

A SWITCH-LEVEL MODEL OF MOS LOGIC CIRCUITS

Randal E. Bryant

*Department of Computer Science,
California Institute of Technology, Pasadena,
CA 91125, USA*

INTRODUCTION

The study of mathematical models of logic circuits in recent times has been limited primarily to the Boolean logic gate model, in which a system consists of a set of unidirectional logic elements (gates) connected by one-way, memoryless wires. In contrast to this restricted model, designers of MOS LSI systems have a rich variety of circuit design techniques at their disposal. Combinational logic can be implemented with logic gates, steering logic and PLA's. Data can be stored in static and dynamic memory, communicated along wires and busses, and directed through pass transistors. Each of these techniques has numerous variations, and hence the designer can tailor a system design according to speed, density and architectural needs. The Boolean gate model lacks this richness, because it fails to reflect the basic structure of MOS systems in which the logic elements (i.e. the field-effect transistors) are bidirectional, and the wires (including the attached transistor gates) have sufficient capacitance to store information. As a consequence, computerized tools and analytic techniques based on the Boolean gate model provide limited assistance for the MOS designer. Many programs such as logic simulators extend the Boolean gate model with special logic elements and additional logic states, but these programs lack generality and accuracy as well as any formal mathematical basis.

In this paper a new logic model will be presented which more closely matches MOS circuit technology and hence can describe the logical behavior of a wide variety of MOS logic circuits in a very direct way. In this *switch-level* model a network consists of a set of nodes connected by transistor "switches", where each node has a state 0, 1 or X (for unknown

or undefined), and each transistor has a state open, closed, or unknown. Transistors have no assigned direction of information flow and can be assigned different strengths to model their behavior in ratioed circuits. Nodes retain their states indefinitely in the absence of applied inputs, giving an idealized model of dynamic memory. Nodes can be assigned different sizes to model the effects of charge sharing in ratioless circuits. The switch-level model differs greatly from both Boolean logic gate and relay models in the way logic states are formed. In keeping with the concept of a *logic* model, however, both the transistor strengths and the node sizes may take on only discrete values, and the electrical operation of a circuit is modeled in a highly idealized way.

This model provides a formal bases for switch-level simulation programs such as the author's MOSSIM (Bryant, 1980, 1981b). These simulators have demonstrated the advantage of a switch-level logic model. They can accurately simulate a wide variety of MOS designs at speeds approaching those of logic gate simulators. Furthermore, since the simulation network corresponds closely to the electrical network, it can be derived from a specification of the mask patterns by a relatively straightforward computer program such as the one described by Baker and Terman (1980). The development of a mathematical model of switch-level networks has led to a simulation algorithm which improves on the previous ones in its generality, accuracy and simplicity. This paper describes material presented in greater detail and with more rigor in Bryant (1981b).

NETWORK MODEL

A switch-level network consists of a set of nodes connected by transistor switches. The nodes are of two types: *input* nodes, labeled i_1, \dots, i_m , and *normal* nodes, labeled n_1, \dots, n_n . Input nodes provide strong signals to the system and are not affected by the actions of the network, much like voltage sources in electrical networks. Examples include the power and ground nodes Vdd and Gnd, as well as all connections to the chip through input pads. Normal nodes have states determined by the operation of the network, and these states are stored dynamically much as the storage of charge in capacitors. Each normal node is assigned a *size* from the set $K = \{\kappa_1, \dots, \kappa_q\}$ to indicate its approximate capacitance relative to other nodes with which it may share charge. The elements of K are totally ordered from κ_1 to κ_q . These node sizes allow a simplified model of charge sharing in ratioless circuits in which the states of the largest node(s) dominate

when a set of nodes is connected by turned-on transistors. Figure 1 shows a switch-level model of a three transistor dynamic RAM circuit in which the bus node has size κ_2 to indicate that it can supply its state to the storage node of the selected bit position during a write operation and to the drain node of the storage transistor during a read operation. Most MOS designs can be modeled with just two different node sizes ($q=2$).

Each normal node n_j has a state $y_j \in \{0,1,X\}$. The states 0 and 1 correspond to the normal Boolean logic states, while the state X indicates that the node has not been properly initialized or that its voltage may lie between the logic thresholds due to either a short circuit or improper charge sharing. Each input node i_j has a state x_j with the same interpretation. The state of a network is given by two vectors x and y indicating the states of the input nodes and normal nodes, respectively.

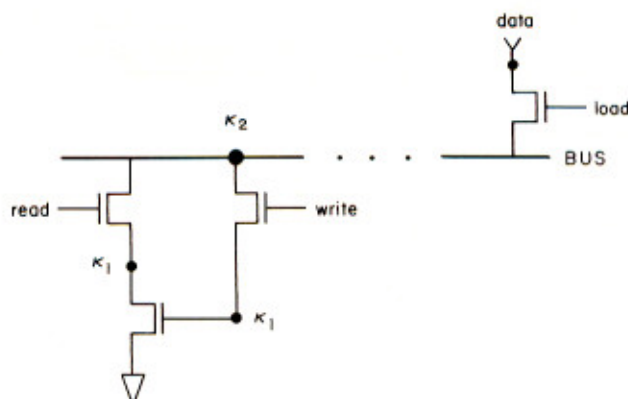


Fig. 1. Switch-Level Model of 3-Transistor Dynamic RAM

A transistor is a three terminal device with terminals labeled "gate", "source" and "drain". It acts as a switch with state determined by the transistor type and the state of the gate node as shown in the following table. The d-type (for "depletion") transistor is used to model both pullup load transistors in depletion mode nMOS circuits and the polysilicon-diffusion layer crossovers seen in some designs.

n-type		p-type		d-type	
gate	effect	gate	effect	gate	effect
0	open	0	closed	0	closed
1	closed	1	open	1	closed
X	unknown	X	unknown	X	closed

Each transistor is assigned a *strength* from the set $\Gamma = \{\gamma_1, \dots, \gamma_p\}$ to indicate its approximate conductance when turned-on relative to other transistors which may form part of a ratioed path. The elements of Γ are totally ordered from γ_1 to γ_p . These transistor strengths allow a simplified model of ratioed circuits in which a path to an input node containing only conducting transistors of strength greater than or equal to some value overrides any path containing a transistor of strength less than this value. Figure 2 shows a switch-level model of an nMOS Nand gate with a pass transistor on its output. Most MOS designs can be modeled with just two transistor strengths ($p=2$), with the load transistors having strength γ_1 and all other transistors having strength γ_2 , although some circuits involve multiple levels of ratioing and hence require more transistor strengths ($p \geq 3$).

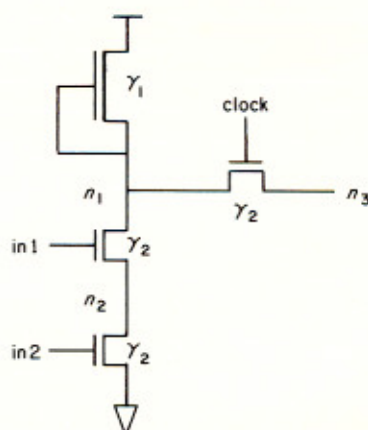


Fig. 2. Switch-level model of nMOS Nand gate.

THE TARGET STATE FUNCTION

The logical behavior of a switch-level network is characterized by its *target state function*. For a given set of input node states x and normal node states y , the target state y' is defined as the node states which the normal nodes would

eventually reach if all transistors were held fixed in states determined by the initial node states. This function describes how the normal nodes attain new logic states due to connections to input nodes or other normal nodes through paths of conducting transistors. This definition ignores the fact that the transistors will change state in response to the changing states of their gate nodes. Thus the target state function only gives an indication of the instantaneous behavior of the network.

The target state function of a switch-level network closely resembles the *excitation* function of a logic gate or relay network, which is defined to give the set of states which would form at the outputs of the logic elements (logic gates of relay coils) in response to the network state given as argument. Huffman (1954) first recognized the importance of the excitation function for characterizing the logical behavior of a network, although he expressed it in terms of a *flow table* containing the excitation state for each possible network state. Thus, although the switch-level model differs greatly from both relay and logic gate models in the way logic states are formed, these models describe the logical behavior of systems in similar ways.

Given a means of computing the target state function, one can implement a form of "unit-delay" logic simulator which simulates the operation of a network by repeatedly applying the target state function. That is, with input nodes set to some state x and normal nodes set initially to state y , the network is simulated until it stabilizes in a state

$$y'' = \lim_{k \rightarrow \infty} T_x^k(y)$$

where the function T_x denotes the target state function for input state x , and the superscript k denotes k applications of the function. For most networks of interest, a stable state will be reached after a bounded number of iterations. Such a method is used by MOSSIM to simulate the effect of each change in clock or data input states. This simulation technique presents the user with a timing model in which the transistors switch one time unit (i.e. one application of T_x) after their gate nodes change state. Such a timing model has proved adequate for testing many LSI designs. Thus a method for computing the target state function provides the key to applying the switch-level model.

LOGIC SIGNALS

For a network containing no transistors in the unknown state,

the formation of logic states on the nodes can be described in terms of an abstraction we call *logic signals*. A logic signal has both state and strength, describing the dominating effect of a network (or subnetwork) at some node, and the relative capacitance or conductance of this effect.

Three types of signals describe the different effects a subnetwork may have on a node. A *charging* signal has state in the set $\{0,1,X\}$ and strength in the set K . It indicates a connection to a set of normal nodes with maximum size equal to the signal strength. A *driving* signal has state in the set $\{0,1,X\}$ and strength in the set Γ . It indicates a set of paths through conducting transistors to input nodes with path strengths equal to the signal strength, where the strength of a path equals the minimum transistor strength in the path. Charging and driving signals of strength s and states $0,1$ and X are denoted $-s$, $+s$ and xs , respectively. Finally, a *null* signal, denoted λ , has a null state N and strength 0 . It indicates an open circuit. The set of signal strength values is totally ordered

$$0 < \kappa_1 < \dots < \kappa_q < \gamma_1 < \dots < \gamma_p$$

indicating that a path to an input node can override a path to a normal node, while either of these can override an open circuit.

Using the set of signal values as domain, we can develop an algebra describing the effects of performing some elementary network transformations. First, when subnetworks described by signals a and b are connected together at a node, the net effect is described by a signal $a \vee b$ equal to the least upper bound of a and b for the partial ordering shown in Fig. 3. That is, a stronger signal will override a weaker, while signals of the same strength will form a signal of this strength and with the same state if they are equal and state X if they conflict. Observe that the set of signal values along with this partial ordering forms a lattice.

Second, when a subnetwork described by a signal a is connected to a node through a transistor in the closed state and having strength s , the net effect is described by a signal $s \circ a$ with state equal to the state of a and strength equal to the minimum of s and the strength of a . That is, a charging signal will connect through unchanged, while the strength of a driving signal may be decreased by the connection. We will adopt a convention that $0 \circ a = \lambda$ i.e. a connection through a 0 conductance gives an open circuit.

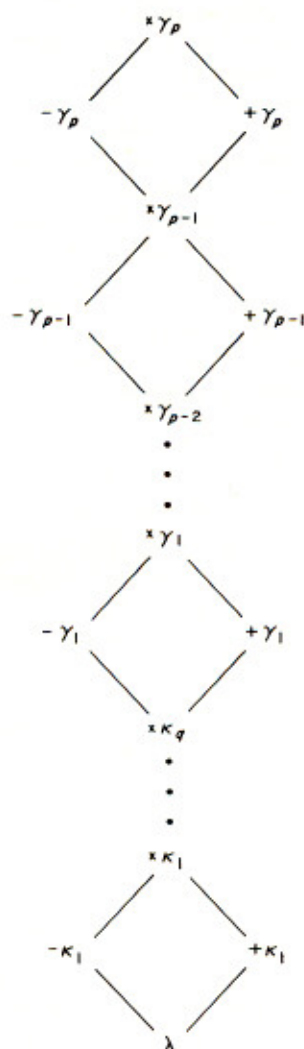


Fig. 3. The Lattice of Logic Signals

EQUATIONS FOR THE TARGET STATE

We wish to develop an equation specifying the target state of a network for a particular input node state x and initial node state y . For the special case in which the network

contains no transistors in the unknown state, the effect of the network on the normal nodes is described by their *steady state signals*, denoted with the vector v , each having state equal to the target state of the node. The values of these signals can be expressed with a matrix equation as follows.

The signal formed by input node i_j , denoted x_j , has state x_j and strength γ_j , the maximum possible strength. The signal formed by normal node n_j , denoted y_j , has state y_j and strength equal to the size of the node. These sets of signals are denoted by the vectors x and y , respectively. The transistors in the closed state are described by two matrices G and E , with g_{ij} equal to the maximum strength transistor in the closed state connecting normal nodes n_i and n_j (or 0 if no such transistor exists), and with e_{ij} describing the analogous connection between normal node n_i and input node i_j . The set of steady state signals v must be the set of minimum values satisfying the equation

$$v = E * x \vee y \vee G * v \quad (1)$$

In this equation $*$ denotes a matrix product with \circ as the analog of multiplication and \vee as the analog of addition. Furthermore, any time a scalar function such as \vee is shown applied to vector arguments, we mean its pointwise extension to a function which yields a vector with elements equal to the result of applying the scalar function to the corresponding elements of the argument(s). This equation expresses the fact that the effect of the network on node n_i equals the combined effect (i.e. least upper bound) of the initial charge on the node as described by the signal y_i ; the direct connections to each input node i_j as described by the signal $e_{ij} \circ x_j$; and the connections to the rest of the network through each other normal node n_j as described by the signal $g_{ij} \circ v_j$. Moreover the set of steady signals equals the set of minimum values satisfying all of these constraints. Observe that this equation has the form $v = f(v)$ with v equal to the least fixed point of the function f .

Equation 1 only applies to networks containing no transistors in the unknown state. In general, however, a network may contain n-type or p-type transistors with gate nodes in the X state. Since this state represents an unknown node voltage anywhere between 0.0 and V_{dd} , the transistor will have an unknown conductance anywhere between 0 and the conductance when fully turned-on. The target state of a node is defined to equal 0 or 1 if and only if it will have this unique state regardless of the conductances formed by the transistors in the unknown state, and otherwise the target state equals X. This definition seems to require trying a possibly

exponential number of cases with the transistors in the unknown state set to all possible combinations of open and closed. Fortunately, the target state of an arbitrary switch-level network can be expressed by a set of matrix equations in an algebra of signal strengths as follows

$$r = E^{\min} \cdot \|x\| + \|y\| + G^{\min} \cdot r \quad (2a)$$

$$u = \text{block}(E^{\max} \cdot [x] + [y] + G^{\max} \cdot u, r) \quad (2b)$$

$$d = \text{block}(E^{\max} \cdot [x] + [y] + G^{\max} \cdot d, r) \quad (2c)$$

In these equations, for a signal a , a equals the strength of a , a equals the strength of a if a has state 1 or X and equals 0 otherwise; and a equals the strength of a if a has state 0 or X and equals 0 otherwise. The operation $+$ gives the maximum of its arguments and \cdot denotes a matrix product with the minimum function as the analog of multiplication and $+$ as the analog of addition. For two strength values a and b , $\text{block}(a, b)$ equals a if $a \geq b$ and equals 0 if $a < b$. The matrices G^{\min} and E^{\min} describe the minimum possible connections between the nodes in which transistors in the unknown have 0 conductance. The matrices G^{\max} and E^{\max} describe the maximum possible connections between the nodes in which transistors in the unknown state are fully conducting.

For a vector r equal to the minimum solution of equation 2a, each element r_i equals the strength of the steady state signal for node n_i when all transistors in the unknown state have 0 conductance. For vectors u and d equal to the minimum solutions of equations 2b and 2c, each element u_i equals the strength of the strongest possible steady state signal on node n_i having state 1 or X for any combination of conductances formed by transistors in the unknown state, while each element d_i equals the corresponding value for signals with states 0 or X. A node n_i will have target state 1 if and only if no possible combination of transistor conductances could give a signal on n_i of state 0 or X, which implies that $d_i = 0$, and similarly it will have target state 0 if and only if $u_i = 0$. Thus the target state can be computed as

$$y_i = \begin{cases} 1, & d_i = 0 \\ 0, & u_i = 0 \\ X, & \text{else.} \end{cases}$$

Thus we have a specification of the target state for an arbitrary network.

COMPUTATION OF THE TARGET STATE

Equation 2a has the form of a fixed point equation $\mathbf{r} = f_s(\mathbf{r})$, where the function f_s is monotonic. The set of signal strengths is finite and totally ordered, and hence it forms a continuous lattice, and any monotonic function over it is continuous, as defined by Scott (1972). Thus, Scott's theorem regarding the least fixed point of a continuous function on a continuous lattice can be applied to show that equation 2a can be solved by an iterative technique where $\mathbf{0}$ denotes a vector of all 0's.

$$\mathbf{r} = \lim_{k \rightarrow \infty} f_s^k(\mathbf{0})$$

Furthermore, convergence will be reached in a bounded number of steps. Using this vector in equation 2b and 2c then gives fixed point equations for \mathbf{u} and \mathbf{d} which can be solved by the same method.

As an example of the computation of the target state, suppose that the network of Fig. 2 has inputs $\text{in}_1 = \text{in}_2 = 1$ and clock = X, and that node n_3 has size κ_1 and initial state 0. The recurrence equation for \mathbf{r} can be written as

$$r_1 = \gamma_1 \uparrow (\gamma_2 \downarrow r_2)$$

$$r_2 = \gamma_2 \uparrow (\gamma_2 \downarrow r_1)$$

$$r_3 = \kappa_1$$

where the operation \uparrow gives the minimum of its arguments. Applying the iterative method the following sequences of values:

$$r_1: 0 \quad \gamma_1 \quad \gamma_2 \quad \gamma_2 \quad \dots$$

$$r_2: 0 \quad \gamma_2 \quad \gamma_2 \quad \gamma_2 \quad \dots$$

$$r_3: 0 \quad \kappa_1 \quad \kappa_1 \quad \kappa_1 \quad \dots$$

from which we get a solution $r_1 = r_2 = \gamma_2$, and $r_3 = \kappa_1$. The recurrence equation for \mathbf{u} can be written as

$$u_1 = \text{block}(\gamma_1 \uparrow (\gamma_2 \downarrow u_2) \uparrow (\gamma_2 \downarrow u_3), \gamma_2)$$

$$u_2 = \text{block}(\gamma_2 \downarrow u_1, \gamma_2)$$

$$u_3 = \text{block}(\gamma_2 \downarrow u_2, \kappa_1)$$

which has a minimum solution $u_1 = u_2 = u_3 = 0$, indicating that regardless of the conductance formed by the pass transistor, no signal with state 1 or X can form on any nodes. Even though the pullup transistor provides a signal of strength $+\gamma_1$ to node n_1 , our computation correctly recognizes that this signal will be overridden by the signal $-\gamma_2$. The recurrence equation for d is

$$\begin{aligned}d_1 &= \text{block}((\gamma_2 + d_2) \uparrow (\gamma_2 + d_3), \gamma_2) \\d_2 &= \text{block}(\gamma_2 \uparrow (\gamma_2 + d_1), \gamma_2) \\d_3 &= \text{block}(\kappa_1 \uparrow (\gamma_2 + d_2), \kappa_1)\end{aligned}$$

which has a minimum solution $d_1 = d_2 = d_3 = \gamma_2$. Thus, since these values are all nonzero, while the values of u are all 0, all three nodes have a target state 0.

If the same network has initial state 1 for n_3 , we would find that $u_3 = \kappa_1$, while all other elements of u and d have the same values as before. This gives target states $y'_1 = y'_2 = 0$, and $y'_3 = X$, indicating that the unknown conductance of the pass transistor creates an ambiguity in the target state of node n_3 .

A logic simulator can be implemented which repeatedly computes the target state using this method. By exploiting the sparseness of the network and the fact that the activity in a network is highly localized, this simulator can operate at speeds comparable to traditional logic gate simulators.

CONCLUSION

The formal model of switch-level networks provides a mathematical link from the physical structure of an MOS system to its logical behavior. This has direct applications in the area of logic simulation, giving logic simulators with greater expressive power and accuracy than those based on the Boolean gate model. Furthermore, other computerized tools and analytic methods can benefit from this ability to move between these two different views of a system.

ACKNOWLEDGEMENTS

This research was supported in part by the United States Department of Energy under contract number DE-AC02-79ER10473, and in part by United States Air Force Contract AFOSR F49620-80-C-0073.

REFERENCES

- Baker, C.M. and Terman, C. (1980). Tools for verifying integrated circuit designs, *Lambda Magazine*, 4th Quarter, pp. 22-30
- Bryant, R.E. (1980). An algorithm for MOS logic simulation, *Lambda Magazine*, 4th Quarter, pp. 46-53.
- Bryant, R.E. (1981a). "A Switch-Level Simulation Model for Integrated Logic Circuits". Phd Thesis, MIT Dept. of EECS.
- Bryant, R.E. (1981b). MOSSIM: a switch-level simulator for MOS LSI, *Proceedings, 18th Design Automation Conference*.
- Huffman, D.A. (1954). The synthesis of sequential switching circuits, *Journal of the Franklin Institute* 257, pp. 161-190, 275-303
- Scott, D.S. (1972). Continuous lattices. In "Toposes, Algebraic Logic and Logic" (Ed. F.W. Lawvere). Springer-Verlag, Berlin