General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

RE-ORDER NO. 70-456

Final Report
Project 02602
Contract 952492

THEORY AND DESIGN OF RELIABLE SPACECRAFT DATA SYSTEMS

213 · 354 · 5090

Project Director
John F. Meyer

September 1970

Prepared for

Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, California

FACILITY FORM 602
N71-26361 (ACCESSION NUMBER) (THRU)
393 (PAGES) 03 (CODE)
CR 118664 (NASA CR OR TMX OR AD NUMBER) 07 (CATEGORY)

RECEIVED
JUN 1971
NASA STI FACILITY
INPUT BRANCH

DEPARTMENT OF ELECTRICAL ENGINEERING
SYSTEMS ENGINEERING LABORATORY
THE UNIVERSITY OF MICHIGAN, ANN ARBOR

RE-ORDER NO. 70-456

Final Report
Project 02602
Contract 952492

THEORY AND DESIGN OF RELIABLE SPACECRAFT DATA SYSTEMS

213 · 354 · 5090

Project Director
John F. Meyer

September 1970

Prepared for

Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, California

N71-26361
(ACCESSION NUMBER)
393
(PAGES)
CR 118664
(NASA CR OR TMX OR AD NUMBER)
(THRU)
03
(CODE)
07
(CATEGORY)
FACILITY FORM 602

RECEIVED
JUN 1971
NASA STI FACILITY
INPUT BRANCH

DEPARTMENT OF ELECTRICAL ENGINEERING
SYSTEMS ENGINEERING LABORATORY
THE UNIVERSITY OF MICHIGAN, ANN ARBOR

1953366

R 0 70-456

THE UNIVERSITY OF MICHIGAN

SYSTEMS ENGINEERING LABORATORY

Department of Electrical Engineering
College of Engineering

Final Report
The University of Michigan Project 02602
Jet Propulsion Laboratory Contract 952492

THEORY AND DESIGN OF RELIABLE SPACECRAFT DATA SYSTEMS

Project Director
John F. Meyer

Prepared for

Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, California

Copy No. 4

TABLE OF CONTENTS

		Page
I.	Objectives	1
II.	Personnel	3
III.	Summary of Results	4
IV.	Technical Report	28
	1. Permanent Memory Faults	29
	Fault Masking	41
	Stable Faults	54
	Stuck-at Faults	73
	Fault-Tolerant State-Assigned Machines	87
	Redundancy of t-Tolerant Realizations	105
	Initial State Fault Masking	117
	2. Faults in Combinational Networks	141
	Mathematical Formulation of Fault Concepts	141
	Fault Masking and Detection in Two Node Systems	155
	Optimization in Two Node Systems	178
	Masking and Detection Analysis in Any System	216
	Single Fault Analysis Algorithm	237
	Fault Location	251
	Location of Masked Faults	271
	3. Fault Diagnosis in Sequential Machines and Networks	286
	Augmentation of Test Outputs	300
	Augmentation of Control Inputs	308
	Diagnosable Machine Realization	320
	State Set Convergence Problem	326
	Acyclic Testing Graph Realization	343
	Sequence Generators and Linear Machines	349
	Network Diagnosis	366
V.	References	389

I. OBJECTIVES

The long range objective of this project, as described in the Statement of Work (Article I, JPL Contract No. 952492) was 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 was concerned with the following problems:

(A) Design and analysis of redundant combinational and sequential networks. This includes 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 includes 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 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 was Professor John F. Meyer, Department of Electrical Engineering and Department of Computer and Communication Sciences, the University of Michigan. Two Research Assistants; Mr. F. Gail Gray and Mr. Koumin (Ken) Yeh worked on the project throughout its duration and contributed significantly to the results obtained. A third assistant, Mr. John R. Kinkel, worked on the project during the first five quarters.

III. SUMMARY OF RESULTS

During the first quarter, a working bibliography was prepared emphasizing the following three areas:

1) Redundancy Techniques

2) Fault Diagnosis

3) Reliability Analysis

Reference to QPR 1 should be made for specific details of this bibliography as well as for a survey of selected works in each area.

At the end of the first quarter, investigations into area 3 were discontinued by mutual agreement between the investigators and the Jet Propulsion Laboratory. During the next five quarters, research was continued with regard to areas 1) and 2) and has been concerned primarily with the following three problems:

1) Permanent memory faults in sequential machines and networks

2) Faults in combinational networks, and

3) Fault diagnosis in machines and networks.

The results of each of these investigations is summarized in the paragraphs that follow, and related open problems are discussed in detail. A 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 has been 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 motivated the study was whether certain types of finite-state behavior are inherently less susceptible to memory faults than others. This latter question was to be fully investigated during the second half (last year and one half) of the original contract period and therefore remains to be answered.

Prior to this investigation, the study of errors in sequential machines had been primarily concerned with temporary input errors [10], [19], [29] and temporary state-transition errors [6] [12]. Our study, on the other hand, is concerned with permanent faults that permanently alter the system structure. In particular, we have chosen to restrict our attention to permanent faults that occur in the memory portion of the machine. This restriction is motivated by the fact that it is memory which distinguishes nontrivial sequential machines from purely combinational systems. The latter have been investigated rather thoroughly with regard to error susceptibility but few of the results apply when memory (with feedback) is introduced. 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. This permits the definition of a machine-theoretic model wherein the result of a permanent memory fault is formulated in terms of a sequential machine M that represents the fault-free sequential switching system and a function μ on the states of M that represents the fault. The result of the fault is then represented by a second machine M^{μ} appropriately determined by M and μ.

Summarizing the research effort that has been concerned with this general model, it was shown, first of all, that a succession of such memory faults can be regarded as a single fault which is simply the (functional) composition of all the faults in the succession. The fundamental problem of relating faulty structure to desired behavior was then formulated in terms of three different types of fault masking:

i) Equivalence masking (e-masking)

ii) Inclusion masking (i-masking)

iii) Reset masking relative to some state subset R (R-masking)

Briefly, a fault μ is e-masked if M^{μ} is behaviorally equivalent to M, μ is i-masked if we require only that M^{μ} do everything that M was able to do, and μ is R-masked if each state r in R is equivalent to the faulty state $\mu(r)$ in the faulty machine M^{μ}. These notions were first compared and then investigated relative to necessary and sufficient conditions for a fault to be e-masked, i-masked, or R-masked. In particular, meaningful sufficient conditions for R-masking have been obtained for (memory) faults in general and, when restricted

to special classes of faults, find application to the design of fault tolerant sequential networks.

Before considering design questions, however, two special classes of faults, namely

i) Stable faults, and

ii) Stuck-at faults

were investigated relative to the operation of compostion (or when interpreted, "combination") and to a natural ordering of such faults which has the interpretation of one fault "dominating" another. A number of interesting results were obtained in this regard and, in turn, applied to simplifying masking conditions for these special classes. In particular, it has been established that a stable fault μ is Q-masked (R-masked with R=Q, the set of all states of the machine) if and only if, whenever two states go to the same faulty state (under μ) these states have the same behavior in the fault-free machine. As a consequence of this characterization, whenever two stable, Q-masked faults commute, then their combination must also be Q-masked. This poses a severe design constraint on Q-masking as verified by later results.

Stuck-at faults are defined for state-assigned machines and are a proper subclass of stable faults. The set of all stuck-at faults of a given machine is closed under composition (i.e., is a semigroup) and a number of results have been obtained that relate the semigroup structure to the structure induced by the natural ordering mentioned

earlier. In particular, it has been shown that the statements "γ can follow μ", "meet of μ and γ exists", and "γ commutes with μ" are all equivalent in case μ and γ are stuck-at faults.

With the above properties established, an investigation of fault-tolerant machine design was initiated where the design problem can be generally stated as follows: given some sequential-machine-realizable behavior B specified, say, by a reduced machine M', design a state-assigned machine M that realizes M' and relative to some specified set of faults F, μ is R-masked (e-masked, i-masked), for all μ in F. This problem was considered first for R-masking with R=Q and in this case a surprising negative result has been established, namely, if a machine M is an n-dimensional realization of a machine M' with at least two nonequivalent states, then it is impossible to Q-mask all single stuck-at zero faults (or alternatively, all single stuck-at one faults).

By relaxing the R-masking constraint so that R is a proper subset of Q, synthesis becomes possible even when R is "complete" in the sense that, for each state in Q, there is an equivalent state in R. The synthesis algorithm produces, for a given machine M and given fault tolerance t, a realization that R-masks (with R complete) all stuck-at faults of degree less than or equal to t. By establishing bounds on machine dimensions that are necessary and sufficient for such realizations it has been shown that less redundant memory is

required than for conventional "replicate-and-vote" realizations. Moreover, as the number of states of the realized machine increases, the redundancy required for a given fault tolerance t approaches a theoretical lower bound. This is in contrast with replication schemes where redundancy (e.g., 3 in the case of triplication) does not vary with machine size. It should be borne in mind, however, that the above comparison is made with regard to memory faults only and does not imply that similar results could be obtained for arbitrary faults.

The synthesis of fault tolerant machines was also considered for the special case where only the behavior of some initial or "reset" state needs to be preserved. Although this investigation was well underway when the project was terminated, it is far from completed. Much of the effort to date has been concerned with "local" properties of the states of an assigned machine and, in particular, constraints that must be placed on the (Hamming) distance between certain state pairs if a set of faults is to be $\{q_0\}$-masked. Based on these properties, some heuristic synthesis methods have been experimented with. However, good synthesis algorithms that result in lower redundancy requirements (as compared to complete R-masking) have yet to be discovered.

As for other problems we had hoped to explore but were unable to, it is presently an open question, for example, as to the "combinational network complexity" of sequential networks that correspond to

machines designed for minimum memory redundancy. This same work must be done on the "state assignment problem" for this class of fault-tolerant machines. There is also the possibility that certain types of behavior will lend themselves more easily to economical network realizations and it would be useful to identify such classes of behavior. This is related, in turn, to the more general problem of identifying behavior classes for which a minimum amount of redundant memory is required for a given fault tolerance, minimum in the sense that no other behavior requiring the same number of nonequivalent states can be realized with fewer memory elements.

Faults in Combinational Networks

The purpose of this investigation is to formalize the concepts of fault-masking, detection, location, and diagnosis as applied to combinational networks in a way that will allow efficent analysis and economical synthesis of redundant fault-tolerant switching networks. It is expected that fundamental relationships between these concepts and basic limitations on their applicability will emerge from this study.

The model used employs a structured digraph with k nodes to represent the k functional blocks of a system and ℓ lines to represent the flow of information through the system. A fault is taken to be k-tuple of functions depicting the individual activity of each of the

functional blocks. Associated with each fault f is a mapping $\alpha(f)$ that describes the system behavior when fault f exists in the system.

A fault f is classified according to the implications that arise as a result of observing the system behavior $\alpha(f)$. The fault f is masked if $\alpha(f)$ is the fault-free behavior of the system and detectable otherwise. It is locatable if the set of faulty nodes can be deduced when the system behavior $\alpha(f)$ is observed, and is A-locatable (locatable to within module A) if when the system behavior $\alpha(f)$ is observed we know that the set of faulty nodes is a subset of A. Various other classifications are also possible.

Primary effort has been directed toward networks in which only single faults are allowed. The basic approach has been to investigate the properties of a two-node series system (in which the outputs of node 1 are the inputs of node 2) and to use these results to determine the properties of three-node series parallel networks which are shown to be univerally applicable in the analysis of a system with any number of nodes. The three-node network consists of an input node (node 1) and an output node (node 3) through which all information flows together with an intermediate node (node 2). In general, some information flows through node 2 and some flows around it. Completely general networks can always be reduced to this form for study.

Within the framework of this model, the concepts of a single masked fault and a single detectable fault are easily characterized. Using these characterizations, one can enumerate the number of single masked faults and the number of single detectable faults in an arbitrary combinational network without computing any faulty behaviors. In the two-node system, the number of single masked faults at node 1 is determined by examining the equivalence relations induced on the input space and intermediate space by the fault-free behavior $\alpha(b)$ and the fault-free node 2 mapping, b_2, respectively. The number of single masked faults at node 2 can be determined by examining the range of the fault-free node 1 mapping b_1. In the universal three-node system, the number of single faults at node 2 that are masked is obtained essentially by combining these two results for the two-node system. These theorems have led to an efficient algorithm for generating a list of all single masked faults in any combinational network. Each of these enumerations is done by examining only the fault-free system structure. At no time is recourse to any faulty structure or behavior necessary.

Important necessary and sufficient conditions for masking and for detecting all single faults at a node have also been discovered. In the two-node system, all single faults at node 1 are masked if and only if b_2 is a constant function; whereas all single faults at node 2 are masked if and only if the output space is degenerate (has only a

single element). It has also been found that in the two-node system all single faults at node 1 are detectable if and only if the inverse image under mapping b_2 of each element in the range of $\alpha(f)$ has cardinality 1. Hence, if b_2 is a 1-1 function, then all single faults at node 1 are detectable. On the other hand, if all faults at node 1 are detectable, then b_2 restricted to the range of b_1 must be a 1-1 function. These results indicate that, in the limit, the output stages of a system are more restrictive to fault masking and more amenable to fault detection than the input stages.

Since 100 percent masking can only be achieved in networks realizing constant functions, it is natural to ask how close to 100 percent one can get with other types of functions. The answer to this question is highly dependent upon the function, i.e. it is possible to get very nearly 100 percent fault masking with some functions but no better than 1 or 2 percent with others when optimizing over the set of all two-node systems realizing the desired function. This very important result partially answers the more fundamental question of whether some functions are more adaptable to fault masking than others. Concerning realizability by a two node system with the largest possible fault masking ratio, each function falls into one of three distinct classes. The class of functions in which the largest fault masking ratios are possible is called Class I and includes, among others, all constant functions and all non-onto

1-1 functions. A function g is in Class I if and only if it possesses the property that $p^p/s^q < \prod_{i=1}^{q} d_i^{d_i}$ where p is the cardinality of the domain of g, q is the cardinality of the range of g, s is the cardinality of the codomain of g, and d_i is the cardinality of the ith equivalence class of the equivalence relation induced by g on its domain. Each Class I function has a unique abstract two node system with the maximum fault masking ratio possible for the function (often however many related hardware networks are represented by the same abstract system). Also, the level of redundancy required to achieve this maximum fault masking is relatively low compared to Class II functions.

The requirement for membership in Class II is that a function g have the property that $p^p/s^q > \prod_{i=1}^{q} d_i^{d_i}$. Class II functions only possess a maximum fault masking ratio in the limit as the amount of redundancy is increased without bound. Thus a high level of redundancy is required in order to approach the maximum possible fault masking ratio. In addition, the maximum fault masking ratios for Class II functions are generally less than those for Class I functions with comparable parameters.

Class III functions have the property that $p^p/s^q = \prod_{i=1}^{q} d_i^{d_i}$. These functions can be realized with a wide variety of abstract networks at all levels of redundancy, all of which possess the maximum possible fault masking ratio for the function. However, the fault

masking ratio obtainable is in the same order as that obtainable for Class II functions and is generally less than that obtainable for Class I functions.

The two-node theory was generalized for the universal three-node system by decomposing the node 3 mapping, b_3, and applying the two-node system results to each function in the decomposition. For example, all single faults at node 2 in the three-node system are masked if and only if each component of the output stage de-domposition is a constant function. All single faults at node 2 are detectable if and only if a projection of the node 1 mapping, b_1, is onto the input space of node 2 and for every component, h_x, of the node 3 decomposition, and for every element y in the range of $\alpha(b)$ $b_2 h_x^{-1}(y)$ has cardinality 1. If the output stage itself is being analyzed, the system reduces to the two-node case. Since complete fault masking is then impossible (except in degenerate cases) it is important to know how closely complete fault masking can be approached in general systems. This is currently an open question.

Several interesting and surpising results were also obtained in the area of fault location. Characterizations of locatable faults were obtained for both the two-node and three-node universal systems. These led to the discovery that all single faults at node 1 (node 2) in the two-node system were locatable if and only if the system were degenerate in some sense. Even in the general system form, it was

impossible to locate all single faults at a node. These results indicate that location is too strict a requirement to impose in most situations. Therefore, the more general notion of [B,A]- locatability was introduced. A fault f is [B,A] - locatable if, for every fault f, $\alpha(f') = \alpha(f)$ implies that $B \subseteq K_{f'} \subseteq A$ where $K_{f'}$ is the set of "faulty" nodes associated with fault f'. It was shown that when a fault f is both [B,A]- locatable and [B',A'] - locatable, then it is also $[B \cup B', A \cap A']$ -locatable. Further, if a fault is [B,A] - locatable and $[B,A] \subseteq [B',A']$ then the fault is also [B',A']- locatable. This result indicates that the minimum interval for which a fault is [B,A] -locatable (i.e. the fault is not [B',A']-locatable for any proper subinterval of [B,A]) compactly describes the "locatability", L(f), of the fault. Further investigation revealed that L(f) is an invariant of the equivalence relation induced on the fault set by the mapping α. In other words, the mapping α induces an equivalence relation on the set of faults in such a way that all the members of the same equivalence class have the same locatability, although the locatability of faults in different equivalence classes may be different. This result has important implications because the set of masked faults compose one entire equivalence class. Hence all masked faults have the same locability. Since the 0-fault is masked, the lower 'edge" of the interval must always be the empty set ϕ, and the locatability of masked faults is

described completely by specifying only the upper boundary. For this reason the locatability of the masked faults was described by a single set (i.e., $L(b) = A$). Finally, it was proved that all masked faults in the general system form are $\{2\}$ -locatable if and only if (1) $b_2 b_1$ is onto W x U and (2) $|b_3 b_2^{-1}(y)| = 1$ for every y in the range of b. In this general notation, a fault is locatable if and only if it is $[K_f, K_f]$ -locatable.

The work described in this section leaves several questions unanswered. Although the problem of enumeration of masked faults and detectable faults based only upon the fault-free behavior and upon properties of the fault-free system structure has been solved, the enumeration problem for locatable faults remains open although the solution should not be too difficult.

Another open problem is to find the set of binary systems realizing function g that possess the greatest possible fault masking ratio. This problem has been completely solved only for the special case of two-node systems. A more fundamental but related question is whether some combinational functions are more adaptable for masking faults than others. The two-node results suggest that the answer to this question is in the affirmative. Preliminary investigation leads to the conjecture of results similar to those described above when maximizing over larger sets of realizations.

A completely open problem is concerned with the detection of masked faults. Although this at first seems to be a contradiction in terms, serious reflection suggests that some means of detecting masked faults is essential if optimum performance is to be expected from redundant designs. Redundancy is built into systems in order to insure that the first few component failures will not effect system operation. However, if this type of operation is to be achieved in practice, we must be certain that all components are functioning properly at the time that the system is put into operation (e.g., a rocket is launched). If a high level of redundancy masks large numbers of faults, then we cannot be certain that all components are operating properly just because the system is performing as expected. Several techniques have been suggested in the literature, such as providing special test outputs [9], [15] whose only function is to detect faults that are masked at the normal system outputs. This method of detecting "masked" faults can be incorporated into the present structure by generalizing the notion of a masked fault as follows: A fault f is ρ-masked for a partition ρ of the output space if for every x in the input space $\alpha(f)(x)$ is ρ-equivalent to $\alpha(b)(x)$. A fault is then masked when ρ is the relation of equality on the output space.

Another technique involves the use of special test inputs [15] which are forced to assume a fixed value (either 0 or 1) during normal

operation of the system but may be varied as desired to test certain redundant parts of the system. This method of detecting "masked" faults is incorporated into our model by using the concept of restriction. A fault is A-masked where A is a subset of the input space if for every x in the set $A, \alpha(f)(x) = \alpha(b)(x)$. These two methods may be easily combined to produce $A\rho$ -masked faults. With these generalizations of the masking concept, a whole area of questions remain to be answered, in particular, conjectures similar to every masking or detection theorem in the main body of this report can be made for $A\rho$-masked faults.

Other open questions relate to specialization of the general results to limited classes of faults (e.g., stuck-at faults). It is expected that certain problems (such as detecting all faults) will have less complex solutions when only limited classes of faults are present, however there are no current results to support or deny this conjecture. Also, the immense problem of designing least cost redundant systems is presently unsolved. It does not appear to be any easier than the general minimization problem so frequently attacked with various degrees of partial success, but never solved in a completely general way.

Diagnosis in Sequential Machines and Networks

Historically, much of the research effort concerned with the diagnosis of digital switching systems has been devoted to techniques

for specifying test sequences for existing systems. On the other hand, the system complexity and component manufacturing technology of modern computers are changing at such a rapid rate that diagnosis techniques developed today may become inadequate tomorrow. In view of this fact, it has been gradually recognized that good testing and diagnosis capabilities should be incorporated in the original design of the system. This is the point of view adopted in the following investigation of diagnosis in sequential machines and networks. In particular, the notions of "test output augmentation" and "control input augmentation" are studied from a machine-theoretic point of view in order to determine what additional machine complexity is required if the corresponding switching system is to have the desired diagnostic capabilities. Properties of networks themselves are also studied in this regard beginning with a concept of "node sensitizing" in a combinational network. Further studies in these areas are expected to bring about better diagnostic techniques and a set of diagnosable design criteria. The following summarizes the specific problems considered and solutions obtained during this period of investigation.

Sequential machines were classified in this study according to machine-theoretic properties pertinent to the design of fault detecting experiments. Since the existence of a distinguishing sequence and its length determine the design complexity and length of test sequence

respectively[13], they were chosen as the basis of classification. A machine is diagnosable if it has a distinguishing sequence and is definitely diagnosable if, for some k, every input sequence of length k is a distinguishing sequence [16]. Homable machines and definitely homable machines were similarly defined. It was found that definitely diagnosable machines can be characterized as convergence-free, definitely homable machines. A general upper-bound for the length of a distinguishing sequence in an n-state diagnosable machine is $n^n(n-1)$ [7], but this upper-bound for a definitely diagnosable machine or a diagnosable definitely homable machine is $\binom{n}{2}$. Thus machine classes with short distinguishing sequences can be identified easily from the hierarchy obtained. Further classification, taking into account synchronizing sequences and other properties, is also possible.

Definite diagnosability was shown to be an unnecessarily restrictive condition for the existence of short fault detecting experiments. Usually all that is required is a short distinguishing sequence and it was shown that if a single-input submachine of an n-state machine is reduced then it has a distinguishing sequence of length no greater than n-1. Thus the problem of designing a diagnosable machine is reduced to that of designing a reduced single-input submachine.

The next problem considered was the use of test output augmentation to obtain a particular type of distinguishing sequence. This problem can be stated briefly as follows: Given a machine $M = (I, O, Q, \delta, \omega)$, construct a machine $M' = (I, OxO', Q, \delta', \omega')$ such that M' "contains" M and M' is diagnosable with a repeated symbol distinguishing sequence. A method of constructing such a machine by augmenting the output set was developed. Machines so constructed were seen to have a smaller upper bound on the length of fault detecting experiment when compared with those obtained by other methods [16].

Similarly, using control input augmentation, the design problem can be stated as: Given a machine M, construct $M' = (I \times I', O, Q, \delta', \omega')$ so that M' "contains" M and M' is diagnosable. A machine is said to be optimally diagnosable if it has a distinguishing sequence of the shortest possible length. It was shown that for any given integers n and p, there is an optimally diagnosable single-input machine with n states and p outputs. Thus using input augmentation any n-state, p-output machine can be made optimally diagnosable by appending a single-input machine with this property. A reset input can also be added so that the over-all length of the fault detecting sequence is reduced.

The problems considered above were generalized to that of "machine realization". In general, the problem is to design a

diagnosable machine which realizes a given machine. An important result of this generalization is the consideration of a different set of optimization criteria which is independent of physical realizations. This also resulted in the consideration of state splitting technique which reduces the size of the realizing output set. Theorems were developed which give the size of the realizing output set and the number of additional states required to break up a given set of state convergences. The problem of eliminating cycles in the testing graph of a single input machine was then considered. Finally, the general problem of designing a diagnosable machine with a bounded length distinguishing sequence was investigated.

Since single-input submachines share many common properties with the whole machine, it is natural to study independently the class of single-input machines which are also known as sequence generators. The length of the minimal distinguishing sequence in a reduced sequence generator M can be determined by the least integer i such that $\Pi_i = \Pi_{i+1}$ where Π_i is the i-equivalence partition on the state set of M. Thus optimally diagnosable sequence generators can be characterized in terms of such partitions. The class of reduced linear machines was found to have some common diagnostic properties with the class of reduced sequence generators. First, if a linear machine has a distinguishing sequence of length i, then

every input sequence of length i is a distinguishing sequence. Thus, the length of a minimal distinguishing sequence can be determined by the least integer i such that $II_i = II_{i+1}$. Equivalently, the length of a minimal distinguishing sequence can be determined from the least i such that the rank of an associated matrix is equal to the dimension of the state space. Thus a reduce 2^n-state linear machine is definitely diagnosable of order no greater than n.

Some properties of linear machines relating to machine diagnosis have also been investigated. First, it has been shown that the i-equivalence relation on a linear machine partitions its state set into cosets when the state space is considered as an additive Abelian group. Based on this observation, it is shown then that if π_i is the i-equivalence partition of the state set of a linear machine M over a Galois field of p^m elements, then $|\pi_{i+1}| = p^{k_i} |\pi_i|$ with $0 \leq k_i < \ell m$, where ℓ is the dimension of the ouput space. If n is the dimension of the state space and ℓ divides n, then M is optimally diagnosable if and only if $|\pi_{i+1}| = q^{\ell} |\pi_i|$ for all $0 \leq i \leq \frac{n}{\ell}$, where $q = p^m$. The optimal diagnosable characterization is then generalized to some non-linear sequential machines.

The problem of diagnosis was also studied from the network structure viewpoint. The motivation of this effort is to identify the classes of networks that can be easily tested and diagnosed. A network graph model appropriate for the analysis of stuck-at faults

in combinational networks was first introduced. Based on this graphical model, a notion of node sensitizing was then introduced and in terms of this, the concept of path sensitizing [1] can be defined. With these definitions, precise statements of some known results are possible. It was also shown that if a node can be both 1-sensitized and 0-sensitized and the sensitized multipath is simply connected, then there is a sensitized path passing through this node. Moreover, it was established that if no constant node exists in a combinational network graph, then every stuck-at fault is T-detectable for some subset of nodes T in the network.

The concept of path sensitizing was then extended to subgraphs where every path is a sensitized path. The notion of "node detection" under some network input x has been found to be a partial ordering of the node set of the sensitized subgraph. This ordering has been applied to obtain properties for the notion of fault detection by a node. Using these results and the concept of multiple fault detection, it was then shown that the set of stuck-at faults in a sensitized path detected by a node under some network input is also multiply detected by the same node under the same input. If the network graph is a tree, then in a rooted sensitized subgraph, any combination of "singly" detected stuck-at faults is also multiply detected by the "root" node under the sensitizing input.

The problems considered above and solutions obtained were directed toward the originally stated objectives of i) developing theory and techniques for determining test-point allocation in order to reduce the time required for diagnosis and ii) investigating questions relating to system organizations that facilitate fault diagnosis. Based on these results, it is felt that the more general problem of fault diagnosis in redundant systems can be readily investigated.

The presence of fault masking in a network gives rise to a contradictory requirement when fault detection is considered, since by definition a fault is detectable if and only if it is not masked. The problems of specifying where to put test points, how to compromise between the testing length and number of test points, and how addition of some control inputs might improve diagnosis resolution remain essentially unsolved. It is believed that the diagnosable machine realization concept introduced in our study can be utilized to answer some of these questions.

Another problem deserving of further investigation is the development of "efficient" algorithms which select a minimum or a near-minimum set of test points so that a given set of faults in a network are detected. Network controllability should also be investigated so that some diagnosable network structures can be identified. These desirable properties include homability of sequential networks

under failures and complete path sensitizing in combinational networks. A set of design criteria should be sought which would achieve these desired diagnostic properties with minimum or near minimum redundancy.

IV. TECHNICAL REPORT

The following is a technical report on the research activities related to this project. (A working bibliography and a survey of recent work in the general reliability area appears in QPR 1.) Investigations were concerned with the three problem areas summarized in Section III: 1) permanent memory faults in sequential machines and networks, 2) faults in combinational networks, and 3) fault diagnosis.

This section includes proofs of all theorems as well as a cohesive discussion of concepts, results, motivation, and interpretation. Examples are also included that illustrate key points. With the exception of the material in QPR 1, this report is self contained.

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 [17] 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 μ on the states of M that represents the fault. The result of the fault is then represented by a second machine M^{μ} appropriately determined from M and μ. Given this representation, it is possible to investigate how the behavior of M^{μ} relates to the behavior of M. Of particular interest is when a fault is tolerated (masked) in the sense that the resulting behavior relates in some specified way to the original behavior. Various types of fault masking are investigated from this point of view including a notion of R-masking, where R is a subset of the state set Q. A special class of stable faults is also considered, the latter being of interest since it contains all faults that represent combinations of stuck-at faults in the individual memory cells of a physical system.

One of the primary goals of the above investigation is its application to the synthesis of fault tolerant sequential switching networks. Several important results have been obtained in this regard and it appears that further investigation of the subject is warrented. Another fundamental question is whether certain types of sequential behavior

are inherently less susceptible to memory faults if the measure of susceptibility is the minimum amount of redundant memory required to mask a specified class of faults. This question has yet to be investigated and remains an important open question.

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 1.1

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

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

A Moore sequential machine is as above except that

v') ω 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 Λ), 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

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

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 1.2

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 as follows:

Transition fcn.	Output fcn. (Mealy)	Output fcn. (Moore)
$\bar{\delta} : Q \times I^* \to Q$	$\bar{\omega} : Q \times I^\dagger \to O$	$\bar{\omega} : Q \times I^* \to O$

where

	Transition fcn.	Output fcn. (Mealy)	Output fcn. (Moore)
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) $x \in I^k$, $a \in I$, $(k>1)$	$\bar{\delta}(q, xa) = \delta(\bar{\delta}(q, x), a)$	$\bar{\omega}(q, xa) = \omega(\bar{\delta}(q, x), a)$	$\bar{\omega}(q, xa) = \omega(\bar{\delta}(q, xa))$

(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 1.3

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 β_q defined as follows:

	Mealy	Moore
	$\beta_q : I^\dagger \to O^\dagger$	$\beta_q : I^* \to O^\dagger$
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) $x \in I^k$, $a \in I$, $(k > 1)$	$\beta_q(xa) = \beta_q(x)\bar{\omega}(q, xa)$	$\beta_q(xa) = \beta_q(x)\bar{\omega}(q, xa)$

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 1.4

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

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

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 1.5

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 $x \in I^\dagger$ (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 1.6

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 1.7

If $M, N \in \mathcal{M}(I, O)$ then M includes N $(M \geq N)$ 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 if 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 δ', and output function ω' 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 interprete $\delta(q, a)$ to be the "next" state given that the "present" state is q and the "present" input is a. To distinguish the functions of memory and decision let $\delta = \mu \bullet \lambda$

(the functional composition of λ and μ, first applying λ) where $\lambda(q, a)$ is the memory input and represents a purely combinational process, and μ is the memory function representing the storage of $\lambda(q, a)$. In case the memory operates properly, μ 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 μ representing faulty memory operation is no longer the identity function and the transition function of the faulty machine is given by

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

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

$$\lambda' = \lambda$$

and hence

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

The above observations can be formalized as follows:

Definition 1.8

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

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

where

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

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 though, under the intended interpreation, 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^{μ} of a fault μ, Q^{μ} is taken to be the range of μ since, under the interpretation, these are the only states accessible from the memory input. The definition of the faulty transition function δ^{μ} follows directly from (1.2)-(1.4). Since the fault occurs in memory, the output function ω^{μ} is essentially the same as ω. This, then, is 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 μ 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, μ is a fault of M and γ is a fault of M^{μ} then

$$(M^{\mu})^{\gamma} = M^{\gamma \bullet \mu} \qquad (1.5)$$

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

Given μ and γ as above, one can also regard γ 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 μ to R. Then

Definition 1.9

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

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 μ (i.e. $\mathcal{R}(\mu) = \mu(Q)$ if $\mu: Q \to Q$) then

Theorem 1.1

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

Proof

By definition γ can follow μ if and only if

$$(*)\quad \mathcal{R}(\gamma|Q^{\mu}) \subseteq Q^{\mu}.$$

Suppose (*) holds and $q \in \mathcal{R}(\gamma \bullet \mu)$. Then there is a state $r \in Q$ such that $q = \gamma(\mu(r))$ and so $q \in \mathcal{R}(\gamma|Q^{\mu})$. Hence $q \in Q^{\mu} = \mathcal{R}(\mu)$ thereby proving that $\mathcal{R}(\gamma \bullet \mu) \subseteq \mathcal{R}(\mu)$. Conversely, suppose that $\mathcal{R}(\gamma \bullet \mu) \subseteq \mathcal{R}(\mu)$ and let $q \in \mathcal{R}(\gamma|Q^{\mu})$. Then for some r, $q = \gamma(\mu(r))$ or, equivalently, $q \in \mathcal{R}(\gamma \bullet \mu)$ which implies $q \in \mathcal{R}(\mu) = Q^{\mu}$, thereby proving (*).

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 ($n \geq 2$) then $(\mu_1, \mu_2, \ldots, \mu_n)$ is a succession of faults of M if

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

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 ($n \geq 2$) then $(\mu_1, \mu_2, \ldots, \mu_n)$ is a succession of faults of M if and only if

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

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

Proof

By induction on n.

Basis: If n=2, Theorem 1.2 reduces to Theorem 1.1.

Induction step: Suppose the theorem holds for n=k and let n=k+1.

Then, by the induction hypothesis it suffices to show that μ_{k+1} can follow $\mu_k \cdot \mu_{k-1} \cdots \mu_1$ if and only if

$$\mathcal{R}(\mu_{k+1} \cdot \mu_k \cdots \mu_1) \subseteq \mathcal{R}(\mu_k \cdot \mu_{k-1} \cdots \mu_1).$$

Letting $\gamma = \mu_{k+1}$ and $\mu = \mu_k \cdot \mu_{k-1} \cdots \mu_1$, the latter holds by Theorem 1.1, thereby proving Theorem 1.2.

As a corollary of Theorem 1.2, if equation (1.5) is extended to a succession of n faults, we have:

Corollary 1.2.1

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 μ of M where μ 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 μ if M^{μ} no longer exhibits the desired behavior. On the other hand, if the desired behavior is preserved under μ then, adopting a term used quite frequently in this context, the fault μ 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 μ 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 μ 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:*

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 μ_1, μ_2, and μ_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^{μ_1}, M^{μ_2} and M^{μ_3} are given by:

---

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

M^{μ_1}:

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

M^{μ_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^{μ_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 β_q^{μ} denote the behavior of M^{μ} for initial state q then, by inspection of M^{μ}, 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. μ_1 is e-masked.

Regarding M^{μ_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 μ_2 is i-masked but not e-masked.

As for M^{μ_3}, we see that no state of M^{μ_3} is equivalent to any state of M and consequently μ_3 is not i-masked.

By definition, a fault μ is e-masked if the faulty machine M^{μ} 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^{μ} 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^{μ} 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 $\beta^{\mu_1}_{\mu_1(2)} = \beta^{\mu_1}_0 = \beta_0 \neq \beta_2$. Accordingly, if we were to attempt to reset the faulty system (represented by M^{μ_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 μ is R-masked if

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

(where β^{μ}_q, as earlier, is the behavior of M^{μ} for initial state q).

Thus if μ is R-masked then M^{μ} 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^{μ} for initial state $\mu(r)$. Referring to Example 1.1 one can easily verify that μ_1 is $\{0, 1, 4, 5\}$-masked and μ_2 is

$\{0, 1, 2, 3\}$—masked. μ_3 is not R-masked if $R \neq \phi$ (ϕ being the empty set; note that every fault is ϕ-masked).

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

Theorem 1.3

If M is a machine with states Q and a fault μ is Q-masked then μ is e-masked.

Proof:

If μ is Q-masked then $\beta_q = \beta^{\mu}_{\mu(q)}$, for all $q \in Q$, and so $B_M \subseteq B_{M^\mu}$. Since $\mathcal{R}(\mu) = Q^\mu$ we have $B_M = B_{M^\mu}$ and therefore μ is e-masked.

That the converse of Theorem 1.3 fails to hold is illustrated by fault μ_1 of Example 1.1. Indeed one can construct a machine M along with a fault μ such that μ is e-masked and yet $R \neq \phi$ implies μ 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

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

Then

Theorem 1.4

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

Proof

If μ is R-masked then

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

But R is complete and so

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

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

To illustrate Theorem 1.4, consider the fault μ_2 of Example 1.1 along with the subset $R = \{0, 1, 2, 3\}$. Then R is complete and, as noted earlier, μ_2 is R-masked. Hence μ_2 must be i-masked and we observed in Example 1.1 that this was the case. The converse of Theorem 1.4 does not hold, that is, a fault μ can be i-masked and yet there is no complete subset R such that μ is R-masked.

If we now look ahead 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 $\{\mu_1, \mu_2, \ldots, \mu_k\}$, μ_i is □-masked (i=1, 2, . . . , k) where □ denotes one of the specific types of masking just discussed. Solutions to this problem require a greater understanding of how a fault μ must relate to a machine M in order that it be □-masked. In particular, it would be convenient to relate μ directly to M without having to completely determine the nature of the faulty machine M^{μ}. The following results are so motivated.

We begin by establishing two lemmas that are used to support later arguments.

Lemma 1.1

If μ is a fault of $M = (I, Q, O, \delta, \omega)$, $R \subseteq Q$ and $\delta(\mu(R) \times I) \subseteq R$ then

$$\bar{\delta}^{\mu}(q,x) \in \mu(R), \text{ for all } q \in \mu(R), x \in I^*.$$

($\bar{\delta}^{\mu}$ is the extended transition function of M^{μ}; see Def. 1.2.)

Proof

If $\delta(\mu(R) \times I) \subseteq R$ then $\mu \cdot \delta(\mu(R) \times I) \subseteq \mu(R)$ or, equivalently, $\delta^{\mu}(\mu(R) \times I) \subseteq \mu(R)$ where δ^{μ} is the transition function of M^{μ}. As $\mu(R)$ is closed under δ^{μ}, M^{μ} restricted to the state set $\mu(R)$ is a submachine of M^{μ}. This implies that

$$\bar{\delta}^{\mu}(q,x) \in \mu(R), \text{ for all } q \in \mu(R), x \in I^*$$

thereby proving the lemma.

If, in addition to $\delta(\mu(R) \times I) \subseteq R$, each state $r \in R$ fails to an equivalent state (when regarded as a state of M), we have:

Lemma 1.2

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

i) $\delta(\mu(R) \times I) \subseteq R$, and

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

then

$$\bar{\delta}(r,x) \equiv \bar{\delta}^{\mu}(\mu(r),x), \text{ for all } r \in R, x \in I^*.$$

Proof

The proof is by induction on the length lg(x) of an input sequence x.

Basis: If $lg(x) = 0$, $x = \Lambda$ and if $r \in R$ we have $\bar{\delta}(r, \Lambda) = r \equiv \mu(r) = \bar{\delta}^{\mu}(\mu(r), \Lambda)$.

Induction step: Suppose true for all $y \in I^k$ and let $r \in R$, $x \in I^{k+1}$.

Then $x = ya$ for some $y \in I^k$, $a \in I$ and

$$\bar{\delta}(r,x) = \bar{\delta}(r,ya) = \delta(\bar{\delta}(r,y),a).$$

By the induction hypothesis,

$$\bar{\delta}(r,y) \equiv \bar{\delta}^{\mu}(\mu(r),y)$$

and by the substitution property of $\equiv$,

$$\delta(\bar{\delta}(r,y),a) \equiv \delta(\bar{\delta}^{\mu}(\mu(r),y),a). \qquad (1.6)$$

If we let $s = \bar{\delta}^{\mu}(\mu(r),y)$, then by condition i) and Lemma 1.1, $s \in \mu(R)$.

Applying condition i) once more, $\delta(s,a) \in R$ and by condition ii) we have:

$$\delta(s,a) \equiv \mu \cdot \delta(s,a) = \delta^{\mu}(s,a). \qquad (1.7)$$

Replacing s by $\bar{\delta}^{\mu}(\mu(r),y)$ and combining equivalences (1.6) and (1.7),

$$\delta(\bar{\delta}(r,y),a) \equiv \delta^{\mu}(\bar{\delta}^{\mu}(\mu(r),y),a).$$

As $x = ya$, this implies that

$$\bar{\delta}(r,x) \equiv \bar{\delta}^{\mu}(\mu(r),x)$$

thereby completing the induction step and proving the lemma.

Applying Lemma 1.2, it follows that conditions i) and ii) are sufficient for μ to be R-masked, that is:

Theorem 1.5

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

i) $\delta(\mu(R)\times I) \subseteq R$, and

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

then μ is R-masked.

Proof

Suppose conditions i) and ii) hold and $r \in R$. Then, by Lemma 1.2,

$$\overline{\delta}(r,x) \equiv \overline{\delta}^{\mu}(\mu(r),x), \text{ for all } x \in I^*$$

which implies

$$\omega(\overline{\delta}(r,x),a) = \omega^{\mu}(\overline{\delta}^{\mu}(\mu(r),x),a), \text{ for all } x \in I^*, a \in I.$$

But, by definition, this says that

$$\beta_r(xa) = \beta^{\mu}_{\mu(r)}(xa), \text{ for all } x \in I^*, a \in I$$

or, equivalentiy

$$\beta_r(y) = \beta^{\mu}_{\mu(r)}(y), \text{ for all } y \in I^{\dagger}.$$

As r is an arbitrary state in R, it follows that μ is R-masked, thereby proving the theorem.

As corollaries to Theorem 1.5 we observe

Corollary 1.5.1

If conditions i) and ii) of Theorem 1.5 hold and R is complete then μ is i-masked.

Corollary 1.5.2

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

Proof

If $R = Q$, condition i) of the theorem is automatically satisfied and thus condition ii) suffices for Q-masking.

Theorem 1.5 gives sufficient conditions for R-masking a fault μ in terms of μ, 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 μ. The best characterization obtained so far is stated in terms of a relation μ_R defined as follows:

Definition 1.14

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

$q \mu_R q'$ if there is some $r \in R$ and some $x \in I^*$ such that

$$\bar{\delta}(r,x) = q \text{ and } \bar{\delta}^{\mu}(\mu(r),x) = q'.$$

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 1.6

If M is a Mealy machine and μ is a fault of M then

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

Proof: Necessity

Suppose μ is R-masked and $q \mu_R q'$. Then there is some state $r \in R$ and some $x \in I^*$ such that

$$\bar{\delta}(r,x) = q \text{ and } \bar{\delta}^{\mu}(\mu(r),x) = q' \qquad (1.8)$$

Since $\beta_r = \beta^{\mu}_{\mu(r)}$, $\beta_r(xa) = \beta^{\mu}_{\mu(r)}(xa)$, for all $a \in I$ and therefore

$$\omega(\bar{\delta}(r,x),a) = \omega^{\mu}(\bar{\delta}^{\mu}(\mu(r),x),a), \text{ for all } a \in I.$$

But $\omega^{\mu} = \omega | Q^{\mu} \times I$ and substituting according to (1.8) we have

$$\omega(q,a) = \omega(q',a), \text{ for all } a \in I,$$

that is, $q \equiv_1 q'$. Thus $\mu_R \subseteq \equiv_1$.

Sufficiency:

Suppose $\mu_R \subseteq \equiv_1$, $r \in R$ and $x \in I^{\dagger}$. Then $x = ya$, for some $y \in I^*$ and $a \in I$, and

$$\bar{\omega}(r,x) = \omega(\bar{\delta}(r,y),a) \qquad (1.9)$$

Since $\bar{\delta}(r,y) \mu_R \bar{\delta}^{\mu}(\mu(r),y)$ and $\mu_R \subseteq \equiv_1$,

$$\omega(\bar{\delta}(r,y),a) = \omega(\bar{\delta}^{\mu}(\mu(r),y),a)$$

$$= \omega^{\mu}(\bar{\delta}^{\mu}(\mu(r),y),a)$$

$$= \bar{\omega}^{\mu}(\mu(r),ya)$$

$$= \bar{\omega}^{\mu}(\mu(r),x)\,. \qquad (1.10)$$

Combining (1.9) and (1.10), $\bar{\omega}(r,x) = \bar{\omega}^{\mu}(\mu(r),x)$ for all $x \in I^{\dagger}$. By the definition of behavior it follows that

$$\beta_r = \beta^{\mu}_{\mu(r)}$$

thereby proving sufficiency.

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

Theorem 1.6'

If M is a Moore machine and μ is a fault of M then

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

Proof

Similar to the proof of Theorem 1.6.

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_0,

then the fault masking of interest is a special case of R-masking where

$$R = \{q_0\}.$$

In this case we will say that μ is q_0-masked (rather than $\{q_0\}$-masked) and write μ_{q_0} instead of $\mu_{\{q_0\}}$ (Definition 1.14). Moreover, the relation μ_R can be described somewhat more simply when $R = \{q_0\}$, that is

$$\mu_{q_0} = \{(\overline{\delta}(q_0,x), \overline{\delta}^{\mu}(\mu(q_0),x)) \mid x \in I^*\}. \qquad (1.11)$$

Using this characterization of μ_{q_0} and applying Theorem 1.6 it follows that:

Theorem 1.7

If M is a Mealy machine and μ is a fault of M then

$$\mu \text{ is } q_0\text{-masked iff } \overline{\delta}(q_0,x) \equiv_1 \overline{\delta}^{\mu}(\mu(q_0),x), \text{ for all } x \in I^*.$$

Proof

If μ is q_0-masked then, by (1.11), $\overline{\delta}(q_0,x)\, \mu_{q_0}\, \overline{\delta}^{\mu}(\mu(q_0),x)$, for all $x \in I^*$. By Theorem 1.6, with $R = \{q_0\}$, it follows that $\overline{\delta}(q_0,x) \equiv_1 \overline{\delta}^{\mu}(\mu(q_0),x)$, for all $x \in I^*$. Conversely, if the latter holds then $\mu_{q_0} \subseteq \equiv_1$ and, by Theorem 1.6, μ is q_0-masked.

The corresponding statement for Moore machines is given by:

Theorem 1.7'

If M is a Moore machine and μ is a fault of M then

$$\mu \text{ is } q_0\text{-masked iff } \bar{\delta}(q_0,x) \equiv_0 \bar{\delta}^{\mu}(\mu(q_0),x), \text{ for all } x \in I^*.$$

The importance of Theorems 1.6 and 1.7 is their application to the design of fault tolerant sequential switching networks. Before discussing such applications, however, we wish to investigate the effects of imposing certain physically motivated constraints on the class of memory faults.

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 a fault itself we have:

Definition 1.15

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

In other words, μ is stable if every state of M^{μ} is stable. Stable faults are of interest since many types of physical memory faults may be represented by a machine fault of this type. In

particular, as is shown in the next subsection, 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.

In mathematical terms, μ is stable if and only if it is an idempotent element of the semigroup of functions on Q. If $\mu: Q \to Q$ is a fault of M, let $\underset{\mu}{\equiv}$ denote the equivalence relation (on Q) induced by μ, that is

$$q \underset{\mu}{\equiv} r \text{ if } \mu(q) = \mu(r). \tag{1.12}$$

then the notion of a stable fault can be alternatively characterized as follows:

Theorem 1.8

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

i) μ is stable

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

iii) $\mu(q) \underset{\mu}{\equiv} q$, for all $q \in Q$.

Proof

i) $\Longrightarrow$ ii): Suppose μ is stable and $r \in \mathcal{R}(\mu)$. Then, for some $q \in Q$, $r = \mu(q)$ which implies

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

ii) $\Longrightarrow$ iii): Suppose $\mu(r) = r$, for all $r \in \mathcal{R}(\mu)$, and $q \in Q$. Then $\mu(q) \in \mathcal{R}(\mu)$ and therefore

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

or, by (1.12),

$$\mu(q) \underset{\mu}{\equiv} q.$$

iii) $\Longrightarrow$ i): If $\mu(q) \underset{\mu}{\equiv} q$, for all $q \in Q$ then $\mu(\mu(q)) = \mu(q)$, for all $q \in Q$, and hence μ is stable.

As we wish to investigate the structure of the set of stable faults of a machine M with states Q, in the discussion that follows, let

$$S(Q) = \{\mu \mid \mu \text{ is a stable fault of } M\}.$$

Our first observation regards the size of S(Q), namely:

Theorem 1.9

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

Proof

Given some nonempty subset $R \subseteq Q$ where $|R| = k$, there are

$$k^{n-k}$$

different stable faults μ such that $\mathcal{R}(\mu) = R$. This is because the k elements of R are fixed under μ (see Theorem 1.8, part ii)), leaving n-k elements, each of which can be assigned one of the k elements of R. Since there are

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

k-element subsets of Q ($1 \leq k \leq n$), the desired result follows.

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

When $|Q| \geq 3$, it is known that S(Q) is <u>not</u> closed under composition, that is, there are stable faults μ and γ such that $\gamma \cdot \mu$ is not stable. Thus, for example, if $Q = \{0,1,2\}$, $\mu = (0,0,2)$ and $\gamma = (0,1,1)$, we have $\gamma \cdot \mu = (0,0,1)$ which is not stable. A necessary and sufficient condition for the composition of stable faults to be stable is given by the following theorem.

<u>Theorem 1.10</u>

If $\mu, \gamma \in S(Q)$ then

$$\gamma \cdot \mu \in S(Q) \text{ iff } q \underset{\gamma}{\equiv} \mu(q), \text{ for all } q \in \mathcal{R}(\gamma \cdot \mu) - \mathcal{R}(\mu).$$

($\underset{\gamma}{\equiv}$ is the equivalence relation induced by γ)

<u>Proof: Necessity</u>

Suppose $\gamma \cdot \mu \in S(Q)$ and $q \in \mathcal{R}(\gamma \cdot \mu) - \mathcal{R}(\mu)$. Then $q \in \mathcal{R}(\gamma \cdot \mu)$ which implies $\gamma(\mu(q)) = q$ (since $\gamma \cdot \mu$ is stable). But $q \in \mathcal{R}(\gamma \cdot \mu)$ also implies $q \in \mathcal{R}(\gamma)$ and so $\gamma(q) = q$. Combining these two equations, $\gamma(\mu(q)) = \gamma(q)$ or, equivalently,

$$\mu(q) \underset{\gamma}{\equiv} q.$$

Sufficiency:

Suppose $\mu(q) \underset{\gamma}{\equiv} q$, for all $q \in \mathcal{R}(\gamma \bullet \mu) - \mathcal{R}(\mu)$. To prove that $\gamma \bullet \mu \in S(Q)$ we will show that $q \in \mathcal{R}(\gamma \bullet \mu)$ implies $\gamma \bullet \mu(q) = q$ (cf. Theorem 8, part ii)). We suppose then that $q \in \mathcal{R}(\gamma \bullet \mu)$ and consider two cases.

Case i): $q \in \mathcal{R}(\mu)$. Then $\mu(q) = q$ which implies $\gamma(\mu(q)) = \gamma(q)$. But $q \in \mathcal{R}(\gamma \bullet \mu)$ implies $q \in \mathcal{R}(\gamma)$ and so $\gamma(q) = q$. Hence $\gamma \bullet \mu(q) = \gamma(\mu(q)) = q$.

Case ii): $q \notin \mathcal{R}(\mu)$. Then $q \in \mathcal{R}(\gamma \bullet \mu) - \mathcal{R}(\mu)$ and, by hypothesis $\mu(q) \underset{\gamma}{\equiv} q$. Thus $\gamma(\mu(q)) = \gamma(q)$ and since $\gamma(q) = q$ (see case i)), we have $\gamma \bullet \mu(q) = \gamma(\mu(q)) = q$.

As a corollary of Theorem 1.10, we note that if one stable fault can follow another (cf. Def. 1.9) then the composite fault is always stable, that is,

Corollary 1.10.1

If $\mu, \gamma \in S(Q)$ and γ can follow μ then $\gamma \bullet \mu \in S(Q)$.

Proof

If γ can follow μ then, by Theorem 1.1, $\mathcal{R}(\gamma \bullet \mu) \subseteq \mathcal{R}(\mu)$. Therefore $\mathcal{R}(\gamma \bullet \mu) - \mathcal{R}(\mu) = \emptyset$ and the condition of Theorem 1.10 is vacuously satisfied.

Generalizing Corollary 1.10.1 it follows that if $(\mu_1, \mu_2, \ldots, \mu_n)$ is a succession of stable faults then $\mu_n \bullet \mu_{n-1} \bullet\bullet\bullet \mu_1$ is stable.

The set S(Q) of stable faults of a machine M can be further investigated in terms of a natural partial ordering* of S(Q) defined as follows [3]:

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

Definition 1.16

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

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.

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) $\underset{\gamma}{\equiv} \subseteq \underset{\mu}{\equiv}$.

Proof: Necessity:

Suppose $\mu \leq \gamma$. Then, by definition, $\gamma \bullet \mu = \mu \bullet \gamma = \mu$ and consequently

$$\mathcal{R}(\mu) = \mathcal{R}(\gamma \bullet \mu) \text{ and } \underset{\mu \bullet \gamma}{\equiv} = \underset{\mu}{\equiv}$$

But, for any pair of functions μ and γ on Q, we have

$$\mathcal{R}(\gamma \bullet \mu) \subseteq \mathcal{R}(\gamma) \text{ and } \underset{\gamma}{\equiv} \subseteq \underset{\mu \bullet \gamma}{\equiv}$$

and therefore

i) $\mathcal{R}(\mu) \subseteq \mathcal{R}(\gamma)$ and ii) $\underset{\gamma}{\equiv} \subseteq \underset{\mu}{\equiv}$.

Sufficiency:

Suppose $\mu, \gamma \in S(Q)$ and condition i) and ii) hold. If $q \in Q$, $\mu(q) \in \mathcal{R}(\mu)$ and, by i), $\mu(q) \in \mathcal{R}(\gamma)$. By part ii) of Theorem 1.8,

$$\gamma(\mu(q)) = \mu(q). \tag{1.13}$$

Also, by part iii) of Theorem 1.8, $\gamma(q) \underset{\gamma}{\equiv} q$ and it follows by condition ii) that $\gamma(q) \underset{\mu}{\equiv} q$, or, equivalently,

$$\mu(\gamma(q)) = \mu(q). \tag{1.14}$$

Combining (1.13) and (1.14) with the fact that q is an arbitrary element of Q, we have

$$\gamma \bullet \mu = \mu \bullet \gamma = \mu,$$

that is, $\mu \leq \gamma$.

The following example illustrates the partial ordering $\leq$ and the characterization given by Theorem 1.11.

Example 1.2

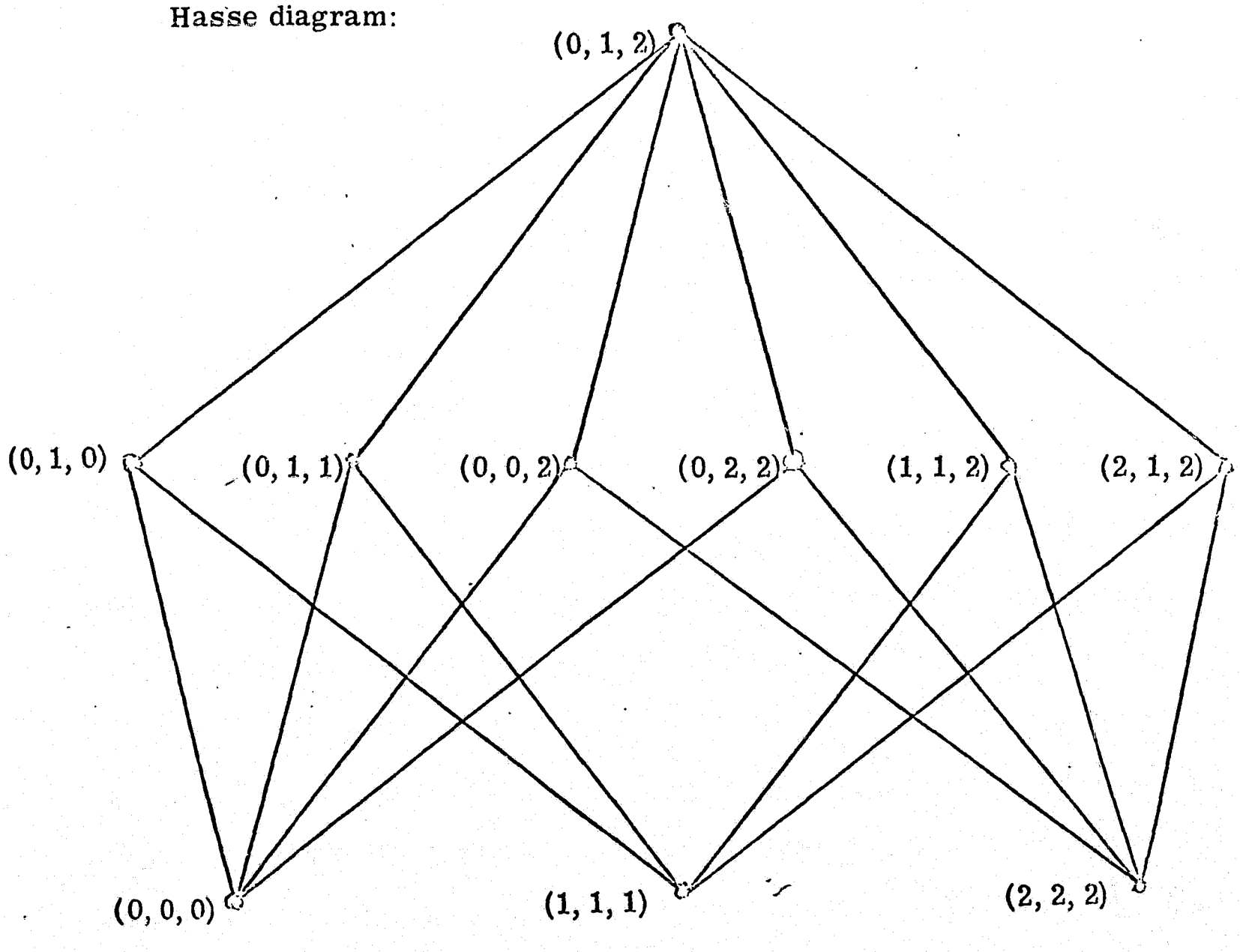
Let $Q = \{0, 1, 2\}$ and denote a stable fault $\mu \in S(Q)$ as the triple

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

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

μ	$\mathcal{Q}(\mu)$	π_μ
(0, 1, 2)	{0, 1, 2}	$\{\bar{0}, \bar{1}, \bar{2}\}$
(0, 1, 0)	{0, 1}	$\{\bar{1}, \bar{02}\}$
(0, 1, 1)	{0, 1}	$\{\bar{0}, \bar{12}\}$
(0, 0, 2)	{0, 2}	$\{\bar{2}, \bar{01}\}$
(0, 2, 2)	{0, 2}	$\{\bar{0}, \bar{12}\}$
(1, 1, 2)	{1, 2}	$\{\bar{2}, \bar{01}\}$
(2, 1, 2)	{1, 2}	$\{\bar{1}, \bar{02}\}$
(0, 0, 0)	{0}	$\{\bar{012}\}$
(1, 1, 1)	{1}	$\{\bar{012}\}$
(2, 2, 2)	{2}	$\{\bar{012}\}$

Accordingly, the partially ordered set (S(Q), $\leq$) has the following Hasse diagram:



Investigating the structure of the partially ordered set $(S(Q), \leq)$ in more detail, one observes first of all that $(S(X), \trianglelefteq)$ is a lattice only in the trivial case when $|Q| = 1$. Moreover, $(S(Q), \trianglelefteq)$ is an upper semilattice only if $|Q| \leq 2$. If $|Q| > 2$, then $S(Q)$ is neither an upper nor a lower semilattice. Nevertheless, the greatest lower bound (meet) and least upper bound (join) do exist for many pairs $\{\mu,\gamma\}$ and the following case is important to our development. To simplify notation, if μ and γ are faults of M we will henceforth let $\gamma\mu$ denote the composition of μ and γ, i.e.

$$\gamma\mu = \gamma \bullet \mu \tag{1.15}$$

and we will let

$$\bar{\mu} = \text{the least (positive) power of } \mu \text{ that is stable.} \tag{1.16}$$

Thus, if $\mu \in S(Q)$ then $\bar{\mu} = \mu$ and if $\mu \notin S(Q)$ then $\bar{\mu} = \mu^n = \mu\mu \ldots \mu$ (n times) where n is such that $\mu^n \in S(Q)$ and $\mu^{n-1} \notin S(Q)$. (Such an integer n exists since some power of every element of a finite semigroup is idempotent [3].) A sufficient condition, under which the meet of $\{\mu,\gamma\}$ (denoted $\mu \wedge \gamma$) exists and is easily expressed, is given by the following theorem.

Theorem 1.12

If $\mu, \gamma \in S(Q)$ and $\overline{\gamma\mu} = \overline{\mu\gamma}$ then

$$\mu \wedge \gamma = \overline{\gamma\mu}.$$

Moreover,

i) $\mathcal{R}(\mu \wedge \gamma) = \mathcal{R}(\mu) \cap \mathcal{R}(\gamma)$

and

ii) $\mu \overline{\overline{\wedge}} \gamma = \overline{\overline{\mu}} + \overline{\overline{\gamma}}$.

(+ denotes join in the lattice of equivalence relations on Q).

Proof.

Let $\lambda = \overline{\gamma\mu} = \overline{\mu\gamma}$ and let m and n be the integers such that $\overline{\gamma\mu} = (\gamma\mu)^m$ and $\overline{\mu\gamma} = (\mu\gamma)^n$. We note first that λ is a lower bound of $\{\mu, \gamma\}$ since

$$\lambda\mu = (\gamma\mu)^m\mu = (\gamma\mu)^{m-1}\gamma\mu^2 = (\gamma\mu)^{m-1}\gamma\mu = (\gamma\mu)^m = \lambda$$

and $\mu\lambda = \mu(\mu\gamma)^n = \mu^2\gamma(\mu\gamma)^{n-1} = \mu\gamma(\mu\gamma)^{n-1} = (\mu\gamma)^n = \lambda$. Thus,

$\lambda\mu = \mu\lambda = \lambda$ and by definition,

$$\lambda \leq \mu.$$

By a similar argument

$$\lambda \leq \gamma$$

and so λ is a lower bound of $\{\mu, \gamma\}$. To prove λ is the greatest lower bound, suppose ϕ is a lower bound of $\{\mu, \gamma\}$, i. e.

$$\phi\mu = \mu\phi = \phi$$

and $$\phi\gamma = \gamma\phi = \phi.$$

Then

$$\phi\lambda = \phi(\gamma\mu)^m = (\phi\gamma)\mu(\gamma\mu)^{m-1} = \phi\mu(\gamma\mu)^{m-1} = \phi(\gamma\mu)^{m-1}$$

and if this process is continued until all the μ and γ terms are absorbed,

$$\phi\lambda = \phi.$$

By a similar argument,

$$\lambda\phi = \phi$$

and therefore $\phi\lambda = \lambda\phi = \phi$ or, equivalently, $\phi \leq \lambda$. Hence λ is the greatest lower bound of $\{\mu, \gamma\}$, that is,

$$\mu \wedge \gamma = \lambda.$$

To prove i), since $\lambda \leq \mu$ and $\lambda \leq \gamma$, by Theorem 1.11,

$$\mathcal{R}(\lambda) \subseteq \mathcal{R}(\mu) \quad \text{and} \quad \mathcal{R}(\lambda) \subseteq \mathcal{R}(\gamma)$$

and therefore

$$\mathcal{R}(\lambda) \subseteq \mathcal{R}(\mu) \cap \mathcal{R}(\gamma).$$

Moreover, equality must hold for suppose $q \in \mathcal{R}(\mu) \cap \mathcal{R}(\gamma)$. Then

$$\lambda(q) = (\gamma\mu)^m(q) = (\gamma\mu)^{m-1}\gamma\,\mu(q)$$

$$= (\gamma\,\mu)^{m-1}\gamma(q), \quad \text{since } \mu \in S(Q)$$

$$= (\gamma\,\mu)^{m-1}(q), \quad \text{since } \gamma \in S(Q).$$

Continuing in this manner, $\lambda(q) = q$ which implies $q \in \mathcal{R}(\lambda)$. Therefore

$$\mathcal{R}(\lambda) = \mathcal{R}(\mu) \cap \mathcal{R}(\gamma)$$

To prove ii), since $\lambda \leq \mu$ and $\lambda \leq \gamma$, by Theorem 1.11,

$$\equiv_\mu \subseteq \equiv_\lambda \quad \text{and} \quad \equiv_\gamma \subseteq \equiv_\lambda$$

and therefore

$$\equiv_{\mu} + \equiv_{\gamma} \subseteq \equiv_{\lambda}.$$

Here again equality must hold for suppose $q \equiv_{\lambda} r$. Since $\mu, \gamma \in S(Q)$ we have

$$q \equiv_{\mu} \mu(q) \equiv_{\gamma} \gamma\mu(q) \equiv_{\mu} \cdots \equiv_{\gamma} (\gamma\mu)^n(q) = \lambda(q)$$

and so $q(\equiv_{\mu} + \equiv_{\gamma})\lambda(q)$. Similarly, $r(\equiv_{\mu} + \equiv_{\gamma})\lambda(r)$ and as $q \equiv_{\lambda} r$ means $\lambda(q) = \lambda(r)$, we have

$$q(\equiv_{\mu} + \equiv_{\gamma})r.$$

Thus $\equiv_{\lambda} = \equiv_{\mu} + \equiv_{\gamma}$ thereby completing the proof of the theorem.

A useful corollary of Theorem 1.12 is the special case where $\gamma\mu = \mu\gamma$, that is

Corollary 1.12.1

If $\mu, \gamma \in S(Q)$ and $\gamma\mu = \mu\gamma$ then

$$\mu \wedge \gamma = \gamma\mu.$$

Moreover,

i) $\mathcal{R}(\gamma\mu) = \mathcal{R}(\mu) \cap \mathcal{R}(\gamma)$

and

ii) $\equiv_{\gamma\mu} = \equiv_{\mu} + \equiv_{\gamma}$.

Proof

If $\mu, \gamma \in S(Q)$ and $\gamma\mu = \mu\gamma$ then

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

Thus $\gamma\mu \in S(Q)$ and as $\mu\gamma = \gamma\mu$, $\mu\gamma \in S(Q)$. In other words, $\overline{\gamma\mu} = \gamma\mu$ and $\overline{\mu\gamma} = \gamma\mu$ from which the corollary follows by Theorem 1.12.

It will become obvious in the development that follows, that the facts established by Theorem 1.12 underlie the inherent difficulty in Q-masking stable faults. Before doing this, however, the theorem can be applied to obtain one of several alternative characterizations of stable faults that commute (relative to the operation of composition).

Theorem 1.13

If $\mu, \gamma \in S(Q)$ then the following statements are equivalent:

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

ii) $\gamma\mu$, $\mu\gamma \in S(Q)$ and $\gamma\mu = \mu \wedge \gamma$.

iii) $\gamma\mu$, $\mu\gamma \in S(Q)$ and $\gamma\mu \leq \mu, \gamma$.

iv) $\gamma\mu$, $\mu\gamma = S(Q)$ and $\mu\gamma\mu = \gamma\mu\gamma$.

v) $\mathcal{R}(\gamma\mu) \subseteq \mathcal{R}(\mu)$, $\mathcal{R}(\mu\gamma) \subseteq \mathcal{R}(\gamma)$, $\equiv_{\gamma} \subseteq \equiv_{\gamma\mu}$, and $\equiv_{\mu} \subseteq \equiv_{\mu\gamma}$.

Proof

i) $\Longrightarrow$ ii): If $\gamma\mu = \mu\gamma$ then, as in the proof of Corollary 1.12.1, $\gamma\mu,\ \mu\gamma \in S(Q)$. By Corollary 1.12.1, the meet $\mu \bigwedge \gamma$ is equal to $\gamma\mu$.

ii) $\Longrightarrow$ iii): If ii) holds, then by hypothesis, $\gamma\mu,\ \mu\gamma \in S(Q)$. Since $\gamma\mu$ is the greatest lower bound of $\{\mu,\gamma\}$ it is therefore a lower bound, i.e. $\gamma\mu \leq \mu$ and $\gamma\mu \leq \gamma$.

iii) $\Longrightarrow$ iv): If iii) holds, then by assumption, $\gamma\mu,\ \mu\gamma \in S(Q)$. Since $\gamma\mu \leq \mu$ we have $\mu(\gamma\mu) = \gamma\mu$ and as $\gamma\mu \leq \gamma$, $\gamma\mu = (\gamma\mu)\gamma$. Combining these identities, $\mu\gamma\mu = \gamma\mu\gamma$.

iv) $\Longrightarrow$ v): If iv) holds then

$$\gamma\mu = (\gamma\mu)^2 = (\gamma\mu\gamma)\mu = (\mu\gamma\mu)\mu = (\mu\gamma)\mu^2 = (\mu\gamma)\mu$$

Thus

$$1)\qquad \mathcal{K}(\gamma\mu) = \mathcal{K}(\mu\gamma\mu) \subseteq \mathcal{K}(\mu).$$

By a similar argument, beginning with $\mu\gamma$,

$$2)\qquad \mathcal{K}(\mu\gamma) = \mathcal{K}(\gamma\mu\gamma) \subseteq \mathcal{K}(\gamma).$$

Regarding the induced equivalence relations,

$$\gamma\mu = (\gamma\mu)^2 = \gamma(\mu\gamma\mu) = \gamma(\gamma\mu\gamma) = \gamma^2(\mu\gamma) = (\gamma\mu)\gamma.$$

Thus

$$3)\quad \underset{\gamma}{\equiv} \subseteq \underset{(\gamma\mu)\gamma}{\equiv} = \underset{\gamma\mu}{\equiv}.$$

By the same argument, beginning with $\mu\gamma$,

$$4)\quad \underset{\mu}{\equiv} \subseteq \underset{(\mu\gamma)\mu}{\equiv} = \underset{\mu\gamma}{\equiv}.$$

v) $\Longrightarrow$ i): Suppose v) holds. By the first two conditions, μ and γ can follow each other and thus, by Corollary 1.10.1, both $\gamma\mu$ and $\mu\gamma$ are stable. Applying Theorem 1.11, by the first condition and the fact that $\underset{\mu}{\equiv} \subseteq \underset{\gamma\mu}{\equiv}$ we have $\gamma\mu \leq \mu$. By the fourth condition and the fact that $\mathcal{R}(\mu\gamma) \subseteq \mathcal{R}(\mu)$ we have $\mu\gamma \leq \mu$. Thus

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

completing the proof of the theorem.

Although some of the characterizations of Theorem 1.13 are useful only in deriving further properties of stable faults, the equivalence of statements i) and iii) has a more direct interpretation. In general, if ϕ and ξ are stable faults of M and $\phi \leq \xi$ then ϕ dominates ξ in the sense that the result of either the succession (ξ, ϕ) (see Def. 1.10) or the succession (ϕ, ξ) is the machine M^{ϕ}. Accordingly, the equivalence of i) and iii) says that the order in which stable faults μ and γ occur is irrelevant if and only if i) the combined

faults $\gamma\mu$ and $\mu\gamma$ are stable and ii) the combination $\gamma\mu$ "dominates" both μ and γ.

Having investigated the structure of stable faults (more precisely, the structure of the partially ordered set $(S(Q), \leq)$, we now wish to examine the effect of the stability restriction on fault masking. In general, if E is a relation on a set Q and $R \subseteq Q$ we will let

$$E|R$$

denote the restriction of E to R, that is, $E|R = E \cap (R \times R)$. In case μ is stable, sufficient conditions for R-masking can be weakened as follows (compare with Theorem 1.5):

Theorem 1.14

If μ is stable fault of M and $R \subseteq Q$ such that

i) $\delta(\mu(R) \times I) \subseteq R$, and

ii) $\underset{\mu}{\equiv} |(R \cup \mu(R)) \subseteq \equiv |(R \cup \mu(R))$

then μ is R-masked.

Proof

By Theorem 1.5 it suffices to show that condition ii) implies $\mu(r) \equiv r$, for all $r \in R$. Suppose then that $r \in R$. Since μ is stable $\mu(r) \underset{\mu}{\equiv} r$ (cf. Theorem 1.8) and as $r \in R$, $\mu(r) \in \mu(R)$, by condition ii), $\mu(r) \equiv r$.

As with Theorem 1.5, the following results follow as corollaries.

Corollary 1.14.1

If conditions i) and ii) of Theorem 1.14 hold and R is complete then a stable fault μ is i-masked.

Corollary 1.14.2

If μ is stable and $\equiv \subseteq \equiv$ then μ is Q-masked (and hence e-masked).
μ

Proof

If $R_i = Q$, condition i) is automatically satisfied and the equivalence relations of condition ii) are no longer properly restricted.

In other words, Corollary 1.14.2 says that if, whenever two states q and r fail to the same state in the faulty machine (i.e., $\mu(q) = \mu(r)$) they are equivalent in the fault-free machine (i.e., $q \equiv r$), then μ will be Q-masked. A rather surprising fact is that this condition is necessary as well, that is

Theorem 1.15

If μ is a stable fault of M then μ is Q-masked if and only if $\equiv \subseteq \equiv$.
μ

Proof

Sufficiency is given by Corollary 1.14.2. To prove necessity suppose μ is Q-masked and $q, r \in Q$ such that $q \underset{\mu}{\equiv} r$. Then $\mu(q) = \mu(r)$ and therefore

$$\beta^{\mu}_{\mu(q)} = \beta^{\mu}_{\mu(r)}.$$

But since μ is Q-masked $\beta_q = \beta^{\mu}_{\mu(q)}$ and $\beta_r = \beta^{\mu}_{\mu(r)}$. Combining these identities,

$$\beta_q = \beta_r$$

or, by definition,

$$q \equiv r.$$

Thus, $q \underset{\mu}{\equiv} r$ implies $q \equiv r$, that is

$$\underset{\mu}{\equiv} \subseteq \equiv.$$

In case M is a reduced machine, $\equiv$ becomes $=$ and as the only fault with $=$ as its induced equivalence relation is the identity function we have

Corollary 1.15.1

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

This corollary is not disturbing for one would expect that a fault tolerant machine should have redundant states. However, the restrictive nature of Q-masking becomes visible in the light of Theorem 1.15 for it says that a stable fault is Q-masked if and only if states failing to the same state are equivalent in the fault-free machine. This fact, along with the knowledge of the structure of $(S(Q), \leq)$ developed earlier, enables us to establish the following important result.

Theorem 1.16

If M is a machine with states Q and $F = \{\mu, \gamma, \ldots, \xi\} \subseteq S(Q)$ such that each fault in F is Q-masked and each pair of faults in F commute, then the combined fault $\xi \cdots \gamma\mu$ is Q-masked.

Proof

Suppose F is a set of stable faults of M satisfying the conditions of the hypothesis. Since each pair of faults in F commute, by repeated application of Corollary 1.12.1 we have

$$\text{g.l.b. } F = \mu \wedge \gamma \wedge \ldots \wedge \xi = \xi \cdots \gamma\mu$$

where if we let $\lambda = \xi \cdots \gamma\mu$ then

$$\underset{\lambda}{\equiv} = \underset{\mu}{\equiv} + \underset{\gamma}{\equiv} + \ldots + \underset{\xi}{\equiv} . \qquad (1.17)$$

As each fault in F is Q-masked, by the previous theorem

$$\underset{\mu}{\equiv}, \underset{\gamma}{\equiv}, \ldots, \underset{\xi}{\equiv} \subseteq \equiv$$

that is, state equivalence is an upper bound (in the lattice of equivalence relations on Q) of each of the induced relations. But by (1.17), $\underset{\lambda}{\equiv}$ is the least upper bound and so

$$\underset{\lambda}{\equiv} \subseteq \equiv .$$

Applying Theorem 1.15 once more, $\lambda = \xi \cdots \gamma\mu$ must be Q-masked, thereby establishing the desired result.

Paraphrasing Theorem 1.16, if F is a set of stable, Q-masked faults such that when combined, the order in which they occur is irrelevant, then the combination of all the faults in F must also be Q-masked. Moreover, any other combination of faults in F must also be Q-masked since the conditions of the hypothesis hold for any nonempty subset of F. Thus, from a synthesis point of view, whenever each fault in a set of stable, commuting faults is to be Q-masked, all combinations of those faults must also be Q-masked. This poses a severe design constraint and, as is shown later in the development, it is impossible to Q-mask even reasonably small sets of faults.

Stuck-at Faults

Let us now focus our attention on the problem of designing fault tolerant sequential switching networks which, for the purposes of the following investigation, can be represented as "state-assigned" sequential machines, that is,

Definition 1.17

A sequential machine $M = (I, Q, O, \delta, \omega)$ is state-assigned if $Q = \{0,1\}^{(n)}$ where n is a positive integer called the dimension of M.

In other words, M is state-assigned of dimension n if its state set is the set of all n-tuples of 0's and 1's. State-assigned machines are a direct representation of sequential networks. Dimension

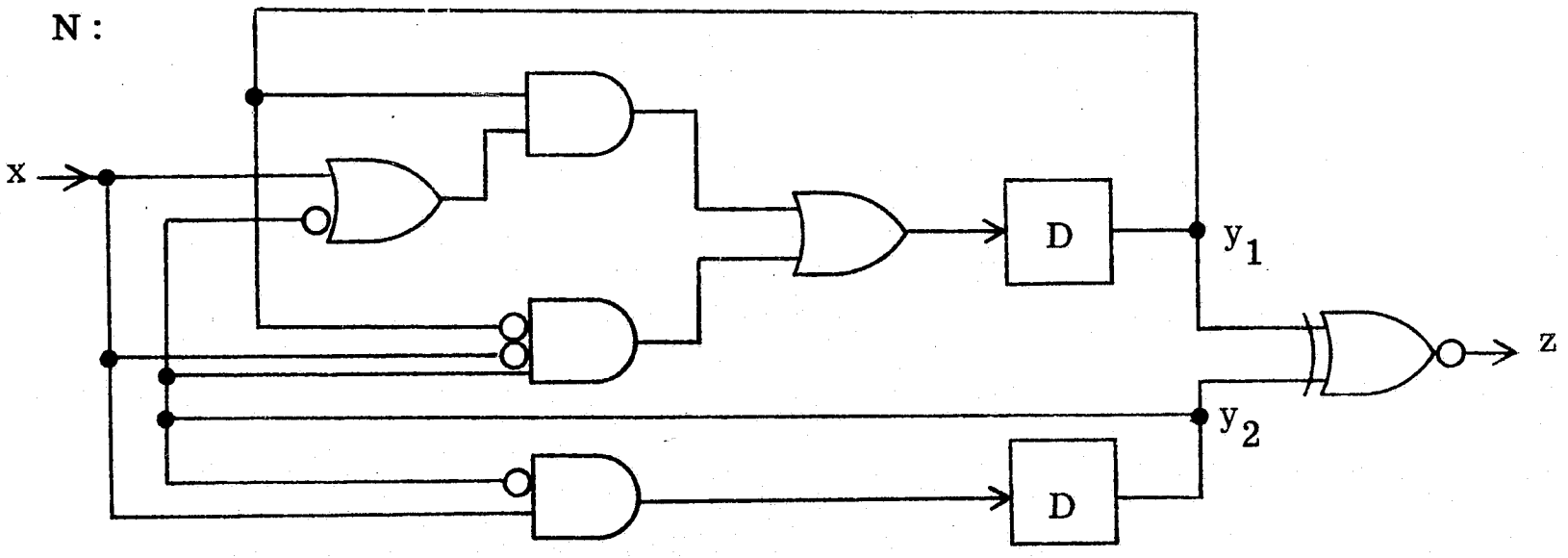
corresponds to the number of flip-flops (which we assume to be delay flip-flops with output voltage levels V_0 and V_1) and a particular n-tuple $(b_1, b_2, \ldots, b_n)$ has the interpretation:

$$b_i = \begin{cases} 1 \text{ if the } i^{th} \text{ flip-flop output is } V_1 \\ 0 \text{ if the } i^{th} \text{ flip-flop output is } V_0, \end{cases}$$

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

Example 1.3

If **N** is the sequential network



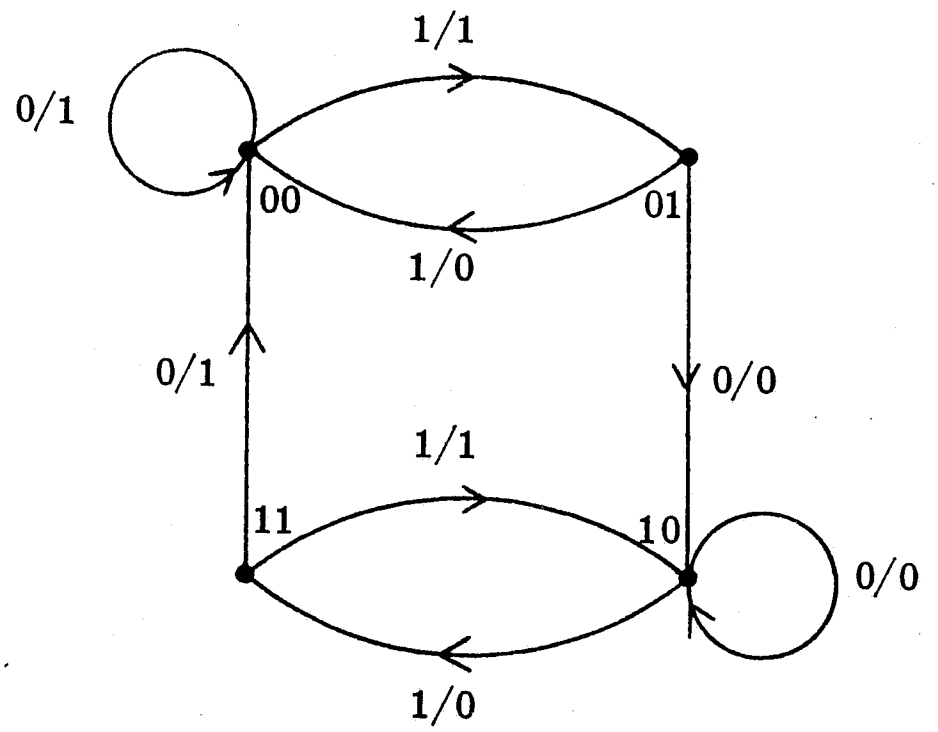
then N is represented by a state assigned machine M having transition-output table:

M:

(y_1, y_2) \ x	0	1
00	00/1	01/1
01	10/0	00/0
10	10/0	11/0
11	00/1	10/1

and transition graph:

G_M:



A sequential network memory fault which consists of various flip-flops "stuck-at 0" and "stuck-at 1" can be represented as follows in the corresponding state-assigned machine. Let

$$S(\{0,1\}) = \{\sigma_0, \sigma_1, \sigma_x\}$$

denote the three stable faults of a machine with states $\{0,1\}$ where

q	$\sigma_0(q)$	$\sigma_1(q)$	$\sigma_x(q)$
0	0	1	0
1	0	1	1

then

Definition 1.18

If M is a state-assigned machine of dimension n and μ is a fault of M then μ is a stuck-at fault if, for all $(b_1, b_2, \ldots, b_n) \in Q$, μ can be expressed as

$$\mu(b_1, b_2, \ldots, b_n) = (\mu_1(b_1), \mu_2(b_2), \ldots, \mu_n(b_n))$$

where $\mu_i \in \{\sigma_0, \sigma_1, \sigma_x\}$, $i = 1, 2, \ldots, n$. If $\mu_i = \sigma_0$ the i^{th} coordinate is stuck-at 0 (sa 0); if $\mu_i = \sigma_i$ the i^{th} coordinate is stuck-at 1 (sa 1). The degree $d(\mu)$ of μ is the number of coordinates which are sa 0 or sa 1, i.e.

$$d(\mu) = |\{i \,|\, \mu_i \in \{\sigma_0, \sigma_1\}\}|$$

If M is an n-dimensional state-assigned machine, let

$$S_n = \{\mu \,|\, \mu \text{ is a stuck-at fault of M}\}.$$

Then there is obviously a 1-1 correspondence between S_n and the set $S(\{0,1\})^{(n)}$ of all n-tuples of stable faults of a 2-state machine. Thus a stuck-at fault μ will sometimes be written

$$\mu = (\mu_1, \mu_2, \ldots, \mu_n) \text{ where } \mu_i \in S(\{0,1\}), \quad i = 1, 2, \ldots, n. \tag{1.18}$$

From the definition it follows immediately that

Theorem 1.17

If $\mu, \gamma \in S_n$, $\mu = (\mu_1, \ldots, \mu_n)$ and $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ then

$$\gamma\mu = (\gamma_1\mu_1, \gamma_2\mu_2, \ldots, \gamma_n\mu_n).$$

Corollary 1.17.1

Every stuck-at fault is stable.

Proof

If $\mu \in S_n$, $\mu\mu = (\mu_1\mu_1, \mu_2\mu_2, \ldots, \mu_n\mu_n)$ and since each μ_i is stable, $\mu\mu = (\mu_1, \mu_2, \ldots, \mu_n) = \mu$.

Corollary 1.17.2

S_n is closed under composition, i.e., $\mu, \gamma \in S_n$ implies $\gamma\mu \in S_n$.

Proof

Since $S(\{0,1\})$ is closed under composition, viz.

σ' \ σ	σ_0	σ_1	σ_x
σ_0	σ_0	σ_1	σ_0
σ_1	σ_0	σ_1	σ_1
σ_x	σ_0	σ_1	σ_x

$\sigma'\sigma$ (1.19)

it follows, by Theorem 1.17, that S_n is closed.

The system $(S_n, 0)$ of all stuck-at faults of an n-dimensional state assigned machine along with the operation of composition is, therefore, a semigroup. Moreover, since very element of S_n is stable (idempotent), $(S_n, 0)$ is a band [3]. (Note that $S(\{0,1\})$ is also a band.)

A stuck-at fault of degree k can also be regarded as a projection of the n-cube $Q = \{0,1\}^{(n)}$ onto some (n-k)-subcube of Q. Moreover, the subcube $\mathcal{R}(\mu)$ uniquely determines μ. To be more precise, let $a \in \{0,1,x\}$ and let

$$B^a = \begin{cases} \{0\} \text{ if } a = 0 \\ \{1\} \text{ if } a = 1 \\ \{0,1\} \text{ if } a = x. \end{cases}$$

Then, following the cubical notation of [21], we let the sequence

$$a_1 a_2 \ldots a_n \quad (a_1 \in \{0,1,x\})$$

denote the subcube $C = B^{a_1} \times B^{a_2} \times \ldots \times B^{a_n}$. The i^{th} coordinate of C is free if $a_i = x$ and the dimension of C is the number of free coordinates.

Theorem 1.18

If $\mu \in S_n$ then

$$\mu = (\sigma_{a_1}, \sigma_{a_2}, \ldots, \sigma_{a_n}) \text{ if and only if } \mathcal{R}(\mu) = a_1 a_2 \ldots a_n.$$

Proof

By definition of σ_0, σ_1, and σ_x,

$$(\sigma_0) = \{0\} = B^0$$

$$(\sigma_1) = \{1\} = B^1, \text{ and}$$

$$(\sigma_x) = \{0,1\} = B^x.$$

Thus

$$\sigma = \sigma_a \Leftrightarrow \mathcal{R}(\sigma) = B^a$$

and as

$$\mathcal{R}((\sigma_{a_1}, \sigma_{a_2}, \ldots, \sigma_{a_n})) = \mathcal{R}(\sigma_{a_1}) \times \mathcal{R}(\sigma_{a_2}) \times \ldots \times \mathcal{R}(\sigma_{a_n})$$

the desired result follows.

Corollary 1.18.1

The function given by $\mu \longmapsto \mathcal{R}(\mu)$ is a 1-1 correspondence between S_n and the set of all subcubes of the n-cube.

Corollary 1.18.2

μ has degree k if and only if $\mathcal{R}(\mu)$ has dimension n-k.

Corollary 1.18.3

If M has dimension n then there are $\binom{n}{k} 2^k$ stuck-at faults of M of degree k ($0 \leq k \leq n$).

Since stuck-at faults are uniquely determined by their range, we will sometimes denote a stuck-at fault μ by the subcube $\mathcal{R}(\mu)$, e.g. if

$$\mu = (\sigma_x, \sigma_0, \sigma_x, \sigma_1)$$

we would write

$$\mu = x0x1.$$

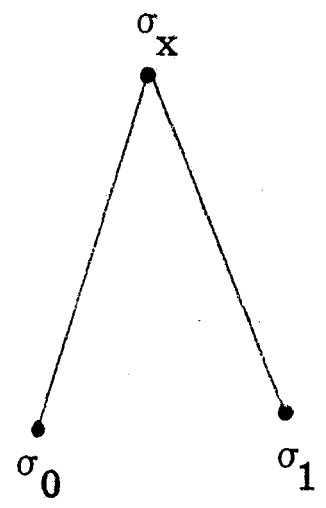
In cubical notation, the composition of two stuck at faults $\mu = a_1 a_2 \ldots a_n$ and $\gamma = b_1 b_2 \ldots b_n$ is the fault $\gamma\mu = c_1 c_2 \ldots c_n$ where c_i is given by the following table:

a_i \ b_i	0	1	x
0	0	1	0
1	0	1	1
x	0	1	x

c_i (1.20)

Let us now consider the structure of the partially ordered set $(S_n, \leq)$ of stuck-at faults relative to the natural partial ordering $\leq$. As one might expect, the structure of $(S_n, \leq)$ enjoys more properties than does $(S(Q), \leq)$.

The properties of $(S_n, \leq)$ are primarily determined by the structure of the partially ordered set $(S(\{0,1\}), \leq)$ of stable faults of a two-state machine. We note first that $(S(\{0,1\}), \leq)$ has the following Hasse diagram.



Thus $S(\{0,1\})$ is an upper semilattice where if σ, $\sigma' \in S(\{0,1\})$, the join $\sigma \vee \sigma'$ (least upper bound of $\{\sigma, \sigma'\}$ is given by the following table:

σ \ σ'	σ_0	σ_1	σ_x
σ_0	σ_0	σ_x	σ_x
σ_1	σ_x	σ_1	σ_x
σ_x	σ_x	σ_x	σ_x

$\sigma \vee \sigma'$ (1.21)

Although $S(\{0,1\})$ is not a lower semilattice (and hence not a lattice), the meet $\sigma \wedge \sigma'$, whenever it exists, is given by the following table:

σ \ σ'	σ_0	σ_1	σ_x
σ_0	σ_0	—	σ_0
σ_1	—	σ_1	σ_1
σ_x	σ_0	σ_1	σ_x

$\sigma \wedge \sigma'$ (1.22)

Since S_n is essentially the n-fold cartesian product of $S(\{0,1\})$, it is immediate, by Theorem 1.17, that the natural partial ordering $\leq$ of S_n can be expressed in terms of the partial ordering of $S(\{0,1\})$ as follows:

Theorem 1.19

If $\mu = (\mu_1, \mu_2, \ldots, \mu_n)$ and $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ then

$$\mu \leq \gamma \text{ iff } \mu_i \leq \gamma_i,\ i = 1, 2, \ldots, n.$$

Accordingly, all the properties just observed for $(S(\{0,1\}, \leq)$ transport to $(S_n, \leq)$ and are summarized in the following theorem.

Theorem 1.20

$(S_n, \leq)$ is an upper semilattice where if $\mu = (\mu_1, \mu_2, \ldots, \mu_n)$ and $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ then

$$\mu \vee \gamma = (\mu_1 \vee \gamma_1, \mu_2 \vee \gamma_2, \ldots, \mu_n \vee \gamma_n).$$

$\mu \wedge \gamma$ exists iff $\mu_i \wedge \gamma_i$ exists $(1 \leq i \leq n)$, and if $\mu \wedge \gamma$ exists then

$$\mu \wedge \gamma = (\mu_1 \wedge \gamma_1, \mu_2 \wedge \gamma_2, \ldots, \mu_n \wedge \gamma_n).$$

Having already investigated properties of the ordering $\leq$ for the more general class of stable faults, let us consider their specialization to stuck-at faults. First of all, the characterization of Theorem 1.11 reduces to

Theorem 1.21

If $\mu, \gamma \in S_n$ then

$$\mu \leq \gamma \text{ if and only if } \mathcal{R}(\mu) \subseteq \mathcal{R}(\gamma).$$

Proof

Since stuck-at faults μ and γ are uniquely determined by their ranges (Theorem 1.18), so must be the ordering $\leq$. Therefore, the necessary condition $\mathcal{R}(\mu) \subseteq \mathcal{R}(\gamma)$, given by Theorem 1.11, must also be sufficient.

Thus if μ and γ are stuck-at faults then μ dominates γ if and only if the range of μ is contained in the range of γ or, in terms of the machines M^{μ} and M^{γ}, if and only if every state of M^{μ} is a state of M^{γ}. This fact is illustrated in the following example.

Example 1.4

Consider the partially ordered set $(S_2, \leq)$ of all stuck-at faults of a 2-dimensional state assigned machine. Then, by Definition 1.18,

$$S_2 = \left\{ \begin{array}{l} (\sigma_x, \sigma_x), (\sigma_x, \sigma_0), (\sigma_x, \sigma_1), (\sigma_0, \sigma_x), (\sigma_1, \sigma_x), \\ (\sigma_0, \sigma_0), (\sigma_0, \sigma_1), (\sigma_1, \sigma_0), (\sigma_1, \sigma_1) \end{array} \right\}$$

where the function tables of these nine stuck-at faults are as follows:

q \ μ	(σ_x, σ_x)	(σ_x, σ_0)	(σ_x, σ_1)	(σ_0, σ_x)	(σ_1, σ_x)	(σ_0, σ_0)	(σ_0, σ_1)	(σ_1, σ_0)	(σ_1, σ_1)
(0,0)	(0,0)	(0,0)	(0,1)	(0,0)	(1,0)	(0,0)	(0,1)	(1,0)	(1,1)
(0,1)	(0,1)	(0,0)	(0,1)	(0,1)	(1,1)	(0,0)	(0,1)	(1,0)	(1,1)
(1,0)	(1,0)	(1,0)	(1,1)	(0,0)	(1,0)	(0,0)	(0,1)	(1,0)	(1,1)
(1,1)	(1,1)	(1,0)	(1,1)	(0,1)	(1,1)	(0,0)	(0,1)	(1,0)	(1,1)

$\mu(q)$

By the function tables (or applying Theorem 1.18), we have

$$\mathcal{R}((\sigma_x, \sigma_x)) = \{(0,0),(0,1),(1,0),(1,1)\} = xx$$

$$\mathcal{R}((\sigma_x, \sigma_0)) = \{(0,0),(1,0)\} = x0$$

$$\mathcal{R}((\sigma_x, \sigma_1)) = \{(0,1),(1,1)\} = x1$$

$$\mathcal{R}((\sigma_0, \sigma_x)) = \{(0,0),(0,1)\} = 0x$$

$$\mathcal{R}((\sigma_1, \sigma_x)) = \{(1,0),(1,1)\} = 1x$$

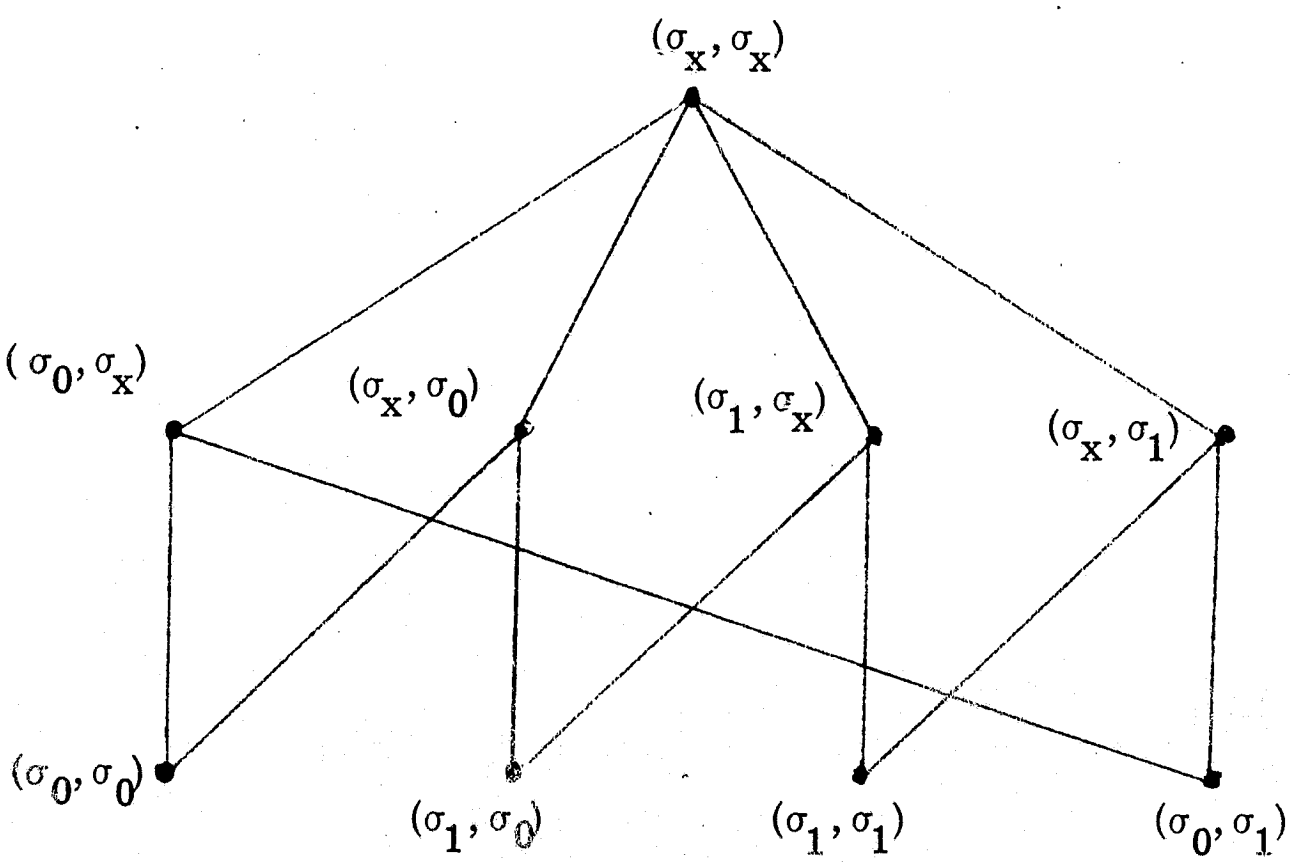
$$\mathcal{R}((\sigma_0, \sigma_0)) = \{(0,0)\} = 00$$

$$\mathcal{R}((\sigma_0, \sigma_1)) = \{(0,1)\} = 01$$

$$\mathcal{R}((\sigma_1, \sigma_0)) = \{(1,0)\} = 10$$

$$\mathcal{R}((\sigma_1, \sigma_1)) = \{(1,1)\} = 11$$

Accordingly, by Theorem 1.21, $(S_2, \leq)$ has the following Hasse diagram:



We have already observed that stuck-at faults are closed under composition (Corollary 1.17.2). Given this fact, coupled with the properties of $\leq$ given by the last three theorems, it is relatively easy to show that several different concepts (introduced earlier for faults and stable faults) become equivalent when restricted to stuck-at faults.

Theorem 1.22

If $\mu, \gamma \in S_n$ then the following statements are equivalent:

i) γ can follow μ (i.e., $\mathcal{R}(\gamma\mu) \subseteq \mathcal{R}(\mu)$)

ii) $\mathcal{R}(\gamma) \cap \mathcal{R}(\mu) \neq \emptyset$.

iii) $\mu \wedge \gamma$ exists

iv) $\mu \wedge \gamma = \mu\gamma$

v) $\gamma\mu = \mu\gamma$

vi) $\gamma\mu \leq \mu$

Proof

i) $\Longrightarrow$ ii): Suppose $\mathcal{R}(\gamma\mu) \subseteq \mathcal{R}(\mu)$. Since, for any faults μ and γ, $\mathcal{R}(\gamma\mu) \subseteq \mathcal{R}(\gamma)$ we have $\mathcal{R}(\gamma\mu) \subseteq \mathcal{R}(\mu) \cap \mathcal{R}(\gamma)$. As $\mathcal{R}(\gamma\mu) \neq \emptyset$, ii) holds.

ii) $\Longrightarrow$ iii): Suppose $\mathcal{R}(\gamma) \cap \mathcal{R}(\mu) \neq \emptyset$ and let $\mu = (\mu_1, \mu_2, \ldots, \mu_n)$ $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)$. Then $\mathcal{R}(\mu_i) \cap \mathcal{R}(\gamma_i) \neq \emptyset$ $(1 \leq i \leq n)$ which implies (since $\mu_i, \gamma_i \in S(\{0,1\})$) that $\mu_i \leq \gamma_i$ or $\gamma_i \leq \mu_i$ $(1 \leq i \leq n)$. Hence $\mu_i \wedge \gamma_i$ exists $(1 \leq i \leq n)$ and by Theorem 1.20, $\mu \wedge \gamma$ exists.

iii) $\Longrightarrow$ iv): Suppose $\mu \wedge \gamma$ exists. Then, by Theorem 1.20, $\mu_i \wedge \gamma_i$ exists $(1 \leq i \leq n)$ and comparing Table (1.22) with (1.19), it follows that $\mu_i \wedge \gamma_i = \gamma_i\mu_i$ $(1 \leq i \leq n)$. Thus, by Theorem 1.20,

$$\mu \wedge \gamma = (\gamma_1\mu_1, \gamma_2\mu_2, \ldots, \gamma_n\mu_n)$$

and by Theorem 1.17, we have $\mu \wedge \gamma = \gamma\mu$.

iv) $\Longrightarrow$ v): Suppose $\mu \wedge \gamma = \gamma\mu$. Since S_n is closed under composition, both $\gamma\mu$ and $\mu\gamma$ are stable and applying Theorem 1.13 (ii) $\Longrightarrow$ i)), $\gamma\mu = \mu\gamma$.

v) $\Longrightarrow$ vi): Suppose $\gamma\mu = \mu\gamma$. Then, by Theorem 1.13 (i) $\Rightarrow$ iii)), we have $\gamma\mu \leq \mu$.

vi) $\Longrightarrow$ i): If $\gamma\mu \leq \mu$, then by Theorem 1.21, $R(\gamma\mu) \subseteq R(\mu)$, i.e., γ can follow μ.

Most interesting is the equivalence of statements i), iii), and v), i.e., "γ can follow μ", "meet of μ and γ exists", and "γ commutes with μ" all mean the same thing in the case of stuck-at faults. On the other hand, no two of the statements of Theorem 1.22 are equivalent relative to the more general class of stable faults.

Fault-Tolerant State-Assigned Machines

Let us now consider the application of the above results to the design of fault-tolerant sequential networks or, what is the same for our purposes, the design of fault-tolerant state-assigned machines. (The latter representation suffices since the faults under consideration are memory faults.) In general, the design problem can be stated as follows:

> Given some sequential-machine-realizable behavior B specified, say, by a reduced machine M' such that $B_{M'} = B$ (cf. Definition 1.4), design a state-assigned

machine M that realizes* M' and relative to some specified set of faults F of M, μ is $\square$-masked for all $\mu \in F$ ($\square$ denotes the specific type of fault masking desired, (e.g. e-, i-, or R-masking).

In what follows we consider first the problem of designing machines which Q-mask faults (i.e., R-mask where R = Q) and will restrict our attention to stuck-at faults. In this case we obtain a surprising negative result, namely, if M is an n-dimensional realization of a machine M' with at least two nonequivalent states then it is impossible to Q-mask certain sets of degree 1 stuck-at faults with as few as n members. More precisely, if we let S_n^0 and S_n^1 denote the set of all degree 1 faults stuck-at zero and stuck-at one, respectively, that is:

$$S_n^0 = \{\mu \in S_n \mid d(\mu) = 1 \text{ and } \mu_i \in \{\sigma_0, \sigma_x\},\ i = 1, 2, \ldots, n\}$$

and

$$S_n^1 = \{\mu \in S_n \mid d(\mu) = 1 \text{ and } \mu_i \in \{\sigma_1, \sigma_x\},\ i = 1, 2, \ldots, n\}$$

then

---

* "Realizes" is defined in the sense of Hartmanis and Stearns [11], p. 28, Def. 1.15.

Theorem 1.23

If M' is a machine with at least two nonequivalent states then, for all n, there does not exist a state-assigned machine M of dimension n (i.e. $Q = \{0,1\}^{(n)}$) such that M realizes M' and μ is Q-masked, for all $\mu \in S_n^0$ (alternatively, for all $\mu \in S_n^1$).

Proof

We will prove the theorem for degree 1 stuck-at 0 faults (the argument for the stuck-at 1 case being the same) and suppose to the contrary, i.e., for some n there is a state-assigned machine M that realizes M' and, for all $\mu \in S_n^0$, μ is Q-masked. Using cubical notation to denote the range of a stuck-at fault, if $\mu, \gamma \in S_n^0$, then

$$\mathcal{R}(\mu) \cap \mathcal{R}(\gamma) = a_1 a_2 \ldots a_n$$

where

$$a_i = \begin{cases} 0 \text{ if } \mu_i = \sigma_0 \text{ or } \gamma_i = \sigma_0 \\ x \text{ otherwise.} \end{cases}$$

Thus $\mathcal{R}(\mu) \cap \mathcal{R}(\gamma) \neq \emptyset$ and by Theorem 1.22 (ii) $\Longrightarrow$ iv) $\gamma\mu = \mu\gamma$, i.e., every pair of faults in S_n^0 commute. By assumption, every fault in S_n^0 is Q-masked and therefore the important fact established in the previous subsection (Theorem 1.16) can be applied to conclude that the combined fault

$$\lambda = \prod_{\mu \in S_n^0} \mu$$

is also Q-masked. But by the composition rules for stuck-at faults (Theorem 1.17 and Table (1.19)),

$$\lambda = (\sigma_0, \sigma_0, \ldots, \sigma_0)$$

that is,

$$\lambda(q) = (0, 0, \ldots, 0), \text{ for all } q \in Q.$$

Thus all states are equivalent under the e-relation induced by λ, i.e.

$$\equiv_\lambda = Q \times Q.$$

As λ is Q-masked, by Theorem 1.15,

$$\equiv_\lambda \subseteq \equiv$$

and therefore it must be the case that

$$\equiv = Q \times Q$$

that is, all states of the realization M are behaviorally equivalent. This contradicts the assumption that M realizes a machine M' with at least two nonequivalent states, thereby proving the theorem.

Since, in most applications, we would hope to be able to mask at least the set of all stuck-at faults of degree 1, Theorem 1.23 says that Q-masking is too restrictive as a practical fault-tolerance constraint. At first glance, this theorem seems to contradict the theory of von Neumann [27] since by triplicating a network that realizes M' and then voting, it would appear that all single stuck-at

faults in the delay flip-flops could be Q-masked. This is not the case, however, since the voting scheme assumes that the networks are "synchronized" thereby restricting the number of states in the composite network that can be used as initial states. If we call this restricted set R, then, as is established in the discussion that follows, all single stuck-at faults can be R-masked. However, $R \subset Q$ and by Theorem 1.23 all these faults cannot be Q-masked.

Let us consider, then, the problem of R-masking stuck-at faults where we allow R to be a proper subset of Q but require that R be complete (cf. Theorem 1.4). In order to simplify the discussion that follows it is convenient to introduce the following notation with regard to stuck-at faults and the states of a state assigned machine. If $\mu \in S_n$ let $X(\mu)$ $0(\mu)$, $1(\mu)$, and $C(\mu)$ denote the index sets of the free coordinates, stuck-at 0 coordinates, stuck-at 1 coordinates, and stuck-at coordinates, respectively, that is, if $\mu = (\mu_1, \mu_2, \ldots, \mu_n)$ (see (1.18)) then

$$\begin{aligned} X(\mu) &= \{i \,|\, \mu_i = \sigma_x\}, \\ 0(\mu) &= \{i \,|\, \mu_i = \sigma_0\}, \\ 1(\mu) &= \{i \,|\, \mu_i = \sigma_1\}, \\ C(\mu) &= 0(\mu) \cup 1(\mu). \end{aligned} \tag{1.23}$$

Similarly if $q = (b_1, b_2, \ldots, b_n) \in \{0,1\}^{(n)}$, let

$$0(q) = \{i \mid b_i = 0\},$$
$$1(q) = \{i \mid b_i = 1\}, \tag{1.24}$$

If $q, q' \in \{0,1\}^{(n)}$, we will let $H(q,q')$ denote the set of coordinate indices for which the coordinates of q and q' differ, that is

$$H(q,q') = 1(q) \oplus 1(q') \tag{1.25}$$

where $\oplus$ denotes symmetric difference. Thus, if $q, q' \in \{0,1\}^{(n)}$, the Hamming distance [20] $h(q,q')$ is given by

$$h(q,q') = |H(q,q')|. \tag{1.26}$$

Note also that

$$H(q,q'') = H(q,q') \oplus H(q',q''), \tag{1.27}$$

a relationship that is quite useful in computing distances.

The following observations relate the index sets of a state q to those of the "faulty" state $\mu(q)$ where, in general, if $A, B \subseteq N_n = \{1, 2, \ldots, n\}$ we will let AB denote $A \cap B$ and let $\overline{A} = N_n - A$.

Lemma 1.3

If $\mu \in S_n$ and $q \in \{0,1\}^{(n)}$ then

i) $0(\mu(q)) = 0(\mu) \cup X(\mu)\ 0(q)$

ii) $1(\mu(q)) = 1(\mu) \cup X(\mu)\ 1(q)$

Proof

If $q = (b_1, b_2, \ldots, b_n)$ then $\mu(q) = (\mu_1(b_1), \mu_2(b_2), \ldots, \mu_n(b_n))$.

Since $\mu_i(b_i) = 0$ if and only if either $\mu_i = \sigma_0$ or $\mu_i = \sigma_x$ and $b_i = 0$,

i) holds by the definitions given above. The proof of ii) is similar.

Lemma 1.4

If $\mu \in S_n$ and $q \in \{0,1\}^{(n)}$, then

$$H(q, \mu(q)) = 1(q)\ 0(\mu) \cup 0(q)\ 1(\mu).$$

Proof

By (1.25),

$$H(q, \mu(q)) = 1(q) \oplus 1(\mu(q))$$

and substituting according to the previous theorem

$$\begin{aligned} H(q, \mu(q)) &= 1(q) \oplus (1(\mu) \cup X(\mu)\ 1(q)) \\ &= 1(q)\ (\overline{1(\mu)}\,(\overline{X(\mu)} \cup \overline{1(q)})) \cup \overline{1(q)}\ 1(\mu) \\ &= 1(q)\ (\overline{1(\mu)}\ \overline{X(\mu)}) \cup \overline{1(q)}\ 1(\mu) \\ &= 1(q)\ \overline{(1(\mu) \cup X(\mu))} \cup \overline{1(q)}\ 1(\mu) \\ &= 1(q)\ 0(\mu) \cup 0(q)\ 1(\mu). \end{aligned}$$

Since the Hamming distance $h(q, \mu(q))$ is given by $|H(q, \mu(q))|$, Lemma 1.4 can be applied to obtain the following properties of distance relative to a stuck-at fault μ.

Theorem 1.24

If $\mu \in S_n$, $d(\mu) = k$, and $q, r \in \{0,1\}^{(n)}$ then

i) $h(q, \mu(q)) \leq k$

ii) $h(q, \mu(q)) = 0 \iff 1(q)\ 0(\mu) = \emptyset$ and $0(q)\ 1(\mu) = \emptyset$

iii) $h(q, \mu(q)) = k \iff 1(q) \subseteq 0(\mu)$ and $0(q) \subseteq 1(\mu)$

iv) $\mu(q) = \mu(r) \Rightarrow h(q, r) \leq k$

Proof

i) Since $H(q, \mu(q)) \subseteq 0(\mu) \cup 1(\mu)$, by Lemma 1.4, and

$$0(\mu) \cup 1(\mu) = C(\mu)$$

we have $|H(q, \mu(q))| \leq |C(\mu)|$, or by definition

$$h(q, \mu(q)) \leq k.$$

ii) $h(q, \mu(q)) = 0 \iff H(q, \mu(q)) = \emptyset$ from which ii) follows by Lemma 1.4.

iii) $h(q, \mu(q)) = k \iff H(q, \mu(q)) = 0(\mu) \cup 1(\mu)$ from which iii) follows by Lemma 1.4.

iv) If $\mu(q) = \mu(r)$ then, by (1.27)

$$H(q, r) = H(q, \mu(q)) \oplus H(\mu(r), r)$$

$$= [1(q)\ 0(\mu) \cup 0(q)\ 1(\mu)] \oplus [1(r)\ 0(\mu) \cup 0(r)\ 1(\mu)].$$

Thus

$$H(q, r) \subseteq 1(q)\ 0(\mu) \cup 0(\mu)\ 1(\mu) \cup 1(r)\ 0(\mu) \cup 0(r)\ 1(\mu)$$

$$\subseteq 0(\mu) \cup 1(\mu)\,.$$

and so

$H(q, r) \subseteq C(\mu)$ which implies $h(q, r) \leq k$.

In what follows we wish to consider the problem of masking all stuck-at faults of i) a given degree or ii) less than or equal to some given degree. To distinguish these cases let

$$S_n(k) = \{\mu \in S_n \mid d(\mu) = k\}, \qquad (1.28)$$

and

$$S_n[t] = \{\mu \in S_n \mid d(\mu) \leq t\}. \qquad (1.29)$$

thus

$$S_n[t] = \bigcup_{k=0}^{t} S_n(k)\,.$$

Given any integer $t \geq 0$ and any finite-state realizable behavior specified by some machine M', the following important result guarantees the existence of an n-dimensional state-assigned realization of M' such that relative to a complete subset R, all faults of degree $\leq t$ are R-masked. The state-assignment used requires that a certain subset of states have a minimum Hamming distance of $2t + 1$ and, in this sense, generalizes Russo's use [23] of distance 3 codes in the design of error tolerant counters.

Theorem 1.25

If M' is a machine and t is a nonnegative integer then there exists a state assigned machine M of dimension n and a complete subset R of the states of M such that M realizes M' and μ is R-masked, for all $\mu \in S_n[t]$.

Proof

Given a machine $M' = (I', Q', O', \delta', \omega')$ and a nonnegative integer t, without loss of generality we can assume M' is reduced (for if M' is not reduced, any realization of a reduced machine M'', where $M'' \equiv M'$, is a realization of M'). If $|Q'| = m$, let n be any integer such that there is a subset R of $\{0,1\}^{(n)}$ with the following two properties:

$$|R| = m \tag{1.30}$$

and

$$\min\{h(r,s) \,|\, r, s \in R,\ r \neq s\} \geq 2t+1 \tag{1.31}$$

In other words, R is an m-word error correcting code having minimum distance 2t + 1 and therefore exists for any specified m and t [20]. We now want to construct a machine M with state set $Q = \{0,1\}^{(n)}$ having the desired properties. To do this, let $K_t(q)$ denote the solid sphere of radius t and center q, i.e.

$$K_t(q) = \{r \,|\, h(q,r) \leq t\} \tag{1.32}$$

and let η be any function from Q onto Q' satisfying the following conditions:

η restricted to R is 1-1 and onto, (1.33)

and

For all $r \in R$, $q \in K_t(r) \Longleftrightarrow \eta(q) = \eta(r)$. (1.34)

Note that such a function η always exists for by (1.30), $|R| = |Q'|$ and by (1.31), each $q \in Q$ is contained in at most one solid sphere $K_t(r)$. Consider now a machine $M = (I', Q, 0', \delta, \omega)$ with δ and ω defined as follows for each $q \in Q$, $a \in I'$:

$$\delta(q, a) = \text{the unique } r \text{ such that}$$
$$r \in \eta^{-1}(\delta'(\eta(q), a)) \text{ and } r \in R \quad (1.35)$$

$$\omega(q, a) = \omega'(\eta(q), a) \quad (1.36)$$

Claim 1): M realizes M'.

By (1.35) we have

$$\eta(\delta(q, a)) = \eta(r) = \delta'(\eta(q), a)$$

and this coupled with (1.36) implies that η is a state homomorphism from M to M' (cf. [11], page 21). Thus M not only realizes M' but is in fact a homomorphic realization.

Claim 2): μ is R-masked, for all $\mu \in S_n[t]$.

If $\mu \in S_n[t]$, by Theorem 1.14, it suffices to show that

i) $\delta(\mu(R) \times I) \subseteq R$.

and

ii) $\underset{\mu}{\equiv} |R \cup \mu(R) \subseteq \equiv |R \cup \mu(R)$.

Condition i) holds for by Definition of δ (1.35), $\delta(Q \times I) \subseteq R$. To prove ii), suppose $r, s \in R \cup \mu(R)$ and $r \underset{\mu}{\equiv} s$, i.e. $\mu(r) = \mu(s)$. In case both $r \in \mu(R)$ and $s \in \mu(R)$, since μ is stable, $r = \mu(r) = \mu(s) = s$ and therefore $r \equiv s$. Suppose then that either $r \notin \mu(R)$ or $s \notin \mu(R)$. If $r \notin \mu(R)$ then $r \in R$ and, by Theorem 1.24 (part iv), if $d(\mu) = k$ then

$$h(r, s) \leq k.$$

But $\mu \in S_n[t]$ implies $k \leq t$ and therefore

$$s \in K_t(r).$$

Recalling the definition of η, by (1.34),

$$\eta(r) = \eta(s). \qquad (1.37)$$

By the same argument, (1.37) holds if $s \notin \mu(R)$ and as η is a state homomorphism, $r \equiv s$.

Claim 3): R is a complete subset of Q.

By definition of η, (1.33) implies each equivalence class of $\underset{\eta}{\equiv}$ contains a state $r \in R$. Since η is a state homomorphism and M' is reduced,

$$\underset{\eta}{\equiv} \;=\; \equiv$$

and therefore each equivalence class of $\equiv$ contains a state $r \in R$, i.e. R is complete. As the three claims just proved are those made in the statement of the theorem, the proof is complete.

Not only does Theorem 1.25 establish the existence of fault tolerant realizations but the construction used in the proof provides the following synthesis algorithm for fault tolerant sequential networks.

Algorithm 1.1

Given some finite-state-realizable behavior B, to design a sequential network N that tolerates, for some specified integer t, all combinations of t or fewer stuck-at faults at flip-flop outputs:

1) Determine a reduced machine $M' = (I', Q', 0', \delta', \omega')$ such that $B_{M'} = B$.

2) Let $m = |Q'|$ and determine an m-word, n-bit error correcting code R having minimum distance $2t + 1$.

3) Determine a function $\eta: \{0, 1\}^{(n)} \rightarrow Q'$ such that

   i) $\eta|R$ is 1-1 and onto,

   ii) for all $r \in R$, $q \in K_t(r) \Longrightarrow \eta(q) = \eta(r)$.

4) Design an n-dimensional state-assigned machine $M = (I', Q, 0', \delta, \omega)$ where

   i) $\delta(q, a)$ = the unique code word in
   $$\eta^{-1}(\delta'(\eta(q), a)),$$

   ii) $\omega(q, a) = \omega'(\eta(q), a)$.

5) Design a sequential network N that is represented by the state-assigned machine M (cf. Example 1.3).

Note that certain choices can be made in the algorithm which influence the complexity of the resulting network. In particular, the choice of R in step 2), and the choice of η in step 3) comprise the "state assignment problem" that remains after the R-masking constraint is imposed. The following example should help to illustrate the algorithm just described.

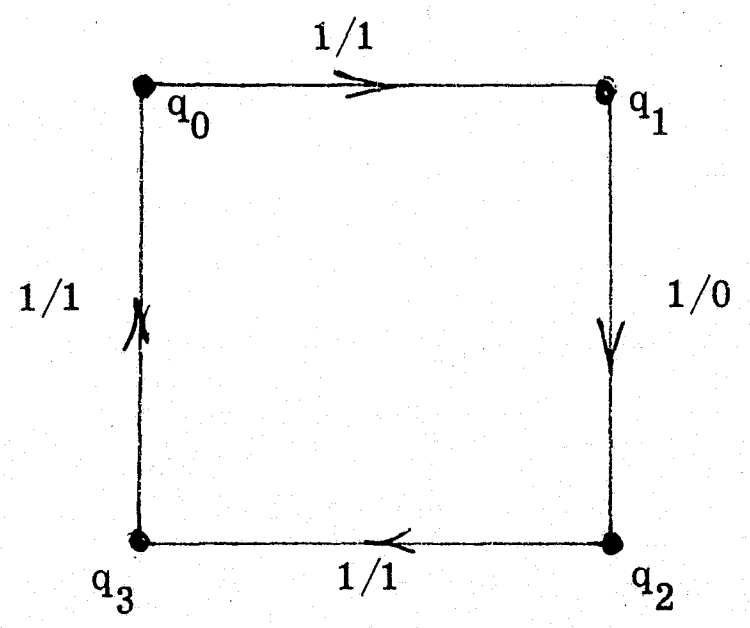
Example 1.5

Suppose that the required behavior is the periodic generation of the sequence 1011 and we wish to design a fault tolerant realization that will R-masked all single stuck-at faults (i.e. t = 1). Beginning with step 1) of Algorithm 1.1:

1) A reduced (autonomous) machine

$$M' = (\{1\},\ \{q_0, q_1, q_2, q_3\},\ \{0, 1\},\ \delta',\ \omega')$$

that realizes the specified behavior is given by



2) Since m = 4 and t = 1, we must find a 4-word, n-bit error correcting code R having minimum distance 3. A 5-bit code having these properties is given by (we let strings denote 5-tuples, e.g. 01101 denotes (0, 1, 1, 0, 1)):

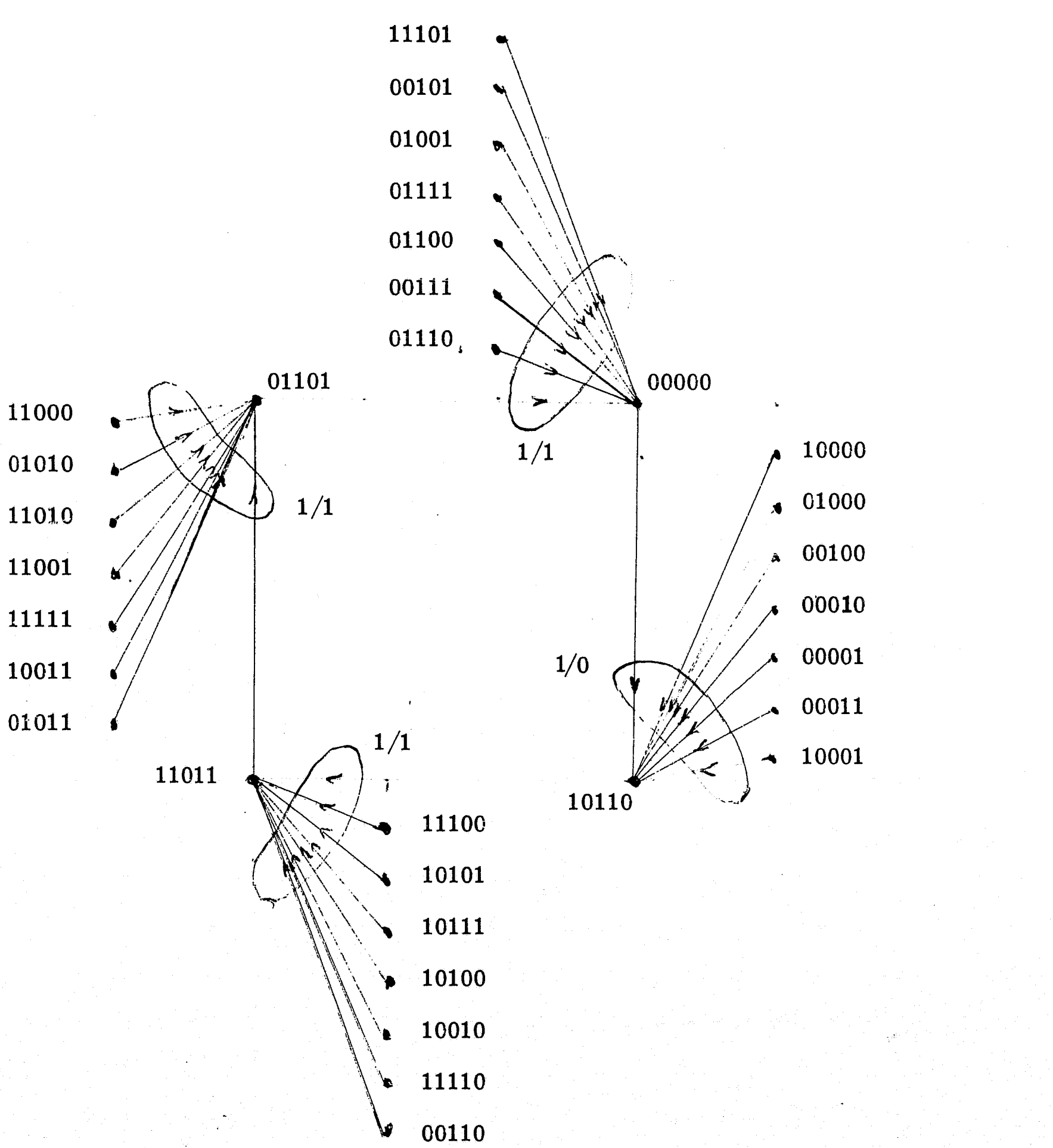
$$R = \{01101, 00000, 10110, 11011\}.$$

3) Let η: $\{0,1\}^{(5)} \longrightarrow \{q_0, q_1, q_2, q_3\}$ where

$$\eta(q) = \begin{cases} q_0 \text{ if } q \in K_1(01101) \cup \{00111, 01110\} \\ q_1 \text{ if } q \in K_1(00000) \cup \{00011, 10001\} \\ q_2 \text{ if } q \in K_1(10110) \cup \{10101, 11100\} \\ q_3 \text{ if } q \in K_1(11011) \cup \{01010, 11000\}. \end{cases}$$

Thus, by inspection, η satisfies conditions i) and ii) of step 3).

4) The machine $M = (\{1\}, \{0,1\}^{(5)}, \{0,1\}, \delta, \omega)$ designed according to step 4) has the following transition graph.



Machine M (Step 4), Example 1.5)

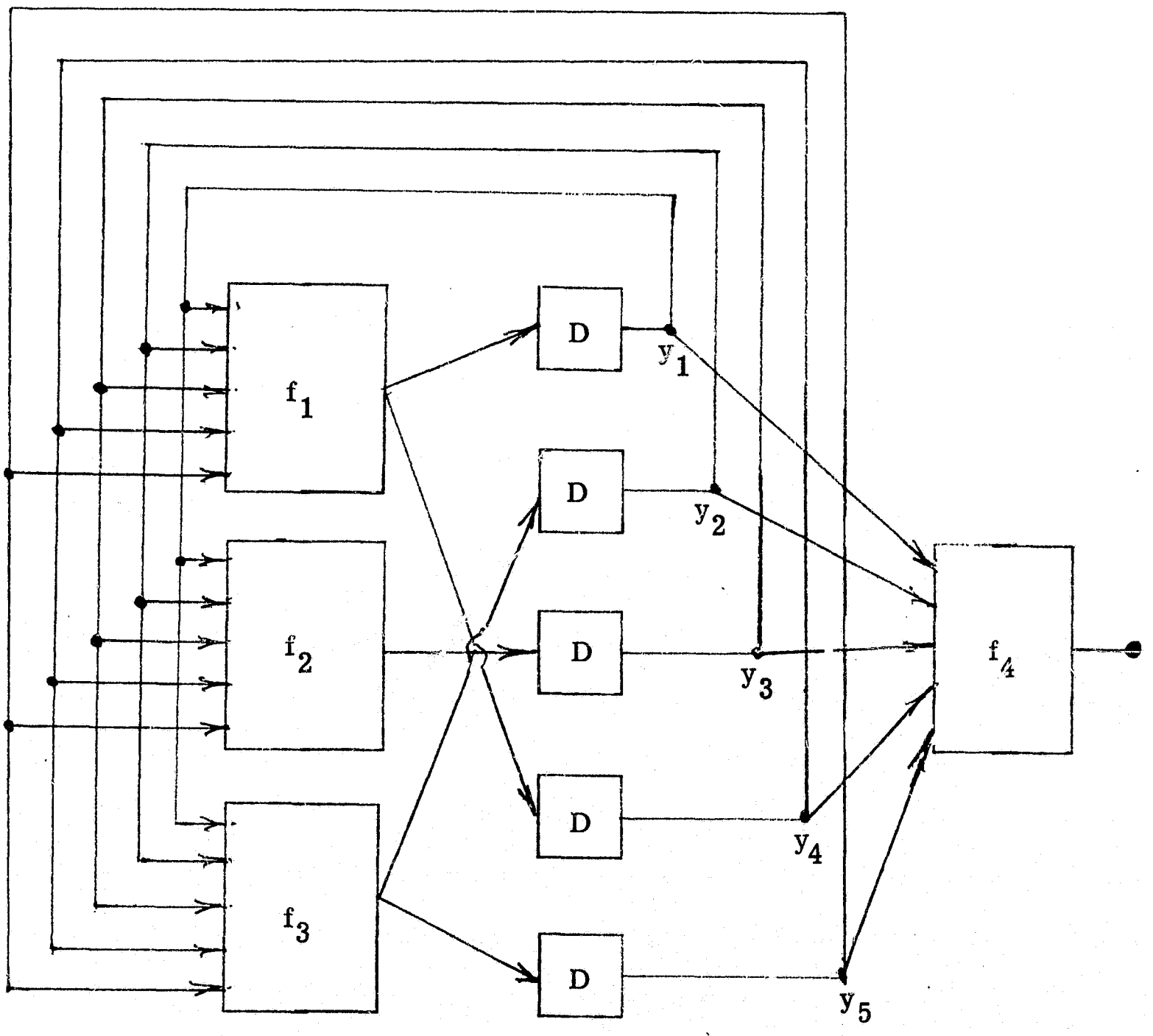
5) A sequential network N which corresponds to machine M (and therefore R-masks all single stuck-at 0 or stuck-at 1 faults at the delay outputs) is shown in schematic form on the following page. The Boolean forms F_i representing switching functions f_i $(1 \leq i \leq 4)$ are as follows:

$$F_1 = \overline{y}_1\,\overline{y}_3(\overline{y}_2 + \overline{y}_4\,\overline{y}_5) + \overline{y}_2\,\overline{y}_5 + y_1(y_3\,\overline{y}_5 + \overline{y}_2(y_3 + \overline{y}_4))$$

$$F_2 = \overline{y}_1(\overline{y}_2(\overline{y}_3 + \overline{y}_4\,\overline{y}_5) + \overline{y}_3(y_4 + \overline{y}_5)) + y_1(y_2(\overline{y}_3 + y_4\,y_5) + \overline{y}_3(\overline{y}_4 + y_5))$$

$$F_3 = y_2\,\overline{y}_3(y_1 + y_4) + \overline{y}_2\,y_3(y_1 + y_4\,\overline{y}_5) + y_1(y_2\,\overline{y}_5 + y_4)$$

$$F_4 = y_1(y_2 + y_3 + y_4) + (y_2 + y_3)\,(y_4 + y_5)$$



Network N (Step 5), Example 1.5)

Thus, using Algorithm 1.1, we observe that a state-assigned machine of dimension 5 (alternatively a network with 5 delay flip-flops) suffices to realize a 4 state machine and yet tolerate all degree 1 stuck-at faults. Comparing this realization with one obtained by realizing M' with a network N' and then triplicating N' and voting, the latter scheme requires 6 flip-flops and hence is less "efficient" in the sense that more memory is required for the same fault tolerance. A more general comparison of this type is the subject of the section that follows.

Redundancy of t-Tolerant Realizations

If B is some sequential behavior (i.e., $B = B_{M'}$ for some sequential machine M') let us say that a state-assigned machine M is a t-tolerant realization of B if there is a subset R of Q such that each $\beta \in B$ is realized by some $r \in R$ and μ is R-masked for all μ in $S_n[t]$. Note that R in this case need not be a complete subset since B_M may properly contain the set $\{\beta_r | r \in R\}$. Thus, for example, any realization obtained using Algorithm 1.1 is t-tolerant. Another type of t-tolerant realization can be obtained by the well known technique or replicating-and-voting, that is, put 2t + 1 copies of M' (where $B_{M'} = B$) in parallel and then take a majority vote of the 2t + 1 outputs.

To compare the efficiency of such realizations, if M is an n-dimensional, t-tolerant realization of B, its (memory) redundancy $\rho(M)$ will be defined as the ratio of the memory capacity of M (in bits) to the memory capacity (in bits) of any reduced machine with behavior B (where $|B| \geq 2$), that is,

$$\rho(M) = \frac{n}{\log_2 |B|} . \tag{1.38}$$

Thus, for example, the machine M (Example 1.5, Step 4) which is a 1-tolerant realization of the behavior of M' has redundancy

$$\rho(M) = \frac{5}{\log_2 4} = 2.5.$$

Relative to some class $\mathcal{M}$ of realizations we can then define a function $\kappa_{\mathcal{M}}$ where, for integers $m \geq 2$ and $t \geq 1$,

$$\kappa_{\mathcal{M}}(m,t) = \min \left\{ \rho(M) \;\middle|\; \begin{array}{l} M \in \mathcal{M} \text{ and, for some B} \\ \text{such that } |B| = m, \text{ M is a} \\ \text{t-tolerant realization of B} \end{array} \right\} . \tag{1.39}$$

In other words, $\kappa_{\mathcal{M}}(m,t)$ is the minimum redundancy of all machines in $\mathcal{M}$ which are t-tolerant realizations of m-state behaviors. Thus comparing two classes $\mathcal{M}$ and $\mathcal{N}$ for given m and t, if $\kappa_{\mathcal{M}}(m,t) < \kappa_{\mathcal{N}}(m,t)$ we can conclude that class $\mathcal{M}$ is preferable to class $\mathcal{N}$ for the t-tolerant realization of m-state behavior.

Let us now compare the two specific realization types mentioned above and denote them $\mathcal{A}$ and $\mathcal{B}$, that is,

$\mathcal{A}$ = set of all realizations obtained using Algorithm 1.1

$\mathcal{B}$ = set of all replicate-and-vote realizations.

Then a lower bound for $\kappa_{\mathcal{A}}(m, t)$ can be immediately obtained from the following lemma.

Lemma 1.5

If B is some sequential behavior with $|B| = m$ and $M \in \mathcal{A}$ is a t-tolerant realization of B having dimension n then n must satisfy the inequality

$$n \geq \log_2 \left[m\left(\sum_{k=0}^{t} \binom{n}{k} \right) \right] .$$

Proof

If $|B| = m$ then the reduced machine M' (Algorithm 1.1, Step 1) has m states. Accordingly, $Q = \{0, 1\}^{(n)}$ must contain an m-word error correcting code having minimum distance $2t + 1$ (Steps 2) and 3)). Thus Q contains m disjoint solid spheres of radius t where each sphere $K_t[q]$ contains

$$\sum_{k=0}^{t} \binom{n}{k}$$

elements. Thus Q must have at least

$$m\left(\sum_{k=0}^{t}\binom{n}{k}\right)$$

elements from which the inequality of the lemma (sometimes called the Hamming bound) follows.

Consequently, a lower bound for $\kappa_{\mathcal{A}}(m,t)$ is given by

Theorem 1.26

$$\kappa_{\mathcal{A}}(m,t) \geq \frac{n}{\log_2 m}$$

where n is the least integer satisfying

$$n \geq \log_2\left[m\left(\sum_{k=0}^{t}\binom{n}{k}\right)\right].$$

Proof

Lemma 1.5 and the definition of $\kappa_{\mathcal{A}}(m,t)$ (1.39).

To obtain a lower bound on the dimension for which a class $\mathcal{A}$ realization will always exist (and thereby an upper bound on $\kappa_{\mathcal{A}}(m,t)$) we have

Lemma 1.6

If B is some sequential behavior with $|B| = m$ then there exists a t-tolerant realization M of B such that $M \in \mathcal{A}$ and the dimension n of M satisfies the inequality

$$n \geq \lceil \log_2 m \rceil + \log_2\left(1 + \sum_{k=0}^{2t-1}\binom{n-1}{k}\right).$$

Proof

By the Varsharmov-Gilbert Bound [26], there exists an n-bit code of minimum distance d having r parity check bits if

$$2^r \geq 1 + \sum_{k=0}^{d-2} \binom{n-1}{k},$$

or equivalently,

$$r \geq \log_2 \left(1 + \sum_{k=0}^{d-2} \binom{n-1}{k}\right).$$

Since a distance of $d = 2t + 1$ suffices for tolerance t and r check bits give n-r information bits,

$$n \geq (n-r) + \log_2 \left(1 + \sum_{k=0}^{2t-1} \binom{n-1}{k}\right).$$

As $\lceil \log_2 m \rceil$ information bits are enough to code m states, by Algorithm 1.1, we obtain the desired result.

Accordingly, the minimum redundancy for $\mathcal{A}$ realizations has the following upper bound.

Theorem 1.27

$$\kappa_{\mathcal{A}}(m,t) \leq \frac{n}{\log_2 m}$$

where n is the least integer satisfying

$$n \geq \lceil \log_2 m \rceil + \log_2 \left(1 + \sum_{k=0}^{2t-1} \binom{n-1}{k}\right).$$

Proof

Lemma 1.6 and the definition of $\kappa_{\mathcal{A}}(m,t)$ (1.39).

In the case of $\mathcal{B}$ realizations, $\kappa_{\mathcal{B}}(m,t)$ is more easily expressed for if $|B| = m$ then the minimum dimension of a state-assigned machine M' realizing B is $\lceil \log_2 m \rceil$. Since a t-tolerant realization M is obtained by replicating M' $2t + 1$ times, $n = \dim M = \lceil \log_2 m \rceil (2t + 1)$. Hence, by Definition 1.39,

$$\kappa_{\mathcal{B}}(m,t) = \frac{\lceil \log_2 m \rceil (2t + 1)}{\log_2 m} . \qquad (1.40)$$

$$\cong 2t + 1 \text{ for } m >> 1$$

$$= 2t + 1 \text{ for } m = 2^s.$$

Thus the minimum redundancy of class $\mathcal{B}$ realizations is essentially independent of behavior size. On the other hand the minimum redundancy of class $\mathcal{A}$ realizations decreases with increasing m. Indeed, by the upper bound (Theorem 1.27) if t is fixed then

$$\lim_{m \to \infty} \kappa_{\mathcal{A}}(m,t) = 1.$$

A more revealing comparison of the two types of realizations is given by Table 1.1 and Figures 1.1 - 1.3.

It is also of interest to consider the function $\kappa = \kappa_{\mathcal{M}}$ where $\mathcal{M}$ is unrestricted (i.e., $\mathcal{M}$ is the class of all state-assigned machines)

m \ t	1			2			3		
	$K_B(m,t)$	$K_A(m,t)$ u.b.	$K_A(m,t)$ l.b.	$K_B(m,t)$	$K_A(m,t)$ u.b.	$K_A(m,t)$ l.b.	$K_B(m,t)$	$K_A(m,t)$ u.b.	$K_A(m,t)$ l.b.
2^1	3	3	3	5	5	5	7	7	7
2^2	3	2.5	2.5	5	4.5	3.5	7	6	5
2^3	3	2	2	5	3.67	3	7	5	3.67
2^5	3	1.8	1.8	5	2.8	2.4	7	3.8	2.8
2^{10}	3	1.4	1.4	5	2.1	1.8	7	2.7	2.1
2^{20}	3	1.25	1.25	5	1.65	1.45	7	2	1.65
2^{50}	3	1.12	1.12	5	1.32	1.22	7	1.5	1.32
2^{100}	3	1.07	1.07	5	1.19	1.13	7	1.29	1.19
2^{1000}	3	1.01	1.01	5	1.03	1.02	7	1.04	1.03

Table 1.1 Some values of $\kappa_B(m,t)$ compared with the upper bound (u.b.) and lower bound (l.b.) on $\kappa_a(m,t)$.

SEMI-LOGARITHMIC 46 5810
3 CYCLES X 140 DIVISIONS
KEUFFEL & ESSER CO.

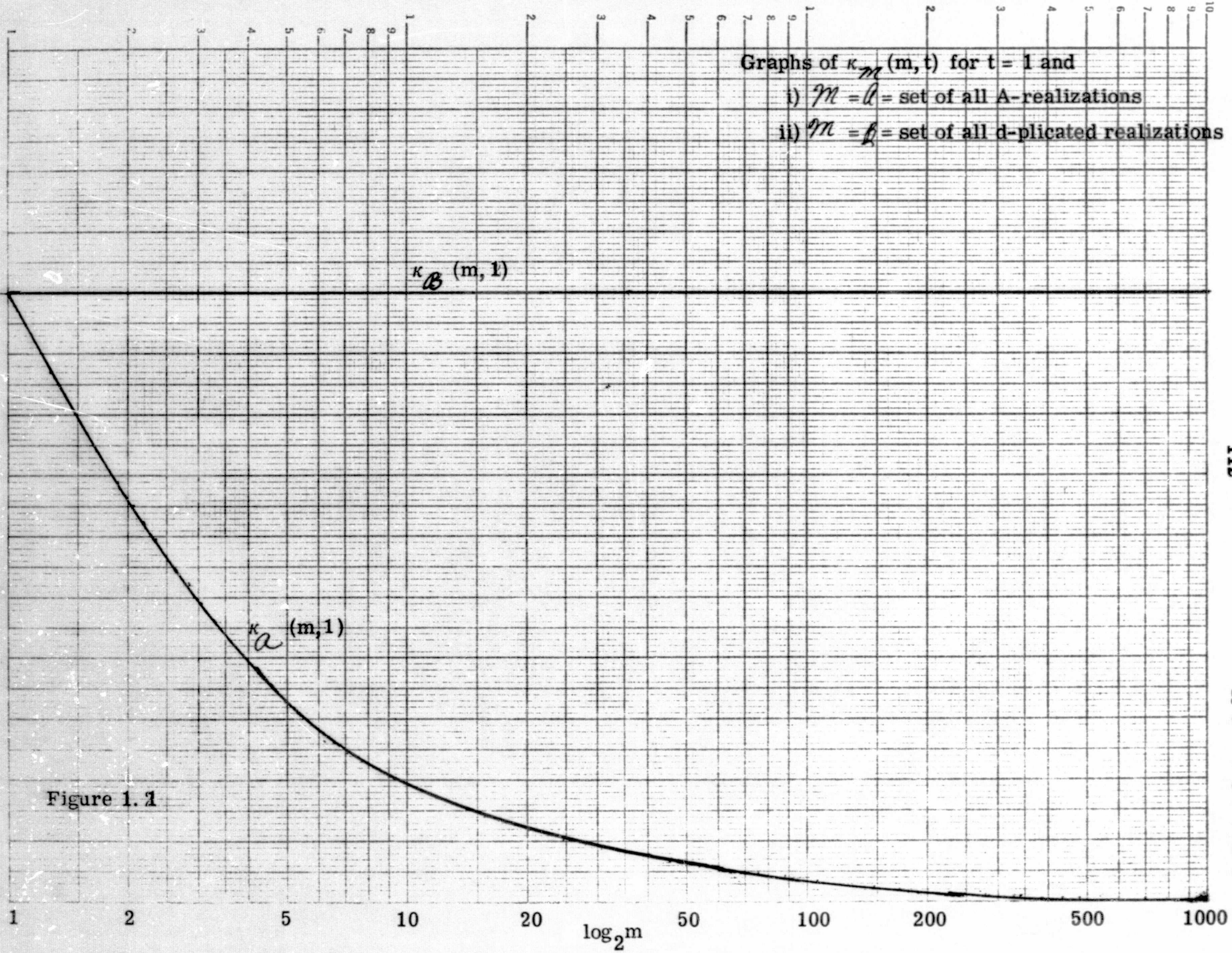


Figure 1.1

RO 70-456

SEMI-LOGARITHMIC 46 5810
3 CYCLES X 140 DIVISIONS
KEUFFEL & ESSER CO.

Graphs of $\kappa_{\mathcal{M}}(m, t)$ for t = 2 and

i) $\mathcal{M} = \mathcal{A}$ = set of all A-realizations

ii) $\mathcal{M} = \mathcal{B}$ = set of all d-plicated realizations

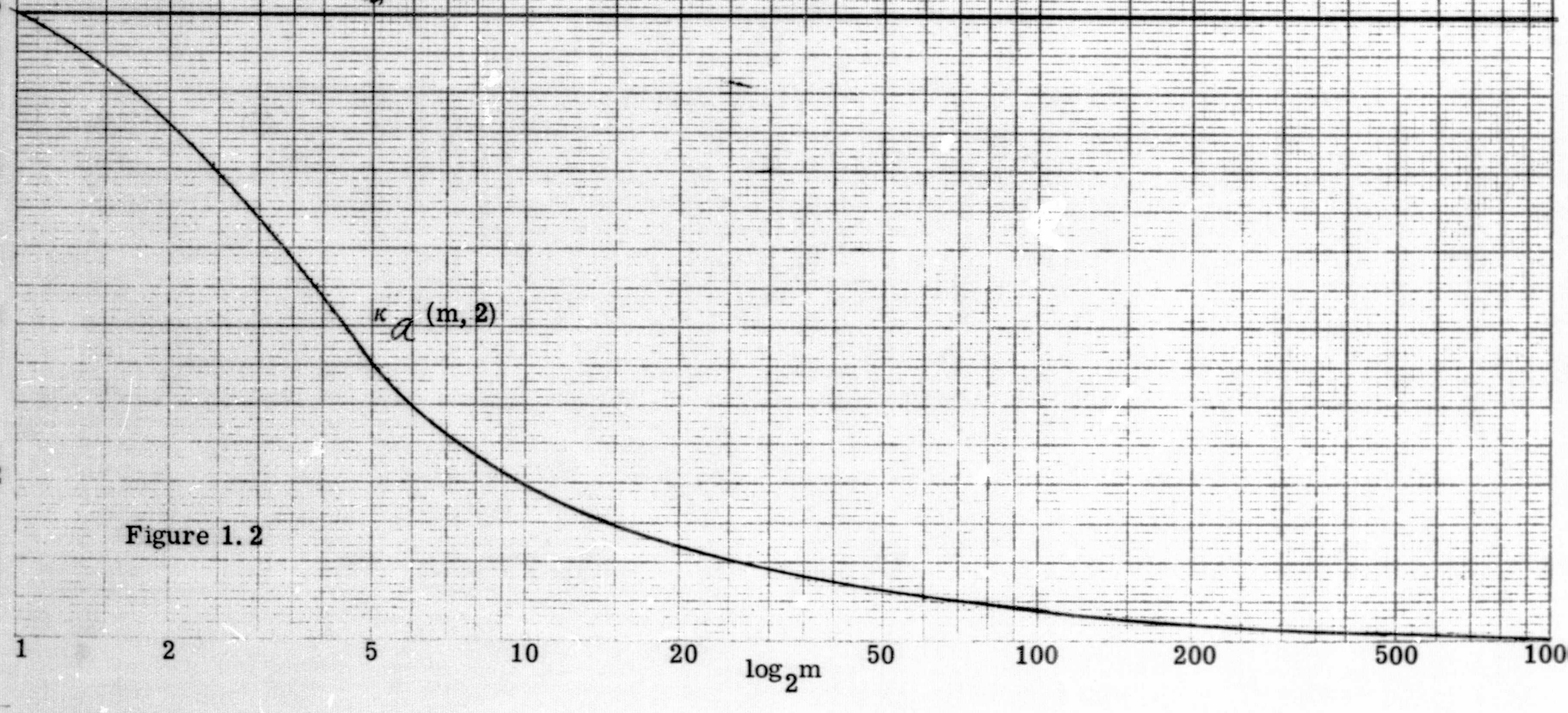
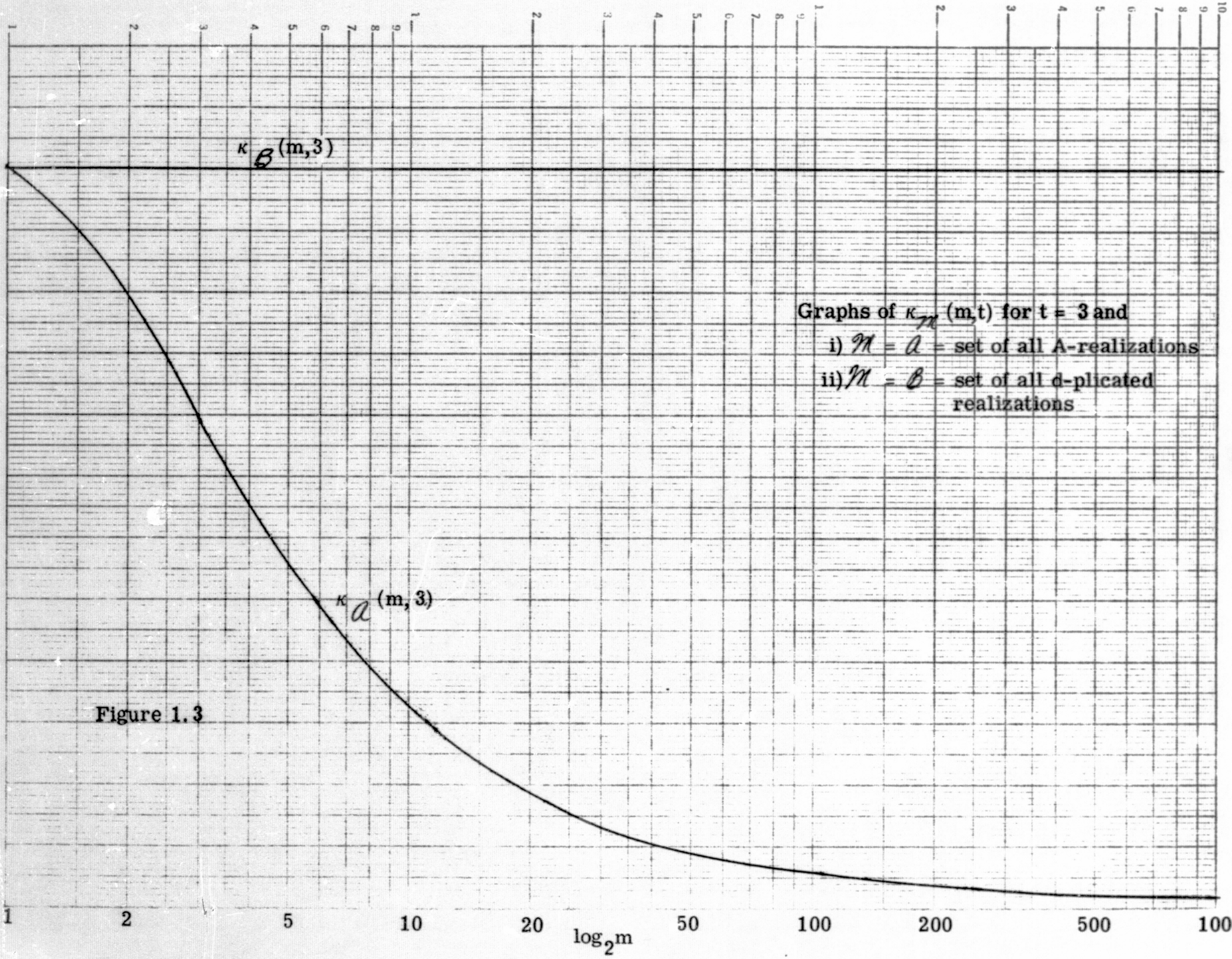


Figure 1.2

K&E SEMI-LOGARITHMIC 46 5810
3 CYCLES X 140 DIVISIONS
KEUFFEL & ESSER CO.



Graphs of $\kappa_{\mathcal{M}}(m,t)$ for $t = 3$ and

i) $\mathcal{M} = \mathcal{A}$ = set of all A-realizations

ii) $\mathcal{M} = \mathcal{B}$ = set of all d-plicated realizations

Figure 1.3

RO 70-456

since the value $\kappa(m,t)$ will be the least redundancy possible for a t-tolerant realization of m-state behavior. A lower bound for $\kappa(m,t)$ is given by the following theorem.

Theorem 1.28

$$\kappa(m,t) \geq 1 + \frac{t}{\log_2 m}$$

Proof

If M is an n-dimensional, t-tolerant realization of B where $|B| = m$ then, by definition, there is a subset R of Q such that i) the states in R realize B and ii) μ is R-masked for all μ in $S_n[t]$.

i) implies

$$|\{\beta_r | r \in R\}| \geq |B| = m \qquad (1.41)$$

and, if μ is a stuck-at fault of degree t, ii) implies μ is R-masked, i.e., by Definition 1.13,

$$\{\beta^{\mu}_{\mu(r)} | r \in R\} = \{\beta_r | r \in R\}. \qquad (1.42)$$

Since the state set $Q^{\mu} = \mathcal{R}(\mu)$ of the faulty machine M^{μ} contains $\{\mu(r) | r \in R\}$,

$$|\mathcal{R}(\mu)| \geq |\{\beta^{\mu}_{\mu(r)} | r \in R\}| \qquad (1.43)$$

and combining (1.41), (1.42) and (1.43) we have

$$|\mathcal{R}(\mu)| \geq m.$$

But $d(\mu) = t$ and by Corollary 1.18.1 the subcube $\mathcal{R}(\mu)$ has dimension n-t, that is,

$$|\mathcal{R}(\mu)| = 2^{n-t}$$

Thus $2^{n-t} \geq m$ which implies $n \geq \log_2 m + t$ and dividing by $\log_2 m$ we have

$$\kappa(m, t) = \frac{n}{\log_2 m} \geq 1 + \frac{t}{\log_2 m}$$

which concludes the proof.

Comparing the functions κ_a and κ, for large m we see that class a realizations have a minimum redundancy that approaches the best possible, e.g., $\kappa_a(2^{100}, 1) = 1.07$ as compared with $\kappa(2^{100}, 1) = 1.01$. In network terms, the bound given by Theorem 1.28 says that a t-tolerant realization of a behavior B requires at least t more flip-flops than a nonredundant realization of B. For equality to hold, i.e.,

$$\kappa(m, t) = 1 + \frac{t}{\log_2 m}$$

it must be the case that for some B, $m = |B|$ is a power of 2 and there exists a t-tolerant realization of B with exactly t redundant flip-flops. Moreover, the machine M representing this network realization must be such that, for all faults μ of degree t, the result M^{μ} is isomorphic to a reduced machine with behavior B. For these reasons, it should be possible to improve the lower bound on $\kappa(m, t)$ over that given by Theorem 1.28.

We note also that the bounds discussed here apply to the set of all sequential behaviors B requiring a given number of states m in their reduced realizations. This does not mean to imply, however, that all m-state behaviors B have minimum t-tolerant realizations of the same dimension. Indeed, we suspect just the opposite, namely for given m and t, there always exist behaviors B and B' such that $|B| = |B'| = m$ and yet the dimension of the least redundant t-tolerant realization of B differs from that of B'.

Initial State Fault Masking

Initial state fault masking, or q_0-masking (cf. Theorem 1.7) is a special case of R-masking ($R - \{q_0\}$) and is of interest since in many applications only the behavior of some initial or "reset" state need be preserved. To mask stuck-at faults from an initial state we begin by specifying the fault set. As earlier (1.29) let $S_n[t]$ denote the set of all faults in S_n of degree $\leq t$. Then the following theorem provides an upper bound on the degree of faults which are q_0-masked.

Theorem 1.29

If M is a state-assigned machine of dimension n, $q_0 \in Q$, the number of nonequivalent states reachable from q_0 is p, and μ is q_0-masked, for all $\mu \in S_n[t]$, then

$$t \leq n - \log_2 p.$$

Proof

If $\mu \in S_n[t]$ and μ is q_0-masked, then $\beta^{\mu}_{\mu(q_0)} = \beta_{q_0}$ which implies that the number of nonequivalent states reachable from $\mu(q_0)$ in M^{μ} must equal p. Hence

$$|\mu(Q)| = |\mathcal{R}(\mu)| \geq p$$

and $d(\mu) \leq t$,

$$2^{n-t} \geq p$$

(cf. Corollary 1.18.2). Thus

$$\log_2 p \leq n-t$$

that is

$$t \leq n - \log_2 p$$

completing the proof.

Having established an upper bound on t we want to analyze the machines M^{μ}, $\mu \in S_n[t]$, to determine the structure of transition functions which q_0-mask faults in $S_n[t]$. An understanding of this structure will permit the derivation of transition functions for M given the behavior to be realized. To begin, let U_{μ} denote the relation μ_{q_0} defined earlier (1.11), that is

$$U_{\mu} = \{(\bar{\delta}(q_0, x), \bar{\delta}^{\mu}(\mu(q_0), x)) \mid x \in I^*\}.$$

The importance of this relation arises from Theorem 1.7, that is, μ is q_0-masked iff

$$\bar{\delta}(q_0, x) \equiv_1 \bar{\delta}^{\mu}(\mu(q_0), x), \text{ for all } x \in I^*.$$

For all $q \in Q_0$, where

$$Q_0 = \{q \mid \bar{\delta}(q_0, x) = q, x \in I^*\}$$

let $[q]_\mu$ denote the set of right relatives of q in U_μ, that is

$$[q]_\mu = \{q' \mid q \; U_\mu \; q'\}. \tag{1.44}$$

This set contains all states reachable from $\mu(q_0)$ in M^μ under an input sequence that takes M from q_0 to q.

Lemma 1.7

For $q \in Q_0$, μ is q_0-masked iff

$q' \in [q]_\mu \Longrightarrow \beta^\mu_{q'} = \beta_q$.

Proof: Necessity:

Suppose $q' \in [q]_\mu$, then there exists $x \in I^*$ such that

$$q = \bar{\delta}(q_0, x)$$

$$q' = \bar{\delta}^{\mu}(\mu(q_0), x).$$

But μ is q_0-masked which says

$$\beta^\mu_{\mu(q_0)} = \beta_{q_0}$$

and implies

$$\beta^\mu_{\bar{\delta}^{\mu}(\mu(q_0), y)} = \beta_{\bar{\delta}(q_0, y)}, \quad \forall\, y \in I^*.$$

In particular, when y=x we have

$$\beta^\mu_{q'} = \beta_q.$$

Sufficiency:

Let $q = q_0$, then

$$\mu(q_0) \in [q_0]_\mu \Longrightarrow \beta^{\mu}_{\mu(q_0)} = \beta_{q_0}$$

and μ is q_0-masked by definition. This concludes the proof.

Clearly, all the states in $[q]_\mu$ are equivalent in M^μ if μ is q_0-masked.

To clarify the preceding discussion consider the following example.

Example 1.6

For the periodic output sequence 0101... let M be an autonomous state-assigned Mealy machine of dimension 3 which realizes this behavior from $q_0 = 000$.

M:

I / Q	$\delta(q, a) / \omega(q, a)$
000	101/0
001	010/1
010	111/0
011	000/1
100	010/1
101	000/1
110	111/0
111	101/1

Let μ be the fault denoted in cubical notation by

$$\mu = 0xx$$

($\mu \in S_3(1)$), that is

q	μ(q)
000	000
001	001
010	010
011	011
100	000
101	001
110	010
111	011

The machine M under the fault μ has the transition table

M^{μ}:

Q \ I	$\delta^{\mu}(q, a)/\omega^{\mu}(q, a)$
000	001/0
001	010/1
010	011/0
011	000/1

Using decimal equivalents of the binary 3-tuples to denote states and beginning with $q_0 = 0$, the sequence of states is 0, 5, 0, 5, ... in M and 0, 1, 2, 3, 0, 1, 2, 3, ... in M^{μ}, so

$$U_{\mu} = \{(0, 0),\ (5, 1),\ (0, 2),\ (5, 3)\}.$$

The partition of the output equivalence relation $\equiv_1$ is:

$$\{\overline{0, 2, 6};\ \overline{1, 3, 4, 5, 7}\}$$

and since $U_\mu \subseteq \equiv_1$, μ is q_0-masked. The sets of right relatives are

$$[0]_\mu = \{0, 2\}; \quad [5]_\mu = \{1, 3\}$$

and the state equivalences $0 \underset{M^\mu}{\equiv} 2$ and $1 \underset{M^\mu}{\equiv} 3$ are easily verified.

Suppose now that M is a state-assigned machine of dimension n and let U_t denote the union of the relations U_μ, $\mu \in S_n[t]$, i.e.

$$U_t = \bigcup_{\mu \in S_n[t]} U_\mu . \tag{1.45}$$

Then by Theorem 1.7 it is immediate that

Theorem 1.30

If M is a Mealy machine then

μ is q_0-masked, $\forall\, \mu \in S_n[t]$ iff $U_t \subseteq \equiv_1$.

In general U_t is neither symmetric nor transitive. However, if we let E_t denote the symmetric, transitive closure of U_t, i.e.

$E_t = \{(q, q') |$ there exist $q_1 = q, q_2, \ldots, q_m = q'$ such that

$q_i U_t q_{i+1}$ or $q_{i+1} U_t q_i$, $i=1, 2, \ldots, m-1\}$

we obtain

Corollary 1.30.1

If M is a Mealy machine then

μ is q_0-masked, $\forall\, \mu \in S_n[t]$ iff $E_t \subseteq \equiv_1$.

Proof: Necessity:

E_t is the smallest equivalence relation containing U_t. Since $\equiv_1$ is also an equivalence relation containing U_t, it must contain E_t.

Sufficiency: Obvious.

According to the corollary there is some flexibility in the choice of behavior for M as long as the partition defined by E_t refines the output partition.

In conjunction with the relation U_t we introduce [q], where

$$[q] = \bigcup_{\mu \in S_n[t]} [q]_\mu = \{q' | q\ U_t\ q'\}. \qquad (1.46)$$

Note that $q \in [q]$ since ι—the identity function on Q—is in $S_n[t]$. Each state of [q] is a state of one or more faulty machines, M^μ, $\mu \in S_n[t]$.

The following example should help to clarify the definitions of U_t, E_t, and [q] and Theorem 1.30.

Example 1.7

For the periodic output sequence 0101 ... let M be a 3 dimensional state-assigned Mealy machine which realizes this sequence beginning with $q_0 = 0$. Let the set of faults be $S_3(1)$ consisting of

$\mu_1 = xx1$ $\quad$ $\mu_2 = xx0$

$\mu_3 = x1x$ $\quad$ $\mu_4 = x0x$

$\mu_5 = 1xx$ $\quad$ $\mu_6 = 0xx$.

The transition function for M and the faulty machines M^{μ} is given by the following table; the entry in row q is $\delta(q,a)/\omega(q,a)$ for M and $\delta^{\mu}(q,a)/\omega^{\mu}(q,a)$ for M^{μ} (there is no entry if $q \notin Q^{\mu}$).

Q \ I	δ	δ^{μ_1}	δ^{μ_2}	δ^{μ_3}	δ^{μ_4}	δ^{μ_5}	δ^{μ_6}
0	7/0		6/0		5/0		3/0
1	3/0	3/0			1/0		3/0
2	7/0		6/0	7/0			3/0
3	1/0	1/0		3/0			1/0
4	6/0		6/0		4/0	6/0	
5	4/1	5/1			4/1	4/1	
6	0/1		0/1	2/1		4/1	
7	0/1	1/1		2/1		4/1	

The state and output sequences for each machine M^{μ} beginning in state $\mu(q_0)$ are:

	states	outputs
M^{μ_1}	1, 3, 1, 3, 1, ...	00000 ...
M^{μ_2}	0, 6, 0, 6, 0, ...	01010 ...
M^{μ_3}	2, 7, 2, 7, 2, ...	01010 ...
M^{μ_4}	0, 5, 4, 4, 4, ...	01000 ...
M^{μ_5}	4, 6, 4, 6, 4, ...	01010 ...
M^{μ_6}	0, 3, 1, 3, 1, ...	00000 ...

Clearly the faults μ_1, μ_4 and μ_6 are not q_0-masked. From the state sequences we obtain

$$U_1 = \{(0,0),(0,1),(0,2),(0,4),(7,3),(7,4),(7,5),(7,6),(7,7)\}.$$

The output equivalence relation is

$$\equiv_1 = \{\overline{0,1,2,3,4};\ \overline{5,6,7}\}$$

and Theorem 1.30 is illustrated because $U_1 \not\subseteq \equiv_1$ and not all faults in $S_3[1]$ are q_0-masked. The sets of right relatives are

$$[0]_{\mu_1} = \{1\}, \quad [0]_{\mu_2} = \{0\}, \quad [0]_{\mu_3} = \{2\}$$
$$[0]_{\mu_4} = \{0,4\}, \quad [0]_{\mu_5} = \{4\}, \quad [0]_{\mu_6} = \{0,1\}$$
$$[7]_{\mu_1} = \{3\}, \quad [7]_{\mu_2} = \{6\}, \quad [7]_{\mu_3} = \{7\}$$
$$[7]_{\mu_4} = \{4,5\}, \quad [7]_{\mu_5} = \{6\}, \quad [7]_{\mu_6} = \{3\}$$

From these we see that, for example, μ_1 is not q_0-masked because the condition of Lemma 1.7.

$$3 \in [7]_{\mu_1} \Longrightarrow \beta_3^{\mu_1} = \beta_7$$

is violated by $\omega(3,a) \neq \omega(7,a)$.

Now let M' be a Mealy machine with the same transition function as M and any output assignment such that all $\mu \in S_3[1]$ are q_0-masked. (At least two assignments exist: $\omega(q,a) = 0$ and $\omega(q,a) = 1$ for all states in Q.) Since

$$E_1 = \{\overline{1,2,3,4,5,6,7}\}$$

and $E_1 \subseteq \equiv_1$ (Corollary 1.30.1), M' cannot realize any non-trivial behavior. In other words, there is no output assignment such that machine M' with that output assignment and the transition function

of M can realize a nontrivial behavior and q_0-mask all $\mu \in S_3[1]$.

To correct the deficiencies of this machine consider the following example.

Example 1.8

Let M be the same machine as the previous example with two exceptions:

1) $\delta(5, a) = 2$

2) $\omega(3, a) = 1$

These changes yield

$$U_1 = \{(0, 0), (0, 1), (0, 2), (0, 4), (7, 3), (7, 5), (7, 6), (7, 7)\}$$

and

$$\equiv_1 = \{\overline{0, 1, 2, 4};\ \overline{3, 5, 6, 7}\}.$$

$U_1 \subseteq \equiv_1$ and all faults in $S_3[1]$ are q_0-masked, a result which is in agreement with Theorem 1.30.

A machine M' with the output assignment

$$\omega(0, a) = \omega(1, a) = \omega(2, a) = \omega(4, a) = 1$$

$$\omega(3, a) = \omega(5, a) = \omega(6, a) = \omega(7, a) = 0$$

and the transition function of M realizes the periodic sequence 1010... such that μ is q_0-masked for all $\mu \in S_3[1]$. From

$$E_1 = \{\overline{0, 1, 2, 4};\ \overline{3, 5, 6, 7}\}$$

and Corollary 1.30.1 we see that this sequence is the only other non-trivial behavior realizable with the transition function of M such that all $\mu \in S_3[1]$ are q_0-masked.

The fact that E_t must be a subset of $\equiv_1$ implies that many state assignments for M will not lead to q_0-masking all faults in $S_n[t]$ because a suitable output assignment cannot be found. Some information to narrow the choice for state assignment can be obtained by considering state equivalence and the set of machines M^{μ}, $\mu \in S_n[t]$. We begin by defining an equivalence relation on $\{[q]_{\mu} \mid q \in Q_0\}$ and another on $\{[q] \mid q \in Q_0\}$.

Definition 1.19

For $q_1, q_2 \in Q_0$, $\mu \in S_n[t]$,

i) $[q_1]_{\mu} \tilde{\mu} [q_2]_{\mu}$ if $\forall\, q'_1 \in [q_1]_{\mu}$, $q'_2 \in [q_2]_{\mu}$, $q'_1 \underset{M^{\mu}}{\equiv} q'_2$;

ii) $[q_1] \sim [q_2]$ if for all $\mu \in S_n[t]$, $[q_1]_{\mu} \tilde{\mu} [q_2]_{\mu}$.

Theorem 1.31

If M is a state assigned Mealy machine with μ q_0-masked, for all $\mu \in S_n[t]$, then for all $q_1, q_2 \in Q_0$

$\exists\, \mu \in S_n[t]$ such that $[q_1]_{\mu} \tilde{\mu} [q_2]_{\mu} \Longrightarrow [q_1] \sim [q_2]$.

Proof

If $[q_1]_{\mu} \tilde{\mu} [q_2]_{\mu}$ and $q'_1 \in [q_1]_{\mu}$, $q'_2 \in [q_2]_{\mu}$ then, by definition,

$$q'_1 \underset{M^{\mu}}{\equiv} q'_2$$

and so

$$\beta^{\mu}_{q'_1} = \beta^{\mu}_{q'_2} .$$

By Lemma 1.7, since μ is q_0-masked

$$\beta^{\mu}_{q'_1} = \beta_{q_1} \text{ and } \beta^{\mu}_{q'_2} = \beta_{q_2}$$

so that

$$\beta_{q_1} = \beta_{q_2} .$$

Now let $\gamma \in S_n[t]$ with $q''_1 \in [q_1]_\gamma$, $q''_2 \in [q_2]_\gamma$; since γ is q_0-masked,

$$\beta^{\gamma}_{q''_1} = \beta_{q_1} = \beta_{q_2} = \beta^{\gamma}_{q''_2}$$

so that

$$q''_1 \underset{M^\gamma}{\equiv} q''_2 .$$

But q''_1 and q''_2 are any states in $[q_1]_\gamma$ and $[q_2]_\gamma$ respectively, so that

$$[q_1]_\gamma \tilde{\gamma} [q_2]_\gamma$$

and since γ is any fault in $S_n[t]$, the result holds for all faults in $S_n[t]$, i.e.

$$[q_1] \sim [q_2]$$

which concludes the proof.

Corollary 1.31.1

If μ is q_0-masked, for all $\mu \in S_n[t]$, then

$q_1 \equiv q_2 \Longleftrightarrow [q_1] \sim [q_2]$.

Therefore q_0-masking all $\mu \in S_n[t]$ implies that if any machine M^{μ} has states in $[q_1]_{\mu}$ and $[q_2]_{\mu}$ (as defined above) which are equivalent, all machines have states in $[q_1]$ and $[q_2]$ which are equivalent.

There are some additional special properties of stuck-at faults that are worth noting before considering the synthesis problem. In terms of the notation developed earlier for these faults (1.23 - 1.24), that is, if $\mu = (\mu_1, \mu_2, \ldots, \mu_n) \in S_n$ then

$$\begin{aligned} X(\mu) &= \{i \,|\, \mu_i = \sigma_x\}, \\ 0(\mu) &= \{i \,|\, \mu_i = \sigma_0\}, \\ 1(\mu) &= \{i \,|\, \mu_i = \sigma_1\}, \\ C(\mu) &= 0(\mu) \cup 1(\mu), \end{aligned} \tag{1.47}$$

and if $q = (b_1, b_2, \ldots, b_n) \in \{0, 1\}^n$, then

$$\begin{aligned} 0(q) &= \{i \,|\, b_i = 0\}, \\ 1(q) &= \{i \,|\, b_i = 1\}, \end{aligned} \tag{1.48}$$

we observe the following for any subset $F \subseteq S_n$.

Theorem 1.32

If $F \subseteq S_n$ and $\bigcap_{\xi \in F} \mathcal{R}(\xi) \neq \phi$ then

$$0(\mu) \cap 1(\gamma) = \phi, \text{ for all } \mu, \gamma \in F.$$

Proof

Suppose $q \in \bigcap_{\xi \in F} \mathcal{R}(\xi)$ and $\mu, \gamma \in F$. By Lemma 1.3

$$0(q) = 0(\mu(q)) = 0(\mu) \cup 0(q)X(\mu)$$

$$1(q) = 1(\gamma(q)) = 1(\gamma) \cup 1(q)X(\gamma)$$

so that

$$0(\mu) \subseteq 0(q) \text{ and } 1(\gamma) \subseteq 1(q). \tag{1.49}$$

But $0(q) \cap 1(q) = \phi$ and therefore

$$0(\mu) \cap 1(\gamma) = \phi$$

which proves the theorem.

The next two observations show how information about the images of a state r under a set of faults can be used to infer knowledge of the coordinates of r.

Theorem 1.33

If $F \subseteq S_n$ and $r = (r_1, r_2, \ldots, r_n) \in \{0,1\}^{(n)}$ then

$$r_j = \begin{cases} 0, \text{ if there exists } \mu \in F \text{ such that } j \in X(\mu) \cap 0(\mu(r)) \\ 1, \text{ if there exists } \mu \in F \text{ such that } j \in X(\mu) \cap 1(\mu(r)) \\ \text{indeterminate, if for all } \gamma \in F,\ j \in C(\gamma). \end{cases}$$

Proof

Immediate from Definitions (1.47) and (1.48).

If further, for some $\mu \in F$, it is known that $\mu(r) \in \bigcap_{\xi \in F} \mathcal{R}(\xi)$ then Theorem 1.33 can be strengthened as follows

Theorem 1.34

If $F \subseteq S_n$, $\mu(r) = s \in \bigcap_{\xi \in F} \mathcal{R}(\xi)$ then

$$r_j = \begin{cases} 0, \text{ if } j \in X(\mu) \text{ and there exists } \gamma \in F \text{ such that } j \in 0(\gamma) \\ 1, \text{ if } j \in X(\mu) \text{ and there exists } \gamma \in F \text{ such that } j \in 1(\gamma) \\ s_j, \text{ if for all } \gamma \in F,\ j \in X(\gamma) \\ \text{indeterminate, if } j \in C(\mu) \end{cases}$$

Proof

If $j \in X(\mu)$ and $\exists\, \gamma \in F$ such that $j \in 0(\gamma)$ then $j \in 0(s)$ since $s \in \bigcap_{\xi \in F} \mathcal{R}(\xi)$ (cf. (1.49)). But $0(s) = 0(\mu(r))$ and hence, by Theorem 1.33, $r_j = 0$. Similarly, if $j \in X(\mu)$ and $\exists\, \gamma \in F$ such that $j \in 1(\gamma)$, $r_j = 1$. The proof of the remaining statements is obvious.

We now consider the application of these observations to the problem of relating the machines M^{μ}, $\mu \in S_n[t]$, in terms of the state sets $[q]_{\mu}$, $q \in Q_0$, to the state assignment for M. In both cases we are dealing with known sets of states [q] and [s] where $s = \delta(q, a)$, $a \in I$. The object is to relate subsets of [q] and [s]

1) to each other by determining the correspondence between states and machines M^{μ}, and then

2) to the machine M by determining the state(s) r which arise from fault free transitions

$$\delta(q', a) = r, \quad q' \in [q]$$

To introduce some notation, let

$$\bar{\delta}^{\mu}([q]_{\mu}, z) = \{s' \mid \text{there exists } q' \in [q]_{\mu} \text{ such that } \bar{\delta}^{\mu}(q', z) = s'\} \tag{1.50}$$

for $z \in I^*$. Then we note that

$$\forall\, \mu \in S_n[k],\ \forall\, z \in I^*,\ \bar{\delta}^{\mu}([q]_{\mu}, z) \subseteq [\bar{\delta}(q, z)]_{\mu} \tag{1.51}$$

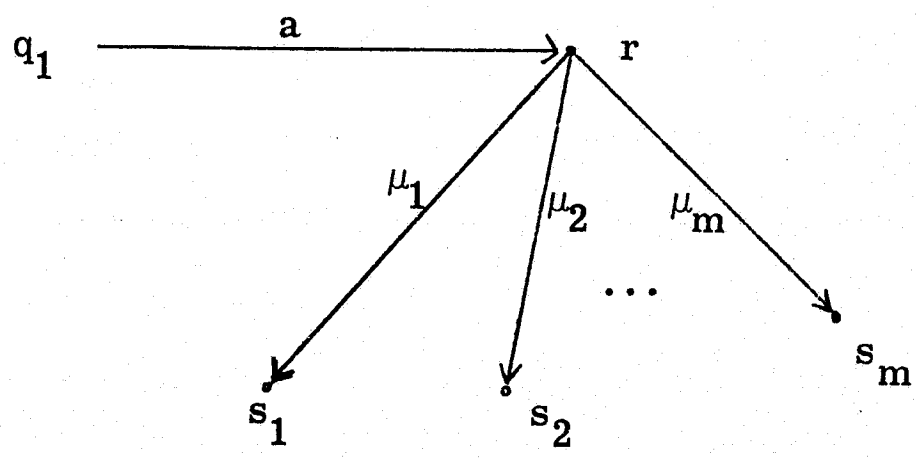
since $s' \in \bar{\delta}^{\mu}([q]_{\mu}, z)$ implies

there exists $q' \in [q]_{\mu}$ such that $^{\mu}(q', z) = s'$

and

$$q\ U_{\mu} q' \Longrightarrow \bar{\delta}(q, z) U_{\mu} \bar{\delta}^{\mu}(q', z) \Longrightarrow s' \in [\bar{\delta}(q, z)]_{\mu}\ .$$

In the first case we deal with the divergence to states $s_1, s_2, \ldots, s_m$ as shown in the diagram



Let $q \in Q_0$ and $q_1 \in \bigcap_{\gamma \in F} [q]_\gamma$ such that $\delta(q_1, a) = r$, $a \in I$ and $\mu_i(r) = s_i$, $\mu_i \in F$, $i=1, 2, \ldots, m$. Then applying Theorem 1.33, information about the states s_i can be used to specify r.

Example 1.9

Let $F = \{\mu_1, \mu_2, \mu_3\}$ where

μ_1 = x0xxx

μ_2 = xx0xx

μ_3 = xxxx0

and M be a state assigned machine such that

$$q_1 = 0 \in \bigcap_{\gamma \in F} [q]_\gamma$$

$$s_1 = \delta^{\mu_1}(q_1, a) = 23$$
$$s_2 = \delta^{\mu_2}(q_1, a) = 27$$
$$s_3 = \delta^{\mu_3}(q_1, a) = 30$$

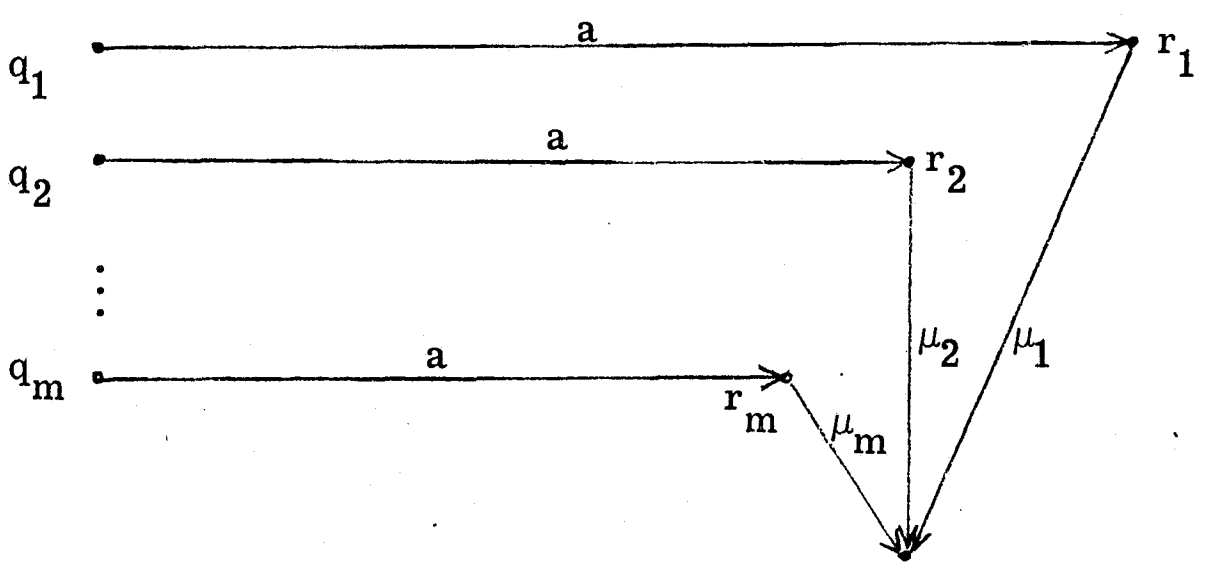
Then for $r = (r_1, r_2, r_3, r_4, r_5)$

$$\mu_{11} = \sigma_x,\ s_{11} = 1 \Longrightarrow r_1 = 1$$
$$\mu_{22} = \sigma_x,\ s_{22} = 1 \Longrightarrow r_2 = 1$$
$$\mu_{13} = \sigma_x,\ s_{13} = 1 \Longrightarrow r_3 = 1$$
$$\mu_{14} = \sigma_x,\ s_{14} = 1 \Longrightarrow r_4 = 1$$
$$\mu_{15} = \sigma_x,\ s_{15} = 1 \Longrightarrow r_5 = 1$$

so that

$$\delta(0, a) = 31.$$

In the second case we are concerned with the convergence of states $q_1, q_2, \ldots, q_m$ to s_1 according to the diagram



More precisely, let $q \in Q_0$ and for $\mu \in F$, $q' \in [q]_\mu$ such that $\delta(q', a) = r$ and $\mu(r) = s_1 \in \bigcap_{\gamma \in F} \delta^\gamma([q]_\gamma, a)$. Then using Theorem 1.34, information about the faults in F can be applied to the specification of states r.

Example 1.10

Let $F = \{\mu_1, \mu_2, \mu_3\}$ where

μ_1 = x1xxx

μ_2 = xxx0x

μ_3 = xxxx1

and M be a state-assigned machine such that

$$q_1 = 9 \in [q]_{\mu_1}$$

$$q_2 = 13 \in [q]_{\mu_1}$$

$$q_3 = 29 \in [q]_{\mu_2}$$

$$q_4 = 5 \in [q]_{\mu_3}$$

$$s_1 = 25 \in \bigcap_{\gamma \in F} \delta^{\gamma}([q]_{\gamma}, a), \quad a \in I.$$

Then for $r_1 = (r_{11}, r_{12}, r_{13}, r_{14}, r_{15})$

$\mu_{11} = \mu_{21} = \mu_{31} = \sigma_x$, $s_{11} = 1 \Longrightarrow r_{11} = 1$

$\mu_{12} = \sigma_1 \Longrightarrow r_{12} = x$

$\mu_{13} = \mu_{23} = \mu_{33} = \sigma_x$, $s_{13} = 0 \Longrightarrow r_{13} = 0$

$\mu_{14} = \sigma_x$, $\mu_{24} = \sigma_0 \Longrightarrow r_{14} = 0$

$\mu_{15} = \sigma_x$, $\mu_{35} = \sigma_1 \Longrightarrow r_{15} = 1$

so that $\delta(9, a) = 17$ or 25. Similarly

$\delta(13, a) = 17$ or 25

$\delta(29, a) = 25$ or 27

$\delta(5, a) = 24$ or 25

To summarize our findings we see that the search for machines which q_0-mask faults can be divided as follows:

1) identify state sets $[q]_\mu$ suitable for q_0-masking given the behavior to be realized and the faults to be masked
2) verify the state sets (state assignment for the machines $M^{(\mu)}$) by deriving the transition function for M as a state assigned machine.

Most of our results apply to the problem of verification; to demonstrate the application of the definitions and theorems to this problem, we consider an example.

Example 1.11

The object of this example is to find

1) sequences of period 7 which can be generated by 5-dimensional machines such that all $\mu \in S_5[1]$ are q_0-masked, and
2) the machines which generate these sequences.

We assume a classification of states and proceed to apply the above results to achieve the objective. Let the faults in $S_5[1]$ be denoted by

$\mu_1 = xxxx1$ $\quad$ $\mu_2 = xxxx0$

$\mu_3 = xxx1x$ $\quad$ $\mu_4 = xxx0x$

$\mu_5 = xx1xx$ $\quad$ $\mu_6 = xx0xx$

$\mu_7 = x1xxx$ $\quad$ $\mu_8 = x0xxx$

$\mu_9 = 1xxxx$ $\quad$ $\mu_{10} = 0xxxx$

Then the states, denoted by the decimal equivalent of the binary 5-tuple, are classified according to the following table:

Q	Q^{μ_1}	Q^{μ_2}	Q^{μ_3}	Q^{μ_4}	Q^{μ_5}	Q^{μ_6}	Q^{μ_7}	Q^{μ_8}	Q^{μ_9}	$Q^{\mu_{10}}$
0	1	0	2	0	4	0	8	0	16	0
31	11	30	30	29	29	27	15	23	27	15
21	21	14	14	13	13	19	9	21	19	1
17	17	6	6	12	12	18	28	17	18	10
3	3	20	22	20	22	25	25	3	25	3
5	5	26	3	25	5	26	26	5	26	5
7	7	24	7	24	7	24	24	7	24	7
0	1	0	2	0	4	0	8	0	16	0

This classification was obtained by trial and error to conform with the conditions imposed by Theorems 1.30 through 1.34. The states in the first row correspond to the initial states for q_0-masking in a 5-dimensional machine. The states in the seventh row are the minumum number necessary to cover all the faults in $S_5[1]$. The rows in between are then filled, working backward from the sixth row, with states such that a state appears in as many columns and as few rows as possible. The states in the column designated by Q^{μ} are in the range of the fault μ and the periodicity of states in each column is 7.

The rows of the classification table are potential sets of right relatives [q] for states q reachable from q_0:

$\{0, 1, 2, 4, 8, 16\}$

$\{11, 15, 23, 27, 29, 30, 31\}$

$\{1, 9, 13, 14, 19, 21\}$

$\{6, 10, 12, 17, 18, 28\}$

$\{3, 20, 22, 25\}$

$\{3, 5, 25, 26\}$

$\{7, 24\}$

If these sets are to be the right relatives [q], $q \in Q_0$, of a machine M that q_0-masks all faults in $S_5[1]$, then by Lemma 1.7 all states in the same set must have the same output; the output assignment must define a partition which is refined by the partition

$$\{\overline{0, 1, 2, 4, 8, 9, 13, 14, 16, 19, 21};\ \overline{3, 5, 20, 22, 25, 26};$$
$$\overline{6, 10, 12, 17, 18, 28};\ \overline{7, 24};\ \overline{11, 15, 23, 27, 29, 30, 31}\}.$$

For the output assignment:

For all $q \in \{0, 1, 2, 4, 8, 9, 13, 14, 16, 19, 21\}$, $\omega(q, a) = b$, $a \in I$

For all $q \in \{11, 15, 23, 27, 29, 30, 31\}$, $\omega(q, a) = c$

For all $q \in \{6, 10, 12, 17, 18, 28\}$, $\omega(q, a) = d$ (1.52)

For all $q \in \{3, 5, 20, 22, 25, 26\}$, $\omega(q, a) = e$

For all $q \in \{7, 24\}$, $\omega(q, a) = f$

this partition implies that sequences of the form

bcbdeef, b, c, d, e, f, $\in \{0, 1\}$

can be generated if a machine M exists with transition function such that the columns of the classification table represent the successive states of the faulty machines M^{μ}. The transition table is therefore derived from this table using Theorems 1.33 and 1.34. For the transitions $\delta(0,a)$, $\delta(18,a)$, and $\delta(25,a)$ we find

$$\delta^{\mu_2}(0,a) = 30 \Rightarrow \delta(0,a) = 30 \text{ or } 31$$

$$\delta^{\mu_4}(0,a) = 29 \Rightarrow \delta(0,a) = 29 \text{ or } 31$$

$$\Rightarrow \delta(0,a) = 31$$

$$\delta^{\mu_4}(20,a) = \delta^{\mu_6}(18,a) = \delta^{\mu_9}(19,a) = \delta^{\mu_7}(28,a) = s$$

$$\Rightarrow \delta(18,a) = 24 \text{ or } 25 \text{ (1100x)}$$

$$s = 25 \Rightarrow \delta(18,a) = 25$$

and

$$\delta^{\mu_7}(25,a) = 26 \Rightarrow \delta(25,a) = 18 \text{ or } 26$$

$$\delta^{\mu_9}(25,a) = 26 \Rightarrow \delta(25,a) = 10 \text{ or } 26$$

$$\Rightarrow \delta(25,a) = 26$$

In summary, the transition table is:

q	δ(q)	q	δ(q)	q	δ(q)	q	δ(q)
0	31	8	15	16	27	24	0
1	10	9	28	17	3	25	26
2	30	10	3	18	25	26	24
3	5	11	21	19	18	27	19
4	29	12	22	20	27	28	25
5	7	13	12	21	17	29	13
6	20	14	6	22	1	30	14
7	0	15	1	23	21	31	21

(1.53)

Note that there is no output assignment such that nontrivial sequences of period less than 7 can be realized with this transition function and all $\mu \in S_5[1]$ are q_0-masked. This is because each column of the classification table has period not less than 7 and 7 is prime.

In conclusion we have found 30 nontrivial sequences of the form

$$bcbdeef, \quad b, c, d, e, f \in \{0,1\}$$

which can be generated by a 5 dimensional machine such that all $\mu \in S_5[1]$ are q_0-masked. The machine to generate a particular sequence has the transition function of (1.53) and the output assignment given by (1.52).

2. FAULTS IN COMBINATIONAL NETWORKS

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

Mathematical Formulation of Fault Concepts

To be useful in the analysis and design of fault tolerant data systems, a model must not only describe the fault-free behavior of the system but also the effects of internal faults on the system behavior. Since a degree of detection and location is usually desirable, the model also needs to reflect the organization of the system to within the detection and location constraints.

Although most data systems currently employ binary signal sets in digital control and data manipulation subsystems, future designs may expand their basic signal sets. [30] In addition, certain kinds of faults produce signals other than the basic data signals of the system. [5], [14] To allow for advancements in this direction, our model also includes information on signal sets that appear in the system.

The organization of a system is represented to the desired degree of sophistication by a directed graph.

Definition 2.1

An (n, m, k, ℓ)-graph is a directed graph (digraph) with $n+m+k$ labeled points and ℓ labeled lines that is connected and has

no directed cycles. Exactly n of the points, called input terminals or inputs, have indegree 0. Lines incident with input terminals are input lines. (Multiple lines incident with a single input terminal all have the same label and count as one line in determining the parameter ℓ.) Exactly m of the points, called output terminals or just outputs, have outdegree 0 and indegree 1. Lines incident with output terminals are output lines. The remaining k points are called nodes.

The nodes of an (n, m, k, ℓ)-graph have both indegree and outdegree greater than or equal to 1. In general, node i has indegree n_i and outdegree m_i where n_i and m_i are positive integers.

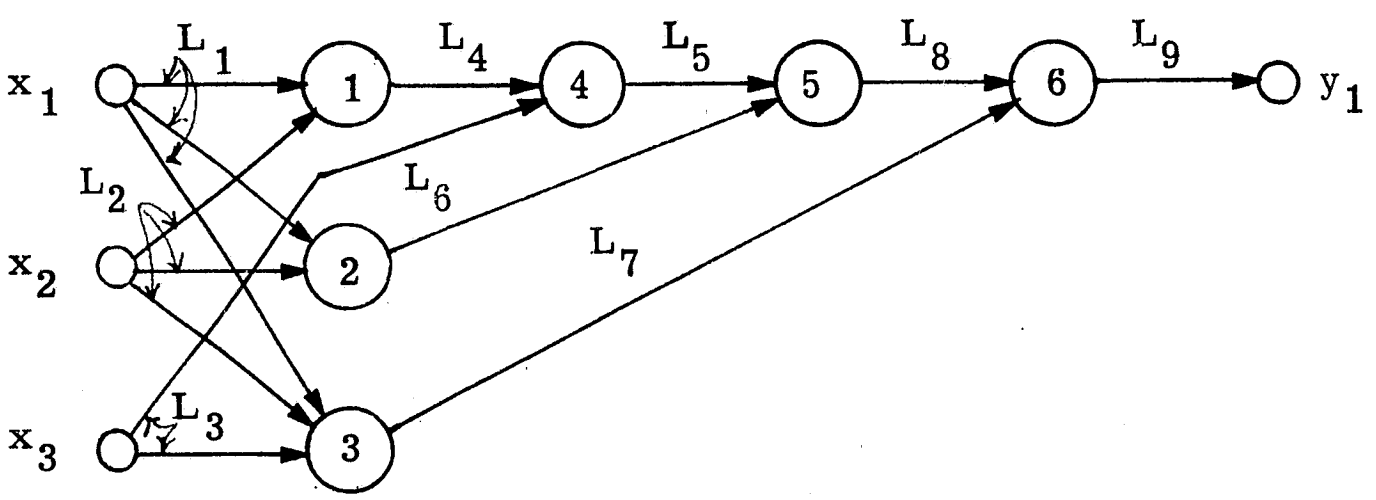
In Figure 2.1, digraph D is a $(3,1,6,9)$-graph. The 3 input terminals are labeled x_1, x_2, x_3 and the output terminal is labeled y_1. The 6 nodes are labeled with the integers $1, 2, \ldots, 6$. It will usually be convenient to use this convention when labeling points in the diagraph. The input lines of D are L_1, L_2, L_3 while the single output line is L_9. The node indegrees and outdegrees are as follows:

$$n_1 = n_2 = n_4 = n_5 = n_6 = 2$$

$$n_3 = 3$$

$$m_1 = m_2 = m_3 = m_4 = m_5 = m_6 = 1$$

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 ℓ is the number of interconnecting lines. A node may



Digraph D

$S = (L_1, \ldots, L_9)$

$L_i = B = \{0,1\},\ 1 \leq i \leq 9$

$F = E_C$

b = (OR, AND, $\overline{\text{XOR}}$, AND, OR, AND)

Figure 2.1
(continued on next page)

Example of a Combinational Network C = (D, S, F, b)

x_1	x_2	x_3	y_1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Truth Table A

function of C

α(b)

x_1	x_2	x_3	y_1
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Truth Table B

malfunction of C

α(OR, AND, $\overline{\text{XOR}}$, AND, 1, AND)

Figure 2.1
(continued from previous page)

Example of a Combinational Network C = (D, S, F, b)

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 in a system are represented by a signal vector, S. The signal vector of an (n, m, k, ℓ)-graph is an ℓ-tuple of sets. When interpreting the model, the set of signals that are possible on the i^{th} line in the system is the set appearing as the i^{th} coordinate of the ℓ-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. The label of line j_i in an (n, m, k, ℓ)-graph is the name of the signal set applicable to line j_i. In digraph D of Figure 2.1, L_5 is the signal set on the line directed from node 4 to node 5. As indicated it is the binary set $B = \{0, 1\}$. In the usual switching circuit interpretation, each coordinate of S will be this same binary set.

Definition 2.2

In an (n, m, k, ℓ)-graph 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 2.3

In an (n, m, k, ℓ)-graph the output space associated with node i, called O_i, is defined as the cartesian product of the signal sets on the output lines of node i, again taken in order of ascending line labels for convenience.

Definition 2.4

In an (n, m, k, ℓ)-graph, $E = \{(g_1, g_2, \ldots, g_k) \mid g_i$ is a mapping from I_i into O_i 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 network. In many applications, only a small fraction of these will occur. For example, in the digraph of Figure 2.1, $I_1 = I_2 = L_1 \times L_2 = B^{(2)}$, $I_3 = L_1 \times L_2 \times L_3 = B^{(3)}$, $O_5 = L_8 = B$, $O_6 = L_9 = B$, etc. $E = \{(g_1, g_2, \ldots, g_6) \mid g_1, g_2, g_4, g_5$ and g_6 are mappings from $B^{(2)}$ into B and g_3 is a mapping from $B^{(3)}$ into B$\}$. The cardinality of E in this graph, 2^{28}, indicates that this set is very large even for small networks.

We are now ready to formally define the concept of a combinational network with more than one possible structure, i.e., a combinational network subject to faulty operation.

Definition 2.5

An (n, m, k, ℓ)-combinational network is a 4-tuple

$$C = (D, S, F, b)$$

where D is an (n, m, k, ℓ)- graph, called the graph of C.

S is an ℓ-tuple of sets, called the signal vector of C.

$F \subseteq E_C$ is called the fault set of C or simply the fault set.

(E_C is defined as in Definition 2.4.)

$b \in F$ is called the 0-fault of C or simply the 0-fault.

Definition 2.6

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

In Figure 2.1, D is the graph of a (3, 1, 6, 9)-combinational network C. Each coordinate of the signal vector is the binary set $B = \{0,1\}$. The fault set of C is the whole set E_C. Each coordinate of the 0-fault of C is a common switching function. The element (OR, AND, XOR, NAND, NOR, AND) is a proper fault of C.

We now relate the elements in our formal definition to the requirements discussed in the introduction. The graph of C describes the organization of the system. Each module of interest appears as one of the k nodes of the graph. Each input terminal of the system is represented by one of the n inputs of the graph; likewise, each system output is represented by one of the m outputs of the graph. The lines of the graph describe the modular interconnection structure of the

system. Each electrically isolated wire in the fault-free system and each wire segment that may become isolated by a proper system fault is represented by one of the ℓ labeled lines of the graph. Wire segments that can never become electrically separated may be represented by several lines of the graph if necessary with the same label. A collection of lines with the same label are treated as one line in determining the parameter ℓ.

In applications, b represents the fault-free structure of the system. The i^{th} coordinate of the vector b represents the mapping performed by the i^{th} module of the system when the system is fault-free. The fault set, F, is the set of all allowed system structures. Since the fault-free structure is an allowed structure, $b \in F$. A system structure different from the fault-free structure is called a proper fault of C. By two structures f and f' being "different" we mean that at least one coordinate of f is a different mapping than the corresponding coordinate of f'. In a system application, a proper fault structure is one in which at least one module is performing a mapping different from the mapping it performs in the fault-free structure. System disturbances that do not change the system structure are not considered proper faults of the system.

Definition 2.7

The input space, I, for combinational network C = (D, S, F, b), is defined as the cartesian product of the signal sets on the input lines of D (taken in order of increasing node label).

There is one coordinate in the input space for each input terminal; hence, the input space is an n-dimensional space. In the usual switching theory interpretation, $I = B^{(n)}$.

Definition 2.8

The output space, O, for combinational network C = (D, S, F, b) is defined as the cartesian product of the signal sets appearing on the output lines of D (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, $O = B^{(m)}$. In the combinational network of Figure 2.1, $I = B^{(3)}$ and O = B.

Definition 2.9

In a combinational network C = (D, S, F, b)

$$T_C = \{g \mid g \text{ is a mapping from I into O}\}$$

The graph of C in a combinational network C = (D, S, F, b) induces a mapping α, called the net mapping, from F into T_C where $\alpha(f)$ is the (combinational) function realized by C under the fault f.

Definition 2.10

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

Definition 2.11

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

Definition 2.12

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

Definition 2.13

The malfunction set of the network C is the set $T - \{t\}$.

The behavior of the system at a given time refers to the I–O mapping that the system is performing at that time. Thus two different system structures may produce different system behaviors or they may produce the same system behavior. 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.

In the combinational network C of Figure 2.1, the function of C is as displayed in Truth Table A. For this network, $t = \alpha(b) = \alpha(OR, AND, \overline{XOR}, AND, OR, AND) = (x_3(x_1 + x_2) + x_1x_2)\ \overline{(x_1 \oplus x_2 \oplus x_3)}$. Truth Table B displays a malfunction of C, $t' = \alpha(OR, AND, \overline{XOR}, AND, 1, AND) = \overline{x_1 \oplus x_2 \oplus x_3}$. This malfunction results from a SA-1 fault at node 5. (SA-1 means "stuck at 1".)

Consider the combinational network

$$C = (D, S, F, b)$$

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

Definition 2.14

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

Note that $K_b = \phi$. This is to be expected since K_f is in a sense a list of the "faulty" nodes under the fault f.

Definition 2.15, Fault Diagnosis Concepts

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

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

(c) $f \in F$ is locatable iff $(f' \in F,\ \alpha(f) = \alpha(f') \Rightarrow K_f = K_{f'})$

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

A detectable fault of C will often be called a failure of C.

In the usual interpretation, a masked fault represents a system structure that produces the same system behavior as the fault-free structure. A proper masked fault is a system structure different from the fault-free structure that produces the same system behavior as the fault-free structure. A detectable fault represents a system structure that produces a system behavior different from the fault-free behavior.

In a typical application, the set K_f represents the set of "faulty" nodes of the system structure represented by the fault f. In other

words, $i \epsilon K_f$ if and only if node i performs a mapping in the system structure f different from the mapping performed by node i in the fault-free structure.

When troubleshooting a system, it is convenient to be able to determine which modules are "faulty" by taking measurements at the system terminals. The "faulty" modules may then be replaced by good ones and repaired at leisure. This reduces down time for repair. A locatable fault represents a system structure whose "faulty" nodes may be identified by terminal measurements alone. A completely diagnosable fault represents a system structure identifiable by terminal measurements alone, i.e., the whole structure can be determined from terminal measurements.

Some Elementary Observations

The previous discussion suggests the following lemmas whose proofs are immediate or trivial. Their inclusion here is justified by frequent reference in the formal proofs of later sections.

Lemma 2.1.1

In any combinational network C = (D, S, F, b) the 0-fault is masked.

Proof Immediate

$\alpha(b) = t$ by Definition 2.10 and b is masked by Definition 2.15(a).

Lemma 2.1.2

In any combinational network C = (D, S, F, b), f ϵ F is masked iff f is not detectable.

Proof

Immediate from Definition 2.15(a) and 2.15(b).

Lemma 2.1.3

In any combinational network C = (D, S, F, b), no masked proper fault is locatable.

Proof

Let f be any masked proper fault of C. Since f is proper, $f \neq b$ and $K_f \neq \phi = K_b$, but since f is masked $\alpha(f) = \alpha(b)$ and f is not locatable (Definition 2.15c) proving the lemma.

Lemma 2.1.4

In any combinational network C = (D, S, F, b), f ϵ F and f completely diagnosable $\Longrightarrow$ f is locatable.

Proof

Let f be completely diagnosable and let f' be any element of F such that $\alpha(f) = \alpha(f')$, then f = f' (Definition 2.15(d)), $K_f = K_{f'}$, and f is locatable (Definition 2.15(c)).

Theorem 2.1

In any combinational network C = (D, S, F, b), the following statements are equivalent.

(1) All proper faults of C are detectable.

(2) b is completely diagnosable.

(3) b is locatable.

Proof

(1) $\Longrightarrow$ (2)

Let (1) hold and assume that b is not completely diagnosable then $\exists f \in F \ni \alpha(f) = \alpha(b) = t$ and yet $f \neq b$. Thus f is proper but not detectable, contradicting the hypothesis.

(2) $\Longrightarrow$ (3) Immediate from Lemma 2.1.4.

(3) $\Longrightarrow$ (1)

Let f be any proper fault of C, then $f \neq b$ (Definition 2.6). Since f is proper, $K_f \neq K_b = \emptyset$. As b is locatable by hypothesis, $\alpha(f) \neq \alpha(b) = t$ (Definition 2.15(c)), and f is detectable (Definition 2.15(b)).

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 2.16

A fault $f = (f_1, f_2, \ldots, f_k)$ in an (n, m, k, ℓ)-combinational network is called a single fault at node i, if $K_f = \{i\}$.

Note that all single faults are by definition proper faults.

Fault Masking and Detection in Two Node Systems with Single Faults

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

Definition 2.17

A general two node system with single faults is an $(n, m, 2, \ell)$-combinational network

$$C = (D, S, F, b)$$

where D is an $(n, m, 2, \ell)$-graph as shown in Figure 2.2

S is an arbitrary ℓ-tuple of sets

$b = (b_1, b_2)$ is an arbitrary element of E_C

$F = \{(f_1, f_2) \mid (f_1, f_2) \in E_C$, and either $f_1 = b_1$ or $f_2 = b_2\}$

E_C is as defined in Definition 2.4.

The line labels are omitted from the graph of C because there are an unspecified number of lines. The degree constraints for the graph of C are:

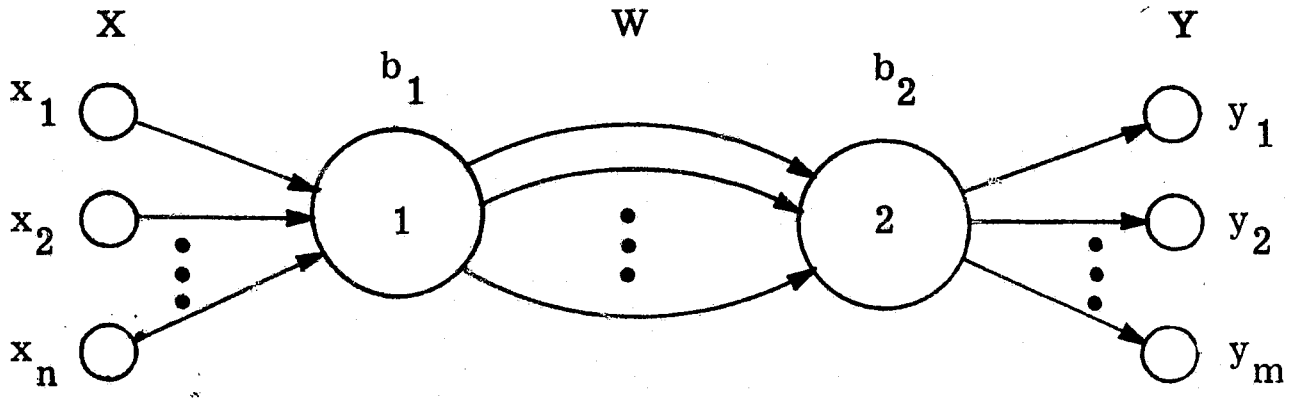


Figure 2.2

Graph of a General Two-node System

$$n_1 = n$$

$$n_2 = m_1$$

$$m_2 = m$$

The input space of C is X and the output space of C is Y. The output space associated with node 1 is W, and by construction, W is also the input space associated with node 2.

From the graph of C, it can be seen that if $f = (f_1, f_2) \in F$, then $\alpha(f_1, f_2) = f_2 f_1$ where $f_2 f_1$ denotes the composition $f_2 \cdot f_1$, first applying f_1. In particular, for the net function, $t = b_2 b_1$.

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

Definition 2.18

In a general two node system

$$w_1 \equiv_{b_2} w_2 \text{ iff } b_2(w_1) = b_2(w_2)$$

The set of equivalence classes of $\equiv_{b_2}$ are the blocks of a partition of W called π_{b_2}.

Masked single faults at node 1 are characterized as follows:

Theorem 2.2

In the general two node system with single faults, a single fault at node 1, $f = (f_1, b_2)$, is masked if and only if

$$\forall x \in X, \quad f_1(x) \underset{b_2}{\equiv} b_1(x)$$

Proof

$$f_1(x) \underset{b_2}{\equiv} b_1(x), \forall x \in X \iff b_2 f_1(x) = b_2 b_1(x) \; \forall x \in X$$

(Definition 2.18)

$$\iff b_2 f_1 = b_2 b_1$$

$$\iff b_2 f_1 = t$$

$$\iff \alpha(f) = t$$

Due to the exclusive character of masked faults and detectable faults (Lemma 2.1.2) the following statement characterizing detectable faults at node 1 is an immediate corollary of Theorem 2.2.

Corollary 2.2.1

In the general two node system with single faults, a single fault at node 1, $f = (f_1, b_2)$, is detectable iff

$$\exists x \in X \text{ such that } f_1(x) \underset{b_2}{\neq} b_1(x)$$

Single faults at node 2 are masked under the following conditions:

Theorem 2.3

In the general two node system with single faults, a single fault at node 2, $f = (b_1, f_2)$ is masked iff

$$f_2 | \mathcal{R}(b1) = b_2 | \mathcal{R}(b1)$$

Proof: Sufficiency

Let $f_2 | \mathcal{R}(b1) = b_2 | \mathcal{R}(b1)$ and let x be any element of X; then by hypothesis

$$b_2 b_1(x) = f_2 b_1(x) \quad \forall x \in X$$

and $b_2 b_1 = f_2 b_1$ or $\alpha(b_1, b_2) = \alpha(b_1, f_2)$, and the fault f is masked (Definition 2.15a).

Necessity

Let f be masked, then $\alpha(f) = \alpha(b) = t$. Let w be any element of the range of b_1. There must exist $x \in X$ such that $b_1(x) = w$ and we have $b_2(w) = b_2 b_1(x) = f_2 b_1(x) = f_2(w)$. Since w was any element of the range of b_1, we have

$$f_2 | \mathcal{R}(b1) = b_2 | \mathcal{R}(b1).$$

This concludes the proof of Theorem 2.3.

Due to the exclusive character of masked faults and detectable faults (Lemma 2.1.2) the following characterization of detectable faults at node 2 follows immediately from Theorem 2.3.

Corollary 2.3.1

In the general two node system with single faults, a single fault at node 2, $f = (b_1, f_2)$, is detectable iff

$$f_2 | \mathcal{R}(b1) \neq b_2 | \mathcal{R}(b1)$$

We will now use our set-theoretic characterization of masked faults to count the number of single faults at node 1 that are masked.

Definition 2.19

For every $y_i \in b_2 b_1(X)$ in the general two node system, define

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

and

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

Note that each c_i is the cardinality of one of the blocks of partition π_{b_2} defined in Definition 2.18. Similarly, each d_i is the cardinality of one of the blocks of $\pi_{b_2 b_1}$ the natural partition on X induced by the mapping $b_2 b_1$ from X into Y.

Definition 2.20

In an (n, m, k, ℓ)-combinational network C = (D, S, F, b), for every i, $1 \leq i \leq k$, define

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

Theorem 2.4

In the general two node system with single faults, let $b_2 b_1(X) = \{y_1, y_2, \ldots, y_q\}$, then

$$|F_1| = -1 + \prod_{i=1}^{q} c_i^{d_i}$$

Proof

Our task is to count the number of single faults at node 1 that satisfy the conditions of Theorem 2.2. Our network allows any mapping $f_1: X \to W$ (except b_1) to appear as the first coordinate of a single fault at node 1. For a moment, we will not concern ourselves with b_1 but will take it into account later. Let $X_i = (b_2 b_1)^{-1}(y_i)$ and $W_i = b_2^{-1}(y_i)$ for every $y_i \in b_2 b_1(X)$. The following illustration depicts the relationship between X_i and W_i and the mappings b_2 and b_1.

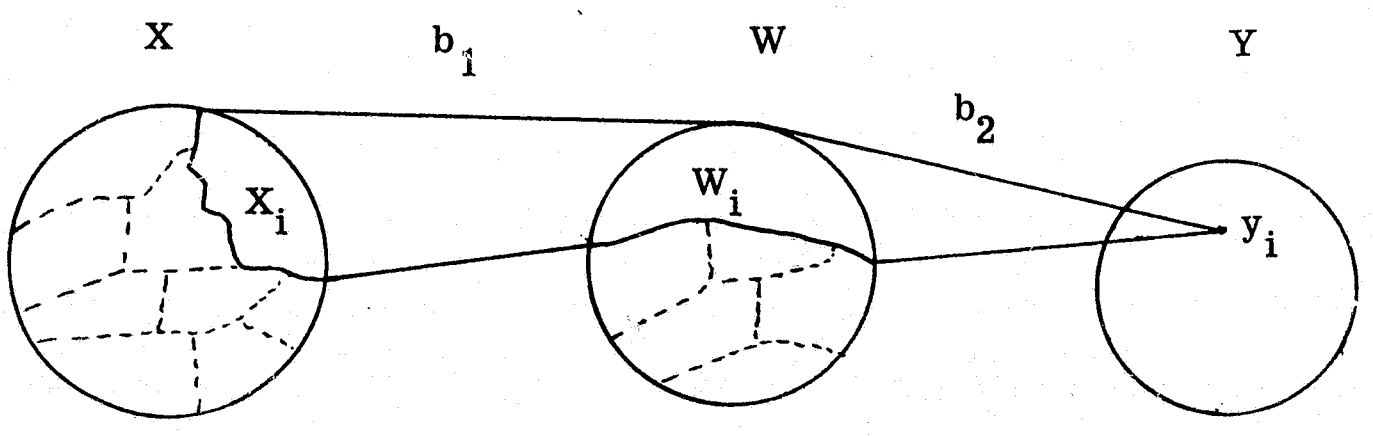


Figure 2.3

Since W_i contains all elements in W that map onto y_i under b_2, then clearly $b_1(X_i) \subseteq W_i$. Furthermore, since Theorem 2.2 requires only that $f_1(x) \underset{b_2}{\equiv} b_1(x)$ for every $x \in X$, we have $f_1(X_i) \subseteq W_i$. There are thus $c_i^{d_i}$ possible subfunctions from X_i into W_i that satisfy Theorem 2.2. Since X_i and W_i are blocks of partitions on X and W respectively, we have

$$\prod_{i=1}^{q} c_i^{d_i}$$

possible functions f_1 from X into W in which $f_1|X_i$ is a subfunction from X_i into W_i for $1 \leq i \leq q$. This is the number of functions f_1 at node 1 for which $b_2 f_1 = b_2 b_1$. Since b_1 is clearly one of those counted, the number of single faults at node 1 that are masked is given by

$$|F_1| = -1 + \prod_{i=1}^{q} c_i^{d_i}$$

proving the theorem.

Corollary 2.4.1

In the general two node system with single faults, if $b_2 b_1(X) = \{y_1, \ldots, y_q\}$, the number of single faults at node 1 that are detectable is given by

$$a^p - \prod_{i=1}^{q} c_i^{d_i}$$

where

$$a = |W|$$

$$p = |X|$$

Proof

Since the total number of single faults at node 1 is given by

$$-1 + a^p$$

the corollary follows immediately from Theorem 2.4 and Lemma 2.1.2.

We now proceed to count the number of single faults at node 2 that are masked.

Theorem 2.5

In the general two node system with single faults

$$|F_2| = -1 + s^{\overline{r}}$$

where

$$s = |Y|$$

$$\overline{r} = |\overline{\mathcal{R}(b1)}|$$

Proof

Faults of the form $f = (b_1, f_2)$ are masked only if $f_2|\mathcal{R}(b1) = b_2|\mathcal{R}(b1)$ (Theorem 2.3) and we see that $f_2(w)$ is specified for every $w \in \mathcal{R}(b1)$. Since $f_2|\mathcal{R}(b1) = b_2|\mathcal{R}(b1)$ is also sufficient to insure that f is masked, $f_2(w)$ may be any element of Y when $w \in \overline{\mathcal{R}(b1)}$. Since $f_2(w_1)$ is in no way dependent upon $f_2(w_2)$ when w_1, $w_2 \in \overline{\mathcal{R}(b1)}$ there are $s^{\overline{r}}$ independent choices for f_2 that satisfy $f_2|\mathcal{R}(b1) = b_2|\mathcal{R}(b1)$: however, one of these is the mapping b_2. Therefore the number of single faults at node 2 that are masked is given by $(-1 + s^{\overline{r}})$, concluding the proof of Theorem 2.5.

The exclusive character of masked and detectable faults makes it easy to count the number of detectable single faults at node 2.

Corollary 2.5.1

In the general two node system with single faults, the number of single faults at node 2 that are detectable is given by

$$s^a - s^{\bar{r}} = s^{\bar{r}}(s^r - 1)$$

where

$$a = |W|$$

$$s = |Y|$$

$$r = |\mathcal{R}(b1)|$$

$$\bar{r} = |\overline{\mathcal{R}(b1)}|$$

Proof

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

$$-1 + s^a$$

Since all unmasked single faults are detectable (Lemma 2.1.2) and since $a = r + \bar{r}$, it follows immediately (Theorem 2.5) that the number of detectable single faults is

$$(-1 + s^a) - (-1 + s^{\bar{r}}) = s^a - s^{\bar{r}} = s^{\bar{r}}(s^r - 1)$$

Considering the system as a whole, we find that

Theorem 2.6

In a general two node system with single faults, if $b_2b_1(X) = \{y_1, y_2, \ldots, y_q\}$, the number of faults that are masked is given by

$$\prod_{i=1}^{q} c_i^{d_i} + s^{\bar{r}} - 1$$

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

$d_i = |(b_2b_1)^{-1}(y_i)|$

$s = |Y|$

$\bar{r} = |\overline{R(b1)}|$

Proof

Immediate from Theorem 2.4 and 2.5 and Lemma 2.1.1.

Theorem 2.7

In a general two node system with single faults if $b_2b_1(X) = \{y_1, y_2, \ldots, y_q\}$, the number of faults that are detectable is given by

$$s^a + a^p - s^{\bar{r}} - \prod_{i=1}^{q} c_i^{d_i}$$

Proof

Immediate from Corollaries 2.4.1 and 2.5.1.

For a completely specified function g, the input space X, the output space Y, the range of g and the partition induced on X by g are fixed. Some of the parameters in the equations that count the

number of masked faults are fixed; e.g. each d_i in Theorem 2.4 and s in Theorem 2.5 are determined by the function g. We can design for improved fault masking by choosing appropriate values for the remaining parameters i.e. for each c_i in Theorem 2.4 and for $\bar{r}$ in Theorem 2.5.

By increasing the value of each c_i, we can increase the number of single faults masked at node 1. But, in order to accomplish this, we either reduce $\bar{r}$, (thereby decreasing the number of single faults masked at node 2) or increase the size of space W. The former is not desirable because we improve node 1 at the expense of node 2. The latter approach seems inviting until we note that in addition to increasing the number of masked single faults at node 1, we also increase the number of single faults at node 1.

By increasing the value of $\bar{r}$, we can increase the number of single faults at node 2 that are masked. Again, in order to accomplish this, we must either decrease the value of some c_i or increase the size of W. The first alternative is not desirable because it decreases the number of single faults that are masked at node 1, and the second alternative again increases the total number of faults. A full discussion of optimization is included in a later section.

This discussion serves to emphasize that increased fault masking can be achieved more "easily" in the design of some functions than in others. For example, if the value of each d_i is large, a small increase in the size of W will result in a larger increase in the number of single

faults that are masked at node 1 than in a system for which the d_i values are small. Also, in a system with a large output space, a small increase in the size of W will result in a larger increase in the number of single faults masked at node 2 than in a system with a small output space. Systems that have both properties, will be easy to fault mask at both nodes. The precise effect of these parameters is discussed in the optimization section.

A system in which all single faults are masked would be very desirable if a high degree of reliability is required. With such a broad definition of fault, one would suspect that such a system design would be very difficult to achieve. This is indeed the case, although it is possible to mask all faults at node 1 or at node 2 for certain types of net functions. Before stating the conditions under which this is possible, it is necessary to establish the following lemmas.

Lemma 2.8.1

If n is a positive integer ≥ 2 and if x_i and y_i, $1 \leq i \leq n$, are positive integers, then

$$(x_1)^{y_1}(x_2)^{y_2}\ldots(x_n)^{y_n} < (x_1 + x_2 + \ldots + x_n)^{(y_1+y_2+\ldots+y_n)}$$

Proof: Basis Step

$$1 < \left(\frac{x_1+x_2}{x_2}\right)^{y_2} \text{ and } (x_1)^{y_1} < (x_1+x_2)^{y_1}$$

because x_1, x_2, y_1, and y_2 are positive integers. Multiplying we get

$$(x_1)^{y_1} < \left(\frac{x_1+x_2}{x_2}\right)^{y_2} \cdot (x_1+x_2)^{y_1} = \frac{(x_1+x_2)^{y_2}}{(x_2)^{y_2}} (x_1+x_2)^{y_1}$$

and

$$(x_1)^{y_1}(x_2)^{y_2} < (x_1+x_2)^{y_1+y_2}$$

Induction Step

Induction Hypothesis

If (n-1) is a positive integer ≥ 2 and if x_i and y_i, $1 \leq i \leq n-1$, are positive integers, then

$$(x_1)^{y_1} \ldots (x_{n-1})^{y_{n-1}} < (x_1+x_2+\ldots+x_{n-1})^{y_1+y_2+\ldots+y_{n-1}}$$

From the induction hypothesis, if x_n and y_n are positive integers

$$(x_1)^{y_1} \ldots (x_{n-1})^{y_{n-1}} < (x_1+x_2+\ldots+x_{n-1}+x_n)^{y_1+y_2+\ldots+y_{n-1}}$$

again

$$(x_n)^{y_n} < (x_1+\ldots+x_n)^{y_n}$$

and multiplying, we get

$$(x_1)^{y_1} \ldots (x_n)^{y_n} < (x_1+\ldots+x_n)^{y_1+\ldots+y_n}$$

proving the lemma.

In this section, $\mathcal{R}(g)$ will denote the range of function g: $\underset{g}{\equiv}$ will denote the equivalence relation induced by g on the domain of g (D(g)); and if x is an element of D(g), $[x]_g$ will denote the equivalence class of $\underset{g}{\equiv}$ that contains x.

Lemma 2.8.2

In the general two node system, if $|b_2 b_1(X)| = q$, then

$$(1) \quad \sum_{i=1}^{q} d_i = |X|$$

and

$$(2) \quad \sum_{i=1}^{q} c_i \leq |W|$$

Proof

Each d_i is the cardinality of a block of the partition, $\Pi_{b_2 b_1}$, induced on X by the mapping $b_2 b_1$. Since d_i is defined for every element in the range of $b_2 b_1$, (1) follows directly from the definition of a partition induced by a mapping.

Each c_i is the cardinality of a block of the partition Π_{b_2} induced on W by the mapping b_2. However, c_i may not be defined for every element in the range of b_2, therefore $\sum_{i=1}^{q} c_i$ may be less than $|W|$. However, since $\mathcal{R}(b_2 b_1) \subseteq \mathcal{R}(b_2)$, c_i cannot be defined for any y_i not in the range of b_2. If c_i is defined for every element in the range of b_2, then $\sum_{i=1}^{q} c_i = |W|$.

Theorem 2.8

In the general two node system with single faults, all single faults at node 1 are masked iff b_2 is a constant function.

Proof: Sufficiency

Let b_2 be a constant function, then there is a single element y in the set $b_2 b_1(X)$, i.e. $q = 1$. Moreover $(b_2 b_1)^{-1}(y) = X$ and $b_2^{-1}(y) = W$ so that $c_1 = |W| = a$ and $d_1 = |X| = p$. Then, from Theorem 2.4

$$|F_1| = -1 + \prod_{i=1}^{q} c_i^{d_i} = -1 + a^p$$

Since this is the total number of single faults at node 1, all must be masked.

Necessity

Let all single faults at node 1 be masked. Then $|F_1| = -1 + a^p$ because this is the total number of single faults at node 1. Let $q = |b_2 b_1(X)|$ and let $b_2 b_1(X) = \{y_1, y_2, \ldots, y_q\}$. If $q \geq 2$, then applying Theorem 2.4, Lemma 2.8.1, and Lemma 2.8.2, we have

$$|F_1| = -1 + \prod_{i=1}^{q} c_i^{d_i} < -1 + \left(\sum_{i=1}^{q} c_i \right)^{\sum_{i=1}^{q} d_i} \leq -1 + a^p$$

a contradiction of hypothesis. Thus $q = 1$, if b_2 is not a constant function, then Π_{b_2} must have at least two blocks and $c_1 = |b_2^{-1}(y_1)| < |W| = a$. Applying Theorem 2.4, Lemma 2.8.2 and this inequality we have

$$|F_1| = -1 + \prod_{i=1}^{1} c_i^{d_i} = -1 + c_1^{d_1} < -1 + a^p$$

a contradiction of hypothesis, and the theorem is proved.

It is immediately evident that if b_2 is a constant function, then the net function is also a constant function. Thus, it is possible to design a system that masks all faults in the front node only if we wish to realize a constant system function. For most functions, we can never hope to mask all faults in the front node.

We now show that with the exception of one singular case the condition of total fault masking at node 2 is unattainable, but first a supporting lemma is developed.

Lemma 2.9.1

The set of single faults at node 2 in a general two node system is empty iff $|Y| = 1$.

Proof

Sufficiency is trivial. If $Y = \{y\}$, then the only mapping $f_2: W \rightarrow Y$ is the constant mapping with $f_2(w) = y, \forall\, w \in W$. Hence for every fault of the form $f = (b_1, f_2)$, and for every $w \in W$, $f_2(w) = b_2(w) = y$. It follows (Definition 2.16) that there are no single faults at node 2.

Let the set of single faults at node 2 be empty, and assume that $|Y| \geq 2$, then $\exists$ two distinct elements of Y, y_1 and y_2. Also, $\exists$ two distinct functions from W into Y, f_1 and f_2; namely the two constant functions that map all of W onto y_1 and y_2 respectively. Both of these

cannot be equal to b_2 and hence either (b_1, f_1) or (b_1, f_2) must be a single fault at node 2 (Definition 2.16), contradicting the hypothesis.

It may now be observed that, in a general two node system with single faults having $|Y| = 1$, all single faults at node 2 are masked, detectable, locatable, and completely diagnosable because the set of single faults at node 2 is empty.

Theorem 2.9

In a general two node system with single faults having $|Y| \geq 2$, there is at least one detectable single fault at node 2.

Proof

Since b_1 is a function from X into W, we must have $r = |\mathcal{R}(b1)| \geq 1$ and $\bar{r} = |\overline{\mathcal{R}(b1)}| < |W| = a$. Since $s = |Y| \geq 2$, we must have

$$s^a - s^{\bar{r}} > 1$$

and the theorem follows directly from Corollary 2.5.1.

In the limit, node 2 is more restrictive with respect to fault masking than node 1 because the system must be degenerate to mask all faults at node 2, however, neither limit can be reached in circuits of practical value. Although these results are severely restrictive for two node systems, they readily generalize for application to larger systems. In the more general setting, we shall see that networks of great importance possess the required properties

especially in sections not producing system outputs. However, when analyzing the final output node, the general system form (see later section) reduces to a two node form. This serves to emphasize that in reasonable circuits we can never mask all faults in the final output node. For completeness, we state the conditions required to mask all faults in the two node system.

Theorem 2.10

In a general two node system with single faults, all faults are masked if and only if $|Y| = 1$.

Proof

To show sufficiency, let $|Y| = 1$ (i.e. $Y = \{y\}$). Then all single faults at node 2 are vacuously masked (Lemma 2.9.1). Since b_2 must be a constant function with $\mathcal{K}(b2) = \{y\}$, all single faults at node 1 are masked (Theorem 2.8). The 0-fault is also masked (Lemma 2.1.1), concluding the sufficiency proof.

To show necessity, let all faults be masked, then certainly all single faults at node 2 must be masked and $|Y| = 1$ (Theorem 2.9). This concludes the proof of the theorem.

If ease of maintenance is the primary goal, we may want to be able to detect all single faults in a subsystem. Our next theorem shows that this is possible in some types of networks. (cf. [25] for a discussion of "perfect" systems, i.e. α is 1-1 and onto.)

Theorem 2.11

In the general two node system with single faults, all single faults at node 1 are detectable iff

$$c_i = 1, \quad 1 \leq i \leq q$$

Proof: Sufficiency

Let $c_i = 1$, $1 \leq i \leq q$, then the number of single faults at node 1 that are detectable is given by Corollary 2.4.1

$$a^p - \prod_{i=1} c_i^{d_i} = a^p - 1$$

which is the total number of single faults at node 1, therefore, all must be detectable.

Necessity

Let all single faults at node 1 be detectable, then the number of detectable faults is

$$a^p - 1$$

Assume that $c_j \geq 2$ for some j, $1 \leq j \leq q$, then Corollary 2.4.1 states that the number of detectable single faults at node 1 is

$$a^p - \prod_{i=1}^{q} c_i^{d_i} = a^p - (c_j^{d_j}) \left(\prod_{\substack{i=1 \\ i \neq j}}^{q} c_i^{d_i} \right) < a^p - 1$$

a contradiction of hypothesis. This concludes the proof of the theorem.

Corollary 2.11.1

In the general two node system with single faults, if b_2 is a 1-1 function from W into Y, then all single faults at node 1 are detectable.

Proof

Immediate from Theorem 2.11.

Corollary 2,11.2

In the general two node system with single faults, if all single faults at node 1 are detectable, then $b_2 | \mathcal{R}(b_1)$ is a 1-1 function from $\mathcal{R}(b_1)$ into Y.

The previous theorem and its corollaries indicate that complete detectability of single faults at node 1 is achievable in non-trivial systems.

Although the condition of total fault masking at node 2 is unattainable there are a large number of networks in which all single faults at node 2 are detectable.

Theorem 2.12

In a general two node system with single faults having $|Y| \geq 2$, all single faults at node 2 are detectable if and only if b_1 is onto W.

Proof: Sufficiency

If b_1 is onto W, then $\mathcal{R}(b1) = W$ and $|\overline{\mathcal{R}(b1)}| = \bar{r} = 0$. The number of single faults at node 2 that are detectable is given by Corollary 2.5.1.

$$s^a - s^{\bar{r}} = s^a - 1$$

Since this is the total number of single faults at node 2, they must all be detectable.

Necessity

If all single faults at node 2 are detectable, then $s^a - s^{\bar{r}} = -1 + s^a$ or $s^{\bar{r}} = 1$. Since $s \geq 2$, we must have $\bar{r} = 0$, and b_1 is onto W. The theorem is proved.

Theorem 2.13

In a general two node system with single faults having $|Y| \geq 2$, all single faults are detectable if and only if

(1) b_1 is onto W

and

(2) b_2 is 1-1.

Proof

To show sufficiency, let (1) and (2) be true. Since b_2 is 1-1, all single faults at node 1 are detectable (Corollary 2.11.1). Since b_1 is onto W, all single faults at node 2 are detectable (Theorem 2.12).

Necessity follows directly from Theorems 2.12 and 2.11.

The limiting conditions for complete single fault detecting at node 1 and at node 2 are less restrictive than those for fault masking in that systems of practical interest exhibit the necessary properties. Even the requirements of Theorem 2.13 for complete single fault detecting in the system are obtainable in useful networks.

Also, unlike the complete fault masking requirements, the fundamental mapping theorem of algebra guarantees us that, for any given function g, there always exists a general two node system with single faults in which all single faults are detectable and whose function is g. This system is obtained by letting W be the set of equivalence classes of the equivalence relation $\underset{g}{\equiv}$. Then b_1 maps x into $[x]_g$ for every $x \in X$ and b_2 maps $[x]_g$ into g(x) for every $[x]_g$ in W. This is consistent because, by definition of $\underset{g}{\equiv}$, every element in $[x]_g$ maps onto the same element in Y.

Even though systems with 1 - 1 functions at node 2 are useful in some applications, such use is not widespread. A 1 - 1 function can at most perform a recoding of elements and may even be the identity function itself. On the other hand, systems with onto functions at node 1 are of universal application. We thus see that the requirements for detecting all single faults at node 1 (Corollary 2.11.1) are much more constraining in practical terms than are the requirements for detecting all single faults at node 2 (Theorem 2.12).

Since the existence of a two node system with all single faults detectable is answered by the fundamental mapping theorem of algebra, the most pressing questions seem to be

(1) How many such systems exist, and

(2) Can they be classified in a meaningful way.

These questions are currently being investigated.

Optimization in Two Node Systems

The important question to be considered here can be stated informally as follows:

> Among all general two node systems with single faults having net function g, which systems are "best" with respect to fault masking?

The answer, of course, depends upon the criteria one chooses to indicate quality, and upon any functional or system constraints that may be imposed. In the present discussion, quality will be indicated by the percentage of faults that are masked. If all faults are equally likely, and independent, then the percentage of faults masked is a direct indicator of system reliability in the traditional sense of probability of success. However, in a typical system, the 0-fault will have a much higher probability than any of the others. But, since the 0-fault is always masked, if all proper faults have the same probability, and if all faults are independent, then the percentage of faults masked is still a direct indicator of probability of success.

Typical constraints that are of interest include, among others, cost, size, weight, and number of interconnections. Each of these quantities is an indirect function of the size of the intermediate space of a two node general system in that, a larger intermediate space will result in a greater cost, greater weight, greater size and a greater number of interconnections for the system. Usually constraints in cost, size, weight, or number of interconnections can be transformed into a constraint on the size of the intermediate space. In this section, we will limit our attention to systems without constraints.

Let us now develop some notation for solving optimization problems. First let Z be the set of general binary two-node systems, i.e.

$$Z = \{D, S, F, b) \,|\, (D, S, F, b)$$

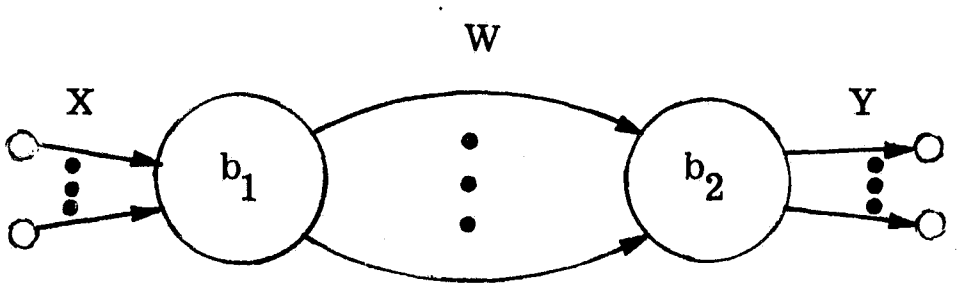
is a general two node system with single faults and

$$S = (S_1, \ldots, S_\ell) \text{ with } S_i = \{0, 1\},\ 1 \leq i \leq \ell\}. \qquad (2.1)$$

Let Z_g be the set of general binary two-node systems having net function g, that is,

$$Z_g = \{z \,|\, z \in Z \text{ and } \alpha(b) = g\}. \qquad (2.2)$$

A typical element of Z_g is shown in Figure 2.4.



C = (D, S, F, b) where D is an (n, m, 2, ℓ)-graph as shown.

$S = (S_1, \ldots, S_\ell)$ with $S_i = \{0, 1\}$, $1 \leq i \leq \ell$

$F = \{e \mid e \in E_C \text{ and } |K_e| \leq 1\}$

$\alpha(b) = b_2\, b_1 = g$

A Typical Element of Z_g

Figure 2.4

If $\mathbb{R}$ is the set of real numbers and z is any element of Z, define a mapping

$$P : Z \longrightarrow \mathbb{R} \tag{2.3}$$

P(z) = the percentage of faults masked in z (e.g. P(z) = .5 if 50% of the faults are masked)

From Theorem 2.6, the number of masked faults in a two node system with single faults is given by

$$\prod_{i=1}^{q} c_i^{d_i} + s^{\bar{r}} - 1 \tag{2.4}$$

where

$$q = |\mathcal{R}(\alpha(b))| = |\mathcal{R}(g)|$$

$$c_i = |b_2^{-1}(y_i)| \qquad \forall\, y_i \in \mathcal{R}(\alpha(b))$$

$$d_i = |b_2\, b_1^{-1}(y_i)| = |g^{-1}(y_i)| \qquad \forall\, y_i \in \mathcal{R}(\alpha(b))$$

$s = |Y|$ = size codomain of g

$$\bar{r} = |\overline{\mathcal{R}(b_1)}|$$

Since the total number of faults in a two node system with single faults is

$$|F| = a^p + s^a - 1 \tag{2.5}$$

where

$a = |W|$ = size of intermediate space of z

$p = |X|$ = size of input space of z

then the percentage of masked faults in system z is

$$P(z) = \frac{\prod_{i=1}^{q} c_i^{d_i} + s^{\bar{r}} - 1}{a^p + s^a - 1} \tag{2.6}$$

The optimization problem reduces to maximizing P(z) over the set Z_g. The values of p, s, and each element of the set $\{d_i \mid 1 \leq i \leq q\}$ in (2.6) are determined by the given function g. Only the values of a, $\bar{r}$, and each element in the set $\{c_i \mid 1 \leq i \leq q\}$ vary over the systems

of Z. However, not all of these variables are independent as we shall now demonstrate.

Certain conditions must be imposed on the system variables to meet the requirements of Definition 2.17. One is that all values of a, $\bar{r}$, $c_1, \ldots, c_q$ must be positive integers (except $\bar{r}$ can be 0). In order to solve such a problem in general, it would be necessary to resort to integer programming techniques. Fortunately, P(z) is a well behaved continuous function of each variable, allowing the application of continuous analysis techniques to find the best non-integer solution. The maximum integer solution is obtained by comparing values of P(z) for integers just below and just above the optimum value in each coordinate. While obtaining the optimum continuous solution, we will only require that each variable be ≥ 1 (except $\bar{r} \geq 0$).

Since each c_i, $1 \leq i \leq q$, is the cardinality of a different equivalence class of $\underset{b_2}{\equiv}$, and since equivalence classes are disjoint, we must have

$$\sum_{i=1}^{q} c_i \leq a \qquad (2.7)$$

If b_1 is onto W, then $\bar{r} = 0$ and since $\bar{r}$ must not be negative, 0 is a lower bound for $\bar{r}$. Since there must be at least q elements in $\mathcal{R}(b_1)$, there can be at most a - q elements in $\overline{\mathcal{R}(b_1)}$, and we have

$$0 \leq \bar{r} \leq a - q \qquad (2.8)$$

Summarizing, we must maximize

$$P(z) = \frac{\prod_{i=1}^{q} c_i^{d_i} + s^{\bar{r}} - 1}{a^p + s^a - 1} \tag{2.9}$$

with constraints

$$\sum_{i=1}^{q} c_i \leq a$$

$$0 \leq \bar{r} \leq a - q$$

$$a, \; q \geq 1$$

$$c_i \geq 1, \quad 1 \leq i \leq q$$

First let us deduce some properties of the optimum continuous solution. Consider the problem space as a (q+2)-dimensional cartesian product space

$$A \times \bar{R} \times C_1 \times C_2 \times \ldots \times C_q \tag{2.10}$$

where

$A = \{x | x \in \mathbb{R}, \; 1 \leq x\}$ = set of values for a

$\bar{R} = \{x | x \in \mathbb{R}, \; 0 \leq x\}$ = set of values for $\bar{r}$

$C_1 = \{x | x \in \mathbb{R}, \; 1 \leq x\}$ = set of values for c_1

$\vdots$

$C_q = \{x | x \in \mathbb{R}, \; 1 \leq x\}$ = set of values for c_q

Let

$$z^1 = (a^1, \bar{r}^1, c_1^1, c_2^1, \ldots, c_q^1) \qquad (2.11)$$

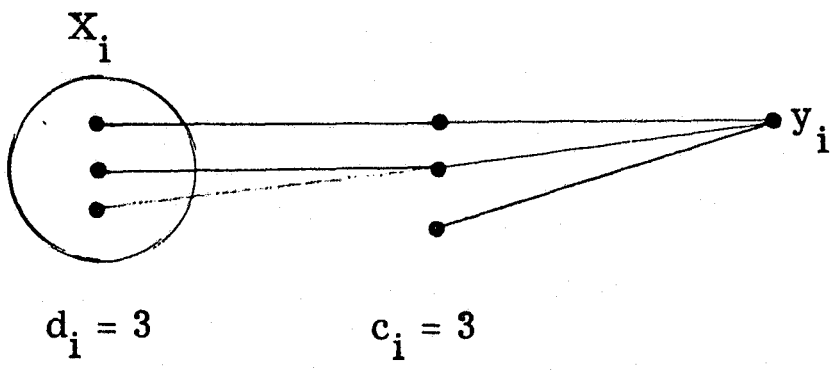
be any point in the space for which

$$\bar{r}^1 < a^1 - q$$

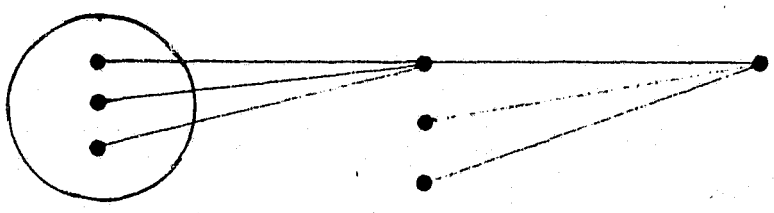
Then, when $s > 1$, there exists a point

$$z^2 = (a^1, a^1 - q, c_1^1, c_2^1, \ldots, c_q^1) \qquad (2.12)$$

such that $P(z^2) > P(z^1)$. (Simply map all d_i elements into any one of the c_i elements for each i, $1 \leq i \leq q$.) For example, if, in z^1, we have



then, in z^2 we would have



In other words, each c_i remains the same as in z^1. However, there are now only q elements in the $\mathcal{R}(b_1)$ and $a^1 - q$ elements in $\mathcal{R}(b_1)$. In Equation 2.9 we see that increasing $\bar{r}$ while keeping all other variables the same, will increase P(z) whenever $s > 1$.

If we let

$$M_g = \max_{z \epsilon Z_g} P(z) \tag{2.13}$$

and

$$S_g = \{z \,|\, z \in Z_g,\ P(z) = M_g\} \tag{2.14}$$

it follows that, when $s > 1$

$$z \in S_g \Longrightarrow \bar{r} = a - q \tag{2.15}$$

Since all optimum solutions for nontrivial networks lie in the hyperplane defined by Equation 2.15, we will restrict our attention to this region.

Now, let

$$z^3 = (a^3, \bar{r}^3, c_1{}^3, c_2{}^3, \ldots, c_q{}^3)$$

be a point in the problem space for which

$$\sum_{i=1}^{q} c_i{}^3 < a^3 \tag{2.16}$$

Then the point

$$z^4 = (a^3, \bar{r}^3, c_1{}^3 + a^3 - \sum_{i=1}^{q} c_i{}^3, c_2{}^3, \ldots, c_q{}^3) \tag{2.17}$$

has the property that

$$P(z^4) > P(z^3)$$

It follows that any point that maximizes Equation 2.9 must have the property

$$\sum_{i=1}^{q} c_i = a \tag{2.18}$$

Since all optimum solutions lie in the hyperplane defined by Equation 2.18, let us restrict our search to this region. The problem may now be restated, for the case when $s > 1$, as follows:

Maximize

$$P(z) = \frac{\prod_{i=1}^{q} c_i^{d_i} + s^{\sum_{i=1}^{q} c_i - q} - 1}{\left(\sum_{i=1}^{q} c_i\right)^{p} + s^{\left(\sum_{i=1}^{q} c_i\right)} - 1} \tag{2.19}$$

with the constraint

$$c_i \geq 1, \ 1 \leq i \leq q$$

We now assume that the -1 term in both the numerator and denominator of Equation 2.19 can be neglected. In nearly all systems, this assumption will be valid. When $s > 1$, the problem reduces to

Maximize

$$P(z) = \frac{\prod_{i=1}^{q} c_i^{d_i} + s^{\sum_{i=1}^{q} c_i - q}}{\left(\sum_{i=1}^{q} c_i\right)^{p} + s^{\left(\sum_{i=1}^{q} c_i\right)}} \tag{2.20}$$

with $c_i \geq 1, \ 1 \leq i \leq q$.

Example 2.1

Let g be described by the following function table,

x_1	x_2	y_1
0	0	1
0	1	0
1	0	0
1	1	0

then s = 2, p = 4, $d_1 = 1$ and $d_2 = 3$. The graph of Figure 2.5 contains two cross-sections of the P(z) surface. Curve 1 is the trace of P(z) in the plane $c_1 = 1.442695\ldots$. Curve 2 is the trace of P(z) in the plane $c_2 = 4.328\ldots$.

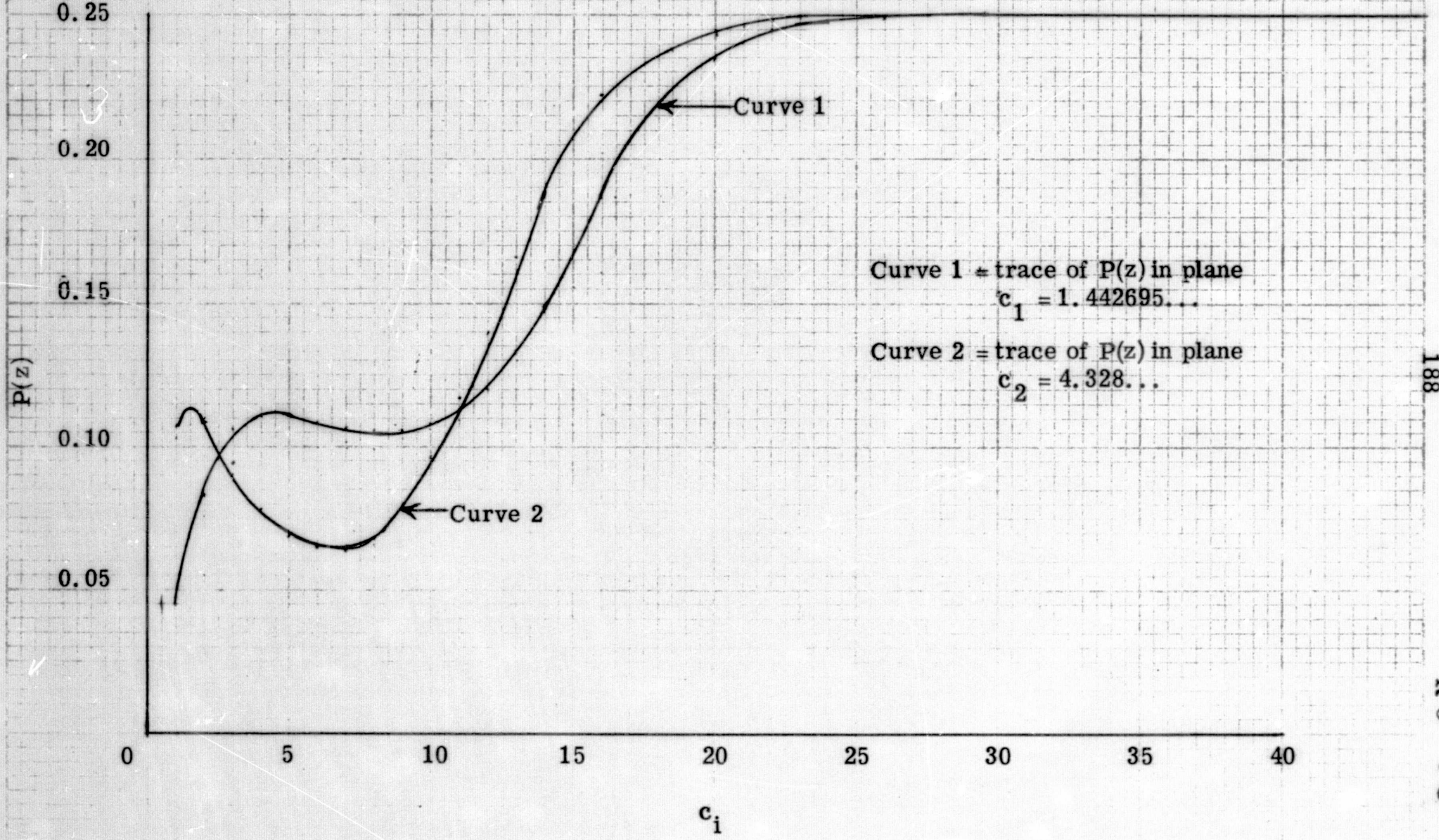
We notice immediately that P(z) asymptotically approaches 0.25 in both planes as the value of a increases without bound. This limit is one candidate for the maximum value of P(z) over the set S_g.

We also notice that P(z) has a peak in both planes at a low value of a. This point will be examined in detail in the next section. Every function displays a critical point at a low value of a and asymptotically approaches a final value as a increases without limit.

Increasing the size of a corresponds roughly to increasing the amount of redundancy in the system. We see that P(z) has a comparatively high value at a low level of redundancy, and actually drops off for a time as the redundancy is further increased. Finally, P(z) begins to rise again and asymptotically approaches a final value as the redundancy becomes very large.

CROSS SECTION - 10 SQUARES TO INCH

Cross Sections of P(z) Surface for Example 2.1



Curve 1 = trace of P(z) in plane $c_1 = 1.442695...$

Curve 2 = trace of P(z) in plane $c_2 = 4.328...$

Figure 2.5

RO 70-456

The first step in the analysis of (2.20) is to determine the critical points of the function P(z). Employing the method of partial derivatives, we can show that a necessary and sufficient condition for $z = (a, \bar{r}, c_1, \ldots, c_q)$ to be a critical point of P(z) is that

$\forall\, k, 1 \leq k \leq q$

$$\left[\frac{d_k}{c_k} - \log_e s\right] \left[\left(\prod_{i=1}^{q} d_i^{d_i}\right) - \frac{p^p}{s^q}\right] = 0 \qquad (2.21)$$

The right hand term is dependent entirely upon the given function. If it is zero, then the slope of the P(z) curve at any point with $d_i/c_i = d_j/c_j$ for all i, j such that $1 \leq i \leq q$, $1 \leq j \leq q$ will be zero. A discussion of networks with this property is deferred to a later section. For functions with the following properties

(1) $s > 1$

and

(2) $\prod_{i=1}^{q} d_i^{d_i} \neq \frac{p^p}{s^q}$

the critical point of interest occurs when

$$c_k = \frac{d_k}{\log_e s} \quad \forall\, k, \quad 1 \leq k \leq q \qquad (2.22)$$

Substituting 2.22 into 2.18 and 2.15 we find that

$$a = \sum_{i=1}^{q} c_i = \frac{p}{\log_e s} \qquad (2.23)$$

$$\bar{r} = a - q = \frac{p}{\log_e s} - q \qquad (2.24)$$

For the example in Figure 2.5, it appears that the critical point may be a local maximum of the functional. If this could be proved to be true in general, then we would have positive proof that adding redundancy beyond a critical value will always reduce the fault masking ratio. Of course, by greatly increasing the amount of redundancy, we will eventually approach the final value. The question then becomes one of determining whether the critical ratio is greater, smaller or equal to the limiting value. All three conditions are possible in non-trivial systems.

Let us now turn our attention to the critical value.

Lemma 2.14.1

In the Euclidean space, $C_1 \times C_2 \times \ldots \times C_q$, with each $C_i = \mathbb{R}$ ($\mathbb{R}$ is the set of real numbers), the point

$$c_j = \frac{d_j}{\log_e s} \qquad 1 \leq j \leq q$$

is a local maximum of the hypersurface

$$P(z) = \frac{\prod_{i=1}^{q} c_i^{d_i} + s^{\sum_{i=1}^{q} c_i - q}}{\left(\sum_{i=1}^{q} c_i\right)^p + s^{\sum_{i=1}^{q} c_i}}$$

when $s > 1$ and $\prod_{i=1}^{q} d_i^{d_i} > \frac{p^p}{s^q}$.

Proof:

By the discussion immediately preceding Lemma 2.14.1 the point

$$c_j = \frac{d_j}{\log_e s} \qquad 1 \leq j \leq q$$

is a critical point of the given hypersurface. To show that it is also a local maximum, it is sufficient to show that the matrix of second partial derivatives of P(z) is negative definite [28], i.e.

$$\begin{bmatrix} f_{11} & f_{12} & f_{13} & \cdots & f_{1q} \\ f_{21} & f_{22} & f_{23} & \cdots & f_{2q} \\ f_{31} & f_{32} & f_{33} & \cdots & f_{3q} \\ \vdots & \vdots & \vdots & & \vdots \\ f_{q1} & f_{q2} & f_{q3} & \cdots & f_{qq} \end{bmatrix} \tag{2.25}$$

is negative definite where $f_{ij} = \left.\dfrac{\partial^2 P(z)}{\partial c_i \, \partial c_j}\right|_{c_k = \frac{d_k}{\log_e s}} \quad 1 \leq k \leq q$

The matrix of (2.25) is negative definite [28], if and only if

$$\begin{vmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \vdots & \vdots & & \\ f_{n1} & f_{n2} & \cdots & f_{nn} \end{vmatrix} \quad \text{is} \quad \begin{array}{l} < 0 \text{ for n odd}, \quad 1 \leq n \leq q \\ > 0 \text{ for n even}, \quad 1 \leq n \leq q \end{array} \tag{2.26}$$

Using the identity

$$\prod_{i=1}^{q} c_i^{d_i} = \prod_{i=1}^{q} \left(\frac{d_i}{\log_e s}\right)^{d_i} = \prod_{i=1}^{q} \left(\frac{1}{\log_e s}\right)^{d_i} \binom{d_i}{d_i} = \left(\frac{1}{\log_e s}\right)^{p} \prod_{i=1}^{q} d_i^{d_i} \tag{2.27}$$

at the critical points, we find that

$$f_{jj} = \frac{(\log_e s)^{2-p}}{p d_j \left[\left(\frac{p}{\log_e s}\right)^{p} + s^{\frac{p}{\log_e s}}\right]^2} \left\{ \frac{(d_j - p)\,(p^p)\left(\prod_{i=1}^{q} d_i^{d_i}\right)}{(\log_e s)^{p}} + s^{\frac{p}{\log_e s}} \left[\frac{(d_j)(p^p)}{s^q} - p \prod_{i=1}^{q} d_i^{d_i} \right] \right\} \tag{2.28}$$

for every j, $1 \leq j \leq q$.

And that, when $j \neq k$

$$f_{jk} = \frac{\frac{p^{p-1}}{(\log_e s)^{p-2}} \left[\frac{\prod_{i=1}^{q} d_i^{d_i}}{(\log_e s)^{p}} + \frac{s^{\frac{p}{\log_e s}}}{s^q} \right]}{\left[\left(\frac{p}{\log_e s}\right)^{p} + s^{\frac{p}{\log_e s}} \right]^2} \tag{2.29}$$

Note that f_{jk} is independent of j and k when $j \neq k$. Thus we conclude that all off-diagonal terms of matrix (2.25) are equal which will

help us to evaluate the determinant of (2.26). The determinant of (2.26) then becomes

$$\begin{vmatrix} f_{11} & F & \cdots & F \\ F & f_{22} & \cdots & F \\ \vdots & \vdots & & \vdots \\ F & F & \cdots & f_{nn} \end{vmatrix} \quad \text{with } F = f_{jk} \quad (j \neq k) \qquad (2.30)$$

Subtracting column 1 from each of the other columns does not alter the determinant value.

$$\begin{vmatrix} f_{11} & F - f_{11} & F - f_{11} & \cdots & F - f_{11} \\ F & f_{22} - F & 0 & \cdots & 0 \\ F & 0 & f_{33} - F & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ F & 0 & 0 & & f_{nn} - F \end{vmatrix} \qquad (2.31)$$

Multiplying row 2 by $\left(\frac{F - f_{11}}{f_{22} - F}\right)$ and subtracting from row 1 does not alter the value of the determinant

$$\begin{vmatrix} f_{11} - F\left(\frac{F-f_{11}}{f_{22}-F}\right) & 0 & F - f_{11} & \cdots & F - f_{11} \\ F & f_{22} - F & 0 & \cdots & 0 \\ F & 0 & f_{33} - F & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ F & 0 & 0 & \cdots & f_{nn} - F \end{vmatrix} \qquad (2.32)$$

Similarly, multiplying row j ($3 \leq j \leq n$) by $\left(\frac{F-f_{11}}{f_{jj}-F}\right)$ and subtracting from row 1 does not change the value of the determinant.

$$\begin{vmatrix} B_n & 0 & 0 & \cdots & 0 \\ F & f_{22}-F & 0 & \cdots & 0 \\ F & 0 & f_{33}-F & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ F & 0 & 0 & \cdots & f_{nn}-F \end{vmatrix} \qquad (2.33)$$

with $B_n = f_{11} - F \sum_{j=2}^{n} \left(\frac{F-f_{11}}{f_{jj}-F}\right)$.

The determinant of (2.33) is now diagonal and has value D_n given by

$$D_n = B_n \prod_{j=2}^{n} (f_{jj} - F) \qquad (2.34)$$

Recall that $F = f_{jk}$ where $j \neq k$. From (2.28) and (2.29) we find that

$$f_{jj} - F = \frac{\left(1 - \frac{1}{d_j}\right)\left(\prod_{i=1}^{q} d_i\right)}{(\log_e s)^{p-2}\left[\left(\frac{p}{\log_e s}\right)^p + s^{p/\log_e s}\right]} \qquad (2.35)$$

Note that $f_{jj} - F$ is negative for all values of j $(1 \leq j \leq q)$, therefore the algebraic sign of D_n depends only upon the value of n and the sign of B_n. From (2.33)

$$B_n = f_{11} + F \sum_{j=2}^{n} \left(\frac{f_{11}-F}{f_{jj}-F}\right) \qquad (2.36)$$

From (2.35)

$$\frac{f_{11} - F}{f_{jj} - F} = \frac{d_j}{d_1} \qquad (2.37)$$

Thus, since $\sum_{i=1}^{n} d_i = p - \sum_{i=n+1}^{q} d_i$

$$B_n = f_{11} + F \sum_{j=2}^{n} \frac{d_j}{d_1} = f_{11} + F\left(\frac{p - d_1 - \sum_{i=n+1}^{q} d_i}{d_1}\right) \qquad (2.38)$$

Using (2.28) and (2.29),

$$B_n = \frac{\frac{1}{d_1}\left\{\left(\frac{p}{s^{\log_e s}}\right)\left[\frac{p^p}{s^q} - \prod_{i=1}^{q} d_i^{d_i}\right] - \left(\sum_{i=n+1}^{q} d_i\right)\left(p^{p-1}\right)\left[\frac{\prod_{i=1}^{q} d_i^{d_i}}{(\log_e s)^p} + \frac{s^{p/\log_e s}}{s^q}\right]\right\}}{(\log_e s)^{p-2}\left[\left(\frac{p}{\log_e s}\right)^p + s^{p/\log_e s}\right]^2} \qquad (2.39)$$

We see that B_n is negative for all n such that $1 \leq n \leq q$ since

$$\frac{p^p}{s^q} < \prod_{i=1}^{q} d_i^{d_i} \tag{2.40}$$

Since each term in $\prod_{i=2}^{n} (f_{jj} - F)$ is negative we conclude that when the inequality of (2.40) is satisfied D_n is negative for odd n and positive for even n. By earlier discussion, this is sufficient for the critical point

$$c_i = \frac{d_i}{\log_e s} \qquad 1 \leq i \leq q$$

to be a local maximum. This proves Lemma 2.14.1.

In the previous section, critical points that lead to local maxima of P(z) in some systems were derived. We now turn our attention to the limit of P(z) as the size of the intermediate space increases without limit. Recall that

$$P(z) = \frac{\prod_{i=1}^{q} c_i^{d_i} + s^{\sum_{i=1}^{q} c_i - q} - 1}{\left(\sum_{i=1}^{q} c_i\right)^p + s^{\sum_{i=1}^{q} c_i} - 1} \tag{2.41}$$

Choosing an arbitrary coordinate, c_j, along which to take the limit, then when $s > 1$

$$\lim_{c_j \to \infty} P(z) = \lim_{c_j \to \infty} \frac{\left(\prod_{i=1}^{q} c_i^{d_i} \right) + s^{\sum_{i=1}^{q} c_i - q} - 1}{\left(\sum_{i=1}^{q} c_i \right)^p + s^{\sum_{i=1}^{q} c_i} - 1} = \frac{\infty}{\infty} \qquad (2.42)$$

Using L'Hospital's rule [28] (p+1) times

$$\lim_{c_j \to \infty} P(z) = \lim_{c_j \to \infty} \frac{(\log_e s)^{p+1} \left(s^{\sum_{i=1}^{q} c_i - q} \right)}{(\log_e s)^{p+1} \left(s^{\sum_{i=1}^{q} c_i} \right)} = \frac{1}{s^q} \qquad (2.43)$$

At this point, it is possible to solve the unconstrained optimization problem. The function P(z) will attain its absolute maximum either at the critical point

$$c_i = \frac{d_i}{\log_e s} \quad 1 \leq i \leq q$$

at a boundary point, or in the limit as $c_i \to \infty$. In the latter case, we can bring P(z) as close as we wish to its absolute maximum by choosing sufficiently large values of the c_i variables. It is noted that this limit is approached very rapidly so that diminishing returns occur at relatively low values of a.

Let

$$P_C = P(z)\Big|_{c_i = \frac{d_i}{\log_e s}} \qquad 1 \le i \le q \tag{2.44}$$

and $$P_L = \lim_{c_i \to \infty} P(z). \tag{2.45}$$

then we can prove the following helpful lemma.

Lemma 2.14.2

In a two node general system $P_L < P_C$ if and only if

$$\left(\frac{p^p}{s^q} < \prod_{i=1}^{q} d_i^{d_i} \right) .$$

Proof:

To show necessity, let $P_L < P_C$ in a general two node system. Then from (2.20), (2.27) and (2.43)

$$\frac{1}{s^q} < \frac{\left(\frac{1}{\log_e s}\right)^p \left(\prod_{i=1}^{q} d_i^{d_i} \right) + \frac{s^{\frac{p}{\log_e s}}}{s^q}}{\frac{p^p}{\left(\log_e s\right)^p} + s^{\frac{p}{\log_e s}}} \tag{2.46}$$

Since s, q, and p are all positive integers, we can multiply both sides of inequality (2.46) by s^q and by the denominator of the right member without destroying the inequality.

$$\frac{p^p}{(\log_e s)^p} + s^{\frac{p}{\log_e s}} < (s^q)\left(\frac{1}{\log_e s}\right)^p \left(\prod_{i=1}^{q} d_i^{d_i}\right) + s^{\frac{p}{\log_e s}} \qquad (2.47)$$

Subtracting $s^{p/\log_e s}$ from both sides of the inequality of (2.47)

$$\frac{p^p}{(\log_e s)^p} < \frac{(s^q)\left(\prod_{i=1}^{q} d_i^{d_i}\right)}{(\log_e s)^p} \qquad (2.48)$$

Since s and p are positive integers, multiplying both sides of (2.48) by $(\log_e s)^p$ maintains the inequality.

$$p^p < (s^q)\left(\prod_{i=1}^{q} d_i^{d_i}\right) \qquad (2.49)$$

Since s and q are positive integers, dividing both sides of (2.49) by s^q will preserve the inequality.

$$\frac{p^p}{s^q} < \prod_{i=1}^{q} d_i^{d_i} \qquad (2.50)$$

This proves necessity. As each step of the necessity proof is reversible, sufficiency is shown by starting with (2.50) and working backwards to (2.46).

Lemma 2.14.3

In a two node general system $P_L > P_C$ if and only if

$$\left(\frac{p^p}{s^q} > \prod_{i=1}^{q} d_i^{d_i}\right) .$$

Proof

Same as Lemma 2.14.2 with $>$ replacing $<$.

Lemma 2.14.4

In a two node general system $P_L = P_C$ if and only if

$$\left(\frac{p^p}{s^q} = \prod_{i=1}^{q} d_i^{d_i} \right)$$

Proof

Let $P_L = P_C$. Then Lemma 2.14.2 implies that

$$\left(\frac{p^p}{s^q} \not< \prod_{i=1}^{q} d_i^{d_i} \right)$$

and Lemma 2.14.3 implies that

$$\frac{p^p}{s^q} \not> \prod_{i=1}^{q} d_i^{d_i}$$

We then must have

$$\frac{p^p}{s^q} = \prod_{i=1}^{q} d_i^{d_i}$$

A similar proof establishes the sufficiency of the lemma.

Thus far, we have considered only the critical points and the upper boundary. One more possibility exists for the maximum value of P(z), namely, the lower boundary. Close scrutiny reveals that the mathematical functions of the $c_i = 1$ boundaries are very difficult to analyze. Therefore, at a slight inconvenience,

let us expand the problem solution space to include the entire positive "hyperquadrant". (Practically there is no inconvenience since a local maximum will not usually occur at an integral point anyway. We will always have to check the closest integral solution, which in the case of a local maximum in the expanded range would be on one of the $c_i = 1$ boundaries.) The new boundaries will be characterized by $c_j = 0$ for some subset of the system variables. On any boundary, P(z) becomes

$$P(z) = \frac{s^{\sum_{i=1}^{q} c_i - q}}{(\sum_{i=1}^{q} c_i)^p + s^{\sum_{i=1}^{q} c_i}} \qquad (2.51)$$

It is obvious that, on a boundary

$$P(z) < \frac{1}{s^q} = P_L \qquad (2.52)$$

except at the origin, when $P(z) = \frac{1}{s^q} = P_L$. As the origin is not of interest to us, we will include in our analysis, the closest point of interest, namely the point with $c_i = 1$ for every i, $1 \leq i \leq q$. However, all other boundary points can be ignored since, by (2.51), we can always obtain a larger value of P(z) by choosing a z such that $P(z) \longrightarrow P_L$.

For a given function g, let

$$P_1 = P(z_1) \tag{2.53}$$

where $z_1 = (q, 0, 1, 1, \ldots, 1)$

i.e., P_1 is the value of P(z) at the lower boundary of the optimization region.

Lemma 2.14.5

For a given function g, if $q > 1$,

$$P_1 < P_L$$

Proof

$$a = \sum_{i=1}^{q} c_i = \sum_{i=1}^{q} 1 = q$$

$$\bar{r} = a - q = q - q = 0$$

From (2.19) and the fact that $q > 1$

$$P_1 = \frac{\prod_{i=1}^{q} (1)^{d_i} + s^0 - 1}{q^p + s^q - 1} = \frac{1}{q^p + s^q - 1} < \frac{1}{s^q} = P_L \tag{2.54}$$

Lemma 2.14.6

If g is a constant function (i.e. q=1)

$$P_1 = P_L$$

Proof

Let g be a constant function (q=1), then, at the lower boundary

$$p = \sum_{i=1}^{q} d_i = d_1$$

$$q = a = \sum_{i=1}^{q} c_i = c_1 = 1$$

$$\bar{r} = a - q = 0$$

$$P_1 = \frac{\prod_{i=1}^{1} c_i^{d_i} + s^0 - 1}{s^q + q^p - 1} = \frac{(1)^p + 1 - 1}{s^q + 1 - 1} = \frac{1}{s^q} = P_L$$

Recall that we are attempting to maximize

$$P(z) = \frac{\prod_{i=1}^{q} c_i^{d_i} + s^{\sum_{i=1}^{q} c_i - q}}{\left(\sum_{i=1}^{q} c_i\right)^p + s^{\sum_{i=1}^{q} c_i}} \qquad (2.55)$$

over all two node general systems Z_g. Also recall that we are representing the solution space as a cartesian product space (see (2.10)).

$$A \times \bar{R} \times C_1 \times \ldots \times C_q \qquad (2.56)$$

The results of the preceding section may now be utilized to state solutions to the unconstrained problem.

Theorem 2.14

If function g is class I, i.e. g has the properties

$$\frac{p^p}{s^q} < \prod_{i=1}^{q} d_i^{d_i} \text{ and } s > 1$$

then the optimum two node general system realizing g is

$$S_g = \left\{\left(\frac{p}{\log_e s}, \frac{p}{\log_e s} - q, \frac{d_1}{\log_e s}, \frac{d_2}{\log_e s}, \ldots, \frac{d_q}{\log_e s}\right)\right\}$$

with

$$M_g = \frac{\prod_{i=1}^{q} d_i^{d_i} + (\log_e s)^p \left(s^{\frac{p}{\log_e s}} - q\right)}{(p^p) + (\log_e s)^p \left(s^{\frac{p}{\log_e s}}\right)}$$

Proof

By Lemmas 2.14.2, 2.14.5, and 2.14.6 $P_1 \leq P_L < P_C$. Since the optimum value of P(z) must be either a critical point (P_C) or on a boundary (P_L or P_1) the theorem is proved. Lemma 2.14.1 verifies that the critical point is indeed a local maximum in this type of system.

Example 2.2

Let g be described by the following function table.

x_1	x_2	y_1	y_2
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

then $s = 4$, $p = 4$, $q = 2$, $d_1 = 3$, and $d_2 = 1$.

$$\frac{p^p}{s^q} = \frac{4^4}{4^2} = 4^2 = 16 < \prod_{i=1}^{q} d_i^{d_i} = (1)^1(3)^3 = 27$$

$$a = \frac{p}{\log_e s} = \frac{4}{\log_e 4} = 2.88539 \approx 3$$

$$\bar{r} = a - q \approx 3 - 2 = 1$$

$$c_1 = \frac{d_1}{\log_e s} = \frac{3}{\log_e 4} = 2.16404 \approx 2$$

$$c_2 = \frac{d_2}{\log_e s} = \frac{1}{\log_e 4} = 0.72135 \approx 1$$

$$S_g = \{(3, 1, 2, 1)\}$$

$$M_g = \frac{\prod_{i=1}^{q} c_i^{d_i} + s^{\bar{r}} - 1}{\left(\sum_{i=1}^{q} c_i\right)^p + s^{\sum_{i=1}^{q} c_i}} = \frac{(1)^1(2)^3 + 4^1 - 1}{(2+1)^4 + 4^{(2+1)}}$$

$$= \frac{11}{81+64} = \frac{11}{145}$$

$$= 0.0758$$

Hence, about 7.6% of all faults are masked in the optimum system. Figure 2.6 shows a cross section of P(z) parallel to the C_1 axis. Note that the maximum point and the maximum value of P(z) is shifted slightly from that predicted by our theory. This is apparent

Cross Section of P(z) Parallel to C_1 axis for Example 2.2

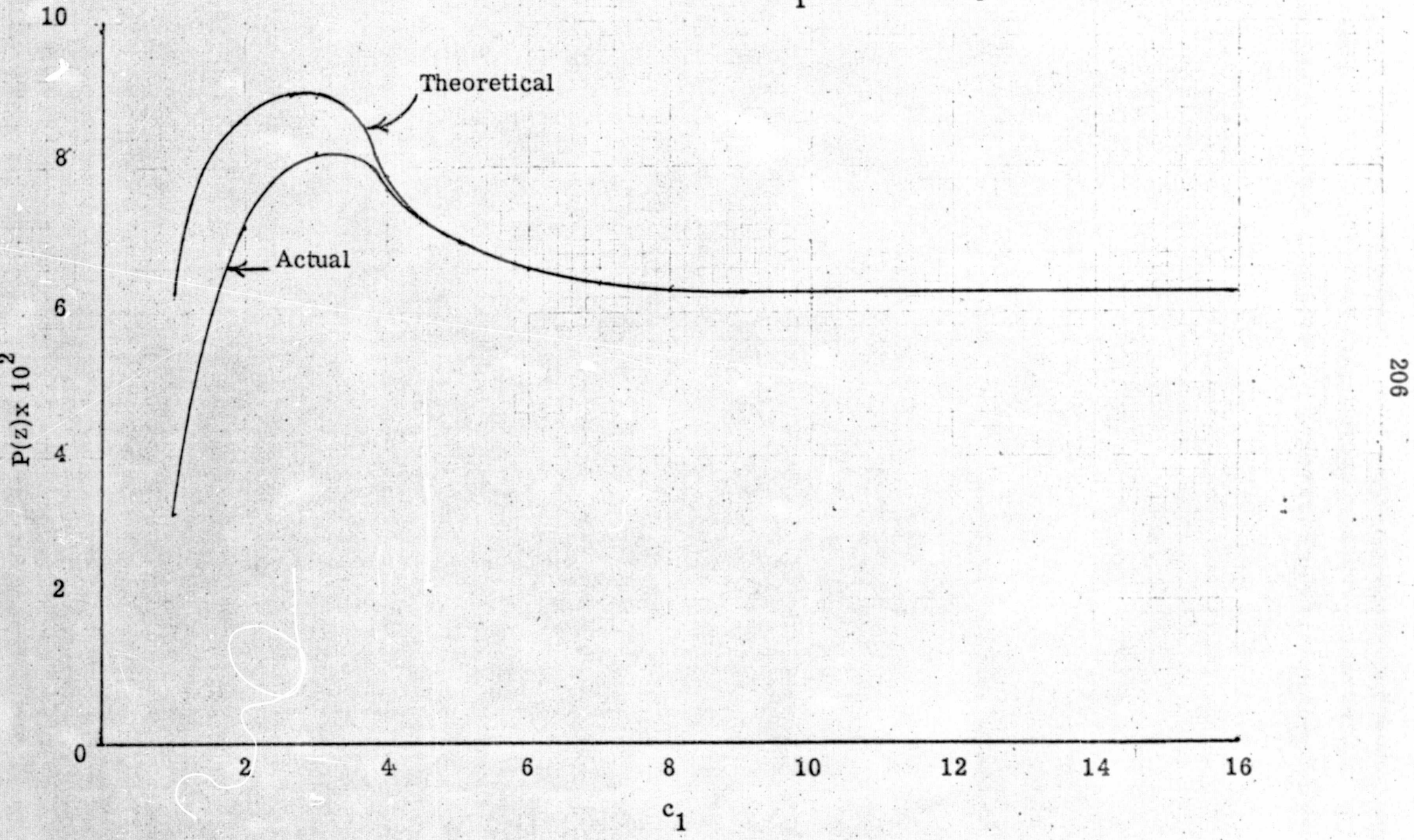


Figure 2.6

because the network is small enough that the two -1 terms neglected in (2.19) have a small effect. The number of faults masked in the optimum system is 10 out of 124. In large systems, this shifting effect will be negligible.

Theorem 2.15

If function g is class II, i.e. g has the properties

$$\frac{p^p}{s^q} > \prod_{i=1}^{q} d_i^{d_i} \quad \text{and } s > 1$$

then the optimum two node general system realizing g occurs in the limit as some $c_i \to \infty$. This limit may be approached as closely as desired with the limiting value of P(z) being

$$M_g = \frac{1}{s^q}$$

Proof

If $q = 1$, then $d_1 = p$ and since $s > 1$;

$$\frac{p^p}{s^q} = \frac{p^p}{s} < p^p = \prod_{i=1}^{q} d_i^{d_i}$$ a contradiction of hypothesis.

Therefore, $q > 1$. By Lemmas 2.14.3 and 2.14.5, $P_1 < P_L > P_C$, and since P(z) must have its maximum value either at a critical point (P_C) or on a boundary (P_L or P_1) the theorem is proved. Example 2.1 illustrates Theorem 2.15.

$$\text{If } \frac{p^p}{s^q} = \prod_{i=1}^{q} d_i^{d_i} \tag{2.57}$$

a separate analysis is required because P(z) has many critical points. When $s > 1$, the analysis remains the same as the general case through (2.21). Recall that the following conditions on the optimum solution have already been found:

$$\bar{r} = a - q$$

$$a = \sum_{i=1}^{q} c_i$$

$$\frac{d_k}{c_k} = \frac{d_j}{c_j} \qquad \forall\, j, k \text{ with } 1 \leq j \leq q, \quad 1 \leq k \leq q$$

This last condition implies that

$$c_i = \left(\frac{c_k}{d_k}\right) d_i \qquad \forall\, i,\ 1 \leq i \leq q; \quad \forall\, k,\ 1 \leq k \leq q \tag{2.58}$$

By (2.21), each point that satisfies (2.58) is a critical point. Using (2.58), we find that

$$\sum_{i=1}^{q} c_i = \sum_{i=1}^{q} \left(\frac{c_k}{d_k}\right) d_i = \frac{c_k}{d_k} \sum_{i=1}^{q} d_i = \left(\frac{c_k}{d_k}\right)(p) \qquad (2.59)$$

and that

$$\prod_{i=1}^{q} c_i^{d_i} = \prod_{i=1}^{q} \left(\frac{c_k}{d_k} d_i\right)^{d_i} = \prod_{i=1}^{q} \left(\frac{c_k}{d_k}\right)^{d_i} (d_i^{d_i})$$

$$= \left(\frac{c_k}{d_k}\right)^{p} \prod_{i=1}^{q} d_i^{d_i} \qquad (2.60)$$

Using (2.57) we have

$$\prod_{i=1}^{q} c_i^{d_i} = \left(\frac{c_k}{d_k}\right)^{p} \frac{p^p}{s^q} = \left(\frac{1}{s^q}\right) \left(\frac{p\, c_k}{d_k}\right)^{p} \qquad (2.61)$$

Substituting (2.59) and (2.61) into (2.20), we have

$$P(z) = \frac{\left(\frac{1}{s^q}\right)\left(\frac{p\, c_k}{d_k}\right)^{p} + \left(\frac{1}{s^q}\right)\left(s^{\frac{p\, c_k}{d_k}}\right)}{\left(\frac{p\, c_k}{d_k}\right)^{p} + s^{\frac{p\, c_k}{d_k}}}$$

$$= \frac{1}{s^q} = P_L \qquad (2.62)$$

Thus, in this case, there are an infinite number of systems that produce the maximum possible fault masking ratio. Stated as a theorem

Theorem 2.16

If a function g is class III, i.e. g has the properties

$$\frac{p^p}{s^q} = \prod_{i=1}^{q} d_i^{d_i} \quad \text{and} \quad s > 1$$

then the optimum set of systems that realize g is

$$S_g = \{(pk,\ pk-q,\ d_1k,\ d_2k,\ \ldots,\ d_qk) \mid pk \geq q\}$$

with

$$M_g = \frac{1}{s^q}$$

Example 2.3

As an example, consider the function g as described by the following table

x_1	x_2	y_1	y_2
0	0	0	1
0	1	1	1
1	0	1	0
1	1	0	0

Here, p=4, s=4, q=4, $d_1 = d_2 = d_3 = d_4 = 1$ so that

$$\frac{p^p}{s^q} = \frac{4^4}{4^4} = 1 = \prod_{i=1}^{q} d_i^{d_i}$$

Theoretically, P(z) should have the same value, namely

$$\frac{1^p}{s^q} = \frac{1}{4^4} = 3.91 \times 10^{-3}$$

at every point where $\frac{c_i}{d_i}$ is a constant for all i, $1 \leq i \leq q$.

However, due to the approximation of (2.20), small networks such

as this one will experience lower values of P(z) for small values of c_i/d_i, as shown by the curves in Figure 2.7. The actual value is 1/2 the theoretical value when $c_i/d_i = 1$ but nearly equal to the theoretical value when $c_i/d_i = 2$. For larger values of c_i/d_i, the expected behavior is observed.

Example 2.4

For larger systems, the actual deviation from theoretical values is completely negligible even for low values of c_i/d_i. For example, consider the function g described by the table

x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4	y_5
0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1	0
0	0	1	0	0	1	1	0	0
0	0	1	1	1	0	0	1	0
0	1	0	0	1	1	0	0	0
0	1	0	1	0	0	0	1	1
0	1	1	0	0	0	1	1	0
0	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	0	0
1	0	0	1	0	0	1	0	0
1	0	1	0	0	1	1	1	0
1	0	1	1	0	0	1	1	0
1	1	0	0	0	1	1	1	0
1	1	0	1	1	1	0	0	0
1	1	1	0	0	1	1	0	0
1	1	1	1	0	1	1	1	0

For this function, $s = 32$, $p = 16$, $q = 10$, $d_1 = 4$, $d_2 = d_3 = d_4 = 2$, $d_5 = d_6 = d_7 = d_8 = d_9 = d_{10} = 1$, so that

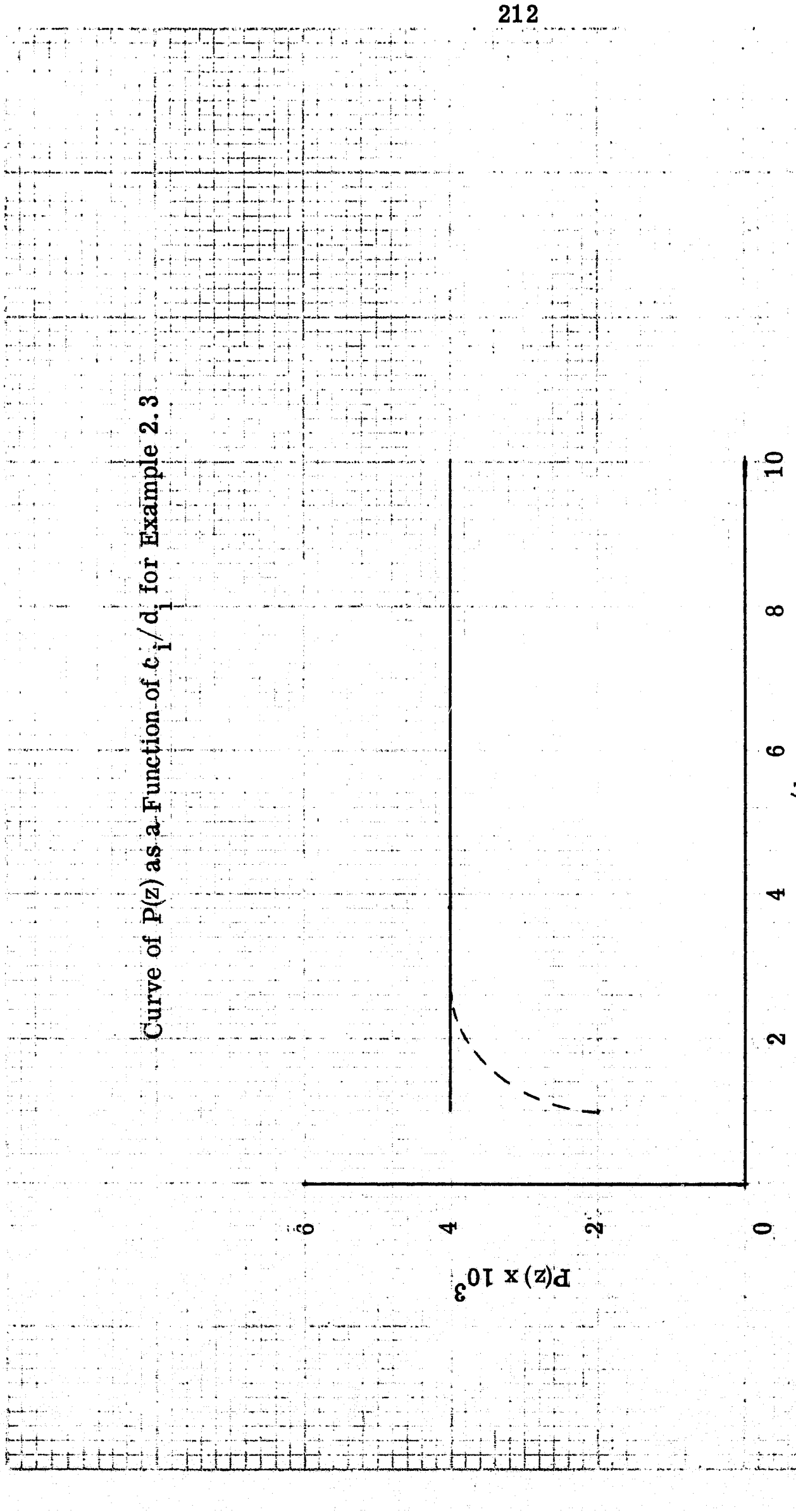

Curve of P(z) as a Function of c_i/d_i for Example 2.3
$P(z) \times 10^3$
0
2
4
6
c_i/d_i
8
10

Figure 2.7

$$\frac{p^p}{s^q} = \frac{16^{16}}{32^{10}} = \frac{2^{64}}{2^{50}} = 2^{14}$$

and

$$\prod_{i=1}^{10} d_i^{d_i} = (4^4)(2^2)(2^2)(2^2) = 2^{14}$$

Theoretically, P(z) should have the same value for any system with c_i/d_i a constant namely

$$P(z) = \frac{1}{s^q} = \frac{1}{32^{10}} = \frac{1}{2^{50}} \approx \frac{1}{1.126 \times 10^{15}} \approx 8.88 \times 10^{-16}$$

In this case, when $c_i/d_i = 1$, the value of P(z) is the same as $\frac{1}{s^q}$ up to 8 significant digits. The exact values (to 20 significant digits) is given in the following table.

c_i/d_i	P(z)
1	8.8817841887295724137 E-16
2	8.8817841970012523234 E-16
3	8.8817841970012523234 E-16
$1/s^q$	8.8817841970012523234 E-16

With $c_i/d_i = 2$, P(z) is the same as $\frac{1}{s^q}$ up to 20 significant digits, which is the most accurate computation at the author's disposal.

Recall that all the analysis following (2.15) requires $s > 1$. When $s = 1$, the output space is degenerate and none of the previous analysis is valid. With $s = 1$, (2.9) becomes

$$P(z) = \frac{\sum_{i=1}^{q} c_i^{d_i}}{a^p} \tag{2.63}$$

As in the general case, we can show that

$$a = \sum_{i=1}^{q} c_i \tag{2.64}$$

With only one element in the output space, g must be a constant function and $q = |\mathcal{R}(g)| = 1$. Hence (2.64) becomes

$$a = \sum_{i=1}^{1} c_i = c_1 \tag{2.65}$$

and (2.63) becomes

$$P(z) = \frac{c_1^{d_1}}{a^p} \tag{2.66}$$

Recalling that

$$p = \sum_{i=1}^{q} d_i = \sum_{i=1}^{1} d_i = d_1 \tag{2.67}$$

we have

$$P(z) = \frac{a^p}{a^p} = 1 \qquad (2.68)$$

i. e. any system that realizes g will mask 100% of all possible single faults. This result is in agreement with Theorem 2. 10 which states that all faults in a general two node system are masked if and only if s = 1. Stated as a theorem

Theorem 2. 17

If function g has the property s = 1, then

$$Sg = Zg$$

and

$$Mg = 1.$$

Fault Masking and Detection Analysis in Any System

To describe the properties of a particular node in an arbitrary combinational network, we first seek a general form into which any network may be put for analysis. Such a form is shown in Figure 2.8. The particular node whose properties are sought is labeled 2. That an arbitrary network can always be represented in this general form with any node appearing in position 2 is apparent if we incorporate the concepts of predecessor and successor nodes.

Definition 2.21

In a directed graph with no directed cycles, node i is a predecessor of node j if there is a directed path from node i to node j. If node i is a predecessor of node j, then node j is a successor of node i. If node i is neither a successor nor a predecessor of node j, then node i is a cognate of node j.

In the combinational network of Figure 2.9, nodes 1, 2, and 4 are predecessors, node 6 is a successor, and node 3 is a cognate of node 5.

If we wish to obtain a general form, as shown in Figure 2.8, to represent an arbitrary combinational network C with node i of C at position 2, we must place all predecessors of node i in C in node 1 in the general form and all successors of node i in C in node 3 in the general form. Cognates of node i in C may be placed in either node 1 or node 3

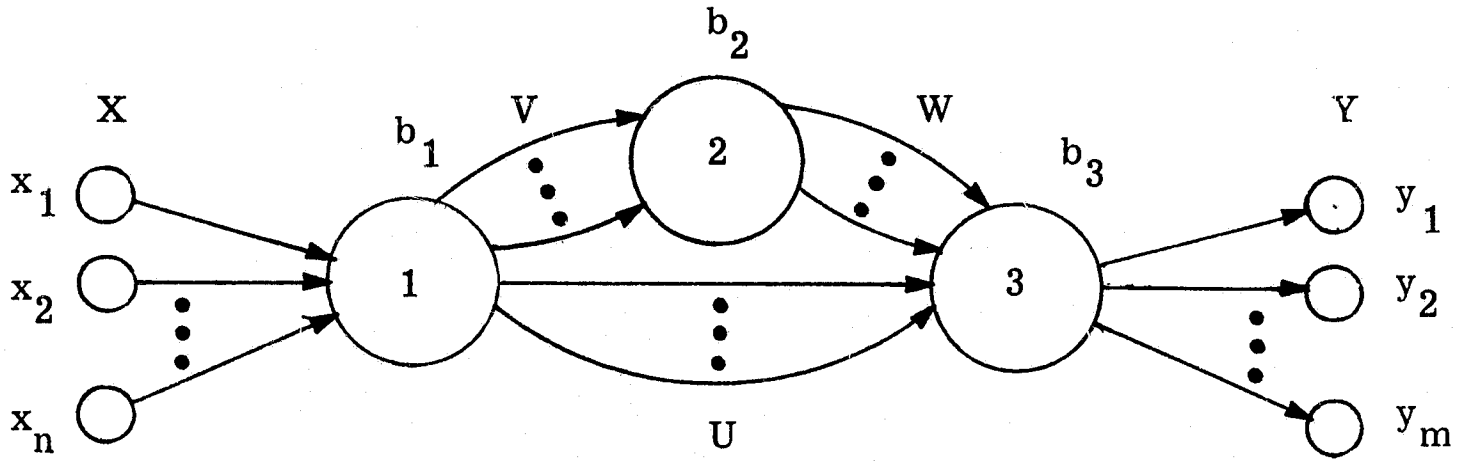


Figure 2. 8

A General Form Graph for Determining the Properties of a Single Node in an Arbitrary System

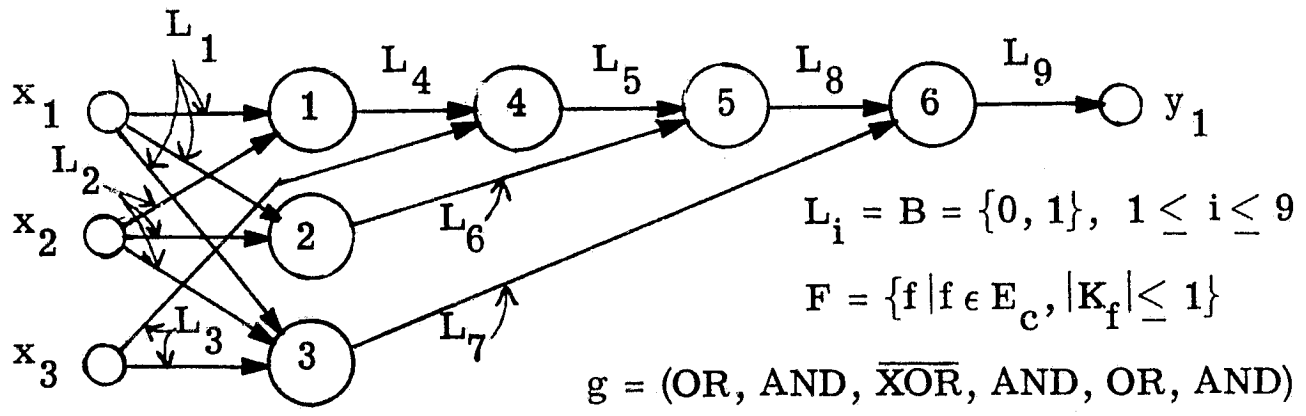
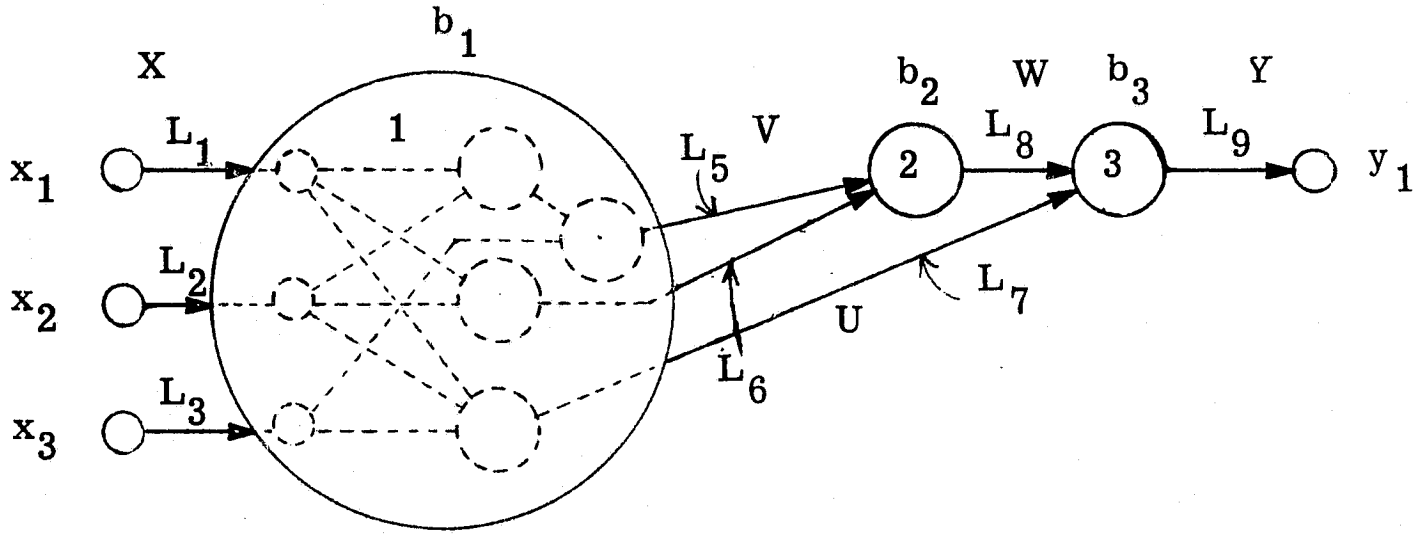
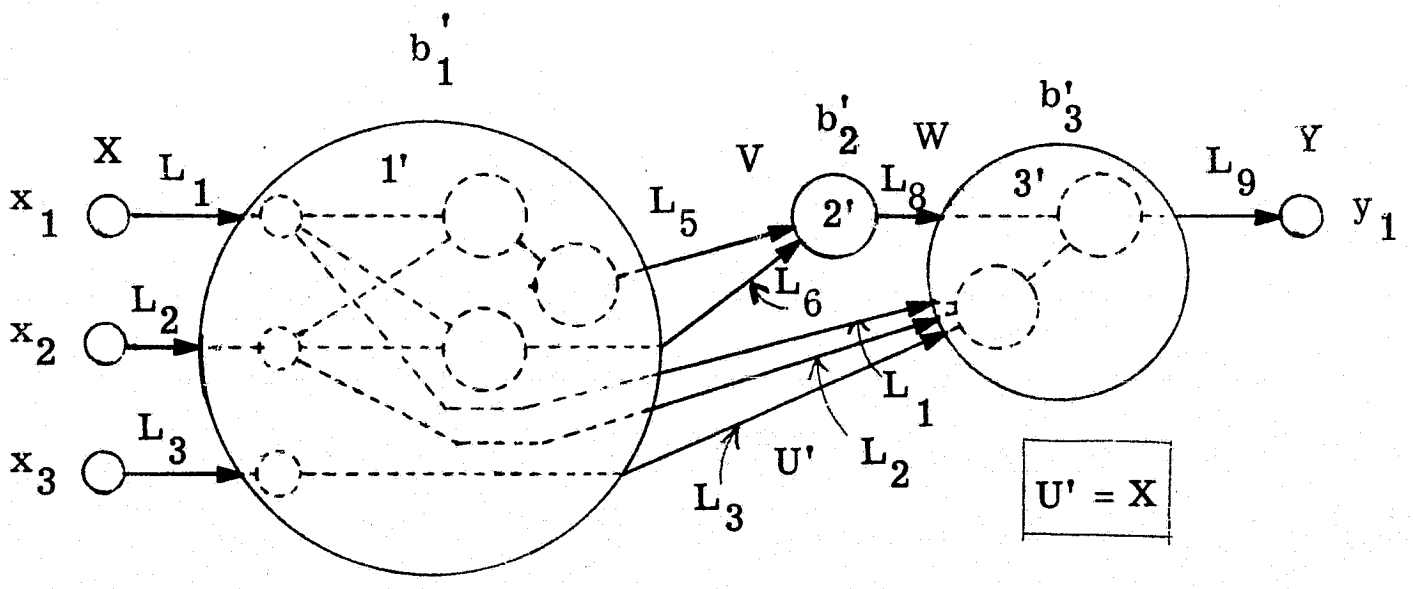


Figure 2. 9



(a)



(b)

Figure 2.10

in the general form. However if a cognate of node i in C is placed in node 3 (node 1), then all successors (predecessors) of the cognate must also be placed in node 3 (node 1).

Figure 2.10 contains two general form representations of the combinational network of Figure 2.9 each with node 5 in position 2. In Figure 2.10a, the cognate of node 5 is included in node 1 of the general form whereas, in Figure 2.10b, the cognate is included in node 3.

It is apparent that space U may be empty in the general form representation of some types of networks. In order to make our general form universally applicable, we will, in such instances, introduce a single line from node 1 to node 3 and make the node 1 mapping from X into U a constant mapping. Then $b_3(u, w)$ will be redefined as $b_3(w)$ for every $u \epsilon U$ and for every $w \epsilon W$.

If we have no original circuit nodes to place at node 3 of the general form, then b_3 will be the identity mapping. Similarly if there are no original circuit nodes to place at node 1 of the general form, then both U and V will be projected subspaces of X. With these conventions, any combinational network C may be put in general form with an arbitrary node of C appearing as node 2.

We now make a formal definition of the general form concept in order to use our model to determine its properties.

Definition 2.22

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

$$C = (D, S, F, b)$$

where

D is an $(n, m, 3, \ell)$-graph (Figure 2.8)

S is an arbitrary ℓ-tuple of sets

$b = (b_1, b_2, b_3) \in E_C$ (See Definition 2.4)

$F = \{f \mid f \in E_C,\ |K_f| \leq 1\}$

We now describe some conventions that are to be applied to the general system form.

Definition 2.23

In the general system form

$I = I_1 = X$ $\quad b_1: X \rightarrow V \times U$

$I_2 = V$ $\quad b_2: V \rightarrow W$

$O_2 = W$ $\quad b_3: W \times U \rightarrow Y$

$O_1 = V \times U$

$I_3 = W \times U$

$O_3 = O = Y$

It is often helpful to view the mapping $b_3: W \times U \rightarrow Y$ as a collection of mappings from W into Y. This will be done in the following way.

Definition 2. 24

For each element u of the space U in a general system form, define

$$h_u: W \rightarrow Y$$

where

$$h_u(w) = b_3(w, u) \qquad \forall w \in W$$

In the discussion that follows, we will let $\underset{u}{\equiv}$ denote the equivalence relation $\underset{h_u}{\equiv}$ induced on W by h_u, that is

$$w_1 \underset{u}{\equiv} w_2 \text{ iff } h_u(w_1) = h_u(w_2)$$

In order to simplify notation, we will define two functions associated with the general form.

Definition 2. 25

$$b_{1v}: X \rightarrow V \qquad b_{1u}: X \rightarrow U$$

If $b_1(x) = (v', u')$, then $b_{1v}(x) = v'$ and $b_{1u}(x) = u'$.

From the graph of C, we can infer that the net function, t, is

$$t(x) = b_3(b_2\, b_{1v}(x),\ b_{1u}(x))$$

We will now use Figure 2.10a to illustrate these concepts. Recall that this figure contains a general system form for the combinational network of Figure 2.9. Table 2.1 illustrates the mapping $b_1 : X \rightarrow V \times U$. Referring to Figure 2. 9, V is the space $L_5 \times L_6$ and U is the space L_7.

Of course X is the space $L_1 \times L_2 \times L_3$. As in most applications, $L_1 = L_2 = L_3 = L_4 = L_5 = L_6 = L_7 = L_8 = L_9 = B = \{0, 1\}$. The special functions b_{1v} and b_{1u} are also contained in Table 2.1.

Table 2.1

	$b_1(x)$	
x	$b_{1v}(x)$	$b_{1u}(x)$
L_1 L_2 L_3	L_5 L_6	L_7
0 0 0	0 0	1
0 0 1	0 0	0
0 1 0	0 0	0
0 1 1	1 0	1
1 0 0	0 0	0
1 0 1	1 0	1
1 1 0	0 1	1
1 1 1	1 1	0

Table 2.2

v	$b_2(v)$
L_5 L_6	L_8
0 0	0
0 1	1
1 0	1
1 1	1

Table 2.3

(w, u)	$b_3(w, u)$
L_8 L_7	L_9
0 0	0
0 1	0
1 0	0
1 1	1

Tables 2.2 and 2.3 contain the mappings $b_2 : V \rightarrow W$ and $b_3 : W \times U \rightarrow Y$ respectively. Again referring to Figure 2.9, W is the space L_8 and Y is the space L_9. Note also that $b_2 = g_5$ and $b_3 = g_6$.

Table 2.4 contains the functions h_0 and h_1. (Recall that $U = L_7 = B = \{0, 1\}$.) Note that $\equiv_1$ is the relation EQUALITY on the set W while $\equiv_0$ is the relation of "belonging" on the set W i.e. $w_1 \equiv_0 w_2$ if and only if w_1 and w_2 belong to the set W.

Table 2.4

W	$h_0(w)$	$h_1(w)$
L_8	L_9	L_9
0	0	0
1	0	1

We are now ready to investigate the properties of the general system form with single faults.

Theorem 2.18

In a general system form with single faults, a single fault at node 2, $f = (b_1, f_2, b_3)$, is masked iff

$$\forall x \in X, \quad b_2\, b_{1v}(x) \underset{b_{1u}(x)}{\equiv} f_2\, b_{1v}(x)$$

Proof

To show sufficiency, let the above equivalence be true for every element in the space X. Now let x be any element in X, then we obtain

$$t(x) = b_3(b_2\, b_{1v}(x),\ b_{1u}(x)) = b_3(f_2\, b_{1v}(x),\ b_{1u}(x))$$

$$= \alpha(f)(x)$$

Hence $t(x) = \alpha(f)(x)$ for all $x \in X$, i.e. $\alpha(f) = t$ and the fault f is masked (Definition 2.15a).

To show necessity, let $f = (b_1, f_2, b_3)$ be masked and assume that for some $x \in X$,

$$b_2\, b_{1v}(x) \underset{b_{1u}(x)}{\neq} f_2\, b_{1v}(x)$$

then we have

$$t(x) = b_3(b_2\, b_{1v}(x),\ b_{1u}(x)) \neq b_3(f_2\, b_{1v}(x),\ b_{1u}(x)) = \alpha(f)(x).$$

Thus, f is detectable (Definition 2.15b) and hence not masked (Lemma 2.1.2), contradicting the hypothesis. This proves necessity and completes the proof of the theorem.

As usual, there is an immediate corollary characterizing detectable single faults at node 2.

Corollary 2.18.1

In a general system form with single faults, a single fault at node 2, $f = (b_1, f_2, b_3)$, is detectable iff $\exists\, x \in X$ such that

$$b_2\, b_{1v}(x) \underset{b_{1u}(x)}{\neq} f_2\, b_{1v}(x)$$

Proof

Immediate from Theorem 2.18 and Lemma 2.1.2.

Our next goal is to count the number of single faults at node 2 that are masked. Some preliminary definitions are required.

Definition 2.26

For every $v \in \mathcal{R}(b_{1v}(x))$ in a general system form, define the equivalence relation, R_v, on W as follows:

$$w_1 R_v w_2 \text{ iff } \forall x \in (b_{1v})^{-1}(v),\ w_1 \underset{b_{1u}(x)}{\equiv} w_2$$

or equivalently

$$w_1 R_v w_2 \text{ iff } \forall x \in (b_{1v})^{-1}(v),\ b_3(w_1, b_{1u}(x)) = b_3(w_2, b_{1u}(x))$$

or equivalently

$$R_v = \underset{x \in (b_{1v})^{-1}(v)}{\bigcap} \underset{b_{1u}(x)}{\equiv} = \underset{u \in b_{1u}[(b_{1v})^{-1}(v)]}{\bigcap} \underset{u}{\equiv}$$

All three forms of this definition will be useful. The last form clearly indicates that

$$R_v \subseteq \underset{b_{1u}(x)}{\equiv} \text{ for every } x \in (b_{1v})^{-1}(v)$$

For the example of Figure 2.10a, we observe (using Table 2.1) that

$$b_{1v}^{-1}(00) = \{000, 001, 010, 100\}$$

and

$$b_{1u}[b_{1v}^{-1}(00)] = \{0, 1\}$$

and therefore

$$R_{00} = \underset{0}{\equiv} \bigcap \underset{1}{\equiv} = \text{EQUALITY}$$

Similarly,

$$b_{1v}^{-1}(01) = \{110\}$$

$$b_{1v}^{-1}(10) = \{011, 101\}$$

$$b_{1v}^{-1}(11) = \{111\}$$

$$b_{1u}[b_{1v}^{-1}(01)] = \{1\}$$

$$b_{1u}[b_{1v}^{-1}(10)] = \{1\}$$

$$b_{1u}[b_{1v}^{-1}(11)] = \{0\}$$

and

$$R_{01} = R_{10} = \underset{1}{\equiv} = \text{EQUALITY}$$

$$R_{11} = \underset{0}{\equiv} = \overline{0, 1} = W \times W$$

Definition 2.27

For every $v \in V$ in a general system form, define the counting constant, c_v, as follows:

$$c_v = \begin{cases} |[b_2(v)]_{R_v}| & \text{if } v \in \mathcal{R}(b_{1v}) \\ |W| & \text{if } v \notin \mathcal{R}(b_{1v}) \end{cases}$$

In the example of Figure 2.10a, we have

$$c_{00} = |[b_2(00)]_{R_{00}}| = |[0]_{R_{00}}| = 1$$

$$c_{01} = |[b_2(01)]R_{01}| = |[1]_{R_{01}}| = 1$$

$$c_{10} = \left|[b_2(10)]_{R_{10}}\right| = \left|[1]_{R_{10}}\right| = 1$$

$$c_{11} = \left|[b_2(11)]_{R_{11}}\right| = \left|[1]_{R_{11}}\right| = 2$$

We are now ready to count the number of single faults at node 2 that are masked.

Theorem 2.19

In a general system form with single faults, the number of single faults at node 2 that are masked is computed as follows:

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

Proof

The proof consists of verifying that the counting constant c_v is the number of distinct images that the element v can assume under a mapping, f_2, at node 2 and still satisfy the requirements of Theorem 2.18. First consider an element, v of V that is not in the range of b_{1v}. Theorem 2.18 places no restriction on the image of v under f_2, therefore there are $|W|$ possible choices for $f_2(v)$ that meet the conditions of Theorem 2.18. By Definition 2.27, this number is c_v. Now consider an element $v \in V$ that is in the range of b_{1v}. Theorem 2.18 requires that $f_2(v)$ be $\underset{b_{1u}(x)}{\equiv}$ equivalent to $b_2(v)$ for every $x \in X$ for which $b_{1v}(x) = v$. By Definition 2.26, this means that $f_2(v)$ and $b_2(v)$ must be R_v equivalent. We thus have $\left|[b_2(v)]_{R_v}\right|$ choices for $f_2(v)$ that meet the requirements of Theorem 2.18 and this number

is c_v (Definition 2.27). Thus, we see that in any case, c_v is the number of distinct images for v under the mapping f_2. From Definition 2.22 we know that for any mapping, f_2, from V into W, (b_1, f_2, b_3) is a fault in the general system form. This means that the choice of $f_2(v_1)$ is completely independent of the choice of $f_2(v_2)$ when $v_1 \neq v_2$. Since all the choices for $f_2(v)$ are independent, we deduce that the number of node 2 mappings that satisfy Theorem 2.18 is given by

$$\prod_{v \in V} c_v$$

We also know that b_2 is one of the mappings counted but that (b_1, b_2, b_3) is not a single fault at node 2 (Definition 2.16). We then conclude that the number of single faults at node 2 that are masked is

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

proving the theorem.

Due to Lemma 2.1.2, we have as an immediate corollary to Theorem 2.19 a means for computing the number of single faults at node 2 that are detectable.

Corollary 2.19.1

In a general system form with single faults the number of single faults at node 2 that are detectable is given by

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

For the example of Figure 2.10a, we have

$$|F_2| = \prod_{v \in V} c_v = c_{00} \cdot c_{01} \cdot c_{10} \cdot c_{11} = 2$$

and the number of detectable single faults at node 2 is

$$|W|^{|V|} - \prod_{v \in V} c_v = 2^4 - 2 = 14$$

Of course b is one of the single faults at node 2 that is masked (Lemma 2.1.1). The other is

$$f = (OR, AND, \overline{XOR}, AND, XOR, AND)$$

See page 237 for a systematic procedure to determine the faults that are masked.

We now investigate the limiting conditions for fault masking and fault detection at node 2, but first a supporting lemma.

Lemma 2.20.1

In a general system form, the set of single faults at node 2 is empty if and only if $|W| = 1$.

Proof

The proof is identical to the proof of Lemma 2.9.1.

In a general system form with $|W| = 1$, all single faults at node 2 are masked, detectable, locatable, and completely diagnosable by the fact that none exist. For this reason, some of the results that follow will assume that the general system form is nontrivial.

Theorem 2.20

In a general system form with single faults, all single faults at node 2 are masked iff

$$\forall u \in \mathcal{R}(b_{1u}), \quad h_u \text{ is a constant function.}$$

Proof

To show sufficiency, let h_u be a constant function for every $u \in \mathcal{R}(b_{1u})$, then we have

$$\underset{u}{\equiv} = W \times W \qquad \forall u \in \mathcal{R}(b_{1u})$$

and since

$$R_v = \overset{\equiv}{\underset{u \in b_{1u}[(b_{1v})^{-1}(v)]}{\frown}} u$$

we have

$$R_v = W \times W \qquad \forall v \in V$$

This implies that

$$\left|[b_2(v)]_{R_v}\right| = |W|, \qquad \forall v \in V.$$

Then, the number of single faults at node 2 that are masked is given by Theorem 2.19

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

Since this is the total number of single faults at node 2 (Definition 2.22) all must be masked.

To show necessity, assume that h_u is not constant for some $u \in \mathcal{R}(b_{1u})$. Then $\exists$ distinct $w_1, w_2 \in W$ for which $b_3(w_1, u) \neq b_3(w_2, u)$. This implies that $w_1 \not\equiv_u w_2$. Since $u \in \mathcal{R}(b_{1u})$, $\exists x \in X$ for which $b_{1u}(x) = u$, and if we let $v = b_{1v}(x)$, then w_1 is not R_v equivalent to w_2 (Definition 2.26). Therefore, $[b_2(v)]_{R_v}$ cannot contain both w_1 and w_2 which implies that $c_v < |W|$ and therefore

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

i.e. not all single faults at node 2 are masked.

If we detect all single faults at node 2, we must satisfy two conditions.

Theorem 2.21

In a general system form (with single faults and $|W| \geq 2$), all single faults at node 2 are detectable iff

$$c_v = 1 \quad \text{for every } v \in V$$

Proof

To show sufficiency, let $c_v = 1$ for every $v \in V$, then the number of detectable faults at node 2 is given by Corollary 2.19.1

$$|w|^{|v|} - \prod_{v \in V} c_v = |w|^{|v|} - 1$$

i.e. all faults at node 2 are masked.

To show necessity, let all single faults at node 2 be detectable, then we have again by Corollary 2.19.1 that

$$|W|^{|V|} - 1 = |w|^{|v|} - \prod_{v \in V} c_v$$

or

$$\prod_{v \in V} c_v = 1$$

which can occur only if

$$c_v = 1 \quad \text{for every } v \in V$$

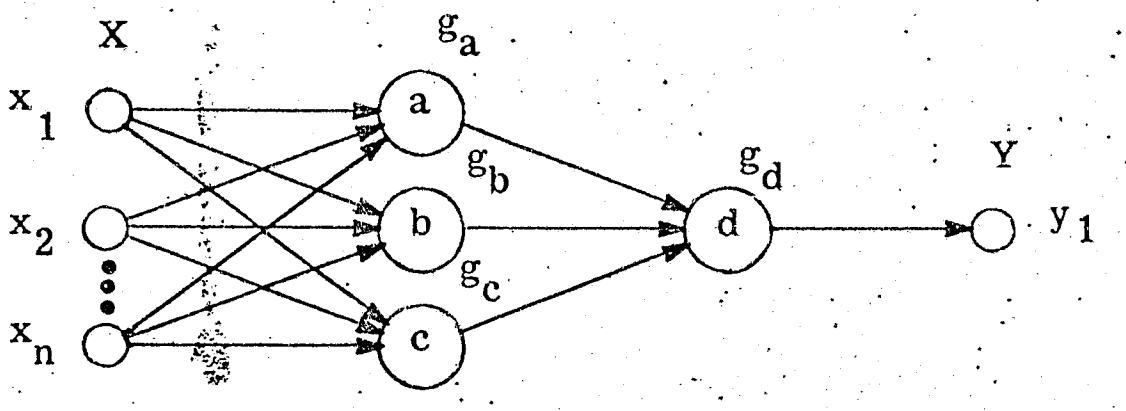
thus proving the theorem.

Unlike many limiting conditions in the two node general system, the limiting conditions in a general system form are obtainable in instances of practical value. For example, the triple modular redundancy scheme satisfies the conditions of Theorem 2. 20.

Example 2. 5 Triple Modular Redundancy

The fault-masking properties of networks incorporating triple modular redundancy are widely known. The analysis presented here strengthens the observed properties of triply modular redundant (TMR) systems and provides some new insight into the effects of the voting element. Figure 2.11 contains a standard arrangement of a TMR system.

The analysis begins with one of the three identical nodes that arise as a result of the triplication process. Due to the symmetry of the network, the effect of each will be the same. Without loss of



$$g_a = g_b = g_c$$

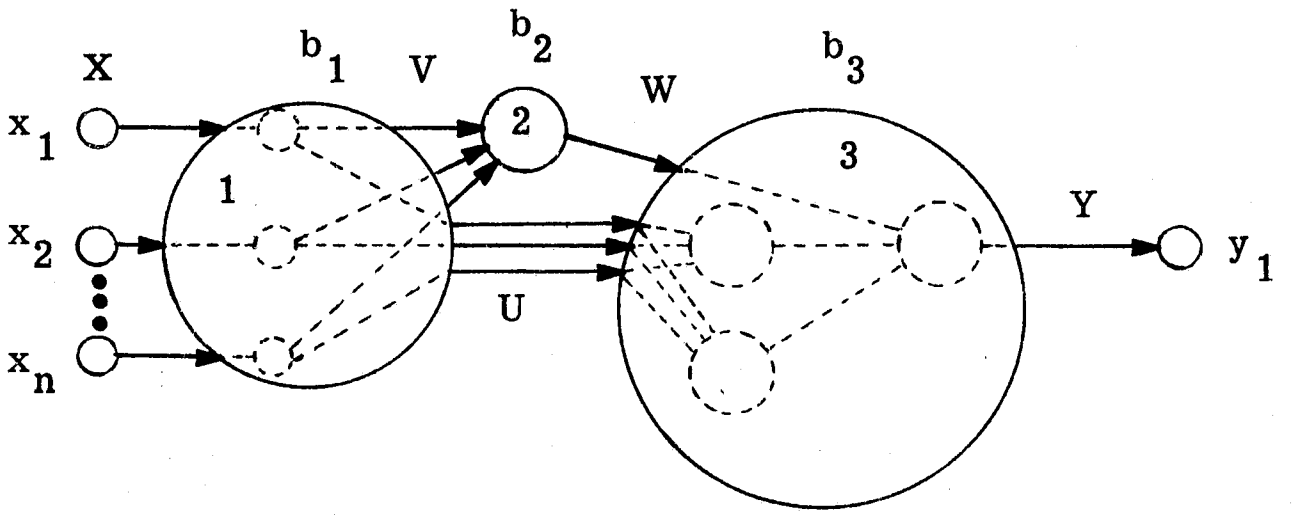
$$g_d = \text{MAJORITY}$$

$$L_i = \{0, 1\},\ 1 \leq i \leq \ell$$

$$F' = \{f \mid f \in E_C,\ K_f \leq 1\}$$

Figure 2.11

A TMR System



$$V = U = X$$

$$b_1 : X \rightarrow X \times X$$

$$b_{1v} = b_{1u} = \text{IDENTITY}$$

$$b_2 = g_a$$

$$b_3 : W \times X \rightarrow Y$$

$$b_3(w,x) = g_d(w, g_b(x), g_c(x))$$

$$L_i = \{0, 1\}, \quad 1 \leq i \leq \ell$$

$$F = \{(b_1, f_2, b_3) \mid f_2 : V \rightarrow W\} \cup A_1$$

where $A_1 \subseteq E_c$

Figure 2.12

A General System Form for the Analysis of Node a in the Combinational Network of Figure 2.11.

generality, node a is selected for computation. Figure 2.12 displays the appropriate general system form for the analysis of node a.

Since we are not currently interested in the behavior of nodes 1 and 3 in the general form it is not necessary to explicitly compute A_1, the single fault set for these nodes. Let x be a fixed but arbitrary element of X. Let w_1 and w_2 be any two elements of W, then

$$h_x(w_1) = b_3(w_1, x) = g_d(w_1, g_b(x), g_c(x))$$

and

$$h_x(w_2) = b_3(w_2, x) = g_d(w_2, g_b(x), g_c(x))$$

By construction, $g_b(x) = g_c(x)$. The definition of MAJORITY function then implies

$$g_d(w_1, g_b(x), g_c(x)) = g_d(w_2, g_b(x), g_c(x))$$

or

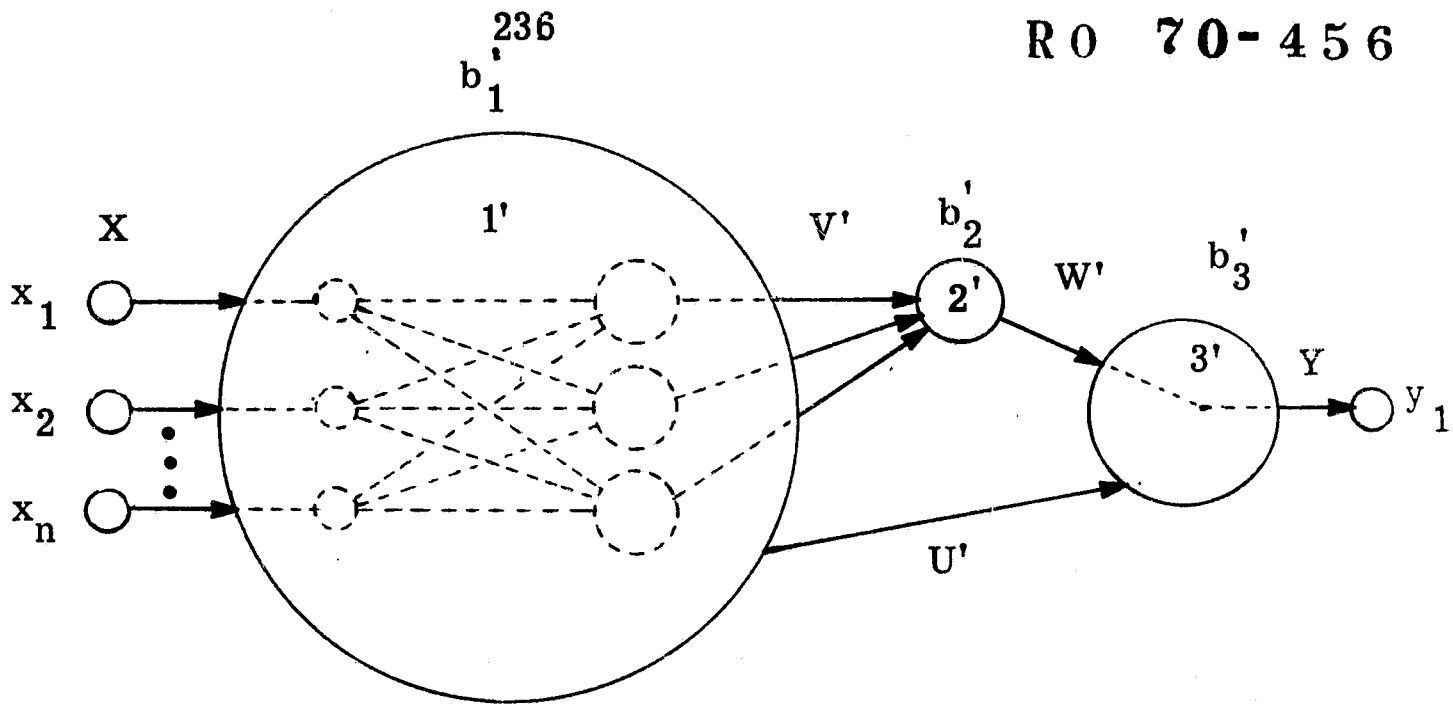
$$h_x(w_1) = h_x(w_2)$$

Since w_1 and w_2 were any two elements of W, h_x is a constant function. Since x was an arbitrary element of X, we have

$$\forall x \in X,\ h_x \text{ is a constant function.}$$

All single faults at node 2, (node a) are masked by Theorem 2.20.

By symmetry, all single faults at nodes b and c are also masked. Our theory demonstrates conclusively that any single fault in the triplicated section of a TMR network is masked.



$W' = Y$

$U' = \{0\}$ is a dummy space

b_1': $X \rightarrow V' \times U'$

$b_1'(x) = (g_a(x), g_b(x), g_c(x), 0)$

$b_{1v}'(x) = (g_a(x), g_b(x), g_c(x))$

$b_{1u}'(x) = 0$

$b_2' = g_d$

b_3': $Y \times U' \rightarrow Y$

$b_3'(y, 0) = y \quad \forall\, y \in Y$

$L_i = \{0, 1\}, \quad 1 \leq i \leq \ell$

$|V'| = 8$

$|\mathcal{R}(b_{1v}')| = 2$

$F = \{(b_1', f_2, b_3') \,|\, f_2: V' \rightarrow Y\} \cup A_2$ where $A_2 \subseteq E_C$

Figure 2.13

A General System Form for the Analysis of Node d in the Combinational Network of Figure 2.11.

Many treatments of TMR networks assume that the MAJORITY gate is perfect in order to eliminate complications resulting from error-prone voters. Our techniques can handle faults in the voter with no more difficulty than that encountered in the analysis of the triplicated section. The modified general form appropriate for voter fault analysis appears in Figure 2.13. The number of single faults at node 2 that are masked is given by Theorem 2.19

$$|F_2| = -1 + |Y|^{|\overline{\mathcal{R}(b_{1v}')}|} = -1 + 2^6 = 63$$

Unlike the results of the analysis of a triplicated node, this information about the voter is seldom, if at all, mentioned in the literature. Two implications are immediate.

First, although the network cannot mask all single faults in the voter, fully 24.7% of the possible single faults are masked. If the most probable voter faults are among those masked, the voter may indeed perform in a highly reliable manner.

The second important result is that 63 other functions could have been placed at node 2 without changing the net function. We then have more alternative designs for economical circuits when reliability is not an important factor.

Single Fault Analysis Algorithm

An algorithm for computing the number of single-faults that are masked at each node in a combinational network will now be described. The capability for generating a list of all such masked faults is inherent

Many treatments of TMR networks assume that the MAJORITY gate is perfect in order to eliminate complications resulting from error-prone voters. Our techniques can handle faults in the voter with no more difficulty than that encountered in the analysis of the triplicated section. The modified general form appropriate for voter fault analysis appears in Figure 2.13. The number of single faults at node 2 that are masked is given by Theorem 2.19

$$|F_2| = -1 + |Y|^{|\overline{\mathcal{R}(b_{1v}')}|} = -1 + 2^6 = 63$$

Unlike the results of the analysis of a triplicated node, this information about the voter is seldom, if at all, mentioned in the literature. Two implications are immediate.

First, although the network cannot mask all single faults in the voter, fully 24.7% of the possible single faults are masked. If the most probable voter faults are among those masked, the voter may indeed perform in a highly reliable manner.

The second important result is that 63 other functions could have been placed at node 2 without changing the net function. We then have more alternative designs for economical circuits when reliability is not an important factor.

Single Fault Analysis Algorithm

An algorithm for computing the number of single-faults that are masked at each node in a combinational network will now be described. The capability for generating a list of all such masked faults is inherent

in the algorithm. Reference should be made to Theorem 2.19 and associated definitions, for the underlying theory.

The algorithm proceeds node by node through the network. At the q-th step, the network is put in a general system form such that node q in the network being analyzed appears as node 2 in the form.

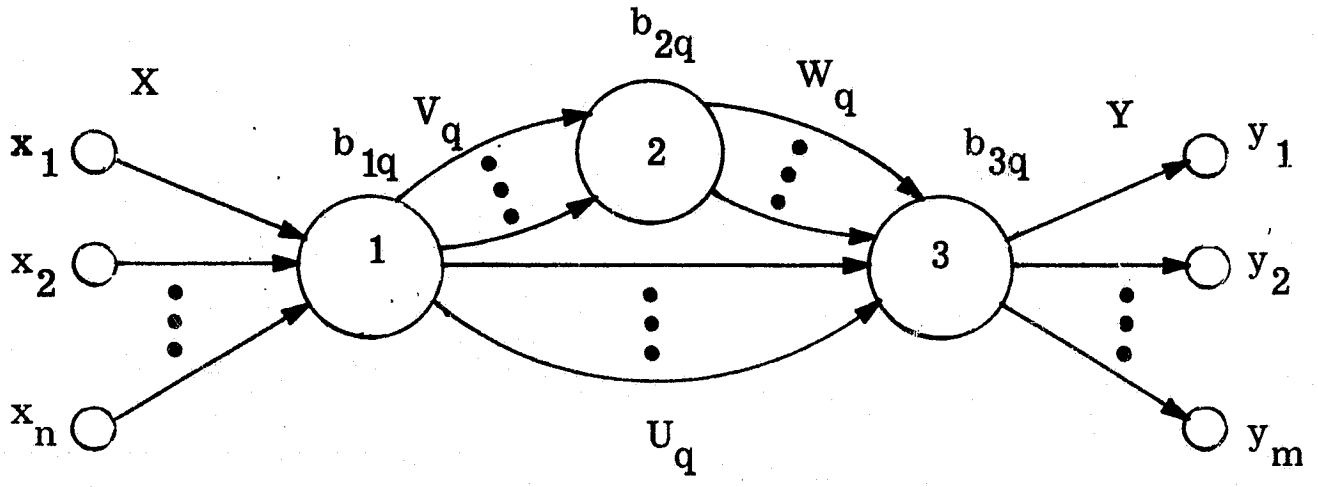
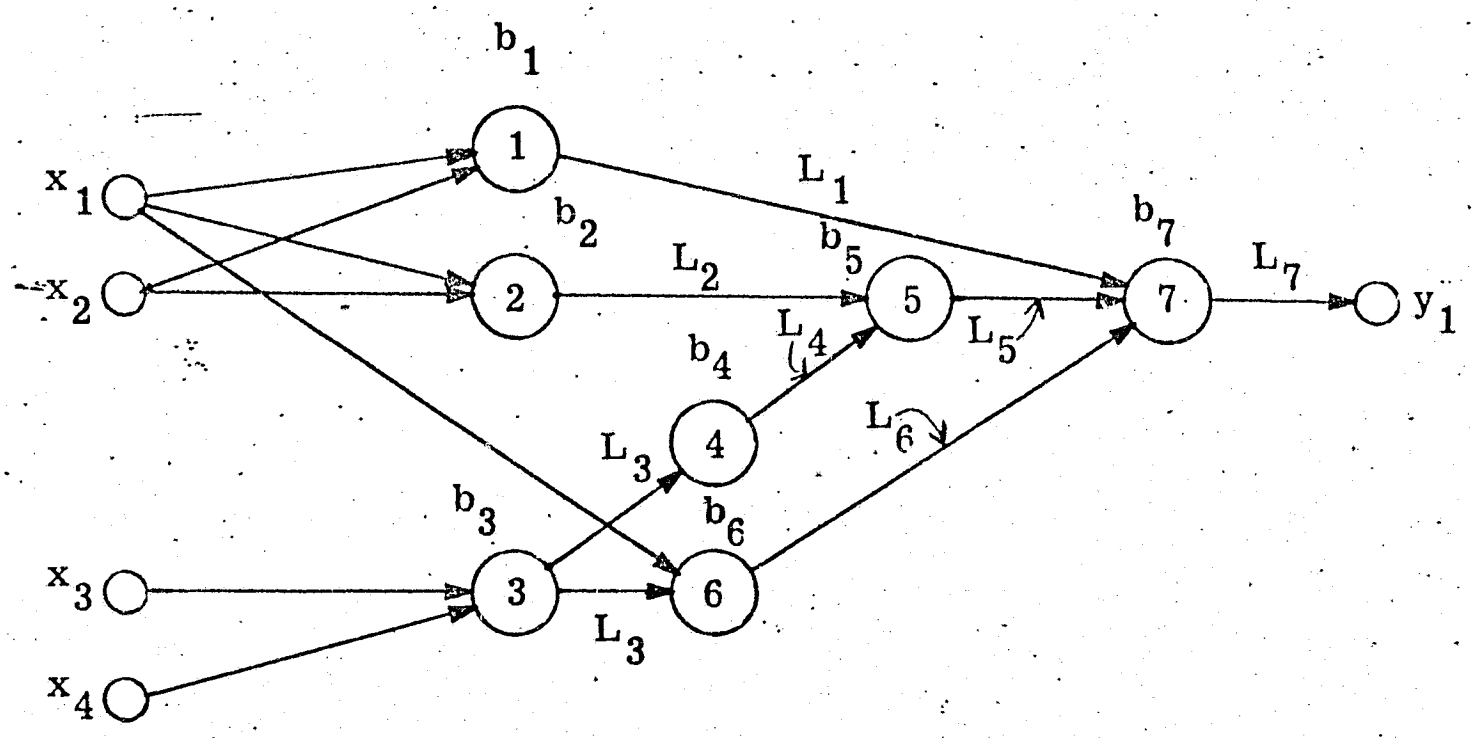


Figure 2.14

A General System Form for Node q.



$b_1 = b_5 = \text{AND}$

$b_2 = \text{NOR}$

$b_3 = \text{XOR}$

$b_4 = \text{NOT}$

$b_6 = b_7 = \text{OR}$

Figure 2.15

Graph for Example

Figure 2.14 is essentially Figure 2.2 with subscripts to indicate that the algorithm is being applied to node q.

In order to reduce the number of computations required to analyze node q, a set of truth tables is generated in a preliminary step that contains all information needed to produce the qth general system form. For each node q the algorithm calls on the FAULT subroutine that calculates the c_{vq} constants required by Theorem 2.19. The FAULT subroutine also calculates $[b_{2q}(v_q)]_{R_{v_q}}$ for every $v_q \in V_q$. These equivalence classes characterize the single faults at node q which are masked.

To illustrate the algorithm, the combinational network whose graph is shown in Figure 2.15 will be considered.

I. Compile a table, called Table 1, that displays all mappings of the form $X \rightarrow O_q$, $1 \leq q \leq k$. O_q is the output space of node q. Figure 2.16 contains the table for the example.

II. Identify a family of minimal cutsets of the combinational network digraph that covers the line set of the digraph. This family will contain a number of sets equal to the length of the longest directed path in the digraph. A collection of cutsets for the example, excluding X appears as the input coordinate sets in the tables of Figure 2.17. Each set O_q should appear in some cutset.

III. For each cutset obtained in step II, except the input space, compile a truth table that displays the mapping from the

cartesian product of the signal sets on the lines of the cutset into the output space. The set of tables for the example appears in Figure 2.17.

IV. Let $q = 1$.

V. Select one of the tables that were compiled in Step III whose input coordinate set includes the set of output lines of node q. Let W_q = the output space of node q; $W_q \times U_q$ = the input space of the table selected; V_q = the input space of node q; $b_{1_q v}$ is the network mapping from X into V_q (contained in Table 1); b_{2q} is the q^{th} coordinate of the 0-fault of the network; if $U_q \neq \phi$, then $M_{1q} = b_{1_q u}$ is the net mapping from X to U_q; if $U_q = \phi$, M_{1q} = NULL; M_{2q} is the mapping from $W_q \times U_q$ into Y contained in the table selected by this step.

VI. Call the FAULT subroutine with parameters $(b_{1q}, b_{2q}, M_{1q}, M_{2q})$.

VII. Compute the number of single faults masked at node q using Theorem 2.19 and the constant table returned by the FAULT subroutine. Figure 2.18 contains a summary for the example.

VIII. Compile a list of faults using the class table returned by the FAULT subroutine.

IX. If $q < k$, set $q = q+1$ and go to step V. Otherwise, STOP.

X_1	X_2	X_3	X_4	L_1	L_2	L_3	L_4	L_5	L_6	L_7
0	0	0	0	0	1	0	1	1	0	1
0	0	0	1	0	1	1	0	0	1	1
0	0	1	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	1	1	0	1
0	1	0	0	0	0	0	1	0	0	0
0	1	0	1	0	0	1	0	0	1	1
0	1	1	0	0	0	1	0	0	1	1
0	1	1	1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	1	1
1	0	0	1	0	0	1	0	0	1	1
1	0	1	0	0	0	1	0	0	1	1
1	0	1	1	0	0	0	1	0	1	1
1	1	0	0	1	0	0	1	0	1	1
1	1	0	1	1	0	1	0	0	1	1
1	1	1	0	1	0	1	0	0	1	1
1	1	1	1	1	0	0	1	0	1	1

Figure 2.16

Table 1 for Example

X_1	L_1	L_2	L_3	L_7
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Table 2

L_1	L_2	L_4	L_6	L_7
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Table 3

L_1	L_5	L_6	L_7
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Table 4

L_7	L_7
0	0
1	1

Table 5

Figure 2.17

Cut-set Tables for Example

Node	# Single Faults Masked	% Single Faults Masked*
1	7	46.6
2	3	20
3	0	0
4	1	$33\frac{1}{3}$
5	3	20
6	0	0
7	15	5.88

Figure 2.18

Analysis of Fault Masking for Network of Figure 2.15

$$* \% \text{ single faults masked} = \frac{\# \text{ masked}}{|W|^{|V|} - 1}$$

FAULT Subroutine

This routine computes a table of counting constants sufficient to compute the number of single-faults masked in node 2 of a general system form. It also provides a list of equivalence classes sufficient to compile a list of all such masked faults. The subroutine requires four parameters: (b_1, b_2, M_1, M_2) where b_1 is the $X \rightarrow V$ mapping of the general system form, b_2 is the node 2 mapping $V \rightarrow W$, M_1 is the X to U mapping, and M_2 is the $W \times U$ to Y mapping.

I. Compute the equivalence relation induced on X by b_1. If M_1 = NULL, compute the equivalence relation, $\overset{M_2}{\equiv}$, induced on W_q by the mapping M_2. Let $V = (v_1, \ldots, v_{|V|})$.

Set i = 1.

II. If $v_i \notin \mathcal{R}(b_1)$, enter W in the CLASS table with tag v_i, enter $|W|$ in the CONSTANT table with tag v_i, and go to STEP XI.

If $v_i \in \mathcal{R}(b_1)$, and M_1 = NULL, enter $[b_2(v_i)]_{M_2}$ in the CLASS table with tag v_i, enter $|[b_2(v_i)]_{M_2}|$ in the CONSTANT table with tag v_i, and go to step XI.

If $v_i \in \mathcal{R}(b_1)$ and $M_1 \neq$ NULL, go to STEP III.

III. Let j = 1.

Set RELATION = the relation on W in which every element is in a single equivalence class.

IV. Let x_{ij} = the jth element in the ith equivalence class of the relation induced on X by b_1.

Obtain $M_1(x_{ij}) = a'_{ij}$.

V. If a'_{ij} is in the h-table, go to step VIII.

VI. Call the h-subroutine with parameters (a'_{ij}, M_2) to compute the equivalence classes of $R_{a'_{ij}}$.

VII. Append the returned relation (in partition form) to the h-table with tag a'_{ij}.

VIII. RELATION = RELATION $\cap$ (h-table entry having tag a'_{ij}). If RELATION = EQUALITY, go to step X, otherwise proceed with step IX.

IX. If all elements in the ith equivalence class have been processed, go to step X, otherwise set j = j+1 and go to step IV.

X. RELATION contains the equivalence classes of the relation R_{v_i}. Enter $[b_2(v_i)]_{R_{v_i}}$ in the class table and $|[b_2(v_i)]_{R_{v_i}}|$ in the constant table, (each with tag v_i).

XI. If all equivalence classes of the relation of Step I have been examined, go to Step XII; otherwise, set i = i+1 and go to Step II.

XII. Return to main routine with the class and constant tables.

Example Computations for the FAULT Subroutine

For the example, we will consider the case where q = 5. Then, the output lines of node 5 (L_5) appears as one of the domain coordinates of Table 4.

$W_5 = L_5$

$V_5 = L_2 \times L_4$

$W_5 \times U_5 = L_1 \times L_5 \times L_6$

$U_5 = L_1 \times L_6$

b_{15v} : $X \rightarrow L_2 \times L_4$ contained in Table 1.

$b_{25} : L_2 \times L_4 \rightarrow L_5$

$M_1 : X \rightarrow L_1 \times L_6$ contained in Table 1.

$M_2 : L_1 \times L_5 \times L_6 \rightarrow L_7$ contained in Table 4.

The parameters passed to the FAULT subroutine ($b_{15v}, b_{25}, M_1, M_2$).

The results of step I are:

$$(b_{15v})^{-1}(00) = \{0101, 0110, 1001, 1010, 1101, 1110\}$$

$$(b_{15v})^{-1}(01) = \{0100, 0111, 1000, 1011, 1100, 1111\}$$

$$(b_{15v})^{-1}(10) = \{0001, 0010\}$$

$$(b_{15v})^{-1}(11) = \{0000, 0011\}$$

$$V_5 = \{v_1, v_2, v_3, v_4\} = \{00, 01, 10, 11\}$$

$$W_5 = \{w_1, w_2\} = \{0, 1\}$$

The remaining steps in the FAULT subroutine are summarized in Figure 2.19. Scan the table in normal reading fashion. Entries indicate processing steps, Blanks indicate no processing.

Step I, II, XI	Step III Step IX	Step IV	Step IV	Step VII		Step VIII	Step X	
i	j	x_{ij}	$M_1(x_{ij}) \in L_1 \times L_6$	h-Table		RELATION	CLASS	CONSTANT
				TAG				
1	1					$\overline{0,1}$		
		0101	01	01	$\overline{0,1}$	$\overline{0,1}$		
	2	0110	01			$\overline{0,1}$		
	3	1001	01			$\overline{0,1}$		
	4	1010	01			$\overline{0,1}$		
	5	1101	11	11	$\overline{0,1}$	$\overline{0,1}$		
	6	1110	11			$\overline{0,1}$	$\overline{0,1}$	2
2	1					$\overline{0,1}$		
		0100	00	00	$\bar{0}:\bar{1}$	$\bar{0}:\bar{1}$*	$\bar{0}$	1
3	1					$\overline{0,1}$		
		0001	01			$\overline{0,1}$		
		0010	01			$\overline{0,1}$	$\overline{0,1}$	2
4	1					$\overline{0,1}$		
		0000	00			$\bar{0}:\bar{1}$*	$\bar{1}$	1

*Terminate computation of RELATION because RELATION is EQUALITY.

Figure 2.19

h-Subroutine

The h-subroutine, for parameters (a'_{ij}, M_2), will compute the equivalence relation (in partition form) induced on space W by the mapping $h_{a'_{ij}}$. The h-table contains equivalence relations $\underset{u}{\equiv}$ for values of u that are of interest. This subroutine will be called exactly $\left|\underset{M_1}{\equiv}\right|$ times during execution of the FAULT subroutine.

I. Set r = 1.

   Set $W = (w_1, w_2, \ldots)$

II. Compute $M_2(w_r, a'_{ij}) = y_r$.

III. If there is a list with identifier y_r, go to Step IV; otherwise create a list with identifier y_r. This list will eventually contain all the elements of $h_{a'_{ij}}^{-1}(y_r)$.

IV. Enter w_r on the list with tag y_r.

V. If $r = |W|$, go to Step VI; otherwise set r = r+1 and go to Step II.

VI. Assemble the equivalence classes of the relation induced on W by $h_{a'_{ij}}$. Each equivalence class consists of the elements on one of the lists generated by Step III. This relation is returned to the calling subroutine.

Sample Execution of h-Subroutine

For example, the h-subroutine execution that produced the h-table entry for line 2 in Figure 2.19 is presented in Figure 2. 20. The entry parameters are:

a'_{11} = 01

M_2 = Table 4

$W_q = \{w_1, w_2\} = \{0, 1\}$

r	w_r	$M_2(w_r, a'_{ij}) = y_r$
1	0	1*
2	1	1**

*Create List 1 with tag 1 and enter $w_1 = 0$ as the first element in the list.

**Append $w_2 = 1$ to list 1.

Figure 2.20

Sample h-Subroutine Calculation

Complexity of Calculations

This algorithm requires on the order of $|X| + |V| + |W|^2$ computations for each node analyzed. An exhaustive search would require on the order of $|X|\ |Y|\ |W|^{|V|}$ computations for each node. The saving is large for nodes with high dimensions.

The number of preliminary steps is on the order of $\ell|X| + \ell(\text{length}) + \delta(\text{length})$ where δ is the largest number of elements in a cutset space, and length refers to the length of a longest directed path in the network. An exhaustive search algorithm would also require such preliminary steps.

Fault Location

Emphasis in preceding sections has been placed on fault masking and fault detection. To facilitate repair and maintenance, it is frequently desirable to have the additional capability of locating possible faults to within an easily serviced subsystem. The possibly faulty subsystem may be replaced by a good subsystem while detailed tests are run on the suspect component. In this way, the system may continue to function during most of the testing process and during the repair process. Justification for the additional cost of locatable systems would have to come from the cost of down time for repair and maintenance. For example, in a deep space probe or manned mission, the crew could not well afford having the guidance system inoperable during minor repair or maintenance procedures. We begin our development with the general two node system.

Theorem 2.22

In the general two node system with single faults, a single fault at node 1, $f = (f_1, b_2)$, is locatable iff

$$\exists\, x_1, x_2 \in X \text{ such that}$$

(1) $x_1 \underset{b_1}{\equiv} x_2$

and (2) $f_1(x_1) \underset{b_2}{\not\equiv} f_1(x_2)$.

Proof: Sufficiency

Let $\exists\, x_1, x_2 \in X$ that satisfy (1) and (2). Let $f' = (b_1, f_2)$ be any fault whose first component is b_1.

(1) $\Longrightarrow b_1(x_1) = b_1(x_2) \Longrightarrow f_2 b_1(x_1) = f_2 b_1(x_2)$

(2) $\Longrightarrow b_2 f_1(x_1) \neq b_2 f_1(x_2)$

and we have $b_2 f_1 \neq f_2 b_1$. Since f' was any fault that was not a single fault at node 1, we have shown that $\alpha(f') = \alpha(f) \Longrightarrow K_f = K_{f'} = \{1\}$. Hence f is locatable by Definition 2.15(c).

Necessity (by contradiction of hypothesis)

Let $f = (f_1, b_2)$ be a locatable single fault at node 1 ($K_f = \{1\}$). Assume $\nexists$ any $x_1, x_2 \in X$ satisfying (1) and (2). Define f_2: $W \to Y$ as follows

(a) $\forall\, w \in \mathcal{R}(b_1)$, $f_2(w) = b_2 f_1(x)$ for some $x \in b_1^{-1}(w)$

(b) $\forall\, w \notin \mathcal{R}(b_1)$, $f_2(w)$ is arbitrary

Consider the fault $f' = (b_1, f_2)$. $K_{f'}$ is either $\{2\}$ or ϕ. In (a), the mapping is well-defined because our assumption says that it is not possible for two elements x_1 and x_2, in $b_1^{-1}(w)$ (i.e. $x_1 \underset{b_1}{\equiv} x_2$) to have different images under $b_2 f_1$, i.e.

$$f_1(x_1) \underset{b_2}{\not\equiv} f_1(x_2).$$

Let x be any element of X. Since $x \underset{b_1}{\equiv} x$ our definition of f_2 forces

$$f_2 b_1(x) = b_2 f_1(x)$$

and we have

$$f_2 b_1 = b_2 f_1$$

or

$$\alpha(f') = \alpha(f)$$

but

$$K_f \neq K_{f'}$$

a contradiction of hypothesis, concluding the proof.

Fault location at node 2 is characterized as follows:

Theorem 2.23

In the general two node system with single faults, a single fault at node 2, $f = (b_1, f_2)$, is locatable iff

$$\exists\, w \in \mathcal{R}(b1) \ni f_2(w) \notin \mathcal{R}(b2)$$

Proof: Sufficiency

Let w be an element of $\mathcal{R}(b1)$ and let $f_2(w) \in \overline{\mathcal{R}(b2)}$. Let $f' = (f_1, b_2)$ be any fault with $K_{f'} \neq K_f = \{2\}$. Since the only proper faults in the system are single faults (Definition 2.17) the second coordinate of f' must be b_2. Obviously, $\forall x \in X$, $b_2 f_1(x) \in \mathcal{R}(b2)$. Since $w \in \mathcal{R}(b1)$, $\exists x' \in X$ such that $b_1(x') = w$, and we have $f_2 b_1(x') = f_2(w) \notin \mathcal{R}(b2)$. We have just seen that $b_2 f_1(x') \in \mathcal{R}(b2)$, and therefore $b_2 f_1 \neq f_2 b_1$. As f' was any fault with $K_{f'} \neq K_f$, it must follow that $\forall f' \in F$, $\alpha(f') = \alpha(f) \Longrightarrow K_{f'} = K_f$, and K_f is locatable (Definition 2.15c).

Necessity by contradiction of hypothesis

Let $f = (b_1, f_{22})$ be allocatable single fault at node 2. Assume $\nexists$ any $w \in \mathcal{R}(b1)$ such that $f_2(w) \notin \mathcal{R}(b2)$. Let x be any element of X. Let $w = b_1(x)$ and $y = f_2 b_1(x) = f_2(w)$. Since $w \in \mathcal{R}(b1)$, our assumption forces $f_2(w) \in \mathcal{R}(b2)$, and $\exists\, w' \in W$ with $b_2(w') = f_2(w)$. Since x was any element of X, we have shown that

$$\forall x \in X,\ \exists\, w' \in W \ni b_2(w') = f_2 b_1(x).$$

Let f_1 be any mapping from X into W such that $\forall x \in X,\ b_2 f_1(x) = f_2 b_1(x)$. The above discussion insures that atleast one such mapping exists.

By Definition 2.17, $f' = (f_1, b_2) \in F$. $(K_{f'} \neq \{2\} = K_f)$. We have shown that $\alpha(f') = \alpha(f)$, and we have a contradiction of the hypothesis that f was locatable (Definition 2.15c). This concludes the proof of the theorem.

Theorem 2.24

In the general two node system with single faults, it is impossible for all single faults at node 1 to be locatable except in the degenerate case when $|W| = 1$.

Proof (by contradiction of hypothesis)

Assume all single faults at node 1 are locatable, and that $|W| \geq 2$; then there exists two distinct elements of W, (w_1 and w_2), and we can consider node faults f_1 and f_2 defined as follows:

$$f_1 : X \rightarrow W$$

$$f_1(x) = \begin{cases} w_2 & \text{if } x \in b_1^{-1}(w_1) \\ b_1(x) & \text{otherwise} \end{cases}$$

and $f_2: W \to Y$

$$f_2(w) = \begin{cases} b_2(w_2) & \text{if } w = w_1 \\ b_2(w) & \text{otherwise} \end{cases}$$

Consider now the faults $f = (f_1, b_2)$ and $f' = (b_1, f_2)$.

If $x \in b_1^{-1}(w_1)$ $\quad b_2 f_1(x) = b_2(w_2)$ (Definition of f_1)

$\quad f_2 b_1(x) = f_2(w_1) = b_2(w_2)$ (Definition of f_2)

If $x \notin b_1^{-1}(w_1)$ $\quad b_2 f_1(x) = b_2 b_1(x)$ (Definition of f_1)

$\quad f_2 b_1(x) = b_2 b_1(x)$ (Definition of f_2)

The last equation follows from the definition of f_2 and the fact that $(x \notin b_1^{-1}(w_1) \Longrightarrow b_1(x) \neq w_1)$. It follows that $b_2 f_1 = f_2 b_1$, or $\alpha(f) = \alpha(f')$, but $K_{f'} \neq \{1\} = K_f$ a contradiction of hypothesis.

We now show that no interesting two node networks have all single faults at node 2 locatable.

Theorem 2.25

In a general two node system with single faults having $|Y| \geq 2$, all single faults at node 2 are locatable iff

(1) b_1 is onto W

and (2) b_2 is a constant function.

Proof: Sufficiency

Let b_1 be onto W and let b_2 be a constant function. Let $f = (b_1, f_2)$ be any single fault at node 2. Since $b_2 \neq f_2$ (Definition 2.16), $\exists\, w \in W$ for which $b_2(w) \neq f_2(w)$. Since b_2 is a constant function, $\mathcal{R}(b2) = \{b_2(w)\}$ and since $f_2(w) \neq b_2(w)$, $f_2(w) \notin \mathcal{R}(b2)$; but $w \in \mathcal{R}(b1)$ because b_1 is onto W. By Theorem 2.13, f is locatable. Since f was any single fault at node 2, all single faults at node 2 must be locatable.

Necessity by Contradiction of Hypothesis

Let all single faults at node 2 be locatable and assume that b_1 is not onto W. Since $|Y| \geq 2$ and b_1 is not onto W, there must exist a single fault at node 2, $f = (b_1, f_2)$, that is masked (Theorem 2.12 and Lemma 2.1.2). Since $\alpha(f) = \alpha(b) = t$ (Definition 2.15a) and $K_f = \{2\} \neq \emptyset = K_b$, f is not locatable. This is a contradiction of the hypothesis and b_1 must be onto W.

Now, let all single faults at node 2 be locatable and assume that b_2 is not a constant function then there must exist two distinct elements in $\mathcal{R}(b2)$, y_1 and y_2. Let $w_1 \in b_2^{-1}(y_1)$ and $w_2 \in b_2^{-1}(y_2)$, and consider the mapping f_2 from W into Y defined as follows

$$f_2(w) = \begin{cases} y_2 & \text{if } w = w_1 \\ b_2(w) & \text{otherwise} \end{cases}$$

For the single fault $f = (b_1, f_2)$ at node 2,

$$K_f = \{2\} \neq \emptyset = K_b$$

If $w_1 \notin \mathcal{R}(b1)$, then $f_2 b_1 = b_2 b_1$ and $\alpha(f) = \alpha(b)$; but since $K_f \neq K_b$, f is not locatable (Definition 2.15c). This is a contradiction of the hypothesis. If $w_1 \in \mathcal{R}(b1)$, let $X_1 = b_1^{-1}(w_1)$. Consider the function, f_1, from X into W defined as follows

$$f_1(x) = \begin{cases} w_2 & \text{if } x \in X_1 \\ b_1(x) & \text{otherwise} \end{cases}$$

Let $f' = (f_1, b_2)$. Let x be any element of X. If $x \in X_1$, we have

$$b_2 f_1(x) = b_2(w_2) = y_2 \qquad \text{(Definition of } f_1\text{)}$$

and

$$f_2 b_1(x) = f_2(w_1) = y_2 \qquad \text{(Definition of } X_1\text{)}$$

If $x \notin X_1$, we have

$$b_2 f_1(x) = b_2 b_1(x) \qquad \text{(Definition of } f_1\text{)}$$

and

$$f_2 b_1(x) = b_2 b_1(x) \qquad \text{(Definition of } f_2\text{)}$$

because

$$b_1(x) \neq w_1.$$

In any case, we have $\forall x \in X$, $b_2 f_1(x) = f_2 b_1(x)$ and $\alpha(f) = \alpha(f')$ but $K_f = \{2\}$ and $2 \notin K_{f'}$. Then $K_f \neq K_{f'}$, a contradiction of hypothesis. The assumption must be incorrect, so that b_2 is a constant function.

Frequently we may wish to mask as many faults as possible within some cost or weight constraint while at the same time making all unmasked faults locatable from the system terminals. Although not impossible, this is also an unreasonable constraint as revealed by the following theorems.

Theorem 2.26

In the general two node system with single faults, all detectable single faults at node 1 are locatable iff b_2 is a constant function.

Proof: Sufficiency

If b_2 is a constant function, then all single faults at node 1 are masked (Theorem 2.8), and all detectable single faults at node 1 are vacuously locatable.

Necessity

This part of the proof is almost identical to the necessity proof for Theorem 2.24. Only a brief summary will be given here. The proof is again by contradiction of hypothesis. Let all detectable faults at node 1 be locatable and assume that b_2 is not a constant function; then $\exists\ w_1, w_2 \in W$ such that $b_2(w_2) \neq b_2(w_1)$. Define f_1 as in the proof of Theorem 2.24 and consider the single fault $f = (f_1, b_2)$ at node 1. Let $x \in b_1^{-1}(w_1)$, then

$$b_2 b_1(x) = b_2(w_1)$$

and

$$b_2 f_1(x) = b_2(w_2) \qquad \text{(Definition of } f_1\text{)}$$

and we have $b_2 b_1(x) \neq b_2 f_1(x)$

and $b_2 b_1 \neq b_2 f_1$

or $\alpha(b_1, b_2) \neq \alpha(f_1, b_2)$

thus (f_1, b_2) is detectable (Definition 2.15b). In the proof of Theorem 2.24, it is shown that (f_1, b_2) is not locatable, a contradiction of hypothesis, and the theorem is proved.

We now show that if only detectable faults at node 2 are to be locatable, one of the constraints of Theorem 2.25 is removed.

Theorem 2.27

In a general two node system with single faults all detectable single faults at node 2 are locatable if and only if b_2 is a constant function.

Proof:

To show sufficiency, let b_2 be a constant function (i.e., $\mathcal{R}(b_2) = \{y\}$) and let (b_1, f_2) be any detectable single fault at node 2, then $f_2 | \mathcal{R}(b_1) \neq b_2 | \mathcal{R}(b_1)$ (Corollary 2.3.1). There must exist $w \in \mathcal{R}(b_1)$ such that $f_2(w) \neq b_2(w) = y$. Since $\mathcal{R}(b_2) = \{y\}$, $f_2(w) \notin \mathcal{R}(b_2)$ and (b_1, f_2) is locatable (Theorem 2.23), but since (b_1, f_2) was any detectable single fault at node 2, it must be true that all detectable single faults at node 2 are locatable.

To show necessity, let all detectable single faults at node 2 be locatable and assume that b_2 is not a constant function. There must exist two distinct elements, y_1 and y_2, in $\mathcal{R}(b_2)$. Let w_1 be any element of $b_2^{-1}(y_1)$ and let w_2 be any element of $b_2^{-1}(y_2)$. Since $y_1 \neq y_2$, we must have $w_1 \neq w_2$. If neither w_1 nor w_2 are in the range of b_1, there must exist $w_3 \in \mathcal{R}(b_1)$ such that $w_1 \neq w_3 \neq w_2$

because the range cannot be empty. Since $y_1 \neq y_2$, $b_2(w_3) = y_3$ must be different from one of them. Without loss of generality, let $y_3 \neq y_1$. Consider the function, f_2, from W into Y defined as follows

$$f_2(w) = \begin{cases} y_1 & \text{if } w = w_3 \\ b_2(w) & \text{otherwise} \end{cases}$$

Now $f = (b_1, f_2)$ is a single fault at node 2 (Definition 2.17) and since

$$f_2(w_3) = y_1 \neq y_3 = b_2(w_3)$$

f is detectable (Corollary 2.3.1). $K_f = \{2\}$. Now, let $X_3 = b_1^{-1}(w_3)$ and consider the function, f_1, from X into W defined as follows

$$f_1(x) = \begin{cases} w_1 & \text{if } x \in X_3 \\ b_1(x) & \text{otherwise} \end{cases}$$

Now, $f' = (f_1, b_2)$ is a fault (Definition 2.17). Let x by any element of X. If $x \in X_3$, we have

$$f_2 b_1(x) = f_2(w_3) = y_1$$

and

$$b_2 f_1(x) = b_2(w_1) = y_1$$

if $x \notin X_3$, we have $b_1(x) \neq w_3$ and

$$b_2 f_1(x) = b_2 b_1(x) \qquad \text{(Definition of } f_1\text{)}$$

$$f_2 b_1(x) = b_2 b_1(x) \qquad \text{(Definition of } f_2\text{)}$$

In any case, $b_2 f_1(x) = f_2 b_1(x)$ and we have $b_2 f_1 = f_2 b_1$ or $\alpha(f) = \alpha(f')$, but since $K_f = \{2\}$ and $2 \notin K_{f'}$, we must have $K_f \neq K_{f'}$. Therefore, f is not locatable (Definition 2.15c) and we have a contradiction of

the hypothesis. If $w_1 \in \mathcal{R}(b1)$, consider the function f_2' from W into Y defined as follows:

$$f_2'(w) = \begin{cases} y_2 & \text{if } w = w_1 \\ b_2(w) & \text{otherwise} \end{cases}$$

Now, $g = (b_1, f_2')$ is a single fault at node 2 (Definition 2.17) and since

$$f_2'(w_1) = y_2 \neq y_1 = b_2(w_1)$$

g is detectable (Corollary 2.3.1). $K_g = \{2\}$. Now, let $X_1 = b_1^{-1}(w_1)$ and consider the function f_1' from X into W defined as follows

$$f_1'(x) = \begin{cases} w_2 & \text{if } x \in X_1 \\ b_1(x) & \text{otherwise} \end{cases}$$

Now $g' = (f_1', b_2)$ is a fault in the system (Definition 2.17). Let x be any element of X. If $x \in X_1$, we have

$$b_2 f_1'(x) = b_2(w_2) = y_2$$

and

$$f_2' b_1(x) = f_2'(w_1) = y_2$$

If $x \notin X_1$, we have $b_1(x) \neq w_1$ and

$$b_2 f_1'(x) = b_2 b_1(x) \qquad \text{(Definition of } f_1')$$

$$f_2' b_1(x) = b_2 b_1(x) \qquad \text{(Definition of } f_2')$$

In any case, we have $b_2 f_1'(x) = f_2' b_1(x)$. Therefore, $b_2 f_1' = f_2' b_1$ or $\alpha(g') = \alpha(g)$; but since $K_g = \{2\}$ and $2 \notin K_{g'}$, we have $K_g \neq K_{g'}$ and g is not locatable (Definition 2.15c). This is a contradiction of our hypothesis. A similar development shows that $w_2 \in \mathcal{R}(b1)$ also leads to a contradiction of hypothesis. Hence, the assumption is false and b_2 must be a constant function. This concludes the proof of Theorem 2.27.

For completeness, we now state necessary and sufficient conditions for locating all faults in two node systems.

Theorem 2.28

In a general two node system with single faults, all single faults are locatable if and only if $|W| = 1$.

Proof

To show sufficiency, let $|W| = 1$. Then obviously b_1 is onto W and b_2 is a constant function thus all single faults at node 2 are locatable (Theorem 2.25 and Lemma 2.9.1). All single faults at node 1 are locatable by Theorem 2.24.

If all single faults are locatable, then obviously all single faults at node 1 are locatable and $|W| = 1$ (Theorem 2.24). This shows necessity and concludes the proof of Theorem 2.28.

Thus, 2-node systems must have degenerate behavior if all single faults are to be locatable. We note, however, that in the limit node 1 is more restrictive than node 2 with respect to total fault location. It will be interesting to see if this difference becomes more

apparent or less apparent as the fault location constraint is relaxed.

Theorem 2.29

In a general two node system with single faults, all detectable single faults are locatable if and only if b_2 is a constant function.

Proof

To show sufficiency, let b_2 be a constant function. Then, when $|Y| \geq 2$, all detectable single faults at node 2 are locatable (Theorem 2.27). When $|Y| = 1$, all single faults at node 2 are locatable (Lemma 2.9.1) and hence all detectable single faults at node 2 are locatable. All detectable single faults at node 1 are locatable by Theorem 2.26.

To show necessity, let all detectable single faults at node 1 be locatable, then b_2 is a constant function (Theorem 2.26).

These results indicate the restrictive nature of locatable faults in two node systems. Location of a large number of faults is virtually impossible. Therefore, we will introduce a more general notion of locatability, which includes our former notion of fault location as a special case. In this more general setting, fault "location" will not be impossible to achieve.

If $C = (D, S, F, b)$ is a combinational network having an (n, m, k, ℓ)-digraph D (n inputs, m outputs, k nodes, ℓ lines), signal set S, fault set F, and fault-free structure b; if f is any fault from the set F; and if $\alpha(f)$ is the system behavior under f, then recall that

f is masked iff $\alpha(f) = \alpha(b)$

f is detectable iff $\alpha(f) \neq \alpha(b)$

In other words, a fault f is masked when the system behavior under fault f is the same as the fault-free behavior, and is detectable otherwise.

If K denotes the set of nodes of C; if $K_f = \{i \mid f_i \neq b_i,\ 1 \leq i \leq k\}$ denotes the set of "faulty" nodes under fault f: and if [B, A] denotes a closed interval in the partially ordered set of subsets of K, i.e. $B, A \subseteq K$, $B \subseteq A$, and $[B, A] = \{X \mid X \subseteq K,\ B \subseteq X \subseteq A\}$; then

Definition 2.28

f is [B, A]-locatable if, for all $g \in F$, $\alpha(g) = \alpha(f) \Rightarrow K_g \in [B, A]$.

This formally states that, if the system behavior is $\alpha(f)$ and f is [B, A]-locatable, then we can infer that the set of faulty nodes includes set B and is included in set A. In other words without any knowledge of the present system structure, we are able to bound the set of faulty nodes by observing the system behavior. In the lattice of subsets of K, B is a lower bound for the set of faulty nodes and A is an upper bound.

Obviously, if the set of faulty nodes includes the set B it also includes any subset B' of B, and if the set of faulty nodes is a subset of A it is also a subset of any superset, A', of A. These simple observations lead to the following theorem.

Theorem 2.30

In an (n, m, k, ℓ)-combinational network C = (D, S, F, b), if fault f is [B, A]-locatable and if [B', A'] is a closed interval of the partially ordered set of subsets of K that includes interval [B, A], then f is [B', A']-locatable.

Proof

If f is [B, A]-locatable, then for all g ϵ F, $\alpha(g) = \alpha(f) \Longrightarrow$ $K_g \epsilon [B, A]$, but since $[B, A] \subseteq [B', A']$, we have $K_g \epsilon [B', A']$ and hence f is [B', A'] locatable.

Theorem 2.30 suggests that the minimum interval [B, A] for which a fault f is [B, A]-locatable would be a good indicator of the "locatability" of the fault f because it represents the most information about the set of faulty nodes that can be deduced from knowledge of the system behavior.

Definition 2.29

The locatability L(f) of a fault f is the minimum interval [B, A] for which f is [B, A]-locatable, i.e., f is not [B', A']-locatable for any proper subinterval of [B, A].

That L(f) is well-defined for a given fault f is a direct consequence of the following theorem.

Theorem 2.31

In a combinational network C = (D, S, F, b), if f is [B, A]-locatable and [B', A']-locatable, then f is [B $\cup$ B', A $\cap$ A']-locatable.

Proof

Since f is [B, A]-locatable, we have

$$\forall\ g \in F,\ \alpha(g) = \alpha(f) \Longrightarrow B \subseteq K_g \subseteq A$$

and since f is [B', A']-locatable, we have

$$\forall\ g \in F,\ \alpha(g) = \alpha(f) \Longrightarrow B' \subseteq K_g \subseteq A'$$

Now

$$B, B' \subseteq K_g \Longrightarrow B \cup B' \subseteq K_g$$

and

$$A, A' \supseteq K_g \Longrightarrow A \cap A' \supseteq K_g$$

and we have

$$\forall\ g \in F,\ \alpha(g) = \alpha(f) \Longrightarrow B \cup B' \subseteq K_g \subseteq A \cap A'$$

i.e. f is $[B \cup B',\ A \cap A']$-locatable concluding the proof of the theorem.

Suppose that a fault f is [B, A]-locatable and [B', A']-locatable but not [B'', A'']-locatable for any proper subinterval [B'', A''] of [B, A] or [B', A']. From Theorem 2.31, f is $[B \cup B', A \cap A']$-locatable, but since $[B \cup B', A \cap A'] \subseteq [B, A]$ it must be that $[B \cup B', A \cap A'] = [B, A]$ for otherwise we would violate the second part of the hypothesis. Similarly, we conclude that $[B \cup B', A \cap A'] = [B', A']$ and further that [B, A] = [B', A']. Thus, L(f) is well-defined.

It should be noted that, although L(f) = [B, A] establishes a lower bound B and an upper bound A on the faulty node set when the system behavior is $\alpha(f)$, it does not imply that any fault α-equivalent to f

actually has faulty node set B or faulty node set A. In fact, if $L(f) = [B, A]$ then B is the greatest lower bound (glb) and A is the least upper bound (lub) of $\{K_g | \alpha(g) = \alpha(f)\}$ in the partially ordered set of subsets of K. Since we have a strict partial ordering, it is clear that the lub and glb of a collection of subsets need not be included in the collection.

As an example, consider a three node system with fault-free structure $b = (b_1, b_2, b_3)$ and fault-free behavior $\alpha(b)$. Let $f = (f_1, b_2, b_3)$ and $f' = (b_1, f_2, b_3)$ be two faults with the same faulty behavior, i.e. $\alpha(f) = \alpha(f') \neq \alpha(b)$. Further, let no other system fault produce the behavior $\alpha(f)$. Then $L(f) = [\phi, \{1, 2\}]$. Note that $K_f = \{1\}$, $K_{f'} = \{2\}$, glb $\{K_f, K_{f'}\} = \phi$ and lub $\{K_f, K_{f'}\} = \{1, 2\}$.

Recall that α is a mapping whose domain is the fault set F. As such it induces an equivalence relation $\underset{\alpha}{\equiv}$, on the set F, namely

$$f \underset{\alpha}{\equiv} f' \iff \alpha(f) = \alpha(f')$$

The close relationship between $\underset{\alpha}{\equiv}$ and locatability is demonstrated by the following theorem.

Theorem 2.32

In a combinational network $C = (D, S, F, b)$, a fault f is $[B, A]$-locatable if and only if all faults α-equivalent to f are $[B, A]$-locatable.

Proof

If f is [B, A]-locatable, then

$$\forall\, g \in F,\ \alpha(g) = \alpha(f) \Longrightarrow K_g \in [B, A] \Longrightarrow B \subseteq K_g \subseteq A$$

Let h be any element of F such that $\alpha(h) = \alpha(f)$, i.e., $h \underset{\alpha}{\equiv} f$. Let e be any element of F that is α-equivalent to h, then

$$\alpha(e) = \alpha(h) = \alpha(f) \Longrightarrow K_e \in [B, A]$$

Hence, h is [B, A]-locatable. Since h was any element of F α-equivalent to f, it must be true that all faults α-equivalent to f are [B, A]-locatable.

Conversely, if all faults α-equivalent to f are [B, A]-locatable, then since $f \underset{\alpha}{\equiv} f$ certainly f is [B, A]-locatable.

It is now apparent that locatability is an invariant of $\underset{\alpha}{\equiv}$, i.e. all faults in the same equivalence class of α have the same locatability. Stated as a corollary, this becomes:

Corollary 2.32.1

Locatability is an invariant of $\underset{\alpha}{\equiv}$ i.e. $f \underset{\alpha}{\equiv} f' \Longrightarrow L(f) = L(g)$.

Proof

If $f \underset{\alpha}{\equiv} f'$, by Theorem 2.32 f is L(f') locatable. This implies $L(f) \subseteq L(f')$. Similarly f' is L(f) locatable which implies $L(f') \subseteq L(f)$. Therefore, $L(f) = L(f')$.

Since locatability is an invariant of $\underset{\alpha}{\equiv}$, we will often refer to the locatability of an α-equivalence class as being the locatability of any fault in the class.

To demonstrate the fact that locatability is not a complete set of invariants for $\underset{\alpha}{\equiv}$, let us return to our example. It is certainly conceivable that two other faults $g' = (g_1, b_2, b_3)$ and $g'' = (b_1, g_2, b_3)$ might exist with $\alpha(f') \neq \alpha(g') = \alpha(g'') \neq \alpha(b)$. Furthermore, if no other fault produces behavior $\alpha(g')$, then the locatability of g' and g'' is also $[\phi, \{1,2\}]$. Since $\alpha(g') \neq \alpha(f')$, then locatability is not a complete set of invariants for $\underset{\alpha}{\equiv}$.

If one wishes only to locate faulty nodes to within some subset $A \subseteq K$ (i.e. to locate to within "module" A), then

Definition 2.30

In a combinational network $C = (D, S, F, b)$, a fault f is A-locatable ($A \subseteq K$) if f is $[\phi, A]$-locatable.

In other words, f is A-locatable if, for all $g \in F$, $\alpha(g) = \alpha(f) \Rightarrow K_g \subseteq A$. Here, the lower bound on the set of faulty nodes is the empty set ϕ, indicating that we cannot guarantee that any particular node is at fault upon observation of behavior $\alpha(f)$. We can, however, restrict our attention to nodes in the set A, since they are the only ones that can be faulty. Note that all faults are K-locatable.

Theorem 2.33

In a combinational network $C = (D, S, F, b)$, a fault f is A-locatable if and only if all faults α-equivalent to f are A-locatable.

Proof

Immediate from Theorem 2.32.

Theorem 2.34

In an (n, m, k, ℓ)-combinational network $C = (D, S, F, b)$, if a fault f is A-locatable and $A \subseteq A' \subseteq K$, then f is A'-locatable.

Proof

f A-locatable $\Longrightarrow$ f is $[\phi, A]$-locatable and hence also $[\phi, A']$-locatable (Theorem 2.30).

Theorem 2.35

In an (n, m, k, ℓ)-combinational network $C = (D, S, F, b)$, if a fault f is both A-locatable and A'-locatable, then f is $(A \cap A')$-locatable.

Proof

Immediate from Theorem 2.31.

If one wishes to specify precisely the set of faulty nodes upon observation of system behavior with no area of uncertainty, we must have

Theorem 2.36

In a combinational network $C = (D, S, F, b)$, a fault f is locatable if and only if f is $[K_f, K_f]$-locatable.

Proof

f is $[K_f, K_f]$-locatable if and only if for all $g \in F$, $\alpha(g) = \alpha(f) \Longrightarrow K_g = K_f$. Indeed, the upper and lower bounds on the set of faulty nodes are identical and hence there can be no uncertainty as to which nodes are faulty. As a direct result of Theorem 2.32, we have

Theorem 2.37

In a combinational network $C = (D, S, F, b)$, a fault f is locatable if and only if all faults α-equivalent to f are locatable.

Location of Masked Faults

That the set of masked faults constitutes an α-equivalence class should be apparent from Definition 2.15. In particular, the set of masked faults is the α-equivalence class designated $[b]_\alpha$ i.e. the α-equivalence class containing the 0-fault. Since locatability is an invariant of $\underset{\alpha}{\equiv}$ we have

Theorem 2.38

If f is a masked fault in a combinational network $C = (D, S, F, b)$, then

$$L(f) = L(b)$$

In other words, the locatability of the class of masked faults is the locatability of the zero fault.

As shown by the following theorem, L(b) can be completely specified by a single set instead of the pair of sets required to describe the locatability of an arbitrary fault.

Theorem 2.39

In a combinational network $C = (D, S, F, b)$, if a masked fault is $[B, A]$-locatable, then $B = \phi$.

Proof

Let masked fault f be $[B, A]$-locatable, then

$$\forall g \in F,\ \alpha(g) = \alpha(f) \Rightarrow B \subseteq K_g \subseteq A$$

Since f is masked, $\alpha(b) = \alpha(f)$, and $B \subseteq K_b \subseteq A$. But $K_b = \phi$, hence $B = \phi$.

Since L(b) is always $[\phi, A]$ for some $A \subseteq K$, we will refer to A as the locatability of b. Since $L(f) = L(b)$ for every masked fault f, the set A is the locatability of all masked faults. An important implication of this result is that an inverse relationship exists between the locatability of masked faults (or equivalently the locatability of the 0-fault) and the size of the masked fault set. This may be more apparent from the following theorem.

Theorem 2.40

In a combinational network $C = (D, S, F, b)$, a masked fault f is A-locatable if and only if, for every $g \in F$, $K_g \not\subseteq A \Rightarrow$ g is detectable.

Proof

If f is A-locatable, then all masked faults are A-locatable (Theorem 2.33), i.e. for every masked fault g, $K_g \subseteq A$. Hence, $K_g \not\subseteq A \Rightarrow$ g is not masked, i.e. g is detectable.

If, $\forall g \in F$, $K_g \not\subseteq A \Rightarrow$ g is detectable, then all masked faults f must have $K_f \subseteq A$, i.e. f is A-locatable.

It should now be clear that if $L(b) = A$, then for all masked faults f, $K_f \subseteq A$. If L(b) is made smaller, then we have a more severe constraint on the number of masked faults. In the extreme, if b is locatable, that is $L(b) = \emptyset$, then all proper faults must be detectable and b is the only masked fault.

Similarly, if we mask more faults, we must gradually reduce the locatability of the masked faults. In this extreme, when all faults are masked, $L(b) = K$, the whole node set. In other words, when the system behavior is $\alpha(b)$, any node in the system may be faulty.

In general, a fault can be K_f-locatable but not locatable. However, in the case of a single detectable fault we have

Theorem 2.41

In any combinational network $C = (D, S, F, b)$, a single detectable fault at node i, $f = (b_1, \ldots, b_{i-1}, f_i, b_{i+1}, \ldots, b_k)$, is $\{i\}$-locatable if and only if it is locatable.

Proof

If f is $\{i\}$-locatable and detectable, then

$$\forall g \in F,\ \alpha(g) = \alpha(f) \Rightarrow K_g \subseteq \{i\} \Rightarrow K_g = \emptyset \text{ or } K_g = \{i\}$$

But, since f is detectable, $\alpha(g) = \alpha(f) \Rightarrow$ g is detectable and, hence $K_g \neq \emptyset$. We then must have

$$\forall g \in F,\ \alpha(g) = \alpha(f) \Rightarrow K_g = \{i\} = K_f$$

i.e. f is locatable.

The converse is trivial. If f is locatable, then, by definition, f is $[\{i\}, \{i\}]$-locatable. Since $[\{i\}, \{i\}] \subseteq [\emptyset, \{i\}]$, f is $[\emptyset, \{i\}]$-locatable (from Theorem 2.30) i.e. f is $\{i\}$-locatable.

Fault Location in General System Forms

We now seek necessary and sufficient conditions for faults to be K_f-locatable in general system forms with single faults.

Lemma 2.42.1

A detectable fault $f = (f_1, f_2, f_3)$ in a general system form with single faults is $\{1, 2\}$-locatable iff

$$\equiv \quad \equiv \quad \not\subseteq \quad \equiv$$
$$b_{1u} \bigcap b_2 b_{1v} \not\subseteq \alpha(f)$$

i.e., $\exists x_1, x_2 \in X \ni$

(1) $x_1 \underset{b_{1u}}{\equiv} x_2$

(2) $x_1 \underset{b_2 b_{1v}}{\equiv} x_2$

(3) $x_1 \underset{\alpha(f)}{\not\equiv} x_2$

Proof

To show sufficiency, let $\exists x_1, x_2 \in X$ satisfy (1), (2), and (3). Also, let f' be any fault of the form (b_1, b_2, f_3), then

$$x_1 \underset{b_{1u}}{\equiv} x_2 \Longrightarrow b_{1u}(x_1) = b_{1u}(x_2) \qquad \text{(i)}$$

and

$$x_1 \underset{b_2 b_{1v}}{\equiv} x_2 \Longrightarrow b_2 b_{1v}(x_1) = b_2 b_{1v}(x_2) \qquad \text{(ii)}$$

From construction

$$\alpha(f')(x_1) = f_3(b_2 b_{1v}(x_1), b_{1u}(x_1))$$

Applying (i) and (ii), we get

$$\alpha(f')(x_1) = f_3(b_2\, b_{1v}(x_2),\ b_{1u}(x_2)) = \alpha(f')(x_2)$$

But

$$x_1 \underset{\alpha(f)}{\not\equiv} x_2 \Rightarrow \alpha(f)(x_1) \neq \alpha(f)(x_2)$$

and it follows that $\alpha(f) \neq \alpha(f')$. Since f' was any fault of the form (b_1, b_2, f_3) it follows that

$$\alpha(f) = \alpha(f') \Rightarrow K_f \subset \{1, 2\}$$

and f' is $\{1, 2\}$-locatable.

To show necessity, let $f = (f_1, f_2, f_3)$ be any detectable fault that is $\{1, 2\}$-locatable. Necessity is shown in three parts by contradiction. First, assume $\exists\, x_1, x_2 \in X$ such that $x_1 \underset{\alpha(f)}{\not\equiv} x_2$, then $\alpha(f)$ must be a constant function and $\exists\, y \in Y$ such that $\forall x \in X$, $\alpha(f)(x) = y$. Since f is detectable, $\alpha(f) \neq \alpha(b)$, hence $b_3(w, u) \neq y$ for some $(w, u) \in W \times U$.

Consider the function f_3' from $W \times U$ into Y defined as follows:

$$\forall (w, u) \in W \times U,\ f_3'(w, u) = y$$

From Definition 2.17, we know that $f' = (b_1, b_2, f_3')$ is an element of F. Obviously, $3 \in K_{f'}$ and $K_{f'} \not\subset \{1, 2\}$, but $\forall x \in X$, $\alpha(f')(x) = y = \alpha(f)(x)$ and $\alpha(f') = \alpha(f)$. Since this is a contradiction of hypothesis (Definition 2.28), the assumption must be false and (3) holds for some $x_1, x_2 \in X$.

Now assume that, among all pairs $x_1, x_2 \in X$ for which (3) holds, that $x_1 \underset{b_2\, b_{1v}}{\not\equiv} x_2$. Consider the mapping $f_3'' : W \times U \rightarrow Y$ defined as follows:

$$f_3''(b_2\, b_{1v}(x),\ b_{1u}(x)) = f_3(f_2 f_{1v}(x), f_{1u}(x))$$

This mapping is well defined because by assumption, whenever $b_2\, b_{1v}(x_1) = b_2\, b_{1v}(x_2)$ then $\alpha(f)(x_1) = \alpha(f)(x_2)$. From Definition 2.17, $f'' = (b_1, b_2, f_3'')$ is an element of F. However, $\forall x \in X$

$$\alpha(f)(x) = f_3(f_2 f_{1v}(x), f_{1u}(x)) = f_3''(b_2\, b_{1v}(x),\ b_{1u}(x)) = \alpha(f'')(x)$$

and since f is detectable, $f'' \neq b$, and $K_{f''} = \{3\} \not\subseteq \{1, 2\}$. Hence f is not $\{1, 2\}$-locatable (Definition 2.28). This is a contradiction of hypothesis, hence the assumption must be false and $\exists\, x_1, x_2 \in X$ such that (2) and (3) are true.

Now assume that for all pairs $x_1, x_2 \in X$ that satisfy (2) and (3) it is true that $x_1 \not\equiv_{b_{1u}} x_2$. Again, the mapping f_3'' is well defined because of our assumption. As above, $\forall x \in X$, $\alpha(f)(x) = \alpha(f')(x)$ and $K_{f''} \not\subseteq \{1, 2\}$ hence f is not $\{1, 2\}$-locatable, a contradiction of hypothesis. Our assumption is false, and there must exist $x_1, x_2 \in X$ that satisfy (1), (2), and (3). This concludes the proof of Lemma 2.42.1.

Lemma 2.42.2

A detectable fault $f = (f_1, f_2, f_3)$ in a general system form with single faults is $\{2, 3\}$-locatable iff

$$\exists\, y \in \mathcal{R}(\alpha(f)) \text{ such that } \forall u \in U,\ h_u^{-1}(y) \cap \mathcal{R}(b2) = \phi$$

Proof

To show sufficiency, let $\exists\, y \in \mathcal{R}(\alpha(f))$ such that $\forall u \in U$, $h_u^{-1}(y) \cap \mathcal{R}(b2) = \phi$, and let f' be any fault for which $K_{f'}$ is not a

subset of $\{2, 3\}$. By Definition 2.17, f' must have the form (f_1, b_2, b_3). Since $y \in \mathcal{R}(\alpha(f))$, $\exists\, x \in X$ for which $\alpha(f)(x) = y$. Assume that

$$\alpha(f')(x) = b_3(b_2\, f_{1w}(x),\ f_{1u}(x)) = y$$

then

$$h_{f_{1u}(x)}(b_2\, f_{1v}(x)) = y \text{ (Definition 2.24)}$$

which contradicts our hypothesis, hence it must be true that

$$\alpha(f')(x) \neq y$$

and that

$$\alpha(f') \neq \alpha(f)$$

Since f' was any fault for which K_f' is not a subset of $\{2, 3\}$, we have

$$\forall f' \in F,\ \alpha(f') = \alpha(f) \Longrightarrow K_f' \subseteq \{2, 3\}$$

and f is $\{2, 3\}$-locatable.

To show necessity, let $f = (f_1, f_2, f_3)$ be $\{2, 3\}$-locatable and assume that $\nexists\, y \in \mathcal{R}(\alpha(f))$ such that $\forall u \in U$, $h_u^{-1}(y) \cap \mathcal{R}(b2) \neq \phi$. Also, let x be any element of X and let $y = \alpha(f)(x)$, then $y \in \mathcal{R}(\alpha(f))$. By assumption, $\exists\, u \in U$ such that $h_u^{-1}(y) \cap \mathcal{R}(b2) \neq \phi$. Let w be an arbitrary element of $h_u^{-1}(y) \cap \mathcal{R}(b2)$, then certainly $w \in \mathcal{R}(b2)$ and $\exists\, v \in V$ for which $b_2(v) = w$. Consider the mapping $f_1'\colon X \rightarrow V \times U$ defined as follows:

$$f_1'(x) = (v, u)$$

where v and u are determined as in the previous discussion. Now, $f' = (f_1', b_2, b_3)$ is a fault of the system (Definition 2.17), and we have for any $x \in X$,

$$\alpha(f')(x) = b_3(b_2\, f_{1v}'(x),\ f_{1u}'(x))$$

$$= b_3(b_2(v), u)$$

$$= b_3(w, u)$$

$$= h_u(w)$$

$$\alpha(f')(x) = y = \alpha(f)(x)$$

where u, v, and w are as discussed. But since f is detectable, $f' \neq b$, and $K_{f'}$ is not a subset of $\{2, 3\}$ and hence f is not $\{2, 3\}$-locatable. This is a contradiction of hypothesis, therefore the assumption is false and $\exists\, y \in \mathcal{R}(\alpha(f))$ such that $\forall u \in U$, $h_u^{-1}(y) \cap \mathcal{R}(b2) = \phi$. This concludes the proof of Lemma 2.42.2.

Theorem 2.42

A detectable fault $f = (f_1, f_2, f_3)$ in a general system form with single faults is $\{2\}$-locatable iff

(1) $\equiv$ b_{1u} b_2 b_{1v} $\not\equiv$ $\alpha(f)$

and

(2) $\exists\, y \in \mathcal{R}(\alpha(f))$ such that $\forall u \in U$, $h_u^{-1}(y) \cap \mathcal{R}(b2) = \phi$.

Proof

To show sufficiency, let (1) and (2) be true, then f is $\{1, 2\}$-locatable by Lemma 2.42.1 and $\{2, 3\}$-locatable by Lemma 2.42.2.

Since $\{1,2\} \cap \{2,3\} = \{2\}$, f is $\{2\}$-locatable by Theorem 2.35.

To show necessity, let $f = (f_1, f_2, f_3)$ be a detectable fault that is $\{2\}$-locatable, then, f is also $\{1,2\}$-locatable and $\{2,3\}$-locatable by Theorem 2.34 and (1) and (2) follow from Lemmas 2.42.1 and 2.42.2 respectively, proving the theorem.

Theorem 2.43

A single detectable fault f at node 2 in a general system form with single faults is locatable iff it is $\{2\}$-locatable.

Proof

To show sufficiency, let f be $\{2\}$-locatable, i.e. f is $[\phi, \{2\}]$-locatable. However, $\alpha(g) = \alpha(f) \Longrightarrow K_g \neq \phi$ or otherwise $\alpha(g) = t \neq \alpha(f)$. Hence f is $[\{2\}, \{2\}]$-locatable or simply locatable.

To show necessity, let f be locatable then f is $[\{2\}, \{2\}]$-locatable and hence $[\phi, \{2\}]$-locatable (Theorem 2.30) i.e. f is $\{2\}$-locatable.

We now consider the important question of the location of masked faults. At first, this may seem to be a contradiction in terms, and indeed, no masked proper fault f can be $[K_f, K_f]$-locatable or locatable in the strict sense of earlier sections. However, it is feasible for a fault f to be K_f-locatable as we shall now demonstrate.

From Theorem 2.40, we

Lemma 2.44.1

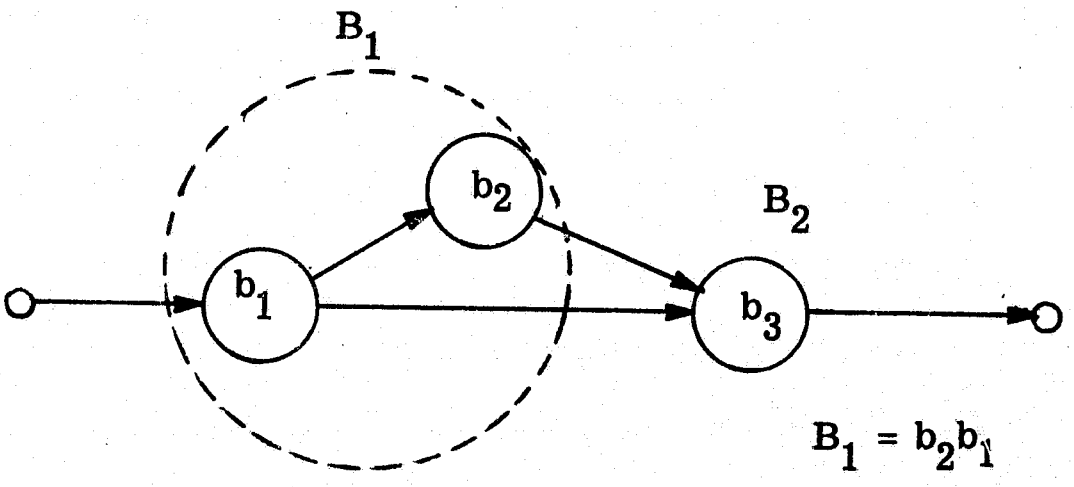
A masked fault in a general system form with single faults is $\{1, 2\}$-locatable if and only if all single faults at node 3 are detectable.

Lemma 2.44.2

All single faults at node 3 in a nontrivial general system form with single faults are detectable if and only if $b_2 b_1$ is onto $W \times U$.

Proof

Let $f = (b_1, b_2, f_3)$ be any single fault at node 3. Since b_1 and b_2 are known, we can combine them into a single node as shown below. The new circuit is a two-node system and the lemma follows directly from Theorem 2.12.



From Lemmas 2.44.1 and 2.44.2, we have

Theorem 2.44

A masked fault in a general system form with single faults is $\{1,2\}$-locatable if and only if $b_2 b_1$ is onto $W \times U$.

The following statement follows directly from Theorem 2.33.

Theorem 2.45

All masked faults in a general system form with single faults are $\{1,2\}$-locatable if and only if $b_2 b_1$ is onto $W \times U$.

We now consider the node set $\{2,3\}$. From Theorem 2.40,

Lemma 2.46.1

A masked fault in a general system form with single faults is $\{2,3\}$-locatable if and only if all single faults at node 1 are detectable.

Lemma 2.46.2

All single faults at node 1 in a general system form with single faults are detectable if and only if

$$|b_3\, b_2^{-1}(y)| = 1 \qquad \forall y \in \mathcal{R}(b)$$

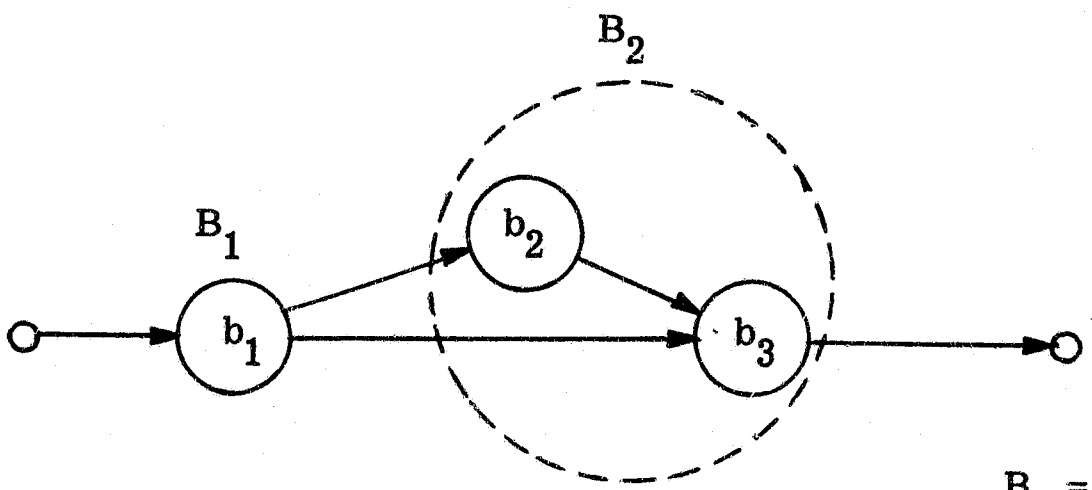
where

$$b_3 b_2 : V \times U \to Y$$

$$b_3 b_2(v,u) = b_3(b_2(v),u)$$

Proof

Let $f = (f_1, b_2, b_3)$ be any single fault at node 1. Since, b_2 and b_3 are known, we can combine them into a single node as shown below.



$B_1 = b_1$

$B_2 = b_3 b_2$

The new circuit is a two node system and the lemma follows directly from Theorem 2.11.

From Lemmas 2.46.1 and 2.46.2 we have

Theorem 2.46

A masked fault in a general system form with single faults is $\{2, 3\}$-locatable if and only if

$$|b_3\, b_2^{-1}(y)| = 1 \qquad \forall y \in \mathcal{R}(b)$$

From Theorem 2.33 we obtain the following version of Theorem 2.46.

Theorem 2.47

All masked faults in a general system form with single faults are $\{2,3\}$-locatable if and only if

$$|b_3 b_2^{-1}(y)| = 1 \qquad \forall y \in \mathcal{R}(b)$$

We are now in a position to discuss locatability of faults in a general system.

Theorem 2.48

A masked fault in a general system form with single faults is $\{2\}$-locatable if and only if

(1) $b_2 b_1$ is onto $W \times U$

(2) $|b_3 b_2^{-1}(y)| = 1 \qquad \forall y \in \mathcal{R}(b)$

Proof

To show necessity, let f be a masked fault that is $\{2\}$-locatable. Since $[\phi,\{2\}] \subseteq [\phi,\{2,3\}]$, f is $\{2,3\}$-locatable (Theorem 2.30) and hence (2) holds (Theorem 2.46). Similarly, since $[\phi,\{2\}] \subseteq [\phi,\{1,2\}]$, f is $\{1,2\}$-locatable (Theorem 2.30)and therefore (1) holds (Theorem 2.44).

To show sufficiency, let (1) and (2) be true, then (1) implies that f is $\{1,2\}$-locatable (Theorem 2.44) and (2) implies that f is $\{2,3\}$-locatable (Theorem 2.46). It then follows that f is $\{2\}$-locatable (Theorem 2.35).

From Theorem 2.33 we have

Theorem 2.49

All masked faults in a general system with single faults are {2}-locatable if and only if

(1) $b_2 b_1$ is onto $W \times U$

(2) $|b_3 b_2^{-1}(y)| = 1 \quad \forall y \in \mathcal{R}(b)$

3. FAULT DIAGNOSIS IN SEQUENTIAL MACHINES AND NETWORKS

Introduction

Fault diagnosis and subsequent repair is an important aspect of satisfying the high reliability requirements of many present-day digital computers. The problem considered here is the fault diagnosis of digital networks.

In diagnosing digital networks, if only the functional inputs and outputs are used, the testing sequence can be very long and the resolution of fault location can be low. Furthermore, in the case of networks where fault masking redundancy is used, it is not possible to detect certain faults using only the input/output tests. The problems of where to put test points, how to compromise between the testing length and the number of test points, and how addition of some control inputs might improve diagnosis have not been well explored.

This section deals with these problems in two different ways. The first part approaches the diagnosis of sequential networks from a machine-theoretic viewpoint. The notions of test point allocation as well as network controllability can be captured by the concept of designing a diagnosable machine in the sense of checking experiment. The second part approaches the problems from a structure viewpoint.

The sequential machines considered here are assumed to be strongly connected and reduced, and it is assumed that malfunctions which occur in the circuit do not increase the number of states in the machine. The results are based on Mealy type sequential machines.

Machine Classification

We begin by introducing the following basic definition.

Definition 3. 1

If M is a sequential machine with states Q and $x \in I^{\dagger}$, then

x is a homing sequence (h. s.) for M if

$$\beta_q(x) = \beta_r(x) \qquad \bar{\delta}(q,x) = \bar{\delta}(r,x), \quad \forall q, r \in Q.$$

M is homable if it has a h. s.. M is k-homable if it is homable and

$$\min\{\ell g(x) \mid x \text{ is a h. s.}\} = k$$

for some positive integer k.

Definition 3. 2

If M is a sequential machine with states Q and $x \in I^{\dagger}$ then

x is a distinguishing sequence (d. s.) for M if

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

M is diagnosable if it has a d. s. M is k-diagnosable if it is diagnosable and

$$\min\{\ell g(x) \mid x \text{ is a d. s.}\} = k$$

for some positive integer k.

It is well known that every reduced machine is homable and every diagnosable machine is reduced. The least upper-bound for the minimum length h. s. is $\binom{n}{2} = \frac{n(n-1)}{2}$ for an n-state reduced machine. An upper-bound for the minimum length d. s., ξ, is

$$\xi = (n-1)n^n$$

for an n-state diagnosable machine [7]. A lower-bound for the length of a h. s. is 1 and that of a d. s. is $\lceil \log_p n \rceil$, where $\lceil \log_p n \rceil$ is the least integer greater than or equal to $\log_p n$, n is the number of states and p is the number of outputs. The latter is stated as the next theorem.

Theorem 3.1

If M is a reduced sequential machine with n states and p outputs and $x \in I^{\dagger}$ is a distinguishing sequence, then

$$\ell g(x) \geq \lceil \log_p n \rceil$$

Proof

Consider any $x \in I^{\dagger}$ with $\ell g(x) = m < \lceil \log_p n \rceil$. It follows that $p^m < n$. Since $\ell g(\beta_q(x)) = \ell g(x) = m$ for any $q \in Q$, the number of different output sequences is at most $p^m < n$. Thus, there exists $q, r \in Q$ such that $\beta_q(x) = \beta_r(x)$ but $q \neq r$ and x cannot be a d. s. of M.

Definition 3.3

If M is a sequential machine with states Q and $q, r \in Q$ $(q \neq r)$ then q and r merge under x $(x \in I^{\dagger})$ if

i) $\overline{\delta}(q, x) = \overline{\delta}(r, x)$.

If in addition to i)

ii) $\beta_q(x) = \beta_r(x)$

then q and r converge under x. A pair of states converge if they converge under some $x \in I^{\dagger}$ and a machine M is convergence free (c.f.) if no pair of states converge.

Example 3.1

Figure 3.1 shows a machine M_1 and its transition graph

	a	b
1	1/0	4/1
2	3/1	5/1
3	4/1	1/0
4	2/0	4/0
5	1/1	5/1

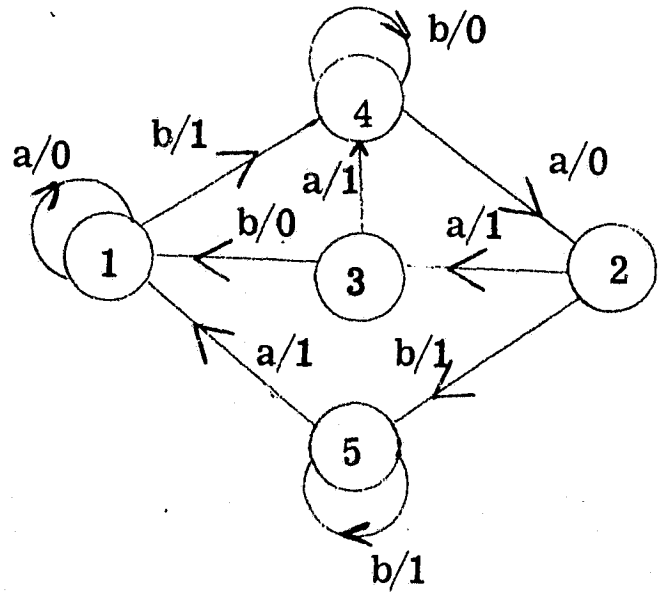


Figure 3.1 Machine M_1

aaa is a d. s. and bb is a h. s. but not a d. s. Since no sequence shorter than 3 is a d. s., M_1 is 3-diagnosable. Similarly M_1 is 2-homable. States 1 and 4 merge under b and states 2 and 5 converge under b.

We first introduce the following theorem due to Hennie [13].

Theorem 3.2

An input sequence x of a machine M is a distinguishing sequence iff it is a homing sequence and no pair of (distinct) states converge under x.

Proof

Let x be a d. s. for machine M.

Then

$$\beta_q(x) = \beta_r(x) \Rightarrow q = r, \quad \forall q, r \in Q$$

But

$$q = r \Longrightarrow \overline{\delta}(q, x) = \overline{\delta}(r, x)$$

$\therefore$ x is a h. s. of M

Moreover, no pair of states converge under x since, by definition of a d. s., $q \neq r \Longrightarrow \beta_q(x) \neq \beta_r(x)$. Conversely suppose x is a h. s. of M such that no pair of states $\{p, q\}$ converges under x, and suppose $\beta_q(x) = \beta_r(x)$. Then $q = r$, for if $q \neq r$ then, as q and r cannot converge under x,

$$\overline{\delta}(q, x) \neq \overline{\delta}(r, x)$$

contradicting the fact that x is a h. s. Thus x is a d. s.

Definition 3.4

A sequential machine M is definitely homable (d. h.) if there is an integer ℓ such that every input sequence of length ℓ is a homing sequence. The least such integer ℓ is called the order of definite homability.

Theorem 3.3

If M is an n-state machine, then M is definitely homable iff every input sequence of length greater than or equal to $\frac{n(n-1)}{2} = \binom{n}{2}$ is a homing sequence of M.

Proof

Suppose to the contrary, i.e. M is d. h. and yet $\exists\, x \in I^{\dagger}$ and $q, r \in Q$, $\overline{\delta}(q, x) \neq \overline{\delta}(r, x)$ such that $\ell g(x) \geq \binom{n}{2}$ and $\beta_q(x) = \beta_r(x)$. Then, for any initial segment y of x, $\overline{\delta}(q, y) \neq \overline{\delta}(r, y)$. We first

claim that for any two initial segments y and z of x,

$$\{\overline{\delta}(q, y),\ \overline{\delta}(r, y)\} \neq \{\overline{\delta}(q, z),\ \overline{\delta}(r, z)\}.$$

Suppose that this is not so, let $\ell g(y) > \ell g(z)$, i.e. z is an initial segment of y and let w be the tail segment of y with z deleted. Then, arbitrary iterations of w, $w^{\dagger}$, cannot be a h. s. of M, since $\beta_q(w^{\dagger}) = \beta_r(w^{\dagger})$ yet $\overline{\delta}(q, w^{\dagger}) \neq \overline{\delta}(r, w^{\dagger})$. This contradicts the assumption that M is d. h. This is illustrated in Figure 3.2. Thus we must conclude that there cannot be any repeated successor state pair of $\{q, r\}$ under x. On the other hand $\ell g(x) \geq \binom{n}{2}$ implies at least $\binom{n}{2} + 1$ distinct pairs are determined by all possible initial segments - an impossibility. Conversely, if $\forall x \in I^{\dagger}$ such that $\ell g(x) \geq \binom{n}{2}$, x is a h. s. of M, then every sequence of length $\ell = \binom{n}{2}$ is a h. s. and M is d. h. This proves the theorem.

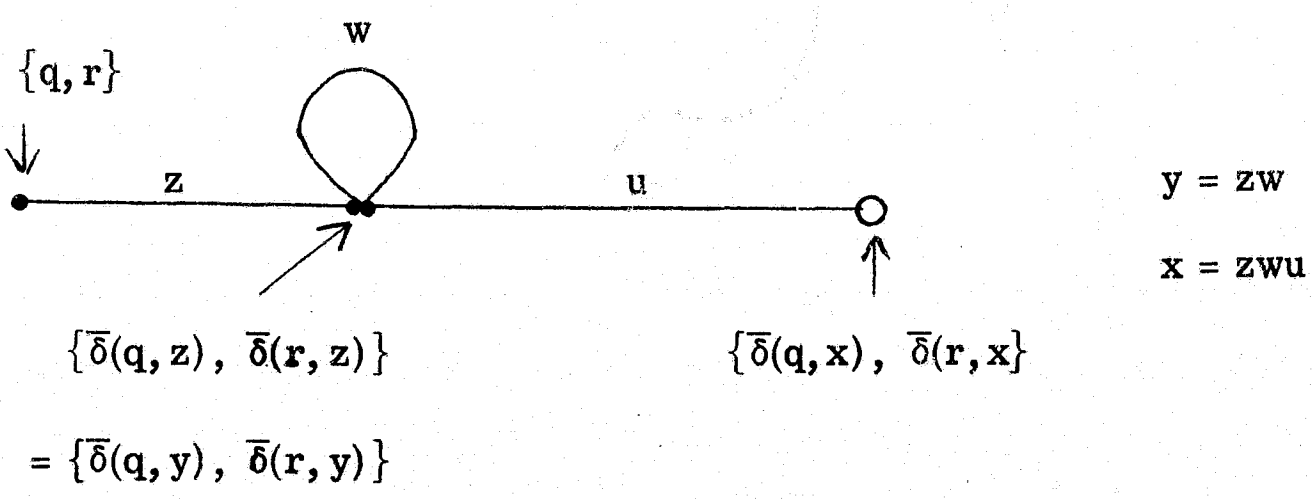


Figure 3.2 Successor State Pairs of $\{q, r\}$ Under x with Repeated Successor State Pair

Corollary 3.3.1

If M is a definitely homable machine of order ℓ, then $\ell \leq \binom{n}{2}$.

Proof

This follows immediately from Theorem 3.3.

Since the general upper-bound on the length of d. s. of a diagnosable machine is very large, we would like to identify classes of machines which have lower bounds. This motivates the study of another machine notion.

Definition 3.5

A sequential machine M is definitely diagnosable if there is an integer ℓ such that every input sequence of length ℓ is a distinguishing sequence. The least such integer ℓ is called the order of definite diagnosability.

The next theorem shows that d. h. is a generalization of d. d.

Theorem 3.4

Every definitely diagnosable machine is definitely homable.

Proof

Since every d. s. is also a h. s., it follows immediately from the definition of d. d. that M is d. h.

The converse of Theorem 3.4 is not true. One simple example is a d. h. machine in which some state pair converges. The next

theorem shows that no pair of states can converge if M is definitely diagnosable.

Theorem 3.5

If M is definitely diagnosable, then M is convergence free.

Proof

Suppose the contrary i.e. M is d.d. but for some $q, r \in Q$ $(q \neq r)$ $\exists x$ such that $\beta_q(x) = \beta_r(x)$ and $\overline{\delta}(q,x) = \overline{\delta}(r,x)$. Let the order of definite diagnosability be ℓ. Then the length of x cannot be equal to or greater than ℓ or otherwise x is a d.s., which implies $q = r$.

Suppose then that $\ell g(x) < \ell$ and let $y = xz$ where $\ell g(y) = \ell$

Now

$$\beta_q(y) = \beta_q(x)\ \beta_{\overline{\delta}(q,x)}(z),$$

$$\beta_r(y) = \beta_r(x)\ \beta_{\overline{\delta}(r,x)}(z)$$

merge under x,

$$\beta_q(y) = \beta_r(y)$$

where $q \neq r$. But $\ell g(y) = \ell \Longrightarrow$ y is d.s., a contradiction.

It is well known that every distinguishing sequence is also a homing sequence. The converse is not generally true. However, if a machine is definitely diagnosable then as a corollary of Theorem 3.5 we have:

Corollary 3.5.1

If M is definitely diagnosable then every homing sequence is a distinguishing sequence.

Proof

If M is d.d. then, by Theorem 3.5, no pair of states converge and, in particular, no pair of states converge under a homing sequence. Thus, by Theorem 3.2 every h.s. of M is a d.s. of M.

We now observe the following alternative characterization of definite diagnosability.

Theorem 3.6

If M is an n-state machine, then M is definitely diagnosable iff every input sequence of length greater than or equal to $\binom{n}{2}$ is a distinguishing sequence of M.

Proof

Suppose to the contrary, i.e. M is d.d. and yet $\exists\, x \in I^{\dagger}$ and $q, r \in Q$, $q \neq r$ such that $\ell g(x) \geq \binom{n}{2}$ and $\beta_q(x) = \beta_r(x)$. From Theorem 3.4, M is d.h. and x must be a homing sequence. But by Corollary 3.5.1, x must be a d.s., contradicting the original choice of x. Thus the assertion must be true. Conversely, if $x \in I^{\dagger}$ such that $\ell g(x) \geq \binom{n}{2}$ x is a d.s. of M, then every sequence of length $\ell = \binom{n}{2}$ is a d.s. and M is d.d. This proves the theorem.

With the above theorem, we obtain immediately the following corollary.

Corollary 3.6.1

If M is definitely diagnosable of order ℓ, then $\ell \leq \binom{n}{2}$.

Recall that the converse of Theorem 3.4 is not true, i.e. a machine can be definitely homable yet not be definitely diagnosable. However, if a convergence free condition is added, then we obtain yet another characterization of definite diagnosability, that is

Theorem 3.7

A machine is definitely diagnosable iff it is both convergence free and definitely homable.

Proof

If M is d.d. then by Theorem 3.5 it is c.f. and by Theorem 3.4 it is d.h. Conversely, if M is d.h. and c.f., then for some k, $x \in I^k \Longrightarrow$ x is a h.s. But M is c.f. and by Theorem 3.2, x is a h.s. $\Longrightarrow$ x is a d.s. Consequently $x \in I^k \Longrightarrow$ x is a d.s., i.e. M is d.d.

Let us call a distinguishing sequence (or homing sequence) minimal if no initial segment of it is also a distinguishing sequence (homing sequence). The notion of definite homability is useful in the sense that if M is definitely homable, the length of a minimal d.s. has the same upper bound as that of a definitely diagnosable machine. This is characterized by the following theorem.

Theorem 3.8

If M is an n-state definitely homable machine and x is a minimal d. s. of M then $\ell g(x) \leq \binom{n}{2}$.

Proof

Suppose there is a minimal d. s. x such that $\ell g(x) > \binom{n}{2}$. Now x is also a h. s. and moreover must be minimal as a h. s. because any initial segment y of x ($y \neq x$) that is a h. s. would be a d. s., contradicting the minimality of x. However by Theorem 3.3, a minimal h. s. must have length $\leq \binom{n}{2}$ - contradicting the original assumption.

Theorem 3.7 characterizes the d.d. property of a machine by means of its being both convergence free and definitely homable. From a practical point of view, we have found it more convenient to characterize a machine by its submachines. A theorem which accomplishes precisely this goal is introduced with the help of the following definition and lemma.

Definition 3.6

Let $M = (I, O, Q, \delta, \omega)$ be a sequential machine. A single-input submachine M_a of M is defined as $M_a = (\{a\}, O, Q, \delta | Q \times \{a\}, \omega | Q \times \{a\})$ where $a \in I$. If $|I| = 1$, M is called a single-input machine, a sequence generator or an autonomous machine. A sequence generator is p-nary if $|\mathcal{R}(\omega)| = p$.

Lemma 3.1

If every single-input submachine of a sequential machine M is reduced, then M is convergence free.

Proof

Suppose M is not c.f. Then $\exists$ q, r ϵ Q and a ϵ I $\ni$ q $\neq$ r but $\omega(q, a) = \omega(r, a)$ and $\delta(q, a) = \delta(r, a)$ and, relative to the extended output function $\bar{\omega}$,

$$\bar{\omega}(q, a^n) = \bar{\omega}(r, a^n), \quad \forall n \geq 1$$

This means q and r are equivalent in M_a and M_a is not reduced. This completes the proof.

The converse of the above lemma is not true; in particular, convergence free autonomous machines need not be reduced.

Theorem 3.9

A sequential machine is definitely diagnosable if and only if

(1) M is definitely homable

and

(2) every single input submachine is reduced.

Proof

First assume M is definitely diagnosable. Then Theorem 3.4 shows that (1) holds. If (2) does not hold, then there is some a ϵ I such that any repeated symbol sequence formed by a is not a d.s. Thus M can not be definitely diagnosable.

Conversely if both (1) and (2) hold, then from Lemma 3.1, (2) guarantees that M is convergence free. (1) and the convergence free condition say that M is definitely diagnosable (Theorem 3.7). Thus, the proof is complete.

This theorem can be viewed as a stronger and more practical version of Theorem 3.7. It is stronger because condition (2) is stronger than the convergence free property as pointed out by Lemma 3.1 and the remarks following it. Interestingly (2) and d.h. is equivalent to c.f. and d.h.

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 in Figure 3.3 to represent the hierarchy of machine classes.

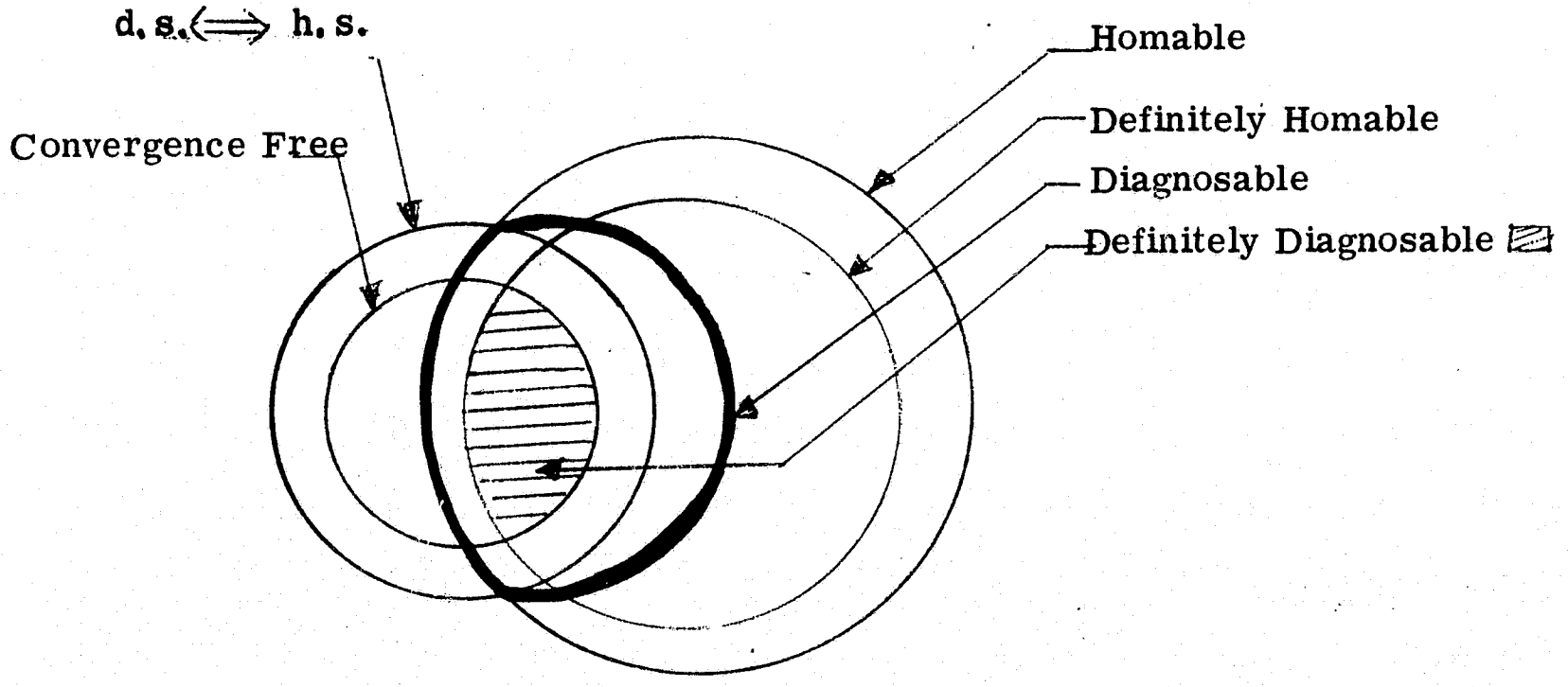


Figure 3.3 Machine Classification Diagram

Note that the class of machines that satisfy the hypothesis of Theorem 3.8 is the intersection of the class of diagnosable machines and that of definitely homable machines.

Augmentation of Test Outputs

We first discuss a method of designing a diagnosable machine by augmenting output logic. 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. However, if the machine has only one input symbol, then "being reduced" is also sufficient for the existence of a d. s., that is

Theorem 3.10

An n-state, single-input machine is reduced iff it has a distinguishing sequence of length less than or equal to n-1.

Proof

Moore [18] has shown that in a reduced n-state machine, any pair of states can be distinguished by a sequence of length n-1. Since we only have a single input alphabet here, any shorter sequence is an initial segment of the longer sequence. Let x_i be the minimum length sequence which distinguishes state pair $\{q, r\} \subseteq Q$. Let $k = \max \{\ell_g(x_i)\}$ over all possible state pairs and let x be the (unique) input of length k. Then x is a distinguishing sequence for the single-input machine and $\ell g(x) \leq n-1$. Conversely, any machine with a d.s. must be reduced.

A distinguishing sequence which has only one input symbol is called a repeated symbol distinguishing sequence (r.d.s.). Theorem 3.10 provides a convenient way of checking whether a sequential machine has any r.d.s. and constructing one if there is none. The following corollary will characterize this property.

Corollary 3.10.1

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

Proof

If the machine has a repeated symbol distinguishing sequence $x = a \cdot a \ldots a$, $a \in I$, then x is also a d.s. for the submachine M_a. Thus M_a is reduced. Conversely, if M_a is a reduced single-input submachine of M, then from the previous theorem it has a d.s. x. Since M_a and M have the same state set, x is also a d.s. for M.

Thus to see whether a machine M has a r.d.s., it is only necessary to examine whether any of its single-input submachines M_a is reduced. Since a single-input machine is reduced iff it is definitely diagnosable, this is equivalent to requiring that some M_a be d.d. 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 [16]. Before stating their procedure, we need to introduce the concept of a testing graph.

Definition 3.7

A testing graph of a sequential machine $M = (I, O, Q, \delta, \omega)$ is a directed graph which is constructed as follows: Each node of the graph corresponds to a pair of distinct states which are either indistinguishable under some input or is a state pair successor of the former. A directed branch is drawn from a node $\{q_i, q_j\}$ to node $\{q_m, q_n\}$ iff $\exists\ a \in I$

$$\omega(q_i, a) = \omega(q_j, a)$$

$$\delta(q_i, a) = q_m$$

$$\delta(q_j, a) = q_n$$

where q_i, q_j, q_m, $q_n \in Q$ and, m and n need not be distinct from i and j, but $i \neq j$ and $m \neq n$.

Example 3.2

The testing graph of a machine M_2 is shown in Figure 3.4.

	a	b
1	1/0	4/1
2	3/0	5/1
3	4/0	1/0
4	2/0	4/0
5	1/1	5/1

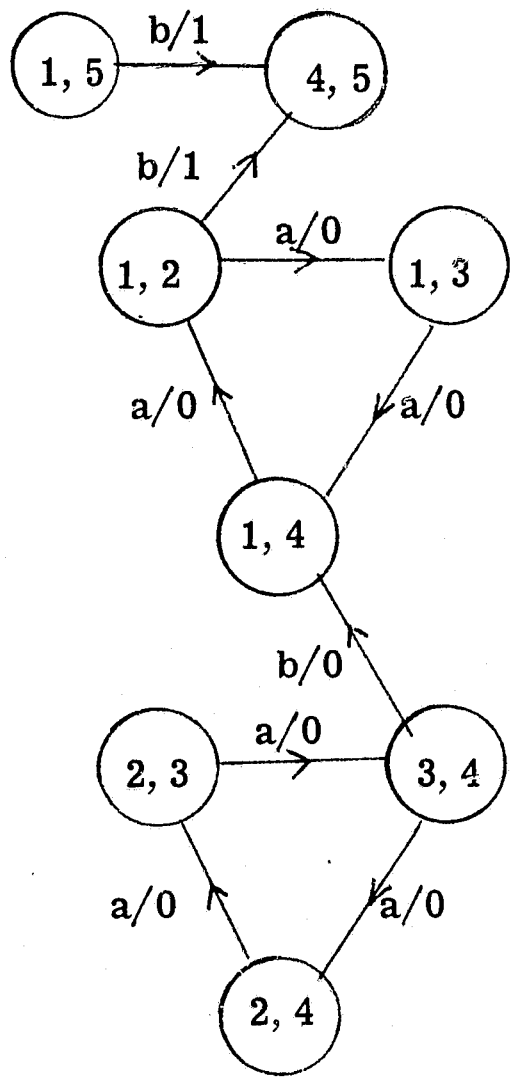


Figure 3.4 Machine M_2 and Its Testing Graph

From Theorem 3.3 and the definition of testing graph, it follows immediately that the definite homable property corresponds to the cycle-free condition in the testing graph. This is stated as the next theorem.

Theorem 3.11

A sequential machine is definitely homable iff its testing graph is cycle-free.

The procedure for constructing a definitely diagnosable machine from an arbitrary machine is now stated in the following.

(1) Assign different new output values to each converging state pair.

(2) Open all the cycles of the testing graph by eliminating the smallest number of branches. A branch in a testing graph is eliminated if different new output values were assigned for that branch.

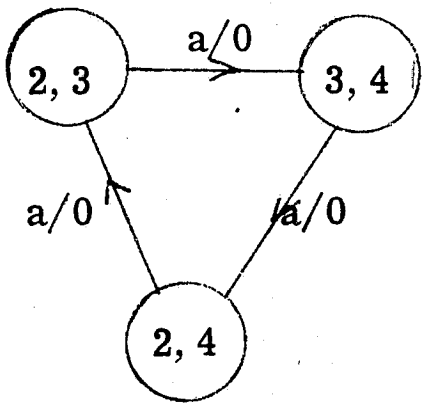
(1) corresponds to establishing the convergence free condition and (2) corresponds to obtaining the definitely homable property in the augmented machine.

Example 3.3

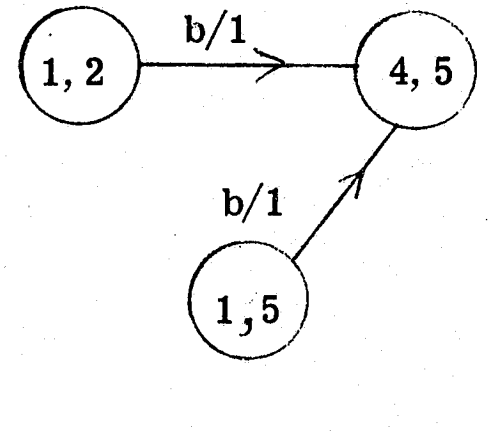
Figure 3.5 shows machine M_3 and the testing graphs of its two single-input submachines M_a and M_b. M_3 is not diagnosable. Both M_a and M_b have a converging state pair, but only the testing graph of M_b is cycle free. For purpose of illustration, let us construct a diagnosable M_a denoting it as M_a'. Applying rule (1), we assign different new output values to the converging state pair 1, 5 under input a. Applying rule (2), we assign different new output values to state pairs 2, 3, opening the cycle in its testing graph. Since the new output of state 4 under input a has to be either different from 2 or 3, we arbitrarily choose to break the state pair 2, 4. The resultant machine M_2' is shown in Figure 3.6.

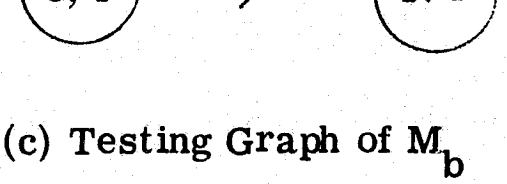
	a	b
1	1/1	4/1
2	3/0	5/1
3	4/0	1/0
4	2/0	4/0
5	1/1	5/1

(a) Machine M_3



(b) Testing Graph of M_a



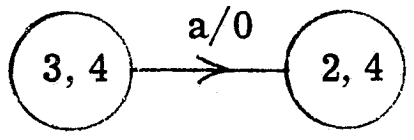


(c) Testing Graph of M_b

Figure 3.5 Machine M_3 and the Testing Graphs of Its 2 Single-Input Submachines

	a	b
1	1/10	4/1x
2	3/00	5/1x
3	4/01	1/0x
4	2/01	4/0x
5	1/11	5/1x

(a) Machine M_3'



(b) Testing Graph of M_a'

Figure 3.6

M_3' has a r.d.s. x = aa. The new output values in column b are left unspecified so that when physical realization takes place, further simplification in physical realization can be obtained.

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 the minimum number of additional output terminals required, each single-input submachine can be analyzed and the submachine which requires the smallest number of additional outputs is then selected. This exhaustive procedure may be very time consuming when the number of inputs or number of states is very large. Unfortunately, there is no simple

rule which can enable systematic selection by inspection. 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.

An upperbound ξ for the minimum length checking experiment using a repeated symbol distinguishing sequence has been derived in [16] and is shown below:

$$\xi = nm + n(m-1)\,\ell + \ell + (m-1)(n-1)^2 \qquad (3.1)$$

where

n = Number of states

m = Number of input symbols

ℓ = Length of minimal r. d. s. used ($\ell \leq (n-1)$)

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. d. s. However, upperbounds for the length of this kind of d. s. may be quite large.

Augmentation of Control Inputs

In this section, we will study the problem of designing a diagnosable machine with a r. d. s. using additional input logic. From Corollary 3. 10. 1, we know that if we append a reduced single-input submachine to the original machine, then the combined machine will have a r. d. s. consisting of the repetition of the newly created input symbol. To illustrate this, let us consider the machine M_4 whose state table is shown below:

M_4 =

	a	b
1	1/0	2/0
2	1/0	3/0
3	1/0	4/0
4	1/1	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_4' =

	a'	b'	c
1	1/0	2/0	2/0
2	1/0	3/0	4/1
3	1/0	4/0	3/0
4	1/1	1/0	1/1

This modified machine has a distinguishing sequence of cc.

An upperbound for the minimum length fault detecting experiment is obtained by replacing m with m+1 in Equation (3.1). This is shown below

$$\xi = n(m+1) + (nm+1)\,\ell + m(n-1)^2 \qquad (3.2)$$

where

n = Number of states

m = Number of original input alphabets

ℓ = Length of distinguishing sequence used

Recall that the minimum length of a d.s. in an n-state, p-output machine is $\lceil \log_p n \rceil$. In the following we will show that in case both p 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

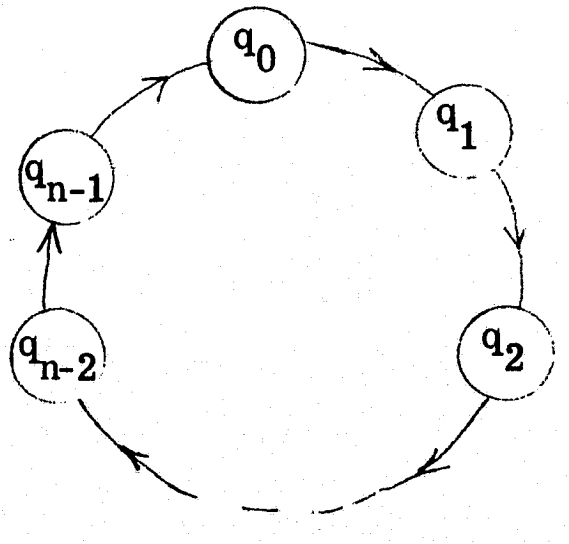
initially in, it is only necessary to shift the register s times. Thus this circuit corresponds to a 2^s state, single-input and binary output machine with a distinguishing sequence of length s. Appending such a single-input machine to a given machine is equivalent to a modification that causes the machine to act as a shift register under certain inputs. The above observation can be generalized and formally stated as follows:

Theorem 3.12

For every integer n, there is an n state, binary output, single-input sequential machine which is optimally diagnosable.

Proof

Consider an n-state, single-input and binary output machine which has a state diagram as shown below:



A binary output sequence of period $n = 2^s$ will contain all possible binary s-tuples [8]. Thus the input sequence of length s is a

d. s. Such a machine can be realized by an s-stage shift sregister of period 2^s which can be constructed as follows: Construct a linear maximum length, s-stage binary shift register which generates a sequence of length 2^s-1. The period 2^s is achieved by module 2 adding the logical function which detects all 0's condition as above. But all periods from 1 to 2^s can be obtained from an s-stage register [8]. Thus n needs not be a power of 2 and the theorem is proved.

Example 3.4

Let us construct an 8-state binary output machine with d. s. of length 3 as outlined in the proof of the previous theorem. From Peterson [20], choose a primitive polynomial $h(x) = x^3+x^2+1$ over GF(2). A maximum length shift register generator obtained according to h(x) has a period of $2^3-1 = 7$. With modification, the shift register can be made to generate a sequence of length $2^3=8$. The state diagram of this 8-state machine and its shift register generator realization circuit are shown in Figure 3. 7.

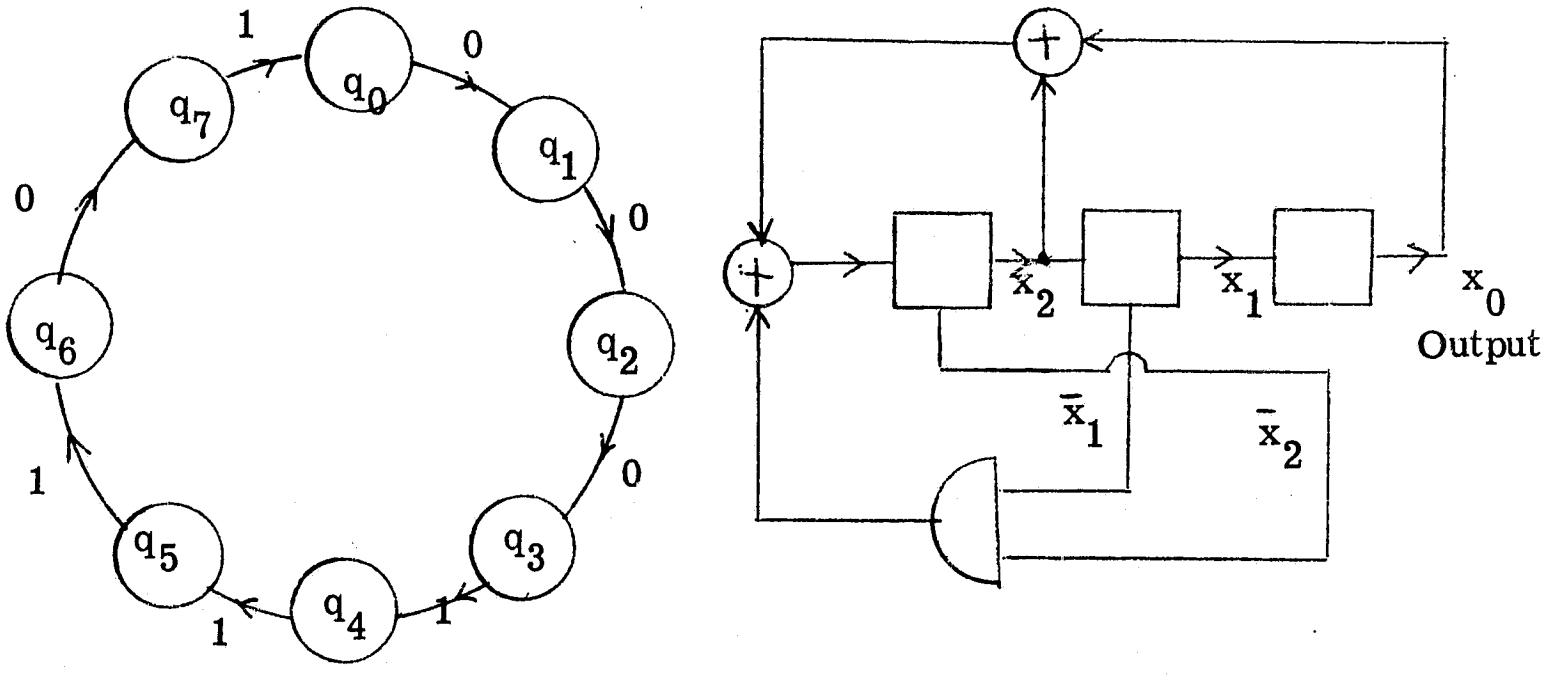


Figure 3.7 An 8-state Machine with d. s. of Length 3.

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 fact, the proof shows that there is a strongly connected single-input machine which satisfies Theorem 3.12.

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 3.13

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

Proof

Consider an n-state, single-input machine with a single loop state structure as shown in the proof of Theorem 3.12. Such a machine structure can be realized by a s-stage shift register. The t output terminals can be spaced evenly among the s-stages so that no more than $\lceil \frac{s}{t} \rceil$ shifts is necessary in order to determine externally the contents of each stage. The machine defined by this circuit then satisfies the condition of the theorem.

An upperbound for the minimum length fault detecting experiment using the above construction of providing a "diagnosable input" is obtained by replacing ℓ in Equation (3.2) with $\lceil \log_p n \rceil$:

$$\xi = n(m+1) + (nm+1)\lceil \log_p n \rceil + m(n-1)^2 \qquad (3.3)$$

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. An upper bound for the minimum length fault detecting experiment in this case of providing both diagnosable input and reset input can be obtained from Equation (3.3) with the following modification. First, since an additional input is created for reset, m should be replaced by m+1. Second, since a reset input has been created, the average length of a transfer sequence should be no longer than

$$\frac{\sum_{i=1}^{n-1}(n-i)}{n-1} = \frac{n(n-1) - \sum_{i=1}^{n-1} i}{n-1}$$

$$= n - \frac{\frac{[1+(n-1)](n-1)}{2}}{n-1} = \frac{n}{2}$$

But for each application of the transfer sequence, we may need to apply the reset input, therefore the average length of the transfer sequence should be no longer than

$$\frac{n}{2} + 1$$

The new upper bound is this

$$\xi = n(m+2) + [n(m+1)+1]\lceil \log_p n \rceil + (m+1)(n-1)\left[\frac{n}{2} + 1\right] \qquad (3.4)$$

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 [16].

Let

ξ_{DD} = Bound of d.d. machines.

ξ_{OD} = Bound of diagnosable machines which have a r.d.s. by augmenting output logic.

ξ_{ID} = Bound of diagnosable machines which have a r.d.s. by appending a single-input machine.

ξ_{IDR} = Same as ξ_{ID} but with additional reset input.

n = Number of states.

m = Number of input symbols in the original machine.

p = Number of output symbols.

Then we have

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

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

$$\xi_{ID} = n(m+1) + (nm+1)\lceil \log_p n \rceil + m(n-1)^2$$

$$\xi_{IDR} = n(m+2) + [n(m+1)+1]\lceil \log_p n \rceil + (m+1)(n-1)\left[\frac{n}{2} + 1\right]$$

All the above equations have been verified except ξ_{DD}. This can be easily derived from Equation (3.1) by only changing the second term in

that equation. First we recalled that an upperbound for the length of any proper d. s. in a d. d. machine is $\binom{n}{2}$. This means that we can let $\ell = \binom{n}{2}$. Next, since a distinguishing sequence is required to identify each terminal state in each input transition, we need nm d. s. instead of n(m-1) as in the case of using r. d. s. Thus the equation for ξ_{DD} is obtained.

The following table shows several numerical evaluations of the four upperbounds.

		p = 2				p = 4			
m	n	ξ_{DD}	ξ_{OD}	ξ_{ID}	ξ_{IDR}	ξ_{DD}	ξ_{OD}	ξ_{ID}	ξ_{IDR}
2	2	10	(8)	13	21	10	(8)	13	21
	4	71	(32)	48	69	71	(32)	39	56
	8	541	(128)	173	212	541	(128)	156	185
	16	4,434	(512)	730	665	4,434	(512)	564	567
4	2	22	(18)	23	33	22	(18)	23	33
	4	143	(82)	90	111	143	82	(72)	90
	8	1,069	354	(334)	346	1,069	354	(302)	305
	16	8,345	1,474	1,240	(1,095)	8,345	1,474	1,110	(883)
8	2	54	(38)	39	57	54	(38)	39	57
	4	351	182	(166)	195	351	182	(141)	158
	8	2,637	806	643	(614)	2,637	806	594	(541)
	16	20,797	3,398	2,432	(1,955)	20,797	3,398	2,202	(1,665)

The circled values represent the smallest number for each p, m, and n. From the above numerical evaluation, it appears that ξ_{IDR} is the smallest among the four bounds compared for large n, m, and p. The numerical ordering of these bounds will be derived in the following.

First, let us consider the difference of ξ_{DD} and ξ_{OD}

$$\xi_{DD} - \xi_{OD} = (nm+1)\frac{n(n-1)}{2} - (nm+1-n)(n-1)$$

$$= (nm+1)(n-1)\left[\frac{n}{2} - 1\right] + n(n-1)$$

For $m > 0$ and $n \geq 2$, it is clear that $\xi_{DD} - \xi_{OD} > 0$ since each term in the above expression is non-negative. Therefore, we conclude $\xi_{DD} > \xi_{OD}$ for $m > 0$ and $n \geq 2$.

Next, let us compare ξ_{OD} and ξ_{ID}

$$\xi_{OD} - \xi_{ID} = -n - (n-1)^2 + [n(m-1)+1](n-1) - (nm+1)\lceil \log_p n \rceil$$

$$= (nm+1)\left[(n-1) - \lceil \log_p n \rceil\right] - \left[n^2 + (n-1)^2\right]$$

Let

$$\phi = (nm+1)\left[(n-1) - 1 - \log_p n\right] - \left[n^2 + (n-1)^2\right]$$

then

$$\xi_{OD} - \xi_{ID} > \phi \qquad \forall \text{ all } n, m \text{ and } p$$

since

$$1 + \log_p n > \lceil \log_p n \rceil.$$

$$\frac{\partial}{\partial m}\,\phi = n[(n-1) - 1 - \log_p n] \geq 0 \qquad \forall\, n \geq 4$$

$$\frac{\partial}{\partial n}\,\phi = m[(n-1) - 1 - \log_p n] + (nm+1)\,\left[1 - \frac{1}{n}\log_p e\right] - (4n-2) > 0$$

for $m \geq 6$ and for all $n \geq 4$ and $p \geq 2$.

But when $n = 6$, $m = 8$ and $p = 2$

$$\phi = (6 \times 8 + 1)\,[5 - 1 - \log_2 6] - [36 + 25]$$

$$= 49 \times 1.41 - 61$$

$$\cong 8 > 0$$

Since ϕ is monotonically increasing with respect to m, n and p for $m \geq 6$, $n \geq 4$, and $p \geq 2$ and since $\phi > 0$ when $n = 6$, $m = 8$ and $p = 6$, we conclude that $\xi_{OD} - \xi_{ID} > \phi > 0$ and $\xi_{OD} > \xi_{ID}$ for $n \geq 6$, $m \geq 8$ and $p \geq 2$. Similarly, relationships between ξ_{ID} and ξ_{IDR} can be derived

$$\xi_{ID} - \xi_{IDR} = -n + (nm+1)\,\lceil \log_p n \rceil + m(n-1)^2 - [n(m+1)+1]\,\lceil \log_p n \rceil - (m+1)(n-1)\left(\frac{n}{2} + 1\right)$$

$$= \frac{mn^2}{2} - \frac{5mn}{2} + 2m - \frac{3n}{2} + 1 - \frac{n^2}{2} - n\lceil \log_p n \rceil.$$

Let

$$\psi = \frac{mn^2}{2} - \frac{5mn}{2} + 2m - \frac{3n}{2} + 1 - n(1 + \log_p n) - \frac{n^2}{2}$$

then

$$\xi_{ID} - \xi_{IDR} \geq \phi \qquad \forall\ m, n \text{ and } p.$$

$$\frac{\partial \psi}{\partial m} = \frac{n^2}{2} - \frac{5n}{2} + 2 = \frac{n}{2}(n-5) + 2 > 0 \qquad \forall\ n \geq 5$$

$$\frac{\partial \psi}{\partial n} = mn - \frac{5m}{2} - \frac{3}{2} - n - (1 + \log_p n) - \log_p e$$

$$= (m-1)\ n - \frac{5m}{2} - [1 + \log_p n + \log_p e] > 0$$

when $m \geq 9$, $n \geq 9$ and $p \geq 2$.

Now when $n = m = 9$ and $p = 2$

$$\psi = \frac{9 \times 81}{2} - \frac{5 \times 81}{2} + 18 - \frac{27}{2} + 1 - 9\ (1 + 3.18)$$

$$- \frac{81}{2}$$

$$= 181 - 91.62 = 89.38 > 0$$

Since ψ is also monotonically increasing with respect to n, m and p for $m \geq 9$, $n \geq 9$ and $p \geq 2$, and $\psi > 0$ when $m = n = 9$ and $p = 2$, it follows that $\xi_{ID} > \xi_{IDR}$ for $m \geq 9$, $n \geq 9$ and $p \geq 2$.

Therefore in general when $m \geq 9$, $n \geq 9$ and $p \geq 2$ the ordering of these bounds is shown below

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

Diagnosable Machine Realization

The concept of designing a diagnosable machine by augmenting output or input terminals have been discussed in the previous sections. In this section we will first look at this problem from a more generalized viewpoint of machine realization. In particular, we will verify that the test output augmentation design will indeed result in a machine which realizes the original machine in the formal sense of realization as defined by Hartmanis and Stearns [11]. The problem of breaking up state set convergence is then considered. This is a necessary step in designing any diagnosable machine when no additional input is used. The problem of eliminating cycles in the testing graph of a single-input machine is next discussed. Finally, the general problem of designing a diagnosable machine with a bounded length d. s. is investigated.

In order to design a diagnosable machine M' from a given machine M, it is sufficient to augment M by adding to it some output terminals and assigning different output symbols to selected transitions. We will verify here that M' so constructed is indeed a realization of M. For convenience of notational reference, we formally state the definition of machine realization.

Definition 3.8

A machine $M' = (I', O', Q', \delta', \omega')$ is said to be a realization of $M = (I, O, Q, \delta, \omega)$ if there exists a triplet of functions (α, ι, ξ) where

$$\alpha : Q \longrightarrow P(Q')^{*} - \{\phi\}$$

$$\iota : I \longrightarrow I'$$

$$\zeta : O' \longrightarrow O$$

such that

(i) $\delta'(\alpha(q), \iota(a)) \subseteq \alpha(\delta(q, a))$ $\forall q \in Q$ and $a \in I$,

(ii) $\zeta(\omega'(q', \iota(a))) = \omega(q, a)$ $\forall q' \in \alpha(q)$ and $a \in I$.

With this definition formally stated, let us verify that a diagnosable machine obtained from a given machine using procedure introduced earlier is indeed a machine realization.

Example 3.5

Machine M'_1 is a diagnosable machine obtained from the given machine M_1 by augmenting one additional output variable which breaks up the (2, 5) state pair convergence under input b. M'_1 has a d. s. of bb.

M_1 =

	a	b
1	1/1	4/1
2	3/0	5/1
3	4/0	1/0
4	2/0	4/0
5	1/1	5/1

$I = \{a, b\}$

$O = \{0, 1\}$

$Q = \{1, 2, 3, 4, 5\}$

M'_1 =

	a	b
1	1/1x	4/1x
2	3/0x	5/10
3	4/0x	1/0x
4	2/0x	4/0x
5	1/1x	5/11

$I' = I = \{a, b\}$

$O' = \{0x, 10, 11\}$

$Q' = Q = \{1, 2, 3, 4, 5\}$

*P(Q') denotes the power set of set Q', i.e. the set of all subsets of Q'.

$\alpha = e_Q$ (identity function of states)

$\iota = e_I$ (identity function of inputs)

$\zeta(10) = 1$

$\zeta(11) = 1$

$\zeta(0x) = 0$

Thus (α, ι, ζ) indeed satisfies the requirement of machine realization and M_1' is a realization of M_1.

The state table of machine M_1' in Example 3.5 has many of its output entries partially specified. This is done to minimize the restrictions placed on the realizing machine so that the physical realization may be simplified. Note that the cardinality of O' is 3 instead of 4 since no new output need to be assigned to distinguish state pair with output 0. To clarify this point, let us rename the output symbols in O' as follows:

A = 10

B = 11

C = 0x

Then M_1' becomes

	a	b
1	1/(A, B)	4/(A, B)
2	3/C	5/A
3	4/C	1/C
4	2/C	4/C
5	1/(A, B)	5/B

$I' = \{a, b\}$

$O' = \{A, B, C\}$

$Q' = \{1, 2, 3, 4, 5\}$

$\alpha = e_Q$

$\iota = e_I$

$\zeta(A) = 1$

$\zeta(B) = 1$

$\zeta(C) = 0$

The above observations bring out several interesting facts. First, the cardinality of O' is a finer measurement of output logic complexity in the realizing machine than using the number of output terminals. Thus the optimization criterion of minimizing the cardinality of the output set in the realizing machine can be used instead of minimizing the output terminals using binary realization. Second, since many of the output entries in the realizing machine are often partially specified, the realizing machine is in fact a set of many completely specified machines. In general, the design problem can be stated as follows:

> Given some sequential machine M, design a new machine M' which realizes M and has a distinguishing sequence

Several variations of this problem are possible by restricting the type of realizations. For example, the result of augmenting test output is a state behavior realization with $\alpha = e_Q$ and $\iota = e_I$. To simplify the discussion that follows it is convenient to introduce the following notational conventions. If $M = (I, O, Q, \delta, \omega)$ and $M' = (I', O', Q', \delta', \omega')$, let $|I| = m$, $|O| = p$, $|Q| = n$, $|I'| = m'$,

$|O'| = p'$ and $|Q'| = n'$. The problem can then be classified into the following three subproblems:

1. M' realizes M with $n = n'$ and $m = m'$. This is equivalent to test output augmentation.
2. M' realizes M with $n = n'$ and $p = p'$. This is equivalent to control input augmentation.
3. M' realizes M with $m = m'$. This is equivalent to state and test output augmentation.

Other subproblems can also be obtained but they are mostly less interesting. For example the problem of augmenting both input and state is relatively uninteresting since we know that augmenting one single input is sufficient to obtain a diagnosable machine. However, although augmenting test output is also sufficient for designing a diagnosable machine, it is possible to reduce the number of outputs needed by increasing the number of states in the realizing machine. We shall first illustrate this point by the following example.

Example 3.6

M_2 is a reduced, strongly connected machine which has no distinguishing sequence. Suppose it is desired to design a diagnosable machine with an rds which realizes M_2 and has a smallest output set. It is easily seen that using state behavior realization input a requires an output set of 4 symbols and input b requires an output

set of size 5. However using state splitting technique M_2 can be realized by a machine M'_2 having output set of size 3.

$M_2 =$

	a	b
1	1/0	2/1
2	1/0	3/0
3	1/0	4/0
4	5/0	2/1
5	6/0	2/1
6	1/1	2/1

$I = \{a, b\}$

$O = \{0, 1\}$

$Q = \{1, 2, 3, 4, 5, 6\}$

$M'_2 =$

	a	b
A	A/β	B/δ
B	G/β	$C/\{\beta, r\}$
C	A/r	$D/\{\beta, r\}$
D	$E/\{\beta, r\}$	B/δ
E	$F/\{\beta, r\}$	B/δ
F	A/δ	B/δ
G	G/r	C/δ

$I' = I = \{a, b\}$

$O' = \{\beta, r, \delta\}$

$Q' = \{A, B, C, D, E, F, G\}$

O'	$\zeta(O')$
β	0
r	0
δ	1

Q	$\alpha(Q)$
1	A, G
2	B
3	C
4	D
5	E
6	F

M'_2 is an output partially specified machine which realizes M_2 and has an rds of aaa.

Problems (1) and (2) have been previously discussed for the case of obtaining a repeated symbol distinguishing sequence. We will devote most of the following discussion to problem (3).

Although the increase in number of states in the realizing machine will in general increase the length of testing and possibly

complicate the physical realization, it has one important advantage. It says that the number of test points that need to be brought out in a given network may be reduced by increasing the number of states. This is a very desirable point in view of the trend in the manufacturing of large scale integrated logic circuit. Generally, a very severe limitation is the number of input-output pins that can be brought out while the increase in cost due to additional components is small.

In the last example, we had observed how state splitting techniques can be used to reduce the size of output sets in the realizing machine. We shall develop a technique which allows such reductions whenever possible. First we shall treat the case of state set convergence and show that by increasing the number of states in the realizing machine, the required output set size or number of output functions can be reduced. Then we shall treat the acyclic testing graph realization of single-input machines.

State Set Convergence Problem

Let $M = (I, O, Q, \delta, \omega)$ be a sequential machine and $S \subseteq Q$, and for any $q, r \in S$, $\delta(q, a) \neq \delta(r, a)$ and $\omega(q, a0 = \omega(r, a) = 0_1$ for some $a \in I$ and $0_1 \in O$. We shall divide the treatment of state set convergence into three cases:

Case 1 $\delta(q, a) \in S$

Theorem 3.14

If $|S| = k$, i. e. M has k states converge to a single state which is also in the set S, then the convergence can be broken up by a realizing output set of no more than $\left\lfloor \frac{k}{2} \right\rfloor + 1$ symbols and with the number of additional states no more than $k - (\left\lfloor \frac{k}{2} \right\rfloor + 1)$, where $\left\lfloor \frac{k}{2} \right\rfloor$ means the largest integer not greater than $\frac{k}{2}$.

Proof

For simplicity let $Ma = (\{a\}, \{0_1\}, S, \delta|S \times \{a\}, \omega|S \times \{a\})$ be the submachine of M. Without loss of generality assume Ma has the following state table description

	a
1	$1/0_1$
2	$1/0_1$
3	$1/0_1$
⋮	⋮
k	$1/0_1$

$S = \{1, 2, 3, \ldots, k\}$

If k is even, let us construct a machine Ma' with $k + \frac{k}{2} - 1$ states which realizes Ma with the assignment (α, ι, ξ) and state table description as follows:

$M'_a =$

	a
1	$q_1/0_{1,1}$
2	$q_2/0_{1,1}$
3	$q_3/0_{1,1}$
⋮	⋮
$\frac{k}{2}-1$	$q_{\frac{k}{2}-1}/0_{1,1}$
$\frac{k}{2}$	$1/0_{1,1}$
⋮	⋮
k	$1/0_{1,k/2+1}$
q_1	$q_1/0_{1,2}$
q_2	$q_2/0_{1,3}$
⋮	⋮
$q_{\frac{k}{2}-1}$	$q_{\frac{k}{2}-1}/0_{1,\frac{k}{2}}$

$M'_a = (\{a\}, 0', S', \delta' | S' \times \{a\}, \omega' | S' \times \{a\})$

$0' = \{0_{1,i} | i=1, 2, \ldots, \frac{k}{2}+1\}$

$S' = \{1, 2, 3, \ldots, k, q_1, \ldots, q_{\frac{k}{2}-1}\}$

$\alpha(i) = i \qquad \forall\, i \in S - \{1\}$

$\alpha(1) = 1 \cup \{q_i | i = 1, 2, \ldots, \frac{k}{2} - 1\}$

$\iota(a) = a$

$\zeta(0_{1,i}) = 0_1, \qquad i=1, 2, \ldots, \frac{k}{2}+1$

If k is odd, let us construct a machine M'_a with $k + \frac{k-1}{2}$ states which realizes Ma with (α, ι, ζ) and state table as follows:

$M''_a =$

	a
1	$q_1/0_{1,1}$
2	$q_2/0_{1,1}$
3	$q_3/0_{1,1}$
⋮	⋮
$\frac{k-1}{2}$	$q_{\frac{k-1}{2}}/0_{1,1}$
$\frac{k-1}{2}+1$	$1/0_{1,1}$
⋮	⋮
k-1	$1/0_{1,\frac{k-1}{2}}$
k	$1/0_{1,\frac{k-1}{2}+1}$
q_1	$q_1/0_{1,2}$
q_2	$q_2/0_{1,3}$
⋮	⋮
$q_{\frac{k-1}{2}}$	$q_{\frac{k}{2}}/0_{1,\frac{k-1}{2}+1}$

$M''_a = (\{a\}, 0'', S'', \delta''|S'' \times \{a\}, \omega''|S'' \times \{a\})$

$0'' = \{0_{1,i} | i=1,2,\ldots,\frac{k-1}{2}+1\}$

$S'' = \{1,2,\ldots,k,q_1,\ldots,q_{\frac{k-1}{2}}\}$

$\alpha(i) = i \quad \forall\, i \in S - \{1\}$

$\alpha(1) = 1 \cup \{q_i | i=1,2,\ldots,\frac{k-1}{2}\}$

$\iota(a) = a$

$\xi(0_{1,i}) = 0_1, \quad i=1,2,\ldots,\frac{k-1}{2}+1$

It can be immediately verified that M'_a and M''_a are reduced. This completes the proof of Theorem 3.14.

It is clear that if $|S| = k$, then it requires an output set of size k to distinguish each state in S if only state behavior realization is used. Thus, the reduction in the size of the output set can be significant.

Example 3.7

Let M_3 be a single input submachine of some machine with all states in M_3 converging to a single state satisfying Theorem 3.14. Using the technique of the proof in Theorem 3.14, we can construct M'_3 which realizes M_3 with an output set of size 4. If no state splitting were used, it will take an output set of size 6 to obtain a reduced realizing machine.

M_3 =

	a
1	1/0
2	1/0
3	1/0
4	1/0
5	1/0
6	1/0

M'_3 =

	a
1	7/00
2	8/00
3	1/00
4	1/01
5	1/10
6	1/11
7	7/01
8	8/10

$\alpha(i) = i \ \forall \ i \in S - \{1\}$

$\alpha(1) = \{1, 7, 8\}$

$\iota(a) = a$

$\zeta(00) = 0$

$\zeta(01) = 0$

$\zeta(10) = 0$

$\zeta(11) = 0$

Case 2 $\quad \delta(q, a) \notin S \quad \forall \quad q \in S$

and $\quad \omega(\delta(q, a), a) \neq \omega(q, a)$

Theorem 3.15

Let M be a machine which has k states converging to a single state which is not one of the converging states. Then the state set convergence can be broken up by a realizing machine M' (with assignment (α, ι, ζ)) with the following conditions:

$$|\zeta^{-1}(\omega(q, a))| \leq \lfloor \frac{k}{2} \rfloor + 1 \qquad \forall \ q \in S$$

$$|\zeta^{-1}(\omega(\delta(q, a), a))| \leq \lfloor \frac{k}{2} \rfloor + 1$$

and the number of additional states needed does not exceed $k - (\lfloor \frac{k}{2} \rfloor + 1)$.

Proof

Let M_a be the submachine of M satisfying the hypothesis, i. e. $M_a = (\{a\}, \{0_1, 0_2\}, S \cup \{\delta(q, a) | q \in S\}, \delta | (S \cup \{\delta(q, a) | q \in S\}) \times \{a\}, \omega | (S \cup \{\delta(q, a) | q \in S\}) \times \{a\})$. Without loss of generality assume M_a has the following state table description

M_a =

	a
1	$q/0_1$
2	$q/0_1$
3	$q/0_1$
.	.
.	.
.	.
k	$q/0_1$
q	$r/0_2$

Then M_a can be realized by M_a' and M_a'' with $k + \frac{k}{2}$ and $k + \frac{k-1}{2} + 1$ states respectively depending on whether k is even or odd. The state tables and corresponding assignments are shown below:

k = even

M'_a =

	a
1	$q_1/\zeta^{-1}(0_1)$
2	$q_2/\zeta^{-1}(0_1)$
⋮	⋮
$\frac{k}{2}-1$	$q_{\frac{k}{2}-1}/\zeta^{-1}(0_1)$
$\frac{k}{2}$	$q/0_{1,1}$
$\frac{k}{2}+1$	$q/0_{1,2}$
⋮	⋮
k	$q/0_{1,\frac{k}{2}+1}$
q	$r/0_{2,1}$
q_1	$r/0_{2,2}$
q_2	$r/0_{2,3}$
⋮	⋮
$q_{\frac{k}{2}-1}$	$r/0_{2,\frac{k}{2}}$

$$O' = \{0_{1,i}, 0_{2,j} \mid i=1,2,\ldots,\tfrac{k}{2}+1,\ j=1,2,\ldots,\tfrac{k}{2}\}$$

$$S' = \{1,2,\ldots,k,q,q_1,\ldots,q_{\frac{k}{2}-1}\}$$

$\alpha(i) = i \qquad \forall\ i \in S$

$\alpha(q) = q \cup \{q_i \mid i=1,2,\ldots,\frac{k}{2}\}$

$\iota(a) = a$

$\zeta(0_{1,i}) = 0_1, \qquad i=1,2,\ldots,\frac{k}{2}+1$

$\zeta(0_{2,j}) = 0_2, \qquad j=1,2,\ldots,\frac{k}{2}$

k = odd

M''_a =

	a
1	$q_1/\zeta^{-1}(0_1)$
2	$q_2/\zeta^{-1}(0_1)$
⋮	⋮
$\frac{k-1}{2}$	$q_{\frac{k-1}{2}}/\zeta^{-1}(0_1)$
$\frac{k-1}{2}+1$	$q/0_{1,1}$
⋮	⋮
k	$q/0_{1,\frac{k-1}{2}+1}$
q	$r/0_{2,1}$
q_1	$r/0_{2,2}$
q_2	$r/0_{2,3}$
⋮	⋮
$q_{\frac{k-1}{2}}$	$r/0_{2,\frac{k-1}{2}+1}$

$$O'' = \{0_{1,i}, 0_{2,i} \mid i=1,2,\ldots,\tfrac{k-1}{2}+1\}$$

$$S'' = \{1,2,\ldots k, q, q_1, \ldots, q_{\frac{k-1}{2}}\}$$

$\alpha(i) = i \quad \forall\, i \in S$

$\alpha(q) = q \cup \{q_i \mid i=1,2,\ldots,\frac{k-1}{2}\}$

$\iota(a) = a$

$$\left.\begin{aligned}\zeta(0_{1,i}) &= 0_1\\ \zeta(0_{2,i}) &= 0_2\end{aligned}\right\}, \quad i=1,2,\ldots,\tfrac{k-1}{2}+1$$

It can be immediately verified that M_a' and M_a'' are reduced. The size of the output set and the number of additional states required are

$$|\zeta^{-1}(\omega(q, a))| = |\zeta^{-1}(0_1)| = \begin{cases} \frac{k}{2} + 1 \text{ when k is even} \\ \frac{k-1}{2} + 1 \quad \text{otherwise} \end{cases}$$

$$|\zeta^{-1}(\omega(\delta(q, a), a))| = |\zeta^{-1}(0_2)| = \begin{cases} \frac{k}{2} \text{ when k is even} \\ \frac{k-1}{2} \quad \text{otherwise} \end{cases}$$

$$\text{Number of additional states required} = \begin{cases} \frac{k}{2} - 1 \text{ when k is even} \\ \frac{k-1}{2} \quad \text{otherwise} \end{cases}$$

Since if further knowledge of the structure of the original machine were used, it may be possible to reduce the above bounds, these are the upper bounds and thus justifies the inequality expressions. This completes the proof of Theorem 3.15.

We must observe here that the size of the realizing output set does not decrease using the above construction technique. However, if we use the coordinate augmentation type realization, the number of additional output functions that are required to break up the state set convergence may be less than that required for simply augmenting output sets. We shall illustrate this point in the following example.

Example 3.8

Let M_4 be a 6 states submachine of some machine with 5 of its states converging to the sixth state in M_4 and satisfying Theorem 3.15. If we use binary realization for the output set, at least 3 new additional output functions are needed for state behavior realization. Yet by increasing 2 more states only 2 new output functions are needed.

M_4 =

	a
1	6/0
2	6/0
3	6/0
4	6/0
5	6/0
6	r/1

M_4' =

	a
1	7/0xx
2	8/0xx
3	6/000
4	6/001
5	6/010
6	r/100
7	r/101
8	r/110

$\alpha(i) = i \quad i=1, 2, \ldots, 5$

$\alpha(6) = \{6, 7, 8\}$

$\iota(a) = a$

$\zeta(\{000, 001, 010\}) = 0$

$\zeta(\{100, 101, 110\}) = 1$

Case 3 $\delta(q, a) \notin S$

and $\omega(\delta(q, a), a) = \omega(q, a) \quad \forall\, q \in S$

Let $r = \delta(\delta(q, a), a) \quad \forall\, q \in S$. Then there are three cases we have to consider. If $r = \delta(q, a) \quad \forall\, q \in S$ then the treatment is identical to case 1. If $r \neq \delta(q, a)$ and $r \notin S \quad \forall\, q \in S$, then the analysis may require the structure knowledge of the whole machine. This will not be treated here since it may be exhaustive in nature. If $r \neq \delta(q, a)$ yet $r \in S \quad \forall\, q \in S$, then similar techniques to case 1 and case 2 can be developed.

Theorem 3.16

Let M be a sequential machine which has all k states in set S converging to a single state q under some input a where $q \notin S$. If $\delta(q, a) \in S$, then the convergence can be broken up by augmenting no more than $k - (\lfloor \frac{k}{2} \rfloor + 1)$ states and by realizing an output set of size no more than $\lfloor \frac{k}{2} \rfloor + 1$ symbols.

Proof

Assume M_a is the single-input submachine of M having state set $S \cup \{q\}$ and satisfying the hypothesis of the theorem. Without loss of generality, let M_a have the following state table.

M_a =

	a
1	$q/0_1$
2	$q/0_1$
⋮	⋮
k	$q/0_1$
q	$1/0_1$

Using arguments similar to the proof of the previous two theorems, we can construct two machines M'_a and M''_a which realize M_a and are reduced. State tables of M'_a and M''_a are shown on the next two pages.

k even

M'_a =

	a
1	$q_1/0_{1,1}$
2	$q_2/0_{1,1}$
⋮	⋮
$\frac{k}{2}-1$	$q_{\frac{k}{2}-1}/0_{1,1}$
k/2	$q/0_{1,1}$
k/2 +1	$q/0_{1,2}$
⋮	⋮
k	$q/0_{1,\frac{k}{2}+1}$
q	$1/0_{1,1}$
q_1	$1/0_{1,2}$
q_2	$1/0_{1,3}$
⋮	⋮
$q_{\frac{k}{2}-1}$	$1/0_{1,k/2}$

$\alpha(i) = i \qquad \forall\, i \in S$

$\alpha(q) = q \cup \{q_i \mid i=1, 2, \ldots, \frac{k}{2}\}$

$\iota(a) = a$

$\zeta(0_{1,i}) = 0_1 , \qquad i=1, 2, \ldots, \frac{k}{2}+1$

k = odd

	a
1	$q_1/0_{1,1}$
2	$q_2/0_{1,1}$
⋮	⋮
$\frac{k-1}{2}$	$q_{\frac{k-1}{2}}/0_{1,1}$
$\frac{k-1}{2}+1$	$q/0_{1,1}$
⋮	⋮
k	$q/0_{1,\frac{k-1}{2}+1}$
q	$1/0_{1,1}$
q_1	$1/0_{1,2}$
q_2	$1/0_{1,3}$
⋮	⋮
$q_{\frac{k-1}{2}}$	$1/0_{1,\frac{k-1}{2}+1}$

$\alpha(i) = i \quad \forall\, i \in S$

$\alpha(q) = q \cup \{q_i \mid i=1,2,\ldots,\frac{k-1}{2}\}$

$\iota(a) = a$

$\zeta(0_{1,i}) = 0_1\,, \qquad i=1,2,\ldots,\frac{k-1}{2}+1$

The size of the augmented state set and the realizing output set can be verified immediately. This completes the proof of Theorem 3.16.

We now formally present the method for breaking up a state set convergence by augmenting both the state set and output set. Let M be a machine which has k states $S = \{s_1, s_2, \ldots, s_k\}$ converging to a single state q under some input a.

1) If $q \in S$, then augment to the original state table $\lceil \frac{k}{2} \rceil - 1$ states, $q_1, q_2, \ldots, q_{\lceil \frac{k}{2} \rceil - 1}$. Set $\delta(q_i, a) = q_i$ and $\omega(q_i, a) = 0_{1, i+1}$ for $i=1, 2, \ldots, \lceil \frac{k}{2} \rceil - 1$. Choose $R = \{r_{i_1}, r_{i_2}, \ldots, r_{i_{\lceil \frac{k}{2} \rceil - 1}}\} \subseteq S$ and $r_{i_1} = q$. Replace $\delta(r_{i_j}, a)$ by q_j and $\omega(r_{i_j}, a)$ by $0_{1,1}$ for $j=1, 2, \ldots, \lceil \frac{k}{2} \rceil$. For the remaining $\lfloor \frac{k}{2} \rfloor + 1$ states in $S - R = \{s_{i_1}, s_{i_2}, \ldots, s_{i_{\lfloor \frac{k}{2} \rfloor + 1}}\}$, replace $\omega(s_{i_j}, a)$ by $0_{1, j}$ for $j+1, 2, \ldots, \lfloor \frac{k}{2} \rfloor + 1$.

2) If $q \notin S$ and $\omega(q, a) \neq \omega(r, a)$ $\forall$ $r \in S$ then augment to the original state table $\lceil \frac{k}{2} \rceil - 1$ states, $q_1, q_2, \ldots, q_{\lceil \frac{k}{2} \rceil - 1}$. Set $\delta(q_i, a) = \delta(q, a)$, $\omega(q_i, a) = 0_{2, i+1}$ for $i=1, 2, \ldots, \lceil \frac{k}{2} \rceil - 1$ and $\omega(q, a) = 0_{2,1}$. Choose $R = \{r_{i_1}, r_{i_2}, \ldots, r_{i_{\lceil \frac{k}{2} \rceil - 1}}\} \subseteq S$. Replace $\delta(r_{i_j}, a)$ by q_j and $\omega(r_{i_j}, a)$ by $\{0_{1, i} \mid i=1, 2, \ldots, \lfloor \frac{k}{2} \rfloor + 1\}$. For the remaining

$\lfloor \frac{k}{2} \rfloor + 1$ states in $S - R = \{s_{i_1}, s_{i_2}, \ldots, s_{i_{\lfloor \frac{k}{2} \rfloor + 1}}\}$, replace $\omega(s_{i_j}, a)$ by $0_{1,j}$ for $j=1, 2, \ldots, \lfloor \frac{k}{2} \rfloor + 1$. This modified state table represents an output partially specified machine which realizes the original machine with the state set convergence of S broken up.

3) If $q \notin S$, $\omega(q, a) = \omega(r, a)$ $\forall$ $r \in S$ and $\delta(q, a) \in S$, then augment to the original state table $\lceil \frac{k}{2} \rceil - 1$ states, $q_1, q_2, \ldots, q_{\lceil \frac{k}{2} \rceil - 1}$. Set $\delta(q_i, a) = \delta(q, a)$, $\omega(q, a) = 0_{1,1}$ and $\omega(q_i, a) = 0_{1, i+1}$ for $i=1, 2, \ldots, \lceil \frac{k}{2} \rceil - 1$. Choose $R = \{r_{i_1}, r_{i_2}, \ldots, r_{i_{\lceil \frac{k}{2} \rceil - 1}}\} \subseteq S$ such that $\delta(q, a) \in R$.

Replace $\delta(r_{i_j}, a)$ by q_j and $\omega(r_{i_j}, a)$ by $0_{1,1}$ for $j=1, 2, \ldots, \lceil \frac{k}{2} \rceil - 1$. For the remaining $\lfloor \frac{k}{2} \rfloor + 1$ states in $S - R = \{s_{i_1}, s_{i_2}, \ldots, s_{i_{\lfloor \frac{k}{2} \rfloor + 1}}\}$ replace $\omega(s_{i_j}, a)$ by $0_{1,j}$ for $j=1, 2, \ldots, \lfloor \frac{k}{2} \rfloor + 1$.

Acyclic Testing Graph Realization

Let $L_1, L_2, \ldots, L_k$ be the set of cycles in the testing graph T_a of a single input submachine M_a of a machine M. Since $L_i \cap L_j = \phi$ for $i \neq j$, it is clear that removing one branch from each L_i is both necessary and sufficient for acyclic realization. A branch in L_i is removed if the corresponding state pair in the realizing machine has distinguishable output symbols. Thus if M realizes M with an assignment $(\mathcal{C}_Q, \mathcal{C}_I, \zeta)$, and $\omega(q, a) = \omega(r, a)$ for $q, r \in Q$, $a \in I$, and $\{q, r\}$ is a node in L_i, then $\zeta^{-1}(\omega(q,a)) \neq \zeta^{-1}(\omega(r, a))$ in the realizing machine will eliminate the branch emanating from the node $\{q, r\}$ in the corresponding testing graph.

It appears to be very difficult to devise an algorithm which will produce either the minimum number of realizing output symbols or output terminals for acyclic testing graph realization. However, a heuristic procedure has been designed which will find an output assignment requiring near-minimum number of additional output terminals. This is presented below.

Two nodes in the testing graph are said to be <u>disjoint</u> if there is no common state. For $0_i \in O$, let ℓ_i be the maximum number of cycles in $T_a \ni$ each has a node with output labeling of 0_i and each pair of nodes is disjoint.

(1) Choose $0_j \in O \ni \ell_j = \max_{i=1}^{k} \ell_i$. Assign $0_{j,1}$ and $0_{j,2}$ to the outpu' of each state pair. This should open exactly ℓ_j cycles.

(2) Remove all branches in T_a which are implied by these assignments. A branch is implied if the assignments of (2) force the removal of this branch.

(3) Repeat (1) and (2) on the remaining cycles.

(4) If (1) - (3) do not open all cycles in T_a, then the remaining cycles must be partially specified, i.e. with the output of one state of a node assigned and the other one not yet assigned. Try to assign the not yet assigned state with the existing new output symbols. If conflicts occur, a new output symbol will have to be created and assigned to the output of this state.

(5) Repeat (4) until all cycles are open. The justification for employing steps (1) to (3) is to try to open all cycles by assigning to each original output symbol, two new realizing output symbols. If this cannot be done, then steps (4) and (5) is employed which augments additional output symbols as are necessary. Thus the near-minimum claim is sustained. If an exhaustive search is used in Steps (4) and (5), then it is conjectured that an absolute minimum on the additional output terminals may be obtained.

For an arbitrary n-state single input machine, $k \leq \binom{n}{2}$. For example, consider a single input identity machine with the same output for every state, i.e. $\delta(q, a) = q$ and $\omega(q, a) = \omega(r, a)$ $\forall q, r \in Q$. The testing graph of this machine consists of $\binom{n}{2}$ nodes, each with a self loop. We will show in the following that it is possible to obtain an acyclic testing graph realization by augmenting both the state set and the output set in a way that reduces the size of the realizing output set when compared with using state behavior realization.

Theorem 3.17

Let $M = (\{a\}, \{0_1\}, Q, \delta, \omega)$ be a single-input, single-output, n-state identity machine, i.e. $\delta(q, a) = q$ and $\omega(q, a) = \omega(r, a)$ $\forall q, r \in Q$. Then it is possible to construct a reduced single-input machine $M' = (\{a\}, O', Q', \delta', \omega')$ which realizes M with

$$|O'| \leq \left\lfloor \frac{n}{2} \right\rfloor + 1$$

$$|Q'| \leq n + \left\lceil \frac{n}{2} \right\rceil - 1$$

Proof

The proof is very similar to that of Theorem 3.14. Without loss of generality, consider M to have a state table description as follows:

$M =$

	a
1	$1/0_1$
2	$2/0_1$
.	.
.	.
.	.
n	$n/0_1$

Then M' shown below realizes M and satisfies conditions stated in the theorem.

$M' =$

	a
1	$1'/0_{1,1}$
2	$2'/0_{1,1}$
⋮	⋮
$\left\lceil \frac{n}{2} \right\rceil - 1$	$\left\lceil \frac{n}{2} \right\rceil - 1'/0_{1,1}$
$\left\lceil \frac{n}{2} \right\rceil$	$\left\lceil \frac{n}{2} \right\rceil /0_{1,1}$
$\left\lceil \frac{n}{2} \right\rceil + 1$	$\left\lceil \frac{n}{2} \right\rceil + 1/0_{1,2}$
⋮	
n	$n/0_{1,\left\lfloor \frac{n}{2} \right\rfloor + 1}$
1'	$1/0_{1,2}$
2'	$2/0_{1,3}$
⋮	⋮
$\left\lceil \frac{n}{2} \right\rceil - 1'$	$\left\lceil \frac{n}{2} \right\rceil - 1/0_{1,\left\lceil \frac{n}{2} \right\rceil}$

$$\alpha(i) = \{i\} \cup \{i'\} \quad \forall\, i = 1, 2, \ldots \left\lceil \frac{n}{2} \right\rceil - 1$$

$$\alpha(i) = i \quad \text{otherwise}$$

$$\iota(a) = a$$

$$\zeta(0_{1,i}) = 0_1 \quad \forall\, i = 1, 2, \ldots \left\lfloor \frac{n}{2} \right\rfloor + 1$$

This completes the proof of Theorem 3.17.

Example 3.9

M_5 is realized by M_5' with their state tables shown below:

M_5 =

	a
1	1/1
2	2/1
3	3/1
4	4/1
5	5/1

M_5' =

	a
1	1'/1 0 0
2	2'/1 0 0
3	3/1 0 0
4	4/1 0 1
5	5/1 0 1
1'	1/1 0 1
2'	2/1 1 0

$$\alpha(i) = \{i\} \cup \{i'\} \quad i = 1, 2$$

$$\alpha(i) = i \quad i = 3, 4, 5$$

$$\iota(a) = a$$

$$\zeta(\{100, 101, 110\}) = 1$$

If we use state behavior realization by augmenting only the output set, the size of the realizing output set is 5 which requires 3 additional output terminals when binary realization is used. M_5' requires an output set of size 3 which requires only 2 additional output terminals using binary realization.

General Problem

We now turn to a more general problem of designing a diagnosable machine without being restricted to repeated symbol distinguishing sequence. Suppose M is a reduced n-state machine. Then from Moore [18], there exists a homing sequence x of length not greater than $\binom{n}{2}$. From Theorem 3.2 we know that if this sequence is not a distinguishing sequence, then there exist some state pair convergences under x. To design a diagnosable machine from M, it is only necessary to break up these state pair convergences using techniques described earlier. The number of additional output symbols needed depends on the number of state pair convergences and their distribution. If there are k state pair convergences, then the number of additional output symbols needs not exceed k. The order of diagnosability may have to be increased if the size of the realizing output set must be bounded by some number. In general, the size of the realizing output set or the number of additional states required are less than that required by designing a rds diagnosable machine. If M is redundant, then a reduced version of M must first be found before applying the technique described above. Additional output symbols are then assigned to distinguish the equivalent states.

Some Diagnosable Properties of Sequence Generators and Linear Machines

We have observed in many instances that the single-input submachines of a sequential machine determine many important properties of the complete machine. For example, a machine is diagnosable with a repeated symbol distinguishing sequence if one of its single input submachines is reduced. Independent studies of this class of machines is therefore justified for the purpose of diagnosis. In fact, this class of machine has been widely studied in the context of machine decomposition and sequence generation. We begin by introducing the following machine properties which can be used to characterize diagnosable machines.

Definition 3.9

A k-diagnosable machine is said to be optimally diagnosable if

$$k = \lceil \log_p n \rceil$$

where

$$n = |Q|$$

$$p = |O|$$

Definition 3.10

Two states $q, r \in Q$ in a sequential machine are k-equivalent if whenever the length of input sequence x is less than or equal to k, $\beta_q(x) = \beta_r(x)$. The partition on Q induced by k-equivalence

(denoted Π_k) is called the k-equivalence partition.

Recall that the lower bound of the length of a distinguishing sequence in an n-state, p-output machine is $\lceil \log_p n \rceil$. Thus an optimally diagnosable machine has a shortest possible d. s. The next lemma relates the length of the shortest d. s. in a sequence generator to the state equivalence partition.

Lemma 3.2

If M is a reduced sequence generator, then M is k-diagnosable iff k is the least integer such that $\Pi_k = \Pi_{k+1}$.

Proof

Suppose M is k-diagnosable, then for any $x \in I^k$ of length k, $\beta_q(x) = \beta_r(x) \Longrightarrow q = r$.

$$\therefore \Pi_k = 0 = \Pi_{k+1}$$

But if $\ell g(x) = k-1$, x is not a d. s. of M and $\exists\, q, r \in Q \ni \beta_q(x) = \beta_r(x)$ yet $q \neq r$.

$$\therefore \Pi_{k-1} \neq 0 = \Pi_k$$

Conversely, if k is the least integer such that $\Pi_k = \Pi_{k+1}$, then since M is reduced and $\Pi_k \geq \Pi_{k+1}$, $\Pi_k - \Pi_{k+1} = 0$. $\Pi_{k-1} \geq \Pi_k$ and $\Pi_{k-1} \neq \Pi_k$, then $\exists\, q, r \in Q \ni q$ and r are k-1 equivalent. Therefore, the length of the distinguishing sequence is k and M is k-diagnosable.

With this lemma, the following characterization of a minimally diagnosable sequence generator is obtained naturally.

Theorem 3.18

Let M be a reduced p-nary sequence generator with n states, where $n = p^m$. Then M is optimally diagnosable iff $p|\Pi_k| = |\Pi_{k+1}| \; \forall k < m$.

Proof

Assume M has a d.s. of length m but $\exists k < m$ such that $|\Pi_{k+1}| \neq p|\Pi_k|$ but $|\Pi_{i+1}| = p|\Pi_i| \forall i < k$. Then clearly $|\Pi_k| = p|\Pi_{k-1}| = p^k$. If $|\Pi_k| = |\Pi_{k+1}|$, then $\Pi_k = \Pi_{k+1}$, since $\Pi_{k+1} \leq \Pi_k$ for all k. This means that the machine is not reduced, contrary to our original assumption that M is reduced. So, let us assume $|\Pi_{k+1}| > |\Pi_k|$. Then

$$p^{k+1} > |\Pi_{k+1}| > p^k$$

Then at least $\exists$ one block B_{k+1} of $\Pi_{k+1} \ni \left\lceil \frac{p^{m-k}}{p-1} \right\rceil \leq |B_{k+1}| \leq p^{m-k}$. The shortest sequence of p-nary partitions that can partition this block to singleton sets is of length m-k. Therefore, the shortest d.s. is of length k+1+m-k = m+1, contrary to our assumtion that M has a d.s. of length m.

Conversely, if $|\Pi_{k+1}| = p|\Pi_k| \forall k \leq m$. Then when k = m-1

$$|\Pi_m| = p|\Pi_{m-1}| = p p^{m-1} = p^m$$

which is the total number of states.

$\therefore$ The machine has a d.s. of length m.

For the case where n is not a power of p, we have the following theorem.

Theorem 3.19

Let M be a reduced p-nary sequence generator with n states. Let B_k be a largest k-equivalence class. Then M is optimally diagnosable iff $|B_k| \leq p^{m-k}$ $\forall$ k $\leq$ m where $m = \lceil \log_p n \rceil$.

Proof

If there exist B_k in Π_k such that $|B_k| > p^{m-k}$, then the shortest possible d.s. that will partition B_k into singleton states is of length $> m-k$. Consequently, the shortest d.s. for M would be $> m-k+k$ and M can not have a d.s. of length m.

Conversely, where k = m

$$|B_k| \leq p^\circ = 1$$

and the sequence of length m is a d.s. This completes the proof.

An important class of sequential machines is the so-called linear machines [2]. It has been found that the linear machines have many

interesting properties relating to diagnosis. We first formally introduce the linear machine in the following definition.

Definition 3.11

A sequential machine $M = (I, O, Q, \delta, \omega)$ is said to be linear over a finite field F if there exist integers n, k, and ℓ such that

$$Q = V^n(F)$$

$$I = V^k(F)$$

$$O = V^{\ell}(F)$$

and linear transformations

$$A: Q \rightarrow Q$$

$$B: I \rightarrow Q$$

$$C: Q \rightarrow O$$

$$D: I \rightarrow O$$

such that

$$\delta(q, a) = Aq + Ba$$

$$\omega(q, a) = Cq + Da$$

In words, a sequential machine is said to be linear if its next state vector and output vector can be represented as linear combinations of the present state and present input vectors. The input-output behavior of a linear machine can be defined recursively in terms of

the linear transformations A, B, C, and D. This is introduced as the next lemma [4].

Lemma 3.3

Let M be the linear machine defined in Definition 3 and $x \in I^n$.

Then $\forall\, q \in Q$

$$\beta_q(x) = K_n q + Q_n x$$

for some linear transformations K_n and Q_n, where

$$K_n: Q \to O^n$$

$$Q_n: I^n \to O^n$$

Proof

Let

$$x = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \quad \text{where } a_i \in I.$$

From Definition 3.11

$$\omega(q, a_1) = Cq + Da_1$$

$$\delta(q, a_1) = Aq + Ba_1$$

$$\bar{\omega}(q,\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}) = \omega(\delta(q,a_1),a_2)$$

$$= \omega(\delta(\delta,a_1),a_2)$$

$$= C\delta(q,a_1) + Da_2$$

$$= C(Aq + Ba_1) + Da_2$$

$$= CAq + CBa_1 + Da_2$$

$$\bar{\delta}(q,\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}) = A\delta(q,a_1) + Ba_2$$

$$= A(Aq + Ba_1) + Ba_2$$

$$= A^2q + ABa_1 + Ba_2$$

$$\vdots$$

$$\bar{\omega}(\mathbf{a},x) = CA^{n-1}q + CA^{n-2}Ba_1 + CA^{n-3}Ba_2 + \ldots$$

$$+ CABa_{n-2} + CBa_{n-1} + Da_n$$

$$\bar{\delta}(q,x) = A^nq + A^{n-1}Ba_1 + A^{n-2}Ba_2 + \ldots + ABa_{n-1} + Ba_n$$

Thus $\beta_q(x)$ can be written as following:

$$\beta_q(x) = \begin{bmatrix} \bar{\omega}(q,a_1) \\ \bar{\omega}(q,\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}) \\ \bar{\omega}(q,\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}) \\ \vdots \\ \bar{\omega}(q,x) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} q +$$

$$\begin{bmatrix} D & O & --- & O & O \\ CB & D & --- & O & O \\ CAB & CB & & O & O \\ \vdots & \vdots & & & \vdots \\ CA^{n-2}B & CA^{n-3}B & CA^{n-4}B & --- CB & D \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix}$$

Then, $\beta_q(x)$ can be written as

$$\beta_q(x) = K_n q + Q_n x$$

where

$$K_n = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}, \quad Q_n = \begin{bmatrix} D & O & --- & O & O \\ CB & D & --- & O & O \\ CAB & CB & --- & O & O \\ \vdots & & & \vdots & \vdots \\ CA^{n-2}B & CA^{n-3}B & --- & CB & D \end{bmatrix}$$

Theorem 3.20

If M is a reduced linear machine, then the following are equivalent:

i) M is definitely diagnosable of order i

ii) i is the least integer such that rank $(K_i) = n$

iii) i is the least integer such that $\Pi_i = \Pi_{i+1}$

Proof

i) $\Longrightarrow$ ii)

Suppose M is definitely diagnosable of order i, then $\forall\, x \in I^i$, we have

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

Recall that

$$\beta_q(x) = K_i q + Q_i x$$

where

$$K_i: Q \to O^i$$

$$O_i: I^i \to O^i$$

First we show that $\beta_q(x) = \beta_r(x) \Longleftrightarrow K_i q = K_i r$. Suppose $\beta_q(x) = \beta_r(x)$, then $K_i q + Q_i x = K_i r + Q_i x\ \forall\, x \in I^i$. Subtracting $Q_i x$ from both sides, we have $K_i q = K_i r$. Conversely, if $K_i q = K_i r$, then $K_i q + Q_i x = K_i r + Q_i x$ and we have $\beta_q(x) = \beta_r(x)$. Since $\forall x \in I^i$

$$\beta_q(x) = \beta_e(x) \Longrightarrow q = r$$

we have

$$K_i q = K_i r \Longrightarrow q = r$$

But this says that K_i is a 1-1 linear transformation from Q to O^i and since dim Q = n, it follows then

$$\text{rank } (K_i) = n$$

By definition of the order of definite diagnosability i is the least such integer.

ii) $\Longrightarrow$ iii)

Suppose rank $(K_i) = n$

$$K_i: Q \to O^i$$

then K_i is a 1-1 linear transformation

$$\therefore K_i q = K_i r \Longrightarrow q = r, \ \forall\ q, r \in Q$$

But $K_i q = K_i r \Longleftrightarrow \beta_q(x) = \beta_r(x), \ \forall\ x \in I^i$

$$\therefore \beta_q(x) = \beta_r(x) \Longrightarrow q = r, \ \forall\ x \in I^i$$

But $\beta_q(x) = \beta_r(x)\ \forall\ x \in I^i \Longleftrightarrow q \equiv r\ (\Pi_i)$

$$\therefore q \equiv r\ (\Pi_i) \Longrightarrow q = r$$

$$\therefore \Pi_i = 0$$

Since $\Pi_i \geq \Pi_{i+1}$, it follows then $\Pi_i = \Pi_{i+1}$.

iii) $\Longrightarrow$ i)

Suppose i is the least integer such that $\Pi_i = \Pi_{i+1}$, then since M is reduced, $\Pi_i = \Pi_{i+1} = 0$. Then $\forall$ pair $q, r \in Q$, $q \neq r$, $\exists\ x \in I^i \ni \beta_q(x) \neq \beta_r(x)$ which in turn implies $K_i q \neq K_i r$. Thus K_i is a 1-1 linear transformation and any $x \in I^i$ is a d.s.. Therefore, M is d.d. of order i.

Corollary 3.20.1

Every reduced linear sequential machine is definitely diagnosable and the maximum order of the definite diagnosability is n, where n = dim Q.

Proof

If M is a reduced linear machine, then $\exists$ i such that $\Pi_i = \Pi_{i+1} = 0$. Now by iii) of Theorem 3.20, M is definitely diagnosable of order i if i is the least integer such that $\Pi_i = \Pi_{i+1}$. Then rank (K_i) = n, since K_i is a 1-1 linear transformation. Since the rank of A is $\leq$ n and by definition of K_i, it follows that $i \leq n$. Thus the maximum $i \ni \text{Ran}(K_i) = n$.

In other words, every input sequence of length n in a reduced linear machine is a d.s. where n is the dimension of the state space. Thus in the case of linear machines, we have a reduced upperbound on the length of d.s.. Note that in a linear machine the concept of being i-diagnosable is equivalent to that of being definitely diagnosable of order i. Since the minimum length of the d.s. in a linear machine can be characterized by the least integer i such that rank (K_i) = n, we can characterize the optimally diagnosable linear machine with the following corollary of Theorem 3.20.

Corollary 3.20.2

Let M be a reduced linear machine with n = dim Q, ℓ = dim $\mathcal{R}(\omega)$ = dim O and k = dim I. Then M is optimally diagnosable iff rank $(K_{\lceil \frac{n}{\ell} \rceil})$ = n.

Recall that a linear machine is reduced if and only if it is definitely diagnosable. This result leads to the observation that the state set is partitioned evenly by each input symbol. Since the state space can be considered as an additive abelian group and the linear transformation $K_i : Q \to O^i$ [see Lemma 3.3] induces a congruence relation on Q, it follows that the partition corresponding to this congruence relation is a coset partition. More precisely,

Lemma 3.4

If M is a linear machine, then every i-equivalence relation partitions the state set into cosets when the state set is considered as an additive abelian group.

The cardinality of these cosets can be shown to be a power of a prime number. Thus, we can obtain a recursive relation on the i-equivalence relation of a linear machine. Denoting the i-equivalence partition on Q as π_i, we have

Theorem 3.21

Let M be a linear machine over a finite field F of characteristic p and $|F| = p^m$, $n = \dim Q$ and $\ell = \dim O$. Then $\forall i$, $\exists k_i$, $0 \leq k_i \leq \ell m$ such that $|\pi_{i+1}| = p^{k_i} |\pi_i|$.

Proof

From Lemma 3.4 we know that π_i partitions Q into cosets. Let H_i be the set of states which are i-equivalent to $\vec{0}$. Then H_i is a subgroup of Q. The order of H_i divides that of Q. Now the cardinality of Q is a power of p, i.e. Q is a p-group, it follows that H_i is also a p-group or $H_i = \{\vec{0}\}$. If $H_i = \{\vec{0}\}$ or $H_i = H_{i+1}$ then clearly $k_i = 0$. If $H_i \neq \{\vec{0}\}$, then $|H_i| = p^{j_i}$

$$|\pi_i| = |Q/H_i| = p^{mn-j_i}$$

$$|\pi_{i+1}| = |Q/H_{i+1}| = p^{mn-j_{i+1}}$$

Therefore

$$|\pi_{i+1}| = p^{j_i - j_{i+1}} |\pi_i|.$$

Since $H_{i+1} \subseteq H_i$, it can be shown that $0 \leq j_i - j_{i+1} < \ell m$. Let $k_i = j_i - j_{i+1}$ and the theorem is proved.

Theorem 3.21 says that in a linear machine the i-equivalence classes of the state set always "grow" as a power of the field characteristic p. By the coset structure, each i-equivalence class has the same cardinality which is also a power of p. In case the dimension of the output space divides that of the state space, we can characterize an optimally diagnosable linear machine in the following way.

Theorem 3.22

Let M be a reduced linear machine with $q = p^m$, ℓ and n given as in Theorem 3.21 and $\ell | n$. Then M is optimally diagnosable iff $|\pi_{i+1}| = q^{\ell}|\pi_i| \ \forall i \ \ 1 \leq i < \frac{n}{\ell}$.

Proof

Assume M is optimally diagnosable, then $\forall x \in I^{\lceil \frac{n}{\ell} \rceil}$, x is a d.s. of M. Since $\ell | n$, $\lceil \frac{n}{\ell} \rceil = \frac{n}{\ell}$, we claim that $|\pi_{i+1}| = q^{\ell}|\pi_i| \ \forall$ $1 \leq i < \frac{n}{\ell}$. Suppose there exists some i such that $|\pi_{i+1}| \neq q^{\ell}|\pi_i|$. From Theorem 3.21, we have $|\pi_{i+1}| = p^{k_i}|\pi_i|$, and our last assumption says $k_i \neq \ell m$, i.e. $k_i < \ell m$. But this says in turn that $|\pi_i| < p^{i\ell m}$ and $|\pi_{\frac{n}{\ell}}| < p^{mn}$. Thus $x \in I^{\frac{n}{\ell}}$ can not be a d.s., contrary to our assumption that M is optimally diagnosable. Conversely, if $|\pi_{i+1}| = q^{\ell}|\pi_i|$, then $\forall x \in I^{\frac{n}{\ell}}$, x is a d.s., i.e. M is optimally diagnosable.

The result of Theorem 3.21 can be generalized to non-linear sequential machines. We first introduce a concept of "generalized equivalence relation" under some input sequence.

Definition 3.12

Let $M = (I, O, Q, \delta, \omega)$ be a sequential machine and let $x \in I^{\dagger}$ where $x = y\,a$ $(y \in I^{*}, a \in I)$. Then $q R_x r$ if $\omega(\bar{\delta}(q, y), a) = \omega(\bar{\delta}(r, y), a)$. The partition induced by R_x is denoted π_x.

In other words, $q R_x r$ if the last output symbol for input sequence x is the same for initial states q and r.

Theorem 3.23

If $M = (I, O, Q, \delta, \omega)$ is a sequential machine, $|Q| = p^n$, $|O| = p$ (p is an integer). Then M is optimally diagnosable iff $\exists\, x = a_1 a_2 \ldots a_n$ such that $|\pi_{x_{i+1}}| = p\,|\pi_{x_i}|$ $\forall\, 1 \leq i < n$ where $x_i = a_1 a_2 \ldots a_i$.

Proof

To prove necessity, suppose there is no $x_n \in I^n$ such that $|\pi_{x_{i+1}}| = p\,|\pi_{x_i}|$ for $1 \leq i < n$. Then $\forall\, x_n \in I^n$, x_n can not be a d.s. of M because $|\pi_{x_n}| \neq p^n$ implies $\pi_{x_n} \neq 0$. Thus by definition, M is not optimally diagnosable. To show sufficiency, suppose M is not optimally diagnosable. Then every sequence of length n can not be a d.s. This says that $\exists\, q \neq r$ in $Q \ni \forall\, x_n \in I^n$, $\beta_q(x) = \beta_r(x)$ i.e. $q \equiv_n r$. This means that $\forall\, x_n \in I^n$, $\exists i$, $1 \leq i < n$ (depending on x_n), such that $|\pi_{x_{i+1}}| \neq p\,|\pi_{x_i}|$.

Theorem 3.23 provides a useful means of finding an optimal distinguishing sequence. Any input symbol which does not partition the state set into p equivalence classes can be excluded as the starting symbol of an optimal d. s. On each i-equivalence class of states any input symbol which does not partition it into p equivalence subclasses is excluded from being the next symbol of an optimal d. s. The process is then iterated until an optimal d. s. of length n is found or no optimal d. s. can be found. In the case of linear machines, the same input symbol can be examined at each step since each input symbol has an "equivalent" effect on the partitioning process.

Returning now to the question of designing an optimally diagnosable machine which realizes a given machine behavior. From Theorem 3. 23 we found that π_x can be used to characterize an optimally diagnosable machine M when M satisfies the theorem's hypothesis. By Definition 3.12, π_x is determined by both the output function ω and the state transition function δ. This observation seems to indicate that with proper choice of δ, we may be able to come up with an optimally diagnosable machine for any given diagnosable machine. This conjecture can be easily disproved by the following arguments. First we recall that if a machine M is diagnosable, then M must be reduced. Let M and M' be two reduced machines such that $M \equiv M'$. Then it can be shown that there exists a strong isomorphism between M and M' [11]. This says that the state behaviors of M and M' are isomorphic. Thus no choice of δ is possible. This conclusion should not discourage

our interest in the study of diagnosable and optimally diagnosable machines. For example, in the case of a redundant machine which realizes a given behavior, there are certain freedom which allow the designer to choose different δ and ω. Furthermore, the selection of δ and ω can be incorporated into the design of fault-tolerant switching network which realizes a given machine behavior. This is the main theme of our investigation.

Network Diagnosis

Network diagnosis is a much more difficult problem than the type of machine diagnosis discussed in the last section. At the outset, before anything can be said about synthesis and sequential networks, analysis techniques must be developed for combinational networks.

A combinational network refers to a physical realization of some Boolean function; for the purpose of this discussion, it is an interconnection of gate-type primitive logic elements. We first introduce a directed graph representation of a combinational network convenient for analyzing gate input-output stuch-at faults. The notion of "path sensitizing" [1] and "node sensitizing" is then formalized so that its limitations and capabilities can be better understood. Finally, the problems of test point allocation and multiple faults detection are discussed.

To derive a useful representation of a network convenient for stuck-at fault analysis, we must first look at the effect of these faults at an elementary level. Consider, for example, a three-input AND gate. The effect of any single input stuck-at-0 is indistinguishable from the effect of the gate output being stuck-at-0. However, the effect of any single input being stuck-at-1 is different from any of the gate output stuck-at faults. If we represent each gate by a node, then a gate output stuck-at fault is just a constant node function. However, if a preceeding gate which feeds one of the input has a non-unity fan out, the input stuck-at faults may not be represented as a constant node function.

To facilitate a uniform treatment of the stuck-at faults as constant node functions, we give the following definition of a (node) network graph.

Definition 3.13A

A (node) network graph of a combinational network is a directed graph representation of that network constructed according to the following rules:

1) Each gate and each input terminal corresponds to a node in the network.
2) A node is added to each branch of a fan-out node. This is done to segregate the state of the fan-out gate and that of the succeeding lines it feeds.
3) A directed branch is drawn from node i to node j when an input of node j comes from node i.

Similarly, a branch oriented network graph can also be constructed where all stuck-at faults are represented by constant branch functions. The following is such a definition.

Definition 3.13B

A (branch) network graph is a directed graph representation of a combinational network which is constructed according to the following rules:

1) Each gate, each network input terminal, and each network output terminal corresponds to a node in the graph.

2) A node is created at the place where fan out occurs except when the fan out occurs at the network input. This is done to segregate the state of the gate output line from that of the succeeding input lines it feeds.
3) A directed branch is drawn from node i to node j when an input of node j comes from node i.

Since a network graph of a combinational network is acyclic, it is possible to label all nodes by integers so that node i is adjacent from nodes of labelings smaller than i. The nodes in the network graph can also be divided into levels. Level 1 consists of all primary input nodes. Level k consists of all nodes which are fed from nodes of level lower than k. Figure 3.8(a) and (b) show a combinational net-work and its (node) network graph. A path in a network graph is defined in the usual manner except that the path is always terminated at a primary output. Thus a path can be uniquely represented by a sequence of nodes with labellings in strictly increasing order.

Similar labeling technique can be applied to the branch network graph as shown in Figure 3.8(c). In the following discussion, we will be dealing with node type network graph only unless otherwise specified.

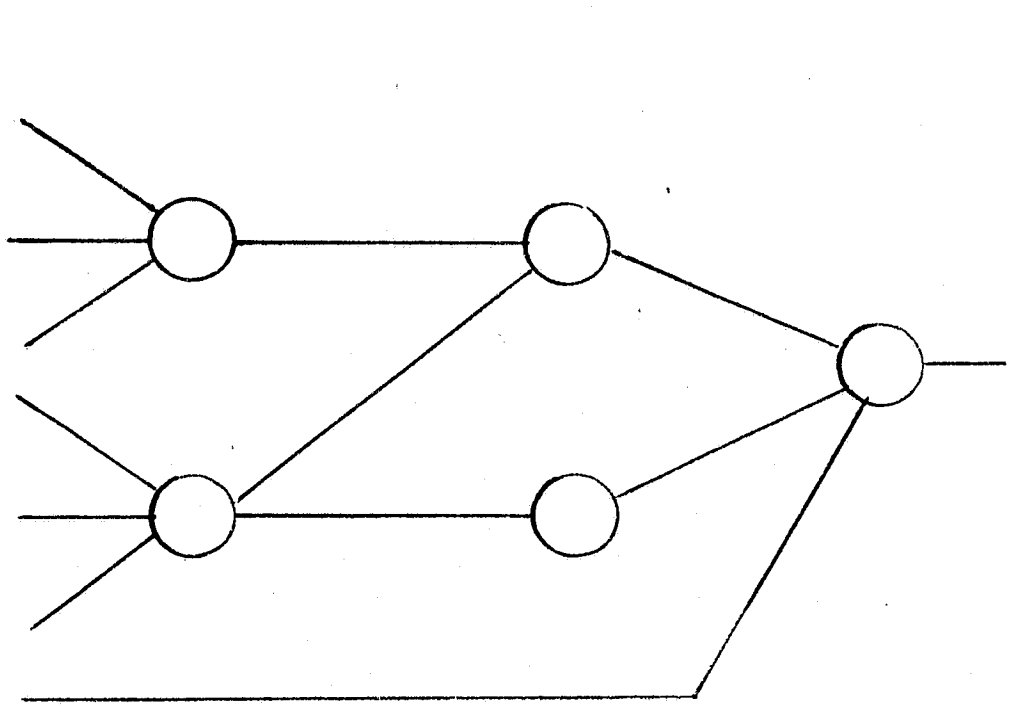


Figure 3.8(a) A Combinational Network N

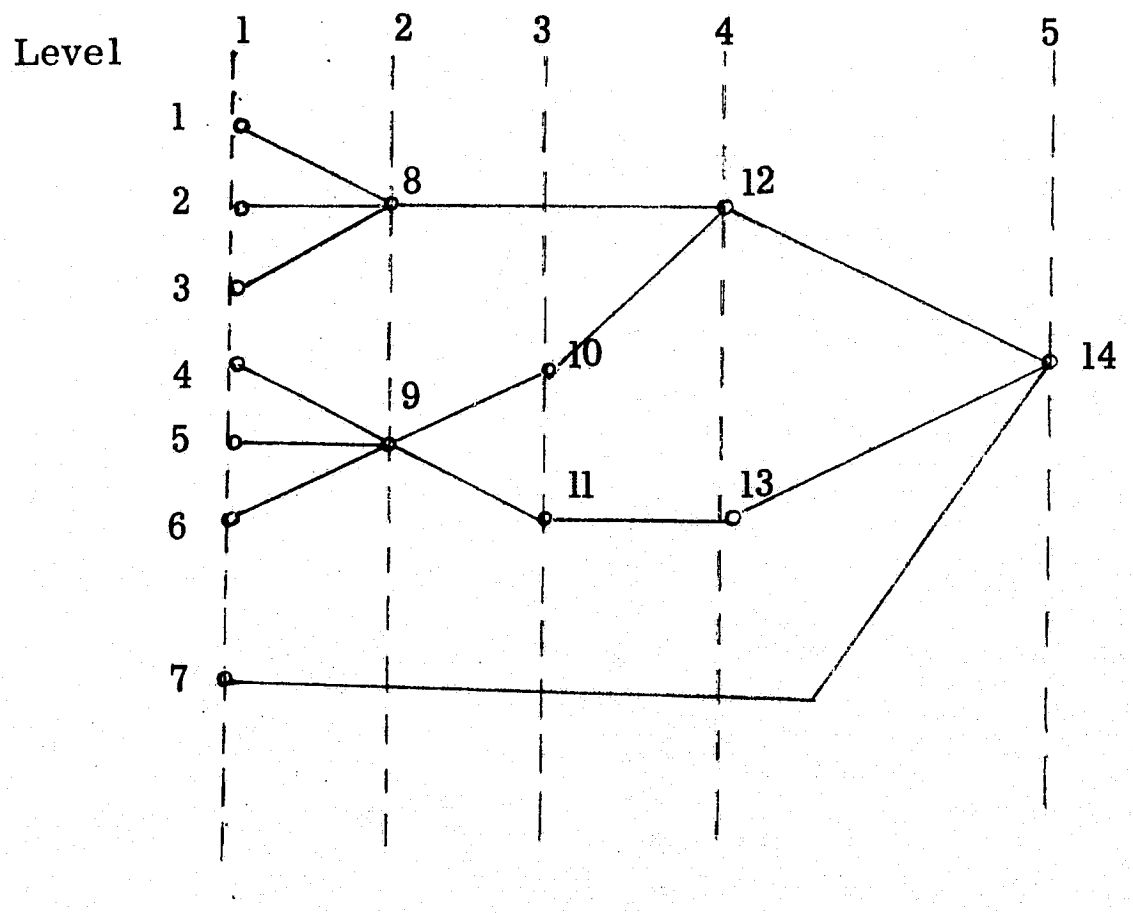


Figure 3.8(b) The (node) Network Graph of (a)

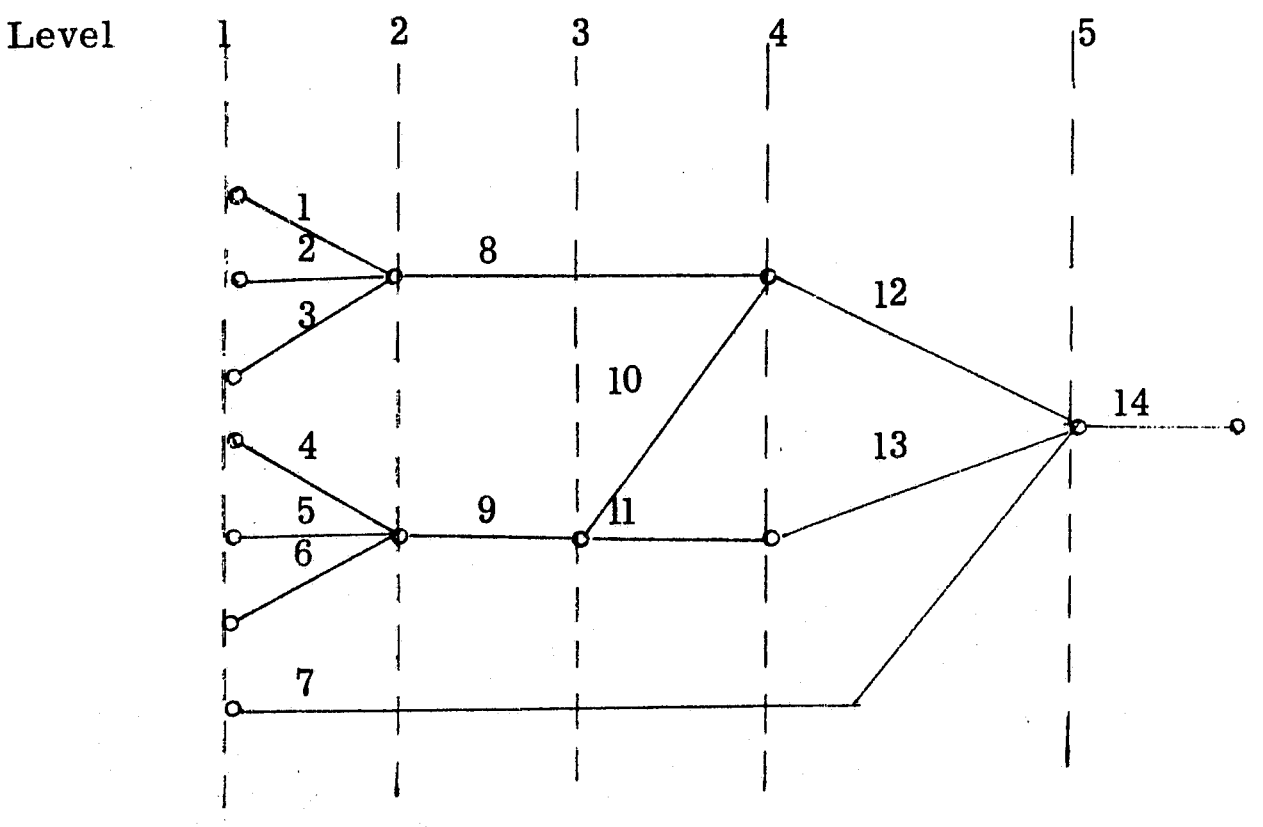


Figure 3.8(c) The (branch) Network Graph of (a)

Let f_i denote the Boolean function realized at node i with respect to the network inputs X and g_i denote the node function at node i. Also let f be the overall Boolean function realized at the network output (assuming a single output network for the purpose of this discussion). In the following, we will formally introduce the concept of "sensitizing" in a network. In general, if f is a Boolean function, we will let $\bar{f}$ denote the complement of f, i.e.,

$$\bar{f}(x) = \overline{f(x)}, \text{ for all } x \in \{0,1\}^{(n)}.$$

Let $f_j^i: X \to \{0,1\}$ be the function realized at node j when g_i is replaced by $\bar{g}_i$.

Definition 3.14

Node j detects node i under x ($x \in X$) if $f_j^i(x) \neq f_j(x)$. This is abbreviated as $j\, d_x\, i$.

Note that every node detects itself since by definition, $f_i^i(x) \neq f_1(x)$.

The notion of one node detecting some other node indicates the potentiality of fault detection by the first node when a fault occurs at the second node. Here, by a "fault" we mean a single fault in a combinational switching network as formally defined in Section 2. Briefly, a (single) fault at node i is the replacement of node function g_i by some other node function g_i' over the same domain, where $g_i \neq g_i'$. When the context is clear, we will simply denote the fault g_i'. A stuck-at 0 (stuck-at 1) fault at node i is denoted as s_i^0 (s_i^1). We formalize the concept of a node detecting a fault in the following

Definition 3.15

Node j detects g_i' under x ($x \in X$) if

i) $j\, d_x\, i$

ii) $g_i'(x_i) \neq g_i(x_i)$, where x_i is the input to node i when x is applied. This is abbreviated as $j\, d_x'\, g_i'$.

Given a network, from the analysis point of view we can only deal with the class of faults which satisfy condition (ii) of Definition 3.15 In the rest of this section we will only consider this class of faults unless explicitly stated.

Observe that if $X_\eta(i,j) = \{x \mid$ node j detects node i under $x\}$ and $X_f(g_i', j) = \{x \mid$ node j detects fault g_i' under x) then $X_f(g_i', j) \subseteq X_\eta(i,j)$. If we call node i [fault g_i'] j-detectable if $\exists x \in X$ node j detects node i [fault g_i'] under x, then the above observation simply says that if fault g_i' is to be j-detectable, node i must be first j-detectable.

Definition 3.16

Node i [fault g_i'] is detectable if $\exists x$ such that an output node detects node i [fault g_i'] under x.

Definition 3.17

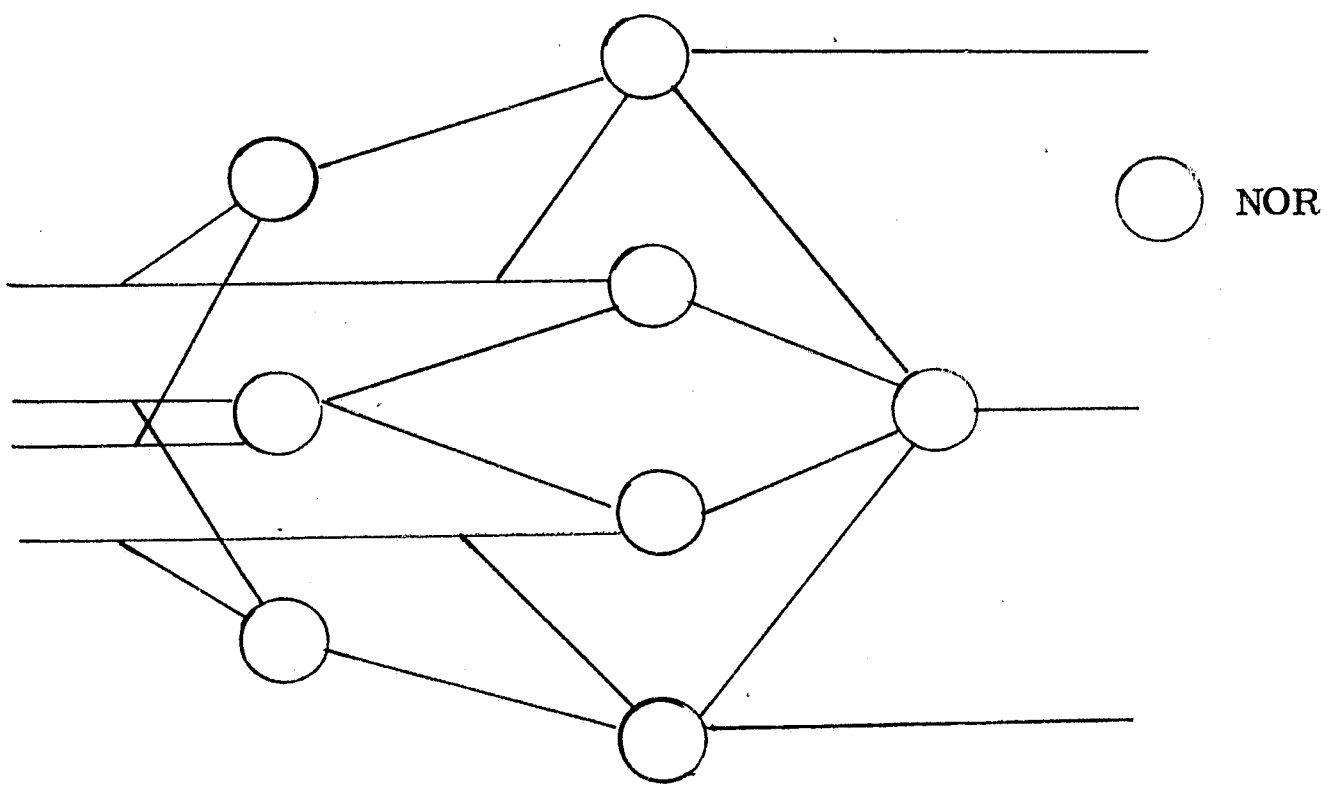
Node i is sensitized under x if an output node detects node i under x. Node i is 0-sensitized [1-sensitized] under x if it is sensitized under x and $f_i(x) = 0$ $[f_i(x) = 1]$.

Definition 3.18

A path $P = i_1, i_2, \ldots, i$ in a combinational network is sensitized under x ($x \in X$) if every node in P detects each preceding node in the path under x. P is 0-sensitized [1-sensitized] under x if it is sensitized under x and $f_{i_1}(x) = 0$ $[f_{i_1}(x) = 1]$.

Example 3.10

Consider the following example of Roth [22].



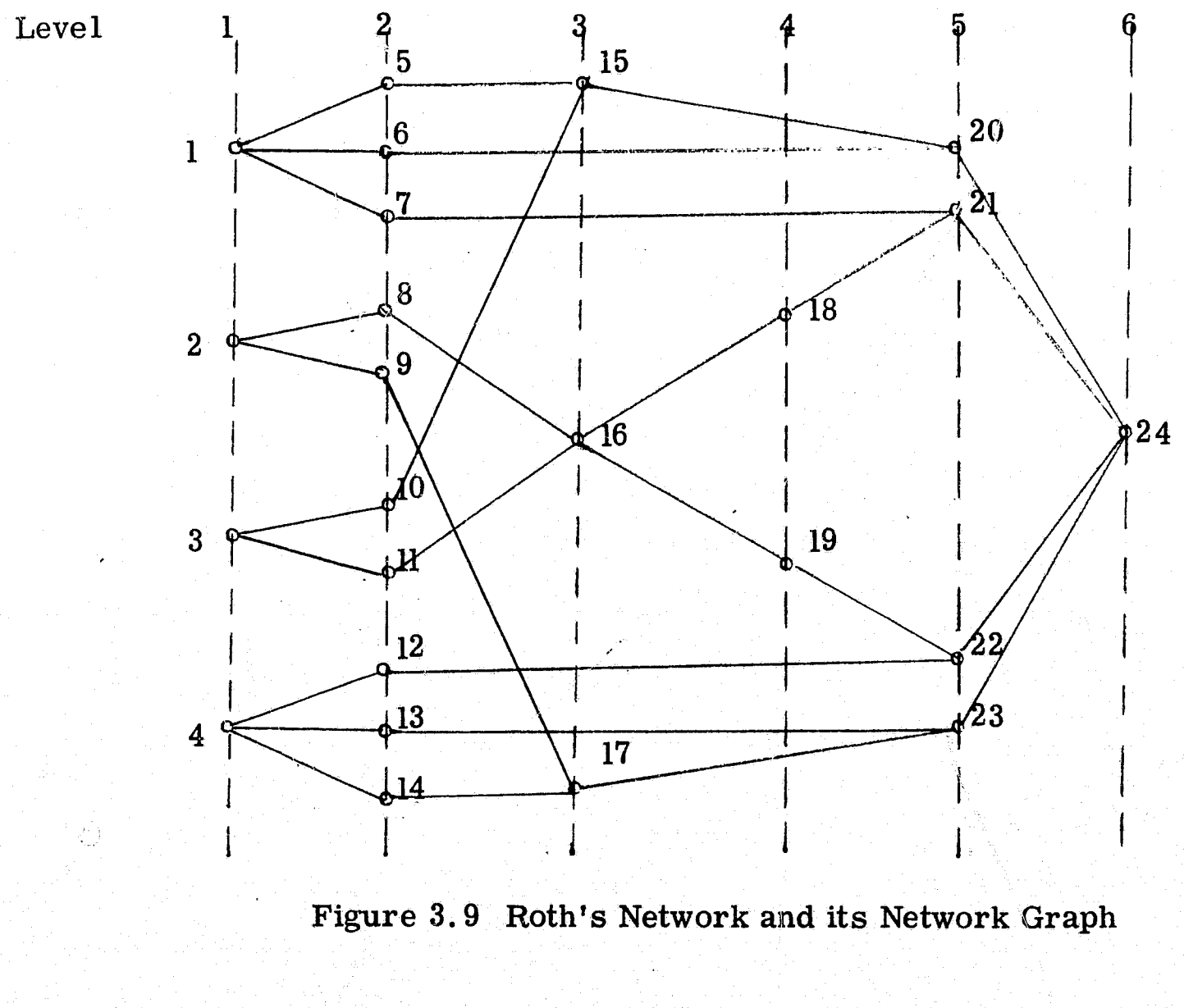


Figure 3.9 Roth's Network and its Network Graph

Node 23 detects node 17 under $x = (0,0,0,0)$. Node 23 detects S_{17}^0 under $x = (0,0,0,0)$. Both node 17 and S_{17}^0 are detectable since output node 24 detects node 17 and S_{17}^0 under $x = (0,0,0,0)$. Node 17 is 1-sensitized under $x = (0,0,0,0)$. The Path $P = 2,9,17,23,24$ is 0-sensitized under $x = (0,0,0,0)$.

With these definitions, we can precisely state a well-known result as the following theorem.

Theorem 3.24

If a path $P = i_1, i_2, \ldots, i_p$ is 0-sensitized under $x \in X$ and 1-sensitized under $y \in X$, then all stuck-at faults $S_{i_j}^0$, $S_{i_j}^1$ $(1 \leq j \leq p)$ are detectable.

Proof

Since by definition of a path, i_p corresponds to the output node and by definition of "path sensitizing" i_p detects every node in P under x and y. Let $g_{i_j}(x_{i_j}) = 0$, then $S_{i_j}^1 \neq g_{i_j}(x_{i_j})$ and by Definition 3.15, the output node detects $S_{i_j}^1$ under x. Similarly, the output node detect $S_{i_j}^0$ under y. Thus all stuck-at faults $S_{i_j}^0$, $S_{i_j}^1$ $(1 \leq j \leq p)$ are detectable by Definition 3.16.

From Definitions 3.17 and 3.18, it is easy to see that if a path P is sensitized under some $x \in X$, then every node in P is also sensitized under x. However, the converse is not necessary true, i.e., there may not be a "sensitized path" passing through a "sensitized node ". One such example was explicitly pointed out by

Schneider [24]. Referring to Figure 3.9 of Example 3.10 Node 16 is 1-sensitized under x = (0, 0, 0, 0) but no path passing through node 16 is sensitized under x = (0, 0, 0, 0). In the following we will give a condition in which the converse also holds. Before we state our results, the following two definitions are needed.

Definition 3.19

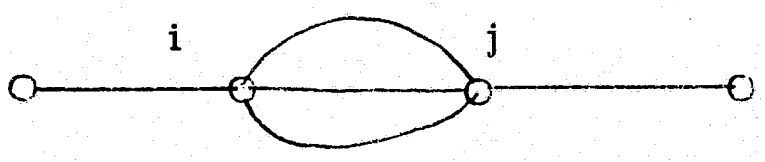
A multipath from node i to node j of a directed graph G is a subgraph of G in which there is at least one path from node i to node j. In the case of a network graph, we always let j be an output node.

Thus, a path is also a multipath. A multipath which is not a path is a strict multipath.

Definition 3.20

A multipath in a network graph is simply connected if there are exactly two nodes in the multipath which have degrees greater than 2.

An example of a simply connected multipath is shown below:



Since we define a path in a network graph to end at a network output, we call a path from node i to node j as a segment or subpath from node i to node j if j is not an output node. Thus a sensitized segment or subpath can be defined similarly.

Definition 3.21

A multipath P is sensitized under x ($x \in X$) if

i) The initial segment before the first fan-out node and the tail segment after the last reconverging node are sensitized subpaths under x.

ii) Every successor node of the first fan-out node detects every node in the initial segment under x.

iii) Each subpath from a fan-out node to the node in the subpath immediately preceding a reconverging node is a sensitized subpath under x.

iv) Each subpath starting from a reconverging node to the node in the subpath immediately preceding another reconverging node is also a sensitized subpath under x.

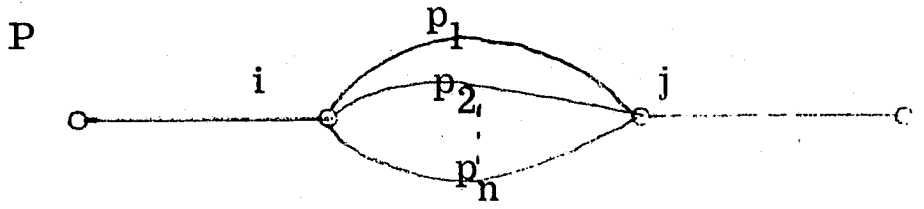
Let us consider combinational networks which use the following gate types: AND, OR, NAND, NOR, and XOR. Then the following theorem gives the result we have expected.

Theorem 3.25

If a node in a combinational network graph can be both 0-sensitized and 1-sensitized and if each sensitized multipath is either a path or simply connected, then there exists a sensitized path passing through the node.

Proof

If the sensitized multipath P is a path then there is nothing to prove. So, let us assume that the P is strict and simply connected.



First, we claim that if node j is an AND gate, then the parity of inversions of subpaths $p_1, p_2, \ldots, p_n$ should be the same. Let $p(p_1)$ represent parity of inversions of path p_1, then $p(p_1) = 0$ if the number of inversions in p_1 is even and $p(p_1) = 1$ otherwise. We claim then $p(p_1) = p(p_2) = \ldots, p(p_n)$. Assume $\exists$ p_k and p_ℓ, $k \neq \ell$. $1 \leq k \leq n$, $1 \leq \ell \leq n$, and $p(p_k) = p(p_\ell)$. Then if g_i is replaced by $\bar{g}_i$, the kth input and ℓth input to node j changes in opposite direction in the sensitized multipath under some x. This means at least one input to node j is 0 and the output of node j remains 0. Thus node j does not detect node i, contrary to our original assumption that P is a sensitized multipath. Thus

$p(p_1) = p(p_2) = \ldots = p(p_n)$. Since node i can be both 0-sensitized and 1-sensitized, and $p(p_1) = \ldots = p(p_n)$, it follows then that $\exists x \in X \ni$ all inputs to node j which are in P are 1's. But this means node j detects every node in each p_i under x. Since node j detects node i under x, there are n single sensitized paths passing through node i. A similar argument can be applied to cases when node j are other gate types except XOR.

If node j is an XOR, then n must be odd. We first show that if n is even, then node j can not detect node i. Let n be even and let k be the number of 0 inputs to node j in P under some $x \in X$. Then there are n-k 1 inputs to node j in P. If k is odd, then there are an odd number of 1 inputs to node j in P. When g_i is replaced by $\overline{g_i}$, k inputs to node j change from 0 to 1 and n-k inputs change from 1 to 0. The parity of 1's to node j remains unchanged and node j does not detect node i under x. It is clear that node j detects every node in each subpath p_i since the output of node i is sensitive to any single input change. Since node j can also detect node i, it follows then that there are n single paths sensitized passing through node i. This completes the proof of the theorem.

As we indicated earlier, in networks containing redundancy, not all faults can be detected by input/output tests. However, if some internal test points are provided, it may be possible to detect some faults which are not otherwise detectable. Our first question is what type of faults can be detected by inserting additional test points?

This question can be answered by the notion of "sensitizing" defined before. First, we need to introduce a new notion. Let T be a subset of the nodes in a combinational network N.

Definition 3.22

Node i [fault g_i'] is T-detectable if $\exists j \in T$, $x \in X \ni$ node j detects node i [fault g_i'].

The set T is to stand for "test points". Let us denote the output node o. Suppose a fault g_i' occurs at node i. g_i' will be detected if o detects g_i' under some $x \in X$. But by Definition , this says that there is some network input x such that i) node o detects node i under x and ii) x produces x_i at node i such that $g_i'(x_i) \neq g_i(x_i)$. If g_i' can not be detected because condition ii) fails, i.e., $\nexists x \in X \ni g_i'(x_i) \neq g_i(x_i)$, then g_i' is called inherently undetectable. If, on the other hand, g_i' can not be detected because node i can not be sensitized, then nevertheless there will exist a node set T such that g_i' is T-detectable for in the worst g_i' is $\{i\}$-detectable. Any T-detectable fault, g_i', can become detectable if nodes in T are "observable".

A node in a network is said to be constant if the function realized at that node is a constant function. The following theorem describes a set of faults which are T-detectable.

Theorem 3.26

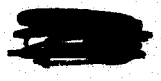
If a combinational network does not have any constant node, then all single node stuck-at faults, S_i^0 and S_i^1, are T-detectable for some set of nodes T in the network graph.

Proof

First, we observe that every node i detects itself under some x since $f_i \neq f_i(x)$ by definition. Since there is no constant node, $\forall$ node i, $\exists\, x, y \in X \quad f_i(x) = 0$ and $f_i(y) = 1$. Then it is clear that node i detects S_i^1 under x and detects S_i^0 under y. Taking T to be the set of nodes in the network, the theorem is proved.

Note that Theorem 3.26 does not apply to node input stuck-at faults since a node may be independent of some of its inputs. However, if every node can be made dependent on each of its input branches then Theorem 3.26 is applicable to all node input and output stuck-at faults.

The classification of faults into inherently undetectable and T-detectable classes is important for the problem of network diagnosis. A significant problem is to find a minimum set of nodes T in a network so that a given set of faults is T-detectable. On the other hand it is not possible to detect inherently undetectable faults without modification of the original network structure. Thus it is expected that the study of the class of inherently undetectable faults will provide some answer to the problem of designing diagnosable networks.



The notion of "path sensitizing" can be generalized to "subgraph sensitizing" where every path is a sensitized path. The sensitized subgraph is useful in dealing with the problem of multiple fault detection later on.

Definition 3.23

A subgraph H of a network graph G is sensitized under x if, for all i, j, which are nodes in H, $\exists$ path from i to j $\Rightarrow$ j detects i under x.

It can be shown that d_x is not in general a transitive relation on the nodes of the network graph. This observation can be easily seen to be equivalent to the fact that there is not necessarily a sensitized path passing through a sensitized node. However, if we restrict d_x to the nodes of a sensitized subgraph, then d_x is a partial ordering. More precisely, if H is a subgraph of G, let N(H) denote the set of nodes in H, then

Theorem 3.27

If H is a sensitized subgraph of a network graph G (under x), then for all $i, j, k \in N(H)$,

1) $j\ d_x\ j$
2) $j\ d_x\ i$ and $i\ d_x\ j \Rightarrow i = j$
3) $j\ d_x\ i$ and $k\ d_x\ j \Rightarrow k\ d_x\ i$.

Proof

1) Already observed.

2) Recall that nodes in an acyclic directed graph can be labelled by integers so that node i is adjacent from nodes of labellings smaller than i. Using this convention, if j d_x i then $j \geq i$. Similarly i d_x j implies $i \geq j$. Therefore i = j.

3) In the sensitized subgraph H, if j d_x i, then by Definition 3.23 there is a path from node i to node j in H. Similarly k d_x j means there is a path from node j to node k. But the reachability in a directed graph is transitive. Thus there is a path from node i to node k and k d_x i.

Thus, it has been shown that d_x is a partial ordering on N(H) if H is a sensitized subgraph. Similar results can be obtained for d'_x although the notion of partial ordering no longer applies. This is stated as the next theorem.

Theorem 3.28

If H is a sensitized subgraph of G under x then, for all $i, j, k \in N(H)$,

1) j d'_x g'_j

2) j d'_x g'_i and i d'_x g'_j $\Rightarrow$ i = j

3) j d'_x g'_i and k d'_x g'_j $\Rightarrow$ k d'_x g'_i

Proof

1) By definition of d'_x, $g'_j(x_j) \neq g_j(x_j)$ and j d_x j, it follows then that j d'_x g'_j.

2) $j\ d'_x\ g'_j \Rightarrow j \geq i$

$i\ d'_x\ g'_i \Rightarrow i \geq j'$

3) $j\ d'_x\ g'_i \Rightarrow g'_i(x_i) \neq g_i(x_i)$ and $j\ d_x\ i$.

$k\ d'_x\ g'_j \Rightarrow g'_j(x_j)$ and $k\ d_x\ j$. Since d_x is transitive on N(H), it follows that $k\ d_x\ i$. Now $k\ d_x\ i$ and $g'_i(x_i) \neq g_i(x_i) \Rightarrow k\ d'_x\ g'_i$.

The problem of multiple fault detection has been considered by many researchers [9, 15] and yet it remained basically unsolved. In the following we will attempt to generalize the notions of single fault detection and sensitized subgraph to study the effect of multiple fault detection in combinational networks.

Definition 3.24

Let $T \subseteq N(G)$, $F_T = \{g'_i \mid i \in T\}$ and $j \in N(G)$. Node j multiply detects fault set F_T under x if

(1) $j\ d'_x\ g'_i\ \forall\ g'_i \in F_T$,

(2) $g_j(x_j^E) \neq g_j(x_j)\ \forall\ E \subseteq F_T$

and

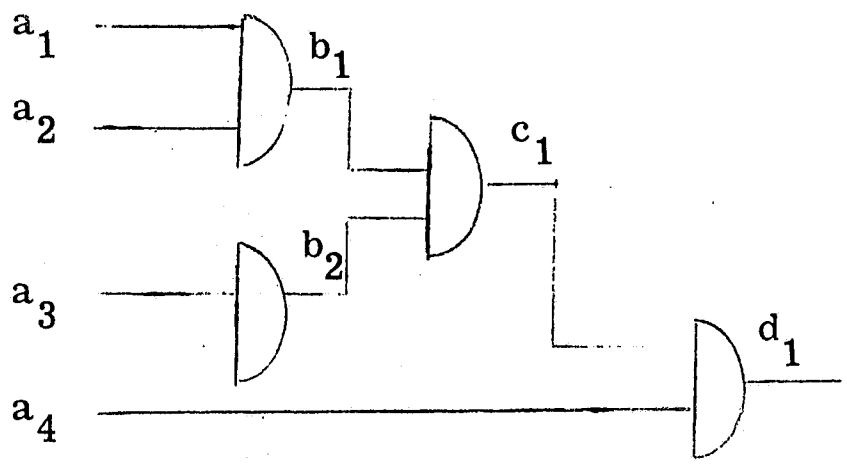
(3) $g'_j(x_j^E) \neq g_j(x_j)\ \forall\ E \subseteq F_T$

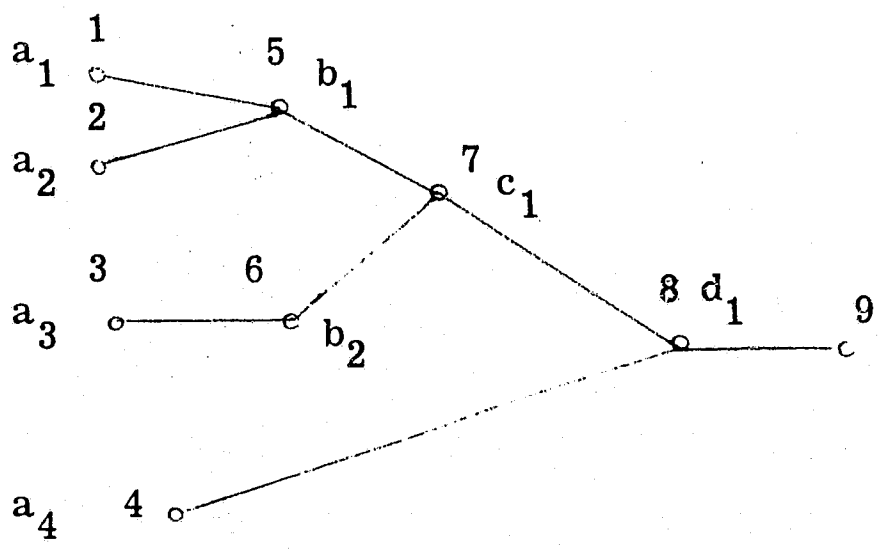
where x_j^E represents input to node j when the fault set E and x is applied to the network.

This notion is sometimes abbreviated as $j\ m\ d'_x\ F_T$.

Example 3.11

Consider the following network and its network graph.





$b_1 = g_5(a_1, a_2) = a_1 a_2$

$b_2 = g_6(a_3) = \bar{a}_3$

$c_1 = g_7(b_1, b_2) = b_1 + b_2$

$d_1 = g_8(c_1, a_4) = c_1 a_4$

Figure 3.10

$T = \{5, 6, 8\}$

$g_5'(a_1, a_2) = a_1$ $\qquad F_T = \{g_5', g_6', g_8'\}$

$g_6'(a_3) = a_3$

$g_8'(c_1, a_4) = a_4$

Consider the input assignment $\alpha(a_1, a_2, a_3, a_4) = (1, 0, 1, 1) = x$.

(1) Node 8 detects $F_T = \{g_5', g_6', g_8'\}$ under $x = (1, 0, 1, 1)$

(2) $x_8 = (c_1, a_4) = (0, 1)$

$$x_8^{\{g_5'\}} = (1, 1) = x_8^{\{g_6'\}} = x_8^{\{g_5', g_6'\}}$$

$$\therefore\ g_8(x_8^{\{g_5'\}}) = g_8(x_8^{\{g_6'\}}) = g_8(x_8^{\{g_5', g_6'\}}) = 1 \neq g_8(x_8) = 0$$

(3) $$g_8'(x_8^{\{g_5'\}}) = g_8'(x_8^{\{g_6'\}}) = g'_8(x_8^{\{g_5', g_8'\}})$$

$$= 1$$

$$\neq g_8(x_8) = 0$$

Thus node 8 multiply detects $F_T = \{g', g_6', g_8'\}$ under $x = (1, 0, 1, 1)$.

Theorem 3.29

If $P = i_1, i_2, \ldots, i_p$ is a sensitized path under x in G, then the set of stuck-at faults in P, $F_{s_x} = \{s_{i_j}^{k_j} | k_j = 0 \text{ or } 1,\ 1 \leq j \leq p\}$ detected under x is multiply detected by node i_p under x.

Proof

Suppose $\exists\ E = \{s_{i_j}^{k_j}, s_{i_\ell}^{k_\ell}, s_{i_m}^{k_m}, \ldots, s_{i_n}^{k_n}\} \subseteq F_{s_x}$ such that

$$g_{i_p}(x_{i_p}^{E}) = g_{i_p}(x_{i_p})$$

Assuming $i_j < i_\ell < i_m < \cdots < i_n$, then $\forall\ i,\ i_j < i < i_\ell$

$$g_i(x_i^{\{s_{i_j}^{k_j}\}}) \neq g_i(x_i)$$

Now, since $s_{i_\ell}^{k_\ell}$ is detected by i_p under x,

$$s_{i_\ell}^{k_\ell} = k_\ell = g_{i_\ell}(x_{i_\ell}^{\{s_{i_j}^{k_j}\}}) \neq g_{i_\ell}(x_{i_\ell}).$$

We claim that $\forall$ i, $i_\ell < i < i_m$

$$g_i(x_i^{\{s_{i_j}^{k_j}, s_{i_\ell}^{k_\ell}\}}) \neq g_i(x_i)$$

Suppose this is not true, then $\exists$ i such that

$$g_i(x_i^{\{s_{i_j}^{k_j}, s_{i_\ell}^{k_\ell}\}}) = g_i(x_i)$$

Choose the least such i, call it i_f. Then $g_{i_f}(x_{i_f}^{\{s_{i_j}^{k_j}, s_{i_\ell}^{k_\ell}\}}) = g_{i_f}(x_{i_f})$

and yet

$$g_{i_{f-1}}(x_{i_{f-1}}^{\{s_{i_j}^{k_j}, s_{i_\ell}^{k_\ell}\}}) \neq g_{i_{f-1}}(x_{i_{f-1}})$$

Then from i_j to i_{f-1}, all node output values differ from their fault free values yet i_f remains unchanged. This says that i_f does not detect $s_{i_j}^{k_j}$, contrary to the assumption that P is a sensitized path by Theorem 3.28. This process can be iterated and a contradiction is obtained for the assumption that $g_{i_p}(x_{i_p}^{E}) = g_{i_p}(x_{i_p})$. Thus i_p multiplely detects F_{s_x} under x.

The above theorem says that in a sensitized path, any simultaneous occurrence of singly stuck-at faults is also multiply detected. This is clearly in line with the common intuition that the last fault in a sensitized path dominates all faults that occur before it.

Theorem 3.29 can be generalized to a sensitized subgraph when the

network graph is a tree. First let us consider a single gate whose gate input is given. The gate output is said to be sensitive to the jth input under the given gate input if when the jth input is complemented, the gate output will also be complemented. Stated more precisely, if g_i is the gate function and $x_i = (a_1, a_2, \ldots, a_r)$ is a given gate input, then gate i is sensitive to the jth input under x_i if

$$g_i(a_1, a_2, \ldots, a_j, a_r) \neq g_i(a_1, a_2, \ldots, \bar{a}_j, \ldots, a_r).$$

A gate function which is restricted to AND, OR, NAND or NOR types function is called a restricted gate function. This class of functions has the following interesting properties.

Lemma 3.5

Let g_i be a restricted gate function. If the output of gate i is sensitive to each of its inputs under some gate input x_i, then it is sensitive to any combination of inputs under x_i.

Proof

Let $x_i = (a_1, a_2, \ldots, a_r)$. If gate i is an AND or NAND gate, then $a_i = 1 \; \forall \; 1 \leq i \leq r$ and for any x_i' such that $x_i' \neq x_i$, we have $g_i(x_i) \neq g_i(x_i')$. Similarly if gate i is an OR or NOR gate, then $a_i = 0 \; \forall \; 1 \leq i \leq r$ and we obtain the same conclusion.

We can proceed now to consider the case where more than a single path is involved. Let G be a combinational tree type network graph

whose gate functions are restricted. The following theorem gives a set of faults which is multiply detected by a node.

Theorem 3.30

Let G be a tree type combinational network graph and H_x be a sensitized subgraph of G under x with a root i_0. Then i_0 multiply detects the set of stuck-at faults F_{H_x} which is detected by i_0 under x.

Proof

We divide the proof into two cases:

Case 1: By Theorem 3.29 if $\forall E \subseteq F_{H_x}$, the faulty nodes in E belong to the same path, then E is multiply detected by i_0 under x.

Case 2: If the faulty nodes in F belong to different branches in H_x, then there exist $i \in N(H_x)$ with in-degree of i greater than 1 and an $E' \subseteq E$ such that the faulty nodes in E' are predecessors of i. Choose the least such i, i_m, then by Lemma 3.5 $g_{i_m}(x_{i_m}^{E'}) \neq g_{i_m}(x_{i_m})$. This argument is then iterated on any such i with in-degree greater than 1 until $E' = E$. Thus $\forall E \leq F_{H_x}$, E is multiply detected by i_0 under x and F_{H_x} is multiply detected by i_0 under x.

REFERENCES

[1] Armstrong, D. B., "On Finding a Nearly Minimal Set of Fault Detection Tests for Combinational Logic Nets", IEEE Transactions on Electronic Computers, v EC-15, 1 (February 1966) 66-73.

[2] Booth, T. L., Sequential Machines and Automata Theory, John Wiley and Sons, Inc., New York, New York (1967) 597.

[3] Clifford, A. H. and G. B. Preston, The Algebraic Theory of Semigroups, vol. I., American Mathematical Society, Providence, Rhode Island (1961).

[4] Cohn, M. and S. Even, "Identification and Minimization of Linear Automata", Report No. SRRC-RR-64-60, Sperry Rand Research Center (August 1964).

[5] Connolly, John B., and W. G. Schmidt, "Failure Erasure Circuitry: A Duplicative Technique of Failure Masking Systems", IEEE Transactions on Electronic Computers, v EC-16, 1 (February 1967) 82-85.

[6] Dauber, P. S., "An Analysis of Errors in Finite Automata", Information and Control, v 8, (June 1965) 295-303.

[7] Gill, A., Introduction to the Theory of Finite-State Machines, McGraw-Hill Book Company, New York, New York (1962).

[8] Golomb, S. W., L. D. Baument, M. F. Easterling, J. J. Stiffler and A. J. Viterbi, Digital Communications, Prentice-Hall, Inc., Englewood Cliffs, New Jersey (1964) 210.

[9] Hannigan, J. M. and C. G. Masters, "Redundant System Test Point Allocation and Mission Reliability Estimating Procedures", IEEE Transactions on Electronic Computers, v EC-16, 5 (October 1967) 591-596.

[10] Harrison, M. A., "An Analysis of Errors in Finite Automata", Information and Control, v 8, (August 1965) 430-450.

[11] Hartmanis, J., and R. E. Stearns, Algebraic Structure Theory of Sequential Machines, Prentice-Hall, Inc., Englewood Cliffs, New Jersey (1966) 211.

[12] Hartmanis, J. and R. E. Stearns, "A Study of Feedback and Errors in Sequential Machines", IEEE Transactions on Electronic Computers, v EC-12, (June 1963) 223-232.

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

[14] Jephson, J. S., P. P. McQuarrie, and R. E. Vogelsberg, "A Three-Value Computer Design Verification System", IBM System Journal, v 8, 3 (June 1969) 178-188.

[15] Kautz, W. H., "Fault Testing and Diagnosis in Combinational Digital Circuits", IEEE Transactions on Computers, v C-17, 4 (April 1968) 352-366.

[16] 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.

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

[18] Moore, E. F., "Gedanken Experiments on Sequential Machines", Automata Studies, 34 (1956) 129-153.

[19] Neumann, P. E., "Error Limiting Coding Using Information Lossless Machines", IEEE Transactions on Information Theory, v IT-10, (April 1964) 108-115.

[20] Peterson, W. W., Error-Correcting Codes, John Wiley and Sons and MIT Press, New York, New York (1961) 283.

[21] Roth, J. P., "Algebraic Topological Methods for the Synthesis of Switching Systems, I", Transactions of the American Mathematical Society, v 88, 2 (July 1958) 301-326.

[22] Roth, J. P., W. G. Bouricius and P.R. Schneider, "Programmed Algorithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits", IEEE Transactions on Electronic Computers, v EC-16, 5 (October 1967) 567-580.

[23] Russo, R. L., "The Synthesis of Error-Tolerant Counters Using Minimum Distance Three State Assignments", IEEE Transactions on Electronic Computers, v EC-14, (June 1965) 359-366.

[24] Schneider, P. R., "On the Necessity to Examine D-Chains in Diagnostic Test Generation - An Example", IBM Journal of Research and Development, v 11, 1 (January 1967) 114.

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

[26] Varsharmov, R. R., "Estimate of the Number of Signals in Error Correcting Codes", Dokl., Akad. Nauk., 5 (1957) 739-741.

[27] von Neumann, J., "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components", Annals of Mathematical Studies, Princeton University Press, Princeton New Jersey, 34 (1956) 43-98.

[28] Widder, D. V., Advanced Calculus, Prentice Hall, Inc., Englewood Cliffs, New Jersey (1961) 520.

[29] Winograd, S., "Input Error Limiting Automata", Journal of the Association for Computing Machinery, v 11, (July 1964) 338-351.

[30] Yoeli, M., and G. Rosenfeld, "Logical Design of Ternary Switching Circuits", IEEE Transactions on Electronic Computers, v EC-14, 1 (February 1965) 19-29.