gives precedence to test patterns that detect the most timing-critical transition faults. Thus, devices with small delay defects are likely to fail earlier during testing.

## References


[1] X. Lin, K.-H. Tsai, C. Wang, M. Kassab, J. Rajski, T. Kobayashi, R. Klingenberg, Y. Sato, S. Hamada, and T. Aikyo, "Timing-Aware ATPG for High Quality At-speed Testing of Small Delay Defects," *ATS 2006*, pp. 139-146.

[2] S. Mitra, E. Volkerink, E.J. McCluskey, and S. Eichenberger, "Delay Defect Screening Using Process Monitor Structures," *Proc. VTS 2004*, pp. 43-48.

[3] K.S. Kim, S. Mitra, and P.G. Ryan, "Delay Defect Characteristics and Testing Strategies," *IEEE D&T of Computers*, Sep.-Oct., 2003, pp. 16.

[4] J.A. Waicukauski, E. Lindbloom, B.K. Rosen, and V.S. Iyengar, " Transition Fault Simulation," *IEEE Design & Test of Computer*, pp.32-38, Apr. 1987.

[5] G.L. Smith, "Model for Delay Faults Based Upon Paths," *Proc. ITC 1985*, pp.342-349.

[6] J.-J. Liou, L.-C. Wang, A. Krstic, and K.-T. Cheng, "Experience in Critical Path Selection for Deep Submicron Delay Test and Timing Validation," *Proc. ASP-DAC 2003*, 751-756.

[7] M. Sharma and J.H. Patel, "Finding a Small Set of Longest Testable Paths that Cover Every Gate," *Proc. ITC 2002*, pp. 974-982.

[8] K. Heragu, J.H. Patel, and V.D. Agrawal, "Segment Delay Faults: A New fault Model," *Proc. VTS 1996*, pp. 32-39.

[9] P. Gupta and M.S. Hsiao, "ALAPTF: A New Transition Fault Model and the ATPG Algorithm," *Proc. ITC 2004*, pp. 1053-1060.

[10] C. Barnhart, "Delay Testing for Nanometer Chips," *Chip Design*, Aug./Sept. 2004.

[11] A. Uzzaman, M. Tegethoff, B. Li, K. McCauley, S. Hamada, Y. Sato, "Not All Delay Tests Are the Same – SDQL Model Shows True-Time," *Proc. of ATS 2006*, pp. 147-152.

[12] Y. Sato, S. Hamada, T. Maeda, A. Takatori, and S. Kajihara, "Evaluation of the Statistical Delay Quality Model," *Proc. ASP-DAC 2005*, pp.305-310.

[13] Y. Sato, S. Hamada, T. Maeda, A. Takatori, and S. Kajihara, "Invisible Delay Quality – SDQM Model Lights Up What Could Not Be Seen," *Proc. ITC 2005*, pp. 1202-1210.

[14] I. Pomeranz and S.M. Reddy, "On N-detection Test Sets and Variable N-detection Test Sets for Transition Faults," *Proc. VTS 1999*, pp. 173-180.

[15] J.L. Carter, V.S. Iyengar, and B.K. Rosen, "Efficient Test Coverage Determination for Delay Faults," *Proc. ITC 1987*, pp. 418-427.

[16] X. Lin, J. Rajski, I. Pomeranz, and S. M. Reddy, "On Static Test Compaction and Test Pattern ordering for Scan Design," *Proc. of ITC 2001*, pp. 1088-1097.

[17] P. Nigh, and A. Gattiker, "Test Method Evaluation Experiments & Data", *Proc. ITC 2000*, pp. 454-463.

[18] R.S. Fetherston, *et. al*., "Testability Features of AMD-K6 Microprocess," *Proc. ITC 1997*, pp. 406-413.

[19] IEEE DASC Standard Delay Format, http://www.eda.org/sdf.