

SEQUENTIAL MACHINE
WITH FAULT-DETECTION CAPABILITY

by

CHENG SAN CHIANG

Submitted in partial fulfillment of the
requirements for the degree of
Master of Applied Science.

Department of Electrical Engineering
Faculty of Science and engineering
University of Ottawa
Ottawa, Canada
January, 1972

ABSTRACT

In the conventional works in fault detection of sequential machines, the experiments are usually restricted on the assumption that the faults do not cause an increase in the number of states of the given machine. In actual practice this condition is not necessarily satisfied. In this work, this restriction of faults not causing an increase in the number of states is eliminated and procedures are developed which locate the faults by the application of the checking sequences to the modified versions of the machines. The concept of fault transition functions is introduced and an adaptive procedure to find the fault transition functions by noting the response to the checking sequence of the machine is also presented.

ACKNOWLEDGEMENTS

The author wishes to express his deep gratitude to his supervisor, Professor G. S. Glinski, for his encouragement and guidance throughout the course of the research.

No thanks can express his gratitude to Dr. C. L. Sheng and Dr. S. R. Das for their inspiring and critical discussions without which this thesis would not have been possible.

Author's sincere appreciation and thanks are due to Dr. W. K. Chung for his stimulating discussions and critical comments on various aspects of this work.

Gratitude is also expressed to the staff and graduate students of the Department of Electrical Engineering for their friendship.

Finally, the author also wishes to acknowledge the National Research Council of Canada for the financial assistance.

TABLE OF CONTENTS

	PAGE
ABSTRACT	i
ACKNOWLEDGEMENTS	ii
CHAPTER 1 : INTRODUCTION	1
CHAPTER 2 : BASIC DEFINITION AND NOTATION	5
CHAPTER 3 : CHECKING EXPERIMENTS FOR SEQUENTIAL MACHINES	8
3.1 A Counter-cycle-type Sequential Machine	8
3.2 Modification of a Sequential Machine to a CC-type Machine	19
3.3 Failures Increasing the Number of States	22
3.4 Extension of a CC-type Sequential Machine to Include Type of Faults Increasing Number of States	25
CHAPTER 4 : FAULT TRANSITION FUNCTIONS	29
4.1 Fault e-transitions	29
4.2 Fault X_u -transitions	36
CHAPTER 5 : CONCLUDING REMARKS	45
REFERENCES	46
VITA	

CHAPTER 1

INTRODUCTION

With the increasing use of modules and integrated circuits in the design of sequential machines, our freedom to check all the interconnections is rather limited. Therefore it becomes important to be able to determine whether or not a given machine is operating correctly through some external measurements. Thus the decision whether a given machine precisely represents the terminal behavior specified by its flow table is made by applying appropriate input sequences to the input terminals of the machine and observing the resulting output sequences.

The idea of performing such experiments on sequential machines was first proposed by Moore¹ in 1956. Since then several methods²⁻⁷ for fault detection of sequential machines have been developed. However, the process of designing checking sequences by applying the above methods is usually complicated and lengthy, except in the cases where the given machines are very simple. In 1964, Hennie⁸ proposed another approach for the design of fault detection experiments for sequential machines, which is more practical when the given machine is strongly connected and possesses a distinguishing sequence, and when the number of states of the machine is not increased as a result of the malfunction. But if the given machine has no distinguishing sequence, then the method becomes a little involved and the procedure yields very long checking experiments.

There exist other approaches which provide relatively simple procedures. For a given sequential machine, instead of finding a checking sequence for the machine directly, we synthesize a sequential machine which is slightly different from the given machine, but satisfies the given specifications and moreover has a checking sequence that can be readily found by following a simple procedure. From this point of view, Kohavi and Lavalley⁹ added some output logic to the machine in order that the machine will possess special distinguishing sequences. The length of checking sequences constructed with this method is usually short (the order of its upper bound of length is mn^3 , where m is the number of input symbols and n is the number of states of the machine), but an important limitation of this method is that the original machine must be strongly connected and the nature of faults be such as not to cause an increase in the number of states of the machine.

In an approach suggested by Murakami, Kinoshita and Ozaki¹⁰ an extra input symbol is added to the original machine in order to get a modified machine which is strongly connected and possesses a distinguishing sequence. By this method, the procedure of constructing the checking sequences becomes very systematic and the length of checking sequences also is considerably shortened (the order of its upper bound of length is mn^2). But this method like other methods mentioned above is applicable only when the number of states of the

machine is not increased by the presence of faults. Thus one major disadvantage of the aforementioned existing procedures for designing checking sequences is the restriction that the faults do not increase the number of states of the machine.

When the number of states of a given sequential machine is less than 2^k , where k is the number of memory elements, there always exists the possibility of the number of states being increased when some faults occur in the given machine. In such cases, in spite of the correctness of the response to the checking sequence that may be designed with any of the aforementioned methods, there is no guarantee that the given machine still operates as it was originally designed to do; there may occur some faults which may cause an increase in the number of states of the machine and the given machine would no longer be the same machine or an equivalent of the original one.

To solve such a practical problem, in this work, we modify a given n -state sequential machine into a counter cycle type sequential machine by the method of Murakami et al. and add another output symbol in order to distinguish the output corresponding to an input X , which induces the counter cycle, when the machine is in any one of the given n states, from the output corresponding to the same input X when the machine is in any state other than the given n states. A checking sequence for a sequential machine constructed in this

manner is more practical, being independent of the restriction of fault not causing an increase in the number of states of the machine. Moreover, we have attempted to find the faulty transition functions by observing the output sequences corresponding to the checking sequences.

In Chapter 2. basic assumptions and notations are presented.

The method of Murakami et al. of constructing a checking sequence is presented in the first part of Chapter 3 based on the assumption that the faults do not cause an increase in the number of states of the machine. while in the last part the assumption is abandoned.

In Chapter 4. an adaptive procedure for finding the fault transition functions by observing the output sequence corresponding to the checking sequence is discussed.

In Chapter 5. some concluding remarks are given

CHAPTER 2

BASIC DEFINITIONS AND NOTATIONS

The sequential machines considered here are completely specified, finite-state, deterministic, synchronous sequential machines. Also a machine on which an experiment will be conducted is considered to be a sealed black box, i.e. we assume that only its input and output terminals are accessible.

Unless otherwise specified, in the present dissertation we use the following notations:

X, e : input symbols,

Z, r : output symbols,

S_i : state,

n : number of states,

k : number of memory elements,

f : state transition function,

g : output function.

An experiment is a process of applying one or more input sequences to the input terminals of a sequential machine, observing the resulting output sequences, applying additional input sequences, if necessary, observing the new output sequences, and so on, until some conclusions can be drawn about the internal behavior of the machine. The experiment in which the applied input sequence is

completely determined in advance is called a preset experiment. In the present study, an experiment performed on a machine will be assumed to be preset, unless stated to the contrary. If the decision as to what input sequence is to be applied next is based on the previous response of the machine, then the experiment will be called an adaptive experiment.

If we are presented with an actual machine and also the flow table from which the machine was built and are required to decide whether or not the given machine is operating correctly by applying an appropriate input sequence to the machine and observing the corresponding output sequence that the machine produces, then the entire process is called a checking experiment or a fault detection experiment. When an appropriate starting state is specified and the machine is started from that initial state, then the input sequence as can be found from the checking experiment is called a checking sequence for the machine.

An input sequence is called a synchronizing sequence for a given machine if its application to the machine always leaves the machine in a certain final state no matter what the initial state is.

An input sequence is called a distinguishing sequence for a machine if and only if by observing the output sequence produced in response to such an input sequence, the initial state of the

machine can be uniquely determined. When a given sequential machine has no distinguishing sequence, the initial state of the machine can be characterized by a set of characterizing sequences. Suppose a machine is in the same state at several points of an experiment. If we apply a different one of the characterizing sequences at each point in the experiment, then the state in question can be uniquely identified from the responses to the characterizing sequences.

An input sequence X is called a homing sequence for a machine if and only if the output sequence in response to X is sufficient to determine the final state of the machine uniquely no matter what initial state it is in.

CHAPTER 3

CHECKING EXPERIMENTS FOR SEQUENTIAL MACHINES

In this chapter, the properties of a counter-cycle-type sequential machine are introduced and the design method of a preset checking sequence for such a machine is presented. An example is given to show that the checking sequence for such a machine fails to detect the occurrence of faults that cause an increase in its number of states. An extension of a counter-cycle-type sequential machine which enables the checking sequence to detect such failures are also developed.

3.1. A Counter-cycle-type Sequential Machine

A counter cycle is a cycle in the output sequence of a machine obtained by successive applications of the input symbol X.

Definition 3.1.1: A counter-cycle-type (CC-type) sequential machine with n states is a sequential machine which has the following two properties:

- (1) $f(S_i, X) = S_{i+1}, \quad g(S_i, X) = 0, \quad \text{for } i=1, 2, \dots, n-1;$
- (2) $f(S_n, X) = S_1, \quad g(S_n, X) = 1;$

where f is the state transition function and g is the output function and the output symbols 0 and 1 can be interchanged in properties (1) and (2).

A CC-type sequential machine has a special characteristic. There exists an input symbol X for this type of machine such that by applying the input symbol X consecutively n times, a cyclic

state transition over all the n states of the machine is induced and the n corresponding output symbols from the machine are the same for all the state transitions except for one as shown in Fig. 1.

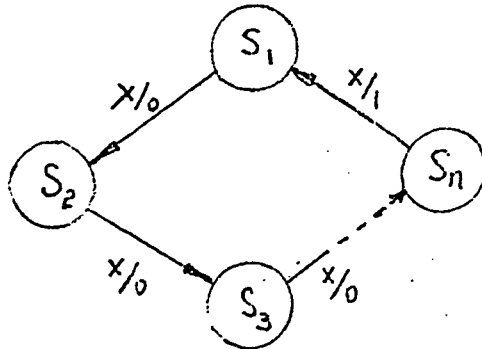


Fig. 1. A CC-type sequential machine.

3.1.1. D_i -sequences

Consider an input sequence which consists of n consecutive occurrences of the input symbol X . Then, from the position of the output symbol 1 in the output sequence corresponding to this input sequence, we can determine what is the initial state and what is the final state of the machine. Therefore, we use such an input sequence as both a homing sequence and a distinguishing sequence for the given sequential machine. For example, if the output symbol 1 appears at the $(n-i+1)$ th position in the corresponding output sequence, then the state S_i is the initial state.

However, in place of the input sequence consisting of n occurrences of the input symbol X , we may sometimes use a shorter input-output distinguishing sequence defined as follows

Definition 3.1.2. For a CC-type sequential machine with n states, we define a D_i -sequence of the state S_i to be an input-output

sequence of length $(n-i+1)$ having the following properties:

(a) The input sequence consists of $(n-i+1)$ occurrences of the input symbol X .

(b) The corresponding output sequence is a sequence whose first $(n-i)$ symbols are the output symbol 0 and whose last symbol is the output symbol 1.

If a given sequential machine has a state transition function and an output function satisfying the properties as outlined in definition 3.1.1, and if it also satisfies the properties of D_i -sequence, then by applying such an input sequence, the initial state S_i of the given sequential machine prior to the application of the input sequence can be determined and the final state of the machine will always be S_1 independent of the value of i , i.e. the machine is sent to state S_1 no matter what initial state S_i it is in. Therefore, we can use the D_i -sequence both as a distinguishing sequence and as a homing sequence of the given sequential machine. Thus we have

If the properties of a D_i -sequence are satisfied by an n state CC-type sequential machine, then the initial state of the machine for this experiment is state S_i and the final state is state S_1 .

3.1.2. T_i -sequences

Assume that the state of a given CC-type sequential machine is S_j . We can always change the state from S_j to S_i ($i > j$) by applying an input sequence consisting of $(i-j)$ consecutive occurrences of the input symbol X . That is, there always exists a transfer sequence which induces state transition from one state to the other for this type of machines.

However, in order to formulate a systematic procedure for obtaining a checking sequence, we will only consider transfer sequences that will induce state transition from S_1 to another state S_i . Thus, we define a T_i -sequence as follows.

Definition 3.1.3: For a CC-type sequential machine, we define a T_i -sequence to be an input-output sequence of length $(i-1)$ with the following properties:

- (a) The input sequence consists of $(i-1)$ occurrences of the input symbol X .
- (b) The output sequence consists of $(i-1)$ occurrences of the output symbol 0 .
- (c) The input sequence induces a state transition from state S_1 to state S_i .

3.1.3. L_n -sequence

There always exists an input-output sequence pair from which we can determine whether or not a given sequential machine is a CC-type sequential machines. We define such an input-output sequence pair as follows.

Definition 3.1.4 For a CC-type sequential machine with n states. L_n sequence is a sequence of length $(2n-1)$ having the following properties:

(a) The input sequence consists of $(2n-1)$ consecutive occurrences of the input symbol X .

(b) The output sequence consists of $n-1$ consecutive occurrences of the output symbol 0 , followed by one output symbol 1 and next followed again by $n-1$ consecutive occurrences of the output symbol 0 .

Theorem 3.1 A sequential machine which satisfies the properties of L_n -sequence has n or more states.

Proof The L_n sequence can be written as follows:

Time:	t_1	t_2	\dots	t_{n-1}	t_n	t_{n+1}	\dots	t_{2n-1}	
Input :	X	X	X	X	X	X	X	X	
State :	S_{t_1}	S_{t_2}	$S_{t_{n-1}}$	S_{t_n}	$S_{t_{n+1}}$	$S_{t_{2n-1}}$	(3.1)		
Output:	0	0	0	1	0	0			

Suppose that a sequential machine satisfies the above L_n -sequence, and let t_i and t_j be two time intervals such that $1 \leq i < j \leq n$. If the sequential machine is in state S_{t_i} , then the sequential machine satisfies the input-output sequence given by

Length:	$n-i+1$	
Input :	$X X X \dots X X X$	(3.2)
Output:	$0 0 0 \dots 0 0 1$	

However, if the sequential machine is in state S_{t_j} , then it does not satisfy the input-output sequence given by Eq. (3.2) Thus, the state S_{t_i} is different from the state S_{t_j} . Therefore, the n states, $S_{t_1}, S_{t_2}, \dots, S_{t_n}$ are all different. Consequently, the sequential machine has at least n distinct states.

Theorem 3.2: If a sequential machine having n states satisfies the properties of L_n -sequence, then it is a CC-type sequential machine as defined by Definition 3.1.1.

Proof From Eq. (3.1) of Theorem 3.1 and the fact that all the n states S_{t_i} ($i=1,2,\dots,n-1$) are different, we have $f(S_{t_i}, X)=S_{t_{i+1}}$ and $g(S_{t_i}, X)=0$ for $i=1,2,\dots,n-1$. That is, the given sequential machine had property (1) of Definition 3.1.1.

Since the given sequential machine has n states, $S_{t_{n+1}}$ must be equal to one of the n states, $S_{t_1}, S_{t_2}, \dots, S_{t_n}$. On the other hand, the given sequential machine in the state $S_{t_{n+1}}$ satisfies the following input-output sequence

Length	n-1					
Input	X	X	...	X	X	(3.3)
Output	0	0	...	0	0	

Amongst the n states, $S_{t_1}, S_{t_2}, \dots, S_{t_n}$, state S_{t_1} is the only one such that if the machine is in that state then the input-output sequence given by Eq.(3.3) is satisfied. Thus, if the given machine satisfies the properties of an L_n -sequence, we have $S_{t_{n+1}} = S_{t_1}$ and

$g(S_t, X) = 1$, that is. the given sequential machine has property (2) of Definition 3.1.1.

It should be noted here that even though a sequential machine has the properties (1) and (2) of Definition 3.1.1. the machine does not satisfy the properties of L_n -sequence unless the initial state is S_1 . Thus, in order to examine, by using the L_n -sequence, whether or not a given sequential machine has properties (1) and (2). we need to have the given sequential machine in state S_1 before applying the L_n -sequence. Since we know that a given sequential machine is definitely in state S_1 if the output of the machine for the input symbol X is 1. so. before applying the L_n -sequence. we simply apply the input symbol X to the machine successively until the machine delivers an output symbol 1, and then we apply the L_n -sequence. We then compare the output of the machine with the corresponding output of the L_n -sequence to check their agreement.

Example 3.1: Let us consider a CC-type sequential machine with three states. Then the D_i -sequences and T_i -sequences defined by Definition 3.1.2 and 3.1.3 respectively are shown in Table 1. The L_n -sequence for the machine defined by Definition 3.1.4. is

$$L_3 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (3.4)$$

The state transition diagram of a one-input sequential machine with three states which satisfies the above L_3 -sequence is unique and

is given in Fig. 2.

Table 1. D_i -sequences and T_i -sequences for $n=3$.

	D_i -sequence	T_i -sequence
S_1	$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$	Λ
S_2	$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$
S_3	$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$

Λ The sequence of length zero.

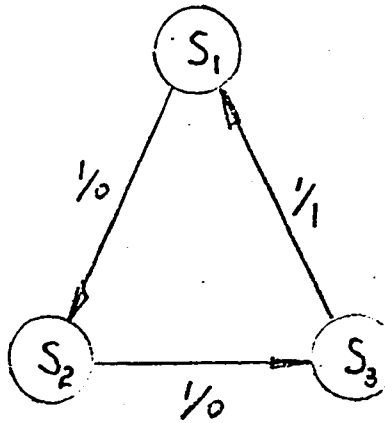


Fig 2 State transition diagram.

3.1.4. Designing Checking Sequences for the CC-type Sequential Machine

We assume first that the faults are restricted to the type which does not cause an increase in the number of states of the machine. This restriction will be abandoned subsequently in Section 3.4. of the thesis.

The machine considered here is a CC-type sequential machine.

with n states and m inputs. Assume that the machine is in state S_1 . (It is implied that if the machine is not originally in state S_1 , we can bring the machine in state S_1 by applying the input symbol X in succession until the machine delivers an output 1). The design procedure is given below.

Procedure 1

Step 1: Construct the L_n -sequence and define an input-output sequence

U_n such that

$$U_n = L_n \begin{pmatrix} X \\ 1 \end{pmatrix}$$

Step 2: Construct the W_{ij} -sequence as follows:

$$W_{ij}\text{-sequence} = (T_i\text{-sequence}) \begin{pmatrix} X_j \\ Z_j \end{pmatrix} (D_k\text{-sequence})$$

for every combination of state S_i ($i=1, 2, \dots, n$) and input symbol

X_j ($j=1, 2, \dots, m-1$); excluding the input symbol X .

Step 3: Concatenate all the W_{ij} sequences obtained above in an arbitrary order after the U_n -sequence.

The sequence thus obtained is a checking sequence for the CC-type sequential machine. The different steps in the procedure are explained below.

Step 1: If we apply the L_n -input sequence to the machine in state S_1 , then the final state of the machine is S_n , provided that the machine behaves properly, in order to formulate a checking sequence systematically, it is convenient to have the state S_1 as the final state of the machine.

Therefore, we use the U_n -sequence to examine whether or not the given machine has properties of Definition 3.1.1 and to get the machine into the final state S_1 .

Step 2: If we apply the W_{ij} -sequence to the given sequential machine which is now in the initial state S_1 , then the T_i -sequence induces state transition from S_1 to S_i , the input (X_j)-output (Z_j) sequence checks whether the output of the machine is indeed Z_j when the input X_j is applied in state S_i , and the D_k -sequence examines whether or not the resulting state is indeed S_k . Since the state transition of the machine induced by the application of input symbol X was already examined by application of the U_n -sequence, the W_{ij} -sequence is constructed for every combination of states S_i ($i=1, 2, \dots, n$) and input symbols X_j ($j=1, 2, \dots, m-1$); excluding the input symbol X .

Step 3: Since each state transition can be examined to see whether it is correct by the application of the W_{ij} -sequence only it has been confirmed by the U_n -sequence that the X -transitions of the machine (i.e. the transitions induced by applying the input symbol X) is correct, so we next concatenate all the W_{ij} -sequences. As the final state obtained by applying each W_{ij} -sequence to the machine in state S_i is always S_1 if the machine behaves properly, we can put all the W_{ij} -sequences in any arbitrary order.

Example 3.2: Let us consider the sequential machine M_1

given by Table 2.

Table 2. Sequential machine M_1 .

	0	1
S_1	$S_3 \cdot 1$	$S_2 \cdot 0$
S_2	$S_2 \cdot 0$	$S_3 \cdot 0$
S_3	$S_1 \cdot 1$	$S_1 \cdot 1$

The machine M_1 is a CC-type sequential machine; since it has three states, the D_i -sequences and T_i -sequences of M_1 are as shown in Table 1 and the U_n -sequence of M_1 is given by

$$U_3 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \quad (3.5)$$

The W_{ij} -sequences are shown in Table 3.

Table 3. The W_{ij} -sequences of M_1 .

	0
S_1	$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$
S_2	$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$
S_3	$\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$

Therefore, the checking sequence and its associated correct output sequence for M_1 is given by

1 1 1 1 1 1 0 1 1 0 1 1 1 1 0 1 1 1
0 0 1 0 0 1, 1 1 0 0 0 1 0 0 1 0 0 1

3.2. Modification of a Sequential Machine to a CC-type Machine

In the preceding section we have dealt with the properties of CC-type sequential machines. In this section we will consider sequential machines which are not of the CC-type. We can transform a given non-CC-type sequential machine with n states to a sequential machine which has a counter cycle by applying the following procedure.

We introduce a new input symbol e and we add an additional column for it in the state transition table of the given non-CC-type sequential machine. We then define the state transition function f and the output function g of the given sequential machine corresponding to this new input e in the following manner

$$f(S_i, e) = S_{i+1}, \quad g(S_i, e) = 0. \quad (i=1, 2, \dots, n-1)$$
$$f(S_n, e) = S_1, \quad \text{and} \quad g(S_n, e) = 1.$$

The resulting sequential machine thus behaves as a CC-type sequential machine. Thus by the addition of a column corresponding to the introduction of a new input symbol in the state diagram, any given sequential machine can be transformed to a CC-type machine. The procedure to find the checking sequence for such a CC-type machine which can be obtained by the above transformation can be

readily formulated by replacing the input symbol X by the new input symbol e in the procedure as given in Section 3.1.

Example 3 3 Let us consider a non-CC-type sequential machine M_2 as given by Table 4. and transform M_2 to a CC-type sequential machine M_2' and then find the checking sequence for M_2' . If we modify M_2' by the aforementioned procedure, we obtain the CC-type machine M_2'' which is given by Table 5.

Table 4. Sequential machine M_2 .

	0	1
S_1	$S_3, 0$	$S_3, 1$
S_2	$S_3, 0$	$S_1, 0$
S_3	$S_2, 1$	$S_3, 1$

Table 5 Modified machine M_2' of M_2 .

	0	1	e
S_1	$S_3, 0$	$S_3, 1$	$S_2, 0$
S_2	$S_3, 0$	$S_1, 0$	$S_3, 0$
S_3	$S_2, 1$	$S_3, 1$	$S_1, 1$

The D_i -sequences and the T_i -sequences of M_2' are shown in Table 6.

Table 6. The D_i -sequences and T_i -sequences of M_2' .

	D_i -sequence	T_i -sequence
S_1	$\begin{pmatrix} e & e & e \\ 0 & 0 & 1 \end{pmatrix}$	Λ
S_2	$\begin{pmatrix} e & e \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} e \\ 0 \end{pmatrix}$
S_3	$\begin{pmatrix} e \\ 1 \end{pmatrix}$	$\begin{pmatrix} e & e \\ 0 & 0 \end{pmatrix}$

The U_n -sequence of M_2' is

$$U_3 = \begin{pmatrix} e & e & e & e & e & e \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

The W_{ij} -sequences of M_2' are shown in Table 7.

Table 7. The W_{ij} -sequences of M_2' .

	0	1
S_1	$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} e \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} e \\ 1 \end{pmatrix}$
S_2	$\begin{pmatrix} e \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} e \\ 1 \end{pmatrix}$	$\begin{pmatrix} e \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} e & e & e \\ 0 & 0 & 1 \end{pmatrix}$
S_3	$\begin{pmatrix} e & e \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} e & e \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} e & e \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} e \\ 1 \end{pmatrix}$

Therefore, the checking sequence and its associated correct output sequence of M_2' is given by

$$\begin{array}{cccccccccccc}
 e & e & e & e & e & e & 0 & e & e & 0 & e & e & e & 1 & e \\
 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
 & & & & & & & & e & 1 & e & e & e & e & e & 1 & e \\
 & & & & & & & & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1
 \end{array} \tag{3.6}$$

3.3. Failures Increasing the Number of States

So far we have assumed that failures do not increase the number of states of a given machine. This condition is not often satisfied in actual practice.

Since the minimal number of secondary state variables, k , needed by an n state machine is such that

$$(\log_2 n) + 1 > k \geq (\log_2 n)$$

where k is a positive integer, the realization of the state table requires k memory elements and the physical circuit which realizes the state transition table possesses exactly 2^k states. When the given machine has of states $n < 2^k$, the number of states could increase because of faults. In this case, the machine could operate as another machine, even if the output sequence corresponding to the checking sequence is the same as that of the original machine.

The following example will prove the above statement.

Example 3.4 Let us consider the sequential machine M_2' given by Table 5. Since the sequential machine M_2' has three states, the required number of memory elements is $k=2$ and the machine possesses exactly $2^k=4$ states. The entries corresponding to the next state and output columns of state S_4 are not specified in the given transition table but they are uniquely and completely specified when the design is completed. Assume that the transition from state S_4 under input e is to state S_2 and the output is 0 i.e.

$$f(S_4, e) = S_2, \tag{3.7}$$

and

$$g(S_4, e) = 0.$$

Suppose now that there occurs some kind of failure which causes the transition from state S_2 to state S_4 under input 1, i.e.

$$f(S_2, 1) = S_4. \tag{3.8}$$

Then the given machine operates as another machine as shown in Table 8.

Table 8 The fault machine M_3 of the machine M_2'

	0	1	e
S_1	$S_3, 0$	$S_3, 1$	$S_2, 0$
S_2	$S_3, 0$	$S_4, 0$	$S_3, 0$
S_3	$S_2, 1$	$S_3, 1$	$S_1, 1$
S_4	- - -	- - -	$S_2, 0$

We may note that if the transition functions and the output functions of M_3 starting in state S_4 are:

$$f(S_4, 0) = S_3 \quad g(S_4, 0) = 0.$$

$$f(S_4, 1) = S_3 \quad \text{and} \quad g(S_4, 1) = 1;$$

the machine M_3 is equivalent to the machine M_2' obtained by merging state S_4 with state S_1 . Besides all these types of equivalent machines, the machine M_3 may also operate as another machine with four states.

The checking sequence and its associated correct output sequence for the given machine M_2' is given below.

e	e	e	e	e	e	0	e	e	0	e	e	e	0	e	e	
0	0	1	0	0	1	0	1	0	0	1	0	0	1	0	1	
						1	e	e	1	e	e	e	e	e	1	e
						1	1	0	0	0	0	1	0	0	1	1

When we apply this checking sequence to the faulty sequential machine M_3 , the same output sequence as that of the correctly operating machine M_2' will be obtained. Nevertheless the machine M_3 is already not the same machine as the original machine M_2' .

From the above example, we can see that to restrict the faults to the types which do not increase the number of states of a machine is not a realistic assumption. In the next section we will consider cases where this restriction will be eliminated.

3.4. Extention of a CC-type Sequential Machine to include Types of Faults Increasing Number of States

If a given sequential machine has $n=2^k$ states then the restriction to consider only types of faults which do not cause an increase in the number of states of the machine is appropriate. In most case in actual practice, the number of states of a given machine is less than 2^k for some k and the assumption that the fault does not increase the number of states is not fully justified as was mentioned in the previous section.

The fact of an increase in the number of states of a machine implies that there exists at least one faulty state transition from state S_i ($i=1,2,\dots,n$) to state S_{n+j} ($j=1,2,\dots,2^k-n$) under input X_u ($u=1,2,\dots,m$) or e .

If there exists such a transition from state S_i to state S_{n+j} in the process of a checking experiment, there must be another state transition from state S_{n+j} to some state S'_i ($i=1,2,\dots,2^k$) under input symbol e . This transition will be examined by L_n or D_k input sequence depending on whether the transition from S_i to S_{n+j} is induced by application of input symbol e or X_u .

If we assign the output entries corresponding to the state S_{n+j} and input symbol e with a new output symbol r , i.e.

$$g(S_{n+j}, e) = r$$

and if we apply an input symbol e when the machine is in state S_{n+j} (both the input symbols of L_n and D_k -sequences are the same e), then an output symbol r will be produced by the machine in response. In other words, if an output symbol r appears in response to L_n or D_k input sequence, it is certain that there is a state transition from S_i to S_{n+j} and there exists a fault which causes an increase in the number of states of the machine.

When a CC-type sequential machine with the above feature operates correctly the output sequence corresponding to the checking sequence is unique when the machine is started in state S_1 . There is no other machine which can produce the same output sequence when supplied with the same checking sequence. The above statement is valid even though the restriction of faults causing an increase in the number of states is abandoned.

Example 3.5: For the sequential machines M_1 and M_2 in Table 2 and Table 5 respectively, the machines with the above feature are obtained as shown in Table 9 and Table 10 respectively. In both the tables, several entries are incompletely specified. Their specification can be based on various economic criteria. But once the design has been completed, all the entries are completely and uniquely specified.

Table 9. The extended machine M_1' of M_1 .

	0	1
S_1	$S_3, 1$	$S_2, 0$
S_2	$S_2, 0$	$S_3, 0$
S_3	$S_1, 1$	$S_1, 1$
S_4	- , -	- , r

Table 10. The extended machine M_2'' of M_2 .

	0	1	e
S_1	$S_3, 0$	$S_3, 1$	$S_2, 0$
S_2	$S_3, 0$	$S_1, 0$	$S_3, 0$
S_3	$S_2, 1$	$S_3, 1$	$S_1, 1$
S_4	- , -	- , -	- , r

Example 3.6: Let us consider the sequential machine M_2'' in Table 10 and assume that $f(S_4, e) = S_2$. When some failure occurs and the failure causes a transition of the form as in Eq. (3.8), then the checking sequence and its associated output sequence are as follows:

e e e e e	0 e	e 0 e	e e 0 e e
0 0 1 0 0 1	0 1	0 0 1	0 0 1 0 1
	1 e	e 1 e e e	e 1 e
	1 1	0 0 r 0 1	0 1 1

The output sequence corresponding to the checking sequence is different from the correct one as given in Eq. (3.6) and also the output symbol r appears in a portion of the output sequence. Thus the machine under test is not the original machine and the number of states is increased because of the occurrence of faults.

So far we have assumed that the output logic corresponding to the input symbol e is perfect. This assumption can be eliminated by applying the error correcting codes to encode the output symbol such that the Hamming distance between two outputs will be larger than $2t+1$, where t is the number of errors.

CHAPTER 4

FAULT TRANSITION FUNCTIONS

In the present study we consider the machine to be a sealed black box and assume that its interconnections are not accessible. By external measurement, i.e. by applying a sequence of inputs to the machine and observing the corresponding sequence of outputs produced, the only information that we can obtain is whether the machine is operating correctly or whether there has occurred some failure that caused some faulty state transitions with corresponding faulty outputs. In the present chapter we shall discuss how we can locate these faulty transitions and spot the faulty outputs in the output sequence produced by the machine in response to the checking sequence.

4.1. Faulty e-transitions

In this section we shall discuss how we can locate the faulty transitions under input symbol e by observing the output sequences produced by extended CC-type sequential machines in response to L_n input sequences of a checking sequence constructed by the procedure of the previous chapter. Since the same faulty output sequence corresponding to an L_n input sequence may be obtained from several different faulty machines (each machine has at least one faulty state transition which is different from that of other faulty machines), we shall first try to find all possible output sequences in response to an L_n input sequence and then will classify them into several classes. Then from these output sequences, we shall try to find the possible faulty transition functions.

As was pointed out previously in subsection 3.1.3, even though the e-transitions, which are the transitions induced by applying input symbol e from state S_i ($i=1, 2, \dots, n$), are correct, the machine does not satisfy the L_n -sequence unless the initial state is S_1 .

During the checking experiment, in order to set the machine in state S_1 , we apply input symbol e to the machine successively until the output symbol l appears in the output terminal. That is,

$$\begin{array}{ccccccc} e & e & e & \dots & e & e & e \\ z & z & z & \dots & z & z & l \end{array} \quad (4.1)$$

where z represents an arbitrary output symbol except l. For a machine with n states, the number of such input symbols e that may be needed for the purpose is one or more, but not exceeding n. Since the original machine has only n states, if we apply n consecutive input symbol e's and the output symbol l does not appear in output terminal, we can conclude that the L_n -sequence is not satisfied. Therefore, the length of the consecutive input symbol e is not necessarily more than n.

Corresponding to the input symbol e, there are three different output symbols, from which $\sum_{i=1}^3 \binom{3}{i} = 7$ different combinations can be formed as shown in Table 4.1. From each combination of output symbols, we can easily construct output sequences which satisfy Eq. (4.1) or contain n output symbols 0 or/and r as shown in Table 4.2. In Table 4.2, the number of output sequences which can be constructed is also shown.

Table 4.1. Different combinations of the output symbols.

	output symbols
1	1
2	0 , 1
3	r , 1
4	0 , r , 1
5	0
6	r
7	0 , r

Table 4.2. All possible output sequences corresponding to a maximum of n consecutive input symbol e.

class	output sequences	number of output sequences
1.1	{ 1 }	1
1.2	{ 0 ^u 1 : 1 ≤ u ≤ n-1 }	n-1
1.3	{ r ^v 1 : 1 ≤ v ≤ 2 ^{k-n} }	2 ^{k-n}
1.4	{ z 1 : z ∈ (0+r) ^w - 0 ^{w-r} - r ^w 2 ≤ w ≤ n-1, 1 ≤ i ≤ 2 ^{k-n} i = # (r) in z , i < w }	$\sum_{w=2}^{n-1} \sum_{i=1}^{2^{k-n}} \binom{w}{i}$
1.5	{ 0 ⁿ }	1
1.6	{ r ⁿ }	1
1.7	{ z : z ∈ (0+r) ⁿ - 0 ^{n-r} - r ⁿ 1 ≤ i ≤ 2 ^{k-n} , i = # (r) in z }	$\sum_{i=1}^{2^{k-n}} \binom{n}{i}$

In the above table, # (r) in z means the number of symbols r in the sequence z. Thus if z = 00r0rr0, then # (r) in z is 3.

For a machine with $n=2^k$ states if there are only two output symbols corresponding to the input symbol e , then the combinations of output symbols is $\sum_{i=1}^2 \binom{2}{i}=3$ and the possible output sequences are those of class 1.1, 1.2, and 1.5. In such a case, the number of possible output sequences is $n+1$ (the sum of the numbers of output sequences of classes 1.1, 1.2, 1.5.).

From Table 4.2, it is seen that the sequences of class 1.1 through class 1.4 satisfy the input-output sequence of Eq.(4.1). By which we can assume that the machine is in state S_1 , though the machine may be in another state S_i ($i=2,3,\dots,2^k$) when there occurs a faulty state transition such that $f(S_n, e)=S_i$. The output sequences of class 1.5 through class 1.7 do not satisfy the input-output sequence of Eq.(4.1), so we can conclude that there occurs some fault and the machine may be in any state S_j ($j=1,2,\dots,2^k$). Even though we still assume that the machine is in state S_1 . So that we can perform the next step to check whether or not e -transitions are correct.

By applying the L_n input sequence to a machine, after the machine is assumed to be in state S_1 and observing its associated output sequence we can determine whether the properties of Definition 3.1.1 are satisfied, i.e. whether e -transitions are the same as it was originally designed. If L_n -sequence is satisfied, thus the input-output sequence is that of Eq. (4.2).

$$\begin{array}{lcl}
 \text{length of the seq. :} & \leftarrow n-1 \longrightarrow & \longleftarrow n-1 \longrightarrow \\
 \text{input :} & e e \dots e e e e e \dots e e & (4.2) \\
 \text{output :} & 0 0 \dots 0 0 1 0 0 \dots 0 0 &
 \end{array}$$

Then e-transitions are correct; otherwise, some fault occurs in the machine and a faulty transition function can be determined from the output sequence.

In response to the L_n input sequence, the all possible output sequences can be divided into eight classes, namely the sequences of seven combinations of three output symbols and a correct sequence, as shown in Table 4.3. For a machine with $n=2^k$ states the possible output sequences are those of class 2.1, 2.2, 2.3 and 2.5.

For e-transitions only, we can consider the machine as a submachine with one input symbol of a given machine. We know that every component of the transition diagram of such a submachine must contain at least a single cycle. From any one of output sequences in Table 4.3, a single cycle can be formed. The set of states and the number of states contained for each class are summarized in Table 4.4.

To determine which set of states exactly form a cycle we need to modify the given machine and to encode the output symbols corresponding to the input symbol e as follows:

$$\begin{aligned}
 g(S_i, e) &= Z_i, \\
 g(S_j, e) &= Z_j;
 \end{aligned}$$

where $i, j=1, 2, \dots, 2^k, i \neq j$ and $Z_i \neq Z_j$. When a machine is designed

in this manner, the length of a checking sequence can be shortened, since the

Table 4.3. All possible output sequences corresponding to L_n .

class	output sequence	number of sequences
2.1	$\{ 0^{n-1} 1 0^{n-1} \}$	1
2.2	$\{ 1^{2n-1} \}$	1
2.3	$\{ 0^{2n-1} \}$	1
2.4	$\{ r^{2n-1} \}$	1
2.5	$\{ (0^u 1)^v 0^w : 1 \leq u \leq n-2, w \leq u, (u+1)v+w=2n-1 \}$	$n-2$
2.6	$\{ (r^u 1)^v r^w : 1 \leq u \leq 2^k-n, w \leq u, (u+1)v+w=2n-1 \}$	2^k-n
2.7	$\{ w^v z : w \in (0+r)^u - r^u - 0^u, 2 \leq u \leq 2^k-1, \}$ $1 \leq i \leq n-1, i = \#(0) \text{ in } w, z < u,$ $1 \leq j \leq 2^k-n, j = \#(r) \text{ in } w, uv+ z =2n-1. \}$	$2^k-n \sum_{j-1}^{j+n-1} \sum_{u=j+1}^u \binom{u}{j}$
2.8	$\{ (w1)^v z : w, i, j \text{ and } u \text{ are the same as those in 2.7.} \}$ $2 \leq u \leq 2^k, z \leq u, (u+1)v+ z =2n-1. \}$	same as 2.7.

In the table z is the preceding subsequence of the sequence w . Thus if

$n=6$ and $w=00r$, then $2n-1=11$, $u=3, v=3$ and $z=00$, therefore, $w^v z=00r00r00r00$,

and $|z|$ is the length of the sequence z .

Table 4.4. Set of states which make up a cycle.

class	set of states	number of states
2.1	$\{ S_i \}$ and $\{ S_n \}$	n
2.2	$\{ S_n \}$	1
2.3	$\{ S_i \}$	1 or more up to $n-1$
2.4	$\{ S_{n+j} \}$	1 or more up to 2^k-n
2.5	$\{ S_i \}$ and $\{ S_n \}$	$u+1$
2.6	$\{ S_{n+j} \}$ and $\{ S_n \}$	$u+1$
2.7	$\{ S_i \}$ and $\{ S_{n+j} \}$	$u^q, 2 \leq q \leq 2^k-1$
2.8	$\{ S_i \}, \{ S_{n+j} \}$ and $\{ S_n \}$	$u+1$

length of the distinguishing sequence is one: but the procedure of constructing a checking sequence is no longer systematic.

The output sequences in Table 4.3 except the class 2.1 are different from Eq. (4.2). If one of the output sequences except the class 2.1 is produced by a machine in response to the L_n input sequence, then some fault occurs within the machine and there is at least one state transition which is different from that of the flow table of the given extended CC-type sequential machine as long as the output logic corresponding to input symbol e is perfect.

These fault state transitions corresponding to the output sequences in Table 4.3 are listed in Table 4.5.

Table 4.5. Faulty state transitions.

class	function	correct	fault	remarks
2.2	$f(S_n, e)$	S_1	S_n	
2.3	$f(S_i, e)$	S_{i+1}	S_{i-a}	$1 \leq i \leq n-1$ $0 \leq a < i$
2.4	$f(S_i, X_u)$	S_i'	S_{n+j}	$f(S_{n+j}', e) = S_{n+j}''$
2.5	$f(S_i, e)$ or $f(S_n, e)$	S_{i+1} S_1	S_{i+b} S_b	$1 \leq i \leq n-1$ $2 \leq b \leq n-i$ $2 \leq c \leq n-1$
2.6	$f(S_n, e)$	S_i	S_{n+j}	$f(S_{n+j}', e) = S_n$
2.7	$f(S_i, e)$	S_{i+1}	S_{n+j}	$f(S_{n+j}', e) = S_i', 1 \leq i \leq n-1$
2.8	$f(S_i, e)$ or $f(S_n, e)$	S_{i+1} S_1	S_{n+j} S_{n+j}	$f(S_{n+j}', e) = S_d', 1 \leq d \leq n$ $f(S_{n+j}', e) = S_i', 1 \leq i \leq n-1$

eeee0e1ee elee eeeee eele eeee eeele eeeee eeee (4.4)
000000 100 0010000 1010000 10010 001010000

From the above output sequence we know that the subsequences after $W_{5,0}$ -sequence, i.e. the sequence to check the transition from state S_5 under input 0, is different from the correct one. Thus the transition from state S_5 under input 0 is not to state S_5 as it was originally designed, but to some other state we do not know; moreover there is no more information about other transitions following $W_{5,0}$ -sequence.

For the purpose of obtaining more information about the transitions of faulty CC-type sequential machines, we use the adaptive experiment, i.e. an experiment in which the decision as to what input subsequence is to apply next is based upon the information obtained from the responses of the preceding sequences. Thus the input sequence generator should in this case be capable of generating a number of sequences based on information fed back to its input from the output terminal of the machine. A schematic representation of the adaptive experiment set up is shown in Fig. 4.1.

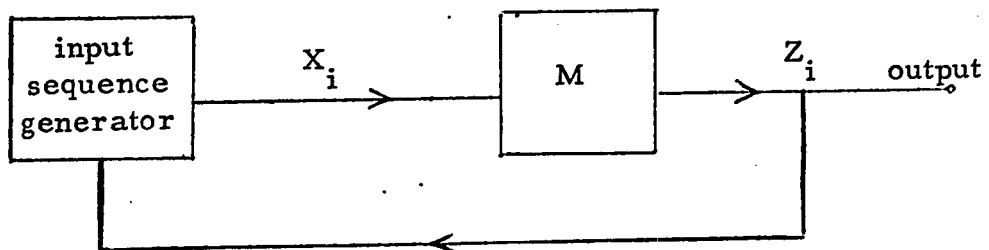


Fig. 4.1. Adaptive experiment.

Table 4.6. An extended sequential machine.

	0	1	e
s_1	$s_3, 1$	$s_4, 1$	$s_2, 0$
s_2	$s_2, 0$	$s_1, 1$	$s_3, 0$
s_3	$s_5, 0$	$s_2, 1$	$s_4, 0$
s_4	$s_1, 1$	$s_5, 0$	$s_5, 0$
s_5	$s_5, 0$	$s_2, 0$	$s_1, 1$
s_6	-, -	-, -	-, r
s_7	-, -	-, -	-, r
s_8	-, -	-, -	-, r

Before executing this experiment, the L_n -sequence should be satisfied, which means that the response of L_n -input sequence is of class 2.1 of Table 4.3. Since T_i or D_i -sequences which will be used in this experiment give us the information of the final state or initial state of a sequential machine and both the T_i and D_i -sequences are dependent on the correctness of e-transitions so if the L_n -sequence is not satisfied, the e-transitions are not correct then the T_i or D_i -sequences could not give us the expected information and the experiment is over; therefore we should first check if the L_n -sequence is satisfied. If the L_n -sequence is satisfied, then the final state of the machine is S_n . In order to use the T_i -sequences to have the machine in a desired state S_i , we need to bring the machine first to state S_1 by applying another input symbol e to the machine whose associated output should be 1.

To determine to which state the transition occurs from S_i under input X_u , we first apply the T_i input sequence to the machine. Since L_n -sequence is satisfied, the machine will change from state S_1 to state S_i on applying T_i -sequence whose associated outputs will be all 0's. Next, we apply the input X_u . If the associated output is Z_j which differs from the correct output X_i , as can be seen from the given flow table, we know that the faulty output function is $g(S_i, X_u) = Z_j$. Then, we apply the input symbol e successively until an output symbol of 1 appears. If, after $2^k - n + 1$ consecutive symbols e being applied

at the input, the symbol 1 does not appear at the output, we next check if the output in response to the $(2^k - n + 1)$ th input e is r . If it is r , we know that there is a transition from state S_i to state S_{n+j} and a cycle exists in the state set S_{n+j} corresponding to the input e (class 2.4 in Table 4.4). If we apply inputs e again the machine will simply recycle and 1 will never appear, so that no more information will be available and thus the experiment is over. If the output in response to the $(2^k - n + 1)$ th input e is 1, we go to the next step but if it is 0, the input symbol e will be applied successively until an output symbol of 1 appears. The response to the consecutive applications of the input symbol e after the symbol X_u should be belonging to the five output sequence groups as shown in Table 4.7.

Table 4.7. Output sequences corresponding to the consecutive applications of the input symbol e after the symbol X_u .

group	output sequences
1	1
2	$0^x 1$
3	$r^y 1$
4	$r^y 0^x 1$
5	$r^{2^k - n + 1}$

$$x=1, 2, \dots, n-1$$

$$y=1, 2, \dots, 2^k - n$$

The length of the output sequence is less than or equal to 2^k .

The transitions from state S_i under the input X_u are shown in Table

4.8. In this table we also show another transition function $f(S_{n+j'}, e)$ which must exist in the faulty machine.

Table 4.8. Next states of S_i .

group	$f(S_i, X_u)$	$f(S_{n+j'}, e)$
1	S_n	
2	S_{n-x}	
3	S_{n+j}	S_n
4	S_{n+j}	S_{n-x}
5	S_{n+j}	$S_{n+j''}$

In the table j, j' and $j'' \in \{1, 2, \dots, 2^k - n\}$ and x is the same as that of the same group in Table 4.7.

If the output sequence corresponding to the consecutive input symbols e is not a sequence of the group 5 shown in Table 4.7, the output symbol 1 will appear in the output terminal and the machine will finally be in state S_1 , so that we can go on to check the possible transitions for all other i and u ($i=1, 2, \dots, n$, $u=1, 2, \dots, m$).

We can now summarize the procedure as follows:

Procedure

Step 1: Apply the input symbol e repeatedly (at most n times) to the machine until the output symbol 1 appears in the output terminal. If 1 does not appear, go to the next step.

Step 2: Apply L_n input sequence to the machine and check to which class of Table 4.3 the associated output sequence belongs. If it

belongs to class 2.1, go to next step; otherwise, the faulty transition function will belong to the corresponding class of Table 4.5, and the experiment is over.

Step 3: To determine to which state the transition from S_i will be under the input X_u , the following steps are taken.

(a) Apply T_i -sequence.

(b) Apply the input X_u next and check whether the associated output Z_i is the same as the one described in the given flow table.

If they are not the same, the faulty output function is $g(S_i, X_u) = Z_i$.

(c) Apply the input symbol e successively up to $2^k - n + 1$ times until the output symbol 1 appears in the output terminal. If a output symbol of 1 does not appear and the $(2^k - n + 1)$ th output symbol is r , then no more information is available and the experiment is over.

Since L_n -sequence has been checked in step 2, there is no transition from state S_i to state S_{n+j} . If $(2^k - n + 1)$ th output symbol is r , then there should be $2^k - n + 1$ consecutive r in the output sequence, i.e.

a cycle is formed by some states of the set of states $\{S_{n+j}\}$.

If the $(2^k - n + 1)$ th output is 0, continue to apply the input symbol e successively until a 1 appears in the output terminal.

(d) Check to which group in Table 4.7 the associated output sequence after the application of the input X_u belongs to determine the state from the corresponding group in Table 4.8. If the state is different from the one described in the given flow table, then

belongs to class 2.1, go to next step; otherwise, the faulty transition function will belong to the corresponding class of Table 4.5, and the experiment is over.

Step 3: To determine to which state the transition from S_i will be under the input X_u , the following steps are taken.

(a) Apply T_i -sequence.

(b) Apply the input X_u next and check whether the associated output Z_i is the same as the one described in the given flow table.

If they are not the same, the faulty output function is $g(S_i, X_u) = Z_i$.

(c) Apply the input symbol e successively up to $2^k - n + 1$ times until the output symbol 1 appears in the output terminal. If a output symbol of 1 does not appear and the $(2^k - n + 1)$ th output symbol is r , then no more information is available and the experiment is over.

Since L_n -sequence has been checked in step 2, there is no transition from state S_i to state S_{n+j} . If $(2^k - n + 1)$ th output symbol is r , then there should be $2^k - n + 1$ consecutive r in the output sequence, i.e.

a cycle is formed by some states of the set of states $\{S_{n+j}\}$.

If the $(2^k - n + 1)$ th output is 0, continue to apply the input symbol e successively until a 1 appears in the output terminal.

(d) Check to which group in Table 4.7 the associated output sequence after the application of the input X_u belongs to determine the state from the corresponding group in Table 4.8. If the state is different from the one described in the given flow table, then

faulty state transition function can be found.

(e) If the output symbol 1 appears in the output terminal in step (c), repeat steps (a), (b), (c) and (d) for other possible state transitions corresponding to every i and u ($i=1,2,\dots,n$; $u=1,2,\dots,m$).

Example : Consider a sequential machine whose original flow table is shown in Table 4.6. Then the checking sequence and its correct output sequence are shown in Eq. (4.3). Now we perform an adaptive experiment following the procedure mentioned above and obtain the following input-output sequences.

$$\begin{array}{cccccc}
 U_5 & W_{1,0} & W_{2,0} & W_{3,0} & W_{4,0} & \\
 eeeeeeeeee & 0eeee & e0eeee & ee0e & eee0eeee & \\
 0000100001 & 1r001 & 01r001 & 0001 & 000100001 & \\
 \\
 W_{5,0} & W_{1,1} & W_{2,1} & W_{3,1} & W_{4,1} & (4.5) \\
 eeee0eeee & leeeee & eleeee & eeleeee & eeleeeee & \\
 0000000001 & 1rr001 & 01001 & 0010001 & 0000rrrr &
 \end{array}$$

In the above input-output sequences, $W_{1,0}$, $W_{2,0}$, $W_{5,0}$, $W_{1,1}$, $W_{2,1}$ and $W_{4,1}$ -sequences are different from those in Eq. (4.3). From Tables 4.7 and 4.8 we obtain faulty and other state transition functions and a faulty output function as shown in Table 4.9.

From the fact that the output sequences after the input X_u of $W_{1,0}$, $W_{1,1}$ and $W_{4,1}$ -sequences are different from each other and $2^k - n = 3$, we can conclude that $S_{n+j_1} \neq S_{n+j_2} \neq S_{n+j_3} \neq S_{n+j_4}$ and $S_{n+j_3} = S_{n+j_4}$; thus $f(S_{n+j_3}, e) = S_{n+j_3}$.

Table 4.9. Faulty transition functions.

sequence	group	$f(S_i, X_u)$	$f(S_{n+j}, e)$	$g(S_i, X_u)$
$W_{1,0}$	4	S_{n+j_1}	S_3	
$W_{2,0}$	4	$S_{n+j'}$	S_3	1
$W_{5,0}$	2	S_1		
$W_{1,1}$	4	S_{n+j_2}	S_3	
$W_{2,1}$	2	S_3		
$W_{4,1}$	5	S_{n+j_3}	S_{n+j_4}	

CHAPTER 5

CONCLUDING REMARKS

In the present dissertation, we have discussed extension of a CC-type sequential machine by the addition of an output symbol in order not to restrict the faults only to the type which does not cause an increase in the number of states of the given machine. It is shown that the checking sequence by such an extended machine is the same as that of the original machine.

An adaptive experiment is developed for the detection of faulty state transitions. All possible output sequences that may occur corresponding to the checking sequences are classified in several groups from which faulty state transition functions can be readily found.

If e-transition is correct and no cycle exists in the state set $\{S_{n+j}\}$, all faulty state transition functions can be determined and a faulty CC-type sequential machine can be identified.

It may be pointed out finally that the introduction of additional output logic is not very costly and can thus be incorporated quite readily. The additional advantages are that the usual restriction of faults not increasing the number of states can be readily done away with, with the resulting checking experiment being very efficient and practical.

REFERENCES

- (1) E. F. Moore, 'Gedanken-experiments of Sequential Machines', Automata Studies. Princeton, N. J., Princeton University Press, 1956, p. 129-153.
- (2) S. Ginsburg, 'On the Length of the Smallest Uniform Experiment which Distinguishes the Terminal States of a Machine!', J. Assoc. Comp. Mach., 5, 266-280 (July 1958).
- (3) T. N. Hibbard, 'Least Upper Bounds on Minimal Terminal State Experiments for Two Classes of Sequential Machines', J. Assoc. Comp. Mach., 8, 601-612 (Oct. 1961).
- (4) A. Gill, 'Introduction to the Theory of Finite state Machines, McGraw-Hill Book Co., New York, 1962.
- (5) S. Seshu and D. N. Freeman, 'The Diagnosis of Asynchronous Sequential Switching Systems', IRE Trans. EC-11 (Aug. 1962).
- (6) Kozo Kinoshita, 'Some Considerations on the Fault Diagnosis of Sequential Circuits', Jour. I.E.C.E., Japan, 46, 9, p 1211 (Sept 1963).
- (7) Shin-ichi Murakami, Kozo Kinoshita and Hiroshi Ozaki. 'Computer Experiments for Finding Fault-detecting Sequences of Sequential Circuits', Jour. I. E. C. E., Japan, 50.10, p.1925 (Oct. 1967).
- (8) F. C. Hennie, 'Fault detecting Experiments for Sequential Circuits', Proc. of 5th Annual Symp. on Switching Circuit Theory and Logical Design', Princeton, N.J., p 95 (Oct. 1964).

- (9) Z. Kohavi and P. Lavallee, 'Design of Sequential Machines with Fault-detection Capabilities', IEEE Trans. Electronic Computers, Vol. EC-16, p. 473-484, August 1967.
- (10) S. Murakami, K. Kinoshita and H. Ozaki, 'Sequential Machines Capable of Fault Diagnosis', IEEE Trans. Electronic Computers, Vol. EC-19, p. 1079-1085, Nov. 1970.
- (11) S. R. Das and C.L. Sheng, 'Identification of Synchronous Sequential Machines by Merging of States', 21st Annual Southwestern IEEE Conference, Texas, U.S.A., (1969 SWIEECO Record, IEEE Catalog No. 69C16-SWIECO, p. 4F1-4F8).
- (12) I. Kohavi and Z. Kohavi, 'Variable Length Distinguishing Sequences and Their Application to the Design of Fault-Detection Experiments', IEEE Trans. Computers, Vol. C-17, 1968, p.792-795.
- (13) S.R. Das and C.L. Sheng, 'On the Paths of a Strongly Connected Machine', Dept. of Electrical Engineering, University of Ottawa, Technical Report No. 69-7, Aug. 1969.
- (14) S.R. Das and C.L. Sheng, 'Some Further Studies on Machine Identification by Merging of States', Trans. of the Engineering Institute of Canada, Vol. 13, No. C-6, July 1970, p. 1-7.
- (15) T.J. Powell, 'Synthesis Requirements for Fault Detection', Proc. 6th Allerton Conference on Circuit and System Theory, 1968, p.761-772.

- (16) S. R. Das and C. L. Sheng, 'On the Identification of Synchronous Sequential Machines', Paper for IFAC Kyoto Symposium, 1970, p. 581-586.

V I T A

NAME: Cheng San Chiang

BORN: Nov. 26, 1933, Taiwan, China.

EDUCATION:

Secondary: Taichung High School.

University: Tokai University .

Tokyo, Japan.

1969, B.E. in E.E.