Statistical Defect-Detection Analysis of Test Sets using Readily-Available Tester Data*

Xiaochun Yu and R. D. (Shawn) Blanton
Department of Electrical and Computer Engineering, Carnegie Mellon University
Pittsburgh, PA 15213

Abstract - At substantial cost, conventional methods for evaluating test quality apply a specially-generated test set to a large population of manufactured chips. In contrast, a new time-efficient framework for evaluating test quality (FETQ) that uses tester data from normal production has been developed and validated. FETQ estimates the quality of both static and adaptive test metrics, where the latter guides test using the results of statistical data analysis. FETQ is innovative since instead of evaluating a single measure of effectiveness (e.g., number of unique defects detected), it provides a confidence interval of effectiveness based on the analysis of a collection of test sets. FETQ is demonstrated by measuring the chip-detection capability of several static and adaptive test metrics using tester data from actual ICs.

Keywords- defect escape; fault model; test metrics; test set

I. INTRODUCTION

The main purpose of manufacturing test has been and continues to be the separation of good ICs from ones that do not operate as desired. Various test approaches have been developed to efficiently detect defects and other parametric forms of IC failure. Some are static in nature and are based on abstractions of the actual defect behavior typically referred to as fault models. Common fault models include the single stuck-at, input-pattern [1], bridge [2], transition [3], and path delay [4]. Other test strategies are based on test metrics. For example, the metric used in N-detect test [5] centers on the number of times a fault is detected to guide test generation. Different fault models and metrics continue to be developed to ensure that the defective-chip escape rate remains below a given limit. It is therefore important to evaluate the quality of a test approach based on its ability to detect actual failing ICs.

Conventionally, experiments involving actual ICs are used to evaluate the quality of various test approaches [5-10]. However, conducting these experiments is both time-consuming and expensive. An approach for evaluating the effectiveness of a test metric or fault model called METER (MEasuring Test Effectiveness Regionally) has been described in [11]. In METER, the defective site(s) affecting each failing chip is identified using some form of diagnosis. If the first failing test corresponds to an increase in coverage at a failure site, then the corresponding fault model or test metric1 is deemed effective in detecting the failure. This method is more cost effective and accurate than conducting actual test experiments, but one shortcoming is that it is solely focused on the metrics under consideration and not the actual tests that are created using the test metric. The quality of a test set, regardless of what metric was used to guide its generation, always collaboratively achieves coverage of other, non-targeted metrics [11]. For example, a test set generated using the stuck-at fault model not only achieves a very high coverage of the targeted stuck-at faults, but also detects a significant number of bridge and input-pattern faults [1]. As a result, defects with behavior similar to what is captured by bridge faults are likely to be detected by a stuck-at test set.

This paper describes a new approach for evaluating the overall chip-detection capability of a test set resulting from the use of a metric, which is referred to as FETQ (framework for efficiently evaluating test quality). Like METER, FETQ uses readily-available tester data from failing ICs and thus does not require silicon experiments. FETQ also uses statistical techniques involving bootstrapping to establish confidence intervals for the test-set quality measures that it generates.

The rest of the paper is organized as follows. Section II provides basic background on the use of various metrics for test generation. Section III describes FETQ for evaluating test sets that result from a given metric, while section IV demonstrates FETQ using actual IC fail data. Finally, in section V, the contributions of this work are summarized and research extensions are discussed.

II. TEST METRICS

A test metric is a guide for test pattern generation. For example, single stuck-at fault coverage, which is the ratio of the number of stuck-at faults detected by a test set divided by the total number of stuck-at faults, is a widely adopted test metric. N-detect coverage, which computes the percentage of faults that are detected at least N times, is also an effective test metric [5, 7, 12, 13].

A test metric is either static or adaptive. A static test metric does not depend on changing parameters, so metrics based solely on coverage (stuck-at, bridge, gate exhaustive [6, 14], etc.) are static since they only depend on the (unchanging) circuit and the associated test set.

Different from static test metrics, an adaptive test metric depends on some sort of parameters that change over time. An example of an adaptive test metric is the defect level (DL) model proposed in [15], where tests that lead to the largest reductions in DL are periodically selected from a test pool.

---

1 This work is supported by the National Science Foundation under award no. CCF-0427382 and a Bertucci graduate fellowship.

---

1 In the rest of this paper, the term ‘metric’ refers to the use of a fault model (e.g., stuck-at) or test metric (e.g., N-detect) for guiding test generation.
depending on the current chip quality being achieved. The DL model used is a function of the current set of tests applied, and the probability of occurrence for each defect type (which varies over time and across manufacturing processes, thus requiring DL to be continuously monitored using the test data produced by failing ICs [18]).

Compared to static test metrics, adaptive test metrics depend on statistical analysis of manufacturing data and are expected to outperform their static counterparts because of the additional information utilized. Thus, it is important to have an effective method for evaluating adaptive test metrics.

III. TEST EVALUATION

In this section, FETQ, a time-efficient and cost-effective method for evaluating test quality using readily-available tester data, is described. Evaluating a static metric using FETQ is described in Section III-A. In Section III-B and III-C, FETQ is extended to handle adaptive test metrics.

A. Static Test Metrics

Conventionally, test-metric evaluation starts by generating n test sets \(\{T_1, T_2, \ldots, T_n\}\) corresponding to n different test metrics \(\{m_1, m_2, \ldots, m_n\}\). Since a larger test set is likely to detect more chip failures, each \(T_i\) is constrained to have the same number of tests, where each \(T_i\) has the highest metric value (i.e., coverage) for \(m_i\) as compared to other test sets. Then, each test set is applied to the same set of manufactured chips and the number of chips that fail each test set is recorded. The test set \(T_i\) which detects the most chips is deemed as the most effective.

Test metrics can be also evaluated without the expense of applying several new test sets to a large volume of chips. Let \(T_{\text{orig}}\) be a test set of size \(|T_{\text{orig}}|\) that has already been applied to a set of manufactured chips, a common situation for normal production testing. For a population of chips that have failed \(T_{\text{orig}}\), let \(F_{\text{orig}}\) be the resulting set of failing-IC datalog files, that is, \(F_{\text{orig}}\) is the set of failing ICs along with the associated test responses. \(T_{\text{orig}}\) and \(F_{\text{orig}}\) can be used to evaluate the quality of any test set produced under the guidance of a particular test metric using the flow illustrated in Figure 1. In the evaluation process, \(T_{\text{orig}}\) serves as a test-vector pool for test selection. Specifically, let \(\{T_1, T_2, \ldots, T_n\}\) be n test sets derived from \(T_{\text{orig}}\) using n different test metrics \(\{m_1, m_2, \ldots, m_n\}\). Each \(T_i\) is of fixed size \(s\), where \(s < |T_{\text{orig}}|\). \(F_{\text{orig}}\) serves as a surrogate for a population of manufactured chips to be tested. Specifically, since \(T_{\text{orig}}\) has been applied directly to chips that produced \(F_{\text{orig}}\), a pattern-detection map that correlates each \(t \in T_{\text{orig}}\) to a set of chips that failed \(t\) can be easily extracted. If the defect affecting a failing chip is not sequence dependent, the response of a failing chip to each \(t \in T_{\text{orig}}\) is independent of the test-vector ordering. Under this assumption, instead of actually applying each test set \(T_i\) to the ICs corresponding to \(F_{\text{orig}}\), the total number of ICs detected by each \(T_i\) is derived by simply inspecting the pattern-detection map. Then, similar to the traditional approach, the test set that “detects” the most failing chips is deemed as the most effective.

Obviously a metric \(m_i\) under evaluation can be more effective than \(m_{\text{orig}}\), the one used to generate \(T_{\text{orig}}\). In other words, for the same test set size, a test set that maximizes the coverage for \(m_i\) can detect more failing chips than the one targeting \(m_{\text{orig}}\). If the flow of Figure 1 is used to evaluate \(m_i\), \(T_i\) cannot detect any defective chips that escape \(T_{\text{orig}}\) since the test-vector pool used to generate \(T_i\) is restricted to \(T_{\text{orig}}\). Thus, the effectiveness of \(m_i\) is underestimated. More precisely, the flow of Figure 1 derives a lower bound of the number chip failures detected by a test set of size \(s\) targeting metric \(m_i\). As a result, if the flow of Figure 1 derives that \(T_i\) detects more chip failures than the subset of \(s\) patterns in \(T_{\text{orig}}\) that maximize the coverage for \(m_{\text{orig}}\), \(m_i\) is more effective than \(m_{\text{orig}}\).

B. Adaptive Test Metrics

Unlike static test metrics, adaptive test metrics change the actual tests over time according to what is being observed in the data collected. To evaluate an adaptive test metric using IC-test experiments, first, the parameters for the adaptive test metric are determined from the statistical analysis of the relevant manufacturing data. Then, the adaptive test metric is used to guide test pattern generation. Finally, the resulting test set is applied to future manufactured chips and the number of failing chips detected by the test set is recorded and compared with the number of chips failed by other test sets that are ideally of the same size.

In adaptive test, a new test set is generated based on the fail data resulting from currently-applied tests. A scenario involving only fail data can be mimicked by (i) using the failing-IC datalog files from a subset of \(T_{\text{orig}}\) for determining the parameter values of an adaptive test metric, and (ii) using \(T_{\text{orig}}\) as a test pool for test selection as described Section III-A. Specifically, two sets of failing chips, \(A_t\) and \(B_t\), are randomly chosen from the readily-available failing-IC datalog files. Since the two sets are chosen from the same failing population, chips in the two sets are affected by the same distribution of failure characteristics. \(A_t\) contains the failing chips that are used for

---

An extension for sequence-dependent defects is discussed in Section V.
determining the parameter values of an adaptive test metric \( m_a \), and \( B_i \) is the set of “future fabricated chips” that will be tested using an adaptive test set that stems from the adaptive test metric. Thus, \( A_i \) and \( B_i \) should be generated independently. Moreover, \( A_i \) should only contain chips that are failed by a subset of \( T_{orig} \). The details for constructing \( A_i \) and \( B_i \), and the test evaluation using \( A_i \) and \( B_i \) are described next.

Chips from \( F_{orig} \) that have failed the first \( s \) patterns in \( T_{orig} \) are randomly selected with replacement and placed into \( A_i \) until it has the same number of chips as \( F_{orig} \). \( A_i \) can be deemed as a random set of chips of size \( |F_{orig}| \) that are failed by the first \( s \) patterns in \( T_{orig} \). \( B_i \) is constructed using the same procedure as \( A_i \), but it is created by randomly selecting chips from all the chips in \( T_{orig} \). Because chips are randomly selected from \( F_{orig} \) with replacement, it is possible that a chip in \( F_{orig} \) appears more than once in \( A_i \) and \( B_i \). As a result, \( A_i \) and \( B_i \) both likely differ from \( F_{orig} \). Since \( A_i \) is created independent from \( B_i, A_i \) is very unlikely to be the same as \( B_i \).

To evaluate an adaptive test set with a test-set size constraint of \( s, A_i \) is used to determine the parameter values \( P_i \) for an adaptive test metric \( m_a \). For example, for the adaptive test metric proposed in [15], \( P_i \) includes the probability of each defect type affecting each site, which can be derived from \( A_i \) using the approach described in [18]. Then, with the values of \( P_i \) determined, a test set \( T_i \) of size \( s \) is selected from the original test set \( T_{orig} \) using test metric \( m_a \). The number of failing chips in \( B_i \) detected by \( T_i \) can be deduced using the pattern-detection map. Since both \( A_i \) and \( B_i \) are constructed from the same failing population \( F_{orig} \), their underlying failing characteristics are from the same distribution. On the other hand, the exact failing characteristics of \( A_i \) and \( B_i \) are likely to differ since they are constructed independently. This evaluation process mimics the real application of an adaptive test set, which has the following two characteristics. First, the distribution of the underlying defect types remains unchanged. Second, the failing characteristics affecting the set of failing chips used to customize the test set (i.e., set \( A_i \)) are likely different from the characteristics affecting the future set of defective chips (i.e., set \( B_i \)) that are tested using the adapted test set \( T_i \), since the two sets are two random samples from the same distribution.

When \( A_i \) and \( B_i \) are identical, a test set is adapted to the failing population for which it is going to be used. This scenario is extremely unlikely and this is the reason that \( F_{orig} \) is not directly used for both metric parameter-value determination and test evaluation.

In reality, the distribution of failing mechanisms may slowly change over time, thus making it likely that the distribution of failing mechanisms used to determine the metric parameter values is a mismatch for future failing chips. Hence, assuming that \( A_i \) and \( B_i \) are from the same failing-mechanism distribution, it may deviate from actuality. However, the assumption is reasonable for evaluating the quality of an adaptive test since the premise behind adaptation is to change the test to match the failing mechanisms to improve failing-IC detection. Change or the rate of change of the failing-mechanism distribution should be used to decide whether or not test should be adjusted for future manufactured chips. Identifying changes in the failing-mechanism distribution, however, is a different but related challenge from the one concerning the quality evaluation of an adaptive test.

C. Confidence via Bootstrapping

An adaptive test metric is evaluated using two pools of failing chips, \( A_i \) and \( B_i \), which are created by random sampling from \( F_{orig} \). Different samples of \( A_i \) may result in different parameter values for a given adaptive test metric. Similarly, for different samples of \( B_i \), the number of failing chips detected by the same adaptive test set may vary. Thus, the number of failing chips detected using an adaptive test strategy is a random variable, \( D_i \), which is distributed according to some unknown distribution \( H_D \). Instead of using a random sample from \( H_D \) to evaluate the quality of a particular test set that would result from a given metric, it is better to estimate the expected value of \( D, E(D) \). Moreover, the confidence interval for \( E(D) \), that is, an interval that contains the expected value of \( D \) with a given probability, can also be estimated.

Suppose \( D_1, \ldots, D_n \) is an independent identically distributed (IID) sample from \( H_D \) with a mean of \( E(D) \) and a standard deviation of \( \sigma_D \). The central limit theorem states that the mean of the sample, \( \bar{D} = \frac{1}{n} \sum_{i=1}^{n} D_i \), has a distribution that is approximately normal with mean \( E(D) \) and standard deviation \( \frac{\sigma_D}{\sqrt{n}} \). Thus, \( E(D) \) can be approximated by \( \frac{1}{n} \sum_{i=1}^{n} D_i \) for large values of \( n \). To be more precise, the \((1 - \alpha)\) confidence interval for \( E(D) \) lies between \( \frac{1}{n} \sum_{i=1}^{n} D_i - z_{\alpha/2} \cdot \frac{\sigma_D}{\sqrt{n}} \) and \( \frac{1}{n} \sum_{i=1}^{n} D_i + z_{\alpha/2} \cdot \frac{\sigma_D}{\sqrt{n}} \), where \( z_{\alpha/2} \) is a standard parameter associated with a normal distribution [16]. Thus, the probability of the \((1 - \alpha)\)-confidence interval capturing \( E(D) \) is \((1 - \alpha)\). Computing the confidence interval for \( E(D) \) requires that \( \sigma_D \) be derived first. Since it is not straightforward to determine \( \sigma_D \) analytically, bootstrapping [16] is employed for this purpose.

Bootstrapping [16] is a method for estimating standard deviation. Let \( X_1, \ldots, X_m \) be an IID sample from an unknown distribution \( H_X \). Let \( H_X \) be the empirical distribution of \( X \) from a data set \( X_1, \ldots, X_m \). Random variable \( Y = g(X_1, \ldots, X_m) \), where \( X_1, \ldots, X_m \) is an IID sample of \( H_X \), which may or may not be the same as \( X_1, \ldots, X_m \). To estimate the standard deviation for \( Y \), bootstrapping relies on two ideas: 1) estimate the standard deviation of \( Y \) for distribution \( H_X \) using the standard deviation of \( Y \) for distribution \( H_X \), and 2) approximate the standard deviation of \( Y \) for distribution \( H_X \) using simulation. Specifically, these two ideas can be implemented in four steps: step-i, draw \( X_1', \ldots, X_m' \) from \( H_X' \), which is equivalent to sampling one data at a time randomly from \( \{X_1, \ldots, X_m\} \) with replacement, step-ii, calculate \( Y = g(X_1', \ldots, X_m') \), step-iii, repeat steps (i) and (ii) \( n \) times to produce \( Y_1, \ldots, Y_n \), and step-iv, use the standard deviation of the sample, i.e., \( \sqrt{\frac{1}{n} \sum_{i=1}^{n} (Y_i - \frac{1}{n} \sum_{j=1}^{n} Y_j)^2} \), to approximate the standard deviation of \( Y \).

Using bootstrapping to evaluate the quality of an adaptive test metric is illustrated in Figure 2. The starting point is a population of failing chips, \( F_{orig} \), each of which has failed at least one test pattern in the original test set \( T_{orig} \) of size \( |T_{orig}| \). As described in Section III-B, \( n \) sets of \( A_i \) and \( B_i \) are constructed. For each \( A_i \) and \( B_i \), the number of chips (\( D_i \)) in \( B_i \)
detected by an adapted test set created using the parameter values derived using $A_i$, is deduced using the pattern-detection map. The standard deviation of $P$ is approximated using the formulation $\sqrt{\frac{1}{n} \sum_{i=1}^{n} (D_i - \frac{1}{n} \sum_{j=1}^{n} D_j)^2}$. The expected number of failing chips in $B_i$ that can be detected using the adaptive test strategy, $E(D)$, is approximated by $\frac{1}{n} \sum_{i=1}^{n} D_i$ for large numbers of $n$, and the $(1 - \alpha)$-confidence interval for $E(D)$ lies between $\frac{1}{n} \sum_{i=1}^{n} D_i - z_{\alpha/2} \cdot \sqrt{\frac{1}{n^2} \sum_{i=1}^{n} (D_i - \frac{1}{n} \sum_{j=1}^{n} D_j)^2}$ and $\frac{1}{n} \sum_{i=1}^{n} D_i + z_{\alpha/2} \cdot \sqrt{\frac{1}{n^2} \sum_{i=1}^{n} (D_i - \frac{1}{n} \sum_{j=1}^{n} D_j)^2}$.

The flow of Figure 2 can also be used to compute the confidence interval for the estimated number of chips detected by a test set of size $n$ generated using adaptive test metric $m_n$.

IV. EXPERIMENT

FETQ is used to evaluate several test metrics applied to LSI test chips fabricated with a 0.11 $\mu$m technology. The test chip design consists of 384 ALUs. Although each ALU is identical in logic and layout, to facilitate the ease of testing, the ALUs were partitioned into several groups, each with its own test set. All groups are used in this experiment and their corresponding test-set sizes and numbers of faulty ALUs detected are given in Table 1. All three test sets exceed 100% single stuck-at fault detection, that is, there are 150, 142, and 146 patterns out of the original three test sets that each constitutes a minimum test set that achieves complete 100% SSL fault coverage. The first 197 test patterns are the same for all three test sets. The test metrics evaluated in this experiment include stuck-at, AND-/OR-type bridge faults between lines within physical proximity, input-pattern (IP) faults applied to gates [1], and a diagnosis-assisted adaptive test (DAT) metric that guides test selection to maximize the coverage of the currently occurring defect types on a per design basis [15].

For the three groups, the flow of Figure 1 is used to compare the quality achieved by a minimal stuck-at test set that achieves 100% stuck-at fault coverage with two other test sets of the same size that instead target bridge and input-pattern faults, respectively. In addition, for each group, 100 bootstrap evaluations that follow the flow of Figure 2 is used to compute the expected number of failing chips detected by an adaptive test set of the same size as the corresponding minimal stuck-at test set. The metric values of various test sets are summarized in Table 2. Figure 3 compares the defective chips detected by the three different static test metrics for all three groups. It shows that the bridge and input-pattern test sets detect three more failing chips than the stuck-at test set, which is somewhat expected since it is well known that the stuck-at fault model does not capture many of the defect behaviors observed in modern ICs [10]. The stuck-at test set detects four failing chips that are not detected by either the bridge or the input-pattern

Table 1: The number of chips detected by different test sets.

<table>
<thead>
<tr>
<th>Tester data set</th>
<th>Test set</th>
<th>No. of tests</th>
<th>No. of failing chips</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Stuck-at</td>
<td>100%</td>
<td>94.6%</td>
</tr>
<tr>
<td></td>
<td>Bridge</td>
<td>99.5%</td>
<td>95.1%</td>
</tr>
<tr>
<td></td>
<td>IP</td>
<td>99.9%</td>
<td>94.6%</td>
</tr>
<tr>
<td></td>
<td>DAT</td>
<td>99.9%</td>
<td>94.9%</td>
</tr>
<tr>
<td>2</td>
<td>Stuck-at</td>
<td>100%</td>
<td>94.7%</td>
</tr>
<tr>
<td></td>
<td>Bridge</td>
<td>99.4%</td>
<td>95.1%</td>
</tr>
<tr>
<td></td>
<td>IP</td>
<td>99.8%</td>
<td>94.6%</td>
</tr>
<tr>
<td></td>
<td>DAT</td>
<td>99.9%</td>
<td>94.9%</td>
</tr>
<tr>
<td>3</td>
<td>Stuck-at</td>
<td>100%</td>
<td>94.7%</td>
</tr>
<tr>
<td></td>
<td>Bridge</td>
<td>99.5%</td>
<td>95.4%</td>
</tr>
<tr>
<td></td>
<td>IP</td>
<td>99.9%</td>
<td>94.6%</td>
</tr>
<tr>
<td></td>
<td>DAT</td>
<td>99.9%</td>
<td>95.2%</td>
</tr>
</tbody>
</table>

Table 2: Coverage for stuck-at, bridge, input-pattern, and DAT-based test sets for each tester data set.

<table>
<thead>
<tr>
<th>Tester data set</th>
<th>Test set</th>
<th>Stuck-at coverage</th>
<th>Bridge coverage</th>
<th>IP coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Stuck-at</td>
<td>100%</td>
<td>94.6%</td>
<td>91.8%</td>
</tr>
<tr>
<td></td>
<td>Bridge</td>
<td>99.5%</td>
<td>95.1%</td>
<td>91.5%</td>
</tr>
<tr>
<td></td>
<td>IP</td>
<td>99.9%</td>
<td>94.6%</td>
<td>92.4%</td>
</tr>
<tr>
<td></td>
<td>DAT</td>
<td>99.9%</td>
<td>94.9%</td>
<td>92.1%</td>
</tr>
<tr>
<td>2</td>
<td>Stuck-at</td>
<td>100%</td>
<td>94.7%</td>
<td>91.5%</td>
</tr>
<tr>
<td></td>
<td>Bridge</td>
<td>99.4%</td>
<td>95.1%</td>
<td>91.7%</td>
</tr>
<tr>
<td></td>
<td>IP</td>
<td>99.8%</td>
<td>94.6%</td>
<td>92.4%</td>
</tr>
<tr>
<td></td>
<td>DAT</td>
<td>99.9%</td>
<td>94.9%</td>
<td>92.0%</td>
</tr>
<tr>
<td>3</td>
<td>Stuck-at</td>
<td>100%</td>
<td>94.7%</td>
<td>91.6%</td>
</tr>
<tr>
<td></td>
<td>Bridge</td>
<td>99.5%</td>
<td>95.4%</td>
<td>91.6%</td>
</tr>
<tr>
<td></td>
<td>IP</td>
<td>99.9%</td>
<td>94.6%</td>
<td>92.4%</td>
</tr>
<tr>
<td></td>
<td>DAT</td>
<td>99.9%</td>
<td>95.2%</td>
<td>92.0%</td>
</tr>
</tbody>
</table>
fault test set however. This is likely due to the fact that the stuck-at test set covers all single stuck-at faults while the bridge and input-pattern test sets of the same size do not. The relative effectiveness of stuck-at, bridge, and IP fault metric derived using FETQ is somewhat expected and agrees with previous silicon experiments [6, 9], which shows that FETQ can be used to effectively evaluate test quality.

Figure 4 compares the total number of failing chips detected by test sets created using the four different test metrics, namely, stuck-at, bridge, input-pattern (IP) fault [1], and the adaptive test metric of [15]. Given the same test-set size, the adapted test set detects the most failing chips. Moreover, it uniquely detects one failing chip that is not detected by any of the other test sets.

The quality of test sets of different sizes that are generated using the adaptive test metric of [15] is also investigated and compared with the quality of the original test set. In this experiment, we consider all chips that are detected by the 197 tests common to each group, which results in 2,521 failing chips. In other words, the 197 common tests are deemed as the original test set \( \mathcal{T}_{orig} \). The test-set size constraints for test selection are 20%, 40%, 60%, and 80% of the size of \( \mathcal{T}_{orig} \). For each test-set size constraint, 100 bootstrap runs are performed using the evaluation flow of Figure 2. The resulting 95% confidence intervals for the expected number of chips detected by the adaptive tests are shown in Figure 5 as vertical bars. The resulting confidence intervals are impressively tight. For example, the largest interval occurs for the 20% test set and corresponds to ±2.35 failing chips. Figure 5 also shows the number of failing chips detected by the first 20%, 40%, 60%, and 80% of the original test set. It is shown that the first 80% of the original test set detects 2,512 defective chips, while an adaptive test set of the same size, i.e., an adaptive test set consisting of 158 tests detects 2,520 defective chips, on average. In addition, with a probability of 95%, the expected number of defective chips detected by an adaptive test set with 158 tests lie in the range from 2519.33 to 2520.61. Figure 5 shows that for all the test set sizes considered, an adapted test set detects more defective chips than the original test set with very high confidence. Moreover, for the purposes of test-time reduction, the size of the adapted test set can be significantly reduced with respect to the size of the original test set. For example, an adapted test set consisting of 79 patterns (40% of the original test set) detects 2,508 failing chips on average, while the first 60% of the original test set only detects 2,501 failing chips. This means 7 more failing chips are detected by the adaptive test strategy of [15] using one-third fewer tests. The adaptive test strategy described in [15] therefore achieves better quality than the original test set, which is both reasonable and expected since the adaptive approach considers additional information which in this case, includes the characteristics of failing ICs.
V. SUMMARY AND EXTENSIONS

A new cost-efficient method for evaluating the statistical quality of both static and adaptive test metrics, which we call FETQ, has been described. Conventional methods for evaluating test quality require applying the test set under evaluation directly to a set of manufactured chips. FETQ, on the other hand, uses readily-available fail data to mimic conventional test-evaluation methods. The evaluation outcome is a statistical estimate of the number of failing chips detected by a test set resulting from a given test metric, as well as a confidence interval for the estimation. This is a much more precise evaluation since it provides a more comprehensive measure of how well a given test metric would perform as opposed to the performance of a particular test set as is the case for the traditional silicon experiment. In addition, since FETQ does not involve additional test of actual ICs, the cost associated with tester access is eliminated. Finally, FETQ goes beyond the work in [11] since the overall quality of the test set resulting from each test metric is evaluated and for each test metric, multiple bootstrapped test sets are statistically analyzed. So overall, FETQ is more precise and significantly less expensive to employ.

As discussed in Section III, due to the use of the pattern-detection map for identifying chips detected by a certain test pattern, FETQ cannot handle sequence-dependent defects (e.g., transistor stuck-open). However, FETQ can be extended to incorporate these types of defects. Specifically, diagnosis can be used to identify the activation requirement of a defect affecting each failing chip that is used for test-quality evaluation. If a chip is found to be affected by a sequence-dependent defect, the circuit states of the new test set are examined to determine whether the diagnosis-derived activation condition is satisfied by the test set (instead of using the pattern-detection map). In this way, the evaluation flow of Figures 1 and 2 can be extended to handle chips affected by sequence-dependent defects.

FETQ examined the quality of tests sets based on the stuck-at, bridge, and input-pattern fault models, as well as an adaptive test metric [15]. The evaluation involved actual IC fail data and showed that the stuck-at test metric is effective in identifying failing chips but is not adequate for ensuring the quality levels required by modern ICs which, of course, is well known. Tests derived using a bridge or input-pattern fault model both detect failures that are not detected by stuck-at tests, demonstrating their utility. However, among the four test sets of the same size, the adapted test set detects the most failing chips, and it uniquely fails one defective chip that is not detected by any of the other three. The test quality of the adapted test set is also compared with the original one used in production. The result of this experiment showed that the adapted test sets detect more failing chips with the same number of test patterns, or alternatively, the same number of failing chips can be detected using significantly fewer tests.

Since FETQ uses the existing population of failing chips for evaluation, it suffers from the following limitations. First, FETQ cannot be used to evaluate the effectiveness of a test metric whose targeted failing mechanism does not occur in the existing failing population. However, this scenario is not very likely since many defects are fortuitously detected by test sets targeting other faults (e.g., stuck-at test set). Second, though FETQ is able to compare the effectiveness of several test metrics, it cannot easily deduce the level of escape reduction produced by a new test set.

REFERENCES