1989

A Theory of Testability with Application to Fault Coverage Analysis

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

Vishwani Agrawal
AT&T Bell Laboratories

Hassan Farhat
University of Nebraska at Omaha

Proceedings of the 1st European Test Conference, 1989.
doi: 10.1109/ETC.1989.36235

A Theory of Testability with Application to Fault Coverage Analysis

Sharad C. Seth
University of Nebraska
Lincoln, Nebraska 68588

Vishwani D. Agrawal
AT&T Bell Laboratories
Murray Hill, New Jersey 07974

Hassan Farhat
University of Nebraska
Omaha, Nebraska 68182

Abstract – **When test vectors are applied to a circuit, the fault coverage increases. The rate of increase, however, could be circuit-dependent. In fact, the actual rise of fault coverage depends on the characteristics of vectors, as well as, on the circuit. The paper shows that the average fault coverage can be computed from circuit testability. A relationship between fault coverage and circuit testability is derived. The mathematical formulation allows computation of coverage for deterministic and random vectors. Applications of this analysis include: determination of circuit testability from fault simulation, coverage prediction from testability analysis, prediction of test length, and test generation by fault sampling.**

1. Introduction

Figure 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 data are assumed to be 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: *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. Of these, the last application was recently described by us in [3]. To make the paper self-contained, we include the analytical framework introduced there and adapt it to the other applications. We also introduce a new technique for estimating the (vector-dependent) testability of a circuit from the results of fault simulation with *fault dropping*.

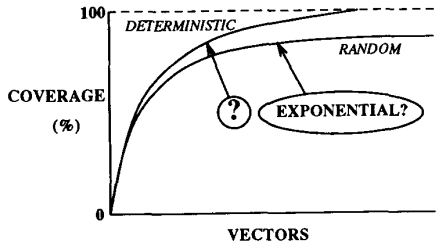

Fig. 1 Fault coverage.

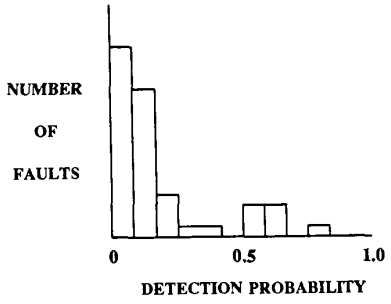

Fig. 2 Typical testability analysis result.

2. A Transform Relation

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 that fault by a random vector. Detection probabilities of faults in a circuit can be represented by a distribution $p(x)$:

$p(x)dx$ = Fraction of detectable faults with probability of detection between x and $x+dx$.

Since x represents probability, $p(x)$ is non-zero (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 [4] and COP [5] determine fault detection probabilities to various degrees of accuracy. General sequential circuits can be analyzed through true-value simulation with random vectors [6]. In Section 3, 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 can not be detected by any test. The percentage of such faults is small but finite, usually less than 5%. We define coverage as

$$\textit{Fault Coverage} = \frac{\textit{detected faults + redundant faults}}{\textit{total faults}} \quad (1)$$

An alternative definition of fault coverage is sometimes used in which the number of redundant faults is subtracted from the total faults instead of adding to detected faults. Even though finding all redundant faults may be very difficult, our method provides an estimation of fault coverage as defined by equation (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

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

Actual coverage by a random vector may differ from this average by a random quantity. However, this variance will be small for circuits with large number of faults (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

$$y_n = \int_0^1 x[1 + (1-x) + (1-x)^2 + \ldots + (1-x)^{n-1}]p(x)dx$$

$$= 1 - \int_0^1 (1-x)^n p(x)dx = 1 - I(n) \quad (2)$$

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.

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

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

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

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

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

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:

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

This equation is valid only for those values of n for which $y_n \leq 1.0$. We use the following approximation:

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

where $1 << n < Y$.

3. Determination of p(x) and I(n)

In [3] we proposed a method of estimating $p(x)$ and $I(n)$ by simulating a sample of faults *without* fault dropping However, the method could be expensive as it requires a change in the normal mode of test generation. In the following analysis we propose a more attractive alternative.

Suppose we simulate a set of n_s faults *with fault dropping*. That is, a fault is dropped from further consideration as soon as it is detected by the fault simulator. The fault set may contain all faults or just a randomly selected subset of all faults in the circuit. For each simulated fault a random-first-detection (RFD) flag is maintained. This flag remembers the vector number at which the fault was first detected *randomly* by a test vector. Since random detection is required, the flag 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 [7], f has the conditional detection-probability distribution:

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

where N is the number of test vectors. The probability density $q\,(x)$ in the above expression represents the *a priori* detection probability of a fault. For simplicity, we assume that before the detection data becomes 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

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

With each vector number i we will have an associated number w_i representing the number of faults whose RFD flags have the value 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 flag is not defined. Recall that n_s is the number of faults in the fault sample and N is the number of test vectors. Also note that a fault chosen as a target for test generation but not detected by any other vectors will be included in this count. The remaining faults included in the w_0 count might or might not be detectable. Even though the sample size n_s is adjusted to exclude faults that were found redundant (undetectable), it is not necessary to cover every detectable fault in the sample. In practice, test generators use some preselected time limit for abortion and, as a result, leave certain faults undetected without classifying them as redundant. 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, we get

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

under the uniform *a priori* distribution $q\,(x)$.

From equations (5) and (6), we can now write the complete detection probability distribution as follows:

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

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

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

Once w_i's have been obtained from fault simulation, $I\,(n)$ can be computed from the above equation.

4. 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. If it is determined by a topological analysis of the circuit [4,5], then 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. They can be 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 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

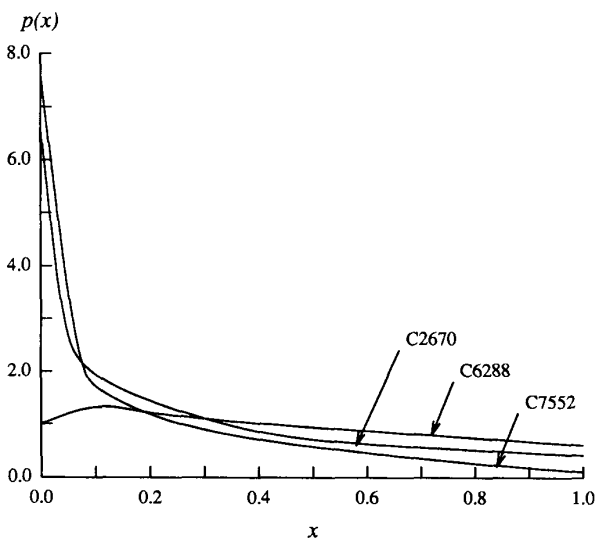

Fig. 3 Experimentally determined $p(x)$.

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

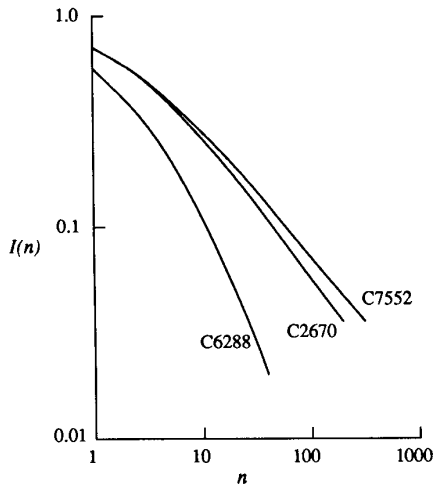

Fig. 4 Experimentally determined $I(n)$.

results are in agreement with the amount of test generation effort necessary for the three circuits.

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 equation (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. We remind the reader that the test vectors in this case are not really random but we assume that they cover non-targeted faults in a random fashion; the results obtained from another experiment where the vectors were generated by a truly random (or a pseudo-random) process are likely to be very different. The following table summarizes the random and deterministic coverages predicted for this circuit:

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%

Test Length. For any given fault coverage the required length of vector set can be easily predicted from equation (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 C7552 shown in Fig. 4, Eq. (4) would predict a deterministic test length of between 80 and 90 for a 95% fault coverage. We generated 113 vectors for this circuit using the test generation scheme described in the next subsection. Using a fault simulator, the fault coverage values for the first 80 and 90 vectors were determined to be, respectively, 90.4% and 92.0%. This circuit is known to have 4.5% redundant faults [8] which are included in our definition (see Eq. (1)) of the fault coverage but not in the coverage reported by the fault simulator. The modified values of the fault coverage, 94.9% and 96.5%, indeed span the 95% fault coverage for which we made the prediction.

Test Generation. Based on the analysis given above we have developed a sampling method for test generation [3]. 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 faults.

It is well known that that the complexity of fault simulation, which is one of the most expensive CAD tasks, grows with the number of faults being simulated. As an example, we provide the data on sequential test generation for a chip with 4,856 faults. A random sample of 1,000 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 seconds

Fault Simulation
- Sample: 86,585 seconds
- All faults: 462,234 seconds

Clearly, there is strong motivation for being able to predict the population coverage without spending such enormous amounts of computer time on fault simulation for all the faults.

Our proposed test-generation-by-fault-sampling approach can be summarized as follows. We start with an initial fixed size sample of 500 faults for test generation and assessment of testability as described in Section 3. If a fault is determined to be redundant it is removed from the sample. When the sample is exhausted, the detection data are used to determine the counts w_i's for $p(x)$ and $I(n)$ computation. Let s be the fraction of sampled faults exhausted by test generation and N be the number of vectors generated. Then the following equation can be used to estimate the fault coverage over all faults:

$$f(N) = 1 - I(N) + sI(N) \tag{9}$$

The reader is referred to [3] for a derivation of this relation. If the estimated fault coverage exceeds the desired coverage, say C, the test generation process can stop, otherwise, we carry out exactly one more cycle 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 Eq. (9), we must have

$$C = 1 - I(N') + s'I(N') \tag{10}$$

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

$$N' = s'Y\ I\ (N') \qquad (11)$$

For any required fault coverage, C, equations (10) and (11) can be solved numerically for s' by eliminating N'.

We illustrate the above procedure for the C2670 circuit which has a fault population of 2,747 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 obtained from this run are shown in Figs. 3 and 4 respectively. These 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 Eq. (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 1.

Table 1 – 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

5. Conclusion

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

(a) A precise relationship is developed between circuit testability and fault coverage.

(b) 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.

(c) 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.

Acknowledgment: We are indebted to Dr. K-T Cheng of AT&T Bell Laboratories for the data on sequential test generation reported in Section 4.

REFERENCES

[1] T. W. Williams, "Test length in self-testing environment," *IEEE Design & Test of Computers*, Vol. 2, pp. 59-63, April 1985.

[2] V. D. Agrawal, S. C. Seth, and C. C. Chuang, "Probabilistically Guided Test Generation," *Proc. Int. Symp. Circ. Syst. (ISCAS)*, Kyoto, Japan, pp. 687-690, June 1985.

[3] V. D. Agrawal, H. Farhat, and S. Seth, "Test Generation by Fault Sampling," *Proc. Int. Conf. Comp. Des. (ICCD-88)*, Rye Brook, NY, October 1988.

[4] S. C. Seth, L. Pan, and V. D. Agrawal, "PREDICT - Probabilistic Estimation of Digital Circuit Testability," *Fault-Tolerant Comp. Symp (FTCS-15) Digest of Papers*, pp. 220-225, June 1985, also FTCS-16 Digest, pp. 318-323.

[5] F. Brglez, "On Testability Analysis of Combinational Networks," *Proc. Int. Symp. Circ. and Syst.*, pp. 221-225, May 1984.

[6] S. K. Jain and V. D. Agrawal, "Statistical Fault Analysis," *IEEE Design & Test of Computers*, Vol. 2, pp. 38-44, February 1985.

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

[8] M. H. Schulz and E. Auth, "Advanced Automatic Test Pattern Generation and Redundancy Identification Techniques," *18th International Symp. on Fault-Tolerant Computing (FTCS-18) Digest of Papers*, Tokyo, Japan, pp. 30-35, June 1988.