11-18-1991

Comparison of Spice and Network C Simulation Models Using the CAM System

Wen-Tsung Yen
Portland State University

AN ABSTRACT OF THE THESIS OF Wen-Tsung Yen for the Master of Science in Electrical Engineering presented on November 18, 1991.

Title: Comparison of SPICE and Network C Simulation Models Using The CAM System.

APPROVED BY THE MEMBERS OF THE THESIS COMMITTEE:

[redacted]

W. Robert Daasch, Chair

[redacted]

Fu Li

[redacted]

Andrew Fraser

The performance of SPICE and Network C (NC) circuit simulator when simulating MOS transistor circuits has been investigated and compared. SPICE analog model, NC analog model and NC MOS_PWL model are the three MOS transistor models being used. The comparison between SPICE and NC includes five areas. They are MOS transistor model, circuit analysis and computational methods, limitation on the ability to simulate circuits containing the MOS transistor diode configuration, run time and the ability to build new circuit component models using derived equations.

The prototype circuit being used is the Content-Addressable Memory (CAM) circuit. The CAM circuit is a good example because its component circuits such as priority resolver circuit and CAM cell circuit can bring out the significant differences when using

SPICE and NC. The priority resolver circuit in CAM is designed to contain MOS transistors connected to behave like diodes so as to show the limitation of the NC MOS_PWL model. The CAM cell circuit has a p-n junction leakage problem which can be modeled using derived equations. The building of new circuit components using derived equations will be demonstrated through this leakage problem.

COMPARISON OF SPICE AND NETWORK C SIMULATION MODELS USING THE CAM SYSTEM

by

WEN-TSUNG YEN

A thesis submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE
in
ELECTRICAL ENGINEERING

Portland State University
1991

TO THE OFFICE OF GRADUATE STUDIES:

The members of the Committee approve the thesis of Wen-Tsung Yen presented November 18, 1991.

[redacted]

W. Robert Daasch, Chair

[redacted]

Fu Li

[redacted]

Andrew Fraser

APPROVED:

[redacted]

Rolf Schaumann, Chairman, Department of Electrical Engineering

[redacted]

C. William Savery, Interim Vice Provost for Graduate Studies and Research

ACKNOWLEDGEMENTS

I wish to give special regards to Dr. Robert Daasch, my academic advisor, who has been patiently guiding and supporting me throughout the process of this research work. His generous comments and helpful suggestions are essential in this work and without them the completion of my thesis would be impossible. I also wish to thank Dr. Andrew Fraser and Dr. Fu Li for providing me valuable comments.

Special thanks to Ms. Shirley Clark for her help during my entire stay at PSU and to Mr. Janaka Jayawardena for providing me the convenience of using the utility equipment.

I would like to especially thank my parents, Mr. Chen-Tao Yen and Mrs. Shen-Shia Yen and my brother Wen-Tzu Yen who have been supporting me financially and emotionally.

Finally, thanks to all my friends at PSU for their constant support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Modern technology has advanced to a point where physical circuits can be represented and tested using software programs in circuit simulators. A designer no longer needs to physically build the circuit, probe various nodes, show the nodes on the oscilloscope and note the voltages and currents of the nodes at each time step. Often times, the designer will have to redesign the circuit and do the testing process all over again when the previous attempt fails. This way of designing not only consumes tremendous amounts of time but also inefficiently uses the resources. The cost needed in this way of designing will definitely exceed that which involves using simulators. Simulators can provide the designer with the worst and best case scenarios with just a few runs of simulation. This way, the designer can easily optimized a design with very little time spent. When designing small circuits, the designer may be able to get by without using simulators. However, when large circuit designs involving hundreds and thousands of devices are needed, it is almost impossible to optimize the design without using simulators. No doubt, simulation tools play an important role in today's design world. To succeed in today's design world, it is equally essential to have good simulators in a design process as to have good knowledge of which type of simulator to use in designing which type of circuit.

MOSFET SIMULATORS OVERVIEW

Different types of circuit design require different kinds of simulation tools. With CMOS (Complementary Metal-Oxide Silicon) circuit design, the types of simulation tools frequently used are circuit level and switch level simulators. Circuit level simula-

tors are usually designed with detail semiconductor device modeling and detail circuit analysis capability. They can provide the user with specific timing information, voltage and current values on nodes in the circuit. However this type of simulator often uses up large amounts of CPU time. Circuit level simulators are perfect for simulating small circuit designs but are not recommended to simulate large circuits. SPICE [Nagal 75] and ASTAP [Weeks 73] are typical examples of circuit level simulators. Their performance can be effectively improved by reducing consumption of CPU time while maintaining certain level of accuracy. These improved circuit level simulators are known as timing mode simulators and MOTIS [ChGK 75] and Network C (NC) [Beckett 88] are two examples of them. MOTIS uses the gate to gate propagation signal which is used in logic simulators and a stored table for device models to reduce the simulation time. NC uses methods such as even driven algorithms, calculation history, decomposition of the circuit into stages and the technique of output prediction based on the given input to reduce the simulation time. The discussion on these methods and a comparison of SPICE and NC will be presented in Chapter II.

Switch level simulators are basically logic simulators. They are named switch level because they model the MOS transistor as some kind of a switch which can be opened, closed or intermediate. Logic simulators such as MOSSIM [Bryant 81] is a good example of switch level simulator. Although the newer version MOSSIM II [Bryant 84] can model accurately a number of MOS transistor characteristics quite accurately, it can only provide the user with just logic information of a circuit. Other logic simulators such as RSIM [Terman 83] and [Ruan 88] can not only provide the logic but also the timing information of the circuit. RSIM models the MOS transistor as an effective resistance. Each node in the circuit has a certain capacitance and RSIM determines the value of the node and the charging and discharging time through the RC network. Ruan uses a different approach. No resistors are allowed in his model. Each active transistor acts like a current-limited switch and is modeled as a current source which has two piecewise

constant level of currents. No mathematical integration, no model evaluation and no table lookup are needed to obtain the timing values in the simulation.

Logic simulators usually model their devices using simple equations so that no lengthy and time consuming calculations are needed. These qualities in the logic simulators make them much more efficient to use, in terms of simulation time, than the circuit level simulators. Logic simulator is a perfect tool to use when only functional or logical understanding of a circuit is required and no precise accuracy is needed.

WHY SPICE AND NC?

Although there are many simulators available, the focus of this project is on the comparison of SPICE and NC. The SPICE circuit simulator has been existing since 1975 when it was first introduced by a research group from the University of California at Berkeley [Nagal 75]. SPICE is designed to be a circuit simulator capable of simulating circuits made up of the traditional electrical components. These components may include resistors, capacitors, inductors, mutual inductors, independent or dependent voltage and current sources, transmission lines, switches, diodes, bipolar junction transistors (BJT), junction field-effect transistors (JFET), metal-semiconductor field-effect transistors (MESFET) and metal-oxide silicon field-effect transistors (MOSFET). We will be using the SPICE3 version of SPICE. NC, on the other hand, has just recently been developed (1988) by William Beckett in the University of Washington [Beckett 88]. NC uses the C programming language to design simulation models and uses these models as components to construct electronic circuits and systems. Both simulators are not design primarily for simulating MOSFET circuits but they work equally well compare to those simulators that are. There are 2 objectives in this project:

1. To compare how the different way of modeling MOSFET in SPICE and NC can affect the performance of the simulator when simulating MOS prototype circuits.

2. Through simulating the Content-Addressable Memory (CAM) as a prototype circuit, we are interested in bringing out the unique characteristics of SPICE and NC.

To understand how the comparison between SPICE and NC is significant and why they are chosen, we need to first discuss the structure of the CAM circuit which will be simulated using both simulators as a MOS prototype circuit. Figure 1 shows a CAM architecture which is made up of a select register, a data register, a mask register, a control circuit, an address decoder, CAM cell array, a priority resolver and a sense amplifier.

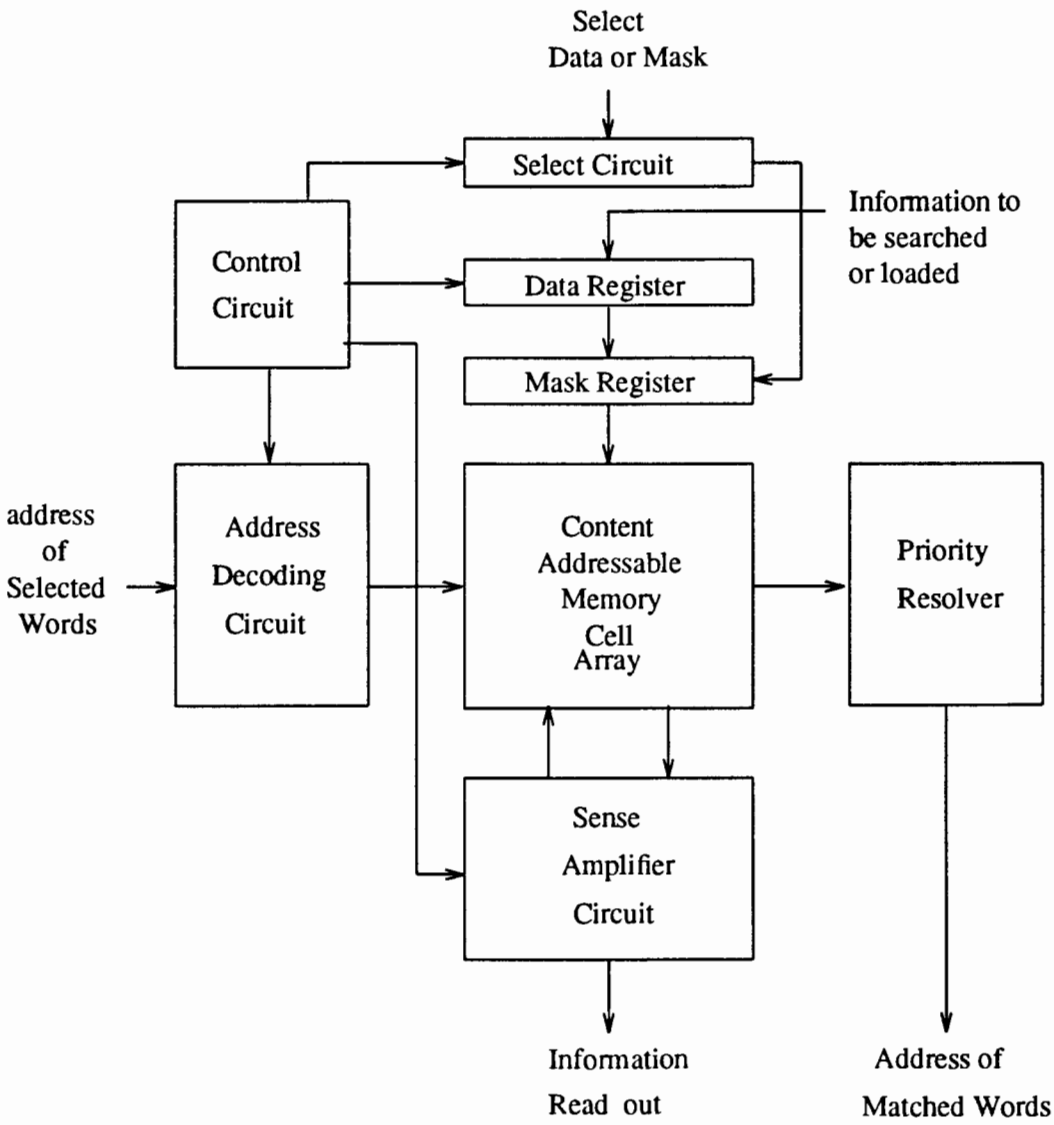

Figure 1. Block diagrams of a CAM architecture.

The CAM is used as a prototype circuit because the design of its component circuits can show the limitation of certain simulators when the simulator is simulating certain types of circuits. For example, the design of the priority resolver consists of NMOS transistor connected to behave like a diode. Due to the method that NC uses to break down circuits into smaller circuits during simulation, the diode configuration transistors may create problems for the NC MOS_PWL model to perform an accurate simulation. Chapter III is devoted to the discussion on this diode configuration transistor problem.

Content-Addressable Memory cell array in the CAM architecture consists of memory cells. These memory cells are designed in such a way that they store information as charge on their gate capacitance of the transistor. The charge is trapped on the gate capacitance of the transistor when the transmission gate (acts as a path) that the charge is passing through is closed. This type of memory cell is known as the dynamic memory cell. One problem can occur when using this method of storing charge. The charge can leak through the p-n junction of the transmission gate and given enough time, the charge stored on the gate capacitance can drop to a voltage which can be misinterpreted as a logic '0' rather than a logic '1'. NC has a unique capability that few simulators have. It allows the user to built his own models to be used as circuit components when needed. This leaking phenomenon can be modeled and become a component. Every time a similar leakage problem occurs, this component can be used. This leakage example is used to show the power of NC in this regard. Chapter IV mainly discuss the development of the leakage model and the results.

The sense amplifier circuit is also discussed in this chapter. Sense amplifier is an analog circuit. However, it is possible to design a sense amplifier using MOS transistors. Sense amplifiers are usually used to detect small differences between two voltages. The voltage difference is then fed to other supporting circuits which helps to restore the original signals. This difference is allowed to be as close as 0.5 volts in order that the signals

be restored in a short amount of time. With this type of circuit, the logic simulator is powerless in getting an accurate simulation. Logic simulators can only interpret inputs as logic '1', '0' or 'X' (don't care state) and with a voltage difference of only 0.5 volts, they are not able to interpret correctly which input is at logic '1' and which input is at logic '0'. Therefore logic simulators are not chosen in our comparison of simulators simulating the CAM circuit.

The discussion in Chapter V is on the performance of NC and SPICE using subcircuits as the way of representing circuits in their input program. The discussion and analysis on the amount of simulation time needed for each simulator as the circuit gets very large is also included in this chapter. The mask network which consists of the data register, mask register and select register is the prototype circuit used in the discussion. Data register is used in the load operation and consists of columns of dlatch circuit which is built using clocked inverters and inverters. The mask register is used in the mask operation and consists of columns of mask circuits which is built using transmission gates and NMOS transistors. The select register is designed with columns of select circuits which is built with NAND gates, NOR gates and inverters which are all CMOS logic gates. The select circuit has a control bit which can be set by the user to select in triggering the mask operation or the load operation. Due to the hierarchical structure of these circuits, they can be represented in the simulator using subcircuits. Both NC and SPICE support the use of subcircuits. Our interest is to see whether using subcircuits can reduce the simulation time required by both simulators and also has the accuracy of the simulation result been sacrificed.

CHAPTER II

SPICE AND NC COMPARISON

The SPICE circuit simulator has been existing since 1975 when it was first introduced by a research group from the University of California at Berkeley. Through the years, SPICE has gained wide acceptance in both the industrial and academic world due to its ability to do accurate circuit analysis. NC, on the other hand, has just been recently developed (1988) by William Beckett in the University of Washington [Becket 88]. NC has a few distinct features that SPICE does not support. For example, NC allows users to build their own models and used these models as circuit components in the simulation. Another advantage NC has is the quickness in performing simulation. Although both simulators are able to not only simulate circuits built with MOS transistors, our focus in this chapter will narrow the comparison of their simulation down to circuits built with MOS transistors. There are mainly 2 areas that will be looked at:

1. The way both simulators model their MOS transistors.
2. The method used by both simulators to analyze and compute their circuits.

A third area which is related to the main focus in this chapter but is of lesser importance is the differences between handling the Input-Output process in SPICE and NC. That discussion can be found in appendix A.

MOSFET MODEL

There are basically three types of MOS transistor models available in SPICE and they are the level 1, level 2 and level 3 model. A level 4 BSIM model has recently been developed and is available in SPICE3. The discussion on level 3 and 4 model is beyond our scope and our emphasis will be on the level 1 and level 2 models. The model in level

1 is based on the famous Shickman-Hodges model [ShiHod 68]. This model is the simplest MOS transistor model in SPICE which is used in verifying the behavior of a design where no great accuracy is required. It uses a few basic parameters to describe the fundamental I-V characteristics of the MOS transistors in their three operating regions. Table I shows the symbols used in the equations along with their description and default values. Table II shows the list of physical constants appear in the equations.

TABLE I

DEFINITION AND DEFAULT VALUES OF COMMON SYMBOLS

Symbol	SPICE Name	Description	Default Value if nonzero
KP	KP	Transconductance parameter	2×10^{-5}
X_{jl}	LD	Lateral diffusion	
λ	LAMBDA	Channel-length modulation	
V_{TO}	VTO	Zero bias threshold voltage	1.0
γ	GAMMA	Body effect parameter	
ϕ_p	PHI	Surface potential (volts)	0.6
N_A		P-type substrate doping (carriers/cm^3)	1.0×10^{15}
μ	UO	Nominal channel mobility ($\frac{cm^2}{N}A$)	600
C_{ox}		Oxide capacitance (farads/$meter^2$)	
t_{ox}	TOX	Oxide thickness (meters)	
T		Temperature (kelvin)	300

TABLE II

DEFINITION AND VALUES OF COMMONLY USED CONSTANTS

Symbol	Description	Constant Value
ε_{si}	Dielectric constant of silicon (Farads/meter)	1.04×10^{-10}
ε_{ox}	Dielectric constant of silicon dioxide (Farads/meter)	3.45×10^{-11}
n_i	Intrinsic carrier concentration, silicon (carriers/cm^3)	1.45×10^{10}
κ	Boltzman's constant (joules/degree-kelvin)	1.38×10^{-23}
q	Electronic charge (coulombs)	1.602×10^{-19}

The equations used in the level 1 model are:

$$I_{DS} = 0 \tag{2.1}$$

2. Nonlinear region: $V_{GS} \geq V_{TH}$ and $V_{GS} - V_{TH} \geq V_{DS}$

$$I_{DS} = \frac{KP}{2} \frac{W}{L - 2X_{jl}}(V_{GS} - V_{TH} - \frac{V_{DS}}{2})V_{DS} \tag{2.2}$$

3. Saturation region: $V_{GS} \geq V_{TH}$ and $V_{GS} - V_{TH} \leq V_{DS}$

$$I_{DS} = \frac{KP}{2} \frac{W}{L - 2X_{jl}}(V_{GS} - V_{TH})^2 V_{DS} \tag{2.3}$$

where

$$KP = \mu C_{ox} \tag{2.4}$$

$$C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} \tag{2.5}$$

$$V_{TH} = V_{t0} + \gamma(\sqrt{2\phi_p + V_{BS}} - \sqrt{2\phi_p}) \tag{2.6}$$

$$V_{D,sat} = V_{GS} - V_{TH} \tag{2.7}$$

$$\gamma = \frac{\sqrt{2\varepsilon_s q N_A}}{C_{ox}} \tag{2.8}$$

$$\phi_p = \frac{kT}{q} \ln \frac{N_A}{n_i} \tag{2.9}$$

The level 2 model is a more complicated model. It includes the correction factor of the conductance in the saturation region, δ and the second-order phenomena in MOS transistors [Glasser 85]. Originally, the charge in the depletion region, Q_B, is assumed to be approximately constant. This approximation is true only when V_{DS} is small. The reason is that when V_{DS} is large, the depletion region thickness is bigger near the drain and shorter near the source. The potential difference in this region actually varies from the drain end to the source end. Q_B calculated using the constant approximation will be smaller than the actual one. The correction factor, δ is introduced to reduce the current at higher V_{DS} and it is defined as

$$\delta = \frac{\gamma}{2\sqrt{V_{BS} + 2\phi_p}} \tag{2.10}$$

The equations with the correction factor are:

1. Cutoff region: $(V_{GS} - V_{TH}) \leq 0$

$$I_{DS} = 0 \tag{2.11}$$

2. Nonlinear region: $V_{GS} \geq V_{TH}$ and $V_{GS} - V_{TH} \geq (1 + \delta)V_{DS}$

$$I_{DS} = \frac{KP}{2} \frac{W}{L - 2X_{jl}}(V_{GS} - V_{TH} - \frac{V_{DS}}{2})V_{DS}(1 + \delta) \tag{2.12}$$

3. Saturation region: $V_{GS} \geq V_{TH}$ and $V_{GS} - V_{TH} \leq (1 + \delta)V_{DS}$

$$I_{DS} = \frac{KP}{2} \frac{W}{L - 2X_{jl}}(V_{GS} - V_{TH})^2 V_{DS}(1 + \delta) \tag{2.13}$$

Level 1 model is no longer valid when the MOS transistor has narrow or short channel. Some special behavior of the MOS transistor will occur in this case. They are subthreshold conduction, reduction in mobility, change in $V_{D,sat}$, V_{t0} and I_{DS} due to narrow channel and short channel effects and the channel length modulation effect. These effects are known as the second-order phenomena in MOS transistors. These effects are not going to be discussed in great detail since no narrow or short channel MOS transistors are used in the CAM circuit. The minimum size transistor used in the design is 2.0 μ length and 3.0 μ width which is not considered short length or width.

NC models the MOS transistor using the equations derived by Glasser and Dobberpuhl [Glasser 85] which is the same as equations (2.11), (2.12) and (2.13). The only short channel device effect NC accounts for is the channel modulation effect. The factor added to account for this is:

$$1 + \frac{V_{DS} - V_{D,sat}}{V_A - V_{D,sat}} \tag{2.14}$$

where

$$V_{D,sat} = \frac{V_{DS} - V_{TH}}{1 + \delta} \tag{2.15}$$

$$V_A = 5L\sqrt{\frac{N_A}{N_T}} \tag{2.16}$$

and $N_T = 10^{15} cm^{-3}$. This factor is derived empirically and one limitation is that it is hard to determine under what condition it is valid. SPICE uses a better approximation which is solved by Baum and Beneking [Baum 70] and can be found in [Divekar 88]. SPICE also includes the parasitic capacitor coupling between the gate, drain and source nodes. Including these capacitors will give a more accurate model of the MOS transistor and thus yield a better simulation result.

CIRCUIT ANALYSIS AND COMPUTATION

SPICE supports primarily three types of analysis nonlinear-dc, nonlinear-transient and ac small-signal analysis. The dc analysis finds the dc operating point of the circuit. This analysis is automatically performed before doing an transient or ac analysis. The purpose is to linearize (for transient) or find a small signal (for ac) model for the non-linear devices. After obtaining the linearize model for the circuit, the resultant linear circuit is analyzed over a time period which is specify by the user. For the ac analysis, the linearized circuit is analyzed over a frequency range rather than a time period. The dc analysis can also provide the user with dc transfer curves and initial conditions of the nodes for transient analysis. The transient analysis is used to compute the output nodes as a function of time over a user specify time period. SPICE is able to not only do nonlinear-dc, nonlinear-transient and linear-ac analysis but also special analysis including noise and temperature varying analysis. However, in our experiment using the CAM circuit, we do not use the special analysis.

SPICE analog model computes the terminal currents of a device from its terminal voltages. The terminal currents are actually a function of the terminal voltages. The matrix form of the equations is the large systems of simultaneous equations that need to be solved.

$$\left[A\right]\left[V_j\right]=\left[I_j\right] \tag{2.17}$$

Matrix A contains the node connections present in the circuit. Matrix V_j consists of the terminal voltages at a certain time step j. Matrix I_j consists of the terminal currents at a certain time step j. The value of I at the present time step, I_j is evaluated based on the terminal voltages at the previous time step, V_{j-1}. I_j is fed into V_j so that a new V_j is obtained. The evaluation process will continue until V_j is constant for each time step, i.e. $V_{j-1}=V_j=V_{j+1}$ or until the evaluation does not converge. The numerical method being used is based on the Newton-Raphson method.

NC analog model uses similar method as compare to SPICE analog model to evaluate the systems of terminal voltage and terminal current equations. The numerical method being used by the analog equation solver is also the Newton-Raphson method. One obvious difference between these two models is that NC analog model uses a simpler first order MOSFET device model while SPICE analog model uses a second order MOSFET device model. The MOS_PWL model, however, is entirely different from what we have seen before.

The execution of the NC program using MOS_PWL model consists of two steps, circuit analysis and computation. NC MOS_PWL model proceeds to analyze the circuit by breaking it down into stages. It assumes that on the average, currents flowing into the gate node of a MOS transistor is zero meaning that there will be no DC coupling between the stages. Each stage will have a portion of nodes from the original circuit. These nodes can be reached from each other by following only the source-drain paths without crossing any gates. These nodes are also the output nodes of that stage. Each transistor with their

source or drain node connected to an output node is part of that stage. A stage is a portion of the original circuit being bounded by the power source, ground and the gate of the transistor.

Figure 2 gives an example of a circuit after it has been partition into stages using the NC MOS_PWL model technique. Base on the description above and to understand how the partition is done this way, first notice that the source and drain node of transistors 1, 4, 5 and 8 does not cross any gate nodes. The drain node of transistor 1 and 5 is connected to the power source and the source node of transistor 4 and 8 are connected to ground. These are two of the three boundaries that define a stage. The other boundary is set at the gate of transistors 9 and 10 which is also connected to the source node of transistors 2 and 6 and drain node of transistors 3 and 7. Also, nodes a, b, c, d and q are output nodes of stages. Nodes a, b, c, and d can be reached from each other by following only the source-drain paths without crossing any gates. Therefore transistors 1, 2, 3, 4, 5, 6, 7 and 8 are in the same stage and transistors 9 and 10 are on another stage.

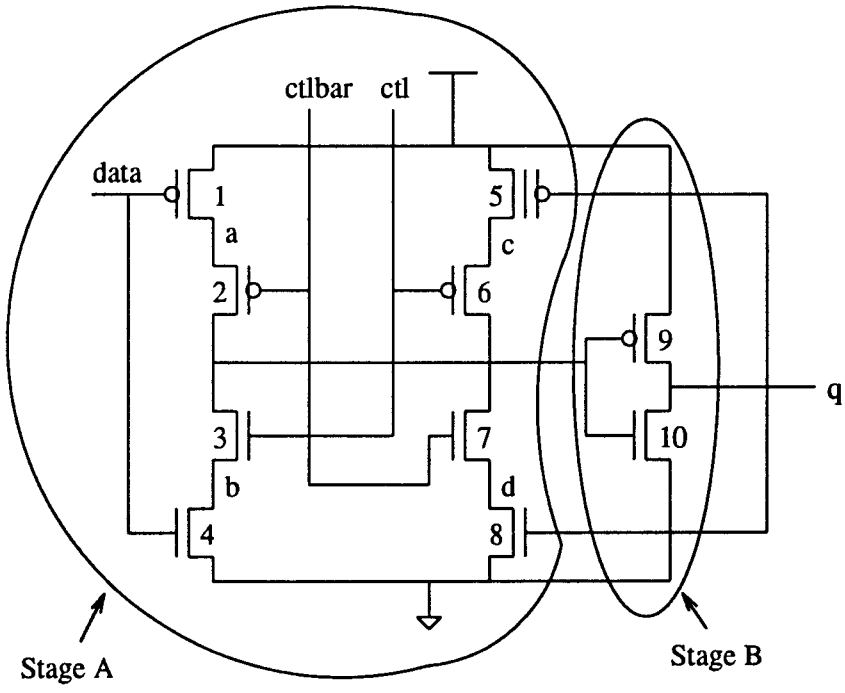

Figure 2. Example of stage partition in NC MOS_PWL model.

MOS_PWL model uses discrete event simulation to control the calculation of the behavior of the circuit. Discrete event simulation means that the computation of the node values is done only when there is a change in state of the input nodes from the previous state. The input nodes are those that are connected to the gates of a stage. The meaning of change in state in NC is different from the logic design's concept of change in state as being the change in input voltage. Beckett [Beckett 88] uses a simple example to illustrate this point. For a typical rising edge, the logic simulator will view it as one change in event but NC views it as two events. The derivative of this edge will result in a rise from zero to a positive value, stays at that value for the same amount of time as the rising time and then goes back to zero again because the rising edge is no longer rising but stays at a constant value. Inputs to the discrete event simulation have to be in the form of state valued function and the outputs are of the same form. Therefore voltages on the nodes are modeled as piece-wise linear (PWL) functions. NC generates the PWL functions by performing continuous time calculation on the nodes and then fitting the resulting curves with PWL forms. Continuous time calculation is defined as the calculation done forward in time from the present time point. NC has no way of knowing what the future input is going to be and therefore the output prediction is based on the predicted value of the input behavior. The following linear equation is used in the prediction of the state of each input node.

$$y = m(t - t_0) + c \tag{2.10}$$

where m is the slope, t_0 is the time of previous change and c is the intercept. As expected, this prediction will created an unbounded value when m is not zero and NC has to clip the value to between V_{DD} and ground.

The computation phase proceeds as follows. At each time point in the forecast range, the system of Kirchoff equations for each stage are solved using the same method employed by the analog models. The branch currents are computed using the equations

in the transistor model. The inputs of the transistor model include the source and drain node voltages, the forecast time, t, and the linear model of the gate node. The output of the transistor model is I_{DS} which is summed into an accumulator. Each node consists of an accumulator which is used to sum the node currents. When t and the linear model is put into (2.10), the gate voltage can be predicted. When all branch currents of the stage have been computed, together with the currents into the gate capacitance, the total branch currents for each node are sent back to the equation solver. The equation solver will continue to wait at each time point for branch currents to be sent back until the residual current vector falls below a preset value. When a complete set of output curves has been computed, the above process will halt. NC fits these curves using a curve fitter which classifies the curve with the number of inflection in the curve's second derivative and try to generate a fit using either 1, 2 or 3 line segments. Each segment is the best fit for the points in that segment. When the curve fitting is done, a constraint is placed to ensure continuity. If the new piece-wise linear model for a node will intersect the present PWL model at a future time point, the first event in the new PWL model will be delayed until that time. Therefore no overlapping of PWL models will occur.

Although this computation method used by MOS_PWL modeling needs less computation time than the fixed time step method used by SPICE in most cases, it can potentially use more computation time due to the fact that the assumption on which the input prediction is based on may change resulting in recomputation. NC uses three methods to prevent this from happening. First, extensive calculation is only likely to occur at the transition of the input nodes since event driven simulation is used. Second, when NC encounters an event on the input node that has already been included in the prediction, it does not compute the results again. Third, NC uses a calculation history to avoid redundant calculation.

The MOS equation solver uses numerical method which is based on the

Broyden's method [Avriel 76] rather than the Newton-Raphson method. The discussion of the detail difference between both methods is beyond our scope. However, it is worthwhile to mention that given the condition that a circuit is unlike an oscillator meaning its behavior is close to the prediction, the number of evaluation can be significantly reduced and thus requiring less simulation time.

PREDICTION

In the experiments that we are going to look at, in terms of simulation time, we are expecting SPICE to be the slowest and NC MOS_PWL model the fastest. SPICE would expect to be the slowest because of its fixed time step method of computation and the lengthy second order effect equations involved in the level 2 MOSFET model. The simulation time for using the NC analog model will be in between SPICE and NC MOS_PWL model. The reason is that NC analog model uses the similar fixed time step method of computation as SPICE but uses a simpler MOSFET model which is same as the one used by the NC MOS_PWL model. NC MOS_PWL model will require the least simulation time because of its prediction method of computation, partitioning its circuits into stages which greatly reduces the set of node equations to be solved and uses a simpler MOSFET model than SPICE. Although the MOSFET model used by NC is much simpler than the one used by SPICE, this model is sufficient to give a reasonable result for most of the CAM component circuits.

CHAPTER III

DIODE CONFIGURATION MOSFET PROBLEM

The circuit in Figure 3 shows a design of a priority resolver circuit which is used in the CAM architecture to resolve the multiple match situation. This circuit is chosen to test the performance of the SPICE analog model, NC analog model and NC MOS_PWL model in the three following aspects:

1. Computation time spent on this circuit.
2. The differences in output accuracy in terms of logic level and voltages.
3. Limitation on the types of circuit that can be simulated, especially circuits with transistors connected to behave like diodes.

When NC MOS_PWL model simulates a circuit, it breaks the circuit down into a set of stages and compute them separately. Also, this model is event driven and the number of events will affect the computing time. NC analog model and SPICE analog model uses the fixed time step method of calculation. We will expect that when these models simulate this circuit, the MOS_PWL model will need less computing time than the other two models. NC analog and SPICE analog model will take about the same computing time. This priority resolver is a good test in this aspect because it has four inputs which may or may not have events depending on the match signals.

In Figure 3, there are several MOS transistors with their gate and drain connected together. This type of MOS transistor will behave like a diode. NC analog model and SPICE analog model will have no problem in handling it because they place no restrictions on how the gate, drain and source of a MOS transistor can be connected. However, MOS transistors connected this way may create a problem for the NC MOS_PWL model

in differentiating different stages. Remember that a stage has the boundaries of the power source, ground and the gate of a transistor. Since NC MOS_PWL model requires that no gates can be crossed in a stage and paths are followed in the source to drain fashion, by connecting the drain and gate together, this requirement has been directly violated and this may lead to a wrong output result.

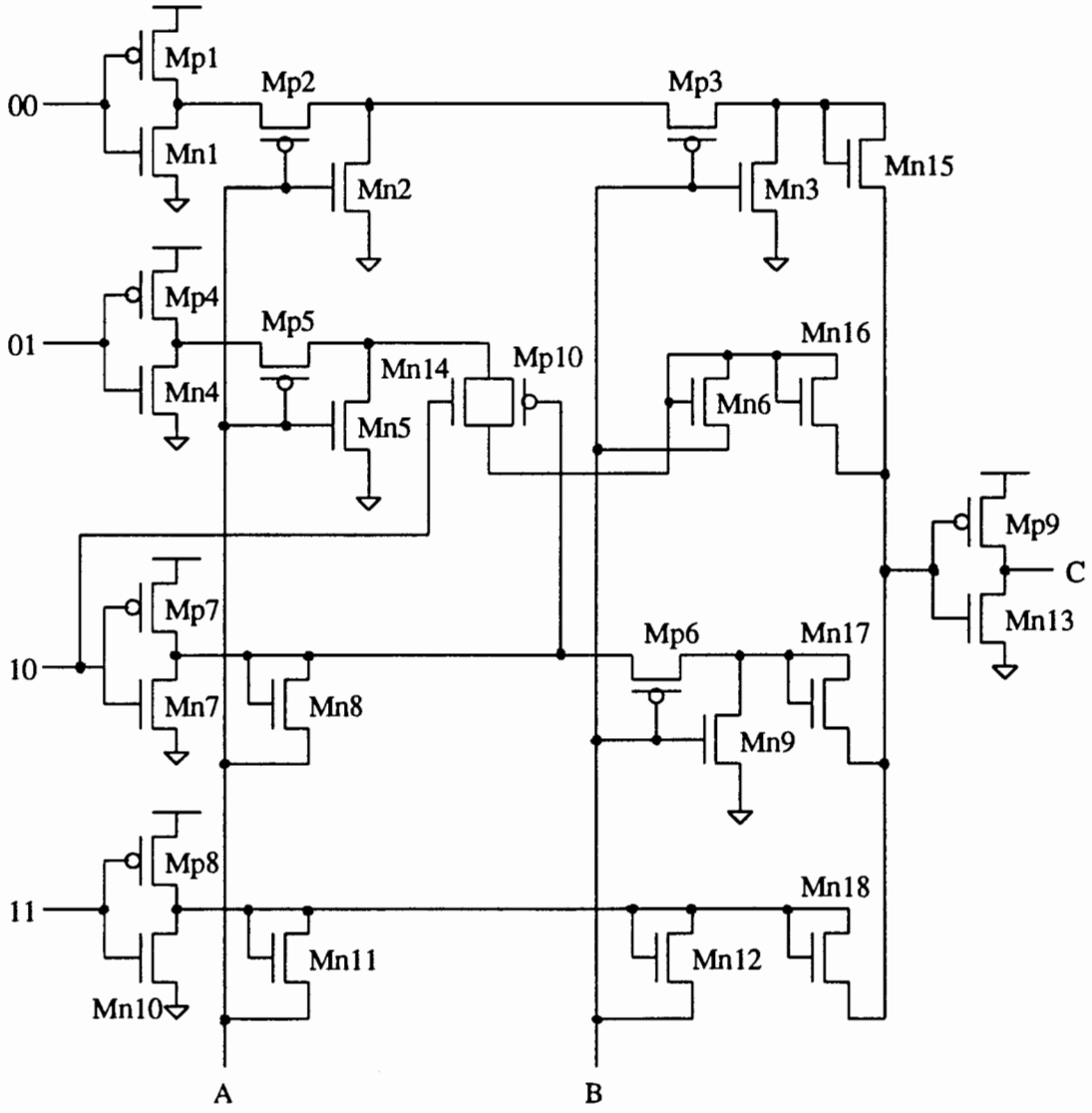

Figure 3. A priority resolver circuit.

CIRCUIT DESCRIPTION

Figure 4 shows the logical representation of the priority resolver circuit which will make the understanding of this circuit easier. In a multiple match situation, the priority resolver is used to select the match line with the highest priority and returns the address of the chosen match line. The four inputs in the circuit are oriented in such a way that 00 has the lowest priority and 11 has the highest. The address of the chosen match line will show up in nodes A and B. The output at node C tells if there has been a match signal at any input.

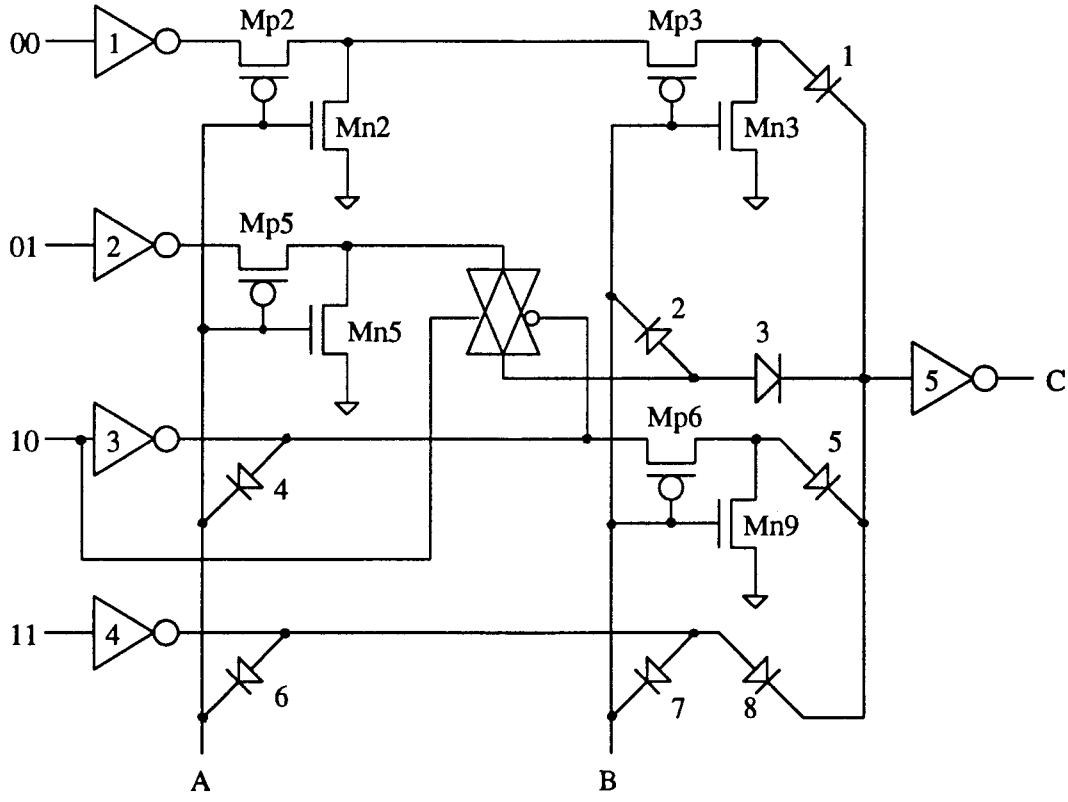

Figure 4. A logical diagram of the priority resolver circuit.

The operation of the priority resolver is actually quite simple. Assume that there are 3 match signals coming into 00, 10 and 11 lines. Ground potential when applied to any of the inputs will be interpreted as a match signal. We can expect nodes A and B to

be both logic high because 11 has the highest priority among 00, 10 and 11. When match signals come into lines 00, 10 and 11, the output nodes at the corresponding inverters will start charging to 5 volts. Once diode 4 starts conducting, the potential at node A will start to raise and as a result transistor Mp5 is turning off. Also, as diode 6 is starting to conduct, so is diode 7 and as a result transistor Mp6 and Mp3 will be turned off. The logic value at nodes A and B are now '1' and '1' which is the address of the highest priority match signal. Also, node C will be at ground potential indicating that there is at least one match signal on any of the four inputs.

In the original design [Koo 70] of the priority resolver circuit, the transmission gate is not included. When that is the case, the output values at nodes A and B are not quite correct when only two match signals appear at 01 and 10. When line 01 has a match signal, the output of the inverter is charging towards 5 volts. Since initially nodes A and B are at ground potential, transistor Mp5 is conducting and diode 2 will start to conduct and node B is no longer at ground potential. Although at the same time diode 4 is starting to conduct and is trying to turn Mp5 off so that no charge comes to diode 2, diode 4 is just not fast enough to do that. Depending on how fast inverters 2 and 3 can charge up, node B may be able to accumulate enough charge before transistor Mp5 is turned off to be interpreted as a logic '1' instead of a logic '0'. When this happens, nodes A and B will be '1' and '1' instead of '1' and '0' which is the wrong result. This competition situation is not desired in a circuit design and needs to be fixed. To solve this problem, a transmission gate is added between the drain node of transistor Mn5 and the diode 2. The gate of the n-type transistor in the transmission gate is controlled by the 10 line and the gate of the p-type transistor is controlled by the output node of the 10 line inverter. In this way, when a match signal comes into the 10 line (ground potential), the transmission gate is immediately turn off and so no charge can reach diode 2 and node B will stay at the initial ground potential. Diode 4 will not be affected by the transmission gate and will still charge up node A and therefore node A is at logic high and node B is at logic low

which is the desired result.

In a CAM system, there are usually more than four match lines. Since this design only takes four match lines at a time, it is then used as a function block in a larger system. To resolve more than four match lines, it is necessary only to cascade function blocks together. For example, in a 16 match line case, 5 function blocks are used where 4 function blocks are used to resolve the 16 match lines and the outputs are cascaded into the fifth one where only the highest priority match line will be pick.

SIMULATION RESULTS

The simulating tools that originally pick up the competition between signal problem are the NC MOS_PWL and analog models. When this same circuit is simulated using the SPICE analog model, the same problem is also discovered. However, a switch-level simulator like the one that Ruan [Ruan 88] described may not pick up the problem. The reason is that there are only three kinds of voltage defined V_{off}, V_{on} and V_{dd} and two kinds of current defined I_{part} and I_{full}. When the current reaches I_{part}, the V_{on} is reached and when the current reaches I_{full}, V_{dd} is reached. Using this definition, there is no way of knowing how much charge has been gone to node B. While using the three models to simulate, we know exactly how much charge is in node B.

NC analog model, MOS_PWL model and SPICE analog model are used to simulate the priority resolver circuit. Since by default NC uses a few different model parameter values compare to SPICE, the parameters in the MOS transistor model in these three types of simulation have been forced to be the same. The simulation results are shown in Tables III, IV and V. In these results, NC MOS_PWL model consumes the least run time for any three cases. The way NC MOS_PWL model is able to cut down on run time is discussed in the section on NC modeling. Due to the similar ways of both NC analog model and SPICE analog model simulate circuits, the run time are fairly close to each

other.

The output voltage at nodes A and B in these three types of simulation are around 3.2 to 3.4 volts. In the ideal situation, the output voltage is expected to be $V_{dd}-V_{To}$ where V_{dd} is the voltage source and V_{T0} is the threshold voltage for the transistor. However, the source to substrate voltage V_{SB} is usually not at zero for all transistors and the body effect must be taken into account. With this in mind, let us do a simple derivation of the theoretical value of the voltage at nodes A and B.

TABLE III

SIMULATION RESULTS OF THE PRIORITY RESOLVER WITH FOUR MATCHES

Simulation Tools	Total Runtime (s)	BitA (v)	BitB (v)	Output Capacitance (pf)
NC MOS_PWL	37.5	4.447	3.472	.1
NC Analog	452.4	3.27	3.27	.1
SPICE Analog	420.4	3.449	3.426	.1

TABLE IV

SIMULATION RESULTS OF THE PRIORITY RESOLVER WITH 00 AND 10 MATCHES

Simulation Tools	Total Runtime (s)	BitA (v)	Output Capacitance (pf)
NC MOS_PWL	42.4	3.262	.1
NC Analog	426.8	3.269	.1
SPICE Analog	320.8	3.37	.1

TABLE V

SIMULATION RESULTS OF THE PRIORITY RESOLVER WITH NO MATCHES

Simulation Tools	Total Runtime (s)	Output Capacitance (pf)
NC MOS_PWL	7.3	.1
NC Analog	75.4	.1
SPICE Analog	110.6	.1

The threshold voltage for a p-type transistor with the body effect is given as

$$V_{Tp} = V_{Tp0} - \gamma_p \left(\sqrt{-V_{SB} + 2(\phi_{Fn})} - \sqrt{2\phi_{Fn}} \right) \tag{3.1}$$

and for n-type transistor is

$$V_{Tn} = V_{Tn0} + \gamma_n \left(\sqrt{V_{SB} + 2(\phi_{Fp})} - \sqrt{2\phi_{Fp}} \right) \tag{3.2}$$

where $\gamma_{n,p}$ are the body effect coefficients, $V_{Tp,n0}$ are the threshold voltages when V_{SB}=0 and $\phi_{Fn,Fp}$ are the fermi potentials. Since only n-type transistors are involved in nodes A and B, only (3.2) is going to be used. ϕ_{Fp} is defined as

$$\phi_{Fp} = -\kappa \frac{T}{q} \ln \frac{N_A}{n_i} \tag{3.3}$$

where κ is the Boltzman's constant, T is the temperature, q is the electronic charge, N_D is the substrate doping and n_i is the intrinsic carrier concentration. The value of κ, q and n_i are listed in Table II on page 8. The value of N_A used in the SPICE analog model is 3.0223×10^{16} and 300 K is used as the value for the temperature. Substitute these values into (3.3) and we get $\phi_{Fp} = -0.376$. The body effect coefficient, γ_n is defined as

$$\gamma_n = \frac{\sqrt{2\varepsilon_{si} q N_A (1.0 \times 10^6)}}{C_{OX}} \tag{3.4}$$

where

$$C_{OX} = \frac{\varepsilon_{ox}}{t_{ox}} \tag{3.5}$$

The value of ε_{ox} is listed in Table II and 2.28×10^{-8} m is used as the value of t_{ox}. Substituting all these values into equation (3.5) we have $C_{OX} = 1.513 \times 10^{-3}$. γ_n can now be computed and has a value of 0.6632. Along with $V_{SB} = 4.0v$ and $V_{Tn0} = 0.972v$, the new threshold voltage turns out to be $V_{Tn} = 1.592v$. Therefore the output voltage at nodes A and B should be at around 3.408 volts. Looking at the results in Table III, all three models have results in agreement with the hand calculated results. The slight difference

may be due to the different V_{SB} being used or the flat band voltage, V_{FB} which is used in the prediction of V_{Tn0}.

In the priority resolver circuit, there are quite a few MOS transistors that are connected to behave like a diode. In the introduction of this section, it is mentioned that this type of transistors may create problems for the NC MOS_PWL model during simulation. Since NC MOS_PWL model uses discrete event driven scheme in its circuit analysis, the input which is a change in state or also known as an event can only occur at the gate node. When the gate node and drain node are shorted together, it is possible that NC MOS_PWL model will complained about no event at the gate node. As the simulation results turn out, NC MOS_PWL model handle this type of diode configuration transistors quite well. NC MOS_PWL model is able to give the correct result because every gate of the diode configuration transistors in the priority resolver is going to have at least one event. For example in Figure 4, diode 1 will get an event from inverter 1, diode 2 and 3 will get an event from inverter 2, diode 4 and 5 will get an event from inverter 3 and diode 6, 7 and 8 will get event from inverter 4. However, let us look at the circuit in Figure 5 which will show the no event problem.

TRANSDUCER SOLUTION

The circuit in Figure 5 is called a reference voltage generator because ideally, no matter what the initial voltage at node out is, the voltage at this node is going to approach $\frac{V_{dd}}{2}$ volts and this voltage can act as a reference voltage to other parts of a circuit. This type of circuit can be found in the sense amplifier part of the CAM system.

This circuit is going to be an ultimate test for the NC MOS_PWL model because both the n and p-type transistors are connected to behave like a diode. The box with a X in it is a transducer. The transducer equates the drain node of each transistor to its gate node. It also equates the derivative of drain node to the derivative of the gate node. NC

analog model and SPICE analog model have no problem in figuring out what the circuit does without the transducer. However, without the transducers, NC MOS_PWL model will complain about no trigger in the circuit and the simulation will halt. There is no trigger because the gate of the transistor has been shorted to the source and since NC MOS_PWL model uses event driven algorithm, it is waiting for an event to occur at the gate. As mention in Chapter II, an event would occur when there is a change in slope in the input gate.

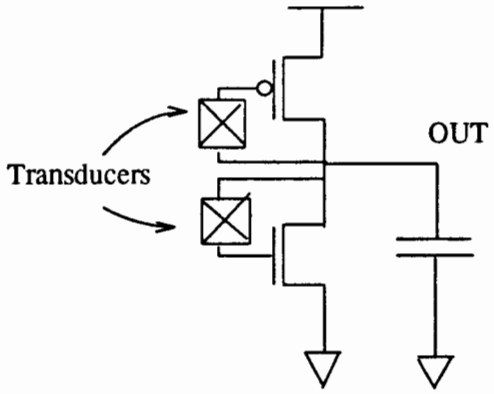

Figure 5. A reference voltage generator circuit.

By equating the derivative of drain node to the derivative of the gate node, we make sure that there will be at least one trigger at the gate node. In the transistor model of MOS_PWL the only trigger found is at the gate node. Also MOS_PWL model requires that there is to be at least one trigger at any input node in a given circuit. With the transducers added, the simulation result shows that the output node is oscillating rapidly between 0 and 5 volts. No matter what the initial value at the output node is, every time when the output node is trying to charge the capacitor up, the opposite transistor will turn on and discharge the capacitor and vice versa. Therefore the output and input node are both constantly changing. If the forecast interval used by the MOS_PWL model is too large, the prediction will not be correct and the model is constanting reevaluating the output node. Thus the MOS_PWL model could not settle down to a

value. The forecast interval has to be decreased in order to obtain a reasonable prediction. The default forecast interval *frcst*, is 1 microsecond and this value has to be shorten to 1 nanosecond before a reasonable prediction is obtained. A shorter time step is also needed so that more samples are taken. Therefore the *mosstep* which is the time step used for calculating the forecast of the value at the nodes has to be decreased too.

CHAPTER IV

MODEL BUILDING IN NC

This chapter uses the four operations performed by the three state CAM cell with sense amplifier circuit as examples to test the performance of SPICE analog, NC analog model and NC MOS_PWL model in the following three aspects:

1. The computation time the simulation tools take to simulate this circuit.
2. The effect of not considering capacitor coupling between the gate, drain, source and bulk of the MOS transistor in NC's calculation has on the encouraging or misleading simulation results.
3. The ability to build one's own NC circuit model using the mathematical equations derived from analyzing a particular circuit and simulate it on the model level (NC) rather than on the circuit level (SPICE).

The computation time that the simulation tools take in simulating circuits has been an important area of study in this research. In this circuit, the size (in terms of numbers of transistors) in the circuit to be simulated can vary depending on which operation the CAM cell is performing. When the CAM cell is performing write and match operations, the sense amplifier is disconnected by disabling the enable line in the sense amplifier circuit. The sense amplifier comes into action in the read and refresh cycle. Therefore it would be interesting to see how the simulation tools respond to different circuit size and which tool has an advantage over small or large circuit. The four operations that CAM cell performs will be discussed in detail and the sequence of events of the operation will also be discussed.

The CAM cell is able to store three logical states, logic '1', logic '0' and DON'T

CARE which is also known as the mask state. The mask state is another interesting topic and is discussed on Chapter V. Storing desired logic levels into the cell is discussed in the write section. This same section also touches on one of the important differences between the way NC and SPICE models MOS transistor. SPICE differs from NC in that SPICE takes the MOS capacitance into consideration during its computation. The MOS transistor has capacitors coupling between the gate, drain, source and bulk depending on which region of operation the transistor is on at a particular time. NC on the other hand does not include these capacitances in its calculation unless the user specifies them. These coupling capacitors can make a difference in the output values being computed. In the worst case, the output values can be wrongly reported as in the case of simulating the write operation using NC with the CAM cell circuit.

The CAM cell is a dynamic storage device and information is stored on the gate capacitor of the transistor. This type of dynamic cell requires fewer transistors to build than the static storage device but it suffers a drawback of needing to refresh periodically. Charge can leak through the p-n junction of the source or drain node in the pass transistor which is used to isolate the gate of the storing transistors from other parts of the circuit. In our design, the storage node is isolated using a transmission gate rather than a pass transistor and a detail discussion of the reason is given in the section on write operation. As logic state is defined by how much voltage the stored node has, it is necessary to compute the leakage current and the time period for the refresh cycle in order to avoid the situation where a logic state is misinterpreted. Uyemura [Uyemura 88] developed a four electrical component p-n junction diode model to replace the actual p-n junction diode and derive the leakage current equation. This equation is the basis of our NC transmission gate charge leakage model and an indepth discussion will be given on the derivation of the leakage current equation.

Through building this charge leakage model, the CAM cell circuit provides us an

opportunity to demonstrate the advantage of using NC over SPICE in that case. As we have mentioned before, the circuit components in NC are all built as models using the C programming language. Whenever a user encounters with a new circuit component and derived a set of equations sufficient to describe that component, base on those equations the user can build a model of that component using the C programming language. The model can then be placed inside the NC library and be used as circuit components. SPICE on the other hand does not allow user to do that. The circuit components in SPICE are limited to the traditional electrical components which have already been built into SPICE, no new ones can be created by the user. It is true that the user can build the desired component as a subcircuit using the traditional electrical components and use that subcircuit as a model for the component. However, the user is not able to build the model directly using the derived equations which describes the component. Thus NC has greater flexibility for the user in this aspect.

CIRCUIT DESCRIPTION AND ANALYSIS

The four basic operations: read, write, match and refresh that the CAM cell performs are discussed here. The CAM cell which is shown in Figure 8 is originated from the design of Mundy *et al.* [Mundy 72]. Our design of the sense amplifier circuit is shown in Figure 6. Both circuits are actually connected together because they have common bit and bitbar lines.

Read Operation

Before the read operation is discussed, the function of the sense amplifier needs to be described first. The sense amplifier plays an important part in the read operation. It is designed with a differential amplifier circuit (transistors Mn1, Mn2, Mn3, Mn4 and Mn5) and 2 pairs of non-inverting inverter restorers (Mp1, Mn6, Mp2, Mn7 and Mp3, Mn8, Mp4, Mn9). The differential amplifier detects small voltage differences between inputs

bit and bitbar and the voltage is amplify through the two inverter restorers. The results of the amplified signal are shown at bitout for the bit input and bitbout for the bitbar input. When the sense amplifier circuit is simulated using SPICE, a plot is shown in Figure 7.

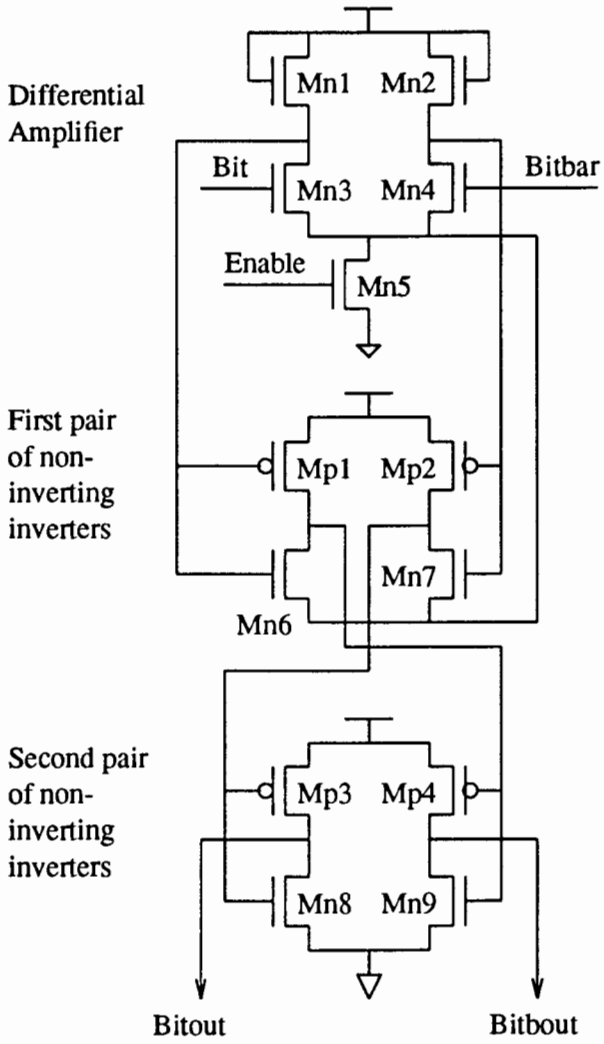

Figure 6. A sense amplifier circuit.

Figure 7 demonstrates how sensitive the sense amplifier is to small voltage differences. The values that inputs bit and bitbar are used to produce this plot are 2.53 and 2.47 respectively. The sense amplifier output for the 2.53 volts is node bitout in the figure. These two values are chosen to show that the sense amplifier can immediately pick up an minimum imbalance of 0.06 volts swing between 2.5 volts of the input vol-

tages and restores 2.53 volts to 5 volts and discharges 2.47 volts to 0 volts. It does not take the sense amplifier very long to accomplish this. Notice that bitout is restored to almost 5 volts and bitbout to 0.2 volts in 0.5 ns. However at 0.3 ns, we can already recognized bitout as logic '1' and bitbout as logic '0'.

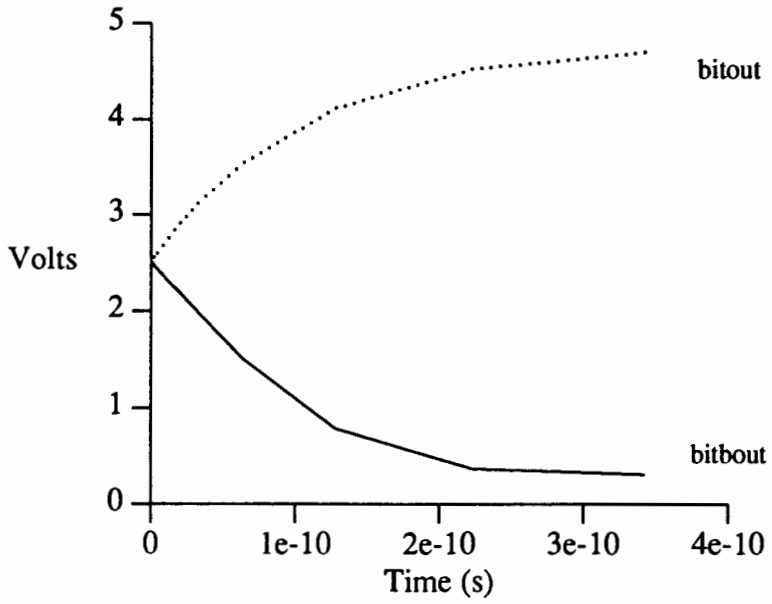

Figure 7. Sense Amplifier Voltage Sensitivity.

Although in actual usage these two input voltages are never that closed to each other, the simulation gives us an idea of how sensitive the sense amplifier circuit is. In actual cases, the voltage levels retrieved from the cell are more like 3.5 volts and 1.5 volts. The sense amplifier circuit does not have to wait for the output lines to go up to 3.5 volts in order to respond. As soon as the signal is above about 2.53 volts and the other line is below 2.5 volts, the sense amplifier is able to read from the cell and the results can be shown. The assist of very fast sensing from the sense amplifier gives to the read operation dramatically reduces the read cycle time.

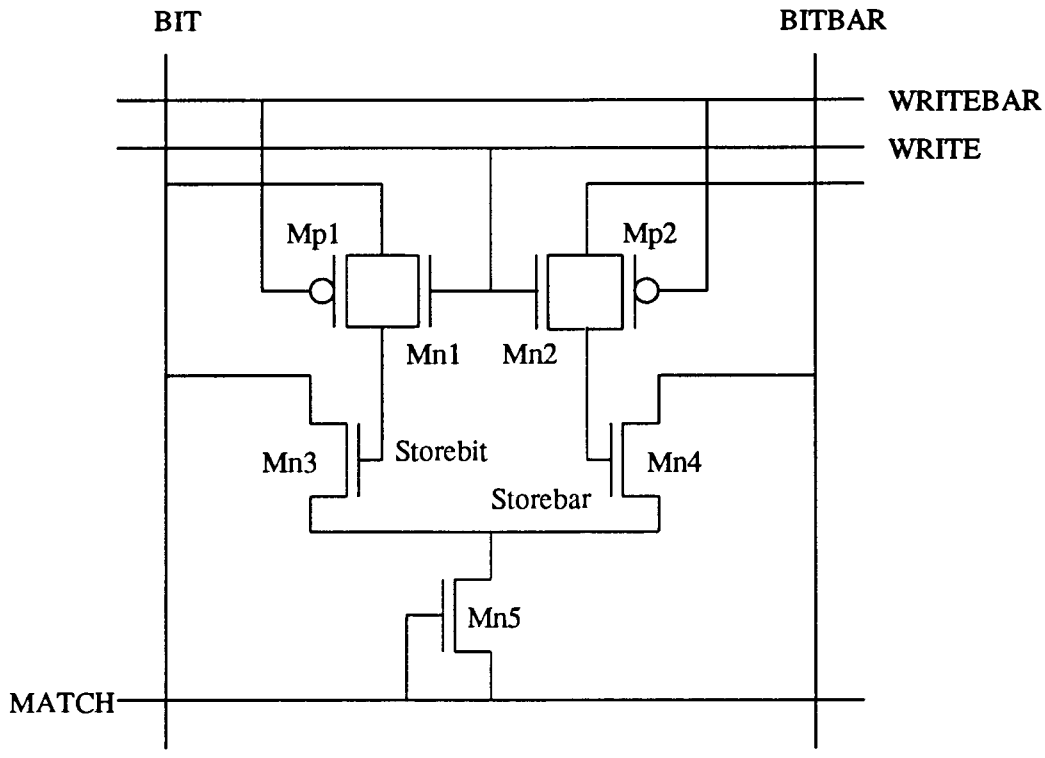

Figure 8. A seven transistor CAM cell circuit.

The function of the read operation is to retrieve data from the cell and restore the data using the sense amplifier. During the read operation, the enable line in the sense amplifier is first charged to logic high so that it comes into action. The bit and bitbar lines are both lowered to logic low. The match line is raised to logic high. The write line is set to logic low and the writebar line to logic high so that the charge that is stored in the gate capacitance of Mn3 and Mn4 in Figure 8 does not escape through Mn1, Mn2, Mp1 and Mp2 but only through Mn3 and Mn4. Since one of the two values stored in the gate capacitance of Mn3 and Mn4 has to be logic high, one of them must be ON and the charge from the match line is going to charge up the corresponding bit line. The sense amplifier can then detect the differences between bit and bitbar lines and determine the value that is stored on the gate capacitance of Mn3 and Mn4.

Read Operation Simulation Results

Figures 9, 10 and 11 shows the simulation plots of the read operation using the SPICE analog, NC analog and MOS_PWL models respectively. Initially node storebit in the CAM cell has 0.5 volts and storebar has 5 volts. Bitout and Bitbout are both initialized to 2.5 volts at the beginning of the simulation so that the output difference between

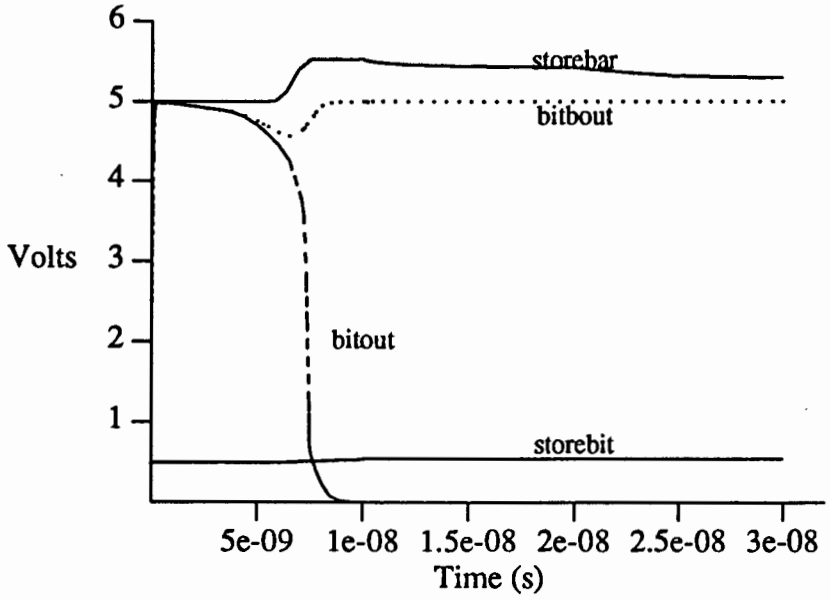

Figure 9. Read operation simulation using SPICE analog model

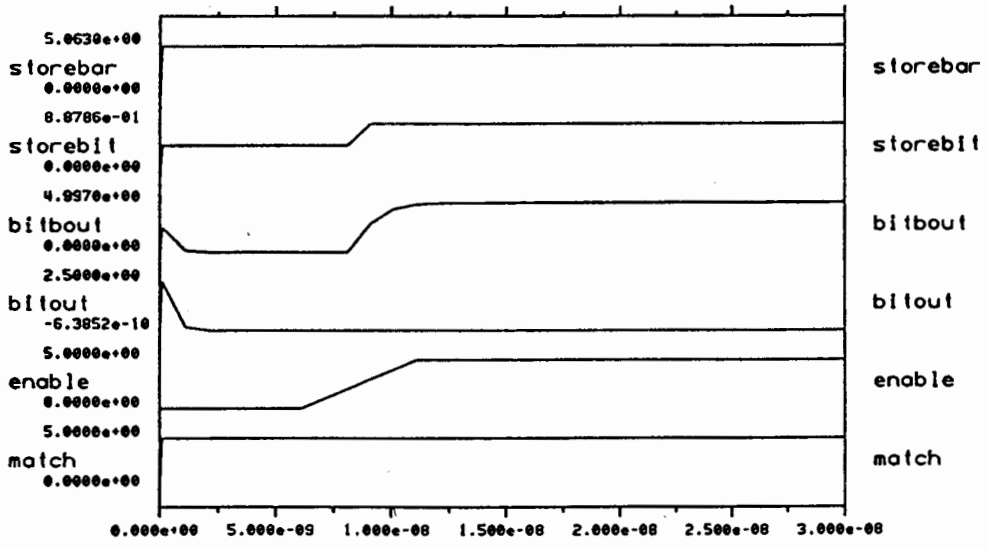

Figure 10. Read operation simulation using NC analog model

them can be significantly seen. The match line starts out at 5 volts and the enable line in the sense amplifier is initially at 0 volts meaning that it is disable. When the enable line starts to charge from 0 volts at 5 ns and reaches 5 volts at 10ns, bitout starts to discharge which indicates that a logic '0' has been read. In the NC simulation result, since the sense amplifier is disable, both bitout and bitbout lines are at low potential initially because of the non-inverting inverters. When the enable line starts to charge from 0 volts at 5 ns and reach 5 volts at 10ns, bitbout line starts to charge to 5 volts indicating that a logic '1' has been read at the bitbar line and bitout line stays at low indicating that a logic '0' has been read at the bit line. Table VI shows the simulation time taken by each simulation model.

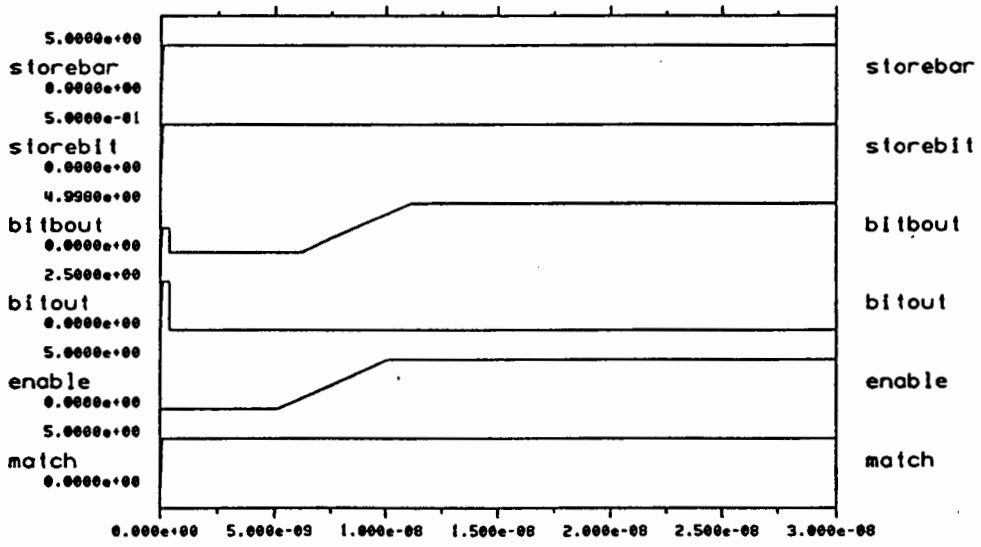

Figure 11. Read operation simulation using NC MOS_PWL model

TABLE VI

SIMULATION TIME OF THE READ OPERATION

Simulation Tools	Total Runtime (s)	Output Capacitance (pf)
NC MOS_PWL	8.8	.05
NC Analog	765.8	.05
SPICE Analog	162.8	.05

Write Operation

Another operation that the CAM cell performs is the write operation which is storing new data into the cell. During the write cycle, the enable line in the sense amplifier circuit is lowered to logic low so that the CAM cell and the sense amplifier are independent of each other. The desired logic values are then placed on the bit and bitbar lines and waited for the write and writebar lines to be ready. When the write line is charged up to 5 volts and the writebar line is lowered to zero meaning that transistors Mn1, Mn2, Mp1 and Mp2 are conducting, the data on either bit or bitbar line is ready to be written into the cell. If the desired logic level at the bitbar line is high, charge will pass through the transmission gate Mn2 and Mp2 and stored at the gate capacitance of Mn4. Since the bitbar line is high, bit line must be low and any initial charge that the gate of Mn3 has will escape through the conducting transmission gate Mn1 and Mp1 to the bit line. This is similar to writing a logic zero to the gate of Mn3. A few nanoseconds later, the write line is lowered and the writebar line is raised again so that the charge on the gate capacitance of Mn3 and Mn4 are trapped. In this way, data is written into the CAM cell.

Write Operation Simulation Results

This Mundy CAM cell suffers from a problem in the write operation [Sodini 89]. The charge stored on the gate capacitance of either Mn3 or Mn4 are supposed to stay there when the write and writebar lines are disabled and the bitbar line is lowered. However this does not happen in this CAM cell write operation. As the bitbar line is lowered, the charge are lost as well. Figure 12 shows the plot of SPICE analog model simulation of the write operation. At 25 ns the bitbar line is lowered and the charge at Mn4 begins to draft away and left with about 2 volts. Wade and sodini [Sodini 89] explains that as the write and writebar lines are disabled, the gate of Mn4 is capacitively coupled to ground. Because a high potential is stored at the gate of Mn4, when the bitbar line is lowered, this

node experiences more capacitive coupling to ground.

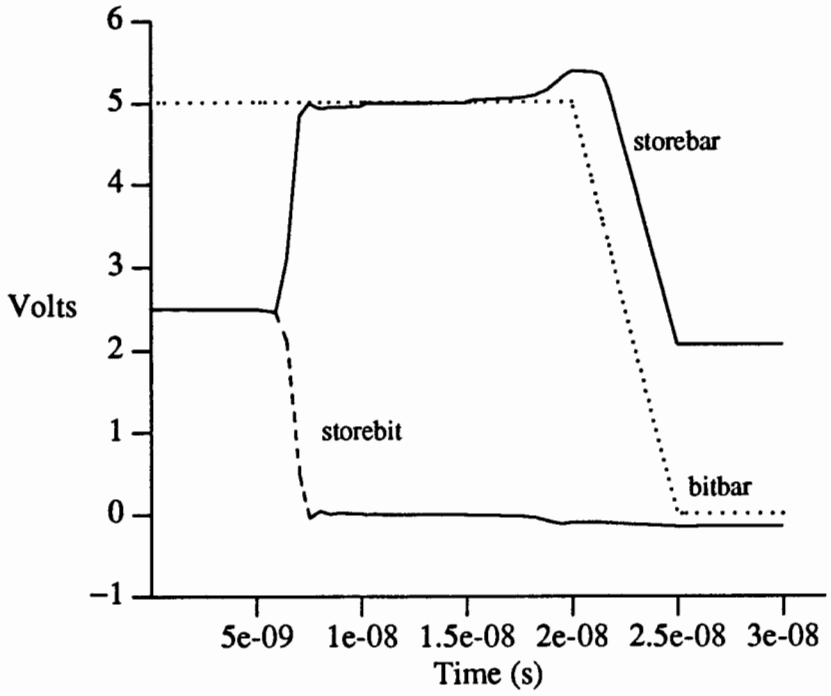

Figure 12. SPICE analog model simulation of the Write operation.

NC analog and MOS_PWL models does not pick up this problem. When the bit-bar line is lowered, NC reports that the charge on the gate of Mn4 is still at 5 volts. The reason that NC is not able to pick up this problem is that NC does not take the capacitive coupling between MOSFET's gate, drain, source and bulk into consideration in its computation. These capacitive couplings can make significant differences in the simulation results. Let's take the three operation regions of the transistor as an example. In the cut-off region, there is the gate to bulk capacitive coupling. In the linear region, there are gate to source and date to drain capacitive couplings. In the saturation region, there is the gate to source capacitive coupling. These capacitances when ignored can result in incorrect simulation results. These capacitances can be included in the NC calculation by the user using the CAP() model in NC. The nodecap function is not able to be used because one of its terminals has to connect to ground. The CAP() model can only be

used in the NC analog model meaning that we are unable to add these interconnecting capacitances into the MOS_PWL model. An inconvenience for the user would be that the user has to calculate these values manually. Table VII shows the simulation time taken by each simulation model.

TABLE VII

SIMULATION TIME OF THE WRITE OPERATION

Simulation Tools	Total Runtime (s)	Output Capacitance (pf)
NC MOS_PWL	13.3	.05
NC Analog	11.3	.05
SPICE Analog	104.8	.05

In Figure 8, the CAM cell consists of seven MOS transistors. The designs from [Mundy 72] and [Sodini 89] both have five n-type transistors connected in exactly the same fashion as Figure 8 shows but without the two p-type transistors. In their case, information coming from the bit and bitbar lines is passed to the gate of Mn3 and Mn4 via pass transistors Mn1 and Mn2 respectively. In our case, information is passed via transmission gates. The reason for using transmission gates instead of pass transistors is that voltages that pass through a pass transistor will be lost by at least a threshold voltage of the transistor. Together with the 'body effect' γ parameter, the logic '1' is represented by about 3 volts. If we assume that we have a source of 5 volts coming into the bit line, the voltage that will be stored on the gate capacitance of Mn3 is going to be roughly about 3.4 volts. On the other hand, with the transmission gate as the passing device, no body effect problem occurs and Mn3 will stored 5 volts. Whether storing 3.4 volts or 5 volts in the cell does not alter the performance of the read, write and match operation. However, since the cell needs to be refresh constantly, storing 3.4 volts in the cell makes the cell needing to fresh more often than storing 5 volts. Thus it makes the performance of the cell less efficient because many cycles are wasted on doing refreshes.

Match Operation

The next operation that the CAM cell does is the match operation which is comparing external data to the ones being stored in the cell to see if they match. During the match operation, the enable line is first lowered to logic low so as to disconnect the sense amplifier from the CAM cell. Since the search does not concern transistors Mn1, Mn2, Mp1 and Mp2, the write line is lowered to zero volts and the writebar line is raised to 5 volts. The next step is to precharge the match line. To do that, the bit and bitbar lines are raised to 5 volts which ensures that after precharging the match line to 5 volts, the charge in the match line is not going to escape to either the bit or bitbar line since either Mn3 and Mn4 is ON. The desired logic level is then placed on the bit and bitbar lines. If the logic level at the bit line is low and the voltage stored at the gate of Mn3 is 5 volts, current is going to flow through the conducting Mn3 resulting in a discharge on the match line which shows a mismatch. If the bit lines connected to each of the conducting transistors remain high, no current will flow and a match is indicated.

Match Operation Simulation Results

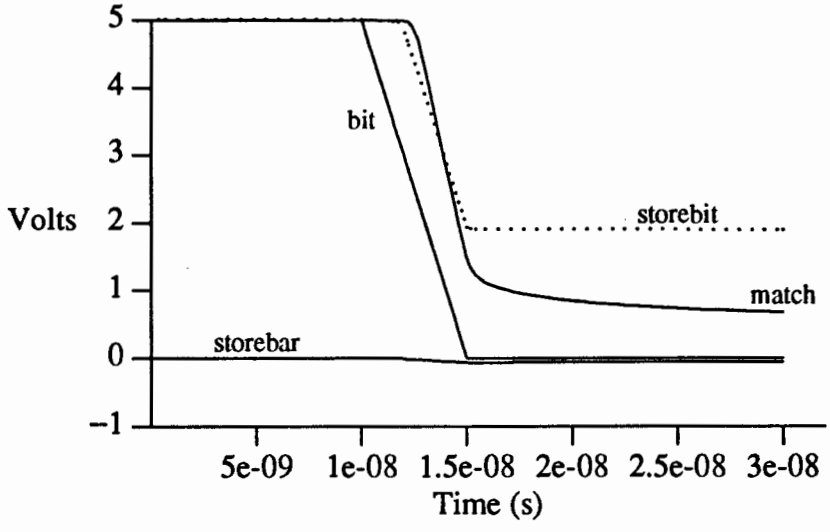

Figure 13. SPICE analog model simulation of the Match operation.

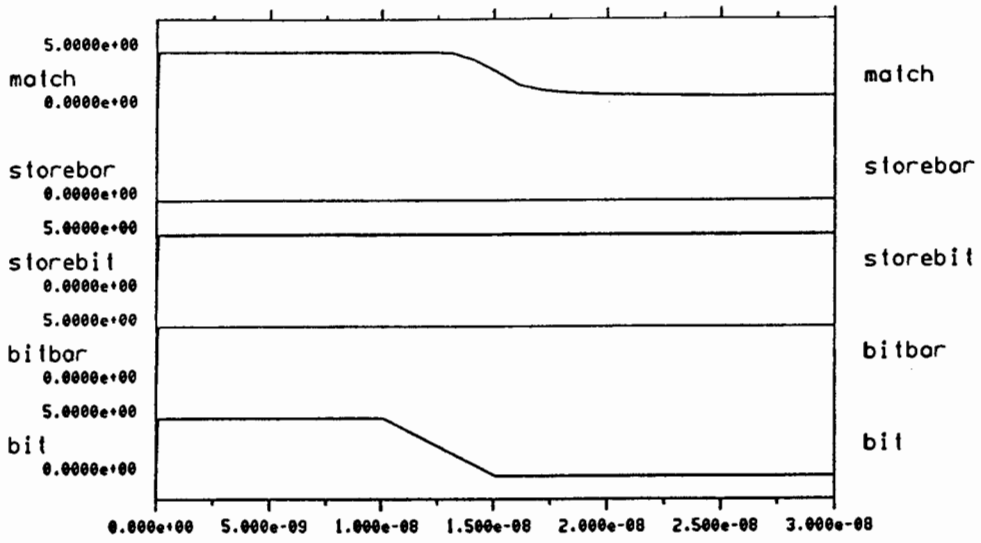

Figure 14. NC analog model simulation of the Match operation.

The simulations shows correct results and Table VIII shows the simulation time of using the SPICE analog and NC analog model.

TABLE VIII

SIMULATION TIME OF THE MATCH OPERATION

Simulation Tools	Total Runtime (s)	Output Capacitance (pf)
NC Analog	46.3	.05
SPICE Analog	100.7	.05

NC MOS_PWL model, however is not able to simulate this match operation because of the diode configuration transistor Mn5. The solution to this problem is the same as the transducer solution that we suggested in Chapter III.

BUILDING THE CHARGE LEAKAGE MODEL

The charge that is stored on gate capacitances in dynamic circuits exhibits charge leakage problem. Charge can leak through the p-n junction of both n and p type transistors and in the CAM cell, charge is leaking through the transmission gate. The leakage

circuit is shown in Figure 15. C_g represents the gate capacitance of either Mn3 or Mn4 in the CAM cell.

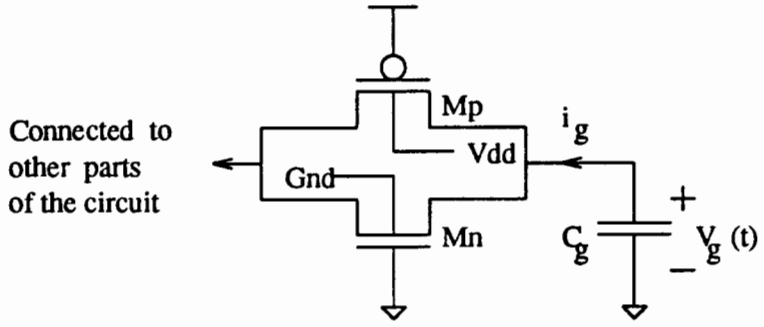

Figure 15. Transmission gate leakage circuit.

Two p-n junctions in this circuit are of interest. The first one is the junction formed between the p+ field implant and the n tub of the p-channel transistor Mp. The other junction is formed between the n+ regions and the p-type substrate of the n-channel transistor Mn. The n-tub bulk of Mp is connected to Vdd and the p-type substrate of Mn to ground. The p-n junction can be represented by diodes and the p-n junction leakage model circuit is shown in Figure 16.

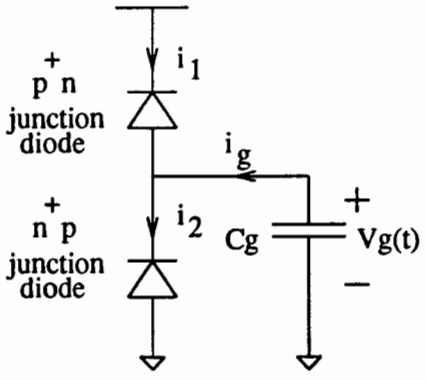

Figure 16. p-n diode model of charge leakage circuit.

To avoid confusion, we would name the p^+n junction to be junction 1 and n^+p junction as junction 2.

A high level analysis of this leakage problem is done by Uyemura [Uyemura 88] who replaces the p-n junction diodes by a four components model, an ideal diode, a big resistor, a reverse current source and a junction capacitor connected in parallel. Figure 17 shows the new diode model circuit.

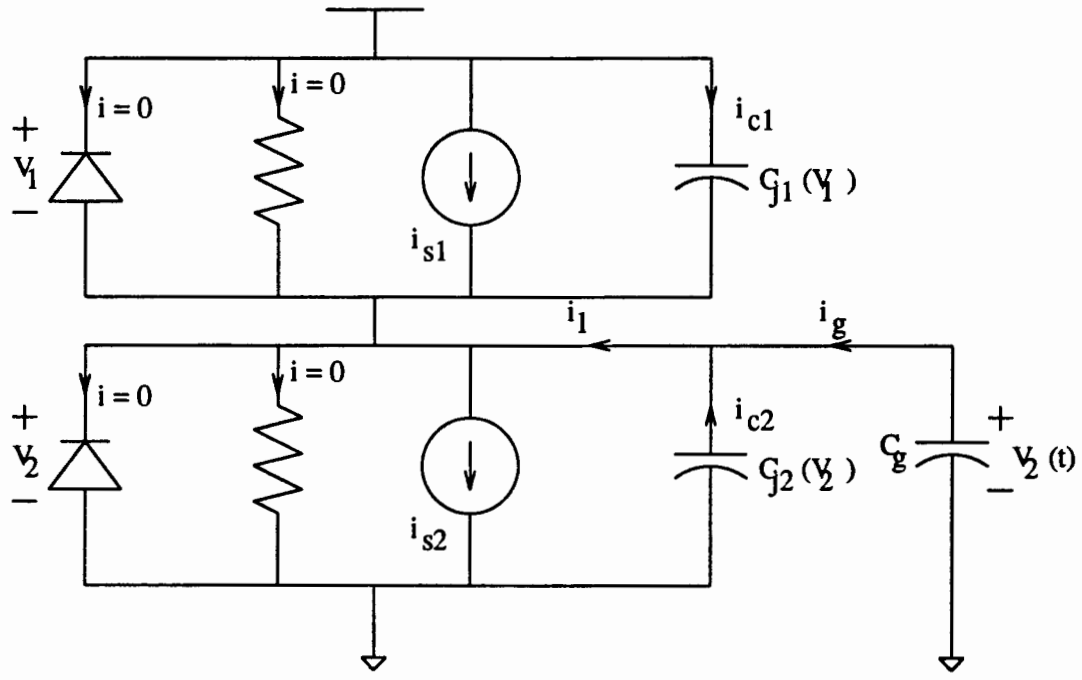

Figure 17. New diode model for the charge leakage circuit.

With this model, an equation of the leakage current in terms of V_2 can be found. With this leakage current and V_2 present, an numerical analysis using NC is possible to find the period of the refresh cycle.

Although the diode model consists of four elements, the ideal diode and the resistor are ignored during the calculation. The fact that the diode is ideal means that no current is passing through it and also the resistor has such a big value that it acts like an open circuit and obviously no current is passing through it. As a result, these two elements are ignored in the calculation of the leakage current. Before getting into any indepth discussion on finding the leakage current, it is necessary to display Table IX and Table X. Table IX shows the symbols and the corresponding names and Table X shows

the constants that are used in the derivation of the leakage model.

TABLE IX

DEFINITION OF COMMON SYMBOLS USED IN THE LEAK MODEL

Symbol	Name
Φ_T	Build-in potential (volts)
x_d	Thickness of depletion region (meters)
l	Length of MOSFET (1×10^{-6} meters)
w	Width of MOSFET (1×10^{-6} meters)
A	Area of the pn junction diode (1×10^{-12} meters)
C_{ox}	Oxide capacitance (farads/$meter^2$)
T_{ox}	Oxide thickness (meters)
τ_o	Average minority carrier lifetimes (seconds)

TABLE X

DEFINITION AND VALUES OF CONSTANTS USED IN THE LEAK MODEL

Symbol	Name	Constant Value
τ_n	Minority carrier lifetimes of n substrate (seconds)	1.0×10^{-7}
τ_p	Minority carrier lifetimes of p substrate (seconds)	1.0×10^{-6}
ε_{si}	Dielectric constant of silicon (Farads/meter)	1.04×10^{-10}
ε_{ox}	Dielectric constant of silicon dioxide (Farads/meter)	3.45×10^{-11}
n_i	Intrinsic carrier concentration, silicon (carriers/cm^3)	1.45×10^{10}
N_A	P-type substrate doping (carriers/cm^3)	1.0×10^{15}
N_D	N-type substrate doping (carriers/cm^3)	1.0×10^{16}
κ	Boltzman's constant (joules/degree-kelvin)	1.38×10^{-23}
T	Temperature (kelvin)	300
q	Electronic charge (coulombs)	1.602×10^{-19}

The important elements to look at in Figure 17 are the current source and the junction capacitor. The value of i_{s1} and i_{s2} are assumed to be approximately equal to the generation current in the depletion region and are given by

$$i_{s1} \approx \frac{qA_1 n_i}{2\tau_o} x_{d1} \text{ and } i_{s2} \approx \frac{qA_2 n_i}{2\tau_o} x_{d2} \tag{4.1}$$

The Φ_T is given by

$$\Phi_T = \frac{\kappa T}{q} ln(\frac{N_D N_A}{n_i^2}) \tag{4.2}$$

The p-n junction depletion region thickness is derived [Glasser 85] to be

$$x_{d1} = \sqrt{\frac{2\varepsilon_{si}\ (\Phi_{T1} + V_1)}{qN_D}} \text{ and } x_{d2} = \sqrt{\frac{2\varepsilon_{si}\ (\Phi_{T2} + V_2)}{qN_A}}. \tag{4.3}$$

Also from x_{d1} and x_{d2}, the junction capacitance is found to be

$$C_{j1} = A_1 \sqrt{\frac{\varepsilon_{si} qN_D}{2(\Phi_{T1} + V_1)}} \text{ and } C_{j2} = A_2 \sqrt{\frac{\varepsilon_{si} qN_A}{2(\Phi_{T2} + V_2)}} \tag{4.4}$$

respectively.

Using kirchoff current law, i_l in Figure 17 is obviously

$$i_l = i_g + i_{c2} \text{ and } i_l = i_{s2} - i_{s1} - i_{c1}$$

Since i_g is the current flowing out of C_g and i_{c2} out of C_{j2}, we have the current and capacitor relation of

$$i_g = -C_g \frac{dV_2}{dt} \text{ and } i_{c2} = -C_{j2} \frac{dV_2}{dt}. \tag{4.5}$$

Equating both i_l together and substituting in the appropriate equations gives us the following equation

$$-C_g \frac{dV_2}{dt} - C_{j2} \frac{dV_2}{dt} = \frac{qA_1 n_i}{2\tau_o} x_{d1} - \frac{qA_2 n_i}{2\tau_o} x_{d2} - C_{j1} \frac{dV_1}{dt} \tag{4.6}$$

In this equation we are interested in finding $\frac{dV_2}{dt}$ in terms of the other parameters. $\frac{dV_2}{dt}$ gives us the derivative of V_2 with respect to time and together with the gate capacitance C_g, i_g can be found. This i_g is the key equation that is used in building the NC transmis-

sion gate leakage current model.

To solve for $\frac{dV_2}{dt}$ in (4.6), we need to know what is V_1 in terms of V_2. From kirchoff voltage law, $V_1 = V_{dd} - V_2$. The following differential equation

$$\frac{dV_2}{dt} = \frac{-qn_i\,(A_2 x_{d2}(V_2) - A_1 x_{d1}(V_1))}{2\tau_o(C_g + C_{j2} + C_{j1})} \tag{4.7}$$

is then derived. This equation has three different capacitors that make up the capacitance. C_g is usually the load capacitor outside a circuit and is considered to be a constant. In our case, it is the gate capacitance of the storing transistor in the CAM cell. Using $\frac{\varepsilon_{ox} A}{T_{ox}}$ and assuming that the transistor has $l = 2.0\ \mu m$ and $w = 3.0\ \mu m$, C_g is found to be 4.14×10^{-15}. On the other hand, the junction capacitors C_{j1} and C_{j2} are not constants because they are a function of V_2. The values of C_{j1} and C_{j2} can be approximated by substituting $\frac{1}{2}V_{dd}$ as V_2. C_{j1} turns out to be 9.78×10^{-19} and C_{j2} to be 3.08×10^{-19}. The order of magnitude difference C_g and C_{j1}, C_{j2} has is in the thousands. Since the value of C_g is obtained using minimum size transistors and this value can only increase rather than decrease, both C_{j1} and C_{j2} is therefore considered negligible. As a result they are ignored in our calculation.

Equation (4.7) can be further simplified by assuming that both transistors in the transmission gate are of equal size. Then $A_1 = A_2$ and $\Phi_T = \Phi_{1,2}$. Substituting them into (4.7), the equation becomes

$$\frac{dV_2}{dt} = \frac{qn_i A}{2\tau_o C_g}\sqrt{\frac{2\varepsilon_{si}\Phi_T}{qN_A}}\left(\sqrt{1+\frac{V_2}{\Phi_T}} - \sqrt{1+\frac{V_{dd}-V_2}{\Phi_T}}\right) \tag{4.8}$$

To find out the leakage current i_g as a function of V_2, we combine $i_g = -C_g\frac{dV_2}{dt}$ with (4.7) and (4.8) to give

$$i_g = \frac{qn_i C_g (A_2 x_{d2}(V_2) - A_1 x_{d1}(V_1))}{2\tau_o (C_g + C_{j2} + C_{j1})} \tag{4.9}$$

and

$$i_g = \frac{qn_i A}{2\tau_o C_g} \sqrt{\frac{2\varepsilon_{si}\Phi_T}{qN_A}} \left(\sqrt{1 + \frac{V_2}{\Phi_T}} - \sqrt{1 + \frac{V_{dd} - V_2}{\Phi_T}} \right) \tag{4.10}$$

Equation (4.9) is used in the creation of NC model for general transmission gate charge leakage. Equation (4.10) is used in NC as a derived equation to model and analyzed the CAM cell charge leakage problem. Figure 18 shows a plot of the leakage voltage against time.

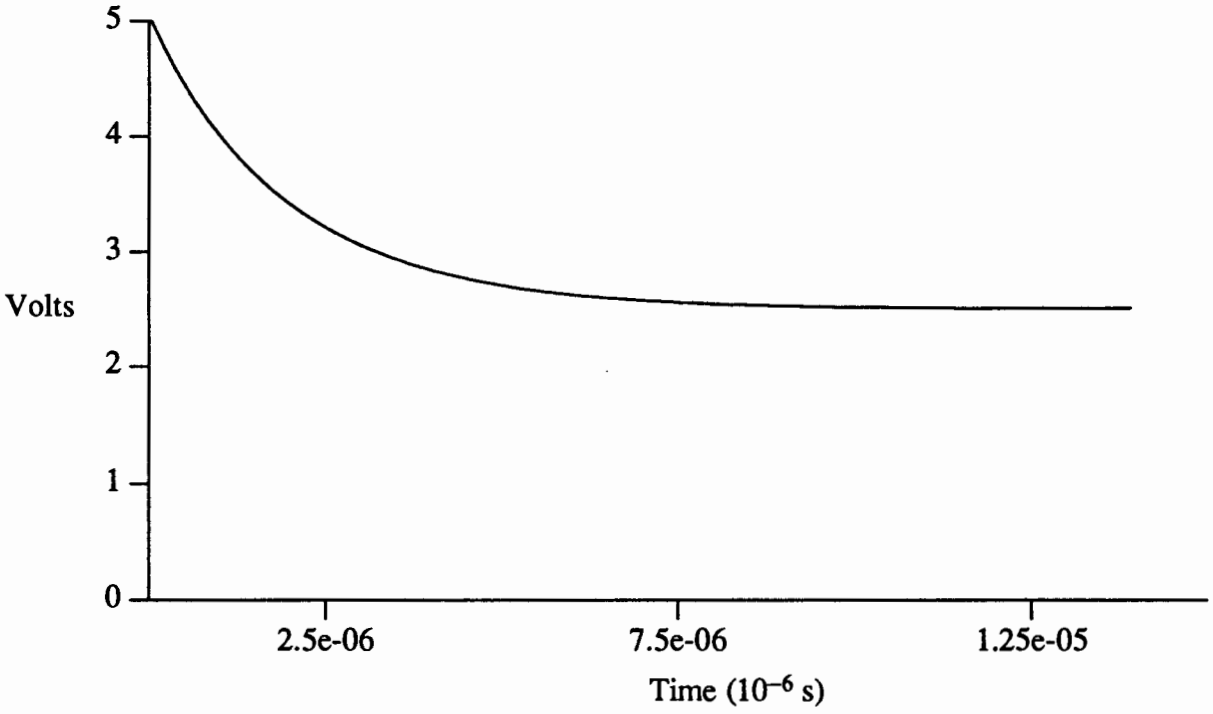

Figure 18. Plot of leakage voltage vs time.

The *tolcon* analog equation solver run control which is used in specifying the maximum acceptable value of the vector of residual node current has a default value of 1 microamps. As the simulation result reveals, the value of i_g is in the order of 10^{-10} amps which has exceeded the default value. Therefore *tolcon* has to be decreased to 10

picoamps so that the leaking event can be seen. The results shows that at around 11 microseconds the stored node would have leaked to around 2.5 volts. Using equation (4.10) as the leakage model, the stored node cannot get to exactly 2.50 volts because Figure 17 which is used to model the leakage phenomenon actually works like a voltage divider. However, it is a fairly good model because the simulation results is in agreement with the simulation results from SPICE analog model. SPICE shows that at about 14 microseconds, the stored node would have leaked to around 2.5 volts. We are unable to show with a plot because such small difference of only 3 microsecond cannot be noticed significantly on the plot.

CHAPTER V

LARGE SYSTEMS AND SUBCIRCUITS

In VLSI design, a circuit may contain hundreds of thousands of devices connected together. To simulate such a huge circuit is not an easy task. The CAM architecture that we have been discussing can become a very large circuit as well. Depending on how many rows and columns the CAM cell array has, the number of data latch (dlatch) circuits, mask circuits, selector circuits, sense amplifier circuits and priority resolver circuits can multiple according to the number of rows and columns. To represent so many devices in each part of the circuit into the input program of the simulator without the use of subcircuits representation is quite difficult to do. Both SPICE and NC supports the subcircuit method of presenting circuits in their input programs. The idea of the subcircuit method is to build modules with circuit components and use these modules repeatedly in the input program so that the user can write less code. The prototype circuit that is chosen is the mask network circuit which is shown in Figure 19. This circuit is a fairly good choice because of it hierarchical structure in its design. There are two aspects that will be examined when the mask network circuit is simulated using the SPICE analog model, NC analog and MOS_PWL model:

1. Whether computation time is different in using the subcircuit method compare to not using the subcircuit method.
2. Which model uses the least run time when the circuit becomes very large.

The dlatch circuit, mask circuit and selector circuit made up the mask network circuit. The number of columns in the mask network register will correspond to the number of columns in the CAM cell array. The function of the mask network circuit is described below. The dlatch circuits in the mask network circuits are used to temporarily store the information bits, one bit per dlatch circuit. These information bits are intended to be loaded into the CAM cell array. The dlatch circuit will wait for the load signal from the control circuit (refer to Figure 1 in Chapter 1) to arrive before it lets the information bits through to the CAM cell array. In this way, the system is synchronized. A truth table of the dlatch circuit is given in Figure 20a.

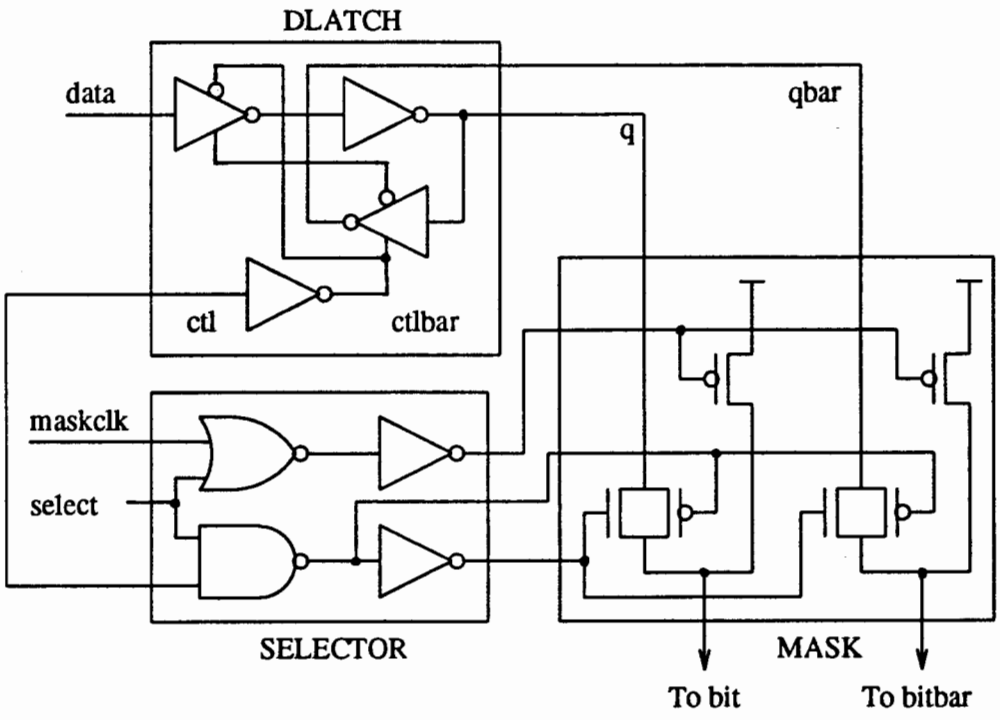

Figure 19. Mask network circuit.

The function of the mask circuit is to mask an entire column of the CAM cell array when desired. After an entire column has been masked, it becomes a DON'T CARE state and the bit and bitbar lines will both be at high potential. The mask circuit is

designed using two p-type transistors and 2 transmission gates. The two p-type transistors are activated in the masking process and the two transmission gates are turned on in the loading process. The user is able to select whether to load data into a column of the CAM cell array or mask that column through setting the select bit. A '0' on select bit will mask the column and a '1' will load the data. The selector circuit is controlled by two clock signals, maskclk and ctl to ensure synchronization in the system.

The two p-type transistors are controlled by the OR logic in the selector circuit. For both bit and bitbar line to be at high potential, the gates of the two p-type transistors must be at low potential. Only when select bit is low and maskclk is low will a mask operation be needed and must pull the gate of the two p-type transistors to low potential. Therefore the OR logic is used. On the other hand, the transmission gates are controlled by the AND logic in the selector circuit. For the transmission gate to turn on, the gate of the n-type transistor in the transmission gate must be at high potential and the gate of the p-type transistor which is in the transmission gate must be at low potential. Only when select bit is high and ctl is high will a load operation be needed. Therefore the AND logic is used. The truth table for AND and OR logic in the selector circuit is given in Figure 21a and 21b respectively.

ctl	ctlbar	data	q
1	0	1	1
1	0	0	0
0	1	X	q

(a)

	Mask Bit	q	A	B
Mask	0	0	1	1
		1	1	1
No Mask	1	0	0	1
		1	1	0

(b)

Figure 20. Truth table for the (a) dlatch and (b) mask circuit.

select	ctl	AND
0	0	0
0	1	0
1	0	0
1	1	1

(a)

select	maskclk	OR
0	0	0
0	1	1
1	0	1
1	1	1

(b)

Figure 21. Truth table for the selector circuit (a) AND and (b) OR logic.

The mask network circuit is designed to have a hierarchical structure. As mentioned above it is built with a dlatch, a selector and a mask circuit. The dlatch circuit is built with 2 clock inverters and 2 inverters. The selector circuit is built with 1 NOR gate, 1 NAND gate and 2 inverters. The mask circuit is built with 6 MOSFETs. The inverter is built with 2 MOSFETs. The clock inverter, NOR gate and NAND gate are all built with 4 MOSFETs. Due to this structure, the logic gates can be represented in the input program of SPICE and NC as subcircuits by using MOSFETs. The dlatch and selector circuits can then be represented as subcircuits using logic gates. The mask network circuit can then be represented as subcircuits using the subcircuits of dlatch, mask and selector. In this way, very large circuit can be represented fairly easy in the input program.

SIMULATION RESULTS

Figures 22, 23 and 24 show the simulation result of a 1 column mask network circuit using SPICE analog model, NC analog and NC MOS_PWL model. The sequence of events simulated are: load '1' to bit line, mask, load '0' to bit line and mask again. The simulation results show that from 0 to 30 nanoseconds, '1' is loaded to the bit line. From 35 to 65 nanoseconds, the column is masked. From 70 to 100 nanoseconds, '0' is loaded to the bit line and from 105 to 120 nanoseconds, the column is masked again. The simulation is made to last for 120 nanoseconds.

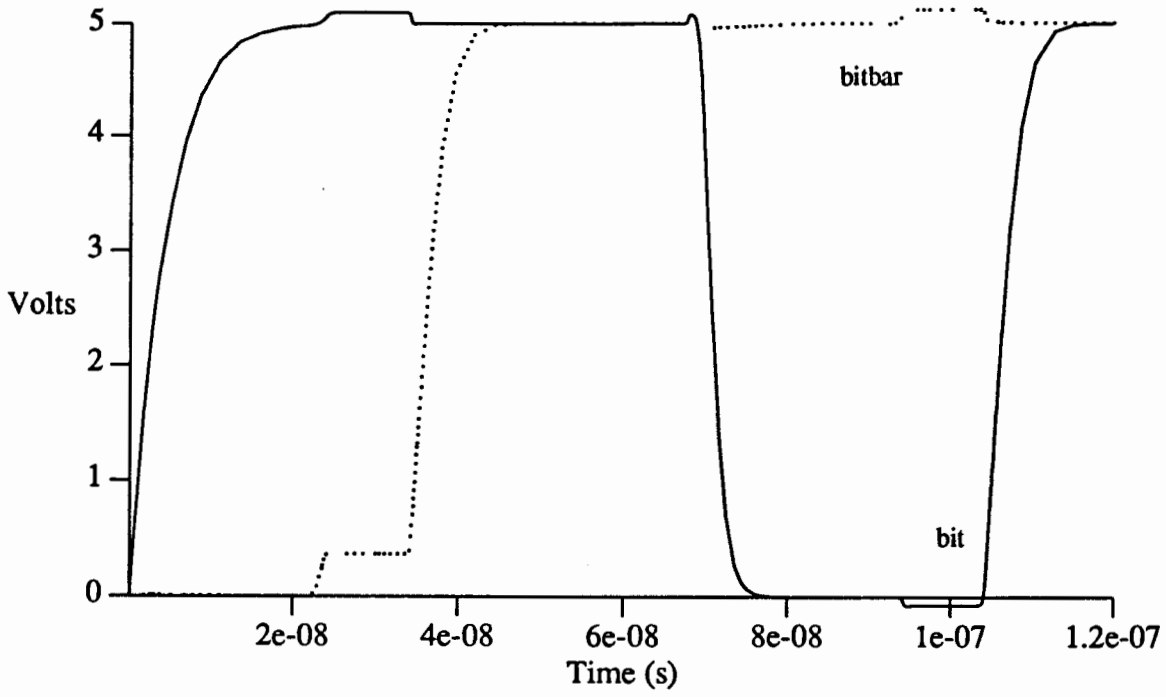

Figure 22. SPICE Analog Model Simulation of 1 Column Mask Network Circuit.

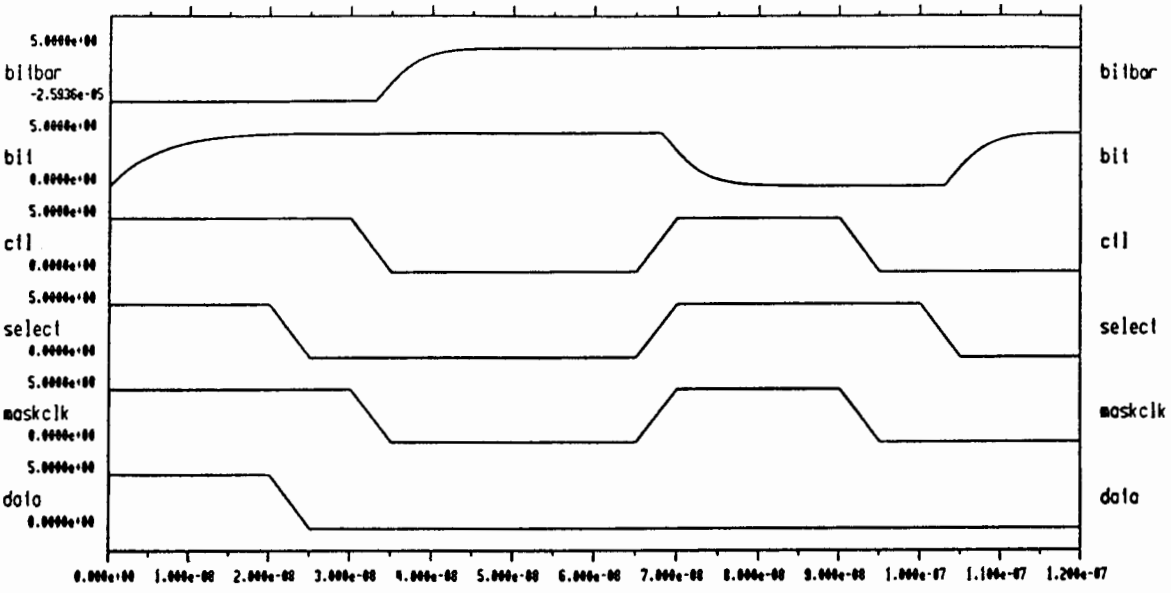

Figure 23. NC Analog Model Simulation of 1 Column Mask Network Circuit.

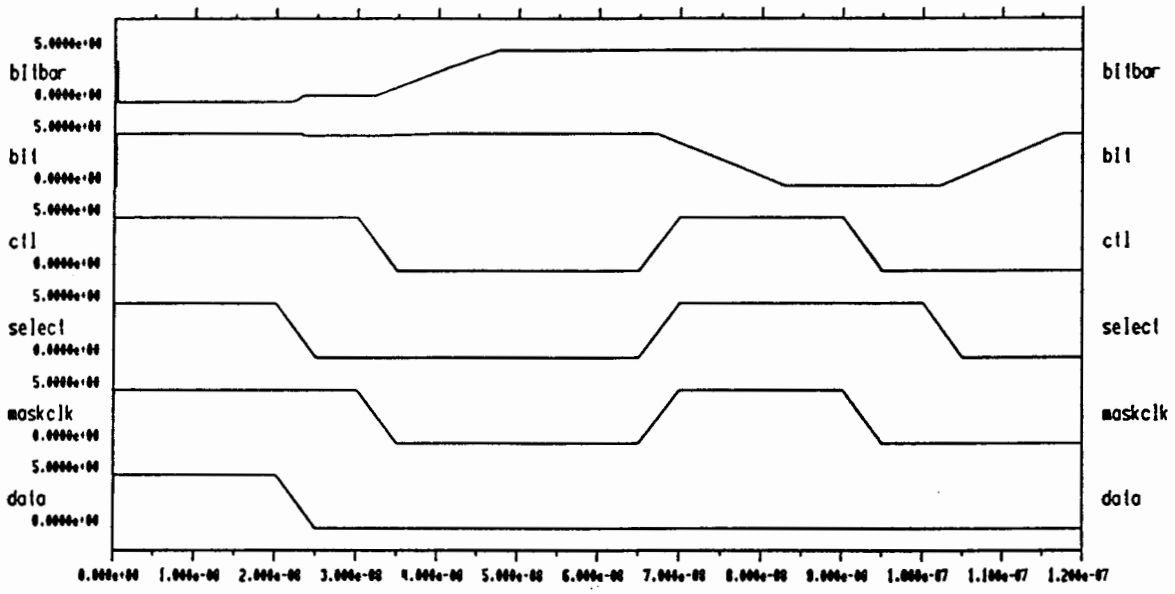

Figure 24. NC MOS_PWL Model Simulation of 1 Column Mask Network Circuit.

In an attempt to see how much run time each model needs when the circuit becomes large, the same sequence of events was used and up to 3 columns of the mask network circuit was simulated. The results are shown in Tables XI, XII and XIII.

TABLE XI

SIMULATION OF 1 COLUMN MASK NETWORK CIRCUIT

Simulation Tools	Total Run Time Without Subcircuits (s)	Total Run Time With Subcircuits (s)
NC MOS_PWL	225.5	220.9
NC Analog	823.0	1114.9
SPICE Analog	928.2	863.1

TABLE XII

SIMULATION OF 2 COLUMN MASK NETWORK CIRCUIT

Simulation Tools	Total Run Time Without Subcircuits (s)	Total Run Time With Subcircuits (s)
NC MOS_PWL	452.8	383.3
NC Analog	1669.5	1938.8
SPICE Analog	1860.2	1703.6

TABLE XIII

SIMULATION OF 3 COLUMN MASK NETWORK CIRCUIT

Simulation Tools	Total Run Time Without Subcircuits (s)	Total Run Time With Subcircuits (s)
NC MOS_PWL	682.2	683.1
NC Analog	2590.8	2776.1
SPICE Analog	2798.2	2641.5

The simulation results reveal that there is not really too much difference in run time between using the subcircuit method or not. However, from the users point of view, using the subcircuit method is to their advantage because large circuits can be represented with less code written. The results also reveals that when the circuits get very large, SPICE analog model needs the longest time to run the simulation. NC analog model runs just slightly faster than SPICE analog model. The NC MOS_PWL model, however is the most efficient model to use because it takes about 4 times less run time than SPICE analog and NC analog model. These results agree with the prediction that we discussed in page 16.

CHAPTER VI

CONCLUSION AND FUTURE WORK

CONCLUSION

The performance of SPICE analog model, NC analog model and NC MOS_PWL model has been examined and compared using the CAM architecture as a prototype circuit. This research is unique in that no such comparison has been done before. The comparison criterias are:

1. MOSFET model being used.
2. Circuit analysis and computational methods.
3. Limitation on the types of circuit that can be simulated.
4. Simulation time taken when simulating circuits.
5. Ability to build new circuit component models using derived equations.

The discussion on device modeling concludes that SPICE analog model uses a second order MOSFET model while NC uses a first order model. The second order model includes the second order MOSFET phenomena that occur when short and narrow channel MOS transistors are used. In this research, the predicted output voltages of the three simulation models agree with each other because no short or narrow channel MOS transistors are used. Therefore using the first order or second order MOSFET model does not make much difference in the output accuracy. However in NC's MOSFET model, the capacitive coupling between the gate to drain, gate to source and drain to source are not included. This leads to a difference in the outputs between the three simulation models in simulating the write operation performed by the Mundy *et al.* [Mundy 72] CAM cell. SPICE is better than NC in this aspect because it includes the MOS capaci-

tive coupling in its calculation. However, the first order model is just a default MOSFET model in the NC library. The user can built a more accurate MOSFET model which includes the second order phenomena and the capacitive coupling between nodes by directly using the mathematical equations which describe the effects. This model building capability is a powerful tool that NC has to overcome its shortcomings. The example that we show is the leakage problem in a dynamic CAM cell. A transmission gate leakage current model is built to compute the time required between refresh cycles. NC is basically an equation solver capable of doing numerical analysis. Therefore users can build electrical component models using a set of derived mathematical equations which describe the behavior of the electrical component. SPICE does not allow users to do that, the users can only use the traditional electrical components that already exist in SPICE. Although it is true that a subcircuit of Figure 12 can be built in SPICE and use that subcircuit as a circuit components, SPICE does not allow users to built models by directly using the derived mathematical equations to describe the behavior of the circuit. When the circuit components needed to build subcircuits increase in number, it is more efficient to use the model building method. The reason is that by using the model building method, the simulator can simulate a circuit in the functional level rather than the circuit level. The simulator uses the functional behavior (described by the derived equations) of the circuit directly without having to spend time calculating the node voltages and currents (circuit level) in the circuit. In this way large amount of computation time can be saved and therefore the model building method is a valuable technique to use today and will surely continue to be used in the future. The model building method has also gain commercial acceptance because the SABER simulator which is commercially available, uses this method.

The circuit analysis and computation method that both SPICE analog and NC analog model uses are about the same. Both use a fixed time step method in evaluating the residual current vector from the node voltages. NC MOS_PWL model uses a

predict-from-input method which turns out to be faster in simulating most of the prototype circuits presented. Due to the way NC MOS_PWL model breaks a circuit down into stages, it has difficulty simulating circuits like the voltage reference generator which has both n and p type MOS transistors connected to behave like diodes. As shown in this thesis, the way to remove this difficulty is to include a transducer which equates both the gate and drain node voltages and currents and reduce the forecast interval.

The behavior of the three simulation models for large circuits and the use of subcircuit method to represent circuits in the NC and SPICE are also examined in this research. The simulation results reveal that using or not using the subcircuit method does not produce big differences in simulation time between the three simulation models. However, when the circuit being simulated has large number of devices in it, the simulation time taken by SPICE analog model and NC analog model is slower than the simulation time taken by NC MOS_PWL model by a factor of four. Therefore NC MOS_PWL model is the preferred model to simulate large circuits.

FUTURE WORK

Only the MOSFET and the p-n junction in MOSFET are examined in this research. This work can be expanded by examining not just the MOSFETs but also the other semiconductor devices. Hopefully, this research will contribute to a better understanding of the limits and advantages of some circuit simulators that designers use today. Based on the advantages and limitations of today's simulators, more improved simulators can be designed.

REFERENCES

[AntoMass 87] Paolo Antognetti and Giuseppe Massobrio, editor, "Semiconductor Device Modeling With SPICE", McGraw-Hill, 1987.

[Avriel 76] M. Avriel, "Nonlinear Programming: Analysis and Methods", Prentice-Hall, 1976.

[Beckett 88] William Beckett, "Modeling Electronic Circuits and Systems in Network C", Department of Computer Science, University of Washington, July 14, 1988.

[Bryant 81] R. Bryant, "MOSSIM:A switch-level simulator for MOS LSI", in Proc. 18th Ass. Comput. Mach. Design Automat. Conf., 1981, pp. 786-790.

[Bryant 84] R. Bryant, "A switch-level model and simulator for MOS digital systems", IEEE Transactions on Computers, Vol. C-33, No. 2, Feb. 1984, pp. 160-177.

[ChGK 75] B. R. Chawla, H. K. Gummel and P. Kozak, "MOTIS--An MOS timing simulator", IEEE Trans. Circuits Syst., pp. 901-910, Dec. 1975.

[Divekar 88] Dileep A. Divekar, "FET Modeling For Circuit Simulation", Kluwer Academic Publishers, 1988.

[Glasser 85] Lance A. Glasser and Daniel W. Dobberpuhl, " The Design and Analysis of VLSI Circuits", Addison-Wesley publishing Company, 1985.

[Koo 70] James T. Koo. "Integrated-Circuit Content-Addressable Memories", IEEE Journal of Solid-State Circuits, Vol. SC-5, No.5, PP.208-214, Oct. 1970.

[Meyer 71] J. E. Meyer, "MOS Models and Circuit Simulation", RCA Review, Vol. 32, pp. 42-63, March 1971.

[Mundy 72] Joseph L. Mundy *et al.*, "Low-Cost Associative Memory", IEEE Journal of Solid-State Circuits, vol. sc-7, No. 5, pp.364-369, Oct. 1972.

[Nagel 73] L. Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits", ERL Memo No. ERL-M520, University of California, Berkely, 1975.

[ShicHod 68] H. Shicman and D.A. Hodges, "Modeling and Simulation of Insulated-Gate Field-Effect Transistor Switching Circuits", IEEE Journal of Solid-State Circuits, Vol. SC-3, No.5, PP.285-289, Sept. 1968.

[Sodini 89] Jon P. Wade and Charles G. Sodini, "A Ternary Content Addressable Search Engine", IEEE Journal of Solid-State Circuits, vol. 24, No. 4, pp 1003-1013, Aug. 1989.

[Staab 86] Sanford Staab, "A Tutorial for Spice3/Nutmeg", University of California, Berkeley, March 20, 1986.

[Terman 83] Chistopher J. Terman, "Simulation Tools for Digital LSI Design", PH. D. dissertation, Massachusetts Institute of Technology, 1983.

[Uyemura 88] John P. Uyemura, "Fundamentals of MOS Digital Integrated Circuits", Addison Wesley Publishing Company, 1988.

[Weeks 73] W. Weeks *et. al.*, " Algorithms for ASTAP -- A Network Analysis Program", IEEE Trans. Circuit Theory, Vol. CT-20, pp. 628-634, Nov. 1973.

APPENDIX

SPICE AND NC I/O PROCESS COMPARISON

When an input program is compiled in SPICE, it starts by initializing some program constants and then read the .title card [AntoMass 87]. The process will stop if it reads an end-of-file, otherwise it calls SETUP and ERRCHK modules to read the input file and check for syntax errors respectively. SPICE then enters into an interactive mode with the user [Staab 86]. If the compiling is successful, the title card will appear on the screen and a prompt follows. Otherwise, the prompt will appear following the error messages. SPICE provides detail syntax error messages, pin-point to specific nodes allowing the user to easily correct the error. In the interactive mode, the user can do a number of things. The user can execute the program, specify which type of analysis is to be done on the circuit, plot the desired nodes, print the node voltages at each time step on all the nodes, edit the original input program to make changes, re-compile and re-execute the program. The user can continue doing this until a satisfy result has been obtained. These steps can also be done using the control cards available in SPICE by placing them in the input program. Using the control card is a slower method because the user has to re-edit and re-run the input program if he wants to print all the node voltages at the end of a simulation but has forgotten to put a .print in the input program in the first place. However, both types of usage provides greater flexibility for the user.

NC, on the other hand, does not support the interactive mode and accepts input circuit description programs quite differently from SPICE. The circuit description and run controls are placed under separate files. Both files are distinguished by using '. *n*' and '. *ctl*' postfix at the base name of the file. The circuit description file contains the usual circuit elements such as resistors, capacitors, semiconductor devices and voltage

sources. The nodes connecting these elements can be descriptive names unlike in SPICE where nodes are numbered. Voltage sources that NC supports are similar to SPICE in that they can be constant or time-varying. The control file consists of run controls such as those found in chapter 7 of [Becket 88]. The most commonly used controls are *start, stop* and *scale* (control the amount of execution time), *preset* (control initial conditions of nodes), *analog, digital* (specify the nodes to be shown) and *pulse* (voltage source control). NC uses a shell script to handle compiling and linking. NC has identical programming language structures as the C programming language. Each devices found in the circuit description is a written module which can be found in the NC library. When the shell script is invoked, the simulation program is compiled, linked to the NC library and then executed. Option '-c' can be used to avoid the linking if users wish to compile the modules separately. This option is particularly useful when the user has built a module of his own and wish to compile it separately. During the compiling process, NC will translate the NC program into a C program (*basename.c*), generate the object file (*basename.o*) and generate an executable model for simulation (*a.out*) The execution of the simulation program will create *basename.pd, basename.ev* and *basename.log* files. The *basename.ev* file contains the record of the events happened on the nodes which have been specified in the control file during the execution. The *basename.ev* file can then be used for plotting the events using the *scp* command.

From a user's point of view, I would say that the interactive nature of SPICE is more convenient to use than NC. SPICE do not generate a lot of confusing files like NC after execution. In the interactive mode, the user may run the circuit several times and he can selectively save the desired simulation from a list SPICE provided. Also the reporting of node voltages in SPICE is more readable than the *basename.ev* file in NC.