Fault detection on sequential machines

Chung-Tao David Wang

FAULT DETECTION ON SEQUENTIAL MACHINES

By

CHUNG-TAO DAVID WANG, 1943-

A

THESIS

submitted to the faculty of

UNIVERSITY OF MISSOURI-ROLLA

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

Rolla, Missouri

1970

Approved by

S. H. Szygenda (advisor) James H. Tracey

C. Y. Ho

ABSTRACT

This paper presents an algorithm for deriving an optimum test sequence for detecting faults in a synchronous machine. In this study, the flow table is used as a tool to generate the fault detection tests. The fault stuck-at-1 (or stuck-at-0) is said to be present when a permanent signal valued 1 (or 0) appears on a component of the machine. Only single faults are treated. The result of the procedure is one or more test sequences guaranteed to detect a set of faults (F_p).

First, sequential machines with feedback lines as memory elements are considered. Then the memory elements are changed to R-S flip-flops. Finally, several suggestions for further work are made.

ACKNOWLEDGEMENT

The author wishes to take this opportunity to express his deep appreciation to his advisor, Dr. Stephen A. Szygenda, for his guidance and helpful advice.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table Page

CHAPTER I

INTRODUCTION

The diagnosis of equipment failures is of great importance in the field of digital system maintenance. In order to maintain digital systems, first it is necessary to detect any possible failure in the equipment and then, to isolate the failure to a repairable subsystem. The goal of this paper is to generate test sequences to detect faults, if any exist, in a sequential machine. The synchronous sequential machine is considered. Machines with different memory elements, such as feedback loops and flip-flops, are investigated separately.

For fault location in combinational logic, Eldred[4] and Armstrong[5] have developed a technique to derive tests by the use of logical diagrams of the system. These techniques are called path sensitizing and equivalent normal form, respectively. Details of these techniques will be given in Chapter III. Roth[8,9] expanded this work with the "Calculus of D-Cubes" to find tests for combinational logic nets. Poage[1] constructed fault tables as a means of performing a special kind of network analysis which is ideally suited for fault detection purposes. Using this technique, tests with minimal length can be found as the solution of the fault table. He also developed a method to detect multiple faults by adding extra columns to the single fault table.

Poage's later work[2,3] treats the problem of sequential machines in a manner analogous to that devised for combinational logic nets. The procedure for combinational and sequential nets can be briefly described as follows:

(A) For the generation of fault detection tests for a combinational logic net, the following procedure is followed.

(1) Perform network analysis; generate output proposition in terms of input propositions and element propositions.

(2) Compute network functions in the presence of faults; find each faulty output proposition.

(3) Construct the single fault table column by column. The fault table is formed by assigning each possible fault in (F_p) as an element in a column, each input combination of (X) as an element in a row, and a cross in row i and column j indicates that the input combination of row i will detect the single fault of column j. Where (F_p) and (X) are defined as a set of faults to be

detected, and a set of fault-detection tests, respectively.

(4) Use column dominance technique to simplify the single fault table.

(5) Add additional columns to the single fault table in order to construct the multiple fault table.

(6) Find the optimum test by performing the row dominance technique.

(B) For a fault detection test sequence of a <u>sequential machine</u>, the procedure is as follows:

(1) Perform network analysis; generate excitation propositions and output propositions in terms of input propositions and element propositions.

(2) Tabulate each fault flow table and one fault-free flow table; label these T_1, T_2,..., T_m and T_0.

(3) In construction of the sequential fault table, sets of sequences which detect each fault are found by forming the product of each fault table with T_0. Then combine all the m product tables into one final sequential fault table. Here the individual product is analogous to the single column of the combinational fault table.

(4) Use column dominance techniques to simplify the sequential fault table.

(5) Expand the table into a sequential fault tree where the optimum test sequence for the set (F_p) is found.

Several other approaches to the test generation for sequential machines have appeared. Galey, Norby and Roth derived tests by logically cutting all the feedback loops so that the circuit would have no cycles. This method, for diagnosis, proceeds as if these cuts were actually made. Seshu[6,7] determined first how a machine could be transformed by a fault. Then he developed four strategies for finding a next test sequence which will provide the maximum amount of information. The information serves the purpose of further partitioning of the faulty machines. In other words, an attempt is made to find a test which would be able to detect or diagnose the greatest number of faults.

This study is an extension of Poage's work. Basically, the author tries to treat the problem of finding faults which may exist in the reset circuitary[3]. Extensions of Poage's techniques to find faults in R-S flip-flop circuits are also considered. Chapter II will present the required background of sequential machines, faults, tests, and the basic concept of how faults can be detected by tests. In the last section, an example is given to demonstrate the above concepts.

CHAPTER II

BACKGROUND

A. Sequential Switching System

A sequential switching circuit is one whose output values, at a given time, depend not only on the present inputs, but also on inputs applied previously. A general model was first introduced by Huffman[12]. Huffman's system is defined as a combinational network C with some of the outputs being fed back to some of the inputs through unit delays. It can be depicted as shown in Figure 2.1.

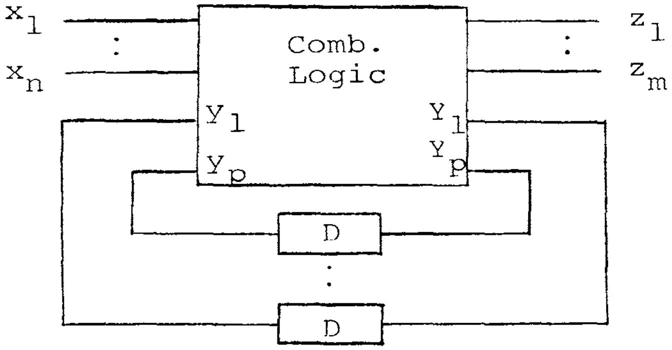
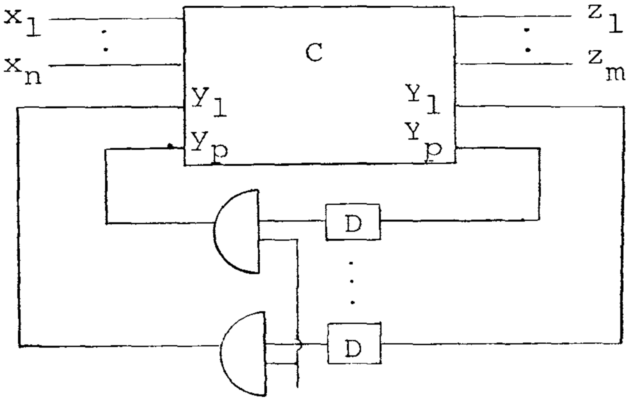


Figure 2.1. Huffman Model of Sequential Circuit

The symbols x and y are used for primary inputs and feedback inputs to the combinational logic C; Y and z are used for feedback outputs and primary outputs, respectively. The history of previous inputs is summarized in the states of the circuit.

B. Synchronous Switching Circuit

A model of a synchronous switching circuit was introduced by Mealy, and is shown in Figure 2.2. The clock pulses occuring at certain times govern the performance of the system. The primary outputs and feedback outputs at time t are determined by the primary and feedback inputs at t. Then the present feedback output, y_{t+1}, will in turn become the feedback input vector y for the next time instant t + 1. This model is used throughout the study.



Clock pulses governing the circuit

Figure 2.2. Synchronous Sequential Circuit

C. Fault

A fault, in general, is a physical defect of one or more components which can cause the circuit to malfunction. The faults considered in this study are restricted by the following assumptions:

(1) Only one fault can occur at a time, i.e., single fault assumption. This implies that a malfunction is detected, located and repaired before the next fault occurs.

(2) Only logical faults are investigated, and the types of faults are known. The fault may be defined as a transformation of a good (or fault-free) machine into one of the different possible, known faulty machines. If there are m possible faulty machines, there are m + 1 machines which should be taken into consideration in finding tests to detect all faults.

(3) It is possible to momentarily reset the feedback lines to a known initial state even under failure conditions.

D. Test

When a particular input is applied to a faulty machine, the resulting output may differ from that of the good machine. If this occurs, it is said that this input or test can detect the fault. For a sequential machine, a sequence of input vectors should be applied in order to detect the fault.

Moreover, the different possible outputs of a machine for a particular input can partition the machines into various groups. Hence, successive inputs, applied to the system, may identify the individual fault, or isolate the fault to the lowest degree. At that stage, the fault is said to be diagnosed.

E. Summary and Example

The purpose of this study is to derive a set of optimum test sequences (X_{op}) to detect a set of faults (F_p) in a given synchronous sequential machine. The optimum test sequence is defined as the shortest sequence of symbols guaranteed to detect a set of m faults in (F_p).

The following notation will be used in this paper:

(F): the set of all possible faults

(F_p): a subset of (F) selected for particular reasons

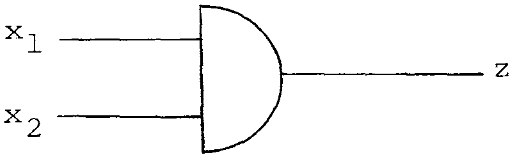
(X): the set of all possible input combinations

(X_{op}): the set of optimum test sequences

The reasons for choosing subsets of (F) is primarily for the purpose of simplifying the procedure of deriving the test sequences. In general, Poage's technique becomes cumbersome and requires a columinous table to find the solution, when all the possible faults are considered at one time.

An example is given to show the concept of test for a combinational circuit.

Example 2.1. Find the test for a s-a-1 fault on an input of an AND gate.



(a)

x_1	x_2	z
0	0	0
0	1	0
1	0	0
1	1	1

(b)

x_1	x_2	z
1	0	0
1	1	1
1	0	0
1	1	1

Table 2.1 (a) A Fault-Free Truth Table

(b) A Fault Truth Table with s-a-1 Fault on x_1

The input combination $\overline{x} = (0,1)$ can detect the fault (x_1, s-a-1), since the outputs are different in the two truth tables.

Chapter III of this study is devoted to the derivation of the optimum test sequences for synchronous sequential machine together with more extensive examples.

CHAPTER III

DERIVATION OF OPTIMUM TEST SEQUENCES

A. Network Analysis

The reason for performing network analysis before deriving test sequences is primarily to show how the fault may affect the operation of the machine. Unfortunately, ordinary Boolean algebra can only describe the normal operation of the switching circuits, but not the operation in the presence of the faults. The result of the network analysis is an expression showing the network function as well as indicating all possible faults. After a simplification procedure, the required tests can be found.

The general strategy for finding a set of fault detection tests, for a class of faults, has been found to start with network analysis. For comparison, Armstrong's Path Sensitizing* and Equivalent Normal Form (enf)[5] techniques are introduced here.

Definition of a Path

A path from G_0 to G_n is defined as a physical connection traced from one primary input to a possible output as shown in Figure 3.1. For example, 'abcde' is considered as a path from input x_1 to output z.

*The Path Sensitizing concept was suggested at Conference on Diagnosis of Failure in Switching Circuit; May 1961.

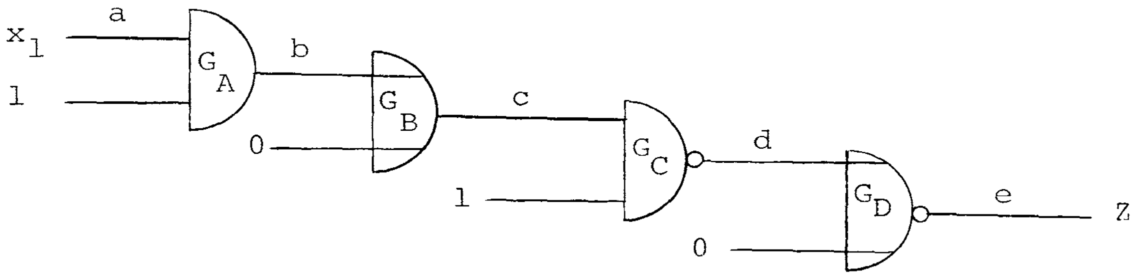


Figure 3.1. A Path

In order to detect a s-a-1 fault on wire a (short for a_1), several initial values should be assigned as shown. This is called the normal condition. Now the change to 1 on wire "a" propagates the faulty signal all the way to z. Under this condition, the path 'abcde' is said to be sensitized, and the fault a_1 is said to be detected. Note that not only a_1 can be detected, but s-a-1's on b, c, and e, as well as, s-a-0 on d can also be detected. By applying the path sensitizing technique, an enf for the net is formed. This is demonstrated by the following example. First, associate with each wire a name, a, b, c, etc., and with each gate a G_1, G_2, G_3, etc. Then level the gates with numbers as shown in Figure 3.2. The enf is found starting with level number 1. The resulting enf is a "sum of products" expression. Each literal consists of an input variable subscripted by a sequence of gate numbers. The variable and its subscript sequence identifies a path from an input to an output.

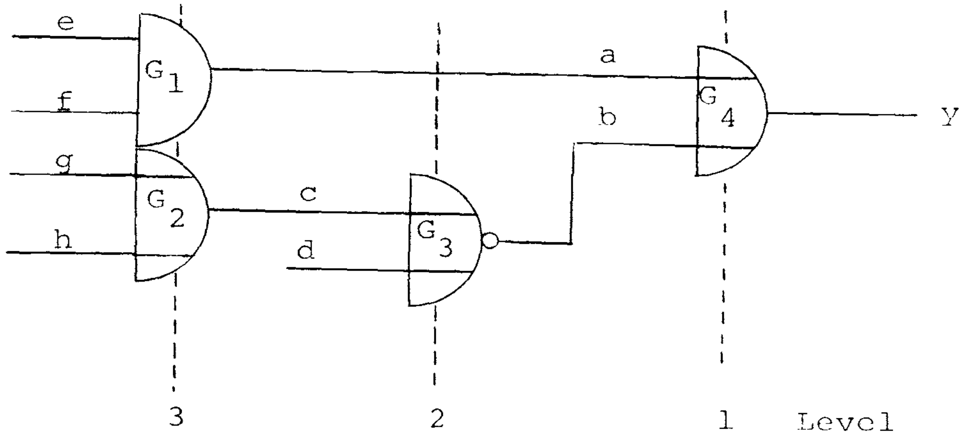


Figure 3.2. An Illustrative Network for enf

$$
\begin{aligned}
y &= (a + b)_4 \\
&= ((ef)_1 + (c'd')_3)_4 \\
&= e_{14}f_{14} + (g'_{234}\ h'_{234})\ d'_{34} \\
&= e_{14}f_{14} + g'_{234}h'_{234}d'_{34} \quad \ldots\ldots\ldots \quad (3.1)
\end{aligned}
$$

It has been shown that the enf provides a tool to construct fault tables[5]. Systematically finding the most desirable test could be accomplished from the fault tables. Desirable tests are those which can sensitize many paths so as to detect many faults.

The network analysis performed in this study is introduced as follows.

Definition of Propositions

1. Input propositions

Associate with each input line, X_i, two input propositions:

x_i: For the signal applied to input line X_i as a value of 1.

x_i': For the signal applied to input line X_i as a value of 0.

2. Element propositions

The elements of the gate network are the wires connecting two gates, primary inputs, outputs, or feedback lines. Elements of the system have the property that they are binary in nature. Hence, any element, a, may appear in any of following states:

(a) The wire a is normal; i.e., its signal may be switched to either value 1 or 0. The element proposition associated with it is a_n.

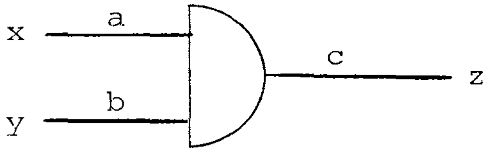
(b) The wire a is stuck-at-one; i.e., a signal of constant value 1 appears on the wire. The element proposition is a_1.

(c) The wire is stuck-at-zero; i.e., a signal of constant value 0 appears on the wire. The element proposition in this case is a_0.

In summary, any element, a, can have three propositions, namely, a_n, a_1, and a_0.

3. The output propositions

The output propositions are expressed in terms of input and element propositions. The output proposition can describe the signal on the wire and the circuit performance of the machine. For example, consider a single AND gate.



The proposition for wires a, b, and c are x, y, and xy, respectively. Further, the proposition which can describe the logic operation in the presence of faults are given as follows:

For convenience, the general propositions will be defined first.

$P_a = x\ a_n + a_1$ to express a signal of value 1 appearing on wire a.

$P'_a = x'a_n + a_0$ to express a signal of value 0 appearing on wire a.

The output propositions of the gate are,

$$P_c = (P_a P_b)c_n + c_1$$
$$= (P_a c_n + c_1)\ (P_b c_n + c_1)$$

A compressed form of notation for the proposition is, for convenience, to drop c_n and have P_c written as follows.

$$P_c = P_a(c_1) \cdot P_b(c_1) \quad \ldots\ldots\ldots\ldots\ldots \quad (3.2)$$

Similarly,

$$\begin{aligned} P'_c &= (P_aP_b)'c_n + c_0 \\ &= (P'_a\ c_n + c_0)\ (P'_b\ c_n + c_0) \\ &= P'_a(c_0) \cdot P'_b(c_0) \quad \ldots\ldots\ldots\ldots\ldots \quad (3.3) \end{aligned}$$

Note that several theorems such as

$(X + Y)Z = XZ + YZ$,

$XY + Z = (X + Z)\ (Y + Z)$,

$X = X + X$,

$XX = X$, etc. have been applied to obtain a standard sum of products form. Further, for convenience, the final results are condensed into their compressed notation form. For a more complicated circuit, the analysis procedure is done backwards, from the output level to the primary input level. At each level, the above theorem and compressed notation technique must be applied to avoid a very combersome standard form. The above definitions and techniques are demonstrated by Example 3.1.

Example 3.1. Find the excitation propositions and output propositions for the following sequential circuit.

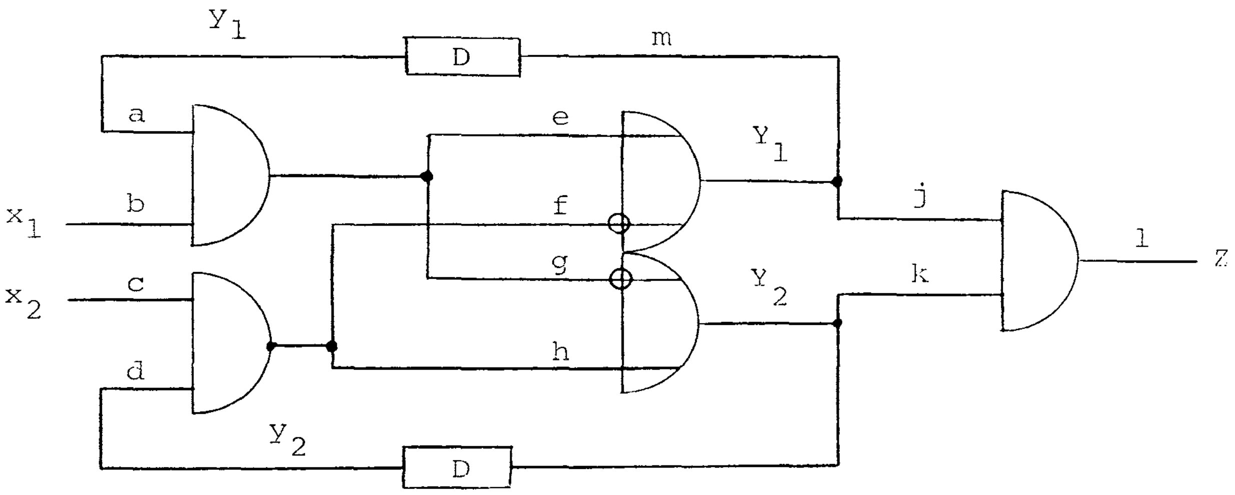


Figure 3.3. A Sequential Circuit With Two Feedbacks

(1) Start from the output, Z. The excitation proposition for Y_1 is

$$P(Y_1) = (P_e + P'_f)\, m_n + m_1 \quad . \; . \; . \; . \; . \; . \; . \; . \; . \qquad (3.4)$$

(2) Apply theorem $(X + Y)Z = XZ + YZ$ and $X + X = X$:

$$P(Y_1) = (P_e m_n + m_1) + (P'_f\, m_n + m_1) \quad . \; . \; . \qquad (3.5)$$

(3) Reduce it into compressed notation form:

$$P(Y_1) = P_e(m_1) + P'_f(m_1) \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \qquad (3.6)$$

Again using the same techniques, one can find the final standard form systematically.

$$P(Y_1) = ((P_a P_b)e_n + e_1)(m_1) + ((P_c P_d)'f_n + f_0)(m_1)$$

$$= P_a P_b(e_1 m_1) + (P'_c + P'_d)(f_0 m_1)$$

$$= P_a(e_1 m_1) \cdot P_b(e_1 m_1) + P'_c(f_0 m_1) + P'_d(f_0 m_1)$$

$$= y_1(a_1e_1m_1)\cdot x_1(b_1e_1m_1) + x_2'(c_0f_0m_1)+y_2'(d_0f_0m_1) \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; (3.7)$$

To sum up, the excitation proposition and output proposition could be found by applying three steps:

(1) Regard each term as an individual output. Find the output proposition in terms of its input propositions and element propositions.

(2) Use proper theorems to expand the expression into a sum of products form.

(3) Reduce the form to its compressed notation form. The three steps are used level by level until the primary inputs have been reached.

Hence, $P(Y_2)$ and $P(Z)$ are found as follows:

$$P(Y_2) = y_1'(a_0g_0n_1) + x_1'(b_0g_0n_1) + x_2(c_1h_1n_1)\cdot y_2(d_1h_1n_1) \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; (3.8)$$

$$\begin{aligned} P(Z) = {} & (y_1(a_1e_1j_1l_1)\cdot x_1(b_1e_1j_1l_1) + x_2'(g_0f_0j_1l_1) \\ & + y_2'(d_0f_0j_1l_1))\cdot(y_1'(a_0g_0k_1l_1) + x_1'(b_0g_0k_1l_1) \\ & + x_2(c_1h_1k_1l_1)\cdot y_2(d_1h_1k_1l_1)) \quad . \; . \; . \; . \; . \; . \; . \quad (3.9) \end{aligned}$$

Output functions in the presence of faults

If the wire b is now stuck-at-1, in the previous example, the element propositions b_1, $a_n, \ldots, l_n$ are assigned to be true, and the others to be false. The excitation functions are formed as follows:

Take the first terms of Equation (3.7), $y_1(a_1e_1m_1) \cdot x_1(b_1e_1m_1)$, to demonstrate the technique. Originally this terms is in the form

$$(((y_1a_n + a_1)e_n + e_1)m_n + m_1)(((x_1b_n + b_1)e_n + e_1)m_n + m_1).$$

Under the fault b_1,

$$b_1 = a_n = e_n = m_n = 1, \text{ and } b_n = a_1 = e_1 = m_1 = 0.$$

Substitute those values into the above term:

$$\begin{aligned} &y_1(a_1e_1m_1) \cdot x_1(b_1e_1m_1)\Big|_{b_1} \\ &= (((y_1 \cdot 1 + 0)1 + 0)\ 1 + 0)(((x_1 \cdot 0 + 1)\ 1 + 0)1 + 0) \\ &= y_1 \cdot 1 = y_1. \end{aligned}$$

Therefore,

$$P(Y_1)\Big|_{b_1} = y_1 + x_2' + y_2' \quad \dots\dots\dots\dots \quad (3.10)$$

$$P(Y_2)\Big|_{b_1} = y_1' + x_2y_2 \quad \dots\dots\dots\dots \quad (3.11)$$

Similiarly,

$$P(Z)\Big|_{b_1} = (y_1 + x_2 + y_2)(y_1 + x_2y_2) \quad \dots\dots\dots \quad (3.12)$$

The above expression is equivalent to a permanent signal of value 1 at wire b, when deriving the output function using ordinary Boolean Algebra.

B. Derivation of Fault Detection Test Sequences

The procedure has been stated in Chapter I. A more detailed description of the overall concept is given in Figure 3.4.

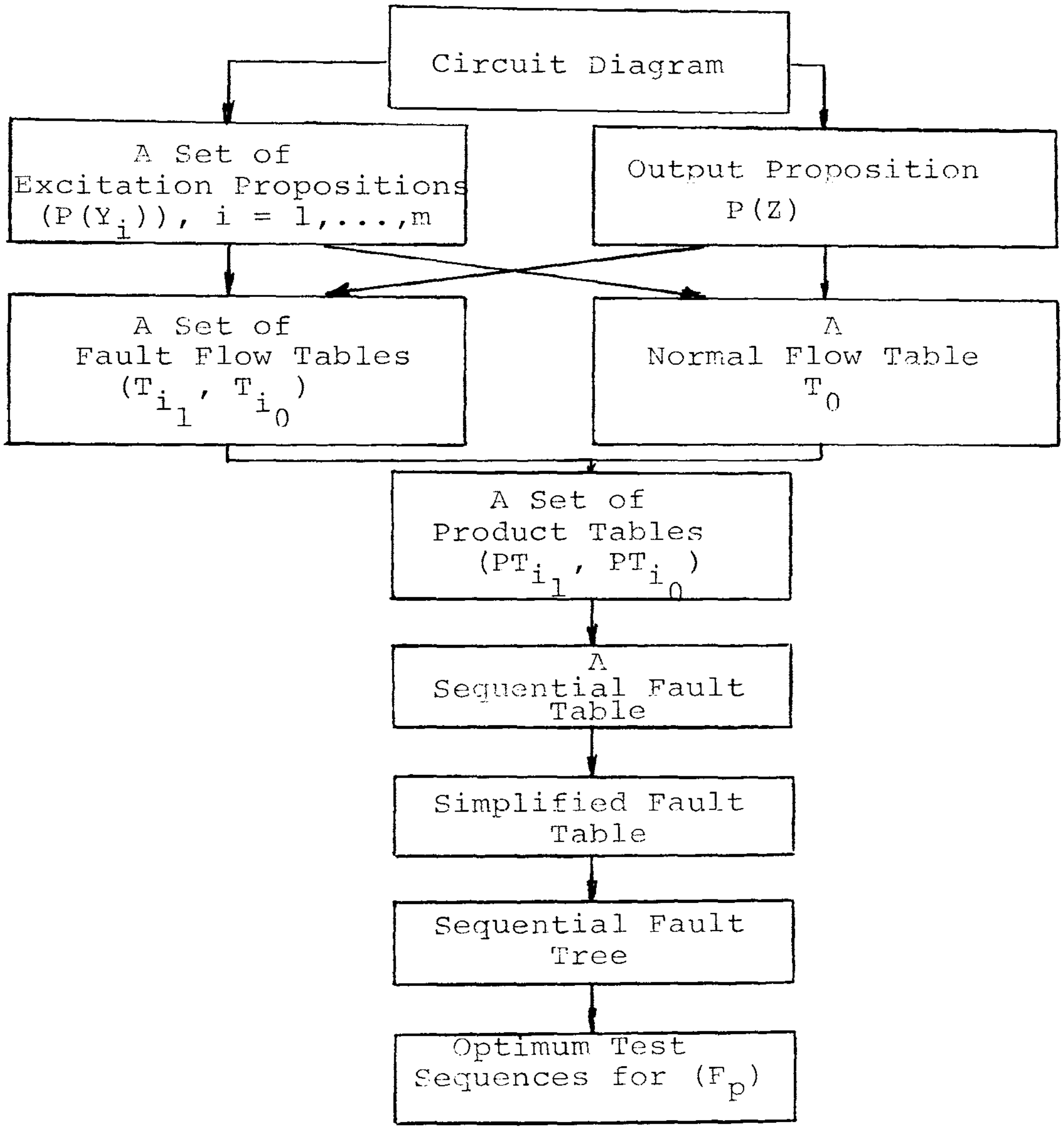

Circuit Diagram
A Set of Excitation Propositions (P(Y_i)), i = 1,...,m
Output Proposition P(Z)
A Set of Fault Flow Tables (T_{i_1}, T_{i_0})
A Normal Flow Table T_0
A Set of Product Tables (PT_{i_1}, PT_{i_0})
A Sequential Fault Table
Simplified Fault Table
Sequential Fault Tree
Optimum Test Sequences for (F_p)

Figure 3.4. Flow Chart of Overall Procedure

The following example is used to illustrate the procedure. The machine is said to be able to reset to a particular initial state (00), even in the presence of fault.

Example 3.2. Find the set of fault detection test sequences for $(F_p) = (a_1, d_1, e_1, q_1)$ in the following sequential logic. Note that not all possible faults are considered.

(1) Circuit diagram

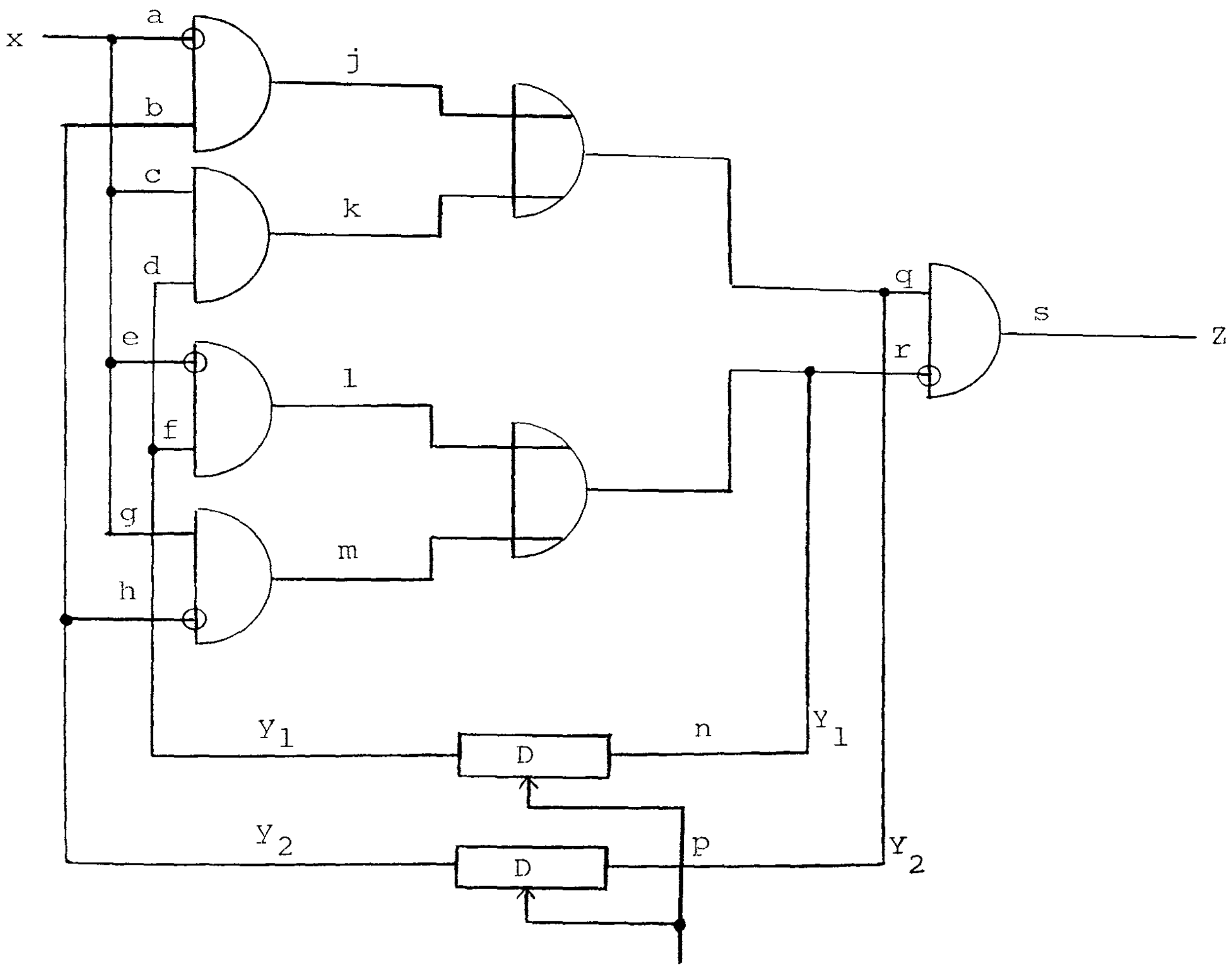


Figure 3.5. A Sequential Network With Two Feedbacks

(2) Excitation propositions and output proposition:

$$P(Y_1) = x'(a_0j_1p_1)\cdot y_1(b_1j_1p_1) + x(c_1k_1p_1)\cdot y_2(d_1k_1p_1),..(3.13)$$

$$P(Y_2) = x'(e_0l_1n_1)\cdot y_2(f_1l_1n_1) + x(g_1m_1n_1)\cdot y_1'(h_0m_1n_1)...(3.14)$$

$$P(Z) = ((x'(a_0j_1q_1s_1)\cdot y_1(b_1j_1q_1s_1) + x(c_1k_1q_1s_1)\cdot y_2(d_1k_1q_1s_1))$$
$$(x(e_1l_0r_0s_1) + y_2'(f_0l_0r_0s_1))(x'(g_0m_0r_0s_1) + y_1(h_1m_0r_0s_1)).$$
$$....(3.15)$$

(3) Construction of the fault-free flow table, T_0, and a set of fault flow tables for (F_p), which are:

$$(T_{i_1}, T_{i_0}) = (T_{a_1}, T_{d_1}, T_{e_1}, T_{q_1}).$$

y_1y_2 \ x	0	1
0 0	00,0	01,0
0 1	01,0	11,0
1 1	11,0	10,1
1 0	10,1	00,0

$\longrightarrow$

\ x	0	1
A	A,0	B,0
B	B,0	C,0
C	C,0	D,1
D	D,1	A,0

Table 3.1. A Fault-Free Flow Table, T_0

In the presence of fault a_1, the propositions reduce to the following form:

$$P(Y_1)|_{a_1} = xy_2 \quad . . . . \quad (3.16)$$
$$P(Y_2)|_{a_1} = x'y_2 + xy_1' \quad . . . . \quad (3.17)$$
$$P(Z)|_{a_1} = xy_2(x + y_2')(x' + y_1) = xy_1y_2 \quad . . . \quad (3.18)$$

Hence the set of fault flow tables are found as shown in Table 3.2.

$y_1y_2 \backslash x$	0	1
0 0	00,0	01,0
0 1	01,0	11,0
1 1	01,0	10,1
1 0	00,0	00,0

$y_1y_2 \backslash x$	0	1
0 0	00,0	11,0
0 1	01,0	11,0
1 1	11,0	10,1
1 0	10,1	10,1

$y_1y_2 \backslash x$	0	1
0 0	00,0	01,0
0 1	00,0	11,0
1 1	10,1	10,1
1 0	10,1	00,0

$y_1y_2 \backslash x$	0	1
0 0	00,1	01,0
0 1	01,0	11,0
1 1	11,0	10,1
1 0	10,1	00,1

$\backslash x$	0	1
A	A,0	B,0
B	B,0	C,0
C	B,0	D,1
D	A,0	A,0

$= T_{a_1}$

$\backslash x$	0	1
A	A,0	C,0
B	B,0	C,0
C	C,0	D,1
D	D,1	D,1

$= T_{d_1}$

$\backslash x$	0	1
A	A,0	B,0
B	A,0	C,0
C	D,1	D,1
D	D,1	A,0

$= T_{e_1}$

$\backslash x$	0	1
A	A,1	B,0
B	B,0	C,0
C	C,0	D,1
D	D,1	A,1

$= T_{q_1}$

Table 3.2. A Set of Fault Flow Tables, $(T_{a_1}, T_{d_1}, T_{e_1}, T_{q_1})$

(4) Construct the set of product tables

$(PT_{i_1}, PT_{i_0}) = (PT_{a_1}, PT_{d_1}, PT_{e_1}, PT_{q_1})$.

To start with the same initial state A, the entries of each product table are found as the product of corresponding states in T_0 and each fault flow table. If a new entry occurs, such as (CB,00), with the same output value, a new row should be added since a new state in the product table has occured. However, the output value cannot indicate the fault yet. Therefore, if an entry like

(DC,10) or (DA,10) occurs, this entry is marked with a cross "X". There is no need to add a new row since the fault is already detected.

The major purpose of a product table is to denote a set of sequences. The internal states of the component machines and their outputs are of secondary importance. Hence, the product table, PT_{a_1}, can be trimmed down to a form shown in Table 3.3.

$T_0 \times T_{a_1} =$

x	0	1
AA	AA,00	BB,00
BB	BB,00	CC,00
CC	CB,00	DD,11
CB	CB,00	DC,00
DD	DA,10	AA,00

→

x	0	1
E	E	F
F	F	G
G	H	I
H	H	X
I	X	E

$= PT_{a_1}$

Flow tables for the remaining faults of (F_p) are derived in a similar manner.

x	0	1
J	J	K
K	K	X

$= PT_{d_1}$

x	0	1
L	L	M
M	N	0
N	N	P
O	X	Q
P	R	X
Q	Q	L
R	R	X

$= PT_{e_1}$

x	0	1
S	X	T
T	T	U
U	U	V
V	V	X

$= PT_{q_1}$

Table 3.3. A Set of Product Tables, (PT_{a_1},PT_{d_1}, PT_{e_1} PT_{q_1})

(5) Construct a sequential fault table. It is started with an initial product state (EJLS), where each literal corresponds to the initial state of each product table. New rows are added when new states occur. The table is completed when there is a row containing all entries which have occured as previous entries.

			x		
a_1	d_1	e_1	q_1	0	1
E	J	L	S	E J L X	F K M T
F	K	M	T	F K N T	G X O U
F	K	N	T	F K N T	G X P U
G	X	O	U	H X X U	I X Q V
G	X	P	U	H X R U	I X X V
H	X	R	U	H X R U	X X X V
I	X	Q	V	X X Q V	E X L X

Table 3.4. Sequential Fault Table

(6) Simplify the sequential fault table by column dominance technique. First remove all the letters and leave only crosses in the table. Fault d_1 need not be considered since its column dominates columns

for faults a_1 and e_1. Any sequence detecting fault a_1 must use entry (XXQV) or entry (XXXV). Fault d_1 can also be detected by these entries and thus contributes nothing to the derivation of the test.

		a_1	d_1	e_1	q_1
x	E J L S				X
	F K M T				
	F K N T				
0	G X O U		X	X	
	G X P U		X		
	H X R U		X		
	I X Q V	X	X		
	E J L S				
	F K M T		X		
	F K N T		X		
1	G X O U		X		
	G X P U		X	X	
	H X R U	X	X	X	
	I X Q V		X		X

Table 3.5. Simplification Process Table

(7) Expand the simplified sequential fault table into a sequential fault tree. In terms of the sequential fault table, an optimum test is the shortest sequence of input symbols using a set of entries having at least one cross in every position. Figure 3.6 illustrates the process of finding a sequential fault tree. The process is as follows.

(a) Start with the initial state, (ELS). Input variable x (valued 0 or 1) or a reset value is applied to the system. The three product states result as the next level of the tree. It is assumed, in this example, that a reset signal and input signal cannot be applied simultaneously. If a reset signal can be applied simultaneously with an input symbol, it must be regarded as an additial input variable. In this event, the number of branches leaving a product state will be twice the number of input symbols; four in the example.

(b) If a cross appears in a product state, all subsequent branches must contain a cross in the same position. If a product state has appeared previously, it is checked out and can no longer be considered as another initial state at this level. Further extension of this branch will result in an idential path and hence is fruitless.

(c) The branch is terminated if all three product states have appeared previously and are to be checked out, or, if a product state with all crosses has occured.

(d) Then, the optimum tests are those input sequences along the branches from the initial state (ELS) to the terminating states with all crosses, (XXX).

Therefore, the set of optimum test sequences is

(X_{op}) = (01101, 11011) . . . . (3.13)

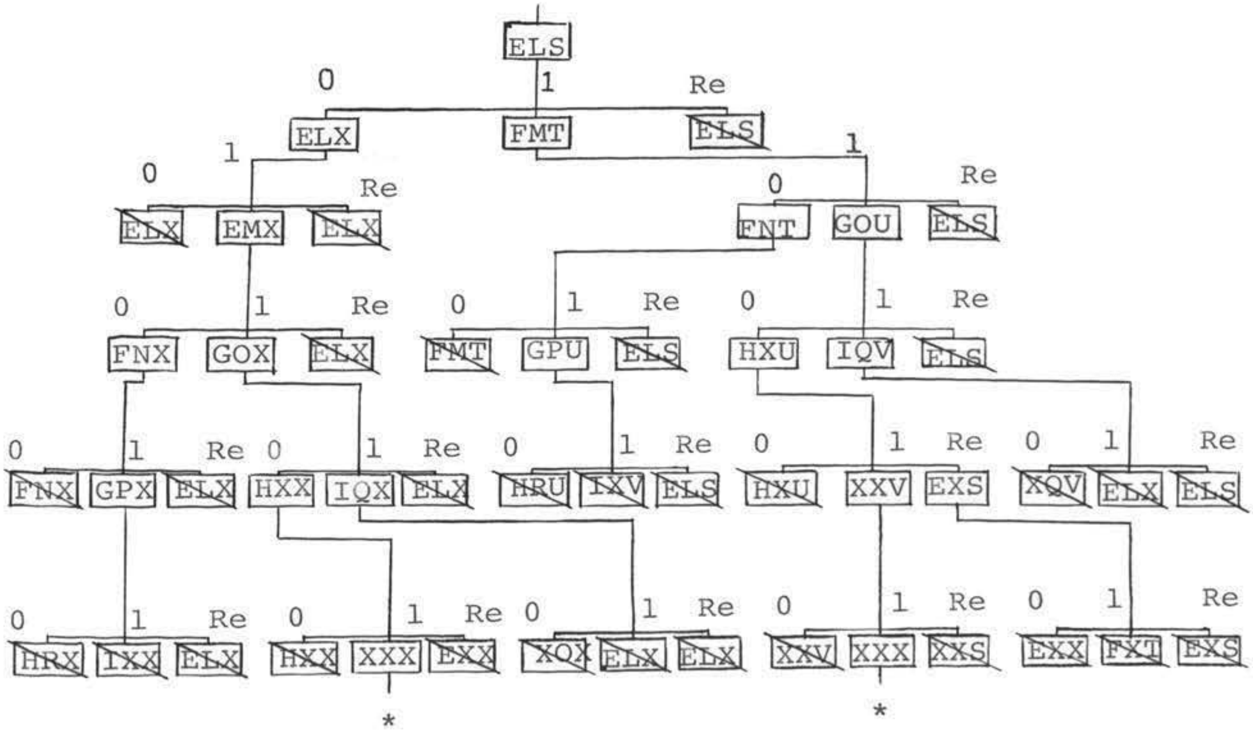


Figure 3.6. Sequential Fault Tree and Results

C. Limitations and Possible Improvements of the Procedure

The procedure presented in the previous sections has several limitations. First, the resulting test sequence can only detect a specific set of faults, (F_p). Practically, if (F_p) contains all the possible m faults, the flow tables will be too large. Hence, it is quite reasonable to consider applying path sensitizing concepts, to determine which faults should be chosen as a set, in order to save unnecessary work. Note that feedback loops should be considered as a part of the path. A suggestion on path sensitizing techniques for sequential machines is presented in Chapter VI.

Secondly, one of the assumptions stated that a machine, even in the presence of faults, may be reset to a particular initial state. What happens when the reset cannot force a defective machine to the initial state, q_λ? The modified system was investigated by Poage and is introduced in the next chapter.

The last limitation is due to the memory consuming nature of the procedure; therefore, it can only be applied to relatively small machines. As a result, practical applications are restricted to small modules or special units. However, many circuits to be checked frequently are comparatively small, and larger networks usually consist of repeated or iterative structures. This procedure can first be applied to small basic modules individually. Then, the results can possibly be combined to obtain an overall and complete test set.

In Chapter IV, detection of faults from a set containing reset failures is considered. The modified system will become more memory consuming since more flow tables should be taken into consideration.

CHAPTER IV

DERIVATION OF OPTIMUM TEST TO DETECT FAULTS FROM A SET CONTAINING RESET FAILURES

A. Introduction

The assumption made for the approach described in the previous chapters states that a reset signal can always force the set of all m+1 machines, M_0, $M_1, \ldots, M_m$ into some initial or starting state, q_λ. What happens when the reset does not guarantee this condition? Two undesirable consequences of this condition will be considered in this chapter:

(1) Certain feedback lines, under fault conditions, can not possibly be reset to a particular initial state q_λ.

(2) Some feedback lines are permanently reset to a particular state q_λ. One may say that the feedback input is stuck at the reset value. It is determined that this reset signal can be found to be equivalent to an input signal, when the reset fault is present.

In both cases, the previous flow tables would have to be modified for deriving fault detection tests. This is done by adding an extra column, called the reset column, to the excitation table, to show the effect of reset failures. The other parts of the procedure are similar to the previous system. The result, however, will consist of test inputs as well as the reset inputs. This is demonstrated in the following example.

Example 4.1. To show the addition of reset columns to the normal flow table and flow tables with reset faults, respectively. The sequential network is shown in Figure 3.5.

(1) The normal excitation table is modified to the following form:

x,Re y_1y_2	0	1	Re
0 0	00,0	01,0	00
0 1	01,0	11,0	00
1 1	11,0	10,1	00
1 0	10,1	00,0	00

Hence the normal flow table turns out to be

x,Re	0	1	Re
A	A,0	B,0	A
B	B,0	C,0	A
C	C,0	D,0	A
D	D,1	A,1	A

Table 4.1. A Normal Flow Table with Reset Column

(2) Suppose now that y_1 can not be reset to an initial 0 state; i.e., the reset signal cannot change the present state of y_1. To show this effect in the excitation table, the Y_1 portion of the reset column is replaced by the y_1 column, but leaves the column for input x unchanged.

y_1y_2 \ x,Re	0	1	Re
0 0	00,0	01,0	00
0 1	01,0	11,0	00
1 1	11,0	10,1	00
1 0	10,1	00,1	00

Y_1Y_2,Z

x,Re	0	1	Re
A	A,0	B,0	A
B	B,0	C,0	A
C	C,0	D,1	D
D	D,1	A,0	D

S,Z

Table 4.2. A Reset Fault Flow Table with Reset Column

(3) Consider now that the memory element y_1 is permanently reset; i.e., the reset column is not changed, but the Y_1 portions of columns x = 0 and x = 1 are replaced by the y_1 portion of the reset column.

y_1y_2 \ x,Re	0	1	Re
0 0	00,0	01,0	00
0 1	01,0	01,0	00
1 1	01,0	00,1	00
1 0	00,1	00,0	00

x,Re	0	1	Re
A	A,0	B,0	A
B	B,0	B,0	A
C	B,0	A,1	A
D	A,1	A,0	A

Table 4.3. Another Reset Fault Flow Table

B. Change in Procedure When Deriving the Optimum Tests

Since a reset does not always return a defective machine to the initial state, more than one product table is required to describe one machine, under fault conditions.

Let the set of faults containing reset faults be $(F_p) = (a_1, e_1, q_1, R_0, R_1)$

(R_0) denoting that both y_1 and y_2 cannot be reset to the initial state (00), and

(R_1) denoting that both y_1 and y_2 are permanently reset to the initial state (00).

Product tables PT_{a_1}, PT_{e_1}, PT_{q_1}, and PT_{R_1} can be easily found as done previously.

For the fault R_0, one does not know which of the initial states to start with. Hence, all the four starting values, A, B, C, and D have to be considered individually to avoid the possibility of a test which fails to detect a fault under certain initial conditions. The set of product tables for fault R_0 is

$$(PT_{R_0}\text{-}A,\ PT_{R_0}\text{-}B,\ PT_{R_0}\text{-}C,\ PT_{R_0}\text{-}D).$$

The final product, of the four product tables, can be formed in a list sequence which detects fault R_0, regardless of the initial states.

(1) Forming the set of flow tables for (F_p)

x,Re	0	1	Re
A	A,0	B,0	A
B	B,0	C,0	A
C	C,0	D,1	A
D	D,1	A,0	A

$= T_0$

x,Re	0	1	Re
A	A,0	B,0	A
B	A,0	C,0	A
C	D,1	D,1	A
D	D,1	A,0	A

$= T_{e_1}$

x,Re	0	1	Re
A	A,0	B,0	A
B	B,0	C,0	B
C	C,0	D,1	C
D	D,1	A,0	D

$= T_{R_0}$

x,Re	0	1	Re
A	A,0	B,0	A
B	B,0	C,0	A
C	B,0	D,1	A
D	A,0	A,0	A

$= T_{a_1}$

x,Re	0	1	Re
A	A,1	B,0	A
B	B,0	C,0	A
C	C,0	D,1	A
D	D,1	A,1	A

$= T_{q_1}$

x,Re	0	1	Re
A	A,0	A,0	A
B	A,0	A,0	A
C	A,0	A,1	A
D	A,1	A,0	A

x,Re	0	1	Re
A	A,0	A,0	A

$= T_{R_1}$

Table 4.4. A Set of Flow Tables

(2) Forming the set of product tables

a) $(PT_{a_1}, PT_{e_1}, PT_{q_1}, PT_{R_1})$

x,Re	0	1	Re
E	E	F	E
F	F	G	E
G	H	I	E
H	H	X	E
I	X	E	E

= PT_{a_1}

x,Re	0	1	Re
L	L	M	L
M	N	O	L
N	N	P	L
O	X	Q	L
P	R	X	L
Q	Q	L	L
R	R	X	L

= PT_{e_1}

x,Re	0	1	Re
S	X	T	S
T	T	U	S
U	U	V	S
V	V	X	S

= PT_{q_1}

x,Re	0	1	Re
E	F	F	E
F	F	G	E
G	G	X	E

= PT_{R_1}

Table 4.5. A Set of Product Tables

b) The set of product tables for R_0,

(PT_{R_0}-A, PT_{R_0}-B, PT_{R_0}-C, PT_{R_0}-D)

x,Re	0	1	Re
AA	AA00	BB00	AA
BB	BB00	CC00	AB
CC	CC00	DD11	AC
AB	AB00	BC00	AB
DD	DD11	AA00	AD
AC	AC00	BD01	AC
BC	BC00	CD01	AC
AD	AD01	BA00	AD
BA	BA00	CB00	AA
CB	CB00	DC10	AB

→

x,Re	0	1	Re
E	E	F	E
F	F	G	H
G	G	I	H
H	H	K	H
I	I	E	L
J	J	X	J
K	K	X	J
L	X	M	L
M	M	N	E
N	N	X	H

= PT_{R_0}-A

x,Re	0	1	Re
AB	AB00	BC00	AB
BC	BC00	CD01	AC
AC	AC00	BD01	AC

→

x,Re	0	1	Re
E	E	F	E
F	F	X	G
G	G	X	G

$= PT_{R_O}\text{-B}$

x,Re	0	1	Re
AC	AC00	BD01	AC

→

x,Re	0	1	Re
E	E	X	E

$= PT_{R_O}\text{-C}$

x,Re	0	1	Re
AD	AD01	BA00	AD
BA	BA00	CB00	AA
CB	CB00	DC10	AB
AA	AA00	BB00	AA
AB	AB00	BC00	AB
BB	BB00	CC00	AB
BC	BC00	CD01	AC
CC	CC00	DD11	AC
AC	AC00	BD01	AC
DD	DD11	AA00	AD

→

x,Re	0	1	Re
E	X	F	E
F	F	G	H
G	G	X	I
H	H	J	H
I	I	K	I
J	J	L	I
K	K	X	M
L	L	N	M
M	M	X	M
N	N	H	E

$= PT_{R_O}\text{-D}$

Table 4.6. A Set of Product Tables for (R_0)

C. Results

Once the product tables have been formed for all faults of (F_p), the derivation of an optimum detection test proceeds exactly as in the previous chapter. In other words, the product of eight product tables for the fault set, $(F_p) = (a_1, e_1, q_1, R_0, R_1)$ is formed to obtain a sequential fault table. Dominance simplification is used, and the resulting test sequence is found to be

$$x_{op} = (1101 \text{ Re } 0).$$

Note that the reset value is treated as a part of the test sequence. It implies that Re = 0 at the beginning.

Compare the results with test derived for the same network in Chapter III:

(a) $(F_p) = (a_1, e_1, q_1)$; $(X_{op}) = (01101; 11011)$.

(b) $(F_p) = (a_1, e_1, q_1, R_0, R_1)$; $(X_{op}) = 1101 \text{ Re } 0$.

The result has the same length, five test inputs, but detects more faults.

The next chapter will describe the derivation of optimum tests for flip-flop circuits. The procedure will be modified to sequential networks containing R-S flip-flops. An example is investigated in detail to demonstrate the proposed technique.

CHAPTER V

FAULT DETECTION IN SEQUENTIAL LOGIC CONTAINING FLIP-FLOPS

A. Introduction

In Chapter III and IV, the memory elements for the sequential machine were represented by the presence of feedback loops. Practically, the flip-flop (bistable multivibrator) is a more useful memory device for binary digit storage and switching functions, because of its speed and bistable characteristics. This chapter is devoted to the derivation of fault detection tests for sequential machines having only R-S flip-flops as memory elements.

Some characteristics of R-S flip-flops are stated as follows:

(1) The Reset-Set flip-flop circuit can be represented by either the symbol shown in Figure 5.1 or by two NOR gates as in Figure 5.2.

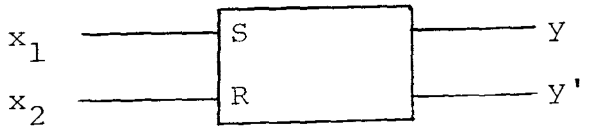


Figure 5.1. R-S Flip-Flop Symbol

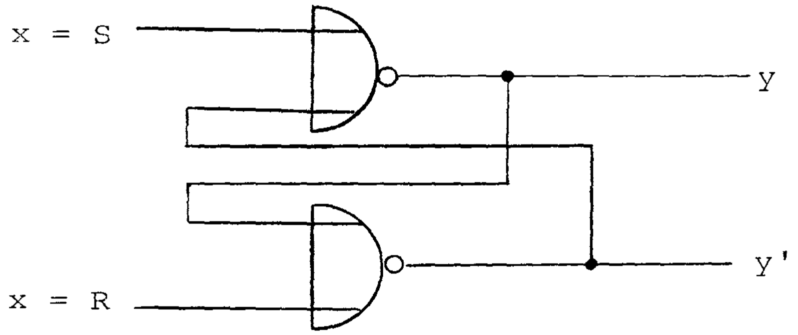


Figure 5.2. R-S Flip-Flop Logic Circuit

where y: Flip-flop output

x: input variable

When x = S, it is called the set input; i.e., S = 1 can set y to one, and when x = R, it is called the reset input; i.e., R = 1 can set y to zero.

(2) The truth table and characteristic function of an R-S flip-flop are shown as follows:

S	R	y	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	1	0	d
1	1	1	d
1	0	0	1
1	0	1	1

$$Y = S + R'y \quad . \; . \; . \; . \; . \; . \; . \; . \; . \qquad (5.1)$$

Table 5.1. Truth Table of R-S Flip-flop

Note that the situation where S = R = 1 is assumed not to occur. It can only happen when certain faults cause this state to occur. This is discussed in a later section.

(3) Therefore, the state transition function is

$$Y = f(x,y)$$

where x = R, S and Y is the result state.

B. Modified Sequential Model

The flip-flop is called an internal memory device. The feedback nature of the flip-flop is not as clearly presented as before. Hence, the general model for a sequential machine described in Chapter II seems to be inaccurate. However, feedback lines are actually contained within the flip-flop circuitry. In this chapter, the faults within the block of flip-flops are not considered.

To express the presence of R-S flip-flops, the synchronous sequential model is modified to the form shown in Figure 5.3.

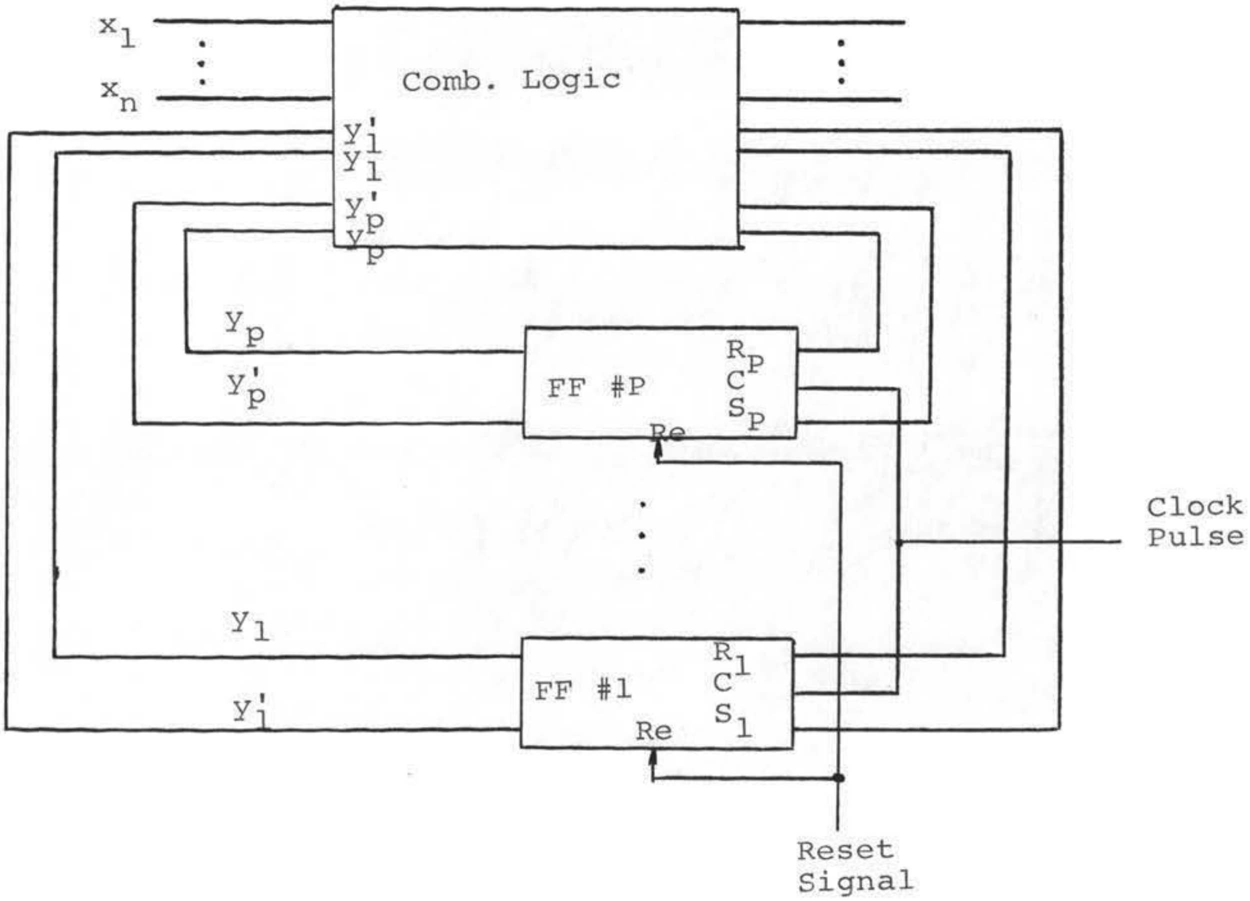


Figure 5.3. A Synchronous Sequential Model with Flip-Flops

where, x_i: primary input, $i = 1,2,\ldots,n$.

z_j: primary output, $j = 1,2,3,\ldots,m$.

S_k: set input for kth flip-flop.

R_k: reset input for kth flip-flop.

y_k: flip-flop output for kth flip-flop

$k = 1,2,\ldots,p$.

C : clock pulse governing the operation of the circuit.

The reset assumption is again used to force all $m + 1$ machines into a particular initial state q_λ.

C. Change of Procedure for Deriving Optimum Tests

(1) Network Analysis

The output proposition remains unchanged. For the excitation proposition, let

P(S) be the excitation proposition for set inputs.

P(R) be the excitation proposition for reset inputs.

(2) Forming the Flow Tables

Again, a given machine is analyzed to obtain a set of $m + 1$ flow tables $(T_0, T_1, \ldots, T_m)$ for machines $(M_0, M_1, \ldots, M_m)$. Here $m + 1$ transition tables are required in order to find the fault flow tables.

(3) Forming the Product Tables

A sequence which detects each fault is found by comparing tables $T_1, T_2, \ldots, T_m$ with T_0, individually. The results are called a set of product tables,

$$(PT_1, PT_2, \ldots, PT_m).$$

(4) The Sequential Fault Table

Combine all the m product tables, resulting in a sequential fault table. The column dominance simplification process is performed, then, to wipe out those sets of sequences which contain (or dominate) other sets. The result is called a simplified sequential fault table.

(5) Sequential Fault Tree and Result
Optimum test sequences are found by expanding the simplified sequential fault table into a sequential fault tree.

A flow chart for the derivation of optimum tests for fault detection in sequential machines containing R-S flip-flops is shown in Figure 5.4. An example is given in the next section to illustrate this procedure.

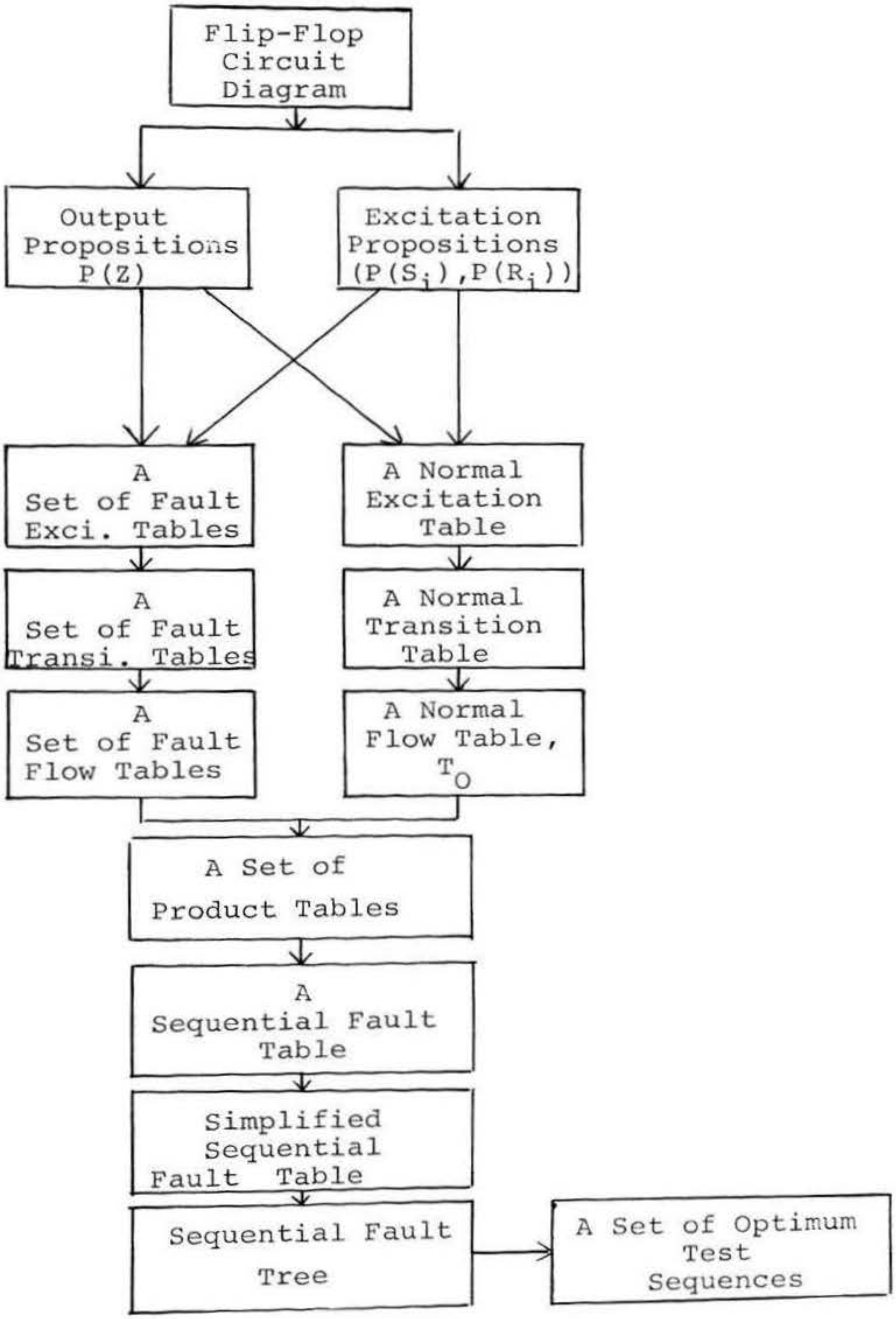

Flip-Flop Circuit Diagram
Output Propositions P(Z)
Excitation Propositions (P(S_i),P(R_i))
A Set of Fault Exci. Tables
A Normal Excitation Table
A Set of Fault Transi. Tables
A Normal Transition Table
A Set of Fault Flow Tables
A Normal Flow Table, T_O
A Set of Product Tables
A Sequential Fault Table
Simplified Sequential Fault Table
Sequential Fault Tree
A Set of Optimum Test Sequences

Figure 5.4. A Modified Flow Chart for Fault Detection of A Flip-Flop Circuit

D. An Example

Find optimum tests for the set of faults

$$(F_p) = (a_1, g_1, l_1).$$

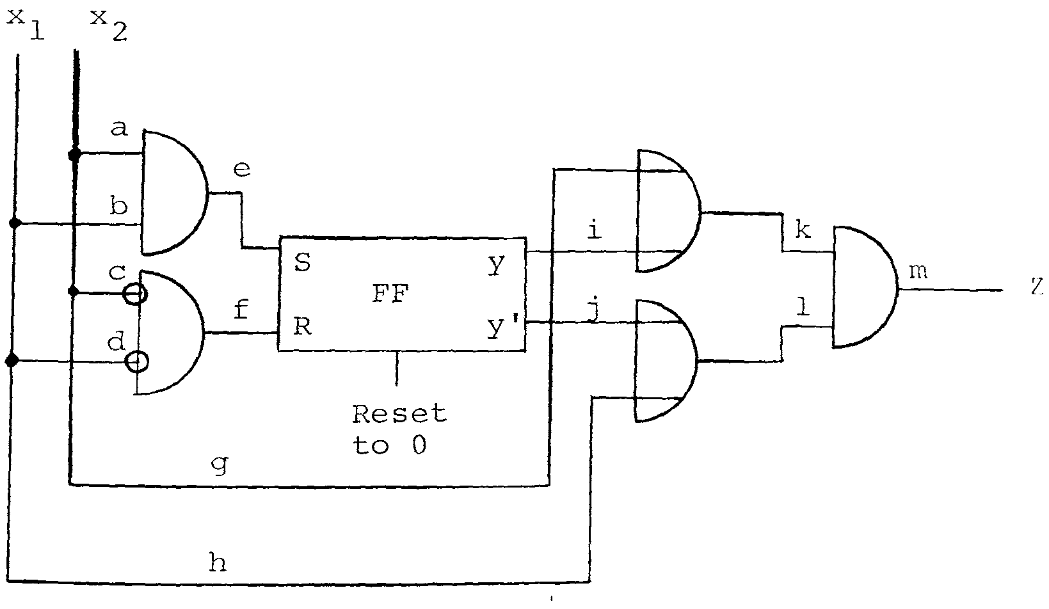


Figure 5.5. A Sequential Net with One R-S Flip-Flop

(1) Network Analysis

$$P(S) = P_a P_b e_n + e_1$$
$$= x_1(a_1 e_1) \cdot x_2(b_1 e_1) \quad . \; . \; . \; . \; . \; . \; . \; . \qquad (5.2)$$

$$P(R) = P'_c P'_d f_n + f_1$$
$$= x'_1(c_0 f_1) \cdot x'_2(d_0 f_1) \quad . \; . \; . \; . \; . \; . \; . \; . \qquad (5.3)$$

$$\begin{aligned} P(Z) &= (P_k+P_l)m_n + m_1 \\ &= P_k(m_1) + P_l(m_1) \\ &= P_iP_j(k_1m_1) + P_jP_h(l_1m_1) \\ &= y(i_1k_1m_1)\cdot x_1(g_1k_1m_1) + y'(j_0l_1m_1)\cdot x_2(h_1l_1m_1) \end{aligned} \quad . \; . \; . \; . \; . \; . \; . \quad (5.4)$$

(2) Forming Flow Tables

a) If no faults occur, the propositions reduce to

$$P(S) = x_1x_2 \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (5.5)$$

$$P(R) = x_1'x_2' \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (5.6)$$

$$P(Z) = yx_1 + y'x_2 \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (5.7)$$

The excitation table, transition table and flow table are

y \ x_1x_2	00	01	11	10
0	01,0	00,1	10,1	00,1
1	01,0	00,0	10,1	00,1

SR,Z

y \ x_1x_2	00	01	11	10
0	0,0	0,1	1,1	0,0
1	0,0	1,0	1,1	1,1

Y,Z

x_1x_2	00	01	11	10
A	A,0	A,1	B,1	A,0
B	A,0	B,0	B,1	B,1

S,Z

Table 5.2. The Normal Case

b) In the presence of fault a_1,

$$P(S)|_{a_1} = x_2 \quad . . . . \quad (5.8)$$

$$P(R)|_{a_1} = x_1'x_2' \quad . . . . \quad (5.9)$$

$$P(Z)|_{a_1} = yx_1 + y'x_2 \quad . . . . \quad (5.10)$$

The circled entries indicate the malfunction caused by fault a_1.

y \ x_1x_2	00	01	11	10
0	01,0	10,1	10,1	00,0
1	01,0	10,0	10,1	00,1

SR,Z

y \ x_1x_2	00	01	11	10
0	0,0	1,1	1,1	0,0
1	0,0	1,0	1,0	1,1

Y,Z

T_{a_1} =

x_1x_2	00	01	11	10
A	A,0	Ⓑ,1	B,1	A,0
B	A,0	B ,0	B,1	B,1

S,Z

Table 5.3. The Fault Flow Table T_{a_1}

c) In the presence of fault g_1,

$$P(S)|_{g_1} = x_1x_2 \quad . . . . \quad (5.11)$$

$$P(R)|_{g_1} = x_1'x_2' \quad . . . . \quad (5.12)$$

$$P(Z)|_{g_1} = y + y'x_2 \quad . . . . \quad (5.13)$$

$y \backslash x_1x_2$	00	01	11	10
0	01,0	00,1	10,1	00,0
1	01,1	00,1	10,1	00,1

SR,Z

$y \backslash x_1x_2$	00	01	11	10
0	0,0	0,1	1,1	0,0
1	0,1	1,1	1,1	1,1

Y,Z

T_{g_1} =

x_1x_2	00	01	11	10
A	A,0	A,1	B,1	A,0
B	A,①	B,①	B,1	B,1

S,Z

Table 5.4. Fault Flow Table T_{g_1}

d) In the presence of fault 1_1,

$P(S)|_{1_1} = x_1x_2$ . . . . (5.14)

$P(R)|_{1_1} = x_1'x_2'$ . . . . (5.15)

$P(Z)|_{1_1} = x_1y$ . . . . (5.16)

$y \backslash x_1x_2$	00	01	11	10
0	01,0	00,0	10,0	00,0
1	01,0	00,0	10,1	00,1

SR,Z

y \ x_1x_2	00	01	11	10
0	0,0	0,0	1,0	0,0
1	0,0	1,0	1,1	1,1

Y,Z

x_1x_2	00	01	11	10
A	A,0	A,0	B,0	A,0
B	A,0	B,0	B,1	B,1

S,Z

Table 5.5. Fault Flow Table T_{1_1}

(3) Forming Product Tables

The set of product tables is formed by the product of T_0 with $T_1, T_2, \ldots, T_m$ individually. When a new state occurs with the same output value, a new row is to be added. If an entry occurs with different output values a cross is then placed in that entry.

a) $PT_{a_1} = T_0 \cdot T_{a_1}$

=

x_1x_2	00	01	11	10
A	AO	A1	B1	AO
B	AO	BO	B1	B1

·

x_1x_2	00	01	11	10
A	AO	B1	B1	AO
B	AO	BO	B1	B1

=

x_1x_2	00	01	11	10
AA	AAOO	AB11	BB11	AAOO
AB	AAOO	AB1O	BB11	ABO1
BB	AAOO	BBOO	BB11	BB11

=

x_1x_2	00	01	11	10
C	C	D	E	C
D	C	X	E	X
E	C	E	E	E

Table 5.6. Product Table PT_{a_1}

b) $PT_{g_1} = T_0 \cdot T_{g_1}$

=

x_1x_2	00	01	11	10
A	AO	A1	B1	AO
B	AO	BO	B1	B1

·

x_1x_2	00	01	11	10
A	AO	A1	B1	AO
B	A1	B1	B1	B1

=

x_1x_2	00	01	11	10
AA	AAOO	AA11	BB11	AAOO
BB	AAO1	BBO1	BB11	BB11

=

x_1x_2	00	01	11	10
F	F	F	G	F
G	X	X	G	G

Table 5.7. Product Table PT_{g_1}

c) $PT_{1_1} = T_0 \cdot T_{1_1}$

=

x_1x_2	00	01	11	10
A	AO	A1	B1	AO
B	AO	BO	B1	B1

·

x_1x_2	00	01	11	10
A	AO	AO	BO	AO
B	AO	BO	B1	B1

=

x_1x_2	00	01	11	10
AA	AAOO	AA1O	BB1O	AAOO
BB	AAOO	BBOO	BB11	BB11

=

x_1x_2	00	01	11	10
I	I	X	X	I
J	I	J	J	J

Table 5.8. Product Table PT_{1_1}

(4) Forming Sequential Fault Table

Combining all the flow tables and starting at the initial state CFI, the sequential fault table is formed by adding a new row to the table when a new state occurs. The table is complete when there is a row for every entry that has appeared previously.

$a_1g_1l_1$ \ x_1x_2	00	01	11	10
C F I	CFI	DFX	EGX	CFI
D F X	CFX	XFX	EGX	XFX
E G X	CXX	EXX	EGX	EGX

x_1	x_2		a_1	g_1	l_1
O	O	CFI			
		DFX			X
		EGX		X	X
O	1	CFI			X
		DFX	X		X
		EGX		X	X
1	1	CFI			X
		DFX			X
		EGX			X
1	O	CFI			
		DFX	X		X
		EGX			X

Table 5.9. Sequential Fault Table and Simplification of Sequential Fault Table

(5) Sequential Fault Tree and Result

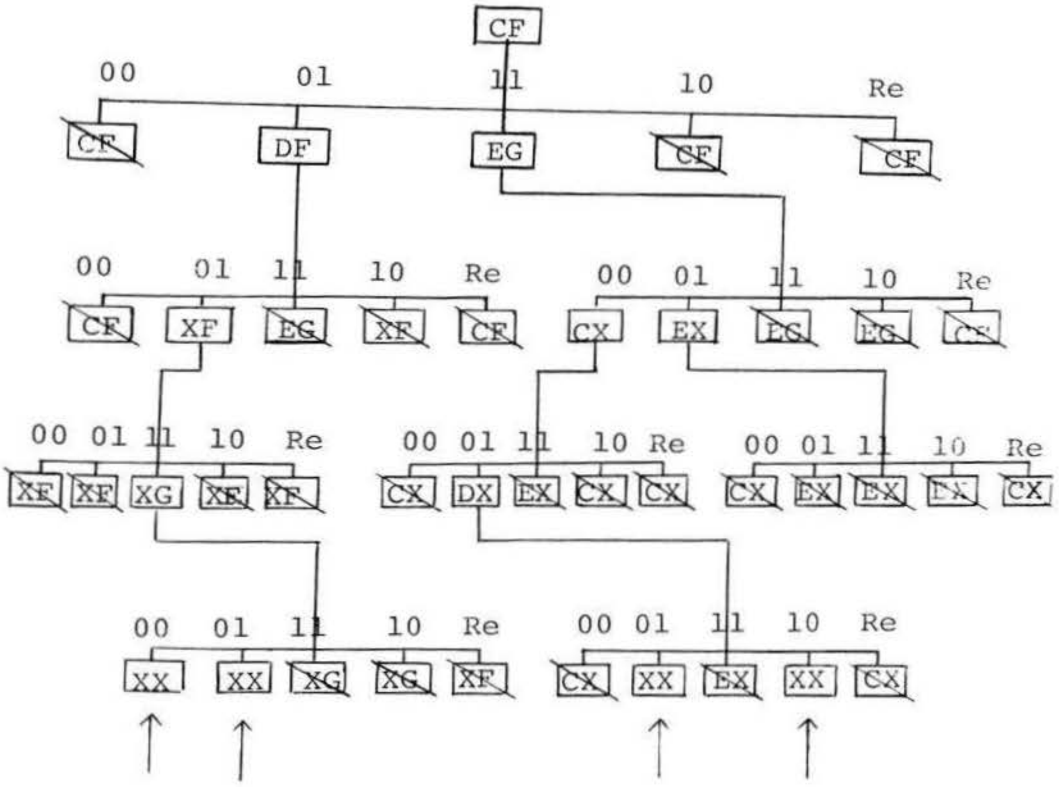


Figure 5.6. Sequential Fault Tree

From Figure 5.6, it can be determined that the resulting set of optimum test sequences are

$$(X_{op}) = ((00,11,01,01);\ (01,11,01,01);\ (01,01,00,11);\ (10,01,00,11))$$

E. Discussion

A fault can sometimes cause R = S = 1 to occur in the transition table. The (11) entry of the table will then provide no information concerning fault detection since the value in the corresponding flow table is a 'don't care' entry. This situation is illustrated by the following example.

Example 5.2.

If the fault d_1 occurs, in the previous network, the excitation and output propositions are reduced to

$$P(S)\big|_{d_1} = x_1x_2 \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (5.17)$$

$$P(R)\big|_{d_1} = x_1' \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (5.18)$$

$$P(Z)\big|_{d_1} = yx_1 + y'x_2 \quad . \; . \; . \; . \; . \; . \; . \; . \; . \; . \quad (5.19)$$

The excitation, transition, and flow tables are formed as follows:

y \ x_1x_2	00	01	11	10
0	00,0	00,1	11,1	01,0
1	00,0	00,0	11,1	01,1

SR,Z

y \ x_1x_2	00	01	11	10
0	0,0	0,1	d,1	0,0
1	1,0	1,0	d,1	0,1

Y,Z

x_1x_2	00	01	11	10
A	A,0	A,1	ⓓ,1	A,0
B,	Ⓑ, 0	B,0	ⓓ,1	Ⓐ,1

S,Z

Table 5.10. Don't Care Entries Caused by Fault d_1

The entries, which are different from the normal flow table, are marked by circles as shown in Table 5.10. Fortunately, the test sequence for d_1 can be found by the other two circles, Ⓐ and Ⓑ , when forming the product table. The 'don't care' conditions are regarded as normal, since they contribute no information for the test sequence. However, if only circles portions of the flow table are don't cares, one cannot detect this particular fault.

F. Summary

The object of this chapter was to describe a technique for deriving optimum test sequences for synchronous sequential machines with R-S flip-flops. It is seen that the reset constraint must be used in the system. Faults in the reset circuitry were not considered.

The R-S flip-flops are represented by blocks. The technique can be applied to the outside of the blocks. The next chapter will present a conclusion to the study, and some suggestions for further work.

CHAPTER VI

CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

A. Conclusions

The purpose of this study was to derive fault detection tests for sequential machines. The procedure starts with network analysis. That is, the machines are characterized by input symbols, internal elements and internal states, which are suitable to form a set fault flow tables. An algorithm is then presented to generate the set of optimum test sequences that is guaranteed to produce an output sequence different from the normal output sequence. Then, the set of fault (F_p) is guaranteed to be detected by the test, if one of the faults in the set occurs.

The sequential machine considered in this study must be in pulse mode operations, since errors caused by timing are not included. Hence the occurance of races, critical or noncritical are automatically avoided. Besides, oscillations are not allowed to occur in the system; otherwise the technique will not work. Consequently, if no ultimate stable state is entered, the fault cannot be detected.

The possible size of the sequential machine which can be analyzed using flow tables as an approach, is comparatively small. This is because the size of the flow table and the sequential fault tree will become too large to handle when the network has a large number of primary inputs and gates. However, the procedure may provide the beginning of a clear approach to fault detection in sequential logic.

B. Suggestions for Further Work

The objective of this section is to present some ideas for more practical approaches to fault detection is sequential logic.

As stated in section 3.3, the path sensitizing concept is fruitful for defining the fault set (F_p). If a proper set of faults is chosen such that the resulting test can sensitize all the paths within the network, all faults in the network will be detected. So far, the technique for sensitizing the path consisting of feedback loops is still unexplored. An immediate approach to the problem is to break those feedback loops and treat the network as if it were combination logic. However, it is hoped that it would not be necessary to break the feedback loops. Instead, it may be possible to use flow tables as an approach to develop a system which is able to consider feedback loops as a section of the path.

Consider the network in Figure 6.1 as an example: The feedback output Y and network output Z, in terms of primary inputs and internal connecting lines, with subscription of the gate numbers are found as follows.

$$Z = (e.f)_3$$
$$= ((d + c)_{23} f_3)$$

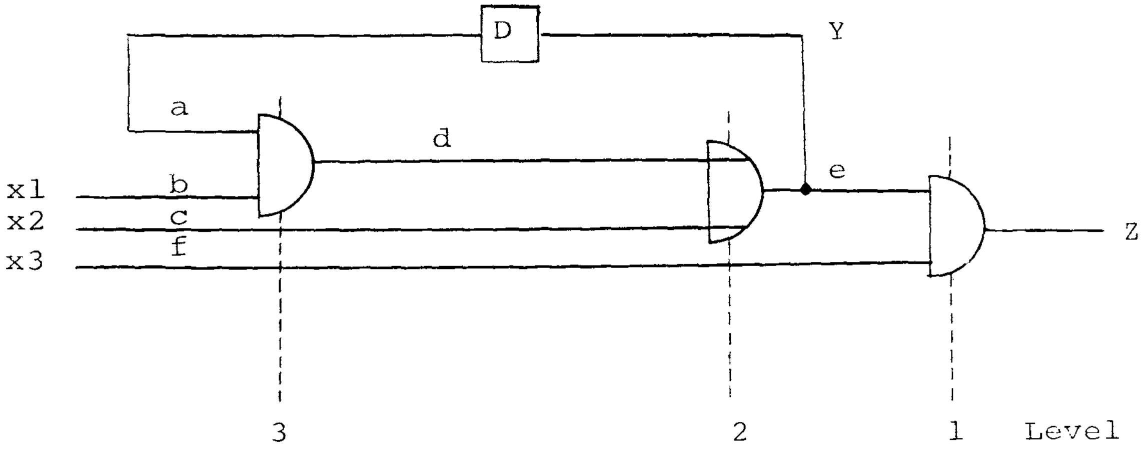


Figure 6.1. A Typical Sequential Circuit

$$Z = ((a \cdot b)_{123})\ f_3 + c_{23}\ f_3$$
$$= a_{123}\ b_{123}\ f_3 + c_{23}\ f_3, \qquad (6.1)$$

$$Y = (d + c)_2$$
$$= (a \cdot b)_{12} + c_2$$
$$= a_{12}\ b_{12} + c_2. \qquad (6.2)$$

The resulting expressions can be called the equivalent normal forms for sequential logic, which is analogous to the enf in combinational logic.

The problem arises when one wishes to generate a set of fault flow tables from the result of the sequential enf. More work on this technique could be fruitful for fault detection procedures.

The procedures which have been described are limited to pulse mode operation. The extension of the system to machines with fundamental mode operations needs to have some kind of timing analysis for the network, such as associating a fix time delay to each gate. Hence, race conditions and oscillations could be considered. The rest of the procedure is found to have no significant difference for machines with level inputs (fundamental mode operation).

VII. BIBLIOGRAPHY

1. Poage, J.F., "Derivation of Optimum Tests to Detect Faults in Combinational Circuits", Proc. 1963 Symp. on Math Theory of Automata, Brooklyn, New York, Polytechniques Press, pp. 483-528.

2. Poage, J.F., "Derivation of Optimum Test Procedures for Switching Circuits", Ph.D. Thesis, Department of Electrical Engineering, Princeton University, January, 1963.

3. Poage, J.F., "Derivation of Optimum Test Sequences for Sequential Machines", Proc. 5th Annual Symposium on Switching Circuits, Theory and Logical Design, October, 1964.

4. Eldred, R.D., "Test Routines Based on Symbolic Logical Statements", ACM Journal, Vol. 6, pp. 33-36, January, 1959.

5. Armstrong, D.B., "On Finding a Near Minimal Set of Fault Detection Tests for Combinational Logic Nets", IEEE Trans. on Electronic Computers, Vol. EC-15, pp. 66-73, February, 1966.

6. Seshu, S., and Freeman, P.H., "The Diagnosis of Asynchronous Sequential Switching System", IRE Trans. on Electronic Computers, Vol. EC-11, No. 4, pp. 459-465, August, 1962.

7. Seshu, S., "On an Improved Diagnosis Program", IEEE Trans. on Electronic Computers (Short Notes), Vol. EC-14, pp. 76-79, February, 1965.

8. Roth, J.P., "A Calculus for Failure Diagnosis", IBM Watson Research Center, Symposium on the Organization of Reliable Automata, February, 1966.

9. Roth, J.P., "Diagnosis of Automata Failures - A Calculus and a Method", IBM Journal, July, 1966.

10. McCluskey, E.J., "Introduction to the Theory of Switching Circuits", McGraw-Hill Book Company, Princeton University Series, 1965.

11. Hellerman, H., "Digital Computer System Principles", McGraw-Hill Book Company, 1967.

VIII. VITA

Chung-tao David Wang was born on December 3, 1943 in China. He received his primary and secondary education in Taipei, Taiwan, China, and attended his undergraduate training at Cheng Kung University in Tainan, Taiwan, China. After his graduation in 1966, he served in the Chinese Army for one year. Then he worked in Thunghi University, Tai-chung, Taiwan, China as a Teaching Assistant of Electrical Engineering until June of 1968.

He has been enrolled in graduate school at the University of Missouri-Rolla and has been a Graduate Assistant in Electrical Engineering since September of 1968.