copy #3

CANONICAL FORMS FOR INFORMATION-LOSSLESS FINITE-STATE LOGICAL MACHINES

DAVID A. HUFFMAN

TECHNICAL REPORT 349

MARCH 25, 1959

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
RESEARCH LABORATORY OF ELECTRONICS
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-78108, Department of the Army Task 3-99-06-108 and Project 3-99-00-100.

Canonical Forms for Information-Lossless Finite-State Logical Machines*

David A. Huffman

Department of Electrical Engineering and Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, Massachusetts

Summary: An important class of finite-state machines transforms input sequences of digits into output sequences in a way such that, after an experiment of any finite length on the machine, its input sequence may be deduced from a knowledge of the corresponding output sequence, its initial and final states, and the set of specifications for the transformations by which the machine produces output sequences from input sequences. These machines are called "information-lossless."

Canonical circuit forms are shown into which any information-lossless machines may be synthesized. The existence of inverses for these circuits is investigated and circuits for their realization are derived.

1. Introduction

An information-lossless transducer is, roughly, one for which a knowledge of the output sequence of symbols is sufficient for the determination of the corresponding sequence of input symbols. Such transducers find application in the preparation of data for transmission through channels in which secrecy is important or in which the signals are subject to man-made or natural noise. Many different types of data preparation have been used. It is the purpose of this paper to derive a single block diagram for the representation of the most general information-lossless transformation that can be achieved by a finite-state machine. Emphasis will be placed on circuits which process streams of binary symbols, even though the results obtained are applicable to other alphabets of symbols.

2. Combinational Circuits

A combinational circuit is a finite-state circuit with only one state and therefore exhibiting no memory. Such circuits are basic components of circuits which do have memories and will be studied briefly here only to establish notation which will be useful later in the paper. Examples of related work appear in references 1 and 2.

For a combinational circuit, the input symbol (or combination of symbols) at any given moment uniquely defines the output symbol (or combination of symbols). An "information-lossless" combinational circuit is defined here as one which has the additional property that the output symbols uniquely determine the input symbols. Equivalently, a "lossless" combinational circuit is one for which no two different input combinations can produce the same output combination. The requisite one-to-one mapping is most easily seen by examining the describing truth-table. For a circuit with n inputs and n outputs the condition of losslessness implies the solvability of the equations showing how the outputs (y) depend upon the inputs (x).

As an example consider the circuit described in Fig. 1. The equations of Fig. 1-b, written in terms of the operations of the logical product and of addition mod-2, are seen to be nonlinear because they contain product terms such as x_1x_3. The lossless condition is satisfied because the rows of the right-hand side of the truth table of Fig. 1-c are merely a permutation of the rows of the left-hand side.

*This work was supported in part by the U.S. Army (Signal Corps), the U.S. Air Force (Office of Scientific Research, Air Research and Development Command), and the U.S. Navy (Office of Naval Research).

In Fig. 1-d, e are demonstrated the "inverse" table of combinations and solutions for the x's in terms of the y's.

In general, n equations in n unknowns have unique solutions if and only if the algebraic expansions of $y_1, y_2, \ldots, y_n$ and of the products of these functions taken 2, 3, ..., and n - 1 at a time do not contain $x_1x_2\ldots x_n$, but if the product function $y_1y_2\ldots y_n$ does contain $x_1x_2\ldots x_n$. The presence or absence of the term $x_1x_2x_3$ in our example corresponds to the oddness or evenness, respectively, of the number of 1-entries in the column which would describe the function in a truth-table. For instance the number of 1's in the y_1 column of Fig. 1-c is even, and thus the algebraic expression for x_1 in Fig. 1-b does not include the term $x_1x_2x_3$. On the other hand the truth-table for the product function $y_1y_2y_3$ would have a 1-entry only in the top row; therefore the algebraic expansion of $y_1y_2y_3$ would contain the term $x_1x_2x_3$.

The notation that will be used for blocks representing various varieties of logical or combinational functional dependencies is given in Fig. 2. The symbols stand for either the binary signals on a single lead or the signals on a multiplicity of leads. In Fig. 2-a the relation is to be read "x determines y," a statement true of any deterministic logical network. For a lossless logical network the additional statement "y determines x" also holds. A further representation (see Fig. 2-c) will be found useful later in this paper. In it the input signals are divided into two sets, one of which, c, is labeled "control." The interpretation for this type of block is "For any possible control signal, c, the input x may be determined from a knowledge of the output y, although the actual mapping of x into y may be a function of c."

3. Terminal Description of Sequential Circuits

A sequential or finite-state circuit[3, 4] can be represented schematically as a combinational circuit with some of its output signals reintroduced as input signals after some delay (see Fig. 3). We treat here only synchronous circuits, for which the signal delay around the feedback loops is the same for all loops and this delay corresponds to the separation between successive digits in the input and output streams of digits. The present state of a finite-state circuit is represented by the set of response signals, s, at the outputs of the feedback loops, and therefore the next state is represented by the set of excitation signals, S, at the inputs of these loops. The fundamental statement of finite-state circuit theory is contained in the relation shown in Fig. 3, which is interpreted "The next state, S, and the output, y, are determined by the present state, s, and the input, x." For a specific circuit these dependencies may be listed in a matrix form (flow table) or in a graph (state diagram) whose nodes represent states of the circuit and whose directed branches represent transitions between states. Both methods will be found useful in this paper.

The correspondence between these two forms may be illustrated by reference to the indicated entry and the indicated transition in Fig. 4, which describes a specific two-state circuit. Each of these is interpreted "When the circuit is in state s_1 and the input symbol is x = 1, the resulting output symbol is y = 0 and the next state is S_2."

4. Definition of Information Quantities

The information quantities that we shall use here are related to the knowledge that an observer of the circuit has when he has a knowledge of the describing flow table and of the sequence of output symbols but no direct knowledge of its input symbols or of its internal states. (These quantities are defined more precisely and illustrated more fully in reference 5.)

Input information is related to the output observer's expectation of a given input symbol. If, for example, the binary input symbols are equally likely and independent of each other, the input information rate is at all times one bit per symbol.

Output information is related to the output observer's expectation of a given output symbol. In the circuit of Fig. 4 this observer knows that the state s_1 can be followed only by transitions which yield the output $y = 0$. Therefore when he knows that the state of the circuit is s_1 and observes that the output is $y = 0$ the corresponding output information is zero.

Information is stored when, from the output observations only it becomes impossible to tell exactly what the state of the circuit is. If, for example, an observer calculates (as he might, if given the data in the paragraph above) that the circuit is in state s_1 or state s_2 with equal probability, then for him the circuit has stored one bit of information. Note that the quantity of information stored in this sense may be arbitrarily large, even for a circuit with only two states and a correspondingly simple realization if only the input symbols are unexpected enough to the observer.

Information is lost when change of internal state takes place in such a way that data about the past history of the circuit input is lost. Its measure is related to the probability that the actual input symbol sequence, rather than any of the other possible input sequences, was responsible for the observed output sequence. For example, if the output observer knows that the initial state of our circuit is s_1 and then sees two zeros in succession as output symbols then, for him, information is lost, even if the final state of the circuit is now revealed to be s_2, since the corresponding input sequence could have been either 0, 1 or 1, 0, and no further analysis of the output data preceding the initial state of s_1 or of the output data following the final state of s_2 will be of any avail in determining which input sequence actually occurred (see Fig. 5-a). If these sequences were equally likely one bit has been lost.

It can be proved that with these definitions of information quantities the following information conservation equation is valid for each step in an indefinitely long sequence of observations:

$$I_{input} = I_{output} + I_{lost} + \Delta I_{stored}$$

5. Definition of Information-Lossless Finite-State Circuit

It is clear from the preceding discussion that information loss occurs in a circuit when two or more input sequences map into the same output sequence, because then the input sequence cannot be uniquely determined if only the output sequence is known. More exactly, a sequential circuit is defined as lossless if and only if there exist no two (not necessarily different) states s_i and s_f and no two different equal-length input sequences $\{x\}$ and $\{x'\}$ and output sequence $\{y\}$ such that both $\{x\}$ and $\{x'\}$ can lead from s_i to s_f and yield $\{y\}$. This is, of course, equivalent to saying that a circuit is lossless if and only if, for an indefinitely long experiment in which the initial and final states and the output sequence are known, the input sequence may be determined if the state diagram description of the circuit is given.

The primary purpose of this paper is to determine how the form of the block diagram for a general finite-state circuit (Fig. 3) needs to be more explicitly specified in order to describe only finite-state circuits that are information-lossless.

6. Class I Information-Lossless Circuits

The clerical procedures for the determination of losslessness can be organized rather neatly. Consider the flow table of Fig. 6-a and the derived table of Fig. 6-b. The first row of this derived table tells us that if the initial state of the circuit is s_1 the next state may be deduced to be either S_4

or S_3 immediately upon determination of the output symbol as $y = 0$ or $y = 1$, respectively. The other rows have similar interpretations. Clearly, the example before us is a special case for which an input symbol always produces an immediate (mod-2) effect upon the output and is characterized by the fact that each of the two transitions away from any state are associated with the two different output symbols. Thus the possibility for "parallel" sequences shown in Fig. 5-b does not exist. For such circuits, which will be called Class I circuits, it is possible to derive "inverse" circuits which, when put in cascade with the original, produce as their output sequence an exact replica of the input sequence of the original. The terminal specifications for these inverse circuits are easily had by completing the table illustrated in Fig. 6-b with entries that tell what x-symbol should be associated with a given transition.

A block diagram showing one possible realization of a Class I circuit is shown in Fig. 7-a, and one possible realization of its inverse is shown in Fig. 7-b. These two circuits differ only in the connections made to the mod-2 adder gate, and therefore we may conclude that the inverse to a Class I circuit can be realized in a circuit having the same number of states as the original. Moreover, since the circuits have a reciprocal relationship, either may be used as the canonical form of a Class I circuit. Each circuit has the requisite property that the input and output digits differ (mod-2) by a fixed function of the circuit state, s.

7. Class II Information-Lossless Circuits

Another case of an information-lossless circuit is shown in Fig. 8. The upper four rows of the table in Fig. 8-b are derived in a manner similar to that used for our previous example, except that now the knowledge of an output symbol does not necessarily lead immediately to a knowledge of the input symbol that produced it. For example, the second row of the derived table is to be interpreted as follows: If the initial state of the original circuit is s_2, then the symbol $y = 0$ must necessarily follow, and as a result we are not now certain whether the following state is S_1 or S_3 (or whether the input symbol was $x = 0$ or $x = 1$).

The first four rows of the new table indicate that confusion may exist between states s_1 and s_3 or between s_1 and s_4. The two symbols s_{13} and s_{14} are entered as designators for rows which are added to the first four rows of the table. Entries for these new rows are found by adding subscripts found in the corresponding entries found in the rows specified by the subscripts of the designator of the new row. For instance, the entry in the $y = 1$ column for the row headed s_{13} is S_{134}, since the entries found in the rows headed s_1 and s_3 were S_3 and S_{14}. The newly derived entry tells us that if we are uncertain as to whether the state of the circuit is s_1 or s_3 and if an output symbol $y = 1$ is observed, our new uncertainty is among S_1, S_3, and S_4. The process of generation of new rows is repeated as long as is necessary. Ultimately, the necessity for new rows is ended, and the table is complete. If in the process of adding subscripts from "component" rows to find the subscripts for "composite" rows no situation is found in which the same subscript is found in two of the component rows, then the circuit being tested is information-lossless. Our present example is a circuit of this type.

It could have been seen directly from Fig. 8-a that the flow table described a lossless circuit, since two and only two transitions lead to each state and each of these transitions is associated with a different output symbol. We shall call such a circuit a Class II circuit. Thus there is no possibility for "parallel" sequences shown in Fig. 5-b. Further, a knowledge of the final state of the circuit and the last output symbol is enough for the determination of the next-to-final circuit state. Thus the input sequence for a finite experiment on a Class II circuit may be determined from a knowledge of the <u>final</u> state and the output sequence, just as the input sequence for a finite experiment on

a Class I circuit may be determined from a knowledge of the initial state and the output sequence.

Since, for a Class II circuit, a knowledge of a state and the output symbol for the transition leading to that state is sufficient for the determination of the preceding state and this input symbol, this is equivalent to saying that the combinational logic of the general block diagram of Fig. 3 is, for Class II circuits, lossless. (See Fig. 9.)

8. General Information-Lossless Circuits

It seems to the author that both Class I and Class II circuits deserve to be called information-lossless; the first, because an inverse circuit can always be specified, and the second, both because a specific decoding procedure can be described once the final state of an experiment is given, and because it is conceptually satisfying that a lossless combinational circuit in which some outputs are reintroduced as inputs after a unit delay is also lossless in the wider sense that we have used in this paper to apply to sequential circuits. It is only fair to point out to the reader that some other, more restricted, definitions of terms similar to information-losslessness as used in this paper have been used, and probably will continue to be used, by others.

There are many circuits which are lossless which are neither purely Class I nor purely Class II circuits. For all of these circuits the test illustrated in Fig. 8-b is valid, but the circuit cannot be synthesized in either of the canonical forms already given. Instead, the more general canonical form shown in Fig. 10 may be shown to be appropriate for any information-lossless circuit. The canonical forms of Fig. 7-a and Fig. 9 can be seen to be special cases of the general form given in Fig. 10.

In order to show that every lossless circuit can be put into the form of Fig. 10 a synthesis procedure will be illustrated. Our working example will start with the flow table of Fig. 11-a. The test table showing losslessness is developed as in the previous example.

The first step in the synthesis procedure involves the assignment of "a-symbols" to some of the sets of states listed as row-headings in the test table. The assignment is made in such a way that each circuit state is associated with at least one a-symbol, and that each set of states given its own symbol leads in the test table to other sets also given their own symbols. Obviously, this procedure could always be followed because we could (even though it would be uneconomical of a-symbols) assign a distinct symbol to each row of the test table. A result of the assignment is that a matrix can be obtained (see Fig. 11-c) which shows how the next symbol, A, depends upon the output, y, and the present symbol, a. Thus

$$y, a \Longrightarrow A \tag{1}$$

Now (see Fig. 11-d) assign to each state having a common a-symbol a distinct "b-symbol" so that each pair of symbols a, b defines one of the circuit states. There will be as many b-symbols necessary as there are members of the largest a-set. (If the a-sets are of different sizes, the empty entries of the matrix can be filled with a "dummy-state," s_o, having the property that $x, s_o \Longrightarrow y = x, S_o$.) Thus

$$a, b \Longrightarrow s \tag{2}$$

$$\text{and } A, B \Longrightarrow S \tag{2'}$$

so that pairs of the present (or next) symbols a, b (or A, B) determine uniquely a present (or next) circuit state, s (or S).

Because of the method of constructing the test table (and our proven assumption that we are dealing with a lossless circuit) it follows that no two member states, s_i and s_j, of a fixed a-set, a_h, can each

lead in a one-step transition to a common state, S_k, and yield a common output symbol, y. The prohibited situation is shown in Fig. 11-e. It follows that, for all physically obtainable combinations of y, a, A, and B,

$$y, a, A, B \Longrightarrow x, b \tag{3}$$

This condition may be replaced by the weaker-appearing condition

$$y, a, B \Longrightarrow x, b \tag{3'}$$

since, from

$$y, a \Longrightarrow A \tag{1}$$

and

$$y, a, B \Longrightarrow x, b \tag{3'}$$

we may derive

$$y, a, B \Longrightarrow y, a, A, B \Longrightarrow x, b$$

The dependence of x, b upon y, a, and B for our example is given in the matrix of Fig. 11-f, which is derived from the data in Figs. 11-a, c, and d.

Our method of assigning symbols guarantees (see relation 2' and Fig. 11-d) that a pair of symbols A, B determines uniquely a next-state, S. Even though S does not uniquely determine a symbol pair A, B, it is clear that at least one pair A, B is determined by each S. If, however, in addition to fixing S we also fix the A-symbol at one of those values which is possible, then it is apparent from our method of assigning symbols that a unique B-symbol is then determined. Thus

$$S, A \Longrightarrow B \tag{4}$$

The general relation

$$x, s \Longrightarrow y, S \tag{5}$$

is true for any finite-state circuit. By successive application of relations 2, 5, 1, and 4, we obtain

$$x, a, b \Longrightarrow x, s, a, b \Longrightarrow x, s, a, b, y, S \Longrightarrow x, s, a, b, y, S, A \Longrightarrow x, s, a, b, y, S, A, B$$

from which we extract, from the first and last terms, that

$$x, a, b \Longrightarrow y, B \tag{6}$$

A matrix derived from Fig. 11-f to show the dependence of y and B upon x, a, and b for our example is given in Fig. 11-g.

The synthesis is now complete, and the embodiment of the derived general relations

$$x, a, b \Longrightarrow y, B \tag{6}$$

$$y, a, B \Longrightarrow x, b \tag{3'}$$

and

$$y, a \Longrightarrow A \tag{1}$$

in a block diagram has been given in Fig. 10. The data from Fig. 11-c and Fig. 11-d may be combined as shown in Fig. 11-h, and the resulting matrix checked for its equivalence to the original specifications for our example in Fig. 11-a.

9. Inverses for General Information-Lossless Circuits

The preceding discussion has indicated (by means of a rather involved synthesis procedure) that every information-lossless circuit can be put into the form of Fig. 10. In addition, any two subcircuits that are in communication with each other in the manner shown in that diagram form an over-all circuit that is lossless.

This is most easily seen by picturing the manner in which the various signals comprising the input, output, and state signals influence each other at the successive steps of an experiment. In Fig. 12-a we have drawn three "stroboscopic views" of the general circuit with subscripts on the signals indicating the successive time intervals of our illustrative three-step experiment. The flow of the a- and b-signals from left to right in this diagram corresponds to the actual flow of these signals around the feedback loops of Fig. 10. The signals on the leads marked with dots are sufficient to determine those on all other leads. Specifically, as in any deterministic finite-state circuit, the initial state (in this case, the signals a_1 and b_1) and the sequence of inputs determine all later states and the circuit output sequence.

If we think of this figure as actually representing an iterative circuit (in which left and right have positional significance) rather than a sequential circuit (in which left and right represent earlier and later moments of time) then we may imagine building an "inverse" circuit as in Fig. 12-b. This is possible because the special nature of the subcircuits labeled "lossless" guarantees that it would be physically possible to design a network with the direction of signal flow reversed on all but the "control" leads. The knowledge of the signals on all leads marked with dots could lead to a knowledge of the signals on all other leads, including the x-signals which it is the job of the "inverse" circuit to reproduce. Specifically, we have represented in this diagram the fact that we may deduce the x-sequence from a knowledge of the y-sequence, a knowledge of the initial value (a_1) of the state of one of the subcircuits, and a knowledge of the final value (B_3) of the state of the other subcircuit.

The iterative network with bilateral flow of information (reference 6) given in Fig. 12-b is not, in general, physically realizable as an analogous finite-state circuit because this would require information flow from the "future" (right) as well as from the "past" (left). Nevertheless, this diagram indicates that a decoding procedure does exist and hence proves that any circuit in the form of Fig. 10 is information-lossless.

We conclude this section with a few more comments about the circuit of Fig. 10. Even though the synthesis procedure has been illustrated for a single binary input (x) channel and a single output (y) channel, it is clear that nothing in the procedure or proofs would keep us from applying the same techniques even if x and y represent signals on a multiplicity of leads. Nor does the number of input and output leads need to be the same. The logical network labeled "lossless" has an appropriate interpretation even if the number of y-leads exceeds the number of x-leads, for then the lossless network corresponds to a system of a certain number of logical equations in a smaller number of unknowns. As long as the equations have a unique solution the associated network is still lossless. This possibility allows the circuit of Fig. 10 to represent the most general method of introducing redundancy into the output of a finite-state machine in such a way that the transformation remains information-lossless.

Some special cases of the canonical form of Fig. 10 may be noted. If the y-signals do not actually influence the right-hand subcircuit and if the number of feedback loops in the left-hand subcircuit is reduced to zero (the lossless network then could be a mod-2 adding device) then the diagram would represent the adding of a machine-generated pseudo-random signal to the input signal to form the output signal. On the other hand, the left subcircuit could represent a simple permutation device which

connects its inputs (x- and b-signals) to its outputs (y- and B-signals) in various orders which are functions of the state of the right subcircuit. A large number of other special cases may be deduced by the reader. The preceding work has shown that these, and other lossless circuits – no matter what the form of the block diagram from which they were designed – can always be synthesized in the canonical form of Fig. 10.

10. Test for Lossless Circuits of Finite "Order"

We have shown that the most general lossless finite-state circuit has no "inverse" which will regenerate the input sequence of the original circuit; some of the data necessary for this regeneration consists of the final state (B_n) of one of the subcircuits at the end of the n-step experiment. Since n may be indefinitely large it is clear that we can place no bound on the number of time intervals between the occurrence of an input (x) signal of the "coding" circuit and its regeneration by a "decoding" circuit. On the other hand, one of the properties of a Class I coding circuit is that its input can be regenerated by an inverse decoding circuit after no delay whatsoever.

Many applications of information-lossless finite-state circuits make it desirable to be able to retrieve the coding circuit input sequence after, at most, some delay, N, which is fixed but which is greater than that (zero delay) given by a Class I circuit. The state diagram restriction which must hold for such circuits is illustrated in Fig. 13. In this diagram we illustrate the restriction that must hold for a decoding delay which will not exceed N = 3 time intervals, but its extension to other delays is obvious. The state represented at the base of the "tree" of transitions is any one of the states in the state diagram to which the test is to be applied. But the test described below must apply to each of the states of the circuit in order for its x-sequence to be retrievable within N = 3 time intervals. We shall call a circuit that meets this test an "N^{th}-order" information-lossless circuit. It is apparent that the class of all N^{th}-order lossless circuits includes the $N\text{-}1^{st}$, $N\text{-}2^{nd}$, ..., and zeroth-order classes.

The diagram of Fig. 13 represents the tree of possibilities which exist, starting at some reference time that we refer to as t = 0. The $2^{N+1} = 16$ nodes at the right side of the diagram represent the sixteen different possible states that could exist at the end of the N^{th} time interval. The heavily-marked chain of transitions indicates the sequence of events if the input sequence is actually $x_o = 1$, $x_1 = 0$, $x_2 = 0$, $x_3 = 1$. The upper subtree corresponds to $x_o = 0$, and the lower subtree to $x_o = 1$. The digits marked on the directed branches illustrate a possible set of output symbol possibilities associated with the various transitions. The test for third-order decodability is, in this case, met because each possible sequence y_o, y_1, y_2, y_3 is associated uniquely either with $x_o = 0$ or with $x_1 = 1$. In other words, no output sequence of the upper subtree can be found in the lower subtree, and vice versa. The meeting of the test guarantees that if an initial state (at t = 0) is known a further knowledge of y_o, y_1, y_2, and y_3 is sufficient to determine x_o, and hence the next state (at t = 1) may, in turn, be deduced. Clearly, if the illustrated test is met for each state of a state diagram, the input symbols may be iteratively deduced from the output symbol stream with a delay of no more than N = 3 time intervals.

The block diagram that we shall give for an N^{th}-order lossless circuit depends heavily upon the definition of binary symbols with the notation K_t^i (for $i = 1, 2, \ldots, N$) which indicate the effect of the actual output symbol y_t occurring at time t upon the possibility of the determination of the input symbol x_{t-N+i} which occurred at time $t - N + i$ if the preceding input sequence were known. The symbol K_t^i equals 1 or 0, according to whether the symbol y_t would or would not lead to the determination of x_{t-N+i}.

In our example (see Fig. 13) for N = 3, $K_o^3 = 0$, since the transition which actually occurred (for $x_o = 1$) and the transition for the other value of x_o would both have given $y_o = 0$, and hence a knowledge of y_o would not lead to a determination of the actual value of x_o. On the other hand $K_2^3 = 1$ because the transition that actually occurred (for $x_2 = 0$, with the values of x_o and x_1 presumed to be known) and the transition for the other value of x_2 would have given opposite values of y_2, and therefore a knowledge of y_2 would lead to a determination of the actual value of x_2.

In the evaluation of $K_2^1 = 0$, for instance, we must examine a larger set of possible values of output symbols. This value of K_2^1 has been determined to be zero because the value of $y_t = y_2$ which actually occurred ($y_2 = 1$) could also have occurred (as at least one of the values of y_2 shown in the upper subtree corresponding to $x_o = 0$ and $t = 2$) if the value of $x_{t-N+i} = x_{2-3+1} = x_o$ had been opposite to the actual value of $x_o = 1$; therefore a knowledge of the actual value of y_2 would not in itself let us deduce the value of x_o, and hence we have evaluated $K_2^1 = 0$.

Note now that the logical sum of K_o^3, K_1^2, and K_2^1 (all of them equal to zero in our example) indicates by the fact that it is zero that the sequence $y_o = 0$, $y_1 = 0$, $y_2 = 1$ does not correspond uniquely to the actual value of x_o ($x_o = 1$) but could also have occurred if x_o had had the other possible value. If one or more of the values K_o^3, K_1^2, or K_2^1 had had the value one, their logical sum would also have been one and this would have indicated that we could deduce x_o from a knowledge of y_o, y_1, and y_2. In general, let us define K_t^o as the logical sum of K_{t-1}^1, K_{t-2}^2, ..., and K_{t-N}^N. Thus K_t^o indicates by its value (1 or 0) whether or not the value of x_{t-N} could be determined from the sequence of values y_{t-N}, y_{t-N+1}, ..., and y_{t-1}.

The main reason for defining the K_t^i-symbols and for deriving from them the K_t^o symbol will now be explained. In our example (Fig. 13) for the indicated set of transitions we have derived that $K_3^o = 0$, since the sequence $y_o = 0$, $y_1 = 0$, $y_2 = 1$ could have occurred for either $x_o = 0$ or $x_o = 1$. If we wish to insure that the test for decodability with delay N = 3 will be met we must insist that the checked transitions (for t = 3) for the subtree associated with $x_o = 1$ will carry output symbols opposite to those associated with the checked transitions in the subtree for $x_o = 0$. In other words, if $K_3^o = 0$ we must arrange in the block diagram we are deriving that y_3 shall be influenced (in a mod-2 fashion) only by the input symbol x_o and not by any later x-symbol. If K_3^o had been equal to unity no such restriction on the mode of operation of the derived circuit would have to be imposed.

11. Canonical Form for N^{th}-order Lossless Circuits

The embodiment of the preceding ideas in a canonical form (see Fig. 14) appears to be rather complicated in spite of the fact that the operation of the circuit is conceptually simple. The past N input digits are stored in the "input section." The signals shown correspond to t = 3, and subscripts indicating this have been given for most of the signals in the diagram. The "output section" (it corresponds to the right-hand subcircuit of Fig. 10) is given that name because it is driven by the circuit output and therefore its state is a function of the past outputs of the circuit. This subcircuit may have a multiplicity of feedback loops even though only one has been drawn. By proper design of its logical network and of the logical network at the top of the block diagram the C-signals (there would, in general, be 2^N such signals) may be chosen to be any arbitrary functions of the actual past output sequence from the circuit. The stored x-signals (perhaps modified by the K^o signal) select one of the C-signals via the "transfer section" as the signal F^{ooo}. The transfers which are indicated allow a signal to flow downward if the associated controlling signal is zero, but side-track it to another lead if that controlling signal is unity. The other signals at the bottom of the transfer section correspond to what the signal F^{ooo} could have been if the x-signals had been other than their actual values, and if the

preceding output sequence had remained fixed.

The $2^N - 1$ G-signals represent differences (mod-2 sums) between the actual F^{ooo} signal and what that signal would have been if other values of x_1, x_2, and x_3 had existed. If a comparison between F^{ooo} and those other signals indicates that they could have been the same, then the appropriate K_t^i-signal will have the value zero, thus indicating that determination of the input x_{t-N+i} would be impossible from a knowledge of y_t alone.

In the K-section these various signals are brought together with the proper timing and logically added so that the resulting value of K_t^o will be zero if and only if the circuit is to be operated in that mode in which the N^{th} preceding input (x_o in our example) alone of the stored inputs influences the output (in a mod-2 fashion).

12. Inverses for N^{th}-order Lossless Circuits

It is possible to build an "inverse" circuit based on the block diagram of Fig. 14 which will regenerate the x-sequence from the y-sequence with a net delay of N time units when the original coding circuit is followed in cascade by the decoding circuit. The C-, D-, and E-signals will be especially important in the development of the inverse circuit. We shall show this circuit (Fig. 15) broken down into several component sections for ease of explanation of its operation. The signals which are shown on the various leads correspond to the signals which would exist at t = 3. Then the input to the inverse circuit is the signal y_3 and its output is the third (in general the N^{th}) previous input, x_o, to the coding circuit which the decoding circuit is just developing. This x_o value is the input to a subcircuit (see Fig. 15-a) which stores the three values of x that precede x_o. We shall also store the three y-signals that precede y_3 (see Fig. 15-b). These y-signals are used to drive four carbon copies of the two subcircuits which develop the eight C-signals of Fig. 14. The result is that we have available the C-signals that are present not only at the present time (t=3) in the coding circuit but also replicas of these signals for the three preceding time intervals.

In order to derive x_o we need to have available the three signals K_o^3, K_1^2, and K_2^1 and also the signal which is their logical sum, K_3^o (see Fig. 15-c). Each of the first three signals will be developed as shown below. The restrictions imposed by the canonical circuit of Fig. 14 guarantee that either $K_o^3 = 1$ or $K_1^2 = 1$ or $K_2^1 = 1$ (more than one of these statements may be true), or that $K_3^o = 0$. Subcircuits will be shown which consider each one of these four possibilities and each of these subcircuits will indicate, if enough information is available, that $x_o = 1$. Absence of an output from a subcircuit will indicate either that $x_o = 0$ or that not enough information is available within that circuit for a determination of x_o. Either all of these circuits give an output of zero, indicating that $x_o = 0$, or at least one of the circuits gives an output of one, in which case we conclude that $x_o = 1$. The four subcircuit output signals are logically added as shown in Fig. 15-d to give x_o.

If at t = 0 we had compared the signal $F_o^{ooo} = x_{-3} \oplus y_o$ with the E-signals we could have deduced that if F_o^{ooo} had a value opposite to E_o^{ool} then $x_o = 0$, but if F_o^{ooo} had a value opposite to E_o^{ooo} then $x_o = 1$. If either of these statements was true then we could deduce that $K_o^3 = 1$. A circuit corresponding to these statements is given in Fig. 15-e. It consists primarily of a part of the transfer section of Fig. 14.

If at t = 1 we had compared the signal $F_1^{ooo} = x_{-2} \oplus y_1$ with the D-signals we could have deduced that if F_1^{ooo} had a value opposite to D_1^{olo} and to D_1^{oll} then $x_o = 0$, but if F_1^{ooo} had a value opposite to D_1^{ooo} and to D_1^{ool} then $x_o = 1$. If either of these statements was true then we could deduce that

$K_1^2 = 1$. The corresponding circuit is shown in Fig. 15-f. At t = 2 a similar comparison between F_2^{000} and the C-signals leads to Fig. 15-g.

The final subcircuit is active only if each of the three previous subcircuits has failed to reach a conclusion about the value of x_o. In that case, $K_o^3 = K_1^2 = K_2^1 = 0$ and $K_3^o = 0$ follows. We may conclude from Fig. 14 that when $K_3^o = 0$ then $y_3 = C_3^{000} \oplus x_o$, and therefore $x_o = C_3^{000} \oplus y_3$ (see Fig. 15-h).

Among other things the reader may wonder why the K^o signal of Fig. 14 was not required to modify the x-signals which control the various transfer sections of Fig. 15. Briefly, at any time that K^o is zero in Fig. 14, then $F^{000} = E^{000} = D^{000} = C^{000}$ and the subcircuits of Figs. 15-e, f, g are automatically kept from improperly deducing that $x_o = 1$.

The work presented in this section has led us to the conclusion that the restrictions on a finite-state circuit necessary for making it information-lossless and of N^{th} order may be shown in a block-diagram canonical form. Moreover, the decoding procedure necessary for recreating the input to the coding circuit exists and also has a circuit representation.

Acknowledgment

The author would like to express his indebtedness to his many colleagues at the Massachusetts Institute of Technology for their criticism and suggestions, without which this work would not have been complete. Special thanks are due Prof. Dean Arden of the Department of Electrical Engineering, M.I.T., who, because of car-pool arrangements, often found himself a captive audience of one.

References

1. D. A. Huffman, "Solvability criterion for simultaneous logical equations," Quarterly Progress Report, Research Laboratory of Electronics, M.I.T., January 15, 1958, pp. 87-88.
2. N. Rouche, "Some properties of Boolean functions," Trans. IRE, vol. EC-7, no. 4, pp. 291-298 (December, 1958).
3. D. A. Huffman, "The synthesis of sequential switching circuits," J. Franklin Inst. 257, 161-190 (March, 1954); 275-303 (April, 1954).
4. E. F. Moore, "Gedanken-experiments on sequential machines," Automata Studies, Annals of Mathematics Studies No. 34 (Princeton University Press, Princeton, N.J., 1956), pp. 129-153.
5. D. A. Huffman, "Information conservation and sequence transducers," Proc. Symposium on Information Networks, Polytechnic Institute of Brooklyn, April 12-14, 1954, pp. 291-307.
6. F. C. Hennie, "Analysis of bilateral iterative networks," Trans. IRE, vol. CT-6 (March, 1959).

(a)

$$\begin{cases} y_1 = 1 \oplus x_1 \oplus x_3 \oplus x_1 x_2 \oplus x_1 x_3 \\ y_2 = 1 \oplus x_1 \oplus x_2 \oplus x_3 \\ y_3 = 1 \oplus x_1 \oplus x_2 \oplus x_1 x_2 \oplus x_2 x_3 \end{cases}$$

(b)

x_1	x_2	x_3	y_1	y_2	y_3
0	0	0	1	1	1
0	0	1	0	0	1
0	1	0	1	0	0
0	1	1	0	1	1
1	0	0	0	0	0
1	0	1	0	1	0
1	1	0	1	1	0
1	1	1	1	0	1

(c)

y_1	y_2	y_3	x_1	x_2	x_3
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	1	0	1
0	1	1	0	1	1
1	0	0	0	1	0
1	0	1	1	1	1
1	1	0	1	1	0
1	1	1	0	0	0

(d)

$$\begin{cases} x_1 = 1 \oplus y_1 \oplus y_3 \oplus y_1 y_2 \\ x_2 = y_1 \oplus y_2 y_3 \\ x_3 = y_2 \oplus y_3 \oplus y_1 y_2 \oplus y_2 y_3 \end{cases}$$

(e)

Fig. 1. Example of a lossless combinational circuit.

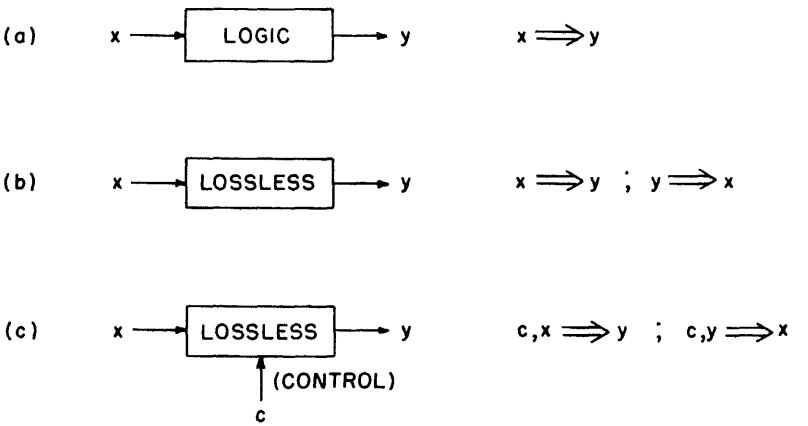


Fig. 2. Notation used for various types of combinational networks.

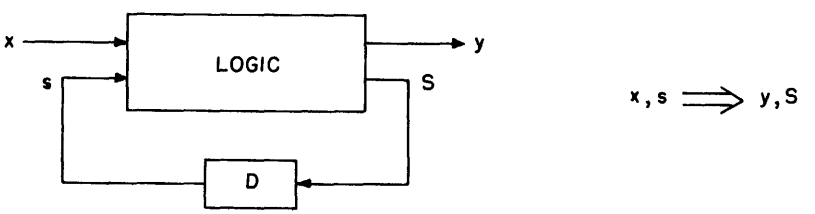
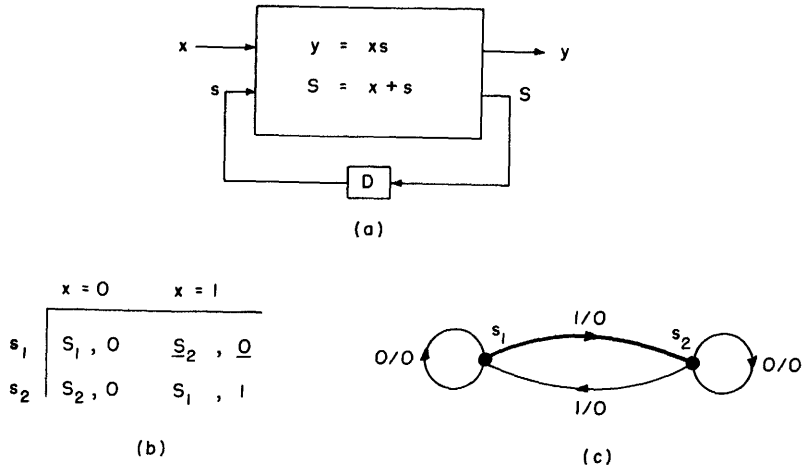


Fig. 3. General form of a finite-state circuit.



Fir. 4. A sequential circuit description and realization.

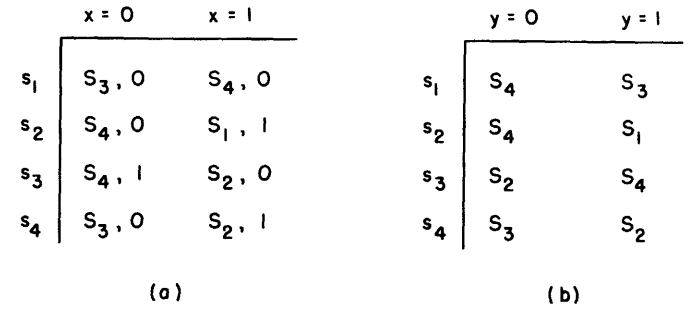


Fig. 6. Tabular test for losslessness applied to a Class I circuit: (a) flow table and (b) test table.

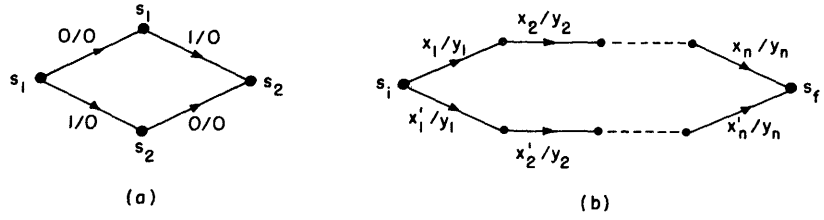


Fig. 5. Illustration of conditions for information loss.

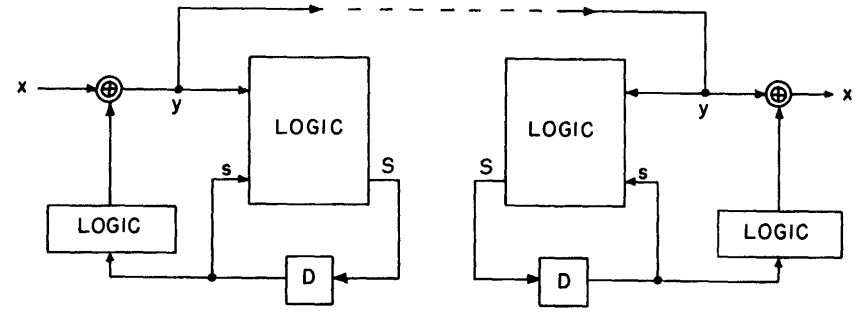


Fig. 7. Canonical forms for a Class I circuit and its inverse.

	x = 0	x = 1
s_1	S_2, 0	S_3, 1
s_2	S_1, 0	S_3, 0
s_3	S_4, 1	S_1, 1
s_4	S_2, 1	S_4, 0

(a)

	y = 0	y = 1
s_1	S_2	S_3
s_2	S_{13}	—
s_3	—	S_{14}
s_4	S_4	S_2
s_{13}	S_2	S_{134}
s_{14}	S_{24}	S_{23}
s_{23}	S_{13}	S_{14}
s_{24}	S_{134}	S_2
s_{134}	S_{24}	S_{1234}
s_{1234}	S_{1234}	S_{1234}

(b)

Fig. 8. Tabular test for losslessness applied to a Class II circuit: (a) flow table and (b) test table.

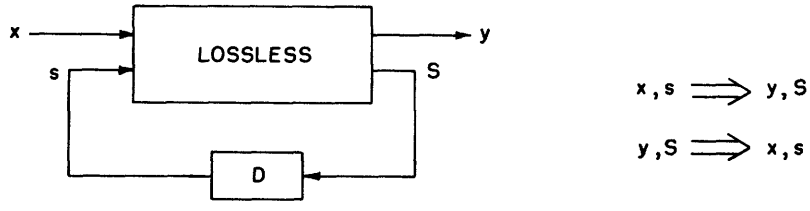


Fig. 9. Canonical form for a Class II circuit.

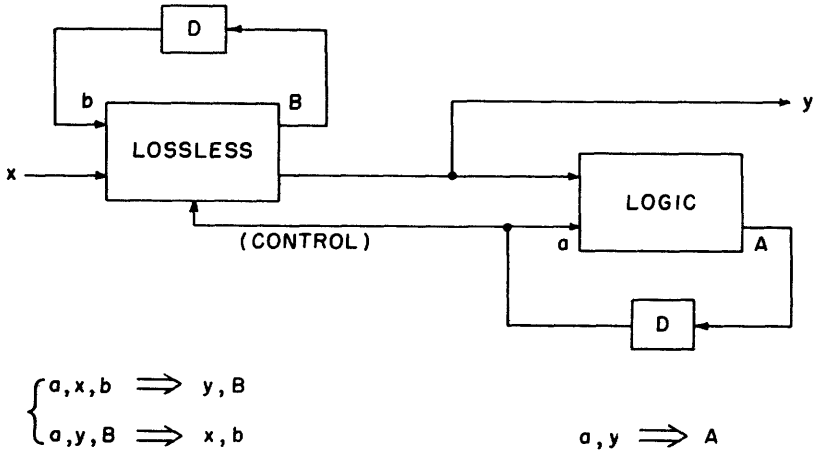


Fig. 10. General canonical form into which every information-lossless finite-state circuit may be synthesized.

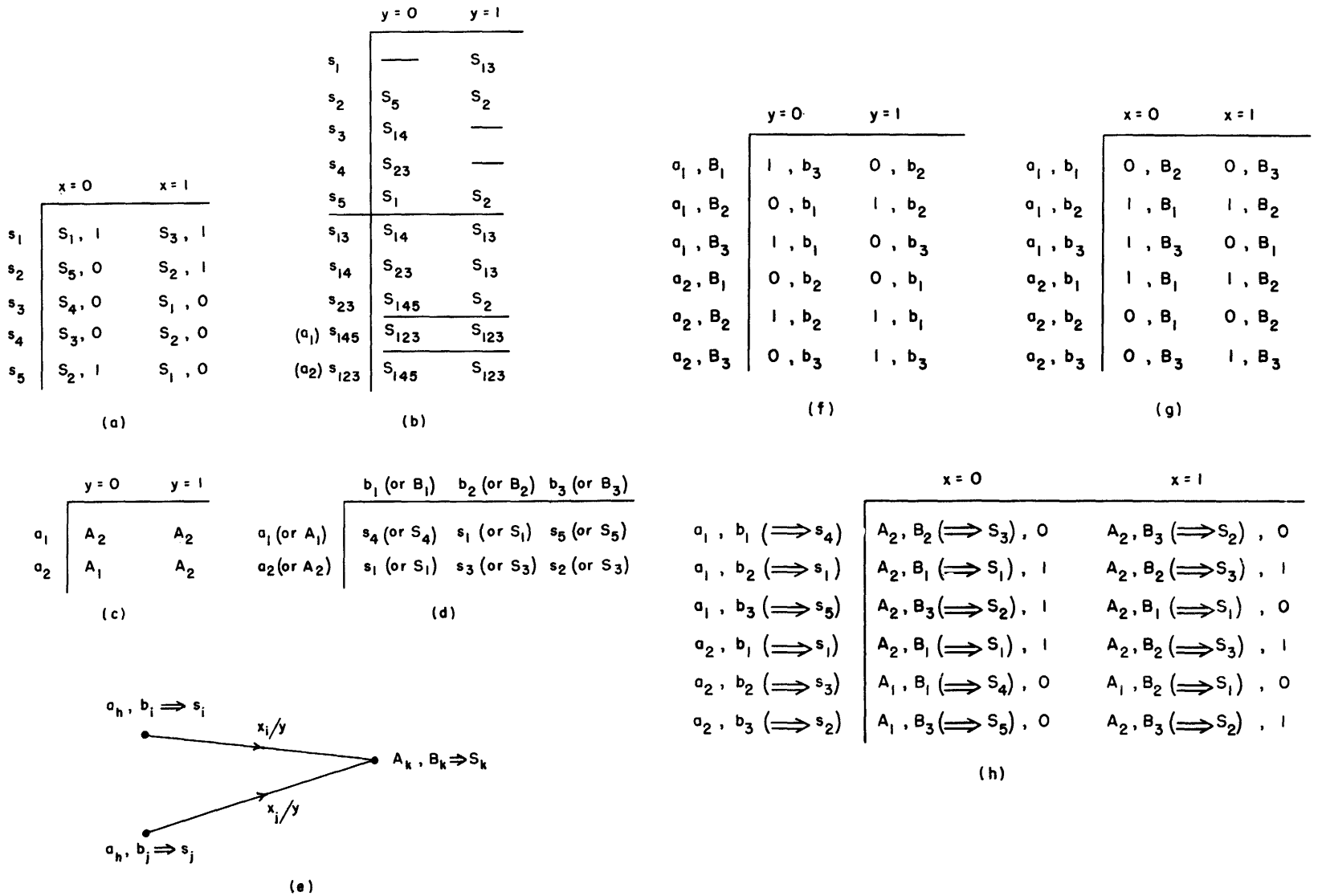


Fig. 11. Example of the synthesis procedure leading to the canonical form of Fig. 10: (a) flow table; (b) test table; (c) matrix showing y, a ⇒ A; (d) matrix showing a, b ⇒ s (or A, B ⇒ S); (e) prohibited situation; (f) matrix showing y, a, B ⇒ x, b; (g) matrix showing x, a, b ⇒ y, B; and (h) matrix showing x, a, b ⇒ A, B, y.

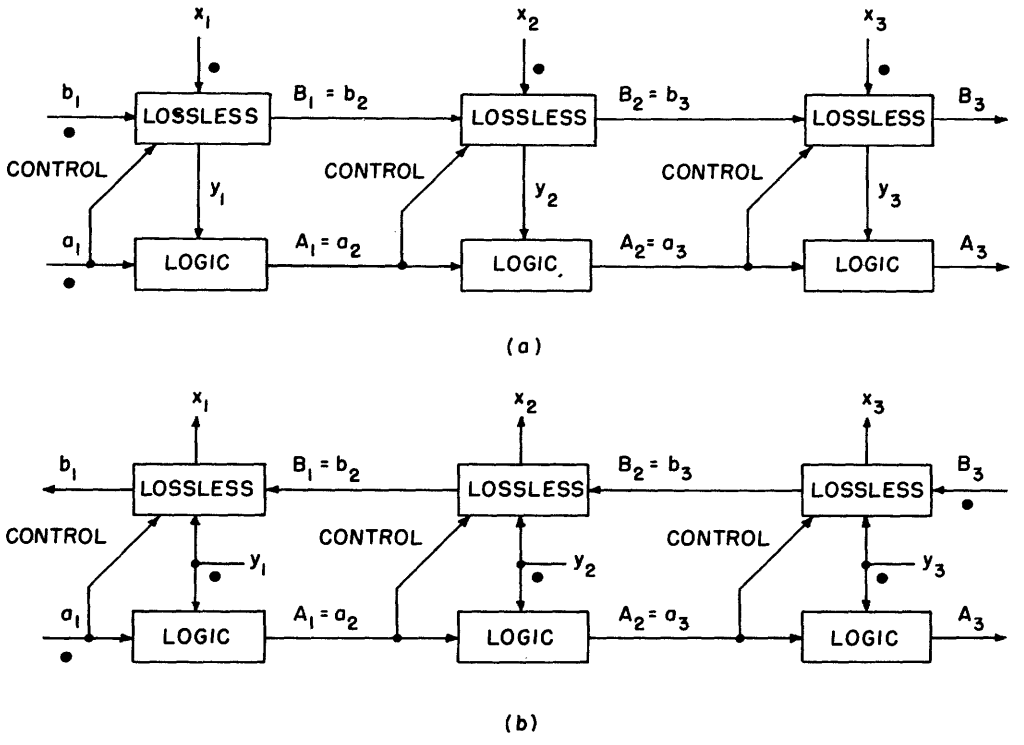


Fig. 12. An information-lossless finite-state circuit and its "inverse" developed as iterative circuits.

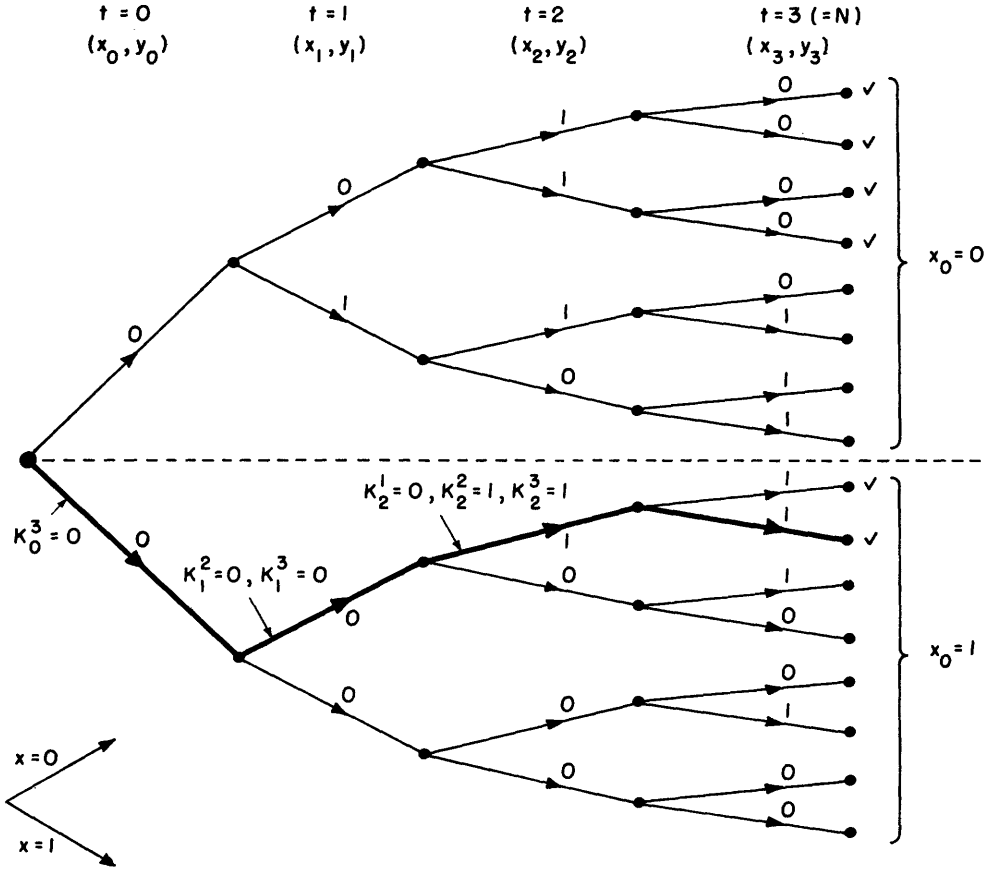


Fig. 13. Illustrating the state-diagram test for a third-order information-lossless circuit.

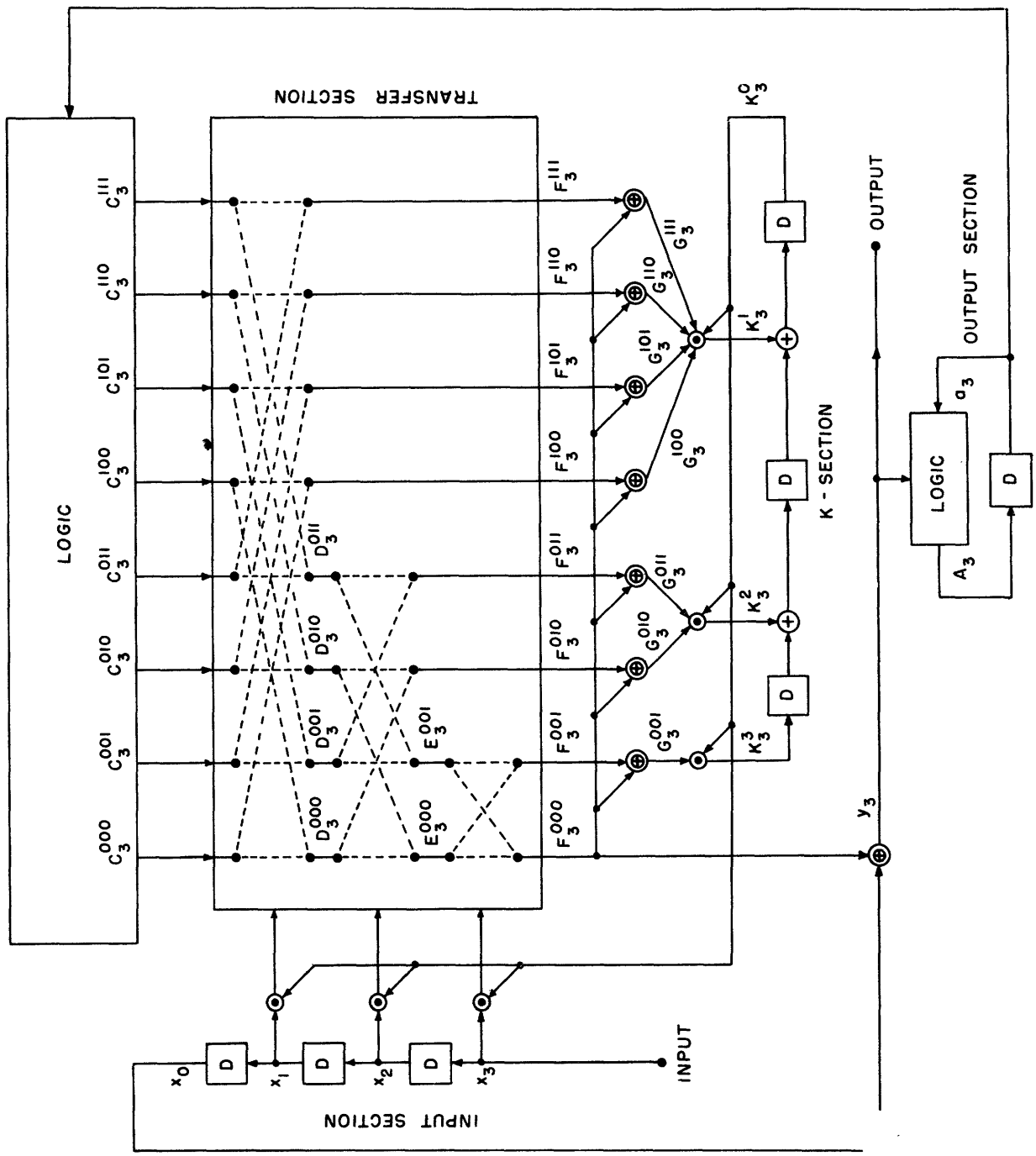


Fig. 14. Block diagram for a third-order (N=3) information-lossless finite-state circuit.

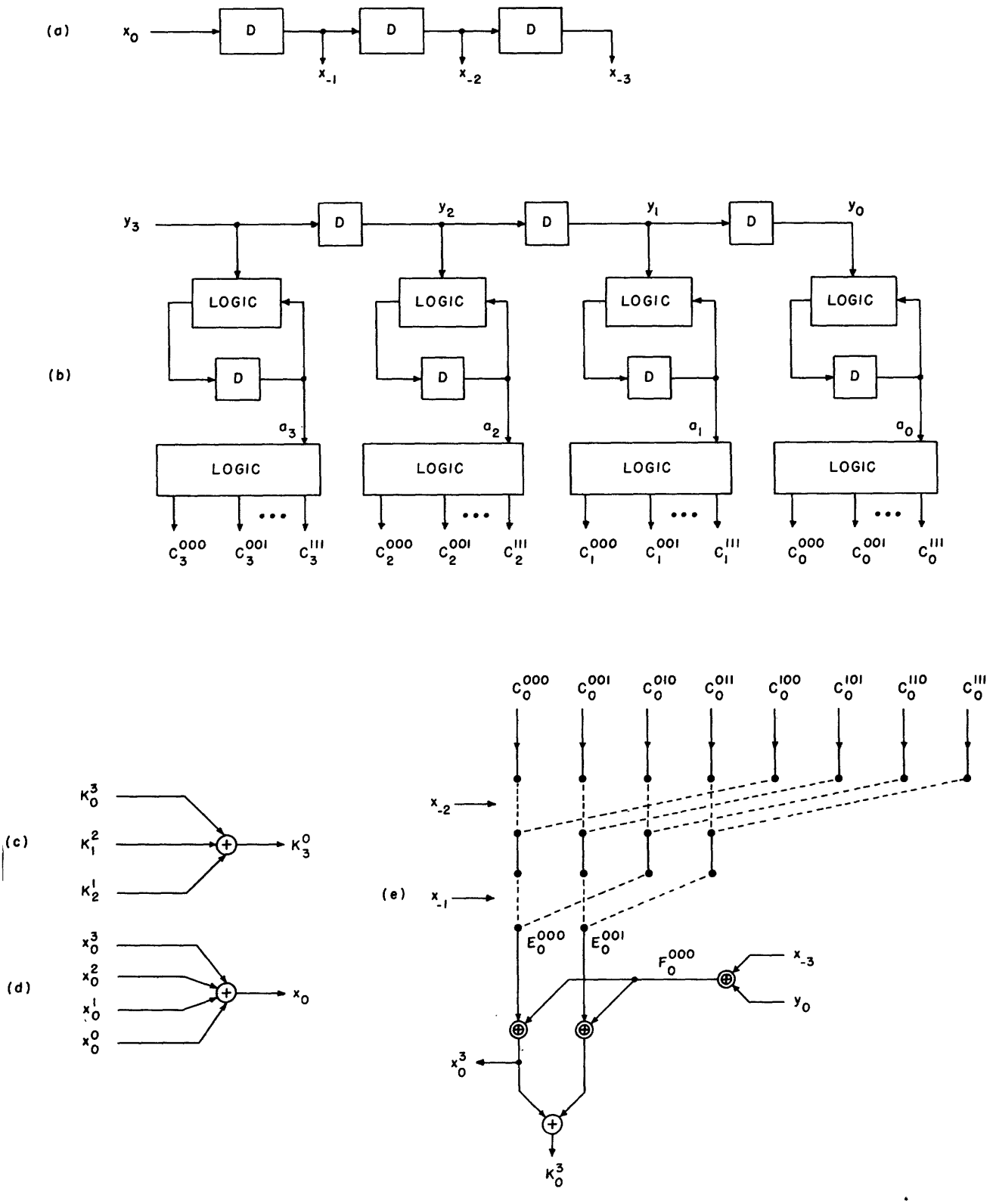


Fig. 15. The components of a circuit inverse to a third-order lossless circuit.

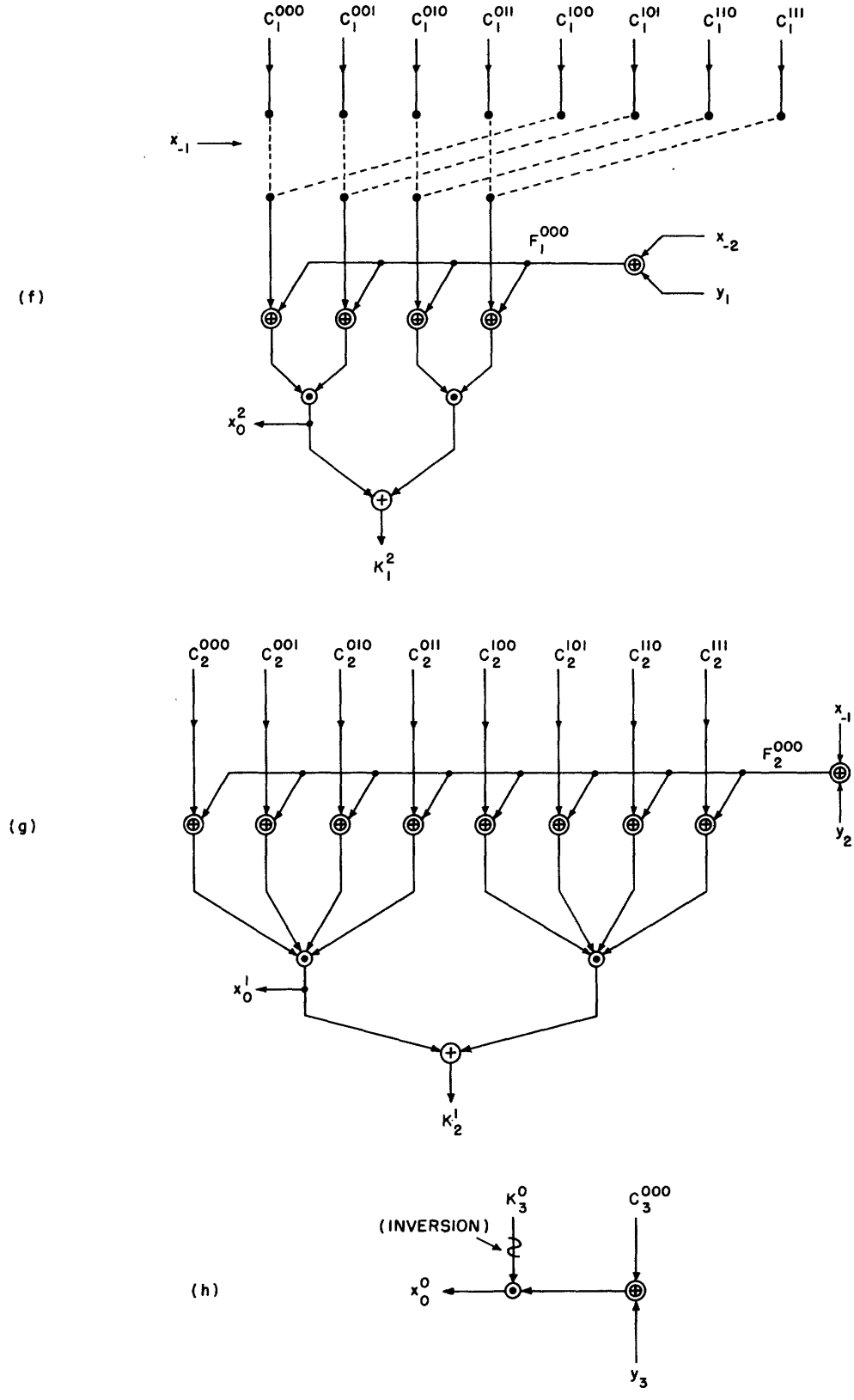


Fig. 15 (cont.) - The components of a circuit inverse to a third-order lossless circuit.

Reprinted from the Transactions of the
1959 International Symposium on Circuit and Information Theory