

TESTABILITY MEASURES -- WHAT DO THEY TELL US?

Vishwani D. Agrawal

Melvin Ray Mercer

Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT

Methods are developed for interpreting the information contained in testability measures. Two types of inferences are sought. First, the relationship between the testability measure for a fault and its detectability is investigated. Second, the question, how testability values can be used to predict the test length required to achieve some given fault coverage, is attempted. Real LSI circuits are used as illustrations.

INTRODUCTION

As logic circuits become more dense, and more complex, the problem of test generation, in general, exceeds the capacity of existing computing resources [1]. One well accepted approach for reducing testing difficulties is to consider circuit testability as early as possible in the design cycle. Several algorithms for measuring testability in digital circuits have been proposed [2]-[6]. Each of these algorithms assumes that testability is an inherent property of the circuit based solely upon the circuit structure (or topology). This assumption allows estimation of the circuit's testability before test generation is begun. One basic requirement of a testability analysis algorithm is that it should be computationally simpler than the test generation. If the testability calculation is fast and if it accurately predicts the difficulty of test generation, then problem circuits can be identified early in their design cycle, and remedial actions can be taken as part of the design. If the testability results are accurate for individual internal circuit nodes, then areas of poor testability can be identified. This information can be used to direct circuit modifications which improve testability.

Clearly good, computationally fast and accurate testability measures would be useful tools for digital circuit designers. Some of the currently used testability measure programs have been described in [7]-[9]. This paper does not suggest a new testability algorithm. It does not suggest modifications or enhancements to existing testability measures. Instead, we ask what useful information comes from a testability measure and we propose exact statistical metrics to evaluate the quality of information produced by a testability measure.

The examples use results from SCOAP, but the statistical criteria can be applied to any testability measure. Only recently, the workers in this area have started looking into interpretations of testability measures. Previous work [10]-[13] on evaluating testability measures has been aimed at comparing the composite testability of several circuits with actual test generation experience. In contrast, we compare detailed testability predictions for the faults within one circuit.

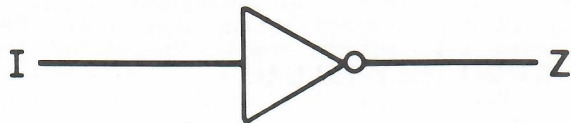
The key point here is that the utility of a designer's response to a testability measure will be determined by the quality of information in the testability measure's results. An automobile driver may be able to fairly accurately judge his speed by the level of engine noise and external visual perception if his speedometer does not work. In contrast, the rate of change of the amount of gasoline in the fuel tank is a poor indicator of the car's speed. The utility of a measure itself may also depend upon how easy it is to compute and to analyze.

The statistical calculations proposed below evaluate the utility of testability measures for the associated designer task.

LIMITATIONS OF TESTABILITY MEASURES

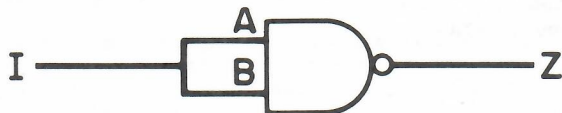
In this discussion we will use one of the testability measures, namely, SCOAP [5], [8]. Similar arguments can be applied to other measures. SCOAP produces six values for each node in a logic circuit: combinational zero controllability CC0, combinational one controllability CC1, combinational observability CO, sequential zero controllability SC0, sequential one controllability SC1, and sequential observability SO. The values of each of these variables grow larger as the required testing effort increases. These values "provide a quantitative measure of the difficulty of controlling and observing the logic value of internal nodes from considerations of circuit topology alone [5]". The values are *estimates* because of the simplifying assumptions made by the algorithm. For example, Figure 1 shows two realizations for a very simple digital circuit -- an inverter. The SCOAP algorithm produces different (CC0) values for the two realizations because it assumes all inputs to all logic elements are independent. In this case, the assumption is not valid for the NAND gate realization.

All testability measures have similar simplifying assumptions so that all results are estimates. The amount of information degradation resulting from such



$$CC\emptyset(Z) = CCI(I) = 1$$

a. Inverter realization



$$CC\emptyset(Z) = CCI(A) + CCI(B) = 2$$

b. NAND realization

Fig. 1 Two realizations of an inverter.

assumptions will vary with the circuit designs themselves and with the way the values are interpreted.

A second limitation of testability measures involves the restricted information source on which they rely. As shown in Figure 2, testability measures produce their results on the basis of circuit structure only. The actual testing process also involves a test pattern generator which is not considered by the testability measure. Clearly, the actual expense and difficulty of testing will depend heavily on the ability and sophistication of the test pattern generator as well as the circuit structure.

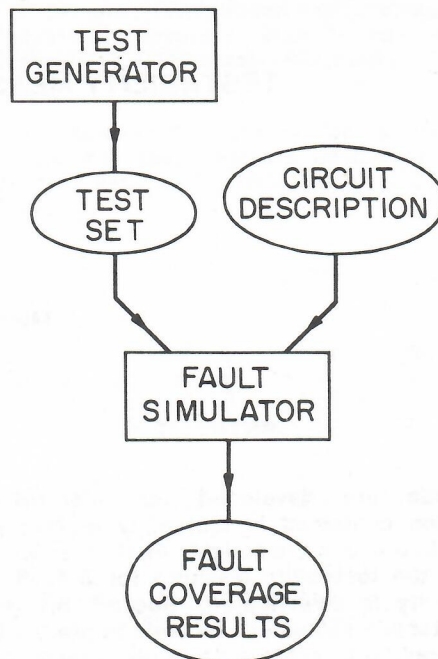
Other limitations exist, but the point is by now clear. Since the raw information from testability measures is not exact, some evaluation of the utility of testability values is necessary. This utility may vary considerably based upon the way in which the values are used.

WHAT DESIGNERS WANT TO KNOW

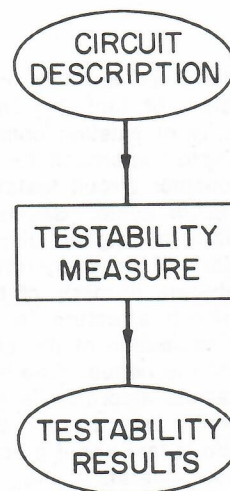
Usually, the test generation process goes in the following way. A set of single stuck-at faults is postulated for nodes in the circuit. Sufficient tests are generated to detect most of these faults. The results are: (1) a set of test patterns, (2) the fraction of detected faults (which serves as a figure of merit for the test set), and (3) the set of detected (or undetected) faults.

Two questions seem to be the most common from logic designers who are potential users of testability measures. The first question is: "Will this device require an inordinate amount of time, level of effort, and/or test length in order to provide acceptable testing?" If the answer to this question is "no", then the second question is never asked. For the case of potential problem designs, the next question is: "What are the problem areas in the design where a modification can ease the testing problem?"

The first question seems to be a simpler one than the second because the first requires some general



a. Fault simulation



b. Testability estimation

Fig. 2 Testability measures and fault simulation

information about the entire design. In contrast, the second question requires detailed information about individual components of the design.

Testability measures usually produce values for each node in the design. This information might be useful for answering the second question. These details must be combined in some way in order to provide information at the higher, overall design level implied in the first question.

Unfortunately, both questions are too vague to allow a meaningful statistical evaluation of the utility of testability measures. However, similar questions can be formulated which allow exact statistical evaluation.

FAULT-SPECIFIC TESTABILITY DATA

Every design eventually produces the results described in the last section--a set of test patterns and detected/undetected fault sets. The statistical measures presented below evaluate the accuracy with which these results can be predicted based upon testability analysis. All these measures compare the testability predictions with actual test pattern generation results for particular design cases.

SCOAP does not directly produce fault-specific data. At a point p in the circuit, there will be three combinational values:

- 1) $CC0(p)$,
- 2) $CC1(p)$,
- and 3) $CO(p)$.

Note that if p is stuck-at-zero, then to detect the fault it will be necessary to drive p to a logic one-- $CC1(p)$ -- and also to observe the resulting logic value at p -- $CO(p)$. Assuming that these tasks are independent, the total effort is:

$$\text{testability}(p \text{ stuck at zero}) = CC1(p) + CO(p).$$

Similarly:

$$\text{testability}(p \text{ stuck at one}) = CC0(p) + CO(p).$$

In this way, simple addition can be used to collect fault-specific measures from SCOAP results. If desired, the sequential SCOAP values ($SC0$, $SC1$, and SO) can also be included, but from our experience it seems that the combinational numbers are always much larger than the sequential numbers so that including the sequential parts in a simple summation makes little or no difference in the final result. Other ways of including sequential values in the above formulation were not investigated.

With the simple mapping described above, a numerical value can be assigned to every circuit fault based on SCOAP's testability values. The actual test generation process produces information about which faults are actually detected and which faults remain undetected. These two pieces of information can be compared statistically to measure the extent of agreement between testability predictions and actual fault detection results.

PROBABILITY OF DETECTION

Whether or not a fault is detectable is a deterministic phenomenon. In particular, all the nonredundant faults are potentially detectable. In practice, however, a test generation procedure has limits with respect to human effort, CPU time, number of test vectors, etc., and certain (non-redundant) faults will remain undetected. Under given circumstances, therefore, a fault will have only a certain probability of being detected. Since testability measures are often regarded as measures of the effort of test generation, we assume that the probability of

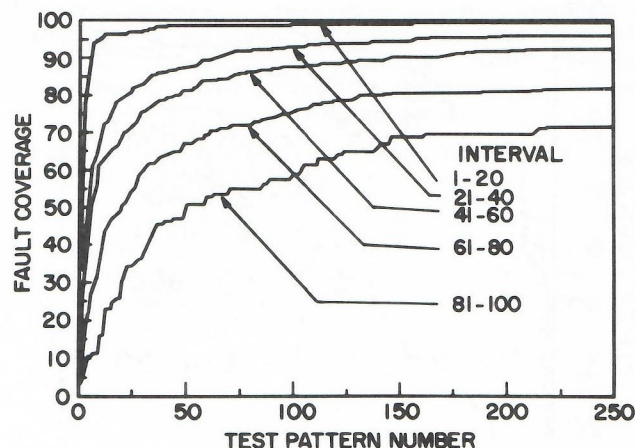


Fig. 3 Fault detection given testability for a large combinational circuit

detection (for a given test generation procedure) should be related to the testability measure.

CONDITIONAL PROBABILITY OF DETECTION GIVEN TESTABILITY VALUE

We ask: Is the probability of detection of a fault in fact dependent on its testability value? In order to answer this question, a set of testability intervals is selected. Each fault is assigned to exactly one testability interval based upon its testability value. The fraction of detected faults in each testability interval corresponds to the probability of detection of faults in the given testability interval. This statistical measure can be evaluated many times as the test patterns are applied during fault simulation. The result is a family of fault coverage plots where each member of the family corresponds to one testability interval.

Figure 3 shows an example of such results for a large combinational circuit with over 3000 faults. Since the testability values in SCOAP increase with required test generation effort the faults with low testability values are quickly detected, and their probabilities of detection quickly approach one. In contrast, faults with the greatest testability values are detected more slowly, and their final probability of detection is only about 0.7. For this example, the testability information seems to be good in that the fault detection performance consistently improves as testability improves. However, note that even the least testable faults have a .7 probability of detection. Thus, if a designer tries to improve circuit testability by reducing the testability values for all faults in the highest interval, then 70% of the effort is wasted.

Figure 4 shows a second example of fault detection performance given testability intervals. The two circuit sizes are almost identical, but this circuit is sequential in nature. For this case, the difference in fault detection performance is not consistently indicated by its testability value -- the plots cross one another, and faults in the testability interval 401-600 have a higher final probability of detection than those in the interval 201-400. Larger testability intervals would probably produce results more like those of Figure 3.

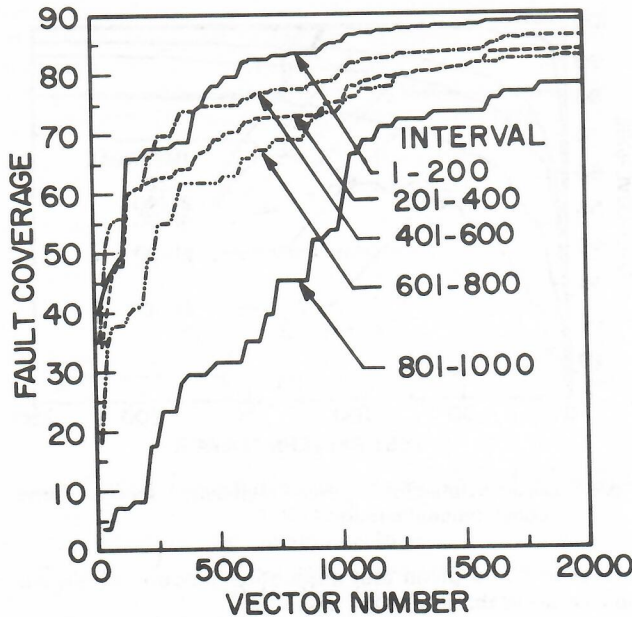


Fig. 4 Fault detection given testability for a large sequential circuit

Generally, the conditional probability of detection given testability value indicates how well the testability measure predicts the detectability of fault sets with similar testability values. This statistical measure also suggests what fraction of the faults in each testability interval will be detected.

Different testability measures or different versions of the same testability measure can be compared via this analysis. The measure which most effectively differentiates faults with high probabilities of detection from those with low probabilities of detection is the most desirable testability measure.

This statistical tool also can provide a figure of merit for a testability measure by indicating what fraction of faults in the least testable interval are in fact undetected after the application of all test patterns. An ideal testability measure would result in 100% of all of the most testable faults and 0% of the least testable faults being detected (for a suitable selection of testability intervals).

COEFFICIENT OF CORRELATION BETWEEN TESTABILITY AND DETECTION

A second statistical calculation provides one simple numerical value which is an estimator of the testability measure's ability to predict which individual faults will be detected and which individual faults will not be detected.

One random variable -- $d(f_i)$ -- is associated with each fault, f_i , in the circuit based on *detection results*. If a fault is detected during simulation, the value +1 is assigned to $d(f_i)$, if the fault is not detected, the value -1 is assigned.

A second random variable -- $\hat{d}(f_i)$ -- is associated with each fault in the circuit based on *testability value*.

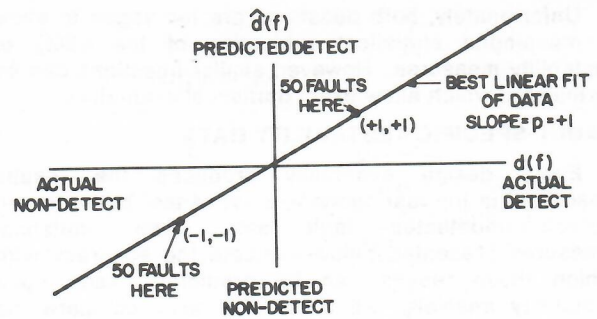


Fig. 5 An example of perfect correlation

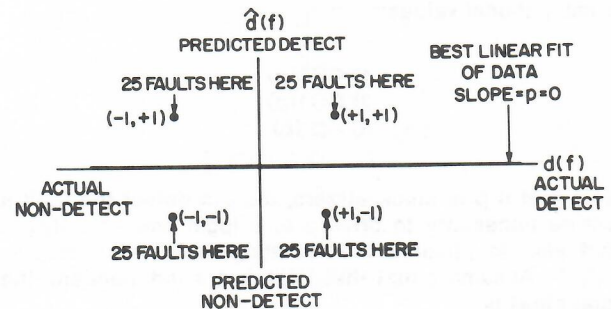


Fig. 6 An example of zero correlation

If a fault's testability is below a threshold to be discussed later, then a predicted detection is inferred, and $\hat{d}(f_i)$ is assigned the value +1. If a fault's testability is above the threshold, then a predicted non-detect is inferred, and $\hat{d}(f_i)$ is assigned the value -1.

The coefficient of correlation (ρ) between $d(f)$ and $\hat{d}(f)$ is a statistical estimate of the testability measure's performance in predicting whether or not faults will be detected. The value of ρ will vary with the selected threshold. In our analysis, all possible threshold values are considered, and the threshold which maximizes ρ is used.

Figure 5 shows one possible case where the two random variables are perfectly correlated and ρ has the value +1. Because (for this example) d and \hat{d} have zero mean and unit variance, ρ may be interpreted as the slope of the best least-mean-square linear fit to the data. In this case, the testability measure is a perfect predictor of fault detection.

Figure 6 shows a second case where the two random variables are uncorrelated and ρ has the value 0. Again d and \hat{d} have zero mean and unit variance, and the best linear fit is a line with zero slope (ρ). In this case, the testability measure gives absolutely no indication of fault detection.

In actual circuits, the value of ρ will be expected to have a value between +1 and 0. The closer ρ is to +1, the better the testability measure's ability to predict individual fault detections.

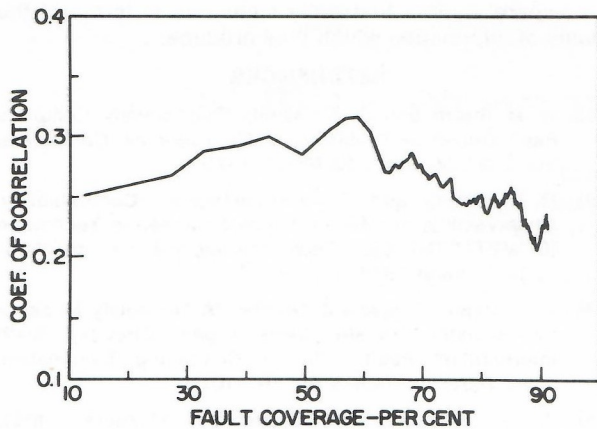


Fig. 7 Coefficient of correlation versus fault coverage for a large combinational circuit

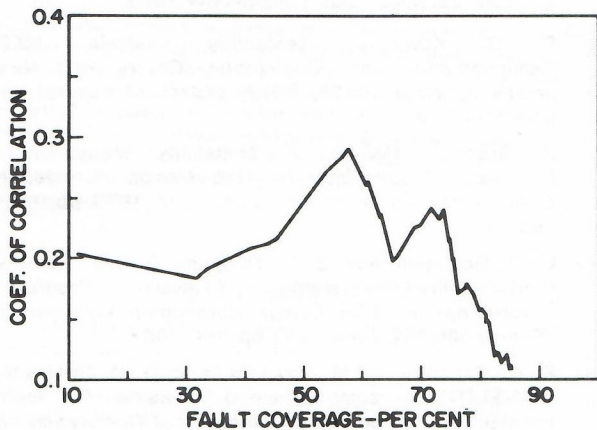


Fig. 8 Coefficient of correlation versus fault coverage for a large sequential circuit

The statistical measurement described above can be made at selected points in the test pattern sequence during fault simulation. The results can be plotted as ρ versus the circuit's fault coverage.

Figure 7 shows such a plot for the same combinational circuit used to produce Figure 3. Figure 8 shows a similar plot for the same sequential circuit used to produce Figure 4. Note that in both cases, the value of ρ is always relatively small (less than 0.4).

Figures 7 and 8 indicate that for these two circuits and their test sequences, the testability measure does a relatively poor job of predicting which individual faults will be detected and which individual faults will remain undetected. On the other hand, Figures 3 and 4 indicate that for the same circuits and test sequences, the same testability measure does a better job of predicting which fault sets have higher probabilities of detection and which fault sets have lower probabilities of detection.

Such data suggest that there is a level of resolution at which testability measures can provide useful information. Attempts to extract more detailed

information about a fault have a small likelihood of success. In particular, testability measures have little chance of predicting individual fault detection performance. Even so, general information at the entire circuit level may be available via testability measures.

DETECTION PROBABILITY AS A FUNCTION OF TESTABILITY MEASURE

In this section we give an analysis which predicts fault coverage information from the testability measure. Suppose a fault in a circuit has a testability measure t . Let its detection probability $p(t)$ be a function bounded between 0 and 1. For SCOAP, t will be the sum of the combinational observability and the combinational controllability of the faulty line. Since $1 \leq t \leq \infty$, we assume

$$p(t) = e^{-\alpha t}, \quad (1)$$

where α is a parameter to be discussed later. Notice that an infinite value of testability measure corresponds to a zero probability of detection. In SCOAP the effort of testing is represented by the computed values. Let us consider a circuit with N faults having testability measures t_1, t_2, \dots, t_N . If the fault detection performance of each test vector is statistically independent, then, after applying v vectors, the probability of detecting the i th fault is

$$1 - [1 - p(t_i)]^v$$

The fault coverage is then given by

$$\begin{aligned} f(v) &= \frac{1}{N} \sum_{i=1}^N \left\{ 1 - [1 - p(t_i)]^v \right\} \\ &= 1 - \frac{1}{N} \sum_{i=1}^N [1 - p(t_i)]^v \end{aligned} \quad (2)$$

Substituting (1) into (2), we get

$$f(v) = 1 - \frac{1}{N} \sum_{i=1}^N [1 - e^{-\alpha t_i}]^v \quad (3)$$

The value of α should depend upon the test generation procedure. In other words, α should relate the test generator to the testability measure which is a function of circuit topology alone.

We will determine α by an experimental procedure. Suppose we have an ordered set of v vectors for a circuit such that the i th fault, which has a testability measure t_i , is detected on the v_i th vector. Then the probability of this event is

$$[1 - p(t_i)]^{v_i-1} p(t_i).$$

We define the likelihood function or the joint probability of detecting the faults in this particular order as

$$L = \prod_{i=1}^N [1 - p(t_i)]^{v_i-1} p(t_i)$$

The value of α that maximizes this probability is called its maximum likelihood estimate. Substituting from (1), and taking logarithm, we get

$$\ln L = -\alpha \sum_{i=1}^N t_i + \sum_{i=1}^N (v_i - 1) \ln [1 - e^{-\alpha t_i}]$$

After differentiating with respect to α and setting the derivative to zero, we obtain the maximum likelihood estimate of α as follows:

$$\frac{1}{L} \frac{dL}{d\alpha} = - \sum_{i=1}^N t_i + \sum_{i=1}^N \frac{(v_i-1) e^{-\alpha t_i} t_i}{1 - e^{-\alpha t_i}} = 0$$

$$\text{or } \sum_{i=1}^N t_i \left[1 - \frac{(v_i-1) e^{-\alpha t_i}}{1 - e^{-\alpha t_i}} \right] = 0 \quad (4)$$

For $\alpha = 0$, the left hand side (LHS) in (4) is $-\infty$. Also, for $\alpha = \infty$, LHS = $\sum_{i=1}^N t_i > 0$. Thus (4) has a solution in the range $0 < \alpha < \infty$.

For the 3,000 gate circuit considered in the previous discussion, a set of 2,000 vectors was generated. Fault simulation determined the vector number (v_i) on which the i th fault was detected. Of the 3949 nonredundant faults 3371 faults were detected by the vector set. The SCOAP program was used to calculate t_i for each fault. Solving (4) numerically, α was obtained as 0.012. This value when substituted in (3) gave estimated fault coverage of 87 percent at 2,000 vectors, 92 percent at 4,000 vectors, and 95 percent at 10,000 vectors.

CONCLUSIONS

It seems clear that good testability information is of significant use to logic designers especially if it is available early in the design cycle. Existing testability measures do provide some information about the circuit as it is implemented. However, care and good engineering judgement should be used in the interpretation of testability results.

For the design cases which have been presented, some conclusions can be drawn. First, the testability data provide a relatively poor indication of whether or not an individual fault will be detected by a given test. Second, the testability data contain some indication of the probability of detection of a fault. The quality (or resolution) of these data seems to vary from one circuit to another. Finally, statistical tools have been presented to estimate the required testing effort for given testability values. The useful applications of this estimate require further investigation.

The statistical tools described here should be useful for any design effort which employs testability measures as part of the design process. They can be used to indicate the level of confidence which should be given to testability results. These same tools can also be used

to compare various testability measures in terms of the quality of information which they produce.

REFERENCES

- [1] O. H. Ibarra and S. K. Sahni, "Polynomially Complete Fault Detection Problems," *IEEE Trans on Computers*, Vol. C-24, pp. 242-249, March 1975.
- [2] H. Y. Chang and G. W. Heimbigner, "Controllability, Observability, and Maintenance Engineering Technique (COMET)," *Bell Syst. Tech. Journal*, Vol. 53, pp. 1505-1534, October 1974.
- [3] J. E. Stephenson and J. Grason, "A Testability Measure for Register Transfer Level Digital Circuits," Sixth International Fault Tolerant Computing Symposium, 1976, *Digest of Papers*, pp. 101-107.
- [4] J. A. Dussault, "A Testability Measure," IEEE Semiconductor Test Conference, Cherry Hill, New Jersey, Oct.-Nov., 1978, *Digest of Papers*, pp. 113-116.
- [5] L. H. Goldstein, "Controllability/Observability Analysis of Digital Circuits," *IEEE Trans. on Circuits and Systems*, Vol. CAS-26, pp. 685-693, September 1979.
- [6] P. G. Kovijanic, "Testability Analysis," IEEE Semiconductor Test Conference, Cherry Hill, New Jersey, October 23-25, 1979, *Digest. of Papers*, pp. 310-316.
- [7] J. Grason, "TMEAS, A Testability Measurement Program," *Proceedings of 16th Design Automation Conference*, San Diego, CA, June 25-27, 1979, pp. 156-161.
- [8] L. H. Goldstein and E. L. Thigpen, "SCOAP: Sandia Controllability/Observability Analysis Program," *Proceedings of 17th Design Automation Conference*, Minneapolis, MN, June 1980, pp. 190-196.
- [9] R. G. Bennetts, C. M. Maunder, and G. D. Robinson, "CAMELOT: a computer-aided measure for logic testability," *Proceedings of International Conference on and Circuits and Computers*, Port Chester, N.Y., Oct. 1-3, 1980, pp. 1162-1165, also *IEE Proc.*, Vol. 128, Pt.E, p. 177-189, September 1981.
- [10] P. G. Kovijanic, "Single Testability Figure of Merit," IEEE Semiconductor Test Conference, Philadelphia, PA., Oct. 27-29, 1981, *Digest of Papers*, pp.521-529.
- [11] B. Dunning and P. Kovijanic, "Demonstration of a figure of merit for inherent testability," IEEE Autotescon, Orlando, Florida, Oct. 19-21, 1981, pp. 515-520.
- [12] S. Takasaki, "Testability Measure Analysis at the Functional Level," Fifth Annual IEEE Workshop on Design for Testability, Vail Colorado, April 21,22, 1982.
- [13] J. Hickman and Theo Powell, "Pre Analysis of LSI Gate-Array Designs," 1982 IEEE Computer Elements Workshop, New York, N. Y., May 20-21, 1982.