A MOS delay Model for Switch-Level Simulation

by

Marcos David Peckel

A thesis submitted to the Faculty of Graduate Studies
and Research in partial fulfillment of the
requirements for the degree of
Master of Engineering

Department of Electrical Engineering
McGill University
Montréal

August 1985

ABSTRACT

A new delay model is proposed in this thesis for the switch-level simulation of MOS circuits. The model, based on the RC representation of the circuit, accounts for most of the factors that influence delays in MOS circuits. The delay model represents the transistor as an *effective resistance* whose value is computed dynamically during simulation according to the size and type of the transistor, its context, capacitive load and gate input waveform. The model has been incorporated into a new switch-level simulator and has been tested on both CMOS and NMOS circuits. Agreement with SPICE is within 10% for most of the cases and the overhead associated with the delay calculation is a factor of 2.5 to 3.

Résumé

Un nouveau modèle de transistor MOS est proposé pour la simulation au niveau commutateur. Le circuit est représenté par un réseau de résistances et de condensateurs. Ce modèle tient compte des effets prédominants qui causent les délais dans les circuits MOS. Dans ce modèle les transistors sont représentés par une résistance équivalente qui est calculée dynamiquement pendant la simulation selon la grosseur et le type de transistor, son contexte, sa charge capacitive et la forme du signal qui le contrôle. Ce modèle fut incorporé à un nouveau simulateur et fut évalué sur des circuits NMOS ainsi que CMOS. Les résultats obtenus sont en-dedans de 10% des résultats obtenus avec SPICE dans la plupart des cas. L'évaluation du délai n'augmente le temps de simulation que par un facteur entre 2.5 et 3.

A mis padres

Acknowledgements

I wish to gratefully thank my advisor, Professor Nicholas Rumin, for both his invaluable and instructive guidance, as well as the enthusiasm and sincere interest he showed towards my work and my person. I would also like to acknowledge Professor Janusz Rajski, for showing interest in my work and sharing helpful discussions with me. Further, I wish to thank my friends in the VLSI group, in whose company the hottest research problems were solved through endless lunches and unforgetable ski tours.

My thanks are also given to a great person for her understanding and support and the long nights devoted to the proof-reading of the text.

Finally, I am grateful to the National Sciences and Engineering Research Council of Canada for providing financial support during the last stage of this work.

TABLE OF CONTENTS

TABLE OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

1.1 Simulation as a design tool

The increasing complexity of VLSI circuits poses a formidable challenge to existing computer-aided-design (CAD) tools. New computer aids used in the design process must be developed regularly in order to keep pace with the advances in the technology. Simulation, in particular, plays an essential role in the design of digital integrated circuits (IC), as it provides a unique way to verify the functionality and performance of a proposed design, before it is sent to the costly fabrication process.

Digital integrated circuits have been traditionally simulated using either circuit simulation or logic simulation. Circuit simulators such as SPICE [Nag75], SCAMPER [Agn80] or ASTAP [Wee73], provide an accurate picture of the electrical behaviour of the circuit, by using very complex transistor models based on the physics of the devices. However, the heavy computational time and storage involved in circuit simulation limit its applicability to circuits containing on the order of one thousand transistors.

A simplified form of circuit analysis for MOS circuits is provided by timing simulators such as MOTIS [Cha75], MOTIC-C [Fan77], and SPLICE [New78]. Timing simulation performs a waveform analysis, while exploiting particular properties of MOS circuits in order to reduce the time and storage requirements of the simulator.

These properties are basically the low activity in MOS digital circuits (latency) and the unilateral nature of most of the circuit configurations. In addition, rather than using analytical models, timing simulators rely on simplified table look-up transistor models in order to speed up the simulation time. However, because timing simulators solve the circuit equations by methods such as Gauss-Jacobi or Gauss-Seidel iteration, they encounter stability and accuracy problems when handling bilateral elements such as transmission gates and floating capacitors, or circuits with tight feedback, such as latches. This problem appears to have been solved by the introduction of mixed-mode simulators such as SPLICE [New78], MOTIS-2 [Che84], and DIANA [Dem78], in which critical configurations are singled out and handled using circuit analysis techniques. Timing simulators operate at between one and two orders of magnitude faster than circuit simulators, but for the analysis of very large circuits they too become impractical in terms of computer time.

On the other hand, gate level simulators such as TEGAS [Szy72] or SALOGS [Cas78] are based on the boolean gate model by which a network is represented by a set of ideal, unidirectional logic gates. Voltages are represented by a discrete set of logic values rather than a continuous voltage function. Propagation delay estimates are also provided by most gate level simulators through the use of a scheduler and tabulated delays. Due to the simple models and the temporal and structural sparsity of logic circuits, gate-level simulators are fast (about three to four orders of magnitude faster than circuit simulators) and can handle very large circuits. They are, however, inadequate for MOS circuits, which contain circuit structures such as transmission gates, dynamic storage, and precharged logic, that can not be adequately modelled by logic gates.

In order to solve the problems presented by circuit and gate level simulators, switch level simulators have been proposed for the logic simulation of large MOS

VLSI circuits [Bry80], [Hay82], [Bry84], [Cer84]. In switch-level simulation, the circuit is modelled at the transistor level, but with each transistor represented as a single bidirectional switch. The switch can be either ON or OFF depending on the logic value at the gate of the transistor. The network is represented as a set of switches and nodes. Similar to gate-level simulation, switch level simulators employ a reduced set of logic values. rather than a continuous voltage function. Transistors can be assigned a *strength* from a discrete set as a measure of current drive capability, so that ratio logic circuits, such as NMOS can be properly modelled. Similarly, nodes can be assigned a *size* according to their capacitance, as a measure of dynamic drive capability, so that charge sharing effects can be included. The logic evaluation on the switch-level network is performed by propagating logic values through ON switches and by updating the state of the switches whenever their gate changes state. By modelling the network at the transistor level, the switch level representation of MOS circuits can be obtained directly from the layout, and corresponds to the actual structure of the design, rather than to its intended function, as is the case in gate-level simulators. This network model allows switch-level simulators to handle typical MOS structures, such as transmission gates, precharged logic, and dynamic storage, which can not be adequately modelled in gate level simulation. The simplicity of the transistor models permits switch level simulators to operate at speeds approaching those of logic simulators.

1.2 Delay calculation in switch-level simulation

Delay calculations can be introduced in switch-level simulators to provide a first-order timing analysis. These delay estimations present a much more formidable problem than in gate-level simulation, because of the bilateral nature of the switch. As a result, the delay through a closed switch depends not only on the entire

subnetwork to which it is connected, but also on its past history. In other words, the delay must be computed every time a switch changes state.

An approach to calculate delays in switch-level simulation based on a two-parameter model was recently reported by Mo and Lightner [MoL84]. The model includes the effect of the input waveform shape in the delay calculations through the network. Reported experimental results show a good agreement between the two-parameter model and SPICE for some MOS circuits.

The traditional approach for delay calculations in switch-level simulators has been to represent the MOS circuit as an equivalent RC network [Mea80], [Pen81], where the ON transistors are replaced by equivalent resistances and all the nodes by tied-to-ground capacitances. Two basic research topics are involved in the calculation of delays using the RC model:

1. Development of efficient algorithms to calculate delays in RC networks.
2. Design of appropriate techniques for modelling the transistor as an effective resistance.

Delay calculation in RC networks

One of the earliest reported attempts at including delay estimations in a switch-level simulator using the RC model, was that of Terman [Ter83]. In his approach, the delay is calculated on a linear RC network using a lumped model with Thevenin equivalents for the resistances and capacitances. This approach results in overestimation of the delay for most practical cases as it assumes all the capacitances in the network as charging/discharging through all the resistances. A more realistic approach is that of Penfield and Rubinstein [Rub83]. They derived upper and lower bounds for delays in RC trees. In their approach, the capacitances in the tree charge/discharge only through the resistance path connecting them to the signal source. The simplicity of their method represents its major success, and it has been

incorporated into some timing-analysis programs [Jou83,Tam83]. However, their method is unable to handle general RC networks containing reconvergent paths. Subsequently, Lin and Mead [Lin84] recognized that delays in linear RC trees can be computed with surprisingly good accuracy using a simple and convenient formulation of the so-called Elmore delay [Elm48]. They extended this method to general RC networks, and devised a very attractive algorithm for incorporating delay calculations into modern switch-level simulator such as MOSSIM II [Bry84].

Switch-level delay modelling

Earlier attempts at modelling the transistor as an effective resistance for delay calculations include those of Terman [Ter83], Lin and Mead [Lin84], and Ousterhout [Ous84]. In the model used by Lin the effective resistance is assumed to be only a function of the size of the transistor. In the model developed by Terman the transistors are classified by the simulator according to 5 different contexts, namely, depletion pullup, enhancement pulldown, enhancement pass transistor, depletion pass transistor, and pulldown with threshold drop. For each of these cases the effective resistance is evaluated as a function of the size of the transistor. The models developed by Terman and Lin can lead to serious errors because they do not include such effects as the load driven by the transistor and the gate input waveform. For delay calculation in CMOS circuits, this is a serious limitation.

A significant contribution to switch-level delay modelling was made by Ousterhout [Ous84]. In his formulation of the delay models for the timing analyzer CRYSTAL [Ous82], Ousterhout developed an efficient method for including the effect of the input waveform slope on the effective resistance of the transistor. For a transistor switching, Ousterhout introduced the rise-time ratio as a measure of how much the transistor is affected by the input waveform. His approach, however, is deficient in that the value of the effective resistance is not adjusted with the load driven by

the transistor, and only one transistor is allowed to switch at a time. Moreover, his method of handling the effect of the input waveform's slope is inappropriate for a switch level simulator.

1.3 Thesis objective

The objective of the work reported in this thesis was to develop a delay model for switch level simulation, based on the RC representation of MOS circuits, that possesses the following characteristics:

- The model is simple to implement in a simulator and is computationally efficient, allowing the user to simulate very large MOS circuits.
- The model accounts for all the most important effects that influence delay in MOS circuits. Modelling the transistor as an RC network, the resistance is a function of not only the type and size of the transistor, but also the context, the load driven by the transistor, and the slope of the input gate waveform.
- The model is able to handle both static and dynamic, NMOS and CMOS circuits.
- The model is easy to update for technological changes including down-scaling of the circuit feature sizes.

1.4 Organization of this thesis

Prior to the presentation of the new switch-level delay model, a survey is presented in Chapter 2 of the MOS modelling techniques used in circuit, timing, and gate-level simulation. Also a discussion of up to date developments in the area of switch-level delay modelling is included. The proposed new model is described in detail in Chapter 3. This includes the evaluation of the required values of capacitances and resistances used to build the MOS RC model, on which the delay calculations are carried out. A description of how the model was derived, and how it is used in

the simulator in which it has been tested is also presented. Chapter 4 presents, for a wide range of MOS circuits, simulation results using the new model and comparisons with SPICE. Most of the examples here are CMOS circuits, as compared to previous work, where all the examples are NMOS [Ter83,Ous83,Lin84]. Conclusions about the performance and limitations of the new model are presented in Chapter 5.

CHAPTER 2

MODELLING FOR SIMULATION

2.1 Introduction

In this chapter the different algorithms and modelling techniques used in MOS simulation are described and analyzed, showing the tradeoffs between accuracy, speed, and storage. The basic structure and models used by circuit, timing and gate-level simulators are presented in Sections 2.2, 2.3 and 2.4. The basic structure of switch-level simulators is discussed in Section 2.5, along with the main algorithms used for logic evaluation and delay calculation. Emphasis is placed on the techniques used for switch-level delay modelling, with reference to previous work in the area, as well as a discussion of the motivation for developing a new MOS switch-level delay model.

2.2 Circuit Simulation and Models

2.2.1 Circuit Simulation

The purpose of circuit simulation is to accurately predict the electrical behaviour of electronic circuits. Circuit simulators, such as SPICE [Nag75], ASTAP [Wee73] and SCAMPER [Agn80], read the topology of the circuit, and using built-in models for its electrical components, describe the circuit as a set of nonlinear, algebro-differential equations, which are solved using numerical methods. SPICE

[Nag75], in particular, uses the Modified Nodal Analysis (MNA) [Ho75] based on Kirchhoff current and voltage laws, to build the circuit's system of equations. For dynamic circuits. i.e., those containing capacitors or inductors, MNA yields a system of nonlinear differential equations of the form:

$$f(\dot{v}, v, t) = 0 \tag{2.1}$$

where v is the vector of node voltages, $\dot{v}$ is the first derivative of v with respect to time, and t is the time.

Two different types of analyses, DC and transient are performed on the circuit. The DC analysis is performed to find the initial values for the voltages at the different nodes of the circuit, with all the capacitors open and the inductors short-circuited. Newton-Raphson iteration is used and the resulting linear system of equations is solved at each iteration using a form of Gaussian elimination [San80].

The transient analysis is much more complex and involves the solution of the complete system of nonlinear algebro- differential equations of (2.1). The structure of a circuit simulator in terms of a flow chart is shown in Fig. 2.1.

The analysis time interval T is divided into small time steps,

$$t_0, \; t_1 = t_0 + t_h, \quad t_2 = t_1 + t_h, \ldots \quad t_n = t_{n-1} + t_h$$

where each increment, t_h, is called the step size. A node voltage at t_{i+1} is predicted on basis of the voltages found at the previous timesteps using a polynomial approximation of the voltage. Various implicit integration methods are used for that purpose including the backward Euler and Trapezoidal equations. As a result, the differential equations which describe the behaviour of dynamic elements such as capacitors and inductors, are discretized and a system of nonlinear algebraic equations is obtained.

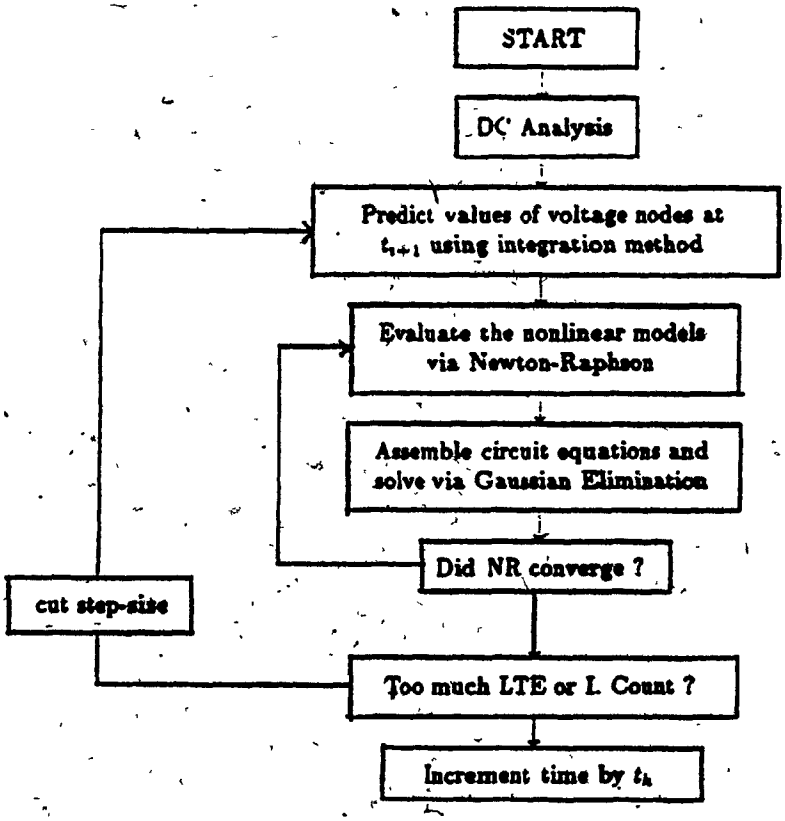


Figure 2.1 General structure of circuit simulators.

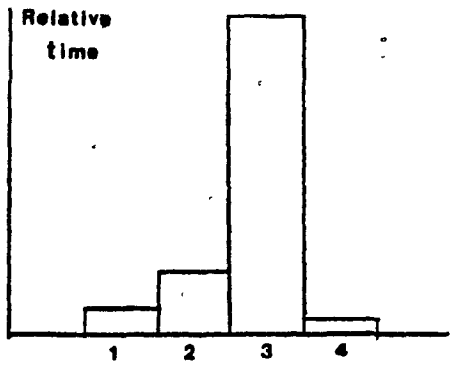
These equations are solved using the Newton Raphson (NR) iteration method, wherein the nonlinear elements are replaced by linear resistors and independent sources [San80]. The resulting linear system of equations, which describes the complete circuit at the particular timepoint, is subsequently solved using Gaussian elimination. If the solution does not converge this loop is repeated until convergence is obtained (typically less than 0.1% difference in the computed node voltages between successive NR iterations).

Once convergence is obtained the error introduced by the implicit integration formula is estimated. This estimation is done by either a Local Truncation Error scheme [Nag75] or by counting the number of iterations required for convergence in

the Newthon Raphson loop [San80]. If the solution at a timepoint is not satisfactory, i.e., the number of iterations in the latter case exceeds a predetermined number, the time step is cut and the analysis is repeated. Otherwise the timestep is incremented by a stepsize and the analysis continues. Some simulators provide a variable stepsize to speed up the simulation algorithm.

Fig. 2.2 shows a bar chart displaying the effective time spent by the computer in each SPICE major block for a medium size circuit [New80]. The bar MOS corresponds to the time needed to evaluate the MOS models which is about 80% of the total computer time, whereas the solution of the linear system accounts only for a small fraction of the total CPU time. However, it has been observed [San80] that the computational complexity of the model evaluation grows linearly with the circuit size, while the complexity of the linear equation solver grows approximately as $n^{1.4}$ (n corresponds to the number of equations). For circuits containing about 3000 devices the time spent in the model evaluation is reduced to about 50% of the total CPU time. Therefore, it can be concluded that the basic limiting factor in terms of execution speed in applying circuit simulation to VLSI circuits is the fact that at each iteration of the Newton Raphson algorithm a large system of equations must be solved. Some techniques, such as node tearing [Yan80] and vector processing [Vla82] have been reported to improve the speed of circuit simulators. By means of node tearing the circuit is partitioned into subnetworks in such a way that only those that are active are analyzed. Vector processing, on the other hand, is suitable for circuits which contain repetitive subcircuits which then can be analyzed in parallel.

Despite its limitations regarding circuit size, circuit simulation remains an essential tool in the VLSI design process. It is irreplaceble when it is important to verify the analog voltage levels of a part of an entire circuit, or when tightly coupled feedback loops need to be considered.



1. Integration of Capacitor Currents
2. Evaluation of LTE
3. MOS model evaluation
4. Linear Equation Solver

Figure 2.2 Relative time spent by SPICE in major component blocks [New80].

2.2.2 MOS models for circuit simulation

The transistor models used by circuit simulators must represent accurately the behaviour of the device. This results in very complex, nonlinear models, which the simulator has to evaluate at each iteration step, as shown in Fig. 2.1. In other words, accuracy is achieved at a high price in terms of computer time and storage.

There are three MOS models available in SPICE [Vla79]. The most complex of the three, MOS3, is a semi-empirical model described by a set of parameters which are obtained by curve-fitting rather than by physical analysis. The MOS2 model is an analytical model based on the work of a several authors and incorporates most of the second order effects for small size devices. The simplest model, MOS1, is similar to the model proposed by Schichman and Hodges [Sch68]. The MOS2 and MOS3 models differ from MOS1 in the introduction of second order effects such as channel length modulation, subthreshold conduction, scattering-limited velocity saturation, and small size effects which play an important role in today's circuit performance as the dimensions of the devices are scaled down. The MOS2 model includes the

effects of small geometry in the threshold voltage, the saturation voltage, and the drain current. MOS3 is designed mainly for small geometry MOSFETS defined as transistors with channel length, $L < 2\mu$ and channel width, $W < 2\mu$. It models the short and narrow channel effects on the threshold voltage (differently from MOS2), the effect of the static drain to gate feedback on the threshold voltage, and the lowered saturation voltage and current due to the velocity saturation of hot electrons.

Fig. 2.3(a) shows a simplified cross-sectional view of an MOS transistor. Its equivalent SPICE model, in the form of an equivalent network, appears in Fig. 2.3(b). This topology is the same for all three models, although the values are evaluated differently. I_d corresponds to the channel current. R_d and R_s represent the parasitic drain and source series resistance. C_{db} and C_{sb}, are the nonlinear, depletion layer capacitances due to the body-drain and body-source junctions respectively. C_{gso}, C_{gdo} and C_{gbo}, represent respectively gate-source, gate-drain, and gate-body overlap capacitances, which are outside the channel region. The thin-oxide channel capacitance, which is not shown in the figure, is calculated by the program and distributed among the gate, source, drain, and bulk regions.

A basic difference that exists in the three models is related to the formulation of the channel current, I_d. MOS1 uses the well known quadratic model to express I_d as a function of the gate-source and drain-source voltages [Hod83]. MOS2, in its formulation of the current I_d, includes such effects as the short and narrow channel, the substrate charge and the degradation in the surface mobility. MOS3, on the other hand, includes an empirical correction factor for small size effects. The concept of weak inversion conduction is introduced in MOS2 and MOS3. A MOSFET is not an ideal device which starts conducting abruptly when the gate voltage, V_{gs}, reaches the threshold level, V_{th}. There is current flowing in the device

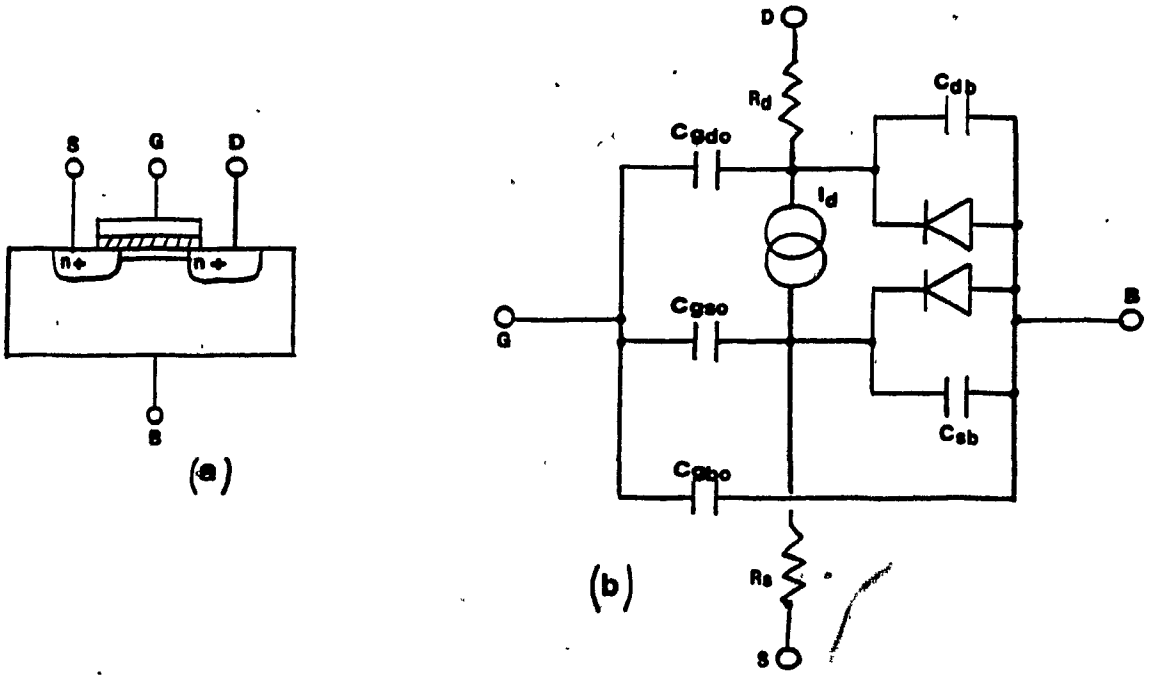


Figure 2.3 (a) Short-channel MOS transistor.
(b) Equivalent SPICE model.

below V_{th}. This weak inversion region is defined around the threshold voltage V_{th}, according to the model developed by Swanson and Meindl [Swa72], as shown in Fig. 2.4.

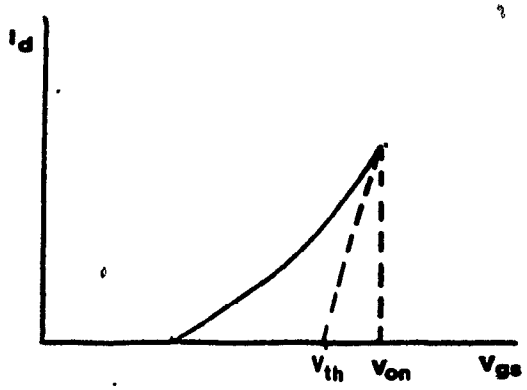


Figure 2.4 Weak inversion region.

Different expressions are used for the channel current I_d in the weak and strong inversion regions, although continuity is assured for the transition between them. Both the MOS2 and MOS3 use equal formulation for the current I_d in this weak inversion region.

The thin-dioxide gate capacitance is calculated by the program during simulation and is distributed between the gate, source, drain, and substrate terminals. In the MOS2 and MOS3 models, it is done using the charge-oriented model proposed by Ward and Dutton [War78], as compared to the voltage-controlled model formulated by Meyer and used in MOS1 [Mey73]. As for the junction capacitances, they are represented as averaged constant capacitances in MOS1, whereas in MOS2 and MOS3 they are modelled as nonlinear charge controlled capacitances. Overlap capacitances , and parasitic resistances are equally formulated in all three models.

The key to obtaining high accuracy in MOS IC simulation using the models described above, depends heavily on the ability to find the technology dependent parameters correctly. These parameters and their significance appear in [Vla79] and [Hod83]. The accuracy of the MOS2 and MOS3 models lies between 5% to 10% of reality. The introduction of second order effects, absent in the MOS1 model, improves accuracy by 20% to 25%. For devices approaching the 2 μ range the MOS3 model should yield more accurate results than those obtained from the MOS2 model [Vla79]. For devices whose dimensions are in the 5 μ range, there is almost no difference in the simulation results using MOS2 and MOS3, provided that proper parameters are used. The evaluation time for the MOS2 model takes about 20% longer than MOS1 and MOS3 [Vla79].

Table 2.1 shows the delay, rise, and fall time obtained for a CMOS 8-to-1 selector circuit based on a 5 μ technology, using the three models available in SPICE2G6, for a particular input sequence. The total CPU simulation time is

also shown. These results which are in concordance with the estimates presented in [Vla79], suggest that the MOS2 model which is *simpler to parametrize than MOS3*, is accurate enough for 5-micron MOS technology. This is important to note, since the new switch level delay model is developed using SPICE MOS2 model and is verified on 5 μ technology circuits.

	MOS1	MOS2	MOS3
Delay (ns)	11.5 24.2	17 30.5	18 31.5
Rise time (ns)	3.5	4.8	5.2
Fall time (ns)	2.5	4.7	5.0
CPU time (sec)	738	828	718

Table 2.1 Simulation of CMOS Selector using SPICE models.

2.3 Timing simulation and models

2.3.1 Timing simulation

Timing simulation is a simplified form of circuit analysis, whose objective is to provide waveform analysis for large MOS circuits. Timing simulators such as MOTIS [Cha75] and MOTIS-C [Fan77] exploit certain properties of MOS digital circuits in order to simplify the time and storage requirements of the simulator. These properties are the low activity in the circuit and the fact that many MOS configurations can be modelled as being unilateral in nature. The result is a simplification in the circuit equations and therefore in the numerical analysis. In addition, the model evaluation time is greatly reduced by using table look-up models for the semiconductor devices.

In the nodal circuit formulation for the time domain analysis, the node voltages may be expressed as:

$$f(\dot{v}, \hat{v}, t) = 0 \qquad (2.2)$$

The general structure of timing simulators in terms of a flow chart appears in Fig. 2.5.

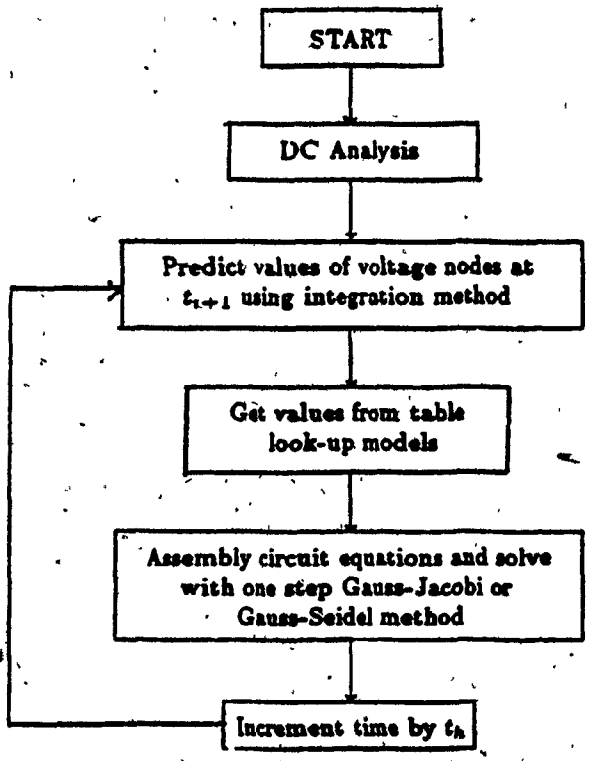


Figure 2.5 General structure of Timing simulators.

As in circuit simulators, the analysis interval time, T, is divided into small timesteps and an integration method is used to convert (2.2) into a set of nonlinear algebraic equations. One of the major simplifications of timing simulation is that the equations of (2.2) are *decoupled* [Hac81], in such a way that the voltage on each node is computed with the voltages of the other nodes held fixed. In other words, the circuit matrix solution of (2.2) is converted into a vectorial product. Timing

simulators use only a single Newton iteration step to solve the nonlinear system of equations. MOTIS uses a form of Gauss Jacobi iteration, by means of which only node voltages at t_n are used to evaluate the node voltage at t_{n+1}. In this method the order in which the elements are processed does not affect the results of the analysis, but substantial timing errors can occur. In order to maintain accuracy when using a single Gauss Jacobi iteration, the time step must be made small enough so that no substantive changes occur at a node during one time step. MOTIS-C, on the other hand, use a one step Gauss Seidel iteration in which nodes already computed at the timepoint are made available for the evaluation of other node voltages. It is evident that in the Gauss Seidel method the order in which the elements are processed can substantially affect the results.

Since the system equations are decoupled and each node voltage is solved independently of the other nodes at each timepoint, different techniques can be used to exploit the inactivity of the circuit. MOTIS and MOTIS-C use a form of a *bypass* scheme by which nodes that have not changed significantly are bypassed. The mixed-mode simulator SPLICE [New78], on the other hand, uses a *selective trace* approach, by which only the fanouts of a node whose voltage has changed are processed.

The equation decoupling scheme used in timing analysis is possible only when the circuit matrix is *diagonal dominant* [Hac81]. In terms of the circuit it means that a capacitor must be connected between each node and the reference node (GND), and that no bilateral elements are present (floating capacitors or transmission gates). In addition, stability problems arise in the solution of the Gauss-Jacobi or Gauss Seidel iteration when there is a strong bilateral coupling between nodes [New78], as in the latch of Fig. 2.6. In some cases it is possible to remove the instability by cutting the timestep so that the delay around the loop becomes larger than the

timestep itself. However, this approach does not work for MOS transmission gates or floating capacitors.

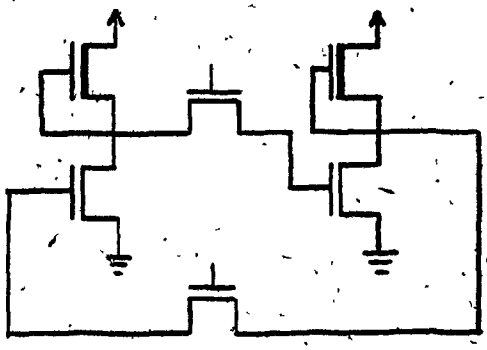


Figure 2.6 MOS LATCH with tight feedback.

In order to overcome the deficiencies of timing simulators in dealing with bilateral elements or tight feedback, mixed mode simulators such as MOTIS-2 [Che84], DIANA [DeM78], and SPLICE [New78] have been developed. These simulators use a mixed-mode approach, by means of which different parts of the circuit are simulated at different levels. In MOTIS-2, the program finds configurations such as transmission gates, floating capacitors, and tight feedback, and simulates them at the circuit level. SPLICE, on the other hand, allows the user to choose at which level different parts of the circuit are to be simulated.

2.3.2 MOS models for timing simulation

One of the essential features of timing simulators is the simplified transistor models, which consist of look-up tables rather than analytical models. Fig. 2.7 shows the equivalent transistor network used by MOTIS, MOTIS-C, and SPLICE, excluding the charge storage elements, which are always represented as constant capacitors in timing simulators [New78].

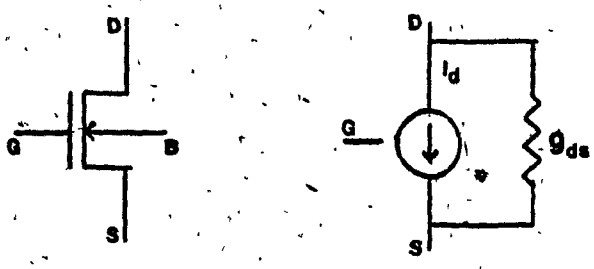


Figure 2.7 Transistor model used in timing simulators.

A straightforward modelling approach would require two tables

$$I_{ds}(v_{gs}, v_{ds}, v_{bs}) = T_d(v_{gs}, v_{ds}, v_{bs}) \tag{2.3}$$

$$g_{ds}(v_{gs}, v_{ds}, v_{bs}) = T_g(v_{gs}, v_{ds}, v_{bs}) \tag{2.4}$$

where T_d and T_g are three-dimensional tables with N entries for each controlling variable. It has been observed empirically that to obtain accurate results, N must be bigger than 50 [New80]. Therefore the storage requirements for such a model, N^3, become totally prohibitive. However, some transformations have been made in the model described above, in order to reduce the storage, while maintaining accuracy and improving efficiency in terms of evaluation time. In the program MOTIS, the effect of the back-bias potential is modelled as a shift in the threshold voltage, V_{th}. This effect is represented by a one-dimensional table

$$V_{th}(v_{bs}) = T_b(v_{bs}) \tag{2.5}$$

and the drain current is obtained by a two-dimensional table:

$$v_{gst} = v_{gs} - v_{th} \tag{2.6}$$

$$I_{ds}(v_{gs}, v_{ds}, v_{bs}) = T_d(v_{gst}, v_{ds}) \tag{2.7}$$

In addition, g_{ds} is approximated from the drain current and the associated load characteristic [Sub84]. The number of entries is, therefore, reduced to $N^2 + N$ per model. In MOTIS the bidimensional table T_{d_s} used to extract the drain current, has 64x64 entries. For NMOS circuits 4 tables are employed, one each for low threshold enhancement, high threshold enhancement, short channel depletion, and long channel depletion devices. In the case of CMOS there are two tables, one for p-channel devices and one for n-channel devices. Each table contains the drain current for a standard transistor [Sub84]. Fig. 2.8 shows a table model used in MOTIS for a low threshold enhancement transistor, for $v_{bs} = 5V$.

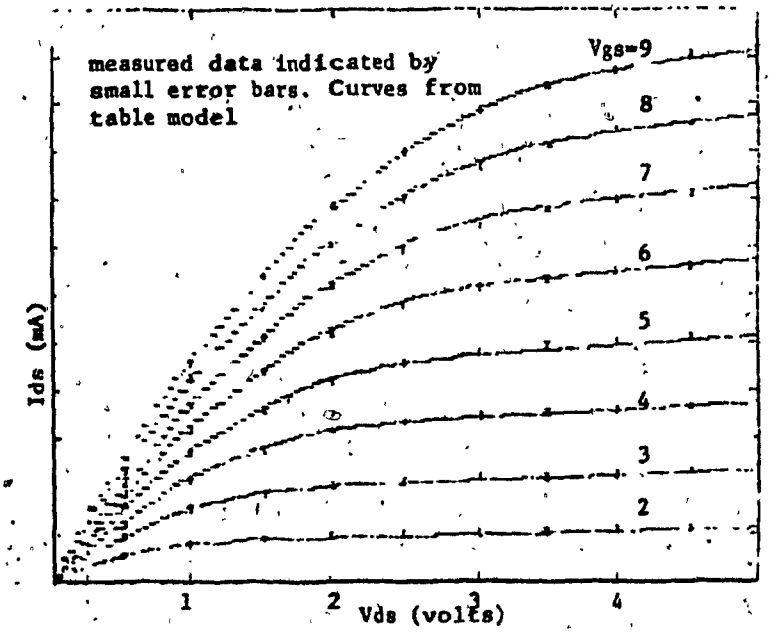


Figure 2.8 Table transistor model as used in MOTIS [Sub84].

The values for the tables are obtained by actual measurements on the devices or by circuit simulation using bidimensional transistor models [ElM77]. If sophisticated circuit models are used, the values for the current I_{ds} stored in the tables can include second order effects, maintaining therefore a good degree of accuracy in the models.

The transistor models in MOTIS-C and SPLICE have been reduced, by means of a number of transformations, to three one-dimensional tables, from which the drain current and the channel conductance are evaluated at any device operating point. Their model is summarized below [New80].

As in MOTIS

$$V_{th}(v_{bs}) = T_b(v_{bs}) \tag{2.8}$$

describes the shift in the threshold voltage due to the back-bias potential.

By means of a clever transformation, the current I_{ds} is expressed in terms of the voltage v_{ds}, holding the gate voltage v_{gs} fixed.

$$I_{ds}(v_{ds}) = T_d(v_{ds}, v_{gsmax}) \tag{2.9}$$

where v_{gsmax} is a constant which corresponds to the maximum permitted gate voltage, and I_{ds} is the channel current. Finally, the channel conductance, g_{ds} (fig. 2.6) is given by:

$$g_{ds}(v_{gs}) = T_g(v_{gs}) \tag{2.10}$$

T_b, T_d, and T_g are all one dimensional tables with N entries, therefore the storage requirement per model has been reduced to $3N$.

Table models are much faster to evaluate than analytical models. The evaluation of the MOS model in MOTIS is about 15 times faster than MOS1 in SPICE [New78]. Accuracy is within 5% of the circuit models although it can be improved by increasing the size of the tables. However, charge storage is modelled in timing simulators as constant capacitors, as compared to the voltage or charge-controlled models used by the circuit models, affecting mainly the accuracy of the obtained waveforms. The use of the table models described above, together with the solution method employed in timing simulation represent a speed improvement which is about one to two orders of magnitude over circuit simulation.

2.4 Gate level models and simulation

Gate level simulators such as TEGAS [Syz74] and SALOG [Cas78], rather than modelling the individual circuit elements, model groups of transistors which perform a logic function as a single, unidirectional block. These blocks are usually basic boolean gates, such as *NAND, NOR, EXOR* or more complex functions, such as *FLIP-FLOPs* and registers. Logic simulation uses a discrete set of logic values rather than the continuous voltage function used in circuit and timing analysis. This set may contain only the two basic logic values, 1 and 0. Most logic simulations, however, use additional logic states such as undefined, U and high impedance, Z. The logic evaluation in most gate-level simulators exploits the latency present in logic circuits in such a way that only gates in which at least one input has changed are simulated while ignoring the rest of the circuit. This process is called *selective trace* [Bre76]. Since in most large digital circuits, only a small number of nodes are changing at the same time, selective trace algorithms greatly enhance the simulation speed. Gate-level simulators run about 4 orders of magnitude faster than circuit simulators.

According to their structure, logic simulators can be either *compiler-driven* or *event-driven* [Bre76]. Compiler driven simulators such as SALOGS [Cas78], compile the description of the circuit into computer code. This approach does not fully exploit the inactivity present in logic circuits because scheduling is not used. In addition, it becomes very inefficient for fault simulation.

Most modern simulators are event-driven [Szy76], since they allow for more versatility in handling delays as well as a reduction of the simulation time. An event is defined as the change of logic state at an output node of an element (logic gate or input signal source). Event-driven simulators implement a time queue [Szy76], in which each entry represents a discrete point in time and contains a pointer to

a list of events which are to occur at that instant of time. The elements whose inputs have changed state are simulated to evaluate the value at their outputs. If the new state at the output of an element is different from the old state, the change is put on the event list and a pointer to this event is added in the time queue at the corresponding time. If the output of a processed element does not change state, due to an event at one or more of its inputs, then the elements connected to this output (fanouts) do not need to be processed.

The program module responsible for adding elements in the time queue is called the scheduler. An event is said to be scheduled at time t, if the element in the time queue points to this event at time t. By means of the scheduler and the time queue, propagation delays can be easily incorporated in event driven simulators.

Some gate level simulators do not include propagation delays. Other logic simulators allow unit delays, i.e., the value at the output of the gate appears one time unit after the inputs, and all the gates are assumed as having the same delay. However, by using a scheduler, preassigned gate propagation delays are introduced in logic simulators. Two different approaches can be used for that purpose, as shown in Fig. 2.9. A precalculated delay, which varies only with the output load, can be introduced at the input or at the output of the gate. For most logic gates, the propagation delay is a function of which of the inputs is changing, therefore, the input approach appears to be more suitable. In some logic simulators, delays are introduced both at the input and at the output of the gate.

This approach of pre-assigned delays, developed for TTL circuits, stems from the fact that TTL delays are mostly inertial. For MOS circuits, however, rise and fall delay times may be quite different and an assignable delay simulator must be able to assign different rise and fall delays to each gate.

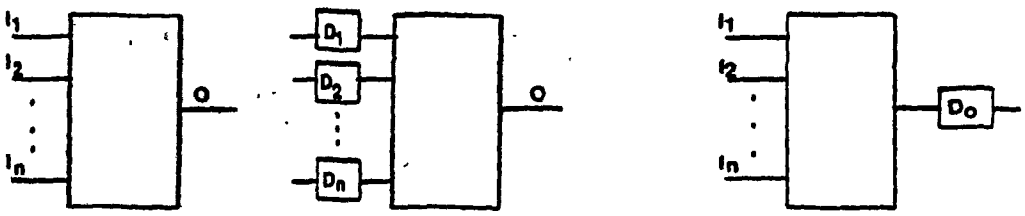


Figure 2.9 Introduction of delays in Gate level simulators.

An attempt to include the effect of signal waveform in NMOS gate level simulation was made by Tokuda et al. [Tok83]. Delays for a given gate are calculated during the preprocessing stage using the input and output capacitances, C_n and C_L, of the gate, and an averaged value for the current of the conducting device. Consider, for example, the simple NMOS inverter of Fig. 2.10. The basic assumption, based on SPICE simulation results, is that the output low-to-high transition delay, t_{lh}, is independent of the input waveform, whereas the output high-to-low transition delay t_{hl} is a function of the rate of change of the input signal. In the first case, t_{lh}, the delay is expressed in terms of the output capacitance only, whereas in the case of t_{hl}, both capacitances, C_L and C_n (which is used to quantify the input slope) are used to compute the delay. For a gate with more than one input, different delays are assigned for each of the inputs.

The expressions used by [Tok83] to calculate the delays are, however, complex and therefore slow in terms of computer speed. The level of accuracy reported for this model for NMOS circuits containing only logic gates is within 20% of SPICE [Tok83].

Gate level simulators were developed during the TTL era when systems were entirely designed with circuit structures which could be modelled by logic gates. These simulators are, however, inadequate for MOS circuits which include such configurations as bidirectional transmission gates, dynamic memory elements and

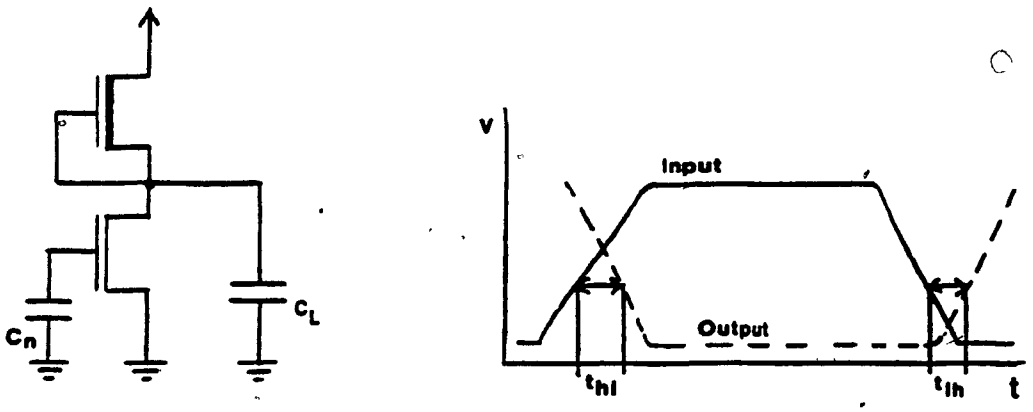


Figure 2.10 Rise and fall time delays of NMOS inverter.

precharged logic which can not be modelled adequately as a set of logic gates. As for delay calculations, the delay in MOS circuits depends on the dynamic loading conditions, and thus can not be properly evaluated using tabulated delays.

2.5 Switch-level simulation

2.5.1 Switch-level models and logic evaluation

Switch-level simulation [Bry80],[Hay80],[Ter83], [Bry84] can be viewed as a compromise between circuit and gate-level simulation for MOS circuits. In switch-level simulation the network is represented at the transistor level but each transistor is modelled in a highly idealized way as a simple, bidirectional switch. In this way the network model, which can be obtained directly from the layout, is based on the actual structure of the design, rather than on its intended function as it is in gate-level simulation. This network model allows switch-level simulation to handle typical MOS structures, such as transmission gates, precharged logic and dynamic storage, which can not be adequately modelled in gate level simulators.

Similar to gate-level simulators, switch-level simulators use a discrete set of logic values, rather than a continuous voltage function. MOSSIM [Bry80], RSIM [Ter83], and MOSSIM II [Bry84] employ a three value logic set 0, 1 or X, where

0 and 1 represent, respectively, the low and high logic voltages, and X represents an undefined state arising from uninitialized nodes, short circuits, or charge sharing. The CSA theory [Hay82], on the other hand, generalizes this approach to an expandable logic set, V, of cardinality $3k + 1$ where k is the number of levels of strength used. The smallest set ($k = 1$) is the four member set, $V_4 : \{Z, 0, 1, U\}$, where Z corresponds to the high-impedance state or disconnected, 1 and 0 are the high and low logic values and U denotes an undefined state. This basic set can be expanded to include any number of levels of strength. A strength is a measure of current drive capability.

The strength relation $\geq$, defined as *equal or greater than* imposes a partial ordering on V. The logic set V with the strength relation $\geq$ can be shown to form a lattice. The Hesse diagram of this lattice for V_7 ($k = 2$) appears in Fig. 2.11. In the lattice shown, 0, 1, and U represent the strongest values, whereas $\tilde{0}$, $\tilde{1}$, and $\tilde{U}$ represent weaker values. In general for $v_1, v_2 \in V$, $v_1 > v_2$ if and only if v_1 is at a higher level than v_2 in the Hesse diagram.

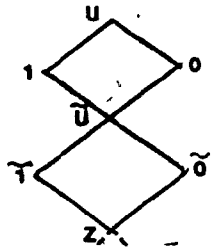


Figure 2.11 Expandable logic set used in the CSA theory.

Network Model

Transistors are modelled as three-terminal devices labeled respectively *gate*, *source* and *drain*. No distinction is made between the source and the drain connections. Each transistor is symmetric and bidirectional and acts as a switch connecting

or disconnecting its source and drain according to the state of the gate and the type of the transistor. Each transistor is assigned a *strength* from a discrete set according to the transistor size, as a measure of its current drive capability, so that ratio logic such as NMOS can be modelled. Short channel transistors have larger strength than long channel transistors of the same width.

Nodes are classified as either *input node* or *storage node*. An input node provides a strong signal to the network (VDD, GND, clocks, data), and its state is not affected by the operation of the network much as a source in an electrical circuit. The state of storage nodes, on the other hand, is determined by the operation of the network, much as a capacitor in an electrical circuit. A storage node holds its state in the absence of connection to other nodes so that dynamic storage is ideally modelled. Each storage node can be assigned a *size* from a discrete set, according to its capacitive value relative to the capacitance of the other nodes with which it may share charge. A larger storage node is assumed to have much greater capacitance than a smaller one. In this way a simple model for charge sharing is provided. When a set of storage nodes share charge, the state of the largest node overrides the state of the smaller nodes.

Logic Evaluation

Most switch-level simulators perform a partition of the network prior to the simulation. MOSSIM ,RSIM, and McSLADE partition the circuit, at the preprocessing, into *transistor groups*, in which the switches are interconnected only via their source and drain terminals and some are connected to the supply VDD or to ground. A MOS circuit partitioned into transistor groups appears in Fig. 2.12. MOSSIM II performs a dynamic partition during simulation [Bry84].

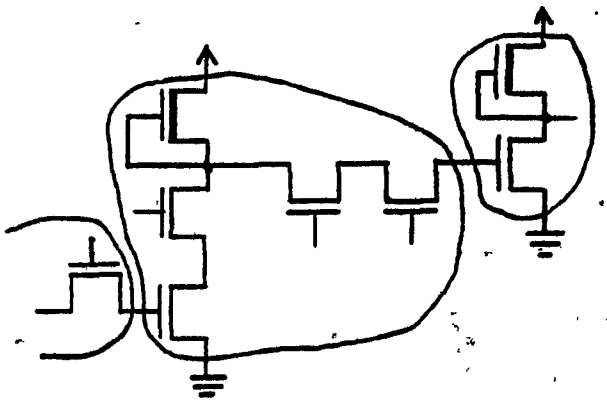


Figure 2.12 Circuit partitioned into transistor groups.

A transistor group is evaluated whenever an event occurs at one of its inputs. The simulation algorithm consists of applying a set of input vectors and propagating the changes due to that vector from one transistor group to the next, until the network reaches a stable state. Only groups in which one or more of its inputs have changed are evaluated.

The logic evaluation inside a transistor group consists of propagating the logic values from the sources through the closed switches. As mentioned above, each switch has an associated strength, according to its size. When a signal propagates through a switch it is attenuated by a factor which is inversely proportional to the strength of the switch. For example, A minsize transistor ($Length/Width = 1/1$ [Mea80]) has a lower attenuation than a transistor with an L/W ratio of 12/1. A logic value can then be defined as a state-attenuation pair, S^{α}, where $S \in \{0, 1, U\}$ is the state and $\alpha \in \{0, \cdots \infty\}$, is the attenuation. With reference to the lattice V of Fig. 2.11, each α_i can be viewed as a level of strength.

To find the logic value of a node, the *Least Upper Bound (LUB)* operation is applied to all the incoming state-attenuation pairs that drive the node [Hay82]. This operation is defined as follows: Let a set of signals, $s_1, s_2 \cdots s_n$ defined on

V be applied to a node c (where V is the CSA logic set described above). The node c assumes the value Z_c, where Z_c is the weakest member of V, such that $Z_c \geq \{s_1, s_2 \cdots s_n\}$ (where $\geq$ has been defined above). The attenuation factor of a signal propagating from a source (VDD or GND) through a path of transistors, corresponds to the attenuation factor of the weakest transistor in the path (which has the largest attenuation).

As an example, consider the simple NMOS inverter of Fig. 2.13. The minsize pull-down transistor, $T1$, has an attenuation, α_1, and the 4, 1 pull up depletion transistor, $T2$, has an attenuation α_2. When the input of the inverter is equal to 1, the logic state 0 propagates through $T1$ whereas the logic state 1 propagates through $T2$. Because $\alpha_1 < \alpha_2$, the attenuation of the 1 is larger than that of the 0. The two state-attenuation pairs interacting at the output of the inverter are: 0^{α_1} and 1^{α_2}. Therefore, in the Hesse diagram of Fig. 2.11, the state 0 belongs to a higher level of strength than the state 1. The LUB operation then determines the state of the output node of the inverter to be 0^{α_1}.

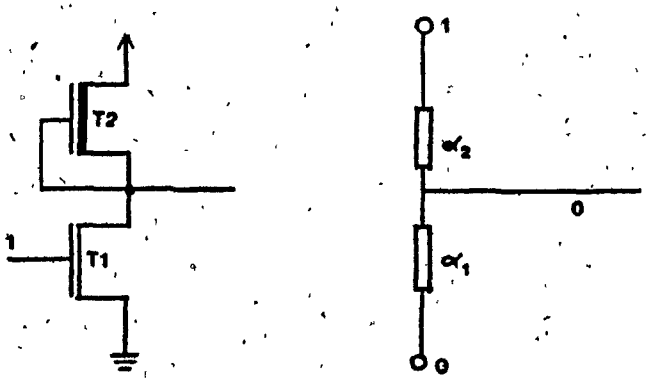


Figure 2.13 Logic evaluation for an NMOS inverter

2.5.2 Delay calculation in switch-level simulation

The traditional approach for introducing timing analysis in logic simulators has been to use assignable delays. However, as explained in Section 2.4, this approach

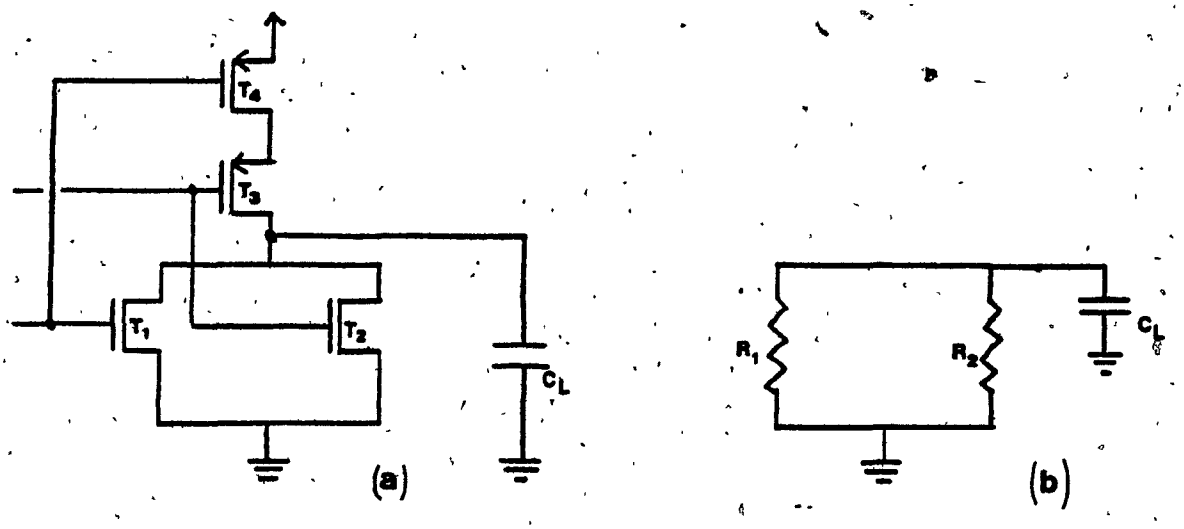


Figure 2.14 (a) CMOS NOR gate with the inputs at 1.
(b) Equivalent RC network.

does not work well for MOS circuits, because delays in such circuits, are very dependent on dynamic loading conditions and the slope of the waveforms on the gate input of the transistors. Moreover, the wide variety of circuit structures in custom VLSI limits the usefulness of the assignable delay approach.

For these reasons it is desirable to incorporate delay estimation algorithms in switch-level simulation. Such algorithms must be simple, so that they can be used for simulation of large circuits, and at the same time, they must predict delays with an accuracy which is not worse than that obtained with gate level simulators.

The delay between two nodes in a circuit is usually defined as the interval time from the signal transition at the first node crossing a threshold to the signal transition at the second node crossing the same or another threshold. The traditional model which has been used to represent the timing behaviour of MOS circuits is the *RC model* [Mea80], [Pen81]. In this model each ON transistor is represented by an effective linear resistance and each node by a capacitor tied to ground. Transistors in the OFF state are modelled as open circuits. Fig 2.14(b) shows the equivalent RC network for the CMOS gate of Fig. 2.14(a) when all its inputs are at logic 1.

Two different aspects must be considered when using the RC model for delay calculation in MOS circuits. The first one is related to the development of efficient algorithms to calculate delays in RC networks. The second aspect refers to modelling the ON transistors as effective resistances.

Delay Calculation methods

One of the first reported attempts at calculating delays in switch-level simulators on the basis of RC networks was RSIM [Ter83]. In RSIM, the delay of a given connected component subnetwork is obtained using Thevenin equivalents. The delay, t_d, at the output node of the subnetwork is given by

$$t_d = R_{thev} C_{sum} \tag{2.11}$$

where R_{thev} is the equivalent Thevenin resistance, and C_{sum}, is the sum of all the capacitances in the subnetwork, as shown for the circuit of Fig. 2.15. This approach results in overestimation of the delay for most practical cases as it suggests that all the resistances in the network discharge through all the capacitances.

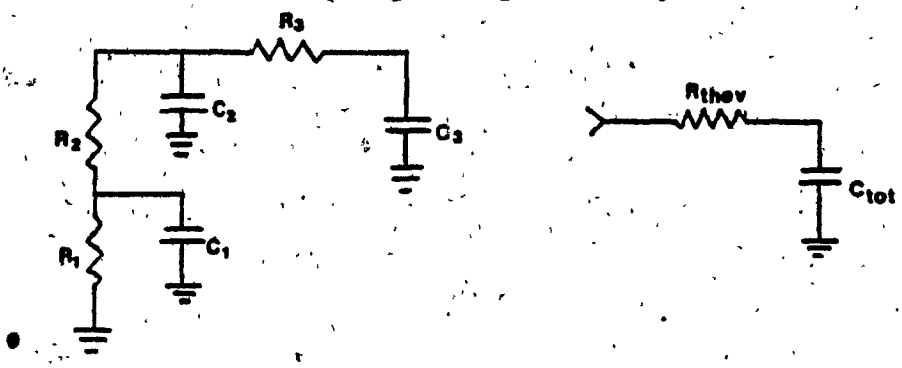


Figure 2.15 Thevenin equivalent of RC network as used by RSIM.

RC Trees

An RC tree-network is one in which there is only one driving source, which is the root of the tree, and only one path between each two nodes. All the capacitances in the tree are tied to ground. An example of such a network appears in Fig. 2.16.

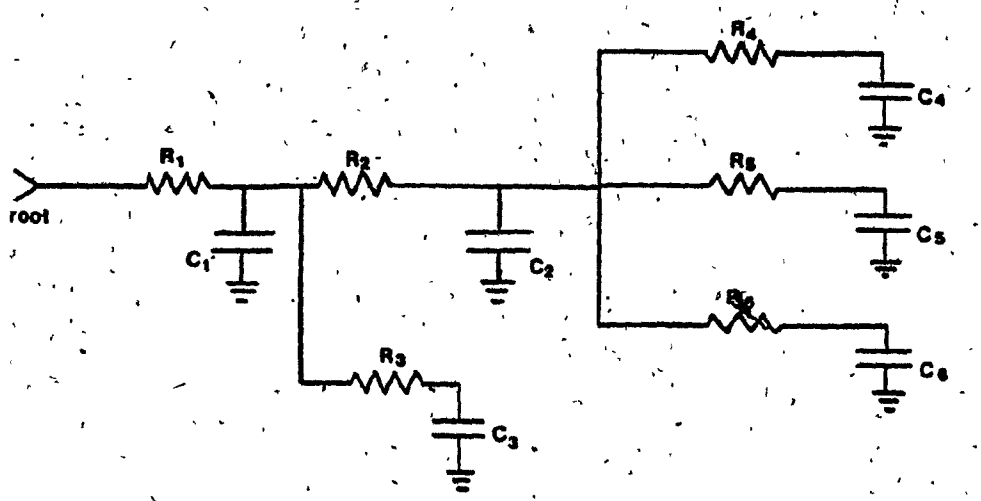


Figure 2.16 Example of RC tree.

Assuming that a step function is applied at the source of the tree, the typical shape of the voltage response at node i has a monotonical form as shown in Fig. 2.17(a) [Rub83]. Moreover, with the exception of the first node (the closest to the root) or possibly the first two nodes, the waveforms at all the nodes in the tree are approximate replicas of each other, but displaced in time as shown in Fig. 2.17(b).

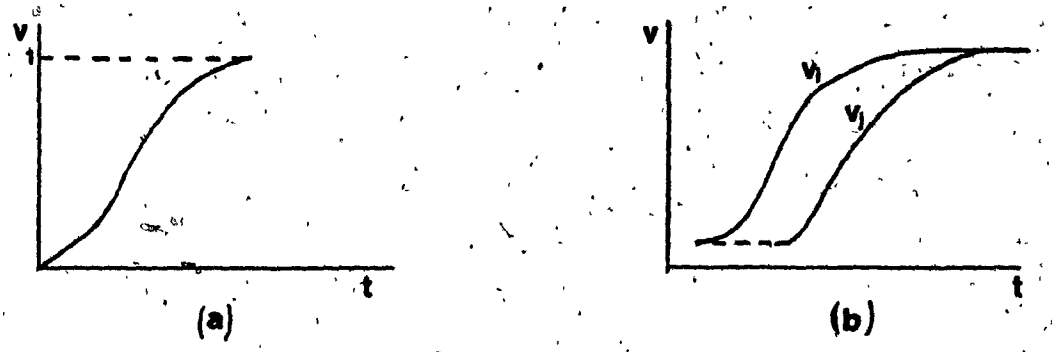


Figure 2.17 (a) Waveform shape at node i of a tree. (b) Waveforms of two adjacent nodes displaced in time.

In such case, the delay between nodes i and j is given by the displacement

in time t_{ij} which can be computed as follows (Fig. 2.17)

$$t_{ij} = \int_0^\infty (1 - v_j(t))\,dt - \int_0^\infty (1 - v_i(t))\,dt \tag{2.12}$$

from Rubinstein et al. [Rub83].

$$\int_0^\infty (1 - v_i(t))\,dt = T_{D_i} \tag{2.13}$$

where

$$T_{D_i} = \sum_{k=1}^{n} R_{ki} C_k \tag{2.14}$$

n is the number of nodes in the tree, C_k is the capacitance at node k, and R_{ki} is the resistance of the portion of the unique path between the source and node k that is common with the unique path between the source and node i. The delay between i and j can thus be expressed as:

$$t_{ij} = T_{D_j} - T_{D_i} \tag{2.15}$$

where T_{D_i} and T_{D_j} are the delays from the source of the tree to nodes i and j respectively. As Penfield and Rubinstein have shown [Rub83], Eq. (2.14) is equal to the first-order moment of the impulse response (of the RC network), which has been called *delay* by Elmore [Elm48]. The formulation of delay given by Eq. (2.14) is valid only if the waveforms at the nodes of the tree are equal in shape as shown in Fig. 2.17(b). In case it is not, the delay estimated is always conservative [Lin84].

Although the Elmore delay formula is not related to a voltage threshold, if the effective resistances of the transistors are extracted with respect to a threshold, Eq. (2.14) can be effectively used to calculate delays in MOS circuits. (See Sec. 3.2). In this case, T_{D_i} corresponds to the delay from a signal propagating from the

root of the equivalent RC tree to node i. For example the delay from the source to node n_4 in the circuit of Fig. 2.16 is given by:

$$t_{d_4} = R_1(C_1 + C_2 + C_3 + C_4 + C_5 + C_6) + R_2(C_2 + C_4 + C_5 + C_6) + R_4C_4 \quad (2.16)$$

Intuitively, the expression of Eq. (2.16) is more realistic than the one in Eq. (2.11), because it is equivalent to saying that each capacitor charges/discharges only through the resistive path which lies between it and the source.

General RC networks

A general RC network may contain reconvergent paths, i.e. two nodes that are connected by more than one path, as the circuit shown in Fig. 2.18.

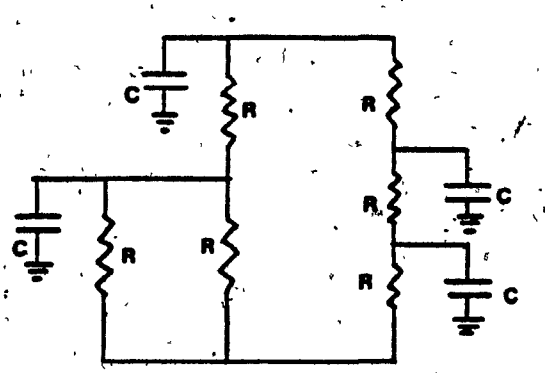


Figure 2.18 RC network containing reconvergent paths.

Lin and Mead [Lin84] generalized the Elmore delay formula of Eq. (2.14) to general RC networks. Their solution consists of decomposing the RC network into an RC tree or trees by *node-splitting*, that is some nodes in the network are split into two or more nodes in order to break the reconvergent paths and obtain a tree. As Lin and Mead have shown, such a decomposition always exists. An example of a decomposed RC network appears in Fig. 2.19. The nodes of the original network are called *primary nodes*, and those in the resulting tree are called *secondary nodes*.

The set of secondary nodes which corresponds to the same primary node are called *equivalent secondary nodes*. In the network of Fig. 2.19(b), n_i is a primary node and n_{ik} : $k = 1, \cdots j$, are equivalent secondary nodes.

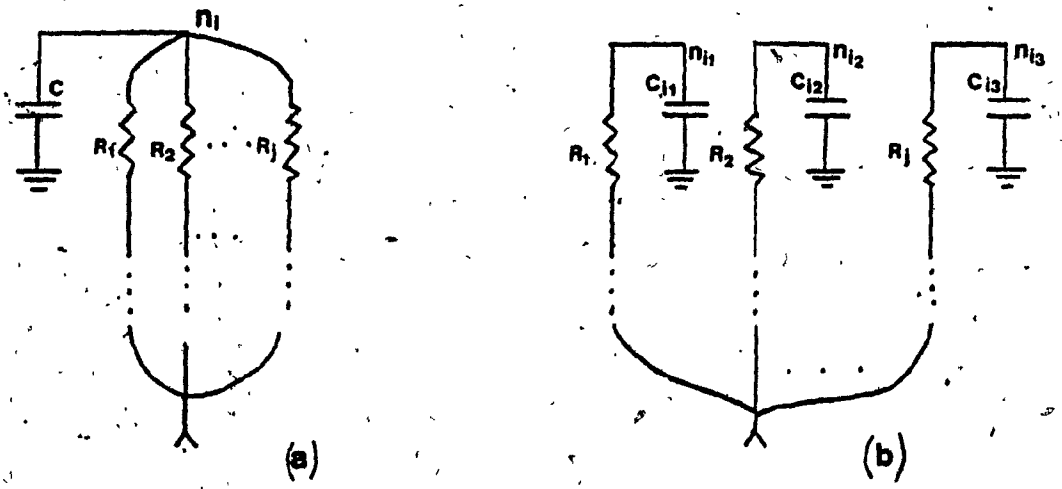


Figure 2.19 (a) Network with j incident branches on node n_i.
(b) Network after node splitting.

Equivalent secondary nodes must have the same delay and this delay must be equal to the delay at the corresponding primary node in the original network. The delays of the secondary nodes are obviously dependent on the values of the capacitances of those nodes. Finding the delays in the original network is equivalent to finding the right capacitance distribution on the secondary nodes. This is done by solving the following set of algebraic, linear equations:

$$t_{i_1} = t_{i_2} = \cdots = t_{i_{j_i}} \quad \forall i, \quad i = 1, \cdots n \tag{2.17a}$$

$$\sum_{k=1}^{j_i} C_{ik} = C_i \quad \forall i, \quad i = 1, \cdots n \tag{2.17b}$$

where t_{ij} is the delay of the secondary node n_{ij} corresponding to the primary node n_i in the original network. C_{ik} is the capacitance of the secondary node n_{ik}, C_i

is the capacitance of node n_i in the original network, n is the number of nodes in the original network that have been split, and j_i is the number of secondary nodes corresponding to the primary node n_i. The expression of Eq. (2.17a) means that the delay at the equivalent secondary nodes is equal, whereas Eq. (2.17b) states that the sum of the capacitances on the equivalent secondary nodes is equal to the capacitance of the corresponding primary node. The system of equations presented in Eq. (2.17) could be solved by using an exact solution method, such as Gaussian elimination. However, this is time consuming and an alternative relaxation method has been proposed by Lin and Mead [Lin84].

Switch-Level Delay Modelling

The second aspect of delay estimation in switch-level simulators using the RC representation of the network refers to the model, on the basis of which the transistors are replaced by effective resistances. There are two types of switch-level delay models, *static* and *dynamic*. Static models refer to the case in which the effective resistance of the transistor is modelled only as a function of static conditions such as size and type of the transistor. Dynamic models, on the other hand, model the transistor according to specific conditions of the circuit at a given time. This approach includes such effects as the load driven by the transistor and the slope of the gate waveform, when computing the value for the effective resistance. The models developed by Terman [Ter83] and Lin [Lin84] belong to the first category. The model developed by Ousterhout [Ous84] belongs to the second category.

According to Terman's model, the effective resistance, R_{eff} is a function of the size of the transistor and the static context. His model recognizes 5 different contexts as shown in Fig. 2.20. In order to obtain the value of the effective resistance for each of the cases of Fig. 2.20, Terman performed simple SPICE simulations, one in each context, using a constant capacitive load.

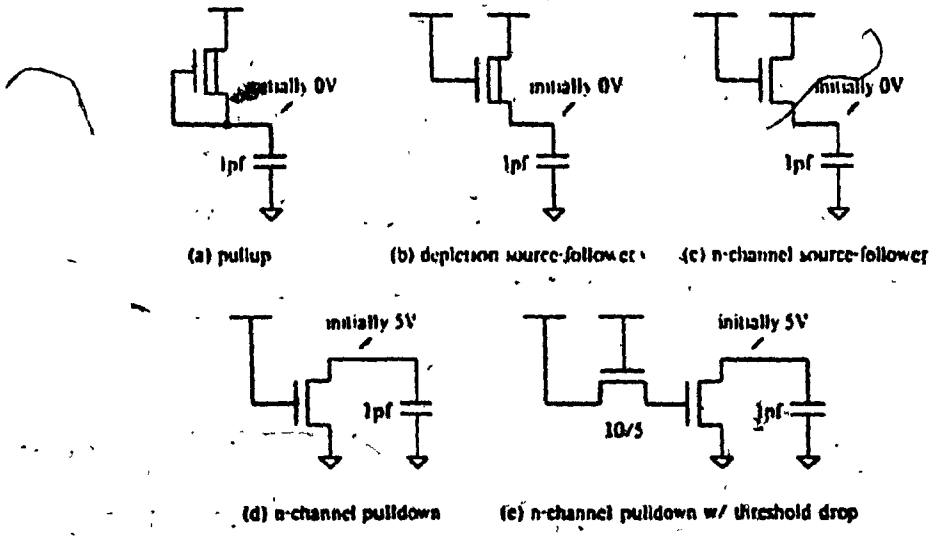


Figure 2.20 Transistor contexts as defined in Terman's model [Ter83].

There are three main sources of inaccuracy in the method of modelling used by Terman.

- The effective resistance is assumed to be independent of the load driven by the transistor. In the case of the n-channel pulldown driving a smaller load than the one used in the extraction of R_{eff}, the result is an underestimation of the actual value.
- The effective resistance does not include the effect of the input waveform. In the case of the pulldown transistor driven by slow signals, a considerable error is introduced in the delay estimation.
- The context used for extracting the value of R_{eff} for the pulldown is not realistic, because in NMOS circuits pulldowns always have a pullup connected to them. The presence of the pullup, strongly affects the value of the effective resistance of the pulldown. For a 4/1 inverter, the difference might be around 20%.

Lin and Mead also used an oversimplified approach to model the transistor as

an effective resistance [Lin81]. The value of R_{eff} is extracted from an 4/1 inverter chain as shown in Fig. 2.21.

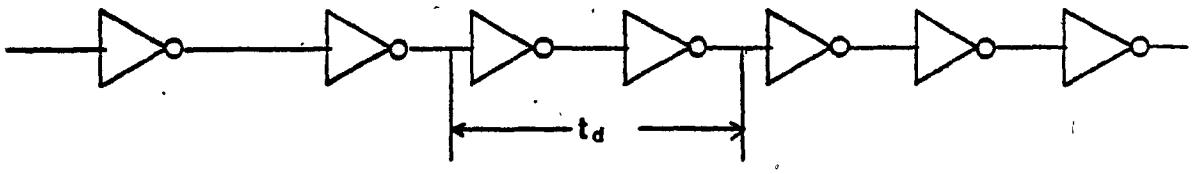


Figure 2.21 Circuit used by Lin and Mead to extract the effective resistance [Lin84].

The inverter pair delay T_d is obtained from SPICE simulation results. It is assumed that the output low-to-high propagation delay is 4 times the output high-to-low delay. The effective resistance, assumed equal on a per/sqr basis for the two transistors, is computed as:

$$R_{eff} = \frac{T_d}{5C_L} \frac{k\Omega}{sqr} \tag{2.18}$$

For any transistor in the circuit, the effective resistance is obtained simply by multiplying the value obtained in Eq. (2.18) by the number of squares of the transistor area. The sources of inaccuracy present in this model are the same as in Terman's model except for the context, which in Lin's model appears more realistic.

Although the static models developed by Terman and Lin present serious inaccuracies, they work reasonably well for NMOS circuits, due to the presence of the big pullup devices, which determine most of the delay in a digital NMOS circuit. It has been observed that the effective resistance of a depletion transistor used as a pullup load is essentially independent of the load (Chap3), and the slope [Tak83]. This is not the case for CMOS circuits, where such a simple modelling method introduces appreciable errors in the delay calculations.

A significant contribution to switch level modelling was made by Ousterhout in the formulation of the models for the timing analyzer CRYSTAL [Ous83]. CRYSTAL was designed to calculate delays in critical paths of MOS digital circuits. In Ousterhout's model the effective resistance of a transistor includes the effect of the input waveform, in addition to the size and type of the transistor. To introduce the effect of the input waveform in the computation of the effective resistance of the transistor, Ousterhout developed the concept of *rise-time ratio.* This ratio is defined as the rise time of the input signal divided by the intrinsic rise time of the stage. The latter is the rise time of the signal appearing at the output of the stage when the input is a step function. The rise-time ratio gives an estimate of how much a transistor can be affected by a particular input slope, *independent of the load and size of the transistor.* The effective resistance of each transistor is obtained from a single table which contains the effective resistance as a function of the rise time ratio. Ousterhout divides the transistors into five different types: enhancement, enhancement driven by a pass transistor, depletion load, super buffer and depletion transistor. For some of them, there is a subdivision depending on whether the transistor is transmitting a 1 or a 0. The introduction of the effect of the slope in the evaluation of the effective resistance of a transistor represents the major contribution of Ousterhout's model. However, the model is deficient in the following areas:

- The value of the effective resistance of the transistor is not adjusted with the capacitive load.
- The objective of the model developed by Ousterhout was timing analysis rather than switch-level simulation. The waveform rise-time used by his model is not available in switch-level simulators, where only delays are known.

The model developed by Ousterhout yields very accurate results (within 10%) for NMOS circuits. No results have been reported about the model's performance with CMOS circuits which pose more complicated modelling problems than NMOS.

The objective of the new switch-level delay model is to overcome the deficiencies present in the previous reported models. It must be computationally effective, so that large circuits can be simulated and it must evaluate delays in CMOS circuits with a good degree of accuracy.

CHAPTER 3

THE DELAY MODEL

3.1 Introduction

The new-switch level simulator, McSLADE [Kho85], in which the delay model to be described here was implemented, models the MOS circuit as a combination of a switch network and a linear network. The switch network, in which transistors have been replaced by bilateral switches, is used for the logic evaluation as described in [Kho85]. The linear network used for the delay calculations, on the other hand, is based on the RC model. Because the delay calculation must be repeated every time a switch changes state, the objective is to derive simple linear approximations to the complex nonlinear dynamics of the transistor. The capacitances of the RC model are computed on the basis of the topological data of the circuit at the preprocessing step and remain constant during the simulation cycle. Therefore, all the effects that lead to variation in the delay are modelled only by the effective resistance, R_{eff}. The value of R_{eff} is calculated during simulation on the basis of the circuit's activity. It is shown to depend not only on the size and type of the transistor, but also on the context, its capacitive load and the slope of the gate input waveform. To establish a relationship between these factors and the effective resistance of the transistor, circuit simulations were performed on basic configurations, using SPICE2G6 [Nag75] and the MOS2 model (Sec. 2.2). The simulation results are the key to building the transistor models.

This chapter describes the extraction and use of the new delay model. As a starting point, a definition of delay is given in Section 3.2. The model for the node capacitances is presented in Section 3.3. Section 3.4 derives a relation between the effective resistance and the main factors that influence delays in MOS circuits. The incorporation of the model in the simulator McSLADE is described in Section 3.5. Section 3.6 illustrates the configurations on the basis of which the model was extracted, and describes the changes that must be made in order to update it for changes in the device technology.

3.2 Definition of Delay

Prior to the building of the model it is necessary to have a consistent and unambiguous definition of delay. *The delay is defined as the interval between the time the input transition crosses a threshold and the time when the output transition crosses the same threshold*, as shown in Fig. 3.1(a). A major consideration in the above definition is the selection of proper thresholds. The threshold voltage of a given gate is defined here as *the voltage V_{th} at which the input voltage, V_{in}, equals the output voltage, V_{out}, in the transfer characteristic*, as shown in Fig 3.1(b)

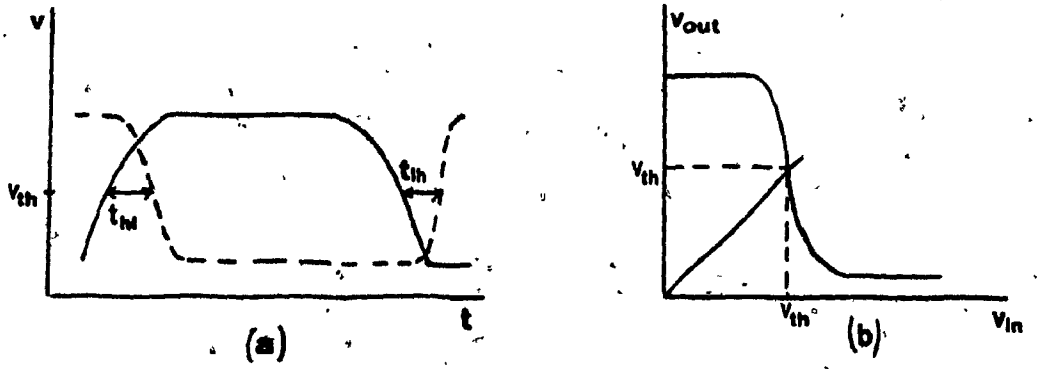


Figure 3.1 Definition of delay. (a) Rise and Fall delay times. (b) Threshold definition.

A problem arises when cascading gates with different threshold levels. In the circuit of Fig. 3.2(a), gates A and B are assumed to have different thresholds, V_{th_a} and V_{th_b}. If $V_{th_a} > V_{th_b}$, the time T is counted twice when computing the delay from a to c, as shown in Fig. 3.2(b). Otherwise, if $V_{th_a} < V_{th_b}$, the time T is not accounted for, as shown in Fig. 3.2(c).

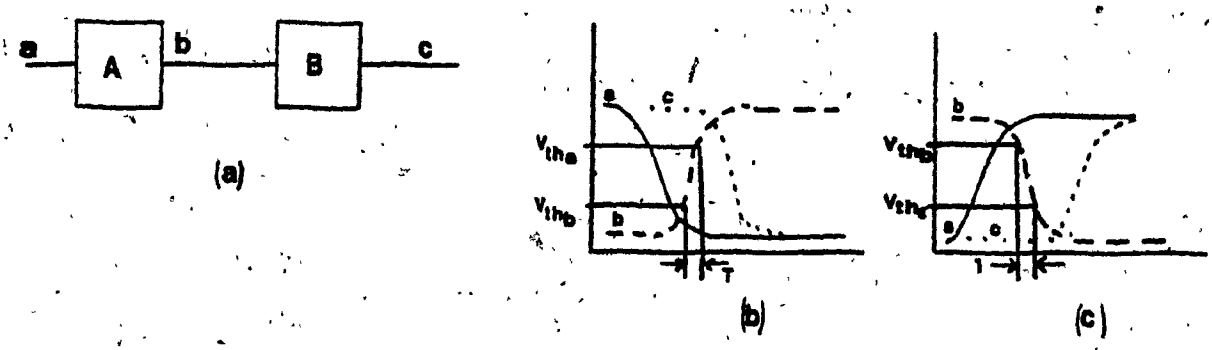


Figure 3.2 (a) Cascading gates with different thresholds. (b) Time T accounted for twice. (c) Time T unaccounted for.

This problem can be avoided by using a *single threshold* for the entire circuit. This threshold must be chosen in such a way that the delays in the different gates of the circuit are always positive. A reasonable solution used here and also adopted by Nham and Bose [Nha80] is to use the threshold of a standard minsize inverter for the whole circuit. By employing this single threshold, most of the gates are assured to have positive delays, although for structures with very different transfer characteristics, a negative delay may result. This is illustrated in the circuit of Fig. 3.3. When a slow signal is applied to the input of the gate, the transition at the output might cross the threshold V_{th_s} before the input does, causing a negative delay for the gate.

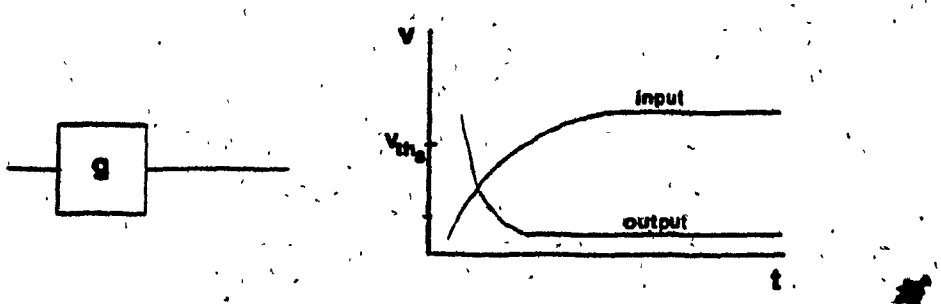


Figure 3.3 Negative delay occurs at the output of G.

The purpose of the above definition of delay is twofold: it will be used in the experimental circuits aimed at extracting the model of the effective resistance and it will also be applied to the test circuits used to assess the model.

3.3 The model for the capacitances

Fig. 3.4(a) shows the MOS junction and distributed thin-dioxide capacitances as used by the SPICE models. The source, drain, and bulk overlap capacitances are shown in Fig. 2.4(b). The same capacitances appear in all the three models, MOS1, MOS2, and MOS3 [Vla79], although their values are computed differently. The capacitances of the RC model used to calculate the delays, on the other hand, must be constant and grounded. The objective is, therefore, to derive a model in which all the MOS capacitances are lumped between the transistor terminals and ground, as shown in Fig. 3.4(b).

There are three capacitances associated with each transistor, C_g, C_s and C_d, connecting, respectively, the gate, source, and drain to ground. Although this model is a rough approximation, it will be shown that, as far as delay calculations are concerned, the lumped model yields very accurate results.

3.3.1 Gate Capacitance

C_g, the gate capacitance, is formed by lumping at the gate the distributed elements of the nonlinear gate-channel thin-dioxide capacitance, as shown in Fig. 3.5.

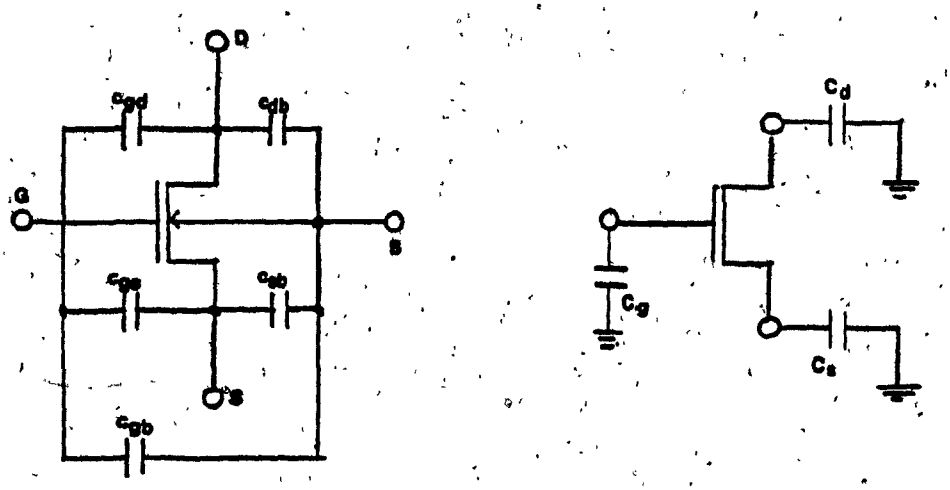


Figure 3.4 (a) MOS Capacitances. (b) Switch-level model.

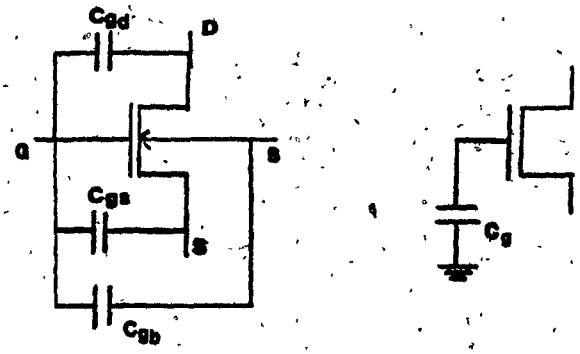


Figure 3.5 Lumped model for the gate capacitance.

A reasonable approximation for C_g is given by the following expression [Hod83]:

$$C_g = C_{ox} W L = \left(\frac{\epsilon}{tox}\right) W L \qquad (3.1)$$

where W and L correspond respectively to the channel width and length, as they appear in the actual layout of the transistor, ϵ is the permitivity of the silicon dioxide and tox is the thickness of the silicon dioxide layer. These two parameters,

ϵ and tox, must be supplied by the user based on technology data. Expression (3.1) is adopted by the new delay model.

3.3.2 Source and Drain capacitance

The source-ground and drain-ground capacitances, C_s and C_d of Fig. 3.4(b), are formed by lumping the source and drain overlap capacitances and the nonlinear, voltage-controlled drain-substrate and source-substrate junction capacitances respectively. They are evaluated as follows [Hod83]:

$$C_d = K_{eq} C_{sb} + C_{gso} \qquad (3.2.a)$$

$$C_s = K_{eq} C_{db} + C_{gdo} \qquad (3.2.b)$$

where C_{gso} and C_{gdo} represent the drain and source overlap capacitances. C_{sb} and C_{db} correspond to the zero-bias, source-substrate and drain-substrate junction capacitances respectively. K_{eq} is a dimensionless parameter used to linearize the zero-bias junction capacitors with the change in the voltage level. The right-hand terms of (3.2) are evaluated using technology and layout parameters as follows:

$$C_{sb} = C_{jo}\, AS + C_{jsw}\, PS \qquad (3.3.a)$$

$$C_{db} = C_{jo}\, AD + C_{jsw}\, PS \qquad (3.3.b)$$

where AS, AD, PS, and PD correspond, respectively, to the area (A) and perimeter (P) of the source (S) and drain (D) regions, obtained directly from the layout of the transistor. C_{jo} and C_{jsw} are the junction, zero-bias, bottom and sidewall capacitances per unit area and unit perimeter, respectively. These capacitances are technology-dependent and must also be supplied by the user.

The definition of K_{eq} for a voltage controlled capacitance is given by:

$$K_{eq} = \frac{C_{eq}}{C_j} \tag{3.4}$$

where C_j is a junction zero-bias capacitance such as C_{sb} and C_{db}. C_{eq} is the equivalent capacitance defined as the voltage independent capacitance that requires the same change in charge as the nonlinear capacitance, for a given voltage range. That is

$$C_{eq} = \frac{\Delta Q}{\Delta V} = \frac{\{Q(V_2) - Q(V_1)\}}{V_2 - V_1} \tag{3.5}$$

where

$$\Delta Q = \int_{V_1}^{V_2} C_j(V)\, dV = \int_{V_1}^{V^2} \left[\frac{1 - V}{\Phi_0}\right]^{-m} dV \tag{3.6.a}$$

and

$$\Delta V = V_2 - V_1 \tag{3.6.b}$$

C_{jo} is the junction capacitance at $V = 0$, and m is the junction grading coefficient. Φ_0 is the junction barrier potential developed across the depletion region, and depends on the temperature and the impurity concentration on either side of the junction. Φ_0 and m are technology-dependent user- supplied parameters. V_2 and V_1 correspond to the high and low voltages of the voltage swing. From (3.5) and (3.6)

$$C_{eq} \quad \frac{-C_{jo}\Phi_0}{(V_2 - V_1)(1 - m)} \left(\frac{1 - V_2}{\Phi_0}\right)^{1-m} \tag{3.7}$$

and from (3.4)

$$K_{eq} = \frac{C_{eq}}{C_{jo}} = \frac{-\Phi_0}{(V_2 - V_1)(1 - m)} \left(\frac{1 - V_2}{\Phi_0}\right)^{1-m} \tag{3.8}$$

The switch-level capacitance model can now be evaluated provided that all the technology and layout parameters be supplied. These parameters are summarized in Table 3.3.1. The expressions for the complete switch-level capacitance model are summarized in table 3.2

Symbol	Parameter
C_j	Bottom zero-bias junction capacitance per unit area
C_{jsw}	Sidewall zero-bias junction capacitance per unit perim.
Φ_0	Junction barrier potential
M_j	Bottom capacitance grading coefficient
M_{jsw}	Sidewall capacitance grading coefficient
tox	Thickness of silicon dioxide layer
ϵ_{ox}	Permitivity of silicon dioxide
K_{eq}	Linearizing coefficient
C_{ox}	Gate capacitance per unit area
C_{gdo}	Gate-drain overlap capacitance per unit length

Table 3.1 Technology data for capacitance Model

Capacitance	Expression
C_g	$C_{ox} W L$
C_d	$K_{eq}(C_j AD + C_{jsw} PD) + C_{gdo}$
C_s	$K_{eq}(C_j AS + C_{jsw} PS) + C_{gso}$

Table 3.2 Switch-level capacitances

The simulator computes the values of the capacitances at the preprocessing step, using the equations of Table 3.2. The Technology data of Table 3.1 is stored in a special file read by the simulator, called *the technology file*. The geometry parameters of the transistor are supplied either by a circuit extractor or by specifying them in the input file.

Example 3.1

Fig. 3.6(a) shows a circuit containing two cascaded CMOS static inverters. From the circuit layout which appears in Fig. 3.6(b), the geometry dimensions of the transistors can be extracted. Technology data is taken from the Northern Telecom CMOS-1B process [Smi83].

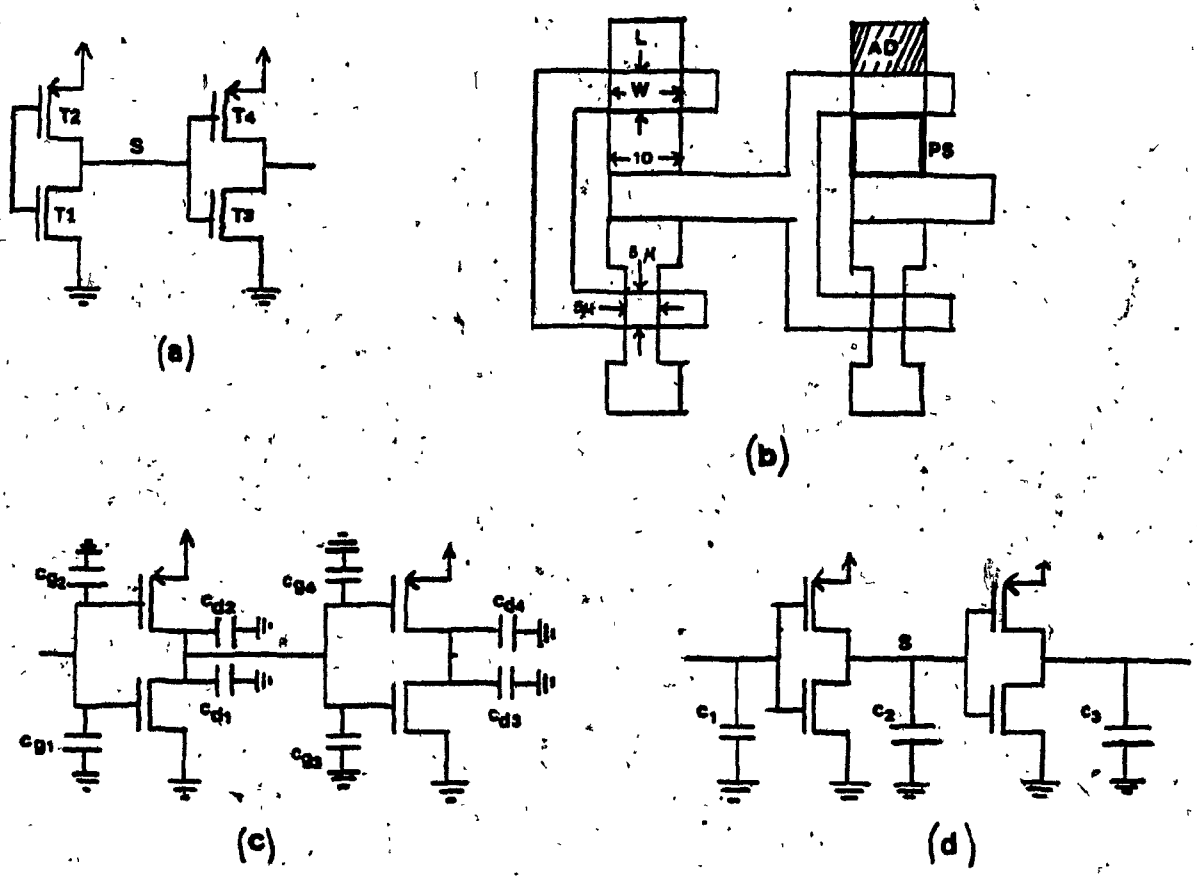


Figure 3.6 (a) Cascaded CMOS inverters. (b) Symbolic Layout of the inverters. (c) Switch-level transistor capacitances. (d) Reduced switch-level capacitance model.

The switch-level capacitance model is shown in Fig. 3.6(c). The values for the different capacitances are evaluated as follows.

Gate capacitances:

$$C_{g1} = C_{g3} = WLCox = 5\mu 5\mu 4.1\,10^{-4}\frac{F}{m^2} = 0.01pF$$

$$C_{g2} = C_{g4} = WLCox = 5\mu 10\mu 4.1\,10^{-4}\frac{F}{m^2} = 0.02pF$$

Drain Capacitances:

$$C_{d1} = C_{d3} = K_{eq}(C_j AD + C_{jsw} PD) + C_{gdo}$$

$$= 0.123\left(0{,}4\,10^{-3}\frac{F}{m^2}75\,10^{-12}m^2 + 0.8\,10^{-9}\frac{F}{m}30\,10^{-6}m\right) + 0.001pF$$

$$= 0.0076pF$$

in a similar way

$$C_{d2} = C_{d4} = 0.0078pF$$

Finally, all the capacitances connected to the same node are added together yielding the circuit of Fig. 3.6(d), where:

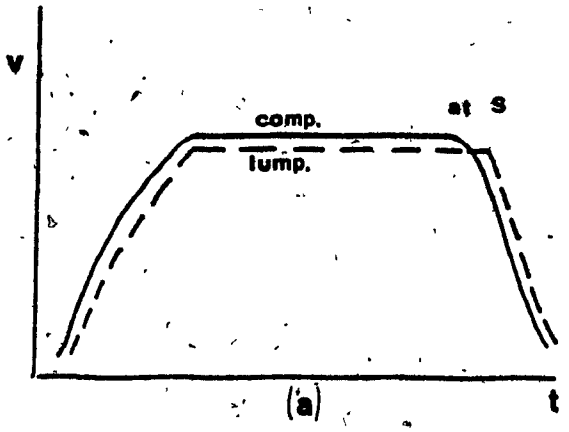
$$C_1 = C_{g1} + C_{g2} = 0.03pF$$

$$C_2 = C_{d1} + C_{d2} + C_{g3} + C_{g4} = 0.045pF$$

$$C_3 = C_{d3} + C_{d4} = 0.015pF$$

The accuracy of the switch-level capacitive model can be assessed by comparing SPICE simulation results on the delays computed in the circuits of Figs 3.6(a) and 3.6(d). In the former case the circuit is simulated using its complete MOS2 model.

In the latter, the transistor capacitances of the SPICE MOS2 model are zeroed and external capacitors with the values obtained above are used. Comparative results for the exact and approximate models are shown in Fig 3.7.



	t_{hl} (ns)	t_{lh} (ns)
Complete	1.74	1.74
Lumped	1.82	1.82

Figure 3.7 (a) Obtained SPICE waveforms on the cascaded CMOS inverters using the complete and the lumped capacitance models. (b) Delays obtained at node S running SPICE on both models.

3.4 The model for the effective resistance

Since the capacitances of the RC model are evaluated in the preprocessing step and remain constant during simulation, the effective resistance must include all the effects that cause variations in the delay and must, therefore, be computed during simulation. The effective resistance is defined as *the equivalent linear resistor that causes the same delay as the transistor, under the same operating conditions.* Consider the circuit of Fig. 3.8. The voltage across the capacitor C_L rises from 0 (initial value) to V_{dd} and at time t_d it crosses a pre-determined threshold v_{th}.

Based on the Lin and Mead expression of delay, given by Eq. (2.13), the value of R_{eff} can be defined simply as:

$$R_{eff} = \frac{t_d}{C_L} \tag{3.9}$$

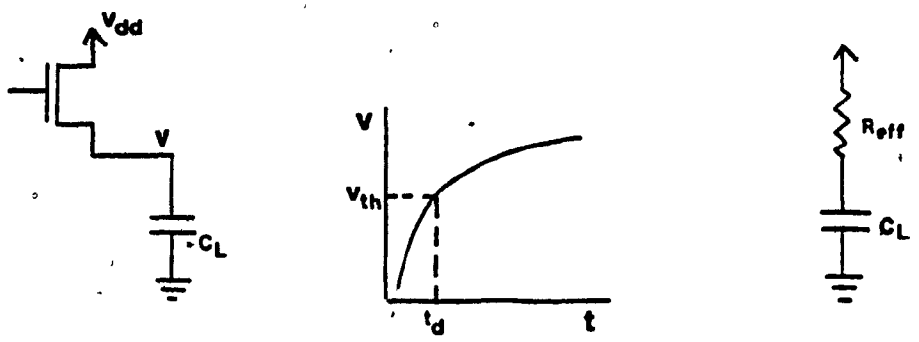


Figure 3.8 Definition of effective resistance

where t_d is the delay as defined in Section 3.2. The effective resistance of a transistor is an empirical model, based on experimental results, that serves the *unique purpose* of delay calculation. A comprehensive model, based on the concept of the effective resistance, can provide a reasonably precise and efficient method for delay calculations in MOS circuits.

If only one single value of effective resistance would suffice to model the transistor, the value obtained in Eq. (3.9) could be used to replace each ON transistor when building the RC network needed for the delay calculations. However, the problem is more complex, as the effective resistance of a transistor depends on both static and dynamic factors of the circuit to be simulated. The effective resistance must be determined separately for each transistor during simulation according to:

Size: The length and width of the active transistor area have a direct effect on the value of the effective resistance.

Type: There are three basic types of transistors used in MOS circuits, for which the effective resistance yields different characteristics. These are n-channel enhancement, p-channel enhancement and n-channel depletion. In addition, transistors with different threshold voltages correspond to different types.

Context: In the static sense, it refers to the context in which a transistor appears in the circuit, i.e., pulldown, pass transistor, etc. In the dynamic sense,

context refers to the actual activity of the transistor at a given time, such as turning ON or OFF, transmitting a high or low voltage, etc.

Load: The nonlinear characteristics of the transistor are such that the effective resistance depends on the load driven by the transistor. This load, in turn, depends on the connectivity of the circuit at a given time.

Gate waveform: When the transistor is changing state due to a signal applied to its gate, the delay depends on how fast the transistor switches. In other words, the rate of change of the input waveform has a direct effect on the value of the effective resistance.

In the following sections, a comprehensive model which takes all these factors into consideration, is developed.

3.4.1 Context dependence

Consider the circuits shown in Fig. 3.9. In case (a), the transistor is already fully ON and a transition occurring at its drain is propagated to the source.

The delay is the time from the transition at the drain crossing a threshold, to the transition in the source crossing the same threshold. In other words as V_s changes, V_d follows but delayed (Fig. 3.9(c)). V_{ds} remains approximately constant and small compared to V_{dd}. The effective resistance, R_{eff}, in this case is relatively low because it is determined by the slope of the output characteristic of the transistor in the triode region, where dV_{ds}/dI_d has a small value, as shown in Fig. 3.9(e).

In case (b), the transition occurring at the gate of the transistor is turning it ON and the delay is computed as the time from the transition at the gate crossing a threshold to the drain reaching the same threshold (Fig. 3.9(d)). Since $V_{ds} = V_{dd}$ initially, it can be seen from Fig. 3.9(f) that the transistor spends a long time in

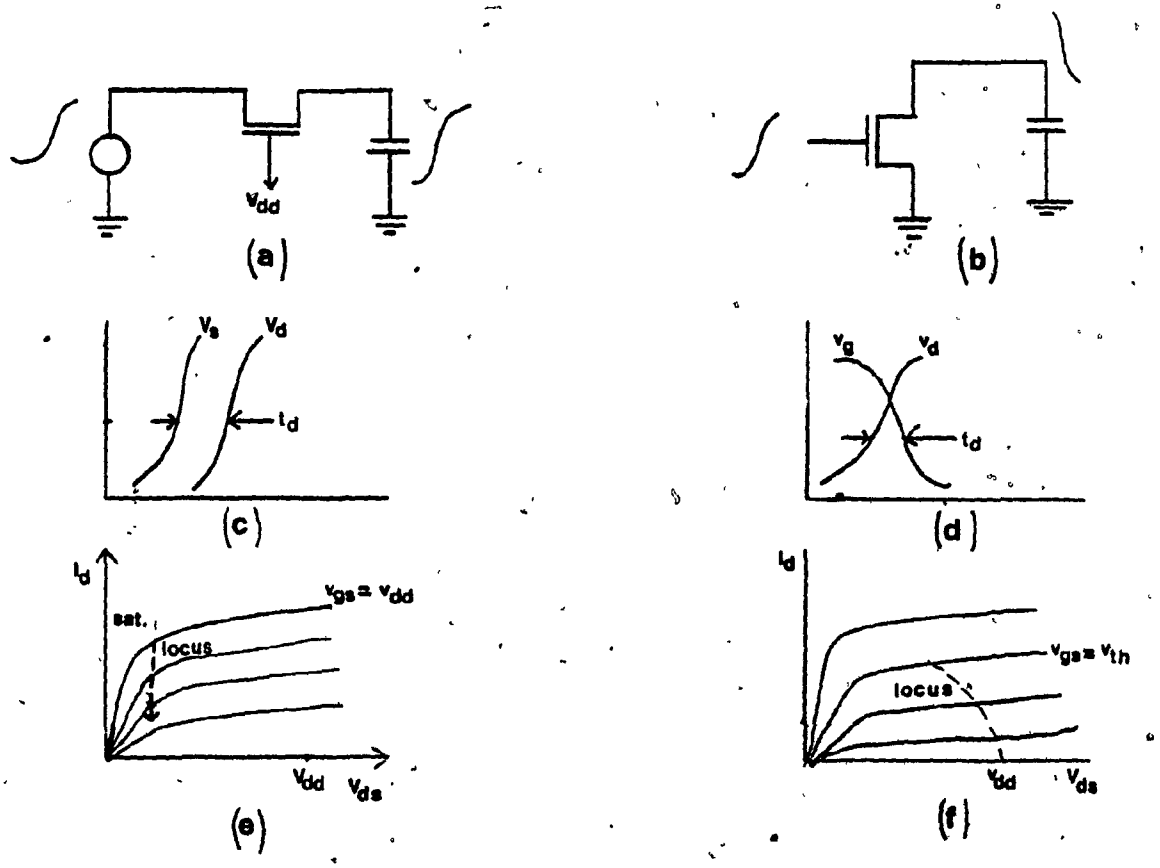


Figure 3.9 (a) Transistor passing. (b) Transistor switching. (c) Delay in trans. passing. (d) Delay in trans. switching. (e) Locus in case of trans. passing. (f) Locus in case of trans. switching.

saturation where dI_d/dV_{ds} is small, therefore the effective resistance will be higher than in the case of Fig. 3.9(a). Hence, it is evident that a distinction should be made in each of the two cases presented above when evaluating the effective resistance of the transistor. In case (b), the transistor is switching from OFF to ON due to a signal applied at its gate, whereas in case (a) it is already ON and is simply transmitting a signal between its source and drain. These two cases are referred to, respectively, as *transistor switching* and *transistor passing* and are recognized as such by the new model when evaluating the effective resistance.

Next, the circuits used in the previous example are expanded to the ones shown in Fig. 3.10.

For the case of the transistor switching two cases are considered:

a) The value transmitted through the transistor is a 1.

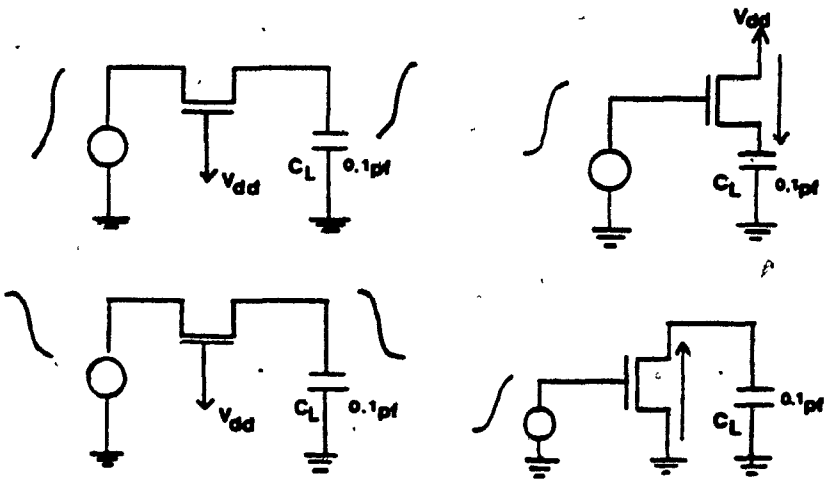


Figure 3.10 Transistor switching and passing transmitting different logic values.

b) The value transmitted through the transistor is a 0.

As for the transistor passing, two cases are also considered:

c) A high-to-low transition is propagating through the transistor.

d) A low-to-high transition is propagating through the transistor.

SPICE simulation results obtained for the 4 cases appear in Table 3.3. They suggest that the effective resistance of a transistor depends also on the logic value being transmitted. In general, for n-channel transistors, the effective resistance is higher when propagating a logic 1 than a logic 0. For p-channel devices R_{eff} is smaller when transmitting a 1 than a 0.

Case	Log. Value Prop.	Delay (ns)	$R_{eff}(k\Omega)$
Switching	0	2.2	22
Switching	1	3.8	38
Passing	0	1.7	17
Passing	1	2.5	25

Table 3.3 SPICE results on propagation of different logic values

The complete context dependence for the effective resistance adopted in the new model is summarized in Table 3.4. It recognizes four different cases, all based on the dynamic activity of the transistor at a given simulation time. These are the only contexts recognized by the present model. Static contexts are not recognized; for example, the model does not differentiate between an n-channel enhancement device used within a logic gate or as a pass transistor. Despite the simplification, surprisingly good results are obtained by using this simple model.

Context
Switching to transmit a 1
Switching to transmit a 0
Passing a 1
Passing a 0

Table 3.4 Transistors according to their context

3.4.2 Load and size dependence

Passing transistors

Fig. 3.11(a) shows an n-channel transistor connected to a load capacitor C_L. SPICE simulations were performed on the circuit, varying the value of the capacitor. The obtained results appear in Fig 3.11(b), and suggest that the effective resistance of passing transistors is essentially independent of the load, except for very small loads (for reference, the gate capacitance of a minsize CMOS inverter is $0.03pF$).

Depletion transistors used as pull-ups in NMOS circuits are a particular example of passing transistors, and it is shown elsewhere in this section that their effective resistance is indeed independent of the capacitive load over a very wide range.

Based on the experimental results obtained above, the model assumes the effective resistance of passing transistors as being independent of the load. For each

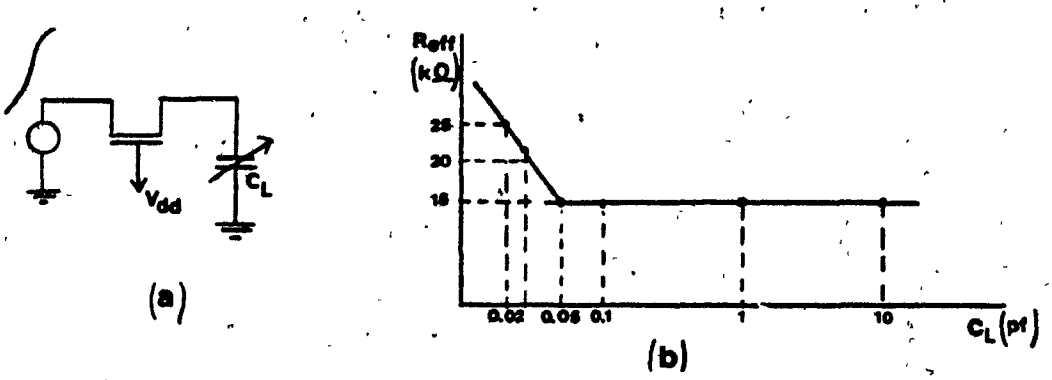


Figure 3.11 Effective resistance of a passing transistor as a function of the capacitive load. (a) Circuit used for the experiment (b) Obtained results plotted on a logarithmic scale.

transistor size two constant values are employed to evaluate the effective resistance of passing transistors. The value, selected during the simulation cycle depends on whether the transistor is propagating a logic 1 or a logic 0, as explained in Section 3.4.1.

Switching transistors

Fig. 3.12 shows a minsize static CMOS inverter, driven by a step function, and the equivalent RC circuits for the two possible output transitions. The step function is chosen in order to avoid any influence on the results by the slope of the input signal (see Sec. 3.4.3).

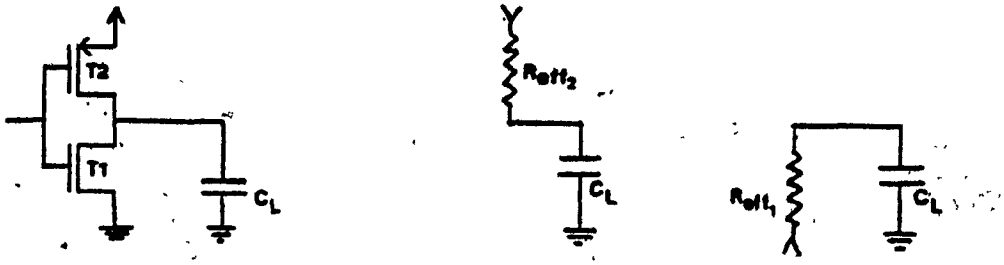


Figure 3.12 Equivalent RC circuits for the two possible transitions at the output of a CMOS inverter.

The inverter is assumed to be symmetric with the p-channel device approximately twice as wide as the n-channel. Circuit simulation was used to extract the values for the effective resistance for the switching n- and p-channel devices transmitting 0 and 1 respectively. For the n-channel device the effective resistance is given by:

$$R_{eff_1} = \frac{t_{hl}}{C_L} \tag{3.10}$$

Similarly for the p-channel device transmitting a logic 1, the effective resistance is given by:

$$R_{eff_2} = \frac{t_{lh}}{C_L} \tag{3.11}$$

It is evident that both measured values include the effect of the opposite transistor switching OFF. If the transfer characteristic is symmetric, $t_{hl} \approx t_{lh}$, and $R_{eff_1} \approx R_{eff_2}$, then a single value would suffice to represent both the n- and p-channel devices. This means that a p-channel device with a *length/width* ratio $L/W = 1/2$ has the same effective resistance transmitting a 1, as a 1/1 n-channel device transmitting a 0. This fact has practical importance in the implementation of the models, as p- and n- channel transistors can be assumed to be of the same type when transmitting opposite logic values, if a proper scaling factor is used to compensate for the size.

By changing the capacitive load, the variation of R_{eff} with C_L shown Fig. 3.13, was obtained. This result shows that the effective resistance is a function of C_L, although for sufficiently large C_L, it becomes constant. It means that, for small loads, the delay of the stage is not a linear function of the load, as compared to large loads where the delay increases linearly with the load. It can be seen from the

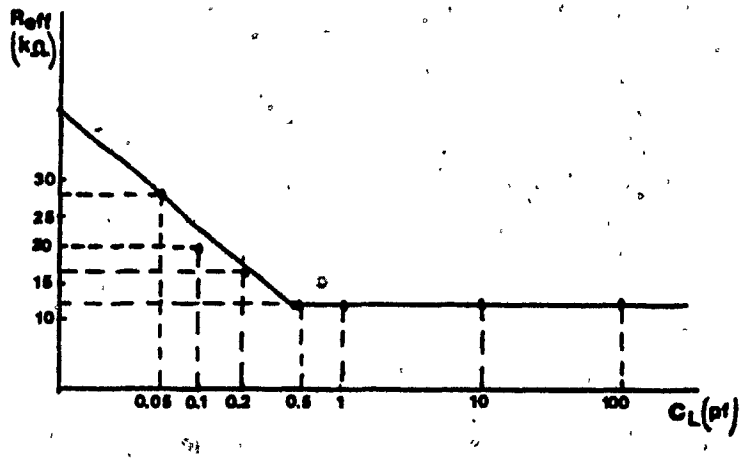
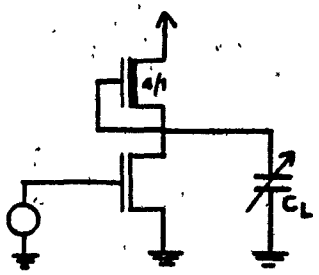


Figure 3.13 Effective resistance of a switching transistor as a function of the load.

numerical results, that the capacitance at which R_{eff} becomes constant is much larger than the one at which the effective resistance of the same transistor when passing, becomes constant (Fig.3.11). The new delay model assumes the effective resistance of a switching transistor as being a function of the load.

As an additional example, Fig. 3.14(b) shows the results for the NMOS standard inverter of Fig. 3.14(a), simulated with different capacitive loads.



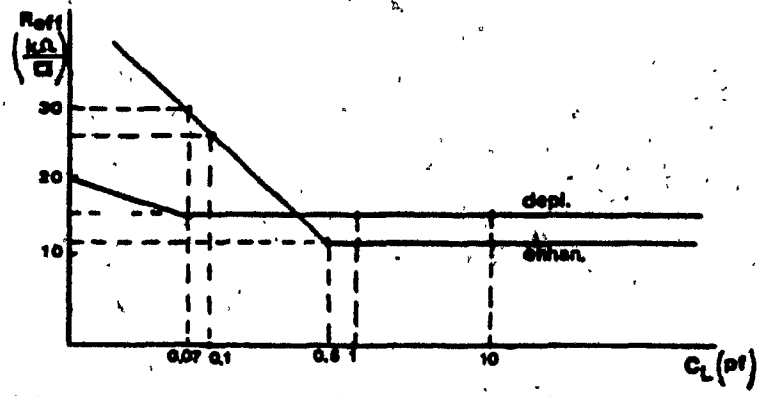


Figure 3.14 (a) NMOS inverter (b) Effective resistance of the pull up and pulldown transistors as a function of the capacitive load.

It can be seen that while the effective resistance of the pull-down enhancement transistor varies substantially with the load (up to a value of 0.5pf) the effective

resistance of the depletion load remains approximately constant (above 0.05pF). This result gives further strength to the observation that the effective resistance of passing transistors such as the depletion pull-up, can be assumed to be independent of the load, whereas the effective resistance of switching transistors (the pull down) is clearly a function of the capacitive load.

To see the relation between transistor size and load, different transistor sizes were simulated with various capacitive loads for each of the two possible switching contexts (see Sec. 3.4.1). Fig. 3.15(a) shows R_{eff} as a function of C_L for 4 n-channel transistor sizes transmitting a logic 0. The effective resistances are given in $k\Omega/sqr$. Although different transistor sizes have different effective resistances, the relative variation of R_{eff} with C_L is the same for every transistor size. Furthermore, for an infinite load all the transistors have the same effective resistance (on a per/sqr basis). This resistance, R_{nom}, corresponds to the case when the delay of the transistor becomes linear with the load.

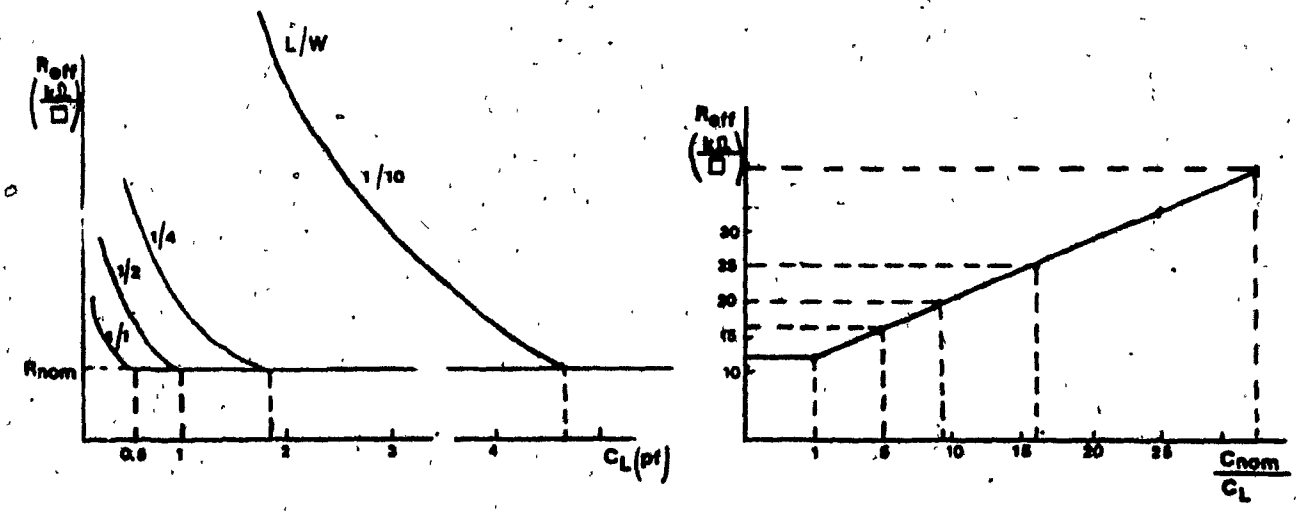


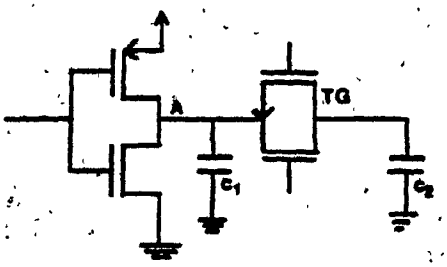
Figure 3.15 (a) Effective resistance as a function of the size and load.
(b) R_{eff} as a function of C_{nom}/C_L.

The value of C_L at which the effective resistance becomes constant, denoted by C_{nom}, is a characteristic of each transistor size, and can be determined by running simulations on simple stages. The quantity C_{nom}/C_L is a measure of the relative load of a stage, and *the variation of R_{eff} with this quantity is independent of the size of the transistor*. This relation is shown in Fig. 3.15(b). Therefore, a table containing R_{eff} as a function of C_{nom}/C_L can be used to extract the value of the effective resistance (on a per/sqr basis) for any transistor size, as a function of the load. The value extracted from the table must be multiplied by the number of squares of the corresponding transistor to obtain the absolute value of the effective resistance. For each transistor type two such tables are needed depending on whether the transistor is transmitting a logic 1 or a logic 0.

The model, therefore, reads the value of C_{nom} for different transistor sizes, and has one table containing the value of R_{eff} for different C_{nom}/C_L's. The value of the effective resistance is extracted from the table, during simulation. The quantity C_{nom}/C_L is computed every time the effective resistance of a switching transistor is needed, due to the dependence of the transistor's effective load C_L on the connectivity of the circuit. In other words, the delay of a switching device is a function of its dynamic load, as illustrated in the example of Fig. 3.16(a). The delay at the output of the inverter depends on whether the transmission gate is ON or OFF, as the SPICE results of Fig. 3.16(b) suggest. For the first case the load at node A is C_1, whereas, when the transmission gate is ON, the load at A is approximately $C_1 + C_2$.

3.4.3 Input slope dependence

An important factor which affects delays in MOS circuits is the gate waveform shape. If a transistor turns ON instantaneously, then its full driving power is applied



case	t_d (ns) at A
TG (OFF)	2.3
TG (ON)	4.1

Figure 3.16 Dynamic loading (a) The load at node A is varied by turning ON or OFF the transmission gate. (b) SPICE delays obtained for both cases.

to the output capacitance, and the transistor has a low effective resistance. If, on the other hand, the transistor turns slowly, then it may do much of the work while only partially turned-ON. In this case the effective resistance will be higher. This is illustrated in Fig. 3.17.(a). The delay of the transistor can be roughly expressed as

$$t_d = \frac{C\Delta V}{I_{d_{av}}} \qquad (3.13.a)$$

and

$$R_{eff} = \frac{t_d}{C_L} = \frac{\Delta V}{I_{d_{av}}} \qquad (3.13.b)$$

where $I_{d_{av}}$ is the average channel current flowing in the transistor, ΔV, is the variation in the voltage across the capacitance, C_L. For fast signals, V_{ds} is not able to follow V_{gs} and full current, I_d initially drives the capacitance as shown in Fig. 3.17(b). In the case of slow signals, V_{ds} follows V_{gs} and hence, the current I_d is lower (Fig. 3.17(b). The result is that, the effective resistance of a transistor driven by a slow signal is bigger than that of a transistor driven by a fast signal.

In MOS circuits, it is assumed that all the transitions are monotonic in nature, and that they differ from each other only in their slope. Therefore the effect of the gate voltage rise time on the delay is referred to as the effect of the slope.

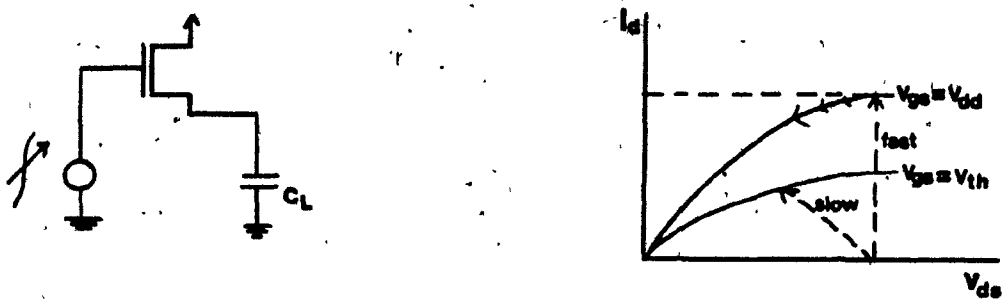


Figure 3.17 (a) The slope of the transistor's gate signal is varied. (b) I_d in the output characteristic for fast and slow signals

Quantification of the input slope

In most MOS circuits, the output rise/fall time of a stage can be assumed to be proportional to the delay of the stage, due to the fact that most of the delay is transitional and not inertial. Fig. 3.18(a) shows a static CMOS inverter. The output rise time is changed by varying the capacitive load connected at the output of the inverter. Fig. 3.18(b) shows the output delay (ns) plotted against the output rise time (Defined as the time it takes the signal to rise from 10% to 90% of its final value).

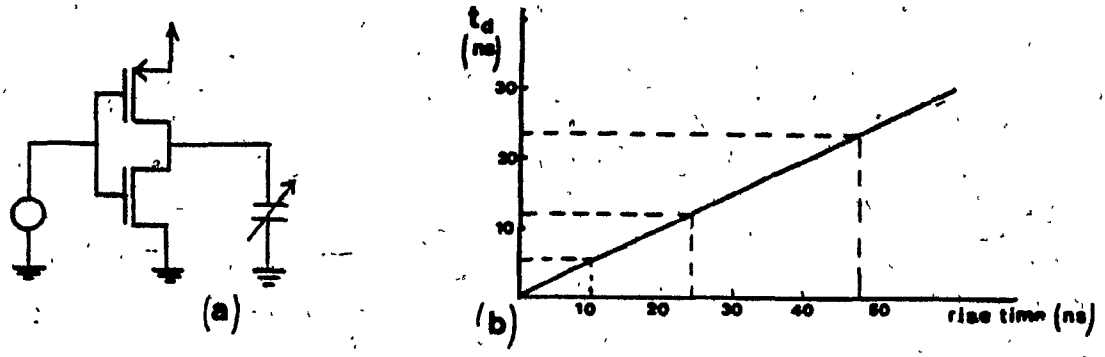


Figure 3.18 The output rise-time as a function of the delay

The linearity in the graph provides the foundation for the previous statement that the slope of a signal is proportional to the delay of the stage that generates

it. Therefore, the slope of the input waveform can be equivalently quantified in terms of the delay of the driving stage, denoted as *previous delay*, t_{pd}. Fig 3.19 shows the effect of the increase in the previous delay on the effective resistance of a switching transistor, with respect to the effective resistance obtained for a step function, $R_{eff_{step}}$ where $t_{pd} = 0$.

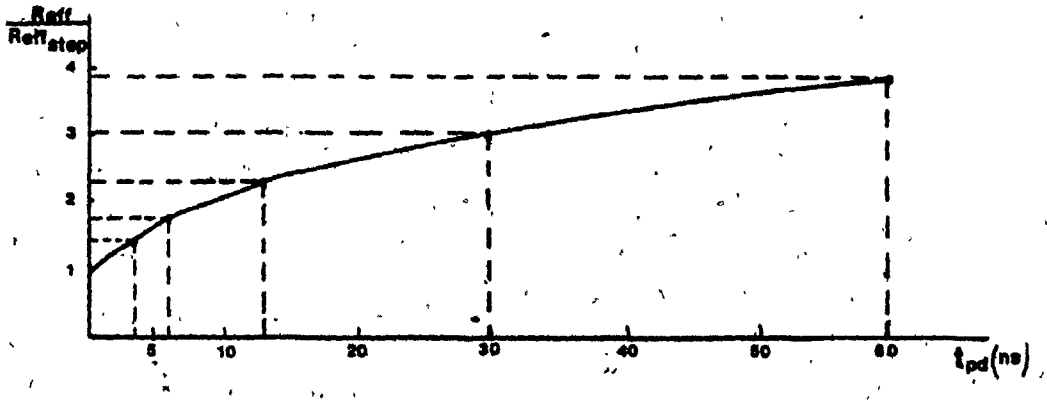


Figure 3.19 The effective resistance as a function of the previous delay, t_{pd}.

The slope ratio

The effect of the slope of the input waveform on the effective resistance itself depends on both the size of the transistor and its load. In general, stages driving large loads or those with narrow transistors are less affected by the slope than stages with small loads or wide transistors. The method used to compensate for the slope effect is based on Ousterhout's discovery that all these factors can be combined into a single ratio [Ous84]. In the present context, an *intrinsic delay*, t_{di}, is defined as the delay of the transistor driven by a step function, and is obtained by multiplying the load driven by the transistor by the effective resistance corresponding to that load. This effective resistance will be denoted by $R_{eff_{step}}$ and is obtained from a table as described in the previous section. Next, the *slope ratio*, $SR = t_{di}/t_{pd}$, is

defined as the intrinsic delay divided by the previous delay. The slope ratio is a measure of how much a stage is affected by a given input slope, and as experimental results show, it is *approximately independent of the load and size of the stage.* A big slope ratio means that the input slope has little effect on the delay of the stage, while a stage with a small slope ratio is strongly affected by the input slope.

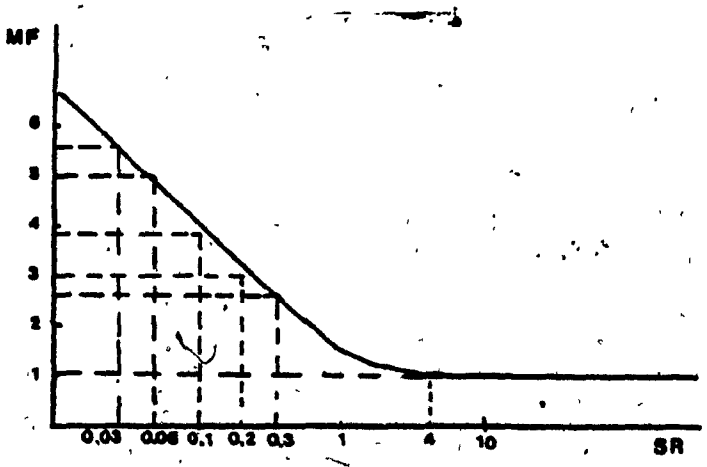


Figure 3.20 Multiplying factor, MF as a function of the slope ratio, SR.

Fig 3.20 shows the influence of the slope ratio on the factor MF by which the effective resistance, $R_{eff_{step}}$ must be multiplied to be adjusted for the input slope. The multiplying factor is defined as:

$$MF = \frac{R_{eff}^{t_{pd}}}{R_{eff_{step}}}$$

where $R_{eff_{step}}$ is the effective resistance of the transistor driven by a step function and $R_{eff}^{t_{pd}}$ is the effective resistance of the transistor when driven by a signal whose delay is t_{pd}.

The model computes the slope ratio for switching transistors and subsequently extracts the value of a multiplying factor by which the effective resistance, previously found in the load table, is multiplied.

3.5 The simulator and the model

During the preprocessing step, McSLADE computes the values of the node capacitances, using technology data supplied by the user, and layout information which can be provided by a circuit extractor. Also during the preprocessing, the simulator partitions the circuit into transistor groups [Bry80], as described in Section 2.5. A transistor group is evaluated whenever an event has occurred on one of its inputs. This evaluation includes three steps; logic evaluation, delay calculation, and scheduling. After the logic evaluation is performed on the switch network of the transistor group, the logic values propagating through the different transistors of the group, and the direction of the propagation are known. Next, the simulator builds the RC model of the network, on the basis of which the delays are to be calculated using the Lin and Mead approach (Sec. 2.5.2). The transistor or transistors that have been turned ON by the present event are treated as switching transistors, for the effective resistance extraction. The other transistors inside the group that were already ON are treated as passing transistors. As the logic values propagating through the transistors are also known, each transistor can be completely categorized according to its context. For the passing transistors, the effective resistance is obtained directly according to the size of the transistor and the logic value propagating through it.

The effective resistance of the switching transistors, on the other hand, is evaluated in three steps as follows:

1. The value of C_{nom} corresponding to the size of the switching transistor is read.
2. The effective resistance is extracted from a table according to the load driven by the transistor (the R_{eff} obtained corresponds to the transistor driven by a step function).

3 The effective resistance is adjusted for the slope of the gate signal, by extracting a value from the second table according to the slope ratio, and multiplying it by the value obtained in step 2.

The complete model consists of 4 basic values for each transistor type, and two sets of two small tables each. The basic values are two effective resistances for the transistor passing a 1 and passing a 0, respectively, and two values of C_{nom} for the switching transistor transmitting a logic 1 and a logic 0. Each set of tables corresponds to the two different logic value (0 or 1) that may propagate through the transistor. The two tables within each set are one table of the effective resistance as a function of the load, and another of the multiplying factor as a function of the slope ratio.

For CMOS circuits, the same set of tables can be used for the n- and p-channel devices. One set corresponds to n-channel devices transmitting a 1 and p-channel devices transmitting a 0. The second set is used for the opposite case, namely n- and p-channel transistors transmitting 1 and 0, respectively. The use of only two sets of tables (instead of four) is possible due to the fact that a p-channel device with a length/width relation of 1/2 has the same effective resistance as a 1/1 n-channel device when transmitting opposite logic values. Therefore, if an appropriate scaling factor is used for the size, n- and p-channel devices transmitting opposite logic values can be assumed to be of the same type. For NMOS circuits, the enhancement devices are treated in the same manner as the n-channel devices in CMOS. Depletion pull-up transistors, on the other hand, are always considered as passing transistors for output low-to-high transitions. For high-to-low transitions they are considered OFF.

Once the RC network has been built, the delays are calculated, and those nodes that have changed state are put in the event scheduler. The time at which

the change in the node state will occur is computed by adding the calculated delay at the node to the present simulation time.

3.6 Extracting and updating the model

This section illustrates a systematic way of obtaining the values used by the model developed in the previous sections. This model is analytically independent of the integrated circuit technology process. However, every time technology changes occur, the numerical values used by the model must be reevaluated.

3.6.1 Capacitance extraction

The switch-level capacitance model is evaluated on the basis of technology parameters similar to those used by SPICE, as shown in Section 3.3. These parameters, summarized in Table 3.1, are known for any IC process, therefore the user has only to update them in the technology file read by the simulator during the preprocessing step (see Sec. 3.3).

3.6.2 Effective resistance extraction

The model used for the evaluation of the effective resistance is more complex and involves performing circuit simulation on some basic configurations.

Prior to the model extraction, proper threshold voltages for both CMOS and NMOS circuits must be chosen. As mentioned in section 3.2, the threshold will be that of the standard minsize inverter. Fig. 3.21 shows the minsize inverter and the obtained transfer characteristic, for both NMOS and CMOS cases. From the transfer characteristics, the numerical values obtained for the threshold voltages are for NMOS circuits, $V_{th} = 2.1$ volts and for CMOS circuits, $V_{th} = 2.5$ volts. These values were used in the extraction and assessment of the model.

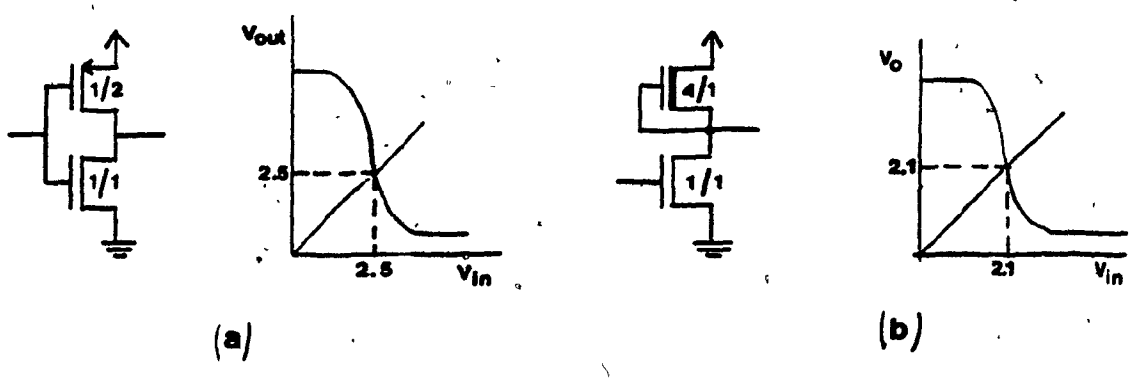


Figure 3.21 Threshold extraction for (a) NMOS (b) CMOS.

Passing transistors

Two values must be obtained each for the transistor transmitting a logic 1 and a logic 0 (low-to-high and high-to-low transitions, respectively) for every transistor size. For that purpose, two different circuits are proposed below for n-channel transistors as shown in Fig. 3.22 for CMOS circuits.

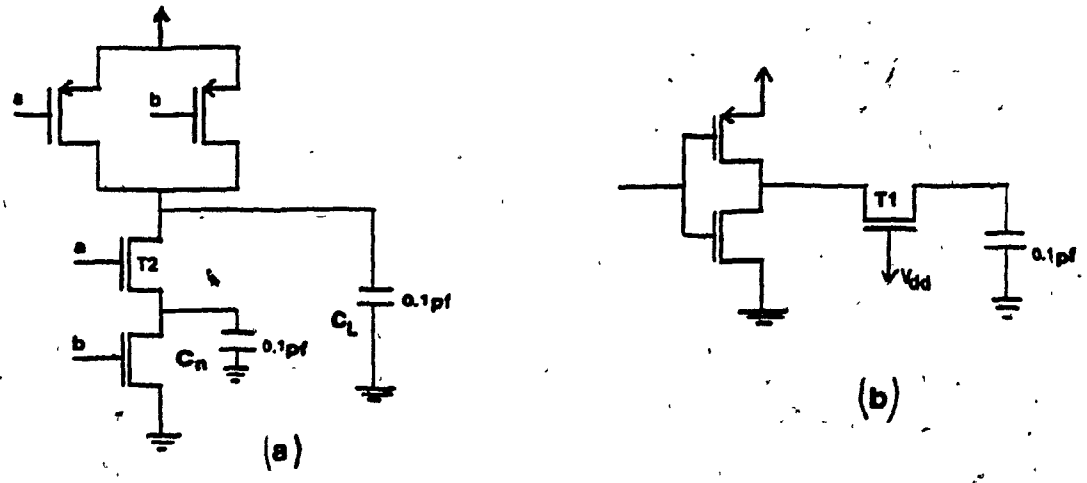


Figure 3.22 Extraction of the effective resistance of passing transistors for CMOS.

In circuit (a) for $a = 1$ and $b : 0 \rightarrow 1$ (b switches from 0 to 1) the high-to-low transition value (propagation of a logic 0) can be obtained for passing transistor $T2$, by taking the time from the voltage at C_n crossing a threshold to the voltage at

C_L crossing the same threshold. For $a = 1$ and $b : 1 \rightarrow 0$ the low-to-high transition value may be obtained. In circuit (b) the method is to induce the transition at the output of the inverter and measure the propagation delay through the ON transistor $T1$. The values of R_{eff} obtained in circuits (a) and (b) for the transmission of both a logic 1 and a logic 0, appear in Table 3.5.

Logic Value	Fig.3.22	$t_d(ns)$	$R_{eff}(k\Omega)$
0	(a)	1.5	15
0	(b)	1.7	17
1	(a)	2.5	25
1	(b)	2.8	28

Table 3.5 Values of R_{eff} for the n-channel pass trafisistor.

Altough the values for R_{eff} obtained in both circuits of Fig. 3.22 for a given logic value are not equal, they are close enough to assume that a single value, for each of the two possible propagations (logic 1 or logic 0) suffices to represent the effective resistance. In other words, either circuit of the two shown in Fig. 3.22 can be used to extract the two effective resistances needed for passing transistors. A similar procedure must be followed for p-channel devices, in which case a 2-input NOR gate can be used. As for NMOS circuits, the effective resistance of depletion pull-ups can be obtained from a simple inverter simulating the output low-to-high transition with a medium size load. The effective resistance of the enhancement devices can be obtained using the simple configurations shown in Fig. 3.23, following the same procedure as for the CMOS case.

The number of transistor sizes whose passing effective resistances are evaluated depends on the particular circuit to be simulated. However, If this circuit includes transistor sizes whose passing resistance is not available, it can be approximated by

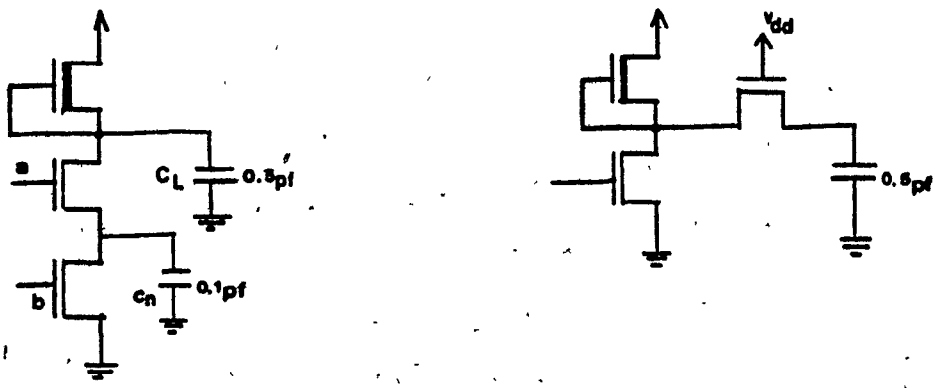


Figure 3.23 Extraction of the effective resistance of passing NMOS transistors

applying linear interpolation/extrapolation to existing values. Table 3.6(a) shows the passing effective resistances obtained for 4 different sizes of n-channel transistors.

Switching transistors

For each switching context (see Table 3.4) two tables must be built: one table to extract the value of the effective resistance as a function of C_{nom}/C_L, and the second table to adjust the R_{eff} with the slope of the driving signal.

The procedure used to obtain the entries for the tables for a switching n-channel transistor transmitting a logic 0 is described below.

First step: Evaluation of C_{nom} for transistors of different sizes. It is obtained by performing simulations on a simple inverter varying the capacitive load. The load at which R_{eff} becomes constant is C_{nom}. The experiment is repeated for various transistor sizes using the circuits shown in Fig. 3.24(a) for NMOS and CMOS. However, in cases where the value of C_{nom} is not available for a particular transistor size, the model approximates it by means of linear interpolation or extrapolation using existing values from other transistor sizes. This is illustrated in Fig. 3.24(b), where C_{nom} is plotted as a function of the transistor size for the CMOS n-channel

devices. For example, if linear extrapolation were used for the transistor with $L/W = 1/10$, the resulting value of C_{nom} would be $5pF$ instead of $4.7pF$. obtained from simulation.

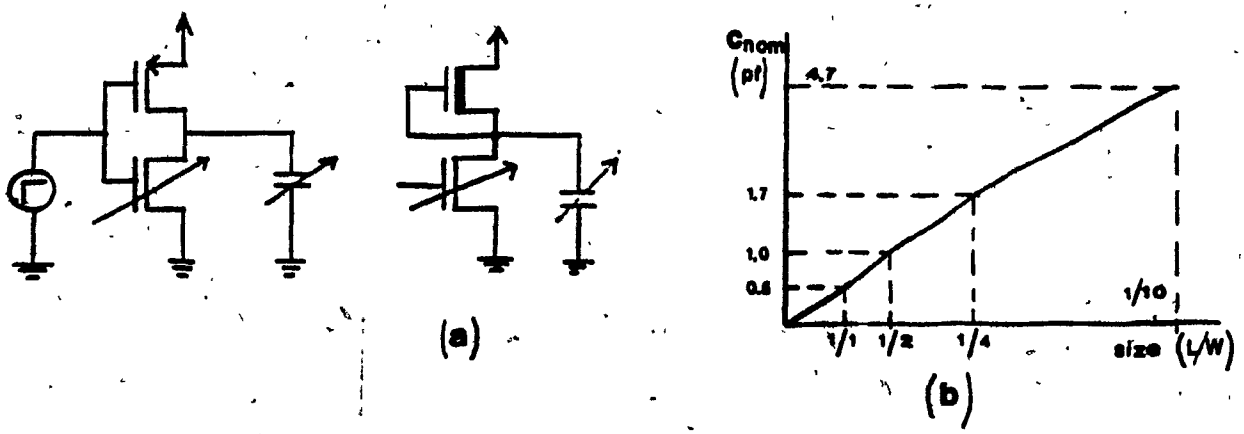


Figure 3.24 Extraction of the values of C_{nom} for NMOS and CMOS for different transistor sizes.

Second step: Evaluation of R_{eff} as a function of C_{nom}/C_L. By taking any transistor size, the entries of the table can be obtained by performing several simulations on the inverter, driving it with a step function, and varying the load C_L as shown in Fig 3.24. The effective resistance for each C_{nom}/C_L must be expressed in $k\Omega/sqr$. If the number of entries in the table is between 15 and 20, interpolation may be avoided during simulation. If the particular value of C_{nom}/C_L does not appear in the table, the program chooses the R_{eff} corresponding to the closest larger C_{nom}/C_L existing in the table.

Third step: Evaluation of the multiplying factor as a function of the slope ratio. The configuration used to extract the multiplying factor as a function of the slope ratio for the n-channel transistor $T3$ appears in Fig. 3.25.

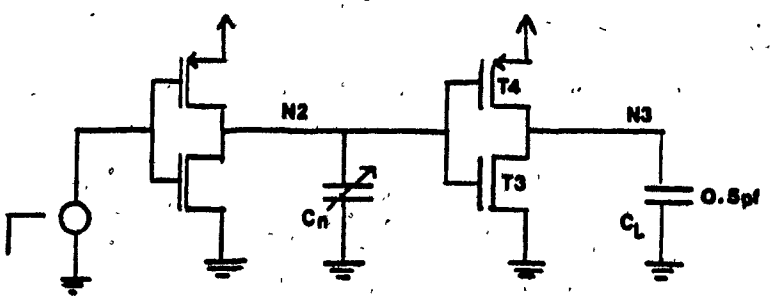


Figure 3.25 Extraction of the multiplying factor as a function of the *sloperatio*.

From the previous step, the effective resistance of transistor $T3$ when driven by a step function, $R_{eff_{step}}$, is known. The slope at the input of the righthand inverter is changed, by varying C_n. The delay of the first inverter, measured at $N2$, is the previous delay, t_{pd}, for the second inverter, and is used to evaluate the slope ratio, $SR = R_{eff_{step}} C_L / t_{pd}$. The different delays obtained at the output of the rightmost inverter as the slope changes, are used to compute the effective resistance of transistor $T3$. By dividing the obtained value of R_{eff} by $R_{eff_{step}}$, the multiplying factor, MF, is obtained for a given slope ratio. The value of C_L may be chosen arbitrarily. The number of entries in the table depends on the type of circuit to be simulated, although for most MOS circuits, between 10 and 15 entries will suffice to avoid interpolation.

The two tables obtained from the procedure above are also used to evaluate the effective resistance of a p-channel transistor transmitting a logic 0, as explained in Section 3.5.

For the switching transistors transmitting *bad* values, namely n-channel devices propagating a 1 or p-channel devices propagating a 0, a similar procedure must be followed to compute the values for the tables, with the difference that the inverter can not be used for that purpose. Instead, the 2-input NAND gate shown in Fig. 3.26 can be used. If $b = 0$ and $a : 0 \rightarrow 1$, the n-channel transistor $T1$ is transmitting

a logic 1. By varying the capacitance C_{n1} first and the slope of the signal driving $T1$ next, the two tables can be obtained. These two tables will be used to compute the effective resistance of the n- and p-channel transistors transmitting a logic 1 and 0, respectively.

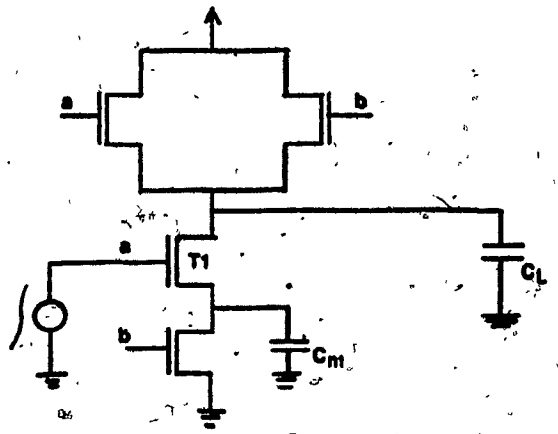


Figure 3.26 Circuit used for the extraction of the model for a switching n-channel transistor transmitting a 1.

The complete model for the Northern Telecom CMOS 1B process is shown in Tables 3.6. They include the adjustment tables for the load and the slope, and values for passing resistances and C_{nom}'s for four transistor sizes. In the tables R_{on_0} corresponds to the effective resistance of a passing transistor transmitting a 0, and R_{on_1} denotes the effective resistance of a passing transistor transmitting a 1. C_{nom_0} and C_{nom_1} denote the C_{nom} of the switching transistor transmitting respectively, a logic 0 and a logic 1. Tables 4.6(b) and 4.6(c) correspond to the case when n and p-channel transistors propagate logic 0 and 1 respectively, whereas 4.6(d) and 4.6(e) correspond to the opposite case. A complete example of how the tables are used to calculate the effective resistances is shown in Section 4.1

Size (L/W)	$R_{on_0}(k\Omega)$	$R_{on_1}(k\Omega)$	$C_{nom_0}(pF)$	$C_{nom_1}(pF)$
1/1	15	25	0.5	0.4
1/2	9	14	1	0.7
1/4	5	9	1.8	1.5
1/10	2	4	4.7	3.7

Table. 3.6(a) Basic Parameters for 4 different n-channel transistors.

C_{nom}/C_L	$R_{eff}(k\Omega/sqr)$
1	12
1.5	13
2.5	14
3.5	15
5	16
5.6	17
6.2	18
9.4	20
12.5	22
16	25
20	29
25	34
30	40
35	44

Table 3.6 (b) R_{eff} as a function of C_{nom}/C_L. for n- and p-channel devices transmitting 0 and 1 respectively.

$SR = R_{eff_{step}} C_L / t_{pd}$	$MF = R_{eff} / R_{eff_{step}})$
4	1
3	1.2
1.8	1.4
1	1.6
0.73	1.9
0.46	2.2
0.35	2.5
0.23	2.8
0.2	3.0
0.16	3.3
0.12	3.6
0.1	3.8
0.07	4.4
0.05	5
0.03	5.5
0.02	6.5

Table 3.6(c) Multiplying factor, MF, as a function of Slope Ratio, SR, for n- and p-channel devices transmitting 0 and 1 respectively.

C_{nom}/C_L	$R_{eff}(k\Omega/sqr)$
1	15
3.5	16
6	18
10	20
16	24
26	30

Table 3.6 (d) R_{eff} as a function of C_{nom}/C_L. n- and p-type transistors transmitting 1 and 0 respectively.

$SR = R_{eff_{step}} C_L / t_{pd}$	$MF = R_{eff} / R_{eff_{step}}$
3	1
2.3	1.3
1.5	1.6
0.75	3.5
0.25	5.7
0.15	8

Table 3.6(e) Multiplying factor, MF, as a function of Slope Ratio, SR, for n- and p-type devices transmitting a 1 and a 0 respectively.

CHAPTER 4

EXPERIMENTS AND RESULTS

The model described in Chapter 3 has been incorporated into a new switch level simulator, McSLADE [Kho85], written in C and running under UNIX 4.2 BSD on a VAX 11/750. This chapter presents a wide variety of MOS circuits used to test the model. Comparisons between SPICE (MOS2 model) and the new model, on the basis of the definition of delay given in Section 3.2, are presented. Where relevant, simulator statistics such as speed and overhead associated with the delay evaluation are supplied. Emphasis is given to CMOS circuits both static and dynamic, because very little has been published in previous work on CMOS, although some NMOS examples are also included. The examples cover different aspects involved in the delay calculation of digital MOS circuits, such as different loads and transistor sizes, special structures, fast and slow signals, etc. The results are used to assess the model in terms of accuracy, speed, and efficiency, and to show its limitations.

The following notation is used throughout the remainder of the chapter:

1: High logic value.

0: Low logic value.

$a: 1 \rightarrow 0$: Node a switches from logic 1 to logic 0.

$a: 0 \rightarrow 1$: Node a switches from logic 0 to logic 1.

t_{lh}: Output low-to-high transition delay time.

t_{hl}: Output high-to-low transition delay time.

t_d: Required delay (t_{hl} or t_{lh}).

PATH $N1 - N2$: Two nodes ($N1$ and $N2$) between which the delay is evaluated.

t_{pd}: Delay of the driving stage (previous delay).

$R^T_{on_0}$: Effective resistance of a passing transistor, T, transmitting a 0.

$R^T_{on_1}$: Effective resistance of a passing transistor, T, transmitting a 1.

$R^T_{sw_0}$: Effective resistance of a switching transistor, T, transmitting a 0.

$R^T_{sw_1}$: Effective resistance of a switching transistor, T, transmitting a 1.

R_{eff}/sqr: Effective resistance per square of area.

Note: All the capacitances which appear in the diagrams correspond to the total capacitance of the node, which consists of the transistor's terminal capacitances and an external capacitor that may have been added to the node.

4.1 2 input CMOS NAND gate

This example illustrates step by step, the extraction of the value of the effective resistance, from the different tables as required by the new model. The expressions used for computing the delay, using the Lin and Mead algorithm are shown as well. Fig. 4.1(a) shows a 2 input CMOS NAND gate. The equivalent RC network used to compute the delay for the case in which $a = 1$ and $b : 0 \rightarrow 1$ appears in Fig 4.1(b).

The delay at the output node of the gate is given by (from Eq. (2.14)):

$$t_{hl} = R^{T1}_{sw_0}(C_n + C_o) + R^{T2}_{on_0} C_o \qquad (4.1)$$

$R^{T2}_{on_0}$ corresponds to the effective resistance of the passing n-channel transistor $T2$ transmitting a 0, and is obtained directly from Table 3.6(a).

$$R_{on_0} = 9k\Omega$$

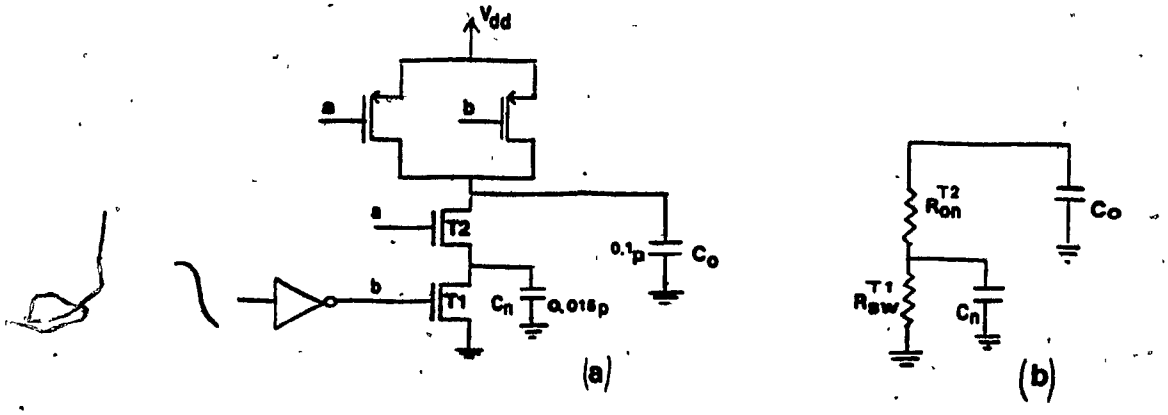


Figure 4.1 (a) CMOS NAND gate. (b) equivalent RC network.

As for $R_{sw_0}^{T1}$, the effective resistance of the switching transistor $T1$ propagating a low logic value, the following steps must be executed to obtain its value.

1. Extract the value of C_{nom} according to the size and type of the transistor. From Table 3.6(a):

$$C_{nom} = 1pF$$

2. Compute C_{nom}/C_L. As can be seen from Fig. 4.1 the total dynamic load connected to transistor T1 is given by:

$$C_L = C_n + C_o = 0.015pF + 0.1pF = 0.115pF$$

and

$$C_{nom}/C_L = 1pF/0.115pF = 8.69$$

3. Obtain the value of the effective resistance per square area as a function of C_{nom}/C_L, from table 3.6(b)

$$R_{eff/sqr} = 19K\Omega/sqr$$

4. This value, $R_{eff/sqr}$, must be multiplied by the number of squares (1/2 in this case) to obtain the value of $R_{eff_{drp}}$, corresponding to the size and load of the

transistor, when driven by a step function.

$$R_{eff_{step}} = (R_{eff}/sqr)\,(\#sqr's) = (19k\Omega/sqr)\,(1/2sqr's) = 9.5k\Omega$$

Note: p-channel transistors transmitting a 1 are treated as n-channel transistors transmitting a 0, but with a double number of squares as their actual dimensions. In the circuit of Fig. 4.1 when evaluating the effective resistance of the 1/2 p-channel devices, the model multiplies R_{eff} by 1 square instead of by 0.5.

5. Compute the slope ratio, SR.

$$SR = R_{eff_{step}} C_L / t_{pd} = \frac{9.5k\Omega\, 0.115pF}{1.4ns} = 0.78$$

Where t_{pd} corresponds to the previous delay, which in the example is the delay of the driving inverter ($t_{pd} = 1.4ns$).

6. The final value of the effective resistance for transistor $T1$ is obtained by multiplying $R_{eff_{step}}$ obtained in step 4 by the multiplying factor, MF, extracted from table 3.6(c), according to the slope ratio.

$$MF = 1.9$$

Hence

$$R_{sw_0}^{T1} = R_{eff_{step}} MF = (9.5)(1.9) = 18k\Omega$$

The delay at the output of the gate given by (4.1) is then equal to:

$$t_{hl} = 18k\Omega\, 0.115pF + 9k\Omega\, 0.1pF = 2.9ns$$

The corresponding SPICE result gives a delay equal to $3.2ns$.

4.2 CMOS NAND-NOR circuit

Fig 4.2 shows a 4-input CMOS NAND gate driving a 3-input NOR gate. All the transistors are minsize (L/W - n : 1/1.p : 1/2). The input of the NAND gate is driven by an inverter. Two different cases are considered. In case (a) the output of the NAND GATE, S_1, is switching from high to low, due to a high-to-low transition at input d, while the other NAND inputs remain at logic high. The change in $S1$, which is connected to one of the inputs of the NOR gate causes, in turn, a transition from 0 to 1, at the output of the NOR gate, S_2. (e and f are held fixed at 0). In case (b), d switches from $1 \rightarrow 0$ causing the output of the NAND gate to go from low to high, and the output of the NOR gate to go from high to low. In both cases the load capacitance C_{L1} is varied to change the slope of the signal driving the input of the NOR gate, so that the effect of the slope can be analyzed. Fig. 4.2(b) shows the equivalent RC networks for case (a). C_n represents the node capacitance of two transistors in series resulting from adding the source capacitance of one transistor to the drain capacitance of the other. The Lin and Mead expressions used to calculate the delays in case (a) are:

At S_1:

$$t_{hl} = R_{sw_0}^{T1}(C_{L1} + 3C_n) + R_{on_1}^{T2}(2C_n + C_{L1}) + R_{on_0}^{T3}(C_n + C_{L1}) + R_{on_0}^{T4}C_{L1}$$

where $R_{sw_0}^{T4}$, the effective resistance of the switching transistor $T4$, is first adjusted with the load $C_{L1} + 3C_n$, and then with the delay of the driving inverter (t_{pd}=1.4ns).

At S_2:

$$t_{lh} = R_{sw_1}^{T7}C_{L2} + (R_{on_1}^{T5} + R_{on_1}^{T6})C_{L2}$$

where $R_{sw_1}^{T7}$, the effective resistance of the switching p-channel transistor $T7$ transmitting a 1, is a function of the load C_{L2} and the delay of the driving NAND gate.

In this case, because the node capacitances between $T5$ and $T6$ and between $T6$ and $T7$ are initially charged to the final value 1, they do not influence the delay at S_2 [Lin84]. Tables 4.2(a) and 4.2(b) compare the delay calculations, at the output node of the gates, for each of the two cases, (a) and (b), with SPICE. The delays obtained from the input of the NAND gate to the output of the NOR gate for the different situations are in all the cases within a 10% agreement with SPICE, although the delays obtained for the individual gates are in some cases within 30% of SPICE.

An interesting situation arises in the evaluation of the delay of the NOR gate in case (b) when $C_{L1} = 10pF$ (A very slow signal is applied to the input of the NOR gate). For very slow signals, and for a stage with a small slope ratio as the NOR gate, there is a significant overestimation of the delay by the new model, due to the fact that as the rise time of the input signal increases, the delay decreases and eventually reaches 0. This situation can not be predicted by the model, because it assumes that the delay of the driven stage increases as the delay of the driving stage increases by a ratio which depends on the slope ratio of the stage.

4.3 CMOS Shift Register Cell

Fig. 4.3 (a) shows a CMOS shift register cell composed of two standard inverters and two transmission gates. Two different cases are considered. In the first case, both transmission gates are already ON when the signal propagates through them. In the second case the transmission gate, TG_1, is turned ON after the transition at the output of inverter INV_1 has settled. The equivalent RC networks of the transmission gates include reconvergent paths, therefore, the node-splitting technique must be used to evaluate the delays (Sec. 2.5). It must be noted that when the transmission gate is transmitting a logic value, the effective resistance of each of the

two transistors must be evaluated according to this value, i.e., if the gate is transmitting a 1 the effective resistance of both the p- and n-channel devices correspond to the case when a 1 is being transmitted. If the transmission gate is passing, the two R_{eff}'s are obtained directly from the technology file (Table 3.6(a)). If both transistors are switching at the same time, the values are obtained from different tables in the model (see Sec. 3.6). If the transistors switch at different times, the one which switches first is considered as passing when the second transistor switches to ON (separate evaluations by the simulator).

Fig. 4.3(b) shows the different RC networks used to evaluate the delay as the signal propagates through the shift register cell. The estimated delays and SPICE results for both of the cases mentioned above appear in Tables 4.3(a) and 4.3(b). The obtained results show that the model estimation of the propagation delay through a switching transmission gate is accurate, whereas the delay calculated for a passing transmission gate is underestimated by about 50%. However, this fact will have little effect in the overall performance of the model, given the negligible delay of passing transmission gates as compared to the delays of other components of the circuit.

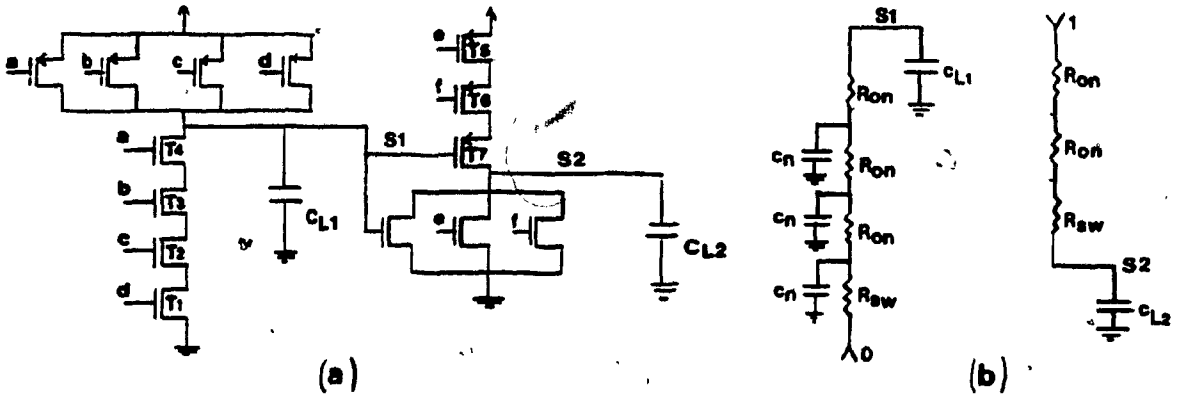


Fig 4.2 (a) A NAND gate drives a NOR gate.
(b) Equivalent RC networks for case (a) shown at foot of Table 4.2(a).

$C_{L1}(pF)$	PATH	$t_d[SPICE](ns)$	$t_d[McSLADE](ns)$
0	$d-S_1$	7.1	5.2
	S_1-S_2	9.3	10
	$d-S_2$	16.4	15.2
0.2	$d-S_1$	16.5	16.5
	S_1-S_2	13	12.3
	$d-S_2$	29.5	28.2
0.5	$d-S_1$	30.4	32.5
	S_1-S_2	17.8	14.1
	$d-S_2$	48.2	46.6
5	$d-S_1$	260.1	289
	S_1-S_2	45.2	32
	$d-S_2$	305.3	321

Table 4.2 (a) Case (a): $a=b=c=1, e=f=0, d=0\rightarrow 1, C_{L2}=0.13pF$

$C_{L1}(pF)$	PATH	$t_d[SPICE](ns)$	$t_d[McSLADE](ns)$
0	$d-S_1$	2.7	2.6
	S_1-S_2	4.8	4.8
	$d-S_2$	7.5	7.4
0.5	$d-S_1$	9.5	8.2
	S_1-S_2	7.5	8.1
	$d-S_2$	17	16.3
1	$d-S_1$	15.8	13
	S_1-S_2	9	8.9
	$d-S_2$	24.8	21.9
10	$d-S_1$	129	121
	S_1-S_2	9	20
	$d-S_2$	138	141

Table 4.2 (b) Case (b): $a=b=c=1, e=f=1, d=1\rightarrow 0, C_{L2}=0.23pF$

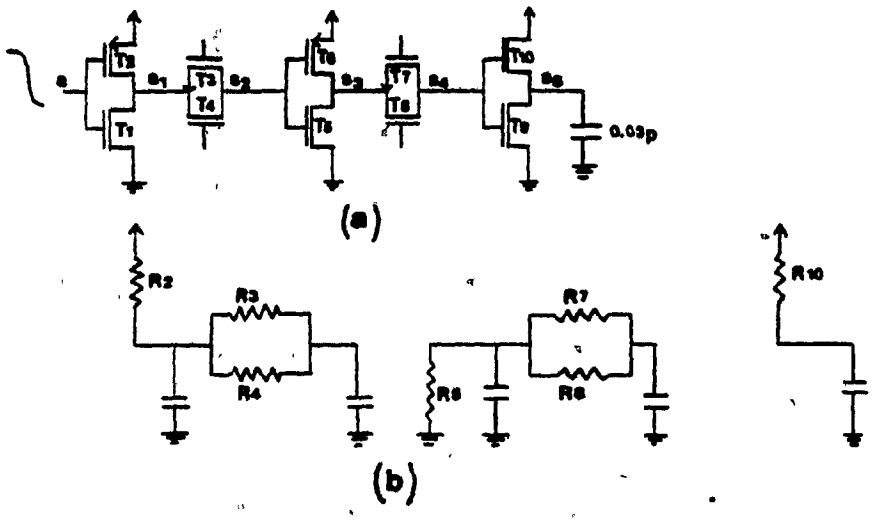


Fig. 4.3 (a) Static CMOS Shift Register Cell
(b) Equivalent RC networks for a high-to-low transition at a

PATH	$t_d[SPICE](ns)$	$t_d[McSLADE](ns)$
$a - S_1$	1.7	1.8
$S_1 - S_2$	0.9	0.5
$S_2 - S_3$	2.3	2.5
$S_3 - S_4$	0.9	0.5
$S_4 - S_5$	1.7	2.2
$a - S_5$	7.5	7.5

Table 4.3 (a) Case (a): $\Phi_1 = \Phi_2 = 1, \overline{\Phi}_1 = \overline{\Phi}_2 = 0, a = 0 \rightarrow 1$

PATH	$t_d[SPICE](ns)$	$t_d[McSLADE](ns)$
$\Phi - S_2$	1.7	1.7
$S_2 - S_3$	2.2	2.2
$S_3 - S_4$	0.9	0.4
$S_4 - S_5$	1.7	2.2
$\Phi - S_5$	6.5	6.5

Table 4.3 (b) Case (b): $a = 0, \Phi_2 = 1, \overline{\Phi}_2 = 0, \Phi_1 = 0 \rightarrow 1, \overline{\Phi}_2 = 1 \rightarrow 0$

4.4 CMOS circuit with different transistor sizes

A CMOS circuit consisting of a 2-input CMOS NOR gate driving two inverters appears in Fig. 4.4. Transistors of different sizes are used in the circuit. In addition, inverter 3 has been deliberately selected with a different transfer characteristic than that of a minsize inverter in which configuration the R_{eff} has been extracted. In this case, due to its different transfer characteristic, when the output of inverter 3 is switching to 0, for example, the current flowing through its minsize n-type transistor is different than the current that flows in the same minsize transistor within a minsize inverter ($L/W = n : 1/1, p : 1/2$). Therefore, the R_{eff} of the minsize n-channel in inverter 3 is different than the value provided by the model and some errors might be expected here in the delay calculations, although as the simulation results show, they are not significant.

Consider the case when $a = 0$ and a double transition occurs at input b of the NOR gate. First, b switches from $1 \rightarrow 0$, and as a result, S_1 goes high while S_2, S_3, and S_4 go low. The second transition, $b : 0 \rightarrow 1$ causes S_1 to go high and S_2, S_3, and S_4 to go low. Table 4.4 shows the obtained delays at all the nodes in the circuit for both transitions. As the results show there is a good agreement with SPICE for all the cases. The fact that one of the constructs has a different transfer characteristic does not appear to affect the performance of the model.

4.5 Dynamic CMOS PLA

Fig. 4.5 shows the critical path of a dynamic CMOS PLA. The design is based on the NORA technique [Gon83]. Poly and metal capacitances are considered. During the precharge phase ($\Phi = 0, \overline{\Phi} = 1$), nodes S_1 and S_2 are precharged to 1 and 0 respectively. The inputs to the right NOR plane arrive during this phase, and are assumed stable at the beginning of the evaluation phase ($\Phi = 1, \overline{\Phi} = 0$).

Assume that at the beginning of the evaluation phase $a = b = c = 0$ and $d = 1$. The expressions for the delay calculation at nodes S_1 and S_2 for this case are given by:

Precharge:

$$At \quad S_1: \qquad t_d = R_{sw_1}^{T1} C_{L1}$$

Evaluation:

$$At \quad S_1: \qquad t_d = R_{sw_0}^{T3}(C_{n1} + C_{L1}) + R_{on_0}^{T2} C_{L1}$$

$$At \quad S_2: \qquad t_d = R_{sw_1}^{T4}(9C_n + C_{L2}) + R_{on_0}(36C_n + 9C_{L2})$$

where C_n represents the node capacitance of two transistors in series resulting from adding the source capacitance of one transistor to the drain capacitance of the other, and R_{on}^0 is the effective resistance of the passing p-channel transistors, which are all equal.

Table 4.5 shows the high level of agreement with SPICE, despite the long chain of p-type transistors.

4.6 NMOS dynamic RAM Cell

Fig. 4.6(a) shows two three-transistor dynamic RAM cells. Information is entered into the RAM through the input inverter when $\Phi_{in} = 1$, and writing operation occurs when one of the write signals, wr, is enabled. Information in the cell is stored in node m. When $rd = 1$, a reading operation is performed. Transistor $T1$ is used to precharge the bus when no writing or reading is occurring. The input sequence of Fig. 4.6(b) was simulated. A logic 1 is being stored in node m_1, and after a cycle, the contents of m_1 is read and placed in the bus. Different slopes were used for the *write* and *read* signals input to the circuit. Table 4.6 shows the predicted transition times at the circuit storage and bus nodes using SPICE and McSLADE, for the different slopes of wr_1 and rd_1.

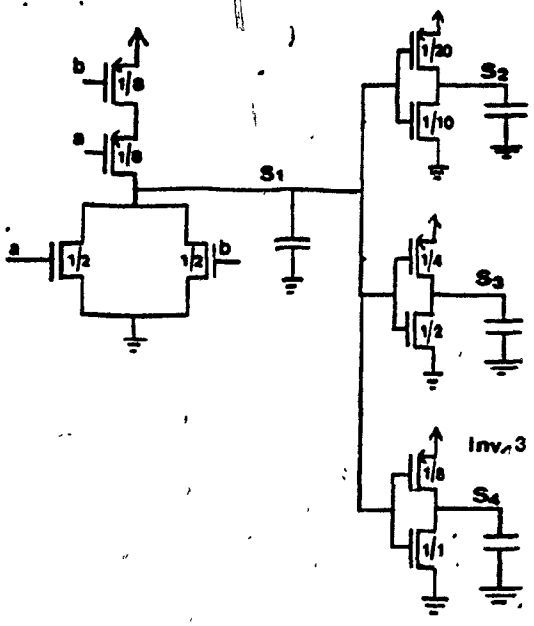


Figure 4.4 CMOS circuit with different transistor sizes

Transition at b	Node	$t_d[SPICE](ns)$	$t_d[MiSLADE](ns)$
$1 \rightarrow 0$	S_1	7.8	7.7
	S_2	15.6	13.8
	S_3	18.4	17.9
	S_4	19.5	16
$0 \rightarrow 1$	S_1	41.4	39.4
	S_2	48.6	45.5
	S_3	50.5	47.5
	S_4	44.5	43.5

Table 4.4

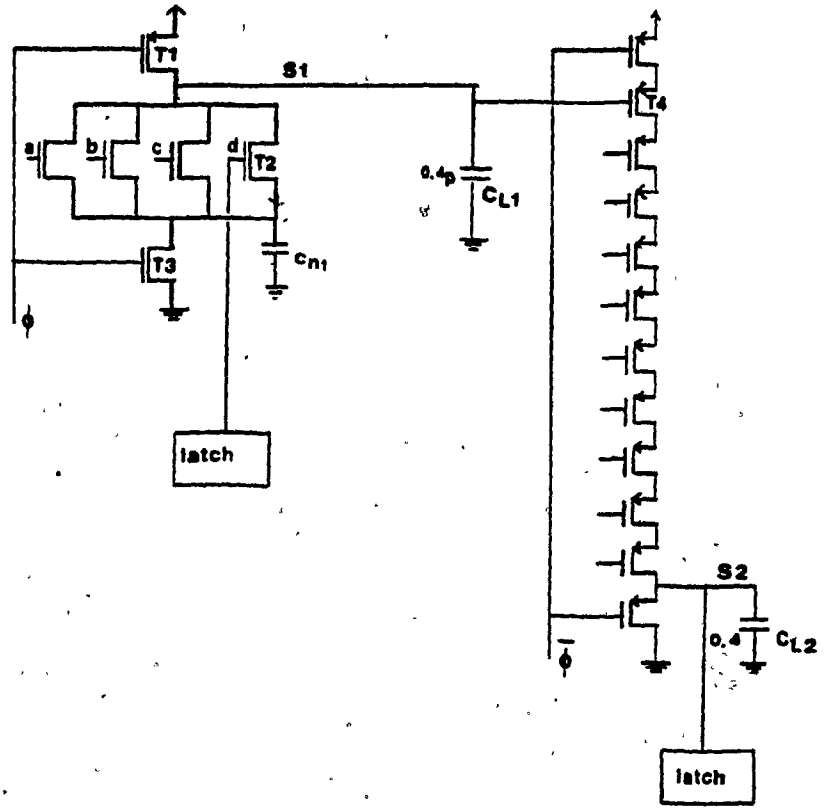


Fig.4.5 CMOS NORA PLA.

CASE	PATH	$t_d[SPICE](ns)$	$t_d[McSLADE](ns)$
Precharge	$\Phi - S_1$	8.2	8.5
Evaluate	$\Phi - S_1$	12.2	12.4
	$S_1 - S_2$	82.2	79.4
	$\Phi - S_2$	94.4	91.8

Table 4.5.

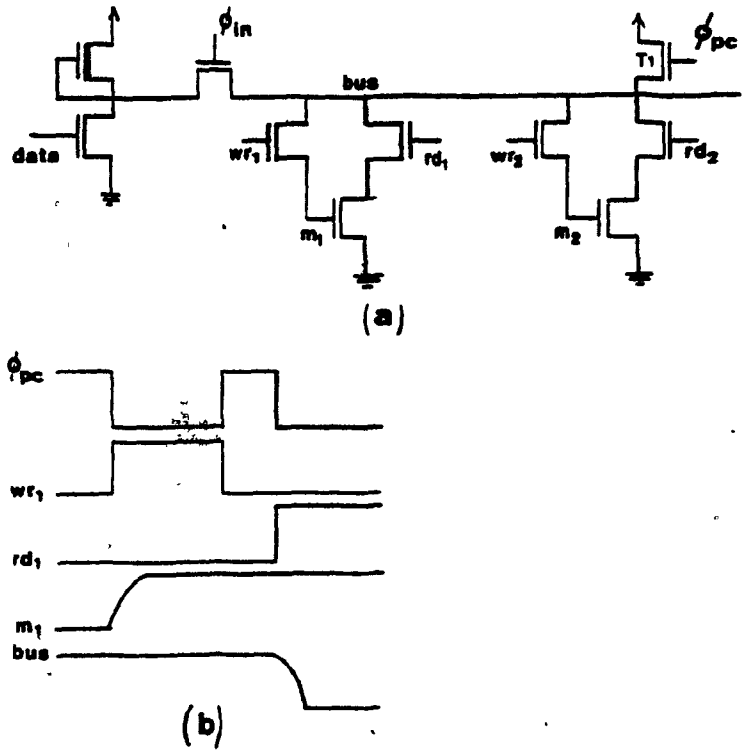


Figure 4.6 (a) Dynamic NMOS RAM Cell.
(b) Input and response sequence.

$t_{in}(ns)$	Node	$t_d[SPICE](ns)$	$t_d[McSLADE](ns)$
1.5	m_2	5.5	6
	bus	20.7	21
5	m_2	5.5	6
	bus	20.7	21

Table 4.6

4.7 Complex Static CMOS Gate

Fig 4.7(a) shows a complex static CMOS gate. Two different cases (shown at the foot of the tables) are analyzed, each of which involves two transistors switching simultaneously. In both cases, the equivalent RC network includes reconvergent paths. In such a case, node splitting (Sec.2.5) is used to reduce the network to an RC-tree, and the delays are computed by a relaxation method [Lin84]. An interesting situation arises in the case (a) , when $b = c = d = 1$, and a and e switch from $0 \rightarrow 1$. The two different RC trees which may result after node splitting, depending on which node is split, are shown in Figs. 4.7(b) and 4.7(c). Due to the fact that the effective resistances of the switching transistors $T1$ and $T2$ are extracted according to the initial load at the beginning of the relaxation, their values migth be different in each node splitting case. Therefore the delay estimates at the output of the gate and at internal nodes may yield different results in each of the cases. In other words, the delays calculated using the Lin and Mead algorithm could depend on how the node splitting is done. Simulation results for this case and for many similar cases show that while the delays estimated at internal nodes are indeed different for each possible resultant RC-tree, the delay estimates at *the output* of the transistor group remain approximately equal for all possible node splittings. Table 4.7(a) compares the delays obtained with McSLADE in the two possible cases of node splitting shown in Figs. 4.7(b) and 4.7(c), for all the nodes in the tree.

The simulation results show that for the output node, S, there is almost no variation in the obtained delays, although for the internal nodes n_1, n_2 and n_3, the differences are considerable. In general, the use of the new delay model within the Lin and Mead algorithm, yields inconsistent results for delays evaluated at internal nodes of groups containing reconvergent paths.

An additional aspect arising from the use of the relaxation algorithm is the fact that the effective resistance of a switching transistor is adjusted with the load only at the beginning of the relaxation process. The capacitances of the resultant subtrees are varied during the relaxation, and at the end their values are different than at the beginning. Therefore, errors might be expected here, in the evaluation of the delays. However the results obtained for this circuit, and for similar ones show that this procedure does not introduce big errors, basically due to the fact that the final values of the capacitances at the end of the relaxation are not very different from those assigned at the beginning (refer to Section 2.5 and to [Lin84]). Tables 4.7(b) and 4.7(c) show the results for the circuit of Fig. 4.7, for case (a) and case (b), as specified at the foot of the corresponding tables, each case for different capacitive loads (C_L) and input slopes (t_{pd}).

4.8 CMOS 1 bit ADDER

Fig. 4.8(a) shows a 1-bit CMOS ADDER containing about 80 transistors. For a given input sequence, simulation results for different transitions at the output and carry out appear in Fig. 4.8(b) and Table 4.8. Accuracy between the new model and SPICE is within 3% for this particular example. The overhead associated with the delay calculation is 2.7, as compared to the unit delay version.

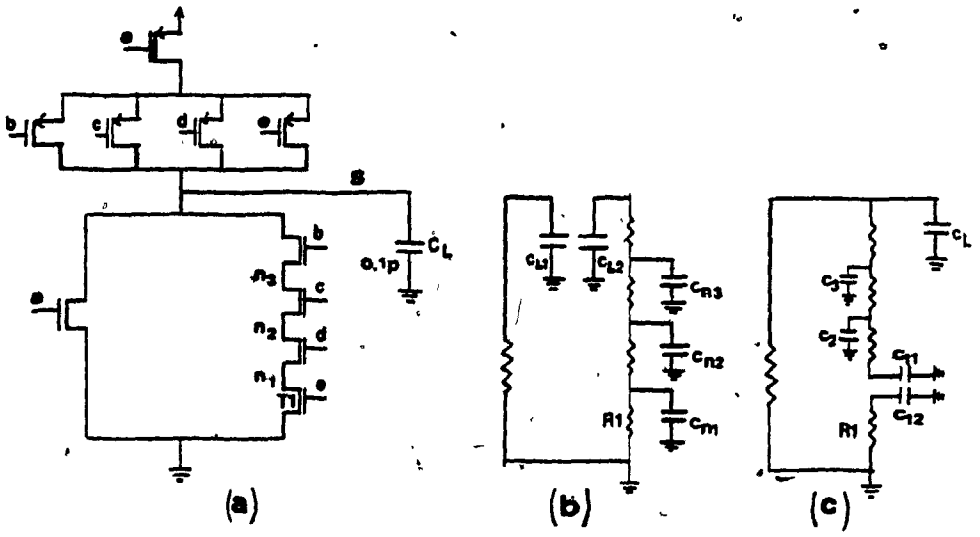


Figure 4.7 (a) Complex CMOS gate. (b) Secondary network after splitting node S. (c) Secondary network after splitting node n_1.

Node	Delay (ns) node split: S	Delay (ns) node split:$N1$
S	3.2	3.1
n_1	1.5	3.0
n_2	2.3	3.0
n_3	2.9	3.1

Table 4.7(a)

$C_L(pF)$	PATH	t_{pd} (ns)	$t_d[SPICE](ns)$	$t_d[McSLADE](ns)$
0.13	a-S	1.4	2.7	3
		3.5	3.5	3.7
		8	4.4	5.3
0 53	a-S	1.4	6.3	6.3
		8	8.9	10.5
		20	11.7	15.3

Table 4.7 (b) Case (a):$b = c = d = 1, a = 0 \rightarrow 1, e = 0 \rightarrow 1$

$C_L(pF)$	PATH	t_{pd} (ns)	$t_d[SPICE](ns)$	$t_d[McSLADE](ns)$
0.1	b-S	1.4	4.7	5.3
		3.5	5.5	6
		20	8	9
		30	8.4	10

Table 4.7 (c) Case (b):$d = e = 1, a = 0, b = 1 \rightarrow 0, c = 1 \rightarrow 0$

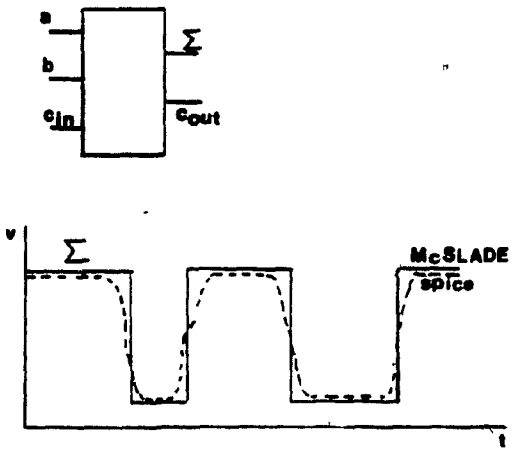


Figure 4.8 1-bit static CMOS ADDER.

Transitions	$T_d[SPICE](ns)$	$T_d[McSLADE](ns)$
Σ	33	32.5
	46.2	47.4
	77 5	80.2
	121	120
C_{out}	34.5	37
	70	71.3

Table 4.8

4.9 CMOS 8-to-1 Selector

An interesting circuit, as far as the model is concerned, is the CMOS 8-to-1 selector shown in Fig. 4.9. The circuit consists mainly of transmission gates, some of which are connected in series. In addition all the transmission gates belong to the same transistor group, i.e., they are all simulated simultaneously. The result is a circuit with many reconvergent paths in which node splitting is applied at different parts of the circuit. Table 4.9 show the times at which the transitions occur at the output of the selector and at some internal nodes, using both SPICE and McSLADE. Agreement is within 3% for all the cases. The overhead associated with the delay calculation as compared to the unit delay simulator is 4.3. The total number of relaxation steps for the input sequence of Table 4.9 is 624. The big overhead is a consequence of the large number of reconvergent paths that must be evaluated at each event. However, very accurate results were obtained for the selector at both internal and output nodes.

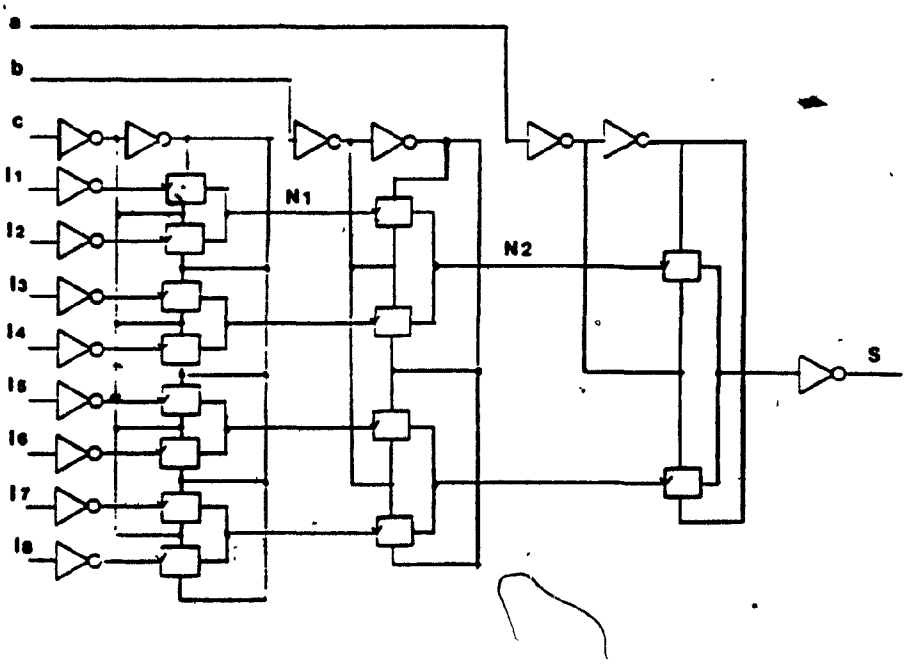


Fig 4.9 8-to-1 CMOS Selector.

Node	Transition	$t_d[SPICE](ns)$	$t_d[McSLADE](ns)$
S	$0 \rightarrow 1$	17	16.8
	$1 \rightarrow 0$	30.7	29.6
	$0 \rightarrow 1$	55.3	56.8
$N1$	$1 \rightarrow 0$	10.5	11.2
	$0 \rightarrow 1$	24.7	25.1
	$1 \rightarrow 0$	47.7	47.5
$N2$	$1 \rightarrow 0$	12.5	12.5
	$0 \rightarrow 1$	27.2	26.1
	$1 \rightarrow 0$	50.5	52.5

Table 4.9

CHAPTER 5

CONCLUSIONS

An improved delay model has been proposed in this thesis for the switch-level simulation of MOS VLSI circuits. It is based on the RC representation of the circuit and accounts for most of the effects that influence delays in MOS circuits. The transistor is modelled by an RC network in which the capacitances are calculated at the preprocessing and remain unchanged during simulation. The resistance, on the other hand, is computed dynamically during simulation, according to the size and type of the transistor, its context, capacitive load and gate input waveform. Two basic contexts are recognized by the model depending on whether the transistor is passing or switching. For transistors passing, two fixed values of effective resistance are used, one for the transistor transmitting a high logic value and the other for a low logic value. For switching transistors the effective resistance is computed according to the dynamic load driven by the transistor and the slope of the gate waveform. The effect of the slope is corrected for by means of a factor that depends on the delay of the driving gate. The model for the switching transistors consists of two small tables for each transistor type. From the first table the value of the effective resistance is extracted according to the load driven by the transistor. From the second table, a multiplying factor which depends on the *slope ratio* is obtained to adjust the value of the effective resistance for the corresponding input slope.

The delay model has been incorporated into a new switch-level simulator, McSLADE, and test runs were performed on a wide variety of MOS circuits, both NMOS and CMOS. The results obtained show that the overall delay estimates of the new model are usually accurate to within 10% of SPICE for medium size circuits, although for individual stages, in some cases, the difference with SPICE may be around 25%. The overhead introduced by the model is between 2.5 and 3 with respect to a unit delay version of the simulator.

The objective of the switch-level delay models is to provide a first-order timing analysis of the circuit, by means of the delay estimates. In the hierarchical environment of the VLSI design process, these estimates can be used to discover design flaws or to identify time-critical paths. In this context the new delay model presented in this thesis yields acceptable results for most applications.

The main problem associated with the new switch-level delay model appears to be a degradation of the delay estimates for very slow signals. In some of the test circuits the slope was varied by up to two orders of magnitude and some of the results obtained for the slow signals were up to 50% in disagreement with SPICE and in most of these cases, the delay was overestimated. This occurs when very slow signals are applied to stages with small slope ratio in which case the output tends to follow the input causing a decrease in the delay. Since the model assumes the delay to be proportional to the rise time, it can not predict a decreasing delay for an increasing rise time (slow signal).

REFERENCES

[Agn80] D. Agnew, "SCAMPER - Circuit Design for the 1980's." *TELESIS*, Vol. 7, pp. 2-9, 1980.

[Bre76] M.A. Breuer and A.D. Friedman, "Diagnosis and Reliable Design of Digital Systems," *Computer Science Press*, Woodland Hills, Calif.. 1976

[Bry80] R.E Bryant, "An Algorithm for MOS Logic Simulation," *Lambda*, vol. 1, pp. 46-53, fourth quarter 1980.

[Bry84] R.E Bryant, "A Switch-Level Model and Simulator for MOS Digital Systems," *IEEE Trans. Comput.*, vol. c-33, pp. 160-177, February 1984.

[Cas78] G. Case, "The SALOGS Digital Logic Simulator," *Proc. 1978 IEEE Int. Simposium on Circuits and Systems*, N.Y. pp 5-10, 1978.

[Cer84] E. Cerny and J. Gecsei, "Simulation of MOS circuits by decision diagrams," *IEEE trans. CADICS*, to appear.

[Cha75] B.R. Chawla, H.K. Gummel and P. Brennan, "MOTIS-An MOS timing simulator," *IEEE Trans. Circuits and Systems*, vol. CAS-22, pp. 504-509, June 1975.

[Che84] C.F. Chen et al. "The Second generation MOTIS mixed-mode Simulator," *Trans. 21st Design Automation Conference*, pp. 10-17, 1984.

[Dem78] H. de Man, G. Arnout, P. Reynaert, "Mixed-mode Circuit Simulation Techniques and their implementation in Diana," *Proc. NATO Advanced Study Institute on Computer Design Aids for VLSI.* Urbino, Italy, 1980.

[ElM74] Y.A. El-Mansy and A.R, Boothroyd, " A Simple Two-Dimensional Saturation Model of Short Channel IGFETS for CAD Applications," *Tech. Digest Inter. Electron Devices Meeting*, Wash. DC, pp. 35-39, 1974.

[Elm48] W.C Elmore, "The Transient Response of Damped Linear Networks with Particular regard to Wide-Band Amplifiers," *Journal of Applied Phisycs*, vol. 19, pp. 55-63, January 1948.

[Fan77] S.P. Fan et al., "MOTIS-C: A new Circuit Simulator for MOS LSI Circuits," *Proc. IEEE Inter. Symp. Circuits and Systems.* pp. 700-703. April 1977.

[Gon83] M.F Goncalves and H.J. De Man, "NORA: A Racefree Dynamic CMOS Technique for Pipelined Logic Structures," *IEEE Journ. Sol. Stat. Circ.*, vol. SC-18, pp. 261-266, June 1983.

[Hac81] G.D. Hachtel and A.L. Sangiovanni Vin., "A Survey of Third-Generation Simulation Techniques," *Proc. IEEE*, pp. 1264-1280, Oct. 1981.

[Hay82] Hayes, "A Unified Switching Theory with Applications to VLSI Design," *Proc. IEEE*, vol. 70, pp. 1140-1151, October 1982.

[Ho75] C.W. Ho et al., "The Modified Nodal Approach to Network Analysis," *IEEE Trans. on Circuits and Systems*, Vol. CAS-22, pp. 501-509, Jan. 1975.

[Hod83] D.A. Hodges and H.G. Jackson, " Analysis and Design of Digital Integrated Circuits," *McGraw Hill*, 1983.

[Jou83] N.P. Jouppi, "T.V: An NMOS Timing Analyzer" *Proc. 3rd Caltech VLSI Conf.*, pp. 71-86, March 1983.

[Kho85] Khordoc K. "McSLADE, A Switch Level Simulator with Accurate Delay Estimation," Master Thesis in preparation, 1985.

[Lin84] T.M. Lin and C.A. Mead, "Signal Delay in General RC Networks," *IEEE Trans. on C.A.D.*, vol.CAD-3, pp.331-349, October 1984.

[Mea80] C.A Mead and L. Conway, "Introduction to VLSI Systems," *Reading, MA: Addison-Wesley*, 1980.

[Mey71] J.E. Meyer, "MOS Models and Circuit Simulation," *RCA Review*, Vol. 32, pp. 42-62, March 1971.

[MoL84] Z.L. Mo and M.R. Lightner, "A Two Parameter Delay Model for Switch-Level Simulation," *Proc. IEEE ICCD*, pp. 481-486, 1984.

[Nag75] Nagel, "SPICE2: A Computer Program To Simulate Semi-conductor Circuits," *ERL Memo ERL-M250*, Electronics Research Lab., U. of C. Berkeley, May 1975.

[Nha80] Nham and A.K. Bose, "A Multiple Delay Simulator for MOS LSI Circuits," *ACM* pp. 610-617, 1980.

[New78] A.R. Newton, "The Simulation of Large-Scale Integrated Circuits," *Memorandum No. UCB/ERL M78/52*, Electronic Research Laboratory, U. of C. Berkeley, July 1978.

[New80] A.R. Newton, "Timing, Logic, and Mixed-mode Simulation for Large MOS Integrated Circuits," *Proc. NATO Advanced Study Institute on Computer Design Aids for VLSI.* Urbino, Italy, 1980.

[Ous83] J.K. Ousterhout, "CRYSTAL: A Timing Analyzer for NMOS VLSI Circuits," *Proc. 3rd. Caltech Conference On VLSI*, Computer Science Press, 1983, pp.57-70.

[Ous84] J.K. Ousterhout, "Switch-Level Delay Models for Digital MOS VLSI," *Proc. IEEE 21st Design Automation Conference*, pp. 542-548. June 1984.

[Pen81] P. Penfield and J. Rubenstein, "Signal Delay in RC Tree Networks," *Proc. 18th Design Automation Conf.*, pp. 613-617; 1981.

[Rub83] J. Rubinstein, P. Penfield, and M. Horowitz, "Signal Delay in RC Tree Networks," *IEEE Trans. C.A.D.*, vol.CAD-2, pp. 202-211, July 1983.

[San80] A.L. Sangiovanni Vincentelli, "Circuit Simulation," *Proc. NATO Advanced Study Institute on Computer Design Aids for VLSI.* Urbino, Italy, 1980.

[Shi68] H. Shichman and D. Hodges, "Modelling and Simulation of Insulated-Gate Field-Effect Transistor Switching Circuits," *IEEE Journal of Solid-State Circuits*,pp. 285-289, Sept. 1968.

[Smi] D. Smith "CMOS-1B Design information and guidelines," Internal report, Semiconductor components group, Northern Telecom Electronics. Ottawa, May 1983.

[Sub84] P. Subramaniam, "Table Models for Timing Simulation," *IEEE Proc. Custom Integrated Circuits Conference*, pp. 310-314, 1984.

[Swa72] R.M. Swanson and J.D. Meindl, "Ion implanted Complementary MOS Transistors in Low-Voltage Circuits," *IEEE Journal Solid-State Circuits*, Vol. SC-7. April 1982.

[Szy72] Szygenda, "TEGAS-Anatomy of a General Purpose Test Generation and Simulation at the Gate and Functional Level," *Proc. 9th Design Automation Conf.*, pp. 116-127. June 1972.

[Szy75] Szygenda and E.W. Thompson, "Digital Logic Simulation in a Time-based Table-driven Environment: part 1, Design Verification," *IEEE Computer*, pp. 24-36, March 1975.

[Tam83] E. Tamura et al., " Path Delay Analysis for Hierarchichal Building Block Layout System," *Proc. 20th Design Automation Conference*, pp. 403-410, 1983.

[Tok83] T. Tokuda et al. "Delay-Time Modeling for ED MOS Logic LSI," *IEEE Trans. on Computer Aided Design*, Vol. CAD-2, pp. 129-134, July 1983.

[Ter83] Terman, "Simulation Tools for Digital LSI Design," Doctoral Dissertation, M. I. T., September 1983.

[Vla82] A. Vladimirescu and D.O. Pederson, " Performance Limits of the CLASSIE Circuit Simulation Program," *Proc. Inter. Symposium on Circuits and Systems*, Rome, pp. 1229-1232, May 1982.

[Vla79] A. Vladimirescu and S. Liu, "The Simulation of MOS Integrated Circuits using SPICE2," *ERL Memo No. M80/7*, Electronics Research Lab. U. of C. Berkeley, June 1979.

[War78] D.E. Ward and R.W. Dutton, " A Charge-Oriented Model for MOS Transistor Capacitances," *IEEE Journal Solid-State Circuits*, Vol.SC-13, Oct. 1978.

[Wee73] Weeks et al, "Algorithms for ASTAP - a Network Analysis Program," *IEEE Trans. Circuit Theory*, vol. CT-20, pp. 628-634, November 1973.

[Yan80] P. Yang et al., "SLATE: A circuit Simulation Program with Latency Exploritation and Node Tearing," *IEEE Proc. Inter. Conference Circuits and Computers*, New York, pp. 353-355, Oct. 1980.