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

Follow this and additional works at: https://digitalcommons.unomaha.edu/mathfacpub

Part of the Computer Sciences Commons, and the Mathematics Commons

Recommended Citation
Agrawal, Vishwani; Farhat, Hassan; and Seth, Sharad C., "Test Generation by Fault Sampling" (1988). Mathematics Faculty Publications, 4.
https://digitalcommons.unomaha.edu/mathfacpub/4

This Article is brought to you for free and open access by the Department of Mathematics at DigitalCommons@UNO. It has been accepted for inclusion in Mathematics Faculty Publications by an authorized administrator of DigitalCommons@UNO. For more information, please contact unodigitalcommons@unomaha.edu.
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 \( p(x) \): 

\[
dx = \frac{\text{Fraction of detectable faults with probability}}{\text{of detection between } x \text{ and } x+dx}
\]

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

\[
\text{Fault Coverage} = \frac{\text{detected faults} + \text{redundant faults}}{\text{total faults}}
\]

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

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

Similarly, the coverage of \( n \) vectors is

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

\[
= 1 - \int_0^1 y^{n-1}p(x)dx = 1 - f(n)
\]

(2)

where \( f(n) \) 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 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:

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

(3)

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

\[
y_n \approx 1 - f(n) + \frac{n}{Y}
\]

where \( 1 \ll n < Y \).

**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

\[
f(n) = sf + (1-s)[1 - f(n)]
\]

(5)

where \( f \) is the coverage of \( n \) vectors in the sample. Thus, \( sf \) 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 \( f = 1 \). 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

\[
f(N) = 1 - f(N) + sf(N)
\]

(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 (6) to the sample where the fault population is \( sY \) instead of \( Y \) and \( f_s = y_s = 1 \). Thus,

\[
N = sY + f(N)
\]

(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 \( f(n) \), 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, \( p(x) \), \( f(n) \), 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 \( f(n) \). 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, then 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 \( n_i \) 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,

\[
\sum_{i=0}^{\infty} w_i = n_i
\]

Using Bayes' theorem [5], we can write

\[
p(x) = \frac{1}{n_i \sum_{i=0}^{\infty} w_i} p_i(x)
\]

(8)

where \( p_i(x) \) is the conditional probability density function for the faults that were detected by \( i \) vectors and not detected by
The probability density of detecting faults in a sample is given by

\[ p_i(x) = \frac{x^i(1-x)^{N-i}q(x)}{\int_0^1 (1-x)^{N-i}q(x) dx} \]  \tag{9}

where \( q(x) \) is the probability density of detecting faults in a sample, and \( p_i(x) \) is the probability density of detecting faults in a sample of size \( i \). The probability density \( q(x) \) is given by

\[ q(x) = 1, \quad 0 \leq x \leq 1, \quad q(x) = 0, \quad \text{otherwise} \]

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. 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 \( 0 \leq x \leq 1.0, \) and \( q(x) = 0, \) otherwise. This gives

\[ p_i(x) = \frac{x^i(1-x)^{N-i}}{B(i+1,N-i+1)} \]

The integral \( I(n) \) of equation (1), defined in equation (2), can be easily evaluated if we substitute the above expression for \( p(x) \). On simplifying, the following result is obtained:

\[ I(n) = \frac{1}{n!} \sum_{i=0}^{n} w_i \prod_{k=0}^{i-1} \frac{N-i+k}{N+1+k} \]  \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.

\[ I(n) \]

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 \( sY = 0.20 \times 7550 = 1510 \). 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
Table 1 — Test generation by fault sampling

<table>
<thead>
<tr>
<th>Sample Size</th>
<th>Circuit Name</th>
<th>C2670</th>
<th>C6288</th>
<th>C7552</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Faults</td>
<td>2747</td>
<td>7744</td>
<td>7580</td>
</tr>
<tr>
<td></td>
<td>Vectors</td>
<td>71</td>
<td>25</td>
<td>81</td>
</tr>
<tr>
<td>500</td>
<td>Sample Cov. (%)</td>
<td>99.4</td>
<td>100.0</td>
<td>99.2</td>
</tr>
<tr>
<td></td>
<td>Estimated Cov. (%)</td>
<td>94.5</td>
<td>97.2</td>
<td>93.3</td>
</tr>
<tr>
<td></td>
<td>Measured Cov. (%)</td>
<td>96.8</td>
<td>97.3</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Total CPU Sec.</td>
<td>3246</td>
<td>36</td>
<td>1180</td>
</tr>
<tr>
<td></td>
<td>Fault Sim. CPU Sec.</td>
<td>13</td>
<td>13</td>
<td>42</td>
</tr>
<tr>
<td>1000</td>
<td>Vectors</td>
<td>96</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sample Cov. (%)</td>
<td>99.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Estimated Cov. (%)</td>
<td>96.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Measured Cov. (%)</td>
<td>97.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total CPU Sec.</td>
<td>3966</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fault Sim. CPU Sec.</td>
<td>18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1500</td>
<td>Vectors</td>
<td></td>
<td>149</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sample Cov. (%)</td>
<td></td>
<td>99.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Estimated Cov. (%)</td>
<td></td>
<td>96.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Measured Cov. (%)</td>
<td></td>
<td>96.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total CPU Sec.</td>
<td></td>
<td>5749</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fault Sim. CPU Sec.</td>
<td></td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>All Faults</td>
<td>Vectors</td>
<td>149</td>
<td>38</td>
<td>297</td>
</tr>
<tr>
<td></td>
<td>Coverage (%)</td>
<td>100.0</td>
<td>100.0</td>
<td>99.7</td>
</tr>
<tr>
<td></td>
<td>Redundant Faults</td>
<td>117</td>
<td>34</td>
<td>131</td>
</tr>
<tr>
<td></td>
<td>Total CPU Seconds</td>
<td>1716</td>
<td>110</td>
<td>33806</td>
</tr>
<tr>
<td></td>
<td>Fault Sim. CPU Seconds</td>
<td>32</td>
<td>52</td>
<td>174</td>
</tr>
</tbody>
</table>

Figure 3 — Sample size versus fault coverage.

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

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 3% 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.

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