A STUDY OF THE MEMORY REQUIREMENTS OF SEQUENTIAL SWITCHING CIRCUITS

DAVID A. HUFFMAN

TECHNICAL REPORT 293

MARCH 14, 1955

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 Army Signal Corps, the Navy Department (Office of Naval Research), and the Air Force (Office of Scientific Research, Air Research and Development Command), under Signal Corps Contract DA36-039 sc-42607, Project 132B; Department of the Army Project 3-99-12-022.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

RESEARCH LABORATORY OF ELECTRONICS

Technical Report 293 March 14, 1955

A STUDY OF THE MEMORY REQUIREMENTS OF SEQUENTIAL SWITCHING CIRCUITS

David A. Huffman

Abstract

The number of elementary binary memory devices necessary for the realization of an arbitrary asynchronous sequential switching circuit is considered. The least upper bound is discovered to be approximately equal to twice the greatest lower bound.

The minimum conceivable interstate transition time for a sequential circuit is the reaction time of a single memory element. A solution which achieves this minimum time is derived and its relationship to the Hamming single-error correcting code is shown.

The fundamental limitations of error correction schemes which compensate for malfunctioning of memory elements are discussed. These schemes are feasible in synchronous circuits but have slightly impaired practicability in asynchronous circuits.

I. MEANS FOR THE TERMINAL DESCRIPTION OF SWITCHING CIRCUITS

A switching circuit has the property that binary signals appear on each of its input leads and on each of its output leads (Fig. 1). The two possible values of each variable are customarily assigned the notations 0 and 1. The meanings of these two symbols depend upon the physical nature of the binary variables represented. Thus, in one circuit the zero and the one may be interpreted as the absence or presence of a ground, respectively, while in another circuit they may be associated with a low or a high voltage, or the absence or the presence of a voltage pulse.

Switching circuits may be divided into two classes: combinational and sequential. A combinational circuit has no "memory"; that is, the output signals produced by the circuit at any time depend only upon the present combination of signals on the input leads. For a sequential circuit, on the other hand, the output signals may be a function not only of the present input signals but also of, in general, all the previous input signals.

The word-description of the terminal action of either a combinational or a sequential switching circuit may easily be misunderstood or may lack precision if the number of input variables is large, or if the relationship of the output variables to them is logically complex. One accurate means of specifying the dependence of the output variables of a combinational circuit upon its input variables is the table of combinations or truth table. In this kind of tabulation each of the possible combinations of values of the input variables is listed, with the output state it produces explicitly written beside it. For example, the table in Fig. 2 describes a circuit that has three input variables (eight input states) and two output variables. The first output is one if, and only if, just two of the input variables have the value one. The second output is one if, and only if, x_2 and x_3 have complementary (opposite) values, regardless of the value of x_1.

The means for the description of a sequential circuit requires a little more explanation (for a detailed description of the synthesis of sequential circuits, see ref. 1). The output of a sequential circuit depends not only upon the present input state but also upon the accumulated effect of the sequence of all previous input states. This accumulated effect we will call the internal state of the circuit. A complete terminal description of a sequential circuit exists only if: for each internal state the next internal state is known as a function of the next input state; and if for each possible combination of input and internal states the output state is given.

One means of tabulating the data mentioned above is called a flow table. (See Fig. 3(a).) The columns of a flow table are associated with the possible input states to the circuit, and the rows correspond to the internal states. Since the exact means of distinguishing physically among the various internal states is not important for a terminal description of the circuit, these states have merely been assigned decimal numbers. The entry found at the intersection of a row and a column consists of two parts. The first part tells what internal state follows next. The second part lists the output state which results from the given input and internal states. For example, the checked entry

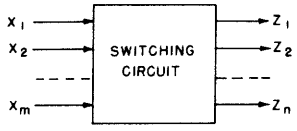


Fig. 1

Schematic diagram of a switching circuit.

X_1	X_2	X_3	Z_1	Z_2
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

Fig. 2

A table of combinations.

Internal state	Input state 00	01	10	11
1	(1)-00	2-00	3-10	4-01✓
2	(2)-00	4-10	1-00	(2)-11
3	(3)-00	1-01	4-11	(3)-11
4	(4)-00	3-11	2-01	1-00

Fig. 3(a)

A flow table.

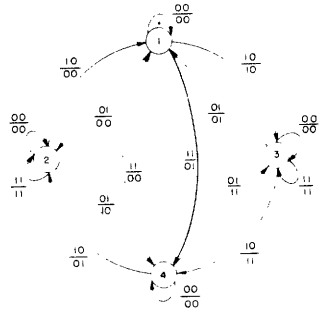


Fig. 3(b)

A flow graph.

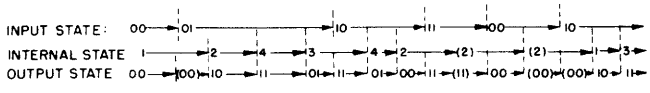


Fig. 3(c)

Sequences derived from Figs. 3(a, b).

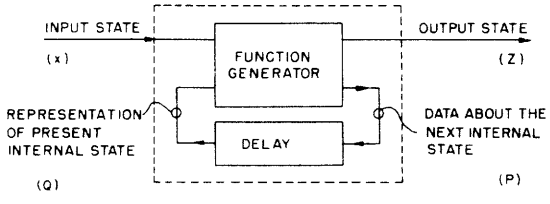


Fig. 4

Functional block diagram of a sequential circuit.

in Fig. 3(a) indicates that, if the input state is 11 and the internal state is "1," the output state is now 01 and the next internal state will be "4" (if the input state remains 11).

If the internal state component of a flow table entry is found in a row associated with that same internal state, the total state (combination of input and internal states) is a stable one, and the entry may be circled to denote this fact. Once circuit action leads to a stable total state, no further changes of internal state can occur until the input state is changed. For all unstable total states the flow table entry tells us what new internal state will follow the present one. If there are no stable total states listed within a given column of a flow table (see the second and third columns of Fig. 3(a)), a recurring cycle of internal states will occur if the circuit input is held fixed at the state corresponding to that column.

An alternate way of listing the data in Fig. 3(a) is the flow graph of Fig. 3(b). In the graph the nodes correspond to internal states, and the directed branches correspond to the transitions listed at an intersection of the flow table. Beside each branch are listed two states; the top designator is the input state; the bottom one is the output state. The heavily lined branch of Fig. 3(b) gives us the same information as does the checked entry of Fig. 3(a).

Figure 3(c) shows the sequences of internal and output states that result from the given sequence of input states if the initial internal state is "1." It is assumed that a definite nonzero time is necessary for changing the internal state. In the example given, when the input changes from its initial state, 00, to 01, the accompanying change of internal state from "1" to "2" does not occur instantaneously. The exact duration of the time lag is of no interest here. However, the ordering in time of the changes in input, internal, and output states is important to us, and the flow table or flow graph allows us to determine what this ordering is – no matter what sequence of input states is applied to the circuit.

A functional diagram that is helpful in understanding the action of a sequential circuit is given in Fig. 4. The function generator (a combinational circuit) is assumed to be instantaneous in its action. For instance, both the input state and the representation of the present internal state may be presented to the function generator as a set of states of operation or nonoperation of relays. The function generator then would consist of a network constructed from contacts on these relays. The ability of this contact network to transmit a ground could then be changed simultaneously with the opening or closing of a contact within the network. Or the function generator might consist of a combinational network designed from rectifiers and vacuum tubes; the response time of this network could be made insignificant in comparison to the delay shown in Fig. 4.

The delay shown in the feedback loop of Fig. 4 will correspond in the sequential circuit itself to the response time of certain elements within the circuit. In a relay switching circuit this delay might be that which exists between the energization of a relay-winding and the subsequent closing of the normally open contacts on the relay. Or again, within an electronic circuit, the delay might be that of an actual delay line.

Actually what we have represented as a delay is a "smoothing" or inertial action. This effectively means that a change of excitation to this kind of memory element must exist for some minimum time in order to cause a change of response. In addition, any feedback loop in which a memory element is connected must have a gain of at least unity. Often this gain must come from the element itself. Regardless of the exact physical nature of the feedback delay, it is its presence that allows a sequential circuit to have a memory.

II. THE NATURE OF THE SECONDARY ASSIGNMENT PROBLEM

In this report we will concentrate our attention upon the manner in which the internal-state information may be coded as sets of values of binary switching variables. We will not be interested in the realization of the function generator itself, since techniques for these realizations already exist. Rather we will consider that the synthesis of a sequential circuit is complete if exact terminal specifications of its function generator have been derived. The realization of the function generator may then proceed, by using, for example, the techniques of Boolean algebra.

The initial step in the synthesis of any circuit is to convert a word statement of its input-output characteristics into a statement in some more explicit form. In a linear circuit this explicit form may be a ratio of polynomials in a complex frequency variable. In a combinational switching circuit it may take the form of a table of combinations or of an expression in Boolean algebra. A flow table may be used to list in detail the terminal properties of a sequential switching circuit.

Once a designer develops the flow table that describes the circuit he wishes to synthesize, two important problems face him. First, how many switching devices need he use in the circuit memory? And second, how is the function generator to be designed so that these devices will be properly controlled? The answers to these questions depend to some extent upon what component devices are used in the memory.

Let us call all devices under the direct control of the circuit input primary devices, and all other devices secondary devices. Of course, the internal state of a circuit must be related to the responses of the secondary devices only.

We will assume, for the time being, that there are available a number of secondary switching devices with precisely equal response times. (This is, of course, impossible to achieve exactly, but the results obtained will show us what problems come about when the response times are not the same.) Since each such device has just two states, a number S of them may have a total of 2^S states among them.

For the flow table of Fig. 3(a) just two secondary devices will be sufficient to give the four internal states required. Let us designate the responses of these two devices with the two binary variables, y_1 and y_2, and assign their four possible states to the internal state designators "1," "2," "3," and "4." (See Fig. 5(a).) Each state designator within the table may then be replaced by the set of secondary variable values assigned

Secondary state, $y = (y_1, y_2)$:		Primary state, $x = (x_1, x_2)$: 00	01	10	11
(1)	00	(1)-00	2-00	3-10	4-01✓
(2)	01	(2)-00	4-10	1-00	(2)-11
(3)	10	(3)-00	1-01	4-11	(3)-11
(4)	11	(4)-00	3-11	2-01	1-00

Fig. 5(a)
Flow table.

y: \ x:	00	01	10	11
00	(00) -00	01-00	10-10	11-01✓
01	(01) 00	11-10	00-00	(01) -11
10	(10) 00	00-01	11-11	(10) -11
11	(11) -00	10 11	01-01	00 -00

Fig. 5(b)
Y-Z matrix.

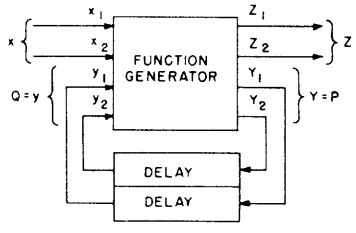


Fig. 5(c)
Hypothetical sequential circuit.

to that state. The resulting matrix is actually a compact form of a table of combinations, and it (Fig. 5(b)) shows the functional dependence of the circuit output, Z, and of the next secondary (internal) state, Y, upon the primary (input) state, x, and the present secondary state, y. The data in the Y-Z matrix may then be used for the design of the function generator shown in the diagram of Fig. 5(c), and the synthesis is complete. The Y-data should be interpreted as the excitations of the secondary devices, as opposed to y, their responses.

The circuit in Fig. 5(c) certainly has the terminal behavior required of it by the original flow table of Fig. 3(a) if the two secondary delays are exactly the same. For example, if x = (1, 1) and y = (0, 0), the next secondary state is designed to be y = (1, 1), and therefore the present excitation of the secondary devices is made to be Y = (1, 1). (See the checked entry of Fig. 5(b).) After some interval of time, the responses of both secondary devices change simultaneously from zero to one, and the transition from y = (0, 0) to y = (1, 1) is accomplished.

Consider now what would happen in the circuit if the two secondary delays were not precisely the same. In that case the transition would actually be y = (0, 0) $\longrightarrow$ y = (1, 0) or y = (0, 0) $\longrightarrow$ y = (0, 1) depending upon whether the first or the second secondary delay were less. For either situation the next input to the function generator would be incorrect and, hence, both the resulting Y-state and Z-state would also be incorrect; that is, the circuit would come to rest in either internal state "2" or internal state "3." The possiblility of this type of undesirable circuit action is called a critical race condition.

It is also possible for a circuit to exhibit a noncritical race condition; that is, one which leads ultimately to the same internal state regardless of the relative delays of the secondary devices. Such noncritical races are illustrated in Section IV. Examples in that section demonstrate a fact stated here without proof: Whenever an internal state transition is accomplished with a noncritical race, the same transition could also be accomplished (without changing the assignment of the secondary states to the rows of the flow table) without a race, by modifying the functions used to control the secondary devices. Because of this fact we will, until further notice, proceed on the assumption that a "proper" assignment of secondary states to the rows of a flow table is one that does not require a race to occur among the secondary devices.

The results of this reasoning may be restated in the following form: A proper secondary assignment is one from which we can derive Y-states that never differ in more than one secondary variable from the y-states they depend upon. Then the relationships among the delays associated with the various secondary devices become unimportant in circuit operation.

The inter-row transitions required by the flow table of Fig. 3(a) cannot be satisfied (without critical race conditions) by two secondary devices no matter how their four possible response states are assigned to the four rows of the table. This conclusion can be justified by noting that transitions are required from the first row of the flow table to each of the three other rows. This in turn requires that the secondary state

designator associated with internal state "1" differ in just one digit from the designators assigned to the three states "2," "3," and "4." This is certainly impossible with just two secondary variables. Therefore any proper secondary state assignment must be based on at least three secondary devices.

III. THE CONCEPT OF INTERMESHED ROW-SETS

In this section a new method is proposed for the solution of the secondary state assignment problem. This method assigns a set of secondary states to each row of the flow table. (Previous methods have insisted that a single state be assigned to each row of the flow table, the table itself being manipulated by procedures such as augmentation and row-splitting so that a proper secondary assignment could be found.) Whenever the secondary response state, y, is one of those that have been associated with a given internal state, we will examine the flow-table row which corresponds to that internal state to find which of several internal states the circuit should produce next. We will then consider that that destination state has been produced whenever the secondary devices have any of the response states whatsoever in the set associated with that destination state.

In Fig. 6(a) we have assigned to the four rows of our illustrative flow table the eight states that are available from three secondary devices. Two secondary states have been assigned to each row. Figure 6(b) is a diagrammatic representation of the assignment; the eight vertices of the diagram are associated with the eight secondary states as well as with decimal designators for the four internal states. The edges of the diagram represent the only permissible changes of secondary state; that is, these edges represent transitions between secondary states in which only one secondary variable changes at a time. This diagram is redrawn in Fig. 6(c) to emphasize the internal states rather than the secondary states.

It is clear from Fig. 6(c) that if the initial secondary state is y = (0, 0, 0) and if it is desired (as it is in the checked entry of the flow table) to make a transition from internal state "1" to state "4," the sequence (0, 0, 0) $\longrightarrow$ (0, 0, 1) $\longrightarrow$ (1, 0, 1) should be made to occur. This is assured by inserting in the Y-Z matrix of Fig. 6(d) the two checked entries shown. Within the same column each of the circled entries corresponds either to internal state "2" or to internal state "3." For these stable total states notice that the Y-state is the same as the y-state, just as in Fig. 5(b).

The flow table tells us that if the input state is x = (1, 0) and if the internal state is "1," the desired change is to internal state "3." The secondary states corresponding to "1" are (0, 0, 0) and (0, 0, 1), and either of the two states (1, 0, 0) or (0, 1, 1) will be a valid destination for the transition. A variety of transitions is possible. Three of these are given in Fig. 6(e). In (i) and (ii) it may take two changes of secondary variable values to leave the internal state "1," depending upon whether we start from (0, 0, 0) or (0, 0, 1). In (iii), however, only one change is ever necessary. The latter transitions are the ones

Internal state:	Secondary state, y:	Input state, x: 00	01	10	11
1	000, 001	(1)-00	2-00	3-10	4-01✓
2	010, 110	(2)-00	4-10	1-00	(2)-11
3	011, 100	(3)-00	1-01	4-11	(3)-11
4	101, 111	(4)-00	3-11	2-01	1-00

Fig. 6(a)
Flow table with secondary state assignment.

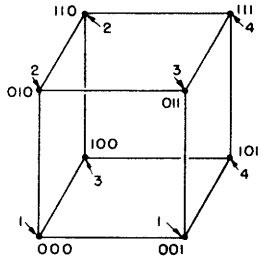


Fig. 6(b)
State diagram.

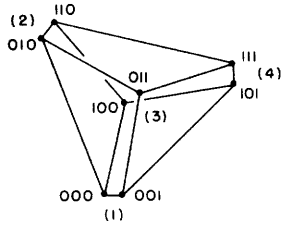


Fig. 6(c)
Modified state diagram.

	y:	x: 00	01	10	11
(1)	000	(000)-00	010-00	100-10	001✓-01
(1)	001	(001)-00	000-00	011-10	101✓-01
(2)	010	(010)-00	110-10	000-00	(010)-11
(3)	011	(011)-00	001-01	111-11	(011)-11
(3)	100	(100)-00	000-01	101-11	(100)-11
(4)	101	(101)-00	100-11	111-01	001 -00
(2)	110	(110)-00	111-10	010-00	(110)-11
(4)	111	(111)-00	011-11	110-01	101 -00

Fig. 6(d)
Y-Z matrix.

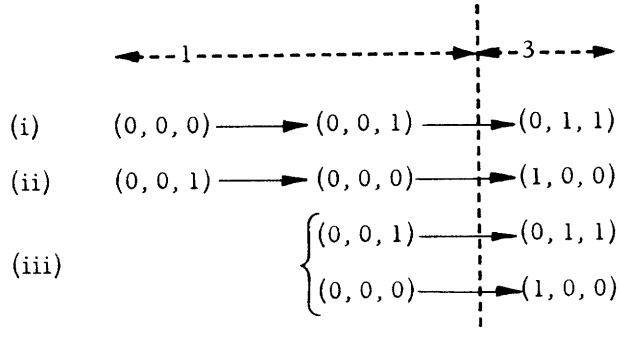


Fig. 6(e)
Alternate transitions possible in Fig. 6(c).

called for by the Y component of the entries in the top two rows of the third column of Fig. 6(d).

The transitions within the flow table of our present example are typical of those occurring when several secondary states are assigned to a single row. Changes of secondary state cause movement within the set of states assigned to the row of origin of the transition, until a state is reached that differs in only one variable from some state assigned to the row of destination; one single change of secondary state will then take us to this destination. Figure 6(d) was completed with this plan in mind.

Let us now define our terms more precisely. Two secondary response states (of a fixed number of variables) are adjacent if their designators differ in just one binary digit. For example (0, 1, 1, 1) is adjacent to (0, 1, 0, 1).

A sequence of states is connected if each consecutive pair of its members is adjacent. (0, 1, 1, 1) → (0, 1, 0, 1) → (1, 1, 0, 1) → (1, 0, 0, 1) is a connected sequence, but (0, 1, 1, 1) → (0, 1, 0, 1) → (1, 0, 0, 1) is not. The lengths of these sequences are three and two, respectively.

A set of states is connected if it is possible, using only states from the set, to construct a connected sequence that joins any member of the set to any other member of the set. For instance [(0, 1, 0, 0); (0, 1, 1, 0); (0, 1, 1, 1); (1, 1, 1, 0)] is a connected set, but [(0, 1, 0, 0); (0, 1, 1, 0); (1, 1, 1, 1); (1, 0, 1, 0)] is not. (We also define as connected, a set which has only one member.)

The distance between two states in a connected set is the length of the shortest possible connected sequence (of states from the set) which joins these two states.

Two connected sets of states are called adjacent if there exists some member of one that is adjacent to some member of the other. [(1, 1, 1, 0); (1, 0, 1, 0)] and [(0, 1, 0, 1); (0, 0, 0, 1); (1, 0, 0, 1); (1, 0, 1, 1)] are adjacent – because of the final members of each set – but [(1, 1, 1, 0); (1, 0, 1, 0)] and [(0, 0, 0, 0); (0, 1, 0, 0)] are not.

The set of secondary response states which is assigned to the i^{th} row of a flow table is called the row-set R_i. The subscript used is the same as that given to the internal state associated with that row. A row-set, R_i, is connected or can be partitioned into two or more connected row-subsets, $R_{i1}, R_{i2}, \ldots$, no two of which are adjacent.

No member of one row-set may also be a member of another row-set. However, two row-sets R_i and R_j will be called meshed if each of the subsets of R_i is adjacent to at least one subset of R_j, and vice-versa.

A collection of row-sets will be called intermeshed if each pair in the collection is meshed. (The row-sets of Fig. 6(a) are intermeshed.)

In the synthesis of a sequential switching circuit we wish to make the assignment of secondary states to the rows of a flow table so that during the transition from an internal state "i" to an internal state "j" no more than one secondary device need change its response state at a given time. This requirement can be met if we assign to the i^{th} and j^{th} rows of the flow table, row-sets R_i and R_j which are meshed. We can easily see that the number of changes of secondary state need never exceed the number of members

of the largest subset of R_i. For example, in Fig. 6(c) and 6(e), a transition from state "1" to state "3" need never be more than two steps long.

If we have an arbitrarily complex flow table of r rows it may be necessary to provide for transitions between all possible pairs of rows. In this case the use of r intermeshed row-sets guarantees a solution to the secondary assignment problem. We may immediately set an upper limit on the number of changes of secondary state necessary for a transition between internal states. This limit is the number of members in the largest row-subset incorporated in the flow table.

In the following sections a number of kinds of intermeshed row-sets will be demonstrated and these will be evaluated according to: (a) the number of secondary variables necessary; and (b) the maximum number of steps which may be required for a transition between internal states.

IV. THE $2S_o + 1$ SOLUTION

The number of secondary states available from S secondary switching devices is 2^S. Since at least r secondary states are necessary for assignment to a flow table having r rows, the smallest conceivable number of secondary devices necessary for the synthesis of the corresponding sequential circuit is S_o, where S_o is defined as the smallest integer satisfying the inequality $2^{S_o} \geq r$.

The first intermeshed row-sets we consider here use $2S_o + 1$ secondary variables. (This $2S_o + 1$ figure appeared in reference 1. However, that result was not obtained from the intermeshed row-set point of view.) That is, 2^{S_o} row-sets will be constructed using $2S_o + 1$ variables. The row-sets have been illustrated for $S_o = 3$, but the reader may easily extend the solution to the general situation. The method of describing row-sets is a tabular one. Each secondary state is associated with an intersection within a matrix. (See Fig. 7.) The entry at the intersection is a decimal number indicating to which row-set the secondary state belongs. The secondary variables have been divided into two groups of three (in general, S_o) variables and an additional variable, y_o. It may easily be verified that each of the eight row-sets have sixteen members and that these members form a single connected set. It is also evident that the members of a row-set fall naturally into a row-half (for $y_o = 1$) and a column-half (for $y_o = 0$). (Notice that the meaning of "row" in the term row-set means a row of the flow table, while in the term row-half it refers to that part of the configuration of entries in Fig. 7 which forms a row-like pattern.)

Let us now see how two arbitrary row-sets are meshed. Recall that for two connected row-sets to be meshed it is necessary only that the two row-sets be adjacent. Consider row-sets R_3 and R_6. These have in common two places of adjacency. The corresponding pairs of states have been labeled d, e and j, k. Any other pair of row-sets can similarly be shown to be adjacent, because the row-half (or column-half) of one is always adjacent to the column-half (or row-half) of the other.

				y_{β_1}	0	0	0	0	1	1	1	1
				y_{β_2}	0	0	1	1	0	0	1	1
y_o	y_{a_1}	y_{a_2}	y_{a_3}	y_{β_3}	0	1	0	1	0	1	0	1
0	0	0	0		0	1	2	3	4	5	6	7
0	0	0	1		0	1	2	3	4	5	6	7
0	0	1	0		0	1	2	3	4	5	6	7
0	0	1	1		0	1	2	3	4	5	6^e	7
0	1	0	0		0	1	2	3	4	5	6	7
0	1	0	1		0	1	2	3	4	5	6	7
0	1	1	0		0	1	2	3^j	4	5	6	7
0	1	1	1		0	1	2	3	4	5	6	7
1	0	0	0		0	0	0	0	0	0	0	0
1	0	0	1		1	1	1	1	1	1	1	1
1	0	1	0		2	2	2	2	2	2	2	2
1	0	1	1		3	3^a	3	3	3^c	3^b	3^d	3
1	1	0	0		4	4	4	4	4	4	4	4
1	1	0	1		5	5	5	5	5	5	5	5
1	1	1	0		6	6	6	6^k	6	6	6	6
1	1	1	1		7	7	7	7	7	7	7	7

Fig. 7

Intermeshed row-sets from $2S_o + 1$ secondary variables.

Next let us determine the maximum necessary inter-row transition time between R_3 and R_6. (This is a typical transition.) Assume that the initial secondary state is that labeled "a." Since the greatest distance between two points in the row-half of R_3 is three, no more than three single changes of secondary state could ever be necessary to arrive at a point of adjacency with row-set R_6. The connected sequence "a" → "b" → "c" → "d" is one which is satisfactory. One additional transition ("d" → "e") takes us to the destination-set R_6. Similar reasoning would be valid if the initial secondary state were in the column-half of a row-set.

The argument given above indicates that the inter-row transition time need never exceed four units for $S_o = 3$. Nevertheless it will be seen below that, if noncritical races are allowed, this time may be reduced to two units. For instance, the transition from

state "a" to state "d" may be accomplished by the use of a race condition in which some or all of the variables y_{β_1}, y_{β_2}, and y_{β_3} may be changing simultaneously. This race would be a noncritical one, since all states which might occur intermediately in the race are also contained in row-set R_3; that is, no matter how the y_β variables are changing, y_o and y_α variables remain fixed. The time required for the race to be completed is the response time of the slowest of the three β-devices. Therefore the effective number of time units necessary for the race to be completed is just one.

Since we have been able to reduce to one unit the time necessary for the transition "a" ⟶ "d" and since the transition "d" ⟶ "e" still takes one time unit, the total time necessary for the transition from R_3 to R_6 is two units. Because these row-sets are typical ones, we may put an upper limit of two time units on inter-row transition times associated with the $2S_o + 1$ secondary state assignment.

It is interesting to put ourselves in the place of the seven secondary devices used in Fig. 7 and imagine what decisions we might be called upon to make. Fundamentally our job can be considered to be that of accepting coded data about the next internal state at our input terminals and, after some delay, of transferring this code to our output terminals. This was the idea underlying the representations of the delay-circuits given in Figs. 4 and 5(c). In Fig. 5(c), however, it was our assumption that the delays in the feedback loop resulted from a hypothetical pair of identical secondary devices whose response times were exactly equal. The circuit we are now designing, on the other hand, must use $2S_o + 1$ devices whose response times cannot be assumed equal, but which do interact with each other so that they, collectively, produce the effect of S_o hypothetical devices.

Let the circuit we represent (see Fig. 8) have S_o two-valued inputs, called $p_1, p_2, \ldots, p_{S_o}$ and S_o two-valued outputs, called $q_1, q_2, \ldots, q_{S_o}$. The values of these variables will be interpreted as binary numbers that correspond to the destination row-set and the present row-set, respectively. For example, if $S_o = 3$, $(p_1, p_2, p_3) = (1, 1, 0)$, and $(q_1, q_2, q_3) = (0, 1, 1)$, we interpret from the values of the p-variables that the next desired internal state is $1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 =$ "6" and, from the value of the q-variables that the present internal state is $0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 =$ "3."

In Fig. 8 the seven secondary devices are symbolized as delays. All other component parts of the diagram represent means of generating various combinational circuits. The boxes marked T have outputs that can be selected as equal to one of two sets of inputs, depending upon the value of a controlling variable (y_o in the present circuit). For example, T_1 assures that when $y_o = 1$ (the situation shown in Fig. 8), the α-devices are self-excited and therefore stable. When $y_o = 0$, the α-devices are excited from the inputs, p_1, p_2, and p_3. A similar use is made of T_2.

By means of T_3 the outputs, q_1, q_2, and q_3, are chosen as equal to the responses of the α- or β-devices depending upon whether the value of y_o is one or zero, respectively. T_4 and the coincidence circuit, C_1, working together, produce a signal c_1 which is unity when $y_o = 0$ and $(y_{\alpha_1}, y_{\alpha_2}, y_{\alpha_3}) = (p_1, p_2, p_3)$, or when $y_o = 1$ and

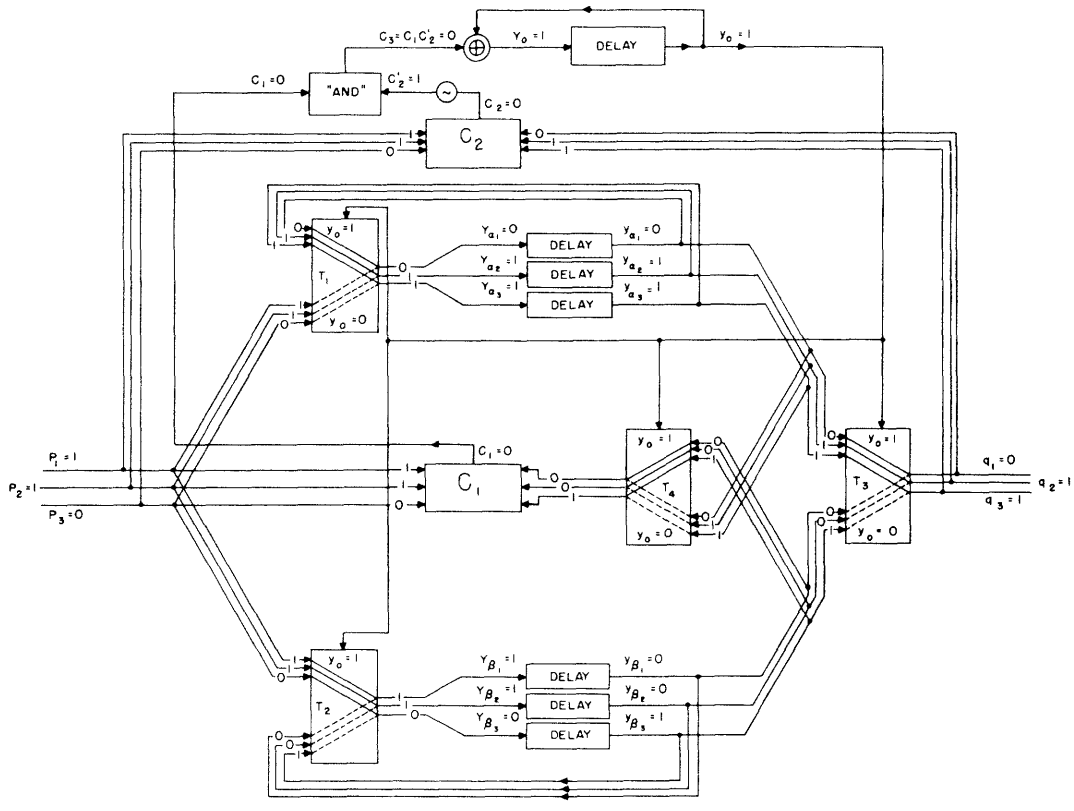


Fig. 8

Diagram of a delay circuit using $2S_o + 1$ devices.

$(y_{\beta_1}, y_{\beta_2}, y_{\beta_3}) = (p_1, p_2, p_3)$, but which is zero otherwise.

Another coincidence circuit, C_2, and an inversion (or complementation) operation produce a signal c_2 which is unity when and only when $(p_1, p_2, p_3) = (q_1, q_2, q_3)$, and another signal c_2' which is zero when and only when $(p_1, p_2, p_3) = (q_1, q_2, q_3)$.

An "and" circuit produces an output c_3 which is unity if and only if both c_1 and c_2' are unity. The signal c_3 and the response y_o are added, modulo two, to generate the excitation of the $\mathscr{Y}_o$ device. Or, in equivalent terms, the excitation Y_o is unity if the values of c_3 and y_a differ, and is zero if they are the same. The net effect is that $\mathscr{Y}_o$ is unstable or stable as c_3 has the value one or zero.

The digits zero and one that have been placed beside the various inputs and outputs of the components of Fig. 8 are the values of signals corresponding to the entry marked "a" in Fig. 7. Note that the α-devices and $\mathscr{Y}_o$ are stable, but that all three of the β-devices are unstable. The associated race condition, already described, will ultimately (essentially in a single time unit) result in the set of values $y_{\beta_1}, y_{\beta_2}, y_{\beta_3}$ being equal to the set of values $Y_{\beta_1}, Y_{\beta_2}, Y_{\beta_3}$. (This corresponds to the entry "d" in Fig. 7.) As soon as this happens, c_1 becomes unity, c_3 becomes unity, and $\mathscr{Y}_o$ becomes unstable.

When the value of y_o becomes zero, $\mathscr{Y}_o$ is stable and the outputs of the four T

circuits become those corresponding to $y_o = 0$. Of particular interest to us is that the set of outputs q_1, q_2, and q_3 become equal to y_{β_1}, y_{β_2}, and y_{β_3} — which are now 1, 1, and 0, respectively. This set of variable values is interpreted as the decimal number "6" = $1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$. Thus the row-set R_6 has been entered at the state "e" in Fig. 7, and the total time for the circuit action (which started from the state "a") has been two time units.

Some additional circuit simplicity could be gained by omitting the coincidence circuit C_2, the inverting circuit, and the "and" circuit, and letting $c_3 = c_1$. However, the device $\mathscr{Y}_o$ would then be continuously unstable for the situation where $(p_1, p_2, p_3) = (y_{a_1}, y_{a_2}, y_{a_3}) = (y_{\beta_1}, y_{\beta_2}, y_{\beta_3}) = (q_1, q_2, q_3)$. If $\mathscr{Y}_o$ represents a relay the corresponding "buzzer" action would undoubtedly be considered undesirable. On the other hand if $\mathscr{Y}_o$ represents a delay-line or an Eccles-Jordan circuit the instability may be allowable.

The circuit we have derived has S_o inputs and S_o outputs. In an electronic circuit these would represent actual leads upon which binary signals appear. For a relay circuit Fig. 8 merely shows how the q's are related to the responses of the secondary relays and how the Y's are related to the p-functions and the secondary responses, y. Since (see Fig. 4) the p's and the Z's can be determined as functions of the x's and the q's, it follows that the Y's and the Z's can also be found as functions of the x's and the y's.

We may summarize circuit action as follows: Two banks of storage (the α-devices and the β-devices) may handle S_o digits each, and are controlled by a signal y_o. The devices in one of the two storage banks are always stable and the circuit output is taken from them. The devices in the other storage bank are excited from the input. The signal y_o is controlled so that it never shifts the output connection from one storage to the other unless sufficient time has elapsed for the input to be stored in the latter.

If the input (p) were to change only during predetermined regular intervals of time and if y_o were to change only after it was certain that a new input had been stored (in the proper storage bank), then the desired circuit action would result. If y_o is a periodically changing signal generated outside the circuit (as in Fig. 9) rather than controlled from the responses of the two storage banks (as in Fig. 8), we will call the circuit so controlled an escapement circuit in contrast to the circuit of Fig. 8, which we will now call simply a "delay" circuit, since it replaces a number of idealized delays.

The escapement circuit of Fig. 9 allows its input to "escape" to the output whenever y_o changes value. For its proper operation it is necessary that the time between changes of value of y_o be sufficiently greater than the response times of the devices within the α- and β-storage banks. In effect this circuit can be seen to consist of S_o different single-input, single-output escapement circuits kept synchronized with each other by means of the y_o signal. These more elementary circuits are what the computer designer would call single-stage shift registers; the y_o signal corresponds to the shift signal. In the design of most shift registers, however, it is assumed that the relative ordering of the delays of the two component elements is fixed, whereas our circuit will operate properly no matter what this ordering is. (See, for example, ref. 2.)

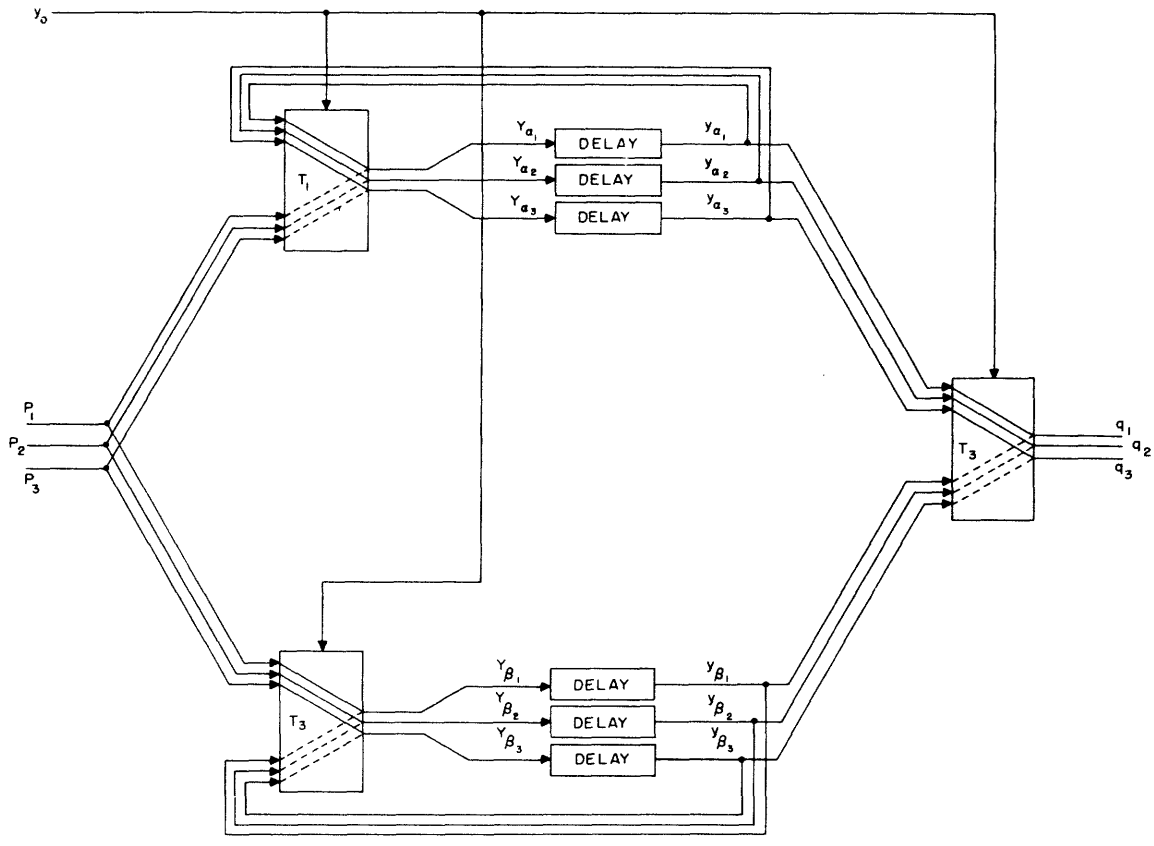
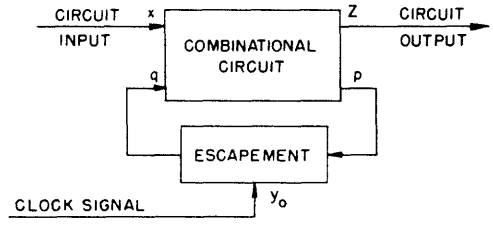
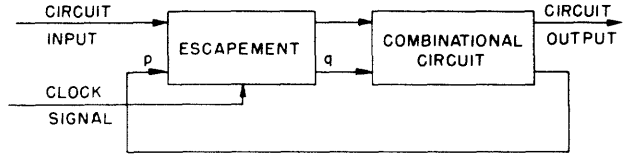


Fig. 9

Diagram of an escapement circuit derived from Fig. 8.



(a)



(b)

Fig. 10

Two forms of synchronous circuits.

When an escapement circuit is used in the feedback link of a sequential circuit (see Fig. 10(a)) the circuit may be called clocked or synchronous. (When a delay circuit is used in the feedback link the circuit may be called asynchronous or free-running.) Sometimes it is convenient to send the circuit input itself, as well as the internal-state code, through an escapement circuit. This makes possible less accurate timing of the changes of variable values on the circuit input leads, since these variations are effectively removed in the escapement circuit.

It is sometimes erroneously thought that the use of a synchronous circuit allows fewer secondary devices to be used than does an asynchronous circuit. Careful investigation of these synchronous circuits will reveal that $2S_o$ delays are necessary for the operation of the circuit, even though perhaps half of these delays may not be associated with specific switching devices. Fundamentally it will be seen (see Sec. V) that it takes essentially $2S_o$ delays for the conversion of the internal state data coded as p to the internal state data coded as q, regardless of how y_o is obtained.

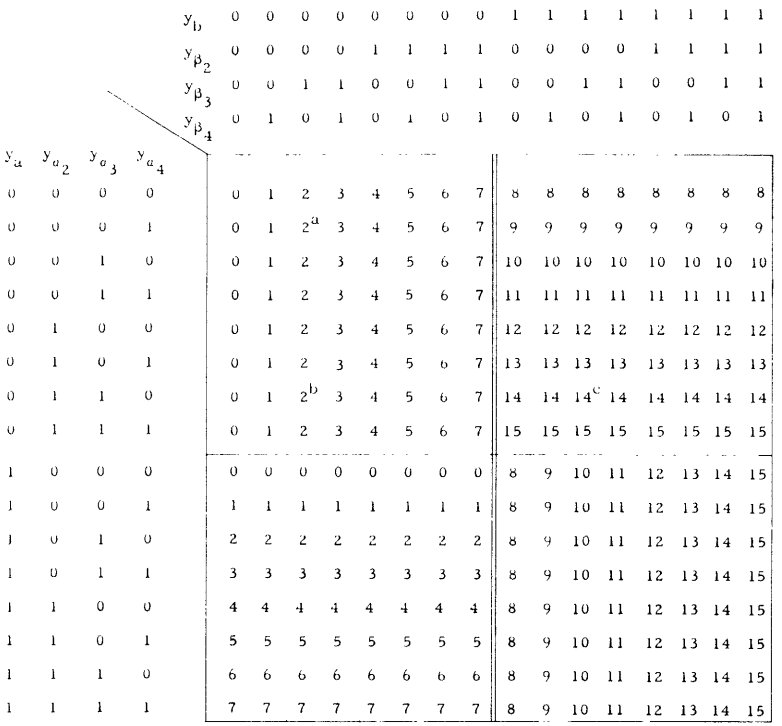
V. THE $2S_o$, $2S_o - 1$, AND $2S_o - 2$ SOLUTIONS

The purpose of the preceding sections was to demonstrate that intermeshed row-sets will always furnish a solution to the secondary assignment problem, that the solution using $2S_o + 1$ secondary devices can be realized in a relatively simple circuit, and that there is a simple relationship between this solution and what are called synchronous circuits. In this section we attempt to determine assignments that use as few secondary devices as are necessary for the realization of the generalized asynchronous circuit, regardless of whether or not these assignments have simple corresponding escapement circuit realizations.

Sixteen intermeshed row-sets that require eight binary variables are shown in Fig. 11. (In the general case 2^{S_o} sets would require $2S_o$ variables.) Each row-set has sixteen members, and as in the $2S_o + 1$ assignment, each falls naturally into a row-half and a column-half which are adjacent. The reader may verify that the row-half (column-half) of each row-set is adjacent to the column-halves (row-halves) of each other row-set, and that therefore the row-sets are intermeshed.

The transition time necessary between any two row-sets may be seen from the typical transition from state "a" in row-set R_2 to the nearest state "c" in row-set R_{14}, via the state "b." Two time units are necessary: the first is required for the race of the y_a-variables from (0, 0, 1) to (1, 1, 0); the second unit is required for the change of response of y_b from 0 to 1.

A modification of the preceding assignment is given in Fig. 12, in which eight variables are used for the generation of twenty-four intermeshed sets. (In the general case, $0.75 \cdot 2^{S_o}$ sets would require $2S_o - 2$ variables.) The maximum transition time necessary between row-sets may occur if the initial state is found in the lower half of the matrix (that is, for $y_a = 1$) and if the destination row-set is in the upper half of the



y_a	y_{a_2}	y_{a_3}	y_{a_4}	y_b	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
				y_{β_2}	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
				y_{β_3}	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				y_{β_4}	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0	0	0		0	1	2	3	4	5	6	7	8	8	8	8	8	8	8	8
0	0	0	1		0	1	2^a	3	4	5	6	7	9	9	9	9	9	9	9	9
0	0	1	0		0	1	2	3	4	5	6	7	10	10	10	10	10	10	10	10
0	0	1	1		0	1	2	3	4	5	6	7	11	11	11	11	11	11	11	11
0	1	0	0		0	1	2	3	4	5	6	7	12	12	12	12	12	12	12	12
0	1	0	1		0	1	2	3	4	5	6	7	13	13	13	13	13	13	13	13
0	1	1	0		0	1	2^b	3	4	5	6	7	14	14	14^c	14	14	14	14	14
0	1	1	1		0	1	2	3	4	5	6	7	15	15	15	15	15	15	15	15
1	0	0	0		0	0	0	0	0	0	0	0	8	9	10	11	12	13	14	15
1	0	0	1		1	1	1	1	1	1	1	1	8	9	10	11	12	13	14	15
1	0	1	0		2	2	2	2	2	2	2	2	8	9	10	11	12	13	14	15
1	0	1	1		3	3	3	3	3	3	3	3	8	9	10	11	12	13	14	15
1	1	0	0		4	4	4	4	4	4	4	4	8	9	10	11	12	13	14	15
1	1	0	1		5	5	5	5	5	5	5	5	8	9	10	11	12	13	14	15
1	1	1	0		6	6	6	6	6	6	6	6	8	9	10	11	12	13	14	15
1	1	1	1		7	7	7	7	7	7	7	7	8	9	10	11	12	13	14	15

Fig. 11

The $2S_o$ secondary assignment.

y_a	y_{a_3}	y_{a_4}	y_{a_5}	y_b	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
				y_{β_3}	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
				y_{β_4}	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				y_{β_5}	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0	0	0		0	1	2	3	4	4	4	4	8	8	8	8	8	9	10	11
0	0	0	1		0	1^i	2	3	5	5	5	5	9	9	9	9	8	9	10	11
0	0	1	0		0	1^h	2	3	6	6^g	6	6	10	10	10	10	8	9	10	11
0	0	1	1		0	1	2	3	7	7	7	7	11	11	11	11	8	9	10	11
0	1	0	0		0	0	0	0	4	5	6	7	12	13	14	15	12	12	12	12
0	1	0	1		1	1^j	1^k	1	4	5	6	7	12	13	14^ℓ	15	13	13	13	13
0	1	1	0		2	2	2	2	4	5	6	7	12	13	14	15	14	14	14	14
0	1	1	1		3	3	3	3	4	5	6	7	12	13	14	15	15	15	15	15
1	0	0	0		16	16	16	16	16	17	18	19	16	17	18	19	20	20	20	20
1	0	0	1		17	17^d	17	17	16	17^e	18	19	16	17^c	18	19	21	21	21	21
1	0	1	0		18	18	18	18	16	17^f	18	19	16	17	18	19	22	22	22	22
1	0	1	1		19	19	19	19	16	17	18	19	16	17	18	19	23	23	23	23
1	1	0	0		20	21	22	23	20	20	20	20	16	16	16	16	20	21	22	23
1	1	0	1		20	21	22	23	21	21	21	21	17	17^b	17^a	17	20	21	22	23
1	1	1	0		20	21	22	23	22	22	22	22	18	18	18	18	20	21	22	23
1	1	1	1		20	21	22	23	23	23	23	23	19	19	19	19	20	21	22	23

Fig. 12

$2S_o - 2$ secondary assignment.

y_a	y_b	y_{α_3}	y_{α_4}	y_{α_5}	y_c = 0	y_{β_3} = 0	y_{β_4} = 0	y_{β_5} = 0	y_c = 1	y_{β_3} = 0	y_{β_4} = 0	y_{β_5} = 1
0	0	0	0	0	0	1	2	3	4	5	6	7
0	0	0	0	1	0	1	2	3	4	5	6	7^d
0	0	0	1	0	0	1	2	3	4	5	6	7
0	0	0	1	1	0	1	2	3	4	5	6	7
0	0	1	0	0	0	1	2	3	4	5	6	7
0	0	1	0	1	0	1	2	3	4	5	6	7
0	0	1	1	0	0	1	2	3	4	5	6	7
0	0	1	1	1	0	1	2	3	4	5	6	7^c
0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	1	1	1	1	1	1	1
0	1	0	1	0	2	2	2	2	2	2	2	2
0	1	0	1	1	3	3	3	3	3	3	3	3
0	1	1	0	0	4	4	4^q	4	4	4	4	4
0	1	1	0	1	5	5	5	5	5	5	5	5
0	1	1	1	0	6	6	6	6	6	6	6	6
0	1	1	1	1	7	7	7^a	7	7	7	7	7^b
1	0	0	0	0	16	16	16	16	16	16	16	16
1	0	0	0	1	17	17^f	17	17	17	17	17	17^e
1	0	0	1	0	18	18	18	18	18	18	18	18
1	0	0	1	1	19	19	19	19	19	19	19	19
1	0	1	0	0	20	20	20	20	20	20	20	20
1	0	1	0	1	21	21	21	21	21	21	21	21
1	0	1	1	0	22	22	22	22	22	22	22	22
1	0	1	1	1	23	23	23	23	23	23	23	23
1	1	0	0	0	24	25	26	27	28	29	30	31
1	1	0	0	1	24	25	26	27	28	29	30	31
1	1	0	1	0	24	25	26^o	27	28	29	30	31
1	1	0	1	1	24	25	26	27	28	29	30	31
1	1	1	0	0	24	25	26^p	27	28	29	30	31
1	1	1	0	1	24	25	26	27	28	29	30	31
1	1	1	1	0	24	25	26	27	28	29	30	31
1	1	1	1	1	24	25	26	27	28	29	30	31

y_c = 1	y_{β_3} = 1	y_{β_4} = 1	y_{β_5} = 1	y_c = 1	y_{β_3} = 1	y_{β_4} = 1	y_{β_5} = 1
8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9
10	10	10	10	10	10	10	10
11	11	11	11	11	11	11	11
12	12	12	12	12	12	12	12
13	13^i	13	13	13^j	13	13	13
14	14	14	14	14	14	14	14
15	15	15	15	15	15	15	15
8	9	10	11	12	13	14	15
8	9	10	11	12	13	14	15
8	9	10	11	12^ℓ	13	14	15
8	9	10	11	12	13	14	15
8	9	10	11	12	13	14	15
8	9	10	11	12^k	13	14	15
8	9	10	11	12	13	14	15
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
16	17^g	18	19	20	21	22	23
16	17	18	19	20	21	22	23
16	17	18	19	20	21	22	23
16	17	18	19	20	21	22	23
16	17^h	18	19	20	21	22	23
16	17	18	19	20	21	22	23
16	17	18	19	20	21	22	23
24	24	24	24	24	24	24	24
25	25	25	25	25	25	25	25
26	26	26^n	26	26^m	26	26	26
27	27	27	27	27	27	27	27
28	28	28	28	28	28	28	28
29	29	29	29	29	29	29	29
30	30	30	30	30	30	30	30
31	31	31	31	31	31	31	31

Fig. 13

The $2S_o - 1$ secondary assignment.

matrix (for $y_a = 0$). The changes of state "a" ⟶ "b" ⟶ "c" ⟶ "d" ⟶ "e" ⟶ "f" ⟶ "g" indicated in Fig. 12 represent a situation requiring this maximum transition time, which is just six units. The changes "a" ⟶ "b" and "e" ⟶ "f" involve races, among the β-devices and α-devices, respectively, but the remaining transitions are associated with changes in state of single variables. Transitions between certain other row-sets may require only four time units. For example,

"h" → "i" → "j" → "k" → "ℓ." (Remember that any member whatsoever of R_{14} is a proper destination for the transition.)

Finally, in Fig. 13, thirty-two row-sets needing nine variables for their construction are listed. In general, the structure listed will allow as many as 2^{S_o} intermeshed row-sets to be constructed with only $2S_o - 1$ variables. The longest time necessary for transition between row-sets is four units. The alphabetic sequence of states is an extended sequence of transitions leading from an initial state in R_7, through $R_{17}, R_{13}, R_{12}, R_{26}$, and finally to R_4.

Figure 14 lists the fewest number of secondary variables sufficient for realization of a given number of intermeshed row-sets, which can be found by utilizing the results of Figs. 12 and 13.

In order to investigate the possibility of further reduction in the number of secondary devices necessary for the formation of a given number of intermeshed row-sets we

Number of rows in flow table	S_o	Number of secondary variables sufficient for realization
1	0	0
2	1	1
3	2	2
4	2	3
5, 6	3	4
7, 8	3	5
9, 10, 11, 12	4	6
13, 14, 15, 16	4	7
17, 18, 19, 20, 21, 22, 23, 24	5	8
25, 26, 27, 28, 29, 30, 31, 32	5	9
.	.	.
.	.	.
.	.	.
$0.50 \cdot 2^{S_o} + 1, \ldots, 0.75 \cdot 2^{S_o}$	S_o	$2S_o - 2$
$0.75 \cdot 2^{S_o} + 1, \ldots, 1.00 \cdot 2^{S_o}$	S_o	$2S_o - 1$

Fig. 14

Upper bounds on the number of secondary variables necessary for the realization of a sequential switching circuit.

consider what data must be stored as binary variables within the collection of devices.

Assume that in a flow table with 2^{S_o} rows any inter-row transition is possible and that some fixed intermeshed row-set assignment has been made. Somewhere in the transition from an arbitrary row-set R_i to some other arbitrary row-set R_j is a time during which a single secondary variable, say y_v, changes its value so that the secondary state just before the change belongs to R_i and just after the change to R_j. Since an intermeshed assignment has been made, there is some rule (perhaps a complex one) by which we can examine a secondary state and tell which of the 2^{S_o} row-sets it belongs to; that is, we can, from the secondary state, derive a binary number having S_o independent digits.

Consider the application of this rule just before and just after the change of y_v. Just before the change, y_v could be used as one of the S_o binary digits which helped to determine the number "i." Just after the change y_v could be used as one of the S_o binary digits which helped to determine the number "j." If representations of these two independent numbers had to exist simultaneously among the secondary devices, a total of $2S_o$ secondary variables would be necessary. However, the y_v variable can be used first in the representation for "i," and, one step later, in the representation for "j." This double usefulness of some secondary variable allows a decrease in the required total number of secondary variables to $2S_o - 1$.

A further extension of this argument indicates that, if as many as S_r devices could become unstable simultaneously without creating a critical race condition, only $2S_o - S_r$ devices would be necessary for the secondary assignment. In the particular case where $S_r = S_o$ it follows that $2S_o - S_r = S_o$. Or, in other words – if S_o devices could become unstable at the same time without producing a critical race – no additional secondary devices would be required. This is the hypothetical situation discussed in Section II.

Thus Fig. 14 lists essentially the least conceivable number of secondary devices sufficient for the realization of a completely arbitrary flow table.

VI. THE $2^{S_o} - 1$ SOLUTION

In the last section we determined that essentially $2S_o - 1$ secondary variables are necessary for the formation of 2^{S_o} intermeshed row-sets and that the corresponding inter-row transition time may be as high as four or six units. Here we ask how low the inter-row transition time for intermeshed sets may be, and how many devices are necessary for the realization of the corresponding delay circuit.

When a secondary state change leads out of a row-set R_i into a row-set R_j the transition may be divided into two parts: The first is a motion within R_i until a point of adjacency with R_j is reached; the second is the additional change of state of a single secondary variable whose action is associated with the crossing of the "boundary" between R_i and R_j. The latter transition cannot possibly be eliminated. The former motion will be unnecessary only if the row sub-set of R_i which we are leaving consists

				y_1	0	0	0	0	1	1	1	1
				y_2	0	0	1	1	0	0	1	1
y_4	y_5	y_6	y_7	y_3	0	1	0	1	0	1	0	1
0	0	0	0		0	3	2	1	1	2	3	0
0	0	0	1		7	4	5	6	6	5	4	7
0	0	1	0		6	5	4	7	7	4	5	6
0	0	1	1		1	2	3	0	0	3	2	1
0	1	0	0		5	6	7	4	4	7	6	5
0	1	0	1		2	1	0	3	3	0	1	2
0	1	1	0		3	0	1	2	2	1	0	3
0	1	1	1		4	7	6	5	5	6	7	4
1	0	0	0		4	7	6	5	5	6	7	4
1	0	0	1		3	0	1	2	2	1	0	3
1	0	1	0		2	1	0	3	3	0	1	2
1	0	1	1		5	6	7	4	4	7	6	5
1	1	0	0		1	2	3^b	0	0	3	2	1
1	1	0	1		6	5	4	7	7	4	5	6
1	1	1	0		7	4	5^a	6	6	5	4	7
1	1	1	1		0	3	2	1	1	2	3	0

Fig. 15
The $2^{S_o} - 1$ secondary assignment.

of just a single member.

If we are to reduce all inter-row transition times to one unit it is necessary that the sub-sets of each row sub-set consist of only one member and that each such sub-set be adjacent to a sub-set from each of the other row-sets. It follows that, for 2^{S_o} row-sets to be intermeshed, there must be at least $2^{S_o} - 1$ secondary variables. Figure 15 lists such an assignment for the case $S_o = 3$.

It may be verified by the reader that the row-sets of Fig. 15 are intermeshed. A typical inter-row transition (from R_5 to R_3) is indicated. During this transition (from state "a" to state "b") only the response variable y_6 changes, and hence only a single time unit is required.

The $2^{S_o} - 1$ solution is related to the single-error correcting Hamming code (see ref. 3). This may be seen by examining the method used for the generation of decimal

entries of Fig. 15. Each decimal entry has an equivalent binary notation, q_1, q_2, q_3. For instance the states labeled "5" could have been given the binary representation $(q_1, q_2, q_3) = (1, 0, 1)$, since $1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 =$ "5." What interests us here is how q_1, q_2, and q_3 are related to the seven secondary response variables $y_1, y_2, \ldots,$ and y_7. The relationships below have been chosen:

$$q_1 = y_4 \oplus y_5 \oplus y_6 \oplus y_7$$

$$q_2 = y_2 \oplus y_3 \oplus y_6 \oplus y_7$$

$$q_3 = y_1 \oplus y_3 \oplus y_5 \oplus y_7$$

where $\oplus$ means addition modulo-two. In equivalent terms, q_1 is unity if, and only if, an odd number of the variables y_4, y_5, y_6, and y_7 have the value one, and so forth. For the state labeled "a" the secondary variables have the values

$$y_1 = 0,\ y_2 = 1,\ y_3 = 0,\ y_4 = 1,\ y_5 = 1,\ y_6 = 1, \text{ and } y_7 = 0.$$

Consequently

$$q_1 = 1 \oplus 1 \oplus 1 \oplus 0 = 1$$

$$q_2 = 1 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$q_3 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$$

and the decimal number 5 has been entered in the matrix rather than the more cumbersome binary number 101.

The combinations of values of the y-variables will give values of q_1, q_2, and q_3 equal to zero if the corresponding modulo-two sums (parity checks) are zero. Thus the states in the row-set R_0 are the same as Hamming's undisturbed message codes.

The functional diagram of a delay circuit based on the assignment of Fig. 15 is given in Fig. 16. Each circle represents a modulo-two addition operation. The block labeled "tree" gives a unity output on one of its seven output leads depending on the combination of the values of r_1, r_2, and r_3 (except for $r_1 = r_2 = r_3 = 0$, in which none of the seven output leads carries the unity signal).

The situation shown in Fig. 16 corresponds to the state "a" in the preceding figure, and it is assumed that it is desired to produce next the internal state "3" – $(p_1, p_2, p_3) = (0, 1, 1)$ – from the present internal state "5" – $(q_1, q_2, q_3) = (1, 0, 1)$. Corresponding components of (p) and (q) are compared. The result (r) of the comparison causes a unity output at the sixth output of the tree. This unity output makes the sixth secondary device unstable. As soon as y_6 becomes equal to zero the modulo-two sums which generate q_1 and q_2 will change in value and (q_1, q_2, q_3) will become (0, 1, 1), as desired.

We have proved that it is possible to make a delay circuit with an over-all response time of only one unit, and that the number of devices necessary is $2^{S_o} - 1$, a number

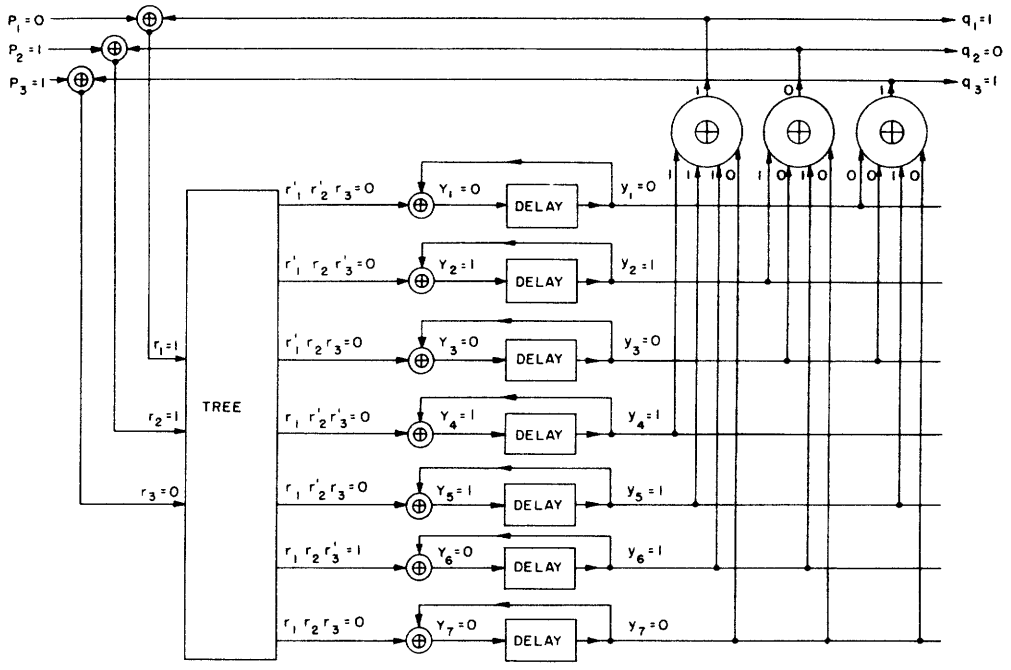


Fig. 16
Block diagram for the realization of the $2^{S_o} - 1$ delay circuit.

which is essentially the same as the number of rows in the flow table describing the sequential circuit we wish to realize.

VII. AN ERROR-CORRECTING SOLUTION

In the last section mention was made of the relationship of the $2^{S_o} - 1$ solution to the Hamming single-error correcting codes. Error-correcting codes may be used in a more conventional way among the secondary devices of a sequential switching circuit. In this section we investigate in an abstract way just what error-correcting schemes can and cannot be expected to do in both asynchronous and synchronous circuits.

For the purposes of this section an "error" will mean an improper response of some secondary device. That is, a relay may fail to operate because of an open winding, an Eccles-Jordan circuit may not change state when its excitation is changed, or a delay line may become inoperative. Error-correction will be said to be effective only if the corresponding delay or escapement circuit still gives the desired terminal action, in spite of the malfunctioning of a single secondary element; error-correction does not mean here that the malfunction itself is corrected, but only that its effect is counteracted as far as the terminal action of the circuit is concerned.

In the asynchronous case a number of error-correcting plans have been investigated. The one associated with the intermeshed row-sets of Fig. 17 has typical advantages and limitations, and is used only for the demonstration of these. Eight row-sets exist; each consists of two sub-sets which are not adjacent to each other. Each sub-set has a center

	y_1	0	0	0	0	1	1	1	1
	y_2	0	0	1	1	0	0	1	1
	y_3	0	1	0	1	0	1	0	1
y_4	y_5	y_6	y_7						
0	0	0	0	0_0	0_1	0_2	7	0_3	7	7	7
0	0	0	1	0_4	6	2	3	4_a	5	1	7
0	0	1	0	0_5	1	5	3	4	2	6	7
0	0	1	1	4	3	3	3	4	4	4	3
0	1	0	0	0_6	1	2	4	3_4	5	6	7
0	1	0	1	2	5	2	2	5_b	5	2	5
0	1	1	0	1	1	6	1	6	1	6	6
0	1	1	1	7	1	2	3	4	5	6	0
1	0	0	0	0_7	6	5	4	3_5	2	1	7
1	0	0	1	6	6	1	6	1	6	1	1
1	0	1	0	5	2	5	5	2	2	5	2
1	0	1	1	7	6	5	3	4	2	1	0
1	1	0	0	3_1	4	4	4	3_0	3_2	3_3	4
1	1	0	1	7	6	2	4	3_6	5	1	0
1	1	1	0	7	1	5	4	3_7	2	6	0
1	1	1	1	7	7	7	0	7	0	0	0

Fig. 17
An error-correcting secondary assignment.

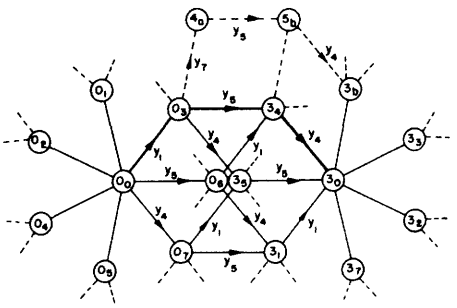


Fig. 18
A state diagram of a fraction of Fig. 17.

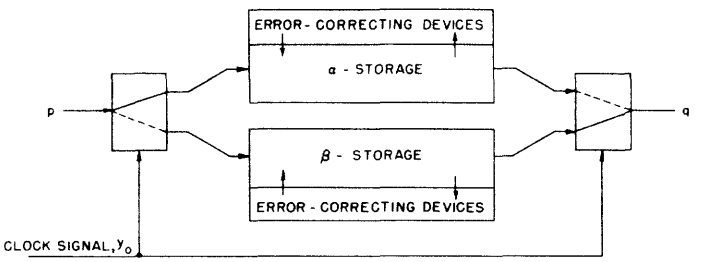


Fig. 19
Error-corrected escapement circuit.

surrounded by seven other states belonging to the same sub-set. The centers have been indicated in the matrix by the heavily outlined cells and they correspond to the error-free codes mentioned in the preceding section; they are states for which each of the three Hamming parity checks gives a zero result. The center of each row sub-set is at a distance three from one or the other of the two centers of each other row-set.

It is assumed that both the origin and destination of a transition between any two row-sets will be centers of sub-sets. For example, if the secondary state is that marked "0_o" and if the destination row-set is R_3, the secondary devices will be excited so that they ultimately reach the state "3_o." The situation may be better understood by looking at Fig. 18. The variable associated with a branch connecting two nodes in this diagram shows in what variable the two states differ; the direction of the arrow indicates the change in value from zero to one. The states "0_o" and "3_o" differ in the values of the three variables y_1, y_4, and y_5. Therefore it is possible, starting from "0_o," to set up a noncritical race condition with the destination "3_o" in which $Y_1 = Y_4 = Y_5 = 1$. Assume that the response times of the three devices are related so that the actual transition is the one which is heavily lined.

If some malfunctioning of, say, $\mathscr{Y}_7$, occurs just after state "0_3" appears, the dotted transition leading to "3_6" will occur. The ultimate row-set will be the one desired, but in the meantime row-sets R_4 and R_5 have been entered. In the corresponding delay circuit improper outputs would therefore occur.

If, instead, the transition desired reaches the state "3_4" and the $\mathscr{Y}_5$ device fails in such a way that $y_5 \longrightarrow 0$ the resulting sequence of states will be

$$"0_o" \longrightarrow "0_3" \longrightarrow "3_4" \longrightarrow "0_3" \longrightarrow "3_5"$$

($\mathscr{Y}_5$ fails)

Again the ultimate row-set will still be R_3, but again a transitory improper output of the delay circuit will occur.

Our conclusion is that it is possible with error-correction codes to build up a "shell" of states around the centers of row sub-sets, and that this process will protect the circuit against secondary device malfunction – when the ultimate destinations of transitions have been reached. During the transition itself, however, – especially when the secondary state is one on the boundary between the row-set of origin and the destination row-set – momentary false circuit operation may occur.

Consider now Fig. 19 which is a much simplified and slightly modified representation of the escapement circuit of Fig. 9. Recall that the input, p, is fed into one (say the α-) storage while the output q is taken from the other (the β-) storage. Recall also that it was assumed that the time between changes of the clock signal, y_o, was greater than the response time of any device in either the α- or the β-storage.

Assume now that additional devices are used in the two storages in an error-correcting scheme (Hamming or otherwise) so that when all devices are stable the

proper output is obtained from storage even though some device therein is responding improperly. (That this is possible follows from the discussion of the asynchronous case given above. All that is necessary is that each error-free state be surrounded by a shell of states which give the same storage bank output.) Because all devices change state in a shorter time than the time in which y_o will again change, the escapement circuit formed will operate satisfactorily even when some device malfunctions. Extension to the case of multiple errors merely necessitates the use of more elaborate error correction within the storage banks.

The difference in the conclusions about the feasibility of error correction in asynchronous and synchronous circuits may be puzzling unless it is remembered that the clock signal, y_o, in the escapement circuit corresponds to the response of the $\mathscr{Y}_o$ device in the delay circuit. (Compare Figs. 8 and 9.) When we assume that the clock signal is dependable in the synchronous circuit, this is equivalent to saying that the $\mathscr{Y}_o$ device is dependable in the corresponding asynchronous circuit. Therefore in the asynchronous delay circuit of Fig. 8 error-correction could be made to accomplish its purpose if the $\mathscr{Y}_o$ device is dependable (even if all others are not), but not necessarily otherwise.

VIII. SUMMARY

A sequential switching circuit may be thought of as a function generator that has as part of its input the circuit input, and as part of its output the circuit output. (See Fig. 4.) The remainder of the generator output is tied back through a delay or escapement circuit to be used as part of the input of the function generator. The data circulated around the feedback path pertain to the accumulation effect of all past circuit inputs. These data can be retained within the circuit because the feedback connection establishes the possibility that the over-all circuit may have more than one internal state, and therefore a memory. The essential problem in the synthesis of a sequential switching circuit is the establishment and accurate control of this memory. In the synthesis method advanced by the author this problem is restated as the problem of assigning states available from several secondary switching elements to the rows of the flow table that describes the circuit to be synthesized.

The primary theme of this report is that the intermeshed row-set concept gives a unity to various solutions of the secondary assignment problem. The problem itself would not even exist if sets of memory elements had associated state diagrams in which each node was at unit distance from every other node, as in Fig. 20(a). Since we are dealing with two-valued elements, no such simple hypothetical state diagram corresponds to the actual sets of elements available. Instead, sets of binary elements have state diagrams similar to that of Fig. 20(b). The construction of intermeshed row-sets is equivalent to tying together selected nodes of the binary state diagram to give sets of states that have the desired tangency properties, as shown in Fig. 20(c) and previously

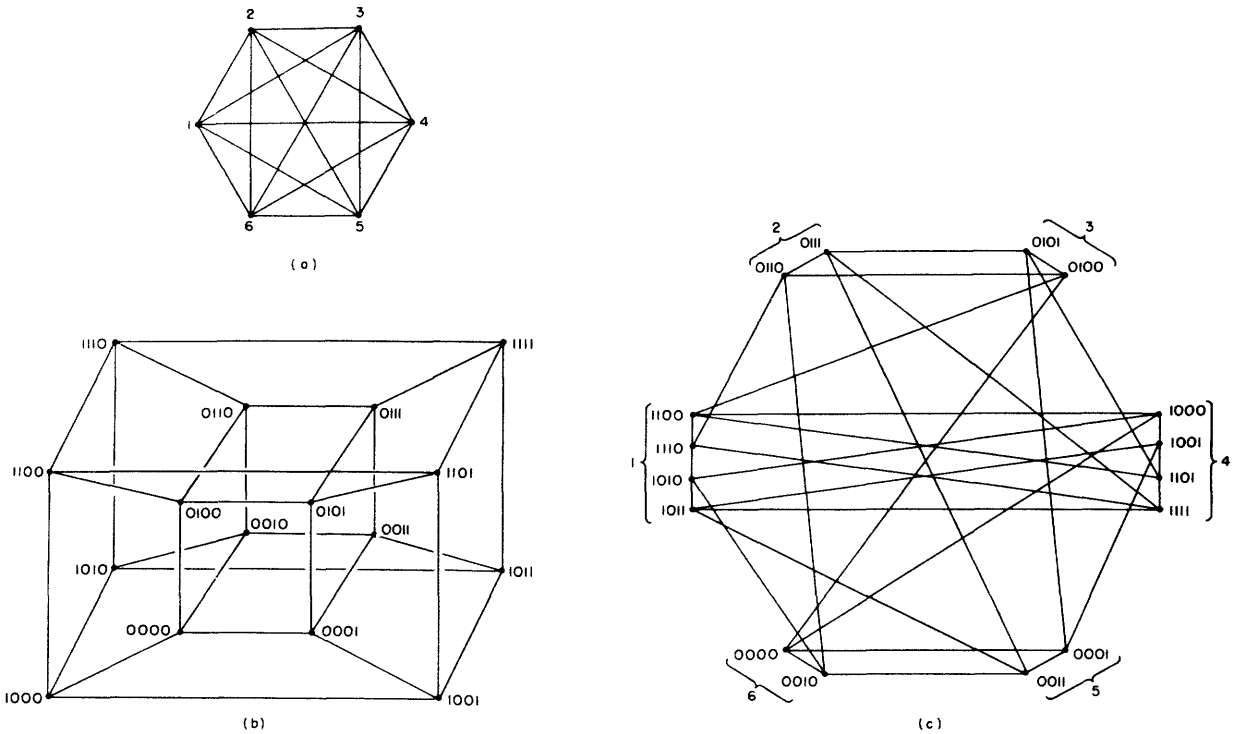


Fig. 20
Evolution of an intermeshed row-set state diagram.

in Figs. 6(b) and 6(c). (The illustration given in Fig. 20(c) is obtained from the $2S_o - 2$ secondary assignment for $S_o = 2$.)

Several secondary assignments have been discussed. Perhaps the $2S_o + 1$ solution is the most important practically, since it is directly related to synchronous circuits. The derivation of the $2S_o + 1$ delay circuit and the related escapement circuit emphasized that the clock signal of the latter corresponds closely to the response of the $\mathscr{Y}_o$ device of the former. Extrapolation of the α- and β-storage idea should furnish a useful way of thinking about what is necessary in designing delay and escapement circuits built from much larger data-storage circuits, such as banks of magnetic cores or electrostatic storage tubes.

The $2S_o - 1$ and $2S_o - 2$ solutions are of theoretical interest and indicate just how few secondary devices can be used to synthesize the general sequential circuit. For these, the inter-row transition time is four or six units. If it is important to minimize this time, the $2^{S_o} - 1$ solution represents the best that can be done: one unit of time for each inter-row transition.

The possibility of counteracting the effects of malfunctioning of a secondary device was studied from a general point of view. This study showed that error-correction is entirely feasible for synchronous circuits, but that it has somewhat limited practicability in asynchronous circuits.

Acknowledgment

The author wishes to acknowledge his indebtedness to Mr. Robert E. Wengert (formerly a graduate student at the Massachusetts Institute of Technology, now with the General Electric Company, Schenectady, New York) who, in an unpublished Master of Science thesis, indicated that a flow table of 2^{S_o} rows could be implemented with $2S_o - 1$ secondary devices, and whose intermeshed row-sets using $2S_o$ and $2S_o - 1$ devices (Figs. 11 and 13 of this report) are less cumbersome than those originally demonstrated by the author.

References

1. D. A. Huffman, The synthesis of sequential switching circuits, J. Franklin Inst. 257, Nos. 3 and 4 (1954); Technical Report 274, Research Laboratory of Electronics, M. I. T. (1954).

2. S. Guterman, R. D. Kodis, and S. Ruhman, Logical and control functions performed with magnetic cores, Proc. I. R. E. 43, No. 3, 291 (1955).

3. R. W. Hamming, Error detecting and correcting codes, Bell System Tech. J. 29, 147-160 (1950).