DESIGN OF BINARY LOGIC SYSTEMS
USING MECHANICAL ELEMENTS

By

GLENN HANCOCK DEWEY

Bachelor of Science
University of Vermont
Burlington, Vermont
1957

Master of Science
University of Vermont
Burlington, Vermont
1959

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
May, 1973

DESIGN OF BINARY LOGIC SYSTEMS

USING MECHANICAL ELEMENTS

Thesis Approved:

Atmaram H. Soni
Thesis Adviser

M. M. Mamoun

J. A. John

Paul A. McCollum

D. D. Durham
Dean of the Graduate College

ACKNOWLEDGMENTS

My sincere gratitude is expressed to all who have helped in any way to make this study possible. In particular I am indebted to Dr. A. H. Soni, Chairman of my Advisory Committee, for his encouragement and guidance throughout the investigation. He kept me going when things looked bleak!

The encouragement and helpful suggestions of the members of my committee, Dr. M. Mamoun, Prof. P. McCollum and Dr. J. Johnson is greatly appreciated.

To my "mechanisms companions", Dr. Matthew Huang, Mr. Rao Dukkipati, Mr. Dilip Kohli and Mr. Syed Hamid, I offer my gratitude.

My indebtedness is expressed to the University of Costa Rica for sending me to Oklahoma State University to continue my studies. I am especially grateful to Ing. Rodrigo Orozco, Head of the Department of Electrical and Mechanical Engineering at the University of Costa Rica, for the confidence he has shown in me.

For my family, Amalia, Maria and Garner, and the love and care I have received I overflow with joy.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure		Page

CHAPTER I

INTRODUCTION

1.1 Review of Literature

In 1854 an English mathematician, George Boole, invented a two-valued algebra. Since that time Boolean algebra was well developed, but as a practical tool of technology it lay idle for nearly a century. Then in 1938 Claude E. Shannon [1]* recognized that the logic of electrical switching circuits is comparable to the logic expressed by Boolean algebra. This was the beginning of a rapid and extensive development in the field of switching circuits. The conventional branch of logic involving the basic connections OR, AND, NOT, NOR and NAND was the earliest to develop.

1.1.1 Conventional Logic

Systematic circuit synthesis was extended from combinational to sequential logic involving MEMORY with the work of Huffman [2] in 1954. Continued advancements in the electronic implementation of these circuits have made possible today's high speed digital computers, as well as many other digital systems.

*Numbers in brackets refer to numbered references in the Bibliography.

Fluid power and fluidic implementation of binary circuits [3, 4] has also been extensively developed in the last decade and a half. Cole [5], Maroney [6] and Woods [7] have contributed methods of logic circuit synthesis in the area of fluid power and fluidics, although the methods are not necessarily restricted to this area.

With respect to mechanical implementation of binary logic, the literature reveals very little. In a series of articles on engineering applications of Boolean algebra, Beizer and Leibholz [8] included a "fictional case history" of the design of a coin changer showing a mechanical circuit to implement the logic. A few mechanical binary elements were suggested in diagrams presented in a series of articles by Bennett [9] on binary logic.

In 1963 a dissertation was written by Brandt [10] on the use of a seven-bar linkage to generate binary logic functions. The linkages of Brandt were not completely satisfactory because the output link did not remain stationary in its two stable positions at all times when it should.

Szekely, Muresan and Lelutiu [11, 12] presented basic concepts in the generation of binary and multipositional logic with mechanical elements. In their studies they included a mechanical relay.

1.1.2 Threshold Logic

Work in a new field of logic, "threshold logic," made its appearance with a publication by McNaughton [13] in 1957. Threshold logic presented a new challenge because it is "more powerful" or "richer" than conventional logic [14, 15]. The power or richness of threshold logic is due to the general ability of a single element to perform the logic of a more complex Boolean expression. Lewis and Coats [14] give an example of a function of seven variables that requires twelve AND-OR gates for its realization but requires only <u>one</u> threshold gate for its realization. Threshold elements promise an appreciable reduction in the total number of components and connections in a logic system. Sheng [16] shows that a threshold gate can be considered a generalization of the conventional AND, OR, NAND and NOR gates.

In spite of the significant advantages of threshold elements, there are also disadvantages. Among the disadvantages is the problem of converting the Boolean representation of a function into the threshold form (see Chapter III). During the years 1957 to about 1965, considerable attention was given to this problem. Dertouzos has been one of the principal contributors. He gives a "Brief History of Threshold Logic," and includes an extensive bibliography [15]. An early and significant contribution was made by Winder [17] in his doctoral dissertation.

Since the appearance of Haring's work [18], there has been a concentrated effort on multithreshold threshold elements. Multithreshold threshold elements represent a generalization of the theory of threshold logic. Among the contributions are those of references [19-22].

Threshold logic has been successfully implemented with electrical circuits. Lewis and Coats [14] give examples of circuits for single threshold logic and Haring [18] mentions the use of a tunnel-diode circuit with three thresholds. Other circuit realizations of multithreshold threshold elements are reported in the references [23-29].

Outside of the electrical field, the only work found on threshold logic implementation is in fluidics. Eckerlin and Bell [30] show how a threshold gate composed of four basic fluidic elements has been developed and applied to a few practical problems.

To the best of the author's knowledge, the use of mechanical linkages in the implementation of threshold logic is virtually unexplored. The "whiffletree" type linkage, a binary to multiposition converter, reported by Chironis [31] does, however, play a significant part in the development of this dissertation. Haas and Crossley [32] also used this type of linkage for a four-bit binary adding mechanism.

1.2 Motivation for Study of Mechanical Logic Synthesis

Mechanical switching and binary-logic controls have traditionally been designed by a trial-and-error, seat-of-the pants approach. Whether such controls are ever recognized as performing "switching or binary-logic" remains doubtful. It seems likely, in most cases, that the mechanical control systems are simply "linkages doing a job."

In this dissertation, the systematic Boolean techniques employed in electronics and fluidics are used for mechanical logic synthesis.

There are applications of binary controls where reliability, simplicity and safety are of greater importance than extremely low response time and very miniature parts. In some of these applications mechanical systems could become advantageous over their counterparts. Where a radioactive or explosive environment is involved, a mechanical linkage would be preferable to an electrical system. Where a compressed fluid would not be readily available, a mechanical linkage could be advantageous over a fluid logic system.

In a recent publication on fluidics the statement is made that, "The only completely successful transducers in this area ("area" refers to the transduction of electrical voltage to fluidic signal) involves an intermediate transduction to mechanical position, . . ." [33]. The signal medium for the logic of this dissertation is position. Thus the concepts presented here could conceivably be used to perform logic operations at the electrical-fluidic interface, where the transducers exist.

A certain coin changer, used with vending machines, employs electronics to perform most of its logic operations. However, a position signal is used to sense the presence or absence of coins in two storage tubes. A logic operation is performed mechanically at the point of transduction of the position signals to voltage signal. By performing the logic mechanically, an electrical switch is eliminated with essentially no added complication to the system.

In a random-access-memory storage unit, a tape head is moved to any of twenty-eight positions with a mechanical linkage [31]. Binary electrical signals are transduced to binary position signals of the linkage input cranks. A whiffletree linkage performs the logic for providing the multiposition output at the memory head.

The examples above illustrate the successful use of linkages in performing logic operations. Such examples motivate the belief that many more applications of mechanical logic will be made if designers are aware that simple linkages can be used systematically in binary control. It is hoped that this study will provide a step in making "linkics" as commonplace as "electronics" and "fluidics"!

1.3 Scope of the Study

The purpose of this dissertation is to present the synthesis of mechanical linkages to generate binary logic expressed by Boolean algebra functions. There are two different approaches to the generation of Boolean logic. They are the approaches of conventional logic and threshold logic. In this study both approaches are utilized to synthesize mechanical logic systems.

1.3.1 Conventional Logic Systems

Simple four-bar mechanisms are shown to be the basis for the generators of the basic conventional logic functions OR, AND, NOT, NOR, NAND and MEMORY. The four-bar mechanisms are assembled in various ways to form linkages which generate these basic functions. Slightly more complicated linkages are synthesized to generate the basic conventional functions COINCIDENCE and OR-EXCLUSIVE. A bistable mechanical power amplifier is developed to make the linkage function generators "active".

The systematic methods employed in electronic and fluidic logic are used in examples of control circuit synthesis. A passive mechanical circuit is synthesized to control the transfer of soap boxes from one conveyor to another on an assembly line. In the second example, an active circuit using memory, is synthesized to control the speed of a rotating shaft. The conventional logic synthesis is treated in Chapter II.

1.3.2 Threshold Logic Systems

Using the approach of threshold logic, linkages are synthesized to generate the 256 functions of three binary variables. The threshold function generators are composed of two parts: a separator linkage and a threshold linkage. The composition method selected here allows for the synthesis of all 256 function generators with a few basic component parts. The basic components are synthesized in Chapter III and include five basic separator linkages and nineteen basic threshold linkages.

Structural synthesis and performance criteria are used in selecting favorable linkages for the basic separator and threshold linkages. Dimensional synthesis methods are applied to all of the linkages. The dimensions are recorded in tabular form. Design charts, presented in Chapter III, give the step-by-step procedure for the direct selection of the separator and threshold linkages needed to generate any one of the 256 functions. Two examples are given for the use of the design charts.

CHAPTER II

SYNTHESIS OF CONVENTIONAL LOGIC FUNCTION GENERATORS

The mechanical linkage components which are capable of simulating conventional logic functions are essentially simple mechanisms. This chapter presents these simple mechanisms and shows how they can be combined to form generators of the basic logic functions. Both passive and active logic systems are described. Examples are given of the synthesis of a combinational logic circuit using passive function generators and a sequential logic circuit using active function generators .

2.1 Basic Linkage Components

The simple mechanisms used to simulate the basic logic functions are shown in Figures 1, 2 and 3. Figure 1 shows a four-link mechanism with A_0A as the input link, AB as the coupler link, BB_0 as the output link, and A_0B_0 as the fixed link. These four links are connected using four pin joints at A_0, A, B, and B_0. The axes of the pin joints are parallel to one another and perpendicular to the plane of the paper. Because of this specific geometry, the mechanism moves in the plane of the paper. For the purpose of logic function simulation, the four-link mechanism is required to move from the position $A_0A_1B_1B_0$ to $A_0A_2B_2B_0$.

Figure 2 shows a crank-slider mechanism with A_0A as the input link, AB as the coupler link and the slider joint at B which translates along

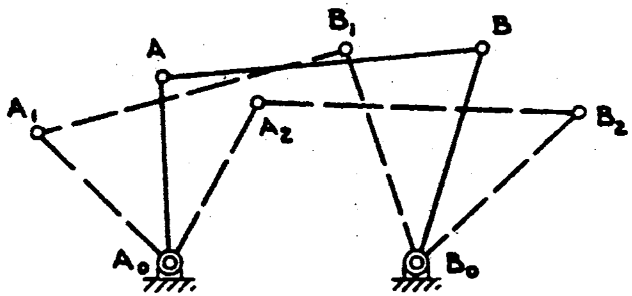


Figure 1. Four Link Mechanism

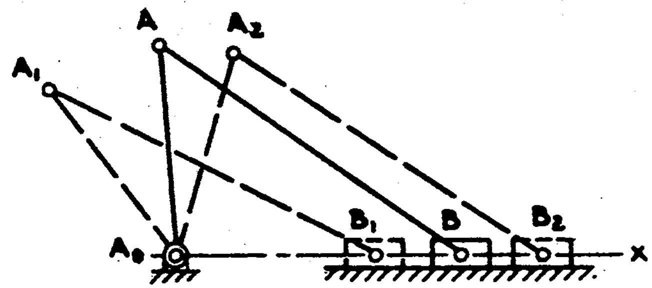


Figure 2. Crank-Slider

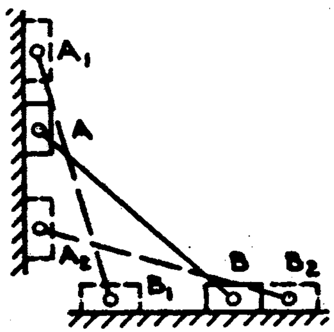


Figure 3. Double-Slider

the fixed axis A_0X. For simulating the logic function, the crank-slider mechanism is required to move from the position $A_0A_1B_1$ to the position $A_0A_2B_2$.

Figure 3 shows a double slider mechanism. The input motion is provided through the slider at A and the output motion is obtained through the slider at B. For simulating the logic functions, the mechanism is required to move from the position A_1B_1 to the position A_2B_2.

The four-link mechanism of Figure 1 is selected when input motion and output motion is rotary. The crank-slider mechanism of Figure 2 is selected when input motion is rotary and output motion is translatory. This mechanism may also be selected when input motion is translatory and output motion is rotary. The double slider mechanism of Figure 3 is selected when input and output motions are translatory.

The input motion to these mechanisms is a variable that can be in one of two states. The two states are described using an analogy of an electrical system. An electrical switch can be open, in which case there is no electrical circuit made beyond the switch, or it can be closed in which case the electrical circuit is made. In Boolean algebra language, the case where the circuit is made is designated by a "1"; the case where the circuit is not made is designated by a "0". The "on" and "off" positions of the switch correspond to the two positions of the mechanisms shown in Figures 1, 2 and 3. These mechanisms may be toggle-actuated in order to retain them in either of the two positions.

2.2 Basic Logic Functions

The basic functions of conventional logic are the OR, AND, NOT, NOR, NAND, MEMORY (Flip-Flop), COINCIDENCE and OR-EXCLUSIVE. The mechanisms described in Figures 1, 2 and 3 can be assembled in several different ways to generate these basic logic functions. Figures 4 through 11 present mechanical assemblies which generate the basic logic functions. In each figure, the logic function is described using the Boolean algebra, the corresponding truth table, and the electrical and mechanical analogues. The symbols A, B, C, etc., are used to represent the input variables and the symbol Z is used to represent the output variable. In the electrical analogue, the "on" and "off" state of the variable describes the "on" and "off" positions of the switch. In the mechanical analogue, the "on" and "off" state of the variable describes the two finitely separated positions of the link representing the variable under consideration. In the following section a description is given of the mechanical assemblies which generate the basic logic functions.

2.3 Linkage Systems Which Generate the Basic Logic Functions

2.3.1 OR Function

The logic function OR has a minimum of two input variables, and the Boolean operator used to express the OR connective is a plus sign with a dot ($\dot{+}$).* If A and B are the two input variables and Z is the output

*The symbol $\dot{+}$ is used for the Boolean operator OR which is to be distinguished from the arithmetic sum $+$.

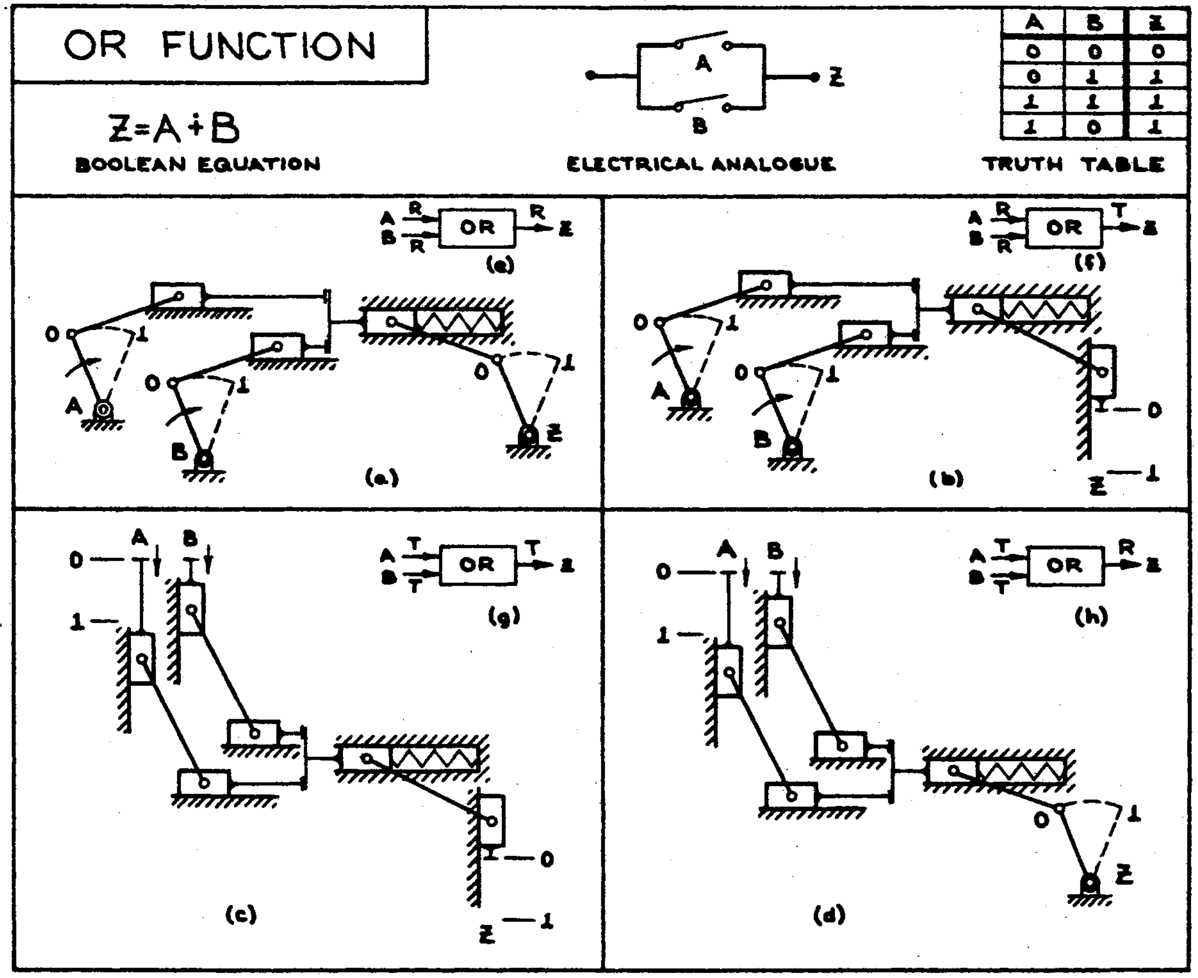


Figure 4. Generators of the OR Function

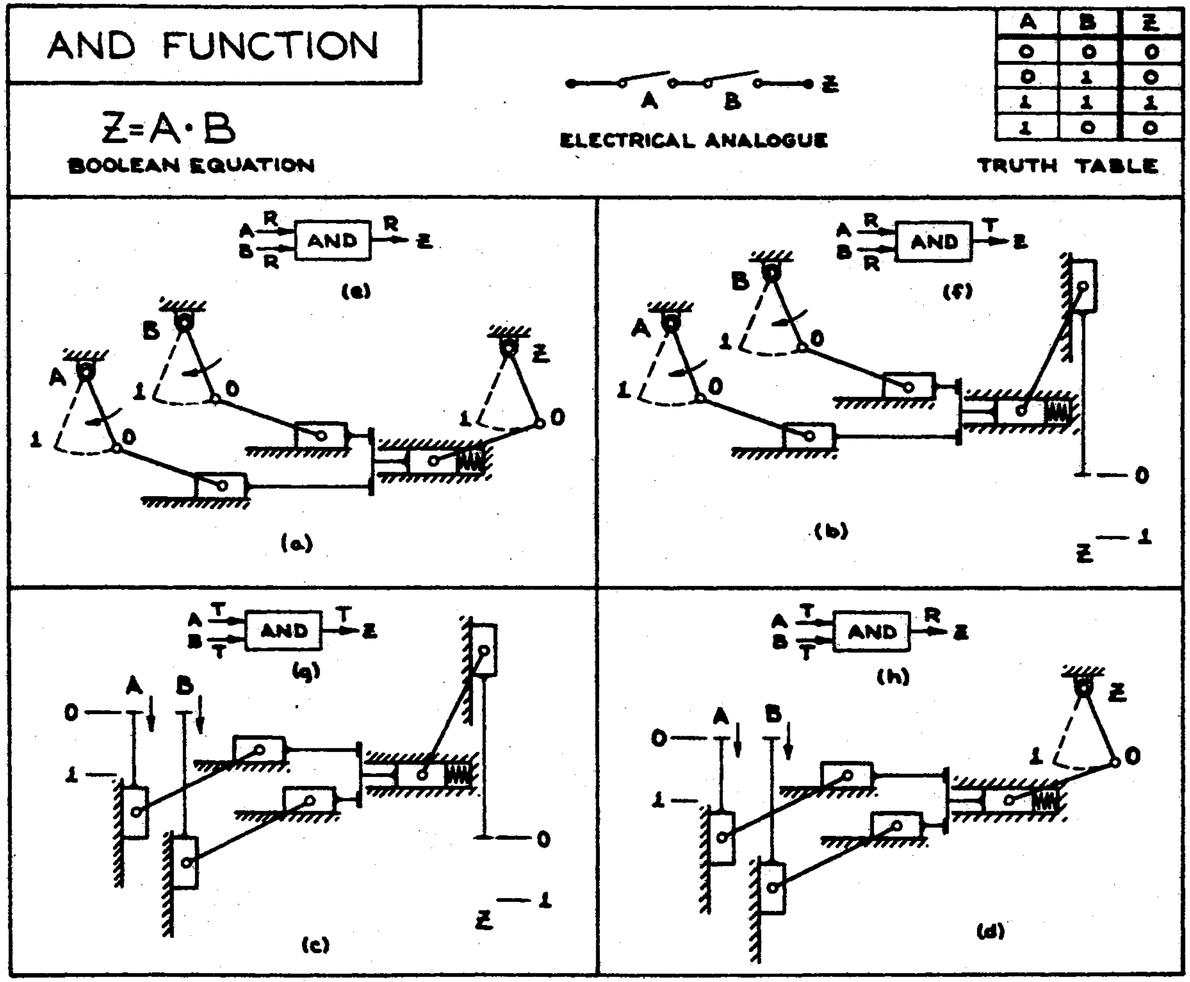
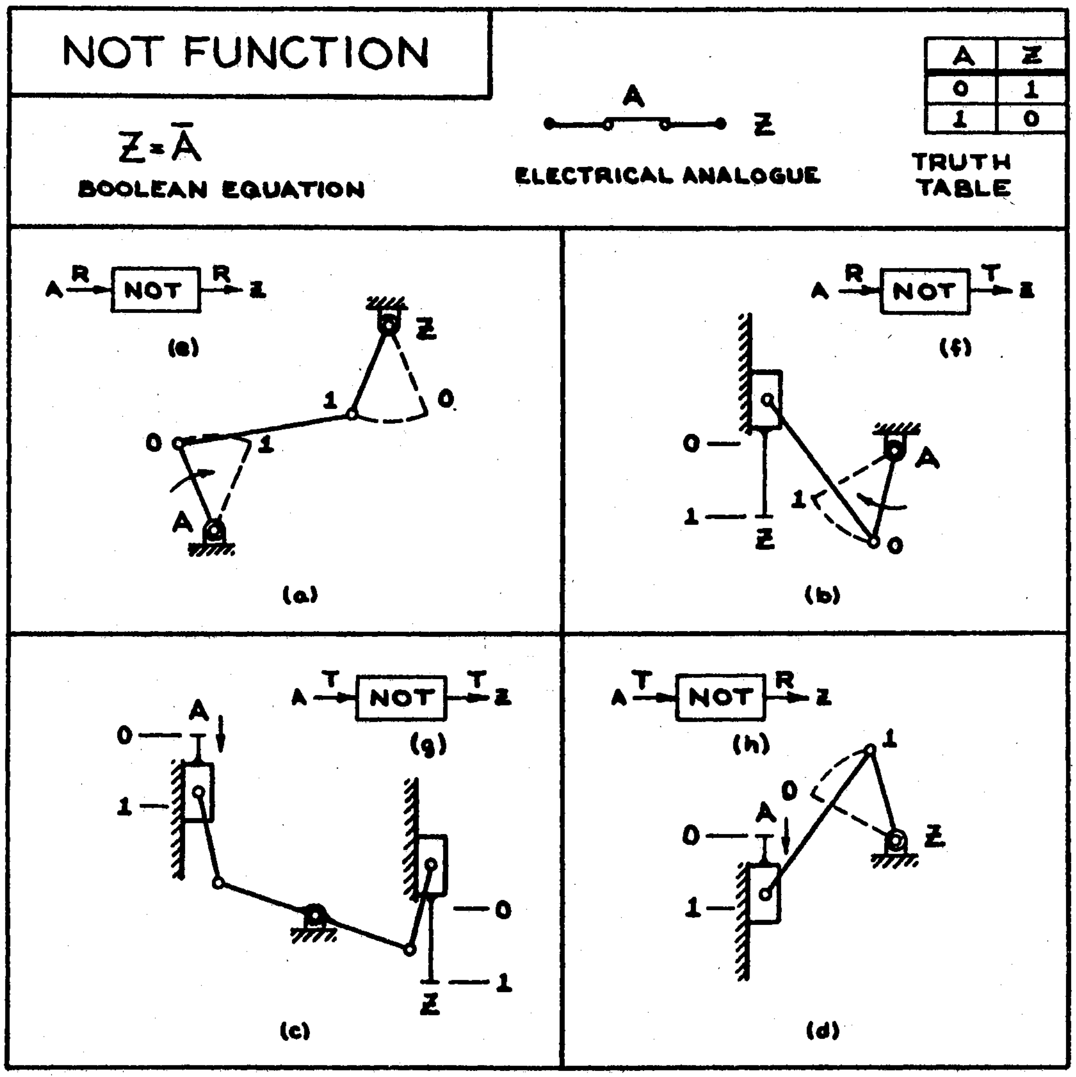


Figure 5. Mechanical Generators of the AND Function



A	Z
0	1
1	0

Figure 6. Mechanical Generators of the NOT Function

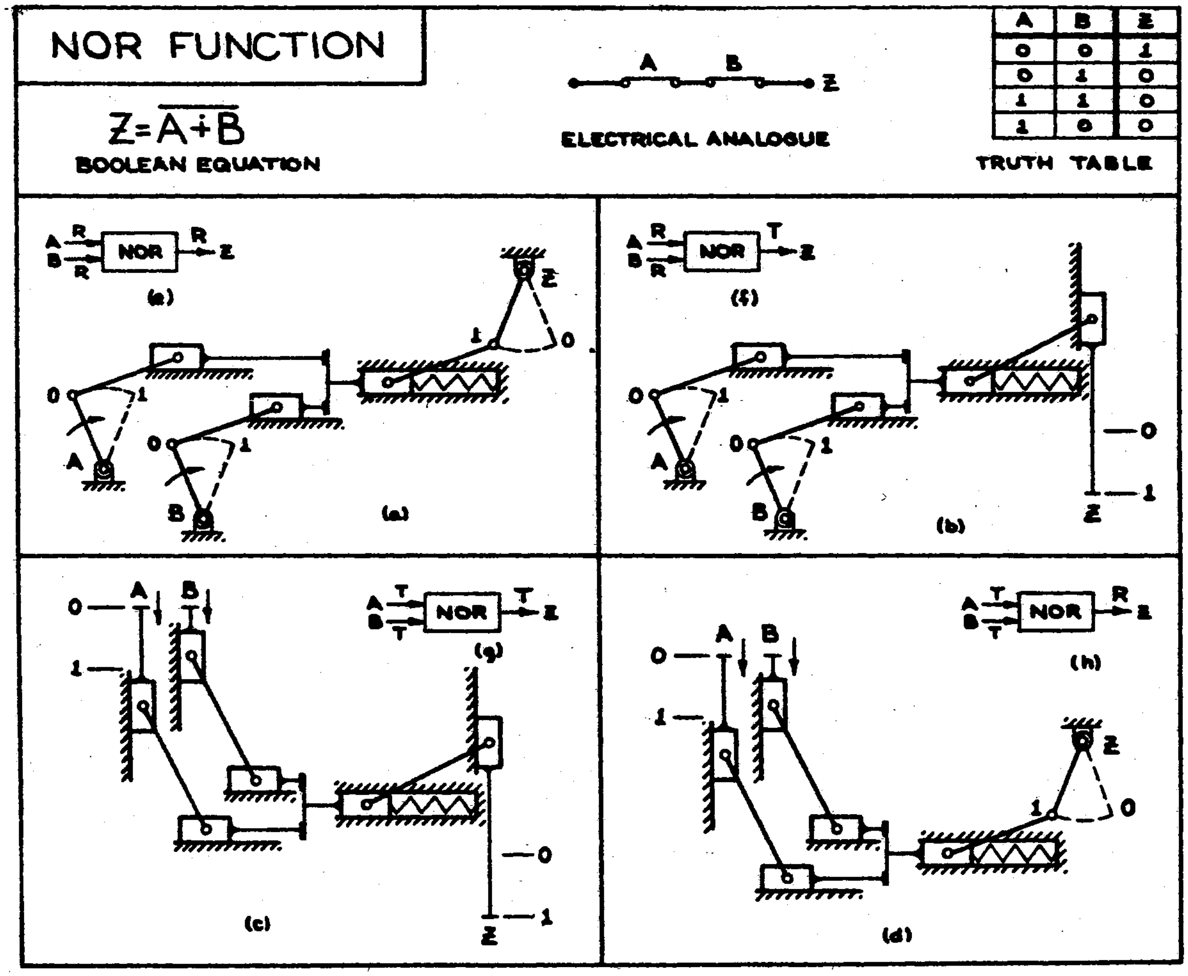


Figure 7. Mechanical Generators of the NOR Function

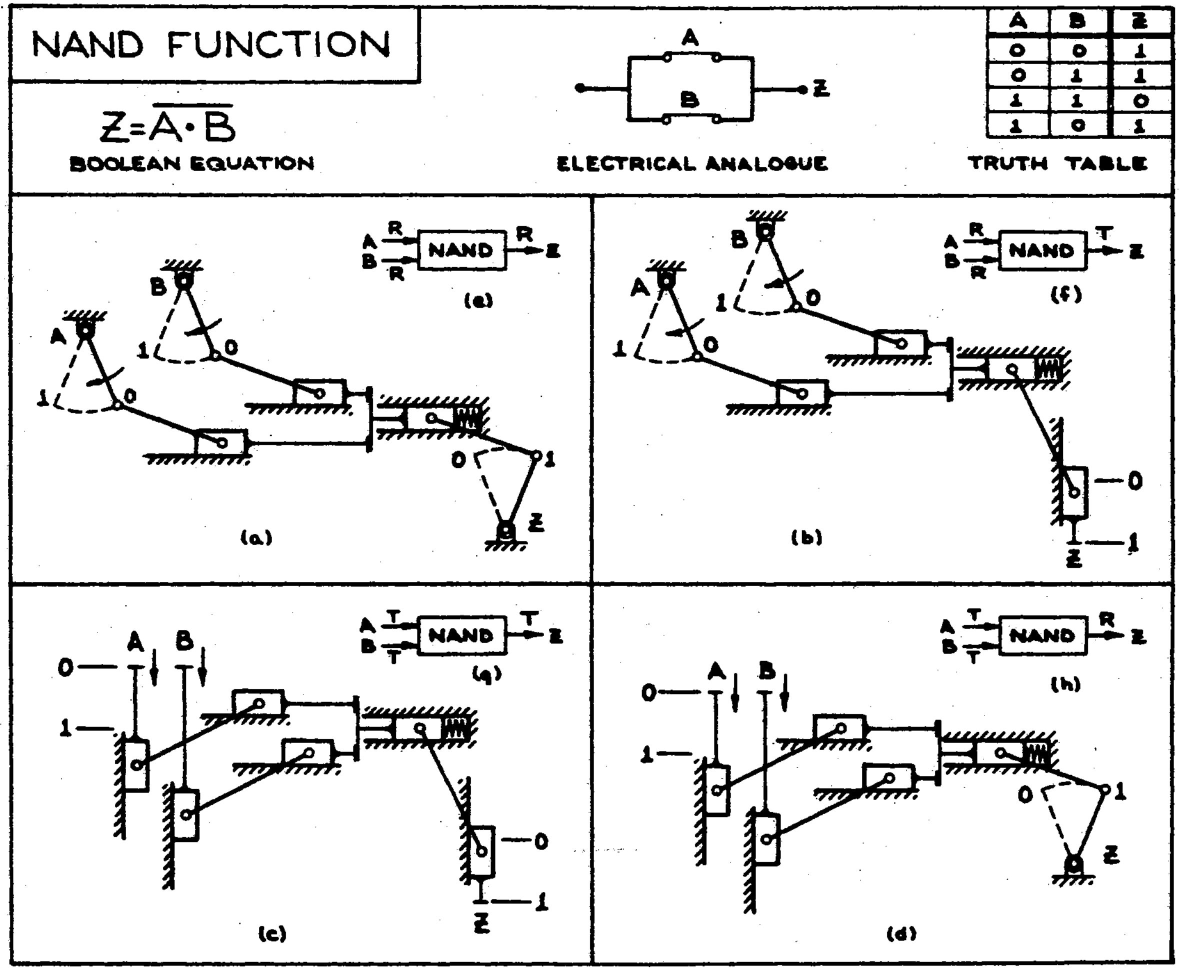


Figure 8. Mechanical Generators of the NAND Function

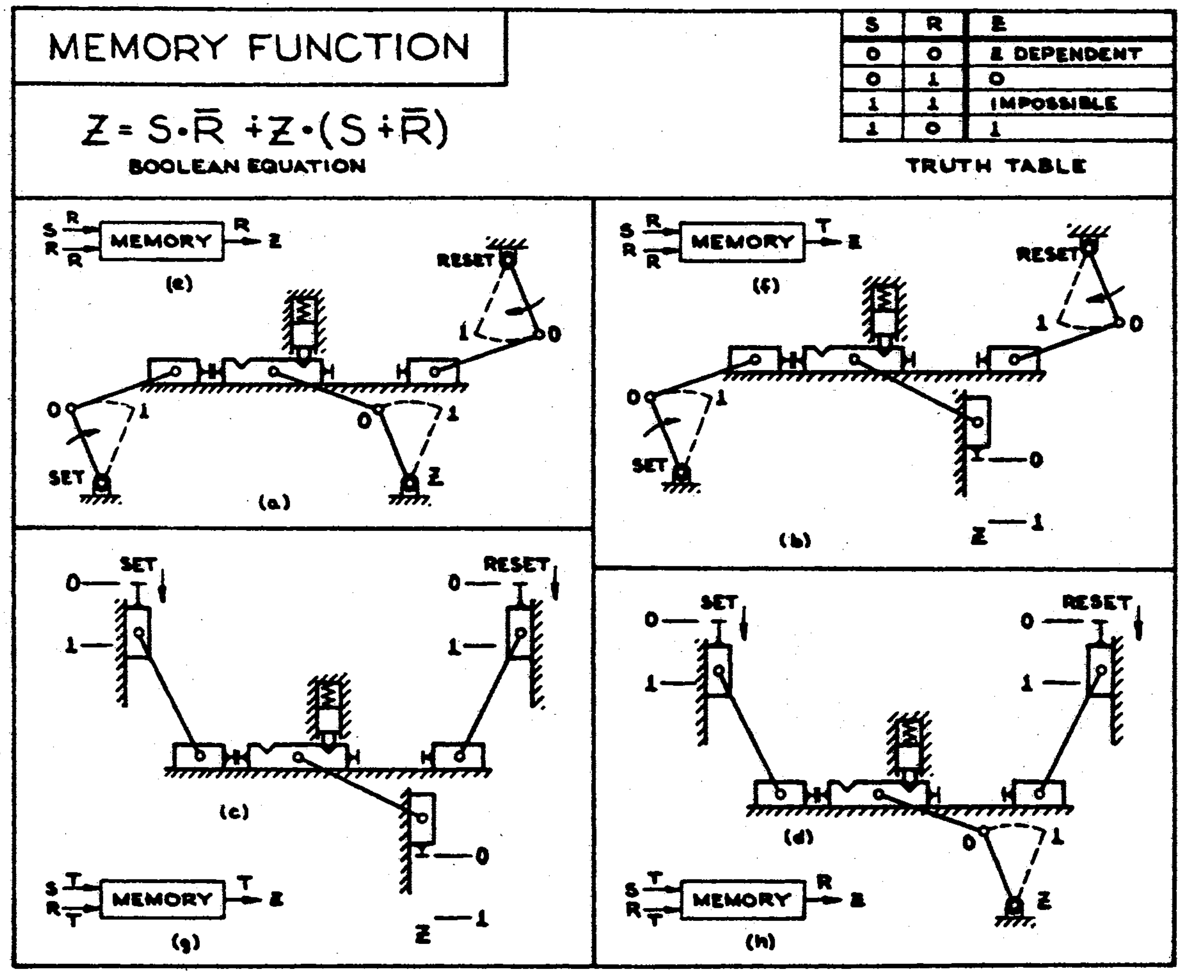


Figure 9. Mechanical Generators of the MEMORY Function

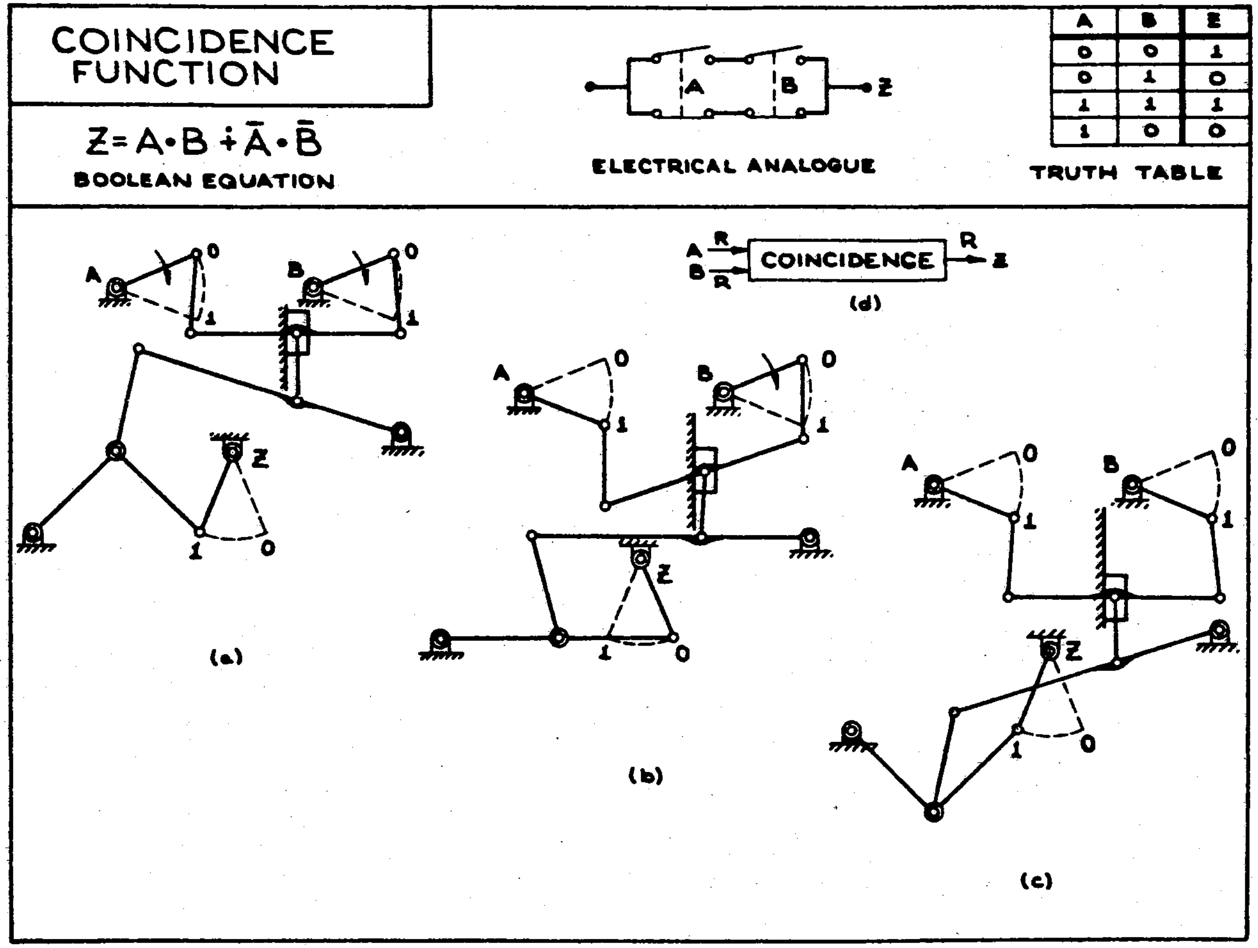


Figure 10. Mechanical Generators of the COINCIDENCE Function

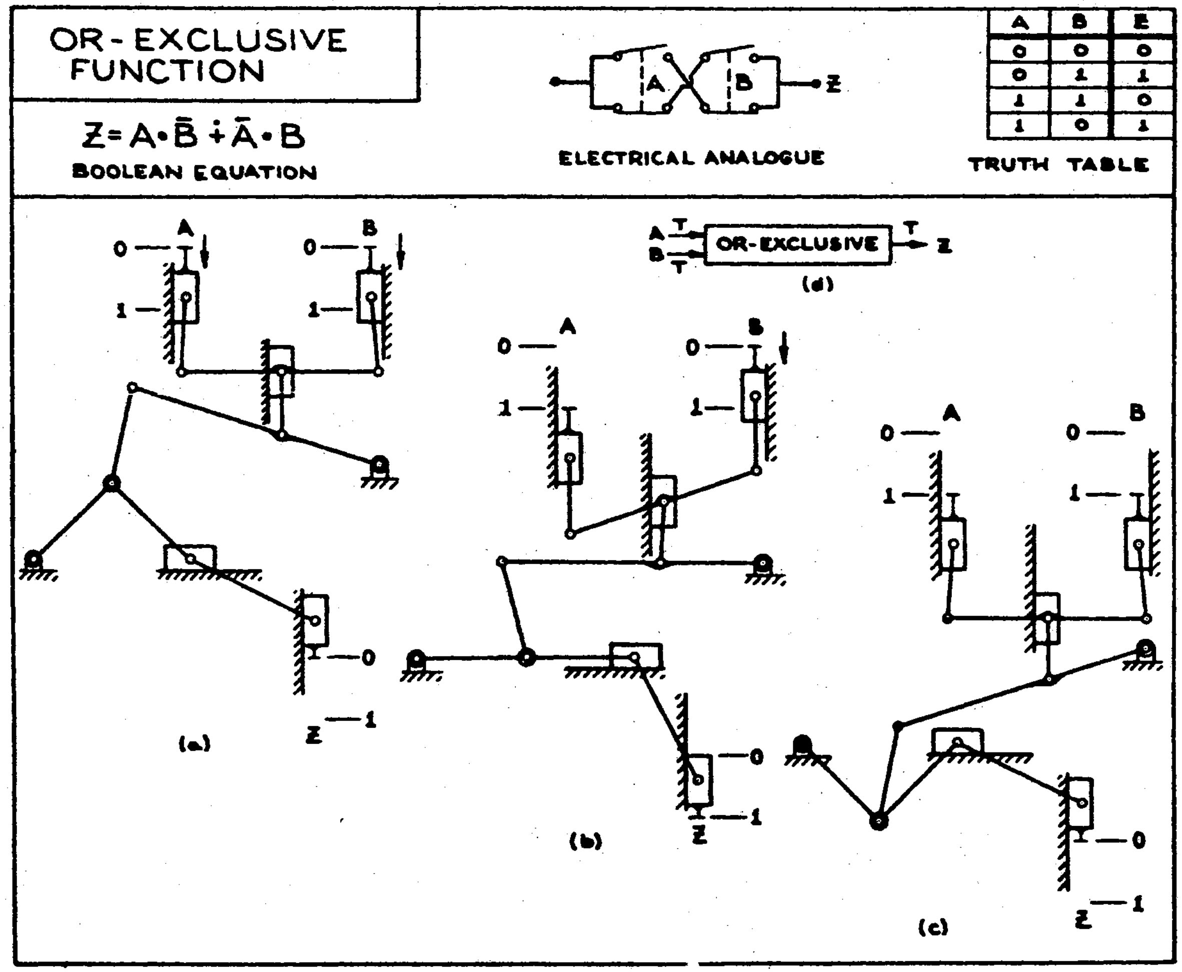


Figure 11. Mechanical Generators of the OR-EXCLUSIVE Function

variable, then the OR function is expressed as

$$Z = A \dotplus B \tag{2.1}$$

The truth table, the electrical analogue, and the mechanical systems that generate the OR function are shown in Figure 4. Examining the truth table and the electrical analogue, it is noted that the OR function exhibits the property that the output Z is "on" when either A or B is "on". The mechanical systems shown at (a), (b), (c) and (d) of Figure 4 demonstrate linkage assemblies displaying the property of the logic function OR. At (a), the variables A, B, and Z execute rotational motion. At (b), the input variables A and B execute rotary motion, while the output variable Z executes translatory motion. At (c), the variables A, B and Z execute translatory motion, and at (d), the input variables A and B execute translatory motion while the output variable Z executes rotary motion.

The mechanical system in Figure 4(a) is built using two crank-slider mechanisms for the variables A and B, and a slider-crank mechanism for the output variable Z. The system at (b) is built using crank-slider and double slider mechanisms. The system at (c) is built using three double slider mechanisms. The mechanical system at (d) is built using two double slider mechanisms and one slider-crank mechanism.

The mechanical systems of Figure 4(a), (b), (c) and (d) are represented schematically using the "black-box" approach in Figure 4(e), (f), (g) and (h). The rotary motions of the variables A, B, or Z are described using the symbol R. The translatory motion of the variables A, B, or Z are described using the symbol T. Thus, for example, Figure 4(e) represents schematically the logic function OR generated by the mechanical system of Figure 4(a).

2.3.2 AND Function

The logic function AND has a minimum of two input variables, and the Boolean operator used to express the AND connective is a dot (•). (The • is often omitted in an algebraic expression since the meaning remains clear.) If A and B are the input variables and Z is the output variable, then the AND function is expressed as

$$Z = A \cdot B \tag{2.2}$$

The truth table, the electrical analogue, and mechanical systems that generate the "AND" logic function are shown in Figure 5. Examining the truth table and the electrical analogue, it is noted that the logic function AND exhibits the property that the output variable Z is "on" only if the variables A and B are both "on". The mechanical systems displaying the logic function AND are shown at (a), (b), (c) and (d) of Figure 5. The corresponding schematic "black-box" notations are shown in Figures 5(e), (f), (g) and (h).

In Figures 4 and 5 it is seen that all input and output variables executing rotary motion turn in a clockwise sense in going from "off" to "on". Also, all input and output variables that execute translatory motion move downward in the figures while going from "off" to "on". These conventions are used for convenience in synthesis of more complicated systems. The conventions are also followed for the generators of the other basic functions.

In all the mechanical systems of Figures 4 and 5 a compression spring is connected to the horizontal output slider. With the OR function, the spring is used to move the output Z from the "on" to the "off" position at any time when all the inputs are moved to the "off" position.

With the AND function, the spring is used to move the output Z from the "off" to the "on" position at any time when all the inputs are moved to the "on" position.

2.3.3 NOT Function

The NOT function has a single input. The Boolean operator which expresses the NOT function is a bar ($\overline{\ }$) over the variable. Thus if A is the input variable and Z is the output variable, the NOT function is expressed as

$$Z = \overline{A} \tag{2.3}$$

The truth table, the electrical analogue and mechanical systems which generate the NOT function are shown in Figure 6. The truth table and the electrical analogue show that the NOT function gives an inversion of the input. The output variable Z is "on" when the input variable A is "off" and vice versa. If the NOT symbol ($\overline{\ }$) is placed over a mathematical variable or expression, then the variable or the expression is said to be "complemented." Thus in Figure 6, Z is the complement of A. The mechanical systems displaying the logic function NOT are shown in Figure 6(a), (b), (c) and (d). The corresponding schematic "black-box" notations are shown in Figure 6(e), (f), (g) and (h).

2.3.4 NOR Function

The NOR function is the complement of the OR function, and has a minimum of two input variables. If A and B are the input variables and Z is the output variable, then the NOR function is expressed as

$$Z = \overline{A \dotplus B} \tag{2.4}$$

The NOR function is presented in Figure 7 in a manner previously described for other functions. It is noted from the truth table that the NOR function exhibits the property that the output variable Z is "on" only when all of the input variables are "off".

2.3.5 NAND Function

The NAND logic function is the complement of the AND function. It also has a minimum of two input variables. If A and B are the input variables and Z is the output variable, then the NAND function is expressed as

$$Z = \overline{A \cdot B} \tag{2.5}$$

The NAND function is presented in Figure 8. It is seen from the truth table that the NAND function exhibits the property that the output variable Z is "on" when either or both of the input variables are "off".

2.3.6 MEMORY (Flip-Flop) Function

There are two input variables to a MEMORY function. They are called the "set" and "reset" variables or "S" and "R", respectively. With Z as the output variable, the MEMORY function is expressed as

$$Z = S \cdot \overline{R} \dotplus Z \cdot (S \dotplus \overline{R}) \tag{2.6}$$

The truth table and mechanical systems which generate the MEMORY function are shown in Figure 9. The Boolean equation states that the output variable Z is "on" when the "set" variable is "on" and the "reset" variable is "off". It also states that the output is maintained "on" as long as the "set" variable remains "on" or the "reset" variable remains "off". The output variable can go to "off" only when the "reset" variable goes to "on". Thus in the case when both the "set" and "reset" variables are "off", the output Z is "on" if it was previously "on" or it is "off" if it was previously "off". As stated in the truth table and as can be observed by examining the mechanical systems at (a), (b), (c) and (d) of Figure 9, the "set" and "reset" variables cannot both be "on" at the same time. This makes it possible to simplify the Boolean equation to

$$Z = S \dotplus Z \cdot \overline{R} \tag{2.7}$$

The corresponding schematic "black-box" notations are shown in Figure 9(e), (f), (g) and (h).

The detent spring shown in Figure 9(a), (b), (c) and (d) provides enough force so that any external forces or torques on the output Z will not cause the output to shift positions when both the S and R inputs are in their "off" positions.

2.3.7 COINCIDENCE Function

The COINCIDENCE function has two input variables, A and B, and one output variable Z. Mathematically, the COINCIDENCE function is expressed as

$$Z = A \cdot B \dotplus \overline{A} \cdot \overline{B} \tag{2.8}$$

The truth table, the electrical analogue, and a mechanical system displaying the CONCIDENCE function are shown in Figure 10. Examination of the truth table and the electrical analogue reveals the property that the output variable Z is "on" only when the input variables A and B are both "on" or both "off". A mechanical system where the input and the output variables execute rotary motion is shown at (a), (b) and (c) of of Figure 10. The three figures show the system in three of its four possible positions: (a) A and B are both "off"; (b) A is "on" and B is "off"; and (c) A and B are both "on". At (d) the corresponding schematic "black-box" notation is shown.

2.3.8 OR-EXCLUSIVE Function

The OR-EXCLUSIVE function is the complement of the COINCIDENCE function, and it has two input variables. If A and B are the input variables and Z is the output variable, the OR-EXCLUSIVE function is expressed as

$$Z = A \cdot \overline{B} \dotplus \overline{A} \cdot B \qquad (2.9)$$

The truth table, the electrical analogue, and a mechanical system are shown in Figure 11. It can be seen that the output variable Z is "on" when either of the variables A or B is "on", but Z is "off" when both A and B are "on". A mechanical system where the input and output variables execute translatory motion is shown at (a), (b) and (c) of Figure 11. These figures show the system in three of its four possible positions. The corresponding schematic "black-box" notation is shown at (d).

2.4 Theorems of Boolean Algebra Proved Mechanically

A list of the theorems of Boolean algebra that are important to the synthesis of mechanical systems is given in Figure 12. These theorems are useful in obtaining equivalent systems as well as in simplifying existing systems. Each theorem is proved using a mechanical system and an electrical analogue.

The mechanical systems are shown with simplified symbols. All input and output motions are of translatory type which are obtained through

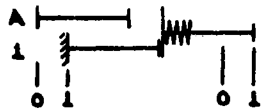
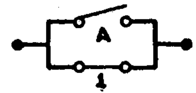
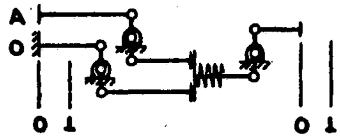
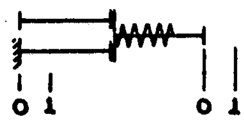
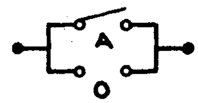
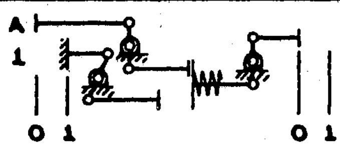
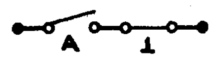
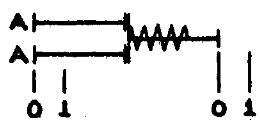
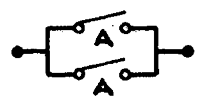
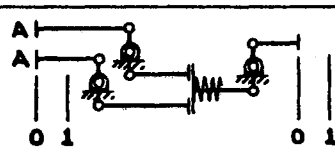
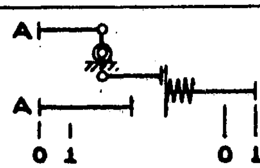
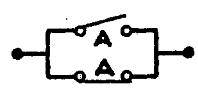
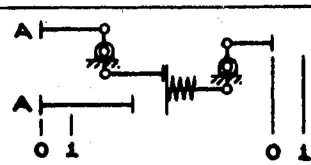
Nº	THEOREM	MECHANICAL SYSTEM	ELECTRICAL ANALOGUE
1	$A \dotplus 1 = 1$		
2	$A \cdot 0 = 0$		
3	$A \dotplus 0 = A$		
4	$A \cdot 1 = A$		
5	$A \dotplus A = A$		
6	$A \cdot A = A$		
7	$A \dotplus \bar{A} = 1$		
8	$A \cdot \bar{A} = 0$		

Figure 12. Twenty-Three Theorems of Boolean Algebra Proved by Mechanical and Electrical Analogues

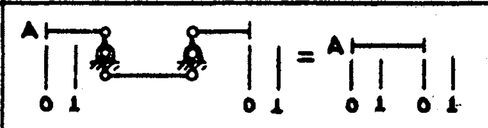
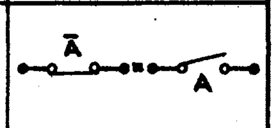
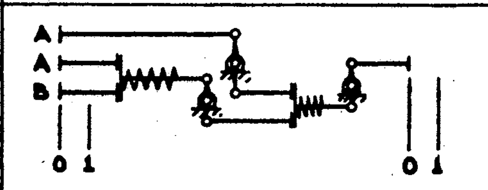
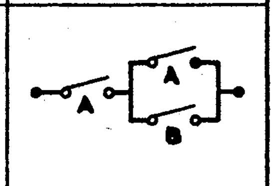
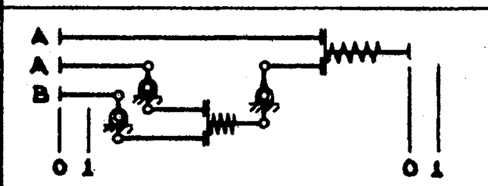
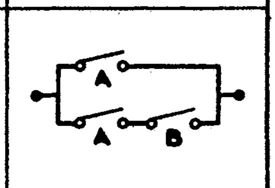
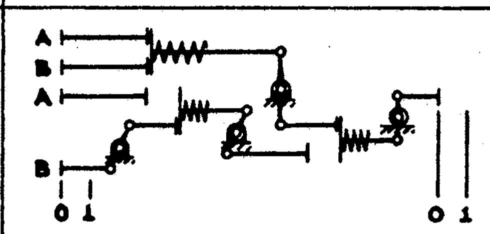
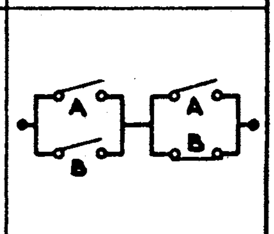
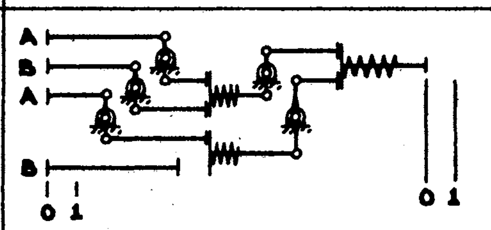
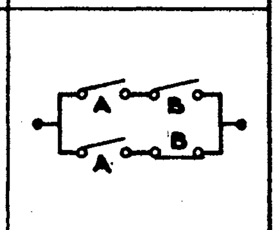
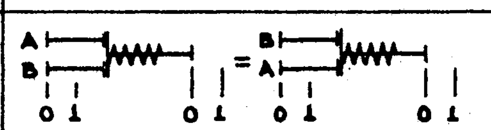
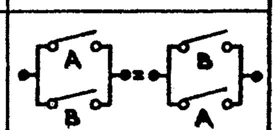
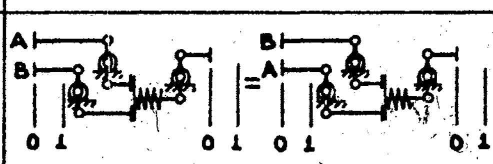
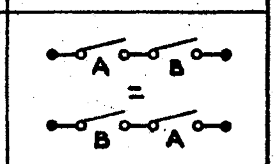
Nº	THEOREM	MECHANICAL SYSTEM	ELECTRICAL ANALOGUE
9	$\overline{(\overline{A})} = A$		
10	$A(A \dot{+} B) = A$		
11	$A \dot{+} A \cdot B = A$		
12	$(A \dot{+} B)(A \dot{+} \overline{B}) = A$		
13	$A \cdot B \dot{+} A \cdot \overline{B} = A$		
14	$A \dot{+} B = B \dot{+} A$		
15	$A \cdot B = B \cdot A$		

Figure 12. (Continued)

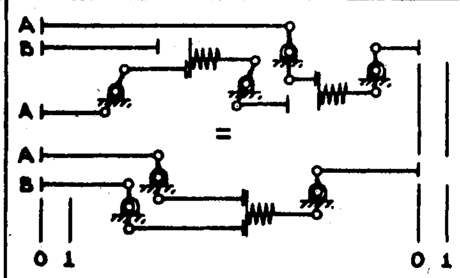
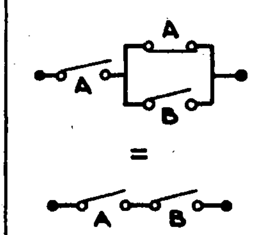
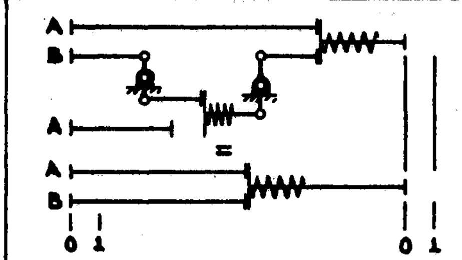
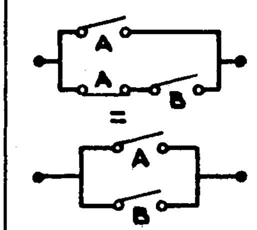
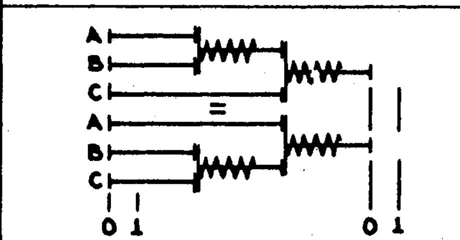
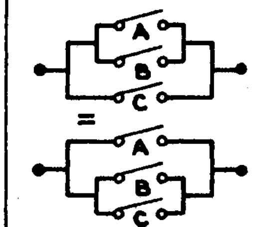
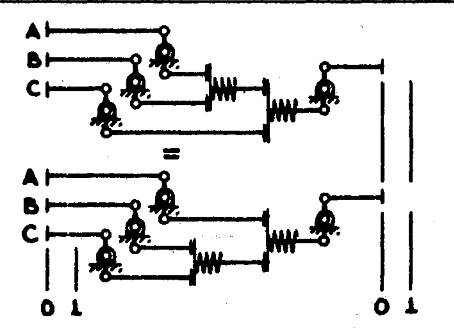
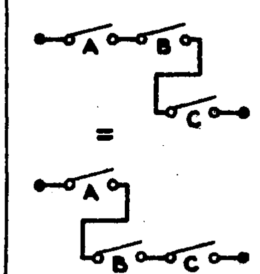
Nº	THEOREM	MECHANICAL SYSTEM	ELECTRICAL ANALOGUE
16	$A(\bar{A} \dotplus B) = A \cdot B$		
17	$A \dotplus \bar{A} \cdot B = A \dotplus B$		
18	$(A \dotplus B) \dotplus C = A \dotplus (B \dotplus C)$		
19	$(A \cdot B)C = A(B \cdot C)$		

Figure 12. (Continued)

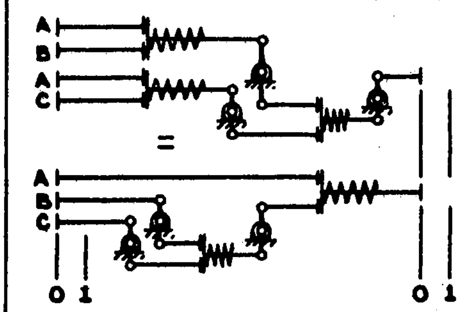
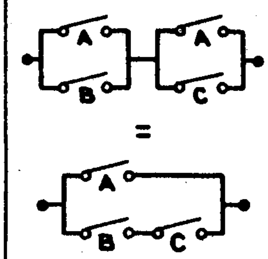
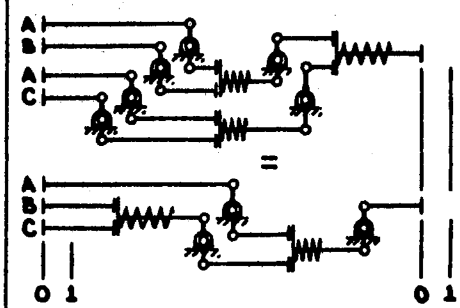
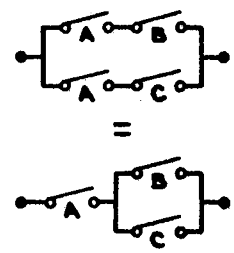
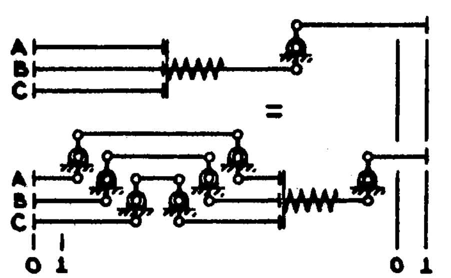
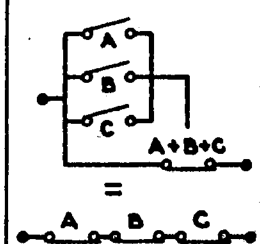
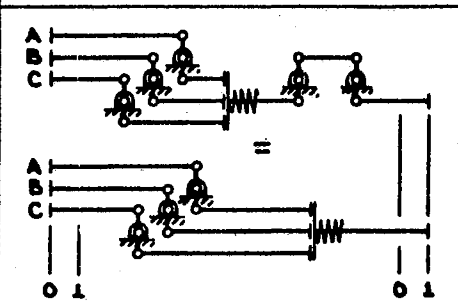
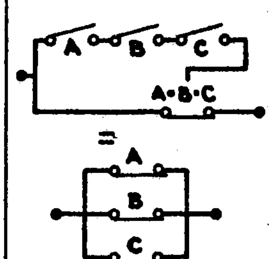
Nº	THEOREM	MECHANICAL SYSTEM	ELECTRICAL ANALOGUE
20	$(A \dotplus B)(A \dotplus C) = A \dotplus B \cdot C$		
21	$A \cdot B \dotplus A \cdot C = A(B \dotplus C)$		
22	$\overline{(A \dotplus B \dotplus C)} = \bar{A} \cdot \bar{B} \cdot \bar{C}$		
23	$\overline{(A \cdot B \cdot C)} = \bar{A} \dotplus \bar{B} \dotplus \bar{C}$		

Figure 12. (Continued)

sliding blocks. These sliding blocks are not shown in the diagrams of Figure 12. In these diagrams the input variables are indicated with the letters A, B, and C.

The theorems from 1 to 13 degenerate a system to a value of 0, 1, or A. The theorems 14, 15, 18 and 19 present equivalent mechanical systems. The theorems 16, 17, 20 and 21 present simplified mechanical systems having fewer elements, and the theorems 22 and 23 resolve a mechanical system into its functional components.

2.5 Example of Logic Circuit Synthesis

On an assembly line a conveyor is transporting soap boxes of rectangular shape. Some of the boxes need to be sorted from the conveyor according to their dimensions. A "sorter mechanism" is available to transfer the desired boxes off the main conveyor onto an auxiliary conveyor (see Figure 13). A translatory displacement signal Z into the sorter mechanism triggers the rejection operation. Four oscillating cam followers are available to detect the dimensions of the boxes and serve as logic variables A, B, C and D. The boxes themselves act as cams when they pass by the followers.

A mechanical control circuit is designed that uses A, B, C and D as input variables and produces the desired output signal (variable Z).

There are four general steps that are used in the synthesis procedure for designing a logic circuit employing basic logic functions such as those of Figures 4 to 11. The four steps are:

1. Write a description of the output requirements and the input conditions. (Such a description must be logically meaningful, unambiguous and non-contradictory.)

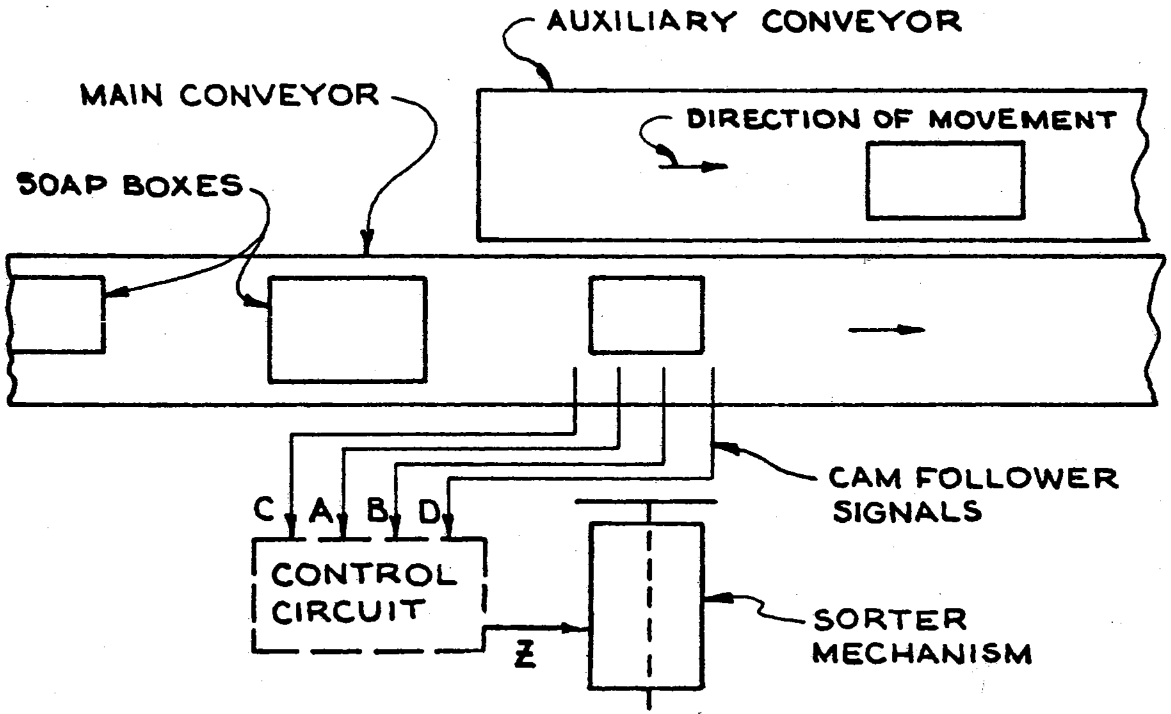


Figure 13. Conveyor System With Sorter Mechanism

2. Devise a method to correctly assign the state of the output variable for every possible combination of the input states.

3. Write a simplified Boolean equation for each output variable.

4. Build a schematic mechanical control circuit.

The first three of these steps are not dependent upon the fact that the logic circuit is synthesized from linkage elements; the three steps are the same whether the circuit is electronic, fluidic, a linkage or whatever. Since the first three steps are not basic to this study they, as applied to the present example, are included in Appendix A.

Step 4 begins with the result of step 3. From Appendix A the simplified Boolean equation for the control circuit of the soap box sorter is

$$Z = CD + A\bar{B}D + \bar{A}BC \tag{2.10}$$

The simplified logic equation could be implemented by using two NOT functions, three AND functions, and one OR function. The need of the two NOT functions is eliminated, however, by observing the equivalence shown in Figure 14. Using one AND system, two AND-NOT systems, and one OR system, the required control circuit is shown in Figure 15(a).

Logic systems with translatory input and output motions are selected in the final design of the mechanical logic system. The rotary motion of the input cam follower variables is converted to translatory motion with the crank-sliders shown in the upper part of Figure 15(a). A vertical slider is used for each of the input variables A, B, C and D. The horizontal sliders of each function are then connected to the vertical sliders as needed. The vertical output sliders of the AND and AND-NOT systems become the inputs to the OR system. Finally the output of the OR system is the required output variable Z.

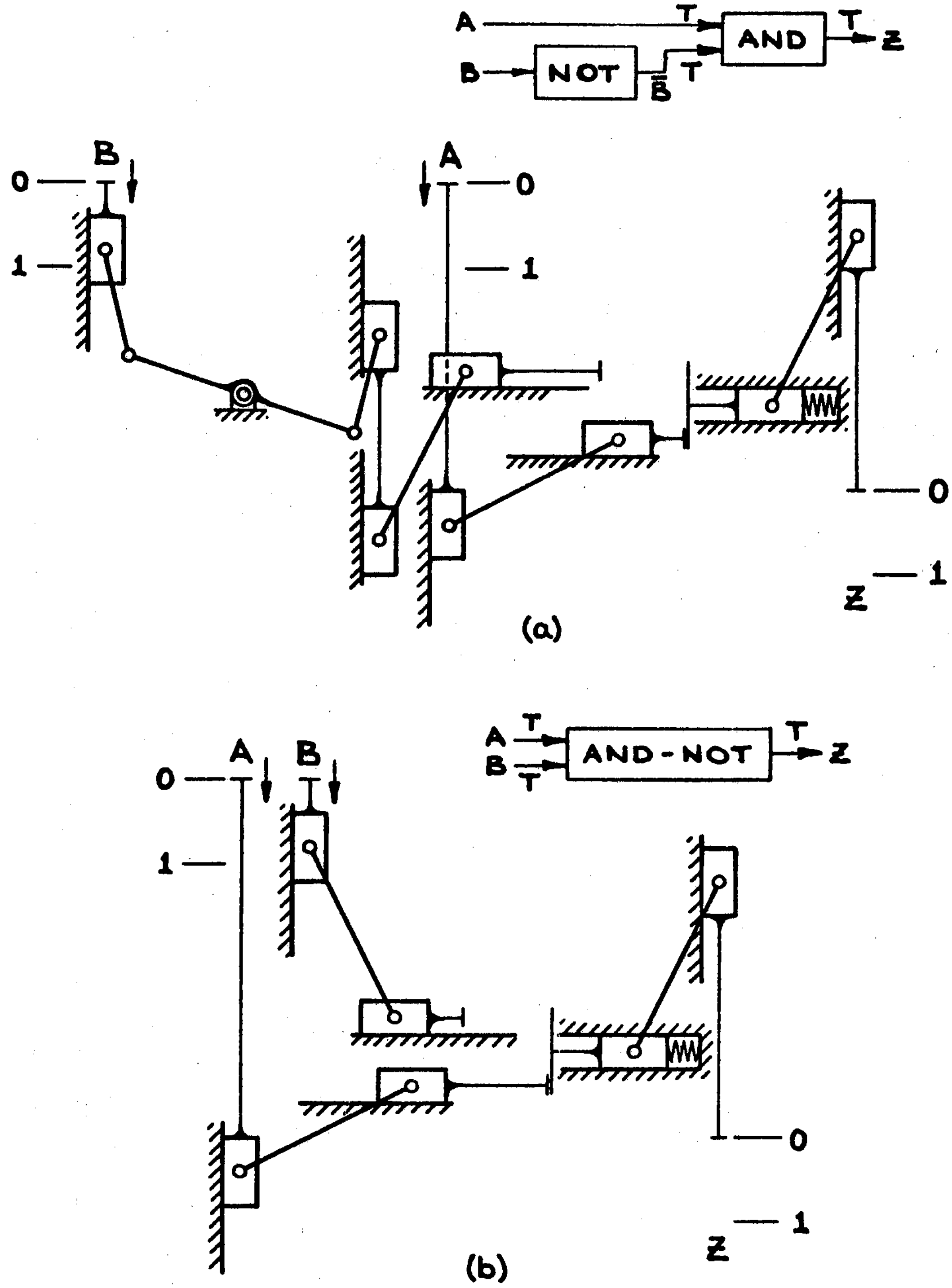


Figure 14. Equivalent Systems for Generating the Logic Function $Z = A \cdot \bar{B}$

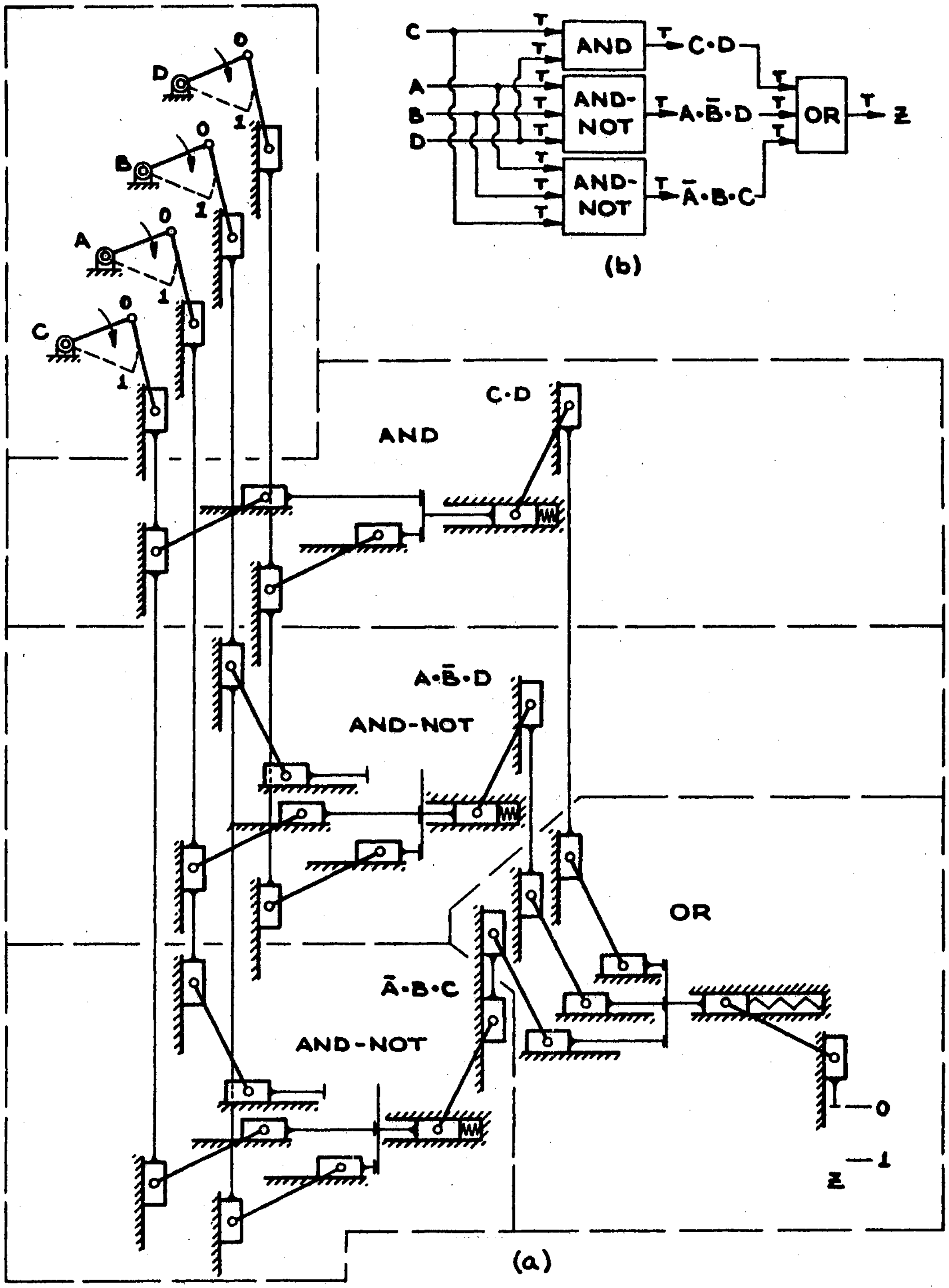


Figure 15. Control Circuit for Sorter Mechanism

The input variables A, B, C and D are shown in their "off" positions and all other inputs and outputs of the AND, AND-NOT, and OR systems are shown in their corresponding positions. In this case, all of the inputs and outputs are in the "off" positions, but this situation is not true with all circuits.

The corresponding schematic "black-box" notation for the system is shown in Figure 15(b).

2.6 Active Function Generators

The control circuit of the previous example is one of combinational logic (where the state of the output depends only upon the present state of the input variables). All of the mechanical logic systems presented are of a "passive" nature. They are passive because all of the energy entering or leaving the systems enters or leaves at the inputs and outputs. There is no force or power amplification at any point in the system.

For complex circuits, especially for sequential circuits where memory systems are needed, the passive elements are inadequate. (Sequential logic is the logic in which the state of the output depends not only upon the present values of the input variables but also upon their values at some previous time.) For the complex circuits, "active" logic systems are needed. Active logic systems are systems in which there is power amplification as well as the generation of the desired logic.

Each active mechanical logic system presented here is made up of two components. The first component is the mechanism (referred to as a function generator) which actually generates the binary logic function. The second component is a power amplifier. The output of the function generator is connected to the input of the amplifier, and thus the output of the amplifier becomes the output of the active logic system.

The function generators are shown in Figure 16. They are basically the same as the mechanical systems shown in Figures 4 through 9. The function generators whose inputs and outputs exhibit translatory motion, however, do not employ vertical sliders. The inputs and outputs are applied directly to the horizontal sliders for convenience in interconnecting the function generators.

In Figure 16(a) the OR function generator is shown. Its inputs A and B and output Z are translatory. The circles on the sliders are used to indicate the points where binary links are connected with revolute pairs. These binary links make the interconnections with other function generators and amplifiers. The schematic "black-box" representation is also shown.

The AND function generator is shown in Figure 16(b). It has the rotary inputs and output A, B and Z. The circles at A, B and Z also represent the points where the interconnecting binary links are attached. The three ternary links act as inverters and have a grounded revolute pair at their centers. The "off" and "on" positions of these ternary links are symmetrical about vertical lines. It should be noted that the displacements of the points A, B and Z of the AND generator in going from the "off" to the "on" positions is the same as the displacements of the points A, B and Z of the OR generator. This design convention is convenient for interconnecting the function generators. It is also noted that the displacement of all input and output points of the function generators is from left to right while changing from "off" to "on".

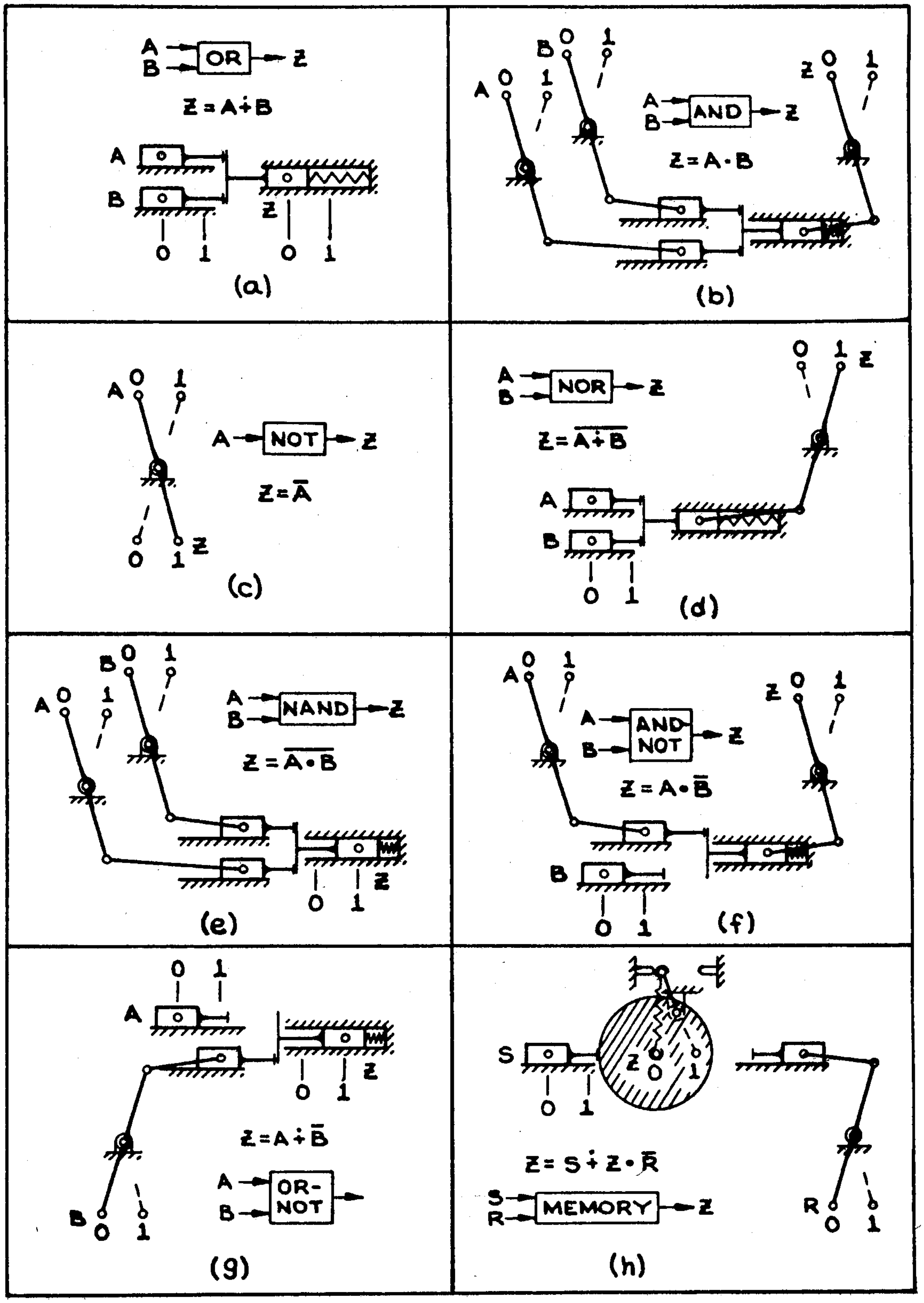


Figure 16. Function Generators for Active Mechanical Logic Systems

The NOT function generator is shown in Figure 16(c). It is a single ternary link with a grounded revolute pair at its center. Since the input A moves from left to right in going from "off" to "on", the output acts inversely and is the complement of A.

The NOR, NAND, AND-NOT and OR-NOT function generators are shown in Figure 16(d), (e), (f) and (g), respectively. It is observed that each of these generators satisfies the conventions described in the previous paragraph. A schematic "black-box" representation is also shown for each generator.

The MEMORY function generator is shown in Figure 16(h). The S (Set) input executes translatory motion, but the R (Reset) input executes rotary motion. The output Z is one link of a double toggle and thus executes rotary motion. This output link is made circular and provides a smooth cam motion.

The AMPLIFIER component is shown in Figure 17. It has a source of external energy for power amplification. The source is of any type which will rotate the powered rollers R_1 and R_2 continuously. The input to the AMPLIFIER is A and executes translatory motion. The output is Z and it executes rotary motion. Connected to the input slider S are two friction rollers F_1 and F_2. The output toggle link T has a cylindrical surface C whose axis is Q. The surface C has a relief indentation at I. In Figure 17 the AMPLIFIER is shown in the "off" position.

The AMPLIFIER functions as follows:

1. The input A, which is moved by the output of some function generator, moves from the "off" to the "on" position. Upon reaching the "on" position the friction roller F_1 makes contact with both the powered roller R_1 and the output toggle link T.

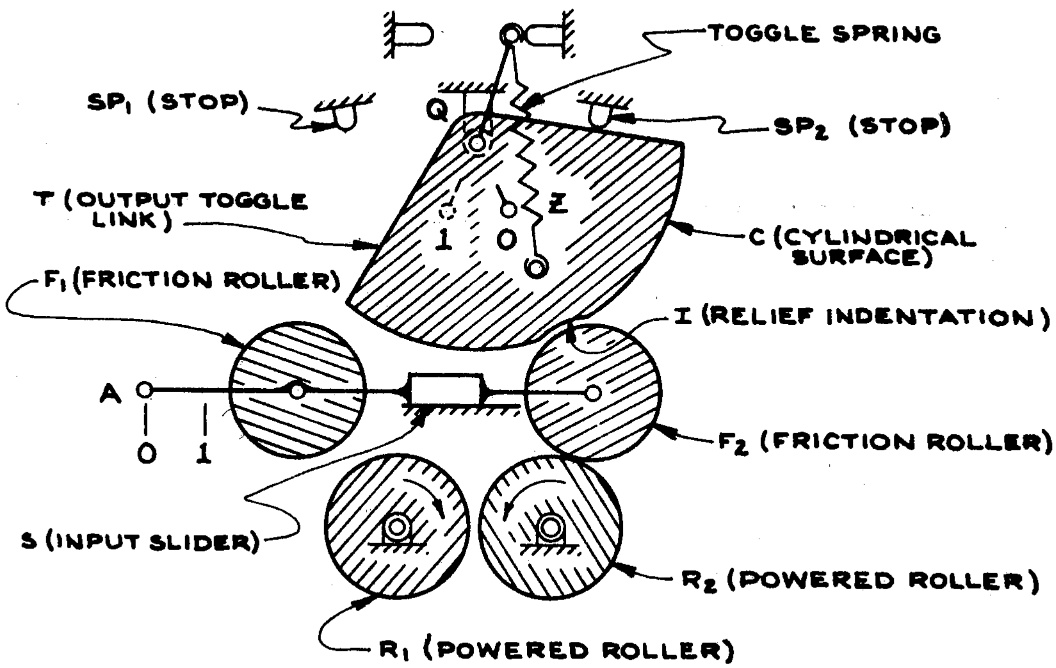


Figure 17. Amplifier for Active Mechanical Logic Systems

2. The clockwise rotation of the powered roller R_1 is transmitted through the friction roller F_1 to the output toggle link T causing T to rotate in a clockwise direction.

3. Just before the output toggle link T reaches the stop Sp_1, the relief indentation I reaches the friction roller F_1 and contact is terminated between F_1 and T.

4. The momentum of the output toggle link T and the action of the toggle spring carry T on to contact the stop Sp_1. The output Z is now in the "on" position.

5. When the AMPLIFIER returns from "on" to "off", an action occurs which is similar to that of steps 1 to 4. It involves the action of the powered roller R_2, the friction roller F_2 and the stop Sp_2.

Because of the design of the AMPLIFIER, the friction roller F_1 is "wedged in" between the powered roller R_1 and the output toggle link T as soon as contact is made by F_1 with both R_1 and T. This action provides the normal and friction forces needed to rotate the output toggle link T. No force is needed at A after contact is made by F_1 with both R_1 and T. Thus the only force needed at the input A to actuate the AMPLIFIER is the small force necessary to overcome the friction of the input slider S.

The force that can be developed at the output Z is limited by 1) the friction force that can be transmitted by the friction roller, 2) the resistance of the toggle spring, and 3) the dynamics of the system. The output Z is connected to one or more inputs of other function generators or to the output of the logic circuit. The toggle spring must provide enough force to hold the output toggle link T against the stop (Sp_1 or Sp_2) while the AMPLIFIER is in a static situation.

The toggle spring must, therefore, provide enough force to overcome the resistance at the inputs of the other function generators or the resistance at the output of the circuit.

Very little has been done with respect to determining amplification ratios. However, as the model, presented later in this chapter, was being constructed, it was found that the output from an AMPLIFIER is capable of controlling at least three function generator imputs.

2.7 Example of Active Logic Circuit Synthesis

The speed of a rotating shaft which has a variable torque requirement is maintained nearly constant by turning a motor on and off as needed. A supercharger is available to provide additional torque if the motor by itself is not sufficient to maintain the desired speed. The signal to turn the motor on is Z_1 and the signal to activate the supercharger is Z_2.

The speed of the shaft is sensed with the device shown in Figure 18. The device provides the two binary signals x_1 and x_2. When the shaft is rotating at the desired speed, the signal combination is $x_1x_2 = 00$. As the shaft speed increases the signals become $x_1x_2 = 10$ and then 11. A reduction in speed causes a signal combination of 01, but, as seen in the figure, a signal combination of 11 cannot be reached by reducing the speed.

A mechanical control circuit is designed that uses the signals x_1 and x_2 as input variables and produces the desired output signals Z_1 and Z_2.

The four general steps described in the synthesis procedure of the example of section 2.5 are observed. As before, the first three steps

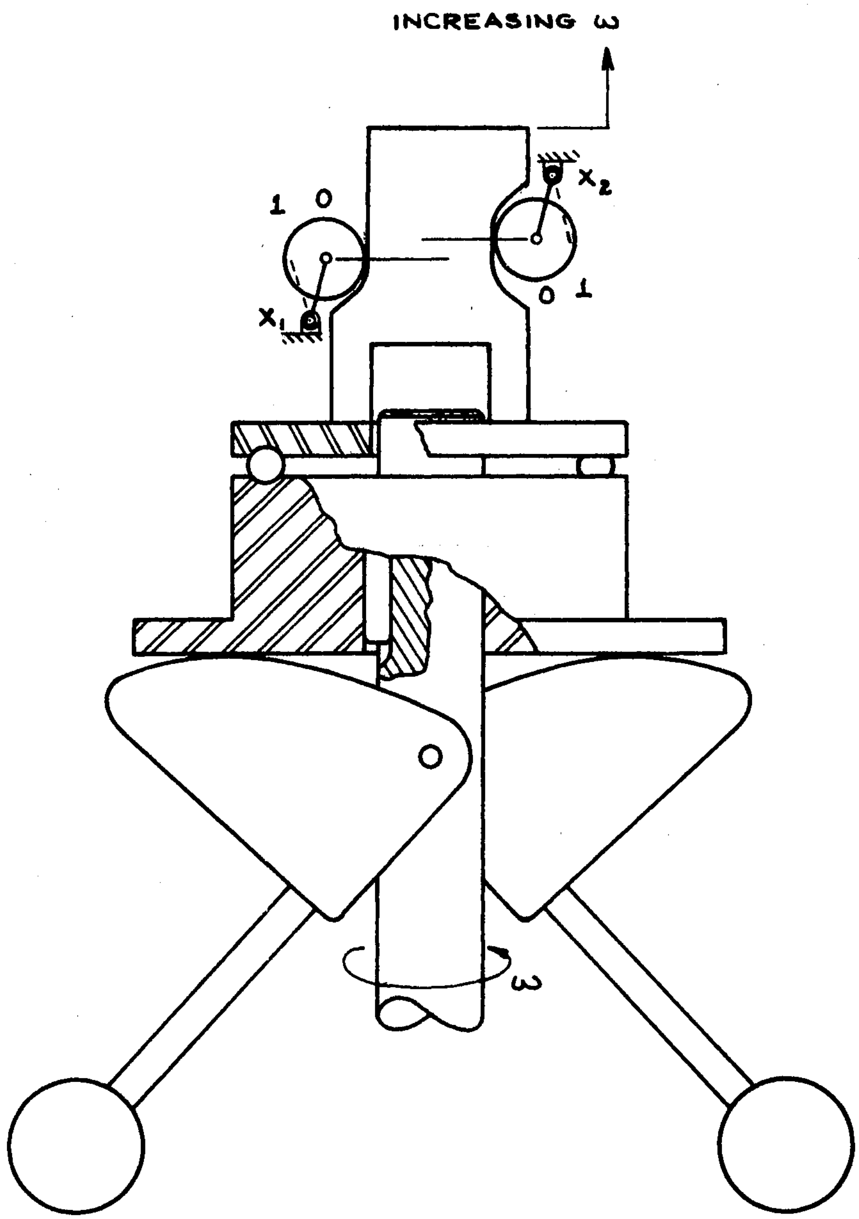


Figure 18. Shaft Speed Sensor and Input Signal Generator

are not basic to this study and for this example are presented in Appendix B.

This shaft speed control circuit, as explained in Appendix B, is one of sequential logic and requires two MEMORY function generators. The outputs from the MEMORY functions are y_1 and y_2. From step 3 of Appendix B, the outputs Z_1 and Z_2 are given in terms of the inputs x_1 and x_2 and the outputs of the MEMORY functions y_1 and y_2 as

$$Z_1 = y_2$$

and (2.11)

$$Z_2 = \overline{x}_1(x_2 \dot{+} \overline{y}_1)$$

Also from Appendix B, the equations for the MEMORY functions are

$$Y_1 = x_2 \dot{+} y_1(\overline{x_1 \cdot \overline{x}_2 \cdot y_2})$$

and (2.12)

$$Y_2 = \overline{x}_1 \dot{+} y_2(\overline{x_1 \cdot x_2})$$

From these equations of the MEMORY functions, the SET and RESET inputs to the two MEMORY function generators are written as

$$S_1 = x_2$$

$$R_1 = x_1 \cdot \overline{x}_2 \cdot y_2$$

$$S_2 = \overline{x}_1$$

and

$$R_2 = x_1 \cdot x_2 \qquad (2.13)$$

Finally, step 4 of the synthesis procedure is to build a <u>schematic mechanical control circuit</u>. The "black-box" schematic of the required control circuit is shown in Figure 19. The way in which the output and secondary equations are implemented can be observed from the figure.

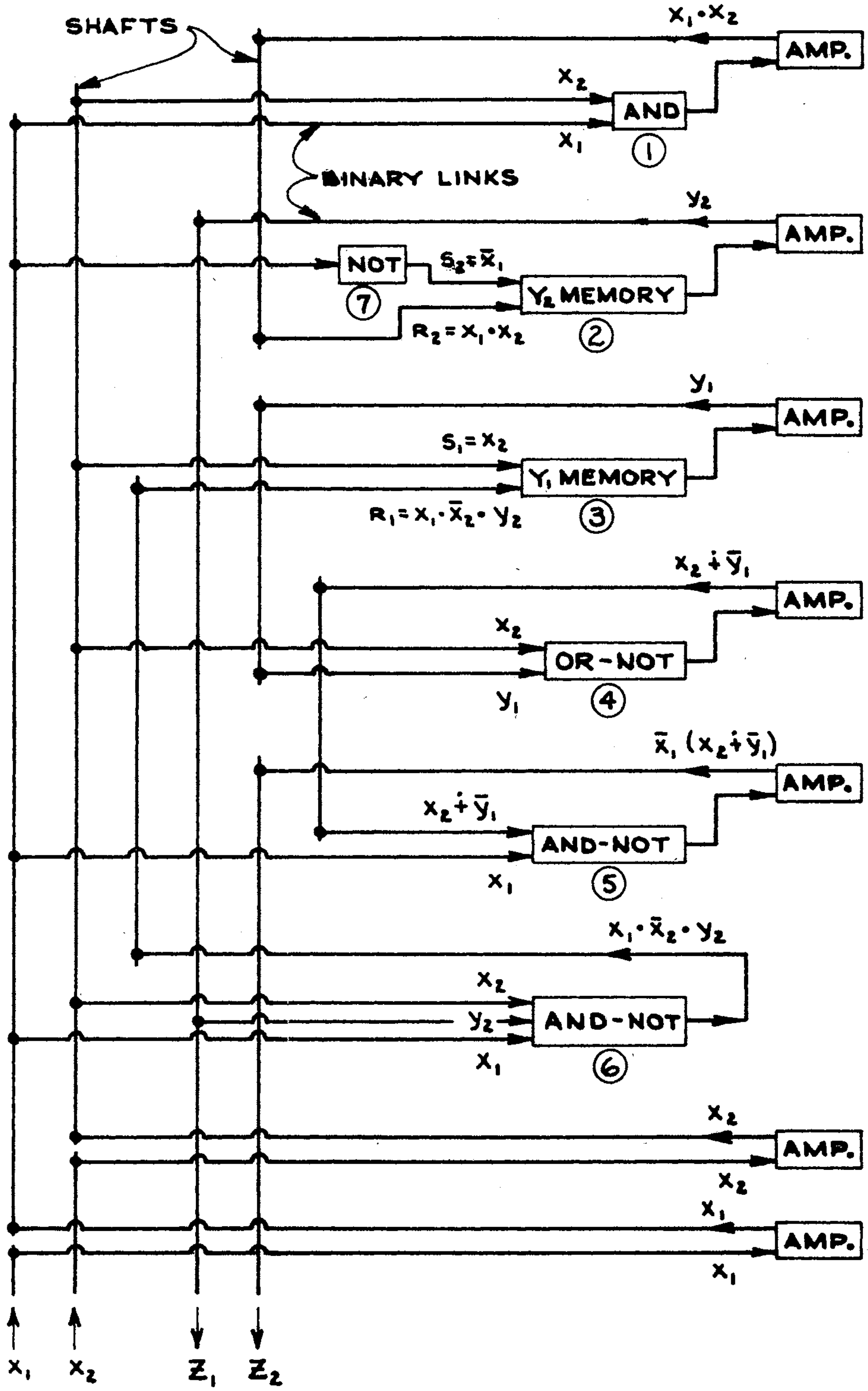


Figure 19. Schematic of the Speed Control Logic Circuit

The seven function generators used are numbered and they consist of one AND generator, one OR-NOT generator, two AND-NOT generators, two MEMORY generators and one NOT generator. The function generators numbered from 1 to 5 are seen to be active systems since their outputs are fed through AMPLIFIERS. A passive generator is used for the NOT function as it is simply an inversion process.

The AND-NOT generator (6) is made passive to avoid a problem of having the SET and RESET signals of the Y_1 MEMORY "on" at the same time in one particular sequence of signal changes. A detailed explanation of this situation is given in Appendix B, section B.2.

In the model of the control system shown in the photograph of Figure 20 the logic function generators are not all in one plane as suggested in the schematic of Figure 19. The generators are "stacked" in parallel planes. This arrangement allows the friction rollers (see Figure 17) of all the AMPLIFIERS to contact the same pair of powered rollers.

In the logic circuit, physical connections are made between inputs, function generators, amplifiers and outputs. As seen in the photograph of Figure 20, these physical connections are made through shafts whose axes are perpendicular to the planes of the function generators. Binary links are used between the shafts and the function generators. In Figure 19 the representation of these shafts and binary links can be noted.

It can also be noted in the photograph that the AMPLIFIERS of the input signals x_1 and x_2 and the AMPLIFIER of the AND-NOT generator (5) are not included. Since the model in the photograph was not actually connected to a motor circuit, these AMPLIFIERS were not necessary.

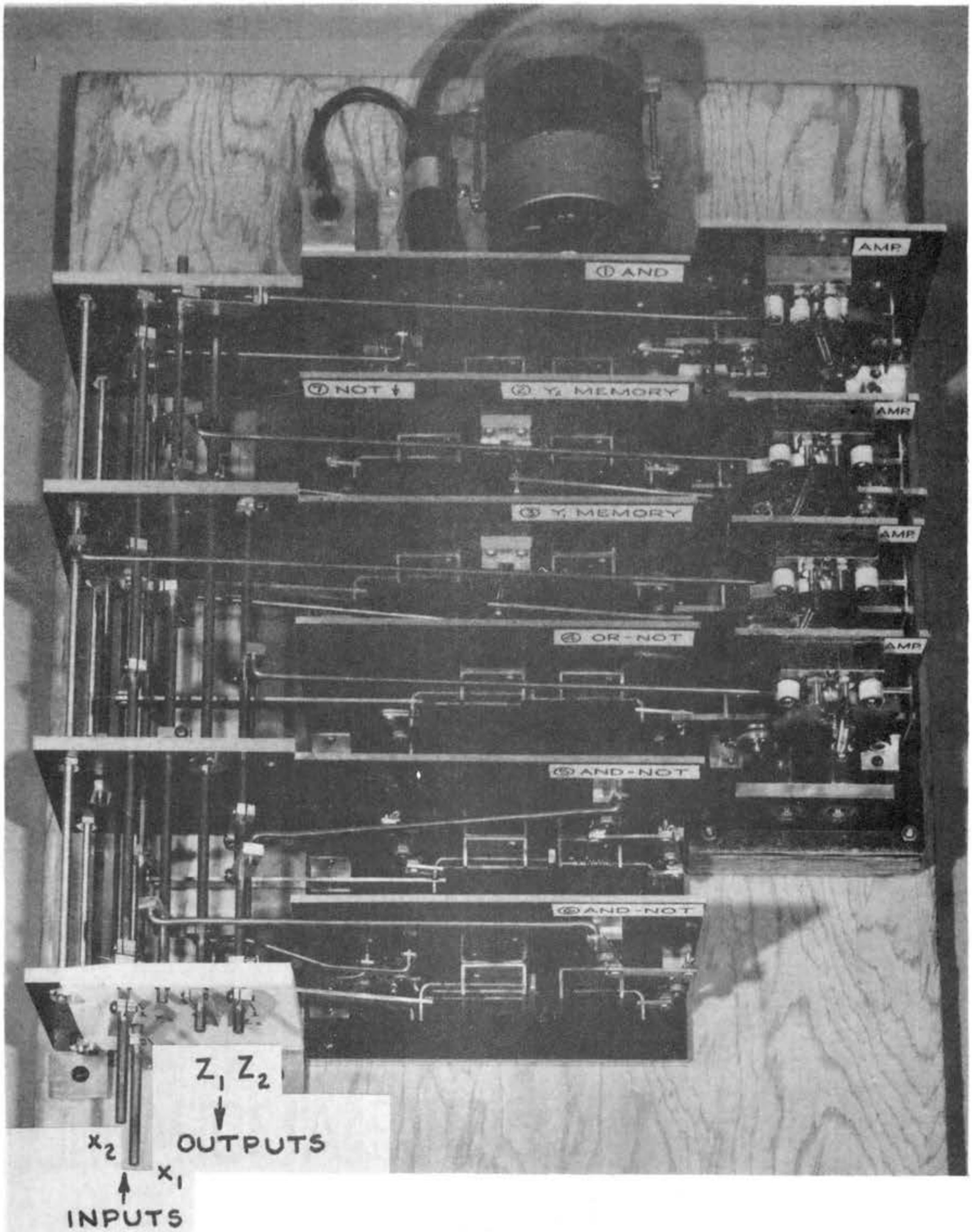


Figure 20. Model of Speed Control Circuit

CHAPTER III

SYNTHESIS OF THRESHOLD LOGIC GENERATORS FOR FUNCTIONS OF THREE VARIABLES

In Chapter II linkage generators of Boolean functions were treated in conventional logic form. In the present chapter the concept of threshold logic is employed to synthesize mechanical generators for all the functions of three variables. Definitions and the basic theory of threshold logic is presented. An example is presented which illustrates the type of linkage used for the threshold logic generators. Structural synthesis is employed to search for alternative linkages which might be used. Performance criteria is established and used to adopt a basic linkage for the generators. Steps are taken to recognize a limited number of basic linkages needed to generate all the functions of three variables and then these linkages are synthesized dimensionally.

Design charts are presented to aid in the direct selection of a linkage to generate any one of the functions of three variables. Two examples are presented to illustrate the use of the design charts.

3.1 Definitions and Theory of Threshold Logic

Boolean functions can be represented as threshold functions. Some functions are single-threshold threshold functions and are known simply as "threshold functions". Others are "multithreshold threshold functions" and are known as such. The threshold functions are considered first.

For each combination of values of the input variables x_1, x_2, . . ., x_n, the corresponding value of the output z is

$$
\begin{aligned}
z &= 1, \text{ if } \sum_{j=1}^{n} w_j x_j \geq T \\
z &= 0, \text{ if } \sum_{j=1}^{n} w_j x_j < T
\end{aligned}
\tag{3.1}
$$

where

T = the threshold

w_j = weights or coefficients; both T and w_j are fixed positive or negative real numbers.

The notation

$$z = \left\langle \sum_{j=1}^{n} w_j x_j \right\rangle T \tag{3.2}$$

is used to represent Equation (3.1) [14].

A geometrical interpretation helps to make these definitions clear. In the geometrical interpretation, each of the Boolean function's n input variables is represented by a coordinate axis of a n-dimensional Euclidean space. Each combination of the input variables is represented by the corner of a unit hypercube (n cube) in the n-space. Consider the two-input variable function

$$z = \bar{x}_1 \dot{+} x_2 \tag{3.3}$$

represented in Figure 21(a). The corners of the 2 cube (square), where z has the values 0 and 1, are called the sets of points P_0 and P_1, respectively. The vertices of P_0 are shown in the figure as undarkened circles (○) and the vertices of P_1 are shown as darkened circles (●).

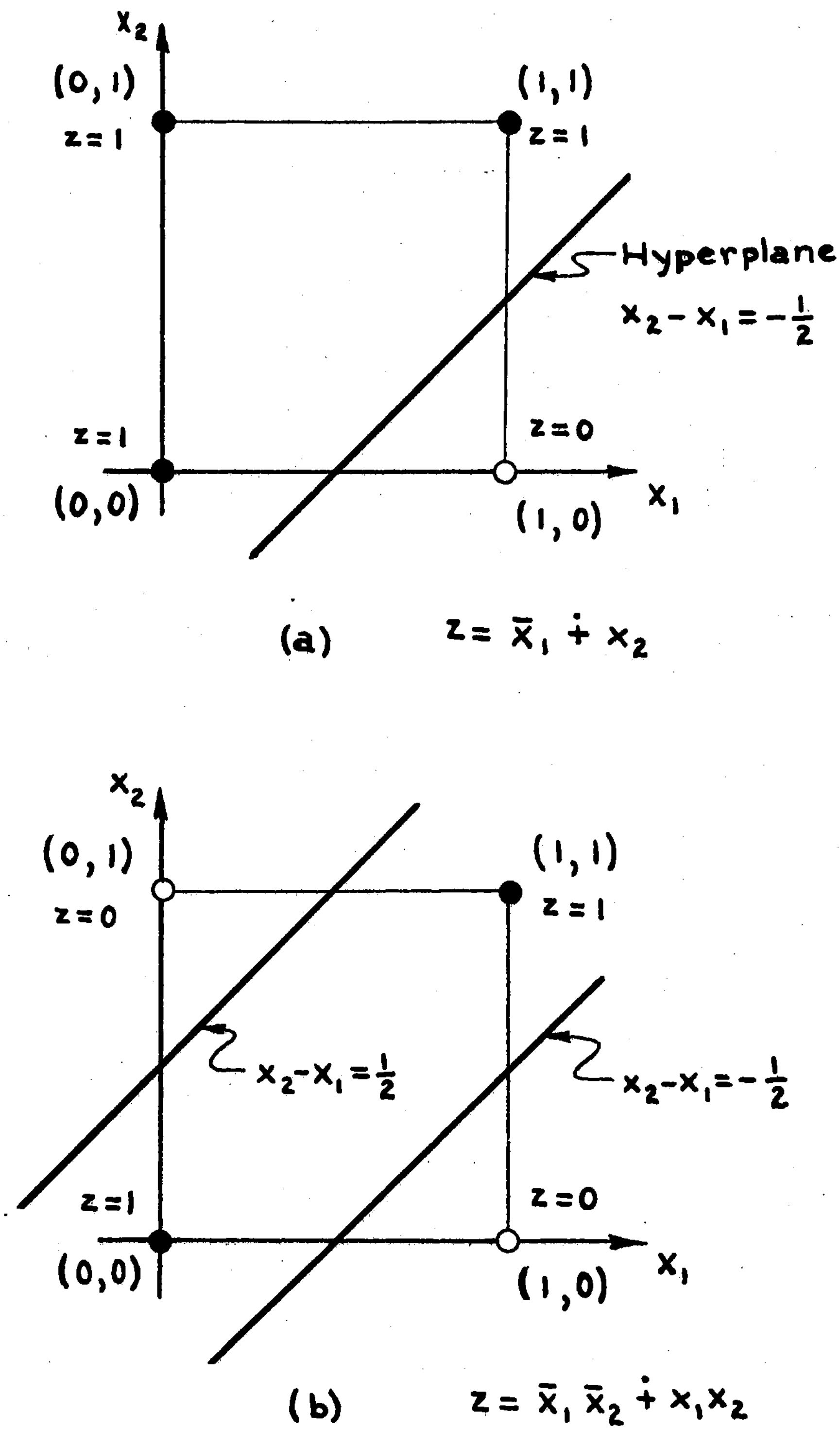


Figure 21. Geometric Interpretation of Threshold Elements

The equation of a hyperplane in the n-space is of the form:

$$\sum_{j=1}^{n} w_j x_j = T \tag{3.4}$$

and if this hyperplane separates P_0 from P_1 (as in Figure 21(a)), then Equation (3.2) truly is a realization of the Boolean function. Thus in Figure 21(a)

$$z = \langle x_2 - x_1 \rangle - \frac{1}{2} = \overline{x}_1 \dot{+} \overline{x}_2 \tag{3.5}$$

In Figure 21(b) the function

$$z = \overline{x}_1 \overline{x}_2 \dot{+} x_1 x_2 \tag{3.6}$$

is represented. By observing the figure, it is evident that there is no way to separate P_0 from P_1 by using a single hyperplane. This fact is sufficient to prove that Equation (3.6) is not a threshold function (single-threshold threshold function). However, it represents a multi-threshold threshold function. This class of functions is treated by Haring [18]. The definition of a multithreshold threshold element as given by Haring is

$$z = 1, \text{ if } \sum_{j=1}^{n} w_j x_j \geq T_1$$

or

$$T_{2i} \geq \sum_{j=1}^{n} w_j x_j \geq T_{2i+1} \quad i=1, 2, 3, \ldots \tag{3.7}$$

where

$$T_m \in (T_1, T_2, \ldots T_k)$$

and

$$T_m > T_{m+1}$$

and

$$z = 0, \text{ otherwise},$$

where z, x_j, w_j, and n are as defined as before,

T_m = m^{th} threshold, and

k = total number of thresholds.

Since Equation (3.7) is a generalization of the single-threshold threshold element [18], Equation (3.1) is included by Equation (3.7). For the sake of abbreviation the ordered set of weights and thresholds $(w_1, \ldots, w_n; T_1, \ldots, T_k)$ that specifies a k-threshold element is denoted by the vector $(\overline{W}; \overline{T})$ and called the MultiThreshold Weight Threshold Vector (MTWTV) [19].

Again observing Figure 21(b), it is seen that P_0 is separated from P_1 using two hyperplanes. In this example the MTWTV is

$$\overline{W}; \overline{T} = (1, -1; \frac{1}{2}, -\frac{1}{2}) \tag{3.8}$$

3.2 Example of a Mechanical Linkage Which Generates a Function of Three Variables

This example is presented to provide an introduction to the type of function generator being synthesized.

3.2.1 Function to be Generated

A linkage is presented that generates the Boolean function

$$z = x_1 \dot{+} x_2 \cdot x_3 \tag{3.9}$$

where x_1, x_2 and x_3 are inputs and z is the output. One threshold realization of this function, in accordance with Equation (3.2) is

$$z = \langle 2x_1 + x_2 + x_3 \rangle_{1.5} \tag{3.10}$$

If

$$2x_1 + x_2 + x_3 = f(x), \tag{3.11}$$

then Table I shows the validity of Equations (3.9), (3.10) and (3.11).

To implement this function mechanically, a linkage made up of two components is used. The first component is called a "separator linkage" and is shown in Figure 22. The second is the "threshold linkage" and is shown in Figure 23. Together the two components form the threshold logic generator of Figure 24.

3.2.2 Separator Linkage

The separator linkage (Figure 22) is a three-degree-of-freedom linkage with the input cranks X_AA, X_BB and X_CC representing the Boolean input variables x_1, x_2 and x_3, respectively. Two distinct positions of the input cranks are used to represent the "on" and "off" values of the input variables x_1, x_2 and x_3. Each of the input cranks is shown in its "off" or "0" position in the figure. When an input crank is rotated clockwise to the position shown with a dashed line, then it is in the "on" or "1" position. The output of the separator linkage is the coupler point F.

The purpose of this separator linkage is to separate the positions of output point F, for each of the input combinations, into relative positions according to column IV of Table I. (The linkage of Figure 22 is similar to the whiffletree type linkage used for the position separation in a binary to decimal converter of a random-access-memory [31]. It is also similar to the linkage used by Haas and Crossley [32] for a four-bit binary adding mechanism.)

TABLE I

TRUTH TABLE OF EQUATIONS
(3-9), (3-10), (3-11) and (3-13)

I	II	III	IV	V	VI	VII	VIII	IX
x_1	x_2	x_3	$f(x)$	z	s_F	s_1	s_2	s_3
0	0	0	0	0	0	0	0	0
0	0	1	1	0	s	0	0	4s
0	1	0	1	0	s	0	4s	0
0	1	1	2	1	2s	0	4s	4s
1	0	0	2	1	2s	4s	0	0
1	0	1	3	1	3s	4s	0	4s
1	1	0	3	1	3s	4s	4s	0
1	1	1	4	1	4s	4s	4s	4s

In order to study the relative positions of the inputs and the output, the points A, B and C of the input cranks are considered to have the displacements s_1, s_2 and s_3, respectively. The values s_1, s_2 and s_3 are also binary in accordance with x_1, x_2 and x_3. When $x_1 = 0$, then $s_1 = 0$ and when $x_1 = 1$, then $s_1 = ks = 4s$. The coefficient k is determined from

$$k = \sum_{j=1}^{3} | w_j | \tag{3.12}$$

where the values of w_j are found from Equation (3.10). An arbitrary distance is taken for s. The values of s_2 and s_3 are also 0 and 4s corresponding to values of x_2 and x_3 of "0" and "1".

In this separator linkage of Figure 22, points E and F are midpoints of the vertical links GB and DA. If the lengths of links GB and DA are great compared to the input displacements 4s, then it is seen that

$$s_F \approx \left[\frac{DF}{DA}\right] [s_1] + \left[\frac{GE}{GB}\right] \left[\frac{FA}{DA}\right] [s_2] + \left[\frac{EB}{GB}\right] \left[\frac{FA}{DA}\right] [s_3] = \frac{1}{2} s_1$$
$$+ \frac{1}{4} s_2 + \frac{1}{4} s_3 \tag{3.13}$$

where s_F is the horizontal component of displacement of the point F. Using Equation (3.13) the values of s_F are calculated for all combinations of the input variables. Together with the values of s_1, s_2 and s_3, the values of s_F are entered in columns VI to IX in Table I. By comparing f(x) and s_F of columns IV and VI it is observed that the linkage does separate the positions of F as desired.

3.2.3 Threshold Linkages

A "threshold linkage" is now employed to obtain the desired output z. This linkage uses the point F (the output coupler point of the

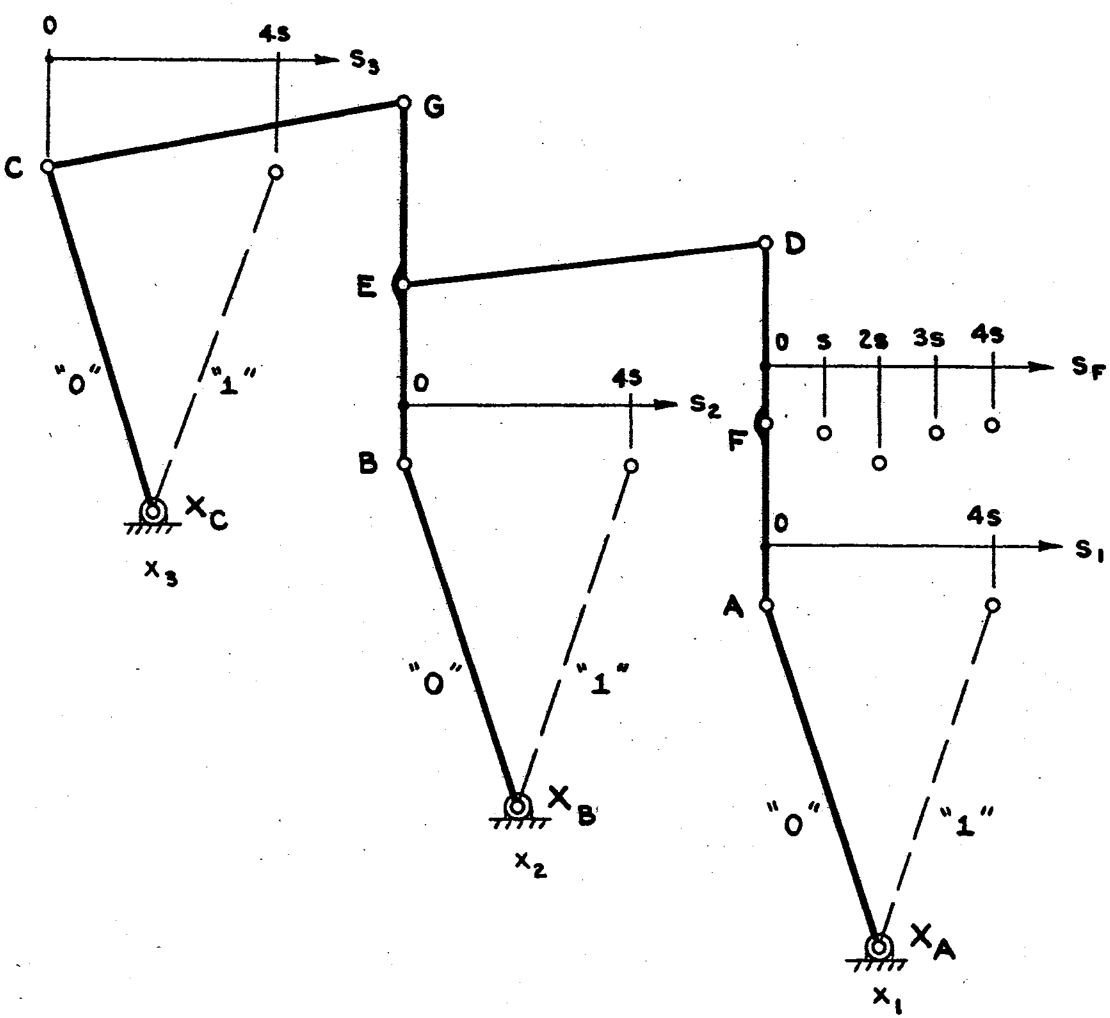


Figure 22. Separator Linkage for $s_F = \frac{1}{2}s_1 + \frac{1}{4}s_2 + \frac{1}{4}s_3$

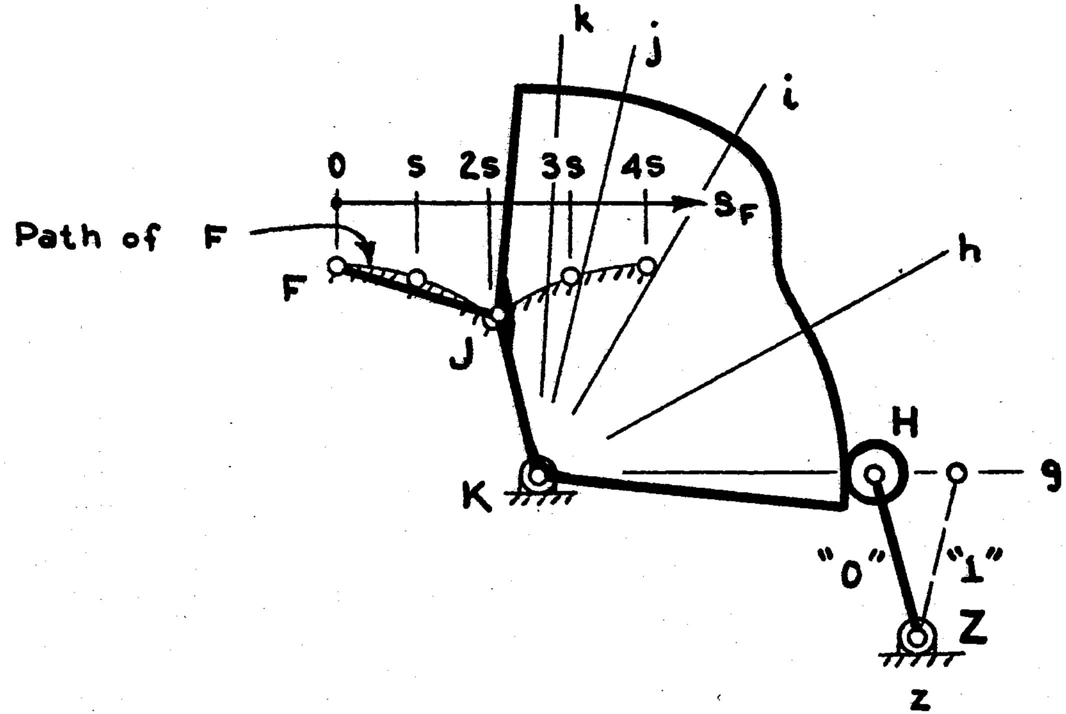


Figure 23. Threshold Linkage for $z = f(s_F)$

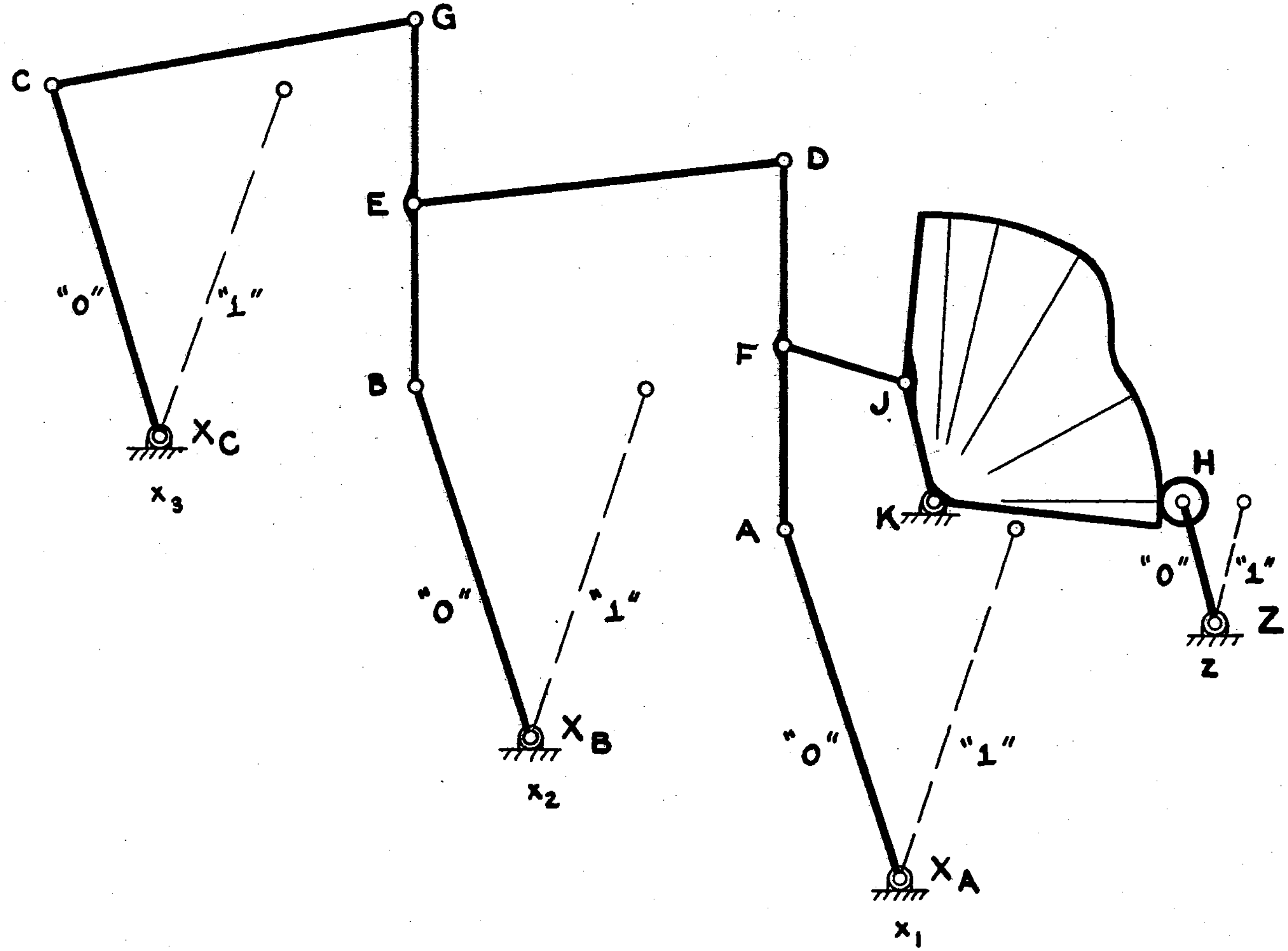


Figure 24. Threshold Logic Generator Composed of Separator Linkage and Threshold Linkage

separator linkage) as the input and a dwell-rise-dwell cam to provide the Boolean output z (see Figure 23). By observing z and s_F of columns V and VI in Table I, it is seen that

$$z = 0 \text{ when } s_F \leq s$$

and

$$z = 1 \text{ when } s_F \geq 2s \tag{3.14}$$

In Figure 23, when the input F is in the position of $s_F = 0$, the output roller follower at H is in contact with the cam along radial line g and the output z is in the "0" position. As F moves to the position where $s_F = s$, the cam rotates counterclockwise so that the follower H is in contact with the cam along line h. The cam has a dwell between g and h, and consequently the output z remains in the "0" position.

As F continues to move to the position $s_F = 2s$, the follower makes contact with the cam along line i. Since there is a rise on the cam between h and i, the output crank HZ rotates clockwise and z changes to the value "1". As F moves so that $s_F = 3s$ and 4s, the follower makes contact with the cam along lines j and k. Between i and k the follower dwells and thus z retains the value "1".

With the separator linkage and the threshold linkage connected together as shown in Figure 24, the output z is the desired output as given by Equation (3.9).

At this point the significance of using a cam linkage for the threshold linkage should be considered. As was just seen, the dwells on the cam provide an easy way of maintaining the output link precisely in the "0" or "1" positions during all input state changes that do not require an output state change. A linkage using only revolute pairs could probably be designed to provide satisfactory dwells; however, it is expected that the numbers of links in the linkage would have to be increased.

3.3 Structural Synthesis of Separator and Threshold Linkages

In this section a search is made for alternate linkages which might be used for the separator and threshold linkages. Individual searches are made for alternate separator and threshold linkages since the entire synthesis of the function generators is based on the component method. In this study cam linkages only are considered for the threshold linkages.

By using performance criteria, basic linkages are selected for the separator and threshold linkages.

3.3.1 Alternate Chains for Separators

In the example of Figure 22, an arbitrary eight-link three-degree-of-freedom linkage is used. As stated earlier, however, similar linkages have been used for a similar purpose [31, 32]. There is a possibility that other linkages can serve as well or better than the one chosen in Figure 22. In order to examine other eight-link three-degree-of-freedom linkages for possible merit, their respective kinematic chains must be enumerated. Since there are relatively few chains to be investigated, the enumeration is done through the use of Franke's condensed notation method as described by Davies and Crossley [34].

The enumeration process is described in Appendix C and the resulting eight-link chains with three-degrees-of-freedom are shown in Figure 25(a) through (f). As is seen later in this chapter, a separator linkage with

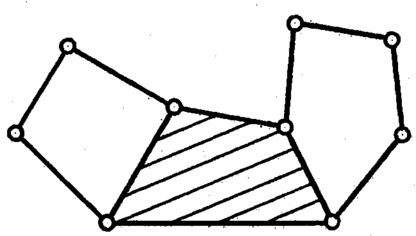
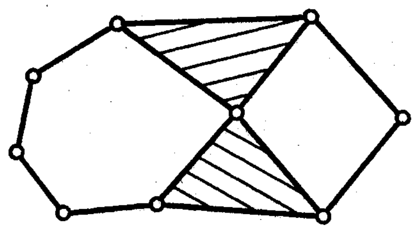
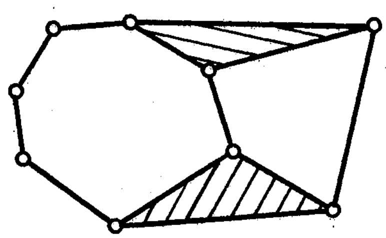
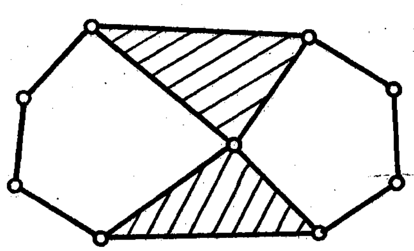
	KINEMATIC CHAINS	USABLE LINKAGES
(a)		None (Chain has fractionated degrees-of-freedom)
(b)		None (Chain has partial degrees-of-freedom)
(c)		None (Chain has partial degrees-of-freedom)
(d)		None (Chain has partial degrees-of-freedom)

Figure 25. Enumeration of Kinematic Chains and Linkages Usable as Separators

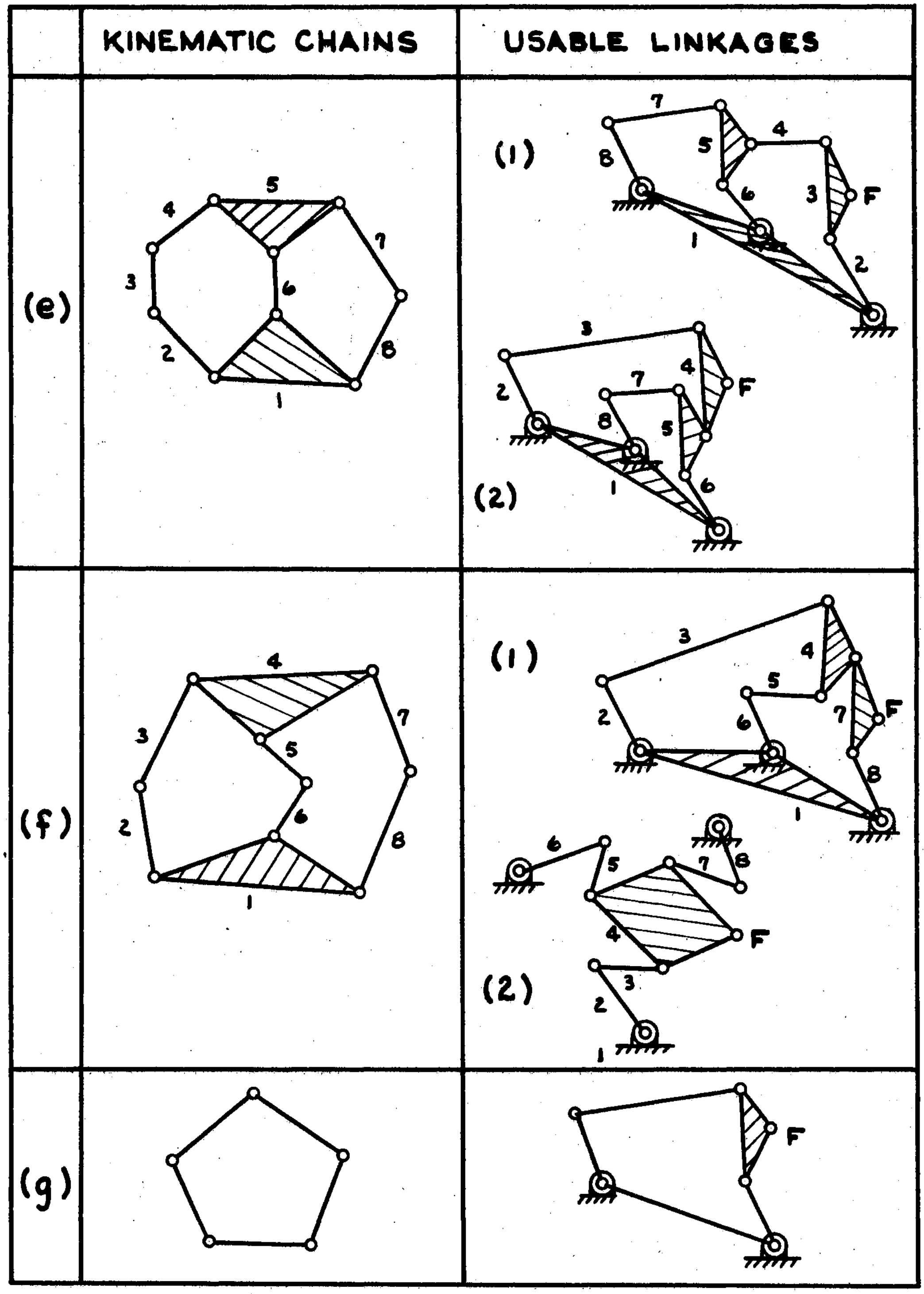


Figure 25. (Continued)

two inputs is also needed. The simplest, and satisfactory, two-degree-of-freedom chain is shown in Figure 25(g). The synthesis of this chain is also treated in Appendix C.

3.3.2 Performance Criteria and Selection of Separator Linkage

From the kinematic chains enumerated in the previous section, an eight-bar linkage is now selected to be used in the synthesis of all the separators. There is only one five-bar linkage and consequently no selection is needed in this category. Criteria for the selection of an eight-bar linkage is as follows:

1. Since there are three inputs to the separator linkages, the grounded link must be other than a binary link. As noted by Haas and Crossley [32], the inputs must be capable of independent activation and the output must respond to all of the inputs.

There are two types of chains appearing in the enumeration of Figure 25 which do not satisfy this requirement. They are chains with fractionated degrees of freedom or chains with partial degrees of freedom [34]. Each of the chains of Figure 25(a) through (d) has fractionated or partial degrees of freedom, and consequently no usable linkages can be derived from them.

There are four linkages derived from the chains of Figure 25(e) and (f) that do satisfy this requirement.

2. The linkages to be selected are of a type that can be synthesized dimensionally using a linear approximation. The linear approximation is obtained by arranging the links in a manner similar to that of Figure 22 and applying an expression such as that of Equation (3.13). (The dimensional synthesis is presented in sections 3.5 and 3.6.)

There are two arrangements of the chain of Figure 25(e) which lend themselves to this linear approximation. They are shown at (1) and (2) of the figure. The linkage at (1) is the arrangement used in the example of Figure 22.

For the chain of Figure 25(f), the only arrangement which lends itself to the use of the linear approximation is shown at (1). For the chain of Figure 25(f), (2), there seems to be no way of having the output point F on link 4 and arranging the linkage for the linear approximation.

After considering the two criteria above, there remain three linkages from which the selection is made. These are the linkages of Figure 25(e), (1) and (2), and (f), (1). The three linkages are very similar and there seems to be very little reason for the selection of one over the others. The only possible advantage to be noted is that the output link 3 of Figure 25(e), (1) is separated from the ternary link 5 with a binary link, whereas at (e), (2) the output link 4 is connected directly to the ternary link 5 and at (f), (1) the output link 7 is connected directly to the ternary link 4. The linkage at (e), (1) may thus be assembled somewhat more conveniently than the others.

The linkage as shown in Figure 25(e), (1) is selected to be synthesized dimensionally for the separators. The dimensional synthesis is explained in section 3.5.

3.3.3 Alternate Chain for Threshold Linkages

The threshold linkage of the example of Figure 23 can be derived from the Watt's chain shown in Figure 26(a). The only other six-link chain is the Stephenson's chain shown in Figure 26(b).

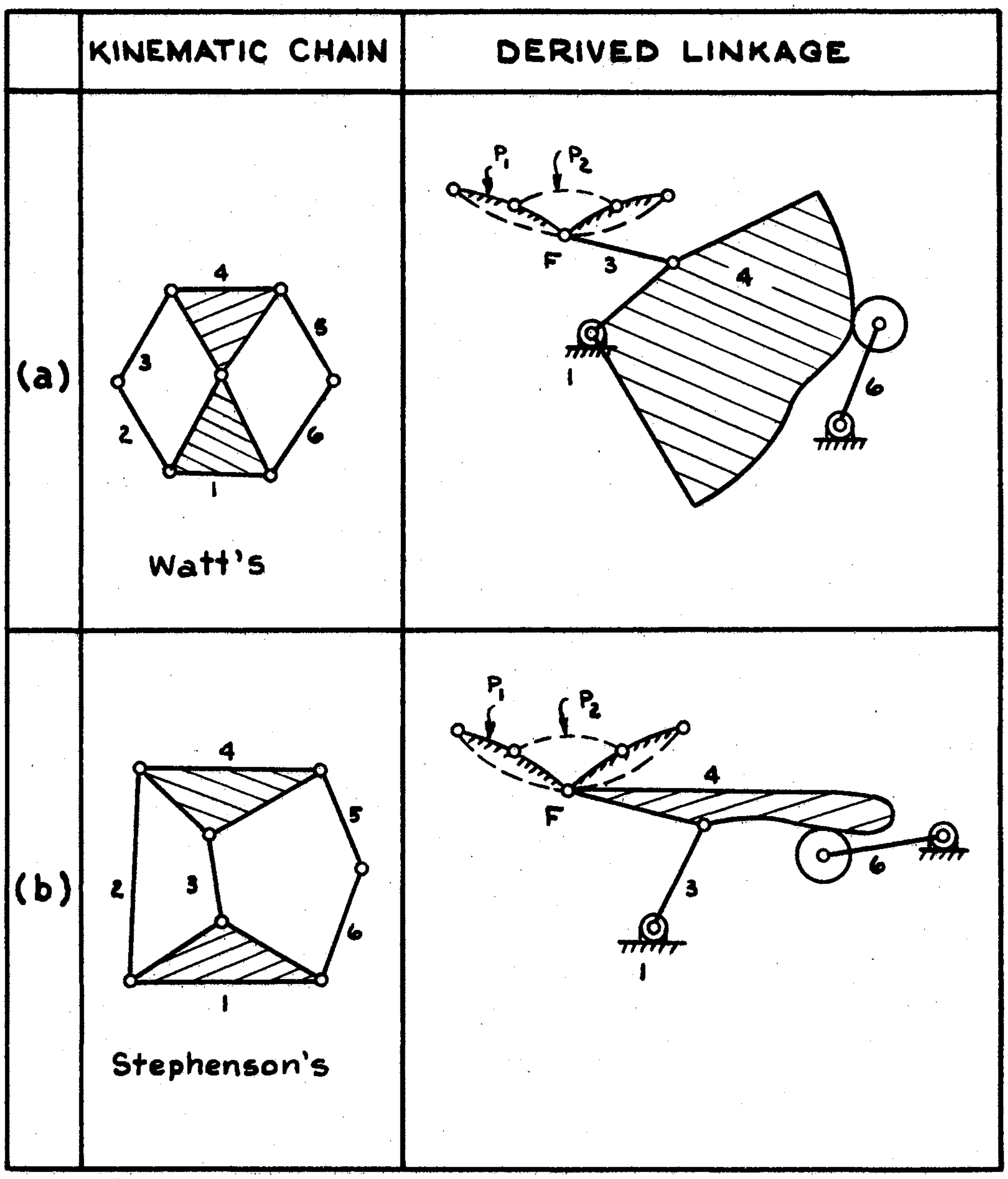


Figure 26. Six-Link Chains and Linkages Derived to Serve as Threshold Linkages

3.3.4 Performance Criteria and Selection of Threshold Linkage

Hain [35] lists a large number of cam linkages derived from the Watt's and Stephenson's kinematic chains shown in Figure 26(a) and (b). The only linkage, however, derived from the Watt's chain which allows for a path input, has an oscillating roller follower output and has only two cams is the one shown in Figure 26(a). The only linkage derived from the Stephenson's chain satisfying the same input and output conditions is the one shown in Figure 26(b). None of the other cam linkages listed by Hain appears to have qualities meriting consideration in this study.

The two cam linkages of Figure 26 are now examined according to performance criteria for the purpose of selecting the most favorable threshold linkage.

The performance criteria to be considered are as follows:

1. The input must be a path which is the path of point F of the separator linkage (Figure 22). It should be noted that F takes more than one path in moving from left to right depending on which of the input variables is being changed. The linkage of Figure 23 is repeated in Figure 27 to illustrate the multiple paths. In Figure 27 it can be seen that s_F changes from 2s to 4s by changing the input combination $x_1\ x_2\ x_3$ from 100 to 101 and then changing it to 111. Also, s_F changes from 2s to 4s by changing the input combination from 011 to 111, but the path of F is different in the two cases. The output z dwells (remains "1") during either of the two changes; hence, the linkage selected must be capable of being synthesized so that the output is insensitive to the different paths of F when the output dwells.

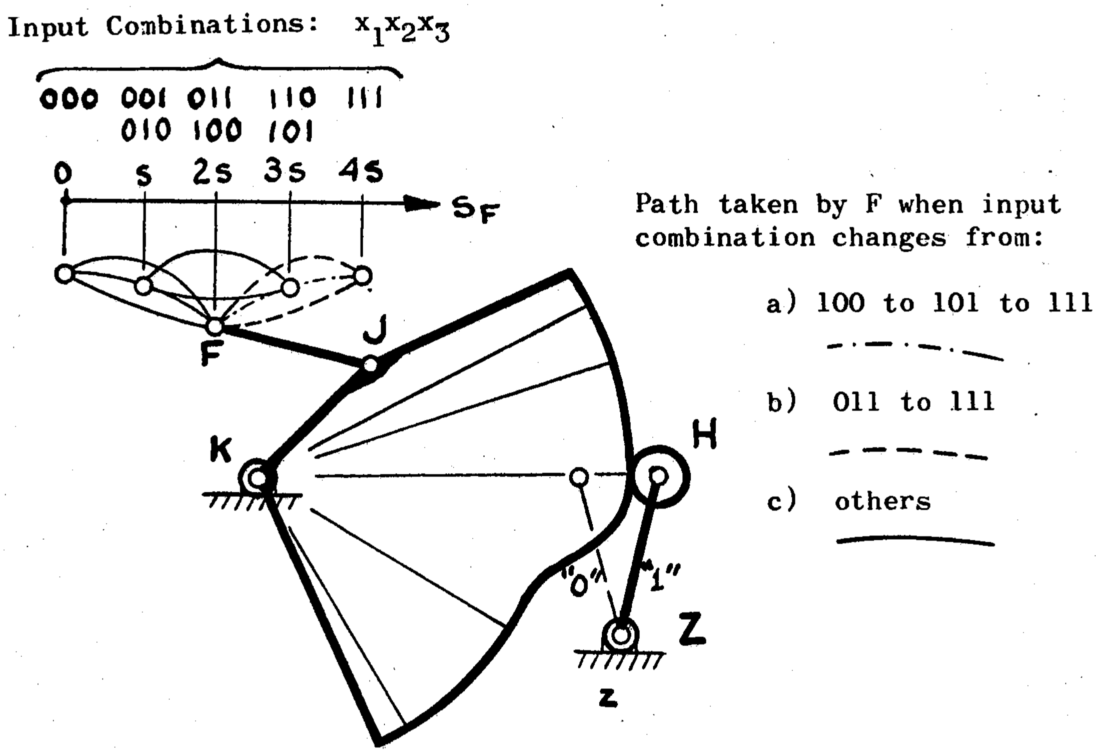


Figure 27. Threshold Linkage Showing Multiple Paths of Point F Between Stable States

In Figure 26, multiple paths of point F are shown for the linkages at both (a) and (b). Because the cam of the linkage at (a) has a fixed pivot (pivot K), the dwells of the output link 6 are not affected by the multiple paths of F. The cam of the linkage at (b), however, is part of the coupler link. In this case if the cam surface is designed so that link 6 dwells when F travels path p_1, then link 6 will not dwell when F travels path p_2.

2. Another desirable characteristic (which is similar to the above requirement) is that the output be relatively insensitive to errors in the position of point F caused by errors in the positions of the inputs at their "0" and "1" positions and errors caused by tolerances on link lengths. These errors will, in effect, create inaccuracies in the paths explained in criterion (1). Consequently, as in criterion (1), the linkage at (a) of Figure 26 is preferable.

3. A third selection criterion is that the dimensional synthesis of the linkage be as simple as possible. The cam surface of the linkage at (a) is composed of simple rises and circular arc dwells, while the cam at (b) would necessitate a more complex shape, even for a single path of the point F. This point is sufficient to make the linkage at (a) the favorable linkage.

The considerations of the performance criteria clearly indicate the selection of the linkage of Figure 26(a) for the threshold linkages.

3.4 Basic Separator and Threshold Linkages Needed

The number of functions of n variable is M, where

$$M = 2^{2^n}.$$

In this chapter of the study n is three and M becomes 256. Since the purpose of the study is to synthesize a mechanical linkage composed of a separator linkage and a threshold linkage of the types selected in the previous section, it is necessary to determine the 256 multithreshold weight threshold vectors (MTWTV). In the papers of Haring and Ohori [22] and Mow and Fu [19] a procedure and tables are given to find the MTWTV which describes each of the Boolean functions of four variables. The results of these two papers are utilized to determine the 256 MTWTV's. This section is devoted to a search for a limited number of basic separator and threshold linkages which can be combined in various ways to form the 256 function generators.

3.4.1 Recognition of Basic Separator Linkages

Mow and Fu [19] persented a table (Table I of their paper) which is used in this study to establish the basic separator linkages needed. The portion of Mow and Fu's table which applies to functions of three variables is repeated in Table II. It is evident in the papers of Haring and Ohori [22] and Mow and Fu [19] that all 256 functions of three variables lead to the thirteen $\overline{W}$; $\overline{T}$ vectors of Table II. Some functions, however, do not lead to a $\overline{W}$; $\overline{T}$ vector identically shown in Table II. Such functions lead to $\overline{W}$; $\overline{T}$ vectors that differ from the thirteen vectors of Table II only in one or more of the following:

1. A change in sign of one or more of the weights w_1, w_2, w_3.
2. A permutation of the weights with respect to the input variables.
3. A change in the value of the thresholds.

The significance of a change in the sign of a weight component such as w_3 can now be shown. If w_3 in the example of Figure 22 is negative

TABLE II*

THREE-VARIABLE EQUIVALENCE CLASSES AND THEIR $\overline{W};\overline{T}$ RELATIONS

Equivalence Class†	$\overline{W};\overline{T}$ ‡
One Threshold	
1	1,0,0,0; 0.5
4	2,1,1,0; 1.5
7	1,1,1,0; 1.5
9	1,1,0,0; 0.5
13	1,1,1,0; 0.5
Two Thresholds	
26	2,1,-1,0; 0.5,-0.5
27	-2,1,1,0; 1.5,-0.5
45	1,1,1,0; 1.5,0.5
46	2,2,-1,0; -,3.5,0.5
73	1,1,1,0; -,2.5,0.5
78	1,1,0,0; -,1.5,0.5
Three Thresholds	
93	3,1,-2,0; 1.5,-0.5,-1.5
163	-1,1,1,0; -,1.5,0.5,-0.5

*This table is taken from Table I presented by Mow and Fu [19].

†The equivalence class numbers are those given in the original table.

‡$\overline{W};\overline{T} = w_1, w_2, w_3; T_1, T_2, T_3$.

instead of positive,

$$f(x) = 2x_1 + x_2 = x_3 \tag{3.15}$$

and

$$s_F = \frac{1}{2} s_1 + \frac{1}{4} s_2 - \frac{1}{4} s_3 \tag{3.16}$$

The convention that the input cranks X_AA, X_BB and X_CC rotate in a clockwise direction when the input variables x_1, x_2 and x_3 change from "0" to "1" is maintained. If X_c is placed above C and called X_C' as shown in Figure 28, instead of below as in Figure 22, then Equation (3.16) is satisfied by the linkage of Figure 28. Similarly the sign of any coefficient of a weight vector can be changed by moving the center of rotation of the corresponding input crank.

From the argument just given, it follows that the basic separator linkages needed are obtained independently of the signs on the weight vectors $\overline{W}$.

It should be evident that a permutation of the weights in the $\overline{W};\overline{T}$ vector merely changes the input cranks of the separator linkage to which the three input variables x_1, x_2, x_3 are attached and that changes in the values of the thresholds do not affect the separator linkages.

From Table II it is now seen that basic separator linkages are needed for only six weight vectors. These vectors are:

(1, 0, 0)	(2, 1, 1)
(1, 1, 0)	(2, 2, 1)
(1, 1, 1)	(3, 2, 1) or (3, 1, 2).

The first of these vectors, (1, 0, 0), is really a trivial case since it is the vector of a single variable. The other five vectors are those

Truth Table

s_1	s_2	s_3	s_F
0	0	0	0
0	0	4s	-s
0	4s	0	s
0	4s	4s	0
4s	0	0	2s
4s	0	4s	s
4s	4s	0	3s
4s	4s	4s	2s

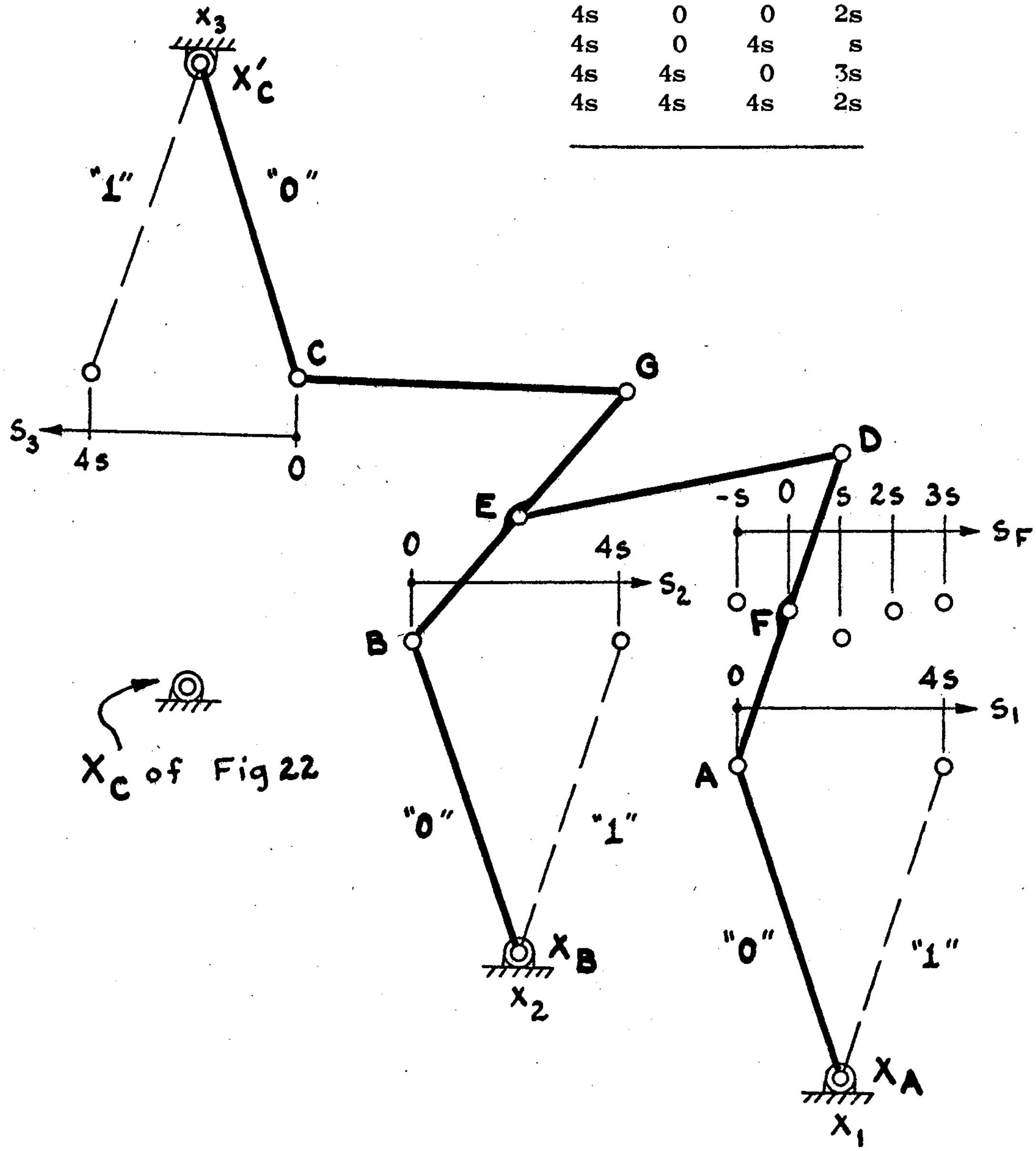


Figure 28. Separator Linkage for $s_F = \frac{1}{2}s_1 + \frac{1}{4}s_2 - \frac{1}{4}s_3$

which must be realized with basic separator linkages. The example of Figure 22 shows a linkage for the vector (1, 1, 2).

3.4.2 Recognition of Basic Threshold Linkages

The basic threshold linkages needed are known through the use of a complete tabulation of the 256 functions. The complete tabulation is shown in Table IV (of section 3.7.1). For the tabulation, the $\overline{W};\overline{T}$ vectors were determined using the procedure and tables given by Haring and Ohori [22] and Mow and Fu [19].

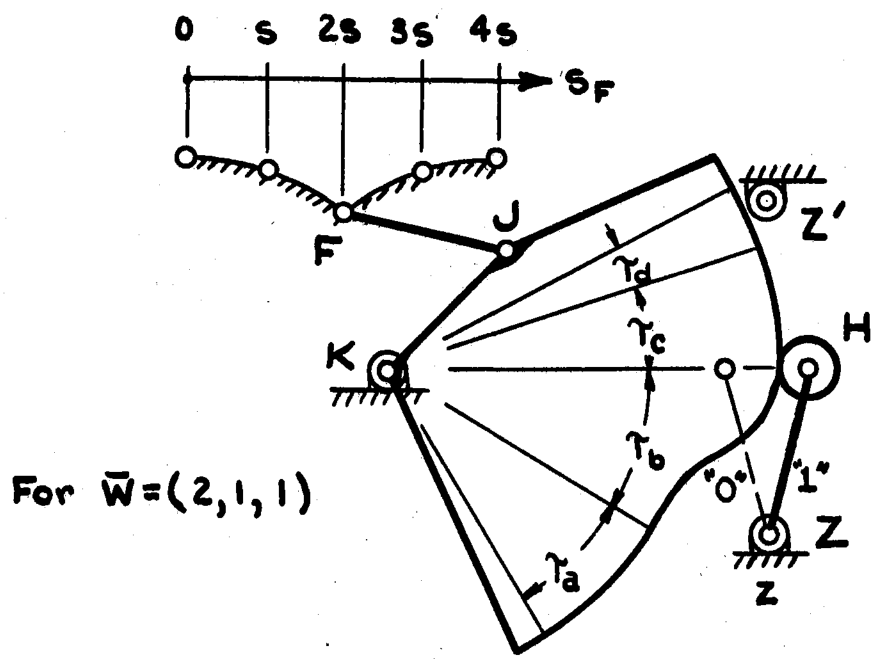
For the present analysis it is convenient to define a "threshold angle vector", $\overline{\tau}$, containing the elements $\tau_a, \tau_b, \ldots, \tau_f$. The elements $\tau_a, \tau_b, \ldots, \tau_f$ are called threshold angles.

The $\overline{\tau}$ vector column of Table IV is included for the purpose of simplifying the recognition of the threshold linkage needed for each function.

Figure 29 shows the threshold linkage of Figure 23 for the purpose of illustrating the threshold angles. The threshold linkage has four "separation angles" which are also τ_a, τ_b, τ_c and τ_d. It is noted that the number of separator angles is k, given by Equation (3.12).

The following shows the separation angles for the generators of all of the functions in terms of the $\overline{W}$ vectors.

$\overline{W}$	k	Separation Angles τ
$(\pm 1, \pm 1, 0)$	2	τ_a, τ_b
$(\pm 1, \pm 1, \pm 1)$	3	τ_a, τ_b, τ_c
$(\pm 2, \pm 1, \pm 1)$	4	$\tau_a, \ldots, \tau_d$
$(\pm 2, \pm 2, \pm 1)$	5	$\tau_a, \ldots, \tau_e$
$(\pm 3, \pm 2, \pm 1)$	6	$\tau_a, \ldots, \tau_f$



τ_b is the threshold angle

Figure 29. Threshold Linkage Showing Separation Angles τ_a, τ_b, τ_c and τ_d and Threshold Angle τ_b

The separation angle τ_a measures the rotation of the cam as s_F changes from s_{Fmin} to $s_{Fmin} + 1$, and the last separation angle for any generator measures the rotation of the cam as s_F changes from $s_{Fmax} - 1$ to s_{Fmax}. For instance, with the linkage of Figure 29 the angle τ_a measures the cam rotation as s_F changes from $s_{Fmin} = 0$ to $s_{Fmin} + 1 = s$. The last separation angle, τ_d, measures the cam rotation as $s_{Fmax} - 1 = 3s$ to $s_{Fmax} = 4s$.

Finally, a threshold angle is any one of the separation angles in which a threshold occurs. At a threshold, the output variable z changes states and there is a rise or return on the cam. In Figure 29 there is only one threshold angle and it is τ_b; the threshold angle vector is $\overline{\tau} = (\tau_b)$.

To further illustrate the meaning of the threshold angles of Table IV, section 3.7.1, the function where FV = 26 is observed. The $\overline{W};\overline{T}$ vector is

$$\overline{W};\overline{T} = (2, -1, -2; 0.5, -2.5) \tag{3.17}$$

This vector indicates that there is a threshold at $s_F = -2.5s$ and another at $s_F = 0.5s$. A graphical representation of the function is shown in Figure 30. The figure shows the threshold angles at τ_a and τ_d; consequently the threshold angle vector in Table IV for the function of FV = 26 is $\overline{\tau} = (\tau_a, \tau_d)$.

The superscripts on the threshold angles of Table IV indicate whether the output variable z changes from "0" to "1" or from "1" to "0" as the threshold is crossed. Using the function of FV = 26 again as an example, the superscript 10 on the threshold angle τ_a^{10} means that as s_F increases from -3s to -2s the output z changes from "1" to "0". This point is also illustrated in Figure 30, where the darkened circle ●

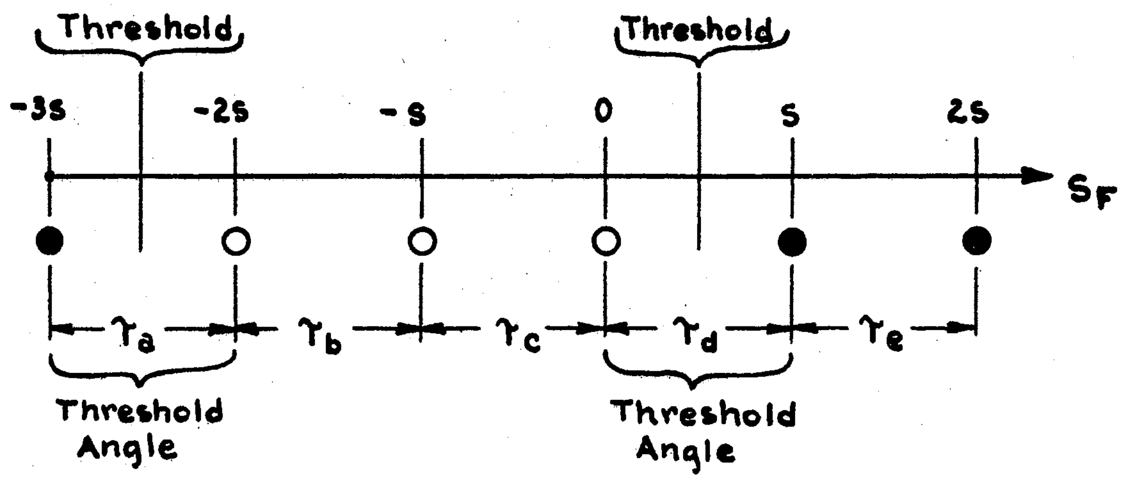


Figure 30. Graphical Representation of the Function of FV = 26 Showing Threshold Angles

at $s_F = -3s$ indicates that $z = 1$ and the undarkened circle ○ at $s_F = -2s$ indicates that $z = 0$. The superscript for τ_d^{01} of the function of FV = 26 is 01, which means that z changes from "0" to "1" as s_F changes from 0 to s.

It is noted that differences in superscripts do not cause "basically" different threshold linkages. If two functions have the same $\overline{W}$ vector and the same threshold angles except that the superscripts are are different, the two linkage function generators are the same except that the center of rotation of one output crank is Z and the center of the other is Z′ (see Figure 29).

Taking the above concepts into consideration, the basic threshold linkages can now be recognized. For each basic separator linkage, a basic threshold linkage is needed for each distinct threshold angle combination (where the superscripts on the threshold angles are disregarded). These combinations are located by scanning Table IV (of section 3.7.1). The combinations are shown in Table III, and the number of basic threshold linkages needed is nineteen.

3.5 Dimensional Synthesis of Separator Linkages

In section 3.3.2 a five-bar and an eight-bar linkage were selected for use as the separator. In section 3.4.1 the need was shown for proportioning these linkages in five ways to produce the basic separators for the weight vectors (1, 1, 0), (1, 1, 1), (2, 1, 1), (2, 2, 1) and (3, 2, 1).

The proportioning of the linkages is accomplished in this section. There are two fundamental steps to the synthesis procedure. The steps are:

TABLE III

BASIC THRESHOLD LINKAGES IN TERMS OF BASIC SEPARATORS AND THRESHOLD ANGLES

Vector of Basic Separator Linkage	Combinations of Threshold Angles			Number of Basic Threshold Linkages
	With One Threshold	With Two Thresholds	With Three Thresholds	
(1, 1, 0)	τ_a τ_b	τ_a, τ_b		3
(1, 1, 1)	τ_a τ_b τ_c	τ_a, τ_b τ_a, τ_c τ_b, τ_c	τ_a, τ_b, τ_c	7
(2, 1, 1)	τ_b τ_c	τ_a, τ_b τ_a, τ_c τ_b, τ_d τ_c, τ_d		6
(2, 2, 1)		τ_a, τ_d τ_b, τ_e		2
(3, 2, 1)			τ_a, τ_b, τ_c	1
			Total	19

1. The vertical links AD and BG (see Figure 22) are proportioned to provide the desired output positions of point F.

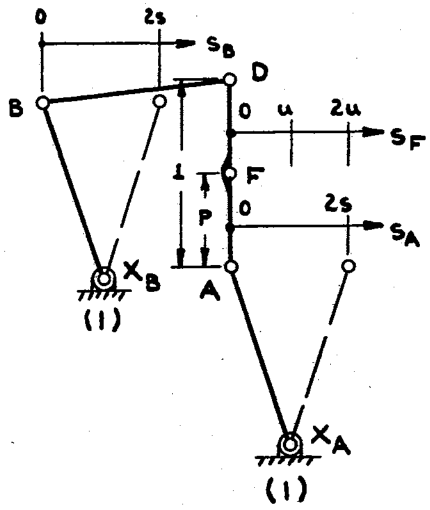
2. The link lengths are determined to provide a large displacement s_F, good link length ratios and good transmission angles.

3.5.1 Proportioning of Links AD and BG

The vertical links AD and BG of the five basic separator linkages are proportioned by using the assumption that the horizontal displacement of the output point F can be approximated by a linear expression. The linear expression used is essentially that of Equation (3.13).

Figure 31 shows the application of the linear expression in proportioning the links AD and BG for each of the basic separators. The $\overline{W}$ vector generated by the linkages is shown at the bottom of each illustration of Figure 31(a) through (e). In each figure, the linkage is shown with the weight that corresponds to the respective input indicated in parentheses. At the left center of the figures, the proportions of the links AD and BG are defined in terms of p_1 and p_2; the fact that AD and BG are both defined as 1 does not mean that they are necessarily the same length. The values of k as shown are found with Equation (3.12).

The horizontal components of translation of the inputs A, B and C are s_A, s_B and s_C, respectively. As the input variables change from the "0" state to the "1" state, the values of s_A, s_B and s_C change from 0 to ks. The horizontal component of output translation is s_F, where the minimum translation is 0 and the maximum ku. In the case at (b), for example, ks = 3s and ku = 3u. The unit of translation u is distinguished from s because with some separators (see Figure 31(d) and (e)) u is greater than s.

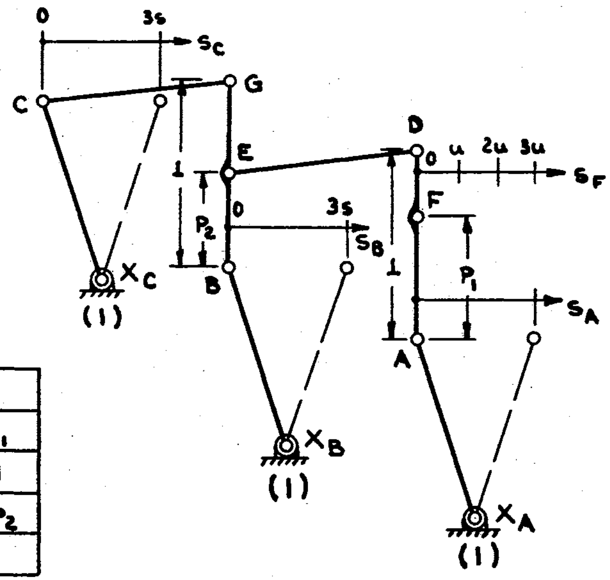


AD = 1
AF = p
k = 2

$$S_F \approx (1-p)S_A + (p)S_B$$

S_A	S_B		S_F	Approximation		Proportions
2s	0		u	$u \approx (1-p)(2s)$		$p = 1/2$
0	2s		u	$u \approx (p)(2s)$		
						$u/s = 1$

(a) $\bar{W} = (1, 1, 0)$



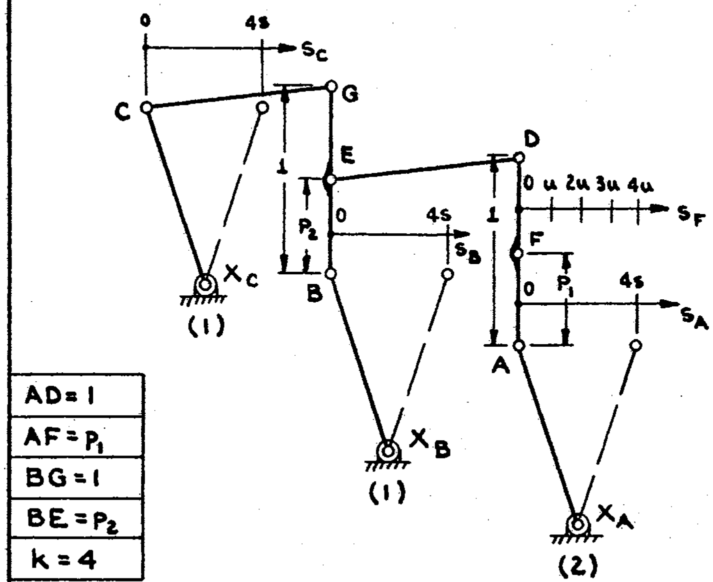
AD = 1
AF = p_1
BG = 1
BE = p_2
k = 3

$$S_F \approx (1-p_1)S_A + (1-p_2)(p_1)S_B + (p_2)(p_1)S_C$$

S_A	S_B	S_C	S_F	Approximation		Proportions
3s	0	0	u	$u \approx (1-p_1)(3s)$		$p_1 = 2/3$
0	3s	0	u	$u \approx (1-p_2)(p_1)(3s)$		$p_2 = 1/2$
0	0	3s	u	$u \approx (p_2)(p_1)(3s)$		$u/s = 1$

(b) $\bar{W} = (1, 1, 1)$

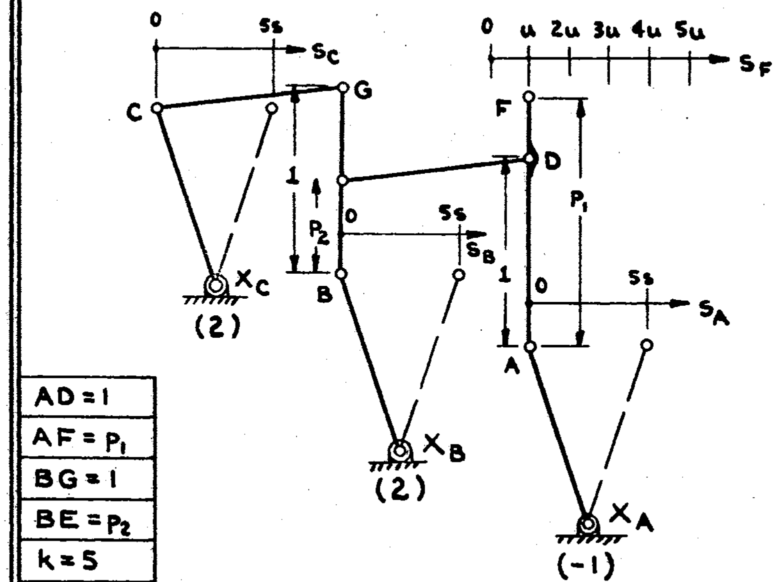
Figure 31. Proportioning of the Vertical Links AD and BG



$$S_F \approx (1-P_1)S_A + (1-P_2)(P_1)S_B + (P_2)(P_1)S_C$$

S_A	S_B	S_C	S_F	Approximation	Proportions
4s	0	0	2u	$2u \approx (1-P_1)(4s)$	$P_1 = 1/2$
0	4s	0	u	$u \approx (1-P_2)(P_1)(4s)$	$P_2 = 1/2$
0	0	4s	u	$u \approx (P_2)(P_1)(4s)$	$u/s = 1$

(c) $\overline{W} = (2, 1, 1)$

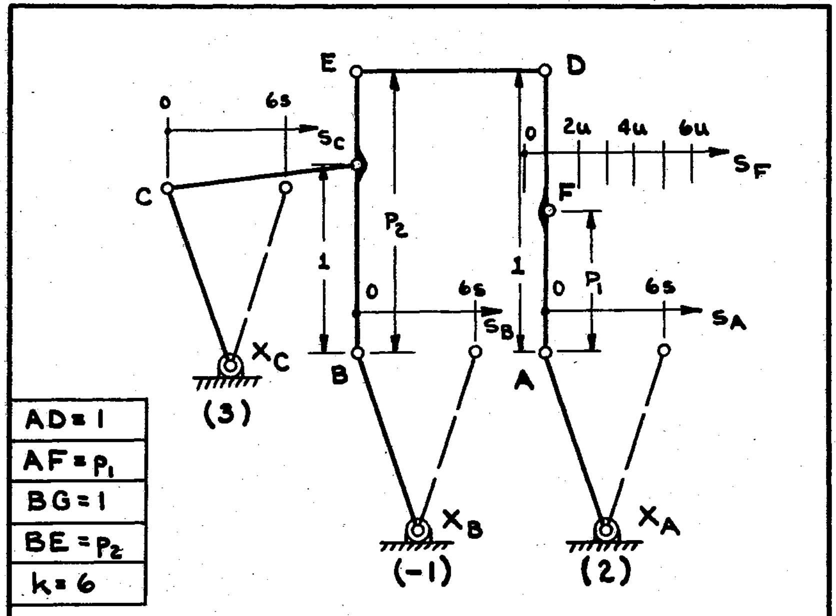


$$S_F \approx -(P_1-1)S_A + (1-P_2)(P_1)S_B + (P_2)(P_1)S_C$$

S_A	S_B	S_C	S_F	Approximation	Proportions
5s	0	0	-u	$-u \approx -(P_1-1)(5s)$	$P_1 = 4/3$
0	5s	0	2u	$2u \approx (1-P_2)(P_1)(5s)$	$P_2 = 1/2$
0	0	5s	2u	$2u \approx (P_2)(P_1)(5s)$	$u/s = 5/3$

(d) $\overline{W} = (2, 2, -1)$

Figure 31. (Continued)



$$s_F \approx (1-p_1)s_A - (p_2-1)(p_1)s_B + (p_2)(p_1)s_C$$

s_A	s_B	s_C	s_F	Approximation	Proportions
6s	0	0	2u	$2u \approx (1-p_1)(6s)$	$p_1 = 1/2$
0	6s	0	-u	$-u \approx -(p_2-1)(p_1)(6s)$	$p_2 = 3/2$
0	0	6s	3u	$3u \approx (p_2)(p_1)(6s)$	$u/s = 3/2$

(e) $\overline{W} = (3, 2, -1)$

Figure 31. (Continued)

In each figure, the linear approximation for s_F is written below the linkages in terms of p_1, p_2, s_A, s_B and s_C. At the bottom of each figure three combinations of s_A, s_B, s_C and s_F are written. Each combination is substituted in the linear approximation for s_F giving the three approximations shown. Finally, the three approximations are used as equalities and solved simultaneously to give the proportions p_1, p_2 and u/s. The values of p_1, p_2 and u/s are shown at the lower right.

It is noted that the $\overline{W}$ vector in Figure 31(d) contains one negative weight (-1). The negative weight occurs because the point F is on the extension of the link AD instead of between the ends A and D. By observing the figure it is seen that when A has a positive horizontal displacement, the point F has a negative horizontal displacement (the right is taken as positive).

The purpose of placing F on the extension of AD is to make the maximum horizontal displacement s_F greater than ks. Thus the maximum value of s_F is made larger without increasing s_A, s_B and s_C. This increase is expressed by the value of $u/s = 5/3$. The larger value of s_F in turn makes the threshold angles of the threshold linkage larger, without increasing the overall size of the separator linkage.

In Figure 31(e) there is also an input with a negative weight (-1). It occurs because the point E is on the extension of the link BG instead of between the ends of BG. Again, the purpose is to increase the maximum s_F without increasing s_A, s_B and s_C. The increase in this case is expressed by $u/s = 3/2$.

With the separators of Figure 31(a) and (b), the value of u/s cannot be increased in the same way as with the linkages of (e) and (f). The value of u/s could be increased with the separator of (c), but it is noted that with the arrangement shown the proportion p_2 is the same as at (b) and (d). Thus the linkages at (b), (c) and (d) differ only in the proportion p_1, which means that they differ only in the position of point F on the link AD. This last fact can in effect be used to say that the number of basic linkages needed for the five basic separators is only three: those of Figure 31(a), (b) and (e).

3.5.2 Determination of Link Lengths

The link lengths are now determined for the linkages of Figure 31(a) through (e). The dimensions are shown in Table V of section 3.7.2 and are given with reference to Figure 33 of section 3.7.2. It is noted that the only difference between the separators at (b), (c) and (d) is in the length of AF.

The synthesis of the dimensions is based on the following:

1. In order to maintain good link length ratios, the links X_AA, X_BB, X_CC, CG, BG, BD and ED are made of the same length ℓ for all the linkages. (In Table V the length ℓ is represented as 1.0.) The link AD of Figure 31(a), (b), (c) and (d) is also made of length ℓ. In Figure 31(e) link AD is made equal to BE which is 1.5ℓ.

2. In the positions of the linkages shown in Figure 31(a) through (e), the links AD and BG are made parallel (vertical on the page). The "0" and "1" positions of the input cranks X_AA, X_BB and X_CC are symmetrical about lines parallel to links AD and BG (vertical lines).

3. The displacements s_A, s_B and s_C of the points A, B and C (see Figure 31), as the input variables change from "0" to "1", are made of the same value s_I for all separators. The value of s_I is made as large a percent of ℓ as possible. As the ratio s_I/ℓ is increased, the minimum

transmission angles decrease. Thus the maximum value of s_I is dictated by selecting a minimum allowable transmission angle μ_m. In this synthesis μ_m is selected as 45°.

The determination of s_I is detailed in Appendix D. An analysis of the transmission angles for all the positions of the separator linkages is also given in Appendix D.

It should be noted that the approximations expressed in Figure 31 become poorer as the ratio of s_I/ℓ is increased. It was found, however, that the inaccuracies caused in s_F by using the value of s_I calculated with $\mu_m = 45^\circ$ were not great enough to create any problem. These slight inaccuracies are absorbed in the synthesis of the threshold linkages as shown in the details of Appendix E (section E.1).

4. Based on the selections and calculations of steps (1) through (3), the coordinates of the fixed centers X_A, X_A', X_B, X_B', X_C and X_C' (see Figure 33 of section 3.7.2) are calculated. The calculations are detailed in Appendix D.

An analysis of the output positions of link AD is combined with the synthesis in Appendix D. This analysis of the separator linkages is needed in the next section for the synthesis of the threshold linkages.

3.6 Dimensional Synthesis of Threshold Linkages

In section 3.3.4 the cam linkage of Figure 26(a) was selected for use as the threshold linkages. In section 3.4.2 the need was established for proportioning this linkage in nineteen ways to produce the basic threshold linkages of Table III.

There are two fundamental steps to the dimensional synthesis procedure for each threshold linkage. They are:

1. The loop FJK is synthesized (see Figure 29) to provide large threshold angles, small dwell angles, good link length ratios and a good transmission angle.

2. The cam and output follower are synthesized to provide good link length ratios and a good cam pressure angle.

The steps 1 and 2 are largely graphical procedures. Greater details are given in Appendix E. Since the loop FJK is synthesized graphically, an output analysis of the cam rotation angle is needed. The analysis allows the threshold angles to be located on the cam with accuracy. This output analysis is part of the computer Program B of Appendix E.

The dimensions which result from the synthesis and analysis are recorded in Table VI of section 3.7.3. The dimensions are given with reference to Figure 34 of section 3.7.3. The coordinates of the fixed centers K, Z and Z' are given, as well as the link lengths. The angles $\delta_1, \delta_2, \ldots, \delta_7$ are the result of the cam rotation analysis and are defined as shown in Figure 34. The δ angles are used, as shown in Table VI, to determine the magnitudes of the threshold angles.

In Figure 34 it is seen that only nine linkages, at (a) through (i), are used to represent the nineteen basic threshold linkages of Table III. Observation of Figure 34 and Table III shows why this is possible. For example, the linkage of Figure 34(a) with dimensions of Table VI, column I, has the two thresholds τ_a and τ_b. This same linkage serves for the other two basic threshold linkages for $\overline{W} = (1, 1, 0)$ shown in Table III. When the linkage is needed with the threshold τ_a only, the space for τ_b is filled with a dwell. When the linkage is needed with the threshold τ_b only, the space for τ_a is filled with a dwell.

Using the reasoning just given, it is seen that the linkage of Figure 34(b) serves for all seven of the basic threshold linkages for $\overline{W} = (1, 1, 1)$. The six basic threshold linkages for $\overline{W} = (2, 1, 1)$ are satisfied with the four linkages of Figure 34(c) through (f). (The linkage with a threshold at τ_b only is satisfied with the linkage of either Figure 34(c) or (e), and the linkage with τ_c only is satisfied with the linkage of either Figure 34(d) or (f).) The two basic threshold linkages for $\overline{W} = (2, 2, 1)$ are satisfied with the two linkages of Figure 34(g) and (h), and the single basic linkage for $\overline{W} = (3, 2, 1)$ is at Figure 34(i).

3.7 Design Charts for Selecting Threshold Logic Generators

In sections 3.5 and 3.6 the methods of dimensional synthesis were shown for all the separator and threshold linkages. In this section the dimensions are given in tabular form (Tables V and VI) and are given with reference to Figures 33 and 34. These figures are drawn to scale so that they represent the actual appearance of the linkages.

In conjunction with the dimensional tables and figures, a step-by-step procedure is presented for selecting the linkage needed to generate any of the 256 functions of three variables. The step-by-step procedure, tables and figures are contained in three design charts. The overall use of the design charts is shown in the flow chart of Figure 32.

Two examples of the use of the design charts are given in section 3.8.

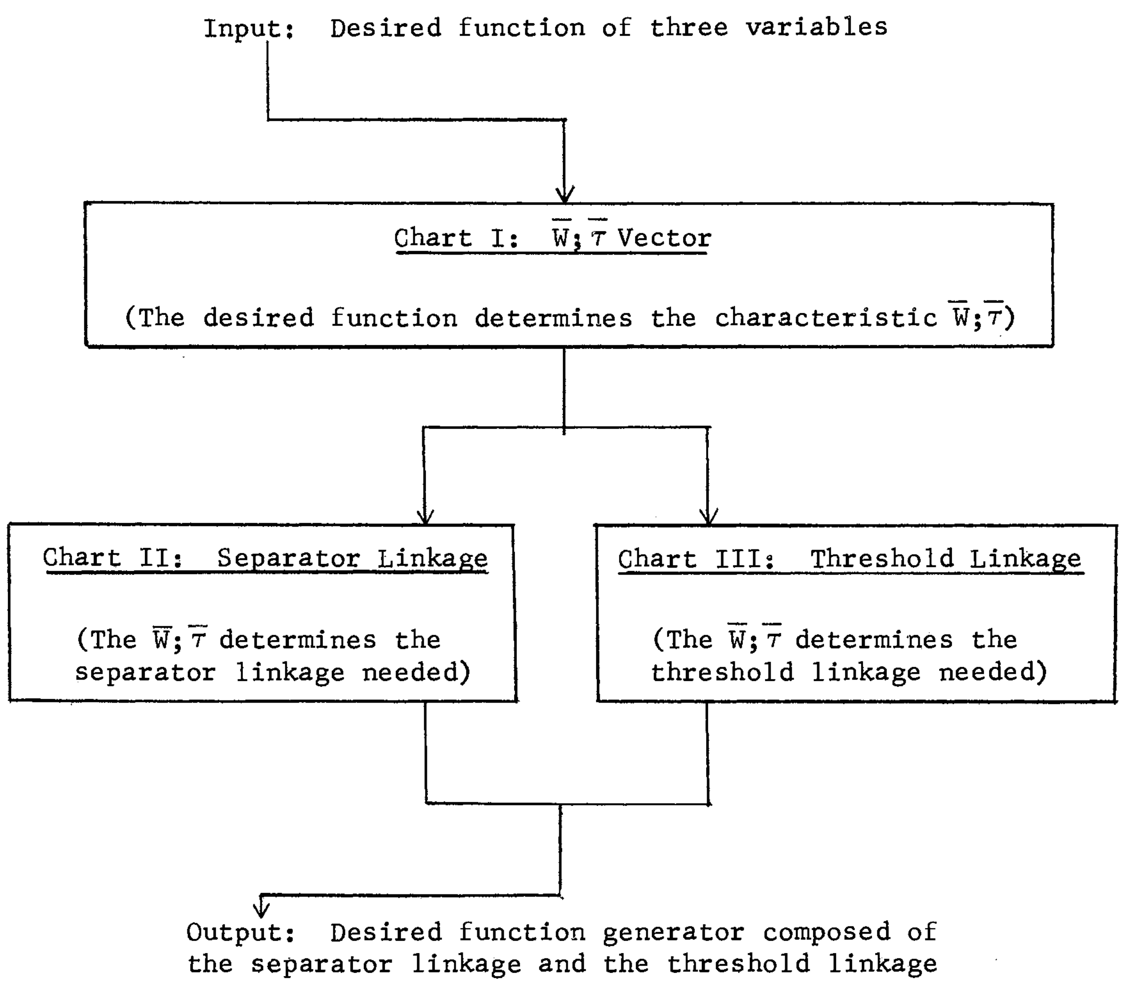


Figure 32. Flow Chart Showing Use of Design Charts in Determining Desired Function Generator

3.7.1 Chart I: Determination of $\overline{W};\overline{\tau}$ Vector

Steps:

1. Expand the given Boolean function to the sum of products canonical form (see examples, section 3.8). (Each term of the equation that has less than three variables is multiplied by $(\overline{x}_i \dotplus x_i)$ where i is a subscript missing in the term. This process is repeated until every term contains three variables; the equation is then in canonical form.)

2. Find the term values TV_i, $i = 0, \ldots, 7$ according to the following listing. If the term exists in the expansion of step 1, the corresponding TV_i has a value of 1; if the term does not exist in the expansion the corresponding TV_i is 0.

Term	Term Value
$\overline{x}_1 \cdot \overline{x}_2 \cdot \overline{x}_3$	TV_0
$\overline{x}_1 \cdot \overline{x}_2 \cdot x_3$	TV_1
$\overline{x}_1 \cdot x_2 \cdot \overline{x}_3$	TV_2
$\overline{x}_1 \cdot x_2 \cdot x_3$	TV_3
$x_1 \cdot \overline{x}_2 \cdot \overline{x}_3$	TV_4
$x_1 \cdot \overline{x}_2 \cdot x_3$	TV_5
$x_1 \cdot x_2 \cdot \overline{x}_3$	TV_6
$x_1 \cdot x_2 \cdot x_3$	TV_7

3. Find the function value FV by substituting the term values TV_i in the equation

$$FV = \sum_{i=0}^{7} \frac{128}{2^i} (TV_i) \equiv 128\ TV_0 + 64\ TV_1 + \ldots + TV_7$$

4. Find $\overline{W};\overline{\tau}$ from Table IV:

(a) If FV ≥ 127 locate $\overline{W};\overline{\tau}$ directly.

(b) If FV > 127 find FV_c = 255 - FV, then locate $\overline{W};\overline{\tau}$ using FV_c.

3.7.2 Chart II: Selection of Separator Linkage

Steps:

1. Select the basic separator linkage from Figure 33 using $\overline{W}$ in the form $(|w_1|, |w_2|, |w_3|)$.

2. Select the fixed centers X_A or X_A', X_B or X_B' and X_C or X_C' using $\overline{W}$ and assign input variables to the input links.

3. Determine the linkage dimensions from Table V.

3.7.3 Chart III: Selection of Threshold Linkage

Steps:

1. Select the basic threshold linkage from Figure 34 using $\overline{W};\overline{\tau}$ in the form $(|w_1|, |w_2|, |w_3|; \tau_a, \ldots, \tau_f)$.

2. Select the fixed center Z or Z′;

(a) If 0 ≤ FV ≤ 127 and the superscript on first threshold of $\overline{\tau}$ is $\begin{cases} 01, \text{ use } Z. \\ 10, \text{ use } Z'. \end{cases}$

(b) If FV > 127 and superscript on first threshold of $\overline{\tau}$ is $\begin{cases} 01, \text{ use } Z'. \\ 10, \text{ use } Z. \end{cases}$

3. Determine linkage dimensions from Table VI.

4. Place rises and returns on the cam according to $\overline{\tau}$. The first τ that appears indicates a rise. If a second τ exists it represents a return and if a third τ exists it represents a rise. The cycloidal motion program is used for rises and returns.

5. Complete the cam with dwells.

TABLE IV

WEIGHT-THRESHOLD ANGLE VECTORS
VS. FUNCTION VALUES

FV or FV_c	Weight Vector W_1, W_2, W_3	Threshold Vector T_1, T_2, T_3	Threshold Angle Vector $\tau_a, \tau_b, \ldots, \tau_f$
0	0, 0, 0		
1	1, 1, 1	2.5	τ_c^{01}
2	1, 1, -1	1.5	τ_c^{01}
3	1, 1, 0	1.5	τ_b^{01}
4	1, -1, 1	1.5	τ_c^{01}
5	1, 0, 1	1.5	τ_b^{01}
6	2, -1, -1	1.5, 0.5	τ_c^{01}, τ_d^{10}
7	2, 1, 1	2.5	τ_c^{01}
8	1, -1, -1	0.5	τ_c^{01}
9	2, 1, -1	2.5, 1.5	τ_c^{01}, τ_d^{10}
10	1, 0, -1	0.5	τ_b^{01}
11	2, 1, -1	1.5	τ_c^{01}
12	1, -1, 0	0.5	τ_b^{01}

TABLE IV (Continued)

FV or FV_c	Weight Vector W_1, W_2, W_3	Threshold Vector T_1, T_2, T_3	Threshold Angle Vector $\tau_a, \tau_b, \ldots, \tau_f$
13	2, -1, 1	1.5	τ_c^{01}
14	2, -1, -1	0.5	τ_c^{01}
15	1, 0, 0	0.5	τ_a^{01}
16	-1, 1, 1	1.5	τ_c^{01}
17	0, 1, 1	1.5	τ_b^{01}
18	-1, 2, -1	1.5, 0.5	τ_c^{01}, τ_d^{10}
19	1, 2, 1	2.5	τ_c^{01}
20	-1, -1, 2	1.5, 0.5	τ_c^{01}, τ_d^{10}
21	1, 1, 2	2.5	τ_c^{01}
22	1, 1, 1	2.5, 1.5	τ_b^{01}, τ_c^{10}
23	1, 1, 1	1.5	τ_b^{01}
24	-1, 1, 1	1.5, -0.5	τ_a^{10}, τ_c^{01}
25	-1, 2, 2	2.5, -0.5	τ_a^{10}, τ_d^{01}
26	2, -1, -2	0.5, -2.5	τ_a^{10}, τ_d^{01}
27	3, 1, -2	1.5, -0.5, -1.5	$\tau_a^{01}, \tau_b^{10}, \tau_d^{01}$

TABLE IV (Continued)

FV or FV_c	Weight Vector W_1, W_2, W_3	Threshold Vector T_1, T_2, T_3	Threshold Angle Vector $\tau_a, \tau_b, \ldots, \tau_f$
28	2, -2, -1	0.5, -2.5	τ_a^{10}, τ_d^{01}
29	3, -2, 1	1.5, -0.5, -1.5	$\tau_a^{01}, \tau_b^{10}, \tau_d^{01}$
30	-2, 1, 1	1.5, -0.5	τ_b^{10}, τ_d^{01}
31	2, 1, 1	1.5	τ_b^{01}
32	-1, 1, -1	0.5	τ_c^{01}
33	-1, 2, 1	2.5, 1.5	τ_c^{01}, τ_d^{10}
34	0, 1, -1	0.5	τ_b^{01}
35	1, 2, -1	1.5	τ_c^{01}
36	1, -1, 1	1.5, -0.5	τ_a^{10}, τ_c^{01}
37	2, -1, 2	2.5, -0.5	τ_a^{10}, τ_d^{01}
38	-1, 2, -2	0.5, -2.5	τ_a^{10}, τ_d^{01}
39	1, 3, -2	1.5, -0.5, -1.5	$\tau_a^{01}, \tau_b^{10}, \tau_d^{01}$
40	1, 1, -2	1.5, 0.5	τ_c^{01}, τ_d^{10}
41	1, 1, -1	1.5, 0.5	τ_b^{01}, τ_c^{10}
42	1, 1, -2	0.5	τ_c^{01}

TABLE IV (Continued)

FV or FV_c	Weight Vector W_1, W_2, W_3	Threshold Vector T_1, T_2, T_3	Threshold Angle Vector $\tau_a, \tau_b, \ldots, \tau_f$
43	1, 1, -1	0.5	τ_b^{01}
44	2, -2, 1	-, 1.5, -1.5	τ_a^{10}, τ_d^{01}
45	-2, -1, 1	-0.5, -2.5	τ_a^{01}, τ_c^{10}
46	3, -2, -1	0.5, -1.5, -2.5	$\tau_a^{01}, \tau_b^{10}, \tau_d^{01}$
47	2, 1, -1	0.5	τ_b^{01}
48	-1, 1, 0	0.5	τ_b^{01}
49	-1, 2, 1	1.5	τ_c^{01}
50	-1, 2, -1	0.5	τ_c^{01}
51	0, 1, 0	0.5	τ_a^{01}
52	-2, 2, -1	0.5, -2.5	τ_a^{10}, τ_d^{01}
53	-2, 3, 1	1.5, -0.5, -1.5	$\tau_a^{01}, \tau_b^{10}, \tau_d^{01}$
54	1, -2, 1	1.5, -0.5	τ_b^{10}, τ_d^{01}
55	1, 2, 1	1.5	τ_b^{01}
56	-2, 2, 1	-, 1.5, -1.5	τ_a^{10}, τ_d^{01}
57	-1, -2, 1	-0.5, -2.5	τ_a^{01}, τ_c^{10}

TABLE IV (Continued)

FV or FV_c	Weight Vector W_1, W_2, W_3	Threshold Vector T_1, T_2, T_3	Threshold Angle Vector $\tau_a, \tau_b, \ldots, \tau_f$
58	-2, 3, -1	0.5, -1.5, -2.5	$\tau_a^{01}, \tau_b^{10}, \tau_d^{01}$
59	1, 2, -1	0.5	τ_b^{01}
60	1, 1, 0	-, 1.5, 0.5	τ_a^{01}, τ_b^{10}
61	2, 2, -1	-, 3.5, 0.5	τ_b^{01}, τ_e^{10}
62	2, 2, 1	3.5, 1.5	τ_a^{01}, τ_d^{10}
63	1, 1, 0	0.5	τ_a^{01}
64	-1, -1, 1	0.5	τ_c^{01}
65	1, -1, 2	2.5, 1.5	τ_c^{01}, τ_d^{10}
66	1, 1, -1	1.5, -0.5	τ_a^{10}, τ_c^{01}
67	2, 2, -1	2.5, -0.5	τ_a^{10}, τ_d^{01}
68	0, -1, 1	0.5	τ_b^{01}
69	1, -1, 2	1.5	τ_c^{01}
70	-1, -2, 2	0.5, -2.5	τ_a^{10}, τ_d^{01}
71	1, -2, 3	1.5, -0.5, -1.5	$\tau_a^{01}, \tau_b^{10}, \tau_d^{01}$
72	1, -2, 1	1.5, 0.5	τ_c^{01}, τ_d^{10}

TABLE IV (Continued)

FV or FV_c	Weight Vector W_1, W_2, W_3	Threshold Vector T_1, T_2, T_3	Threshold Angle Vector $\tau_a, \tau_b, \ldots, \tau_f$
73	1, -1, 1	1.5, 0.5	τ_b^{01}, τ_c^{10}
74	2, 1, -2	-, 1.5, -1.5	τ_a^{10}, τ_d^{01}
75	-2, 1, -1	-0.5, -2.5	τ_a^{01}, τ_b^{10}
76	1, -2, 1	0.5	τ_c^{01}
77	1, -1, 1	0.5	τ_b^{01}
78	3, -1, -2	0.5, -1.5, -2.5	$\tau_a^{01}, \tau_b^{10}, \tau_d^{01}$
79	2, -1, 1	0.5	τ_b^{01}
80	-1, 0, 1	0.5	τ_b^{01}
81	-1, 1, 2	1.5	τ_c^{01}
82	-2, -1, 2	0.5, -2.5	τ_a^{10}, τ_d^{01}
83	-2, 1, 3	1.5, -0.5, -1.5	$\tau_a^{01}, \tau_b^{10}, \tau_d^{01}$
84	-1, -1, 2	0.5	τ_c^{01}
85	0, 0, 1	0.5	τ_a^{01}
86	1, 1, -2	1.5, -0.5	τ_b^{10}, τ_d^{01}
87	1, 1, 2	1.5	τ_b^{01}

TABLE IV (Continued)

FV or FV_c	Weight Vector W_1, W_2, W_3	Threshold Vector T_1, T_2, T_3	Threshold Angle Vector $\tau_a, \tau_b, \dots, \tau_f$
88	-2, 1, 2	-, 1.5, -1.5	τ_a^{10}, τ_d^{01}
89	-1, 1, -2	-0.5, -2.5	τ_a^{01}, τ_c^{10}
90	1, 0, 1	-, 1.5, 0.5	τ_a^{01}, τ_b^{10}
91	2, -1, 2	-, 3.5, 0.5	τ_b^{01}, τ_e^{10}
92	-2, -1, 3	0.5, -1.5, -2.5	$\tau_a^{01}, \tau_b^{10}, \tau_d^{01}$
93	1, -1, 2	0.5	τ_b^{01}
94	2, 1, 2	3.5, 1.5	τ_a^{01}, τ_d^{10}
95	1, 0, 1	0.5	τ_a^{01}
96	-2, 1, 1	1.5, 0.5	τ_c^{01}, τ_d^{10}
97	-1, 1, 1	1.5, 0.5	τ_b^{01}, τ_c^{10}
98	1, 2, -2	-, 1.5, -1.5	τ_a^{10}, τ_d^{01}
99	1, -2, -1	-0.5, -2.5	τ_a^{01}, τ_c^{10}
100	1, -2, 2	-, 1.5, -1.5	τ_a^{10}, τ_d^{01}
101	1, -1, -2	-0.5, -2.5	τ_a^{01}, τ_c^{10}
102	0, 1, 1	-, 1.5, 0.5	τ_a^{01}, τ_b^{10}

TABLE IV (Continued)

FV or FV_c	Weight Vector W_1, W_2, W_3	Threshold Vector T_1, T_2, T_3	Threshold Angle Vector $\tau_a, \tau_b, \ldots, \tau_f$
103	-1, 2, 2	-, 3.5, 0.5	τ_b^{01}, τ_e^{10}
104	-1, -1, -1	-0.5, -1.5	τ_b^{01}, τ_c^{10}
105	-1, 1, 1	1.5, 0.5, -0.5	$\tau_a^{10}, \tau_b^{01}, \tau_c^{10}$
106	-1, -1, 2	1.5, -0.5	τ_b^{10}, τ_d^{01}
107	1, 1, -1	0.5, -0.5	τ_a^{10}, τ_b^{01}
108	-1, 2, -1	1.5, -0.5	τ_b^{10}, τ_d^{01}
109	1, -1, 1	0.5, -0.5	τ_a^{10}, τ_b^{01}
110	1, 2, 2	3.5, 0.5	τ_a^{01}, τ_d^{10}
111	2, 1, -1	0.5, -0.5	τ_a^{10}, τ_b^{01}
112	-2, 1, 1	0.5	τ_c^{01}
113	-1, 1, 1	0.5	τ_b^{01}
114	-1, 3, -2	0.5, -1.5, -2.5	$\tau_a^{01}, \tau_b^{10}, \tau_d^{01}$
115	-1, 2, 1	0.5	τ_b^{01}
116	-1, -2, 3	0.5, -1.5, -2.5	$\tau_a^{01}, \tau_b^{10}, \tau_d^{01}$
117	-1, 1, 2	0.5	τ_b^{01}

TABLE IV (Continued)

FV or FV_c	Weight Vector W_1, W_2, W_3	Threshold Vector T_1, T_2, T_3	Threshold Angle Vector $\tau_a, \tau_b, \ldots, \tau_f$
118	1, 2, 2	3.5, 1.5	τ_a^{01}, τ_d^{10}
119	0, 1, 1	0.5	τ_a^{01}
120	2, -1, -1	1.5, -0.5	τ_b^{10}, τ_d^{01}
121	-1, 1, 1	0.5, -0.5	τ_a^{10}, τ_b^{01}
122	2, 1, 2	3.5, 0.5	τ_a^{01}, τ_d^{10}
123	-1, 2, 1	0.5, -0.5	τ_a^{10}, τ_b^{01}
124	2, 2, 1	3.5, 0.5	τ_a^{01}, τ_d^{10}
125	1, -1, 2	0.5, -0.5	τ_a^{10}, τ_b^{01}
126	1, 1, 1	-, 2.5, 0.5	τ_a^{01}, τ_c^{10}
127	1, 1, 1	0.5	τ_a^{01}

TABLE V

SEPARATOR LINKAGE DIMENSIONS

	$\overline{W}$	I (1,1,0)	II (1,1,1)	III (2,1,1)	IV (2,2,1)	V (3,2,1)
Coordinates of Fixed Centers	x_{X_A}	0.00000	0.00000	0.00000	0.00000	0.00000
	y_{X_A}	0.00000	0.00000	0.00000	0.00000	0.00000
	x_{X_A}'	0.00000	0.00000	0.00000	0.00000	0.00000
	y_{X_A}'	1.89976	1.89976	1.89976	1.89976	1.89976
	x_{X_B}	-0.99396	-0.99396	-0.99396	-0.99396	-1.00000
	y_{X_B}	0.89023	0.39023	0.39023	0.39023	0.00000
	x_{X_B}'	-0.99396	-0.99396	-0.99396	-0.99396	-1.00000
	y_{X_B}'	2.78999	2.28999	2.28999	2.28999	1.89976
	x_{X_C}	--	-1.98791	-1.98791	-1.98791	-1.99396
	y_{X_C}	--	1.28045	1.28045	1.28045	0.89023
	x_{X_C}'	--	-1.98791	-1.98791	-1.98791	-1.99396
	y_{X_C}'	--	3.18022	3.18022	3.18022	2.78999
Link Lengths	X_AA	1.00000	1.00000	1.00000	1.00000	1.00000
	X_BB	1.00000	1.00000	1.00000	1.00000	1.00000
	X_CC	--	1.00000	1.00000	1.00000	1.00000
	AD	1.00000	1.00000	1.00000	1.00000	1.50000
	DB	--	--	--	--	--
	DE	1.00000	1.00000	1.00000	1.00000	1.00000
	BE	--	0.50000	0.50000	0.50000	1.50000
	BG	--	1.00000	1.00000	1.00000	1.00000
	CG	--	1.00000	1.00000	1.00000	1.00000
	AF	0.50000	0.66667	0.50000	1.33333	0.75000
Input Angles	$\Delta\theta$	36.43273	36.43273	36.43273	36.43273	36.43273
	$\theta_{"0"}$	108.21630	108.21630	108.21630	108.21630	108.21630
	$\theta_{"1"}$	71.78363	71.78363	71.78363	71.78363	71.78363

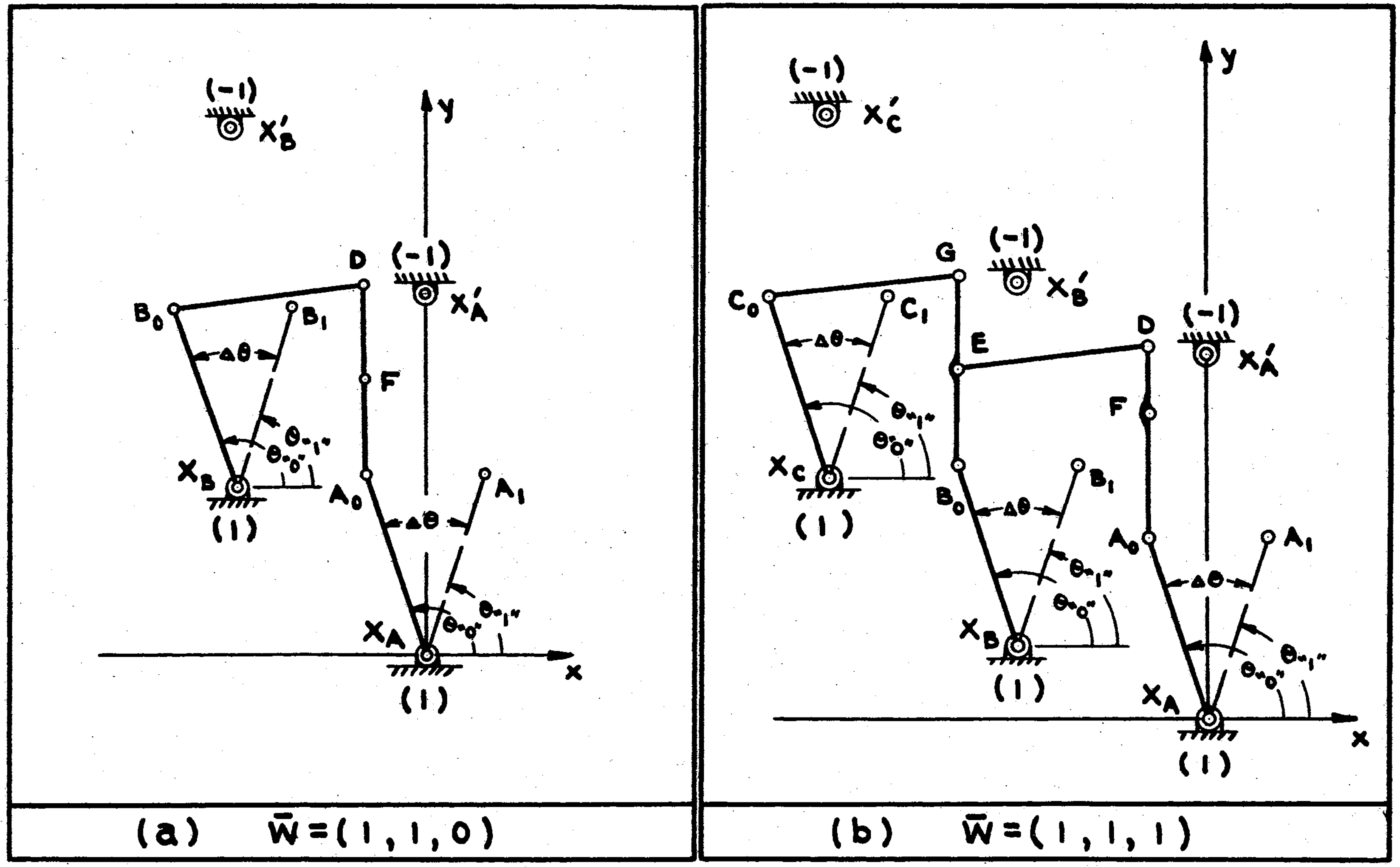
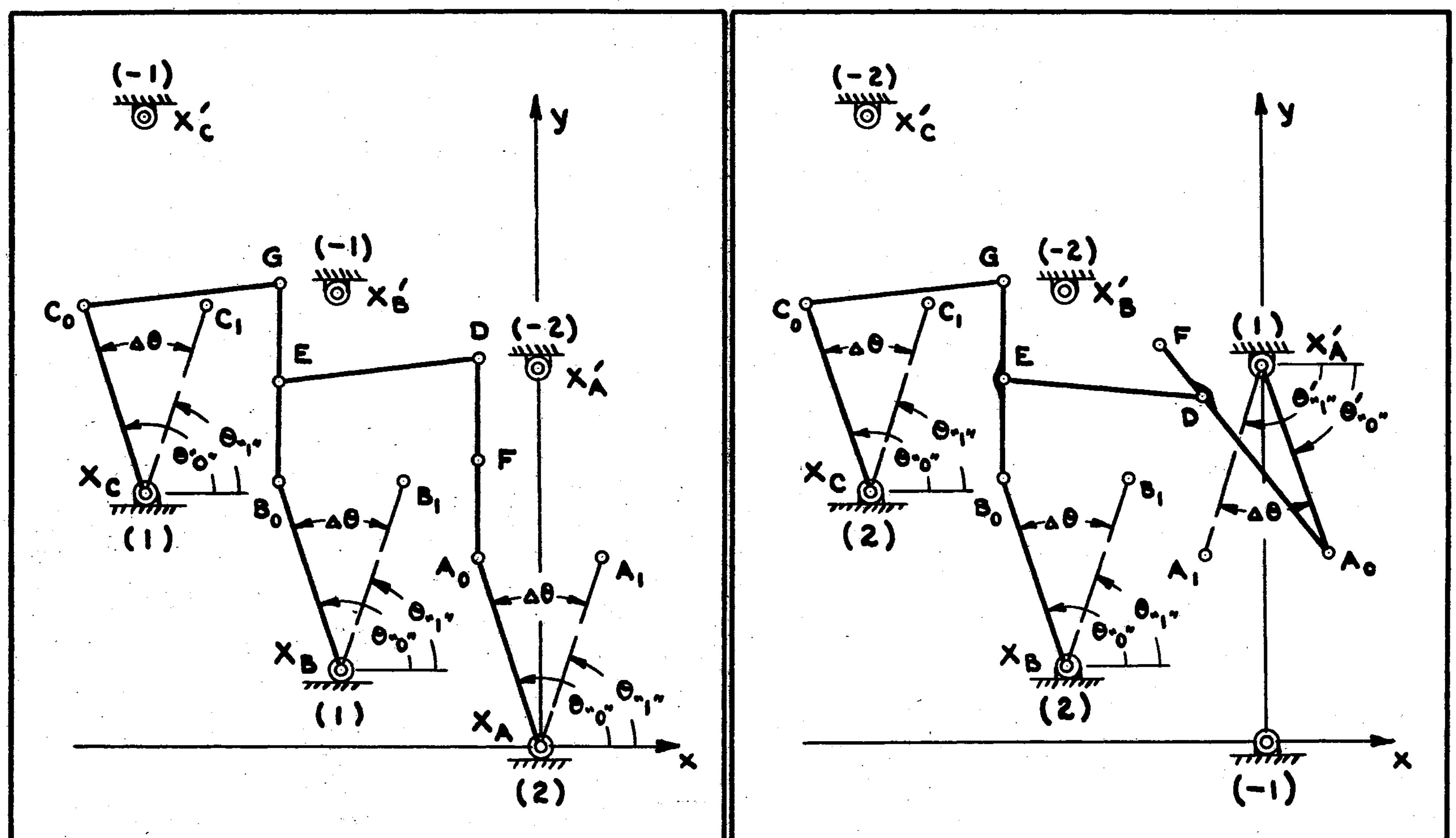


Figure 33. Separator Linkages



(c) $\bar{W} = (2, 1, 1)$ (d) $\bar{W} = (2, 2, 1)$

Figure 33. (Continued)

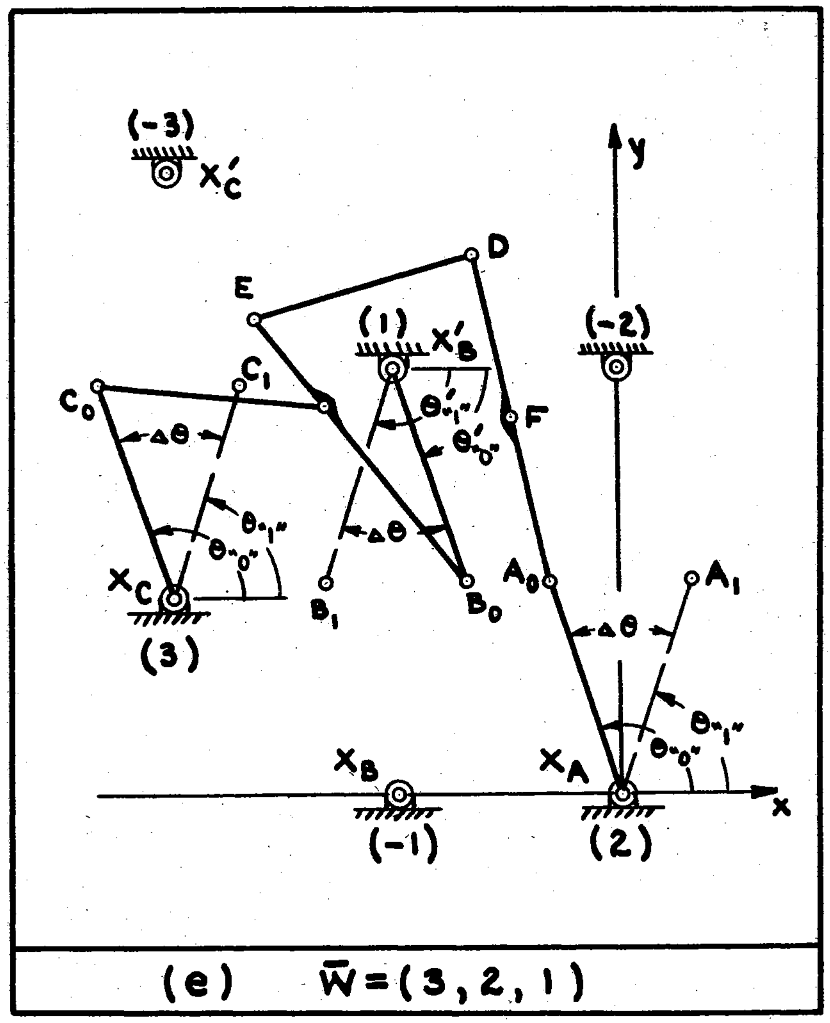


Figure 33. (Continued)

TABLE VI

THRESHOLD LINKAGE DIMENSION

		I (1, 1, 0; τ_a, τ_b)	II (1, 1, 1; τ_a, τ_b, τ_c)	III (2, 1, 1; τ_a, τ_b)	IV (2, 1, 1; τ_a, τ_c)	V (2, 1, 1; τ_b, τ_d)	VI (2, 1, 1; τ_c, τ_d)	VII (2, 2, 1; τ_a, τ_d)	VIII (2, 2, 1; τ_b, τ_e)	IX (3, 2, 1; τ_a, τ_b, τ_d)
Coordinates of Fixed Centers	x_K	0.28252	0.21289	0.22534	0.12961	-0.12960	-0.22534	0.64560	-0.63185	0.12603
	y_K	2.01024	1.92828	0.87133	1.85347	1.85347	0.87133	2.98600	2.92247	1.17928
	x_Z	1.16194	1.09230	1.33475	1.23902	0.97981	0.88407	1.75501	0.47756	1.23544
	y_Z	1.67778	1.59582	0.53887	1.52101	1.52101	0.53887	2.65354	2.59001	0.84682
	x_Z'	1.16194	1.09230	1.33475	1.23902	0.97981	0.88407	1.75501	0.47756	1.23544
	y_Z'	2.34270	2.26074	1.20379	2.18593	2.18593	1.20379	3.31846	3.25493	1.51174
Link Lengths	FJ	0.40000	0.33900	0.50200	0.50200	0.50200	0.50200	1.00000	1.00000	0.45300
	JK	0.57000	0.37700	0.38900	0.35800	0.35800	0.38900	0.64700	0.62500	0.39800
	KH	0.77000	0.77000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
	ZH	0.35000	0.35000	0.35000	0.35000	0.35000	0.35000	0.35000	0.35000	0.35000
Roller Radius	R_r (max)	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000
Cam Rotation Angles	$\theta_{11(0)}$	249.89700	240.28790	101.30510	279.20522	168.82400	156.51320	247.32250	199.16370	118.18610
	δ_1	30.43504	31.88648	23.82632	23.84822	18.21710	12.25926	26.23706	0.02318	27.81996
	δ_2	30.43542	31.88600	23.82681	23.34860	18.21750	12.25949	43.08242	24.37606	53.57556
	δ_3	64.58416	31.53956	50.81484	42.24343	49.72710	27.00482	43.08352	24.37705	74.17320
	δ_4	--	61.83807	50.81484	42.24343	49.72710	27.00488	62.08396	44.60435	74.34611
	δ_5	--	65.85151	64.85433	72.30055	66.86780	52.67398	62.08454	44.60517	106.53520
	δ_6	--	65.85198	64.85453	72.30088	66.86820	52.67436	85.99259	64.75100	101.85770
	δ_7	--	104.47160	77.81855	91.97027	91.97036	77.81852	86.42655	89.87910	109.23120
Threshold Angles	τ_a	δ_1 30.43504	δ_3 31.53956	δ_1 23.82632	δ_1 23.84822	--	--	δ_1 26.23706	--	δ_1 27.81996
	τ_b	$\delta_3-\delta_2$ 34.14874	$\delta_4-\delta_1$ 29.95159	$\delta_3-\delta_2$ 26.98803	--	$\delta_3-\delta_2$ 31.50960	--	--	$\delta_2-\delta_1$ 24.35288	$\delta_2-\delta_1$ 25.75560
	τ_c	--	$\delta_7-\delta_6$ 38.61962	--	$\delta_5-\delta_4$ 30.05712	--	$\delta_5-\delta_4$ 25.66910	--	--	--
	τ_d	--		--	--	$\delta_7-\delta_6$ 25.10216	$\delta_7-\delta_6$ 25.14416	$\delta_6-\delta_5$ 23.90805	--	δ_5+24 δ_5 24.00000
	τ_e	--	--	--	--	--	--	--	$\delta_7-\delta_6$ 25.12811	--
	τ_f	--	--	--	--	--	--	--	--	--
Output Angles	$\Delta\theta$	36.43273	36.43273	36.43273	36.43273	36.43273	36.43273	36.43273	36.43273	36.43273
	$\theta_{"0"}$	108.21630	108.21630	108.21630	108.21630	108.21630	108.21630	108.21630	108.21630	108.21630
	$\theta_{"1"}$	71.78363	71.78363	71.78363	71.78363	71.78363	71.78363	71.78363	71.78363	71.78363

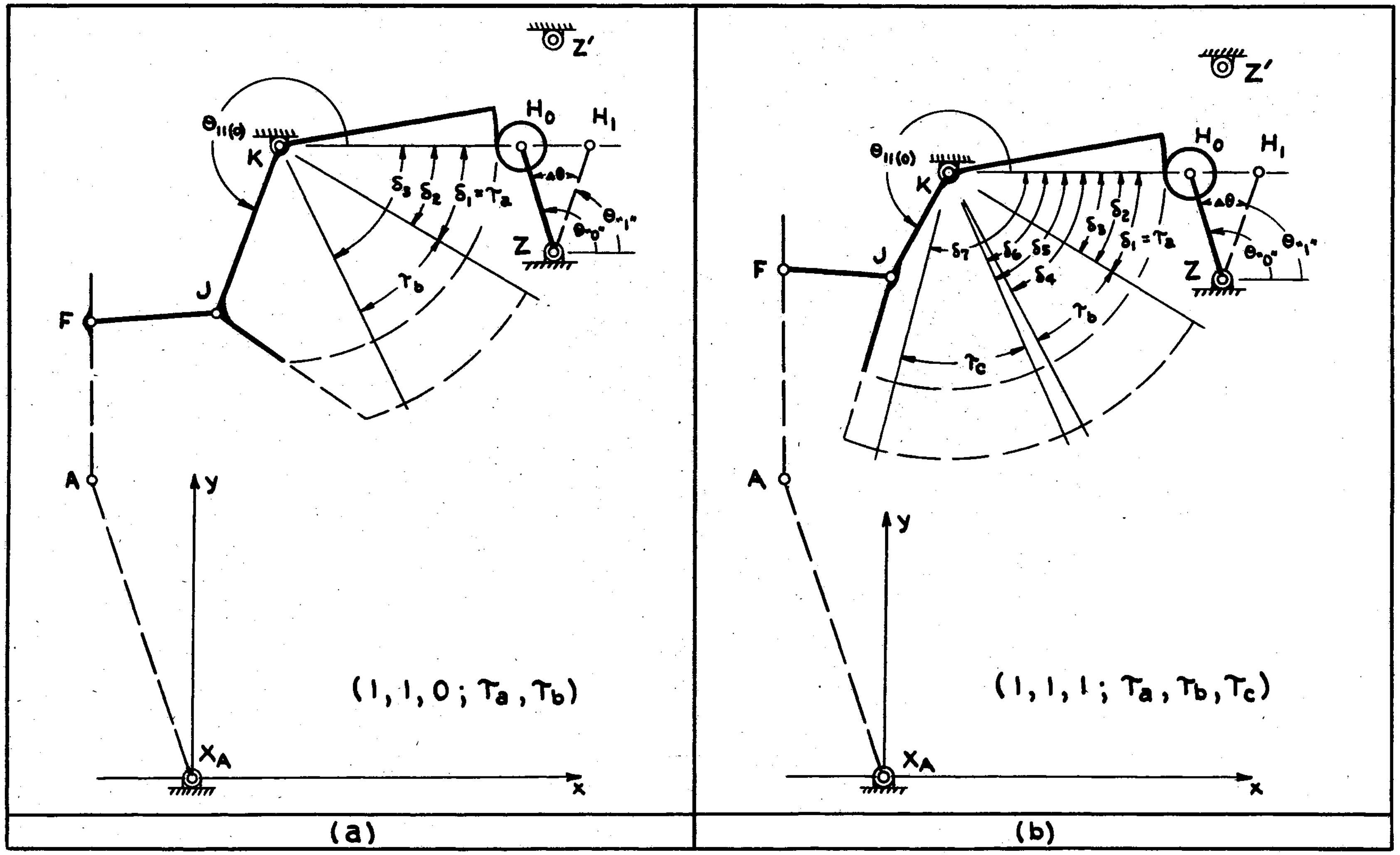


Figure 34. Threshold Linkages

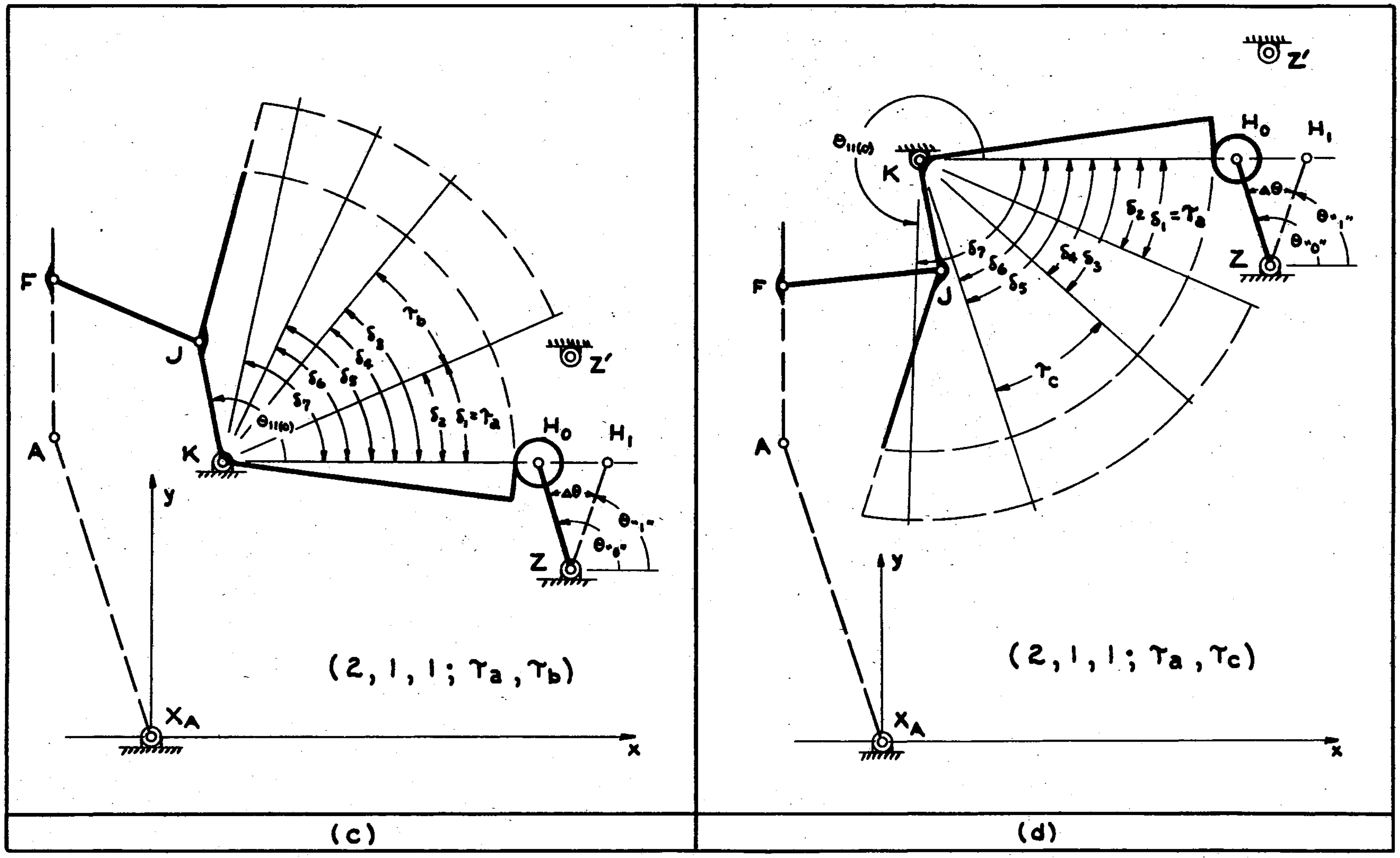


Figure 34. (Continued)

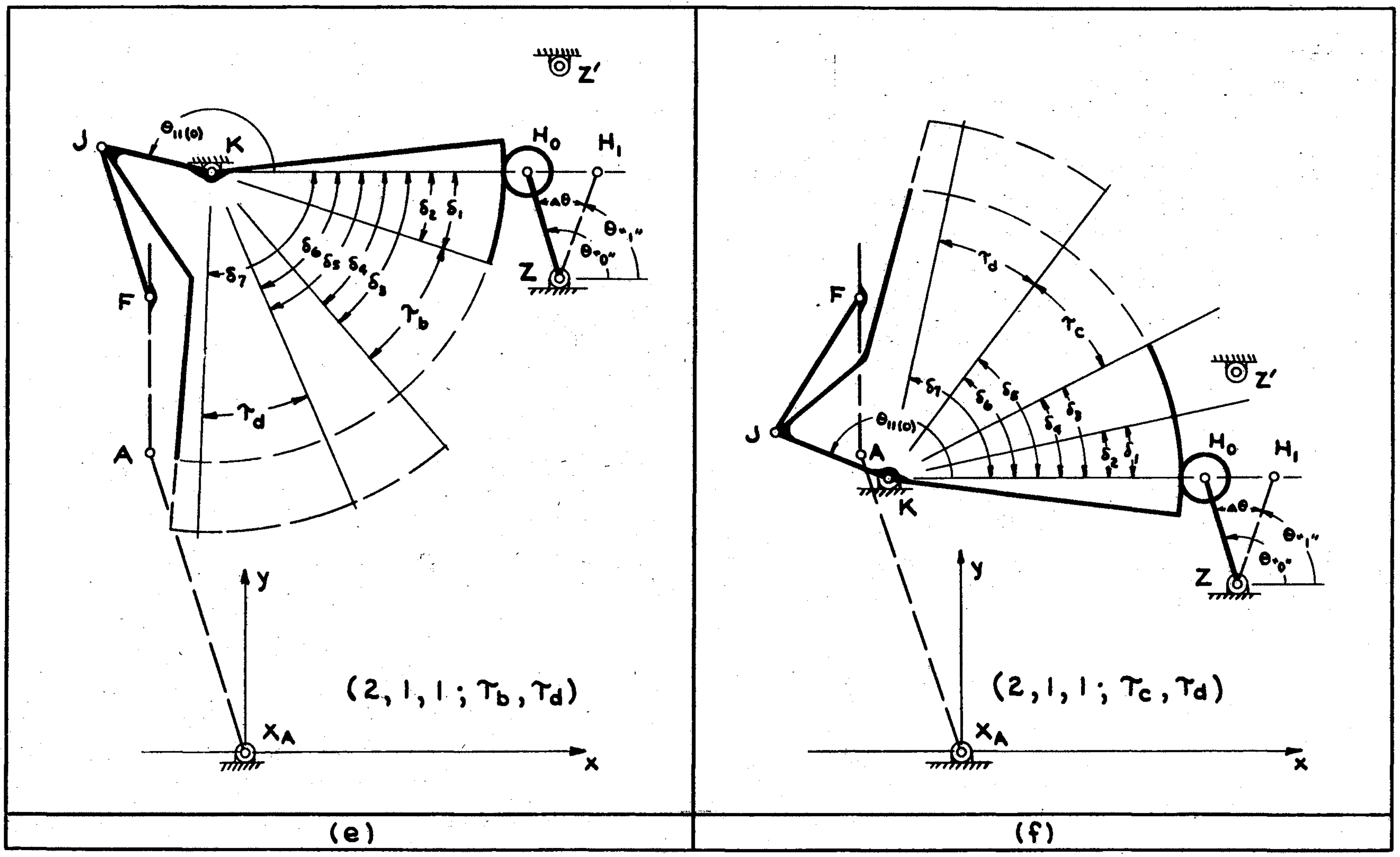


Figure 34. (Continued)

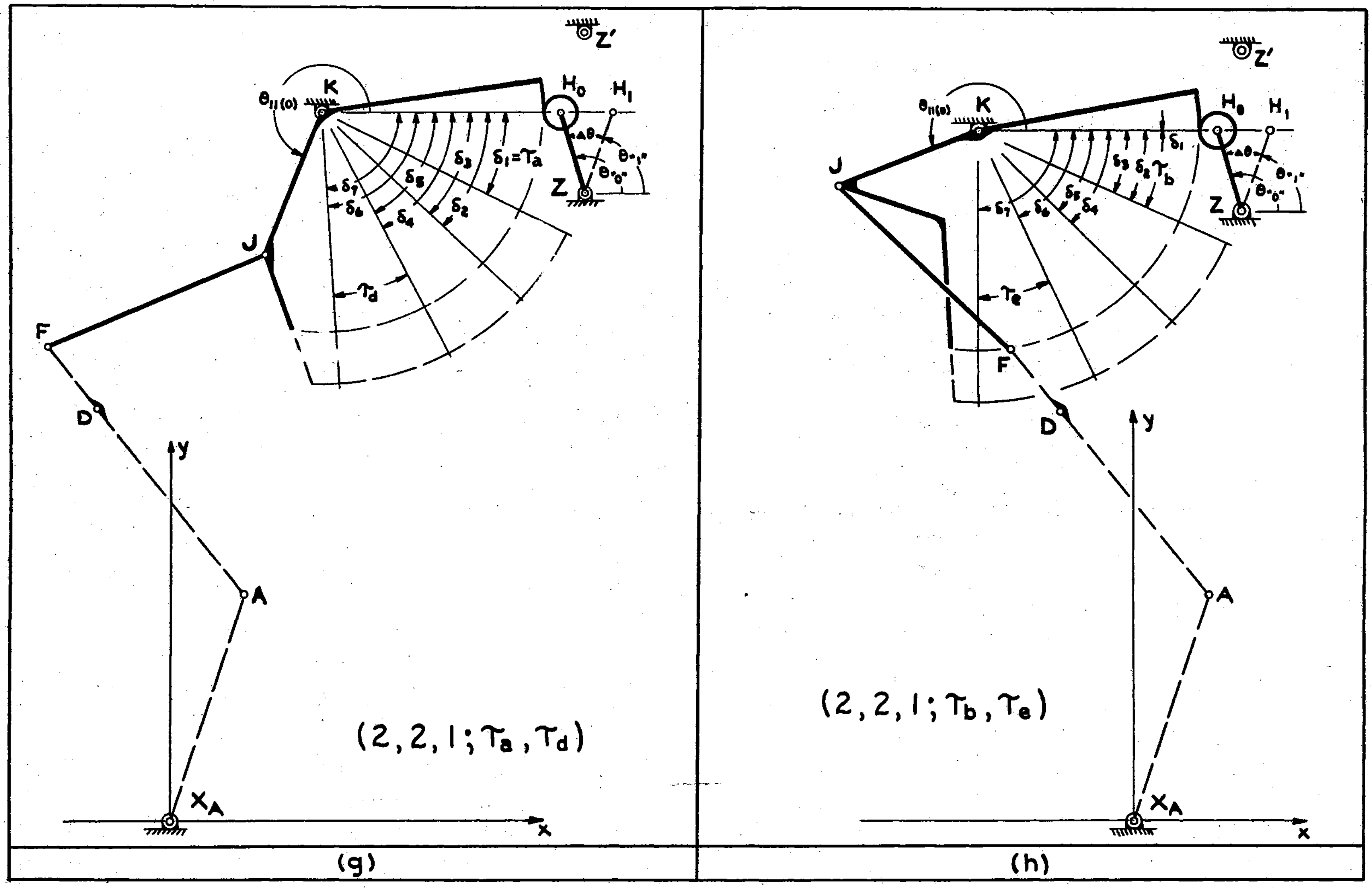


Figure 34. (Continued)

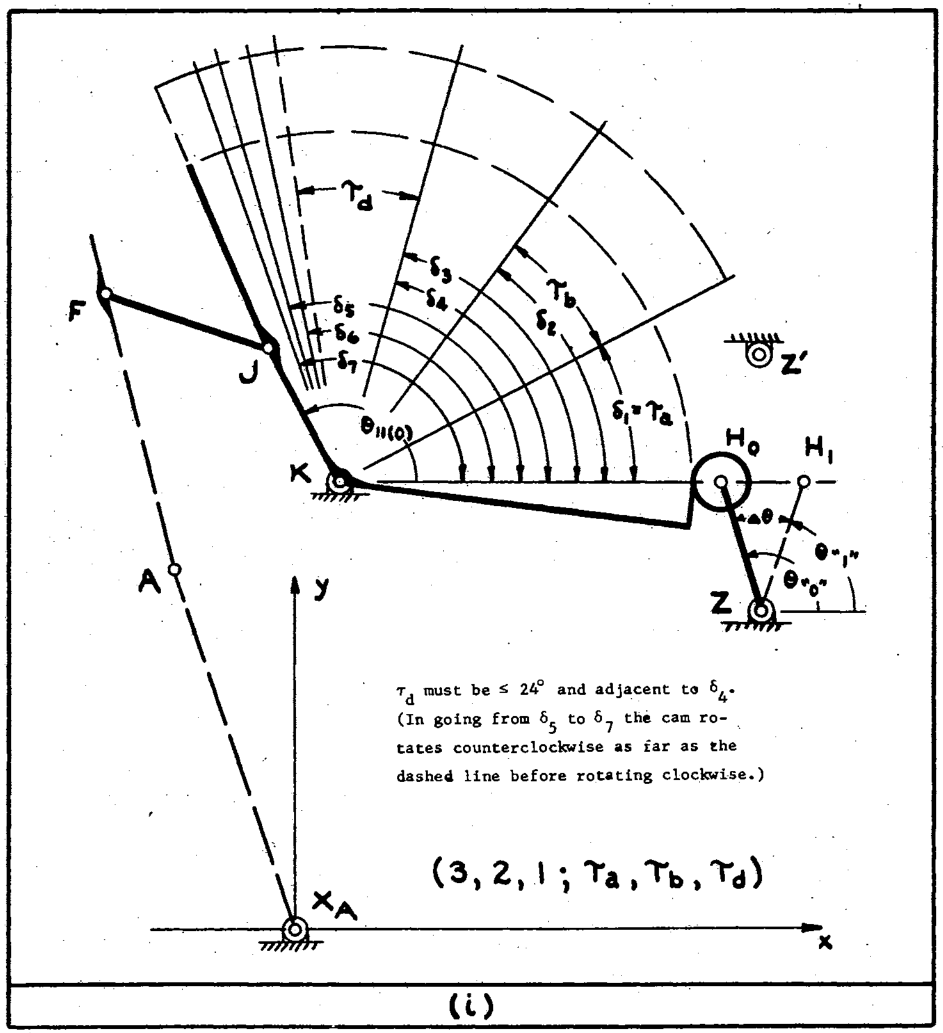


Figure 34. (Continued)

3.8 Examples of Synthesis Using Design Charts

In this section two examples are presented to illustrate the use of the design charts of section 3.7. In each example the simplified Boolean equation is given. The procedural steps given with the design charts are followed by number and the resulting threshold logic generators are shown in Figures 35 and 36 of sections 3.8.1 and 3.8.2.

3.8.1 Example 1

The simplified Boolean equation of the function to be generated is

$$z = \bar{x}_2\bar{x}_3 \dotplus x_1\bar{x}_2 \dotplus x_2x_3$$

The use of the design charts is as follows:

Chart I (Steps)

1. The function is expanded to canonical form as follows:

$$z = (\bar{x}_1 \dotplus x_1)\ \bar{x}_2\bar{x}_3 \dotplus x_1\bar{x}_2\ (\bar{x}_3 \dotplus x_3) \dotplus (\bar{x}_1 \dotplus x_1)\ x_2x_3$$

$$= \bar{x}_1\bar{x}_2\bar{x}_3 \dotplus x_1\bar{x}_2\bar{x}_3 \dotplus x_1\bar{x}_2x_3 \dotplus \bar{x}_1x_2x_3 \dotplus x_1x_2x_3$$

2. The term values are found from the table as

$$TV_i = 1; \quad i = 0,3,4,5,7$$

$$TV_i = 0; \quad i = 1,2,6$$

3. The function value is calculated as

$$FV = \frac{128}{2^0} + \frac{128}{2^3} + \frac{128}{2^4} + \frac{128}{2^5} + \frac{128}{2^7}$$

$$= 128 + 16 + 8 + 4 + 1$$

$$= 157$$

4. Since $FV > 127$, FV_c is found as

$$FV_c = 255 - 157$$

$$= 98$$

Now, using FV_c, $\overline{W};\overline{\tau}$ is found from Table IV as

$$\overline{W};\overline{\tau} = (1,2,-2;\ \tau_a^{10},\ \tau_d^{01})$$

Chart II (Steps)

1. The linkage of Figure 33(d) is selected (see Figure 35).

2. The fixed centers X_A', X_B' and X_C are selected. The input variables are assigned as: x_1 to $X_A'A$, x_2 to X_CC and x_3 to $X_B'B$. (The centers X_A', X_B and X_C' could be used equally well.)

3. The linkage dimensions are determined from Table V, column IV but are not repeated here.

Chart III (Steps)

1. The linkage of Figure 34(g) is selected (see Figure 35).

2. Since FV > 127 and the superscript on τ_a^{10} is 10, the fixed center Z is used.

3. The linkage dimensions are determined from Table VI, column VII but are not repeated here.

4. A rise is placed in the threshold angle τ_a and a return in τ_d.

5. The remainder of the cam has dwell surfaces as shown in Figure 35.

The threshold generator of Figure 35 can now be compared with a system of conventional elements needed to generate the same function. Since the function in simplified form is

$$z = \overline{x}_2\overline{x}_3 \dot{+} x_1\overline{x}_2 \dot{+} x_2x_3$$

the conventional system will include one NOR generator, one AND-NOT generator and one AND generator, each with two inputs, and one OR generator with three inputs. If conventional logic generators of Chapter II

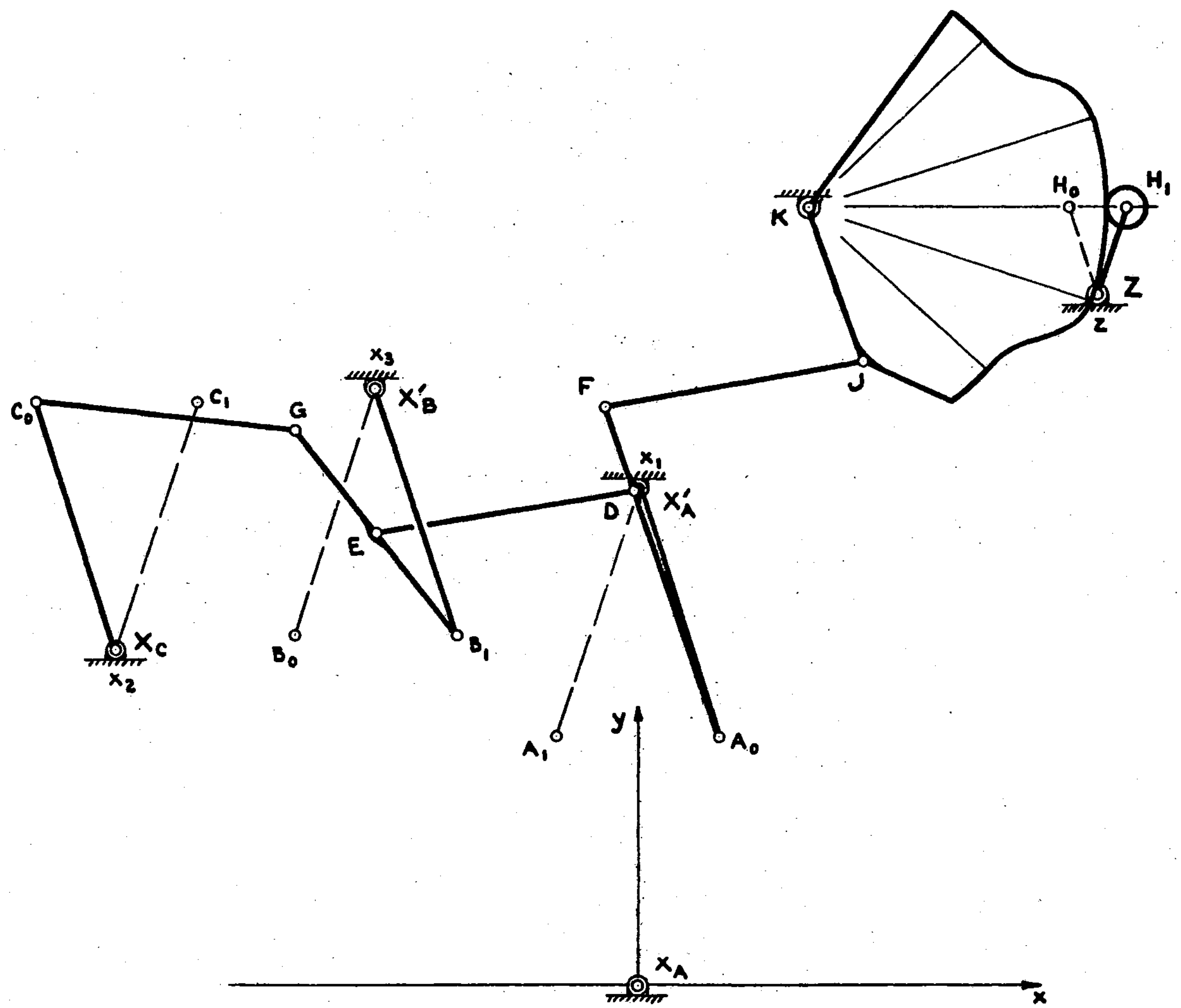


Figure 35. Threshold Logic Generator

are used to implement the system, it is seen that the linkage will be considerably more complex than the threshold generator of Figure 35.

3.8.2 Example 2

In digital systems, adders and subtractors are basic arithmetic units. In this example the threshold generators which act as an adder are selected from the design charts. There are three inputs to an adder. They are the augend (x_1), the addend (x_2), and the input carry (x_3). There are two outputs from the adder which are the sum (z_1) and the output carry (z_2).

The sum of the three inputs is "1" if one of the inputs is "1" or if all three inputs are "1". Hence the equation for z_1 is

$$z_1 = \bar{x}_1\bar{x}_2x_3 + \bar{x}_1x_2\bar{x}_3 + x_1\bar{x}_2\bar{x}_3 + x_1x_2x_3$$

The output carry is "1" if two of the inputs are "1" or if all three inputs are "1". The output carry is given in simplified form as

$$z_2 = x_1x_2 + x_1x_3 + x_2x_3$$

The use of the charts to select the linkage for the sum z_1 is as follows. The steps for finding the linkage for the output carry will not be given, but the linkage will be described.

<u>Chart I</u> (Steps)

1. The function z_1 is in canonical form as given.
2. The term values are found from the table as

$$TV_i = 1; \quad i = 1, 2, 4, 7$$

$$TV_i = 0; \quad i = 0, 3, 5, 6$$

3. The function value is calculated as

$$FV = \frac{128}{2^1} + \frac{128}{2^2} + \frac{128}{2^4} + \frac{128}{2^7}$$

$$= 64 + 32 + 8 + 1$$

$$= 105$$

4. From step 3, $FV < 127$; therefore, $\overline{W};\overline{\tau}$ is found from Table IV as

$$\overline{W};\overline{\tau} = (-1,1,1;\ \tau_a^{10},\ \tau_b^{01},\tau_c^{10})$$

Chart II (Steps)

1. The linkage of Figure 33(b) is selected (see Figure 36).

2. The fixed centers X_A', X_B and X_C are selected. The input variables are assigned as: x_1 to $X_A'A$, x_2 to X_BB and x_3 to X_CC. (Other selections could be used equally well.)

3. The linkage dimensions are determined from Table V, column II but are not repeated here.

Chart III (Steps)

1. The threshold linkage of Figure 34(b) is selected (see Figure 36).

2. Since $FV < 127$ and the superscript on τ_a^{10} is 10, the fixed center Z' is used.

3. The dimensions are determined from Table VI, column II but are not repeated here.

4. A rise is placed in the threshold angle τ_a, a return in τ_b and a rise in τ_c.

5. The remainder of the cam has dwell surfaces as shown in Figure 36.

Figure 36 shows the linkage which generates the sum (z_1) of a binary adder. The linkage needed for the output carry (z_2) is the same as that of Figure 36 except that the input fixed centers are X_A, X_B and

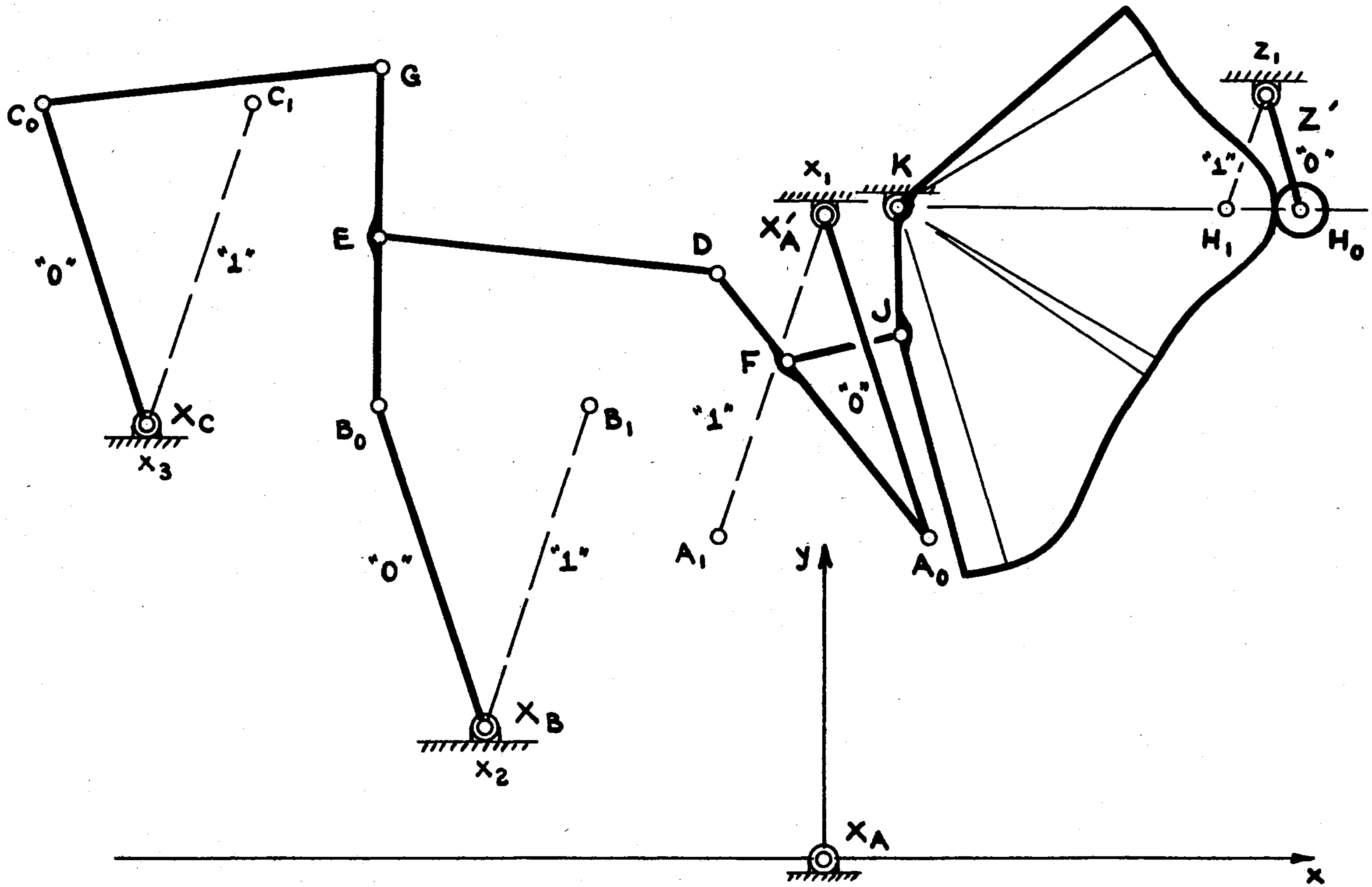


Figure 36. Summing Linkage of a Binary Adder

X_C, the output center is Z and there is a single threshold in the angle τ_b. The separation angles τ_a and τ_c have dwell surfaces.

A comparison of the linkage of Figure 36 with a system of conventional logic generators reveals the real "logic power" of the threshold generators. Since the Boolean function

$$z_1 = \bar{x}_1\bar{x}_2x_3 \dotplus \bar{x}_1x_2\bar{x}_3 \dotplus x_1\bar{x}_2\bar{x}_3 \dotplus x_1x_2x_3$$

is in simplified form, it is seen that three AND-NOT generators and one AND generator, all with three inputs, and one OR generator, with four inputs, are needed to generate the function. By observing the linkages of Chapter II needed to generate this function, it should be clear that the conventional logic system would require far more links than the threshold generator of Figure 36.

CHAPTER IV

CONCLUSIONS AND RECOMMENDATIONS

4.1 Conclusions

The objective of this research has been to develop mechanical linkage systems to implement binary logic. The binary logic has been treated in two basic forms: conventional logic and threshold logic.

In the realm of conventional logic, simple four-link mechanisms have been combined in systems or assemblies to generate the basic logic functions OR, AND, NOT, NOR, NAND and MEMORY. Slightly more complex mechanisms have been used to generate the COINCIDENCE and OR-EXCLUSIVE functions. The proof of twenty-three theorems of Boolean algebra has been established with illustrative mechanical systems.

The systems of linkages which generate the binary logic have been shown to be passive because energy can enter and leave only at the inputs and outputs. To design active logic systems, a bistable mechanical amplifier has been developed. The active systems are composed of a linkage which generates the binary logic and one of the mechanical amplifiers.

The application of existing methods of logic circuit synthesis to mechanical logic circuits has been illustrated. A practical problem of a sorting mechanism which involves combinational logic has been solved using a passive mechanical circuit. A shaft speed control problem involving sequential logic has been solved with an active mechanical circuit. A physical model of the speed control circuit was constructed. The model served to confirm that the logic synthesis was correct. It also confirmed that the function generators and amplifiers operate as proposed.

In conclusion regarding conventional logic systems, it should be noted that the mechanical logic function generators developed can be used to implement any control circuit that can be expressed in Boolean equation form.

In the area of threshold logic, linkages have been synthesized to generate the 256 functions of three binary variables. The linkages, called threshold logic generators, are made up of two component linkages. One component is a "separator linkage" and the other is a "threshold linkage".

The selection of these components was suggested by the nature of the Multi-Threshold Weight Threshold Vector ($\overline{W}$; $\overline{T}$) itself. The weight part of the vector ($\overline{W}$) suggested a linear summing device, and the separator linkage served in this capacity. The threshold part of the vector ($\overline{T}$) suggested a device to control the "on" and "off" of the output at certain position of the summing device, and the threshold linkage served in this capacity.

The study shows how all 256 functions can be generated through the use of five basic separator linkages and nineteen basic separator linkages.

The separator linkages have been synthesized from an eight-link three-degree-of-freedom kinematic chain and from the five-link two-degree-of-freedom kinematic chain. Structural synthesis was applied to enumerate all of the eight-link three-degree-of-freedom and the five-link two-degree-of-freedom kinematic chains. Then, to these chains, performance criteria were applied to select favorable linkages for the separators. The linkages selected were arranged so that their dimensional synthesis as a linear summing device was done with a linear approximation.

Cam linkages have been used for the threshold linkages. A cam linkage with an oscillating roller follower output was selected. The input to the threshold linkage was required to follow a path. From many six-link cam linkages enumerated by Hain [35], the two which satisfied these input-output requirements were considered; one linkage was derived from the Watt's chain and the other from the Stephenson's chain. Performance criteria were applied in selecting the linkage derived from the Watt's chain. With the cam linkage selected, simple rise, dwell and return programs have provided for maintaining the output link precisely in its "0" or "1" positions at all times except when a change in the output state was required.

In the study it has been noted that the cams used with the threshold linkages provide an easy way to obtain the required dwells of the output link. A threshold linkage using only revolute pairs could undoubtedly be synthesized with satisfactory dwells of the output link; however, it is expected that the number of links would have to be increased.

The separator and threshold linkages have been synthesized dimensionally. A combination of graphical and analytical techniques have been used. Favorable link length ratios and cam pressure angles have been maintained throughout the synthesis. A graphical technique was devised to show that all transmission angles in the synthesis have been maintained at 45° or larger.

Design charts with linkage dimensions have been included in section 3.7. The design charts make possible the direct selection of the separator linkage and the threshold linkage which combine to generate any one of the 256 functions of three variables. In making the selection from the charts, no knowledge is needed of Boolean logic or of the theories of linkage synthesis.

Two example problems showing the use of threshold generators have been presented. One example selects the generator whose output gives the binary sum of three binary variables. The two examples serve as a good conclusion to threshold logic, since both illustrate the great "logical power" of threshold logic generators.

4.2 Recommendations

To further develop the application of mechanical linkages to logical control systems, it is recommended that research be conducted in the following areas:

1. Systematic application of mechanical linkages to the synthesis of proportional and hybrid (binary and proportional mixed) control systems. This study treated the systematic application of linkages to binary logic only. Electronic, fluidic and fluid power elements have been used systematically in both proportional and hybrid systems. The use of mechanical counterparts should also be considered. Mechanical devices will have to be designed to perform such functions as comparitor, digitizer and rectifier.

2. Development of other types of mechanical amplifiers. Mechanical clutches are in reality power amplifiers. Many types of clutches exist and some of them probably can be altered to form bistable amplifiers, hopefully to form amplifiers with advantages over the amplifier developed in the present study.

3. Study of the dynamics of the mechanical systems. Time response, "fan-in" and "fan-out" of the logic generators are important with regard to the applicability of the elements to particular problems. ("Fan-in" is the maximum number of input connections which can be made to a logic generator without causing it to malfunction. "Fan-out" is the maximum number of elements which can be connected in parallel to the output of a similar element and still be switched when the output of the first element changes.)

4. Application of the threshold generator linkage to functions of more than three variables. As the number of variable increases, the number of possible thresholds increases. Also, the horizontal separation (speaking in terms of the component method of this study) of the output point decreases on a separator linkage of given size. These two conditions are expected to make it more difficult to synthesize the four variable function generators than the three variable generators of the present study.

5. Use of alternate linkages and methods of synthesis for the threshold logic generators. The synthesis of threshold linkages using revolute pairs only and computerized optimization techniques should be investigated. In order to provide satisfactory dwells of the output link in its "0" and "1" positions at all times except when a change in

the output state is required, the linkage is expected to need more than the six links needed for the cam linkage of this study.

The synthesis of the separator linkages by some direct computer optimization technique, instead of the linear approximation method used in this study, should also be considered. Linkages with improved transmission characteristics and link length ratios can undoubtedly be synthesized.

Another alternative is to use basically the same linkage shown in Figure 24, but to include the loop $X_A AFJKX_A$ as part of the separator. The cam and the output follower HY will then become the only elements of the threshold linkage. The use of linear approximations can be eliminated and computerized synthesis can be used to directly maximize the threshold angles, minimize the cam rotation angles where thresholds do not exist and maximize the minimum transmission angles. It seems that this synthesis proposition can provide an excellent function generator; however, the optimization program will be quite complex.

6. Consideration of a direct synthesis method. Instead of using the two components of a threshold generator, a single linkage with revolute pairs should be considered. Such a linkage will have as many degrees of freedom as the number of inputs of the function to be generated.

Analytical optimization methods can be used and the synthesis equations can be written directly from the truth table of the function.

As with the other revolute pair linkages, difficulty is anticipated in keeping the output link immobile during all input state changes that do not require an output change. To strictly satisfy this condition, it is expected that the number of links needed will be greater than the theoretical minimum. Also to satisfy this condition, it is expected that design equations will be needed for intermediate positions (positions other than the "0" and "1" positions) of the input links. These intermediate position considerations will signify a considerable increase in the number of design equations to be written.

7. Study of the possibility of miniturization of the logic function generators. For many applications it is desirable to have function generators as small as possible. However, as linkages are made very small, precision and tolerances become significant factors and the linkages become expensive or even impossible to manufacture. The extent to which the generators can be economically miniturized, thus, has a bearing on the general usability of the function generators.

BIBLIOGRAPHY

(1) Shannon, C. E. "A Symbolic Analysis of Relay and Switching Circuits." Trans. AIEE, Vol. 57 (1938), 713-723.

(2) Huffman, D. A. "The Synthesis of Sequential Switching Circuits." Journal of the Franklin Institute, Vol. 257, No. 3 (1954), 161-190 and Vol. 257, No. 4 (1954), 275-303.

(3) Fitch, E. C. Fluid Logic. (Unpub. manuscript, Stillwater, Okla: School of Mechanical and Aerospace Engineering, 1966.)

(4) Foster, K., and G. A. Parker. Fluidics: Components and Circuits. New York: Wiley-Interscience, 1970.

(5) Cole, J. H. "Synthesis of Optimum Complex Fluid Logic Sequential Circuits." (Unpub. Ph. D. dissertation, Stillwater: Oklahoma State University, 1968.)

(6) Maroney, G. E. "A Synthesis Technique for Asynchronous Digital Control Networks." (Unpub. M. S. thesis, Oklahoma State University, 1969.)

(7) Woods, R. L. "The State Matrix Method for the Synthesis of Digital Logic Systems." (Unpub. M. S. thesis, Oklahoma State University, 1970.)

(8) Beizer, B., and S. W. Leibholz. "Analyzing Combinational Circuits by Boolean Matrices and Karnaugh Maps." Electrical Manufacturing, Vol. 62, No. 1 (July, 1958), 100-109.

(9) Bennett, W. S. "Binary Logic." Product Engineering, Vol. 33, No. 1 (Jan. 8, 1962), 79-83 and Vol. 34, No. 21 (Oct. 14, 1963), 95-104.

(10) Brandt, D. A. "Mechanical Binary Logic Function Generators, Vol. 1 and 2." (Unpub. Ph. D. dissertation, Purdue University, August, 1963.)

(11) Szekely, I., T. Muresan, and M. Lelutiu. "Mecanismes a Commande Logique Avec des Elements Rigides et Elastiques." Rev. Roum. Sci. Techn. - Mec Appl. Vol. 10, No. 4 (1965), 1059-1069.

(12) Szekely, I., and T. Muresan. "Les Regimes de Fonctionnement des Mecanismes a Elements Rigides Multipositionnels." Rev. Roum Sci. Techn. - Mec. Appl. Vol. 11, No. 3 (1966), 797-805.

(13) McNaughton, R. "Unate Truth Functions." IRE Trans. Electron. Computers, EC-10, March 1961, pp. 1-6, and previously available as TEchnical Report No. 4, Applied Mathematics and Statistics Laboratory, Stanford University, October, 1957.

(14) Lewis, P. M., and C. L. Coates. Threshold Logic. New York: John Wiley and Sons, 1967.

(15) Dertouzos, M. L. Threshold Logic: A Synthesis Approach. Cambridge, Mass: The M.I. T. Press, 1965.

(16) Sheng, C.L. Threshold Logic. Toronto: The Ryerson Press; New York: The Academic Press, 1969.

(17) Winder, R. O. "Threshold Logic." (Unpub. Ph. D. dissertation, Princeton University, 1962.)

(18) Haring, D. R. "Multi-Threshold Threshold Elements." IEEE Trans. Electron. Computers, Vol. EC - 15, No. 1 (Feb. 1966), 45-65.

(19) Mow, C. W. and K. S. Fu. "An Approach for the Realization of Multithreshold Threshold Elements." IEEE Trans. Electron. Computers, Vol. C - 17, No. 1 (Jan. 1968), 32-46.

(20) Haring D. R. and R. J. Diephuis. "A Realization Procedure for Multithreshold Threshold Elements." IEEE Trans. Electron. Computers, Vol. EC - 16, No. 6 (Dec. 1967), 828-835.

(21) Haring, D. R. "Multi-Threshold Building Blocks." IEEE Trans. Electron. Computers, Vol. EC - [illegible], No. 4 (August 1966), 662-663.

(22) Haring D. R. And D. Ohori. "A Tabular Method for the Synthesis of Multi-threshold Threshold Elements." IEEE Trans. Electron. Computers, Vol. EC - 16, No. 2 (April 1967), 216-220.

(23) Menger, K. S. "A Modulo Two Adder for Three Inputs Using a Single Tunnel Diode." IRE Trans. Electron. Computers, Vol. EC-10 (Sept. 1961), 530-531.

(24) Amodei, J. J. and W. F. Kosonocky. "High-speed Logic Circuits Using Common-base Transistors and Tunnel Diodes." RCA Review, Vol. 22 (Dec. 1961), 669-684.

(25) Amodei, J. J. "High-speed Adders and Comparators Using Transistors and Tunnel Diodes." IEEE Trans. Electron. Computers, Vol. EC - 13 (Oct. 1964), 563-575.

(26) Renton, C. A., and B. Rabinovici. "Tunnel-diode Full Binary Adder. IRE Trans. Electron. Computers, Vol. EC - 11 (April 1962), 213-217.

(27) Horton, J. W., and A. G. Anderson. "A Full Binary Adder Employing Two Negative Resistance Diodes." IBM J. Res. and Dev., Vol. 2 (July 1958), 223-231.

(28) Rabinovice, B., and C. A. Renton. "A Transistor Flip-flop Full Binary Adder. IRE Trans. Electron. Computers, Vol. EC - 11 (Dec. 1962), 789-791.

(29) Miro, J., H. Apple, and D. Vlack. "A Nonlinearly Separable Logic Circuit." Proc. IEEE, Vol. 52 (Sept. 1964), 1081-1082.

(30) Eckerlin, H. M. and N. R. Bell. "Fluidic Realization of Threshold Logic." ASME Publication, 68 - WA/FE - 28, pp. 1-9.

(31) Chironis, N. P. "Whiffletree Linkage Whisks a Load to a New Position." Product Engineering, Vol. 38, No. 14 (July 3, 1967), 58-60.

(32) Haas, S. L. and F. R. E. Crossley. "Structural Synthesis of a Four-Bit Binary Adding Mechanism." Trans. of the ASME, Journal of Engineering for Industry, Vol. 91, No. 1 (Feb. 1969), 240-250.

(33) Manning, J. R. "Fluidic Control Devices and Systems. ASME Publication, 70 - Flcs - 15, (1970), 4.

(34) Davies, T. H., and F. E. Crossley. "Structural Analysis of Plane Linkages by Franke's Condensed Notation." Journal of Mechanisms, Vol. 1 (1966), 171-183.

(35) Hain, K. "The Production of Large Oscillating Angles by Cam-Linkage Mechanisms." Proceedings - International Conference on Mechanisms (Yale University, March 27-30, 1961), 63-90.

(36) Jensen, P. W. Cam Design and Manufacture. New York: The Industrial Press, 1965.

APPENDIX A

EXAMPLE OF LOGIC CIRCUIT SYNTHESIS

This appendix presents three steps of the logic circuit synthesis for the example of the soap box sorter. The steps presented here are the three which are not dependent upon the fact that the control circuit is synthesized from linkage elements.

The three steps are:

1. Write a description of the output requirements and the input conditions. (Such a description must be logically meaningful, unambiguous, and non-contradictory.)

In the direction of movement on the conveyor the boxes have a possibility of two lengths a_1 and a_2, where a_1 is less than a_2. The cross-section (section perpendicular to the conveyor) has a possibility of two dimensions (b and c) for height and width. The dimension b is less than c. These cross-sections can be squares (b x b or c x c) or they can be rectangles (b x c or c x b).

The boxes are to sorted if:

a) They have the length a_2, or

b) They have a rectangular cross-section (b x c or c x b).

All other boxes are to be retained on the main conveyor.

The cam actuated input variables A, B, C, and D are arranged to function in the following manner:

a) Variable A detects the height and is normally in the "off" position. It is "on" only when a box of height c is passing.

b) Variable B detects the width and is normally "off". It is "on" only when a box of width c is passing.

c) Variables C and D detect the length, and they are normally "off". At any time a box is passing one of the variables, the variable is in the "on" position. The cam followers, which provide the two variables, are separated along the conveyor by a distance greater than a_1 but less than a_2. Thus, if a box of length a_1 passes, C and D are not "on" at the same time. If a box of length a_2 passes, both C and D are "on".

d) The cam followers are placed so the boxes reach the variables in the sequence C, A, B, and D (see Figure 13). The distance between the cam followers A and D is also less than a_1.

e) The boxes are spaced on the conveyor so that no two boxes can be passing the input signals at the same time.

2. Devise a method to correctly assign the state of the output variable for every possible combination of the input states.

To fulfill this requirement a truth table and a flow chart are used as shown in Table VII and Figure 37. The truth table is prepared by writing the 2^4, or 16, possible combinations of the input states. Each combination appears as a "row". The values of Z describing the output state are assigned by using the flow chart of Figure 37.

The flow chart shows the sequence of the input states caused by each distinct box that passes the stations C, A, B, and D. It is used to determine which input state combinations cause the output variable Z to be "on". The chart is constructed as follows:

TABLE VII

TRUTH TABLE OF CONTROL CIRCUIT FOR
SORTER MECHANISM

	Inputs				Output
Row	A	B	C	D	Z
1	0	0	0	0	0
2	0	0	0	1	0
3	0	0	1	1	1
4	0	0	1	0	0
5	0	1	1	0	1
6	0	1	1	1	-
7	0	1	0	1	0
8	0	1	0	0	-
9	1	1	0	0	0
10	1	1	0	1	0
11	1	1	1	1	1
12	1	1	1	0	0
13	1	0	1	0	0
14	1	0	1	1	-
15	1	0	0	1	1
16	1	0	0	0	-

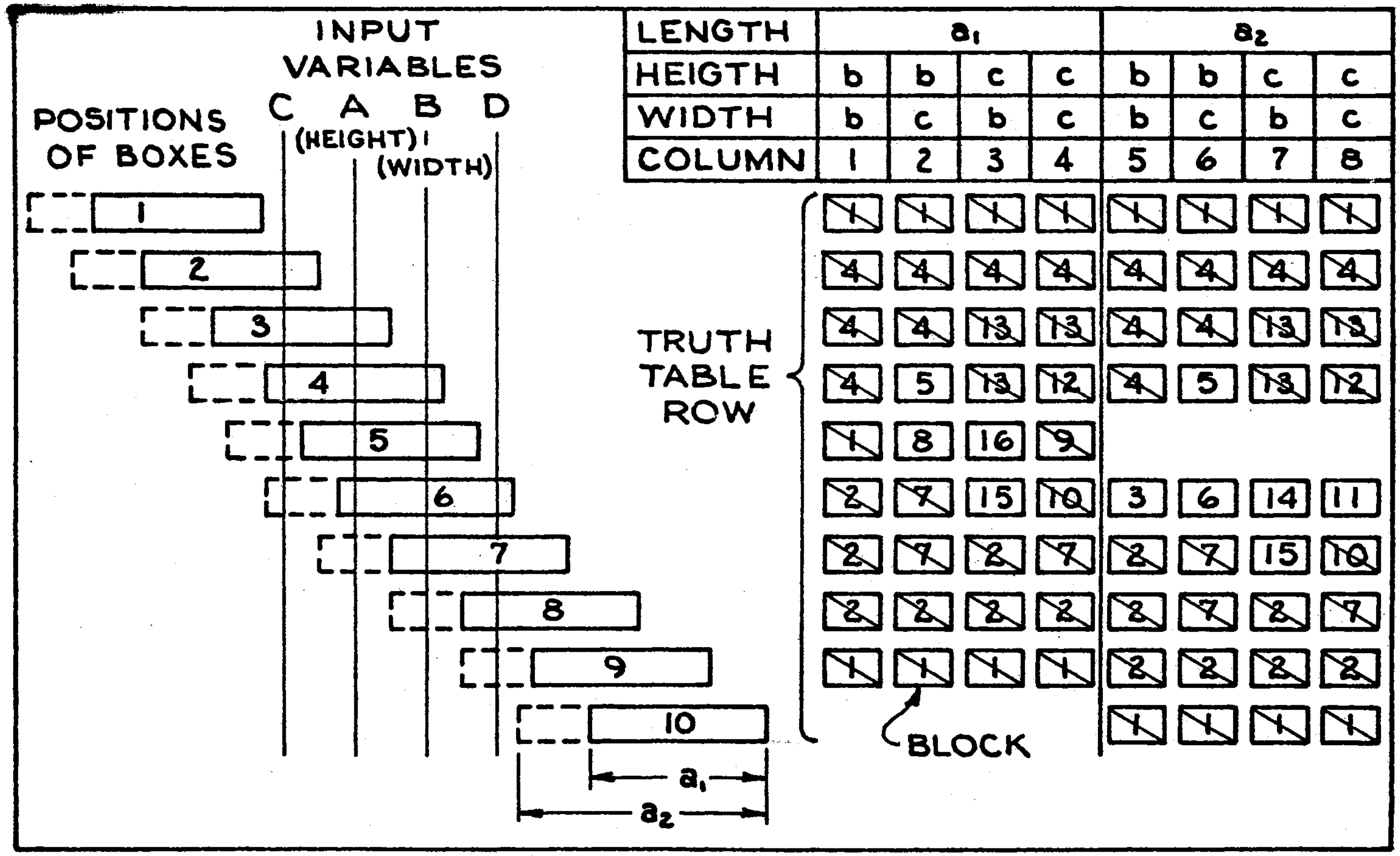


Figure 37. Flow Chart of Truth Table Rows for Consecutive Positions of Boxes

a) Each distinct "position" of the boxes along the conveyor with respect to the input variables A, B, C, and D is represented at the left. Each solid rectangle represents a position of a box of length a_1, and each solid plus dashed rectangle represents a position of a box of length a_2. There are 10 such positions, but it is noted that for a box of length a_1 positions 9 and 10 are not distinct. Likewise for a box of length a_2, positions 4 and 5 are not distinct.

b) A "column" is prepared for each distinct box that passes. The different combinations of length, height, and width account for eight distinct boxes. In each column, for each position, a "block" is drawn. Then in each block is placed the number of the truth table row representing the corresponding input state combination. For instance, when a box of length a_1, height b, and width c is in position 4, the input states are A = 0, B = 1, C = 1, and D = 0. This combination is given in row 5 of Table VII, so the number 5 is placed in the block of position 4 and column 2.

By reading a column from top to bottom the input state sequence caused by the passing of a particular box is known. The sequence caused by a box of column 2 is given by the rows 1, 4, 4, 5, 8, 7, 7, 2 and 1 in the truth table. Of course, if the box is sorted at some intermediate position the sequence is broken and the inputs return to "off".

c) Only the boxes of columns 1 and 4 are retained on the main conveyor since all other boxes are of length a_2 or have a rectangular cross-section (b x c or c x b). To retain the boxes, the output state is "off" in all truth table rows found in these two columns. The rows are 1, 4, 2, 13, 12, 9, 10, and 7. Wherever these row numbers appear in the flow chart, a diagonal line is drawn through the block to make it clear that Z cannot be "on" in that position.

d) In each of the six columns 2, 3, 5, 6, 7, and 8 sorting occurs. Thus Z must be "on" in at least one row represented in these columns. In columns 5 and 8 it is seen that Z must be "on" in rows 3 and 11. However, columns 2, 3, 6, and 7 must be further examined. If Z in "on" in rows 5 and 15, all sorting requirements are met, and the output Z corresponding to rows 8, 16, 6, and 14 has "don't care" situations. This means that Z can be either "off" or "on" for the rows 8, 16, 6, and 14. The "don't care" situation is represented using a dash (-) in the truth table.

3. <u>Write a simplified Boolean equation for each output variable.</u>

The output equation is formed by including a term for every truth table row where the output Z in "on." Where a "don't care" condition appears, a term may or may not be written. The result in this case can be shown to be as simple as possible by including terms for the rows 6 and 14. Hence from the truth table (Table VII)

$$Z = \bar{A}\bar{B}CD \dot{+} \bar{A}BC\bar{D} \dot{+} \bar{A}BCD \dot{+} ABCD \dot{+} A\bar{B}CD + A\bar{B}\bar{C}D \qquad (A.1)$$

This equation is simplified by using the theorems of Figure 12 as follows:

See theorem(s)

$$Z = (\bar{A}\bar{B} \dot{+} \bar{A}B \dot{+} AB \dot{+} A\bar{B})CD \dot{+} \bar{A}BC\bar{D} \dot{+} A\bar{B}\bar{C}D \qquad (14, 21)$$

$$= [\bar{A}(\bar{B} \dot{+} B) \dot{+} A(B + \bar{B})]CD \dot{+} \bar{A}BC\bar{D} \dot{+} A\bar{B}\bar{C}D \qquad (21)$$

$$= (\bar{A} \dotplus A)CD \dotplus \bar{A}BC\bar{D} \dotplus A\bar{B}\bar{C}D \qquad (7, 4)$$

$$= CD \dotplus \bar{A}BC\bar{D} \dotplus A\bar{B}\bar{C}D \qquad (7, 4)$$

$$= C(D \dotplus \bar{D}\bar{A}B) \dotplus A\bar{B}\bar{C}D \qquad (15, 21)$$

$$= C(D \dotplus \bar{A}B) \dotplus A\bar{B}\bar{C}D \qquad (17)$$

$$= CD \dotplus A\bar{B}\bar{C}D \dotplus C\bar{A}B \qquad (21, 14)$$

$$= D(C \dotplus \bar{C}A\bar{B}) \dotplus C\bar{A}B \qquad (15, 21)$$

$$= D(C \dotplus A\bar{B}) \dotplus C\bar{A}B \qquad (17)$$

and finally

$$Z = CD \dotplus A\bar{B}D \dotplus \bar{A}BC \qquad (21, 15)$$

$$(A.2)$$

It should be noted that other methods, such as the use of Karnaugh Maps, may be employed to simplify the logic equation.

APPENDIX B

EXAMPLE OF ACTIVE LOGIC CIRCUIT SYNTHESIS

This appendix presents three steps of the logic circuit synthesis for the example of the rotating shaft speed controller. The steps presented here are the three which are not dependent upon the fact that the control circuit is synthesized from linkage elements.

The second part of this appendix presents the reason for the need of a passive AND-NOT generator in the control circuit of Figure 19.

B.1 Three Synthesis Steps

The three steps of synthesis are:

1. Write a description of the output requirements and the input conditions.

There are three possible combinations of the output signals Z_1 and Z_2. They are Z_1Z_2 = 00 (both the motor and supercharger are off), 10 (the motor is on and the supercharger is off) and 11 (both the motor and the supercharger are on). The combination of 01 never exists since the supercharger cannot be on unless the motor is on.

Since the motor and supercharger can only be "off" or "on" (there are no intermediate settings) the torque provided is never exactly right to maintain a constant speed. For this reason the system is constantly going through speed cycles. The desired speed cycle is as follows:

1) The output variables are $Z_1Z_2 = 10$ while the speed increases and the input variables sequence through $x_1x_2 = 00$, 10, and 11.

2) The output variables are at $Z_1Z_2 = 00$ while the speed decreases and the input variables sequence through $x_1x_2 = 11$, 10, 00.

At any time the input variables deviate from the desired cycle the additional torque of the supercharger is needed to restore the system to the desired cycle.

2. <u>Devise a method to correctly assign the state of the output variables for every possible combination of the input states.</u>

Because of the cycling involved with the input states, a primitive flow table provides a valuable method for depicting all possible input state combinations and their sequences. It also shows the associated output states. The primitive flow table is shown in Table VIII.

In Table VIII the circled numbers represent "stable states." The stable states indicate combinations of the input variables and the corresponding output states at which the system will operate indefinately if no changes are made to the input variables. It is convenient to interpret the table by starting at stable state 1 and following the possible speed cycles. The circled 1 indicates that the input combination of $x_1x_2 = 00$ and the output combination of $Z_1Z_2 = 10$ produces a stable state. The other numbers in a row indicate the next possible input state combinations. These numbers represent "unstable states." In row 1 the unstable states are 2 and 3. If the system is running in stable state 1, and then the speed of the shaft increases sufficiently, the input state combination changes to $x_1x_2 = 10$. The system has changed to unstable state 2 and must pass to a stable state 2 which has the input combination of $x_1x_2 = 10$. The stable state 2 is found in row

TABLE VIII

PRIMITIVE FLOW TABLE

ROW	INPUTS $x_1 x_2$				OUTPUTS	
	00	01	11	10	Z_1	Z_2
1	1	3	-	2	1	0
2	4	-	5	2	1	0
3	1	3	-	-	1	1
4	4	3	-	2	1	1
5	-	-	5	6	0	0
6	1	-	-	6	0	0

2, and the associated output remains as $Z_1Z_2 = 10$. If the shaft speed continues to increase from this stable state 2, the next input state combination is $x_1x_2 = 11$. The corresponding unstable state is the state 5 of row 2, and from here the system passes to the stable state 5 of row 5 where the output combination is $Z_1Z_2 = 00$. (Since the speed has increased too much the motor is shut off so the shaft will slow down to the desired speed.)

Now the speed decreases and the input combination becomes $x_1x_2 = 10$ as given by the unstable state 6 of row 5. The next state is stable state 6 of row 6 where the outputs remain $Z_1Z_2 = 00$. Since the motor is still "off" the speed must continue to reduce so that the inputs again become $x_1x_2 = 00$. This unstable state is given by 1 in row 6, and the following stable state is 1 of row 1. Now the desired cycle of speed variation is completed, and the system is ready to make the same cycle again or to make some other cycle. All possible speed cycles can be mapped out on the primitive flow table in a similar way.

3. Write a simplified Boolean equation for each output variable.

In the primitive flow table it is seen that two of the input state combinations have unique stable state output combinations. When the input combination is $x_1x_2 = 01$ the only stable state output combination is $Z_1Z_2 = 11$. When the input combination is $x_1x_2 = 11$ the only stable state output combination is $Z_1Z_2 = 00$. It is also seen that for the input combination of $x_1x_2 = 00$ there are two output combinations. These combinations are $Z_1Z_2 = 10$ and 11 for the stable states 1 and 4, respectively. For the input combination of $x_1x_2 = 10$ there are two output combinations, $Z_1Z_2 = 10$ and 00, for the stable states 2 and 6, respectively.

Because there are input state combinations with more than one stable state output combination, this problem is one of sequential logic and is much more complicated than the previous example. The output equations cannot be written directly from the primitive flow table as the output equation was written from the truth table in the example of Appendix A.

Because this is a problem of sequential logic, memory elements must be used. Equations for the memory elements as well as output equations must be written. The memory element equations are called secondary equations and are of the form

$$Y = S \dotplus y\bar{R} \tag{B.1}$$

where: y is the output signal from the memory,

S is the set signal,

R is the reset signal.

The output and secondary equations are obtained through the use of Table IX. There are six columns in the table and they are constructed as follows:

1. Columns I, II, III, and VI contain the same information as the primitive flow table (Table VIII). Each number in column I represents the stable state of that number. (For the following discussion it is not necessary to say "stable state"; simply "state" is sufficient.) Column II contains the possible states that can follow next after the state of column I. Thus state 2 or state 3 can follow next after state 1. Columns III and IV contain the values of the output and input variables which correspond to the states of column 1.

TABLE IX

SYNTHESIS TABLE

I	II	III		IV		V		VI			
STATE	NEXT	OUT-PUTS		TOTAL SIGNAL				MEMORIES			
				INPUTS		SEC.		SET	RES.	SET	RES.
		Z_1	Z_2	x_1	x_2	y_1	y_2	Y_1	$\overline{Y}_1$	Y_2	$\overline{Y}_2$
1	2, 3	1	0	0	0	1	1	-	0	S	0
2	4, 5	1	0	1	0	0	1	0	R	-	0
3	1	1	1	0	1	1	1	S	0	-	0
4	2, 3	1	1	0	0	0	1	0	-	S	0
5	6	0	0	1	1	1	0	-	0	0	R
6	1	0	0	1	0	1	0	S	0	0	-

OUTPUT AND SECONDARY EQUATIONS

$Z_1 = y_2 \qquad Z_2 = \bar{x}_1(x_2 \dotplus \bar{y}_1)$

$Y_1 = x_2 \dotplus y_1\,(\overline{x_1 \cdot \bar{x}_2 \cdot y_2})$

$Y_2 = \bar{x}_1 \dotplus y_2\,(\overline{x_1 \cdot x_2})$

	SET	RESET
Y_1	x_2	$x_1 \cdot \bar{x}_2 \cdot y_2$
Y_2	$\bar{x}_1$	$x_1 \cdot x_2$

2. Columns V and VI are developed with several intermediate steps. The following description elaborates the steps for determining the entires of these columns and for writing the output and secondary equations.

a) The secondary variable y_1 is introduced to provide a signal for state 1 which is distinct from the signal of state 4. In column V under y_1 a "1" is entered in state 1 and a "0" is entered in state 4. The signals for these two states are now distinct and are $x_1x_2y_1 = 001$ and $x_1x_2y_1 = 000$, respectively.

b) The secondary variable y_2 is introduced to provide a signal for state 2 which is distinct from the signal of state 6. Under y_2 a "1" is entered in state 2 and a "0" is entered in state 6.

c) Column VI is now prepared to indicate the states in which the MEMORY generators Y_1 and Y_2 must be SET and RESET. (The capital letters Y_1 and Y_2 refer to the function generators themselves, and the small letters y_1 and y_2 refer to the output variables of the generators.) Consider state 1, for example, where the value of y_1 is prescribed in step 2.a) to be "1" ("on"). It is convenient, although not absolutely necessary, that the corresponding MEMORY generator Y_1 be SET in a state that precedes state 1. This "prepared path" concept contributes to decreasing the time lost between input variable changes and the production of the new output variable values. Thus, since states 3 and 6 (as seen in column II) can precede state 1, an S (representing SET) is placed in states 3 and 6 of column VI Y_1.

Using the same reasoning an S is placed in column VI Y_2 in states 1 and 4. This assures that the Y_2 MEMORY generator is SET in states that precede state 2. The "prepared path" concept is also observed in the resetting of the MEMORY generators. Thus an R (representing RESET) is placed in state 2 under $\bar{Y}_1$ so that the Y_2 MEMORY generator is reset in the state preceding state 4. Also an R is placed in state 5 under $\bar{Y}_2$ so that the Y_2 MEMORY is reset in the state which precedes state 6.

d) More of the values of the secondary variables are now entered in column V. Since the Y_1 MEMORY generator is SET in states 3 and 6, the secondary variable y_1 is "on" in those states, and "1's" are entered in column V. Similarly "1's" are entered for y_2 in states 1 and 4, and "0's" are entered for y_1 in state 2 and for y_2 in state 5. This leaves "don't care" situations for y_1 in state 5 and for y_2 in state 3. (The "don't cares" are later changed to "1's" or "0's" in order to make the output and secondary equations as simple as possible.

e) The output equations are now written. The simplest combination is sought for the total signal $(x_1x_2y_1y_2)$ that makes the output Z_1 "on" in states 1, 2, 3, 4 and "off" in states 5 and 6. Since there are relatively few states involved, this is done by inspection. It is noted that if the "don't care" of y_2 in state 3 is made a "1", then the values of Z_1 correspond exactly to the values of y_2. Thus the first output equation is

$$Z_1 = y_2 \tag{B.2}$$

If the "don't care" for y_1 in state 5 is also changed to "1", it is seen that there are two equally simple equations to make Z_2 "on" in states 3 and 4. They are:

$$Z_2 = \bar{x}_1 x_2 \dot{+} \bar{x}_1 \bar{y}_1 \tag{B.3}$$

or

$$Z_2 = x_2 y_2 \dot{+} \bar{x}_1 \bar{y}_1 \tag{B.4}$$

Equation (B.3) can be written as

$$Z_2 = \bar{x}_1 (x_2 \dot{+} \bar{y}_1) \tag{B.5}$$

This form of the equation is advantageous since it can be implemented with one OR-NOT generator and one AND-NOT generator. In the form of Equation (B.3) three logic generators are needed: one AND-NOT, one NOR and one OR.

f) At this step the rest of the entries are made in column VI. The information entered is for the purpose of writing the SET and RESET equations. The SET of the Y_1 MEMORY cannot be "on" in states 2 and 4 since y_1 is "off". Thus "0's" are entered in states 2 and 4 under Y_1. In states 1 and 5 the SET signal for the Y_1 MEMORY could be "on", but since it doesn't have to be, "don't cares" are entered. Under Y_1, "0's" are entered where the RESET signal cannot be "on". "Don't cares" are entered where the RESET may be "on". Similar information is entered under Y_2 and $\bar{Y}_2$ for the SET and RESET of the Y_2 MEMORY.

g) Finally the SET and RESET equations are written giving the secondary equations. For the SET and RESET of the Y_1 MEMORY the signal must be limited to the variables x_1, x_2, and y_2. It is observed that the Y_1 MEMORY can be set in state 5 instead of state 6 since state 5 has a "don't care" and since state 6 always follows state 5. The SET equation is now very simple and is

$$S_1 = x_2 \tag{B.6}$$

The simplest RESET signal for the Y_1 MEMORY is seen to be

$$R_1 = x_1 y_2 \tag{B.7}$$

As will be explained in section B.2 it becomes necessary to use the slightly more complicated signal

$$R_1 = x_1 \bar{x}_2 y_2 \tag{B.8}$$

The SET and RESET signals for the Y_2 MEMORY are limited to the x_1, x_2 and y_1 variables. The simplest combination to give the SET signal is

$$S_2 = \bar{x}_1 \tag{B.9}$$

This makes the SET be "on" in state 3 as well as in the required states 1 and 4. There are two equally simple combinations to give the RESET signal. They are:

$$R_2 = x_1 x_2 \tag{B.10}$$

$$R_2 = x_1 y_1 \tag{B.11}$$

Equation (B.11) would make the RESET be on in state 6 as well as state 5. Equation (B.10) is selected.

The two output and two secondary equations are shown below Table IX. The SET and RESET signals are also shown in tabular form.

B.2 Time Delays Caused by Amplifiers

In Table IX it can be seen that as the system goes from state 2 to state 5, the RESET signal of the Y_1 MEMORY generator goes "off" and the SET signal goes "on". The RESET signal goes "off" because the input signal x_2 goes "on". The SET signal goes "on" because x_2 goes "on". By using a passive AND-NOT generator (6), the x_2 input is in effect linked directly to both the SET and the RESET of the Y_1 MEMORY generators. Thus the SET goes "on" and the RESET goes "off" simultaneously. If an active generator (6) is used, the x_2 signal would still be linked directly to the SET of the MEMORY generator, but the x_2 connection to the RESET would have a time delay in the displacement. The time delay would be caused by the AMPLIFIER of the AND-NOT generator (6). This situation would mean that the SET signal of the Y_1 MEMORY would try to go "on" before the RESET signal would go "off". Because of the physical construction of the MEMORY generator the SET and RESET signals cannot both be "on" at the same time, and an active AND-NOT generator (6) is not usable.

The need for the $\bar{x}_2$ variable in the RESET equation $R_1 = x_1 \cdot \bar{x}_2 \cdot y_2$ can also be understood from this discussion. If the equation were simply $R_1 = x_1 \cdot y_2$, the RESET would be turned "off" (in going from state 2 to state 5) by y_2 going "off". But y_2 is delayed in going "off" by the two AMPLIFIERS of the generators (1) and (2). Thus as x_2 would go "on" the SET would try to go "on" while the RESET would still be "on". Again the SET and RESET signal cannot be "on" at the same time. The conclusion is that the $\bar{x}_2$ is needed in the equation, and it makes the RESET go "off" simultaneously with the SET going "on".

APPENDIX C

ENUMERATION OF KINEMATIC CHAINS FOR SEPARATOR LINKAGES

In this appendix the enumeration of kinematic chains for use as possible separator linkages is explained. Steps described by Haas and Crossley [32] and the method of Franke's condensed notation as described by Davies and Crossley [34] are used. The eight-link three-degree-of-freedom chains and the five-link two-degree-of-freedom chains are enumerated.

C.1 Eight-Link Chains

The Chebyshev-Grubler formula for the degrees-of-freedom (mobility) M of a planar linkage is

$$M = 3(N-1) - 2J \tag{C.1}$$

where N is the number of links and J is the number of revolute pairs. (It should be noted that in reality a kinematic chain, which has no fixed link, has a different number of degrees-of-freedom than a linkage, which has a fixed link. However, in this study, kinematic chains are physically used only when they become linkages; thus it is convenient to consider the degrees-of-freedom of a chain to be the same as the degrees-of-freedom of a linkage derived from the chain.)

Equation (C.1) is solved for 2J and becomes

$$2J = 3(N-1) - M \tag{C.2}$$

Since J is the number of revolute pairs, 2J represents the number of "half revolute pairs" or "ends" that exist on the N links of the chain. To illustrate, in Figure 38 the chain has six revolute pairs and twelve ends. Each binary link has two ends and each ternary link has three ends. Each quaternary link has four ends, etc.

For M = 3 and N = 8 Equation (C.2) is solved for the number of ends 2J and gives

$$2J = 3(8-1) - 3 = 18$$

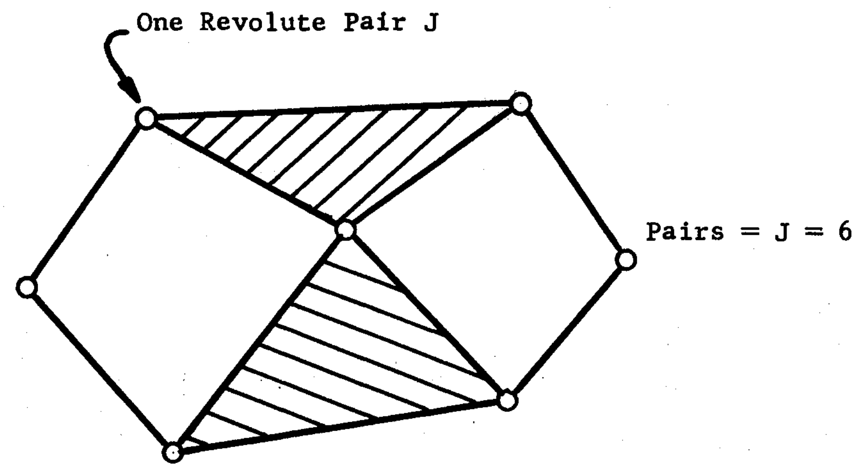
The enumeration of all the desired chains is now made by finding all the ways in which 18 ends can be distributed on 8 links, and then finding all ways in which the 8 links can be assembled to form chains.

Table X shows the two possible distributions, ① and ②, of the 18 ends. In distribution ①, seven links have two ends and are binary, and the eighth link has four ends and is quaternary. In distribution ② there are six binary links and two ternary links.

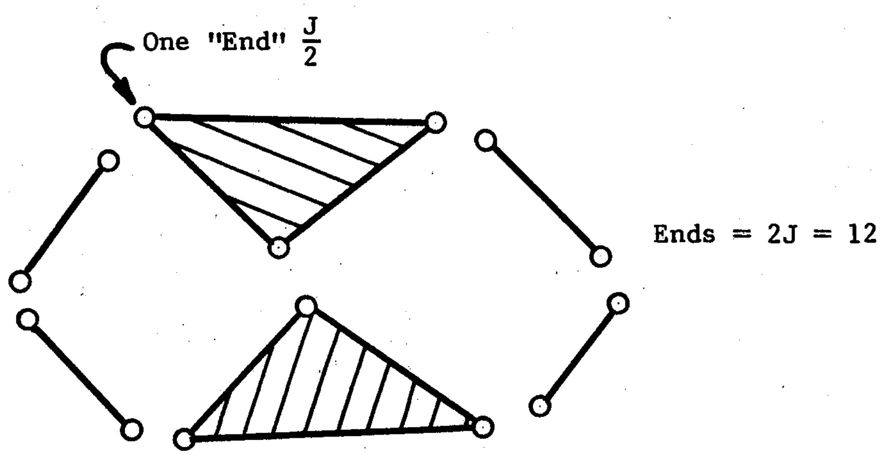
Franke's condensed notation [34] is employed to find the distinct ways in which the two distributions can be assembled to form kinematic chains. The symbolic notation of Franke uses a "molecule" to represent the kinematic chain. The molecules and the respective kinematic chains are shown in Figure 39. The molecules are constructed by using two symbols:

1. A circle to represent a link with more than two ends (a polygonal link) and the circle has a number within indicating the number of ends.

2. A line to represent a direct connection between polygonal links, a single binary link between polygonal links or a string of binary links between polygonal links. Each line has an associated number which



(a) Chain With Six Revolute Pairs



(b) Links of Chain of (a) Showing Twelve "Ends"

Figure 38. Revolute Pairs and Link "Ends" of a Six-Link Kinematic Chain

TABLE X

DISTRIBUTION OF EIGHTEEN ENDS ON EIGHT KINEMATIC LINKS

Distribution ⇓	Links 1	2	3	4	5	6	7	8
1	2	2	2	2	2	2	2	4
2	2	2	2	2	2	2	3	3
	Ends							

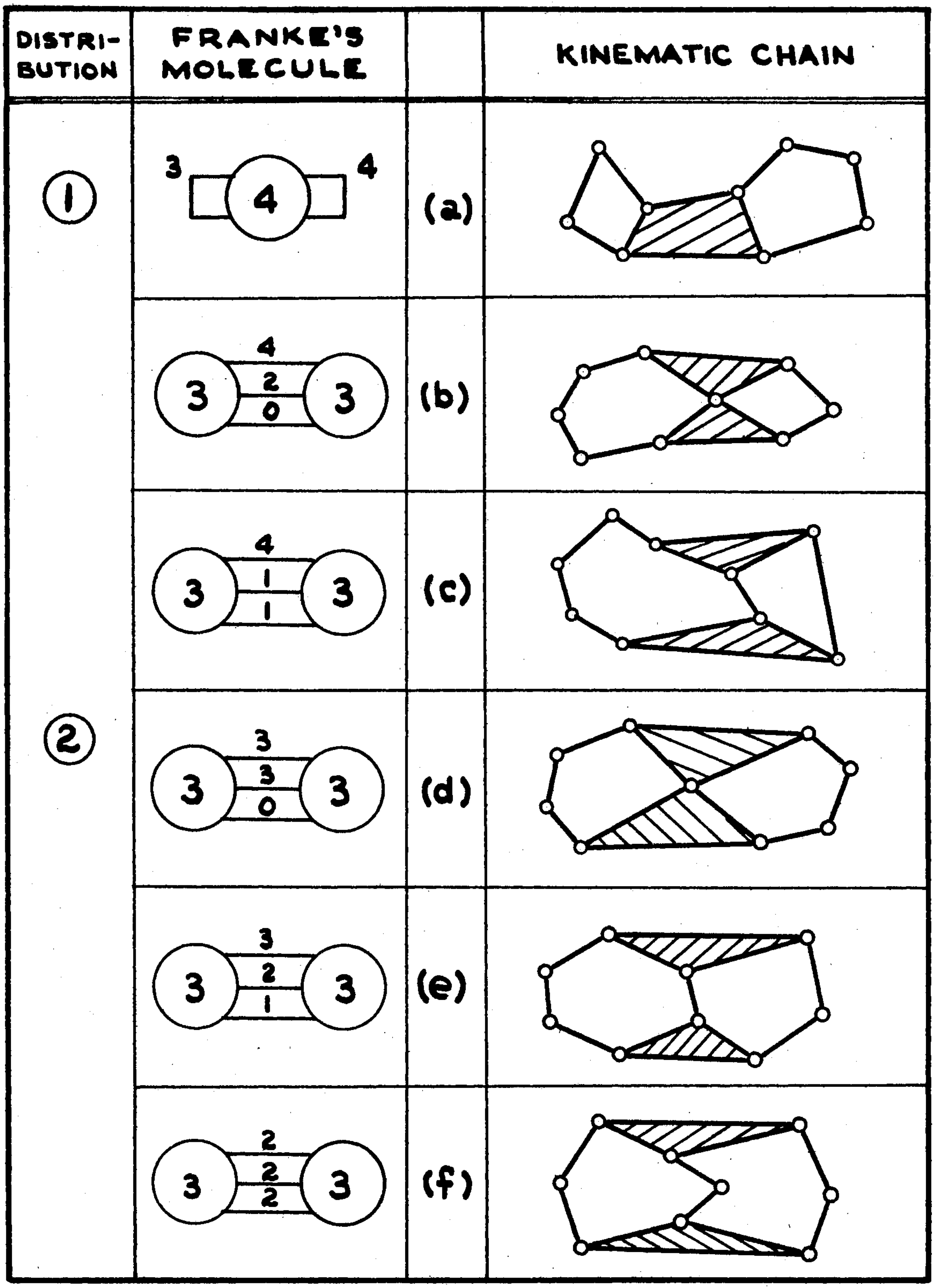


Figure 39. Enumeration of Kinematic Chains With
M = 3 and N = 8

indicates the number of binary links in the string. A zero means a direct connection. The number of lines leading to a circle must be the same as the number within the circle.

In Figure 39, for distribution (2), there are two ternary links and thus two circles with the number 3 within. The circles are joined with three lines. On the three lines are placed different combinations of numbers whose sum is six to represent the six binary links of the chain. It should be noted that the combinations cannot include two zeros or a one and a zero since these combinations eliminate the mobility between the polygonal links. Figure 40 illustrates this point.

For distribution (1) there is only one way for the seven binary links to be attached to the quaternary link and have mobility of all of the links. This arrangement is shown in Figure 39(a).

C.2 Five-Link Chain

To determine the five-link two-degree-of-freedom chain, $M = 2$ and $N = 5$. Substitution in Equation (C.2) gives the number of ends as

$$2J = 3(5-1) - 2 = 10$$

The ten ends are distributed on the five links in only one way as shown in Table XI. Since all five links are binary, the only kinematic chain which results is a single loop of binary links as shown in Figure 41.

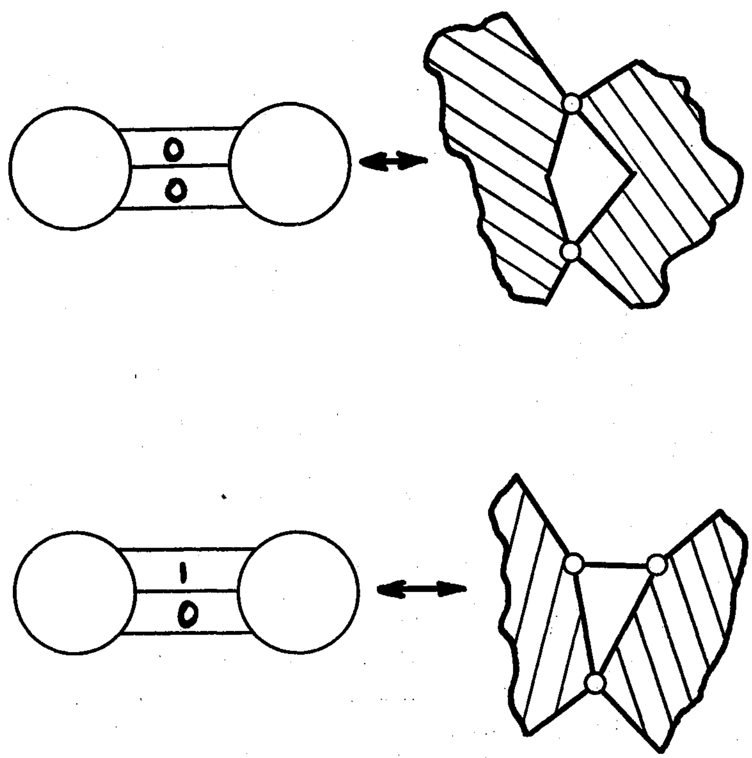


Figure 40. Immobile Linkage Connections
(Structures)

TABLE XI

DISTRIBUTION OF TEN ENDS ON FIVE KINEMATIC LINKS

Distribution	Links				
⇓	1	2	3	4	5
1	2	2	2	2	2
	Ends				

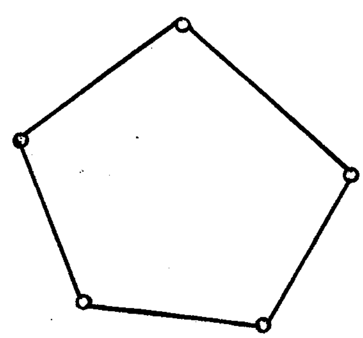


Figure 41. Five-Link Two-Degree-of-Freedom Kinematic Chain

APPENDIX D

SYNTHESIS AND ANALYSIS PROCEDURES

FOR SEPARATOR LINKAGES

This appendix contains the calculation of the value of the entity s_I as defined in section 3.5.2. It also explains the application of the computer to the synthesis of the separators and to the output angle analysis of the separators. The third part of this appendix shows a method for monitoring transmission angle values. The method is applied to an analysis of all of the transmission angles of the separator linkages.

A computer Program A (Table XII) is included at the end of this appendix.

D.1 Determination of s_I

The value of s_I as defined in section 3.5.2 is shown in Figure 42. The minimum transmission angle μ_m is shown for the joint D. (In this study the transmission angle is defined as the acute angle between the coupler link and the output; if necessary one link is extended for the acute angle.)

From Figure 42 the following equations are written:

$$a = \ell \sin \alpha \tag{D.1}$$

$$s_I = \ell \cos (\mu_m + \alpha) \tag{D.2}$$

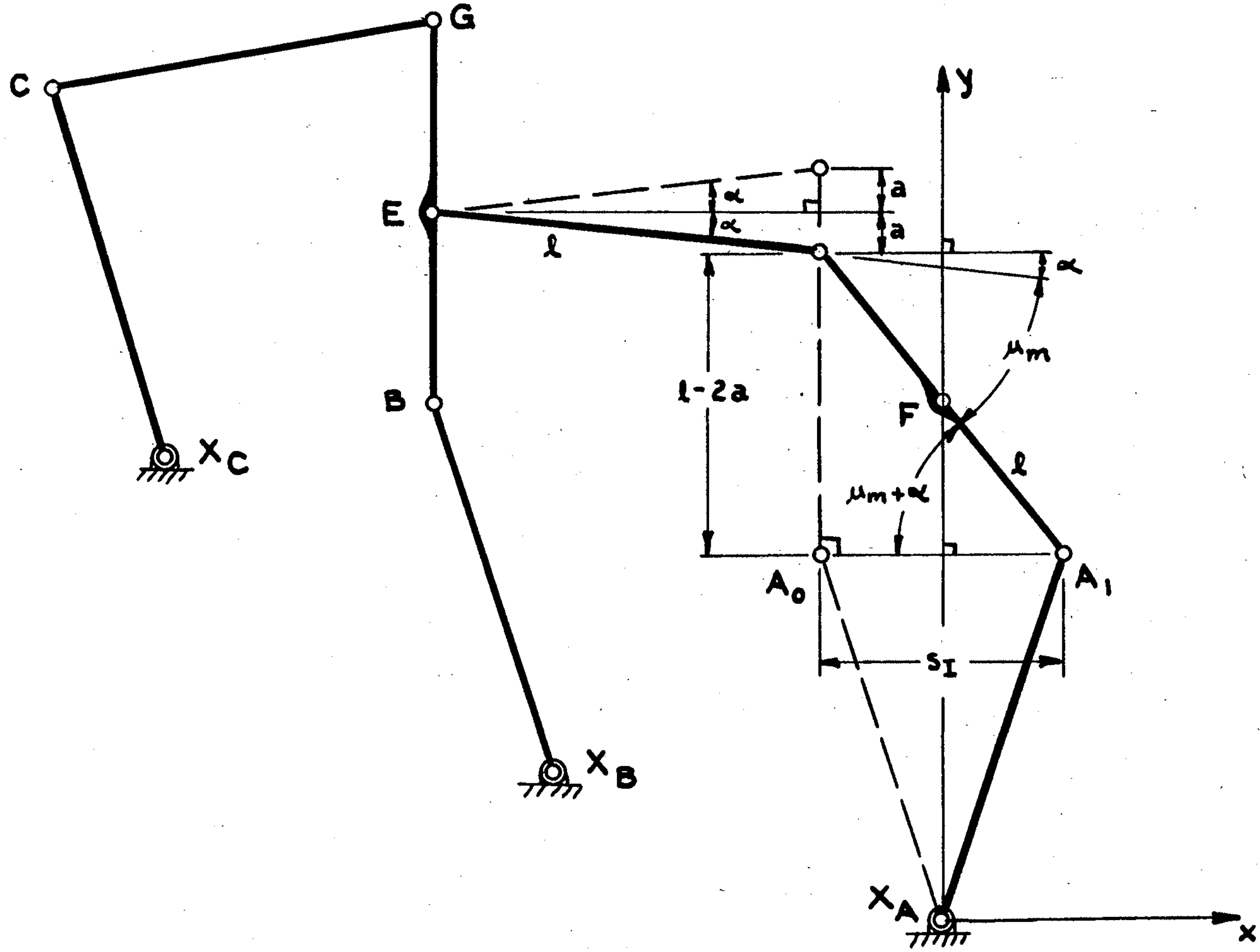


Figure 42. Determination of s_I

$$(s_I)^2 + (\ell - 2a)^2 = \ell^2 \tag{D.3}$$

When a and s_I are eliminated from Equation (D.3) and the substitutions

$$\cos\alpha = \frac{1 - \tan^2\frac{\alpha}{2}}{1 + \tan^2\frac{\alpha}{2}}$$

and

$$\sin\alpha = \frac{2\tan\frac{\alpha}{2}}{1 + \tan^2\frac{\alpha}{2}}$$

are made, Equation (D.3) becomes

$$C_1 \tan^4\frac{\alpha}{2} + C_5 \tan^3\frac{\alpha}{3} + C_6 \tan^2\frac{\alpha}{2} + C_7 \tan\frac{\alpha}{2} + C_1 = 0 \tag{D.4}$$

where $C_1 = \cos^2\mu_m$

$C_2 = -4$

$C_3 = 4 + \sin^2\mu_m - \cos^2\mu_m$

$C_4 = 2\cos\mu_m \sin\mu_m$

$C_5 = 2C_2 + 2C_4$

$C_6 = 2C_1 + 4C_3$

$C_7 = 2C_2 - 2C_4.$ (D.5)

The roots of Equation (D.4) are found with the computers using the SSP subroutine POLRT. The value of α corresponding to Figure 42 is found to be 6.302212 degrees, where the value of μ_m in the coefficients of Equation (D.5) is 45 degrees. The value of S_I is found using Equation (D.2) in the computer Program A, Table XII, at the end of this appendix.

D.2 Dimensional Synthesis and Output Analysis

In this section the calculations are made for the coordinates of the fixed centers X_A, X_A', X_B, X_B', X_C and X_C' of the separator linkages. The angles $\Delta\theta$ through which the inputs rotate as they change from the "0" to the "1" position is also calculated. The calculations are based on the link lengths selected in section 3.5.2 and the value of s_I determined in section D.1.

An analysis of the positions of the output point F for each of the separators is needed for the synthesis of the threshold linkages. The analysis is combined with the determination of the coordinates of the fixed centers X_A, . . . X_C' in the computer Program A (Table XII) shown at the end of this section.

D.2.1 Input Rotation $\Delta\theta$

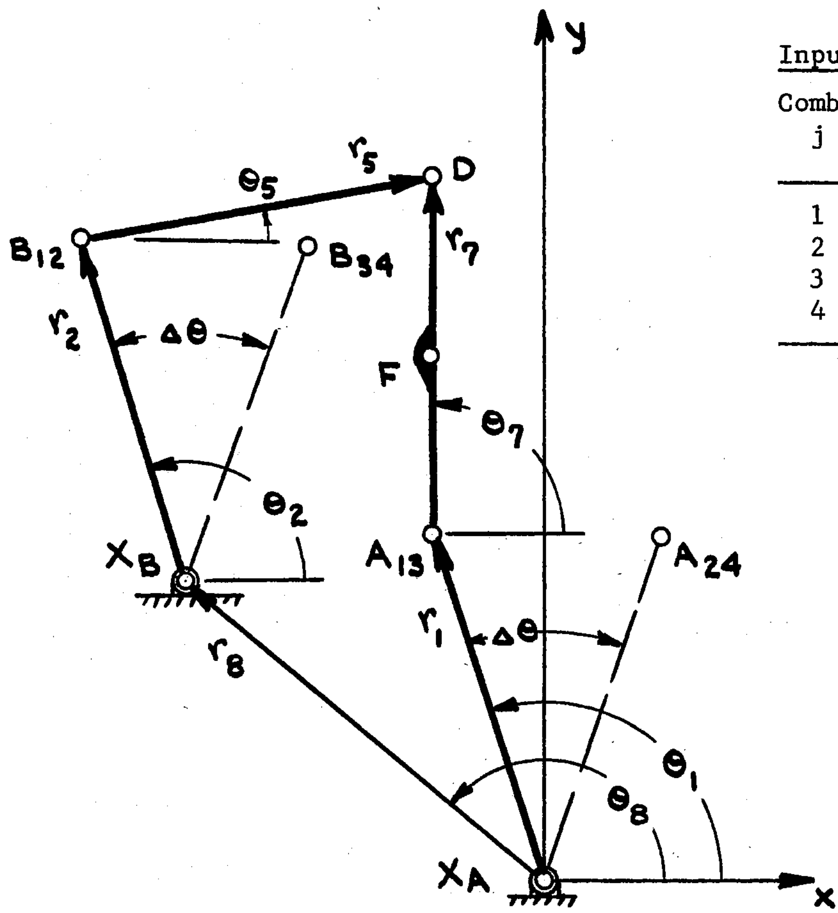
The separator linkages are shown in Figures 43 and 44 in terms of the parameters of the synthesis and the analysis. The input crank rotations all have the same value since all of the cranks have the same length ℓ and since the chord intercepted by each rotation is s_I. The value of $\Delta\theta$ is calculated in computer Program A and is given by

$$\Delta\theta = 2 \sin^{-1} \left(\frac{s_I}{2\ell}\right) \tag{D.6}$$

Since the input cranks X_AA, X_BB and X_CC have their "0" and "1" positions symmetrically located about a vertical line, the binary values of θ_1, θ_2 and θ_3 are

$$\theta_{"0"} = 90^\circ + \frac{\Delta\theta}{2}$$

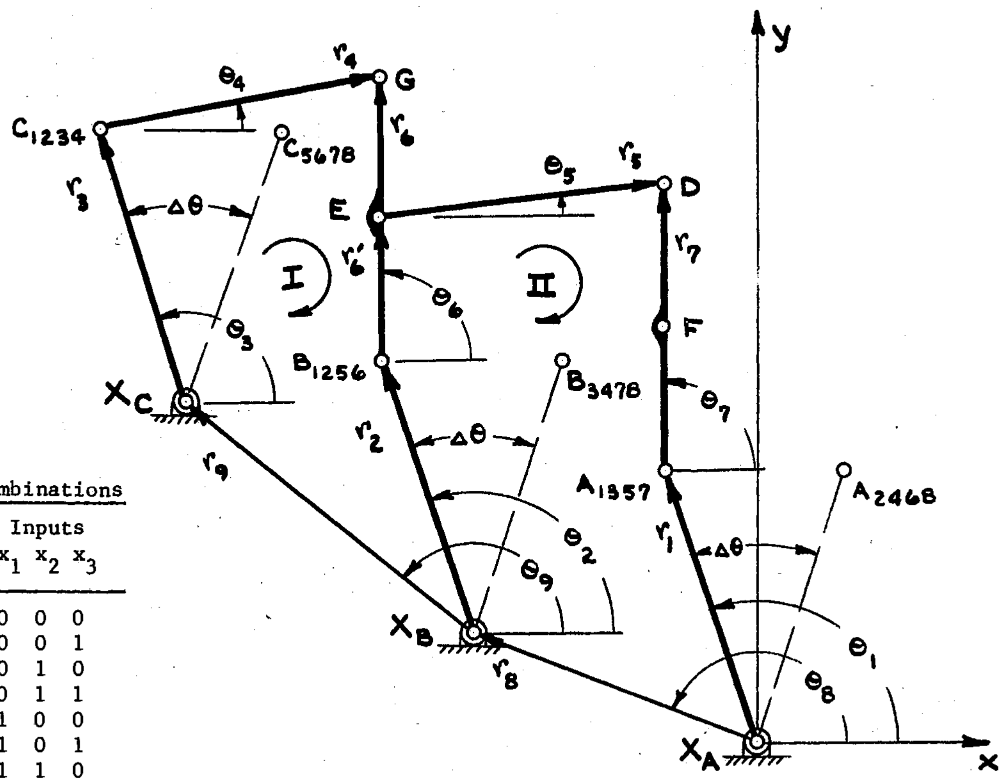
and



Input Combinations

Comb. j	Inputs x_1	x_2
1	0	0
2	0	1
3	1	0
4	1	1

Figure 43. Synthesis and Output Analysis of Separator of $\overline{W} = (1,1,0)$



Input Combinations

Comb. j	Inputs x_1	x_2	x_3
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

Figure 44. Synthesis and Output Analysis of Separators for $\overline{W} = (1,1,1)$, $(2,1,1)$, $(2,2,1)$ and $(3,2,1)$

$$\theta_{"1"} = 90^\circ - \frac{\Delta\theta}{2} \tag{D.7}$$

D.2.2. Coordinates of Fixed Centers

The coordinates of the fixed centers X_A, . . . X_C' are based on the coordinate system shown in Figures 43 and 44. In the computer Program A the coordinates are found by repeatedly applying the loop model of Figure 45(a) to the loops of each of the separators shown in Figures 43 and 44. In each application of the loop model, the vector of d_6 is the unknown. The components of the vector of d_6 give the desired coordinates.

A subroutine SYNSIX of Program A applies the model to any loop of the separator linkages. The subroutine is based on the following analysis of Figure 45(a).

Writing the loop equation and separating it into its real and imaginary parts gives the x and y components of d_6:

$$x_6 = d_6\cos\phi_6 = d_3\cos\phi_3 + d_4\cos\phi_4 - d_1\cos\phi_1 - d_2\cos\phi_2 - d_5\cos\phi_5$$

and

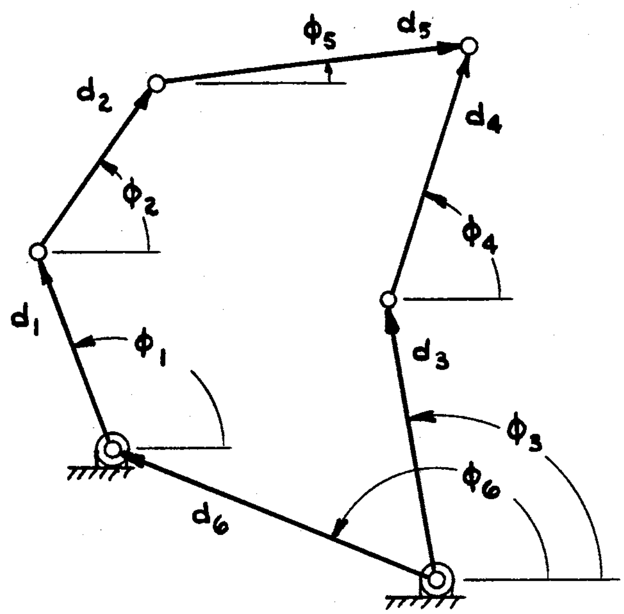
$$y_6 = d_6\sin\phi_6 = d_3\sin\phi_3 + d_4\sin\phi_4 - d_1\sin\phi_1 - d_2\sin\phi_2 - d_5\sin\phi_5 \tag{D.8}$$

Now

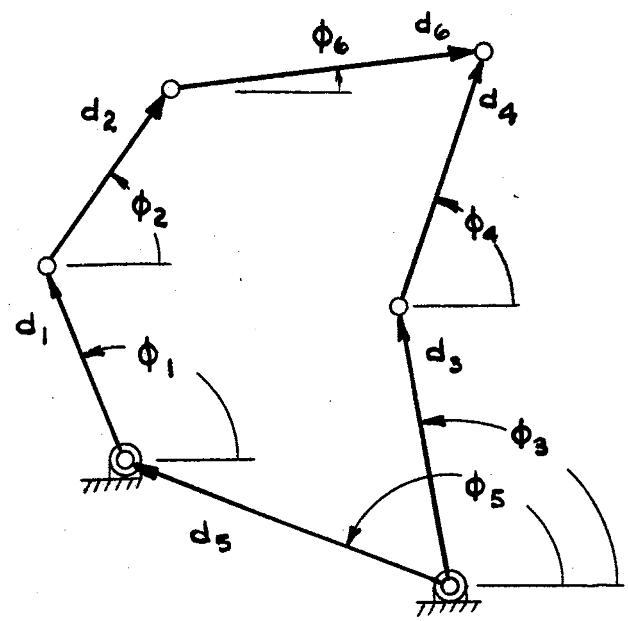
$$d_6 = \sqrt{x_6{}^2 + y_6{}^2} \tag{D.9}$$

In order to determine the correct quadrant as well as the magnitude of ϕ_6, the sine and cosine of ϕ_6 are found:

$$\sin\phi_6 = \frac{y_6}{d_6}$$



(a) Synthesis: d_6 and ϕ_6



(b) Analysis: ϕ_4

Figure 45. Six-Link Loop Models

and

$$\cos\ \phi_6 = \frac{x_6}{d_6} \tag{D.10}$$

Finally

$$\phi_6 = \tan^{-1}\frac{\sin\ \phi_6}{\cos\ \phi_6} \tag{D.11}$$

When the loop model of Figure 45(a) is applied to the five-bar separator (Figure 43), the value of d_2 is made zero. The value of d_2 is also made zero when applying the loop model to loop I of Figure 44.

Since the values for $\Delta\theta$, $\theta_{"0"}$, $\theta_{"1"}$ and the coordinates of the fixed centers are given in Table V, no computer output is shown with Program A.

D.2.3 Output Analysis θ_7

Since the positions of the output point F, needed for the synthesis of the threshold linkages, are dependent upon the angle θ_7 (Figures 43 and 44), an output analysis is needed. To perform the output analysis a loop model as shown in Figure 45(b) is repeatedly applied to the loops of the separators.

A subroutine ANLSIX, of computer Program A, applies the loop model to any loop of the separators. The subroutine is based on the following analysis of Figure 45(b).

The real and complex parts of the loop equation give

$$\cos\phi_{6j} = E_4\cos\phi_{4j} + E_3\cos\phi_{3j} - E_5\cos\phi_5 - E_2\cos\phi_{2j} - E_1\cos\phi_{1j}$$

and

$$\sin\phi_{6j} = E_4\sin\phi_{4j} + E_3\sin\phi_{3j} - E_5\sin\phi_5 - E_2\sin\phi_{2j} - E_1\sin\phi_{1j} \tag{D.12}$$

where

$$E_i = \frac{d_i}{d_6} ; \quad i = 1, \ldots , 5 \tag{D.13}$$

and $j = 1, 2, \ldots , n$; $n = 4$ or 8

The values of j represent the input combinations as shown in the tables of Figures 43 and 44. For the separator of Figure 43, the number of combinations n is four. For the separators of Figure 44, n is eight. (In the figures the input joints A, B and C are subscripted with the numbers j. The subscripting is for convenience in recognizing the positions of the input cranks for each input combination.)

From Equation (D.12)

$$\cos\phi_{6j} = E_4\cos\phi_{4j} + CO_j$$

and

$$\sin\phi_{6j} = E_4\sin\phi_{4j} + SI_j \tag{D.14}$$

where

$$CO_j = C_{3j} - C_5 - C_{2j} - C_{1j}$$

$$SI_j = S_{3j} - S_5 - S_{2j} - C_{1j}$$

and

$$\left.\begin{aligned} C_{ij} &= E_i\cos\phi_{ij} \\ S_{ij} &= E_i\sin\phi_{ij} \end{aligned}\right. \quad i = 1, 2, 3 \tag{D.15}$$

The angle ϕ_6 is eliminated from Equation (D.14) by squaring and adding. The result gives

$$CO_j\cos\phi_{4j} + SI_j\sin\phi_{4j} = F_j \tag{D.16}$$

where

$$F_j = \frac{1 - (E_4)^2 - (CO_j)^2 - (SI_j)^2}{2E_4} \tag{D.17}$$

The substitution in Equation (D.16) of

$$\cos\phi_{4j} = \frac{1 - T^2\gamma_j}{1 + T^2\gamma_j}$$

and

$$\sin\phi_{4j} = \frac{2T\gamma_j}{1 + T^2\gamma_j} \tag{D.18}$$

where

$$T\gamma_j = \tan\frac{\phi_{4j}}{2} \tag{D.19}$$

gives

$$T^2\gamma_j - G_{1j}T\gamma_j + G_{2j} = 0 \tag{D.20}$$

where

$$G_{1j} = \frac{2SI_j}{GD_j}$$

$$G_{2j} = \frac{F_j - CO_j}{GD_j}$$

and

$$GD_j = F_j + CO_j \tag{D.21}$$

The values of $T\gamma_j$ are found by solving Equation (D.20):

$$T\gamma_j = \frac{G_{1j} \pm H_j}{2} \tag{D.22}$$

where

$$H_j = \sqrt{(G_{1j})^2 - 4G_{2j}} \tag{D.23}$$

Finally, by using the Equation (D.18), the values of $\emptyset_{4j}$ are given as

$$\emptyset_{4j} = \tan^{-1}\left(\frac{\sin\emptyset_{4j}}{\cos\emptyset_{4j}}\right) \qquad \text{(D.24)}$$

There are two values of $T\gamma_j$ found from Equation (D.22) for each value of j. Consequently there are two values of $\emptyset_{4j}$ calculated from Equation (D.24). In the Program A, the unwanted values of $\emptyset_{4j}$ are rejected by rejecting all values of $\emptyset_{4j}$ outside a known range of $\emptyset_{4j}$.

As an example, the computer output values of θ_7 are given at the end of Program A for the separator whose weight vector is $\overline{w} = (2, 1, 1)$.

D.3 Transmission Angle Analysis

D.3.1 Graphical Monitoring Concept

A graphical concept can be employed to monitor transmission angles. The concept is explained through the use of Figure 46. The point A is the input to the linkage and follows some prescribed path. Link AB is the coupler and BB_0 is the output. The transmission angle μ is the acute angle between the directions of the coupler link and the output link.

The figure is constructed by allowing $A\,B\,B_0$ to assume any position. From B, link AB is drawn at the angle $\mu = 90^\circ$ which locates A_{90}. A circular arc (labelled 90°) is drawn with B_0 as center and a radius of B_0A_{90}. Now, if the linkage $A\ B\ B_0$ were to move so that A followed the 90° arc, the transmission angle would remain constant at 90°. In a similar way two arcs are drawn for a constant $\mu = 60^\circ$ and two for a constant 30°.

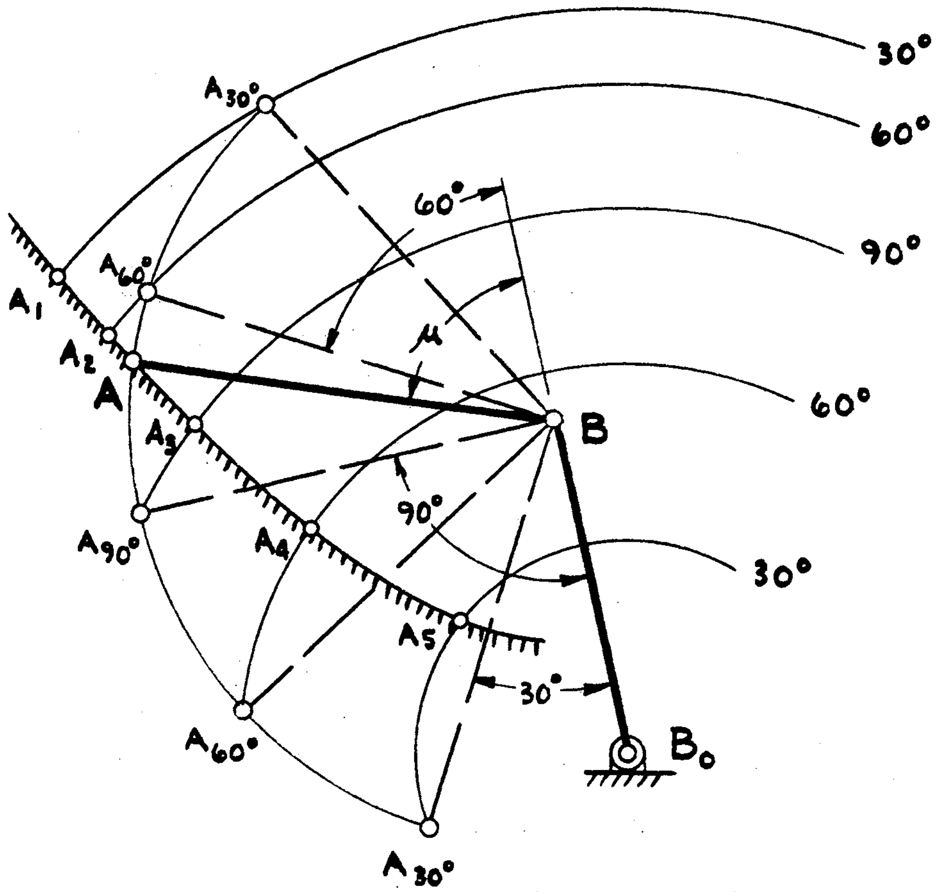


Figure 46. Graphical Monitoring of Transmission Angle

Finally, the transmission angle μ can be easily monitored as the input A travels along its prescribed path. When A is at A_1, A_2, A_3, etc., then μ has the values of 30°, 60°, 90°, etc., respectively. Additional arcs of constant μ can be added if a closer monitoring of μ is desired.

D.3.2. Transmission Angle Analysis for Separators

The graphical monitoring process of the previous section is utilized to analyze all the transmission angles of the separators. Observation of the dimensions of Figure 33 indicates that three analyses are sufficient to cover the analysis of all the transmission angles:

1. The single loop of Figure 33(a) and the loop $X_BX_CCGBX_B$ of Figure 33(b) through (e) are identical. The transmission angle μ of Figure 33(a) is at joint D. In Figure 33(b) through (e), μ is at joint G. Thus the analysis of μ shown in Figure 47(a) and (b) covers this group of transmission angles.

At (a) of Figure 47 the input is B and the output is either DA_0 or DA_1. (The link X_AA is considered to be fixed in either of the two positions X_AA_0 or X_AA_1 at any time when B is moving.) Consequently there are two sets of constant μ curves as shown. Examination of the figure reveals that as B moves from B_0 to B_1 with A at A_0, μ decreases from about 83° to about 55°. As B moves from B_0 to B_1 with A at A_1, μ increases from 45° to 90° and then decreases to about 83°.

At (b) of the same figure, the input is A and the output is either DB_0 or DB_1 depending on whether X_BB is considered fixed in position X_BB_0 or X_BB_1. The analysis of the transmission angle is similar to that of the figure at (a).

2. The loop $X_A X_B BEDAX_A$ of Figure 33(b), (c) and (d) are identical, with the transmission angle at D. For the case where E is the input and link DA is the output, it can be shown that the analysis of μ is identical to the analysis of Figure 47(a) except that the paths of E differ from the paths of B. The paths differ, however, only between the ends B_0 and B_1; thus the range of μ is the same as in Figure 47(a).

For the case where A is the input and DE is the output, E can be fixed in any of three positions. Consequently, there are three sets of curves of constant μ for the analysis. Figures for this analysis are not shown, but μ remains 45° or greater for all possible positions of the links.

3. The loop $X_A X_B BEDAX_A$ of Figure 33(e) is distinct from all others. For the case where E is the input and DA is the output, there are two sets of curves of constant μ since A can be fixed in either of the two positions A_0 or A_1.

For the case where A is the input and DE is the output, there are four positions in which E can be fixed. In this case there are four sets of curves of constant μ in the analysis. This analysis (figures are not included) shows that μ is not less than 45° in all possible positions of the linkage.

From the complete analysis as described, it can be seen that the minimum transmission angle for all the separators occurs in situations when a pair of links are in the position shown for the links ED and DA_1 of Figure 42. In that position the transmission angle is 45°, and hence it is verified that the minimum value of μ for the separator linkages is 45°.

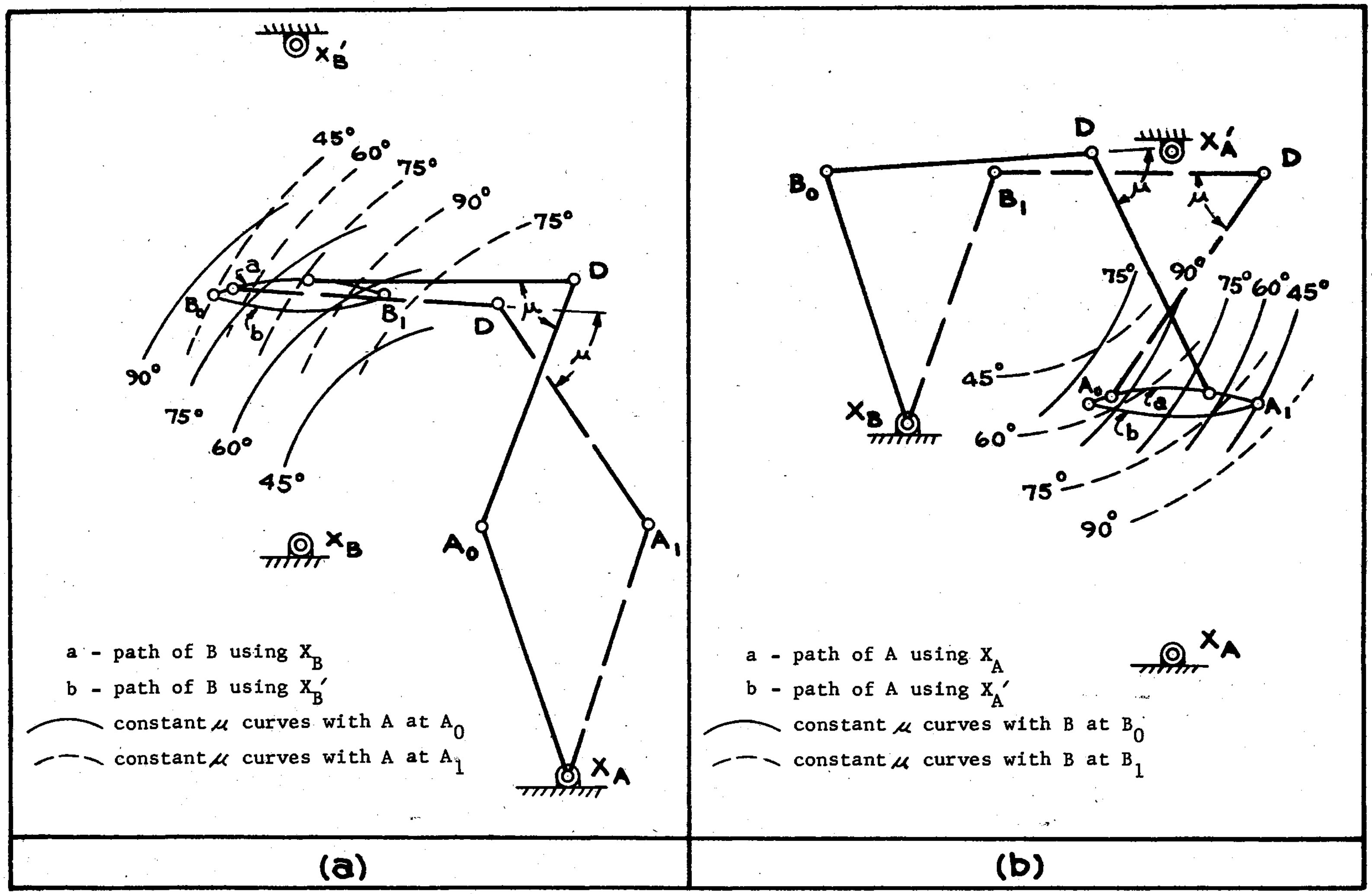


Figure 47. Analysis of Separator Linkage Transmission Angles

TABLE XII

COMPUTER PROGRAM A

```
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     *                                                                         *
C     *                                                                         *
C     *                                                                         *
C     *        PURPOSE                                                          *
C     *                ANALYSIS AND SYNTHESIS OF SEPARATOR LINKAGES             *
C     *                                                                         *
C     *        DESCRIPTION OF PARAMETERS                                        *
C     *                VL   =LENGTH OF BASIC LINK                               *
C     *                DMU  =MINIMUM VALUE OF TRANSMISSION ANGLES               *
C     *                DALP=INITIAL VALUE OF THETA 4                            *
C     *                                                                         *
C     *        INPUT DATA                                                       *
C     *                NEEDS ONE CARD FOR  VL,DMU,DALP  WITH FORMAT(3F10.0)     *
C     *                                                                         *
C     *        SUBROUTINES REQUIRED                                             *
C     *                SUBROUTINE  ANLSIX                                       *
C     *                SUBROUTINE  SYNSIX                                       *
C     *                                                                         *
C     *                                                                         *
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION T1(8),T2(8),T3(8),T4(8),T5(8),T6(8),T7(8),                SPRTR001
     1T10(8),T11(8),T6D(8),T7D(8),T11D(8)                                 SPRTR002
      DIMENSION D(6),P(6),AF(5)                                           SPRTR003
    8 FORMAT (8F10.6)                                                     SPRTR004
   42 FORMAT (1H1,10X,'SAMPLE OF OUTPUT ANGLE THETA 7')                   SPRTR005
   44 FORMAT (18X,'FOR LINKAGE OF')                                       SPRTR006
   46 FORMAT (17X,'W;T=(2,1,1;A,C)')                                      SPRTR007
   48 FORMAT (/10X,'INPUT COMBINATION      THETA 7')                      SPRTR008
   62 FORMAT (1H1,10X,I2//(/10X,F11.7))                                   SPRTR009
   64 FORMAT (/18X,I2,9X,F10.6)                                           SPRTR010
      READ(5,8) VL,DMU,DALP                                               SPRTR011
      PI=3.1415926535                                                     SPRTR012
      DEG=180.0/PI                                                        SPRTR013
      RAD=PI/180.0                                                        SPRTR014
      RMU=DMU*RAD                                                         SPRTR015
      ALP=DALP*RAD                                                        SPRTR016
      S=VL*COS(RMU+ALP)                                                   SPRTR017
C                                                                         SPRTR000
C     INPUT ANGLE CALCULATIONS                                            SPRTR000
C                                                                         SPRTR000
      DT2=ARSIN(S/(VL*2.0))                                               SPRTR018
      DTD2=DT2*DEG                                                        SPRTR019
      DT=2.0*DTD2                                                         SPRTR020
      T0D=90.0+DTD2                                                       SPRTR021
      T1D=90.0-DTD2                                                       SPRTR022
      TOR=T0D*RAD                                                         SPRTR023
      T1R=T1D*RAD                                                         SPRTR024
      TR=90.0*RAD                                                         SPRTR025
      DO 11 J=1,7,2                                                       SPRTR026
      T1(J)=TOR                                                           SPRTR027
   11 CONTINUE                                                            SPRTR028
      DO 12 J=2,8,2                                                       SPRTR029
      T1(J)=T1R                                                           SPRTR030
   12 CONTINUE                                                            SPRTR031
      DO 21 J=1,2                                                         SPRTR032
      T2(J)=TOR                                                           SPRTR033
   21 CONTINUE                                                            SPRTR034
      DO 22 J=5,6                                                         SPRTR035
      T2(J)=TOR                                                           SPRTR036
   22 CONTINUE                                                            SPRTR037
      DO 23 J=3,4                                                         SPRTR038
      T2(J)=T1R                                                           SPRTR039
   23 CONTINUE                                                            SPRTR040
      DO 24 J=7,8                                                         SPRTR041
      T2(J)=T1R                                                           SPRTR042
   24 CONTINUE                                                            SPRTR043
      DO 31 J=1,4                                                         SPRTR044
      T3(J)=TOR                                                           SPRTR045
   31 CONTINUE                                                            SPRTR046
      DO 32 J=5,8                                                         SPRTR047
      T3(J)=T1R                                                           SPRTR048
   32 CONTINUE                                                            SPRTR049
      R1=VL                                                               SPRTR050
      R2=VL                                                               SPRTR051
      R5=VL                                                               SPRTR052
C                                                                         SPRTR000
C     SEPARATOR FOR W=(1,1,0)                                             SPRTR000
C                                                                         SPRTR000
      R7=VL                                                               SPRTR053
      T5(1)=ALP                                                           SPRTR054
      T7(1)=TR                                                            SPRTR055
      N=4                                                                 SPRTR056
      AK7=0.5                                                             SPRTR057
      D(1)=R2                                                             SPRTR058
      D(2)=0.0                                                            SPRTR059
      D(3)=R1                                                             SPRTR060
      D(4)=R7                                                             SPRTR061
      D(5)=R5                                                             SPRTR062
      P(1)=T2(1)                                                          SPRTR063
      P(2)=1.0                                                            SPRTR064
      P(3)=T1(1)                                                          SPRTR065
      P(4)=T7(1)                                                          SPRTR066
      P(5)=T5(1)                                                          SPRTR067
      CALL SYNSIX (D,P,R8,T8,X8,Y8)                                       SPRTR068
      T8D=T8*DEG                                                          SPRTR069
      D(5)=R8                                                             SPRTR070
      D(6)=R5                                                             SPRTR071
      CALL ANLSIX (N,D,T2,T1,T1,T8,T7)                                    SPRTR072
      DO 41 J=1,N                                                         SPRTR073
      T7D(J)=T7(J)*DEG                                                    SPRTR074
   41 CONTINUE                                                            SPRTR075
      XA=0.0                                                              SPRTR076
      YA=0.0                                                              SPRTR077
      XAP=0.0                                                             SPRTR078
      YAP=2.0*R2*SIN(T1R)                                                 SPRTR079
      XB=X8                                                               SPRTR080
      YB=Y8                                                               SPRTR081
      XBP=X8                                                              SPRTR082
      YBP=YB+YAP                                                          SPRTR083
      BD=R5                                                               SPRTR084
      K=1                                                                 SPRTR085
      AF(1)=0.5*R7                                                        SPRTR086
      XC=1000.0                                                           SPRTR087
      YC=1000.0                                                           SPRTR088
      XCP=1000.0                                                          SPRTR089
```

```
      YCP=1000.0
      R3=1000.0
      R6P=1000.0
      R6=1000.0
      R4=1000.0
      DE=1000.0
      WRITE(6,62)K,XA,YA,XAP,YAP,XB,YB,XBP,YBP,
     1XC,YC,XCP,YCP,R1,R2,R3,R7,DE,BD,
     2R6P,R6,R4,AF(K),DT,T0D,T1D
      WRITE(6,42)
      WRITE(6,44)
      WRITE(6,46)
      WRITE(6,48)
      DO 67 J=1,N
      WRITE(6,64)J,T7D(J)
   67 CONTINUE
C
C     SEPARATORS FOR  W=(1,1,1), (2,1,1), (2,2,1), (3,2,1)
C
      N=8
      R3=VL
      R4=VL
      R6=VL
      DE=R5
      BD=1000.0
      T4(1)=ALP
      DO 71 L=1,2
      IF (L.EQ.2) GO TO 51
      R6P=0.5*VL
      R7=VL
      AF(2)=(2.0/3.0)*R7
      AF(3)=0.5*R7
      AF(4)=(4.0/3.0)*R7
      T5(1)=ALP
      T6(1)=TR
      T7(1)=TR
      GO TO 53
   51 R6P=1.5*VL
      R7=1.5*VL
      AF(5)=0.5*R7
      T5(1)=0.0
      T6(1)=TR
      T7(1)=TR
   53 CONTINUE
C
C     LOOP 1
C
      D(1)=R3
      D(2)=0.0
      D(3)=R2
      D(4)=R6
      D(5)=R4
      P(1)=T3(1)
      P(2)=T3(1)
      P(3)=T2(1)
      P(4)=T6(1)
      P(5)=T4(1)
      CALL SYNSIX (D,P,R9,T9,X9,Y9)
```

```
      T9D=T9*DEG
      D(5)=R5
      D(6)=R4
      CALL ANLSIX (N,D,T3,T3,T2,T9,T6)
      DO 55 I=1,N
      T6D(I)=T6(I)*DEG
   55 CONTINUE
C
C     LOOP 2
C
      D(1)=R2
      D(2)=R6P
      D(3)=R1
      D(4)=R7
      D(5)=R5
      P(1)=T2(1)
      P(2)=T6(1)
      P(3)=T1(1)
      P(4)=T7(1)
      P(5)=T5(1)
      CALL SYNSIX (D,P,R8,T8,X8,Y8)
      T8D=T8*DEG
      D(5)=R8
      D(6)=R5
      CALL ANLSIX (N,D,T2,T6,T1,T8,T7)
      DO 61 I=1,N
      T7D(I)=T7(I)*DEG
   61 CONTINUE
      XA=0.0
      YA=0.0
      XAP=0.0
      YAP=2.0*R2*SIN(T1R)
      XB=X8
      YB=Y8
      XBP=XB
      YBP=YB+YAP
      XC=XB+X9
      YC=YB+Y9
      XCP=XC
      YCP=YC+YAP
      IF (L.EQ.2) GO TO 65
      DO 60 K=2,4
      WRITE(6,62)K,XA,YA,XAP,YAP,XB,YB,XBP,YBP,
     1XC,YC,XCP,YCP,R1,R2,R3,R7,DE,BD,
     2R6P,R6,R4,AF(K),DT,T0D,T1D
      WRITE(6,42)
      WRITE(6,44)
      WRITE(6,46)
      WRITE(6,48)
      DO 68 J=1,N
      WRITE(6,64)J,T7D(J)
   68 CONTINUE
   60 CONTINUE
      GO TO 71
   65 K=5
      WRITE(6,62)K,XA,YA,XAP,YAP,XB,YB,XBP,YBP,
     1XC,YC,XCP,YCP,R1,R2,R3,R7,DE,BD,
     2R6P,R6,R4,AF(K),DT,T0D,T1D
```

```
      WRITE(6,42)
      WRITE(6,44)
      WRITE(6,46)
      WRITE(6,48)
      DO 69 J=1,N
      WRITE(6,64)J,T7D(J)
   69 CONTINUE
   71 CONTINUE
      STOP
      END
```

SYNSIX

```
      SUBROUTINE SYNSIX (D,P,D6,P6,X6,Y6)
      DIMENSION D(5),P(5),S(5),C(5)
      DO 12 I=1,5
      S(I)=D(I)*SIN(P(I))
      C(I)=D(I)*COS(P(I))
   12 CONTINUE
      Y6=S(3)+S(4)-S(1)-S(2)-S(5)
      X6=C(3)+C(4)-C(1)-C(2)-C(5)
      D6=SQRT(X6*X6+Y6*Y6)
      SP6=Y6/D6
      CP6=X6/D6
      P6=ATAN2(SP6,CP6)
      RETURN
      END
```

ANLSIX

```
      SUBROUTINE ANLSIX (N,D,P1,P2,P3,P5,P4)
      DIMENSION D(6),P1(8),P2(8),P3(8),P4(8),E(5),
     1C(4,6),S(4,8),P(3,8)
      PI=3.1415926535
      RAD=PI/180.0
      DEG=180.0/PI
      DO 15 I=1,5
      E(I)=D(I)/D(6)
   15 CONTINUE
      DO 20 J=1,N
      P(1,J)=P1(J)
      P(2,J)=P2(J)
      P(3,J)=P3(J)
   20 CONTINUE
      DO 25 I=1,3
      DO 25 J=1,N
      C(I,J)=E(I)*COS(P(I,J))
      S(I,J)=E(I)*SIN(P(I,J))
   25 CONTINUE
      C5=E(5)*COS(P5)
      S5=E(5)*SIN(P5)
      DO 35 J=1,N
      CO=C(3,J)-C5-C(2,J)-C(1,J)
      SI=S(3,J)-S5-S(2,J)-S(1,J)
      F=(1-E(4)*E(4)-CO*CO-SI*SI)/(2.0*E(4))
      GD=F+CO
      G1=2.0*SI/GD
      G2=(F-CO)/GD
      GR=G1*G1-4.0*G2
      H=SQRT(GR)
      TP=(G1+H)/2.0
   27 SP=2.0*TP/(1.0+TP*TP)
      CP=(1.0-TP*TP)/(1.0+TP*TP)
      P4(J)=ATAN2(SP,CP)
      IF (P4(J).GT.0.73*PI) GO TO 28
      GO TO 30
   28 TP=(G1-H)/2.0
      GO TO 27
   30 CONTINUE
   35 CONTINUE
      RETURN
      END
```

SAMPLE OF OUTPUT ANGLE THETA 7
FOR LINKAGE OF
W;T=(2,1,1;A,C)

INPUT COMBINATION	THETA 7
1	89.999527
2	126.697586
3	72.320328
4	108.696930
5	72.320328
6	108.697037
7	51.302185
8	89.999527

APPENDIX E

SYNTHESIS AND ANALYSIS PROCEDURES

FOR THRESHOLD LINKAGES

E.1 Synthesis of Loop FJK

An example is presented to illustrate the synthesis of the loops FJK of the threshold linkages. The synthesis is for the linkage of $\overline{W}$ = (2, 1, 1) (see Table III) with two threshold angles τ_a and τ_c.

A graphical overlay procedure is used as follows to determine the lengths of links FJ and JK and to locate center K with respect to center X_A.

1. All paths of point F are drawn (Figure 48) including the five stable points of F. The paths and stable points are located by using the lengths of X_AA and AF as given in Table V and the values of θ_7 as given in the output of computer Program A (Table XII). (The analysis of the separator linkages in Appendix D shows that there actually are eight stable points of F. However, for graphical purposes some are so close together as to be considered the same point.)

2. Using overlays and trial and error, lengths for links FJ and JK are found. The threshold angles τ_a and τ_c are made relatively large while maintaining the lengths of FJ and JK not less than about $\ell/3$. (The value $\ell/3$ is selected as an approximate minimum for maintaining good link length ratios.) The resulting values of τ_a and τ_c are about 24° and 30°, respectively.

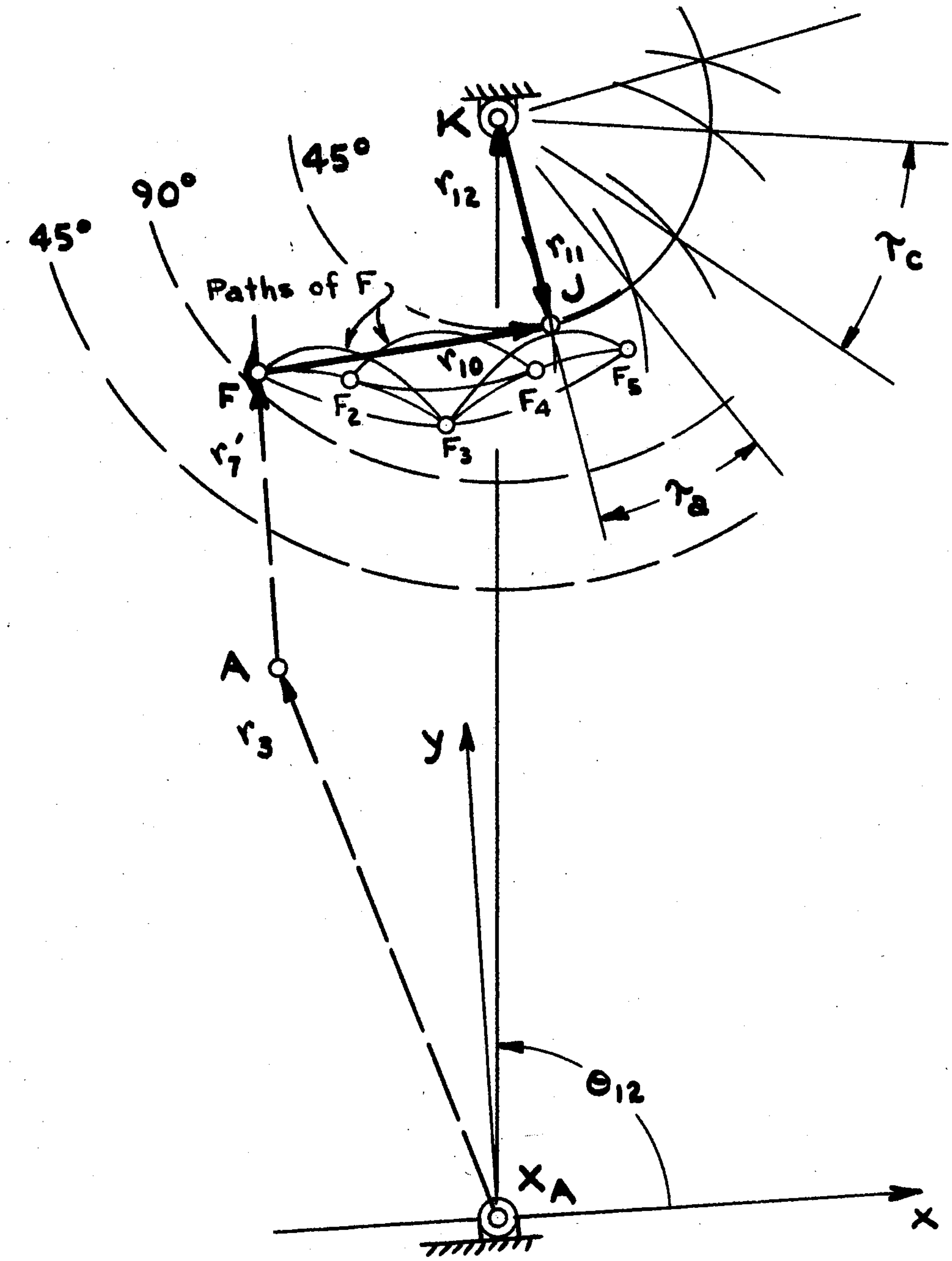


Figure 48. Synthesis of Loop FJK

3. At this point, interaction is made with the synthesis of the cam and follower (section E.3) to determine whether the threshold angle (cam rotation angle) of 24° is large enough to meet the cam specifications. Since 24° is satisfactory the synthesis continues.

4. Arcs are constructed with centers at the five stable points of F. The radius of the arcs is equal to the value of FJ found in step 2. An overlay is constructed as shown in Figure 49. The overlay includes circular arcs for monitoring the transmission angle as explained in Appendix D. The overlay is used to locate the center K so that threshold angles τ_a and τ_c are about 24° or greater and so that the paths of F fall as close as possible to the 90° arc of constant μ.

For the final linkage as shown in Figure 48, it is seen that the transmission angle for all possible locations of point F is greater than 45° but less than 90°.

In section 3.5.2, reference was made to the fact that slight inaccuracies caused in s_F by the approximations of Figure 31 are absorbed in the synthesis of the threshold linkages. It is now observed that these slight inaccuracies simply cause slight differences in the positions of the arcs drawn from the five stable points of F in this step. The overlays are applied to the arcs as they are drawn, and consequently the "inaccuracies are absorbed."

The coordinates of the fixed center K are calculated in computer Program B (Table XIII, found at the end of this appendix) using

$$x_K = r_{12} \cos \theta_{12}$$
$$y_K = r_{12} \sin \theta_{12} \tag{E.1}$$

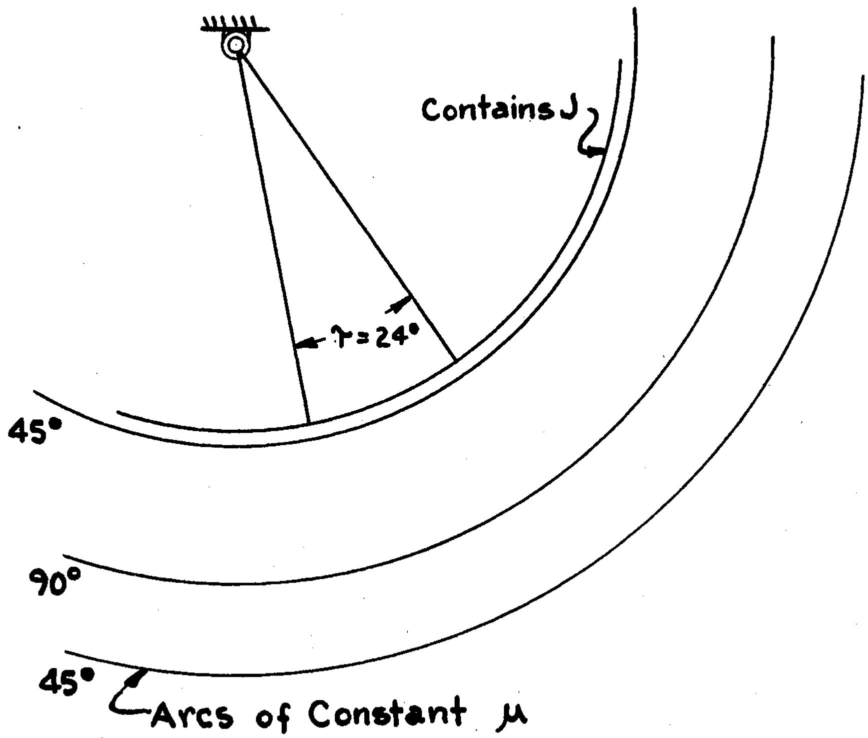


Figure 49. Overlay for Locating K While Monitoring μ in Threshold Linkage Synthesis

E.2 Cam Rotation Analysis θ_{11}

The analysis of the cam rotation is based on the application of the loop model of Figure 45(b) to the loop of Figure 50. In the loop X_A A FJ K X_A, there are two inputs, θ_1 and θ_7. For each input combination, θ_1 has the value $\theta_{"0"}$ or $\theta_{"1"}$ and θ_7 has the magnitudes given by the output analysis of the separator linkages in Program A of Appendix D. The link lengths and the value of θ_{12} used in the analysis are found as described in section E.1.

The analysis is made with a part of the computer Program B (Table XIII) given at the end of this appendix. Since the model loop of Figure 45(b) is applied in the present calculations, the mathematical analysis is the same as that of section D.2.3

The output of the Program B gives four possible values of δ for each input combination. These four values of δ are the result of the two roots given by Equation (D.23). The approximate desired values of θ_{11} and δ are known from the graphical synthesis of section E.1, hence the desired values of θ_{11} and δ are selected from the computer output by observation.

E.3 Synthesis of Cam and Follower

The synthesis of the cam and follower involves the determination of (see Figure 51):

1. The length of the output link HZ,
2. the displacement of the follower h,
3. the prime circle radius KH_0,
4. the radius of the roller of the follower R_r, and

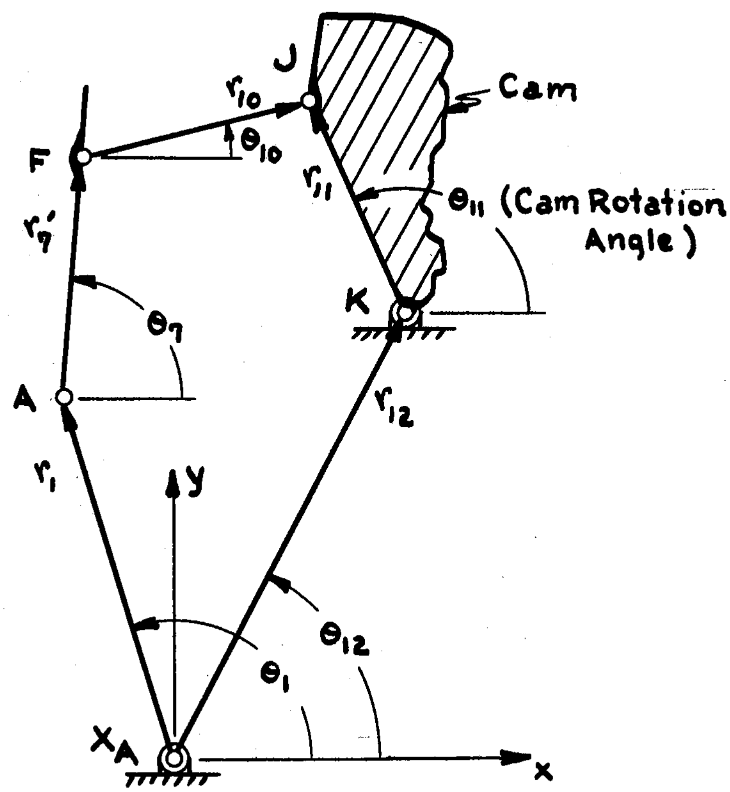


Figure 50. Analysis of Cam Rotation Angle

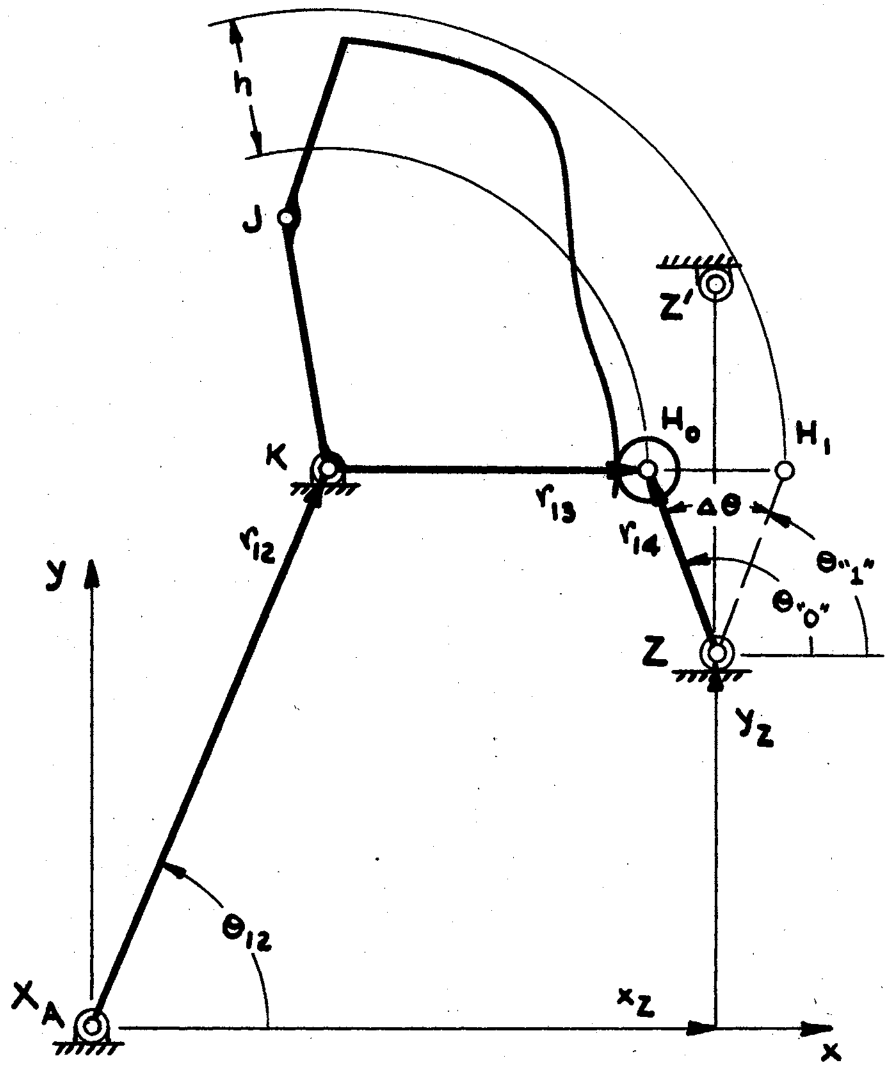


Figure 51. Synthesis of Cam and Follower

5. the location of the fixed centers Z and Z' with respect to the fixed center X_A.

The determination of the five quantities is made as follows:

1. The length of the output crank HZ is selected to be 0.35 ℓ. This value is greater than $\ell/3$, the minimum for good link length ratios.

2. The output rotation of link HZ is selected to be equal to the input rotation angles $\Delta\theta$. Thus the displacement of the cam H_0H_1 is given by

$$h = H_0H_1 = 2\ r_{14}\ (-\cos\ \theta_{"0"}) \qquad \text{(E.2)}$$

The value of h is determined using the computer Program B (Table XIII) given at the end of this section.

3. The minimum prime circle radius KH_0 is dependent upon the follower length HZ, the displacement h, the cam rotation angles (threshold angles τ), the type of motion program, and the maximum allowable pressure angle.

Regarding the threshold angles τ, Table VI is observed. It is seen that all the threshold angles for the linkages of columns I and II were made (in the synthesis of section E.1) about 30° or greater. For the linkages of columns III through IX the threshold angles were made about 24° or greater. Because of the two minimum angles for τ, 30° and 24°, two minimum prime circle radii are determined. The motion program selected is cycloidal and the maximum allowable pressure angle selected in this design is 45°.

The determination of the minimum KH_0 is made with a graphical technique given by Jensen [36]. (The technique is not explained here.) With respect to accuracy, this graphical technique is completely satisfactory. Any small variation from an accurate minimum value of KH_0

merely alters the maximum pressure angle; and a pressure angle slightly greater than 45° is not critical.

The value of KH_0 obtained when $\tau = 24^\circ$ is 0.77ℓ as shown in Table IV, columns I and II. For $\tau = 30^\circ$, the value obtained is 1.0ℓ as shown in columns III through IX.

4. The radius of the roller follower R_r cannot be greater than the minimum radius of curvature of the cam profile. A maximum value for R_r of 0.1ℓ is selected. This value is confirmed to be conservative by Jensen's [36] nomogram for radius of curvature; cycloidal motion.

5. The fixed centers Z and Z$'$ are located with respect to X_A. The coordinates of the two centers are given by

$$\begin{aligned} x_Z &= x_K + r_{13} - r_{14} \cos \theta_{\text{"}0\text{"}} \\ y_Z &= y_K - r_{14} \sin \theta_{\text{"}0\text{"}} \\ x_Z' &= x_Z \\ y_Z' &= y_Z + 2r_{14} \sin \theta_{\text{"}0\text{"}} \end{aligned} \tag{E.3}$$

The values for these coordinates are determined with the computer Program B (Table XIII).

With the Program B, it is noted that no output is included. The reason is that all results obtained from the program are included in Table VI.

TABLE XIII

COMPUTER PROGRAM B.

```
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C      *                                                                     *
C      *                                                                     *
C      *                                                                     *
C      *       PURPOSE                                                       *
C      *              ANALYSIS AND SYNTHESIS OF THRESHOLD LINKAGES           *
C      *
C      *       DESCRIPTION OF PARAMETERS                                     *
C      *              T140 =  ANGLE OF LINKS 1,2, AND 3 IN THEIR             *
C      *                      '0' POSITION                                   *
C      *              N=      NUMBER OF INPUT COMBINATIONS                   *
C      *              D(I)=   LINK LENGTHS                                   *
C      *              P1(I)=  INPUT ANGLES OF LINK 1                         *
C      *              P2(I)=  INPUT ANGLES OF LINK 2                         *
C      *              P5=     ANGLE OF LINK 5                                *
C      *              K=      INDEX INDICATING WHETHER THRESHOLD             *
C      *                      ANGLES ARE 24 OR 30 DEGREES                    *
C      *
C      *       INPUT DATA                                                    *
C      *              CARD 1. T140 WITH FORMAT F10.0                         *
C      *              CARD 2. N WITH FORMAT I2                               *
C      *              CARD 3. (D(I),I=1,6) WITH FORMAT 8F10.0                *
C      *              CARD 4. (P1(I),I=1,N) WITH FORMAT 8F10.0               *
C      *              CARD 5. P2 WITH FORMAT F10.0                           *
C      *              CARD 6. P5 WITH FORMAT F10.0                           *
C      *              CARD 7. K WITH FORMAT I2                               *
C      *                                                                     *
C      *                                                                     *
C      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DIMENSION D(6),P1(8),P2(8),P3(8),P4(8),E(5),
     1C(4,8),S(4,8),P(3,8),P41(8),P42(8),DP41(8),DP42(8)
      PI=3.1415926535
      RAD=PI/180.0
      DEG=180.0/PI
   11 FORMAT (8F10.6)
   12 FORMAT (I2)
   28 FORMAT (1H1,(/12X,F10.6))
   50 FORMAT (1H1,8X,'SAMPLE OF OUTPUT ANGLES THETA 11 AND DELTA FOR LIN
     1KAGE OF W;T=(2,1,1;A,C)'//
     25X,'J            THETA 11                                 DELTA'//
     314X,'(1)         (2)           (11)           (12)          (21)
     4      (22)')
   51 FORMAT (/,5X,I1,3X,F11.6,5(2X,F11.6))
      READ(5,11)T140
      T140=T140*RAD
      DO 45 L=1,9
      READ(5,12)N
      READ(5,11)(D(I),I=1,6)
      READ(5,11)(P1(I),I=1,N)
      READ(5,11)(P2(I),I=1,N)
      READ(5,11)P5
      READ(5,12)K
      WRITE(6,50)
      DO 22 I=1,N
      P1(I)=P1(I)*RAD
      P2(I)=P2(I)*RAD
   22 P3(I)=90.0*RAD
      P5=P5*RAD
C
C     DETERMINATION OF COORDINATES OF FIXED
C           CENTERS K, Z AND ZP
C
      XK=-D(5)*COS(P5)
      YK=-D(5)*SIN(P5)
      IF (K.EQ.30) GO TO 23
      R13=1.0
      GO TO 24
   23 R13=0.77
   24 CONTINUE
      R14=0.35
      XZ=XK+R13-R14*COS(T140)
      YZ=YK-R14*SIN(T140)
      XZP=XZ
      YZP=YZ+2.0*R14*SIN(T140)
      R10=D(6)
      R11=D(4)
      HHP=2.0*R14*(-COS(T140))
```

```
C                                                                              TSHLD000
C       ANALYSIS OF CAM ROTATION ANGLE THETA 11                                TSHLD000
C                                                                              TSHLD000
        DO 15 I=1,5                                                            TSHLD450
        E(I)=D(I)/D(6)                                                         TSHLD460
     15 CONTINUE                                                               TSHLD470
        DO 20 J=1,N                                                            TSHLD480
        P(1,J)=P1(J)                                                           TSHLD490
        P(2,J)=P2(J)                                                           TSHLD500
        P(3,J)=P3(J)                                                           TSHLD510
     20 CONTINUE                                                               TSHLD520
        DO 25 I=1,3                                                            TSHLD530
        DO 25 J=1,N                                                            TSHLD540
        C(I,J)=E(I)*COS(P(I,J))                                                TSHLD550
        S(I,J)=E(I)*SIN(P(I,J))                                                TSHLD560
     25 CONTINUE                                                               TSHLD570
        C5=E(5)*COS(P5)                                                        TSHLD580
        S5=E(5)*SIN(P5)                                                        TSHLD590
        DO 35 J=1,N                                                            TSHLD600
        CO=C(3,J)-C5-C(2,J)-C(1,J)                                             TSHLD610
        SI=S(3,J)-S5-S(2,J)-S(1,J)                                             TSHLD620
        F=(1-E(4)*E(4)-CO*CO-SI*SI)/(2.0*E(4))                                 TSHLD630
        GD=F+CO                                                                TSHLD640
        G1=2.0*SI/GD                                                           TSHLD650
        G2=(F-CO)/GD                                                           TSHLD660
        GR=G1*G1-4.0*G2                                                        TSHLD670
        H=SQRT(GR)                                                             TSHLD680
        TP1=(G1+H)/2.0                                                         TSHLD690
        TP2=(G1-H)/2.0                                                         TSHLD700
        D1=1.0+TP1*TP1                                                         TSHLD710
        D2=1.0+TP2*TP2                                                         TSHLD720
        SP1=2.0*TP1/D1                                                         TSHLD730
        SP2=2.0*TP2/D2                                                         TSHLD740
        CP1=(1.0-TP1*TP1)/D1                                                   TSHLD750
        CP2=(1.0-TP2*TP2)/D2                                                   TSHLD760
        P41(J)=ATAN2(SP1,CP1)                                                  TSHLD770
        P42(J)=ATAN2(SP2,CP2)                                                  TSHLD780
        P1(J)=P1(J)*DEG                                                        TSHLD790
        P2(J)=P2(J)*DEG                                                        TSHLD800
        P41(J)=P41(J)*DEG                                                      TSHLD810
        P42(J)=P42(J)*DEG                                                      TSHLD820
        D11=P41(J)-P41(1)                                                      TSHLD830
        D12=P41(J)-P42(1)                                                      TSHLD840
        D21=P42(J)-P41(1)                                                      TSHLD850
        D22=P42(J)-P42(1)                                                      TSHLD860
        WRITE(6,51)J,P41(J),P42(J),D11,D12,D21,D22                             TSHLD870
     35 CONTINUE                                                               TSHLD880
        WRITE(6,28)XK,YK,XZ,YZ,XZP,YZP,R10,R11,R13,HHP,R14                     TSHLD890
     45 CONTINUE                                                               TSHLD900
        STOP                                                                   TSHLD910
        END                                                                    TSHLD920
```

SAMPLE OF OUTPUT ANGLES THETA 11 AND DELTA FOR LINKAGE OF W;T=(2,1,1;A,C)

J	THETA 11		DELTA			
	(1)	(2)	(11)	(12)	(21)	(22)
1	165.565521	-80.794785	0.0	246.360306	-246.360306	0.0
2	168.538391	-56.946533	2.972870	249.333176	-222.512054	23.848251
3	168.538620	-56.946136	2.973099	249.333405	-222.511658	23.848648
4	-38.551392	-169.787338	-204.116913	42.243393	-335.352783	-88.992554
5	-38.551193	-169.787445	-204.116714	42.243591	-335.352783	-88.992661
6	-8.494057	-165.458633	-174.059570	72.300720	-331.023926	-84.663849
7	-8.493729	-165.458420	-174.059250	72.301056	-331.023926	-84.663635
8	11.175558	-142.394562	-154.389954	91.970337	-307.959961	-61.599777

VITA

Glenn Hancock Dewey

Candidate for the Degree of

Doctor of Philosophy

Thesis: DESIGN OF BINARY LOGIC SYSTEMS USING MECHANICAL ELEMENTS

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Wardsboro, Vermont, May 11, 1935, the son of Garner G. and Mary W. Dewey. On December 17, 1964 married Amalia Vargas M.

Education: Graduated from Leland and Gray Seminary (High School), Townshend, Vermont, in 1953; received the Bachelor of Science degree in Mechanical Engineering from the University of Vermont in 1957; received the Master of Science degree in Mechanical Engineering from the University of Vermont in 1959; completed the requirements for the Doctor of Philosophy degree at Oklahoma State University in May, 1973.

Professional Experience: Teaching at the University of Vermont, 1957-1959; teaching at Methodist high schools in Havana, Cuba and San Jose, Costa Rica, 1959-1966; teaching at the University of Costa Rica, San Jose, Costa Rica, 1966-1969; working as graduate teaching assistant at the Department of Mechanical Engineering, Oklahoma State University, 1971-1972.