DOCUMENT OFFICE 36-412
RESEARCH LABORATORY OF ELECTRONICS
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

[redacted]

#2

A STUDY OF ASYNCHRONOUS LOGICAL FEEDBACK NETWORKS

STEPHEN H. UNGER

LOAN Copy Only

TECHNICAL REPORT 320

APRIL 26, 1957

RESEARCH LABORATORY OF ELECTRONICS
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS

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 U. S. Army (Signal Corps), the U. S. Navy (Office of Naval Research), and the U. S. Air Force (Office of Scientific Research, Air Research and Development Command), under Signal Corps Contract DA36-039-sc-64637, Department of the Army Task 3-99-06-108 and Project 3-99-00-100.

RESEARCH LABORATORY OF ELECTRONICS

Technical Report 320 April 26, 1957

A STUDY OF ASYNCHRONOUS LOGICAL FEEDBACK NETWORKS

Stephen H. Unger

This report is based on a thesis submitted to the Department of Electrical Engineering, M.I.T., in partial fulfillment of the requirements for the degree of Doctor of Science, May 1, 1957.

Abstract

The systems considered are asynchronous, dc-level, sequential switching circuits; that is, there are no clock pulses, and the signals are represented, not by pulses, but by variables that can assume values in either of two nonoverlapping ranges. The analysis of these circuits is discussed, with emphasis on the problem of choosing state variables.

A relationship has been established between the number of rows of a reduced flow matrix and the feedback index of the associated circuit and, furthermore, the need for amplification around each feedback loop has been proved.

The major portion of the research is devoted to a study of the effects of stray delays on the operation of sequential switching circuits. By inserting delay elements in all feedback loops, and using hazard-free combinational circuits, correct operation can be obtained despite the stray delays. It has been shown that, for a subset of all sequential functions, delay elements need not be used if certain design procedures are adhered to. All other functions require delay elements if the possibility of hazards (malfunctioning caused by stray delays) is to be averted, but methods are presented by means of which a single delay element is sufficient for the synthesis of any sequential function. These procedures usually entail more complex circuitry than would otherwise be needed, but several other techniques are presented that may, in certain cases, be useful in minimizing the number of delay elements at a relatively small cost in terms of other components.

I. INTRODUCTION

In this section we shall describe the type of system that will be considered, present some of the nomenclature that will be used, and outline the scope of the research and the principal results that were obtained.

Figure 1 represents the type of system with which we are concerned. The signals at the input and output terminals can take on either of two values (designated zero and one), and changes in input signals may be made at any time. (Actually there must be a minimum time interval between successive input changes, the length of this interval being determined by the reaction times of the components.) Note that these systems are asynchronous in the sense that there are no clock pulses, time is not quantized and signals are represented by dc levels - not by pulses.

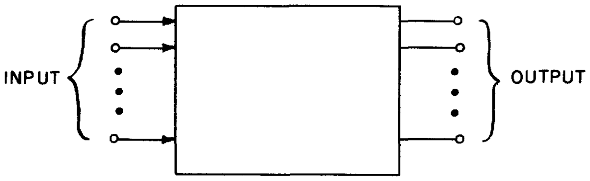


Fig. 1. Sequential switching circuit.

The set of signals present at the input terminals are referred to collectively as the "input state" (for brevity, the "input") and the term "output state" (or "output") refers to the set of output signals. If the output state is always uniquely determined by the present input state, then the system is called a "combinational circuit." A more general situation occurs in the case of a "sequential circuit" in which the output state is a function of past as well as of present inputs. We shall refer to input-output relationships for sequential switching circuits as "sequential functions."

Although it is assumed that the reader is generally familiar with the basic concepts of combinational and sequential switching circuits (see refs. 1-7), a recapitulation of some of the important concepts will be presented for orientation purposes, and for introducing some notation.

Combinational switching functions are specified by means of tables of combinations (truth tables) and sequential functions can be expressed in the form of flow tables.

Figure 2 illustrates the flow-table form that we shall use. The columns correspond to input states and the rows represent internal states of the network, so that each cell in the table corresponds to a total state of the system. Output states are indicated by the entries in the upper right-hand corners of the cells, and the number in the center of each cell indicates what the next internal state will be. When the next internal state is the same as the present, the state number is circled to emphasize that the system is in a stable state.

	INPUT x_1, x_2			
INTERNAL STATE	00	01	11	10
1	00 ① "a"	10 4 "b"	01 2	00 ①
2	11 3	11 3	01 ②	01 ②
3	11 ③	11 ③	00 4	01 2
4	11 3	10 ④ "c"	00 ④	00 1

Fig. 2. Flow table.

We are concerned only with functions whose unstable states lead immediately to stable states. Figure 2 represents such a function. For example, if the total state is "a" and if x_2 is changed from zero to one, the state following "a" will be "b", an unstable state. The entry in "b" indicates that the next internal state will be 4, and so the next total state will be "c", which is stable.

Sequential functions can be realized by systems of the form shown in Fig. 3, which are essentially combinational circuits with some of the output signals fed back into some of the input terminals. The part of each of these feedback paths that is external to the combinational network is called a "state branch." As we shall see later, it is often necessary to introduce delay elements into some or all of the state branches.

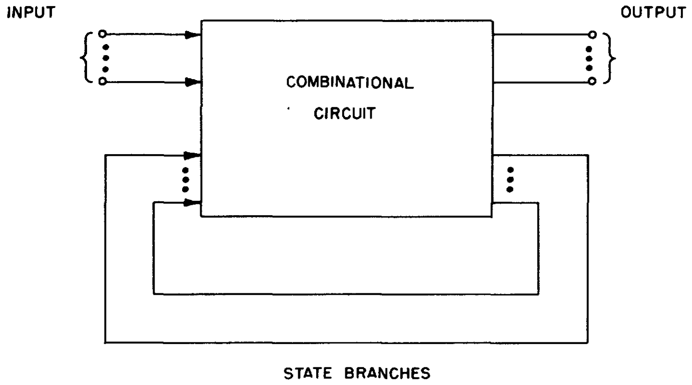
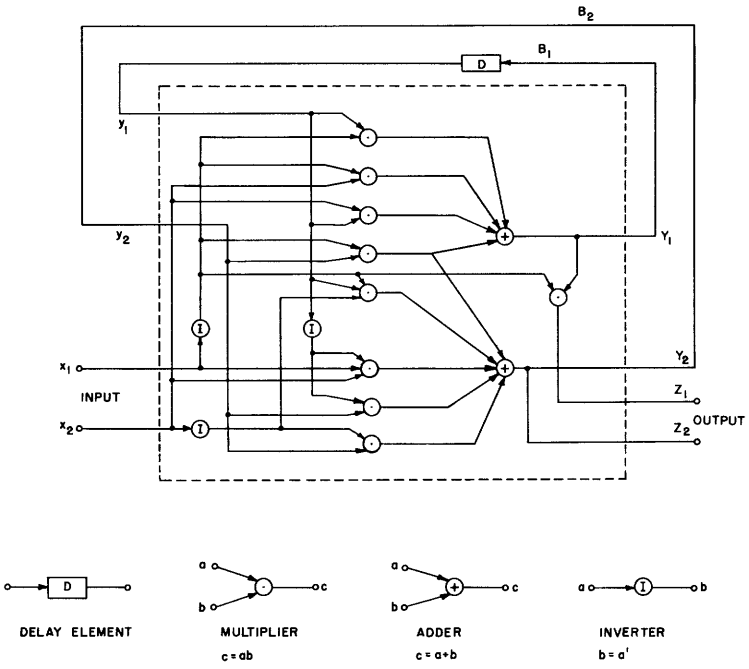


Fig. 3. Form of sequential switching circuit.

The terms "state-branch excitation" (designated by Y_i for the i^{th} branch) and "state-branch response" (designated by y_i) are used to refer to signals entering and leaving the state branches. The y-signals are collectively associated with the internal states of the system. In other words, to each row of the flow table for that network, there corresponds at least one state of the state-branch responses.

The combinational circuits are networks of elementary logical devices such as logical adders, logical multipliers, modulo-two adders, inverters, and so forth. The physical means for realizing these devices will not concern us here.

Figure 4a is a functional diagram of a sequential circuit that realizes the flow table of Fig. 2. The portion of the circuit enclosed in the dotted box is the combinational part, and a delay element appears in state branch B_1. The correspondence between rows of the flow table and y-states is shown in Fig. 4b, which is called a "flow matrix." The reader may verify the fact that the operation of the circuit depicted in the functional diagram is correctly described by the flow matrix.



(a)

INPUT x_1 x_2

INTERNAL STATE	00	01	11	10	y_1 y_2
1	00 ①	10 4	01 2	00 ①	00
2	11 3	11 3	01 ②	01 ②	01
3	11 ③	11 ③	00 4	01 2	11
4	11 3	10 ④	00 ④	00 1	10

(b)

Fig. 4. (a) Functional diagram of sequential switching circuit.
(b) Flow matrix for circuit of (a).

Any physical network corresponding to the abstract model represented by a functional diagram such as Fig. 4a will inevitably include delays in the signal paths and logical devices. A design theory that does not take into account the existence of such stray delays will often lead to unreliable circuits.

We shall also show how the feedback index of a sequential circuit is related to the complexity of the associated function as measured by the variety of possible inter-row transitions in the flow table. We shall begin by describing a procedure for analyzing general sequential networks to obtain the appropriate sequential function (if one exists) in terms of a flow table.

A principal objective of this research is to show how the existence of stray delays should influence synthesis procedures. We shall demonstrate that there exists a class of sequential functions that cannot be physically realized unless delay elements are included in the networks to counter the effect of the stray delays. All other functions (recognizable by means of a reasonably simple criterion) can be synthesized without delay elements. Furthermore, we shall prove constructively that a single delay element is always sufficient for the synthesis of any function.

When we say that the signal is "1" at a certain point in the system, what we mean is that, at the corresponding point in the actual circuit, the value of some continuous physical variable, such as voltage, falls within a range of values which we have specified as representing "1." The existence in our circuits of signals whose magnitudes have continuous ranges means that attenuation may play a part in determining the behavior of the system, particularly when there are feedback loops in the network.

It will be shown in Section III that every closed loop in a sequential circuit must have an incremental gain greater than one. If we define the feedback index of a network as the minimum number of branches that must be cut in order to break all feedback loops [a definition similar to the one used by Mason (8)], then we can say that the number of amplifiers needed is given by the feedback index of the network.

II. ANALYSIS OF SEQUENTIAL SWITCHING CIRCUITS

Given an arbitrary network of logical devices and delay elements, how do we go about finding a description of its terminal behavior? If there are no feedback paths in the circuit, the solution is quite simple, and all that need be done is to record the output state corresponding to each input state. The absence of closed loops guarantees that the signal at every point in the network will ultimately be a function of the input state.

Consider the problem of analyzing a network containing a section, as shown in Fig. 5. We attempt to write an expression for output Z_1 in terms of the input state, and begin with the expression $Z_1 = ad$. Next, we trace our way back through the network in an effort to obtain expressions for a and d in terms of the input variables. In the case of d, however, we find that $d = c + e = c + bd$. No further progress can be made in defining d because, if $c = 0$ and $b = 1$, d can be either 0 or 1 and still satisfy the above equation. Therefore, any expression for Z_1 must involve d (or some other signal at a point within the feedback loop). Hence Z_1 cannot be expressed as a function of the input state alone. This is a characteristic property of sequential switching circuits.

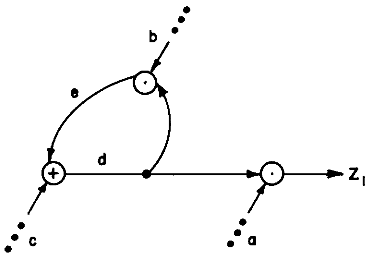
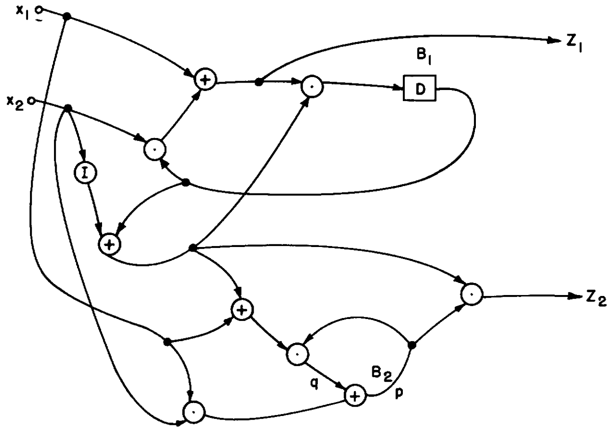


Fig. 5. Circuit fragment with feedback loop.

Given a network containing feedback loops, the first step in the analysis is to arrange it in the form of Fig. 3, wherein there are no feedback loops or delay elements inside the box labeled "combinational circuit." Once this is accomplished, the input of each state branch can be expressed as a function of the input state and the outputs of the state branches, which together constitute the input of the combinational circuit. It is then a routine matter, by using the analysis method of Huffman (5) (in which the terms "secondary device" or "secondary relay" correspond to our term "state branches") to derive a flow table for the network.

The problem of arranging the circuit in the form of Fig. 3 is equivalent to the selection of an appropriate set of state branches. This set must contain all delay elements, and there must be at least one member in every feedback loop of the network. A straightforward procedure for choosing the state branches is, first, to select all delay element branches, and then to choose as many other branches as are necessary to ensure that cutting all the state branches will interrupt all feedback loops. The chosen set may not be unique, even in the number of members, but all sets that meet the stated criteria will lead to correct results. For the sake of simplicity, it is best to choose the number of state branches to be as small as possible.

The network of Fig. 6a will be used to illustrate the method. First, since there is a delay element in the circuit, we choose it as the first state branch, B_1. Next, we can



(a)

	$x_1 x_2$ 00	01	11	10	$y_1 y_2$
1	① 00	① 00	2 10	4 10	00
2	② 01	1 00	② 10	3 11	01
3	2 01	③ 11	③ 11	③ 11	11
4	1 00	④ 10	3 10	④ 10	10

(b)

Fig. 6. (a) Network for analysis. (b) Matrix for network of (a).

select p as the second state branch B_2, thus interrupting the remaining feedback loop. Note that branch q could have been picked instead of p.

Having selected the state branches, we proceed to write equations for the state-branch excitations and for the outputs:

$$Y_1 = (x_1 + x_2y_1)(y_1 + x_2')$$
$$Y_2 = x_1x_2 + (x_1 + x_2' + y_1)y_2$$
$$Z_1 = x_1 + x_2y_1$$
$$Z_2 = y_2(y_1 + x_2')$$

Now, following Huffman (5), we can obtain the flow matrix of Fig. 6b, with columns corresponding to x-states, rows corresponding to y-states, and entries determined by the Y's and Z's.

There exist many networks of logical elements and delays whose terminal behavior cannot be uniquely determined from this kind of functional diagram alone. For example, if critical race conditions (5, 6) exist, circuit operation will sometimes depend upon the relative magnitudes of certain delay elements or stray delays. The presence of critical race conditions will be revealed when we try to derive a flow matrix from the state-branch equations. In Section IV we shall discuss at length the problem of hazards, another circuit "disease" that makes prediction of terminal behavior impossible if the values of the stray delays are unspecified.

Let us now return to the problem of selecting state branches for a given network. It has been pointed out that there is not always a unique set of state branches, and that any choice satisfying the stated criteria would lead to a correct analysis. But now we shall confuse the situation by stating that, in some cases, different choices of state branches may lead to different flow tables. Consider, for example, the circuit and flow matrices of Fig. 7.

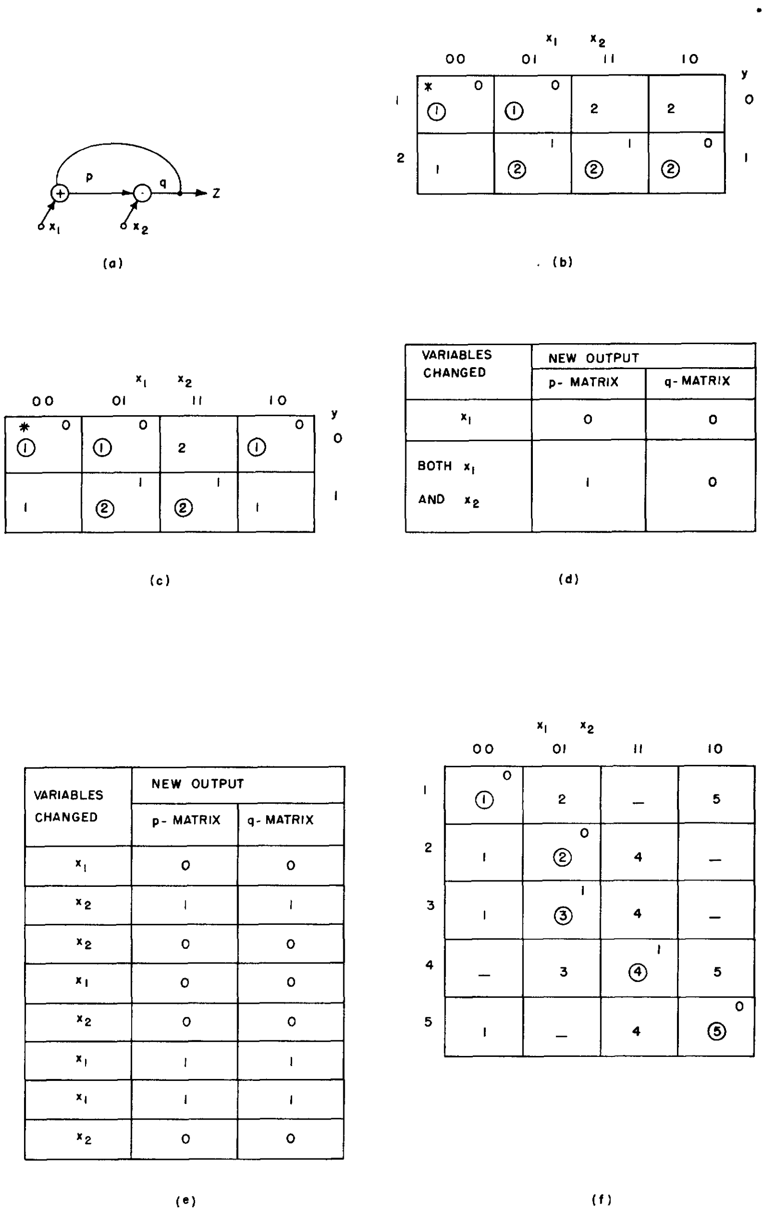
The matrix of Fig. 7b was obtained by choosing p as the state branch, while q was the state branch used in deriving the table of Fig. 7c. (Henceforth, we shall refer to these matrices as the p-matrix and the q-matrix.)

The two matrices certainly look different, but in itself this does not necessarily indicate that they are actually different, since it is always possible to alter the appearance of a flow matrix without changing any of its essential properties. We must examine the situation more closely.

Note that if $x_1 = x_2 = 0$, then in both matrices only one stable total state exists (indicated by starred cells in the diagrams), and the output is the same in both cases. Thus the two states must be equivalent, and reference to the circuit diagram confirms this fact. Suppose that we begin with the starred state in the p-matrix and apply a sequence of input changes, recording after each change the Z-value corresponding to the new total state indicated by the p-matrix. Then we begin at the starred state of the q-matrix and repeat the process, using the same sequence of input changes.

The results of a brief experiment of this kind are tabulated in Fig. 7d, which shows that the two matrices indicate different outputs for the second step of the sequence. Thus the two matrices do differ in a nontrivial sense.

But suppose we restrict ourselves to sequences in which only one input variable is permitted to change at each step. The results of a typical experiment (again be-



(b)

	x_1x_2 00	01	11	10	y
1	* 0 ①	0 ①	2	2	0
2	1	1 ②	1 ②	0 ②	1

(c)

	x_1x_2 00	01	11	10	y
1	* 0 ①	0 ①	2	0 ①	0
2	1	1 ②	1 ②	1	1

(d)

VARIABLES CHANGED	NEW OUTPUT	
	p- MATRIX	q- MATRIX
x_1	0	0
BOTH x_1 AND x_2	1	0

(e)

VARIABLES CHANGED	NEW OUTPUT	
	p- MATRIX	q- MATRIX
x_1	0	0
x_2	1	1
x_2	0	0
x_1	0	0
x_2	0	0
x_1	1	1
x_1	1	1
x_2	0	0

(f)

	x_1x_2 00	01	11	10
1	0 ①	2	–	5
2	1	0 ②	4	–
3	1	1 ③	4	–
4	–	3	1 ④	5
5	1	–	4	0 ⑤

Fig. 7. (a) Network. (b) p-matrix. (c) q-matrix. (d) First test. (e) Second test. (f) Flow table for single changes in input.

ginning at the starred states) are shown in Fig. 7e. Both matrices predict the same output sequence, and, by using Huffman's methods (5), we can show that both are essentially equivalent to the flow table of Fig. 7f for single-input changes; hence they will yield identical outputs for all input sequences of this type.

Thus we conclude that the two matrices are equivalent, provided that only one input at a time is permitted to change, but they are not equivalent if this restriction is removed.

The ambiguity with regard to the choice of flow matrices stems, not from the analysis method, but from the circuit itself. When the matrices indicate different outputs, there will be a corresponding uncertainty about how the circuit itself behaves. Specifically, if $x_1 = 1$ and $x_2 = 0$, then Fig. 7a tells us that the output will be 0, but it gives no indication about what would happen if x_1 and x_2 were changed simultaneously.

The insertion of a delay element in branch p or branch q of the network would resolve the ambiguity concerning the circuit behavior, and would also cause our analysis to lead uniquely to the p-matrix or to the q-matrix, since we would choose the branch with the delay for our state branch.

In general, we can expect these situations to arise whenever the branches with delay elements must be augmented by branches without delays in order to form a state-branch set. The results of Section IV will enable us to decide whether or not, in such cases, the behavior of the circuit will be well-defined for single input changes.

III. FEEDBACK AND GAIN

The primary objective of this section is to show that amplifiers are needed in sequential switching circuits. We shall also have something to say about the number needed and where they should be placed. The discussion is divided into two parts. First, we shall show how feedback loops are associated with sequential switching circuits, and then we shall establish the need for gain around loops.

3.1 THE FEEDBACK REQUIREMENT

When we are given a sequential function to synthesize, the first step, generally, is to express it in terms of a flow table with an irreducible number of rows (5,6). The next step is to assign to each row one or more states of a set of binary-valued variables which we shall call "state variables." (Each state variable corresponds to the response of a state branch in the derived circuit.)

The states assigned to the rows must be selected in such a manner that critical races will not occur for any of the transitions in the flow table. If S_0 is the smallest integer that is at least equal to the base-two logarithm of the number of rows of the reduced flow table, then at least S_0 state variables will be necessary (to allow one state for each row) and $2S_0 - 1$ state variables will always be sufficient (6). The minimum number of state variables that will permit a satisfactory row assignment in any particular case depends upon the detailed structure of the flow table.

We are now in a position to state what is to be proved in this section. Given a sequential function with the properties that

1. every unstable state leads to a stable state, and

2. the output is specified only for stable states (although it may be required that no output signal change more than once as a result of any change in the input state), then any flow matrix free of critical races which corresponds to that function will be reducible unless the excitation of each state variable is a function of the response of that same variable. (A flow matrix will be considered reducible if any of its rows can be eliminated without changing the corresponding function or introducing critical races. Otherwise it is irreducible.)

Thus, under the stated hypothesis, each state branch of any circuit derived from an irreducible flow matrix appears in a feedback loop that contains no other state branches.

Proof of the preceding statement will involve the concept of an excitation (or Y-) matrix (5), which is a specification (obtainable from a flow matrix) of the state-branch excitations as a combinational function of the system input and the state-branch responses.

We can now proceed to the proof. Assume that, in the Y-matrix derived from the given flow matrix, there exists some Y_r that is independent of y_r. We shall now partition the total states of the system into two equal sets with a one-to-one correspondence between the members. This is accomplished by first defining P as the set of all total states for which $Y_r = y_r$. Then we form a second set Q from P by choosing as members of Q all total states of the system that differ from P-states only in the value of y_r.

If s_q is the Q-state derived from some s_p, a P-state, then Y_r must have the same value in both states, since, by hypothesis, Y_r is independent of y_r, and s_p and s_q differ only in y_r. Therefore, since $Y_r = y_r$ in s_p, Y_r must differ from y_r in s_q, and so s_q cannot be a member of P as well as Q. Therefore P and Q are disjoint and equal, and we could have defined Q as the set of all states for which $Y_r \neq y_r$.

Now let us assume that, whenever a race condition occurs involving y_r as one of the unstable variables, y_r will change before any other y can respond. This is equivalent to saying that, in a circuit derived from the flow matrix, the delay in state branch B_i is small relative to the delays in the other state branches. Since, by hypothesis, no critical races exist, this agreement to allow y_r to win all of its races does not alter any essential property of the flow matrix.

Now observe that any system conforming to the above assumptions will be unstable in every Q-state and will change to the corresponding P-state (in which its stability depends upon the stability of the state variables other than y_r). In other words, whenever a Q-state is entered, y_r will change immediately, regardless of the values of the Y's, except for Y_r. Therefore, in the Q-states we can assign arbitrary values to all Y's except Y_r without affecting the behavior of the system. In particular, let us choose values for these Y's so that the values for each Q-state are the same as those in the associated P-state. (The Y_r values were originally the same, as we have already pointed out.) Thus all Y-values in the revised matrix are independent of y_r.

Since all members of Q are unstable, arbitrary output values can be assigned to these states. If we assign to each Q-state the same output entries as those for the associated P-state, then the output will also be independent of y_r.

But now consider any pair of rows differing only in the y_r values. Both the output entries and the Y-matrix entries are identical for both rows, so that the corresponding flow-matrix entries also match, and we can simply merge the members of each of these row pairs, and thus eliminate y_r. The only effect of these mergers is to eliminate meaningless transitions from Q-states to P-states. In all pertinent respects the reduced flow matrix is equivalent to the original matrix. The reduction process can be repeated if necessary until, for all k, Y_k is a function of y_k. This completes the proof, which is essentially the proof given previously by the author (9).

3.2 THE NEED FOR AMPLIFIERS

In a physical realization of a switching circuit, the signals at all points are actually voltages, currents, resistances, or other variables with continuous ranges. (Our discussion will be semantically less awkward if we talk about a specific kind of signal such as voltage. No loss of generality will occur, since the argument applies equally well with any other variable.)

In a properly designed logical system the signal voltages at all points will be constrained to fall into one of two sets of values which are well separated from each other. Signals with values in one set will be given the binary value one, while signals with values in the other set will have the binary value zero. For example, we might arrange our system so that signal voltages exceeding +10 volts represent ones, negative signal voltages represent zeros, and no signal voltages (except during transients) fall between zero and +10 volts. In some cases it is desirable to use different voltage sets at various points in the system.

Now consider the k^{th} state variable of a reduced flow matrix of the type discussed in section 3.1. It follows from the theorem proved in section 3.1 that there must be states of the system variables (x's and y's) other than y_k, such that $Y_k = y_k$. Thus, in any system derived from that matrix there will be a signal path from the output end of the k^{th} state branch B_k through the network to the input end of B_k, for each state in question.

Together with B_k itself, this path forms a closed loop around the output end of B_k. For the condition considered, y_k can have the value zero or one, so that the loop must be bistable in the sense that the signal established around it can have either of two values.

Assume that the binary signal one is represented by voltages exceeding E_1, and that voltages below E_0 are interpreted as binary zeros, where $E_1 > E_0$. Let us now cut the feedback loop containing B_k, apply an input voltage e_a at the input end of the cut, and measure the resulting steady-state voltage e_b which appears at the other end of the break. Our object is to examine the loop gain function L which is expressed as a plot of e_b versus e_a. The experiment will be performed with all x's and y's (except y_k) fixed at values such that $Y_k = y_k$.

We assume that the system is stable when the loop is closed so that the response of B_k in the original network is constant after a transient interval following any input change. Since, without the cut, which was made for test purposes, e_a is identical with e_b (both occur at the same point in the circuit), it follows that stable values of the voltage at the output of B_k in the original circuit can occur only at points on L for which $e_a = e_b$. Furthermore, because of the positive nature of the feedback, a point on L can correspond to a stable state only if the incremental loop gain $(\Delta e_b/\Delta e_a)$ is less than unity in a neighborhood around that point.

We have already seen that the response of B_k may be stable at either zero or one. Thus there must be at least two stable points on the loop-gain characteristic, one of which must occur for a value of $v_a = V_1 > E_1$, and another for $v_a = V_0 < E_0$. Part of a loop characteristic with the necessary properties is shown in Fig. 8a. Since there are

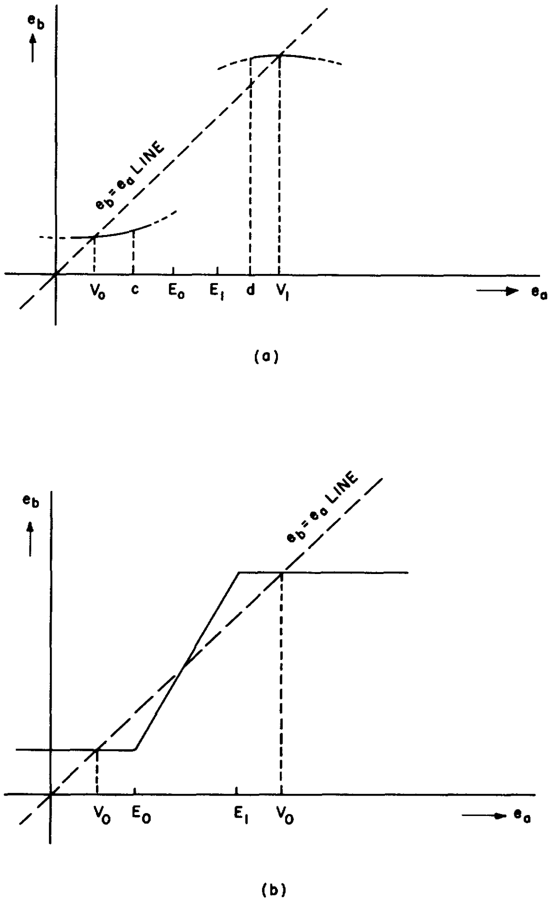


Fig. 8. (a) Part of a satisfactory loop characteristic.
(b) Example of a satisfactory loop characteristic.

neighborhoods around V_0 and V_1 in which the slope of L is less than unity, there must be a point c to the right of V_0 and a point d to the left of V_1 so that $L(c) < c$ and $L(d) > d$. It then follows that $-L(c) > -c$, and hence $L(d) - L(c) > d - c$, and $\frac{L(d) - L(c)}{d - c} > 1$. Thus the average value of the slope of L between c and d exceeds unity, and hence the incremental loop gain must exceed unity in some region between c and d (or become infinite at one or more points). This fact becomes readily apparent if we try to complete the L-curve of Fig. 8a with a single-valued function whose slope never exceeds one in the region between c and d.

A function that would be satisfactory as a loop-gain characteristic, but by no means the only one, is shown in Fig. 8b. It seems reasonable to claim that a device capable of providing incremental gain is some sort of amplifier. With this definition in mind, we have shown that there must be a feedback loop containing an amplifier associated with each state branch of sequential circuits of the type that has been discussed.

The feedback loops considered in the previous argument are those that furnish signal paths from the output of a given state branch back to its input without passing through any other state branches. The existence of such a path for each state variable was established in section 3.1. Now we shall consider other kinds of feedback loops.

First, let us dispose of "pseudo loops," such as the one in Fig. 9, characterized by the fact that, if the loop is broken at any point, the signal at the output end of the break is independent of the signal at the input end of the break. In such a case there is actually no feedback, and the signal at any point in the loop is uniquely determined by the values of the variables that serve as inputs of the loop (P and R in the example). Hence, amplification is unnecessary in a pseudo loop.

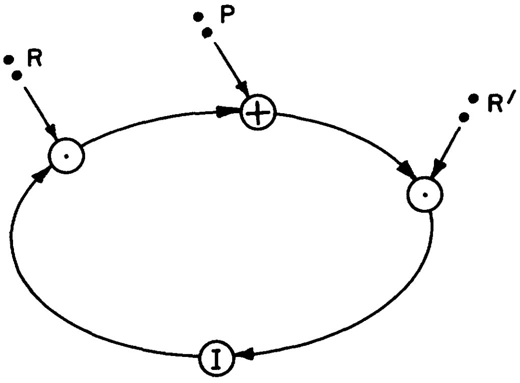


Fig. 9. Example of pseudo loop.

Next, we shall consider loops passing through more than one state branch. For example, this may imply, for three state branches, B_a, B_b, and B_c, that

1. Y_a is partially dependent on y_b,
2. Y_b is partially dependent on y_c, and
3. Y_c is partially dependent on y_a.

If the system variables, other than y_a, y_b, and y_c, can assume such values (simultaneously) that $Y_a = y_b$, $Y_b = y_c$ and $Y_c = y_a$, then a genuine feedback loop exists. The signal flowing around the loop can take on either of the two possible steady-state values, and therefore, as was shown previously, an amplifier is needed in the loop. The same conclusions apply if there is an even number of primed variables in the chain of dependencies, as, for example, when $Y_a = y_b'$, $Y_b = y_c$ and $Y_c = y_a'$. An odd number of inversions would lead to instability.

Since, as is obvious from the synthesis procedure, all feedback loops contain state branches, we have now discussed all types of feedback loops.

A set of branches in a network will be defined as a "feedback-cut set" if the removal of all members of the set interrupts all genuine feedback loops, and if replacing any one member of the set restores at least one loop. We summarize the results obtained by stating that every sequential switching circuit must contain some feedback-cut set with an amplifier in every branch of the set.

The state branches of any network derived from a reduced flow matrix constitute a feedback-cut set, and usually amplifiers are placed at these locations. However, it is interesting to note that there are circuits of this type for which the set of state branches is not the smallest feedback-cut set.

An examination of the function described by the flow table of Fig. 10a indicates that it cannot be described by any table with less than three rows. (We place no restrictions on the permissible input changes.) Thus at least two state variables are required to allow for three internal states. Fig. 10b shows the flow matrix that re-

$x_1 \; x_2$

	00	01	11	10
1	2	2	① 0	① 0
2	② 0	② 0	3	1
3	2	2	③ 1	③ 1

(a)

$x_1 \; x_2$

	00	01	11	10	$y_1 \; y_2$
1	2	2	① 0	① 0	0 0
2	② 0	② 0	3	1	0 1
3	2	2	③ 1	③ 1	1 1
4	2	2	3	3	1 0

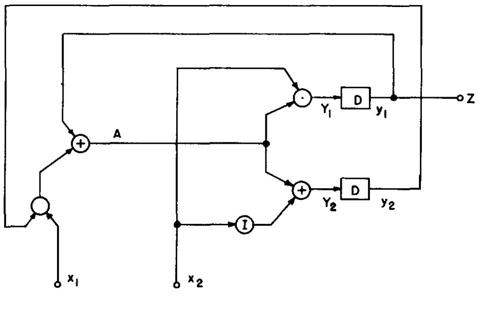
(b)

$$Z = y_1$$

$$Y_1 = x_1(x_2y_2 + y_1)$$

$$Y_2 = x_1' + y_1 + x_2y_2$$

(c)



(d)

Fig. 10. (a) Flow table. (b) Flow matrix. (c) Excitation and output equations. (d) Network that requires one amplifier.

sults from a particular row assignment. (Entries for the fourth row were chosen to simplify the circuit.) The equations in Fig. 10c are the excitation and output equations derived from the matrix, and the network designed in accordance with these equations is shown in Fig. 10d.

Since the branch labeled "A" in the diagram is common to the only two feedback loops, this one branch constitutes a feedback-cut set, and hence a single amplifier located at this point is sufficient for this circuit. In this example the flow table can be reduced to a two-row table describing a new function equivalent to the original function if only one input at a time is permitted to change.

The general problem of finding the minimum number of amplifiers needed for realizing a given flow matrix has not been solved.

IV. DELAYS AND HAZARDS

As is the case with all physical devices, the logical elements used in switching circuits do not operate instantaneously. Transit-time phenomena, stray reactances, hole storage effects - all act to retard rapid signal changes in any system. Mighty efforts are made to minimize these factors, but they can never be entirely eliminated. Furthermore, because of their dependence on second-order effects, the relative magnitudes of the stray delays in any network cannot be predicted with any degree of confidence, and variations may occur even among units built to the same specifications.

The switching-circuit engineer must therefore accept the fact that stray delays of uncertain magnitude may be distributed throughout the system that he designs. Given only an upper bound on the delay magnitudes, he must be able to construct networks whose operation is essentially independent of the relative values of the stray delays. If his design calls for the use of delay elements, then their magnitudes should also be expected to vary from unit to unit. (It may be assumed, however, that the delay element values can be made to exceed the stray delay values by some minimum margin.)

A network that meets these requirements in that its operation is not critically dependent on relative delay magnitudes will be defined as a proper network. But if the terminal behavior of a network is subject to variation because of changes in the relative magnitudes of stray delays or of delay elements, then hazard or critical race conditions are said to exist.

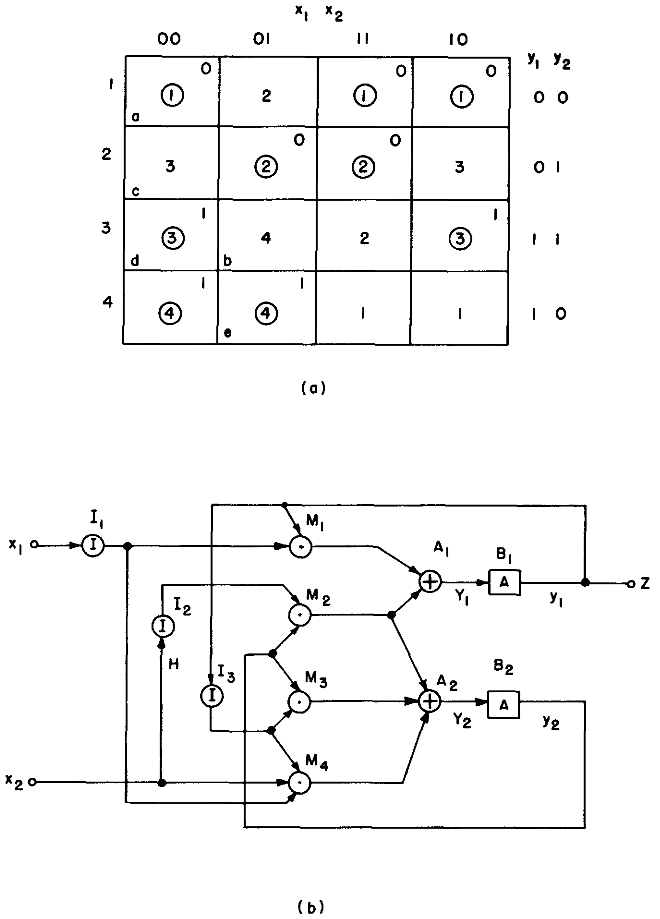
Since critical race conditions can be avoided in a routine manner by making appropriate row assignments (5,6), we shall assume here that this is always done and concern ourselves only with the problem of hazards.

It will be useful to classify hazards in two categories. If the only effect of a hazard is to produce momentary false output signals immediately after an input change, then it will be referred to as a "transient" hazard. If its effects are not confined to the period immediately following the input change that initiates the hazard, it will be called a "steady-state" hazard.

4.1 EXAMPLE OF A HAZARD CONDITION

A typical hazard situation is shown in Fig. 11, which consists of a flow matrix and a circuit derived from that matrix by standard procedures. If we assume that the stray delays in the B_1 and B_2 branches are relatively large compared with the delays in the other branches, then the circuit behavior will be correctly characterized by the flow matrix. In fact, one way of assuring proper operation is to insert delay elements in B_1 and B_2.

Suppose now that a relatively large delay appears in the branch labeled "H" in Fig. 11b. Then, if the system is in the state labeled "a" in Fig. 11a, and if x_2 is turned on



$x_1\,x_2$	00	01	11	10	$y_1\,y_2$
1	① 0 a	2	① 0	① 0	0 0
2	3 c	② 0	② 0	3	0 1
3	③ 1 d	4 b	2	③ 1	1 1
4	④ 1	④ 1 e	1	1	1 0

Fig. 11. (a) Flow matrix. (b) Circuit with hazard derived from (a).

(switched from zero to one), the next stable state will be in row 4 and not in row 2, as indicated in the flow matrix. This comes about as follows.

First, the switching of x_2 changes the output of the multiplier gate labeled M_4 in Fig. 11b to unity, and this switches on Y_2, the output of A_2. Then, y_2 goes on and changes the signal at one input lead of M_2 to unity. The other input to this gate remains at unity, since the delay in H prevents the x_2 change from having an immediate effect at this point. Therefore, a one-signal appears at the output of M_2, penetrates through A_1, and turns on Y_1. This, in turn causes y_1 to go on and a feedback path through M_1 and A_1 holds y_1 on independently of the output of M_2. A second effect of the change of y_1 to unity is that a zero appears at the output of I_3. Thus only zeros can appear at the outputs of M_3 and M_4. Now $y_1 = y_2 = 1$, and the system is in the state marked "b" in Fig. 11a. But only the output of M_2 is holding y_2 on and, as soon as the effect of the original x_2-change penetrates I_2 and the delay in H, a zero appears at the output of M_2, causing the output of A_2 to go to zero and the system to be in row 4.

This behavior could have been predicted from the structure of the flow matrix. When x_2 changes, y_2 becomes unstable and switches. Because of the delay in H, the effects of the y_2 action penetrates to Y_1 before the direct effect of the x_2-change, despite the fact that these actions actually occur in reverse order. Hence, as far as Y_1 is concerned, the system state switches from a to c, where y_1 is unstable. Then y_1 switches to one and moves the internal state of the system to row 3. Once this occurs, Y_1 remains at unity value even after the effect of the x_2-change reaches it. Finally, since y_2 is unstable in state b, it changes again and the system settles down in state e. In a later section we shall present a general method for detecting hazards by examining flow tables.

4.2 PROPER CIRCUITS WITH INERTIAL DELAY ELEMENTS

It will be necessary at this point to distinguish between two types of delay. The effect of a pure (dead-time) delay is simply to convert an input signal f(t) into an output signal f(t-D), D being the value of the delay. An inertial delay (defined only in terms of binary signals) behaves like a pure delay except that input changes which persist for a time less than D, the delay magnitude, are ignored by the output. Thus rapid signal fluctuations are filtered out by such a device, which is somewhat analogous to a low-pass filter. An electromechanical relay is an example of a physical device that behaves very much like an inertial delay.

It has been pointed out in reference 5 that the insertion of inertial delay elements in all state branches of a sequential switching circuit will eliminate all steady-state hazards, and that placing such devices in cascade with all output leads as well will also eliminate transient hazards. This is because the durations of false signals transmitted to state-branch input terminals as a result of the presence of stray delays are too small to affect the state-branch responses, assuming that the inertial delays in the state branches are sufficiently large. In the example discussed in Section 4.1 for instance, an inertial delay in B_1 would have prevented it from responding to the brief unity signal that flashed on at its input after y_2 changed.

4.3 PROPER CIRCUITS WITH PURE DELAY ELEMENTS AND HAZARD-FREE COMBINATIONAL CIRCUITS

Suppose we change one input signal to a multi-input combinational switching circuit. Then ideally, some of the output signals may change to new values, and some may remain the same. But the presence of stray delays may cause an even number of signal changes at one or more of the output terminals, in addition to the normal responses. For example, at an output terminal whose signal is supposed to be zero both before and after the input change, a one may appear momentarily. The signal sequence at an output terminal that ideally should change its signal from one to zero might be one-zero-one-zero. This effect, which we shall refer to as a "combinational" hazard, is discussed by Huffman (10), who presents a constructive proof of the following statement: It is possible to construct a network for any combinational function with the property that in the steady state the output states are as specified by that function, and with the property that a single change in any one input signal never produces more than a single signal change at any output terminal. Such a combinational circuit will be designated "hazard-free."

Making use of this statement, we shall now show that a pure delay in each state branch is sufficient to insure proper operation of any sequential switching circuit if the following conditions are satisfied:

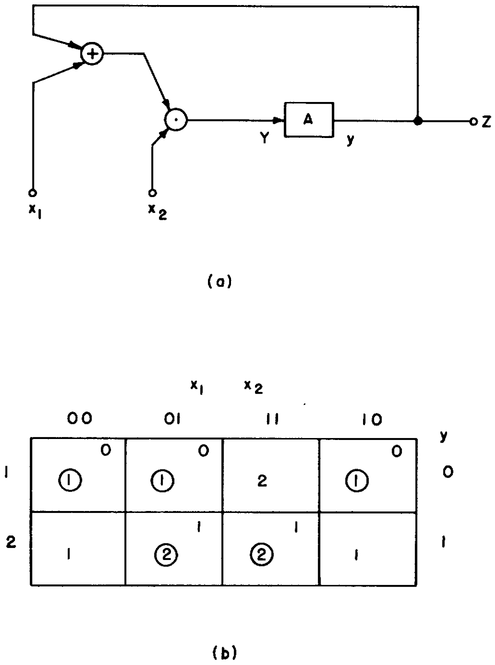
1. There are no race conditions.
2. The combinational circuit block with the circuit arranged in the form of Fig. 3 is hazard-free.
3. Only one input signal at a time is changed.

With the network in the form of Fig. 3, let d_{ij} represent the total amount of stray delay in the path through the combinational-circuit block between the i^{th}-input terminal and the j^{th}-output terminal. (If no such path exists, then d_{ij} is undefined.) If M and m are the uppèr and lower bounds on the range of values of these d's, then we shall assume that there is a pure (or inertial) delay element in each state branch that has a value equal to at least M - m.

Now, with the circuit in a stable state, let one of the system-input signals change. If this does not cause any state-variable excitation to switch, then there will be no further changes in the inputs of the combinational-circuit block, and hence no chance for a hazard to occur, since it follows from assumptions 2 and 3 that the combinational-circuit block will operate properly. If the input change leads to a change in the internal state of the system, then, after a time interval not less than m, one of the state-variable excitations will switch. Subsequently, at least M - m units of time later, the response of this state branch will change and send a new signal back into the combinational circuit. But, by the time this happens, the effect of the original input-signal variation will have passed through all stray delays in that circuit. Hence the combinational circuit will again react correctly to a switching of just one of its input signals. If further internal-state changes occur, the same reasoning shows that the effects of each change will be presented one by one to the combinational circuit as a series of changes in only one input signal at a time. The state-branch delays thus serve to slow down the sequence of signal changes fed to the combinational circuit so that it can absorb the effects of each change before the next one occurs. As long as this is done there is no possibility of malfunctioning, since the values of all signals leaving the combinational circuit are always in accordance with the values envisioned in the design.

4.4 PROPER CIRCUITS WITHOUT DELAY ELEMENTS

The circuit shown in Fig. 12a contains no delay elements; but, if only one input signal at a time is permitted to change, it will properly realize the flow matrix shown in Fig. 12b. In other words, regardless of how we distribute delays among the branches of the circuit, its operation will still be in accordance with the given specifications, provided that we do not space the input changes too closely together.



	$x_1 x_2$ 00	01	11	10	y
1	① 0	① 0	2	① 0	0
2	1	② 1	② 1	1	1

Fig. 12. (a) Circuit without delay elements. (b) Flow matrix for circuit of (a).

There exist examples of more complicated functions that can also be realized with circuits that have this property, and our objective now is to characterize the entire class of sequential functions that can be properly realized without using delay elements. In the next section we shall show that this property is not shared by all functions. (Hereafter we shall assume that only single input changes are allowed.)

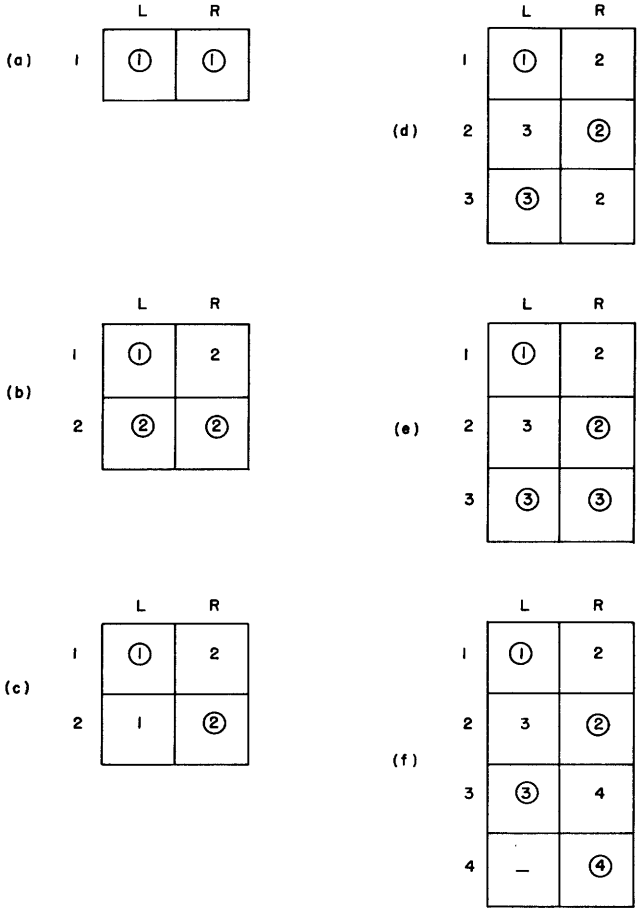
We now want to prove that: Any flow table can be properly realized without delay elements unless it contains an "essential hazard." We define an essential hazard as the existence of a state S_0 and an input variable x, with the property that, starting with the system in S_0, three consecutive changes in x bring the system to a state other than the one arrived at after the first change in x. The flow table of Fig. 2 contains an essential hazard involving state a and input x_2. If, starting from state a, x_2 is changed once, the new state will be c. Two more changes in x_2 lead to row 3 instead of back to c in row 4. (There are two more essential hazards in this table. Can you find them?)

Now let us proceed with the proof that the absence of essential hazards is a sufficient condition for the existence of proper circuits without delay elements.

First we shall enumerate all possible results of a sequence of three changes in one variable x, starting with the system in state Ll. (Total states of the system will be designated as Li or Ri, referring to the row i entries in column L or column R. The L and R columns differ only in the values of x.)

Fig. 13a illustrates the case in which a change in x leads to a stable state in the same row as the initial state. If this is the case, then no other rows need be considered, since changes in x alone cannot lead to a new internal state. Now we consider cases in which the initial change in x leads to a stable state R2 in a new row. Starting with the system in R2, a change in x could lead to a stable state which is also in row 2, as is shown in Fig. 13b. Another possibility, shown in Fig. 13c, is that the second change in x sends the system back into state L1. In both cases only rows 1 and 2 are relevant to the discussion,since there are no transitions possible out of this region during the sequence of changes in x that we are considering.

The third (and final) possible outcome of a change in x from state R2 is that a transition to state L3 will occur. This contingency can be further subdivided into the three cases represented by Fig. 13d, e, and f. With the system in state L3, switching



(a)

	L	R
1	①	①

(b)

	L	R
1	①	2
2	②	②

(c)

	L	R
1	①	2
2	1	②

(d)

	L	R
1	①	2
2	3	②
3	③	2

(e)

	L	R
1	①	2
2	3	②
3	③	③

(f)

	L	R
1	①	2
2	3	②
3	③	4
4	–	④

Fig. 13. Flow-table patterns for three changes of one variable.

x can produce a transition back to state R2, as shown in Fig. 13d. A second possibility is shown in Fig. 13e, in which both entries in row 3 are stable. Finally, as shown in Fig. 13f, a change in x with the system in L3 might lead to a state R4 in a new row. Thus Fig. 13 shows all possible patterns in a flow table when activity is limited to allowing a single input variable to change three times.

Assume that we are using such row assignments that only one state variable changes in the course of a transition between any pair of rows. In particular, let y_{ij} be the variable that differentiates row i from row j. Such an assignment (to be discussed in detail later) can be made for any flow table (6). Furthermore, suppose that all combinational circuits used in the system are hazard-free. (This provision ensures that superfluous excitation changes will not occur because of nonideal behavior of the combinational circuits.)

Now, for each case shown in Fig. 13, examine the possible consequences of a change in x when the system is in state L1.

In case (a), none of the state variables become unstable, so that there is no chance of malfunctioning.

In case (b) only y_{12} changes and, regardless of the order in which the changes in x and y_{12} are sensed by the other state branches, none of them can become unstable, since, in the region concerned, there are no transitions involving state variables other than y_{12}. However, a transient hazard is possible unless the output state in L2 is the same as in either L1 or R2. This is because the output circuits might, because of stray delays, "see" the change in y_{12} before the change in x, so that the output circuits might see the sequence of states as L1-L2-R2. (When we say that a variable A "sees" a change in the variable B, we mean that enough time has elapsed, relative to the stray-delay magnitudes, to enable the effect of the A-change to influence the value of B.)

Again, in case (c), only y_{12} can become unstable in the region under examination, so that no other state branch can be affected. There is no opportunity for a transient hazard to occur, since the output will change only after the output circuits recognize the change in x. (We generally assume that, in a given input column, the output states for an uncircled i-entry are the same as for the circled i-entry in that column.)

Case (d) presents a more difficult problem, although we shall show that the results are the same as those obtained for the first three cases considered, provided that certain additional criteria are observed in making the row assignments.

First, let us see where the potential trouble lies. A change in x occurring with the system in L1 causes y_{12} to change eventually. After this happens, it is possible that the effects of stray delays will cause Y_{23} to see the change in y_{12} before the change in x. Should this occur, y_{23} will then switch (since, according to our flow table, Y_{23} will act as though the system were in state L2). Some other variable Y_r might then see y_{23} as the first variable to change, and this would mean that Y_r would behave as though the system were in some state outside of the region shown in Fig. 13d. If y_r were unstable in that state, then it would change, and this, in turn, could lead to further undesired circuit activity. An example of such a case (Fig. 14) will be discussed later.

It will be convenient to introduce some additional background material and notation before proceeding with a detailed analysis of this case.

We shall briefly discuss a row assignment scheme with the property that only one state variable need change when a transition is made between any pair of rows. This method, based on the Hamming error-correcting code (11), was developed by Huffman (6).

In order to illustrate the discussion, let us consider a row assignment for an eight-row flow table, using state variables $y_1, y_2, \ldots y_7$. The 2^7 possible y-states will be partitioned into eight row sets $S_0, S_1, \ldots, S_7$, each with 2^4 members, and one of these sets will be assigned to each row of the flow table.

A y-state belongs to S_i if and only if the q's determined by the parity relations (involving modulo-two addition)

$$q_0 = y_1 \oplus y_3 \oplus y_5 \oplus y_7$$

$$q_1 = y_2 \oplus y_3 \oplus y_6 \oplus y_7$$

$$q_2 = y_4 \oplus y_5 \oplus y_6 \oplus y_7$$

correspond to the digits of the number i written in binary form. (That is, if $i = q_2 2^2 + q_1 2^1 + q_0 2^0$.)

Members of S_0 correspond to error-free Hamming code words, while the states comprising any other S_i correspond to Hamming code words with errors in the i^{th} position (y_i).

Suppose we wish to change the y-state of the system from some members of S_i to a member of another row set S_j. First we express j in binary form, thus obtaining the q-values associated with members of S_j. Since $i \neq j$, the set of q-values which are obtained in a similar manner for S_i will not be identical to the set corresponding to S_j, and our next step is to specify the q's that are different for the two row sets. Given these q's, we can specify the subscript of the y that must be changed in order to make the desired transition, since only one y will be common to the parity relations specifying the members of a specific set of q's. (A more complete and more general description of the parity-check method will be found in ref. 11.)

Note that this computation involves only S_i and S_j (not s) and that i and j enter into it symmetrically. This means that the same y would be obtained for any other member of S_i and that a transition can also be made from any member of S_j to the set S_i by switching the same y-variables. We shall say that S_i and S_j are separated by this y.

For example, suppose the system is in y-state 0110010, which is a member of S_7, as can be seen from the parity relations, and that we wish to change the state to some (any one will do) member of S_4. The q-values of S_4 are $q_0 = 0$, $q_1 = 0$ and $q_2 = 1$, while for S_7 $q_0 = q_1 = q_2 = 1$. Therefore q_0 and q_1 are different for the two sets. Since y_3

appears in the parity equations for q_0 and q_1, but not in the q_2 relation, changing y_3 will bring about the required transition.

These results can be conveniently expressed with the aid of a binary operation on a pair of non-negative integers. We define a(+)b as the sum of those powers of two appearing in the binary representations of a or b but not of both. For example, 3(+)7 = 4 and 5(+)3 = 6. This operation is clearly associative and commutative, and x(+)x = 0, for any x.

Now if S_i and S_j are any two row sets, then a generalization of our previous results indicates that i(+)j is the subscript of the y-variable separating S_i and S_j. The same ideas are also contained in the statement that if y_k is changed when the system is in row set S_i, then the subscript of the new row set is given by k(+)i.

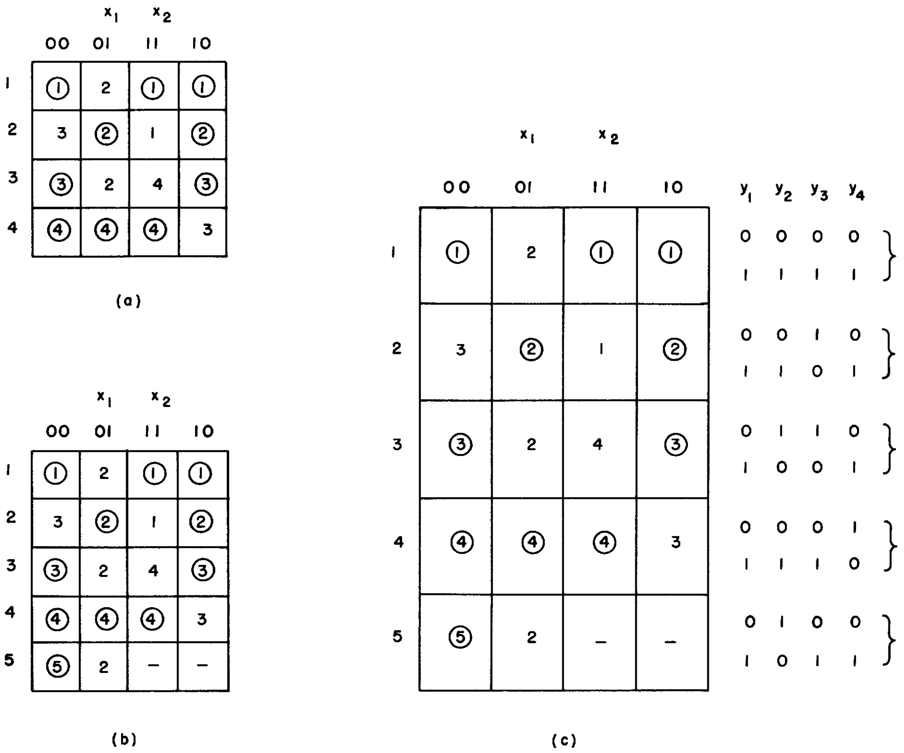
Now we can return to our study of the case shown in Fig. 13d. Assume that a Hamming assignment is used and that row sets S_a, S_b and S_c are assigned to rows 1, 2, and 3, respectively. (Henceforth we shall refer to a group of rows associated as in Fig. 13d as a d-trio.) What we have denoted previously as y_{12} and y_{23} are respectively $y_{a(+)b}$ and $y_{b(+)c}$ in our new terminology.

According to the previous argument, if we change x while the system is in L1, then $y_{a(+)b}$ changes and $y_{b(+)c}$ may misconstrue the order of the previous events and also change. Some other y-variable might see $y_{b(+)c}$ change before $y_{a(+)b}$ and thus see the system in the row set separated from S_a by $y_{b(+)c}$. This set, S_t, which we shall call a "trap" set, has the subscript, t = a(+)[b(+)c] = a(+)b(+)c. Note that, since the trap-set subscript is a symmetric function of a, b, and c, it is invariant with respect to permutations of the three row sets S_a, S_b, and S_c. For instance, assigning S_a to row 3, S_b to row 1, and S_c to row 2 would still leave the trap-set subscript at the value a(+)b(+)c.

Suppose the trap set S_t turned out to be the same as one of those row sets assigned to rows 1, 2, and 3, such as S_a. Then our above equation implies that a = t = a(+)b(+)c. Using the fact that x(+)x = 0, we obtain b(+)c = 0 and b = c. This contradicts our implicit assumption that different row sets were assigned to each row, and hence t cannot equal a, b, or c.

Since, during a case (d) transition, some of the y-variables may see the system in the trap set (with the input corresponding either to L or R), it is essential that the flow-matrix entry for the trap-set states be chosen so that the system ultimately reaches the correct row set. Thus, if we have a flow table with a case (d) transition, we can freely assign row sets to members of the d-trio, but these three row sets then determine the identify of a fourth row set, S_t, which cannot be arbitrarily assigned.

This point is illustrated by the flow table shown in Fig. 14a, in which rows 1, 2, and 3 constitute a d-trio in the first two input columns. If S_0, S_1, and S_2 of a three-variable Hamming assignment were assigned to the first three rows, we could not assign S_3 to the fourth row, because S_3 is the trap set (0(+)1(+)2 = 3) and the entry in the 01 column of row 4 is such that the d-transition might terminate there instead of



(a)

	$x_1 x_2$ 00	01	11	10
1	①	2	①	①
2	3	②	1	②
3	③	2	4	③
4	④	④	④	3

(b)

	$x_1 x_2$ 00	01	11	10
1	①	2	①	①
2	3	②	1	②
3	③	2	4	③
4	④	④	④	3
5	⑤	2	–	–

(c)

	$x_1 x_2$ 00	01	11	10	y_1 y_2 y_3 y_4
1	①	2	①	①	0 0 0 0 1 1 1 1
2	3	②	1	②	0 0 1 0 1 1 0 1
3	③	2	4	③	0 1 1 0 1 0 0 1
4	④	④	④	3	0 0 0 1 1 1 1 0
5	⑤	2	–	–	0 1 0 0 1 0 1 1

Fig. 14. (a) Flow table with d-trio. (b) Trap row added.
(c) Workable row assignment.

in row 2. Thus the trap set must be assigned to a new row of the flow table (which we call a "trap row") with entries that will ensure that the d-transition terminates correctly.

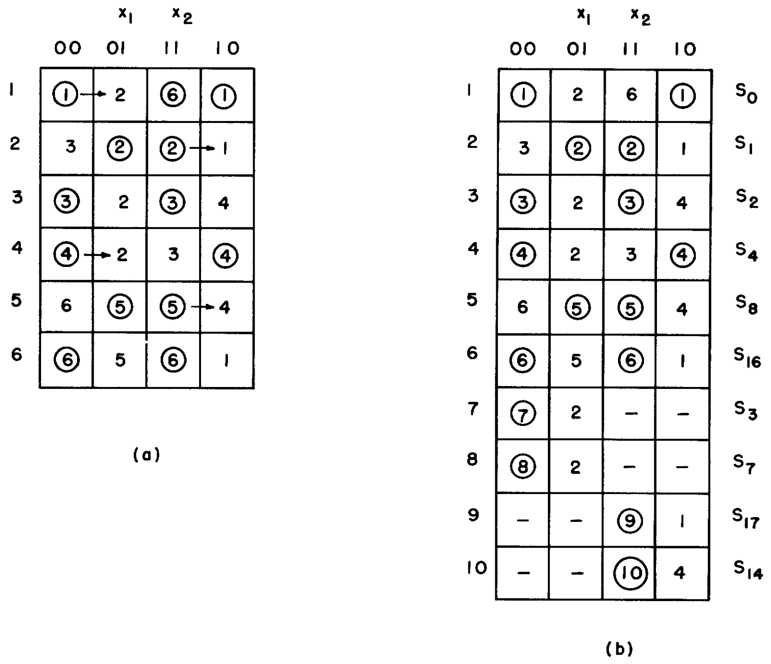
This has been done in Fig. 14b, in which row 5 is the trap row. The row 5 entry in the 00 column could have been chosen as 1, 2 or 3, as well as the 5 shown in the figure, because these four states differ only in the values of y_{12} and y_{23}, and a careful analysis of the situation reveals that in the course of the d-transition y_{12} and y_{23} never see the system in the 00 column of row 5, regardless of the effects of stray delays. (The row 5 entries in columns 11 and 10 can be chosen arbitrarily.) A proper assignment (though not of the Hamming type) is shown in Fig. 14c.

Complications may arise when several d-trios are present in the same table, since trap sets must be provided for all d-trios in such a manner that conflicting demands are never made on the trap-row entries. We shall now show how this can always be accomplished.

Suppose that we assign row set S_0 to the first row of a flow table, S_1 to row 2, S_2 to row 3, S_4 to row 4, and in general $S_{2^{i-2}}$ to the i^{th} row (for $i \geq 2$). Now assume that some set of rows a, b, and c constitutes a d-trio. Then the trap set corresponding to that trio has an index given by $t = 2^{a-2}(+)\ 2^{b-2}(+)2^{c-2}$ (only two terms might be present in the sum if row 1 belongs to the trio, since the index of the row set assigned to row 1 is zero). But from the definition of the (+) operation we can see that $t = 2^{a-2} (+) 2^{b-2} (+) 2^{c-2} = 2^{a-2} + 2^{b-2} + 2^{c-2}$. If another d-trio a', b', and c' also has a trap-set index equal to t, then $2^{a'-2} + 2^{b'-2} + 2^{c'-2} = 2^{a-2} + 2^{b-2} + 2^{c-2}$. By writing the sums on each side of this equation in binary form we can see that this equation implies that there is a one-to-one correspondence between members of (a,b,c) and (a',b',c'). In other words, if the given row assignment scheme is used, then only d-trios composed of the same sets of rows can have the same trap sets. As long as this is true, there cannot be conflicting requirements on the trap-row entries. The trap-row entry in the input column corresponding to L in Fig. 13d can either be stable or can lead back to any member row of the related d-trio, while the trap-row entry in the column corresponding to R must indicate a transition to the member of the d-trio that is stable in that column.

The method is illustrated by Fig. 15, the arrows in Fig. 15a indicating the initial transitions of four d-trios in the given flow table. In Fig. 15b a row assignment has been made and the trap rows have been added. Rows 7, 8, 9, and 10 are the trap rows for the d-trios that have initial transitions in rows 1, 4, 2, and 5, respectively. The index of the row set for row 8, for example, was computed by summing the indices of the row sets assigned to rows 4, 2, and 3.

Note that the highest index of all the row sets has a value of 17. The direct use of the Hamming code scheme requires that the number of variables employed be obtained



		x_1	x_2	
	00	01	11	10
1	① →	→ 2	⑥	①
2	3	②	② →	→ 1
3	③	2	③	4
4	④ →	→ 2	3	④
5	6	⑤	⑤ →	→ 4
6	⑥	5	⑥	1

(a)

		x_1	x_2		
	00	01	11	10	
1	①	2	6	①	s_0
2	3	②	②	1	s_1
3	③	2	③	4	s_2
4	④	2	3	④	s_4
5	6	⑤	⑤	4	s_8
6	⑥	5	⑥	1	s_{16}
7	⑦	2	–	–	s_3
8	⑧	2	–	–	s_7
9	–	–	⑨	1	s_{17}
10	–	–	⑩	4	s_{14}

(b)

Fig. 15. (a) Flow table with four d-trios.
(b) Augmented table showing row assignment and trap rows.

from the formula $2^n - 1$, where n is an integer. Thus, in this example, the smallest value of $2^n - 1$ that is at least equal to 17 for integral values of n is 31. The general method given here will always work, but it does not necessarily yield the most economical assignment.

Let us now review the situation depicted in Fig. 13d. If we use the row assignment method described above, then the relevant portions of the flow table might look like Fig. 16 (specific numbers have been assigned to the rows and row-state subscripts, but this does not imply any loss of generality). Starting with the system in L1, a change in x would cause y_1 to change. If y_3 interprets the order of these two occurrences correctly, then nothing else happens and the system arrives in R2 as specified. If, however, y_3 sees the change in y_1 before the change in x, then y_3 switches. Following this, any y-variable that sees the y_3-change before the y_1-change would then see the system in L4 or R4 (depending on when it sees the change in x). Since y_1 has already seen the change in x occur (otherwise y_1 would not have switched in the first place), it must see the system in the R column, where all Y_1 excitation values are the same and leave y_1 stable. Furthermore, y_3, having recognized the y_1-change, cannot see the system in row 4, and so it too remains stable until it sees the change in x, whereupon it assumes the value that it should have in row 2. All other y-variables will also remain stable, since only y_1 and y_3 are ever unstable in any of the states shown in Figs. 13 and 16. Basically, the system operates correctly because nothing that can happen as a result of the L1-R2 transition can take the system out of the region encompassed by rows 1, 2, 3, and 4. Once the change in x is recognized throughout the circuit, all excitations will be correct, since all entries in the R-column (for the four rows in question) lead to row 2.

	L	R	
1	①	2	S_0
2	3	②	S_1
3	③	2	S_2
4	④	2	S_3

Fig. 16. Flow-table segment with trap row.

Transient hazards may occur unless the output state specified for L1 or R2 is also the output state for L2, L3 and L4.

We have now shown that, regardless of the effects of stray delays, proper operation can always be assured for the transitions shown in Fig. 13a, b, c, and d without resorting to the use of delay elements. This is not true for the transitions shown in Fig. 13e and f.

In case (e) the change in x might be slow in reaching Y_{23}, as in the case (d), and this could cause Y_{23} to react as though the system were in L2. The resulting alteration in y_{23} would then move the internal state to row 3, and if the new y_{23} signal arrived at Y_{23} before the initial change in x, then Y_{23} would not change again, even after the news of the change arrived in x. Hence the system state would become R3 instead of R2. The same sequence of events might occur for case (f) except that the activity would not cease when the system entered R3, but instead a transition to R4 would then occur.

Note that in cases (a), (b), (c), and (d), if we start the system in L1 and change x once, the new state should be the same as the one reached after two additional changes in x. If this experiment is performed for cases (e) and (f), then the specifications for these flow-table segments demand that after the first change in x the state should be R2 and that two more changes in x should lead to different states (R3 and R4, respectively). Thus, a flow table contains what we have defined as an essential hazard, if, and only if, it includes a sector of the type shown in Fig. 13e or f.

The results attained thus far can be stated as follows: If a flow table is free of essential hazards, then it is always possible to construct a circuit corresponding to that table without using delay elements, and in such a manner that no steady-state hazards occur. Furthermore, transient hazards can also be avoided if

1. for each case (b) region of the flow table, the output of the state corresponding to L2 is the same as the output of the states corresponding to either (or both) L1 or (and) R2, and

2. for each case (d) region, either the outputs are the same for the states corresponding to L1, L2, L3, and L4, or the outputs are the same for the states corresponding to L2, L3, L4, and R2 (or both).

Although we have demonstrated that flow tables with essential hazards cannot be properly realized with Hamming row assignments (unless delay elements are used) it remains to be shown that there is no other method for avoiding the use of delay elements in such cases. This will be proved in section 4.5.

We conclude with a few remarks about systems in which several input signals are permitted to change simultaneously. Only in very special cases can such systems be properly realized without resorting to delay elements. The reason for this is that unequal stray delays in cascade with the input terminals may cause the circuit to behave as though the simultaneous changes occurred in some arbitrary order. Improper operation may then result unless the function happens to have the rather restrictive property that, for every stable state, changing all members of any set of input variables will lead to the same new stable state regardless of the order in which the changes occur.

4.5 PROOF THAT DELAY ELEMENTS ARE REQUIRED FOR THE SYNTHESIS OF FUNCTIONS WITH ESSENTIAL HAZARDS

There are two basic steps in this demonstration. First, we shall prove that any proper sequential switching circuit without delay elements can be represented in the form shown in Fig. 17, in which there are separate excitation circuits for each state branch. Henceforth, we shall use the term "separate-excitation form" to describe such an arrangement. (Only one system input is shown for the sake of simplicity, and the output circuits have been omitted, since they will not be relevant to this discussion.)

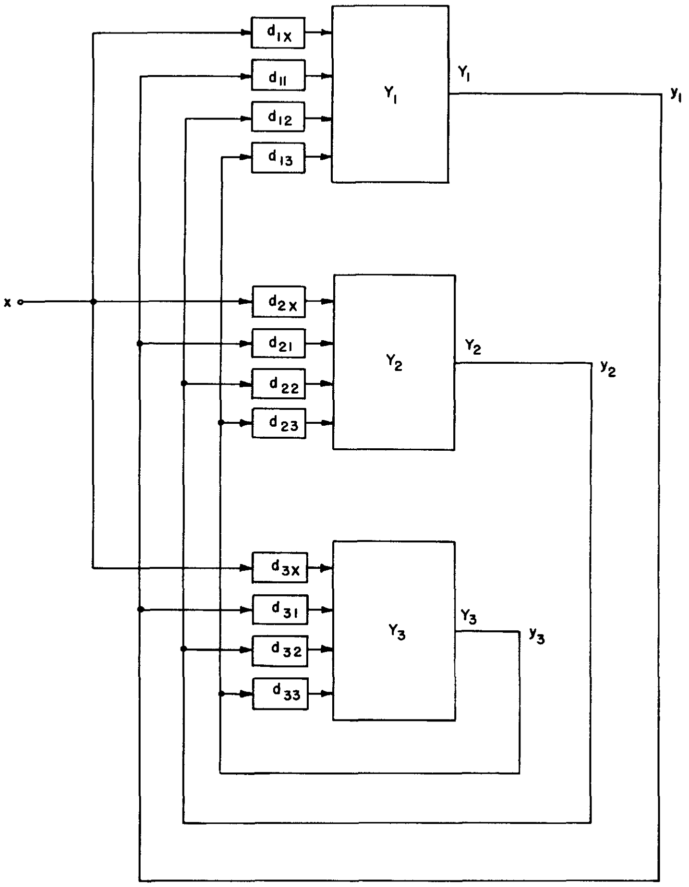


Fig. 17. Separate-excitation form of sequential switching circuit.

The elements labeled "d_{ij}" represent stray inertial delays in the leads between the input of the Y_i circuit and the output of the j^{th} state branch. In other words, a change in y_j will be recognized at the Y_i terminal only after a time lapse of d_{ij}. Similarly, d_{ix} is the delay between x and Y_i.

The second step in the proof will make use of the result obtained in the first step to show that, for any circuit corresponding to a flow table with an essential hazard, it is possible to choose a set of values for the stray delays such that the circuit will not always act in accordance with the flow table, if no delay elements are used.

Now let us see how an arbitrarily chosen proper sequential network, such as the one in Fig. 18a, can be redrawn in the form of a separate-excitation network. (Incidentally, the circuit shown is simply a random arrangement of elements, doing nothing in particular.) The method to be used will entail an increase in the number of state branches in the circuit.

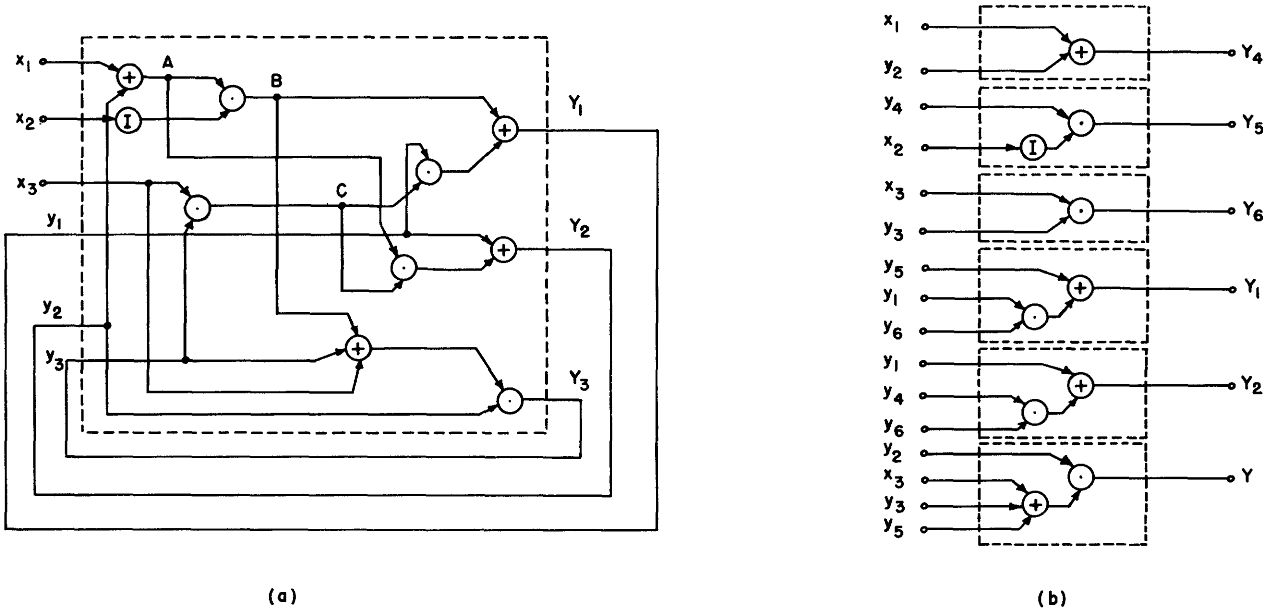


Fig. 18. (a) Sequential network. (b) Separate-excitation form of circuit for (a).

First select some point in the network (other than a point connected directly to an input terminal or to a state branch) such that signal variations at that point can affect more than one state-branch excitation (assuming that all feedback paths are interrupted by cutting all of the state branches). Then label the branch terminating at this node as a new state branch. Repeat this process until no such points remain. Obviously this is a finite process, since we start with a finite number of nodes and eliminate one at each step.

In Fig. 18a the signal at point A affects Y_1, Y_2, and Y_3, so that, according to the above procedure, a state branch with a response that we shall call y_4 is assumed to terminate at A. The excitation circuit for y_4 is shown at the top of Fig. 18b. Similarly, the signals at B and C each affect several state-branch excitations, and therefore additional state branches, B5 and B6, are associated with them in Fig. 18b.

The diagram shown in Fig. 18b is identical with the original network for Fig. 18a, the connections between terminals Y_i and y_i have been omitted to avoid cluttering up the diagram. A flow matrix derived on the basis of the transformed diagram will have, in general, more rows than the original flow matrix because of the additional state variables, but the two matrices will be equivalent in the sense that the same flow table can be derived from each matrix. This follows from the fact that the same circuit generated both matrices, and since the circuit is, by definition, proper, any description of its terminal behavior must be independent of the manner in which it is derived. (We assume throughout this discussion that only one input at a time is permitted to change.)

Suppose that we are given the separate-excitation Y-matrix of a circuit that realizes a flow table with a region containing an essential hazard, as shown in Fig. 19. (In this diagram v may equal 3, but not 1 or 2.)

	L	R
1	①	2
2	3	②
3	③	v
v	-	ⓥ

Fig. 19. Flow-table region with essential hazard.

We shall see that ideal behavior (no hazards and zero delay) can be assumed for each of the Y-circuits, and that only the d's in Fig. 17 need be adjusted by some malicious demon in order to cause malfunctioning on the part of the over-all system. In order to be able to specify a satisfactory (for the demon) set of d's, we need only consider the Y-matrix of the separate-excitation form of the circuit, ignoring the detailed contents of the combinational Y-boxes of Fig. 17.

Our specific object now will be to choose the d's in such a manner that, for at least one y-state corresponding to row 1 (Fig. 19), an input change from column L to column R, with the system initially in that y-state, will lead to state-Rv, instead of to R2.

If the given Y-matrix describes a circuit behaving in accordance with the given flow table, then given s_1, some y-state assigned to row-1, there must exist three other y-states, s_2, s_3, and s_v, associated with rows 2, 3, and v, respectively, and having the following properties:

1. With the input state corresponding to the L-column, and the y-state at s_1, a change in x might cause the internal state to ultimately reach s_2 (possibly after passing through several intermediate states).

2. Similarly, with the system initially in Rs_2, a change in x might lead to y-state s_3.

3. Another change in x, with the system initially in Ls_3, might lead to y-state s_v. (Note that there may, in general, be several such sets of states. We choose one set.)

A consequence of property (1) is that there exists a sequence of y-states, beginning with s_1 and ending with s_2, that is such that, if s_k precedes s_{k+1} in that sequence, then the Y-values in the Y-matrix for column R and y-state s_k are such that the next y-state might be s_{k+1}. We do not exclude the possibility of critical races being involved in this process, and if several such chains of states exist, we choose one, calling it C_{12}.

For example, suppose that the states s_1 and s_2 are 0000 and 1111, respectively. Then, in column R, we might find that in s_1 the Y-matrix entries are 0011 (a race condition, since two state variables are unstable). One possible outcome of the race could make the next y-state 0001, where, let us say, the Y-excitations are 1001. This, in turn, would lead the system to y-state 1001. Finally, the excitations at this state might be 1111, so that a possible result of the ensuing race could be a transition to y-state 1111 (which is s_2). The sequence of states we are looking for is therefore 0000-0001-1001-1111. Conceivably, for the same Y-matrix, there might have been other sequences between s_1 and s_2, such as 0000-0011-0010-1010-1011-1111.

Similar chains must also exist linking s_2 to s_3 (in column L) and linking s_3 to s_v (in column R). Let C_{23} and C_{3v} be a pair of such sequences. A combined sequence, obtained by stringing together C_{12}, C_{23}, and C_{3v}, will be called a C-sequence.

Our next step is to express in more compact form the essential information contained in these sequences. Instead of writing down the actual states, let us now indicate, in order, only the y-variables that actually change in the steps from state to state. If two or more y-variables change simultaneously, then we write them in brackets. The two sequences specified previously would then be written as $y_4y_1(y_2y_3)$ and $(y_3y_4)y_4y_1y_4y_2$,respectively. Let us now relabel the y-variables in the following manner: Call the first variable appearing in the C-sequence y_1, and assign the other integers in order so that $i < j$ implies that the first appearance of y_i in C precedes the first appearance of y_j. Thus, if we have a C-sequence originally written as $xy_4y_2y_3xy_2y_1xy_4y_3(y_5y_2)y_1$ (the x's are used to denote input changes, which separate the three component sequences), we note that the order in which the y's make their initial appearances in the sequence is; y_4, y_2; y_3, y_1, y_5, and so the following transformations are made: $y_4 \rightarrow y_1$, $y_2 \rightarrow y_2$, $y_3 \rightarrow y_3$, $y_1 \rightarrow y_4$, and $y_5 \rightarrow y_5$, so that the sequence is now written as $xy_1y_2y_3xy_2y_4xy_1y_3(y_5y_2)y_4$. This relabeling is purely a means of simplifying our notation, and introduces no new concepts.

Let us now designate with stars all the y-variables that appear an odd number of times in the C-sequences, and also underline the first appearance of each y-variable in C. The preceding sequence is then written as $x\underline{y}_1\overset{*}{\underline{y}}_2\underline{y}_3x\overset{*}{y}_2\underline{y}_4xy_1y_3(\overset{*}{\underline{y}}_5\overset{*}{y}_2)y_4$. Note that if the system is in s_1, and if each starred variable switches, the y-state will be s_v. We shall now show that it is possible to assign the d-values (Fig. 17) in such a manner that, if the system is in state Ls_1, and a change in x occurs, each starred y-variable in the C-sequence will change once and each unstarred y-variable in the C-sequence will change twice, leaving the system in state Rs_v, corresponding to row-v instead of row-2.

The action will take place in two "rounds," and we shall quantize our delay values so that the events occur only at $t = 0, 1, 2, \ldots,$ etc. Round one begins when x changes at $t = 0$, with the system initially in state Ls_1. Then, one by one, the y's in the sequence change, in increasing numerical order of their subscripts. In other words, y_i changes at time i. None of the changes occurring before time i are seen by y_i until $t = i$. Thereupon, y_i simultaneously sees changes in all of the y_j's that have the property that N_{ji} is odd. We define N_{ji} as the number of y_j appearances preceding the underlined y_i in C, but not bracketed with it. This means that at $t = i$, y_i suddenly sees the internal state of the system change from s_1 to the state in our state sequence at which y_i first becomes unstable. Putting it another way, y_i behaves as though at $t = i$ all of the steps in the C-sequence leading up to the underlined y_i (except for changes appearing in a bracket with the underlined y_i) have occurred. For example, in our sample sequence, at $t = 4$, y_4 sees the changes in y_1 and y_3, but sees no other changes during round 1. At $t = 5$, y_5 sees changes in x and in y_4.

All y's except those initially appearing in the middle section (the C_{23} part) of the C-sequence see the change in x occur during this round, along with the other changes. After a y-variable has switched, it sees no further changes (including its own) in round 1, which terminates at $t = n$, where n is the number of variables appearing in C.

Round 2 commences at $t = n+i$, and the variables again react in increasing numerical order of subscripts. At $t = n+i$ the variable y_i sees simultaneously those changes that have occurred after time i, which will cause y_i to act as though the system were in state Rs_v. (This includes seeing the change in x if the underlined y_i is in C_{23}.) The result is that y_i changes during round 2 if, and only if, it is an unstarred variable in C.

This effect will be achieved if, at time n+i, y_i sees those changes that occurred in the values of starred variables during round 1 that it did not see at time i, and the second changes in those unstarred variables that y_i saw changing during round 1.

Thus round 2 serves to restore unstarred variables to their pre-round 1 states (which correspond to their values in s_1 and s_v). Note that if $j > i$, and if y_j is unstarred, then y_j will change twice, but, because of the inertial effect of d_{ij}, y_i will not see either change.

For example, in the sequence given above, during round 2, y_4 will see, at $t = 9$, changes in x, y_2, y_5, y_1, and y_3. (The last two changes occurred during round 2.) Thus y_4 will now have seen a total of two changes in y_1, two changes in y_3, one change in y_2, one change in y_5, and one change in x, so that y_4 sees only y_2, y_5, and x in different states than at the beginning of the process. The system then seems to y_4 to be in Rs_v; therefore y_4 switches. As far as y_5 is concerned, at time 10 it sees changes in y_2, y_5, and y_4.

Now let us see what values the d_{ij}'s must have in order to bring about the events described above. (Remember that d_{ij} is the time it takes y_i to see a change in y_j. The unit of time may be arbitrarily chosen.) First of all, if the underlined y_i appears in C_{12} or C_{3v}, then $d_{ix} = i$, so that y_i sees the change in x during round 1. If the initial y_i appears in C_{23} then $d_{ix} = n+1$, and y_i will see the change in x during round 2. Five disjoint cases must now be considered in order to specify the remaining d's.

1. If $i > j$ and N_{ji} (as defined) is odd, then $d_{ij} = i-j$. This ensures that at $t = i$ (during round 1) y_i will see changes in all y_j's that have undergone net value changes in C before the appearance of the initial y_i.

2. If $i > j$, N_{ji} is even, and y_j is unstarred, then $d_{ij} = L$ (a long time, say, greater than $2n+1$). Changes in unstarred y_j's should never be seen by y_i if they are not necessary to bring about the y_i change in round 1.

3. If $i > j$, N_{ji} is even, and y_j is starred, then $d_{ij} = n+i-j$. Changes in starred y_j's that do not meet condition 1, should be seen by y_i at $t = n+1$ (in round 2).

4. If $i \leq j$ and y_j is unstarred, then $d_{ij} = L$. Such y_j's fall into the same category as those described in case 2.

5. If $i \leq j$ and y_j is starred, then $d_{ij} = n+1-j$. The y_j's of this type are to be treated like those of case 3.

We have now specified all of the d-values.

If the d-values in the circuit differ slightly from the values specified above, will the behavior of the system still be in accord with our description of rounds 1 and 2? The answer is yes, provided that the deviations are sufficiently small. Suppose that, because of slightly inaccurate d-values, the excitation Y_i changes three times within the interval from $t = i$ to $t = i+e$ (instead of once at $t = i$, as we have specified), where e is small compared with unity. Although this may cause y_i also to change several times, since we have not specified delays in the state branches, the inertial nature of the delays in Fig. 17 that appear between y_i and each excitation circuit will absorb the two extra changes, and so subsequent operation will be unaffected, except for slight deviations in timing which may, in turn, cause brief, false, signal changes at a later time. In general, if the errors in the d-values are so small that even the cumulative effects of a series of errors will not add up to produce incorrect excitations that last as long as one time unit, then the system will still operate so that it terminates in state Rv.

This concludes our proof that, if an essential hazard exists in a flow table, any circuit realization of that flow table is liable to operate improperly because of the effects of stray delays, if delay elements are not used.

Note that if we can determine in advance (possibly on the basis of estimates arrived at by considering the number of elements in the various paths) that the stray delays will not be unfavorably distributed, then delay elements need not be employed.

Another conceivable way in which delay elements might be dispensed with might be to assign two sets of y-states, designated A and B, to each row of the flow table. The system would have two modes of operation, one using only A-sets, and the other using only B-sets, and there would be no way for a shift to occur from one mode to the other during normal operation. The y-states would be so chosen that, for any set of stray-delay values, the system would operate correctly in at least one of the modes. (Probably more than two modes would generally be required.) We would then arrange matters so that when the system is first switched on, the initial internal state corresponds to the proper mode for the existing distribution of d's in the circuit. In other words, we thwart the demon by letting him select the stray-delay values on the assumption that the y-state corresponding to L1 (Fig. 18) is sometimes s_1 (see first step of our proof), and then operating the system in a mode that will never allow it to enter y-state s_1. Such a procedure will not work if the stray delays are liable to change after the system has been in operation for a while. In effect, this method resembles the one mentioned in the previous paragraph, since a knowledge of the stray-delay values is assumed, although in this case we might somehow make the system automatically ascertain and act upon the relevant data.

4.6 CIRCUIT REALIZATIONS WITH A MINIMUM NUMBER OF DELAY ELEMENTS

It is always possible to obtain a proper realization of a function with essential hazards simply by inserting delay elements in each state branch, as was pointed out in sections 4.2 and 4.3. Such a procedure would naturally be followed if amplifiers with relatively long inherent delays had to be used in the state branches, and in such cases there would probably be no need for specific components whose sole use was for obtaining delay.

However, for systems employing certain kinds of physical devices, it might be necessary to use special elements to achieve delays, and it would then be desirable to know how the required number of delay elements could be minimized. We shall discuss various methods of using delay elements, including several general techniques by means of which a single delay element is sufficient for the proper synthesis of any sequential switching function.

First, let us see how trouble can be avoided when a single essential hazard exists, as in the flow-matrix section shown in Fig. 20. (The arrow indicates the hazardous transition.) If a delay element is associated with y_2, then no trouble can occur, since, by the time y_2 responds to the change in x, y_1 will already have seen x switch, and hence will remain stable.

Proper operation can also be ensured if an inertial delay is placed in the state branch corresponding to y_1. If this is done, then even though y_1 may see the change in y_2 before the change in x, and hence become unstable, it will not respond incorrectly, since, before it can change, it will have seen the change in x, and hence become stable again.

	x			
	0	1	y_1	y_2
1	①	→2	0	0
2	3	②	0	1
3	③	③	1	1

Fig. 20. Section of flow table with essential hazard.

If a row assignment requiring only a single variable change for any row-to-row transition is used, and if one of the two methods discussed above is employed whenever an essential hazard occurs, then proper operation will result. In many cases in which several essential hazards are present, it is possible to make the row assignments in such a manner that the same y-variable is involved in more than one hazardous transition, so that a single delay element serves a multiple purpose.

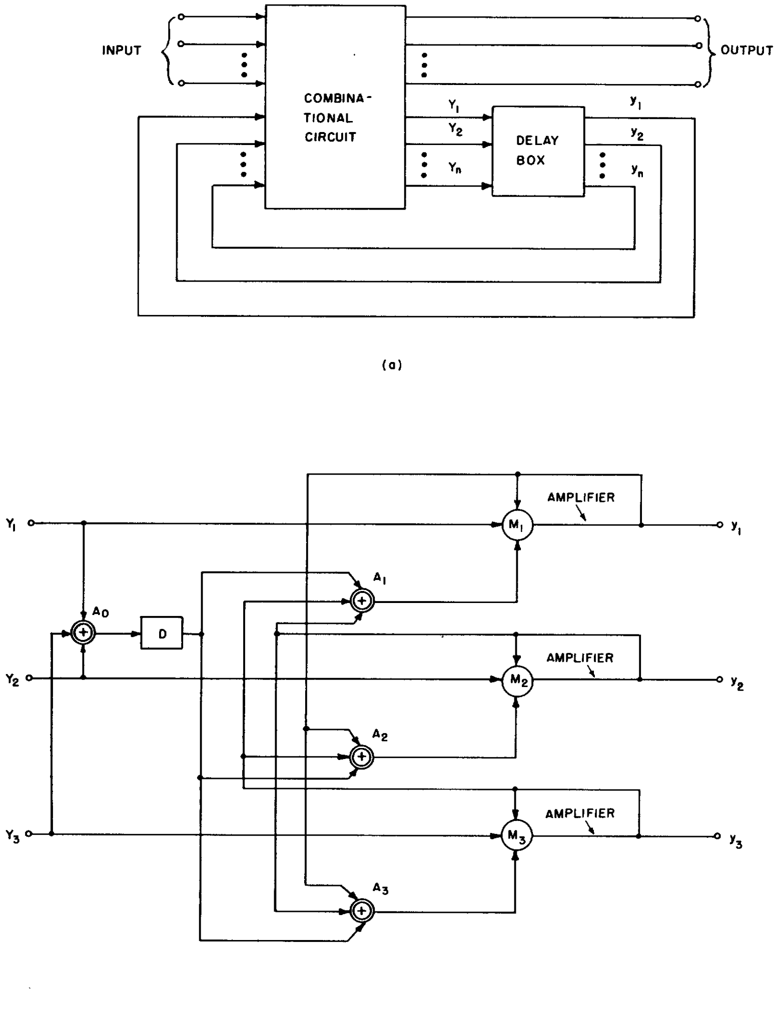
An interesting example of a situation in which such economy can be achieved is illustrated by Fig. 21, a flow matrix for a 4-state binary counter. Every transition involves an essential hazard, and in each case y_3 corresponds either to y_1 or y_2 in the example illustrated by Fig. 20. Thus, if an inertial delay element is associated with y_3, then proper operation will always result.

	x				
	0	1	y_1	y_2	y_3
1	①	2	0	0	0
2	3	②	0	0	1
3	③	4	0	1	1
4	5	④	0	1	0
5	⑤	6	1	1	0
6	7	⑥	1	1	1
7	⑦	8	1	0	1
8	1	⑧	1	0	0

Fig. 21. Flow matrix for a four-state binary counter.

Consider, for instance, the effect of a change in x when the system is initially in state 0-3 (column 0 and row 3). After the change in x, y_3 becomes unstable, but, because of the delay, it does not change until the other y-variables (particularly y_1) have seen the change in x. Thus no trouble will occur during this action. If the system is initially in state 1-4, then a change in x will be followed by a change in y_1, and it is possible for y_3 to see the last change occur first. This means that y_3 temporarily sees the system in state 1-5, and therefore becomes unstable. But, because of the inertial delay, y_3 will not respond to the false signal, and once again malfunctioning will be averted. This method can be applied to counters with any number of states; one inertial delay element always being sufficient.

We shall now present a method for properly synthesizing any sequential switching function, by using only one delay element (which may be pure or inertial). Fig. 22a shows such a circuit, which is characterized by the following properties:



(a)

(b)

Fig. 22. (a) Single-delay circuit with delay box. (b) Delay box.

1. The box labeled "combinational circuits" is hazard-free and generates Y- and Z-functions that correspond to a row assignment in which only one state variable changes for any particular row-to-row transition.

2. If only one Y at a time is allowed to change, and if these Y-changes are not spaced too closely in time, then the delay box behaves as though a delay element D_i connects each Y_i with the corresponding y_i.

It has already been demonstrated in section 4.3 that, under these conditions, such a circuit will operate properly. It now remains to be shown how the delay box can be constructed by using a single delay element, regardless of the number of input-output pairs.

Such a circuit for three variables (it can be generalized to any number) is shown in Fig. 22b. The elements denoted by circled plus signs are modulo-two adders (the output is the modulo-two sum of the inputs), the M's are majority elements (the output is unity if, and only if, at least two of the three inputs are ones) and the D is a pure or inertial delay element. The amplifiers shown in Fig. 22b are necessary because of the feedback around the M's (see Section III). They also amplify the signals in the state branches, so that no other amplifiers are necessary in the system.

In order to see how this circuit works, let us focus our attention on the y_2-output terminal. We can trace a path from Y_2 to y_2, which passes through adder A_0, the delay element, adder A_2, and the majority element M_2. In A_0, signals from all the other Y-terminals are added to the Y_2-signal, and in A_2 the resulting signal (delayed) is added to signals from all the y-terminals except y_2. If we assume that $Y_i = y_i$ for all i's except possibly for i = 2, then, in the steady state, the output of A_2 should correspond to the input from Y_2, since all the other signals that were added to Y_2 cancel out in pairs ($Y_i \oplus y_i = 0$ if $Y_i = y_i$). Thus, if we examine the inputs to M_2, we see that two of them are equal to Y_2 (one comes directly from the Y_2-terminal, and the other, which we have just discussed, comes from A_2), so that y_2, the output of M_2, must equal Y_2, and since y_2 is also the third input to M_2, all three inputs are the same. The same argument can be applied to each of the other Y_i-y_i pairs; thus it follows that the circuit is stable, with all y_i's equal to the corresponding Y_i's.

Suppose now that Y_2 is switched. One effect is that one input of M_2 changes, but this is not enough to switch y_2. The other effect is that the output of A_0 changes. Nothing else can happen until the output of D changes (after a delay), whereupon the output of each A_i switches. The effect of this is to change one input of each M_i. Since one input to M_2 (from Y_2) has already changed, y_2 switches. For all other values of i, the Y_i and y_i inputs of M_i remain fixed so that no y_i, except for y_2, can change. After y_2 changes, a second input (from y_2) to each A_i, except for A_2, switches, so that the outputs of these A_i's are restored to their original values. The change in y_2 is also transmitted to the third terminal of M_2, so that once again, for every i, all three inputs of M_i are in agreement. At this point the circuit is in an equilibrium state equivalent to the one that existed before Y_2 switched, and another input change can be applied. Thus the circuit of Fig. 22b can serve as the delay box in Fig. 22a, and we have shown that a single delay element is sufficient for the proper synthesis of any function.

An implicit assumption (stemming from section 4.3) concerning the use of the preceding circuit form is that only one system input at a time is permitted to change. However, a closer examination of our delay box indicates that this restriction is superfluous (if transient hazards are tolerable).

Suppose that during some time interval (small compared with the delay magnitude) all of the Y-inputs to the delay box are subject to random variations, provided only that at the end of the interval no more than one Y has undergone a net change. Assume, first, that none of the Y-values undergoes a net change. Then, if D is inertial, its output remains constant, and so none of the y's can switch. If D is a pure delay then its output will, in general, change several times (an even number). But by the time this occurs, all of the Y's have resumed their original values, so that, again, none of the y's can switch.

Next suppose that Y_k undergoes a net change (all other Y's changing an even number of times). If D is inertial, then its output will change once after a delay; and then y_k will change once, with the result that there is no effect from transient changes. The D-output may switch several times (an odd number) if D is pure, but by the time these changes begin, only Y_k will show a change, so that only y_k can switch. The value of y_k will fluctuate in synchronism with the output of D, and then settle down to agree with the new Y_k value.

If, in the circuit of Fig. 22a, several system-input variables are allowed to switch simultaneously, then, after an input change, at several Y-terminals there may be an even number of superfluous Y-changes, even though the combinational circuit is hazard-free. According to our analysis of the delay box, this can at worst cause an even number of superfluous changes in the value of the y_k that is supposed to switch, but no malfunctioning can result, since the state-variable excitations in the system will be the same for both values of y_k. False output signals might occur briefly when the inputs first change, but otherwise the system will still work correctly.

We might think that similar reasoning would indicate that it is not necessary to use hazard-free combinational circuits, since the delay box acts somewhat like a set of inertial delays. This is not true, however, as the following argument shows. Suppose that the result of an input change is to alter Y_k. Then, after a delay, the output of D changes, and y_k switches. The effect of the y_k-change is fed back into the combinational circuit and the result might be an even number of Y_j-changes ($j \neq k$) because of combinational hazards. These Y_j-changes could penetrate to alter y_j if the y_k-change has not yet reached A_j. Further action might then occur and, in general, the outcome will be uncertain.

In conclusion we can say that, for any sequential function, a proper physical representation is possible by using the circuits of Fig. 22, provided that the row assignment used calls for no more than one state-variable change following any input change, and that only one input at a time is switched. If the last restriction is omitted, then transient hazards may occur.

The number of state variables that must be used in connection with this method grows linearly with the number of rows of the flow table to be realized. To be more precise, the number of state variables needed for the most complex n-row flow table is $2^{S_0} - 1$, where S_0 is the smallest integer with the property that $2^{S_0} \geq n$. In general, however, if we do not try to minimize the number of delay elements, then the number of required state variables increases logarithmically with the number of rows. With S_0 as defined above, the number of state variables sufficient for the synthesis (6) of any n-row table is $2S_0 - 1$.

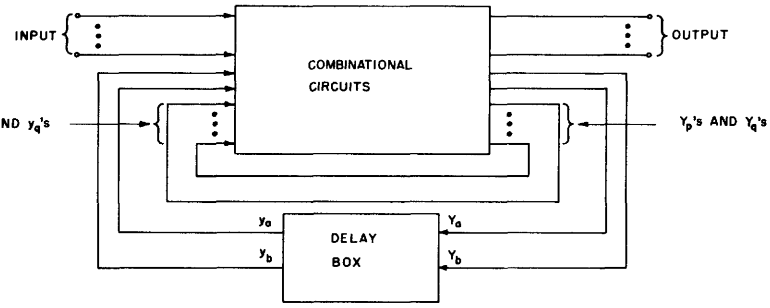
We shall now demonstrate a general method for a one-delay-element synthesis that calls for the use of $2S_0$ state variables. This method is based on Huffman's $2S_0$ assignment (6). An example of such an assignment for an 8-row flow table is shown in Fig. 23a. The state variables are y_a, y_{q1}, y_{q2}, y_b, y_{p1}, and y_{p2}, and the numbers in the matrix indicate the rows to which the y-states are assigned. For example, the 2-entry labeled "g" in the diagram tells us that, for one of the y-states in the row set corresponding to row 2, $y_a = y_b = 0$, $y_{q1} = 0$, $y_{q2} = 1$, $y_{p1} = 0$, and $y_{p2} = 1$. Note that y_a and y_b designate the quadrant in which a given state lies, and that within a quadrant the y_p-variables indicate the column and the y_q-variables indicate the row. To expand the number of row sets, we increase (by equal numbers) the number of y_p- and y_q-variables, retaining the general form of Fig. 23a.

Consider now a circuit in the form of Fig. 23b, in which an arbitrary sequential function is realized by using a $2S_0$ row assignment and a 2-variable delay box (of the type shown in Fig. 22b) in the y_a and y_b state branches. No delay elements appear anywhere else in the circuit, and all combinational networks are hazard-free.

Suppose that the system is stable in the y-state corresponding to the member of row set 2 labeled "g" in Fig. 23a, and that following a change in x the new state is to be in row 4. For the original input state and all y-states in row set 2, all y-variables should be stable. For the new input state, a transition from g to h (in row set 4) will occur if, in all those members of row set 2 that lie in the upper left-hand quadrant of Fig. 23a, the Y-excitations correspond to the coordinates of h ($Y_a = 0$, $Y_b = 1$, $Y_{q1} = 0$, $Y_{q2} = 1$, $Y_{p1} = 1$, $Y_{p2} = 0$). This means that the y_q-variables remain stable, some of the y_p-variables (in this case both of them) become unstable and switch, and y_b becomes unstable and eventually changes (after a delay). All of the y_p-changes will have occurred, because of the action of the delay box, and will have been seen by all variables before y_b switches. The y_q-variables will certainly remain stable throughout the process, since, no matter in what order they see the changes in x and y_p, they will still see

		y_b	0	0	0	0	1	1	1	1
		y_{p1}	0	0	1	1	0	0	1	1
		y_{p2}	0	1	1	0	0	1	1	0
y_a	y_{q1}	y_{q2}								
0	0	0	1	1	1	1	1	2	3	4
0	0	1	2	2^g	2	2	1	2	3	4^h
0	1	1	3	3	3	3	1	2	3	4
0	1	0	4	4	4	4	1	2	3	4
1	0	0	5	6	7	8	5	5	5	5
1	0	1	5^f	6	7	8	6	6	6	6
1	1	1	5	6	7	8	7	7	7	7
1	1	0	5	6	7	8	8	8	8	8

(a)



(b)

Fig. 23. (a) $2S_0$ assignment for an eight-row flow table.
(b) "One-delay-element" circuit with $2S_0$ assignment.

the system in one of the 2-states (in which they are stable), and by the time they see y_b switch they will have seen all of the other changes, so that they will see the system in state h, in which the y_q's are still stable. Since the y_q-variables remain unaltered, similar reasoning will show that none of the other y-variables can ever see the system in any state except a 2-state in the upper left-hand quadrant, or state h, regardless of the effects of stray delays.

The same argument applies to a transition such as the transition from g in row set 2 to row set 5. Here, the y_q-variables and y_b are fixed and the y_p's (in this case only y_{p2}) and y_a change, so that the final state is the one marked "f". All other transitions are of the same form, whereby moves in one dimension within a quadrant through y_p- or y_q-changes are followed by a hop to an adjacent quadrant through y_a- or y_b-changes. A delay box can be shared by y_a and y_b, since in no case do both of these variables have to change during the same transition.

V. CONCLUSION

The basic theory of sequential switching circuits was developed primarily with reference to relay systems, and its application to networks composed of other types of elements came largely as an afterthought (5). In this report we have considered some aspects of the theory that are applicable regardless of the physical components used, but which, for practical reasons, are not of pressing interest in the relay case.

In the analysis of a relay sequential switching circuit there is no problem involved in choosing state variables. We simply select the responses of the secondary relays (those with excitations that are not solely functions of the input). However, in the case of more general networks there may not be any components analogous to secondary relays, and the problem of choosing state variables is not trivial. We have shown in Section II that the solution is to select a set of branches which includes all delay elements, so that cutting all members of this set interrupts all feedback loops. If the signals at the output ends of these branches (called state branches) are chosen as state variables, then the analysis of the circuit may be completed in a routine manner (5).

The role played by amplifiers in sequential switching circuits may be overlooked if only relay systems are considered, because every relay is an amplifier, and hence it is never necessary to worry about the placement of amplifiers in the circuit. This problem cannot, in general, be ignored when other kinds of physical devices are used, and we have shown in Section III that amplifiers are necessary in all feedback loops. Furthermore, the feedback index of a network that realizes a given function is related to the number of rows in the reduced flow matrix corresponding to that function. Usually, the feedback index equals the number of state variables appearing in the reduced flow matrix, but it can be smaller (as in the example of Fig. 10). Since the number of amplifiers needed in a circuit depends upon the feedback index, it is desirable to know how to obtain minimum-index circuits for a given function. Work remains to be done on this problem.

The most important part of this research has been concerned with the effects of stray delays in causing deviations from ideal behavior (called hazards) on the part of sequential switching networks. These effects can always be eliminated by using sufficiently large delay elements in the state branches. Due to the fact that relays act as delay elements, the hazard problem does not usually arise in its most general form in relay systems.

We have shown that there exists a class of sequential functions (easily identifiable) that cannot be properly realized unless delay elements are used. Furthermore, it has been demonstrated that a single delay element is adequate for the proper synthesis of any sequential function. One of the methods presented for a one-delay synthesis requires $2S_0$ state branches for the most general function. No way has been found to reduce this number to $2S_0 - 1$, but it remains to be proved that it is impossible to do so.

The one-delay methods are of theoretical interest, but from a practical point of view the savings in delay elements may not warrant the added circuit complexity which is the price that must be paid. However, in many cases, less general techniques may serve to reduce the number of necessary delay elements without an undue increase in the number of other components. The reasoning employed in Section IV to identify hazard conditions should be useful in attacking specific design problems.

We have been dealing here with the study of asynchronous, dc-level, sequential switching circuits from a theoretical point of view, without regard for specific types of components. It might be useful to extend some of the ideas presented here to asynchronous pulsed circuits, synchronous systems, and other modes of operation of sequential systems. A start has been made in this direction (12,13). From a practical point of view, it would be interesting to study, both analytically and experimentally, switching circuits that are based on various kinds of physical components, in order to determine the extent to which the ideas presented here can be profitably applied. The relative cost of artificially adding delays in different kinds of circuits, as compared with the cost of the other necessary elements, is pertinent to the decision concerning the importance of using design techniques that minimize the required number of delay elements.

Acknowledgement

The author wishes to express his gratitude to his thesis supervisor, Professor D. A. Huffman, for encouragement and guidance, and for suggesting one of the basic problems that was attacked. Professor S. H. Caldwell and Professor R. M. Fano also contributed useful advice and criticism.

References

1. C. E. Shannon, A symbolic analysis of relay and switching circuits, Trans. AIEE 57, 713 (1938).

2. W. Keister, A. Ritchie, and S. Washburn, The Design of Switching Circuits (D. Van Nostrand Company, Inc., New York, 1951).

3. Synthesis of Electronic Computing and Control Circuits, Annals of the Computation Laboratory, Harvard University, Vol. 27 (Harvard University Press, Cambridge, Mass., 1951).

4. M. Karnaugh, Trans. AIEE, Part I, Vol. 72, pp. 593-99 (1953).

5. D. A. Huffman, The synthesis of sequential switching circuits, Technical Report 274, Research Laboratory of Electronics, M.I.T., Jan. 10, 1954.

6. D. A. Huffman, A study of the memory requirements of sequential switching circuits, Technical Report 293, Research Laboratory of Electronics, M.I.T., March 14, 1955.

7. D. A. Huffman, Problems of sequential circuit synthesis, paper presented at West Coast IRE Convention, Los Angeles, California, Aug. 21-25, 1956.

8. S. J. Mason, Feedback theory - Some properties of signal flow graphs, Proc. IRE 41, 1144 (1953).

9. S. H. Unger, A theorem on feedback in sequential switching circuits, Quarterly Progress Report, Research Laboratory of Electronics, M.I.T., April 15, 1955, p. 62.

10. D. A. Huffman, Design of hazard-free switching circuits, Association for Computing Machinery Conference, 1955.

11. R. W. Hamming, Error detecting and error correcting codes, Bell System Tech. J. 29, 147 (1950).

12. S. H. Unger, Timing considerations in clocked switching circuits, Quarterly Progress Report, Research Laboratory of Electronics, M.I.T., Jan. 15, 1956, p. 113.

13. S. H. Unger, A new mode of operation for sequential switching circuits, Quarterly Progress Report, Research Laboratory of Electronics, M.I.T., Jan. 15, 1957, p. 108.