1988

Test Generation by Fault Sampling

Vishwani Agrawal
AT&T Bell Laboratories, Murray Hill, NJ

Hassan Farhat
University of Nebraska at Omaha

Sharad C. Seth
University of Nebraska-Lincoln

Proceedings of the 1988 IEEE International Conference on Computer Design: VLSI in Computers and Processors, 1988. ICCD '88.
doi: 10.1109/ICCD.1988.25659

TEST GENERATION BY FAULT SAMPLING

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

Hassan Farhat
University of Nebraska, Omaha, NE 68182

Sharad Seth
University of Nebraska, Lincoln, NE 68588

ABSTRACT – **This paper presents a novel technique of generating tests from a random sample of faults. The entire fault population of the circuit is randomly divided into two groups. Only one group, usually the smaller one, is used for test generation by the test-generator and fault-simulator programs. This group is known as the sample and its coverage is deterministic. The coverage of faults in the remaining group is similar to that of random vectors and is estimated from the distribution of fault detection probabilities in the circuit. As the sample size increases, the fraction of unsampled faults reduces. At the same time, a larger sample yields more test vectors to increase the random coverage. The analysis in the paper determines the coverage of random and deterministic vectors from the detection probability distribution. However, a two-pass test generation process requires no prior knowledge of this distribution. Test generation in the first pass is started with an arbitrary sample size. From the deterministic coverage in the sample, the detection probability distribution is determined using the Bayes' theorem. Based on this distribution, the random coverage in the unsampled fault population is estimated and, if necessary, a second pass of test generation is executed with an appropriately larger sample. The sampling procedure is illustrated by several examples.**

Introduction

A typical test generation process is summarized as follows: select an as-yet-undetected fault, generate a test for it, and simulate all other faults; update the fault list by dropping the faults detected by the test; repeat until the desired fault coverage is reached.

The total cost of test generation has two easily identified components, namely, the costs of test generation and fault simulation, respectively. The second component could predominate if the circuit is very large or is sequential. The cost here refers to the use of computing resources (CPU, memory, etc.). Reducing the relative cost of fault simulation in the test generation process is the primary motivation for the present work.

Suppose only a randomly chosen sample of faults is initially placed on the fault list which is used to generate tests by the above procedure. Two questions need answers: 1) For a given coverage of the generated tests in the sample, what is the fault coverage for all faults? and 2) Can we determine the smallest sample size for tests to have a given coverage of all faults? Notice, this problem is different from that of fault sampling for coverage determination [1]. In sampling for coverage determination, we take a random sample of faults and determine the coverage of faults in the sample by the given tests. This coverage is a statistical estimate of the coverage over all faults. The accuracy of the estimate is dependent only on the sample size.

In coverage determination, the tests and the fault sample are derived through two independent processes. In test generation, on the other hand, we use some sample faults for generating tests. Thus our tests are somewhat biased toward detecting the faults in the sample. It is for this reason that the answers to the above questions are not obvious.

In this paper, we will provide a mathematical framework for probabilistic analysis and formulate procedures for test generation by fault sampling.

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 [2] and COP [3] determine fault detection probabilities to various degrees of accuracy. General sequential circuits can be analyzed through true-value simulation with random vectors [4]. In a later section, we will give a method of estimating $p(x)$ from fault simulation.

Definition of Fault Coverage

Fault coverage is defined as 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 by 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

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

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

where $I(n)$ is the integral in the last equation.

Fault Coverage by 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 other new 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 $y_n \leqslant 1.0$. We use the following approximation:

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

where $1 << n < Y$.

Sample Size Determination

Suppose we randomly sample a fraction s of the total of Y faults. We then generate n vectors for detecting faults in the sample. The total coverage is given by

$$f(n) = sf_s + (1-s)[1 - I(n)] \quad (5)$$

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 equation (5) reduces to

$$f(N) = 1 - I(N) + sI(N) \quad (6)$$

Of course, for a 100% fault coverage in the entire circuit, i.e., for $f(N) = 1$, either $s = 1$ (all faults sampled), or $N = \infty$ (infinite number of vectors applied). In general, for any other value of $f(N)$, the sample size can be obtained if we apply equation (4) to the sample where the fault population is sY instead of Y and $f_s = y_N = 1$. Thus,

$$N = sY \cdot I(N) \quad (7)$$

For any required coverage, $f(N)$, equations (6) and (7) can be solved numerically for s by eliminating N. The total number, Y, of faults in the circuit is known and we will determine $p(x)$ and $I(n)$, empirically.

Proposed Test Generation Procedure

A flow chart of the test generation procedure is given in Fig. 1. We do not assume any knowledge of the fault detection probabilities. Hence, a precise sample size can not be determined. For this reason, two passes are used. In Pass 1, a fault sample of arbitrary size (500 in Fig. 1) is used. At the end of this pass, using the available fault detection data, $p(x)$, $I(n)$, and the total fault coverage are estimated. If the fault coverage requirement is higher, then the necessary sample size is estimated from the now available $I(n)$. In Pass 2, the sample size is increased to this value.

Determination of p(x). Faults in the sample are simulated in Pass 1 without *fault dropping.* Thus, for each fault a random-detection count is determined. If a fault was used by the test generator as a target for deterministic test generation, the random-detection count of this fault is reduced by 1. The adjusted random-detection count of a fault is the number of times it is detected by the Pass 1 vectors. During test generation, any fault found to be redundant is removed from the sample fault list. Let n_s be the adjusted sample size. Also, let w_i be the number of faults in the sample with random-detection count of i, where $i = 0,1,2, \ldots,N$ and N is the number of vectors generated in Pass 1. Clearly,

$$\sum_{i=0}^{N} w_i = n_s.$$

Using Bayes' theorem [5], we can write

$$p(x) = \frac{1}{n_s}\sum_{i=0}^{N} w_i \, p_i(x) \quad (8)$$

where $p_i(x)$ is the *conditional* probability density function for the faults that were detected by i vectors and not detected by

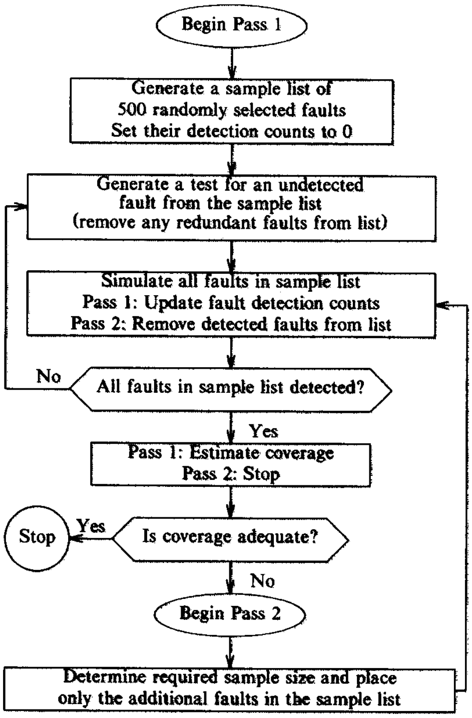
Begin Pass 1
Generate a sample list of 500 randomly selected faults Set their detection counts to 0
Generate a test for an undetected fault from the sample list (remove any redundant faults from list)
Simulate all faults in sample list Pass 1: Update fault detection counts Pass 2: Remove detected faults from list
No
All faults in sample 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 sample list

Fig. 1 Test generation by fault sampling.

$N-i$ vectors. This density function is given by

$$p_i(x) = \frac{x^i(1-x)^{N-i}q(x)}{\int_0^1 \xi^i(1-\xi)^{N-i}q(\xi)d\xi} \tag{9}$$

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, $q(x) = 1.0$, for $0 \leqslant x \leqslant 1.0$, and $q(x) = 0$, otherwise. This gives

$$p_i(x) = \frac{x^i(1-x)^{N-i}}{B(i+1, N-i+1)} \tag{10}$$

where the function in the denominator is known as the *beta function* [6] which, for computational purposes, can be represented either in terms of *gamma functions* or, for integer arguments, by factorials. Notice that $p_0(x)$ is the distribution of faults that were not detected but might 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 time limit and leave some faults undetected without classifying them as redundant.

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:

$$I(n) = \frac{1}{n_s}\sum_{i=0}^{N}\left(w_i \prod_{k=1}^{n}\frac{N-i+k}{N+1+k}\right) \tag{11}$$

Once w_i have been obtained from fault simulation in Pass 1, $I(n)$ is computed from the above equation. It is then used for estimating the total fault coverage and the required sample size as will be discussed in the next section.

Experimental Results

We used the proposed sampling method for generating tests for three of the larger ISCAS circuits [7]: C2670, C6288, and C7552. For each circuit, a random sample of 500 faults was used in Pass 1. Tests were generated using a Podem [8] test generator and a deductive fault simulator [9] running on a VAX 8600 computer. A graph of $I(n)$, as obtained from equation (11), is shown in Fig. 2.

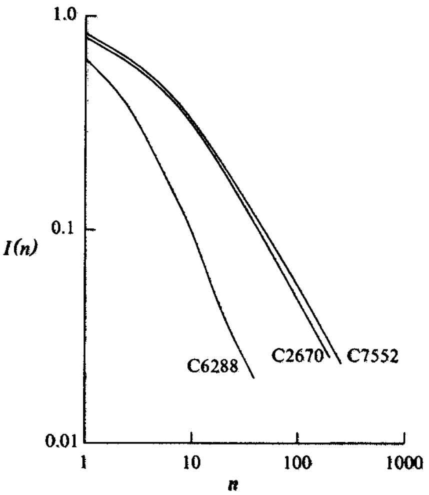

Fig. 2 $I(n)$ determined from 500-fault samples.

Next, we use equations (6) and (7) to obtain the required sample size s (expressed as a fraction of total faults Y) for a given fault coverage $f(N)$. The result is shown in Fig. 3. To illustrate its use let us assume that the required fault coverage is 95%. For the circuit C6288, we require $s = 0.05$. Since $Y = 7744$, we should use a random sample of $0.05 \times 7744 = 387$ faults. As Pass 1 already used a larger sample, Pass 2 is not needed. The required sample size for C2670 is $sY = 0.33 \times 2747 = 907$ and that for C7752 is $sY = 0.2 \times 7550 = 1510$. The second pass for these circuits was executed with 1000 and 1500 faults, respectively.

The results are given in Table 1. The estimated coverages shown in the Table were obtained from equation (6) and the data of Fig. 3. Measured coverages were obtained by fault simulation of all faults with vectors generated from fault samples. The CPU times are for a VAX 8600 computer.

The last section of Table 1 gives the result of test generation with all faults. The number of redundant faults, thus found, was used to adjust the measured fault coverage according to equation (1). The number of redundant faults is 117 (4.26%) in C2670, 34 (0.44%) in C6288, and 131 (1.74%) in C7552. In the case of C7552 circuit, 0.3% faults were left undetected by the test generator due to time limit.

The total CPU time shown in Table 1 includes the time of fault simulation which is also given separately. The total time is dominated by the attempted test generation runs for the faults that turned out to be either redundant or undetectable due to time limit. Fault simulation time for sample cases is always

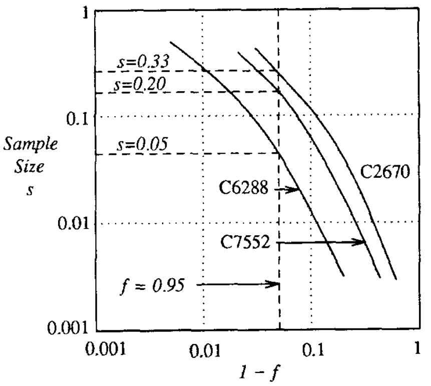

Fig. 3 Sample size versus fault coverage.

Table 1 — Test generation by fault sampling

Sample Size	Circuit Name →	C2670	C6288	C7552
	Total Faults →	2747	7744	7550
500	Vectors →	71	25	81
	Sample Cov. (%) →	99.4	100.0	99.2
	Estimated Cov. (%) →	94.5	97.2	93.3
	Measured Cov. (%) →	–	98.3	–
	Total CPU Sec. →	3246	36	1180
	Fault Sim. CPU Sec. →	13	13	42
1000	Vectors →	96		
	Sample Cov. (%) →	99.4		
	Estimated Cov. (%) →	96.8		
	Measured Cov.(%) →	97.3		
	Total CPU Sec. →	3966		
	Fault Sim. CPU Sec. →	18		
1500	Vectors →			149
	Sample Cov. (%) →			99.4
	Estimated Cov. (%) →			96.6
	Measured Cov. (%) →			96.5
	Total CPU Sec. →			5749
	Fault Sim. CPU Sec. →			79
All Faults	Vectors →	149	38	297
	Coverage (%) →	100.0	100.0	99.7
	Redundant Faults →	117	34	131
	Total CPU Seconds →	17186	110	33806
	Fault Sim. CPU Seconds →	32	52	174

smaller. However, a major advantage of sampling, which Table 1 does not show, is a significantly reduced memory requirement of fault simulation.

Figure 4 shows the estimated and the measured coverage in C7552 circuit for vectors generated from a sample of 1500 faults. The estimated coverage was obtained from equation (5) and the measured coverage has been adjusted by counting the redundant faults as detected.

Conclusion

It is shown that random fault sampling can be effectively used for test generation. The sample size is a function of the required fault coverage and the testability of the circuit. For a circuit with high testability (e.g., C6288), a sample of just 5% faults will provide tests for a 95% coverage. On the other hand, for a circuit with a relatively poor testability (e.g., C2670), one may have to sample 33% faults for the same coverage. For most practical cases, the sampling approach will mean significant saving in the computation and storage needs of fault simulation.

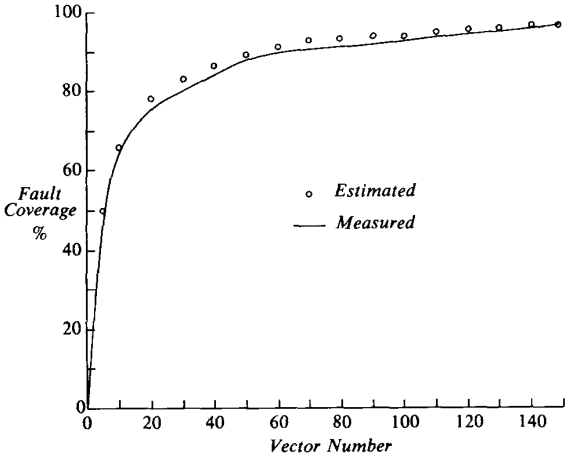

Fig. 4 Coverage of tests for C7552 circuit using a 1500-fault sample.

The two-pass test generation procedure eliminates the need for prior testability analysis to determine a precise sample size. The sample size is obtained as a by-product of the first pass. This determination, however, requires fault simulation without dropping the detected faults. Such a procedure would normally be considered wasteful. We have, therefore, developed an alternative formulation, also based on the Bayes' theorem, for determining the detection probability distribution that relies on fault simulation with fault dropping.

Acknowledgment — The authors are thankful to E. J. Moran for a discussion on test generation by fault sampling.

REFERENCES

[1] V. D. Agrawal, "Sampling Techniques for Determining Fault Coverage in LSI Circuits," *J. Digital Systems*, pp. 189-202, 1981.

[2] 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.

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

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

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

[6] I. S. Gradshteyn and I. M. Ryzhik, *Tables of Integrals, Series, and Products*, Academic Press, New York, 1980. Pages 938 and 948.

[7] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits and a target translator in Fortran," *Proc. Int. Symp. on Circuits and Systems; Special Session on ATPG and Fault Simulation*, pp. 663-698, June 1985.

[8] P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits," *IEEE Trans. Computers*, Vol. C-30, pp. 215-222, March 1981.

[9] D. B. Armstrong, "A Deductive Method for Simulating Faults in Logic Circuits," *IEEE Trans. Computers*, Vol. C-21, pp. 464-471, May 1972.