INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure complete continuity.

2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed, a definite method of "sectioning" the material has been followed. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.

University Microfilms International
300 N. Zeeb Road
Ann Arbor, MI 48106

Order Number 1334301

Analysis of approaches to synchronous faults simulation by surrogate propagation

Lee, Chang-Hwa, M.S.

The University of Arizona, 1988

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark ✓.

1. Glossy photographs or pages ______
2. Colored illustrations, paper or print ______
3. Photographs with dark background ______
4. Illustrations are poor copy ______
5. Pages with black marks, not original copy ✓
6. Print shows through as there is text on both sides of page ______
7. Indistinct, broken or small print on several pages ✓
8. Print exceeds margin requirements ______
9. Tightly bound copy with print lost in spine ______
10. Computer printout pages with indistinct print ______
11. Page(s) ______ lacking when material received, and not available from school or author.
12. Page(s) ______ seem to be missing in numbering only as text follows.
13. Two pages numbered ______. Text follows.
14. Curling and wrinkled pages ______
15. Dissertation contains pages with print at a slant, filmed as received ______
16. Other ______

UMI

ANALYSIS OF APPROACHES TO SYNCHRONOUS FAULTS SIMULATION BY SURROGATE PROPAGATION

BY

Chang-Hwa Lee

A Thesis Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
for the Degree of

MASTER OF SCIENCE
WITH A MAJOR IN ELECTRICAL ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

1 9 8 8

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowlegment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department of Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Chang-Hwa Lee

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Fredrick J. Hill — 26 July 1988

Fredrick J. Hill
Professor of
Electrical and Computer Engineering

Date

ACKNOWLEDGMENTS

Most of the credit for the research goes to professor Fredrick J. Hill to whom I am most indebted not only for technical advice but also for his patience throughout this research. I also would like to thank my family for their support and encouragement.

TABLE OF CONTENTS

PAGE

LIST OF ILLUSTRATIONS 6

LIST OF TABLES 8

ABSTRACT 9

CHAPTER

1. INTRODUCTION 10

1.1 Objectives 10

1.2 SCIRTSS 12

2. FAULT IN DIGITAL CIRCUITS 17

2.1 Fault Model 17

2.2 Fault Simulation 20

3. ILLUSTRATION OF SYNCHRONOUS FAULTS SIMULATION BY SURROGATE WITH EXCEPTIONS. 22

3.1 Subnetworks 22

3.2 Forward Simulation for Good Network 23

3.3 Forward Simulation for Reconvergent Fan-out Point 24

3.3.1 Fan-out Free Region 24

3.3.2 F.O.S. Sensitivity 29

3.4 Backward to Search the Sensitive Paths 33

3.5 Identify the Compound Faults and Multiply Stored Faults 37

TABLE OF CONTENTS--Continued

CHAPTER PAGE

4. FAULT COLLAPSING 45

5. INPUT DATA 50

5.1 AHPL for RNDPRU.SCR 50

5.2 AHPL for ADD8X4.SCR 52

5.3 AHPL for ADD8X8.SCR 54

5.4 AHPL for ADD8X16.SCR 56

5.5 AHPL for TEST88.SCR 59

5.6 AHPL for MSFCF4.SCR 62

6. SIMULATION RESULTS AND COMPARISON . . . 65

6.1 Circuit RNDPRU.SCR 66

6.2 Circuit ADD8X4.SCR 70

6.3 Circuit ADD8X8.SCR 73

6.4 Circuit ADD8X16.SCR 76

6.5 Circuit TEST88.SCR 79

6.6 Circuit MSFCF4.SCR 82

7. SUMMARY AND SUGGESTIONS FOR FURTHER WORK . 85

APPENDIX 1: CIRCUIT DESCRIPTION OF ADD8X8.SCR. . 87

8. REFERENCES 113

LIST OF ILLUSTRATIONS

FIGURE		PAGE
1.1	SCIRTSS Block Diagram	15
1.2	Generating the SCIRTSS Circuit Data Base . .	16
2.1	Circuit of CMOS Nor Gate	19
2.2	Main Tasks of Fault Simulation	21
3.1	Nondisjoint Partition of Network	26
3.2	The FFR's of Subnetwork	27
3.3	Sensitivity of Node in an FFR	28
3.4	Parallel Computation of FOS Sensitivity . .	31
3.5	Backward Propagation of Node Sensitivity . .	35
3.6	Backward to Search the Sensitive Paths . . .	36
3.7	Reconvergent Multiply Stored Fault	41
3.8	Compound Fault	42
3.9	Computation of Multiply Stored Faults . . .	44
5.1	Cascade of Adders	61

LIST OF ILLUSTRATION--Continued

FIGURE PAGE

6.1 Time Division of SFSSE Simulation 65

6.2 Total Time for Clock Period of SFSSE Simulation for Circuit RNDPRU.SCR 69

6.3 Total Time for Clock Period of SFSSE Simulation for Circuit ADD8X4.SCR 72

6.4 Total Time for Clock Period of SFSSE Simulation for Circuit ADD8X8.SCR 75

6.5 Total Time for Clock Period of SFSSE Simulation for Circuit ADD8X16.SCR 78

6.6 Total Time for Clock Period of SFSSE Simulation for Circuit TEST88.SCR 81

6.7 Total Time for Clock Period of SFSSE Simulation for Circuit MSFCF4.SCR 84

LIST OF TABLES

TABLE PAGE

3.1 Coding of Node Values 32

6.1 Time Comparison Between SCIRTSS and SFSSE for Circuit RNDPRU.SCR 68

6.2 Time Comparison Between SCIRTSS and SFSSE for Circuit ADD8X4.SCR 71

6.3 Time Comparison Between SCIRTSS and SFSSE for Circuit ADD8X8.SCR 74

6.4 Time Comparison Between SCIRTSS and SFSSE for Circuit ADD8X16.SCR 77

6.5 Time Comparison Between SCIRTSS and SFSSE for Circuit TEST88.SCR 80

6.6 Time Comparison Between SCIRTSS and SFSSE for Circuit MSFCF4.SCR 83

ABSTRACT

This thesis describes a new simulation technique, Synchronous Faults Simulation by Surrogate with Exception, first proposed by Dr. F. J. Hill and has been initiated under the direction of Xiolin Wang. This paper reports early results of that project.

The Sequential Circuit Test Sequence System, SCIRTSS, is an automatic test generation system which is developed in University of Arizona will be used as a target to compare against the results of the new simulator.

The major objective of this research is to analyze the results obtained by using the new simulator SFSSE against the results obtained by using the parallel simulator SCIRTSS. The results are listed in this paper to verify superiority of the new simulation technique.

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

The process of development of a complete test for a digital circuit is usually divided into Test Generation and Verification by Fault Simulation which may or may not be performed iteratively.

The systematic procedure for Test Generation described in the recent literature calls for generating test vectors manually or for single faults using either the D-algorithm or PODEM. The effectiveness of these two methods are measured by Fault Simulation performed on individual combinational logic networks only.

Fault simulation is an integral part of a complete test. Due to the advance of LSI and the concept of scannable register-to-register realization of sequential logic, large combinational logic networks with hundreds to thousands of gates are being implemented with increasing frequency. The basic techniques for a clock mode fault simulator apply to all of the simulators currently used for simulating combinational logic, some of which have already been used in various forms of sequential circuit fault simulation. These are introduced below with their

advantages and disadvantages.

1. Parallel Simulation : It takes advantage of the prevalent word-oriented operations in today's computer architecture. each fault corresponds to a bit position in a word. Simulation iterates with a word-full of faults at a time. However, it becomes cumbersome to take advantage of the decreasing number of untested faults as test pattern application continuous over time.

2. Deductive Simulation : At the time of simulating each gate the list of all faulty networks for which the output of that gate differs from the good network is determined. There are two problems associated with the deductive simulation method. The fault set determination at each gate requires set operations with large overhead; and the fault sets along the paths to the output can be quite large, presenting a storage and access problems.

3. Concurrent Simulation : As each gate of the good network is simulated, a representative gate of each faulty network for which the value of any input differs from the good network value is constructed, connected, and simulated as well.

This thesis will present a new approach, Synchronous Faults Simulation by Surrogate with Exceptions (SFSSE), first proposed by Dr. F.J. Hill. A project calling for implementation and comparison of several versions of SFSSE has been initiated under the direction of Xiolin Wang. This thesis reports early results of that project.

1.2 SCIRTSS

The SCIRTSS is an automatic test generation system for sequential circuit. Figure 1.1 illustrates the basic flow diagram of SCIRTSS. This system accepts a functional description of a digital circuit which described in a hardware programming language, AHPL [4]. AHPL is a clocked mode, register transfer type language which can be used to describe synchronous and asynchronous sequential circuits. The program written by AHPL first compiled into a logic gate level description which is a format for interface with SCIRTSS.

The input of test vector for simulation is based on the concatenation of an input sequence found by an application of two separate heuristic tree searches. (see Figure 1.1 block B and C) Block B is a single sensitization search which finds an input sequence causes a particular network fault to be driven into a flip_flop or primary

outputs. Block C is a single fault propagation search which tries to find an input sequence propagate a stored fault from its original point to either primary outputs or secondary outputs. Once such an input sequence has been found, the parallel fault simulation is applied. Not only is this segment verified for a single fault in question, but all other faults whose effects are driven to the circuit output by that segment are identified by the simulator and reduced from the fault list.

Simulation of a sequence segment will usually drive the effects of other faults into memory elements. Thus the loop blocks A and C is self perpetuating until about 90% of the faults are found. Eventually user may use a modified D-algorithm to generate a set of test vectors if a sensitive path exits. This input vector then detected more faults if possible.

To run SCIRTSS, the user prepared a parameter file which consists of circuit dependent information and heuristic weights for guiding the searches. SCIRTSS will be used iteratively until the time limit, specified by the user, is reached or some error occurs. The user may save the status of the circuit at end of each run and use the stored status as an initial situation for the next run. The output file produced by SCIRTSS contains detected

faults in each clock and information used in the process of simulation. Therefore, the user may catch the simulating process and make a adjustment for the use of SCIRTSS next time.

Although SCIRTSS has been recently upgraded to the status of a very convenient to use test generation tool and a tool capable of handling potential VLSI designs, the running time of simulation in each clock period has not changed. The reason for that is the parallel simulation processes all faults, which are represented by computer's words, for every clock period.

Since the new simulator, SFSSE, has been implemented and proved that the new simulator enjoyed a significant speed advantage over the parallel simulator, the automatic test generation system, SCIRTSS, will eventually include the new simulator in place of the parallel simulator.

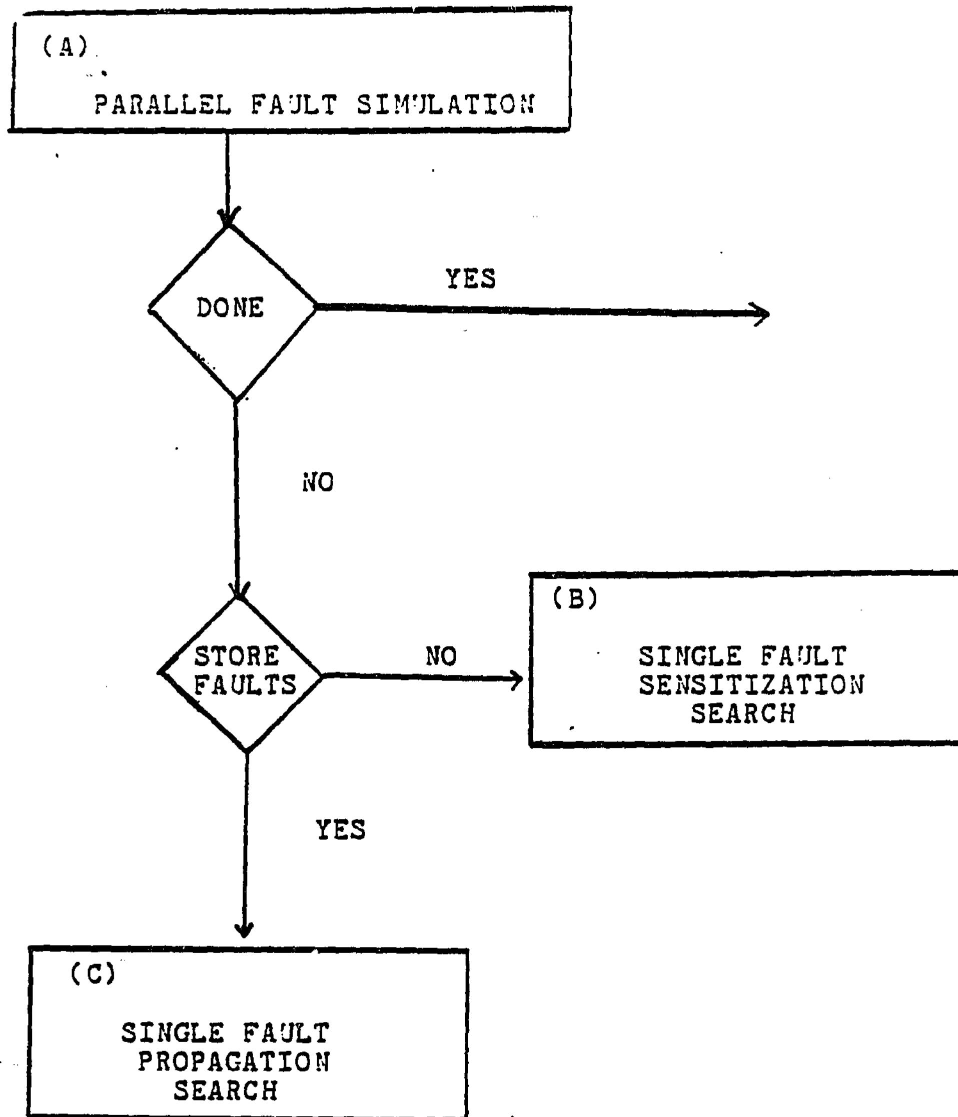


Figure 1.1 SCIRTSS

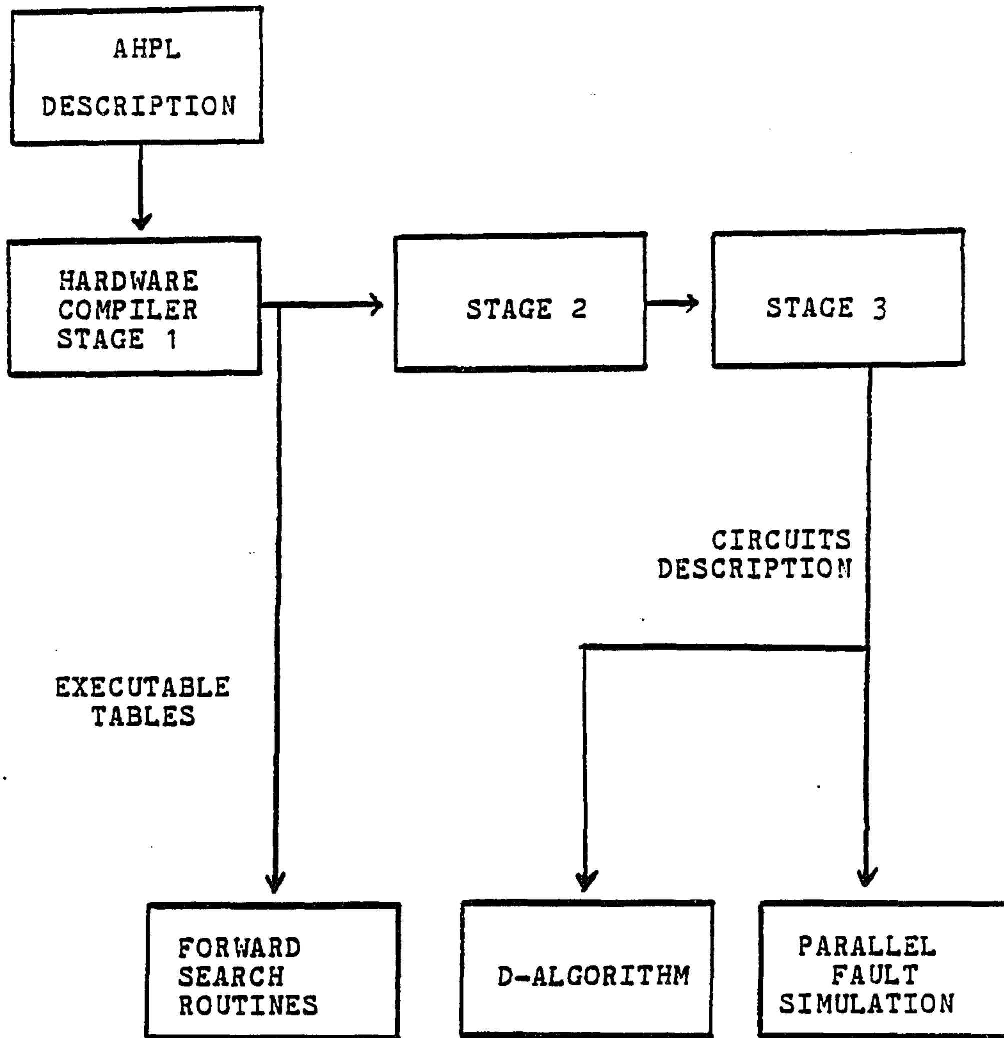


Figure 1.2 Generating the SCIRTSS Circuit Data Base

CHAPTER 2

FAULTS IN DIGITAL CIRCUITS

If a network deviates from its specified behavior, then a failure is said to have occurred. The most common causes of failure in a network are physical defects. The term fault is used to describe a physical defect which may or may not cause failure. In a network, faults may occur due to defective components, lines shorted to ground,lines shorted to power supply, or lines shorted to other lines "short circuited". other faults worth mentioning are the MOS physical failures, known as transistor stuck-open and stuck-short.

2.1 FAULT MODEL

Many defects can occur in digital products during manufacturing. These defects can vary greatly, depending on the process technology used. Existing solutions do not model each and every manufacturing defect. Instead they rely on simplified failure models (which has been used with great success in the industry). The most popular of these models is the "single stuck-at fault", which assumes that a product can contain at most one manufacturing defect, and that this defect can be represented as a single

wire staying fixed at a high voltage (a stuck-at one), or staying fixed at low voltage (a stuck-at zero). For the remainder of this chapter and the following chapters, the notation SA1 and SA0 will be adopted to represent these two models respectively. Another fault that can be attributed to manufacturing defects is the existence of an undesirable connection between two of more lines. Such situation leads to a "fighting" between lines with different logical values, which may result in one logical value dominating the other or both lines assuming unknown values. This type of fault, known as a "bridge fault", needs special treatment and hence will not be discussed beyond this point. The last type of faults to be introduced here are peculiar to MOS technology. These are transistor stuck-short, and stuck-open. Figure 2.1 shows a CMOS "nor" gate with pull-up and pull-down network conducts, but not both. A transistor is stuck-short when both networks conduct simultaneously. The defect of the stuck-short transistor on the gate output line is similar to that of a bridge fault, henceforth, it will not be considered. On the contrary, a transistor is stuck-open,when both the pull up and the pull down networks are simultaneously not conducting. In this case, the output of the gate assumes its previous logical value.

Figure 2.1 Circuit of CMOS NOR Gate

2.2 FAULT SIMULATION

Fault simulation is the process of imitating the behavior of a network in response to an input stimulus, in the existence of a fault. The resultant response is then compared with the expected one. If the responses are identical for all possible stimulii, then the fault will have no effect on the network, and is said to be undetectable. On the other hand, if the response is different for a stimulus, then the presence of the fault is said to be detectable by that stimulus. The stimulus is called a test for that fault. In general terms, fault simulation is the process of determining how manufacturing defects alter the behavior of a logic network.

In fault simulation there are, in addition, the requirements for accurate fault modeling capability, and fast running times for very large circuits and fault sets.

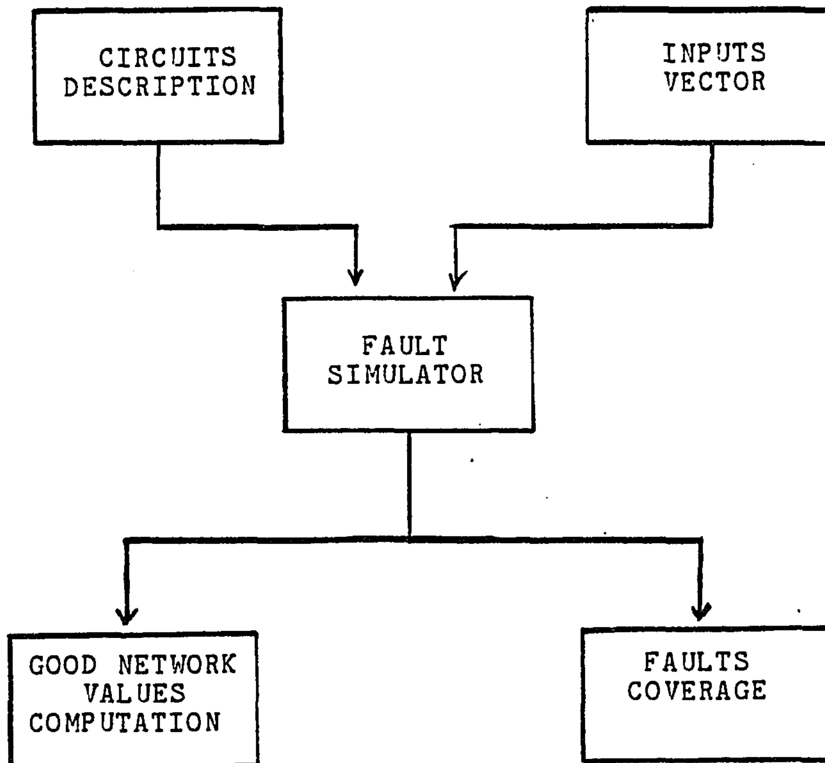


Figure 2.2 Main Tasks of Fault Simulation

CHAPTER 3

SYNCHRONOUS FAULTS SIMULATION BY SURROGATE WITH EXCEPTIONS

3.1 SUBNETWORKS

For the purpose of developing an overall simulation strategy. A common approach to solve a problem is to partition the problem into smaller parts, find solutions for the parts, and then combine the solutions for the parts into a solution for the whole. This approach , especially when used recursively, will give a complete solutions to problems in which the subproblems are smaller versions of the problem. The problem to be solved in this paper is that of finding an efficient algorithm for fault simulation of a logic network. If the network is divided into independent but not necessarily disjoint subnetworks, and every subnetwork is divided into disjoint regions, then an algorithm designed for a region can be generalized for the whole network.

Subnetworks will correspond to each primary and secondary outputs. As in classical sequential circuits, primary outputs refer to the external outputs of the whole circuit; secondary outputs refer to connections from the combinational logic network to memory element inputs.

3.2 FORWARD SIMULATION FOR GOOD NETWORK

After the whole circuit has been partitioned to many subnetworks depending on the total number of the primary and secondary outputs, the next step is to calculate the good values for each node in the subnetwork.

In the parallel simulator, suppose there are N inputs for the AND gate , it takes N * (N-1)/2 times to accomplish the computation of the output value. However in this research, the all inputs is put into a vector which will be compared with the vector FFFFFFFF for AND gate. If these two vectors are the same, then the output of the AND gate is logic 1; otherwise logic 0. Similarly, the input vector will be compared with the vector 00000000 for an OR gate. If these two vectors are the same ,then the output of the OR gate is logic 0; otherwise logic 1. Therefore the new simulator needs only one operation to compute the output value of the gate. the speed of computation for the output of node is faster than that of parallel simulator. The same process is also fit for the NAND gate and NOR gate.

3.3 FORWARD SIMULATION FOR RECONVERGENT FAN_OUT POINT

3.3.1 FAN_OUT FREE REGION (FFR)

Setting up a combinational logic network for fault simulation by inference from surrogate fault simulation requires the nondisjoint partitioning of the network into seperate subnetwork, one for each primary output as illustrated in Figure 3.1. Gates are duplicated as necessary to form an independent realization of each primary output. Once partitioning is accomplished , any remaining fan_out within a subnetwork must necessarily be reconvergent fan_out. Each reconvergent fan_out node is called a fan_out stem (FOS). Behind each FOS is a fan_out free region (FFR) illustrated in Figure 3.2.

For a given logic network N and a given input test vector T, if the value of the network output , f, is a function of the value of node Y , then Y is network sensitive in N.

Network sensitivity relates directly to the notion of a Boolean difference. That is , df/dy evaluated for test vector T is 1 if and only if Y is network sensitive in N. The following definition for a fan_out free region is consistent.

Definition :

A node x in a FFR with FOS y is FFR sensitive if and only if

$$dy/dx = 1$$

Figure 3.3 depicts one particular FFR behind FOS g of a subnetwork with output f. Because all paths from the arbitrary node h in the FFR pass through the FOS g. the following theorem is an application of the chain rule of Boolean difference [6].

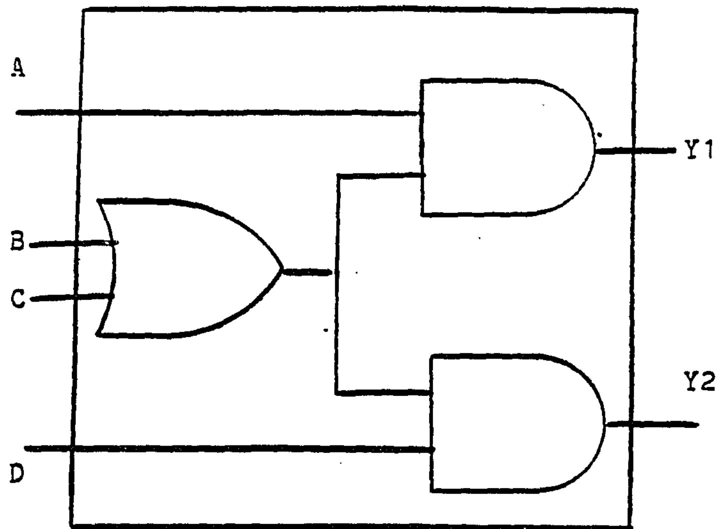
Theorem 3.1

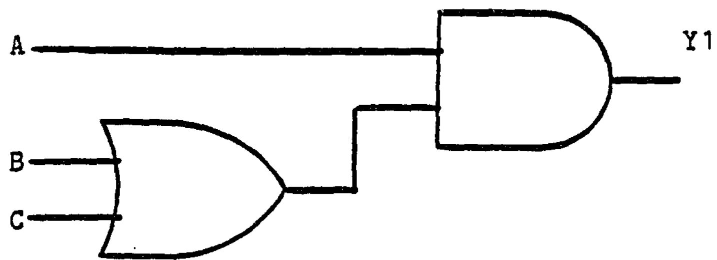
If a node g is subnetwork sensitive in a subnetwork with output f and node h is FFR sensitive in an FFR with FOS g, then h is a subnetwork sensitive.

Proof

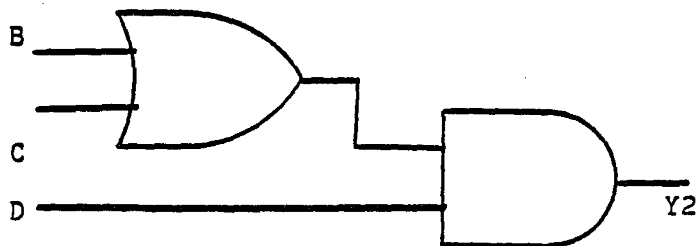
Let g be subnetwork sensitive and let h be FFR sensitive for FOS g. By definition dg/dh = 1 and df/dg = 1. Therefore, by the chain rule of Boolean differences [1].

$$df/dh = (df/dg) * (dg/dh).$$





SUBNET 1



SUBNET 2

Figure 3.1 Nondisjoint Partition of Network

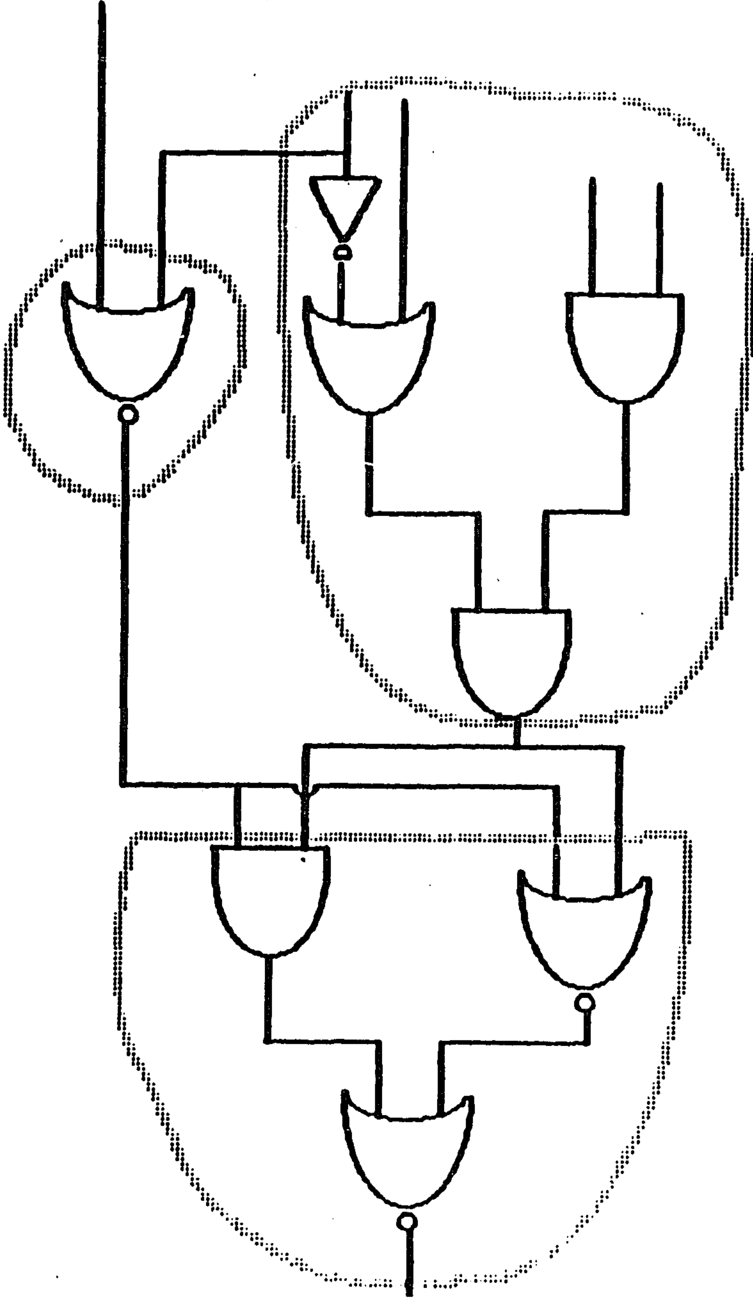


Figure 3.2 The FFR's of Subnetwork

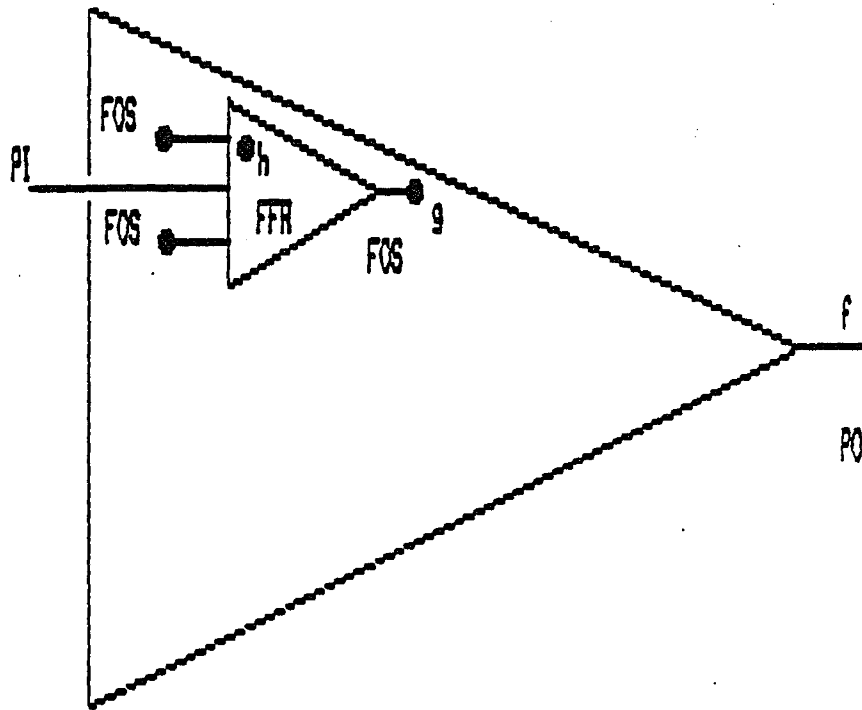


Figure 3.3 Sensitivity of Node in an FFR

3.3.2 F.O.S. Sensitivity

The subnetwork sensitivity of a particular FOS y may be determined by simply imposing the variable y as the value of that node and computing values of successive gates using y and the test vector input values. The output of any gate may now assume any one of the four values 0, 1, y, ^y. These four values may be represented by two bits of a binary vector as given by Table 3.1. It may be seen that the first bit of the vector will always be the good network value. The remaining bit, which indicates whether the node is a function of the FOS node will be called a FOS bit.

A vector representing the values of all FOS's at a network node will consist of one bit for the good and one bit for each FOS. The number of fan_out stems will rarely exceed the word length of the host computer, so each binary vector may be assigned to a computer work. The simulation would resemble but be much simpler and faster than a parallel fault simulation. It would not be necessary to impose faults at each gate input as in the parallel simulation. Except for adjusting the value of one bit at each FOS, the simulation would require no more time than a simulation of the good network.

As simulation begins, all bits in vectors representing gate inputs are assigned good network value. Each gate is simulated in turn by vector execution of the gate operation on the input vectors. After the vector at FOS I is computed, the value of the good network at that node is checked, and the opposite value is assigned to bit I of the vector just computed for the node. Execution of this process for a network with two fan_out stems is shown in figure 3.4.

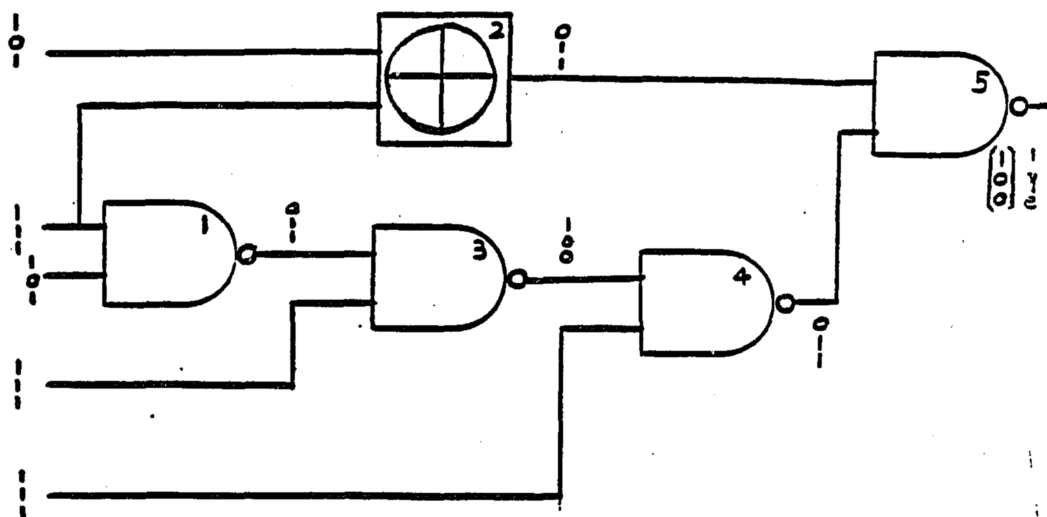


Figure 3.4 Parallel Computation of FOS Sensitivity

Table 3.1 Coding of Node Values

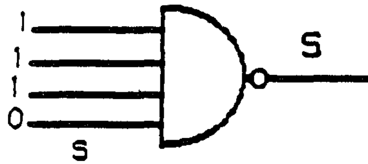
when good network value at FOS = 0			when good network value at FOS = 1		
good bit	FOS bit	node value	good bit	FOS bit	node value
0	0	0	0	0	0
0	1	y	0	1	ˆy
1	0	ˆy	1	0	y
1	1	1	1	1	1

3.4 BACKWARD TO SEARCH THE SENSITIVE PATHS

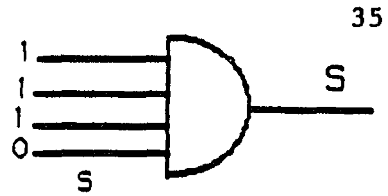
A list of detected faults corresponding to a sensitive FOS will be determined for every fan_out free region in a subnetwork. The subnetwork fault list will then be the union of these fault lists for sensitive fan-out stems. A necessary condition for a line to be sensitive is that the node at its destination must be sensitive. The sensitivity of the node at its source depends not only on the destination node type, but also on its input pattern. To illustrate this, a line which is sensitive will be tagged with sensitivity value S. The absence of S implies that a line is not sensitive. Figure 3.5 lists gate input sensitivity for distinct combinations of fault_free input values. From Figure 3.5a, 3.5c and 3.5d, it could be concluded that an input to a NAND gate is sensitive only if other fault_free inputs have value 1. Although only one AND gate is shown, sensitivity values for AND gate are the same as for NAND gates. The rest of type gates may be verified the sensitivity by the similar method.

Figure 3.6 illustrates an example of backward to search over all sensitive paths. The values of primary inputs ,secondary inputs and the good values of gates are shown; the wire numbers are shown with parentheses. From Figure 3.6, the good value of gate 6 is logic 1, wire

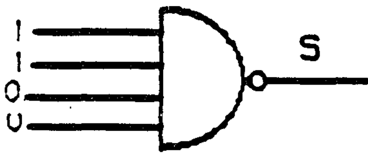
1 and wire 2 are also logic 1, since gate 6 is an AND gate, both wires 1 and 2 are sensitive to the output of gate 6. By the same token, the wires 3, 5 and 6 are in fact sensitive paths.



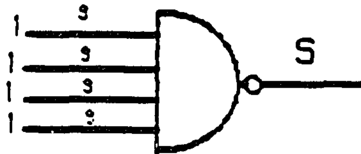
(a)



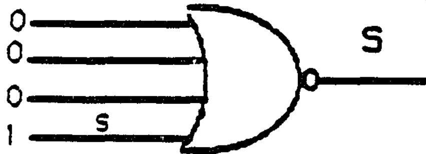
(b)



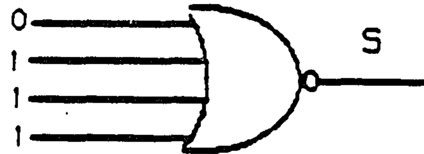
(c)



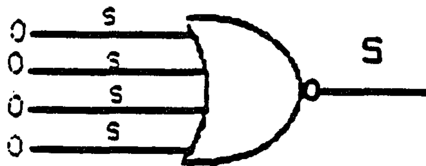
(d)



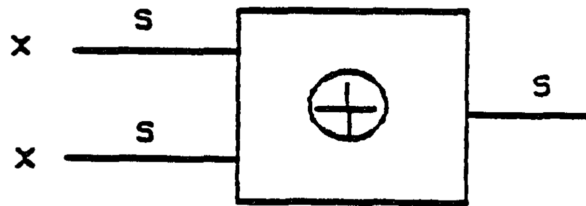
(e)



(f)



(g)



(h)

Figure 3.5 Backward Propagation of Node Sensitivity

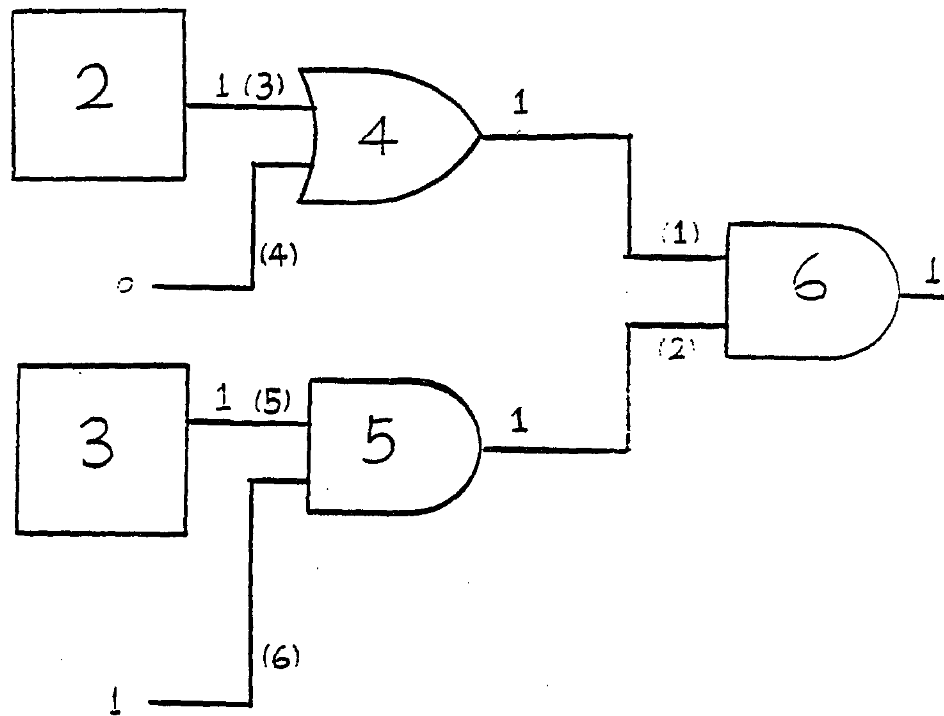


Figure 3.6 Backward to Search the Sensitive Paths

3.5 IDENTIFY THE COMPOUND FAULTS AND MULTIPLY STORED FAULTS

After each clock period of simulation of a sequential circuit, associated with each memory element will be the good network value and a list of all faults stored in that memory element.

Definition 1 :

A fault is stored in a memory element after a clock period of simulation, if the value of that memory element for the corresponding single fault network differs from that of good network.

Two tasks have to be accomplished when a new test pattern is going to be applied for the next clock period of execution.

Task 1 :

The stored faults are propagated into the combination logic network and depending on the circuit possibly to new secondary inputs or to primary outputs.

Task 2 :

Simulate the combination logic subnetwork to decide those faults driven to primary outputs or secondary outputs

from their points of origin in the subnetwork by using the new test pattern.

A convenient state of affairs would have the effect of a single fault stored in more than one memory element be independent. It would then only be necessary to determine the subnetwork sensitivity of each secondary input. Faults are propagated to the output of a subnetwork would be the union of fault lists stored at all memory elements sensitive within that subnetwork. The final fault lists for the clock period are then the union of these fault lists .This process seems to result in an accurate fault simulation , but unfortunately a single fault stored in more than one memory element will interact [1].

Definition 2 :

A compound fault is a fault which is stored in a memory element whose output is connected to the subnetwork containing the point of origin of that fault.

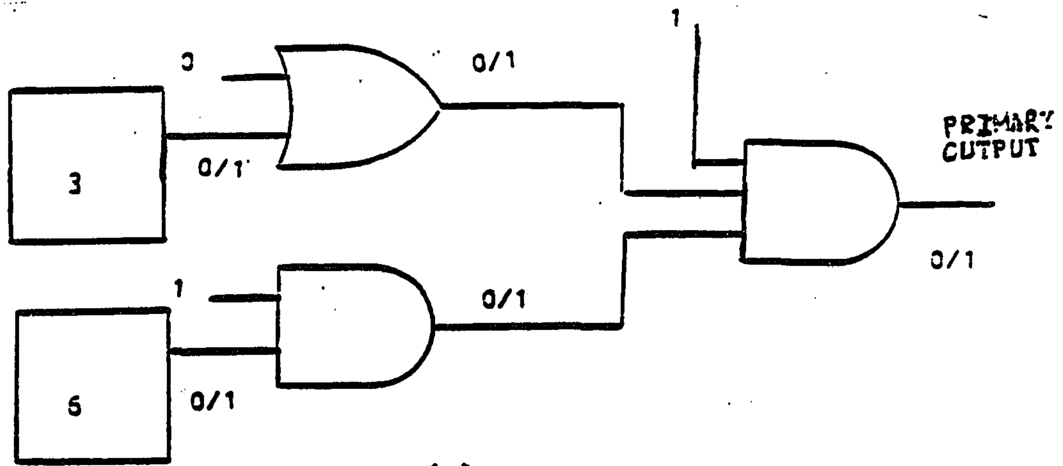
Definition 3 :

A multiply stored fault is a fault which is stored in two or more memory elements whose outputs are connected to the same subnetwork.

Equally well both a multiply fault and a compound fault resemble coexisting multiple faults in the combination logic network. Using the notation g/f, where g is the good network value and f is the faulty network value. Figure 3.7a illustrate a case where a fault multiply stored in two flip-flops will be detected , where a fault stored in only one of the two memory elements would not be detected .This is due to the fact that the AND gate is equivalent to a convergent gate and the paths converging at its input have different signal values. Figure 3.7b illustrates a case in which a fault effect stored only in memory element 3 would be detected, while the same fault stored in both memory elements 3 and 6 (values indicated in parentheses) would not be detected.

The situation is very similar for a compound fault. Figure 3.8a illustrate a case where the storage of the effect of gate 2 input SA1 (stuck-at-1) in memory element 6 prevents the detection of the fault at its point of origin. Also the equivalent convergent gate (gate 2) has different signal values for its converging paths. The values for no stored faults (in parentheses) show the original fault to be detected. Figure 3.8b shows a case where a compound fault would not be detected in the absence of the stored fault.

Since the compound faults and multiply stored faults are uncertain to be detected, there is considerable motivation to identify them as a special case for faults simulation.



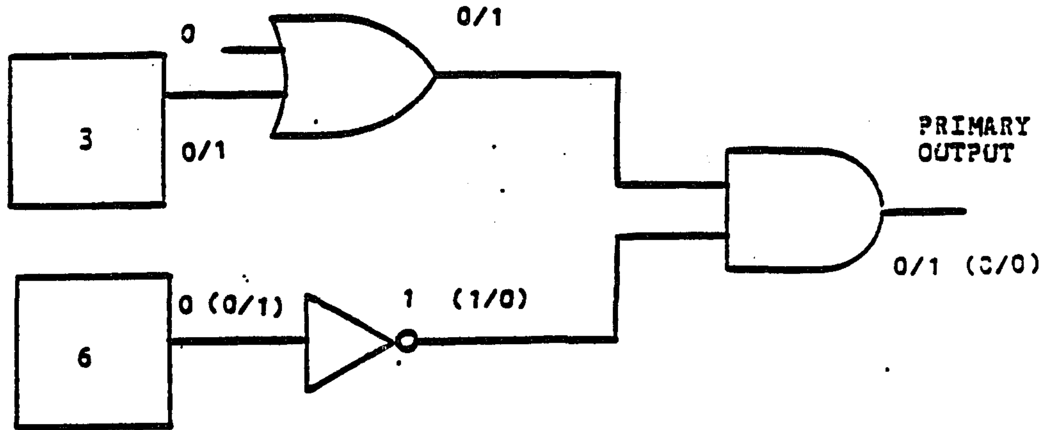
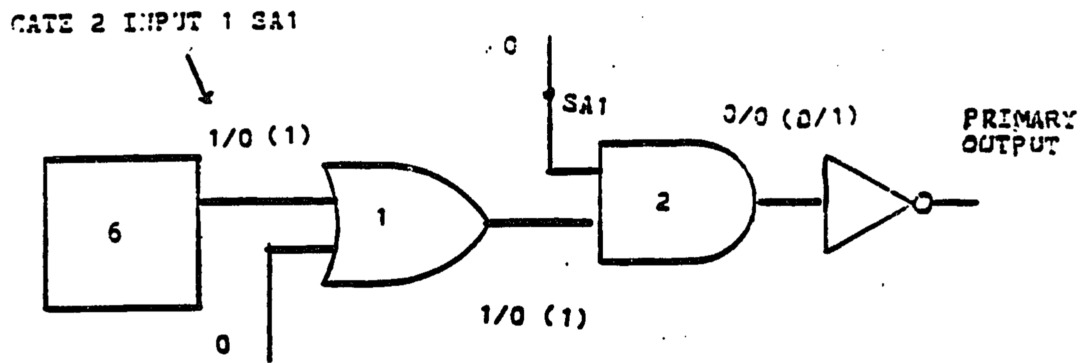
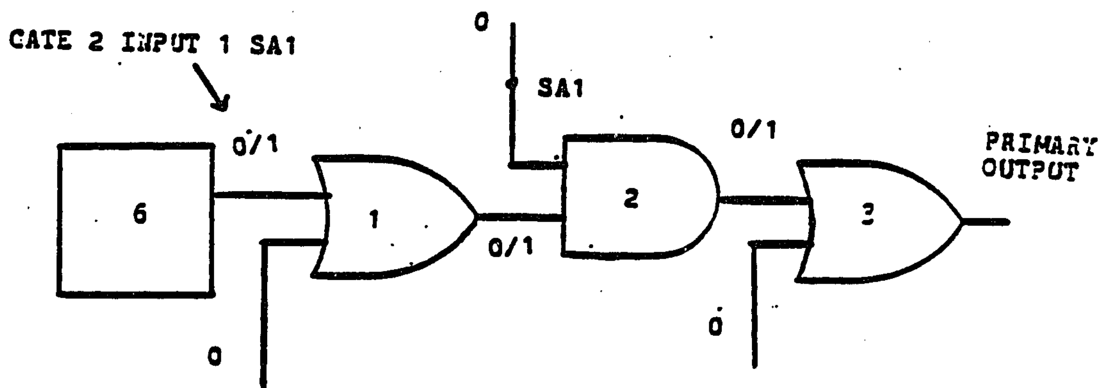


Figure 3.7 Reconvergent Multiply Stored Fault



(a)



(b)

Figure 3.8 Compound Fault

For each subnetwork, there could be more than one secondary input , which is the memory element with sensitive faults. The way to identify compound faults is to check if the stored faults in a particular memory element are also in the faults list of that subnetwork. Therefore, we may iteratively use the logic AND to compare the stored faults in each secondary input with the faults list of that subnetwork. Consequently, we will get compound faults for each secondary input in that particular subnetwork.

If there are more than one secondary input to a subnetwork,then the multiply stored faults must be identified for each secondary input. The identification of the multiply stored faults for one secondary input is to use logic AND to compare the stored faults with the stored faults in the rest of memory elements. This procedure will be executed for every individual secondary input to identify the multiply stored faults.

See Figure 3.9, supposedly there are two secondary inputs A and B in a subnetwork. To obtain the multiply stored faults for A, it need to operate logic AND the stored faults in memory element A with the stored faults in memory element B.

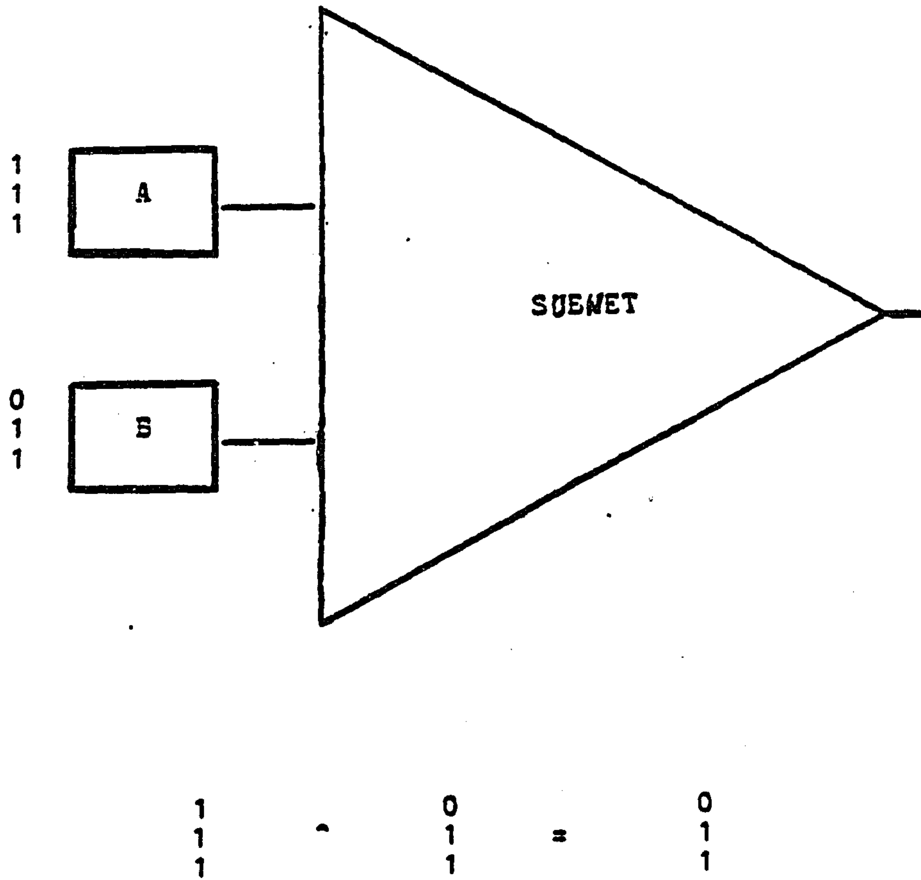


Figure 3.9 Computation of Multiply Stored fault

CHAPTER 4

FAULT COLLAPSING

Minimization techniques for the selection of efficient sets of diagnostic tests for combinational circuits normally begin with the consideration of a fault table in which each test is represented by a row and each fault by a column. The entry t(i,j) in the ith row and jth column is 1 if and only if test i detects fault j. Selecting a minimal set of tests having the desired fault diagnosis properties is a covering problem analogous to finding a minimum set of prime implicants for a combinational switching function. As in the prime implicant covering problem, the fault table can be reduced by means of row and column dominance conditions before test selection begins [7]. The type and extent of the reduction depends on the desired resolution of the test set, that is , fault detection ,or fault location to within some unit.

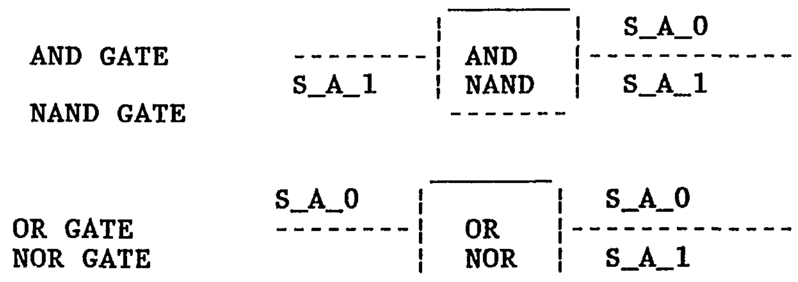
Column dominance as applied to a complete fault table is related to the relative detectability of faults. Column j dominates column k if for every 1 in columns k there is also a 1 in column j. If the two columns of the table are identical then each dominates the other and the faults corresponding to these two columns are

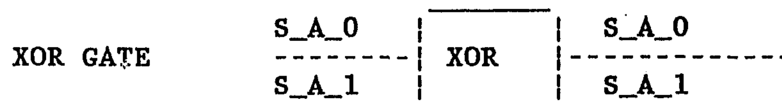
indistinguishable. In this case, any test which detects one fault will also detect the other.

Fault collapsing is the process of combining faults by means of implication relationships derived from a study of the network concerned. These relationships form a partial ordering relative to the detection of faults. Classes of indistinguishable faults are found before deriving diagnostic tests rather than after, and only one fault from each class need then be considered.

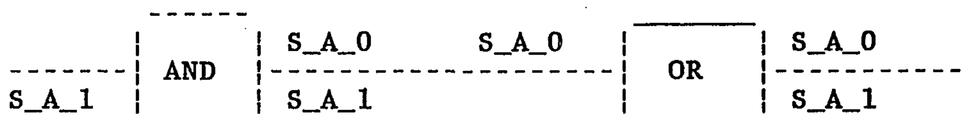
The fault identification in SFSSE differs from SCIRTSS. The difference of faults location between SFSSE and SCIRTSS is illustrated as follow.

For all fan-out wires, the faults location of the new simulator has been fixed by the routine FLTCLP.MOD. This part of result will be the same as in SCIRTSS. For no fan-out wires, the faults location of SCIRSS is as follows.



XOR GATE 

IF AND GATE CONNECTS WITH OR GATE :



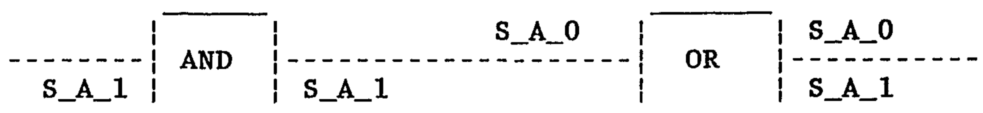
FAULTS COLLAPSING :

MODE 0 : NO COLLAPSING

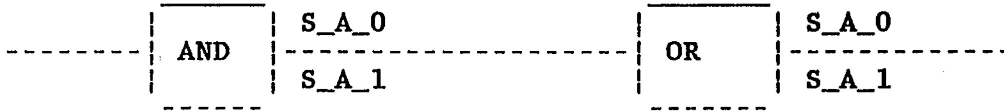
MODE 1 : COLLAPSE AND GATE OUTPUT S_A_0

MODE 2 : COLLAPSE AND GATE OUTPUT S_A_1

Comparing the faults location of SCIRTSS with that of the new simulator as follow :



FAULTS AT OUTPUT OF GATES OF THE NEW SIMULATOR



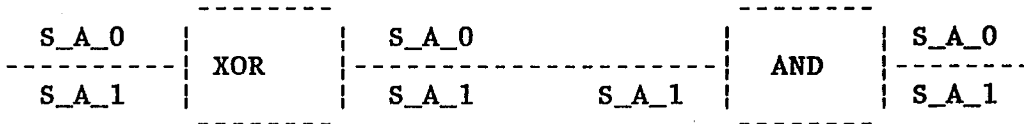
IF XOR GATE CONNECTS WITH AND GATE :

AFTER MODE 1

SCITRSS

2 FAULTS
AT INPUT

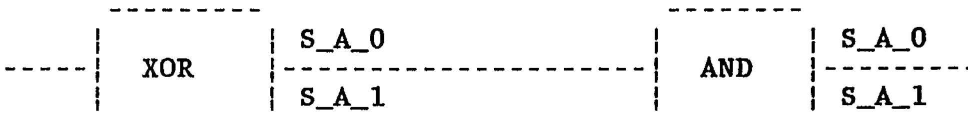
3 FAULTS



NEW SIMULATOR

NO FAULT
AT INPUT

2 FAULTS



In SCIRTSS , the faults location , SA0 and SA1, are assigned at output wire of all gates and SA0 at input wires of NOR and OR gate ; SA1 at input wires of NAND and AND gate, SA0 and SA1 at input wires of XOR gate. In mode 0, no faults are suppressed.

In the new simulator, the faults are assigned at the output wires of all gates. Consequently there are fewer faults assigned than in SCIRTSS and the fault location is different from that of SCIRTSS. The routine FLTCLP.MOD makes the fault numbers in the simulation result the same, but the fault identification remains different.

The routine FLTCLP.MOD which was one of the independent contributions of this thesis is divided into two parts. The first part accepts the result of the new simulation and sorts the numbered detected faults as the order of the results of SCIRTSS. From the output of this part of routine, the user may check the results of the new simulator with that of SCIRTSS easily.

The second part of routine FLTCLP.MOD collapses the equivalent faults for the simulated circuit. Since the faults location on the wire in the new simulator are assigned differently than in SCIRTSS, the reduction of faults number is also different from that of SCIRTSS.

CHAPTER 5

INPUT DATA

In this chapter, six circuits carefully selected to be used as input data for the new simulator will be discussed. The six circuits, RNDPRU.SCR, ADD8X4.SCR, ADD8X8.SCR, ADD8X16.SCR, TEST88.SCR, MSFCF4.SCR are separately studied as follows. The research results will be shown in chapter 6.

5.1 AHPL FOR RNDPRU.SCR

Circuit RNDPRU.SCR is a random generator process module. The purpose of selecting this circuit is to verify that SFSSE, the new simulator, is working properly for a highly sequential circuit.

Number of Circuit Elements = 156

Number of Flip_Flops = 26

Number of Control States = 5

Number of External Inputs = 3

```
MODULE:RNDPRU.SCR
EXINPUTS:Z;START;TEST;CLOCK.
EXOUTPUTS:TRIG;AVEOUT[4];FRACT[4];TOUT.
MEMORY:TOT[8];NUM[8];CNT[4];LAST.
CLUNITS:INC1[4]<:INCTST{4}.
CLUNITS:INC2[8]<:INCTST{8}.
PULSES:CLOCK.
BODY SEQUENCE:CLOCK.

1       =>(^START)/(1).

2       TOT <= 8$0; NUM <= 8$0.

3       CNT <= 4$0; LAST<=Z.

4       CNT<=INC1(CNT); TRIG=CNT[0];

        =>(CNT[0]+TEST,^CNT[0]&^TEST)/(5,4).

5       TOT*(^Z&LAST + Z&^LAST) <=
        (INC2(TOT)!(TOT[1:7],START))*(^TEST,TEST);
        LAST<=Z;
        NUM<= (INC2(NUM)!(NUM[1:7],START))*(^TEST,TEST);
        TRIG=\1\; CNT <= 4$0;
        =>(&/NUM,^(&/NUM))/(1,4).
ENDSEQUENCE
CONTROLRESET(1);
AVEOUT = TOT[0:3];
FRACT = TOT[4:7];
TOUT = NUM[0].
END.

CLU:INCTST(OP) {I}.
INPUTS:OP[I].
OUTPUTS:TERMOUT[I].
CTERMS:TA[I].
BODY
     FOR J = 0 TO I-1 CONSTRUCT
                TERMOUT[J] = OP[J]@ TA[J]
     ROF;
     FOR J = I-1 TO 0 CONSTRUCT
                IF J=I-1 THEN
                        TA[J]=\1\
                ELSE
                        TA[J]=OP[J+1]&TA[J+1]
                FI
     ROF.
END.
```

5.2 AHPL FOR ADD8X4.SCR

Circuit ADD8X4.SCR is a cascade of adders constructed from four 8 bit carry lookahead adders with carry in. The reasons for selecting it are this circuit is considerable logic and has numerous fan-out stems.

Number of Circuit Elements = 622
Number of Flip_Flops = 39
Number of Control States = 3
Number of External Inputs = 42

```
MODULE:ADD8X4.SCR
EXINPUTS:X[8];A[8];B[8];C[8];D[8];RUN;LOOK;CLOCK.
MEMORY:RR<4>[9].
EXOUTPUTS:Z[9].
CLUNITS:ADD[9]<:ADDLA8.
BODY SEQUENCE: CLOCK.

     1 =>(^RUN)/(1).

     2 RR<0> <= ADD(X;A;\0\);
       RR<1> <= ADD(RR<0>[1:8];B;\0\);
       RR<2> <= ADD(RR<1>[1:8];C;\0\);
       RR<3> <= ADD(RR<2>[1:8];D;\0\);
       =>(LOOK,^LOOK)/(3,2).

     3 RR<1> <= RR<0>;
       RR<2> <= RR<1>;
       RR<3> <= RR<2>;
       =>(^LOOK,LOOK)/(1,3).

ENDSEQUENCE
CONTROLRESET(1);
Z = RR<3>.
END.
```

Circuit ADD8X4.SCR--Continued

```
CLU:ADDLA8(X;Y;CIN).
INPUTS:X[8];Y[8];CIN.
OUTPUTS:TERMOUT[9].
CLUNITS:LA[6]<:CLA.
CLUNITS:PG[2]<:PROPGEN.
CLUNITS:SUMBIT<:SUM.
CTERMS:G[8];P[8];C[9];GC.
BODY
FOR I=0 TO 7 CONSTRUCT
G[I],P[I]=PG(X[I];Y[I])
ROF;
C[5:8]=LA[2:5](G[4:7];P[4:7];CIN);
GC=LA[0](G[4:7];P[4:7];CIN)+LA[1](G[4:7];P[4:7];CIN)&CIN;
C[1:4]=LA[2:5](G[0:3];P[0:3];GC);
C[0]=LA[0](G[0:3];P[0:3];GC) + LA[1](G[0:3];P[0:3];GC)&GC;
TERMOUT[0] = C[0];
FOR I=0 TO 7 CONSTRUCT
TERMOUT[I+1]=SUMBIT(P[I];C[I+1])
ROF.
END.
CLU: CLA(G;P;CIN).
INPUTS: G[4]; P[4]; CIN.
OUTPUTS: TERMOUT[6].
BODY
TERMOUT[5] = CIN;
TERMOUT[4] = G[3] + P[3]&CIN;
TERMOUT[3] = G[2] + P[2]&G[3] + P[3]&P[2]&CIN;
TERMOUT[2] = G[1] + P[1]&G[2] + P[2]&P[1]&G[3]
             + P[3]&P[2]&P[1]&CIN;
TERMOUT[0] = G[0] + P[0]&G[1] + P[1]&P[0]&G[2]
             + P[2]&P[1]&P[0]&G[3];
TERMOUT[1] = P[3]&P[2]&P[1]&P[0].
END.
CLU: PROPGEN(X;Y).
INPUTS: X[1];Y[1].
OUTPUTS:TERMOUT[2].
BODY
TERMOUT[0]=X&Y;
TERMOUT[1]=X&^Y + ^X&Y.
END.
CLU:SUM(PROP;CIN).
INPUTS:PROP;CIN.
OUTPUTS:TERMOUT.
BODY
TERMOUT=PROP&^CIN + ^PROP&CIN.
END.
```

5.3 AHPL FOR ADD8X8.SCR

Circuit ADD8X8.SCR is a cascade of adders constructed from eight 8 bit CLADDs with carry in. The basic circuit structure is very similar to the circuit ADD8X4.SCR, but with twice as many adders.

Number of Circuit Elements = 1246
Number of Flip_Flops = 75
Number of Control States = 3
Number of External Inputs = 74

```
MODULE:ADD8X8.SCR
EXINPUTS:X[8];A[8];B[8];C[8];D[8];E[8];F[8];G[8];H[8];RUN
        ;LOOK;CLOCK.
MEMORY:RR<8>[9].
EXOUTPUTS:Z[9].
CLUNITS:ADD[9]<:ADDLA8.
BODY SEQUENCE: CLOCK.
     1 =>(^RUN)/(1).

     2 RR<0> <= ADD(X;A;\0\);
       RR<1> <= ADD(RR<0>[1:8];B;\0\);
       RR<2> <= ADD(RR<1>[1:8];C;\0\);
       RR<3> <= ADD(RR<2>[1:8];D;\0\);
       RR<4> <= ADD(RR<3>[1:8];E;\0\);
       RR<5> <= ADD(RR<4>[1:8];F;\0\);
       RR<6> <= ADD(RR<5>[1:8];G;\0\);
       RR<7> <= ADD(RR<6>[1:8];H;\0\);
       =>(LOOK,^LOOK)/(3,2).

     3 RR<1> <= RR<0>;
       RR<2> <= RR<1>;
       RR<3> <= RR<2>;
       RR<4> <= RR<3>;
       RR<5> <= RR<4>;
       RR<6> <= RR<5>;
       RR<7> <= RR<6>;
       =>(^LOOK,LOOK)/(1,3).
```

```
ENDSEQUENCE
CONTROLRESET(1);
Z = RR<7>.
END.

CLU:ADDLA8(X;Y;CIN).
INPUTS:X[8];Y[8];CIN.
OUTPUTS:TERMOUT[9].
CLUNITS:LA[6]<:CLA.
CLUNITS:PG[2]<:PROPGEN.
CLUNITS:SUMBIT<:SUM.
CTERMS:G[8];P[8];C[9];GC.
BODY
FOR I=0 TO 7 CONSTRUCT
G[I],P[I]=PG(X[I];Y[I])
ROF;
C[5:8]=LA[2:5](G[4:7];P[4:7];CIN);
GC=LA[0](G[4:7];P[4:7];CIN)+LA[1](G[4:7];P[4:7];CIN)&CIN;
C[1:4]=LA[2:5](G[0:3];P[0:3];GC);
C[0]=LA[0](G[0:3];P[0:3];GC) + LA[1](G[0:3];P[0:3];GC)&GC;
TERMOUT[0] = C[0];
FOR I=0 TO 7 CONSTRUCT
TERMOUT[I+1]=SUMBIT(P[I];C[I+1])
ROF.
END.

CLU: CLA(G;P;CIN).
INPUTS: G[4]; P[4]; CIN.
OUTPUTS: TERMOUT[6].
BODY
TERMOUT[5] = CIN;
TERMOUT[4] = G[3] + P[3]&CIN;
TERMOUT[3] = G[2] + P[2]&G[3] + P[3]&P[2]&CIN;
TERMOUT[2] = G[1] + P[1]&G[2] + P[2]&P[1]&G[3]
             + P[3]&P[2]&P[1]&CIN;
TERMOUT[0] = G[0] + P[0]&G[1] + P[1]&P[0]&G[2]
             + P[2]&P[1]&P[0]&G[3];
TERMOUT[1] = P[3]&P[2]&P[1]&P[0].
END.
```

```
CLU: PROPGEN(X;Y).
INPUTS: X[1];Y[1].
OUTPUTS:TERMOUT[2].
BODY
TERMOUT[0]=X&Y;
TERMOUT[1]=X&^Y + ^X&Y.
END.

CLU:SUM(PROP;CIN).
INPUTS:PROP;CIN.
OUTPUTS:TERMOUT.
BODY
TERMOUT=PROP&^CIN + ^PROP&CIN.
END.
```

5.4 AHPL FOR ADD8X16.SCR

Circuit ADD8X16.SCR is a cascade of adders constructed from sixteen 8 bit CLADDs with carry in. The basic circuit structure is the same with the circuit ADD8X8.SCR, but twice as large.

Number of Circuit Elements = 2494

Number of Flip_flops = 147

Number of Control States = 3

Number of External Inputs = 138

```
MODULE:ADD8X16.SCR
EXINPUTS:X[8];A[8];B[8];C[8];D[8];E[8];F[8];
         G[8];H[8];RUN;LOOK;CLOCK;
         I[8];J[8];K[8];L[8];M[8];N[8];O[8];P[8].
MEMORY:RR<16>[9].
EXOUTPUTS:Z[9].
CLUNITS:ADD[9]<:ADDLA8.
BODY SEQUENCE: CLOCK.
1       =>(^RUN)/(1).

2       RR<0> <= ADD(X;A;\0\);
        RR<1> <= ADD(RR<0>[1:8];B;\0\);
        RR<2> <= ADD(RR<1>[1:8];C;\0\);
        RR<3> <= ADD(RR<2>[1:8];D;\0\);
        RR<4> <= ADD(RR<3>[1:8];E;\0\);
        RR<5> <= ADD(RR<4>[1:8];F;\0\);
        RR<6> <= ADD(RR<5>[1:8];G;\0\);
        RR<7> <= ADD(RR<6>[1:8];H;\0\);
        RR<8> <= ADD(RR<7>[1:8];I;\0\);
        RR<9> <= ADD(RR<8>[1:8];J;\0\);
        RR<10> <= ADD(RR<9>[1:8];K;\0\);
        RR<11> <= ADD(RR<10>[1:8];L;\0\);
        RR<12> <= ADD(RR<11>[1:8];M;\0\);
        RR<13> <= ADD(RR<12>[1:8];N;\0\);
        RR<14> <= ADD(RR<13>[1:8];O;\0\);
        RR<15> <= ADD(RR<14>[1:8];P;\0\);
        =>(LOOK,^LOOK)/(3,2).

3       RR<1> <= RR<0>;
        RR<2> <= RR<1>;
        RR<3> <= RR<2>;
        RR<4> <= RR<3>;
        RR<5> <= RR<4>;
        RR<6> <= RR<5>;
        RR<7> <= RR<6>;
        RR<8> <= RR<7>;
        RR<9> <= RR<8>;
       RR<10> <= RR<9>;
       RR<11> <= RR<10>;
       RR<12> <= RR<11>;
       RR<13> <= RR<12>;
       RR<14> <= RR<13>;
       RR<15> <= RR<14>;
       =>(^LOOK,LOOK)/(1,3).
ENDSEQUENCE
CONTROLRESET(1);
Z = RR<15>.
END.
```

Circuit ADD8X16.SCR--Continued

```
CLU:ADDLA8(X;Y;CIN).
INPUTS:X[8];Y[8];CIN.
OUTPUTS:TERMOUT[9].
CLUNITS:LA[6]<:CLA.
CLUNITS:PG[2]<:PROPGEN.
CLUNITS:SUMBIT<:SUM.
CTERMS:G[8];P[8];C[9];GC.
BODY
FOR I=0 TO 7 CONSTRUCT
G[I],P[I]=PG(X[I];Y[I])
ROF;
C[5:8]=LA[2:5](G[4:7];P[4:7];CIN);
GC=LA[0](G[4:7];P[4:7];CIN) + LA[1](G[4:7];P[4:7];CIN)&CIN;
C[1:4]=LA[2:5](G[0:3];P[0:3];GC);
C[0]=LA[0](G[0:3];P[0:3];GC) + LA[1](G[0:3];P[0:3];GC)&GC;
TERMOUT[0] = C[0];
FOR I=0 TO 7 CONSTRUCT
TERMOUT[I+1]=SUMBIT(P[I];C[I+1])
ROF.
END.
CLU: CLA(G;P;CIN).
INPUTS: G[4]; P[4]; CIN.
OUTPUTS: TERMOUT[6].
BODY
TERMOUT[5] = CIN;
TERMOUT[4] = G[3] + P[3]&CIN;
TERMOUT[3] = G[2] + P[2]&G[3] + P[3]&P[2]&CIN;
TERMOUT[2] = G[1] + P[1]&G[2] + P[2]&P[1]&G[3]
             + P[3]&P[2]&P[1]&CIN;
TERMOUT[0] = G[0] + P[0]&G[1] + P[1]&P[0]&G[2]
             + P[2]&P[1]&P[0]&G[3];
TERMOUT[1] = P[3]&P[2]&P[1]&P[0].
END.
CLU: PROPGEN(X;Y).
INPUTS: X[1];Y[1].
OUTPUTS:TERMOUT[2].
BODY
TERMOUT[0]=X&Y;
TERMOUT[1]=X&^Y + ^X&Y.
END.
CLU:SUM(PROP;CIN).
INPUTS:PROP;CIN.
OUTPUTS:TERMOUT.
BODY
TERMOUT=PROP&^CIN + ^PROP&CIN.
END.
```

5.5 AHPL FOR TEST88.SCR

Circuit TEST88.SCR is constructed from eight 8 bit combinational logic units denoted for convenience by ADD. This logic units are not really adders, (see Figure 5.1) the left most module is the first stage, i.e. , a separate module with 18 external inputs for the first 8 bit addition operation. The resultant sum will be latched in an 8 bit register. The output of the register is connected to the inputs of the second level adder, with the corresponding data coming from 8 external input lines. Therefore, the second module receives one half of its inputs from the first module and the other half from external input lines. In the same manner, the rest six modules receives its inputs from previous module and external lines.

Number of Circuit Elements = 704

Number of Flip_Flops = 67

Number of Control States = 3

Number of External Inputs = 74

```
MODULE:TEST8X8.SCR
EXINPUTS:X[8];A[8];B[8];C[8];D[8];E[8];F[8];G[8];H[8]
         ;RUN;LOOK;CLOCK.
MEMORY:RR<8>[8].
EXOUTPUTS:Z[8].
CLUNITS:ADD[8]<:LOGTEST.
BODY SEQUENCE: CLOCK.
```

Circuit TEST88.SCR--Continued

```
1       =>(^RUN)/(1).

2       RR<0> <= ADD[0:7](X;A);
        RR<1> <= ADD[0:7](RR<0>;B);
        RR<2> <= ADD[0:7](RR<1>;C);
        RR<3> <= ADD[0:7](RR<2>;D);
        RR<4> <= ADD[0:7](RR<3>;E);
        RR<5> <= ADD[0:7](RR<4>;F);
        RR<6> <= ADD[0:7](RR<5>;G);
        RR<7> <= ADD[0:7](RR<6>;H);

        =>(LOOK,^LOOK)/(3,2).

3       RR<1> <= RR<0>;
        RR<2> <= RR<1>;
        RR<3> <= RR<2>;
        RR<4> <= RR<3>;
        RR<5> <= RR<4>;
        RR<6> <= RR<5>;
        RR<7> <= RR<6>;
        =>(^LOOK,LOOK)/(1,3).

ENDSEQUENCE
CONTROLRESET(1);
Z = RR<7>.
END.

CLU:LOGTEST(X;Y).
INPUTS:X[8];Y[8].
OUTPUTS:TERMOUT[8].
BODY
TERMOUT[0]=(X[0]&^Y[0] + ^X[0]&Y[0]) & X[7] + Y[7];
FOR I=1 TO 7 CONSTRUCT
TERMOUT[I]=(X[I]&^Y[I] + ^X[I]&Y[I]) & X[I-1] + Y[I-1]
ROF.
END.
```

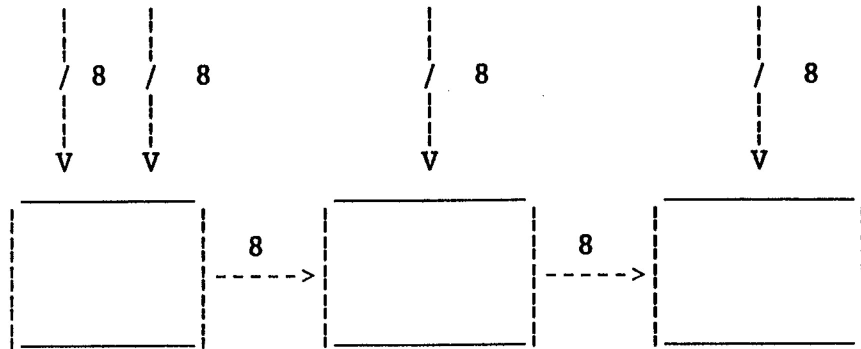


Figure 5.1 Cascade of Adders

5.6 AHPL FOR MSFCF4.SCR

Circuit MSFCF4.SCR is a cascade of adders constructed from four 8 bit CLADDs with carry in. The basic circuit structure is the same with the circuit ADD8X4.SCR, but the resultant sum will be latched in a nine 9 bit register depending on the DECORDER. This circuit is selected for testing of the handling of compound faults by the new simulator.

Number of Circuit Elements = 603

Number of Flip_Flops = 48

Number of Control States = 3

Number of External Inputs = 36

```
MODULE:MSFCF4.SCR
EXINPUTS:SEL[2];A[8];B[8];C[8];D[8];RUN;LOOK;CLOCK.
MEMORY:RR<4>[9];OUTREG[9].
BUSES:OUTBUS[9].
OUTPUTS:Z[9].
CLUNITS:ADD[9]<:ADDLA8.
CLUNITS:DCD[4]<:DECODE.
BODY SEQUENCE: CLOCK.
     1 =>(^RUN)/(1).

     2 RR<0> <= ADD(RR<0>[1:8];A;\0\);
       RR<1> <= ADD(RR<1>[1:8];B;\0\);
       RR<2> <= ADD(RR<2>[1:8];C;\0\);
       RR<3> <= ADD(RR<3>[1:8];D;\0\);
       =>(LOOK,^LOOK)/(3,2).

     3 OUTBUS = RR * DCD[0:3](SEL[0:1]);
       OUTREG <= OUTBUS;
       =>(1).
        ENDSEQUENCE
CONTROLRESET(1);
Z = OUTREG.
END.
```

Circuit MSFCF4.SCR--Continued

```
CLU:ADDLA8(X;Y;CIN).
INPUTS:X[8];Y[8];CIN.
OUTPUTS:TERMOUT[9].
CLUNITS:LA[6]<:CLA.
CLUNITS:PG[2]<:PROPGEN.
CLUNITS:SUMBIT<:SUM.
CTERMS:G[8];P[8];C[9];GC.
BODY
FOR I=0 TO 7 CONSTRUCT
G[I],P[I]=PG(X[I];Y[I])
ROF;
C[5:8]=LA[2:5](G[4:7];P[4:7];CIN);
GC=LA[0](G[4:7];P[4:7];CIN) + LA[1](G[4:7];P[4:7];CIN)&CIN;
C[1:4]=LA[2:5](G[0:3];P[0:3];GC);
C[0]=LA[0](G[0:3];P[0:3];GC) + LA[1](G[0:3];P[0:3];GC)&GC;
TERMOUT[0] = C[0];
FOR I=0 TO 7 CONSTRUCT
TERMOUT[I+1]=SUMBIT(P[I];C[I+1])
ROF.
END.

CLU: CLA(G;P;CIN).
INPUTS: G[4]; P[4]; CIN.
OUTPUTS: TERMOUT[6].
BODY
TERMOUT[5] = CIN;
TERMOUT[4] = G[3] + P[3]&CIN;
TERMOUT[3] = G[2] + P[2]&G[3] + P[3]&P[2]&CIN;
TERMOUT[2] = G[1] + P[1]&G[2] + P[2]&P[1]&G[3]
             + P[3]&P[2]&P[1]&CIN;
TERMOUT[0] = G[0] + P[0]&G[1] + P[1]&P[0]&G[2]
             + P[2]&P[1]&P[0]&G[3];
TERMOUT[1] = P[3]&P[2]&P[1]&P[0].
END.
CLU: PROPGEN(X;Y).
INPUTS: X[1];Y[1].
OUTPUTS:TERMOUT[2].
BODY
TERMOUT[0]=X&Y;
TERMOUT[1]=X&^Y + ^X&Y.
END.
CLU:SUM(PROP;CIN).
INPUTS:PROP;CIN.
OUTPUTS:TERMOUT.
BODY
TERMOUT=PROP&^CIN + ^PROP&CIN.
END.
```

Circuit MSFCF4.SCR--Continued

```
CLU:DECODE(OP).
INPUTS:OP[2].
OUTPUTS:TERMOUT[4].
BODY
 TERMOUT[0] = ^OP[0] & ^OP[1];
 TERMOUT[1] = ^OP[0] &  OP[1];
 TERMOUT[2] =  OP[0] & ^OP[1];
 TERMOUT[3] =  OP[0] &  OP[1].
END.
```

CHAPTER 6

SIMULATION RESULTS AND COMPARSION

For the purpose of comparing the running time of the new simulator with that of SCIRTSS, the running time of simulation for the new simulator is divided into three parts. Figure 6.1 illustrates the procedures of each part. The running time of Part A includes the time spends on forward simulation of good network and the parallel simulation of fan-out stems. The running time of Part B includes the time spent on the identification of excited faults and the formation of fault lists. The running time of Part C includes the time spent on the identification and simulation of multiply stored faults and compound faults. Figure 6.2 to Figure 6.7 are drew on the basis of these three parts.

For Clock Period

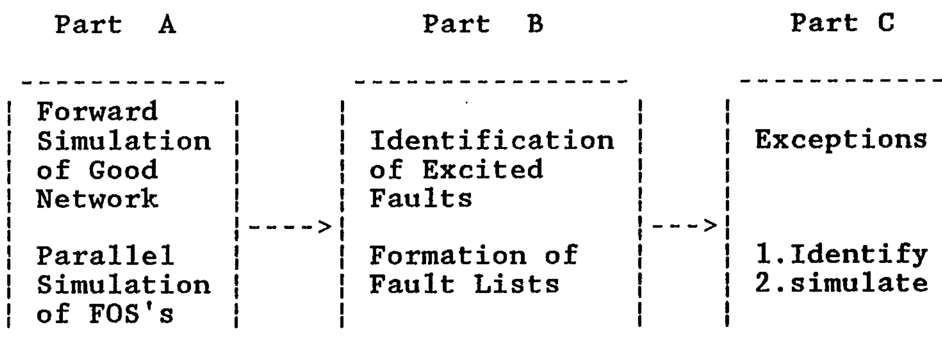


Figure 6.1 Time Division of SFSSE Simulation

Now let us compare the running time for the five circuits (see Figure 6.2, 6.3, 6.4, 6.5 and 6.6) The results in TABLES are taken directly from the outputs of the parallel simulator and the new simulator. Column 1 indicates the particular periods of fault simulation; Column 2 indicates the faults are detected in each specific period; Column 3 indicates time spent on forward simulation of good network and parallel simulation of FOS's; Column 4 indicates time spent on identification of excited faults and formation of fault lists; Column 5 indicates time spent on identification and simulation of multiply stored faults and compound faults; and Column 6 indicates the total spent time of each simulation period.

6.1 Circuit RNDPRU.SCR

Circuit RNDPRU.SCR is a small sequential circuit with 3 external inputs, 26 flip_flops and 156 circuit elements. The purpose of selecting this circuit is to test the efficiency of the new simulator over the parallel simulator on a small circuit and to check accuracy on highly sequential circuit. (see Table 6.1) The original number of fan_out stems is 18. After the partition of this circuit the number of fan-out stems increases to 48. Because the total number of faults is only 422, the

parallel simulator uses fewer computer words for representing faults than for larger circuits. The necessary operations through CPU for the small circuit in parallel simulation are proportionally decreased. The disadvantage of parallel simulator is not so obvious in that the speed of new simulator is 1/2 that of the parallel simulator over the first 20 periods. When more faults have been detected and less multiply stored faults and compound faults are stored and the Part C simulating time is declines. At the 300th period, the total time of new simulator is 1/5 that of parallel simulator.

Table 6.1 Time Comparison Between SCIRTSS and SFSSE for Circuit RNDPRU.SCR

* RNDPRU.SCR

* THE ORIGINAL NUMBER OF F.O.S. : 18

* NUMBER OF F.S.O. AFTER PARTITIONING : 48

* FAULT COLLAPSING

NUMBER OF CLOCK PERIOD	PERCENTAGE OF FAULTS FOUND	Part A TIME OF FORWARD SIMULATION	Part B TIME OF BACKWARD SIMULATION	Part C TIME OF EXCEPTION SIMULATION	TOTAL TIME OF SCIRTSS
1	22/422	*	*	*	1.23
2	22/422	0.04	0.06	0.07	0.52
10	112/422	0.03	0.06	0.15	0.56
20	151/422	0.03	0.05	0.14	0.55
50	293/422	0.04	0.04	0.06	0.54
100	332/422	0.05	0.05	0.04	0.55
200	375/422	0.04	0.05	0.03	0.53
300	395/422	0.04	0.04	0.03	0.54

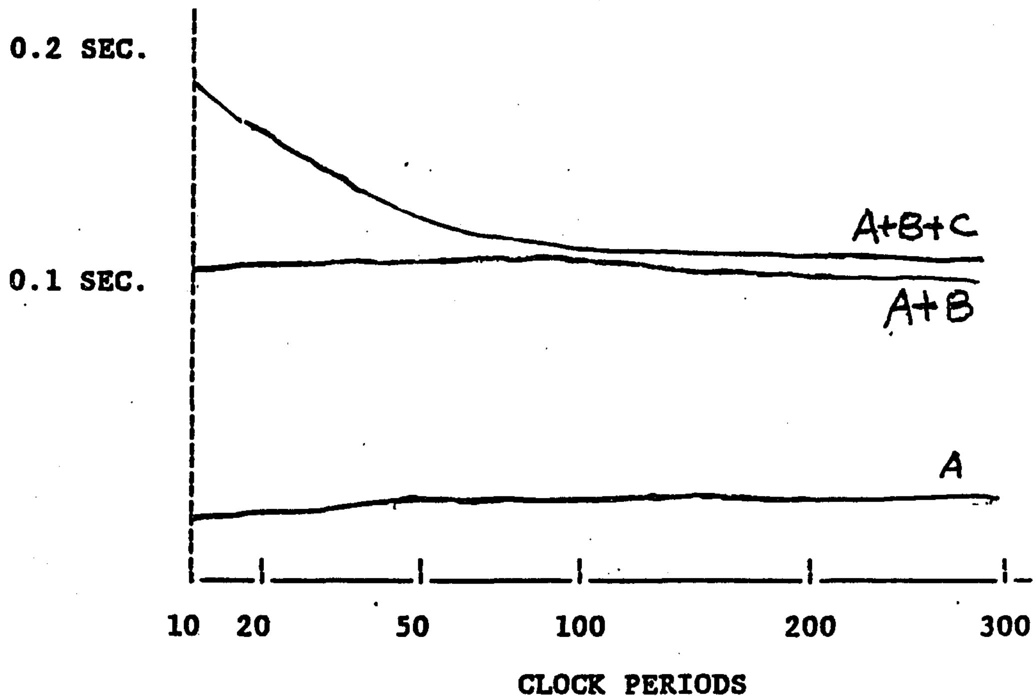


Figure 6.2 Total Time for clock period of SFSSE Simulation for Circuit RNDPRU.SCR

6.2 Circuit ADD8X4.SCR

Circuit ADD8X4.SCR is very similar to the circuit ADD8X8.SCR. The purpose of selecting this circuit is to compare with a smaller circuit but with the same type. The basic structure of the circuit is four 8 bit CLAADs. It has 42 external inputs, 39 flip_flops and 622 circuit elements. The original number of fan_out stems is 126, but after partition the number of fan-out stems (FOS) of the circuit is 513. Because this circuit is smaller than circuit ADD8X8.SCR, the words used by the computer for parallel simulator are less than that of circuit ADD8X8.SCR. Therefore, the disadvantage of the parallel simulator is less obvious than expected for ADD8X8.SCR. Within the first 10 periods, the average time of new simulatr is approximately 6 times faster than the parallel simulator. (see Table 6.2) When more faults have been detected, and less multiply stored faults and compound faults are stored, the 50th period reaches 10 times faster than the parallel simulator.

Table 6.2 Time Comparison Between SCIRTSS and SFSSE for Circuit ADD8X4.SCR

* ADD8X4.SCR

* THE ORIGINAL NUMBER OF F.O.S. : 126

* THE NUMBER OF F.O.S. AFTER PARTITIONING : 513

* FAULT COLLAPSING

		Part A	Part B	Part C	
NUMBER OF CLOCK PERIOD	FAULTS FOUND	TIME OF FORWARD SIMULATION	TIME OF BACKWARD SIMULATION	TIME OF EXCEPTION SIMULATION	TIME OF SCIRTSS
1	19/1355	*	*	*	12.48
2	108/1355	0.29	0.21	0.45	6.58
10	724/1355	0.30	0.13	0.46	7.14
20	1007/1355	0.28	0.14	0.10	7.52
50	1146/1355	0.25	0.10	0.07	6.12
100	1191/1355	0.25	0.12	0.11	6.01
150	1215/1355	0.27	0.11	0.06	5.88
200	1232/1355	0.29	0.09	0.07	5.76
300	1243/1355	0.26	0.11	0.07	5.77

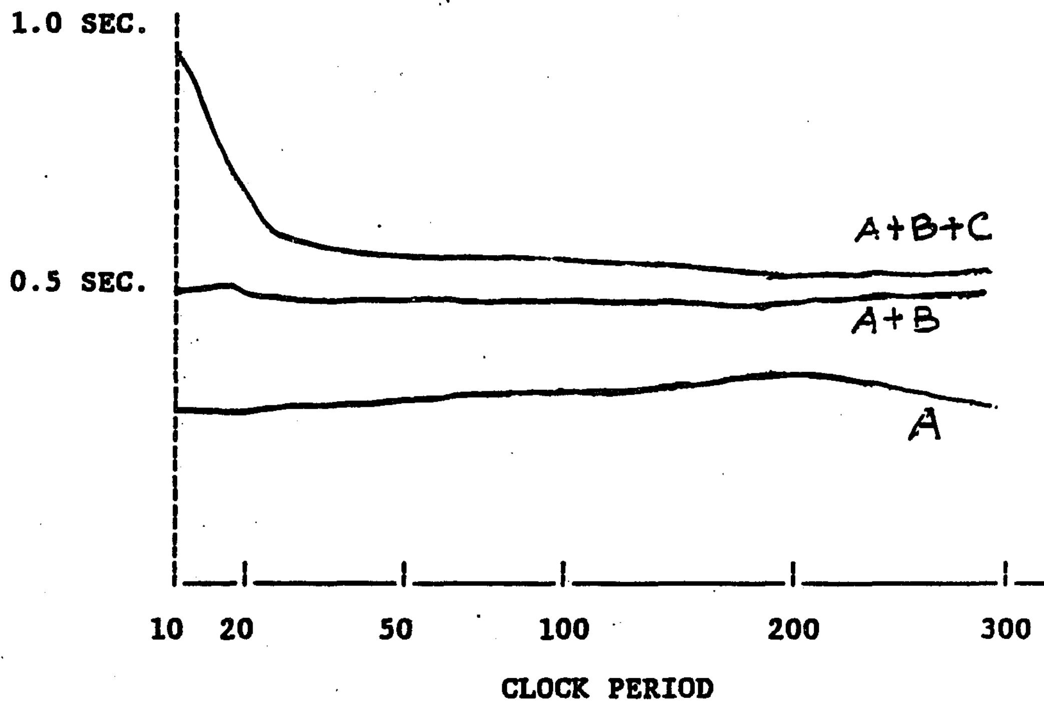


Figure 6.3 Total Time for Clock Period of SFSEE Simulation for Circuit ADD8X4.SCR

6.3 Circuit ADD8X8.SCR

Circuit ADD8X8.SCR is a larger circuit compared with the circuit ADD8X4.SCR. It has 74 external inputs, 75 flip_flops and 1246 circuit elements. The basic structure of this circuit is a cascade of adders constructed from eight 8 bit CLAADs. The original number of fan_out stems is 250, after partition the number of fan-out stems (FOS) of this circuit is 1025. (see Table 6.3) Because this circuit is a large circuit with 3858 faults, the parallel simulator needs more words to represent the position of faults. Within 50 periods, the average simulation time of parallel simulator is approximately 26.00 seconds, but only 3.00 seconds for the new simulator. When more faults have been detected, and less multiply stored faults and compound faults are stored, the new simulator starts enjoying an significant speed advantage over the parallel simulator. At the 300th period, the total time of the new simulator is 1.5 seconds, but that of parallel simulator is 24.91. On the average the new simulator is more than 15 times faster than that of the parallel simulator.

Table 6.3 Time Comparison Between SCIRTSS and SFSSE for Circuit ADD8X8.SCR

* ADD8X8.SCR

* THE ORIGINAL NUMBER OF F.O.S. : 250

* NUMBER OF F.O.S. AFTER PARTITIONING : 1025

* FAULT COLLAPSING

NUMBER OF CLOCK PERIOD	FAULTS FOUND	Part A: TIME OF FORWARD SIMULATION	Part B: TIME OF BACKWARD SIMULATION	Part C: TIME OF EXCEPTION SIMULATION	TIME OF SCIRTSS
1	19/3858	*	*	*	49.67
2	25/3858	0.86	0.41	1.33	24.48
10	861/3858	0.94	0.70	4.13	26.98
20	2177/3858	0.94	0.77	1.15	26.47
50	3091/3858	0.87	0.63	0.74	25.74
100	3298/3858	0.89	0.59	0.20	25.24
200	3376/3858	0.88	0.60	0.17	24.87
300	3405/3858	0.88	0.43	0.19	24.91

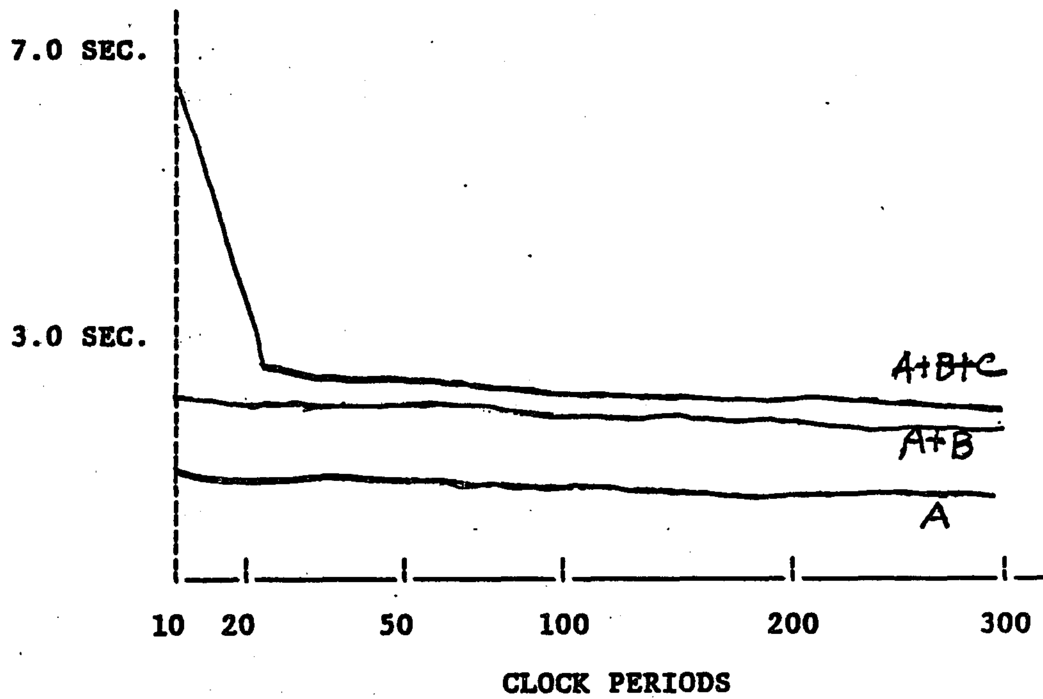


Figure 6.4 Total Time for Clock Period of SFSSE Simulation for Circuit ADD8X8.SCR

6.4 Circuit ADD8X16.SCR

Circuit ADD8X16.SCR is the largest circuit, with the same type, compared with the circuit ADD8X8.SCR and ADD8X4.SCR. It has 138 external inputs, 147 flip_flops and 2494 circuit elements. The original number of fan-out stems is 498, after partition the number of fan-out stems (FOS) is 2049. (see Table 6.4) Because this circuit is too large to be simulated by SCIRTSS at the present time, column 6 the total running time by SCIRTSS is left blank for the future work. Since the type of circuit ADD8X16.SCR is similar to the smaller circuit ADD8X8.SCR and ADD8X4.SCR, it is still meaningful to simulate this large circuit with the new simulator and to compare the running time with that of smaller circuits. As expected, the running time of Part A and Part B for each clock period is very close; with the more faults have been detected the less running time spent on Part C. The Part C is very decisive step which affects the total running time during the simulation. If the less multiply stored faults and compound faults are identified and stored in the memory elements, the less computer words are used to represent those faults positions for exceptional simulation. In Table 6.4, the 10th clock period for Part C the running time is 13.83 seconds, but for the 100th clock period only needs 0.74 seconds for Part C.

Table 6.4 Time Comparison Between SCIRTSS and SFSSE for Circuit ADD8X16.SCR

* ADD8X16.SCR

* THE ORIGINAL NUMBER OF F.O.S. : 498

* NUMBER OF F.O.S. AFTER PARTITIONING : 2049

* FAULT COLLAPSING

NUMBER OF CLOCK PERIOD	FAULTS FOUND	Part A TIME OF FORWARD SIMULATION	Part B TIME OF BACKWARD SIMULATION	Part C TIME OF EXCEPTION SIMULATION	TIME OF SCIRTSS
2	25/6356	2.59	1.75	3.98	
10	1529/6356	2.74	4.49	13.83	
20	3581/6356	2.54	4.10	10.69	
50	5376/6356	2.56	3.00	3.29	
100	5572/6356	2.03	2.39	0.74	
200	5743/6356	2.42	2.08	0.92	
300	3405/3858	2.53	1.26	0.44	

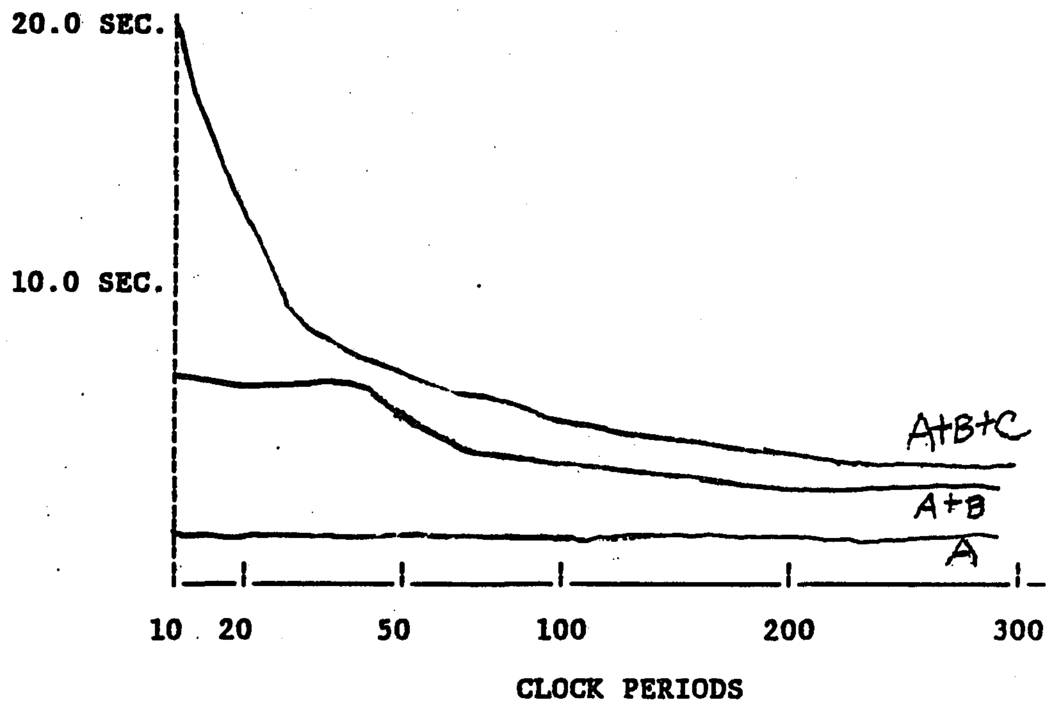


Figure 6.5 Total Time for Clock Period of SFSSE Simulation for Circuit ADD8X16.SCR

6.5 Circuit TEST88.SCR

Circuit TEST88.SCR is a sequential circuit with 74 external inputs, 67 flip_flops and 704 circuit elements. Originally the circuit has 129 fan_out stems , after the partition of the circuit it has 129 fan_out stems also. In the first period, the faults are stored in the memory elements the detected faults are found from the external outputs. From second period to 300th period, the time spent on Part A and Part B of each period is very close since the size of the circuit is fixed. However the time spent on Part C has tremendous decline. The reason for that is depending on the number of multiply stored faults and compound faults are found. As soon as the more faults have been detected, the less multiply stored faults and compound faults will be stored and the less time will be spent on Part C. (see Table 6.5) The total faults in circuit TEST88.SCR is 1976. Figure 6.6 shows the total time required by each clock period of faults simulation. In second period, 23 faults is detected using 0.87 second which is only 1/10 comparing with the time of parallel simulation. The 300th period only needs 0.50 second which is 1/14 of the total running time of SCIRTSS. The curve for the new simulator declines dramatically, but the parallel simulator uses a almost the same time for each clock period of the fault simulation.

Table 6.5 Time Comparison Between SCIRTSS and SFSSE for Circuit TEST88.SCR

* TEST88.SCR

* THE ORIGINAL NUMBER OF F.O.S. : 129

* NUMBER OF F.O.S. AFTER PARTITIONING : 129

* FAULT COLLAPSING

NUMBER OF CLOCK PERIOD	FAULTS FOUND	Part A: TIME OF FORWARD SIMULATION	Part B: TIME OF BACKWARD SIMULATION	Part C: TIME OF EXCEPTION SIMULATION	TIME OF SCIRTSS
1	17/1976	*	*	*	14.78
2	23/1976	0.11	0.13	0.63	8.46
10	298/1976	0.13	0.17	0.68	8.21
20	868/1976	0.12	0.12	0.39	7.81
50	1276/1976	0.12	0.11	0.36	7.95
100	1568/1976	0.12	0.10	0.31	7.60
200	1735/1976	0.12	0.10	0.28	7.62
300	1819/1976	0.12	0.10	0.28	7.23

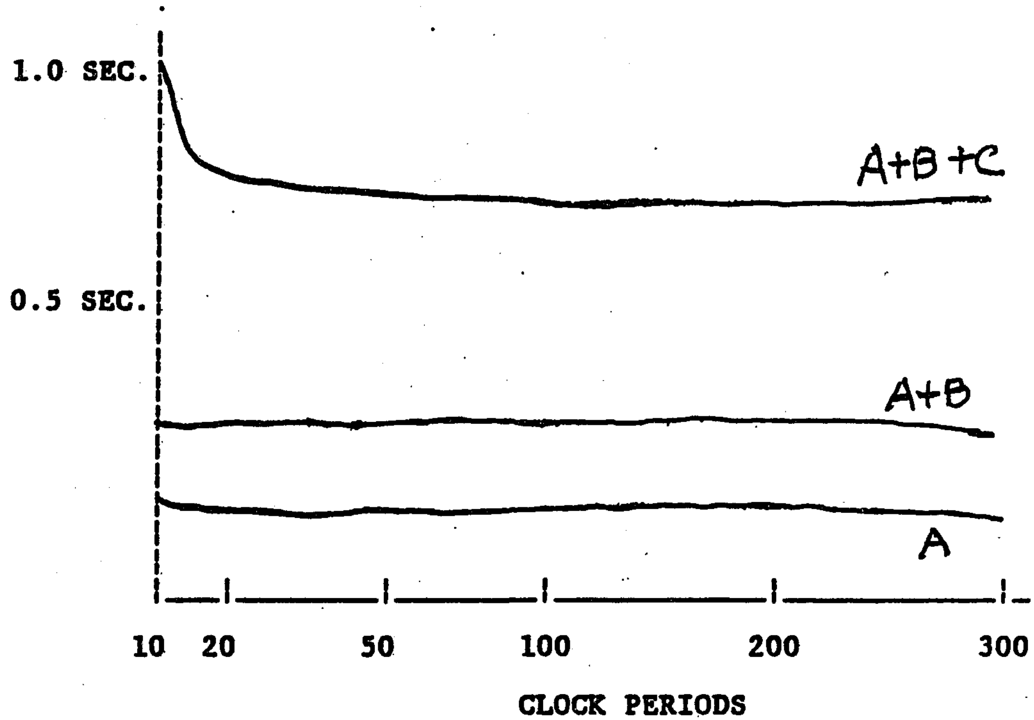


Figure 6.6 Total Time for Clock Period of SFSSE Simulation for Circuit TEST88.SCR

6.6 Circuit MSFCF4.SCR

Circuit MSFCF4.SCR is a special circuit for the purpose of testing the new simulator. There are 36 external inputs, 48 flip_flops and 603 circuit elements. Comparing with the circuit TEST88.SCR, MSFCF1.SCR is a smaller circuit but with more fan_out stems 539, more multiply stored faults and more compound faults. The total number of faults in this circuit is 1880. Within the first 50 periods , it suffered to spend time on simulating the large amount of multiply stored faults and compound faults that slows down the new simulator but spent only half of the time that parallel simulator did. (see Table 6.6) When the more faults have been detected, the less multiply stored faults and compound faults are found, the new simulator enjoys the same as for the previous circuits speed advantage over the parallel simulator. The total time of new simulator only needs 1/10 time of parallel simulator to the 300th period.

Table 6.6 Time Comparison Between SCIRTSS and SFSSE for Circuit MSFCF4.SCR

* MSFCF4.SCR

* THE ORIGINAL NUMBER OF F.O.S. : 128

* NUMBER OF F.O.S. AFTER PARTITIONING : 539

* FAULT COLLAPSING

		Part A	Part B	Part C	
NUMBER OF CLOCK PERIOD	FAULTS FOUND	TIME OF FORWARD SIMULATION	TIME OF BACKWARD SIMULATION	TIME OF EXCEPTION SIMULATION	TIME OF SCIRTSS
1	18/1880	*	*	*	11.43
2	19/1880	0.33	0.25	0.47	5.51
10	236/1880	0.42	0.36	1.62	5.71
20	623/1880	0.42	0.41	2.73	5.73
50	1398/1880	0.46	0.33	1.08	5.69
100	1660/1880	0.30	0.20	0.38	5.59
200	1731/1880	0.29	0.17	0.08	5.51
300	1746/1880	0.29	0.18	0.10	5.54

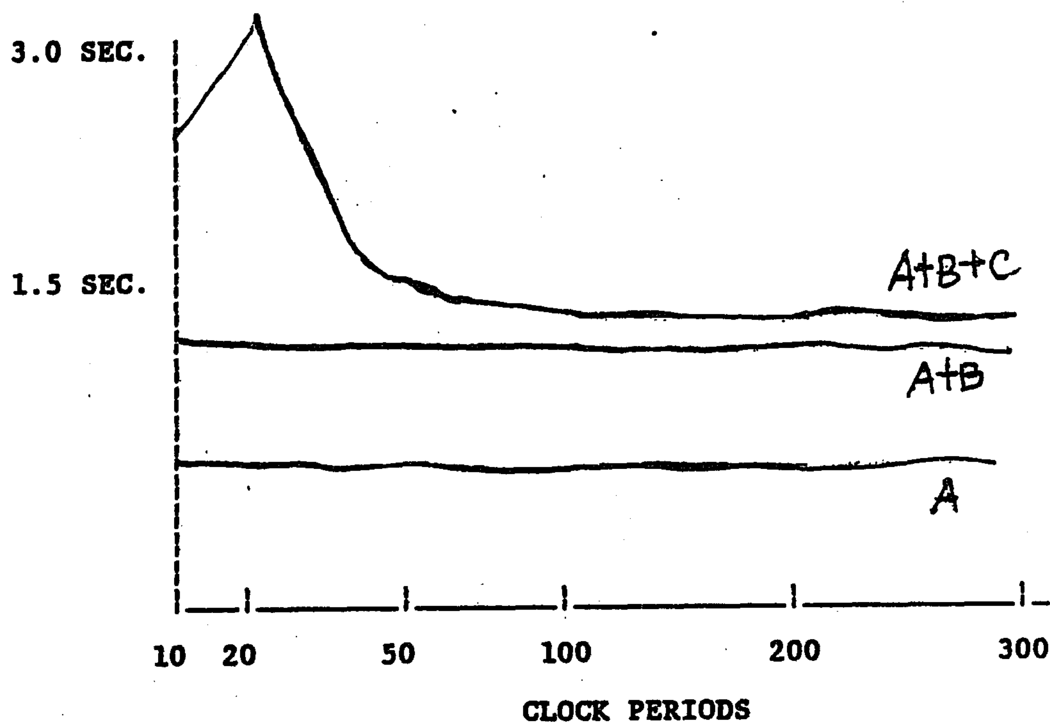


Figure 6.7 Total Time for Clock Period of SFSSE Simulation for Circuit MSFCM4.SCR

CHAPTER 7

SUMMARY AND SUGGESTIONS FOR FURTHER WORK

The simulation time for Part A and Part B is relatively constant for the selected six circuits in the new simulator. The reason for that is to simulate the good value and the sensitive fan-out stems on the fixed size of circuit will use the approximately same time. However, the simulation time for Part C depends on how many multiply stored faults and compound faults are stored. During the first five or ten clock periods, there are many multiply stored faults and compound faults are stored, the simulation time on Part C is increasing. When more faults have been detected, the less multiply stored faults and compound faults are stored and the simulation time for Part C declines.

The circuit sizes for the testing of the new simulator include small circuit with 156 circuit elements and large circuit with 1246 circuit elements. The simulation of multiply stored faults and compound faults in each circuit demonstrated the advantage of the new simulator over the parallel simulator. The result of

testing the new simulator using 6 circuits was successful. The analysis of 5 circuits indicate that the time spent by the synchronous faults simulation by surrogate with exceptions is ten times better than that of the parallel simulator.

Since the current implementation dose not make the fault location the same with that of SCIRTSS, its upgrading to reassign the fault location is necessary.

APPENDIX 1 : CIRCUIT DESCRIPTION OF ADD8X8.SCR

CIRCUIT DESCRIPTION:

ELEMENT	TYPE	INPUT CONNECTIONS	
1	DFF	140	
2	DFF	141	
3	DFF	142	
4	DFF	226	2
5	DFF	231	2
6	DFF	236	2
7	DFF	241	2
8	DFF	246	2
9	DFF	251	2
10	DFF	256	2
11	DFF	261	2
12	DFF	266	2
13	DFF	76	77
14	DFF	78	77
15	DFF	79	77
16	DFF	80	77
17	DFF	81	77
18	DFF	82	77
19	DFF	83	77
20	DFF	84	77
21	DFF	85	77
22	DFF	86	77
23	DFF	87	77
24	DFF	88	77
25	DFF	89	77
26	DFF	90	77
27	DFF	91	77
28	DFF	92	77
29	DFF	93	77
30	DFF	94	77
31	DFF	95	77
32	DFF	96	77
33	DFF	97	77
34	DFF	98	77
35	DFF	99	77
36	DFF	100	77
37	DFF	101	77
38	DFF	102	77
39	DFF	103	77
40	DFF	104	77
41	DFF	105	77
42	DFF	106	77
43	DFF	107	77

44	DFF	108	77
45	DFF	109	77
46	DFF	110	77
47	DFF	111	77
48	DFF	112	77
49	DFF	113	77
50	DFF	114	77
51	DFF	115	77
52	DFF	116	77
53	DFF	117	77
54	DFF	118	77
55	DFF	119	77
56	DFF	120	77
57	DFF	121	77
58	DFF	122	77
59	DFF	123	77
60	DFF	124	77
61	DFF	125	77
62	DFF	126	77
63	DFF	127	77
64	DFF	128	77
65	DFF	129	77
66	DFF	130	77
67	DFF	131	77
68	DFF	132	77
69	DFF	133	77
70	DFF	134	77
71	DFF	135	77
72	DFF	136	77
73	DFF	137	77
74	DFF	138	77
75	DFF	139	77
76	OR	387	1173
77	OR	2	3
78	OR	388	1174
79	OR	389	1175
80	OR	390	1176
81	OR	391	1177
82	OR	392	1178
83	OR	393	1179
84	OR	394	1180
85	OR	395	1181
86	OR	516	1182
87	OR	517	1183
88	OR	518	1184
89	OR	519	1185
90	OR	520	1186
91	OR	521	1187
92	OR	522	1188

93	OR	523	1189
94	OR	524	1190
95	OR	645	1191
96	OR	646	1192
97	OR	647	1193
98	OR	648	1194
99	OR	649	1195
100	OR	650	1196
101	OR	651	1197
102	OR	652	1198
103	OR	653	1199
104	OR	774	1200
105	OR	775	1201
106	OR	776	1202
107	OR	777	1203
108	OR	778	1204
109	OR	779	1205
110	OR	780	1206
111	OR	781	1207
112	OR	782	1208
113	OR	903	1209
114	OR	904	1210
115	OR	905	1211
116	OR	906	1212
117	OR	907	1213
118	OR	908	1214
119	OR	909	1215
120	OR	910	1216
121	OR	911	1217
122	OR	1032	1218
123	OR	1033	1219
124	OR	1034	1220
125	OR	1035	1221
126	OR	1036	1222
127	OR	1037	1223
128	OR	1038	1224
129	OR	1039	1225
130	OR	1040	1226
131	OR	1161	1227
132	OR	1162	1228
133	OR	1163	1229
134	OR	1164	1230
135	OR	1165	1231
136	OR	1166	1232
137	OR	1167	1233
138	OR	1168	1234
139	OR	1169	1235
140	OR	144	1236
141	OR	146	1172

142	OR	1171	1237
143	NAND	11073	
144	AND	1	143
145	NAND	143	
146	AND	1	145
147	AND	11009	11001
148	NAND	11009	
149	AND	148	11001
150	NAND	11001	
151	AND	11009	150
152	OR	151	149
153	AND	11010	11002
154	NAND	11010	
155	AND	154	11002
156	NAND	11002	
157	AND	11010	156
158	OR	157	155
159	AND	11011	11003
160	NAND	11011	
161	AND	160	11003
162	NAND	11003	
163	AND	11011	162
164	OR	163	161
165	AND	11012	11004
166	NAND	11012	
167	AND	166	11004
168	NAND	11004	
169	AND	11012	168
170	OR	169	167
171	AND	11013	11005
172	NAND	11013	
173	AND	172	11005
174	NAND	11005	
175	AND	11013	174
176	OR	175	173
177	AND	11014	11006
178	NAND	11014	
179	AND	178	11006
180	NAND	11006	
181	AND	11014	180
182	OR	181	179
183	AND	11015	11007
184	NAND	11015	
185	AND	184	11007
186	NAND	11007	
187	AND	11015	186
188	OR	187	185
189	AND	11016	11008
190	NAND	11016	

191	AND	190	11008		
192	NAND	11008			
193	AND	11016	192		
194	OR	193	191		
195	AND	-11000	194		
196	OR	195	189		
197	AND	189	188		
198	AND	-11000	188	194	
199	OR	198	197	183	
200	AND	183	182		
201	AND	189	182	188	
202	AND	-11000	182	188	194
203	OR	202	201	200	177
204	AND	177	176		
205	AND	183	176	182	
206	AND	189	176	182	188
207	OR	206	205	204	171
208	AND	176	182	188	194
209	AND	-11000	208		
210	OR	209	207		
211	AND	210	170		
212	OR	211	165		
213	AND	165	164		
214	AND	210	164	170	
215	OR	214	213	159	
216	AND	159	158		
217	AND	165	158	164	
218	AND	210	158	164	170
219	OR	218	217	216	153
220	AND	153	152		
221	AND	159	152	158	
222	AND	165	152	158	164
223	OR	222	221	220	147
224	AND	152	158	164	170
225	AND	210	224		
226	OR	225	223		
227	NAND	219			
228	AND	227	152		
229	NAND	152			
230	AND	219	229		
231	OR	230	228		
232	NAND	215			
233	AND	232	158		
234	NAND	158			
235	AND	215	234		
236	OR	235	233		
237	NAND	212			
238	AND	237	164		
239	NAND	164			

240	AND	212	239
241	OR	240	238
242	NAND	210	
243	AND	242	170
244	NAND	170	
245	AND	210	244
246	OR	245	243
247	NAND	203	
248	AND	247	176
249	NAND	176	
250	AND	203	249
251	OR	250	248
252	NAND	199	
253	AND	252	182
254	NAND	182	
255	AND	199	254
256	OR	255	253
257	NAND	196	
258	AND	257	188
259	NAND	188	
260	AND	196	259
261	OR	260	258
262	NAND	-11000	
263	AND	262	194
264	NAND	194	
265	AND	-11000	264
266	OR	265	263
267	AND	11017	5
268	NAND	11017	
269	AND	268	5
270	NAND	5	
271	AND	11017	270
272	OR	271	269
273	AND	11018	6
274	NAND	11018	
275	AND	274	6
276	NAND	6	
277	AND	11018	276
278	OR	277	275
279	AND	11019	7
280	NAND	11019	
281	AND	280	7
282	NAND	7	
283	AND	11019	282
284	OR	283	281
285	AND	11020	8
286	NAND	11020	
287	AND	286	8
288	NAND	8	

289	AND	11020	288		
290	OR	289	287		
291	AND	11021	9		
292	NAND	11021			
293	AND	292	9		
294	NAND	9			
295	AND	11021	294		
296	OR	295	293		
297	AND	11022	10		
298	NAND	11022			
299	AND	298	10		
300	NAND	10			
301	AND	11022	300		
302	OR	301	299		
303	AND	11023	11		
304	NAND	11023			
305	AND	304	11		
306	NAND	11			
307	AND	11023	306		
308	OR	307	305		
309	AND	11024	12		
310	NAND	11024			
311	AND	310	12		
312	NAND	12			
313	AND	11024	312		
314	OR	313	311		
315	AND	-11000	314		
316	OR	315	309		
317	AND	309	308		
318	AND	-11000	308	314	
319	OR	318	317	303	
320	AND	303	302		
321	AND	309	302	308	
322	AND	-11000	302	308	314
323	OR	322	321	320	297
324	AND	297	296		
325	AND	303	296	302	
326	AND	309	296	302	308
327	OR	326	325	324	291
328	AND	296	302	308	314
329	AND	-11000	328		
330	OR	329	327		
331	AND	330	290		
332	OR	331	285		
333	AND	285	284		
334	AND	330	284	290	
335	OR	334	333	279	
336	AND	279	278		
337	AND	285	278	284	

338	AND	330	278	284	290
339	OR	338	337	336	273
340	AND	273	272		
341	AND	279	272	278	
342	AND	285	272	278	284
343	OR	342	341	340	267
344	AND	272	278	284	290
345	AND	330	344		
346	OR	345	343		
347	NAND	339			
348	AND	347	272		
349	NAND	272			
350	AND	339	349		
351	OR	350	348		
352	NAND	335			
353	AND	352	278		
354	NAND	278			
355	AND	335	354		
356	OR	355	353		
357	NAND	332			
358	AND	357	284		
359	NAND	284			
360	AND	332	359		
361	OR	360	358		
362	NAND	330			
363	AND	362	290		
364	NAND	290			
365	AND	330	364		
366	OR	365	363		
367	NAND	323			
368	AND	367	296		
369	NAND	296			
370	AND	323	369		
371	OR	370	368		
372	NAND	319			
373	AND	372	302		
374	NAND	302			
375	AND	319	374		
376	OR	375	373		
377	NAND	316			
378	AND	377	308		
379	NAND	308			
380	AND	316	379		
381	OR	380	378		
382	NAND	-11000			
383	AND	382	314		
384	NAND	314			
385	AND	-11000	384		
386	OR	385	383		

387	AND	346	2
388	AND	351	2
389	AND	356	2
390	AND	361	2
391	AND	366	2
392	AND	371	2
393	AND	376	2
394	AND	381	2
395	AND	386	2
396	AND	11025	14
397	NAND	11025	
398	AND	397	14
399	NAND	14	
400	AND	11025	399
401	OR	400	398
402	AND	11026	15
403	NAND	11026	
404	AND	403	15
405	NAND	15	
406	AND	11026	405
407	OR	406	404
408	AND	11027	16
409	NAND	11027	
410	AND	409	16
411	NAND	16	
412	AND	11027	411
413	OR	412	410
414	AND	11028	17
415	NAND	11028	
416	AND	415	17
417	NAND	17	
418	AND	11028	417
419	OR	418	416
420	AND	11029	18
421	NAND	11029	
422	AND	421	18
423	NAND	18	
424	AND	11029	423
425	OR	424	422
426	AND	11030	19
427	NAND	11030	
428	AND	427	19
429	NAND	19	
430	AND	11030	429
431	OR	430	428
432	AND	11031	20
433	NAND	11031	
434	AND	433	20
435	NAND	20	

436	AND	11031	435		
437	OR	436	434		
438	AND	11032	21		
439	NAND	11032			
440	AND	439	21		
441	NAND	21			
442	AND	11032	441		
443	OR	442	440		
444	AND	-11000	443		
445	OR	444	438		
446	AND	438	437		
447	AND	-11000	437	443	
448	OR	447	446	432	
449	AND	432	431		
450	AND	438	431	437	
451	AND	-11000	431	437	443
452	OR	451	450	449	426
453	AND	426	425		
454	AND	432	425	431	
455	AND	438	425	431	437
456	OR	455	454	453	420
457	AND	425	431	437	443
458	AND	-11000	457		
459	OR	458	456		
460	AND	459	419		
461	OR	460	414		
462	AND	414	413		
463	AND	459	413	419	
464	OR	463	462	408	
465	AND	408	407		
466	AND	414	407	413	
467	AND	459	407	413	419
468	OR	467	466	465	402
469	AND	402	401		
470	AND	408	401	407	
471	AND	414	401	407	413
472	OR	471	470	469	396
473	AND	401	407	413	419
474	AND	459	473		
475	OR	474	472		
476	NAND	468			
477	AND	476	401		
478	NAND	401			
479	AND	468	478		
480	OR	479	477		
481	NAND	464			
482	AND	481	407		
483	NAND	407			
484	AND	464	483		

485	OR	484	482
486	NAND	461	
487	AND	486	413
488	NAND	413	
489	AND	461	488
490	OR	489	487
491	NAND	459	
492	AND	491	419
493	NAND	419	
494	AND	459	493
495	OR	494	492
496	NAND	452	
497	AND	496	425
498	NAND	425	
499	AND	452	498
500	OR	499	497
501	NAND	448	
502	AND	501	431
503	NAND	431	
504	AND	448	503
505	OR	504	502
506	NAND	445	
507	AND	506	437
508	NAND	437	
509	AND	445	508
510	OR	509	507
511	NAND	-11000	
512	AND	511	443
513	NAND	443	
514	AND	-11000	513
515	OR	514	512
516	AND	475	2
517	AND	480	2
518	AND	485	2
519	AND	490	2
520	AND	495	2
521	AND	500	2
522	AND	505	2
523	AND	510	2
524	AND	515	2
525	AND	11033	23
526	NAND	11033	
527	AND	526	23
528	NAND	23	
529	AND	11033	528
530	OR	529	527
531	AND	11034	24
532	NAND	11034	
533	AND	532	24

534	NAND	24			
535	AND	11034	534		
536	OR	535	533		
537	AND	11035	25		
538	NAND	11035			
539	AND	538	25		
540	NAND	25			
541	AND	11035	540		
542	OR	541	539		
543	AND	11036	26		
544	NAND	11036			
545	AND	544	26		
546	NAND	26			
547	AND	11036	546		
548	OR	547	545		
549	AND	11037	27		
550	NAND	11037			
551	AND	550	27		
552	NAND	27			
553	AND	11037	552		
554	OR	553	551		
555	AND	11038	28		
556	NAND	11038			
557	AND	556	28		
558	NAND	28			
559	AND	11038	558		
560	OR	559	557		
561	AND	11039	29		
562	NAND	11039			
563	AND	562	29		
564	NAND	29			
565	AND	11039	564		
566	OR	565	563		
567	AND	11040	30		
568	NAND	11040			
569	AND	568	30		
570	NAND	30			
571	AND	11040	570		
572	OR	571	569		
573	AND	-11000	572		
574	OR	573	567		
575	AND	567	566		
576	AND	-11000	566	572	
577	OR	576	575	561	
578	AND	561	560		
579	AND	567	560	566	
580	AND	-11000	560	566	572
581	OR	580	579	578	555
582	AND	555	554		

583	AND	561	554	560	
584	AND	567	554	560	566
585	OR	584	583	582	549
586	AND	554	560	566	572
587	AND	-11000	586		
588	OR	587	585		
589	AND	588	548		
590	OR	589	543		
591	AND	543	542		
592	AND	588	542	548	
593	OR	592	591	537	
594	AND	537	536		
595	AND	543	536	542	
596	AND	588	536	542	548
597	OR	596	595	594	531
598	AND	531	530		
599	AND	537	530	536	
600	AND	543	530	536	542
601	OR	600	599	598	525
602	AND	530	536	542	548
603	AND	588	602		
604	OR	603	601		
605	NAND	597			
606	AND	605	530		
607	NAND	530			
608	AND	597	607		
609	OR	608	606		
610	NAND	593			
611	AND	610	536		
612	NAND	536			
613	AND	593	612		
614	OR	613	611		
615	NAND	590			
616	AND	615	542		
617	NAND	542			
618	AND	590	617		
619	OR	618	616		
620	NAND	588			
621	AND	620	548		
622	NAND	548			
623	AND	588	622		
624	OR	623	621		
625	NAND	581			
626	AND	625	554		
627	NAND	554			
628	AND	581	627		
629	OR	628	626		
630	NAND	577			
631	AND	630	560		

632	NAND	560	
633	AND	577	632
634	OR	633	631
635	NAND	574	
636	AND	635	566
637	NAND	566	
638	AND	574	637
639	OR	638	636
640	NAND	-11000	
641	AND	640	572
642	NAND	572	
643	AND	-11000	642
644	OR	643	641
645	AND	604	2
646	AND	609	2
647	AND	614	2
648	AND	619	2
649	AND	624	2
650	AND	629	2
651	AND	634	2
652	AND	639	2
653	AND	644	2
654	AND	11041	32
655	NAND	11041	
656	AND	655	32
657	NAND	32	
658	AND	11041	657
659	OR	658	656
660	AND	11042	33
661	NAND	11042	
662	AND	661	33
663	NAND	33	
664	AND	11042	663
665	OR	664	662
666	AND	11043	34
667	NAND	11043	
668	AND	667	34
669	NAND	34	
670	AND	11043	669
671	OR	670	668
672	AND	11044	35
673	NAND	11044	
674	AND	673	35
675	NAND	35	
676	AND	11044	675
677	OR	676	674
678	AND	11045	36
679	NAND	11045	
680	AND	679	36

681	NAND	36			
682	AND	11045	681		
683	OR	682	680		
684	AND	11046	37		
685	NAND	11046			
686	AND	685	37		
687	NAND	37			
688	AND	11046	687		
689	OR	688	686		
690	AND	11047	38		
691	NAND	11047			
692	AND	691	38		
693	NAND	38			
694	AND	11047	693		
695	OR	694	692		
696	AND	11048	39		
697	NAND	11048			
698	AND	697	39		
699	NAND	39			
700	AND	11048	699		
701	OR	700	698		
702	AND	-11000	701		
703	OR	702	696		
704	AND	696	695		
705	AND	-11000	695	701	
706	OR	705	704	690	
707	AND	690	689		
708	AND	696	689	695	
709	AND	-11000	689	695	701
710	OR	709	708	707	684
711	AND	684	683		
712	AND	690	683	689	
713	AND	696	683	689	695
714	OR	713	712	711	678
715	AND	683	689	695	701
716	AND	-11000	715		
717	OR	716	714		
718	AND	717	677		
719	OR	718	672		
720	AND	672	671		
721	AND	717	671	677	
722	OR	721	720	666	
723	AND	666	665		
724	AND	672	665	671	
725	AND	717	665	671	677
726	OR	725	724	723	660
727	AND	660	659		
728	AND	666	659	665	
729	AND	672	659	665	671

730	OR	729	728	727	654
731	AND	659	665	671	677
732	AND	717	731		
733	OR	732	730		
734	NAND	726			
735	AND	734	659		
736	NAND	659			
737	AND	726	736		
738	OR	737	735		
739	NAND	722			
740	AND	739	665		
741	NAND	665			
742	AND	722	741		
743	OR	742	740		
744	NAND	719			
745	AND	744	671		
746	NAND	671			
747	AND	719	746		
748	OR	747	745		
749	NAND	717			
750	AND	749	677		
751	NAND	677			
752	AND	717	751		
753	OR	752	750		
754	NAND	710			
755	AND	754	683		
756	NAND	683			
757	AND	710	756		
758	OR	757	755		
759	NAND	706			
760	AND	759	689		
761	NAND	689			
762	AND	706	761		
763	OR	762	760		
764	NAND	703			
765	AND	764	695		
766	NAND	695			
767	AND	703	766		
768	OR	767	765		
769	NAND	-11000			
770	AND	769	701		
771	NAND	701			
772	AND	-11000	771		
773	OR	772	770		
774	AND	733	2		
775	AND	738	2		
776	AND	743	2		
777	AND	748	2		
778	AND	753	2		

779	AND	758	2
780	AND	763	2
781	AND	768	2
782	AND	773	2
783	AND	11049	41
784	NAND	11049	
785	AND	784	41
786	NAND	41	
787	AND	11049	786
788	OR	787	785
789	AND	11050	42
790	NAND	11050	
791	AND	790	42
792	NAND	42	
793	AND	11050	792
794	OR	793	791
795	AND	11051	43
796	NAND	11051	
797	AND	796	43
798	NAND	43	
799	AND	11051	798
800	OR	799	797
801	AND	11052	44
802	NAND	11052	
803	AND	802	44
804	NAND	44	
805	AND	11052	804
806	OR	805	803
807	AND	11053	45
808	NAND	11053	
809	AND	808	45
810	NAND	45	
811	AND	11053	810
812	OR	811	809
813	AND	11054	46
814	NAND	11054	
815	AND	814	46
816	NAND	46	
817	AND	11054	816
818	OR	817	815
819	AND	11055	47
820	NAND	11055	
821	AND	820	47
822	NAND	47	
823	AND	11055	822
824	OR	823	821
825	AND	11056	48
826	NAND	11056	
827	AND	826	48

828	NAND	48			
829	AND	11056	828		
830	OR	829	827		
831	AND	-11000	830		
832	OR	831	825		
833	AND	825	824		
834	AND	-11000	824	830	
835	OR	834	833	819	
836	AND	819	818		
837	AND	825	818	824	
838	AND	-11000	818	824	830
839	OR	838	837	836	813
840	AND	813	812		
841	AND	819	812	818	
842	AND	825	812	818	824
843	OR	842	841	840	807
844	AND	812	818	824	830
845	AND	-11000	844		
846	OR	845	843		
847	AND	846	806		
848	OR	847	801		
849	AND	801	800		
850	AND	846	800	806	
851	OR	850	849	795	
852	AND	795	794		
853	AND	801	794	800	
854	AND	846	794	800	806
855	OR	854	853	852	789
856	AND	789	788		
857	AND	795	788	794	
858	AND	801	788	794	800
859	OR	858	857	856	783
860	AND	788	794	800	806
861	AND	846	860		
862	OR	861	859		
863	NAND	855			
864	AND	863	788		
865	NAND	788			
866	AND	855	865		
867	OR	866	864		
868	NAND	851			
869	AND	868	794		
870	NAND	794			
871	AND	851	870		
872	OR	871	869		
873	NAND	848			
874	AND	873	800		
875	NAND	800			
876	AND	848	875		

877	OR	876	874
878	NAND	846	
879	AND	878	806
880	NAND	806	
881	AND	846	880
882	OR	881	879
883	NAND	839	
884	AND	883	812
885	NAND	812	
886	AND	839	885
887	OR	886	884
888	NAND	835	
889	AND	888	818
890	NAND	818	
891	AND	835	890
892	OR	891	889
893	NAND	832	
894	AND	893	824
895	NAND	824	
896	AND	832	895
897	OR	896	894
898	NAND	-11000	
899	AND	898	830
900	NAND	830	
901	AND	-11000	900
902	OR	901	899
903	AND	862	2
904	AND	867	2
905	AND	872	2
906	AND	877	2
907	AND	882	2
908	AND	887	2
909	AND	892	2
910	AND	897	2
911	AND	902	2
912	AND	11057	50
913	NAND	11057	
914	AND	913	50
915	NAND	50	
916	AND	11057	915
917	OR	916	914
918	AND	11058	51
919	NAND	11058	
920	AND	919	51
921	NAND	51	
922	AND	11058	921
923	OR	922	920
924	AND	11059	52
925	NAND	11059	

926	AND	925	52		
927	NAND	52			
928	AND	11059	927		
929	OR	928	926		
930	AND	11060	53		
931	NAND	11060			
932	AND	931	53		
933	NAND	53			
934	AND	11060	933		
935	OR	934	932		
936	AND	11061	54		
937	NAND	11061			
938	AND	937	54		
939	NAND	54			
940	AND	11061	939		
941	OR	940	938		
942	AND	11062	55		
943	NAND	11062			
944	AND	943	55		
945	NAND	55			
946	AND	11062	945		
947	OR	946	944		
948	AND	11063	56		
949	NAND	11063			
950	AND	949	56		
951	NAND	56			
952	AND	11063	951		
953	OR	952	950		
954	AND	11064	57		
955	NAND	11064			
956	AND	955	57		
957	NAND	57			
958	AND	11064	957		
959	OR	958	956		
960	AND	-11000	959		
961	OR	960	954		
962	AND	954	953		
963	AND	-11000	953	959	
964	OR	963	962	948	
965	AND	948	947		
966	AND	954	947	953	
967	AND	-11000	947	953	959
968	OR	967	966	965	942
969	AND	942	941		
970	AND	948	941	947	
971	AND	954	941	947	953
972	OR	971	970	969	936
973	AND	941	947	953	959
974	AND	-11000	973		

975	OR	974	972		
976	AND	975	935		
977	OR	976	930		
978	AND	930	929		
979	AND	975	929	935	
980	OR	979	978	924	
981	AND	924	923		
982	AND	930	923	929	
983	AND	975	923	929	935
984	OR	983	982	981	918
985	AND	918	917		
986	AND	924	917	923	
987	AND	930	917	923	929
988	OR	987	986	985	912
989	AND	917	923	929	935
990	AND	975	989		
991	OR	990	988		
992	NAND	984			
993	AND	992	917		
994	NAND	917			
995	AND	984	994		
996	OR	995	993		
997	NAND	980			
998	AND	997	923		
999	NAND	923			
1000	AND	980	999		
1001	OR	1000	998		
1002	NAND	977			
1003	AND	1002	929		
1004	NAND	929			
1005	AND	977	1004		
1006	OR	1005	1003		
1007	NAND	975			
1008	AND	1007	935		
1009	NAND	935			
1010	AND	975	1009		
1011	OR	1010	1008		
1012	NAND	968			
1013	AND	1012	941		
1014	NAND	941			
1015	AND	968	1014		
1016	OR	1015	1013		
1017	NAND	964			
1018	AND	1017	947		
1019	NAND	947			
1020	AND	964	1019		
1021	OR	1020	1018		
1022	NAND	961			
1023	AND	1022	953		

1024	NAND	953	
1025	AND	961	1024
1026	OR	1025	1023
1027	NAND	-11000	
1028	AND	1027	959
1029	NAND	959	
1030	AND	-11000	1029
1031	OR	1030	1028
1032	AND	991	2
1033	AND	996	2
1034	AND	1001	2
1035	AND	1006	2
1036	AND	1011	2
1037	AND	1016	2
1038	AND	1021	2
1039	AND	1026	2
1040	AND	1031	2
1041	AND	11065	59
1042	NAND	11065	
1043	AND	1042	59
1044	NAND	59	
1045	AND	11065	1044
1046	OR	1045	1043
1047	AND	11066	60
1048	NAND	11066	
1049	AND	1048	60
1050	NAND	60	
1051	AND	11066	1050
1052	OR	1051	1049
1053	AND	11067	61
1054	NAND	11067	
1055	AND	1054	61
1056	NAND	61	
1057	AND	11067	1056
1058	OR	1057	1055
1059	AND	11068	62
1060	NAND	11068	
1061	AND	1060	62
1062	NAND	62	
1063	AND	11068	1062
1064	OR	1063	1061
1065	AND	11069	63
1066	NAND	11069	
1067	AND	1066	63
1068	NAND	63	
1069	AND	11069	1068
1070	OR	1069	1067
1071	AND	11070	64
1072	NAND	11070	

1073	AND	1072	64		
1074	NAND	64			
1075	AND	11070	1074		
1076	OR	1075	1073		
1077	AND	11071	65		
1078	NAND	11071			
1079	AND	1078	65		
1080	NAND	65			
1081	AND	11071	1080		
1082	OR	1081	1079		
1083	AND	11072	66		
1084	NAND	11072			
1085	AND	1084	66		
1086	NAND	66			
1087	AND	11072	1086		
1088	OR	1087	1085		
1089	AND	-11000	1088		
1090	OR	1089	1083		
1091	AND	1083	1082		
1092	AND	-11000	1082	1088	
1093	OR	1092	1091	1077	
1094	AND	1077	1076		
1095	AND	1083	1076	1082	
1096	AND	-11000	1076	1082	1088
1097	OR	1096	1095	1094	1071
1098	AND	1071	1070		
1099	AND	1077	1070	1076	
1100	AND	1083	1070	1076	1082
1101	OR	1100	1099	1098	1065
1102	AND	1070	1076	1082	1088
1103	AND	-11000	1102		
1104	OR	1103	1101		
1105	AND	1104	1064		
1106	OR	1105	1059		
1107	AND	1059	1058		
1108	AND	1104	1058	1064	
1109	OR	1108	1107	1053	
1110	AND	1053	1052		
1111	AND	1059	1052	1058	
1112	AND	1104	1052	1058	1064
1113	OR	1112	1111	1110	1047
1114	AND	1047	1046		
1115	AND	1053	1046	1052	
1116	AND	1059	1046	1052	1058
1117	OR	1116	1115	1114	1041
1118	AND	1046	1052	1058	1064
1119	AND	1104	1118		
1120	OR	1119	1117		
1121	NAND	1113			

1122	AND	1121	1046
1123	NAND	1046	
1124	AND	1113	1123
1125	OR	1124	1122
1126	NAND	1109	
1127	AND	1126	1052
1128	NAND	1052	
1129	AND	1109	1128
1130	OR	1129	1127
1131	NAND	1106	
1132	AND	1131	1058
1133	NAND	1058	
1134	AND	1106	1133
1135	OR	1134	1132
1136	NAND	1104	
1137	AND	1136	1064
1138	NAND	1064	
1139	AND	1104	1138
1140	OR	1139	1137
1141	NAND	1097	
1142	AND	1141	1070
1143	NAND	1070	
1144	AND	1097	1143
1145	OR	1144	1142
1146	NAND	1093	
1147	AND	1146	1076
1148	NAND	1076	
1149	AND	1093	1148
1150	OR	1149	1147
1151	NAND	1090	
1152	AND	1151	1082
1153	NAND	1082	
1154	AND	1090	1153
1155	OR	1154	1152
1156	NAND	-11000	
1157	AND	1156	1088
1158	NAND	1088	
1159	AND	-11000	1158
1160	OR	1159	1157
1161	AND	1120	2
1162	AND	1125	2
1163	AND	1130	2
1164	AND	1135	2
1165	AND	1140	2
1166	AND	1145	2
1167	AND	1150	2
1168	AND	1155	2
1169	AND	1160	2
1170	NAND	11074	

1171	AND	2	11074
1172	AND	2	1170
1173	AND	4	3
1174	AND	5	3
1175	AND	6	3
1176	AND	7	3
1177	AND	8	3
1178	AND	9	3
1179	AND	10	3
1180	AND	11	3
1181	AND	12	3
1182	AND	13	3
1183	AND	14	3
1184	AND	15	3
1185	AND	16	3
1186	AND	17	3
1187	AND	18	3
1188	AND	19	3
1189	AND	20	3
1190	AND	21	3
1191	AND	22	3
1192	AND	23	3
1193	AND	24	3
1194	AND	25	3
1195	AND	26	3
1196	AND	27	3
1197	AND	28	3
1198	AND	29	3
1199	AND	30	3
1200	AND	31	3
1201	AND	32	3
1202	AND	33	3
1203	AND	34	3
1204	AND	35	3
1205	AND	36	3
1206	AND	37	3
1207	AND	38	3
1208	AND	39	3
1209	AND	40	3
1210	AND	41	3
1211	AND	42	3
1212	AND	43	3
1213	AND	44	3
1214	AND	45	3
1215	AND	46	3
1216	AND	47	3
1217	AND	48	3
1218	AND	49	3
1219	AND	50	3

1220	AND		51	3
1221	AND		52	3
1222	AND		53	3
1223	AND		54	3
1224	AND		55	3
1225	AND		56	3
1226	AND		57	3
1227	AND		58	3
1228	AND		59	3
1229	AND		60	3
1230	AND		61	3
1231	AND		62	3
1232	AND		63	3
1233	AND		64	3
1234	AND		65	3
1235	AND		66	3
1236	AND		3	1170
1237	AND		3	11074
1238	OR	_ 1	67	
1239	OR	_ 2	68	
1240	OR	_ 3	69	
1241	OR	_ 4	70	
1242	OR	_ 5	71	
1243	OR	_ 6	72	
1244	OR	_ 7	73	
1245	OR	_ 8	74	
1246	OR	_ 9	75	

_ INDICATES 9 EXTERNAL OUTPUTS.

REFERENCES

1. Fredrick J. Hill, Eltayeb Abuelyman, Huang Wei-Kang, and Shen Guo-Qiang, "a New Two Task Algorithm for Clock Mode Fault Simulation in Sequential Circuits" 25th Design Automation Conference June 1988.

2. Mohsenni Behbahani, A., "Ph D Dissertation, Department of Electrical and Computer Engineering ,University of Arizona,1984"

3. Patel, M. Y., "Master Thesis, Department of Electrical and Computer Engineering, University of Arizona, 1985".

4. Hill, F. J. ,petal., "Hardware Compilation from an RTL to a Storage Logic Array Target," IEEE Transactions on Computer Aided Design. Vol. CAD-3, July, 1984.

5. Hill, F. J., and Peterson, G.R., " Digital systems : Hardware Organization and Design, 3rd ed.,John Wiley and Sons, New York,1987.

6. Kohavi, Zvi , "Switching and finite automata theory," McGraw-Hill, Inc. Second edition 1978.

7. Donald R. Schertz, " A New Representation for Fault in Combinational Digital Circuits, " IEEE Transactions on Computers. Vol. c-21, NO.8, August 1972, pp. 858-866.

8. Douglas B. Armstrong " A Deductive Method for Simulating Faults in Logic Circuits," IEEE Transactions on Computers,Vol. c-21, NO.5, May 1972, pp. 464-471.

9. Roth, J. P., "Diagnosis of Automata Failures : A Calculus and a Method," IBM J. Res. Dev. 10: 278-281, 1966.

10. Roth, J. P., W. G. Bouricius, and P. R. Schneider, "Programmed Algorithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits," IEEE Transaction on Electron. Computer. EC-16(10):567- 580, 1967.

11. Seshu, S., "On an Improved Diagnosis Program," IEEE Transaction on Electronic Computers, Vol. EC-14, Feb. 1965, pp. 76-81.

12. Hong, S. J . , " Fault Simulation Strategy for Combinational Logic Networks," Processings of 8th Annual International Conference on Fault Tolerant Computing , 1978, pp. 96-99.

13. Mohsenni Behbahani, A., and Patel, M. Y., "SCIRTSS User's Manual," Ver. 4.0 , Dec. 1984.