REPORT R-354 MAY, 1967

COORDINATED SCIENCE LABORATORY

AD 653523

A PROCEDURE FOR RANKING DIAGNOSTIC TEST INPUTS

THEO J. POWELL

UNIVERSITY OF ILLINOIS - URBANA, ILLINOIS

This work was supported by the Joint Services Electronics Program (U.S. Army, U. S. Navy, and U.S. Air Force) under Contract DA-28 043 AMC 00073(E).

ACKNOWLEDGMENT

The author is indebted to Professor Gernot Metze whose suggestions, comments, and criticisms helped greatly during the research work and the writing of this thesis.

The author is also grateful to Ralph Marlett and Ross Schneider for their helpful discussions and to Edward Davidson for reading through the manuscript and making suggestions.

The support extended by the National Aeronautical and Space Administration is appreciated.

TABLE OF CONTENTS

1. INTRODUCTION

Today, digital computers are being built larger and with greater complexity. They are being used in applications connected with space flights, telephone systems, and plant control which require highly reliable operation. When a component in the machine becomes faulty, the time needed to detect and locate the faulty component should be minimized. Such a faulty component will be called the fault. Rapid procedures to locate and detect faults in computers are needed to aid maintenance personnel with maintenance of computers.

Let a machine be defined as the logical circuit of a computer. This thesis will be mainly concerned with combinational circuits. A machine which has a fault will give a set of outputs different from that given by the same machine with no faults, for some set of inputs. Most faults in a machine can be detected and specified by examining the outputs of the machine and by comparing them with the different output sets that a machine with the different possible faults produces. Several procedures for diagnosing to the individual faults have been developed [1, 3, 4, 5, 6]. The purpose of these procedures has been to find the specific fault in as few tests as possible and as quickly as possible. The assumptions that are made by many of these procedures, and that are also made in this thesis, are: 1) The machine will have at most one fault; 2) the possible faults that can occur in a machine are a logic element stuck at "1" or "0," and lines between the logic elements opened; and 3) the fault is well behaved, i. e., it is not intermittent.

Many computers today are constructed in small packages. Knowing which fault has occurred is not as important as knowing which package contains the fault; for when a package has a fault, the whole package is replaced. Chang [2] has taken this into consideration in his procedure for reducing redundant tests and thereby for selecting a near optimal set of tests for diagnosing a machine. His procedure is based on a pass or fail comparison of a machine containing a fault, to a fault-free machine. It is useful for a machine with a single line output but does not consider the additional information that a multi-output machine provides. A single line output is defined as an output with two output symbols, "1" and "0"; in this study each output symbol will be referred to as an output. This study considers the problem of selecting a near-optimal set of tests for diagnosing a multi-output machine with the goal of finding only the package wherein the fault lies. This procedure therefore removes redundant tests from a set of tests that can be applied to a multi-output machine.

One criterion for this near-optimal selection of tests has been to find a procedure that can be applied to large combinational logic circuits. Optimal procedures have been developed for small combinational circuits [1, 3], but if these procedures are applied to combinational circuits of reasonable size, the computations and the bookkeeping involved would be so voluminous that they would render the procedures impractical.

The method developed by this study can be applied to large combinational logic circuits since it examines only the package separation that each possible additional test gives when applied together with previously selected tests. The test that gives the greatest separation of packages is chosen and added to the list of selected tests. Although an optimal set of tests is not always chosen, at least a near-optimal set is selected. This method does not examine the package separations of several combinations of tests applied together which optimum procedures must do, and therefore the bookkeeping is greatly reduced, making this test procedure more practical.

In Chapter 2 various procedures for package diagnosis are discussed and a procedure for choosing a near-optimal set of tests is developed. Chapter 3 contains the development of a probabilistic formula to calculate probabilistic weights which are used for selecting the tests to be applied for diagnosis. An example of finding a set of tests with this procedure for one stage of an adder is also found in this chapter. In Chapter 4 a possible extension of the procedure to sequential machines is given along with other possible areas of research.

2. PACKAGE LEVEL DIAGNOSIS

In diagnosing failures of a machine, several procedures have been developed for selecting the set of tests that is applied to a machine. A test is defined as a possible input to a machine to determine the output of the machine. The output then is compared with the output of the fault-free machine. One main goal of computer diagnosis is to minimize the number of tests for complete diagnosis of a machine so as to shorten the time for diagnosis and reduce program space for a stored program data processor. Most test selection procedures have been developed on a fault level diagnosis basis [1, 3, 4, 5, 6]. Fault level diagnosis is defined as diagnosing a faulty circuit until the specific fault is known or it is known that no fault exists in the circuit.

The trend for machine construction today is to build several logic circuits in the same package for the reduction of size and the greater ease in replacing components. When a fault occurs, the whole package is replaced. For diagnosis of such a circuit only the package containing the fault need be found, and not the specific fault. Therefore diagnosis of such a machine should require fewer tests since less information is needed. Diagnosing a circuit until it is known which package contains the fault, or until it is known that the machine contains no faults, will be called diagnosis to the package level.

Few procedures have been developed for selecting tests for diagnosis to the package level which determines the faulty package but not an individual fault. Chang [2] derived a procedure for removing redundant tests for diagnosing a machine to the package level. His procedure is useful only if the machine essentially has just two outputs, namely, "1" and "0." However, many machines have more than two outputs and hence give more information than just a "1" or a "0" for each input test. Utilizing this extra information in selecting a sequence of tests improves the procedure for removing redundant tests.

In diagnosing to the package level, a very general procedure that requires only a knowledge of the truth table of each package would be desirable. This procedure would then, from the knowledge of the interconnections of the packages, diagnose the combinational circuit to the package level. For such a procedure, ways to detect dependency of the output of each package on its inputs would be needed. If no dependency can be determined, then all possible inputs for each package are needed to make sure that the package is fault-free. With dependency of the output on its inputs known, different sets of tests that would indicate whether a package has a fault would be needed, and from the interconnection of packages a diagnostic procedure would then be developed to diagnose to the package level.

On the other hand, if the circuit of each package is known then it is known immediately which faults can occur and what set of inputs will give a correct output for the package even if a fault is in the package and what set will detect the fault. The problem of separating packages is much less complex and the diagnostic procedure for the separation can be started at once. For these reasons it is assumed that the logic in each package is known.

2.1 Prime Implicant Diagnostic Procedure

Before considering the probability approach, it will be instructive to examine an optimal procedure.

Let p_i represent the i^{th} package and let the possible faults within the i^{th} package be $f_1^i, f_2^i, \ldots, f_k^i$. Although diagnosis to the package level does not require the identification of the separate faults within each package, each fault must be listed to insure that a specific fault f_j^i in package i has been separated from the faults of the k^{th} package, where all other faults in package i may have been separated from those of package k. Thus each fault must be separated by at least one test, in a set of tests, from all faults in the other packages in order to diagnose to the package level. This leads to a method for diagnosis with multiple outputs which is somewhat analogous to the prime implicant problem for logic design. After applying a test to a combinational circuit, different outputs will be given for the different faults, depending upon the fault. Note that a fault simply changes the fault-free machine to a different machine with different outputs. From these different outputs the following table (cf. figure 2) can be formed:

Down the side of the table the different possible tests are listed. Across the top of the table are listed all possible pairs of faults, such that the pair of faults is not from the same package. A + will be placed in the t_m, f_j^i/f_n^k table position if test t_m separates fault f_j^i from fault f_n^k, or in other words, if a machine with fault f_j^i produces a different output from a machine with fault f_n^k. When a machine with fault f_j^i produces the same output as a machine with fault f_n^k, then the table position t_m, f_j^i/f_n^k is left blank.

An example will clarify this procedure. Let a combinational circuit have packages p_1, p_2, and p_3 with two possible faults each. Let the five tests have four possible outputs for the various faults as shown in figure 1.

		t_1	t_2	t_3	t_4	t_5
P_1	f_1^1	00	11	00	01	00
	f_2^1	00	10	00	01	11
P_2	f_1^2	00	10	11	01	00
	f_2^2	01	00	01	10	00
P_3	f_1^3	01	00	01	10	11
	f_2^3	00	11	01	10	11

FR-1267

Figure 1

The fault test pattern table is shown in figure 2.

	f_1^1/f_1^2	f_1^1/f_2^2	f_1^1/f_1^3	f_1^1/f_2^3	f_2^1/f_1^2	f_2^1/f_2^2	f_2^1/f_1^3	f_2^1/f_2^3	f_1^2/f_1^3	f_1^2/f_2^3	f_2^2/f_1^3	f_2^2/f_2^3
t_1		+	+			+	+		+			+
t_2	+	+	+			+	+	+	+	+		+
t_3	+	+	+	+	+	+	+	+	+	+		
t_4		+	+	+		+	+	+	+	+		
t_5			+	+	+	+			+	+	+	+

FR-1268

Figure 2

Consider the pairs of possible faults, i. e., f_j^i/f_q^k, as elements of a set. Then let test t_m be considered as a set of these pairs and contain element f_j^i/f_q^k if and only if a plus sign is in the t_m, f_j^i/f_q^k entry of the table. Then test t_s will be a prime implicant if and only if any test t_r such that $t_r \supseteq t_s$ implies that $t_r = t_s$. Thus if a test is not a prime implicant it is a proper subset of a prime implicant test, while prime implicants are proper subsets of none of the other tests. In the example, $t_1 \subset t_2$ since there is a plus sign in a row of t_2 whenever there is a plus sign in the row t_1. Furthermore, since t_2 is not a proper subset of t_3, t_4, or t_5, it is a prime implicant. The first step in the diagnostic procedure is to find the prime implicant tests according to this new definition of a prime implicant.

The remaining step is the same as a step in the prime implicant problem for logic design which is that of finding a minimum cover of the fault pairs. A minimum cover is the minimum number of prime implicants whose union will contain all elements f_j^i/f_q^k where $i \neq k$. In the example, t_2 and t_5 or t_3 and t_5 are minimum covers. Thus one concludes that tests t_5 and either test t_2 or t_3 will diagnose the circuits to the package level.

This procedure is an optimum procedure for combinational circuits with multiple outputs since a minimum number of tests can be found by this minimum cover. However, one serious drawback in using this procedure for diagnosis is the vast amount of bookkeeping necessary to find a set of tests that will cover all faults. As an example, suppose that a combinational circuit has five modules with ten faults in each module; then the size of the fault pattern array similar to figure 2 will be the number of tests by 1000. Formulating this fault pattern table, finding the prime implicants, and finding the minimum cover would be very time consuming.

2. 2 Probability Weighting Procedure

A more practical procedure is the following which orders the possible tests by a probability procedure and finds a near-optimal set of diagnostic tests. For this procedure the tests and the outputs of the machines with different faults are listed in tables as in figure 1. The columns of the tables represent the possible faults that can occur. In the f_j^i, t_k entry is listed the output that the machine with fault f_j^i would give if test t_k were applied.

Similar to Chang's method for selecting tests [2], a weight for each test is given for the purpose of selecting the best test to be applied next to the machine for diagnosis. At least a near-optimal set, if not an optimal set, of tests is obtained by this approach while not requiring the exhaustive trials of all possible test combinations. This procedure weights the possible tests that can be applied to the machine according to probability. In calculating the probability it is assumed that each output for every test is equally likely and that there are enough outputs available so that a hypothetical test could completely diagnose the circuit. The weight is then the probability that a hypothetical test will diagnose the circuit after the test being weighted has been applied. The test with the highest weight is chosen and applied to the machine. Some separation of packages is obtained. Next, the tests are applied again to the machine to see how much more they will separate the remaining packages in conjunction with the test already selected. The probability that another hypothetical test will separate all remaining packages not separated by the test selected and the test being weighted taken together, is the new weight. A second test with the highest weight is added to the selected test set, and the process is continued. Each time a new weight is found for the tests not selected. This weight is the probability that a hypothetical test will separate the remaining packages not separated by the test being weighted and the selected tests taken together. After all packages have been separated the procedure is terminated, and the selected test set will diagnose the machine to the package level.

2.3 Assumed Number of Outputs

To find the weight of each test, it is necessary to see how many packages would still need to be separated if the test to be weighted were applied after previously selected tests. If for a given test t_m, a fault f_j^i in a machine gives a different output from fault f_q^k, then f_j^i is separated from fault f_q^k. Of course the interest is not in separating faults but packages, and so if all faults of a package have been separated from all other faults in the other packages, then this package has been separated. Note also that if it is important to determine whether the machine is fault-free, then the fault-free machine can be considered as a fault of a fictitious package denoted by f_0^0 whose outputs for the tests are the same as the machine's truth table. This will assure that the fault-free machine is separated from all other packages with faults.

In calculating the weights, it is necessary to determine how many possible outputs need be assumed. The weight of a test is the probability that a hypothetical test will separate all remaining packages not separated by the tests already selected and the test being weighted taken together. A sufficient number of outputs needs to be assumed for this hypothetical test in order for the probability not to be zero. The number of assumed outputs may then be more or less than the number of actual outputs of the machine. The number assumed is taken as the maximum number of packages that have not been separated after applying the tests selected for diagnosis along with any one of the tests being weighted. The probability is under the assumption that each output for every fault in the next applied test is equally likely to occur. Finding the probability is then simply a process of counting the number of ways that outputs can occur, such that the remaining undiagnosed circuit will be diagnosed to the package level. This number is divided by the total number of possible combinations of outputs to give what will be called the probabilistic weight, denoted W_p. In this procedure of diagnosis the probabilities are the weights, and it has been decided that a weight of zero is not to be given to any test since this would eliminate a comparison of all tests weighted zero. Therefore by using the maximum number of packages left to be separated as the assumed number of outputs, none of the weights will be zero.

Let the probabilities of the various tests order the tests and call this the test order assuming a number of outputs. This test ordering is not necessary to choose the test with the highest weight, but it has been introduced to help explain some properties of this procedure and to make comparisons. The test with the highest probabilistic weight will have order designated 1 and will be the test to be applied next for diagnosis.

Assuming more outputs than the machine actually has, is equivalent to putting the outputs of several tests together so that the machine appears to have more outputs. For example, if only two outputs are possible then four possible combinations of outputs, i. e., 2^2 can be obtained by putting the outputs of two different tests together. If the number of outputs assumed is equal to the machine's actual number of outputs raised to a power r, then the weight is also the probability of diagnosing the machine with r tests. Of course the assumption of all outputs having equal probability for all faults is the same.

Assuming more outputs may rank one test higher than another, whereas for fewer assumed outputs the tests may be ranked equally. An example of this is seen in figure 3.

Both weights are equal, assuming two outputs; however, assuming three outputs gives test t_2 a higher weight. For test t_2 and with three outputs, different combinations of two outputs among the three faults in package p_2, with the third output for fault f_2^1, can separate the packages, whereas with two outputs all three of the faults in package p_2 had to have the same output. There is less than half the number of combinations for two outputs among two faults than among three faults, and so with three assumed outputs, test t_2 has the higher weight.

		t_1	t_2	
P_1	f_1^1	0	1	
	f_2^1	0	0	
P_2	f_1^2	0	0	
	f_2^2	0	0	
	f_3^2	1	0	
W_p		0.125	0.125	Assuming 2 outputs
		0.222	0.296	Assuming 3 outputs

(W_p = Probabilistic weight) FR-1269

Figure 3

There are cases where assuming more outputs will switch the order of tests that are close together in a test set, but usually the test order will remain the same. Tests that are weighted higher will often leave fewer total faults or fewer packages left to separate. Increasing the number of possible outputs will increase all the probabilistic weights. Usually the tests will not change orders, however, since a fewer number of faults or a fewer number of packages will still have more combinations of outputs that will separate them.

2.4 Example with Probabilistic Weights

In order to make the procedure clearer, consider the diagnosis of the following example that Chang used in his paper [2]. His example has only two outputs, either 1 or 0; however, the multi-output probability method is also applicable to this case since a single line output is a trivial multi-output.

The set of tests that Chang used are shown in figure 4.

The test patterns can be written differently to see more clearly which packages still need to be separated after the first test has been applied if the first test is t_i for i = 1, ..., 8. This is done in figure 5(a) and (b).

In figure 5(b) it is seen that t_1 has a "0" in each package. The test t_1 does give some information for diagnosis since it separates f_1^1, f_2^3, and f_1^5 from the other faults, and hence the probabilistic weight should not be zero. An assumption of at least five outputs is required in order that the probability of the remaining packages being separated by a hypothetical test is not zero after applying test t_1. Let these five outputs be called a, b, c, d, and e. A test pattern that will completely separate the packages that have output "0" in test t_1 must have the same output for all faults in the package, and this output must not occur in any of the other packages. After referring to figure 5(b) one can see that one such possible output pattern that would separate these packages is:

		t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
P_1	f_1^1	1	0	0	1	0	0	0	0
	f_2^1	0	0	0	1	0	0	1	1
P_2	f_1^2	0	0	1	1	0	1	0	0
	f_2^2	0	1	0	1	0	0	1	1
	f_3^2	0	1	0	0	0	0	1	0
P_3	f_1^3	0	0	0	1	1	0	1	0
	f_2^3	1	1	0	0	0	1	0	1
P_4	f_1^4	0	0	0	1	1	0	1	0
	f_2^4	0	1	0	1	1	0	0	0
	f_3^4	0	1	0	0	1	0	0	1
P_5	f_1^5	1	0	1	0	1	1	0	0
	f_2^5	0	1	1	0	1	1	0	0
Chang's Weight	=	24	30	25	30	35	29	29	26
Chang's Ordering of Tests	=	8	2	7	2	1	4	4	6
Probabilistic Weight	=	0.0029	0.0177	0.0275	0.0472	0.4261	0.0531	0.0118	0.0059×10^{-2}
Probability Ordering of Tests	=	8	5	4	3	1	2	6	7

FR-1270

Figure 4

		t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
P_1	f_1^1	1			1				
	f_2^1				1			1	1
P_2	f_1^2			1	1		1		
	f_2^2		1		1			1	1
	f_3^2		1					1	
P_3	f_1^3				1	1		1	
	f_2^3	1	1				1		1
P_4	f_1^4				1	1		1	
	f_2^4		1		1	1			
	f_3^4		1			1			1
P_5	f_1^5	1		1		1	1		
	f_2^5		1	1		1	1		

FR-1272

Figure 5a

		t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
P_1	f_1^1		0	0		0	0	0	0
	f_2^1	0	0	0		0	0		
P_2	f_1^2	0	0			0		0	0
	f_2^2	0		0		0	0		
	f_3^2	0		0	0	0	0		0
P_3	f_1^3	0	0	0			0		0
	f_2^3			0	0	0		0	
P_4	f_1^4	0	0	0			0		0
	f_2^4	0		0			0	0	0
	f_3^4	0		0	0		0	0	
P_5	f_1^5		0		0			0	0
	f_2^5	0			0			0	0

Figure 5b

FR-1273

P_1	f_2^1	a
P_2	f_1^2	b
	f_2^2	b
	f_3^2	b
P_3	f_1^3	c
P_4	f_1^4	d
	f_2^4	d
	f_3^4	d
P_5	f_2^5	e

FR-1271

There are 5! ways for five outputs to completely separate the packages. Also there are 5^9 ways for five outputs to occur since there are nine faults. Therefore the probability of completely separating these remaining faults is $5!/5^9$. The next test to separate all packages after applying test t_1 must also separate the faults f_1^1, f_2^3, and f_1^5. There are 5(4)(3) ways for five outputs to separate these faults out of a total 5^3 possible test patterns. The probability for completely diagnosing these faults is $60/5^3$.

No dependency exists between the event of separating the 1's in figure 5(a) and the event of separating the 0's in figure 5(b); and so the probabilistic weight W_p is simply the product of the probabilities of these two independent events. The independence results from the assumption that every output is equally likely for every fault.

For test t_1 the probabilistic weight is calculated:

$$W_p = (5!/5^9)(60/5^3) = 7200/5^{12} = 0.0029 \times 10^{-2}$$

By similar procedures the other probabilistic weights are found and recorded in figure 4 along with the test order.

Test t_5 has the highest weight and is therefore chosen as the test to apply first in diagnosis. After applying test t_5 the remaining test patterns and packages to be separated are shown in figure 6. Figure 6 could be split up again as figure 4 was in figure 5(a) and (b); however, this is not necessary to calculate the probabilities. Since there are at most three packages left to be separated from each other, the maximum number of outputs needed is equal to or less than three. Test t_1 does not separate packages p_3, p_4, and p_5 completely, and thus at least three outputs need to be assumed for calculating the weights.

Let us choose t_6 as an example for calculating the probability. After tests t_5 and t_6 have been applied to the circuit it can be seen that package p_1 with faults f_1^1 and f_2^1 and package p_2 with faults f_2^2 and f_3^2 need to be separated. With three outputs there are 18 ways that these packages can be separated. Since there are four faults, the probability of separating p_1 from p_2 is $18/3^4$. The probability of separating p_2 with fault f_1^2 from p_3 with fault f_2^3 is 2/3 under the same conditions. It is noted that package p_5 is completely separated, and so the probability of separating

		t_1	t_2	t_3	t_4	t_6	t_7	t_8
P_1	f_1^1	1	0	0	1	0	0	0
	f_2^1	0	0	0	1	0	1	1
P_2	f_1^2	0	0	1	1	1	0	0
	f_2^2	0	1	0	1	0	1	1
	f_3^2	0	1	0	0	0	1	0
P_3	f_2^3	1	1	0	0	1	0	1
P_3	f_1^3	0	0	0	1	0	1	0
P_4	f_1^4	0	0	0	1	0	1	0
	f_2^4	0	1	0	1	0	0	0
	f_3^4	0	1	0	0	0	0	1
P_5	f_1^5	1	0	1	0	1	0	0
	f_2^5	0	1	1	0	1	0	0
W_p	=	0.0049	0.0195	0.0073	0.0293	0.0439	0.0146	0.0012
Test Ordering by W_p		6	3	5	2	1	4	7
Chang's Ordering		5	2	5	2	1	4	7

FR-1274

Figure 6

		t_1	t_2	t_3	t_4	t_7	t_8
P_1	f_1^1	1	0	0	1	0	0
	f_2^1	0	0	0	1	1	1
P_2	f_2^2	0	1	0	1	1	1
	f_3^2	0	1	0	0	1	0
P_2	f_1^2	0	0	1	1	0	0
P_3	f_2^3	1	1	0	0	0	1
P_3	f_1^3	0	0	0	1	1	0
P_4	f_1^4	0	0	0	1	1	0
	f_2^4	0	1	0	1	0	0
	f_2^4	0	1	0	0	0	1
W_p =		0.0313	0.5000	0.0156	0.0625	0.0625	0.0625
Test Ordering by W_p		5	1	6	2	2	2
Chang's Ordering		5	1	6	4	2	3

FR-1275

Figure 7

it is 1.0 while the probability of separating p_3 from p_4 is $24/3^4$. Multiplying these independent probabilities together will give the probabilistic weight of the test.

$$W_p = (18/3^4)(2/3)(24/3^4)(1.0) = 0.0439$$

Other probabilistic weights are recorded in figure 6.

Since t_6 has the highest weight, it is applied and produces the test patterns as shown in figure 7. Package p_5 is omitted since it has been separated. By the same procedure the probabilistic weights are calculated and are recorded in figure 7. After applying test t_2 which has the highest weight, the only remaining packages to be separated are p_3 with fault f_1^3 and p_4 with fault f_1^4. All other packages have been separated, and the test patterns are shown in figure 8.

		t_1	t_2	t_4	t_7	t_8
P_3	f_1^3	0	0	1	1	0
P_4	f_1^4	0	0	1	1	0
W_p	=	0.500	0.500	0.500	0.500	0.500
Test Orders		1	1	1	1	1

FR-1276

Figure 8

All probabilistic weights are the same as the weight of test t_2 and with the same number of assumed outputs. Therefore there is no further separation obtained by applying any of the remaining tests which indicates that faults f_1^3 and f_1^4 are indistinguishable. The diagnosis process then is terminated with the application of tests t_5, t_6, and t_2.

2.5 Comparison

It is instructive to compare the similarities and differences between Chang's ordering of the tests and the probabilistic ordering. Each time the tests were weighted in this example, the test with the highest weight by Chang's procedure also had the highest probabilistic weight. Therefore the test selected to be added next to the diagnostic set of tests was the same by both procedures. Since there are five packages to be separated there must be at least three tests for separating all the packages, and if one examines all other possible combinations of three tests, he will see that tests t_5, t_6, and t_2 are the best combination for diagnosis. In both procedures the best possible tests were selected.

After ordering the tests in figure 4 it is noted that the probabilistic weights ordered test t_6 as second while Chang's procedure ordered test t_2 as second and test t_6 as fourth. By examining these two tests one sees that test t_2 does not separate any of the packages from the rest. Every package after applying test t_2 has a zero for at least one fault, and if this test were selected, a minimum of three additional tests would be required to separate all packages. (Since two of these faults in different packages have identical test signatures, i. e., the same outputs for all tests, they are indistinguishable; however, this is assumed not known until the end of the diagnosis. If they could be separated, at least three more tests would be required.) On the other hand, test t_6 separates package p_5 completely from packages p_4 and p_1. There remain four packages to be diagnosed that have 0's recorded under test t_6 and three packages that have 1's recorded. Thus there is a possibility that the circuit could be completely diagnosed with just two remaining tests if test t_6 is used. (Test t_5 has been ignored for this comparison.)

For this specific example with test t_5 omitted, the same additional tests are required whether test t_2 is chosen or test t_6, but the comparison indicates the reason for the difference in ordering. Chang's method weights slightly more the splitting of the remaining faults to be diagnosed into two equal parts while the probabilistic weight simply selects the test with the highest probability of complete diagnosis with a certain number of additional tests. Of course in many cases the test which splits the remaining faults into equal separations is the test with the highest probabilistic weight as was seen in the selection of the best tests by both methods in Chang's example.

Examining a select set of test patterns will show more clearly some differences of ordering tests between the probabilistic weights and Chang's weights. This set will also indicate one weakness of both procedures.

Consider the set of tests and their weights as shown in figure 9. Two sets of tests will completely separate all faults; either tests t_1, t_6, t_7, and t_8 or tests t_2, t_3, t_4, and t_5. It can be seen that Chang's procedure weights tests t_1, t_6, and t_8 of the first set highest while the probability weighting procedure weights tests t_2 and t_5 of the second set highest. Notice that every test in the first set of tests except one has the same number of 0's and 1's, while every test in the second set separates one package each. Chang's procedure weights tests t_1, t_6, or t_8 highest since the 0's and 1's are divided more evenly in them. The probability weighting procedure weights tests t_2 and t_5 highest since there are more possible combinations with five outputs that would separate the remaining four packages than there are combinations which would separate the five packages. After applying either test t_2 or t_5 there are output patterns that would completely separate the remaining packages with two tests together, whereas a minimum of three tests is required for complete diagnosis after test t_1, t_6, or t_8. Admittedly for this example, two tests that would separate the remaining packages after applying t_2 or t_5 would have the highest weight of the tests listed and would be chosen first by both methods; however, this is not always the case for sets of packages and tests. The probability weighting

		t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
P_1	f_1^1	0	0	1	1	1	0	1	1
	f_2^1	1	0	1	1	1	1	0	0
	f_3^1	0	0	1	1	1	0	1	1
P_2	f_1^2	1	1	0	1	1	0	1	0
	f_2^2	0	1	0	1	1	1	0	1
P_3	f_1^3	1	1	1	0	1	0	0	0
	f_2^3	0	1	1	0	1	1	1	1
P_4	f_1^4	1	1	1	1	0	1	1	0
	f_2^4	0	1	1	1	0	0	0	1
	f_3^4	1	1	1	1	0	1	1	0
P_5	f_1^5	0	1	1	1	1	0	0	0
	f_2^5	1	1	1	1	1	1	1	1
Chang's Weight		29	27	20	20	27	29	28	29
Chang's Test Order		1	5	7	7	5	1	4	1
W_p =		0.0059	0.0430	0.0111	0.0111	0.0430	0.0059	0.0059	0.0059
W_p Test Order		5	1	3	3	1	5	5	5

FR-1277

Figure 9

procedure chooses the test that at least makes a shorter diagnosis possible.

If eight outputs are assumed in the probability weighting procedure, it turns out that test t_1 has a higher probabilistic weight than t_2. Thus the order switches, indicating that with an additional three tests (or 2^3 outputs) there exist more combinations that will separate the remaining packages for test t_1 than for test t_2. By assuming three additional tests, Chang's ordering and the probabilistic ordering are the same for tests t_1 and t_2. However, this approach ignores the possibility of diagnosis with fewer tests.

This points out one of the reasons for choosing the smallest possible number of assumed outputs that will still give all tests a weight other than zero. By choosing the smallest number of outputs, the probabilities correspond more closely to the minimum number of tests that will diagnose the circuit. Another reason for choosing the smallest possible number of outputs is to make the computations of the weights easier.

Neither Chang's procedure nor the probability weighting procedure always chooses the optimum test sequence for all cases. This can be seen if the only tests available for diagnosing the packages in figure 9 are tests t_1, t_2, t_3, t_4, and t_5. Chang's procedure would pick test t_1 first and then the other four tests. Test t_1 is not needed since the other four tests are sufficient to diagnose the circuit to the package level. In the same way, if the only available tests are tests t_1, t_2, t_6, t_7, and t_8, then the probability weighting procedure would pick test t_2 first and then the other four tests. Test t_2 in this case contributes nothing to the diagnosis to the package level since the other four tests are necessary and separate the packages themselves.

As was pointed out in the introduction, optimality was traded for the process of choosing the next best test without looking at all possible combinations of tests. A glance at the two cases above yields the optimum set of tests because of the simplicity of this problem and because the eye can see all possible combinations of tests. But of course in larger combinational logic circuits, considering all possible combinations of tests is an enormous task.

3. PROBABILISTIC WEIGHT FORMULA

3.1 Weight Formula Derivation

Calculating the probabilities can be a fairly tedious job if done by hand. In order to facilitate the matter, the following iterative formula has been developed to permit the probabilities to be calculated by computer. The following recursive formula calculates the number of possible ways of assigning NT possible outputs to N packages.

Define:

$$NX\,(I_1, I_2, \ldots, I_N) = 1$$

and

$$NX\,(I_1, I_2, \ldots, I_k, 0, 0, \ldots, 0) =$$

$$\sum_{I_{K+1}=1}^{T} \binom{T+N-(K+1)}{I_{K+1}} \left[\sum_{J_{K+1}=1}^{I_{K+1}} (-1)^{J_{K+1}-1} \binom{I_{K+1}}{J_{K+1}-1} (I_{K+1} - J_{K+1} + 1)^{NF_{K+1}} \right] NX(I_1, I_2, \cdots, I_{K+1}, 0, \cdots, 0)$$

FR-1303

$T = NT - N - I_1 - I_2 - \ldots - I_k + k + 1$

NT = Total number of outputs assumed

N = Number of packages to be separated

NF_{k+1} = Number of faults that are in the $k+1^{th}$ package

Each I_j represents the number of outputs that appear in the j^{th} package, for $j = 1, 2, \ldots, k+1$.

$NX\,(I_1, I_2, \ldots, I_k, 0, \ldots, 0)$ is the number of ways that $T + N - (k+1)$ outputs will separate all remaining packages indexed greater than k, with I_j outputs in the j^{th} package for $j = 1, \ldots, k$.

The number of possible combinations of NT outputs that will completely separate all packages is given by $NX(0, 0, \ldots, 0)$. The probabilistic weight is given by:

$$W_p = \frac{NX(0, 0, \ldots 0)}{NT^{NFT}} ;$$

where $NFT = \sum_{k=1}^{N} NF_k$ (the total number of faults).

In order to verify that this formula calculates the probabilistic weight as desired, consider the individual terms. The maximum number of outputs T that may appear in a package and still leave a sufficient number of outputs to cover the remaining packages is the total number of assumed outputs NT less the number of outputs assigned already, and less at least one output for each of the remaining packages. Therefore $T = NT - I_1 - I_2 - \ldots - I_k - (N - (k+1))$. The minimum number required is 1. The term

$$\binom{T + N - (k+1)}{I_{k+1}} \qquad (1)$$

is the number of ways that I_{k+1} outputs of the total number of outputs NT less the previously used outputs can be selected. These I_{k+1} outputs are then applied to the faults of package k+1.

The number of ways that these I_{k+1} outputs can be assigned to NF_{k+1}

faults is given by:

$$\sum_{J_{k+1}=1}^{I_{k+1}} (-1)^{J_{k+1}-1} \binom{I_{k+1}}{J_{k+1}-1} (I_{k+1} - J_{k+1} + 1)^{NF_{k+1}} \qquad (2)$$

To verify this, consider I outputs labeled $x_1, x_2, x_3, x_4, \ldots, x_I$. There are $(I)^{NF}$ ways that these outputs can be arranged where NF is the number of faults in this package. But we seek only those arrangements that have used all I outputs; we do not want to include those combinations with fewer than I outputs. There are $\binom{I}{1}(I - 1)^{NF}$ combinations that have (I - 1) outputs that have been included in the number $(I)^{NF}$, and so we must subtract $\binom{I}{1}(I - 1)^{NF}$ from $(I)^{NF}$. Next we must consider how many combinations of (I - 2) we have so far. The term $(I)^{NF}$ has $\binom{I}{2}(I - 2)^{NF}$ too many. To see how many of these have already been subtracted in the $\binom{I}{1}$ $(I - 1)^{NF}$ term consider the matrix of outputs:

$$\begin{matrix} - & x_2 & x_3 & x_4 & \cdots & x_I \\ x_1 & - & x_3 & x_4 & \cdots & x_I \\ x_1 & x_2 & - & x_4 & \cdots & x_I \\ \cdot & & & & & \\ \cdot & & & & & \\ \cdot & & & & & \\ x_1 & x_2 & x_3 & x_4 & \cdots & - \end{matrix}$$

The dashes indicate the output missing in the listing of the (I-1) combinations.

Notice that there are $\binom{I}{2}$ ways to choose any two outputs omitted and that there are $\binom{2}{1}$ rows of the matrix that can have the same remaining outputs with one more output omitted. For example, the first two rows omitting x_1 in the second row and x_2 in the first row have outputs x_3, x_4, ..., x_I. Since all other outputs have an x_j, deleted for $3 \leq j \leq I$, all other rows with one more output deleted will have either x_1 or x_2 or both. Since a row of the array represents one of the $\binom{I}{1}$ terms of (I - 1) outputs, the number of combinations of $(I - 2)^{NF}$ has been subtracted twice, once too many, and therefore $\binom{I}{2}(I - 2)^{NF}$ must be added to $(I)^{NF} - \binom{I}{1}(I-1)^{NF}$.

Consider now the number of (I - k) outputs. These combinations also must be removed from the total number of combinations since only I output combinations are wanted. There are $\binom{k}{0}\binom{I}{k}(I - k)^{NF}$ combinations in the $\binom{k}{0}(I)^{NF}$ term; $\binom{k}{1}\binom{I}{k}(I - k)^{NF}$ combinations in the $\binom{I}{1}(I - 1)^{NF}$ term that have been subtracted and $\binom{k}{2}\binom{I}{k}(I-k)^{NF}$ combinations in the $\binom{I}{2}(I - 2)^{NF}$ term that have been added. The validity of this last number can be demonstrated by the array:

$$\begin{array}{cccccc}
- & - & x_3 & x_4 & \cdots & x_I \\
- & x_2 & - & x_4 & \cdots & x_I \\
- & x_2 & x_3 & - & \cdots & x_I \\
\cdot & & & & & \\
\cdot & & & & & \\
\cdot & & & & & \\
x_1 & - & - & x_4 & \cdots & x_I \\
x_1 & - & x_3 & - & \cdots & x_I \\
\cdot & & & & & \\
\cdot & & & & & \\
\cdot & & & & & \\
x_1 & x_2 & x_3 & x_4 & \cdots & -
\end{array}$$

There are $\binom{I}{k}$ ways to choose the k outputs to be deleted. Of these k outputs there are $\binom{k}{2}$ ways to choose two that have already been deleted from the rows of the array. Hence there are $\binom{k}{2}\binom{I}{k}(I - k)^{NK}$ combinations that have k outputs deleted in the term $\binom{I}{2}(I - 2)^{NF}$. By a similar argument, the $\binom{I}{m}(I - m)^{NF}$ term has included $\binom{k}{m}\binom{I}{k}(I - k)^{NF}$ combinations of k outputs. Therefore the coefficient of $\binom{I}{k}(I - k)^{NF}$ for each term forms a binomial distribution. Summing these up to see how many times $\binom{I}{k}(I - k)^{NF}$ need be added or subtracted indicates that just one term is needed since the final term of the binomial coefficient is 1 and

$$\sum_{k=0}^{I}\binom{I}{k}(-1)^k = 0.$$

Thus the term in (2) gives the number of combinations of I_{k+1} specific outputs where each of these outputs is used at least once. The product of these two terms (1) and (2) gives the total number of allowed combinations that can occur with I_{k+1} outputs.

The term $NX(I_1, I_2, \ldots, I_k, I_{k+1}, 0, \ldots, 0)$ (3)

gives the number of combinations that the packages indexed greater than k+1 have with $NT - I_1 - I_2 - \ldots - I_{k+1}$ possible outputs. By multiplying (3) by (1) and (2) and summing the possible values of I_{k+1}, it is clear that $NX(I_1, I_2, \ldots, I_k, 0, 0, \ldots, 0)$ is found and gives the total number of combinations that will separate completely the packages indexed greater than k with I_1 outputs in the first package, I_2 outputs in the second package, and so forth up to I_k outputs in the k^{th} package.

After $NX(0, 0, \ldots, 0)$ is calculated, the probabilistic weight is given by:

$$W_p = \frac{NX(0, 0, \ldots, 0)}{NT^{NFT}}$$

The term NT^{NFT} is the total number of ways that NT outputs can be assigned to the total number of faults, NFT.

A program that calculates the probabilistic weight for up to three packages is included in the Appendix.

3.2 Multi-Output Example

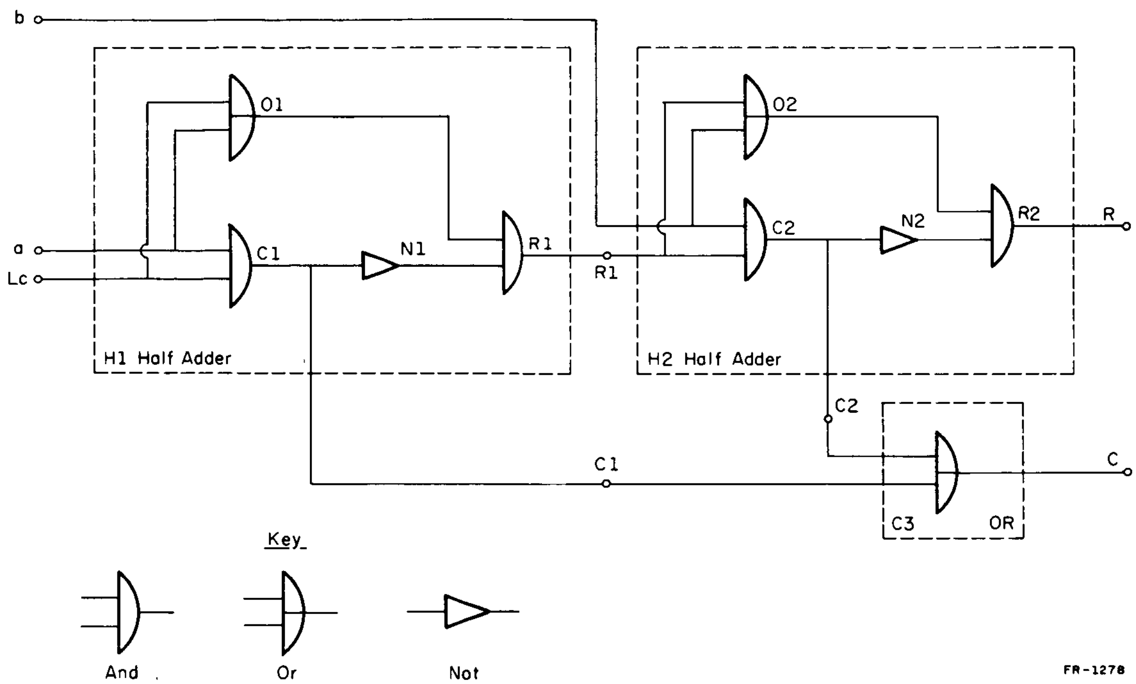
The example selected is a single stage of a parallel adder composed of two half adders and one "OR" gate. The circuit is shown in figure

10.

The faults and test inputs were simulated on the simulation portion of Seshu's Sequential Analyzer [6], and the dictionary of possible machine failures is shown in figure 11. Some of the possible faults that could occur between packages were included arbitrarily in one of the packages. The fault R1 to C2 OPEN was included in the package H2 while faults C1 to C OPEN and C2 to C OPEN were included in package C3. The fault-free machine called the good machine was considered as a separate package to assure that all faults would be separated from it for complete diagnosis.

The test fault pattern in figure 12 lists the various faults that correspond to the dictionary of machine failures in figure 11. The outputs are shown in the entries of figure 12. Under the tables the probabilities have been recorded as well as the test order. The figures from figure 12 to figure 17 show the complete diagnosis of the full adder stage to the package level.

The full adder stage was completely diagnosed to the package level with test inputs 010, 011, 101, 001, 000, and 100. Two possible tests were not required, and in three cases the best choice was arbitrary between two tests. It can be noted that a few indistinguishable faults were included in the same package since the locations of these faults were close to each other in the physical circuit. Since these indistinguishable faults were in the same package, no weight was given to separating them, and therefore no useless search for a test that would separate them was made.



b
a
Lc
O1
C1
N1
R1
H1 Half Adder
R1
O2
C2
N2
R2
R
H2 Half Adder
C2
C1
C
C3
OR
Key
And
Or
Not
FR-1278

Figure 10

DICTIONARY OF MACHINE FAILURES FOR LOGIC CIRCUIT CKT1

MACH NOS	FAILURE			MACH NOS	FAILURE		
1	GOOD MACHINE			2	A	TO C1	OPEN
3	LC	TO C1	OPEN	4	C1	OUTPUT 0	
5	C1	OUTPUT 1		6	A	TO 01	OPEN
7	LC	TO 01	OPEN	8	01	OUTPUT 0	
9	01	OUTPUT 1		10	01	TO R1	OPEN
11	N1	TO R1	OPEN	12	R1	OUTPUT 0	
13	R1	OUTPUT 1		14	R1	TO C2	OPEN
15	B	TO C2	OPEN	16	C2	OUTPUT 0	
17	C2	OUTPUT 1		18	R1	TO 02	OPEN
19	B	TO 02	OPEN	20	02	OUTPUT 0	
21	02	OUTPUT 1		22	C1	TO C	OPEN
23	C2	TO C	OPEN	24	C	OUTPUT 0	
25	C	OUTPUT 1		26	N2	TO R2	OPEN
27	02	TO R2	OPEN	28	R2	OUTPUT 0	
29	R2	OUTPUT 1					

INPUT ORDER A B LC

OUTPUT ORDER R2 C

Figure 11

Package &	Fault No.	Test (Inputs)							
		000	001	010	011	100	101	110	111
H1	2	00	01	10	11	10	01	01	11
	3	00	10	10	01	01	01	11	11
	4	00	10	10	01	10	10	01	01
	5	01	01	11	11	01	01	11	11
	6	00	10	10	01	00	01	10	11
	7	00	00	10	10	10	01	01	11
	8	00	00	10	10	00	01	10	11
	9	10	10	01	01	10	01	01	11
	10	10	10	01	01	10	01	01	11
	11	00	10	10	01	10	11	01	01
	12	00	00	10	10	00	01	10	11
	13	10	10	01	01	10	11	01	01
H2	14	00	10	01	01	10	01	01	01
	15	00	01	10	01	01	01	01	11
	16	00	10	10	10	10	01	10	11
	17	01	01	01	01	01	01	01	01
	18	00	00	10	01	00	01	01	11
	19	00	10	00	01	10	01	01	01
	20	00	00	00	01	00	01	01	01
	21	10	10	10	01	10	11	01	11
	26	00	10	10	11	10	01	11	11
	27	10	10	10	01	10	11	01	11
	28	00	00	00	01	00	01	01	01
	29	10	10	10	11	10	11	11	11
C3	22	00	10	10	01	10	00	01	10
	23	00	10	10	00	10	01	00	11
	24	00	10	10	00	10	00	00	10
	25	01	11	11	01	11	01	01	11
M	1	00	10	10	01	10	01	01	11
	W_p	0.1812	2.5365	2.8161	2.6090	2.5365	0.2347	2.6090	0.6177 $\times 10^{-10}$
	T.O.	8	4	1	2	4	7	2	6

FR-1279

Figure 12

Package & Fault No.		Test (Inputs)						
		000	001	011	100	101	110	111
H1	2	00	01	11	10	01	01	11
	3	00	10	01	01	01	11	11
	4	00	10	01	10	10	01	01
	6	00	10	01	00	01	10	11
	7	00	00	10	10	01	01	11
	8	00	00	10	00	01	10	11
	11	00	10	01	10	11	01	01
	12	00	00	10	00	01	10	11
H2	15	00	01	01	01	01	01	11
	16	00	10	10	10	01	10	11
	18	00	00	01	00	01	01	11
	21	10	10	01	10	11	01	11
	26	00	10	11	10	01	11	11
	27	10	10	01	10	11	01	11
	29	10	10	11	10	11	11	11
C3	22	00	10	01	10	00	01	10
	23	00	10	00	10	01	00	11
	24	00	10	00	10	00	00	10
M	1	00	10	01	10	01	01	11
H1	5	01	01	11	01	01	11	11
C3	25	01	11	01	11	01	01	11
H1	9	10	10	01	10	01	01	11
	10	10	10	01	10	01	01	11
	13	10	10	01	10	11	01	01
H2	14	00	10	01	10	01	01	01
	17	01	01	01	01	01	01	01
W_p		0.0042	0.0477	1.0820	0.0477	0.1485	1.0821	0.0094 $\times 10^{-6}$
T.O.		7	4	1	4	3	1	6

Figure 13

FR-1280

Package & Fault No.		Test (Inputs)					
		000	001	100	101	110	111
H 1	3	00	10	01	01	11	11
	4	00	10	10	10	01	01
	6	00	10	00	01	10	11
	11	00	10	10	11	01	01
H 2	15	00	01	01	01	01	11
	18	00	00	00	01	01	11
	21	10	10	10	11	01	11
	27	10	10	10	11	01	11
C 3	22	00	10	10	00	01	10
M	1	00	10	10	01	01	11
H 1	2	00	01	10	01	01	11
H 2	26	00	10	10	01	11	11
	29	10	10	10	11	11	11
H 1	7	00	00	10	01	01	11
	8	00	00	00	01	10	11
	12	00	00	00	01	10	11
H 2	16	00	10	10	01	10	11
H 1	9	10	10	10	01	01	11
	10	10	10	10	01	01	11
	13	10	10	10	11	01	01
H 2	14	00	10	10	01	01	01
	17	01	01	01	01	01	01
	Wp	0.1159	0.1545	0.5866	4.1062	0.0410	1.7598 $\times 10^{-3}$
	T.O.	5	4	3	1	6	2

FR-1281

Figure 14

Package & Fault No.	Test (Inputs) 000	001	100	110	111
H1 { 3	00	10	01	11	11
H1 { 6	00	10	00	10	11
H2 { 15	00	01	01	01	11
H2 { 18	00	00	00	01	11
M { 1	00	10	10	01	11
H1 { 11	00	10	10	01	01
H2 { 21	10	10	10	01	11
H2 { 27	10	10	10	01	11
H1 { 2	00	01	10	01	11
H2 { 26	00	10	10	11	11
H1 { 7	00	00	10	01	11
H1 { 8	00	00	00	10	11
H1 { 12	00	00	00	10	11
H2 { 16	00	10	10	10	11
H1 { 9	10	10	10	01	11
H1 { 10	10	10	10	01	11
H2 { 14	00	10	10	01	01
H2 { 17	01	01	01	01	01
W_p	0.4877	8.779	3.9018	1.9509 $\times 10^{-2}$	0.4877
T.O.	4	1	2	3	4

FR-1282

Figure 15

Package & Fault No.		Test (Inputs)			
		000	100	110	111
H1	3	00	01	11	11
	6	00	00	10	11
M	1	00	10	01	11
H1	11	00	10	01	01
H2	21	10	10	01	11
	27	10	10	01	11
H1	9	10	10	01	11
	10	10	10	01	11
H2	14	00	10	01	01
W_p		0.2500	0.0625	0.0625	0.2500
T.O.		1	3	3	1

FR-1283

Figure 16

Package & Fault No.		Test (Inputs)		
		100	110	111
H1	3	01	11	11
	6	00	10	11
M	1	10	01	11
W_p		1.00	1.00	0.25
T.O.		1	1	3

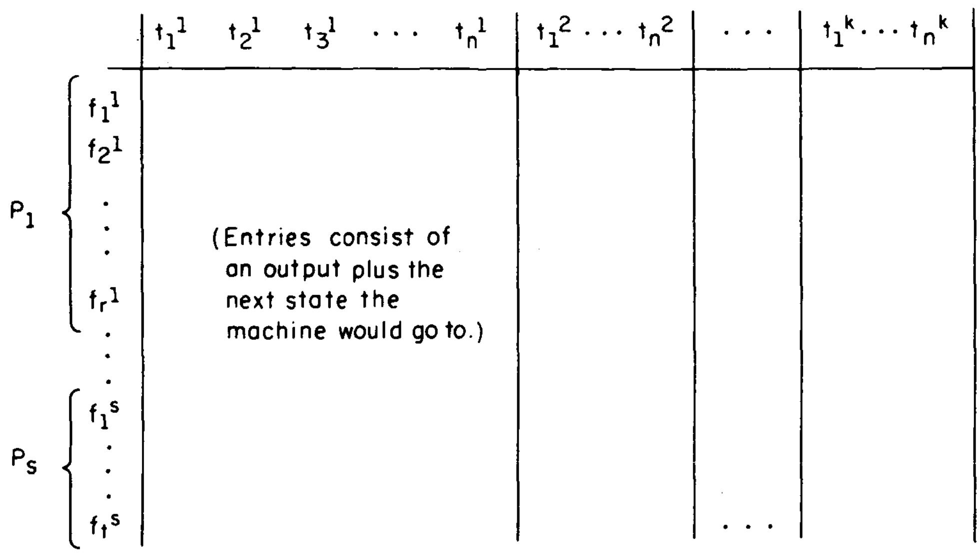
FR-1284

Figure 17

4. FUTURE WORK AND CONCLUSIONS

4.1 Sequential Machine Extension

The probability weighting procedure could be extended to sequential machines. A possible extension would involve forming and using the following tables:



FR-1285

Figure 18

It is assumed that there are k states and s packages with a varying number of faults in each package. A row of the table represents a fault in a given package with its outputs for various test inputs. Each column represents a test input with the machine in a specific state. For example, the test t_m^q is the test input t_m applied to the sequential machine with the machine in state q. The states of the machine must be accounted for in the bookkeeping since a fault sequential machine may not only have a different output but also a different next state when compared with the fault-free machine. Therefore each table position entry has the output for the applied test as well as the next state that the machine with the fault would go to. In applying the probability weighting procedure it is assumed that an initial state of the machine is given or that the machine can be put into a given state. If the state that the machine goes to is observable it will also partition the packages and shorten the diagnostic process.

If state i is the initial state then the subarray shown in figure 19 is formed by all columns of tests with superscript i and all rows.

When this array is used, the probabilistic weights are found and the first test to be applied is selected. After the first test has been applied, the array, from which to find the next test, is obtained by first examining the next state that each fault would put the machine in. If fault f_j^i after applying the first test puts the machine in state h, then the portion of the f_j^i row under column $t_1^h, \ldots, t_n^h$ of figure 18 would be placed in the new array for row f_j^i. The tests of the new array drop their superscripts. After the array has been completed it contains the condition of the machine after the

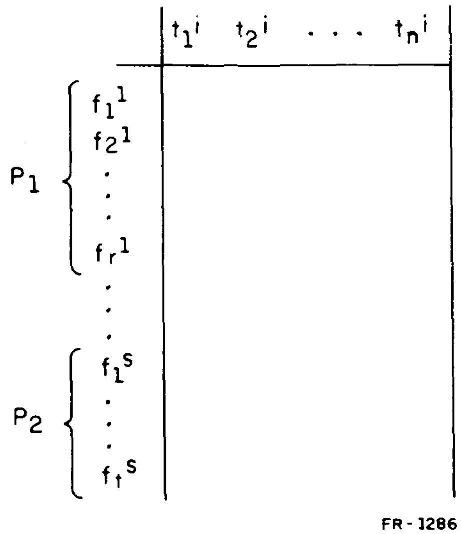


Figure 19

second test has been applied. This array thus shows the outputs that the sequential machine would give and its next state for any given fault after applying the first test and any second test. The probabilistic weights are then calculated and a next test selected. This process is continued until the machine is diagnosed.

4.2 Future Research Areas

Applying probabilistic weights to sequential machines will require, in addition, a procedure for determining the next test to be applied when the present possible tests do not give any additional diagnostic information, but other tests in other states do. A criterion is needed for terminating the diagnostic process so that all packages are separated that can be, but indistinguishable faults in different packages do not cause useless tests to be inserted in the sequence of diagnostic tests. A possible research problem is to provide the detailed criteria for the extension of this probability weighting method to sequential machines.

The probability weighting procedure could be altered in two ways that may improve the procedure for combinational machines. One possibility would be to purposely weight some of the tests zero in order to shorten the time for calculating the weights. The purpose for weighting the tests each time is to select the best test that would next be applied to the machine. The order of the tests is unnecessary especially for those tests which give the least information. Weighting the tests that give the least information zero would not prevent selecting the best test to be applied. Giving the weight of zero to some tests could be done by assuming fewer outputs for the hypothetical test so that this test cannot separate the remaining packages. For example, if five packages remain after applying a test and only four outputs are assumed possible for the hypothetical test, then the probabilistic weight is zero. Other tests that reduce the number of packages left to be diagnosed sufficiently would still have non-zero probabilistic weights which would provide the means for selecting the best test. Under this new consideration a procedure would have to be developed for determining how many outputs should be assumed for the next test. Too few assumed outputs may jeopardize the selection of the best test while too many assumed outputs would not take the greatest advantage of the shortened procedure.

Another variation of the probability weighting procedure to consider is the calculation of the probabilistic weights based on two or three tests taken at a time in order to select the best two or three next tests for diagnosis. This approach would increase the chances of achieving an optimal set of diagnostic tests but would require more bookkeeping and computations. For certain problems this may be desirable.

A further area of research would be to examine more closely diagnosis to the package level with only a truth table for each package. What assumptions are necessary or useful in forming such a general diagnostic procedure?

A multiple output optimum procedure that would diagnose to the package level is another research problem. A procedure similar to Bouknight's [1] might be considered.

What types of elements should be included in packages that would lend themselves better to diagnosis to the package level? Since package level diagnosis is possible, could this be extended to a computer block diagnosis? If so, what are the advantages? The disadvantages? These questions suggest further areas of research.

4.3 Conclusions

This investigation has produced an instructive, although not practical, optimum procedure for diagnosing multi-output machines to the package level somewhat analogous to the prime implicant problem in logic design. The investigation also has produced a procedure for selecting a set of tests sequentially to diagnose multi-output machines to the package level, thereby eliminating redundant tests. Although the procedure may not always select the optimum set, it does select a near-optimal set for combinational machines. The probability weighting procedure has the advantage that the set of selected tests is found without exhaustive trial combinations of tests which other optimal procedures require. The tests are added to the set used for diagnosis one at a time. After applying previously selected tests, the test that has the largest probabilistic weight is chosen and added to the set. The probabilistic weight is the probability that a hypothetical test would completely separate all remaining unseparated packages after the selected tests and the test being weighted had been applied to the machine. Since diagnosis is accomplished to the package level, fewer tests are usually required than would be for diagnosing to the fault level.

LIST OF REFERENCES

1. Bouknight, W. J., "On the Generation of Diagnostic Test Procedures," Coordinated Science Laboratory Report 292, University of Illinois, May 1966.

2. Chang, H. Y., "An Algorithm for Selecting an Optimum Set of Diagnostic Tests," IEEE Transactions on Electronic Computers, Volume EC-14, Number 5, pp. 706-710, October 1965.

3. Howarter, D. R., "The Selection of Failure Location Tests by Path Sensitizing Techniques," Coordinated Science Laboratory Report 316, University of Illinois, August 1966.

4. Seshu, S. and Freeman, D. N., "The Diagnosis of Asynchronous Sequential Switching Systems," IRE Transactions on Electronic Computers, Volume EC-11, Number 4, pp. 459-465, August 1962.

5. Seshu, S., "On an Improved Diagnosis Program," Coordinated Science Laboratory Report 207, University of Illinois, May 1964.

6. Seshu, S., "The Logic Organizer and Diagnosis Programs," Coordinated Science Laboratory Report 226, University of Illinois, July 1964.

APPENDIX

The program SEPPROB is a FORTRAN program for finding the probability of separating up to three packages. The inputs are as follows: MNP is the maximum number of outputs, CKN is the number of tests whose probabilities are to be calculated, NM is the number of packages, and NF(i), the number of faults in the i^{th} package.

```
      program sepprob
      dimension
     'nf(3),jj(3),nx(16,16,16)
      common nf, jj, nx, mno, ll, nm, mindx
c     mno=maximum number of outputs.  ckn=number of tests
      read 1, mno, ckn
1     format (1i2,1f3.1)
      ck=Ø.Ø
2     read 3, nm,nf(1),nf(2),nf(3)
3     format (4i2)
      ck=ck+1.Ø
      do 4 k=1,nm
4     jj(k)=Ø
      call nindxm(i1m,nf(1))
      do 9 i1=1,i1m
      jj(1)=i1
      call nindxm(i2m,nf(2))
      do 8 i2=1,i2m
      jj(2)=i2
      call nindxm(i3m,nf(3))
      do 5 i3=1,i3m
      jj1=jj(1)
      jj2=jj(2)
5     nx(jj1,jj2,i3)=1
      jj(3)=Ø
8     call nxtv(nf(3))
```

```
      jj(2)=Ø
9     call nxtv(nf(2))
      jj(1)=Ø
      call nxtv(nf(1))
      fxn=nx(Ø,Ø,Ø)
      denom=mno''(nf(1)+nf(2)+nf(3))
      pt=fxn/denom
      print 25, mno, nf(1), nf(2), nf(3), pt
25    format (//,1x,19hmax number outputs ,1i2,3x,17hnumber faults per,
     18h module ,2(1i2,1h,),1i2,/,1x,14hprobability = ,1f7.5)
      if(ck.eq.ckn) 19,2
19    end
c     to calculate nx(i1,i2,Ø,Ø) from sum over j of nx(i1,i2,j,Ø)
      subroutine nxtv(nfk)
      dimension
     'nf(3),jj(3),nx(16,16,16)
      common nf, jj, nx, mno, ll, nm, mindx
      call nindxm(maxi,nfk)
      if(nfk.eq.Ø) 56,53
53    maxj=mindx+nm-ll-1
      imodl=Ø

      do 54 nn=1,maxi
      imodc=Ø
      do 55 mm=1,nn
```

```
      mmlo=mm-1
55    imodc=imodc+(-1)''mmlo'nbinomc(nn,mmlo)'(nn-mmlo)''nfk
      jj(ll+1)=nn
54    imodl=imodl+imodc'nbinomc(maxj,nn)'nx(jj(1),jj(2),jj(3))
      jj(ll+1)=Ø
      jj1=jj(1)
      jj2=jj(2)
      jj3=jj(3)
      nx(jj1,jj2,jj3)=imodl
      return
56    do 57 k=1,mno
      jj(ll)=k
      jj1=jj(1)
      jj2=jj(2)
      jj3=jj(3)
57    nx(jj1,jj2,jj3)=1
      jj(ll)=Ø
      end
c     to find next index maximum
      subroutine nindxm(ikm,nfks)
      dimension
     'nf(3),jj(3),nx(16,16,16)
      common nf, jj, nx, mno, ll, nm, mindx
      ll=Ø
      k=1
```

```
79    if(jj(k).eq.Ø) 8Ø,81
81    ll=ll+1
      k=k+1
      if(ll.eq.nm) 8Ø,79
8Ø    mindx=mno-nm+ll+1
      do 82 kk=1,ll
82    mindx=mindx-jj(kk)
      if (mindx.gt.nfks) 83,84
83    ikm=nfks
      return
84    ikm=mindx
      end
      function nbinomc(nnn,nrr)
      if (nrr.eq.nnn) 177,173
173   if (nrr.eq.Ø) 177,174
174   fnn=nnn
      rr=nrr
      binomc=fnn/rr
      nrrlo=nrr-1
      do 176 k=1,nrrlo,1
      sub1=nnn-k
      sub2=nrr-k
      subm=sub1/sub2
176   binomc=binomc'subm
      nbinomc=binomc
```

```
      go to 178
177   nbinomc=1
178   end
      end
x
..
..
```

DISTRIBUTION LIST AS OF APRIL 1, 1967

1 Dr. Edward M. Reilley
Asst. Director (Research)
Ofc. of Defense Res. & Engrg.
Department of Defense
Washington, D. C. 20301

1 Office of Deputy Director
(Research and Information Rm. 3D1037)
Department of Defense
The Pentagon
Washington, D. C. 20301

1 Director
Advanced Research Projects Agency
Department of Defense
Washington, D. C. 20301

1 Director for Materials Sciences
Advanced Research Projects Agency
Department of Defense
Washington, D. C. 20301

1 Headquarters
Defense Communications Agency (333)
The Pentagon
Washington, D. C. 20305

50 Defense Documentation Center
Attn: TISIA
Cameron Station, Bldg. 5
Alexandria, Virginia 22314

1 Director
National Security Agency
Attn: TDL
Fort George G. Meade, Maryland 20755

1 Weapons Systems Evaluation Group
Attn: Col. Daniel W. McElwee
Department of Defense
Washington, D. C. 20305

1 National Security Agency
Attn: R4-James Tippet
Office of Research
Fort George G. Meade, Maryland 20755

1 Central Intelligence Agency
Attn: OCR/DD Publications
Washington, D. C. 20505

1 Colonel Kee
AFRSTE
Hqs. USAF
Room 1D-429, The Pentagon
Washington, D. C. 20330

1 Colonel A. Swan
Aerospace Medical Division
Brooks Air Force Base, Texas 78235

1 AUL3T-9663
Maxwell AFB, Alabama 36112

1 AFFTC (FTBPP-2)
Technical Library
Edwards AFB, California 93523

1 Space Systems Division
Air Force Systems Command
Los Angeles Air Force Station
Los Angeles, California 90045
Attn: SSSD

1 Major Charles Waespy
Technical Division
Deputy for Technology
Space Systems Division, AFSC
Los Angeles, California 90045

1 SSD(SSTRT/Lt. Starbuck)
AFUPO
Los Angeles, California 90045

1 Det. #6, OAR (LOOAR)
Air Force Unit Post Office
Los Angeles, California 90045

1 Systems Engineering Group (RTD)
Technical Information Reference Branch
Attn: SEPIR
Directorate of Engineering Standards
& Technical Information
Wright-Patterson AFB, Ohio 45433

1 ARL (ARIY)
Wright-Patterson AFB, Ohio 45433

1 Dr. H. V. Noble
Air Force Avionics Laboratory
Wright-Patterson AFB, Ohio 45433

1 Mr. Peter Murray
Air Force Avionics Laboratory
Wright-Patterson AFB, Ohio 45433

1 AFAL (AVTE/R.D. Larson)
Wright-Patterson AFB, Ohio 45433

2 Commanding General
Attn: STEWS-WS-VT
White Sands Missile Range,
New Mexico 88002

1 RADC (EMLAL-I)
Griffiss AFB, New York 13442
Attn: Documents Library

1 Academy Library (DFSLB)
U. S. Air Force Academy
Colorado Springs, Colorado 80912

1 Lt. Col. Bernard S. Morgan
Frank J. Seiler Research Laboratory
U. S. Air Force Academy
Colorado Springs, Colorado 80912

1 APGC (PGBPS-12)
Elgin AFB, Florida 32542

1 Commanding Officer
Human Engineering Laboratories
Aberdeen Proving Ground, Maryland 21005

1 Director
U. S. Army Engineer Geodesy, Intelligence
and Mapping
Research and Development Agency
Fort Belvior, Virginia 22060

1 Commandant
U. S. Army Command and General Staff College
Attn: Secretary
Fort Leavenworth, Kansas 66270

1 Dr. H. Robl
Deputy Chief Scientist
U. S. Army Research Office (Durham)
Box CM, Duke Station
Durham, North Carolina 27706

1 Commanding Officer
U. S. Army Research Office (Durham)
Attn: CRD-AA-IP (Richard O. Ulsh)
Box CM, Duke Station
Durham, North Carolina 27706

1 Librarian
U. S. Army Military Academy
West Point, New York 10996

1 The Walter Reed Institute of Research
Walter Reed Medical Center
Washington, D. C. 20012

1 Commanding Officer
U. S. Army Electronics R&D Activity
Fort Huachuca, Arizona 85163

1 Commanding Officer
U. S. Army Engineer R&D Laboratory
Attn: STINFO Branch
Fort Belvoir, Virginia 22060

1 Commanding Officer
U. S. Army Electronics R&D Activity
White Sands Missile Range, New Mexico 88002

1 Dr. S. Benedict Levin, Director
Institute for Exploratory Research
U. S. Army Electronics Command
Fort Monmouth, New Jersey 07703

1 Director
Institute for Exploratory Research
U. S. Army Electronics Command
Attn: Mr. Robert O. Parker, Executive
Secretary, JSTAC (AMSEL-XL-D)
Fort Monmouth, New Jersey 07703

1 Commanding General
U. S. Army Electronics Command
Fort Monmouth, New Jersey 07703

Attn: AMSEL-SC
RD-D
RD-G
RD-GF
RD-MAT
XL-D
XL-E
XL-C
XL-S
HL-D
HL-CT-R
HL-CT-P
HL-CT-L
HL-CT-O
HL-CT-I
HL-CT-A
NL-D
NL-A
NL-P
NL-R
NL-S
KL-D
KL-E
KL-S
KL-T
VL-D
WL-D

1 Chief of Naval Research
Department of the Navy
Washington, D. C. 20360
Attn: Code 427

3 Chief of Naval Research
Department of the Navy
Washington, D. C. 20360
Attn: Code 437

2 Naval Electronics Systems Command
ELEX 03
Falls Church, Virginia 22046

1 Naval Ship Systems Command
SHIP 031
Washington, D. C. 20360

1 Naval Ship Systems Command
SHIP 035
Washington, D. C. 20360

2 Naval Ordnance Systems Command
ORD 32
Washington, D. C. 20360

2 Naval Air Systems Command
AIR 03
Washington, D. C. 20360

2 Commanding Officer
Office of Naval Research Branch Office
Box 39, Navy No. 100 F.P.O.
New York, New York 09510

1 AFETR Technical Library
(ETV, MU-135)
Patrick AFB, Florida 32925

1 AFETR (ETLLG-I)
STINFO Officer (For Library)
Patrick AFB, Florida 32925

1 Dr. L. M. Hollingsworth
AFCRL (CRN)
L. G. Hanscom Field
Bedford, Massachusetts 01731

1 AFCRL (CRMXLR)
AFCRL Research Library, Stop 29
L. G. Hanscom Field
Bedford, Massachusetts 01731

1 Colonel Robert E. Fontana
Department of Electrical Engineering
Air Force Institute of Technology
Wright-Patterson AFB, Ohio 45433

1 Colonel A. D. Blue
RTD (RTTL)
Bolling Air Force Base, D. C. 20332

1 Dr. I. R. Mirman
AFSC (SCT)
Andrews AFB, Maryland 20331

1 Colonel J. D. Warthman
AFSC (SCTR)
Andrews AFB, Maryland 20331

1 Lt. Col. J. L. Reeves
AFSC (SCBB)
Andrews AFB, Maryland 20331

2 ESD (ESTI)
L. G. Hanscom Field
Bedford, Massachusetts 01731

1 AEDC (ARO, INC)
Attn: Library/Documents
Arnold AFS, Tennessee 37389

2 European Office of Aerospace Research
Shell Building
47 Rue Cantersteen
Brussels, Belgium

5 Lt. Col. Robert B. Kalisch
Chief, Electronics Division
Directorate of Engineering Sciences
Air Force Office of Scientific Research
Arlington, Virginia 22209

1 U. S. Army Research Office
Attn: Physical Sciences Division
3045 Columbia Pike
Arlington, Virginia 22204

1 Research Plans Office
U. S. Army Research Office
3045 Columbia Pike
Arlington, Virginia 22204

1 Commanding General
U. S. Army Materiel Command
Attn: AMCRD-RS-DE-E
Washington, D. C. 20315

1 Commanding General
U. S. Army Strategic Communications Command
Washington, D. C. 20315

1 Commanding Officer
U. S. Army Materials Research Agency
Watertown Arsenal
Watertown, Massachusetts 02172

1 Commanding Officer
U. S. Army Ballistics Research Laboratory
Attn: V. W. Richards
Aberdeen Proving Ground
Aberdeen, Maryland 21005

1 Commandant
U. S. Army Air Defense School
Attn: Missile Sciences Division, C&S Dept.
P.O. Box 9390
Fort Bliss, Texas 79916

1 Redstone Scientific Information Center
Attn: Chief, Document Section
Redstone Arsenal, Alabama 35809

1 Commanding General
Frankford Arsenal
Attn: SMUFA-1310 (Dr. Sidney Ross)
Philadelphia, Pennsylvania 19137

1 U. S. Army Munitions Command
Attn: Technical Information Branch
Picatinney Arsenal
Dover, New Jersey 07801

1 Commanding Officer
Harry Diamond Laboratories
Attn: Dr. Berthold Altman (AMXDO-TI)
Connecticut Avenue and Van Ness Street, N.W.
Washington, D. C. 20438

1 Commanding Officer
U. S. Army Security Agency
Arlington Hall
Arlington, Virginia 22212

1 Commanding Officer
U. S. Army Limited War Laboratory
Attn: Technical Director
Aberdeen Proving Ground
Aberdeen, Maryland 21005

1 Commanding Officer
Office of Naval Research Branch Office
219 South Dearborn Street
Chicago, Illinois 60604

1 Commanding Officer
Office of Naval Research Branch Office
1030 East Green Street
Pasadena, California 91101

1 Commanding Officer
Office of Naval Research Branch Office
207 West 24th Street
New York, New York 10011

1 Commanding Officer
Office of Naval Research Branch Office
495 Summer Street
Boston, Massachusetts 02210

8 Director, Naval Research Laboratory
Technical Information Officer
Washington, D. C. 20390
Attn: Code 2000

1 Commander
Naval Air Development and Material Center
Johnsville, Pennsylvania 18974

2 Librarian
U. S. Naval Electronics Laboratory
San Diego, California 95152

1 Commanding Officer and Director
U. S. Naval Underwater Sound Laboratory
Fort Trumbull
New London, Connecticut 06840

1 Librarian
U. S. Navy Post Graduate School
Monterey, California 93940

1 Commander
U. S. Naval Air Missile Test Center
Point Mugu, California 95468

1 Director
U. S. Naval Observatory
Washington, D. C. 20390

2 Chief of Naval Operations
OP-07
Washington, D. C. 20350

1 Director, U. S. Naval Security Group
Attn: G43
3801 Nebraska Avenue
Washington, D. C. 20016

2 Commanding Officer
Naval Ordnance Laboratory
White Oak, Maryland 21162

1 Commanding Officer
Naval Ordnance Laboratory
Corona, California 91720

1 Commanding Officer
Naval Ordnance Test Station
China Lake, California 93555

1 Commanding Officer
Naval Avionics Facility
Indianapolis, Indiana 46218

1 Commanding Officer
Naval Training Device Center
Orlando, Florida 32813

1 U. S. Naval Weapons Laboratory
Dahlgren, Virginia 22448

1 Weapons Systems Test Division
Naval Air Test Center
Patuxtent River, Maryland 20670
Attn: Library

1 Head, Technical Division
U. S. Naval Counter Intelligence Support Center
Fairmont Building
4420 North Fairfax Drive
Arlington, Virginia 22203

1 Mr. Charles F. Yost
Special Asst. to the Director of Research
National Aeronautics and Space Administration
Washington, D. C. 20546

1 Dr. H. Harrison, Code RRE
Chief, Electrophysics Branch
National Aeronautics and Space Administration
Washington, D. C. 20546

1 Goddard Space Flight Center
National Aeronautics and Space Administration
Attn: Library C3/TDL
Green Belt, Maryland 20771

1 NASA Lewis Research Center
Attn: Library
21000 Brookpark Road
Cleveland, Ohio 44135

1 National Science Foundation
Attn: Dr. John R. Lehmann
Division of Engineering
1800 G Street, N.W.
Washington, D. C. 20550

1 U. S. Atomic Energy Commission
Division of Technical Information Extension
P. O. Box 62
Oak Ridge, Tennessee 37831

1 Los Alamos Scientific Laboratory
Attn: Reports Library
P. O. Box 1663
Los Alamos, New Mexico 87544

2 NASA Scientific & Technical Information Facility
Attn: Acquisitions Branch (S/AK/DL)
P. O. Box 33
College Park, Maryland 20740

1 Director
Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

1 Polytechnic Institute of Brooklyn
55 Johnson Street
Brooklyn, New York 11201
Attn: Mr. Jerome Fox
Research Coordinator

1 Director
Columbia Radiation Laboratory
Columbia University
538 West 120th Street
New York, New York 10027

1 Director
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801

1 Director
Stanford Electronics Laboratories
Stanford University
Stanford, California 94305

1 Director
Electronics Research Laboratory
University of California
Berkeley, California 94720

1 Director
Electronic Sciences Laboratory
University of Southern California
Los Angeles, California 90007

1 Professor A. A. Dougal, Director
Laboratories for Electronics and Related Sciences Research
University of Texas
Austin, Texas 78712

1 Division of Engineering and Applied Physics
210 Pierce Hall
Harvard University
Cambridge, Massachusetts 02138

1 Aerospace Corporation
P. O. Box 95085
Los Angeles, California 90045
Attn: Library Acquisitions Group

1 Professor Nicholas George
California Institute of Technology
Pasadena, California 91109

1 Aeronautics Library
Graduate Aeronautical Laboratories
California Institute of Technology
1201 East California Boulevard
Pasadena, California 91109

1 Director, USAF Project RAND
Via: Air Force Liaison Office
The RAND Corporation
1700 Main Street
Santa Monica, California 90406
Attn: Library

1 The Johns Hopkins University
Applied Physics Laboratory
8621 Georgia Avenue
Silver Spring, Maryland 20910
Attn: Boris W. Kuvshinoff
Document Librarian

1 Hunt Library
Carnegie Institute of Technology
Schenley Park
Pittsburgh, Pennsylvania 15213

1 Dr. Leo Young
Stanford Research Institute
Menlo Park, California 94025

1 Mr. Henry L. Bachmann
Assistant Chief Engineer
Wheeler Laboratories
122 Cuttermill Road
Great Neck, New York 11021

1 School of Engineering Sciences
Arizona State University
Tempe, Arizona 85281

1 University of California at Los Angeles
Department of Engineering
Los Angeles, California 90024

1 California Institute of Technology
Pasadena, California 91109
Attn: Documents Library

1 University of California
Santa Barbara, California 93106
Attn: Library

1 Carnegie Institute of Technology
Electrical Engineering Department
Pittsburgh, Pennsylvania 15213

1 University of Michigan
Electrical Engineering Department
Ann Arbor, Michigan 48104

1 New York University
College of Engineering
New York, New York 10019

1 Syracuse University
Department of Electrical Engineering
Syracuse, New York 13210

1 Yale University
Engineering Department
New Haven, Connecticut 06520

1 Airborne Instruments Laboratory
Deerpark, New York 11729

1 Bendix Pacific Division
11600 Sherman Way
North Hollywood, California 91605

1 General Electric Company
Research Laboratories
Schenectady, New York 12301

1 Lockheed Aircraft Corporation
P. O. Box 504
Sunnyvale, California 94088

1 Raytheon Company
Bedford, Massachusetts 01730
Attn: Librarian

1 Dr. G. J. Murphy
The Technological Institute
Northwestern University
Evanston, Illinois 60201

1 Dr. John C. Hancock, Director
Electronic Systems Research Laboratory
Purdue University
Lafayette, Indiana 47907

1 Director
Microwave Laboratory
Stanford University
Stanford, California 94305

1 Emil Schafer, Head
Electronics Properties Info Center
Hughes Aircraft Company
Culver City, California 90230

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* University of Illinois Coordinated Science Laboratory Urbana, Illinois 61801	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

A PROCEDURE FOR RANKING DIAGNOSTIC TEST INPUTS

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*

POWELL, Theo

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
May, 1967	55	6

8a. CONTRACT OR GRANT NO. DA 28 043 AMC 00073(E) b. PROJECT NO. 20014501B31F c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) R-354 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

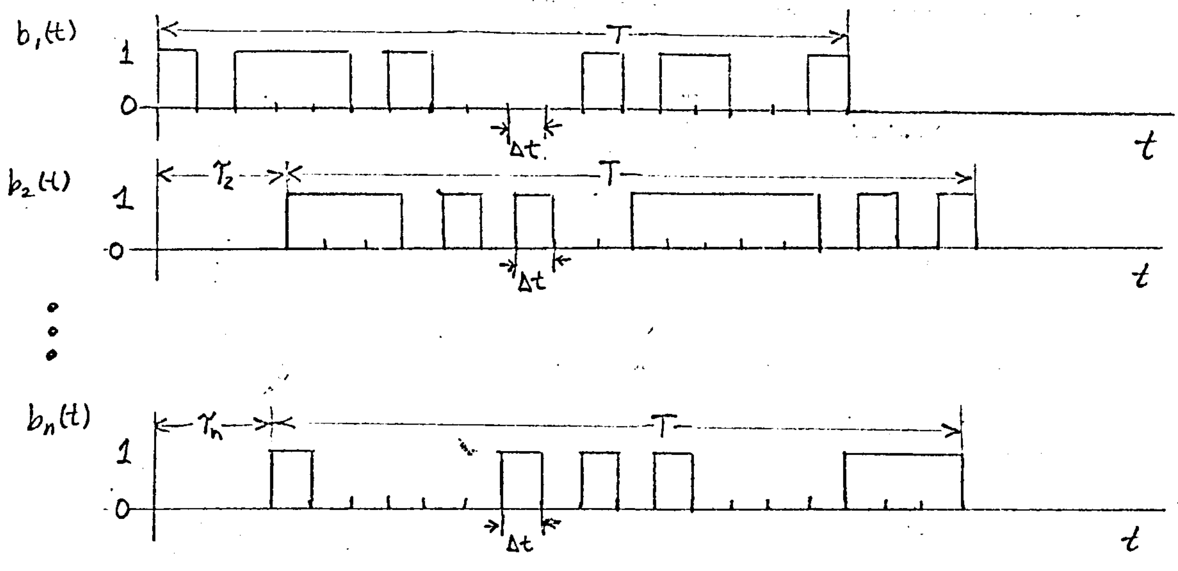
10. DISTRIBUTION STATEMENT

Distribution of this report is unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Joint Services Electronics Program thru U. S. Army Electronics Command Ft Monmouth, New Jersey 07703

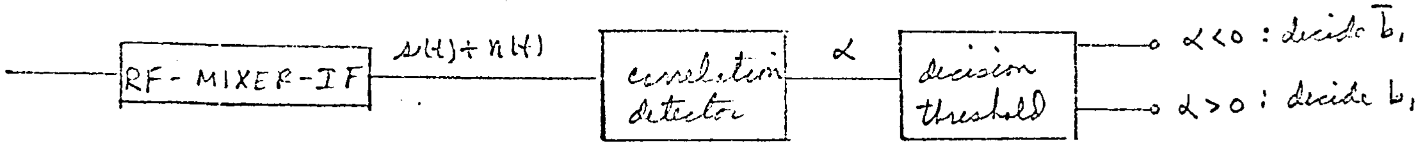
13. ABSTRACT

NONE



$b_1(t)$
1
0
T
Δt
t
$b_2(t)$
τ_2
T
1
0
Δt
t
$b_n(t)$
τ_n
T
1
0
Δt
t

Fig. 3



RF - MIXER - IF
s(t) + n(t)
correlation detector
α
decision threshold
α < 0 : decide $\bar{b}_1$
α > 0 : decide b_1

Fig. 4