Switch Level Simulation

Submitted by Blessil George
Reveiwer: Malinky Ghosh

***ABSTRACT*- Switch level simulation has become a common means for accurate modeling of MOS circuit behavior. This paper discusses the switch level simulation model as described by [1]. The basic network model and other parameters are thoroughly discussed. Thereafter the paper discusses a new algorithm develop by [2] for detecting logic gate implementation and accurately modeling their switch level behavior. The algorithm has also successfully increased the speed of simulation.**

1. INTRODUCTION

Simulation of large circuits is often performed at the switch level. Simulation at this level has the advantage that several detailed phenomena such as charge sharing, bidirectional signal flow, signal strength and resolution of conflicting signals are accurately modeled. A logic gate implementation can be modeled at the switch level using a simple Boolean function, describing its logic operation and strength information. This model represents a circuit in terms of its exact transistor structure but describes the electrical behavior in a highly idealized way. It expresses transistor conductance and node capacitances by discreet strength and size values; represents node voltages by discrete states 0, 1, X and makes no attempt to model exact circuit timing.

Programs based on switch level can operate at speeds approaching those of their counter parts based on more traditional gate level models. Applications like logic simulators, fault simulators, test pattern generators use switch level models successfully.

2. SWITCH LEVEL ALGORITHMS

To accommodate the bidirectional nature of the transistors, the programs modeling circuit at switch level compute the state of a node by applying graph algorithms to trace the connections between nodes formed by conducting transistors. These programs have several drawbacks. Considerable effort is often required to adapt existing techniques for use at the switch level. Computing node states using graph algorithms to the transistor data structure requires significantly greater effort than computing the output of the logic gates. Although special purpose switch level simulators have been constructed, their requirement of hardware is much more.

3. THE SWITCH LEVEL MODEL

This section gives and overview of the model in terms of a cleaner notation and defines the symbolic analysis problem.

Network Model

A switch level network consists of nodes and transistors. Node can be of two types;

Input nodes: they act as connection to a signal source external to the chip supplying power, ground, clock or data.

Storage nodes: These are like capacitors in any electrical circuit which retain charge in absence of inputs and can also share charge with other storage elements.

Voltage on nodes are represented by 0, 1 or X; where 0 represents low, 1 represents high and X represents an uninitialised network or an error condition.

The node capacitance with respect to other nodes are specified in terms of its size from the set {1,2,…..,k}.When a set of storage nodes are connected the nodes of maximum size determine the outcome. The input nodes have greater size than the other nodes in the circuit. These nodes in a circuit are connected by the transistors. A transistor terminal is labeled as gate, source and drain. In switch model simulation its viewed as a resistive element connecting the drain and source controlled by the voltage value in the gate. The transistors are bidirectional elements allowing current flow in both directions. They are considered to be of 3 types.

D type: it always conducts

P type: it conducts when the gate has state 0

N type: it conducts when the gate has state 1

The transistor states 1, 0 and X represent the transistors are nonconducting, fully conducting and indeterminate respectively.

Another parameter determining the transistors are their strength. Its defined as a set {k+1, k=2, k=3,…………, w-1}. It defines the transistor conductance with respect to the other transistors in the circuit.

The channel graph

In general channel graph represents the interconnection structure of a switch –level network. This graph has the storage nodes of the circuit as vertices. It describes the static interconnections between the storage nodes formed by the source-drain connections of the transistors. It can be either unidirected or bidirected depending upon the structural properties of the circuit.

The steady state response

The steady state response function describes the behavior of a sub network. The steady state response for a node equals the state (0, 1 or X) this node would attain if the transistors were held fixed for long enough for the nodes to stabilize.

4. FUNCTIONAL ABSTRACTION OF LOGIC GATES

In this paper we will study a method for detecting a logic gate implementation and accurately modeling its switch level behavior. In a switch level circuit description each circuit element (transistor or circuit node) can display the full range of switch level phenomena. We call the possible operation of an element or a group of elements its functional domain. When a circuit element is considered as a part of over all circuit its actual operation is only a subset of this functional domain which we call its functional application. For accurate simulation, only the functional application rather than the entire functional domain should be modeled.

The functional domain of a static gate implementation includes charge sharing effects between its nodes, charge storage effects at each node and bidirectional signal propagation through each of its transistors. Most of these phenomena need not be modeled because they do not occur in the actual operation of the gate. A static CMOS gate is characterized by pull up and pull down functions that are duals of each other. There is always a path from the gate output to either the power or the ground but not simultaneously to both. some of the phenomena that are present in the functional domain but not in functional application are:

1. Charge sharing and charge storing.
2. Bi-directional signal flow.
3. Conflicting signal Resolution.

Because of these functional application of a logic gate is greatly simplified. Modeling a logic gate is further simplified by the fact that its inputs all drive transistor gates. Only the logic states of the inputs is needed and there strengths can be ignored. The logic state of the output is thus only a function of logic state of inputs. The only feature which has to be considered is the strength of pull-up and pull down of the gate output.

5. ABSTRACTION ALGORITHMS

Gate abstraction for switch level simulation consists of three parts:

1. logic analysis-obtains the logic pull-up and pull down functions of a dc-connected component and determines if it represents a valid logic gate.
2. strength analysis: analyses the strength of pull-up and pull down paths and substitutes the transistor implementation with this information. It incorporates the information in a gate descriptor.
3. generates an evaluation routine for gate descriptors.

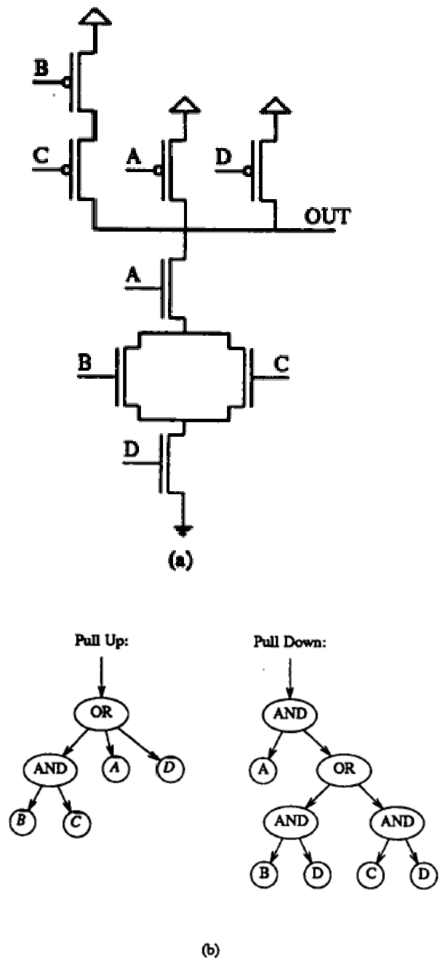

Figure 1: Pull up and Pull down LFT of a logic gate.

Logic analysis

The algorithm performs logic analysis in three phases as explained below:

Identifying Potential gate outputs

A node is classified as a potential gate output (PGO) if it is connected to a channel of a N-type transistor and a channel of a P-type transistor. For each PGO node, the dc-connected transistors are then examined to determine if the structure represents a logic gate.

Constructing the logic function tree

In the second phase transistors that are dc-connected to the PGO are traversed to form a directional tree that represents logic function of the transistor network. The nodes of the logic function tree are either AND or OR functions and its leaves are circuit signals either direct or in inverted form. The number of children for each node is arbitrary. One logic tree is generated for the N-type section of the network and one is generated for the P-type section of the network. With this the logic pull up and pull down functions are expressed independently. They are duals of each other. With this type of representation the algorithm traces N-type and P-type trees separately. While tracing it checks the network connectivity. If an illegal construct is encountered it either aborts the gate detection process or removes the function of the illegal construct from the tree and continues with the gate detection process. The algorithm performs a depth first trace of all possible paths from a PGO node. For each traversed transistor, an AND gate with a leaf node is added to the existing tree, for each circuit node with multiple fan-out, an OR-gate is added to the tree.

Conversion to Sum of product notation

In phase 2 the pull-up and pull down function of transistor network is represented in the form of a LFT. For each valid gate the function represented by P-tree is a dual of that represented by N-tree. Many times the generated logic function trees are quite different and depend on used design style, the used circuit extraction program and the layout of the circuit. It is difficult to determine if the two trees represent dual functions of each other. In order to avoid these design related problems the logic analysis converts the LFT in to sum of products (SOP) representation. This conversion simplifies the test for equivalence. For a P-LFT, the SOP notation of the dual of the function is generated, where as for the N-LFT the SOP notation of the function itself is generated. Redundancies are removed before N-SOP and P-SOP are checked for equivalence.

6. STRENGTH ANALYSIS

Switch level simulation utilizes a conductance model to determine the strength of a transistor network. When the circuit considered contains transistors connected in series the combined strength of the circuit is considered to be the minimum of the strengths of the corresponding transistors. When the transistor is connected in parallel the combined strength of circuit is the maximum of the individual transistor strengths. For example, consider a gate network. It consists of several parallel branches, each having many transistors connected in series. Therefore the strength of each branch is the minimum conductance of each transistor in the branch whereas the strength of the gate is the maximum strength of the corresponding branches. When all the branches have same strength they are said to be strength consistent. When both P section (connected to Vdd) and N section (connected to ground) have same strengths the gate is said to strength consistent. The SOP notation each product correspond to each branch of the circuit. Therefore the strength of the product is the minimum strength of the constituents. If either the pull up function or the pulls down functions do not contain equal products then they are referred to as strength inconsistent.

7. GATE EVALUATION METHODS

The gate abstraction program that is dealt with after the strength and logic analysis has two attributes; the gate descriptor and the gate evaluation routine. The gate descriptor consists of a list of parameters required for the gate evaluation. When the simulator encounters the gate descriptor it passes the data in it to the gate evaluation routine. The further method of evaluation differs for consistent and in consistent strength gates.
For strength consistent gates, the gate is described by the operation it performs the number of inputs, the strength of the pull up or pull down functions. A gate built of transistors can be a NAND, NOR or OR, AND or NOT type. Evaluation is in this case performed by a look up table or a polling routine. Since the gate descriptor consists of the number of input and the up down strengths only one evaluator is required for one type of gate. The routine evaluates the gate inputs sequentially and requires evaluation time though every iteration only one or two sentences.

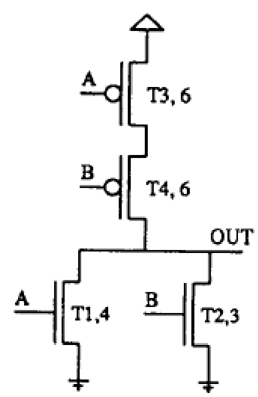

Figure 2: Logic gate having different pull up and pull down strengths

For strength inconsistent gates the gate output strength is not only a function output but also the gate inputs. The evaluator uses a truth table consisting of the output states and strength of each with respect to the input states. Gates with a maximum of 4 inputs are modeled this way; this is because the size of truth table increases exponentially with the number of inputs to each gate. For larger gates a method called the Coded Personality Matrix is used.

Evaluation method for consistent gates is faster and requires less storage space than the strength inconsistent gates.

8. CONCLUSIONS

The algorithm mentioned in the paper presents an accurate means of increasing the switch level simulation. For most logic gates the modeling was made easier using this algorithm. On experimental analysis of several large circuits, it's seen that the algorithm replaced most of the transistors and increased the simulation speed by 1.81.

9. REFERENCE

[1] R. E. Bryant, "A switch level model and simulator for the MOS digital systems," IEEE Trans Comput, vol. C-33, pp. 160-177, Feb 1984.

[2] R. E. Bryant, "Boolean analysis of MOS Circuits," IEEE Trans Comput, vol: 6, Issue 4, pp. 634-649, July 1987.

[3] D.T. Blaauw, D.G. Saab, P.Banerjee, "Functional abstraction of logic gates for switch-level simulation", IEEE, 1991.