1969

Study of diagnosability of binary address decoders

Girishchandra Mahendrakumar Gandhi

Department:

STUDY OF DIAGNOSABILITY OF BINARY ADDRESS DECODERS

by

GIRISHCHANDRA MAHENDRAKUMAR GANDHI, 1945-

A

THESIS

submitted to the faculty of

THE UNIVERSITY OF MISSOURI -- ROLLA

in partial fulfillment of the requirement for the

Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

Rolla, Missouri

1969

Approved by

S. S. Sy... (advisor) J. H. Tracey

C. Y. Ho

ABSTRACT

This paper studies the diagnosability of various types of binary address decoders. An attempt is made to develop a theory of diagnosis for logical faults that might occur in these logic nets. The developed theory is used to analyze these logic nets for various input combinations. Finally, the derivation of optimum diagnostic test sequences is considered.

ACKNOWLEDGEMENTS

The author wishes to express his sincere thanks to Dr. Stephen A. Szygenda for his guidance and advice in the preparation of this thesis.

Appreciation is also extended to Connie Hendrix for her typing efforts.

TABLE OF CONTENTS

Page

Abstract i

Acknowledgements ii

Table of Contents iii

List of Figures iv

List of Tables v

I. Introduction 1

II. Preliminaries 3

III. Simple Tree Decoder 7

IV. Dual Tree Decoder 32

V. Conclusions 49

References 50

Vita 52

LIST OF FIGURES

Figure		Page
1	n-Bit Binary Decoder	5
2	AND Circuit Logic Simple Tree Decoder for n = 4	10
3	% Diagnosability for the Number of Tests Applied to a Simple Tree Decoder	31
4	Dual Tree Decoder for n = 4	33
5	Symbolic Design of a Dual Tree Decoder for n = 8 and 9	35
6	General Symbolic Diagram of Dual Tree Decoder When n is a Power of 2	37
7	Dual Tree Decoder for n = 8	38
8	General Symbolic Diagram of Dual Tree Decoder When n is Not a Power of 2	42
9	Dual Tree Decoder for n = 5	43
10	% Diagnosability for the Number of Tests Applied to a Dual Tree Decoder	48

LIST OF TABLES

Table		Page
I	% Diagnosability for the Number of Tests Applied	29
II	% Diagnosability Against the Number of Tests Applied	46

I. INTRODUCTION

This paper investigates the diagnostic properties of fault detection tests for simple and dual tree binary address decoders. The reliability and maintainability of a circuit is affected by any fault that might occur in that circuit. Diagnosis is an effort to insure the maintainability of these circuits. In most cases, this approach is based on the assumption that only logical stuck at one (s-a-1) or stuck at zero (s-a-o) logical faults can occur. The following assumptions are used throughout this study: there are only two input lines to each gate, and if one of the inputs is s-a-1 the other is considered to be 1 for fault detection. The problem of diagnostic test generation for combinational circuits has been studied extensively.

Armstrong[5] has described a path sensitizing concept and reduction of a net to its equivalent normal form. This procedure becomes cumbersome when the number of paths through the net is large. Hence, this approach is only applicable for small circuits. Kautz[4] has described a procedure which can detect and locate the fault. The drawback of this approach is that only a small number of faults can be considered. Amar[7] has described a procedure of diagnosis for large combinational networks, but the tests derived are not necessarily optimum. Poage[8] has derived optimum numbers of tests to detect faults in combinational

networks. For this approach, tests needed to detect single faults are derived by simplifying the fault table. Again, this procedure is only good for small circuits.

The major goal of this study has been to determine the diagnosability that can be achieved through the application of incomplete test sets. This would provide an optimization of test that is not easily achievable for large decoders by any of the existing techniques. With the information provided by this study it would be possible to design a rapid (minimum) fault detection sequence of tests for any specified percentage of possible faults.

II. Preliminaries

Throughout this paper, we will consider any deviation from the proper network behavior as a failure. These network failures can be arrived at from a number of sources, such as element aging or breakdown, shorted or opened interconnections and noise. These sources of failures can never be completely eliminated so that there is always a non-zero probability that any combinational network may fail. Although network failure is an ever-present possibility, the probability of failure can be materially decreased by careful element design, by the use of long-life components, by reliable interconnection techniques, and by careful shielding from noise. Hence, the first step in any attempt to improve the reliability of a network is careful attention to element design and network fabrication.

Most sources of network failure can be classified as (1) intermittent (2) catastrophic and (3) drift.

An intermittent source is a non-permanent, non-destructive change which may cause a network to pass back and forth between the failed and non-failed condition. Intermittent sources of failure include such factors as noise, shorted or opened interconnections caused by vibrations, and marginal parameters. This type of failure is not considered in this study.

A catastrophic failure is a descructive or permanent change which causes a network to fail and to remain in a failed condition. Catastrophic sources of failures include such things as burned out components and permanently shorted or opened interconnections. This type of failure is considered in this study. This failure can be diagnosed by some tests. Diagnosis is a procedure to identify a failure if it exists in the circuit, by a test we mean the process of applying a set of inputs to the machine or gates and observing the corresponding output. Both intermittent and catastrophic sources are assumed to be sudden and unanticipated.

In contrast, a failure caused by drift passes first through an intermittent stage due to marginal parameters and finally into a catastrophic stage when the parameters have drifted far enough to cause a permanent failure. Drift sources of failures include such items as aging or wear of components.

A transient failure is caused by intermittent and/or drift failure sources, and a permanent failure is caused by catastrophic and/or drift failure sources.

The following constraints are imposed on this study:

(1) A single fault assumption, i.e. one and only one failure is assumed to have occured at any one time.

(2) A physical element (wire) of a gate network may exist in any one of the three elements states.

(a) The wire is normal in which case its signal may switch between the value 1 and the value 0.

(b) The wire is permanently 1, in which case a signal of constant value 1 appears on the wire, this is termed as stuck at one (s-a-1).

(c) The wire is permanently 0, in which case a signal of constant value 0 appears on the wire, this is termed as stuck at zero (s-a-0).

(3) Decoder is an AND circuit logic and the gates consist of only diodes.

Figure 1 provides a block diagram of an n input address decoder.

In an address decoder there are n input lines (considering uncomplemented variables only) and 2^n output lines

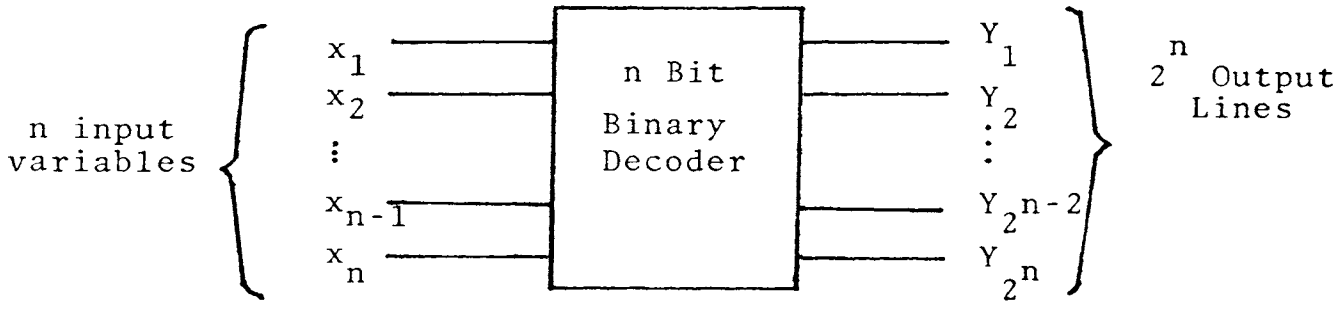


Figure 1: n-Bit Binary Decoder

Definition:

A binary address decoder is a circuit with n input variables and 2^n output lines arranged so that for each of the possible 2^n states of the input lines, one and only one output line is a 1 and the rest are 0.

Thus, from the circuit point of view, a decoder is a circuit with multiple inputs and outputs. From a practical point of view, the decoder is a important circuit because it translates the representation of a number from the conventional highly compact positional notation to a 1-out-of-n form that is required for physical selection of a single desired object from among several objects.

Figure 1 represents an n-bit binary decoder in which $\vec{x} = x_1, x_2, \ldots, x_n$ are input lines and $\vec{y} = y_1, y_2, \ldots, y_{2^n}$ represent the output vector for a specified output.

Therefore,

$$y_1 = x_1\, x_2\, x_3 \ldots x_{n-1}\, x_n$$
$$y_2 = x_1\, x_2\, x_3 \ldots x_{n-1}\, x_n'$$
$$\vdots$$
$$y_{2^n} = x_1'\, x_2'\, x_3' \ldots x_{n-1}'\, x_n'$$

From the definition of an address decoder for each input combination one of the y_i will be 1, the others will be zero.

III. Simple-Tree Decoder

A simple tree decoder is a particular type of multi-terminal logic block network having a single input vector which may be connected to any one of a number of outputs. As defined previously, one and only one output is connected to the input at a given time. A simple tree decoder is a less costly approach than a one level decoder and is shown in figure 2.

The following properties exist for a simple tree decoder,

a) Levels (L) = n-1

b) Fan-in (FI) = 2

c) Fan-Out (FO) = 2 (internal
= 2^{n-1} (for input drivers)

The following theorems are developed to establish the diagnosability of address decoders:

Theorem: 1

2^n test vectors are required to detect all s-a-o faults on output lines of a binary address decoder.

Proof:

Consider Figure 1

Let $y_1 = x_1\ x_2\ x_3 \ldots x_n$

$y_2 = x_1\ x_2\ x_3 \ldots x_n'$

$\vdots \quad \vdots\ \vdots\ \vdots\ \vdots\ \vdots\ \vdots\ \vdots$

$y_{2^n} = x_1'\ x_2'\ x_3' \ldots x_n'$

From the decoder definition, it can be seen that y_1 the output vector can only be one when the input vector is $x_1x_2x_3 \ldots x_n$ otherwise it is zero. Similar statements can be made for the other outputs. Also, each output vector is one when the respective input vector is applied.

Let the output vector y_1 be 1 only when all variables x_1 through x_n are one. The only possible fault is s-a-o on the output line of the vector y_1. If we apply other than $x_1x_2x_3....x_n$ as an input vector there will be no effect on the output for this s-a-o fault.

Therefore, this s-a-o fault can only be detected by $x_1x_2x_3...x_n$ input vector because it gives y_1 to be one under fault free conditions.

Hence, from the above arguments it is clear that to detect s-a-o faults on the output lines, it is necessary to apply all the possible test vectors.

It is now proven that 2^n test vectors are required to detect all the s-a-o faults on the output lines of the decoder.

Example:

Let n = 4

Therefore, the number of output lines is $2^n = 16$.

Let $y_1 = x_1\ x_2\ x_3\ x_4 = 1$ under fault free conditions.

The possible fault on this line is s-a-o. This fault can only be detected by applying test $x_1x_2x_3x_4$,where all

literals are 1 because under fault-free conditions this output is 1 and the others are zero and under faulty conditions all outputs are zero.

Similar statements can be made for any other output line.

Therefore, for n = 4 sixteen tests are required to detect all s-a-o faults on the outputs.

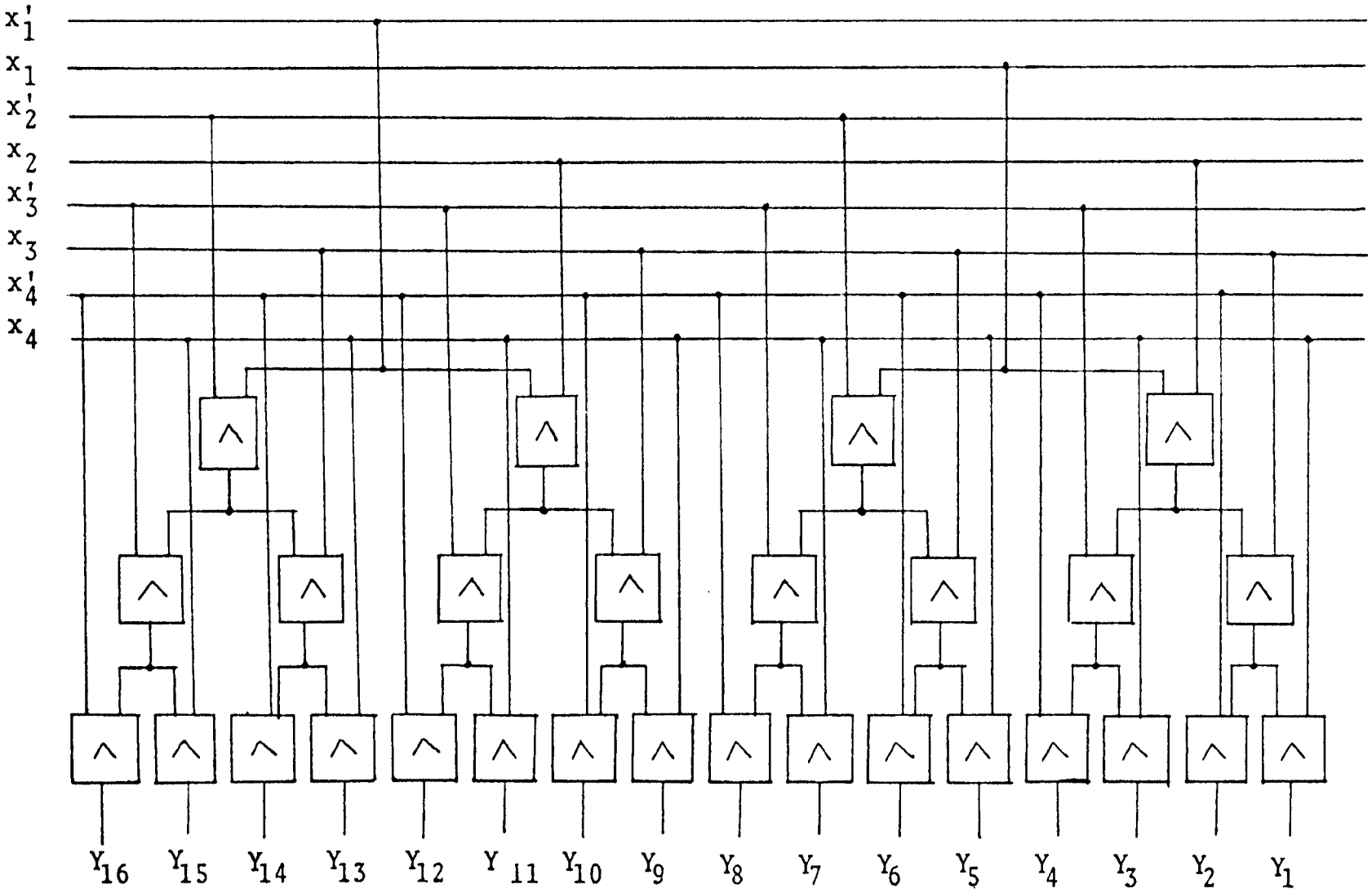


Figure 2 : AND Circuit Logic Simple Tree Decoder for n=4

Theorem: 2

2^n test vectors are required to detect the s-a-1 faults on the incoming lines from the primary bus to gates in the last level of a simple tree decoder.

Proof:

$$\text{Let } y_1 = x_1\ x_2\ x_3\ \ldots\ x_{n-1}\ x_n \quad \text{——} \quad (1)$$

$$y_2 = x_1\ x_2\ x_3\ \ldots\ x_{n-1}\ x_n' \quad \text{——} \quad (2)$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$y_{2^n} = x_1'\ x_2'\ x_3'\ \ldots\ x_{n-1}'\ x_n' \quad \text{——} \quad (2^n)$$

In a simple tree decoder each gate has one input coming directly from the primary bus and the other line from a previous level gate output.

In Equation (1) it can be seen that the vector $x_1x_2\ldots x_{n-1}$ is formed from the (L-1)th level gate output and x_n comes directly from the primary bus.

Now, if $x_1x_2x_3\ldots x_{n-1}$, x_n test is applied the y_1 vector will be 1 and the rest zero. This means that y_1 is one only when x_1, x_2, x_3,...,x_{n-1}, x_n are all one. Also observe the output vector y_2. $y_2 = x_1x_2x_3\ldots x_{n-1}x_n'$. Under test y_1, x_n' is equal to zero. The possible fault on x_n' is s-a-1. Therefore, under this fault y_2 is also 1. This means that this fault can be detected by the $x_1x_2x_3\ldots x_{n-1}x_n$ test, because under this test there are two outputs being

active. If we apply other than this test there will be no effect on the output and the fault can not be detected.

Similar arguments can be made for the rest of the lines.

Therefore, from the above arguments it can be seen that to detect s-a-1 faults on the incoming lines from the primary bus to each gate in the last level in a simple tree decoder, it is necessary to apply all possible test vectors.

Example:

Let $n = 4$

Let $y_1 = x_1x_2x_3x_4$

$y_2 = x_1x_2x_3x_4'$

Let y_1 be equal to 1 under fault free conditions. This means that $x_1 = x_2 = x_3 = x_4 = 1$ and the complements are zero. Therefore possible fault on x_4' is s-a-1. The x_4' line is directly coming from the primary bus. This fault can be detected by applying $x_1x_2x_3x_4$ as a test, because under this test y_1 and y_2 both are one. Other tests have no effect on the output.

Similar arguments hold good for other lines.

Therefore, for $n = 4$, sixteen tests are required to detect these s-a-1 faults.

In theorems (1) and (2) it is proven that to detect all faults it is necessary to apply all the possible tests.

It is possible that other checking techniques may be used to diagnose these output faults. Hence, the question now is how many tests are required to detect the remaining

faults in the decoder.

(A) Number of tests required to detect all faults per level.

A simple tree decoder is a symmetric circuit for any number of inputs. For any n, there are always four gates at the first level; eight gates at the second level and 2^n gates at the Lth level.

Therefore, it can be concluded that in a simple tree decoder each level has a number of gates which is a power of 2 i.e. Per level there are $4,8,16,\ldots,2^{n-2},2^{n-1},2^n$ gates.

$$\text{Let } y_1 = x_1\, x_2\, x_3 \ldots\ldots x_{n-1}\, x_n \quad \text{——} \quad (1)$$

$$y_2 = x_1\, x_2\, x_3 \ldots\ldots x_{n-1}\, x_n' \quad \text{——} \quad (2)$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$y_{2^n} = x_1'\, x_2'\, x_3' \ldots\ldots x_{n-1}'\, x_n' \quad \text{——} \quad (2^n)$$

Equations (1) through (2^n) are the outputs at the Lth level. It has been proven in theorems (1) and (2) that it is necessary to apply all test vectors to detect all faults. Therefore, at the Lth level it is necessary to apply all tests to detect all faults at that level.

Since there are four gates at the first level, there will be four outputs in the form of x_1x_2, x_1x_2', $x_1'x_2$ and $x_1'x_2'$.

Let x_1x_2 be equal to one under fault free conditions. Note that the others will be zero.

The possible fault on the x_1x_2 line is s-a-o. This can only be detected when all outputs are zero.

The above fault can be detected by applying a $x_1x_2$dd... d test (d=don't care) because, under this test all outputs will be zero. If we apply other than this test, there will be no effect on the output and the fault will not be detected.

Similarly if x_1x_2', $x_1'x_2$ or $x_1'x_2'$ s-a-o, it is necessary to apply x_1x_2' dd .... d, $x_1'x_2$ dd ... d and $x_1'x_2'$dd...d respectively for fault detection.

Therefore, to detect all faults at the first level it is necessary to apply four tests, one from each of the following sets of tests.

$x_1x_2$dd . . . . d

x_1x_2'dd . . . . d

$x_1'x_2$dd . . . . d

$x_1'x_2'$dd . . . . d

Similar arguments hold good for other levels.

Hence, to detect all faults per level it is necessary to apply 4,8,16,...,2^{n-2}, 2^{n-1}, 2^n tests respectively.

Example:

Let n = 4

The outputs at the first level are x_1x_2, x_1x_2', $x_1'x_2$ and $x_1'x_2'$.

Let $x_1'x_2'$ be equal to 1 under fault free conditions, the others are zero by definition.

To detect s-a-o faults on the line $x_1'x_2'$ it is necessary to apply $x_1'x_2'dd$ tests because, under these tests all outputs will become zero. If we apply other than these tests there will be no effect on the output and the fault will not be detected.

Similarly, to detect s-a-o faults on lines $x_1'x_2$, x_1x_2' and x_1x_2 it is necessary to apply $x_1'x_2dd$, $x_1x_2'dd$ and x_1x_2dd tests respectively.

Therefore, to detect all faults at the first level it is necessary to apply the tests as follows: x_1x_2dd, $x_1x_2'dd$, $x_1'd_2dd$ and $x_1'x_2'dd$.

Similarly, for the next level the following test vectors are necessary to detect all faults at that level, $x_1x_2x_3d$, $x_1x_2x_3'd$, $x_1x_2'x_3d$, $x_1x_2'x_3$, $x_1'x_2x_3d$, $x_1'x_2x_3'$, $x_1'x_2'x_3d$, $x_1'x_2'x_3'd$. For the last level it is necessary to apply all tests to detect all faults at that level.

They are,

$$x_1x_2x_3x_4,\ x_1x_2x_3x_4',\ x_1x_2x_3'x_4,\ x_1x_2x_3'x_4',$$

$$x_1x_2'x_3x_4,\ x_1x_2'x_3x_4',\ x_1x_2'x_3'x_4,\ x_1x_2'x_3'x_4',$$

$$x_1'x_2x_3x_4,\ x_1'x_2x_3x_4',\ x_1'x_2x_3'x_4,\ x_1'x_2x_3'x_4',$$

$$x_1'x_2'x_3x_4,\ x_1'x_2'x_3x_4',\ x_1'x_2'x_3'x_4,\ x_1'x_2'x_3'x_4'.$$

(b) Diagnosability:

It has been shown that to detect all faults per level, it is necessary to apply $4,8,16,\ldots,2^{n-2},2^{n-1},2^{n}$ tests. Now, the question is how many faults can be detected in other levels by applying some specified number of tests. For example, at the first level four tests are applied to detect all the faults at that level, but by applying these four tests how many faults can be detected in other levels. The ratio of the number of faults can be detected to the total number of faults present is called the diagnoability.

Let Ft = Total number of faults in a decoder of any n.

FLT = Number of faults can be detected by applying a specified number of tests.

Then,

$$\text{Diagnosability (D)} = \frac{FLT}{FT}$$

and % diagnosability $= \frac{FLT}{FT} \times 100\%$

Part (1):

Let $Y_1 = x_1x_2x_3 \ldots x_{n-1}x_n$ ——— (1)

$\vdots \quad \vdots\ \vdots\ \vdots \quad \vdots\ \vdots\ \vdots\ \vdots\ \vdots \quad \vdots$

$Y_2^{n-3} = x_1'x_2'x_3' \ldots x_{n-2}'x_{n-1}x_n$ ——— (2^{n-3})

$Y_2^{n-2} = x_1'x_2'x_3' \ldots x_{n-2}'x_{n-1}x_n'$ ——— (2^{n-2})

$Y_2^{n-1} = x_1'x_2'x_3' \ldots x_{n-1}'x_n$ ——— (2^{n-1})

$Y_2n = x_1'x_2'x_3' \ldots x_{n-1}'x_n'$ ——— (2^{n})

Let $Y_2n=1$ under fault free conditions and hence the other Y_i are all zero. Since $Y_2n=1$, $x_1', x_2', x_3', \ldots, x_n'$ all must be equal to one. From the circuit it can be seen that the $x_1'x_2'x_3'\ldots x_{n-1}'$ vector is coming from the (L-1)th level gate output and is branched into two adjacent gates at the Lth level. Since $Y_2n=1$, the $x_1'x_2' \ldots x_{n-1}'$ is equal to one under fault free conditions. Therefore, the only possible fault on this line is s-a-o. This line branches to two adjacent gates at the Lth level, therefore, the s-a-o faults has effect on these two gates (i.e. the effect on Y_2n-1 and Y_2n vectors).

The above fault can be detected by applying $x_1'x_2'$ $x_3'\ldots x_{n-1}$d tests because under these tests a-1 the outputs will be zero. If we apply other than these tests there will be no effect on the output and the fault will not be detected.

Since the simple tree decoder is completely symmetrical, similar arguments can be made for the other lines. Therefore, from the above arguments it can be concluded that to detect s-a-o faults at the (L-1)th level it is required to apply any one of the two adjacent outputs.

Now, consider the output $x_1'x_2' \ldots x_{n-2}'$ at the (L-2)nd level. Let it be equal to 1 under fault free conditions. Therefore, the only possible fault on this line is s-a-o. This effects the four adjacent outputs at the Lth level. This fault can be detected by $x_1'x_2'x_3' \ldots x_{n-2}$dd tests because under these tests all outputs will become zero

Otherwise, there will be no effect on the output and the fault will not be detected. The $x'_1 x'_2 x'_3 \ldots x'_{n-2}$ dd vector represents four tests. Similar arguments hold good for the other lines.

Similarly, at the first level consider the output $x'_1 x'_2$ and let it be equal to 1. Therefore, the only possible fault on this line is s-a-o. This fault can be detected by test $x'_1 x'_2$ d ... d because under this test all outputs will become zero.

In general, when there is s-a-o fault on any output, at any level, this fault can be detected by a test comprised of the input variable combinations which makes up the output vector on which the fault occurs.

Part 2:

Consideration of the s-a-1 faults on outputs at the (L-1)th level.

Let $x_1 x_2 \ldots x_{n-1} x_n$ be equal to one under fault-free conditions. This means that all digits x_1 through x_n are one. Therefore, at the (L-1)th level $x_1 x_2 \ldots x_{n-1}$ vector is equal to one and others are zero.

Consider $x'_1 x'_2 \ldots x'_{n-1} = 0$ under fault free conditions. The possible fault on this line is s-a-1. Since, this line is branched into two gates at the Lth level this fault has an effect on the two gates $x'_1 x'_2 \ldots x'_{n-1} x'_n$ and $x'_1 x'_2 \ldots x'_{n-1} x_n$ will be effected by this fault. This fault can be

detected by applying any other tests except the two which are effected by the fault because under those tests there will be no effect on the output and the fault will not be detected. Whereas, by applying other than these two tests there will be two outputs active and hence the fault is detected. This means that at (L-1)th level the s-a-1 fault on the output can be detected by applying 2^n-2 tests. Similar arguments hold good for any other level.

Let us consider the s-a-1 fault on the output at the first level. Let $x_1'x_2'$ be equal to zero under fault-free conditions. The possible fault on this line is s-a-1. This fault can be detected by any test other than that which consists of the $x_1'x_2'$ vector = 1 (i.e. $x_1'x_2'$d ... dd) because under these tests there will be no effect on the output and the fault will not be detected. Whereas, by applying other than these tests there will be effect on the output and the fault is detected. This means that this fault can be detected by $3(2^{n-2})$ tests.

Hence, in general, it can be concluded that the s-a-1 fault on any output and at any level can be detected by applying any other tests except those which are affected by the fault.

Example:

Let n = 4

Let x_1' x_2' x_3' be one of the outputs at (L-1)th level and let it be equal to zero under fault-free conditions.

The possible fault on this line is s-a-1. This line is branched into two gates at the output level. The outputs of these two gates are $x'_1 x'_2 x'_3 x'_4$ and $x'_1 x'_2 x'_3 x_4$. Therefore, the s-a-1 fault has an effect on these two vectors. Hence, the s-a-1 fault can be detected by applying any other test except $x'_1 x'_2 x'_3 d$, because under this test there will be no effect on the output and the fault will not be detected. This fault can be detected by applying any one of $2^n - 2^4 = 2^4 - 2 = 14$ tests. Similar arguments hold good for other lines.

Let $x'_1 x'_2$ be equal to zero and be one of the outputs at the first level under fault free conditions. The possible fault on this line is s-a-1. This fault has an effect on $x'_1 x'_2 x'_3 x'_4$, $x'_1 x'_2 x'_3 x_4$, $x_1 x'_2 x_3 x'_4$ and $x'_1 x'_2 x_3 x_4$. Therefore, this fault can be detected by applying any other test except $x'_1 x'_2 dd$ because, this test has no effect on the output. This fault can be detected by any one of $3(2^{n-2}) = 3(2^{4-2}) = 12$ tests.

Part 3:

Consideration of the s-a-1 faults on each branch:

Let $Y_n = x'_2 x'_1 x'_2 x'_3 \ldots x'_{n-1} x'_n$

$x'_1 x'_2 x'_3 \ldots x'_{n-1}$ is coming from the (L-1)th level of output. Let $x'_1 x'_2 x'_3 \ldots x'_{n-1}$ be equal to zero under fault-free conditions. This means that one or more of the binary digits are zero. Therefore, the possible fault on this line is s-a-1.

To detect this fault x'_n must be 1, otherwise, there will be no effect on the output and the fault will not be detected. This fault can be detected by applying any of the ddd ... d x'_n tests except $x'_1\ x'_2\ x'_3 \ldots x'_n$ because, under this test there will be no effect on the output and the fault will not be detected.

According to the truth table this fault can be detected by all even numbered output vectors except the one which has no effect on the output. Out of 2^n outputs there are 2^{n-1} even and odd numbered output vectors. Therefore $(2^{n-1}-1)$ tests can detect this fault.

Now let $Y_2^{n-1} = x'_1\ x'_2 \ldots x'_{n-1}\ x_n$ and let $x'_1\ x'_2 \ldots x'_{n-1}$ be zero under fault-free conditions. The possible fault on this line is s-a-1.

To detect this fault x_n must be 1. This fault can be detected by dd ... dx_n tests except $x'_1\ x'_2 \ldots x'_{n-1}\ x_n$ test because under this test there will be no effect on the output and the fault will not be detected.

According to the truth table this fault can be detected by all odd numbered output vectors except the one which has no effect on the output.

Similar arguments hold good for other branched lines at the Lth level.

Let the lines which are branched on the right side be numbered as 1 and those on the left side be numbered as 2.

Hence, the s-a-1 fault on line 1 can be detected by all odd numbered output vectors except the one which has no effect on the output and the s-a-1 fault on line 2 can be detected by all even numbered output vectors except the one which has no effect on the output.

Consider the branched input lines at the (L-1)th level:

Let $x_1' x_2' \ldots x_{n-2}' x_{n-1}$ be one of the output vectors at the (L-1)th level. A vector $x_1' x_2' \ldots x_{n-2}'$ is branched into two gates at the (L-1)th level from the (L-2)nd level of output. Let $x_1' x_2' \ldots x_{n-2}'$ is equal to zero under fault free conditions, then the possible fault on this line is s-a-1. To detect this fault x_{n-1}' must be 1. Therefore, this fault can be detected by dd ... x_{n-1}'d tests except $x_1' x_2' \ldots x_{n-2}' x_{n-1}$d tests because under these tests there will be no effect on the output and the fault will not be detected.

Let $x_1' x_2' \ldots x_{n-2}'$ is equal to zero under fault-free conditions, then the possible fault on this line is s-a-1. To detect this fault, x_{n-1} must be equal to 1. This fault can be detected by dd ... dx_{n-1}d tests except $x_1' x_2' \ldots x_{n-2}' x_{n-1}$d tests because, under this test there will be no effect on the output and the fault will not be detected. Similar arguments hold good for all other lines.

Hence, from the above arguments it can be concluded that the two test patterns are different. Therefore, tests should

be selected in such a way that the s-a-1 faults on the right-hand side lines and those on the left-hand side lines can be detected. This can be done by selecting half even and half odd numbered tests from the first half and last half of total tests respectively.

Similarly, let x_1' be branched into two gates at the first level. Let $x_1'x_2'$ be equal to zero under fault free conditions. Let x_1' be equal to zero therefore possible fault on x_1' is s-a-1. To detect this fault x_2' must be one. Therefore this fault can be detected by $x_1'x_2'$d ... d tests. x_1' is also branched into other gates whose output is $x_1'x_2$. Let x_1' be equal to zero. Therefore, possible fault on x_1' is s-a-1. To detect this fault x_2 must be equal to one. Hence, this fault can be detected by $x_1'x_2$d ... d. Similar arguments hold good for other lines.

Example:

Let n = 4

Let $Y_{16} = x_1'\ x_2'\ x_3'\ x_4'$

and let $Y_{15} = x_1'\ x_2'\ x_3'\ x_4$

The $x_1'\ x_2'\ x_3'$ vector is coming from the (L-1)th level and is branched into two gates at the Lth level. Let Y_{16} vector be equal to zero under fault free conditions. Let $x_1'x_2'x_3'$ vector be zero. The possible fault on this line s-a-1. To detect this fault x_4' must be one. Therefore, this fault can be detected by dddx_4' tests except $x_1'x_2'x_3'x_4'$ because under

this test there will be no effect on the output and the fault can not be detected. Therefore, the following tests will detect this fault.

$x_1x_2x_3x_4'$, $x_1x_2x_3'x_4'$, $x_1x_2'x_3x_4'$, $x_1x_2'x_3'x_4'$, $x_1'x_2x_3x_4'$, $x_1'x_2x_3'x_4'$ and $x_1'x_2'x_3x_4'$, i.e. $2^{4-1} - 1 = 2^3-1 = 7$ tests can detect the above fault. Now, consider $Y_{15} = x_1'x_2'x_3'x_4$. Let $x_1'x_2'x_3'$ be equal to zero under fault free conditions. The possible fault on this line is s-a-1. To detect this fault x_4 must be equal to one. Therefore, this fault can be detected by d d d x_4 tests except $x_1'x_2'x_3'x_4$ test because under this test there will be no effect on the output and the fault will not be detected. Therefore, this fault can be detected by the following tests:

$x_1x_2x_3x_4$, $x_1x_2x_3'x_4$, $x_1x_2'x_3x_4$, $x_1x_2'x_3'x_4$, $x_1'x_2x_3x_4$, $x_1'x_2x_3'x_4$ $x_1'x_2'x_3x_4$, i.e. $2^{4-1} - 1 = 7$ tests can detect the above faults.

Similar arguments can be made for any other branched s-a-1 faults at any other level.

(C) Formula development for diagnosability:

It has been proven that four tests are required to detect all faults at the first level and they are chosen from the four different sets of tests as follows:

$x_1' x_2' dd \ldots d$
$x_1' x_2 dd \ldots d$
$x_1 x_2' dd \ldots d$
and $x_1 x_2 dd \ldots d$

There are $4,8,\ldots, 2^{n-2}, 2^{n-1}, 2^{n}$ gates per level. It has been proven in theorem 1 that all s-a-o faults on the output can be detected by applying all the possible test vectors. Only four tests are applied to detect all faults at the first level. This means that $(2^{n}-4)$ s-a-o faults cannot be detected at the Lth level by applying four tests. Also, it has been proven in theorem 2 that the s-a-1 faults on the primary bus line at the Lth level can be detected by applying all the possible test vectors. Therefore, $(2^{n}-4)$ s-a-1 faults cannot be detected at the Lth level by applying four tests.

In part 3, it has been proven that the s-a-1 faults on the lines which are numbered as 2 can be detected by applying all even numbered output vectors except the one which has no effect on the output. Also, the s-a-1 faults on the lines which are numbered as 1 can be detected by applying all odd numbered output vectors except the one which has no effect on the output. Therefore, the four tests should be selected in such a way that the s-a-1

faults on the right hand side lines and those on the left hand side lines can be detected. This can be done by selecing two even and two odd numbered tests from the above mentioned tests.

Hence, by applying four tests it is found that $2(2^n-4)$ faults cannot be detected at the Lth level. Let $2^n = G_L$ where G_L = number of gates at the Lth level.

There are 2^{n-1} gates at the (L-1)th level. Now, the question is how many faults cannot be detected at this level by applying four tests. In part 1, it is proven that the s-a-o fault at the (L-1)th level can be detected by applying one of the two adjacent output vectors. Only four tests are applied and therefore $(2^{n-1}-4)$ s-a-o faults cannot be detected at the (L-1)th level. Also, by applying four tests it is not possible to detect $(2^{n-1}-4)$ s-a-1 faults on the primary bus. Hence, $2(2^{n-1}-4)$ faults cannot be detected at the (L-1)th level by applying four tests. Let $2^{n-1}=G_{L-1}$, where G_{L-1} is equal to the number of gates at the (L-1)th level. Similar arguments hold good for other levels.

Also, similar arguments hold good for $8,16,\ldots,2^{n-2}$, $2^{n-1},2^n$ tests the same as with the 4 tests. Hence, in general, it can be concluded that $2(G_L-T)$ faults cannot be detected per level, where G_L represents the number of gates per level L - 1,2,3,...,n-1, and T represents the number of tests applied. Therefore, $F_L = 2(G_L-T)$.

where F_L = a number of faults not detected per level

$L = 1,2,\ldots,n-1$

Example:

Let $n = 4$

$F_L = 2(G_L - T)$

Number of gates at the Lth level is equal to 16 for $n = 4$

First, let $T = 4$

Therefore, $F_3 = 2(16-4)$

$= 24$ faults cannot be detected

$F_2 = 2(8-4)$

$= 8$ faults cannot be detected

$F_1 = 2(4-4)$

$= 0$

Hence, totally 32 faults cannot be detected by applying 4 tests for $n = 4$.

Therefore, % Diagnosability = $\frac{F_{LT}}{F_t} \times 100$

F_t = total number of gates *4

$= 28*4$

$= 112$ faults

Hence,

% Diagnosability = $\frac{112-32}{112} \times 100$

$= 71.43\%$

Now let T = 8

$F_3 = 2(16-8) = 16$ faults cannot be detected

$F_2 = 2(8-8) = 0$

$F_{LT} = 16$

$$\%\text{Diagnosability} = \frac{112-16}{112} \times 100$$

$$= 85.71\%$$

And 100% diagnosability can be achieved by applying all sixteen tests because all faults can be detected.

The following analysis presents the diagnosability for various size simple tree binary address decoders.

Table I:

% Diagnosability for the Number of Tests Applied

No. of inputs	Number of tests applied	% Diagnosability
4	16	100.00%
	8	85.71%
	4	71.43%
5	32	100.00%
	16	86.66%
	8	73.33%
	4	63.33%
6	64	100.00%
	32	87.00%
	16	74.10%
	8	64.50%
	4	58.00%
8	256	100.00%
	128	88.50%
	64	76.00%
	32	66.00%
	16	59.80%
	8	55.50%
	4	53.50%
12	4096	100.00%
	2048	87.90%
	1024	75.00%
	512	65.62%

(con't)

No. of inputs	Number of tests applied	% Diagnosability
	256	59.50%
	128	55.50%
	64	53.10%
	32	51.70%
	16	50.80%
	8	50.50%
	4	50.30%
16	65536	100.00%
	32768	87.60%
	16384	75.20%
	8192	65.60%
	4096	59.50%
	2048	55.50%
	1024	53.10%
	512	51.70%
	256	51.00%
	128	50.60%
	64	50.30%
	32	50.20%
	16	50.15%
	8	50.10%
	4	50.00%

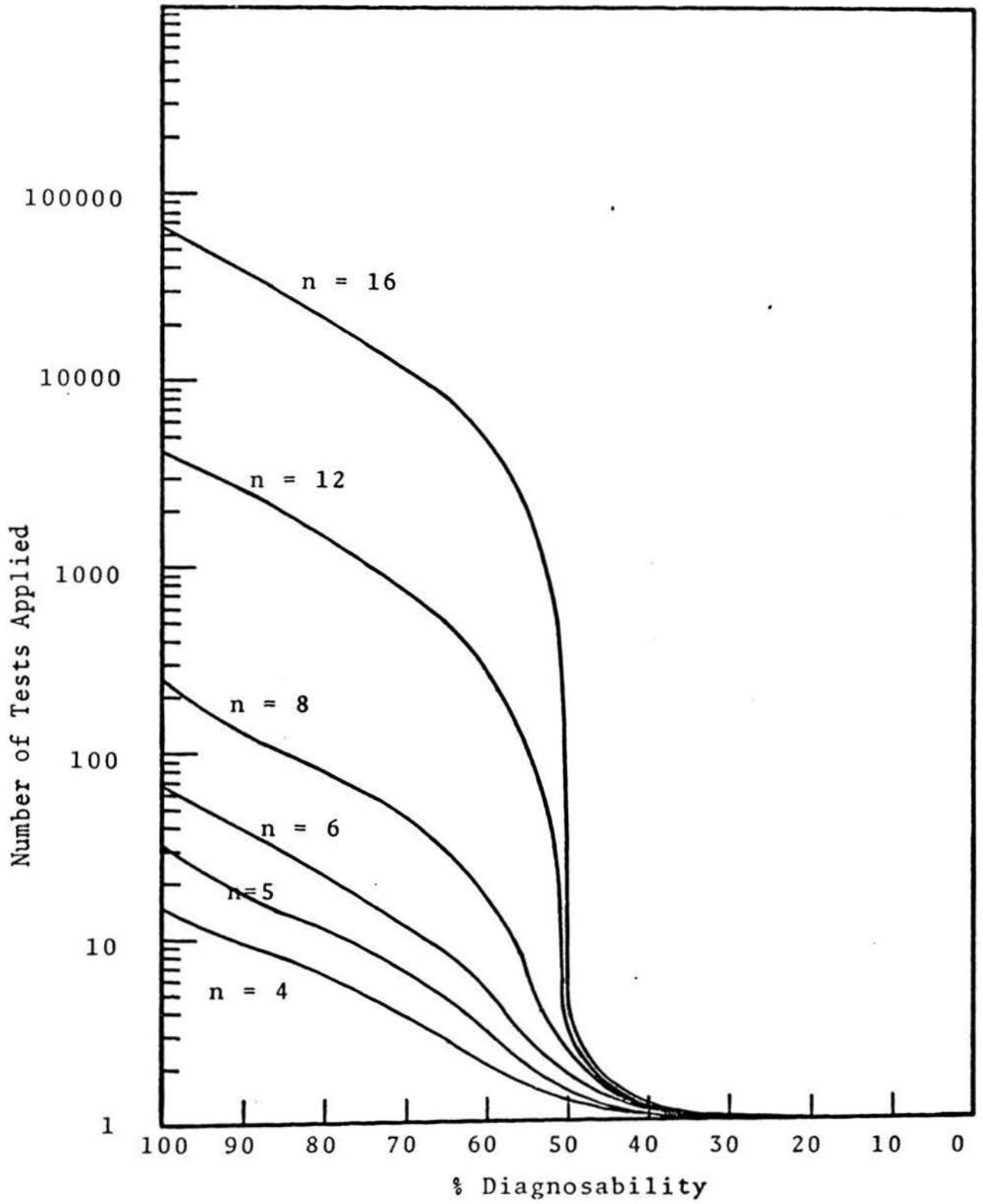


Figure 3: % Diagnosability for the Number of Tests Applied to a Simple Tree Decoder

IV. DUAL TREE DECODER

The dual tree decoder is a different and less costly circuit than the simple tree, although its internal fan-out requirements are higher. The basic idea is to decode the input variables in pairs in one level of circuits, then combine each output of these with every other output to produce the next level of outputs. This process continues until all outputs are derived. A circuit for four input variables is shown in figure 4.

The dual tree may be designed by a symbolic procedure which will now be described and illustrated in figure 5. Let n equal the number of input variables. A diagram containing boxes with a number in each box is constructed according to the following rules (these follow the pair-combination principle described above).

1. The box on the far right is labeled 2^n.
2. Each box (except the last) feeds one and only one box.
3. Work back to successive boxes, each box requiring two inputs. The numbers in the two boxes feeding any given box must be such that
   a. Their product is the label on the given box.
   b. Each number is a power of 2 and as nearly equal as possible.

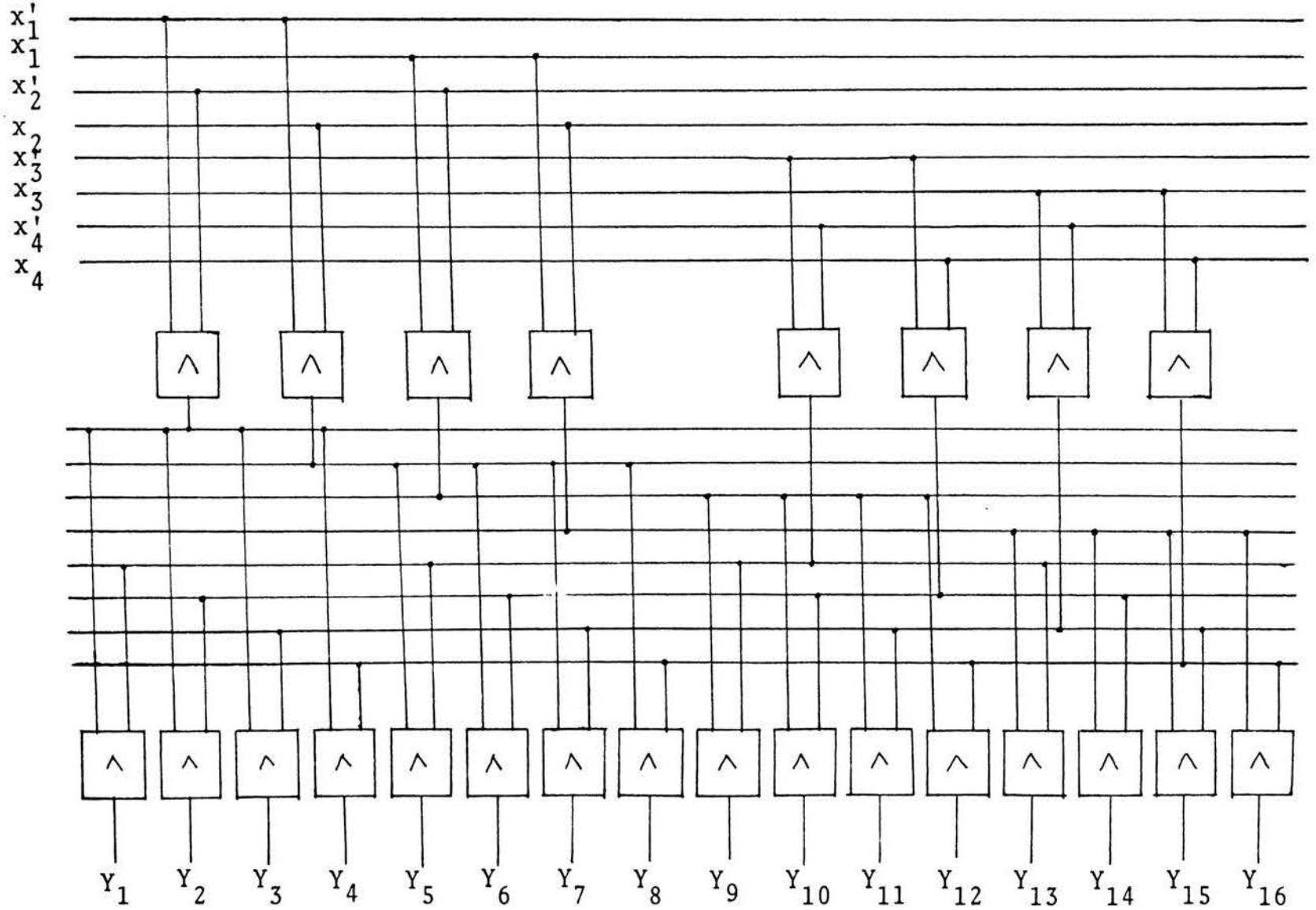


Figure 4 : Dual Tree Decoder for n = 4

There are no boxes labeled 2 since these correspond to inputs.

Each box in the symbolic diagram represents a set of two-input AND circuits. The number of AND circuits in a set is the number placed in the box. Note that this is the product of the numbers in the boxes feeding the box in question. Each symbolic line represents a set of physical lines, each line in the set being a 1 only for one of the combinations of the variables involved. The number of AND blocks is the sum of the numbers in all the boxes; the circuit cost in twice this.

Properties of Dual Tree Decoders:

$$\text{C - Count} \equiv 2n \sum_{j\equiv 1}^{\lceil \log_2 n} \frac{2^{2^j}}{2^j} \quad \text{for n a power of 2.}$$

$$C_{2m} \equiv 2C_m + 2^{2m+1}$$

$$C_{2m+1} \equiv C_m + C_m + 1 + 2^{2m+2};\ C_1 = 0$$

for n is not a power of 2; substitute $1 \leq m \leq n$.

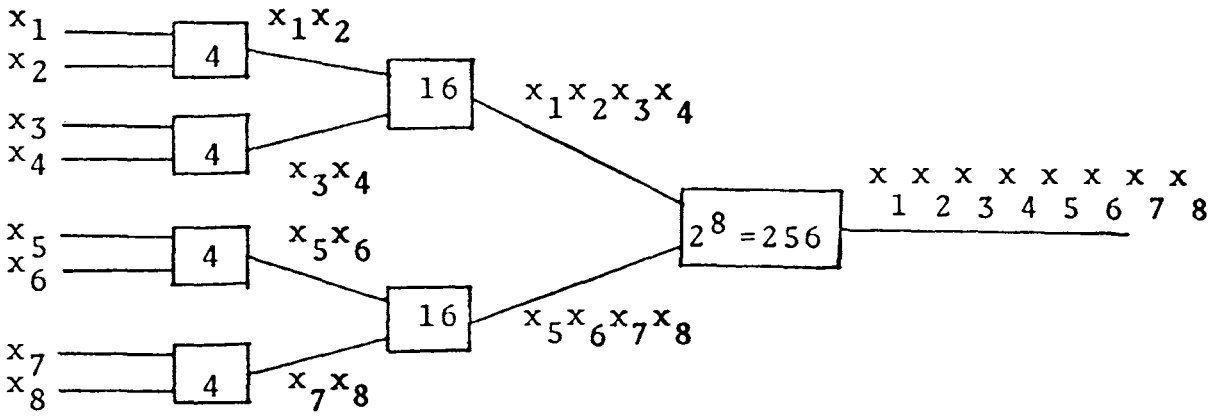
L - levels $\equiv \lceil \log_2 n$

FI - Fan-in $\equiv 2$

FO - Fan-out $\equiv 2^{\lceil n \div 2}$ [internal]
2 for input drivers

Symbolic Design of Dual Tree Decoders

n = 8



Number of AND circuits = 256+16+16+4+4+4+4

= 304

Number of inputs/AND circuits = 2

Cost = 608

Number of levels = $\lceil \log_2 8 \rceil$ = 3

Fan-out = 16 (feeding last level)

n = 9 bits

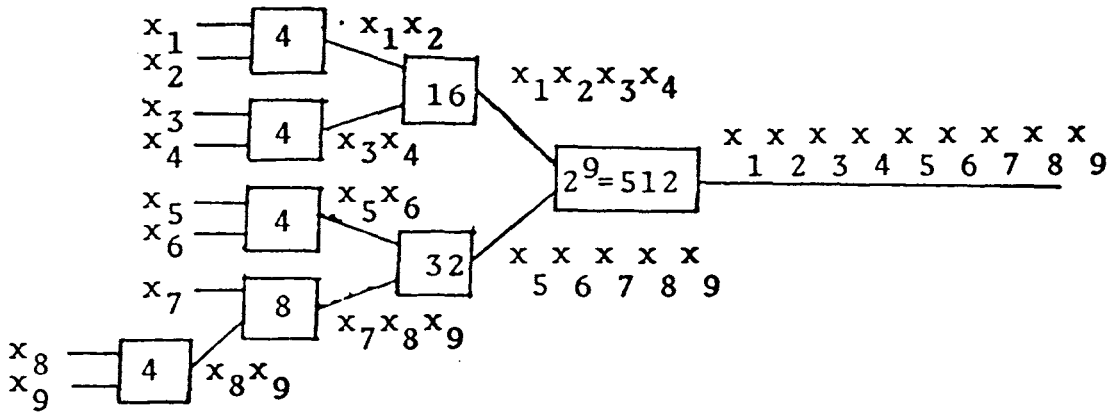


Figure 5: Symbolic design of a dual tree decoder for n = 8 and 9

(A) Number of tests required to detect all faults per level:

From the properties of the dual tree decoder it can be seen that the circuit is symmetrical for n inputs, where n is a power of 2. So first, we will consider the dual tree decoder for the n inputs equal to any power of 2 and then we will consider the other possibilities.

(1) n is a power of 2:

It has been proven in theorem 1 that it is necessary to apply all possible tests (2^n) to detect all s-a-o faults on the output lines. Therefore, in dual tree decoders it is necessary to apply all possible tests to detect all faults at the Lth level.

Now, at the (L-1)th level there are two blocks and each block contains $2^{n/2}$ gates, therefore, there are $2^{n/2}$ output lines per block. It can be seen from Figure 6 that each block has different symbolic output lines at the (L-1)th level. Let one of the blocks have $x_1x_2\ldots x_{m+1}x_{m+2}$ as an output line and the other block have $x_zx_{z+1} \ldots x_{n-1}x_n$ as an output.

Let $x_1x_2\ldots x_{m+1}x_{m+2}$ be equal to one under fault free conditions. The possible fault on this line is s-a-o. This fault can only be detected if the true output is different from the faulty output. To detect this fault, it is necessary to apply $x_1x_2\ldots x_{m+1}x_{m+2}$dd ... d, because under this test all outputs will be zero and the fault can be detected.

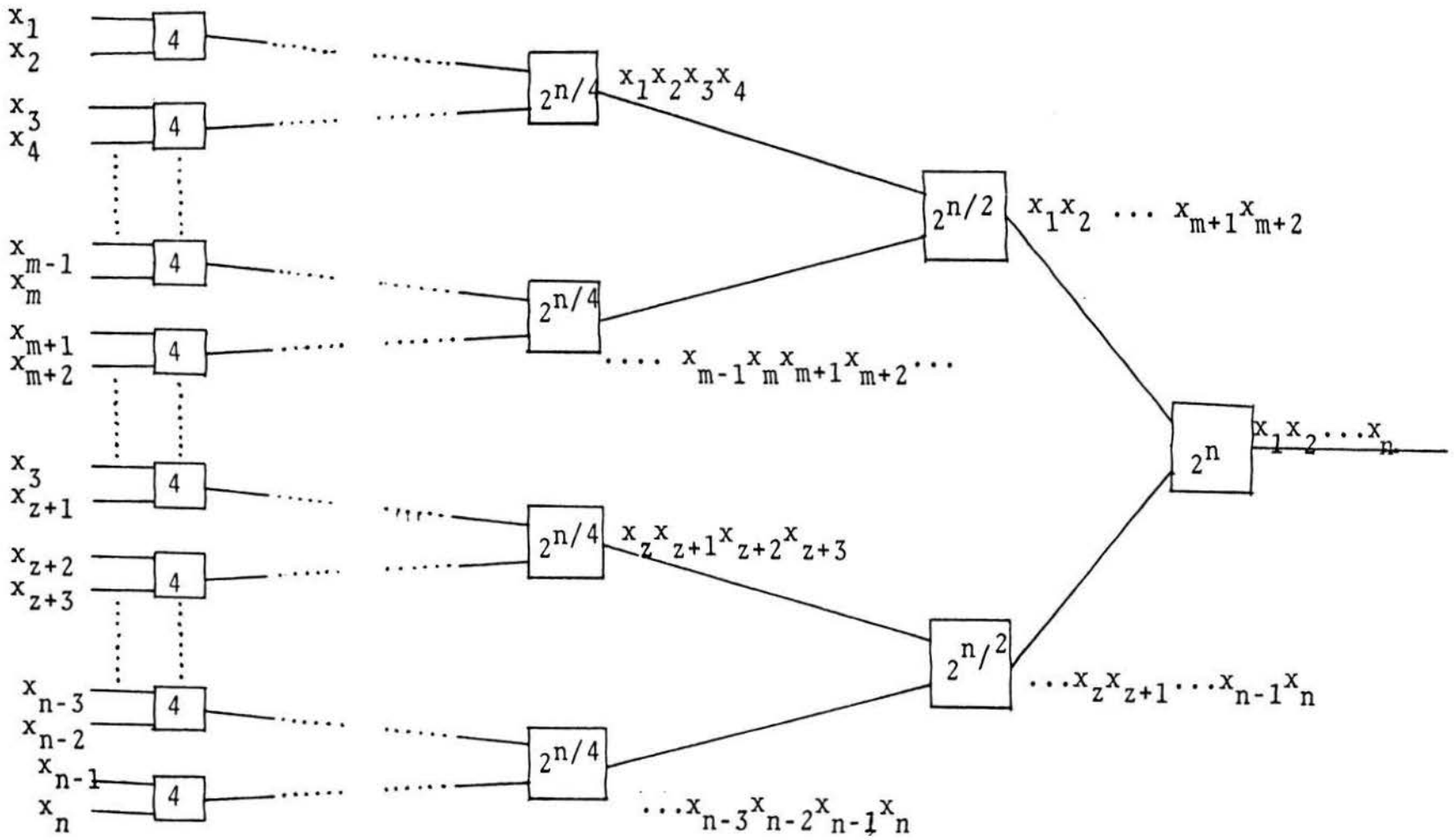


Figure 6 : General Symbolic Diagram of Dual Tree Decoder When n is a Power of 2

Let $x_z x_{z+1} \ldots x_{m-1} x_n$ be equal to one under fault free condition. The possible fault on this line is s-a-o. This fault can be detected by applying dd ... $dx_z x_{z+1} \cdots x_{n-1}$ x_n because under this test all the outputs will be zero.

Combining these two test vectors which will take care of both the s-a-o faults, the resultant test vector obtained is given by $x_1 x_2 \ldots x_{m+2} x_z \ldots x_{m+1} x_n$. Similar arguments hold good for the other output lines at the (L-1)th level.

Therefore, from the above arguments it can be concluded that by combining all the tests with respect to don't cares it can be seen that to detect all faults at the (L-1)th level it is necessary to apply $2^{n/2}$ tests. Similarly to detect all faults at (L-2)nd,..., 1st level, it is necessary to apply $2^{n/4}$,...,4 tests respectively. This can be better explained by taking an example with n = 8.

Example: n = 8

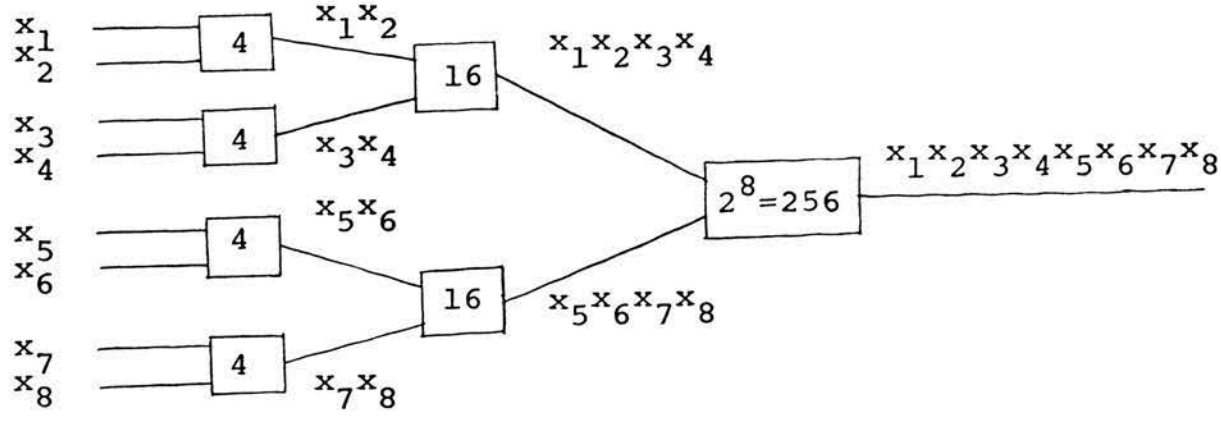


Figure 7: Dual tree decoder for n = 8

At the Lth level, let $x_1 x_2 \ldots x_7 x_8$ be the symbolic output line. Under fault free conditions it is equal to 1. Therefore a possible fault on this line is s-a-o. This fault can only be detected by applying $x_1 x_2 \ldots x_7 x_8$, because under this test all outputs will be zero. Similar arguments hold for other output lines. Hence, to detect all faults at the Lth level it is necessary to apply $2^n = 2^8 = 256$ tests.

Now, consider the symbolic output lines at the (L-1)th level. Let $x_1 x_2 x_3 x_4$ be equal to 1 under fault-free conditions. A possible fault on this line is s-a-o. This fault can be detected by applying $x_1 x_2 x_3 x_4$ dddd, because under this test all outputs will be zero. Other tests have no effect on the output and the fault will not be detected. Consider that $x_5 x_6 x_7 x_8$ is equal to one under fault free conditions. The possible fault on this line is s-a-o. This fault can be detected by applying dddd $x_5 x_6 x_7 x_8$ because under this test all outputs will be zero. Other tests have no effect on the output and the fault will not be detected. By combining these two test vectors, which will take care of both the s-a-o faults; the resultant test vector obtained is given by $x_1 x_2 \ldots x_7 x_8$. Similar arguments hold for the other output lines at the (L-1)th level. There are sixteen possible tests which can detect all faults at this level, i.e. $2^{n/2} = 2^{8/2} = 16$ tests.

Similarly for the first level, $2^{n/4} = 2^{8/4} = 4$ tests will take care of all faults at the 1st level.

s-a-1 Fault Consideration

It has been proven in theorem 1 that it is necessary to apply all possible tests (2^n) to detect all s-a-o faults at the Lth level. These 2^n tests can detect all faults in the circuit. Therefore, the s-a-1 faults at the Lth level can be detected by applying 2^n tests. Similar arguments will hold good for other levels.

Hence, it can be concluded that the s-a-1 faults at the Lth, (L-1)th, (L-2)nd .... 1st level can be detected by applying 2^n, $2^{n/2}$, $2^{n/4}$, ..., 4 tests respectively.

(ii) n is not a power of 2

In figure 8 "$\lfloor$" represents the lower ceiling and "$\lfloor$" represented the upper ceiling. When n is even there will be the same number of gates in both the blocks at the (L-1)th level and when n is odd there will be more gates in one of the blocks than the other. This is true for any other level. At the first level there is either an even or odd number of blocks depending upon the number of inputs applied.

There are two symbolic output lines at the (L-1)th level. Let $x_1, x_2, \ldots, x_z$ be the symbolic line coming from one of the blocks and $x_{z+1} \ldots x_{n-1}\ x_n$ be the symbolic line coming from the other block. Consider $x_1 x_2 \ldots x_z$ as containing one less variable than the vector $x_{z+1} \ldots x_{n-1}\ x_n$. Let $x_1 x_2 \ldots x_z$ be equal to one under fault free conditions. Then the possible fault on this line is s-a-o. This fault can be detected by any test $x_1 x_2 \ldots x_z$ dd...d because under these tests all outputs will be zero. Other than these tests have no effect on the output and the fault will not be detected.

Consider $x_{z+1} \ldots x_{n-1}\ x_n$ being equal to one under fault free conditions. The possible fault on this line is s-a-o. This fault can be detected by any test dd...d $x_{z+1} \ldots x_{n-1}\ x_n$ because under these tests all outputs will be zero. Other than these tests have no effect on the output and the fault will not be detected.

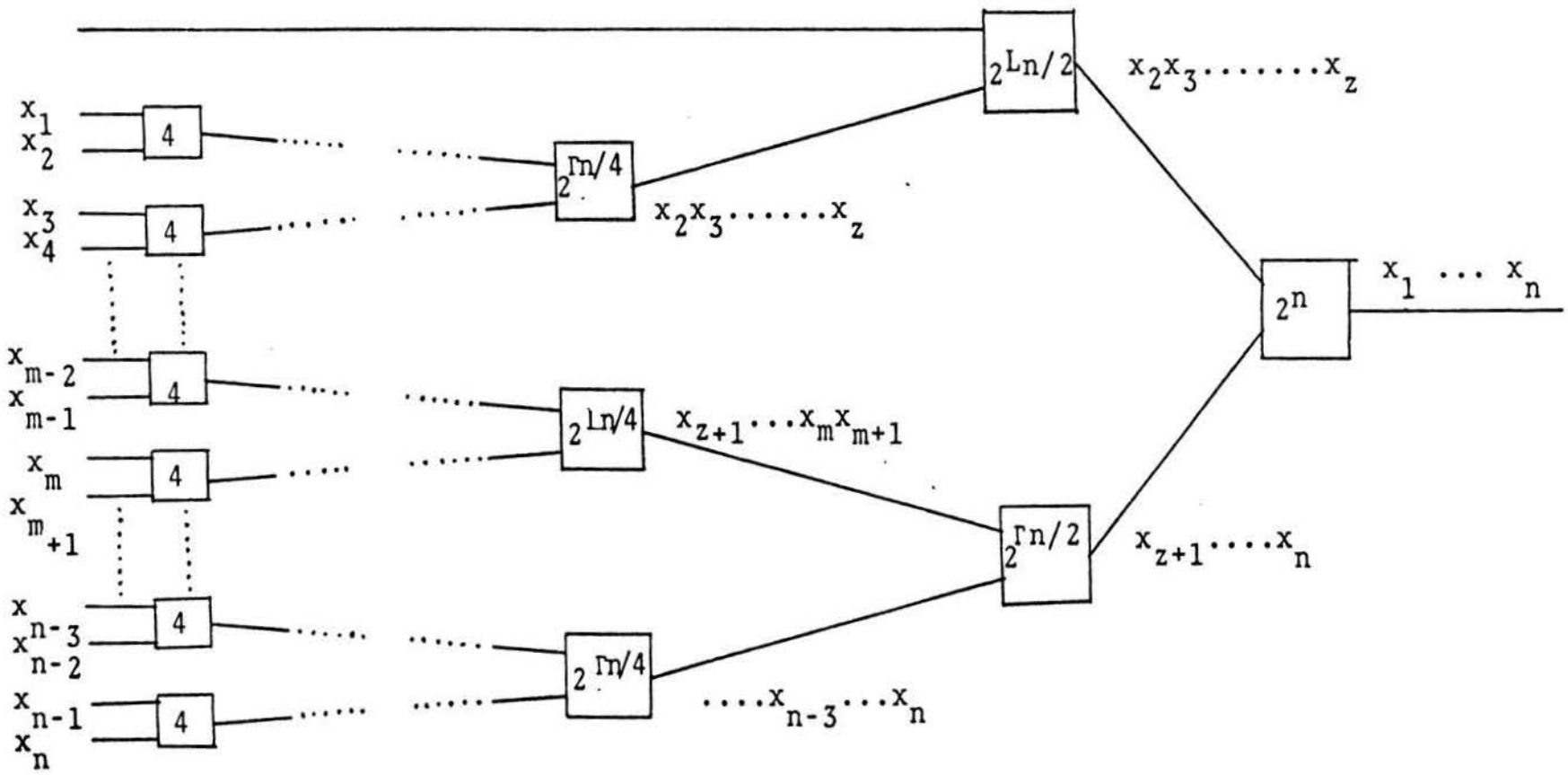


Figure 8. General Symbolic Diagram of Dual-Tree Decoder When n is not a Power of 2

By combining the above two test vectors, which will take care of both the s-a-o faults, the resultant test vector is given by $x_1 x_2 \ldots x_z x_{z+1} \ldots x_{n-1} x_n$. Similar arguments can be made for other lines at the (L-1)th level. Hence, to detect all faults at the (L-1)th level it is necessary to apply $2^{\lceil n/2}$ test vectors.

Similarly to detect all faults at the (L-2)nd, (L-3)rd, ..., 1st level, it is necessary to apply $2^{\lceil} n/4, 2^{\lceil} n/8, \ldots, 4$ tests respectively. This can be better explained by taking an example with n = 5.

Example: n = 5

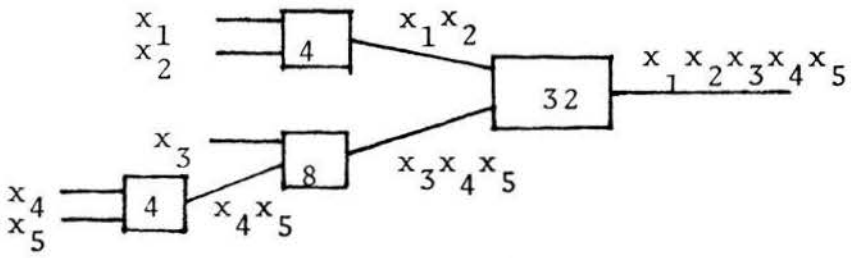


Figure 9: Dual Tree Decoder for n = 5

At the (L-1)th level there are two blocks consisting of four and eight gates. Let $x_1 x_2$ be the symbolic line from one of the blocks and let it be one under fault free conditions. The possible fault on this line is s-a-o. This fault can be detected by applying and test $x_1 x_2$ ddd because under these tests all outputs will be zero. Other

than these tests have no effect on the output and the fault will not be detected. Consider $x_3x_4x_5$ be equal to one under fault free condition. The possible fault on this line is s-a-o. This fault can be detected by applying $ddx_3x_4x_5$ tests, because under these tests all outputs will be zero. By combining the above two test vectors, which will take care of both s-a-o faults, the resultant test vector is given by $x_1x_2x_3x_4x_5$. Similar arguments hold good for other lines. Hence, to detect all faults at the (L-1)th level it is necessary to apply eight tests.

i.e. $2^{\Gamma n/2}=2^{\Gamma 5/2}=2^3=8$ tests.

Let x_4x_5 be the symbolic output line at the Lth level, let it be one under fault free conditions. The possible fault on this line is s-a-o. This fault can be detected by applying $x_1x_2x_3dd$ tests because under these tests all outputs will be zero. Other than these tests have no effect on the output and the fault will not be detected. Similar arguments hold good for other three lines. Hence, at least four tests are necessary to detect all faults at the 1st level.

i.e. $2^{\Gamma n/4}=2^{\Gamma 5/4}=2^2=4$ tests.

Diagnosability

An attempt was made to develop a theory of diagnosability in an analogous manner to that developed for the simple tree decoder. Due to the lack of symmetry for various input conditions this approach was not as successful as hoped. Fragments of a theory were developed for the various input conditions. However, these were not of a significant enough nature to be presented herein. Its complexity far outweighed any aesthetic quality. For these reasons, the developed material was used as an aid to the hand computation of diagnosability for the cases of interest. These results are presented in Table II. A graph of these results is also presented in Figure 10.

TABLE II

% Diagnosability Against the Number of Tests Applied

No. of inputs "n"	No. of tests applied	% Diagnosability
4	16	100.00%
	8	91.10%
	4	79.30%
5	32	100.00%
	16	91.60%
	8	883.40%
	4	68.60%
6	64	100.00%
	32	91.70%
	16	86.60%
	8	79.10%
	4	58.00%
8	256	100.00%
	128	89.40%
	64	84.00%
	32	81.20%
	16	77.60%
	8	56.1%
	4	43.2%
12	4096	100.00%
	2048	88.00%
	1024	82.60%
	512	79.20%

Table II (continued)

No. of inputs "n"	No. of tests applied	% Diagnosability
	256	77.40%
	128	76.80%
	64	75.60%
	32	63.50%
	16	49.30%
	8	33.33%
	4	29.80%
16	65536	100.00%
	32768	87.30%
	16384	82.00%
	8192	78.60%
	4096	76.40%
	2048	75.20%
	1024	75.10%
	512	75.00%
	256	74.10%
	128	63.90%
	64	49.50%
	32	39.10%
	16	29.20%
	8	26.00%
	4	25.40%

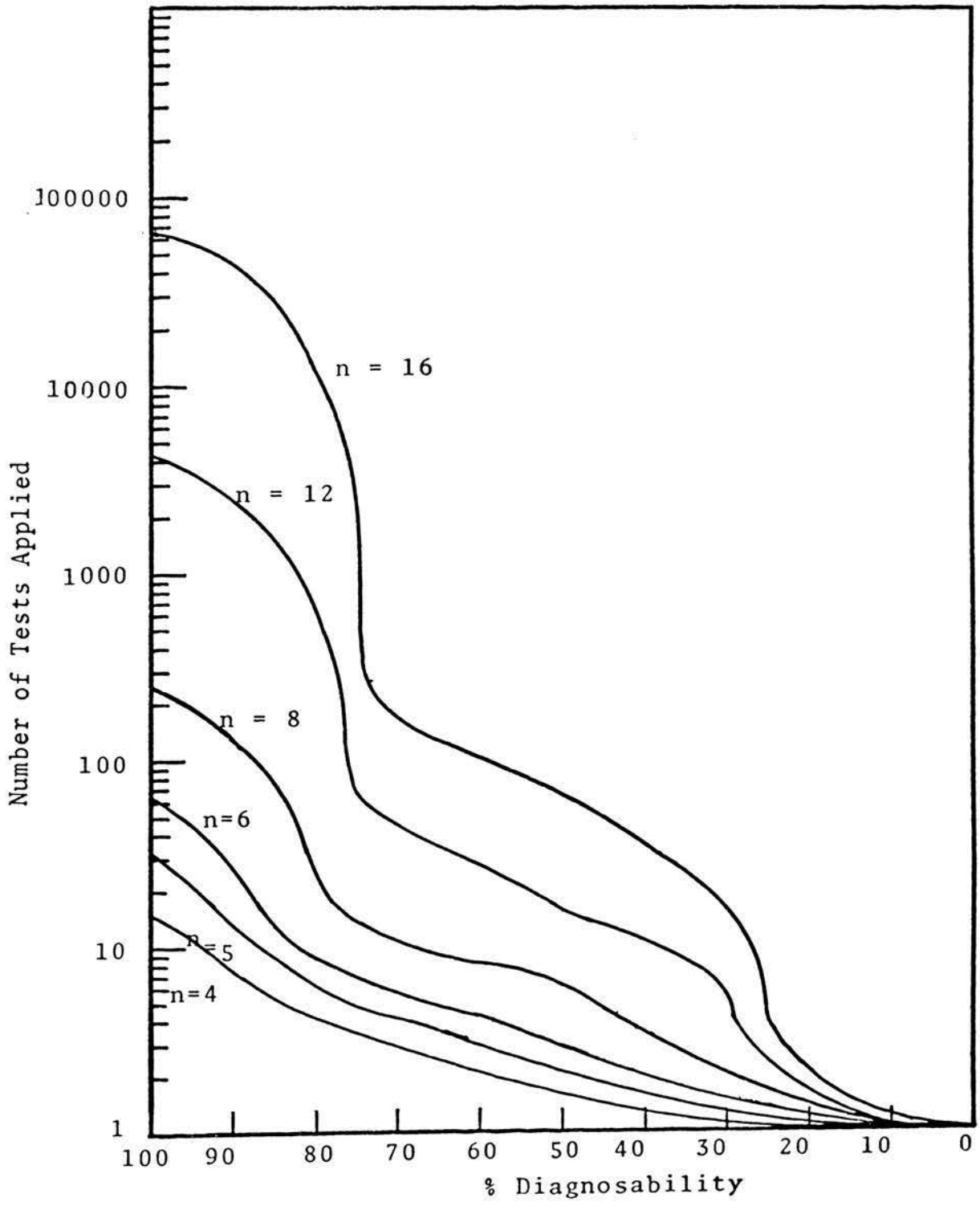


Figure 10: % Diagnosability for the Number of Tests Applied to a Dual Tree Decoder

V. CONCLUSIONS

This study has considered single logical faults of the s-a-1 and s-a-o type, for simple and dual tree binary address decoders. It has been demonstrated that all faults per level can be diagnosed by the following two results:

1. All faults per level in a simple tree binary address decoder can be detected by applying $4,8,16,\ldots,2^{n-2},2^{n-1},2^{n}$ tests.

2. All faults per level in a dual tree binary address decoder can be detected by applying $4,\ldots,2^{\lceil n/4}$, $2^{\lceil n/2}$, 2^{n} tests.

The % diagnosability characteristics for simple and dual tree binary address decoders were plotted in Figures 3 and 10 respectively. From these characteristics it can be concluded that in a simple tree binary address decoder the % diagnosability falls rapidly up to the point when $2^{n/2}$ tests are applied, then the decrease becomes gradual. But in the case of dual tree binary address decoder up to the point of $2^{n/2}$ tests it falls gradually and immediately after it falls rapidly.

REFERENCES

1. "Digital Computer System Principles", Herbert Hellerman, McGraw-Hill Cook Company, G7-14671, 1967, pp. 195-201.

2. "Analysis and Synthesis of Self Repairing Techniques for Digital Computers", S.A. Szygenda, Ph.D. dissertation, Northwestern University, 1968.

3. "Failure Analysis of Binary Address Decoders", M.S. Thesis, K. Gandhi, University of Missouri-Rolla, 1969.

4. "Fault Testing and Diagnosis in Combinational Digital Circuits", William H. Kautz, IEEE Trans. of Computers, April 1968, pp. 352-367.

5. "On Finding a Nearly minimal Set of Fault Detection Tests for Combinational Logic Nets", D.B. Armstrong, IEEE Trans. on Electronic Computers, Vol. EC-15, No. 1, February 1966, pp. 66-73.

6. "An Algorithm for Selecting an Optimum Set of Diagnostic Tests", Herbert Y. Chang, IEEE Trans. of Electronic Computers, Vol. EC-14, No. 5, October 1965, pp. 706-711.

7. "Diagnosis of Large Combinational Networks", V. Amar and N. Condulmari, IEEE Trans. on Electronic Computers, October 1967, pp. 675-679.

8. "Derivation of Optimum Tests to Detect Faults in Combinational Circuits", J.F. Poage, Proc. Symp. on Mathematical Theory of Automata, New York: Polytechnic Press, 1963, pp. 483-528.

VITA

Girishchandra Mahendrakumar Gandhi was born on January 16, 1945 in Ahmedabad, India. He received his primary and secondary education in Ahmedabad. He received a Bachelor of Science Degree in Electrical Engineering from Birla Vishvakarma Engineering College in June 1967.

He has been enrolled in the graduate school of the University of Missouri-Rolla since September, 1968 and has been a graduate assistant during 68-69.