#3

THE SYNTHESIS OF SEQUENTIAL SWITCHING CIRCUITS

D. A. HUFFMAN

LOAN COPY Only

TECHNICAL REPORT NO. 274

JANUARY 10, 1954

RESEARCH LABORATORY OF ELECTRONICS
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS

Reprinted from the JOURNAL OF THE FRANKLIN INSTITUTE
Vol. 257, Nos. 3 and 4, March and April, 1954

The Research Laboratory of Electronics is an interdepartmental laboratory of the Department of Electrical Engineering and the Department of Physics.

The research reported in this document was made possible in part by support extended the Massachusetts Institute of Technology, Research Laboratory of Electronics, jointly by the Army Signal Corps, the Navy Department (Office of Naval Research), and the Air Force (Air Materiel Command) under Signal Corps Contract DA36-039 sc-100, Project 8-102B-0; Department of the Army Project 3-99-10-022; and in part by fellowships from the Sperry Gyroscope Company and the International Business Machines Corporation.

Reprinted from JOURNAL OF THE FRANKLIN INSTITUTE,
Vol. 257, No. 3, March, 1954
Printed in U. S. A.

THE SYNTHESIS OF SEQUENTIAL SWITCHING CIRCUITS *

BY

D. A. HUFFMAN [1]

ABSTRACT

An orderly procedure is developed by which the requirements of a sequential switching circuit (one with memory) can be reduced to the requirements of several combinational switching circuits (those without memory). Important in this procedure are:

1. the flow table: a tabular means by which the requirements of a sequential switching circuit may be stated precisely and by which redundancy in these requirements may be recognized and eliminated, and
2. the transition index: a new variable which indicates the stability (or lack of stability) of a switching device.

The role of those switching devices which are not directly controlled by the input of a sequential switching circuit is investigated thoroughly. The resulting philosophy, which is exploited in synthesis procedures for circuits using either relay or vacuum-tube switching devices, is valid for circuits using other devices as well.

PART I †

I. INTRODUCTION TO RELAY CIRCUIT THEORY

Historical Background

In 1938, C. E. Shannon established an orderly algebraic procedure for the treatment of relay contact networks.[2] This major theoretical advance was based on an analogy with the calculus of propositions used in symbolic logic. In 1948, G. A. Montgomerie [3] described a concise

* This paper is derived from a dissertation submitted in partial fulfillment of the requirements for the degree Doctor of Science at the Massachusetts Institute of Technology.

[1] Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Mass.

† Part II will be published in this JOURNAL for April, 1954.

[2] C. E. Shannon, "A Symbolic Analysis of Relay and Switching Circuits," *Trans. AIEE*, Vol. 57, pp. 713–723 (1938).

[3] G. A. Montgomerie, "Sketch for an Algebra of Relay and Contactor Circuits," *Jour. IEE*, Vol. 95, Part III, pp. 303–312 (1948).

method of specifying static (or "combinational") relay circuits which has since been used extensively by Shannon [4] and others, and is now called a "table of combinations." The need for similar orderly analysis methods for circuits with dynamic relay action was realized by Montgomerie,[3] and some of the matrices described here bear superficial resemblance to his, even though they were developed independently.

Assumed Form of the Switching Circuit

Our generalized relay switching circuit will have p input terminals, and q output terminals. Each of the input terminals will be connected to a terminal (called the *control terminal*) of the coil winding of a single-winding relay; the other end of the winding will be connected to the proper supply voltage. (See Fig. 1, for example.) These relays, which are under the direct control of the circuit inputs, are designated as X or *primary* relays. All other relays (if any) in the circuit are to be called Y or *secondary* relays. Their number will be designated s.

Each of the s secondary relays will be controlled by a network of contacts from (in general) all of the relays—both primary and secondary —in the circuit. Similar networks of contacts (the Z networks) will lead to each of the q output terminals from ground.

Contacts associated with a given relay (R) will be given the lower-case designations. *Normally open* contacts are labeled with the *unprimed* letter (r), and the *normally closed* contacts with the *primed* letter (r'). We shall assume that, for a given relay:

1. All normally open contacts are open (or closed) at the same time. Likewise, all normally closed contacts are open (or closed) simultaneously.

2. When the normally open contacts for a given relay are open (or closed), the normally closed contacts are closed (or open), and vice versa.

Algebraic Description of the Relays

A *binary* (two-valued) variable is one which may, at any given time, assume just one of two possible *complementary* values. The binary variables used in describing a relay circuit represent the absence or presence of a metallic path. These variables are assigned the values zero and one. A contact, or a network of contacts, has a *transmission of unity* when it is *closed*, and a *zero transmission* when it is open.[5]

In terms of the transmission concept, $r = 0$ will be interpreted to mean: The normally open contacts on the relay R are open; that is, the

[4] C. E. Shannon, "The Synthesis of Two-Terminal Switching Circuits," *Bell System Tech. Jour.*, Vol. 28, pp. 59–98 (1949).

[5] Transmission is thus somewhat analogous to admittance. The dual concept, that of *hindrance*, is somewhat analogous to impedance; a closed circuit has a zero hindrance and an open circuit has unity hindrance. We shall use the idea of transmission exclusively.

relay is *unoperated.* Similarly $r = 1$ will mean: The normally open contacts on R are closed; that is, the relay is *operated.* Of course—with the assumptions made previously—whenever $r = 0$, then $r' = 1$ and whenever $r = 1$, then $r' = 0$. Thus r and r', when considered as contact variables, always have complementary values.[6]

A relay is *energized* or *de-energized* according as the transmission of the contact network in series with its winding has a transmission of unity or zero, respectively. It will be convenient to use the upper-case letter R as the notation for the transmission of this network. Thus, when $R = 1$, the relay is energized; and when $R = 0$, the relay is de-energized.

It is necessary to make a fine distinction between the state of operation of a relay and its state of energization, as reflected by the variables r and R, respectively. If r and R have the same value the relay is in a *stable* state. If instead, r and R have complementary values the relay is in an unstable state. Since both r and R may each have only the values zero and one, there are just four mutually exclusive situations which may arise:

1. $r = 0$ and $R = 0$; a stable state.
2. $r = 0$ and $R = 1$; an unstable state; the relay is unoperated and is energized. If R continues to have the value one, then eventually the relay will become operated and the value of r will change from zero to one. The time required is called the relay *operate* time.
3. $r = 1$ and $R = 0$; an unstable state; the relay is operated and is de-energized. If R continues to have the value zero, then eventually the relay will become unoperated and the value of r will change from one to zero. The time required is called the relay *release* time.
4. $r = 1$ and $R = 1$; a stable state.

We now define a new variable, τ_R, called the *transition index* which reflects the stability or instability of a relay R. When $\tau_R = 0$ the relay is stable; when $\tau_R = 1$ the relay is unstable.

For convenience we will make use of an algebraic operation which we shall call cyclic addition,[7] and for which the notation is $\oplus$. It will have the following properties:

$$0 \oplus 0 = 1 \oplus 1 = 0; \qquad 0 \oplus 1 = 1 \oplus 0 = 1. \tag{1}$$

In terms of this notation

$$\tau_R = r \oplus R \qquad \text{and} \qquad R = r \oplus \tau_R. \tag{2-a, b}$$

[6] The priming notation may be used whenever the idea of complementation exists. Thus, $0' = 1$ and $1' = 0$.

[7] In the language of congruences, τ_R is congruent, modulo two, to the sum of r and R. See, for example, Birkoff and MacLane, "A Survey of Modern Algebra," New York, The Macmillan Company, 1941, pp. 23–29.

Combinational and Sequential Circuits

Consider for the time being a switching circuit which has no secondary relays; for example, that of Fig. 1(*a*). For this kind of circuit the contact networks which lead to the output terminals must be composed of contacts from relays which are under the direct influence of the input (the primary relays). Since each primary relay may be either operated or unoperated, there are 2^p possible states of operation for the p primary relays, taken collectively. Each of these *primary states* determines uniquely a transmission of either zero or unity at each output terminal. In Fig. 1(*a*), for instance, there is an output ground if X_1 is operated or if X_2 is unoperated, or both. If X_1 is unoperated and if X_2 is operated, however, the output will be ungrounded.

Consider now a circuit having secondary relays as well as primary relays—for example that of Fig. 1(*b*). Assume that all three of the relays are unoperated. Operation of the X_2 relay has no effect on the Y relay but does result in a ground at the output terminal (since the y' contact is closed). As long as Y remains unoperated, the output is grounded or not grounded according to whether X_2 is or is not operated.

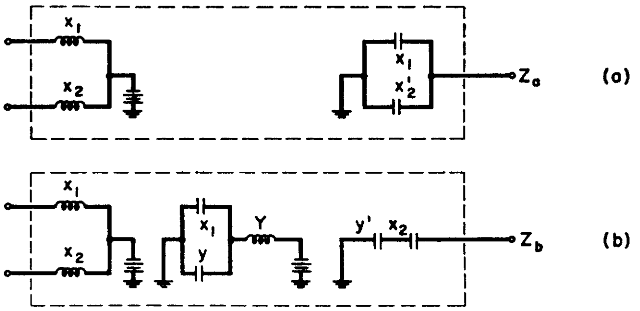


Fig. 1. Illustrating combinational and sequential switching circuits.

Let us now assume that the X_1 relay is operated. The closing of its normally open contact, x_1, energizes and subsequently operates the Y relay, and closes the y contact. Now the state of operation of X_1 cannot affect Y since the latter is permanently energized through its own normally open contact. (Permanently, that is, until the supply voltages are removed.) Now the y' contact in the output network is open and operation of X_2 cannot ground the output terminal.

Notice that the output transmission is not a unique function of the primary relay states of operation but depends as well on the past history of the circuit. The circuit may be said to have a *memory;* it "remembers" whether or not X_1 has been operated.

We may generalize from these two simple examples: In a circuit having no secondary relays there can be no "memory"; the states of operation of the primary relays uniquely determine the output transmissions. Such a circuit is called a *combinational* circuit. In a circuit having secondary relays, the possibility of a "memory" exists since the states of operation may not uniquely determine the output transmissions. A circuit having secondary relays will be called a *sequential* circuit.

Specification of the Terminal Characteristics of Switching Circuits

The terminal characteristics of a combinational circuit may be described equally well by specifying the nature of the contact networks at the outputs. A useful means of specification of these networks (or of *any* two-terminal contact network) is the *table of combinations.* In it are listed the 2^p possible (collective) states of operation of the p primary relays which may contribute contacts to such an output network; beside each of these 2^p states is listed the transmission of the network. In Table I, the table of combinations tells us that the Z_a network (Fig. 1(a)) has a transmission of zero if and only if X_1 is unoperated and X_2 is operated.

TABLE I.—*Tables of Combinations for the Networks in Fig. 1.*

(a)			(b)			(c)		
x_1	x_2	Z_a	x_1	y	Y	x_2	y	Z_b
0	0	1	0	0	0	0	0	0
0	1	0	0	1	1	0	1	0
1	0	1	1	0	1	1	0	1
1	1	1	1	1	1	1	1	0

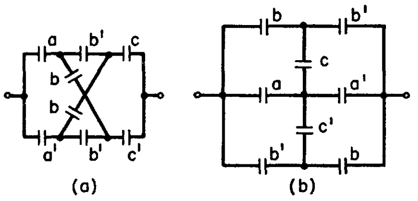
The precise specification of a sequential circuit is more difficult. We can list the transmissions of the Y and Z networks in tables of combinations (see Table I(b) and (c)) but this listing tells us little about the terminal characteristics of the circuit or what function "memory" plays in its operation.

Up to the present time no precise way of tabulating the terminal characteristics of the general sequential switching circuit has been published. The *flow table* developed in Section II is the author's answer to this problem. Without an exact specification, such as the one the flow table gives, we cannot hope to be able to lay down rules for the synthesis of the general sequential circuit.

Comments on the Synthesis of Contact Networks

Our synthesis method for sequential circuits will lead to the listing of the transmission requirements for several Y and Z networks in tables of combinations. The physical realization of contact networks which meet the requirements in a table of combinations has received much study and there is much research yet to be done. We will not attempt to show how to design such networks but will usually diagram one possible physical realization and the reader may verify that it corresponds to its table of combinations.[8]

Part of the problem of synthesizing contact networks exists because several quite different-appearing networks may be terminally equivalent. It is not always easy to decide on a criterion of merit for these different networks or to be sure that once a "good" network is found a "better" one does not also exist. However, a common denominator for all equivalent contact networks is the table of combinations. For example, Fig. 2 gives two equivalent networks along with the table of combinations which they have in common.



a	b	c	Network Transmission
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

(c)

FIG. 2. Two equivalent networks and their table of combinations.

II. THE ANALYSIS OF SEQUENTIAL RELAY CIRCUITS

Matrix Representation for Two-Terminal Networks

In each of the relay switching circuits discussed in this paper, there is a two-terminal contact network connected between ground and the control terminal of each secondary relay. These networks have been given the notations Y_1, Y_2, $\cdots$, Y_s. In addition, there are two-terminal networks between ground and each of the output terminals. These networks are to be given the notations Z_1, Z_2, $\cdots$, Z_q. Each of the Y and Z networks may be a function of all the contact variables for the circuit—that is, of the $(p + s)$ variables $x_1, x_2, \cdots, x_p, y_1, y_2, \cdots, y_s$. Therefore, each of these transmission functions may be represented in a table of combinations with rows corresponding to all possible combinations of the $(p + s)$ contact variables. It is more convenient, however, to use modified tables of combinations which, for the sake of compactness, are put in the form of rectangular matrices.

To illustrate the construction of such matrices, Table II shows the tables of combinations for the three secondary relay control networks

[8] For further study on contact networks the reader is referred to W. Keister, A. E. Ritchie and S. Washburn, "The Design of Switching Circuits," New York, D. Van Nostrand Company, Inc., 1951.

For recent papers on the reduction of combinational circuit requirements to simple forms, refer to M. Karnaugh, "The Map Method for Synthesis of Combinational Logic Circuits," AIEE Technical Paper No. 53-217, April, 1953; or to E. E. Veitch, "A Chart Method for Simplifying Truth Functions," *Proc. Assn. for Computing Machinery*, May, 1952; or to W. H. Burkhart, "Theorem Minimization," *ibid.*

TABLE II.—*Tables of Combinations for the Y and Z Networks of Fig. 3.*

x_1	x_2	y_1	y_2	y_3	Y_1	Y_2	Y_3	Z	
0	0	0	0	0	0	0	0	0	
0	0	0	0	1	1	0	1	0	
0	0	0	1	0	1	1	0	0	
0	0	0	1	1	0	1	1	0	
0	0	1	0	0	0	0	0	0	
0	0	1	0	1	1	0	1	1	*
0	0	1	1	0	1	1	0	0	
0	0	1	1	1	0	1	1	0	
0	1	0	0	0	1	0	0	0	
0	1	0	0	1	1	0	0	0	
0	1	0	1	0	0	0	0	0	
0	1	0	1	1	0	0	0	0	
0	1	1	0	0	1	0	0	0	
0	1	1	0	1	1	0	0	0	
0	1	1	1	0	1	1	1	0	
0	1	1	1	1	1	1	1	0	
1	0	0	0	0	0	1	0	0	
1	0	0	0	1	0	0	1	0	
1	0	0	1	0	0	1	0	0	
1	0	0	1	1	0	0	1	0	
1	0	1	0	0	1	0	0	0	
1	0	1	0	1	1	0	0	0	
1	0	1	1	0	1	0	0	0	
1	0	1	1	1	1	0	0	0	
1	1	0	0	0	1	0	0	0	
1	1	0	0	1	1	0	0	0	
1	1	0	1	0	0	0	0	0	
1	1	0	1	1	0	0	0	0	
1	1	1	0	0	1	0	0	0	
1	1	1	0	1	1	0	0	0	
1	1	1	1	0	1	0	0	0	
1	1	1	1	1	1	0	0	0	

and the single output network of the circuit of Fig. 3. Table III gives the corresponding matrix forms. In general, the Y matrix will have as entries ordered s-tuples of the proper values of Y_1, Y_2, $\cdots$, Y_s; and similarly, the Z matrix will have as its entries ordered q-tuples of the proper values of Z_1, Z_2, $\cdots$, Z_q. Positions in the matrix will be given in the form $(x;y)$, where x is the ordered p-tuple of the values of the primary contact variables $x_1, x_2, \cdots, x_p$, and where y is the ordered s-tuple of the values of the secondary contact variables $y_1, y_2, \cdots, y_s$. For example, the entry "1" in the (00;101) position of the Z matrix of

Table III(b) corresponds to the starred row of Table II. In this same starred row the values for Y_1, Y_2, and Y_3 are one, zero, and one, respectively. And so the entry in the (00;101) position of the Y matrix of Table III(a) is "101."

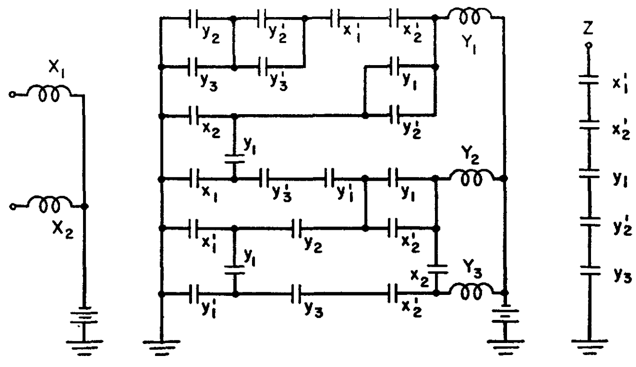


FIG. 3. A sequential switching circuit.

TABLE III.—*Y and Z Matrices Formed from the Data of Table II.*

(a)
The Y Matrix

y \ x:	00	01	10	11
000	000	100	010	100
001	101	100	001	100
010	110	000	010	000
011	011	000	001	000
100	000	100	100	100
101	101	100	100	100
110	110	111	100	100
111	011	111	100	100

(b)
The Z Matrix

y \ x:	00	01	10	11
000	0	0	0	0
001	0	0	0	0
010	0	0	0	0
011	0	0	0	0
100	0	0	0	0
101	1	0	0	0
110	0	0	0	0
111	0	0	0	0

The Composite Transition Matrix

Next we shall form the *composite transition matrix*, τ, for our present example. We shall make use of the equations

$$\tau_{Y1} = Y_1 \oplus y_1, \quad \tau_{Y2} = Y_2 \oplus y_2, \quad \text{and} \quad \tau_{Y3} = Y_3 \oplus y_3, \qquad (3\text{-}a, b, c)$$

which correspond to Eq. 2(a) applied to the three secondary relays. The mechanics of construction of τ are as follows:

For a given primary relay state, $x = (x_1, x_2, \cdots, x_p)$, and for a given secondary relay state $y = (y_1, y_2, \cdots, y_s)$—that is, for a given *total* relay state $(x;y)$,—there is an entry $(Y_1, Y_2, \cdots, Y_s)$ in the Y matrix. In τ, the proper entry at the $(x;y)$ position is $(\tau_{Y1}, \tau_{Y2}, \cdots, \tau_{Ys})$

$= (Y_1 \oplus y_1, Y_2 \oplus y_2, \cdots, Y_s \oplus y_s)$. For instance, the entry in the (00;010) position of the Y matrix of Table III(a) is "110." Since this entry is in the row corresponding to $y = 010$, then we find the proper entry in the τ matrix of Table IV(a) by adding cyclicly the corresponding components of "110" and "010." The result is "100," and this is inserted in the (00;010) position of the τ matrix. This derived entry is to be interpreted: "For the total relay state (00;010), the relay Y_1 is in an unstable state, but the relays Y_2 and Y_3 are both in stable states."

TABLE IV.—*The τ Matrix Derived from Table III.*

y: \ x:	00	01	10	11
000	000	100	010	100
001	100	101	000	101
010	100	010	000	010
011	000	011	010	011
100	100	000	000	000
101	000	001	001	001
110	000	001	010	010
111	100	000	011	011

Later, in the synthesis process, we shall need to reverse the procedure above and derive the Y matrix from the τ matrix. Then we shall need to make use of the equations

$$Y_1 = \tau_{Y1} \oplus y_1, \qquad Y_2 = \tau_{Y2} \oplus y_2, \cdots, Y_s = \tau_{Ys} \oplus y_s, \tag{4}$$

which correspond to Eq. 2(b) applied to the s secondary relays. In order to derive each entry in the Y matrix, we shall have to add cyclicly the components of the appropriate value of $y = (y_1, y_2, \cdots, y_s)$ to each component of the corresponding entry in τ.

In a τ matrix, each entry consisting entirely of zeros indicates stability for all the secondary relays. If an entry contains a single digit one as a component, then just one secondary relay is in an unstable state, and the entry tells us what the secondary relay state will next be if the input state remains unchanged. For instance, in the τ matrix of our present example, the "100" entry at the (00;010) position indicates that the Y_1 relay is unstable; and thus, if the input state remains "00," the resulting total relay state will be (00;110). A glance at τ reveals that all the secondary relays are stable for this new total relay state, since the entry at this position consists entirely of zeros. The circuit action described above is diagrammed by the heavy arrow in Fig. 4.

An important fact to observe in the use of a composite transition matrix is that a change of state in the secondary relays is indicated by vertical movement in the matrix and in the associated diagram, while a change of input state is accompanied by motion in a horizontal direction. Moreover, once a stable state has been reached for all secondary relays, then further circuit changes can occur only if modification is made in the input state. In other words, if the circuit has a stable secondary relay state, and if the input state is changed from the existing value to another one, then the focal point of our attention must first be moved horizontally into the column corresponding to the new input state; and if further changes are to occur, these must be in the vertical direction.

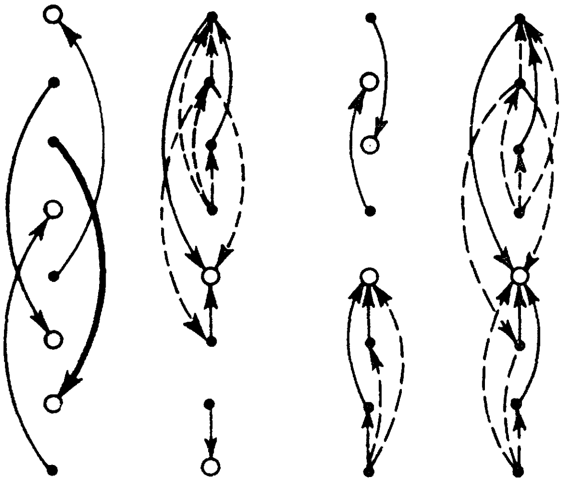


FIG. 4. Transition diagram for the matrix given in Table IV.

Race Conditions

In a composite transition matrix the presence of two or more of the digits "1" indicates that at least two secondary relays are simultaneously unstable, and that a *race condition* exists. In such a case, several secondary relay actions are possible, depending upon the magnitudes of the operating and release times for the relays involved and upon the past history of their excitations. For example, in Table IV, if the total relay state is (10;111), the corresponding entry is "011." This entry indicates that a "race" develops between the relays Y_2 and Y_3 to see which relay will become released first. If Y_2 wins the "race" and is released first, the resulting secondary relay state is "101." If, however, Y_3 is the first relay to release, then the secondary relay state becomes "110." But if a "tie" develops and both relays release at the same time, the secondary relay state will be "100." These three possibilities are indicated by dotted lines rising from the bottom node in the third column of Fig. 4.

All race conditions in Fig. 4 are indicated by dotted lines. In this figure each race condition is *non-critical*, since each of the alternate possibilities leads eventually to the same *ultimate* secondary relay state. For example (see the right-hand column of Fig. 4): If the input state is "11," the ultimate secondary relay state will be "100," no matter what the initial secondary relay state was.

Not all race conditions are of the noncritical variety. It may be that the alternate possibilities present in a race condition lead to different ultimate circuit conditions. Consider, for instance, Table V (and Fig. 5). Here, several ultimate circuit conditions are possible if the total relay state is initially (00;010).

TABLE V.—*A τ Matrix Illustrating Race Conditions and Cycles.*

y: \ x:	00	01	10	11
000	010	000	000	001
001	000	100	011	010
010	101	001	100	110
011	010	010	100	001
100	000	010	000	001
101	000	001	000	010
110	000	100	100	001
111	010	000	100	100

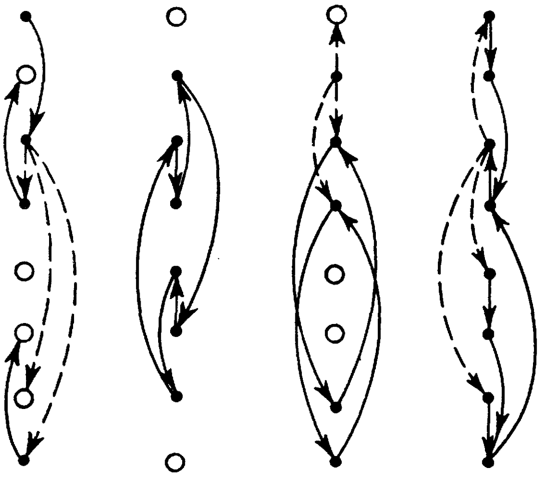


FIG. 5. Transition diagram for τ matrix of Table V.

Cycles and Ultimately Stable Terminal Action

It may be that, for a given input state, internal circuit action is continuously sustained because at least one secondary relay is always maintained in an unstable state. For this situation there will be no single ultimate secondary relay state, but instead there will be a *cycle* of states occurring. The column of Table V (and of Fig. 5) for which the input state is "01" illustrates such a cycle. Here the secondary relay states · · ·, 001, 101, 100, 110, 010, 011, · · · occur in cyclic fashion.

If the output state of a relay switching circuit remains constant for all the secondary relay states of a cycle, we shall say that the circuit has *ultimately stable terminal action.*

It is also possible that race conditions and cycles may coexist in a switching circuit. Two examples of this situation are given in the columns of Table V which correspond to the input states "10" and "11."

The Flow Table

Our attention will now be limited to the analysis of those relay switching circuits in which there are no cycles, and where all race conditions (if any exist) are of the noncritical type. For this kind of circuit we can construct a *flow table, F.* Our running example (Fig. 3, Tables II, III, and IV, and Fig. 4) is of the proper classification, and the associated flow table is given in Table VI.

TABLE VI.—*The Flow Table Derived from Table IV and Fig. 4.*

x: *y*:	00	01	10	11
000	①	5	3	7
001	8	5	②	7
010	9	5	③	7
011	④	5	2	7
100	1	⑤	⑥	⑦
101	⑧	5	6	7
110	⑨	10	6	7
111	4	⑩	6	7

The circles in *F* correspond in position to those entries in τ which are made up of zeros only, and which therefore represent those circuit conditions for which all secondary relays are stable. The circles are numbered serially; the order of assignment of the numbers is unimportant. The remaining entries in *F* are uncircled, and each such entry tells what stable circuit condition will ultimately result if the circuit is put in a total relay state corresponding to the entry in question and if the input state remains unchanged. This stable circuit condition can be derived by analyzing the composite transition matrix, τ. With this rule, and aided by the diagram of Fig. 4, if necessary, we obtain Table VI.

Since we are going to state our synthesis problems in terms of flow tables, it will be worth while to take time to understand thoroughly how a flow table is used and what its entries mean. The flow table can give us a graphic idea of what circuit actions can occur in a relay circuit. Just as in the composite transition matrix, horizontal motion in the flow table corresponds to modification of the input state. Within each row of the table the following rules hold:

1. Each circled entry in a row of a flow table indicates a stable circuit condition, and no further changes will occur unless the input state is modified.
2. Each circled circuit condition within a row of a flow table can lead to any other circuit condition (circled or uncircled) which is listed in the same row of the table.
3. Each uncircled entry in a row of a flow table indicates the stable circuit condition which will ultimately follow if the input state is left constant. This stable circuit condition, circled, will be listed in the same column of the table, but in another row.

For example (see the fifth row of Table VI): If the existing circuit condition is denoted by "6," then circuit conditions "1," "5," or "7" can follow, depending upon whether the input state is changed from "10" to "00," "01," or "11," respectively. If, in particular, circuit condition "1" results (by changing the input state from "10" to "00"), we must next concentrate our attention on the first row of the flow table, because the circled "1" entry appears in this row.

The switching circuit which we have used in this section on analysis is one which is designed to be used in conjunction with an electrically operated lock. The single output of the unit is to act as the switch which energizes the lock so that it can be opened. The two input terminals are connected to two keys, K_1 and K_2, respectively—each of which has a single normally open contact. The "combination" which is to open the lock is the following (the numbers refer to entries in Table VI):

Starting with both keys released (1), K_1 is first depressed (3) and then released (9); K_2 is then depressed (10) and released (4); finally, K_1 is again depressed (2) and the lock is to open (8) when K_1 is then released.

After the lock has been opened (8), depressing either or both of the keys (5, 6, or 7) will allow it to be locked again. If a mistake is made in working the "combination" for the lock, then one of the circuit conditions 5, 6, or 7 will result, and the combination may be started again only after both keys are first released (1).

Summary of Material on Analysis

The transition index τ_R has been defined. Its use led to the composite transition matrix τ. τ and Z (the output matrix) give us all the information that the original circuit diagram can. With a knowledge of the succession of input states, we may determine—with the aid of τ—the states of each relay of the switching circuit. Then, by looking at the proper entry in Z, the output state is determined.

We have concerned ourselves only with the relative order of state changes, and not with the actual timing in the circuit. With this outlook, we have seen some of the situations which may arise. For a circuit with no sustained cycles [9] or potentially critical race conditions, we have defined a flow table F which is useful in determining ultimate circuit conditions, if the input state is changed only after the secondary relays have reached stable states.

III. THE SYNTHESIS OF A SEQUENTIAL RELAY CIRCUIT

Introductory Remarks

In this section our attention will be limited to those design problems in which the input state changes only after a stationary output state has been reached, and in which such an output state will always occur: in other words, those circuit actions which could be those of a relay circuit with ultimately stable terminal action.

The most difficult and the most important part of a synthesis of a relay switching circuit—as with most other syntheses—is the problem of saying what we want to do. The flow table does for sequential relay circuits which have ultimately stable output what the table of combinations does for combinational-type contact networks. In each case the table gives in a precise way meanings that dozens of qualifying statements in a word specification might never be able to convey.

We must keep in mind the situation which faces the designer. First, he would like to specify what the circuit output state is to be for all possible sequences of input states, and not for just a single sequence. It is all very well to say what the output states should be for some "normal" sequence of inputs; but if there is even a possibility that other sequences might occur, then circuit action must be specified for these sequences also. A complete problem specification must indicate clearly what happens for each conceivable set of circumstances.

Secondly, if the circuit designer can honestly say that he cares only about the output state as some function of the input state, he cannot fairly say in the problem statement what secondary relays he wants to use. The problem statement can then be made only in terms of what signals are available (the input states) and what signals are desired (the output states).

[9] It is also possible to write flow tables for circuits with cycles or for circuits in which it is important to specify output states in addition to those which correspond to total relay states for which the circuit is stable. In such flow tables, entries of a type "ⓘj" may be utilized. An entry "ⓘj" is the "destination" of all entries with uncircled i's and it directs our attention to the entry with a circled "j" as the next circuit condition of interest.

The Problem Statement and the Flow Table

In order to illustrate the various problems which arise during the synthesis of a sequential circuit, we shall first limit ourselves to a specific example. The circuit we wish to design has some of the properties of a "delay line." It is to have two input terminals and two output terminals. We want the output state to be the same as the last previous input state. In order to illustrate how restrictions on the input state are taken care of, we shall specify that we shall use our circuit only under those circumstances which allow the input state to change by one variable at a time. In other words, for example, modification of the input state from "01" to "00" or to "11" will be allowed, but from "01" to "10" will be impossible. And so, after our synthesis is complete, it would not be fair to complain of improper circuit action for these forbidden changes of input state; for we specify here that they cannot occur.

With the restrictions named above, there will be just two separate circuit conditions possible for each input state, and each of these will be associated with a different output state. For instance, if the input state is "00," it could have been preceded by either "01" or "10." The corresponding output state then will be either "01" or "10," respectively.

TABLE VII.—*A Flow Table and Output Data for a "Delay Line" Circuit.*

(*a*) Flow Table				(*b*) Output Data	
x:				Stable Circuit Condition	Output State
00	01	10	11		
①	3	5		1	01
②	3	5		2	10
1	③		7	3	00
1	④		7	4	11
2		⑤	8	5	00
2		⑥	8	6	11
	4	6	⑦	7	01
	4	6	⑧	8	10

Consequently, there will be eight stable circuit conditions—two for each of the four possible input states. These are numbered serially and listed as circled entries in Column (*a*) of Table VII, each in a separate row and each in the proper column. The output state associated with each is listed in Column (*b*).

Next we list the uncircled entries of the flow table, making use of the three rules stated previously (see p. 173), and being careful to make certain the conditions in the word statement of the problem are satisfied.

The positions in the flow table which correspond to unallowable input transitions are left blank.[10]

The first row of our derived flow table informs us, for instance, that circuit condition "1" (which may occur for input state "00") may lead to either circuit condition "3" or "5," and Table VII, Column (*b*) verifies that each of these latter conditions will give the proper output state, "00."

Condensation of the Flow Table by Row Merging

From our discussion of flow tables in the section on analysis, we know that there is going to be one secondary relay state assigned to each row of the flow table. If our object is to reduce the number of secondary relays as far as possible, we would like to be sure that our problem has been stated in a flow table with as small a number of rows as possible. Sometimes it is possible to *merge* two rows of a flow table. Each such *merger* will reduce the number of rows by one. The rule for merging of rows in a flow table is this:

Table VIII.—*Hypothetical Derivation of a* τ *Matrix Corresponding to Table VII.*

(*a*) Flow Table

y: \ x:	00	01	10	11
00	①	③	5	7
11	②	3	⑤	8
10	1	④	6	⑦
01	2	4	⑥	⑧

(*b*) The τ Matrix

y: \ x:	00	01	10	11
00	00	00	11	10
11	00	11	00	10
10	10	00	11	00
01	10	11	00	00

Two or more rows of a flow table may be merged if—and only if—for each input state, these rows do not have conflicting entries. An entry which appears in any one of the merged rows will appear in the composite row. Those entries which are circled in any one of the merged rows will be circled in the row resulting from the merger.

Application of this rule to Column (*a*) of Table VII indicates that the following pairs of rows may be merged: one and three, two and five, four and seven, and six and eight. The condensed flow table which results from making all four mergers is given in Table VIII.[11] The

[10] We may also leave a position in a flow table vacant if, for the corresponding input-state transition, we do not care to define the circuit action to follow. See the further discussion of this point in *Input-Output Sets*, page 183 of this paper.

[11] In our example the result of making all possible row mergers gives an answer which is independent of the order in which the mergers are made. In *Requirements for a Unique Flow-Table Condensation*, page 188 of this paper, however, it will be proved that the result of condensing a flow table to the point where no further mergers are possible may not be unique if (as in our example) there are "vacancies" in the original table: that is, if some modifications of input state are prohibited.

merger of rows one and three, for example, results in the first row of this new table.

The first row of Column (*a*) in Table VIII preserves the information contained in each of the component rows. It retains the information that circuit condition "1" leads to condition "3" or condition "5," depending on whether the input state is changed from "00" to "01" or to "10," respectively. Similarly, this first row tells us (as did the third row of Table VII, Column (*a*)) that circuit condition "3" may lead either to condition "1" or to condition "7," depending on the change of input state.

In the circuit which we shall synthesize from the condensed table, circuit condition "3" would lead to circuit condition "5" *if* the input state could be changed from "01" to "10." But the "if" is an enormous one; for we stated firmly during the specification of the problem that this transition in input state could not occur. Thus it is important to recognize that, since this information which the top row of Table VIII, Column (*a*) gives is about a hypothetical change in input state, it will neither be of value to us nor cause us trouble. Similar situations are apparent in each of the four rows in this same Column (*a*).

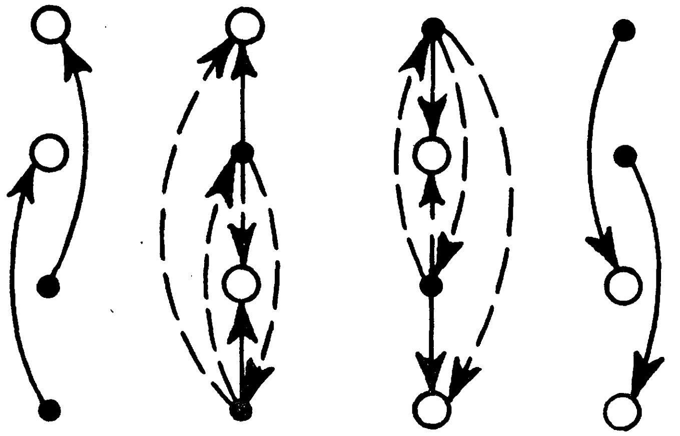


FIG. 6. Transition diagram for the derivation in Table VIII.

The Assigning of Secondary Relay States to the Rows of the Flow Table

Since there are four rows in the condensed flow table, it will be necessary to have at least two secondary relays in order to assign a distinct secondary relay state to each row. If we were to hypothesize that the operating and release times of each secondary relay were precisely the same, and that this time was the same for each of the secondary relays, then the problem of assigning secondary relay states to the rows of the flow table would be easy indeed. For then we could assign them in some arbitrary fashion to the rows, as has been done in Column (*a*) of Table VIII. The assumption that all operating and release times were all the same would guarantee that all race conditions would end in "ties," and we could then form the composite transition matrix of Column (*b*) of Table VIII, which corresponds to the diagram of Fig. 6. The transitions which correspond to the "tied races" are indicated by the solid lines in the second and third columns of the figure.

From a practical point of view, the presence of race conditions such as those discussed above is perilous; for if the operating characteristics of a relay change slightly over a period of time, one or more of the transitions indicated by dotted lines becomes possible. These transitions would result in improper operation of the switching circuit.

We can avoid the difficulties of race conditions in which "ties" *must* be achieved in order for proper circuit operation to occur (and therefore in which some entries in τ *must* indicate two or more relays to be unstable). What we need to do is to assign secondary relay states to the flow table in such a way that in τ, only *one* secondary relay is indicated to be unstable at a given time. After having done this, we may then reconsider some of the entries in τ and allow, if we desire, *noncritical* race conditions to exist.

For our present example, one assignment of secondary relay states which avoids critical race conditions is that given in Table IX, Column (*a*).

TABLE IX.—*An Alternate Development of τ and Y from Column* (*a*) *of Table VII.*

(*a*) Flow Table

y: \ x:	00	01	10	11
00	①	③	5	7
01	②	3	⑤	8
10	1	④	6	⑦
11	2	4	⑥	⑧

(*b*) τ Matrix

y: \ x:	00	01	10	11
00	00	00	01	10
01	00	01	00	10
10	10	00	01	00
11	10	01	00	00

(*c*) Y Matrix

y: \ x:	00	01	10	11
00	00	00	01	10
01	01	00	01	11
10	00	10	11	10
11	01	10	11	11

Derivation of the τ and Y Matrices

The τ matrix corresponding to our choice of secondary relay states is given in Table IX, Column (*b*). By reversing the procedure of analysis, we add cyclicly to each entry in τ the secondary relay state assigned to the row in which the entry is found. The Y matrix of Column (*c*) results.

Derivation of the Z Matrix

By consulting the output data of Table VII, Column (*b*), we may immediately assign to the Z matrix those output states which correspond to the eight stable circuit conditions of Table IX, Column (*a*). The result is the partially completed matrix of Table X.

TABLE X.—*Illustrating Development of the Z Matrix.*

(*a*) Partial *Z* Matrix

x: y:	00	01	10	11
00	01	00		
01	10		00	
10		11		01
11			11	10

(*b*) Completed *Z* Matrix

x: y:	00	01	10	11
00	01	00	00	01
01	10	00	00	10
10	01	11	11	01
11	10	11	11	10

If we can honestly say that we are concerned with the output state *only* after all the secondary relays of the switching circuit become stable, then any choice whatsoever may be made in the output states which are used to complete the *Z* matrix.

However, there are certain advantages to completing the matrix as we have indicated in Table X, Column (*b*). Here we have assigned to each entry in *Z* the output state associated with the corresponding entry in *F*, even if the entry is uncircled in *F*. For example, both the (01;00) and (01;01) positions in *Z* are given the output state "00," since these correspond to those positions in *F* which contain "3" as an entry, and since the circuit condition "3" is associated with the output state "00." If *Y* is completed in this way, we assure that, as the input state is changed from its present value to a new value, the output state immediately becomes and remains constant at the value corresponding to the ultimate circuit condition.

TABLE XI.—*The Derived Tables of Combinations.*

x_1	x_2	y_1	y_2	Y_1	Y_2	Z_1	Z_2
0	0	0	0	0	0	0	1
0	0	0	1	0	1	1	0
0	0	1	0	0	0	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	0	1	0	1	1
0	1	1	1	1	0	1	1
1	0	0	0	0	1	0	0
1	0	0	1	0	1	0	0
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	1
1	1	0	0	1	0	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	0	0	1
1	1	1	1	1	1	1	0

The Tables of Combinations and the Final Circuit

We may reverse the steps taken in the section on analysis and decompose the Y and Z matrices—listing the entries in tables of combinations instead of in the matrix form. The data in Table XI result from the breaking down of the entries found in Tables IX(c) and X(b).

The purpose of this paper is to demonstrate that the problem of the design of sequential relay circuits may be reduced to the problem of the design of combinational-type contact networks. Our task is then complete, and the designer of combinational circuits may now take the data in Table XI and design corresponding secondary relay controlling networks and the output networks. One possible realization of these is given in the final circuit diagram of Fig. 7.

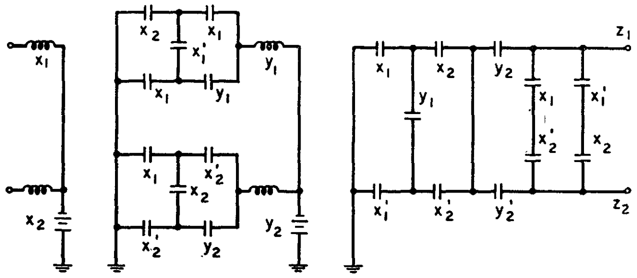


FIG. 7. The synthesized "delay-line" circuit.

IV. FLOW TABLE MANIPULATION

Simplification of the Original Problem Statement

In our zest for listing all possible circuit actions in a problem specification, it is conceivable that we may say more than we actually need to.

TABLE XII.—*The Simplification of a Problem Statement.*

(a)					(b)					(c)				
x:					x:					x:				
00	01	10	11	Output State	00	01	10	11	Output State	00	01	10	11	Output State
①	7	2	6	0	①	7	2	6	0	①	7	2	6	0
1	4	②	3	0	1	4	②	3	0	1	4	②	3	0
1	4	5	③	1	1	4	5	③	1	1	4	5	③	1
1	④	5	6	0	1	④	5	6	0	1	④	5	6	0
1	4	⑤	6	0	1	4	⑤	6	0	1	4	⑤	6	0
1	4	5	⑥	0	1	4	5	⑥	0	1	4	5	⑥	0
1	⑦	5	8	0	1	⑦	5	3	0	1	⑦	5	3	0
1	4	5	⑧	1	1	4	5	③	1					

Two or more circuit conditions which we have listed separately in a flow table might, if proper means of analysis were available, be recognized as a single condition. The elimination of this redundancy in the problem statement may be quite important to us if our object is to get as simple a circuit realization as possible. In the following sections we show that this overstatement of a problem can occur, and we shall give rules for detecting it.

As an example to show how redundancy in a flow table might arise, let us examine Table XII, Column (*a*) and the corresponding output data. This table specifies a circuit which delivers an output ground only at the end of the input sequence "00" → "10" → "11" or at the end of the input sequence "00" → "01" → "11." The first sequence corresponds to the sequence of circuit conditions "1" → "2" → "3," and the second sequence to the sequence of circuit conditions "1" → "7" → "8." The word specification of the problem is satisfied in the flow table, but we may suspect that "3" and "8" could actually be considered as the same circuit condition, since each of them is associated with the same input state and the same output state.

Let us assume, tentatively, that conditions "3" and "8" are equivalent. If this is true, any possible sequence of input states must give identical outputs whether we start with the circled entry "3" or with the circled entry "8." The third and eighth rows of Column (*a*) of Table XII tell us that given changes of input state result in the same ultimate circuit conditions, whether we start initially from "3" or from "8"; and so our original assumption is valid.

Recognition of the equivalence of conditions "3" and "8" allows us to reduce the number of circled entries in the flow table by replacing the designator "8" with the designator "3." If this is done everywhere in the table, Column (*b*) is the result. In this column the eighth row is now redundant and it may be eliminated, as in Table XII, Column (*c*). The effect of this procedure can be expressed alternatively by saying that we have merged rows three and eight, and that the merger was valid because circuit conditions "3" and "8" were equivalent.

Of course we may continue to condense the flow table by merging the fourth, fifth, and sixth rows into a single row. These latter mergers, however, would not change the number of circled entries in the table.

Now that our appetite for simplifying flow tables has been whetted, we may be tempted to see equivalence when it does not exist. In Table XII, Column (*c*), for instance, circled entries "4" and "7" are both associated with the same input state and both give the same output state. If the initial circuit condition is "4," a change of input state from "01" to "11" produces the condition "6." On the other hand, if the initial circuit condition is "7," the same change of input state (from "01" to "11") gives the condition "3." But conditions "3" and "6" cannot be equivalent because they give different output states. Therefore "4" and "7" cannot be considered equivalent either.

TABLE XIII.—*The Simplification of a Problem Statement.*

(a)					(b)					(c)				
x:					x:					x:				
00	01	10	11	Output State	00	01	10	11	Output State	00	01	10	11	Output State
6	①	4	7	10	6	①	3	7	10	6	①	3	7	10
6	②	3	7	10	6	①	3	7	10	5	1	③	8	11
5	1	③	8	11	5	1	③	8	11	⑤	1	3	7	11
5	2	④	8	11	5	1	③	8	11	⑥	1	3	8	00
⑤	2	3	7	11	⑤	1	3	7	11	5	1	3	⑦	00
⑥	1	3	8	00	⑥	1	3	8	00	6	1	3	⑧	01
5	1	4	⑦	00	5	1	3	⑦	00					
6	2	4	⑧	01	6	1	3	⑧	01					

Sometimes the reasoning is not so straightforward as it was in the above example. For instance, in Table XIII, Column (*a*), we might examine for equivalence the entries "1" and "2." If we assume for the time being that they are equivalent, then examination of the flow table indicates that the validity of our assumption depends on whether or not entries "3" and "4" are equivalent. When we look at the third and fourth rows, we find in turn that "3" and "4" are equivalent if "1" and "2" are. But this follows from the original assumption. Similar reasoning would hold if we were to assume initially the equivalence of "3" and "4." Our conclusions must be: "1" and "2" are equivalent, and so are "3" and "4."

Because of the equivalences discovered above, we may condense Column (*a*) of Table XIII by merging the first and second rows and by merging the third and fourth rows. The procedure is first to replace "2" by "1" and "4" by "3" everywhere in the table (see Column (*b*)) and then to eliminate the redundant rows as in Column (*c*).

Statement of Rules

With the aid of reasoning which we have done above, we now define equivalence as follows:

Two circled entries of a flow table may be considered equivalent if, and only if,

1. each entry appears in the same column of the flow table (is associated with the same input state), and if
2. each entry is associated with the same output state, and if,
3. for each sequence of input states which could start equally well from either of the two circuit conditions involved, the corresponding output sequence is independent of which circuit condition was used as the starting point.

Another way of saying this same thing is this: In a flow table, two circled entries which are associated with the same input state and with the same output state are equivalent *unless* there exists some sequence of input states which may start from either of these two circled entries, and yet which gives corresponding output sequences which differ from each other.

There will be further illustration of these qualifications in the following sections.

Problem Statements Which Have a Unique Simplification

Input-Output Sets

The circled entries of a flow table may be partitioned into mutually exclusive *input-output sets.* Within each such set the members all have a given input state and a given output state in common. If two entries belong to different input-output sets, then either their input states or their output states, or both, differ. For example, in Table XIV, Column (*a*) we may place the entries into the following input-output sets:

(1,7,9,12); (5,14); (2,6,11); (4,8,13); and (3,10,15).

Notice that the pattern of vacancies in the rows corresponding to the members of each set is the same. For instance, the second, sixth, and eleventh rows have the following patterns of entries:

5	②		3
5	⑥		10
5	⑪		3

In each of these rows the position in the third column is left vacant.

For flow tables which are constructed in such a way that the vacancy pattern in the rows corresponding to the members of any given input-output set is the same for all members of the set, there is a unique way of eliminating redundancy and therefore a unique minimum-circled-entry form of the table. This statement will be proved later in the section entitled *Equivalence-Sets* (see page 185).

Meanwhile, let us note that a vacancy in a row of a flow table may indicate one of two things: Either (1) the input transition corresponding to the vacancy is impossible, or (2) for this transition we do not care to define the resulting circuit action. In the latter case we must be satisfied with whatever circuit action actually occurs when the relay circuit is in operation.

When we have a flow table before us, we need not be concerned with the reasons for vacancies—only that they exist. Therefore, for purposes of the following discussion, let us assume that vacancies in a flow table indicate that the corresponding transition in input state is impossible.

Testing for Equivalence

Imagine now that we have before us two identical relay switching circuits of the kind described by a given flow table. Assume also that this flow table meets the requirement that the same vacancy pattern shall exist for all members of a given input-output set. If we wanted to test two different circuit conditions, "i" and "j," for equivalence, we might do this by putting the two switching circuits into the conditions "i" and "j," respectively.

Imagine now that a given sequence of input states is impressed upon each of the circuits. If, at any time in this sequence, we find that an input transition which is possible for one circuit is not possible for the other, we might immediately conclude that the respective conditions of the two circuits are not listed in the same input-output set. Therefore we would have found one sequence of input states which, starting with the initial circuit conditions "i" and "j," did not result in the same output sequences. We should then conclude that "i" and "j" are not equivalent.

TABLE XIV.—*A Unique Simplification of a Problem Statement.*

(*a*)

x:				Output
00	01	10	11	State
①	11	4	10	01
5	②		3	11
5	2	13	③	11
12		④	15	00
⑤		8		10
14	⑥		10	11
⑦	6	8	3	01
7		⑧	3	00
⑨	11	13	10	01
12	6	13	⑩	11
5	⑪		3	11
⑫	2	4	15	01
1		⑬	10	00
⑭		8		10
1	6	4	⑮	11

(*b*)

x:				Output
00	01	10	11	State
①	2	4	10	01
5	②		3	11
5	2	4	③	11
1		④	10	00
⑤		8		10
5	⑥		10	11
⑦	6	8	3	01
7		⑧	3	00
①	2	4	10	01
1	6	4	⑩	11
5	②		3	11
①	2	4	10	01
1		④	10	00
⑤		8		10
1	6	4	⑩	11

(*c*)

x:				Output
00	01	10	11	State
①	2	4	10	01
5	②		3	11
5	2	4	③	11
1		④	10	00
⑤		8		10
5	⑥		10	11
⑦	6	8	3	01
7		⑧	3	00
1	6	4	⑩	11

With the aid of the reasoning above, we know that if any two circuit conditions are equivalent to a third, then any input sequence which is possible starting from any one of these conditions is possible starting from the other two, and corresponding output sequences are the same for all. It follows that each of the three conditions is equivalent to the others.

Equivalence-Sets

By the logical extension of the argument above, we know that, for a flow table meeting our restriction of uniform vacancy pattern for all members of an input-output set, the input-output sets themselves may be partitioned into *equivalence-sets.* These equivalence-sets are to be defined in such a way that each member of the set is equivalent to every other member of the set, and so that no member of one equivalence-set is a member of another.

For Column (*a*) of Table XIV, we may—by techniques discussed in preceding sections—determine that the equivalence-sets are: (1,9,12); (7); (5,14); (2,11); (6); (4,13); (8); (3); and (10,15).

It is easiest to test for equivalence when a flow table is in its *primitive* form: that is, with one circled entry per row. If we have determined that the flow table meets our required restrictions on the placement of vacancies, we may place the entries in equivalence-sets. If our object is to simplify the problem statement as far as possible (reduce the number of circled entries to a minimum), the best we can do is to replace each member of an equivalence-set by a single entry from that set, and then to eliminate the resulting superfluous rows of the table. This minimum number of circled entries is clearly unique, and is the same as the number of equivalence-sets.

In Column (*b*) of Table XIV, we have replaced the members of each equivalence-set by the member of this set which has the lowest numerical designation, and then we have eliminated the redundant rows. The table we have derived (Column (*c*)) is the simplest possible way of stating the terminal action specified by the original flow table of Column (*a*).

A Problem Simplification Which is not Unique

If a circuit is described by a primitive flow table which does not meet the requirements of uniformity of vacancy pattern within each input-output set, then it may not be possible to partition the entries of the flow table into equivalence-sets. Practical situations in which there may not be this required uniformity may occur if the input is the output of another sequential switching circuit, or if a human being arbitrarily restricts the input transitions allowable in accordance with his knowledge of the internal state of the circuit.

TABLE XV.—*Two Simplifications of the Same Problem Statement.*

(*a*)

x:				
00	01	10	11	Output State
①	3		5	01
②	7		8	01
6	③	4	5	00
1		④		10
9		4	⑤	11
⑥	3	4		01
9	⑦		8	11
2	3		⑧	01
⑨		4	8	01

(*b*)

x:				
00	01	10	11	Output State
①	3	4	5	01
②	7	4	8	01
1	③	4	5	00
1		④		10
2		4	⑤	11
2	⑦		8	11
2	3		⑧	01

(*c*)

x:				
00	01	10	11	Output State
①	3		5	01
②	7		8	01
6	③	4	5	00
1		④		10
6		4	⑤	11
⑥	3	4	8	01
6	⑦		8	11
2	3		⑧	01

In Column (*a*) of Table XV, the flow table does not meet the restrictions of uniformity of vacancy pattern; see, for example, the first and sixth rows. Inspection of the flow table and the output data in Column (*a*) shows us that the only input sequence which may start with both condition "1" and condition "6" is that beginning "00, 01, · · ·," and that for this sequence, condition "3" follows immediately whether we start with "1" or with "6." Thus "1" and "6" have a sort of equivalence. We shall call it *pseudo-equivalence* in order to distinguish it from the true equivalence, which we discussed earlier (see section beginning on page 183).

By reasoning similar to that used above, we may establish that entries "2" and "9" possess this pseudo-equivalence also. And the same is true for entries "6" and "9." The pseudo-equivalences which we have found (there are no more equivalences of any kind) in the flow table of Column (*a*) are summarized here.

"1" ↔ "6" "2" ↔ "9" "6" ↔ "9"

Notice that we cannot form mutually exclusive equivalence-sets as we did with the tables meeting the uniform vacancy requirement.

The inability to form sets of equivalent entries leads to situations which were not possible for flow tables meeting the uniform vacancy restriction. If we replace "6" by "1" and "9" by "2" in Column (*a*) of Table XV, and if we then merge the pairs of rows which contained these circled entries, the flow table of Column (*b*) can be derived. (When we compare, for example, the first row of Column (*b*) with the first and sixth rows of Column (*a*), we find that we have retained the information in the two component rows.) If, instead, we replace "9" by "6" everywhere in Column (*a*), and then merge the rows containing these circled entries, Column (*c*) follows.

Now consider Columns (*b*) and (*c*) of Table XV, both of which represent the same problem, and in neither of which can the number of circled entries be reduced further. It might well be that two different people would have stated the problem in these two forms in the first place, and so we cannot guarantee that there is a unique "simplest" primitive form of the flow table for the general sequential circuit problem.

Row Mergers

An Example in Which the Flow-Table Condensation is not Unique

Column (*a*) of Table VIII had a unique minimum-row form in which no further row mergers were possible. In general, however, when a flow table has "vacancies" there may not be a unique minimum-row form. In the flow table of Column (*a*) of Table XVI, for example, let us assume that the output states associated with the various circuit conditions are such that no redundancies exist, and therefore that no two circled entries are equivalent. In this table, any one of the following pairs of rows may be merged:

TABLE XVI.—*Illustrating Non-Unique Flow-Table Condensations.*

(*a*)

x: 00	01	10	11
①	2	5	
1	②		4
6		③	8
	2	3	④
1		⑤	4
⑥	2	3	
1	⑦		8
	7	3	⑧

(*b*)

x: 00	01	10	11
①	②	⑤	4
6	7	③	⑧
⑥	2	3	④
1	⑦		8

(*c*)

x: 00	01	10	11
①	2	⑤	4
1	②	3	④
⑥	2	③	8
1	⑦	3	⑧

1, 2   1, 5   2, 4   2, 5   3, 6   3, 8   4, 6   7, 8

If we merge rows one, two, and five; rows three and eight; and rows four and six; the result is that shown in Column (*b*). If, instead, we merge rows one and five, rows two and four, rows three and six, and rows seven and eight, the resulting condensed flow table is that of Column (*c*).

The flow tables of Columns (*b*) and (*c*) in Table XVI do not have rows which can be merged further; yet each contains the same information that the table of Column (*a*) does. From this simple example, therefore, we have demonstrated that, in general, a flow table cannot—by row merging—be reduced to a unique minimum-row form.

Requirements for a Unique Flow-Table Condensation

The result obtained in the previous section leads us naturally to the question: "Are there flow tables in which the operation of row merging leads us to a unique minimum-row form of the table?" The answer is "Yes." If a flow table has no "vacancies" (that is, if no input transitions are prohibited), there is such a unique form. We shall see below the reasoning behind our affirmative answer.

For any flow table of the "completely filled" type, we may form mutually exclusive sets of rows in which the pattern of entries in the rows of each set is the same. (The first three rows in Column (*a*) of Table XVII comprise such a set.) Then all the rows within each set may be merged to give a single composite row which will have the same entry pattern as the members of the set. After all the rows within each set have been merged, the resultant condensed flow table has as few rows as we can obtain. This minimum-row result is unique, and independent of the order in which the mergers were made.

Table XVII.—*Illustrating a Unique Flow-Table Condensation.*

(*a*)

x:			
00	01	10	11
①	3	2	8
1	3	②	8
1	③	2	8
5	7	2	④
⑤	7	2	4
1	3	⑥	4
9	⑦	6	8
9	7	6	⑧
⑨	7	6	8

(*b*)

x:			
00	01	10	11
①	③	②	8
⑤	7	2	④
1	3	⑥	4
⑨	⑦	6	⑧

By way of illustration of the principles stated above, consider Column (*a*) of Table XVII. The entry patterns present in the various rows of the table allow the following mutually exclusive sets of rows to be formed: (1,2,3); (4,5); (6); and (7,8,9). The minimum-row form of this table is found by merging all rows of a set with each other. The result in our present example is the flow table shown in Column (*b*) of Table XVII.

Row Splitting

For the sake of completeness we must include among the possible manipulations on a flow table a process which is the opposite of row merging. It will be called *row splitting*. Future investigation may show that this process is sometimes useful in modifying a flow table so that the assignment of secondary relay states is made easier.

When a row of a flow table is split into several other rows, the pattern of entry designations in each of these new rows must be equivalent to that of the original row. In the flow table of Column (*a*) of Table XVIII, one way of splitting the row containing the circled entries one and five, and of splitting the row containing the circled entries four and six, is that shown in Column (*b*). In this derived table the rows resulting from the splitting process give the same description of circuit action that the original flow table did, and we could obtain again the original flow table by making all possible mergers in Column (*b*).

TABLE XVIII.—*Illustrating Splitting of Rows in a Flow Table.*

(*a*)

x: 00	01	10	11
①	6	⑤	3
1	②	4	3
7	2	⑧	③
⑦	⑥	④	3

(*b*)

x: 00	01	10	11
①	6	5	3
1	6	⑤	3
1	②	4	3
7	2	⑧	③
⑦	6	4	3
7	⑥	④	3

Another more complicated type of row splitting occurs if the row we attempt to split has but one circled entry. Then the effect is to increase the number of circled entries in the flow table; or we may say that the single circled entry of such a row is itself split. The rules for *splitting an entry* into two or more parts are easy to state. If we split an entry "i" into m different entries, we take the row of the flow table which contains the circled entry "i" and replace it by m rows which contain circled entries "i_1," "i_2," $\cdots$, "i_m." Each of these m circled entries is to be equivalent to the others. Thus each will be in the same column as the original entry "i," and the output state associated with each will be the same as that of the original entry. In each of the rows of the new flow table we insert uncircled entries which are either the same as the corresponding entries in the original table or equivalent to them.

By way of illustration, let us, in Column (*a*) of Table XIX, split the entry "1" into three parts, and the entry "3" into two parts. One way of doing this is that given in Column (*b*). Notice that in this latter table the uncircled entries "1" and "3" have been replaced by equivalent entries everywhere in the table. We could, of course, simplify the table of Column (*b*) to produce again the flow table of Column (*a*).

Table XIX.—*Illustrating Splitting of Entries in a Flow Table.*

(*a*)

x:

00	01	10	11
①	3		4
7	3	②	
	③	5	4
7	6	5	④
1		⑤	
7	⑥	2	
⑦	3		4

(*b*)

x:

00	01	10	11
$\textcircled{1_1}$	3_1		4
$\textcircled{1_2}$	3_2		4
$\textcircled{1_3}$	3_2		4
7	3_2	②	
	$\textcircled{3_1}$	5	4
	$\textcircled{3_2}$	5	4
7	6	5	④
1_2		⑤	
7	⑥	2	
⑦	3_1		4

Combinations of the techniques above may be used to split the rows in a flow table. (See the example leading up to Table XXV.) The test for the validity of such splits is always to determine whether or not the rows resulting from each split may be merged again to give the original row.

(*To be continued.*)

Reprinted from JOURNAL OF THE FRANKLIN INSTITUTE.
Vol. 257, No. 4, April, 1954
Printed in U. S. A.

THE SYNTHESIS OF SEQUENTIAL SWITCHING CIRCUITS*

BY

D. A. HUFFMAN[1]

PART II†

V. GENERALIZATION OF THE SECONDARY RELAY ASSIGNMENT PROBLEM

On the Number of Secondary Relays Required

For s secondary relays there are 2^s secondary relay states available for assignment to the rows of the flow table. Since we want each of the n rows of the flow table to be assigned a separate and distinct secondary relay state, there must be at least as many secondary relay states available as there are rows in the table. In other words, it is necessary that $n \leqq 2^s$.

The least conceivable number of secondary relays which we shall need for the flow table will be called s_0; and s_0 will have the property that it is the least integer which satisfies the inequality above.

It can also be demonstrated, by actually specifying a circuit, that certainly no more than $2s_0 + 1$ secondary relays are necessary for the synthesis of a circuit which is described by a flow table with n rows.[12]

Derivation of Formalized Assignment Criteria

Rather than try to build up a cumbersome catalog of situations which may arise in the assignment of secondary relay states, we shall immediately point out some other objectives we seek in the synthesis of any relay switching circuits.

It is our purpose to design circuits in such a way that the relationships among the operate and release times for the various relays play no part in the circuit operation. This requirement eliminates the possibility of including critical race conditions in the properties of the synthesized circuit. Where no race conditions exist, it is always possible for circuit operation to occur in such a way that successive secondary relay states involve only *unit* changes of state: that is, changes in which a single secondary relay variable is modified (see, for example, Table IV and Fig. 4).

In a flow table in which only unit changes of secondary relay state are to occur, each uncircled entry "k" must lead to the circled entry

* This paper is derived from a dissertation submitted in partial fulfillment of the requirements for the degree Doctor of Science at the Massachusetts Institute of Technology.

[1] Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Mass.

† Part I was published in this JOURNAL for March, 1954.

[12] See *The "$2s_0 + 1$" Realization*, page 284 of this paper

"k" (in the same column of the table) by a succession of *adjacent* secondary relay states. We will say then that each uncircled "k" entry is *connected* to the circled "k" entry.

We shall call a *set* of secondary relay states *connected* if each member of the set can be connected to every other member of the set by at least one succession of adjacent states from the set. For example, in Table XXIV(a), the 8-set (for $k = 8$) is composed of the secondary relay states "100," "010," and "110." This 8-set is connected, since (for instance) the state "010" may be connected to the state "100" by the succession "010"–"110"–"100."

In the synthesis procedure, therefore, we wish to assign secondary relay states to the rows of the flow table *in such a way that each k-set is connected.* (There will be as many k-sets as there are circled entries.) This assures us that the τ matrix can be constructed in such a way that only unit changes of state are indicated.

An important fact to keep fixed in mind in assigning secondary relay states is that the adjacency of two states is not affected by making a given interchange of variables in the state designators, or by complementing corresponding variables in the two states. For instance, the two states "1011" and "1010" are adjacent. If we interchange the second and third variables in each state, we get "1101" and "1100," respectively, and these new states are still adjacent. If, instead, we complement the first variable in each of the original state designators, the results are "0011" and "0010," respectively, and again the modified states are adjacent.

The reasoning above tells us that once we have any solution to the problem of assigning secondary relay states to the rows of the flow table, then we may perform given interchanges of variables and/or complementations on any number of the secondary relay states, and still have a valid assignment.

An Ideal Situation for Assignment

The Problem Statement

Here we include a synthesis example in which the solution of the assignment problem is straightforward. Its purpose is to show some of the goals which we should strive for, even though—in the syntheses of most switching circuits—they will not be so readily reached.

The circuit we wish to synthesize has two input leads and four output leads (see Fig. 8). The input restrictions are such that only

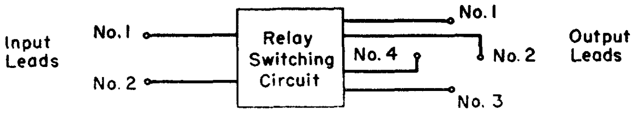


FIG. 8. A switching circuit to be synthesized.

one input lead may be grounded at a time, and so that neither input lead may be grounded unless both are first ungrounded. This latter restriction prohibits the transitions in input state from "01" to "10," and vice versa.

Of the four output leads, one—and only one—is to be grounded at a time. With the second input ungrounded, each grounding of the first input lead is to "advance" the position of the output ground by one step (clockwise on the schematic diagram). Removal of the input ground is to have no effect upon the output: for example, if initially both input leads are ungrounded and the No. 2 output lead is the one grounded, grounding of the first input lead is to remove the ground from the No. 2 output lead and impress the ground instead upon the No. 3 output lead. Subsequent removal of the input ground is to have no further effect on the output.

TABLE XX.

(a)					(b)			
x:				Output	x:			
00	01	10	11	State	00	01	10	11
①	8	10		1000	①	8	10	
②	5	11		0100	②	5	11	
③	6	12		0010	③	6	12	
④	7	9		0001	④	7	9	
1	⑤			1000	1	⑤	⑨	
2	⑥			0100	2	⑥	⑩	
3	⑦			0010	3	⑦	⑪	
4	⑧			0001	4	⑧	⑫	
1		⑨		1000				
2		⑩		0100				
3		⑪		0010				
4		⑫		0001				

The grounding of the second input lead, on the other hand, is to make the position of the output ground "retreat" to the next counterclockwise lead in the schematic diagram.

Our synthesized circuit is then to be a sort of "reversible counter"; a ground on one input lead is to have an effect opposite to that of a ground on the other input lead.

The Synthesis

A flow table corresponding to our problem statement is the one given in Table XX, Column (*a*). In it, no equivalences exist, but the mergers indicated in Column (*b*) may be made. In this latter flow table there are eight rows, and thus we know that at least three secondary relays will be necessary for a realization.

Table XXI.

(*a*) Flow Table

x: / *y*:	00	01	10	11
000	①	8	10	
001	4	⑧	⑫	
010	1	⑤	⑨	
011	④	7	9	
100	2	⑥	⑩	
101	③	6	12	
110	②	5	11	
111	3	⑦	⑪	

(*b*) τ Matrix

x: / *y*:	00	01	10	11
000	000	001	100	
001	010	000	000	
010	010	000	000	
011	000	100	001	
100	010	000	000	
101	000	001	100	
110	000	100	001	
111	010	000	000	

(*c*) *Y* Matrix

x: / *y*:	00	01	10	11
000	000	001	100	(101)
001	011	001	001	(001)
010	000	010	010	(010)
011	011	111	010	(110)
100	110	100	100	(100)
101	101	100	001	(000)
110	110	010	111	(011)
111	101	111	111	(111)

(*d*) *Z* Matrix

x: / *y*:	00	01	10	11
000	$\underline{1000}$	1000	1000	(1000)
001	0001	$\underline{0001}$	$\underline{0001}$	(0001)
010	1000	$\underline{1000}$	$\underline{1000}$	(1000)
011	$\underline{0001}$	0001	0001	(0001)
100	0100	$\underline{0100}$	$\underline{0100}$	(0100)
101	$\underline{0010}$	0010	0010	(0010)
110	$\underline{0100}$	0100	0100	(0100)
111	0010	$\underline{0010}$	$\underline{0010}$	(0010)

Since each *k*-set consists of just two members, we may make each such set connected if we assign adjacent secondary relay states to these two members. One assignment satisfying this restriction is that of Table XXI, Column (*a*). It is ideal in the sense that a change in state of no more than one secondary relay ever is required in passing from any circuit condition to any other. Further, this assignment can be made without any modification of the flow table from the form given in Column (*b*) of Table XXI, other than reordering of its rows.

Clearly there will be many flow tables for which this fortuitous assignment may not be made. Nevertheless the operational advantages of this kind of assignment are such that any flow table should first be optimistically examined with the hope that it is a case of this type.

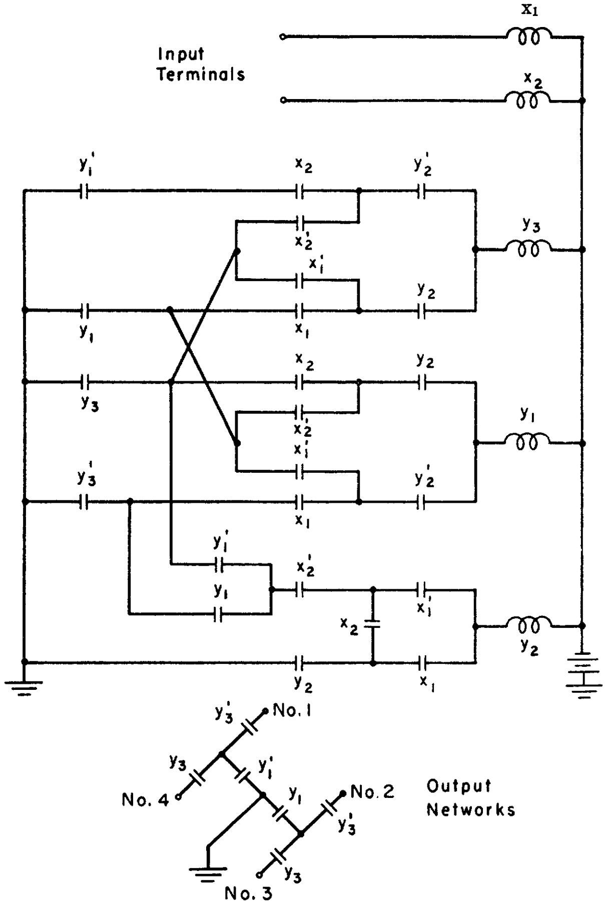


FIG. 9. The synthesized "reversible counter."

The τ and Y matrices were developed in the usual manner. The underlined entries of the Z matrix correspond to circled entries in the flow table and therefore are the output states associated with these entries. In order to simplify the output networks the additional entries in the Z matrix were chosen so that the output state does not change to its ultimate value until the proper secondary relay action has been completed.[13] The parenthesized entries in the Y and Z matrices corre-

[13] Contrast this procedure with that in *The Tables of Combinations and the Final Circuit*, Part I of this paper, page 180.

spond to non-occurring total relay states, and they were chosen to simplify the contact network structures in the circuit diagrammed in Fig. 9.

A Non-Ideal Situation for Assignment

The Problem Statement

The flow table in Column (*a*) of Table XXII describes a circuit which has two outputs, each somewhat under the direct control of its respective input. Starting from condition "1" (for which neither of the inputs and neither of the outputs is grounded) grounding of the x_1 input grounds the z_1 output ("3"); grounding of the x_2 input grounds the z_2 output ("2"). But simultaneous grounding of both x_1 and x_2 results in no ground at the output ("10"). In the latter case no possibility of an output ground exists ("8" and "9") until the circuit is returned to condition "1" by the ungrounding of both inputs.

In case a ground on the z_1 output lead was originally obtained ("3") it will remain ("5" and "7") until such a time as it is removed by the appearance of a ground signal on the x_2 input lead only ("8"). Similarly, in case a ground on the z_2 lead was originally obtained ("2") it will remain until such time as it is removed by the appearance of a ground on the x_1 input lead only ("9").

The Synthesis

Since all input transitions are allowed, the flow table is completely filled. Its minimum-row form is unique (see Table XXII, Column (*b*)) and at least $s_0 = 2$ secondary relays will be necessary for a realization.

TABLE XXII.

(*a*)

x: 00	01	10	11	Output State
①	2	3	10	00
6	②	9	4	01
7	8	③	5	10
6	2	9	④	01
7	8	3	⑤	10
⑥	2	9	4	01
⑦	8	3	5	10
1	⑧	9	10	00
1	8	⑨	10	00
1	8	9	⑩	00

(*b*)

x: 00	01	10	11
①	2	3	10
⑥	②	9	④
⑦	8	③	⑤
1	⑧	⑨	⑩

TABLE XXIII.

(*a*) Flow Table

y: \ x:	00	01	10	11
000	①	2	3	10
001	⑥	②	9	④
010	⑦	8	③	⑤
011				
100	1	⑧	⑨	⑩
101			9	
110		8		
111				

(*b*) τ Matrix

y: \ x:	00	01	10	11
000	000	001	010	100
001	000	000	100	000
010	000	100	000	000
011				
100	100	000	000	000
101			001	
110		010		
111				

Notice that the flow table requires that, in order to make the 1-, 2-, and 3-sets connected, the secondary state assigned to its first row be adjacent to the states assigned to its fourth, second, and third rows, respectively. Because no one of the four states (00, 01, 10, and 11) which are available from two relays can be adjacent to each of the other states, we know that two relays will not be sufficient for a proper secondary assignment. Therefore, for this flow table, at least three relays are necessary.

In actuality no proper assignment can be made to the flow table as it stands in Column (*b*) of Table XXII, regardless of how many secondary relays are used. We may recognize the dilemma which exists if we appreciate that each secondary relay state is either *even* or *odd*, depending on whether the state designator has an even or odd number of unity components. Thus, the state "10010" is even, while "10110" is odd. If two states (such as the two above) are adjacent, one must be even and the other odd.

In Table XXII, Column (*b*), therefore, if we choose to assign an odd state to the first row of the table, then the state associated with the second row must be even (since the 2-set must be connected and the two members of the set must then be adjacent). If an odd secondary relay state is associated with the first row of the table—as we have assumed—that associated with the fourth row must also be even, since the 1-set must be connected. But the second and fourth rows cannot both be assigned even states, since this makes it impossible for the two members of the 9-set to be adjacent.

The dilemma above may be bypassed if we *augment* the flow table by inserting additional "8" and "9" entries and if we assign the secondary states as shown in Column (*a*) of Table XXIII. The purpose of augmenting the flow table in this way is to make both the 8-set and the 9-set connected. The corresponding partial τ matrix is that of Column (*b*).

Another possible solution to the assignment problem may occur if we do not make all possible row mergers in the original primitive table.

Table XXIV.

(*a*) Flow Table

y: \ x:	00	01	10	11
000	①	2	3	10
001	6	②	9	4
010	7	8	③	5
011				
100	1	⑧	⑨	⑩
101	⑥	2	9	④
110	⑦	8	3	⑤
111				

(*b*) τ Matrix

y: \ x:	00	01	10	11
000	000	001	010	100
001	100	000	100	100
010	100	100	000	100
011				
100	100	000	000	000
101	000	100	001	000
110	000	010	100	000
111				

For instance, the assignment made in Table XXIV is a valid one for our present example.

If we split the entries in each row of the table of Column (*b*) of Table XXIV and make the secondary state assignment of Table XXV, Column (*a*) (notice the symmetry about the dotted horizontal line), we obtain a flow table in which no more than one secondary relay need

Table XXV.

(*a*) Flow Table

y: \ x:	00	01	10	11
000	①$_1$	2_1	3_1	10_2
001	⑥$_1$	②$_1$	9_1	④$_1$
010	⑦$_1$	8_1	③$_1$	⑤$_1$
011	1_2	⑧$_1$	⑨$_1$	⑩$_1$
100	1_1	⑧$_2$	⑨$_2$	⑩$_2$
101	⑦$_2$	8_2	③$_2$	⑤$_2$
110	⑥$_2$	②$_2$	9_2	④$_2$
111	①$_2$	2_2	3_2	10_1

(*b*) τ Matrix

y: \ x:	00	01	10	11
000	000	001	010	100
001	000	000	010	000
010	000	001	000	000
011	100	000	000	000
100	100	000	000	000
101	000	001	000	000
110	000	000	010	000
111	000	001	010	100

TABLE XXVI.

(*a*) *Y* Matrix

y \ x:	00	01	10	11
000	000	001	010	100
001	001	001	011	001
010	010	011	010	010
011	111	011	011	011
100	000	100	100	100
101	101	100	101	101
110	110	110	100	110
111	111	110	101	011

(*b*) *Z* Matrix

y \ x:	00	01	10	11
000	$\underline{00}$	00	00	00
001	$\underline{01}$	$\underline{01}$	01	$\underline{01}$
010	$\underline{10}$	10	$\underline{10}$	$\underline{10}$
011	00	$\underline{00}$	$\underline{00}$	$\underline{00}$
100	00	$\underline{00}$	$\underline{00}$	$\underline{00}$
101	$\underline{10}$	10	$\underline{10}$	$\underline{10}$
110	$\underline{01}$	$\underline{01}$	01	$\underline{01}$
111	$\underline{00}$	00	00	00

change state in going from one circuit condition to another.[14] The τ, Y, and Z matrices corresponding to this latter assignment are given in Table XXV, Column (*b*) and Table XXVI. The final circuit diagram is shown in Fig. 10.

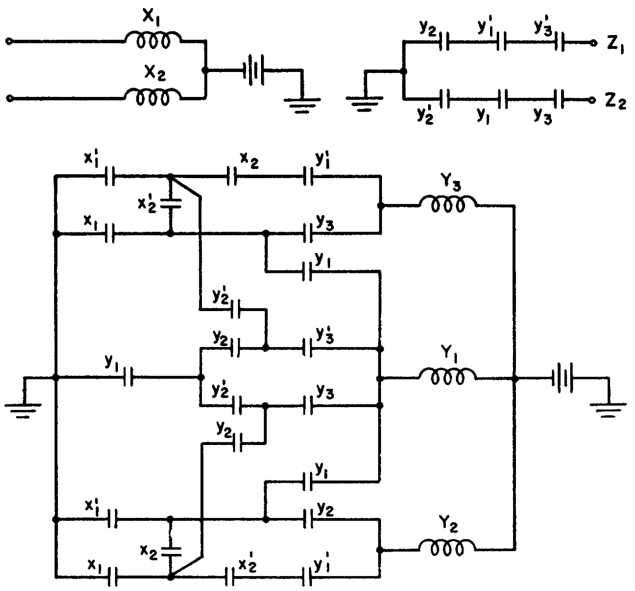


FIG. 10. The circuit derived from the flow table of Table XXV.

[14] It is easy to prove that *any* four-row table may be processed in the manner of this paragraph and that the resultant circuit is one in which a change in circuit condition requires, at most, a change of state of just one secondary relay. Inherent in such a proof is the conclusion that *any* four-row table may be realized in a circuit which uses three secondary relays.

VI. STANDARD CIRCUIT REALIZATIONS

Purpose

For some complex flow tables it may not be practically possible for the designer to find a solution to the problem of assigning secondary relay states, even though he knows (from the preceding section) what principles should guide him. For the general sequential circuit (with ultimately stable terminal action) the author knows of no definitive procedure which assigns secondary relay states to the flow table in such a way that all k-sets are connected, and so that a minimum number of secondary relays is used. Because of this fact it is valuable for a circuit designer to have at hand "standard" circuit realizations for the secondary relay control networks. The following two sub-sections demonstrate that such realizations exist, no matter what the form of the flow table may be.

The "$2s_0 + 1$" Realization

In this section we shall demonstrate that, for a flow table with n rows, a circuit can be designed which uses exactly $2s_0 + 1$ secondary relays.[15] Since, for the arbitrary flow table with n rows, no general method has yet been found which uses fewer relays, we must consider this as an upper limit on the number needed. If other design methods can be found which will be valid for any flow table and which will always require fewer than $2s_0 + 1$ secondary relays for the circuit realization, then a new and better upper limit will have been discovered.[16]

The "$2s_0 + 1$" realization is one which is valid regardless of the details of construction of the particular flow table. It is a "standard" circuit in the sense that certain main features remain the same, no matter what flow table we wish to give a circuit realization. The schematic circuit diagram for the "$2s_0 + 1$" realization is given in Fig. 11 for the particular case $s_0 = 3$. In order to adapt this circuit to a given flow table, we need only design the various C networks (and their complements) and substitute them into the standard circuit. The number of C networks it will be necessary to design will be the same as s_0.

Even though the idea of connected k-sets is valid for the "$2s_0 + 1$" realization, it is also cumbersome, and we shall satisfy ourselves with a direct demonstration that all race conditions which occur are noncritical in nature.

In order to discuss the actions of the secondary relays, we shall break up these relays into two groups (the α-group and the β-group) of s_0 relays each; the remaining relay will be called Y_0. Thus the $2s_0 + 1$ secondary relays will be called $Y_{\alpha 1}$, $Y_{\alpha 2}$, $\cdots$, $Y_{\alpha s_0}$, $Y_{\beta 1}$, $Y_{\beta 2}$, $\cdots$, $Y_{\beta s_0}$, and Y_0.

[15] s_0 was defined in *On the Number of Secondary Relays Required*, this paper, page 275.

[16] In recent research the author has proved that $2s_0 - 1$ is always a sufficient number of secondary relays to implement a flow table. However, for relay circuits, this upper bound does not correspond to a simple standard circuit, as does the $2s_0+1$ bound. The results of this research will be published later this year.

It will be easier to demonstrate how a circuit can be constructed with $2s_0 + 1$ secondary relays if we work with an example. The example we have chosen is that of Table XXVII(a). Here there is a flow table of $n = 7$ rows and $2s_0 + 1$ is found to be equal to seven. The secondary relay states assigned have been indicated beside the rows of the table. Each secondary relay state, y, has been decomposed into components consisting of the states of the α-group of relays, those of the β-group of relays, and the state of the single relay, Y_0. In our example the notation for the secondary relay state is $y = (y_\alpha, y_\beta, y_0)$.

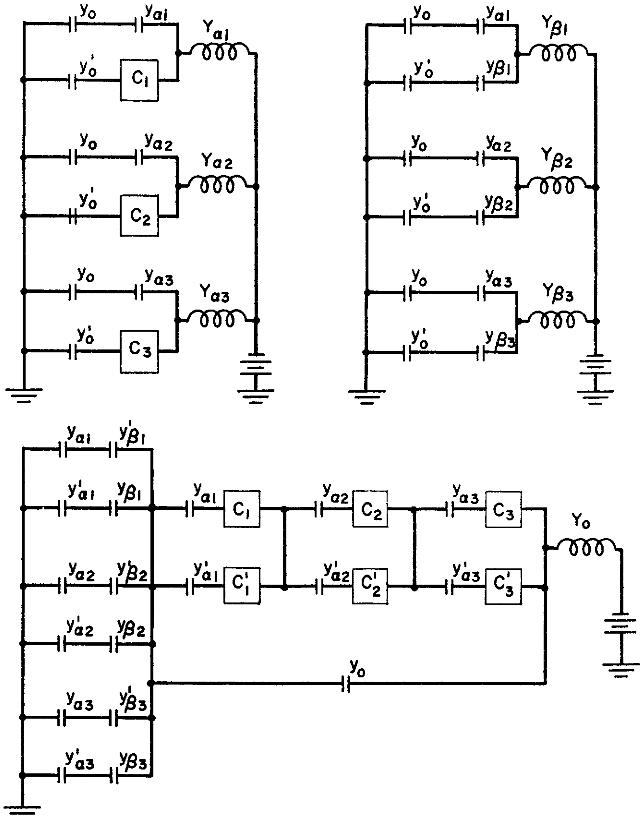


FIG. 11. The "$2s_0 + 1$" circuit realization (for $s_0 = 3$).

y_α and y_β represent the states of the α- and the β-groups of three (s_0) relays each, and y_0 is the state of the remaining relay. For instance, in the third row of the table in (a) of Table XXVII, $y_\alpha = y_\beta = 010$ and $y_0 = 0$.

We soon notice in (a) of Table XXVII that each secondary relay state, y, has been chosen of the form $y = (y_i, y_i, 0)$. In other words, each row has assigned to it a set of secondary relay states so that $y_\alpha = y_\beta$ and $y_0 = 0$. Since each of the α- and β-groups of relays has three (s_0) members, we may be sure that there are enough secondary relay states

(y_i, y_i, 0) available to go around among the rows of the flow table. (The particular values of y_i assigned to the various rows of the table are completely arbitrary.)

A circuit can be constructed to utilize the assignment of secondary relay states which is given in Table XXVII(*a*). Figure 11 represents such a circuit. Especial attention should be paid to the construction of the contact networks controlling the relays in the α-group and in the β-group, and to that controlling the y_0 relay, because the form of these networks is the same no matter what the number of rows in the flow table with which we start.

TABLE XXVII.—*A Flow Table and the Corresponding C_1 Network in Matrix Form.*

(*a*)

$y = (y_\alpha, y_\beta, y_0)$: \ x :	000	001	010	011	100	101	110	111
000 000 0	①	10	㉙	⑫	13	㉖	⑱	16
001 001 0	21	㉘	③	17	㉓	⑭	18	⑧
010 010 0	9	②	⑪	12	23	14	⑮	㉔
011 011 0	⑨	⑩	3	④	13	⑲	7	24
100 100 0	1	⑳	11	㉕	⑤	19	㉚	8
101 101 0	9	2	㉒	⑰	5	6	⑦	⑯
110 110 0	㉑	20	22	4	⑬	⑥	15	㉗

(*b*)

$y_\beta = (y_{\beta 1}, y_{\beta 2}, y_{\beta 3})$: \ x :	000	001	010	011	100	101	110	111
0 0 0	0	0	0	0	1	0	0	1
0 0 1	1	0	0	1	0	0	0	0
0 1 0	0	0	0	0	0	0	0	0
0 1 1	0	0	0	0	1	0	1	0
1 0 0	0	1	0	1	1	0	1	0
1 0 1	0	0	1	1	1	1	1	1
1 1 0	1	1	1	0	1	1	0	1
1 1 1								

In Fig. 11, the *C* networks are made up of contacts from the primary relays and of contacts from the β-group of secondary relays. Now we shall investigate the construction of a typical C network. The motives behind this construction will be evident when circuit action is described later.

Each position in the flow table of (*a*) in Table XXVII corresponds to a fixed combination of $y_\beta = (y_{\beta 1}, y_{\beta 2}, y_{\beta 3})$ and of $x = (x_1, x_2, x_3)$. The matrix for each C network will be in terms of combinations of these values of y_β and x. In (*b*) of Table XXVII we have derived the entries in the matrix representation for the C_1 network. This network will be used later as a building block in forming the controlling network for the $Y_{\alpha 1}$ relay and also for the network which controls the Y_0 relay. (In general, the two-terminal network C_j is used in the control networks for the $Y_{\alpha j}$ and Y_0 relays.)

Suppose, for instance, that we wish to find the proper entry in the (110; 011) position of the C_1 matrix. We should first look at the corresponding position in the flow table. It contains an uncircled "7." Notice now that the *circled* "7" is in a row corresponding to $y_{\alpha 1} = 1$. ($y_{\alpha 1}$ is the first component of y.) And so we make the entry in the (110; 011) position of the C_1 matrix the digit "1." If we continue to fill in the C_1 matrix by inserting these ultimate values of $y_{\alpha 1}$, the result is (*b*) of Table XXVII. (The last row of the matrix may be completed in any arbitrary fashion.)

Now that we have given the rule for construction of the C networks in Fig. 11, we shall soon see the reason for specifying the networks in this way. Whenever the circuit is in a stable condition, our method of formation of the C networks assures that $C_1 = y_{\alpha 1}$, $C_2 = y_{\alpha 2}$, and $C_3 = y_{\alpha 3}$. And also, for any stable circuit condition, $y_{\alpha 1} = y_{\beta 1}$, $y_{\alpha 2} = y_{\beta 2}$, and $y_{\alpha 3} = y_{\beta 3}$. These restrictions, applied to the typical physical circuit configuration of Fig. 11, tell us that each of the seven secondary relays is in a stable state. Let us investigate now what happens as we change the circuit from condition "3" to condition "21." The starting point for our discussion of circuit changes is the stable circuit condition "3" for which

$$\begin{array}{llll} C_1 = 0, & C_2 = 0, & C_3 = 1, & \\ y_{\alpha 1} = 0, & y_{\alpha 2} = 0, & y_{\alpha 3} = 1, & \\ y_{\beta 1} = 0, & y_{\beta 2} = 0, & y_{\beta 3} = 1, & \text{and} \quad y_0 = 0. \end{array}$$

The flow table in (*a*) of Table XXVII indicates that a change of input state from "010" to "000" should result in the ultimate circuit condition "21." Of particular interest to us is the action of the various groups of secondary relays.

The method of building the C networks was such that, whenever the input state is modified, these C networks immediately assume transmissions corresponding to ultimate states of the relays in the α-group. In our example this means that change of the input state from "010" to "000" modifies the C network transmissions to $C_1 = 1$, $C_2 = 1$, and

$C_3 = 0$. These correspond to the value of $y_\alpha = 110$, which is associated with the circuit condition "21." We now have

$$\begin{array}{llll} C_1 = 1, & C_2 = 1, & C_3 = 0, & \\ y_{\alpha 1} = 0, & y_{\alpha 2} = 0, & y_{\alpha 3} = 1, & \\ y_{\beta 1} = 0, & y_{\beta 2} = 0, & y_{\beta 3} = 1, & \text{and} \quad y_0 = 0. \end{array}$$

This change in the transmission of the C networks assures that the relays in the α-group are now energized according to their ultimate states of operation (see Fig. 11). It can be observed that the relay Y_0, and the relays in the β-group are still in stable states.

After a time, all the relays in the α-group have reached their ultimate states of operation. Then

$$\begin{array}{llll} C_1 = 1, & C_2 = 1, & C_3 = 0, & \\ y_{\alpha 1} = 1, & y_{\alpha 2} = 1, & y_{\alpha 3} = 0, & \\ y_{\beta 1} = 0, & y_{\beta 2} = 0, & y_{\beta 3} = 1, & \text{and} \quad y_0 = 0. \end{array}$$

Y_0 becomes energized (see the controlling network for Y_0 in Fig. 11) just as soon as corresponding values of the transmissions of the C networks and the states of operation of the α-relays become equal—*and not before.* As soon as Y_0 becomes operated,

$$\begin{array}{llll} C_1 = 1, & C_2 = 1, & C_3 = 0, & \\ y_{\alpha 1} = 1, & y_{\alpha 2} = 1, & y_{\alpha 3} = 0, & \\ y_{\beta 1} = 0, & y_{\beta 2} = 0, & y_{\beta 3} = 1, & \text{and} \quad y_0 = 1. \end{array}$$

Operation of Y_0 does not change the stability of the relays in the α-group. But just as soon as—*and not before*—Y_0 is operated, the β-relays become unstable. Eventually these β-relays will become operated in the same way as their counterparts in the α-group. Then

$$\begin{array}{llll} C_1 = 1, & C_2 = 1, & C_3 = 0, & \\ y_{\alpha 1} = 1, & y_{\alpha 2} = 1, & y_{\alpha 3} = 0, & \\ y_{\beta 1} = 1, & y_{\beta 2} = 1, & y_{\beta 3} = 0, & \text{and} \quad y_0 = 1. \end{array}$$

During the time before the states of operation of the β-relays match up with the corresponding states of operation of the α-relays, the transmissions of the various C networks will be changing, since they are functions of the transmissions of contacts on the β-group of relays. During this time the Y_0 relay is kept operated by the path to ground through its own y_0 contact. However, as soon as all β-relays are operated in the same pattern as the α-relays, Y_0 becomes de-energized and eventually unoperated. Now

$$\begin{array}{llll} C_1 = 1, & C_2 = 1, & C_3 = 0, & \\ y_{\alpha 1} = 1, & y_{\alpha 2} = 1, & y_{\alpha 3} = 0, & \\ y_{\beta 1} = 1, & y_{\beta 2} = 1, & y_{\beta 3} = 0, & \text{and} \quad y_0 = 0. \end{array}$$

All secondary relays are again in stable states and again $y_\alpha = y_\beta$ and $y_0 = 0$. The circuit is at last in condition "21" and this was the desired result.

The changes in secondary relay state which have been described in the preceding paragraphs can be summed up in the briefer graphical form in Table XXVIII. The heavy arrows indicate where race conditions exist, but we have demonstrated that these are not critical.

TABLE XXVIII.—*Secondary Relay Actions.*

y:	$(y_\alpha$,	y_β,	$y_0)$
	001	001	0
	↓	·	·
		·	·
	110	001	0 ⤵
	110	001	1 ↙
	·	↓	·
	·		·
	110	110	1 ⤵
	110	110	0 ↙

The control network configurations in the circuit of Fig. 11 are characteristic of the "$2s_0 + 1$" realization. So also are the actions of the groups of secondary relays given in Table XXVIII. To sum up:

When the circuit is stable, then an input-state modification which leads to a circuit condition which is represented in a new row of the flow table produces secondary relay action, as given below.

1. The α-relays change to the "new" states.
2. The Y_0 relay operates.
3. The β-relays change to the "new" states.
4. The Y_0 relay releases again to its unoperated state.

The "One-Relay-Per-Row" Realization

An interesting realization of a sequential circuit results if each row of the flow table is associated with its own relay. Thus, for a table with n rows, there will be n secondary relays. For each stable circuit condition represented in the ith row of the table, the ith (and only the ith) secondary relay is to be operated. For example, we might assign secondary relay states to the flow table shown in Table XXIX.

We wish to specify controlling networks for the secondary relays so that when a stable circuit condition represented in the ith row of the table (and associated, therefore, with the ith secondary relay) is followed by a stable condition in the jth row, then the only intermediate secondary relay state is to be one in which both the ith and the jth (and only these) secondary relays are operated. For instance—in Table XXIX—if we want to pass from condition "3" to condition "21," the following secondary relay action will occur:

1. If the circuit is in condition "3," the total relay state is (010; 0100000).
2. If the input state is now changed to "000," then the total relay state immediately changes to (000; 0100000).
3. We wish the state in (2) above to be followed by the state (000; 0100001).
4. The next, and final, total relay state is to be (000; 0000001).

TABLE XXIX.—*A Flow Table.*

y: \ x:	000	001	010	011	100	101	110	111
1000000	①	10	㉙	⑫	13	㉖	⑱	16
0100000	21	㉘	③	17	㉓	⑭	18	⑧
0010000	9	②	⑪	12	23	14	⑮	㉔
0001000	⑨	⑩	3	④	13	⑲	7	24
0000100	1	⑳	11	㉕	⑤	19	㉚	8
0000010	9	2	㉒	⑰	5	6	⑦	⑯
0000001	㉑	20	22	4	⑬	⑥	15	㉗

In order to illustrate the general rule for formation of the secondary relay controlling networks, we shall design these networks for the second and seventh secondary relays; these are the relays associated with the circuit conditions "3" and "21," respectively.

The controlling networks for each of the secondary relays may be considered to be in the general form shown in Fig. 12. If the relay Y_i is not operated, then the closing of the A network will result in its operation. On the other hand, if Y_i is operated and the A network is open, then opening the B network will result in the release of the relay to its unoperated state.

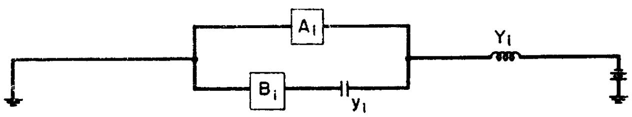


FIG. 12. Illustrating the design of secondary relay control networks.

Let us now form the network Y_2. The A network is to be closed whenever any one of the following statements is true:

1. The input state is "010" and the fourth relay is operated.
2. The input state is "100" and the third relay is operated.
3. The input state is "101" and the third relay is operated.
4. The input state is "111" and the fifth relay is operated.

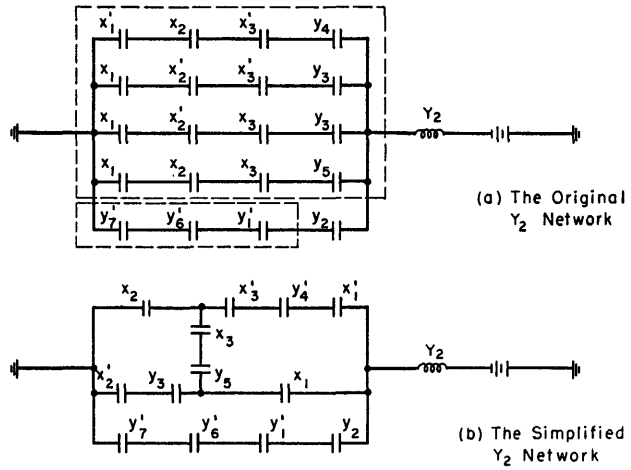


FIG. 13. Design of the Y_2 network.

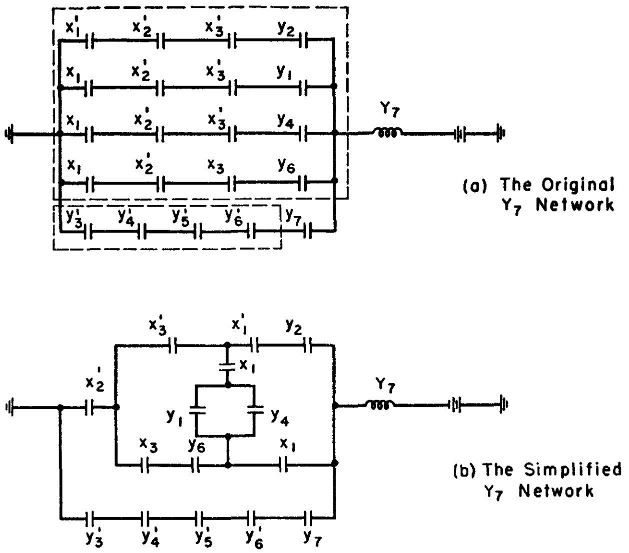


FIG. 14. Design of the Y_7 network.

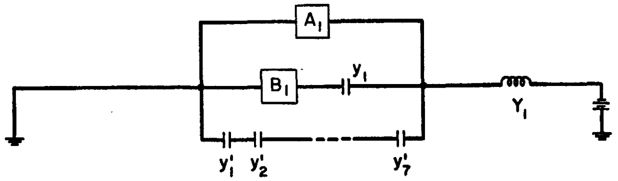


FIG. 15. A modified control network for the Y_1 relay.

Each condition above is one which ultimately leads to a stable circuit condition represented in the second row of the table. The circuit realization of the conditions above is given in the upper network of Fig. 13(*a*).

The B network for the second secondary relay is to be one which opens whenever other secondary relays become operated. However, inspection of the flow table shows that the only rows which may be entered from circled entries in the second row are:

(1) the seventh; (2) the sixth; (3) the first.

The circuit realization which satisfies the requirements above is given in the lower network of Fig. 13(*a*).

In Fig. 13(*b*) is a simplified terminally-equivalent version of the Y_2 network which is obtained from Fig. 13(*a*).

In Fig. 14(*a*) and (*b*) are given the original and the simplified control networks for the Y_7 relay.

One important practical problem would arise in the operation of the circuit which we have designed above. Assume that the supply voltage for the secondary relays has been disconnected from these relays—as it would be during a shutdown period. Then each secondary relay is unoperated and therefore $y_1 = y_2 = \cdots = y_7 = 0$. Examination of the networks controlling the secondary relays (those of Figs. 13(*a*) and 14(*a*) are typical) would reveal that the transmission of none of these networks can become unity until one of the secondary relays becomes operated. But none of these relays, in turn, can become operated (or even energized) until transmission of its controlling network becomes unity. These facts lead us to the conclusion that, as the circuit stands, proper operation cannot occur if the power is once turned off. To remedy this defect in the circuit, we may add to one of the secondary relay control networks (say Y_1) a parallel branch consisting of a cascade connection of normally closed contacts from each of the secondary relays (see Fig. 15). Now when the voltage supply is connected to the relays, Y_1 will operate initially and the ensuing circuit action can be that specified by the flow table.

There are two most interesting characteristics of a circuit derived in the manner above:

1. The maximum time necessary for transfer from one stable circuit condition to another is equal to the release time for the "old" relay, plus the operating time of the "new" relay. (In our example above, Y_2 is the "old" relay and Y_7 the "new" relay.)

2. Since only one secondary relay is operated when the circuit is stable, and since the greatest number of relays ever operated is two (and this is only a transient condition), then there is always very low current drain from the voltage supply.

VII. SUMMARY OF THE SYNTHESIS PROCEDURE

In the preceding sections we have developed a number of attitudes and procedures which have been useful in formalizing the steps of our synthesis procedure. We have demonstrated that the problem of designing a relay switching circuit with sequential and ultimately stable terminal action may be reduced to the equivalent problem of designing several combinational-type contact networks. The steps which we advise taking in the synthesis of a sequential relay circuit are these:

1. Make a statement of the problem in terms of a flow table in such a way that the specifications of the desired circuit action are exhaustively listed.
2. Investigate the conciseness of the problem statement and the possibility of simplification. We have proved that, for a flow table with "uniform vacancy" within each input-output set, the ultimate simplification will be unique.
3. Determine possible row mergers in the flow table so that the table may be shortened in length. We have proved that, for a circuit in which any input transition is possible, the form of the minimum-row condensed form of the flow table will be unique.
4. Assign secondary relay states to the rows of the flow table in such a way that all k-sets are connected. To make this assignment, it may be necessary to augment the flow table by additional uncircled entries. If desired, we may use the operation of row splitting as a supplementary tool.
5. Develop a τ matrix so as to realize the transitions indicated by the flow table. If we have made each k-set connected, the existence of at least one proper τ matrix is guaranteed.
6. Derive, by established procedure, a Y matrix corresponding to the τ matrix.
7. Place the output data in the Z matrix. If we wish the ultimate output state to appear as soon as possible after the input state is changed, we insert in each position of the Z matrix the output state associated with the corresponding numerical designator in the flow table, whether this designator be circled or not.
8. List the information in the Y and Z matrices in tables of combinations for the use of the designer of the static contact networks. With the initial assumptions we have made about the characteristics of the component relays, these secondary relay control networks and output networks will force the circuit to operate with the action specified by the original flow table.

In a particularly complex synthesis problem, we may find it impossible to solve by inspection the problem of assigning secondary relay states to the rows of the flow table; or we may wish to take advantage of the properties peculiar to one of the "standard" circuit realizations.

In this case, circuit realization may be made in terms of the "$2s_0 + 1$" circuit or of the "one-relay-per-row" circuit.

VIII. CRITIQUE OF MATERIAL ON SEQUENTIAL RELAY CIRCUITS, WITH EMPHASIS ON THE ROLE OF THE SECONDARY RELAY

The switching circuits we have considered in the preceding sections may be affected at any given time by some combination of several binary input variables, and these circuits may in turn act upon their surroundings by some combination of several binary output variables. Our main efforts have been to show that the operating requirements of a sequential (dynamic) switching circuit may be restated as a set of requirements for circuits with combinational (static) characteristics. These sequential requirements may be stated conveniently in a flow table.

The switching devices which we have used in the circuits of the preceding chapters have been relays. Some of these (the primary relays) are under the direct control of the input; the excitation of the others (the secondary relays) is a function of the states of operation of the various secondary relays as well as of the input state. In fact, we could *define* a secondary relay as one which is not under the direct influence of the input.

The output state of a sequential relay circuit may be a function of the states of operation of both the primary and the secondary relays. If there were no secondary relays, then the output would necessarily be a function of the input state only, and the circuit would not be a sequential one, but a combinational one. And so we may say that in a sequential circuit there must necessarily be devices (one or more) which are not under the direct control of the input. What is the function of these secondary devices?

Under the assumptions that we have made for the secondary relays —that all normally open contacts are either open or closed at the same time, and that the transmissions of all the normally closed contacts are always complementary to those of the normally open contacts—then only one property of such a relay seems to be necessary. This is: that a *time lag* should occur between its state of excitation (energization) and its state of response (operation). If we were to rob the secondary relays of this property, we could never put one into an unstable state, and there could never be step-by-step transitions between the various internal circuit configurations. We may say that, from a theoretical point of view, the time lag between the excitation and the response of the various secondary relays is a property *sufficient* to insure proper operation of the sequential relay circuits. This statement is evident if one agrees to the validity of the individual steps taken in the synthesis proposed in the earlier sections of this paper.

From an operational point of view, the relay circuit designer knows that the assumption of complementary transmission for the normally open and normally closed contacts on a relay is not altogether valid. If the "complementary transmission" assumption is not valid, then it may be that the transmission of the contact network which excites some secondary relay, Y_i, may change value several times before arriving at the transmission dictated by the existing total relay state. This non-ideal situation creates what the relay circuit designer knows as an operating *hazard*. Such a hazard would be potentially undesirable if the response time (the operate time or the release time) of the relay Y_i were smaller than the time required for the transmission of the controlling network to settle down at its proper value. For then the response of Y_i would not be that intended by the designer, and improper operation of the over-all circuit might result.

Fortunately a finite amount of time (the response time) is necessary before the amount of energy in the magnetic field of a relay changes enough to change the state of operation of the relay. This *smoothing* effect tends to decrease the importance of an operating hazard. To eliminate the importance of a hazard altogether, the smoothing effect should be large enough that the response time of each secondary relay is greater than the *hazard time* of its controlling network.

Another property which is inherent in each relay may be termed that of *requantization*. By this term we mean to emphasize that the ultimate response of a relay is two-valued (because of the two-valued nature of the transmissions of its contacts), even though the magnitude of its magnetic field changes continuously in accordance with the "smoothed" excitation. This requantization property prevents corruption of the ground signals as they are handled by the secondary relays.

We may state the following conclusions:

1. Ideally, it is necessary and sufficient that there be a non-zero time between the excitation and response of a secondary relay.
2. Practically, it is important that a secondary relay also have the properties of excitation-smoothing and of requantization of this smoothed excitation as a two-valued output.

If we were to endow the secondary relay with human attributes, we could say that each one is a *decision-making* element. After each change in stimulation from its environment, it waits a time to "decide" what the final stimulation is to be, and then reacts accordingly. The operating "hazard" occurs when this "yes-no" decision is based on observation over an insufficient length of time.

IX. THE SYNTHESIS OF ELECTRONIC SWITCHING CIRCUITS

Electronic switching circuits have been more and more used in recent years, particularly in applications such as calculating machinery, where their speed of operation and lack of moving parts place them at a distinct advantage over relay switching circuits. The underlying principles of electronic switching circuits are, however, the same as those we investigated in relay circuits, and these similarities will be pointed out in this section.

Combinational and Sequential Circuits

An electronic switching circuit may be combinational or sequential in nature just as a relay circuit is. A "combinational" network will refer to an electronic circuit which has several input leads upon which various combinations of "high" or "low" voltages may be impressed, and one or more output leads upon which, at a given time, either a "high" or "low" voltage appears. For a combinational network the functional dependence of an output voltage on the input voltages may be expressed by the entries in a table of combinations. (See, for example, Fig. 16.) On the other hand, a flow table will be used to describe the terminal characteristics of a sequential electronic switching circuit.

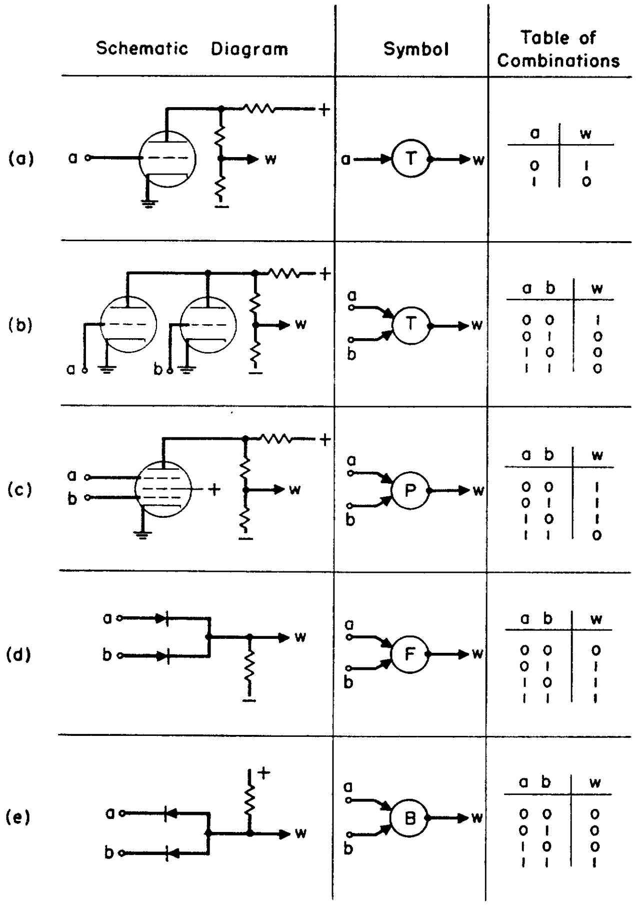


FIG. 16. Basic electronic combinational networks.

There is a variety of ways of giving an electronic realization to the requirements listed in a table of combinations. We will limit ourselves to circuits using standard vacuum-tubes (triodes and pentodes) and simple rectifiers. In Fig. 16 are listed five of the most used configurations of vacuum-tubes and rectifiers along with their associated tables of combinations and the symbols which will be used to represent them in this paper. These basic circuits and others are commonly used as building blocks for more complex networks. It is to be expected that, as research on them progresses, other devices, such as transistors and special-purpose switching tubes, will be utilized in building combinational networks.

The variables used in describing electronic switching circuits represent voltages. Such a voltage may be either "high"—in which case the variable is assigned a value of "1"—or "low," and then the variable is given the value "0." (Just how high "high" is and how low "low" is will be discussed below.)

The single triode is the most common way of complementing a two-valued voltage. When its input voltage is low (low enough to cut off the plate current) the output voltage is high, and when the input voltage is high (high enough to cause appreciable plate current to flow) the output voltage is low.

For the common-plate triode pair the output voltage is high only if both input voltages are low. For the pentode the output voltage is low only if both inputs are high.

The "forward" rectifier connection has the property that its output voltage is low only if both input voltages are low (lower than the voltage to which the associated resistor is returned). For the "backward" rectifier connection a high output results only when both inputs are high (higher than the voltage to which the associated resistor is returned).

There have been several papers written about the realization of combinational type networks using tubes and rectifiers. The book by Professor H. H. Aiken and his associates at Harvard University[17] represents the most notable original contribution in this field. The author cannot attempt to improve upon the excellent material presented there.

In this same book there appears, under the heading of "Triggers, Rings, and Digit Counters" (Chapter 8), an approach to the sequential circuit problem. This approach cannot be satisfying to this author because it fails to take into account the consequences of excitation and response not being identical, either physically or algebraically. Let us emphasize again that the fact of instability in a switching device arises from the distinction which we must make between excitation and response, and a sequential circuit can be designed properly only by the successful control of this instability.

[17] Staff of the Computation Laboratory, "Synthesis of Electronic Computing and Control Circuits," Cambridge, Harvard University Press, 1951.

Just as in our previous work with relay circuits our synthesis method for electronic switching circuits consists of a listing of the circuit requirements in a flow table and then a reduction of these requirements to a set of specifications for several combinational circuits. All of the techniques for manipulation of flow tables, assignment of secondary response states, and derivation of the τ, Y, and Z matrices are just as valid for electronic circuits as they are for relay circuits. It is only the physical realization of the Y and Z matrices that is different for the two types of circuits. The simple example of the next section will illustrate the underlying similarities and the physical differences involved.

The Synthesis Procedure Applied to a Simple Problem

Let us synthesize an electronic switching circuit which will have two input leads and two output leads. Assume that the voltages on the two input leads may not change simultaneously. The requirements (listed in the flow table of Table XXX(a)) for the circuit are these: A low voltage is to appear on both output leads if either of the input voltages is low (conditions "1," "2," and "3"). A high voltage is to appear on the Z_1 output lead (condition "4") only if both input voltages are high and if, just previously, only the X_1 input voltage was high. Similarly, the Z_2 output voltage is to be high only if both input voltages are high and if, just previously, only the X_2 input voltage was high.

The flow table listing these requirements may be condensed as shown

TABLE XXX.

(a) Primitive Flow Table

x: 00	01	10	11	z:
①	2	3		00
1	②		5	00
1		③	4	00
	2	3	④	10
	2	3	⑤	01

(b) Condensed Flow Table

x: 00	01	10	11
①	2	③	④
1	②	3	⑤

(c) τ Matrix

y: \ x:	00	01	10	11
0	0	1	0	0
1	1	0	1	0

(d) Y Matrix

y: \ x:	00	01	10	11
0	0	1	0	0
1	0	1	0	1

(e) Z Matrix

y: \ x:	00	01	10	11
0	00	00	00	10
1	00	00	00	01

in Table XXX(*b*). The secondary assignment and derivation of the τ, Y, and Z matrices are straightforward and are the same as if the synthesis were of a relay circuit. The final result of the synthesis procedure is the listing of the requirements for three electronic combinational networks (the Y, Z_1, and Z_2 networks). One possible circuit utilizing networks designed from these requirements is given in Fig. 19. Intermediate steps in its derivation are given in Figs. 17(*a*) and 18(*a*). These intermediate steps will be explained below.

Generation of Secondary Excitation and Response Voltages

Let us concentrate our attention on the Y excitation network. There are a variety of ways in which it may be designed, and four of these are given within the dotted-line limits of the symbolic diagrams of Fig. 17. The design of these combinational networks is an extensive subject in itself and the reader is referred to the Harvard University text for further information on it.

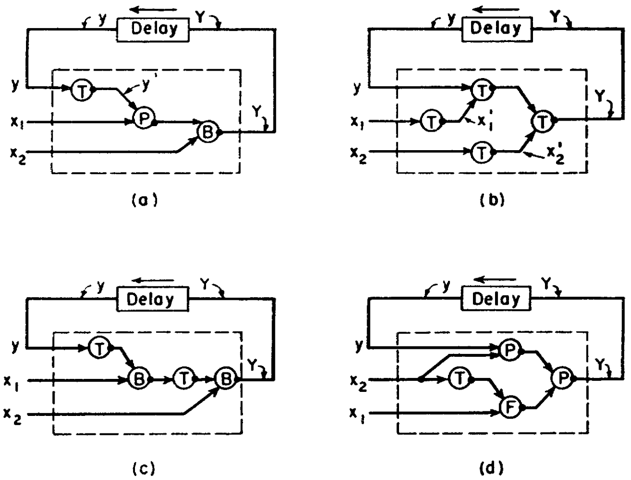


FIG. 17. Symbolic diagrams of four equivalent networks.

Notice that the secondary excitation voltage, Y, is produced by (is a function of) the three voltages x_1, x_2, and y. Notice also that the response voltage, y, is made to lag the excitation voltage, Y, and that the connection which accomplishes this delay also creates a sort of feedback loop in the circuit. (It may be proved that, for proper circuit action, the voltage gain around the loop created must be greater than unity.) Usually it is not necessary to add the delay indicated, because some delay already exists around the loop due to distributed capacity, lead inductance, and the electron transit times within the vacuum tubes. However our analysis shows exactly at what point delay should be added if this addition becomes necessary.

It is clear from the diagrams of Fig. 17 that in electronic circuits designed by our synthesis methods there may not be a *device* which is analogous to the secondary relay of preceding sections. Instead, as in our present example, we may have to think about *secondary delay locations*. Thus, for example, the number s_0, which has been defined earlier in this paper,[18] will refer in electronic circuits to the minimum conceivable number of secondary delay locations necessary to realize a given flow table; at these delay locations there may not be a switching device, and perhaps it may not even be necessary to add additional delay at such points.

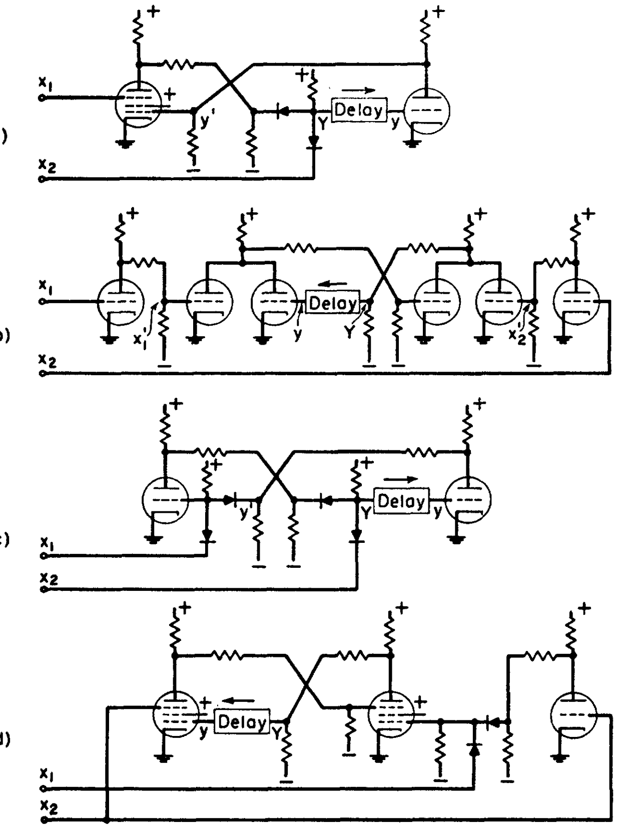


FIG. 18. Schematic diagrams of four equivalent networks.

It is true that when feedback loops are added in the symbolic diagram they—in conjunction with the associated combinational networks—often produce interconnections of rectifiers and vacuum tubes which may be recognized as familiar. For example, observe the schematic diagrams of Fig. 18. (These correspond to the symbolic diagrams of

[18] See *On the Number of Secondary Relays Required,* page 275 of this paper.

Fig. 17.) The two center triodes of Fig. 18(*b*) are seen to be connected as an Eccles-Jordan circuit. The adjacent triodes have often been called "puller" tubes. And, in Fig. 18(*c*) there appears a modification of the Eccles-Jordan circuit.

Let us emphasize here that the familiar-looking circuits of Fig. 18 have been *derived* as a *by-product* of our synthesis procedure; they were not used as a starting point for it. What should concern us is that correctly designed *combinational* networks will insure proper operation of the total electronic switching circuit. The derivation of "trick" circuits—even though interesting, and perhaps surprising—should be considered as an incidental matter.

Let us return briefly to the synthesis of our circuit. If we use the circuit of Fig. 18(*a*) to generate the secondary voltage, y (and its complement, y'), then additional rectifiers may be added in order to give the output voltages, Z_1 and Z_2. The final circuit, with the two Z networks added, is given in Fig. 19.

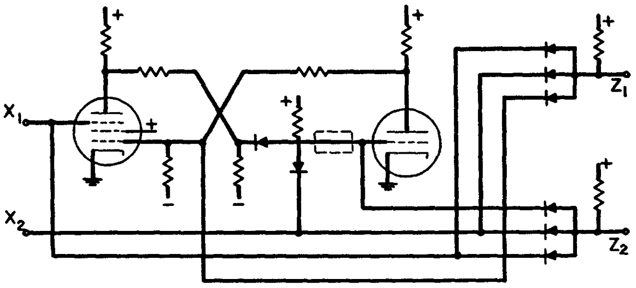


FIG. 19. The derived electronic switching circuit.

Comments on Treatment of the Eccles-Jordan Circuit

Since the Eccles-Jordan circuit is a commonly-used one, it may be of interest to examine whether or not the one of Fig. 18(*b*) is being used in the conventional manner. Let us consider what happens when both the X_1 and X_2 voltages are low. (This actually happens for circuit condition "1" listed in Table XXX(*a*).) Examination of the circuit diagram will show that, for these input voltages, the plate voltages on both of the two inner triodes (the Eccles-Jordan triodes) are low. This certainly is not the way that the Eccles-Jordan circuit is usually operated, but we see here that it is one possible (and here, useful) mode of operation.

Many of the algebraic inaccuracies this author believes exist in other methods which have been proposed for the treatment of Eccles-Jordan circuits arise directly from failure to recognize that the plate voltages of such a circuit are *not* algebraically complementary, since the possibility exists for both triodes of the pair to be conducting simultaneously.

Hazards

Since we have established a direct analogy between relay circuits and electronic circuits, it is natural to wonder what, in electronic circuits, is related to the "hazard" discussed for relay circuits. (See Section VIII.) It will be remembered that the possibility that a secondary excitation network might change its transmission several times before coming to the value dictated by the total relay state was termed a hazard. For electronic circuits the possibility that a secondary excitation network may give an excitation voltage which may change level (low to high, or high to low) several times before coming to the level dictated by the total circuit state will also, by analogy, be called a hazard.

For example, in Fig. 17(*d*) the voltage X_2 may be processed by the electronic devices along two separate paths before finally affecting the excitation voltage, *Y*. If the delays along these two paths are not the same, then the possibility for a hazard exists. In general, the existence of two or more parallel paths over which signals are processed in a combinational network may introduce a hazard.

As we consider electronic switching circuits of greater and greater complexity it is easy to visualize that the possibility of hazards in the various secondary excitation networks becomes increasingly serious. If we are to continue the analogy with relay switching circuits we should now ask ourselves: "How can the functions of smoothing and requantization be best accomplished at the secondary delay locations?"

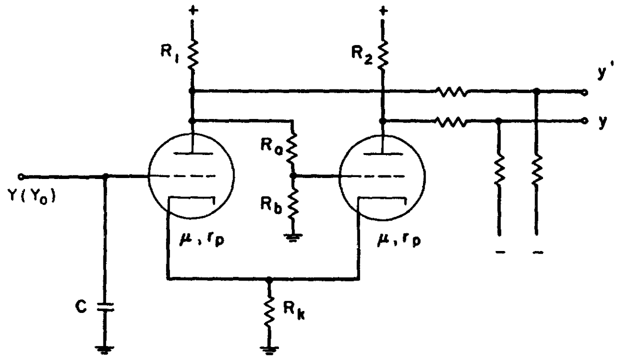


Fig. 20. A smoothing and requantizing device.

Elimination of the Effect of Hazards

A sufficiently large capacity will certainly accomplish the necessary smoothing if it is connected at the delay location. However it will also tend to corrupt the two-valued nature of the secondary response voltage there. On the other hand, it is easy to use a simple direct-coupled amplifier to accomplish the requantization of the secondary voltage but such a simple amplifier has practically no smoothing properties. If we use a capacity in conjunction with the direct-coupled amplifier, an increase in the value of capacity required for smoothing makes it necessary that the gain of the amplifier also increase, in order to preserve the requantization property. What is needed is an amplifier with an effectively infinite gain and with a terminal at which the smoothing capacity may be connected without affecting this gain.

One relatively simple circuit which will have the properties needed is shown in Fig. 20. If the quantity

$$(R_1 + r_p)(R_2 + r_p) + R_k(\mu + 1)[R_1 + R_2 + 2r_p - \mu R_1 R_b/(R_a + R_b)]$$

is positive the circuit will have an infinite gain. (R_1 and R_2 represent the total resistive loading at each plate.) The circuit has the property that, for all practical purposes, the voltage at each plate is either "high" or "low" and does not assume intermediate values. If the voltage at the input grid varies continuously from a low value to a high value, at some critical intermediate voltage the voltage at the plate of the opposite triode will suddenly change from low to high. As the grid voltage decreases from its high value to its low value the same plate voltage will jump again to its low value.

If the input voltage to this device is some excitation voltage, Y, which has been smoothed by the capacity at the input grid, then the plate voltage described above can be used as the requantized response voltage, y. Its complement, y', appears at the other plate terminal.

In use in complicated switching circuits such a requantizing circuit would replace the delay lines at the secondary delay locations where hazards exist. The smoothing at each of these locations could be made as great as necessary to eliminate the effect of hazards by adding sufficient capacity at the input grid of the device.

Completion of the Analogy

In case this requantizing device (or some similar one) is used at each secondary delay location we will have a circuit completely analogous to one using relays. The voltage (high or low) from the electronic excitation network corresponds to the transmission (unity or zero) of the contact network exciting a relay; the smoothed excitation voltage corresponds to the magnetic field of the relay; the two complementary requantized response voltages correspond to the complementary transmissions of the normally open and normally closed contacts on the relay.

Acknowledgment

The author wishes to acknowledge his indebtedness to Professor S. H. Caldwell and to the students in the switching circuits course at the Massachusetts Institute of Technology for the willingness of their ears while the ideas presented in this paper were still in the rudimentary stage.