Fail-Safe Realization of Sequential Machines

TADAŌ TAKAOKA*

Electrical Communication Laboratories, Nippon Telegraph and Telephone Public Corporation, Musashino, Tokyo, Japan

AND

TOSHIHIDE IBARAKI

Department of Applied Mathematics and Physics, Faculty of Engineering, Kyoto University, Kyoto, Japan

It is known that a sequential machine is realized by a binary sequential circuit (BSC). In this paper we try to construct a fail-safe BSC realizing a given sequential machine with the additional constraint that its output either takes the correct value or fails from 0 to 1 if the outputs of logic components in BSC, such as AND, OR, NOT gates and DELAY elements, fail asymmetrically from 0 to 1. The word "fail-safe" means that the failure 0 → 1 is considered safe while the reverse is not. It is rather easy to construct a fail-safe BSC for a given sequential machine. The main purpose of this paper is to construct a fail-safe BSC with the smallest number of DELAY's.

First we construct a positive BSC with the smallest number of DELAY's. A BSC is positive if its logic components are all positive. Positive BSC's are fail-safe but the converse is not true. Then we also construct a fail-safe BSC with the smallest number of DELAY's. We see that the procedure for the latter construction is more complicated than the former.

I. INTRODUCTION

Fail-safe systems for switching functions were first presented by Watanabe and Takahashi (1965), and have been studied by Mine and Koga (1967), Hashimoto, Tokura and Kasami (1967), Hirayama, Watanabe and Urano (1969) and others. In a fail-safe system for a switching function, if the inputs or the logic components fail asymmetrically from 0 to 1, the induced failure

* Part of this work is included in the Ph.D. dissertation of T. Takaoka.
on the output is also asymmetric from 0 to 1, where the value 1 is considered to be safe even if it is erroneous.

A fail-safe binary sequential circuit (fail-safe BSC) realizing a given machine was presented by Tokura, Kasami and Ozaki (1966) whose behavior is analogous to that of fail-safe systems for switching functions except that no failure on the input is assumed in this case. Fail-safe BSC's were also studied in Watanabe, Takahashi and Enomoto (1966) and Tohma (1970). In Tokura et al. (1966), for constructing a fail-safe BSC realizing a given machine, the constant weight code, especially the half weight code, is used for the state assignment. This assignment is sometimes redundant in the sense that the number of bits (i.e., the number of DELAY's) used for the state assignment can be further reduced.

In the present paper we attempt to construct a fail-safe BSC realizing a given machine with the smallest number of bits. It is shown that if all the switching functions used in the BSC are positive with respect to the variables representing internal states, the BSC is fail-safe. This BSC is called positive. We give an algorithm for constructing a positive BSC realizing a given machine with the smallest number of bits. This is done by making use of a certain partial order with the substitution property defined on the given machine.

In general, however, the number of bits required in a fail-safe BSC may be able to be reduced further if we can dispense with the positiveness assumption imposed on the fail-safe BSC. In this case the procedure to obtain a fail-safe BSC with the smallest number of bits becomes somewhat complicated and may be difficult to use in practice, though a method for that is presented.

II. SEQUENTIAL MACHINES AND BINARY RELATIONS
WITH THE SUBSTITUTION PROPERTY

In this section we discuss basic properties of binary relations with the substitution property, as a preparation for the subsequent discussion. It is an extension of the well-known concept of partitions with the substitution property studied by Hartmanis and Stearns (1966). This extension is necessary because we will deal with partial orders, rather than partitions, with the substitution property.

**Definition 1.** A **sequential machine**, abbreviated by a **machine**, is a quintuple $S = <S, M, \beta, \Sigma, \Delta>$ where
FAIL-SAFE SEQUENTIAL MACHINES

S: a finite set of internal states
Σ: a finite set of input alphabet
Δ: a finite set of output alphabet
M: a mapping $S \times \Sigma \rightarrow S$ (next state function)
$\beta$: a mapping $S \rightarrow \Delta$ (output function).

This machine is called Moore type (see for example Harrison, 1965). $\Sigma^*$ denotes the set of all the words (or strings) generated from the input alphabet $\Sigma$ including the null string $\lambda$. The mapping $M$ is extended from $S \times \Sigma \rightarrow S$ to $S \times \Sigma^* \rightarrow S$ recursively by

$$M(s, \lambda) = s$$
$$M(s, ax) = M(M(s, a), x),$$

where $s \in S$, $a \in \Sigma$ and $x \in \Sigma^*$. Then we have

$$\forall s \in S, \forall x, \forall y \in \Sigma^* \ M(s, xy) = M(M(s, x), y).$$

Let $\lg(x)$ denote the length of the word $x$.

**Definition 2.** Let $A$ be a binary relation on $S$, that is, $A \subseteq S \times S$. $A$ has the substitution property (SP) if

$$\forall (s_1, s_2) \in S \times S \ ((s_1, s_2) \in A \Rightarrow \forall x \in \Sigma^*(M(s_1, x), M(s_2, x)) \in A).$$

Note that this definition is equivalent to

$$\forall (s_1, s_2) \in S \times S \ ((s_1, s_2) \in A \Rightarrow \forall \sigma \in \Sigma(M(s_1, \sigma), M(s_2, \sigma)) \in A).$$

**Definition 3.** Let $A \subseteq S \times S$. The core with SP (CSP) of $A$, written $\bar{A}$, is defined by

$$(s_1, s_2) \in \bar{A} \Leftrightarrow \forall x \in \Sigma^*(M(s_1, x), M(s_2, x)) \in A.$$

**Lemma 1.** The CSP $\bar{A}$ of $A$ has SP and $\bar{A} \subseteq A$.

**Proof.**

$$(s_1, s_2) \in \bar{A} \Leftrightarrow \forall x \in \Sigma^*(M(s_1, x), M(s_2, x)) \in A$$

$$\Rightarrow \forall x, \forall z \in \Sigma^*(M(s_1, xz), M(s_2, xz)) \in A$$

$$\Rightarrow \forall x \in \Sigma^*(\forall z \in \Sigma^*(M(s_1, x), x), M(M(s_2, x), z)) \in A$$

$$\Rightarrow \forall x \in \Sigma^*(M(s_1, x), M(s_2, x)) \in \bar{A}. \quad \text{Q.E.D.}$$
DEFINITION 4. Let $A \subseteq S \times S$. The sequence of sets, $\{A_i\}$, is defined by

$$(s_1, s_2) \in A_i \iff (\forall x \in \Sigma^*(\lg(x) \leq i \Rightarrow (M(s_1, x), M(s_2, x)) \in A))$$

Note that $A_0 = A$.

LEMMA 2. We have the following results:

(i) $\forall i \geq 0 A_i \supseteq A_{i+1}$ (nonincreasing property).

(ii) $\exists i \geq 0, \forall h \geq 0 (A_i = A_{i+h})$ (convergence).

(iii) $\forall i \geq 0, \forall h \geq 0 (A_i = A_{i+h})$.

(iv) $\forall i \geq 0 (s_1, s_2) \in A_{i+1} \iff (s_1, s_2) \in A_i \land \forall \sigma \in \Sigma (M(s_1, \sigma), M(s_2, \sigma)) \in A_i$ (recursive formula).

(v) $A_i = A_{i+1} \Rightarrow i \leq 2n - 2$.

Proof.

(i) is obvious.

(ii) follows because $A$ is finite and the sequence $\{A_i\}$ is nonincreasing.

(iii) Let $A_i = A_{i+1}$. Then from (iv) we have

$$(s_1, s_2) \in A_{i+1} \iff ((s_1, s_2) \in A_i \land \forall \sigma \in \Sigma (M(s_1, \sigma), M(s_2, \sigma)) \in A_i)$$

$\iff ((s_1, s_2) \in A_{i+1} \land \forall \sigma \in \Sigma (M(s_1, \sigma), M(s_2, \sigma)) \in A_{i+1})$

$\iff (s_1, s_2) \in A_{i+2}$.

Hence we have that $A_{i+1} = A_{i+2} = A_{i+3} = \cdots$.

(iv)

$$(s_1, s_2) \in A_{i+1} \iff (\forall x \in \Sigma^*(\lg(x) \leq i + 1 \Rightarrow (M(s_1, x), M(s_2, x)) \in A))$$

$\iff (\forall x \in \Sigma^*(\lg(x) \leq i \Rightarrow (M(s_1, x), M(s_2, x)) \in A))$

$\land (\forall \sigma \in \Sigma, \forall x \in \Sigma^*(\lg(x) \leq i) \Rightarrow (M(M(s_1, \sigma), x), M(M(s_2, \sigma), x)) \in A))$

$\iff ((s_1, s_2) \in A_i \land \forall \sigma \in \Sigma (M(s_1, \sigma), M(s_2, \sigma)) \in A_i)$.

(v) follows from the fact that if $(M(s_1, x), M(s_2, x)) = (s_1', s_2')$, the pair $(s_1', s_2')$ can be reached by some word whose length is less than or equal to $2n - 2$ (e.g., Harrison (1965)).

Q.E.D.

LEMMA 3. $A_\infty = \bar{A}$ for any $A \subseteq S \times S$. 
From these facts we have the following algorithm for obtaining $\bar{A}$ from a given $A \subseteq S \times S$.

**Algorithm 1.**

1. Let $A_0 = A$.
2. Compute $A_{i+1}$ from $A_i$ by the following recursive formula.
   $$(s_1, s_2) \in A_{i+1} \iff ((s_1, s_2) \in A_i \land \forall \sigma \in \Sigma(M(s_1, \sigma), M(s_2, \sigma)) \in A_i).$$
3. If $A_{i+1} = A_i$, go to (v). Otherwise go to (iv).
4. Increase $i$ by one and go to (ii).
5. Let $\bar{A} = A_i$ and halt.

Note that $\bar{A} = A_i$ for the smallest integer $i$ such that $A_i = A_{i+1}$ and $\bar{A}$ is uniquely computed from $A$.

**Lemma 4.** For any $A, B \subseteq S \times S$

$$A \subseteq B \Rightarrow \bar{A} \subseteq \bar{B}.$$

**Proof.** Assume that $A \subseteq B$. Then

$$(s_1, s_2) \in \bar{A} \iff \forall x \in \Sigma^* (M(s_1, x), M(s_2, x)) \in A$$

$$\Rightarrow \forall x \in \Sigma^* (M(s_1, x), M(s_2, x)) \in B$$

$$\iff (s_1, s_2) \in \bar{B}.$$  

Q.E.D.

**Lemma 5.** $\bar{A}$ is the maximal set with SP included in $A$.

**Proof.** Suppose that $B \subseteq S \times S$ has SP. Clearly $\bar{B} = B$. From Lemma 4 we have

$$B \subseteq A \Rightarrow B \subseteq \bar{A} \Rightarrow B \subseteq A.$$

Thus $\bar{A}$ is the maximal set with SP included in $A$.

Q.E.D.

**Definition 5.** For a given machine $S$, $SP(S)$ is the set of all sets $A$ such that $E \subseteq A \subseteq \Omega$ and $A$ has SP, where $\Omega = S \times S$ and $E$ is the equality relation on $S$, that is,

$$E = \{(s, s) \mid s \in S\}.$$
We introduce an order into $SP(S)$ by set inclusion. For $A, B \in SP(S)$, $A \cap B$ is the greatest lower bound and $A \cup B$ is the smallest upper bound.

**Theorem 1.** The algebraic system $SP(S) = \langle SP(S), \subseteq, \cap, \cup \rangle$ forms a lattice.

**Proof.** $SP(S)$ has the unique maximal element $\Omega$ and the unique minimal element $E$. For $A, B \in SP(S)$, let us prove that $A \cap B = A \cap B$. Obviously $A \cap B \subseteq A \cap B$ and

$$(s_1, s_2) \in A \cap B \iff (s_1, s_2) \in A \land (s_1, s_2) \in B$$

$$\Rightarrow \forall x \in \Sigma^*(M(s_1, x), M(s_2, x)) \in A$$

$$\land \forall x \in \Sigma^*(M(s_1, x), M(s_2, x)) \in B$$

$$\Rightarrow \forall x \in \Sigma^*(M(s_1, x), M(s_2, x)) \in A \land B$$

shows that $A \cap B$ has $SP$. Therefore $A \cap B = A \cap B$. Next for distinct $A, B \in SP(S)$ assume that there exist two distinct minimal upper bounds $C$ and $D$ of $A$ and $B$. Then we have that $A \cup B \subseteq C \cap D \subseteq C, D$ which is a contradiction because $C \cap D \in SP(S)$, and $C$ and $D$ are minimal upper bounds for $A$ and $B$.

Q.E.D.

### III. Fail-Safe and Positive Orders on a Sequential Machine

To obtain a fail-safe realization of a given machine, it is useful to distinguish state errors $s_i \rightarrow s_j$ of the machine according to whether they cause failures $1 \rightarrow 0$ on the output (i.e., not fail-safe) or not (i.e., fail-safe). It will be shown that the set of all the fail-safe state errors (denoted $P$) is a partial order. Then two kinds of partial orders, called positive order and $FS$ order, included in $P$ receive special attention since they play important roles in the state assignment for the fail-safe realization.

In this section we investigate properties of these two kinds of partial orders and give algorithms to obtain them. The actual construction of binary sequential circuits satisfying the fail-safe condition will be discussed in Section IV.

**Definition 6.** A set $R \subseteq T \times T$ is called a **partial order** on the set $T$ if it satisfies the following three conditions

(i) $\forall t \in T \ (t, t) \in R$ (Reflexivity).
(ii) \( \forall (t_1, t_2) \in T \times T, ((t_1, t_2) \in R \land (t_2, t_1) \in R \Rightarrow t_1 = t_2) \) (Anti-symmetry).

(iii) \( \forall (t_1, t_2), \forall (t_2, t_3) \in T \times T, ((t_1, t_2) \in R \land (t_2, t_3) \in R \Rightarrow (t_1, t_3) \in R) \) (Transitivity).

The algebraic system \( T = \langle T, R \rangle \) is called a partially ordered set. If \((t_1, t_2) \in R\), \( t_2 \) is said to be in higher order than \( t_1 \) (as to the order \( R \)). A partially ordered set \( T = \langle T, R \rangle \) is illustrated by a graph in the conventional way such that, if \((t_1, t_2) \in R\), the node of \( t_2 \) is written in a higher position than that of \( t_1 \) and connected by an edge (or a sequence of edges). \( R \subseteq T \times T \) is said to be a pseudo order if it satisfies only two conditions (i) and (iii).

A pseudo ordered set \( T = \langle T, R \rangle \) is illustrated similarly to the case of a partially ordered set with the added rule that if \((t_1, t_2) \in R\) and \((t_2, t_1) \in R\), the nodes of \( t_1 \) and \( t_2 \) are written in the same level.

**Example 1.** Let \( T = \{t_1, t_2, t_3\} \) and \( R = \{(t_1, t_1), (t_2, t_2), (t_3, t_3), (t_1, t_3), (t_2, t_3)\} \). \( R \) is a partial order and the partially ordered set \( T = \langle T, R \rangle \) is illustrated in Fig. 1. Let \( R' = \{R, (t_1, t_2), (t_2, t_1)\} \). \( R' \) is a pseudo order and the pseudo ordered set \( T = \langle T, R' \rangle \) is illustrated in Fig. 2.

Hereafter we assume a partial order \( \preceq \) defined on the set of the output alphabet \( \Delta \) of a given machine \( S = \langle S, \kappa, \beta, \Sigma, \Delta \rangle \). For example, if \( \Delta = \{0, 1\} \), the order \( 0 < 1 \) is usually assumed.
**Definition 7.** A state error \((s_1, s_2)\) is an element of \(S \times S\), which may be interpreted as the error \(s_1 \not\leq s_2\) occurred in machine \(S\). A state error \((s_1, s_2)\) is said to satisfy the *fail-safe condition* \((FS\) condition) if

\[
\forall x \in \Sigma^* \beta(M(s_1, x)) \leq \beta(M(s_2, x)).
\]

A set \(A \subseteq S \times S\) satisfies the \(FS\) condition if

\[
\forall (s_1, s_2) \in S \times S((s_1, s_2) \in A \Rightarrow \forall x \in \Sigma^* \beta(M(s_1, x)) \leq \beta(M(s_2, x))).
\] (1)

In the above definition a failure on the output is regarded safe if it is from the smaller to the greater with respect to the order \(\leq\).

**Definition 8.** For a given machine \(S\), the set \(P \subseteq S \times S\) is defined by

\[
(s_1, s_2) \in P \iff \forall x \in \Sigma^* \beta(M(s_1, x)) \leq \beta(M(s_2, x)).
\]

\(P\) is the maximum error set which satisfies the \(FS\) condition. Obviously \(P\) is unique.

**Definition 9.** For a given machine \(S\), a set \(Q \subseteq S \times S\) is called an \(FS\) order of \(S\) if \(Q\) is a partial order on the set \(S\) and satisfies the \(FS\) condition. In other words, \(Q\) is an \(FS\) order if \(Q\) is a partial order and \(Q \subseteq P\).

**Lemma 6.** If a machine \(S\) is reduced, the set \(P\) is a partial order.

**Proof.** Reflexivity and transitivity are obvious. As for anti-symmetry we see that

\[
(s_1, s_2) \in P \land (s_2, s_1) \in P \Rightarrow \forall x \in \Sigma^* \beta(M(s_1, x)) \leq \beta(M(s_2, x))
\]

\[
\land \forall y \in \Sigma^* \beta(M(s_1, y)) \geq \beta(M(s_2, y))
\]

\[
\Rightarrow \forall x \in \Sigma^* \beta(M(s_1, x)) = \beta(M(s_2, x))
\]

\[
\Rightarrow s_1 = s_2. \quad \text{Q.E.D.}
\]

Note that, if \(S\) is not reduced, \(P\) is in general a pseudo order. Thus we have the next theorem.

**Theorem 2.** \(P\) is the maximal \(FS\) order of \(S\), provided that \(S\) is reduced.

**Lemma 7.** \(P\) has \(SP\).
Proof.

$$(s_1, s_2) \in P \Rightarrow \forall x, \forall z \in \Sigma^* \beta(M(s_1, xz)) \leq \beta(M(s_2, xz))$$

$$\Rightarrow \forall x \in \Sigma^*(\forall z \in \Sigma^* \beta(M(M(s_1, x), z)) \leq \beta(M(M(s_2, x), z)))$$

$$\Rightarrow \forall x \in \Sigma^*(M(s_1, x), M(s_2, x)) \in P.$$  Q.E.D.

Definition 10. For a given machine $S$, a set $Q \subseteq S \times S$ is said to be a positive order of $S$ if $Q$ is an $FS$ order and has $SP$.

Clearly we have the following theorem.

Theorem 3. $P$ is the maximal positive order of $S$, provided that $S$ is reduced.

Example 2. For the reduced machine $S$ given in Table I, the maximal $FS$ (or positive) order $P$ is given by $((s_i, s_i) \in E$ are omitted for simplicity).

$$P = \{(s_1, s_2), (s_1, s_3), (s_2, s_3), (s_1, s_4), (s_1, s_6), (s_2, s_3), (s_2, s_4), (s_3, s_4), (s_4, s_5), (s_5, s_6)\}.$$

Table I

<table>
<thead>
<tr>
<th>State</th>
<th>Input</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>0</td>
<td>$s_1$</td>
<td>$s_2$</td>
</tr>
<tr>
<td>$s_2$</td>
<td>0</td>
<td>$s_1$</td>
<td>$s_3$</td>
</tr>
<tr>
<td>$s_3$</td>
<td>0</td>
<td>$s_4$</td>
<td>$s_5$</td>
</tr>
<tr>
<td>$s_4$</td>
<td>1</td>
<td>$s_1$</td>
<td>$s_6$</td>
</tr>
<tr>
<td>$s_5$</td>
<td>1</td>
<td>$s_1$</td>
<td>$s_6$</td>
</tr>
<tr>
<td>$s_6$</td>
<td>1</td>
<td>$s_6$</td>
<td>$s_6$</td>
</tr>
</tbody>
</table>

$P$ is also illustrated in Fig. 3. As an example, consider state error $(s_1, s_2)$. By applying some $x \in \Sigma^*$, we see that state error $(s_1, s_2)$ induces state errors $(s_2, s_3), (s_1, s_4), (s_2, s_5), (s_3, s_4), (s_4, s_5), (s_5, s_6), (s_1, s_6)$ and $(s_2, s_6)$ (e.g.,
$M(s_1, 1) = s_2$ and $M(s_2, 1) = s_3$ imply that $(s_1, s_2)$ induces $(s_2, s_3)$. However, all induced state errors $(s_i, s_j)$ satisfy $\beta(s_i) \leq \beta(s_j)$ and hence $(s_1, s_2)$ satisfies the FS condition. Similarly it is possible to make sure that all $(s_i, s_j) \in P$ satisfy the FS condition.

Now we present an algorithm to obtain the above $P$ for a given machine $S$.

**Definition 11.** For a given machine $S$, the sequence of sets $\{P_i\}$ is defined by

\[
(s_i, s_j) \in P_i \iff \forall x \in \Sigma^* (\log(x) \leq i \Rightarrow \beta(M(s_i, x)) \leq \beta(M(s_j, x))).
\]

Let $S$ be partitioned into $S_\delta$'s as

\[
S_\delta = \{s \in S \mid \beta(s) = \delta\}, \quad S = \bigcup_{\delta \in \mathbb{D}} S_\delta.
\]

Then we have

\[
P_0 = \bigcup_{\delta \ll \delta'} S_\delta \times S_{\delta'}.
\]

**Lemma 8.** For a given machine $S$, if a set $R \subseteq S \times S$ is a partial order on $S$, then $\overline{R}$ is also a partial order on $S$.

**Proof.**

(i) $(s, s) \in \overline{R}$.

(ii) $(s_1, s_2) \in \overline{R} \land (s_2, s_1) \in \overline{R} \Rightarrow (s_1, s_2) \in R \land (s_2, s_1) \in R \Rightarrow s_1 = s_2$. 

(iii) \((s_1, s_2) \in R \land (s_2, s_3) \in R \Rightarrow \forall x \in \Sigma^*(M(s_1, x), M(s_2, x)) \in R \land \forall y \in \Sigma^*(M(s_2, y), M(s_3, y)) \in R \Rightarrow \forall x \in \Sigma^*(M(s_1, x), M(s_3, x)) \in R \Rightarrow (s_1, s_3) \in R\). Q.E.D.

**Theorem 4.** It holds that \(P = P_\infty = \bar{P}_0\).

**Proof.** Obvious from Definition 8, Definition 11 and Lemma 3. Q.E.D.

Consequently \(P\) can be obtained by the following algorithm.

**Algorithm 2.**

(i) Let \(A = P_0\).

(ii) Go to Algorithm 1.

(iii) Let \(P = A\).

**Example 3.** For the machine \(S\) given in Table I we compute \(P\) (the equality relation \(E\) is omitted from each \(P_i\) for notational simplicity).

\[
P_0 = \{(s_1, s_2), (s_2, s_1), (s_1, s_3), (s_3, s_1), (s_2, s_3), (s_3, s_2), (s_1, s_6), (s_6, s_1), (s_2, s_4), (s_4, s_2), (s_2, s_5), (s_5, s_2), (s_3, s_4), (s_4, s_3), (s_4, s_0), (s_0, s_4), (s_5, s_6), (s_6, s_5), (s_6, s_0), (s_0, s_6)\}.
\]

\[
P_1 = \{(s_1, s_2), (s_2, s_1), (s_1, s_3), (s_3, s_1), (s_1, s_4), (s_4, s_1), (s_2, s_5), (s_5, s_2), (s_2, s_6), (s_6, s_2), (s_3, s_4), (s_4, s_3), (s_4, s_0), (s_0, s_4), (s_5, s_6), (s_6, s_5), (s_6, s_0)\}.
\]

\[
P_2 = \{(s_1, s_2), (s_1, s_3), (s_2, s_3), (s_3, s_1), (s_1, s_4), (s_4, s_1), (s_2, s_5), (s_5, s_2), (s_2, s_6), (s_6, s_2), (s_3, s_4), (s_4, s_3), (s_4, s_0), (s_0, s_4), (s_5, s_6), (s_6, s_5), (s_6, s_0)\}.
\]

\[
P_3 = P_2, \quad P = P_2.
\]

This \(P\) is the same as the one shown in Example 2.

**Definition 12.** For a given reduced machine \(S\), the algebraic system \(FS(S) = <FS(S), \subseteq, \cap, \cup>\) is defined similarly to Definition 5, where \(FS(S)\) is the set of all the \(FS\) orders of \(S\).

**Theorem 5.** For a given reduced machine \(S\), the algebraic system \(FS(S) = <FS(S), \subseteq, \cap, \cup>\) forms a lattice.
Proof. \( FS(S) \) has the unique maximal element \( P \) and the unique minimal element \( E \). For \( Q, R \in FS(S) \), we see that \( Q \cap R = Q \cap R \) because \( Q \cap R \subseteq Q \cap R \) by definition and any \( (s_1, s_2) \in Q \cap R \) satisfies the three conditions of Definition 6. Similarly to Theorem 1 we see that \( Q \cup R \) is unique. Q.E.D.

Without proof we state the following lemma.

Lemma 9. Let \( R \subseteq S \times S \) be a partial order on \( S \). For \( (s_1, s_2) \in R \), if it does not hold that

\[
\exists s \in S((s_1, s) \in R \land (s, s_2) \in R)
\]

the set \( R' = R - \{(s_1, s_2)\} \) is a maximal partial order smaller than \( R \).

Using this lemma we can generate all \( FS \) orders in \( FS(S) \) for a reduced machine \( S \) from \( P \) to \( E \) iteratively. For the machine \( S \) given in Table I, the lattice \( FS(S) \) is illustrated in Fig. 4.
DEFINITION 13. For a given reduced machine $S$, $P(S)$ is the set of all the positive orders of $S$. The algebraic system $P(S) = \langle P(S), \subseteq, \cap, \cup \rangle$ is defined similarly to Definition 5.

THEOREM 6. For a given reduced machine $S$, the algebraic system $P(S) = \langle P(S), \subseteq, \cap, \cup \rangle$ forms a lattice. Furthermore, $P(S) = SP(S) \cap FS(S)$ (i.e., $P(S)$ is a sublattice of $SP(S)$ and $FS(S)$).

Proof. $P(S)$ has the unique maximal element $P$ and the unique minimal element $E$. For $Q, R \in P(S)$, we see that $Q \cap R = Q \cap R$ because $Q \cap R \subseteq Q \cap R$ and $Q \cap R$ is obviously a positive order. Similarly to Theorem 1 we see that $Q \cup R$ is unique. The relation $P(S) = SP(S) \cap FS(S)$ is obvious from Definitions 10 and 13. Q.E.D.

LEMMA 10. Let $R$ be a maximal partial order smaller than $Q \in P(S)$. Then $R$ is a maximal positive order smaller than $Q$.

Proof. From Lemma 1 and Lemma 8, it follows that $\bar{R} \in P(S)$. Suppose that there exists $R' \in P(S)$ such that $\bar{R} \subseteq R' \subseteq R$. Then from Lemma 5 $\bar{R'} = R'$ and then $R' \subseteq \bar{R}$ follows from Lemma 4, which is a contradiction. Q.E.D.

From these facts we have the following algorithm for computing $P(S)$ for a given reduced machine $S$.

ALGORITHM 3.

(i) Go to Algorithm 2 and compute $P$.

(ii) Let $i = 0$.

(iii) Let $\Phi_0 = \{P\}$.

(iv) Let $\mathcal{K} = 1$, $\mathcal{Y} = \emptyset$ (the empty set).

(v) If $Q^{(i)} \neq E$, list up all the maximal partial orders $R_{j}$'s such that $R_{j} \subseteq Q^{(i)}$, where $\Phi_i = \{Q^{(1)}_{i}, \ldots, Q^{(n)}_{i}\}$, using lemma 9. If $Q^{(i)}_k = E$, go to (ix).

(vi) Go to Algorithm 1 and compute $\bar{R}_{j}$ for each $R_j$.

(vii) Let $\mathcal{Y}$ be replaced by $\mathcal{Y} \cup \mathcal{Y}_{k}$, where $\mathcal{Y}_{k} = \{\bar{R}_{j} \mid R_{j} \subseteq Q^{(i)}_{k}\}$.

(ix) If $k = n_i$ go to (xi). Otherwise go to (x).

(x) Increase $k$ by one and go to (v).

(xi) Let $\Phi_{i+1} = \mathcal{Y} - (\Phi_0 \cup \cdots \cup \Phi_i)$. 
(xii) If $\Phi_{i+1} = \emptyset$ go to (xiv). Otherwise go to (xiii).

(xiii) Increase $i$ by one and go to (iv).

(xiv) Let $P(S) = \Phi_0 \cup \Phi_1 \cup \cdots \cup \Phi_i$ and halt.

Using this algorithm $P(S)$ is computed for the machine $S$ given in Table I and is illustrated in Fig. 5.

---

**Fig. 5.** Lattice of positive orders $P(S)$. 
IV. STATE ASSIGNMENT PROBLEM

In this section we assume that $\Sigma = \{0, 1\}$ and $\Delta = \{0, 1\}$ with the order $0 < 1$ defined on $\Delta$. The case of $|\Delta| > 2$ and $|\Sigma| > 2$ will be discussed in Section V. We construct a binary sequential circuit (BSC) which realizes a given machine $S$ and satisfies the following condition: If some output values of the logic components used in the BSC fail asymmetrically from 0 to 1 (it is assumed that the input makes no failure) the induced failure (if any) on the output of the BSC is also asymmetric from 0 to 1. (Note that this failure on the output includes a failure which is observed after applying some input string to the BSC.) This condition is called the fail-safe condition (FS condition).

When we construct a BSC, it may be intuitively clear that, it is necessary to define a state assignment such that any error of state vectors caused by asymmetric failures $0 \rightarrow 1$ of the logic components does not conflict with positive or FS order, in order that the resulting BSC satisfies the FS condition. In the following, we will present the precise meaning of this statement and give algorithms to obtain such state assignment.

Let $L = \{0, 1\}$ be a Boolean space.

**Definition 14.** A BSC is a system $\langle f, g \rangle$ where $f = (f_1, \ldots, f_m)$, $f_i : L^{m+1} \rightarrow L (i = 1, 2, \ldots, m)$ and $g : L^m \rightarrow L$. A BSC $\langle f, g \rangle$ realizes a machine $S = \langle S, M, \beta, \Sigma, \Delta \rangle$ under a mapping $\psi : S \rightarrow L^m$ ($\psi$ is called a state assignment), or $\langle f, g, \psi \rangle$ is a realization of $S$, if $\psi$ is one-to-one and

$$\forall s \in S, \forall \sigma \in \Sigma \psi(M(s, \sigma)) = f(\psi(s), \xi)$$

$$\forall s \in S \beta(s) = g(\psi(s)),$$

where $\xi$ is a binary variable whose value is equal to that of $\sigma$. A BSC is illustrated in Fig. 6. In this figure, $x(t)$ and $x(t+1)$ denote the present and the next state vectors respectively, and $\xi(t)$ and $\eta(t)$ denote the present input and the present output, that is,

$$x(t+1) = f(x(t), \xi(t))$$

$$\eta(t) = g(x(t)).$$

The vector of functions $f(x, \xi)$ is realized by the state logic, and the function $g(x)$ is realized by the output logic.

If $\langle f, g, \psi \rangle$ satisfies the FS condition, it is called a fail-safe realization (FS realization).
**DEFINITION 15.** The order on $L$ is defined by $0 < 1$. The order on $L^m$ is defined by $\mathbf{x} \leq \mathbf{x}' \iff x_1 \leq x_1', ..., x_m \leq x_m'$, where $\mathbf{x} = (x_1, ..., x_m)$ and $\mathbf{x}' = (x_1', ..., x_m')$.

$L^m$ is a partially ordered set, which is denoted by $\langle L^m, \leq \rangle$. For a realization $\langle f, g, \psi \rangle$ of a given machine $S$, the partially ordered set $\langle \psi(S), \leq \rangle$ is defined as a restriction of $L^m$ to $\psi(S) \subseteq L^m$.

**DEFINITION 16.** For a realization $\langle f, g, \psi \rangle$ of a given machine $S$, let

$Q_\psi \subseteq S \times S$ be defined by

$$(s_1, s_2) \in Q_\psi \iff \psi(s_1) \leq \psi(s_2).$$

Clearly $Q_\psi$ is a partial order on $S$. The partially ordered set $S$ with the order $Q_\psi$ is denoted by $S = \langle S, Q_\psi \rangle$.

**EXAMPLE 4.** Consider the machine $S$ given by Table I. Let a state assignment $\psi$ as shown in Fig. 7 be given. The partially ordered set $S = \langle S, Q_\psi \rangle$ is the same as the one induced on the set $\psi(S)$ by the order on $L^m$. In this case, $\langle S, Q_\psi \rangle$ is equal to $P$ of Examples 2 and 3, and shown in Fig. 3.
**Definition 17.** A realization \( \langle f, g, \psi \rangle \) of a machine \( S \) is called a *positive realization* of \( S \), if \( f(x, \xi)'s \) are all positive with respect to \( x \), i.e.

\[
x \preceq x' \Rightarrow f(x, \xi) \preceq f(x', \xi),
\]

and \( g(x) \) is positive with respect to \( x \).

\[
\psi(S_6) = (111) \\
\psi(S_5) = (101) \\
\psi(S_4) = (011) \\
\psi(S_3) = (001) \\
\psi(S_2) = (010) \\
\psi(S_1) = (000)
\]

**Fig. 7.** Fewest bit assignment for positive realization.

For a given machine \( S \), obviously a positive realization is an FS realization since the failure \( x \rightarrow x' \) such that \( x \preceq x' \) causes the state failure

\[
f(x, \xi) \rightarrow f(x', \xi), \quad \text{where} \quad f(x', \xi) \succeq f(x, \xi),
\]

and the output failure \( g(x) \rightarrow g(x') \), where \( g(x') \succeq g(x) \).

First we investigate the positive realization of \( S \).

**Lemma 11.** If \( \langle f, g, \psi \rangle \) is a positive realization of a machine \( S \), then \( \langle S, Q_\psi \rangle \) is a positive order.

**Proof.**

\[
(s_1, s_2) \in Q_\psi \iff \psi(s_1) \preceq \psi(s_2) \Rightarrow g(\psi(s_1)) \preceq g(\psi(s_2)) \iff \beta(s_1) \preceq \beta(s_2).
\]

\[
(s_1, s_2) \in Q_\psi \iff \psi(s_1) \preceq \psi(s_2) \Rightarrow \forall \xi \in L \ f(\psi(s_1), \xi) \leq f(\psi(s_2), \xi)
\]

\[
\Rightarrow \forall \sigma \in \Sigma, \psi(M(s_1, \sigma)) \preceq \psi(M(s_2, \sigma))
\]

\[
\Rightarrow \forall \sigma \in \Sigma(M(s_1, \sigma), M(s_2, \sigma)) \in Q_\psi.
\]

Q.E.D.

**Lemma 12.** For a realization \( \langle f, g, \psi \rangle \) of a given machine \( S \), if the partially ordered set \( \langle S, Q_\psi \rangle \) is a positive order, we can make the functions \( f(x, \xi) \) and \( g(x) \) positive with respect to \( x \).
Proof. From the assumption, the functions $f$ and $g$ satisfy the following.

\[
\psi(s_1) \leq \psi(s_2) \iff (s_1, s_2) \in \mathcal{Q}_\psi \Rightarrow \beta(s_1) \leq \beta(s_2)
\]

\[
\land \forall \sigma \in \Sigma(M(s_1, \sigma), M(s_2, \sigma)) \in \mathcal{Q}_\psi \Rightarrow g(\psi(s_1)) \leq g(\psi(s_2))
\]

\[
\land \forall \sigma \in \Sigma \psi(M(s_1, \sigma)) \leq \psi(M(s_2, \sigma))
\]

\[
\Rightarrow g(\psi(s_1)) \leq g(\psi(s_2)) \land \forall \xi \in L f(\psi(s_1), \xi) \leq f(\psi(s_2), \xi).
\]

Hence $f$ and $g$ are positive with respect to $x \in \psi(S)$, though the value of $f$ and $g$ remains free for $x \notin \psi(S)$. As proved in Ibaraki and Muroga (1971), we can extend $f$ and $g$ so that they are positive with respect to all $x \in L^m$. Q.E.D.

The number $m$ for a realization $\langle f, g, \psi \rangle$ of a given machine $S$ is that of bits (or DELAY's) used in the BSC. From the above lemmas we have the following theorem.

**Theorem 7.** The fewest bit assignment $\psi$ for a positive realization $\langle f, g, \psi \rangle$ of a given reduced machine $S$ is obtained by finding the least dimensional boolean space $L^m$ for which there exists $\psi$ such that $\langle S, \mathcal{Q}_\psi \rangle$ is a positive order of $S$.

**Example 5.** Consider the machine $S$ given in Table I. As shown in Fig. 5, $S$ has 6 positive orders $Q_0(=P), Q_1, \ldots, Q_5(=E)$. $Q_0$ leads to a three bit assignment as shown in Fig. 7, in which $\langle S, Q_0 \rangle$ is the same as $\langle S, P \rangle$ (see Example 4). It is easily seen that $Q_1, \ldots, Q_5$ cannot yield any assignment with less number of bits. In particular, $Q_5$ corresponds to the assignment by the constant weight code, (i.e., the weights of all $\psi(s_i)$'s are equal). The assignment by the constant weight code requires at least four bits in this case.

Now we turn to the fewest bit assignment for an FS realization of a given machine.

**Definition 18.** Let

\[
S_1 = \langle S_1, M_1, \beta_1, \Sigma, \Delta \rangle \quad \text{and} \quad S_2 = \langle S_2, M_2, \beta_2, \Sigma, \Delta \rangle
\]

be two machines. $S_1$ is a submachine of $S_2$, or $S_2$ is an extension of $S_1$, if $S_1 \subseteq S_2$ and

\[
\forall s \in S_1, \forall \sigma \in \Sigma M_1(s, \sigma) = M_2(s, \sigma)
\]

\[
\forall s \in S_1 \beta_1(s) = \beta_2(s).
\]

If $S_1$ is a submachine of $S_2$ we have that $M_2(S_1, \Sigma) \subseteq S_1$ where

\[
M_2(S_1, \Sigma) = \{M_2(s, \sigma) \mid s \in S_1, \sigma \in \Sigma\}.
\]
Definition 19. Let \( \langle f, g, \psi \rangle \) be a realization of a given machine \( S \). An \( \langle f, g, \psi \rangle \)-extension of \( S \) is a machine \( S' = \langle S', M', \beta', \Sigma, A \rangle \) such that \( |S'| = 2^m \) and \( \langle f, g, \psi \rangle \) is a realization of \( S' \), where \( \psi \) is here considered as an extended one-to-one mapping \( S' \rightarrow L^m \). (Note that \( S \) is then a sub-machine of \( S' \).)

Example 6. For the machine \( S \) given in Table I, there is a realization \( \langle f, g, \psi \rangle \) with an \( \langle f, g, \psi \rangle \)-extension as shown in Table II.

<table>
<thead>
<tr>
<th>State (Input)</th>
<th>Output</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_1 ) (000)</td>
<td>0</td>
<td>( s_1 ) (000)</td>
<td>( s_2 ) (100)</td>
</tr>
<tr>
<td>( s_2 ) (100)</td>
<td>0</td>
<td>( s_1 ) (000)</td>
<td>( s_2 ) (110)</td>
</tr>
<tr>
<td>( s_3 ) (110)</td>
<td>0</td>
<td>( s_4 ) (001)</td>
<td>( s_5 ) (110)</td>
</tr>
<tr>
<td>( s_4 ) (001)</td>
<td>1</td>
<td>( s_1 ) (000)</td>
<td>( s_5 ) (011)</td>
</tr>
<tr>
<td>( s_5 ) (011)</td>
<td>1</td>
<td>( s_1 ) (000)</td>
<td>( s_6 ) (111)</td>
</tr>
<tr>
<td>( s_6 ) (111)</td>
<td>1</td>
<td>( s_8 ) (011)</td>
<td>( s_6 ) (111)</td>
</tr>
<tr>
<td>( a ) (010)</td>
<td>0</td>
<td>( s_1 ) (000)</td>
<td>( s_9 ) (110)</td>
</tr>
<tr>
<td>( b ) (101)</td>
<td>1</td>
<td>( a ) (010)</td>
<td>( s_9 ) (111)</td>
</tr>
</tbody>
</table>

Theorem 8. Let \( \langle f, g, \psi \rangle \) be a realization of a given reduced machine \( S \). If \( \langle f, g, \psi \rangle \) is an FS realization of \( S \), then \( \langle S, Q_0 \rangle \) is an FS order of \( S \) and there exists an \( \langle f, g, \psi \rangle \)-extension \( S' \) whose maximum positive order \( P' \) (which is defined by Definition 8) satisfies

\[
\forall (s_1, s_2) \in W \times Wf(\psi(s_1) \leqslant \psi(s_2) \Rightarrow (s_1, s_2) \in P'),
\]

where \( W \) is inductively defined as follows:

\[
W_0 = S \cup \{s' \in S' | \exists s \in S \psi(s) \leqslant \psi(s')\}
\]

\[
W_k = W_{k-1} \cup \{s' \in S' | \exists s \in W_{k-1}, \exists \xi \in L f(\psi(s), \xi) \leqslant \psi(s')\}
\]

for \( k = 1, 2, \ldots \)

\[
W = W_\infty.
\]
(Since \( S' \) is finite and \( W_k \) is nondecreasing, \( W_k = W_{k-1} \) holds for a finite \( k \) \((\leq 2^m)\). Then \( W = W_k \) holds.) Conversely, if \( \langle S, Q_{\phi} \rangle \) is an FS order of \( S \) and there exists an \( \langle f, g, \psi \rangle \)-extension \( S' \) whose maximum positive order \( P' \) satisfies the above condition, then \( \langle f, g, \psi \rangle \) is an FS realization of \( S \). (Note that \( P' \) is in general a pseudo order since \( S' \) may not be reduced.)

Proof. For \( s_1, s_2 \in S \), if \( \psi(s_1) \leq \psi(s_2) \), the state error \( s_1 \rightarrow s_2 \) may occur, and \( (s_1, s_2) \) must be an element of \( P \), i.e., \( \langle S, Q_{\phi} \rangle \) is an FS order. Next note that it is possible to reach any state vector \( \psi(s) \), \( s \in W \) of the BSC \( \langle f, g, \psi \rangle \) by applying suitable inputs and suitable failures \( \psi(s') \rightarrow \psi(s'') \) such that \( \psi(s') \leq \psi(s'') \). Therefore, if \( \psi(s_1) \leq \psi(s_2) \) for \( s_1, s_2 \in W \), the state error

---

**Fig. 8.** Fewest bit assignment for FS realization.

**Fig. 9.** Partially ordered set \( \langle S, Q_{\phi} \rangle \).
s_1 \rightarrow s_2 may occur, and hence (s_1, s_2) must be an element of P'. The converse is obvious. Q.E.D.

Note that \( \langle S, Q_\psi \rangle \) is an FS order of S if and only if \( Q_\psi \subseteq P \). Therefore from Theorem 8, it is in principle possible to obtain the fewest bit FS realization of S, if we search all possible \( \langle f, g, \psi \rangle \)-extensions of every assignment \( \psi \) satisfying \( Q_\psi \subseteq P \) and check whether they satisfy the conditions of Theorem 8. It is, however, computationally difficult to execute due to the excessive number of such possibilities. On the other hand, the search for the fewest bit positive realization appears much easier, since the number of positive orders of S is usually considerably smaller than that of FS orders. In general, however, it seems possible that there exists an S for which an FS realization requires less number of bits than any positive realization.

**Example 7.** For the state assignment given in Table II and Fig. 8 the partially ordered set \( \langle S, Q_\psi \rangle \) is given in Fig. 9. The partially ordered set \( \langle S', Q_\psi \rangle \) and the pseudo ordered set \( \langle S', P' \rangle \) are illustrated in Fig. 10 and Fig. 11. \( \langle S, Q_\psi \rangle \) and \( \langle S', P' \rangle \) satisfy the conditions given in Theorem 8. The functions f and g are given by

\[
\begin{align*}
  f_1(x_1, x_2, x_3, \xi) &= (x_1 + x_2 + \bar{x}_3)\xi \\
  f_2(x_1, x_2, x_3, \xi) &= x_1 x_3 \bar{\xi} + (x_1 + x_2 + x_3)\xi \\
  f_3(x_1, x_2, x_3, \xi) &= x_1 x_2 \bar{\xi} + x_3 \xi \\
  g(x_1, x_2, x_3) &= x_3.
\end{align*}
\]

![Fig. 10. Partially ordered set \( \langle S', Q_\psi \rangle \).](image-url)
This gives an FS realization of the machine $S$, which is not a positive realization. Note that $f_1$ is negative with respect to $x_3$.

V. EXTENSION TO MULTIPLE INPUT MULTIPLE OUTPUT SEQUENTIAL CIRCUITS

In this section, we give a brief sketch of a construction method of fail-safe sequential circuits with multiple inputs and/or multiple outputs, which are realization of sequential machines with $|\Sigma| \geq 3$ and/or $|\Delta| \geq 3$.

DEFINITION 20. Let one-to-one mappings $\gamma: \Sigma \rightarrow L^p$ and $\delta: \Delta \rightarrow L^q$ be given ($\gamma$ and $\delta$ are binary codings of $\Sigma$ and $\Delta$ respectively). A BSC (with multiple inputs and multiple outputs) is a system $\langle f, g \rangle$ where $f = (f_1, \ldots, f_m)$, $f_i: L^{m+p} \rightarrow L$ ($i = 1, 2, \ldots, m$) and $g = (g_1, \ldots, g_q)$, $g_j: L^m \rightarrow L$ ($j = 1, 2, \ldots, q$). A BSC $\langle f, g \rangle$ realizes a machine $S = \langle S, M, \beta, \Sigma, \Delta \rangle$ under a mapping $\psi: S \rightarrow L^m$ and the above input and output codings, or $\langle f, g, \psi \rangle$ is a realization of $S$, if $\psi$ is one-to-one and

$$\forall s \in S, \forall \sigma \in \Sigma \psi(M(s, \sigma)) = f(\psi(s), \gamma(\sigma))$$
$$\forall s \in S \delta(\beta(s)) = g(\psi(s)).$$

The output vector of $S$, i.e., $\delta(\tau), \tau \in \Delta$, is denoted by $y$. 

![Fig. 11. Pseudo ordered set $\langle S', P' \rangle$.](image-url)
In this case, a BSC satisfies the fail-safe condition if for any asymmetric failures of logic components from 0 to 1, the induced failure (if any) on each output is asymmetric from 0 to 1.

Now assume that the coding $\delta: \Delta \rightarrow L^a$ satisfies

$$\forall \tau, \tau' \in \Delta (\delta(\tau) \leq \delta(\tau') \Rightarrow \tau \leq \tau'),$$

where $\leq$ on $\Delta$ is given prior to Definition 7. (Note that in the case of single output of Section IV, the coding $\delta$ implicitly assumed is the identity mapping $\delta: 0 \mapsto 0, 1 \mapsto 1$. Thus the above condition is obviously satisfied.)

With $\delta$ satisfying this condition, any realization $\langle f, g, \psi \rangle$ of $S$ subject to the fail-safe condition can be considered as a fail-safe realization of $S$. This is because any asymmetric failure of logic components from 0 to 1 causes the output failure $y \rightarrow y'$ for which $y \leq y'$ holds, and $y \leq y' = \delta(\tau) \leq \delta(\tau') \Rightarrow \tau \leq \tau'$ if $y = \delta(\tau)$ and $y' = \delta(\tau')$ for some $\tau, \tau' \in \Delta$. Therefore, we see that the concept of the fail-safe machines with $|\Delta| = 2$ can be directly extended to machines with $|\Delta| \geq 3$, as far as outputs fail in such a way that $\tau' \in \Delta$ satisfying $\delta(\tau') = y'$ exists for any resulting output $y'$. In case there exists no $\tau' \in \Delta$ such that $\delta(\tau') = y'$, we consider that

$$\delta^{-1}(y') = \{\tau \in \Delta \mid \delta(\tau) \leq y' \land \text{no } \tau'(\neq \tau) \in \Delta(\delta(\tau) \leq \delta(\tau') \leq y')\},$$

and define $\tau_i < \delta^{-1}(y')$ if $\tau_i \in \delta^{-1}(y')$. This means that failures on outputs such as $\tau_i \rightarrow \{\tau_{i_1}, \tau_{i_2}, \ldots, \tau_{i_k}\}$, where $\tau_i \in \{\tau_1, \tau_2, \ldots, \tau_{1_i}\}$, are defined to be safe. With this definition, the concept of the fail-safe machines is consistently extended to the multiple output case.

To obtain a BSC realizing $S$ and satisfying the fail-safe condition, theory developed in Section IV can be easily modified by replacing $g$ by $g$. The detail is hence omitted.

Note that the input coding $\gamma: \Sigma \rightarrow L^p$ can be any one-to-one mapping since we do not assume the input failure.

As an example, consider the case in which $\Delta = \{\tau_1, \tau_2, \tau_3\}$ with the

![Fig. 12. Partially ordered set $\Delta$.](image-url)
partial order $\tau_1 < \tau_2$ and $\tau_1 < \tau_3$. The partially ordered set $\langle \Delta, \leq \rangle$ is illustrated in Fig. 12. The coding $\delta: \Delta \rightarrow L^2$ may be given as shown in Fig. 13, in which $\delta(\tau_1) < \delta(\tau_2)$, $\delta(\tau_1) < \delta(\tau_3)$ and the condition of $\delta$ as mentioned above is obviously satisfied. $\delta^{-1}(1, 1)$ is considered as $\{\tau_2, \tau_3\}$ and failures such as $\tau_2 \rightarrow \{\tau_2, \tau_3\}$ and $\tau_3 \rightarrow \{\tau_2, \tau_3\}$ are considered safe. From this, we can obtain a $BSC$ with two outputs.

\[
\begin{align*}
\delta(\tau_2) &= (10) \\
\delta(\tau_3) &= (01) \\
\delta(\tau_1) &= (00)
\end{align*}
\]

Fig. 13. Binary coding $\delta$ of $\Delta$.

VI. CONCLUSIONS

Let $n$ be the number of states of a given machine $S$. We can realize (not necessarily an $FS$ realization) the machine $S$ with $\lceil \log_2(n) \rceil + 1$ bits, where $[a]$ is the greatest integer smaller than $a$. For the state assignment by the half weight code, the same number of bits are asymptotically required for an $FS$ realization of $S$ if $n$ is sufficiently large [7]. But for small $n$, we can further reduce the number of bits necessary for an $FS$ realization. For this purpose, $FS$ orders and positive orders are introduced and algorithms for obtaining an $FS$ realization and a positive realization of a given machine $S$ by a $BSC$ with the fewest bits are developed.

Although the algorithm for a positive realization with the fewest bits works efficiently, an $FS$ realization with the fewest bits seems difficult to find because of the excessive number of possibilities to be examined. The improvement of the latter algorithm will be one of the main subjects in the future research.
ACKNOWLEDGMENT

The authors are indebted to Professor H. Mine of Kyoto University for his support of the study.

RECEIVED: April 9, 1971

REFERENCES


