

Lehigh University Lehigh Preserve

Theses and Dissertations

1-1-1983

A Testability Measure.

Stephen Louis Kessler

Follow this and additional works at: <http://preserve.lehigh.edu/etd>

 Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Kessler, Stephen Louis, "A Testability Measure." (1983). *Theses and Dissertations*. Paper 2344.

This Thesis is brought to you for free and open access by Lehigh Preserve. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Lehigh Preserve. For more information, please contact preserve@lehigh.edu.

A Testability Measure

By

Stephen Louis Kessler

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Electrical Engineering

Lehigh University

1983

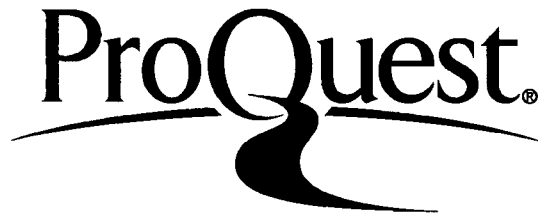
ProQuest Number: EP76620

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest EP76620

Published by ProQuest LLC (2015). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

Certificate of Approval

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

Dec 6, 1982

(date)

Professor in Charge

Chairman of Department

Acknowledgements

The author would like to extend his gratitude to Professor Alfred K. Susskind for introducing him to the subjects of design for testability and fault detection. This thesis is based upon ideas generated by Professor Susskind; it would never have been written without his invaluable help and inspiring direction. The author would also like to thank Debbie Hamann for typing the manuscript.

Table of Contents

	Page
Abstract	1
Chapter 1 TM Fundamentals	2
1.0 Introduction	2
1.1 Scope of the TM Algorithm	3
1.2 Algorithm Objectives	3
1.3 Definition of Terms	4
1.4 Overview of the TM	9
Chapter 2 Details of the TM Algorithm	22
2.0 Introduction	22
2.1 Controllability Calculations	22
2.1.1 Combinational Circuit Elements	24
2.1.2 Sequential Circuit Elements	31
2.1.3 Treatment of Circuits With Feedback Loops	36
2.2 Observability Calculations	37
2.2.1 Observability Calculations For Nodes With Fanout	41
2.2.2 Observability Calculations For Sequential Circuit Elements	45
2.2.3 Treatment of Circuits With Feedback Loops	48

	Page	
Chapter 3	Benchmark Calculations	49
	3.0 Introduction	49
	3.1 Benchmark Calculations For Combinational Circuits	49
	3.2 Benchmark Calculations For Sequential Circuits	52
Chapter 4	Algorithm Performance	63
	4.0 Introduction	63
	4.1 Exactness of the TM For Fanout Free Combinational Circuits	64
	4.2 Exactness of the TM For Feedback Free and Fanout Free Shift Register Circuits	74
	4.3 Observability of Stem Fanout Leads	80
	4.4 New TM Observability Calculations	84
	4.5 TM Calculations For Redundant Circuits	86
Chapter 5	Conclusions	88
Bibliography		92
Appendix		93
Vita		96

List of Figures

	Page	
1-1	Flow Chart of the TM Algorithm	10
1-2	Block Diagram of a Sequential Network With Feedback Loops	18
1-3	Two Bit Counter	20
2-1	Example Circuit	23
2-2	Two Input And Gate	24
2-3	n Input And Gate	26
2-4a	Proper Decomposition of a Four Input And Gate	27
2-4b	Improper Decomposition	27
2-5	Two Input Or Gate	28
2-6	n Input Or Gate	29
2-7	Inverter	30
2-8	RS Flip-Flop	31
2-9	Flip-Flop Example Circuit	33
2-10	D Flip-Flop	34
2-11	JK Flip-Flop	35
2-12a	m Input And Gate	38
2-12b	Gate Equivalent Circuit	38
2-13a	m Input Or Gate	39
2-13b	Gate Equivalent Circuit	40
2-14	A Typical Fanout Node	41

	Page	
2-15	A Node With Multiple Fanout	44
2-16	D Flip-Flop Equivalent Circuit	45
2-17	JK Flip-Flop Equivalent Circuit	46
2-18	Example Sequential Circuit	48
3-1	Two Input Exclusive-Or Network	50
3-2a	Example State Table	53
3-2b	Example Synchronizing Tree	54
3-3	Controllability Tree For Figure 3-2a	56
3-4	State Assignment For Figure 3-2a	57
3-5	Observability Tree For Figure 3-2a	60
4-1	$n \geq 2$, i-th Level And (Or) Gate	68
4-2	$n \geq 2$, i-th Level Decomposed Nand (Nor) Gate	70
4-3	D Flip-Flop N-Stage Shift Register	75
4-4	Fractional Observability Error For Combinational Stem Fanout Nodes	82
4-5	Two Input Exclusive-Or Cell	83
A-1	Circuit Example #1	93
A-2	Circuit Example #2	94

Abstract

This thesis presents the current status of an algorithm which is used to calculate how testable a digital circuit is. The algorithm, or testability measure, is easier than calculating the entire test set. The algorithm calculates controllability and observability figures for each and every node in a given combinational or sequential circuit. These figures are approximations to the actual amount of time, and fraction of total input combinations which are needed to control and observe a given circuit node. Algorithm results can be compared to benchmark figures to determine their accuracy. Testability measure results are shown to be exact for fan-out-free combinational circuits and feedback-free shift register circuits which are made using D flip-flops. Poor results are found to occur among the observability figures for stem fanout nodes, which showed up most noticeably in multiple level parity trees.

Chapter 1

TM Fundamentals

1.0 Introduction

In this thesis we present a testability measure, abbreviated TM. The testability measure is an algorithm which works from a digital circuit at the gate and flip-flop level to produce a metric for each lead, or node in the given circuit. Testability measure results can be used to determine how testable the given circuit is. A small result, or figure, indicates that a node is difficult to test, while conversely, easily tested nodes have large figures.

This chapter contains the background information that is necessary to be able to use the TM. The following section defines the applicability of the testability measure. The algorithm's objectives are discussed in the third section, Section 1.2. In the next section we define the terms that are used in this thesis. The last section contains an overview of the algorithm. A flow chart is included to add clarity to the discussion.

The remaining chapters contain details of the TM calculations and the performance of the algorithm. Chapter 2 presents details of the algorithm calculations. In Chapter 3 we show how to calculate exact testability

figures and thus judge their accuracy. The TM's major strong and weak points are discussed in Chapter 4. The last chapter, Chapter 5, contains our concluding remarks on the algorithm and our ideas concerning future research.

1.1 Scope of the TM Algorithm

TM calculations are performed on digital circuits. The permitted class of circuits includes combinational circuits and clocked sequential logic networks. The algorithm has been formulated to operate only on circuits with a single output, so that multiple outputs must be treated as an array of single outputs. Redundant networks (circuits which contain excessive logic) and asynchronous circuits are not included in the permitted class of circuits. The permitted combinational logic gates include And, Or, Nand, Nor gates and inverters. Exclusive - Or gates must be broken down into a more basic form. D flip-flops and JK flip-flops are the permitted sequential logic elements. For SR flip-flops our algorithms are incomplete.

1.2 Algorithm Objectives

The objective of this thesis is to formulate a testability measure that is indicative of testing

difficulty. The algorithm must show which portions of a given circuit are hard to test, and which are easy to test. It also must be easier to compute than finding the entire test set. If this were not true, then there would be no advantage in using the TM. And finally, we should be able to compare the testability measure's results to a rigorous measure's results. Thus we want to create a measure which is easy to calculate and has results that are meaningful.

A primary feature of our TM is that the results have meaning. They are approximations to exact results in purely combinational networks. In sequential circuits they are approximations to benchmark calculations. The benchmark results, while not exact, do indicate how testable a circuit is.

1.3 Definition of Terms

The TM requires that all combinational logic be level organized. Level organized circuits are set up in the following manner. All primary inputs (PI) will be placed at the left-hand side of the circuit. The first level, the left-most set of gates, have as inputs only PI's and complemented PI's. The next level should have as inputs the outputs of the first-level gates,

complements of the first level outputs and only complemented or uncomplemented PI's. A gate, G_1 , cannot be on the same level with a gate, G_2 , if the output of G_2 is used as an input to G_1 . Gates can have as inputs only gate outputs of the previous level(s) and possibly PI's. Thus a gate at level i must have at least one input from a gate output at level $i-1$ and can have other inputs which are either PI's or outputs of gates in level 1, 2, ..., $i-2$.

Each possible primary input combination is called a vector. If there are N PI's, then there are 2^N distinct vectors. The term vector is also associated with sequential circuits. A state vector in a sequential circuit is the set of bits that make up the coding of a state. A circuit with M flip-flops has 2^M different state vectors.

The TM generates two figures for each node in a circuit. One of these is the controllability figure, a concept first developed by Goldstein (see Reference (2)). Each node has two controllability figures, the one-controllability and the zero-controllability. The one-controllability describes the ease, or difficulty, of setting a node to a one. The one-controllability of node x is denoted by C_x^1 . The zero-controllability of node x , denoted C_x^0 , describes the difficulty of setting

node x to a zero. Control of a lead to a one (zero) is dependent on the fraction of the total number of vectors which set the lead to a one (zero) and on the amount of time that must pass before the node is actually set. To describe these factors the one (zero)-controllability is split into the fractional one (zero)-controllability and the one (zero) time frame number respectively. The time frame number, abbreviated TFN, denotes the number of clock periods, or time frames, which are needed to control a node to a one (zero). The TFN is taken from Kovijanic[4]. In a purely combinational network, for example, the one (zero)-TFN is equal to zero for all nodes because the circuit is unclocked (gate delays are ignored). In Eq. (1-1) we write the one (zero)-controllability of node x as a two-tuple

$$C_x^{1(0)} = \{A, B\} \quad (1-1)$$

where A is the fractional one (zero)-controllability and B is the one (zero)-TFN. The fractional one- and zero-controllabilities are restricted to the range [0,1]; thus in Eq. (1-1) we have

$$0 \leq A \leq 1. \quad (1-2)$$

We must ensure that the fractional controllabilities never exceed these bounds. Any results which are out of

bounds are forced back into the permitted range by using Eq. (1-3).

$$\text{If } A > 1, \text{ then } A = 1.0 \quad (1-3)$$

$$\text{If } A < 0, \text{ then } A = 0.0$$

Another property of the fractional one- and zero- controllabilities is that in the same time frame they sum to one for any node, x , in a network. Using Eq. (1-1) we have

$$C_x^1 = \{A, B\}$$

$$C_x^0 = \{D, E\}$$

and

$$A + D = 1.0 \quad (1-4)$$

Frequently, we only discuss the one-controllability because the zero-controllability can easily be obtained from Eq. (1-4). The detailed discussion in Chapter 2 relies heavily on this equation.

The other TM figure is the observability. The observability of lead x , another concept formulated by Goldstein[2], is denoted by $OBS(x)$. Lead x is observable if a change of signal on x can result in a change at the primary output. Thus to observe a lead x we must "sensitize" a path from lead x to the primary

output. The fraction of the total number of vectors which sensitizes a path from the lead to the primary output is called the fractional observability. Fractional observabilities are always limited to the range [0,1]. If a value becomes greater (less) than one(zero) then it is immediately rounded off to one (zero). The time needed to propagate a signal change from a lead to the primary output is the second factor which influences the observability of a lead. Time is measured in units of clock periods or time frames. The amount of time required to observe a lead is referred to as the time-frame number or TFN for short. The observability of lead x, broken down into its two influencing factors, is written as a two-tuple in Eq. (1-5)

$$\text{OBS}(x) = \{F,G\} \quad (1-5)$$

where F = fractional observability of lead x

G = the TFN.

Overflow and underflow conditions are treated by using Eq. (1-6).

$$\begin{aligned} \text{For } F > 1 &\rightarrow F = 1.0 \\ F < 0 &\rightarrow F = 0.0 \end{aligned} \quad (1-6)$$

1.4 Overview of the TM

Figure 1-1 is a flow chart of the TM algorithm. It highlights the procedure and order of the TM calculations; details are contained in the next chapter. The controllability calculations are performed first and the network is processed level-by-level, proceeding from inputs to outputs. This is another idea which was first formulated by Goldstein in Reference [2]. For sequential circuits with feedback loops we iterate through the levels until the fractional controllability figures converge. Observability calculations are performed second, proceeding from output to input.* We do not iterate the OBS calculations. The remainder of this section is devoted to explaining selected portions of Fig. 1-1.

* see Goldstein, Reference [2].

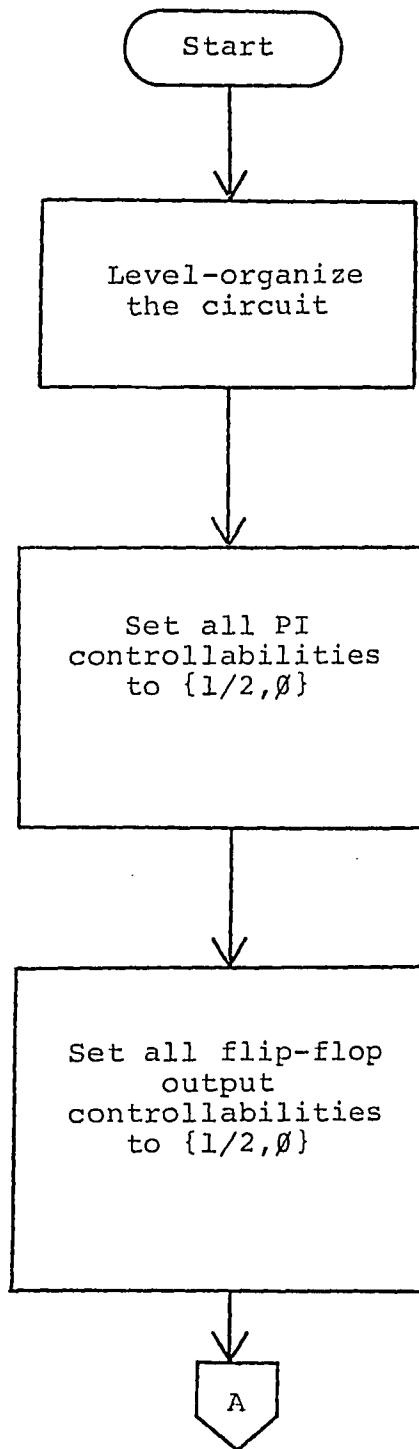
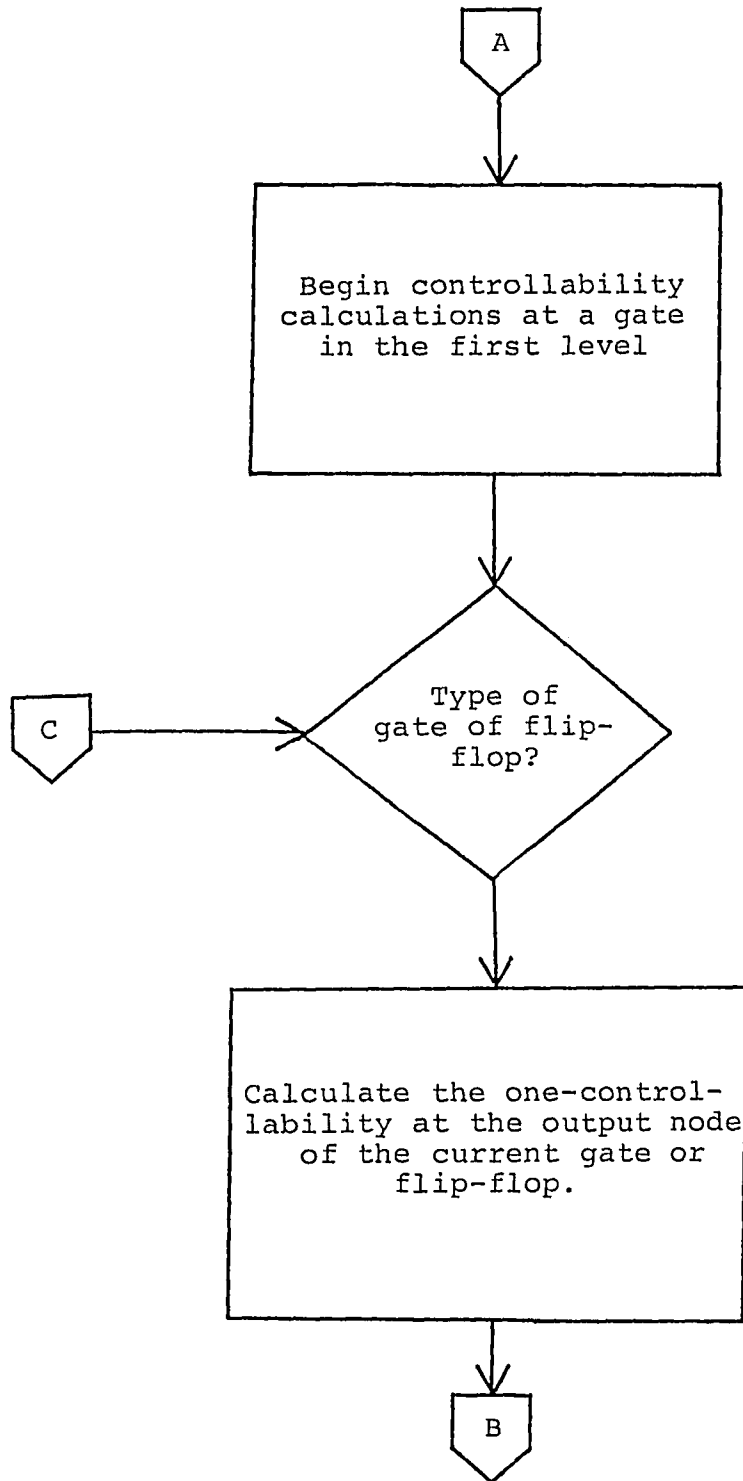
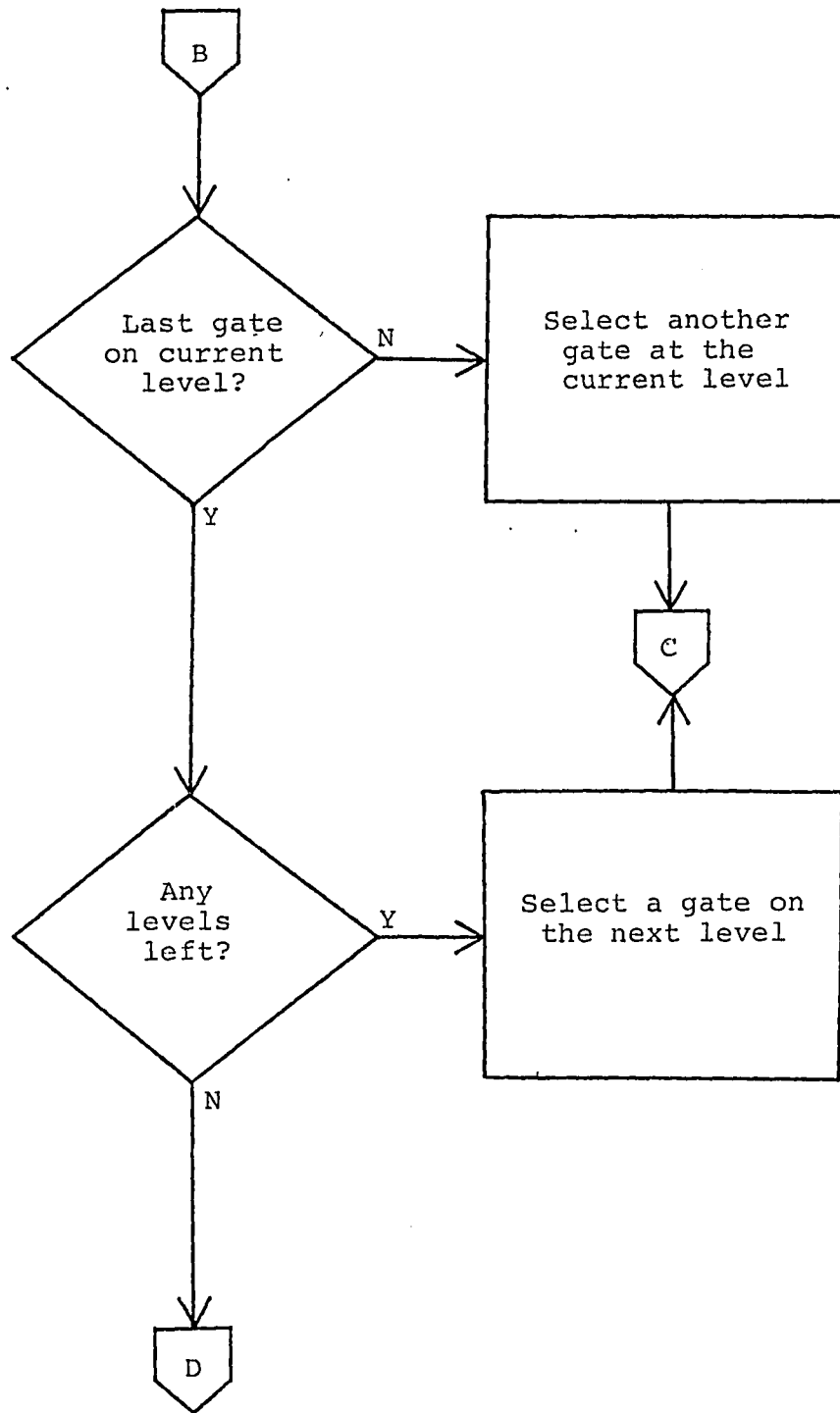
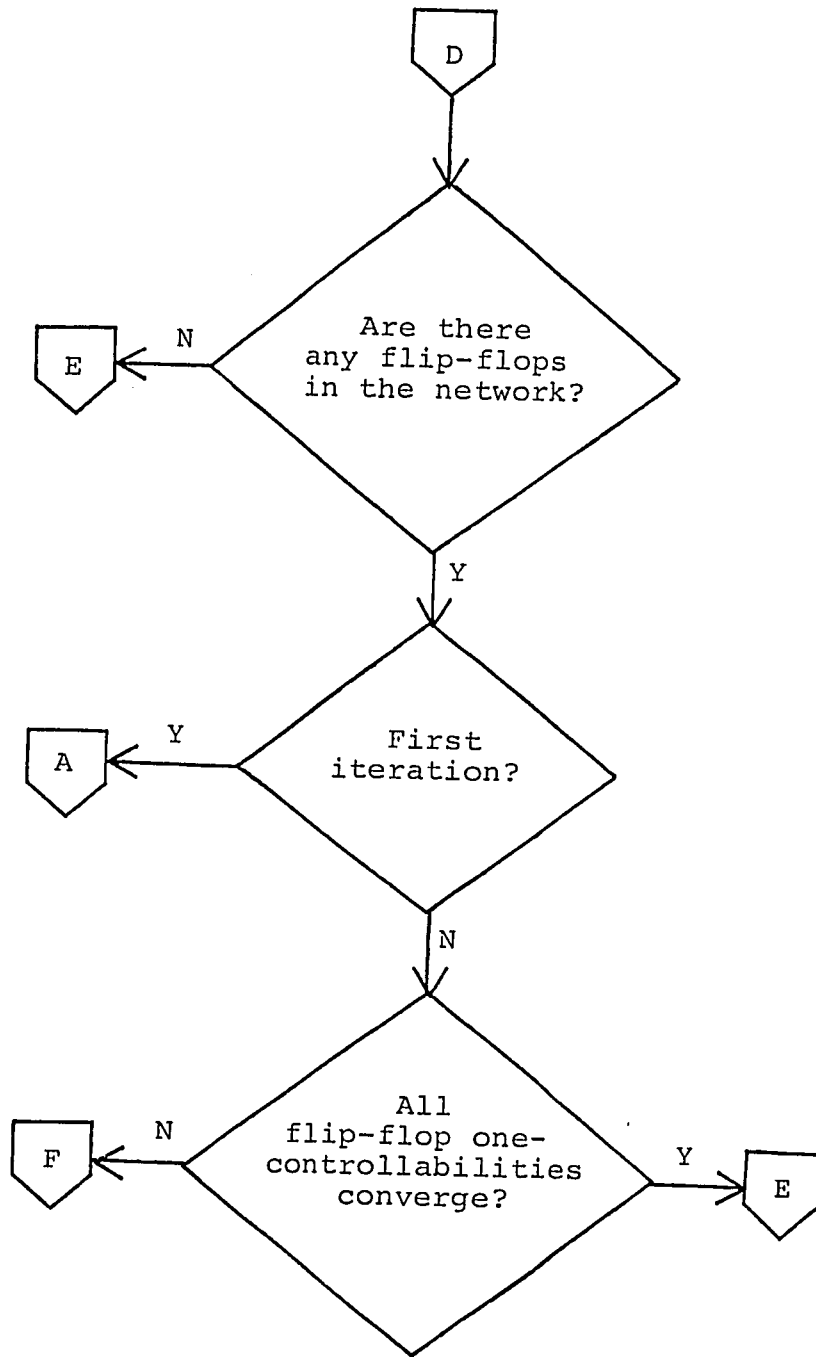
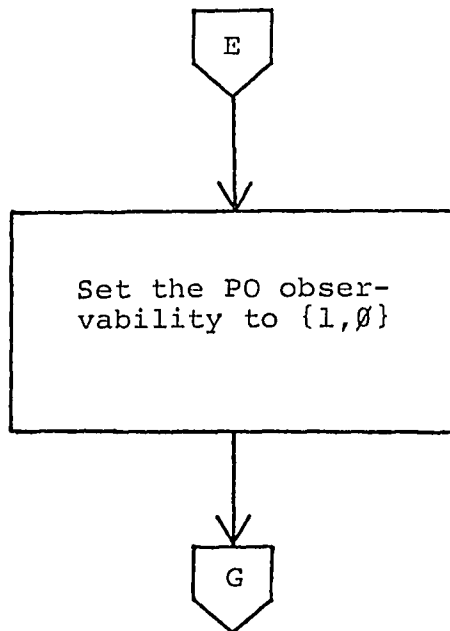
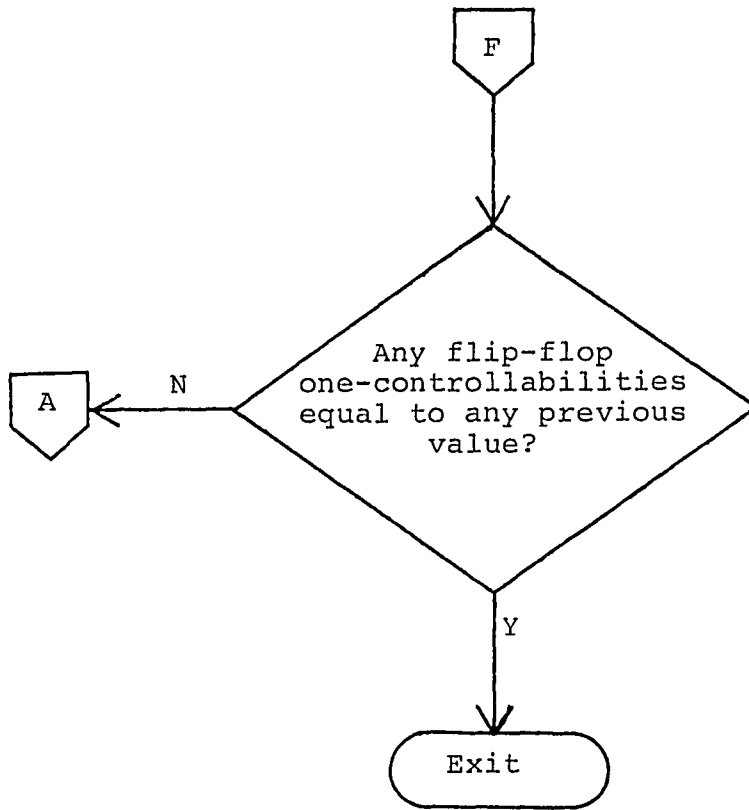


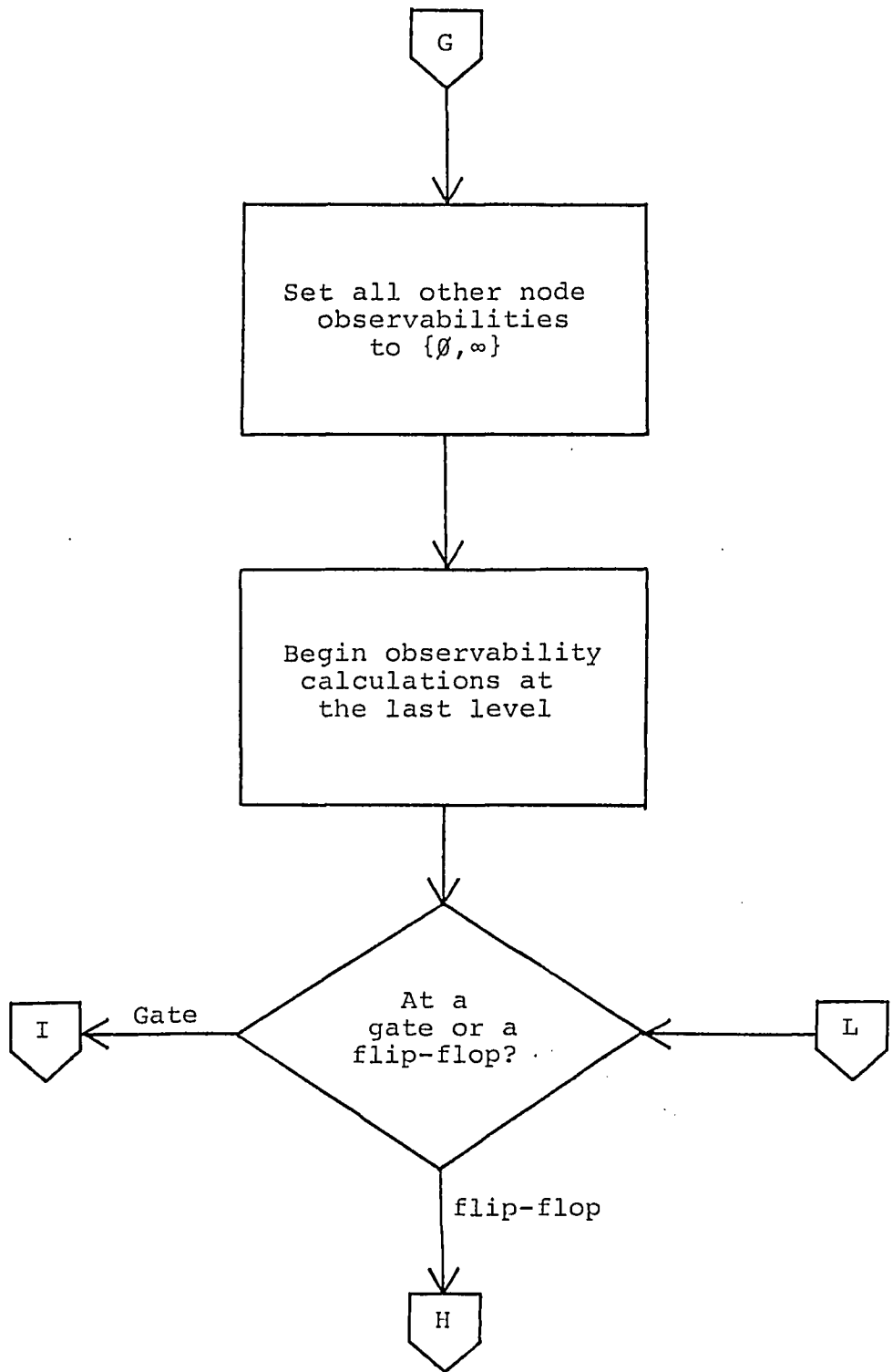
Figure 1-1 Flow Chart of the TM Algorithm.

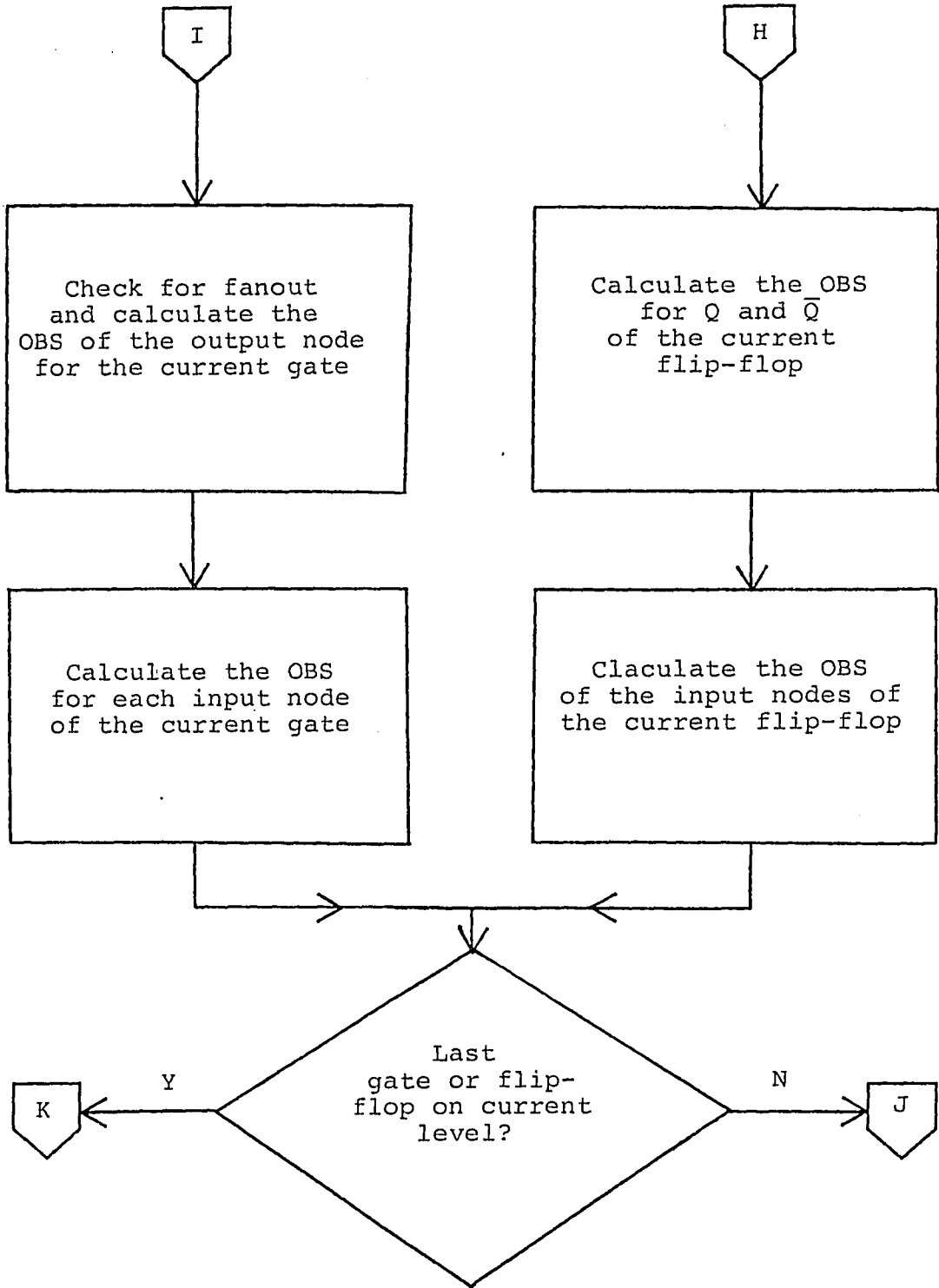


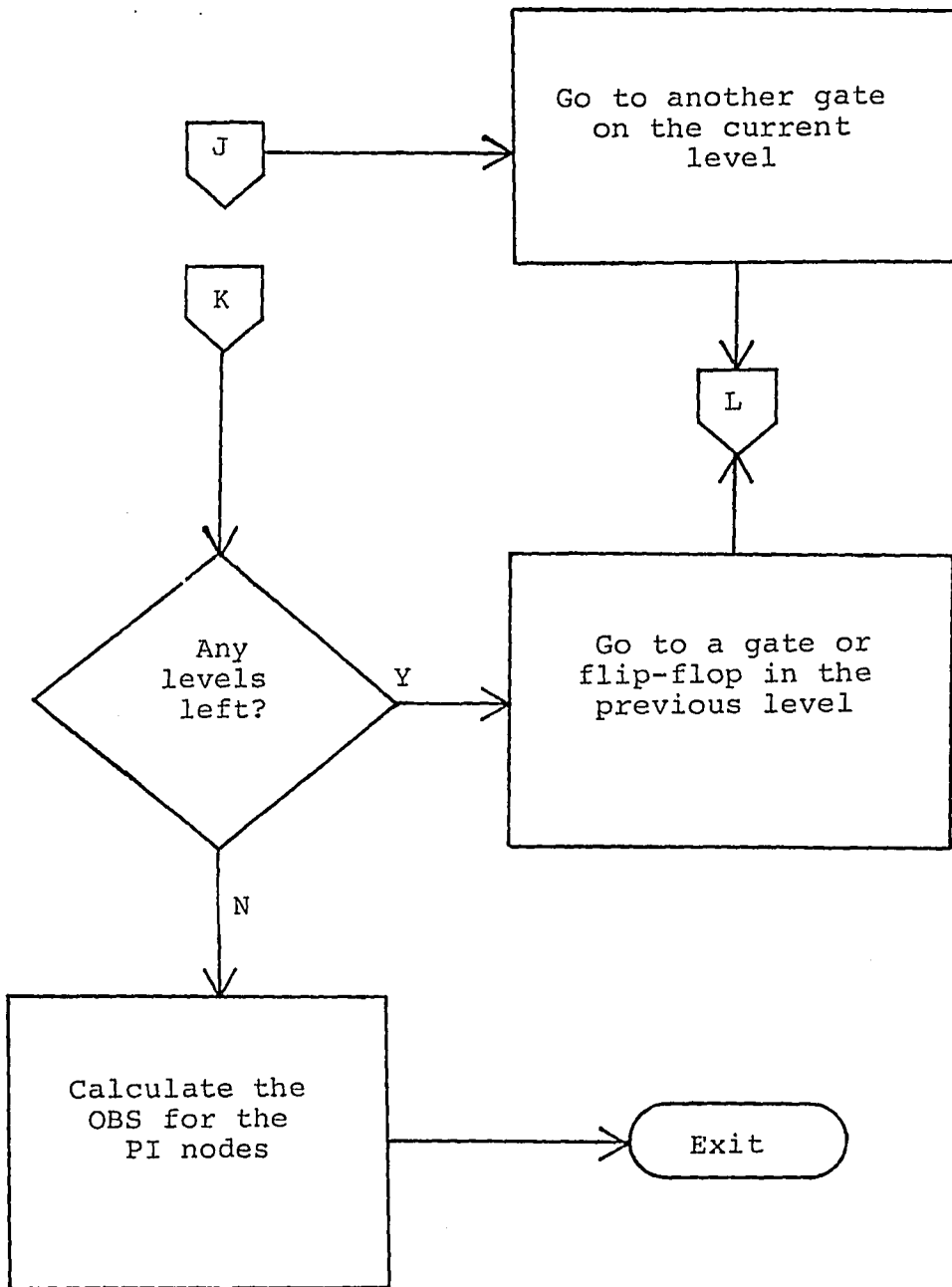












A sequential network is diagrammed in Fig. 1-2. The inputs to flip-flops 1 thru N are leads α thru ν , respectively. Leads 1 thru N are the respective flip-flop outputs. For circuits of this type we need to redefine

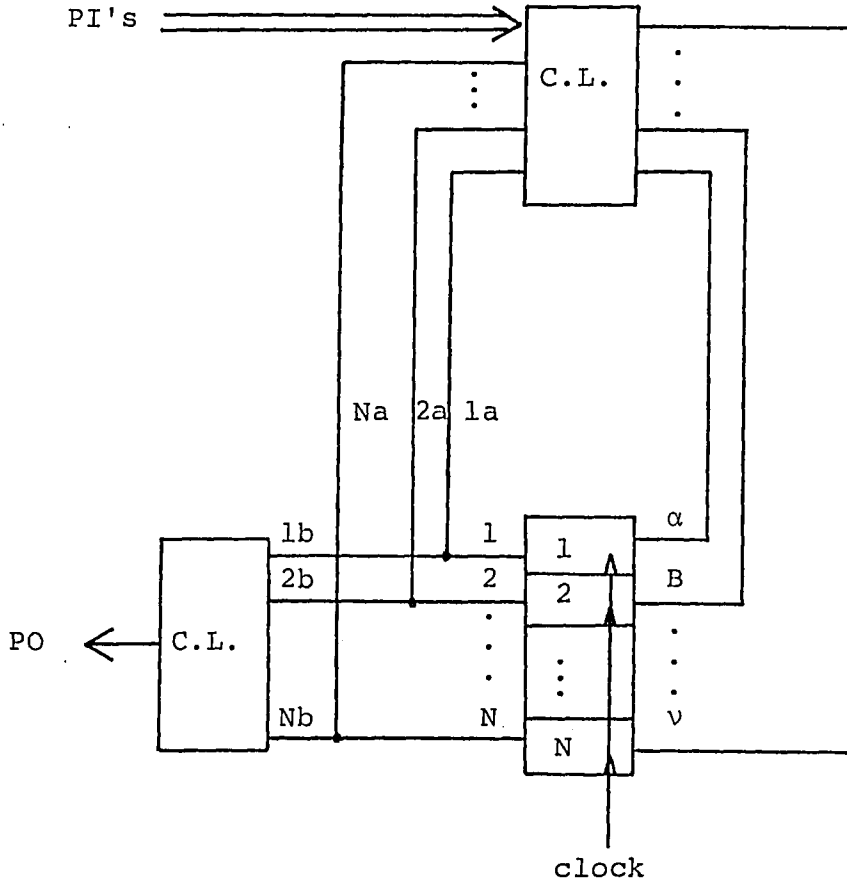


Figure 1-2 Block Diagram of a Sequential Network With Feedback Loops.

the initial level. This will not eliminate the feedback loops, but it will show us how to process this type of network. The initial or first level has as inputs only primary inputs or flip-flop outputs. It does not have inputs from other gate outputs. The higher levels are defined the same as before. As an example consider the two-bit counter with carry output in Fig. 1-3. The gates with outputs 2 thru 5 and 12 are at level one, the gates with outputs 6 and 7 are at level two, and the flip-flops are at level three. Once the flip-flop one (zero)-controllabilities are initialized, this scheme of level organizing allows us to process the network without running into any undefined values. Within each level the order of processing the nodes is arbitrary because order does not alter the controllability figures. The order we have chosen in our calculations is from top to bottom.

As stated previously, we loop thru the calculations level by level until the fractional one (zero)-controllability figures for all flip-flop output nodes converge. Equation (1-7) is the formula used to determine convergence of the fractional one (zero)-controllability, where i is set equal to the current iteration number.

$$|c_Q^{1(o)}(i) - c_Q^{1(o)}(i-1)| \leq \delta \quad (1-7)$$

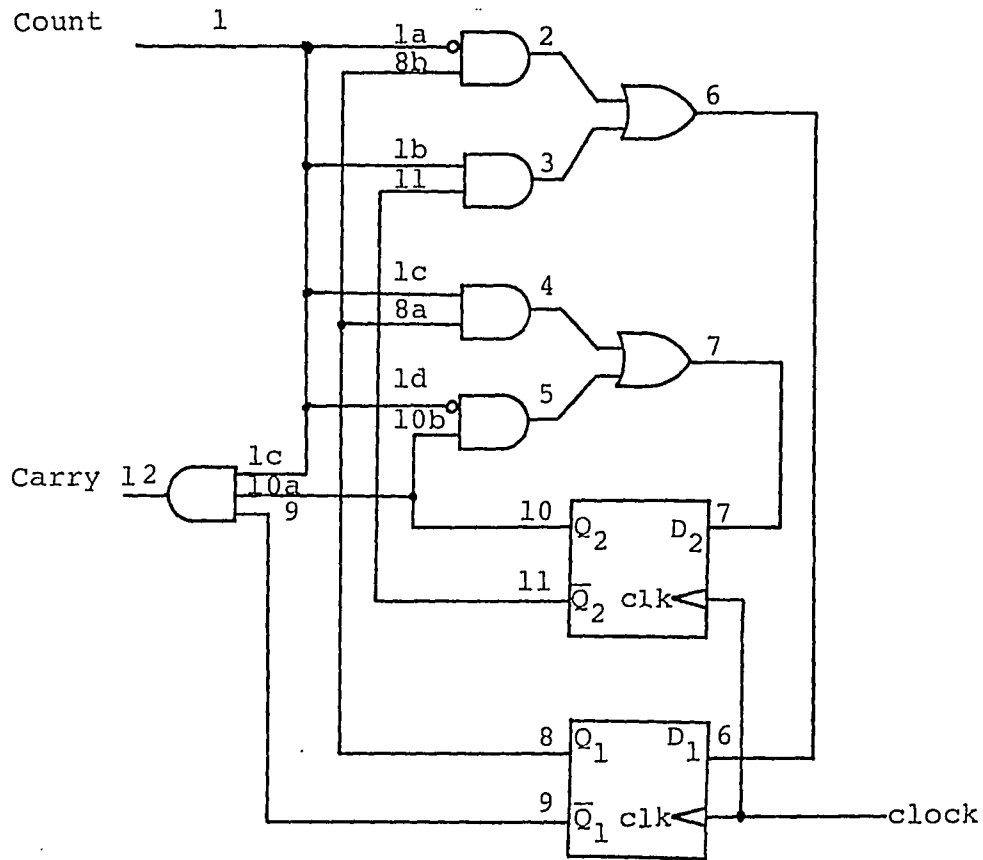


Figure 1-3 Two-Bit Counter.

The convergence factor, δ , is chosen here as $0.05 * 1/2^n$ where n equals the number of PI's in the circuit.

The first part of the flow chart in Fig. 1-1, the portion before offpage connector E, shows how the fractional one (zero)-controllabilities are performed. This part of the flow chart also describes TFN calculations for feedback-free sequential networks, i.e., networks

with finite memory span. TFN calculations for circuits with feedback are explained in more detail in Section 2.1.3.

At the end of the controllability calculations for sequential networks a check is made for oscillating results. If the controllability figures are found to oscillate the algorithm is immediately exited.

Chapter 2

Details of the TM Algorithm

2.0 Introduction

In this chapter the details of the testability measure (TM) algorithm will be presented. The controllability calculations for combinational and clocked sequential circuits will be explained first. Then we will explain the observability calculations for combinational and clocked sequential circuits. Illustrative examples are provided within this chapter to help clarify the TM calculations.

2.1 Controllability Calculations

To perform the TM controllability calculations we need to form lead variable lists for all the leads in the circuit. Each list is a set which has as elements the independent variables and/or flip-flop output variables. The lead variable list for lead j , L_j , contains independent variable x_i^* and/or flip-flop variable Q_i^* if the logic function for lead j depends on x_i^* and/or Q_i^* . If the inversion parity between lead j and x_i^* (or Q_i^*) is even, then $x_i^* = x_i$ ($Q_i^* = Q_i$); if the inversion parity is odd, then $x_i^* = \bar{x}_i$ ($Q_i^* = \bar{Q}_i$). Note that it is possible for L_j to contain both x_i and \bar{x}_i (or Q_i and \bar{Q}_i) if

multiple paths (with even and odd inversion parity) exist between the primary input x_i (or flip-flop output Q_i) and lead j .

In the course of the calculations pairs of lead-variable lists are compared and categorized. To compare a pair of lists, L_i and L_j , first apply list reduction to the lead-variable lists. The reduced lists, L'_i and L'_j , are obtained by deleting all variables which appear in both L_i and L_j in complemented and uncomplemented form.

As an illustration of how to form the lead-variable lists, consider the following example circuit.

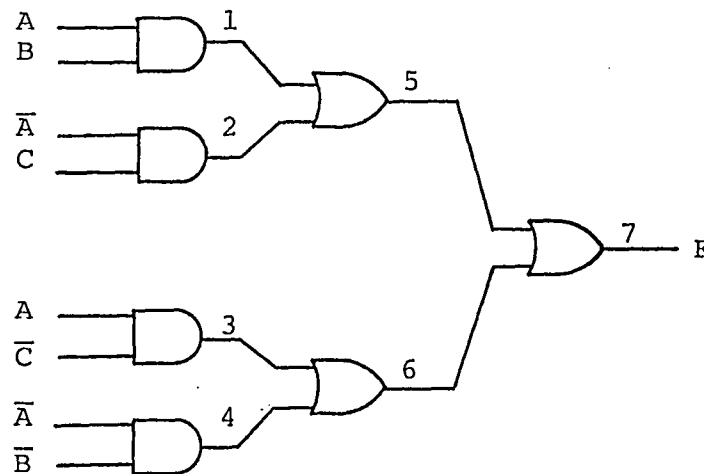


Figure 2-1 Example Circuit.

The lead-variable lists are:

$$\begin{aligned}L_1 &= \{A, B\} & L_5 &= L_1 \cup L_2 = \{A, \bar{A}, B, C\} \\L_2 &= \{\bar{A}, C\} & L_6 &= \{A, \bar{A}, \bar{B}, \bar{C}\} \\L_3 &= \{A, \bar{C}\} & L_7 &= \{A, \bar{A}, B, \bar{B}, C, \bar{C}\} \\L_4 &= \{\bar{A}, \bar{B}\}\end{aligned}$$

List reduction can be performed on the lists for the inputs of gate 7*. Thus we obtain:

$$L_5' = \{B, C\} \text{ and } L_6' = \{\bar{B}, \bar{C}\}$$

where we deleted A and \bar{A} from both lists. Then we get

$$L_7' = \{B, \bar{B}, C, \bar{C}\}.$$

Note that it is possible to obtain reduced lists that are empty.

2.1.1 Combinational Circuit Elements

For the two input And gate shown in Fig. 2-2



Figure 2-2 Two Input And Gate.

* Gates are referenced by referring to the gate's output node number.

where $C_1^1 = \{f, F\}$; $C_1^0 = \{1-f, F\}$

$C_2^1 = \{g, G\}$; $C_2^0 = \{1-g, G\}$

we have $C_3^1 = \{h, \max(F, G)\}$ (2-1)

and $C_3^0 = \{1-h, \min(F, G)\}$ (2-2)

The fractional one-controllability of lead 3, h, is determined by comparing L_1^1 and L_2^1 and using the appropriate formula. If:

- A. L_1^1 and L_2^1 are independent (have no common elements), then

$$h = f * G \quad (2-3)$$

- B. $L_1^1(L_2^1)$ is a subset of $L_2^1(L_1^1)$, then

$$h = \min(f, g) \quad (2-4)$$

- C. The elements of $L_1^1(L_2^1)$ are the complements of those in $L_2^1(L_1^1)$, then

$$h = f + g - 1 \quad (2-5)$$

- D. If L_1^1 and L_2^1 do not simply fall into one of the above cases, then each of the partially applicable formulas is used and the results are averaged.

When an And gate has three or more inputs:

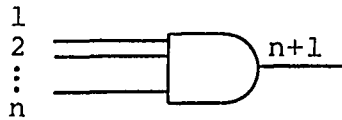


Figure 2-3 n Input And Gate.

where $C_1^1 = \{a, A\}$; $C_1^0 = \{1-a, A\}$

$C_2^1 = \{b, B\}$; $C_2^0 = \{1-b, B\}$

⋮ ; ⋮

$C_n^1 = \{n, N\}$; $C_n^0 = \{1-n, N\}$

we have $C_{n+1}^1 = \{h, \max(A, B, \dots, N)\}$ (2-6)

and $C_{n+1}^0 = \{1-h, \min(A, B, \dots, N)\}$. (2-7)

To find h use associativity to decompose this And gate into a tree using as few two input And gates as possible. The tree should also contain as few levels as possible. A tree containing a minimum number of gates and levels is called a minimum level tree. Thus to decompose a four input And gate we use the circuit shown in Fig. 2-4a

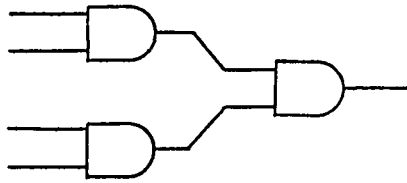


Figure 2-4a Proper Decomposition of a Four
Input And Gate.

and not the circuit in Fig. 2-4b.

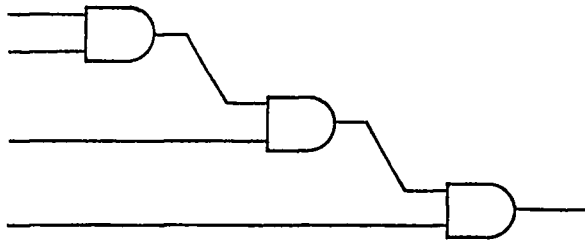


Figure 2-4b Improper Decomposition.

We arrange the decomposition in such a manner that as many gates as possible have their inputs such that one of the above conditions (i.e., A,B or C) is fully met. In the case of independence decomposition is not necessary since formula (2-3) can be easily extended to

$$h = a * B * C * \dots * n \quad (2-3a)$$

For a two input Or gate:



Figure 2-5 Two Input Or Gate.

where $C_1^1 = \{f, F\}$; $C_1^0 = \{1-f, F\}$

$C_2^1 = \{g, G\}$; $C_2^0 = \{1-g, G\}$

we have $C_3^1 = \{h, \min(F, G)\}$ (2-8)

and $C_3^0 = \{1-h, \max(F, G)\}$ (2-9)

The fractional one-controllability of lead 3, h , is determined by comparing L_1^1 and L_2^1 and using the appropriate formula. If:

A. L_1^1 and L_2^1 are independent, then

$$h = g + f - f * g \quad (2-10)$$

B. subset, then

$$h = \max(f, g) \quad (2-11)$$

C. complement, then

$$h = f + g \quad (2-12)$$

- D. If L_1^1 and L_2^1 do not simply fall into one of the above cases, then take the average of the results of each of the partially applicable formulas.

When an Or gate has three or more inputs:

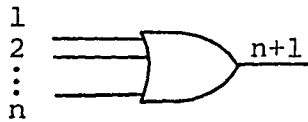


Figure 2-6 n Input Or gate.

where $C_1^1 = \{a, A\}$; $C_1^0 = \{1-a, A\}$

$C_2^1 = \{b, B\}$; $C_2^0 = \{1-b, B\}$

\vdots ; \vdots

$C_n^1 = \{n, N\}$; $C_n^0 = \{1-n, N\}$

we have $C_{n+1}^1 = \{h, \min(A, B, \dots, N)\}$ (2-13)

and $C_{n+1}^0 = \{1-h, \max(A, B, \dots, N)\}$ (2-14)

To find h decompose this Or gate into a minimum level tree of two input Or gates. We pair up the leads so that as many gates as possible have their inputs such that one of the above conditions is fully met. For the

case of independence, decomposition is not necessary since formula (2-10) can be easily extended. From (2-10) we have

$$1 - h = g + f - f * g = (1-f) * (1-g)$$

and for Fig. 2-6 this becomes

$$1 - h = (1-a) * (1-b) \dots *(1-n) \quad (2-10a)$$

The treatment of inverters is very straightforward.

For the inverter



Figure 2-7 Inverter.

where $C_1^1 = \{f, F\}$; $C_1^0 = \{1-f, F\}$

we have $C_2^1 = C_1^0$ and $C_2^0 = C_1^1$. (2-15)

The last two combinational circuit elements treated by our algorithm are Nand and Nor gates. The formulas for these are obtained by decomposing a Nand into an And gate and an inverter, and a Nor into an Or gate and an inverter. Thus we have:

$$C_{\text{Nand}}^1 \equiv C_{\text{And}}^0 \quad ; \quad C_{\text{Nand}}^0 \equiv C_{\text{And}}^1$$

$$C_{\text{Nor}}^1 \equiv C_{\text{Or}}^0 \quad ; \quad C_{\text{Nor}}^0 \equiv C_{\text{Or}}^1 .$$

2.1.2 Sequential Circuit Elements

Recall that when sequential circuit elements are present in a circuit we must iterate through the circuit until the fractional one-controllabilities converge. Thus it is obvious that the fractional controllability figures are iteration dependent. This means that at the next iteration, $i+1$, the one-controllability figure for node j , $C_j^1(i+1)$, will be a function of one or more one-controllability figures from the current iteration.

For the RS flip-flop shown in Fig. 2-8

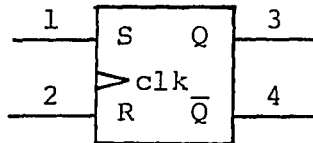


Figure 2-8 RS Flip-Flop.

we define the set controllabilities:

$N(i) \triangleq$ do nothing

$S(i) \triangleq$ set

$R(i) \triangleq$ reset

where $S(i) + R(i) + N(i) = 1$

and $C_1^1(i) = \{f(i), F\}$; $C_1^0(i) = \{1-f(i), F\}$ *

$C_2^1(i) = \{g(i), G\}$; $C_2^0(i) = \{1-g(i), G\}$.

$f(i)$ and $g(i)$ are the fractional one-controllabilities at the i -th iteration for leads 1 and 2 respectively. So we have

$$C_3^1(i+1) = C_4^0(i+1) = \{h(i+1), i+1\} \quad (2-16)$$

$$C_3^0(i+1) = C_4^1(i+1) = \{1-h(i+1), i+1\} \quad (2-17)$$

$$h(i+1) = S(i) + N(i) * h(i) \quad (2-18a)$$

and $1 - h(i+1) = R(i) + N(i) * (1-h(i))$. (2-18b)

$h(i+1)$ is the fractional one-controllability for the next iteration. In Eq. (2-18a) we can see that the flip-flop output fractional one-controllability figures for the next iteration, $i+1$, are a function of the current iteration figures. Thus we calculate the fractional one-controllabilities of the flip-flop outputs for the next iteration during the current iteration. Consider the following example.

* Note that all the TFN's are not iteration dependent. More will be said about the TFN's later on in this chapter.

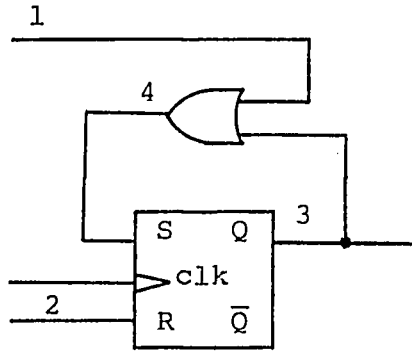


Figure 2-9 Flip-Flop Example Circuit.

During the initial iteration, $i=0$, $C_4^1(o)$ is a function of the initial condition of Q (i.e., $C_3^1(o)$). But to calculate $C_4^1(1)$ we need $C_3^1(1)$. In order to avoid this undefined condition we calculate $C_3^1(1)$ during the initial iteration, i.e., at $i=0$.

The variables $S(i)$, $R(i)$ and $N(i)$ are found by the following formulas which refer back to Fig. 2-8. If:

- A. leads 1 and 2 are primary inputs (abbreviated P.I.'s), then

$$N(i) = S(i) = R(i) = 1/3 \forall i \quad (2-19)$$

- B. leads 1 and 2 are not P.I.'s, then

$$S(i) = f(i) * (1-N(i))/(f(i) + g(i)) \quad (2-20)$$

$$R(i) = g(i) * (1-N(i))/(f(i) + g(i)) \quad (2-21)$$

and $N(i)$ is found by comparing the reduced lead variable lists. If L_1' and L_2' :

1. are independent, then

$$N(i) = (1-f(i)) * (1-g(i)) \quad (2-22)$$

2. are complements, then

$$N(i) = 0 \quad (2-23)$$

3. are subsets, then

$$N(i) = 1 - \max(f(i), g(i)) \quad (2-24)$$

4. do not fall simply into one of the above cases, then find $N(i)$ by taking the average of the results from each of the partially applicable formulas.

For the D flip-flop shown in Fig. 2-10

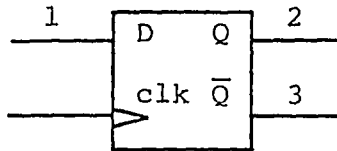


Figure 2-10 D Flip-Flop.

where $C_1^1(i) = \{f(i), F\}$; $C_1^0(i) = \{1-f(i), F\}$

we have $C_2^1(i+1) = C_3^0(i+1) = \{f(i), i+1\}$ (2-25)

and $C_2^0(i+1) = C_3^1(i+1) = \{1-f(i), i+1\}$.

For the JK flip-flop shown in Fig. 2-11

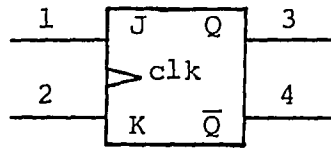


Figure 2-11 JK Flip-Flop.

where $C_1^1(i) = \{f(i), F\}$; $C_1^0(i) = \{1-f(i), F\}$

$C_2^1(i) = \{g(i), G\}$; $C_2^0(i) = \{1-g(i), G\}$

we have $C_3^1(i+1) = C_4^0(i+1) = \{h(i+1), i+1\}$ (2-27)

and $C_3^0(i+1) = C_4^1(i+1) = \{1-h(i+1), i+1\}$. (2-28)

To find $h(i+1)$ we compare the reduced lead variable lists for leads 1 and 2. If:

A. L_1^1 and L_2^1 are independent, then

$$h(i+1) = f(i) * [1-h(i)] + [1-g(i)] * h(i) \quad (2-29)$$

B. L_1^1 and L_2^1 are complements, then

$$h(i+1) = f(i) \quad (2-30)$$

C. L_1' and L_2' are subsets, then

$$h(i+1) = 1/2 * |f(i) - g(i)| + \min(f(i), g(i)) \\ + h(i) * (1-f(i)-g(i)) \quad (2-31)$$

D. lead labels 1 and 2 are equal (i.e., they come from the same fanout point), then

$$h(i+1) = f(i) + h(i) - 2 * f(i) * h(i) \quad (2-32)$$

E. L_1' and L_2' do not fall into one of the above classes, then take the average of the results obtained from each of the partially applicable formulas.

2.1.3 Treatment of Circuits With Feedback Loops

When a circuit contains feedback loops it will also contain flip-flops (because we do not consider asynchronous circuits), but the converse is not necessarily true. The fractional one-controllabilities are calculated by iterating through the circuit until all of the flip-flop fractional one-controllabilities converge. The convergence criterion has been stated in Chapter 1 (see Eq. (1-6)).

Calculation of the TFN is different from the fractional controllability calculations. The time frame number for a flip-flop output node is the iteration number. Thus each time we iterate through the network we increment the previous TFN to obtain the new TFN. We stop incrementing a flip-flop output node's TFN when the fractional controllability figure for this node converges, and in general, TFN computation continues until all flip-flop fractional controllability figures converge. Hence the final TFN values are dependent upon how and when the flip-flop fractional controllabilities converge.

2.2 Observability Calculations

We now proceed to describe the observability calculations for combination circuit elements. For a two input And gate (refer back to Fig. 2-2),

$$\text{where } C_1^1 = \{a, A\} \quad C_2^1 = \{b, B\}$$

$$\text{and } \text{OBS}(3) = \{c, C\}$$

$$\text{we have } \text{OBS}(1) = \{b * c, \max(B, C)\} \quad (2-33)$$

$$\text{and } \text{OBS}(2) = \{a * c, \max(A, C)\}. \quad (2-34)$$

For And gates with three or more inputs

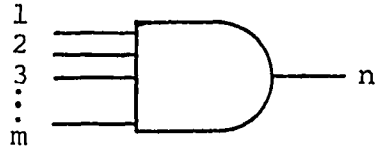


Figure 2-12a m Input And Gate.

we calculate the OBS(1) by forming the gate equivalent circuit of Fig. 2-12a as shown in Fig. 2-12b.

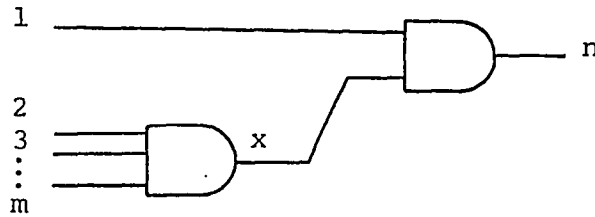


Figure 2-12b Gate Equivalent Circuit.

where $C_1^1 = \{a, A\}$ $OBS(n) = \{n, N\}$

$C_2^1 = \{b, B\}$

⋮

$C_m^1 = \{m, M\}$

$C_x^1 = \{x, X\}$; $X = \max(B, C, \dots, M)$

The fractional one-controllability of lead x, x, is found using the procedure for And gates in Section 2.1.1.

Thus we have

$$\text{OBS}(1) = \{x * n, \max(X,N)\}. \quad (2-35)$$

Note that if leads 2 thru m are independent, Eq. (2-35) becomes

$$\text{OBS}(1) = \{b * c * \dots * m * n, \max(X,N)\}. \quad (2-35a)$$

This procedure is repeated for input leads 2 thru m. For two-input Or gates (refer to Fig. 2-5),

where $C_1^O = \{1-a, A\}$ $\text{OBS}(3) = \{c, C\}$

and $C_2^O = \{1-b, B\}$

we have $\text{OBS}(1) = \{(1-b) * c, \max(B, C)\}$ (2-36)

and $\text{OBS}(2) = \{(1-a) * c, \max(A, C)\}$ (2-37)

For Or gates with three or more inputs we calculate

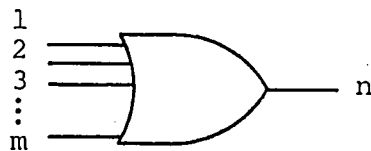


Figure 2-13a m Input Or Gate.

the OBS(1) by forming the gate equivalent circuit of Fig. 2-13a as shown in Fig. 2-13b.

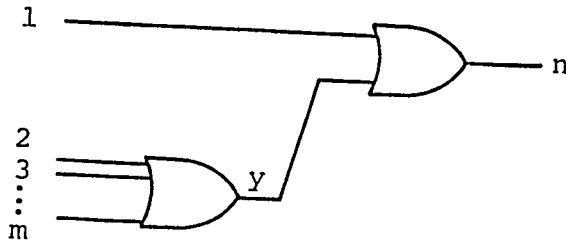


Figure 2-13b Gate Equivalent Circuit.

where

$$C_1^O = \{1-a, A\} \quad \text{OBS}(n) = \{n, N\}$$

$$C_2^O = \{1-b, B\}$$

$$\vdots$$

$$C_m^O = \{1-m, M\}$$

$$C_y^O = \{1-y, y\}, \quad y = \max(B, C, \dots, M)$$

The fractional one-controllability of lead y, y , is found using the procedure for Or gates in Section 2.1.1. Thus we have

$$\text{OBS}(1) = \{(1-y) * n, \max(y, N)\} \quad (2-38)$$

If leads 2 thru m are independent, Eq. (2-38) becomes

$$\text{OBS}(1) = \{(1-b) * (1-c) * \dots * (1-m) * n, \max(y, N)\}. \quad (2-38a)$$

This procedure is repeated for the input leads 2 thru m .

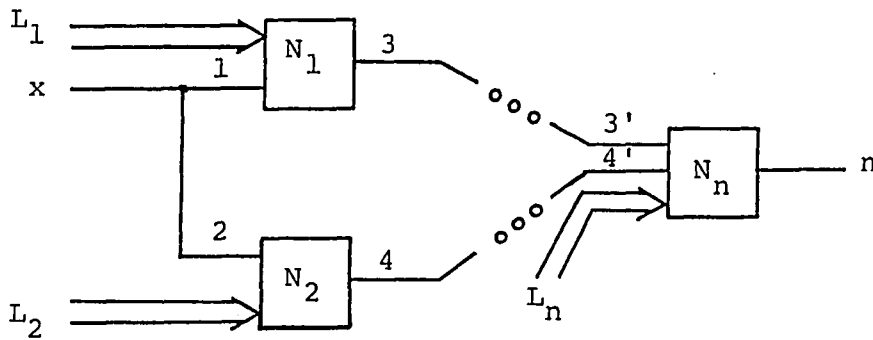
For an inverter (refer to Fig. 2-7),

where $OBS(2) = \{a, A\}$

we have $OBS(1) \equiv OBS(2)$ (2-39)

2.2.1 Observability Calculations for Nodes With Fanout

We first consider nodes with two fanout paths. In Fig. 2-14 lead x is the stem of the fanout, and leads 1 and 2 are on the fanout paths of x . N_n is the first gate or flip-flop at which the fanout reconverges.



$OBS(1) = \{f, F\}$; $OBS(2) = \{g, G\}$

Figure 2-14 A Typical Fanout Node.

To calculate $OBS(x)$, form the path-variable list for each path. A path-variable list for lead i , P_i , is the union of the lead variable lists that are associated with all the gates or flip-flops that lie on the path from lead i to the input of the gate at which the fanout

first reconverges. In Fig. 2-14 for P_2 the gate at which the fanout first reconverges is N_n . Note that P_2 does not include L_n , because it is at the point of fanout reconvergence. The path-variable list for the stem of the fanout, lead x , is

$$P_x = P_1 \cup P_2. \quad (2-40)$$

In the course of the observability calculations pairs of path-variable lists are compared and categorized. To compare a pair of lists, P_1 and P_2 , first form the reduced path-variable lists P'_1 and P'_2 . The reduced path-variable lists are obtained by deleting all elements that appear in barred and unbarred form in both lists. For example, let

$$P_1 = \{A, \bar{A}, B, \bar{C}, D, \bar{D}\} \quad ; \quad P_2 = \{A, \bar{A}, \bar{B}\}$$

Then the reduced path-variable lists are

$$P'_1 = \{B, \bar{C}, D, \bar{D}\} \quad ; \quad P'_2 = \{\bar{B}\}.$$

Note that it is possible to obtain

$$P'_q = P'_r = \{\emptyset\}.$$

Next we state the rules for calculating the observability of the stem of a fanout node, x . If the

inversion parity* of the signal on lead 1 (refer to Fig. 2-14) is the same as the inversion parity of the signal on lead 2, then

$$\text{OBS}(x) = \{f + g, 1/2 * (F+G)\}. \quad (2-41)$$

If the inversion parity of the signals does not satisfy the above condition, then three subcases are considered.

- A. If the elements in P'_1 that appear in one form, x_j^* , do not appear in their opposite form, \bar{x}_j^* , in P'_2 , then

$$\text{OBS}(x) = \{|f-g| \quad 1/2 * (F+G)\}. \quad (2-42)$$

- B. If the condition under A above does not hold or P'_1 and P'_2 have no common elements (including $P'_1 = P'_2 = \{\emptyset\}$), then

$$\text{OBS}(x) = \{f + g, 1/2 * (F+G)\}. \quad (2-43)$$

- C. If both conditions above are partially applicable to the elements in P'_1 and P'_2 , then

$$\text{OBS}(x) = \{\max(f,g), 1/2 * (F+G)\}. \quad (2-44)$$

* The inversion parity of a lead is determined from the point of fanout to the input of the gate at which the fanout first reconverges.

To extend the above rules to leads with multiple fanout nodes, i.e., three or more, consider the following figure.

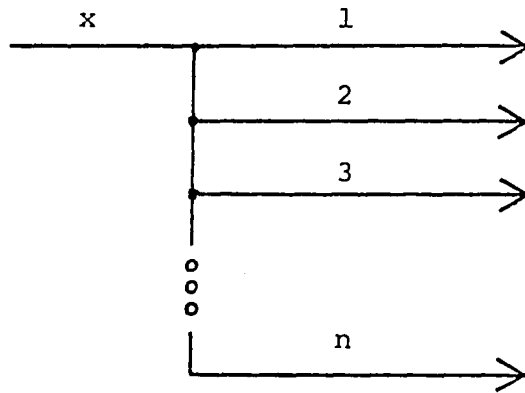


Figure 2-15 A Node With Multiple Fanout.

First group the paths into three sets:

Set a - leads with even inversion parity

Set b - leads with odd inversion parity

Set c - leads with both even and odd inversion parity

The observabilities for sets a and b are easily computed using the rule for nodes with the same inversion parity. To calculate the OBS(set c), form a pyramid (tree) of the nodes in this set. The nodes should be paired off so that the stated conditions, cases A and B on pp. 43 are fully met as often as possible. Next compute the observability of sets a and b combined. Denote this as

OBS(x'). This computation requires $P_{\text{set } a}$ and $P_{\text{set } b'}$ where the path-variable list for each set is the union of the path-variable lists of the leads in each particular set. Finally compute the OBS(x) using OBS(x'), OBS(set c), $P_{x'}$ and $P_{\text{set } c}$. $P_{x'}$ is the union of $P_{\text{set } a}$ and $P_{\text{set } b'}$.

2.2.2 Observability Calculations For Sequential Circuit Elements

The observability calculations for D flip-flops are stated with reference to the circuit in Fig. 2-16.

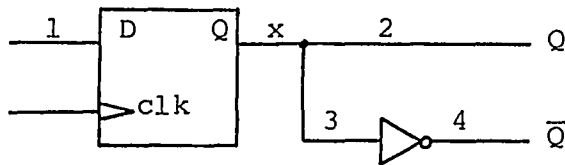


Figure 2-16 D Flip-Flop Equivalent Circuit.

$$\text{OBS}(Q) = \text{OBS}(2) = \{f, F\} ;$$

$$\text{OBS}(\bar{Q}) = \text{OBS}(3) = \{g, G\}$$

and we have

$$\text{OBS}(1) = \{h, 1/2 * (F+G) + 1\} \quad (2-45)$$

The fractional observability figure for lead x , h , is determined by using f , g and the rules for leads that have fanout, which were given in Section 2.2.1. If either node Q or \bar{Q} , leads 2 or 4 respectively, are unused, then the unused lead is unobservable. The unused lead is then ignored when calculating the observability of lead x . For example, if lead 2 is unused, then Eq. (2-45) becomes

$$\text{OBS}(1) = \{g, G+1\}. \quad (2-45a)$$

If lead 4 is unused we have

$$\text{OBS}(1) = \{f, F+1\}. \quad (2-45b)$$

JK flip-flop observability calculations* are stated with reference to the circuit in Fig. 2-17. Let

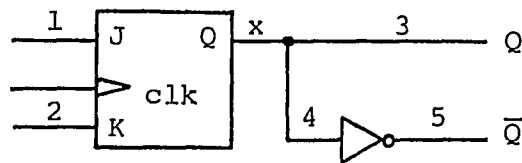


Figure 2-17 JK Flip-Flop Equivalent Circuit.

* Observability calculations were not determined for RS flip-flops. The JK flip-flop OBS equations were not verified via experimental calculations.

$$C_3^1 = \{e, E\} \quad ; \quad C_3^0 = \{1-e, E\}^*$$

$$\text{OBS}(3) = \{f, F\} \quad ; \quad \text{OBS}(4) = \{g, G\}$$

$$\text{and} \quad \text{OBS}(x) = \{h, H\} \quad ; \quad H = 1/2 * (F+G). \quad (2-46)$$

Note that h is calculated using the rules for fanout leads. If either node Q or \bar{Q} , leads 3 or 5 respectively, are unused, then the unused lead is unobservable. The unused lead is then ignored when calculating the observability of lead x and we have

$$\text{OBS}(x) = \{h, H\}$$

$$\text{where} \quad h = \begin{cases} f & \text{if lead 5 is unused} \\ g & \text{if lead 3 is unused} \end{cases}$$

$$\text{and} \quad H = \begin{cases} F & \text{if lead 5 is unused} \\ G & \text{if lead 3 is unused} \end{cases} \quad (2-46a)$$

The flip-flop input observabilities are

$$\text{OBS}(1) = \{(1-e) * h, \max(E, H) + 1\} \quad (2-47)$$

$$\text{and} \quad \text{OBS}(2) = \{e * h, \max(E, H) + 1\}. \quad (2-48)$$

The reasoning behind Eq. (2-47) (Eq. (2-48)) is very straightforward. When $Q^- = 0$ ($Q^- = 1$) a change in the

* These are the converged figures.

J(K) input is reflected in a change in the output, $Q^+(\bar{Q}^+)$, no matter what the state of input K(J).

2.2.3 Treatment of Circuits With Feedback Loops

The presence of feedback loops only affects the OBS calculations of flip-flop output nodes. For each flip-flop we determine which nodes or fanout paths are in feedback loops. Those nodes which are found to lie in such loops are ignored when calculating the observability of a flip-flop. Thus in Fig. 2-18 lead

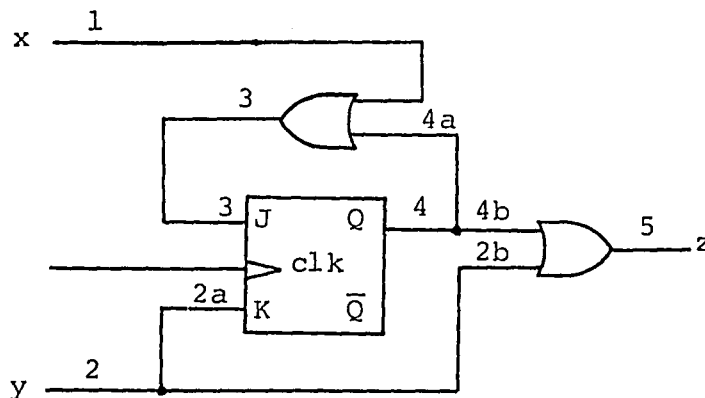


Figure 2-18 Example Sequential Circuit.

4a (which is on a fanout path) is in a feedback loop; hence $OBS(4) = OBS(4b)$. Note that \bar{Q} is unused and hence, it too is ignored when calculating the observability of leads 3 and 2a.

Chapter 3

Benchmark Calculations

3.0 Introduction

This chapter describes how to do the benchmark calculations. These calculations generate figures with which we may compare our TM results, and thus determine their accuracy. The calculations are split into two major groups; those for purely combinational circuits and those for sequential circuits. Within each group there are separate calculations for controllability and observability. Example calculations are included to help clarify the concepts.

3.1 Benchmark Calculations For Combinational Circuits

The controllability benchmark calculations are very straightforward. At each lead, i , determine the function, F_i , in terms of the primary inputs. The fractional one-controllability for lead i is the number of solutions to $F_i = 1$ divided by 2^N , where N is the number of primary inputs. The fractional zero-controllability is the number of solutions to $F_i = 0$ divided by 2^N . Note that the fractional zero-controllability does not need to be calculated, since 2^N minus the number of

solutions to $F_i = 1$ divided by 2^N yields the desired figure and is a much simpler computation. Consider the circuit shown in Fig. 3-1.

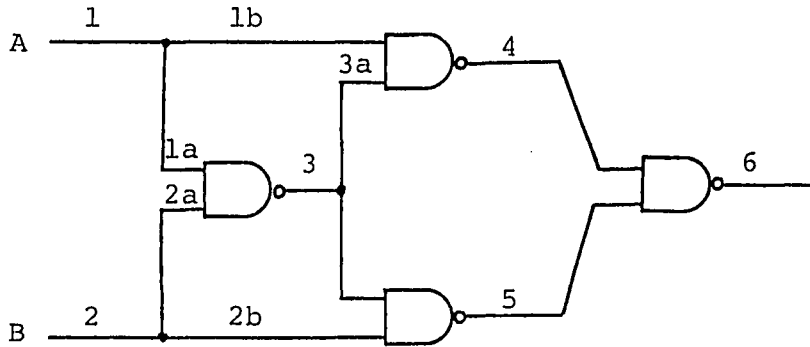


Figure 3-1 Two Input Exclusive-Or Network.

We wish to determine the benchmark fractional one-controllability figure for lead 4. Thus

$$F_4 = \overline{A \cdot \overline{AB}} = \overline{A} + AB = \overline{A} + B = 1$$

which yields 3 solutions. The desired figure is

$$3/2^2 = 3/4.$$

The controllability TFN's for all leads in a combinational circuit are all zero by definition, since no clock pulses are needed to control any leads in the circuit.

We use the Boolean difference (abbreviated BD) to calculate the fractional observability benchmark figures. Given a function f of x_1, x_2, \dots, x_n the BD is defined as

$$df/dx_j = f(x_1, x_2, \dots, x_j=0, \dots, x_n) \oplus f(x_1, x_2, \dots, x_j=1, \dots, x_n).$$

The BD is the ring sum of the function with $x_j = 0$ and $x_j = 1$. The number of solutions to

$$df/dx_j = 1$$

divided by 2^N is the fractional observability of lead x_j . Due to the nature of the ring sum, df/dx_j , equals one if and only if the function has different values for $f(x_j=0)$ and $f(x_j=1)$. Thus if lead x_j switches values this action will be observed at the output of the circuit. The observability TFN's for all leads in a combinational circuit are all zero, since no clock pulses are needed to observe any leads in the circuit. Let us calculate the fractional observability of lead 3a in Fig. 3-1. In terms of this lead the function is

$$F = \overline{(Ax_{3a})} \cdot \overline{(B \cdot \overline{AB})}$$

$$= Ax_{3a} + \overline{AB}$$

Thus

$$dF/dx_j = \bar{A}B \oplus (\bar{A}B+A) = A$$

and the desired figure is

$$2/2^2 = 1/2.$$

3.2 Benchmark Calculations For Sequential Circuits

The controllability benchmark calculations are performed using a modified synchronizing tree. We will first define the synchronizing sequence and then show how to find the synchronizing tree for a given machine. Then we will show the modifications needed to find the fractional one-controllability figures.

"A synchronizing sequence of a machine, M, is a sequence which takes M to a specified final state, regardless of the output or the initial state."¹ Not all machines possess such a sequence.

A synchronizing tree is constructed for a given machine by ignoring the outputs and by, at the j-th level in the tree, listing the state ambiguity at each node which results after the first j inputs.² For a machine with N input leads, i.e., N PI's, the j-th level will contain at most 2^{Nj} nodes. The state ambiguity at

1 Zvi Kohavi, Switching and Finite Automata Theory (New York: McGraw-Hill Book Company, 1978), p. 456.

2 Ibid.

each node does not contain repeated entries. The initial ambiguity contains all the states of the given machine. A node becomes terminal if it contains a single entry or if it is a repetition of an ambiguity for a node at some preceding level.³ The synchronizing tree for the state table⁴ in Fig. 3-2a is shown in Fig. 3-2b.

PS	NS, z	
	x=0	x=1
A	B, 0	D, 0
B	A, 0	B, 0
C	D, 1	A, 0
D	D, 1	C, 0

Figure 3-2a Example State Table.

3 Ibid., 457.

4 Ibid., 455 and 457.

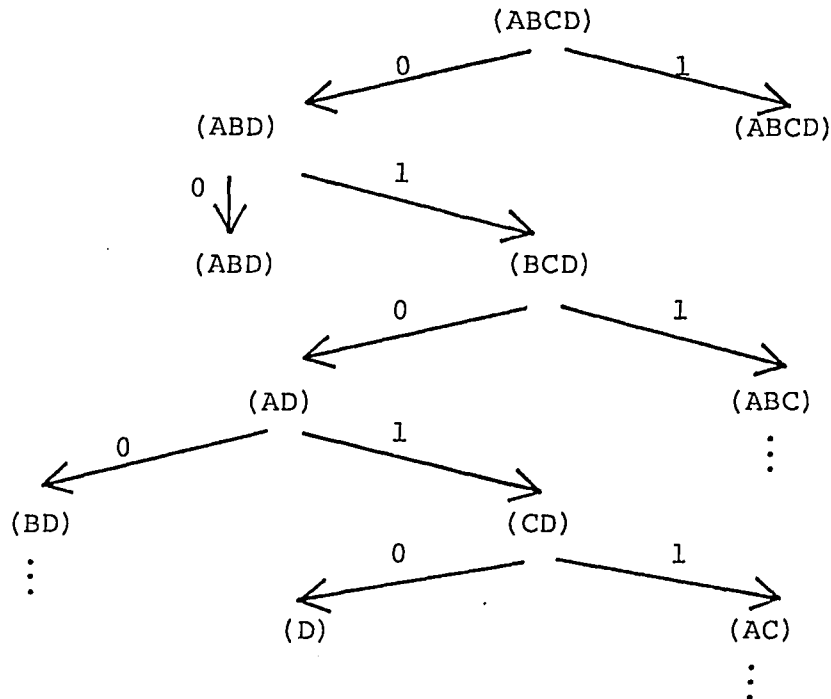


Figure 3-2b Example Synchronizing Tree.

Thus for this example one synchronizing sequence is 0 1 0 1 0 and D is the final state.

The modified synchronizing tree, or controllability tree, works from an expanded state table. To find the expanded state table list the state assignment for each state, noting which state variable corresponds to which flip-flop in the circuit. If a state has n multiple assignments, so that the circuit has n equivalent states, then list the n assignments separately in the new table. In other words, we split all the equivalent states and list them separately in the expanded state table. The

table must also include any transient states which are present in the given machine. Thus our expanded state table is actually the machine's transition table. So if the machine has q flip-flops, i.e., q state variables, then the expanded state table will have 2^q rows.

Next find the synchronizing tree for the expanded state table. We associate with each element in the initial ambiguity, a probability of starting in the corresponding row of the expanded state table. Since we assume an equally likely starting condition, the assigned probability for each element is $1/2^q$. As nodes are added to the tree we calculate the probability for each element in the state ambiguity, given that a specific input has occurred. For the state table shown in Fig. 3-2a (we assume it is in expanded form) we have the controllability tree of Fig. 3-3. Note that each level of this tree contains all the nodes which result from all possible input combinations. Thus for a machine with N PI's, the j -th level will contain 2^{Nj} nodes. Before we show how to terminate this tree we first describe the benchmark one-controllability figures for sequential circuits.

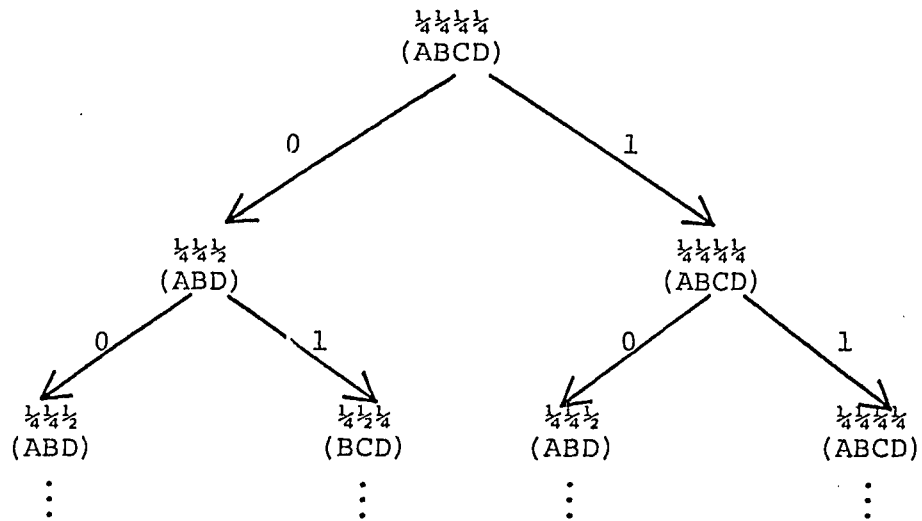


Figure 3-3 Controllability Tree For Figure 3-2a.

To calculate the fractional one-controllability figures for a flip-flop output terminal, Q_i , first determine the set of states that have state vectors such that the flip-flop in question has a one (zero) at its output. This set is called the one-controllability set for Q_i , denoted $CS(Q_i)$. Figure 3-4 shows the state assignment for the state table of Fig. 3-2a. Thus for Q_1 we have $CS(Q_1) = \{C,D\}$. At a given sequence length, i.e., tree level, add together the probabilities which correspond to members of the controllability set for Q_i . Then divide this total by the number of paths, or nodes, at that sequence length. By summing the probabilities at a given sequence length we are actually

State Assignment		PS
Q_1	Q_2	
0	0	A
0	1	B
1	1	C
1	0	D

Figure 3-4 State Assignment For Figure 3-2a.

calculating the probability that flip-flop output Q_i will be at logic 1. Division by the total number of paths assumes that each path, or input sequence, is disjoint and equally likely. The fractional one-controllabilities for Q_1 for sequences of length one and two are given by:

sequence length, $\ell = 1$

$$1/2 * [1/2 + 1/4 + 1/4] = 1/2$$

$\ell = 2$

$$1/4 * [1/2 + 1/2 + 1/4 + 1/2 + 1/4 + 1/4] = 9/16$$

Termination of the tree occurs when the fractional one-controllability figures for all Q_i converge. Although we have not proved that the figures will always

converge, intuition and all of our experiments to date have not disproved this idea. This is, admittedly, a very weak argument and should be supported by stronger facts.

Currently the TM fractional one-controllability figures from the j -th iteration are compared with the controllability tree figures from the j -th level. The comparison is performed for $j = 0$, the initial values, through to the converged results. The levels in the modified synchronizing tree correspond to the controllability TFN's. We lack benchmark fractional one-controllability figures for non-Q nodes in a sequential circuit. (This is another area which needs more research).

The final topic of this section, and of this chapter, is the benchmark observability calculations. These calculations yield results which are used to compare the accuracy of the TM observability figures. This benchmark procedure only yields fractional observability figures for the flip-flop output nodes in a sequential circuit.

The benchmark observability calculations are also derived from the expanded state table. This table is used to form unit distant state pairs. For a sequential

circuit with q flip-flops, i.e., q state variables, there will be q sets of state pairs; one set for each flip-flop, and each state-pair set contains 2^{q-1} state pairs. Each set is formulated for a specific state variable Q_i . The elements in each pair have complementary values in Q_i and equal values for the remaining state variables; hence the state pairs are unit distant.

Once the pairs have been found we use a tree of all the input sequences to find how many of the state-pair elements are distinguished, or split. The state-pair set for Q_i is the initial level in the tree; the next level of the tree contains state-pairs which have as elements the next states of the pairs in the initial level and possibly split elements. The third level contains state-pairs whose elements are the next states of the second level pairs, split elements which are the next states of the split elements on the second level and possibly newly split elements. The j -th level corresponds to the state-pairs for an input sequence of length j . A state pair, sp_i , is split for a given input if the outputs for each element in sp_i are different. When this occurs we can observe which state in sp_i the circuit was in. The fractional observability figures are calculated for each level of the tree. The figure

for Q_i at level j is given by the formula

$$\text{fractional OBS}(Q_i) = \frac{\# \text{ elements split at level } j}{\text{Total \# elements at level } j}$$

Let us find the observability tree for the state table of Fig. 3-2a using the state assignment of Fig. 3-4. Normally we need to find two trees, one for Q_1 and Q_2 . As an illustration of the observability benchmark calculations we will only find the tree for Q_2 . The state-pairs are (AB) and (CD). The tree is shown in Fig. 3-5.

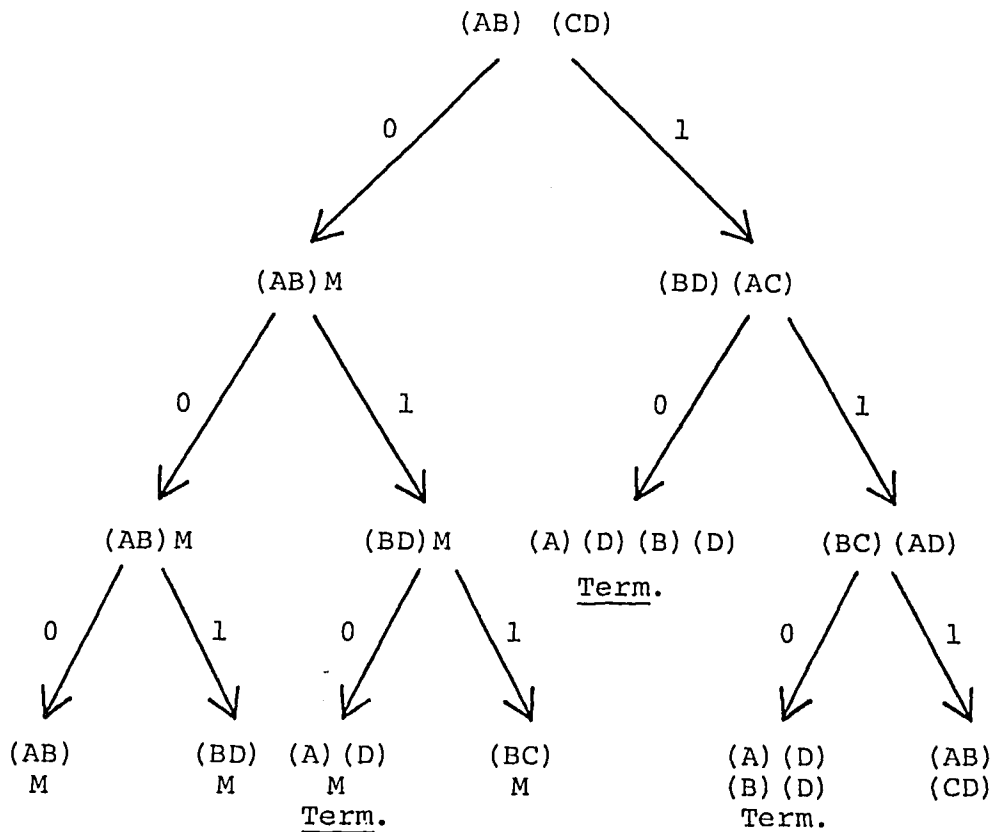


Figure 3-5 Observability Tree For Figure 3-2a.

The M in Fig. 3-5 represents a merged condition. Merged conditions occur when the elements in a state pair go to the same next state with identical outputs. When this occurs, one will never be able to distinguish these elements. Note that M actually represents two elements, a merged state pair. Nodes become terminal when, under a given input, all the state pairs either split or merge. A terminal node could also contain some combination of split and merged pairs. The fractional observability figures for Q_2 at levels 1, 2 and 3 are:

$$\text{level 1} \quad \text{fract OBS}(Q_2) = \frac{0}{8} = 0$$

$$\text{level 2} \quad \text{fract OBS}(Q_2) = \frac{4}{16} = \frac{1}{4}$$

$$\text{level 3} \quad \text{fract OBS}(Q_2) = \frac{(2*4) + 2 + 4}{2 * 16} = \frac{7}{16}$$

Although node 10 is terminal, it still contributes to the totals at the third level. The number of elements at the j -th level is $2^q * 2^{Nj}$, where j = level, N = number of input leads and q = number of state variables. Currently the fractional observability tree figures are compared to TM figures that have TFN's equal to the level of the tree. For example, if

$$\text{OBS}(Q_i) = \left\{ \frac{3}{4}, 2 \right\}$$

then we compare this to the tree figure from level 2.

A major problem still exists with the observability tree. The algorithm has no provision for terminating the tree. While it is true that some branches will become terminal, it is not true in general that all the branches will become terminal. One idea would be to terminate the tree when either all the branches become terminal, or if the fractional observability figure converges. That the figures will converge has not been proven, nor do we have significant data in this area.

Chapter 4

Algorithm Performance

4.0 Introduction

In this chapter we present the strong and weak points of the TM. The discussion concentrates on the major aspects of the algorithm's performance. The chapter begins with a proof of the exactness of the TM figures for fanout-tree combinational circuits. The TM is also shown to be exact for feedback free and fanout-free shift register circuits implemented with D flip-flops, i.e., shift registers which have only serial-in serial-out. Observability calculations for stem fanout leads are shown to be the weakest area of the TM calculations. This is shown to be especially true for the primary inputs in multi-leveled parity trees. An alternate method of calculating the observability figures is presented and shown to be not much of an improvement over our original method. We close the chapter with some remarks about the TM calculations for redundant circuits.

4.1 Exactness of the TM For Fanout-Free Combinational Circuits

The TFN's for fanout-free combinational circuits are always exact and can be proven by the following simple argument. Because these circuits are not clocked we need not wait a clock period (gate delays are ignored) to control or observe any of the circuit nodes. Thus the TFN's for each and every node in a combinational circuit are zero. By definition in the TM, all primary inputs have $TFN = 0$. As defined in Chapter 2, the operations used for TFN calculations in fanout-free combinational circuits are max, min and equivalence. Thus it is obvious that all the TFN's in these types of circuits will always be equal to zero.

The remainder of this section is devoted to proving the exactness of the fractional controllability and observability figures. For the rest of this section only whenever we write C_x^1 , C_x^0 , or $OBS(x)$, we are referring to the fractional one- and zero-controllabilities, and the fractional observabilities of lead x respectively.

We continue the discussion with a theorem concerning controllability calculations for inverters.

Theorem #1: For an i-th level inverter if the input controllability figure is exact then the TM output

controllability will also be exact.

Proof: Using Fig. 2-7 the TM dictates

$$C_1^1 = C_2^0 \quad (3-1a)$$

and $C_1^0 = C_2^1 \quad (3-1b)$

For an inverter to have a one (zero) at the output, there must be a zero (one) at the input. All input zero (one) vectors, that is all primary input combinations which yield a zero (one) on lead 1, will generate a one (zero) at the output, i.e., lead 2. Hence the fraction of input zero (one) vectors equals the fraction of output one (zero) vectors. Thus from Eqs. (3-1a) and (3-1b) if the input controllability is exact, then the output controllability will be exact.

To prove that the TM is exact for And and Or gates we first need to show that the one (zero)-controllability figures are actually a probability measure.

Theorem #2: For any lead x , in a fanout-free combinational circuit the TM one-and zero-controllabilities of lead x constitute a probability measure.

Proof: "A probability measure on the sample space Ω is a function of subsets of Ω satisfying three axioms: (i) For every set $A \subset \Omega$, the value of the function is a non-negative number, $P(A) \geq 0$.

(ii) For any two disjoint sets A and B, the value of the function for their union $A + B$ is equal to the sum of its value for A and its value for B, $P(A+B) = P(A) + P(B)$ for $A \cdot B = \emptyset$.

(iii) The value of the function for Ω (as a subset) is equal to 1, $P(\Omega) = 1$.⁵

In our sample space, Ω , there are only two events. Let us denote them as A and B, where

$A \triangleq$ lead x is at logic zero,

and $B \triangleq$ lead x is at logic one.

We define

$$P(A) \triangleq C_x^0 \quad (4-1)$$

and $P(B) \triangleq C_x^1 \quad (4-2)$

Recall that for fanout-free combinational circuits we work from Eqs. (2-3), (2-10) and (2-15). Note that Eq. (2-10) is simply the multiplication of the zero-controllability figures put in terms of the one-controllabilities. We also set all primary input controllabilities to 1/2. Because we always either multiply controllabilities or use equality the gate output controllabilities, and all leads, everywhere will always be in the

5 Kai Lai Chung, Elementary Probability Theory With Stochastic Processes (New York: Springer-Verlag, 1979), pp. 23-4.

range $[0,1]$. Hence axiom (i) is satisfied. It is intuitively obvious that A and B are complements. This also implies that these events are mutually exclusive, that is

$$A \cap B = \emptyset.$$

Also the union of A and B comprise the entire sample space Ω . Thus

$$A \cup B = \Omega$$

or
$$P(A \cup B) = P(\Omega)$$

where we have used standard probability notation. That lead x will be logic one or zero is a certainty (note that the algorithm considers only fault-free digital operation). Therefore

$$P(A \cup B) = P(\Omega) = 1.$$

Using the TM property that the zero- and one-controllabilities always sum to one and Equations (4-1) and (4-2) we obtain

$$P(A \cup B) = P(\Omega) = 1 = C_X^1 + C_X^1 = P(A) + P(B).$$

Thus axioms (ii) and (iii) are satisfied. Q.E.D.

The next theorem states that the TM is exact at any level i, provided the controllabilities are exact at the

i-1 level.

Theorem #3: For an i-th level n-input And or Or gate in a tree circuit if the input controllabilities are exact then the output controllabilities will be exact.

Proof: In Fig. 4-1 let

$x \triangleq$ The event lead x is a one (zero) for an i-th level And (Or) gate.

$A \triangleq$ The event lead 1 is a one (zero) for an i-th level And (Or) gate.

$B \triangleq$ The event lead 2 is a one (zero) for an i-th level And (Or) gate.

·
·
·

$N \triangleq$ The event lead n is a one (zero) for an i-th level And (Or) gate.

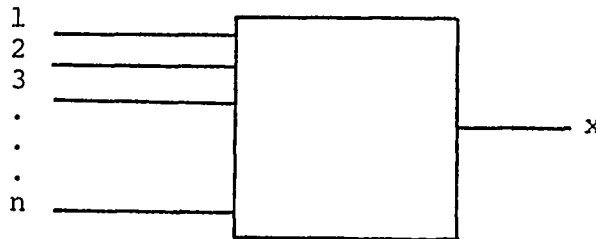


Figure 4-1 $n \geq 2$, i-th Level And (Or) Gate.

Now we know

$$P(X) = P(ABC\dots N)$$

Since the network is fan-out free, the inputs to the i-th level gate are all functions of different primary inputs. This means that each input is an independent variable, or event. Thus for independent events

$$\begin{aligned} P(X) &= P(ABC\dots N) \\ &= P(A) * P(B) * P(C) \dots * P(N) \end{aligned}$$

Using Theorem #2 this becomes

$$\begin{aligned} P(X) &= C_X^\alpha = P(A) * P(B) * P(C) \dots * P(N) \\ &= C_1^\alpha * C_2^\alpha * C_3^\alpha \dots * C_n^\alpha \end{aligned} \quad (4-3)$$

where $\alpha = \begin{cases} 1 & \text{for an And gate} \\ 0 & \text{for an Or gate} \end{cases}$

Equation (4-3) is equivalent to Eq. (2-3a) for the And gate and to (2-10a) for the Or gate. Thus if the input controllabilities of an i-th level gate are exact, then the output controllabilities will be exact.

Theorem #3 is now expanded to include Nand and Nor gates.

Corollary #1: For an i-th level n-input Nand or Nor gate in a tree, if the input controllabilities are

exact, then the output controllabilities will be exact.

Proof: In Fig. 4-2 we decompose the n-input Nand (Nor) gate into an n-input And (Or) gate and an inverter. By Theorem #3 we know that the controllability figure for lead x^1 will be exact if the input figures are exact. From Theorem #1 we know that the figure for lead x will be exact if the figure for lead x^1 is exact. Q.E.D.

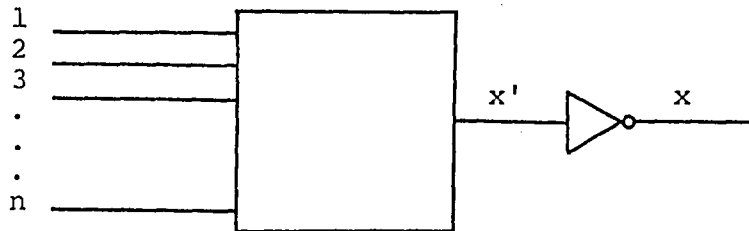


Figure 4-2 $n \geq 2$, i-th Level Decomposed Nand (Nor) Gate.

We are now ready to show that the controllability figures are always exact in combinational circuits with no fanout.

Theorem #4: The controllability measure is exact for all leads in a fanout free combinational network.

Proof: By definition the controllability values for the primary inputs are exact. Since the circuit is fanout free, each input of every first-level gate is a primary input. Hence these first-level gates all have

exact inputs. Thus by Theorem #3 all the outputs of the first-level gates are exact. Now each input of all the second-level gates is either a first-level gate output or a primary input. So all of these second-level inputs are exact. Hence by Theorem #3 all the second-level outputs are exact. The third-level gates all have inputs which are either second-level outputs, first-level outputs, or primary inputs. Since all of these inputs are exact, Theorem #3 tells us that all the third-level outputs will be exact. The next level's outputs will be exact, from Theorem #3, because this level has inputs from preceding levels which were shown to be exact. Thus each succeeding level will have exact output controllability figures because the figures for all preceding levels are exact. Q.E.D.

To complete the proof of the exactness of the TM we must show that the observability figures are exact. The following theorem states that if the observability of the output lead of an i -th level gate is exact, then the input lead observability figures are also exact.

Theorem #5: For an i -th level And, Or, or Inverter, if the output lead's observability figure is exact, then the input observability figures are exact.

Proof: Refer to Fig. 4-1. In this figure $n = 1$ for an inverter. The inverter observability equation,

Eq. (2-39), proof is obvious. To prove the exactness for And (Or) gates, we will use the event definitions in the proof of Theorem #3. In addition we define:

$X \triangleq$ The event lead x is observable.

$A \triangleq$ The event lead 1 is observable.

$B \triangleq$ The event lead 2 is observable.

·
·
·

$N \triangleq$ The event lead n is observable.

To observe input lead 1 the output lead, x, must be observable and 2 thru n must be set to a one (zero) for an And (Or) gate. In terms of probability, this translates to

$$P(A) = P(XABC\dots N)$$

Because the network is fanout free the action of controlling leads 2 thru n to a one (zero) does not affect the ability to observe leads x or 1. Thus the events X,A,B...N are independent and the above equation becomes

$$P(A) = P(X) * P(A) * P(B) \dots * P(N)$$

Using Theorem #2 this becomes

$$P(A) = P(X) * C_2^\alpha * C_3^\alpha \dots C_n^\alpha$$

where $\alpha = \begin{cases} 1 & \text{for an And gate} \\ 0 & \text{for an Or gate} \end{cases}$

Next we let

$$\text{OBS}(1) \triangleq P(A)$$

$$\text{OBS}(x) \triangleq P(X)$$

and obtain

$$\text{OBS}(1) = \text{OBS}(x) * C_2^\alpha * C_3^\alpha \dots * C_n^\alpha \quad (4-4)$$

This equation is identical to Eqs. (2-35a) for And gates and (2-38a) for Or gates. That this can be shown to be true for leads 2 thru n should be obvious. Thus the TM's input observability figures will be exact if the output lead's observability is exact.

It is now possible, with the aid of Theorem #5, to show that the TM observability figures are exact for every lead in a combinational circuit that has no fanout.

Theorem #6: The observability measure is exact for all leads in a fanout free combinational network.

Proof: The last, n-th, level in the circuit is at a gate which has as its output lead the primary output of the entire network. By definition this lead has an

observability of one. This figure is exact because a primary output is always observable. The inputs to this gate, at the n -th level, have observability figures which are known to be exact by Theorem #5. Gates at level $n-1$ feed into the inputs of the gate at level n . Thus the observabilities of the output leads at level $n-1$ are exact. By Theorem #5 we know that all the inputs to the gates at the n -th-1 level have exact observabilities. It is now obvious that we can work our way back, level by level, to the primary inputs at the first level and state that the primary inputs have exact observability figures. Q.E.D.

4.2 Exactness of the TM For Feedback Free Shift

Register Circuits

The Testability Measure produces exact results for a special class of sequential networks. This special class contains shift register circuits made from D flip-flops which do not have fanout or feedback loops. The TM fractional controllability, fractional observability and observability TFN's for this class of circuits exactly match our benchmark figures. Although these results are not exact in the conventional sense, we will show that they do exactly match our intuitive ideas

about shift register operation.

Figure 4-3 shows an N-stage shift register consisting of only D flip-flops. In this circuit node N is the primary output, z, and x is the primary input. There is no feedback in this circuit and except for the clock lines, there is no fanout either.

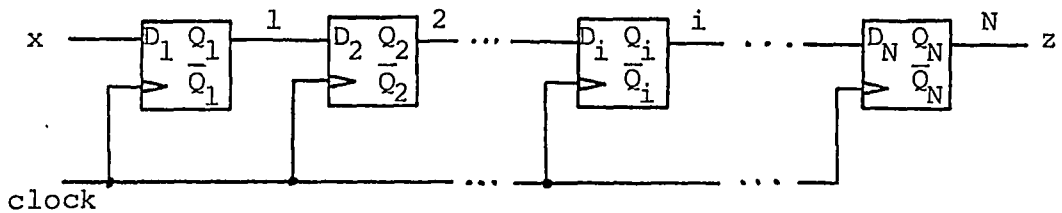


Figure 4-3 D Flip-Flop N-Stage Shift Register.

Theorem #7: The TM fractional controllability figure for lead i in an arbitrary length shift register made from D flip-flops is equal to the corresponding benchmark figure.

Proof: From Eq. (2-25) and the circuit configuration in Fig. 4-3 it is obvious that the fractional one- and zero-controllability of lead i, for all i, $1 \leq i \leq N$, is equal of $1/2$. Because the shift register has N stages, there are 2^N states in this machine. The next state vector is composed of the left most N-1 bits from the initial state, and the input bit. Let us call the N-1 bits from the initial state the core bits, or core for short.

Now it is obvious that there are 2^{N-1} different cores which occur twice among the 2^N states. Each core has two distinct successors in the modified synchronizing tree. Hence at level one each core appears four times and each state appears twice. At the second level each core appears eight times while each state appears four times. Therefore at level j in the modified synchronizing tree, for input sequences of length j , each of the 2^N states appears 2^j times, and there are 2^j paths in the tree. Thus at level j each state has probability

$$1/2^N * (1/2^j) * 2^j = 1/2^N$$

assuming each state and path are equally likely. For any lead i in the shift register it is intuitively obvious that there are 2^{N-1} members, or states in the one-controllability set and 2^{N-1} members in the zero-controllability set. Thus by the procedure in Section 3-2, the benchmark fractional one (zero)-controllability for lead i is

$$2^{N-1} * (1/2^N) * 2^j * (1/2^j) = 1/2$$

Q.E.D.

That the fractional one (zero)-controllability is $1/2$ for all leads in a shift register made from D flip-flops should come as no surprise. This fact matches

our intuitive ideas about this circuit. Since we can control any flip-flop output only by shifting data in from the x input, flip-flop controllability is equal to the x input's controllability. And because the x input is a primary input, its one (zero)-controllability is equal to 1/2.

The observability of lead N, the primary output in Fig. 4-3, is given by

$$\text{OBS}(Q_N) = \{1,0\}.$$

From Eq. (2-45b) we have

$$\text{OBS}(Q_{N-1}) = \{1,1\}.$$

Working back towards the primary input by (2-45b) we have

$$\begin{aligned} \text{OBS}(Q_{N-2}) &= \{1,2\} \\ &\vdots \\ \text{OBS}(Q_i) &= \{1,N-i\} && (4-5) \\ &\vdots \\ \text{OBS}(Q_1) &= \{1,N\} \\ \text{OBS}(x) &= \{1,N+1\} \end{aligned}$$

Using our distinguishing tree analysis we claim that for Q_i we have

$$\text{OBS}(Q_i) = \{1, N-i+1\} \quad (4-6)$$

where $N-i+1$ is the length of the input sequence you must apply to be able to observe node Q_i . Before we can prove the validity of Eq. (4-6) we first need the following theorem.

Theorem #8: For an N stage shift register there are 2^N state vectors, and exactly one way to form the 2^{N-1} unit distant state pairs for node Q_i .

Proof: Each stage may store either a zero or a one. Thus by the fundamental counting rule the number of state codes, or vectors, for N stages is

$$\underbrace{2 * 2 * \dots * 2}_{N \text{ terms}} = 2^N.$$

Consider one state vector, j , out of the 2^N total vectors. A unit distant state pair including vector j , to observe lead Q_i , necessitates finding another vector, k , which has a complementary value in the i -th bit. Now for vectors j and k to also be unit distant they must have identical values in the remaining $N-1$ bits. Thus there is only one way to pick k , and only one way to pair off the 2^N vectors. In Theorem #9 we prove the validity of Eq. (4-6).

Theorem #9: For an N stage shift register the distinguishing tree analysis gives the observability of node Q_i as

$$\text{OBS}(Q_i) = \{1, N-i+1\}.$$

In other words, all sequences of length $N-i+1$ will split all the state pairs and allow you to observe node Q_i .

Proof: The 2^{N-1} state pairs represent all of the possible unit distant ambiguities concerning the i -th flip-flop in the shift register. The crux of the problem is to resolve these ambiguities. The only way to do this is via some input sequence which yields different outputs for each state vector in each of the state pairs. Now since each state pair, from Theorem #8, is different only in the i -th bit, we must shift this bit to the output where it can be observed. Thus we must shift thru $N-i$ stages until the i -th bit is stored in the N -th stage. Because we are dealing with a Mealy machine we need one additional input to observe the output. Hence the required sequence length is $N-i+1$. Whether we shift in a zero or a one will not alter the outputs observed while the sequence of length $N-i+1$ is applied to the input. Thus we don't care which input value is selected, as long as $N-i+1$ values are applied to the input. Therefore all sequences of length $N-i+1$ are

usable to split the state pairs and to observe lead i , or node Q_i . Q.E.D.

It should be obvious to the reader that Eqs. (4-5) and (4-6) are almost identical. The TM results are different from the benchmark results only in the TFN. Recall that the TM assigns a value of zero to the TFN of the primary output, whereas in the distinguishing tree analysis we need an input sequence of length 1 to observe the output. The TM does not take into account the fact that a shift register is a Mealy machine and that a time frame is needed to observe the primary output. In the TM we assumed that the primary output was readily available. Thus the testability measure TFN's will always differ from the benchmark TFN's by one.

4.3 Observability of Stem Fanout Leads

Lead x in Fig. 2-15 is a stem fanout lead. The TM fractional observability figures for these leads frequently turn out poorly. As will be shown later in this section these figures are poor for a large, important class of circuits. Figure 4-4 shows a plot of the fractional observability error for stem fanout nodes in combinational circuits. This plot contains results taken from 33 nodes in 13 example circuits. The error

is plotted as a bar graph in terms of the number of nodes with a given error in vectors. The error in vectors is calculated using Eq. (4-7).

$$\text{Error} = |A - B| * 2^N \quad (4-7)$$

where A = the TM fractional observability
 B = the Boolean difference fractional observability
 N = the number of primary inputs

One of the main objectives of this thesis is to create a testability measure which has meaningful results. To accomplish this the results should closely approximate the exact figures. Figure 4-4 shows that this objective is not always reached. The plot has far too many results with errors greater than two vectors; nor are there any results with zero error. It was hoped that there would be almost no nodes with errors of three or more vectors. These erroneous results are very discouraging.

There exists a large important class of circuits, parity trees, for which the fractional observability figures are poor. The parity tree is made from exclusive-Or gates which in turn are made from two inverters, two And gates and an Or gate. The exclusive-Or gate

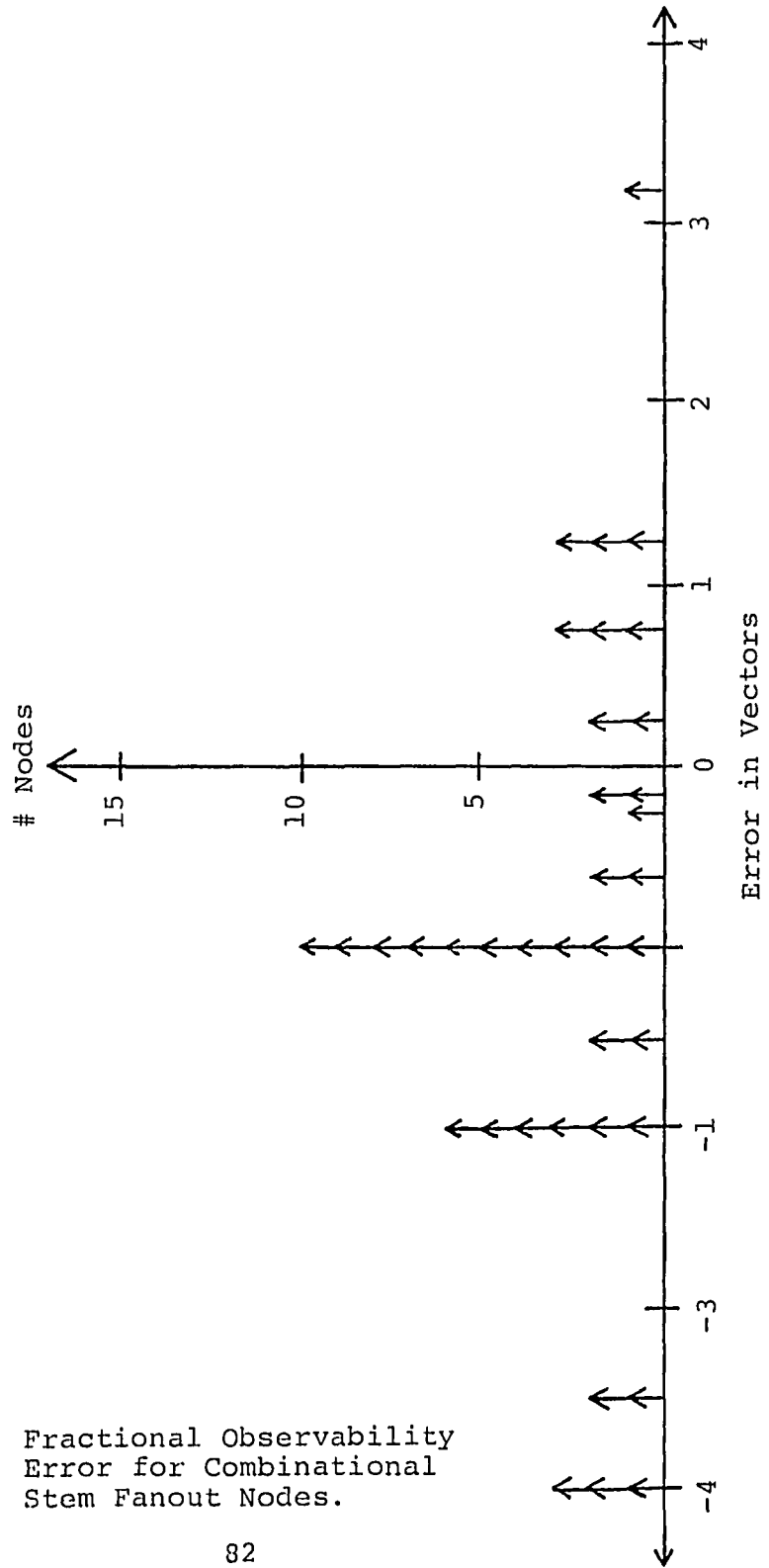


Figure 4-4 Fractional Observability Error for Combinational Stem Fanout Nodes.

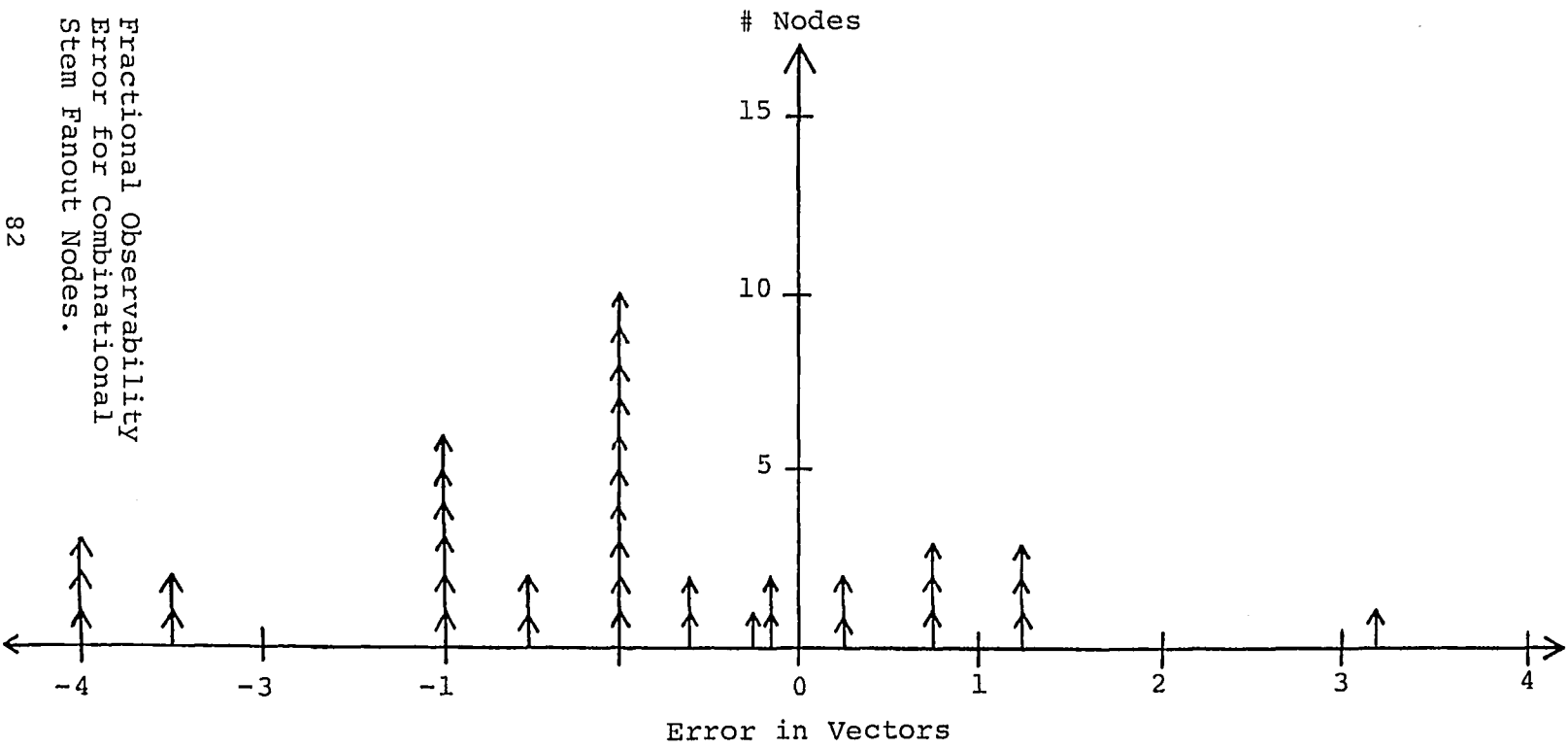


Figure 4-4 Fractional Observability Error for Combinational Stem Fanout Nodes.

configuration is called a cell and is shown in Fig. 4-5. Note that the cell in Fig. 4-5 has two levels, and a parity tree with two cells has four levels. As more levels are added to the tree, the TM fractional observability figures for the primary inputs (the stem fanout leads) worsen.

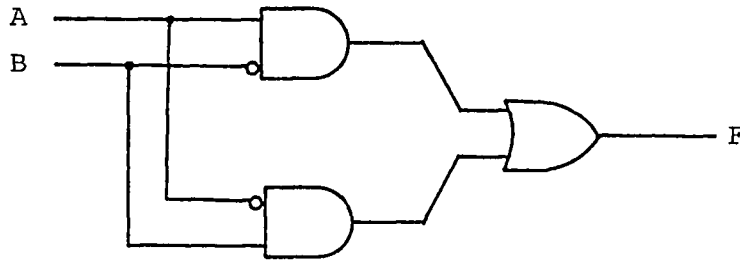


Figure 4-5 Two Input Exclusive-Or Cell.

No. of Levels/Cells	TM Fractional Observability
2/1	0.750
4/2	0.563
6/3	0.422
8/4	0.316
10/5	0.237

Table 4-1

Table 4-1 shows how the TM fractional observability figures for the primary inputs are affected by increasing the number of levels in the parity tree. The exact fractional observability of all stem fanout leads, as calculated by the Boolean difference, in a parity tree is one. Note how quickly the TM figures deteriorate. In a one-cell tree the error is one vector, while in a five-cell tree it is 48.8 vectors! This class of circuits clearly shows how inadequate the TM is for calculating the observability of stem fanout leads.

4.4 New TM Observability Calculations

The generally disappointing performance of the TM observability calculations prompted an attempt to improve upon these calculations. One of the more promising ideas appears in Eqs. (4-8) and (4-9); Figs. 2-12a and 2-13a, respectively, must be referenced to understand these equations. The advantages for an m-input And gate:

$$\text{OBS}(\alpha) = \left\{ d * \frac{n}{a}, \max(A, N, D) \right\} \quad (4-8)$$

where $1 \leq \alpha \leq m$ $\text{OBS}(n) = \{d, D\}$

$$C_n^1 = \{n, N\} \quad C_\alpha^1 = \{a, A\}$$

for an m-input Or gate:

$$\text{OBS}(\alpha) = \left\{ d * \frac{1-n}{1-a}, \max(A, N, D) \right\} \quad (4-9)$$

where $1 \leq \alpha \leq m$ $\text{OBS}(n) = \{d, D\}$

$$C_n^O = \{1-n, N\} \quad C_\alpha^O = \{1-a, A\}$$

of this idea are that no new controllability figures must be calculated and the m-input gate does not have to be decomposed. Thus this idea is simpler and quicker to use. Note that in the degenerate case, i.e., all m-inputs are independent, Eqs. (4-8) and (4-9) become Eqs. (2-35a) and (2-38a), respectively.

The investigation of the accuracy of these equations involved reworking our previous example circuits. In the examples the error, on the average, worsened by 0.6 vectors. This small decrease in accuracy is the price paid for the large simplification of the TM observability calculations. Unfortunately Eqs. (4-8) and (4-9) suffer from the same drawbacks that plaques our original mode of calculation. The new equations perform poorly for fanout nodes and especially poorly for parity trees. Table 4-2 shows the TM fractional observability results for a multi-level parity tree using the cell in Fig. 4-5. The results of Table 4-2 clearly contain larger errors

than the corresponding results in Table 4-1.

No. of Cells/Level	TM Fractional Observability
2/1	0.667
4/2	0.444
6/3	0.296
8/4	0.198
10/5	0.132

Table 4-2

4.5 TM Calculations For Redundant Circuits

In the course of investigating this algorithm it was discovered that the TM figures for redundant leads were sometimes poor. Although this area was not extensively explored, in some of the examples which had redundant leads the observability results contained sizable errors. We have included two such example circuits along with their respective TM and exact results in the Appendix. The first example contains a redundant And gate and a redundant primary input. The errors in this circuit are quite reasonable. But in the next example the figures contain huge errors. The worst error, 7 vectors, occurs at node eh. This error occurs because

the TM does not take redundancy into account. Thus it is possible for these types of circuits to cause the TM to generate highly erroneous results.

Chapter 5

Conclusions

This thesis contains the current status of an algorithm which measures how testable a given circuit is. Combinational and clocked sequential circuits with single outputs and no redundancy are the types of networks that the TM operates on. Redundant networks are not allowed because they can produce highly erroneous figures. The permissible circuit elements include And, Or, Nand and Nor gates, Inverters, D flip-flops and JK flip-flops. The flip-flops are not permitted to contain clear and preset inputs. RS flip-flops were incompletely studied.

Our algorithm meets the objectives which have been specified as essential characteristics of a testability measure. The TM is easier to compute than the test set. In combinational networks one way of finding the test set for stuck-at faults necessitates finding solutions

$$x^* \cdot df/dx = 1 \quad (5-1)$$

to Eq. (5-1)⁶ for each node in the circuit. This can be a very long and tedious process and can be much more

⁶ Kohavi, op. cit., 228-234.

time consuming than the TM calculations. Test-set generation in sequential circuits is even longer and more complex than the corresponding calculations for combinational circuits.⁷ Although the TM figures indicate testing difficulty, another objective of ours, there is no single comprehensive figure. Testing difficulty of a node is currently indicated by controllability and observability figures which have small fractional values and large TFN's. It is a matter of judgement as to what a "poor" TFN is, since no comparison criteria has been established. But since the TM figures are meaningful, and not simply numbers, discerning testing difficulty, or ease, is not impossible.

Ease of TM figure computation and meaningful results are two important features of our algorithm. The fractional controllability and fractional observability figures are fractions between zero and one which represent the portion of vectors that can be used to control or observe a given node. Controllability and observability TFN's represent the number of time frames or clock periods that must pass before a given node can be controlled or observed. Another strength of the algorithm is the ability to compare the TM figures to benchmark

7 Ibid., 449-470, 476-491.

figures, making it possible to check on the accuracy of the TM figures. Exact TM figures are obtained for fan-out free combinational networks and for feedback-free shift registers made from D flip-flops. Thus our algorithm performs well on simple network structures.

Even though the TM has many strong points, it is not without its weak points. The fractional observability figures for stem fanout leads are one such weak point because they frequently contain large errors. The largest errors are found in parity tree networks. As more levels are added to the tree, the error in the fractional observability figures increases. This error becomes unacceptably large very quickly, at the third level (see Table 4-1).

This algorithm is not complete and consequently there are some areas which need additional research. The TM can only be used in conjunction with single output networks. Refinements should be made on the TM to enable it to be used on multiple output circuits, since the majority of actual circuits are of this type. Controllability TFN calculations in sequential networks also needs more work. The TM should be altered to produce controllability TFN's which are exact for feedback-free shift register networks. Although we have formulated observability equations for JK flip-flops,

they have not been studied. In addition to studying these equations, new equations should be formulated to calculate the observability of RS flip-flops. The benchmark calculations for sequential circuits, while a useful addition to the algorithm, also need additional research. Currently there is no way to terminate the controllability and observability trees. If we could terminate these trees, then we could generate converged benchmark figures. There are also no benchmark figures for controllability TFN's and for non-Q nodes in a sequential circuit. And finally the poor results for the fractional observability of stem fanout nodes demonstrates that this portion of the TM must be reformulated.

TM calculations are currently done by hand. This is adequate for SSI and MSI circuits but not for LSI and VLSI size circuits. TM calculations for large networks should be done on a computer. To do this necessitates translating the contents of Chapters 1 and 2 into a computer program which can work from a circuit diagram as its input. Thus we must also automate the circuit diagram. The best way to automate the algorithm is to generate TM equations on the first pass through the network, then loop through the equations as many times as is needed. This procedure should yield the most efficient computerized version of the TM.

Bibliography

- [1] Chung, Kai L. Elementary Probability Theory With Stochastic Processes. New York: Springer-Verlag, 1979.

- [2] Goldstein, Lawrence H. "Controllability/Observability Analysis of Digital Circuits". Springfield, VA: National Technical Information Service, U.S. Department of Commerce, SAND 78-1895 (Nov. 1978).

- [3] Kohavi, Zvi. Switching and Finite Automata Theory. New York: McGraw-Hill Book Company, 1978.

- [4] Kovijanic, P.G. "Interactive Testability Analysis". (The publication is not known)

- [5] Susskind, Alfred K. "Testability and Reliability of LSI", Griffiss Air Force Base, NY: Rome Air Development Center, RADC-TR-80-384 (Jan. 1981), 99-122.

Appendix

Redundant circuit example #1. $F = AB + ABC$

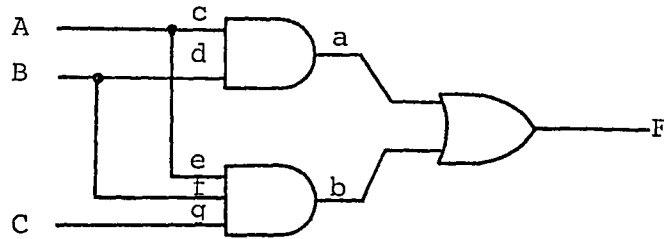


Figure A-1 Circuit Example #1.

Summary of Results

Ckt Node	TM		Exact		Errors (absolute)	
	C^1	OBS	C^1	OBS	C^1	OBS
output F	1/4	1	1/4	1	-	-
lead a	1/4	7/8	1/4	7/8	-	-
lead b	1/8	3/4	1/8	3/4	-	-
input A	1/2	5/8	1/2	1/2	-	1/8 = 1 vector
input B	1/2	5/8	1/2	1/2	-	1/8
input C	1/2	3/16	1/2	0	-	3/16 = 1.5 vectors

one vector = 1/8

Redundant circuit example #2.

$$F = (\overline{A}\overline{B}\overline{C} + \overline{C}D + (\overline{B}D + \overline{C}D)) + (ABC) + (\overline{B}D + \overline{C}D)$$

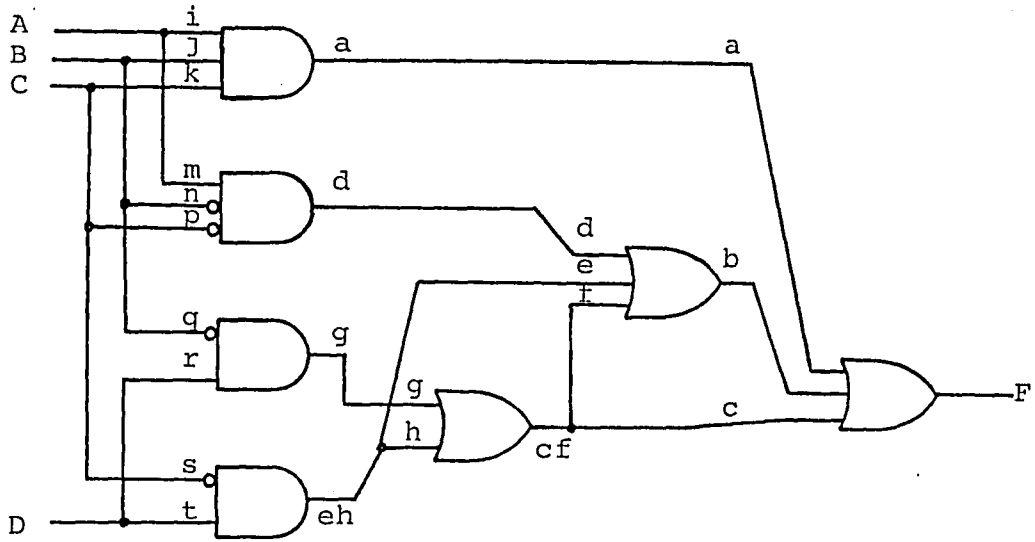


Figure A-2 Circuit Example #2.

Summary of Results

Ckt Node	TM		Exact		Errors (absolute)	
	C ¹	OBS	C ¹	OBS	C ¹	OBS
output F	0.452	1	0.563	1	0.110	-
lead a	0.125	0.615	0.125	0.563	-	0.053 = 0.84 vectors
lead b	0.385	0.553	0.438	0.5	0.053	0.053
lead cf	0.344	0.941	0.375	0.563	0.031	0.379 = 6.06 vectors
lead d	0.125	0.363	0.125	0.5	-	0.137 = 2.2 vectors
lead g	0.25	0.706	0.25	0.563	-	0.144 = 2.3 vectors
lead eh	0.25	1	0.25	0.563	-	0.438 = 7 vectors
input A	0.5	0.244	0.5	0.375	-	0.131 = 2.1 vectors
input B	0.5	0.444	0.5	0.375	-	0.069 = 1.1 vectors
input C	0.5	0.591	0.5	0.375	-	0.216 = 3.45 vectors
input D	0.5	0.853	0.5	0.625	-	0.228 = 3.65 vectors

one vector = 1/16 = 0.063

Summary of Results

Ckt Node	TM		Exact		Errors (absolute)	
	C ¹	OBS	C ¹	OBS	C ¹	OBS
output F	0.452	1	0.563	1	0.110	-
lead a	0.125	0.615	0.125	0.563	-	0.053 = 0.84 vectors
lead b	0.385	0.553	0.438	0.5	0.053	0.053
lead cf	0.344	0.941	0.375	0.563	0.031	0.379 = 6.06 vectors
lead d	0.125	0.363	0.125	0.5	-	0.137 = 2.2 vectors
lead g	0.25	0.706	0.25	0.563	-	0.144 = 2.3 vectors
lead eh	0.25	1	0.25	0.563	-	0.438 = 7 vectors
input A	0.5	0.244	0.5	0.375	-	0.131 = 2.1 vectors
input B	0.5	0.444	0.5	0.375	-	0.069 = 1.1 vectors
input C	0.5	0.591	0.5	0.375	-	0.216 = 3.45 vectors
input D	0.5	0.853	0.5	0.625	-	0.228 = 3.65 vectors

one vector = 1/16 = 0.063

Vita

Stephen Louis Kessler was born on April 19, 1957 in Levitown, Pennsylvania. Stephen and his brother, David, grew up together in Philadelphia where they lived with their mother, Rosalyn. Stephen graduated from Northeast Public High School in 1975 ranking 50th in a class of 1100. He then attended Lehigh University earning a B.S.E.E. degree in 1979. Stephen continued his education at Lehigh University's Graduate School from 1979 thru 1982. During this period he worked as a Teaching Assistant for the Department of Electrical Engineering. He also worked part-time for the Naval Surface Weapons Center in Dahlgren, Virginia, IBM in Endicott, New York, and for Bethlehem Steel's Homer Research Labs in Bethlehem, Pennsylvania. In January 1983, Stephen received a Master of Science degree in Electrical Engineering from Lehigh. Since August, 1982 he has worked for IBM in Endicott, New York. Stephen's interests include running, bicycling, chess, bridge and history.