1990

A Statistical Theory of Digital Circuit Testability

Sharad C. Seth
University of Nebraska-Lincoln, seth@cse.unl.edu

Vishwani D. Agrawal
AT&T Bell Labs, Murray Hill, NJ

Hassan Farhat
University of Nebraska at Omaha

A Statistical Theory of Digital Circuit Testability

SHARAD C. SETH, VISHWANI D. AGRAWAL, AND HASSAN FARHAT

Abstract—**When test vectors are applied to a circuit, the fault coverage increases. The rate of increase, however, could be circuit dependent. A relation between the average fault coverage and circuit testability is developed in this paper. The statistical formulation allows computation of coverage for deterministic and random vectors. We discuss the following applications of this analysis: determination of circuit testability from fault simulation, coverage prediction from testability analysis, prediction of test length, and test generation by fault sampling.**

Index Terms—**Fault coverage estimation, probabilistic testability, random pattern testability, statistical sampling, testability measures.**

I. Introduction

Fig. 1 shows the nature of results obtained from fault simulation. It is speculated that the fault coverage of random vectors follows an exponential law [1]. There is no general agreement on how the coverage of deterministic vectors might be represented. Fig. 2 shows the distribution of faults in a circuit according to their detection probabilities [2]. Such a distribution is presumably useful in assessing the testability of a circuit. However, in the absence of an explicit relationship between the probabilistic testability and fault coverage, designers often find it difficult to use testability data to estimate the size of the required test vector set or the fault coverage of a given vector set. The specific problem solved in this paper is *to find a relationship between probabilistic testability and fault coverage.*

Applications of the analysis presented in this paper are 1) assessing circuit testability from fault simulation, 2) extrapolation of partial fault simulation results where full fault simulation is very expensive, 3) finding the size of test sets for random and deterministic vectors, and 4) fault sampling for test generation.

II. Fault Coverage and Circuit Testability

We will first define two quantities that are relevant to fault analysis and then establish a relation between them.

Detection Probability: The *detection probability* of a fault is the probability of detecting the fault by a random vector. Detection probabilities of faults in a circuit can be represented by a distribution $p(x)$:

$$p(x)\,dx = \text{Fraction of detectable faults with probability of detection between } x \text{ and } x + dx.$$

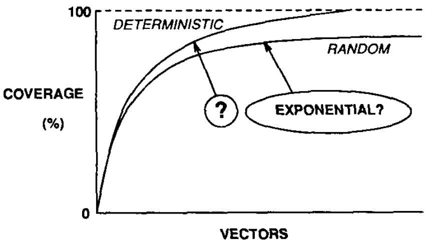

Fig. 1. Fault coverage.

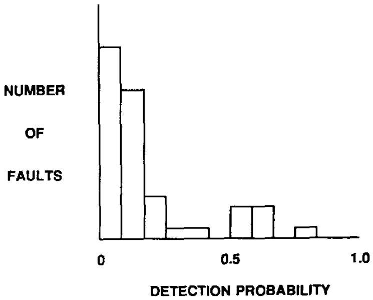

Fig. 2. A typical testability analysis result.

Since x represents probability, $p(x)$ is nonzero (and positive) only for values of x between 0 and 1. Also,

$$\int_0^1 p(x)\,dx = 1.$$

Notice that $p(x)$ is the distribution of only the *detectable* faults.

The distribution $p(x)$ for a circuit can be determined in several different ways. Testability analyses like PREDICT [3] and COP [4] determine fault detection probabilities to various degrees of accuracy. General sequential circuits can be analyzed through true-value simulation with random vectors [5]. In Section III, we present a method of estimating $p(x)$ from fault simulation.

Fault Coverage: Fault coverage is the percentage (or fraction) of faults covered by test vectors. Generally, this coverage is over the set of all single stuck-at faults after it has been reduced by fault collapsing. To remove ambiguity, we will use a slightly modified definition. Most large circuits contain some redundant faults. By definition, these faults cannot be detected by any test. The percentage of such faults is small but finite, usually less than 5%. We define coverage as

$$\text{Fault Coverage} = \frac{\text{detected faults}}{\text{total faults} - \text{redundant faults}}. \tag{1}$$

An alternative definition of fault coverage is sometimes used in which the number of redundant faults is added to detected faults instead of subtracting from the total faults [6], [7]. Even though finding all redundant faults may be very difficult, our method provides an estimation of fault coverage as defined by (1).

Fault Coverage of Random Vectors: Since there are $p(x)\,dx$ faults with detection probability x, the mean coverage among these faults by a random vector is $xp(x)\,dx$. Suppose we apply a sequence of random vectors to the circuit. The mean coverage by the first vector is

$$y_1 = \int_0^1 xp(x)\,dx.$$

Manuscript received July 7, 1989; revised November 6, 1989.

S. C. Seth is with the Department of Computer Science and Engineering, University of Nebraska, Lincoln, NE 68588.

V. D. Agrawal is with AT&T Bell Laboratories, Murray Hill, NJ 07974.

H. Farhat is with the Department of Mathematics, University of Nebraska, Omaha, NE 68182.

IEEE Log Number 8933880.

IEEE Transactions on Computers,
Volume: 39 , Issue: 4, pp. 582-586.
doi: 10.1109/12.54854

Actual coverage by a random vector may differ from the mean by a random quantity. However, the variance will be small for almost all circuits (this follows from the *central limit theorem* in statistics). After removing the faults detected by the first vector, the distribution of detection probabilities of the remaining faults can be shown to be $(1-x)p(x)$. Thus, the coverage of two vectors is

$$y_2 = y_1 + \int_0^1 x(1-x)p(x)\,dx = \int_0^1 x[1+(1-x)]p(x)\,dx.$$

Similarly, the coverage of n vectors is

$$\begin{aligned} y_n &= \int_0^1 x[1+(1-x)+(1-x)^2+\cdots+(1-x)^{n-1}]p(x)\,dx \\ &= 1-\int_0^1 (1-x)^n p(x)\,dx = 1-I(n). \end{aligned} \tag{2}$$

where $I(n)$ is the integral in the last equation. If we consider n as a continuous variable and define new variables, $\omega = -\ln(1-x)$ and $\xi = n+1$ then we have

$$F(\xi) = \int_0^\infty e^{-\xi\omega}P(\omega)\,d\omega$$

where $F(\xi) = 1 - y_{\xi-1}$ and $P(\omega) = p(1-e^{-\omega})$. The last equation represents the Laplace transform. If we consider the number of vectors analogous to *time* (number of vectors is, in fact, proportional to the testing time) and the detection probability distribution analogous to *frequency*, then the above analysis expresses a transform relation as is often used in the filter theory. Future investigations may reveal new applications in the present context.

Fault Coverage of Deterministic Vectors: We assume the deterministic vectors to have the following properties.

1) Every vector detects at least one new fault that was not covered by the previous vectors.

2) Every vector may also detect some previously undetected faults depending on their detection probabilities.

For sequential circuits, the same properties are applicable to *vector sequences*. For a combinational circuit with a total of Y faults, the coverage by the first deterministic vector is

$$y_1 = \frac{1}{Y} + \left(1-\frac{1}{Y}\right)\int_0^1 xp(x)\,dx.$$

The first term on the right-hand side is the coverage due to the fault for which this vector was generated and the second term is the random coverage from the remaining faults.

Similarly, the coverage by the first two vectors is

$$y_2 = y_1 + \frac{1}{Y} + \left(1-\frac{2}{Y}\right)\int_0^1 x(1-x)p(x)\,dx.$$

Here, the first term is the fault coverage by the first vector, the second term is the coverage of the single target fault for which the second vector is derived, and the third term is the additional random coverage by the second vector. Proceeding recursively, we obtain y_n in the following form:

$$y_n = 1 - I(n) + \frac{n}{Y}\left[1 + I(n) - \int_0^1 \frac{1-(1-x)^n}{nx}p(x)\,dx\right]. \tag{3}$$

The right-hand side contains three parts. The first part, $1-I(n)$, is the random detection as given by (2). The second part, n/Y, is the deterministic coverage by n vectors. The remaining part represents the reduction in the random coverage as faults are continuously being removed from the fault population for the purpose of deterministic vector generation. Equation (3) is valid only for those values of n for which $y_n \le 1.0$. We use the following approximation:

$$y_n \approx 1 - I(n) + \frac{n}{Y} \tag{4}$$

where $1 \ll n < Y$. This approximation is valid for large number of vectors; however, the number of vectors should not be nearly as large as the number of faults. Notice, that y_n attains the value 1.0 at some definite value of n rather than increasing asymptotically as in the random vector case.

III. Determination of $p(x)$ and $I(n)$

Suppose we simulate a set of n_s faults *with fault dropping*. That is, a fault is dropped from further consideration by the fault simulator as soon as it is detected. The fault set may contain all the faults or just a randomly selected subset of the faults in the circuit. With each simulated fault a random-first-detection (RFD) variable is associated. It is used to store the vector number at which the fault was first detected *randomly* by a test vector. Faults which are never randomly detected will have the RFD value undefined (or 0 if initialized that way). Since random detection is required, the RFD value of a fault targeted for deterministic test generation is not affected by the generated vector. During test generation, any fault found to be redundant is removed from the sample fault list. Let a fault f be randomly-first-detected at vector number i. Then, using Bayes theorem [8], f has the conditional detection probability distribution

$$p_i(x) = \frac{x(1-x)^{i-1}q(x)}{\int_0^1 x(1-x)^{i-1}q(x)\,dx} \qquad i = 1, 2, \cdots, N$$

where N is the number of test vectors. The probability density $q(x)$ in the above expression represents the *a priori* detection probability distribution of faults. For simplicity, we assume that before the detection data become available, the detection probability of a fault can be anywhere between 0 and 1. Thus, $q(x) = 1$ for $0 \le x \le 1$, and $q(x) = 0$, otherwise. This gives

$$p_i(x) = i(i+1)x(1-x)^{i-1} \qquad 0 \le x \le 1. \tag{5}$$

With each vector number i we have an associated w_i representing the number of faults whose RFD value is i. Further,

$$w_0 \triangleq n_s - \sum_{i=1}^{N} w_i$$

is the count of all the faults in the sample whose RFD value is not defined. Here n_s is the number of faults in the fault sample and N is the number of test vectors. A fault chosen as a target for test generation but not detected by any other vector will be included in this count. Every fault included in the w_0 count has the property that it was not randomly detected by any of the N vectors and thus will have the Bayesian detection probability distribution

$$p_0(x) = \frac{(1-x)^N q(x)}{\int_0^1 (1-x)^N q(x)\,dx}.$$

After evaluating the integral, using the uniform *a priori* distribution for $q(x)$, we get

$$p_0(x) = (N+1)(1-x)^N \qquad 0 \le x \le 1. \tag{6}$$

Equations (5) and (6) allow the determination of the complete detection probability distribution as follows:

$$p(x) = \frac{1}{n_s}\sum_{i=0}^{N} w_i p_i(x). \tag{7}$$

Since this estimate is a sum of $N+1$ random variables, for a reasonable accuracy, the number N of vectors should be large.

Evaluation of I(n): The integral $I(n)$, defined in (2), can be easily evaluated if we substitute the above expression for $p(x)$. On

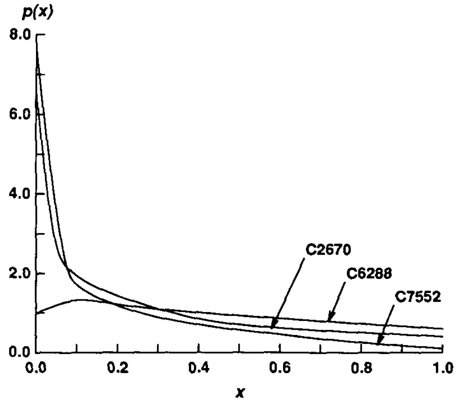

Fig. 3. Experimentally determined $p(x)$.

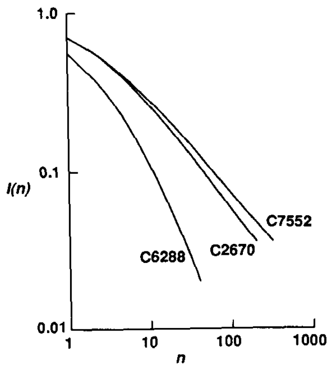

Fig. 4. Experimentally determined $I(n)$.

simplification, the following result is obtained:

$$I(n)=\frac{w_0(N+1)}{n_s(n+N+1)}+\frac{1}{n_s}\sum_{i=1}^{N}\frac{i(i+1)w_i}{(n+i)(n+i+1)}. \tag{8}$$

Again, the accuracy of this estimate will improve as the number N of vectors is increased. Once w_i's have been obtained from fault simulation, $I(n)$ can be computed from the above equation.

IV. Applications

We discuss four applications of the analysis presented above.

Testability Assessment: The function $p(x)$ (or the function $I(n)$ derived from it), represents the testability of the circuit. It can be determined by a topological analysis of the circuit [3], [4], in which case it represents testability by random vectors. However, a determination from fault coverage data will include the characteristics of test vectors also. In the earlier stages of a design, such an assessment of testability can be useful. Designers often write functional vectors for design verification. Since these vectors are not written for specific fault targets they can be regarded as random and used to determine $p(x)$. Large values of $p(x)$ near $x = 0$ will signal a testing problem.

In our model for deterministic test generation, we assumed that a test generated for a fault will behave like a random vector for other faults. Under the assumption, it is possible to estimate the functions $p(x)$ and $I(n)$ even during the standard (deterministic) test generation process as described in the last section. Figs. 3 and 4 show the results for three ISCAS circuits [2].

The $p(x)$ data in Fig. 3 were obtained in each case while generating tests for a sample of faults. Note that the random pattern testability exhibited in this figure is dependent not only on the circuit but also on the random pattern characteristics of the derived test vectors. As a simple measure of testability, we may use the area under the curve for detectabilities (x values) less than a certain threshold value, say, 0.1. Under the criterion, C6288 is significantly more testable than the other two circuits. Among the other two circuits, C2670 is slightly more testable than C7552. Similar conclusions can be drawn from the data for $I(n)$ shown in Fig. 4. These results are in agreement with the amount of test generation effort necessary for the three circuits.

TABLE I
Computed Fault Coverage for C7552

Vector Number	Random Coverage	Deterministic Coverage
5	60.8%	60.9%
20	81.1%	81.4%
50	89.4%	90.1%
100	93.4%	94.8%
140	94.9%	96.7%

Fault Coverage Determination: Once the functions $p(x)$ and $I(n)$ have been determined, the fault coverage can be estimated for any length of the vector set. Equation (2) is used for random vectors, and (4) for deterministic vectors.

As an example, we will consider the evaluation of fault coverage for the C7552 circuit using the data presented in Figs. 3 and 4. Table I summarizes the random and deterministic coverages predicted for this circuit.

Test Length: For any given fault coverage the required length of vector set can be easily predicted from (4). Such a prediction would be useful in planning of testing for a complex VLSI device.

As an example, from the $I(n)$ data for C2670 shown in Fig. 4, (4) would predict a deterministic test length between 80 and 90 vectors for a 95% fault coverage. We generated 113 vectors for this circuit using a Podem-based automatic test pattern generation program. From the fault simulator data, the coverage values of the first 80 and 90 vectors were determined to be 90.4% and 92.0%, respectively. This circuit is known to have 4.5% redundant faults [9] which must be subtracted from the total faults according to our definition [see (1)] of fault coverage. The modified values of the fault coverage, 94.2% and 96.3%, indeed span the 95% fault coverage for which we made the prediction.

Test Generation: The total cost of automatic test genration has two easily identified parts: the costs of test generation and fault simulation, respectively. The cost here refers to the use of computing resources (CPU time, memory, etc.). The fault simulation cost often predominates if the circuit is very large and/or is sequential. As an example, we provide the data on sequential test generation for a chip with 4856 faults. A random sample of 1000 faults was chosen for test generation. A sequence of 842 test vectors was generated and found to cover 98.2% of the faults in the sample. In a separate run, the fault coverage of the same sequence of test vectors was determined to be 82% over the whole fault population. The run times for this experiment on a VAX8650 computer were as follows:

Test Generation: 64 062 s
Fault Simulation:
Sample: 86 585 s
All faults: 462 234 s.

Reducing the relative cost of fault simulation in the test generation process is the primary motivation in the proposed approach.

Based on the analysis given earlier we propose a sampling method for test generation. In this method, vectors are generated using a random sample of faults. The analysis provides the size of the sampled fault set that will be required for any given fault coverage. Also, the coverage of the generated vectors over the entire fault population is estimated without simulating all the faults.

Let Y be total number of faults in the circuit of which a fraction s is randomly chosen for test generation. After n vectors have been generated the total fault coverage is given by

$$f(n)=sf_s+(1-s)[i-I(n)] \tag{9}$$

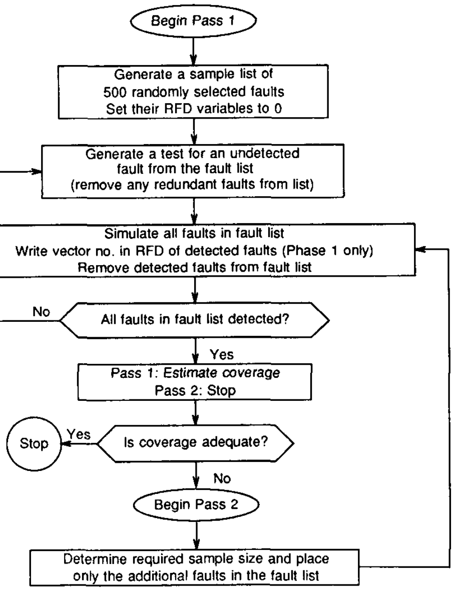
Begin Pass 1
Generate a sample list of 500 randomly selected faults Set their RFD variables to 0
Generate a test for an undetected fault from the fault list (remove any redundant faults from list)
Simulate all faults in fault list Write vector no. in RFD of detected faults (Phase 1 only) Remove detected faults from fault list
No
All faults in fault list detected?
Yes
Pass 1: Estimate coverage Pass 2: Stop
Stop
Yes
Is coverage adequate?
No
Begin Pass 2
Determine required sample size and place only the additional faults in the fault list

Fig. 5. Test generation by fault sampling.

where f_s is the coverage of n vectors in the sample. Thus, sf_s is the deterministic coverage contributed by the sampled faults and the second term gives the random coverage over the unsampled faults. Without loss of generality, in the following, we assume $f_s = 1$. That is, we will generate vectors to detect all faults in the sample. Suppose the number of these vectors is N. Then (9) reduces to

$$f(N) = 1 - I(N) + sI(N). \qquad (10)$$

Our proposed test-generation-by-fault-sampling is a two-pass procedure as shown in Fig. 5. Notice that this procedure differs from that described in another paper [6] where the faults were simulated without dropping the detected faults. We start Pass 1 with a random sample of 500 faults for test generation and assessment of testability as described in Section III. The size 500 of this initial sample is purely arbitrary and is chosen for convenience as it is neither too large nor too small. If a fault is determined to be redundant it is removed from the sample. When the sample is exhausted, the detection data (RFD variables of the sampled faults) are used to determine the counts w_i's for $p(x)$ and $I(N)$ computation. Then (10) is used to estimate the total fault coverage of the generated vectors. If the estimated fault coverage exceeds the desired coverage, say C, the test generation process can stop, otherwise, we carry out Pass 2 of test generation on a larger fault sample.

Let s' be the required sample size and assume that it is exhausted by generation of N' vectors. Making the appropriate substitutions in (10), we must have

$$C = 1 - I(N') + s'I(N'). \qquad (11)$$

In addition, rewriting (4) when a sample $s'Y$ of faults is completely covered by N' vectors, we have

$$N' = s'YI(N'). \qquad (12)$$

For any required fault coverage C, (11) and (12) can be solved numerically for s' by eliminating N'.

We illustrate the procedure for the C2670 circuit which has a fault population of 2747 single stuck type faults. A random sample of 500 faults was chosen for test generation. Of these, 12 faults were determined to be redundant by the test generator. The remaining 488-fault sample was exhausted by 65 test vectors. The $p(x)$ and $I(n)$ testability functions were obtained from this run as described in Section III. The 65 vectors were estimated to cover 94% of all the faults in the circuit. We chose 95% as the target fault coverage and determined the requisite sample size to be 35% (961 faults). We added an additional 500 randomly chosen faults to the original sample. Before restarting the test generation process we needed to simulate the 65 already generated vectors on these additional faults. In the second test-generation pass, an additional 31 test vectors were generated to cover a total of 978 faults in the enlarged sample; the remaining faults were determined to be redundant by the test generator. The estimated coverage of the 96 generated vectors was determined to be 96.4% according to (10). In a separate fault simulation run carried out for verification of results, the actual fault coverage of these vectors was determined to be 97.2% (this includes 4.5% redundant faults). Similar experiments were carried out for two other ISCAS circuits: C6288 and C7552. The results are summarized in Table II.

TABLE II
Test Generation by Fault Sampling

Sample Size	Circuit Name →	C2670	C6288	C7552
	Total Faults →	2747	7744	7550
500	Vectors →	65	34	77
	Adjusted sample size →	488	500	496
	Sample Cov. (%) →	100.0	100.0	100.0
	Estimated Cov. (%) →	94.0	97.0	93.3
	Measured Cov. (%) →	–	98.4	–
	Test gen. CPU Sec. →	1300	20	698
	Fault Sim. CPU Sec. →	14	20	45
978	Vectors →	96		
	Sample Cov. (%) →	100.0		
	Estimated Cov. (%) →	96.4		
	Measured Cov.(%) →	97.2		
	Test gen. CPU Sec. →	2900		
	Fault Sim. CPU Sec. →	20		
1485	Vectors →			142
	Sample Cov. (%) →			100.0
	Estimated Cov. (%) →			95.9
	Measured Cov. (%) →			95.6
	Test gen. CPU Sec. →			2855
	Fault Sim. CPU Sec. →			78

V. Conclusion

Briefly, the contributions of our work can be summarized as follows:

1) A statistical relationship is developed between circuit testability and fault coverage.

2) A method is presented to estimate circuit testability from fault simulation data collected in the normal course of test generation. The testability, so estimated, takes account of both the circuit topology and the characteristics of test vectors.

3) Applications of interest to test engineers include fault coverage prediction for random and deterministic vectors, test length prediction for a desired fault coverage, and test generation by fault sampling.

4) Several case studies verify the usefulness and precision of the proposed model.

We have presented a method of testability assessment through fault simulation with fault dropping. This is more economical compared to our earlier method of testability assessment which requires fault simulation *without* fault dropping [6]. As we pointed out, for a fault in a sequential circuit the *test* is not just a vector but is a sequence of vectors. The random vector coverage formula can still be applied to sequential circuits. However, the deterministic coverage part needs modification and will require further investigation.

Acknowledgment

We are indebted to Dr. K.-T. Cheng of AT&T Bell Laboratories for the data on sequential test generation reported in Section IV.

References

[1] T. W. Williams, "Test length in self-testing environment," *IEEE Design Test Comput.*, vol. 2, pp. 59–63, Apr. 1985.

[2] V. D. Agrawal, S. C. Seth, and C. C. Chuang, "Probabilistically guided test generation," in *Proc. Int. Symp. Circuit Syst. (ISCAS)*, Kyoto, Japan, June 1985, pp. 687-690.

[3] S. C. Seth, L. Pan, and V. D. Agrawal, "PREDICT—Probabilistic estimation of digital circuit testability," in *Fault-Tolerant Comput. Symp. (FTCS-15) Dig. Papers*, June 1985, pp. 220-225, also *FTCS-16 Dig.*, pp. 318-323.

[4] F. Brglez, "On testability analysis of combinational networks," in *Proc. Int. Symp. Circuit Syst.*, May 1984, pp. 221-225.

[5] S. K. Jain and V. D. Agrawal, "Statistical fault analysis," *IEEE Design Test Comput.*, vol. 2, pp. 38-44, Feb. 1985.

[6] V. D. Agrawal, H. Farhat, and S. Seth, "Test generation by fault sampling," in *Proc. Int. Conf. Comput. Design (ICCD-88)*, Rye Brook, NY, Oct. 1988, pp. 58-61.

[7] S. C. Seth, V. D. Agrawal, and H. Farhat, "A theory of testability with applications to fault coverage analysis," in *Proc. 1st Euro. Test Conf.*, Paris, France, Apr. 1989, pp. 139-143.

[8] A. Papoulis, *Probability, Random Variables, and Stochastic Processes.* New York: McGraw-Hill, 1965, sect. 4.4.

[9] M. H. Schulz and E. Auth, "Advanced automatic test pattern generation and redundancy identification techniques," in *18th Int. Symp. Fault-Tolerant Comput. (FTCS-18) Dig. Papers*, Tokyo, Japan, June 1988, pp. 30-35.

0018-9340/90/0400-0586$01.00 © 1990 IEEE