2

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-567

A Survey of and an Introduction to Fault Diagnosis Algorithms

F. P. Mathur

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

October 1, 1972

45

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No. 33-567	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A SURVEY OF AND AN INTRODUCTION TO FAULT DIAGNOSIS ALGORITHMS		5. Report Date October 1, 1972 6. Performing Organization Code
7. Author(s) F. P. Mathur		8. Performing Organization Report No.
9. Performing Organization Name and Address JET PROPULSION LABORATORY California Institute of Technology 4800 Oak Grove Drive Pasadena, California 91103		10. Work Unit No. 11. Contract or Grant No. NAS 7-100 13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address NATIONAL AERONAUTICS AND SPACE ADMINISTRATION Washington, D.C. 20546		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This report surveys the field of fault diagnosis and introduces the reader to some of the key algorithms and heuristics currently in use. Fault diagnosis is an important and a rapidly growing discipline. This is important to the Jet Propulsion Laboratory's research efforts in the design of self-repairable computers because the present diagnosis resolution of its fault-tolerant computer is limited to a functional unit or processor. Better resolution is necessary before failed units can become partially reusable. The approach that holds the greatest promise is that of resident microdiagnostics; however, that presupposes a microprogrammable architecture for the computer being self-diagnosed. The presentation here is tutorial and contains examples. An extensive bibliography of some 220 entries is included.

17. Key Words (Selected by Author(s))	18. Distribution Statement
Computer Applications and Equipment Electronic Components and Circuits	Unclassified -- Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	38	

HOW TO FILL OUT THE TECHNICAL REPORT STANDARD TITLE PAGE

Make items 1, 4, 5, 9, 12, and 13 agree with the corresponding information on the report cover. Use all capital letters for title (item 4). Leave items 2, 6, and 14 blank. Complete the remaining items as follows:

3. Recipient's Catalog No. Reserved for use by report recipients.

7. Author(s). Include corresponding information from the report cover. In addition, list the affiliation of an author if it differs from that of the performing organization.

8. Performing Organization Report No. Insert if performing organization wishes to assign this number.

10. Work Unit No. Use the agency-wide code (for example, 923-50-10-06-72), which uniquely identifies the work unit under which the work was authorized. Non-NASA performing organizations will leave this blank.

11. Insert the number of the contract or grant under which the report was prepared.

15. Supplementary Notes. Enter information not included elsewhere but useful, such as: Prepared in cooperation with... Translation of (or by)... Presented at conference of... To be published in...

16. Abstract. Include a brief (not to exceed 200 words) factual summary of the most significant information contained in the report. If possible, the abstract of a classified report should be unclassified. If the report contains a significant bibliography or literature survey, mention it here.

17. Key Words. Insert terms or short phrases selected by the author that identify the principal subjects covered in the report, and that are sufficiently specific and precise to be used for cataloging.

18. Distribution Statement. Enter one of the authorized statements used to denote releasability to the public or a limitation on dissemination for reasons other than security of defense information. Authorized statements are "Unclassified–Unlimited," "U. S. Government and Contractors only," "U. S. Government Agencies only," and "NASA and NASA Contractors only."

19. Security Classification (of report). NOTE: Reports carrying a security classification will require additional markings giving security and downgrading information as specified by the Security Requirements Checklist and the DoD Industrial Security Manual (DoD 5220.22-M).

20. Security Classification (of this page). NOTE: Because this page may be used in preparing announcements, bibliographies, and data banks, it should be unclassified if possible. If a classification is required, indicate separately the classification of the title and the abstract by following these items with either "(U)" for unclassified, or "(C)" or "(S)" as applicable for classified items.

21. No. of Pages. Insert the number of pages.

22. Price. Insert the price set by the Clearinghouse for Federal Scientific and Technical Information or the Government Printing Office, if known.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

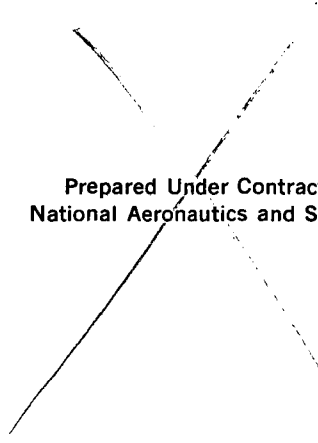
Technical Memorandum 33-567

A Survey of and an Introduction to Fault Diagnosis Algorithms

F. P. Mathur

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

October 1, 1972



Prepared Under Contract No. NAS 7-100
National Aeronautics and Space Administration

PREFACE

The work described in this report was performed by the Astrionics Division of the Jet Propulsion Laboratory.

CONTENTS

FIGURES (contd)

ABSTRACT

This report surveys the field of fault diagnosis and introduces the reader to some of the key algorithms and heuristics currently in use. Fault diagnosis is an important and a rapidly growing discipline. This is important to the Jet Propulsion Laboratory's research efforts in the design of self-repairable computers because the present diagnosis resolution of its fault-tolerant computer is limited to a functional unit or processor. Better resolution is necessary before failed units can become partially reuseable. The approach that holds the greatest promise is that of resident microdiagnostics; however, that presupposes a microprogrammable architecture for the computer being self-diagnosed. The presentation here is tutorial and contains examples. An extensive bibliography of some 220 entries is included.

I. INTRODUCTION

The general field of system testing is an all-encompassing discipline ranging from fabrication processes, parts testing, and production techniques to the final system checkout. This report deals with the area of fault diagnosis as applied to digital computers and systems.

The term fault diagnosis includes both fault detection and fault location, where fault detection is the process of determining whether or not the system is fault-free, whereas fault location is the process of localizing the fault to specific components, sets of components, modules, or subsystems, depending on the diagnostic resolution desired.

Early methods of fault diagnosis in digital computers consisted of diagnostic programs designed to periodically check the major functions of the computer. Often this meant, at best, the testing of the instruction set of the machine using representative sets of data and known corresponding results.

The most commonly used fault model to represent system failures is the "line stuck at 1, line stuck at 0" model. This model assumes that failures are such that any line may become permanently stuck at logical 1 or logical 0 level. This is denoted by s-a-1 or s-a-0, respectively. This model thus includes "lines open" type of failures and also the "shorted lines" type of failures that are such as not to induce feedback as a result of the short.

In general, the problem of fault diagnosis may be succinctly stated thus: By applying signals to the inputs and observing the responses at the outputs, it is determined whether or not the network is operating properly; if not, it must be determined what changes from the norm have occurred.

Exhaustive methods very quickly become unmanageable. If a network consists of n edges, where each edge may be either fault-free, s-a-0, or s-a-1, then the total number of possible "stuck-at" faults amounts to $3^n - 1$ or 3^n faults, including the null fault. If only single faults are considered, then the total number of faults including the null fault is $2n + 1$. As an illustration, a NAND gate realization of an EXCLUSIVE-OR circuit consists of 9 edges; thus, the total number of possible faults that may arise are $3^9 = 19,663$. However, if the single fault assumption is made, the number of faults reduces to $2 \times 3 + 1 = 7$.

In order to make the problem of fault diagnosis more manageable, various simplifying assumptions are made, and algorithms are developed which are applicable to specific classes of digital networks or are more efficient under specific constraints than under others. Typical subclassifications of digital networks indicate whether the network is combinatorial or sequential, and in the latter case whether it is synchronous or asynchronous; whether the network has single or multiple outputs; whether single or multiple faults are considered; whether there is fan-out or no fan-out, and if there is fan-out, whether it is reconvergent or non-reconvergent; whether or not the network is redundant; and, finally, whether the particular technique developed is heuristic or algorithmic.

This report describes Gedanken experiments of faulty machine identification, diagnosing and homing experiments, Armstrong's equivalent normal form test generation method, the Boolean Difference techniques, Roth's D-Algorithms, and the newly evolving branch of microdiagnostics. Finally, an extensive bibliography of fault diagnosis has been compiled.

II. GEDANKEN EXPERIMENTS — SEQUENTIAL MACHINES

The material presented here is based on the work of E. F. Moore entitled "Gedanken — Experiments on Sequential Machines" and also that of A. Gill. In Gedanken experiments the finite-state sequential machine is considered to be a "black box" with inputs to which stimuli can be applied and outputs on which the responses may be observed. Only in this way can the behavior, characterization, and contents of the black box be determined.

III. MACHINE AND FAULT IDENTIFICATION EXPERIMENTS

In the problem of identifying the fault of a machine, the faulty version of the machine is simply regarded as another machine. Thus, the set of all possible failed machines under the class of known faults and the fault-free machine constitutes a class of machines. The experiment is to identify the particular unknown failed machine from this set of machines. A subproblem to the machine identification problem is that of state identification. In the latter case the complete state transition table of the finite-state sequential machine is known, and the problem is to determine its initial state, or, in the case of the homing experiment, to be able to drive it into a known state. In solving problems of fault identification both the machine identification and the state identification techniques are used.

The procedure is as follows. From the known machine M which may malfunction in a set of known ways, the possible faulty machines M_1, M_2, $\cdots$ are characterized by, e.g., specifying their transition tables. These faulty machines are then reduced to their minimal form and constitute an exclusive class of machines $\{M_1, M_2, \cdots, M_n\}$ such that no state in M_i is equivalent to any state in M_j $(j \neq i)$. The disjunction machine ΔM of M_1, M_2, $\cdots$, M_n is then formed. The fault identification problem in the original machine M is now reduced to that of identifying the final state, i.e., conducting a homing experiment on the machine ΔM.

IV. DIAGNOSING AND HOMING EXPERIMENTS

Figure 1 illustrates the theory underlying diagnosing and the allied problem of homing. The pertinent terms are ordered from the simple and obvious to the more complex and unobvious and are defined as follows:

Experiment. The process of applying input sequences to the input terminals of a machine, observing the resulting output sequences, and drawing conclusions based on these observations.

Preset Experiment. An experiment in which the applied input sequence is completely determined in advance.

Adaptive Experiment. An experiment in which the applied input sequence is composed of two or more subsequences, each subsequence (except the first) being determined on the basis of the previous response.

Copy. A machine is a copy of another machine if both have identical transition tables, and if both are at the same state before experiments commence.

Simple Experiments. Experiments in which only one copy of the machine is required.

Multiple Experiments. Experiments in which more than one copy of the machine is required.

Length of an Experiment. The total number of input symbols applied in the course of the experiment.

Order of an Experiment. The number of input sequences of which the experiment is composed.

Multiplicity of an Experiment. The number of copies required of the machine under investigation. (Note: a simple experiment is an experiment of multiplicity = 1; a multiple experiment is an experiment of multiplicity ≥ 2).

Diagnosing Problem. Machine M in one of the states $\{s_1, \cdots, s_m\}$, transition table known, to find this state.

Homing Problem. Machine M in one of the states $\{s_1, \cdots, s_m\}$, transition table known, to pass M into a known state.

Diagnosing Experiment. An experiment that solves the diagnosing problem.

Homing Experiment. An experiment that solves the homing problem. (Note: every diagnosing experiment is also a homing experiment; the converse is not true.)

Admissible Set A(M). Of machine M, the set of states $\{s_1, \cdots, s_m\}$ one of which is the initial state of M.

Admissible States. The states in the admissible set A(M).

m-Wise Diagnosing Problem. The problem of identifying the initial state of M where A(M) is arbitrarily m.

Pair-Wise Diagnosing Problem. The problem of identifying the initial state of M where m = 2.

Diagnosing Sequence. For $\{s_{i0}, s_{j0}\}$ is an input sequence of length n - 1 or less which, when applied to M in s_{i0} and M in s_{j0}, yields distinct output sequences.

Minimal Diagnosing Sequence. $E(s_{i0}, s_{j0})$ for the pair $\{s_{i0}, s_{j0}\}$ in a diagnosing sequence of length ℓ where s_{i0} are ℓ distinguishable and $(\ell - 1)$ equivalent $(1 \leq \ell \leq n - 1)$.

An Example

To find the minimal diagnosing sequence E(1, 2) for the pair of states $\{1, 2\}$ of the machine M whose state diagram is as shown in Fig. 2. First construct the P_k tables. P_k is the partition of S according to the k-equivalence of states in S. By observation of Fig. 2, the state transition table shown in Fig. 3a is derived. We apply the following algorithm:

Algorithm

To determine minimal diagnosing sequences for state pairs.

(1) Construct P_k tables.

Find ℓ such that s_{i0}, s_{j0} are adjoint rows in $P_{\ell-1}$ and disjoint in P_ℓ. Let k = 1.

(2) If $(\ell - k) > 0$ go to step 3.

If $(\ell - k) = 0$ E_k is given the heading of any column in the Z_v subtable of M such that rows s_{ik-1}, s_{jk-1} are distinct. $E_{u1}, E_{u2}, \cdots, E_{uk}$ is the minimal diagnosing sequence for $\{s_{i0}, s_{j0}\}$.

(3) E_{uk} is the heading of any column in $P_{\ell-k}$ such that rows s_{ik-1}, s_{jk-1} in this column have differently subscripted entries. These entries are s_{ik}, s_{jk}.

Increment k by 1 and return to step 2.

Now, having already constructed the P_k tables, we start from the P_3 that is the last table in which rows 1 and 2 are adjoint. Rows 1 and 2 in the P_3 table have distinct subscripts in entries 4c and 5d which appear in column β. β, then, is the first symbol in E(1, 2). In the P_2 table, rows 4 and 5 have

distinct subscripts in entries 3b and 2a which appear in column α. α, then, is the second symbol in E(1, 2). In the P_1 table, rows 3 and 2 have distinct subscripts in entries 5b and 1a, which appear in column α. α, then, is the third symbol in E(1, 2). Alternatively, β could have been chosen as the third symbol, since rows 3 and 2 have distinct subscripts in entries 1a and 5b, which appear in column β. In the Z_v subtable, rows 1 and 5 have distinct entries (0 and 1) in column α. α, then, is the fourth and last symbol in E(1, 2). Thus, (1, 2) is either $\beta\alpha\alpha\alpha$ or $\beta\alpha\beta\alpha$. From the P_0 table or Fig. 2 it can be readily verified that when $\beta\alpha\alpha\alpha$ is applied to M at state 1 and state 2 the last output symbol is 1 and 0, respectively. Consequently, if $\{1, 2\}$ is the admissible set of M, the distinguishing experiment may be conducted by applying $\beta\alpha\alpha\alpha$ and observing the last output symbol. If this symbol is 1, the initial state is 1; if this symbol is 0, the initial state is 2.

The minimal diagnosing sequences for all pairs of states $\{s_{i0}, s_{j0}\}$ in M are listed in Table 1. The last two columns in this table indicate the last output symbols Z_i and Z_j observed when the minimal diagnosing sequence is applied to s_{i0} and s_{j0}, respectively.

V. EQUIVALENT NORMAL FORM (ENF)

The ENF method is that of Armstrong (1966). The equivalent normal form is an equivalent two-level circuit representation that corresponds to a multi-level circuit. It is a test generation method well suited to the selection of efficient (near minimal) test sets. It is easily programmable on a computer; however, it becomes unmanageable for large circuits. The ENF method has not been proved to be algorithmic; however, a counter example has not been discovered.

A. ENF Method

To construct ENF of a combinational circuit with single output:

(1) Write the Boolean expression for the circuit. Corresponding to each gate G_j of the circuit, there shall be a pair of parentheses in the expression, the parentheses being labeled with the subscript j. The subexpression corresponding to G_j shall be en-

closed in parentheses. All literals in the Boolean expression will denote input variables.

(2) Expand to sum-of-products normal form. When a pair of parentheses is removed, its associated label is attached to each literal inside the parentheses as a subscript. No redundant terms are removed.

An example is shown in Fig. 4, where

$$
\begin{aligned}
f &= \left[\left((A \cdot B)_1 + (B \cdot C)_2\right)_3 \cdot D\right]'_4 \\
&= \left[(A_1 \cdot B_1 + B_2 \cdot C_2)_3 \cdot D\right]'_4 \\
&= \left((A_{13} \cdot B_{13} + B_{23} \cdot C_{23}) \cdot D\right)'_4 \\
&= (A_{13} \cdot B_{13} + B_{23} \cdot C_{23})'_4 + D'_4 \\
&= (A_{13} \cdot B_{13})'_4 \cdot (B_{23} \cdot C_{23})'_4 + D'_4 \\
&= (A_{134} + B'_{134}) \cdot (B'_{234} + C'_{234}) + D'_4 \\
&= A'_{134} \cdot B'_{234} + A'_{134} \cdot C'_{234} + B'_{134} \cdot B'_{234} + B'_{134} \cdot C'_{234} + D'_4 \\
&= A'_{\alpha} \cdot B'_{\beta} + A'_{\alpha} \cdot C'_{\beta} + B'_{\alpha} \cdot B'_{\beta} + B'_{\alpha} \cdot C'_{\beta} + D'_{\gamma}
\end{aligned}
$$

which is the ENF sum-of-products form where α, β, and γ are path from inputs to the circuit output. The ENF form is realized by the two-level AND-OR circuit shown in Fig. 5.

B. ENF Theorem

A test for a literal appearance in the ENF sensitizes the corresponding path in the circuit. Thus, if a set of literal appearances can be selected whose corresponding paths together contain every vertex of the circuit, and if a set of tests can be found which tests at least one appearance of each literal for the S-a-1 and S-a-0 faults, then the set of tests detects every fault of the circuit.

For example, in the above the test for a S-a-1 fault for the literal A'_{α} is $A' = 0$, $B' = 0$, $C' = 1$ and $D' = 0$. From the theorem this input vector (0,0,1,0) sensitizes the path α; i.e., G1G3G4 beginning at A.

VI. BOOLEAN DIFFERENCE ALGORITHMS

The Boolean difference method first applied to fault-diagnosis by Sellers et al. (1967) has been described by Akers (1959) and also by Amar and Condulmari (1967). Unlike the tabular or exhaustive search methods described earlier, Boolean difference approach is an elegant systematic equation-solving procedure. Algorithms have been implemented which can handle combinational networks with up to fifty primary input and output variables. Boolean difference techniques have also been extended to generate test patterns for asynchronous sequential circuits.

A. Basic Concept

If $f(x_1, \cdots, x_n)$ is a switching function of the n variables $x_1, \cdots, x_n$ then the Boolean difference is defined as:

$$df(x_1, \cdots, x_n)/dx_i = df(X)/dx_i$$

$$\triangleq f(x_1, \cdots, x_i, \cdots, x_n) \oplus f(x_1, \cdots, \overline{x}_i, \cdots, x_n)$$

which is also equivalent to

$$f(x_1, \cdots, 1, \cdots, x_n) \oplus f(x_1, \cdots, 0, \cdots, x_n)$$

This implies that Boolean difference $df(X)/dx_i$ is a function of (n - 1) variables $x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n$.

From the definition of EXCLUSIVE-OR function it follows that when $df(X)/dx_i = 1$

$$f(x_1, \cdots, 1, \cdots, x_n) \neq f(x_1, \cdots, 0, \cdots, x_n)$$

and when $df(X)/dx_i = 0$

$$f(x_1, \cdots, 1, \cdots, x_n) = f(x_1, \cdots, 0, \cdots, x_n)$$

Thus $df(X)/dx_i = 1$ is a condition for f(X) being dependent on the value of x_i. Which implies that a change at x_i always causes a change at f(X)

when $df(X)/dx_i = 1$, and x_i does not cause a change at f(X) when $df(X)/dx_i = 0$. Hence $df(X)/dx_i = 1$ is a sufficient condition for the path from x_i to F to be sensitized.

In general, $df(X)/dx_i$ is neither 1 nor 0 but some Boolean function not containing x_i; hence, input test patterns may be selected in order to satisfy the condition $df(X)/dx_i = 1$. Now, since $df(X)/dx_i$ is independent of x_i in general, two, i.e., a pair of test patterns, can be selected corresponding to in one case $x_i = 1$ and in the other $x_i = 0$.

The Boolean difference approach may be used to detect inversion-type faults ($1 \rightarrow 0$, $0 \rightarrow 1$) in addition to the conventional "stuck-at" faults.

An example of Boolean difference is shown in Fig. 6. We seek to find the conditions under which an error in x_1 causes an error at the output.

Solution

$$df(X)/dx_1 = d(x_1x_2 + x_3 + x_4)/dx_1$$

using the property

$$d(A + B)/dx_i = \overline{A}\, dB/dx_i \oplus B\, dA/dx_i \oplus dA/dx_i \cdot dB/dx_i$$

$$df(X)/dx_1 = (\overline{x_3 + x_4})\, d(x_1x_2)/dx_1$$

Now, using the property

$$d(A \cdot B)/dx_i = A\, dB/dx_i \oplus B\, dA/dx_i \oplus dA/dx_i \cdot dB/dx_i$$

$$df(X)/dx_1 = (\overline{x_3 + x_4}) \cdot x_2\, dx_1/dx_1$$

which, from the property ($dA/dx_i = 1$ if A depends only on x_i)

$$df(X)/dx_1 = (\overline{x_3 + x_4}) \cdot x_2$$

$$= \overline{x}_3 \cdot \overline{x}_4 \cdot x_2$$

$\overline{x}_3 \cdot \overline{x}_4 \cdot x_2 = 1$ when $x_2 = 1$, $x_3 = 0$ and $x_4 = 0$.

This is the condition when an error in x_1 causes an erroneous output.

The input test patterns for detecting failure at x_1 are (x_1, x_2, x_3, x_4) = (0,1,0,0) and (1,1,0,0).

B. Comments on the Boolean Difference

The astute reader will have observed that the Boolean difference concerns itself with only input variables, and the reader may well ask how does one generate test patterns for "internal" failures? The solution is to break the circuit at that point and consider it as an extra input to the circuit. Having obtained test patterns for this extra input one works backward to specify the test patterns, using only the original circuit inputs. Also, it is possible to form a two-level equivalent circuit of any multi-level circuit (see Subsection V, Equivalent Normal Form). Also, partial Boolean difference, Boolean difference chains, and Boolean difference of multiple variables have been defined. Application of Boolean difference to sequential circuit uses a similar approach to that of the D-algorithm of first modeling the circuit using the Moore model and cutting the feedback lines to reduce the model to an equivalent combinational circuit during delta time intervals. A summary of some basic properties of Boolean difference is found in Table 2.

References on the Boolean difference are found in Bibliography entries 2, 3, 10, 11, 44, 65, 91, 92, 93, 94, 134, 135, 152, 190, 191, and 192.

VII. D-ALGORITHM

The D-algorithm is the first method for generating tests for nonredundant combinational circuits that has been proved to be algorithmic; i.e., if a test exists for detecting a failure, then by the application of the D-algorithm one can find this test. The D-algorithm was formulated by J. P. Roth (see Bibliography entries 169, 170, 178, 180) and is precisely and elegantly expressed in terms of the calculus of cubical complexes. The D-algorithm is a systematic application and extension on the basic concept of path sensitization and is a logical culmination of Eldred's pioneering work. Besides algorithms for generating tests for specific faults Roth's program TESTDETECT solves the related problem of enumerating the set of all faults

that a particular test can detect. Finally the D-algorithm has been extended to apply to asynchronous sequential logic by transforming the problem of finding a single failure in the sequential circuit to that of finding multiple failures in an iterative combinational circuit constructed from the sequential model by the cutting of all its feedback lines. The D-algorithms DALG and DALG II, as well as their extensions: the sequential circuit heuristics, known as iterative test generator (ITG) and macroblock test generator (MTG) have been programmed in the APL language.

It should be noted that since the D-algorithm guarantees obtaining tests for non-redundant combinational networks then by implication the D-algorithm may also be used to detect redundancy of design in combinational circuits for which complete tests are not generatable.

A. D-Algorithm Definitions

Singular cover is a rearranged compact form representation of a truth-table (an x is used to denote that the corresponding variable may be a 1 or 0).

Singular cube rows of singular covers are termed singular cubes.

D-cubes represent the input-output behavior of the failing as well as the good circuit.

Primitive D-cubes of a logic block (element) is a subset of D-cubes defined as those D-cubes having a D or $\overline{D}$ on the output coordinate.

Primitive D-cube of failure is the particular primitive D-cube of a failing block. These are determined by both the logic function and the assumed failure of the block.

D-drive is the operation of constructing a D-path or set of D-paths (sensitized path) from the failed block to a primary output.

D-frontier consists of all the blocks in the circuit having D's or $\overline{D}$'s on some of their inputs and X's on their outputs.

Consistency operation is the justification of all 0, 1 values imposed during D-drive in terms of primary-input values.

D-signals are 5-valued logic $(0, 1, X, D, \overline{D})$ used to describe the behavior of a circuit with failures. The symbol D ($\overline{D}$) denotes a logical value which is 1 (0) in the good circuit and 0 (1) in the circuit with failure.

B. Algorithm to Form Primitive D-cubes From Singular Covers

Step 1. Intersect cubes (rows) of the singular cover, these cubes must have different output values.

Step 2. The intersection rules of the input values are:

$1 \cap 0 = \overline{D}$

$0 \cap 1 = D$

$X \cap X = X$

$0 \cap 0 = 0 \qquad X = X \cap 0 = 0$

$1 \cap 1 = 1 \qquad X = X \cap 1 = 1$

C. Algorithm to Form Primitive D-cubes of Failure

Step 1. Use cubes of the singular cover.

Step 2. Intersect pairs of cubes.

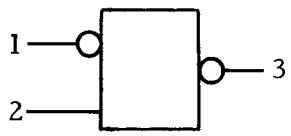
Step 3. Use same intersection rules as the algorithm to form primitive D-cubes.

Step 4. Select one cube of each pair from the singular cover, the other member is the corresponding cube from the singular cover of the failed gate.

Step 5. Assign bars by the following convention: A vertex of the intersection have value $\overline{D}$ if the vertex has value 0 in the cube of the failure-free block, and value 1 in the cube of the failed block. All other cases are assigned values D.

Example

Given the truth table for the INHIBIT-NOT logical block:



1	2	3
0	0	1
0	1	0
1	0	1
1	1	1

(1) To determine its singular cover:

Singular cover 1	2	3
0	1	0
X	0	1
1	X	1

(2) To determine the primitive D-cubes:

Primitive D-cube 1	2	3	its dual	1	2	3
D	1	D		$\overline{D}$	1	$\overline{D}$
0	D	$\overline{D}$		0	$\overline{D}$	D
D	$\overline{D}$	D		$\overline{D}$	D	$\overline{D}$

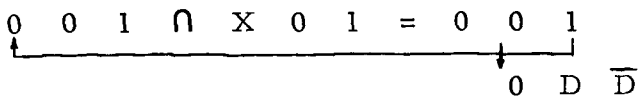
This is determined from the 0-cube 0 1 0 in the singular cover, which is used as the generator. Generators have the property that a change in a single input must force a change in the output. A change in either input or block inputs causes a change in the output.

A change in input 1:

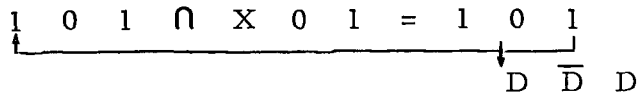
$$1\ 1\ 1 \cap 1\ X\ 1 = 1\ 1\ 1$$

$$D\ 1\ D$$

A change in input 2:



A change in inputs 1 and 2:



D. Circuit-Labeling Convention

(1) Assign an integer label to each vertex of the circuit.

(2) Designate gates (blocks) by the label of the vertex corresponding to its output. Note: each vertex corresponds to either a primary input signal or a gate output signal.

(3) Assign vertex number by using the "leveling rule" which is that the integer associated with a block shall be greater than the integers of all vertices which feed it.

E. D-Algorithm (DALG-II)

Start by applying a primitive D-cube of failure at the failing block (logic element), thus obtaining a D or $\overline{D}$ at its output.

(The goal is to construct a D-path (sensitized path) or set of D-paths, from the failed block to a primary output, an operation called D-drive.)

Initially the D-frontier (defined to consist of all the blocks in the circuit having D's or $\overline{D}$'s on some of their inputs and X's on their outputs) consist of the successors of the failed block. The D-frontier is then moved toward the primary outputs by applying suitable primitive D-cubes to the blocks in the D-frontier thus advancing the D-frontier itself, until a D or $\overline{D}$ has been imposed on a primary output. At this point, a connected D-path has been established from the failed block to the primary output, and the D-drive is terminated.

Now, all 0,1 values imposed during D-drive have to be justified in terms of the primary-input values. This is accomplished by the consistency operation; i.e., iteratively applying suitable singular cubes to all blocks having output values not justified by their inputs, if this is possible, until all 0,1 signals have been driven back to primary inputs. An example is shown in Fig. 8.

To construct a test for "line 7 s-a-0," first the primitive D-cube of failure 1 1 D is used to impose values 1 1 D on lines 1, 2 and 7. The primitive D-cube D 0 D is then applied, thus driving a D through block 9. As a consequence, a 0 must be imposed on line 3. The D on line 9 is now driven to the primary-output 12 by applying to lines 9, 10 and 12 the primitive D-cube D 0 $\overline{D}$. This completes the D-drive. The consistency operation now amounts to justifying the 0 value on line 10. This is done by imposing a 0 on line 4. The obtained test for "7 s-a-0" is 1100xx.

References on D-algorithms are found in Bibliography entries 18, 19, 20, 39, 43, 65, 67, 154, 161, 169, 170, 171, 173, 174, 175, 176, 177, 178, 180, and 186.

VIII. MICRODIAGNOSTICS

Microprogram routines stored in read-only memories, in contrast to the traditional control unit (CPU) of the classical computer, act as the master control governing at the gate level the data flow of the machine.

The traditional machine language diagnostics consisting of function tests, measurements tests, utility programs, and diagnostic monitors apply at the level of machine instructions and data words. However, microdiagnostics or diagnostic microprograms have a much finer resolution in that the data paths are controlled at the gate level thus enabling the application of diagnostic tests at the circuit level. Another advantage is that the amount of hard core (or unaccessible circuitry) is drastically reduced.

An example of the application of microdiagnostics is the IBM System 360 model 85. Its diagnostic routines are written in a microinstruction language and are executed out of a read-and-write form of control storage.

They claim that 85% of all failures in the model 30 CPU are fault-locatable to four or less SLT cards.

In the author's view, the approach that holds the greatest promise for the diagnosis of closed (inaccessible to human intervention) computer system is that of resident microdiagnostic; however, that presupposes a microprogrammable architecture of the computer being self-diagnosed.

References on microdiagnostics are found in Bibliography entries 8, 31, 79, and 96.

BIBLIOGRAPHY

1. Abdali, S. K., "On Proving Sequential Machines," IEEE Trans. Comput., Vol. C-20, No. 12, pp. 1563-1566, Dec. 1971.

2. Akers, S. B., "On a Theory of Boolean Functions," J. SIAM, Vol. 7, No. 4, pp. 487-498, Dec. 1959.

3. Amar, V. and Condulmari, N., "Diagnosis of Large Combinational Networks," IEEE Trans. Electron. Comput., Vol. EC-16, No. 5, pp. 675-680, Oct. 1967.

4. Armstrong, D. B., "On Finding a Nearly Minimal Set of Fault Detection Tests for Combinational Logic Nets," IEEE Trans. Electron. Comput., Vol. EC-15, No. 1, pp. 66-73, Feb. 1966.

5. Arnold, F. T., "Automatic Testing of Wiring for Circuit Card Panels," in IEEE Proceedings of the 1963 Joint Automatic Control Conference, pp. 49-53, held at University of Minnesota, Minneapolis, June 19-21, 1963.

6. Ashenhurst, R. L., "The Decomposition of Switching Functions," in Proceedings at the International Symposium on Theory of Switching, pp. 74-116, Harvard Univ., Cambridge, Mass., 1957.

7. Barron, D. W., and Hartley, D. F., "Techniques for Program Error Diagnosis on EDSAC 2," Comput. J., Vol. 6, pp. 44-49, 1963.

8. Bartow, N., and McGuire, R., "System/360 Model 85 Microdiagnostics," AFIPS Proceedings of the Spring Joint Computer Conference, pp. 191-197, Atlantic City, May, 1970.

9. Bashkow, T. R., Friets, J., and Karson, A., "A Programming System for Detection and Diagnosis of Machine Malfunctions," IEEE Trans. Electron. Comput., Vol. EC-12, No. 1, pp. 10-17, 1963.

10. Bearnson, L. W., and Carroll, C. C., "On the Design of Minimum Length Fault Tests for Combinational Circuits," Digest of the International Symposium on Fault-Tolerant Computing, pp. 1-4, March 1971. (See also IEEE Trans. Comput., Vol. C-20 No. 11, pp. 1353-1361, Nov. 1971.)

11. Beebe, P. L., "Evaluation of an Automatic Test and Checkout System," Proceedings of the 2nd Hawaii International Conference on System Sciences, pp. 897-900, University of Hawaii, 1969.

12. Bennetts, R. G., and Lewin, D. W., "Fault Diagnosis of Digital Systems—A Review," Computer, Vol. 4, No. 4, pp. 12-21, July/August 1971.

13. Betancourt, R., "Derivation of Minimum Test Sets for Unate Logical Circuits," IEEE Trans. Comput., Vol. C-20, No. 11, pp. 1264-1269, Nov. 1971.

14. Bogomolov, A. M., and Tvedokhebov, V. A., "Conditions for the Existence of Diagnostic Tests for Complex Systems," Proceedings of the 1968 International Federation of Information Processing Congress, Edinburgh, Vol. 1, pp. 280-282, 1968.

15. Bossen, D. C., and Hong, S. J., "Cause-Effect Analysis for Multiple Fault Detection in Combinational Networks," IBM TR-00.2161, Poughkeepsie Lab., New York, Jan. 25, 1971. (See also Digest of the International Symposium on Fault Finding, pp. 40-43, March 1971.)

16. Bossen, D. C., and Hong, S. J., "Cause-Effect Analysis for Multiple Fault Detection in Combinational Networks," IEEE Trans. Comput., Vol. C-20, No. 11, pp. 1252-1257, Nov. 1971.

17. Bossen, D. C., Ostapko, D. L., and Patel, A. M., "Optimum Test Patterns for Parity Networks," AFIPS Conference Proceedings of the Fall Joint Computer Conference, Vol. 37, pp. 63-68, 1970.

18. Bouricius, W. G., et al., "Interactive Design of Self-Testing Circuitry," Purdue Univ. Centennial Symposium on Information Processing, pp. 73-80, April 1969.

19. Bouricius, W. G., et al., "Algorithms for Detection of Faults in Logic Circuits," IEEE Trans. Comput., Vol. C-20, No. 11, pp. 1258-1263, Nov. 1971.

20. Bouricius, W. G., et al., "Interactive Design of Self-Testing Circuitry," IBM Research Report, RC-2444, April 18, 1969.

21. Breslow, D. H., "Automatic Fault Location Using Building-Block Logic," Proceedings of the 6th Annual National Symposium on Reliability and Quality Control, pp. 449-458, 1960.

22. Breuer, M., "Generation of Fault Detection Tests for Sequential Circuits," Digest of the International Symposium of Fault-Tolerant Computing, pp. 18-21, March 1971.

23. Breuer, M. A., "Functional Partioning and Simulation of Digital Circuits," IEEE Trans. Comput., Vol. C-19, No. 11, pp. 1038-1046, Nov. 1970.

24. Breuer, M. A., "General Survey of Design Automation of Digital Computers," Proceedings of the IEEE, Vol. 54, No. 12, December 1966.

25. Breuer, M. A., "Fault-Detection in a Linear Cascade of Identical Machines," Proceedings of the IEEE 9th Annual Symposium on Switching and Automata Theory, 68-C-50C, pp. 235-243, 1968.

26. Breuer, M. A., "A Random and an Algorithmic Technique for Fault Detection Test Generation for Sequential Circuits," IEEE Trans. Comput., Vol. C-20, No. 11, pp. 1364-1370, Nov. 1971.

27. Breuer, M. A., "Generation of Fault Tests for Linear Logic Networks," IEEE Trans. Comput., Vol. C-21, No. 1, pp. 79-83, Jan. 1972.

28. Brown, F. D., McAllister, N. F., and Perry, R. P., "An Application of Inverse Probability to Fault Isolation," IRE Trans. Mil. Electron., Vol. MIL-6, No. 3, pp. 260-267, 1962.

29. Brule, J. D., Johnson, R. A., and Kletsky, E. J., "Diagnosis of Equipment Failures," IRE Trans. Reliabil. Cont., Vol. RQC-9, pp. 23-34, April 1960.

30. Buyanov, B. B., Domanitskiy, S. M., and Ozernoy, V. M., "Tests for Logical Systems of Unifunctional Elements," Engineering Cybernetics, No. 2, pp. 83-92, 1966.

31. Calhoun, R. C., "Diagnostics at the Microprogramming Level," Modern Data, p. 58, May 1969.

32. Caroll, A. B., et al., "A Method of Diagnostic Test Generation," AFIPS Conference Proceedings, Vol. 34, 53CC, pp. 221-228, Boston, May 1969.

33. Carroll, B. D., "A Tabular Method for Generating Test Sequences for Sequential Networks," Proceedings of the Fifth Hawaii International Conference on System Sciences, pp. 290-292, University of Hawaii, 1972.

34. Chan, S. P., "On Fault Detection and Isolation in Linear Networks and Systems," Proceedings of the 2nd Hawaii International Conference on System Sciences, pp. 889-892, University of Hawaii, 1969.

35. Chang, H. Y., "A Distinguishability Criterion for Selecting Efficient Diagnostic Tests," AFIPS Conference Proceedings of the Spring Joint Computer Conference, Vol. 3, pp. 529-534, Atlantic City, May 1968.

36. Chang, H. Y., "An Algorithm for Selecting an Optimum Set of Diagnostic Tests," IEEE Trans. Electron. Comput., Vol. EC-14, No. 5, pp. 706-711, Oct. 1965.

37. Chang, M. Y., "Techniques for Diagnosity Faults in Switching Systems," Purdue Univ. Centennial Symposium on Information Processing, pp. 60-72, April 1969.

38. Chang, H. Y., "Figures of Merit for the Diagnostics of a Digital System," IEEE Trans. Reliab., Vol. R-17, No. 3, pp. 147-152, 1968.

39. Chang, H. Y., Manning, E., and Metze, G., "Fault Diagnosis of Digital Systems," p. 159. John Wiley & Sons, Inc., New York, 1970.

40. Chang, H. Y., and Thomas, W., "Methods of Interpreting Diagnostic Data for Locating Faults in Digital Machines," Bell System Technical Journal, Vol. 46, No. 2, pp. 289-317, 1967.

41. Charayev, G. G., "Fault Diagnosis for Logic Devices Using Three-Input Elements," Engineer. Cyb., No. 4, pp. 718-723, 1970.

42. Charayev, G. G., "Testing of Operative State and Fault Diagnosis in an Incompletely Homogeneous Two-Dimensional Structure," Automat. Remote Cont., Vol. 29, No. 7, pp. 1130-1137, 1968.

43. Chiang, A. C. L., Reed, I. S., and Banes, A. V., "Path Sensitization, Partial Boolean Difference, and Automated Fault Diagnosis," IEEE Trans. Comput., Vol. C-21, No. 2, pp. 189-195, Feb. 1972.

44. Chu, W. W., "Adaptive Diagnosis of Faulty Systems," Oper. Res., Vol. 16, No. 5, pp. 915-927, 1968.

45. Chu, W. W., "Some Recent Results on Diagnosis of System Failures," presented at the XVI General Assembly of International Scientific Radio Union, Ottawa, Canada, August 18-29, 1969.

46. Cioffi, G., and Fiorillo, E., "Diagnosis and Utilization of Faulty Universal Tree Circuits," Proceedings of the AFIPS Spring Joint Computer Conference, Vol. 34, pp. 139-147, 1969.

47. Clegg, F. W., The SPOOF: a New Technique for Analyzing the Effects of Faults on Logic Networks, Technical Report No. 11, SU-SEL-70-073, Stanford University, Digital Systems Laboratory, August 1970.

48. Clegg, F. W., and McCluskey, E. J., Algebraic Properties of Faults in Logic Networks, Technical Report no. 4, SU-SEL-69-078, Stanford University, Digital Systems Laboratory, March 1970.

49. Clegg, F. W., and McCluskey, E. J., "The Algebraic Approach to Faulty Logic Networks," Digest of the International Symposium on Fault-Tolerant Computing, pp. 44-45, March 1971.

50. Cohen, D. J., and Manning, E. G., A Fault Simulator for Research Applications, Research Report CSRR 2012, Univ. of Waterloo, Dept. of Applied Analysis and Computer Science, Waterloo, Ontario, July 1969.

51. Correia, M., Ferguson, D. K., and Millham, E. A., "Production Circuit Array Test System," Digest of the 1968 IEEE Computer Conference, pp. 81-82, 1968.

52. DeAtley, E., "LSI Testing Is a Large-Scale Headache," Electron. Design, Vol. 16, pp. 24-34, July-Sept. 1969.

53. Dent, J. J., "Diagnostic Engineering Requirements," AFIPS Conference Proceedings of the Spring Joint Computer Conference, Vol. 32, pp. 503-508, 1968.

54. Droulette, D. L., "Recovery Through Programming System/360-System/370, AFIPS Conference Proceedings of the Spring Joint Computer Conference, Vol. 38, pp. 467-476, 1971.

55. Duke, K. A., Schnurmann, M. D., and Wilson, T. I., "System Validation by Three-Level Modeling Synthesis," IBM J. Res. Develop., March 1971.

56. Eckert, J. P., Jr., "Checking Circuits and Diagnostic Routines," IRE National Convention Record, Pt. 7, pp. 62-65, 1953.

57. Eichelberger, E. B., "Hazard Detection in Combinational and Sequential Switching Circuits," in Proceedings of the 5th Annual Symposium on Switching Circuit Theory and Logical Design, Princeton University, N. J., November 1964.

58. Eldred, R. D., "Effect of Microelectronics on Error Diagnosis," Computer Design, Vol. 6, No. 11, pp. 6-8, 1967.

59. Eldred, R. D., "Test Routines Based on Symbolic Logic Elements," J. ACM, Vol. 6, No. 1, pp. 33-36, Jan. 1959.

60. Estrin, G., "Diagnosis and Prediction of Malfunctions in the Computing Machine at the Institute for Advanced Study," IRE National Convention Record, Pt. 7, pp. 59-61, 1953.

61. Farmer, D. E., "A Strategy for Detecting Faults in Sequential Machines Not Possessing Distinguishing Sequences," AFIPS Conference Proceedings of the Fall Joint Computer Conference, Vol. 37, pp. 493-501, 1970.

62. Flomenhoft, M. J., "A System of Computer Aids for Designing Logic Circuit Tests," Proceedings of the 7th Design Automation Workshop, pp. 128-132, IBM Custom Systems, Poughkeepsie, N. Y., 1970.

63. Friedman, A. D., "Fault Detection in Redundant Circuits," IEEE Trans. Electron. Comput., Vol. EC-16, No. 1, pp. 99-100, Feb. 1967.

64. Friedman, A., and Menon, P., "Fault Detection in Digital Circuits," Prentice-Hall, New York, 1971.

65. Friedman, E., DALG — A Program for Test Pattern Generation in Combinational Logical Circuits, Technical Memorandum 33-516, Jet Propulsion Laboratory, Pasadena, Calif., Nov. 15, 1971.

66. Furia, N., "Fault Detection and Isolation of Malfunctions in Digital Subsystems," Proceedings of the 2nd Hawaii International Conference on System Sciences, pp. 885-888, University of Hawaii, 1969.

67. Galey, J. M., Norby, R. E., and Roth, J. P., "Techniques for the Diagnosis of Switching Circuit Failures," AIEE Proceedings of the 2nd Annual Symposium on Switching Circuit Theory and Logical Design, pp. 152-160, Detroit, October 1961. (See also IEEE Trans. Commun. Electron. Vol. 33, No. 74, pp. 509-514, Sept. 1964.)

68. Gault, J. W., Robinson, J. P., and Reddy, S. M., "Multiple Fault Detection in Combinational Networks," IEEE Trans. Comput. Vol. C-21, No. 1, pp. 31-36, Jan. 1972.

69. Gill, A., "Introduction to the Theory of Finite State Machines," pp. 142 ff, McGraw-Hill Book Co., Inc., New York, 1962.

70. Goetz, F. M., "Computer Aided Diagnostic Design for Electronic Switching Systems," Proceedings of the 7th Design Automation Workshop, IBM Custom Systems, Poughkeepsie, N. Y., pp. 178-190, 1970.

71. Goldberg, J., Levitt, K. N., and Short, R. A., Techniques for the Realization of Ultra-reliable Spaceborne Computers, Final Report, Phase I, Contract NAS 12-33, SRI Project 5580, Stanford Research Institute, Menlo Park, Calif., 1966.

72. Gönenc, G., "A Method for the Design of Fault Detection Experiments," IEEE Trans. Comput., Vol. C-19, No. 6, pp. 551-558, June 1970.

73. Gorovoi, V. R., "Synthesis of Optimal Switching Circuits With Simultaneous Construction of Test Tables," Automat. Remote Cont., Vol. 29, No. 3, pp. 481-490, 1968.

74. Gorskii, Y. M., and Novorusskiy, V. V., "A Procedure for Probabalistic Evaluation of the Reliability Hypotheses in Complex Diagnostic Problems," Engin. Cyb., No. 5, pp. 110-116, 1967.

75. Gould, J. S., "On Automatic Testing of On-line Real-Time Systems," AFIPS Conference, Proceedings of the Spring Joint Computer Conference, Vol. 38, pp. 477-484, 1971.

76. Gray, F., and Meyer, J. F., "Locatability of Faults in Abstract Networks," Digest of the International Symposium on Fault-Tolerant Computing, pp. 30-33, March 1971.

77. Gray, F. G., and Meyer, J. F., "Locatability of Faults in Combinational Networks," IEEE Trans. Comput., Vol. C-20, No. 11, pp. 1407-1412, Nov. 1971.

78. Groenig, S. R., Maki, G. K., and Swain, D. H. III, "Self-Fault Detecting Asynchronous Sequential Circuits," Proceedings of the Fifth Hawaii International Conference on System Sciences, pp. 280-282, University of Hawaii, 1972.

79. Guffin, R. M., "Microdiagnostics for the Standard Computer MLP-900 Processor," IEEE Trans. Comput., Vol. C-20, No. 7, pp. 803-808, July 1971.

80. Hadlock, F., "On Finding a Minimal Set of Diagnostic Tests," IEEE Trans. Electron. Comput., Vol. EC-16, No. 5, pp. 674-675, Oct. 1967.

81. Hamming, W. R., "Error Detecting and Error Correcting Codes," Bell System Technical Journal, Vol. xxix, No. 2, pp. 147-160, April 1950.

82. Hanne, J. R., and Jennings, R. M., "User Assurances in LSI Discretionary Routine," Digest of the 1968 IEEE Computer Conference, pp. 27-28, 1968.

83. Happ, W. W., and Sarkisian, E., "Combinational Techniques for Fault Identification in Multi-terminal Networks," Proceedings of the 1968 Annual Symposium on Reliability, Vol. 1, No. 1, pp. 477-485. IEEE Catalog 7C50, Jan. 1968.

84. Hardie, F. H., and Schocki, R. J., "Design and Use of Fault Simulation for Saturn Computer Design," IEEE Trans. Electron. Comput., Vol. EC-16, No. 4, pp. 412-429, August 1967.

85. Hennie, F. C., "Fault Detecting Experiments for Sequential Circuits," IEEE Proceedings of the 5th Annual Symposium on Switching Circuit Theory and Logical Design, pp. 95-110, Princeton Univ., N. J., November 1964.

86. Hill, F. J., and Meyer, C. S., "Interaction With a Simulation for the Determination of Fault-Detection Sequences for LSI Circuits," Proceedings of the IEEE Conference on Resources Roundup, IEEE Publication 69C-12-REG6, pp. 251-257, April 1969.

87. Hornbuckle, G. D., and Spann, R. N., "Diagnosis of Single-Gate Failures in Combinational Circuits," IEEE Trans. Comput., Vol. C-18, No. 3, pp. 216-220, March 1969.

88. Hornfeck, W. A., and Carroll, C. C., "Algorithms for Generating Complete and Minimum Test Schedules for Combinational Networks," Proceedings of the Fifth Hawaii International Conference on System Sciences, pp. 296-299, University of Hawaii, 1972.

89. Hsiao, M. Y., and Chia, D. K., Boolean Difference for Automatic Test Pattern Generation, IBM TR-00.2149, Poughkeepsie Lab, New York, Dec. 29, 1970.

90. Hsiao, M. Y., and Chia, D. K., "Boolean Difference for Fault-Detection in Asynchronous Sequential Machines," Digest of the International Symposium on Fault-Tolerant Computing, pp. 9-13, March 1971.

91. Hsiao, M. Y., and Chia, D. K., "Boolean Difference for Fault-Detection in Asynchronous Sequential Machines," IEEE Trans. Comput., Vol. C-20, No. 11, pp. 1356-1361, Nov. 1971.

92. Hsiao, M. Y., Sellers, F. F., Chia, D. K., "Fundamentals of Boolean Difference for Test Pattern Generation," Proceedings of the Fourth Annual Princeton Conference on Information Science and Systems, Princeton University, N. J., March 1970.

93. Ichikawa, T., and Watanabe, T., "A Systematic Method of Finding Diagnostic Test Functions," Electron. Commun. Japan, Vol. 52-C, No. 4, pp. 165-172, 1969.

94. Jelinek, H. J., "A Technique for Fault Detection and Isolation in Continuous Networks," Proceedings of the Fifth Hawaii International Conference on System Sciences, pp. 286-289, University of Hawaii, 1972.

95. Jelinek, K. J., "System Fault Diagnosis," Proceedings of the 2nd Hawaii International Conference on System Sciences, pp. 873-876, University of Hawaii, 1969.

96. Johnson, A. M., "The Microdiagnostics for the IBM System 360 Model 30," IEEE Trans. Comput., Vol. C-20, No. 7, pp. 798-803, July 1971.

97. Johnson, R. A., "An Information Theory Approach to Diagnosis," Proceedings of the 6th Symposium on Reliability and Quality Control, pp. 102-109, 1960.

98. Johnson, R. A., Kletsky, E., and Brule, J., Diagnosis of Equipment Failures, Technical Report I, AD-213876, Syracuse University Research Institute, 1960.

99. Johnson Jr., W. R., "Proving Out Large PC Boards: Which System Is Best for You?" Electronics, Vol. 44, No. 6, March 15, 1971.

100. Jones, E. R., and Mays, C. H., "Automatic Test Generation Methods for Large Scale Integrated Logic," IEEE J. Sol. State Circ., SC-2 pp. 221-226, 1967.

101. Kajitani, K., Tezuka, Y., and Kasahara, Y., "Diagnosis of Multiple Faults in Combinational Circuits," Electron. Communic. Japan, Vol. 52-C, No. 4, pp. 123-131, 1969.

102. Karibski, V. V., Parkhomenko, P. P., and Sogomonian, E. S., "Diagnostics of Combinational Circuits," Proceedings of the Third Congress of International Federation of Automatic Control, Paper No. 35D, London, June 1966.

103. Kautz, W. H., "Automatic Fault Detection in Combinational Switching Networks," Proceedings of the 2nd Annual Symposium on Switching Circuit Theory and Logical Design, Detroit, pp. 195-214, Oct. 1961.

104. Kautz, W. H., Bibliography on Fault Testing and Diagnosis in Digital Logic Circuitry With Addendum I, II, and III, Stanford Research Institute, Menlo Park, Calif., December 1970.

105. Kautz, W. H., "Fault Diagnosis in Combinational Digital Circuits," Digest of the First Annual IEEE Computer Conference, pp. 2-5, Sept. 1967.

106. Kautz, W. H., "Fault Testing and Diagnosis in Combinational Digital Circuits," IEEE Trans. Comput., Vol. C-17, No. 4, pp. 352-366, April 1968.

107. Kautz, W. H., "Testing for Faults in Combinational Cellular Logic Arrays," Proceedings of the Eighth Annual Symposium on Switching and Automata Theory, Austin, pp. 161-174, 1967.

108. Kella, J., "Sequential Machine Identification," IEEE Trans. Comput., Vol. C-20, No. 3, pp. 332-338, March 1971.

109. Kime, C. R., "An Analysis Model for Digital System Diagnosis," IEEE Trans. Comput., Vol. C-19, No. 11, pp. 1063-1073, November 1970.

110. Kime, C. R., "A Diagnosability Analysis Model for Digital Systems," Proceedings of the 2nd Hawaii International Conference on System Sciences, pp. 881-884, University of Hawaii, 1969.

112. Kime, C. R., and Ellenbecker, M. J., "The Generalized Fault-Table and Its Use in Diagnostic Test Selection," Proceedings of the IEEE National Electronics Conference, Vol. 25, pp. 663-667, 1970.

113. Kletsky, E. J., "An Application of the Information Theory Approach to Failure Diagnosis," IRE Trans. Reliab. Qual. Cont., RQC-9, pp. 29-39, 1960.

114. Knowlton, K. C., "A Combination Hardware-Software Debugging Systems," IEEE Trans. Comput., Vol. C-17, No. 1, pp. 84-86, January 1968.

115. Koga, Y., "A Checking Method of Wiring," Proceedings of the 7th Design Automation Workshop, IBM Custom Systems, Poughkeepsie, N.Y., pp. 173-178, 1970.

116. Koga, Y., Chen, C., and Naemura, K., "A Method of Test Generation for Fault Location in Combinational Logic," AFIPS Proceedings of Fall Joint Computer Conference, Vol. 37, pp. 69-78, 1970.

117. Kohavi, I., and Kohavi, Z., "Variable-Length Distinguishing Sequence and Their Application to the Design of Fault-Detection Experiments," IEEE Trans. Comput., Vol. C-17, No. 8, pp. 792-795, August 1968.

118. Kohavi, Z., and Kohavi, I., "New Techniques for the Design of Fault-Detection Experiments for Sequential Machines," Proceedings of Hawaii International Conference on System Sciences, University of Hawaii, January 1968.

119. Kohavi, Z., and Lavallee, P., "Design of Diagnosable Sequential Machines," AFIPS Proceedings of the Spring Joint Computer Conference, Vol. 30, pp. 713-718, 1967.

120. Kohavi, Z., and Lavallee, P., "Design of Sequential Machines With Fault-Detection Capabilities," IEEE Trans. Electr. Comput., Vol. EC-16, No. 4, pp. 473-484, August 1967.

121. Kohavi, Z., and Spires, D. A., "Designing Sets of Fault-Detection Tests for Combinational Logic Circuits," IEEE Trans. Comput., Vol. C-20, No. 12, pp. 1463-1468, December 1971.

122. Kriz, T. A., "A Path Sensitizing Algorithm for Diagnosis of Binary Sequential Logic," Proceedings of 1970 IEEE International Computer Group Conference, Washington, D.C., pp. 250-259, June 1970.

123. Kriz, T. A., "Machine Identification Concepts of Path Sensitizing Fault Diagnosis," IEEE Conference Record of the Tenth Annual Symposium on Switching and Automata Theory, pp. 174-181, October 1969.

124. Kruskal, J. B., and Hart, R. E., "A Geometric Interpretation of Diagnostic Data From a Digital Machine: Based on a Study of the Morris, Illinois, Electronic Central Office," Bell System Technical Journal, pp. 1299-1338, October 1966.

125. Landgraff, R. W., "Design of Diagnosable Iterative Arrays," IEEE Trans. Comput., Vol. C-20, No. 8, pp. 867-877, August 1971.

126. Larsen, R. W., and Reed, I. S., "Redundancy by Coding Versus Redundancy by Replication for Failure-Tolerant Sequential Circuits," IEEE Trans. Comput., Vol. C-21, No. 2, pp. 130-136, February 1972.

127. Lee, S. C., and Lee, E. T., "Fault Diagnosis of Neural Networks," Record of the 21st Annual Southwestern IEEE Conference, Texas, Catalog No. 71C17-SWIECO, pp. 423-429, April 1971.

128. Mandelbaum, D., "A Measure of Efficiency of Diagnostic Tests Upon Sequential Logic," IEEE Trans. Electron. Comput., Vol. EC-13, No. 5, October 1964.

129. Manning, E., "On Computer Self-Diagnosis, Part I-Experimental Study of a Processor," IEEE Trans. Electron. Comput. Vol. EC-15, No. 6, pp. 873-881, December 1966.

130. Manning, E., "On Computer Self-Diagnosis, Part II-Generalizations and Design Principles," IEEE Trans. Electron. Comput., Vol. EC-15, No. 6, pp. 882-890, Dec. 1966.

131. Manning, E. G., and Chang, H. Y., "A Comparison of Fault Simulation Methods for Digital Systems," Digest of the 1st Annual IEEE Computer Conference, pp. 10-13, Chicago, September 1967.

132. Marinos, P. N., "A Method of Deriving Minimal Complete Sets of Test Input Sequences Using Boolean Differences," Proceedings 1970 IEEE International Computer Group Conference, pp. 240-246, 1970.

133. Marinos, P. N., "Derivation of Minimal Complete Sets of Test-Input Sequences Using Boolean Differences," IEEE Trans. Comput., Vol. C-20, No. 1, pp. 25-32, January 1971.

134. Marlett, R. A., and Fok, C., "Failure Mode Analysis and Pattern Generation," Digest of the IEEE International Convention, New York, March 22-25, 1971.

135. Martin, R. L., "The Design of Diagnosable Sequential Machines," Proceedings of Hawaii International Conference on System Science, pp. 619-622, University of Hawaii, January 1968.

136. Mayeda, W., and Ramamoorthy, C. V., "Distinguishability Criteria in Oriented Graphs and Their Application to Computer Diagnostics I," IEEE Trans. Circuit Theory, Vol. CT-16, No. 4, pp. 448-454, Nov. 1969.

137. McAleer, H. T., "A Look at Automatic Testing," IEEE Spectrum, Vol. 8, No. 5, pp. 63-78, May 1971.

138. McCluskey, E. J., "Test and Diagnosis Procedure for Digital Networks," Computer, Vol. 4, No. 1, January-February, 1971.

139. McCluskey, E. J., Jr., "Introduction to the Theory of Switching Circuits," McGraw-Hill Book Co., Inc., New York, 1965.

140. McCluskey, E. J., and Clegg, F. W., "Fault Equivalence in Combinational Logic Networks," Technical Note No. 10, Stanford University, Digital System Laboratory, Stanford, Calif., March 1971.

141. McCluskey, E. J., and Clegg, F. W., "Fault Equivalence in Combinational Logic Networks," IEEE Trans. Comput., Vol. C-20, No. 11, pp. 1286-1293, November 1971.

142. Mei, K. C. Y., Fault Dominance in Combinatorial Circuits, Technical Note No. 2, Stanford University, Digital Systems Laboratory, August 1970.

143. Menon, P. R., and Friedman, A. D., "Fault Detection in Iterative Logic Arrays," Proceedings of the 2nd Hawaii International Conference on System Sciences, pp. 893-896, University of Hawaii, 1969. (See also IEEE Trans. Comput., Vol. C-20, No. 5, May 1971).

144. Meyer, J. F., and Yeh, K., "Diagnosable Machine Realizations of Sequential Behavior," Digest of the International Symposium on Fault-Tolerant Computing, pp. 22-25, March 1971.

145. Meyers, E. F., "Diagnosis and Testing, Aerospace Digital Computers," Digest of the International Symposium on Fault-Tolerant Computing, pp. 55-59, March 1971.

146. Miller, R. E., Switching Theory, Vol. I, John Wiley & Sons, Inc., New York, 1965.

147. Moore, E. F., "Gedanken Experiments on Sequential Machines," Automata Studies, pp. 129-153, Princeton University Press, N. Y., 1956.

148. Murakami, S., Kinoshita, K., and Ozaki, H., "Sequential Machines Capable of Fault Diagnosis," Trans. on Comput., Vol. C-19, No. 11, pp. 1079-1085, November 1970.

149. Myers, H. J., and Hsiao, M. Y., "An APL Algorithm for Calculating Boolean Difference," IEEE Maintainability Conference, St. Louis, Missouri, November 1968.

150. Neishtadt, I. S., "Method for the Construction of Diagnostic Tests for Combinational Logic Circuits," Automation and Remote Control, Vol. 28, No. 9, pp. 1346-1353, 1967.

151. Opferman, D. C., and Tsao-Wu, N. T., "On a Class of Rearrangeable Switching Networks, Part II: Enumeration Studies and Fault-Diagnosis," Bell System Technical Journal, Vol. 50, No. 5, pp. 1601-1618, May-June 1971.

152. Patch, F. D., and Zobniw, L. M., "Real-Time Diagnosis of Logic Assemblies," Proceedings of the 7th Design Automation Workshop, pp. 108-116, 1970.

153. Peeler, D.L., Meredith, P.H., Richards, L.M., and Clark, C.O., "Automatic Checkout Systems for Titan III and Apollo Guidance Computer Programs," IEEE Trans. Electron. Comput., Vol. EC-16, No. 5, pp. 580-590, October 1967.

154. Perlman, M., Theory and Calculus of Cubical Complexes, Technical Memo 324-33, Jet Propulsion Laboratory, Pasadena, Calif., March 1969 (JPL internal document).

155. Peterson, W. W., and Rabin, M. O., "On Codes for Checking Logical Operations," IBM Journal, Vol. 3, No. 2, pp. 163-168, April 1959.

156. Poage, J.F., "Derivation of Optimum Tests to Detect Faults in Combinational Circuits," PB 157804-18, Princeton Univ., N.J., March 1962.

157. Poage, J.F., and McCluskey Jr., E.J., "Derivation of Optimum Test Sequences for Sequential Machines," Proceedings of the 5th Annual Symposium on Switching Circuit Theory and Logical Design, pp. 121-132, Princeton University, N. J., November 1964.

158. Powell, T.J., "A Procedure for Selecting Diagnostic Tests," IEEE Trans. Comput. Vol. C-18, No. 2, pp. 168-175, February 1969.

159. Preparata, F.P., "Some Results on Sequentially Diagnosable Systems," Proceedings of Hawaii International Conference on System Sciences, pp. 623-626, University of Hawaii, January 1968.

160. Preparata, F.P., Metze, G., and Chien, R.T., "On the Connection Assignment Problem of Diagnosable Systems," IEEE Trans. Electron. Comput., Vol. EC-16, No. 6, pp. 848-854, December 1967.

161. Putzolu, G.R., et al., "Algorithms for Detection of Faults in Logic Circuits," Digest of the International Symposium on Fault-Tolerant Computing, pp. 5-8, March 1971.

162. Quiet, E., "An Automated Method for Producing Diagnostic Programs," Proceedings of the 7th Design Automation Workshop, 1970, pp. 132-138. IBM Custom Systems, Poughkeepsie, N.Y.

163. Ramamoorthy, C.V., "A Structural Theory of Machine Diagnosis," AFIPS Conference Proceedings of the Spring Joint Computer Conference, Atlantic City, N.J., Vol. 30, pp. 743-756, 1967.

164. Ramamoorthy, C. V., "Fault-Tolerant Computing: An Introduction and an Overview," IEEE Trans. Comput., Vol. C-20, No. 11, pp. 1241-1244, November 1971.

165. Ramamoorthy, C. V., and Mayeda, W., "Computer Diagnosis Using the Blocking Gate Approach," IEEE Trans. Comput., Vol. C-20, No. 11, pp. 1294-1299, November 1971.

166. Rault, J. C. L. G., "A Graph Theoretical and Probabilistic Approach to the Fault-Detection of Digital Circuits," Digest of the International Symposium on Fault-Tolerant Computing, pp. 26-29, March 1971.

167. Ravi, C. V., "Fault Location in Memory Systems by Program," AFIPS Proceedings of the Spring Joint Computer Conference, held in Boston, Mass., May 14-16, 1969. Vol. 34, pp. 393-401. AFIPS Press, Montvale, N. J.

168. Romero, R., and Taylor, G. W., "810A PBX System Control, Fault Detection and Recovery," Proceedings of IEEE International Conference on Communications, 91C28-COM, pp. 16-22 to 16-16, June 14-16, 1971.

169. Roth, P. J., "Diagnosis of Automata Failures: A Calculus and a Method," IBM J. Res. Devel., Vol. 10, No. 4, pp. 278-291, July 1966.

170. Roth, P. J., "An Algorithm to Compute a Test to Distinguish Between Two Failures in Logic Circuits," Proceedings of the 1970 IEEE International Computer Group Conference, Washington, D. C., pp. 247-249, June 1970.

171. Roth, J. P., "Systematic Design of Automata," AFIPS Proceedings of the Fall Joint Computer Conference, pp. 1093-1100, 1965.

172. Roth, J. P., "The Validity of Kron's Method of Tearing," Proceedings of the National Academy of Sciences, Vol. 41, No. 8, pp. 559-600, August 1955.

173. Roth, J. P., "Minimization Over Boolean Trees," IBM Journal of R&D, Vol. 4, No. 5, pp. 543-558, November 1960.

174. Roth, J. P., "Minimization Over Boolean Graphs," IBM Journal of R&D, Vol. 6, No. 2, pp. 227-238, April 1962.

175. Roth, J. P., "A Calculus and an Algorithm for the Multiple-Output 2-Level Minimization Problem," IBM Research Report, RC-2007, Feb. 6, 1968.

176. Roth, J. P., "On a Method of Design to Facilitate Testing," IBM Research Report, RC-2853, April 17, 1970.

177. Roth, J. P., A Pragmatic Theory of Algorithms, IBM Technical Publication, TR 00.918; also presented at International Symposium on Relay Systems Theory and Finite Automata, IFAC, Moscow, September-October 1962.

178. Roth, P.J., Bouricius, W.G., and Schneider, P.R., "Programmed Algorithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits, "IEEE Trans. Electron. Comput., Vol. EC-16, No. 5, pp. 567-580, October 1967.

179. Roth, J.P., and Wagner, E.G., "Algebraic Topological Methods for the Synthesis of Switching Systems, Part III: Minimization of Non-singular Boolean Trees," IBM Journal of Research and Development, Vol. 4, No. 4, pp. 1-19, October 1959.

180. Roth, P.J., Wagner, E.G., and Putzolu, G.R., A Formal Theory of Cubical Complexes, IBM Report, 1964.

181. Russell, J.D., and Kime, C.R., "Structural Factors in the Fault Diagnosis of Combinatorial Networks," Digest of the International Symposium on Fault-Tolerant Computing, pp. 34-39, March 1971 (See also IEEE Trans. Comput., Vol. C-20, No. 11, pp. 1276-1285, November 1971.)

182. Schertz, D.R., and Metze, G., "On the Design of Multiple Fault Diagnosable Networks, "Digest of the International Symposium on Fault-Tolerant Computing, pp. 46-50, March 1971.

183. Schertz, D.R., and Metze, G.A., "On the Indistinguishability of Faults in Digital Systems," IEEE Proceedings of the 6th Annual Allerton Conference on Circuit and System Theory, pp. 752-760, held in Monticello, Ill., October 2-4, 1968.

184. Shertz, D.R., and Metze, G.A., "The Use of Connection Graphs for the Detection of Digital Faults," IEEE Proceedings of the 7th Annual Allerton Conference, pp. 261-271, held in Monticello, Ill., October 8-10, 1969.

185. Shertz, D.R., and Metze, G., "On the Design of Multiple Fault Diagnosable Networks," IEEE Trans. Comput. Vol. C-20, No. 11, pp. 1361-1364, November, 1971.

186. Schneider, P.R., "On the Necessity to Examine D-chains in Diagnostic Test Generation – an Example," IBM J. Res. Devel., Vol. 11, No. 1, 1967.

187. Sellers, F.F., Hsiao, M.Y., and Bearnson, L.W., "Error Detecting Logic." McGraw-Hill Book Co., Inc., New York, 1968.

188. Sellers, F.F., Hsiao, M.Y., and Bearnson, L.W., "Analyzing Errors With the Boolean Difference," IEEE Trans. Comput., Vol. C-17, No. 7, pp. 676-683, July 1968 (see also Digest of the First Annual IEEE Computer Conference, pp. 6-9, September 1967).

189. Seshagiri, N., "A Decision Table Approach to Self-Diagnostic Computers," Proceedings of the IEEE, Vol. 55, No. 12, pp. 2180-2181, 1967.

190. Seshu, S., "On an Improved Diagnosis Program," IEEE Trans. Electron. Comput. Vol. EC-14, No. 1, pp. 76-79, February 1965.

191. Seshu, S., and Freeman, D. A., "The Diagnosis of Asynchronous Sequential Switching Systems," IRE Trans. Electron. Comput., Vol. EC-11, No. 4, pp. 459-465, August 1962.

192. Seshu, S., and Waxman, R., "Fault Isolation in Conventional Linear Systems, —a Feasibility Study," IEEE Trans. Reliabil., Vol. R-15, No. 1, pp. 11-16, May 1966.

193. Seth, S. C., "Fault-Diagnosis of Combinational Cellular Arrays," Proceedings of the 7th Annual Allerton Conference, pp. 272-283, October 1969.

194. Short, R. A., and Goldberg, J., "Soviet Progress in the Design of Fault-Tolerant Digital Machines," IEEE Trans. Comput., Vol. C-20, No. 11, pp. 1337-1352, November 1971.

195. Smith, G. R., and Yau, S. S., "A Programmed Fault-Detection Algorithm for Combinational Switching Networks," Proceedings of the National Electronics Conference, Vol. XXV, Chicago, pp. 668-673, December 1969.

196. Su, S. Y. H., and Cho, Y.-C., "A New Approach to the Fault Location of Combinational Circuits," IEEE Trans. Comput., Vol. C-21, No. 1, pp. 21-30, January 1972.

197. Suda, R., "An Application-Oriented Multiprocessing System, V: the Diagnostic Monitor," IBM Systems J., Vol. 6, No. 2, pp. 116-123, 1967.

198. Szygenda, S. A., "A Self-Repairing Memory System for Digital Computers," Proceedings of the 2nd Hawaii International Conference on System Sciences, pp. 877-880, University of Hawaii, 1969.

199. Szygenda, S. A., "Problems Associated With the Implementation and Utilization of Digital Simulators and Diagnostic Test Generation Systems," Digest of the International Symposium on Fault-Tolerant Computing, pp. 51-54, March 1971.

200. Szygenda, S. A. and Flynn, M. J., "Coding Techniques for Failure Recovery in a Distributive Modular Memory Organization," AFIPS Proceedings of Spring Joint Computer Conference, Vol. 38, pp. 459-466, 1971.

201. Szygenda, S. A., and Goldbogen, G. C., "Implementation and Extension of Multidimensional Path Sensitivity in a Simulation and Diagnosis System," Proceedings of the 7th Annual Allerton Conference, pp. 284-292, October 1969.

202. Szygenda, S. A., et al., "Implementation of Synthesized Techniques for a Comprehensive Digital Design, Verification and Diagnosis System," Proceedings of the Fifth Hawaii International Conference on System Sciences, University of Hawaii, pp. 293-295, 1972.

203. Takaoka, T., and Mine, H., "N-Fail-Safe Logical Systems," IEEE Trans. Comput., Vol. C-20, No. 5, pp. 536-542, May 1971.

204. Thurber, K. J., "Fault Location in Cellular Arrays," AFIPS Proceedings of the Fall Joint Computer Conference, Vol. 35, pp. 81-88, 1969.

205. Tokura, N., Kasami, T., and Hashimoto, A., "Fail-Safe Logic Nets," IEEE Trans. Comput., Vol. C-20, No. 3, pp. 323-330, March 1971.

206. Tsiang, S. H., and Ulrich, W., "Automatic Trouble Diagnosis of Complex Logic Circuits," Bell System Technical Journal, Vol. 41, pp. 1177-1200, 1962.

207. Ulrich, E. G., "Exclusive Simulation of Activity in Digital Networks," Commun. ACM, Vol. 12, No. 2, pp. 102-110, February 1969.

208. Urbano, R. H., "Modifiable Nets With Malfunctioning Elements," Proceedings of the Fifth Hawaii International Conference on System Sciences, pp. 283-285, University of Hawaii, 1972.

209. Vedeshenkov, V. A., "Construction of Test Tables for Detecting Logical Errors of Electronic Combinational Devices," Automation and Remote Control, Vol. 29, No. 3, pp. 491-500, 1968.

210. Volkov, A.F., Vedeshenkov, V.A., and Zenkin, V.D., "Automatic Failures Diagnosis in the Control Computers," Proceedings of the 3rd Congress of International Federation of Automatic Control, Paper 42E, London, June 1966.

211. Walters, L. R., "Diagnostic Programming Techniques for the IBM Type 701 E. D. P. M.," IRE National Convention Record, Pt. 7, pp. 55-58, 1963.

212. Wang, K. C., "Design of Asynchronous Sequential Machines With Fault-Detection Capabilities," Proceedings of the 7th Annual Allerton Conference, pp. 237-242, October 1969.

213. Weitzenfeld, A. A., and Happ, W. W., "Combinational Techniques for Fault Identification in Multiterminal Networks," IEEE Trans. Reliabil., Vol. R-16, No. 3, pp. 93-99, 1967.

214. Whitney, G. E., "Algebraic Fault Analysis for Constrained Combinational Networks," IEEE Trans. Comput., Vol. C-20, No. 2, pp. 141-148, February 1971.

215. Winograd, S., "Redundancy and Complexity of Logical Elements," Inform. Cont., Vol. 5, pp. 177-194, 1963.

216. Yau, S. S., and Orsic, M., "Fault Diagnosis and Repair of Cutpoint Cellular Arrays," IEEE Trans. Comput., C-19, pp. 259-262, March 1970.

217. Yau, S. S., and Tang, C. K., "Universal Logic Circuits and Their Modular Realizations," AFIPS Proceedings of the Spring Joint Computer Conference, Vol. 32, pp. 297-305, 1968.

218. Yau, S.S., and Tang, Y.-S., "An Efficient Algorithm for Generating Complete Test Sets for Combinational Logic Circuits," Digest of the International Symposium on Fault-Tolerant Computing, pp. 14-17, March 1971 (see also IEEE Trans. Comput., Vol. C-20, No. 11, pp. 1245-1251, November 1971).

219. Yen, Y. T., "A Method of Automatic Fault-Detection Test Generation for Four-Phase MOS LSI Circuits," AFIPS Proceedings of the Spring Joint Computer Conference, Vol. 34, pp. 215-220, 1969.

220. Yen, Y. T., "Computer-Aided Test Generation for Four-Phase MOS LSI Circuits," IEEE Trans. Comput., Vol. C-18, No. 10, pp. 890-894, October 1969.

221. Yermilov, V.A., and Kudyukov, N.I., "Fault Diagnosis in Digital Systems," Engin. Cyber., Vol. 8, No. 3, pp. 553-557, May-June 1970.

Table 1. Minimal diagnosing sequences for state pairs in M

s_{i0}	s_{j0}	$E(s_{i0}, s_{j0})$	Z_i	Z_j
1	2	$\beta\alpha\alpha\alpha$	1	0
1	3	$\alpha\alpha$	0	1
1	4	α	0	1
1	5	α	0	1
2	3	$\alpha\alpha$	0	1
2	4	α	0	1
2	5	α	0	1
3	4	α	0	1
3	5	α	0	1
4	5	$\alpha\alpha\alpha$	1	0

Table 2. A summary of some basic properties of Boolean difference

$$df(X)/dx_i \triangleq f(x_1, \cdots, x_i, \cdots, x_n) \oplus f(x_1, \cdots, \overline{x}_i, \cdots, x_n)$$

$$df(X)/dx_i = f(x_1, \cdots, 1, \cdots, x_n) \oplus f(x_1, \cdots, 0, \cdots, x_n)$$

$$df(X)/dx_i = 0 \text{ if } f(X) \text{ is independent of } x_i$$

$$df(X)/dx_i = 1 \text{ if } f(X) \text{ depends only on } x_i$$

$$df(\overline{X})/dx_i = df(X)/dx_i$$

$$df(X)/dx_i = df(X)/d\overline{x}_i$$

$$\frac{d}{dx_i}(df(X)/dx_j) = \frac{d}{dx_j}(df(X)/dx_i)$$

$$d(f(X) \cdot G(X))/dx_i = f(X)\, dG(X)/dx_i \oplus G(X)\, df(X)/dx_i \oplus df(X)/dx_i \cdot dG(X)/dx_i$$

$$d(f(X) + G(X))/dx_i = \overline{f}(X)\, dG(X)/dx_i \oplus G(X)\, df(X)/dx_i \oplus df(X)/dx_i \cdot dG(X)/dx_i$$

$$d(f(X) \oplus G(X))/dx_i = df(X)/dx_i \oplus dG(X)/dx_i$$

$$d(f(X) \cdot G(X))/dx_i = f(X)\, dG(X)/dx_i \text{ if } f(X) \text{ is independent of } x_i$$

$$d(f(X) + G(X))/dx_i = \overline{f(X)}\, dG(X)/dx_i \text{ if } f(X) \text{ is independent of } x_i$$

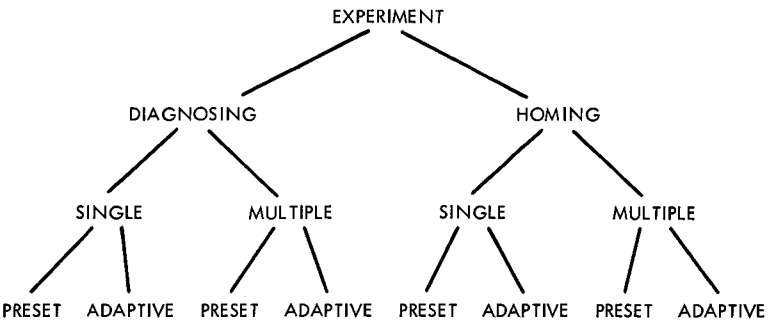


Fig. 1. Classification of identification experiments

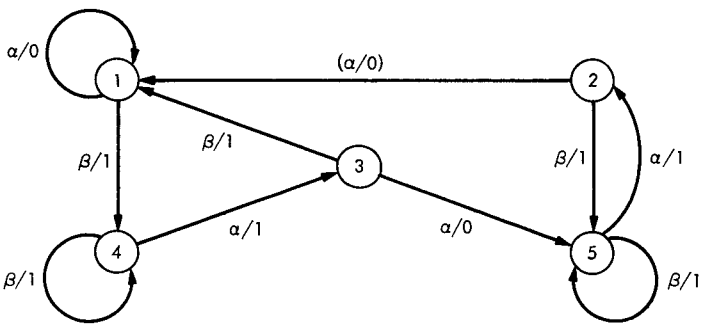


Fig. 2. Machine M transition diagram

(a) P_0

S_v \ X_v	Z_v		S_{v+1}	
	α	β	α	β
1	0	1	1	4
2	0	1	1	5
3	0	1	5	1
4	1	1	3	4
5	1	1	2	5

P_0 TABLE DETERMINED FROM FIGURE 2

(b) P_1

Σ_1	S_v \ S_{v+1}	α	β
a	1	1a	4b
	2	1a	5b
	3	5b	1a
b	4	3a	4b
	5	2a	5b

P_1 TABLE (DETERMINED FROM THE P_0 TABLE). THE RESPONSE SUBTABLE IS NOT SHOWN AGAIN. Σ_1, THE 1-EQUIVALENCE CLASSES ARE FORMED BY ADJOINING THOSE STATES IN THE TRANSITION TABLE HAVING IDENTICAL ROWS IN THE RESPONSE SUBTABLE i.e., STATES ARE ADJOINT IN P_1 IF AND ONLY IF, FOR EVERY INPUT SYMBOL, THEY YIELD IDENTICAL OUTPUT SYMBOLS

(c) P_2

Σ_2	S_v \ X_v	S_{v+1} α	S_{v+1} β
a	1	1a	4c
	2	1a	5c
b	3	5c	1c
c	4	3b	4c
	5	2a	5c

P_2 TABLE (DETERMINED FROM THE P_1 TABLE). Σ_2, THE TWO-EQUIVALENCE CLASSES ARE FORMED BY ADJOINING THOSE ADJOINT STATES IN THE P_1 TABLE THAT HAVE IDENTICAL SUBSCRIPT ROWS. DISJOINT ROWS IN P_1 REMAIN DISJOINT IN P_2

(d) P_3

Σ_3	S_v \ X_v	S_{v+1} α	S_{v+1} β
a	1	1a	4c
	2	1a	4d
b	3	5d	1a
c	4	3b	4c
d	5	2a	5d

(e) P_4

Σ_4	S_v \ X_v	S_{v+1} α	S_{v+1} β
a	1	1a	4d
b	2	1a	5e
c	3	5e	1a
d	4	3c	4d
e	5	2b	5e

P_4 TABLE CANNOT BE REFINED FURTHER, SINCE THE EQUIVALENCE CLASSES HAVE NO ADJOINT ROWS WITH DISTINGUISHABLE SUBSCRIPT ROWS. THE MOST REFINED PARTITION $\hat{P} = P_4$.

Fig. 3. P_k tables

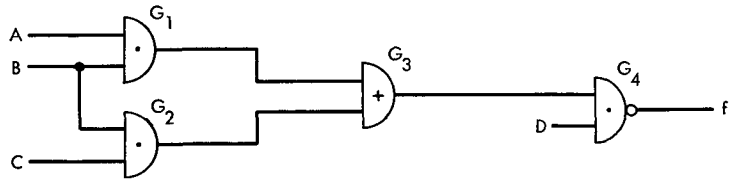


Fig. 4. Example of ENF of a combinational circuit with single output

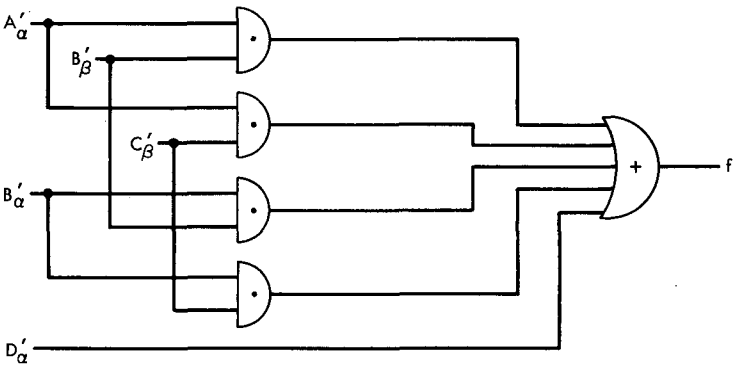


Fig. 5. Two-level AND-OR circuit

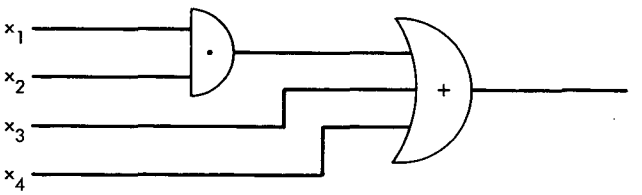


Fig. 6. An example of Boolean difference

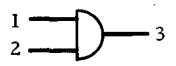
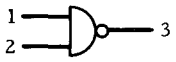
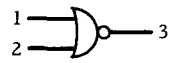
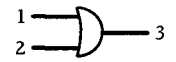
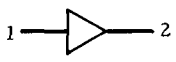
Block	Truth table	Singular cover	Primitive D-cubes
 AND	1 2 3 1 1 1 1 0 0 0 1 0 0 0 0	1 2 3 1 1 1 0 X 0 X 0 0	1 2 3 D 1 D 1 D D
 NAND	1 2 3 1 1 0 1 0 1 0 1 1 0 0 1	1 2 3 1 1 0 0 X 1 X 0 1	1 2 3 $\overline{D}$ 1 D 1 $\overline{D}$ D
 NOR	1 2 3 1 1 0 1 0 0 0 1 0 0 0 1	1 2 3 0 0 1 1 X 0 X 1 0	1 2 3 $\overline{D}$ 0 D 0 $\overline{D}$ D
 OR	1 2 3 1 1 1 1 0 1 0 1 1 0 0 0	1 2 3 0 0 0 1 X 1 X 1 1	1 2 3 D 0 D 0 D D
 NOT	1 2 0 1 1 0	1 2 X $\overline{X}$	

Fig. 7. Singular covers and primitive D-cubes of some common logic blocks

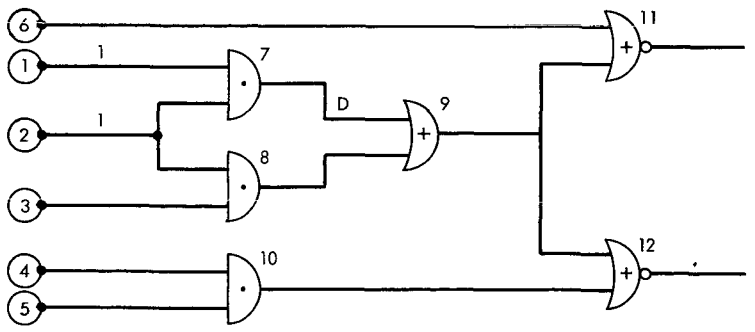


Fig. 8. Example of D-algorithm to construct test for "line 7 s-a-0"

NASA — JPL — Coml., L.A., Calif.