Design of Diagnosable Sequential Machines
Utilizing Extra Outputs

HIDEO FUJIWARA, STUDENT MEMBER, IEEE, AND KOZO KINOSHITA, MEMBER, IEEE

Abstract—This paper is concerned with the problem of designing easily testable sequential machines, output-observable machines, for which there exist very short checking experiments. A sequential machine for which any initial state can be uniquely determined only by the output response is said to be output-observable. An algorithm is developed to modify a given machine to an output-observable one by adding a minimum number of extra outputs. This method is based on the fact that the output-observable realization of a given machine M exists if and only if M is semi-FSR realizable (a special type of feedback shift register realization).

For the k-output-observable sequential machine, we can find a checking sequence ω₀|ω₁ such that ω₀ is an input-output sequence which passes through all the transitions of the given state table and ω₁ is an arbitrary input-output sequence of length k. Since a checking sequence must pass through all the transitions of the given state table, the length of the checking sequence ω₀|ω₁ is nearly minimum.

Index Terms—Checking experiments, diagnosable sequential machines, fault detection, output-observable machines, semi-feedback shift register (FSR) realizability.

I. INTRODUCTION

THE PROBLEM considered here is the design of easily testable sequential machines for which there exist very short checking experiments. A checking experiment on a sequential machine is the application of input sequences to the input terminals and observation of the output sequences at the output terminals to determine whether or not the machine is operating correctly. An approach to the design of checking experiments, called the transition checking approach, was first introduced by Hennie [1]. However, for machines without distinguishing sequences, his procedure yields very long test sequences. Hence for machines that do not have any
TABLE I
Machine \( M_1 \)

<table>
<thead>
<tr>
<th>P.S.</th>
<th>N.S.</th>
<th>( z_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4, 0</td>
<td>2, 0</td>
</tr>
<tr>
<td>2</td>
<td>3, 0</td>
<td>2, 0</td>
</tr>
<tr>
<td>3</td>
<td>4, 1</td>
<td>3, 1</td>
</tr>
<tr>
<td>4</td>
<td>5, 1</td>
<td>5, 1</td>
</tr>
<tr>
<td>5</td>
<td>5, 1</td>
<td>1, 1</td>
</tr>
</tbody>
</table>

distinguishing sequences, new approaches are proposed to this problem. One approach is to modify a given machine by adding extra inputs \([2] - [4]\) or outputs \([5] - [10]\) so that the modified machine has a distinguishing sequence. For an \( n \)-state, \( m \)-input symbol machine, these procedures give a bound on the length of checking sequences that is approximately \( mn^2 \). Therefore, for machines with a large number of states, these procedures yield very long experiments, which make them impractical.

In order to overcome this, we introduce the output-observable sequential machines which have checking sequences of short length. For a \( k \)-output-observable sequential machine, we can find a checking sequence \( \omega_k \) such that \( \omega_1 \) is an input-output sequence which passes through all the transitions of the given state table and \( \omega_k \) is an arbitrary input-output sequence of length \( k \). Since a checking sequence must pass through all the transitions of the given state table, it is shown that the procedure of organizing checking sequences is simple and systematic.

The first half of this paper describes a method for the modification of a given machine to an output-observable one by adding a minimum number of extra outputs. Our aim is to determine the minimal amount of additional output logic which is necessary and sufficient to obtain the property of output observability. This method is based on the fact that the output-observable realization of a given machine \( M \) exists if and only if \( M \) is semi-feedback shift register (FSR) realizable. The second half of this paper presents a procedure for the design of very short checking experiments for the output-observable sequential machines.

II. OUTPUT-OBSERVABILITY AND SEMI-FSR REALIZABILITY

A sequential machine \( M \) will be represented by a quintuple \( M = (S, I, Z, \delta, \lambda) \) where \( S \) is a finite set of states, \( I \) is the input alphabet, \( Z \) is the output alphabet, \( \delta: S \times I^* \rightarrow S \) is the next state function, and \( \lambda: S \times I^* \rightarrow Z^* \) is the output function. The sequential machines considered in this paper are assumed to be reduced and strongly connected Mealy machines such that binary codes are already assigned to their output symbols, i.e., the output function \( \lambda \) is represented by a direct product \( z_1 \times \cdots \times z_p \) of binary output functions \( z_1, \cdots, z_p \).

A partition on a set of states \( S \) is a collection of disjoint subsets of \( S \), called blocks, such that their set union is \( S \). A relation \( \gamma \) on \( S \) corresponding to a partition \( \pi \) is a relation such that \( S_i \sim S_j \) for \( S_i, S_j \in S \) if and only if \( S_i \) and \( S_j \) belong to the same block of \( \pi \). \( \pi \) is the partition on \( S \) such that \( S_i \sim S_j \) if and only if \( S_i \sim S_j \) and \( S_i \sim S_j \), \( \pi + \tau \) is the partition on \( S \) such that \( S_i \sim S_j \) if only if \( S_i \sim S_j \) or \( S_i \sim S_j \). The following two partitions are called trivial: the zero \( \equiv \) identity partition \( 0 \equiv I \) in which all elements of \( S \) form one block, and the zero \( \equiv \) identity partition \( 0 \equiv I \) in which every block is a singleton. \( \pi \) is a refinement of \( \tau \) (denoted by \( \pi \leq \tau \)) if and only if \( S_i \sim S_j \) implies \( S_i \sim S_j \).

The transition graph (defined in Nichols \[16\]) of a partition \( \pi \) is a graph in which each vertex corresponds to a block of \( \pi \) and there is an arc from vertex \( i \) to vertex \( j \) if and only if there is a state \( S_i \in B_i \) (\( B_i \) is the block of \( \pi \) corresponding to vertex \( i \)) and an input \( I \) such that \( \delta(S_i, I) = S_j \in B_j \).

A partition \( \pi \) is a shift register partition (SRP) \[16\] if and only if the transition graph of \( \pi \) is a subgraph of some Good diagram \[16\]. \( \pi \) has length \( l \) if it is a subgraph of the Good diagram of an \( l \)-stage shift register.

As an example, consider machine \( M_1 \) given by Table I and a partition \( \pi = [1, 2, 3, 4, 5] \). We obtain the transition graph shown in Fig. 1(a). This transition graph is a subgraph of the Good diagram of a two-stage shift register shown in Fig. 1(b). Therefore, the partition \( \pi = [1, 2, 3, 4, 5] \) is an SRP.

The following theorem follows directly from the definition of SRP.

Theorem 1 (Nichols \[11\]): A realization of \( M \) using a shift register of length \( l \) exists if and only if \( M \) has an SRP of length \( l \).

Note that in this realization, each state is given a single coding and that the states grouped together in a block of the SRP are given the same coding in the corresponding shift register.

Suppose that we are given a list of SRP's of \( M \) and have found a realization using \( p \) shift registers of length \( k_1, k_2, \ldots, k_p \) (see Fig. 2). Then, for each shift register in the realization, there must be a corresponding SRP in the list. Let \( \pi_1, \pi_2, \ldots, \pi_p \) be the set of SRP's corresponding to the realization, and let \( \pi = \pi_1 \pi_2 \ldots \pi_p \). Then, in this realization, the states grouped together in a block of \( \pi \) are given the same coding in the corresponding \( p \) shift registers. We will call this partial FSR (feedback shift register) realization “semi-FSR realization,” and define it as follows.

Definition 1: Let \( Y_{ij}(j = 1, 2, \ldots, k_i; i = 1, 2, \ldots, p) \) be the internal state variables, and let \( Y_{ij}(t) \) be the value of \( Y_{ij} \) at time \( t \) where each \( k \) is a positive integer. A sequential machine \( M \) is called \( k_1, k_2, \ldots, k_p \)-semi-FSR realizable with respect to a partition \( \pi \) if the state machine\(^1\) of \( M \) can be realized with the state assignment which satisfies the following conditions.

\(^1\) State machine of \( M = (S, I, Z, \delta, \lambda) \) is defined as a triple \( M_0 = (S, I, \delta) \) (see [14]).
realization. A semi-FSR code coincides with the ordinary FSR code.

Fig. 1. (a) Transition graph for \( \pi = [1; 23; 4; 5] \) in machine \( M_1 \).
(b) Good diagram for a two-stage shift register.

Condition 1: \( Y_{ij}(t) = Y_{ij}(t-1) \) for \( j = 2,3,\ldots,k_i \) and \( i = 1,2,\ldots,p \).

Condition 2: \( S_i \not\sim S_j \) if and only if

\[
(y_{1i}^j, y_{2i}^j, \ldots, y_{ki}^j) = (y_{1i}^j, y_{2i}^j, \ldots, y_{ki}^j)
\]

where \( (y_{1i}^j, y_{2i}^j, \ldots, y_{ki}^j) \) denotes a binary code of the state assignment corresponding to state \( S_i \).

When \( \pi \) is the zero partition, the semi-FSR realizability coincides with the ordinary FSR realizability. Hence, the semi-FSR realization is a generalization of ordinary FSR realization.

As an example, consider machine \( M_1 \) given by Table I and a partition \( \pi = [1; 23; 4; 5] \). The state assignment shown in Table II satisfies the following conditions.

\[
\begin{array}{ccc}
\text{p.s.} & Y_1 & Y_2 \\
1 & 0 & 1 \\
2 & 0 & 0 \\
3 & 0 & 0 \\
4 & 1 & 0 \\
5 & 1 & 1 \\
\end{array}
\]

Condition 2: \( S_i \not\sim S_j \) if and only if \( (y_{1i}^1, y_{2i}^1) = (y_{1i}^2, y_{2i}^2) \) where \( (y_{1i}^1, y_{2i}^1) \) denotes a binary code corresponding to state \( S_i \).

Hence, \( M_1 \) is 2-semi-FSR realizable with respect to \( \pi \).

Lemma 1: A sequential machine \( M \) is \( k \)-semi-FSR realizable with respect to a partition \( \pi \) if and only if there is an SRP of length \( k \) in \( M \).

Proof: This can be proved immediately from Theorem 1, Definition 1, and the definition of SRP. Q.E.D.

Lemma 2: A sequential machine \( M \) is \( k_1, k_2, \ldots, k_r \)-semi-FSR realizable with respect to \( \pi \) if and only if there exist \( p \) partitions \( \pi_1, \pi_2, \ldots, \pi_p \) such that \( \pi = \pi_1 \pi_2 \cdots \pi_p \) and for each \( i \) \( (i = 1,2,\ldots,p) \) \( M \) is \( k_i \)-semi-FSR realizable with respect to \( \pi_i \) where each \( k_i \) is a positive integer.

Proof: Suppose that \( M \) is \( k_1, k_2, \ldots, k_r \)-semi-FSR realizable with respect to \( \pi \). From Definition 1, there exists a realization with state variables \( Y_{ij} \)'s satisfying the following conditions.

\[
Y_{ij}(t) = Y_{ij}(t-1) \quad \text{for} \quad 2 \leq j \leq k_i \quad \text{and} \quad l \leq i \leq p.
\]

Condition 2: \( S_i \not\sim S_j \) if and only if

\[
(y_{1i}^j, y_{2i}^j, \ldots, y_{ki}^j) = (y_{1i}^j, y_{2i}^j, \ldots, y_{ki}^j).
\]

Define the partition \( \pi_i \) \( (1 \leq l \leq p) \) corresponding to the relation \( \not\sim \) defined as \( S_i \not\sim S_j \) if and only if \( (y_{1i}^1, \ldots, y_{ki}^1) = (y_{1i}^2, \ldots, y_{ki}^2) \). Then, from this and (1), it is clear that \( M \) is \( k_i \)-semi-FSR realizable with respect to \( \pi_i \) \( (1 \leq l \leq p) \). Therefore, we have only to show that \( \pi = \pi_1 \pi_2 \cdots \pi_p \). From (2) we have that \( S_i \not\sim S_j \) if and only if

\[
(y_{1i}^1, \ldots, y_{ki}^1) = (y_{1i}^2, \ldots, y_{ki}^2).
\]

This equation holds if and only if \( (y_{1i}^j, \ldots, y_{ki}^j) = (y_{1i}^l, \ldots, y_{ki}^l) \) for all \( l \leq i \leq p \), i.e., \( S_i \not\sim S_j \) for all \( l \). Hence we have that \( S_i \not\sim S_j \) if and only if \( S_i \not\sim S_j \) for all \( l \). Therefore, \( \pi = \pi_1 \pi_2 \cdots \pi_p \).

The converse can be proved similarly by Definition 1.

Q.E.D.

Definition 2: A sequential machine \( M \) is called \( k_1, k_2, \ldots, k_r \)-output-observable with respect to the output function \( z_1 \times z_2 \times \cdots \times z_p \) and a partition \( \pi \) if the following conditions are satisfied, where each \( k_i \) is a nonnegative integer.
Condition 1: The knowledge of the present state of $M$ is sufficient to uniquely determine the succeeding output sequence of length $k_j$ observed at the output function $z_j$ for every $j$ ($j = 1, 2, \ldots, p$).

Condition 2: Let $\mu_i$ be the output sequence of length $k_i$ observed at $z_i$ when the initial state is $S_i$. Then $S_i \succeq S_j$ if and only if $(\mu_{i1}, \ldots, \mu_{ip}) = (\mu_{j1}, \ldots, \mu_{jp})$ for all $S_i$ and $S_j \in S$.

When $\pi$ is the zero partition, $M$ is called output-observable.

For example, a sequential machine $M_1$ shown in Table I is 1-output-observable with respect to $z_1$ and $\pi_1 = \{1, 2, 3, 4, 5\}$. A sequential machine $M_2$ shown in Table II is 1,2-output-observable with respect to $z_1 \times z_2$ and the zero partition, and thus $M_2$ is output-observable.

Lemma 3: A sequential machine $M$ is $k_1, k_2, \ldots, k_p$-output-observable with respect to the output function $z_1 \times z_2 \times \cdots \times z_p$ and a partition $\pi$ if and only if there exist $p$ partitions $\pi_1, \pi_2, \ldots, \pi_p$ such that $\pi = \pi_1 \times \pi_2 \times \cdots \times \pi_p$ for each $i$ ($i = 1, 2, \ldots, p$) $M$ is $k_i$-output-observable with respect to the output function $z_i$ and the partition $\pi_i$, where each $k_i$ is a nonnegative integer.

Proof: Suppose that $M$ is $k_1, k_2, \ldots, k_p$-output-observable with respect to $z_1 \times z_2 \times \cdots \times z_p$ and $\pi$. From Definition 2, the following conditions are satisfied.

Condition 1: The knowledge of the present state $S_i$ of $M$ is sufficient to uniquely determine the succeeding output sequence $\mu_i$ of length $k_i$ observed at $z_i$ for every $j$ ($j = 1, 2, \ldots, p$).

Condition 2: Let $S_i \succeq S_j$ if and only if $(\mu_{i1}, \ldots, \mu_{ip}) = (\mu_{j1}, \ldots, \mu_{jp})$.

Define the partition $\pi_i (1 \leq l \leq p)$ corresponding to the relation $\succeq$ defined as $S_i \succeq S_j$ if and only if $\mu_{il} = \mu_{jl}$. Then, from this and (1), it is clear that $M$ is $k_i$-output-observable with respect to $\pi_i (1 \leq l \leq p)$. Therefore, we have only to show that $\pi = \pi_1 \times \pi_2 \times \cdots \times \pi_p$. From (2) we have that $S_i \succeq S_j$ if and only if $(\mu_{i1}, \ldots, \mu_{ip}) = (\mu_{j1}, \ldots, \mu_{jp})$, and this equation holds if and only if $\mu_{il} = \mu_{jl}$ for all $l (1 \leq l \leq p)$, i.e., $S_i \succeq S_j$ for all $l$. Hence we have that $S_i \succeq S_j$ if and only if $S_i \succeq S_j$ for all $l$. Therefore, $\pi = \pi_1 \times \pi_2 \times \cdots \times \pi_p$.

The converse can be proved similarly by Definition 2. Q.E.D.

Theorem 2: The necessary and sufficient condition for a sequential machine $M$ to be modified by adding a binary output function $z$ so that it will be $k$-output-observable with respect to the output function $z$ and a partition $\pi$ is that $M$ is $k$-semi-FSR realizable with respect to $\pi$ where $k$ is a positive integer.

Sufficiency: Suppose that $M$ is $k$-semi-FSR realizable with respect to $\pi$. Let $Y_1, Y_2, \ldots, Y_i$ be its state assignment variables, and let $Y_i(t)$ be a value of $Y_i$ at time $t$. Then from Definition 1, the following conditions are satisfied.

Condition 1: $Y_i(t) = Y_{i-1}(t - 1)$ for $2 \leq i \leq k$.

Condition 2: $S_i \succeq S_j$ if and only if $(y_{i1}, \ldots, y_{ip}) = (y_{j1}, \ldots, y_{jp})$, where $(y_{i1}, \ldots, y_{ip})$ denotes a binary code of the state assignment corresponding to state $S_i$.

Define a binary output function $z$ such that $z(S_i) = y_{i1}$ for each $S_i \in S$. Every length-$k$ output sequence $Z_1Z_2 \cdots Z_{k+1}$ observed at the output $z$ starting at time $t$ is $Y_k(t)Y_{k+1}(t + 1) \cdots Y_{k+1}(t + k - 1)$. Form (1) this sequence equals $Z_k(t)Y_{k-1}(t) \cdots Y_1(t)$, which is a binary code corresponding to state $S_t$ at time $t$. Hence, each length-$k$ output sequence $\mu_i$ observed at the output $z$ is uniquely determined only by the initial state $S_i$ and $\mu_i = y_{i1} \cdots y_{ip}$. From (2) we have that $S_i \succeq S_j$ if and only if $\mu_i = \mu_j$. Therefore, $M$ is $k$-output-observable with respect to $z$ and $\pi$.

Necessity: Suppose that $M$ has been modified by adding a binary output function $z$ so that it is $k$-output-observable with respect to the output function $z$ and $\pi$. Then from Definition 2, the following conditions are satisfied.

Condition 1: The knowledge of the present state is sufficient to uniquely determine the succeeding output sequence of length $k$ observed at the output function $z$.

Condition 2: Let $\mu_i$ be this output sequence when the initial state is $S_i$. Then $S_i \succeq S_j$ if and only if $\mu_i = \mu_j$.

When $\mu_i = Z_1Z_2 \cdots Z_k$, let a state assignment be $(y_{i1}, \ldots, y_{ip}) = (Z_{i1}, \ldots, Z_{ik})$ for state $S_i$. If $\delta(S_i, I_q) = S_i$ for some input $I_q$, then $\mu_i = Z_1Z_2 \cdots Z_kZ_{i+1}$ where $Z_{i+1}$ is uniquely determined by $S_i$ and $I_q$ from (1). Hence, we can define a feedback function $f$ such that $f(S_i, I_q) = Z_{i+1}$. Since $\mu_i = Z_1Z_2 \cdots Z_k$ and $\mu_j = Z_1Z_2 \cdots Z_kZ_{j+1}$ for $\delta(S_j, I_q) = S_j$, we have $Y_1(t) = Y_{i-1}(t - 1)$ for $2 \leq l \leq k$. From (2) we have $S_i \succeq S_j$ if and only if $\mu_i = \mu_j$, and thus $(y_{i1}, \ldots, y_{ip}) = (y_{j1}, \ldots, y_{ip})$. Therefore $M$ is $k$-semi-FSR realizable with respect to $\pi$. Q.E.D.

Theorem 3: Let $M$ be a sequential machine. Then the following four conditions are equivalent.

Condition 1: There exist $p$ binary output functions $z_1, \ldots, z_p$ such that $M$ is $k_1, k_2, \ldots, k_p$-output-observable with respect to the output function $z_1 \times z_2 \times \cdots \times z_p$ and a partition $\pi$.

Condition 2: There exist $p$ binary output functions $z_1, z_2, \ldots, z_p$ and $p$ partitions $\pi_1, \pi_2, \ldots, \pi_p$ such that $M$ is $k_1$-output-observable with respect to $z_1$ and $\pi$, for $1 \leq i \leq p$ and $\pi_1 \pi_2 \cdots \pi_p = \pi$.

Condition 3: There exist $p$ partitions $\pi_1, \pi_2, \ldots, \pi_p$ such that $M$ is $k_1$-semi-FSR realizable with respect to $\pi_i$ for $1 \leq i \leq p$ and $\pi_1 \pi_2 \cdots \pi_p = \pi$.

Condition 4: $M$ is $k_1, k_2, \ldots, k_p$-semi-FSR realizable with respect to $\pi$.
Proof: From Lemma 3, Conditions 1 and 2 are equivalent. From Theorem 2, Conditions 2 and 3 are equivalent. And it follows immediately from Lemma 2 that Condition 3 is equivalent to Condition 4. Q.E.D.

III. OUTPUT-OBSERVABLE SEQUENTIAL MACHINES

In this section we show an algorithm for modifying a given machine to an output-observable one by adding a minimum number of extra outputs. For a given sequential machine $M$ with a binary output function $z$, we can find a minimum partition $\pi$ and a minimum integer $k$ such that the machine $M$ is $k$-output-observable with respect to $z$ and $\pi$. This method is shown in the following.

Procedure $A$

Step 1: Set $\pi(0) = I$ and $l = 1$.

Step 2: For every state $S_i$, test whether all the output sequences of length $l$ observed at the output function $z$ with the machine $M$ initially in state $S_i$ are the same. If “no” for some state $S_i$, set $\pi = \pi(l - 1)$ and $k = l - 1$, and stop. If “yes” for all states, then define a relation $\delta_0$ such that $S_i \delta_0 T_i$, if and only if $\mu_i(l) = \mu_j(l)$ where $\mu_i(l)$ is the output sequence of length $l$ corresponding to state $S_i$.

Step 3: If $\pi(l) = 0$, then set $\pi = 0$ and $k = l$, and stop. If $\pi(l) = \pi(l - 1)$, then set $\pi = \pi(l)$ and $k = l - 1$, and stop. Otherwise, set $l = l + 1$ and go to Step 2.

Step 2 is a process to test if $M$ is $l$-output-observable with respect to the output function $z$ and a partition $\pi(l)$. If $M$ is known not to be $l$-output-observable with respect to $z$ and $\pi(l)$ in Step 2, then the minimum partition is $\pi(l - 1)$. This process is continued until $\pi(l)$ becomes either $\pi(l - 1)$ or the zero position.

It is clear that $\pi(l) \leq \pi(l - 1)$ for each $l$ in Step 3. Therefore, Procedure $A$ terminates in a finite amount of time, i.e., is an algorithm.

To prove that the partition obtained by means of Procedure $A$ is minimum, it will be sufficient to show that $\pi(l) = \pi(l - 1)$ implies $\pi(l + 1) = \pi(l)$ for each $l$.

Assume that $\pi(l) = \pi(l - 1)$ for some $l$ in Step 3. Then $M$ is $m$-output-observable with respect to $z$ and $\pi(m)$ for all $m(1 \leq m \leq l)$. Since $M$ is $l$-output-observable with respect to $z$ and $\pi(l)$, we have that $S_i \delta_0 T_i$ implies $\mu_i(l) = \mu_j(l)$. Let $\mu_i(l) = Z_iZ_2 \cdots Z_l$. Then all the output sequences of length $l - 1$ corresponding to states $S_i, I_i$, and $\delta(S_i, I_i)$ for all inputs $I_i$ and $I_j$, are the same and can be denoted by $Z_2Z_3 \cdots Z_l$. Therefore, $\delta(S_i, I_i, I_j) = \delta(S_i, I_j)$ for all inputs $I_i$ and $I_j$. Since $\pi(l) = \pi(l - 1)$, $\delta(S_i, I_i, I_j) = \delta(S_i, I_i)$ for all $I_i$ and $I_j$, and thus all the output sequences of length $l - 1$ corresponding to states $S_i, I_i$, and $\delta(S_i, I_j)$ are the same and can be denoted by $Z_2Z_3 \cdots Z_l$. Therefore, all the output sequences of length $l + 1$ corresponding to state $S_i$ and $S_j$ are the same, i.e., $\mu_i(l + 1) = \mu_j(l + 1) = Z_2Z_3 \cdots Z_l$. This implies $S_i \delta_0 T_j$ if and only if $\mu_i(l + 1) = \mu_j(l + 1)$. Moreover, it is obvious that $\pi(l + 1) \leq \pi(l)$ for each $l$ in Procedure $A$. Therefore, $\pi(l) = \pi(l + 1)$.

Suppose that, for a given sequential machine $M$, $\pi_i$ and $k_1(1 \leq i \leq p)$ have been obtained by means of Procedure $A$; then $M$ is $k$-output-observable with respect to the output function $z_i$ and the partition $\pi_i$, for each $i(1 \leq i \leq p)$. If $\pi_1 \pi_2 \cdots \pi_{p} = 0$, then $M$ is output-observable. If $\pi_1 \pi_2 \cdots \pi_{p} > 0$, then we have the following theorem.

Theorem 4: The necessary and sufficient condition for a sequential machine $M$ (see Fig. 3) to be modified by adding $s$ binary functions $w_1, w_2, \cdots, w_s$ so that it will be $k_1, k_1, \cdots, k_p, l_1, \cdots, l_s$-output-observable with respect to the output function $z_1 \times z_2 \times \cdots \times z_p \times w_1 \times w_2 \times \cdots \times w_s$ is that there exist $s$ partitions $\pi_1, \pi_2, \cdots, \pi_{s}$ such that $M$ is $l_i$-semi-FSR realizable with respect to $\pi_i$ for each $i(1 \leq i \leq s)$ and $\pi_1 \pi_2 \cdots \pi_{s} > 0$, where each $k_i$ is a nonnegative integer and each $l_i$ is a positive integer.

Proof: This can be proved readily from Theorem 3 and the fact that $M$ is $k_1$-output-observable with respect to $z_1$ and $\pi_i$ for each $i(1 \leq i \leq p)$. Q.E.D.

From Lemma 1 and Theorem 4 we have the following corollary.

Corollary 1: The necessary and sufficient condition for a sequential machine $M$ to be modified by adding $s$ binary output functions $w_1, w_2, \cdots, w_s$ so that it will be $k_1, k_1, \cdots, k_p, l_1, \cdots, l_s$-output-observable with respect to the output function $z_1 \times z_2 \times \cdots \times z_p \times w_1 \times w_2 \times \cdots \times w_s$ is that there exist $s$ SRP’s $\tau_1, \tau_2, \cdots, \tau_s$ of length $l_1, l_2, \cdots, l_s$, respectively, such that $\pi_1 \pi_2 \cdots \pi_{s} \tau_1 \tau_2 \cdots \tau_{s} = 0$.

Corollary 1 shows that if we can find the least possible number of SRP’s $\tau_1, \tau_2, \cdots, \tau_s$ such that

$$\pi_1 \pi_2 \cdots \pi_{s} \tau_1 \tau_2 \cdots \tau_{s} = 0,$$

then we can modify the machine $M$ to an output-observable one by adding a minimum number of extra outputs. The problem of generating all the SRP’s for a given machine has been investigated by Nichols [16].

Suppose that we have obtained the least number of SRP’s $\tau_1, \tau_2, \cdots, \tau_s$ satisfying the condition of Corollary 1. Then we can construct binary output functions $w_j(1 \leq j \leq s)$ satisfying the condition of Corollary 1 as follows. Let $Y_{j1} Y_{j2} \cdots Y_{ji} \cdots$ be the state assignment variables of the $l_i$-stage shift register corresponding to SRP $\tau_i$ (see Fig. 4), and let $(y_{j1} Y_{j2} \cdots Y_{ji} \cdots)$ be a binary code corresponding to state $S_i$. Note that each state is given a single coding. Define a binary output function $w_j$ such that $w_j(S_i) = y_{ji}$ for $S_i \in S$, in the same way as shown in the proof of Theorem 2.

Summarizing this argument, we can present the following procedure for modifying a given machine so that it will be output-observable by adding a minimum number of extra outputs.

Procedure $B$—Modification Algorithm

Step 1: Given a sequential machine $M$ having binary output functions $z_1, z_2, \cdots, z_p$, find a minimum partition $\pi_i$ and $k_i$ for each $z_i (1 \leq i \leq p)$ by means of Procedure $A$. 
IV. FAULT DETECTION FOR OUTPUT-OBSERVABLE SEQUENTIAL MACHINES

In this section we consider fault detection experiments for \( k_1, k_2, \ldots, k_p \)-output-observable sequential machines. Let \( M \) be the fault-free machine with the output function \( z_1 \times z_2 \times \cdots \times z_p \) and let \( M' \) be the tested (possibly faulty) machine with the output function \( z'_1 \times z'_2 \times \cdots \times z'_p \). Assume that the class of allowable failures satisfies the following conditions.

**Condition 1**: Any failure which occurs is assumed to occur throughout the test.

**Condition 2**: A failure which increases the number of states in the machine does not occur.

**Condition 3**: A faulty machine \( M' \) is still \( k_1, k_2, \ldots, k_p \)-output-observable with respect to \( z'_1 \times z'_2 \times \cdots \times z'_p \) and some partition \( \pi \), i.e., the knowledge of the present state of \( M' \) is sufficient to uniquely determine the succeeding output sequence of length \( k \) observed at the output function \( z_i \) for all \( i(1 \leq i \leq p) \).

In Section II we have shown that \( k_1, k_2, \ldots, k_p \)-output-observable sequential machines can be realized as binary FSR's of the form shown in Fig. 2. For sequential machines with shift registers, let us consider a fault that results when one of the stages of any FSR is either stuck-at-1 or stuck-at-0. The output, then, at the last stage of the faulty FSR will be a sequence of identical values. Therefore, the present state of the faulty FSR is sufficient to uniquely determine the succeeding output sequence (a sequence of identical values) of length \( k \) where \( k \) is the length of the fault-free FSR. Hence, this fault satisfies the above fault assumption. Any stuck-at fault in the combinational circuit is also included by the above fault condition.

Under these assumptions, let us design a checking sequence. Given a \( k_1, k_2, \ldots, k_p \)-output-observable sequential machine \( M \), let \( w_1 \) be an input–output sequence that passes through all the transitions of the state table of \( M \), and let \( w_2 \) be an arbitrary input–output sequence of length \( k \) where \( k = \max \{ k_1, \ldots, k_p \} \). It will be proved in the following theorem that the input–output sequence \( w_2 \), called C-sequence, is a checking sequence.

**Theorem 5**: Let \( M \) be an output-observable sequential machine. Then the sequential machine satisfying\(^2\) the C-sequence of \( M \) is isomorphic to \( M \).

**Proof**: Let \( M' = (S', I, Z', \delta', \lambda') \) be a sequential machine satisfying the C-sequence of \( M \). From the failure assumption, \( M' \) is \( k_1, \ldots, k_p \)-output-observable with respect to \( z'_1 \times z'_2 \times \cdots \times z'_p \) and some partition \( \pi \). Let \( S \) and

---

\(^2\) We say that a sequential machine satisfies an input–output sequence if, applying the input sequence, the output sequence is obtained.
$S_i'$ be the states of $M$ and $M'$, respectively, at time $t$ in the C-sequence. Define a mapping $f: S' \rightarrow S$ such that $f(S_i') = S_i$ for each time $t$. We first show that this yields a well-defined mapping.

Now suppose that $S_{t_1} \neq S_{t_2}$ at time $t_1$ and $t_2$. This implies that

$$z_i(t_i)z_i(t_i + 1) \cdots z_i(t_i + k_i - 1)$$

for some $i$, since $M$ is $k_1k_2 \cdots k_p$-output-observable with respect to $z_1 \times z_2 \times \cdots \times z_p$ and the zero partition. Since $M'$ satisfies the C-sequence of $M$,

$$z_i'(t_i)z_i'(t_i + 1) \cdots z_i'(t_i + k_i - 1) = z_i(t_i)z_i(t_i + 1) \cdots z_i(t_i + k_i - 1) \quad \text{for} \quad j = 1, 2.$$

Therefore,

$$z_i'(t_i)z_i'(t_i + 1) \cdots z_i'(t_i + k_i - 1)$$

This implies $S_i' \neq S_i$, since $M'$ is $k_1k_2 \cdots k_p$-output-observable with respect to $z_1' \times z_2' \times \cdots \times z_p'$ and $\pi$. This implies $S_{t_1}' \neq S_{t_2}'$. Hence, $S_{t_1} \neq S_{t_2}$ implies $S_{t_1}' \neq S_{t_2}'$.

This shows that $f$ is well-defined.

Since the C-sequence passes through all the states of $S$, the mapping $f$ is a surjection. Moreover, from the failure assumption (2), we have $|S'| \leq |S|$ where $|S|$ means the number of states in $S$. Thus, $f$ is a bijection.

Let $I_i$ and $Z_i$ be the input and output symbols, respectively, at time $t$ in the C-sequence. From the definition of $f$, we have $f(S_i'(I_i)) = f(S_i(I_i)) = S_i$. Hence, $S_i, S_i'$ are related by $S_i = \delta(S_i(I_i)) = \delta(f(S_i'(I_i)))$ for any time $t$.

This holds for all states and all input symbols of $M$, since the C-sequence passes through all the transitions of the state table of $M$. Hence, $f$ is an isomorphism of $M'$ onto $M$.

Q.E.D.

Theorem 5 implies that only the correctly operating machine satisfies the C-sequence of $M$. However, the converse is not always true, i.e., the correctly operating machine does not always satisfy the C-sequence when the machine under test is not initially in the starting state of the C-sequence of $M$. So the machine under test should be initially in the starting state of the C-sequence when the C-sequence is to be applied. This can be done by applying a homing sequence.$^3$ For $k_1k_2 \cdots k_p$-output-observable sequential machines, any input sequence of length $k$ ($k = \max \{k_1, k_2, \cdots, k_p\}$) is a homing sequence. The entire checking experiment can be summarized as follows.

Step 1: By applying an arbitrary input sequence $X_1$ of length $k$, determine the final state $S_0$.

Step 2: Construct an input sequence $X_2$ which passes through all the transitions of $M$ initially in state $S_0$.

Step 3: Apply the input sequence $X_2$ followed by an arbitrary input sequence $X_3$ of length $k$ (the C-sequence of $M$), and observe the response. The machine under test is correct if it responds correctly to the input sequence $X_2X_3$. Otherwise, the machine is faulty.

Example 2: Consider machine $M_2$, given by Table III, which is 1,2-output-observable with respect to output function $z_1 \times z_2$ and the zero partition. By applying an arbitrary input sequence of length $k$ ($k = \max \{1, 2\} = 2$) and observing the output sequence of length 1 and 2 at the output terminals $z_1$ and $z_2$, respectively, we can establish the initial state and the final state. Suppose that the machine is in the state 1, then the shortest input–output sequence $\omega_1$ that passes through all the transitions of $M_2$ is obtained as follows:

<table>
<thead>
<tr>
<th>Input</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Output</td>
<td>$z_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$z_2$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

As the final state is 5, the following sequence is an input–output sequence $\omega_2$ of length 2 starting at state 5:

<table>
<thead>
<tr>
<th>Input</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Output</td>
<td>$z_2$</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$z_2$</td>
<td>1</td>
</tr>
</tbody>
</table>

Then a checking sequence for $M_2$ is the following:

<table>
<thead>
<tr>
<th>Input</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Output</td>
<td>$z_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$z_2$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The problem of obtaining the shortest input sequence $X_2$ in Step 2 can be reduced to the traveling salesman problem. Consider a directed graph consisting of a finite set $V$ of vertices together with a collection $U$ of ordered pairs of vertices, called arcs, in which associated with each arc $(v_i, v_j)$ is a number $d_{ij} \geq 0$ (which we shall call the distance between $v_i$ and $v_j$). Any sequence of vertices, in which every vertex of the graph appears at least once and the first and last vertices are identical, is called a tour. A tour may be written as $t = (v_1, v_2, \ldots, v_p, v_1)$. The length of the tour, denoted by $L(t)$, is the sum of the arc lengths over the arcs included in the tour, i.e.,

$$L(t) = \sum_{(v_i, v_j) \in t} d_{ij}.$$

The problem of finding the shortest tour is the well-known traveling salesman problem [15].

As the application of the traveling salesman problem, we have the following procedure for finding the shortest input sequence which passes through all the transitions in the given state diagram or the state graph.
Step 1: Construct an interchange graph $G$ of the given state graph, and set $d_{ij} = 1$ for all arcs $(v_i,v_j)$ in the graph $G$.

Step 2: Find the shortest tour in the graph $G$ by the method for solving the traveling salesman problem, and construct the input sequence starting from state $S_0$, which corresponds to the tour.

Although the comparison of our method with the previous methods [1]–[8] is difficult because of the different fault assumptions, we will show some advantages of our method. Since a checking experiment must check all the transitions of $M$, it must pass through at least all the transitions. This follows from the fact that no checking sequence can be shorter than the shortest input sequence $X_2$ which passes through all the transitions. Therefore $|X_2| \leq L_0$ where $|X_2|$ is the length of the input sequence $X_2$ and $L_0$ is the length of the minimum checking experiment. Furthermore, $|X_1| = |X_2| = k < n$ where $n$ is the number of states of $M$. Consequently, $|X_1X_2X_3| = 2k + |X_1| < 2n + L_0$, i.e., the length of the checking experiments for the $n$-state output-observable sequential machines is at most $2n + L_0$. These are nearly minimum checking experiments, and hence are much shorter than those described in previous work.

Since the checking experiments presented here have only to pass through all the transitions in order to check all the transitions, the procedure is much simpler than the previous method [1]–[8]. However, when one tries to obtain the shortest input sequence which passes through all the transitions, one must apply such a method as the traveling salesman problem, so the amount of computation may become huge, in general.

V. CONCLUSION

In this paper we have introduced output-observable sequential machines as the easily testable sequential machines, and have described a procedure for the modification of a given sequential machine to an output-observable one by adding a minimum number of extra outputs. This procedure is mainly based upon the fact that the output observability of a sequential machine is equivalent to the semi-FSR realizability of it. We have also presented a procedure for the organization of simple, short, and efficient checking experiments for output-observable machines. For such machines, the checking experiments have only to pass through all the transitions of the given state table. In this sense, the output-observable machines have the advantage of being easy to test.

ACKNOWLEDGMENT

The authors wish to acknowledge the support and encouragement of Prof. H. Ozaki of Osaka University, Osaka, Japan.

---

Hideo Fujinura (S’70) was born in Nara, Japan, on February 9, 1946. He received the B.E. and M.E. degrees in electronic engineering from Osaka University, Osaka, Japan, in 1969 and 1971, respectively. He is currently a Ph.D. student in the Department of Electronic Engineering, Osaka University. His research interests are switching theory and automata theory, and he specializes in the development of fault diagnosis of digital circuits.

Mr. Fujinura is a member of the Institute of Electronics and Communication Engineers of Japan and the Information Processing Society of Japan.

Kozo Kinoshita (S’58–M’64) was born in Osaka, Japan, on June 21, 1936. He received the B.E., M.E., and Ph.D. degrees in communication engineering from Osaka University, Osaka, Japan, in 1959, 1961, and 1964, respectively.

Since 1964 he has been with Osaka University, where he is now an Associate Professor of Electronics Engineering. His fields of interest are switching theory, automata theory, and fault diagnosis of information processing systems.

Dr. Kinoshita is a member of the Institute of Electronics and Communication Engineers of Japan and the Information Processing Society of Japan.