INSTITUTE OF TECHNOLOGY

AIR UNIVERSITY

UNITED STATES AIR FORCE

SCHOOL OF ENGINEERING

THESIS

WRIGHT-PATTERSON AIR FORCE BASE, OHIO

AF-WP-B-SEP 60 5M

ASYNCHRONOUS DIGITAL LOGIC CIRCUITS USING NEURISTORS

THESIS

Presented to the Faculty of the School of Engineering of
the Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

BY

Andreus August Piske, Jr., B.S.
1/Lt USAF

Graduate Astronautics

August 1961

Preface

This thesis is the result of my investigation of the application of neuristors to asynchronous digital logic circuits. My investigation commenced with a study of the neuristor, continued with a study of digital logic principles, and concluded with an attempt to construct specific asynchronous circuits while looking for a general method for circuit construction.

The symbology and terminology that have been used in this thesis are common, in general, to texts treating of digital logic circuits or Boolean algebra.

I am indebted to my thesis advisor, Professor J. Lubelfeld for having introduced me to the subject of digital logic circuits and neuristors. Further, I am grateful for the assistance and guidance given me by Captain F. M. Brown. Finally, I wish to acknowledge my indebtedness to my wife for her typing of both the preliminary draft and the final report.

Andreus A. Piske, Jr.

Contents

Contents

List of Figures

List of Figures

Abstract

The neuristor is an active device which acts as a one-dimensional channel along which a pulse may propagate at a constant velocity. Neuristors can be joined in two basic junction types: T and S junctions. A T-S junction is a combination of the two basic junction types. Using these three junction types, "and", "or" and "not" circuits can be constructed. These basic circuits can be combined with a variable "tree" to construct any arbitrary combinational circuit. The neuristor has "built-in" delay. Sequential circuits can be analysed as delay devices and reduced to combinational circuits, which can be constructed using neuristors.

GA/EE/61-3

ASYNCHRONOUS DIGITAL LOGIC CIRCUITS USING NEURISTORS

I. Introduction

Purpose

The purpose of this independent study was to investigate the use of neuristors in the construction of asynchronous digital logic circuits. Specifically, the primary objective of this study has been the development of a formalized method for the design and the construction of asynchronous sequential circuits.

A neuristor is an active device which acts as a one-dimensional channel along which a signal in the form of an electrical pulse or group of pulses (pulse train) may propagate (Ref 1:iii). An active device (here differentiated from a passive device) is an energy source capable of initiating or generating an electrical pulse rather than merely "carrying" a pulse once it has been generated. The pulse exhibits certain properties which are enumerated in Chapter II of this report.

The class of asynchronous digital logic circuits consists of all those digital logic circuits in which

clock pulses are not employed; the circuits are "free-running" (Ref 2:6).

Historical Background

The neuristor is a hypothesized device conceived by H. D. Crane. The conception of the neuristor was stimulated by a consideration of the question: "Is it possible to build an electronic computer in an environment in which good conductors, or 'wires', are not available for carrying signals?" (Ref 1:iii).

Crane has demonstrated that any synchronous sequential digital logic circuit can be constructed in a plane using only neuristors (Ref 1:116-120). Further, he has designed a specific example of an asynchronous computer (Ref 1:171-190). He has not, however, attempted to develop a formalized method for designing asynchronous sequential circuits.

The two-dimensional property of neuristor networks, that is, the property that permits the construction of networks in a plane, has been emphasized by Crane in his thesis (Ref 1:118-120). This two-dimensional character is maintained through the use of control-crossings (see figure 2.5). In this report an attempt has been made to construct only planar networks and to maintain the two-dimensional character of neuristor networks.

A sequential circuit may be analysed as a delay device. A method exists, based on this method of analysis, which can be used for the design of sequential circuits (Ref 2:6). Herein lies one of the advantages of using the neuristor for sequential circuit design: the neuristor is inherently a delay device--it has "built-in" delay characteristics.

Delay in a neuristor is measured directly in terms of physical path length. This property makes the neuristor ideally suited for asynchronous circuits since "race" conditions can easily be avoided.

The neuristor has not as yet been physically realized, but its feasibility has been demonstrated, and methods for its realization have been proposed (Ref 1:8-24).

Scope

This report is concerned with the construction of both combinational circuits and sequential circuits. The construction and use of the basic combinational circuits are considered. Only those sequential circuits for which the state of the circuit is a function of a finite number of past inputs are considered.

Since no neuristor exists at present, the basic

premise of this study was that the neuristor would be developed with those properties specified in the definition of the neuristor. The property of constant propagation velocity is especially pertinent to the whole development of asynchronous circuits.

Plan of Report

The remainder of this report consists of four chapters. Chapter II treats primarily of the neuristor and its basic junction types. In Chapter III the basic combinational circuits are developed, and their use in the construction of an arbitrary circuit is demonstrated. Six specific sequential design problems are considered and a general method for circuit construction is proposed in Chapter IV. In Chapter V the results of the investigation are stated and the recommendation for further study is presented.

Preliminary Statement of Results

The primary result of this study has been the development of a method for the design and construction of asynchronous sequential logic circuits. This method has been demonstrated through the consideration of six specific design problems.

A secondary result has been the construction of the

basic combinational circuits, "and", "or" and "not". The use of these basic circuits in the construction of an arbitrary combinational circuit has been demonstrated.

II. Neuristor Principles

Definition of Neuristor

Crane has defined a neuristor as "a device having the form of a one-dimensional channel along which signals may propagate, the signals taking the form of propagating discharges having the following properties:

1. Threshold stimulability
2. Attenuationless propagation
3. Uniform velocity of propagation
4. Refractive period following the passage of a discharge past any point of a channel (during this period, that portion of the channel cannot propagate a second discharge)" (Ref 1:5).

The neuristor belongs to a class of devices which includes the human nerve fiber, or neuron. A consideration of the characteristics of the neuron may lead to a better understanding of the operation of the neuristor.

If a small electric probe is used to impress a D. C. voltage on a human nerve fiber, a discharge process occurs after a certain voltage has been reached. This discharge (hereafter also referred to as a signal or a pulse) propagates along the fiber in both directions away from the point of probe application. (The fact that a certain voltage must be reached before the discharge

occurs results from the property of threshold stimulability of the nerve fiber.) The discharge propagates at a constant velocity and is unattenuated. As the signal proceeds along the fiber it releases, or discharges, the energy stored at each point. There is attenuationless propagation because the released energy replaces the energy dissipated due to transmission losses. Once a discharge has been initiated along a nerve fiber, another discharge cannot be initiated until after a certain time period has elapsed. This time period is the refractory period (Ref 1:2-3).

The refractory period of a neuristor results from a loss of the stored energy to a discharge as it propagates. This refractory period is a function of the time required to restore to the neuristor the energy that has been lost to the discharge (Ref 1:2-3).

A refractory distance, D_R, can be associated with a refractory period, T_R. The fundamental relationship $D_R = VT_R$, where V is velocity of propagation, can be applied to determine the refractory distance. This refractory distance is the length of neuristor that is in the relaxed state immediately behind a discharge and is the minimum spacing between pulses.

In accordance with Crane's usage the term "line" will hereafter be used synonymously with neuristor.

Neuristor Variables

Two variables are related to the propagation of a discharge along a neuristor, namely, the trigger strength, or trigger variable, and the energy storage level, or storage variable. These variables are so related that at any instant and at any position along a line the change with respect to time of either "generates" the other. (This relationship is analogous to the relationship of the electric field and magnetic field in linear transmission of electromagnetic radiation.) A signal can be visualized as propagating as a result of an alternate action of energy discharge which causes an increase in trigger strength which causes another energy discharge. This alternate action continues as the signal propagates (Ref 1:21).

Neuristor Junctions

Neuristors can be interconnected in two basic junction types. If the lines are coupled in their trigger variables, the junction is called a T junction; if they are coupled in their storage variables, the junction is called an S junction. A third junction type is a

combination of the T and S junctions and is called a T-S junction.

The T Junction. The T junction is illustrated in figure 2.1. Three lines are joined in a T junction, indicated by a dot. If a pulse A propagates on line 1 to the T junction, it triggers lines 2 and 3 and pulses A' and A" propagate away from the junction on these lines.

In general, if n lines were joined to line 1 at a single T junction, pulse A would trigger all n lines upon reaching the junction, and a pulse would propagate on each line away from the junction. Also, a pulse arriving at the T junction on any of the n + 1 lines triggers the n remaining lines and a pulse propagates away from the junction on each of these n lines (Ref 1:27-29).

The S Junction. An S junction is a region in which two neuristors are coupled in their storage variables, that is, a region in which two lines share a common energy source. The entrances or ends of the S junctions are the points at which this common energy source begins or terminates.

The symbol for the S junction, as shown in figure 2.2, consists of three parallel marks drawn between two neuristors. Pulse B on line 1 discharges the stored energy

of this line as it propagates. As it passes over the S junction it releases the stored energy here and by doing so causes both lines to be relaxed for the refractory period. Pulse B continues on line 1 after passing over the S junction. The consequence of causing line 2 to relax is that if pulse B' arrives at the right end of the S junction within one refractory period of the time that B passed there, no energy is available at this point to support B' and it simply disappears.

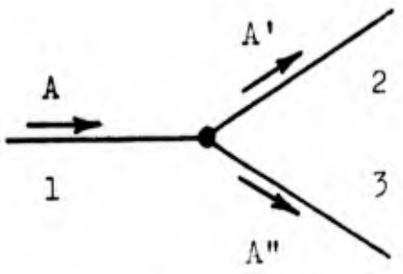


Fig. 2.1
T Junction

(From Ref 1:41)

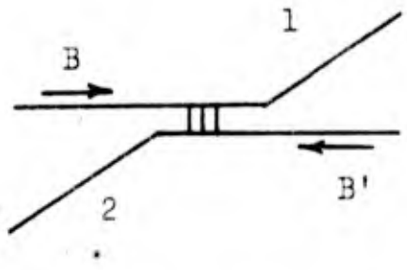


Fig. 2.2
S Junction

(From Ref 1:41)

Now consider the situation where pulses B and B' arrive simultaneously at the left and right ends, respectively, of the S junction. Each discharges the common

energy source as it propagates toward the center where the pulses meet. Now each line is in a relaxed state on either side of the center of the S junction, no energy is available for further propagation of the pulses, and they both disappear. It should be noted that mutual annihilation of two pulses can be assured if the pulses arrive simultaneously at opposite ends of the S junction (Ref 1:29-32).

The T-S Junction. The combination of a T and an S junction is shown in figure 2.3. If a pulse A arrives at the T-S junction along line 1, it triggers lines 2 and 3.

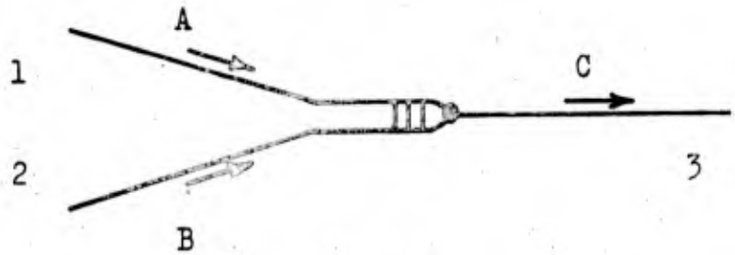


Fig. 2.3

T-S Junction

(From Ref 1:41)

Because pulse A has caused line 2 to relax in the region of the S junction, line 2 cannot support a pulse and only

pulse C propagates away from the junction on line 3. A pulse B arriving at the T-S junction along line 2 similarly causes only a pulse C on line 3.

Now should pulses A and B both arrive at the T-S junction, three possible situations exist. If the pulses arrive at least one refractory distance apart, two pulses will be generated on line 3. If the pulses arrive within one refractory distance of each other, only the first pulse will ever reach the T part of the junction, and one pulse will result on line 3. And finally, the situation could exist where pulses A and B arrive simultaneously at the beginning of the T-S junction. This simultaneous pulse arrival situation will be avoided in the use of the T-S junction in this paper, and will not be considered here (Ref 1:34-36).

Pulse Annihilation

The two methods of pulse annihilation using neuristors are:

1. Mutual annihilation of two pulses moving in opposite directions on an S junction
2. A pulse running off the end of an open line.

The first method has been discussed previously. The second method means that a pulse reaching the end

of an open line disappears. This will prove a convenient method of pulse annihilation in certain applications.

Delay

An intentional or a desired delay in a line will be indicated by one of the symbols in figure 2.4. This delay can of course be achieved merely by adding into the circuit the proper amount of line. The unspecified delay is used in instances where the physical dimensions of the circuit determine the amount of delay required, as in the basic combinational circuits. The specific delay symbol is used in circuits where the amount of delay is a function of the spacing between pulses representing the circuit control variables, as in the sequential circuits.

Controlled-Crossing

Crane's controlled-crossing structure is shown in figure 2.5. This circuit permits the planar crossing of two lines. Pulses propagating from A to B or from C to D do not mutually interfere provided that the minimum time interval at which these pulses arrive at the structure is the "settling time" of the circuit. After a pulse propagating from C to D enters the structure (arrives at junction T_1), before a pulse can arrive at S_1

(propagating from A to B) all lines in the structure must have returned to their rest state. This means that the last pulse (of the pulses generated as a result of the arrival of the pulse at T_1) to disappear from the structure, must have disappeared at least one refractory period before the pulse arrives at S_1.

The symbol for the controlled-crossing will be a circle around the crossing lines.

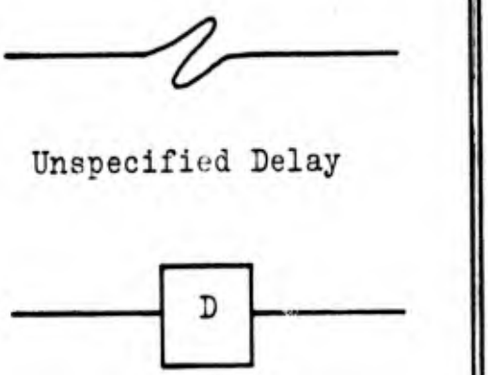


Fig. 2.4
Delay Symbols

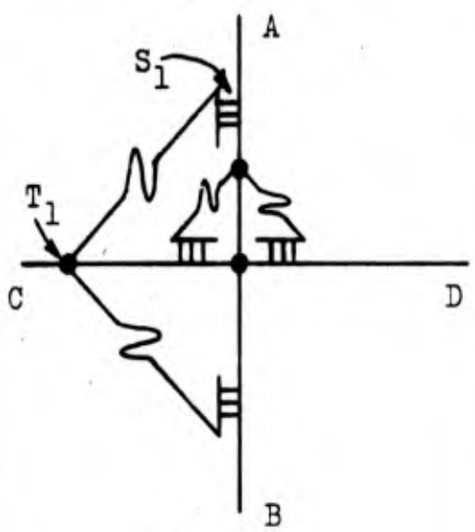


Fig. 2.5
Controlled-Crossing Structure

(From Ref 1:51)

III. Combinational Logic Circuits

A general method for the construction of combinational circuits is developed in this chapter. First the basic circuits are constructed using appropriate neuristor networks.

Basic Circuits

The basic transmission functions "and", "or", and "not" are developed in this section. A "mod 2" circuit is developed as a result of the "not" circuit configuration.

The "And" Circuit. The transmission function (designated by TF) "and" is realized by a circuit which yields an output pulse if and only if a pulse enters on each input line.

The "and" circuit for three control variables is shown in figure 3.1. For the proper operation of this circuit, all inputs representing the control variables at any particular time must enter simultaneously.

Assume that all lines present pulses at the inputs. Pulses X and Y generate pulses X', X", Y' and Y" at the T junctions. Pulses Z and Y' enter simultaneously at opposite ends of junction S_1 and are annihilated.

Pulse Y" proceeds to junction S_2 where it and pulse X' enter simultaneously at opposite ends of junction S_2 and are annihilated. Pulse X" proceeds on to the output.

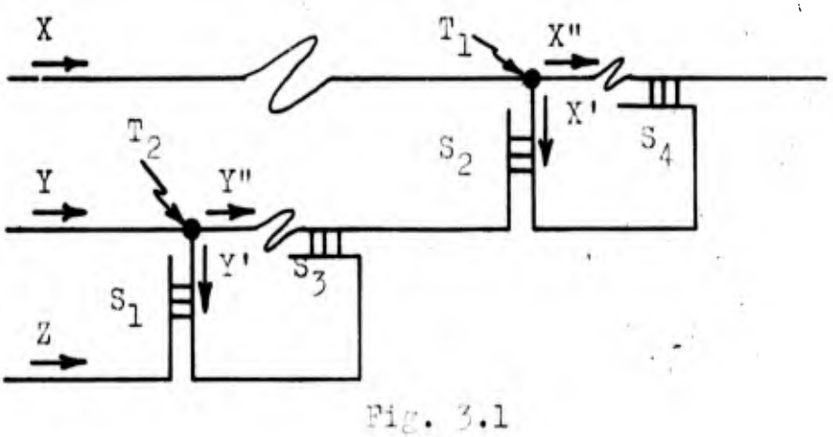


Fig. 3.1

Three-Control-Variable "And" Circuit
(TF = XYZ)

Now consider what occurs if pulses enter on Y and Z but no pulse enters on X. Although pulse Y" proceeds to S_2, it disappears off the end of the line, and no pulse gets to the output.

If pulses enter on X and Z but no pulse enters on Y, pulse Z disappears off the end of the line at S_1. Pulse X generates pulses X' and X", but X' is not destroyed on S_2 and proceeds to S_4. Pulses X' and X" enter simultaneously at opposite ends of S_4 and are annihilated. There is no output pulse.

There are a total of 2^3, or eight, possible input pulse combinations for the circuit with three binary control variables. The remaining cases will not be considered, but an analysis of each can be made that is similar to that made for the three cases discussed here.

The three unspecified delays indicated in figure 3.1 are necessary to achieve pulse synchronization on the S junctions. Two such synchronizations are required on each section of the circuit involving two adjacent control variables. On the Y-Z section, for example, in order that pulses Y and Z arrive simultaneously at opposite ends of S_1, the length of line, or distance, from the Y input to the upper entrance of S_1 must equal the distance from the Z input to the lower entrance of S_1. The distances from T_2 to both entrances of S_3 must also be equal. Analogous distance requirements exist for the X-Y section of the circuit.

This "and" circuit can easily be extended to four, five, or n control variables by the addition of sections either above X or below Z. The only consideration must be the delay time for the complete operation of the circuit.

The delay time of this circuit is the time required for a pulse to propagate from the X input to the output,

since all other lines terminate prior to reaching the output. The delay time of an "and" circuit is normally a function of the number of control variables. Although two "and" circuits may have the same number of input variables, if the vertical spacing between the input lines in the first circuit is greater than that of the second, the delay time of the first circuit will be greater than the delay time of the second.

The "Or" Circuit. The transmission function "or" is realized by a circuit which yields an output pulse if a pulse enters the circuit on at least one input line.

The "or" circuit is shown in figure 3.2. The proper operation of the "or" circuit depends on the simultaneous entrance of all inputs.

Assume that all lines present pulses at the inputs. Pulses X and Y generate pulses X', X", Y', and Y". Pulses Z and Y' are annihilated on S_1. Pulse Y" generates Y"'. But Y"' and X' are annihilated on S_2. Pulse X"', generated by pulse X", proceeds to the output.

Now consider the case where pulses enter on X and Y but no pulse enters on Z. Pulses Y', Y", Y"', X', X", and X"' are generated as before, but pulses Y"' and X' are annihilated on S_2, and X"' propagates to the output.

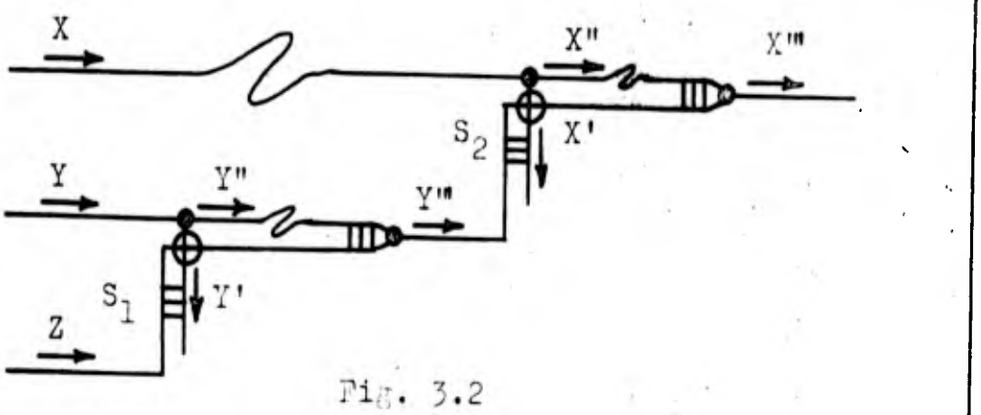


Fig. 3.2

Three-Control-Variable "Or" Circuit
(TF = X + Y + Z)

The three combinations which include an input pulse on only one input line are fairly obvious. If a pulse enters on any one line, it generates pulses which ultimately result in the generation of X''' which proceeds to the output.

As was the case for the "and" circuit discussed previously, there are eight possible input combinations for the "or" circuit. The remaining cases will not be discussed.

Three unspecified delays are shown in the circuit diagram for the "or" circuit. In this circuit, as in the "and" circuit, pulse synchronization on all S junctions

must be obtained. Two synchronizations are again required on each section of the circuit. On the X-Y section, for example, the distance from the Y input to the lower entrance of S_2 must be equal to the distance from the X input to the upper entrance of S_2. Also, the distances from both the X and Y inputs to the T junction where X"' is generated must be equal. Analogous distance requirements exist for the Y-Z section of the circuit.

The "or" circuit can be extended to any number of control variables, the only consideration again being the delay time of the circuit. The delay time for this circuit is the time required for a pulse to travel from any input to the output. The statements made about the variation in delay time of "and" circuits apply also to "or" circuits.

It should be noted that through the use of Crane's controlled-crossing at the points above junctions S_1 and S_2 the important two-dimensional structure of neuristor logic circuits is preserved. It is possible to use the controlled-crossing at these two crossover points because pulses will never simultaneously appear at these points. Pulses propagate in one direction only on the two lines joined by both controlled-crossings.

The "Not" Circuit. The last basic circuit to be discussed in this chapter is the "not" circuit shown in figure 3.3. The "not" circuit is to yield a pulse at the output whenever the control variable, in this case X, is not a one.

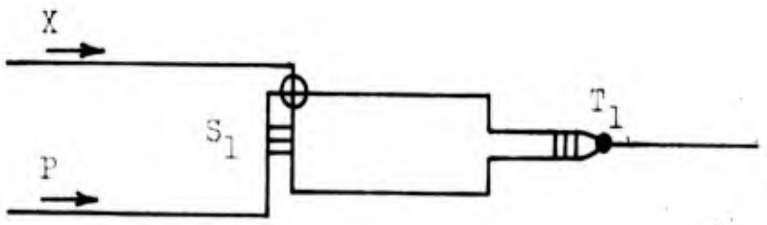


Fig. 3.3

"Not" Circuit ($TF = \overline{X}$)

A requirement for the operation of this circuit is the presence of a pulse, P, which arrives at the input simultaneously with the X input.

There are two distance requirements for the "not" circuit. First, the distance from the X input to the upper entrance of S_1 must equal the distance from the P input to the lower entrance of S_1. Second, the distances from both inputs to T_1 must be equal, this common distance determining the delay time of the circuit.

Each time an input arrives at the upper entrance

to S_1 a pulse must simultaneously arrive at the lower entrance of S_1. If the X input is a pulse, that is, X = 1, the two pulses will be annihilated on S_1 and the output will be a zero, or $\overline{X}$. If X = 0 the pulse coming from P will reach T_1 and generate an output pulse, again representing $\overline{X}$.

A controlled-crossing at the point above S_1 maintains the two-dimensional circuit structure.

The "Mod 2" Circuit. The "not" circuit can be modified to form a "mod 2" circuit, that is, a circuit for which the output is the sum, modulo-two, of the inputs. If the P pulse input is replaced by a control-variable input, the transmission function for the circuit is the sum, modulo-two, of the inputs. (The symbol $\oplus$ represents sum, modulo-two.) For example, if the P pulse input in figure 3.3 is replaced by the input of control variable Y, then the transmission function for the circuit is the sum, modulo-two, of X and Y, that is, $TF = X \oplus Y$. (Note that $\overline{X} = X \oplus 1$ and another way of writing the transmission function for the "not" circuit is $TF = X \oplus 1$.)

The circuits discussed in this section will be represented hereafter by blocks labeled "A", "O", "N"

or "M2", representing "and", "or", "not" or "mod 2" circuits, respectively.

General Method for Circuit Construction

Before a general method for the construction of combinational circuits can be developed, a method for representing simultaneously a control variable at several different points must be discussed. Figure 3.4

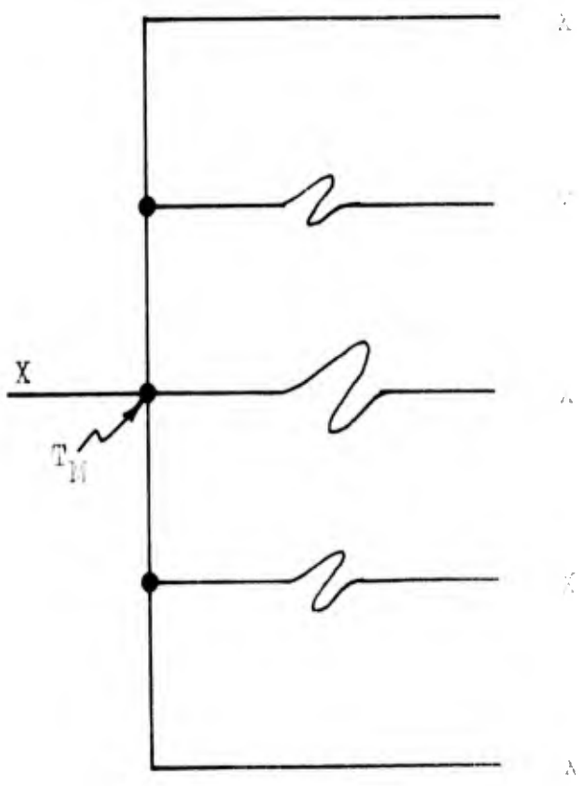


Fig. 3.4

One-Variable "Tree"

illustrates a method of representing control variable X on lines which could furnish inputs to five different circuits. This circuit is called a one-variable "tree". The problem is to have the input which represents X at any time arrive at all five points simultaneously. This situation can be achieved by making the length of line from the master T junction (designated by T_M) to each of the five points equal, the unspecified delays implying this condition.

In order to achieve the situation where more than one (three in this particular case) control variable can be represented at several points, the arrangement illustrated in figure 3.5 is used. This circuit is called a three-variable "tree". The delays required are omitted from the diagram for the sake of simplicity. Controlled-crossings are employed to maintain the two-dimensional character of the circuit.

Inputs representing X, Y, and Z at any time arrive at the three master T junctions simultaneously.

The pattern shown in figure 3.6 is used to insure that pulses on crossing lines never arrive simultaneously at any controlled-crossing. The vertical spacing between the control variable lines in any group must be equal to

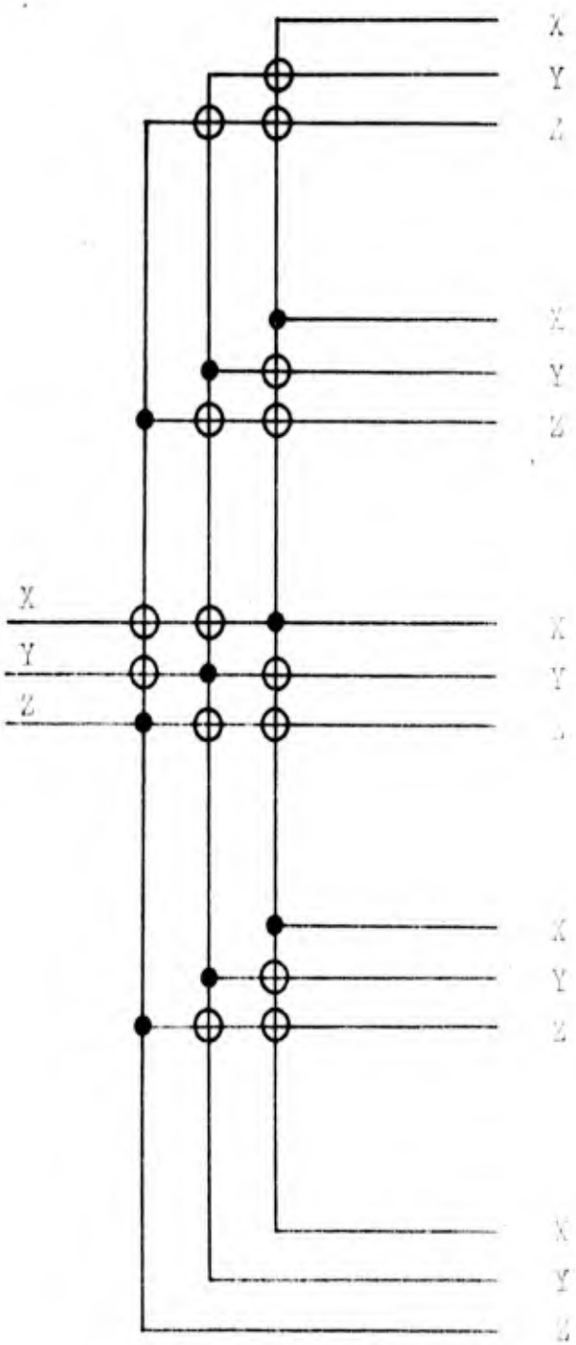


Fig. 3.5

Three-Variable "Tree"

the vertical spacing between the master input lines. With reference to figure 3.5, this means that the vertical spacing between the original X and Y input lines, for example, must equal the vertical spacing between every pair of X and Y input lines on the circuit inputs.

A method for making available a number of control variables and their complements is shown in figure 3.7. In this instance, three control variables are illustrated. This method uses a minimum of "not" circuits.

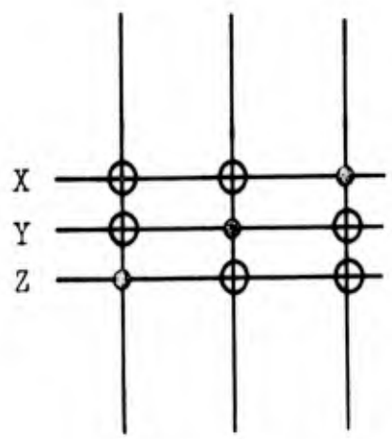


Fig. 3.6

Master T Junction Arrangement

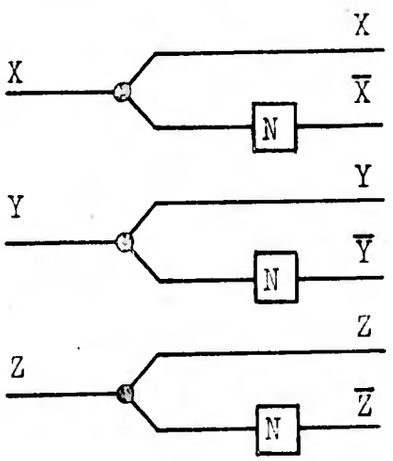


Fig. 3.7

Control-Variables and Complements

Example. As an example, consider the transmission function, $TF = \overline{X}Y + X\overline{Y} + X\overline{Z} + YZ$. In figure 3.8 is shown

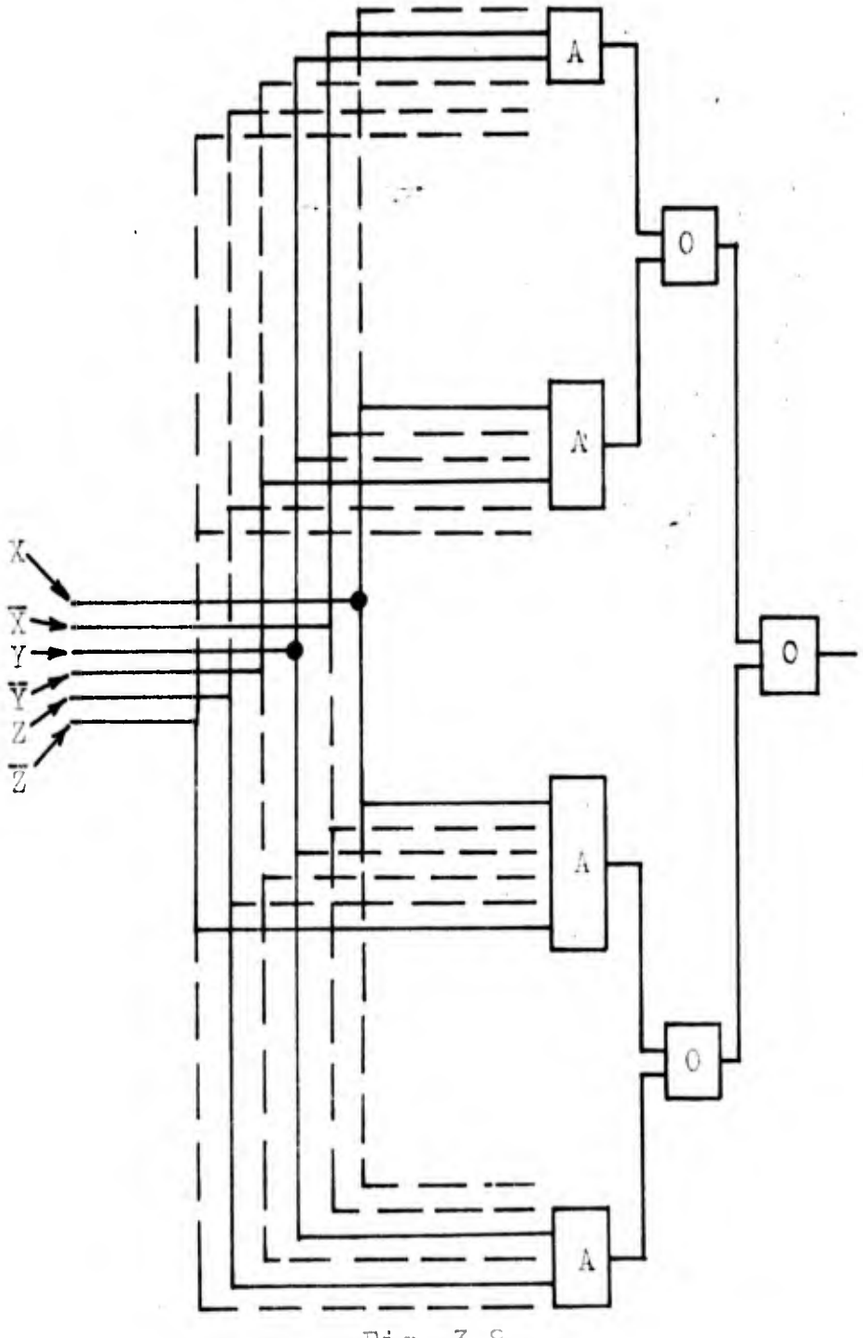


Fig. 3.8

Circuit Diagram for Example
($TF = \overline{X}Y + X\overline{Y} + X\overline{Z} + Y\overline{Z}$).

one circuit for which TF = $\overline{X}Y + X\overline{Y} + X\overline{Z} + YZ$ is the transmission function. In the diagram, the controlled-crossings and the circuit for generating the complements of the control variables are omitted. The circuit actually employs a "pruned" tree, but the unused portions of the tree are shown as dotted lines to clarify what the appearance of a complete tree would be.

The different delay times of the "and" circuits in figure 3.8 pose a slight problem since the "or" inputs must be received simultaneously. This problem can be solved through the use of unspecified delays (not shown on diagram) in the lines between the "and" circuits and the "or" circuits. The delay time of the "slowest" "and" circuit (in this case, the second from the bottom of the diagram) is determined and this is added to the time required for a pulse to travel from this "and" circuit to the "or" circuit involved to yield a total delay. The total delay of the other three "and" circuits must equal the total delay of the slowest circuit.

IV. Sequential Logic Circuits

The operation of a sequential circuit is usually specified by a word statement from which a state diagram is formed. If any redundancy exists in the state diagram, the redundancy is eliminated, and state assignments are made by associating memory states with circuit states. Next-state and output matrices are formed from the state diagram. Boolean expressions are formulated using these matrices. The Boolean expressions specify the combinational circuits required. This synthesis procedure reduces the sequential problem to a combinational problem by removing the time dependence of the original sequential problem (Ref 2:8).

Huffman has demonstrated that all sequential circuits can be analysed as delay devices (Ref 2:8-17). This condition results because a sequential circuit is a memory device and a memory device is a delay device. The neuristor is ideally suited for use in sequential circuits since it has a "built-in" delay capability.

In order to demonstrate the method to be used in designing sequential circuits, six specific design problems are solved in this chapter. Finally a method is proposed

for use in constructing any circuit which is required to solve a design problem of the type demonstrated in the examples.

The term "dit" (short for discrete interval of time) is defined as the time interval between steps in the operation of a sequential circuit. In an asynchronous circuit a dit is the time interval between inputs of the control variable. In a synchronous circuit a dit is the time interval between clock pulses (Ref 2:5).

Method of Circuit Design

The synthesis method developed by Huffman, employing the delay method of analysis, is used in the design of sequential circuits. The specific delay symbol introduced in Chapter I is used in the circuit diagrams and the prefix D (superscript "1" understood) is used to indicate a delay on one dit of the prefixed quantity. The symbols D^2 and D^3 indicate delays of two and three dits, respectively, and, in general, D^n indicates a delay on n dits.

Examples of Design Problems

The six examples which follow may be classified in the following manner: Examples one and six are modular counters; examples two and three are sequence detectors;

and examples four and five are modified sequence detectors.

Example 1. Design a circuit which yields an output pulse each time the total number of ones which have entered the circuit is equal to an odd number. This is a "mod 2" counter.

The state diagram for this circuit is shown in figure 4.1. The zero state exists when the circuit has received

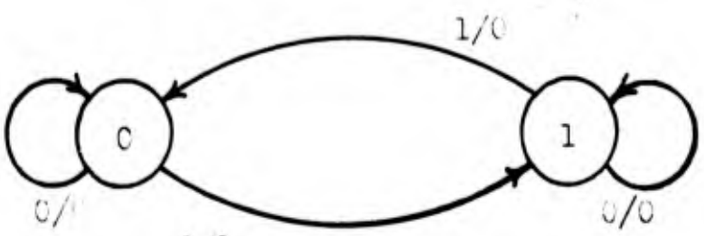


Fig. 4.1
State Diagram for Example 1

an even number of ones; the one state exists when the circuit has received an odd number of ones. The "edges" associated with a particular state indicate the changes of state that occur for either possible input. The two numbers associated with each edge indicate the input that caused the change (number on left of slant line) and the

output that resulted from this change (number on right of slant line). For example, if the circuit is in the zero state and a one input is received, the circuit changes to the one state and yields an output pulse. This is indicated by the edge directed from the "0" state to the "1" state and the associated symbol "1/1".

There can obviously be no redundency in this state diagram. If the memory states "0" and "1" are associated respectively with circuit states "0" and "1", the state assignment is as shown on the state diagram. This state assignment results in the next-state and output matrices shown in figure 4.2 where X is the input, or control,

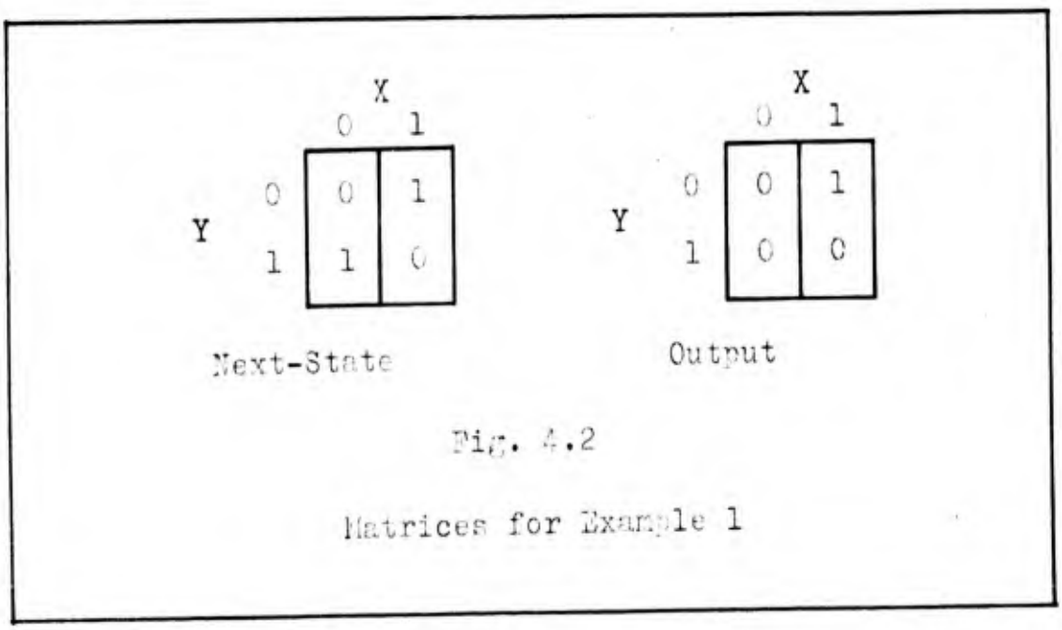


Fig. 4.2

Matrices for Example 1

variable and Y is the present state of the circuit.

From the above matrices the Boolean expressions for the output Z and present state are

$$Z = X\,\overline{Y} \tag{1}$$

$$Y = (\overline{DX})(DY) + (DX)(\overline{DY}) = DX \oplus DY \tag{2}$$

The derivation of Eq (2) is explained here since it may not be immediately obvious. The next-state matrix specifies that the next state is "1" when the present state is "1" and the present input is "0", or when the present state is "0" and the present input is "1". This statement is equivalent to Eq (2) which specifies that the present state is "1" when the previous state was "1" and the previous input was "0", or when the previous state was "0" and the previous input was "1". That this latter statement is equivalent to Eq (2) becomes apparent when one recalls that any quantity prefixed by D represents that quantity one dit earlier, or that quantity previously.

The circuit diagram is shown in figure 4.3. A detailed circuit is shown in figure 4.4 so that the reader may obtain an appreciation for the actual neuristor circuit.

The operation of the circuit with the input sequence 11001 will now be considered. (The left-hand digit

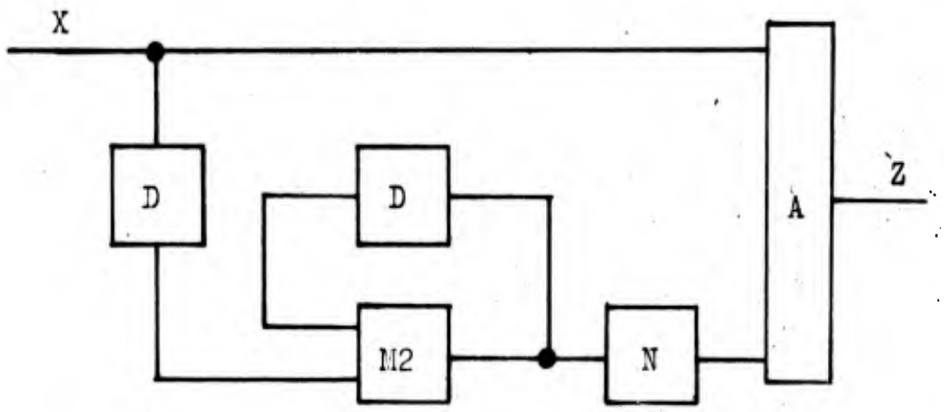


Fig. 4.3

Circuit Diagram for Example 1

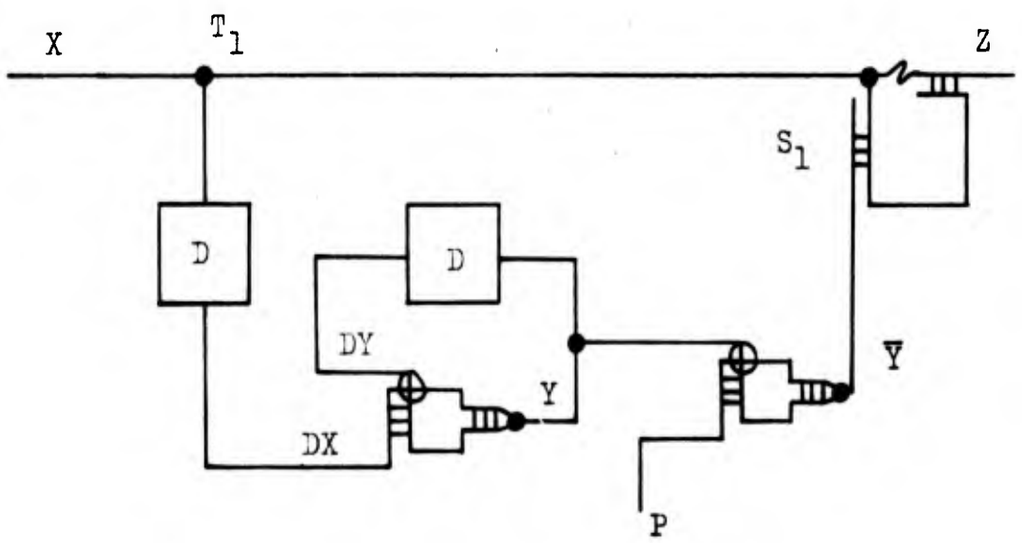


Fig. 4.4

Detailed Circuit for Example 1

represents the input which enters first.) A table which depicts the circuit operation for the entire input sequence is shown below. At t_1 a pulse enters the circuit and proceeds to the first T junction where it triggers the two connecting lines. The pulse which proceeds to the "and" circuit arrives there simultaneously with the P pulse which has progressed through the "not" circuit (since Y = 0), and the output is one. At t_2, one dit later, X = 1 again, DX = 1 and DY = 0; therefore Y = 1 and $\overline{Y}$ = 0. As a result the output is zero. At t_3 X = 0, and regardless of the value of $\overline{Y}$, the output is zero.

Table I
Example 1 Circuit Operation

	X	DX	Y	DY	$\overline{Y}$	Z
t_1	1	0	0	0	1	1
t_2	1	1	1	0	0	0
t_3	0	1	0	1	1	0
t_4	0	0	0	0	1	0
t_5	1	0	0	0	1	1

The nature of the general neuristor network can be seen from a closer study of the detailed circuit for the problem under consideration. The fact to be re-emphasized is that a discharge travels at a constant finite velocity along a line and that instantaneous transmission cannot be assumed. In a neuristor network, in order to have

proper pulse synchronization, every length of line must be included when the delay time of a circuit is computed. For example, the total time required for a pulse to travel directly from the input to the "and" circuit must be equal to the time required for a pulse to travel through the "mod 2" circuit and the "not" circuit to the "and" circuit. That is, the total delay for the pulse which travels directly from junction T_1 along the "upper" line to S_1 must be exactly one dit less than the delay time for the pulse which travels along the "lower" line between the same two junctions. This fact poses no problem since the delay times for the "not" and "mod 2" circuits can be determined.

Example 2. Design a circuit which will yield an output pulse each time that the sequence 010 has been received by the circuit. This is a three-digit sequence detector.

The memory of this circuit consists of the last two inputs that the circuit has received. As a result, the state diagram consists of four possible states as shown in fugure 4.5. The association of circuit states with memory states is "automatic" since the memory states determine the number of circuit states. There cannot possibly be any duplication of states and hence no

redundancy exists in the state diagram.

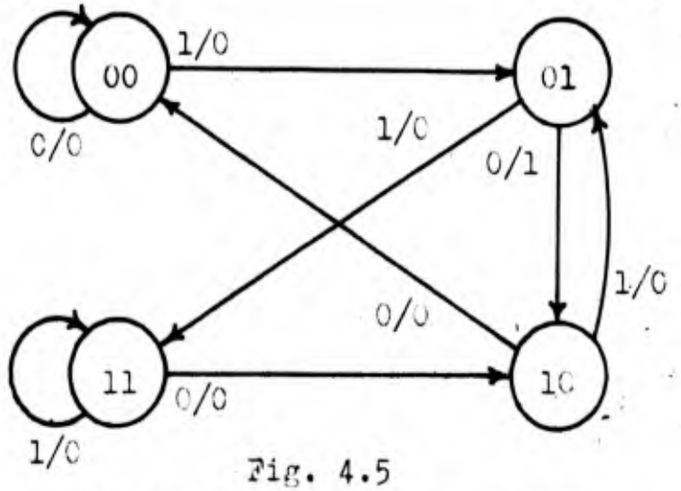


Fig. 4.5
State Diagram for Example 2

The next-state and output matrices are shown in figure 4.6. The resulting Boolean expressions are

$$Z = \overline{X}\,\overline{Y}_1 Y_2 \tag{3}$$

$$Y_2 = DX \tag{4}$$

$$Y_1 = DY_2 = D^2X \tag{5}$$

The circuit diagram for this example is shown in figure 4.7.

Y_1 Y_2	X = 0	X = 1
0 0	0 0	0 1
0 1	1 0	1 1
1 1	1 0	1 1
1 0	0 0	0 1

Next-State

Y_1 Y_2	X = 0	X = 1
0 0	0	0
0 1	1	0
1 1	0	0
1 0	0	0

Output

Fig.4.6

Matrices for Example 2

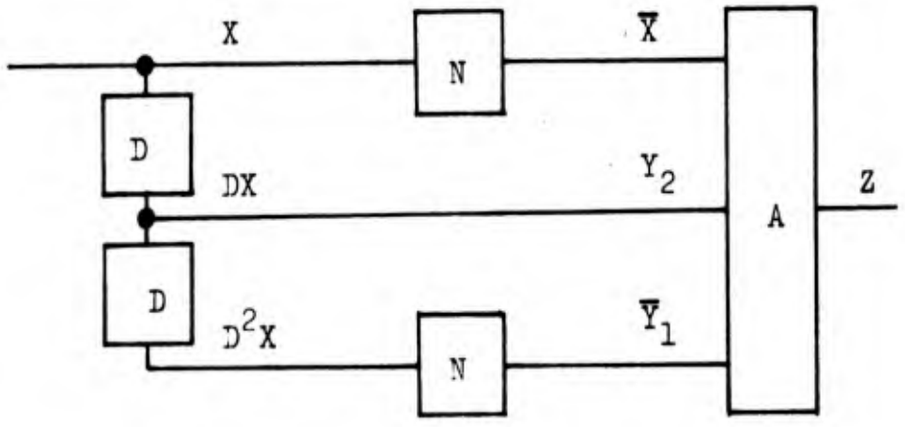


Fig. 4.7

Circuit Diagram for Example 2

Example 3. Design a sequence detector which will detect the sequence 0110.

The state diagram is shown in figure 4.8. The

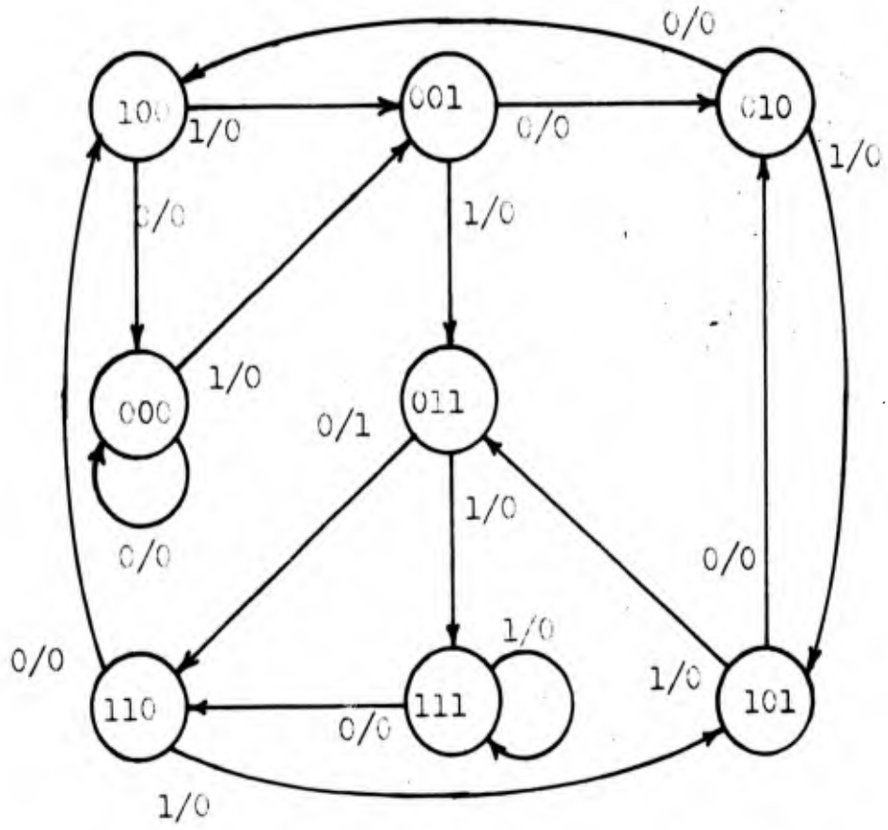


Fig. 4.8
State Diagram for Example 3

memory of this circuit consists of the last three inputs received. The associated matrices are shown in figure 4.9. In this instance, the Y's are not

repeated in the output matrix.

Y_1	Y_2	Y_3	X = 0	X = 1
0	0	0	000	001
0	0	1	010	011
0	1	1	110	111
1	0	1	010	011
1	1	1	110	111
1	1	0	100	101
1	0	0	000	001
0	1	0	100	101

Next-State

X = 0	X = 1
0	0
0	0
1	0
0	0
0	0
0	0
0	0
0	0

Output

Fig. 4.9

Matrices for Example 3

The Boolean expressions are

$$Z = \bar{X}\,\bar{Y}_1 Y_2 Y_3 \tag{6}$$

$$Y_3 = DX \tag{7}$$

$$Y_2 = DY_3 = D^2X \tag{8}$$

$$Y_1 = DY_2 = D^3X \tag{9}$$

The circuit diagram for this example is shown in figure 4.10.

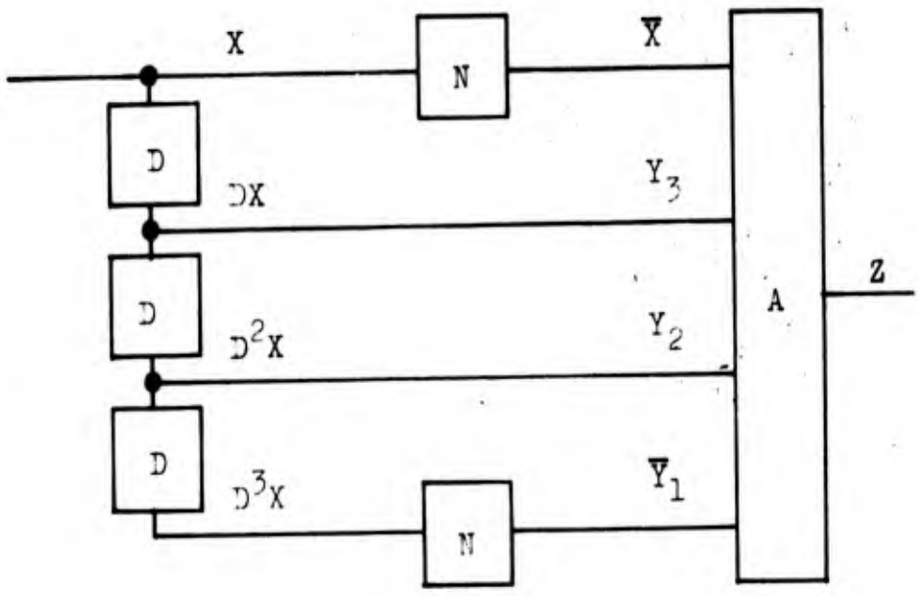


Fig. 4.10

Circuit Diagram for Example 3

Example 4. Design a circuit which will yield an output if the first input of four inputs is a one followed by at least two ones out of the next three inputs or if the first input is a zero followed by at least two zeros out of the next three inputs.

The state diagram, except for outputs, and next-state matrix for this circuit are identical to those for the preceding example. This condition results because the memory states of this circuit consist of the last three inputs received, which was true of the circuit in the previous example.

The output matrix is shown in figure 4.11. The resulting Boolean expressions are

$$Z = \bar{Y}_1\bar{Y}_2\bar{Y}_3 + Y_1Y_2Y_3 + \bar{X}\,\bar{Y}_1\bar{Y}_2Y_3 + X\,Y_1\bar{Y}_2Y_3 + X\,Y_1Y_2\bar{Y}_3 + \bar{X}\,\bar{Y}_1Y_2\bar{Y}_3 \quad (10)$$

$$Y_3 = DX \quad (11)$$

$$Y_2 = DY_3 = D^2X \quad (12)$$

$$Y_1 = DY_2 = D^3X \quad (13)$$

The Karnaugh map of Eq (10) is shown in figure 4.12. From this map, if all one adjacencies are factored, the resulting expression for the output is

$$Z = \bar{Y}_1\bar{Y}_2\bar{Y}_3 + \bar{X}\,\bar{Y}_1\bar{Y}_3 + \bar{X}\,\bar{Y}_1\bar{Y}_2 + Y_1Y_2Y_3 + X\,Y_1Y_3 + X\,Y_1Y_2 \quad (14)$$

which contains two less literals than the previous output expression.

Y_1	Y_2	Y_3	X = 0	X = 1
0	0	0	1	1
0	0	1	1	0
0	1	1	0	0
1	0	1	0	1
1	1	1	1	1
1	1	0	0	1
1	0	0	0	0
0	1	0	1	0

Fig. 4.11

Output Matrix for Example 4

Y_2Y_3 \ XY_1	00	01	11	10
00	1	0	0	1
01	1	0	1	0
11	0	1	1	0
10	1	0	1	0

Fig. 4.12

Karnaugh Map of Output Expression for Example 4

The circuit diagram for this circuit is shown in figure 4.13. In this circuit the tree introduced in Chapter II is used. The pattern shown in figure 4.14 is used to form the circuit control variables. (Note that $D\overline{X} = \overline{DX}$.)

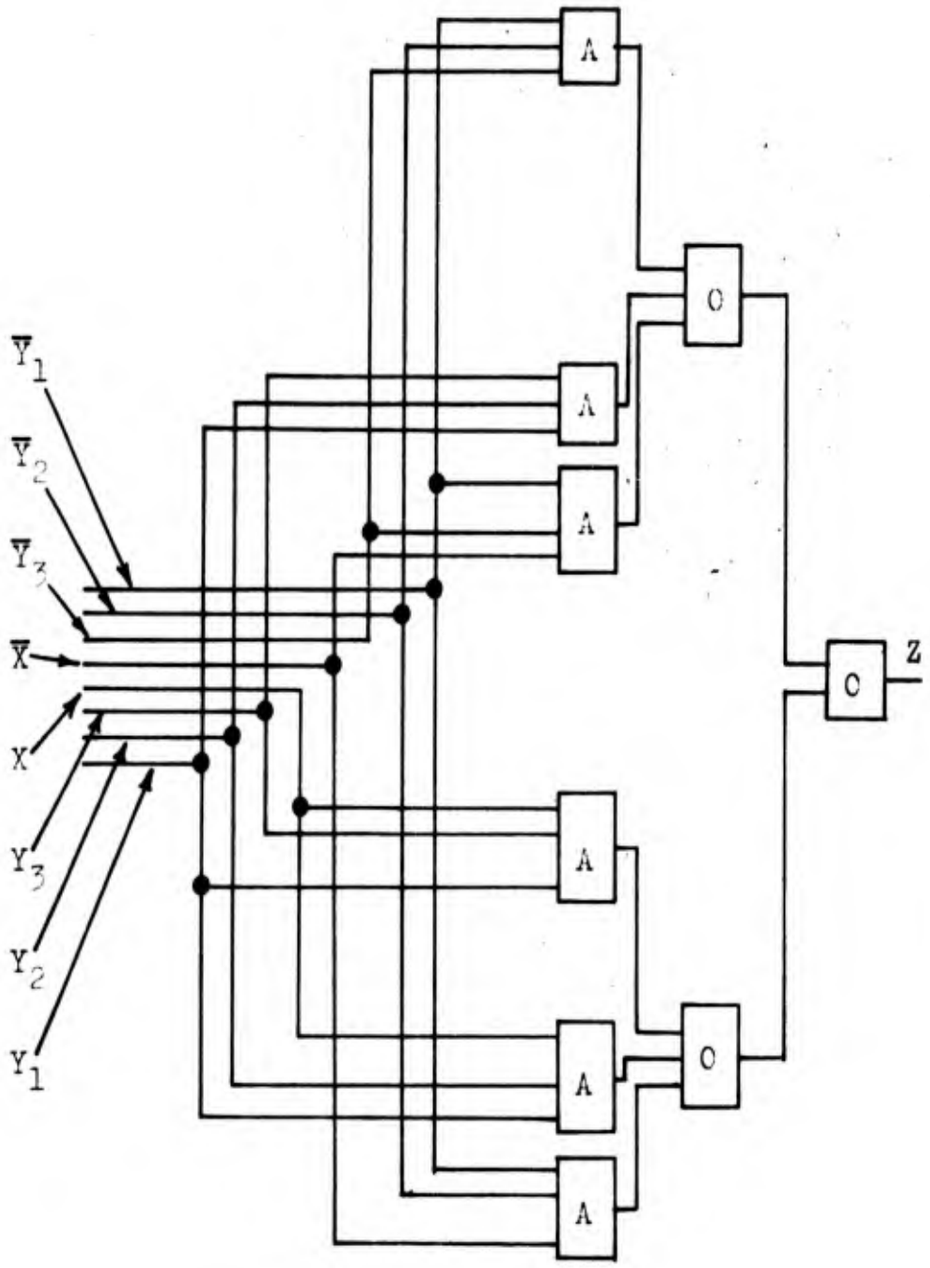


Fig. 7.13

Circuit Diagram for Example 4

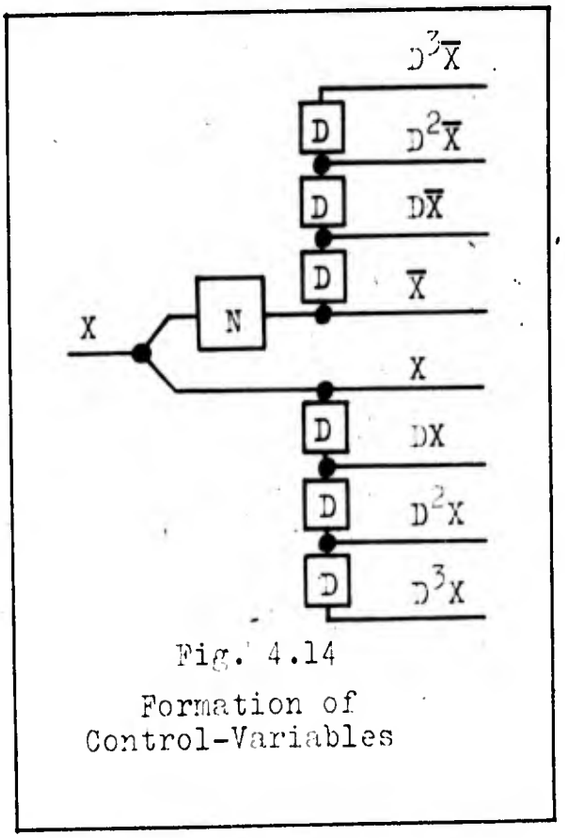


Fig. 4.14
Formation of Control-Variables

Example 5. Design a circuit which will yield an output if a five-input group does not contain three consecutive ones.

The memory of this circuit consists of the last four inputs received. The associated matrices are shown in figure 4.15. The state diagram is not shown because it is not necessary. It should be apparant from the previous examples that if the memory states of a circuit consist of a series of past inputs, no redundancy can possibly exist in the state diagram, and the next-state and output matrices can be written down immediately. The state diagram for this circuit consists of sixteen states because four previous inputs determine the memory state of the circuit.

$Y_1Y_2Y_3Y_4$	X = 0	X = 1
0 0 0 0	0000	0001
0 0 0 1	0010	0011
0 0 1 1	0110	0111
0 0 1 0	0100	0101
0 1 0 0	1000	1001
0 1 0 1	1010	1011
0 1 1 1	1110	1111
0 1 1 0	1100	1101
1 1 0 0	1000	1001
1 1 0 1	1010	1011
1 1 1 1	1110	1111
1 1 1 0	1100	1101
1 0 0 0	0000	0001
1 0 0 1	0010	0011
1 0 1 1	0110	0111
1 0 1 0	0100	0101

Next-State

X = 0	X = 1
1	1
1	1
1	0
1	1
1	1
1	1
0	0
1	1
1	1
1	1
0	0
0	0
1	1
1	1
1	0
1	1

Output

Fig. 4.15

Matrices for Example 5

The Boolean expressions are

$$\overline{Z} = X\,\overline{Y}_1\overline{Y}_2Y_3Y_4 + \overline{Y}_1Y_2Y_3Y_4 + Y_1Y_2Y_3Y_4 + Y_1Y_2Y_3\overline{Y}_4 + X\,Y_1\overline{Y}_2Y_3Y_4 \quad (15)$$

$$Y_4 = DX \quad (16)$$

$$Y_3 = DY_4 = D^2X \quad (17)$$

$$Y_2 = DY_3 = D^3X \quad (18)$$

$$Y_1 = DY_2 = D^4X \quad (19)$$

Eq. (15) becomes

$$\overline{Z} = X\,\overline{Y}_2Y_3Y_4(\overline{Y}_1 + Y_1) + Y_2Y_3Y_4(\overline{Y}_1 + Y_1) + Y_1Y_2Y_3\overline{Y}_4 \qquad (20)$$

$$\overline{Z} = X\,\overline{Y}_2Y_3Y_4 + Y_2Y_3Y_4 + Y_1Y_2Y_3\overline{Y}_4 \qquad (21)$$

The Karnaugh map for Eq (21) is shown in figure 4.16. If the zero adjacencies are factored in groups of four, the resulting expression is

$$\overline{Z} = X\,Y_3Y_4 + Y_2Y_3Y_4 + Y_1Y_2Y_3 \qquad (22)$$

which has two less literals than Eq (21).

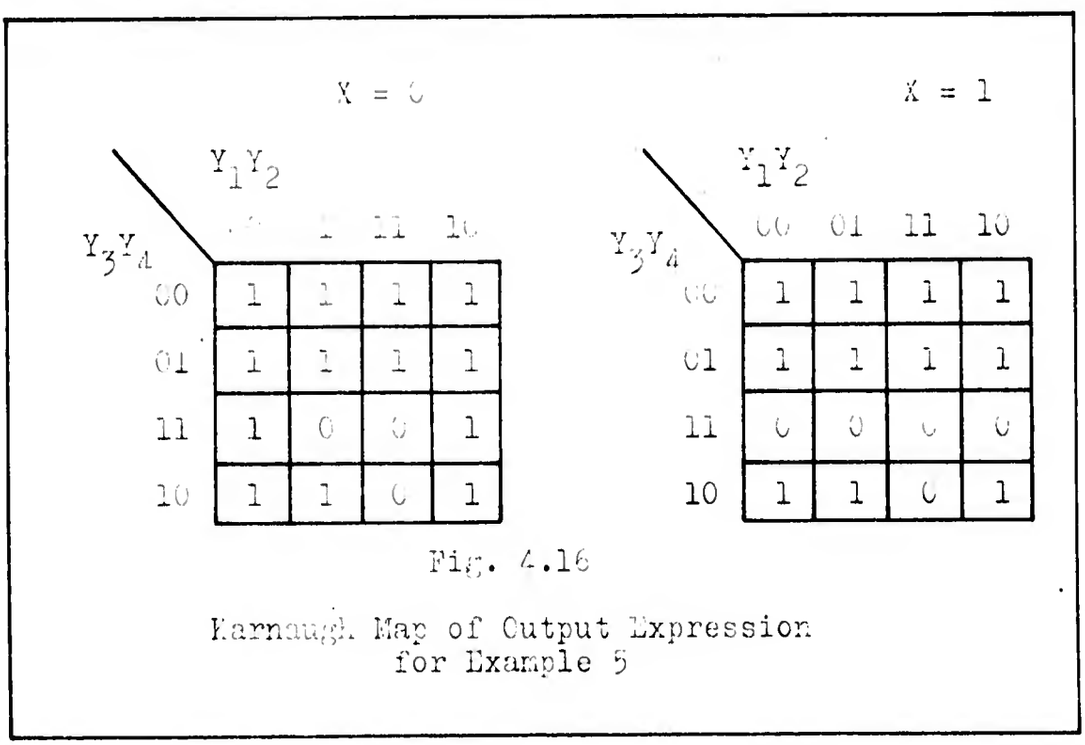


Fig. 4.16

Karnaugh Map of Output Expression for Example 5

Eq (22) can be factored to

$$\overline{Z} = (X + Y_2)Y_3Y_4 + Y_1Y_2Y_3 \tag{23}$$

which becomes

$$Z = \overline{(X + Y_2)(Y_3Y_4)} \quad \overline{(Y_1Y_2Y_3)} \tag{24}$$

$$Z = (\overline{X + Y_2} + \overline{Y_3Y_4})(\overline{Y}_1 + \overline{Y}_2 + \overline{Y}_3) \tag{25}$$

$$Z = (\overline{X}\,\overline{Y}_2 + \overline{Y}_3 + \overline{Y}_4)(\overline{Y}_1 + \overline{Y}_2 + \overline{Y}_3) \tag{26}$$

The circuit diagram for this example is shown in figure 4.17. The input variables are formed using a

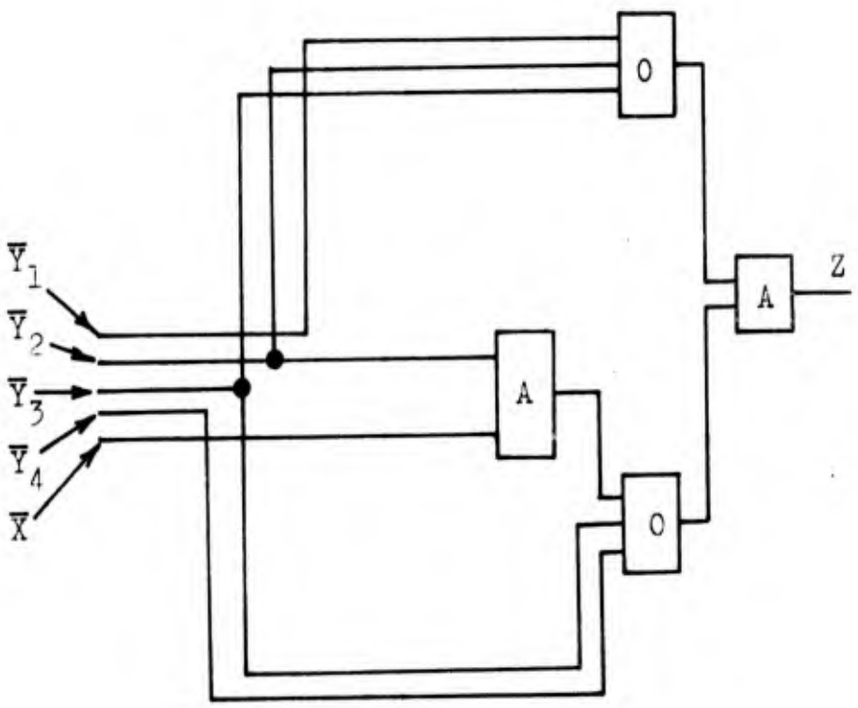


Fig. 4.17

Circuit Diagram for Example 5

pattern similar to that in figure 4.14.

Example 6. Design a modified "mod 3" counter which behaves in the following manner: If the input is a one, the circuit behaves as a "mod 3" counter, that is, the circuit produces the outputs 0, 1, and 2; if the input is a zero, the output is 0.

This circuit has three possible states as shown by the state diagram in figure 4.18. A two-digit number is required to represent the output since there are three possible output conditions.

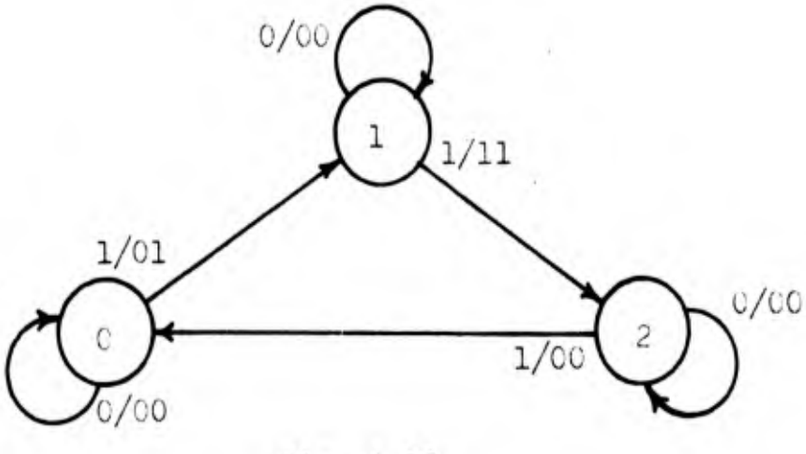


Fig. 4.18

State Diagram for Example 6

State zero is the state for which the total number of ones which has entered the circuit is one, four, seven, etc., state one is the state for which the total number of ones is two, five, eight, etc., and state two is the state for which the total number of ones is zero, three, six, etc. There can obviously be no redundancy in states.

The next-state and output matrices before state assignment are shown in figure 4.19. These matrices are shown in figure 4.20 after the state assignment has been made. This state assignment is just one of many possible assignments.

Y \ X	0	1
0	0	1
1	1	2
2	2	0

Next-State

Y \ X	0	1
0	00	01
1	00	11
2	00	00

Output

Fig. 4.19

Matrices for Example 6
(Before State Assignment)

The Boolean expressions are

$$Z_1 = X\,\overline{Y}_1 Y_2 \tag{27}$$

$$Z_2 = X\,\overline{Y}_1 \tag{28}$$

$$Y_1 = (\overline{DX})(DY_1)(DY_2) + (DX)(\overline{DY}_1)(DY_2) \tag{29}$$

$$Y_2 = (\overline{DX})(DY_2) + (DX)(\overline{DY}_1) \tag{30}$$

where Z_1 and Z_2 represent, respectively, the left-hand output digit and the right-hand output digit as shown in the output matrix.

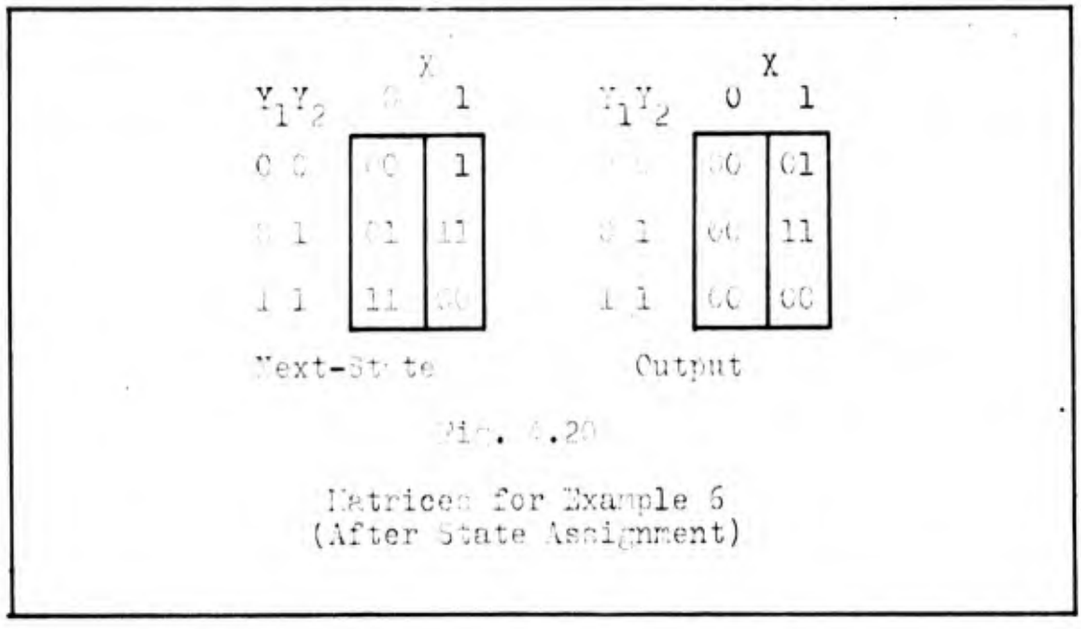


Fig. 4.20

Matrices for Example 6
(After State Assignment)

The circuit diagram is shown in figure 4.21. The variables X, $\overline{X}$, DX, and $\overline{DX}$ are formed as shown in figure 4.14. Note the method used to accomplish the

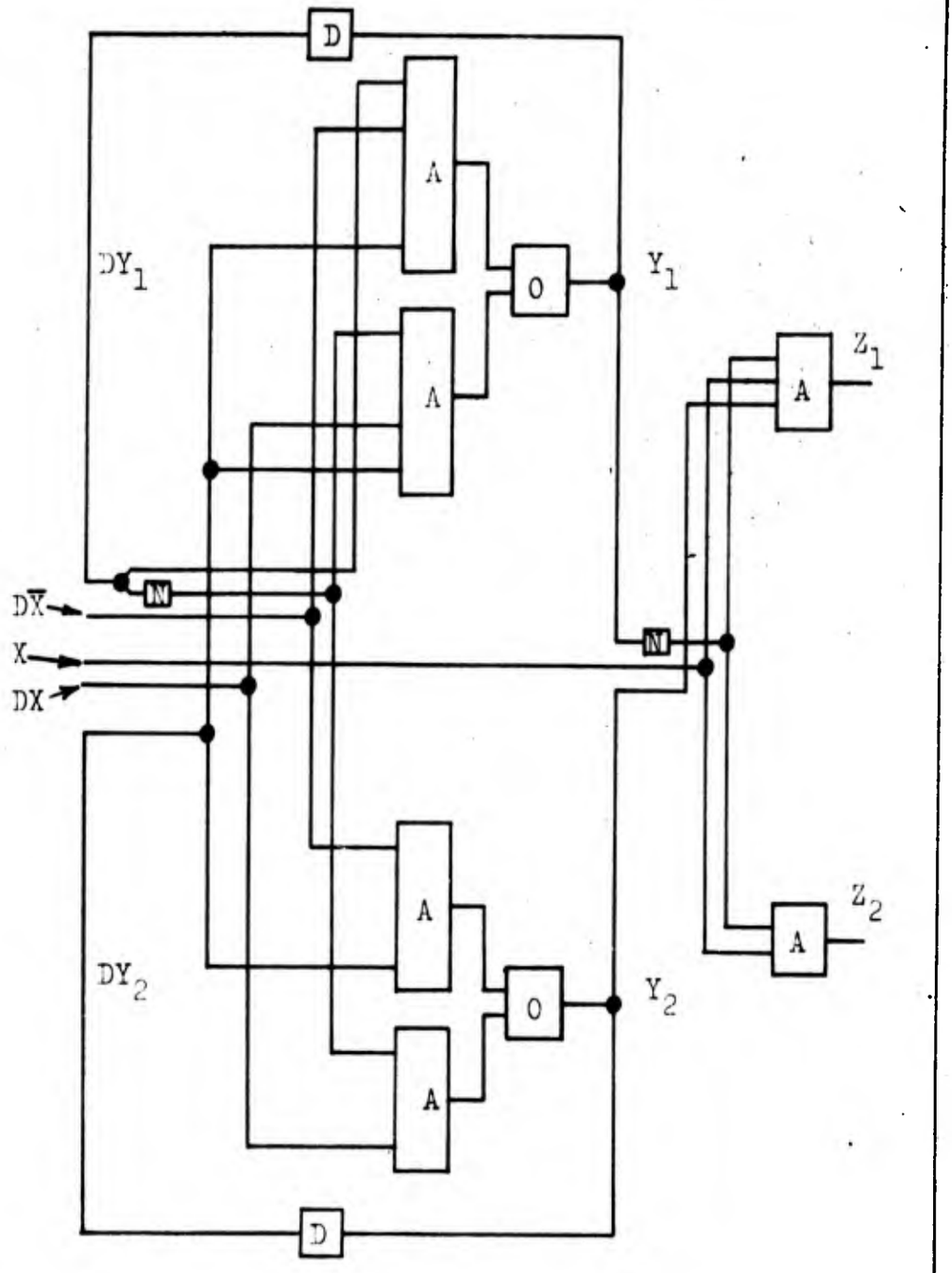


Fig. 4.21

Circuit Diagram for Example 6

feedback required in this circuit. The one dit delay in each of the two feedback loops signifies that the Y_1 and Y_2 outputs must arrive at the tree input simultaneously with the group of inputs to the tree arriving after the Y_1 and Y_2 outputs have occurred. The total delay in the feedback must be one dit.

The circuit has been designed such that Y_1 and Y_2 pulses appear simultaneously. These and the original circuit control variable X are used as inputs to a second tree as shown on the diagram, and the circuit outputs are formed using this tree.

Proposed General Method of Circuit Construction

In order that a specific circuit diagram can be drawn, certain assumptions are made about the requirements of the circuit to be constructed. These assumptions are not limiting in any sense, and the method employed here can be extended to any sequential circuit for which the circuit state is a function of a finite number of past inputs.

This discussion is based on the assumption that the circuit required has four feedback loops. The circuit is assumed to require the control variables X, DX, $\overline{X}$, and $\overline{DX}$. The principles developed here do apply as well to

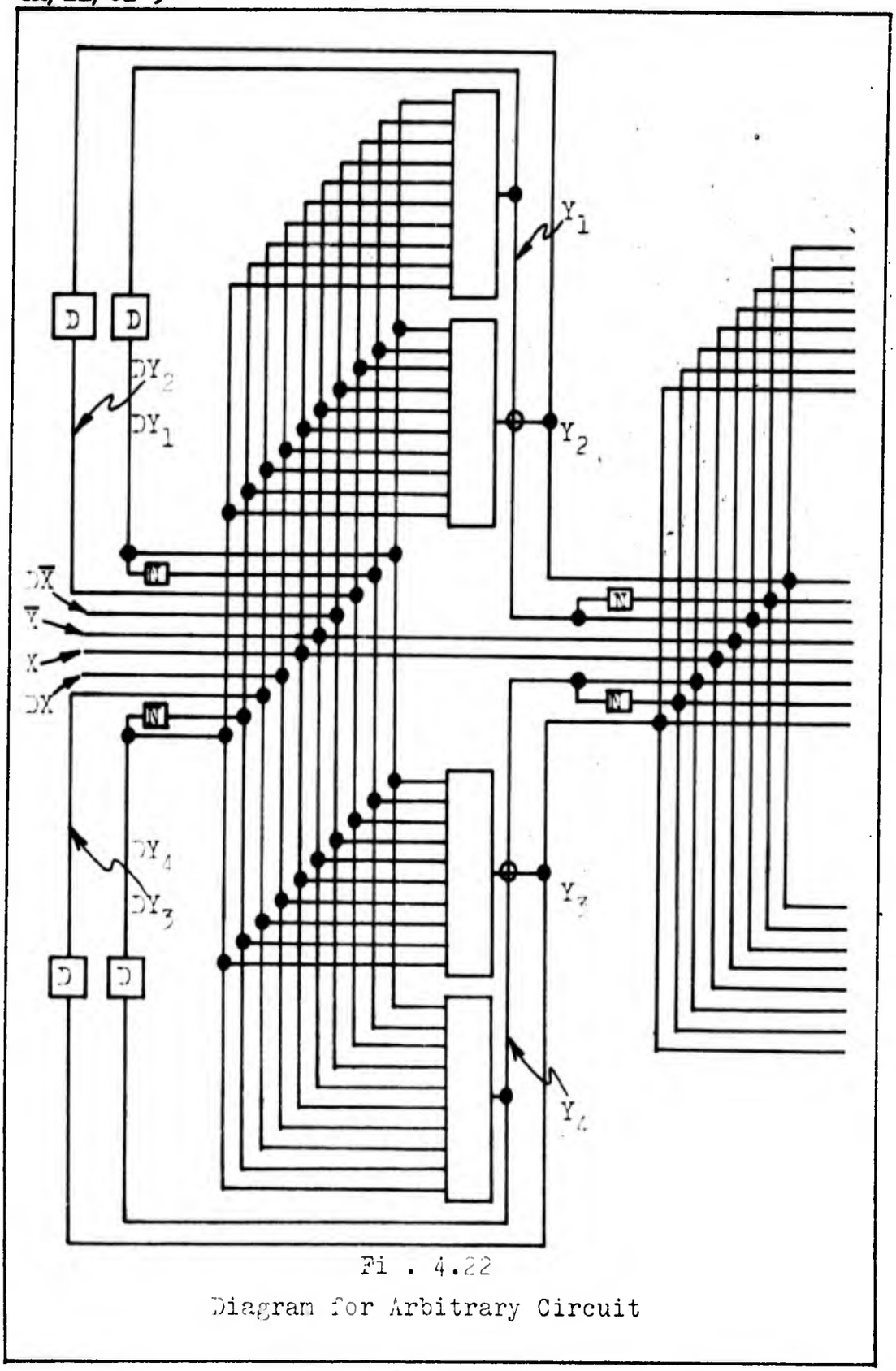
circuits with more feedback loops and more input variables.

The circuit diagram is shown in figure 4.22. Note that all but two of the control-crossings are omitted from the diagram.

The two control-crossings shown are not part of either tree structure. The pulses for variables Y_1, Y_2, Y_3 and Y_4 appear simultaneously at the outputs of the four basic combinational circuits. Consequently pulses will never arrive at the controlled-crossings simultaneously.

The P pulses for all four "not" circuits must pass across other lines. The arrangement shown in figure 4.23 is used to complement DY_1. Distance is measured from the master junctions in a direction counter to pulse flow to insure that pulses arrive simultaneously at points 1 and 2. The P pulse must arrive at the "not" circuit simultaneously with the DY_1 input which means that the P pulse has not yet arrived at the controlled-crossing at the instant that the DY_1 pulse is at point 1 and the DY_2 pulse is at point 2. Pulses will never simultaneously appear at the controlled-crossing. Variable DY_4 is complemented in a similar manner.

The circuit used to complement Y_4 is shown in figure 4.24.



Fi . 4.22

Diagram for Arbitrary Circuit

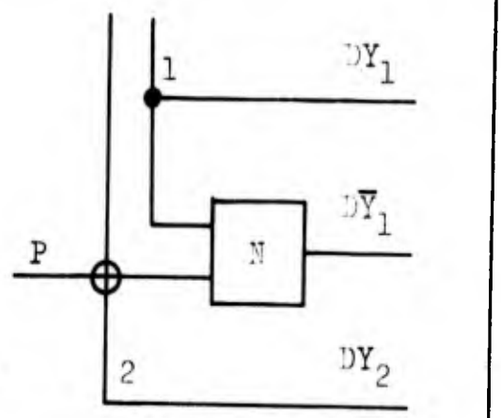


Fig. 4.23
Complementing of DY_1

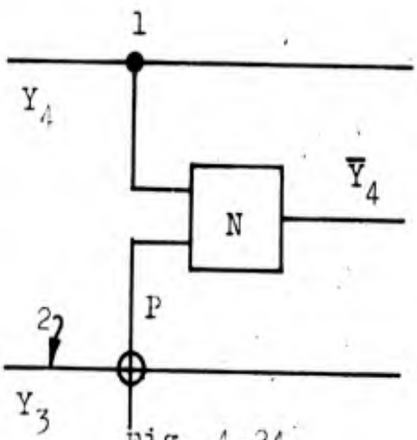


Fig. 4.24
Complementing of Y_4

Pulses are to arrive simultaneously at the master T junctions. Pulses Y_4 and Y_3 arrive simultaneously at points 1 and 2, respectively. The P pulse for the "not" circuit is at the controlled-crossing at the same time. Variable $\bar{Y}_1$ is generated in a similar manner.

The second tree supplies the inputs to the combinational circuits required to generate the sequential circuit outputs (Z's).

The combinational circuits on the first tree obviously need not be limited to four in number. Every variable

formed by these circuits can be delayed and fed back, and each can be complemented by an extension of the pattern shown in figure 4.23. Each variable used as an input for the second tree may also be complemented by an extension of the pattern shown in figure 4.24.

V. Conclusion

Results of Investigation

This study has resulted in the development and demonstration of a method for the design and construction of asynchronous digital logic circuits using neuristors. All neuristor circuits have been constructed in two dimensions.

The basic combinational circuits "and", "or" and "not" have been developed, along with a "mod 2" circuit. The basic circuits have been employed, together with a variable tree, to construct an arbitrary combinational logic circuit.

Six specific examples of sequential design problems have been considered. The problems have been solved using the delay method of analysis. The six examples have included three types of problems: modular counters, sequence detectors, and modified sequence detectors.

A general method for the construction of asynchronous sequential logic circuits has been proposed. This method is applicable to any problem for which the output is a function of a finite number of past inputs.

Recommendation for Future Study

Since no neuristor exists at present, any attempts at minimization of neuristor assemblies must be based on the assumption that such minimization consists of using the least amount of neuristor assembly and requiring the least amount of time to accomplish the desired end. No attempt to achieve minimization has been made in this investigation.

It is recommended that in a future study one particular sequential problem be solved and the circuit constructed using first the asynchronous approach developed here and then the synchronous approach developed by Crane (Ref 1:116). Then minimization criteria pertaining to both length of line and time of circuit operation can be applied to each of the two circuits to determine which is the better approach: asynchronous or synchronous.

Bibliography

1. Crane, H. D. Neuristor Studies. Technical Report No. 1506-2. Stanford University, Stanford, California: Stanford Electronics Laboratory, 11 July 1960.

2. Huffman, D. A. "The Philosophies of Sequential Circuit Behavior." Proceedings of the First Midwest Symposium on Circuit Theory, paper no. 10: 1-24 Symposium on Circuit Analysis conducted by the University of Illinois, Urbana, Illinois (1955).

Vita

[redacted]

he enrolled at Tulane University, New Orleans. In 1952 he was appointed a Midshipman in the U. S. Navy and entered the U. S. Naval Academy in June, 1952. In June, 1956 he was graduated with the degree of Bachelor of Science and was commissioned as a Second Lieutenant in the U. S. Air Force. His military assignments prior to his coming to the Institute of Technology were as Missile Maintenance Officer and Missile Operations Officer in the Air Training Command and in the Tactical Air Command.

Permanent address: [redacted]

This thesis was typed by Mrs. Jean P. Piske

UNCLASSIFIED