

THE PATH CHECKING METHOD APPLIED
TO MULTIPLE FAULT ANALYSIS
OF SWITCHING CIRCUITS

Dissertation for the Degree of Ph. D.
MICHIGAN STATE UNIVERSITY
TZE TZONG CHEN
1976



This is to certify that the

thesis entitled

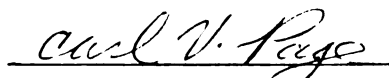
THE PATH CHECKING METHOD APPLIED TO MULTIPLE
FAULT ANALYSIS OF SWITCHING CIRCUITS

presented by

Tze Tzong Chen

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Computer Science


Major professor

Date Dec. 7, 1976

0-7639



ABSTRACT

THE PATH CHECKING METHOD APPLIED TO MULTIPLE FAULT ANALYSIS OF SWITCHING CIRCUITS

By

Tze Tzong Chen

In this thesis, the path checking method is presented for the fault detection and diagnosis of switching circuits, both combinational and sequential. In the case of sequential circuits, signal flows in the circuits and their relevance to the state diagram can be exposed by this method which further facilitate the fault analysis.

Using this method, every path of the switching circuit is checked for its ability to pass both signals, 0 and 1. The same approach is applied to both combinational and sequential circuits, with the sequential circuits treated as an extension of the combinational circuits. Thus, for sequential circuits, the fault analysis is conducted at the circuit level rather than at the state diagram level.

The only restriction imposed on the types of circuits that are considered is that every path of the circuits be sensitizable. Although the EXCLUSIVE OR logical gate is excluded from this study, its exclusion is mainly

for the purpose of a straightforward presentation of the path checking method, rather than any theoretical difficulties involved. Thus the method can easily be extended to include this gate.

Based on this method, four algorithms are presented for fault detection and fault diagnosis of combinational and sequential circuits. For both cases, combinational and sequential, the results obtained from the fault detection algorithms are used as a basis for the fault diagnosis algorithms. The diagnosis process will proceed down to the level of equivalent faults. However, the diagnosis process can terminate at any stage if the level of diagnosis is satisfactory.

The path checking method, as presented in this thesis, can be extended to cover switching circuits which include unsensitizable as well as partially sensitizable paths, i.e., circuits containing redundant circuitry. The extended method will then be applicable to most switching circuits, including LSI circuits, that are encountered in current fault analysis studies.

THE PATH CHECKING METHOD APPLIED TO MULTIPLE
FAULT ANALYSIS OF SWITCHING CIRCUITS

By

Tze Tzong Chen

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science

1976

ACKNOWLEDGMENTS

I am grateful to Dr. Carl V. Page, the chairman of my guidance committee, for his guidance and encouragement during the course of this thesis. My thanks also go to Dr. Richard J. Reid, Dr. Richard C. Dubes, the late Dr. Leo Katz, Dr. Weinberg Bernhard, Dr. Christie G. Enke, and to Dr. Morteza A. Rahimi for serving on my guidance committee and for reviewing this work.

I wish to express my thanks to the Division of Engineering Research for their financial support during the early stage of the writing of this thesis.

Finally, I am deeply indebted to my parents for their patience and encouragement.

TABLE OF CONTENTS

	Page
LIST OF TABLES	vi
LIST OF FIGURES	vii
 CHAPTER	
I. INTRODUCTION AND THE STATE OF THE ART	1
1.1 Definition of Reliable Computing	2
1.2 Forms and Causes of Hardware Failure in Digital Systems	3
1.3 Examples of Logical Faults	3
1.4 Strategies to Circumvent Hardware Failure in Digital Systems	*6
1.4.1 Preventive Type Approaches	6
1.4.2 Corrective Type Approaches	8
1.5 Contribution and Organization of the Thesis	11
II. SWITCHING CIRCUITS AND THEIR FAULT ANALYSIS	14
2.1 Mathematical Models for Switching Circuits	14
2.2 Logical Faults of Switching Circuits	20
2.3 Methods of Test Generation for Combinational Circuits	22
2.3.1 Truth Table Method	23
2.3.2 Path Sensitizing Method	23
2.3.3 The D-Algorithm	27
2.3.4 Poage's Method	35
2.3.5 Boolean Difference Method	37
2.3.6 Chang's CGEM Method	38
2.4 Methods of Test Generation for Sequential Circuits	41

CHAPTER	Page
2.4.1 Poage's Method	42
2.4.2 Other Methods of Fault Detection of Sequential Circuits	46
III. FAULT DETECTION AND DIAGNOSIS OF COM- BINATIONAL CIRCUITS BY THE PATH CHECKING METHOD	48
3.1 Some Network Properties of Com- binational Circuits	49
3.2 Path Sensitization of Combinational Circuits	53
3.2.1 Sensitizing a Path	54
3.2.2 Unsensitizable Paths of Com- binational Circuits	55
3.2.3 Maximal, Non-Maximal and True Sensitization	59
3.3 Faults of Combinational Circuits	69
3.4 Fault Detection of Combinational Circuits	75
3.4.1 The Path Sensitization Table and Maximal Compatible Sets	77
3.4.2 The Path Checking Fault Detection Algorithm	87
3.4.3 An Example	89
3.5 Fault Diagnosis of Combinational Circuits	99
3.5.1 The Diagnosis Approach	99
3.5.2 The Path Checking Fault Diagnosis Algorithm	109
3.5.3 An Example	113
3.6 Characteristics of the Method as Applied to Combinational Circuits	116
IV. FAULT DETECTION AND DIAGNOSIS OF SEQUENTIAL CIRCUITS BY THE PATH CHECKING METHOD	119
4.1 Some Network Properties of Sequential Circuits	121
4.2 Path and Path Sequence Sensitization of Sequential Circuits	128

CHAPTER	Page
4.2.1 Sensitizing a Path and a Path Sequence	128
4.2.2 Unsensitizable Paths in Sequential Circuits	132
4.2.3 Maximal and True Sensitization of Path Sequences	136
4.3 Faults of Sequential Circuits	142
4.4 Positioning the Sequential Circuits in a Desired, Known State	144
4.5 Fault Detection of Sequential Circuits	150
4.5.1 The Path Sequence Sensitization	151
4.5.2 The Path Sequence Checking Fault Detection Algorithm	162
4.5.3 An Example	166
4.6 Fault Diagnosis of Sequential Circuits	179
4.6.1 The Diagnosis Approach	180
4.6.2 The Path Sequence Checking Fault Diagnosis Algorithm	186
4.6.3 An Example	190
4.7 Characteristics of the Method as Applied to Sequential Circuits	196
V. CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK	198
5.1 Conclusions	198
5.2 Suggestions for Future Work	199
BIBLIOGRAPHY	202

LIST OF TABLES

Table	Page
2.1. The flow table of the machine of Figure 2.3	20
2.2. Single propagation D-cubes of the com- binational circuit of Figure 2.5	31
2.3. Flow tables for good and faulty circuits for example of Poage's method	44
2.4. Product tables for example of Poage's method	45
3.1. The PST for Example 3.5	82
4.1. The PSST for Example 4.5	157

LIST OF FIGURES

Figure	Page
1.1. An open diode in an AND circuit	4
1.2. A transistor-diode OR circuit	5
2.1. A basic model for combinational circuits . . .	15
2.2. A basic model for sequential circuits	16
2.3. A Mealy type state diagram	19
2.4. Decision elements	21
2.5. A combinational circuit for the path sensitizing method	25
2.6. An OR gate and its singular cover and propagation D-cubes	28
2.7. A combinational circuit for Chang's method	39
2.8. A sequential circuit for Poage's method	43
3.1. Examples of paths, subpaths, subcircuits, partial circuits, and constituent circuits of a combinational circuit	53
3.2. A simple combinational circuit with fan-out reconvergence	56
3.3. Examples of maximally sensitized and non-maximally sensitized paths	65
3.4. Examples of truly sensitized paths	68
3.5. Examples of multiple and equivalent faults	76
3.6. A combinational circuit with fan-out reconvergence for Example 3.5	80

Figure	Page
3.7. Graphical representation of the diagnosis process	110
4.1. A sequential circuit for Example 4.1	126
4.2. A sequential circuit containing partially unsensitizable path sequences	133
4.3. A sequential circuit with all paths sensitizable	137
4.4. A sequential circuit for Example 4.4	149
4.5. A sequential circuit for Example 4.5	154
5.1. Partially sensitizable paths result from concatenating circuits C_I and C_{II} to form an integrated ^I circuit	201

CHAPTER I

INTRODUCTION AND THE STATE OF THE ART

Digital computers are increasingly being relied upon as an integral part in the day-to-day operations of various sectors of modern societies. The speed and efficiency of computers in processing massive amounts of information have found their applications in almost any branch of modern life, from government agencies to private industries, from weather predictions to patient diagnosis, from research laboratories to factory assembly lines.

A more sophisticated application of computers is found in the field of system monitoring and control where computers are programmed to perform a task independently. A few examples of these might be highway and urban traffic control, satellite flight control, and factory production line control.

With the increasing reliance on digital systems, the problems of their reliability have become a topic of practical concern and have developed to be an area of academic interest. In addition, the current trend toward higher speed, more complexity, and more miniaturization in digital systems introduces more stringent requirements on

reliability standards. The significance of the reliability of digital systems is self-evident and needs no further justification.

1.1 Definition of Reliable Computing

Various definitions [31] have been proposed for the term "reliability" of computers. Even though their phrasings are quite different, they all agree in the more generic sense. A typical definition is given here as "the ability to perform its functions correctly and on schedule even in the presence of errors." In the definition, the emphasis on "on schedule" and "in the presence of error" should not be overlooked. "On schedule" introduces the timing factor which is critical for accurate computing, especially in real-time computing. The presence of errors in a digital system is allowed provided it does not affect the system's ability to perform its functions correctly. This provides for the possibility of introducing additional hardware to enhance the levels of reliability of computing. The definition of reliability given is very general and ambiguous and hence some specific measures of reliability are desirable to allow for practical analysis of reliability problems. Some of the commonly used reliability parameters are lifetime and mean time between average failure-rate.

1.2 Forms and Causes of Hardware Failure in Digital Systems

Errors in the digital system hardware result from a variety of causes. The following causes are responsible for most of the circuit faults in the system [31].

- a. External environmental interference such as heat and humidity.
- b. Internal environmental interference such as noise.
- c. Inappropriate design or construction.
- d. Aging and deterioration of hardware with time.

The circuit faults resulting from these causes can take on various forms. However, the type of faults which are most widely studied and are also the concern of this thesis are called "logical faults." These are the faults which affect the intended logical behavior of circuits. These faults might be permanent or intermittent. Physical failures such as an opened or closed circuit in a component, or a broken wire in connections, or erroneous design or construction might cause permanent faults. On the other hand, noise interference, close design tolerances, aging and deterioration of components are some causes of intermittent faults.

1.3 Examples of Logical Faults

The following are typical examples of hardware failure which are represented by logical stuck-at type faults.

Example 1.1

Figure 1.1 shows a diode AND circuit and its logical representation. The output at y will register 1 volt whenever both inputs x_1 and x_2 have a voltage of 1 volt simultaneously. However, if the diode at x_1 is open, the output would be at 1 volt whenever input x_2 is at 1 volt. This situation is equivalent to x_1 sticking at 1 volt, or logically x_1 sticking at 1 if positive logic is assumed.

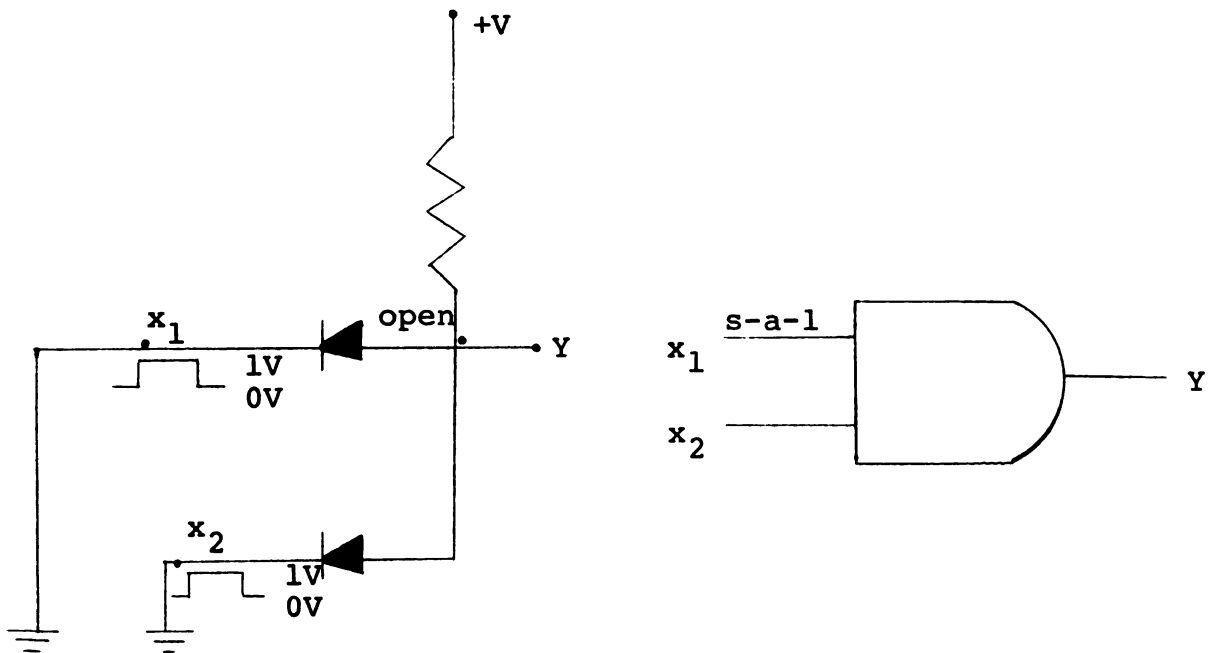


Fig. 1.1.--An open diode in an AND circuit.

Example 1.2

Consider a transistor-diode OR circuit in Figure 1.2. In this circuit, returning the base voltage to $-V_B$ will reverse bias the emitter-base and collector-base junctions and the two diodes and the output voltage V_{CE}

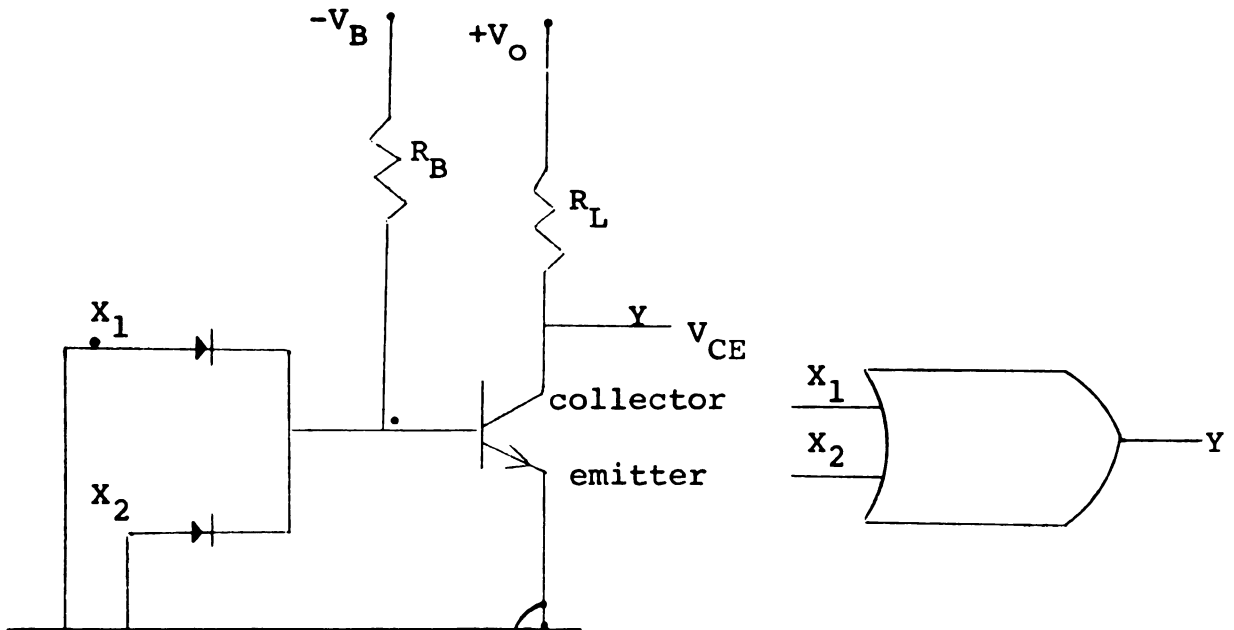


Fig. 1.2.--A transistor-diode OR circuit.

will be approximately equal to the supply voltage V_O . When 1 volt is applied to either or both inputs at x_1 and x_2 , the transistor will turn on and the output voltage V_{CE} will drop to zero indicating the presence of inputs at x_1 or x_2 or both. In this case, negative logical values are adopted for the output. Here an open collector lead will cause the output voltage V_{CE} to be equal to V_O resulting in the output sticking at a 0 logical fault. On the other hand, a short between the collector and emitter leads will create a sticking at 1 fault on the output.

Although not all hardware errors can be translated into stuck-at type logical faults, stuck-at type faults

constitute the main portion of hardware failures encountered in actual experience.

1.4 Strategies to Circumvent Hardware Failure in Digital Systems

There are numerous approaches available to cope with the reliability problems caused by the factors listed in section 1.2. However, they can be categorized into two types of approaches, namely preventive and corrective.

1.4.1 Preventive Type Approaches

This type of approach seeks to prevent the causes of errors from the digital system, or if errors should occur in the system, finds methods to mask them so that normal functions of the system still can be maintained in the presence of these errors.

1. Ensuring proper operational environments and appropriate operational procedures. Suitable computer room temperature, humidity control and use of suitable transmission lines are a few examples of precautions needed to ensure proper performance of the system. Establishment and observance of operational guidelines often would considerably reduce system down times and failure rates.
2. Use more reliable components and design. Higher design and manufacturing specifications to sustain noise levels and power supply fluctuations, heat

and humidity variations are desirable in a reliable system. But this approach is dictated by available technology and economic constraints.

3. Redundancy. This approach uses additional hardware to mask certain circuit faults and thus still maintain correct logical functions in the presence of these faults. Through this method, any desirable degree of reliability can be obtained, however, at correspondingly higher costs.

Considerable research work has been devoted to this approach to study various redundancy techniques. It appears that this approach is widely adopted in systems where higher reliability is desired and repair work is difficult or impossible.

The first important study on this method was done by Von Neumann [40] and published in Automata Studies. In which he proposed two configurations of redundancy; one of which is called "multiplexing." In this configuration more than one wire is used to carry a single signal. The outputs of each logical operation (each signal of which is carried by more than one wire) are woven through a "restoring organ" to enhance the reliability of these outputs for later stages of logical operations. Later Moore and Shanon [28] introduced "quadrupling" method which was applied to relay-type networks. This method was extended by Tryon [39] to develop a method called "quadding"

which applies to logical gate-type circuits with AND, OR, and NOT decision elements. It requires four times as many circuits and masks all single faults. There are other types of redundancy techniques such as the statistical decision approach by Pierce [31] and the coding theory by Armstrong [1].

1.4.2 Corrective Type Approaches

Even with the added protection of redundancy, a computer system cannot be guaranteed to be error-free during its entire life span. A single signal error in all its redundant wires, for example, would be detectable at the outputs of the circuit and thus could cause system malfunction. It is through the continuing testing and correcting approach that a system can be ensured to maintain normal operations at all times.

The corrective procedures in these approaches involve three stages.

1. Applying the developed tests to the system to determine whether it is functioning properly (error detection).
2. Determining the causes of malfunction and locations of faults in case the system is malfunctioning (error diagnosis).
3. Taking corrective measures to restore the system to normal operations.

Of the three stages, the third stage is most straightforward and is a simple task once the initial two stages have been accomplished. While the first and second stages involve complicated circuit functional analyses and testing and have been subjects of research. Numerous methods in error detection and diagnosis of digital circuits have been developed. However, the work is far from complete, especially in the area of sequential circuits.

The common underlying approach of all the methods is designing a test set for the circuit which will detect all faults in the circuit. They differ only in the methods by which these tests are obtained. Examples of these methods will be presented in Chapter II, while a brief discussion now will set the stage for them.

The most simple and straightforward method in error detection and diagnosis is the truth table method. In this method an error is assumed, the truth tables for the fault-free circuit and for the faulty circuit are compared. This comparison will reveal the inputs which detect this fault. This is done for each fault to be investigated. This method is effective only for relatively small circuits, but not for medium or bigger size circuits due to the considerable amount of computations and storage needed.

The path sensitizing method is more effective and faster than the truth table method. It has been widely

investigated and its ideas are extended to develop new methods. The idea is to propagate an assumed error to an output terminal along a sensitized path in a forward-tracing phase, and to trace back from assigned signal points to the inputs to find input tests for the error in the backward-tracing phase. One disadvantage of the path-sensitizing method is that a test may not always be found even though it exists.

Roth's [35] D-algorithm method represents a sophisticated extension of the path sensitizing method. Sophisticated because, instead of treating the error itself as a unit of operation, this method puts all the tests of all possible errors at a logical gate in a "D-Cube." The tests of the circuit are calculated by applying a set of operational rules called "D-Calculus" on these cubes in the "D-Drive" phase (forward-tracing phase) and in the consistency operation (backward-tracing phase). This method has the advantage that it can detect a wider range of faults than stuck-at type faults. The number of D-cubes in a circuit is directly proportional to the number of logical blocks. This fact would impose less demands on storage in computations. Also it is shown that the algorithm will always find the test for a given error, if it exists.

Other algebraic methods by Poage [32], Sellers et al. [36] and Chang [5] for finding the tests for

combinational circuits have also been developed. They all utilize the algebraic properties of the Boolean expressions of the circuits to obtain the tests. As is with most of the previous cases, the big disadvantage of these methods lies in the storage requirements in the computations of these tests.

Relatively few papers on the fault detection and diagnosis of sequential circuits have been published. Hennie [14] proposes a general procedure for synthesizing checking experiments on sequential circuits having a distinguishing sequence. His procedure is based on the logical behavior of the circuit. Other papers by Moore [29], Seshu and Freeman [37], Meyer [25], Hsieh [15] also approach this study at the logical behavior level.

The path checking method, which is an extension of the path sensitizing method, is the topic of this thesis. The advantage of this method is that this method can be applied to both combinational and sequential circuits and thus allows error analysis of sequential circuits at the network level.

1.5 Contribution and Organization of the Thesis

Although a considerable amount of research has been devoted to fault analysis of combinational circuits with respectable results, not very much attention has been given to sequential circuits, even though they constitute the

basic foundation of digital computing circuitry. This results partly from the complexities involved in the analysis of sequential circuits. Also in those published works in this area, the approach used in the diagnosis process has mostly been at the state diagram level of the sequential circuit. Yet, it seems that an approach which studies the fault problems at the circuit level possesses advantages.

This thesis studies fault analysis of combinational and sequential circuits at the circuit level using the path checking method. It appears that the path checking method is a suitable approach for attacking the complexities of signal flows in sequential circuits caused by feedback. Through this method, a systematic view of the complex signal flows in the circuit can be obtained, especially in the case of sequential circuits, where signal flows and their relevance to the state diagram become apparent. Although the combinational case is also treated in the thesis, the sequential case is intended to be the major contribution.

Chapter I defines the problem area and gives a brief discussion of published works in this area. Chapter II presents some background materials, preliminaries and terminology of the field. Examples of published methods are also presented in this chapter. Chapter III will include all the related materials in the detection and diagnosis of combinational circuits by the path checking

method. The sequential circuit case is treated in Chapter IV. Chapter V presents a summary and conclusion of the thesis, discusses unsolved problems and suggests materials for future research.

CHAPTER II

SWITCHING CIRCUITS AND THEIR FAULT ANALYSIS

This chapter is devoted to a review of the related background materials in switching theory and fault analysis of switching circuits. Some representative examples of published results in the field are also included to show various methods used to approach the subject matter. This review, however, is far from complete and interested readers are referred to the original works for more detailed discussions.

2.1 Mathematical Models for Switching Circuits

The term "switching circuits" is used to include both combinational and sequential circuits which together form the basic computing hardware of digital systems. These circuits are usually represented by a mathematical model which describes their functional behavior in Boolean algebra. In the case of sequential circuits, another mathematical model called the sequential machine is used to describe the logical behavior of sequential circuits. The sequential machine can be projected into a schematic

diagram called the state diagram which is a convenient way to express the logical behavior of the sequential circuit [27].

Consider a combinational switching circuit having n inputs x_i , $1 \leq i \leq n$, and m outputs z_j , $1 \leq j \leq m$, as shown in Figure 2.1.

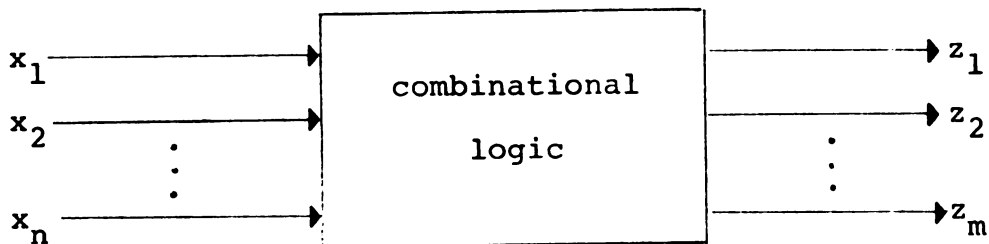


Fig. 2.1.--A basic model for combinational circuits.

Definition 2.1

A switching circuit is a combinational switching circuit (or combinational circuit) if its output variables z_j , $1 \leq j \leq m$, can be expressed as m Boolean functions of its input variables, x_i , $1 \leq i \leq n$,

$$z_j = f_j (x_1, x_2, \dots, x_n) \quad 1 \leq j \leq m.$$

The outputs of a combinational circuit depend entirely on the current inputs and thus are time independent. In a sequential switching circuit (or sequential circuit), however, the outputs are functions not only of present inputs, but also of past inputs as well. These past inputs are stored in a set of delay elements and are

fed back as part of the inputs in the next input-output reaction. These feedback variables are called state variables [27, 23].

Consider a switching circuit with n inputs x_i , $1 \leq i \leq n$, m outputs z_j , $1 \leq j \leq m$, and p state variables y_k , $1 \leq k \leq p$, as shown in Figure 2.2.

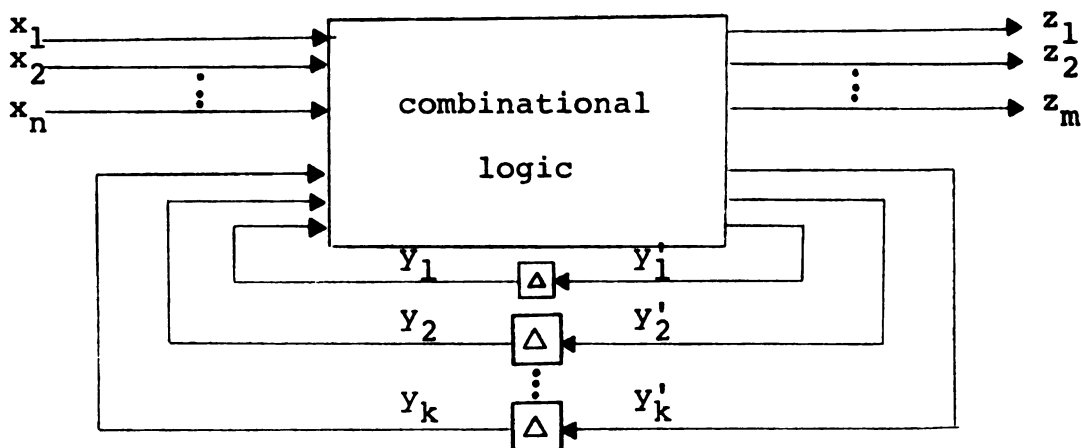


Fig. 2.2.--A basic model for sequential circuits.

Definition 2.2

A switching circuit is a sequential switching circuit (or sequential circuit) if its output and next state variables can be expressed by two sets of Boolean functions. The output variables z_j , $1 \leq j \leq m$, are expressed by m Boolean functions of input and present state variables x_i , $1 \leq i \leq n$, y_k , $1 \leq k \leq p$. The next state variables y'_k , $1 \leq k \leq p$ are expressed by k Boolean functions of input and present state variables.

$$z_j = f_j (x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_k) \quad 1 \leq j \leq m$$

$$y'_k = g_k (x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_k) \quad 1 \leq k \leq p$$

Sequential circuits are classified as either "synchronous" or "asynchronous." Synchronous sequential circuits are characterized by the use of clock pulses to synchronize the signal propagation in the circuit. Asynchronous sequential circuits are not provided with such synchronization control and thus sometimes possess race conditions caused by asynchronized reaction times which causes circuit malfunction.

Another mathematical model used to logically describe a sequential circuit is commonly adopted [27]. Consider the same sequential circuit with input variables $X = \{x_i; 1 \leq i \leq n\}$, output variables $Z = \{z_j; 1 \leq j \leq m\}$, and state variables $Y = \{y_k; 1 \leq k \leq p\}$. Let

$$i = X|_{X=a} \text{ be an input,}$$

$$w = Z|_{Z=b} \text{ be an output,}$$

$$s = Y|_{Y=c} \text{ be a present state and}$$

$$s' = Y|_{Y=c'} \text{ be a next state of the sequential circuit,}$$

where a , b , c , and c' are vectors of signal values which the sequential circuit assumes. The sequential circuit as given by Definition 2.2 can now be interpreted as follows. Given the sequential circuit originally in state s at time

t, it will produce an output w and enter into the next state s' at time t+1 after being given an input i.

Formally, this can be described in a mathematical model called the sequential machine.

Definition 2.3

A sequential machine M is defined as an ordered 5-tuple (Q, I, W, τ, ω) where

1. Q is a finite set of internal states,
2. I is a finite set of inputs,
3. W is a finite set of outputs,
4. τ is a transition mapping of a subset of $Q \times I$ onto a subset of Q,
5. ω is an output mapping of a subset of $Q \times I$ onto a subset of W.

The transition mapping τ defines the next state $s' \in Q$ and the output mapping defines the output $\omega \in W$ of the machine originally in state $s \in Q$ with input $i \in I$.

A sequential machine is connected if given $s_1 \in Q$, there exists $s_2 \in Q$ and an input sequence i^* such that

$$\tau (s_1, i^*) = s_2.$$

A sequential machine is strongly connected if given $s_1, s_2 \in Q$, there exists i^* such that $\tau (s_1, i^*) = s_2$.

A sequential machine can be represented in a graphic form called the state diagram. In a state diagram, every state is represented by a circle, and every transition

mapping represented by a directed line between the corresponding circles with the corresponding input and output shown along the line. Figure 2.3 shows a Mealy type state diagram.

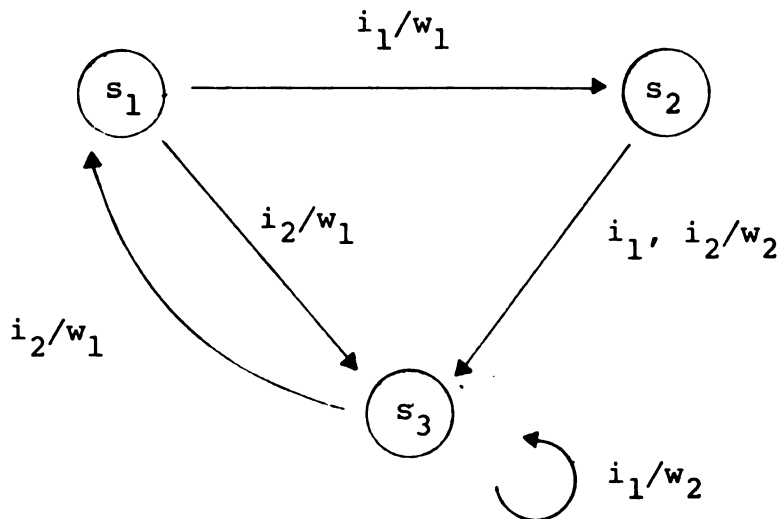


Fig. 2.3.--A Mealy type state diagram.

A sequential machine can also be represented in a tabular form called the flow table [27]. Every state of the machine is represented by a row and every input is represented by a column. The next state and output are then placed in the entry as specified by the transition mapping like s/w . Table 2.1 shows the flow table of the sequential machine of Figure 2.3.

The basic logical elements used to realize the switching circuits are of two types; relay contacts type and decision elements type. The decision elements type is more popular in practical applications and is the only type

Table 2.1.--The flow table of the machine of Figure 2.3.

	i_1	i_2
s_1	s_2/w_1	s_3/w_2
s_2	s_3/w_2	s_3/w_2
s_3	s_3/w_2	s_1/w_1

considered in the thesis. This type generally includes AND, OR, NOT, NAND, and NOR. These elements are enough to realize most of the switching circuits found in real applications. Their schematic notations are shown in Figure 2.4.

2.2 Logical Faults of Switching Circuits

A physical fault is a physical defect of a component or wire in a circuit which may cause the circuit to malfunction. Such a fault is called a logical fault if it affects the proper logical behavior of a switching circuit. Most of the logical faults (or simply faults) can be interpreted as stuck-at type faults in which a line input to or output from a decision element sticks at a fixed signal value and thus is not assuming the proper signal values as expected.

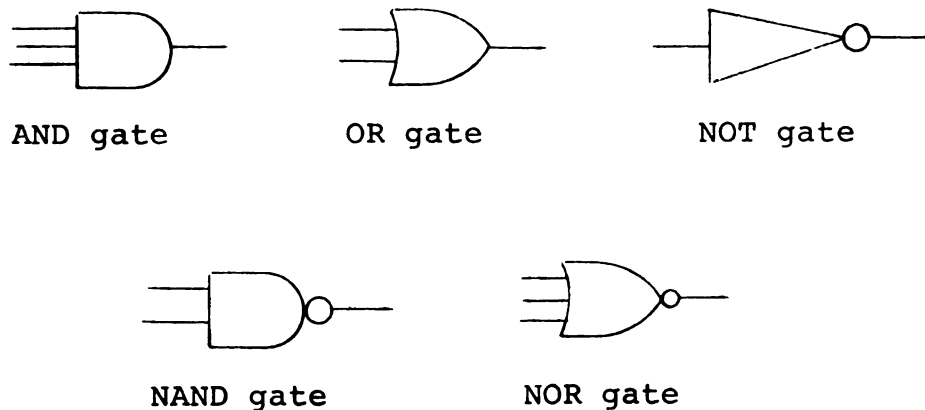


Fig. 2.4.--Decision elements.

The following definitions present some of the important concepts found in many fault analysis studies [4, 28, 31].

Definition 2.4

A test for a fault (or faults) of a combinational circuit is an input vector which, when applied to the combinational circuit, can detect the presence or absence of the fault (or faults) through the comparison of the output vector thus produced with the output vector of the same test applied to a fault-free copy of the combinational circuit. A test for a fault (or faults) of a sequential circuit is similarly defined with the distinction that the test may be a sequence of input vectors.

The same meaning for detection as used in Definition 2.4 applies to the following definitions.

Definition 2.5

A fault is detectable if there exists at least one test which detects this fault. A fault is undetectable if it is not detectable.

Definition 2.6

Two faults are equivalent if there does not exist a test which detects one but not the other.

Definition 2.7

A test for a combinational circuit is a set of input vectors such that:

1. It detects all detectable faults of the combinational circuit,
2. No subset of it can detect all the detectable faults of the combinational circuit.

Definition 2.8

A test for a sequential circuit is a sequence of input-state vectors such that

1. It detects all detectable faults of the sequential circuit,
2. No subset of it detects all detectable faults of the sequential circuit.

2.3 Methods of Test Generation For
Combinational Circuits

Most of the published methods of test generation apply to combinational circuits only. In this section, a

review of some of the best known methods is presented to show the different approaches available for generating tests for combinational circuits.

2.3.1 Truth Table Method

The truth table method is the most basic and straight-forward way to generate tests. Assuming a single fault in a combinational circuit, the truth tables of both the good and the faulty copy of the circuit are compared to obtain the input vectors which produce different output vectors for both circuits. The same procedure is done for every detectable fault. Complete detection tests can then be built from these results.

Though simple and straight-forward, the considerable amount of storage and computations involved make this method unattractive for even circuits of moderate size.

2.3.2 Path Sensitizing Method

The one-dimensional path sensitizing method has been widely studied and applied in developing other methods for fault analysis of combinational circuits. Armstrong [1] of Bell Telephone Laboratories and Stieglitz of IBM are prominently linked with this method even though they never published their works on this method.

The idea is first to arbitrarily find a sequence of gates starting from the site of fault leading to an output edge so that the fault can travel along this path to the

output edge. This portion of the method is called the forward tracing phase. In the forward tracing phase, all the input edges to the gates along this path are assigned signal values such that the output edge at the end of the path responds solely to the signal value at the edge of fault. In this way the path is sensitized.

After setting up the sensitized path in the forward tracing phase, a primary input vector to the circuit must be found which realizes the signal values assigned to the input edges of the gates along the path. This portion of the method is called the backward tracing phase. The input vectors thus found are tests for the fault. An example will best illustrate the procedure.

Example 2.1

Consider the circuit in Figure 2.5 in which edge e_7 sticks at 1 (s-a-1). Suppose the sequence of gates selected is $G_7, G_{12}, G_{13}, G_{15}$ with the corresponding path of edges $p_1 = e_7, e_{10}, e_{12}, e_{13}, e_{15}, z_2$. To sensitize the path and test if e_7 can assume the signal value 0, let

$$e_7 = e_{10} = e_{12} = e_{13} = e_{15} = z_2 = 0,$$

$$x_4 = 1,$$

$$x_5 = 0,$$

$$\text{and } e_8 = 1,$$

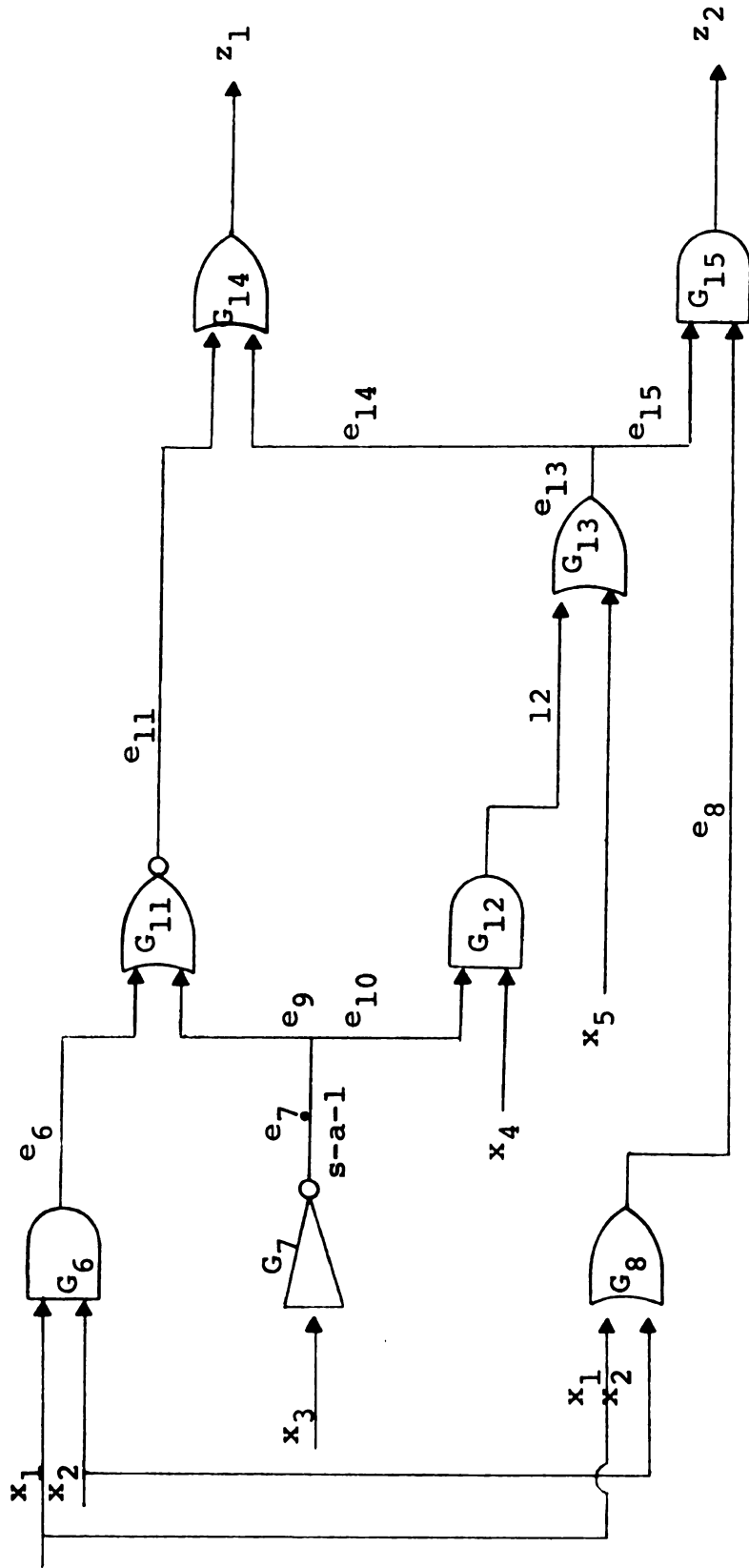


Fig. 2.5.--A combinational circuit for the path sensitizing method.

where x_4 , x_5 , and e_8 are so assigned such that the path is sensitized. This completes the forward tracing phase. In the backward tracing phase, one or more primary input vectors are to be found which yield the signal values assigned thus far. To do this, the components x_1 , x_2 , and x_3 of the input vector have to be decided to satisfy $e_7 = 0$ and $e_8 = 1$. To satisfy $e_7 = 0$ let $x_3 = 1$. To realize $e_8 = 1$, there are three possibilities,

$$x_1 = 1, x_2 = 0;$$

$$x_1 = 0, x_2 = 1;$$

$$\text{or } x_1 = 1, x_2 = 1,$$

which can form three different input vectors. Each of these can be used for sensitizing the chosen path. They are

$$(1, 0, 1, 1, 0) = (x_1, x_2, x_3, x_4, x_5),$$

$$(0, 1, 1, 1, 0), \text{ and}$$

$$(1, 1, 1, 1, 0).$$

Applying one of the three tests to the combinational circuit, one finds

- a. e_7 s-a-l if $z_2 = 1$, or
- b. e_7 does not s-a-l if $z_2 = 0$.

Several observations concerning this method are made as follows.

- a. Since sometimes it is necessary to make a choice out of several possibilities in assigning signal values during the backward-tracing phase, it can happen that inappropriate choice may lead to a conflict with assigned signal values. If this occurs, one must back up and try the other alternatives until the conflict is resolved and a test is found.
- b. The method is not algorithmic. It has been shown that a test cannot be found using this method for a combinational circuit even though it exists.
- c. Finally, this method has been proven very useful in practice and the computer running time and storage requirements are economical.

The failure of the path sensitizing method to find a test for a combinational circuit where there is one is caused by the fact that only one path is sensitized at one time using this method. To overcome this defect, Roth develops a method which sensitizes all possible paths simultaneously from the site of fault to the circuit outputs.

2.3.3 The D-Algorithm

Roth's D-algorithm [35] represents the first algorithmic approach in generating tests for combinational circuits. The formulation of the method is in terms of a calculus called D-calculus for cubical complexes formed

for the logical elements of the circuit. The concepts and terminology as used by him are briefly explained as follows.

a. Singular cover

A singular cover of a gate is a compact truth table of the gate which can be used to describe any logical element. A singular cover of a two-input OR gate is shown in Figure 2.6 in which x represents a "don't care" signal value.

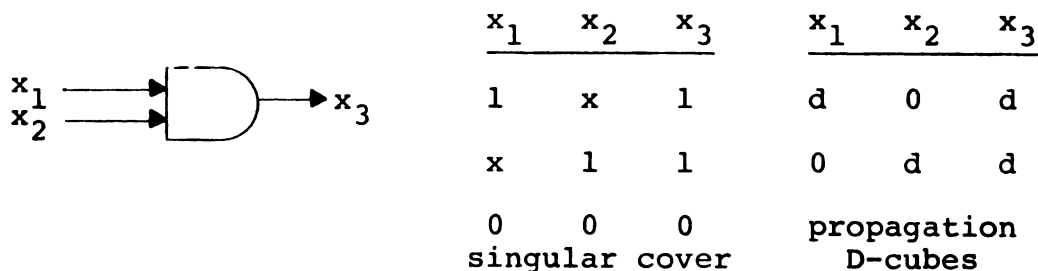


Fig. 2.6.--An OR gate and its singular cover and propagation D-cubes.

b. Propagation D-cube

The propagation D-cubes of a gate contain all of the possible sensitized paths through that gate and are derived from the singular cover of the gate. For the OR gate in Figure 2.6, two propagation D-cubes are obtained as shown in the same figure where d represents the "variable" on that path.

c. Primitive D-cube of a fault

The primitive D-cubes of a fault on a gate are identical to the propagation D-cubes for the gate. However, the primitive D-cubes of a fault represent the tests for that fault.

d. D-Intersection

Sensitized paths starting from the sites of faults to the output edges of a combinational circuit are formed by interconnecting propagation D-cubes using D-intersection operations.

Roth uses a binary operator Ω to define the D-intersection. Some of the binary operations are shown as follows.

$$d \Omega x = d$$

$$1 \Omega x = 1$$

$$x \Omega 0 = 0$$

$$d \Omega d = d$$

$$x \Omega d = d$$

The discussions of the D-algorithm are best given through an example.

Example 2.2

Consider the circuit of Figure 2.5 with fault $f = e_7$ s-a-1. The propagation D-cubes of all the gates in the

circuit are shown in Table 2.2 where d' represents the complement of d .

The D-algorithm is divided into two phases, the D-drive phase and the Consistency phase. In the D-drive phase, all possible sensitized paths for a fault are formed leading from the fault site to the circuit output edges. Input vectors to the circuit compatible with the signal values assigned thus far are then found in the consistency phase.

A. The D-drive phase.

1. Label the primitive D-cube of the fault as test cube 0, tc^0 .

$$tc^0 = \begin{matrix} & 3 & 7 & 9 & 10 \\ & 1 & d' & d' & d' \end{matrix}$$

The variable d' means that the corresponding edge should have a signal value of 0 if the fault is absent and 1 otherwise.

2. Define the D-fanout set of a test cube tc as the set of gates fed by the edges of tc having signal variable d or d' . The D-fanout set df^0 of tc^0 is

$$df^0 = \{11, 12\}.$$

3. The next step is to D-intersect the test cube with the propagation D-cubes of each gate of its D-fanout. The propagation D-cubes of {11} are

Table 2.2.--Single propagation D-cubes of the combinational circuit of Figure 2.5.

edges gates	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	z_1	z_2
6	1	d				d											
6	d	1				d											
7			d				d'		d'	d'							
8	0	d						d									
8	d	0						d									
11						0			d		d'						
11						d			0		d'						
12				1						d		d					
12				d						1		d					
13					0							d	d	d	d		
13					d							0	d	d	d		
14											0			d		d	
14											d			0		d	
15								1							d		d
15								d							1		d

$$tc^{01} = \frac{3 \ 6 \ 7 \ 9 \ 10 \ 11}{1 \ 0 \ d' \ d' \ d' \ d'},$$

$$tc^{02} = \frac{3 \ 4 \ 7 \ 9 \ 10 \ 12}{1 \ 1 \ d' \ d' \ d' \ d'}.$$

4. The D-drive phase continues with successively repeating steps 2 and 3 for each new test cubes obtained at each stage until either the test cube reaches an output edge or ends in a non-output edge. The final test cubes obtained for the example are

$$tc^{012} = \frac{3 \ 6 \ 7 \ 9 \ 10 \ 11 \ 14 \ z_1}{1 \ 0 \ d' \ d' \ d' \ d' \ 0 \ d'}, c$$

$$tc^{0111} = \frac{3 \ 4 \ 5 \ 6 \ 7 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15}{1 \ 1 \ 0 \ 0 \ d' \ d' \ d' \ d' \ d' \ d' \ d' \ d'},$$

$$tc^{0112} = \frac{3 \ 4 \ 6 \ 7 \ 9 \ 10 \ 11 \ 12 \ 14 \ z_1}{1 \ 1 \ 0 \ d' \ d' \ d' \ d' \ d' \ 0 \ d'}, c$$

$$tc^{0221} = \frac{3 \ 4 \ 5 \ 6 \ 7 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15}{1 \ 1 \ 0 \ 0 \ d' \ d' \ d' \ d' \ d' \ d' \ d' \ d'},$$

$$tc^{0222} = \frac{3 \ 4 \ 5 \ 7 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ z_1}{1 \ 1 \ 0 \ d' \ d' \ d' \ 0 \ d' \ d' \ d' \ d' \ d'}, c$$

$$tc^{0223} = \frac{3 \ 4 \ 5 \ 7 \ 8 \ 9 \ 10 \ 12 \ 13 \ 14 \ 15 \ z_2}{1 \ 1 \ 0 \ d' \ 1 \ d' \ d' \ d' \ d' \ d' \ d' \ d'}, c$$

$$tc^{01111} = \frac{3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ z_2}{1 \ 1 \ 0 \ 0 \ d' \ 1 \ d' \ d' \ d' \ d' \ d' \ d' \ d' \ d'}, c$$

Those test cubes reaching an output edge as indicated by a C above represent the sensitized paths between the fault site and an output edge. These test cubes are now ready for the consistency phase.

B. The Consistency phase.

In this phase, one traces along each test cube reaching an output edge, from those edges of known signal values towards the input edges of the circuit in order to determine signal values for those edges not yet assigned. To do this, one starts with D-intersecting the test cube with the singular cover of the gate whose output edge is the highest numbered edge of known signal value (0 or 1) on the test cube. The process is repeated for all other edges of known signal value on the test cube. This is done in descending numerical order until either all the input edges of the circuit have been assigned signal values or a conflict of signal assignments exists, in which case, the path cannot be sensitized.

Considering tc^{012} , one finds, from the circuit of Figure 2.5

$$e_{14} = 0 \rightarrow e_{13} = e_{15} = 0.$$

The D-intersection of tc^{012} with the singular cover of G_{13} results in

$$x_5 = e_{12} = 0.$$

Similarly, the D-intersections of tc^{012} with other singular covers concerned find

$$x_4 = 0,$$

$$\text{and } x_1 = 0, x_2 = 0,$$

$$\text{or } x_1 = 1, x_2 = 0,$$

$$\text{or } x_1 = 0, x_2 = 1.$$

Thus the computations of the consistency phase on tc^{012} result in three tests which sensitize the path

$P_{012} = x_3, e_7, e_9, e_{11}, z_1$. The tests are

$$t_{012}^1 = (0, 0, 1, 0, 0) = (x_1, x_2, x_3, x_4, x_5),$$

$$t_{012}^2 = (0, 1, 1, 0, 0),$$

$$t_{012}^3 = (1, 0, 1, 0, 0).$$

Since e_7 and z_1 have different signal values as can be seen on tc^{012} , the fault f will show up as z_1 assuming the signal value 0 upon applying one of the tests above to the circuit.

Different tests can be obtained from the other test cubes, all of which will detect the same fault.

This concludes the example and the discussions of the procedures of the D-algorithm method. The computations for even a single fault are still quite lengthy as

demonstrated in the example. In actual situations, tests for multiple faults are more frequently demanded. It is possible to apply the D-algorithm on multiple faults by computing tests for each individual fault and then concatenating these tests. However, the total computations needed will be quite substantial.

2.3.4 Poage's Method

Poage [32] developed a mathematical method for generating a minimal set of tests to detect all single and multiple faults. His method applies to both gate and relay contact combinational circuits. He also developed a method for test generation for sequential circuits which will be discussed later in this chapter.

His method is a sort of calculus for faults. For every edge in the combinational circuit, three Boolean variables called fault parameters are defined as follows.

$$j_0 = 1 \quad \text{iff edge } j \quad \text{s-a-0}$$

$$j_1 = 1 \quad \text{iff edge } j \quad \text{s-a-1}$$

$$j_n = 1 \quad \text{iff edge } j \text{ is fault-free}$$

At any given time, only one of the three fault parameters can exist. Then if edge j carries a signal having the Boolean value w , a w^* is defined for the edge as:

$$w^* = w \cdot j_n \vee j_1.$$

The expression means that edge j will have a signal value of 1 either it is fault-free carrying the signal value 1 or it s-a-1. Similarly, a w^{*} is defined for edge j to give the conditions for its signal value to be 0 as

$$w^{*} = w' \cdot j_n \vee j_0.$$

This last expression can also be obtained by complementing w^{*} using the following relations.

$$j_n' = j_0 \vee j_1$$

$$j_1' = j_0 \vee j_n$$

$$j_0' = j_1 \vee j_n$$

$$j_n \cdot j_1 = j_n \cdot j_0 = j_1 \cdot j_0 = 0$$

Now, for a given fault-free combinational circuit realizing some Boolean function f , the corresponding f^{*} and f^{*} can be calculated by successively substituting each edge of the circuit with its $*$ expressions. The formulations of f^{*} and f^{*} can start either from the input edges working towards the output edges or from the output edges working backward towards the input edges. After complete substitutions, the f^{*} and f^{*} will be functions of input variables and fault parameters of the circuit edges. The $*$ expressions for a given set of faults are then obtained by substituting the corresponding fault parameters with the faulty signal values to realize the fault

conditions. Let \bar{f}^* and \bar{f}^{*} be the $*$ expressions of the combinational circuit with the given set of faults, a test t which detects these faults is calculated using the following relation

$$t = (f^* \cdot \bar{f}^{*}) \vee (f^{*} \vee \bar{f}^*).$$

Poage's method is easy to use for relatively small combinational circuits. But for large scale combinational circuits, the computations required in calculating the $*$ expressions would be quite substantial. Also the method is based on the assumption of the existence of tests. Such is not the case in real situations.

2.3.5 Boolean Differen

Sellers et al. [36] developed another mathematical method called the Boolean difference method for generating tests for combinational circuits. For a combinational circuit with an output variable f and input variables x_1, x_2, \dots, x_n , the Boolean difference of the output variable with respect to one of the input variables x_i is defined as

$$\frac{df}{dx_i} = f(x_1, x_2, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) + f(x_1, x_2, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$$

where $+$ is the EXCLUSIVE OR operation. If $\frac{df}{dx_i} = 0$, then f is independent of x_i . On the other hand, if $\frac{df}{dx_i} = 1$, f will be a function of x_i and then

$x_i \frac{df}{dx_i}$ is a test for x_i s-a-0

and $x_i' \frac{df}{dx_i}$ is a test for x_i s-a-1.

Tests for all the input variables and edges of the circuit are then combined to form tests for the entire circuit.

2.3.6 Chang's CGEM Method

Chang [5] developed a mathematical model called the Complete Gate Equivalent Model for combinational circuits. Using this model, a set of Boolean expressions in the form of sums of products are formulated to completely describe the logical functions realized by a combinational circuit. Each product term of the expressions represents a set of paths which he calls a z_{j-1} or z_{j-0} set depending on the signal value at the output when such a set is sensitized. Faults occurring on the set will cause shrinkage or growth of the product term of the set corresponding to a decrease or increase of the number of logical gates covered by the set. To detect the faults, every product term must be investigated for shrinkage or growth. The tests for the combinational circuit are thus obtained from testing every product term for shrinkage or growth. The method is explained using one of his examples as follows.

Example 2.3

The Boolean expressions for the circuit in Figure 2.7 as obtained from its CGEM model are:

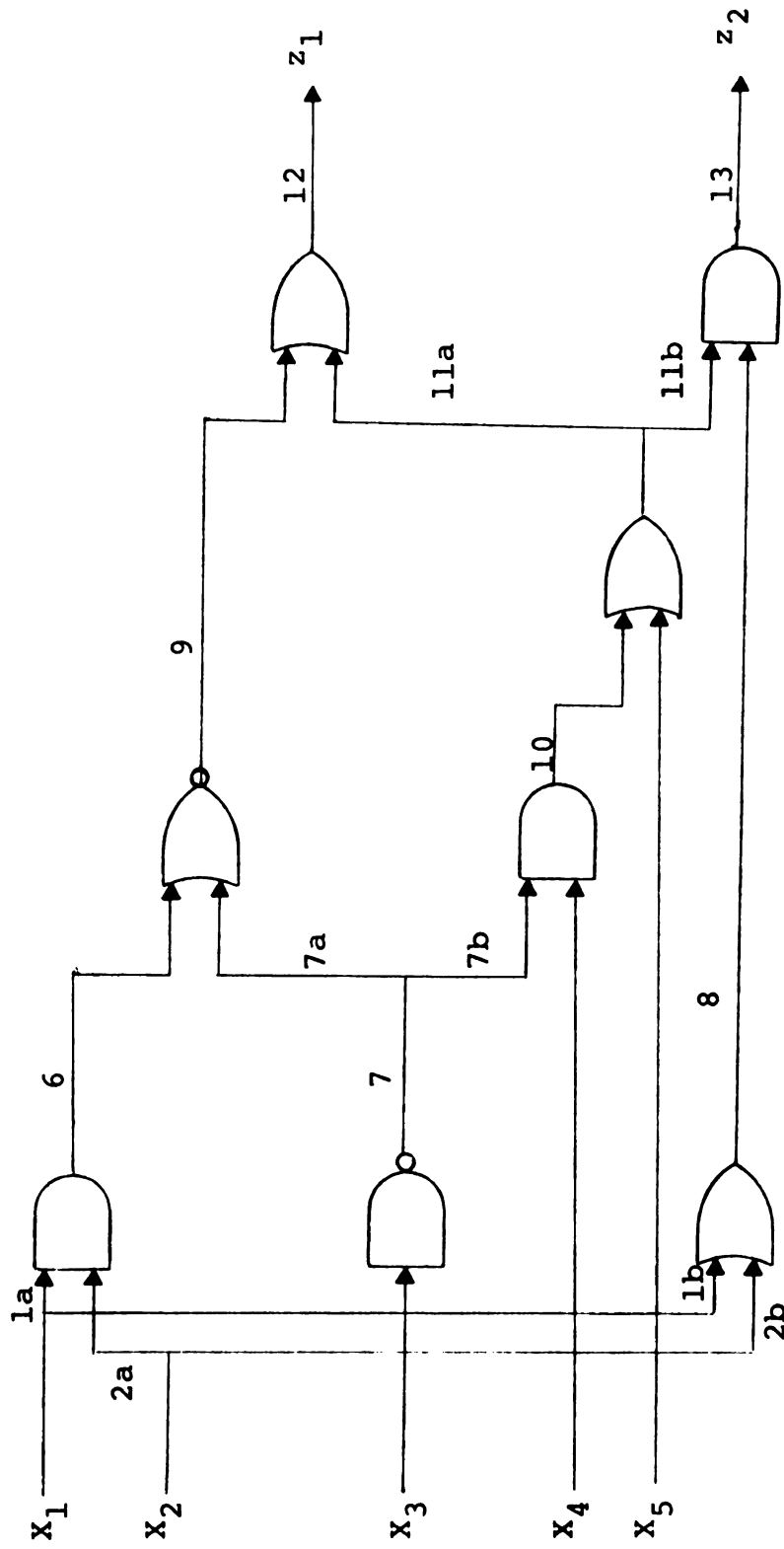


Fig. 2.7.--A combinational circuit for Chang's method.

$$E_1'' = \bar{x}_1 x_3 + \bar{x}_2 x_3 + \bar{x}_3 x_4 + x_5,$$

$$\bar{E}_2'' = \bar{x}_1 \bar{x}_2 + x_3 \bar{x}_5 + \bar{x}_4 \bar{x}_5,$$

where \bar{x}_i means x_i assumes 0 and E_i is the Boolean expression for z_i .

To detect shrinkage on $\bar{x}_1 x_3$ of E_1'' , for example, an input vector, M_1 , can be found by intersecting this term with the complement of every other term of E_1'' .

$$M_1 = \bar{x}_1 x_3 \overline{(\bar{x}_2 x_3)} \overline{(\bar{x}_3 x_4)} \bar{x}_5 = \bar{x}_1 x_2 x_3 \bar{x}_5$$

Since $\bar{x}_1 x_3$ will cause z_1 to assume 1 as indicated by E_1'' assuming 1 under a fault-free situation, the above calculation can be interpreted as M_1 will detect a shrinkage of $\bar{x}_1 x_3$, i.e., stuck-at-0 faults on paths represented by $\bar{x}_1 x_3$, if all other terms assume 0. In effect $\bar{x}_1 x_3$ is sensitized with M_1 .

To detect a growth on $\bar{x}_1 x_3$, i.e., stuck-at-1 faults, every variable of $\bar{x}_1 x_3$ must be investigated since a growth can occur to each variable. The calculations are as follows,

$$M_8 = x_3 \quad \bar{E}_1'' = x_1 x_2 x_3 \bar{x}_5,$$

$$M_9 = \bar{x}_1 \quad \bar{E}_1'' = \bar{x}_1 \bar{x}_3 \bar{x}_4 \bar{x}_5.$$

Thus $\{M_1, M_8, M_9\}$ will detect all faults on paths of $\bar{x}_1 x_3$. Doing this for every term of the expressions and finding a

minimum covering for all the M's terms obtained, a test is formed.

Chang's method is straightforward and his CGEM model is flexible to suit a wide range of combinational circuits. However, separate tests are needed for fault location when using his method.

2.4 Methods of Test Generation for Sequential Circuits

Fault analysis of sequential circuits is a more involved and complicated task than that of combinational circuits. First, tests must be generated for every possible state of the sequential circuit. Hence the amount of computations required to generate the total set of tests for the sequential circuit is proportional to the number of states it possesses. Second, there is the problem of initiating the sequential circuit to some known state before testing can be performed on it. This initialization procedure is in itself a necessary portion of any fault detection technique for sequential circuits.

Most methods known for fault detection of sequential circuits approach the problem at the state level; while relatively few methods treat the problem at the circuit level. The approach of Poage's method can be classified as belonging to the latter category.

2.4.1 Poage's Method

Poage's method [33] for test generation for sequential circuits is essentially an extension of his method for combinational circuits. The feedback loops of the sequential circuit are assumed open to make the circuit combinational. The formulation of the good and faulty circuits are then done using his method for combinational circuits. Also a reset mechanism is assumed to be available to set the circuit to some known starting state. After these initial steps, the flow tables corresponding to the good and faulty circuits are built. The main part of Poage's method involves combining these flow tables into a composite fault table from which optimal test sequences for given faults are derived. The method is demonstrated using an example as follows.

Example 2.4

The circuit for the example is shown in Figure 2.8 with the set of faults T ,

$$T = \{e_2^1, e_3^1, e_4^1, e_5^0\}$$

where e_i^1 and e_i^0 indicate e_i s-a-1, e_i s-a-0 respectively.

Conceptually opening the feedback and applying Poage's method for combinational circuits to the resulting circuit, one gets the output and next state expressions for the good and each faulty circuit as follows.

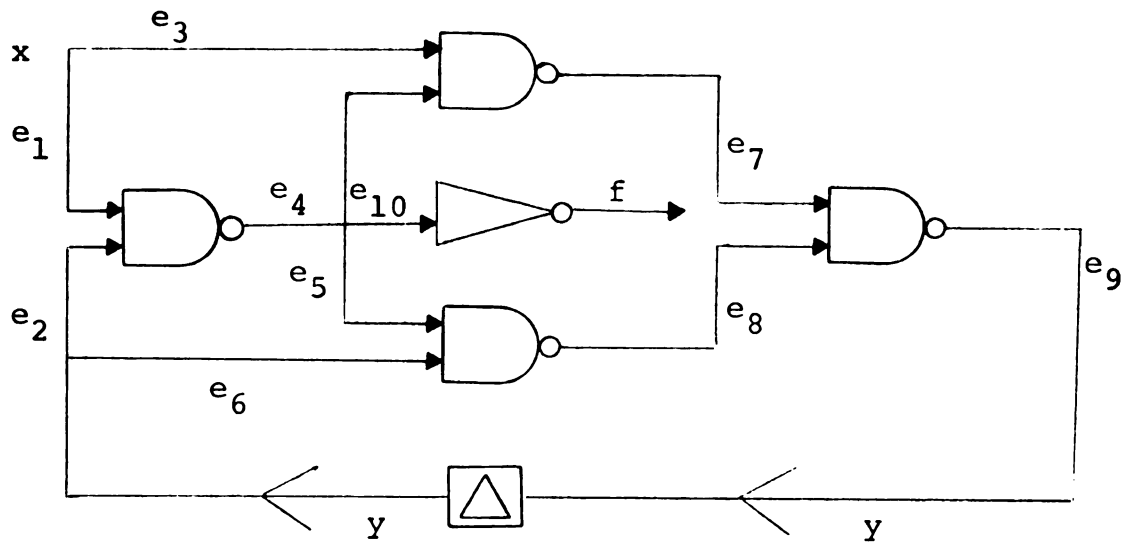


Fig. 2.8.--A sequential circuit for Poage's method.

For good circuit M	$Y = x'y \vee xy'$ $f = xy$
For e_2^1 circuit M_2	$Y_2 = x'y$ $f_2 = x$
For e_3^1 circuit M_3	$Y_3 = x' \vee y'$ $f_3 = xy$
For e_4^1 circuit M_4	$Y_4 = x \vee y$ $f_4 = xy$
For e_5^0 circuit M_5	$Y_5 = xy'$ $f_5 = xy$

Corresponding to these Boolean expressions, the flow tables for the good and faulty circuits are built as shown in Table 2.3.

Table 2.3.--Flow tables for good and faulty circuits for example of Poage's method.

		x	
		0	1
y	T	0/0	1/0
	1	1/0	0/1

		x	
		0	1
y	T ₂	0/0	0/1
	1	1/0	0/1

		x	
		0	1
y	T ₃	1/0	1/0
	1	1/0	0/1

		x	
		0	1
y	T ₄	0/1	1/0
	1	1/1	1/1

		x	
		0	1
y	T ₅	0/0	1/0
	1	0/0	0/1

The main idea of Poage's method is to find input sequences which will produce different outputs for the good and the faulty circuits. To do this, the next step is to form a set of product tables by pairwise concatenating the flow table of the good circuit with that of each of the faulty circuits. This concatenating procedure starts with the common starting state of the sequential circuit. The procedure terminates as soon as a product output with different components appears in the product table.

The product tables thus formed for the example are shown in Table 2.4 where the product outputs with different components are marked in circles.

Test sequences for a given fault are derived from the corresponding product table. The test sequences are

Table 2.4.--Product tables for example of Poage's method.

		x				x		
		0	1			0	1	
YY ₂	T x T ₂							
	0 0	00/00	10/01	YY ₃	T x T ₃			
	0 1	01/00	10/01		0 0	01/00	11/00	
1 1	11/00	00/11	0 1		01/00	10/01		
		x				x		
		0	1			0	1	
YY ₄	T x T ₄							
	0 0	00/00	11/00	YY ₅	T x T ₅			
	1 1	11/00	01/11		0 0	00/00	11/00	
0 1	01/00	11/01	1 1		10/00	00/11		
						1 0	10/00	01/10

simply input sequences which, when applied to the corresponding product table, yield the product output with different components. For example, the test sequences for product table $T \times T_2$ detecting e_2^1 can readily be seen as

$$t_{21} = 0^n, 1 \quad n \geq 0.$$

To find a single optimal test sequence to detect all of the given faults of the sequential circuit, Poage forms the sequential fault table which is simply a composite table of the product tables. The optimal test sequence can then be found from this sequential fault table. Using his technique, the optimal test sequence is found to be

$$t = 0, 1, 0, 1, 1,$$

in which 0, 1 covers $T \times T_2$ and $T \times T_3$,

0, 1, 0, 1 covers $T \times T_2$, $T \times T_3$ and $T \times T_5$ and

0, 1, 0, 1, 1 covers $T \times T_2$, $T \times T_3$ and $T \times T_4$.

Poage's method in finding optimal test sequences provides a closer look at the fault detection problem at the circuit level. It is also straightforward and easy for computer implementation. But his method involves calculations for each individual fault and hence is not very economical from a practical viewpoint.

2.4.2 Other Methods of Fault Detection of Sequential Circuits

Most other methods in designing test sequences for sequential circuits are based on the input-output relations of the corresponding sequential machine. The earliest work using this approach is done by Moore [29] in his gedanken-experiments. In this experiment, he treated the sequential machine under test as a black box on which input sequences are to be applied. Considering the set of sequential machines consisting of the fault-free machine and its failed copies each containing one of the permissible faults, the input-output experiment induces a partition on the set with respect to the outputs obtained. Successive experiments will refine the partition and will finally identify each machine of the set with one of the permissible faults or as fault-free.

Moore's approach was widely applied by other researchers in designing other methods for generating test experiments called checking experiments for sequential machines. In most cases, these methods apply to sequential machines possessing certain characteristics. Hennie [14] introduced the transition checking approach for generating checking experiments for sequential machines which possess a distinguishing sequence (an input sequence used to identify the initial state of the machine) and do not increase the number of states from malfunction. Hsieh [15] developed several procedures for generating checking experiments for sequential machines which do not possess any distinguishing sequences. His methods, however, are based on other types of characteristic sequences; the compound distinguishing sequence, the resolving sequence, the compound resolving sequence, and the simple I/O sequence.

Researchers like Kime [17], Kohavi [18, 19], and others have published their results along the same line.

Some authors have suggested addition of extra external hardware, modification of the sequential machine or embedding the machine in a larger equivalent machine [17, 18, 22] to create certain desirable characteristics for designing checking experiments. Detailed discussions of these methods will not be presented in the paper.

CHAPTER III

FAULT DETECTION AND DIAGNOSIS OF
COMBINATIONAL CIRCUITS BY THE
PATH CHECKING METHOD

The path checking method approaches the task of fault detection and diagnosis of switching circuits by checking all the paths of the circuits to ensure that they propagate the signals correctly. Through this process, a set of detection tests are obtained which can be used for detection purposes. The detection phase will locate the given faults to some general areas of the circuits. The diagnosis phase will then proceed based on these results.

In this chapter, the path checking method as used in the fault detection and diagnosis of combinational circuits is presented. Two algorithms for fault detection and diagnosis are formulated using this method.

The type of combinational circuits treated using this method possesses the following characteristics:

- a. Multiple inputs and multiple outputs.
- b. Path fan-out reconvergence allowed.
- c. All paths being sensitizable.

Thus the only restriction imposed on the type of combinational circuits is that every path of the circuit be sensitizable. This, however, will not place any limitations on the range of real applications of this method, since unsensitizable paths represent redundancy and their removal will not affect the logical behavior of the circuits.

Faults treated are both single and multiple, but are of permanent stuck-at type. The type of logical elements considered are as discussed in Chapter II.

3.1 Some Network Properties of Combinational Circuits

A brief study of some of the network structural properties of the combinational circuits as they relate to the path checking method will be presented here. This discussion will help one visualize the combinational circuits from the "path" point of view.

A combinational circuit can be viewed as a set of interconnected paths such that the leading and trailing edges of the paths form the input and output edges of the circuit. For a single-output combinational circuit all of its paths will eventually converge to the single trailing edge. All fan-out paths residing in it will have to meet one another before reaching the trailing edge. But in the case of a multiple-output combinational circuit, fan-out paths need not meet one another, since they may lead to different output edges of the circuit. The multiple-output

combinational circuit is characterized by the existence of fan-outs branching to different output edges of the circuit. Such fan-outs do not reconverge. Other fan-outs may lead to the same output edges thus forming fan-out reconvergence.

Consider a combinational circuit C with n input edges x_i , $1 \leq i \leq n$, m output edges z_j , $1 \leq j \leq m$ and the set of q labelled edges $E = \{e_k, 1 \leq k \leq q\}$. The edges are so labelled such that the output edge of any logical element will have a label with higher order than that of the label assigned to any input edge of the logical element. Also fan-out edges (edges which are branches of a common edge) are assigned different labels. A path and a subpath are defined as follows.

Definition 3.1

- A path p of C is a sequence of edges $p = l_1, l_2, \dots, l_p$ such that
1. l_1 , the leading edge of the path, is an input edge of C ,
 2. l_p , the trailing edge of the path, is an output edge of C , and
 3. l_i and l_{i+1} , $1 \leq i \leq p-1$, are either respectively an input edge and the output edge of the same logical element or l_{i+1} is a fan-out edge of l_i .

Definition 3.2

A subpath p_s of C is a section of a path p such that p_s includes either the leading edge or the trailing edge of p .

The number of paths in a multiple-output combinational circuit is of interest in estimating the amount of computation needed in designing tests. For a multiple-output combinational circuit possibly with fan-out reconvergence, the number of paths is found in the following theorem.

Theorem 3.1

The number of paths in a multiple-output combinational circuit C with n input edges, m output edges, k fan-outs and l logical elements are

$$n_p = n + \sum_{i=1}^k (n_{F_i} - 1) = m + \sum_{i=1}^l (n_{G_i} - 1)$$

where n_{F_i} is the number of branches at fan-out F_i and n_{G_i} is the number of edges feeding logical element G_i .

Proof: To compute the number of paths from the number of input edges, since every path must start with an input edge, the number of paths is the sum of this value plus extra paths introduced with fan-out branches. For a fan-out F branching into n_F paths, $n_F - 1$ will be the extra paths introduced with this fan-out. To obtain the number of paths from the number of output edges,

interchange the roles of inputs and outputs, with each logical element playing the role of a fan-out. Q.E.D.

The concepts of subcircuits, partial circuits, and constituent circuits of a combinational circuit are presented here as they will prove useful in later discussions.

Definition 3.3

A subcircuit within a combinational circuit includes all the subpaths of the circuit leading to an edge (need not be a trailing edge) of the circuit.

Definition 3.4

A partial circuit within a combinational circuit includes a set of paths sharing an output edge of the combinational circuit as their common trailing edge.

Definition 3.5

A constituent circuit of a multiple-output combinational circuit includes all the paths sharing an output edge of the combinational circuit as their common trailing edge.

Example 3.1

These definitions are illustrated in the combinational circuit C of Figure 3.1. Paths include p_1 and p_2 ; subpaths include p_{s1} and p_{s2} ; P is the partial circuit formed from p_1 and p_2 ; while C_2 is the subcircuit formed

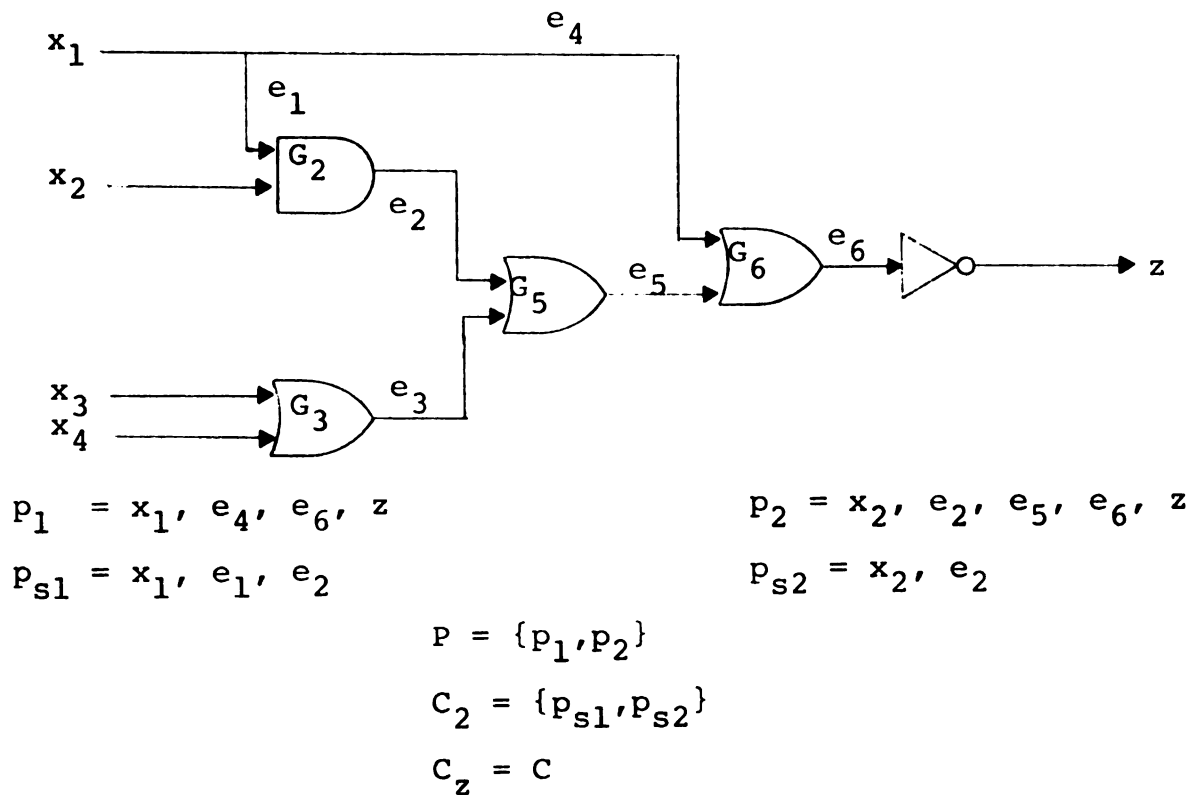


Fig. 3.1.--Examples of paths, subpaths, subcircuits, partial circuits, and constituent circuits of a combinational circuit.

from p_{s1} and p_{s2} with the subscript 2 indicating that the edge e_2 is its output edge and the constituent circuit C_2 corresponds to the combinational circuit C itself.

3.2 Path Sensitization of Combinational Circuits

If a path of a combinational circuit is to be checked for correct signal propagation, it must be sensitized. However, an input vector sensitizing a path needs not necessarily guarantee its sensitization due to the existence of faults. This section discusses the problems of absolute path sensitization and designs input vectors

which ensure absolute path sensitization for combinational circuits.

3.2.1 Sensitizing a Path

When a signal propagates through a logical element, it may assume one of two possible modes, sensitizing or dominant. For example, the signal value 0 would propagate through an OR gate sensitizingly, since for it to pass through this gate, all other edges incident on it should take on the sensitizing signal 0 too. On the other hand, 1 can propagate through the OR gate regardless of other signals incident on the gate, i.e., dominantly. Thus the OR gate will be a sensitizing gate to 0 and a dominant gate to 1. This is true of all the logical elements considered in this thesis. To sensitize a path, all the edges incident on it must assume the corresponding sensitizing signals and thus the signals propagating through the path are sensitized.

Definition 3.6

A path is said to be sensitized if a change of the signal value at its leading edge alone will cause a change of the signal value at its trailing edge.

A signal can propagate through a path even though the path is not sensitized. A signal is said to propagate through a path if, traveling down the path, it maintains the same signal value when going through a gate of even

parity, and changes its signal value when going through a gate of odd parity. Also the path is said to propagate the signal. A signal propagating through a path is said to be of 0-type for the path if it shows up as 0 at the trailing edge; it is of 1-type if it shows up as 1. However, for a path to be tested, it still has to be sensitized.

Definition 3.7

A path p is said to pass an s-type signal, indicated by p^s , if it is sensitized and its s-type signal propagates through it. P is said to be tested to pass s if it is sensitized and its s-type signal is applied to its leading edge. It is tested if both signals are applied to its leading edge. And it is checked if it passes both signals.

Even though combinational circuits considered in this thesis are required to possess sensitizable paths only, generally, combinational circuits may include unsensitizable paths. Unsensitizable paths may also result from faults in the circuits.

3.2.2 Unsensitizable Paths of Combinational Circuits

The unsensitizability of a path in passing a signal is caused by the fact that some edges incident to the path are not free to assume the corresponding sensitizing values because of fan-out reconvergence. If a path is unsensitizable in passing a signal s , then there is at least

an edge on the path at which an \bar{s} -type fault (assuming \bar{s} at the trailing edge when detected) is not detectable.

Consider, for example, the simple combinational circuit in Figure 3.2 with three paths, $p_1 = x_1, e_5, z$, $p_2 = x_2, e_3, e_5, z$ and $p_3 = x_2, e_4, z$. In the circuit, p_1 is unsensitizable in passing both signals; p_2 is unsensitizable in passing 0 and p_3 is completely sensitizable. To sensitize p_1 , e_3 should assume 0 and e_4 assume 1 causing an impossible situation for x_2 . For p_2 to pass 0, x_2 should assume 0 which makes the sensitizing edge e_4 impossible to assume 1. Thus x_1 s-a-0 and e_5 s-a-1 are undetectable. Actually the circuit is a unitary circuit and e_5 represents redundancy and hence can be removed from the circuit resulting in the simple circuit consisting of p_1 only.

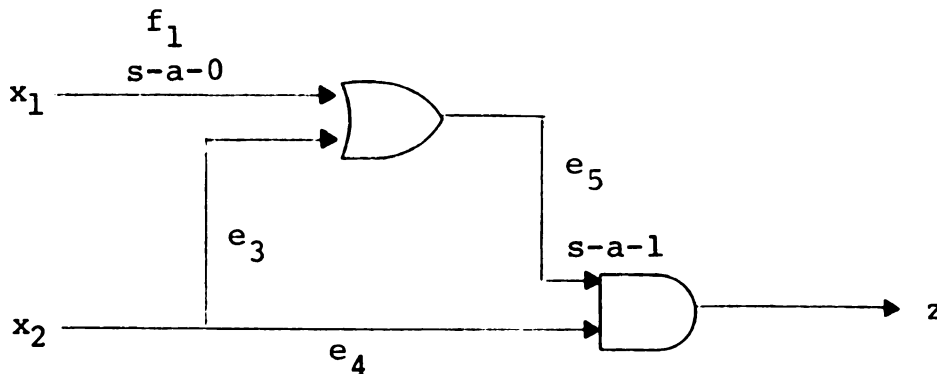


Fig. 3.2.--A simple combinational circuit with fan-out reconvergence.

Definition 3.8

A path is s-sensitizable if it can be tested to pass s , otherwise it is not s -sensitizable or s-unsensitizable. It is sensitizable if it can be tested to pass both signals, partially sensitizable if it can be tested to pass only one signal and cannot be tested to pass the other.

Partially sensitizable and unsensitizable paths pose extreme difficulties in fault analysis of the combinational circuits. First, they may be caused sensitizable through the presence of other faults and second, a group of faults consisting of both detectable and undetectable faults may become undetectable. For example, e_4 s-a-1 in Figure 3.2 causes p_1 sensitizable and e_4 s-a-1 concurrently with x_1 s-a-0 is not detectable.

The elimination of partially sensitizable and unsensitizable paths from a combinational circuit will not alter the original logical behavior of the circuit. This is proved in the following theorem.

Theorem 3.2

For every combinational circuit in which not every path is sensitizable, there exists a combinational circuit with all paths sensitizable which realizes the same logical behavior.

Proof: Given a combinational circuit C_1 with partially sensitizable or unsensitizable paths, another

combinational circuit C_2 with all paths sensitizable will be built from C_1 by removing all paths which are s -unsensitizable ($s = 0$ or 1). An unsensitizable path is also s -unsensitizable. If a path p of C_1 is s -unsensitizable with an input vector i , then there exists at least one gate on p at which at least one of the incident edges assumes the corresponding dominant value. Let such a gate which is closest to the trailing edge of p be called the front gate for i . Since p is s -unsensitizable, there exists a front gate for every s -type input vector (i.e., an input vector which assigns the s -type signal to the leading edge of p) to p . Of all the front gates, let G be the gate closest to the leading edge of p . Then the edge e of p feeding G is never sensitized for all the s -type input vectors to p and its s - a - \bar{s} fault f is not detectable. If \bar{s} is dominant to G , then f will propagate down p to some edge e' feeding G' and the fault f' : e' s - a - s' is equivalent to f . Since f (f') is undetectable, one can assign the sensitizing value \bar{s} (s') to e (e') permanently, or equivalently, remove e (e') from C , without affecting the logical behavior of C_1 . Edges on the subpath between the leading edge of p and e (e') which are not shared by other paths can also be removed. Sequentially removing all the s -unsensitizable paths from C_1 , one arrives at the equivalent circuit C_2 with all paths sensitizable. Q.E.D.

3.2.3 Maximal, Non-Maximal and True Sensitization

If a path of a combinational circuit is sensitized with an input vector, it may not actually be sensitized under a fault detecting situation. This is due to the fact that a faulty subpath may be used as a sensitizing subpath and thus the path in question is not actually sensitized with the input vector. To prevent such situations from affecting a detection test, one should seek input vectors which assure maximal or true sensitization for every path of the circuit. To do this an incident sub-circuit needs to be maximally assigned if possible.

Definition 3.9

A subcircuit is said to be maximally assigned a signal value s or simply s -maximally assigned, if, with an input vector, s propagates through every subpath of it. Otherwise it is said to be s -nonmaximally assigned.

If a subcircuit C_i is s -maximally assigned, then its output edge e_i will assume the signal value s unless e_i sticks at \bar{s} . This is proved in the following theorem.

Theorem 3.3

If the output edge e_i of a subcircuit C_i does not stick at \bar{s} , the input vector i , with which C_i is s -maximally assigned, will assign s to e_i .

Proof: If e_i does not stick at \bar{s} , then there exists at least one subpath of C_i which is free of \bar{s} -type faults.

With the input vector i , s will propagate through this subpath and e_i will assume s . Q.E.D.

For example, with an input vector $i = (x_1, x_2, x_3, x_4) = (0, 0, 0, 0)$ applied to the circuit of Figure 3.1, the output edge e_5 of subcircuit C_5 will assume 0 unless e_5 s-a-1. Either x_1 and x_2 s-a-1 or x_3 and x_4 s-a-1 will result in e_5 s-a-1. Otherwise i will cause e_5 to assume 0.

When a path p is sensitized to pass s , some of its incident edges are also simultaneously s -sensitized if they assume the sensitizing value s . Their correctly assuming the sensitizing values in sensitizing p is of no concern because being sensitized, the values they assume are detected at the trailing edge of p . However, the other edges called strictly sensitizing edges, which are not simultaneously sensitized with p , must be guaranteed to assume the correct sensitizing values in order for p to be actually sensitized. To do this the corresponding sensitizing subcircuits, called strictly sensitizing subcircuits must be maximally assigned if possible.

Definition 3.10

A path p is said to be s-maximally sensitized, if on being sensitized to pass s , every strictly sensitizing subcircuit is \bar{s} -maximally assigned. If this is possible, it is said to be s-maximally sensitizable. And it is said to be maximally sensitizable if this is possible for both

values of s . If p is not s -maximally sensitized, it is said to be s -non-maximally sensitized.

Maximal sensitization means actual sensitization and a path will always be maximally sensitized if possible. However, a path might not be maximally sensitizable if the combinational circuit possesses fan-out reconvergence. A path is maximally sensitizable if it doesn't involve fan-out reconvergence.

Theorem 3.4

A path p with the trailing edge z of a combinational circuit C is maximally sensitizable if no fan-out paths with the same trailing edge reconverge before reaching z .

Proof: Since every subpath of C which is incident on p is part of a path with z as the trailing edge, hence it does not involve fan-out reconvergence. Thus each such subpath is free to assume its corresponding sensitizing value for the sensitization of p . This implies that every subcircuit incident on p can be maximally assigned its corresponding sensitizing value. Q.E.D.

From this theorem, one finds that every path of a fan-out reconvergence free combinational circuit is maximally sensitizable.

Corollary 2.1

Every path of a combinational circuit free of fan-out reconvergence is maximally sensitizable.

Proof: Since the combinational circuit is free of fan-out reconvergence, from the above theorem, every path is thus maximally sensitizable. Q.E.D.

The path sensitization of combinational circuits with fan-out reconvergence is more involved than was found in the case of combinational circuits free of fan-out reconvergence. The complication arises from the fact that not every path of the circuit is maximally sensitizable. For those paths which are not s-maximally sensitizable, one seeks, instead of s-maximal sensitization, to make sure that every non-maximally assigned strictly sensitizing subcircuit is capable of providing the right sensitizing value.

For a path p which is s-non-maximally sensitized. consider one of its strictly sensitizing, non-maximally assigned subcircuit C_i with the sensitizing value \bar{s} . Since it is not \bar{s} -maximally assigned, its subpaths can be divided into two sets for this particular input vector: those through them \bar{s} propagates to the output edge e_i of C_i and the others \bar{s} does not propagate through. The first set are responsible for providing the sensitizing value \bar{s} and at least one subpath of this set should be able to pass \bar{s} in order to make e_i assume \bar{s} . Thus, in this connection,

one seeks a path p_i which contains such a subpath of C_i and that p_i pass \bar{s} .

Definition 3.11

Given p s -non-maximally sensitized with an input vector i and with one of its strictly sensitizing, non-maximally assigned subcircuits C_i , the output edge of which is e_i . Let \bar{s} propagate through a subpath p_{si} to e_i providing the sensitizing value \bar{s} in the sensitization of p . A path p' containing p_{si} is called a critical sensitizing path of p with i . The signal value \bar{s} is the critical sensitizing value of p' . p_{si} is called a critical sensitizing subpath, with the critical sensitizing value \bar{s} .

Note that, with respect to a particular strictly sensitizing, non-maximally assigned subcircuit C_i , there are possibly more than one critical sensitizing paths. Each of these passing the sensitizing value \bar{s} would be sufficient in sensitizing p , with respect to C_i . One can say that p depends on one of these paths, with respect to C_i , in its sensitizing with an input vector i . It will be expressed in a notation as follows.

$$i: p^s / \{ (C_i, p_i^s); (C_j, p_j^s); \dots \}$$

The notation means given p non-maximally sensitized to pass s with an input vector i , p depends, for actual sensitization, on p_i passing s_i , with respect to C_i , and on p_j passing s_j with respect to C_j , etc. If there is only

one such input vector i , or where i is understood, i will be omitted. C_i and C_j will also be omitted if confusion will not hence occur.

These definitions are illustrated in the following example.

Example 3.2

In the combinational circuit with fan-out reconvergence as shown in Figure 3.3, $p_1 = x_1, e_{18}, z$ is 1-maximally sensitized, $p_6 = x_4, e_8, e_{13}, e_{17}, e_{19}, z$ and $p_8 = x_5, e_{15}, e_{17}, e_{19}, z$ are 1-non-maximally sensitized with the input vector indicated in the figure, where the subcircuit C_{16} is not maximally assigned. $p_3 = x_2, e_{11}, e_{16}, e_{19}, z$ is the only critical sensitizing path for C_{16} .

To actually sensitize an s -non-maximally sensitizable path p with some input vector i , then, with each of its strictly sensitizing non-maximally assigned subcircuit, at least one of the critical sensitizing paths, p' , must be \bar{s} -maximally sensitized directly or indirectly with some other input vectors. Indirectly, because, like p , it may be \bar{s} -non-maximally sensitizable. Then the same process for searching actual sensitization must be done for p' . However the search is guaranteed to terminate with positive results as will be proved later. Thus the actual sensitization of p to pass s must be accompanied by a set of input vectors which, directly or indirectly, maximally

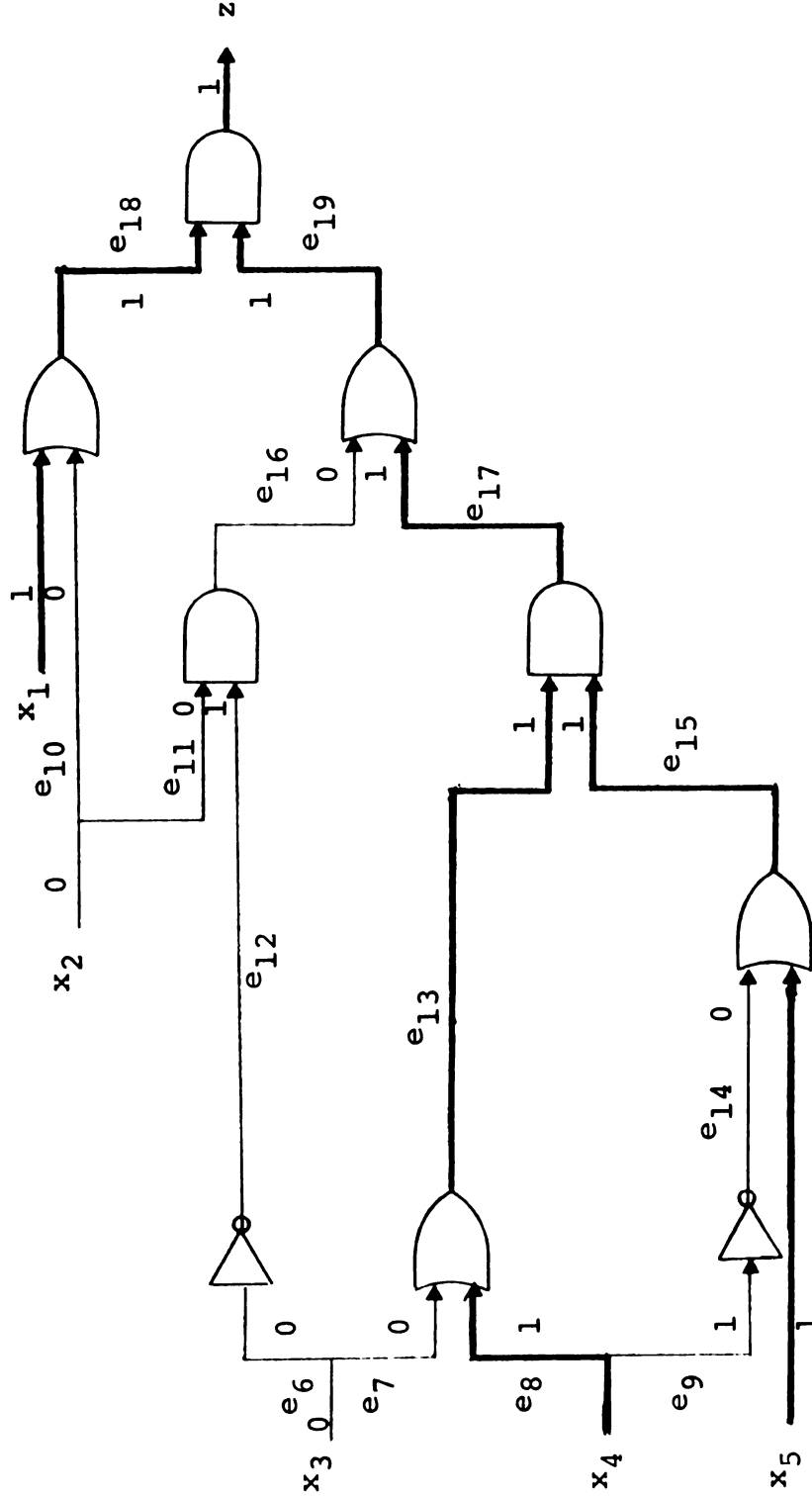


Fig. 3.3.--Examples of maximally sensitized and non-maximally sensitized paths.

sensitize at least one critical sensitizing path passing \bar{s} for each non-maximally assigned strictly sensitizing sub-circuit.

Definition 3.12

Given

$$i: p^S / \{(C_1, p_1); (C_2, p_2); \dots; (C_n, p_n)\}$$

and a set of input vectors I , where each p_i , $1 \leq i \leq n$, is a critical sensitizing path with respect to the strictly sensitizing subcircuit C_i with the critical sensitizing value \bar{s} , p is said to be s-maximally sensitized of first order with i , within I if each p_i is \bar{s} -maximally sensitized with some $i' \in I$. Similarly p is s-maximally sensitized of second order if each p_i is either \bar{s} -maximally sensitized or \bar{s} -maximally sensitized of first order within I . And p is s-maximally sensitized of nth order if each p_i is either \bar{s} -maximally sensitized of $(n-1)$ th order or less.

With the above definitions, true sensitization of a path can be conveniently defined as follows.

Definition 3.13

A path p is said to be s-truly sensitized with an input vector i , within a set of input vectors I if there exists a number n such that p is s-maximally sensitized of n th order with i , within I .

Example 3.3

To illustrate these definitions, consider the circuit of Figure 3.3 with two of the input vectors i_1 shown in Figure 3.3 and i_2 in Figure 3.4. Paths p_6 and p_8 (see example 3.2) are 1-non-maximally sensitized in Figure 3.3 with the critical sensitizing path $p_3 = x_2, e_{11}, e_{16}, e_{19}, z$. But p_3 again is 0-non-maximally sensitized in Figure 3.4 with the critical sensitizing path $p_1 = x_1, e_{18}, z$. Finally p_1 is maximally sensitized in Figure 3.3. Thus p_6, p_8 and p_3 are all truly sensitized within i_1 and i_2 .

The question of whether the search for true sensitization will always end with positive results will have to be answered during path sensitization. In the case of combinational circuits in which every path is sensitizable, the answer is yes. Because otherwise such circuits containing faults on the paths not truly sensitizable only would be still proved fault-free. But it can be proved that if a combinational circuit has every path sensitizable, then at least one fault is detectable if it contains at least one fault.

Theorem 3.5

In a combinational circuit with every path sensitizable, every s -non-maximally sensitizable path is s -truly sensitizable.

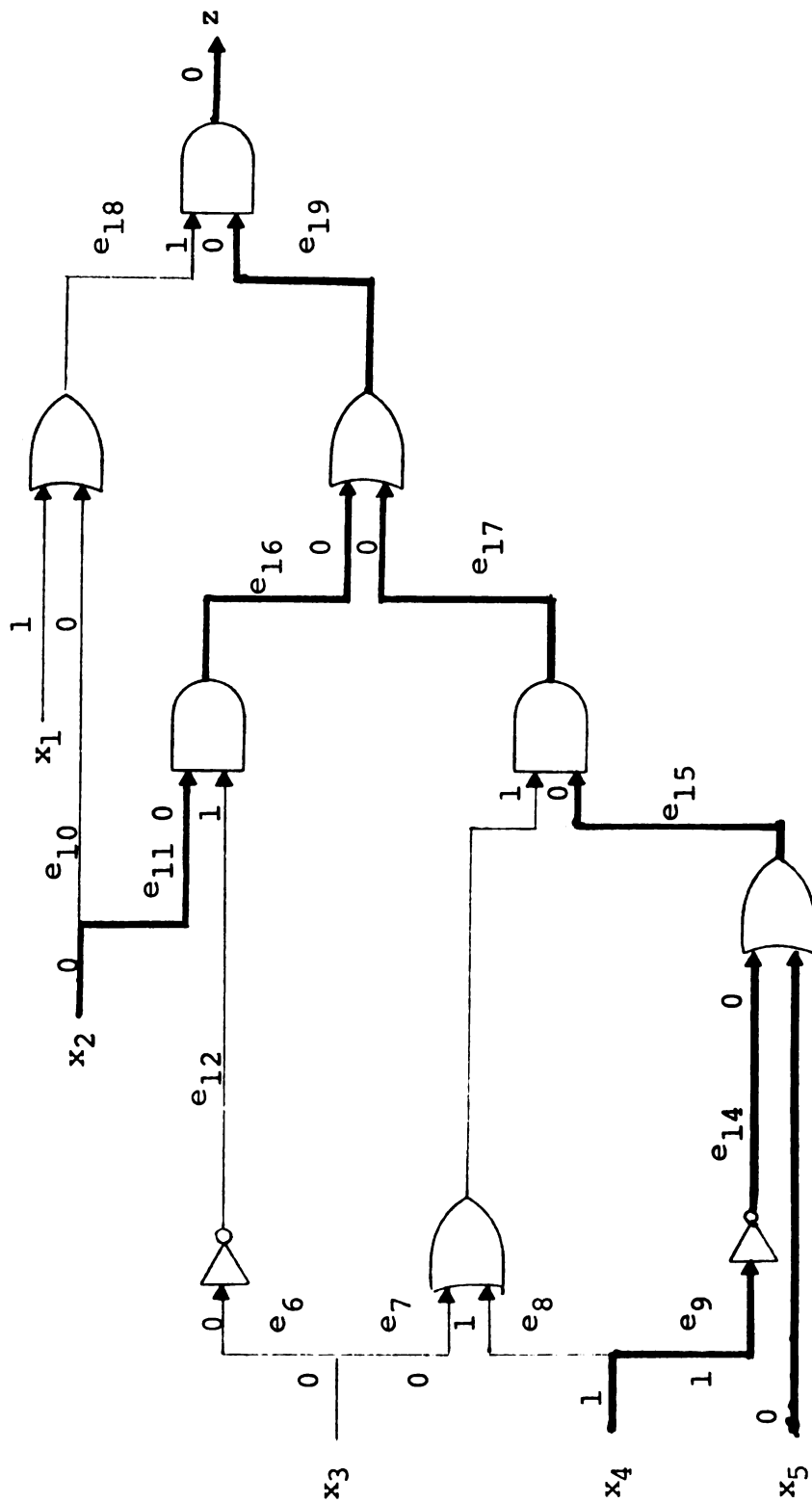


Fig. 3.4.--Examples of truly sensitized paths.

Proof: First of all, if a combinational circuit with every path sensitizable, then at least one fault is detectable if it contains at least one fault. Otherwise it would mean that the reduced circuit resulting from the original circuit with the faults is equivalent to the original circuit. This in turn would imply that there were redundant edges in the original combinational circuit. However, a combinational circuit with every path sensitizable is free of redundant edges.

Now if there existed a path which was not s -truly sensitizable, then each of its critical sensitizing path would not be \bar{s} -truly sensitizable and so on. Thus there would exist a set of paths such that for each of the set, there would be nothing to depend on in its s' -sensitization for some s' . Hence its s' -sensitization cannot be guaranteed. Because of this, there could exist a set of faults on this set alone in the whole circuit which would be undetectable. This is in clear contradiction to the sensitizability assumption. Q.E.D.

3.3 Faults of Combinational Circuits

In this section, faults in the combinational circuit will be analyzed and interpreted in terms of path checking. Faults interact in the combinational circuit to form undetectable faults, multiple faults and equivalent faults. These phenomena will also be studied from the viewpoint of path checking. It appears that the total

fault structure of the switching circuit can be clearly interpreted through this viewpoint.

Faults propagate as do signals. A fault $f: e \text{ s-a-s}$ with e feeding a gate G is said to be sensitizing to G if s is sensitizing to G , it is dominant to G if s is dominant to G . Thus faults will propagate through gates to which they are dominant and stop at gates where they are sensitizing. A fault is a 0-type fault on a path p if it would cause the trailing edge of p to assume 0 when sensitized and 1-type otherwise. So if a path is checked, then, being able to pass both signals, it is free of stuck-at faults.

Theorem 3.6

If a path is actually sensitized and tested to pass both signals, it is free of stuck-at faults iff it is checked.

Proof: Given a path passing both signals, and suppose there are some stuck-at faults on it. Consider the fault closest to the trailing edge. Since the path is sensitized, the fault will appear at the trailing edge regardless of the signal applied to the leading edge contracting to the assumption. Thus the path is free of stuck-at faults. Conversely, if the path is free of stuck-at faults, then since it is guaranteed to be sensitized, it should pass both signals. Q.E.D.

If every path of a combinational circuit is individually determined for its ability in passing both signals, then all the detectable faults in the circuit can be located down to equivalent faults. This will be discussed in the diagnosis process later. Faults interact to form detectable, undetectable, single, multiple and equivalent faults within a given fault picture.

Definition 3.14

A fault picture is the set of all faults verified and detectable that occur at any given stage of fault analysis of a combinational circuit under test.

A single fault is a fault which is by itself detectable and can be exactly located. An s-type single fault will occur at the output edge of an s-type maximal faulty subcircuit.

Definition 3.15

A subcircuit C_i of a combinational circuit C is an s-type faulty subcircuit if every path of C containing a subpath of C_i is such that either it is \bar{s} -unsensitizable within the given fault picture (due to other faults) or that the \bar{s} signal cannot propagate through it. C_i is an s-type maximal faulty subcircuit if there does not exist another s-type faulty subcircuit which includes C_i .

The following theorem finds the locations of single faults.

Theorem 3.7

A fault of a combinational circuit C is an s -type single fault iff it occurs at the output edge of an s -type maximal faulty subcircuit C_i with which there exists at least a path of C containing a subpath of C_i which is \bar{s} -sensitizable.

Proof: Consider an s -type fault f on some edge e_i which is the output edge of a subcircuit C_i . Since e_i sticks at the s -type fault, so every path of C containing a subpath of C_i must be either \bar{s} -non-sensitizable or \bar{s} -sensitizable but cannot pass the signal \bar{s} . Since f is detectable, then at least one such path p is \bar{s} -sensitizable. Also there exists at least a path p' of C containing the subpath between f and the trailing edge of p such that p' passes \bar{s} . Hence f doesn't propagate down to p' . On the other hand, given C_i , since e_i cannot assume the \bar{s} type signal whenever it is \bar{s} -sensitized, it sticks at the s -type fault. And since it is \bar{s} -sensitizable along p , its sticking at the s -type fault is detectable. Q.E.D.

When every path containing a subpath of an s -type maximal faulty subcircuit is \bar{s} -non-sensitizable, within a given fault picture, then the fault at the output edge of the subcircuit is undetectable. This is due to the fact that faults occur on all the critical sensitizing paths of every path along which the fault in question is being \bar{s} -sensitized. This phenomenon causes undetectable faults,

multiple faults and equivalent faults during the diagnosis process and will be called fault interaction. A multiple fault consists of a set of individual faults which are detectable only as a group.

Definition 3.16

A multiple fault is a set of faults such that these faults are detectable only as a set.

First consider undetectable faults not within a maximal faulty subcircuit. These faults will be called uncovered undetectable faults. Faults within a maximal faulty subcircuit are not detectable and are not of concern. These faults also occur at the output edges of the maximal faulty subcircuits. But in this case, not a path containing a subpath of the maximal faulty subcircuits is \bar{s} -sensitizable.

Theorem 3.8

A fault of a combinational circuit C is an s -type uncovered undetectable fault iff it occurs at the output edge of an s -type maximal faulty subcircuit C_i with which every path of C containing a subpath of C_i is \bar{s} -unsensitizable.

Proof: Let f be an s -type uncovered undetectable fault which occurs on some edge e_i which is the output edge of some subcircuit C_i . Since f does not propagate down along some path which contains it, C_i must be maximal. And

since not one path containing a subpath of C_i is \bar{s} -sensitizable, f is not detectable. On the other hand, given C_i and the fact that every path of C containing a subpath of C_i is \bar{s} -non-sensitizable, then every s -type fault in C_i is undetectable which is equivalent to e_i sticking at the s -type fault which is undetectable. Q.E.D.

Uncovered undetectable faults might interact among themselves to form multiple faults. An uncovered undetectable fault by itself (without other uncovered undetectable faults) is undetectable. However, such a fault, say of s -type, might become detectable if at least one path containing a subpath of its s -type maximal faulty subcircuit becomes \bar{s} -sensitizable and hence the fault in question becomes \bar{s} -sensitized along the path. This will occur if some other uncovered undetectable faults propagate down along the corresponding sensitizing paths, due to another faults, to provide the correct sensitizing values. In other words, an uncovered undetectable fault becomes sensitized with other uncovered undetectable faults. If a set of such uncovered undetectable faults are mutually sensitizing, then they form a multiple fault. Notice that each individual fault cannot be individually sensitized.

Theorem 3.9

A multiple fault f is formed from a set of mutually sensitizing uncovered undetectable faults.

Proof: Since each individual fault of f is undetectable, and detectable as a set, it must be an uncovered undetectable fault. And since it is undetectable without the other individual faults of f , it must be sensitized with the rest of f . Q.E.D.

Finally, addition or removal of undetectable faults, both covered or uncovered but uninteracting, from a given set of faults can create an equivalent set of faults. This is obvious since the added or removed faults are undetectable, no input vector will detect the change.

Example 3.4

Consider the circuit of Figure 3.5 in which f_1 , f_3 , f_7 and f_{11} are detectable faults. Given these faults, f_{18} and f_9 are each by itself undetectable and are detectable together. Also f_{17} is undetectable. Addition of it to $\{f_7, f_{11}\}$ results in z_2 s-a-l. Thus $\{f_7, f_{11}\}$ is equivalent to f_{z_2} : z_2 s-a-l.

3.4 Fault Detection of Combinational Circuits

In this section, the design of detection tests for combinational circuits based on the path checking method is discussed and an algorithm for forming such tests is presented. The main purpose of a detection test is to determine whether a circuit is fault-free or faulty, but not to locate the faults which is the objective of a diagnosis process. However, one of the characteristics of the

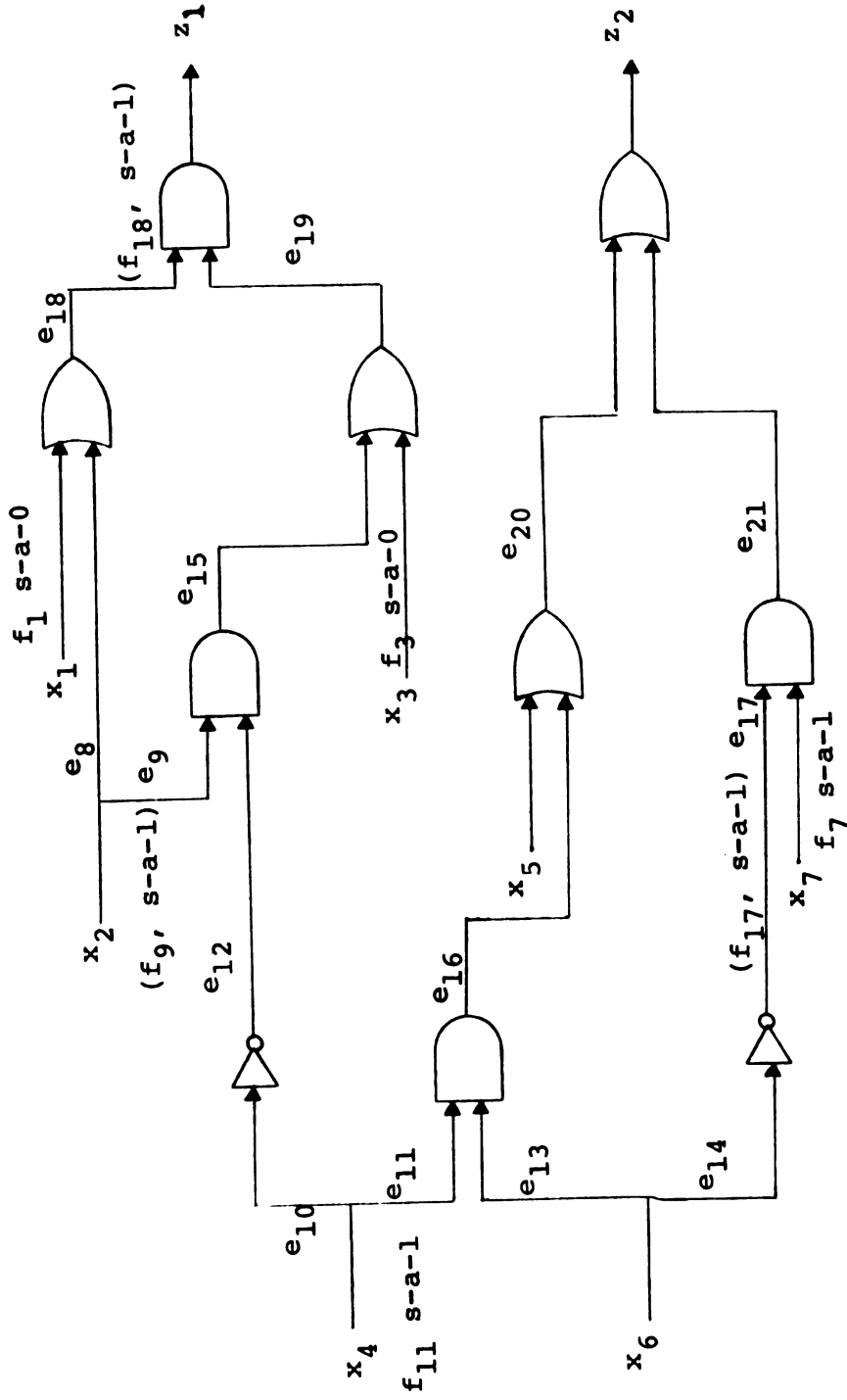


Fig. 3.5.--Examples of multiple and equivalent faults.

path checking method is that the results of a detection test will indicate the approximate areas of faults of a combinational circuit if it is determined to be faulty. Thus the results of a detection test will be used for the diagnosis algorithm discussed in the next section.

The basic principle behind the path checking method is to make certain that every path can pass both signal values 0 and 1 in order to prove that it is fault-free. That principle is used here in synthesizing detection tests. But since the presence of faults can prevent actual sensitization, hence the detection tests should be designed such that every path of the combinational circuit is s-maximally sensitized or, if impossible, s-truly sensitized to ensure its actual sensitization. It will be proved later in the section that a combinational circuit is fault-free iff every path of it is checked.

3.4.1 The Path Sensitization Table and Maximal Compatible Sets

The main point in designing detection tests for a combinational circuit is that each of the input vectors used in the tests should s-maximally sensitize the paths being sensitized with the input vector. For those paths not s-maximally sensitizable, then their s-true sensitization will be required within the input vectors included in the tests. Such an input vector is called a maximal compatible input vector or mci. For a given

combinational circuit, a set of mci's can be found from which the detection tests will be formed. Such a set is called the maximal compatible set, MCS, of the combinational circuit and it will be shown to be unique for a given combinational circuit.

The MCS is built from the path sensitization table, PST. The PST is a table which includes all the input vectors sensitizing the paths of a combinational circuit.

The Construction of the PST

Consider a multiple-output combinational circuit with fan-out reconvergence C with n_i input edges, n_o output edges and a total of n_e edges including input and output edges with each edge uniquely labelled. Fan-out edges succeeding the fan-out points are differently labelled. Let there be n_p paths.

$$P = \{p_i; 1 \leq i \leq n_p\}$$

The PST consists of input vectors sensitizing every path of P and is built as follows:

1. There are n_e columns corresponding to the n_e edges of C.
2. For every path p, there are two rows to test p pass 0 and 1 denoted by p^0 , p^1 respectively.
3. For any given row corresponding to some p passing i (0 or 1) p^i , assign the corresponding sensitizing

values to edges incident on p and i to edges on p with even parity and \bar{i} to edges on p with odd parity between the edges and the trailing edge of p .

4. Enter signal values to edges which can be determined through fan-outs and NOT gates, from edges already assigned values. Also the input edges to OR gates whose output edges having been assigned 0 and the input edges to AND gates whose output edges having been assigned 1 can also be determined.
5. Rows corresponding to the same constituent circuit are grouped together (for convenience).

Except for the arrangement of the rows within it, the PST thus formed is unique. Notice that only those edges on the paths, incident on the paths and edges related to them are assigned values.

Example 3.5

Consider a 7-input 2-output combinational circuit C , as in Figure 3.6, with a total of 23 edges and 10 paths.

$$P = \{p_i; 1 \leq i \leq 10\}$$

where

$$p_1 = x_1, e_{18}, z$$

$$p_2 = x_2, e_8, e_{18}, z$$

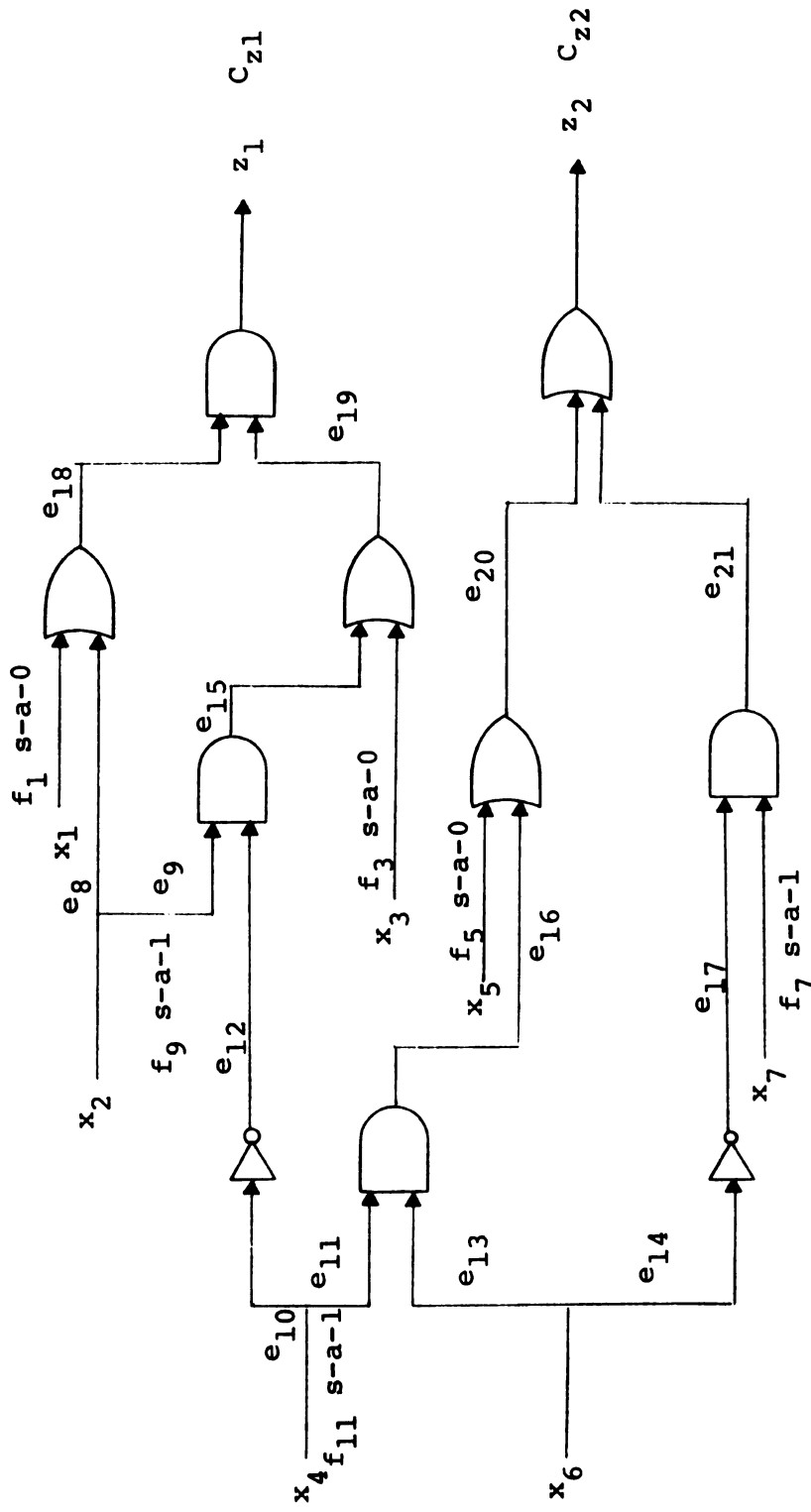


Fig. 3.6.--A combinational circuit with fan-out reconvergence for Example 3.5.

$$p_3 = x_2, e_9, e_{15}, e_{19}, z$$

$$p_4 = x_4, e_{10}, e_{12}, e_{15}, e_{19}, z$$

$$p_5 = x_3, e_{19}, z_1$$

$$p_6 = x_5, e_{20}, z_2$$

$$p_7 = x_4, e_{11}, e_{16}, e_{20}, z_2$$

$$p_8 = x_6, e_{13}, e_{16}, e_{20}, z_2$$

$$p_9 = x_6, e_{14}, e_{17}, e_{21}, z_2$$

$$p_{10} = x_7, e_{21}, z_2$$

There are two constituent circuits C_{z1} and C_{z2} each consisting of five paths.

$$C_{z1} = \{p_1, p_2, p_3, p_4, p_5\}$$

$$C_{z2} = \{p_6, p_7, p_8, p_9, p_{10}\}$$

The PST will be a table of 20 rows, 2 for each path, and 23 columns corresponding to the 23 edges of C. The table is shown in Table 3.1. Each row is labelled p_i^x indicating path p_i is sensitized to pass x . Three types of entries have been assigned values; all other entries still remain blank. Those entries on the path sensitized are assigned values according to their parities. For example, x_1 , e_{18} and z_1 of p_1^0 assume 0 so that p_1 is sensitized to pass 0. The second type are those which are incident on the

Table 3.1.--The PST for Example 3.5.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}	e_{15}	e_{16}	e_{17}	e_{18}	e_{19}	e_{20}	e_{21}	z_1	z_2
P_1^0	0	0						0	0									0	1			0	
P_1^1	1	0						0	0									1	1			1	
P_2^0	0	0						0	0									0	1			0	
P_2^1	0	1						1	1									1	1			1	
P_3^0		0	0	0				0	0	0	0	1			0			1	0			0	
P_3^1		1	0					1	1			1			1			1	1			1	
P_4^0		1	0	1				1	1	1	1	0			0			1	0			0	
P_4^1		1	0	0				1	1	0	0	1			1			1	1			1	
P_5^0			0												0			1	0			0	
P_5^1			1												0			1	1			1	
P_6^0					0											0					0	0	0
P_6^1					1											0					1	0	1
P_7^0				0	0	1				0	0	1	1	1		0	0				0	0	0
P_7^1				1	0	1				1	1	0	1	1		1	0				1	0	1
P_8^0				1	0	0				1	1	0	0	0		0	1				0	0	0
P_8^1				1	0	1				1	1	0	1	1		1	0				1	0	1
P_9^0					0	1	1						1	1		0	0				0	0	0
P_9^1					0	0	1						0	0		0	1				0	1	1
P_{10}^0					0	0	0						0	0		0	1				0	0	0
P_{10}^1					0	0	1						0	0		0	1				0	1	1

path; these entries should assume the corresponding sensitizing values. For p_1 , there are e_8 and e_{19} assuming 0 and 1 respectively to sensitize p_1 . The third type are those entries whose values can now be determined through fan-outs. For p_1^0 , x_2 and e_9 will of course assume the same value as e_8 and are so assigned.

The Forming of the MCS

The PST lists the minimum requirements for the paths to pass the signals. A set of the rows of the PST might be compatible to each other and thus can be combined to form a new row and the individual paths are then simultaneously sensitized to pass the signals.

Definition 3.17

A set of rows of the PST are compatible to one another if there are no contradictory entries on the same column (i.e., 0 and 1 not on the same column). A maximal compatible set of rows is a compatible set such that it is not a subset of another compatible set. A maximal compatible row is formed by combining all the rows of a maximal compatible set of rows through intersecting the entries of the same column (the intersection of x and blank is x).

The first step in forming the MCS is to form the maximal compatible rows for each constituent circuit of C . For example, continuing with the preceding example, p_1^0

and p_2^0 can be combined to form a maximal compatible row for C_{z1} and p_9^1 and p_{10}^1 can be combined for C_{z2} .

$$\begin{aligned} v_{11} &= \{p_1^0, p_2^0\} \\ &= (0,0,\dots,0,0,\dots,0,1,\dots,0) \\ v_{25} &= \{p_9^1, p_{10}^1\} \\ &= (\dots,0,0,1,\dots,0,0,\dots,0,1,\dots,0,1,\dots,1) \end{aligned}$$

Each of the maximal compatible rows v_i represents a set of paths simultaneously sensitized. Hence v_i represents some partial circuit P_i being sensitized to pass some s . The blank entries of v_i represent the strictly sensitizing subcircuits C_i 's incident on P_i and they will be maximally assigned, if possible.

In the second step, the blank entries are free to assume any values except for those entries on fan-out paths. For entries not on fan-out subpaths within each C_i , assign the same type of signal values as the output edges e_i of C_i . Then all related entries which can be determined thus far through fan-outs, NOT gates, output edges of OR gates assuming 0 and output edges of AND gates assuming 1, are thus determined. Fan-out entries are finally assigned. Each fan-out will assume all the values without causing contradiction of assignments, i.e., both 0 and 1 if possible. In this step all the entries of v_i corresponding to the constituent circuit C_{zi} will be determined.

Thus, for example, v_{11} and v_{25} become

$$v_{11} = (0,0,1,0,,,,0,0,0,0,1,,,0,,,0,1,,,0),$$

$$v_{25} = (,,,0,0,0,1,,,0,0,1,0,0,,0,1,,,0,1,,1).$$

In the final step, all the maximal compatible rows are again combined to form maximal compatible rows for the entire circuit C. Those entries undetermined thus far are unrelated in even maximal sensitization and are free to assume any values. Their corresponding input edges are assigned, say 0 and thus all the entries are determined. The rows now obtained are maximal compatible input vectors and the whole set is the MCS. For example, combining v_{11} , v_{25} one finds the mci

$$\begin{aligned} i_4 &= \{v_{11}, v_{25}\} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7) \\ &= (0,0,1,0,0,0,1). \end{aligned}$$

A mci is non-maximally assigned if its partial circuit is not maximally sensitized. With each mci non-maximally assigned, those paths not maximally sensitized need a set of critical sensitizing paths passing the corresponding critical sensitizing values for the paths to be truly sensitized.

For the example, the dependence set of i_4 is

$$i_4: p_1^0, p_2^0 / \{p_5^1\}.$$

Thus every s -maximally sensitizable path of C can be thus sensitized with an input vector of the MCS. Also every s -truly sensitizable path of C is thus sensitized with an input vector of C and within the MCS.

Theorem 3.10

Every s -maximally sensitizable path is s -maximally sensitized with some input vector of the MCS.

Proof: Let p_i^s be s -maximally sensitizable in the combinational circuit C . Then in being sensitized to pass s , every strictly sensitizing subcircuit C_i incident on p can be \bar{s} -maximally assigned. Since this is possible, it is done in forming the MCS. Q.E.D.

Theorem 3.11

Every s -non-maximally sensitizable path is s -truly sensitizable within the MCS.

Proof: Let p_i^s not s -maximally sensitizable in the combinational circuit C . Then p_i is s -truly sensitizable within all the input vectors of C . Thus there exists a set of input vectors with the corresponding sensitizing values that p_i depends on. Further every path p_j of the set is s' -maximally sensitizable with the corresponding critical sensitizing value s' . Now, from Theorem 3.10, p_j is s' -maximally sensitized with some input vector of the MCS. Hence p_i is s -truly sensitized within the MCS. Q.E.D.

The set MCS thus formed is unique.

Theorem 3.12

The MCS is unique.

Proof: First all the steps are performed in a definite sequence. Second, for each step, the entries are assigned definite values except for fan-out assignments in the final step. But here fan-outs are assigned all possible non-contradicting values. Also maximal compatibilization should produce unique results. Q.E.D.

3.4.2 The Path Checking Fault Detection Algorithm

The Path Checking Fault Detection Algorithm can now be conveniently formulated. Given a combinational circuit C with the set of paths

$$P = \{p_i; 1 \leq i \leq n_p\},$$

a fault detection algorithm to detect the presence of faults in C is formulated as follows.

Step A. Form the MCS for C.

Step B. Form the detection tests T for C. A detection test $t \in T$ is formed in the following way.

1. It consists of mci's of the MCS.
2. Every p_i^s is included in t for $1 \leq i \leq n_p$, $s = 0$ and 1 .
3. Every p_i^s is either s-maximally sensitized or s-truly sensitized within t.
4. It is not a subset of another test.

Step C. Apply a $t \in T$ to C . If every path is checked, then C is fault-free, otherwise it is faulty.

The validity of the algorithm is proved in the following theorems. First the existence of a detection test is proved.

Theorem 3.13

There exists at least a detection test t as defined in the algorithm.

Proof: The only point that needs be pointed out is that if a path p is s -truly sensitizable only, then it depends on a set of paths which is each s' -maximally sensitizable for some s' . Since every s' -maximally sensitizable path is s' -maximally sensitized in the MCS, hence p is s -truly sensitizable within the MCS. Q.E.D.

The validity of a test is proved next.

Theorem 3.14

A multiple-output combinational circuit is fault-free iff a detection test detects no faults. It is faulty iff the detection test detects the presence of faults.

Proof: If the circuit is fault-free, then certainly the test will not detect any faults. On the other hand, if the test detects no faults, then since every path is actually sensitized, maximally or truly, it is guaranteed to be checked, the existence of faults would have been detected at the output edges contradicting to the given

results. In the other case, if the circuit is faulty, then those faults on paths s -maximally sensitized are guaranteed to be detected. For those faults on paths s -truly sensitized, then faults on the critical sensitizing paths may create multiple faults. However, there exists a set of paths s' -maximally sensitized upon which the paths in question depend. Faults on the set are detected. If the set is fault-free then faults on s -truly sensitized paths are detected. If the test detects the existence of faults, the circuit cannot be fault-free, because a fault-free circuit will have every path pass both signals. Q.E.D.

The results of a detection test serve two important functions in addition to detecting the existence of faults. First, they indicate the approximate areas of faults. The partial circuits s -maximally sensitized and found to be fault-free are guaranteed to be free of \bar{s} type faults. And the \bar{s} type faults are only to be found in other parts of the circuit. Also the partial circuits s -maximally sensitized and found to be faulty are the areas where some of the \bar{s} type faults occur. Second, the results serve as the basis for the diagnosis process.

3.4.3 An Example

Example 3.6

Continue with the preceding examples with the combinational circuit C of Figure 3.6. The maximal compatible

rows for the constituent circuits C_{z1} and C_{z2} are formed as follows in the first step.

$$\text{MCR} = \{\text{MCR}_1, \text{MCR}_2\}$$

$$\text{MCR}_1 = \{v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_{16}\}$$

$$\text{MCR}_2 = \{v_{21}, v_{22}, v_{23}, v_{24}, v_{25}\}$$

$$v_{11} = \{p_1^0, p_2^0\}$$

$$= (0,0,\dots,0,0,\dots,0,1,\dots,0,)$$

$$v_{12} = \{p_3^0, p_5^0\}$$

$$= (,0,0,0,\dots,0,0,0,0,1,\dots,0,\dots,1,0,\dots,0,)$$

$$v_{13} = \{p_4^0, p_5^0\}$$

$$(,1,0,1,\dots,1,1,1,1,0,\dots,0,\dots,1,0,\dots,0,)$$

$$v_{14} = \{p_1^1, p_5^1\}$$

$$= (1,0,1,\dots,0,0,\dots,0,\dots,1,1,\dots,1,)$$

$$v_{15} = \{p_2^1, p_5^1\}$$

$$= (0,1,1,\dots,1,1,\dots,0,\dots,1,1,\dots,1,)$$

$$v_{16} = \{p_2^1, p_3^1, p_4^1\}$$

$$= (0,1,0,0,\dots,1,1,0,0,1,\dots,1,\dots,1,1,\dots,1,)$$

$$v_{21} = \{p_6^0, p_7^0, p_9^0\}$$

$$= (,,,0,0,1,1,,,0,0,1,1,1,,0,0,,,0,0,,0)$$

$$v_{22} = \{p_6^0, p_8^0, p_{10}^0\}$$

$$= (,,,1,0,0,0,,,1,1,0,0,0,,0,1,,,0,0,,0)$$

$$v_{23} = \{p_6^1\}$$

$$= (,,,,1,,,,,,,,,0,,,,1,0,,1)$$

$$v_{24} = \{p_7^1, p_8^1\}$$

$$= (,,,1,0,1,,,1,1,0,1,1,,1,0,,,1,0,,1)$$

$$v_{25} = \{p_9^1, p_{10}^1\}$$

$$= (,,,,0,0,1,,,,,,,,0,0,,0,1,,,0,1,,1)$$

In the second step, the blank entries of each of the maximal compatible rows belonging to the same constituent circuit are assigned the corresponding sensitizing values according to their parities. Some of these entries may have two possible assignments. The results are as follows.

$$v_{11} = (0,0,1,0,,,,,0,0,0,0,1,,,0,,,0,1,,,0,)$$

$$v_{12} = (1,0,0,0,,,,,0,0,0,0,1,,,0,,,1,0,,,0,)$$

$$v_{13} = (1,1,0,1,,,1,1,1,1,0,,,0,,,1,0,,,0,)$$

$$v_{14} = (1,0,1,1,,,0,0,1,1,0,,,0,,,1,1,,,1,)$$

$$v_{15} = (0,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1)$$

$$v_{16} = (0,1,0,0,1,1,0,0,1,1,1,1,1,1,1,1,1,1)$$

$$v_{21} = (0,0,1,1,0,0,1,1,1,0,0,0,0,0,0,0,0,0)$$

$$v_{22} = (1,0,0,0,1,1,0,0,0,0,1,0,0,0,0,0,0,0)$$

$$v_{231} = (0,1,0,0,0,0,1,0,0,0,1,1,1,0,0,1,1,1)$$

$$v_{232} = (0,1,1,0,0,0,1,1,1,0,0,0,1,0,0,1,0,1)$$

$$v_{24} = (1,0,1,0,1,1,0,1,1,1,1,0,1,0,1,1,0,1)$$

$$v_{25} = (0,0,0,1,0,0,1,0,0,0,1,0,0,1,0,1,1,1)$$

Notice that with v_{23} , there are two possible assignments for x_6 .

In the third step, performing the maximal compatibility operation on the set of maximal compatible rows obtained in the second step results in the set of maximal compatible input vectors MCS, for C.

$$\text{MCS} = \{i_j; 1 \leq j \leq 18\}$$

$$\begin{aligned} i_1 &= \{v_{11}, v_{21}\} = \{p_1^0, p_2^0, p_6^0, p_7^0, p_9^0\} \\ &= (0,0,1,0,0,1,1) = (i_1, i_2, i_3, i_4, i_5, i_6, i_7) \end{aligned}$$

$$\begin{aligned} i_2 &= \{v_{11}, v_{231}\} = \{p_1^0, p_2^0, p_6^1\} \\ &= (0,0,1,0,1,0,0) \end{aligned}$$

$$i_3 = \{v_{11}, v_{232}\} = \{p_1^0, p_2^0, p_6^1\}$$

$$= (0, 0, 1, 0, 1, 1, 0)$$

$$i_4 = \{v_{11}, v_{25}\} = \{p_1^0, p_2^0, p_9^1, p_{10}^1\}$$

$$= (0, 0, 1, 0, 0, 0, 1)$$

$$i_5 = \{v_{12}, v_{21}\} = \{p_3^0, p_5^0, p_6^0, p_7^0, p_9^0\}$$

$$= (1, 0, 0, 0, 0, 1, 1)$$

$$i_6 = \{v_{12}, v_{231}\} = \{p_3^0, p_5^0, p_6^1\}$$

$$= (1, 0, 0, 0, 1, 0, 0)$$

$$i_7 = \{v_{12}, v_{232}\} = \{p_3^0, p_5^0, p_6^1\}$$

$$= (1, 0, 0, 0, 1, 1, 0)$$

$$i_8 = \{v_{12}, v_{25}\} = \{p_3^0, p_5^0, p_9^1, p_{10}^1\}$$

$$= (1, 0, 0, 0, 0, 0, 1)$$

$$i_9 = \{v_{13}, v_{22}\} = \{p_4^0, p_5^0, p_6^0, p_8^0, p_{10}^0\}$$

$$= (1, 1, 0, 1, 0, 0, 0)$$

$$i_{10} = \{v_{13}, v_{24}\} = \{p_4^0, p_5^0, p_7^1, p_8^1\}$$

$$= (1, 1, 0, 1, 0, 1, 0)$$

$$i_{11} = \{v_{14}, v_{22}\} = \{p_1^1, p_5^1, p_6^0, p_8^0, p_{10}^0\}$$

$$= (1, 0, 1, 1, 0, 0, 0)$$

$$i_{12} = \{v_{14}, v_{24}\} = \{p_1^1, p_5^1, p_7^1, p_8^1\}$$

$$= (1, 0, 1, 1, 0, 1, 0)$$

$$i_{13} = \{v_{15}, v_{22}\} = \{p_2^1, p_5^1, p_6^0, p_8^0, p_{10}^0\}$$

$$= (0, 1, 1, 1, 0, 0, 0)$$

$$i_{14} = \{v_{15}, v_{24}\} = \{p_2^1, p_5^1, p_7^1, p_8^1\}$$

$$= (0, 1, 1, 1, 0, 1, 0)$$

$$i_{15} = \{v_{16}, v_{21}\} = \{p_2^1, p_3^1, p_4^1, p_6^0, p_7^0, p_9^0\}$$

$$= (0, 1, 0, 0, 0, 1, 1)$$

$$i_{16} = \{v_{16}, v_{231}\} = \{p_2^1, p_3^1, p_4^1, p_6^1\}$$

$$= (0, 1, 0, 0, 1, 0, 0)$$

$$i_{17} = \{v_{16}, v_{232}\} = \{p_2^1, p_3^1, p_4^1, p_6^1\}$$

$$= (0, 1, 0, 0, 1, 1, 0)$$

$$i_{18} = \{v_{16}, v_{25}\} = \{p_2^1, p_3^1, p_4^1, p_9^1, p_{10}^1\}$$

$$= (0, 1, 0, 0, 0, 0, 1)$$

For those mci's not maximally assigned, their dependence sets are as follows.

$$i_1: p_1^0, p_2^0 / \{p_5^1\}$$

$$i_2: p_1^0, p_2^0 / \{p_5^1\}$$

$$i_2: p_6^1 / \{p_{10}^0\}$$

$$i_3: p_1^0, p_2^0 / \{p_5^1\}$$

$$i_3: p_6^1 / \{p_7^0\}$$

$$i_4: p_1^0, p_2^0 / \{p_5^1\}$$

$$i_5: p_3^0, p_5^0 / \{p_1^1\}$$

$$i_6: p_3^0, p_5^0 / \{p_1^1\}$$

$$i_6: p_6^1 / \{p_{10}^0\}$$

$$i_7: p_3^0, p_5^0 / \{p_1^1\}$$

$$i_7: p_6^1 / \{p_7^0\}$$

$$i_8: p_3^0, p_5^0 / \{p_1^1\}$$

$$i_{13}: p_2^1, p_5^1 / \{p_4^0\}$$

$$i_{14}: p_2^1, p_5^1 / \{p_4^0\}$$

$$i_{16}: p_6^1 / \{p_{10}^0\}$$

$$i_{17}: p_6^1 / \{p_7^0\}$$

In forming the detection tests, the dependence sets can be excluded from consideration in this example. Notice that, of the sets, p_5^1 depends on p_4^0 which is 0-maximally sensitized and is included in every test. All

the others, p_{10}^0 , p_7^0 , p_1^1 , p_4^0 , p_7^0 are all maximally sensitized and each included in every test.

It seems a tedious task to form the tests, but a little observation will reduce it to a simple one. For example, p_1^0 and p_2^0 appear only in i_1 , i_2 , i_3 , and i_4 , so every test should include one of these four mci's. And also p_9^1 , p_{10}^1 appear in i_4 , i_8 and i_{18} , so one of the three mci's should be in every test, etc. Working along this line, one should find it easy to form a test. Although more tests can be formed, there are 12 tests each consisting of 5 mci's that are most economical in terms of test size. Define this set as T_e , one has

$$T_e = \{t_i; 1 \leq i \leq 12\},$$

$$t_1 = \{i_5, i_6, i_9, i_{12}, i_4\},$$

$$t_2 = \{i_5, i_8, i_9, i_{12}, i_2\},$$

$$t_3 = \{i_6, i_{15}, i_9, i_{12}, i_4\},$$

$$t_4 = \{i_6, i_{18}, i_9, i_{12}, i_1\},$$

$$t_5 = \{i_8, i_{15}, i_9, i_{12}, i_2\},$$

$$t_6 = \{i_8, i_{16}, i_9, i_{12}, i_1\},$$

$$t_7 = \{i_5, i_{16}, i_{10}, i_{11}, i_4\},$$

$$t_8 = \{i_5, i_{18}, i_{10}, i_{11}, i_2\},$$

$$t_9 = \{i_6, i_{15}, i_{10}, i_{11}, i_4\},$$

$$t_{10} = \{i_6, i_{18}, i_{10}, i_{11}, i_1\},$$

$$t_{11} = \{i_8, i_{15}, i_{10}, i_{11}, i_2\},$$

$$t_{12} = \{i_8, i_{16}, i_{10}, i_{11}, i_1\}.$$

Consider the set of faults F on the circuit in Figure 3.6,

$$F = \{f_1, f_9, f_3, f_5, f_7, f_{11}\}$$

$$f_1: x_1 \text{ s-a-0},$$

$$f_9: e_9 \text{ s-a-1},$$

$$f_3: x_3 \text{ s-a-0},$$

$$f_5: x_5 \text{ s-a-0},$$

$$f_7: x_7 \text{ s-a-1},$$

$$f_{11}: e_{11} \text{ s-a-1}.$$

Of the set F , f_9 is made undetectable by f_1 and f_5 , f_7 , f_{11} reduce C_{z2} to p_8 and p_9 which is equivalent to z_2 s-a-1. Let $t = t_1$. Applying t_1 to C , the output vectors obtained are as follows:

$$o_5 = (0,1) = (z_1, z_2),$$

$$o_{16} = (1,1),$$

$$o_9 = (0,1),$$

$$o_{12} = (0,1),$$

$$o_4 = (0,1).$$

Thus the results of the test reflect the following detections:

i_5 detects $\{p_6, p_7, p_9\}$ s-a-1,

i_9 detects $\{p_6, p_8, p_{10}\}$ s-a-1,

i_{12} detects $\{p_1, p_5\}$ s-a-0.

Notice that the partial circuit in each case is s-maximally sensitized and thus its detection is ensured. Also the partial circuit is faulty in at least one path, but not necessarily all the paths due to the fact that all the paths are simultaneously sensitized.

The partial circuits s-maximally sensitized and found to be fault-free are

i_{16} : $\{p_2, p_3, p_4\}$ pass 1,

i_9 : $\{p_4, p_5\}$ pass 0,

i_{12} : $\{p_7, p_8\}$ pass 1,

i_4 : $\{p_9, p_{10}\}$ pass 1.

Being s-maximally sensitized, these parts of the circuit are sure to be free of the types of faults being suspected. Other detections involve s-non-maximal

sensitization and cannot be readily interpreted, but will be treated in the diagnosis process.

Thus the results indicate the approximate areas of faults in the circuit as well as serve as the basis for the diagnosis process.

3.5 Fault Diagnosis of Combinational Circuits

In this section the approach to locate faults in the combinational circuit by determining the ability of every path of the circuit in passing the signals is discussed. Using this approach, an algorithm is developed to locate all the detectable faults in the circuit based on the results of a detection test. Even though different detection results might be obtained from different tests used, the diagnosis process based on the different results should always locate the same set of faults except possibly for equivalent faults.

3.5.1 The Diagnosis Approach

A detection test as designed in the previous section does not individually check every path of a combinational circuit in passing both signals. But rather a set of compatible paths are simultaneously checked for a certain type of faults using a maximal compatible input vector. Faults thus detected might not occur on every path of the set. Thus a diagnosis process is needed to actually locate the faults within the set. However, the results of

a detection test do indicate the approximate areas where both types of faults will occur. The responsibilities of a diagnosis process are then to locate all the faults within these areas. These areas include all the paths each of which is suspected of not being able to pass some signal s . Thus every path of these areas has to be tested to pass the corresponding signal s if it is s -sensitizable within the existing fault picture. If it is not s -sensitizable, then it has to be tested for the possibility of being part of some multiple fault. If every path of these areas is thus tested, then all the faults of the combinational circuit can be located down to equivalent faults. A path is said to be suspected of sticking at s if it has not been proved to pass the signal s . Or, simply it is a suspected path.

Theorem 3.15

If every s -sensitizable suspected path of a combinational circuit is tested to pass s , and every s -unsensitizable suspected path is tested for possible \bar{s} -type multiple faults, the faults of the combinational circuit will be located down to equivalent faults.

Proof: First an s -unsensitizable suspected path which does not include any multiple faults can be considered as either sticking at \bar{s} or as being able to pass s because of the fact it is not s -sensitizable within the given fault picture. After every s -sensitizable suspected

path has been tested to pass s , and every s -unsensitizable suspected path has been tested for \bar{s} -type multiple faults, the remaining s -unsensitizable paths can then be assigned as either sticking at \bar{s} or passing s . Hence every path is thus determined for its ability to pass both signals. All the maximal faulty subcircuits can then be identified and all the single faults are thus located. Multiple faults are of course identified during the process. Thus every type of faults is included in the process of testing every individual path. Different sets of the equivalent faults can result from the different assignments given to the s -unsensitizable paths. Q.E.D.

Location of Single Faults

Locating single faults is equivalent to identifying maximal faulty subcircuits. To do this, a set of input vectors must be found which detects a maximal faulty subcircuit.

A set of paths P_i of a combinational circuit is said to define a subcircuit C_i if P_i totally forms C_i except for the subpath between the output edge of C_i and the common trailing edge of P_i .

The following theorem expresses the characteristics of a set of input vectors which detect a maximal faulty subcircuit.

Theorem 3.16

Given a maximal subcircuit C_i defined by a set of paths P_i with at least one $p \in P_i$ which is s -sensitizable within the existing fault picture, for a set of input vectors I to determine C_i as faulty of \bar{s} -type, it must test for the failure of the signal s in propagating through every s -sensitizable path p .

Proof: If C_i is faulty of \bar{s} -type, then every $p \in P_i$ which is s -sensitizable cannot pass s . Thus s cannot propagate even through one path of P . Since there exists at least one s -sensitizable path, I can be found. Q.E.D.

The formation of such a set of input vectors is defined as follows.

Definition 3.18

Given a maximal subcircuit C_i with the output edge e_i and defined by a set of paths P_i with the trailing edge z_i , C_i is suspected of being faulty of \bar{s} -type. A fault sensitizing vector I for C_i , or C_i sensitizing vector is a set of input vectors which is composed as follows.

1. The subpath between e_i and z_i is sensitized by every $i \in I$. Using part of C_i in the sensitization should be avoided if possible.
2. Every fan-out of different parities of C_i should be assigned all the different possible non-contradicting signal values resulting in several different input vectors for I .

3. The rest of C_i is s -maximally assigned.

For example, given the fault $f_1: x_1$ s-a-1 in Figure 3.4, for the fault $f_{17}: e_{17}$ s-a-0, a C_{17} sensitizing set would include two input vectors $I = \{(d, 1, 1, 1, 1), (d, 1, 1, 0, 1)\}$. Notice that d can assume any value because of f_1 and x_4 is assigned both values due to the fan-out of different parities.

It is not hard to see that such a set of input vectors will detect a maximal faulty subcircuit. The main point is that the subcircuit is applied with all possible input vectors to cause its output edge to assume the signal value s in order to disprove its sticking at \bar{s} . If at least one input vector of I causes the corresponding output edge to assume s , then C_i is not faulty of \bar{s} -type and the smaller subcircuits within C_i faulty of the same type will have to be tested similarly.

Location of Multiple Faults and Multiple Paths

After every s -sensitizable path of the combinational circuit has been determined in passing both signals, those paths which are s -unsensitizable given the proved faults can be combined in locating multiple faults.

Since multiple faults are formed through the mutual sensitization of a set of uncovered undetectable faults, input vectors must be formed which cause these undetectable faults to propagate and meet at some gate where they

mutually sensitize one another. For this to happen, the faults must all be sensitizing to the gate at which they meet and each is sensitized to propagate through the gate. A multiple fault occurs only on a set of paths called a multiple path.

Definition 3.19

An s -sensitizable multiple path is a set of paths P_i , each of which is to be tested to pass s , p_j^s , and it has the following characteristics.

1. Every $p_j \in P_i$ is s -unsensitizable,
2. There exists a set of logical gates G on P_i such that at least two paths of P_i meet at each $g \in G$. Further the set G is such that the s -type signal propagating through paths of P_i is dominant to G ,
3. P_i is s -sensitizable, i.e., every $p_j \in P_i$ is s -sensitizable except for those sensitizing edges belonging to other paths of P_i . These edges will assume the dominant value s , and feed the logical gates of G ,
4. No subset of P_i has the above characteristics.

The following theorem finds the locations of multiple faults.

Theorem 3.17

An \bar{s} -type multiple fault occurs only on an s -sensitizable multiple path P_i .

Proof: Let P_i be the set on which f occurs. Since f is multiple and of \bar{s} -type, so each $p \in P_i$ is s -unsensitizable. In order for $f_i, f_j \in f$ to sensitize each other, \bar{s} must be sensitizing to the gate $g \in G$ where the corresponding two paths P_i and P_j of P_i meet. Hence s is dominant to g . Also in order for f to be sensitized, P_i must be s -sensitizable. Finally since f should not include another multiple fault, hence P_i should not include another multiple path. Q.E.D.

During the diagnosis process, it should not be too complicated in identifying multiple paths. Because usually multiple paths are formed from fan-out paths due to the fact that unsensitizability is caused by fan-outs. For example with the faults $f_1: x_1$ s-a-0 and $f_{12}: e_{12}$ s-a-1 in Figure 3.4, the two paths $p_2 = x_2, e_{10}, e_{18}, z$ and $p_3 = x_2, e_{11}, e_{16}, e_{19}, z$ become a multiple path and $f_{18}: e_{18}$ s-a-1 and $f_{11}: e_{11}$ s-a-1 become a multiple fault. A multiple fault sensitizing input vector which s -sensitizes the multiple path P_i will then detect the multiple fault f .

Sensitization Involving Unsensitizable Critical Sensitizing Paths

During the diagnosis process, it might occur that unsensitizable critical sensitizing paths have to be used in the sensitization of a suspected path. Being unsensitizable in passing the critical sensitizing values,

these critical sensitizing paths are not known to be able to provide the critical sensitizing values. However, it can be shown that the path to be s -sensitized for detecting \bar{s} -type detectable faults on it can be considered as s -sensitizable by assuming that all the critical sensitizing paths are capable of providing the necessary critical sensitizing values even though they cannot be tested to pass the critical sensitizing values with the existing fault picture. This is proved in the following theorem.

Theorem 3.18

Given a path p to be s -sensitized and a set of critical sensitizing paths $P_u = \{p_i; i = 1, \dots, n\}$ with the critical sensitizing values $\{s_i; i = 1, \dots, n\}$ where $s_i = 0$ or 1 . Each p_i is s_i -unsensitizable. Further let $p^s / \{p_1^s, p_2^s, \dots, p_n^s\}$ and p is s -unsensitizable if any $p_i \in P_u$ is not used as a critical sensitizing path. Then p is s -sensitizable and hence the \bar{s} -type faults on it are detectable iff every $p_i \in P_u$ is capable of providing the corresponding critical sensitizing value.

Proof: If $p_i \in P_u$ passes s_i , then p is s -sensitized as the critical sensitizing values are provided. If the \bar{s} -type faults on p are detectable, then p is s -sensitized which implies every p_i provides the necessary critical sensitizing value otherwise p becomes s -unsensitizable because $p^s / \{p_1^s, p_2^s, \dots, p_n^s\}$ and every p_i^s is needed.

Q.E.D.

Thus p is s -sensitizable only by assuming every unsensitizable critical sensitizing path as capable of providing the critical sensitizing value.

Outlines of the Diagnosis Process

In addition to actually determining the suspected paths to be faulty or fault-free, some of the suspected paths can be indirectly determined through the determination of other related suspected paths. A suspected path is said to be confirmed if it is proved to be faulty of the suspected faults. It is said to be disproved if it is proved to be free of the suspected faults. It is determined if it is either confirmed or disproved.

Suspected paths can be determined with fault sensitizing vectors. For suspected paths to be indirectly determined, the following operation is defined.

The indirect determination of suspected paths includes

1. If a set of paths was s -maximally sensitized and found faulty of \bar{s} -type, then if all paths of the set except one are disproved of \bar{s} -type faults, the remaining path is indirectly confirmed faulty of \bar{s} -type.
2. If a set of paths was s -non-maximally sensitized and found faulty of \bar{s} -type, then if all paths of the set except one are disproved of \bar{s} -type faults and the dependence set is proved able to pass the

critical sensitizing values, the remaining path is indirectly confirmed faulty of \bar{s} -type.

3. If a set of paths was s-non-maximally sensitized and found free of \bar{s} -type faults, then if its dependence set is proved able to pass the critical sensitizing values, the set is indirectly disproved faulty of \bar{s} -type.
4. If a subcircuit was s-sensitized and found faulty of \bar{s} -type, then if all of its dependence sets are proved able to pass the critical sensitizing values, the set of paths defining the subcircuit is indirectly confirmed faulty of \bar{s} -type.

According to their capabilities in passing the signals, the paths of the combinational circuit can be divided into four sets during the fault diagnosis process. They are the disproved or good set P_G , the confirmed or bad set P_B , the suspected set P_S and the unsensitizable set P_U . A diagnosis process is then to establish the set of faults F by completely establishing the set P_B . This is done in the following steps.

1. Reduce P_S to null by determining and assigning every $p_i^S \in P_S$ to one of the other three path sets.
2. Find faulty multiple paths from P_U and include them in P_B .
3. Assign every $p_i^S \in P_U$ which remains to either P_G or P_B without creating any new faults for the circuit. And finally

4. Establish F from P_B .

The process can be graphically represented as in Figure 3.7.

3.5.2 The Path Checking Fault Diagnosis Algorithm

The basis of the diagnosis algorithm is derived from the results of a detection test. It consists of the four sets P_G , P_B , P_U and P_S as obtained in the following way.

Given a combinational circuit C with the set of paths P and tested with a detection test t , the basis of the diagnosis algorithm is formed as follows.

1. Include every $p_i \in P$ as p_i^S in P_G if p_i is s-maximally sensitized with some $i \in t$ which detects no suspected faults.
2. Include every $p_i \in P$ as p_i^S in P_G which is indirectly disproved of sticking at \bar{s} .
3. Include every $p_i \in P$ as p_i^S in P_B which is indirectly confirmed of sticking at \bar{s} .
4. Include every $p_i \in P$ as p_i^S in P_U which is s-unsensitizable with the results of the first three steps.
5. Include the remaining paths of P as p_i^S in P_S for the appropriate s .

Given a n_o -output combinational circuit C with the set of paths P and n_o constituent circuits C_{z_i} , $1 \leq i \leq n_o$, and tested with a fault detection test t . C is faulty

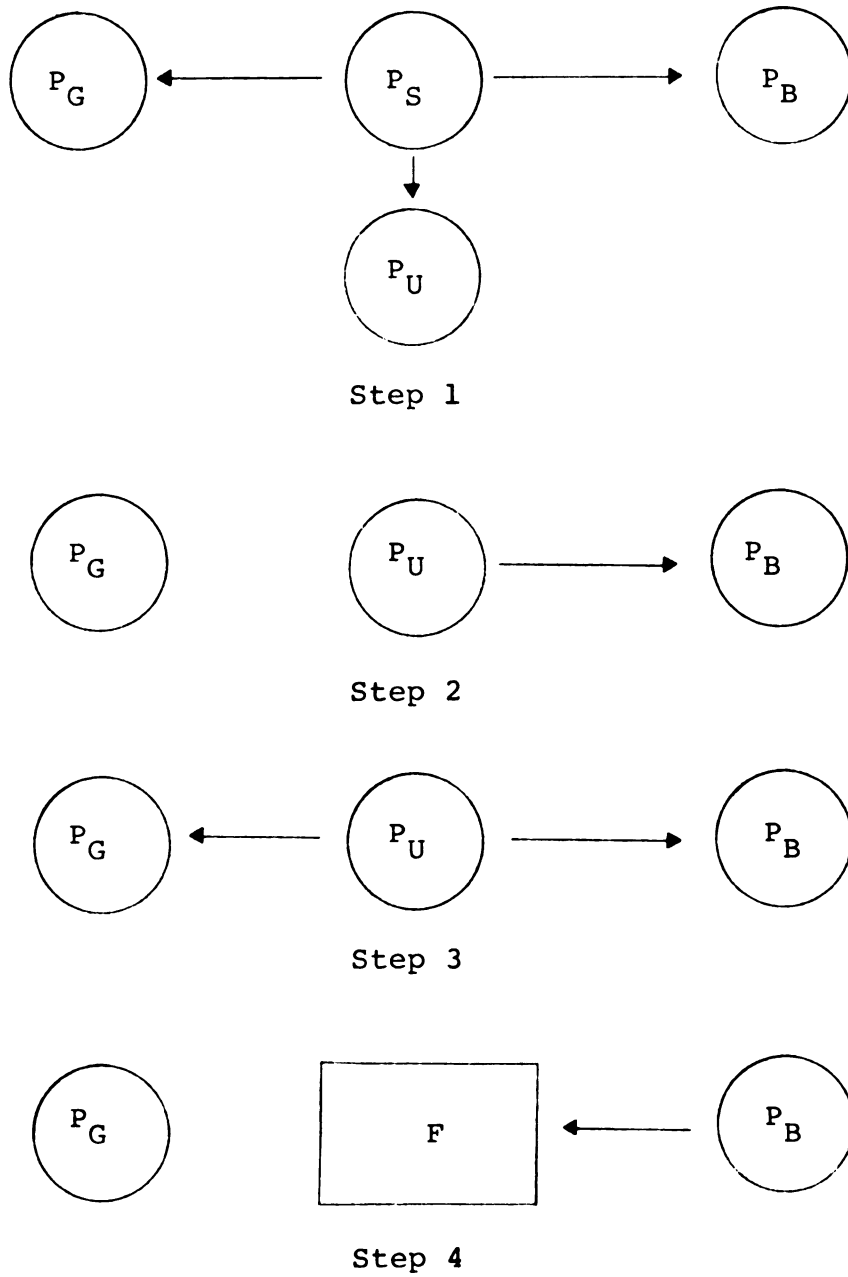


Fig. 3.7.--Graphical representation of the diagnosis process.

with the resulting diagnosis basis $B = \{P_G, P_B, P_U, P_S\}$. A fault diagnosis algorithm to locate all the detectable single and multiple faults, the set of faults F , is presented as follows.

- A. For each C_{zi} of C , find a set of s -sensitizable paths $P_i^S \in P_S$ which, except for the s -unsensitizable subpaths of a subcircuit C_i , define C_i . If no such P_i^S exists, then find such a set P_i^S which depends on P_U for critical sensitizing paths. Notice that $s = 0$ or 1 .
- B. Form a set of C_i -sensitizing input vectors I_i for each C_i .
- C. Form a set of maximal compatible vectors I_C from the sets I_i 's formed in B. The undetermined entries in each $i \in I_C$ can be arbitrarily assigned.
- D. Apply I_C to C . If C_i is known to be s -sensitized with I_C , then if C_i is found faulty of \bar{s} -type move P_i^S to P_B , or if C_i is found free of \bar{s} -type faults and C_i is defined by one path only, move P_i^S to P_G . Otherwise mark C_i as tested.
- E. Move every p_i^S from P_S to either P_G or P_B if it is thus indirectly determined. Move p_i^S to P_U if p is caused s -unsensitizable within the existing fault picture.
- F. Repeat A through E until all maximal faulty subcircuits have been tested. A subcircuit previously

- tested should not be chosen in Step A. Move the remaining P_S to P_U .
- G. Test every s-sensitizable multiple path $P_i^S \in P_U$ for multiple faults. If multiple faults are detected, remove P_i^S from P_U and include it in P_B .
 - H. Remove any one $P_i^S \in P_U$ to either P_G or P_B .
 - I. Remove every $p_i^S \in P_U$ to either P_G or P_B if it is thus indirectly determined.
 - J. Repeat H and I until P_U is reduced to null.
 - K. For every $P_i^S = \{p_j^S; 1 \leq j \leq n_i\}$ in P_B defining a maximal faulty subcircuit C_i of \bar{s} -type, with the output edge e_i include $f_i: e_i \text{ s-a-}\bar{s}$ in F . If f_i propagates down some path and stops at e_j assuming the signal value s' , then instead of f_i , include $f_j: e_j \text{ s-a-}s'$ in F . In this case f_i is equivalent to f_j . Remove P_i^S from P_B . Since even one p^S can define a maximal faulty subcircuit, so P_B is thus reduced to null.

The validity and convergence of the algorithm is proved in the following theorem.

Theorem 3.19

The path checking fault diagnosis algorithm will locate all the detectable single and multiple faults of a combinational circuit C in a finite number of steps.

Proof: Every s-sensitizable path $p^S \in P_S$ is included in either P_G or P_B only if it is so determined.

Otherwise it is always included into P_U . And every s-sensitizable multiple path $P_i^S \in P_U$ is included in P_B if it is thus identified. Hence the remaining p_i^S 's $\in P_U$ are s-unsensitizable and thus their assignments as done in the algorithm do not introduce additional faults and thus do not affect the original fault picture. So every path of C is being determined for its sensitizability, including sensitizability through faults providing the sensitizing values (multiple paths), all the faults, single and multiple, can be thus identified. Since in identifying maximal faulty subcircuits, the algorithm moves from the outputs of the circuit towards the inputs of the circuit and each maximal faulty subcircuit is tested at most once, the algorithm will terminate in a finite number of steps.

Q.E.D.

3.5.3 An Example

Example 3.7

Continuing with the results of Example 3.6, one finds the basis as

$$P_G = \{p_2^1, p_3^1, p_4^1, p_4^0, p_5^0, p_7^1, p_8^1, p_9^1, p_{10}^1\},$$

$$P_S = \{p_1^0, p_2^0, p_3^0, p_1^1, p_5^1, p_6^0, p_7^0, p_8^0, p_9^0, \\ p_{10}^0, p_6^1\},$$

$$P_B = P_U = \emptyset.$$

To apply the algorithm, there are six maximal faulty subcircuits from P_S as follows,

$$P_{18}^0 = \{p_1^0, p_2^0\}, \text{ defining } C_{18},$$

$$P_3^0 = \{p_3^0\}, \text{ defining } C_9,$$

$$P_1^1 = \{p_1^1\}, \text{ defining } C_1,$$

$$P_5^1 = \{p_5^1\}, \text{ defining } C_3,$$

$$C_{z2} = \{p_6^0, p_7^0, p_8^0, p_9^0, p_{10}^0\} \text{ and}$$

$$P_6^1 = \{p_6^1\}, \text{ defining } C_5.$$

For C_{z1} , since $\{p_1^0, p_2^0\}/\{p_5^1\}$, $\{p_3^0\}/\{p_1^1\}$, $\{p_1^1\}/\{p_5^1\}$, and $\{p_5^1\}/\{p_4^0\}$, so $\{p_5^1\}$ should be chosen first because $p_4^0 \in P_G$. The maximal faulty subcircuit for C_{z2} is C_{z2} .

The C_3 -sensitizing input vector to test p_5^1 is

$$I_1 = \{i_{19}\} = (0,0,1,1,\dots).$$

For C_{z2} , the 0-type sensitizing set is

$$I_2 = \{i_{20}, i_{21}\},$$

$$i_{20} = (\dots, 0, 0, 0, 0),$$

$$i_{21} = (\dots, 0, 0, 1, 0).$$

Since they are not compatible, the blank entries can be arbitrarily assigned. Applying the sensitizing input

vectors to C, one finds both P_5 and C_{z2} faulty. That is p_5 cannot pass 1 and every path of C_{z2} fails to pass 0.

The updated basis becomes

P_G : unchanged,

$$P_S = \{p_1^0, p_2^0, p_3^0, p_1^1\},$$

$$P_B = \{p_5^1, p_6^0, p_8^0, p_9^0, p_{10}^0\},$$

$$P_U = \{p_6^1\}.$$

where p_6 becomes 1-unsensitizable due to $\{p_6^1\}/\{p_7^0; p_{10}^0\}$.

Of the set P_S , $\{p_1^0, p_2^0\}/\{p_5^1\}$, $\{p_1^1\}/\{p_5^1\}$, so these two subcircuits also become 0-unsensitizable and 1-unsensitizable respectively. But since $\{p_3^0\}/\{p_1^1\}$, so p_1 can be assumed to be capable of providing the critical sensitizing value 1. The 0-type P_3 -sensitizing vector is then

$$I_3 = \{i_{22}\} = (1, 0, 0, 0, \dots).$$

Applying i_{22} to C, one gets $z_1 = 0$ detecting no faults.

Thus p_3 also becomes 0-unsensitizable and P_S is to be all included in P_U .

$$P_U = \{p_1^0, p_2^0, p_3^0, p_1^1, p_6^1\}$$

From this, the following 0-sensitizable multiple paths are formed.

$$P_{m1} = \{p_1, p_2\}$$

$$P_{m2} = \{p_1, p_3\}$$

The input vector $i_{m1} = (0,0,0,1,,)$ testing P_{m1}^0 detecting no multiple faults. The same input vector also tests P_{m2}^0 for multiple faults. Thus P_U is unchanged. Different assignments for P_U will result in different sets of equivalent faults. For C_{z2} , it is clear that C_{z2} sticks at 1 which is equivalent to $\{f_5, f_7, f_{11}\}$. If one includes P_U in P_G , then p_5 failing to pass 1, i.e., f_3 , will be the only detectable fault which is equivalent to $\{f_1, f_9, f_3, f_5\}$.

3.6 Characteristics of the Method as Applied to Combinational Circuits

In reviewing and comparing the path checking method, as it is applied to combinational circuits, to the other methods included in Chapter II, the following characteristics concerning this method are observed.

First, the main differences between this method and the other methods are that in this method no assumptions about the existence of certain faults are made in developing the tests, and that every path of the combinational circuit is guaranteed to be sensitized if possible with the tests formed from this method. For example, Roth's method [35] assumes the existence of some certain faults in order to formulate tests for that faults. Chang's

method, even though employing, in effect, path sensitization, does not require that the paths be maximally sensitized when possible. While, with the path checking method, every path of the combinational circuit is either maximally sensitized or truly sensitized to achieve actual sensitization.

Second, as a result of the emphasis on actual sensitization, every test from this method carries detection as well as diagnosis information and the detection test serves partial diagnosis functions.

Third, during the fault detection and diagnosis process with this method, the approximate faulty areas of the circuit are clearly defined at each stage of the process. This will prove very useful in fault analysis of integrated circuits.

Fourth, this method can be extended to cover combinational circuits which include EXCLUSIVE OR gates. Since every signal value is sensitizing to these gates, to actually sensitize a path containing such gates, every edge feeding such a gate and incident on the path should assume the same signal value, in the sensitization of this path, throughout a detection test. Thus the maximal compatibility procedures have to be modified to achieve this requirement.

Fifth, this method can be extended to cover combinational circuits which include redundant circuitry. By

employing the concept of "multiple paths" as presented in this chapter, unsensitizable and partially sensitizable paths in such circuits can be incorporated into the path checking method.

Finally, the path checking method can also be applied to sequential circuits as will be discussed in the next chapter.

CHAPTER IV

FAULT DETECTION AND DIAGNOSIS OF
SEQUENTIAL CIRCUITS BY THE
PATH CHECKING METHOD

From the path point of view, the fault analysis of sequential circuits can be regarded as an extension of the fault analysis of combinational circuits. This is evident if one treats the feedback lines of a sequential circuit as "connectors" of the paths found in the combinational portion of the sequential circuit. Thus instead of paths, the analysis will involve sequences of paths with each path sequence beginning with a circuit input edge and ending with a circuit output edge. The main approach used in the analysis is to study the combinational portion of the sequential circuit. To do this, every path of the "embedded" combinational circuit should be included in some path sequence checked in order for the path to be checked. Thus the path sequences of a sequential circuit covered in a fault analysis should cover all the paths of its combinational portion. Further, fault detection and diagnosis of sequential circuits involve a more complicated process

as well as some additional considerations that are unique to sequential circuits.

One of the requirements that confronts the fault detection and diagnosis of sequential circuits is "synchronization" of the feedback lines. Synchronization provides the necessary control over the feedback signals so as to correctly sensitize the path sequences of the circuits. Another requirement before a fault analysis is undertaken is to guide the sequential circuits to some known state. This initial procedure is necessary in order to correctly interpret the test results and to accurately detect and locate the faults in the circuits.

This chapter is devoted to the fault analysis of sequential circuits. As in the previous chapter, the analysis is divided into two parts: fault detection and fault diagnosis. In addition to the synchronization requirement, the limitations placed on the types of circuits, faults and logical elements of sequential circuits considered are identical to that of combinational circuits. In this case, every path should be embedded in at least one s-sensitizable path sequence. Also faults on feedback lines are included in this case.

The basic concepts and approaches from the combinational case are adopted and extended from the "path" viewpoint to the "path sequence" viewpoint to apply to

sequential circuits. Most of the nomenclature from the previous case is also adopted with the like modifications.

4.1 Some Network Properties of Sequential Circuits

Just as a combinational circuit can be viewed as a set of interconnected paths, a sequential circuit can be considered as a set of interconnected path sequences. From this viewpoint, the logical behavior of the sequential circuit can thus be interpreted in terms of signal propagation within this set. It is along this approach that the fault analysis of the sequential circuit proceeds. In this section, a brief discussion of some of the network properties from such a viewpoint is presented.

A sequential circuit with all its feedback lines disconnected reduces to a combinational circuit. The combinational circuit thus obtained has all of the network properties as discussed in Chapter III.

The combinational circuit obtained from disconnecting all of the feedback lines of a sequential circuit will be called the associated combinational circuit of the sequential circuit. With all of its feedback lines mathematically disconnected, it becomes clear that all the given stuck-at type faults of a sequential circuit can be completely translated into faults of its associated combinational circuit. Hence the logical starting point in fault

analyzing a sequential circuit is to check every path of its associated combinational circuit.

When the terminology defined in Chapter III for combinational circuits is used in this chapter in connection with a sequential circuit, it will be meant to refer to its associated combinational circuit. Thus a path of a sequential circuit will be used to mean a path of its associated combinational circuit.

Consider a sequential circuit C with n input edges x_i , $1 \leq i \leq n$, m output edges, z_j , $1 \leq j \leq m$, q state input edges (present state variables), y_k , $1 \leq k \leq q$, q state output edges (next state variables), y'_k , $1 \leq k \leq q$, and the set of n_p paths $P = p_i$; $1 \leq i \leq n_p$, a path sequence and a subpath sequence of C are defined as follows.

Definition 4.1

A path sequence $ps = l_1, l_2, \dots, l_p$, of C is a sequence of paths of C such that

1. l_1 , the leading path, contains an input edge of C as its leading edge,
2. l_p , the trailing path, contains an output edge of C as its trailing edge, and
3. l_i and l_{i+1} , $1 \leq i \leq p-1$, are such that l_i contains a state output edge y' of C as its trailing edge and l_{i+1} contains the state input edge y , which is connected to y' through a feedback line, as its leading edge.

A subpath sequence ps_s of C is a section of a path sequence ps such that ps_s includes either the leading edge or the trailing edge of ps or a middle section of ps .

Thus a path sequence of a sequential circuit is a sequence of paths which starts with an input edge and ends with an output edge of the circuit, with the paths in between connected by feedback lines. A path sequence thus defined need not be of finite length.

To check a path of a sequential circuit, it must be embedded in a path sequence so that it can be tested to pass signals which propagate from the input edges, through the paths, to the output edges of the circuit.

Theorem 4.1

For every path of a sequential circuit, there exists at least one path sequence which contains it.

Proof: A path which is not part of any path sequence is either not accessible from the inputs of the sequential circuit or signals passing through it will never reach the outputs of the sequential circuit. Hence such a path serves no logical functions at all and thus need not be considered as part of the circuit. Q.E.D.

Analogous to the combinational case, partial circuits and constituent circuits can also be defined on sequential circuits.

Definition 4.2

A partial circuit of a sequential circuit C consists of a set of path sequences of C which share a circuit output edge of C as their common trailing edge of their trailing paths.

Definition 4.3

A constituent circuit of C consists of the set of all path sequences which share a circuit output edge as their common trailing edge of their trailing paths.

For a path p of a sequential circuit, there can be infinite number of path sequences which contain it, considering the fact that there may exist a path within the path sequence which can be infinitely repeated within the sequence. Thus, of the set of all possible path sequences which contain p , there exists a subset which is of interest in checking p . Such a subset is called the set of minimal path sequences of p .

Definition 4.4

A minimal path sequence ps of a path p of a sequential circuit C is a path sequence which satisfies these two conditions.

1. It contains p .
2. It cannot qualify as a path sequence if any path(s) are deleted from it.

The set of minimal path sequences of p is the set of path sequences consisting of all the minimal path sequences of p.

Examples of these definitions are illustrated in the following example.

Example 4.1

For the sequential circuit C of Figure 4.1, there are nine paths.

$$P = \{p_i; 1 \leq i \leq 9\}$$

$$p_1 = x_1, e_7, e_{14}, z_1 \text{ or}$$

$$x_1, e_7, e_{14}, y'_4$$

$$p_2 = x_1, e_8, e_{13}, y'_6$$

$$p_3 = x_2, e_{14}, z_1 \text{ or}$$

$$x_2, e_{14}, y'_4$$

$$p_4 = x_3, y'_6$$

$$p_5 = y_4, e_9, e_{10}, e_{14}, z_1 \text{ or}$$

$$y_4, e_9, e_{10}, e_{14}, y'_4$$

$$p_6 = y_4, e_9, e_{11}, e_{15}, z_2 \text{ or}$$

$$y_4, e_9, e_{11}, e_{15}, y'_5$$

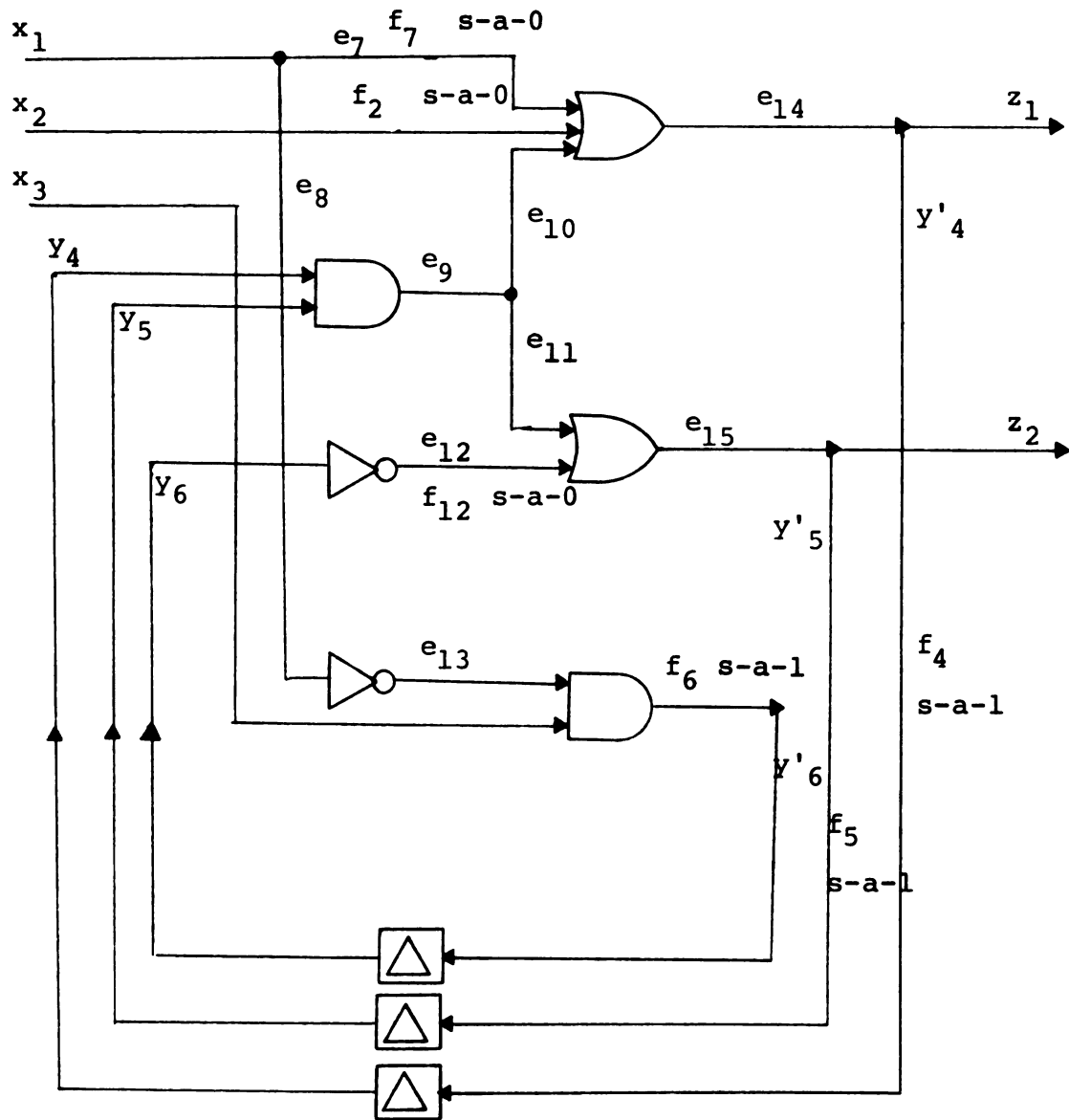


Fig. 4.1.--A sequential circuit for Example 4.1.

$$p_7 = y_5, e_9, e_{10}, e_{14}, z_1 \text{ or}$$

$$y_5, e_9, e_{10}, e_{14}, y'_4$$

$$p_8 = y_5, e_9, e_{11}, e_{15}, z_2 \text{ or}$$

$$y_5, e_9, e_{11}, e_{15}, y'_5$$

$$p_9 = y_6, e_{12}, e_5, z_2 \text{ or}$$

$$y_6, e_{12}, e_5, y'_5$$

The set of minimal path sequences PS_9 of p_9 is

$$PS_9 = \{ps_{91}, ps_{92}\},$$

$$ps_{91} = p_2, p_9,$$

$$ps_{92} = p_4, p_9.$$

The set of minimal path sequences PS_8 of p_8 is

$$PS_8 = \{ps_{81}, ps_{82}, ps_{83}, ps_{84}\},$$

$$ps_{81} = p_1, p_6, p_8,$$

$$ps_{82} = p_2, p_9, p_8,$$

$$ps_{83} = p_3, p_6, p_8,$$

$$ps_{84} = p_4, p_9, p_8.$$

The set of minimal path sequences PS_6 of p_6 is

$$PS_6 = \{ps_{61}, ps_{62}, ps_{63}, ps_{64}\},$$

$$ps_{61} = p_1, p_6,$$

$$ps_{62} = p_2, p_9, p_7, p_6,$$

$$ps_{63} = p_2, p_6,$$

$$ps_{64} = p_4, p_9, p_7, p_6.$$

Thus, PS_9 , PS_8 , and PS_6 can be combined to form a partial circuit.

A subpath sequence $ps_s = p_2, y_6$ can be formed from ps_{91} and consists of a leading section of ps_{91} .

4.2 Path and Path Sequence Sensitization of Sequential Circuits

As was stated earlier in this chapter, the fault analysis of sequential circuits is in essence an extension of the fault analysis of combinational circuits; whereas one was concerned with path checking in the former case, he has to deal with path sequence checking in the latter. As a matter of fact, it can be said that the analyst is confronted with the same problems in both cases except that the problems are formulated either in terms of paths or path sequences.

4.2.1 Sensitizing a Path and a Path Sequence

In order for a path of a sequential circuit to be checked, it must be embedded in a path sequence and the path sequence must be sensitized to pass the signals. If

a path sequence consists of more than one path, then the signals have to propagate through its sequence of paths to reach an output edge of the circuit from an input edge of the circuit. To do this an initial state and a series of input vectors to the circuit have to be found so that the path sequence is sensitized all the way to pass its testing signals. The point thus becomes clear that the feedback lines of the sequential circuit have to be synchronized to ensure this requirement.

Definition 4.5

A path sequence $ps = p_1, p_2, \dots, p_n$ of a sequential circuit C is sensitized with a sequence of input vectors, $is = i_1, i_2, \dots, i_n$, applied to C starting from state s_1 , if p_1, p_2, \dots, p_n are sequentially sensitized with the input-state vectors $(i_1, s_1), (i_2, s_2), \dots, (i_n, s_n)$, where $s_i = \tau(s_{i-1}, i_{i-1}), 2 \leq i \leq n$.

The definitions for signal propagation through a path sequence, a path sequence passing an s-type signal, a path sequence tested to pass s, ps^s , and a path sequence checked would be similarly defined as a path in Chapter III. The only difference is that, in this case, one deals with a set of paths in a sequential order. Also a path sequence being s-sensitizable or s-unsensitizable are similarly defined.

The main point that one is concerned with, however, is that a path of a sequential circuit must be sensitizable in order for it to be checked.

Definition 4.6

A path p of a sequential circuit is sensitized if it is embedded in a path sequence ps which is sensitized. p passes an s-type signal if ps passes an s-type signal. p is tested to pass s if ps is tested to pass s . p is tested if ps is tested to pass both signals and checked if ps passes both signals.

A path of a sequential circuit being s-sensitizable, s-unsensitizable, sensitizable or partially sensitizable are defined as in Chapter III but based on the above definition.

The set of minimal path sequences of a path p needs not contain all the s-sensitizable path sequences that one has to consider in checking p . A minimal path sequence of p may be s-unsensitizable. In this case, it can be, but not necessarily, expanded by inserting paths in it to make it s-sensitizable. The resulting s-sensitizable path sequences will be called expanded minimal path sequences and will take the place of the original minimal path sequence. For example, the minimal path sequence $ps_{g1} = p_1, p_6, p_8$ of p_8 in Example 4.1 is not 1-sensitizable. This is because in 1-sensitizing p_1 , then p_6 becomes

1-unsensitizable. Thus one can insert p_5 into ps_{81} resulting in $ps'_{81} = p_1, p_5, p_6, p_8$ which is 1-sensitizable.

The set of path sequences of p and the set of path sequences of C that are all one has to consider in fault analysis of C are defined as follows.

Definition 4.7

The set of path sequences of a path p of a sequential circuit C is a set of path sequences PS which is formed as follows.

1. Include the set of minimal path sequences of p in PS .
2. If $ps \in PS$ is s -unsensitizable ($s = 0$ or 1), include its s -sensitizable expanded minimal path sequences in PS .
3. If, after step 2 is done for all minimal path sequences, $ps \in PS$ is unsensitizable, delete ps from PS .

The set of path sequences of C is the union of the sets of path sequences of all the paths of C .

The following theorem shows the value of the set of path sequences of C .

Theorem 4.2

The set of path sequences PS of a sequential circuit C consists of all the s -sensitizable path sequences of minimal length (no portions can be deleted).

Proof: From Definition 4.7, it is clear that every s -sensitizable path sequence of minimal length is included in PS. Also every $ps \in PS$ is s -sensitizable for some s as required by the definition. Q.E.D.

For the remaining portion of this chapter, whenever a path sequence is referred, it will belong to the set of path sequences of the sequential circuit.

As with combinational circuits, the reconvergence of path sequences can cause some paths of a sequential circuit to be partially sensitizable or completely unsensitizable. This can happen even though the associated combinational circuit has all its paths sensitizable.

4.2.2 Unsensitizable Paths in Sequential Circuits

If a path of a sequential circuit is s -unsensitizable, then there is at least one edge on the path at which an \bar{s} -type fault is undetectable. These edges thus represent redundancy.

Consider, for example, the sequential circuit in Figure 4.2. In this circuit, every path sequence containing the path $p_7 = y_4, e_{10}, e_{12}, y'_4$ cannot be sensitized to pass 0. This is because whenever p_7 is sensitized to pass 0, the other fan-out path $p_6 = y_4, e_9, e_{11}, z$ is also sensitized to pass 0 which in turn causes y_3 to assume 0 preventing the succeeding path p_6 of the path sequence being considered from being sensitized. Thus the

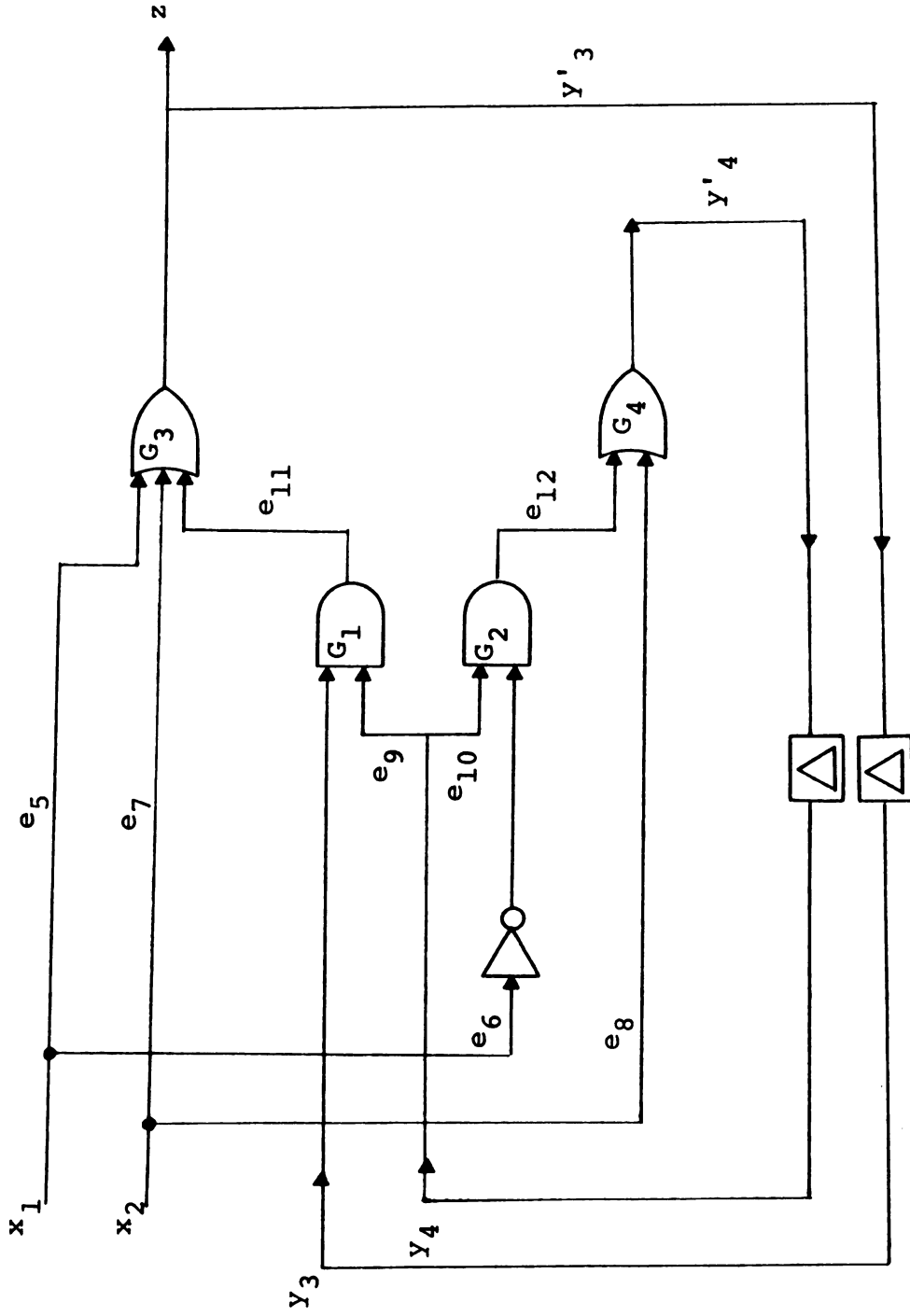


Fig. 4.2.--A sequential circuit containing partially unsensitizable path sequences.

edge e_{10} , which is found on p_7 only, sticking at 1 is undetectable. In this case, the fan-outs on the edge y_4 reconverge, after the feedback lines, at the logical element G_1 .

As with the combination case, a sequential circuit with all paths sensitizable can be derived from any given sequential circuit containing partially sensitizable or unsensitizable paths. A path of a sequential circuit is s -sensitizable if it can be embedded in at least one s -sensitizable path sequence. It is s -unsensitizable if no such path sequences can be found. The following theorems are useful in determining if a sequential circuit has all its paths sensitizable.

Theorem 4.3

A sequential circuit C will have all of its paths sensitizable iff for every path p_i of C , there exist at least two path sequences ps_j, ps_k (they can be the same path sequence), of the set of path sequences PS_i of p_i such that ps_j is s -sensitizable and ps_k is \bar{s} -sensitizable.

Proof: If C has all its paths sensitizable, then ps_j and ps_k can be found. Since PS_i contains all the s -sensitizable and \bar{s} -sensitizable path sequences containing p_i hence ps_j and ps_k are included in PS_i . On the other hand, if not one $ps \in PS_i$ is s -sensitizable or \bar{s} -sensitizable, then clearly p_i is not sensitizable. Q.E.D.

The elimination of partially sensitizable and unsensitizable paths will not alter the logical behavior

of the sequential circuit. This is proved in the following theorem.

Theorem 4.4

For every sequential circuit in which not every path is sensitizable, there exists a sequential circuit with all paths sensitizable which realizes the same logical behavior.

Proof: The proof is similar to that of Theorem 3.2. Given a sequential circuit C_1 with partially or unsensitizable paths, another sequential circuit C_2 with all paths sensitizable will be built from C_1 by removing all paths which are s -unsensitizable ($s = 0$ or 1). Since C_1 contains s -unsensitizable paths, find a path p_i such that all paths of its set of path sequences PS_i are s -unsensitizable. The rest of the proof is similar to that of Theorem 3.2 if one focuses his attention on the associated combinational circuit of C_1 . Q.E.D.

For example, the path p_7 of the circuit C of Figure 4.2 is 0-unsensitizable. The set of path sequences PS_7 of p_7 includes

$$PS_{71} = p_2, p_7, p_6,$$

$$PS_{72} = p_4, p_7, p_6,$$

both being 0-unsensitizable where

$$p_2 = x_1, e_6, e_{12}, y'_4,$$

$$p_4 = x_2, e_8, y'_4,$$

$$p_6 = y_4, e_9, e_{11}, z,$$

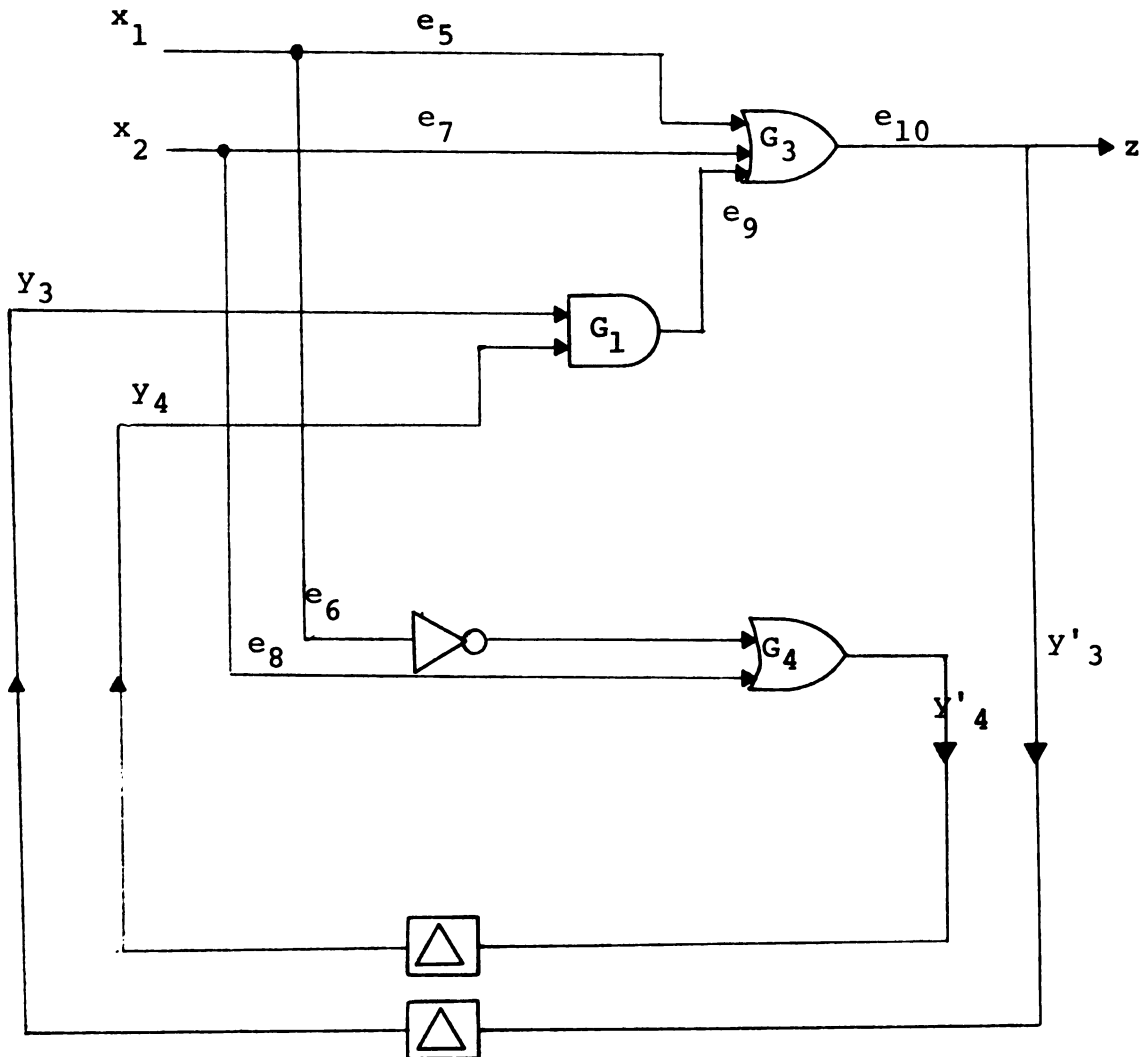
$$p_7 = y_4, e_{10}, e_{12}, y'_4.$$

Of p_7 , the edge e_{10} is not shared by any other paths of C . Thus e_{10} s-a-l is undetectable. Removing both e_{10} and the resulting redundant gate G_2 from C results in C_2 with all paths sensitizable as shown in Figure 4.3.

4.2.3 Maximal and True Sensitization of Path Sequences

The sensitization of a path sequence is defined in terms of the sensitization of its consisting paths. Thus the maximal and true sensitization of a path sequence will be expressed in a similar way. However, in defining the maximal and true sensitization of a path sequence, one must consider, in addition to path-wise maximal sensitization, the maximal sensitization of the whole path sequence which in turn, requires the maximal assignment of all the strictly sensitizing sequential subcircuits.

For example, to sensitize $ps = p_4, p_6$ of the circuit of Figure 4.3, the sensitizing edge y_3 must assume 1 in sensitizing p_6 , which in turn implies the sequential subcircuit C_{y_3} must be 1-maximally assigned, along with



$$P_1 = x_1, e_5, e_{10}, z \text{ or } x_1, e_5, e_{10}, y'_3$$

$$P_2 = x_1, e_6, y'_4$$

$$P_3 = x_2, e_7, e_{10}, z \text{ or } x_2, e_7, e_{10}, y'_3$$

$$P_4 = x_2, e_8, y'_4$$

$$P_5 = y_3, e_9, e_{10}, z \text{ or } y_3, e_9, e_{10}, y'_3$$

$$P_6 = y_4, e_9, e_{10}, z \text{ or } y_4, e_9, e_{10}, y'_3$$

$$i_0 = (0,0) = (x_1, x_2), i_1 = (0,1), i_2 = (1,0), i_3 = (1,1)$$

$$s_0 = (0,0) = (y_3, y_4), s_1 = (0,1), s_2 = (1,0), s_3 = (1,1)$$

Fig. 4.3.--A sequential circuit with all paths sensitizable.

the sensitization of p_4 , in order for ps to be maximally sensitized.

Definition 4.8

Given a path sequence $ps = p_1, p_2, \dots, p_n$ of a sequential circuit C , sensitized with a sequence of input-state vectors to C , $is = v_1, v_2, \dots, v_n$, a sequential subcircuit C_e , with the output edge e incident on p_i , $1 \leq i \leq n$, consists of a set of subpath sequences and/or sections of path sequences all with the common trailing edge e . The sequential subcircuit is formed as follows, starting from e proceeding towards the leading edges of the consisting subpath sequences or sections of path sequences.

1. Include the subcircuit with the output edge e , of the associated combinational circuit of C , in C_e .
2. If C_e includes a state input edge y , then C_e should be extended to cover the constituent circuit C_y , of the associated combinational circuit.
3. Repeat 2 to correspond to $p_{i-1}, p_{i-2}, \dots, p_2, p_1$.
Thus for the example above, the incident sequential subcircuit C_{y_3} incident on p_4 , in the sensitization of ps is

$$C_{y_3} = \{p_1, y_3; p_3, y_3; p_5, y_3; p_6, y_3\}.$$

The maximal assignment of a sequential subcircuit will be similarly defined as a subcircuit in a combinational circuit. With this, the maximal sensitization of a path sequence can be readily defined.

Definition 4.9

A path sequence $ps = p_1, p_2, \dots, p_n$ is s-maximally sensitized, if every $p_i, 1 \leq i \leq n$ is s-maximally sensitized and every strictly sensitizing sequential subcircuit is \bar{s} -maximally assigned. It is s-maximally sensitizable if this is possible. It is maximally sensitizable if it is possible for both values of s.

If a path sequence is to be s-sensitized, but is not s-maximally sensitizable, then, as in the combinational case, s-true sensitization should be sought instead. In this situation, as in the combinational case, at least one subpath sequence for every \bar{s} -non-maximally assigned sensitizing sequential subcircuit must be included in some \bar{s} -maximally sensitizable path sequence in order to attain s-true sensitization for the path sequence in question. Such a path sequence is called a critical sensitizing path sequence, with \bar{s} as its critical sensitizing value, as in the combinational case. In this way, the s-sensitization of a path sequence ps, is said to depend on its critical sensitizing path sequences, ps_i, ps_j , etc., being able to pass their corresponding critical sensitizing values, $s_i,$

s_j, \dots , in order for ps to be actually sensitized to pass s . This is expressed in notation as follows.

$$\underline{is}: ps^s / \{(C_i, ps_i^{s_i}); (C_j, ps_j^{s_j}); \dots\},$$

where \underline{is} is the sequence of input-state vectors used to sensitize ps to pass s , C_i is the non-maximally assigned sensitizing sequential subcircuit corresponding to p_i , etc. If confusion will not thus occur, \underline{is} , C_i and C_j will be omitted.

If a path sequence ps is defined to be s -maximally sensitized of n th order with a sequence of input-state vectors, \underline{is} , within a set of sequences of input-state vectors, IS , as in Definition 3.12, the s -true sensitization of ps can be defined as follows.

Definition 4.10

A path sequence ps is said to be s -truly sensitized with a sequence of input-state vectors \underline{is} , and within a set of sequences of input-state vectors IS , if there exists a number n such that ps is s -maximally sensitized of n th order with \underline{is} , and within IS .

Example 4.2

The circuit C of Figure 4.3 will be used to illustrate these definitions. The path sequence $ps_4 = p_4, p_6$ is 1-maximally sensitized with $\underline{is}_4 = i_3, i_0$ applied to C in state s_3 . The path sequence $ps_2 = p_2, p_6$ is not

1-maximally sensitized with $\underline{is}_2 = i_0$, i_0 applied to C in state s_3 since the sequential subcircuit C_{y_3} is not 1-maximally assigned. The critical sensitizing path sequence section is p_5, y_3 or p_6, y_3 with the critical sensitizing value of 1. The path sequence $ps_{52} = p_3, p_5$ is 1-maximally sensitized with $\underline{is}_{52} = i_1$, i_0 applied to C in s_0 . Thus with ps_4, ps_{52} 1-maximally sensitized, ps_2 is 1-truly sensitized with is_2 and within $(i_3, s_3), (i_0, s_3), (i_1, s_0), (i_0, s_3)$, with which ps_4 and ps_{52} are 1-maximally sensitized respectively.

The maximal and true sensitization of a path of a sequential circuit are defined in terms of its embedding path sequence.

Definition 4.11

A path p of a sequential circuit is s-maximally sensitizable if at least one path of its set of path sequences PS is s-maximally sensitizable, otherwise it is s-non-maximally sensitizable. If p is s-non-maximally sensitizable, then it is s-truly sensitizable if at least one path of PS is s-truly sensitizable. The question of s-true sensitizability of the s-non-maximally sensitizable paths of a sequential circuit also arises in connection with its fault analysis. As in the combinational case, every s-non-maximally sensitizable path is s-truly sensitizable.

Theorem 4.5

In a sequential circuit with every path sensitizable, every s-non-maximally sensitizable path is s-truly sensitizable.

Proof: The proof is similar to that of Theorem 3.5. If there existed a path which was not s-truly sensitizable, then since none of its set of path sequences is s-maximally sensitizable, there could exist a set of undetectable faults in contradiction to the assumption that every path of the circuit is sensitizable. And the assumption causes at least one detectable fault if the circuit is faulty.

Q.E.D.

4.3 Faults of Sequential Circuits

Faults in sequential circuits will be analyzed and interpreted, as in combinational circuits, in terms of path sequence checking. Thus by determining the ability of the path sequences in passing both signals, faults in the sequential circuit can be detected down to equivalent faults. The basic approach used in studying faults of a sequential circuit is the same as that used for a combinational circuit, with the path sequences taking the place of paths.

If a path sequence is actually sensitized and tested to pass both signals, it is free of stuck-at faults iff it is checked. The proof for this would be similar to that of Theorem 3.6 and is thus omitted.

The same types of faults, undetectable, single, multiple and equivalent faults, are also found in a sequential circuit. In addition, faults may occur on feedback lines.

The definitions for Section 3.3 will also apply in this section with "paths" replaced by "path sequences" and "a combinational circuit" by "a sequential circuit."

If a set of subpath sequences defines an s -type maximal faulty sequential subcircuit, then an \bar{s} -type fault will occur at the output edge of such a subcircuit. Similar to Theorem 3.7, an s -type fault in a sequential circuit is single iff it occurs at the output edge of an s -type maximal faulty sequential subcircuit. The proof would be analogous.

If an s -type maximal faulty sequential subcircuit has every subpath sequence \bar{s} -unsensitizable, i.e., not one path sequence including it is \bar{s} -sensitizable, then the s -type fault at its output edge is undetectable. Such undetectable faults might form multiple faults as in a combinational circuit.

Thus all the faults in a sequential circuit are identified, in the same way as in a combinational circuit, within the context of the set of path sequences selected for a test to cover all the paths of the sequential circuit. The following example will illustrate the different types of faults.

Example 4.3

Consider the sequential circuit of Figure 4.1. The fault $f_6:y'_6$ s-a-1 will be a single fault caused by the two subpath sequences x_1, e_8, e_{13}, y'_6 and x_3, y'_6 sticking at 0 and 1 respectively. With the faults f_7, f_2, f_{12} all s-a-0, then $f_5:y'_5$ s-a-1, will not be detectable because y'_4 cannot assume the sensitizing value 1 due to the fan-out at e_9 . Similarly, $f_4:y'_4$ s-a-1 will not be detectable in the presence of f_7, f_2 and f_{12} . However, f_4 and f_5 collectively form a detectable multiple fault in the presence of f_7, f_2 and f_{12} .

4.4 Positioning the Sequential Circuits
in a Desired, Known State

Given a sequential circuit under test in a random and unknown state, it must first be initialized to some desirable and known state before a detection or diagnosis experiment can be performed on it. The positioning of the circuit in the desired state means applying a selected input sequence and observing the corresponding output sequence to ensure that the circuit has been correctly positioned. This section is devoted to designing such input sequences and additional procedures that are necessary to position the sequential circuit under test at some desirable and known state.

The positioning procedure consists of two steps; the first step drives the circuit from some unknown state

to any known state from which the second step drives the circuit to a desirable state according to the state diagram of the circuit.

Before one attempts to design a method of obtaining the positioning input sequences a couple of observations will be made. First, if the circuit under test is actually faulty, then it might be possible that no input sequences will ever drive the circuit to the desired state. This, of course, will be indicated by the absence of the correct corresponding output sequences expected. In this case, the circuit in effect has been detected faulty. Second, if the circuit is initially in a state from which the desired state is unaccessible even though the circuit is fault-free, it is impossible then to drive the circuit to the desired state. For such circuits, fault detection and diagnosis are clearly confined to those states accessible from the starting state the circuit was in at the time of test.

The basic idea used in the designing of the positioning input sequences for the first step is to assign signal values to the state variables of the circuit, or, if impossible for some of the state variables, to find out the signal values they assume throughout this step. Thus the signal values of all the state variables, and thus the current state of the circuit are determined at the end of the application of such an input sequence. In order for

this to be possible, the state variables of the circuit should each be observable at the output edges. A state variable is observable at the output edges if there exists at least an input sequence such that, through the application of the input sequence, a subpath sequence starting with the state input edge of the state variable and ending with a circuit output edge is sensitized.

It can be proved that every state variable of a sequential circuit is observable if every path of the sequential circuit is sensitizable. This is because unobservable state variables will in no way contribute to the logical behavior of the circuit and hence are redundant.

In addition to being observable, a state variable may also be controllable, which makes it possible for some signal value to be assigned to the state variable from the input edges of the circuit. A state variable is controllable from the input edges if there exists at least an input sequence such that through the application of the input sequence, a subpath sequence starting with a circuit input edge and ending in the state input edge of the state variable is sensitized. Thus a state variable being controllable provides part of the logical function of introducing some of the applied signal values to the memory of the circuit.

Through the controllability and observability of its state variables, a sequential circuit can be

positioned in a known state regardless of the original state it assumed before the application of a positioning input sequence. The positioning input sequences, however, will be determined during the positioning process.

Theorem 4.6

For every sequential circuit with all paths sensitizable, there exists at least one input sequence which drives the circuit to a known state from an unknown state.

Proof: First, since all paths of the circuit are sensitizable, all state variables of the circuit are observable. Also at least one of the state variables is also controllable. Otherwise the circuit could not be sequential. Hence every state variable of the circuit will either be controlled or observed during the application of a selected input sequence which can be determined during the positioning process. Since the number of state variables is finite, the signal values on all the state variables can be determined with an input sequence of finite length. And thus at the end of the application of such an input sequence, the state of the circuit is known. Q.E.D.

Design of Positioning Input Sequences

A positioning input sequence consists of two parts, the known-state part and the desired-state part.

To design the known-state part, first assign the signal variable s_i to the state variable y_i for all the

state variables. Then whenever the value of y_i is known, it is assigned to s_i . Then for every state variable not controllable, find the input sequence which causes it observable and assign the value to its signal variable. After all the state variable not controllable have thus been determined, the remaining undetermined variable can easily be determined since they are controllable. The known-state part will then consist of such input sequences.

The desired-state part is then calculated from the state diagram of the circuit.

The following example will help illustrate the procedure.

Example 4.4

Consider the circuit C of Figure 4.4 with two state variables y_1 and y_2 . Clearly y_2 is controllable while y_1 is not. Because for all the possible assignments to x and y_2 , the next state variable y'_1 is still a function of y_1 . Thus there is no way to assign a signal value to y_1 . However, y_1 is observable when y_2 has a signal value of 1. Thus an input sequence of $\underline{is} = 1,1$ will sensitize the path $p = y_1, e_a, e_b, z$. And at the end of \underline{is} , z will have the signal value y_1 assumed at the previous state of the circuit. From this one can calculate the previous state, and thus the current state of the circuit.

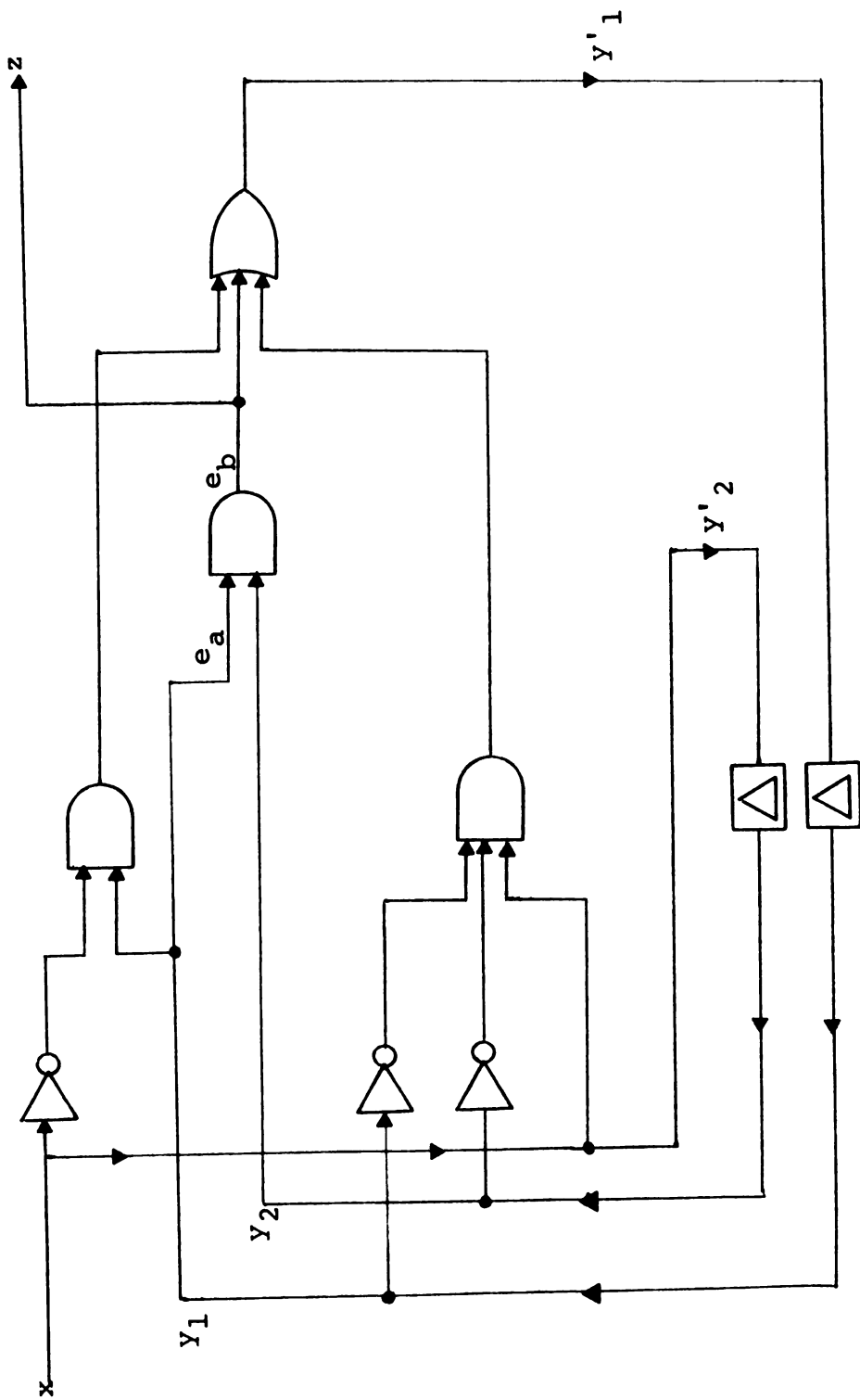


Fig. 4.4.--A sequential circuit for Example 4.4.

4.5 Fault Detection of Sequential Circuits

In this section, the design of detection tests and an algorithm for conducting the fault detection of sequential circuits are presented. The results of such a detection test can be used for fault diagnosis as will be discussed in the next section.

A faulty sequential circuit may also be detected, apart from a detection test, through the positioning procedure to initialize the circuit in a starting state, when the expected output sequences are not obtained as a response to the positioning input sequences applied. In this case the same diagnosis algorithm will also be used to locate the faults.

The basic approach in designing and fault detecting a sequential circuit is to check every path of the sequential circuit. This means that for every path of the sequential circuit, there should be included in the test at least two path sequences (they may be the same) which contain the path and one of which is sensitized to pass 1 and the other sensitized to pass 0. Every path sequence should be either s-maximally sensitized or s-truly sensitized within the test.

Another aspect in designing a detection test, that is unique to sequential circuits is the sequencing of the individual input-state vectors to form testing input sequences. This is necessary to provide for the

consistency of the present next state serving as the next current state throughout the testing sequences.

4.5.1 The Path Sequence Sensitization Table and Maximal Compatible Sets

The whole procedure used in forming the Path Sequence Sensitization Table, PSST, and the MCS for a sequential circuit is similar to that of a combinational circuit; the main difference is that, instead of the paths, one is working with the path sequences of the sequential circuit. The MCS, from which a detection test is formed, is built from the PSST. The PSST is a table which includes all the path sequences of the sequential circuit.

The Construction of the PSST

Consider a sequential circuit C with n_i input edges, n_s state input edges, n_o output edges and n_s state output edges. The edges of C , including fan-out edges, are uniquely labelled. Let there be a total of n_e edges which include the circuit input edges, the state input edges, the circuit output edges, the state output edges and the remaining edges of C . Let PS be the set of path sequences of C .

$$PS = \{ps_i; 1 \leq i \leq n_{PS}\}$$

The PSST consists of input-state vectors (input vector concatenated with state vector) each of which

sensitizes one or more paths of C . The PSST is formed as follows.

1. There are n_e columns corresponding to the n_e edges of C .
2. For every path sequence $ps \in PS$, there are two sets of rows to test ps pass 0 and 1, denoted by ps^0 and ps^1 respectively.
3. For a given set of rows corresponding to ps_i^s , assign the corresponding sensitizing values to edges incident on ps , s to edges on ps with even parity, \bar{s} to edges on ps with odd parity between the edges and the trailing edge of ps .
4. Enter signal values to edges which can be determined through fan-outs, NOT gates and feedback lines. The input edges to OR gates whose output edges assuming 0 and the input edges to AND gates whose output edges assuming 1 can also be determined.

Except for the arrangement of the rows within it, the PSST thus formed is unique. The only edges assigned values on the PSST are those on the path sequences, those incident on the path sequences and those whose values are thus determined from the first two types of edges. An example illustrates the procedure.

Example 4.5

Consider the sequential circuit C of Figure 4.5 with four circuit input edges, two circuit output edges and three state variable edges. There are a total of 23 edges and 15 paths.

$$p_1 = x_1, e_{14}, e_{17}, z_1$$

$$p_2 = x_1, e_{14}, y'_5$$

$$p_3 = x_2, e_8, e_{14}, e_{17}, z_1$$

$$p_4 = x_2, e_8, e_{14}, y'_5$$

$$p_5 = x_2, e_9, e_{15}, e_{18}, z_1$$

$$p_6 = x_2, e_9, e_{15}, y'_7$$

$$p_7 = x_3, e_{10}, e_{15}, e_{18}, z_1$$

$$p_8 = x_3, e_{10}, e_{15}, y'_7$$

$$p_9 = x_3, y'_6$$

$$p_{10} = x_4, e_{16}, z_2$$

$$p_{11} = y_5, e_{11}, e_{12}, e_{16}, z_2 \quad .$$

$$p_{12} = y_5, e_{11}, e_{13}, z_1$$

$$p_{13} = y_6, e_{11}, e_{12}, e_{16}, z_2$$

$$p_{14} = y_6, e_{11}, e_{13}, z_1$$

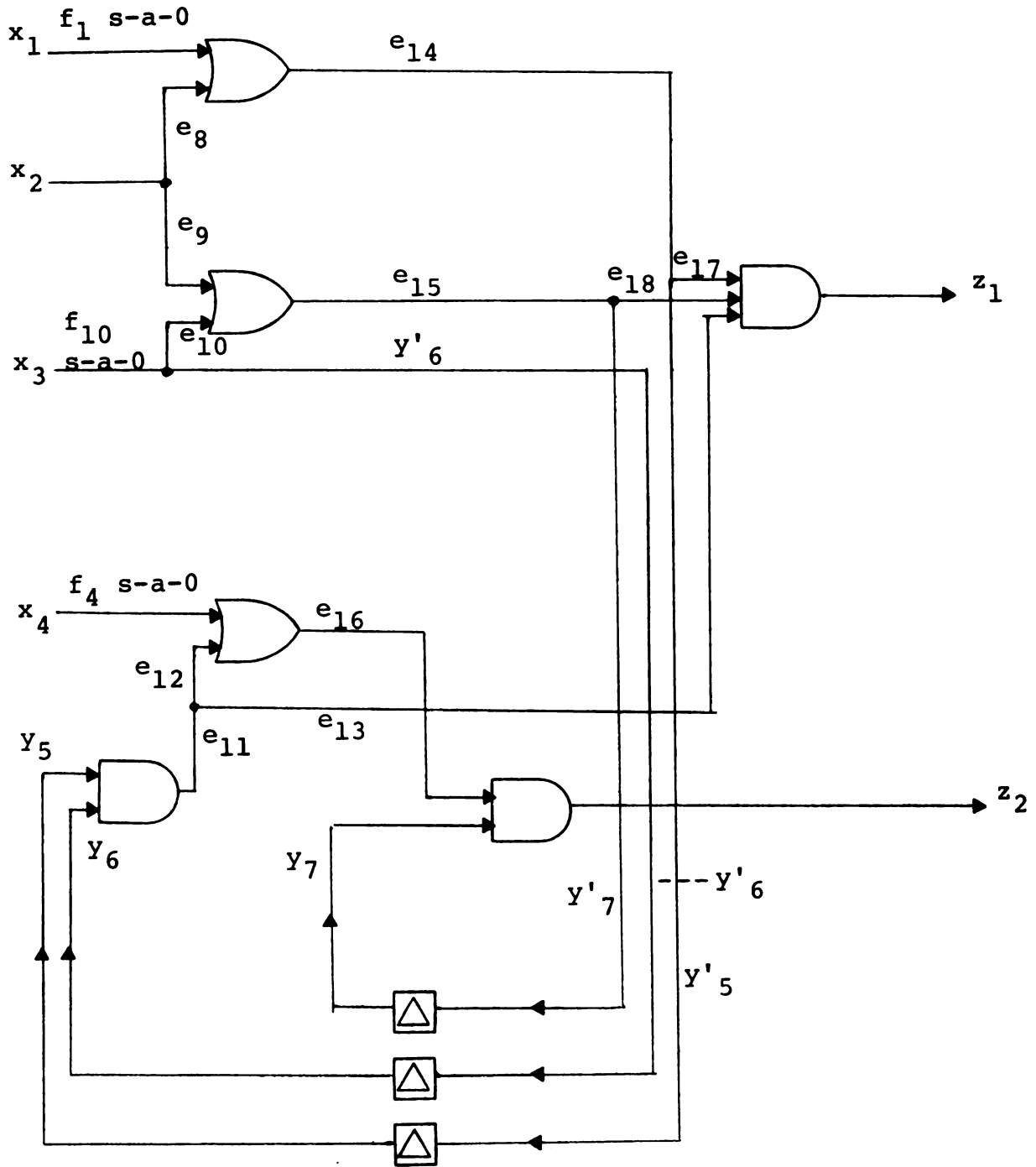


Fig. 4.5.--The sequential circuit for Example 4.5.

$$P_{15} = y_7, z_2$$

The sets of path sequences of the paths are as follows.

$$PS_1 = \{ps_1\} = p_1$$

$$PS_2 = \{ps_{21}, ps_{22}\}$$

$$ps_{21} = p_2, p_{11}$$

$$ps_{22} = p_2, p_{12}$$

$$PS_3 = \{ps_3\} = p_3$$

$$PS_4 = \{ps_{41}, ps_{42}\}$$

$$ps_{41} = p_4, p_{11}$$

$$ps_{42} = p_4, p_{12}$$

$$PS_5 = \{ps_5\} = p_5$$

$$PS_6 = \{ps_6\}$$

$$ps_6 = p_6, p_{15}$$

$$PS_7 = \{ps_7\} = p_7$$

$$PS_8 = \{ps_8\}$$

$$ps_8 = p_8, p_{15}$$

$$PS_9 = \{ps_{91}, ps_{92}\}$$

$$ps_{91} = p_9, p_{13}$$

$$ps_{92} = p_9, p_{14}$$

$$PS_{10} = \{ps_{10}\} = p_{10}$$

$$PS_{11} = \{ps_{21}, ps_{41}\}$$

$$PS_{12} = \{ps_{22}, ps_{42}\}$$

$$PS_{13} = \{ps_{91}\}$$

$$PS_{14} = \{ps_{92}\}$$

$$PS_{15} = \{ps_6, ps_8\}$$

Thus there are a total of 13 path sequences,

$$PS = \{ps_1, ps_{21}, ps_{22}, ps_3, ps_{41}, ps_{42}, ps_5, ps_6, ps_7, ps_8, ps_{91}, ps_{92}, ps_{10}\}.$$

The PSST will be a table of 26 sets of rows, two sets for each path sequence passing both 0 and 1, and 23 columns corresponding to the 23 edges of C. It is shown in Table 4.1.

The Forming of the MCS

The process used in forming the MCS from the PSST is somewhat different from that of the combinational case. This is due to the fact that within the same constituent

circuit, there are path sequences of different lengths (number of paths). In forming the maximal compatible rows for the constituent circuits, only rows of the same order will be combined to form maximal compatible rows. Given a path sequence $ps = p_1, p_2, \dots, p_n$, the row p_i^s ; $1 \leq i \leq n$, will be called of $(n - i + 1)$ th order. Another point that needs attention is that in assigning signal values to the path sequences, the state input edge and state output edge of the same state variable should assume the same signal value at all times to ensure consistency of signal assignments for the path sequences.

The first step in forming the MCS is to form the MCS for each of the constituent circuits. To do this, the rows of the first order of the path sequences belonging to the same constituent circuit are first considered in forming maximal compatible rows. Then if a set of path sequences PS whose rows of i th order form some maximal compatible row, the rows of $(i + 1)$ th order of PS only will be used further in forming maximal compatible rows. In this way, the path sequences of the same constituent circuit are considered in forming the MCS starting from the trailing paths working towards the leading paths.

For example, for the constituent circuit C_{z1} (Example 4.5) the rows p_1^1 (ps_1^1), p_7^1 (ps_7^1), p_{14}^1 (ps_{92}^1), p_{12}^1 (ps_{22}^1) and p_{12}^1 (ps_{42}^1) of the PSST form a maximal compatible row $(1, 0, 1, 1, 1, 1) = (x_1, x_2, x_3, x_4,$

y_5, y_6, y_7). Of the path sequences, ps_1 and ps_7 are of length 1 and ps_{22}, ps_{42} and ps_{92} are of length 2. Thus the rows $p_2^1 (ps_{22}^1), p_4^1 (ps_{42}^1)$ and $p_9^1 (ps_{92}^1)$ of 2nd order can only be considered further for forming maximal compatible rows. Of the three rows, $p_2^1 (ps_{22}^1)$ and $p_9^1 (ps_{92}^1)$ form a maximal compatible row $(1,0,1,.,.,.)$ and $p_4^1 (ps_{42}^1)$ and $p_9^1 (ps_{92}^1)$ form another maximal compatible row $(0,1,1,.,.,.)$.

In the second step, on the maximal compatible rows, the entries corresponding to the sequential subcircuits incident on the path sequences being sensitized are assigned values. These sequential subcircuits are assigned values as was done in the combinational case: maximally assigned if possible.

For example, of the path sequences previously considered, ps_1 and ps_7 have all the entries corresponding to C_{z1} assigned in the maximal compatible row $(1,0,1,.,1,1,1)$. For ps_{22} and ps_{92} with the rows $(1,0,1,.,.,.), (1,0,1,.,1,1,1)$, all the sensitizing sequential subcircuits have also been assigned values. The sequential subcircuit incident on $p_2 (ps_{22})$ is the subpath x_2, e_8 which has been assigned in $(1,0,1,.,.,.)$. The sequential subcircuits incident on $p_{12} (ps_{22})$ are the subpath sequence x_3, y'_6, y_6 , the sequential subcircuits C_{e17} and C_{e18} . The subpath sequence has been assigned with $(1,0,1,.,.,.)$ and C_{e17} and C_{e18} have been completely assigned with $(1,0,1,.,1,1,1)$.

Since it takes a sequence of maximal compatible rows to sensitize a path sequence of length larger than one, such a sequence will be called a maximal compatible row sequence. It should also be pointed out that a row as referred to here actually is an incompletely defined input-state vector.

In the third step, all the maximal compatible row sequences will be used to form maximal compatible row sequences for the entire circuit. However, in doing so, two considerations need to be made. First, it is necessary to form input-state vector sequences from the row sequences later in designing the tests. So the state variable components of the row sequences which have not been assigned values should be left free to assume all possible values in order to make concatenations of row sequences possible. Second, in order to require every individual row perform detection in a test, row sequences of equal lengths only will be used in forming maximal compatible row sequences in this step.

The state variable components then will be assigned the value x , which is compatible to itself only.

In the final step, the remaining entries in the maximal compatible row sequences are arbitrarily assigned values as they don't affect the sensitization of the path sequences. Or their assignments can be delayed until the input-state vectors are being formed later.

The set of input-state vector sequences thus obtained is the MCS. The following theorems prove that the MCS is all that one needs in forming the detection tests. First the s-maximal sensitization of s-maximally sensitizable path sequences with the MCS is proved.

Theorem 4.7

Every s-maximally sensitizable path sequence is s-maximally sensitized with some input-state vector sequence from the MCS.

Proof: First, every s-sensitizable path sequence is included in the PSST. Given ps s-maximally sensitizable with an input-state vector sequence is . Thus with is , every strictly sensitizing sequential subcircuit is maximally assigned. Now in forming the MCS from the PSST, the strictly sensitizing sequential subcircuits of ps are assigned values in the second step. In that step every sensitizing sequential subcircuits are maximally assigned if possible. Hence since ps is found in the PSST, is will be found in the MCS. Q.E.D.

The next theorem proves the s-true sensitization of s-non-maximally sensitizable path sequences within the MCS.

Theorem 4.8

Every s-non-maximally sensitizable path sequence is s-truly sensitizable within the input-state vector sequences that can be formed from the MCS.

Proof: First, every s-non-maximally sensitizable path sequence ps is s-truly sensitizable with some is within all the possible vector sequences to the sequential circuit. That means all the critical sensitizing path sequences of ps found in is are either s'-maximally sensitizable or indirectly s'-maximally sensitizable. In other words, ps depends on a set of path sequences which are s'-maximally sensitizable for some values of s' . But the set are s'-maximally sensitized within the MCS from Theorem 4.7.

Q.E.D.

4.5.2 The Path Sequence Checking Fault Detection Algorithm

The Path Sequence Checking Fault Detection Algorithm can now be formulated. One problem of fault detection of a sequential circuit that was not found in the fault detection of a combinational circuit is the necessity to ensure the proper starting state of the sequential circuit before a test is applied. Thus it will be assumed that the actual starting state of the sequential circuit can be known so as to guarantee that the correct starting state is obtained before a test is applied.

Given a sequential circuit C with the set of paths

$$P = \{p_i; 1 \leq i \leq n_p\},$$

a fault detection algorithm to detect the presence of faults is formulated as follows.

Step A. Form the MCS for C .

Step B. Form the detection tests T for C . A detection test $t \in T$ is formed as follows.

1. Select a set of path sequences PS_t , such that every path of C is included in the set.
2. Find a set of input-state vector sequences, IS_t , from the MCS such that every ps_i^s , $ps \in PS_t$, $s = 0$ and 1 , is s -maximally sensitized with some $\underline{is} \in IS_t$, or s -truly sensitized with \underline{is} , within IS_t .
3. The set IS_t is not a subset of another IS'_t that can be formed in step 2.
4. Form an input sequence with the set IS_t , placing maximally assigned input-state vector sequences (i.e., they s -maximally sensitize their corresponding path sequences) at the beginning portion of the sequence, if possible. Appropriately assign values to entries with x in the input sequence so that the current and next state variables have the same signal

values. Arbitrarily assign values to the remaining entries.

Step C. Position C to some starting state s_0 so that a test $t \in T$ is applicable to C. If C is detected faulty during the positioning to the starting state, as found from the incorrect output vectors, the presence of faults have been detected, and there is no need to proceed in the algorithm.

Step D. Apply a $t \in T$ to C. If every path is checked, then C is fault-free, otherwise it is faulty.

The validity of the algorithm is proved in the following theorems.

Theorem 4.9

There exists at least a detection test as defined in the algorithm.

Proof: As was shown previously, one can find a set of input-state vector sequences as defined in 2 of Step B. Thus a test can always be found. The state variable components of the input-state vector sequences have been allowed to assume any values and hence a sequence of input-state vector sequences can be designed. Q.E.D.

Theorem 4.10

A sequential circuit (with all paths sensitizable) is fault-free iff a detection test, as described in the Path Sequence Checking Fault Detection Algorithm, detects

no faults. It is faulty iff the detection test detects the presence of faults.

Proof: Since every path of the circuit is included in the test, thus the whole circuit including the feedback lines are included in the test. If the circuit C is fault-free, then certainly the detection test t will detect no faults. If t detects no faults, then those path sequences s -maximally sensitized are free of \bar{s} -type faults. Then those path sequences s -truly sensitized in turn are free of \bar{s} -type faults because all of their critical sensitizing path sequences are s -maximally sensitized. And since the starting state is guaranteed to be correct, hence C is fault-free. On the other hand, if C is faulty, then since the starting state is correct, t will detect faults, otherwise C would be fault-free. If t detects the presence of faults, then faults will occur on some path sequences s -maximally sensitized or s -truly sensitized or both. Otherwise, with the correct starting state, the output vector sequence produced would not reflect detection of faults on s -maximally sensitized path sequences, and thus also would not reflect detection on s -truly sensitized path sequences. Q.E.D.

4.5.3 An Example

Example 4.6

The circuit C in Figure 4.5 will be used in the example. The PSST is shown in Table 4.1. The first step in forming the MCS is to form the maximal compatible rows for both constituent circuits C_{z1} and C_{z2} .

$$\text{MCR} = \{\text{MCR}_1, \text{MCR}_2\}$$

For C_{z1} , the rows of first order are used to form the following maximal compatible rows.

$$v_{11} = \{p_1^0, p_3^0\}$$

$$= (0, 0, 1, , 1, 1, ,)$$

$$v_{12} = \{p_{12}^0 (ps_{22}^0), p_{12}^0 (ps_{42}^0)\}$$

$$= (, , , , 0, 1, 1, , , , 0, 0, 0, 1, 1, , 1, 1, 1, , 1, 0, ,)$$

$$v_{13} = \{p_5^0, p_7^0\}$$

$$= (1, 0, 0, , 1, 1, ,)$$

$$v_{14} = \{p_{14}^0\}$$

$$= (, , , , 1, 0, , , , , 0, 0, 0, 1, 1, , 1, 1, 1, , 1, 0, ,)$$

$$v_{15} = \{p_1^1, p_7^1, p_{12}^1 (ps_{22}^1), p_{12}^1 (ps_{42}^1), p_{14}^1\}$$

$$= (1, 0, 1, , 1, 1, 1)$$

$$\begin{aligned}
 v_{16} &= \{p_3^1, p_5^1, p_{12}^1 (ps_{22}^1), p_{12}^1 (ps_{42}^1), p_{14}^1\} \\
 &= (0, 1, 0, , 1, 1, 1)
 \end{aligned}$$

The rows of second order corresponding to ps_{22}^0 and ps_{42}^0 of v_{12} are combined to form

$$\begin{aligned}
 v_{17} &= \{p_2^0 (ps_{22}^0), p_4^0 (ps_{42}^0)\} \\
 &= (0, 0, 1, , , ,).
 \end{aligned}$$

For v_{14} , there is only one row of second order

$$\begin{aligned}
 v_{18} &= \{p_9^0 (ps_{92}^0)\} \\
 &= (, , 0, , , ,).
 \end{aligned}$$

For v_{15} , two maximal compatible rows can be formed.

$$\begin{aligned}
 v_{19} &= \{p_4^1 (ps_{42}^1), p_9^1 (ps_{92}^1)\} \\
 &= (0, 1, 1, , , ,)
 \end{aligned}$$

$$\begin{aligned}
 v_{110} &= \{p_2^1 (ps_{22}^1), p_9^1 (ps_{92}^1)\} \\
 &= (1, 0, 1, , , ,)
 \end{aligned}$$

Similarly for v_{16} , there are two maximal compatible rows.

$$\begin{aligned}
 v_{111} &= \{p_4^1 (ps_{42}^1), p_9^1 (ps_{92}^1)\} \\
 &= (0, 1, 1, , , ,) \\
 &= v_{19}
 \end{aligned}$$

$$\begin{aligned}
 v_{112} &= \{p_2^1 (ps_{22}^1), p_9^1 (ps_{92}^1)\} \\
 &= v_{110}
 \end{aligned}$$

So for C_{z1} , the maximal compatible rows formed in the first step are

$$\begin{aligned}
 MCR_1 &= \{v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_{16}, v_{17}, v_{18}, \\
 &\quad v_{19}, v_{110}\}.
 \end{aligned}$$

For C_{z2} , the maximal compatible rows formed from the first order rows are

$$\begin{aligned}
 v_{21} &= \{p_{11}^0 (ps_{21}^0), p_{11}^0 (ps_{41}^0), p_{10}^0\}, \\
 &= (,,,0,0,1,1),
 \end{aligned}$$

$$\begin{aligned}
 v_{22} &= \{p_{15}^0 (ps_6^0), p_{15}^0 (ps_8^0)\}, \\
 &= (,,,1,,0,0),
 \end{aligned}$$

$$\begin{aligned}
 v_{23} &= \{p_{13}^0 (ps_{91}^0), p_{10}^0\}, \\
 &= (,,,0,1,0,1),
 \end{aligned}$$

$$\begin{aligned}
 v_{24} &= \{p_{11}^1 (ps_{21}^1), p_{11}^1 (ps_{42}^1), p_{15}^1 (ps_8^1), p_{13}^1\}, \\
 &= (,,,0,1,1,1),
 \end{aligned}$$

$$\begin{aligned}
 v_{25} &= \{p_{15}^1 (ps_6^1), p_{10}^1\}, \\
 &= (,,,1,1,0,1),
 \end{aligned}$$

$$\begin{aligned}
 v_{26} &= \{p_{15}^1 (ps_8^1), p_{10}^1\}, \\
 &= (,,,1,,1,1).
 \end{aligned}$$

The maximal compatible rows formed from the second order rows are as follows.

$$\begin{aligned}
 v_{27} &= \{p_2^0 (ps_{21}^0), p_4^0 (ps_{41}^0)\} \\
 &= (0,0,1,,,,)
 \end{aligned}$$

$$\begin{aligned}
 v_{28} &= \{p_6^0 (ps_6^0), p_8^0 (ps_8^0)\} \\
 &= (,0,0,,,,)
 \end{aligned}$$

$$\begin{aligned}
 v_{29} &= \{p_9^0 (ps_{91}^0)\} \\
 &= (,1,0,,,,)
 \end{aligned}$$

$$\begin{aligned}
 v_{210} &= \{p_2^1 (ps_{21}^1), p_8^1 (ps_8^1), p_9^1 (ps_{91}^1)\} \\
 &= (1,0,1,,,,)
 \end{aligned}$$

$$\begin{aligned}
 v_{211} &= \{p_4^1 (ps_{41}^1), p_9^1 (ps_{91}^1)\} \\
 &= (0,1,1,,,,)
 \end{aligned}$$

$$\begin{aligned}
 v_{212} &= \{p_6^1 (ps_6^1)\} \\
 &= (,1,0,,,,)
 \end{aligned}$$

$$\begin{aligned}
 v_{213} &= \{p_8^1 (ps_{81}^1)\} \\
 &= (,0,1,,,,)
 \end{aligned}$$

Thus for C_{z2} ,

$$\text{MCR}_2 = \{v_{21}, v_{22}, v_{23}, v_{24}, v_{25}, v_{26}, v_{27}, v_{28}, \\ v_{29}, v_{210}, v_{211}, v_{212}, v_{213}\}.$$

In the second step in forming the MCS, every sensitizing sequential subcircuits are assigned values maximally, if possible. For example, notice that ps_{92} being 0-sensitized with $is_{92} = v_{18}, v_{14}$ has the sensitizing sequential subcircuit C_{y5} and the sensitizing subcircuit C_{e17} and C_{e18} . To maximally assign C_{y5} , v_{18} becomes

$$v_{18} = (1, 1, 0, \dots),$$

and v_{14} becomes

$$v_{14} = (\dots, 1, 0, 1, \dots, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, \dots).$$

To maximally assign C_{e17} and C_{e18} , v_{14} becomes

$$v_{14} = (1, 1, 1, \dots, 1, 0, 1).$$

The other maximal compatible rows can be similarly further defined as follows.

$$v_{12} = (1, 1, 1, \dots, 0, 1, 1)$$

$$v_{22} = (\dots, 1, 1, 0, 0)$$

$$v_{26} = (\dots, 1, 0, 1, 1)$$

$$v_{28} = (1, 0, 0, \dots)$$

$$v_{29} = (1, 1, 0, \dots)$$

$$v_{212} = (1, 1, 0, \dots)$$

$$v_{213} = (0, 0, 1, \dots)$$

In the third step in forming the MCS, assign x to the state variable components, y_5, y_6, y_7 , which have not yet been assigned, in every row. And then forming maximal compatible row sequences from the row sequences of equal lengths found in the second step, one gets the following input-state vector sequences.

$$\underline{is}_1 = \{v_{11}\} = \{p_1^0, p_3^0\}$$

$$= (0, 0, 1, \dots, 1, 1, x)$$

$$\underline{is}_2 = \{v_{17}, v_{12}, v_{27}, v_{21}\} = \{ps_{22}^0, ps_{42}^0, ps_{21}^0, ps_{41}^0, p_{10}^0\}$$

$$= (0, 0, 1, \dots, x, x, x), (1, 1, 1, 0, 0, 1, 1)$$

$$\underline{is}_3 = \{v_{13}\} = \{p_5^0, p_7^0\}$$

$$= (1, 0, 0, \dots, 1, 1, x)$$

$$\underline{is}_4 = \{v_{28}, v_{22}\} = \{ps_6^0, ps_8^0\}$$

$$= (1, 0, 0, \dots, x, x, x), (\dots, 1, 1, 0, 0)$$

$$\begin{aligned}\underline{is}_5 &= \{v_{18}, v_{14}, v_{29}, v_{23}\} = \{ps_{92}^0, ps_{91}^0, p_{10}^0\} \\ &= (1, 1, 0, , x, x, x), (1, 1, 1, 0, 1, 0, 1)\end{aligned}$$

$$\begin{aligned}\underline{is}_6 &= \{v_{23}\} = \{p_{10}^0\} \\ &= (, , , 0, 1, 0, 1)\end{aligned}$$

$$\begin{aligned}\underline{is}_7 &= \{v_{21}\} = \{p_{10}^0\} \\ &= (, , , 0, 0, 1, 1)\end{aligned}$$

$$\begin{aligned}\underline{is}_8 &= \{v_{110}, v_{15}, v_{210}, v_{24}\} = \{ps_{22}^1, ps_{92}^1, ps_{21}^1, \\ &\quad ps_8^1, ps_{91}^1, p_1^1, p_7^1\} \\ &= (1, 0, 1, , x, x, x), (1, 0, 1, 0, 1, 1, 1)\end{aligned}$$

$$\begin{aligned}\underline{is}_9 &= \{v_{210}, v_{24}, v_{112}, v_{16}\} = \{ps_{22}^1, ps_{92}^1, \\ &\quad ps_{21}^1, ps_8^1, ps_{91}^1, p_3^1, p_5^1\} \\ &= (1, 0, 1, , x, x, x), (0, 1, 0, 0, 1, 1, 1)\end{aligned}$$

$$\begin{aligned}\underline{is}_{10} &= \{v_{19}, v_{15}, v_{211}, v_{24}\} = \{ps_{42}^1, ps_{92}^1, \\ &\quad ps_{41}^1, ps_{91}^1, p_1^1, p_7^1\} \\ &= (0, 1, 1, , x, x, x), (1, 0, 1, 0, 1, 1, 1)\end{aligned}$$

$$\begin{aligned}\underline{is}_{11} &= \{v_{211}, v_{24}, v_{111}, v_{16}\} = \{ps_{41}^1, ps_{91}^1, \\ &\quad ps_{42}^1, ps_{92}^1, p_3^1, p_5^1\} \\ &= (0, 1, 1, , x, x, x), (0, 1, 0, 0, 1, 1, 1)\end{aligned}$$

$$\begin{aligned}\underline{is}_{12} &= \{v_{15}\} = \{p_1^1, p_7^1\} \\ &= (1, 0, 1, 1, 1, 1)\end{aligned}$$

$$\begin{aligned}\underline{is}_{13} &= \{v_{16}\} = \{p_3^1, p_5^1\} \\ &= (0, 1, 0, 1, 1, 1)\end{aligned}$$

$$\begin{aligned}\underline{is}_{14} &= \{v_{18}, v_{14}, v_{212}, v_{25}\} = \{ps_{92}^0, ps_6^1, p_{10}^1\} \\ &= (1, 1, 0, x, x, x), (1, 1, 1, 1, 1, 0, 1)\end{aligned}$$

$$\begin{aligned}\underline{is}_{15} &= \{v_{17}, v_{22}, v_{213}, v_{26}\} = \{ps_{22}^0, ps_{42}^0, \\ &\quad ps_8^1, ps_{10}^1\} \\ &= (0, 0, 1, x, x, x), (1, 1, 1, 1, 0, 1, 1)\end{aligned}$$

$$\begin{aligned}\underline{is}_{16} &= \{v_{25}\} = \{p_{10}^1\} \\ &= (, , , 1, 1, 0, 1)\end{aligned}$$

$$\begin{aligned}\underline{is}_{17} &= \{v_{26}\} = \{p_{10}^1\} \\ &= (, , , 1, 0, 1, 1)\end{aligned}$$

In the final step, the remaining entries can be assigned any values. However, it would be preferable to delay the assignments until the test sequence is to be formed in order to form maximal compatible input-state vector sequences at that time.

For those path sequences not s-maximally sensitized with the input-state vector sequences, their dependence sets are listed in the following.

$$\underline{is}_1: p_1^0, p_3^0 / \{p_7^1; ps_{22}^1 \text{ or } ps_{42}^1 \text{ or } ps_{92}^1\}$$

$$\underline{is}_2: ps_{21}^0, ps_{41}^0, p_{10}^0 / \{ps_8^1\}$$

$$\underline{is}_3: p_5^0, p_7^0 / \{p_1^1; ps_{22}^1 \text{ or } ps_{42}^1 \text{ or } ps_{92}^1\}$$

$$\underline{is}_4: ps_6^0, ps_8^0 / \{p_{10}^1\}$$

$$\underline{is}_5: ps_{91}^0, p_{10}^0 / \{ps_6^1\}$$

$$\underline{is}_6: p_{10}^0 / \{ps_6^1; ps_{91}^0\}$$

$$\underline{is}_7: p_{10}^0 / \{ps_8^1; ps_{21}^0 \text{ or } ps_{41}^0\}$$

$$\underline{is}_{14}: p_{10}^1 / \{ps_{91}^0\}$$

$$\underline{is}_{15}: p_{10}^1 / \{ps_{21}^0 \text{ or } ps_{41}^0\}$$

$$\underline{is}_{16}: p_{10}^1 / \{ps_{91}^0\}$$

$$\underline{is}_{17}: p_{10}^1 / \{ps_{21}^0 \text{ or } ps_{41}^0\}$$

This concludes the construction of the MCS. In Step B, the detection tests are designed. There are two sets of path sequences each cover all the paths of C.

$$PS_1 = \{ps_1, ps_{22}, ps_3, ps_{41}, ps_5, ps_6, ps_7, ps_8, \\ ps_{91}, ps_{92}, ps_{10}\}$$

$$PS_2 = \{ps_1, ps_{21}, ps_3, ps_{42}, ps_5, ps_6, ps_7, ps_8, \\ ps_{91}, ps_{92}, ps_{10}\}$$

To cover PS_1 , there are two sets of input-state vector sequences from the MCS that can be formed, each of which qualifies to form an input sequence.

$$IS_1 = \{\underline{is}_1, \underline{is}_2, \underline{is}_3, \underline{is}_4, \underline{is}_5, \underline{is}_8, \underline{is}_{11}, \underline{is}_{14}\}$$

$$IS_2 = \{\underline{is}_1, \underline{is}_2, \underline{is}_3, \underline{is}_4, \underline{is}_5, \underline{is}_9, \underline{is}_{10}, \underline{is}_{14}\}$$

To cover PS_2 , the same two sets are also formed. Thus IS_1 and IS_2 are the only sets of input-state vector sequences for consideration in the synthesis of input sequences.

To form an input sequence from IS_1 , the state variables can now be assigned values. Since all ps_i^s are not s-maximally sensitized with \underline{is}_4 , it will be placed at the end of the input sequences. If one forms an input sequence as

$$t_1 = \underline{is}_2, \underline{is}_5, \underline{is}_8, \underline{is}_{11}, \underline{is}_{14}, \underline{is}_4,$$

then the state variables can be assigned as follows and thereby \underline{is}_1 and \underline{is}_3 can be eliminated from t , since they form maximal compatible vectors with \underline{is}_2 and \underline{is}_4 respectively.

$$\underline{is}_2 = (0,0,1,,1,1,1), (1,1,1,0,0,1,1)$$

$$\underline{is}_5 = (1,1,0,,1,1,1), (1,1,1,0,1,0,1)$$

$$\underline{i}_{s_8} = (1,0,1,,1,1,1), (1,0,1,0,1,1,1)$$

$$\underline{i}_{s_{11}} = (0,1,1,,1,1,1), (0,1,0,0,1,1,1)$$

$$\underline{i}_{s_{14}} = (1,1,0,,1,0,1), (1,1,1,1,1,0,1)$$

$$\underline{i}_{s_4} = (1,0,0,,1,1,1), (,,1,1,0,0)$$

The remaining entries can assume any values since they do not serve any purposes in the test. Let all assume zeros.

Let the sets of inputs, states and outputs of C be assigned as follows.

$$i_0 = (0,0,0,0) = (x_1, x_2, x_3, x_4)$$

$$i_1 = (0,0,0,1)$$

.
.
.

$$i_{15} = (1,1,1,1)$$

$$s_0 = (0,0,0) = (y_5, y_6, y_7)$$

$$s_1 = (0,0,1)$$

.
.
.

$$s_6 = (1,1,1)$$

$$o_0 = (0,0) = (z_1, z_2)$$

$$o_1 = (0,1)$$

$$o_2 = (1,0)$$

$$o_3 = (1,1)$$

Then the input-state vector sequences can be expressed in terms of inputs and states of the circuit as follows.

$$\underline{is}_2 = i_2, s_7, i_{14}, s_3$$

$$\underline{is}_5 = i_{12}, s_7, i_{14}, s_5$$

$$\underline{is}_8 = i_{10}, s_7, i_{10}, s_7$$

$$\underline{is}_{11} = i_6, s_7, i_4, s_7$$

$$\underline{is}_{14} = i_{12}, s_5, i_{15}, s_5$$

$$\underline{is}_4 = i_8, s_7, i_1, s_4$$

Similarly from IS_2 , t_2 is formed as

$$t_2 = \underline{is}_2, \underline{is}_5, \underline{is}_9, \underline{is}_{10}, \underline{is}_{141}, \underline{is}_4$$

with

$$\underline{is}_9 = (1,0,1,0,1,1,1), (0,1,0,0,1,1,1),$$

$$\underline{is}_{10} = (0,1,1,0,1,0,1), (1,0,1,0,1,1,1),$$

$$\underline{is}_{141} = (1,1,0,0,1,1,1), (1,1,1,1,1,0,1),$$

or in terms of inputs and states,

$$\underline{is}_9 = i_{10}, s_7, i_4, s_7,$$

$$\underline{is}_{10} = i_6, s_5, i_{10}, s_7,$$

$$\underline{is}_{141} = i_{12}, s_7, i_{15}, s_5.$$

In Step C, the circuit should be positioned in s_7 for either test. Since every state variable is controllable from the input variables, an input of i_{14} or i_{15} alone should guide C to s_7 .

In step D, applying the test sequence

$$t = i_{15}, t_1,$$

to C and observing its output vector sequence thus produced constitute a detection test.

Let the set of faults be F occurring on C, in Figure 4.5 where

$$F = \{f_1, f_{10}, f_4\},$$

$$f_1: x_1 \text{ s-a-0}, f_{10}: e_{10} \text{ s-a-0} \text{ and } f_4: x_4 \text{ s-a-0}.$$

The output vector sequences obtained, o , under a fault-free condition and $o(F)$ with F, as well as the state sequence obtained with F, $s(F)$ are listed as follows.

s_{t1}		s_7	s_3	s_7	s_5	s_7	s_7	s_7	s_7	s_5	s_5	s_7	s_4	s_0
$i_{15}t_1$	i_{15}	i_2	i_{14}	i_{12}	i_{14}	i_{10}	i_{10}	i_6	i_4	i_{12}	i_{15}	i_8	i_1	
o_{t1}		o_1	o_0	o_3	o_0	o_3	o_3	o_3	o_1	o_0	o_1	o_1	o_0	
$s_{t1}(F)$		s_7	s_2	s_7	s_5	s_7	s_2	s_2	s_7	s_5	s_5	s_7	s_0	s_0
$o_{t1}(F)$		o_1	o_0	o_3	o_0	o_1	o_0	o_0	o_3	o_0	o_0	o_1	o_0	

The results of the detection test clearly indicate the presence of faults. For example, the first discrepancy of output vectors was detected with the application of $i_{15} = i_{10}, s_7, i_{10}, s_7$ which resulted in the output vector sequence of o_1, o_0 from the fault F , instead of o_3, o_3 under a fault-free situation. The results can be further analyzed to serve as a basis for a diagnosis procedure. This is discussed in the next section.

4.6 Fault Diagnosis of Sequential Circuits

The basic approach in the fault diagnosis of sequential circuits is the same as that of combinational circuits: to determine the ability of the paths of the sequential circuit in passing both signals. The diagnosis process, however, is more complicated due to the necessity of positioning the state variables when detecting input-state vector sequences are to be applied to the circuit.

An algorithm is presented in this section to locate all the detectable faults in the circuit. As in the combinational case, different sets of equivalent faults can be obtained from different assignments of s-unsensitizable path sequences.

4.6.1 The Diagnosis Approach

The process of checking the paths of the sequential circuit involves identifying maximal faulty subcircuits of the associated combinational circuit for single faults. After this step, multiple faults can then be located by checking multiple path sequences.

Location of Single Faults

The method of identifying maximal faulty subcircuits in a sequential circuit is more complicated than that of the combinational case. In determining a suspected maximal faulty subcircuit, every path of the subcircuit which has not been determined in its ability to pass the corresponding signal should be tested to pass that signal.

Given a detectable \bar{s} -type maximal faulty subcircuit within a given fault picture, there exists at least a path of the subcircuit which is s-sensitizable; i.e., the path can be embedded in an s-sensitizable path sequence. To confirm the maximal faulty subcircuit, then, every s-sensitizable path of the subcircuit must be shown to fail

to pass s . The proof is similar to Theorem 3.16 and is omitted.

The following definition describes the formation of input-state vector sequences to sensitize a maximal faulty subcircuit.

Definition 4.12

Given an \bar{s} -type suspected maximal faulty subcircuit C_i defined by a set of paths P_i with e_i as the output edge of C_i . A set of C_i -sensitizing input-state vector sequences is formed as follows:

1. For every path $p_j \in P_i$ which is s -sensitizable within the given fault picture, find all the s -sensitizable path sequences containing p_j . Then find the input-state vector sequences each of which causes the s -type signal to propagate between the leading edges and e_i of the path sequences and the s -type signal is sensitized between e_i and the trailing edges of the path sequences.
2. Form maximal compatible input-state vector sequences IS_i from the input-state vector sequences found in 1 for all s -sensitizable paths of P_i .
3. Find input-state vector sequences to drive the sequential circuit to the desired starting states of IS_i .

The following theorem proves the existence and validity of the input-state vector sequences, as found in

Definition 4.12, in sensitizing the maximal faulty subcircuits.

Theorem 4.11

A set of C_i -sensitizing input-state vector sequences as defined in Definition 4.12 exists and will s -sensitize C_i , suspected faulty of \bar{s} -type.

Proof: Since C_i is s -sensitizable within the given fault picture, hence there exists at least a path sequence containing $p_j \in P_i$ which is s -sensitizable on the subpath sequence between e_i and its trailing edge. Hence an input-state vector sequence exists which can s -sensitize the path sequence as described. Now since IS_i contains all such input-state vector sequences s -sensitizing all the s -sensitizable path sequences defining C_i , hence IS_i will detect C_i . Also for every s -sensitizable path sequence ps_j defining C_i , the circuit can be driven to a starting state for $\underline{is}_j \in IS_i$ to s -sensitize ps_j , otherwise ps_j would be s -unsensitizable. Q.E.D.

Example 4.7

To illustrate Definition 4.12 consider the circuit C of Figure 4.3 with C_{e_9} suspected of s -a-0. For y_3 , the path sequences will be either $ps_{51} = p_1, p_5$ or ps_{52} (see Example 4.2). This is because if e_5 and e_7 both s -a-0, then C_{e_9} s -a-0 is not detectable. Similarly for y_4 , the path sequences will be ps_2 and ps_4 . If the fault picture

being e_5 s-a-0, then the input sequence $(,1,,), (0,0,1,1)$ will 1-sensitize C_{e_9} along ps_{52} and ps_4 . If e_7 s-a-0, then the input sequence $(1,1,,), (0,0,1,1)$ will also 1-sensitize C_{e_9} along ps_{51} and ps_4 . In this example, no initial positioning input sequences are needed.

Location of Multiple Faults

The situation in which multiple faults in a sequential circuit are detected is similar to that of a combinational circuit; the difference being that the paths containing the detectable multiple faults must be embedded in multiple path sequences for the multiple faults to be detected.

Definition 4.13

An s-sensitizable multiple path sequence is a set of path sequences PS_i , each $ps_j \in PS_i$ is to be tested to pass s, and it has the following characteristics.

1. Each $ps_j \in PS_i$ is by itself s-unsensitizable.
2. There exists a set of gates G on PS_i such that the s-type signal propagating on path sequences of PS_i is dominant to every $g \in G$.
3. PS_i is s-sensitizable, i.e., every $ps_j \in PS_i$ is s-sensitizable except for those incident edges belonging to other path sequences of PS_i . These edges are feeding gates of G and will assume the dominant value s in the s-sensitization of ps_j .

4. No subset of PS_i has the above characteristics.

Thus, as in a combinational circuit, multiple faults will occur on multiple path sequences in a sequential circuit. The proof is similar to that of Theorem 3.17 and will not be repeated.

Sensitization Involving Unsensitizable Critical Sensitizing Paths

In the s-sensitization of a path sequence involving unsensitizable paths to provide the necessary sensitizing values, the unsensitizable sensitizing paths can be assumed to be capable of providing the right sensitizing values. This was proved in Theorem 3.18.

Outline of the Diagnosis Process

As in the combinational case, the diagnosis process seeks to establish the fault set F starting from the results of a detection test. Based on the detection results, the four sets P_G , P_B , P_U , P_S can be formed. A diagnosis process will then reduce P_U and P_S to null and thus will completely establish P_G and P_B . From P_B the set of faults F , or its equivalent, will be located.

However, there is one situation in the diagnosis process of a sequential circuit which was not found in the combinational case. If a sequential circuit was found faulty during the initial state positioning for a detection test, then a detection test was never applied to it.

Thus, in this case, the most one can do is to, based on the output vector sequences obtained from the positioning input sequences, assign maximal faulty subcircuits to the output edges of the sequential circuit. And the set P_S will include all the p_i^S or ps_i^S and the other three sets will be null as the starting basis of the diagnosis process.

A suspected path or path sequence can also be indirectly determined as follows.

1. If a set of path sequences was s-maximally sensitized and found faulty of \bar{s} -type, then if all path sequences of the set except one are disproved of \bar{s} -type faults, the remaining path sequence is indirectly confirmed faulty of \bar{s} -type.
2. If a set of path sequences was s-non-maximally sensitized and found faulty of \bar{s} -type, then if all path sequences of the set except one are disproved of \bar{s} -type faults and the dependence set is proved able to pass the critical sensitizing values, the remaining path sequence is indirectly confirmed faulty of \bar{s} -type.
3. If a set of path sequences was s-non-maximally sensitized and found free of \bar{s} -type faults, then if its dependence set is proved able to pass the critical sensitizing values, the set is indirectly disproved faulty of \bar{s} -type.

4. If all paths of a path sequence that can contain \bar{s} -type faults were disproved faulty of \bar{s} -type, the path sequence is indirectly disproved faulty of \bar{s} -type.
5. If a path sequence was confirmed faulty of \bar{s} -type, then if all of its paths except one are disproved faulty of \bar{s} -type, the remaining path is confirmed faulty of \bar{s} -type.
6. If a sequential subcircuit was s -sensitized and found faulty of \bar{s} -type, then if all of its dependence sets are proved able to pass the critical sensitizing values, the set of path sequences defining the sequential subcircuit is indirectly confirmed faulty of \bar{s} -type.

The graphical representation of the diagnosis process for a combinational circuit in Figure 3.7 also describes the diagnosis process of a sequential circuit.

4.6.2 The Path Sequence Checking Fault Diagnosis Algorithm

The diagnosis algorithm proceeds with the basis $B = \{P_G, P_B, P_U, P_S\}$ obtained from the results of a detection test. Given a sequential circuit C with the set of paths P and tested with a detection test t which covers the set of path sequence PS . The basis B is formed as follows.

1. For every path sequence $ps_i \in PS$ s -maximally sensitized with some $\underline{is} \in t$ which detects no \bar{s} -type faults on ps_i , include ps_i^s in P_G .
2. For every path sequence $ps_i \in PS$ which is indirectly disproved of sticking at \bar{s} , include ps_i^s in P_G .
3. For every path sequence $ps_i \in PS$ which is indirectly confirmed of sticking at \bar{s} , include ps_i^s in P_B .
4. Include every path sequence $ps_i \in PS$ as $ps_i^s \in P_U$ if it becomes s -unsensitizable within the fault picture established thus far.
5. Include the remaining path sequences as ps_i^s in P_S for the appropriate s .

If C was found faulty, during an initial state positioning process, include every path sequence $ps_i \in PS$ as ps_i^s , for both values of s , in P_S . The other sets of B are null.

Given a sequential circuit C with the set of paths P . It is found faulty either in the application of an initial state positioning input sequence or from a detection test t which covers the set of path sequences PS . The resulting basis established is $B = \{P_G, P_B, P_U, P_S\}$. A fault diagnosis algorithm to locate all the detectable single and multiple faults, F , is presented as follows.

- A. Find a set of s -sensitizable path sequences $PS_i^s \in PS$ which, except for the s -unsensitizable paths of

a maximal faulty subcircuit C_i , defines C_i . If no such PS_i^s exists, then find such a set which depends on P_U for critical sensitizing paths.

Notice that $s = 0$ or 1 .

- B. Form a set of C_i -sensitizing input-state vector sequences IS_i .
- C. Apply IS_i to C . If C_i is known to be s -sensitized with IS_i , then if C_i is found faulty of \bar{s} -type, move PS_i^s to P_B , or if C_i is found free of \bar{s} -type faults and C_i is defined by one path sequence only, move PS_i^s to P_G . Otherwise mark C_i as tested.
- D. Move every $ps_i^s \in P_S$ to either P_G or P_B if ps_i^s is thus indirectly determined. Also move ps_i^s to P_U if it becomes s -unsensitizable within the existing fault picture.
- E. Repeat A through D until all maximal faulty subcircuits have been tested. A maximal faulty subcircuit tested should not be chosen in step A. Move the remaining $ps_i^s \in P_S$ to P_U .
- F. Test every s -sensitizable multiple path sequence $PS_i^s \in P_U$ for multiple faults. If multiple faults are detected, move PS_i^s to P_B .
- G. Assign any $ps_i^s \in P_U$ to either P_G or P_B .
- H. Remove $ps_i^s \in P_U$ to either P_G or P_B if it is thus indirectly determined.
- I. Repeat G through H until P_U is reduced to null.

J. For every $PS_i^S \in P_B$ which defines a maximal sub-circuit C_i faulty of \bar{s} -type, with the output edge e_i , include $f_i: e_i \text{ s-a-}\bar{s}$ in F . If f_i propagates down some path and stops at e_j assuming the signal value s' , then include $f_j: e_j \text{ s-a-}s'$, instead of f_i , in F . Remove PS_i^S from P_B . Repeat this until P_B is reduced to null.

The validity and convergence of the algorithm is proved in the next theorem.

Theorem 4.12

The path sequence checking fault diagnosis algorithm for the sequential circuit will locate all the detectable single and multiple faults of the circuit in a finite number of steps.

Proof: A fault in a sequential circuit is detectable if it can be sensitized to an output edge of the circuit within the given fault picture. The rest of the proof is similar to that of Theorem 3.19; the only difference is that one works with the path sequences which cover all the paths of the circuit in this case, rather than the paths of the circuit. Q.E.D.

4.6.3 An Example

Example 4.8

Continue with the results obtained from the detection test in Example 4.6, one finds the following basis,

$$P_G = \{ps_{22}^0, ps_{42}^0, ps_{92}^0, ps_{41}^1, ps_{91}^1, p_3^1, ps_{92}^1\},$$

where ps_{22}^0 and ps_{42}^0 are determined from \underline{is}_2 , ps_{92}^0 from \underline{is}_5 and ps_{41}^1 , ps_{91}^1 from \underline{is}_{11} , p_3^1 from ps_{42}^1 , and ps_{92}^1 from ps_{91}^1 ,

$$P_S = \{p_1^0, p_3^0, ps_{41}^0, p_5^0, ps_6^0, p_7^0, ps_8^0, ps_{91}^0, p_{10}^0, p_1^1, ps_{22}^1, p_5^1, ps_6^1, p_7^1, ps_8^1, p_{10}^1\},$$

$$P_B = P_U = \emptyset.$$

Since e_{z1} and e_{z2} are free of faults, the maximal faulty sequential subcircuits are found as follows.

$$P_{17}^0 = \{p_1^0, p_3^0\} \text{ defining } C_{17}$$

$$P_{18}^0 = \{p_5^0, p_7^0\} \text{ defining } C_{18}$$

$$PS_{16}^0 = \{p_{10}^0, ps_{41}^0, ps_{21}^0, ps_{91}^0\} \text{ defining } C_{16}$$

$$PS_7^0 = \{ps_6^0, ps_8^0\} \text{ defining } C_7$$

$$PS_1^1 = \{p_1^1, ps_{21}^1, ps_{22}^1\} \text{ defining } C_1$$

$$PS_9^1 = \{p_5^1, ps_6^1\} \text{ defining } C_9$$

$$PS_{10}^1 = \{p_7^1, ps_8^1\} \text{ defining } C_{10}$$

$$P_4^1 = \{p_{10}^1\} \text{ defining } C_4$$

To start the diagnosis procedure from the outputs of the circuit, C_{17} will be determined first. The fault-sensitizing input-state vector sequence for C_{17} is

$$\begin{aligned} \underline{is}_{18} &= (1,1,1,,,,), (0,0,1,,1,1,1), \\ &= v_{181}, v_{182}, \end{aligned}$$

with \underline{is}_{18} : $p_1^0, p_3^0 / \{p_7^1\}$.

The output vector sequence obtained from the application of \underline{is}_{18} to C is

$$o_{18} = (,), (0,1),$$

which indicates no detection of faults. Hence the basis is not updated. For C_{18} , the input-state vector sequence and the resulting output sequence are

$$\underline{is}_{19} = v_{181}, (1,0,0,,1,1,1) = v_{181}, v_{19},$$

$$\underline{is}_{19}: p_5^0, p_7^0 / \{p_1^1\},$$

$$o_{19} = (,), (0,1).$$

Again no faults are detected and the basis is not updated due to the undetermination of p_1^1 .

For C_{16} , there are two fault-sensitizing input-state vector sequences,

$$IS_{16} = \{\underline{is}_2, \underline{is}_{20}\},$$

$$\underline{is}_{20} = (1,1,0,\dots), (\dots,0,1,0,1) = v_{201}, v_{202},$$

$$\underline{is}_2: p_{10}^0, ps_{41}^0, ps_{21}^0 / \{ps_8^1\},$$

$$\underline{is}_{20}: p_{10}^0, ps_{91}^0 / \{ps_6^1\}.$$

The input-state vector sequence \underline{is}_2 was used in the detection test. The output vector sequence from \underline{is}_{20} is

$$o_{20} = (,), (,0).$$

No faults are detected and again the basis is not updated.

For C_7 , the fault-sensitizing vector sequence and the resulting output vector sequence are

$$\underline{is}_{21} = (1,0,0,\dots), (\dots,1,1,0,0) = v_{211}, v_{212}$$

$$\underline{is}_{21}: ps_6^0, ps_8^0 / \{p_{10}^1\},$$

$$o_{21} = (,), (,0).$$

Again no faults are detected and the basis is not updated.

For C_1 , the fault-sensitizing vector sequence and the output vector sequence are

$$\begin{aligned}\underline{is}_{22} &= v_{181}, (1,0,1,,1,1,1), (1,1,1,0,1,1,1), \\ &= v_{181}, v_{221}, v_{222}, \\ o_{22} &= (,), (0,1), (0,0).\end{aligned}$$

Hence the results indicate that ps_{22} s-a-0 is confirmed. This in turn indirectly confirms p_1 s-a-0. Because of p_1 s-a-0, p_5 , p_7 become 0-unsensitizable. Also p_7 becomes 1-unsensitizable. This also in turn makes p_1 and p_3 0-unsensitizable. Thus the basis is updated as follows.

P_G : unchanged

$$P_S = \{ps_{41}^0, ps_6^0, ps_8^0, ps_{91}^0, p_{10}^0, p_5^1, ps_6^1, ps_8^1, p_{10}^1\}$$

$$P_B = \{p_1^1, ps_{22}^1\}$$

$$P_U = \{p_5^0, p_7^0, p_7^1, p_1^0, p_3^0\}$$

The fault sensitizing vector sequence for C_{10} is

$$\begin{aligned}\underline{is}_{23} &= v_{181}, v_{221}, (,,,1,1,1,1), \\ &= v_{181}, v_{221}, v_{23}.\end{aligned}$$

The resulting output vector sequence is

$$o_{23} = (,), (0,1), (0,0).$$

Since \underline{is}_{23} : $ps_8^1/\{p_{10}^1\}$, so the basis is unchanged.

For C_9 , the fault-sensitizing vector sequence is

$$\begin{aligned}\underline{is}_{24} &= v_{181}, (1,1,0,,1,1,1), (,,1,1,0,1), \\ &= v_{181}, v_{241}, v_{242}.\end{aligned}$$

The output vector sequence obtained is

$$o_{24} = (,), (1,1), (,0)$$

which indicates no detection of faults on p_5 and detection of faults on ps_6 . But since p_5 is 1-maximally sensitized with \underline{is}_{24} , so p_5 passing 1 is confirmed. Which in turn indirectly confirms ps_6 passing 1.

Also indirect determination finds $ps_{91}^0, p_{10}^0 \in P_G$ which in turn finds $p_{10}^1 \in P_B$. Again ps_6 and ps_8 become 0-unsensitizable, ps_8 becomes 1-unsensitizable due to $p_{10}^1 \in P_B$. And ps_{41} becomes 0-unsensitizable because of $ps_8^1 \in P_U$. Hence the basis is updated as follows.

$$P_G = \{ps_{22}^0, ps_{42}^0, ps_{92}^0, ps_{41}^1, ps_{91}^1, p_3^1, ps_{92}^1, p_5^1, ps_6^1, ps_{91}^0, p_{10}^0\}$$

$$P_B = \{p_1^1, ps_{22}^1, p_{10}^1\}$$

$$P_U = \{p_5^0, p_7^0, p_1^0, p_3^0, p_7^1, ps_6^0, ps_8^0, ps_{41}^0, ps_8^1\}$$

Of the set P_U , the following multiple path sequences can be formed.

$$PS_{m1} = \{p_3, p_5\}$$

$$PS_{m2} = \{ps_{41}, ps_6\}$$

The fault sensitizing input-state vector sequence testing PS_{m1}^0 is

$$\underline{is}_{m1} = v_{181}, (0,0,0,,1,1,1) = v_{181}, v_{m1}.$$

The resulting output vector sequence is

$$o_{m1} = (,), (0,).$$

Thus no multiple faults are detected.

For PS_{m2} , the fault-sensitizing input-state vector sequence testing PS_{m2}^0 and the resulting output vector sequence are

$$\underline{is}_{m2} = (0,0,1,,,,), (0,0,1,0) = v_{m21}, v_{m22},$$

$$o_{m2} = (,), (,0).$$

Also no multiple faults are detected.

Now different assignments of elements of P_U to either P_G or P_B will result in equivalent sets of faults. If one assigns $p_1^0 \in P_B$, then $p_3^0 \in P_B$ is determined. Because of these assignments, p_5 and p_7 become 0-sensitizable and p_7 becomes 1-sensitizable and hence $p_5^0, p_7^0 \in P_G$, $p_7^1 \in P_B$ are obtained. And no multiple faults are created. Also if one assigns $ps_6^0 \in P_B$, then $ps_8^0 \in P_B$ is determined. Also $ps_{41}^0 \in P_G$ results because of $ps_8^0 \in P_B$.

Finally ps_8^1 can be randomly assigned since ps_8 is still 1-unsensitizable. Let $ps_8^1 \in P_B$. Hence the basis resulting from these assignments looks as follows.

$$P_G = \{ps_{22}^0, ps_{42}^0, ps_{92}^0, ps_{41}^1, ps_{91}^1, p_3^1, \\ ps_{92}^1, p_5^1, ps_6^1, ps_{91}^0, p_{10}^0, p_5^0, p_7^0, \\ ps_{41}^0\}$$

$$P_B = \{p_1^1, ps_{22}^1, p_{10}^1, p_1^0, p_3^0, p_7^1, ps_6^0, \\ ps_8^0, ps_8^1\}$$

The set of faults obtained from P_G is

$$F_T = \{f_1, f_{10}, f_4, f_{17}, f_7\},$$

where

$$f_{17}: e_{17} \text{ s-a-l},$$

$$f_7: e_7 \text{ s-a-l}.$$

Since both f_7 and f_{17} are not detectable in the presence of F , hence both sets of faults, F_T and F , are equivalent.

4.7 Characteristics of the Method as Applied to Sequential Circuits

The observations of Section 3.6 regarding the path checking method as applied to combinational circuits similarly apply to sequential circuits. In addition to these observations, the following further observations can

be made of the path checking method as applied to sequential circuits.

First, the fault analysis proceeds at the circuit level, rather than at the state diagram level. Thus one does not have to be concerned with the various characteristics of the sequential machine, i.e., state accessibility, equivalency of states, etc.

Second, this method is general enough to cover a wide range of sequential circuits. The only condition is that every path of the circuit under test is sensitizable. If the method as presented in this thesis is extended to cover sequential circuits containing redundant circuitry, then the method will be applicable to practically all types of sequential circuits found in current fault analysis studies.

CHAPTER V

CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

5.1 Conclusions

This thesis presents a method to detect and locate stuck-at type logical faults in switching circuits by checking every path of the circuits for its ability to pass both signals. The method approaches both combinational and sequential circuits from the same "path" viewpoint, with the sequential circuit considered as an extension of the combinational circuit.

Based on the path checking approach, two detection algorithms are developed for the generation of detection tests for combinational and sequential circuits. The results of a detection test can then be used as a basis for the fault diagnosis algorithms. The checking of maximal faulty subcircuits is the basic approach used in developing the two fault diagnosis algorithms for combinational and sequential circuits. The fault diagnosis process is more involved than the fault detection process because of its dependence on the current fault pictures as found throughout the process. However, the fault diagnosis algorithms can

terminate at any point during the diagnosis process if the corresponding degree of fault diagnosis is satisfactory.

Since the detection process is straightforward, the detection algorithms can be easily implemented. The implementation of the diagnosis algorithms, however, is more complicated due to the dependence of the diagnosis algorithms on the current fault picture.

The method serves as a starting point for fault analysis of sequential circuits at the circuit level. The various difficulties and the resulting restrictions encountered during the development of the method are discussed as follows for further research.

5.2 Suggestions for Future Work

One obvious extension of the method that needs further research is to allow unsensitizable and partially sensitizable paths in the switching circuits considered. This extension allows, in effect, redundant circuitry to be added to the circuits and thus covers most types of switching circuits that are found in current fault analysis studies. Unsensitizable and partially sensitizable paths were encountered during the diagnosis process as a result of other faults in the circuit. The difficulties of sensitizing these paths were solved, in this thesis, by treating these paths as multiple paths. Thus, it seems that the concept of "multiple paths" is one approach to check the redundant parts of the switching circuits.

The EXCLUSIVE OR gates were excluded from consideration in this thesis not because of any unsolved difficulties with them, but because of the need for extra attention to paths containing them. A path going through an EXCLUSIVE OR gate is always sensitized, at the gate, regardless of the signal value on the other input edge of the gate. Thus it is necessary for the other input edge of the EXCLUSIVE OR gate to assume the same signal value, throughout the computation of a test, in order for the path in question to be validly sensitized. This will hence modify the procedures for designing the tests as presented in the thesis.

Further studies need to be done on the various relations and interpretations of the various characteristics at the sequential machine level in terms of the path structure at the circuit level. For example, equivalence of states in a sequential machine can be interpreted in terms of path sensitization in its realized circuits.

Finally, more research needs to be done to adopt this method to integrated circuits. When individual component circuits are combined to form integrated circuits, unsensitizable or partially sensitizable paths may be created in the integrated circuits even though all the individual component circuits contain sensitizable paths only, as shown in Figure 5.1. In the figure the paths $p_1 = i_a, e_c, e_e, z$, $p_2 = i_b, e_d, e_f, z$ are 1-

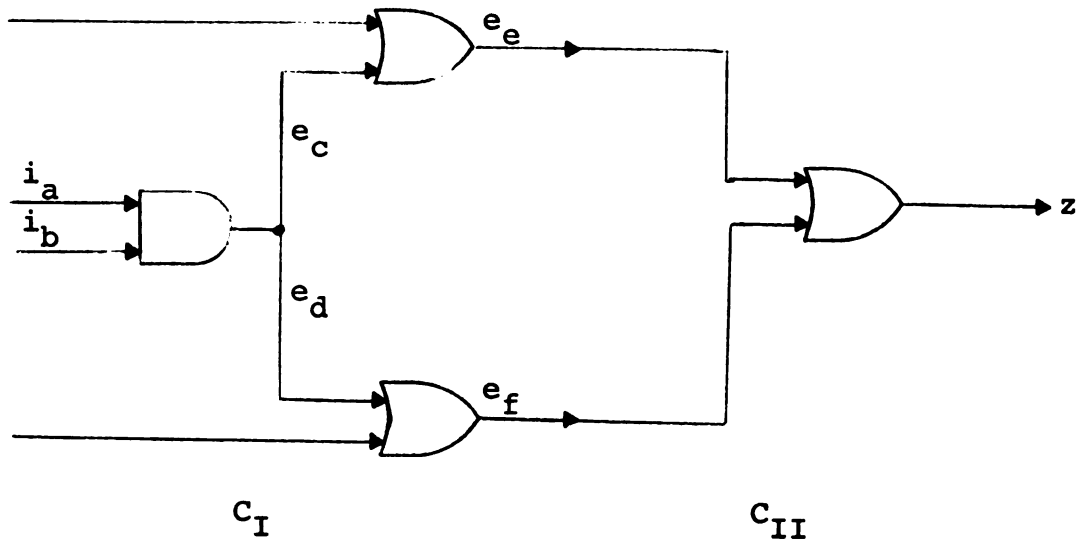


Fig. 5.1.--Partially sensitizable paths result from concatenating circuits C_I and C_{II} to form an integrated circuit.

unsensitizable. The multiple path approach might be used to solve this problem. Also it might be interesting to investigate the methods to combine individual detection tests for the component circuits to form "integrated tests" for the resulting integrated circuits. This could prove to be a worthy effort.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Armstrong, D. B. "A General Method of Applying Error Correction to Synchronous Digital Systems." Bell System Tech. J., Vol. 40, pp. 577-593.
2. Breuer, M. A. "Testing for Intermittent Faults in Digital Circuits." IEEE Trans. Comput., Vol. C-22, pp. 241-246, March 1973.
3. Carter, W. C. "Fault-Tolerant Computing: An Introduction and a Viewpoint." IEEE Trans. Comput., Vol. C-22, pp. 225-228, March 1973.
4. Chang, H. Y., Manning, E., and Metze, G. Fault Diagnosis of Digital Systems. New York: Wiley-Interscience, 1970.
5. Chang, L. "Fault Analysis of Combinational Logic Networks." Ph.D. Thesis, Michigan State University, 1974.
6. Clegg, F. W. "Algebraic Properties of Faults in Logics Networks." Ph.D. Thesis, Stanford University, 1970.
7. Du, M., and Weiss, D. "Multiple Fault Detection in Combinational Circuits: Algorithms and Computational Results." IEEE Trans. Comput., Vol. C-22, pp. 235-240, March 1973.
8. Farmer, D. E. "Algorithms for Designing Fault-Detection Experiments for Sequential Machines." IEEE Trans. Comput., Vol. C-22, pp. 159-167, February 1973.
9. Friedman, A. D. "Faults Detection in Redundant Circuits." IEEE Trans. Electron. Comput., Vol. EC-16, pp. 99-100, February 1967.
10. Harrison, M. A. "An Analysis of Errors in Finite Automata." Inform. Contr., Vol. 8, pp. 435-450, August 1965.

11. Hartmanis, J., and Stearns, R. E. "A Study of Feedback and Errors in Sequential Machines." IEEE Trans. Electron. Comput., Vol. EC-12, pp. 223-232, June 1963.
12. Hartmanis, J., and Stearns, R. E. Algebraic Structure Theory of Sequential Machines. Englewood Cliffs, N.J.: Prentice-Hall, 1966.
13. Hecht, H. "Figure of Merit for Fault-Tolerant Space Computers." IEEE Trans. Comput., Vol. C-22, pp. 246-250, March 1973.
14. Hennie, F. C. "Fault Detecting Experiments for Sequential Circuits." Proc. 5th Annual Symp. Switching Theory and Logical Design, November 1964.
15. Hsieh, E. P. "Checking Experiments for Sequential Machines." IEEE Trans. on Computers, Vol. C-20, pp. 1152-1168, October 1971.
16. Kamal, S. "Diagnosis of Intermittent Faults in Digital Systems." Ph.D. Thesis, Michigan State University, 1973.
17. Kime, C. R. "A Failure Detection Method for Sequential Circuits." Dept. of Elect. Eng., Tech. Rep. 66-13, University of Iowa, 1966.
18. Kohavi, I., and Kohavi, Z. "Variable-Length Distinguishing Sequences and Their Applications to the Design of Fault-Detection Experiments." IEEE Trans. Comput. (Short Notes), Vol. C-17, pp. 792-795, August 1968.
19. Kohavi, Z., and Lavalley, P. "Design of Sequential Machines with Fault-Detection Capabilities." IEEE Trans. Electron. Comput., Vol. EC-16, pp. 473-484, August 1967.
20. Kriz, T. A. "Machine Identification Concepts of Path Sensitizing Fault Diagnosis." IEEE Conf. 10th Annual Symposium on Switching and Automata Theory, pp. 171-181.
21. Lyons, R. E., and Vanderkulk, W. "The Use of Triple-Modular Redundancy to Improve Computer Reliability." IBM J. Res. Dev., Vol. 6, pp. 200-209, April 1962.
22. Martin, R. L. "The Design of Diagnosible Sequential Machines." Proc. Hawaii Internat. Conf. Syst. Sci., 1968.

23. McCluskey, E. J., Jr. Introduction to the Theory of Switching Circuits. New York: McGraw-Hill, 1965.
24. McCluskey, E. J., Jr., and Clegg, F. W. "Fault Equivalence in Combinational Logical Networks." IEEE Trans. Comput., Vol. C-20, pp. 1286-1293, November 1971.
25. Meyer, J. F. "Fault Tolerant Sequential Machines," IEEE Trans. Comput., Vol. C-20, pp. 1167-1177, October 1971.
26. Meyer, J. F. "On the Structure of Fault Tolerant Sequential Machines." Proc. 3rd Hawaii Int. Conf. Sci. (Part I), pp. 443-447, January 1970.
27. Miller, R. E. Switching Theory Vol. II: Sequential Circuits and Machines. New York: John Wiley & Sons, Inc., 1965.
28. Moore, E. F., and Shannon, C. E. "Reliable Circuits Using Less Reliable Relays." J. Franklin Inst., Vol. 262, pp. 191-208, September 1956, and pp. 281-297, October 1956.
29. Moore, E. F. "Gedanken Experiments on Sequential Machines." Automata Studies, Annals of Mathematics Studies, no. 34, pp. 129-153. Princeton, N.J.: Princeton University Press, 1956.
30. Peterson, W. W. Error-Correcting Codes. New York: Wiley and Cambridge, Mass.: M.I.T. Press, 1961.
31. Pierce, W. H. Failure-Tolerant Computer Design, Chapt. III and IV. New York: Academic Press, 1965.
32. Poage, J. F. "Derivation of Optimal Tests to Detect Faults in Combinational Circuits." Proc. Symp. on Mathematical Theory of Automata, Polytechnic Institute of Brooklyn, pp. 483-528, 1963.
33. Poage, J. F., and McCluskey, E. J., Jr. "Derivation of Optimum Test Sequences for Sequential Machines." Proc. 5th Annual Symp. Switching Theory and Logical Design, 1964.
34. Ramamoorthy, C. V. "Fault-Tolerant Computing: An Introduction and An Overview." IEEE Trans. Comput., Vol. C-20, pp. 1241-1244, November 1971.

35. Roth, J. P. "Diagnosis of Automata Failures: A Calculus and a Method." IBM J. Res. Devel., Vol. 10, pp. 278-291, July 1966.
36. Sellers, F. F., Hsiao, M. Y., and Bearnson, L. W. "Analyzing Errors with the Boolean Difference." IEEE Trans. Comput., Vol. C-17, pp. 676-683, July 1968.
37. Seshu, S., and Freeman, D. N. "The Diagnosis of Asynchronous Sequential Switching Systems." IRE Trans. on Electronic Computers, Vol. EC-11, pp. 459-465, 1962.
38. Su, S. Y. H., and Chang, S. Y., and Breuer, M. A. "Location of Multiple Stuck-Type Faults in Combinational Networks." IEEE Comput. Soc. Reposit. R-72-196, 1972.
39. Tryon, J. G. "Quaded Logic." Redundancy Techniques for Computing Systems, pp. 205-228. Washington, D.C.: Spartan Books.
40. Von Neumann, J. "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components." Automata Studies, Annals of Mathematics Studies, no. 34, pp. 43-98. Princeton University Press, 1956.

