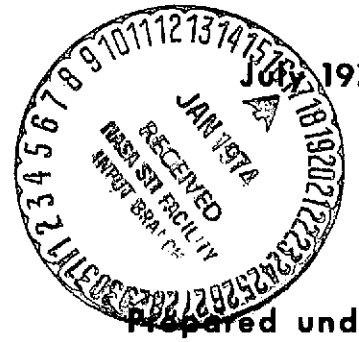
010261-7-T

On-Line Diagnosis of Sequential Systems

R. J. SUNDSTROM

under the direction of
Professor J. F. Meyer



January 1973

Prepared under
NASA Grant NGR23-005-463

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

THE UNIVERSITY OF MICHIGAN

SYSTEMS ENGINEERING LABORATORY

Department of Electrical and Computer Engineering
College of Engineering

SEL Technical Report No. 72

ON-LINE DIAGNOSIS OF SEQUENTIAL SYSTEMS

by

Robert J. Sundstrom

Under the direction of
Professor John F. Meyer

July 1973

Prepared under

NASA Grant
NGR23-005-463

Table of Contents

	Page
1. Introduction	1
2. Discrete-time Systems	6
3. Realizations	14
4. Resettable Systems with Faults	24
5. Fault Tolerance and Errors	36
6. On-line Diagnosis	48
7. Preliminary Results	56
8. Possibilities for Further Investigation	64
References	68

1. Introduction

For many applications, especially those in which a computer is controlling a real-time process (e.g. telephone switching, flight control of an aircraft or spacecraft, control of traffic in a transportation system, etc.), it is desirable to continuously monitor the performance of the system, as it is being used, to determine whether its actual behavior is tolerably close to the intended behavior. It is this sort of monitoring which we mean by the term "on-line diagnosis." Implementation of on-line diagnosis may be external to the system, both internal and external, or completely external. In the last extreme, on-line diagnosis is sometimes referred to as "self-diagnosis" or "self-checking" ([1], [2]).

On-line diagnosis plays a very important role in almost every ultra-reliable computer system which has ever been proposed (see [2], [3], or [4] for example), and a lesser but still important role in many conventional systems. For example, the IBM System/360 utilizes checking circuits to detect errors [5]. The signals generated by these circuits are used in some models to freeze the computer so that the instruction which was currently executing may be retried if possible, and to assist in the checkout and repair of the computer if the automatic retry attempt fails. Ultra-reliable computers typically use the signals generated by the monitoring device to provide the computer system with the information it needs to automatically reconfigure itself so as to avoid using any faulty circuits. One other use for such

signals is to simply inform the system user that the system is not operating properly and that there may be errors in his data.

In general, on-line diagnosis is used to verify that the system is operating properly; or conversely, to signal that it is in need of repair. In most computer systems this task is also performed in some part by off-line diagnosis. By off-line diagnosis we are referring to the process of removing the system from its normal operation and applying a series of prearranged tests to determine whether any faults are present in the system. There are major differences between on-line and off-line diagnosis and it is important to be aware of the capabilities and the limitations of each.

One basic difference is that on-line diagnosis is a continuous process whereas off-line diagnosis has a periodic nature. Due to this only permanent faults can be diagnosed with off-line diagnosis because if a fault is transient in nature it may not be in the system when it is tested. On the other hand, since on-line diagnosis is a continuous monitoring process both permanent and transient faults can be diagnosed. Also, with off-line diagnosis the system must be removed from its normal operation to apply the tests and this may not be acceptable in a real-time application.

The cost of either form of diagnosis depends on the nature of the system to be diagnosed, the technology to be used in building the system, and the degree of protection against faulty operation that is required. With on-line diagnosis the cost is almost totally in the

design and construction of extra hardware. With off-line diagnosis the cost is in the initial generation of the tests and in the subsequent storage and running of these tests.

In general, off-line diagnosis is useful for factory testing and for applications where immediate knowledge of any faulty behavior is not essential. Off-line diagnosis is also useful for locating the source of trouble once such trouble is indicated by on-line diagnosis. For example, Bell System's No. 1 ESS[4] uses duplicate processors to continually check one another and once a discrepancy is detected off-line diagnosis is used to determine which processer exhibited the erroneous behavior and to locate the faulty module in that processer.

In the MARCS study [2] a more integrated use of on-line diagnosis is proposed whereby a number of checking circuits observe the performance of various parts of the computer. With a scheme such as this information about the location of a fault can be obtained from knowledge of which checking circuit indicated the trouble.

Both forms of diagnosis have been used to check the operation of computers from the very first machines until the present time. In a short paper published in 1957 Eckert [6] informs us that off-line diagnosis was relied upon for the ENIAC computer, that the BINAC system had duplicate processors, and the the UNIVAC used a more economical on-line diagnosis scheme involving 35 checking circuits. During the past decade, however, the development of theory and

techniques for fault diagnosis in digital systems and circuits have focused mainly on problems of off-line diagnosis (see [1] and [7] for example).

The work that has been done on on-line diagnosis is mainly in the area of techniques. One early paper is Kantz's study [8] of fault detection techniques for combinational circuits. In this paper he investigated a number of techniques including the use of codes and the possibility of greater economy if immediate detection of errors was not necessary. Many of the more common on-line diagnosis techniques have been gathered together and published in a book by Sellers, Hsiao, and Beardson [9]. Much of what is in this book and a large portion of the techniques that can be found elsewhere in the literature are concerned with special circuits such as adders and counters. For example, see the papers by Avizienis [10], Rao [11], and Dorr [12].

Relative little work can be found on the theory of on-line diagnosis. In one of the earliest works of a theoretical nature Peterson [13] showed that on adder can be checked using a completely independent circuit which adds the residue, module some base, of the operands. He went on to show that any independent check of this type was a residue class check. Another interesting theoretical result was published by Peterson and Rabin [14]. They showed that combinational circuits can differ greatly in their inherent diagnosability and that in some cases virtual duplication is necessary. A later and

more general paper is that of Carter and Schneider [15]. They propose a model for on-line diagnosis which involves a system and an external checker. To be on-line diagnosable the system must produce non-code outputs when it fails and the external checker must signal the occurrence of such an output. The checking circuits that they consider indicate the presence of faults in the checkers themselves in addition to faults in the systems they are monitoring.

With decreasing cost of logic and the increasing use of computers in real-time applications where erroneous operation can result in the loss of human life and/or large sums of money the use of on-line diagnosis can be expected to increase greatly in the near future. The importance of this area along with the relative lack of theoretical research is our motivation for initiating this study of on-line diagnosis.

2. Discrete-Time Systems

On-line diagnosis is inherently a more complex process than off-line diagnosis because of two complicating factors: i) it has to deal with input over which it has no control and ii) faults can occur as the system is being diagnosed. We would like to build a theory of on-line diagnosis using conventional models of time-invariant (stationary, fixed) systems (e.g. sequential machines, sequential networks etc.). However, due to the second factor mentioned above these conventional models can no longer be used to represent the dynamics of the system as it is being diagnosed. A system which is designed and built to behave in a time-invariant manner becomes a time-varying system as faults occur while it is in use. Therefore, a more general representation based on time-varying systems is required. Based on this fundamental observation we have developed what we believe to be an appropriate model for the study of on-line diagnosis.

Definition 1

Relative to the time-base $T = \{\ldots, -1, 0, 1, \ldots\}$, a discrete-time system (with finite input and output alphabets) is a system

$$S = (I, Q, Z, \delta, \lambda)$$

where I is a finite set, the input alphabet

Q is a set, the state set

Z is a finite set, the output alphabet

δ: $Q \times I \times T \rightarrow Q$, the transition function

λ: $Q \times I \times T \rightarrow Z$, the output function.

The interpretation of a discrete-time system is a system which, if at time t is in state q and receives input a, will at time t emit output symbol $\lambda(q, a, t)$ and at time $t + 1$ be in state $\delta(q, a, t)$. In the special case where the functions δ and λ are independent of time (i.e., are time-invariant), the definition reduces to that of a (Mealy) sequential machine. In the discussion that follows we will assume, unless otherwise qualified, that S is finite-state (i.e., $|Q| < \infty$).

To describe the behavior of a system, we first extend the transition and output functions to input sequences in the following natural way. If I^* is the set of all finite length sequences over I (including the null sequence Λ) then:

$$\bar{\delta}: Q \times I^* \times T \rightarrow Q$$

where, for all $q \in Q$, $a \in I$, $t \in T$:

$$\bar{\delta}(q, \Lambda, t) = q$$

$$\bar{\delta}(q, a, t) = \delta(q, a, t)$$

$$\bar{\delta}(q, a_1 a_2 \ldots a_n, t) = \delta(\bar{\delta}(q, a_1 a_2 \ldots a_{n-1}, t), a_n, t + n-1).$$

Similarly, if $I^+ = I^* - \{\Lambda\}$:

$$\bar{\lambda}: Q \times I^+ \times T \rightarrow Z$$

where, for all $q \in Q$, $a \in I$, $t \in T$:

$$\bar{\lambda}(q, a, t) = \lambda(q, a, t)$$

$$\bar{\lambda}(q, a_1 a_2 \ldots a_n, t) = \lambda(\bar{\delta}(q, a_1 a_2 \ldots a_{n-1}, t), a_n, t + n-1).$$

Relative to these extended functions, the behavior of S in state q is the function

$$\beta_q: I^+ \times T \rightarrow Z$$

where $$\beta_q(x, t) = \bar{\lambda}(q, x, t).$$

Thus, if the state of the system is q and it receives input sequence x starting at time t, then $\beta_q(x, t)$ is the output emitted when the last symbol in x is received (i.e. the output at time $t + |x| - 1$ ($|x|$ = length (x))).

Many investigations of on-line diagnosis and fault tolerance have studied redundancy schemes such as duplication and triplication. Typically they have not dealt with the problem of starting each copy of a machine in the same state. In this study we will be examining these schemes and others for which the same problem arises. Since many existing systems have reset capabilities, and since this feature solves the above synchronizing problem we will use a special type of system for which the reset capabilities are explicitly specified. This explicit specification of the reset capability is essential since it is an important part of the total system and is just as subject to faults as any other portion of the system.

Definition 2

A resettable discrete-time system (resettable system) is a system

$$S = (I, Q, Z, \delta, \lambda, R, \rho)$$

where $(I, Q, Z, \delta, \lambda)$ is a discrete-time system

R is a finite nonempty set, the reset alphabet

ρ: $R \times T \rightarrow Q$, the reset function.

A resettable system is resettable in the sense that if reset r is applied at time t - 1 then $\rho(r, t)$ is the state at time t. This method of specifying reset capability is a matter of convenience. This feature could just as well have been incorporated as a restriction on the transition function relative to a distinguished subset of input symbols called the reset alphabet. Thus a resettable discrete-time system can indeed be regarded as a special type of discrete-time system. If δ, λ, and ρ are all independent of time the definition reduces to that of a resettable sequential machine. Thus a resettable machine can be viewed as a resettable system which is invariant under time-translations.

Given a resettable system we can view it as a system organized as in Figure 1.

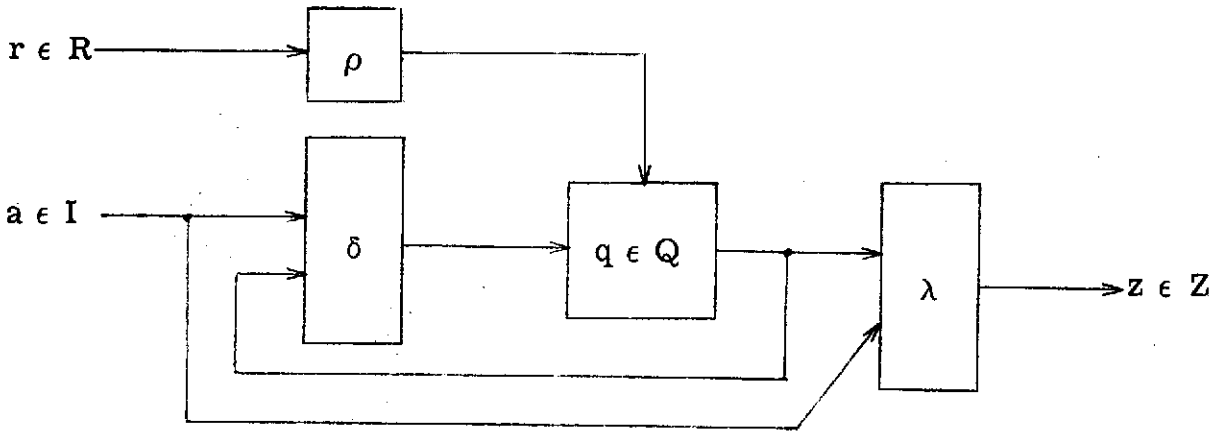


Figure 1 Schematic Diagram for S = (I, Q, Z, δ, λ, R, p)

We will represent sequential machines in the usual manner, i.e. via transition tables or state graphs. Resettable machines are represented by minor extensions of these two methods. The transition table of a resettable machine is identical to that of a machine with the addition of one column on the right to accommodate the reset function. If $\rho(r) = q$ then r will appear in the last column of the q row. Similarly, the state graph of a resettable machine is identical to that of a machine with the addition of one short arrow for each $r \in R$. This arrow will be labeled r and will point to state $\rho(r)$.

Example 1

Let M_1 be the sequence generator with reset alphabet $\{0\}$ and input alphabet $\{1\}$ which has been implemented by the circuit in Figure 2.

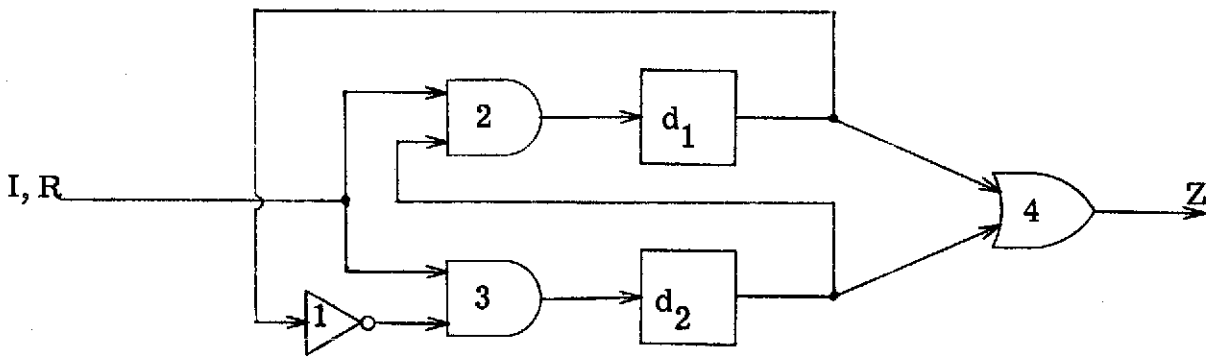


Figure 2 Circuit for M_1

Then the transition table and the state graph for M_1 are as shown in Figures 3 and 4.

Q \ I	1	R
00	01/0	0
01	11/1	
10	00/1	
11	10/1	

Figure 3 Transition Table for M_1

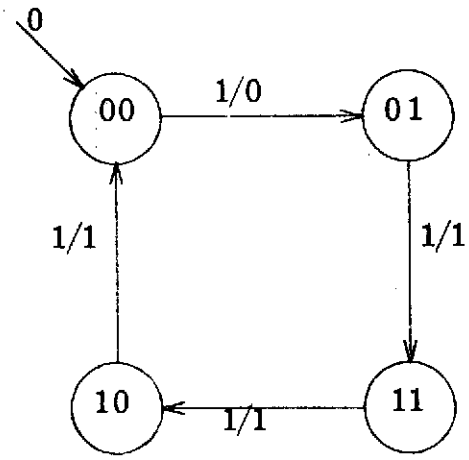


Figure 4 State Graph for M_1

The circuit in Figure 2 is also an implementation of a similar machine M_2 with input alphabet $\{0, 1\}$. The state graph for M_2 is shown in Figure 5.

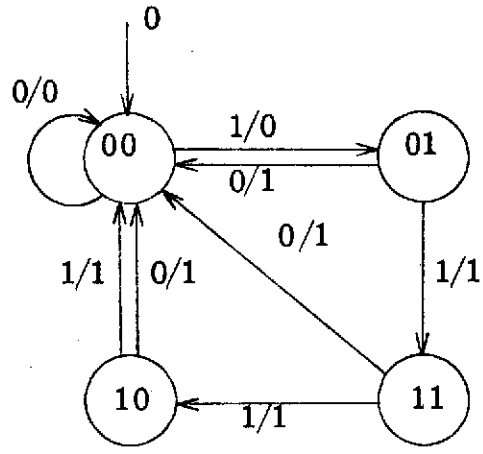


Figure 5 State Graph for M_2

Thus, in M_2 the input symbol "0" can be intrepreted as a regular input or as a reset input. In M_2 the outputs for input 0 are explicitely specified whereas in M_1 they may be regarded as classical "don't cares."

In general, we have no convenient representations for discrete-time systems and resettable systems. About all we can do is specify each of the functions δ, λ, and ρ explicitly. However, most of the systems that we will deal with will be truly time-varying at only a few points in time and thus can be described by the machines they resemble in the intervals between these points.

Example 2

Suppose that M_1 was implemented as in Figure 2 and that this circuit operated perfectly up to time 100 when gate 2 became stuck-at-0. What actually existed was not a resettable machine but a (time-varying) resettable system S which looks like M_1 up to time 100 and like a different machine, say M_1', thereafter. The graph for M_1' is shown in Figure 6.

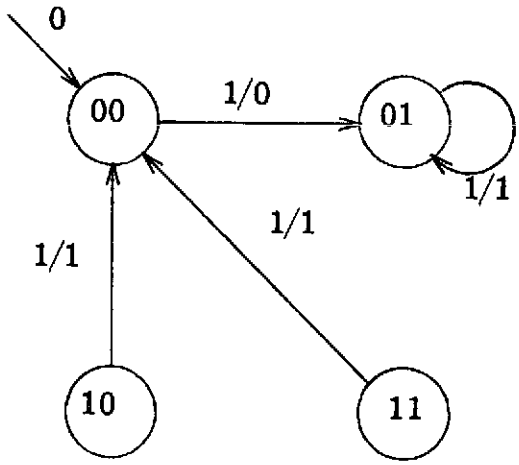


Figure 6 Resettable Machine M_1'

We can represent S as follows:

$$S = \begin{cases} M_1 & \text{for } t < 100 \\ M_1' & \text{for } t \geq 100. \end{cases}$$

By this we mean that $I = I_1 = I_1'$ and likewise for Q, Z, and R, and that

$$\delta(q, a, t) = \begin{cases} \delta_1(q, a) & \text{for } t < 100 \\ \delta_1'(q, a) & \text{for } t \geq 100 \end{cases}$$

and similarly for λ and ρ.

For resettable systems we take the definitions of $\bar{\delta}$, $\bar{\lambda}$, and β_q to be the same as those for systems. It is also convenient in the case of resettable systems to specify behavior relative to a reset input r that is released at time t, that is, the behavior of S for condition (r, t) ($r \in R$, $t \in T$) is the function

$$\beta_{r,t}: I^+ \longrightarrow Z$$

where

$$\beta_{r,t}(x) = \beta_{\rho(r,t)}(x, t).$$

If $t = 0$, $\beta_{r,0}$ is referred to as the behavior of S for initial reset r and is denoted simply as β_r.

3. Realizations

Discrete-time systems are a straightforward generalization of sequential machines and many notions that we are familiar with in the context of sequential machines can be generalized in a similar manner to apply to discrete-time systems. In this section we will look in some detail at the generalized notion of a realization. As in other sections, our emphasis here will be toward those aspects of the theory that will be useful to us in our study of on-line diagnosis. We begin by stating Meyer and Zeigler's definition of realization for sequential machines [16].

Definition 3

If M and $\tilde{M}$ are sequential machines then M realizes $\tilde{M}$ (written $M_\rho \tilde{M}$) if there is a triple of functions $(\sigma_1, \sigma_2, \sigma_3)$ where $\sigma_1: (\tilde{I})^+ \rightarrow I^+$ is a semigroup homomorphism such that $\sigma_1(\tilde{I}) \subseteq I$, $\sigma_2: \tilde{Q} \rightarrow Q$, $\sigma_3: Z' \rightarrow \tilde{Z}$ where $Z' \subseteq Z$, such that for all $\tilde{q} \in \tilde{Q}$ and all $x \in (\tilde{I})^+$, $\tilde{\beta}_{\tilde{q}}(x) = \sigma_3(\beta_{\sigma_2(\tilde{q})}(\sigma_1(x)))$.

It has been shown by Leake [17] that this strictly behavioral definition of realization is equivalent to the structurally oriented definition of Hartmanis and Stearns [18].

The following definition extends the above notion in a natural manner to include discrete-time systems.

Definition 4

If S and $\tilde{S}$ are two discrete-time systems then S realizes $\tilde{S}$ ($S\rho\tilde{S}$) if there is a triple of functions $(\sigma_1, \sigma_2, \sigma_3)$ where $\sigma_1: (\tilde{I})^+ \rightarrow I^+$ is a

semigroup homomorphism such that $\sigma_1(\tilde{I}) \subseteq I$, σ_2: $\tilde{Q} \rightarrow Q$, σ_3: $Z' \rightarrow \tilde{Z}$ where $Z' \subseteq Z$, such that for all $\tilde{q} \in \tilde{Q}$, for all $t \in T$, and for all $x \in (\tilde{I})^+$, $\tilde{\beta}_{\tilde{q}}(x,t) = \sigma_3(\beta_{\sigma_2(\tilde{q})}(\sigma_1(x),t))$.

If S and $\tilde{S}$ are resettable systems our definition of realization is somewhat different. Inherent in this definition is our presupposition that a resettable system will be reset before every use.

Definition 5

If S and $\tilde{S}$ are two resettable systems then S realizes $\tilde{S}$ ($S\rho\tilde{S}$) if there is a triple of functions $(\sigma_1, \sigma_2, \sigma_3)$ where σ_1: $(\tilde{I})^+ \rightarrow I^+$ is a semigroup homomorphism such that $\sigma_1(\tilde{I}) \subseteq I$, σ_2: $\tilde{R} \rightarrow R$, σ_3: $Z' \rightarrow \tilde{Z}$ where $Z' \subseteq Z$, such that for all $\tilde{r} \in \tilde{R}$, for all $t \in T$, and for all $x \in (\tilde{I})^+$, $\tilde{\beta}_{\tilde{r},t}(x) = \sigma_3(\beta_{\sigma_2(\tilde{r}),t}(\sigma_1(x)))$.

In the case where S and $\tilde{S}$ are time-invariant resettable systems (i.e., resettable machines) all mention of time can be deleted from the above definition.

Thus for each $\tilde{r} \in \tilde{R}$ and $t \in T$ the behavior of S for condition $(\sigma_2(\tilde{r}), t)$ is the same (modulo input encoding and output decoding) as the behavior of $\tilde{S}$ for condition $(\tilde{r}, t)$.

Example 3

Let $\tilde{M}_3$ and M_3 be the resettable machines shown in Figures 7 and 8.

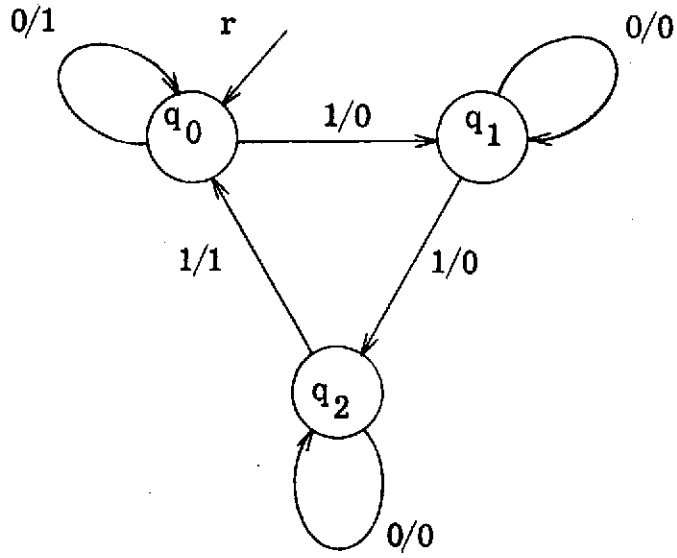


Figure 7 Resettable Machine $\tilde{M}_3$

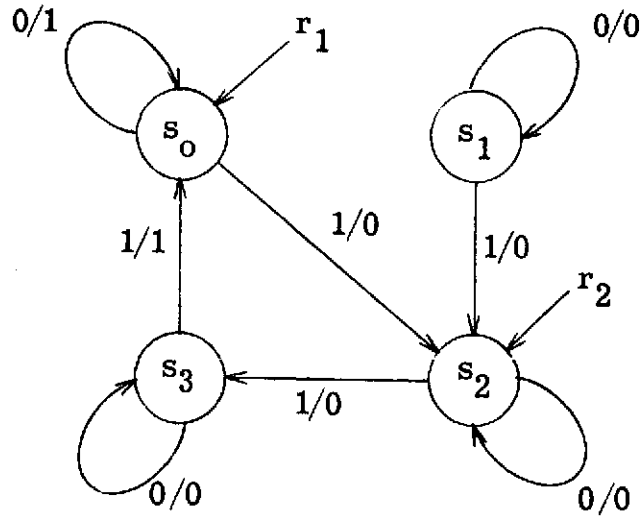


Figure 8 Resettable Machine M_3

Then $M_3\rho\tilde{M}_3$ under the triple $(\sigma_1, \sigma_2, \sigma_3)$ where $\sigma_1:(\tilde{I}_3)^+ \longrightarrow I_3^+$ is the identity, σ_2: $\tilde{R}_3 \longrightarrow R_3$ is defined by $\sigma_2(r) = r_1$, and σ_3: $Z_3 \longrightarrow \tilde{Z}_3$ is the identity. To verify this claim we need only observe that $\tilde{\beta}^3_r(x) = \beta^3_{r_1}(x)$ for all $x \in (\tilde{I}_3)^+$.

Notice that the definition of realizes for resettable systems is less restrictive than that for discrete-time systems in the sense that where they are both resettable we only require the realizing system to mimic the behavior of the reset states of the realized system; while in the case where they are not resettable the realizing system must mimic the behavior of every state of the realized system. On the other hand, the definition in the resettable case is more restrictive in the sense that for each reset state in the realized system not only does there exist a state in the realizing system which mimics its behavior, but we also know how to get to that state.

For the special case of time-invariant resettable systems (i.e., resettable machines) the above remarks will be made more precise in the following result which is analogous to the result due to Leake that we have cited earlier. Let M be a resettable machine. The reachable part of M is the set

$$\{p \in Q \mid p = \delta(\rho(r), x) \text{ for some } r \in R,\ x \in I^*\}.$$

A machine M is ℓ-reachable if any state in the reachable part of M can be entered into by a reset alone or by a reset followed by an input sequence x with $|x| \leq \ell$. Clearly, any resettable machine M is $(|Q|-1)$-reachable.

Example 4

The reachable part of M_3 (see Example 3) is $\{s_0, s_2, s_3\}$. M_3 is 2-reachable since $\rho_3(r_1) = s_0$, $\delta_3(\rho_3(r_1), 1) = s_2$, and $\overline{\delta}_3(\rho_3(r_1), 11) = s_3$.

Theorem 1

Let M and $\tilde{M}$ be two resettable machines. Let P and $\tilde{P}$ be the reachable parts of M and $\tilde{M}$. Then M realizes $\tilde{M}$ if and only if there exists a 4-tuple of functions $(\eta_1, \eta_2, \eta_3, \eta_4)$ where

$$\eta_1: \tilde{I} \rightarrow I$$

$$\eta_2: \tilde{P} \rightarrow \mathcal{P}(P) - \phi$$

$$\eta_3: Z \rightarrow \tilde{Z}$$

$$\eta_4: \tilde{R} \rightarrow R$$

such that

i) $\delta(\eta_2(\tilde{p}), \eta_1(a)) \subseteq \eta_2(\tilde{\delta}(\tilde{p}, a))$ for all $\tilde{p} \in \tilde{P}$ and $a \in \tilde{I}$

ii) $\eta_3(\lambda(p, \eta_1(a))) = \tilde{\lambda}(\tilde{p}, a)$ for all $\tilde{p} \in \tilde{P}$, $a \in \tilde{I}$, and $p \in \eta_2(\tilde{p})$

iii) $\rho(\eta_4(\tilde{r})) \in \eta_2(\tilde{\rho}(\tilde{r}))$ for all $\tilde{r} \in \tilde{R}$.

Proof: (Necessity) Assume $M \rho \tilde{M}$. Then there exists an appropriate triple of functions $(\sigma_1, \sigma_2, \sigma_3)$ such that $\tilde{\beta}_{\tilde{r}}(x) = \sigma_3(\beta_{\sigma_2(\tilde{r})}(\sigma_1(x)))$. Therefore

$$\tilde{\beta}_{\tilde{\rho}(\tilde{r})}(uv) = \sigma_3(\beta_{\rho(\sigma_2(\tilde{r}))}(\sigma_1(uv)))$$

for each $\tilde{r} \in \tilde{R}$, $u \in I^*$, and $v \in I^+$.

Hence,

$$\tilde{\beta}_{\tilde{\delta}(\tilde{\rho}(\tilde{r}),\, u)}(v) = \sigma_3(\beta_{\bar{\delta}(\rho(\sigma_2(\tilde{r}))\,,\,\sigma_1(u))}(\sigma_1(v))).$$

Thus for each $\tilde{p} \in \tilde{P}$ there is a $p \in P$ such that

$$\tilde{\beta}_{\tilde{p}}(v) = \sigma_3(\beta_p(\sigma_1(v))).$$

Consider $\eta_2: \tilde{P} \longrightarrow \mathcal{P}(P) - \phi$ defined by

$$\eta_2(\tilde{p}) = \{p \in P \mid \sigma_3(\beta_p(\sigma_1(v))) = \tilde{\beta}_{\tilde{p}}(v), \text{ for all } v \in I^+\}$$

and consider $\eta_1: \tilde{I} \longrightarrow I$ defined by

$$\eta_1(a) = \sigma_1(a).$$

Claim: The 4-tuple $(\eta_1, \eta_2, \sigma_3, \sigma_2)$ satisfy i), ii), and iii).

i) Let $p \in \eta_2(\tilde{p})$. We must show $\delta(p, \eta_1(a)) \in \eta_2(\tilde{\delta}(\tilde{p}, a))$.

$$\begin{aligned}\tilde{\beta}_{\tilde{\delta}(\tilde{p},\, a)}(x) &= \tilde{\beta}_{\tilde{p}}(xa) \\ &= \sigma_3(\beta_p(\sigma_1(xa))) \\ &= \sigma_3(\beta_{\delta(p,\, \sigma_1(a))}(\sigma_1(x))) \\ &= \sigma_3(\beta_{\delta(p,\, \eta_1(a))}(\sigma_1(x))).\end{aligned}$$

Hence, $\delta(p, \eta_1(a)) \in \eta_2(\tilde{\delta}(\tilde{p}, a))$.

ii) Let $p \in \eta_2(\tilde{p})$. We must show $\sigma_3(\lambda(p, \eta_1(a))) = \tilde{\lambda}(\tilde{p}, a)$.

$$\begin{aligned}\tilde{\lambda}(\tilde{p}, a) &= \tilde{\beta}_{\tilde{p}}(a) \\ &= \sigma_3(\beta_p(\eta_1(a))) \\ &= \sigma_3(\lambda(p, \eta_1(a))).\end{aligned}$$

iii) Let $\tilde{r} \in \tilde{R}$. We must show $\rho(\sigma_2(\tilde{r})) \in \eta_2(\tilde{\rho}(\tilde{r}))$.

$\tilde{\beta}_{\tilde{r}}(x) = \sigma_3(\beta_{\sigma_2(\tilde{r})}(\sigma_1(x)))$ implies that

$\rho(\sigma_2(\tilde{r})) \in \eta_2(\tilde{p}(\tilde{r}))$.

(Sufficiency) Suppose there exists functions $(\eta_1, \eta_2, \eta_3, \eta_4)$ as in the statement of the theorem. Let σ_1: $(\tilde{I})^+ \rightarrow I^+$ be the natural extension of η_1 to sequences. I.e., $\sigma_1(a_1 \ldots a_n) = \eta_1(a_1) \ldots \eta_1(a_n)$.

Claim: $M \rho \tilde{M}$ under $(\sigma_1, \eta_4, \eta_3)$.

Consider ζ: $\tilde{P} \rightarrow P$ where

$\zeta(\tilde{p})$ = some $p \in \eta_2(\tilde{p})$ such that

$\rho(\eta_4(\tilde{r})) = \zeta(\tilde{\rho}(\tilde{r}))$.

Let $x = ya$ where $a \in I$. Then

$$\eta_3(\beta_{\eta_4(\tilde{r})}(\sigma_1(x))) = \eta_3(\beta_{\rho(\eta_4(\tilde{r}))}(\sigma_1(x)))$$

$$= \eta_3(\beta_{\zeta(\tilde{\rho}(\tilde{r}))}(\sigma_1(x)))$$

$$= \eta_3(\lambda(\bar{\delta}(\zeta(\tilde{\rho}(\tilde{r})), \sigma_1(y)), \sigma_1(a)))$$

$$= \eta_3(\lambda(p, \sigma_1(a))) \text{ where } p \in \eta_2(\bar{\tilde{\delta}}(\tilde{\rho}(\tilde{r}), y))$$

$$= \tilde{\lambda}(\bar{\tilde{\delta}}(\tilde{\rho}(\tilde{r}), y), a)$$

$$= \tilde{\beta}_{\tilde{\rho}(\tilde{r})}(ya)$$

$$= \tilde{\beta}_{\tilde{r}}(x).$$

This completes the proof of Theorem 1.

In this study we will not be concerned with the more general theoretical aspects of realizations. What we desire from realizations

is the following. Given a resettable system $\tilde{S}$ we will want to find a resettable system S such that S can do every thing that $\tilde{S}$ can and S has the on-line diagnosis properties that are needed. Generally we will think of S as having two sets of output terminals; one which is used in place of the output terminals of $\tilde{S}$ and the other which is used solely for diagnosis.

To formalize this notion of a system having more than one set of output terminals we introduce the notion of a structured set. As defined by Zeigler [19], a set k is <u>structured</u> by injecting it into a cross product of an indexed family $\{K_i \mid i \in N\}$. In what follows we will take N to be a finite ordered set such as the first n integers. Thus a structure assignment is a one-one map from K into $\times_{i \in N} K_i$. Normally we do not mention this map explicitly but will consider K (once structured) as a subset of $\times_{i \in N} K_i$. Given a structured set K a family of <u>coordinate projections</u> $\{P_i \mid i \in N\}$ where $P_{i_j} : K \rightarrow K_{i_j}$ is defined by

$$P_{i_j}(k_{i_1}, \ldots, k_{i_j}, \ldots, k_{i_n}) = k_{i_j}.$$

With these notions in mind the special type of realization which will be used in our theory of on-line diagnosis can be presented.

Definition 6

Let S and $\tilde{S}$ be two resettable systems with Z structured so that $Z \subseteq Z_1 \times Z_2$. Then S <u>d-realizes</u> $\tilde{S}$ ($S_{\rho_d}\tilde{S}$) if $S_{\rho}\tilde{S}$ under the triple of functions $(\sigma_1, \sigma_2, \sigma_3)$ where $\sigma_3 = \sigma_3' \circ P_1$ for some $\sigma_3' : Z_1 \longrightarrow \tilde{Z}$.

I.e., $S_{\rho_d}\tilde{S}$ if $S_{\rho}\tilde{S}$ and the output decoding is independent of the second coordinate of Z. In this case Z_1 is called the <u>principle output</u> and is given the more mnemonic name Z_P and Z_2 is called the <u>augmented output</u> and is given the name Z_A. Thus, $Z \subseteq Z_P \times Z_A$.

Given that $S_{\rho_d}\tilde{S}$ we can define two new functions associated with $\beta_{r,t}$, the behavior of S for condition (r, t). The first one will be the behavior function of S with respect to the output terminals which are used to mimic $\tilde{S}$ and the second will be the behavior function of S with respect to the output terminals which are used solely for diagnosis. More precisely, the <u>principle behavior of S for condition (r, t)</u> is the function

$$\gamma_{r,t} : I^+ \longrightarrow Z_P$$

where

$$\gamma_{r,t}(x) = P_1(\beta_{r,t}(x)) \text{ for each } x \in I^+$$

or more compactly,

$$\gamma_{r,t} = P_1 \circ \beta_{r,t}.$$

The augmented behavior of S for condition (r, t) is the function

$$\alpha_{r,t}: I^+ \rightarrow Z_A$$

where $$\alpha_{r,t} = P_2 \circ \beta_{r,t}.$$

Thus $\beta_{r,t}(x) = (\gamma_{r,t}(x), \alpha_{r,t}(x))$ for all $x \in I^+$. We now extend these functions in a natural way. For $r \in R$ and $t \in T$ let

$$\hat{\beta}_{r,t}: I^+ \rightarrow Z^+$$

where for all $a_1 \ldots a_n \in I^+$

$$\hat{\beta}_{r,t}(a_1 \ldots a_n) = \beta_{r,t}(a_1) \ldots \beta_{r,t}(a_1 a_2 \ldots a_n).$$

Likewise let $\hat{\gamma}_{r,t}$ and $\hat{\alpha}_{r,t}$ denote the natural extensions of $\gamma_{r,t}$ and $\alpha_{r,t}$ to Z_P^+ and Z_A^+ respectively.

4. Resettable Systems with Faults

Our model of a "resettable system with faults" is a specialization of Meyer's general model of a "system with faults" [20].

> Informally, a "system with faults" is a system, along with a set of potential faults of the system and description of what happens to the original system as the result of each fault. The original system and the systems resulting from faults are members of one of two prescribed classes of (formal) systems, a "specification" class for the original system and a "realization" class for the resulting systems. More precisely, we say that a triple $(\mathcal{S}, \mathcal{R}, \rho)$ is a (system) representation scheme if
>
> i) $\mathcal{S}$ is a class of systems, the specification class,
> ii) $\mathcal{R}$ is a class of systems, the realization class,
> iii) ρ: $\mathcal{R} \rightarrow \mathcal{S}$ where, if $R \in \mathcal{R}$, R realizes $\rho(R)$.
>
> By a class of systems, in this context, we mean a class of formal systems, i.e. a set of formally specified structures of the same type, each having an associated behavior that is determined by the structure [20].

In this study we are concerned with the reliable use of a system. I.e., we are concerned with degradations in structure which Meyer calls "life defects". This is contrasted with reliable design in which case we would be concerned with "birth defects". Thus, in our case, a specification is a realization and we choose a representation scheme $\mathcal{R} = (\mathcal{R}, \mathcal{R}, \rho)$ where ρ is the identity function on $\mathcal{R}$.

Assuming that a faulty resettable system has the same input, output, and reset alphabets as the fault-free system S, the following class of resettable systems will suffice as a realization class:

$$\mathcal{S}(I, Z, R) = \{S' \mid S' = (I, Q', Z, \delta', \lambda', R, \rho')\}.$$

In summary, the representation scheme that we are choosing for our study of on-line diagnosis is the scheme $(\mathcal{R}, \mathcal{R}, \rho)$ where $\mathcal{R} = \mathcal{S}(I, Z, R)$ and ρ is the identity function on $\mathcal{R}$.

In such a scheme the seemingly difficult problem of describing faults and their results becomes relatively straightforward. Before we state our particular notion of a fault and its results we will repeat here Meyer's general notion of a "system with faults" [20].

A system with faults in a representation scheme $(\mathcal{S}, \mathcal{R}, \rho)$ is a structure (S, F, ϕ) where

i) $S \in \mathcal{S}$
ii) F is a set, the faults of S
iii) ϕ: $F \rightarrow \mathcal{R}$ such that, for some $f \in F$, $\rho(\phi(f)) = S$.

If $f \in F$, the system $S^f = \phi(f)$ is the result of f. If $\rho(S^f) = S$ then f is improper (by iii), F contains at least one improper fault); otherwise it is proper. A realization S^f is fault-free if f is improper; otherwise S^f is faulty [20].

In applying this notion to our study we must first define what we mean by a fault of a resettable system. Given a resettable system $S \in \mathcal{S}(I, Z, R)$, a fault f of S can be regarded as a transformation of S into another system $S' \in \mathcal{S}(I, Z, R)$ at some time τ. Accordingly, the resulting faulty system looks like S up to time τ and like S' thereafter. Since S may be in operation at time τ we must also be concerned with the question of what happens to the state of S as this transformation takes place. We handle this with a function θ from the state set of S to that of S'. The interpretation of θ is that if S is in state q immediately before time τ then S' is in state $\theta(q)$ at time τ. More precisely,

Definition 7

If $S \in \mathcal{S}(I, Z, R)$, a fault of S is a triple

$$f = (S', \tau, \theta)$$

where $S' \in \mathcal{S}(I, Z, R)$, $\tau \in T$, and θ: $Q \longrightarrow Q'$.

Given this formal representation of a fault of S, the resulting faulty system is defined as follows.

Definition 8

The result of $f = (S', \tau, \theta)$ is the system

$$S^f = (I, Q^f, Z, \delta^f, \lambda^f, R, \rho^f)$$

where $Q^f = Q \cup Q'$

$$\delta^f(q, a, t) = \begin{cases} \delta(q, a, t) \text{ if } q \in Q \text{ and } t < \tau - 1 \\ \theta(\delta(q, a, t)) \text{ if } q \in Q \text{ and } t = \tau - 1 \\ \delta'(q, a, t) \text{ if } q \in Q' \text{ and } t \geq \tau \end{cases}$$

$$\lambda^f(q, a, t) = \begin{cases} \lambda(q, a, t) \text{ if } q \in Q \text{ and } t < \tau \\ \lambda'(q, a, t) \text{ if } q \in Q' \text{ and } t \geq \tau \end{cases}$$

$$\rho^f(r, t) = \begin{cases} \rho(r, t) \text{ if } t < \tau \\ \theta(\rho(r, t)) \text{ if } t = \tau \\ \rho'(r, t) \text{ if } t > \tau. \end{cases}$$

(Arguments not specified in the above definitions may be assigned arbitrary values.)

In justifying this representation of the resulting faulty system one should regard a fault $f = (S', \tau, \theta)$ as actually occurring between time $\tau - 1$ and τ. Note that, for any fault f of S, $S^f \in \mathcal{S}(I, Z, R)$.

Example 5

Recall that in Example 2 M_1 was transformed into M_1' at time 100. We would say now that $f = (M_1', 100, e)$, where e is the identity function, is a fault of M_1 and that S is the result of f (i.e., $S = M_1^f$).

Example 6

Again consider M_1 as implemented by the circuit in Figure 2 and let g be the fault which is caused by d_1 becoming stuck-at-1 at

time 50. Then $g = (M_1'', 50, \theta)$ where M_1'' is an indicated in Figure 9 and $\theta : Q_1 \longrightarrow Q_1''$ is defined as follows:

q	θ(q)
00	10
01	11
10	10
11	11

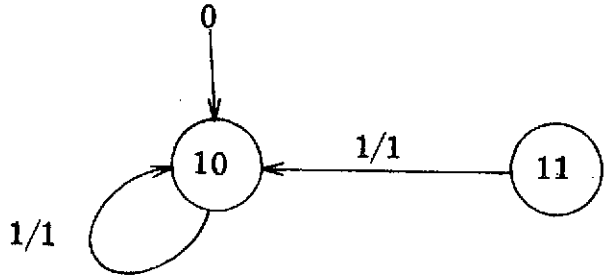


Figure 9 Resettable Machine M_1''

M_1^g will behave as M_1 up to time 50 and thereafter it will produce a constant sequence of 1's.

To complete the model, a resettable system with faults, in this representation scheme, is a structure

$$(S, F, \phi)$$

where $S \in \mathcal{S}(I, Z, R)$, F is a set of faults of S including at least one improper fault (e.g., $f = (S, 0, e)$ where e is the identity function), and ϕ: $F \longrightarrow \mathcal{S}(I, Z, R)$ where $\phi(f) = S^f$, for all $f \in F$. Given this definition, we can drop the explicit reference to ϕ in denoting a resettable system with faults, i.e., (S, F) will mean (S, F, ϕ) where ϕ is as defined above.

In the remainder of this study we will be dealing almost exclusively with resettable systems. Thus we will refer to resettable systems simply as systems and to resettable machines as machines.

A word is in order about our definition of faults. The interpretation here is one of effect, not cause, e.g. we don't talk of stuck-at-1 OR gates but rather of the system which is created due to some presumed physical cause. We will refer to these physical causes as component failures or simply as failures. A fault, by our definition, consists of precisely that information which is needed to define the system which results from the fault. This allows us to treat faults in the abstract; independent of specific network realizations of the system and without reference to the technology employed in this realization and the types of failures which are possible with this technology. We are insured, however, that for each fault we have enough information to access the structural and behavioral effects of the fault; in particular as these effects relate to fault diagnosis and tolerance.

There are limits, however, to how much can be done with a purely effect oriented concept of faults. When a system is sufficiently structured to allow a reasonable notion of what may cause a fault we certainly will want to make use of this notion. When this is the case we may, through an abuse in language, refer to a specific failure at time τ as a fault. What we will mean is that we have stated a cause

of fault and that there is a unique fault which is the result of this failure at time τ.

It is interesting to see what the scope of our definition of fault is in terms of the types of failures which will result in faults. Recall that a fault f of a system S is a triple, $f = (S', \tau, \theta)$, where $S' \in \mathcal{S}(I, Z, R)$. Thus S' is a (resettable) system with the same input, output, and reset alphabets as S. The previous sentence contains, implicitly, almost every restriction that we have put on faults. First of all, S' is a (resettable) system. Thus it remains within our universe of discourse. In particular, its reset inputs still act like reset inputs. I.e., they cause S' to go into a particular state regardless of the state it was in when the reset input was applied. The restrictions on the input, output, and reset alphabets are reasonable since after a fault occurs the system presumably will have the same input and output terminals as it had before the fault occurred.

We see that since a fault f is a triple (S', τ, θ) with S' a (time-varying) system that we will have considerable latitude in the types of causes of faults which we may consider. In particular, we may consider simultaneous permanent failures in one or more components, simultaneous intermittent failures in one or more components, or any combination of the above occurring at the same or varying times. For example, a fault f may be caused by an AND gate becoming stuck-at-1

at time τ_1, followed by an OR gate becoming stuck-at-0 at time τ_2. Our main interest will be the case where the fault is caused by the failure of only one component, since usually such a failure will be diagnosed before a second failure occurs. In the case where a fault of a machine M is caused by a permanent failure of one or more components at only one time f will be of the form (M', τ, θ).

Let us now compute the behavior of S^f in state q. Let $x = a_1 \ldots a_n \in I^+$. Then

$$\beta_q^f(x,t) = \overline{\lambda}^f(q,x,t)$$

$$= \lambda^f(\overline{\delta}^f(q, a_1 \ldots a_{n-1}, t), a_n, t+n-1).$$

There are three cases which must be considered.

Case i) $q \in Q$ and $t + n-1 < \tau$. Then

$$\beta_q^f(x,t) = \lambda(\overline{\delta}(q, a_1 \ldots a_{n-1}, t), a_n, t+n-1)$$

$$= \beta_q(x,t).$$

Case ii) $q \in Q$, $t + n-1 \geq \tau$, and $t < \tau$. Say $t + n-m = \tau$. Then

$$\beta_q^f(x,t) = \lambda'(\overline{\delta}'(\theta(\overline{\delta}(q, a_1, \ldots, a_{n-m}, t)), a_{n-m+1} \ldots a_{n-1},$$
$$t+n-m), a_n, t+n-1)$$

$$= \beta'_{\theta(\overline{\delta}(q, a_1 \ldots a_{n-m}, t))}(a_{n-m+1} \ldots a_n, t+n-m)$$

$$= \beta'_{\theta(\overline{\delta}(q,y,t))}(z,t) \text{ where } y = a_1 \ldots a_{n-m}$$
$$\text{and} \quad z = a_{n-m+1} \ldots a_n.$$

Case iii) $q \in Q'$ and $t \geq \tau$. Then

$$\begin{aligned}\beta^f_q(x,t) &= \lambda'(\bar{\delta}'(q, a_1 \ldots a_{n-1}, t), a_n, t + n-1) \\ &= \beta'_q(x,t).\end{aligned}$$

Thus we have proved:

Theorem 2

Let S be a system and $f = (S', \tau, \theta)$ a fault of S. Then for each $t \in T$ and $x \in I^+$

$$\beta^f_q(x,t) = \begin{cases} \beta_q(x,t) \text{ if } q \in Q \text{ and } t + |x| \leq \tau \\ \beta'_{\theta(\bar{\delta}(q,y,t))}{}^{(z,\tau)} \text{ if } q \in Q,\ t + |x| > \tau, \text{ and } t < \tau \\ \qquad\qquad \text{where } x = yz \text{ and } |y| = \tau - t \\ \beta'_q(x,t) \text{ if } q \in Q' \text{ and } t \geq \tau. \end{cases}$$

(As in the definitions of δ^f and λ^f arguments not specified may be assigned arbitrary values.)

Corollary 2.1

Let S be a system and $f = (S', \tau, \theta)$ a fault of S. Then for each $r \in R$, $t \in T$, and $x \in I^+$

$$\beta^f_{r,t}(x) = \begin{cases} \beta_{r,t}(x) \text{ if } t + |x| \leq \tau \\ \beta'_{\theta(\bar{\delta}(\rho(r,t),y,t))}{}^{(z,\tau)} \text{ if } t + |x| > \tau \text{ and} \\ \qquad\qquad t \leq \tau \text{ where} \\ \qquad\qquad x = yz \text{ and } |y| = \tau - t \\ \beta'_{r,t}(x) \text{ if } t > \tau. \end{cases}$$

Proof: By its definition

$$\beta^{f}_{r,t}(x) = \beta^{f}_{\rho^{f}(r,t)}(x,t).$$

Again we have three cases to consider.

Case i) $t + |x| \leq \tau$. Then $t < \tau$ and $\rho^{f}(r,t) = \rho(r,t) \in Q$. Therefore by Theorem 2

$$\begin{aligned}\beta^{f}_{\rho^{f}(r,t)}(x,t) &= \beta_{\rho(r,t)}(x,t)\\ &= \beta_{r,t}(x).\end{aligned}$$

Case ii) $t + |x| > \tau$ and $t \leq \tau$. If $t < \tau$ then $\rho^{f}(r,t) = \rho(r,t) \in Q$ and case ii) of Theorem 2 applies with $\rho(r,t)$ in place of q. If $t = \tau$ then $\rho^{f}(r,t) = \theta(\rho(r,t)) \in Q'$ and case iii) of the theorem applies giving us

$$\begin{aligned}\beta^{f}_{\rho^{f}(r,t)}(x,t) &= \beta'_{\theta(\rho(r,t))}(x,t)\\ &= \beta'_{\theta(\bar{\delta}(\rho(r,t),\Lambda,t))}(x,t).\end{aligned}$$

Case iii) $t > \tau$. In this case $\rho^{f}(r,t) = \rho'(r,t) \in Q'$. Therefore

$$\begin{aligned}\beta^{f}_{\rho^{f}(r,t)}(x,t) &= \beta'_{\rho'(r,t)}(x,t)\\ &= \beta'_{r,t}(x).\end{aligned}$$

We have noted that we will often be interested in the physical cause of a fault. For example, in a network realization of a machine we may be interested in faults which are caused by a specific NAND gate becoming stuck-at-1. Since this gate failure results in different

faults as we consider it occurring at different times it seems natural to give a name to this family of faults. More generally, we will define an equivalence relation on a set of faults such that a family of faults such as we have just mentioned will be an equivalence class.

Definition 9

Let F be a set of faults of a system S and let $f_1 = (S_1, \tau_1, \theta_1)$ and $f_2 = (S_2, \tau_2, \theta_2)$ be in F. Then f_1 is equivalent to f_2 ($f_1 \equiv f_2$) if S_1 and S_2 are such that

i) $Q_1 = Q_2$

ii) $\delta_1(q, a, t + \tau_1) = \delta_2(q, a, t + \tau_2)$ for all $q \in Q$, $a \in I$, and $t \in T$

iii) $\lambda_1(q, a, t + \tau_1) = \lambda_2(q, a, t + \tau_2)$ for all $q \in Q$, $a \in I$, and $t \in T$

iv) $\rho_1(r, t + \tau_1) = \rho_2(r, t + \tau_2)$ for all $r \in R$, and $t \in T$

and if $\theta_1 = \theta_2$.

We can think of equivalent faults as being time-translations of one another.

Theorem 3

The above relation is an equivalence relation.

Proof: It is clearly reflexive, symmetric, and transitive because "=" has these properties and because the quantifiers, for all $q \in Q$ etc., are independent of the particular fault.

Notation: We denote then equivalence class of F which contains the fault f by $[f]_F$. When the class of faults is clear we will drop the F.

Generally if F is not mentioned we take it to be the set of all possible faults of a system S. We let $f_i = (S_i, i, \theta)$ denote the fault in $[f]$ which occurs at time i. When dealing with behaviors β^{f_i} will denote the behavior of S^{f_i}, and β^i will denote the behavior of S_i.

From the definition we can see that if $f = (M', \tau, \theta)$ where M' is a machine then $[f] = \{(M', t, \theta) \mid t \in T\}$.

Let f be a fault of a machine M. It is clear from Definition 9 that $f_i \equiv f_j$ implies that $\beta^i_q(x, t + i) = \beta^j_q(x, t + j)$ for all $t \in T$. Likewise,

$$\beta^i_{r,\, t+i}(x) = \beta^j_{r,\, t+j}(x) \text{ for all } t \in T.$$

Since M is time-invariant it is a direct consequence of Theorem 2 and the above observation that there is a similar relation between the behaviors of M^{f_i} and M^{f_j}. More precisely,

Theorem 4

Let f be a fault of M and let $f_i, f_j \in [f]$. Then for all $q \in Q$, $x \in I^+$, $r \in R$, and $t \in T$

$$\beta^{f_i}_q(x, t + i) = \beta^{f_j}_q(x, t + j)$$

and

$$\beta^{f_i}_{r,\, t+i}(x) = \beta^{f_j}_{r,\, t+j}(x).$$

5. Fault Tolerance and Errors

Given a system with faults (S, F) and a proper fault $f \in F$, an immediate question is whether the faulty system S^f is usable in the sense that its behavior resembles, within acceptable limits, that of the fault-free system S. We will use the general notion of a "tolerance relation" [20] to make more precise what is meant by "acceptable limits." A tolerance relation for a representation scheme $(\mathcal{S}, \mathcal{R}, \rho)$ is a relation τ between $\mathcal{R}$ and $\mathcal{S}$ ($\tau \subseteq \mathcal{R} \times \mathcal{S}$) such that, for all $R \in \mathcal{R}$, $(R, \rho(R)) \in \tau$ (i.e. $\rho \subseteq \tau$). In this section we will develop the particular notions of "acceptable limits" that we will be using in this study of on-line diagnosis.

At this point in our development we will assume that we are given two systems S and $\tilde{S}$ where $S \rho_d \tilde{S}$. Thus the principle and augmented behaviors of S will be defined. More generally, assume that we are given any system S with structured output $Z \subseteq Z_P \times Z_A$. Such a system will be called an output-augmented system. Clearly the definitions of principle and augmented behaviors apply to output-augmented systems.

If $f = (S', \tau, \theta)$ is a fault of S then since the output alphabet of S^f is the same as that of S it can be given the same structure, and henceforth we will always assume that this has been done. Accordingly, we can compare the principle and augmented behaviors of S^f with those of S.

Note that any system S can be considered as an output-augmented system by considering Z to be $Z \times \{0\}$. Given a system S with unstructured output alphabet Z we will assume this trivial augmentation structure. In this case the principle behavior of S will be identical to the behavior of S.

Definition 10

Let f be a fault of a system S. Then f is tolerated by S for resets at time t if

$$\beta_{r,t}(x) = \beta^{f}_{r,t}(x) \quad \text{for each } r \in R \text{ and } x \in I^{+}.$$

In the special case where f is tolerated by S for resets at time 0 we will simply say f is tolerated by S.

Note that this is a very refined notion of fault tolerance. A coarser notion, and one more in keeping with the literature, would be behavioral equivalence for resets at any time. We prefer our finer definition for with it the effects of time can be more naturally analyzed. One question which we will study later is: For resets at how many (and which) times must a fault be tolerated for it to be tolerated for resets at any time?

Theorem 5

Let $f = (S', \tau, \theta)$ be a fault of machine M. Then f is tolerated by M for resets at time t if and only if $f_{\tau - t}$ is tolerated by M.

Proof: $f_{\tau-t}$ is tolerated by M $\iff \beta_{r,o}(x) = \beta^{f_{\tau-t}}_{r,o}(x)$

$\iff \beta_{r,t}(x) = \beta^{f}_{r,t}(x)$

$\iff$ f is tolerated by M for resets at time t.

The second implication follows from Theorem 4 and the hypothesis that M is a machine (i.e., a time-invariant system).

Thus, $f_i, f_j, f_k, \ldots$ is tolerated by M for resets at time $t_1, t_2, t_3, \ldots$ respectively if and only if $\{f_{i-t_1}, f_{j-t_2}, f_{k-t_3}, \ldots\}$ is tolerated by M where by F is tolerated by M we mean that each $f \in F$ is tolerated by M. Due to this we will always consider resets to be released at time 0 when dealing with fault tolerance of machines and no generality will be lost. Clearly, due to Theorem 4 we can do this same sort of thing for any other behavioral attribute.

Example 7

Let M_4 be the sequence generator shown in Figure 10. This machine could be implemented by the circuit shown in Figure 11.

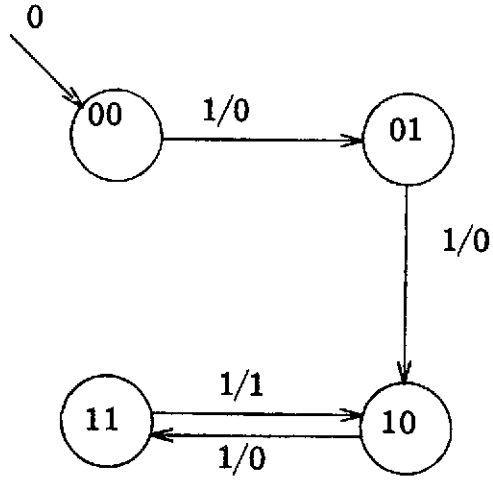


Figure 10 Machine M_4

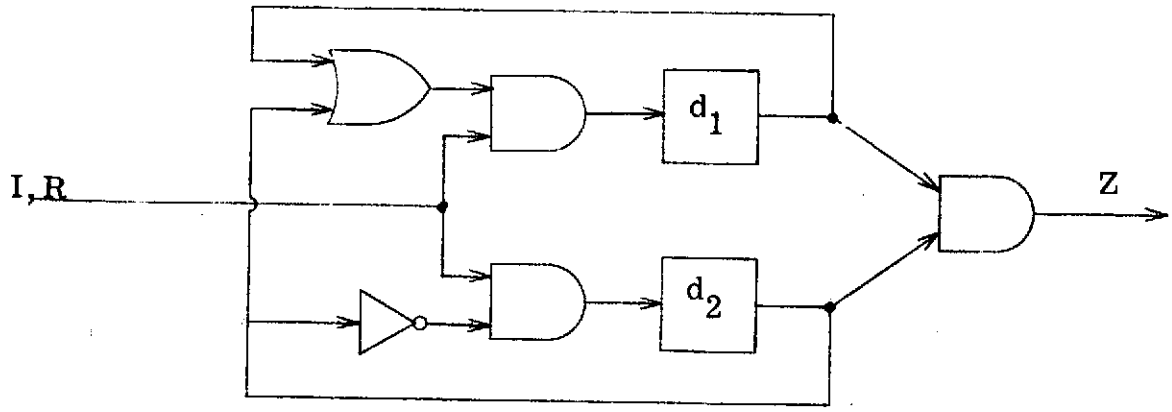


Figure 11 Circuit for M_4

Let f be a fault of M_4 which is caused by d_1 becoming stuck-at-1 at time τ. Then $f = (M_4', \tau, \theta)$ where M_4' is the machine represented by the graph in Figure 12 and θ is as indicated below.

q	$\theta(q)$
00	10
01	11
10	10
11	11

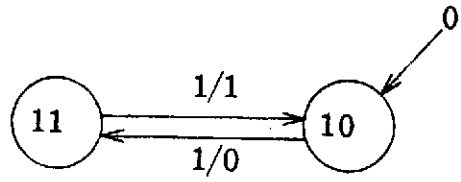


Figure 12 Machine M_4'

Consider f_{-1}, i.e. the fault $(M_4', -1, \theta)$, and note that $\beta_0^{f_{-1}}(11) = 1$ whereas $\beta_0(11) = 0$. Thus f_{-1} is not tolerated by M_4. On the other hand both M_4 and $M_4^{f_{-1}}$ will produce the sequence 00010101. . . when reset at -10. Thus f_{-1} is tolerated by M_4 for resets at -10. By

applying Theorem 5 one can learn that f_i is not tolerated by M_4 for resets at time i + 1 and that f_9 is tolerated by M_4.

Recall that our goal is to develop a theory of on-line diagnosis for time-invariant systems and that we have introduced time-varying systems only to be able to represent the dynamics of time-invariant systems as faults occur. However, it has been the case thus far that this theory has generalized in a straightforward manner to a theory of on-line diagnosis for time-varying systems. For example, we have defined a fault of a system where we could have simply defined a fault of a machine, and we have defined a notion of fault tolerance for systems.

From this point on generalizations of this sort will not be valid for we will always be considering resets to be released at time 0 and for time-varying systems this simplification is not possible. A theory of on-line diagnosis of systems could be developed along the line of what we will present for machines but we will no longer pursue it.

Definition 11

Let f be a fault of a machine M and let g be an arbitrary function from Z into some set $\hat{Z}$. Then f is g-tolerated by M if for each r in R and x in I^+

$$g(\beta_r(x)) = g(\beta_r^f(x)).$$

If $g = P_1(P_2)$ then g-tolerated corresponds to behavioral correctness with respect to the principle (augmented) behavior and we will use the

suggestive term γ-tolerated (α-tolerated). If $M_{\rho_d} \tilde{M}$ under the triple of functions $(\sigma_1, \sigma_2, \sigma_3)$ then σ_3-tolerated becomes important for it corresponds to correctness with respect to the originally specified behavior, i.e., the behavior of $\tilde{M}$.

Note that f is tolerated by M implies that f is g-tolerated by M for every g. Also, f is tolerated if and only if f is γ-tolerated and α-tolerated.

Due to the definitions of the α and γ functions (in terms of projections composed with the β function) definition and theorems concerning β can generally be transformed into corresponding definitions and theorems which relate to the α and γ functions. This is true in general for any behavior function of the form $g \circ \beta$. When this is the case, as in the next definition, only the β function will be mentioned explicitly.

Definition 12

Let f be a fault of M, $r \in R$, and $x \in I^+$. Then f with initial reset r and input x will cause an error if

$$\hat{\beta}^f_r(x) \neq \hat{\beta}_r(x).$$

To avoid this cumbersome phrase if $\hat{\beta}^f_r(x) \neq \hat{\beta}_r(x)$ we will simply say that (f, r, x) is an error, and when it is clear that we are interested not only in the erroneous output sequence but also in how it arises we will say that $\hat{\beta}^f_r(x)$ is an error.

When we are interested in errors with respect to other behavior functions we will use the phrases: γ-error, α-error, or most generally, g-error.

Example 8

Recall that in example 7 $f = (M_4', \tau, \theta)$ was a fault of M_4 and that $\beta_0^{f_{-1}}(11) \neq \beta_0(11)$. Thus $(f_{-1}, 0, 11)$ is an error and 01 is the erroneous output sequence caused by this error.

Clearly, $\hat{\gamma}_r^f(x)$ is an γ-error implies $\hat{\beta}_r^f(x)$ is an error but not conversely. Observe that $\hat{\beta}_r^f(x) \neq \hat{\beta}_r(x)$ implies $\hat{\beta}_r^f(xy) \neq \hat{\beta}_r(xy)$ for all $y \in I^*$. Thus $\hat{\beta}_r^f(x)$ is an error implies $\hat{\beta}_r^f(xy)$ is also.

If $y \in I^+$ and $a \in I$ are such that $\hat{\beta}_r^f(ya)$ is an error but $\hat{\beta}_r^f(y)$ is not, then ya is a minimal error input for M^f with initial reset r. In this case $\beta_r^f(x) \neq \beta_r(x)$ where $x = ya$ and we say that (f, r, x) (alternatively, $\beta_r^f(x)$) is a minimal error.

Note that if f is tolerated then f can cause no errors. Equivalently, if there exists $r \in R$ and $x \in I^+$ such that $\hat{\beta}_r^f(x)$ is an error then f is not tolerated. The converse to this is also true. Namely, if f is not tolerated then there exist $r \in R$ and $x \in I^+$ such that $\hat{\beta}_r^f(x)$ is an error.

Our definition of tolerated induces a relation τ on $\mathfrak{R}$ where $M^f \tau M$ if and only if f is tolerated by M. If f is improper then $M^f = M$ and thus f is tolerated by M. Hence $M \tau M$, and therefore τ is a tolerance relation. Likewise γ-tolerated and α-tolerated induce tolerance rela-

tions τ_γ and τ_α. We say that a fault f is τ-diagnosable if f is not tolerated by M, (i.e. $M^f \not\sim M$). Thus f is τ-diagnosable if and only if f will cause an error for some initial reset r and input x. Finally, we note that since f is tolerated implies that f is γ-tolerated, as sets $\tau \subseteq \tau_\gamma$. Thus it is possible to consider faults which are τ_γ-tolerated and τ-diagnosable.

Often we will be in a situation where we are concerned with a machine M tolerating a set of faults which are all caused by the same phenomenon but which may occur at any time. More specifically, let f be a fault of M. We would like a result which assured us that if some finite subset of [f] was tolerated by M then all of [f] was tolerated by M. Later we will be interested in the same problem with regard to diagnosis. The following notion of equivalent errors will be very useful to us as we investigate this problem.

Informally, we will say that two errors (f_i, r_i, x) and (f_j, r_j, y) with $i, j \geq 0$ are equivalent if they are caused by equivalent faults, if the inputs x and y are such that M^{f_i} and M^{f_j} will receive identical input sequences from time i and time j respectively, and if the initial resets r_i and r_j and the inputs x and y are such that M with initial reset r_i and input x would arrive at time i to the same state to which it would arrive at time j given the initial reset r_j and the input y. In other words, from time i in M^{f_i} and time j in M^{f_j} exactly the same thing will happen to exactly the same systems modulo a translation in time. More precisely,

Definition 13

Let $f = (S', \tau, \theta)$ be a fault of M and let $f_i, f_j \in [f]$ with $i, j \geq 0$. Let (f_i, r_i, x) and (f_j, r_j, y) be two errors. Then (f_i, r_i, x) is equivalent to (f_j, r_j, y) ($(f_i, r_i, x) \equiv (f_j, r_j, x)$) if

i) $x = x_1 z$ and $y = y_1 z$ where $|x_1| = i$ and $|y_1| = j$.

ii) $\bar{\delta}(\rho(r_i), x_1) = \bar{\delta}(\rho(r_j), y_1)$.

It is easy to see that this relation is in fact an equivalence relation. I.e., it is reflexive, symmetric, and transitive.

The next result shows us one way in which we can manufacture equivalent errors and it has an immediate corollary in the realm of fault tolerance. This result is a simple consequence of the fact that any state which is reachable in an ℓ-reachable machine is reachable by time ℓ.

Theorem 6

Let f be a fault of an ℓ-reachable machine M and let (f_i, r, x) be an error where $i \geq 0$. Then there exists an equivalent error (f_j, s, y) with $0 \leq j \leq \ell$.

Proof: Let $x = x_1 z$ where $|x_1| = i$ and let $q = \bar{\delta}(\rho(r), x_1)$. Since q is in the reachable part of M and M is ℓ-reachable there exists $s \in R$ and $y_1 \in I^*$ such that $\bar{\delta}(\rho(s), y_1) = q$ and $|y_1| \leq \ell$. Take $j = |y_1|$ and $y = y_1 z$. Clearly, (f_j, s, y) is an error and by its construction it is equivalent to (f_i, r, x).

Corollary 6.1

Let f be a fault of an ℓ-reachable machine M and suppose that $\{f_0, \ldots, f_\ell\}$ is tolerated by M. Then $\{f_0, f_1, \ldots\}$ is tolerated by M.

Proof: Assume that f_i with $i \geq 0$ is not tolerated by M. Then there exists an error (f_i, r, x). By Theorem 6 there exists an equivalent error (f_j, s, y) with $0 \leq j \leq \ell$. Therefore f_j is not tolerated by M. Contradiction. Hence, f_i is tolerated by M for all $i \geq 0$.

Corollary 6.2

Let f be a fault of M with reachable part P. Suppose that $\rho(R) = P$ and that f_0 is tolerated by M. Then $\{f_0, f_1, \ldots\}$ is tolerated by M.

Proof: Since $\rho(R) = P$, M is 0-reachable. Apply Corollary 6.1.

Now we will focus our attention on faults which occur before time 0. In the previous results we have excluded this case because if f_i and f_j are equivalent faults with i or j less than 0 there is, in general, no relation with respect to resets at time 0 between the behaviors of M^{f_i} and M^{f_j}. However, in the important special case where $f = (M', \tau, \theta)$ any $f_i \in [f]$ with $i < 0$ will, with respect to resets released at time 0, cause identical behavior. This is because $f_i = (M', i, \theta)$ and by Corollary 2.1, $\beta_r^{f_i}(x) = \beta'_r(x)$ for all $i < 0$.

Theorem 7

Let $f = (M', \tau, \theta)$ be a fault of M. Then $\beta_r^{f_i}(x) = \beta'_r(x)$ for all $r \in R$, $x \in I^+$, and $i < 0$. In addition, if f_j is tolerated by M for some $j < 0$ then f_i is tolerated by M for all $i < 0$.

Proof: We have already shown the first statement. Thus $\beta_r^{f_i}(x) = \beta_r^{f_j}(x)$ for all $i, j < 0$ and clearly one is tolerated if and only if the other is tolerated.

If $f = (M', \tau, \theta)$ is a fault of M we think of f as affecting the reset mechanism of M if $\rho'(r) \neq \theta(\rho(r))$ for some $r \in R$. If this is not the case then a further result, similar to Theorem 7, can be obtained.

Theorem 8

Let $f = (M', \tau, \theta)$ be a fault of M and suppose that $\rho'(r) = \theta(\rho(r))$ for all $r \in R$. Then $\beta_r^{f_o}(x) = \beta_r'(x)$ for all $r \in R$ and $x \in I^+$. In addition, if f_j is tolerated by M for some $j \leq 0$ then f_i is tolerated by M for all $i \leq 0$.

Proof: Since $\rho'(r) = \theta(\rho(r))$, it is immediate from Corollary 2.1 that $\beta_r^{f_o}(x) = \beta_r'(x)$. Therefore $\beta_r^{f_j}(x) = \beta_r^{f_i}(x)$ for all $i, j \leq 0$ and the result follows from this.

Combining Theorem 7 with Corollary 6.1 we have

Theorem 9

Let $f = (M', \tau, \theta)$ be a fault of an ℓ-reachable machine M and suppose that $\{f_{-1}, f_o, \ldots, f_\ell\}$ is tolerated by M. Then [f] is tolerated by M.

We finish this section by restating Corollary 6.2 and Theorem 8 as a result which in some sense is the best possible.

Theorem 10

Let M be a machine with reachable part P and let $f = (M', \tau, \theta)$ be a fault of M. Suppose $\rho'(r) = \theta(\rho(r))$ for each r in R, $\rho(R) = P$, and f_j is tolerated by M for some $j \leq 0$. Then $[f]$ is tolerated by M.

Proof: By Theorem 8 f_i is tolerated by M for all $i \leq 0$. Therefore f_o is tolerated by M, and thus by Corollary 6.2 f_i is tolerated by M for all $i \geq 0$. Thus, $[f]$ is tolerated by M.

6. On-line Diagnosis

Before we can present our concept of on-line diagnosis in the framework that we have built we need one final definition.

Definition 14

Let S_1 and S_2 be two systems. If $R_1 = R_2$ and $Z_1 \subseteq I_2$ then the series connection of S_1 and S_2 is the system

$$S_1 * S_2 = (I_1, Q, Z_2, \delta, \lambda, R_1, \rho)$$

where $$Q = Q_1 \times Q_2$$

$$\delta((q_1, q_2), a, t) = (\delta_1(q_1, a, t), \delta_2(q_2, \lambda_1(q_1, a, t), t))$$

$$\lambda((q_1, q_2), a, t) = \lambda_2(q_2, \lambda_1(q_1, a, t), t)$$

$$\rho(r, t) = (\rho_1(r, t), \rho_2(r, t)).$$

Schematically, $S_1 * S_2$ can be pictured as in Figure 13.

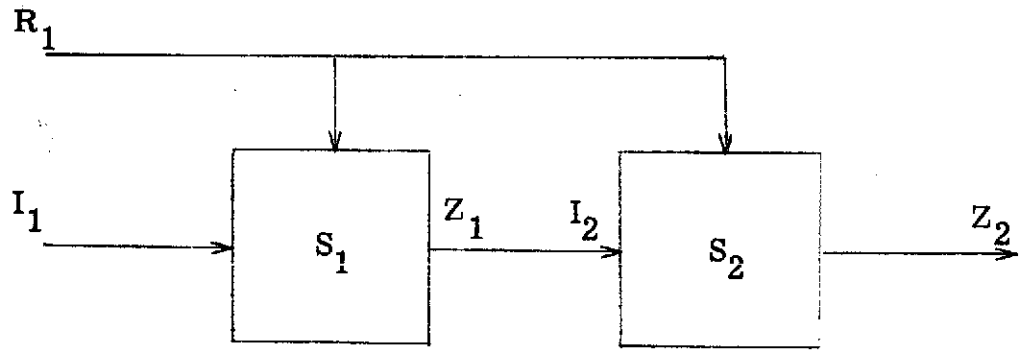


Figure 13 The Series Connection of S_1 and S_2

Given a series connection $S_1 * S_2$ as above we will let β^1, β^2, and β^* denote the behavior functions of S_1, S_2 and $S_1 * S_2$ respectively. We now state the intuitive result that the behavior of $S_1 * S_2$ is equal to the extended behavior of S_1 composed with the behavior of S_2.

Theorem 11

Let $S_1 * S_2$ be the series connection of S_1 with S_2. Let $r \in R_1$, $x \in I_1^+$, $q_1 \in Q_1$, $q_2 \in Q_2$, and $t \in T$. Then

$$\beta^*_{(q_1, q_2)}(x, t) = \beta^2_{q_2}(\hat{\beta}^1_{q_1}(x, t), t)$$

and $$\beta^*_{r,t}(x) = \beta^2_{r,t}(\hat{\beta}^1_{r,t}(x)).$$

Proof: We will first derive $\bar{\delta}$.

Claim: $\bar{\delta}((q_1, q_2), x, t) = (\bar{\delta}_1(q_1, x, t), \bar{\delta}_2(q_2, \hat{\beta}^1_{q_1}(x, t), t))$.

Proof by Induction: Let $|x| = 1$. Then

$$\begin{aligned}\bar{\delta}((q_1, q_2), x, t) &= \delta((q_1, q_2), x, t) \\ &= (\delta_1(q_1, x, t), \delta_2(q_2, \lambda_1(q_1, x, t), t)) \\ &= (\bar{\delta}_1(q_1, x, t), \bar{\delta}_2(q_2, \hat{\beta}^1_{q_1}(x, t), t)).\end{aligned}$$

Assume the result is true for all x in I^+ of length n-1. Let $|x| = n$ and $x = ya$ where $|y| = n-1$. Then

$$\begin{aligned}
\bar{\delta}((q_1, q_2), ya, t) &= \delta(\bar{\delta}((q_1, q_2), y, t), a, t+n-1) \\
&= \delta((\bar{\delta}_1(q_1, y, t), \bar{\delta}_2(q_2, \hat{\beta}^1_{q_1}(y, t), t)), a, t+n-1) \\
&= (\delta_1(\bar{\delta}_1(q_1, y, t), a, t+n-1), \delta_2(\bar{\delta}_2(q_2, \hat{\beta}^1_{q_1}(y, t), t), \\
&\qquad \lambda_1(\bar{\delta}_1(q_1, y, t), a, t+n-1), t+n-1)) \\
&= (\bar{\delta}_1(q_1, ya, t), \delta_2(\bar{\delta}_2(q_2, \hat{\beta}^1_{q_1}(y, t), t), \beta^1_{q_1}(ya, t), t+n-1) \\
&= (\bar{\delta}_1(q_1, ya, t), \bar{\delta}_2(q_2, \hat{\beta}^1_{q_1}(ya, t), t)).
\end{aligned}$$

Having proved our claim the rest follows directly from the definitions. Again let $|x| = n$, $x = ya$, and $|y| = n-1$.

$$\begin{aligned}
\beta^*_{(q_1, q_2)}(x, t) &= \bar{\lambda}((q_1, q_2), x, t) \\
&= \lambda(\bar{\delta}((q_1, q_2), y, t), a, t+n-1) \\
&= \lambda((\bar{\delta}_1(q_1, y, t), \bar{\delta}_2(q_2, \hat{\beta}^1_{q_1}(y, t), t)), a, t+n-1) \\
&= \lambda_2(\bar{\delta}_2(q_2, \hat{\beta}^1_{q_1}(y, t), t), \lambda_1(\bar{\delta}_1(q_1, y, t), a, t+n-1), t+n-1) \\
&= \lambda_2(\bar{\delta}_2(q_2, \hat{\beta}^1_{q_1}(y, t), t), \beta^1_{q_1}(ya, t), t+n-1) \\
&= \bar{\lambda}_2(q_2, \hat{\beta}^1_{q_1}(ya, t), t) \\
&= \beta^2_{q_2}(\hat{\beta}^1_{q_1}(x, t), t)
\end{aligned}$$

This establishes the first equation. The second equation is an immediate consequence of this one.

We are now ready to define our notion of on-line diagnosis. This concept involves an external detector D (assumed to be fault-free) and a time-delay k within which any error produced by a fault must be detected. More precisely, let (M, F) be a machine with faults, D a machine with $R_D = R$ and $Z \subseteq I_D$, and k a nonnegative integer. Then

Definition 15

(M, F) is (D, k)-diagnosable if

i) the behavior of $M * D$ is the constant 0 function for each initial reset $r \in R$;

ii) for each $f \in F$ the system $M^f * D$ is such that if (f, r, x) is a minimal γ-error then $\hat{\beta}^*_r(xy) \neq 0^{|xy|}$ for all $y \in I^*$ with $|y| = k$.

More generally, if $\mathcal{D}$ is a set of machines then (M, F) is ($\mathcal{D}$, k)-diagnosable if there exists a D in $\mathcal{D}$ such that (M, F) is (D, k)-diagnosable.

Note that i) implies $0 \in Z_D$, the output alphabet for D. Each $z \in Z_D$ other than 0 is called a fault-detection signal. The choice of the symbol "0" to indicate that the machine M is operating properly is purely for notational convenience. In general we could let any subset of Z_D indicate proper operation and let the complement of this set in Z_D be the set of fault-detection signals. In a practical application this choice would depend on the design constraints on the

detector.

The two conditions in this definition can be paraphrased as:

i) The detector should never emit a fault detection signal if the machine that it is monitoring is fault-free.

ii) The detector must emit a fault detection signal within k time steps of the occurrence of the first γ-error produced by the faulty machine, regardless of the input after the error.

Thus, D observes the output of M^f and must make a decision based on this observation as to whether M^f has produced a γ-error. This decision may take some time -- thus the parameter k. The complexity of D is a measure of the difficulty of this decision. Note that the detector takes no part in the computation of the output of M.

The on-line diagnosis problem can now be stated as:

Given a machine $\tilde{M}$, a class of faults F, a class of machines $\mathcal{D}$, and a delay k find an (economical) d-realization M of $\tilde{M}$ such that (M, F) is $(\mathcal{D}, k)$-diagnosable.

Two major types of questions that will be of interest to us are questions of existence and economy. Questions of existence will be of the form: Given $\tilde{M}$, F, $\mathcal{D}$, and k does there exist an M such that M d-realizes $\tilde{M}$ and (M, F) is $(\mathcal{D}, k)$-diagnosable? Questions of economy will be of the form: Given that $(\tilde{M}, \tilde{F})$ is $(\tilde{\mathcal{D}}, \tilde{k})$-diagnosable can we discover an M such that M d-realizes $\tilde{M}$ and (M, F) is $(\mathcal{D}, k)$-diagnosable where $\mathcal{D}$

is more restricted than $\tilde{\mathcal{D}}$ and/or $k<\tilde{k}$. In answering questions such as these we seek methods for designing machines with these properties.

Other fundamental questions are: What time-space tradeoffs are possible between the complexity of D and the magnitude of the time-delay k? The detector, since it is fault-free, can be considered as the "hard-core" in this model. Thus, in our last question we are inquiring as to the effect of the complexity of the "hard-core" on the complexity of the total machine-detector configuration.

In the next section we will present some results which will begin to answer these questions. We finish this section with two definitions which will distinguish two special types of diagnosis.

Definition 16

(M, F) is $(\mathcal{D}, k)$-detectable if it is $(\mathcal{D}, k)$-diagnosable and each $f \in F$ either is not tolerated or is improper.

Definition 17

(M, F) is (k)-self-diagnosable if (M, F) is (D, k)-diagnosable where D is the trivial machine which implements the projection P_2. I.e., the augmented output of M serves as the output of the detector.

Example 9

Suppose that a d-realization of M_1 (see example 1) is desired which is (0)-self-diagnosable for the class of faults F which is caused by any failure which affects one delay element. M_5 as

represented by Figure 14 and implemented as shown in Figure 15 is such a realization.

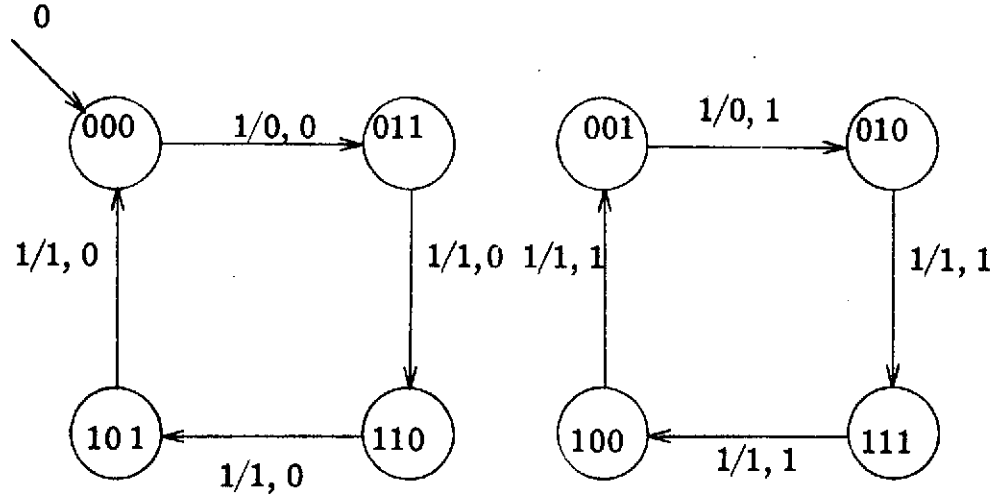


Figure 14 Machine M_5

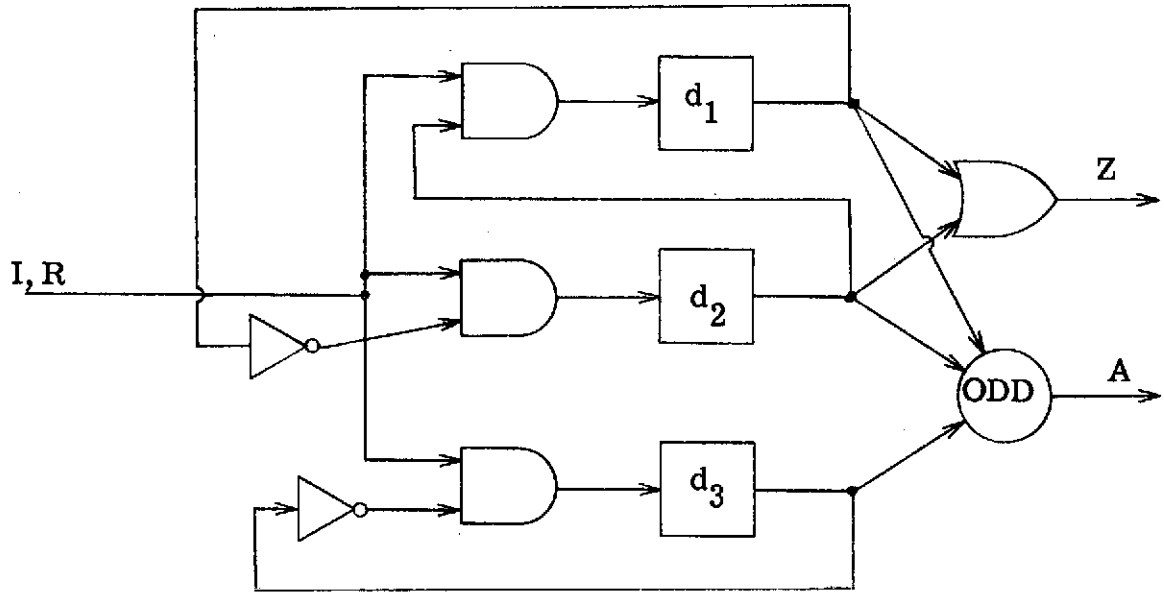


Figure 15 Circuit for Machine M_5

(M_5, F) is (0)-self-diagnosable because the added delay, d_3, acts as a parity bit and thus any erroneous value on the output of any of the

delays can be detected by the "ODD" gate which produces a 1 output if and only if the parity of its inputs is odd.

7. Preliminary Results

Here we present a potpourri of results which will begin to answer some of the general questions we have posed and which will help us to understand the nature of diagnosis as we have defined it. The first result of this section shows us that if we allow the detector to become as complex as the system it is observing then we have, in effect, created an oracle which can diagnose nearly every fault.

Theorem 12

Let $\tilde{M}$ be any machine and let $M \rho_d \tilde{M}$ where M is the machine formed from $\tilde{M}$ by augmenting the output with a copy of the input. Let F be any set of faults which are α-tolerated by M, and let $\mathcal{S}$ be the unrestricted class of all machines. Then (M, F) is $(\mathcal{S}, 0)$-diagnosable.

Proof: Let D be a copy of $\tilde{M}$ along with an equivalence gate which produces a 0 if and only if the principle output of M is identical with the result as computed by the copy of $\tilde{M}$ in D. Clearly (M, F) is (D, 0)-diagnosable. Pictorially, we can view M * D as in Figure 16.

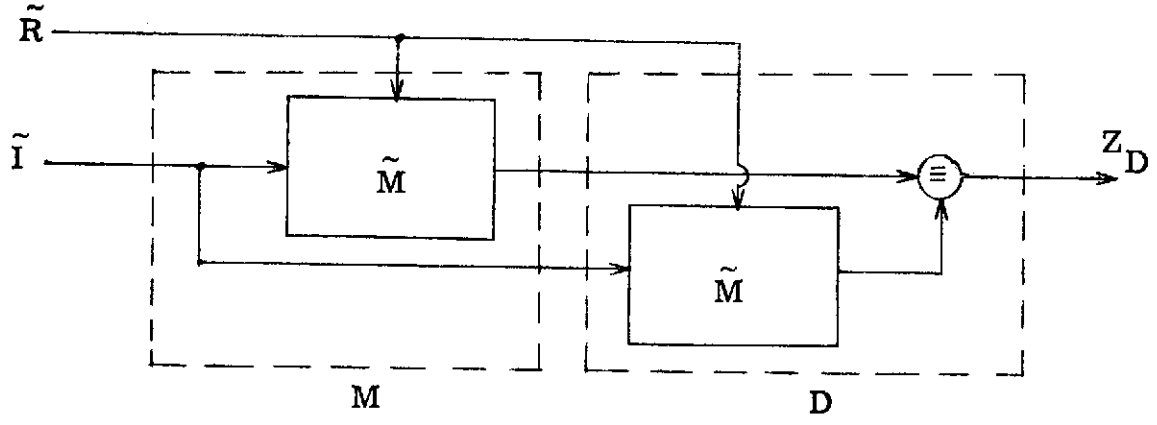


Figure 16 Diagnosis Via Duplication in the Detector

This result clearly indicates that any further result of interest must involve a limitation on the complexity of D and/or the amount or type of output augmentation allowed. This motivates the following extension of our definition of diagnosis. Let M be an output-augmented machine with $Z \subseteq Z_P \times Z_A$, and let n be a positive integer. Then

Definition 18

(M, F) is $(\mathcal{D}, k, n)$-diagnosable if (M, F) is $(\mathcal{D}, k)$-diagnosable and $|Z_A| \leq n$.

A result similar to Theorem 12 can be obtained by drawing the dashed lines in Figure 16 in a different manner as shown in Figure 17.

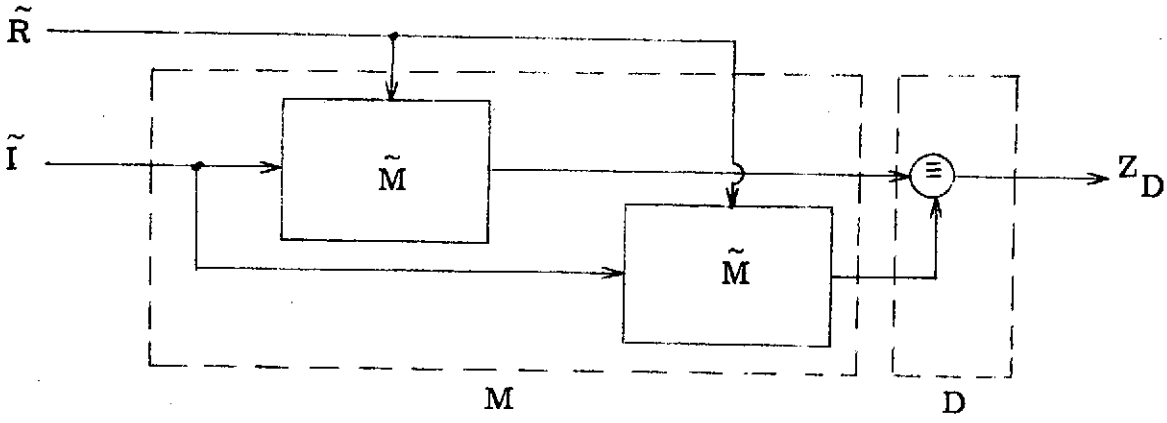


Figure 17 Diagnosis Via Duplication in the Realization

This situation is more realistic than the previous one for now faults which may affect either or both copies must be taken into account. However, this is still a powerful diagnosis technique since clearly any fault which affects only one copy of $\tilde{M}$ and many which affect both copies will be diagnosable.

The next result will help us to see the relationship between fault diagnosis and fault tolerance.

Theorem 13

Let M be a machine and F a class of faults such that F is γ-tolerated by M. Then (M, F) is $(D_0, 0)$-diagnosable where D_0 is the trivial machine which produces a constant 0 output.

Proof: Condition i) is clearly satisfied and condition ii) is trivially satisfied since if M γ-tolerates F then f can cause no γ-errors for any f in F.

The decision in this case can be trivially made since no γ-errors are ever produced. The situation for tolerated faults is not so simple as this result may seem to indicate for it must be remembered that γ-tolerated does not imply tolerated and thus a γ-tolerated fault could be detected through an error which only showed up in the augmented output.

We will now develop some results concerning diagnosis which are analogous to Corollaries 6.1 and 6.2 and to Theorems 7 through 10. Let D be a detector for a machine M. It will often be the case that the second coordinate of the state of $M * D$ can be uniquely determined from the first coordinate. In particular, this is always the case when $|Q_D| = 1$. More formally, the series connection of M_1 with M_2 is synchronized if there exists a function $h: Q_1 \rightarrow Q_2$ such

that for each (q_1, q_2) in the reachable part of $M_1 * M_2$, $h(q_1) = q_2$. Such a function is called the synchronizing function of $M_1 * M_2$ and it must satisfy $h(\rho_1(r)) = \rho_2(r)$ for each r in R. We can now state the counterpart of Corollary 6.1.

Theorem 14

Let M be an ℓ-reachable machine and let D be a detector for M such that $M * D$ is synchronized. Suppose that $(M, \{f_o, \ldots, f_\ell\})$ is (D, k)-diagnosable. Then $(M, \{f_o, f_1, \ldots\})$ is (D, k)-diagnosable.

Proof: Condition i) of Definition 15 is immediately satisfied. Let $f_i \in [f]$ with $i \geq 0$, and let $\gamma_r^{f_i}(x)$ be a minimal γ-error. Since Theorem 6 applies to γ-errors as well as to errors (β-errors) there exists an equivalent γ-error $\gamma_s^{f_j}(y)$ with $0 \leq j \leq \ell$. Since $\gamma_r^{f_i}(x)$ is minimal it follows that $\gamma_s^{f_j}(y)$ is also minimal.

Since f_j is diagnosed by D we know that $M^{f_j} * D$ will produce a nonzero output sequence for every input sequence yu with $|u| = k$ if started with initial reset s. We need only show that $M^{f_i} * D$ with initial reset r and any input sequence xu will do the same.

Let μ^j, μ^i, and β^* represent the behavior functions of $M^{f_j} * D$, $M^{f_i} * D$, and $M * D$ respectively. Let $x = x_1 z$ and $y = y_1 z$ where $|x_1| = i$ and $|y_1| = j$. Since the γ-errors are equivalent $\overline{\delta}(\rho(r), x_1) = \overline{\delta}(\rho(s), y_1)$. Say $\overline{\delta}(\rho(r), x_1) = q$. Thus both M^{f_i} and M^{f_j} will be in

state $\theta(q)$ at times i and j respectively. Let $h: Q \rightarrow Q_{D_{f_j}}$ be the synchronizing function of $M * D$. Then both $M^{f_i} * D$ and $M^{f_j} * D$ will be in state $(\theta(q), h(q))$ at times i and j respectively. Now since D is time invariant and since $\beta^i_{\theta(q)}(w, i) = \beta^j_{\theta(q)}(w, j)$ for all w in I^+ it follows from Theorem 11 that

$$\hat{\mu}^i_{(\theta(q), h(q))}(zu, i) = \hat{\mu}^j_{(\theta(q), h(q))}(zu, j).$$

We know $\hat{\mu}^j_S(yu) \neq 0^{|yu|}$ and clearly the nonzero symbol cannot be produced prior to time j. Therefore $\hat{\mu}^j_{(\theta(q), h(q))}(zu, j) \neq 0^{|zu|}$ for all $u \in I^+$ with $|u| = k$. This implies $\hat{\mu}^i_{(\theta(q), h(q))}(zu, i) \neq 0^{|zu|}$ and hence $\hat{\mu}^i_r(xu) \neq 0^{|xu|}$. Therefore $(M, \{f_o, f_1, \ldots\})$ is (D, k)-diagnosable.

Corollary 14.1

Let M be a machine with reachable part P and let D be a detector for M such that $M * D$ is synchronized. Suppose that $\rho(R) = P$ and that (M, f_o) is (D, k)-diagnosable. Then $(M, \{f_o, f_1, \ldots\})$ is (D, k)-diagnosable.

Proof: $\rho(R) = P$ implies M is 0-reachable. Apply Theorem 14.

Our next two results are analogous to Theorems 7 and 8.

Theorem 15

Let $f = (M', \tau, \theta)$ be a fault of M and suppose that (M, f_j) is (D, k)-diagnosable for some $j < 0$. Then $(M, \{\ldots, f_{-2}, f_{-1}\}$ is (D, k)-diagnosable.

Proof: By Theorem 7, $\beta_r^{f_i}(x) = \beta_r^{f_j}(x)$ for all $i, j < 0$. The result is an immediate consequence of this fact.

Theorem 16

Let $f = (M', \tau, \theta)$ be a fault of M such that $\rho'(r) = \theta(\rho(r))$ for all $r \in R$. Suppose that (M, f_j) is (D, k)-diagnosable for some $j \leq 0$. Then $(M, \{\ldots, f_{-1}, f_0\})$ is (D, k)-diagnosable.

Proof: By Theorems 7 and 8, $\beta_r^{f_i}(x) = \beta_r^{f_j}(x)$ for all $i, j \leq 0$.

Combining Theorems 14 and 15 yields

Theorem 17

Let M be an ℓ-reachable machine and let D be a detector for M such that $M * D$ is synchronized. Let $f = (M', \tau, \theta)$ be a fault of M and suppose that $(M, \{f_{-1}, f_0, \ldots, f_\ell\})$ is (D, k)-diagnosable. Then $(M, [f])$ is (D, k)-diagnosable.

We terminate this line of development by stating the combination of Corollary 14.1 with Theorem 16.

Theorem 18

Let M be a machine with reachable part P and suppose that $\rho(R) = P$. Let D be a detector for M such that $M * D$ is synchronized. Let $f = (M', \tau, \theta)$ be a fault of M such that $\rho'(r) = \theta(\rho(r))$ for all $r \in R$. If (M, f_j) is (D, k)-diagnosable for some $j \leq 0$ then $(M, [f])$ is (D, k)-diagnosable.

The following result shows that under some conditions if the output is not allowed to be augmented then there is a restriction on the detector which indicates that diagnosis will generally be difficult.

Theorem 19

Let M be a machine and f a fault of M which is not tolerated. Supposethat (M, f) is (D, k, 1)-diagnosable, and that $\lambda(P, I) = Z$ where P is the reachable part of M. Then $|Q_D| > 1$.

Proof: If $|Q_D| = 1$ then the output of D at any time depends only on its input at that time. Since M can product any symbol in Z the output of D must be 0 for each input or we would contradict the requirement that the behavior of M * D is the zero function. But f is not tolerated and (M, F) is (D, k)-diagnosable. Therefore D must be able to produce a nonzero output. Contradiction.

The reason for stating this next result is simply to make note of a limitation of self-diagnosis -- namely that there are some faults (those which cause γ-errors but which also cause the fault detection signal to be stuck-at-0) that can never be self-diagnosed.

Theorem 20

Let (M, F) be (k)-self-diagnosable. Then F contains no fault f which is not γ-tolerated and for which $\alpha_r^f = 0$ for all $r \in R$.

Proof: Obvious.

Note that any fault which only affects the reset mechanism is tolerated, and thus is diagnosable, if it occurs at or after time 0. On the other hand if such a fault occurs before time 0 it may be relatively difficult to diagnose. More precisely,

Theorem 21

Let $f = (M', \tau, \theta)$ be a fault of a machine M where $\tau < 0$ and $M' = (I, Q, Z, \delta, \lambda, R, \rho')$. Suppose that (M, f) is (D, k)-diagnosable and that there is an $r \in R$ and $x \in I^+$ such that $\hat{\gamma}^f_r(x)$ is a γ-error with $\rho'(r) \in P$, the reachable part of M. Then $|Q_D| > 1$.

Proof: Assume $|Q_D| = 1$. Then the behavior of $M^f * D$ will be $\lambda_D(\beta^f_r(x))$ where $\lambda_D: I_D \longrightarrow Z_D$ is the function realized by D. Thus $\lambda_D(\beta^f_r(z)) \neq 0$ for some $z \in I^+$. But $\beta^f_r(z) = \beta_{\rho'(r)}(z) = \beta_p(z)$ where $p = \rho'(r) \in P$. Now $p \in P$ implies that there exist $m \in R$ and $u \in I^*$ such that $p = \bar{\delta}(\rho(m), u)$. Thus

$$\beta^f_r(z) = \beta_{\bar{\delta}(\rho(m), u)}(z) = \beta_m(uz).$$

Now

$$\lambda_D(\beta^f_r(z)) \neq 0 \text{ implies that } \lambda_D(\beta_m(uz)) \neq 0.$$

But this contradicts the hypothesis that (M, f) is (D, k)-diagnosable. Hence $|Q_D| > 1$.

8. Possibilities for Further Investigation

In this report we have taken a fresh look at on-line diagnosis from a theoretical point of view. Our first observation was that conventional models were not suitable for studying this problem and consequently we introduced the notion of a resettable time-varying system. With this as our basic model the notions of a fault as a transformation of a system S into another system S' at a time τ, and of the result of the fault as a system which looks like S up to time τ and like S' thereafter came very naturally. The companion notions of fault tolerance and errors were then introduced and in Section 6 we completed our formal model with the definition of $(\mathcal{D}, k)$-diagnosable. In this section we also made the first formal statement of the on-line diagnosis problem and we outlined some of the questions that will need to be answered to adequately solve this problem.

In Section 7 we made a start at answering some of these questions and at understanding the nature of on-line diagnosis. However, we have just begun to scratch the surface of the problem and much more work remains to be done. Further work could be carried out along the lines presented below.

Except for some of the examples and for the rudimentary structure introduced by output augmentation we have been dealing with abstract (i.e., totally unstructured) systems. Such an approach is good for developing formally the concepts involved in our theory but some of the questions raised can best be studied in a more

structured environment. One reason for this is that with a structured system we can consider the causes of faults. For example, given an abstract system it makes no sense to speak of the set of faults caused by component failures of a certain type or by bridging failures. However, given a structured representation of a system (e.g., a circuit diagram) we can discuss these and other types of failures (causes) and determine the resulting faults (effects).

There are many different structural levels that could prove useful to a further investigation into the theory of on-line diagnosis. Three levels which we believe will be important are: the binary state-assigned level, the logical circuit level, and the subsystem-network level. These levels and the basis for their potential usefulness are explained in the following paragraphs.

A machine M is said to be binary state-assigned if $Q = \{0, 1\}^n$ for some positive integer n. Given such a machine we can speak of stuck-at-0 and stuck-at-1 and any other type of memory failure. The faults corresponding to these failures can be enumerated and comparisons can be made between various schemes for diagnosing these faults. Memory faults have been studied before in other contexts (see [21] and [22] for example) and they are an important class of faults for an number of reasons. As we have seen, only a limited amount of structure is needed to discuss them. Thus memory faults can be analyzed before the circuit design of the machine

is complete. Also, it is memory which distinguishes truly sequential system from purely combinational (one-state) systems. Combinational systems are inherently easier than sequential systems to analyze and a number of techniques for the on-line diagnosis of such systems are known (see [8] and [9] for example).

A system possesses structure at the logical circuit level if a representation of the system is given in terms of a logical circuit composed of primitive logical elements. These may be of the AND-OR variety, threshold elements, or any similar elements of a "building block" nature depending upon the technology being considered. This level is useful for investigating failures in the primitive components. The circuit in Figure 2 is an example of a structural representation at this level and the failure of this circuit discussed in example 2 is a simple example of the analysis that can be conducted at this level.

The subsystem-network level is the most general of these three levels. In general, any system which is represented in terms of a network of subsystems is said to have the subsystem-network level of structure. At this level we could study the problem of implementing on-line diagnosis on a whole computer whereas with the other levels the emphasis would be on diagnosing one module. Note that in our definition of diagnosis the detector is not constrained to give simply a yes-no response. It could also provide extra information for use in automatic fault location. Thus at this level we could study the

problem of which subsystems must be explicitly observed by the detector to achieve some desired fault location property.

One problem that cannot be naturally studied with our model at any structural level is the problem of automatic reconfiguration of the system under the control of the detector. To study this problem our model would have to allow for feedback from the detector to the system it is observing and at the present time this is not allowed.

References

[1] Chang, H. Y., E. G. Manning, and G. Metze, Fault Diagnosis of Digital Systems, John Wiley and Sons, Inc., New York, 1970.

[2] Carter, W. C., D. C. Jessep, W. G. Bouricius, A. B. Wadia, C. E. McCarthy, and F. G. Milligan, "Design Techniques for Modular Architecture for Reliable Computer Systems," IBM Res. Rept. RA 12, March 1970.

[3] Avizienis, A., G. C. Gilley, F. P. Mathur, D. A. Rennels, J. A. Rohr, and D. K. Rubin, "The STAR (Self-Testing and Repairing) Computer: An Investigation of the Theory and Practice of Fault-tolerant Computer Design," IEEE Trans. on Computers, Vol. C-20, No. 11, Nov. 1971, pp. 1312-1321.

[4] Downing, R. W., J. S. Nowak, and L. S. Tuomenoksa, "No. 1 ESS Maintenance Plan," Bell System Technical Journal, Vol. 18, No. 5, Part 1, Sept. 1964, pp. 1961-2019.

[5] Carter, W. C., H. C. Montgomery, R. J. Preiss, and H. J. Reinheimer, "Design of Serviceability Features for the IBM System/360," IBM Journal, Vol. 8, No. 2, 1964, pp. 115-126.

[6] Eckert, J. P., "Checking Circuits and Diagnostic Routines," Instruments and Automation, Vol. 30, August 1957, pp. 1491-1493.

[7] Friedman, A. D., and P. R. Menon, Fault Detection in Digital Circuits, Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

[8] Kautz, W. H., "Automatic Fault Detection in Combinational Switching Networks," Stanford Research Institute Project No. 3196, Technical Report 1, Menlo Park, California, April 1961.

[9] Sellers, F. F., M. Hsiao, and L. W. Bearnson, Error Detection Logic for Digital Computers, McGraw-Hill, Inc., 1968.

[10] Avizienis, A., "Concurrent Diagnosis of Arithmetric Processors," Digest of the First Annual IEEE Computer Conference, Chicago, Illinois, Sept. 1967, pp. 34-37.

[11] Rao, T. R. N., "Error-Checking Logic for Arithematic-Type Operations of a Processor," IEEE Trans. on Computers, Vol. C-17, No. 9, Sept. 1968, pp. 845-849.

[12] Dorr, R. C., "Self-Checking Combinational Logic Binary Counters," IEEE Trans. on Computers, Vol. C-21, No. 12, Dec. 1972, pp. 1426-1430.

[13] Peterson, W. W., "On Checking an Adder," IBM Journal, Vol. 2, April 1958, pp. 166-168.

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

[15] Carter, W. C., and P. R. Schneider, "Design of Dynamically Checked Computers," Proc. of the IFIPS, Edinburgh, Scotland, August 1968, pp. 878-883.

[16] Meyer, J. F., and B. P. Zeigler, "On the Limits of Linearity," Theory of Machines and Computations (Edited by Z. Kohavi and A. Paz), Academic Press, New York, 1971, pp. 229-241.

[17] Leake, R. J., "Realization of Sequential Machines," IEEE Trans. on Computers (correspondence), Vol. C-17, No. 12, 1968, p. 1177.

[18] Hartmanis, J. and R. E. Stearns, Algebraic Structure Theory of Sequential Machines, Prentice-Hall, Englewood Cliffs, New Jersey, 1966.

[19] Zeigler, B. P., "Toward a Formal Theory of Modeling and Simulation: Structure Preserving Morphisms," Journal of the ACM, Vol. 19, No. 4, Oct. 1972, pp. 742-764.

[20] Meyer, J. F., "A General Model for the Study of Fault Tolerance and Diagnosis," Proc. of the 6th Hawaii International Symposium on System Sciences, January 1973, pp. 163-165.

[21] Meyer, J. F., "Fault Tolerant Sequential Machines," IEEE Trans. on Computers, Vol. C-20, No. 10, Oct. 1971, pp. 1167-1177.

[22] Yeh, K., "A Theoretic Study of Fault Detection Problems in Sequential Systems," Systems Engineering Laboratory Technical Report No. 64, The University of Michigan, Ann Arbor, 1972.