29 August 1969

Quarterly Progress Report 2
Covering the Period 10 May to 9 August 1969
The University of Michigan Project 02602
Jet Propulsion Laboratory Contract 952492



# THEORY AND DESIGN OF RELIABLE SPACECRAFT DATA SYSTEMS

Project Director

J. F. Meyer

This work was performed for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration under Contract NAS7-100.

Prepared for

Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, California

Copy No. 3

TABLE OF CONTENTS


| | | Page |
|---|---|---|
| I. | Objectives | 1 |
| II. | Personnel | 3 |
| III. | Summary of Technical Status | 4 |
| IV. | Technical Progress Report | 9 |
| | Permanent Memory Faults | 10 |
| | Fault Masking in Combinational Networks | 37 |
| | Diagnosis of Sequential Machines | 57 |
| | References | 71 |

# I. OBJECTIVES

The long range objective of this project, as described in the Statement of Work (Article I, JPL Contract No. 952492) is to conduct a study of theory and techniques applicable to the design, analysis and fault diagnosis of reliable spacecraft data systems. In accomplishing this effort, the investigation will be concerned with the following problems:

(A) Design and analysis of redundant combinational and sequential networks. This shall include the development of mathematical models for the study of temporary and permanent faults in switching networks, the results having application to the design of ultrareliable subsystems of the type prevalent in existing science data systems such as counters, sequence generators for timing and encoding, analog-to-digital converters and scratchpad memories. Explore in detail errors which result from permanent malfunctions of memory in sequential switching systems.

(B) Fault diagnosis of redundant systems at both the component and subsystem level. This shall include investigating the problem of specifying test and checkout procedures for systems in which the reliability has been enhanced using redundancy techniques which mask internal faults. Specific areas to be investigated shall include:

(i) Development of efficient diagnostic algorithms for sequential switching networks which contain redundancy.

(ii) Development of theory and techniques for determining test-point allocation in order to reduce the time (relative to input/output testing) needed to isolate and locate faults.

(iii) Investigate questions relating to how a data system should be organized to best facilitate both pre-flight and in-flight fault diagnosis.

## II. PERSONNEL

The principal investigator on the project is Professor John F. Meyer, Department of Electrical Engineering and Department of Computer and Communication Sciences, the University of Michigan. Three Research Assistants; Mr. F. Gail Gray, Mr. John R. Kinkel, and Mr. Koumin (Ken) Yeh have been working full-time on the project during the past quarter.

## III. SUMMARY OF TECHNICAL STATUS

During the past quarter, investigations have been initiated with regard to the following three problems:

1) Permanent memory faults

2) Fault masking in combinational networks

and 3) Fault diagnosis of sequential machines

The technical status of each of these investigations is summarized briefly in the paragraphs that follow. Also included is a discussion of planned efforts for the next period. A detailed technical report on each of these studies is contained in the body of this report (Section IV).

### Permanent Memory Faults

The purpose of this investigation is to study permanent memory faults in sequential switching systems and, in particular, the relationship between such faults and the resulting system behavior. One of the primary applications of this knowledge is the design of fault-tolerant switching systems. In addition to obtaining synthesis algorithms, a fundamental question which underlies the study is whether certain types of finite-state behavior are inherently less susceptible to memory faults than others.

The study is based on a machine-theoretic model wherein the result of a permanent fault in memory is formulated in terms of a sequential machine M that represents the fault-free sequential

switching system and a function $\mu$ on the states of M that represents the fault. The result of the fault is then represented by a second machine $M^{\mu}$ appropriately determined by M and $\mu$. Summarizing the research effort during the past quarter, it has been shown, first of all, that a succession of such faults can be regarded, alternatively, as some single fault which is simply the composition of the faults in the succession. With regard to the fault-masking problem, several formal notions of masking have been introduced, compared, and investigated with respect to properties that imply or are implied by a certain type of masking. Finally, a special class of faults called "stable" faults has been studied with regard to its basic properties and to how these properties relate to the masking problem.

During the next quarter, we will begin study of the synthesis problem by applying these results to the design of fault tolerant sequential switching networks. In particular, given the desired behavior and the class of faults that are to be masked, we wish to investigate state assignment procedures for realizations that mask the specified faults. In addition, relative to each of the various types of masking under consideration, we will continue to investigate conditions that are necessary and/or sufficient for a given type of masking.

Fault Masking in Combinational Networks

A mathematical model for studying fault-masking and fault-diagnosis in combinational networks has been developed that allows a

hierarchy of fault-diagnosis concepts to be defined. Necessary and sufficient conditions have been obtained for a single fault to be masked or detected in a two node and in a three node system. It has been shown that any system may be transformed into the three node system for the purpose of analyzing system response to single faults at a node or for analyzing the response to multiple faults in a connected subsystem. Necessary and sufficient conditions have been found for masking all faults and for detecting all faults in a connected subsystem of a large system.

Theorems have been discovered that allow efficient enumeration of the number of single faults masked (and/or the number of single faults detected) at a node in a general system. These theorems also apply to multiple faults in a connected subsystem. Based on these theorems, an algorithm is being developed for analyzing single-fault masking in a general network. The algorithm can also be used for analyzing a network for multiple fault masking in connected subsystems. This effort will be reported in the next Quarterly Progress Report.

Effort during the next quarter will also include employment of the combinational network model in conjunction with input space partitions to prove some general results about fault-masking when all faults are assumed equally likely.

Work will continue on the development of an algorithm to analyze general networks for fault-masking. Known functional

decomposition schemes will be investigated for possible application to reliable design. The concepts of test point and test input will be incorporated into the model in order to begin a study of test point allocation for the diagnosis of redundant combinational networks. Further attention will be given to the problem of masking specific types of faults.

Fault Diagnosis of Sequential Machines

A study of the problem of designing sequential machines with fault detecting capabilities has been initiated, beginning with a classification of machines according to some machine-theoretic properties pertinent to the design of fault detecting experiments. This permits identification of classes of machines having short distinguishing sequences. Necessary and sufficient conditions for the existence of a repeated symbol distinguishing sequence and a bound on its length have also been obtained. Based on these conditions, methods for constructing sequential machines with such distinguishing sequences are being developed. Machines so constructed yield shorter fault detecting experiments than the original bound given by Kohavi and Lavallee.

During the next quarter we intend to explore the possibility of applying certain fault detecting experiments to the problem of fault location in sequential networks. It is easily seen that if failures preserve the diagnosable and strongly connected properties of the original machine then each of the faults can be uniquely identified.

This may have particular application to types of faults such as those caused by the malfunction of memory elements which can not be easily located by the conventional technique of cutting feedback lines. Questions such as the relationship between state assignment and the preservation of strongly-connectedness under memory faults and the relationship between definite diagnosability and the preservation of diagnosability under memory faults will also be investigated.

## IV. TECHNICAL PROGRESS REPORT

The following is a technical progress report on the research activity of the past quarter. Investigations during this period were concerned with the three problem areas summarized in Section III: 1) permanent memory faults, 2) fault masking in combinational networks and 3) fault diagnosis in sequential machines.

The report is quite comprehensive but omits, for the most part, detailed examples and proofs of theorems. This is done in the interest of providing a more cohesive discussion of concepts and results and more commentary regarding motivation and interpretation. Proofs and examples which are omitted will be included in the first annual report.

# 1. PERMANENT MEMORY FAULTS

The purpose of this investigation is to study permanent memory faults in sequential switching systems and in particular the relationship between such faults and the resulting system behavior. It has been shown [ 5 ] that the result of a permanent fault in memory can be formulated in terms of a sequential machine M that represents the fault-free sequential switching system and a function $\mu$ on the states of M that represents the fault. The result of the fault is then represented by a second machine $M^{\mu}$ appropriately determined from M and $\mu$. Given this represention, it is possible to investigate conditions under which the behavior of $M^{\mu}$ relates in some specified way to the behavior of M.

The fundamental question which underlies the search for such conditions is whether certain types of finite-state behavior are inherently less susceptible to memory faults than others. One measure of susceptibility is the minimum amount of redundant memory required to reliably realize the behavior when the realization is subject to some specified class of faults.

We begin with a review of several basic concepts of sequential machine theory in order to precisely establish the terminology and notation used throughout the discussion.

Definition 1.0

A Mealy sequential machine is a system $M = (I, Q, O, \delta, \omega)$ where

i) I is a finite set of input symbols,

ii) Q is a finite set of states,

iii) O is a finite set of output symbols,

iv) $\delta$ is a function from $Q \times I$ into Q, the transition function of M,

v) $\omega$ is a function from $Q \times I$ into O, the output function of M.

A Moore sequential machine is as above except that

v') $\omega$ is a function from Q into O.

To describe the behavior of a sequential machine M, let A be any finite set, $A^*$ the set of all sequences (words, strings) over A ($A^*$ includes the null sequence $\Lambda$), and if $x \in A^*$ let

$$lg(x)$$

denote the length of x (the number of symbols in the sequence x). Then for each nonnegative integer k, we define the set

$$A^k = \{x \mid x \in A^* \text{ and } lg(x) = k\}$$

which is simply the set of all sequences over A of length k. Note that, in terms of the sets $A^k$,

$$A^* = \bigcup_{k=0}^{\infty} A^k.$$

If we let $A^\dagger$ denote all sequences except the null sequence then

$$A^\dagger = \bigcup_{k=1}^{\infty} A^k$$

Using this notation we extend the transition and output functions of a sequential machine as follows:

Definition 1.1

If $M = (I, Q, O, \delta, \omega)$ is a sequential machine its extended transition function $\bar{\delta}$ and extended output function $\bar{\omega}$ are defined inductively as follows:

| | Transition fcn. | Output fcn. (Mealy) | Output fcn. (Moore) |
|---|---|---|---|
| | $\bar{\delta}: Q \times I^* \to Q$ | $\bar{\omega}: Q \times I^+ \to O$ | $\bar{\omega}: Q \times I^* \to O$ |
| where | | | |
| i) | $x \in I^0 = \{\Lambda\}$, $\bar{\delta}(q,x) = q$ | undefined | $\bar{\omega}(q,x) = \omega(q)$ |
| ii) | $x \in I^1 = I$, $\bar{\delta}(q,x) = \delta(q,x)$ | $\bar{\omega}(q,x) = \omega(q,x)$ | $\bar{\omega}(q,x) = \omega(\delta(q,x))$ |
| iii) | $x \in I^k, a \in I$, $(k \geq 1)$ $\bar{\delta}(q,xa) = \delta(\bar{\delta}(q,x), a)$ | $\bar{\omega}(q,xa) = \omega(\bar{\delta}(q,x), a)$ | $\bar{\omega}(q,xa) = \omega(\bar{\delta}(q,xa))$ |

(Note that given values of $\bar{\delta}$, $\bar{\omega}$ for all $x \in I^k$, (iii) defines values of $\bar{\delta}$, $\bar{\omega}$ for all $y \in I^{k+1}$.)

In terms of these extended functions, the behavior of M relative to some fixed state $q \in Q$ is defined as follows.

Definition 1.2

If $M = (I, Q, O, \delta, \omega)$ is a sequential machine and $q \in Q$, the behavior of M for initial state q is a function $\beta_q$ defined as follows:

| | Mealy | Moore |
|---|---|---|
| | $\beta_q : I^{\dagger} \to O^{\dagger}$ | $\beta_q : I^{*} \to O^{\dagger}$ |

where

| | | | |
|---|---|---|---|
| i) | $x \in I^0 = \{\Lambda\}$, | undefined | $\beta_q(x) = \omega(q)$ |
| ii) | $x \in I^1 = I$, | $\beta_q(x) = \omega(q, x)$ | $\beta_q(x) = \omega(q)\omega(\delta(q, x))$ |
| iii) | $x \in I^k$, $a \in I$, $(k \geq 1)$ | $\beta_q(xa) = \beta_q(x)\bar{\omega}(q, xa)$ | $\beta_q(xa) = \beta_q(x)\bar{\omega}(q, xa)$ |

Note that if M is a Mealy machine then

$$\lg(\beta_q(x)) = \lg(x),$$

i.e., an input sequence of length k produces an output sequence of length k. On the other hand, if M is a Moore machine then

$$\lg(\beta_q(x)) = \lg(x) + 1$$

since an output symbol is associated with the initial state q.

Definition 1.3

The behavior of a sequential machine M with states Q is the set

$$B_M = \{\beta_q \mid q \in Q\}.$$

In other words, the behavior of M is the collection of input-output transformations such that each transformation in the collection can be realized with an appropriate choice of initial state. Note that distinct states of M need not give rise to distinct behaviors, i.e., it may be the case that $q \neq r$ and yet $\beta_q = \beta_r$. This observation leads

to the following fundamental concept of machine theory.

Definition 1.4

If $M = (I, Q_M, O, \delta_M, \omega_M)$ and $N = (I, Q_N, O, \delta_N, \omega_N)$ are sequential machines (of the same type), $q \in Q_M$, and $r \in Q_N$ then q is equivalent to r ($q \equiv r$) if $\beta_q = \beta_r$.

In words, state q of machine M is equivalent to state r of N if M when started in q has the same input-output behavior as N when started in r. It should be obvious that in the special case where $M = N$, $\equiv$ is an equivalence relation on $Q_M$. One also notes that state equivalence can be characterized in terms of the extended output functions as follows:

$$q \equiv r \text{ iff } \bar{\omega}_M(q,x) = \bar{\omega}_N(r,x) \qquad (1.1)$$

for all $x \in I^+$ (Mealy case) or for all $x \in I^*$ (Moore case). (This characterization is the one most often used as the definition of state equivalence).

Extending the notion of state equivalence to machines we have:

Definition 1.5

If M and N are sequential machines (having the same input alphabet I and output alphabet O) then M is equivalent to N ($M \equiv N$) if $B_M = B_N$.

In other words, equivalent machines are identical when viewed externally. If we let $\mathcal{M}(I,O)$ denote the set of all sequential

machines with input symbols I and output symbols O then $\equiv$ is clearly an equivalence relation on $\mathcal{M}(I,O)$. In comparing the behavior of machines, it is convenient to introduce a second notion that is somewhat weaker than machine equivalence, namely

Definition 1.6

If $M, N \in \mathcal{M}(I,O)$ then <u>M includes N</u> ($M \geq N$) if $B_M \supseteq B_N$.

Thus if M includes N, each state of N is equivalent to some state of M but there may be states of M not equivalent to any state of N. Paraphrasing the notion, M includes N if M can do anything that N does. From the definitions it is obvious that M and N are equivalent if and only M includes N and N includes M. Accordingly, the notion of "includes" determines a partial ordering of the set of all equivalence classes of machines in $\mathcal{M}(I,O)$.

In terms of these basic notions of machine structure and behavior, suppose now that in some physical system represented by a sequential machine M, there is a permanent fault which permanently alters the structure of the system but results in a configuration which is still machine-representable. In this case one can represent the result of the fault as a second machine:

$$M' = (I, Q', O, \delta', \omega')$$

where the states Q', transition function $\delta'$, and output function $\omega'$ of the faulty machine are related in some way to the original machine M. A more precise statement of this relationship depends, of course,

on more detailed knowledge of the fault.

In what follows we restrict our attention to faults that occur in the memory portion of the physical system. This restriction is motivated by the fact that it is memory which distinguishes nontrivial sequential switching systems from purely combinational systems. The restriction also has the advantage that the function of memory is the same from machine to machine, that is, to store the information presented at the memory input.

In a sequential machine the transition function represents both decision and memory processes in that we interpret $\delta(q, a)$ to be the "next" state given the "present" state is q and the "present" input is a. To distinguish the functions of memory and decision let $\delta = \mu \cdot \lambda$ (the functional composition of $\lambda$ and $\mu$, first applying $\lambda$) where $\lambda(q, a)$ is the memory input and represents a purely combinational process, and $\mu$ is the memory function representing the storage of $\lambda(q, a)$. In case the memory operates properly, $\mu$ is simply the identity function on the states Q. Hence,

$$\delta = \lambda. \tag{1.2}$$

Suppose, now, that there is some permanent fault in memory that causes certain of the memory inputs to be stored improperly. Then the function $\mu$ representing faulty memory operation is no longer the identity function and the transition function of the faulty machine is given by

$$\delta' = \mu \cdot \lambda' \tag{1.3}$$

Assuming that there are no faults in the combinational processing, we have

$$\lambda' = \lambda$$

and hence

$$\delta' = \mu \cdot \lambda' = \mu \cdot \lambda = \mu \cdot \delta . \tag{1.4}$$

The above observations can be formalized as follows:

Definition 1.7

If M is a sequential machine, a (memory) fault of M is a function on the states of M.

Definition 1.8

If $M = (I, Q, O, \delta, \omega)$ is a sequential machine and $\mu: Q \to Q$ is a fault of M, the result of $\mu$ is the sequential machine

$$M^{\mu} = (I, Q^{\mu}, O, \delta^{\mu}, \omega^{\mu})$$

where

i) $Q^{\mu} = \mu(Q)$ (the range of $\mu$),

ii) $\delta^{\mu} = \mu \cdot \delta$ restricted to $Q^{\mu} \times I$,

iii) $\omega^{\mu} = \omega$ restricted to $\begin{cases} Q^{\mu} \times I \text{ (Mealy)} \\ Q^{\mu} \text{ (Moore)} \end{cases}$

Note that, by definition, the identity function (on Q) is regarded as a fault even through, under the intended interpretation, it represents

fault-free operation. Thus the identity function is referred to as an improper fault, all other faults being proper. In defining the result $M^{\mu}$ of a fault $\mu$, $Q^{\mu}$ is taken to be the range of $\mu$ since, under the interpretation, these are the only states accessible from the memory input. The definition of the faulty transition function $\delta^{\mu}$ follows directly from (1.2) - (1.4). Since the fault occurs in memory, the output function $\omega^{\mu}$ is essentially the same as $\omega$. Definitions 1.7 and 1.8 thus comprise the basic model of permanent memory error upon which the following investigation is based.

Before discussing the effects of faults on behavior, we note that a fault $\mu$ can represent either some single physical fault in the corresponding switching system or the culmination of a series of many physical faults. For this reason we should make precise what is meant by one fault "following" another. We note first of all that if M is a sequential machine, $\mu$ is a fault of M and $\gamma$ is a fault of $M^{\mu}$ then

$$(M^{\mu})^{\gamma} = M^{\gamma \cdot \mu} \tag{1.5}$$

This follows by Definitions 1.7 and 1.8 and says that the result of successive faults $\mu$ of M and $\gamma$ of $M^{\mu}$ can be regarded as the result of the single fault $\gamma \cdot \mu$, the composition of $\mu$ and $\gamma$ (with the codomain of $\gamma \cdot \mu$ extended to Q).

Given $\mu$ and $\gamma$ as above, one can also regard $\gamma$ as a fault of the original machine M provided the following condition is satisfied.

If $\mu$: $Q \to Q$ and $R \subseteq Q$, let $\mu|R$ denote the restriction of $\mu$ to R. Then

Definition 1.9

If $\mu, \gamma$ are faults of M then $\gamma$ can follow $\mu$ if $\gamma|Q^{\mu}$ is a fault of $M^{\mu}$.

Although the definition reflects the interpretation of the notion "can follow", a more convenient characterization is given by the following theorem. If we let $\mathcal{R}(\mu)$ denote the range of a function on $\mu$ (i.e. $\mathcal{R}(\mu) = \mu(Q)$ if $\mu$: $Q \to Q$) then

Theorem 1.1

If $\mu$, $\gamma$ are faults of M then $\gamma$ can follow $\mu$ if and only if $\mathcal{R}(\gamma \cdot \mu) \subseteq \mathcal{R}(\mu)$.

This observation follows immediately from Definitions 1.7 and 1.9. The above is easily generalized to allow for a succession of more than two faults.

Definition 1.10

If $\mu_1, \mu_2, \ldots, \mu_n$ are faults of M then $(\mu_1, \mu_2, \ldots, \mu_n)$ is a succession of faults of M if

$$\mu_{i+1} \text{ can follow } \mu_i \cdot \mu_{i-1} \cdots \mu_1$$

for $i = 1, 2, \ldots, n-1$.

Theorem 1.1 can then be generalized as follows.

Theorem 1.2

If $\mu_1, \mu_2, \ldots, \mu_n$ are faults of M then $(\mu_1, \mu_2, \ldots, \mu_n)$ is a succession of faults of M if and only if

$$\mathcal{R}(\mu_{i+1} \cdot \mu_i \cdots \mu_1) \subseteq \mathcal{R}(\mu_i \cdot \mu_{i-1} \cdots \mu_1)$$

for $i = 1, 2, \ldots, n-1$.

If we now extend Definition 1.8 to successive faults in the obvious way, it follows that

Theorem 1.3

The result of a succession of faults $(\mu_1, \mu_2, \ldots, \mu_n)$ of M is the machine

$$M^{\mu_n \cdot \mu_{n-1} \cdots \mu_1}.$$

In other words, the result of a succession of faults of M can be regarded as the result of a single fault $\mu$ of M where $\mu$ is just the composition of all the faults in the succession (taken in the order with which they occur). Thus multiple physical faults can be analyzed in terms of a single machine fault and, more generally, the various effects of any prescribed set of physical faults can be analyzed by studying the individual effect of each fault in some appropriately determined set of machine faults.

Fault Masking

Let us now consider the fundamental problem of relating faulty structure to desired behavior. Informally we can say that

a machine M has "failed" under some fault $\mu$ if $M^{\mu}$ no longer exhibits the desired behavior. On the other hand, if the desired behavior is preserved under $\mu$ then, adopting a term used quite frequently in this context, the fault $\mu$ is "masked." The precise sense in which a fault is masked depends, of course, on what is meant by "desired behavior." In what follows, we propose several types of masking which we feel have meaningful interpretation.

Perhaps the most natural choice of desired behavior for the faulty machine is a behavior equal to that of the fault-free machine. In this case we say that

Definition 1.11:

A fault $\mu$ of M is e-masked if $M^{\mu} \equiv M$.

If we require only that the faulty machine be able to do everything the original machine did then

Definition 1.12:

A fault $\mu$ of M is i-masked if $M^{\mu} \geq M$.

Clearly, if a fault is e-masked it is i-masked.

Example 1.1

Consider the modulo 3 counter (Mealy type) having the following transition-output table:*

---

*The entry in row q and column a is $\delta(q, a)/\omega(q, a)$.

M:

| Q \ I | 0 | 1 |
|---|---|---|
| 0 | 0/1 | 1/0 |
| 1 | 1/0 | 2/0 |
| 2 | 2/0 | 3/1 |
| 3 | 3/1 | 4/0 |
| 4 | 4/0 | 5/0 |
| 5 | 5/0 | 0/1 |

and faults $\mu_1$, $\mu_2$, and $\mu_3$ given by:

| q | $\mu_1(q)$ | $\mu_2(q)$ | $\mu_3(q)$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 4 | 1 | 2 |
| 2 | 0 | 2 | 2 |
| 3 | 5 | 0 | 5 |
| 4 | 4 | 4 | 0 |
| 5 | 5 | 4 | 5 |

Then the faulty machines $M^{\mu_1}$, $M^{\mu_2}$ and $M^{\mu_3}$ are given by:

$M^{\mu_1}$:

| Q \ I | 0 | 1 |
|---|---|---|
| 0 | 0/1 | 4/0 |
| 4 | 4/0 | 5/0 |
| 5 | 5/0 | 0/1 |

$M^{\mu_2}$:

| Q \ I | 0 | 1 |
|---|---|---|
| 0 | 0/1 | 1/0 |
| 1 | 1/0 | 2/0 |
| 2 | 2/0 | 0/1 |
| 4 | 4/0 | 4/0 |

$M^{\mu_3}$:

| Q \ I | 0 | 1 |
|---|---|---|
| 0 | 0/1 | 2/0 |
| 2 | 2/0 | 5/1 |
| 5 | 5/0 | 0/1 |

In the machine M, $\beta_0 = \beta_3$, $\beta_1 = \beta_4$, $\beta_2 = \beta_5$. If we let $\beta^{\mu}_{q}$ denote the behavior of $M^{\mu}$ for initial state q then, by inspection of $M^{\mu}$, we have:

$$\beta_0^{\mu_1} = \beta_0,\ \beta_4^{\mu_1} = \beta_1,\ \beta_5^{\mu_1} = \beta_2$$

and so $M^{\mu_1} \equiv M$, i. e. $\mu_1$ is e-masked.

Regarding $M^{\mu_2}$:

$$\beta_0^{\mu_2} = \beta_0,\ \beta_1^{\mu_2} = \beta_1,\ \beta_2^{\mu_2} = \beta_2$$

but $\beta_4^{\mu_2} \neq \beta_q$, for all $q \in Q$. Hence $\mu_2$ is i-masked but not e-masked.

As for $M^{\mu_3}$, we see that no state of $M^{\mu_3}$ is equivalent to any state of M and consequently $\mu_3$ is not i-masked.

By definition, a fault $\mu$ is e-masked if the faulty machine $M^{\mu}$ has the same terminal behavior as the fault-free machine M. In physical terms, this says that the faulty circuit or system represented by $M^{\mu}$ can do the same thing as the original system represented by M. This is not to say, however, that every state

$\mu(q) \in Q^{\mu}$ behaves in $M^{\mu}$ as state q does in M, i. e. it may be the case that $\beta^{\mu}_{\mu(q)} \neq \beta_q$. This is illustrated in Example 1.1 where $\beta^{\mu_1}_{\mu_1(2)} = \beta^{\mu_1}_{0} \neq \beta_2$. Accordingly if we were to attempt to reset the faulty system (represented by $M^{\mu_1}$) to state 2 it would actually reset to state $\mu_1(2) = 0$ and consequently exhibit a different behavior than expected after reset. Since the ability to reset to some known behavior is desirable in certain applications, we introduce the following notion.

Definition 1.13:

If M is a machine with states Q and $R \subseteq Q$ then a fault $\mu$ is R-masked if

$$\beta^{\mu}_{\mu(r)} = \beta_r, \quad \text{for all } r \in R$$

(where $\beta^{\mu}_{q}$, as earlier, is the behavior of $M^{\mu}$ for initial state q).

Thus if $\mu$ is R-masked then $M^{\mu}$ is resetable to every state $r \in R$ in the sense that the behavior of M for initial state r is the same as the behavior of $M^{\mu}$ for initial state $\mu(r)$. Referring to Example 1.1, one can easily verify that $\mu_1$ is $\{0, 1, 4, 5\}$-masked and $\mu_2$ is $\{0, 1, 2, 3\}$-masked. $\mu_3$ is not R-masked if $R \neq \phi$ ($\phi$ being the empty set; note that every fault is $\phi$-masked).

Relating R-masking to e-masking and i-masking we note the following facts.

Theorem 1.4

If M is a machine with states Q and a fault $\mu$ is Q-masked then $\mu$ is e-masked.

The proof of 1.4 is immediate from Definitions 1.11 and 1.13. That the converse fails to hold is illustrated by fault $\mu_1$ of example 1.1. Indeed one can construct a machine M along with a fault $\mu$ such that $\mu$ is e-masked and yet $R \neq \phi$ implies $\mu$ is not R-masked.

If M is a machine with states Q and behavior $B_M$ let us say that a subset R of Q is complete if

$$\{\beta_r | r \in R\} = B_M.$$

Then

Theorem 1.5

If M is a machine with states Q, R is complete ($R \subseteq Q$) and $\mu$ is R-masked then $\mu$ is i-masked.

Proof: If $\mu$ is R-masked then

$$\{\beta_r | r \in R\} = \{\beta^{\mu}_{\mu(r)} | r \in R\}.$$

But R is complete and so

$$B_M = \{\beta^{\mu}_{\mu(r)} | r \in R\} \subseteq \{\beta^{\mu}_{\mu(q)} | q \in Q\} = B_{M^{\mu}}.$$

In other words $M^{\mu} \geq M$ and hence $\mu$ is i-masked.

To illustrate Theorem 1.5, consider the fault $\mu_2$ of Example 1.1 along with the subset $R = \{0, 1, 2, 3\}$. Then R is complete and, as noted earlier, $\mu_2$ is R-masked. Hence $\mu_2$ must be i-masked and

we observed in Example 1.1 that this was the case. The converse of Theorem 1.5 does not hold, that is, a fault $\mu$ can be i-masked and yet there is no complete subset R such that $\mu$ is R-masked.

Let us now look ahead, for a moment, to the synthesis problem; that is, given some behavior B specified say by a reduced machine M' such that $B_{M'} = B$, design a machine M that realizes M' and relative to some specified set of faults $\{\mu_1, \mu_2, \ldots, \mu_k\}$, $\mu_i$ is $\square$-masked (i=1, 2, ..., k) where $\square$ denotes one of the specific types of masking just discussed. Solutions to this problem require a greater understanding of how a fault $\mu$ must relate to a machine M in order that it be $\square$-masked. In particular it would be convenient to relate $\mu$ directly to M without having to completely determine the nature of the faulty machine $M^{\mu}$. The following results are so motivated.

For a machine M with states Q let $\equiv$ denote the relation of state equivalence on Q.

Theorem 1.6

If $\mu$ is a fault of M and $\mu(q) \equiv q$, for all $q \in Q$, then $\mu$ is Q-masked (and hence e-masked).

Theorem 1.6 can be proved by showing that there exists a (machine) homomorphism $\eta$ from the faulty machine $M^{\mu}$ onto a reduced machine equivalent to M. This implies $M^{\mu} \equiv M$ (i.e., $\mu$ is e-masked). Moreover, $\eta$ can be chosen such that $\eta(\mu(q)) \equiv q$ which, by the behavior preserving property of homomorphisms, implies that $\mu$ is Q-masked. A detailed

proof of this theorem and theorems that follow will be supplied in a later report.

A somewhat more general form of Theorem 1.6 is the following.

Theorem 1.7

If $\mu$ is a fault of M and $R \subseteq Q$ such that

i) $\mu(r) \equiv r$, for all $r \in R$

and

ii) $\delta(\mu(R) \times I) \subseteq R$

then $\mu$ is R-masked.

Corollary

If $\mu$ and R satisfy the conditions of Theorem 1.7 and R is complete then $\mu$ is i-masked.

Note that when $R = Q$, condition ii) is automatically satisfied and Theorem 1.7 reduces to Theorem 1.6.

Theorems 1.6 and 1.7 give sufficient conditions for Q-masking and, more generally, R-masking a fault $\mu$ in terms of $\mu$, the state-equivalence relation for M, and the transition function of M. The conditions, however, are not necessary and to date we have been unable to discover necessary and sufficient conditions for R-masking that can be easily stated in terms of properties of M and $\mu$. The best characterization obtained so far is stated in terms of a relation $\mu_R$ defined as follows:

Definition 1.14

If M is a machine with states Q and $R \subseteq Q$ then, for all $q, q' \in Q$,

$$q \mu_R q' \text{ if } \exists r \in R \text{ and } x \in I^* \text{ such that}$$
$$\bar{\delta}(r, x) = q \text{ and } \bar{\delta}^{\mu}(\mu(r), x) = q'.$$

($\bar{\delta}^{\mu}$ is the extended transition function of $M^{\mu}$.)

If further we let $\equiv_1$ denote the relation of 1-equivalence on the states of M (i.e. $q \equiv_1 q'$ if $\beta_q(a) = \beta_{q'}(a)$ for all $a \in I$) then:

Theorem 1.8

If M is a Mealy machine and $\mu$ is a fault of M then

$$\mu \text{ is R-masked iff } \mu_R \subseteq \equiv_1.$$

An analogous statement for Moore machines involves 0-equivalence (i.e. $q \equiv_0 q'$ if $\omega(q) = \omega(q')$):

Theorem 1.8'

If M is a Moore machine and $\mu$ is a fault of M then

$$\mu \text{ is R-masked iff } \mu_R \subseteq \equiv_0.$$

In many applications, a sequential switching system will have a distinguished "reset state" where only the behavior of interest is the input-output function that results when the system is initially in the reset state. If such a system is represented by a machine M and the reset state by some distinguished "initial state" of M, say $q_o$, then the fault masking of interest is a special case of R-masking where

$$R = \{q_o\}.$$

In this case we will say that $\mu$ is $q_o$-masked (rather than $\{q_o\}$-masked) and write $\mu_{q_c}$ instead of $\mu_{\{q_o\}}$ (Definition 1.14). Moreover, the relation $\mu_R$ can be described somewhat more simply when $R = \{q_o\}$, that is

$$\mu_{q_o} = \{(\bar{\delta}(q_o, x),\ \bar{\delta}^{\mu}(\mu(q_o), x)) \mid x \in I^*\}. \tag{1.6}$$

Using this characterization of $\mu_{q_o}$ and applying Theorem 1.8 it follows that:

Theorem 1.9

If M is a Mealy machine and $\mu$ is a fault of M then

$\mu$ is $q_o$-masked iff $\bar{\delta}(q_o, x) \equiv_1 \bar{\delta}^{\mu}(\mu(q_o), x)$, for all $x \in I^*$.

Theorem 1.9'

In the statement of Theorem 1.9 replace "Mealy" by "Moore" and $\equiv_1$ by $\equiv_0$.

Applications of Theorems 1.6 - 1.9 to the design of fault-tolerant switching networks is presently under investigation and this activity will be reported on in the next Quarterly Progress Report.

Stable Faults

If M is a machine and $\mu$: $Q \to Q$ is a fault of M, we may interpret $\mu(q)$ as the state stored when the memory input is q and in case $\mu(q) \neq q$, q is stored erroneously. In general, if we now attempt to store $\mu(q)$, it too may be stored erroneously, i.e. it may be the case that

$\mu(\mu(q)) \neq \mu(q)$. Borrowing some terminology from the theory of asynchronous machines we say that $\mu(q)$, in this case, is unstable. On the other hand, if $\mu(\mu(q)) = \mu(q)$, then $\mu(q)$ is stable. Extending this notion to the fault itself we have:

Definition 1.15

If $\mu$: $Q \to Q$ is a fault of M then $\mu$ is stable if $\mu \cdot \mu = \mu$ (i.e. $\mu(\mu(q)) = \mu(q)$, for all $q \in Q$).

In other words, $\mu$ is stable if every state of $M^{\mu}$ is stable. In mathematical terms, $\mu$ is stable if and only if it is an idempotent element of the semigroup of functions on Q. Accordingly, the notion of a stable fault can alternatively be characterized as follows.

Theorem 1.10

If $\mu$: $Q \to Q$ is a fault of M then the following statements are equivalent:

i) $\mu$ is stable

ii) $\mu(r) = r$, for all $r \in \mathcal{R}(\mu)$

iii) $\mu(q) \in \mu^{-1}(\mu(q))$, for all $q \in Q$.

Stable faults are of interest since many types of physical memory faults may be represented by a machine fault of this type. In particular, a combination of "stuck at 0" and "stuck at 1" faults in one or more two-state memory cells is represented by a stable fault of the corresponding sequential machine.

If M is a sequential machine with states Q let

$$S(Q)$$

denote the set of all stable faults of M. As S(Q) is just the set of all idempotent elements of the full transformation semigroup, a partial ordering* of S(Q) can be defined as follows [ 1 ]:

Definition 1.16

If $\mu, \gamma \in S(Q)$ then $\mu$ is under $\gamma$ ($\mu \leq \gamma$) if $\gamma \cdot \mu = \mu \cdot \gamma = \mu$.

In general, if E is a set of idempotents, the partial ordering defined above is referred to as the natural partial ordering of E. In the case of stable faults, the ordering has a much more revealing characterization. If $\mu$: $Q \to Q$ let $\underset{\mu}{\equiv}$ denote the equivalence relation on Q induced by $\mu$, that is

$$q \underset{\mu}{\equiv} r \text{ if } \mu(q) = \mu(r). \tag{1.7}$$

Then

Theorem 1.11

If $\mu, \gamma \in S(Q)$ then $\mu \leq \gamma$ if and only if

i) $\mathcal{R}(\mu) \subseteq \mathcal{R}(\gamma)$

and

ii) $\underset{\gamma}{\equiv} \subset \underset{\mu}{\equiv}$.

---

*A relation R on a set A is a partial ordering of A if R is reflexive, antisymmetric, and transitive.

In other words $\mu$ is under $\gamma$ if and only if the range of $\mu$ is contained in the range of $\gamma$ and $q \underset{\gamma}{\equiv} r$ implies $q \underset{\mu}{\equiv} r$, for all $q, r \in Q$.

Example 1.2

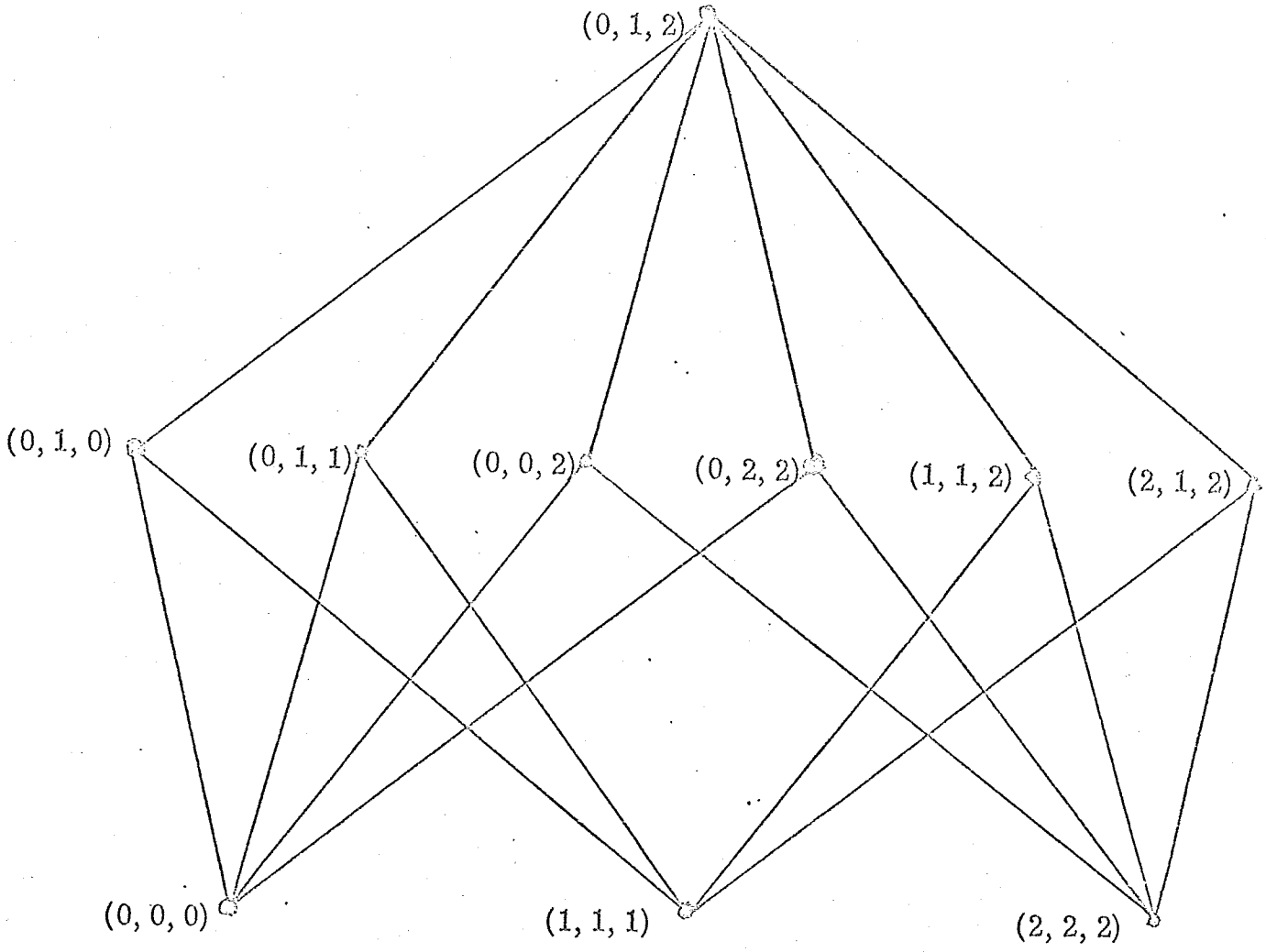
Let $Q = \{0, 1, 2\}$ and denote a stable fault $\mu \in S(Q)$ as the triple

$$(\mu(0), \mu(1), \mu(2)).$$

If further we let $\Pi_\mu$ denote the partition of Q corresponding to the equivalence relation $\underset{\mu}{\equiv}$ then, for each $\mu \in S(\{0, 1, 2\})$, $\mathcal{R}(\mu)$ and $\Pi_\mu$ are given by the following table.

| $\mu$ | $\mathcal{R}(\mu)$ | $\Pi_\mu$ |
|---|---|---|
| (0, 1, 2) | {0, 1, 2} | $\{\overline{0}, \overline{1}, \overline{2}\}$ |
| (0, 1, 0) | {0, 1} | $\{\overline{1}, \overline{02}\}$ |
| (0, 1, 1) | {0, 1} | $\{\overline{0}, \overline{12}\}$ |
| (0, 0, 2) | {0, 2} | $\{\overline{2}, \overline{01}\}$ |
| (0, 2, 2) | {0, 2} | $\{\overline{0}, \overline{12}\}$ |
| (1, 1, 2) | {1, 2} | $\{\overline{2}, \overline{01}\}$ |
| (2, 1, 2) | {1, 2} | $\{\overline{1}, \overline{02}\}$ |
| (0, 0, 0) | {0} | $\{\overline{012}\}$ |
| (1, 1, 1) | {1} | $\{\overline{012}\}$ |
| (2, 2, 2) | {2} | $\{\overline{012}\}$ |

Accordingly, the natural ordering of these faults has the following Hasse diagram:


(0, 1, 2)
(0, 1, 0)
(0, 1, 1)
(0, 0, 2)
(0, 2, 2)
(1, 1, 2)
(2, 1, 2)
(0, 0, 0)
(1, 1, 1)
(2, 2, 2)

By Theorem 1.11 and the preceding example it should be obvious that a stable fault is uniquely specified by its range $\mathcal{R}(\mu)$ and induced equivalence relation $\underset{\mu}{\equiv}$. Also, it is relatively easy to determine the number of stable faults that are possible for a machine with n states.

Theorem 1.12

$$\text{If } |Q| = n \text{ then } |S(Q)| = \sum_{k=1}^{n} \binom{n}{k} k^{n-k}$$

Thus, for example, of the 10 billion possible faults of a machine with 10 states, 2,137,921 are stable faults.

Let us now relate the notion of a stable fault to some of the concepts introduced earlier. First of all, we note that if one stable fault can follow (Def. 19) another then the composite fault is stable, that is,

Theorem 1.13

If $\mu, \gamma$ are stable faults of M and $\gamma$ can follow $\mu$ then $\gamma \cdot \mu$ is a stable fault of M.

Proof: Let $q \in \mathcal{R}(\gamma \cdot \mu)$. Since $\gamma$ can follow $\mu$, $q \in \mathcal{R}(\mu)$ (Theorem 1.1) and since $\mu$ is stable, $\mu(q) = q$ (Theorem 1.10). Thus

$$\gamma(\mu(q)) = \gamma(q).$$

But $q \in \mathcal{R}(\gamma \cdot \mu)$ implies $q \in \mathcal{R}(\gamma)$ and as $\gamma$ is also stable

$$\gamma(q) = q.$$

Combining the above identities

$$\gamma(\mu(q)) = q$$

for all $q \in \mathcal{R}(\gamma \cdot \mu)$ or equivalently (by Theorem 1.10) $\gamma \cdot \mu$ is stable.

Generalizing Theorem 1.13 it follows that if $(\mu_1, \mu_2, \ldots, \mu_n)$ is a succession of stable faults then $\mu_n \cdot \mu_{n-1} \cdots \mu_1$ is stable.

Also of interest are the conditions under which the order of occurrence of faults is irrelevant or, more formally, when faults commute (with respect to the operation of composition). This question is answered by the following theorem.

Theorem 1.14

If $\mu$ and $\gamma$ are stable faults of M then the following statements are equivalent:

i) $\gamma \cdot \mu = \mu \cdot \gamma$

ii) $\mathcal{R}(\gamma \cdot \mu) \subseteq \mathcal{R}(\mu)$, $(\mu \cdot \gamma) \subseteq \mathcal{R}(\gamma)$,

$\equiv_{\gamma} \subset \equiv_{\gamma \cdot \mu}$, and $\equiv_{\mu} \subset \equiv_{\mu \cdot \gamma}$.

iii) $\mu \cdot (\gamma \cdot \mu) = \gamma \cdot \mu$ and $\gamma \cdot (\mu \cdot \gamma) = \mu \cdot \gamma$

iv) $\gamma \cdot \mu \leq \mu$ and $\mu \cdot \gamma \leq \gamma$.

Condition iv) gives an interesting characterization of commutativity in that it relates directly to the natural ordering of stable faults. If $\gamma$ can follow $\mu$ we can interpret $\mu \leq \gamma$ as meaning $\mu$ "dominates" $\gamma$ in the sense that $\gamma$ has no further effect once $\mu$ has occurred. Accordingly, by part iv) of Theorem 1.14, the order in

which $\mu$ and $\gamma$ occur is irrelevant if and only if the succession $(\mu, \gamma)$ dominates $\mu$ and $(\gamma, \mu)$ dominates $\gamma$.

Finally, with regard to fault masking, if we examine the most restrictive type of masking (Q-masking) under the assumption that a fault is stable we find that a rather easily tested condition is both necessary and sufficient for Q-masking (compare with Theorem 1.6).

Theorem 1.15

If M is a machine with states Q and $\mu$ is a stable fault of M then

$$\mu \text{ is Q-masked iff } \underset{\mu}{\equiv} \subseteq \equiv.$$

($\equiv$ denotes the relation of state equivalence on Q.)

In other words, a stable fault $\mu$ of M is masked if and only if $\mu(q) = \mu(r)$ implies $q \equiv r$, for all $q, r \in Q$.

Corollary

If M is a reduced machine then no proper stable fault of M can be Q-masked.

Theorem 1.15 is an important result in the sense that the restrictive nature of Q-masking is now quite obvious. If only a relatively few stable faults are to be masked, it is conceivable that one could Q-mask all faults. On the other hand it appears that Q-masking a reasonable number of stable faults will be very difficult and, in many cases, impossible.

"Reliability"?

## 2. FAULT-MASKING IN COMBINATIONAL NETWORKS

Before any interesting questions about fault masking can be explored, a model must be developed that can be used efficiently to analyze the fault-masking properties of networks with redundancy. The ideas of fault and failure should arise naturally in the model.

### THE CONCEPT OF NET

The term net is intended to include all layout information about a system. It must identify all the devices in a system, and all the signals that can appear in the system. It must also account for the manner in which devices are interconnected.

Definition

An $(n, m, k, \ell)$-net is a 2-tuple

$$P = (D, S)$$

D is a connected loop-free directed graph (digraph) with n+m+k labeled points and $\ell$ labeled lines. Exactly n of the points, called input terminals (or just inputs), have indegree 0 and outdegree greater than or equal to 1. Similarly, exactly m points, called output terminals (or just outputs), have outdegree 0 and indegree 1. The remaining k points are called nodes, and have both indegree and outdegree greater than or equal to 1. In general, node i will have indegree $n_i$ and outdegree $m_i$ where $n_i$ and $m_i$ are positive integers.

S is an $\ell$-tuple of sets.

When interpreting the model in a physical sense, n is the number of inputs, m is the number of outputs, k is the number of internal nodes, and $\ell$ is the number of interconnecting lines. A node may be a simple logic gate or a complex sub-system. Hence, if desired, a complex system may be decomposed into several sub-systems for ease of analysis. The model can be applied to the decomposition by allowing each sub-system to be considered as a node. Then, the model may be applied separately to each sub-system. Analysis of a modular structure at any level of complexity is possible.

The signals that may appear on the $i^{th}$ line in the system is the set appearing as the $i^{th}$ coordinate of the $\ell$-tuple S. Each line leaving the same input node must have the same signal set; otherwise, it is permissible for each of the sets to be different. However, in the usual switching circuit interpretation, each coordinate of S will be the binary set $B = \{0, 1\}$.

Definition

The input space associated with node i, called $I_i$, is defined as the cartesian product of the signal sets specified by S for the input lines to node i. The coordinate sets for the input space shall be taken in the order of ascending line labels, for consistency.

Definition

The output space associated with node i, called $O_i$, is defined

as the cartesian product of the signal sets on the output lines, again taken in order of ascending line labels for convenience.

Definition

$E = \{(g_1, g_2, \ldots, g_k) \mid g_i \text{ is a mapping from } I_i \text{ into } O_i \text{ for } 1 \leq i \leq k\}$.

In the usual switching circuit interpretation, $g_i$ in the above definition is a mapping from $B^{(n_i)}$ into $B^{(m_i)}$. The set E represents all theoretically possible combinations of nodal actions in the net. In many physical systems, only a small fraction of these will ever occur.

A COMBINATIONAL NETWORK MODEL

Definition

An (n, m, k, $\ell$)-combinational network, is a 3-tuple

$$C = (P, F, b)$$

where P is an (n, m, k, $\ell$)-net

$F \subseteq E$, F is called the fault set

$b \in F$, b is called the 0-fault

Definition

A proper fault of the combinational network C = (P, F, b) is an element of the set F-{b}.

Note that the 0-fault of C is not a proper fault of C. In the usual interpretation, the 0-fault corresponds to the fault-free condition.

## THE BEHAVIOR OF A COMBINATIONAL NETWORK

### Definition

The input space, I, for combinational network C = (P, F, b) is defined as the cartesian product of the signal sets on the lines of P leaving the input terminals taken in order of increasing node label. There is one coordinate in the input space for each terminal; hence, the input space is an n-dimensional space. In the usual switching theory interpretation, $I = B^{(n)}$.

### Definition

The output space, O, for combinational network C is defined as the cartesian product of the signal sets appearing on the lines of P that terminate on the output terminals taken in order of increasing node labels.

There is one coordinate for each output terminal; hence, the output space is an m-dimensional space. In the usual switching circuit interpretation, $O = B^{(m)}$.

### Definition

$T_p = \{\text{mappings from I into O}\}$

The net P in a combinational network C = (P, F, b) induces a mapping $\alpha$, called the net mapping, from F into $T_p$ in a natural way. When a fault $f = (f_1, f_2, \ldots, f_k)$ occurs in a net P, the net performs some mapping from I into O. This mapping is the image of the fault f under $\alpha$.

Definition

The function of the network C is the element $t = \alpha(b)$.

Definition

The function set of network C is the set $T = \alpha(F)$.

Definition

A malfunction of C is an element of the set $T-\{t\}$.

In a physical interpretation, the function of the system is the behavior of the system when it is fault-free. The function set is the set of all possible behaviors that can result from faults in the network. A malfunction is some behavior different from the fault-free behavior.

## A HIERARCHY OF FAULT DIAGNOSIS CONCEPTS

Consider the combinational network

$$C = (P, F, b).$$

Recall that $b = (b_1, b_2, \ldots, b_k)$ is the fault-free condition in the network.

Definition

$$\forall\ f = (f_1, f_2, \ldots, f_k) \in F, \quad \text{let } K_f = \{i \mid 1 \leq i \leq k,\ f_i \neq b_i\}$$

$$\forall\ f \in F, \quad \text{let } J_f = \{f' \mid f' \in F,\ K_f = K_{f'}\}$$

Fault Diagnosis Concepts

$f \in F$ is masked iff $\alpha(f) = t$

$f \in F$ is detectable iff $\alpha(f) \neq t$

$f \in F$ is locatable iff $(f' \in F,\ \alpha(f) = \alpha(f') \Rightarrow f' \in J_f)$

$f \in F$ is completely diagnosable iff $(f' \in F,\ \alpha(f) = \alpha(f') \Rightarrow f = f')$

Note that the 0-fault is always masked, and that detectable faults are always proper faults. On the other hand, the 0-fault may or may not be locatable and may or may not be completely diagnosable. If the network has a non-empty set of proper masked faults, then the 0-fault is not locatable and not completely diagnosable. However, if the set of proper masked faults is empty, then the 0-fault is locatable and completely diagnosable.

Definition

A failure of C is a detectable fault of C.

It should be obvious that

$\{\text{masked faults of C}\} \cup \{\text{failures of C}\} = F$ and that

$\{\text{masked faults of C}\} \cap \{\text{failures of C}\} = \phi$

A masked fault in the usual interpretation is a change in structure of the system from its fault-free structure that preserves the fault-free behavior of the system. A failure of a system is a change in structure that causes a change in behavior.

In terms of actual diagnosis, there is no experiment that may be performed on the system terminals to distinguish a masked fault structure from the 0-fault structure. However, there always exists a terminal experiment to determine the presence or absence of a detectable fault structure, although the particular detectable fault, if present, may not be known.

If f is a locatable fault, with associated sets $K_f$ and $J_f$, then there exists a terminal experiment that will verify the presence or absence of a fault in the set $J_f$, although the particular member of the set $J_f$ that is present may not be revealed.

If f is a completely diagnosable fault, then there exists a terminal experiment to verify the presence or absence of the fault f.

## SINGLE FAULT ANALYSIS

The masking of single faults in a combinational network is often of extreme importance because the probability of a single fault is usually much greater than the probability of a multiple fault. For this reason, it is frequently desirable to protect the circuit against the occurrence of certain single faults.

### Definition

A fault $f = (f_1, f_2, \ldots, f_k)$ in an $(n, m, k, \ell)$-combinational network is called a single fault if

(1) $f_i \neq b_i$ for some i with $1 \leq i \leq k$

and

(2) $f_j = b_j$ $j \neq i$ with $1 \leq j \leq k$

Note that all single faults are by definition proper faults.

## A SIMPLE TWO NODE SYSTEM

The analysis begins by considering a two-node combinational network. This special type of network is easier to work with than more general types, and the results obtained can be used directly to

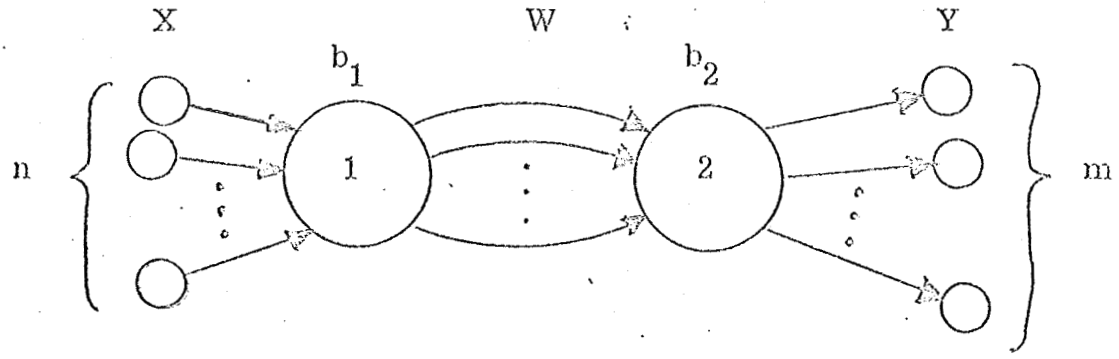



Figure 2.1

Graph of a simple two-node system.

solve the single fault masking problem in general networks. Figure 2.1 displays the graph, P, of an $(n, m, 2, \ell)$-combinational network, C, with

$$C = (P, F, b)$$

$$P = (D, S)$$

where

P is as shown in Figure 2.1

$$b = (b_1, b_2)$$

$$F = \{(f_1, f_2) \mid (f_1, f_2) \in E,\ f_1 = b_1,\ f_2 \neq b_2\}$$
$$\cup \{(f_1, f_2) \mid (f_1, f_2) \in E,\ f_1 \neq b_1,\ f_2 = b_2\}$$
$$\cup \{b\}$$

The lines are not labeled because we are allowing arbitrary signal sets. For this graph,

$$n_1 = n$$

$$m_1 = n_2$$

$$m_2 = m$$

In the usual application, $t = b_2 \cdot b_1 = b_2 b_1$

$$X = B^{(n)} = B^{(n_1)}$$

$$W = B^{(m_1)} = B^{(n_2)}$$

$$Y = B^{(m_2)} = B^{(m)}$$

However, our analysis will be completely general in that arbitrary signal sets are allowed.

Since $b_2$ is a mapping from W into Y, $b_2$ induces an equivalence relation, $R_{b2}$, on the set W defined as follows:

$$w_1 \underset{R_{b2}}{\equiv} w_2 \text{ if } b_2(w_1) = b_2(w_2)$$

The set of equivalence classes of $R_{b2}$ are the blocks of a partition of W called $\pi_{b2}$.

Theorem 2.1

A single fault at node 1, $f = (f_1, b_2)$, is masked if and only if

$$f_1(x) \underset{R_{b2}}{\equiv} b_1(x)$$

Corollary 2.1.1

A single fault at node 1 is a failure of C if and only if there exists an $x \in X$ such that

$$f_1(x) \underset{R_{b2}}{\not\equiv} b_1(x)$$

The corollary follows directly from Theorem 1 because of the exclusive character of masked faults and failures.

The next two theorems answer important questions about complete fault-masking and complete fault-detecting.

Theorem 2.2

All single faults at node 1 are masked iff $b_2$ is a constant function.

Theorem 2.3

All single faults at node 1 are failures iff

$$|R_{b2}[b_1(x)]| = 1 \qquad \forall x \in X$$

Corollary 2.3.1

If $b_2$ is a 1-1 function from W into Y, then all single faults of C at node 1 will be failures.

A systematic method for counting the number of single faults at node 1 that are masked will now be developed.

Definition

$$F_1 = \{f \mid f \in F, \text{ f is a single masked fault at node 1}\}$$

Definition

For every $y_i \in b_2\, b_1(X)$, define

$$c_i = |b_2^{-1}(y_i)|$$

and

$$d_i = |(b_2\, b_1)^{-1}(y_i)|$$

Theorem 2.4

$$|F_1| = -1 + \prod_{i \mid y_i \in b_2 b_1(X)} c_i^{d_i}$$

The minus 1 appears because the second term counts the 0-fault which is not a single fault.

Note that $|F_1| = 0$ only if every $c_i$ is 1. From the definition of $c_i$, this implies that the pre-image of each element of $b_2 b_1(X)$ in the space

W must be a single element. This is the same as saying that every equivalence class of $R_{b2}$ that intersects the range of $b_1$ must consist of a single element of W. The counting theorem is seen to support Theorem 2.3.

If $b_2$ is a 1-1 function, then all $c_i$ must be 1. Our counting theorem indicates no single faults masked at node 1 under this condition as stated by Corollary 2.3.1.

Corollary 2.4.1

The number of single faults at node 1 that are failures is given by

$$|W|^{|X|} - \prod_{i|y_i \epsilon b_2 b_1(X)} c_i^{d_i}$$

When $b_2$ is a constant function, there is only one element in the set $b_2 b_1(X)$. Also, $b_2^{-1}[b_2 b_1(X)] = W$. This means that $c = |W|$ and $d = |X|$, hence Corollary 2.4.1 states that there are no single faults at node 1 that are failures. This implies that all single faults must be masked, as stated by Theorem 2.2.

Also, since $\sum c_i \leq |W|$ and $\sum d_i = |X|$, we see that the only time this expression is zero is under the conditions just stated.

The following theorems and corollaries answer similar questions about node 2.

Theorem 2.5

A single fault of C at node 2, $f = (b_1, f_2)$ is masked iff

$$f_2|\mathcal{R}_{b1} = b_2|\mathcal{R}_{b1}$$

Corollary 2.5.1

A single fault of C at node 2, $f = (b_1, f_2)$, is a failure of C iff

$$f_2 | \mathcal{R}_{b1} \neq b_2 | \mathcal{R}_{b1}$$

Lemma 2.6.1

The proper fault set of C is empty, iff $|Y| = 1$.

Theorem 2.6

It is impossible for all single faults of C at node 2 to be masked except in the singular case when $|Y| = 1$, in which case the set of single faults at node 2 is empty.

Theorem 2.7

All single faults of C at node 2 are failures if and only if $b_1$ is onto W.

Theorem 2.5 adds substance to the often presented hypothesis that error in the final gate in a network will always contribute some error to the system. In particular, we show that it is never possible to mask all single faults at the final output node except in the trivial case when the system doesn't "do" anything.

Before presenting a theorem to enumerate the number of single faults at node 2 that are masked, a couple of definitions are required.

Definition

$$s = |Y|$$

Definition

$$r = |\bar{A}_{b1}|$$

Definition

$$F_2 = \{f | f \in F,\ f \text{ is a single masked fault at node 2}\}$$

Theorem 2.8

$$|F_2| = -1 + s^r$$

Corollary 2.8.1

The number of detectable single faults at node 2 is given by

$$|Y|^{|W|} - s^r.$$

THE GENERAL SYSTEM

To describe the faults masked at a particular node in a general combinational network, we first seek a general form into which any network may be put for analysis. Such a form is shown in Figure 2.2. The following theorems stated in terms of the general form may be used to analyze faults in an arbitrary system.

The general form is an $(n, m, 3, \ell)$-combinational network

$$C = (P, F, b)$$

$$P = (D, S)$$

where

D is as shown in Figure 2.2

S is an arbitrary $\ell$-tuple of sets

$$b = (b_1, b_2, h)$$

$$F = \{(f_1, f_2, f_3) | (f_1, f_2, f_3) \in E,\ f_1 = b_1,\ f_3 = h\}$$

In describing fault behavior at node 2, it is convenient to decompose the h mapping.

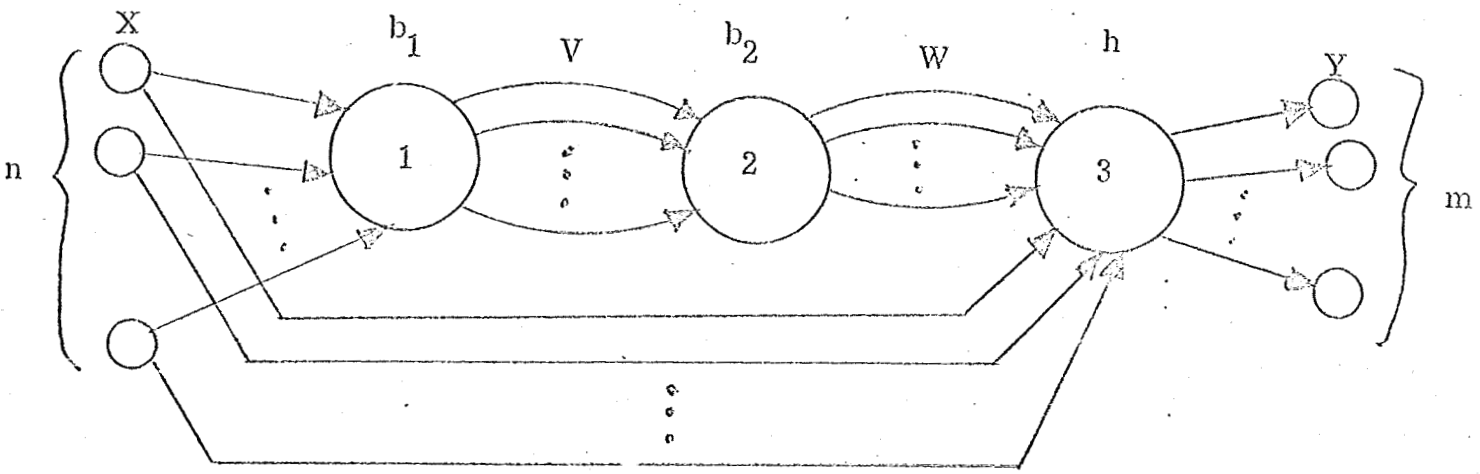



Figure 2. 2

General form for single fault analysis

at a specified node.

Definition

For each input combination, x, define the mapping $h_x$ as follows:

$$h_x: \quad W \rightarrow Y$$
$$w \rightarrow h(x, w)$$

The mapping $h_x$ completely describes the action of node 3 on space W when the input combination is x. With a knowledge of the effect that nodes 1 and 2 have on the input combination x, we may use $h_x$ to complete the description of the circuit action on x.

Each mapping $h_x$ induces an equivalence relation on the set W.

Definition

$R_x$ is the equivalence relation on W induced by the mapping $h_x$ from W into Y. This equivalence relation is obtained in the usual way.

$$w_1 \underset{R_x}{\equiv} w_2 \quad \text{if} \quad h_x(w_1) = h_x(w_2)$$

In a physical system, $w_1$ is $R_x$ equivalent to $w_2$ if and only if the system output when the input configuration is x and the W configuration is $w_1$ is equal to the system output when the input configuration is x and the W configuration is $w_2$.

Theorem 2.9

A single fault, $f = (b_1, f_2, h)$, at node 2 in the general form is masked iff

$$b_2 b_1(x) \underset{R_x}{\equiv} f_2 b_1(x) \qquad \forall\, x \in X$$

Corollary 2.9.1

A single fault, $f = (b_1, f_2, h)$, at node 2 in the general form is a failure iff

$$b_2 b_1(x) \not\underset{R_x}{=} f_2 b_1(x) \quad \text{for some } x \in X$$

Theorem 2.10

All single faults at node 2 in the general form are masked iff $h_x$ is a constant function for all $x \in X$.

Theorem 2.11

All single faults at node 2 in the general form are failures iff

(1) $b_2 b_1(x) \underset{R_x}{=} w \Rightarrow b_2 b_1(x) = w$

and

(2) $|W| \neq 1 \Rightarrow b_1$ is onto V

Before introducing the counting theorem for the general case, it is necessary to extend some definitions used for the simple two node system.

First, we define a counting constant for each element in the set V. In the two node case, we defined a counting constant for each $x \in X$, since X was then the input space to the node of interest. This was done indirectly through the constants $c_i$ and $d_i$. We must now account for both kinds of masked faults encountered during the analysis of the two node case. One type arose because of the action of the following node (here h), and the other type because of the action of the preceding node,

(here $b_1$). Both may be accounted for in the same process, as we shall show.

Consider first the range of the mapping $b_1$. Let v be any element in this range. The counting constants for elements in the range of $b_1$ cover the cases where the mapping h allows some variation in the image of v under $b_2$. These arise in the same manner as those that are counted by the $c_x$ constants in the two node case. However, here $b_1^{-1}(v)$ may contain more than one element, hence $h_x$ must allow the image of v to vary within an equivalence class of $R_x$ for each $x \in b_1^{-1}(v)$ before any fault that changes the image of v from its value under $b_2$ can be masked. Hence, to find how many images of v produce the same system output, we must look at a new equivalence relation on W.

Definition

For every $v \in \mathcal{R}_{g1}$, define the equivalence relation, $R_v$, on the set W as follows:

$$w_1 \underset{R_v}{\equiv} w_2 \text{ if } h_x(w_1) = h_x(w_2) \qquad \forall\, x \in g_1^{-1}(v)$$

Definition

For every $v \in \mathcal{R}_{b1}$, define the counting constant $c_v$ as follows:

$$c_v = \left| R_v[b_2(v)] \right|$$

We recall from the two node case that elements of V not in the range of $b_1$ may be mapped onto any element of W. This can be done

in $|W|$ ways, thus the counting constant for elements not in the range of $b_i$ is defined as follows:

Definition

For every $v \in \overline{\mathcal{R}}_{b1}$, define the counting constant, $c_v$, as follows:

$$c_v = |W|$$

Theorem 2.12

The number of single faults at node 2 that are masked in the general form is given by

$$|F_2| = -1 + \prod_{v \in V} c_v$$

Corollary 2.12.1

The number of single faults at node 2 that are failures is given by

$$|W|^{|V|} - \prod_{v \in V} c_v$$

We see that the expression in Theorem 2.12 is zero only when all $c_v$ are equal to 1. All $c_v$ are equal to 1 only when

(1) each equivalence class of $R_v$ intersecting the range of $b_2$ has only one element

and

(2) $|W| = 1$, or $\mathcal{R}_{b1} = V$

Statements (1) and (2) above are equivalent to the conditions (1) and (2) of Theorem 2.11. The counting theorem is thus consistent with Theorem 2.11, because if the number of single faults masked is

zero, then all single faults must be failures.

The expression in Corollary 2.12.1 is zero only when each $c_v$ is equal to $|W|$. This can only happen if every $h_x$ is a constant function. Hence, Corollary 2.12.1 is consistent with Theorem 2.10.

# 3. DIAGNOSIS OF SEQUENTIAL MACHINES

Design of fault detecting experiments for a sequential machine can be greatly simplified if the machine possesses some distinguishing sequences thus permitting unique identification of the initial state at each step of the experiment. Unfortunately, not every sequential machine has distinguishing sequences. The problem considered here is to obtain, for an arbitrary sequential machine, a modified machine which contains the original machine and possesses some special distinguishing sequences.

The sequential machines considered here are assumed to be strongly connected, reduced, and the malfunctions which occur in the circuit do not increase the number of states in the machine.

The design of a diagnosable machine in which every input sequence of a certain length is a distinguishing sequence was first studied by Kohavi and Lavallee [4]. A machine which possesses this property is called "definitely diagnosable" (D.D.). They have proposed a method of constructing such a machine from an arbitrary sequential machine by augmenting additional output logic purely for the purpose of testing. However, definite diagnosability is not a necessary condition for designing short fault detecting experiments nor is it the most economical method. A closer examination indicates that a machine having a short distinguishing sequence is generally sufficient for designing such experiments.

First, we generalize the D.D. property and obtain a classification of machines according to various degrees of diagnosability and homability from a machine-theoretical viewpoint. This may lend some insight into various levels of machine diagnostic capability.

Second, a method of constructing a machine to possess a repeated symbol distinguishing sequence by augmenting its output symbols is presented. Machines so constructed are seen to have a reduced upper bound on the minimum length distinguishing sequence and consequently have shorter fault detecting experiments.

Finally, a second method is presented which concerns the solution of the same problem by augmenting the machine input symbols. This is done by constructing a reduced single-input machine and appending it to the original machine. It is shown that it is always possible to construct an n-state, k-output single-input machine so that its distinguishing sequence is of minimal length, i.e. of length $[\log_k n]$ where $[\log_k n]$ is the least integer greater than or equal to $\log_k n$ and both k and n are powers of 2.

## BASIC DEFINITIONS

The following definitions are based on Mealy type sequential machines. The notation used is consistent with that of Section 1.

Definition

Let M be a machine and $x \in I^{\dagger}$. We say that x is a distinguishing sequence (D. S.) for M if

$$\beta_q(x) = \beta_r(x) \Longrightarrow q = r \qquad \forall\, q, r \in Q$$

Definition

Let M be a machine and $x \in I^{\dagger}$. We say that x is a homing sequence (H. S.) for M if $\forall\, q, r \in Q$

$$\beta_q(x) = \beta_r(x) \Longrightarrow \bar{\delta}(q,x) = \bar{\delta}(r,x)$$

Definition

Let M be a sequential machine and $q, r \in Q$ $(q \neq r)$. We say that q and r converge under sequence $x \in I^{\dagger}$ if

$$\beta_q(x) = \beta_r(x) \text{ and } \bar{\delta}(q,x) = \bar{\delta}(r,x)$$

In case only $\bar{\delta}(q,x) = \bar{\delta}(r,x)$ holds, we say that q and r merge under x.

If no pair of states converge, we say that the machine is convergence free (C. F.)

Definition

Let M be a sequential machine. Then M is said to be definitely diagnosable (D. D.) if there is an integer $\ell$ such that every input sequence of length $\ell$ is a distinguishing

sequence. The least such integer $\ell$ is called the order of diagnosability.

MACHINE CLASSIFICATIONS

We first observe the following alternative characterization of the definitely diagnosable property.

Theorem 3.1

If M is an n-state machine, then M is definitely diagnosable if and only if every input sequence of length greater than or equal to $\frac{n(n-1)}{2}$ is a distinguishing sequence of M.

Lemma 3.1

(Hennie [3]) An input sequence is a D.S. iff it is a H.S. that causes no convergence.

Lemma 3.2

If there exists k such that $\ell g(x) = k$ implies that x is a D.S., then no state pair converges under any H.S.

It is well known that every distinguishing sequence is also a homing sequence. However, the converse is not generally true. If a machine is definitely diagnosable then the latter is also true. This is stated in the next theorem.

Theorem 3.2

If a machine is D.D. then every H.S. is also a D.S.

Note that the converse of Theorem 3.2 is not true. For example, a machine which does not have any D.S. and H.S. satisfies the above property vacuously but this machine is clearly not D.D.

Theorem 3.3

If M is D.D. then M is convergence free or simply abbreviated as

$$\text{D.D.} \Longrightarrow \text{C.F.}$$

We say that a machine is diagnosable if it has a distinguishing sequence. Similarly, we say that a machine is homable if it has a homing sequence. Next, we define a notion of definitely homable analogous to that of definitely diagnosable.

Definition

A machine M is definitely homable (D.H.) if there is an integer k such that every input sequence of length k is a homing sequence. The least such integer k is called the order of homability.

Note that the least such k is $\leq n(n-1)/2$ as can be seen from the fact that there are at most $n(n-1)/2$ nodes in the testing graph of an n-state machine. The testing graph of a machine is constructed from the set of all state pairs which yield the same output response for some input and their non-merging successor state pairs.

The D. H. property corresponds essentially to the loop free condition in the testing graph. The next theorem relates properties of D. D. and D. H.

Theorem 3. 4

Any definitely diagnosable machine is also definitely homable i.e. D. D. $\Longrightarrow$ D. H.

The converse of theorem 3. 4 is not true. One simple example is a D. H. machine which has some state pair convergence.

Theorem 3. 5

If no two different states converge in a reduced machine, then every H. S. is also a D. S.

Theorem 3. 6

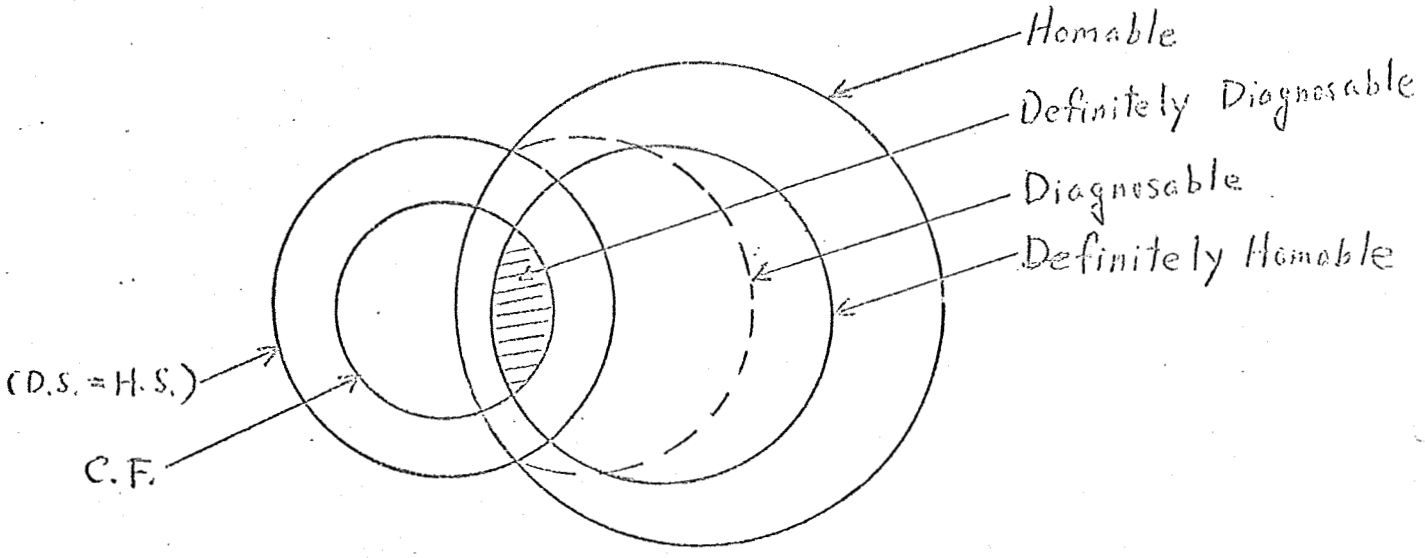
A machine is D. D. iff it is both C. F. and D. H.

Let us call a distinguishing sequence or homing sequence proper if no proper subsequence of it is also a distinguishing sequence or homing sequence respectively. The notion of definite homability may be useful in the sense that its proper D. S. have the same upper bound as that of a D. D. machine. This is characterized by the following theorem.

Theorem 3. 7

Let M be an n-state D. H. machine. If M is also diagnosable, then the upper bound of its proper D. S. is $n(n-1)/2$.

It is clear that if a machine is definitely homable then it is also homable. If a machine is diagnosable then it is also homable. To summarize what we have done so far, a Venn diagram is constructed to represent the hierarchy of machine classes.




Note that the class of machines as defined in Theorem 3.7 is the intersection of the class of diagnosable machines and that of definitely homable machines.

## CONSTRUCTION OF A DIAGNOSABLE MACHINE WITH A REPEATED SYMBOL DISTINGUISHING SEQUENCE BY AUGMENTING OUTPUT LOGIC

Recall that we mentioned earlier that although "being reduced" is a sufficient condition for the existence of a homing sequence in a

sequential machine, such is not the case for distinguishing sequences. Lemma 3.1 gives us a sufficient condition for the existence of a D.S. This is re-stated in terms of the diagnosable notion in the next theorem.

Theorem 3.8

A sequential machine is diagnosable if it has a homing sequence x such that no pair of states converge under x.

However, if the machine has only one input symbol, then "being reduced" is also sufficient for the existence of a D.S. The following theorem formalizes the above observation.

Theorem 3.9

If an n-state, single input machine is reduced, then it has a proper D.S of length at most n-1.

A distinguishing sequence which has only one input symbol is called a repeated symbol distinguishing sequence (R.S.D.S.). Theorem 3.9 provides a convenient way of checking whether a sequential machine has any R.S.D.S. and constructing one if there is none. The following corollary will characterize this property.

Corollary 3.9.1

A sequential machine has a repeated symbol distinguishing sequence if and only if it has a reduced single-input submachine.

Here by a single-input submachine of a machine we mean a submachine whose state table is a column of the state table of the complete machine.

Thus to see whether a machine M has any R.S.D.S., it is only necessary to examine whether any of its single-input submachines $M_1, M_2, \ldots, M_m$ is reduced. Since a reduced single-input machine is a definitely diagnosable machine, it is only necessary to make some $M_{i_j}$ definitely diagnosable if we want to obtain an $i_j$ - R.S.D.S. A general procedure for constructing a definitely diagnosable machine from the original machine by augmenting the original output symbols has been outlined by Kohavi and Lavallee [4].

In choosing an input symbol to obtain a reduced single-input submachine, optimization criteria of choosing either one that gives rise to minimal additional output logic, or one that results in the shortest D.S. may be used. To obtain minimal additional output logic in the final realization, it is generally desirable to look for a single-input submachine of the original machine whose largest equivalence class induced by the partition of states according to their output response is minimal among all the single-output submachines. The length of D.S. would be reduced if we use more additional output symbols. Here a compromise is generally needed between acceptable length of a D.S. and the amount of additional hardware required.

The upperbound of the length of a checking experiment using a repeated symbol distinguishing sequence is $\xi$:

$$\xi \leq nm + n(m-1)\ell + \ell + (m-1)(n-1)^2$$

where

n = number of states

m = number of input symbols

$\ell$ = length of proper D. S. [which is $\leq (n-1)$]

In the general case, it may be possible to construct a diagnosable machine which requires less additional hardware than that required to construct a diagnosable machine with a R. S. D. S. However, the upperbound of the length of this kind of D. S. may be quite large.

## CONSTRUCTION OF A DIAGNOSABLE MACHINE WITH A REPEATED SYMBOL DISTINGUISHING SEQUENCE USING ADDITIONAL INPUT LOGIC

Consider the machine M whose state table is shown below

M =

| | a | b |
|---|---|---|
| $q_1$ | $q_1/0$ | $q_2/0$ |
| $q_2$ | $q_1/0$ | $q_3/0$ |
| $q_3$ | $q_1/0$ | $q_4/0$ |
| $q_4$ | $q_1/1$ | $q_1/0$ |

This machine does not have any distinguishing sequence. Now let us construct a reduced 1-column machine and append it to the original state table. The modified machine is shown below with the appended column on the right of the state table:

$M' =$

| | a' | b' | c |
|---|---|---|---|
| $q_1$ | $q_{1/0}$ | $q_{2/0}$ | $q_{2/0}$ |
| $q_2$ | $q_{1/0}$ | $q_{3/0}$ | $q_{4/1}$ |
| $q_3$ | $q_{1/0}$ | $q_{4/0}$ | $q_{3/0}$ |
| $q_4$ | $q_{1/1}$ | $q_{1/0}$ | $q_{1/1}$ |

This modified machine has a distinguishing sequence of cc.

The upperbound of the length of the fault detecting experiment is now modified as shown below

$$\xi \leq n(m+1) + (nm+1)\ell + m(n-1)^2$$

where

n = number of states

m = number of original input alphabets

$\ell$ = length of the distinguishing sequence used.

We know that for an n-state, k-output machine, the lower bound on the length of a D.S. is $[\log_k n]$. We will show that in case both k and n are powers of 2, we can always construct a single-input machine which has a D.S. of this length.

For purposes of illustration, let us consider the binary output case of a s-stage shift register with the last stage memory output being monitored externally. To see what state the machine was initially in, it is only necessary to shift the register s times. Thus this circuit corresponds to a $2^s$ state, single-input and binary output

machine with a distinguishing sequence of length s. Appending such a single-input machine to a given machine is equivalent to a modification that causes the machine to act as a shift register under certain inputs. The above observation can be formally stated as follows:

Theorem 3.10

There is a $n = 2^s$ state, binary output, single-input machine which has a distinguishing sequence of length s.

Thus, any $2^s$ state, binary output machine can be made to possess a repeated symbol distinguishing sequence of length s by augmenting to the original machine a reduced n-state, single input machine which satisfies Theorem 3.10.

In general if both the number of output symbols and the number of states are powers of 2, we can always find a single-input machine which has a distinguishing sequence of the shortest possible length. This is stated in the next theorem.

Theorem 3.11

There is a $k = 2^t$ output, $n = 2^s$ state, single-input machine which has a distinguishing sequence of length $\lceil \log_k n \rceil = \lceil \frac{s}{t} \rceil$.

The upperbound of the length of the fault detecting experiment using the above construction of providing a "diagonsable input" is thus:

$$\xi \leq n(m+1) + (nm+1)[\log_k n] + m(n-1)^2$$

The last term in the equation above comes from the possible need of applying transfer sequences in the experiment. This last term may be decreased if we provide a reset input to the modified machine. The upper-bound of the length of the fault detecting experiment in this case of providing both diagnosable input and reset input becomes:

$$\xi \leq n(m+2) + [n(m+1)+1][\log_k n] + \frac{(m+1)n(n-1)}{2} + n$$

COMPARISON OF UPPERBOUNDS

Let us now compare the upperbounds of the length of the fault detecting experiment derived in this report to that given by Kohavi and Lavallee [4].

Let

$\xi_{DD}$ = Bound of D. D. machines.

$\xi_{DD}$ = Bound of diagnosable machines which have a R. S. D. S. by augmenting output logic.

$\xi_{ID}$ = Bound of diagnosable machines which have a R. S. D. S. by appending a single-input machine.

$\xi_{IDR}$ = Bound as $\xi_{ID}$ with additional reset input.

n = number of states.

m = number of input symbols in the original machine.

k = number of output symbols.

Then we have

$$^{*}\xi_{DD} \leq nm+(nm+1)\frac{n(n-1)}{2} + (m-1)(n-1)^2$$

$$\xi_{OD} \leq nm + [n(m-1)+1](n-1) + (m-1)(n-1)^2$$

$$\xi_{ID} \leq n(m+1) + (nm+1)[\log_k n] + m(n-1)^2$$

$$\xi_{IDR} \leq n(m+3) + [n(m+1)+1][\log_k n] + \frac{n(n-1)(m+1)}{2}$$

From a numerical evaluation, it has been shown that $\xi_{IDR}$ is the smallest among the four bounds compared for general n, m and k. The numerical ordering of these bounds is shown below:

$$\xi_{IDR} \leq \xi_{ID} \leq \xi_{OD} \leq \xi_{DD}$$

for n, k and $m \geq 4$.

---

* This bound was not originally stated correctly by Kohavi and Lavallee.

## REFERENCES


1. Clifford, A. H. and G. B. Preston, The Algebraic Theory of Semigroups, v 1, American Mathematical Society, Providence. Rhode Island (1961) 224.

2. Hennie, F. C., "Fault Detecting Experiments for Sequential Circuits", Proceedings of the Fifth Annual Switching Theory and Logical Design Symposium, v S-164 (1964) 95-110.

3. Hennie, F. C., Finite-State Models for Logical Machines, John Wiley and Sons, Inc.; New York, New York (1968) 466.

4. Kohavi, Z. and P. Lavallee, "Design of Sequential Machines with Fault Detection Capabilities", IEEE Transactions on Electronic Computers, v EC-16, 4(August 1967) 473-484.

5. Meyer, J. F., "Memory Failure in Sequential Machines", Workshop on the Organization of Reliable Automata, Pacific Palisades, California (February 1966).

6. Urbano, R. H., "Reliability, Redundancy, Capacity, and Universality in Polyfunctional Nets", Workshop on the Organization of Reliable Automata, Pacific Palisades, California (1966) 1-21.