ZYCAD
CORPORATION ®

November 15,
1988.

1380 Willow Road, Menlo Park, California 94025

(415) 321-8574
Fax: (415) 321-3167

Ramesh Narayanaswamy
Project Leader
Zycad Corp.

Publications Office
Department of Computer Science
University of Waterloo
Waterloo
Ontario
Canada N2L3G1

Sir:

I would like a copy of the following Technical Report from the your Computer Science ~~Dept,~~ sent to my address given above.

1. J. A. Brzozowski, **A Model of Sequential Machine Testing**, Research Report - **CS-88-12.**

Thanking you in advance,

Sincerely Yours,

sent with compliments S.D. Nov. 24/88

UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT

UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT

A Model for Sequential Machine Testing

J.A. Brzozowski
H. Jürgensen

Research Report
CS-88-12

April, 1988

A Model for Sequential Machine Testing[1]

J. A. Brzozowski[2]

H. Jürgensen[3]

Abstract: A mathematical framework for the testing and diagnosis of sequential machines is developed. A very general fault model is used in which a faulty machine is represented as a sequential machine, possibly with state and output sets different from those of the good machine. The set of all possible behaviours is conveniently represented by a non-deterministic finite automaton, called the fault schema. A deterministic finite automaton, called the fault observer, describes the process by which one gains information from the observation of the responses to test sequences. A non-deterministic automaton is derived from the fault-observer; this automaton, called the non-deterministic tester, models all possible conclusions that could be drawn from observing the circuit under test. This model is suitable to serve as a generator of test sequences. Moreover, it can be used for describing testing with deterministic as well as with random sequences. Probabilities are associated with the fault schema and with the tester. These, together with a stochastic source of input symbols, provide a probabilistic test model. As a particular application we consider the testing and diagnosis of random-access memories by random test sequences. Our model generalizes the work by David et al. on the calculation of the length of a random test sequence required to guarantee that the probability of detection of a fault exceeds a prescribed threshold.

1. Introduction

Present day integrated circuits and larger digital networks involve extremely sophisticated processes, are very complex, and contain hundreds of thousands of components. There are numerous causes of failure [Abr], and careful

[1] This work was supported by the Natural Sciences and Engineering Research Council of Canada, Grants A0871 and A0243. A preliminary version of this paper has been presented at the Second Workshop on New Directions in IC Testing, Winnipeg, Manitoba, April 1987.

[2] Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1.

[3] Department of Computer Science, The University of Western Ontario, London, Ontario, Canada, N6A 5B7.

testing is an indispensable step in the production of a system. The generation of tests, their application, and the interpretation of the test results is a very time-consuming and expensive process.

In general, it is practically impossible to devise tests which would account for all the faults that may occur. Thus the usual approach is to formulate an idealized and simplified fault model [Abr]. For example, one may assume that a wire may become permanently stuck-at-0, that is, the signal on the wire is always logical 0; such an effect may arise due to different physical failures, such as a short to ground or a broken connection. A good fault model should account for a very large percentage of the likely failures.

In "deterministic testing" one analyses the circuit behaviour in the presence of a fault and attempts to construct a test sequence which will detect that fault. However, even the problem of deciding whether such a test sequence exists is NP-complete in general [Fuj]. Also, even if a test sequence is known, it may be difficult or impossible to apply it if the circuit under test is embedded in a large system.

An alternative approach is to apply random or pseudo-random test sequences. Now, however, one faces the following type of question: Suppose the circuit under test has behaved properly for a large number of random test vectors; how confident can one be that the circuit is fault-free? It is clear that a precise mathematical model is required before one can perform the analysis which will answer this question.

This paper was inspired by [Fu1, Fu2] which describe random testing of memories. In these papers, the length of a random test sequence required to guarantee the detection of a fault with a probability exceeding some threshold is calculated with the aid of certain Markov chains. In the present paper we establish a rigorous mathematical framework in which such problems can be attacked. In particular, we develop general algorithms for constructing the required Markov chains. We also extend these ideas to fault diagnosis.

The paper is structured as follows: A summary of the results is given in Section 2. Section 3 contains the basic definitions and terminology. A fault schema describing all possible correct and faulty behaviours is defined in Section 4. The process of deducing knowledge from test results is captured by the definition of an observer in Section 5. In Section 6, the concepts of testing and diagnosis are formalized. Next, after introducing the basic terminology concerning probabilistic automata and information sources in Section 7, the concepts of observer, testing, and diagnosis are embedded in a probabilistic test model in Section 8. Section 9 contains some concluding remarks.

2. Summary of Results

We develop a precise mathematical model of the testing and diagnosis of sequential machines. The model has the following properties:

- It is a generalization of the work of Hennie from the 1960's on deterministic diagnosis and, at the same time, of the recent work of David et al. on probabilistic testing, that is, testing with random sequences of test vectors.
- The model uses the well-established notation, terminology, and basic results from standard theory of finite automata.
- The main feature of the model is the "observer," which uses the knowledge of the applied input test vector and the resulting output vector. This observer can be constructed algorithmically from the description of the given correct machine and its likely faulty versions.
- The observer deduces the maximum possible information from each observation, namely the information concerning the machine type, its starting state and its present state. Thus the model is capable of representing the most detailed diagnosis process possible.
- A general method is developed for deriving specialized smaller versions of the observer, if only partial information is required. Thus, for example, we may wish to identify only the machine type, or to detect only whether or not the machine is faulty. All such questions are handled in a uniform framework.
- We show how a finite automaton may be derived from the observer to serve as the minimal acceptor of test sequences or a minimal test sequence generator.
- The basic deterministic observer is naturally transformed to a probabilistics observer which is described in terms of standard Markov chain theory.
- A number of concepts presented only informally in the work of Fuentes et al. are formalized in our model.
- For one particular case of a faulty memory cell we prove analytically a conjecture of Fuentes et al. concerning the length of a random test sequence required to guarantee that the probability of detection exceeds a given threshold.

We believe that the framework developed here constitutes a solid foundation for future research in this area.

3. Basic Notions

In this section we introduce notation and review several fundamental notions. For further details, the references [Wo, St] should be consulted.

An *alphabet* is a finite, non-empty set. Let X be an alphabet; then X^* denotes the set of words over X, including the empty word ε, and $X^+ = X^* \setminus \{\varepsilon\}$. For a word $w \in X^*$, $|w|$ denotes the length of w. A *language* over X is a subset of X^*.

A *deterministic finite semi-automaton* is a triple (Q, X, δ) with Q a finite, non-empty set, the *set of states*, X an alphabet, the *input alphabet*, and with $\delta : Q \times X \to Q$, the *transition function.* As usual, δ is extended to a function of $Q \times X^*$ into Q by requiring

$$\delta(q, w) = \begin{cases} q, & \text{if } w = \varepsilon, \\ \delta(\delta(q, v), x), & \text{if } w = vx \text{ with } x \in X,\ v \in X^*. \end{cases}$$

A *deterministic finite acceptor* is a quintuple $A = (Q, X, \delta, q_0, Q_F)$ such that (Q, X, δ) is a deterministic finite semi-automaton, $q_0 \in Q$, and $Q_F \subseteq Q$. Then q_0 is the *initial state*, and Q_F is the set of *final* or *accepting states.* The *language accepted by* A is the set

$$L(A) = \{w \mid \delta(q_0, w) \in Q_F\}.$$

A *deterministic finite Mealy automaton* is a quintuple $A = (Q, X, Y, \delta, \lambda)$ where (Q, X, δ) is a deterministic finite semi-automaton, Y is an alphabet, the *output alphabet*, and $\lambda : Q \times X \to Y$ is the *output function.*

A *non-deterministic finite semi-automaton* is a triple (Q, X, η) with Q and X as above; the *(non-deterministic) transition function* η is a mapping

$$\eta : Q \times X \to 2^Q$$

which, for every state $q \in Q$ and every input symbol $x \in X$, determines the set $\eta(q, x)$ of potential successor states. As usual, η is extended to a mapping of $2^Q \times X^*$ into 2^Q. A *non-deterministic finite acceptor* is a quintuple $A = (Q, X, \eta, Q', Q_F)$ such that (Q, X, η) is a non-deterministic finite semi-automaton, $Q' \subseteq Q$ is the *set of initial states* of A, and $Q_F \subseteq Q$ is the set of *final* or *accepting* states. The *language accepted by* A is the set

$$L(A) = \{w \mid \eta(Q', w) \cap Q_F \neq \emptyset\}.$$

As is well-known, the accepting power of deterministic and non-deterministic finite acceptors is the same. This is usually proved by the so-called power set construction [Wo]. For a non-deterministic finite acceptor A, let $\mathcal{P}(A)$ denote the deterministic finite acceptor, obtained by the power set construction, whose state set consists of precisely those states which can be reached from the initial state.

A *non-deterministic Mealy automaton* is a quadruple (Q, X, Y, η) with Q and X as before, where Y is an alphabet, the *output alphabet*, and $\eta : Q \times X \to 2^{Q \times Y}$ is the *transition-and-output function.*

Let $\mathcal{A}(Q, X)$ and $\mathcal{A}_{\text{nd}}(Q, X)$ denote the classes of deterministic, and non-deterministic semi-automata, respectively, which have X as the input alphabet and a subset of Q as state set. Similarly, let $\mathcal{A}(Q, X, Y)$ and $\mathcal{A}_{\text{nd}}(Q, X, Y)$ denote the corresponding classes of Mealy automata whose output alphabets are subsets of Y.

In the sequel, when referring to (semi-)automata, we shall usually omit the word 'finite,' as no infinite (semi-)automata will be considered in this paper.

4. Fault Schema

A formal description of fault assumptions needs to be provided as the basis of any rigorous treatment of circuit testing. We choose to use deterministic Mealy automata as the formal tool. Not only do they seem to be the natural abstractions in this context, but they also turn out to simplify the subsequent constructions of fault analysis and test models. Similar ideas were already discussed in 1964 by Poage and McCluskey [Po] and Hennie [He2]. However, Poage and McCluskey assumed that the good and faulty circuits can all be reset to the same initial state, whereas nò such assumption is made here. Our work also generalizes the work of Hennie, as will be shown later.

The correct behaviour of the circuit under consideration will be given as a deterministic (Mealy) automaton, the "good machine." If $A_0 = (Q_0, X, Y_0, \delta_0, \lambda_0)$ is such a good machine, then Q_0 denotes the set of "physical states" of the circuit, X denotes the input alphabet or set of "actions" applied to the circuit, Y_0 is the set of outputs, δ_0 is the "correct" or intended transition function, and λ_0 is the "correct" or intended output function. This model includes the case when certain inputs do not result in an output, for example, the case of a writing operation in a memory. In such a case Y_0 would contain a special symbol, \$ say, meaning "no output," which would be the formal output symbol. A faulty version A_1 of A_0 will also be a deterministic Mealy automaton. It will have the same input alphabet X. However, its set Q_1 of physical states, its output alphabet Y_1, its transition function δ_1, and its output function λ_1 could be different from Q_0, Y_0, δ_0, and λ_0, respectively. In the sequel, we assume without special mention that there is a set Q of states such that the state sets of the good machine and of its faulty versions are subsets of Q. Similarly, we assume that there is an alphabet Y such that the output alphabets of the good machine and of its faulty versions are subsets of Y. This allows for a very simple identification of corresponding states and output symbols of different versions of a machine.

In summary, a fault is just a deviation from the good machine. Let $A_0 = (Q_0, X, Y_0, \delta_0, \lambda_0) \in \mathcal{A}(Q, X, Y)$ be a good machine. A *fault* of A_0 is a deterministic automaton $A_1 = (Q_1, X, Y_1, \delta_1, \lambda_1) \in \mathcal{A}(Q, X, Y)$ which is different from A_0.

We shall now provide three examples, or rather groups of examples, to illustrate the definitions. They address quite different issues in circuit testing. The first one concerns faults in memory cells; we demonstrate how a combination of fault types can be modelled. The second example shows how faulty read operations in a memory fit into our framework. The third example explains that faults in combinational circuits can also be adequately formulated in our model.

Example 4.1 Let $B_0 = (Q_0, X, Y_0, \delta_0, \lambda_0)$ be an automaton which describes the behaviour of a pair (i, j) of memory cells. Here $Q_0 = \{0,1\}^2$ is the set of the four possible states of the two cells. Let

$$X = \{w_0^i, w_1^i, w_0^j, w_1^j, r^i, r^j\}$$

be the input alphabet where we interpret w_k^l as writing k into cell l, and r^l as reading cell l. Let $Y_0 = \{0, 1, \$\}$ be the output alphabet, with \$ meaning "no output." The transition function δ_0 is given by Figure 1(a) where we simplify the notation by writing 01 instead of $(0,1)$, etc. The output function λ_0 is defined as follows:

$$\lambda_0((p_1, p_2), x) = \begin{cases} p_1, & \text{if } x = r^i, \\ p_2, & \text{if } x = r^j, \\ \$, & \text{if } x \in \{w_0^i, w_1^i, w_0^j, w_1^j\}. \end{cases}$$

Now consider a fault of the following type: If cell i contains 0 and 1 is written into cell i, then cell j will become 1 if it was 0. This fault type is denoted by $\uparrow i \Rightarrow \uparrow j$ in [Ful]. Let $B_1 = (Q_1, X, Y_1, \delta_1, \lambda_1)$ be this fault as shown in Figure 1(b) where $Q_1 = Q_0$ and $Y_1 = Y_0$. The output function λ_1, which is not shown in the diagram, is identical to λ_0.

The second type of fault to be considered is a stuck-at-0 fault in a single cell. Of course, the natural state set for this kind of fault would be the set $\{0,1\}$ or just $\{0\}$, if cell i is the only cell considered. However, in order to apply the fault to B_0, we define it again on the set $\{0,1\}^2$. Let $B_2 = (Q_2, X, Y_2, \delta_2, \lambda_2)$ be a stuck-at-0 fault of cell i. Then $Q_2 = \{(0,0), (0,1)\}$, $Y_2 = Y_0$, and δ_2 is defined in Figure 1(c). λ_2 is the restriction of λ_0. Let $B_3 = (Q_3, X, Y_3, \delta_3, \lambda_3)$ denote the analogous automaton for cell j stuck-at-0.

The simultaneous presence of two or more faults in a given circuit is called a *multiple fault*. This can again be represented by an automaton.

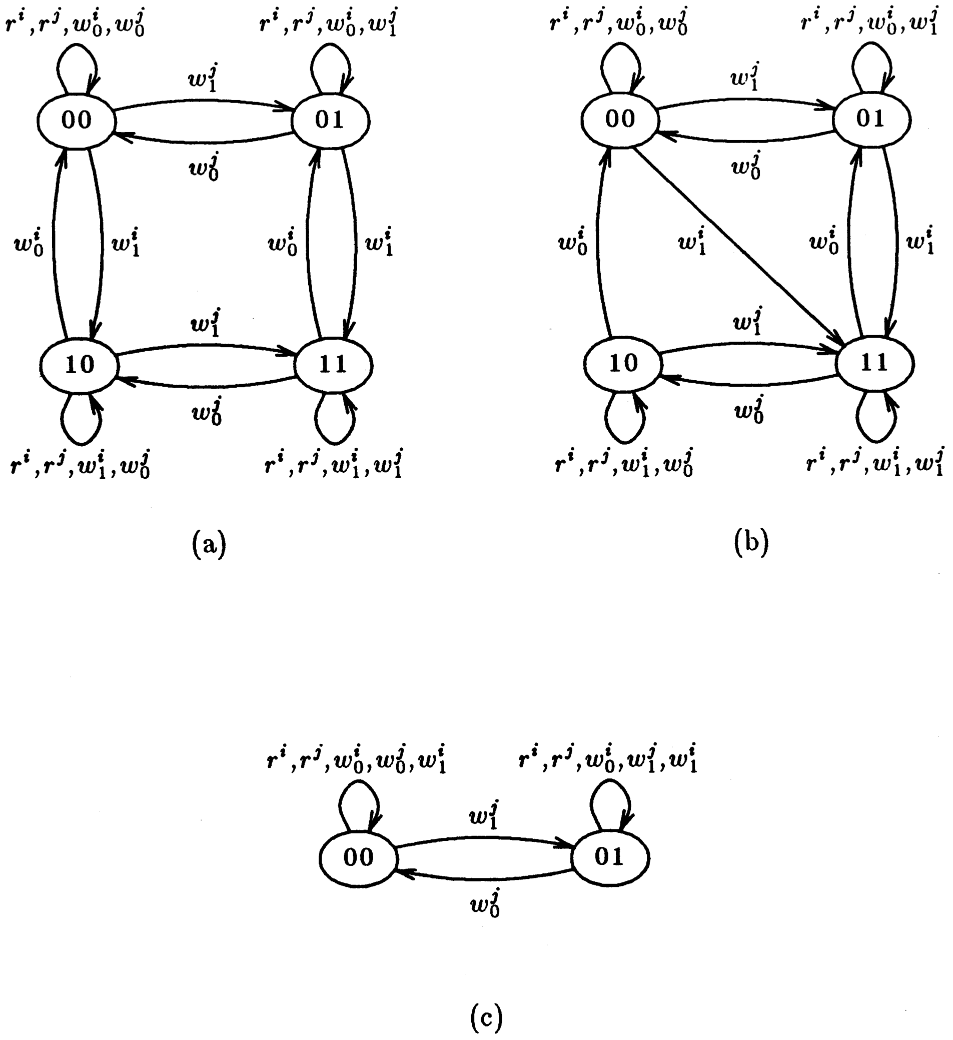


Figure 1. (a) Good machine B_0; (b) fault B_1, $\uparrow i \Rightarrow \uparrow j$; (c) fault B_2, cell i stuck-at-0.

For example, the double fault "cell i stuck-at-0" and "cell j stuck-at-0" is defined as follows: The automaton $B_4 = (Q_4, X, Y_4, \delta_4, \lambda_4)$ represents the double fault, where $Q_4 = \{(0,0)\}$, $Y_4 = \{0, \$\}$,

$$\delta_4((0,0), x) = (0,0)$$

and

$$\lambda_4((0,0), x) = \begin{cases} 0, & \text{if } x = r^i \text{ or } x = r^j, \\ \$, & \text{otherwise,} \end{cases}$$

for all $x \in X$.

It should be pointed out here that not all pairs of faults are "compatible." For example, the fault "cell i stuck-at-0" and the fault "cell i stuck-at-1" are clearly incompatible.

Let $A_0 \in \mathcal{A}(Q, X, Y)$ be a good machine. A *fault model* for A_0 is a finite family $\mathcal{F}_{A_0} = \{A_i = (Q_i, X, Y_i, \delta_i, \lambda_i) \mid i \in I\}$ of faults of A_0 where I is a finite index set. Note that some of these may be multiple faults. Assume that $0 \notin I$ and let $I_0 = I \cup \{0\}$.

With the fault model $\mathcal{F}_{A_0}$ we associate the non-deterministic automaton $F(\mathcal{F}_{A_0})$ which we call a *fault schema* and which is defined as follows: $F = (Q, X, Y, \eta)$ where $(q, y) \in \eta(p, x)$ if and only if there exists an $i \in I_0$ such that $\delta_i(p, x) = q$ and $\lambda_i(p, x) = y$. Thus the fault schema $F(\mathcal{F}_{A_0})$ expresses the potential behaviour of a circuit under the given fault assumptions. In a later section of this paper, probabilities will be added to the fault model and to the fault schema. The transition part of the function η of the fault schema for Example 4.1 is shown in Figure 2; the output part of the function η is obvious and has been omitted from the diagram.

Our second example deals with faulty read operations.

Example 4.2 The good machine B'_0 models a single memory cell as shown in Figure 3(a). B'_1 describes the fault of "inverted reading," while B'_2 illustrates a fault with "destructive reading." Observe that we write $r/0$ and $r/1$ to mean that the read operation r yields 0 and 1 as output, respectively. The diagrams of these faults are shown in Figure 3(b) and (c). The fault schema is shown in Figure 3(d).

As the next example shows, combinational circuits can also be modelled quite adequately within our framework.

Example 4.3 A combinational circuit is an automaton with a single state. The automaton B''_0 as shown in Figure 4(a) models a NOR gate. To simplify the presentation we use the following encoding of inputs: $00 \mapsto 0$, $01 \mapsto 1$, $10 \mapsto 2$, and $11 \mapsto 3$. Similar encodings are used in the sequel without special mention. The input lines of the gate are labelled a and b. The input symbol

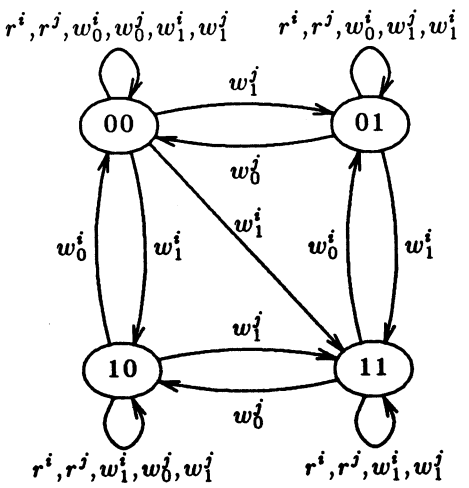

$r^i, r^j, w_0^i, w_0^j, w_1^i, w_1^j$
$r^i, r^j, w_0^i, w_1^j, w_1^i$
w_1^j
00
01
w_0^j
w_1^i
w_0^i
w_1^i
w_0^i
w_1^i
w_1^j
10
11
w_0^j
$r^i, r^j, w_1^i, w_0^j, w_1^j$
r^i, r^j, w_1^i, w_1^j

Figure 2. Fault schema $F(\mathcal{F}_{B_0})$.

2 means an input of 1 on line a and an input of 0 on line b; the symbols 0, 1, and 3 have similar interpretations.

The automaton B_1'' models the fault of line a stuck-at-0. The automata B_2'' and B_3'' describe two faults typical with CMOS realizations of gates. A CMOS realization of the NOR gate is shown in Figure 4(c). B_2'' models a stuck-open fault of n-transistor T_1, while B_3'' describes a stuck-on fault of p-transistor T_2.

In B_2'', when the input is 2 (that is, $a = 1, b = 0$), the output node c is electrically isolated and remembers its previous value due to the capacitance of the node. Thus we have two states, $c = 0$ and $c = 1$. If $a = 0$ and $b = 0$, c becomes 1; if $b = 1$, c is driven to 0; when $a = 1$ and $b = 0$, the previous state is remembered. This results in Figure 4(d).

For B_3'', the circuit behaves normally unless the input is 2 (that is $a = 1, b = 0$). Then n-transistor T_1 is closed and p-transistor T_2 is stuck-on instead of being open. Consequently, there is a conducting path from V_{DD} to ground, and the output voltage takes on some intermediate value $\times$ between V_{DD} and ground.

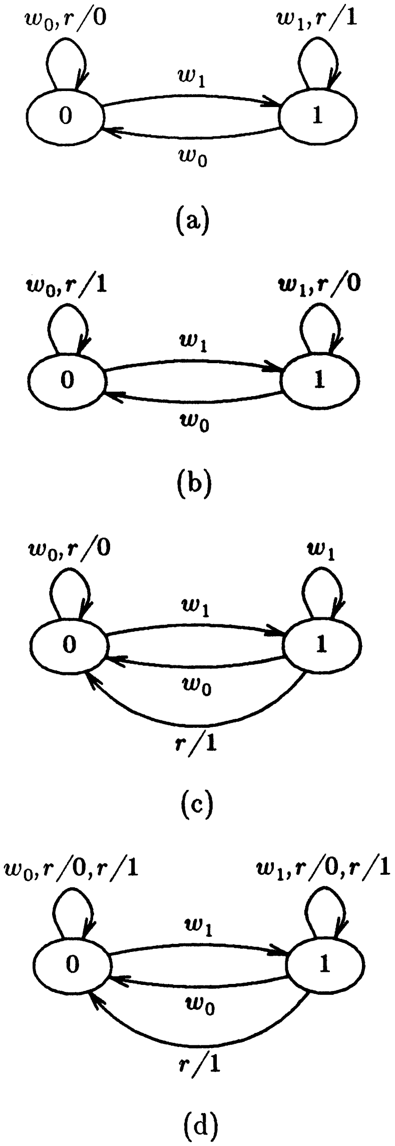


Figure 3. Faulty read operations: (a) good machine B'_0; (b) fault B'_1, inverted reading; (c) fault B'_2, destructive reading; (d) fault schema $F(\mathcal{F}_{B'_0})$.

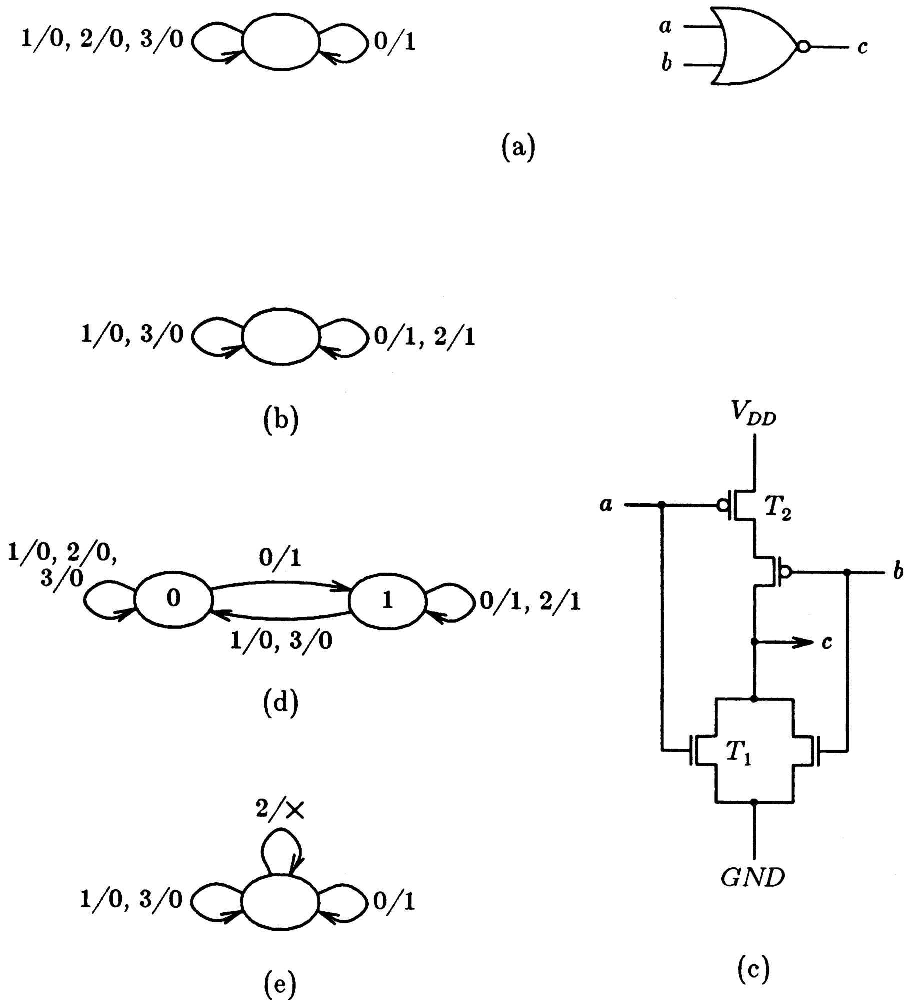


Figure 4. Combinational circuit faults: (a) the good machine B''_0, a **NOR** gate; (b) the fault B''_1, input a stuck-at-0; (c) CMOS realization of the **NOR** gate; (d) the fault B''_2, transistor T_1 stuck-open; (e) the fault B''_3, transistor T_2 stuck-on.

5. Fault Observer

In this section we provide a formal definition of the notion of fault observer. It will be used as a basis for the concepts of fault diagnosis and detection, both in deterministic and probabilistic settings.

Let $A_0 = (Q_0, X, Y_0, \delta_0, \lambda_0) \in \mathcal{A}(Q, X, Y)$ be a good machine and let $\mathcal{F} = \mathcal{F}_{A_0} = \{A_i = (Q_i, X, Y_i, \delta_i, \lambda_i) \mid i \in I\}$ be a finite family of faults of A_0. Without loss of generality we assume that $I = \{1, \ldots, n\}$. Using A_0 and $\mathcal{F}$ we define a deterministic semi-automaton Δ with starting state; Δ models the process of fault detection and fault diagnosis from observations of the behaviour of the circuit under test. Δ can be constructed algorithmically from A_0 and $\mathcal{F}$. The idea of the construction is quite simple: In order to test a given circuit A, the experimenter feeds inputs into A and observes its outputs. At the same time, he runs copies of the good and faulty machines feeding them with the same inputs and observing their outputs. Since he cannot know the initial state of A, he will start the experiment with copies of the good machine and each of the faults initialized to any of their possible states. When the output of a copy of the good machine or a fault is found to disagree with the output of A, that copy is eliminated from the contest. That is, at any given moment, the observed behaviour of A is consistent with that of all machine copies still in the contest.

The *(deterministic) fault observer*

$$\Delta = \Delta(A_0, \mathcal{F}) = (D, X \times Y, \delta, d_0)$$

is defined and interpreted as follows. Let A be a circuit which is to be tested. Δ models the process by which one arrives at the conclusion that A is correct or faulty and possibly also which of the automata A_i, $i \in I_0$, the circuit A actually is. Even its initial and present states may be identified. Some auxiliary concepts are needed before Δ can be defined.

With every deterministic finite Mealy automaton

$$A' = (Q', X, Y', \delta', \lambda')$$

with $Q' \subseteq Q$ and $Y' \subseteq Y$ one associates a deterministic finite semi-automaton $\widehat{A'} = (\widehat{Q'}, X \times Y, \widehat{\delta'})$ as follows: Let ω be a new state (not in Q) and $\widehat{Q'} = Q' \cup \{\omega\}$. Define

$$\widehat{\delta'}(q, (x, y)) = \begin{cases} \delta'(q, x), & \text{if } \lambda'(q, x) = y, \\ \omega, & \text{if } \lambda'(q, x) \neq y, \end{cases}$$

and

$$\widehat{\delta'}(\omega, (x, y)) = \omega$$

for $q \in Q'$, $x \in X$, and $y \in Y$.

For all $i \in I_0$, let $\widehat{A}_i = (\widehat{Q}_i, X \times Y, \widehat{\delta}_i)$ be the semi-automaton associated with A_i. In a state $d \in D$ we record which set of states the circuit could be in if it was any A_i started in some state q, $q \in Q_i$, $i \in I_0$. A typical state d of Δ is a tuple with components of the form $d_{i,q}$ where $d_{i,q} \in \widehat{Q}_i$. If Δ is in state d then $d_{i,q}$ is the state of A, if A happens to be A_i when started in state q. Thus, let

$$K' = \{(i,q) \mid i \in I_0,\ q \in Q_i\}$$

and

$$D = \{d \mid d_{i,q} \in \widehat{Q}_i,\ (i,q) \in K'\}.$$

Initially, the experimenter knows nothing about A, that is, A could be any of the machines A_i in any of the states $q \in Q_i$. Therefore,

$$[d_0]_{i,q} = q$$

for $(i,q) \in K'$. Here and in the sequel, if e is an expression or symbol denoting a tuple, we (may) use $[e]_{i,q}$ to denote its (i,q)-th component.

To define the transition function δ, let $d, d' \in D$. One has $d' = \delta(d,(x,y))$ for $x \in X$ and $y \in Y$ if and only if

$$d'_{i,q} = \widehat{\delta}_i(d_{i,q},(x,y))$$

holds true for all $i \in I_0$ and all $q \in Q_i$.

Example 5.1 In Figure 5 we show the good machine B_0 of a memory cell and the fault B_1 of the cell being stuck-at-0, as well as the corresponding fault observer $\Delta(B_0, \mathcal{F}_{B_0})$ where $\mathcal{F}_{B_0} = \{B_1\}$. In the notation for states of the observer, a semicolon separates the different machines.

In Example 5.1 and Figure 5 one sees that the observer may become quite large even for a very simple fault model. Indeed, the only a priori bound on the number of states of the observer is m^{mn+m} where $m = |Q| + 1$ and $n = |I|$. However, two comments are important at this point:

- if all the information about the potential identity of the circuit A which is being tested (the machine type, the initial state, and the current state) is required, then the observer cannot be made any smaller; on the other hand—as is to be explained in the next section—if the goal of testing can be achieved with less information, then a modified version of the standard reduction theory of finite automata can be applied to yield smaller models;
- our model allows for an algorithmic construction of the observer; this implies that quite large observers may be still manageable as long as the construction is "handed over" to a computer.

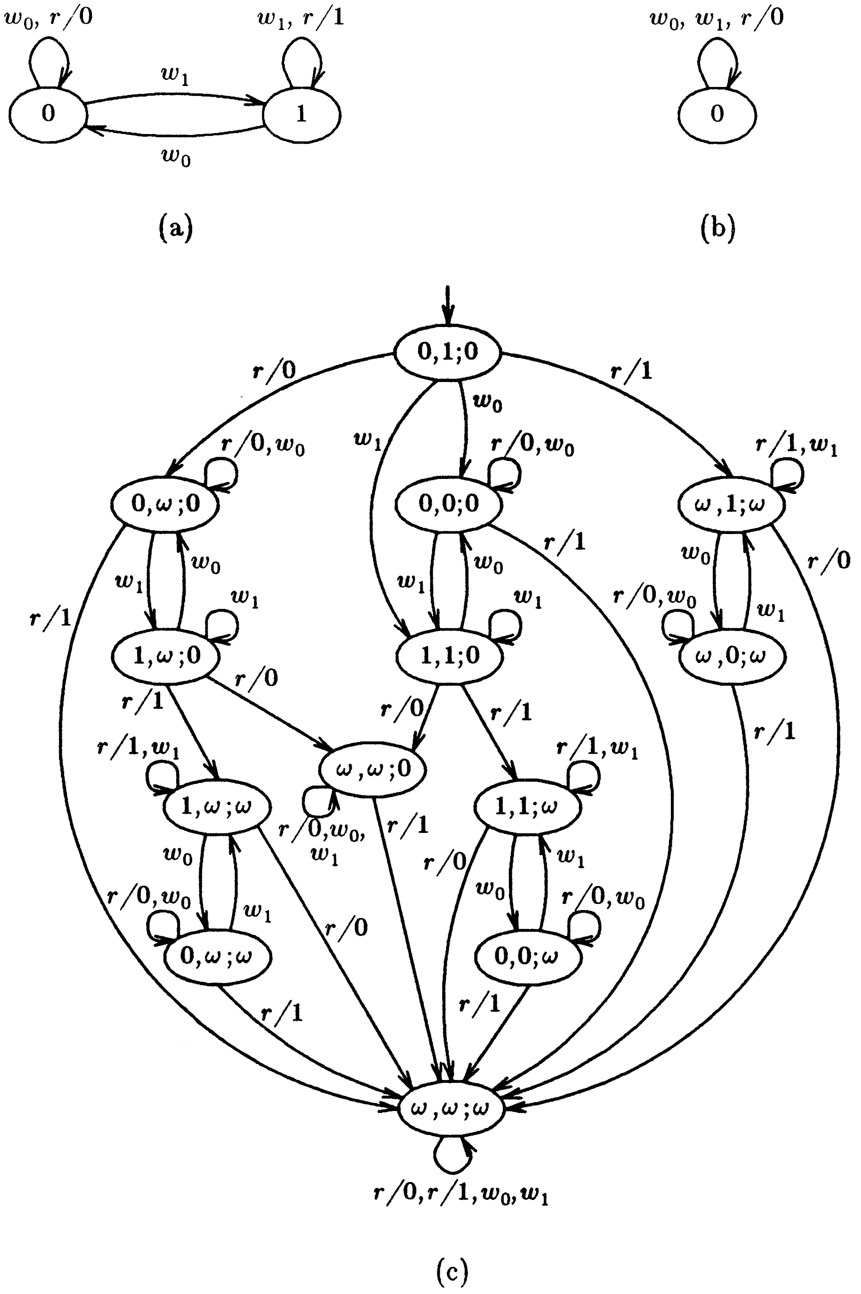


Figure 5. Memory cell stuck-at-0: (a) the good machine B_0; (b) the fault B_1; (c) the observer $\Delta(B_0,\{B_1\})$.

6. Deterministic Testing and Diagnosis

In this section we define the notion of diagnosis. The goal in circuit testing is to determine whether the given circuit is faulty or not. More accurately, the actual goal may depend on various requirements and may vary between the extremes of exact identification of the fault and just the detection of the presence of a fault. Our definition takes care of this range of possible goals.

The maximum amount of information that can be obtained from a test of a given circuit A with fault model $\mathcal{F}_{A_0}$ is the machine type (that is, the value of the index i), the initial state q, and the present state q'. Frequently, one does not need all this information. For example, it may suffice to identify the machine type or just to determine whether A is faulty or not. The properties we may wish to identify for a given purpose can be conveniently specified by a partition $B = \big(B_1, \ldots, B_r\big)$ of the set K, where $B_1, \ldots, B_r$ are the blocks of B and

$$K = \{(i,q,q') \mid i \in I_0,\ q,q' \in Q_i\}.$$

In order to be able to express the fact that a state d of the observer $\Delta(A_0, \mathcal{F}_{A_0}) = (D, X \times Y, \delta, d_0)$ has a property specified by some block of B, we introduce the notion of the contents $\|d\|$ of a state d of Δ as follows:

$$\|d\| = \{(i,q,q') \mid (i,q) \in K',\ q' = d_{i,q} \neq \omega\}$$

where K' is defined as in the previous section. Intuitively speaking, a state d identifies the property B_i if $\|d\| \subseteq B_i$. This leads to the following formal definition.

Definition 6.1 A state d of the fault observer Δ is said to be *B-decided* if $\|d\| \subseteq B_i$ for some block B_i of B. An input word $w \in X^*$ *B-diagnoses,* if and only if $\delta(d_0, (w,v))$ is B-decided for all output words $v \in Y^{|w|}$.

The following are typical examples of useful partitions.

- *Maximum information partition:* Let $B_{\mathrm{MI}} = \big(B_{i,q,q'} \mid (i,q,q') \in K\big)$ with $B_{i,q,q'} = \{(i,q,q')\}$. A word w B_{MI}-diagnoses if and only if $\|\delta(d_0,(w,v))\|$ contains at most one triple $(i,q,q') \in K$ for every $v \in Y^{|w|}$. If w is applied to the circuit A under test and v is the observed output word, then $\|\delta(d_0,(w,v))\| = \{(i,q,q')\}$ means that A has been identified as machine A_i started in state q and with present state q'. On the other hand, $\|\delta(d_0,(w,v))\| = \emptyset$ implies that A is not described by any machine in the fault model.
- *Machine and initial state partition:* Let $B_{\mathrm{IS}} = \big(B_{i,q,-} \mid (i,q) \in K'\big)$ with $B_{i,q,-} = \{(i,q,q') \mid q' \in Q_i\}$. A word w B_{IS}-diagnoses if and only if the projection of $\|\delta(d_0,(w,v))\|$ onto K', which is given by

$(i,q,q') \mapsto (i,q)$, contains at most one element $(i,q) \in K'$ for any $v \in Y^{|w|}$. Thus, $\emptyset \neq \|\delta(d_0,(w,v))\| \subseteq B_{i,q,-}$ means that A has been identified as machine A_i started in state q.

- *Machine and present state partition:* Let $B_{\mathrm{PS}} = \big(B_{i,-,q'} \mid (i,q') \in K'\big)$ with $B_{i,-,q'} = \{(i,q,q') \mid q \in Q_i\}$. A word w B_{PS}-diagnoses if and only if the projection of $\|\delta(d_0,(w,v))\|$ onto K', which is given by $(i,q,q') \mapsto (i,q')$, contains at most one element $(i,q') \in K'$ for any $v \in Y^{|w|}$. Thus, $\emptyset \neq \|\delta(d_0,(w,v))\| \subseteq B_{i,-,q'}$ means that A has been identified as machine A_i with present state q'.
- *Machine partition:* Let $B_{\mathrm{M}} = \big(B_i \mid i \in I_0\big)$ where $B_i = \{(i,q,q') \mid q,q' \in Q_i\}$. B_{M}-diagnosing means that one determines which machine A_i the circuit A actually is.
- *Fault partition:* Let $B_{\mathrm{F}} = \big(B_0, B_{\neq 0}\big)$ where $B_0 = \{(0,q,q') \mid q,q' \in Q_0\}$ and $B_{\neq 0} = \{(i,q,q') \mid i \in I \text{ and } q,q' \in Q_i\}$. In this case, B-diagnosing just results in distinguishing faulty from good circuits.

In general, B-diagnosis has the following interpretation: Let w be an input word which B-diagnoses, and let v be the output obtained from A, the circuit under test, as a reaction to the input w. If $\|\delta(d_0,(w,v))\| \neq \emptyset$ and $\|\delta(d_0,(w,v))\| \subseteq B_i$, then A has been identified as having the property B_i. On the other hand, if $\|\delta(d_0,(w,v))\| = \emptyset$, then A is not described by the fault model.

Example 6.2 The special partitions listed above are as follows for the fault model of Example 5.1:

$$\begin{aligned}
B_{\mathrm{MI}} &= \big(\{(0,0,0)\},\{(0,0,1)\},\{(0,1,0)\},\{(0,1,1)\},\{(1,0,0)\}\big),\\
B_{\mathrm{IS}} &= \big(\{(0,0,0),(0,0,1)\},\{(0,1,0),(0,1,1)\},\{(1,0,0)\}\big),\\
B_{\mathrm{PS}} &= \big(\{(0,0,0),(0,1,0)\},\{(0,0,1),(0,1,1)\},\{(1,0,0)\}\big),\\
B_{\mathrm{M}} &= \big(\{(0,0,0),(0,0,1),(0,1,0),(0,1,1)\},\{(1,0,0)\}\big),
\end{aligned}$$

and

$$B_{\mathrm{F}} = B_{\mathrm{M}}.$$

For example, the following are the B_{IS}-decided states of the observer:

$$(0,\omega;\omega),(1,\omega;\omega),(\omega,0;\omega),(\omega,1;\omega),(\omega,\omega;0),(\omega,\omega;\omega).$$

The set of B_{M}-decided states includes these and the following states:

$$(0,0;\omega),(1,1;\omega).$$

The input word rw_1r B_{IS}-diagnoses, while the word w_1r B_{M}-diagnoses but does not B_{IS}-diagnose.

We now define a non-deterministic finite acceptor, the non-deterministic B-tester, which will be used to clarify the notions of test sequence and test sequence generation. Essentially, the non-deterministic B-tester is obtained from the observer by the following two steps:

- by omitting the "output part" Y from the input alphabet $X \times Y$ of the observer; this introduces non-determinism;
- by introducing the set of B-decided states as the set of final states.

Definition 6.3 The *non-deterministic B-tester* for

$$\Delta(A_0, \mathcal{F}_{A_0}) = (D, X \times Y, \delta, d_0)$$

is the non-deterministic finite acceptor

$$\overline{\Delta}(A_0, \mathcal{F}_{A_0}, B) = (D, X, \overline{\delta}, \{d_0\}, F)$$

where F is the set of B-decided states of Δ and

$$\overline{\delta}(d, x) = \{d' \mid \exists y \in Y : \delta(d, (x, y)) = d'\}.$$

A word $w \in X^*$ is *strongly accepted* by $\overline{\Delta}$ if $\overline{\delta}(d_0, w) \subseteq F$. Let $L_{\mathbf{strong}}(\overline{\Delta})$ denote the set of words strongly accepted by $\overline{\Delta}$, the *strong language* of $\overline{\Delta}$.

The following observation is an immediate consequence of the definitions:

Remark 6.4 *A word w B-diagnoses if and only if $w \in L_{\mathbf{strong}}(\overline{\Delta})$.*

Observe that this result has two practically relevant consequences:

- the non-deterministic B-tester can be used to distinguish between sequences which B-diagnose and sequences which don't;
- the non-deterministic B-tester can be transformed into a test sequence generator.

Moreover—as will be shown in another section of this paper—the formal model of testing developed so far translates readily into a probabilistic setting.

We now proceed to show that the set of all words which B-diagnose is a regular set having some special properties. To show that $L_{\mathbf{strong}}(\overline{\Delta})$ is regular, one uses a variant of the power set construction. Recall that $\mathcal{P}(\overline{\Delta})$ is that part of the deterministic finite acceptor, obtained from $\overline{\Delta}$ by the power set construction, which contains precisely the states which are reachable. We modify the set of final states of $\mathcal{P}(\overline{\Delta})$ as follows to construct the deterministic finite acceptor $\widetilde{\mathcal{P}}(\overline{\Delta})$. A set $D' \subseteq D$ is a final state of $\widetilde{\mathcal{P}}(\overline{\Delta})$ if and only if $D' \subseteq F$, that is, if and only if every $d \in D'$ is B-decided. The acceptor $\widetilde{\mathcal{P}}(\overline{\Delta})$ is called the *deterministic B-tester.* Obviously, $L(\widetilde{\mathcal{P}}(\overline{\Delta})) = L_{\mathbf{strong}}(\overline{\Delta})$. This proves the following:

Proposition 6.5 *The set $L_{\mathrm{strong}}(\overline{\Delta})$, that is, the set of words which B-diagnose, is regular.*

The following observation will be useful in the sequel; it is easily verified.

Remark 6.6 *For $B \in \{B_{\mathrm{MI}}, B_{\mathrm{IS}}, B_{\mathrm{PS}}, B_{\mathrm{M}}, B_{\mathrm{F}}\}$, if a state $d \in D$ is B-decided, then every successor state d' of d is B-decided.*

Informally, this means that, once we have obtained some information (for example, that A is machine i in present state q) we cannot forget this information by applying further inputs. (After applying some input we still know that it is machine i and we still know its present state.) In view of the remark, a partition B is called *closed* if it satisfies the following condition: For every $d \in D$, if d is B-decided then every successor state of d is B-decided.

Proposition 6.7 *Let L be the set of words which B-diagnose. If B is closed, then $L = LX^*$.*

Proof: If d is B-decided then $\delta(d,(x,y))$ is also B-decided for all $x \in X$ and $y \in Y$. Therefore, if w B-diagnoses, then wx also B-diagnoses for every $x \in X$. This proves $LX^* \subseteq L$. □

The language $L_{\mathrm{strong}}(\overline{\Delta})$ is precisely the set of test sequences which, when applied to a circuit, will B-diagnose it. Thus, the shortest words in $L_{\mathrm{strong}}(\overline{\Delta})$ are the shortest test sequences. The deterministic B-tester $\widetilde{\mathcal{P}}(\overline{\Delta})$ distinguishes between test sequences and sequences which are not test sequences. If this distinction is the only point of interest, then standard reduction algorithms can be applied to $\widetilde{\mathcal{P}}(\overline{\Delta})$ in order to obtain a minimal acceptor for test sequence recognition—or a minimal test sequence generator.

As has been mentioned before, the observer may be far larger than required for the particular diagnosis task; the same remark applies to the non-deterministic B-tester. It seems natural to reduce both models modulo the task requirements. This is described in greater detail as follows.

Assume that $B = (B_1, B_2, \ldots, B_k)$, where the blocks are enumerated in some fashion. Since B is a partition, each B-decided state d can have its contents $\|d\|$ in only one block B_j unless $\|d\| = \emptyset$. Assign a Moore output $\gamma(d)$ to each state $d \in D$ as follows:

$$\gamma(d) = \begin{cases} 0, & \text{if } d \text{ is not } B\text{-decided}, \\ j, & \text{if } \emptyset \neq \|d\| \subseteq B_j, \\ \infty, & \text{if } \|d\| = \emptyset. \end{cases}$$

Now reduce the resulting Moore machine Δ' using standard reduction techniques. This is illustrated in the example below.

Example 6.8 We use the fault model and the observer of Example 5.1. No reduction of the observer is possible for $B = B_{\mathrm{MI}}$. Consider $B = B_{\mathrm{IS}}$. In this case, the state $(0,\omega;\omega)$ would be considered equivalent to $(1,\omega;\omega)$; similarly, $(\omega,0;\omega)$ and $(\omega,1;\omega)$ would be considered equivalent. However, as the read operation behaves differently on these states, no reduction is possible in the case of $B = B_{\mathrm{IS}}$ either.

Now consider $B = B_{\mathrm{PS}}$. In this case we would start with the equivalences

$$(0,\omega;\omega) \sim (0,0;\omega) \sim (\omega,0;\omega)$$

and

$$(1,\omega;\omega) \sim (1,1;\omega) \sim (\omega,1;\omega).$$

As a consequence we get

$$(0,\omega;0) \sim (0,0;0)$$

and

$$(1,\omega;0) \sim (1,1;0).$$

The resulting "reduced observer" is shown in Figure 6. In the diagram, a state $(x^*;y)$ denotes the equivalence class of $(x,\omega;y)$.

Finally, consider $B = B_{\mathrm{M}}$ and note that $B_{\mathrm{M}} = B_{\mathrm{F}}$ in this case. It turns out that reduction with respect to B_{F} results in the same automaton as that for B_{PS}.

Reduction with respect to a partition B, as illustrated in Example 6.8 results in a minimal automaton which still distinguishes the properties expressed by the blocks of B. There is another natural notion of reduction; in this case no distinction is made between B-decided states, and reduction is carried out according to standard algorithms. This corresponds to assigning an output of 0 to all states that are not B-decided and an output of 1 to all the states that are B-decided. This is illustrated in the following example.

Example 6.9 We continue with Example 5.1 and consider $B = B_{\mathrm{F}} = B_{\mathrm{M}}$. One starts with B-decided states made equivalent. Using the notation of the previous example, we find the additional equivalence classes $(0^*;0)$ and $(1^*;0)$. Moreover, $(0,1;0)$ is equivalent to the states in $(1^*;0)$. The resulting reduced automaton is shown in Figure 7. Observe, that this automaton is identical to the one obtained in [Ful].

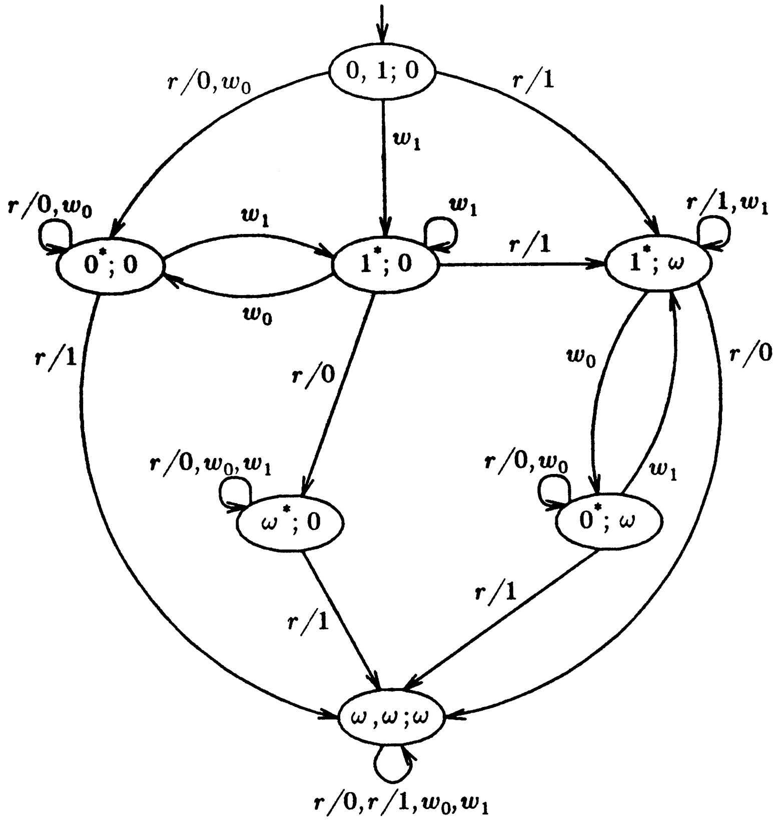


Figure 6. Observer of Example 5.1 reduced for B_{PS}.

The reduction as described in the first of these two examples is defined with respect to an equivalence relation $\sim_B$ on the set of states of the observer. The relation $\sim_B$ is the coarsest equivalence such that $d \sim_B d'$ implies

$$\|\delta(d, (w, v))\| \subseteq B_i \iff \|\delta(d', (w, v))\| \subseteq B_i$$

and

$$\|\delta(d, (w, v))\| = \emptyset \iff \|\delta(d', (w, v))\| = \emptyset$$

for all $w \in X^*$, $v \in Y^{|w|}$, and $B_i \in B$. This condition implies that $\sim_B$ is an automaton congruence. Let $\Delta / \sim_B$ denote the resulting factor automaton.

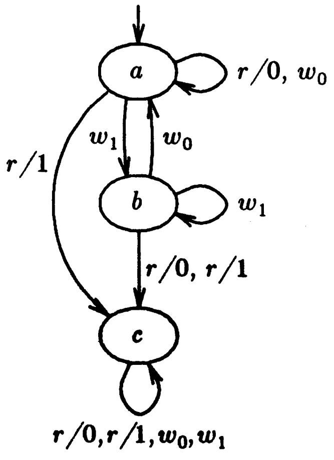


Figure 7. Reduction with all B_{M}-decided **states equivalent.**

On the other hand, reduction as described in the second of these examples is defined with respect to another equivalence relation $\equiv_B$ on the set of states of the observer. The relation $\equiv_B$ is the coarsest automaton congruence which has the set of B-decided states as one equivalence class. Let $\Delta/\equiv_B$ denote the resulting factor automaton.

Remark 6.10 *$\Delta/\equiv_B$ is the minimal acceptor for the language consisting of all B-diagnosing words. $\Delta/\sim_B$ is the minimal automaton which distinguishes the input words according to the properties in B.*

In [Hel] and [He2] input words of a finite automaton with special properties related to diagnostic experiments are considered. We indicate here, how some of the notions introduced in [Hel] and [He2] arise as special cases of our concepts. There only a single machine with an unknown initial state is considered.

An input word is distinguishing in the sense of [Hel], if knowledge of the output word uniquely determines the starting state. The natural generalization to our situation is as follows: A word $w \in X^*$ is said to be distinguishing if knowledge of the output word uniquely determines the machine index i and its initial state q. Formally, w is *distinguishing*, if for every $v \in Y^{|w|}$ there is a pair $(i,q) \in K'$ such that $\|\delta(d_0,(w,v))\| \subseteq B_{i,q,-}$. Thus, a word is distinguishing if and only if it B_{IS}-diagnoses.

An input word is homing in the sense of [Hel], if knowledge of the output word uniquely determines the present state. We generalize this as follows: A word $w \in X^*$ is said to be homing if knowledge of the output word uniquely determines the machine index i and its present state q'. Formally,

w is *homing*, if for every $v \in Y^{|w|}$ there is a pair $(i,q') \in K'$ such that $\|\delta(d_0,(w,v))\| \subseteq B_{i,-,q'}$, that is, if and only if it is B_{PS}-diagnosing.

The observer obtained by our construction for Example 5.1 differs from the one constructed in [Ful] considerably. Two critical points deserve to be mentioned:

- in [Ful] no convincing reason is given for the choice of the initial states; as our construction shows, the initial state is uniquely determined.
- in [Ful] the assumption is made, that the circuit under test is indeed faulty. The testing will only have to detect this fact.

Whereas the first remark is crucial for the correctness of the testing results, the second one mainly affects the size of the observer and the tester. We generalize this assumption as follows: Let A_0 be the good machine and $\mathcal{F}_{A_0} = \{A_i \mid i \in I\}$ a fault model for A_0. Let $K_0 \subseteq K'$. We modify the the construction of the observer Δ so as to include the assumption that the circuit under test is one of the machines A_i started in state q for $(i,q) \in K_0$. This assumption is referred to as the K_0-assumption.

Consider the following equivalence relation on D: $d \sim_{K_0} d'$ if $d = d'$ or if $d_{i,q} = \omega = d'_{i,q}$ for all $(i,q) \in K_0$. This equivalence is an automaton congruence because, if d and d' are equivalent, so are their successors. Now $\Delta_{K_0} = \Delta/\sim_{K_0}$ is the observer under the K_0-assumption. Of course, $\Delta/\sim_{K_0}$ need not be reduced.

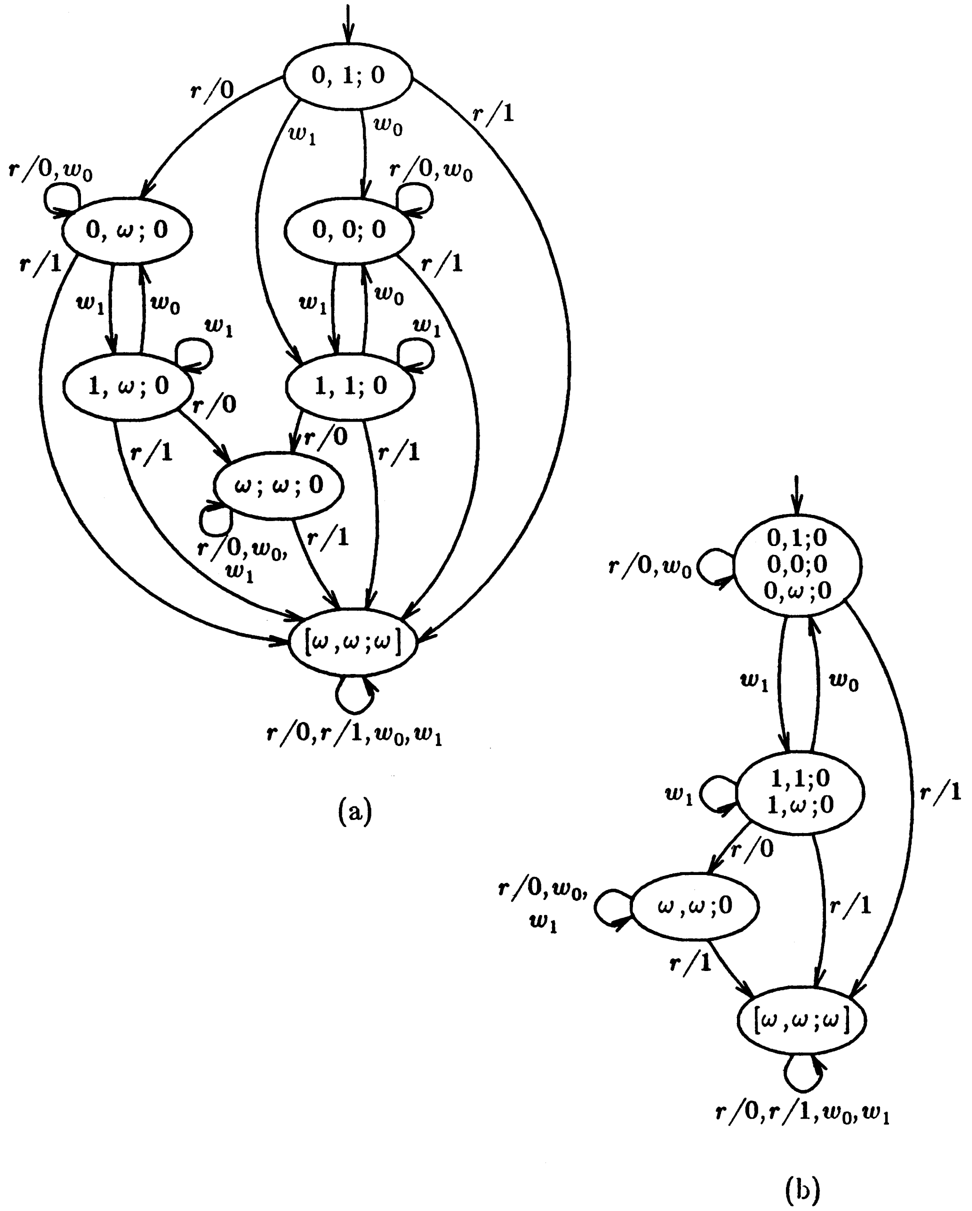
Intuitively, the equivalence $\sim_{K_0}$ lumps together all those states which are impossible under the K_0-assumption. Closed partitions of D turn into closed partitions of $D/\sim_{K_0}$. Figure 8 shows the observer of Example 5.1 using the assumption $K_0 = \{(1,0)\}$ of [Ful] and its reduced version with respect to B_{F}.

The introduction of an assumption K_0 via a modification of the construction of the observer and the B-tester built from the observer is quite convenient in the non-probabilistic situation as it helps to keep the resulting automata "small." We shall see that in the probabilistic case a simpler approach can be taken by assigning probabilities appropriately.

7. Further Basic Notions

Our next step in modelling the testing of circuits involves adding probabilities to our models. Before doing so, we review a few basic notions concerning probabilistic automata and (information) sources [Do], [St].

A *probabilistic finite semi-automaton* is a triple (Q,X,H) with Q and X as above, and with H the probabilistic transition function. Thus $H(q' \mid q,x)$ is the probability of the next state being q', given that the current state is q and the input symbol is x. Again, H is extended to input



(a)

(b)

Figure 8. Observer under K_0-assumption: (a) observer; (b) reduced with respect to B_F.

words so that $H(q' \mid q, w)$ is the probability of the state reached being q' given that the current state is q and the input word is w. A *probabilistic state* is a probability distribution over Q. For a probabilistic state ξ,

$$H(\cdot \mid \xi, w)$$

is the probabilistic state reached from ξ under input w. A *probabilistic finite acceptor* is a quintuple $A = (Q, X, H, \xi_0, Q_F)$ such that (Q, X, H) is a probabilistic finite semi-automaton, ξ_0 is a probabilistic state, the *initial state*, and $Q_F \subseteq Q$ is the set of *final* or *accepting* states. For $\lambda \geq 0$, the *language accepted by A with threshold* λ is the set

$$L_\lambda(A) = \{w \mid H(Q_F \mid \xi_0, w) > \lambda\}.$$

A *probabilistic Mealy automaton* is a quadruple (Q, X, Y, H) where Y is an alphabet, the *output alphabet*, and H is the probabilistic transition-and-output function. In other words, $H(q', y \mid q, x)$ is the probability of the next state being q' and the output being y, given that the current state is q and the input is x.

For an alphabet X, let X^ω denote the set of infinite sequences *(ω-words)* over X. A *source* is a pair $\mathcal{S} = (X, \psi)$ with X an alphabet and ψ a probability distribution over X^ω [Fe]. In most of the rest of this paper, only the special cases of memoryless and Markov sources are considered. They are defined below.

As usual, we let $\psi(w) = \psi(wX^\omega)$ for $w \in X^*$. Observe also, that $\psi(X^t w)$ is the probability of $\mathcal{S}$ sending w beginning at time t for $t \geq 0$; here and in the sequel we assume discrete time starting at 0. We define $\psi_t(w) = \psi(X^t w)$.

The following two special types of sources are of particular interest in the sequel: In a *memoryless* source one has

$$\psi(x_0 x_1 \cdots x_t) = \prod_{i=0}^{t} \psi(x_i)$$

for every $t \geq 0$ and any $x_0, x_1, \ldots, x_t \in X$. In a *(homogeneous) Markov source* one has a probability distribution ψ_0 over X and a set of conditional probabilities $\mu(x' \mid x)$ with $x', x \in X$ such that

$$\psi(x_0 x_1 \cdots x_t) = \psi_0(x_0) \cdot \prod_{i=1}^{t} \mu(x_i \mid x_{i-1})$$

for all $t \geq 0$ and any $x_0, x_1, \ldots, x_t \in X$. Clearly, with X considered as the state space, a Markov source is just a *Markov chain* over X with transition matrix

$$M = (\mu(x' \mid x))_{x, x' \in X} \cdot$$

Its distribution at time t, $t \geq 0$ is given by $\psi_t = \psi_0 M^t$ when ψ_0 and ψ_t are considered as row vectors.

8. Probabilistic Test Model

In this section we add probabilities to our models for fault detection and diagnosis. In particular, we introduce the notions of "probabilistic fault schema," "probabilistic fault observer," and "test model." We also continue to expand our examples to illustrate how certain actual test situations can be expressed in our setting. The proposal of [Ful] turns out to form a special case of our model.

A test model consists of three parts: the (probabilistic) fault schema describes the behaviour of a circuit A assuming certain kinds of faults; the probabilistic observer describes how tests and their outcomes are to be interpreted; finally, the test procedure is defined using a test source. All three components have the following in common: a finite set Q of possible *physical states,* a finite set X of allowable *actions,* and a finite set Y of possible *outputs.*

Let $A_0 \in \mathcal{A}(Q,X,Y)$ be a good machine, and let $\mathcal{F}_{A_0} = \{A_i = (Q_i, X, Y_i, \delta_i, \lambda_i) \mid i \in I\}$ be a fault model for A_0. Without loss of generality, assume that $Y = \bigcup_{i \in I_0} Y_i$ and $Q = \bigcup_{i \in I_0} Q_i$. Consider a probability distribution $\pi = (\pi_0, \ldots, \pi_n)$ over I_0. We interpret π_i as the probability of a randomly chosen circuit being of type i. By definition, the faults A_i are given in such a way that they are mutually exclusive. Typically, π would be determined experimentally in the circuit production process.

With these data, one defines a probabilistic automaton

$$F = F(\mathcal{F}_{A_0}, \pi) = (Q, X, Y, H),$$

the ***probabilistic fault schema,*** as follows: For $q, q' \in Q$, $x \in X$, and $y \in Y$ one has

$$H(q', y \mid q, x) = \Bigg(\sum_{\substack{i \in I_0 \text{ with} \\ \delta_i(q,x)=q' \\ \lambda_i(q,x)=y}} \pi_i \Bigg) \Bigg/ \Bigg(\sum_{\substack{i \in I_0 \text{ with} \\ q \in Q_i}} \pi_i \Bigg).$$

Thus, $H(q', y \mid q, x)$ is the probability of an output y and a transition of A from state q into state q' under input x, when the probability distribution π of faults is taken into account.

Example 8.1 Let B_0 and $\mathcal{F}_{B_0}$ be as in Example 4.1. The probabilistic fault schema $F(\mathcal{F}_{B_0}, \pi)$ is shown in Figure 9. The outputs are irrelevant in this case and, therefore, have been omitted. To illustrate the calculations, let $q = 10$, $x = w_1^j$. Then one has $q' = 11$ in B_0 and B_1, $q' = 10$ in B_3, and $q = 10 \notin Q_2 \cup Q_4$. Thus

$$H(11 \mid 10, w_1^j) = (\pi_0 + \pi_1)/(\pi_0 + \pi_1 + \pi_3).$$

Whenever $H(q' \mid q, x) = 0$ we do not show the edge from q to q' labelled x. If $H(q' \mid q, x) = 1$ the probability is not shown, to keep the diagram simpler.

In the next step we define a probabilistic fault observer. A *probabilistic fault observer* is a 4-tuple $\widehat{\Delta}(A_0, \mathcal{F}_{A_0}, \pi, \alpha) = (D, X, H, D_0)$ with the following properties: A_0 is the good machine; $\mathcal{F}_{A_0} = \{A_i \mid i \in I\}$ is a fault model for A_0; X is the input alphabet; π is a probability distribution over I_0; for each $i \in I_0$ and $q \in Q$, $\alpha_{i,q} = \alpha(q \mid i)$ is the probability, that a machine A_i is initially in state q. Of course, $\alpha_{i,q} = 0$ for $q \notin Q_i$. Again, like the probabilities π_i, the data $\alpha_{i,q}$ would have to be determined experimentally.

The construction of $\widehat{\Delta}$ starts from the (deterministic) fault observer $\Delta(A_0, \mathcal{F}_{A_0})$. $\widehat{\Delta}$ is a probabilistic semi-automaton with starting state. Its set of states is the set D, the set of states of Δ. Its initial state distribution is deterministic and given by

$$D_0(d) = \begin{cases} 1, & \text{if } d = d_0, \\ 0, & \text{otherwise,} \end{cases}$$

for $d \in D$, where d_0 is the starting state of Δ. Truly probabilistic behaviour arises from the uncertainty about outputs only. In defining the probabilistic transition function H, we use the notation

$$\beta(d) = \sum_{i \in I_0} \sum_{\substack{q \in Q_i \\ d_{i,q} \neq \omega}} \alpha(q \mid i)\pi_i$$

where $d \in D$. Then H is given by

$$H(d' \mid d, x) = \begin{cases} 0, & \text{if } \beta(d) > 0 \text{ and } \forall y \in Y : \delta(d, (x, y)) \neq d', \\ \beta(d')/\beta(d), & \text{if } \beta(d) > 0 \text{ and } \exists y \in Y : \delta(d, (x, y)) = d', \\ 0, & \text{if } \beta(d) = 0 \text{ and } d' \neq (\omega, \ldots, \omega), \\ 1, & \text{if } \beta(d) = 0 \text{ and } d' = (\omega, \ldots, \omega), \end{cases}$$

for $d, d' \in D$ and $x \in X$.

The first two cases of the definition of H take care of the situation, when the present state of the observer still contains pairs (i, q) which have

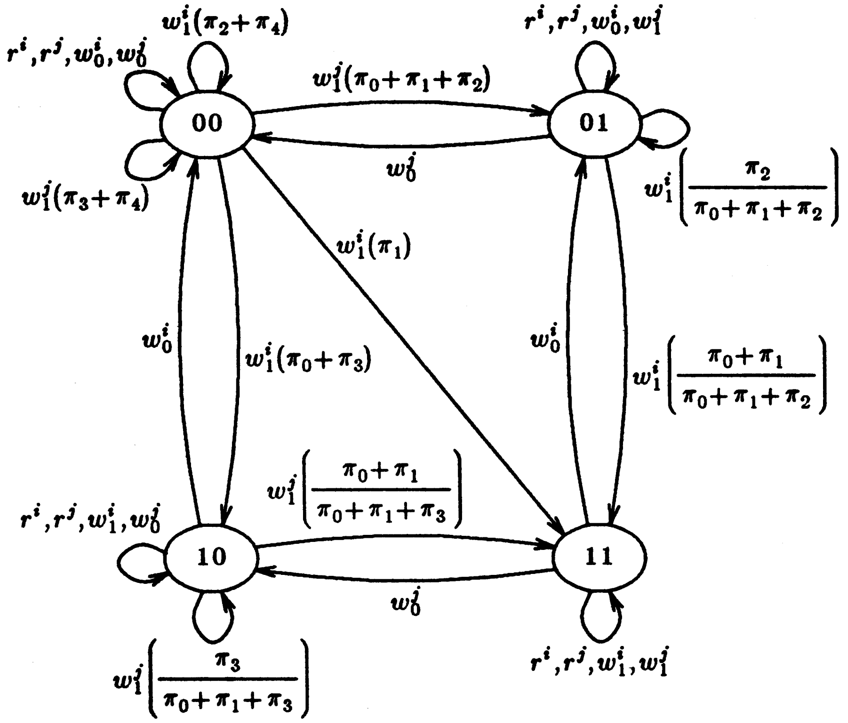

$w_1^i(\pi_2+\pi_4)$
r^i,r^j,w_0^i,w_0^j
$w_1^j(\pi_3+\pi_4)$
00
$w_1^j(\pi_0+\pi_1+\pi_2)$
w_0^j
r^i,r^j,w_0^i,w_1^j
01
$w_1^i\left(\frac{\pi_2}{\pi_0+\pi_1+\pi_2}\right)$
$w_1^i(\pi_1)$
w_0^i
$w_1^i(\pi_0+\pi_3)$
w_0^i
$w_1^i\left(\frac{\pi_0+\pi_1}{\pi_0+\pi_1+\pi_2}\right)$
$w_1^j\left(\frac{\pi_0+\pi_1}{\pi_0+\pi_1+\pi_3}\right)$
r^i,r^j,w_1^i,w_0^j
10
11
w_0^j
$w_1^j\left(\frac{\pi_3}{\pi_0+\pi_1+\pi_3}\right)$
r^i,r^j,w_1^i,w_1^j

Figure 9. Probabilistic fault schema $F(\mathcal{F}_{B_0},\pi)$.

non-zero a priori probabilities $a_{i,q}\pi_i$. The third and fourth cases deal with the situation when all pairs (i,q) contained in d have probability $\alpha_{i,q}\pi_i = 0$. Clearly, $\sum_{d'\in D} H(d' \mid d,x) = 1$ for all $d \in D$ and $x \in X$ because every A_i is deterministic.

An assumption $K_0 \subseteq K'$, as discussed in one of the previous sections, can be easily expressed by letting

$$\pi_i = 0 \quad \text{if} \quad (i,q) \notin K_0 \quad \text{for all} \quad q \in Q_i$$

or

$$\alpha_{i,q} = 0 \quad \text{if} \quad (i,q) \notin K_0 \quad \text{and} \quad (i,q') \in K_0 \quad \text{for some} \quad q' \in Q_i.$$

This automatically eliminates transitions which are impossible according to K_0 and leads to the probabilistic version $\widehat{\Delta}_{K_0}$ of Δ_{K_0}. Let H_{K_0} denote its probabilistic transition function.

Proposition 8.2 *Let $\mathcal{F}_{A_0}$ be a fault model and let Δ_{K_0} and $\widehat{\Delta}_{K_0}$ be the deterministic and probabilistic observers, respectively, with assumption K_0, $K_0 \subseteq K'$. For $d \in D$, $d' \in D \setminus \{(\omega,\dots,\omega)\}$ and $x \in X$, if the probability of a transition from d to d' under x in $\widehat{\Delta}_{K_0}$ is greater than 0, then there is such a transition in Δ_{K_0}.*

Proof: By the definition, if d does not contain any $(i,q) \in K_0$, then $\beta(d) = 0$ and $H_{K_0}(d' \mid d,x) = 0$ for $d' \neq (\omega,\dots,\omega)$. If d contains some $(i,q) \in K_0$ then $\beta(d)$ may be greater than 0. If it is not, then $H_{K_0}(d' \mid d,x) > 0$ implies $d' = (\omega,\dots,\omega)$. Thus we may assume that $\beta(d) > 0$. But then $H_{K_0}(d' \mid d,x) > 0$ implies that $\delta(d,(x,y)) = d'$ for some $y \in Y$ and $\beta(d') > 0$. Hence, there is a transition from d' to d in Δ_{K_0} under input (x,y). □

Example 8.3 We continue using Example 5.1. Using the assumption $K_0 = \{(1,0)\}$, one finds that $\widehat{\Delta}_{K_0}$ as shown in Figure 10 is actually deterministic, as true probabilistic behaviour arises from the uncertainty about the outputs only. (Note that $\widehat{\Delta}_{K_0}$ corresponds to Δ_{K_0} of Figure 8(a). In Figure 10, all the transitions shown have probability 1 and transitions with probability 0 are not shown.) On the other hand, without the assumption K_0, the observer $\widehat{\Delta}$ is truly probabilistic at certain transitions. For instance, one has

$$H(d' \mid (0,1;0),r) = \begin{cases} \alpha_{0,1}\pi_0, & \text{if } d' = (\omega,1;\omega), \\ \alpha_{0,0}\pi_0 + \pi_1, & \text{if } d' = (0,\omega;0). \end{cases}$$

Now we combine the ideas developed so far into the definition of a test model. A *test model* is a 6-tuple $\Theta = (A_0, \mathcal{F}_{A_0}, \pi, \alpha, \mathcal{S})$ with the following properties: A_0, $\mathcal{F}_{A_0}$, π, and α define a probabilistic fault observer; $\mathcal{S}$ is a source with alphabet X, the *test source.*

The interpretation of Θ is as follows: The source $\mathcal{S}$ provides the input symbols to the circuit A which is to be tested and also to the "observer" $\widehat{\Delta}$ who runs the good machine and the various faulty machines in parallel.

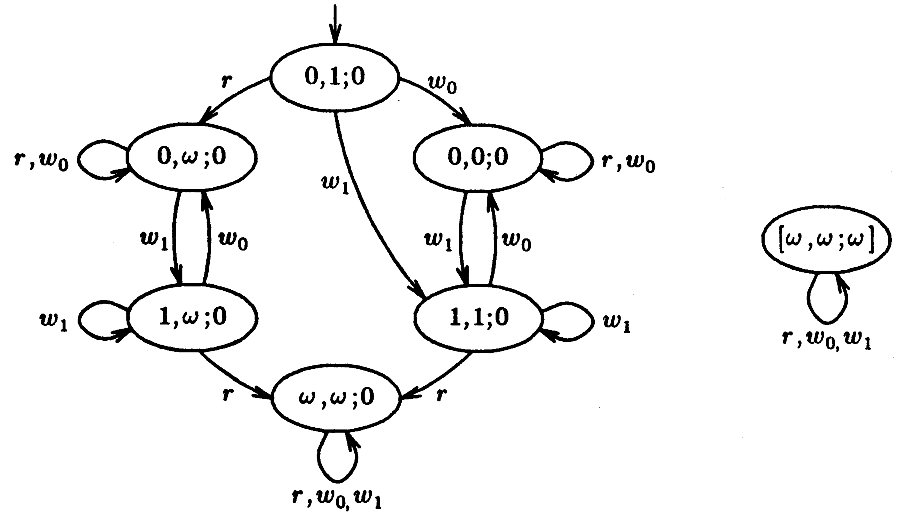


Figure 10. The observer $\widehat{\Delta}_{K_0}$ of Example 5.1 with assumption $K_0 = \{(1,0)\}$. All transition probabilities are equal to 1.

Transitions of A take place according to probabilities defined by H in the probabilistic fault schema $F = F(\mathcal{F}_{A_0}, \pi) = (Q, X, H)$. Which inputs are applied, depends on the characteristics of the source $\mathcal{S}$.

Let $\mathcal{S} = (X, \psi)$. We assume that $\mathcal{S}$ is independent of $\widehat{\Delta}$; this excludes the possibility of the observer influencing the test sequence. Note that deterministic testing is not excluded by this assumption. On the other hand, adaptive testing is not covered. Additional restrictions on $\mathcal{S}$ will be useful, however. In particular, assuming that $\mathcal{S}$ is a Markov source simplifies the mathematical discussion significantly. In this context it should be pointed out, that in [Ful] a memoryless source is used.

A test model Θ with $\mathcal{S}$ a homogeneous Markov source is called a *Markov test model.* In the rest of this section we consider Markov test models only. We show how to determine the transition matrix of a Markov chain describing the behaviour of a Markov test model. This procedure clarifies and generalizes the mathematics used in [Ful].

The state set of the Markov chain $\mathcal{C}_\Theta$ is the set $D \times X$. The entries $\mu(d', x' \mid d, x)$ of the transition matrix M are defined as follows:

$$\mu(d', x' \mid d, x) = \psi(x' \mid x)\Pi(d' \mid d, x).$$

Of course, if $\mathcal{S}$ is a memoryless or even a deterministic source, the notation can be simplified considerably.

For $x \in X$ let $\psi_0(x)$ be the probability of $\mathcal{S}$ sending x in step 0. Define $\mu_i(d, x)$ as the probability of state (d, x) of the Markov chain at step i and let μ_i denote the row vector with entries $\mu_i(d, x)$. Then $\mu_0(d, x) = \psi_0(x)D_0(d)$ and $\mu_i = \mu_0 M^i$.

As in previous sections, let K be the set of all triples (i, q, q') with $i \in I_0$ and $q, q' \in Q_i$. Let B be a partition of K. We can now define the probabilistic analogue of B-diagnosis.

Definition 8.4 Let ε be a real number with $0 \leq \varepsilon \leq 1$. The test model ***ε-B-diagnoses at step t*** if

$$\mu_t(B) = \sum_{x \in X} \sum_{B'} \sum_{\|d\| \in B'} \mu_t(d, x) \geq \varepsilon$$

where the second summation extends over all blocks B' of B.

Recall that a set of states F of a Markov chain is called *closed* if for every state $f \in F$, the probability of going from f to some state outside F is 0. With probability 1, a finite Markov chain will eventually enter one of its closed sets [Do].

Proposition 8.5 *Let $\Theta = (A_0, \mathcal{F}_{A_0}, \pi, \alpha, \mathcal{S})$ be a Markov test model, let B be a closed partition of K, and let $F = F' \times X$ where K and X are as above and where F' is the set of B-decided states of the observer $\Delta(A_0, \mathcal{F}_{A_0})$. Then F is a closed set and $\mu_t(B) \leq \mu_{t+1}(B)$ for all t.*

Proof: The state set of $\mathcal{C}_\Theta$ is finite. Moreover, if d is a B-decided state of the observer Δ then every successor state of d is B-decided. Therefore, F is closed. By the definition, $\mu_t(B)$ is the probability of the chain $\mathcal{C}_\Theta$ being in F at time t. As F is closed, this probability can only increase as t increases. □

By the general theory of Markov chains [Do], the state transition process will eventually enter some closed set with probability 1. Let us assume that there is no closed set $\widehat{F} \subseteq (D \times X) \setminus F$ such that $\mu_t(\widehat{F}) > 0$ for some t. Roughly speaking, this means that there is no set of states in which the test model could get "trapped" without the ability to diagnose. Such an assumption was made implicitly in [Ful]. In this case,

$$\mu_t((D \times X) \setminus F) < \gamma' \gamma^t$$

for some γ and γ' with $0 < \gamma < 1$, $1 < \gamma'$, and for all t. Bounds on γ and γ' can be computed from the transition matrix M. Now let ε be a real number such that $0 < \varepsilon < 1$. By ε we denote the reliability threshold of probabilistic testing. We intend to determine a lower bound on the number t of steps to be executed by the probabilistic tester in order that F be reached with a probability no less that $1 - \varepsilon$. Assume that $t > \frac{-\log \gamma'}{\log \gamma}$ so that $\gamma' \gamma^t < 1$. From

$$\mu_t(F) > 1 - \gamma' \gamma^t \geq 1 - \varepsilon$$

one computes $\gamma' \gamma^t \leq \varepsilon$, that is, $\log \gamma' + t \log \gamma \leq \log \varepsilon$ or $t \geq \frac{\log \varepsilon - \log \gamma'}{\log \gamma}$. This result is summarized in the following observation.

Proposition 8.6 *Let Θ be a Markov test model, let B be a closed partition of K, and let $F = F' \times X$ where F' is the set of B-decided states of the observer and X is the input alphabet. Assume that there is no closed set $\widehat{F}$ in $(D \times X) \setminus F$ such that $\mu_t(\widehat{F}) > 0$ for some t. Then there are constants γ and γ' with $0 < \gamma < 1$ and $1 < \gamma'$ such that for every ε with $0 < \varepsilon < 1$ the inequality $t \geq \frac{\log \varepsilon - \log \gamma'}{\log \gamma}$ implies $\mu_t(F) > 1 - \varepsilon$. Moreover, upper bounds on γ and γ' can be computed from the transition matrix M of $\mathcal{C}_\Theta$.*

Example 8.7 Consider the fault model of Example 5.1 with the assumption $K_0 = \{(1,0)\}$. Moreover, we assume that there is an additional input symbol $\not{e}$ in X, which does not change the state and results in no output, that is, a "do-nothing" operation. This operation will be needed later in comparing our work with that of [Ful]. As $\widehat{\Delta}_{K_0}$ is deterministic, the transition probabilities in the test model arise from the source $\mathcal{S}$ only. Suppose that $\mathcal{S}$ is memoryless with probabilities $\psi(r)$, $\psi(w_0)$, $\psi(w_1)$, and $\psi(\not{e})$. Then the state set of Δ_{K_0} can be used as the state set of the test model. The resulting Markov chain—after merging of equivalent states—is shown in Figure 11.

To complete this section, we indicate how the model of [Ful] is obtained as a special case of our model. In [Ful] only memory cells are dealt with. Suppose the memory to be tested consists of m cells and we are testing for stuck-at-0 faults as in Example 5.1. The test source $\mathcal{S}$ issues read or write instructions for the m cells at random. In our model, this would mean that in addition to the reading and writing actions we require a "do-nothing" action $\not{e}$ in the alphabet X as shown in Example 8.7 and Figure 11. Implicitly, this is actually required in [Ful], too. The action $\not{e}$ corresponds to actions performed on other cells. Like [Ful], we assume that $\mathcal{S}$ is memoryless with actions equally probable. Thus $\psi(r) = \psi(w) = 2\psi(w_0) = 2\psi(w_1) = \frac{1}{2m}$ and $\psi(\not{e}) = 1 - \frac{1}{m}$. Moreover, the memory is assumed to be faulty, that is, we assume $K_0 = \{(1,0)\}$ as shown in Example 8.7. Using the state set of Δ_{K_0} as the state set of $\mathcal{C}_\Theta$ as in Example 8.7, we get the following transition

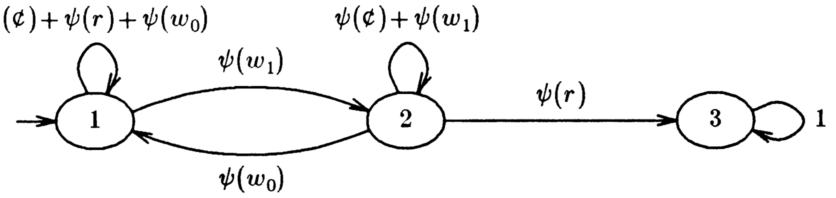


Figure 11. Test model for Example 5.1 with memoryless source with assumption $K_0 = \{(1,0)\}$.

matrix M for $\mathcal{C}_\Theta$

$$M = \begin{pmatrix} \frac{1}{2m} + \frac{1}{4m} + \frac{m-1}{m} & \frac{1}{4m} & 0 \\ \frac{1}{4m} & \frac{1}{4m} + \frac{m-1}{m} & \frac{1}{2m} \\ 0 & 0 & 1 \end{pmatrix} = \left(m_{i,j}\right)_{i,j=1,2,3}$$

with the indices $1,2,3$ of the rows and columns of M corresponding to the states $(0^*;0)$, $(1^*;0)$, and $(\omega^*;\omega)$, respectively. It turns out that $\mathcal{C}_\Theta$ has only a single reachable closed set, that is, the set $F = \{(\omega^*;\omega)\}$. The other states, that is, those corresponding to the indices 1 and 2, are transient.

Given a threshold ε, one is interested in the number t of steps required to guarantee fault detection with a probability no less than $1-\varepsilon$. Let $t_0(\varepsilon, m)$ be the smallest t such that the probability of detection is greater than or equal to $1 - \varepsilon$. As in [Ful], we consider the quotient $t_0(\varepsilon, m)/m$, called the *length coefficient.* In [Ful], it is conjectured that the length coefficient is bounded in the case of single stuck-at faults. This conjecture is proved in the sequel. First we derive an exact expression for the probability of detection.

Lemma 8.8 *Consider random testing of stuck-at-*0 *faults of single-bit memory cells with equally likely actions. Let*

$$p_1 = 1 + \frac{m(4 + 2\sqrt{2}) - 1}{8m^2 - 8m + 1}$$

and

$$p_2 = 1 + \frac{m(4 - 2\sqrt{2}) - 1}{8m^2 - 8m + 1}.$$

Then the probability of detection by step t is equal to

$$1 + \frac{\sqrt{2}-1}{2p_1^t} - \frac{\sqrt{2}+1}{2p_2^t}.$$

The sequence of these values is strictly increasing for $t \to \infty$, $t \geq 1$, and it converges to 1.

Proof: To simplify notation, let $x_t = m_{1,3}^{(t)}$, $y_t = m_{2,3}^{(t)}$, and

$$M = \begin{pmatrix} \alpha & 1-\alpha & 0 \\ \beta & \gamma & 1-\beta-\gamma \\ 0 & 0 & 1 \end{pmatrix}$$

where $\alpha = (4m-1)/(4m)$, $\beta = 1/(4m)$, and $\gamma = (4m-3)/(4m)$. Note that x_t is also the probability of detection by step t, as the test model is started in state 1.

The values of x_t and y_t satisfy the following recursive equations:

$$\begin{aligned} x_{t+1} &= \alpha x_t + (1-\alpha)y_t, \\ y_{t+1} &= \beta x_t + \gamma y_t + (1-\beta-\gamma). \end{aligned}$$

As starting values one has $x_0 = y_0 = 0$. To solve this recursion, consider the generating functions

$$F(z) = \sum_{t=0}^{\infty} x_t z^t \qquad \text{and} \qquad G(z) = \sum_{t=0}^{\infty} y_t z^t$$

of x_t and y_t, respectively. From the recursion one computes the following two equations:

$$\begin{aligned} F(z) - \alpha z F(z) - (1-\alpha)zG(z) &= 0, \\ G(z) - \beta z F(z) - \gamma z G(z) - \frac{(1-\beta-\gamma)z}{1-z} &= 0. \end{aligned}$$

Solving for $F(z)$ yields

$$F(z) = \frac{(1-\alpha)(1-\beta-\gamma)z^2}{(1-z)(1-\alpha z-(1-\alpha)\beta z^2-\gamma(1-\alpha z)z)}.$$

The denominator of this expression for $F(z)$ has three roots p_1, p_2, and p_3 where $p_3 = 1$. One verifies that p_1 and p_2 have the values given in the lemma. Thus all the roots are distinct. Decomposing $F(z)$ into partial fractions yields

$$
\begin{aligned}
F(z) &= \frac{1}{1-z} - \frac{p_1(p_2-1)}{(p_2-p_1)(1-z/p_1)} + \frac{p_2(p_1-1)}{(p_2-p_1)(1-z/p_2)} \\
&= \frac{1}{1-z} + \frac{\sqrt{2}-1}{2(1-z/p_1)} - \frac{\sqrt{2}+1}{2(1-z/p_2)}.
\end{aligned}
$$

One expands this again into a power series and compares coefficients to compute the value of x_t as claimed.

For the rest of this proof, let

$$\varepsilon_t = \frac{1-\sqrt{2}}{2p_1^t} + \frac{1+\sqrt{2}}{2p_2^t}.$$

We first show that $\varepsilon_t \geq 0$ for all t. Assuming the contrary is equivalent to

$$(1+\sqrt{2})^2 < (p_2/p_1)^t.$$

By direct computation one verifies that

$$0 < p_2/p_1 = \frac{8m-4-2\sqrt{2}}{8m-4+2\sqrt{2}} < 1.$$

Thus

$$(p_2/p_1)^t \leq 1 < (1+\sqrt{2})^2,$$

a contradiction! Now assume that $\varepsilon_{t+1} \geq \varepsilon_t$ for some t, $t \geq 1$. This is equivalent to

$$(p_2/p_1)^t \geq (1+\sqrt{2})^2 \cdot \frac{p_1(p_2-1)}{p_2(p_1-1)}.$$

One computes that

$$(1+\sqrt{2})^2 \cdot \frac{p_1(p_2-1)}{p_2(p_1-1)} = 1$$

which implies

$$(p_2/p_1)^t \geq 1.$$

This is impossible for $t \geq 1$. Finally, it is clear that $\lim_{t\to\infty} \varepsilon_t = 0$. □

The lemma guarantees that $t_0(\varepsilon, m)$ exists and can be used as a bound on the length of a test sequence required to guarantee fault detection with a probability no less than $1 - \varepsilon$.

To indicate that p_1 and p_2 of Lemma 8.8 depend on m we write $p_1(m)$ and $p_2(m)$, respectively. Let

$$\varepsilon_t(m) = \frac{1-\sqrt{2}}{2p_1(m)^t} + \frac{1+\sqrt{2}}{2p_2(m)^t}.$$

The following table lists a few numerical results with t of the form $t = lm$:

m	$p_1(m) \approx$	$p_2(m) \approx$	l	$\varepsilon_{lm}(m) \approx$
1000	1.00085	1.00015	40	0.0034476
			48	0.0010683
			49	0.0009227
			50	0.0007970
1000000	1.00000085	1.00000015	40	0.0034491
			48	0.0010688
			49	0.0009232
			50	0.0007975

From these values, one obtains $t_0(10^{-3}, 10^6)/10^6 = 49$, for instance. It is interesting to note that the simulation results of [Ful] are quite close to this value.

In the following lemma we provide an approximation of $\varepsilon_{lm}(m)$ for large m. This approximation is then used to prove the existence of a bound on the length coefficient.

Lemma 8.9 *Consider random testing of stuck-at-0 faults of single-bit memory cells with equally likely actions. Let p_1 and p_2 be defined as in Lemma 8.8, and let*

$$\varepsilon_t(m) = \frac{1-\sqrt{2}}{2p_1(m)^t} + \frac{1+\sqrt{2}}{2p_2(m)^t}.$$

For every fixed integer l, $l > 0$, the sequence $\varepsilon_{lm}(m)$ converges to

$$\varepsilon^{(l)} = \lim_{m\to\infty} \varepsilon_{lm}(m) = \frac{1-\sqrt{2}}{2e^{l(\frac{1}{2}+\frac{\sqrt{2}}{4})}} + \frac{1+\sqrt{2}}{2e^{l(\frac{1}{2}-\frac{\sqrt{2}}{4})}}.$$

Moreover,

$$\lim_{l\to\infty} \varepsilon^{(l)} = 0.$$

Proof: The limits are obtained using classical formulae. □

Theorem 8.10 *Consider random testing of stuck-at-*0 *faults of single-bit memory cells with equally likely actions. Then the length coefficient* $t_0(\varepsilon, m)/m$ *is bounded for every* ε, $0 < \varepsilon < 1$.

Proof: Let l_0 be such that $\varepsilon^{(l)} < \varepsilon/2$ for all $l \geq l_0$. The fact that $\varepsilon_{l_0 m}(m)$ converges to $\varepsilon^{(l_0)}$ as $m \to \infty$ implies the existence of m_0 with $|\varepsilon^{(l_0)} - \varepsilon_{l_0 m}(m)| \leq \varepsilon/2$ for all $m \geq m_0$. Hence $\varepsilon_{l_0 m}(m) \leq \varepsilon$ for all $m \geq m_0$. Let

$$s = \max\{l_0, \max\{\lceil t_0(\varepsilon, m)/m \rceil \mid m < m_0\}\}.$$

Then $\varepsilon_{sm}(m) \leq \varepsilon$, and s is an upper bound for the length coefficient. □

The method used to prove the existence of a bound on the length coefficient for random testing of stuck-at faults is not easily generalized. It relies on obtaining an expression for the probability of detection and even on the special form of this expression. An obvious alternative would be, to try and use classical bounds obtained by the general theory of Markov chains and applied in Proposition 8.6. The following paragraphs illustrate this idea for the example above—they show that these methods will not lead to satisfactory results in general.

One computes the parameters γ and γ' of Proposition 8.6 as follows (see [Do]): Consider the smallest positive integer s such that

$$m_{1,1}^{(s)} + m_{1,2}^{(s)} < 1 \quad \text{and} \quad m_{2,1}^{(s)} + m_{2,2}^{(s)} < 1$$

where $m_{i,j}^{(s)}$ is the (i,j)-entry of M^s. Let ϑ be the larger of these two sums. In the example, $s = 2$ and

$$\vartheta = \max\{1 - \frac{1}{8m^2}, 1 - \frac{1}{m} - \frac{3}{8m^2}\} = 1 - \frac{1}{8m^2}.$$

Now let

$$\gamma = \vartheta^{1/s} = \sqrt{1 - \frac{1}{8m^2}}$$

and

$$\gamma' = \vartheta^{-1} = \gamma^{-2}.$$

Then by [Do],

$$m_{i,1}^{(t)} + m_{i,2}^{(t)} \leq \gamma' \gamma^t$$

for $i = 1, 2$ and, consequently,

$$\mu_t(\{1,2\}) \leq \gamma' \gamma^t$$

for $t \geq s$.

Thus, the number of steps t required to detect the presence of a fault with a probability no less than $1 - \varepsilon$ is bounded by

$$t'_0(\varepsilon, m) = \frac{\log \varepsilon - \log \gamma'}{\log \gamma} = \frac{\log \varepsilon}{\log \gamma} + 2$$

from above.

It turns out that this upper bound on $t_0(\varepsilon, m)$ is not tight enough to prove the boundedness of the length coefficient. In fact, $t'_0(\varepsilon, m)/m \to \infty$ as $m \to \infty$. Numerical calculations show that $t'_0(\varepsilon, m)$ is so much larger than $t_0(\varepsilon, m)$ that it is useless even for practical purposes, when the limiting behaviour might be less relevant. A lower bound $t''_0(\varepsilon, m)$ on $t_0(\varepsilon, m)$ can be derived in a similar fashion. That bound is not tight enough either; it yields $t''_0(\varepsilon, m)/m \to 0$ as $m \to \infty$.

These observations seem to indicate that classical bounds of Markov chain theory will not yield acceptable approximations of the length coefficient. New tools have to be developed to deal with this task in general.

We close with a general remark about random testing: While testing for stuck-at faults with random sequences is clearly inefficient, the following argument seems to be convincing [Fu1]. Suppose we need to test for many types of complex faults. In deterministic testing, one may have to apply an exhaustive test for each fault type in the worst case. In randomized testing, the test would have to be applied only for the length required by the "worst" of the faults.

9. Concluding Remarks

In summary, we have developed a precise mathematical model of the testing process along with algorithms for its construction. The methods presented apply to both deterministic and random testing. Given this model, we are now in a position to study questions about testing of sequential circuits within the framework of the classical theories of finite automata and stochastic processes.

References

[Abr] J. A. Abraham, V. K. Agarwal: Test Generation for Digital Systems. [Pr], 1–90.

[Do] J. L. Doob: *Stochastic Processes.* John Wiley & Sons, Inc., New York, 1953.

[Fe] W. Feller: *An Introduction to Probability Theory and Its Applications.* John Wiley & Sons, Inc., New York, 1968 (3rd edition).

[Fu1] A. Fuentes, R. David, B. Courtois: Random Testing versus Deterministic Testing of RAMs. *Proceedings of the 16th International Symposium on Fault Tolerant Computing Systems,* IEEE, 1986, 266–271.

[Fu2] A. Fuentes: *Contribution à l'étude du test aléatoire de memoires RAM.* Thèse, Institut National Polytechnique de Grenoble, 1986.

[Fuj] H. Fujiwara: *Logic Testing and Design for Testability.* MIT Press, 1985.

[Ha] J. P. Hayes: Detection of Pattern-Sensitive Faults in Random-Access Memories. *IEEE Trans. Computers* **C-24** (1975), 150–157.

[He1] F. C. Hennie: *Finite-State Models for Logical Machines.* John Wiley & Sons, New York, 1968.

[He2] F. C. Hennie: Fault-Detecting Experiments for Sequential Circuits. *Proceedings of the 5th Annual Symposium on Switching Theory and Logical Design,* IEEE, 1964, 95–110.

[Mc] W. H. McAnney, P. H. Bardell, V. P. Gupta: Random Testing for Stuck-at Storage Cells in an Embedded Memory. *Proceedings of the 1984 International Test Conference,* IEEE, 1984, 157–166.

[Mo] E. F. Moore: Gedanken-Experiments on Sequential Machines. *Automata Studies,* 129–153. Princeton University Press, Princeton, N.J., 1956.

[Po] J. F. Poage, E. J. McCluskey: Derivation of Optimum Test Sequences for Sequential Machines. *Proceedings of the 5th Annual Symposium on Switching Theory and Logical Design,* IEEE, 1964, 121–132.

[Pr] D. K. Pradhan (ed.): *Fault-Tolerant Computing.* 2 volumes, Prentice-Hall, Englewood Cliffs, N.J., 1986.

[St] P. H. Starke: *Abstract Automata.* North-Holland Publ. Co., Amsterdam, 1972.

[Wo] D. Wood: *Theory of Computation.* Harper & Row, New York, 1987.

Printing Requisition/Graphic Services

14115

1. **Please complete unshaded areas** on form as applicable.
2. **Distribute copies** as follows: White and Yellow to Graphic Services. **Retain Pink Copies for your records.**
3. On completion of order the **Yellow copy will be returned** with the printed material.
4. **Please direct enquiries, quoting requisition number and account number,** to **extension 3451.**

TITLE OR DESCRIPTION
A Model for Sequential Machine Testing CS-88-12

DATE REQUISITIONED: April 7/88
DATE REQUIRED: ASAP Please
ACCOUNT NO.: 1 2 6 6 3 1 8 4 1

REQUISITIONER- PRINT: J.A. Brzozowski
PHONE: x4441
SIGNING AUTHORITY: Yvonne Fink / gaB

MAILING INFO –
NAME: Sue DeAngelis
DEPT.: Computer Science
BLDG. & ROOM NO.: MC 6081E
☒ DELIVER
□ PICK-UP

Copyright: I hereby agree to assume all responsibility and liability for any infringement of copyrights and/or patent rights which may arise from the processing of, and reproduction of, any of the materials herein requested. I further agree to indemnify and hold blameless the University of Waterloo from any liability which may arise from said processing or reproducing. I also acknowledge that materials processed as a result of this requisition are for educational use only.

NUMBER OF PAGES: 38
NUMBER OF COPIES: 100

TYPE OF PAPER STOCK
□ BOND □ NCR ____ PT. ☒ COVER □ BRISTOL □ SUPPLIED □ Alpac Ivory 140M

PAPER SIZE
□ 8½ x 11 □ 8½ x 14 □ 11 x 17 □ 10x14 Glosscoat (stock) 10 pt.

PAPER COLOUR
□ WHITE □ ________

INK
☒ BLACK □ Rolland Tint

PRINTING
□ 1 SIDE ____ PGS. ☒ 2 SIDES ____ PGS.

NUMBERING
FROM ____ TO ____

BINDING/FINISHING
☒ COLLATING □ STAPLING □ ____ HOLE PUNCHED □ PLASTIC RING

FOLDING/PADDING: 7x10 saddle stitched
CUTTING SIZE:

Special Instructions

SPECIAL BEAVER COVER WITH BLACK INK FORMAT

COPY CENTRE — OPER. NO. / BLDG. / MACH. NO.

DESIGN & PASTE-UP — OPER. NO. / TIME / LABOUR CODE

			D01
			D01
			D01

TYPESETTING — QUANTITY

PAP000000				T01
PAP000000				T01
PAP000000				T01

PROOF

PRF				
PRF				
PRF				

NEGATIVES — QUANTITY / OPER. NO. / TIME / LABOUR CODE

FLM				C01
FLM				C01
FLM				C01
FLM				C01
FLM				C01

PMT

PMT				C01
PMT				C01
PMT				C01

PLATES

PLT				P01
PLT				P01
PLT				P01

STOCK

				001
				001
				001
				001

BINDERY

RNG				B01
RNG				B01
RNG				B01
MIS000000				B01

OUTSIDE SERVICES

$ ____ COST

TAXES – PROVINCIAL □ FEDERAL □

Printing Requisition/Graphic Services

12161

1. Please complete unshaded areas on form as applicable.
2. Distribute copies as follows: White and Yellow to Graphic Services. Retain Pink Copies for your records.
3. On completion of order the Yellow copy will be returned with the printed material.
4. Please direct enquiries, quoting requisition number and account number, to extension 3451.

TITLE OR DESCRIPTION CS-88-12 A Model for Sequential Machine Testing J.A. Brzozowski

DATE REQUISITIONED May 7, 1990

DATE REQUIRED 2-3 weeks

ACCOUNT NO. 1 2 6 6 3 1 8 4 1

REQUISITIONER- PRINT Colleen Bernard

PHONE 2192

SIGNING AUTHORITY ✓

MAILING INFO – NAME Colleen Bernard DEPT. CS BLDG. & ROOM NO. DC-2314 ☒ DELIVER □ PICK-UP

Copyright: I hereby agree to assume all responsibility and liability for any infringement of copyrights and/or patent rights which may arise from the processing of, and reproduction of, any of the materials herein requested. I further agree to indemnify and hold blameless the University of Waterloo from any liability which may arise from said processing or reproducing. I also acknowledge that materials processed as a result of this requisition are for educational use only.

NUMBER OF PAGES 38

NUMBER OF COPIES 10

TYPE OF PAPER STOCK Alpac Ivory 140M

□ BOND □ NCR ____ PT. □ COVER □ BRISTOL □ SUPPLIED □ ________

PAPER SIZE 10x14 Glosscoat

□ 8½ x 11 □ 8½ x 14 □ 11 10 pt Rolland Tint

PAPER COLOUR

□ WHITE ☒ ________

INK

☒ BLACK □ ________

PRINTING

□ 1 SIDE ____ PGS. ☒ 2 SIDES ____ PGS.

NUMBERING FROM TO

BINDING/FINISHING

☒ COLLATING □ STAPLING □ ____ HOLE PUNCHED □ PLASTIC RING

FOLDING/PADDING 7x10 saddle stitched

CUTTING SIZE

Special Instructions

Please provide with Beaver covers.

Black ink on covers and inside.

COPY CENTRE OPER. NO. BLDG. MACH. NO.

DESIGN & PASTE-UP

OPER. NO.	TIME	LABOUR CODE
		D 0 1
		D 0 1
		D 0 1

TYPESETTING

	QUANTITY			
P A P 0 0 0 0 0				T 0 1
P A P 0 0 0 0 0				T 0 1
P A P 0 0 0 0 0				T 0 1

PROOF

P R F				
P R F				
P R F				

NEGATIVES

	QUANTITY	OPER. NO.	TIME	LABOUR CODE
F L M				C 0 1
F L M				C 0 1
F L M				C 0 1
F L M				C 0 1
F L M				C 0 1

PMT

P M T				C 0 1
P M T				C 0 1
P M T				C 0 1

PLATES

P L T				P 0 1
P L T				P 0 1
P L T				P 0 1

STOCK

				0 0 1
				0 0 1
				0 0 1
				0 0 1

BINDERY

R N G				B 0 1
R N G				B 0 1
R N G				B 0 1
M I S 0 0 0 0 0				B 0 1

OUTSIDE SERVICES

$ ________ COST

TAXES – PROVINCIAL □ FEDERAL □