

**THE UNIVERSITY OF MICHIGAN  
COMPUTING RESEARCH LABORATORY**

---

**PSEUDO-BOOLEAN LOGIC CIRCUITS**

**John P. Hayes**

**CRL-TR-33-84**

**August 1984**

**Room 1079, East Engineering Building  
Ann Arbor, Michigan 48109  
USA  
Tel: (313) 763-8000**



# **PSEUDO-BOOLEAN LOGIC CIRCUITS**

by

**John P. Hayes**

**Department of Electrical Engineering and Computer Science  
University of Michigan  
Ann Arbor, MI 48109**

**June 1984**

**This research was supported by the National Science Foundation under Grant ECS-8214709.**

## Abstract

A new class of switch-level logic circuits intended for modeling digital MOS VLSI circuits is presented. These circuits, which are called pseudo-Boolean, are composed of a single (voltage) source, connectors, switches, attenuators and wells. The latter two devices are digital versions of resistors and capacitors, respectively, and may assume an arbitrary but finite number of different sizes. Signals are bidirectional, and are assigned a finite set of values of the form  $(v, s)$ , where  $v$  corresponds to voltage level, and  $s$  corresponds to electrical current or charge level (logical strength). It is shown that these signal values and the associated logical operations form a generalization of Boolean algebra called pseudo-Boolean or Heyting algebra. The analysis of pseudo-Boolean circuits using discrete counterparts of Kirchoff's Current Law and the Superposition Principle is discussed, as well as the application of pseudo-Boolean techniques to digital simulation.

**Keywords:** Digital simulation, logic design, MOS circuits, pseudo-Boolean algebra, switch-level simulation, switching theory, VLSI design.

## 1. Introduction

In the design of MOS VLSI circuits, based for example, on the Mead-Conway philosophy [1], a central role is played by CAD programs that can accurately simulate the behavior and layout of very large logic circuits. Traditional gate-level logic simulators are of limited value for this purpose, since they cannot directly model such MOS logic elements as transistor switches, pull-up loads, bidirectional buses, and the like [2]. Analog simulators such as SPICE, on the other hand, although capable of modeling all types of MOS circuits at the electrical level, require too much computation time when applied to very large digital circuits. To deal with this situation, a new class of simulation programs called switch-level simulators have recently been developed [3,4,5]. These simulators employ transistor-like devices as primitive components, and so can more accurately model the behavior and layout structure of VLSI circuits than gate-level simulators. Because they represent circuit behavior by a small set of discrete logic values rather than a potentially infinite number of analog values, switch-level simulators can also handle much more complex circuits than analog simulators.

This paper investigates the theoretical basis of switch-level simulation models, with the goal of better understanding their underlying algebraic structure. Previous work in this area includes the connector-switch-attenuator (CSA) theory of Hayes [2,5], and the switch-level model theory of Bryant [4,7]. In the present paper a new and very general class of CSA switch-level logic circuits is defined which seem especially useful for MOS VLSI simulation. These circuits encompass most existing types of logic circuits, and allow the modeling accuracy to be varied systematically by changing the number of logic values and component parameters used. It is shown that the algebraic structure



underlying these circuits is a generalization of Boolean algebra called pseudo-Boolean or Heyting algebra [8], hence the logic circuits in question are termed pseudo-Boolean.

Section 2 defines the global structure of pseudo-Boolean logic circuits, and derives the set  $L_n$  of static logic values used from discrete current, voltage, and resistance concepts. In Sec. 3 the lattice structure of  $L_n$  is explored, and is shown to constitute a pseudo-Boolean algebra. A wide range of both unidirectional and bidirectional primitive logic elements for pseudo-Boolean circuits are discussed in Sec. 4. The use of dynamic logic values and digital charge-storage devices (wells) to model sequential behavior is considered in Sec. 5. Finally, Sec. 6 presents some applications of the proposed theory.

## 2. Basic Concepts

The logic values of interest are required to represent the binary electrical signals employed in real logic circuits. This binary requirement is met by permitting only one type of source device  $S$  for generating the logical constants 0 and 1.  $S$  corresponds to an electrical voltage source that produces a constant potential difference of  $V$  volts across its terminals. The terminals of  $S$  are held at two distinct voltage levels  $V_L$  and  $V_H = V_L + V$ , representing logical 0 and 1, respectively.

To ensure that all normal voltage levels are confined to the binary set  $\{V_H, V_L\}$ , we require that all source devices have one terminal connected to a common point (ground). This prevents two copies of  $S$  from being connected in series to generate a third voltage level  $V_L + 2V$ . We also require all voltage-like measurements to be made relative to ground. This eliminates the possibility of obtaining another voltage of the form  $-V$  from  $S$ . Thus all copies of the source  $S$  are effectively connected in parallel, with one terminal connected to a common ground that serves as a reference for logic value measurement.

Such a parallel connection of sources is functionally equivalent to a single source. Hence, without much loss of generality, we only consider logic circuits that contain a single voltage source  $S$  corresponding to the main electrical power supply. (Note that a dual theory can also be developed in which  $S$  is a current rather than a voltage source).

The overall structure of a single-source (pseudo-Boolean) logic circuit  $N$  is depicted in Fig. 1. Its purpose is to realize functions of the form  $f_j(X)$ , where  $X = (x_1, x_2, \dots, x_n)$  denotes a set of externally controlled input variables. The function value  $f_j(X)$  has a measurable voltage-like component  $v_j \in \{0, 1\}$ .  $v_j$  is determined by the connections established between the  $f_j$  output terminal of  $N$  and either the terminals  $S_0$  and  $S_1$  of the source  $S$  or the primary inputs  $X$ . These connections are made or broken by logic elements such as switches, gates, etc. For example, if switch settings in  $N$  create a path from  $f_j$  to the constant terminal  $S_1$  as shown in Fig. 2a, then  $v_j$  assumes the value 1. Similarly, a path joining  $f_j$  to  $S_0$  makes  $v_j = 0$ ; see Fig. 2b.

In most types of logic circuits,  $v_j$  is restricted to the set  $\{0, 1\}$ . However, there are two other possible values that  $v_j$  can assume, which are also illustrated in Fig. 2. If  $f_j$  is simultaneously connected to both sides of  $S$  (Fig. 2c), a situation that could result from a short-circuit fault, then  $v_j$  assumes a third value denoted  $U$ . This logic value can be thought of as an intermediate voltage level  $V_U$  lying somewhere between  $V_L$  and  $V_H$ , e.g.,  $(V_H - V_L)/2$ .  $f_j$  can also be completely disconnected from the voltage source  $S$ , as indicated in Fig. 2d, in which case it is assigned the value  $Z$  corresponding to the familiar high-impedance state. The output condition  $v_j = Z$  occurs explicitly in tristate logic circuits; in other logic circuits,  $Z$  is only encountered on internal lines. The set  $L_1 = \{0, 1, U, Z\}$  constitutes the basis from which all logic values are derived. It is easily seen that if the circuit  $N$  of Fig. 1 is composed of switches that make and break connec-



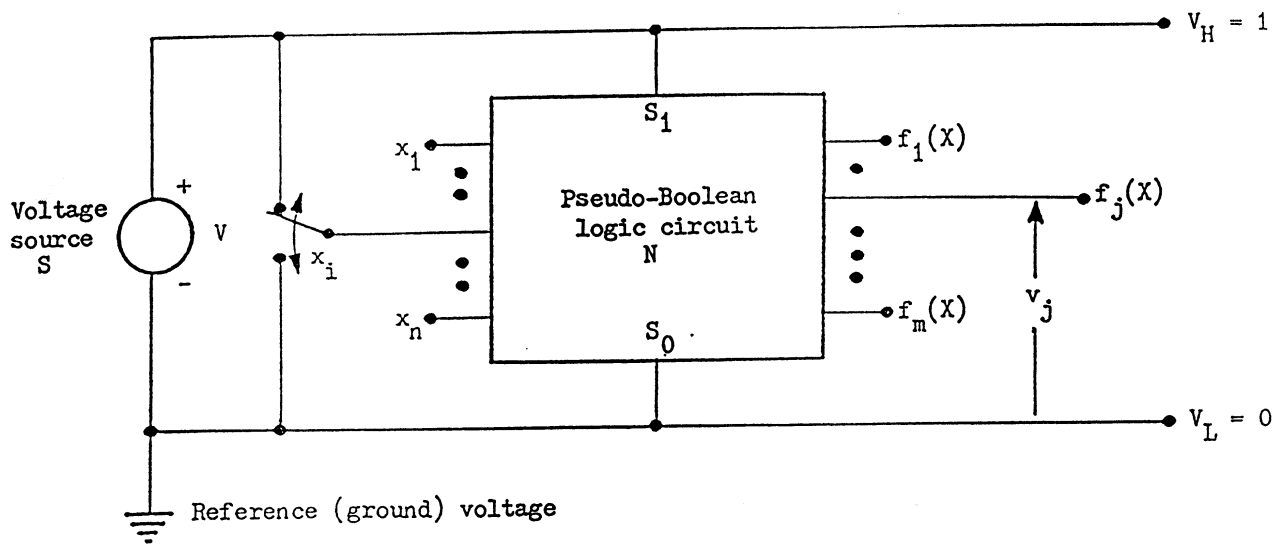


Fig. 1. Overall structure of a pseudo-Boolean circuit.

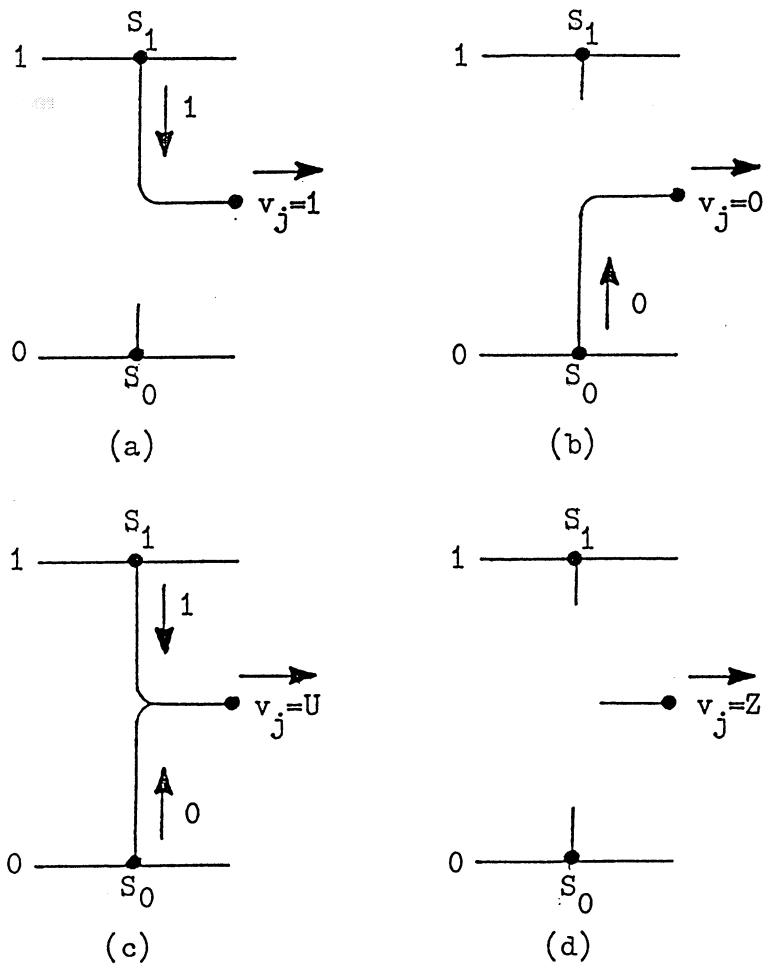


Fig. 2. Voltage values assumed by output  $f_j(X)$  of Fig. 1.

tions, any logic function defined on  $L_1$ , or its binary subset  $B_2 = \{0, 1\}$ , can be realized.

With ideal on-off switches of the type defined in Fig. 3 we can construct connector-switch (CS) networks [2] to realize any logic function defined on  $L_1$ . Positive and negative switches correspond to n-channel and p-channel MOS transistors, respectively. Figure 4, for instance, shows a CS implementation of the Boolean NAND function  $z = \overline{x_1 x_2}$  defined on  $B_2 = \{0, 1\}$ . This circuit closely models the structure and behavior of a CMOS NAND gate. The logic values  $Z, U \in L_1$ , are useful in analyzing the behavior of logic circuits of this type under fault conditions [9].

From an electrical viewpoint, a CS circuit such as that of Fig. 4 is a source-resistor network [10] in which only two resistance values occur:  $R_0 = 0$  corresponding to an ideal conductor or a closed switch, and  $R_n = \infty$  corresponding to an open switch. It is very useful to introduce a fixed number  $n-1$  of additional finite nonzero resistance values  $R_1, R_2, \dots, R_{n-1}$  where  $R_{i+1} > R_i$ . We now have a set  $R = R_0, R_1, \dots, R_n$  of  $n+1$  resistance values from which we get a corresponding set of electric current values  $I = I_0, I_1, \dots, I_n$  via Ohm's Law

$$I_i = \frac{V}{R_i} \quad (1)$$

Here  $V$  is the (fixed) magnitude of the voltage produced by circuit's source  $S$ .

Suppose now that resistors with values defined in  $R$  are introduced into the network  $N$  of Fig. 1. Thévenin's theorem [10] implies that the circuit feeding any output  $f_j$  is equivalent to the circuit appearing in Fig. 5.  $R_j$  is the effective resistance of the source as seen by  $f_j$ . As discussed later, by suitably defining the rules for combining discrete resistances, we can ensure that  $R_j \in R$ . We can therefore characterize the signal appearing at  $f_j$  by the pair  $(V_j, I_j)$ , where  $V_j$  is the open-circuit voltage at  $f_j$ , and  $I_j = \frac{V_j}{R_j}$  is the short-circuit current at  $f_j$ . We could equally well replace  $(V_j, I_j)$  by

Type	Symbol	Behavior	
		$c = 1$	$c = 0$
Positive			
Negative			

Fig. 3. Two types of ideal switch.

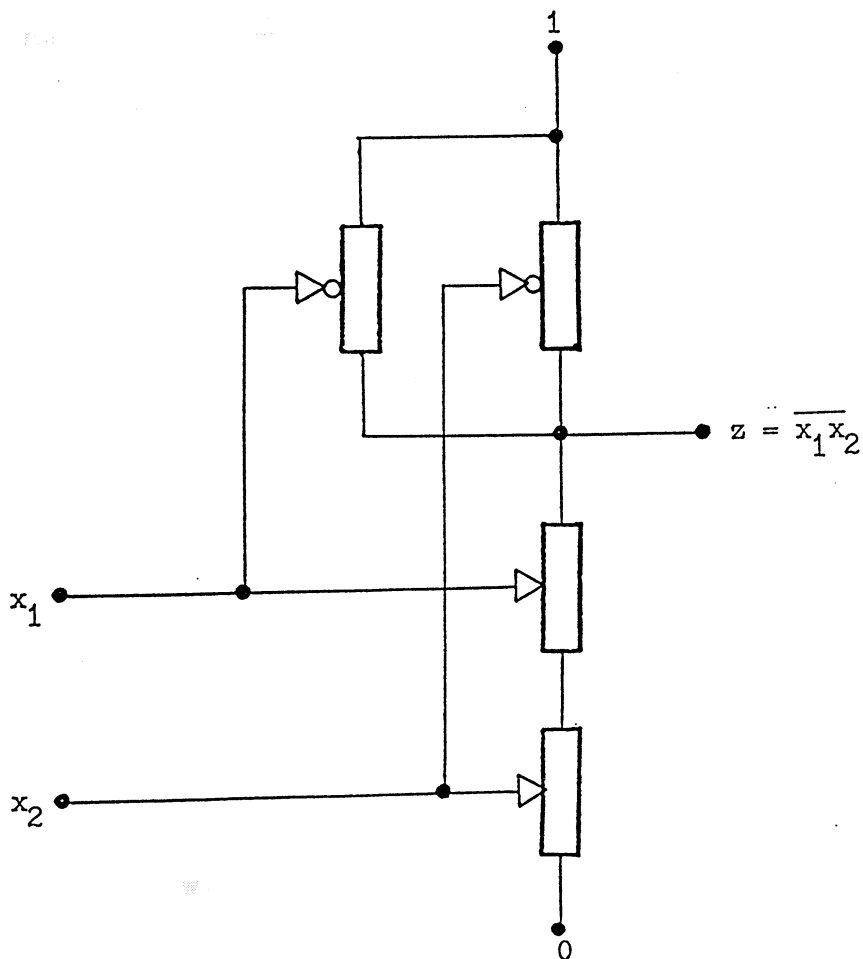


Fig. 4. Connector-switch model of a CMOS NAND gate.

$(V_j, R_j)$  or  $(V_j, G_j)$ , where  $G_j = 1/R_j$  denotes electrical conductance. It is sometimes convenient to represent each logical signal value more abstractly by a pair of the form  $(v_i, s_j)$ , where  $v_i \in \{0, 1, U, Z\} = L_i$  and  $s_j \in \{0, 1, \dots, n\} = S_n$ .  $v_i$  may be called the signal's *level*, while  $s_j$  is its *strength*. As noted above,  $s_j$  can be interpreted as current, resistance, or conductance. Here  $s_j = 0$  represents the maximum strength corresponding to  $R_0 = 0$  and  $I_0 = \infty$ , while  $s_j = n$  represents the minimum strength corresponding to  $R_n = \infty$  and  $I_n = 0$ . (Note that a slightly different strength numbering convention is used in [2,9].) The three levels 0, 1, U can be combined with all members of  $S_n - \{n\}$  to yield  $3n$  distinct logic values. All values of the form  $(v_i, n)$  and  $(Z, s_j)$  are taken to be equivalent to the high-impedance state denoted by  $(Z, n)$ , because the zero current  $I_n = 0$  can only be drawn from an open-circuited source. Hence the  $n+1$  strength levels  $S_n$  give rise to a set  $L_n$  of  $3n+1$  distinct signal values. For example, with  $n=2$  we obtain the seven values

$$L_2 = \{(0, 0), (0, 1), (1, 0), (1, 1), (U, 0), (U, 1), (Z, 2)\}$$

$(v, 0)$  may be termed the strong [5] or forcing [3] version of  $v$ , while  $(v, 1)$  is the weak or non-forcing version.

Resistor-like devices that only assume values from the discrete set  $R$  are termed *attenuators* [2] and are denoted by the symbol given in Fig. 5b. The attenuator transforms a strong signal  $(v, s)$  to a weaker signal  $(v, s')$  where  $s' > s$ ; note, however, that it does not change the logical level  $v$  of the signal. (A more precise definition of attenuator behavior is given later). Circuits like that of Fig. 4 that also contain attenuators have been called connector-switch-attenuator (CSA) circuits [2]. For example, if an attenuator  $R$  is placed in series with each switch in Fig. 4, we obtain a CMOS NAND model in which the source-drain resistance of each transistor in the on state is  $R$  rather

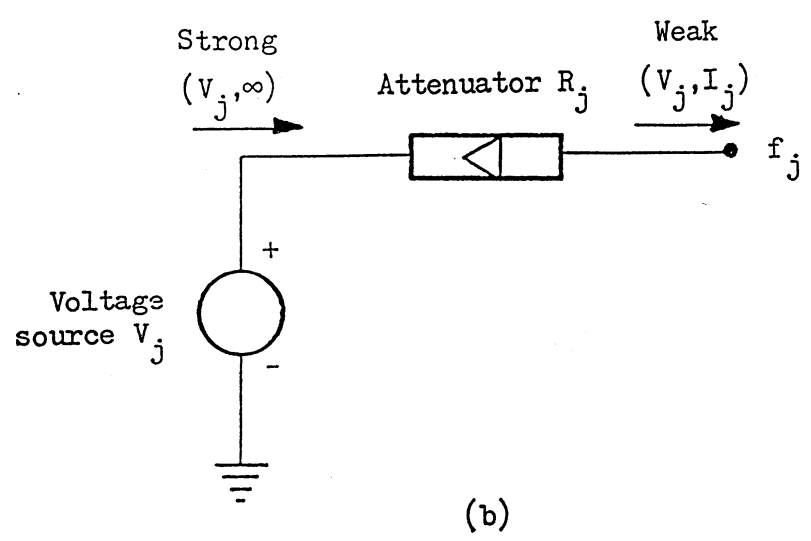
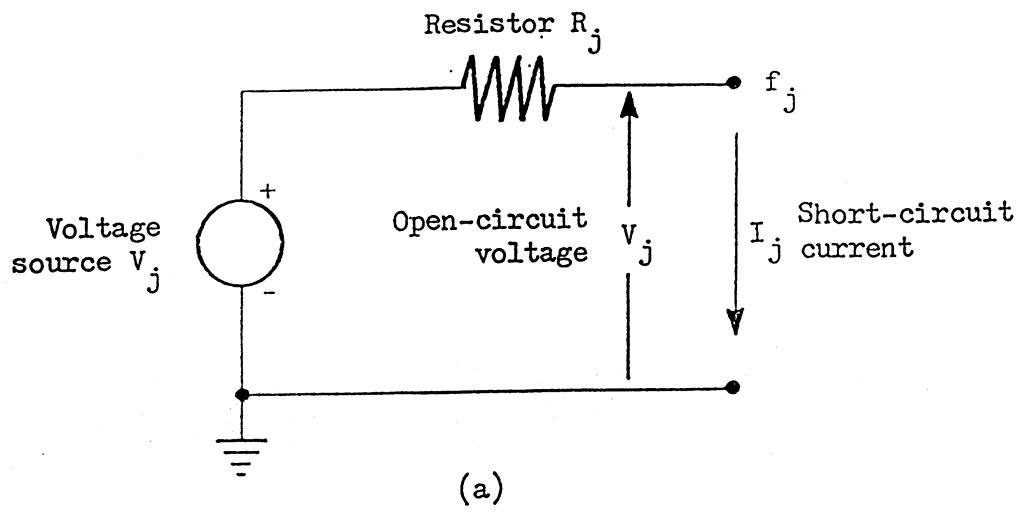


Fig. 5. (a) Thévenin equivalent of a source-resistor circuit; (b) Corresponding source-attenuator circuit.

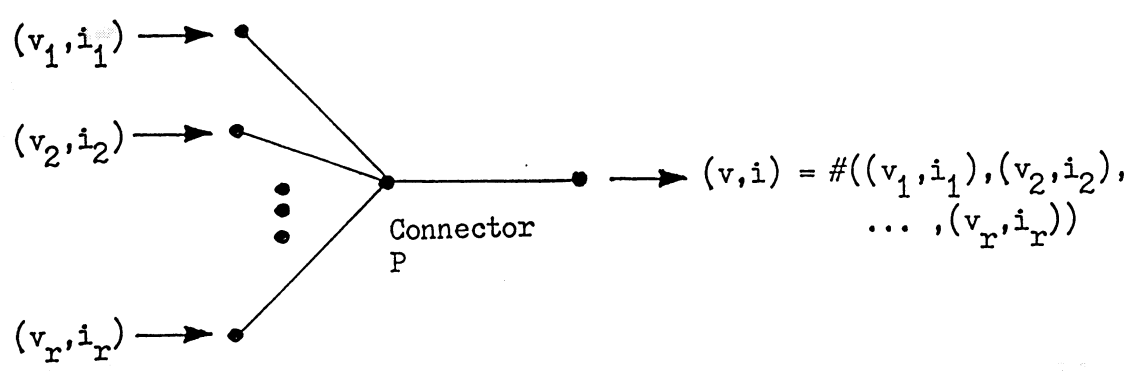


Fig. 6. Behavior of a connector with respect to  $L_n$ .

than zero. This switch-attenuator model of a transistor corresponds to the primitive switch used in MOSSIM [4].

### 3. Signal Evaluation

Suppose that  $r$  voltage-current signals  $(v_1, i_1), (v_2, i_2), \dots, (v_r, i_r)$  from  $L_n$  are applied to a connector  $P$  as in Fig. 6.  $P$  assumes a value  $S(P) = (v, i)$  termed the *state* of  $P$ , which is required to be a member of  $L_n$  also. Kirchoff's Current Law suggests that the strength (current) components of the applied signals might be summed to yield the strength of  $S(P)$  thus:

$$i = \sum_{j=1}^r i_j \quad (2)$$

Ordinary summation, however, can lead to values of  $i$  that are not in the  $(n+1)$ - member set  $I$  of available current values. We therefore replace (2) by the approximation

$$i = \underset{j=1}{\overset{r}{\text{MAX}}} \{i_j\} \quad (3)$$

which takes  $i$  to be the maximum of the applied current signals, thereby guaranteeing that  $i \in I$ . Thus the (current) strength of the output signal  $(v, i)$  is that of the strongest input signal. Equation (3) can be regarded as a discrete version of Kirchoff's Current Law appropriate to pseudo-Boolean circuits.

The voltage component  $v$  of the connector state  $S(P) = (v, i)$  in Fig. 6 may be computed as follows. Let  $W = \{v_1, v_2, \dots, v_m\}$  denote the voltage levels of all applied signals  $(v_k, i_k)$  for which  $i_k = \underset{j=1}{\overset{r}{\text{MAX}}} \{i_j\} = i$ . There are four possible cases:

- (a)  $v = U$  if  $U \in W$  or  $0, 1 \in W$ .

- (b)  $v = 0$  if  $0 \in W$  and  $1, U \notin W$ .
- (c)  $v = 1$  if  $1 \in W$  and  $0, U \notin W$ . (4)
- (d)  $v = Z$  if  $0, 1, U \notin W$ .

These rules for combining discrete voltages are consistent with the behavior of analog voltages. For example, if  $(0, i)$  and  $(1, i)$  are the only signals applied to  $P$ , we would expect  $P$  to assume an intermediate voltage level, represented here by  $U$ , that lies approximately halfway between the 0 and 1 levels.

The foregoing rules for computing  $(v, i)$  can be combined into a single operation  $\#$  by observing that  $L_n$  is a lattice. The  $\#$  operator corresponds to the lattice operator that is variously called least upper bound, join, and or. Thus for Fig. 6 we can write

$$(v, i) = \#((v_1, i_1), (v_2, i_2), \dots, (v_r, i_r))$$

The lattice structure of  $L_n$  has been recognized for some time as underlying switch-level simulators. We now demonstrate that  $L_1, L_2, \dots, L_n$  form a class of pseudo-Boolean algebras, which explains our use of the term pseudo-Boolean logic circuit.

A lattice  $L$  is a set with a partial ordering  $\leq$  such that every pair of elements  $a, b \in L$  have a least upper bound  $a \cup b = \#(a, b)$ , and a greatest lower bound  $a \cap b$ . An element  $1$  in  $L$  with the property  $a \leq 1$  for all  $a \in L$  is termed a unit element, while an element  $0$  such that  $0 \leq a$  for all  $a \in L$  is a zero element. A finite lattice, but not necessarily an infinite one, always contains unit and zero elements. An element  $c \in L$  is called the pseudo-complement of  $a$  relative to  $b$ , if  $c$  is the greatest element such that  $a \cap c \leq b$ ; following [8]  $c$  is denoted by  $a \Rightarrow b$ . A lattice  $L$  is said to be relatively pseudo-complemented if  $a \Rightarrow b$  exists for all  $a, b \in L$ . Such a lattice must contain a unit element  $1$ , since for any  $a \in L, a \Rightarrow a = 1$ . A relatively pseudo-

complemented lattice with a zero element  $0$  is called a pseudo-Boolean or Heyting algebra. For every element  $a$  in a pseudo-Boolean algebra, there is an element  $a \Rightarrow 0$ , which is called the pseudo-complement of  $a$  and is denoted by  $\bar{a}$ . A pseudo-Boolean algebra is therefore an abstract algebra  $L$  with the binary operations  $\cup$ ,  $\cap$ , and  $\Rightarrow$ , the unary operation  $\bar{\phantom{a}}$ , and the special elements  $0$  and  $1$ . This type of algebra was first defined about 1930 to characterize intuitionistic logic [8]. Boolean algebras form a special class of pseudo-Boolean algebras in which the pseudo-complement operator  $\bar{\phantom{a}}$  is replaced by a stronger complement operator  $\overline{\phantom{a}}$  (overbar). Pseudo-Boolean algebras obey the associative, commutative and distributive laws for  $\cap$  and  $\cup$ . The pseudo-complement operator obeys such familiar laws as

$$a \cap \bar{a} = 0$$

and

$$\overline{(a \cup b)} = \bar{a} \cap \bar{b}$$

However, the idempotence law  $a = \overline{\overline{a}}$  of Boolean algebra is replaced by

$$a \leq \overline{\overline{a}}$$

in pseudo-Boolean algebra. It follows directly from the foregoing definition that  $L_n$  is a pseudo-Boolean algebra with  $0 = (Z, n)$  and  $1 = (U, 0)$ . Values of the form  $(0, i)$  and  $(1, i)$  are each other's pseudo-complement relative to  $(U, i+1)$ . Only in the case  $n=1$  is  $L_n$  also a Boolean algebra.

Returning to the general pseudo-Boolean circuit of Fig. 1, we see that the value assumed by an output signal  $f_j$  with respect to  $L_n$  is determined by the effective resistance of the paths linking  $f_j$  to the source  $S$ . Figure 7 shows a representative case; compare Fig. 2.  $R_i$  and  $R_k$  model a voltage divider circuit that which is only capable of



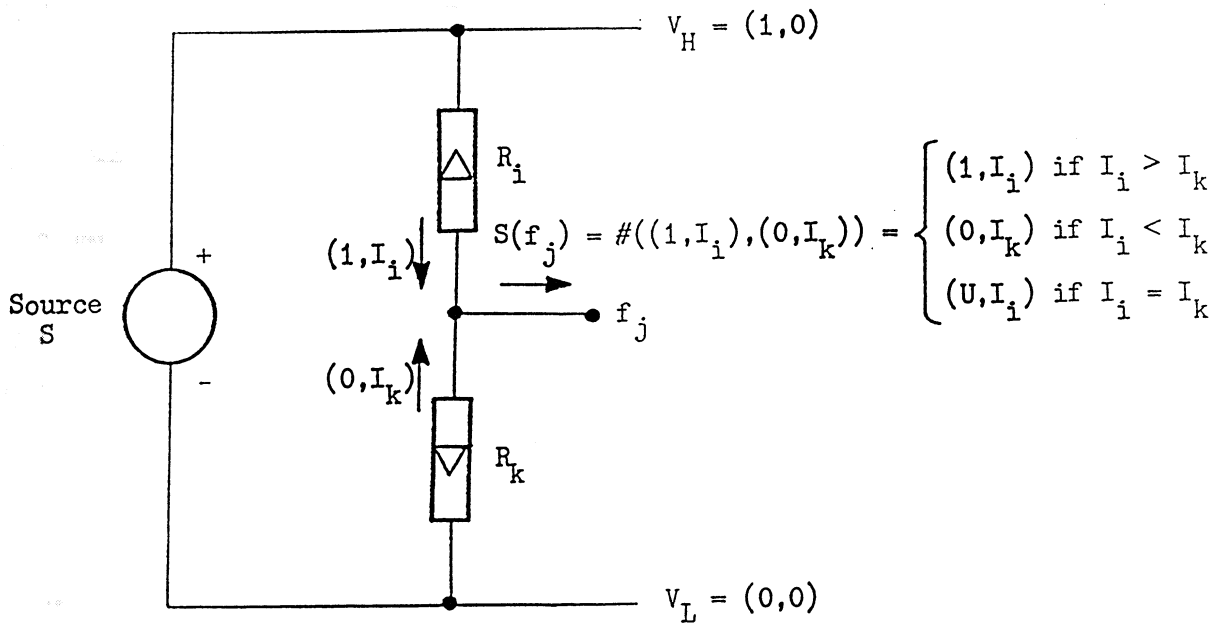


Fig. 7. Voltage-current values assumed by output  $f_j(X)$  of Fig. 1.

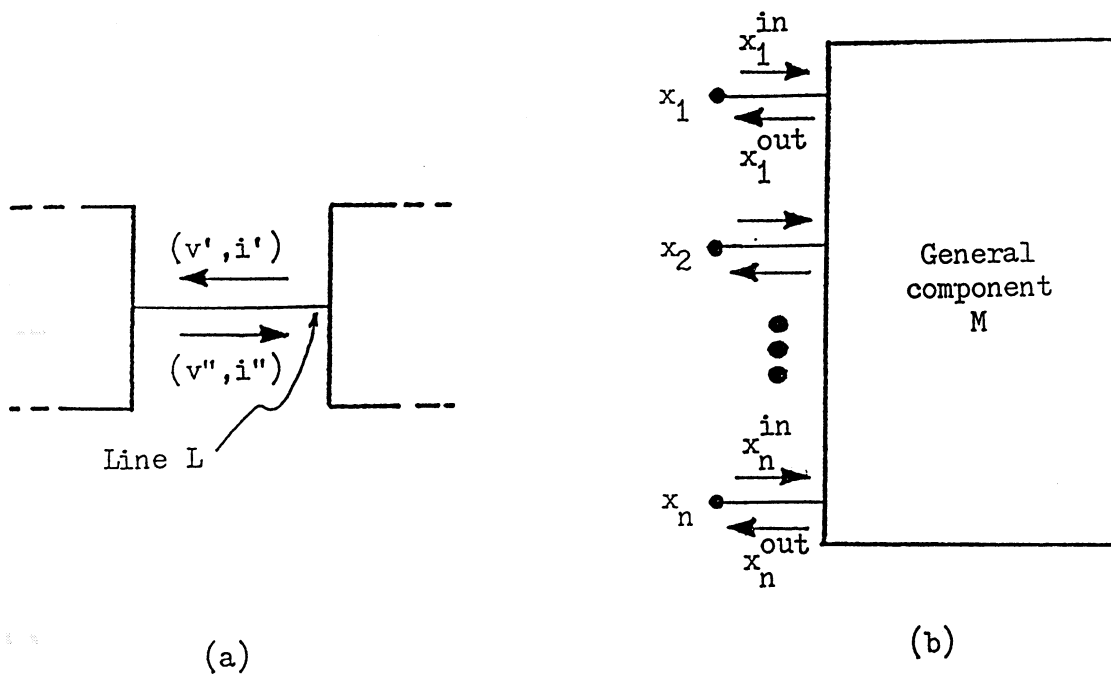


Fig. 8. Bidirectional signals associated with (a) a line  $L$ ; (b) an arbitrary component  $M$ .

assigning  $v(f_j)$  to the discrete voltage levels  $0,1,U,Z$ . (Recall that when  $I_i = I_k = 0$ ,  $(U, I_i) = (Z, n)$ .) As observed already,  $f_j$  is usually restricted to the 0 and 1 levels in logic circuits. Consequently, the attenuator circuit of Fig. 7 acts as a threshold circuit with respect to voltages, only assigning them to the 0 and 1 levels. The existence of such a strong threshold effect is characteristic of binary circuits. The addition of an unrestricted number of strength levels allows a much wider range of logic circuit behavior and structure to be modeled than is possible with conventional Boolean logic circuits.

As Fig. 7 suggests, an abstract logic signal  $(v, s)$  may be assigned a direction indicated by an arrow in much the same way as an electric current. The source  $S$  of a circuit  $N$  supplies maximum-strength signals of the form  $(1,0)$  and  $(0,0)$  from which all other signals in  $N$  are ultimately derived; these signals have a natural direction away from  $S$ . In general, a line  $L$  can have two opposing signals  $(v', s')$  and  $(v'', s'')$  associated with it, as illustrated in Fig. 8a. The resultant state  $S(L)$  of  $L$  is expressed by

$$S(L) = (v, s) = \#((v', s'), (v'', s'')) \quad (5)$$

Equation (5) defines a *Superposition Principle* analogous to that of electrical network theory, where summation of voltages and currents is replaced by the lattice operator  $\#$ .

In general, a pseudo-Boolean circuit can be constructed from any primitive components with bidirectional input-output signals defined on  $L_n$ ; see Fig. 8b. Each output signal  $x_i^{out}$  is typically a function of the input signals  $x_1^{in}, x_2^{in}, \dots, x_k^{in}$ , and may be defined by a truth table or an equation. For example, if  $M$  in Fig. 8b is a simple connector, we can define its behavior by the following set of pseudo-Boolean equations:

$$\begin{aligned}
x_1^{out} &= \#(x_2^{in}, x_3^{in}, \dots, x_n^{in}) \\
x_2^{out} &= \#(x_1^{in}, x_3^{in}, \dots, x_n^{in}) \\
&\dots \\
x_n^{out} &= \#(x_1^{in}, x_2^{in}, \dots, x_{n-1}^{in})
\end{aligned}$$

The state  $S(M)$  of the connector is given by

$$\#(x_1^{in}, x_1^{out}) = \#(x_2^{in}, x_2^{out}) = \dots = \#(x_n^{in}, x_n^{out})$$

Unidirectional lines are represented by setting either  $x_i^{in}$  or  $x_i^{out}$  to the null signal  $(Z, n)$ . Thus pseudo-Boolean circuits can be defined which represent any unidirectional components such as gates, multiplexers, etc., as well as bidirectional devices such as buses and transmission gates. Classical Boolean logic circuits, and special circuit types such as tristate logic, wired logic, and open-collector logic are all examples of pseudo-Boolean circuits.

#### 4. Component Types

With the foregoing machinery, we can define a powerful set of logic elements to model many different types of digital circuits. Figure 9 shows an assortment of connectors that transmit signals unchanged in either one or two directions. The triangle symbol can be thought of as a current amplifier (a unidirectional device) with unit gain; it can also be viewed as an arrowhead denoting signal direction. In the case of Fig. 9c, for instance, the signal  $b^{in}$  applied to  $b$  has no affect on  $a$ , since by the Superposition Principle,  $S(a) = \#(a^{in}, (Z, n)) = a^{in}$ .

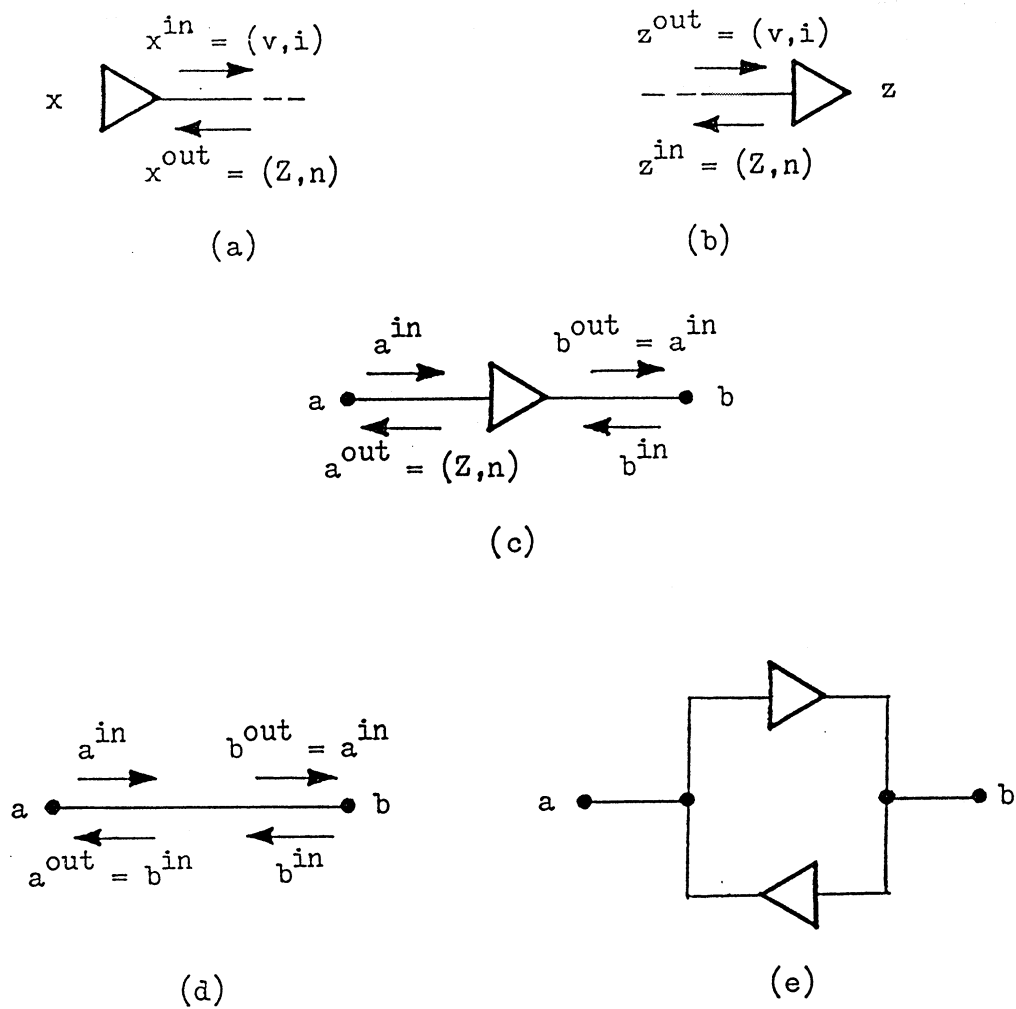


Fig. 9. Terminals and connectors: (a) input terminal; (b) output terminal; (c) unidirectional line; (d) bidirectional line; (e) equivalent circuit for a bidirectional line.

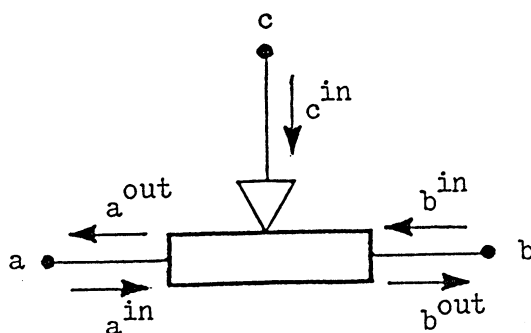


Fig. 10. A three-terminal positive switch representing a bipolar or MOS transistor.

A positive switch may be represented as shown in Fig. 10. Here the control input  $c$  is assumed to be a unidirectional input terminal, while the  $a$  and  $b$  terminals are bidirectional. The on-off state of the switch can be made any function of  $a^{in}, b^{in}, c^{in}, a^{out}, b^{out}, c^{out}$  that reflects the underlying device technology. For example, if  $c^{in} = (1, i_j)$  turns the switch on, and  $c^{in} = (0, i_j)$  turns it off for any  $i_j$ , we can specify the corresponding  $a^{out}$  and  $b^{out}$  signals in the following truth-table form, which is the usual definition of an ideal switch,

$c^{in}$	$a^{out}$	$b^{out}$	Switch State
$(1, i_j)$	$b^{in}$	$a^{in}$	On
$(0, i_j)$	$(Z, n)$	$(Z, n)$	Off

(6)

We can also define more complex behavior where the switch's state depends, say, on the relative voltage levels of the  $b$  and  $c$  terminals; this is typical of both bipolar and MOS transistors. Consider the following partial definition of switch behavior.

$S(a)$	$c^{in}$	$a^{out}$	$b^{out}$	Switch State
$(0, i_j)$	$(1, i_k)$	$b^{in}$	$a^{in}$	On
$(0, i_j)$	$(0, i_k)$	$(Z, n)$	$(Z, n)$	Off
$(1, i_j)$	$(0, i_k)$	$(Z, n)$	$(Z, n)$	Off
$(1, i_j)$	$(1, i_k)$	$(Z, n)$	$(Z, n)$	Off

(7)

Let the switch be initially off with  $S(a) = a^{in} = (0, i_j)$ ,  $b^{in} = (1, i_m)$  and  $c^{in} = (1, i_k)$ ,

where  $i_m > i_j$ . According to (7), the switch should immediately turn on, making  $S(a) = \#(a^{i_n}, b^{i_n}) = (1, i_m)$ . This new value of  $S(a)$  turns the switch off, causing  $S(a)$  to resume its original value  $(0, i_j)$ . Thus a contradictory condition is obtained corresponding to oscillation in a physical device. An even richer range of switch behavior, including sequential behavior, becomes possible when the values  $(U, i_j)$  and  $(Z, n)$  are permitted on the control input  $c$ .

As demonstrated by Fig. 5, the function of an attenuator is to reduce the strength of an input signal. It therefore resembles a current amplifier with a gain less than unity, as suggested by the reversed amplifier symbol in the attenuator symbol. While an attenuator is often used only to attenuate unidirectional signals, it most frequently represents an analog resistor, an inherently bidirectional device. The general behavior of a bidirectional attenuator, and a more appropriate symmetric circuit symbol for it are specified in Fig. 11. An attenuator is associated with a current  $I_j$ , determined by its size (resistance value)  $R_j$ , according to Eq. (1). It transmits an applied signal  $(v, i)$  leaving  $v$  unchanged, but reducing  $i$  to the lesser of  $i$  and  $I_j$ . This behavior approximates the current-reducing role of an analog resistor, while ensuring that the resulting current is confined to the discrete set  $I$ .

The relationship between electrical and pseudo-Boolean models is further illustrated by the attenuator circuits of Fig. 12. The size  $r$  of an attenuator equivalent to the series connection of  $k$  attenuators of size  $r_1, r_2, \dots, r_k$  as in Fig. 12a is given by

$$r = \underset{j=1}{\overset{k}{MAX}} \{r_j\} \quad (8)$$

corresponding to the resistor equation

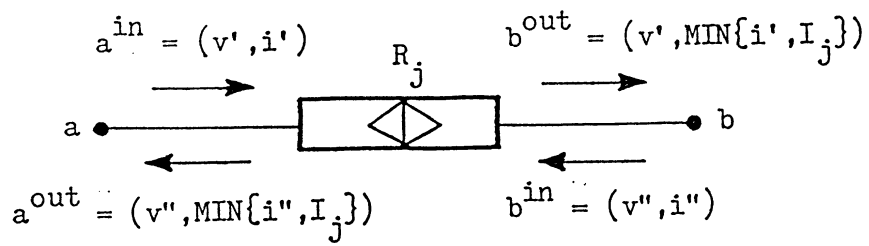
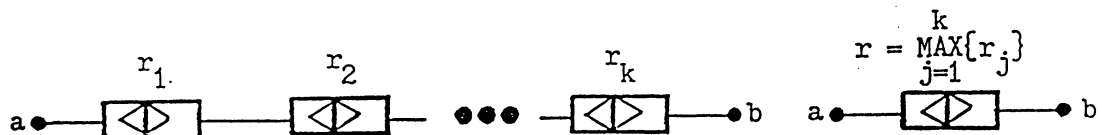
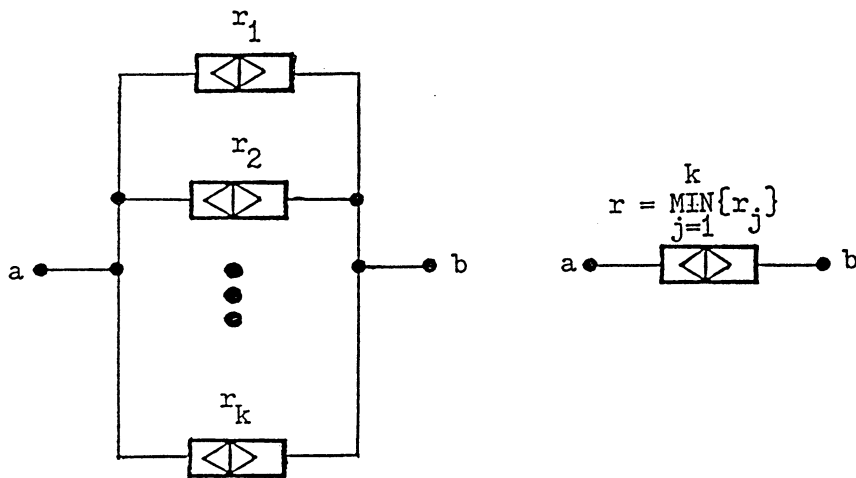


Fig. 11. A bidirectional attenuator of size  $R_j = V/I_j$ .



(a)



(b)

Fig. 12. Equivalent circuits for attenuators connected (a) in series; (b) in parallel.

$$r = \sum_{j=1}^k r_j \quad (9)$$

The correctness of (8) follows from the fact that a signal  $(v', i')$  transmitted through the chain of resistors is reduced to  $(v', i'') = (v', \text{MIN}\{i', i_1, i_2, \dots, i_k\})$  where  $i_j = \frac{V}{r_j}$ . If  $r = \text{MAX}\{r_j\}$ , then  $i = \frac{V}{r} = \text{MIN}\{i_j\}$ , hence  $(v', i'') = (v', i)$ , which is the signal produced by  $r$  alone. In a similar fashion it can be shown that the parallel connection of attenuators in Fig. 12b is equivalent to a single attenuator  $r$  defined by the equation

$$r = \text{MIN}_{j=1}^k \{r_j\}$$

This corresponds to the analog resistor equation

$$\frac{1}{r} = \sum_{j=1}^k \frac{1}{r_j}$$

Often when modeling logic circuits, a relatively small number of strength levels or, equivalently, attenuator sizes, provide a good approximation to analog behavior. MOS logic circuits, for instance, can be usefully approximated using three attenuation values specified as follows

$$R_0 = 0$$

$$R_1 = k \text{ where } 0 < k < \infty$$

$$R_2 = \infty$$

$R_1$  can be used to represent the load or pull-up device normally included in nMOS or pMOS gates. In CMOS circuits that do not contain load devices,  $R_1$  may be used to represent the non-zero source-drain resistance of a switched-on transistor. Better approximations can be obtained by introducing additional attenuator/strength values.



Suppose, for example, that we must model circuits with up to  $k$  resistors  $r_1, r_2, \dots, r_k$  connected in series as in Fig. 12a, where the resistors may assume  $m \leq k$  different values. It is characteristic of logic circuits that the relevant resistance values are far apart, i.e.,  $R_{i+1} \gg R_i$ , so that the voltages and currents they produce can be segregated into two groups as required for binary behavior. This separation of the resistance values means, for example, that in series connections like Fig. 12a, the maximum resistance dominates the others. Hence the pseudo-Boolean attenuator combination rule (8) is a good approximation to the analog resistor combination rule (9). It is easily shown that if we have only  $m$  attenuator types  $R = R_1, R_2, \dots, R_m$  defined so that  $R_{i+1} > 2kR_i$ , then for any series connection of  $k$  or fewer attenuators, Eq. (8) yields the equivalent attenuator size from  $R$  that is as close as possible to the analog value defined by Eq. (9).

Many variants of the foregoing components are possible. The unidirectional connector of Fig. 9c can be generalized to an amplifier that transforms  $a^{in} = (v, i)$  to  $b^{out} = (v, \text{MAX}\{i, I_j\})$ ; such a device is the inverse of a unidirectional attenuator. A two-terminal switch can be defined that approximates the behavior of an electronic diode. Conventional logic devices such as gates, decoders, delay elements, ROM's, etc., can also be included in pseudo-Boolean circuits, provided their behavior is defined on an appropriate subset of  $L_n$ .

## 5. Dynamic Behavior

The dynamic behavior of integrated circuits is primarily due to resistive-capacitive effects. It is therefore extremely useful to be able to include in pseudo-Boolean circuits a digital charge-storage element corresponding to an analog capacitor. Such a device,

termed a well, has been defined in [2]; similar devices are implicit in the stored-charge signal state found in some simulators [3,4].

Just as in Sec. 3 we quantized resistance into  $n+1$  discrete values  $R = (R_0, R_1, \dots, R_n)$ , we now quantize capacitance into  $p+1$  discrete values  $C = (C_0, C_1, \dots, C_p)$  where  $C_{i+1} > C_i$ . These capacitance values represent the  $p+1$  distinct well sizes that are recognized. A well  $C_i$  stores a maximum charge  $Q_i$  defined by the usual capacitor equation

$$Q_i = C_i V$$

Thus charge is also quantized into  $p+1$  discrete values. It is convenient to make  $C_0 = Q_0 = 0$  and  $C_p = Q_p = \infty$  corresponding to the capacitance of an open circuit and a conductor (short circuit), respectively.

A well  $C_j$  charges and discharges in a manner similar to a capacitor. The state of  $C_j$  can be measured by the instantaneous voltage level across it, and is confined to the value set  $L_1 = \{0, 1, U, Z\}$ . The charging and discharging process involves several discrete steps as indicated in Fig. 13, with the duration of these steps determined by the well size  $C_j$  and the size  $R_i$  of the attenuator through which charging or discharging occurs. These step sizes may be chosen to approximate the exponential charging and discharging behavior of an analog capacitor, which is indicated by broken lines in Fig. 13. Thus during charging (Fig. 13a), the voltage  $v$  across the well  $C_j$  is initially 0. At time  $t_1$  it changes to  $U$  as  $C_j$  enters a partially charged condition. Finally, at time  $t_2$ ,  $C_j$  is fully charged and  $v = 1$ .

A partially or fully charged well acts as a dynamic voltage source generating a time-varying output signal of the form  $(v(t), i(t))$ . To distinguish this from a static sig-

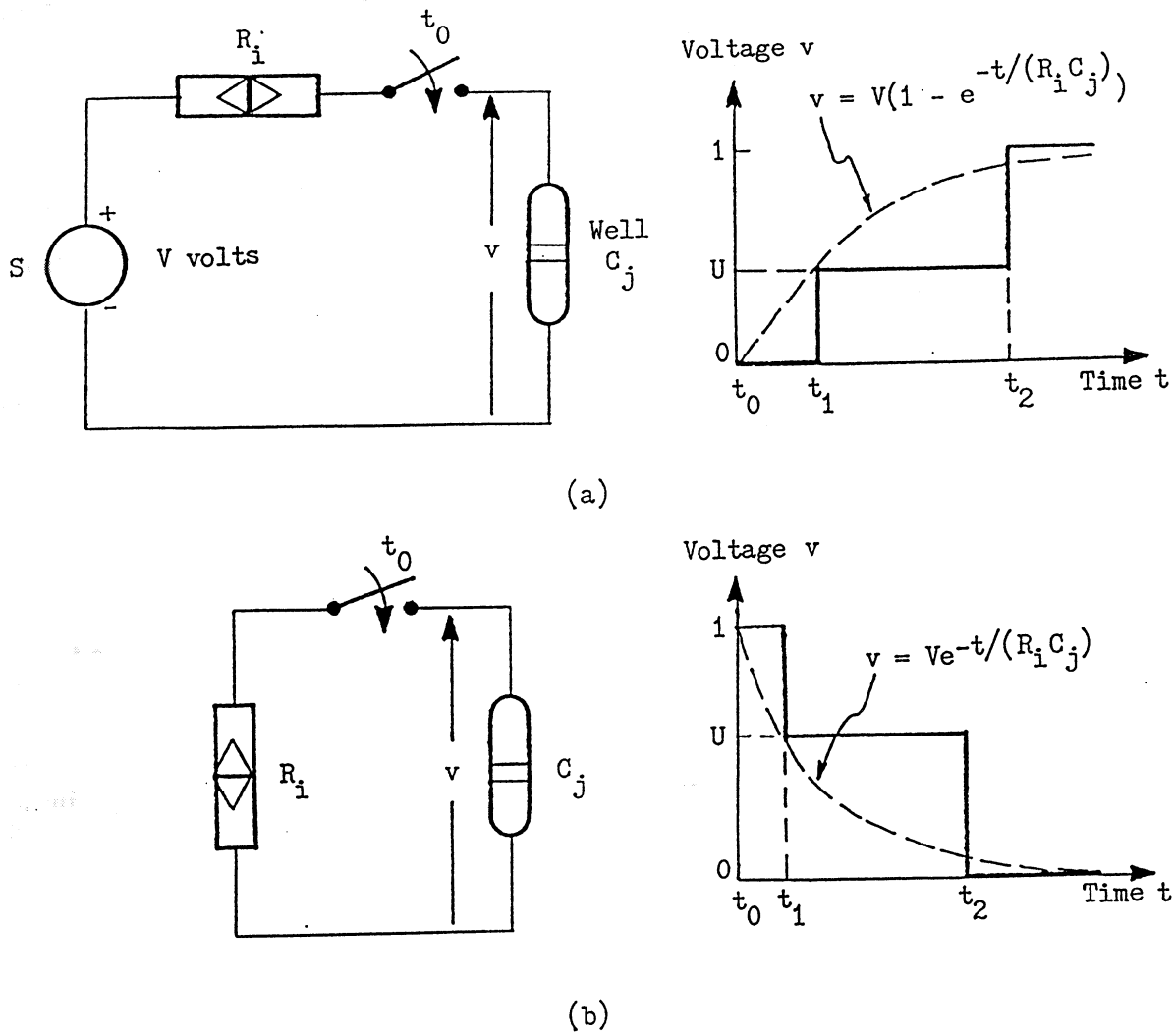


Fig. 13. Behavior of a well: (a) charging; (b) discharging.

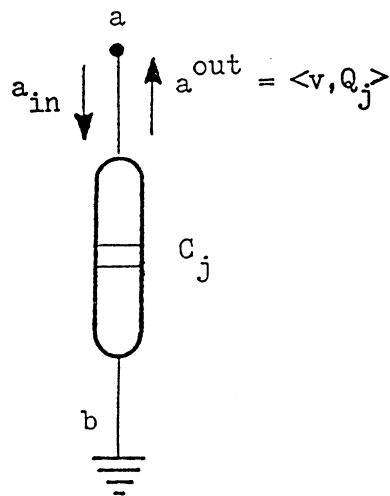


Fig. 14. Signals associated with a well.

nal  $(v, i)$ , we employ the notation  $\langle v(t), i(t) \rangle$  for dynamic signals produced by wells. The value of the signal generated by well  $C_j$  at any time can be specified by  $\langle v, C_j \rangle$  or  $\langle v, Q_i \rangle$  where  $v \in \{0, 1, U, Z\}$ . Note that  $v = U$  only when  $C_j$  is partially charged, and  $v = Z$  only when one or both terminals of  $C_j$  are disconnected, so that it is incapable of charging or discharging.

Because wells behave like voltage sources, the problem of creating new undefined voltage values discussed in Sec. 2 occurs if capacitors can be interconnected in arbitrary fashion. For instance,  $k$  fully-charged wells connected in series produce an effective voltage of  $kV$ . As in Sec. 2, we eliminate this possibility by requiring all wells and the static voltage source to share a common terminal, namely the circuit ground. Note that it is not necessary to restrict the number of wells appearing in a pseudo-Boolean circuit; their sizes are restricted to the set  $C$ . With these assumptions, every well  $C_j \in C$  in a pseudo-Boolean circuit has the general behavior shown in Fig. 14. The grounded terminal  $b$  has the strongest static 0 signal, namely  $(0, 0)$ , applied to it, and so is permanently in the  $(0, 0)$  state. The remaining terminal outputs a dynamic signal  $a^{out}$  which represents the state of the well. The externally applied input signal  $a^{in}$  may be static or dynamic, or may result from a combination of static and dynamic signals.

First, consider the behavior of well  $C_j$  in Figs. 13 and 14 when  $a^{in}$  is static. At time  $t_0$  let  $a^{in} = (d_1, i)$  and  $a^{out} = \langle d_2, j \rangle$  where  $d_1, d_2 \in \{0, 1, U\}$ . After some time  $\tau(i, j)$ , whose value depends on the strengths of  $a^{in}$  and  $a^{out}$ , i.e., on the associated  $R, C_j$  time constant,  $S(a)$  and  $a^{out}$  become  $(d_1, i)$  and  $\langle d_1, i \rangle$ , respectively. Thus the static  $a^{in}$  overrides the dynamic  $a^{out}$  after a delay  $\tau(i, j)$ . We can therefore specify the sequential behavior of  $S(a) = a$  as follows:

$$a(t + \tau(i, j)) = \#((d_1, i), \langle d_1, i \rangle) \quad (10)$$

In this way discrete delays are introduced into pseudo-Boolean circuits as a consequence of the interaction between attenuators and wells, i.e., between resistance and capacitance.

If  $a^{in} = (Z, n)$  in Fig. 14, then  $a^{out}$  and  $S(a)$  remain indefinitely at  $\langle d_2, j \rangle$ . Now suppose that  $a^{in} = \langle d_1, i \rangle$ , which corresponds to connecting the  $a$ -terminal of a second well  $C_i$  to  $C_j$ . We must model the resultant charging and discharging behavior while limiting the well states to the prescribed set of dynamic values. Just as large attenuators override the effects of smaller ones, we allow the larger well to override the smaller one. Hence if  $C_j > C_i$ , and  $a^{in} = \langle d_1, i \rangle$  is applied by  $C_i$  to  $C_j$  (see Fig. 14), then  $S(a)$  and  $a^{in}$  both eventually change to  $a^{out} = \langle d_2, j \rangle$ . For example, if  $a^{in} = \langle 1, i \rangle$  and  $a^{out} = \langle 0, j \rangle$ , then  $C_j$  effectively discharges  $C_i$  without itself becoming charged. Similarly if  $a^{out} = \langle 1, j \rangle$ ,  $C_j$  charges  $C_i$ , with the size difference between  $C_i$  and  $C_j$  ensuring that  $C_j$  remains fully charged. In the case where  $C_i = C_j$ , charge sharing occurs between  $C_i$  and  $C_j$  when their states differ. Thus if  $a^{in} = \langle 0, i \rangle = \langle 0, j \rangle$  and  $a^{out} = \langle 1, j \rangle$ , then  $a^{in}$  and  $a^{out}$  both become  $\langle U, i \rangle$ .

The foregoing analysis implies that with  $p+1$  dynamic strength levels, i.e.,  $p+1$  well sizes, there are  $3p+1$  distinct dynamic logic values of the form  $\langle v, q \rangle$  which form a pseudo-Boolean algebra  $\hat{L}_p$  isomorphic to  $L_p$ , the algebra of static values  $(v, i)$  with  $p+1$  attenuator sizes. It is readily seen that when  $p=1$ ,  $\hat{L}_1$  and  $L_1$  coincide, so that there is no distinction between static and dynamic values. The zero elements  $\langle Z, p \rangle$  and  $(Z, n)$  of  $\hat{L}_p$  and  $L_n$ , respectively are also identical, since each represents the signal produced by an open circuit, i.e., the high-impedance state. If  $n+1$  attenuator sizes and  $p+1$  well sizes are permitted in a pseudo-Boolean circuit, then the corresponding algebras  $L_n$  and

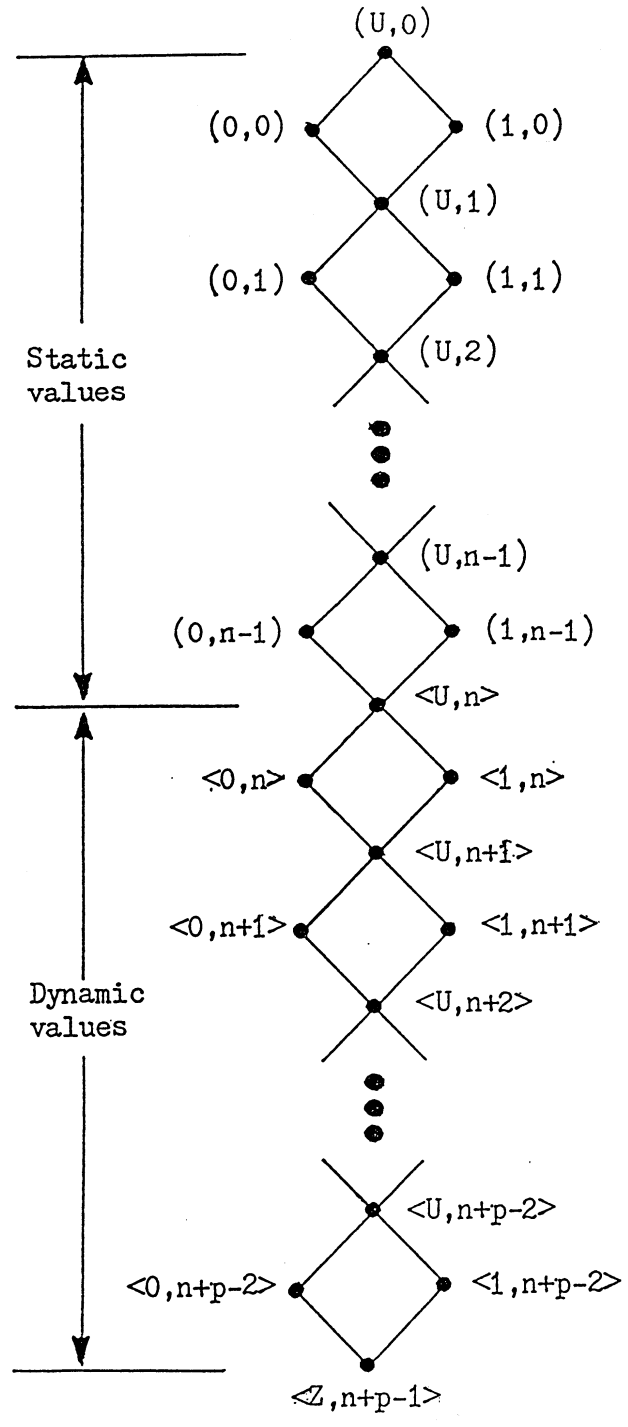


Fig. 15. The lattice structure of the pseudo-Boolean algebra  $L_{n,p}$ .

$\hat{L}_p$  are merged to form a single  $[3(n+p-1)-1]$ -member pseudo-Boolean algebra whose structure is illustrated in Fig. 15. This algebra, denoted  $L_{n,p}$ , is isomorphic to  $L_{n+p-1}$  and may be formed by concatenating the dynamic algebra  $\hat{L}_{p-1}$  to the static algebra  $L_n$ . This concatenation reflects the fact that any dynamic signal can eventually be overridden by any static signal. Using the dagger symbol of [11] for lattice concatenation we can write

$$L_{n,p} = L_n \dagger \hat{L}_{p-1}$$

for  $n \geq 1$  and  $p > 1$ ; when  $p = 1$ ,  $L_{n,p}$  reduces to  $L_1 = \hat{L}_1$ . In general,  $L_{n,p}$  is used to analyze the behavior of pseudo-Boolean circuits in which there are up to  $n-1$  finite nonzero attenuator sizes and  $p-1$  finite nonzero well sizes. In Fig. 15 the strength levels of  $\hat{L}_{p-1}$  are numbered  $n, n+1, \dots, n+p-1$ , to emphasize that they are weaker than the strength values appearing in  $L_n$ .

All interactions between signals applied to any connector of a pseudo-Boolean circuit  $N$  are determined by the least upper bound operator  $\#$  for  $L_{n,p}$ . When both static and dynamic signals are involved, signal transitions are delayed by an appropriate amount as in Eq. (10). For example, suppose that a connector  $a$  has a stable dynamic signal  $\langle 1, i \rangle$  applied to it at time  $t$ , so that the connector state  $a(t)$  is  $\langle 1, i \rangle$ . If the static signal  $(0, j)$  is now applied to  $a$ , it remains in state  $\langle 1, i \rangle$  for a period  $\tau_F$  and then changes to  $(0, j)$  thus:

$$\begin{aligned} a(t) &= \langle 1, i \rangle \\ a(t+\tau_F) &= \#(\langle 1, i \rangle, (0, j)) = (0, j) \end{aligned}$$

$\tau_F$  can be viewed as the signal (voltage) fall time, and is a function of the signal strength levels  $i$  and  $j$ . Similarly, a rise time  $\tau_R$ , not necessarily equal to  $\tau_F$ , can be associated

with the interaction of  $\langle 0, i \rangle$  and  $\langle 1, j \rangle$ . The time delays derived from the interaction of static and dynamic signals in this manner are more accurate and more natural than those obtained using unidirectional lumped delay elements of the kind employed by most logic simulators. Of course, lumped delays can also be included in pseudo-Boolean circuits, if desired.

## 6. Applications

Pseudo-Boolean circuits provide a unified framework for the analysis of many types of logic circuits [2]. Their practical significance lies in their ability to model the behavior and, to a lesser extent, the layout structure of MOS circuits of the kind used in VLSI design. In particular, they constitute the theoretical basis for switch-level simulation programs, which have become widely used since the early 1980's for VLSI design verification. These simulators can perform more detailed behavioral analysis than traditional gate-level simulators, using component and signal types that better reflect the physical properties of real circuits. By varying the number of strength levels recognized, i.e., the number of distinct resistance and capacitance sizes, useful tradeoffs between simulation accuracy and computational complexity can be made.

Figure 16 shows several pseudo-Boolean approximations to an MOS switching transistor which is the basic building block of VLSI circuits. The simplest model is the ideal switch of Fig. 16b, whose behavior is defined in (6) and Fig. 3. The input line  $c$  corresponding to the transistor's gate terminal, controls the path between the switch's bidirectional  $a$  and  $b$  terminals. The behavior of circuits such as that of Fig. 4, which are composed solely of ideal switches, can be defined using the four-valued signal set  $L_1$ . Figure 16c shows a better transistor approximation in which the source-drain resistance



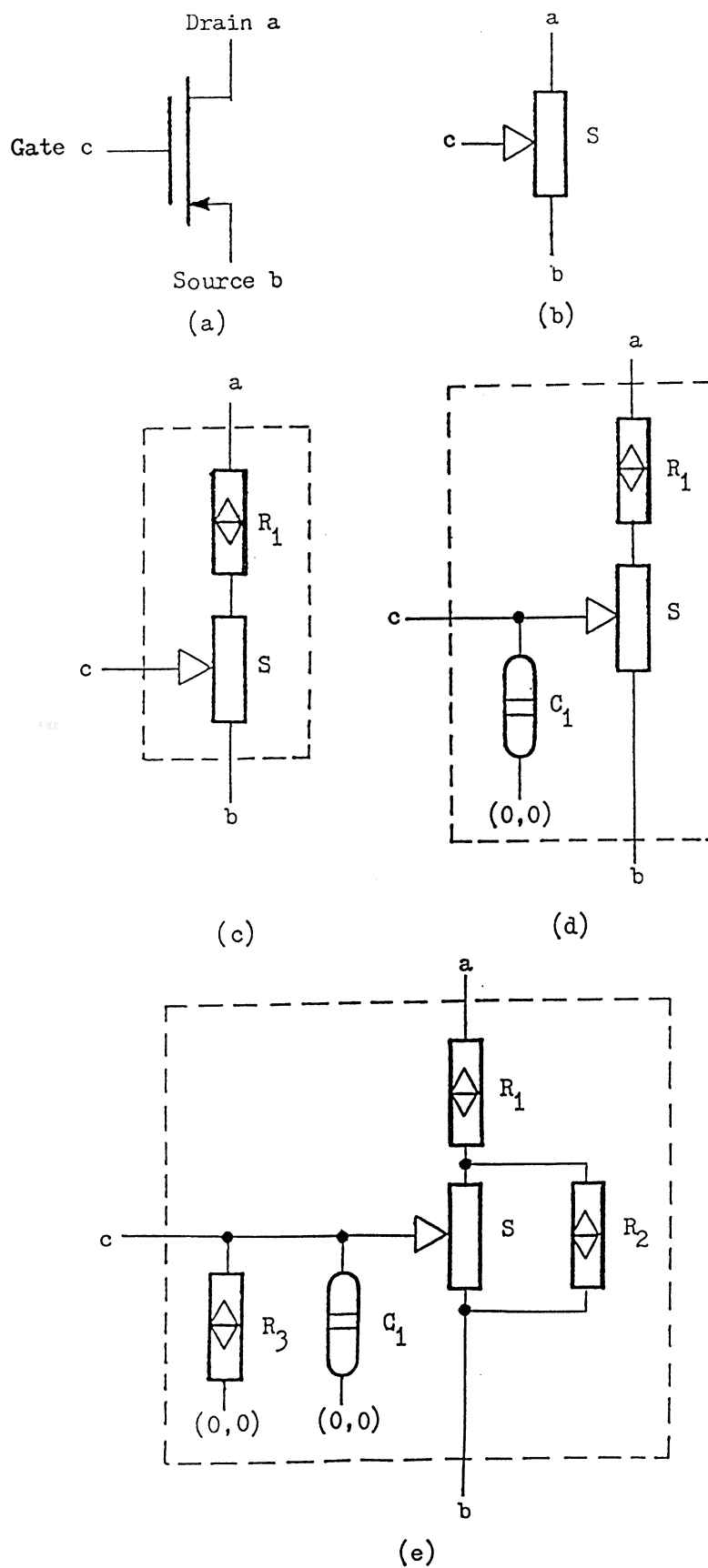


Fig. 16. (a) An nMOS switching transistor; (b-e) various pseudo-Boolean models of this transistor.

in the on-state is represented by the attenuator  $R_1$ . In this case  $L_2$  is the appropriate pseudo-Boolean algebra for defining transistor behavior. Note that we can replace the circuit of Fig. 16b by a primitive switch that has the following behavior, where  $a^{in} = (v_a, i_a)$  and  $b^{in} = (v_b, i_b)$ ; compare (6),

$c^{in}$	$a^{out}$	$b^{out}$	Switch State
$(1, i_j)$	$(v_b, MIN \{i_b, I_1\})$	$(v_a, MIN \{i_a, I_1\})$	On
$(0, i_j)$	$(Z, n)$	$(Z, n)$	Off

Figure 16d adds a well  $C_1$  to the preceding model to represent the gate-substrate capacitance of the transistor; this circuit may be analyzed using  $L_{2,2} = L_2 \dagger \hat{L}_1$ . Note that if  $c = (Z, n)$ , i.e., the transistor's gate terminal is open-circuited, the well can hold the switch  $S$  in its previous on or off state indefinitely. Much more complex transistor models are possible, but require increasingly larger pseudo-Boolean algebras to describe them. The circuit of Figure 16e, for instance, includes two additional attenuator sizes  $R_2$  and  $R_3$ , representing the off (source-drain) resistance and input (gate-source) resistance of the transistor, respectively. The algebra  $L_4 \dagger L_1$  having 16 values would be necessary to cover this case.

An inverter stage from a two-phase nMOS dynamic shift-register [12] is depicted in Fig. 17a.  $Q_1$ ,  $Q_2$  and  $Q_4$  are switching transistors, while  $Q_3$  is a clocked load transistor.  $C_1$  denotes the gate-substrate capacitance of  $Q_2$ , which plays a key role as a temporary data storage device in dynamic circuit operation. An equivalent pseudo-Boolean circuit for the inverter stage appears in Fig. 17b. Here the switch-attenuator transistor model of Fig. 16c represents  $Q_1$ ,  $Q_3$  and  $Q_4$ , while the switch-attenuator-well model of Fig. 16d represents  $Q_2$ . As observed earlier, these transistor models, which are enclosed in broken

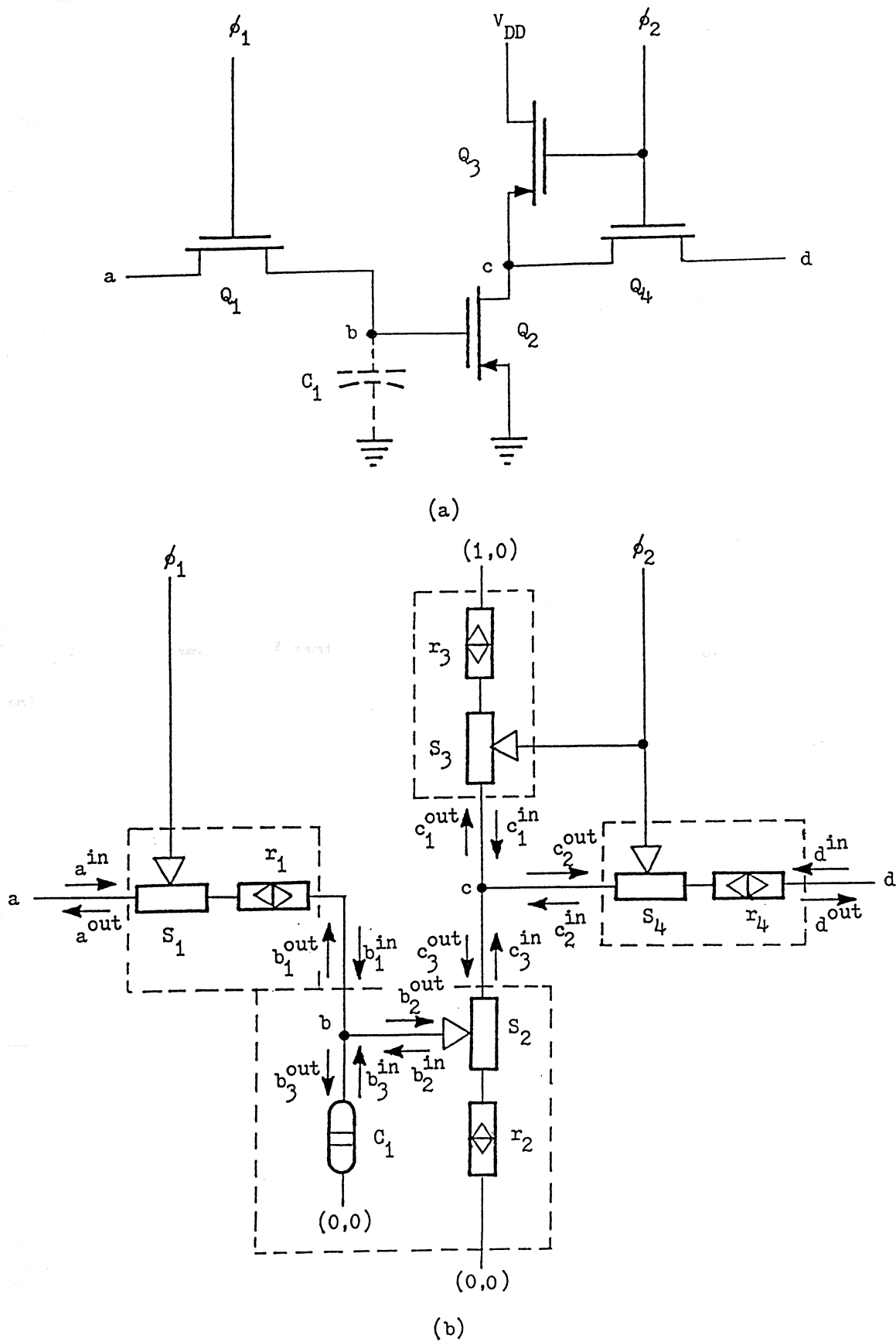


Fig. 17. (a) Inverter stage of a dynamic shift register; (b) pseudo-Boolean model.

lines in Fig. 17b, can be defined as primitive elements, thus preserving a one-to-one correspondence between the structure of the pseudo-Boolean circuit model and that of the electrical circuit (Fig. 17a). The attenuators  $r_1$ ,  $r_2$  and  $r_4$  can all be assigned a relatively small resistance value  $R_1$ , while attenuator  $r_3$ , representing a load element is assigned a much larger value  $R_2$ . Thus the circuit is characterized by the parameters  $R = \{R_0 = 0, R_1, R_2, R_3 = \infty\}$  and  $C = \{C_0 = 0, C_1, C_2 = \infty\}$ , and the 13-member pseudo-Boolean algebra  $L_{3,2} = L_3 \uparrow \hat{L}_1$ . A weaker, but nevertheless useful, approximation to this circuit can be obtained by setting  $r_1 = r_2 = r_4 = 0$ , in which case the ten values of  $L_{2,2} = L_2 \uparrow \hat{L}_1$  suffice to describe its behavior.

We now briefly illustrate the analysis of pseudo-Boolean circuits using Fig. 17b as an example. We are primarily interested in determining the states of the connectors  $a$ ,  $b$ ,  $c$ , and  $d$  with respect to  $L_{3,2}$ ; the other connectors have either constant values ( $V_{DD}$  and ground) or externally controlled clock signals ( $\phi_1$  and  $\phi_2$ ) applied to them. Suppose that the circuit is in the initial state

$$(\phi_1, \phi_2, a, b, c, d) = ((0, 0), (0, 0), (1, 0), \langle 0, 3 \rangle, \langle Z, 4 \rangle, \langle Z, 4 \rangle)$$

whose strength levels are defined as in Fig. 15. Assume that well  $C_1$  is discharged, and all clock signals and switches are initially in the off state. Node  $b$  is held at the weakest 0 value  $\langle 0, 3 \rangle$  by the well, while  $c$  and  $d$  are disconnected from all signal sources and thus assume the high-impedance state  $\langle Z, 4 \rangle$ .

Now suppose that  $\phi_1$  changes from  $(0, 0)$  to  $(1, 0)$ . This turns switch  $S_1$  on, thereby applying  $a^{in} = (1, 0)$  to attenuator  $r_1$ . This in turn causes  $b_1^{in}$  to change from  $\langle Z, 4 \rangle$  to  $(1, 1)$ , and the well  $C_1$  begins to charge. After some time  $\tau(R_1, C_1)$ ,  $b$  reaches the 1 level causing switch  $S_2$  to turn on, and immediately changing  $c$  from  $\langle Z, 4 \rangle$  to  $(0, 1)$ .

Since  $S_4$  is still switched off,  $d$  remains unchanged. If  $\phi_1$  now returns to  $(0, 0)$ ,  $C_1$  holds  $S_2$  in the on state, and the signal of interest have the following states.

$$(\phi_1, \phi_2, a, b, c, d) = ((0, 0), (0, 0), (1, 0), \langle 1, 3 \rangle, (0, 1), \langle Z, 4 \rangle)$$

Next suppose that  $\phi_2$  becomes  $(1, 0)$ . This causes both  $S_3$  and  $S_4$  to switch on.  $c_1^{in}$  changes from  $\langle Z, 4 \rangle$  to  $(1, 2)$ , and  $c$  assumes the state  $\#((1, 2), (0, 1)) = (0, 1)$ . This state is applied via  $c_2^{out}$  and  $r_4$  to  $d$  causing  $d^{out}$  to become  $(0, 1)$ . Thus the circuit is now in the state

$$(\phi_1, \phi_2, a, b, c, d) = ((0, 0), (1, 0), (1, 0), \langle 1, 3 \rangle, (0, 1), (0, 1))$$

Most existing switch-level simulators implicitly use subsets of  $L_{n, p}$  corresponding to very small values of  $n$  and  $p$ . The 10-valued pseudo-Boolean algebra  $L_{2, 2} = L_2 \uparrow \hat{L}_1$  is explicitly used by the fault simulator CSASIM [5]. CSASIM recognizes one finite non-zero attenuator size  $R_1$  and one finite non-zero well size  $C_1$ . In nMOS or pMOS circuit analysis, for example,  $R_1$  is typically equated to the resistance of the load transistor of a logic gate;  $C_1$  is equated to the gate-substrate capacitance of a switching transistor. Essentially the same pseudo-Boolean algebra is used in LOGIS [3], one of the first commercial simulators with comprehensive switch-level simulation capabilities. LOGIS does not explicitly identify the zero element  $\langle Z, 3 \rangle$ , corresponding to the high-impedance state  $Z$ , as a logic value. Hence, it is described as a 9-valued simulator, whose values consist of all combinations of the three voltage levels  $\{0, 1, U\}$  and the three strength levels  $\{0, 1, 2\}$ . In LOGIS an attenuator is termed a "resistive gate." LOGIS has no explicit digital storage device corresponding to a well; instead a circuit node  $P$  is allowed to be in a "trapped charge" state, which is tantamount to connecting a well of fixed size  $C_1$  between  $P$  and ground. The following table compares the ways in which the logic values

of  $L_{2,2}$  are interpreted by CSASIM and LOGIS.

Value	CSASIM	LOGIS
(0,0)	Strong static 0	Forcing low (unlimited charge sink)
(1,0)	Strong static 1	Forcing high (unlimited charge source)
(U,0)	Strong static U	Forcing indeterminate (short-circuit)
(0,1)	Weak static 0	Non-forcing low (restricted charge sink)
(1,1)	Weak static 1	Non-forcing high (restricted charge source)
(U,1)	Weak static U	Non-forcing indeterminate (unknown charge)
<0,2>	Dynamic 0	High-Z low (minimum trapped charge)
<1,2>	Dynamic 1	High-Z low (maximum trapped charge)
<U,2>	Dynamic U	High-Z indeterminate (unknown trapped charge)
<Z,3>	High-impedance state	

A second version of the CSASIM simulator in which  $L_{n,p}$  is the logic value set, and  $n$  and  $p$  are treated as user-selectable parameters, is currently under development at the University of Michigan.

## 7. Discussion

A new class of logic circuits that accurately represent the structure and behavior of many important types of digital circuits has been presented. It has been shown that the mathematical structure underlying these circuits is a type of pseudo-Boolean algebra. The circuits in question can therefore be seen as a generalization of gate-type or contact-type logic circuits based on Boolean algebra. At the same time, pseudo-Boolean circuits are discrete approximations to electrical circuits, where parameters like voltage, current, charge, resistance, and capacitance are confined to finite closed sets. We have demonstrated that basic electrical properties of signals such as bidirectionality, Kirchoff's Current Law, and the Superposition Principle, can be rigorously mapped from

the electrical to the logical plane. Thus it can be concluded that pseudo-Boolean circuits occupy a well-defined complexity level (the switch level) lying between the classical electrical and logic circuit levels.

A variety of approaches to the analysis of switch-level circuits have been proposed, including graph-theoretical methods [4,7] and characteristic functions [6,13]. In this paper an alternative technique based on the superposition of bidirectional signals has been described. It has been implemented explicitly in the simulation program CSASIM [5], and appears to be especially useful for simulating both the normal and the faulty behavior of complex MOS circuits.

#### **Acknowledgement**

The contributions of Masato Kawai to the development of the CSASIM simulator are gratefully acknowledged.

## References

- [1] C. Mead and L. Conway: *Introduction to VLSI Systems*, Reading, Mass., Addison-Wesley, 1980.
- [2] J.P. Hayes: "A unified switching theory with applications to VLSI design." *Proc. IEEE*, vol. 70, pp. 1140-1151, Oct. 1982.
- [3] Information Systems Design, Inc.: *Logis User's Manual*, Santa Clara, circa 1978.
- [4] R.E. Bryant: "MOSSIM: a switch-level simulator for MOS LSI," *Proc. 18th Design Automation Conf.*, Nashville, pp. 786-790, June 1981.
- [5] M. Kawai and J.P. Hayes: "An experimental MOS fault simulation program CSASIM," *Proc. 21st Design Automation Conf.*, Albuquerque, pp. 2-9, June 1984.
- [6] J.P. Hayes: "A logic design theory for VLSI," *Proc. Second Caltech Conf. on VLSI*, Pasadena, pp. 455-476, Jan. 1981.
- [7] R.E. Bryant: "A switch-level model and simulator for MOS digital systems," *IEEE Trans. Computers*, vol. C-33, pp. 160-177, Feb. 1984.
- [8] H. Rasiowa and R. Sikorski: *The Mathematics of Metamathematics*, Polska Akad. Nauk Monografie Math., vol. 41, Warsaw, 1963.
- [9] J.P. Hayes: "Fault modeling for digital MOS integrated circuits," *IEEE Trans. Computer-Aided Design*, vol. CAD-3, pp. 200-207, July 1984.
- [10] G.H. Hostetter: *Engineering Network Analysis*, New York, Harper and Row, 1984.
- [11] R. Balbes and P. Dwinger: *Distributive Lattices*, Columbia, Mo., Univ. of Missouri Press, 1974.
- [12] W.N. Carr and J.P. Mize: *MOS/LSI Design and Application*, New York, McGraw-Hill, 1972.
- [13] E. Cerny and J. Gecsei: "Functional description of connector-attenuator-switch networks," Université de Montréal, Dépt. d'Informatique et de Recherche Operationelle, Pub. No. 479, June 1983.