JOURNAL OF ELECTRONIC TESTING: Theory and Applications 19, 271–284, 2003


# Modeling Fault Coverage of Random Test Patterns

HAILONG CUI AND SHARAD C. SETH
*University of Nebraska-Lincoln, Lincoln, Nebraska, USA*
hcui@qualcomm.com
seth@cse.unl.edu

SHASHANK K. MEHTA
*Indian Institute of Technology, Kanpur, India*
skmehta@cse.iitk.ac.in

*Received March 30, 2001; Revised September 10, 2002*

Editor: H.-J. Wunderlich

**Abstract.** We present a new probabilistic fault coverage model that is accurate, simple, predictive, and easily integrated with the normal design flow of built-in self-test circuits. The parameters of the model are determined by fitting the fault simulation data obtained on an initial segment of the random test. A cost-based analysis finds the point at which to stop fault simulation, determine the parameters, and estimate fault coverage for longer test lengths. Experimental results on benchmark circuits demonstrate the effectiveness of this approach in making accurate predictions at a low computational cost. As compared to the cost of fault-simulating all the test vectors, the savings in computational time for larger circuits ranged from four to fourteen times. We also present an analysis of the mean and the variance of the fault coverage achieved by a random test of a given length. This analysis and simulation results demonstrate that while the mean coverage is determined by the distribution of the detectabilities of individual faults, the dual distribution of fault coverage of individual test vectors determines the variance.

**Keywords:** probabilistic model, BIST, fault-coverage prediction, cost-benefit analysis of fault simulation, variance of fault coverage

## 1. Introduction

*Motivation.* Assessing the random-pattern testability of a circuit is often necessary in VLSI testing. In random built-in self-test (BIST) a very large number of pseudo-random patterns can be applied to the circuit under test but the designer must ensure that the circuit is random-pattern testable. If it is, the designer must further determine the number of test patterns that would achieve the desired fault coverage. Otherwise, several available corrective actions may be chosen, including embedding deterministic test patterns for hard-to-detect faults in the pseudo-random test sequence [1, 2], inserting test points to improve the circuit's testability [3–5], or altering the scan-testing scheme [6].

One way to measure the testability of a circuit for a given set of test patterns is through fault simulation but this may be prohibitively expensive if the circuit is large or the number of patterns is high. In such situations it is attractive to seek accurate mathematical models for fault coverage estimation. Developing such a model is our main objective in this paper.

The model may be integrated with the random BIST as follows. We perform fault simulation in incremental

steps. After each step, we check if a stopping criterion based on the marginal cost of fault-coverage increase is satisfied, this check being based on our new model for fault coverage estimation. If the stopping criterion is reached before the desired fault coverage, the model is used to estimate the fault coverage achievable with additional number of test patterns. The ISCAS-89 benchmark s9234 (scan version) provides a good illustrative example. For this circuit the stopping criterion is reached after 187,892 test patterns with 90.6% fault coverage. Suppose a maximum of one million random test patterns could be affordably applied to this circuit. Our model predicts 92.6% fault coverage after one million patterns (as compared to 92.1% measured for an actual sample). The predictive model saves 76.6% time over actual fault simulation in this example. Empirically, the gain seems to grow for larger circuits.

In the above example, the estimation of fault coverage would lead to one of two situations: the estimate is below the acceptable fault coverage or otherwise. In the first case, the random-pattern-resistant faults can be identified at the stopping point using statistical fault analysis (STAFAN [7]). We note that it is quite easy to instrument and extend a fault simulator to facilitate such analysis. If test points were inserted to improve the circuit's testability, the proposed approach may be used again to validate that the improvement is sufficient. In the second case, the verification that the fault coverage is indeed sufficient may be carried out by full fault simulation, if feasible. Otherwise, simulating a sample of remaining faults could serve the same purpose, with the sample size chosen to provide the required degree of confidence [8].

*Previous Work.* To be useful, a fault-coverage model should be *accurate*, *simple* (i.e. involve only a small number of parameters), *predictive*, and *fit easily in the normal design flow*. We briefly review and evaluate the fault-coverage models that have appeared in the literature in terms of these desirable properties.

The simplest fault-coverage model is the one-parameter RC-model, due to Williams [9]. It was inspired by the observed similarity of the fault-coverage curve with the time response of a RC circuit. However, as we show later, the model is too simplistic to capture the full range of fault coverage accurately (see Fig. 6).

Goel [10] provides another simple fault-coverage model for projecting the cost of deterministic automatic test pattern generation (ATPG). Based on empirical observations, it represents the fault coverage curve before the *knee* by an exponential and after the knee by a linear function. One problem with the application of this model to random testing is precisely defining when the knee is reached. Another problem is that the linear extrapolation, carried out over a very large number of random vectors, can result in a large error. In our experiments the prediction error exceeded 4% for some examples.

More complex fault-coverage models are based on a circuit's *detection profile* which describes the distribution of random-pattern testability of faults in the circuit [11–13]. In [11], the detection profile is determined by Bayesian estimation and requires deterministic ATPG on a sample of faults. During the ATPG, the method further requires keeping track of if and when a fault is first detected fortuitously, i.e. without being targeted for test generation. These requirements do not fit well with the random pattern test generation for BIST.

In the beta model [12, 13], the detection profile has three parameters but no closed form. The authors propose a method of parameter estimation based on deterministic ATPG and fault simulation without fault dropping. Thus, again the complexity is added to the normal flow of the random BIST. On the other hand, if the parameters are determined by a numerical method, the computational cost can be very high and may indeed exceed the time required for fault simulation. This is borne out by our experiments with the benchmark circuit s15850. Even for the small fitting range of vectors, [1–6800], the curve fitting takes 751 seconds while it takes only 201 seconds to fault-simulate one million vectors.

In addition to the references cited above, there is a large body of work on pseudo-random testing and BIST that is relevant as background to this paper but is not being explicitly cited here. The reader may refer to the pioneering text [14] for fundamental concepts and techniques and to the recent book [15] for an update.

*Paper Outline.* The rest of the paper is organized as follows. In Section 2 we formally describe dual distributions associated with random-pattern testability and their implicit constraints. Then, in Section 3, we analyze the dual distributions to obtain a lower bound and an approximation to the variance of fault coverage. Section 4 is the main section of the paper, where we introduce our fault coverage model and show how it can be characterized and applied to fault-coverage analysis of random tests. Section 5 concludes the paper with a summary of contributions and an outline of future work.

## 2. Fault and Vector Profiles

In this section we define the concepts of detectability of a fault and coverage of a vector and show how both distributions are required to determine the mean and variance of fault coverage. Earlier papers have used the *detection profile* (i.e. the distribution of detectability of faults) to capture the many-to-many relationship between faults and test vectors. However, the detectability profile can accurately model only the mean but not the variance of the cumulative fault coverage. For the latter, we must also know the dual distribution of *coverage* by individual vectors. We coin the terms *fault profile* and *vector profile* to distinguish between the two distributions. We introduce the notation shown in Table 1 to facilitate further discussion.

The *detectability* of a fault $f_i$ is defined as:

$$x_i = \frac{\text{number of vectors detecting } f_i}{\text{Total number of all possible vectors}}$$

In realty it can only assume discrete values but is often assumed to be continuous to simplify analysis. The *fault profile* $p(x)$ represents the density of the detectability distribution. For the dual case, we define *coverage* of a test vector as the fraction of the total number of faults detected by it. Then, the *vector profile* $g(y)$ represents the density of the coverage distribution.

Conceptually, the dual distributions can also be viewed in terms of a bipartite graph (Fig. 1), in which edges connect each fault to the vectors detecting the fault. In the figure, $x_i$ (respectively, $y_j$) represent the detectability (respectively, coverage) of fault $f_i$ (respectively, vector $V_j$). Obviously, the two profiles are not completely independent; the bipartite relationship depicted in Fig. 1 constrains them. In particular, cumulative degrees of the nodes on the left and right hand side must be equal. Let $N$ (respectively, $V$) be the total number of faults (respectively, vectors). Then there are

*Table 1.* Summary of notation.

| | |
|---|---|
| $N$: | Total number of faults |
| $M$: | Total number of detectable faults |
| $n = M/N$: | Fraction of detectable faults |
| $V$: | Total number of all possible vectors; $V = 2^m$ for a combinational circuit with $m$ inputs |
| $F_t$: | Number of faults detected by the first $t$ vectors |
| $\langle F_t \rangle$: | Expected value of $F_t$ |
| $F(t)$: | Normalized value of $\langle F_t \rangle$ |

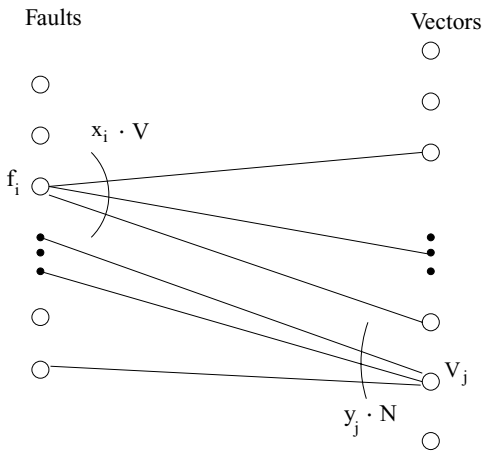


*Fig. 1.* The dual view of fault and vector profiles.

$p(x_i) \cdot N$ number of fault nodes in the graph with degree $x_i \cdot V$. The cumulative degree of all the fault nodes is given by:

$$\sum_i (p(x_i) \cdot N) \cdot (x_i \cdot V) = N \cdot V \sum_i p(x_i) \cdot x_i$$

On the vector side, the cumulative degree of all the vector nodes is:

$$\sum_j (g(y_j) \cdot V) \cdot (y_j \cdot N) = N \cdot V \sum_i g(y_j) \cdot y_j$$

Equating the two and canceling the constants $N$ and $V$ we obtain a constraint between the fault and vector profiles:

$$\sum_i p(x_i) \cdot x_i = \sum_j g(y_j) \cdot y_j$$

The continuous version of this constraint is:

$$\int x p(x)\, dx = \int y g(y)\, dy \tag{1}$$

Thus, the mean of the coverage is fixed once the detectability profile is fixed. However, this doesn't fix the vector profile, and different vector profiles can indeed affect the cumulative fault coverage. As a simple example, consider two cases shown in the bipartite graphs of Fig. 2, where we have three faults $f_1$, $f_2$, $f_3$ and three vectors $v_1$, $v_2$, $v_3$. If a fault $f_i$ is detected by vector $v_j$ there is an edge between them in the graph. From the

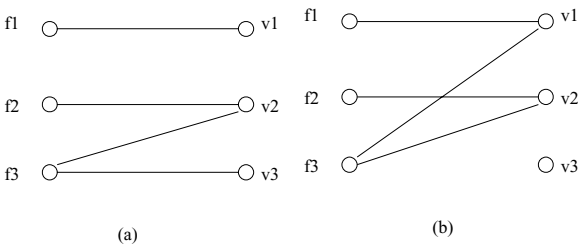


*Fig. 2.* Many-to-many relationship between faults and vectors.

*Table 2.* Fault and vector profiles for the example.

| | Fault profile | | Vector profile | |
|---|---|---|---|---|
| Situation | $x$ | $p(x)$ | $y$ | $g(y)$ |
| (a) | 1/3 | 2/3 | 1/3 | 2/3 |
| | 2/3 | 1/3 | 2/3 | 1/3 |
| (b) | 1/3 | 2/3 | 0 | 1/3 |
| | 2/3 | 1/3 | 2/3 | 2/3 |

figure, the fault and the vector profiles can be easily determined and are shown in Table 2. The fault profiles are seen to be identical in the two cases but the vector profiles are different. It is easily verified that in both cases the expected fault coverage for the first random vector will be 4/9. The variance, on the other hand, is different: 2/81 in the first case and 8/81 in the second case. The difference in the variance suggests that for two circuits with the same fault profile, the fault coverage curves can be quite different if their vector profiles are different. How do the fault and vector profiles look like for real circuits? We have analyzed them for the scan versions of ISCAS-89 benchmarks. Figs. 3 and 4 show the fault and vector profiles for two benchmark circuits. In general, we found that the fault profile is characterized by many spikes and jumps. The vector profile is less chaotic and often resembles a single bell-shaped curve with small variance, similar to the first chart in Fig. 4. The effect of vector profile on the variance of fault coverage in real circuits can be seen in the statistics we collected during 100 independent runs on each circuit. During each run, 20,000 randomly generated patterns were fault simulated on the scan versions of these circuits. In each case, the variance about the mean is shown by a band in Fig. 5. The range of variance reduces to a fraction of a percent in the first case but is about 4.5% in the second case. The narrower error band of circuit s5378 is partially due to its low-variance vector profile.

## 3. Variance Analysis

For a single fault $f_i$ with detectability $x_i$, define the random variable $R_{i,t}$:

$$R_{i,t} = \begin{cases} 0 & \text{if } f_i \text{ is detected during the first } t \text{ vectors} \\ 1 & \text{otherwise} \end{cases}$$

Assume that vectors are randomly generated, then the probability that fault $f_i$ with detectability $x_i$ remains undetected after $t$ vectors is:

$$P(R_{i,t} = 1) = (1 - x_i)^t$$

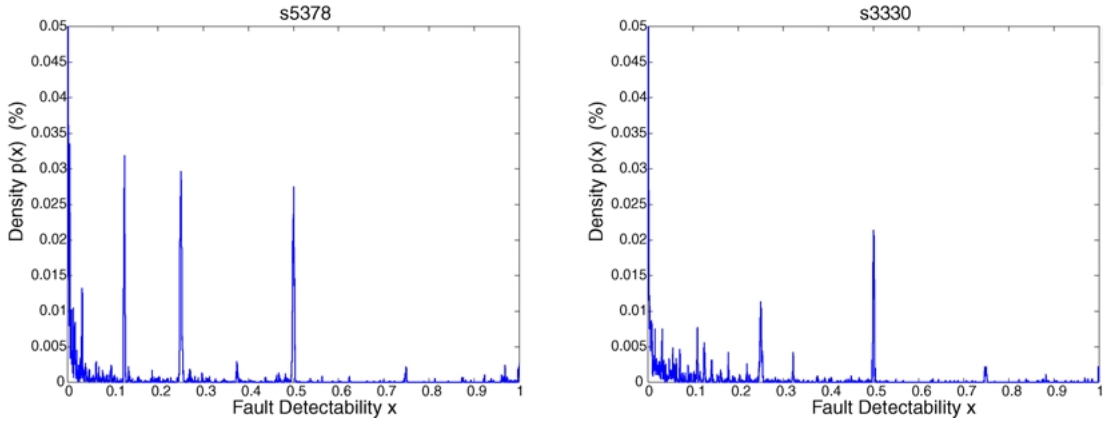


*Fig. 3.* Fault profiles of two benchmark circuits.

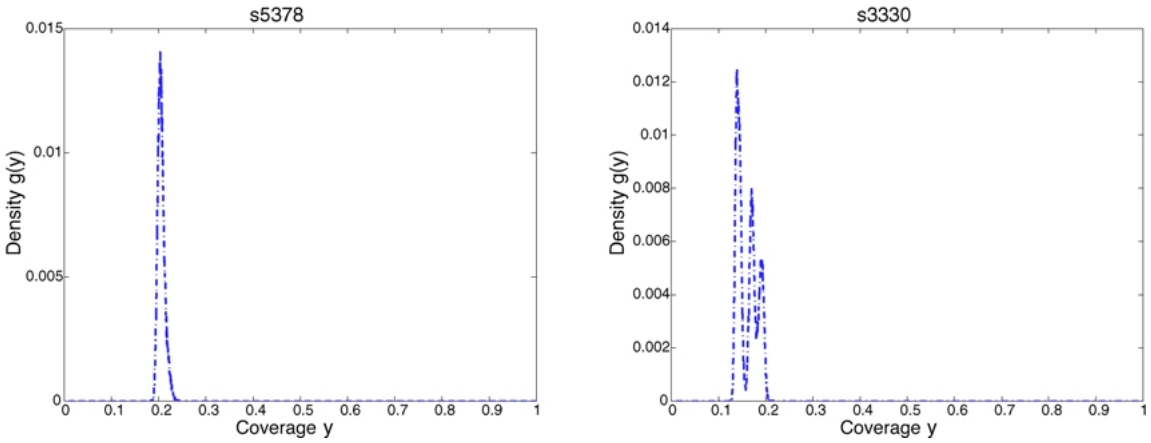


*Fig. 4.* Vector profile of two benchmark circuits.

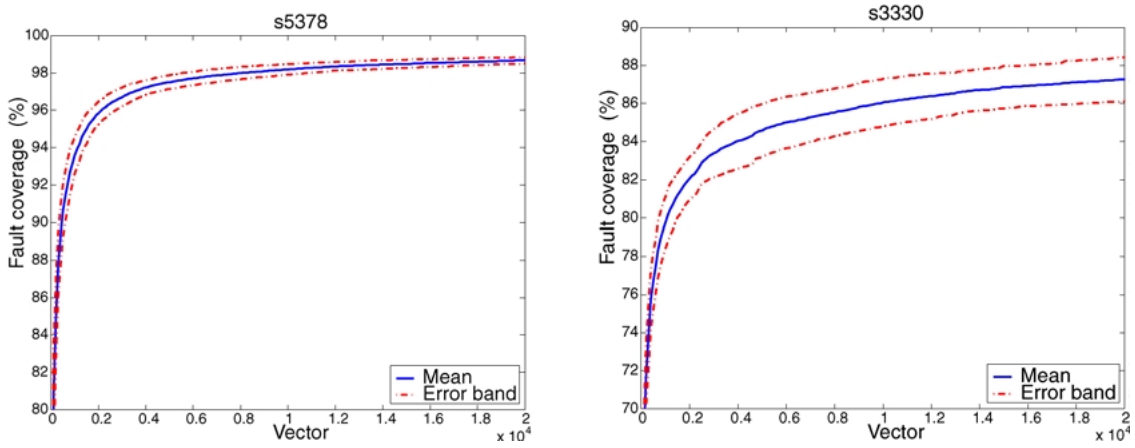


*Fig. 5.* Empirical error bands of benchmark circuits.

hence, the expected value of $R_{i,t}$ and its variance, respectively, are:

$$\begin{aligned} \langle R_{i,t} \rangle &= (1 - x_i)^t \\ Var(R_{i,t}) &= (1 - x_i)^t - (1 - x_i)^{2t} \end{aligned}$$

The number of faults detected by the first $t$ vectors will be:

$$F_t = N - \sum_{i=1}^{N} R_{i,t}$$

with the expected value:

$$\langle F_t \rangle = N - \sum_{i=1}^{N} \langle R_{i,t} \rangle \tag{2}$$

and the variance:

$$\begin{aligned} Var(F_t) &= Var\left( \sum_{i=1}^{N} R_{i,t} \right) \\ &= \sum_{i=1}^{N} Var(R_{i,t}) + \sum_{i} \sum_{j \neq i} cov(R_{i,t} R_{j,t}) \end{aligned} \tag{3}$$

where $Var(R_{i,t})$ is the variance of $R_{i,t}$, and $cov(R_{i,t} R_{j,t})$ is the covariance between $R_{i,t}$ and $R_{j,t}$. The first term accounts for the statistical effect of all faults when their detection is independent, and the second term reflects the correlation in the detection of faults.

An exact quantification of the variance in Eq. (3) appears to be very difficult. However, a lower bound and an approximation for it can be derived, as we show in Section 3.2.

### *3.1. Expected Fault Coverage*

The expected fault coverage has been discussed in [11]. For completeness, we include an abbreviated derivation below.

Substitute $P(R_{i,t} = 1) = (1 - x_i)^t$ in Eq. (2). Then, the expected fault coverage at time $t$ is:

$$\begin{aligned}\langle F_t \rangle &= N - \sum_{i=1}^{N} (1 - x_i)^t \\ &= N - N \int_0^1 p(x)(1 - x)^t dx \\ &= N \left[ 1 - \int_0^1 p(x)(1 - x)^t dx \right]\end{aligned}$$

After normalization, we have

$$F(t) = 1 - \int_0^1 p(x)(1 - x)^t dx \tag{4}$$

Hence, as noted earlier, the expected fault coverage only depends on the fault profile, independent of the vector profile $g(y)$.

### *3.2. A Lower Bound for the Variance of Fault Coverage*

As no good model for $cov(R_{i,t} R_{j,t})$ is known, it is difficult to model the variance of the fault coverage accurately. However, we can derive a lower bound for it (see Appendix B):

$$\begin{aligned}Var(F(t)) \geq \frac{N-1}{N}((1 - \langle y \rangle)^2 + Var(y))^t \\ + \frac{1 - F(t)}{N} - (1 - F(t))^2\end{aligned} \tag{5}$$

where $\langle y \rangle$ and $Var(y)$ are the mean and the variance of coverage.

Following the profile constraint in Eq. (1), $\langle y \rangle$ equals to the mean of the detectability. The variance of the coverage would indeed affect the variance of the fault coverage as shown in the examples in Section 2.

One approximation to the variance of the fault coverage can be found in Appendix B:

$$\begin{aligned}Var(F(t)) \cong \frac{N-1}{N}(\psi_1)^t \left( 1 + \frac{t(t-1)\psi_2}{(\psi_1)^2} \right) \\ + \frac{1 - F(t)}{N} - (1 - F(t))^2\end{aligned} \tag{6}$$

where $\psi_1$ and $\psi_2$ are constants depending on both the vector profile and the fault profile. Unfortunately, the constants do not have closed-form expressions and can only be determined through an expensive process, e.g. curve-fitting or simulation. In Appendix B, we also derive the following approximation to the fault-coverage variance that is based on assuming independence of fault-detection:

$$Var(F(t)) \cong \frac{F(2t) - F(t)}{N} \tag{7}$$

This approximation is easy to compute and, experimentally, is seen to be just as effective as Eq. (6) in fault-coverage estimation (see Table 5).

## 4. Fault Coverage Model for Random Patterns

We have seen that the expected fault coverage depends on the circuit's fault profile. As the examples in Fig. 3 illustrate, however, it is not possible to represent the fault profile accurately with an analytic function. This problem can be dealt with in two ways. First, a Bayesian approach can be used to approximate $p(x)$ by a polynomial in $x$ [11]. In this case, the number of test vectors determines the degree of the approximating polynomial. Second, a flexible analytic function, with an *a priori* fixed and small number of parameters, may be used for the approximation. In the literature the beta function[1] has been shown to be quite effective because of its ability to model different shapes with different values of its two parameters. The following *beta model* was derived in [12].

$$p(x) = (1 - n)\delta(x) + n \frac{1}{B(\alpha, \beta)} x^{\alpha - 1}(1 - x)^{\beta - 1}$$

where $n$ represents the fraction of detectable faults, and $\delta(x)$ is Dirac's delta function. $B(\alpha, \beta)$ is the beta function used as the normalization constant.

When this $p(x)$ is substituted in Eq. (4), we get

$$F(t) = n \cdot \left( 1 - \frac{\Gamma(t + \beta)\Gamma(\alpha + \beta)}{\Gamma(t + \alpha + \beta)\Gamma(\beta)} \right) \tag{8}$$

where $\Gamma(\cdot)$ is the gamma function. This equation is attractive because of its ability to model different shapes with different values of its two parameters. Unfortunately, it does not have a closed form. Two alternatives to get around this problem were discussed in Introduction and both were shown to suffer from undesirable characteristics.

We will now derive a closed form approximation for $F(t)$ by measuring the ratio of incremental fault coverage and the remaining detectable faults. The incremental fault coverage for vector $t+1$ is:

$$\begin{aligned} F(t+1)-F(t) &= \int_0^1 xp(x)(1-x)^t dx \\ &= \int_0^1 x(1-x)^t \left[(1-n)\delta(x) \right. \\ &\quad \left. + n\frac{1}{B(\alpha,\beta)}x^{\alpha-1}(1-x)^{\beta-1}\right] dx \\ &= n\frac{1}{B(\alpha,\beta)}\int_0^1 x^{(\alpha+1)-1}(1-x)^{\beta+t-1}dx \\ &= n\frac{B(\alpha+1,\beta+t)}{B(\alpha,\beta)} \end{aligned}$$

Further, Eq. (8) can be rewritten as:

$$\begin{aligned} n-F(t) &= n\frac{\Gamma(t+\beta)\Gamma(\alpha+\beta)}{\Gamma(t+\alpha+\beta)\Gamma(\beta)} \\ &= n\frac{\Gamma(t+\beta)\Gamma(\alpha)/\Gamma(t+\alpha+\beta)}{\Gamma(\beta)\Gamma(\alpha)/\Gamma(\alpha+\beta)} \\ &= n\frac{B(\alpha,\beta+t)}{B(\alpha,\beta)} \end{aligned}$$

hence, the ratio

$$\begin{aligned} \frac{F(t+1)-F(t)}{n-F(t)} &= \frac{B(\alpha+1,\beta+t)}{B(\alpha,\beta+t)} \\ &= \frac{\alpha}{t+\alpha+\beta} \end{aligned}$$

Next, we approximate the derivative of $F(t)$ by its first order difference, i.e. $dF(t) \cong F(t+1)-F(t)$. The continuous version of this equation is:

$$\frac{dF(t)}{dt} = \frac{\alpha}{t+\alpha+\beta}(n-F(t))$$

With the initial condition $F(0)=0$, the solution to the above equation is:

$$\begin{aligned} F(t) &= n\left(1-\left[\frac{\alpha+\beta}{t+\alpha+\beta}\right]^{\alpha}\right) \\ &= n\left(1-\frac{1}{(At+1)^{\alpha}}\right) \end{aligned} \tag{9}$$

where $A=1/(\alpha+\beta)$. Equation (9) provides a three-parameter power model for the fault coverage at time $t$.

### 4.1. *Characterization of the Fault Coverage Model*

Once we have the fault coverage model and the variance approximation, the characterization is done by weighted nonlinear least square (NLS) curve fitting ([16], p. 27).

Let $Y(t_i)$ be the actual fault coverage at time $t_i$, $i=1\ldots k$, where $k$ is the number of sampling points. Also, let $F(t;\theta)$ be the model where $\theta$ is the parameter set. Then because of the random error, we have the following equation:

$$Y(t_i) = F(t_i;\theta) + e(t_i)$$

where $e(t_i)$ is the discrepancy between the data and the model at time $t_i$.

Then the objective of the weighted NLS curve fitting is to find the parameter set $\theta'$ such that

$$\sum_i^k \frac{(Y(t_i)-F(t_i;\theta'))^2}{\sigma^2(e(t_i))}$$

is minimized, i.e., the variance of the error $\sigma^2(e(t_i))$ is used as a weighting function. Here $F(t_i;\theta)$ is given by Eq. (9) and $\sigma^2(e(t_i))$ is given by Eq. (7).

### 4.2. *Comparison with Other Models*

The original RC model [9] does not consider redundant faults but can be easily extended as follows to account for them:

$$F(t) = n(1-t^{-\alpha})$$

This equation can be considered to be a special case of our model (Eq. (9)) with $t$ substituting $At+1$. The simplification reduces the accuracy of the RC model, as confirmed by the comparison in Fig. 6. Note that the best fit for the RC model consistently over-estimates around the knee and underestimates beyond the knee. The error comparison for several ISCAS-89 benchmark circuits will be presented in Section 4.5.

The non-parametric detection profile model [11] is not amenable to curve fitting hence it cannot be compared directly with our model. We have already pointed out the limitations of this Bayesian method to random tests in Introduction.

The beta model [12, 13] is parametric and was developed for deterministic test generation. Like the

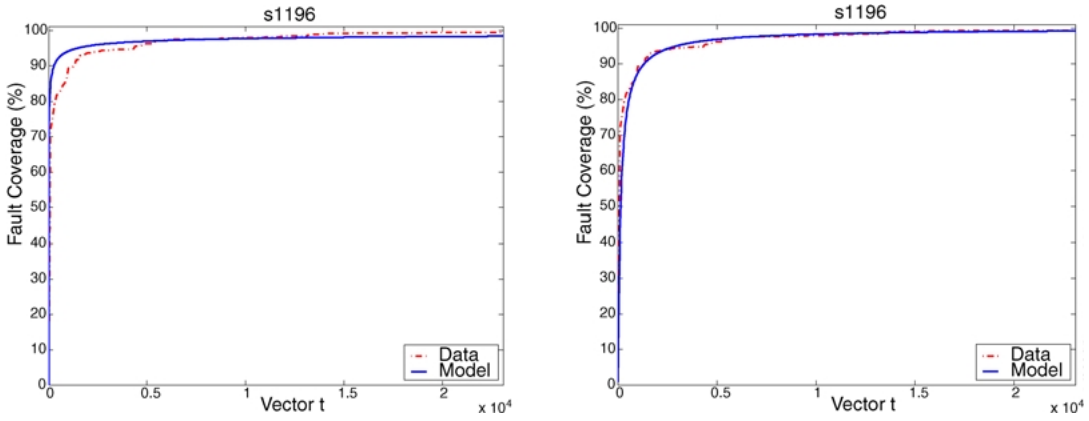


*Fig. 6.* Curve fitting using the full range of data: Williams' model (left) vs. our model (right).

Bayesian method, it is not well suited to random test generation. The reasons are as follows:

(a) If the parameters are determined by fault sampling (as recommended by the authors) then deterministic ATPG and full fault simulation (without fault dropping) are required. These steps are not easy to integrate in the random-BIST design flow.
(b) If the parameters are determined by curve fitting, the beta model requires $O(t)$ time to calculate the fault coverage at time $t$ vs. $O(1)$ time required by our model. The overall CPU-time comparison is shown in Table 3. Notably, not only is the CPU time for our model two or three orders of magnitude smaller but also it is almost independent of the fitting range. For the beta model, the CPU time rises nearly as the square of the fitting range. Even for a small fitting range 1–6800, it consumes 750.49 CPU seconds on a SGI Origin 2000 machine. Considering that it takes just over 200 seconds on the same machine to fault-simulate one million random patterns for this circuit, it is clear that the beta model is not practical in this case.

### *4.3. Cost-Benefit Analysis of Random ATPG*

The computing cost of random ATPG can be divided into two parts: the cost of random pattern generation and the cost of fault simulation to grade the test vectors. The first part requires little CPU time and hence can be ignored. The cost of fault simulation may be measured in terms of the number of logic gates evaluated during simulation. This number may vary with the fault simulation algorithm used. However, if a detailed analysis is not required a simplifying assumption may be used that every fault incurs a unit cost to simulate. The same assumption is implicit in the analysis presented by Goel [10].

*Table 3.* CPU time comparison for the circuit s15850.

| Fitting range | 1–3264 | 1–6800 | 1–13600 | 1–27200 |
|---|---|---|---|---|
| The new model | 0.63s | 1.24s | 1.279s | 2.149s |
| The beta model | 188.93s | 750.49s | 2658.09s | 6022.13s |

As only the undetected faults are simulated each time, the total cost up to time $t$, $C(t)$, will be proportional to the sum of remaining faults from 1 to $t$. Expressing this in the continuous form, we have:

$$C(t) = \int_0^t (1 - F(x))\,dx$$

So the rate of cost increase at time $t$ will be

$$dC(t)/dt = 1 - F(t)$$

On the other hand, the rate of fault-coverage increase is $dF(t)/dt$. Now, consider the ratio of these two rates:

$$R_t = \frac{dF(t)}{dC(t)} = \frac{F'(t)}{1 - F(t)}$$

Obviously, as $t \to \infty$,

$$F'(t) \to 0$$

and

$$1 - F(t) \to 1 - n, \quad 1 - n \geq 0$$

If the circuit has many redundant faults, $1 - n$ will be significant, otherwise, $1 - n$ is close or equal to 0.

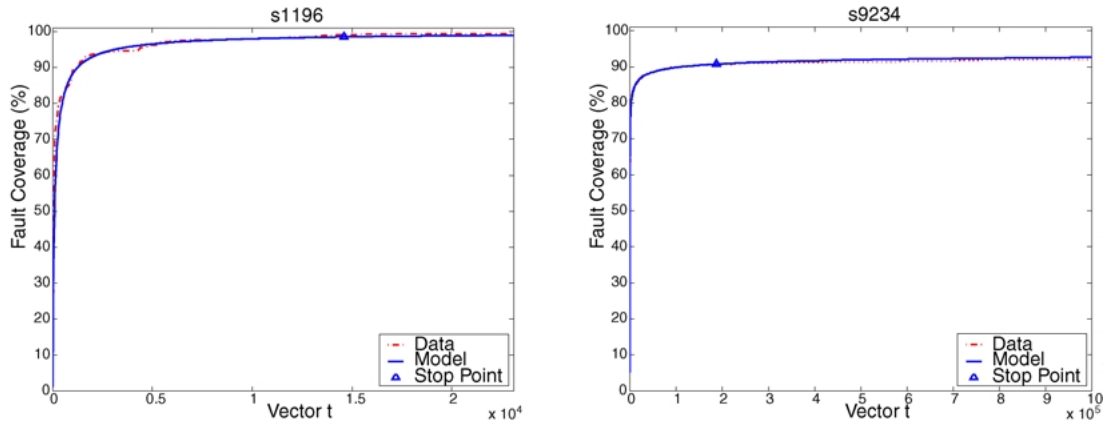


*Fig. 7.* Fault simulation stops when a criterion, based on cost-benefit analysis, is satisfied.

Therefore, $R_t$ will go to zero relatively quickly for a random-test-resistant circuit vs. a random-testable circuit. This property can be exploited for an early identification of a random-test-resistant circuit. We select a threshold value $\theta$ such that when $R_t \leq \theta$ the test generation process terminates. This stopping criterion effectively distinguishes circuits with a substantial number of redundant or random-pattern-resistant faults. In Fig. 7, circuit s1196 is more testable, and the test generation process continues until it reaches a very high coverage just before $t = 15{,}000$ (indicated by the short triangle). For the circuit $s9234$, the process stops before $t = 200{,}000$ where the threshold value of $10^{-6}$ is reached.

### 4.4. *Stopping Criterion*

Once characterized, the model can be used to predict the fault coverage. An important issue in model characterization is deciding how much fault-simulation data is enough for this purpose. Intuitively, fault simulation should be continued beyond the "knee" of the fault-coverage curve. However, this intuitively appealing notion is highly dependent on the scale of data presentation: as the coverage data is zoomed in, many more knees may be revealed in what earlier appeared to be a straight line. What then should be the scale used to determine the knee? We bypass this vexing problem by using a stopping criterion based on the benefit/cost ratio of the last section with a lower bound on the desired fault coverage. Specifically, we stop fault simulation and determine the final values of the model parameters if the cumulative fault coverage reaches a lower bound $\lambda$ *or* the benefit/cost ratio $R_t$ falls below a threshold $\theta$. $\lambda$ is set according to the required fault coverage while $\theta$ is set experimentally to balance the cost vs. the accuracy of fault-coverage prediction.

Based on this stopping criterion, a flow chart of the prediction process is shown in Fig. 8. It starts with a

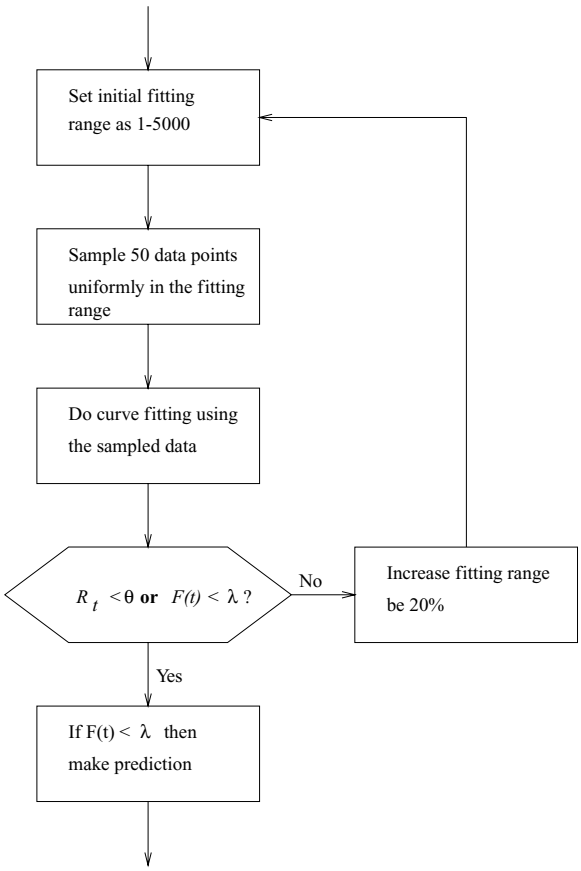


*Fig. 8.* The prediction process.

small fitting range (5000 random patterns) for fault simulation and samples 50 points uniformly in this range for curve fitting. The stopping criterion is next used to decide whether to stop or continue the process of fault simulation, data-point sampling, and curve fitting for another iteration. In the latter case, the fitting range is increased by 20% over the last value. When the stopping criterion is met, if the desired fault coverage is reached, there is no need for prediction. Otherwise, the model is used to decide whether or not the desired fault coverage will be achieved within the allowable number of random patterns.

### *4.5. Experimental Results*

In order to demonstrate the predictive power of the model for random ATPG, we carried out the process described above on the scan versions of the ISCAS89 benchmark circuits. The lower bound $\lambda$ on the fault coverage is assumed to be 99.5%. For this value of $\lambda$, $\theta = 10^{-6}$ was found experimentally to give better than 1% accuracy in fault-coverage estimation. We assume one million to be the maximum affordable number of test vectors.

The results are shown in Table 4. The columns are arranged left to right as follows. After circuit name, the next two columns give the vector number and fault coverage at the stopping point. This is followed by the column named "Total vectors" that has the value of either one million vectors (if the desired fault coverage of 99.5% is not achieved by this time) or a smaller number of vectors that achieve the desired fault coverage. The next two columns give, respectively, the actual fault coverage achieved at the vector number given in "Total vectors" and the fault coverage predicted after curve-fitting. This is followed by the column indicating the error in prediction. The next three columns give the CPU times, respectively, for curve fitting, fault simulation to the stopping point, and the *additional* fault simulation for the total number of vectors given by "Total vectors". These times were recorded for the SGI Origin 2000. The last column shows the percentage of additional fault simulation time required over the fault simulation time to the stopping point.

The upper part of Table 4 is for the large circuits in the benchmark set. It can be seen that for all of these circuits, this method provides an effective way of identifying the stopping point and affording significant savings in time as seen in the last column. The error in prediction never exceeds 0.5% in magnitude.

The other six circuits are too small to demonstrate the full benefit of the proposed technique. The initial fitting range for the last two circuits was set to [1–500]. Here, the curve fitting time dominates the fault simulation time in three cases. It may be noted, however, that these times were obtained on Matlab in which curve fitting programs are interpreted. If the programs were written in C/C++, the curve fitting time could be further reduced.

Table 5 compares the resulting errors when the curve fitting required in the procedure described in the last section is carried in three different ways: by our weighted NLS method using Eq. (7) for variance (the

*Table 4.* Experimental results on the ISCAS89 benchmark circuits (scan version).

| | | | | % FC at total | | | Times (secs) | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Circuit | Stop vector | Stop % FC | Total vectors | Actual | Predicted | Error | CF | $FC_{st}$ | $FC_{add}$ | % Saving in time |
| s9234 | 187892 | 90.6 | $10^6$ | 92.1 | 92.6 | 0.5 | 3.2 | 20.8 | 81.9 | 394 |
| s15850 | 130480 | 94.1 | $10^6$ | 95.9 | 95.3 | −0.5 | 3.4 | 24.0 | 177.2 | 738 |
| s3330 | 187892 | 91.4 | $10^6$ | 93.6 | 93.6 | 0.0 | 4.0 | 6.0 | 30.0 | 498 |
| s13207 | 324678 | 98.4 | $10^6$ | 98.5 | 98.6 | 0.1 | 3.9 | 21.1 | 86.7 | 412 |
| s38584 | 75509 | 95.6 | $10^6$ | 95.7 | 95.9 | 0.2 | 2.8 | 43.6 | 630.1 | 1444 |
| s38417 | 561044 | 98.9 | $10^6$ | 99.2 | 99.1 | −0.1 | 3.7 | 127.6 | 598.3 | 469 |
| s1196 | 14634 | 99.1 | 23100 | 99.5 | 99.0 | −0.5 | 4.5 | 0.1 | 0.2 | 214 |
| s1238 | 75509 | 94.9 | $10^6$ | 94.9 | 95.4 | 0.5 | 3.0 | 0.8 | 9.0 | 1171 |
| s1423 | 36414 | 99.0 | $10^6$ | 99.1 | 99.1 | 0.0 | 2.5 | 0.3 | 8.3 | 2624 |
| s5378 | 52437 | 99.0 | $10^6$ | 99.1 | 99.5 | 0.4 | 2.6 | 1.8 | 30.1 | 1686 |
| s1488 | 1868 | 99.3 | 2200 | 99.5 | 99.2 | −0.3 | 1.7 | 0.03 | 0.03 | 82 |
| s1494 | 3228 | 99.1 | $10^6$ | 99.2 | 100 | 0.8 | 6.8 | 0.1 | 8.40 | 168 |

*Table 5.* Comparison of errors in fault-coverage estimation.

| | % Error using | | |
|---|---|---|---|
| Circuit | Eq. (7) | Eq. (6) | William's |
| s9234 | 0.5 | 0.1 | 0.9 |
| s15850 | −0.5 | −0.9 | −0.8 |
| s3330 | 0.0 | −1.2 | −0.3 |
| s13207 | 0.1 | 0.0 | 0.5 |
| s38584 | 0.2 | 0.2 | 0.7 |
| s38417 | −0.1 | 0.0 | −0.5 |
| s1196 | −0.5 | −0.6 | −1.8 |
| s1238 | 0.5 | 0.5 | 1.9 |
| s1423 | 0.0 | 0.0 | 0.3 |
| s5378 | 0.4 | 0.4 | 0.6 |
| s1488 | −0.3 | −0.2 | −2.4 |
| s1494 | 0.8 | 0.8 | 0.7 |

method used to report the results in Table 4), by our weighted NLS method again but using Eq. (6) for variance, and by the William's RC model. The errors are seen to be comparable for the two weighted NLS cases while they are consistently higher for the William's model (except for s1494 where the latter is slightly lower).

The comparison of our model against the beta models is omitted here because, computationally, it was not feasible to fit the hundreds of thousand of vector data in the beta models (see Section 4.2). The curve fitting time for both the RC model and the new model, in contrast, is consistently within a few seconds.

## 5. Conclusion

We presented a new fault coverage model along with a suggested method of its use in BIST applications. The model is easily characterized by just the fault-simulation data and experimental results attest to its efficiency and accuracy.

Although the projected fault coverage agrees well with the actual data in almost all cases, inaccuracies in predictions can arise for two reasons:

(a) As the variance of fault coverage is an approximation, when the approximation is not good, the weighted nonlinear least square fitting may deviate from the true value and lead to inaccurate predictions.

(b) The fault coverage curve is a time series. There is strong dependence between successive data points, which violates the assumptions of nonlinear least square fitting [16]. This may contribute to error in parameter estimation. We have used a first-order autoregressive error process (AR(1), [16]) to reduce the dependency.

Our preliminary investigations show that the fault coverage model can be extended to deterministic ATPG based on the assumption that in targeted test generation, the non-targeted faults are randomly detected [11]. The variance of the fault-coverage estimate can be shown to be determined by the fault coverage due to the random part.

Our current focus is on developing a more precise understanding of the effect of vector profile on the variance of fault coverage. We would also like to evaluate the usefulness of the model on larger industrial examples.

## Appendix A: Definitions of Some Mathematical Functions

The *gamma function* is:

$$\Gamma(\alpha) = \int_0^\infty e^{-x} x^{\alpha-1} dx, \quad \alpha > 0$$

It has the following property:

$$\Gamma(\alpha + 1) = \alpha\Gamma(\alpha)$$

The *beta function* is:

$$B(\alpha, \beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1} dt, \quad \alpha > 0, \beta > 0$$

In terms of gamma function, beta function can also be expressed as following:

$$B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)}$$

The *Dirac's delta function* is defined by the following two equations:

$$\delta(x-a) = 0, \quad \text{for } x \neq a, \quad \text{and} \quad \int_{-\infty}^{\infty} \delta(x-a)\, dx = 1$$

## Appendix B: A Lower Bound on the Variance of the Fault Coverage

From Eq. (3),

$$Var(F_t) = \sum_{i=1}^{N} Var(R_{i,t}) + \sum_i \sum_{j \neq i} cov(R_{i,t} R_{j,t})$$

Recall that $cov(X, Y) = \langle XY \rangle - \langle X \rangle \langle Y \rangle$, therefore,

$$\begin{aligned} Var(F_t) = &\sum_{i=1}^{N} Var(R_{i,t}) + \sum_i \sum_{j \neq i} \langle R_{i,t} R_{j,t} \rangle \\ &- \sum_i \sum_{j \neq i} \langle R_{i,t} \rangle \langle R_{j,t} \rangle \end{aligned} \tag{10}$$

The first part can be evaluated as following:

$$\begin{aligned} \sum_{i=1}^{N} Var(R_{i,t}) &= \sum_{i=1}^{N} ((1 - x_i)^t - (1 - x_i)^{2t}) \\ &= \sum_{i=1}^{N} (1 - x_i)^t - \sum_{i=1}^{N} (1 - x_i)^{2t} \\ &= \sum_{i=1}^{N} \langle R_{i,t} \rangle - \sum_{i=1}^{N} \langle R_{i,2t} \rangle \\ &= (N - \langle F_t \rangle) - (N - \langle F_{2t} \rangle) \end{aligned} \tag{11}$$

And the third part:

$$\begin{aligned} &\sum_i \sum_{j \neq i} \langle R_{i,t} \rangle \langle R_{j,t} \rangle \\ &= \sum_i \langle R_{i,t} \rangle \cdot \left( \sum_j \langle R_{j,t} \rangle - \langle R_{i,t} \rangle \right) \\ &= \sum_i \langle R_{i,t} \rangle \cdot (N - \langle F_t \rangle - \langle R_{i,t} \rangle) \\ &= (N - \langle F_t \rangle)^2 - (N - \langle F_{2t} \rangle) \end{aligned} \tag{12}$$

For the second part,

$$\sum_i \sum_{j \neq i} \langle R_{i,t} R_{j,t} \rangle = N(N-1) \cdot P_t$$

where $P_t$ is the following probability:

*Randomly pick two faults, what is the expected probability $P_t$ that both remain undetected after time $t$?*

Denote random variable $Z$ as the portion of vectors that detect neither of the two randomly picked faults, then the probability that both remain undetected is $Z^t$ and the expectation of the probability $P_t = \langle Z^t \rangle$, which is actually the $t$th moment of random variable $Z$. Because faults have different detectability $x_i$'s, and vectors have different coverage $y_j$'s, the distribution of $Z$ would be much difficult to obtain, not to mention a general expression for its higher moments $\langle Z^t \rangle$.

The expectation of $Z$, however, can be found by using the vector profile. Considering vector $v_j$ with coverage $y_j$, the probability it doesn't detect the first fault is $(1 - y_j)$. And the probability it doesn't detect both faults would be $(1 - y_j)^2$. Therefore the expected value of $Z$ is:

$$\begin{aligned} \langle Z \rangle &= \int_0^1 (1 - y)^2 g(y)\, dy \\ &= \int_0^1 (1 - 2y + y^2) g(y)\, dy \\ &= 1 - 2 \int_0^1 y g(y)\, dy + \int_0^1 y^2 g(y)\, dy \\ &= 1 - 2\langle y \rangle + \langle y \rangle^2 + Var(y) \\ &= (1 - \langle y \rangle)^2 + Var(y) \end{aligned}$$

Note that $\int_0^1 y^2 g(y) dy$, the 2nd moment of the vector profile, evaluates to $\langle y \rangle^2 + Var(y)$. Since $Z$ is a positive random variable, we have the following inequality (see Appendix C),

$$\langle Z^t \rangle \geq \langle Z \rangle^t$$

Therefore

$$P_t \geq ((1 - \langle y \rangle)^2 + Var(y))^t$$

and

$$\sum_i \sum_{j \neq i} \langle R_{i,t} R_{j,t} \rangle \geq N(N-1)[(1 - \langle y \rangle)^2 + Var(y)]^t \tag{13}$$

By applying delta method (see [17], pp. 54–55), an approximation to $P_t$ is:

$$P_t \cong \langle Z \rangle^t \left( 1 + \frac{t(t-1) Var(Z)}{\langle Z \rangle^2} \right)$$

Combine all the three parts in 11, 12, and 13, then we get a lower bound on the variance of the fault coverage:

$$\begin{aligned} Var(F_t) \geq\ & N(N-1)((1 - \langle y \rangle)^2 + Var(y))^t \\ &+ (N - \langle F_t \rangle) - (N - \langle F_t \rangle)^2 \end{aligned}$$

When normalized,

$$Var(F(t)) \geq \frac{N-1}{N}((1-\langle y\rangle)^2 + Var(y))^t + \frac{1-F(t)}{N} - (1-F(t))^2$$

And the approximation:

$$Var(F(t)) \cong \frac{N-1}{N}\langle Z\rangle^t\left(1 + \frac{t(t-1)Var(Z)}{\langle Z\rangle^2}\right) + \frac{1-F(t)}{N} - (1-F(t))^2$$

Write $\langle Z\rangle$ and $Var(Z)$ as constants $\psi_1$ and $\psi_2$ in (0, 1), we have:

$$Var(F(t)) \cong \frac{N-1}{N}(\psi_1)^t\left(1 + \frac{t(t-1)\psi_2}{(\psi_1)^2}\right) + \frac{1-F(t)}{N} - (1-F(t))^2$$

If we assume the detection of the faults are independent, i.e. the covariance $cov(R_{i,t}R_{j,t}) = 0$ then the second and third parts in formula 10 are canceled off. From formula 11, we have a simple approximation:

$$Var(F_t) \cong (N - \langle F_t\rangle) - (N - \langle F_{2t}\rangle)$$

After normalization,

$$Var(F(t)) \cong \frac{F(2t) - F(t)}{N}$$

## Appendix C: Jensen Inequality

Given positive random variable $X$ and a convex function $\phi$ such that

$$\phi(at + (1-t)b) < t\phi(a) + (1-t)\phi(b)$$

and if $\phi(X)$ is positive, then

$$\langle\phi(X)\rangle \geq \phi(\langle X\rangle)$$

Now that $x^t$ is a convex function,

$$\langle X^t\rangle \geq \langle X\rangle^t$$

See [18] for detail.

## Acknowledgment

Huan-Chih Tsai's contribution of some of the fault simulation data reported in the paper is gratefully acknowledged. This work was supported by the NSF Grant Nos. CCR-9806795 & CCR-9971167 and the University of Nebraska-Lincoln Center for Communication and Information Science.

## Note

1. See Appendix A for the definitions of beta and other mathematical functions used in this paper.

## References

1. G. Kiefer and H.-J. Wunderlich, "Deterministic Bist with Partial Scan," *Journal of Electronic Testing: Theory and Applications*, vol. 16, pp. 169–177, June 2000.
2. N.A. Touba and E.J. McCluskey, "Altering a Pseudo-Random bit Sequence for Scan-Based BIST," in *International Test Conference*, 1996, pp. 167–175.
3. J.P. Hayes and A.D. Friedman, "Test Point Placement to Simplify Fault Detection," *IEEE Trans. on Computers*, vol. C-33, pp. 727–735, July 1974.
4. H.-C. Tsai, K.-T. Cheng, C.-J. Lin, and S. Bhawmik, "Efficient Test-Point Selection for Scan-Based BIST," *IEEE Trans. on VLSI Systems*, vol. 6, pp. 667–676, Dec. 1998.
5. N.A. Touba and E.J. McCluskey, "Test Point Insertion Based on Path Tracing," in *14th IEEE VLSI Test Symposium*, 1996, pp. 2–8.
6. H.-C. Tsai, K.-T. Cheng, and S. Bhawmik, "Improving the Test Quality for Scan-Based Bist Using a General Test Application Scheme," in *Proceedings of the Design Automation Conference*, 1999, pp. 748–753.
7. S.K. Jain and V.D. Agrawal, "Statistical Fault Analysis," *IEEE Design & Test of Computers*, vol. 2, pp. 38–44, Feb. 1985.
8. V.D. Agrawal, "Sampling Techniques for Determining Fault Coverage in LSI Circuits," *Journal of Digital Systems*, vol. V, pp. 189–201, June 1981.
9. T. Williams, "Test Length in a Self-Testing Environment," *IEEE Design & Test of Computers*, vol. 2, no. 2, pp. 59–63, 1985.
10. P. Goel, "Test Generation Costs Analysis and Projections," in *Proceedings of the Design Automation Conference*, 1980, pp. 77–84.
11. S. Seth, V. Agrawal, and H. Farhat, "A Statistical Theory of Digital Circuit Testability," *IEEE Transactions on Computers*, vol. 39, no. 4, pp. 582–586, 1990.
12. H. Farhat and S. From, "A Beta Model for Estimating the Testability and Coverage Distributions of a VLSI Circuit," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 4, pp. 550–554, 1993.
13. S. From and H. Farhat, "Confidence Intervals for Expected Coverage from a Beta Testability Model," *Computers & Mathematics with Applications*, vol. 24, no. 7, pp. 97–107, 1992.

14. P.H. Bardell, W.H. McAnney, and J. Savir, *Built-in test for VLSI*, John Wiley & Sons, Inc., 1987.
15. C.E. Stroud, *A Designer's Guide to Built-In Self-Test*, Boston: Kluwer Academic Publishers, April 2002.
16. G. Seber and C. Wild, *Nonlinear Regression*, John Wiley & Sons, Inc., 1989.
17. N.L. Johnson, S. Kotz, and A.W. Kemp, *Univariate Discrete Distributions*, John Wiley & Sons, Inc., 1992.
18. M. Schaefer, "Note on the $k$-Dimensional Jensen Inequality," *The Annals of Probability*, vol. 4, no. 3, pp. 502–504, 1976.

**Hailong Cui** received the B.E. degree from Beijing Institute of Light Industry, China, in 1993, and the M.S. degree in computer science from Beijing University, China, in 1996, and the Ph.D. degree in computer science from University of Nebraska, Lincoln, in 2001.

He joined Qualcomm Inc. in the testing group in 2001. His current research interests include failure analysis, defect level estimation, and test plan optimization.

**Sharad Seth** is a professor of Computer Science and Engineering at the University of Nebraska-Lincoln and the Director of the Center for Communication and Information Science. His current research interests in the testing area include low-power test generation for built-in self-test circuits, function-based testability, and design verification. Other research interests include boolean synthesis and document image analysis. He serves on the Editorial Board of *JETTA*.

**Shashank Mehta** is an associate professor of computer science and engineering at Indian Institute of Technology, Kanpur. His current research effort is in hybrid test generation techniques using functional specification and partial structural information. His other interests include computational geometry and graph algorithms.