Delay Fault Model Evaluation Using Tester Response Data

Cheng Xue and R. D. (Shawn) Blanton
Advanced Chip Test Laboratory
Carnegie Mellon University, Pittsburgh PA USA
http://www.ece.cmu.edu/~actl/

Abstract

Most chip producers perform delay testing to detect chips that are affected by defects that adversely affect timing. Several delay fault models have been introduced to guide delay test generation. But similar to static (i.e., slow speed) testing, there is always the question of which fault models are best for ensuring quality. MEasuring Test Effectiveness Regionally (METER) is an approach for evaluating fault model effectiveness. Compared to the conventional test experiment, METER is extremely inexpensive and provides a more thorough evaluation of the quality achievable by a particular fault model. In this work, we describe an extension to METER (called DELAY-METER) that allows the effectiveness of delay fault models to be precisely evaluated. Application of DELAY-METER to the production fail data from an IBM ASIC demonstrates that new and existing delay fault models can be evaluated using conventional tester response data, i.e., data logs collected from production fails through the application of tests generated using conventional fault models.

1 Introduction

Delay testing refers to the process of generating and applying tests for detecting “delay defects”. A delay defect alters the delay of a circuit element (wire, gate, etc.), possibly preventing all logic elements from attaining steady state for one or more specific clock periods. Various delay fault models have been developed to guide the generation of delay-based test vectors. The most widely-used models include the transition delay fault (TDF) model, and the path delay fault (PDF) model. The TDF model targets each gate output for a slow-to-rise and slow-to-fall delay fault while the PDF model targets the cumulative delay along one or more paths [1]. Testing based on the TDF model is affordable and simple. However, it does not guarantee the detection of small delay defects [2]. A small delay defect increases delay slightly and therefore may be only detectable by sensitizing certain timing-critical paths, which can change from chip instance to chip instance due to fabrication variations. Research has shown that tests generated through the use of the PDF model have the potential to capture defective chips that escape tests generated using the TDF model [2]. But testing every path is quite impractical since the number of paths increases exponentially with circuit size. Thus use of the PDF model is typically limited to the critical paths or some subset thereof [3]. Focusing only on critical paths is also not easy, since the increase in delay variation coupled with decreased tolerance to manufacturing variations makes identification of critical paths extremely difficult [4].

To address these challenges, various test metrics have been developed based on the aforementioned fault models. A test metric is not necessarily meant to model a defect but instead specifies how tests should be generated for detecting defects. For example, the N-detect TDF metric requires each TDF to be detected N times by the generated test set. TARO [5], which is an acronym for Transition fault propagated to All the Reachable Outputs, is another test metric based on the TDF model. TARO requires the propagation of slow-to-rise and slow-to-fall transitions through each gate i to every output that can be reached by gate i. K longest paths per gate (KLPG) [6] is a test metric based on the PDF model. For each gate i, the objective of KLPG is to propagate a slow-to-rise and a slow-to-fall transition through the K longest paths that contain gate i. Different from KLPG, the output-deviation metric [7] calculates the output deviations for each test pattern generated by the PDF model, and selects the tests with the highest deviation values.

To ensure quality, chip manufacturers typically combine tests resulting from a variety of models. Understanding the relative effectiveness of each model is key therefore for optimally selecting the best mix. Effectiveness of a model equates to how well tests generated under the guidance of the model can actually detect real defects. Conventionally, model effectiveness is measured in an ad hoc fashion, occasionally investigated through tester experiments involving real chips. A tester experiment typically involves generating a separate set of tests for each model, followed by the application of each test set separately to a population of fabricated chips. The models are then typically evaluated and compared based on the number of failed chips detected by their corresponding tests sets. One example is the small-delay-defect testing experiment described in [8]. Conducting test experiments of this nature incurs extra expense and time since they require new test programs, test generation, and actual test execution. Most importantly, the failed chips detected by the test sets corresponding to a given model may be serendipitous in nature, that is, the actual behavior of the defect may not match the behavior predicted by the model. For example, a test set generated using the TDF model may fortuitously propagate a slowed transition along a critical path, which detects defective chips that better match the PDF model. Such a result incorrectly credits the TDF model with defect detection, especially if the same TDF is detected without failure.

MEasuring Test Effectiveness Regionally (METER) [9] is an approach to evaluate model effectiveness that solely relies on conventional tester response data collected from failed chips.

1 In the remaining part of this paper we will refer to both a fault model and a test metric as simply a ‘model’.
and thus is extremely inexpensive. METER analyzes tester response data to locate possible suspect defect regions within failed chips. Models are then evaluated within these suspect regions by correlating changes in model coverage with defect detection [9]. DELAY-METER, introduced in this work, is an extension of METER that is applicable to several delay models that includes, for example, TDF, N-detect TDF, TARO [5], KLPG [6] and KLPO. (KLPO, K Longest Paths per Output, is a new metric inspired by the output-deviation metric [7] that requires the K longest paths per each output to be tested.) The effectiveness of these models is evaluated using DELAY-METER based on the already-available tester response data collected from an IBM 130nm ASIC. It should be noted however that DELAY-METER is not limited to the models examined here.

The rest of this paper is organized as follows: Section 2 introduces the four steps of DELAY-METER. Section 3 describes how each step is applied in the data-analysis experiment and the corresponding results. Section 4 describes extensions to DELAY-METER, while Section 5 summarizes the overall contributions of this work.

2 DELAY-METER

DELAY-METER consists of four steps:

1) Tester response data pre-processing;
2) Suspect-region identification;
3) Fault selection;
4) Model evaluation.

The primary objective of DELAY-METER is to evaluate the effectiveness of delay models based on whether tests generated by a particular delay model can effectively detect chip failures. DELAY-METER uses conventional tester-response data to evaluate new and existing models. Conventional tester response data includes the data logs collected from production fails through the application of tests generated using conventional fault models. The data logs are assumed to contain failed output-pin information, but simple pass/fail data can also be used. The test environment (i.e., temperature, supply voltage, test-application speed, etc.) assumed for the models evaluated must be compatible with the one used in production test. For example, the data logs from a slow, one-capture test, typically used for stuck-at fault test, cannot be used to evaluate a two-cycle delay fault model.

The first step of DELAY-METER is tester response data pre-processing, which selects chips and test patterns from the data logs that are appropriate for model evaluation. DELAY-METER then identifies possible suspect regions within each failed chip. Faults inside suspect regions are simulated using the selected test patterns and are classified as either effective or ineffective. Finally, the effectiveness of a model is evaluated by correlating tester response data with effective faults. The details of each of these steps are explained next.

2.1 Tester response data pre-processing

Tester response data pre-processing includes failed chip and test pattern selection. The main purpose of this step is to select chips and test patterns suitable for evaluation. Not all test patterns are necessarily used in later steps. For example, if the tester stops collecting data when the limit on output mismatches has been reached or other termination criteria are met, then only the test patterns up to the last failing pattern of each chip should be used for suspect-region identification (step 2), fault selection (step 3), and model evaluation (step 4). To ensure the evaluation is accurate, all tester passing patterns and tester failing patterns (test patterns which a chip passes or fails on the tester, respectively) should be selected, but this greatly increases the time needed for fault selection. For lower accuracy and faster speed, selecting only a subset of the tester passing patterns, and all tester failing patterns is an attractive choice. Depending on the methods used in suspect-region identification and model evaluation, all chips or only a subset can be selected for evaluation. For example, if diagnosis is used in suspect-region identification, then only chips that are “diagnosable” are used.

2.2 Suspect-region identification

A suspect region is (ideally) a small portion of the failed chip that is believed to contain a defect. It can be a gate, a wire, a layout polygon, several gates along a path, all gates in a particular layout region, etc. Several different approaches can be used to identify suspect regions, ranging from all sensitized regions, all regions reported by diagnosis, to regions identified via PFA (physical failure analysis) of a failed chip. A region is “sensitized” if one or more errors created by an activated defect in the region propagate to one or more outputs. PFA of a chip provides the greatest level of precision, but incurs significant cost. The number of chips that undergoes PFA is therefore typically small, and thus cannot lead to a statistically-significant sample. Diagnosis is time efficient and less expensive since it mainly involves gate-level circuit simulation, but the suspect regions reported by diagnosis may not include the failure since diagnosis is not perfect. Using all sensitized regions is much more likely to include the failure but the number of suspect regions can increase significantly, thus reducing evaluation precision accordingly.

Diagnosis is used in this work to indentify suspect regions, and the tester response data is assumed to contain failed output-pin information. Failed output-pin information is not necessary since diagnosis can also be used to identify suspect regions, with less fidelity however, in situations where only limited fail information is available (e.g., the first-failing pattern is only known). In diagnosis, signal lines in the transitive fan-in of the failed outputs are typically fault-simulated and compared with the tester response data. The output generated by diagnosis is a limited set of suspect regions believed to be possible locations of failure. In the experiment described in Section 3, we do the following: For the TDF model and TARO, suspect regions include all the sensitized gates which are in the transitive fan-in of failing outputs. For the PDF models, all non-robustly testable paths that terminate at a failed output are identified as suspect regions. Different path-selection methods are then utilized.
according to the particulars of the test metrics. For KLPG, paths that pass through suspect regions identified using the TDF model are selected. For KLPO, a new test metric inspired by the output-deviation metric [7], paths are selected based on each output.

**Fault selection**

For each failed chip, all the faults within the suspect regions are simulated using the selected test patterns to identify those faults that are detected by the tester failing patterns. Through comparison of the fault simulation result and tester response data, these faults are classified as either “effective faults” or “ineffective faults”. Only effective faults (if any) are selected for gauging the effectiveness (the defect-detection capacity) of the corresponding model.

An effective fault is only detected by tester failing patterns and never by a tester passing pattern. For every tester failing pattern, faults from various fault models are detected, but many of these faults are also detected by tester passing patterns. For example, either the output stuck-at-0 or stuck-at-1 faults is always detected at mismatched outputs, but that does not necessarily mean the stuck-at fault model is very effective in detecting defects, since these stuck-at faults are very likely detected by tester passing patterns as well. A fault detected by one or more tester passing patterns implies the detection of this fault does not guarantee defect detection. By our definition, such a fault does not contribute to the effectiveness of a fault model.

However, for special test metrics that requires multiple detection of the same fault by different test patterns, like N-detect and TARO, a fault is deemed effective if, for any subset of tests that satisfy the metric, there includes at least one tester failing pattern. For an N-detect metric, if one fault is detected by N-1 tester passing patterns, and the Nth time it is detected by a tester failing pattern, it can still be deemed as an effective fault. However if one fault is detected by more than N tester passing patterns, then there exists a subset of test patterns which detect this fault N times, but does not include one tester failing pattern. Such a fault is deemed as ineffective. Similarly, TARO requires a slowed transition caused by a TDF to be propagated to every reachable output. If a fault has four reachable outputs, and a subset of tester passing patterns propagates the slowed transition of this fault to all four outputs, it is not an effective fault. It is worth mentioning that the definition of an effective fault for an N-detect metric and TARO is consistent with traditional fault models. For a traditional fault model, a fault is deemed effective if, for any test pattern that detects this fault, it must be a tester failing pattern.

Table 1 illustrates effective fault selection. The example chip has three tester failing patterns (patterns 1, 3 and 5) that cause errors at either output A or B. Fault simulation reveals that Fault 1 is only detected by pattern 3. Since Fault 1 is never detected by any tester passing patterns, it is deemed as an effective fault. On the other hand, Fault 2 is detected by pattern 2, a tester passing pattern. So for a traditional fault model like the PDF model, Fault 2 is not effective. But for an N-detect metric \(N \geq 2\), since Fault 2 is only detected by one tester passing pattern (pattern 2), any subset of tests that detects Fault 2 \(N \geq 2\) times has to include at least one tester failing pattern. For TARO, Fault 2 has two reachable outputs (output A and B), any subset of tests that propagates the slowed transition of Fault 2 to output A and B has to include pattern 5, which is a tester failing pattern. So Fault 2 is deemed effective for an N-detect metric \(N \geq 2\) and TARO.

**Table 1: Comparing the fault simulation and tester responses for classifying effective and ineffective faults.**

<table>
<thead>
<tr>
<th>Test pattern</th>
<th>Tester</th>
<th>Fault 1</th>
<th>Fault 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fail at A</td>
<td>Pass</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>Pass</td>
<td>Pass</td>
<td>Fail at A</td>
</tr>
<tr>
<td>3</td>
<td>Fail at A</td>
<td>Fail at A</td>
<td>Fail at A</td>
</tr>
<tr>
<td>4</td>
<td>Pass</td>
<td>Pass</td>
<td>Pass</td>
</tr>
<tr>
<td>5</td>
<td>Fail at B</td>
<td>Pass</td>
<td>Fail at B</td>
</tr>
</tbody>
</table>

**Model evaluation**

In a conventional test experiment, the effectiveness of a model equates to the number of chips detected by its corresponding test set. In this work, we define effectiveness of a model with respect to a failed chip. Specifically, effectiveness is equated to the percentage of tester failing patterns of the particular chip that detect one or more effective faults from the model. Tester failing patterns that detect effective faults imply the model guarantees defect detection, while tester failing patterns that do not detect effective faults suggest some defect mechanism cannot be captured by the fault model. The higher the effectiveness, the more failures caused by the defect are consistent with the model. Higher effectiveness also suggests applying extra test patterns using the corresponding model may reduce the escape rate. Conventional effectiveness can still be calculated by counting how many chips have effectiveness greater than zero.

![Figure 1: Comparison between the tester response and the simulation result for (a) one fault and (b) three effective faults.](image-url)

Shown in Figure 1a is a graphical representation of a simulation result of a fault and a tester response. The horizontal rectangle represents test patterns that detect the fault (labeled “simulation fail”). The vertical rectangle represents the tester failing patterns (labeled “tester fail”). Test
As shown in Figure 1b, rectangles labeled 1, 2 and 3 represent evaluation. PDFs A and B have TFSP very effective in detecting this defect. Although effective when either PDF A or B is sensitized, but the PDF model is of tester failing patterns: some failed chip the effective faults. We define effectiveness of a model for represents the tester failing patterns that detect at least one of the actual defect affecting this chip can cause a mismatch where each PDF A or B is sensitized, but the PDF model is very effective in detecting this defect. Although effective PDFs A and B have TFSF≠0, they can be used in model evaluation.

In the ideal case, TFSP=0 and TPSF=0, which would imply the two rectangles of Figure 1a would completely coincide with each other. When TPSF≠0, there is at least one test pattern that detects the fault, but does not fail the chip. By definition, such a fault is not effective, but may be if the model used is N-detect or TARO. When TFSP≠0, there is at least one test pattern that fails the chip, but does not detect the fault. Such a test pattern may detect other effective faults for this failed chip however. For example, if a failed chip has three tester failing patterns, where patterns 1 and 3 detect one effective PDF A, pattern 2 detects another effective PDF B, and both PDFs A and B have TPSF=0, TFSP≠0. It is possible the actual defect affecting this chip can cause a mismatch when either PDF A or B is sensitized, but the PDF model is very effective in detecting this defect. Although effective PDFs A and B have TFSF≠0, they can be used in model evaluation.

As shown in Figure 1b, rectangles labeled 1, 2 and 3 represent three different effective faults (their TPSF all equal to zero), each of which covers a different but not a mutually exclusive set of tester failing patterns. Together the shaded area represents the tester failing patterns that detect at least one of the effective faults. We define effectiveness of a model for some failed chip A as the number of tester failing patterns that detect at least one effective fault divided by the total number of tester failing patterns:

\[
\text{Eff}_{\text{MULTIPLE}}(A) = \frac{\sum_{F_i\in\text{TPSF}(F)}|\text{TPSF}(F_i)|}{|\text{TF}|} \quad (1)
\]

where \(F_i\) is an effective fault within one of the suspect regions of chip A. \(\text{Eff}_{\text{MULTIPLE}}\) is equated to the size of the shaded area divided by the size of the “tester fail” rectangle in Figure 1b.

Other evaluation metrics can also be employed. For example, instead of using multiple faults to evaluate a fault model, the effectiveness can also be calculated using a single fault. Each effective fault \(F_i\) is assigned an effectiveness value from \(|\text{TPSF}|/|\text{TF}|\). The effectiveness for a chip is equal to the highest effectiveness value among all effective faults of that chip:

\[
\text{Eff}_{\text{SINGLE}}(A) = \frac{\max_{F_i\in\text{TPSF}(F)}(|\text{TPSF}(F_i)|)}{|\text{TF}|} \quad (2)
\]

3 Experiment

Table 2 summarizes how the four steps of DELAY-METER are applied to the five delay models investigated. The tester response data used in this work stems from the failure logs of an IBM 130nm ASIC design. The delay test applied to these chips is generated using the TDF model. But these tests are sufficient for gauging the effectiveness of other delay models since it was observed in [6] that tests generated from one fault model can also achieve high coverage for other fault models. The delay test set contains 11,896 delay test patterns and achieves 67.5% TDF coverage. Tester response data includes tester failing patterns and the corresponding failing outputs from 1,837 failing chips. Among the 1,837 chips, 482 chips are not diagnosable, that is, a commercial diagnosis tools fails to report any candidates. 652 chips are diagnosable and have a perfect TDF candidate, i.e., TPSF=TFSP=0 and TF=SF. 703 chips are diagnosable but do not have a perfect TDF candidate. These 703 failing chips likely contain defects whose behaviors are better captured by other models, but are serendipitously detected by the tests generated using the TDF model. These chips are ideal for evaluating what models should be used in “top-off” test generation if transition delay test is assumed as a baseline. So in the tester response data pre-processing step, the 703 chips and all passing and failing patterns are selected for this experiment.

Table 3: Average effectiveness for the models investigated.

<table>
<thead>
<tr>
<th>Effectiveness</th>
<th>TDF</th>
<th>TARO</th>
<th>3-detect TDF</th>
<th>KLPG (K=1)</th>
<th>KLPO (K=1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. EffSINGLE</td>
<td>37.6%</td>
<td>41.5%</td>
<td>53.1%</td>
<td>23.8%</td>
<td>24.7%</td>
</tr>
<tr>
<td>Avg. EffMULTIPLE</td>
<td>51.7%</td>
<td>55.7%</td>
<td>68.4%</td>
<td>33.6%</td>
<td>32.5%</td>
</tr>
</tbody>
</table>

Table 3 summarizes the evaluation for the five models for the 703 chips selected. The first row, labeled “Avg. EffSINGLE,” gives the average effectiveness of each model for all chips using Equation 2. The second row, labeled “Avg. EffMULTIPLE,” gives the average effectiveness of each model for all chips using Equation 1. Higher (average) effectiveness means the
corresponding model has a greater ability to detect defects using the corresponding model.

Figure 2 shows how average effectiveness (Eq. 1) changes with the number of paths using a very liberal interpretation of KLPO and KLPG. The y-axis is the average effectiveness, and the x-axis is the number of paths. For KLPO, the longest paths that are sensitized by the tester-failing patterns that reach each failing output are selected in the suspect-region identification step, and the effectiveness is calculated using the effective PDFs formed from these paths. The same procedure is used for KLPG, except paths that pass through sensitized gates are selected instead. The effectiveness and the number of paths are measured per chip and the value is averaged for all 703 chips. The plots in Figure 2 reveal that KLPO and KLPG effectiveness gradually increases as other (shorter) paths are added to the suspect regions. It also shows that defects are more likely located on the paths selected using KLPO than KLPG.

![Figure 2: Average EffMULTIPLE for KLPG and KLPO as a function of number of paths selected.](image)

Figure 3 shows the percentage of chips that have effectiveness greater than zero as a function of \( N \) for the \( N \)-detect TDF metric (the result is the same for both “SINGLE” (Eq. 1) and “MULTIPLE” (Eq. 2) effectiveness since the value of effectiveness is not relevant). An effectiveness greater than zero indicates the \( N \)-detect TDF metric is guaranteed to detect the failed chip (not fortuitously). The results of Figure 3 help test engineers select an optimal value of \( N \) by weighing the effectiveness against number of tests required to achieve the corresponding value of \( N \)-detect.

Figure 4a is a Venn diagram for the TDF model, TARO and the 3-detect TDF metric. Figure 4b is a Venn diagram for the TDF model, KLPG and KLPO. These Venn diagrams show how many chips are guaranteed to be detected (not fortuitously) by one or multiple models, which is the same as how many chips have effectiveness greater than zero. In Figure 4a, four chips are uniquely detected (not fortuitously) by TARO and not by the 3-detect TDF metric and the TDF model. 74 chips are uniquely detected (not fortuitously) by the 3-detect TDF metric. Fourteen chips are detected (not fortuitously) both by TARO and the 3-detect TDF metric.

![Figure 3: Percentage of the 703 chips that have effectiveness greater than zero for the \( N \)-detect TDF metric.](image)

![Figure 4: Venn diagrams showing the number of chips with effectiveness greater than zero for (a) TDF, 3-detect TDF and TARO, and (b) TDF, KLPG and KLPO.](image)

Figure 4a shows that the 3-detect TDF metric better detects defects than TARO for this fabricated ASIC. Figure 4b shows
that PDF-model-based metrics like KLPG and KLPO detects many chips that the TDF model may not detect. These results are similar to the one obtained from the conventional tester experiment reported in [8]. But unlike conventional tester experiments, the category “OTHER” shown in Figure 4 cannot be reported by a tester experiment. Specifically, there are 75 and 98 chips classified into “OTHER” for Figures 4a and 4b, respectively, which means none of the models examined here can guarantee their detection. In a conventional tester experiment, the serendipitous detection of these chips would be wrongly credited to the models under investigation.

4 Discussion

In the experiment of Section 3, we used DELAY-METER to evaluate several delay models using conventional tester response data. DELAY-METER is not limited however to the models examined here, and is not limited to conventional tester response data. For example, a new ATPG tool that targets small-delay defects was developed in [8]. The objective of the tests generated by this tool are to detect each TDF along paths with minimum slack, or paths with slack smaller than a pre-set limit. In the conventional tester experiment, 12.6% of the failed chips are only detected by the small-delay test generated by this new ATPG tool. DELAY-METER can analyze the small-delay tester response data to check whether the failures of these chips are associated with the detection of a TDF along small-slack paths, or can analyze a conventional tester response data to help decide the slack limit parameter for the new ATPG tool. Another metric, output-deviation [7], calculates the deviation of each node and targets paths whose output has maximum deviation. DELAY-METER can evaluate this metric by calculating the deviation of outputs on tester failing patterns and tester passing patterns, and checking whether tester failing patterns have a larger deviation on the failed outputs.

The sample size (number of chips, number of fails per chip, etc.) required for DELAY-METER to produce reliable results is an interesting topic and therefore is the focus of on-going research. It is important to point out that the number of chips used in a conventional tester experiment also varies from experiment to experiment, and none of them discuss the impact on the confidence of the result. Since the tester response data used by DELAY-METER can be collected from production test, it is easier to obtain more failure data for increasing confidence. Although the confidence of the evaluation result is a function of the number of fails collected per chip, DELAY-METER can use various suspect-region identification methods and does not require a specific lower limit on number of fails per chip.

5 Conclusion

This work introduces DELAY-METER, and demonstrates its ability to evaluate several delay models using conventional production test data from an IBM ASIC. Experiment results show that the 3-detect TDF metric is more effective in detecting defects than TARO for this fabricated chip. For the test metrics based on the PDF model, KLPO uses fewer paths to achieve higher effectiveness than KLPG.

Compared to conventional test experiments, DELAY-METER does not require extra expense and time in developing new test programs, generating new tests, extra test execution, etc. Finally, DELAY-METER extends METER [9] by enabling the evaluation of delay fault models, and by introducing the notions of effective-fault selection and a new multiple-fault evaluation equation.

Acknowledgment

The authors thank the National Science Foundation for sponsoring this research under contract no. 0916828.

Reference