Switch-Level Modeling of MOS Digital Circuits

Randal E. Bryant
Department of Computer Science
California Institute of Technology
Pasadena, CA 91125

TM #5012

This paper will appear in the Proceedings of ISCAS 1982

This work was funded in part by Defense Advanced Research Contracts Agency ARPA Order Number 3771 and by the Caltech Silicon Structures Project

Switch-Level Modeling of MOS Digital Circuits

Randal E. Bryant

Department of Computer Science
California Institute of Technology

Abstract

The switch-level model describes the logical behavior of digital circuits implemented in metal oxide semiconductor (MOS) technology. In this model a network consists of a set of nodes connected by transistor "switches" with each node having a state 0, 1, or X, and each transistor having a state open, closed, or unknown. The logic simulator MOSSIM II has been implemented with this model as its basis. MOSSIM II can simulate a wide variety of MOS circuits at speeds approaching those of event-driven logic gate simulators. The simulator can apply additional tests to detect potential timing errors, unrestored logic levels in CMOS, and unrefreshed dynamic charge. This paper provides an overview of the switch-level model and how it is applied in MOSSIM II.

1 Introduction

The study of formal models for digital systems has progressed little beyond the Boolean gate model, in which a network consists of a set of unidirectional logic elements (gates) connected by one-way, memoryless wires. In contrast to this restricted model, designers of metal-oxide semiconductor (MOS) digital systems have a rich set of design techniques at their disposal. Combinational logic can be implemented with logic gates, programmed logic arrays (PLA's), or pass transistors networks. Data can be stored in static or dynamic registers and memory cells, communicated along wires or busses, and directed through pass transistors. All of these techniques have numerous variations, and hence the designer can tailor the design according to speed, density, and architectural needs. The Boolean gate model cannot accurately describe such behavior because it fails to reflect the basic structure of MOS circuits in which bidirectional logic elements (the field-effect transistors) are connected by multidirectional wires with sufficient capacitance (including attached transistor gates) to store information dynamically.

This mismatch between the formal model and the circuit technology has hampered the development of both manual and computer-based techniques for describing the logical behavior of MOS systems. A number of logic simulators for MOS systems have been developed by extending the Boolean gate model with additional logic states (e.g. "high impedance", "charged 0", "charged 1", etc.) and special logic elements (such as unidirectional pass transistor models), but these extensions lack any mathematical basis and are limited in their generality and accuracy. Inevitably the user must translate the design into a form compatible with the simulator, and the resulting simulation is inherently biased toward the user's understanding of the functionality of the circuit. Many other analytic techniques developed for other forms of logic circuits such as race detection and fault modeling have not been applied successfully to MOS circuits due to this lack of an adequate logic model.

To correct this problem the switch-level model [2] was developed to describe the logical behavior of MOS circuits. In this model a network consists of a set of nodes connected by transistor switches, with node states 0, 1, and X (for uninitialized or invalid) and transistor states open, closed, and 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 in the absence of applied inputs, giving an idealized model of dynamic memory. Nodes can be assigned different sizes to model the effects of their relative capacitances in charge sharing. In keeping with the concept of a logic model, all state and parameter values are from small, discrete sets, and the electrical operation of a circuit is modeled in a highly idealized way. The switch-level representation of an MOS design can be derived directly from the mask specification by a reasonably straightforward circuit extraction program, because the network elements correspond directly to the actual circuit elements. This allows an unbiased test of the circuit as it will actually be fabricated. By developing a model specifically for MOS systems, the model can have greater generality, accuracy, and mathematical rigor than ad hoc extensions of the Boolean gate model.

The simulator MOSSIM II [4] has been implemented with the formal switch-level model as its basis. It has many of the characteristics of earlier switch-level simulators such as the author's MOSSIM[3] and Terman's ESIM.[1] These programs have already demonstrated the utility of the switch-level representation of MOS digital systems. They have been used to verify designs containing over 10,000 transistors and operate at speeds approaching those of conventional logic gate simulators. MOSSIM II improves on these programs with increased accuracy, generality, and speed. This paper will present an overview of the switch-level model and how it is applied in MOSSIM II.

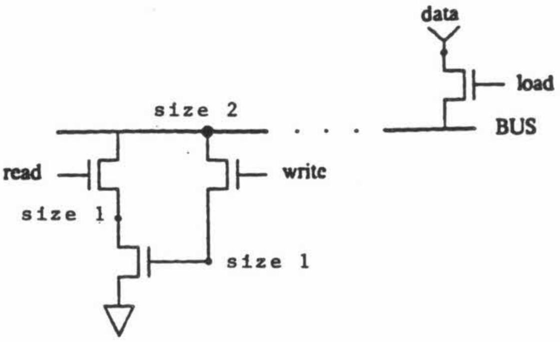

Figure 2-1: Three Transistor Dynamic RAM

2. Network Model

A switch-level network consists of a set of nodes connected by transistor switches. Each node has a state in the set {0,1,X}, with 0 and 1 corresponding to the normal Boolean logic states, and with X indicating an uninitialized node or a node voltage which may lie between the two logic thresholds due to a short circuit or improper charge sharing. 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. Storage nodes have states determined by the operation of the network, and these states are stored dynamically much like the storage of charge in capacitors. Each storage node has a size in the set {1,...,q} to indicate its approximate capacitance relative to other nodes with which it may share charge. These size parameters allow a simplified model of charge sharing in which the state(s) of the largest node(s) have precedence when storage nodes are connected by conducting transistors but are isolated from input nodes. Figure 2 shows a switch-level model of a three transistor dynamic RAM circuit in which the bus node has size 2 to indicate that it can supply 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 circuits can be modeled with at most two node sizes (q=2), with high capacitance nodes such as pre-charged busses assigned size 2 and all others assigned size 1.

A transistor is a three terminal device with terminals labeled "gate", "source", and "drain". No distinction is made between the source and drain connections -- every transistor is a symmetric, bidirectional device. Each transistor has both a strength in the set {1,...,p} and a type in the set {**n**,**p**,**d**}. The strength of a transistor indicates (in a simplified way) its conductance when turned on relative to other transistors which may form part of a ratioed path. When there is at least one path of conducting transistors to a storage node from some input node, the node is driven to a logic state dependent only on the strongest path(s), where the strength of a path equals the minimum transistor strength in the path. CMOS circuits do not involve ratioing and hence can be modeled with one transistor strength (p=1). Most nMOS circuits can be modeled with just two strengths (p=2), with pullup transistors having strength 1 and all others having strength 2.

The three transistor types **n**, **p**, and **d** correspond to n-type, p-type, and depletion mode devices, respectively. A transistor acts as a switch between source and drain controlled by the state of its gate node as follows:

gate	**n**	**p**	**d**
0	open	closed	closed
1	closed	open	closed
X	unknown	unknown	closed

The meaning of the unknown state will be discussed later. Figure 2 shows a switch-level representation of an nMOS Nand gate.

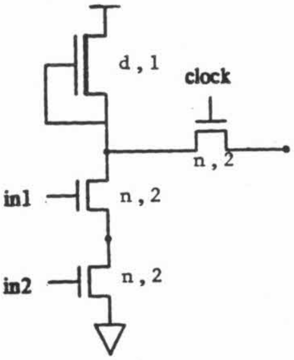

Figure 2-2: nMOS Nand Gate with Pass Transistor

As can be seen, the switch-level model represents an MOS circuit in a highly idealized way. Only enough information about the circuit parameters is included to describe the logical behavior, and it may not predict the true behavior in cases of marginal charge sharing or ratioing.

3. The Excitation State

The operation of a switch-level network is described by its excitation function. For a given set of input and initial storage node states, the excitation function yields the set of states which the storage nodes would eventually reach if all transistors were held fixed in states determined by the initial states on their gate nodes. This function describes how the storage nodes attain new logic states due to connections to input nodes or to other storage 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, and hence the excitation function only gives an indication of the instantaneous behavior of the network. Huffman[6] defined a similar form of excitation function (in the form of an excitation table) to describe the logical behavior of relay networks, and this technique has also been applied to logic gate networks. 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 digital systems in similar ways.

Given a means of computing the excitation function, a "unit-delay" logic simulator can be implemented which simulates the operation of the network by repeatedly computing the excitation state and setting the nodes to this state until a stable state is reached. Such a method is used by MOSSIM II 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 a transistor switches 1 time unit (i.e. one computation of the excitation state) after its gate node changes state, but the effects of signals propagating through paths of conducting transistors is simulated with no delay. This timing model only loosely approximates the actual circuit timing. It is assumed that the circuit's logical behavior is not critically dependent on its timing behavior. Other means for verifying that a circuit is free of timing errors will be discussed later.

MOSSIM II computes the excitation function by solving a set of equations in a simple, discrete algebra of logic signals. A logic signal has both state and strength, describing the effect of a subnetwork at a node in terms of the logic state which it would create on the node if the rest of the network were removed, and the relative capacitance or conductance with which it supplies this state. A set of matrix equations in this algebra expresses how the steady state signals are formed on the storage nodes for a given set of input and initial storage node states. The states of these steady state signals equal the excitation states of the nodes.

In general a network may contain transistors in the unknown state, i.e. n-type or p-type transistors with gate nodes in the X state. Since the X state represents an unknown voltage between (inclusively) 0.0 and Vdd, such a transistor can have arbitrary conductance ranging between its conductance when open (i.e. 0.0) and when closed. The excitation state of a node is defined to equal 0 or 1 if and only if it would have this unique state regardless of the conductances of transistors in the unknown state, and otherwise it equals X. A straightforward implementation of this definition would require an algorithm of exponential complexity (or worse if one considers intermediate conductance values), in which the excitation state is computed with the transistors in the unknown state set to all possible combinations of open and closed. Fortunately, the same result can be achieved with a linear algorithm which first finds a consistent network state with as many nodes set to 1 or X as possible and then finds a consistent network state with as many nodes set to 0 or X as possible. The excitation state equals 0 if the first pass fails to set a node to 1 or X, equals 1 if the second pass fails to set it to 0 or X, and otherwise equals X. Furthermore, a simulation algorithm can be implemented which exploits the sparseness of the network interconnections and the locality of activity to reduce the amount of computation to a level comparable with event-driven logic gate simulators.

4.TimingVerification

Many MOS circuits are designed to operate with multi-phase, nonoverlapping clocks such that no critical races can occur as long as the clock and data inputs change slowly enough to allow the circuit to stabilize between each set of changes. MOSSIM II can verify that a circuit obeys this property for a particular input sequence by applying a ternary simulation model[5]. This verification is more powerful than could be obtained from any circuit simulator, because it is independent of the actual circuit parameters. With ternary simulation, the effect of a set of changing clock or data inputs is simulated by first setting these inputs to X and operating the simulator until a stable state is reached. This has the effect of setting every node which may be in transition to X. Then the inputs are set to their new values and the simulator is operated until a stable state is reached. As a consequence, any nodes which depend only on inputs or stable internal nodes attain a Boolean state, while nodes which depend on the relative ordering of the transitions remain in the X state to indicate a possible critical race. Ternary simulation performs a very stringent test of the circuit. A node is set to 0 or 1 if and only if it would have this unique state regardless of the switching delays in the transistors and the line delays in the wires. The ternary simulation algorithm has linear complexity in the size of the circuit and hence can be used in simulating large circuits over long input sequences. Experience has shown that ternary simulation operates only 2 to 3 times slower than the normal unit delay simulation.

Even with ternary simulation, however, timing errors caused by glitches due to transient charge sharing effects may not be detected. MOSSIM II can augment ternary simulation with a test for this dynamic charge sharing. That is, during the first part of the simulation of a set of input changes, a node is set to X not only if it is in transition, but also if it would have a different excitation state if all connections to input nodes were ignored. During the second part, the network is simulated with the inputs set to their final values as before. As a result, nodes which could be affected by the ordering of transitions or by the presences of glitches remain in the X state. As with ternary simulation, this test is very conservative, but any circuit which passes it is guaranteed to be free of any timing errors for the particular sets of input sequences simulated.

5.AdditionalTests

In addition to detecting timing errors, MOSSIM II can detect other forms of circuit design errors. For example, the simulator can check for unrestored logic levels in CMOS in which a node has a voltage one threshold below Vdd or above Vss due to improper use of n-type and p-type transistors. By implementing a modified form of the excitation function equations, the simulator will set such nodes to X. As another example, a limit can be placed on the charge retention time (measured in clock cycles) such that a charged node is set to X if it goes unrefreshed for too long. All of these tests are performed by simply setting mode switches in the simulator and hence are quite convenient.

6.Conclusion

The development of a formal model for MOS circuits has already yielded practical results in the form of the logic simulator MOSSIM II. Unlike earlier simulators based on ad hoc extensions of the Boolean gate model, MOSSIM II achieves a high degree of generality and accuracy while achieving a comparable level of performance.

The simulator can also apply a variety of tests to verify that a circuit is designed according to some methodology which helps insure its correct functionality. For example, ternary simulation and dynamic charge sharing tests are used to detect potential timing errors independent of the actual circuit parameters. Other tests detect unrestored logic levels in CMOS and unrefreshed dynamic charge. All of these tests use the X state to represent an error and hence require a careful and efficient modeling of the X state. This would be very difficult to achieve without the guidance of a formal model.

The switch-level representation of a circuit seems well suited for a realistic fault model for MOS, because it can describe the effects of faulty transistors (stuck open or closed), short circuits (nodes merged together) and open circuits (nodes split apart). However, the response of a faulty circuit to a test sequence is often history sensitive, because every node can store state dynamically. Thus, traditional algorithms for deriving test sequences for combinational logic have little value in MOS. The development of fault analysis techniques for MOS based on the switch-level model remains an important area of research.

7.references

[1]
Baker, C., and C. Terman.
Tools for Verifying Integrated Circuit Designs.
Lambda Magazine :22-30, Fourth Quarter, 1980.

[2]
Bryant, R.
A Switch-Level Simulation Model for Integrated Logic Circuits.
PhD thesis, Massachusetts Institute of Technology, 1981.

[3]
Bryant, R.
MOSSIM: A Switch-Level Simulator for MOS LSI.
In Eighteenth Design Automation Conference, pages 786-790. ACM, 1981.

[4]
Bryant, R., Schuster, M., and Whiting, D.
MOSSIM II: A Switch-Level Simulator for MOS LSI, User's Manual
Department of Computer Science, California Institute of Technology, 1982.

[5]
Brzozowski, J. A., and M. Yoeli.
On a Ternary Model of Gate Networks.
IEEE Transactions on Computers :178-183, March, 1979.

[6]
Huffman, D. A.
The Synthesis of Sequential Switching Circuits.
Journal of the Franklin Institute 257 :161-190, 275-303, March and April, 1954.