Theory of Logical Nets*

ARTHUR W. BURKS† AND JESSE B. WRIGHT†

Summary—**The aim of this paper is to place on a more rigorous basis the application of two-valued logic to digital computing circuits. To this end we study the formal properties of various logical nets some of which represent such circuits to a useful degree. A logical net is an array constructed from two primitive elements: a "stroke element," by means of which circuit components performing logical functions may be represented, and a "delay element," used in the representation of memory components.**

With each net is associated a set of equations useful in studying the behavior of that net. Various types of nets (well-behaved, deterministic, well-formed) are precisely defined and properties of each type are correlated with properties of the associated sets of equations. The relations between the various types of nets are investigated and some study is made of the kinds of operations performed by each. Finally, the question: "To what class of nets do digital computer circuits correspond?" is discussed.

Introduction

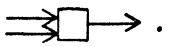
IT HAS BEEN shown by Shannon[1] and McCulloch and Pitts[2] how two-valued[3] symbolic logic may be employed to characterize the behavior of digital computing circuits; e.g., relay circuits,[4] digital electronic computing machines, neuron nets. The purpose of the present paper is to help place the application of two-valued logic to such circuits on a formal and rigorous basis.

We are concerned with two kinds of entities, logical *nets* and digital computing *circuits*. A net N usefully *represents* a circuit C (alternatively, C *physically realizes* N) when the physical behavior of C is mirrored in an idealized but nevertheless useful way by the logical behavior of N. To realize the purpose mentioned in the summary above, we will define various kinds of nets, study their formal properties and discuss the extent to which they can be physically realized.

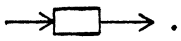
Primitive Elements

Since the physical components used in digital computing circuits perform either a logical function or a memory[5] function, or both, the logical analysis of these circuits is facilitated by employing distinct primitive elements for these two functions. Moreover, a single primitive is sufficient for each function, so only two primitive elements are required: the stroke element and the delay element.

A *stroke element* consists of a nucleus with two input *wires* and an *output wire* and is symbolized by:



Each net wire has one of two states (0, 1) at each discrete point of time 0, 1, 2, 3, · · · . Moreover, for each time point t the output wire of a stroke element is in the state 0 if and only if both input wires are in the state 1 (Sheffer's stroke function). A delay element consists of a nucleus, an input wire, and an output wire, and is symbolized by:



Each of its wires has one of two states (0, 1) at each discrete point of time, as in the case of the stroke element. The output wire of a delay element is in state 0 at $t=0$, and thereafter it possesses the state possessed by the input wire at the prior point of time.

A *net* is a finite array of elements interconnected so that only the free ends of wires are connected. A *junction* of a net is a point common to one or more ends of wires. If there is no arrowhead at a junction it is an *input junction;* otherwise it is an *output junction.* An output junction all of whose output wires are stroke element output wires is a *stroke output junction;* otherwise it is a *delay output junction.*

We have chosen these primitives because they possess all the following three properties: (1) nets of them are very useful in studying the behavior of digital computing circuits, (2) they are sufficiently simple to permit convenient logical analysis, and (3) small nets of them correspond to the physical components of digital computers. There are, of course, logically equivalent sets of primitives. The stroke element may be replaced by a conjunctive element and a negation element or by a disjunctive element and a negation element. The stroke element may also be replaced by a material implication element, the equivalent of:

Material implication is not by itself a sufficient basis for

* Decimal classification: 621.375.2. Original manuscript received by the Institute, March 9, 1953.

† University of Michigan, Ann Arbor, Mich.

[1] Claude E. Shannon, "A symbolic analysis of relay and switching circuits," *Transactions AIEE*, vol. 57, pp. 713–723; 1938.

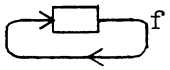
[2] Warren L. McCulloch and Walter Pitts, "A logical calculus of the ideas immanent in neuron activity," *Bull. of Math. Biophysics*, vols. 5–6, pp. 115–133; 1943–44. The diagrams used are similar to John von Neumann's modification of the diagrams herein; cf. Douglas R. Hartree, "Calculating Instruments and Machines," University of Illinois Press, pp. 97–111; 1949.

[3] Though the present paper is concerned exclusively with two-valued logic, its results are to a certain extent applicable to circuits containing wires having more than two significant states. This application can be made by allowing several wires of a logical net to correspond to a single wire of a physical circuit. Thus ten of the sixteen different states of four binary net wires can represent ten discrete electrical states of a single circuit wire.

[4] The theory developed here is directed mainly to electronic computer circuits, though it is to a degree applicable to relay circuits.

[5] The British "storage" is better than the American "memory" here, since what are called memory components do not by themselves recall or associate information, but only store it.

the propositional calculus, but it and the constant falsehood 0 are;[6] the latter may be realized by:

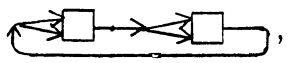


in our system, since $f_t \equiv 0$ for all t.

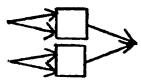
Finally, the delay element may be replaced, in the presence of the stroke element, by a binary counter with an input for counting and an output in the state 0 or 1 according to the state of the counter, or by a flip-flop with a set input, a reset input, and an output. It is not difficult to show that the equivalent of a delay element can be constructed from either of these and the stroke element.

Well-Behaved Nets

For a net N to usefully represent a circuit C there must be a correlation between some of the wires, or sets of wires, of N and some of the wires, or sets of wires, of C such that the states of the designated wires of C are represented by the states of the corresponding wires of N. Now in any well-behaved circuit the state of every wire at every time t is determinate, and, moreover, all wires joined to the same junction are in the same state. Consequently, we are interested only in nets in which (1) the state of every wire is uniquely determined by the defining properties of the primitive elements and (2) the states of all wires attached to a given junction are the same. We shall call such nets well-behaved (w.b.).[7]



is not well-behaved because it does not meet the first requirement, while:



is not well-behaved because of the second requirement. Physical circuits composed of components which in isolation realize stroke elements and which are connected according to these diagrams would, of course, have some definite behavior (the wires of such a circuit might oscillate between two states), but this behavior would be controlled by factors we have deliberately omitted from our idealization and would not mirror the logical behavior of the nets.

In this section we shall define precisely the class of well-behaved nets. We begin by correlating to each net a set of logical equations descriptive of its behavior. Any propositional function of one variable which ranges over the natural numbers and whose values are 0 (false) and 1 (true) will be called, simply, a *function*. We associate with each junction of a net a distinct function *variable*. (These variables will also be used to name the junction and each of its wires.) A variable is an *input*, *output*, *stroke output*, or *delay output variable* according as its junction is an input, output, stroke output, or delay output junction. Finally, we *associate* with each net N a *set of equations* $E(N)$ obtained as follows. For each stroke element with input wires g and h and output wire f write the *stroke equation* $f \equiv Sgh$, and for each delay element with input wire g and output wire f write the *delay equation* $f \equiv Dg$. In accord with the interpretation of our primitives, the first equation is equivalent to $f_t \equiv \sim g_t \vee \sim h_t$ and the latter is equivalent to $f_0 \equiv 0$ and $f_{t+1} \equiv g_t$, where t ranges over the natural numbers. It is sometimes convenient to have an infinite sequence $E_0(N), \cdots, E_t(N), \cdots$ of sets of equations derived from $E(N)$ as follows: $E_0(N)$ is the result of replacing each stroke equation $f \equiv Sgh$ of $E(N)$ by $f_0 \equiv Sg_0h_0$ and each delay equation $f \equiv Dg$ of $E(N)$ by $f_0 \equiv 0$, while $E_t(N)$ (for each $t>0$) is the result of replacing each $f \equiv Sgh$ by $f_t \equiv Sg_th_t$ and each $f \equiv Dg$ by $f_t \equiv g_{t-1}$.

Given any set of (delay and stroke) equations, a net with which this set is associated may be constructed as follows. Make a junction for each variable occurring in the given set of equations and label it with that variable. For each stroke equation $f \equiv Sgh$ place a stroke element from g and h to f and for each delay equation $f \equiv Dg$ place a delay element from g to f. We shall make the correspondence between nets and sets of equations one-one by arbitrarily identifying all sets of equations differing only with respect to the variables used and with respect to the order of the variables following a stroke operator.

The concept of a well-behaved net introduced informally in the first paragraph of this section may be precisely defined in terms of the uniqueness of the solution of the associated set of equations. A net N with input junctions $a^1, \cdots, a^J$ and output junctions $b^1, \cdots, b^K$ is *well-behaved* if and only if for each sequence of functions $a^1, \cdots, a^J$ $(J \geqq 0)$ there exists a unique sequence of functions $b^1, \cdots, b^K$ $(K>0)$ such that the a^j's and b^k's satisfy $E(N)$. (Here and hereafter when $I=0$, $f^1, \cdots, f^I$ is to be interpreted as denoting the null sequence.)

Some auxiliary notions, of use later, will now be defined. Let T be any *transformation* or mapping from a sequence of functions $f^1, \cdots, f^J$ $(J \geqq 0)$ to a function g.[8] If g is the same for all sequences, T is a *constant transformation*.[9] A transformation may be regarded as an operation on zero or more denumerable sequences of binary digits (0's and 1's), the values of the corresponding functions for $t=0, 1, \cdots$. Thus the delay transforma-

[6] Alonzo Church, "Introduction to Mathematical Logic," Princeton University Press, Princeton, N. J., part I, p. 3; 1944.

[7] It is worth noting that a circuit might be well-behaved for a certain combination of inputs and not for others. This suggests defining a concept of well-behavedness for nets which is relative to a certain class of input sequences. However, it is simpler to work with an absolute sense of well-behaved net and this is what will be done here.

[8] If a transformation T_1 of I arguments $(I \geqq 0)$ and a transformation T_2 of J arguments $(J>I)$ are such that there exists a sequence of numbers $i_1 < i_2 < \cdots < i_I \leqq J$ such that for all $f^1, \cdots, f^J$ $T_1(f^{i_1}, \cdots, f^{i_I}) = T_2(f^1, \cdots, f^J)$, then T_1 and T_2 shall be called the same transformation.

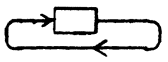
[9] The values of the function g itself are not necessarily the same. There is a constant transformation to the sequence 010101 . . . .

tion D shifts the sequence it operates on one position to the right and places a zero at its beginning. A *junction* g of a well-behaved net N *realizes a transformation* T of I arguments ($I \geqq 0$) if, and only if, among the input junctions $a^1, \cdots, a^J$ of N there are I of them $(a^{h_1}, \cdots, a^{h_I})$ such that for each sequence of functions $a^1, \cdots, a^J$ the g which satisfies $E(N)$ is equal to $T(a^{h_1}, \cdots, a^{h_I})$. A junction which realizes a constant transformation is called a *constant junction.*

A constant transformation to g is *periodic* if and only if there are natural numbers x and y, $x>0$, such that for every natural number n, $g_{t+nx+y} \equiv g_{t+y}$.

Theorem I: *Every constant transformation realized by a well-behaved net is periodic.*

Proof: Consider a w.b.n. N with junctions $g^1, \cdots, g^K$ where g^1 is a constant junction. Form $\overline{N}$ from it (with each g^K becoming $\bar{g}^K$) by connecting:



to all input junctions of N. Since g^1 realizes a constant transformation, $\bar{g}^1$ realizes the same transformation. Since every junction of $\overline{N}$ is a constant junction, there is a single matrix M satisfying $E(\overline{N})$, where the rows of the matrix give the solution for the $\bar{g}^k$'s and the columns $m_0, m_1, \cdots$ give the state of $\overline{N}$ for times 0, 1, $\cdots$ respectively. $\overline{N}$ has at most 2^K states, and hence for some x and y, $m_x = m_{x+y}$. Form a new matrix M' such that for $t<x$, $m_t' = m_t$ and for $t \geqq x$, $m_t' = m_{t+y}$. We will prove that $\bar{g}^1$ is periodic by showing that $M = M'$. Any matrix satisfies $E(\overline{N})$ if and only if (1) the first column satisfies $E_0(\overline{N})$ and (2) for every t greater than 0 the $t-1st$ and t'th columns satisfy $E_t(\overline{N})$. But if t_1 and t_2 are both greater than zero, $E_{t_1}(\overline{N})$ and $E_{t_2}(\overline{N})$ differ only in their arguments. Hence, condition (2) is equivalent to all pairs of adjacent columns, relabelled for $t=0$ and $t=1$, satisfying $E_1(\overline{N})$. Now any pair m_{t-1}' and m_t' ($t>0$) satisfies $E_1(\overline{N})$ in this way, for all pairs m_{t-1} and m_t satisfied $E_1(\overline{N})$ in this way, and m_{x-1}', m_x' is the same pair as m_{x-1}, m_x (since $m_x = m_{x+y}$). Since $\bar{g}^1$ is periodic, g^1 is also.

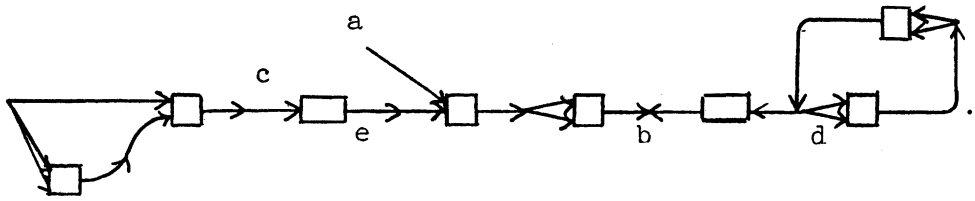
A transformation T from $a^1, \cdots, a^J$ to g ($J \geqq 0$) is *primitive recursive* whenever g can be defined by the operations of primitive recursion and substitution from the successor function, the constant functions, the identity functions, and $a^1, \cdots, a^J$.[10]

Theorem II: *There are primitive recursive transformations not realized by well-behaved nets.*

Proof: For example, the constant transformation to, $f_t \equiv 1$ for t a perfect square, $f_t \equiv 0$ otherwise (i.e. 1100100001 $\cdots$) is clearly a primitive recursive transformation, yet it is a constant transformation which is not periodic and hence by *Theorem* I it cannot be realized by any well-behaved net. It is worth noting in this connection that because of its infinite tape a Turing machine[11] is not a net in our sense.

Deterministic Nets

Not all well-behaved nets are physically realizable. Consider N_1 shown in the diagram below. It is well-behaved, with $c_t \equiv 1$ (for all t) and hence $b_0 \equiv 0$, $b_{t+1} \equiv a_{t+1}$ and finally $d_t \equiv a_{t+1}$. Thus, junction d "anticipates" the future state of input junction a, and the delay element db performs a kind of inverse delay or predictive operation. Obviously no circuit can perform such a predictive function, for there is no circuit component which can mirror the behavior of the delay element db. More generally, delay elements are intended to represent mechanisms in which signals on input wires cause signals to appear, after a suitable delay, on output wires, and we are interested primarily in nets which allow this interpretation. In other words, we are interested in nets in which all delay elements perform a memory or storage, rather than anticipatory or predictive function. We will call nets of this kind deterministic nets.



Consider next N_2, formed from N_1 by joining a and e. N_2 is well-behaved, with $e_0 \equiv 0$, $e_{t+1} \equiv 1$, $b_t \equiv e_t$, and $d_t \equiv 1$. Though the constant transformation to d can, in this case, be realized by a well-formed net which can in turn be physically realized (*Theorem* XIII), it is nevertheless the case that the delay element db performs in N_2 a predictive rather than a memory function, and that hence N_2 is not *deterministic.* That db does behave in this way may be seen by examining $E_0(N_2)$, $E_1(N_2)$, etc. $E_0(N_2)$ does not determine a unique value of d_0; rather, both $d_0 \equiv 1$ and $d_0 \equiv 0$ satisfy $E_0(N_2)$. It is only when $E_1(N_2)$ is considered that the possibility of $d \equiv 0$ is excluded. This analysis leads directly to one definition of deterministic net. Let N have input junctions $a^1, \cdots, a^J$ ($J \geqq 0$) and output junctions $b^1, \cdots, b^K$. First definition: N is *deterministic* if and only if for each sequence $a_0^1, \cdots, a_0^J; \cdots; a_t^1, \cdots, a_t^J$ ($t \geqq 0$) there exists a

[10] S. C. Kleene, "Recursive predicates and quantifiers," *Transactions AMS*, vol. 53, p. 42; 1943. (In this definition, function is used in a sense broader than that defined earlier in this section, since it includes any single-valued function of one or more natural numbers whose values are natural numbers.)

[11] A. M. Turing, "On computable numbers, with an application to the Entscheidungs-Problem," *Proc. London Math. Society*, vol. 42, pp. 230–265; 1936–37.

unique sequence $b_t^1, \cdots, b_t^K$ satisfying the union of $E_0(N), \cdots, E_t(N)$.

Where a net with input junctions is involved, this definition means roughly that the states of the input junctions for the past and the present determine the states of the output junctions for the present. An alternative conception of determinism is that of the state of the net at the immediate past $(t-1)$ and the state of the input junctions at the present (t) determining the state of the output junctions at the present. Since this conception is also useful, we will give a second definition of determinism and then prove that the two definitions are equivalent. Second definition: N is *deterministic* if and only if both of the following hold: (1) for each $a_0^1, \cdots, a_0^J$ there exists a unique $b_0^1, \cdots, b_0^K$ such that $a_0^1, \cdots, a_0^J; b_0^1, \cdots, b_0^K$ satisfies $E_0(N)$; and (2) if $t>0$ then for each $a_0^1, \cdots, a_0^J; \cdots; a_{t-1}^1, \cdots, a_{t-1}^J$ and $b_0^1, \cdots, b_0^K; \cdots; b_{t-1}^1, \cdots, b_{t-1}^K$ satisfying the union of $E_0(N), \cdots, E_{t-1}(N)$ and each $a_t^1, \cdots, a_t^J$, there exists a unique $b_t^1, \cdots, b_t^K$ such that $a_{t-1}^1, \cdots, a_{t-1}^J; a_t^1, \cdots, a_t^J; b_{t-1}^1, \cdots, b_{t-1}^K; b_t^1, \cdots, b_t^K$ satisfies $E_t(N)$. (If $J=0$ various of the sequences involved are to taken to be the null sequence as before.)

Theorem III: *The two definitions of determinism are equivalent.*

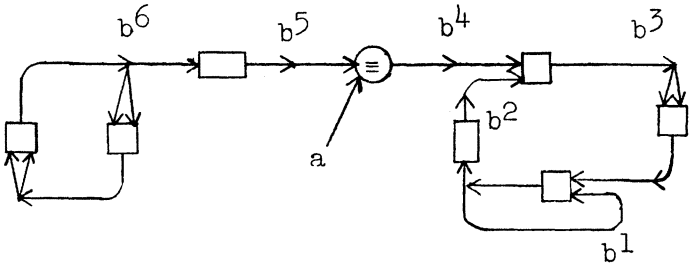
Proof: That the second definition implies the first may be shown by a simple induction. The proof that the first definition implies the second is as follows. Let N be deterministic by the first definition. Then $a_0^1, \cdots, a_0^J; \cdots; a_{t-1}^1, \cdots, a_{t-1}^J$ determines a unique $b_0^1, \cdots, b_0^K; \cdots; b_{t-1}^1, \cdots, b_{t-1}^K$ satisfying the union of $E_0(N), \cdots, E_{t-1}(N)$; and both of these sequences plus $a_t^1, \cdots, a_t^J$ determine a unique $b_t^1, \cdots, b_t^K$ satisfying the union of $E_0(N), \cdots, E_{t-1}(N)$ plus $E_t(N)$. But since $E_t(N)$ involves only the a^i's and b^k's at $t-1$ and t, it follows that the $a_{t-1}^1, \cdots, a_{t-1}^J; a_t^1, \cdots, a_t^J; b_{t-1}^1, \cdots, b_{t-1}^K$ determines a unique $b_t^1, \cdots, b_t^K$ satisfying $E_t(N)$. This last argument makes use of the fact that long-term memory (memory lasting more than one unit of time) in a net is the result of a succession of unit delays. This would not be the case, for example, if we had an additional two-wire primitive element capable of n units of delay, $n>1$; if that were the case the two definitions would not be equivalent.

Theorem IV: *Every deterministic net is well-behaved.*

Proof: This is obvious from the definitions.

We shall next define some concepts which will be of use in characterizing nondeterministic as well as other kinds of w.b.n. Any junction to which two or more output wires are connected is a *multiple junction.* A junction f *directly drives* a junction g if and only if f is the input of a stroke element whose output is g. (The input of a delay element does not directly drive its output.) f *drives* g if and only if there exists a sequence $f^1, f^2, \cdots, f^I$ such that f^1 is f, f^I is g, and f^i directly drives f^{i+1} for $i<I$. The driving relation is transitive: if f drives g and g drives h then f drives h. A *stroke cycle* is a sequence of junctions $f^0, \cdots, f^{I-1}$ such that $f^{i \bmod I}$ directly drives $f^{(i+1) \bmod I}$.

It will be observed that N_1 and N_2 contain multiple junctions, stroke cycles and delay elements. Not every nondeterministic w.b.n. contains a multiple junction, however, as the following net proves:



where the circle with the equivalence sign in it represents a net (constructable from stroke elements) realizing $b^4 \equiv (b^5 \equiv a)$. $b_t^1 \equiv 1$ and hence $b_{t+1}^2 \equiv 1$, also $b_t^3 \equiv 1$. These two equations imply that $b_{t+1}^4 \equiv 0$, which in turn implies that $b_{t+1}^5 \equiv \sim a_{t+1}$. Since $b_t^6 \equiv b_{t+1}^5$, it follows that $b_t^6 \equiv \sim a_{t+1}$, and so the net is nondeterministic.

Theorem V: *Every nondeterministic w.b.n. contains a stroke cycle and a delay element.*

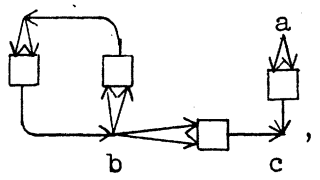
Proof: It will be simpler to give the proof of the first part in the next section (following *Theorem* X). The second half may be stated alternatively: every w.b. *stroke net* (a net constructed exclusively of stroke elements) is deterministic. The proof of it is as follows: Let N be a w.b. stroke net with input junctions $a^1, \cdots, a^J$ and output junctions $b^1, \cdots, b^K$. Then $E(N)$ contains no delay equations, all functions in $E_t(N)$ have t as an argument (never $t-1$), and hence the state of the net at t cannot place any restrictions on the state of the net at $t-1$, and vice versa. Therefore, for each $a_t^1, \cdots, a_t^J$ there is a unique $b_t^1, \cdots, b_t^K$ satisfying $E_t(N)$, since for each $a^1, \cdots, a^J$ there is a unique $b^1, \cdots, b^K$ satisfying $E(N)$. The basic idea of this proof can be employed to prove a stronger result concerning w.b. stroke nets, namely:

Theorem VI: *All transformations realized by w.b. stroke nets are truth-transformations.*

The concept of truth-transformation is defined as follows. Some transformations T (from $a^1, \cdots, a^J$) to g are equivalent to a sequence of mappings $T_0, T_1, \cdots$, each T_t being from a sequence of J truth-values to a single truth-value (0, 1). If all T_t are identical, T is said to be a *truth-transformation.* The result that all transformations realized by w.b. stroke nets are truth-transformations is a justification of an earlier statement that the stroke element performs only a logical function; a w.b. stroke net has no memory, its state at a given time never depends on its state at a prior time and never influences its state at a later time.

Well-Formed Nets

Earlier reference was made to physical components which perform a logical rather than memory function and we have represented these in our system by stroke nets. In actual fact, these components function much the same as the components realizing delay elements in that signals on their input wires cause signals to appear, after a certain delay, on output wires. The differences between the temporal lag inherent in a logical component and that in a memory component are that the former is generally smaller than the latter and generally serves no useful purpose. On this account, and in the interest of logical simplicity, our theory includes a temporal delay only in the delay element. As a consequence, however, we should not expect the requirement of determinism to eliminate all nets in which there is a backward passage of causal influence through stroke elements. Thus in the w.b.n.:



information flows from the output to the inputs of the stroke element *bc*, yet this net is deterministic, with $a_t \equiv \sim c_t \equiv b_t$. Since no component can mirror the logical behavior of the stroke element *bc*, though the transformation it realizes is easily realized physically, not all stroke elements in deterministic nets perform functions which can be duplicated by physical components.

In the present section we will define a class of nets, called well-formed nets (w.f.n.), which excludes the example just discussed. For reasons given in the next section, this is the most useful class of nets for the study of the behavior of digital computing circuits. There are w.f.n. representing, with various degrees of utility and directness, systems of neurons (in which the time intervals are of the order of one millisecond), some systems of electromechanical relays (in which the time intervals are of the order of milliseconds), and many electronic computing machines composed of vacuum tubes, crystal rectifiers, acoustic delay lines, and electrostatic storage tubes. In particular, w.f.n. characterize very closely the behavior of that type of electronic digital computer whose action is governed by a *clock* which is the source of equally-wide equally-spaced standard pulses (with the time interval between pulses being of the order of one microsecond).[12] In fact, our theory is especially directed towards such computers. In this case, a 1 represents the occurrence of a pulse and a 0 the absence of a pulse, or vice versa.

The class of *well-formed* nets is defined recursively as follows, with the understanding that no net is w.f. unless its being so follows from these rules.

[12] Douglas R. Hartree, "Calculating Instruments and Machines," University of Illinois Press, pp. 97–111; 1949.

(1) (a) Stroke rule: A stroke element is w.f.
(b) Delay rule: A delay element is w.f.

(2) Assume N_1 and N_2 are disjoint w.f.n. with junctions $f^1, \cdots, f^J$ and $g^1, \cdots, g^K$ respectively.
(a) Juxtaposition rule: The juxtaposition of N_1 and N_2 is w.f.
(b) Cascade rule: The result of joining junctions $f^{q_1}, \cdots, f^{q_I}$ of N_1 to distinct input junctions $g^{p_1}, \cdots, g^{p_I}$ of N_2 respectively is w.f.
(c) Input connection rule: The result of joining input junctions f^p and f^q of N_1 is w.f.
(d) Cycle rule: If all the wires connected to f^p of N_1 are delay element input wires, then the result of joining any f^q of N_1 to f^p is w.f.

A w.f.n. N may conveniently be studied by means of $E(N)$. $E(N)$ is *unitary* if and only if no variable has more than one left-hand occurrence (to the left of the equivalence sign in a stroke or delay equation) in $E(N)$. $E(N)$ is *regular* if and only if there is an ordering (called a *regular ordering*) of $E(N)$ such that for all f if f has a right-hand stroke equation occurrence in $E(N)$ then there is no left-hand occurrence of f in that or any later equation. The regular ordering of a regular set of equations can be obtained by using the concept of rank. *Ranks* may be assigned to certain of the junctions and corresponding equations of any net as follows. First, assign the rank 0 to all input junctions and all delay output junctions to which no stroke element output wires are connected. Then iterate as long as possible the general step: Assign the rank r to all junctions f such that some junction directly driving f has rank $r-1$ and every junction directly driving f has a rank $\leqq r-1$. Finally, to every equation $f \equiv$— which is such that the junction f has a rank, assign the rank of f.

Theorem VII: *N possesses no multiple junctions if and only if $E(N)$ is unitary; and N contains no stroke cycles if and only if $E(N)$ is regular.*

Proof: The first half is obvious. The *only if* part of the second half follows from the fact that the set of equations for a stroke cycle is not regular and hence any set of equations containing them is not regular. That $E(N)$ is regular if N contains no stroke cycles may now be proved as follows. We prove first that since N contains no stroke cycles every junction of N is assigned a rank. Assume some junction f of N has no rank. Then there must be a junction g^1 which directly drives it which has no rank. In turn, there must be a g^2 directly driving g^1 and without rank, ad infinitum. But there are only a finite number of junctions in N, so some variable g^i in the sequence $g^1, g^2, \cdots$ must occur twice in this sequence. Hence N contains a stroke cycle, contrary to our assumption. Since every junction of N is assigned a rank, every equation of $E(N)$ is assigned a rank, and these equations may be arranged in order of ascending rank. We prove finally that this ordering is regular. Consider any equation of the form $f \equiv Sgh_1$ where g is of rank

r. Then f is of rank $\geqq r+1$; any equation of the form $g \equiv$ — is of rank r and hence must precede $f \equiv Sgh$.

Theorem VIII: *N is w.f. if and only if $E(N)$ is both unitary and regular.*

Proof: We prove first that if N is w.f. it contains no multiple junctions and no stroke cycles. A stroke element or delay element by itself (cf. stroke and delay rules) contains no multiple junctions or stroke cycles. Moreover, if N_1 and N_2 contain no multiple junctions or stroke cycles, the result of combining them by any of the four rules of combination will not contain any multiple junctions or stroke cycles. It follows by *Theorem* VII that if N is w.f., $E(N)$ is both unitary and regular. We will prove the converse by giving a stepwise process for constructing N on the basis of S, a regular ordering of $E(N)$. Let S_i be that subsequence of S which consists of the first i equations of S. The net associated with S_1 is w.f. (by rules 1, 2*c*, and 2*d*) since $E(N)$ is regular. Assume now that $N_1{}^i$ is formed from S_{i-1} and is w.f. and consider the i'th equation of S. The element associated with it may have two input wires which are paired or not; in the former case use the input connection rule to get $N_2{}^i$; in the latter case take the element itself to be $N_2{}^i$. $N_2{}^i$ may have no input wire connected to a junction of $N_1{}^i$ in N or one or more such; in the former case juxtapose $N_2{}^i$ and $N_1{}^i$; in the latter case cascade $N_1{}^i$ onto $N_2{}^i$. Finally, consider the output f of $N_2{}^i$. Since S_i is regularly ordered and unitary, the only occurrences of f other than the left-hand occurrence in the last equation of S_i are right-hand delay equation occurrences in the equations of S_i. Hence the output of $N_2{}^i$ can be connected into the net, if necessary, by the cycle rule.

Theorem IX: *$E(N)$ is regular if and only if every junction of N has a rank.*

Proof: In *Theorem* VII it was shown that if every junction of N has a rank, then $E(N)$ is regular; and also that if $E(N)$ is regular, then N contains no stroke cycles and that if N contains no stroke cycles then every junction of N has a rank.

Thus the following four properties of N are equivalent: (1) N is w.f., (2) N has no multiple junctions and no stroke cycles, (3) N has no multiple junctions and every junction of N has a rank, and (4) $E(N)$ is both unitary and regular, and any one may be employed as a criterion of a w.f.n. The formation rules are naturally and simply applied when a net is being constructed. Given a complicated net, however, it may be difficult to determine whether or not it can be constructed by the rules. In this case a test procedure based on criterion (3) is useful. Label all junctions of rank 0, stopping if any are multiple junctions (since then N is not w.f.). Then iterate the following step as long as possible: label all junctions of rank r, stopping if any are multiple junctions. If this procedure terminates without a multiple junction being discovered, the N is w.f. if and only if all junctions have been assigned ranks (by *Theorem* IX).

Theorem X: *Every w.f.n. is deterministic, hence well-behaved.*[13]

Proof: The determinism of a w.f.n. follows from the fact that, in the presence of unitariness, regularity is stronger than determinism, allowing not only sequential computation with respect to time but also sequential computation at a given time. For each t the state of a w.f. N may be computed by proceeding through a regular ordering S of $E(N)$ as follows. If the equation at hand is a delay equation $f \equiv Dg$, then for $t=0$, $f_0 \equiv 0$ and for $t>0$, $f_t \equiv g_{t-1}$, where g_{t-1} is known from the state of the net at $t-1$. If the equation at hand is a stroke equation $f \equiv Sgh$, then, since S is in regular order, g and h have no left-hand occurrences in that or any later equations of S. Hence g (h) either has a left-hand occurrence in an earlier equation of S, in which case g_t (h_t) is already known from the preceding computation; or g (h) has no left-hand occurrence in any equation, in which case it is an input variable and g_t (h_t) is given. Since $E(N)$ is unitary, this procedure gives a unique value for each f_t. Hence N is deterministic and by *Theorem* IV it is also w.b. An alternative way of looking at the computation is in terms of rank: the states of all junctions of rank 0 may be determined, then those for rank 1, etc.

By a procedure similar to that just used, we will now prove the first half of *Theorem* V, namely, that every nondeterministic w.b.n. contains a stroke cycle. This may be stated alternatively as: Every w.b.n. containing no stroke cycles is deterministic. The proof is as follows. Let N be a w.b.n. with no stroke cycles. By *Theorem* VII $E(N)$ is regular. Hence the behavior of N can be computed in the manner indicated in the last paragraph, and since N is w.b. the result will be unique.

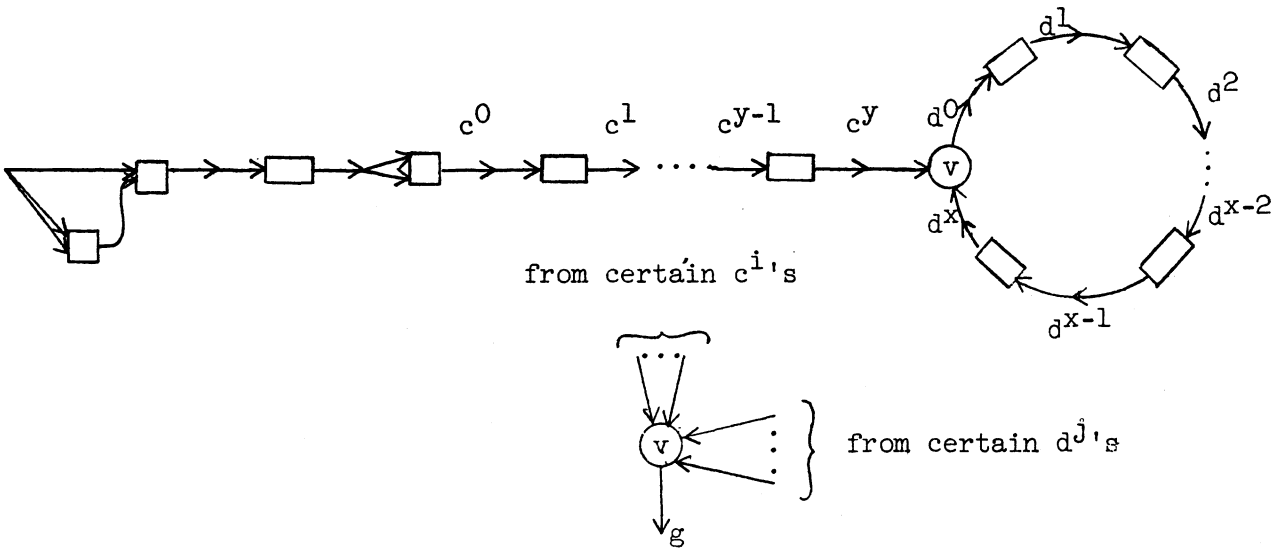
N_1 is a *subnet* of N if and only if N_1 may be formed from N by a succession, perhaps null, of the following operations: (1) separating a wire from a junction; (2) deleting an element.

Theorem XI: *N is w.f. if and only if all the subnets of N are w.b.*[14]

Proof: If N is w.f., it contains no multiple junctions or stroke cycles, hence neither do any of its subnets. Therefore each of its subnets is w.f. and hence w.b. Next we show that if N is not w.f. it contains a subnet which is not w.b. There are two cases to consider, according to whether N contains a multiple junction or not. (1) Suppose N contains a multiple junction f such that $f \equiv$ — and $f \equiv$... are in $E(N)$. Separate the input wires of the elements associated with each of the equations from their junctions. The result is not w.b. (2) Suppose N contains no multiple junctions but does contain a stroke cycle $f^0, \cdots, f^{I-1}$. We may assume without loss of generality that the associated sequence of equations is $f^{I-1} \equiv Sf^{I-2}g^{I-2}, \cdots, f^0 \equiv Sf^{I-1}g^{I-1}$. If any of

[13] The example given at the beginning of this section shows that the converse does not hold.

[14] This theorem was suggested by Paul Henle.



these equations is of the form $p \equiv Sqq$, separate one of the input wires of the corresponding stroke element from its junction, making it a net input. Then if any g^i input wire of an element corresponding to one of these equations is connected to an output junction, separate it from that junction so that it is a net input. The result is not w.b., shown by applying 1's to all input junctions.

Theorem XII: *A transformation is realizable by a w.f. stroke net if and only if it is a truth-transformation.*

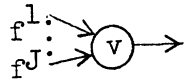
Proof: We have already proved (*Theorem* VI) that all transformations realized by w.b. (hence by w.f.) stroke nets are truth-transformations, so it need only be proved that all truth-transformations can be realized by w.f. stroke nets. Since Sheffer's stroke function is a sufficient primitive for defining all truth-functions in the logical sense, it is a sufficient primitive for defining all truth-transformations. The process of definition involves substitution and the use of the same variable in more than one argument place; these are mirrored by the cascade rule and the input connection rule, respectively.

The last half of the theorem shows that in the presence of the formation rules a stroke element is a sufficient primitive for our purposes. It follows from the definition of truth-transformation that there are only two constant truth-transformations: $g_t \equiv 1$ and its negation $h_t \equiv 0$. The former is realized by the w.f.n.:

which is of interest because it is a realization in our system of a *clock*.[15]

Theorem XIII: *A constant transformation is realized by a w.f.n. if and only if it is periodic.*

Proof: The *only if* part of this theorem follows from *Theorems* I and X. The converse is proved by showing how to construct a net realizing any g such that $g_{t+nx+y} \equiv g_{t+y}$. The net which is shown above is a schematic construction of the desired net, where:



represents a J-input disjunction, readily constructed from stroke elements. Note that $c_0{}^0 \equiv 1$, $c_{t+1}{}^0 \equiv 0$, and in general $c_t{}^i \equiv 1$ if and only if $t = i$, while $d_t{}^j \equiv 1$ if and only if $t = j + nx + y$. For a particular constant transformation to g this schema may be converted into a net realizing this transformation as follows. If $g_t \equiv 1$ for any t less than y, connect c^t to one of the inputs of the disjunctive net whose output is g; if $g_t \equiv 1$ for any t as large as y and less than $x+y$, connect d^t to one of the inputs of the disjunctive net whose output is g. g is the desired junction.

Theorem XIV: *Every transformation realized by a w.f.n. is primitive recursive.*

Proof: Too long to be included here, but the essential idea is as follows. Let N be w.f. with input junctions $a^1, \cdots, a^J$ and output junctions $b^1, \cdots, b^K$ in regular order, and let the sequence $b_0{}^1, \cdots, b_0{}^K; b_1{}^1, \cdots, b_1{}^K; \ldots$ be relabeled $e_1, \cdots, e_K; e_{K+1}, \cdots, e_{2K}; \cdots$. Since the behavior of N can be computed in the manner indicated in the proof of *Theorem* X, it is possible to define a mathematical function F (from natural numbers to natural numbers) which is primitive recursive with respect to the a^i's and is such that:

$$F(n) = 2^{e_{n+1}} \cdot \cdots \cdot p_{2K}{}^{e_{n+2K}},$$

where p_k is the k-th prime, in order of magnitude. Each b^k can then be defined primitive recursively in terms of F as follows: $b_t{}^k$ equals the first exponent of $F(tK+k-1)$.

W.F.N. and Digital Computing Circuits

In this section we will discuss the realization of w.f.n. by digital computing circuits. Consider first the fact that both the delay element and the stroke element (or small nets of it) are realized by physical components in

[15] As mentioned earlier in this section, our theory is especially applicable to circuits synchronized by pulses from a standard clock.

which the input wire states causally determine the output wire state. The output of the physical realization of a delay or stroke element in isolation has the property that its state can be causally determined by the states of its input wires and the formation rules for w.f.n. are such that every output wire of a circuit realizing a w.f.n. has this property. Thus the backward passage of causal influence through stroke elements (cf. the first net discussed in the preceding section) never occurs in a w.f.n. Hence, as far as these considerations are concerned every w.f.n. is physically realizable. There are, however, certain aspects of our idealization of a stroke element which require this conclusion to be restricted. In particular, though the delay of a component realizing a simple stroke net is generally small compared to the basic unit of time of the system, the same does not hold for circuits composed of such components. A practical way of dealing with this point is to construct circuits which perform fairly simple logical transformations and then feed the result into a delay line (as well as into power amplifying and retiming circuits), the temporal lag of the line being such that the total delay is one or a few units of time. This involves placing an upper limit on the allowable rank of a junction and on the number of junctions which drive a given junction. These restrictions could be removed from the net theory by taking as a primitive element a stroke element driving a delay element, but the additional complications would not be worth the gain. In any event, there is an over-all size limitation on the realizability of w.f.n.

The discussion of deterministic nets showed that if a circuit is represented by a well-behaved net that net is deterministic. Consider now the question: Are there any circuits represented by deterministic nets which could not be adequately represented by w.f. nets? To keep the range of application of the logic of nets as broad as possible, we have left the concept of representation (and its converse, realization) somewhat indefinite. It is sufficiently broad that some so-called *static* circuits and some mixed static and pulse circuits, as well as some pulse (*dynamic*) circuits can under suitable limitations be represented by w.f.n. The answer to the above question depends on the particular interpretation made and hence we will not attempt to give a general answer here; we will, however, make a few relevant comments. Engineers sometimes connect the outputs of two or more physical components together; this is a way of realizing logical disjunction and is better represented in our system by stroke nets realizing logical disjunction than by nets containing multiple junctions. Also, feedback loops are sometimes employed for memory purposes, as in the flip-flop, but these are better represented in our system by means of delay elements rather than by stroke cycles.[16]

On the other hand, consider the following example of a deterministic net not w.f. which may be physically realized in one sense of physical realization. Start with a single physical component realizing $f \equiv (g \cdot \sim h)$ and connects its inputs together so that $f \equiv (g \cdot \sim g) \equiv 0$. Its physical behavior would probably be unaffected by connecting its output back to its input. However, this connection, which introduces a stroke cycle in the corresponding net, is pointless, since it neither increases the number of transformations realized nor decreases the number of elements in the net. Thus this example gives rise to the following two theoretical questions which are worth investigating: (1) Can deterministic nets not w.f. realize transformations not realized by w.f.n.? and (2) Can they realize some transformations more efficiently? The answer with regard to (1) is given by:

Theorem XV: *Every transformation realized by a deterministic net is realized by a w.f.n.*

Proof: Let N be deterministic with input junctions $a^1, \cdots, a^I$, delay output junctions $b^1, \cdots, b^J$, and stroke output junctions $c^1, \cdots, c^K$; and let $\overline{N}$ be the desired w.f.n. whose junctions include the corresponding junctions $\bar{a}^1, \cdots, \bar{a}^I$; $\bar{b}^1, \cdots, \bar{b}^J$; and $\bar{c}^1, \cdots, \bar{c}^K$. We define a sequence of truth-values (0, 1) $x_1, \cdots, x_J$ to be *admissible*, relative to $E(N)$, if and only if it satisfies either of the following two conditions (cf. the second definition of determinism): (1) it consists of all 0's, and (2) there is some $t>0$ and some sequence $a_0^1, \cdots, a_0^I$; $b_0^1, \cdots$; b_0^J; $c_0^1, \cdots$; c_0^K, $\cdots$; $a_t^1, \cdots$; a_t^I; b_t^1, $\cdots$; b_t^J; $c_t^1, \cdots$; c_t^K satisfying the union of $E_0(N)$, $\cdots$, $E_t(N)$ and such that for each j, $b_t^j = x_j$. Let $E_t^c(N)$ consist of all equations of $E_t(N)$ of the form $c_t^k \equiv$— (i.e., of the form $c_t^k \equiv S \cdots$). Since N is deterministic it then follows that for each $a_t^1, \cdots, a_t^I$ and each admissible $b_t^1, \cdots, b_t^J$ there is a unique $c_t^1, \cdots, c_t^K$ such that $a_t^1, \cdots, a_t^I$; $b_t^1, \cdots, b_t^J$; $c_t^1, \cdots, c_t^K$ satisfies $E_t^c(N)$. Now define for each c_t^k a mapping T_t^k as follows: for each $a_t^1, \cdots, a_t^I$ and each admissible $b_t^1, \cdots, b_t^J$, $T_t^k(a_t^1, \cdots, a_t^I; b_t^1, \cdots, b_t^J)$ is that value of c_t^j which satisfies $E_t^c(N)$; for each $a_t^1, \cdots, a_t^I$ and each nonadmissible $b_t^1, \cdots, b_t^J$ $T_t^k(a_t^1, \cdots, a_t^I; b_t^1, \cdots, b_t^J) = 0$. Since each $E_t^c(N)$ is of the same form (i.e., a difference in t involves only a difference in the arguments), we may define for every c^k a truth-transformation T^k equivalent to the sequence of mappings $T_0^k, T_1^k, \cdots$. The net $\overline{N}$ may now be constructed as follows. For each b^j pick an equation of $E(N)$ of the form $b^j \equiv Df$ and connect a delay element from $\bar{f}$ to $\bar{b}^j$ in $\overline{N}$. For each $\bar{c}^k$ of $\overline{N}$ construct a w.f. stroke net (*Theorem* XII) from the $\bar{a}^i$'s and $\bar{b}^i$'s realizing T^k. The transformations realized by the $\bar{b}^i$'s and $\bar{c}^k$'s in $\overline{N}$ are the transformations realized by the b^i's and c^k's of N respectively, and since no stroke cycles and no multiple junctions were created in the formation of $\overline{N}$, $\overline{N}$ is w.f.

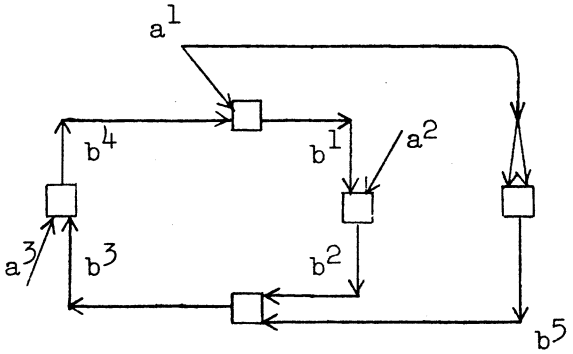
Consider now question (2). This involves the concept of simplicity. For nets composed of stroke and delay elements a rough measure of simplicity is given by the number of each. Some results concerning this rough measure will now be stated. Let us arbitrarily define N_1

[16] By altering the defining equations of a stroke-element we can obtain a "stroke net" which performs a memory function. For example, a net with inputs s (set) and r (reset) and junctions f and g such that $f \equiv (\sim s \cdot \sim g)$ and $g \equiv [\sim f \cdot (\sim r \vee s)]$ is a representation of a flip-flop under the stipulation that if $s_t \equiv r_t \equiv 0$ then (1) if $t=0$, then $f_t \equiv 0$ and (2) if $t>0$, then $f_t \equiv f_{t-1}$.

to be *as simple as* N_2 whenever N_1 has no more stroke elements and no more delay elements than does N_2. It should be remembered that logically equivalent primitives are not necessarily equivalent where simplicity is involved, so our subsequent theorems on simplicity do not necessarily hold for alternative sets of primitive elements logically equivalent to the set consisting of the stroke and delay elements (Primitive Elements). Question (2) can now be more precisely stated: Given a deterministic net N not w.f., is there a w.f.n. as simple as N which realizes the same transformations as N? The answer is in the negative.

Theorem XVI: *There is a deterministic net N such that no w.f.n. realizing all the transformations realized by N is as simple as N.*

Proof: Let N be:



N contains the stroke cycle b^1, b^2, b^3, b^4 and hence is not w.f., but by *Theorem* IV it is deterministic. It is easily verified that N realizes the distinct transformations:

$$b^1 \equiv (\sim a^1 \vee a^3)$$
$$b^2 \equiv (\sim a^2 \vee a^1 \sim a^3)$$
$$b^3 \equiv a^1 \vee a^2$$
$$b^4 \equiv \sim a^3 \vee \sim a^1 \sim a^2$$
$$b^5 \equiv \sim a^1$$

Consider now a w.f.n. N_1 with at least junctions b^1, b^2, b^3, b^4, and b^5, realizing the corresponding transformations. For each b^k, $1 \leqq k \leqq 5$ there must be exactly one equation in $E(N_1)$ of the form $b^k \equiv S$—. Replace $b^5 \equiv S$— by $b^5 \equiv Sa^1a^1$; this modification of N_1 does not change the total number of stroke elements in N_1 nor the number of transformations realized by N_1. Let the minimal rank of b^1, b^2, b^3, b^4 of N_1 be R and let b^x be one of the junctions b^1, b^2, b^3, b^4 of N_1 of this minimal rank. None of the transformations b^1, b^2, b^3, b^4 is equivalent to Sfg where f and g are a^1, a^2, a^3, or b^5, and hence b^x cannot be realized from a^1, a^2, a^3, b^5 by less than two stroke elements. Consequently, there is a junction b^6 directly driving b^x which is different from a^1, a^2, a^3, and b^5. b^6 is of rank less than R and hence realizes a transformation different from those realized by b^1, b^2, b^3, and b^4. But N_1 contains five other junctions realizing b^1, b^2, b^3, b^4 and b^5, and so N_1 contains at least six stroke elements. The set got from $E(N)$ by replacing $b^1 \equiv Sa^1b^4$ by $b^1 \equiv Sa^1b^6$ and $b^6 \equiv Sa^3a^3$ is associated with a six-stroke-element w.f.n. that realizes all the transformations realized by N.

Because *Theorem* XVI holds it is of interest to find necessary and sufficient conditions for a deterministic net to have the property that there exists as simple a w.f.n. realizing the same transformations. We have obtained a sufficient condition for this property involving the concept of a properly driven net.

Any junction of a net N is defined to be a *properly driven junction* of N if and only if it belongs to the smallest class satisfying the following rules:

(1) each input junction and each delay output junction is properly driven, and

(2) if g and h are properly driven and $f \equiv Sgh$, then f is properly driven.

A *properly driven net* (p.d.n) is then defined to be a w.b.n. in which every junction is properly driven. The following key theorem concerning p.d.n. is formulated in terms of the following concept. A set of equations $E'(N)$ is a *complete subset* of $E(N)$ if and only if for each output junction f of N $E'(N)$ contains at least one equation of $E(N)$ of the form $f \equiv$—.

Theorem XVII: *If N is p.d. E(N) contains a complete subset E'(N) which is unitary and regular.*

Proof: $E'(N)$ may be constructed as follows. For each delay output junction f of N select one equation of the form $f \equiv D$—. Then iterate this step: Add to $E'(N)$ any equation $f \equiv Sgh$ of $E(N)$ satisfying the conditions: (1) there is no equation of the form $f \equiv$— already in $E'(N)$; (2) both g and h satisfy the condition of having a left-hand occurrence in an equation of $E'(N)$ or being an input variable. It is obvious from its mode of construction that $E'(N)$ is unitary and regular, and it follows from the definition of a properly driven junction (since the above procedure will catch every properly driven junction, hence every junction of the net that) $E'(N)$ is a complete subset of $E(N)$.

Theorem XVIII: *If N is p.d., there is a w.f.n. as simple as N realizing all the transformations realized by N.*

Proof: The required w.f.n. is any subnet of N to which is associated a complete subset of $E(N)$ which is both unitary and regular.

Theorem XIX: *Every p.d.n. is deterministic.*

Theorem XX: *Every stroke cycle of a p.d.n. contains at least one multiple junction* (a characteristic distinguishing p.d.n. from other deterministic nets).

Proof: For if N is p.d. and contains a stroke cycle which has no multiple junctions, any complete subset of $E(N)$ must contain the equations of this stroke cycle and hence cannot be regular.

ACKNOWLEDGMENT

The writing of this paper and the research which it reports were done under the sponsorship of the Burroughs Adding Machine Corporation (completed December 23, 1952).

The authors wish to thank Carl H. Pollmar for many helpful suggestions.