Correspondence

An Information Theoretic Approach to Digital Fault Testing

VISHWANI D. AGRAWAL

***Abstract*—The concepts of information theory are applied to the problem of testing digital circuits. By analyzing the information throughput of the circuit an expression for the probability of detecting a hardware fault is derived. Examples are given to illustrate an application of the present study in designing efficient pattern generators for testing.**

***Index Terms*—Logic testing, statistical communication theory, statistical testing, test generation.**

I. Introduction

The concepts of information theory have proved their usefulness in combatting noise-related errors of communication. Recently, attempts have been made to apply information theory to digital testing [1], [2]. In [1] information output is used as a testability measure and in [2] system testability is analyzed by defining an information measure on the circuit graph. The contribution of the present work, on the other hand, lies in the area of test generation. The effect of hardware faults on the flow of information through a digital circuit is studied. When a noise-free environment is assumed, the only errors in the output are caused by the hardware faults. Thus, the probability of error also represents the probability of detecting the fault. An expression for fault detection probability is derived which shows that this probability is maximized when the information output of the circuit is maximized. Our analysis, therefore, gives a justification for using the information output as a testability measure [1]. This result is also consistent with the *principle of maximum entropy* [3], [4], which states that one can attach maximum confidence to testing a system if the entropy at the output of the system is maximized during the test. Another argument that supports this result is Shannon's coding theorem [5]. According to this theorem, error suppression is impossible when the transmission rate exceeds the maximum possible rate of a communication channel. In practice, an error suppressing or coding scheme is difficult to devise for a transmission rate that is equal to the maximum rate.

The examples in Sections V and VI illustrate the construction of software pattern generators from functional descriptions of circuits that will optimize the fault detection probability. Such pattern generators are useful for testing permanent as well as intermittent faults. For simplicity, most of the results are derived for combinational circuits with a generalization to sequential circuits attempted in Section VIII.

II. Information Flow Through a Digital Circuit

Some authors [6]–[8] have treated computing machines from a point of view of information theory. But their analyses are directed toward estimating computational work done by the machine and as such do not give insight into the consequences of faults. Our treatment, therefore, is quite different.

Let us consider a digital circuit as a two-port device with an input and an output port. Each port contains a group of lines carrying information in the form of binary patterns. Using Shannon's formula [5], the information content on a group of k lines can be written as

$$H = -\sum_{i=1}^{2^k} p_i \log_2 p_i \text{ bits/pattern} \tag{1}$$

where p_i is the probability of occurrence of the ith pattern among the total of 2^k possible patterns. The information H is maximum when all possible patterns have an equal probability of occurrence, that is, $p_i = 2^{-k}$ and $H = k$bits per pattern.

As an example, consider a two-input AND gate. The maximum input information is 2 bits/pattern when the patterns 00, 01, 10, and 11 each have a probability 0.25. In this case, the output will be a 1 with a probability 0.25 and a 0 with a probability 0.75. Thus, the output information is

$$H_o = -0.25 \log_2 0.25 - 0.75 \log_2 0.75 = 0.811 \text{ bit/pattern.}$$

Notice that the circuit has an information loss of $2.0 - 0.811 = 1.189$ bits/pattern. This loss, however, can be reduced. Consider an information source producing 2-bit patterns such that each bit is 1, independently of the other, with a probability q. Now the output of the AND gate will be a 1 with probability q^2 and a 0, with probability $1 - q^2$. The output information will be maximized to 1 bit/pattern when $q^2 = 0.5$ or $q = 0.707$. In this case, the total input information is 1.745 bits/pattern and the loss is reduced to 0.745 bit/pattern. *Thus, the information throughput of a digital circuit is a function of the characteristics of the circuit and the characteristics of the source producing the input information.*

Logic circuits are decisionmaking devices. Using the input information, they make certain decisions and produce the result as the output information. Thus, the information at the output completely depends upon the input information. In general, some information is lost in these decisions. When such a loss occurs, the input information cannot be recovered from the output.[1] Only in the special case of an *information lossless network*, an inverse network is possible [10]. An example of information lossless network is an encoder for which the inverse network is the decoder.

Memory devices store information which can be used in the future. Consider a random access memory (RAM) of 2^x words of $k$bits each. The memory may have $x + k + 1$ input lines (x address lines + k data lines + 1 read-write line) and k output data lines. When the memory is used to store random data, each word contains $k$bits of information. Also, when the words are randomly accessed, the information supplied to the address lines during a read or write operation would be x bits. If we assume that the memory does a read or write with equal probability on every input pattern, then the average input information is $x + k/2 + 1$ bits/pattern and the average output information is $k/2$

Manuscript received September 9, 1980; revised January 15, 1981.
The author is with Bell Laboratories, Murray Hill, NJ 07974.

[1] For instance, a circuit can be designed to accept digitally represented values of temperature, pressure, and humidity and then make a binary decision, based upon certain risk/cost criteria, whether or not it will rain. Now just by knowing this decision it is impossible to recover the values of the input variables. In thermodynamics, a process in which the entropy increases is called irreversible; an entropy increase is analogous to information decrease since Shannon's information is just the negative of the thermodynamic entropy [9].

0018-9340/81/0800-0582$00.75 © 1981 IEEE

bits/pattern. This is because the read-write line carries one bit per pattern, while during half the patterns that write there is no output and during the other half that read there is no information given to the data input lines. Thus, the average information loss is $x + 1$ bits/pattern.

III. Information and Hardware Faults

In Shannon's theory of communication [5], errors are caused by noise. Noise can cause errors in digital circuits, but such errors are usually intermittent. Coding schemes are sometimes used to detect and correct these errors. Parity check and Hamming codes are examples of such codes. In most of our present discussion, however, we are concerned with a different class of errors. These errors are permanent (except for the intermittent errors considered in Section VII) and are caused by hardware faults.

The signals handled by digital circuits are binary and, therefore, their faults are, for practical purposes, modeled as stuck-at-0 and stuck-at-1 fault conditions. Consider a line (hardware connection) in a digital circuit which transmits information in the form of 1's and 0's. If the probabilities of a bit being 1 or 0 are q and $1 - q$, respectively, then the information carried by this line is

$$H(q) = -q \log_2 q - (1 - q) \log_2 (1 - q). \tag{2}$$

A stuck-at-1 fault on this line makes $q = 1$ and a stuck-at-0 fault results in $q = 0$. In either case, the information transmitted by the line becomes zero because $H(1) = H(0) = 0$. Thus, the presence of the fault will be indicated at the output of the line as a loss of information. A faulty line may be embedded in the circuit such that its output is not directly observable. In such cases the observable information at the output of the circuit may contain a small amount of information coming from the faulty line and the detection of the fault would become difficult.

Another complication arises due to the fan-outs. Suppose a line carrying H bits of information branches out into two lines. Now if one of the fan-outs has a fault and the information through it drops to zero, the same information is still available on the other fan-out line. In case of reconvergent fan-outs, the loss of H bits due to fault on one fan-out line may not manifest itself readily at the circuit output.

There is one more aspect of fault detection that must be discussed. If we have a sequence 0101101 passing through a line, the fault stuck-at-1 will convert it to 1111111 and the fault stuck-at-0 will convert it to 0000000. We say that our sequence is sensitive to both the faults. On the other hand, a sequence 1111111 is sensitive only to the stuck-at-0 fault. Thus, for any given fault there is only one sequence (all 1's for stuck-at-1 or all 0's for stuck-at-0) that is not altered by the fault. As we increase the length of the sequence we find that all 1 and all 0 sequences imply $q = 1$ and 0, respectively. In either case the information on the line as given by (2) is zero. In general, one is interested in detecting both stuck-at-1 and stuck-at-0 faults; we would, therefore, like to avoid the all 1's and all 0's sequences. *This means that for sensitization of hardware faults; in a circuit element, it is necessary that this element be transmitting a nonzero amount of information.*

IV. Probability of Detecting a Fault

Let us consider a circuit containing n input and m output lines. The circuit is connected to an information source (or pattern generator) supplying H_i bits/pattern on the n input lines. Also assume that the output information on the m output lines is H_o bits/pattern. From (1), $H_i \leq n$ and $H_o \leq m$. Now we consider the influence of a stuck-type of fault on a line inside the circuit. In order to study the amount of information flowing through the fault-site, let us consider a partition of the circuit as shown by the dotted line PP in Fig. 1. This partition cuts through k lines, including the one on which the fault is being considered, each carrying information from left to right. Information flow from left to right implies that the information carried through the partition once should not pass through it again.

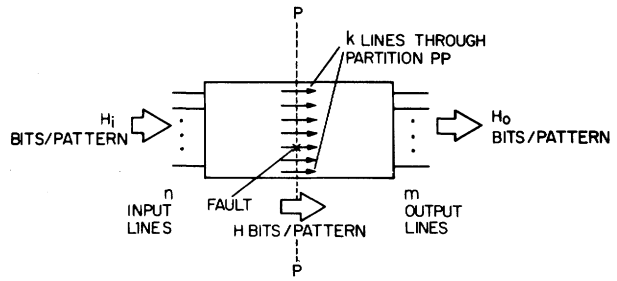

Fig. 1. Information flow through a circuit.

Let the total information flowing through the partition PP be H bits/pattern, where $H_i \geq H \geq H_o$. These bounds simply indicate that the H bits of information are found at some intermediate step during the processing of information by a circuit that does not contain any information sources. Consider now a sequence of T input patterns. For large T, according to McMillan's theorem[2] [11], there are approximately 2^{HT} high probability sequences that can occur at the partition PP, each with a probability 2^{-HT}. Similarly, at the primary output of the circuit, there are 2^{H_oT} high probability sequences, each having a probability 2^{-H_oT}. On an average, therefore, the same output sequence can be produced by $2^{(H-H_o)T}$ different sequences at the partition PP. This is shown in Fig. 2, where each black dot represents an output sequence and the circles represent pattern sequences at the partition PP. Next assume that the H bits through the partition are equally divided among the k lines, then each line in the partition would carry H/k bits/pattern. This means that the faulty line, in the absence of a fault, can have any one of the $2^{HT/k}$ equiprobable bit-strings of length T. The fault will transform any one of these bit-strings to either all 0's (stuck-at-0) or all 1's (stuck-at-1). Now consider a sequence of T patterns applied to the fault-free circuit. By examining the output sequence we know that any one of the $2^{(H-H_o)T}$ possible sequences may have occurred at the partition PP. The pattern sequence at PP in the faulty circuit will differ only in the bit-string at the faulty line. Fig. 3 gives an example of a pattern sequence as modified by the fault stuck-at-0; as shown, the same modified sequence can be obtained by any one of the several sequences. Only if the modified sequence produces an output sequence that is different from that produced in the fault-free circuit, will the fault be detected.

Fig. 4 shows a set of 2^{HT} sequences. If we consider a subset of sequences which differ in the bit-string on just one among k lines, then this subset will contain $2^{HT/k}$ sequences. Similarly, in any randomly chosen subset of n sequences, there will be $n^{1/k}$ sequences of the above type. Suppose the sequence marked X is the sequence as modified by the fault at PP. The fault-free circuit may have any of the $2^{HT/k}$ sequences shown as X, A, B, C, where A, B, C differ from X only in the faulty line bit-string. As shown in Fig. 4, if A or X was the sequence produced at PP by the input patterns, then the fault will not be detected. But, if B or C were produced at PP, the fault will be detected. Now there are $2^{(H-H_o)T}$ sequences that produce the same output as that produced by X. Of these, only $2^{(H-H_o)T/k}$ sequences belong to the subset of $2^{HT/k}$ sequences described above. Therefore, the probability of detecting the fault is

$$P(T) = 1 - \frac{2^{(H-H_o)T/k}}{2^{HT/k}} = 1 - 2^{-H_oT/k}. \tag{3}$$

[2] McMillan's theorem applies to sequences which are ergodic and stationary. A sequence is stationary if the statistical properties remain the same as one moves along the sequence. A stationary sequence is ergodic if an appreciable interpattern correlation exists at most over a finite number of patterns. For circuits that are combinational or have a relatively small number of storage elements, these conditions may be easily satisfied if the input patterns are uncorrelated. The sequence, however, may not be strictly ergodic stationary when it is designed to execute a function involving a series of linked operations. In the latter case the analysis of ergodic stationary model may still serve as an approximation. For a concise discussion of McMillan's theorem, see [12].

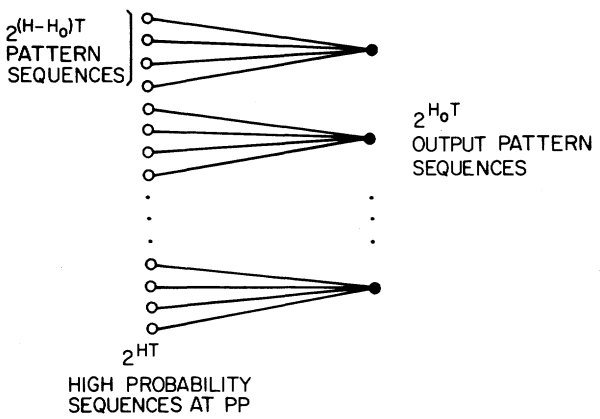

Fig. 2. Schematic representation of input-output relationship.

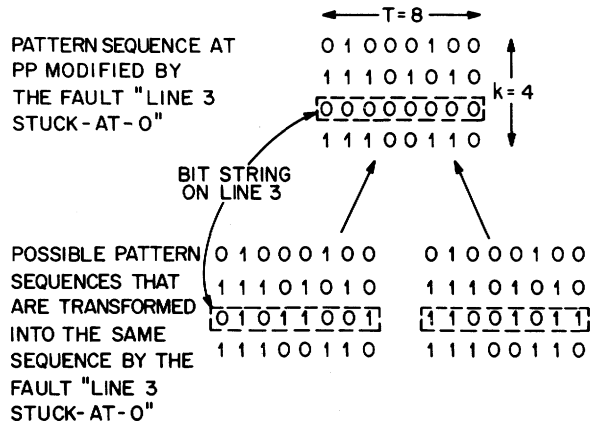

Fig. 3. Transformation of sequences of 8 patterns by a fault at a partition with 4 lines.

The above formula indicates that for a given length (T) of sequence the probability of detecting a fault would increase with output information H_o. This is equivalent to Jaynes' *maximum entropy principle* [3], [4] as used in reliability analysis. Also, for given T and H_o, the probability of detection will be smallest for those faults which lie on a partition having the largest number of lines (k). Such faults are usually encountered immediately after large fan-outs since fan-outs tend to increase the number of lines without increasing the total amount of information. Another assumption that was used in deriving (3) was that the information H bits/pattern was equally divided among k lines. This may not be true and, therefore, the probability of detection of faults will be lower on some lines and higher on others in the same partition. Equation (3) represents a probability computed on the basis of average information through each line. The condition can be somewhat met if one makes sure during the testing that the information flow over the whole circuit is as uniform as possible [13], [14].

V. Example

The use of the above formula will now be illustrated by a simple example of a 10-input AND gate, where the input patterns are generated randomly, setting each of the 10 input lines independently to 1 with probability q and to 0 with probability $1 - q$. Total input information is

$$H_i = 10[-q \log_2 q - (1 - q) \log_2 (1 - q)] \text{ bits/pattern.} \quad (4)$$

The output line will have a 1 only when all the inputs are 1. The probability of this event is q^{10}. Therefore, the output information is

$$H_o = -q^{10} \log_2 q^{10} - (1 - q^{10}) \log_2 (1 - q^{10}) \text{ bits/pattern.} \quad (5)$$

Fig. 5 shows a plot of H_i and H_o as a function of q. H_o attains a

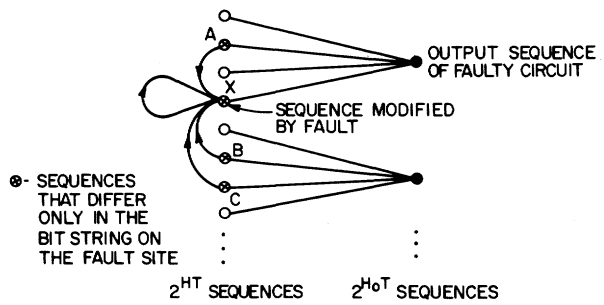

Fig. 4. Schematic representation of transformations of sequences by a fault into the same faulty sequence.

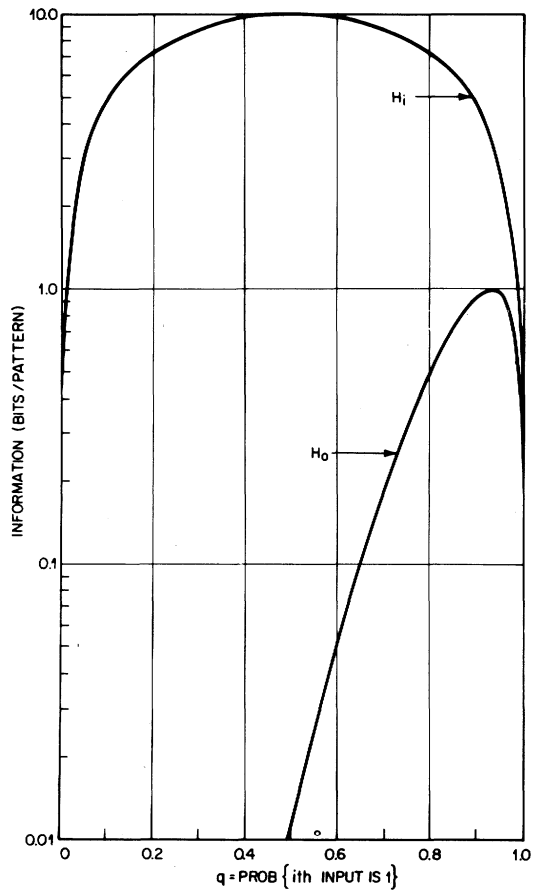

Fig. 5. Input and output information of a 10-input AND gate as a function of the probability (q) of 1's in the input patterns.

maximum when $q^{10} = 0.5$ or $q = 0.933$. One would thus expect the circuit to be tested most efficiently when the input patterns are generated with $q = 0.933$.

Input patterns were constructed by using a computer generated random number for each bit position. The random numbers were generated such that they were uniformly distributed in the interval [0.0, 1.0]. A bit was set to a 1 when the corresponding random number was less than or equal to q, otherwise, the bit was set to a 0. Enough random patterns were generated so that every stuck-at-0 and stuck-at-1 fault on all the lines was detected by at least one pattern. The procedure was repeated for various values of q and the final number of patterns in each case is shown as a point in Fig. 6. From (3) we can determine the number of patterns that will detect an input fault with high probability, say $P(T) = 0.99$. Thus

$$1 - 2^{-H_oT/k} = 0.99$$

or

$$T = 6.67k/H_o.$$

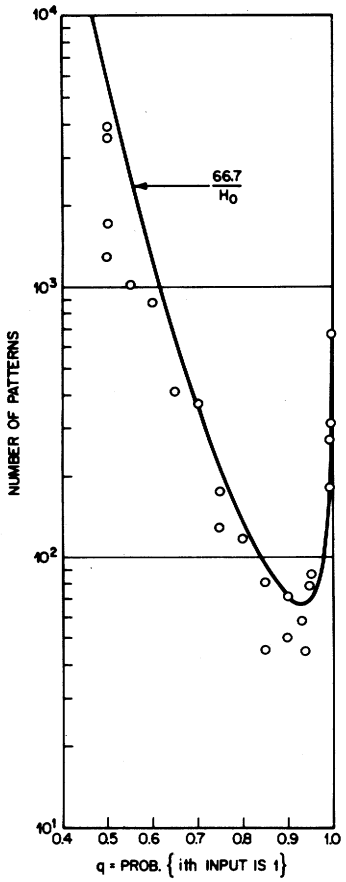

Fig. 6. Number of patterns required for detecting all stuck-type faults in a 10-input AND gate as a function of the probability q used in generating patterns. Points indicate results of test generation experiments and the curve is the computed number of patterns required for detecting an input fault with 99 percent probability.

The continuous curve in Fig. 6 gives the theoretical length (T) of sequence that will detect any input fault with probability 0.99. The plotted values of sequence length were obtained from the above equation by substituting H_o as a function of q from (5). The fact that most of the points lie below the curve should be viewed as a qualitative verification of the theoretical result.

VI. Design of a Functional Pattern Generator

The simple example of an AND gate discussed above illustrates how the probability of fault detection can be improved by suitably designing the pattern generator. We now consider a more complex example. The circuit used in this example is a 4-bit multiplier [15] shown in Fig. 7. This circuit has eight input lines (four for the multiplier and four for the multiplicand, each of which can be a binary integer between 0000 and 1111) and eight output lines. The possible outputs are binary representations of decimal integers between 0 and 225. Certain prime numbers such as $17, \cdots, 43, \cdots$, etc., and certain other numbers like 46 which cannot be factorized with both factors less than 15 cannot appear at the output. In fact, there are only 90 different integers which can occur at the output of this circuit. Therefore, the maximum possible information output is $H_o = \log_2 (90) = 6.49$ bits/pattern. Fig. 8 shows the flowchart of a pattern generator which maximizes the output information. An output integer Z is generated by using a random number function $U(0, 225)$ which produces any integer from 0 to 225 with equal probability.

The integer Z is then factorized to form k pairs of integer factors such that each factor is between 0 and 15. If no factors are possible (i.e., $k = 0$), meaning that either Z is a prime number greater than 15 or at least one factor must be greater than 15, then a new value for Z is obtained. Otherwise, any one factor-pair, $Z = X(J).Y(J)$, is selected randomly among the k available pairs. Binary representations of $X(J)$ and $Y(J)$ then form the input pattern.

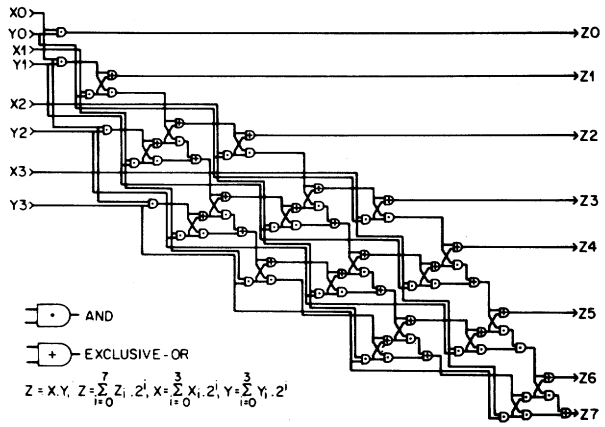

Fig. 7. Four-bit multiplier circuit, as it appeared in [15].

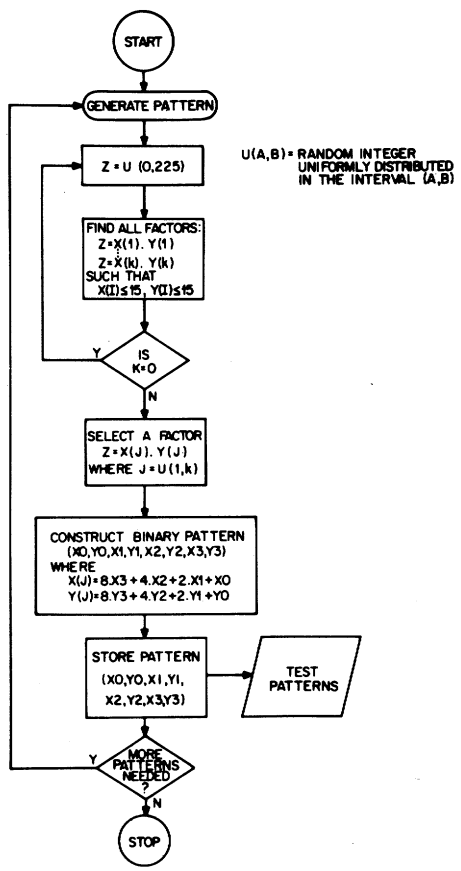

Fig. 8. Pattern generator for the four-bit multiplier. Input patterns produce each possible output with equal probability.

The patterns from the above pattern generator were used in a fault simulation of the multiplier circuit. All stuck-at faults were simulated and the cumulative percentage of detected faults is shown in Fig. 9 as a function of the number of patterns (the points in the plot correspond to the patterns that detected some new faults undetected thus far). In several independent runs the number of patterns for the detection of all faults always ranged near 40. For comparison, patterns with equiprobable 0 or 1 on each input line were evaluated in a similar manner and their result is also shown in Fig. 9.[3] In this case, complete fault detection required 100 or more patterns.

[3] Equiprobable 0 or 1 in patterns assure that each of the 256 possible input patterns will be selected with equal probability. In this case, the outputs will occur with unequal probability. For example, a 00000000 (or decimal 0) at the output will be about 15 times more frequent than 00011010 (or decimal 26). Thus, the output information will be less than 6.49 bits/pattern.

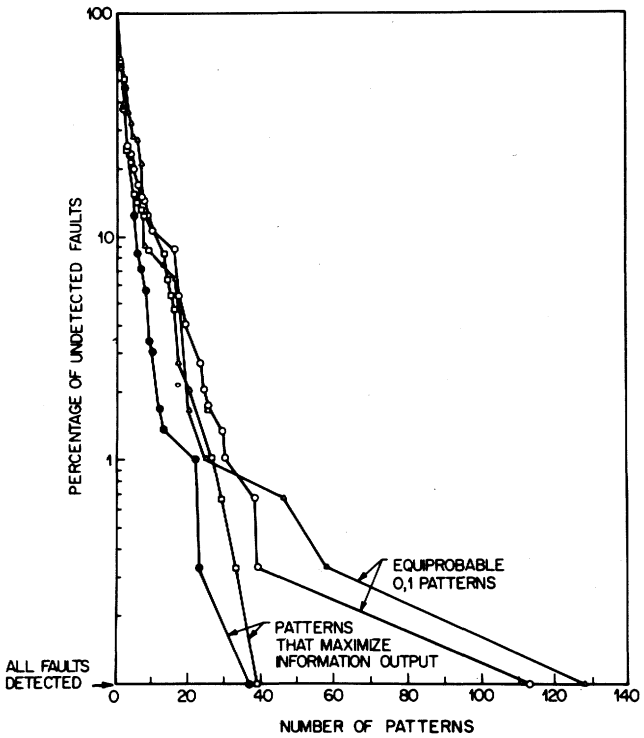

Fig. 9. Results of fault simulation of the four-bit multiplier circuit with patterns generated using the procedure of Fig. 8 and those with equally probable 0's and 1's.

It must be pointed out that a pattern generator of the type shown in Fig. 8, which maximizes the information output, uses only the function of the circuit and hence is not dependent upon its implementation. In practice, some implementations may be more difficult to test. For example, a circuit implementation with many reconvergent fan-outs and large fan-ins may be more difficult to test as compared to another implementation with fewer reconvergent fan-outs and smaller fan-ins. For any given implementation, however, our pattern generator is likely to be more effective against any other statistical pattern generator.

VII. Intermittent Faults

Intermittent faults, whenever active, are like the permanent faults. Because of their intermittent nature they are more difficult to detect. Often a test has to be applied repeatedly to test such faults [16]. In our model we will assume that an intermittent fault is present only p percent of the time and is absent otherwise. The influence of such faults is similar to the errors caused by noise. According to Shannon's coding theorem [5], the errors can be suppressed (or corrected) for a communication channel only if the rate of transmission does not exceed the channel capacity. The error suppression is achieved by making use of the redundancy afforded by the lower rate of transmission.

In a digital circuit the maximum information output is analogous to the channel capacity. When the input information is more than the output information (i.e., there is loss in the circuit), there is already redundancy in the circuit. Further lowering of output information rate will increase the redundancy resulting in greater error suppression. Thus, when a circuit transmits at the maximum possible rate, the error suppression is minimum. The above arguments point to the fact that our pattern generator which maximizes information output should also be effective in detecting the intermittent faults having a noise-like nature.

In order to study the performance of pattern generators in detecting intermittent faults, a Monte Carlo experiment was carried out using a fault simulator. In the multiplier circuit of Fig. 7 a fault was picked randomly from the list of all stuck-at-1 and stuck-at-0 faults. This fault was simulated using the patterns from the generator of Fig. 8.

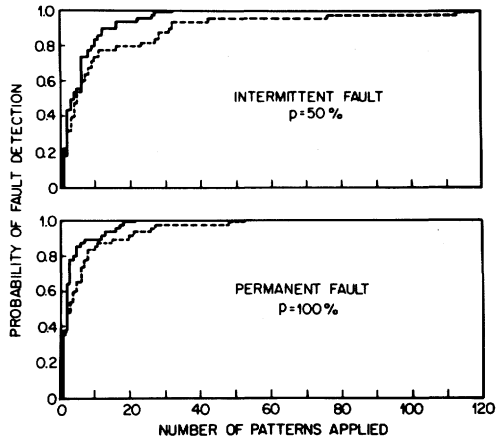

Fig. 10. Monte Carlo probability of detecting the intermittent faults that are present for half the time (p = 50 percent) and the permanent faults (p = 100 percent). Solid lines correspond to the pattern generator of Fig. 8, and the dotted lines to the patterns with equiprobable 0's and 1's.

The results of the fault simulation were examined only for p percent of the applied patterns. For other patterns, even if the fault was detectable it remained unreported, thus simulating the absence of the fault for $(1 - p)$ percent of patterns. The experiment was repeated for 50 sample faults and the result is shown in Fig. 10 by solid curves for p = 100 percent and 50 percent. The dotted curve was obtained in a similar experiment with input patterns having equiprobable zeros and ones. The advantage of increasing the information output is evident in both cases of p = 100 percent (permanent faults) and p = 50 percent (intermittent faults).

VIII. Sequential Circuits

A sequential circuit, in addition to the combinational elements, also contains memory elements. A portion of the information supplied at the primary inputs flows to the primary outputs, while some of this information gets stored in the memory elements. The stored information becomes available to the combinatorial part of the circuit when the next pattern is applied. Thus, the information flowing through the memory elements may arrive at the primary outputs after several patterns have been applied. In other words, a sequence of patterns may be required to produce an output sequence from a sequential circuit. For example, to produce an output from a microprocessor, one may execute an instruction requiring an ordered sequence of patterns at the input. Completely random patterns, on the other hand, are unlikely to execute an instruction and produce a meaningful output. Thus, the information output of a sequential circuit, when input patterns are completely random, could be very small and such patterns may not be effective in testing.

Sequential circuits normally contain feedback paths. In partitioning the circuit in the procedure of Section IV, it is necessary that the feedback lines should not pass through the partition [17]. This is because the feedback lines do not carry any new information that has not already passed through the partition. In fact, the feedback lines carry a part of this information to the memory elements for later use. This procedure does not exclude the faults on the feedback lines since these faults are identical to the faults at the output of the logic elements that produce the feedback signals.

The information flow on a per pattern basis for sequential circuits can be calculated as follows. Consider a circuit that is designed to perform n operations. Let us assume that the jth operation requires an input sequence of v_j patterns and can produce m_j distinct output sequences. The information output of this operation will be maximum when each of the m_j output sequences are made equiprobable. In this case the average information output of the jth operation will be

$$h_j = \frac{\log_2 m_j}{v_j} \quad \text{bits/pattern.} \tag{6}$$

Now if the probability of executing the jth operation is p_j, then the average information output of the circuit is given by

$$H_o = \sum_{j=1}^{n} p_j h_j \quad \text{bits/pattern.} \tag{7}$$

In order to generate patterns, first, p_j's should be assigned to the various operations such that the output information as given by (7) and (6) is maximized. The pattern generator would then proceed by selecting operations with the assigned probabilities and generating an input pattern sequence for the selected operation. An input pattern sequence is generated from the functional description of the circuit and data patterns are selected to make all possible outputs equiprobable.

IX. Conclusion

The problem of digital testing has been discussed from a new viewpoint. Although the analysis is based upon a model of a digital circuit where all signals are assumed to have certain statistical properties and the information is assumed to distribute uniformly among the lines, the results have qualitative importance. The main result, according to which a maximized information output implies highest probability of fault detection, is used in constructing pattern generators for testing. These pattern generators make use of only the functional description of the circuit and hence are independent of the actual hardware implementation.

Apart from the digital testing, two other applications of the analysis presented here may be suggested. Since the concepts of information theory are applicable to the digital as well as to the analog signals, the analysis could possibly be extended to the testing of analog circuits. A second application might be in software testing. Quite recently, the use of random test inputs in software testing has been reported [18]. In that work it was shown that the effectiveness of random data was strongly dependent on the interval from which the data were drawn. The authors concluded that good results could be obtained if problem-specific information was used in generating the random data. The procedure of maximizing the program output information while selecting test data should prove useful in software testing.

Acknowledgment

The author would like to acknowledge the contribution of E. Edelman in preparation of this manuscript.

References

[1] J. A. Dussault, "A testability measure," in *Dig. Semiconductor Test Symp.*, Cherry Hill, NJ, Oct.-Nov. 1978, pp. 113-116.
[2] P. Caspi, A. Mili, and C. Robach, "An information measure on nets—Application to the testability of digital systems," in *Information and Systems*, B. Dubuisson, Ed. New York: Pergamon, 1978, pp. 35-39.
[3] E. T. Jaynes, "New engineering application of information theory," in *Proc. 1st Symp. Eng. Appl. of Random Function Theory and Prob.*, J. L. Bogdanoff and F. Kozin, Eds. New York: Wiley, 1963, pp. 163-203.
[4] M. Tribus, "The use of the maximum entropy estimate in the estimation of reliability," in *Recent Developments in Information and Decision Processes*, R. E. Machol and P. Gray, Eds. New York: Macmillan, 1962, pp. 102-140.
[5] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379-423, July 1948.
[6] I. L. Lebow, "Communication in digital systems," in *Information Theory*, C. Cherry, Ed. London: Butterworths, 1961, pp. 99-108.
[7] L. Brillouin, *Science and Information Theory*, 2nd ed. New York: Academic, 1962, ch. 19.
[8] D. A. Bell, *Information Theory and Its Engineering Applications*, 3rd ed. New York: Pitman, 1962, ch. 9.
[9] L. Brillouin, *Scientific Uncertainty and Information*. New York: Academic, 1964, pp. 5-15.
[10] F. F. Sellers, Jr., M. Y. Hsiao, and L. W. Bearnson, *Error Detecting Logic for Digital Computers*. New York: McGraw-Hill, 1968, pp. 221-225.
[11] B. McMillan, "The basic theorems of information theory," *Ann. Math. Stat.*, vol. 24, pp. 196-219, 1953.
[12] L. S. Schwartz, *Principles of Coding, Filtering, and Information Theory*. Baltimore: Spartan, 1963, pp. 42-43.
[13] E. Pfaffelhuber, "Minimax information gain and minimum discrimination principle," in *Topics in Information Theory*, I. Csiszar and P. Elias, Eds. Amsterdam, The Netherlands: North-Holland, 1977, pp. 493-519.
[14] J. E. Shore and R. W. Johnson, "Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 26-37, Jan. 1980.
[15] N. Pippenger, "Complexity theory," *Sci. Amer.*, vol. 238, pp. 114-124, June 1978.
[16] P. K. Varshney, "On analytical modeling of intermittent faults in digital systems," *IEEE Trans. Comput.*, vol. C-28, pp. 786-791, Oct. 1979.
[17] V. D. Agrawal, "Information theory in digital testing—A new approach to functional test pattern generation," in *Proc. IEEE Int. Conf. Circuits and Comput.*, Port Chester, NY, Oct. 1-3, 1980, pp. 928-931.
[18] R. A. DeMillo, R. J. Lipton, and F. G. Sayward, "Hints on test data selection: Help for the practicing programmer," *Computer*, vol. 11, pp. 34-41, Apr. 1978.

Fault Diagnosis in a Boolean n Cube Array of Microprocessors

J. R. ARMSTRONG AND F. G. GRAY

***Abstract*—Fault-tolerant characteristics of a Boolean n cube array of microprocessors are analyzed. Connectivity properties of the network graph are used to show that n processor or link failures are required to isolate a processor. For processor failures the network is shown to be n (one step) diagnosable. A testing algorithm is presented which can diagnose up to n processor failures.**

***Index Terms*—Array, diagnosability, faults, fault tolerance, microprocessor, network.**

I. Introduction

In designing a fault-tolerant array of microprocessors, it is advantageous to use an interconnection network with more than one path between processor nodes. The Boolean n cube interconnect possesses this property and thus was studied for application in a fault tolerant computing system.

II. Connectivity and Diagnosability

Communication in the Boolean n cube network is based on message and broadcast algorithms such as those defined in [1]. The effect of failures is to disrupt these mechanisms and isolate processors from each other. Our analysis of the fault-tolerant characteristics of the Boolean n cube network is based on a graph theoretic model. In this model the processors are the graph nodes and the full duplex links interconnecting the processors are the graph edges. The major concern of this correspondence is with processor failure. A processor failure is assumed to remove a node from the graph and all edges connected to that node. The connectivity results given here also discuss link

Manuscript received November 23, 1979; revised January 28, 1981. This work was supported by the National Science Foundation under Grant MCS 75-06543 A01.

The authors are with the Department of Electrical Engineering, Virginia Polytechnic Institute and the State University of Virginia, Blacksburg, VA 24061.

0018-9340/81/0800-0587$00.75 © 1981 IEEE