Design of diagnosable sequential machines

by ZVI KOHAVI*

Project MAC and Department of Electrical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts

and

PIERRE LAVALLEE*

Applied Research Laboratories
Xerox Corporation
Rochester, New York

INTRODUCTION
A diagnosable sequential machine is one which possesses a distinguishing (or diagnosing) sequence(s) and thus permits us to uniquely identify the various states of the machine by inspecting its response to the distinguishing sequence.

The problem of determining the properties of a synchronous sequential machine by observing its response to various input sequences was introduced by Moore. In his paper Moore further considered the problem of determining whether a given machine accurately describes its terminal behavior as specified by a state table or a state diagram. His approach, however, has very little practical significance since it leads into extremely long experiments. A different procedure for the design of fault detection experiments for sequential machines was introduced by Hennie. This procedure yields good results for machines which possess distinguishing sequences and when the actual circuit has no more states than the correctly operating circuit. For machines which do not have any distinguishing sequences, Hennie's procedure yields very long experiments which makes them impractical. Further development of this approach was done by Kime. From the nature of the problem it seems that the design of fault detection experiments for arbitrary sequential machines will always lead into lengthy experiments which are extremely hard to apply in any practical situation. With the increasing use of modules and integrated circuits it becomes necessary, however, to be able to determine from terminal experiments whether or not a given circuit operates properly. More effort must be made to design circuits which are easy to maintain and to which simple, and practical fault detection experiments can be designed.

The objective of this paper is to present a method for designing sequential circuits in such a way that they will be made to possess special distinguishing sequences and to which there exist very short fault detection experiments. In order to obtain these special and important properties we have to modify the original design and to add additional output logic. Our aim is to determine the minimal amount of additional output logic which is necessary in order to obtain these special properties.

It should be emphasized that the approach presented in this paper yields experiments which are applied only to the terminals of the circuit and not to any point within the circuit. The terminals, however, are predesigned to enable efficient maintenance of the circuit.

The sequential machines considered in this paper are assumed to be finite-state, synchronous, deterministic, strongly connected and completely specified. The machines are of the Mealy model, where the output is a function of both the state and the input.

As an experiment on a machine we define the application of input sequences to the input terminals and the recording of the corresponding response from its output terminals. If the experiment is de-
signed to take the machine through all possible transitions in such a way that a definite conclusion can be reached whether or not the machine operates correctly, it is said to be a fault detection experiment. At the beginning of an experiment the machine is said to be in the starting state. The experiments discussed in this paper are simple and preset, i.e., it is assumed that only a single copy of the machine is available to the experimenter and that the entire input sequence is predetermined, independently of the outcome of the experiment. An extensive discussion of the various experiments can be found in Gill. 5

**Definition 1.**
Let $M$ be a sequential machine having $n$ states. An input sequence $x_0$ is said to be a distinguishing sequence (or a diagnosing sequence) if, when applied to $M$ it yields $n$ different output sequences depending on the initial states. Hence, by observing the response of $M$ to $x_0$ and if $M$ operates correctly, the initial state of $M$ at the start of $x_0$ can be determined.

**Definition 2.**
An input sequence $y_n$ is said to be a homing sequence if the response of $M$ to its application uniquely determines the final state of the machine independently of the starting state. Every reduced sequential machine possesses a homing sequence while only a limited number of machines have distinguishing sequences. Every distinguishing sequence (DS) is also a homing sequence (HS) while the converse is not true.

Let $S$ be the set of states of machine $M$. An admissible set is any subset of $S$ (including $S$ itself) which is known to contain the starting state.

A five state machine $M$ is represented by Table I. It is obvious that $M$ does not have any DS since both states 1 and 2 under 0 input map into state 1 and produce an output of 0, while states 2 and 5 under 1 input map into state 5 with a 1 output.

<table>
<thead>
<tr>
<th>Table I — Machine $M$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>P.S.</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

Our objective is to obtain a machine $M'$, which contains $M$, by adding some output logic to $M$ such that $M'$ will possess any arbitrary predescribed distinguishing sequence, or sequences. In order to arrive, in a systematic manner, to a solution which requires the least amount of additional output terminals the following procedure is proposed.

**Definitely diagnosable machines.**
The state table of $M$ may be written as illustrated in the top half of Table II. The pair $x/z$ corresponds to input $x$ and output $z$. The entries of the table are the "next states" corresponding to every input-output pair. For example, from state 1 under input 1 the machine goes to state 4 with an output of 1. This is denoted by entering a 4 in column 1/1 and a dash (—) in column 1/0. In a similar manner the next states of $M$ are entered into the testing table.

The lower half of the testing table consists of all possible admissible pairs and their implications. For example, corresponding to admissible pair 12 is a pair 11 under column 0/0 and a pair 45 under column 1/1. The lower half of the table is derived in a straightforward manner from the top half. The implied entries represent the conditions under which the admissible pairs are distinguishable. Pair 13 is distinguishable by an experiment starting with a 0 if and only if pair 15 is, while pair 12 is indistinguishable by such an experiment since it implies a repeated pair 11. An admissible pair which does not imply any other pair, i.e., all the entries in the corresponding row are dashes (see pair 14), can be omitted from the table (pairs 24 and 35 have therefore been omitted from the testing table). Whenever an entry in the testing table consists of a repeated state (11 in row 12), that entry is circled. A circle around 11 implies that both states 1 and 2 are merged under input 0 into state 1 and hence are indistinguishable by any experiment starting with an input 0. In order to obtain a machine $M'$ which contains $M$ and possesses a DS which starts with a zero, at first we have to add an output terminal and assign different output states to each state which implies a circled entry.

The testing graph is derived from the testing table. Each node corresponds to an admissible pair. If an admissible pair implies another pair (13 implies 15 in Table II) a directed branch is drawn leading from the admissible pair into the implied pair. Only implied pairs which are not circled (i.e., without repeated states) need to be considered for the derivation of the graph.

The graph for machine $M$ which consists of seven nodes is given in Figure 1. The labelling of the branches corresponds to the top headings of the testing table. An inspection of the graph reveals that it contains a loop 45-23-15-45, i.e., starting from the pair 45 the input sequence 001 does not distinguish
**Definition 3**

A sequential machine $M$, having $n$ states, is said to be **definitely diagnosable** (DD) if any sequence of length $\varphi \leq n(n-1)/2$ is a DS for the admissible set $S = (S_1, S_2, \ldots, S_n)$.

If no repeated states (circled entries) exist in the testing table then there are no two states in $M$ which map into the same next state and yield identical outputs. A loop-free graph guarantees that the longest path includes, at most, $\leq n-1$ nodes corresponding to all possible admissible pairs. Hence, the following theorem results.

**Theorem 1**

The necessary and sufficient conditions for a sequential machine $M$ to be definitely diagnosable is that the corresponding testing graph is loop-free and that no repeated states (circled entries) exist in the testing table.

Theorem 1 defines the conditions under which a sequential machine $M$ is DD. Machine $M$ is therefore not DD since 11 and 55 exist in the testing table and the testing graph is not loop-free. To obtain a corresponding machine $M'$ such that $M'$ contains $M$ and is DD we have to augment the output terminals of $M$. The procedure for modifying the output is summarized as follows:

(i) Eliminate all the circled entries by assigning different output states to the corresponding next state entries.

(ii) Open all the loops of the testing graph by eliminating the smallest number of branches in the graph. A branch is eliminated by assigning different output states to the next state entries which are covered by the node into which it terminates. The choice of branches for elimination is based on two criteria. The first criterion is the reduction of the paths in the graph, i.e., a branch is chosen for elimination if it opens a loop and in addition opens some paths in the graph. The aim is to minimize the length of the longest path in the graph. The second criterion is the minimization of the number of eliminated branches.

We shall apply this procedure to machine $M$.

Step (i) is accomplished by assigning to the outputs associated with the next state entries 1, in column $x=0$ and states 1 and 2, the values 00 and 01, respectively. Similarly from the circled entry 55 in the testing table we conclude that the outputs associated with the next state entries of states 2 and 5 (column $x=1$) must be modified to be 10 and 11, respectively. These steps ensure that the testing table of machine $M'$ is free of repeated entries. Step (ii) is accomplished by opening the loop in the graph. This can be done by eliminating any one of the three branches in the loop. However, the cancellation of the branch from 15 to 45 opens an additional path (13-15-45). After this elimination the longest path includes four nodes (12-45-23-15), this will result in a very long DS and hence, an attempt is made to open this path by the elimination of the branch 45-23. The first branch from 15 to 45 is labeled 1/1, hence states 1 and 5 go, under input 1, into states 4 and 5 respectively with an output of 1. State 5 has a modified output 11 (from the application of rule i), therefore state 1 must be assigned an output of 10.

In the same manner the second branch labeled 0/1 from 45 into 32 may be opened by assigning the outputs associated with next states 3 and 2 with distinct outputs 10 and 11, respectively.

In the resulting table for machine $M'$ (Table III), we note that three output entries are incompletely specified. Their specification can be made accord-
The preceding procedure did not affect the next-state or the output behavior of machine M since the output Z of machine M' is identical to Z of machine M. The augmentation of M results in an addition of the output terminal Z1. The main advantage of M' over M is in the fact that M' is DD and thus efficient checking experiments can be constructed for fault detection in M' while this is not so easy for machine M.

**Theorem 2**

Let the longest path in a loop-free testing graph of machine M be m (i.e., the longest path contains m branches) and let the testing table be free of repeated states, then any sequence whose length is, at most, m+1 symbols is a DS on M.

*Proof:* If the m input labels on the longest path in the testing graph are chosen in their order of appearance as a DS, only the two states (node in the graph) from which this path originates will not be distinguished or yield different responses. The path terminates in a node which corresponds to some pair of states, any other input symbol added to the above sequence will distinguish between these states since no other pair of states is implied.

In machine M' the longest path in the testing graph (after the elimination of the branches according to steps i and ii) consists of a single branch, hence the DS is of length 2. This is illustrated by the diagnosing tree in Figure 2, from which it is evident that any sequence of length two is a DS on M'.

**Theorem 3**

To every strongly connected sequential machine M there corresponds another DD sequential machine M', which is obtained by an addition of some output terminals to M. The original machine may or may not possess any DS.

The proof of this theorem is obvious from the preceding discussion and from the construction of M'.

For any 2^n-states machine we need, at most, n additional output terminals. However, experience indicates that one or two additional terminals will be sufficient for most machines.

The block diagram of the desired solution is given in Figure 3.

The technique presented in this section is applicable with minor modifications, to the cases of Information Lossless (IL), Information Lossless of Finite order (ILF) and Finite Memory automata, i.e., to every sequential machine there exists a corresponding machine which is IL and ILF. Similarly to every arbitrary sequential machine there corresponds a finite memory machine. The respective machines are obtained by modifying the output logic of the original ones.

**A procedure for the design of checking experiments**

The construction of checking experiments for machines which do not possess a DS or simple locating sequences is extremely difficult. Even when some locating sequences can be found the length of the experiment is so great that it becomes almost impractical to apply. The bound on the length of such experiments given by Hennie2 for the general case is m^n(n+1)!. This is a result which makes the checking experiment impractical to apply repeatedly to any machine. In the previous section we showed that every machine can be modified to become a DD machine. In Figure 3
this section we outline a procedure for the construction of checking experiments for DD and a more general class of sequential machines. The experiments are relatively short and it seems that the ease of maintenance and fault detection more than compensates for the minor amount of added logic. We therefore propose to predesign sequential machines with certain DS's by which simple checking experiments can be constructed without the need to obtain any information from the interconnections within the machine.

Let $X_o$ be a DS, of length $L$, for $M$. Define $T(S_i, S_j)$ to be a transfer sequence which takes $M$ from state $S_i$ into state $S_j$.

The proposed procedure for the construction of checking experiments involves the use of DS's with repeated symbols, i.e., 000 or 111 etc. For machine $M'$ choose $X_o=00$. Assume that $M'$ is in a starting state 1 at the beginning of the experiment. If it is not in 1 it is always possible to bring it into 1 since every sequence of length 2 is also a HS. In order to simplify the notation we shall use the decimal value of the output, hence 10=2, 11=3 etc.

The experiment starts with the application of the DS $X_o=00$ to ascertain that $M'$ is actually in state 1. According to the state table the machine remains in state 1 under 0 input. To verify that 1 is a stable state apply another 0 input following the first $X_o$. The input to $M'$ at this stage, consists of three consecutive zeros. The first two zeros serve to check the initial state while the last two zeros (which are the same DS) serve to check a transition from 1 under input 0. The machine remains in state 1 until an input 1 is applied, followed by $X_o$ to check the transition from 1 to 4. Provided that $M$ operates correctly a 1 input takes $M$ to state 4 with an output of 2. $X_o$ takes $M$ into state 5 through state 3. To check the transitions from 4 to 3 and 3 to 5 we keep applying 0 inputs. As long as the machine goes through new transitions we keep applying the same 0 inputs. When a new transition cannot be obtained with an input of 0, a perturbation by means of a 1 input is applied followed by the DS $X_o$. Assuming that the don't care entries are zeros the checking experiment at this point is as follows:

Input: 0 0 0 0 1 0 0 0 0 0
State: 1 1 1 1 4 3 5 2 1 1 1
Output: 0 0 0 2 0 3 1 0 0

It is evident that an application of another input of 0 does not yield any new transition and hence we must apply an input of 1. Provided we can verify by the end of the experiment that the studied machine contains five states and operates in accordance with Table III, the above experiment serves as a check on the transitions from states 1,2,3,4 and 5 under input of 0 and from state 1 under an input of 1. An application of 1 at this stage of the experiment will not yield any new transition since it takes $M$ into state 4 while the transition from 1 to 4 has already been checked. Hence a transfer sequence is needed to take the machine from state 1 into some state $S_j$ such that the machine goes through "checked" transitions only. This guarantees that the machine actually terminates is $S_j$ if it has operated correctly up to this stage. Apply $T(1,4)=1$ followed by another 1 and $X_o$. At the end of this part $M$ is in state 5. Since every transition under 0 input has already been checked a 1 input is applied followed by $X_o$. This part of the experiment is as follows:

Input: 1 1 0 0 1 0 0
State: 1 4 4 3 5 5 2 1
Output: 2 0 2 0 3 3 1

At this point the only unchecked transitions are from states 3 and 2 under 1 input. Hence, apply $T(1,3)=10$ followed by 1 and $X_o$. This part leaves the machine in state 1. $T(1,2)=1000$ is next applied followed by 1$X_o$. The complete checking experiment for $M'$ requires 29 input symbols and is given as follows:

An inspection of the input-output symbols reveals that there are at least 5 states to the machine under examination, since there are five different responses to the application of the input sequence 00, i.e., responses 00, 20, 03, 31, 10. Similarly it is straightforward to show that a machine that satisfies this input-output relations and has five states must be identical to the state table given in Table III except for relabelling its states. In the same manner we could have constructed an experiment using $X_o=11$ as a DS.

The upper bound on the length of these experiments is

$$\varphi = n.m + n(m-1).L + L + (m-1) \cdot (n-1)^2$$

where $n$ and $m$ correspond, respectively, to the number of states and distinct inputs in $M$. This bound is smaller by far than any other known algorithm for the construction of fault detection experiments for sequential machines.

CONCLUSIONS

The testing graph and the associated technique of branch cutting by the addition of output logic has been introduced as a method to embed any sequential ma-
chine in a definitely diagnosable machine. It can be shown that this technique also applies in embedding a sequential machine which is not IL or ILF into one which has this property; the same applies to the finite memory property. The method answers the question of what is the minimum necessary additional output logic that is needed to obtain a machine with specific properties. This concept is applied and specialized to the problem of checking experiments, where the property that the sequential machine which embeds the original one should have a distinguishing sequence which consists of repeating the same input symbol a certain number of times; this requirement was seen to be a special case of a machine which is definitely diagnosable, and techniques were presented to obtain such properties. It must be emphasized that the compromise that has been adopted in this paper (added logic versus complex and lengthy experiments) is not a very costly one and provides for the first time, a working tool for input-output black box types of experiments as presently the situation exists for integrated circuits. It is felt, however, that more effort is needed in order to better utilize the properties of the DD machines and the numerous distinguishing sequences available. For a longer range study it seems that the properties of the DD machines should be utilized to obtain checking experiments which not only tell us that something is wrong with the machine but ones which indicate what is wrong with it, i.e., some type of error correcting checking experiments.

ACKNOWLEDGMENT

Work reported herein was supported in part by Project MAC, an M.I.T. research program sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Number Nonr-4102(01).

REFERENCES

1 E F MOORE
   Gedanken-experiments on sequential machines
   Automata Studies C E Shannon and J McCarthy editors
   pp 129-153 1956
2 F C HENNIE
   Fault detecting experiments for sequential circuits
   Proceedings of the Fifth Annual Symposium on Switching
   Theory and Logical Design pp 95-110 1964
3 C R KIME
   A failure detection method for sequential circuits
   Department of Electrical Engineering University of Iowa
   Technical Report # 66-13 January 1966
   Also IEEETEC pp 113-115 February 1966
4 G H MEALY
   A method for synthesizing sequential circuits
   Bell Sys Technical Journal Vol 34 pp 1045-1080
   September 1955
5 A GILL
   Introduction to the theory of finite-state machines
   McGraw Hill Book Company 1962