

# Probabilistic Analysis of Random Test Generation Method for Irredundant Combinational Logic Networks

PRATHIMA AGRAWAL, STUDENT MEMBER, IEEE, AND VISHWANI D. AGRAWAL, MEMBER, IEEE

**Abstract**—In this paper the random test generation method for large logic circuits is analyzed. Formulas for the detection probability and the number of random input patterns required to complete the test generation with a high probability are obtained for an irredundant fan-out-free combinational network tree consisting of identical  $n$ -input NAND gates. The quantitative estimates for the number of random input patterns required for test generation appear to depend upon the number of levels in the circuit and the fan-ins of the gates. Experimental results for actual computer logic circuits are given and show the validity of the approach.

**Index Terms**—Combinational networks, detection probability, fault detection, path sensitizing, probabilistic analysis of logic, random test generation.

## I. INTRODUCTION

TEST GENERATION for both combinational and sequential logic networks can be broadly classified as either probabilistic or deterministic. An example of the former is the random test generation method [1]–[5]. One-dimensional path sensitizing [6], analytical techniques such as Boolean difference [7], algorithmic techniques as the  $D$ -algorithm [8], [9] fall under the latter category. Breuer [3] gives both random and algorithmic methods for generating tests for sequential circuits. A general review of these methods can be found in [10]–[13].

In this paper, we give an analysis of the random test generation method. For simplicity, we shall confine ourselves to irredundant combinational circuits with single stuck type, i.e., s-a-0 and s-a-1, faults. In the random method, the effect of a failure is propagated to the circuit output by applying random stimuli to the primary inputs. By using a simple simulator, the outputs of the faulty and the fault-free networks are compared. If the comparison fails, then the random input pattern is retained as a test. For complete fault detection, this experiment is continued until each fault has been detected by at least one pattern.

For the purpose of probabilistic analysis, a simple tree structure consisting of identical  $n$ -input NAND gates

[14]–[16] is used as a model. The lines from the primary inputs to the primary outputs are identified according to the levels at which they occur. All the primary inputs are at the level zero and the levels are incremented by one, every time the signal path goes through a gate. The level of a gate is the same as the level of its output line. Using two valued logic, the probabilities of a line in a particular level  $l$  assuming the value 0 or 1 are obtained as a function of  $l$  and the number of fan-ins  $n$  of the gate. From these a formula for the probability of sensitizing a path through  $L$  levels by at least one out of  $M$  independent random input patterns is derived.

Theoretical and experimental results for circuits of different maximum number of levels  $L$  are presented. These results show the validity of the analysis model. They also show how the theory will aid the user of random test generation method in obtaining the number of random patterns which will provide a complete test set with a high probability.

## II. PROBABILISTIC ANALYSIS OF THE MODEL

For analysis we consider a fan-out-free tree structure as shown in Fig. 1 (a). The basic block in this tree is the  $n$ -input NAND gate of Fig. 1(b). It is also assumed that all the inputs of a gate belong to the same level. Let the probabilities of a logical 0 and 1 occurring at level  $l$  be  $p_l^0$  and  $p_l^1$ , respectively. Then  $p_l^0$  is given by [17],

$$p_l^0 = (p_{l-1}^1)^n = (1 - p_{l-1}^0)^n, \quad (1)$$

since this is the probability of having  $n$  1's in the  $(l-1)$ th level. Clearly, the probability of a 1 occurring in the level  $l$  is

$$p_l^1 = 1 - p_l^0 = 1 - (p_{l-1}^1)^n. \quad (2)$$

*Example:* For a two-input NAND tree, assuming  $p_0^0 = p_0^1 = 0.5$ ,  $p_1^0 = 0.25$  and  $p_1^1 = 0.75$  for level 1. For level 2,  $p_2^0 = 0.5625$ ,  $p_2^1 = 0.4375$ , and so on.

The number of fan-ins at any level  $l$  can be expressed as a function  $n(l)$ . In general,  $n(l)$  may be regarded as a random variable. However, for simplicity, in our model we assume  $n(l) = n$ , for all  $l$ . Assuming equal probabilities for the occurrence of 1's and 0's at level zero,  $p_l^1$ , and  $p_l^0$  are shown in Fig. 2 for  $n = 2$ . It is seen that these probabilities oscillate around 0.5 and for large  $l$ , they practically take the extreme values (0 or 1) in the alternate levels. For larger  $n$ , the stabilization of probabilities

Manuscript received March 22, 1974; revised November 25, 1974. This work was supported in part by the NSF under Grant 23886.

P. Agrawal was with the Computer Center, Indian Institute of Technology, New Delhi, India. She is now with the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, Calif. 90007.

V. D. Agrawal is with the School of Radar Studies, Indian Institute of Technology, New Delhi, India.

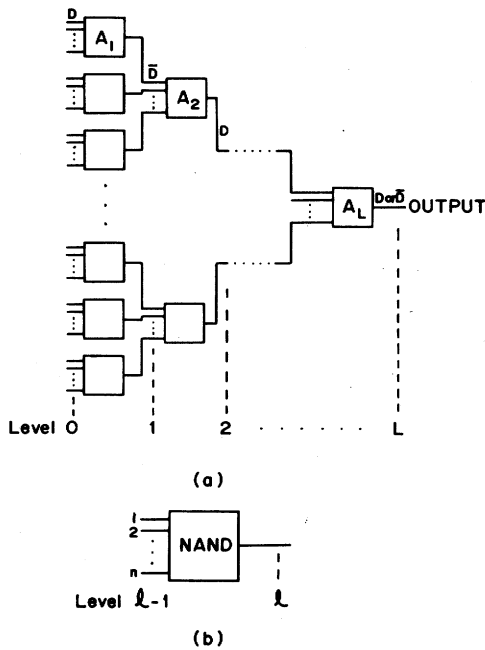


Fig. 1. NAND tree network used as analysis model.

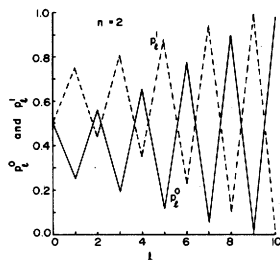


Fig. 2. Probabilities  $p_l^0$  and  $p_l^1$  as functions of  $l$  for  $n = 2$ .

oscillating between the extreme values will occur at much lower levels.

Let us now compute the probability of sensitizing a path of length  $L$ . By sensitizing a path, we mean propagating the effect of the fault from the site of its occurrence to a primary output through  $L$  levels. For the fan-out-free NAND tree, this problem reduces to that of finding  $(n - 1)$ , 1's at the inputs of each of the  $L$  gates along the path, since the effect of the fault is present in one input of each gate. In the notation of  $D$ -algorithm [8] this is similar to having a  $D$  or a  $\bar{D}$  at one input of each gate.  $D(\bar{D}) = 1(0)$  for the fault-free network and  $0(1)$  for the faulty one. We use this notation for representing and propagating faults due to the clear insight it provides into the process of path sensitization. For example, in Fig. 1(a), to propagate a fault at one input of gate  $A_1$ , to the output, all the inputs of the gates  $A_1, A_2, \dots, A_L$ , which are not  $D$  or  $\bar{D}$ , should be set to 1.

Next, let  $P(L)$  be the probability of propagating a  $D$  or  $\bar{D}$  through a path of length  $L$ . We shall consider only the faults involving the primary inputs, as every fault in a fan-out-free network is indistinguishable from an input fault [16]. The probability  $P(L)$  can be written as

$$P(L) = P(0) \prod_{k=0}^{L-1} (p_k^1)^{n-1}, \tag{3}$$

where  $(p_k^1)^{n-1}$  is the probability of having  $(n - 1)$ , 1's in level  $k$  and  $P(0)$  is the probability of a  $D$  or  $\bar{D}$  occurring at the fault site. For the input faults, without the loss of generality,  $P(0) = 1$ .  $L$  is the level of the primary output. We call  $P(L)$ , which is the probability of sensitizing a path of length  $L$  by a random input, the detection probability. For example, for a NAND tree with  $n = 2$ ,  $P(1) = 0.5$ ,  $P(2) = 0.375$ ,  $P(3) = 0.164$ , and so on. Since we are considering only the input faults, the same pattern which detects the s-a-0 fault, will also detect the s-a-1 fault just by changing the  $D$  on the particular input line to a  $\bar{D}$ .

Now let  $M$  independent random input patterns be applied to detect the fault. If  $P(L, M)$  is the probability of sensitizing a path of length  $L$  by at least one out of  $M$  patterns, then

$$\begin{aligned} P(L, M) &= 1 - (\text{Prob. of none of the } M \text{ patterns} \\ &\quad \text{sensitizing the path}) \\ &= 1 - \{1 - P(L)\}^M. \end{aligned} \tag{4}$$

Solving (4) for  $M$ , we get

$$M = \frac{\log \{1 - P(L, M)\}}{\log \{1 - P(L)\}}. \tag{5}$$

$P(L, M)$  can be considered as a measure of confidence we attain in sensitizing a path of length  $L$  by  $M$  patterns.

Computations were done for the case where the primary inputs can assume the values 0 and 1 with equal probabilities, i.e.,  $p_0^0 = p_0^1 = 0.5$ . Detection probability was computed for various values of  $L$  using (2) and (3), and is shown in Fig. 3 for  $n = 2$  and 3. The zig-zag nature of these curves is not surprising since for the NAND tree, the probability  $p_l^1$  oscillates around 0.5. This figure clearly indicates the difficulty of sensitizing a path as the complexity of the circuit increases. The complexity of a fan-out-free network, evidently, is a function of the fan-ins and the number of levels. The points on this figure were obtained experimentally as described in the next section. Fig. 4 shows  $M$  as computed from (5) for  $P(L, M) = 0.5, 0.9$  and  $0.99$ , and  $n = 2$ . This gives, quantitatively, the effort required for sensitizing a path of any length by random inputs.

### III. SIMULATED EXPERIMENT

An experiment was performed on an actual printed circuit card of a computer. This card consisted of integrated circuit packages containing {AND, OR, NOT} combinational logic. There were 56 inputs, 4 outputs, and 72 gates which formed the longest path of nine levels. Since an equivalent circuit using only NAND logic is always possible [16], [17], the probabilistic results of the previous section may be expected to be applicable, at least macroscopically. Random input patterns, in which each input can be either 0 or 1 with equal probability, were generated on a computer and the circuit response was obtained from

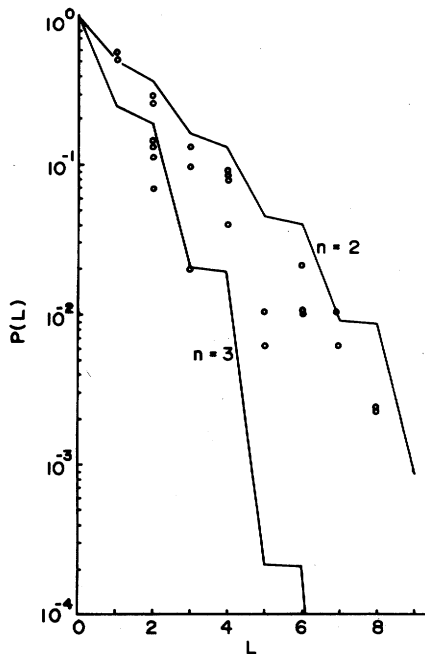


Fig. 3. Detection probability,  $P(L)$ , as a function of the number of levels,  $L$ , in the path. (0—Experimental results.)

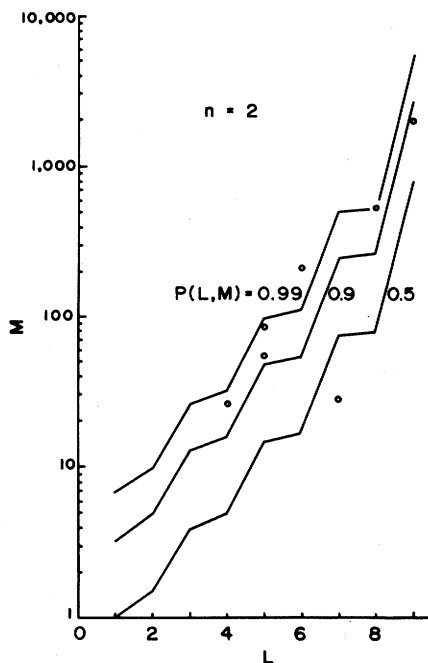


Fig. 4. Number of random input patterns,  $M$ , required to sensitize a path of  $L$  levels with 50, 90 and 99 percent probabilities. (0—Experimental results.)

a simulator. A TEST-DETECT program [5], [8] was then used to find the input to output paths sensitized by each pattern. The fraction of random input patterns which sensitized a path is shown in Fig. 3 as the experimental detection probability against the levels in the path. The fan-ins in the circuit varied from 1 to 8, however, the average fan-in was found to be 2.25. Thus it is not surprising that a majority of the experimental points are between the theoretical curves corresponding to  $n = 2$  and  $n = 3$ . Next the probability  $P(L, M)$  was obtained for several

paths of the circuit by the Monte Carlo experiment. In this experiment, sets of  $M$  random input patterns were applied and the experimental estimate of  $P(L, M)$  was computed as the fraction of these sets which sensitized the path. This probability for a four-level path is shown in Fig. 5 by the stepped curve. For comparison, the theoretical curve for the four-level NAND tree model, as computed from (4) is also shown. Agreement between the two curves seems reasonable. Similar results were obtained for several other paths in the circuit.

#### IV. APPLICATION OF RESULTS

The random test generation method was applied to a number of printed circuit cards of the above type, containing up to about 500 lines. The flow chart of the method is shown in Fig. 6. When the computation was carried to a sufficiently large number of input patterns, complete test sets for detecting all the single stuck type faults were obtained in most cases. Undetected faults were found to be redundant. The number of input patterns required to generate a complete test set is shown as the experimental points in Fig. 4, against the maximum level in the circuit. The theoretical curves in this figure correctly predict the number of patterns required for test generation. It thus appears that the test generation can be practically completed by applying a number  $M$ , of input patterns, such that the probability  $P(L, M)$  for sensitizing the longest path in the network is close to unity.

The random test generation is based on the assumption that a complete test set can usually be obtained from random input patterns forming a subset of the  $2^N$  possible patterns, where  $N$  is the number of primary inputs. The theoretical results actually give the number of patterns in this subset (of course, in a probabilistic sense) which can be used in designing the test generation experiments.

#### V. CONCLUSION

In this paper the problem of testing logic networks by random input patterns is analyzed using a NAND tree model. The detection probability and the number of input patterns needed for fault detection are obtained as functions of the number of levels and the number of fan-ins. Generally, a network with  $N$  inputs can be completely tested by  $2^N$  input patterns. But as  $N$  increases,  $2^N$  becomes enormous. In practice, however, a number much lower than this is required for a complete test generation. It is interesting to note that for the model network of Fig. 1, the number of primary inputs is a function of the maximum level in the network. Thus if  $L$  is the maximum level, the number of primary input lines is  $n^L$  and there are  $2^{n^L}$  possible input patterns. The number  $M$ , of patterns required for sensitizing a path from an input to the output, even for a high probability (say,  $P(L, M) = 0.99$ ) for  $n = 2$ , remains much less than  $2^{2^L}$  as  $L$  is increased (Fig. 4). Thus the analysis indicates how the random method of test generation is useful for analyzing large logic circuits [1], [2], [4], [5].

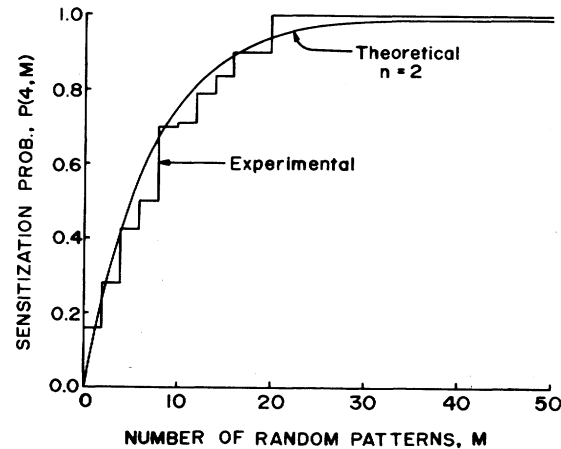


Fig. 5. Probability of path sensitization for a four-level path as a function of the number of input random patterns.

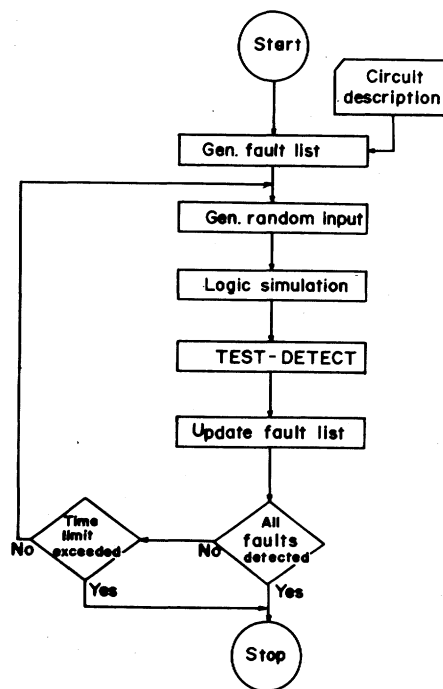


Fig. 6. Flow chart of random test generation.

Actual experimental results are compared with the theoretical ones and there is good agreement between them. The circuits used in the experiments mostly had {AND,OR,NOT} logic. A microscopic (gate-by-gate) examination of these circuits revealed that the logic had a free type structure, the fan-ins varied from 1 to 8 (average fan-in being slightly greater than 2) and multiple fan-outs were also present. Thus the general agreement of theory with the experiment indicates the validity of the NAND model for analyzing the practical circuits. From the results it is found that an analysis model can be selected with levels equal to the maximum level in the circuit and the constant fan-in in the model can be taken as the integer nearest to the average fan-in in the circuit. It may be

remembered that the theory is based upon the average characteristics of the network and may be applicable to large tree type combinational networks only. It may also be pointed out that for the circuits having storage elements, the detection probability will depend upon the order in which the input pattern sequence is arranged. This case is not covered by the present analysis.

#### REFERENCES

- [1] C. Tanaka, "Parallel simulation of digital systems," Dep. Comput. Sci., Univ. of Illinois, Urbana, Rep. 382 (ILLIAC IV Document 211), Apr. 1970.
- [2] V. Moreno, "A logic test generation system using a parallel simulator," Center for Advanced Computation, Univ. of Illinois, Urbana, ILLIAC IV Document 243, Feb. 1971.
- [3] M. A. Breuer, "A random and an algorithmic technique for fault detection test generation for sequential circuits," *IEEE Trans. Comput.* (Short Notes) (*Special Issue on Fault-Tolerant Computing*), vol. C-20, pp. 1364-1370, Nov. 1971.
- [4] J. C. L. G. Rault, "A graph theoretical and probabilistic approach to the fault detection of digital circuits," in *Dig. of Papers, Int. Symp. Fault-Tolerant Computing*, IEEE Comput. Soc. Pub. 71C-6-C, Mar. 1971, pp. 26-29.
- [5] V. D. Agrawal and P. Agrawal, "An automatic test generation system for Illiac IV logic boards," *IEEE Trans. Comput.* (Short Notes), vol. C-21, pp. 1015-1017, Sept. 1972.
- [6] D. B. Armstrong, "On finding a nearly minimal set of fault detection tests for combinational logic nets," *IEEE Trans. Electron. Comput.*, vol. EC-15, pp. 66-73, Feb. 1966.
- [7] F. F. Sellers, M. Y. Hsiao, and L. W. Bearnson, "Analyzing errors with the Boolean difference," *IEEE Trans. Comput.*, vol. C-17, pp. 676-683, July 1968.
- [8] J. P. Roth, W. G. Bouricius, and P. R. Schneider, "Programmed algorithms to compute tests to detect and distinguish between failures in logic circuits," *IEEE Trans. Electron. Comput.* (*Special Issue on Aerospace Computers*), vol. EC-16, pp. 567-580, Oct. 1967.
- [9] W. G. Bouricius *et al.*, "Algorithms for detection of faults in logic circuits," *IEEE Trans. Comput.* (*Special Issue on Fault-Tolerant Computing*), vol. C-20, pp. 1258-1264, Nov. 1971.
- [10] H. Y. Chang, E. G. Manning, and G. Metzger, *Fault Diagnosis of Digital Systems*. New York: Wiley-Interscience, 1970.
- [11] A. D. Friedman and P. R. Menon, *Fault Detection in Digital Circuits*. Englewood Cliffs, N. J.: Prentice-Hall, 1971.
- [12] M. A. Breuer, Ed., *Design Automation of Digital Systems*, vol. I. Englewood Cliffs, N. J.: Prentice-Hall, 1972.
- [13] A. K. Susskind, "Diagnostics for logic networks," *IEEE Spectrum*, vol. 10, pp. 40-47, Oct. 1973.
- [14] A. Mukhopadhyay, Ed., *Recent Developments in Switching Theory*. New York: Academic, 1971.
- [15] I. Berger and Z. Kohavi, "Fault detection in fanout-free combinational networks," *IEEE Trans. Comput.*, vol. C-22, pp. 908-914, Oct. 1973.

- [16] J. P. Hayes, "A NAND model for fault diagnosis in combinational networks," *IEEE Trans. Comput.*, vol. C-20, pp. 1496-1506, Dec. 1971.
- [17] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in *John von Neumann Collected Works*, vol. V., A. H. Taub, Ed. New York: Pergamon, 1963, pp. 329-378.



**Prathima Agrawal** (S'74) was born in Mysore, India. She received the B.E. and the M.E. degrees, (with distinction), both in electrical communication engineering, from the Indian Institute of Science, Bangalore, India, in 1964 and 1967, respectively, and the M.S. degree in electrical engineering from the University of Rochester, Rochester, N. Y., in 1968. At present, she is working for her Ph.D. at the University of Southern California, Los Angeles.

During 1964-65, she was a technical assistant in the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India. During 1967-68, she was a teaching assistant in electrical engineering at the University of Rochester, Rochester, N. Y. From 1968 to 1970 she worked on digital simulation and design diagnostics while she was employed on the Illiac IV Project, as research assistant and later as research programmer at the Department of Computer Science, University of Illinois, Urbana. She continued this work in 1970-71 when she was with the Automation

Technology, Inc., Champaign, Ill. During 1973-74, she worked at the Computer Center of the Indian Institute of Technology, New Delhi, India, on design automation and diagnosis of digital systems. Since 1974 she has been pursuing research in the same area at the University of Southern California, Los Angeles.



**Vishwani D. Agrawal** (S'68-M'70) was born in Allahabad, India, on February 7, 1943. He received the B.Sc. degree from the University of Allahabad, Allahabad, India, in 1960, the B.E. degree in telecommunication engineering with honours from the University of Roorkee, Roorkee, India, in 1964, the M.E. degree in electrical communication engineering with distinction from the Indian Institute of Science, Bangalore, India, in 1966, and the Ph.D. degree in electrical engineering

from the University of Illinois, Urbana, in 1971.

During 1966-67 he taught in the Department of Electrical Engineering at the Indian Institute of Technology, New Delhi, India. From 1967 to 1970 he was with the University of Illinois, Urbana, where he worked on phased arrays at the Antenna Laboratory and taught at the Department of Electrical Engineering. During 1970-71 he was with the Automation Technology, Inc., Champaign, Ill., working on the Illiac IV Project. During 1971-72, as a Senior Scientist at E. G. & G., Inc., Albuquerque, N. Mex., he worked on the design and evaluation of wide-band antennas. Since 1972, he has been an Assistant Professor at the School of Radar Studies, Indian Institute of Technology, New Delhi, India, where he teaches and conducts research on phased array radars.

## The Weighted Random Test-Pattern Generator

H. DANIEL SCHNURMANN, MEMBER, IEEE, ERIC LINDBLOOM, AND ROBERT G. CARPENTER

**Abstract**—A heuristic method for generating large-scale integration (LSI) test patterns is described. In particular, this paper presents a technique for generating statistically random sequences to test complex logic circuits. The algorithms used to obtain a set of tests by means of weighted logic signal variations are included. Several techniques for assigning these weights and for varying them are discussed on the basis of the primary algorithm. Also described is a means of obtaining a minimal number of test patterns. This approach has proved successful in obtaining fault-detecting patterns.

**Index Terms**—Fault-detecting patterns, heuristic algorithm, large-scale integration, testing, testing algorithms, test-pattern generator, weighted random patterns.

Manuscript received December 3, 1973; revised January 2, 1975. The authors are with the System Products Division, IBM Corporation, East Fishkill, N.Y. 12533.

### INTRODUCTION

**T**HE FIELD of test-pattern generation has grown in importance with the advent of large-scale integration (LSI). Numerous algorithms have been developed over the years with the object of arriving at a set of logic patterns that detect stuck faults in a circuit.

At present, most test-pattern generators (TPG's) are deterministic in nature and revolve around the concept of sensitizing a path between inputs and outputs. These TPG's can be divided into two categories: fault-oriented and path-oriented. Their mode of operation has been described in the literature [1]-[4]. However, deterministic TPG's are often unable to cope with the increasing com-