# INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

**The quality of this reproduction is dependent upon the quality of the copy submitted.** Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

UMI®
800-521-0600

# Optimization of High-Speed CMOS Circuits with Analytical Models for Signal Delay

**Jingyuan Sun**

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partical Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec
Canada

June 1998

# ABSTRACT

## Optimization of High-Speed CMOS Circuits with Analytical Models for Signal Delay

Jingyuan Sun

Global optimization of high speed and high integration CMOS VLSI circuit is greatly in need in deep submicron regime with respect to signal delay, chip area and power dissipation. Accurate modeling of signal path delays is of particular importance in optimization. Although circuit level simulators like SPICE produce accurate and detailed delay information, analytical delay models are required in general because of the time consuming computation in SPICE simulations. New analytical delay models for both inverter and non-inverter stage of CMOS circuit in deep submicron regime are proposed in this work. The modeling takes into account circuit topologies and ramp input effect. The models were studied for various CMOS circuits of different complexities. Simulation results show an overall 10% difference and a considerably speed-up as compared to SPICE level 3 simulator. Based on the new analytical delay model, a circuit optimization program is developed, which is aiming to provide designer first hand information on circuit delay, area and power consumption and to help designer find the optimum design among different circuit topologies and transistor sizings, especially in submicron region. The program reads in circuit description from SIS - a multi-level logic synthesis and minimization system, maps it into CMOS circuit stages, analyzes the performance and finds the optimal circuit topology and sizing according to the design criteria.

## Acknowledgments

I would like to express my gratitude to my thesis supervisor Dr. A. Al-Khalili for his interest and guidance during the course of this work. My thanks also go to my husband Feng-wen Sun for his encouragement during my graduate studies.

Jingyuan Sun

# Table of Contents

# List of Figures

# List of Tables

# CHAPTER 1 Introduction and Literature Review

Through the last decades, CMOS technology has evolved as a major technology for VLSI design, leading to more and more complex chips operating at high speed with high performance. The demand for high speed, high integration and high performance circuits pushes VLSI circuit design into deep submicron regime. To solve the resulting complicated design problems, global optimization of digital CMOS VLSI circuits is needed with respect to signal delay, chip area and power dissipation. Accurate modeling of signal path delays is of particular importance in optimizing high-speed integrated circuits. Although device level simulators like SPICE produce accurate and detailed delay information, analytical models are required in general since SPICE simulation time increases rapidly with the number of transistors in the circuit. Different models have been developed to reduce the simulation time while keeping an acceptable accuracy. Depending on different levels defined, there are transistor level models and macromodels.

## 1.1 Macromodels

Different types of subcells are considered to be the building blocks of the circuit instead of individual devices in macromodeling. The general approach of macromodeling is to characterize the delay contribution of each type of subcell in the circuit with a single or a few design parameters in order to obtain a low dimensional optimization problem, and hence, to reduce the computational effort [1]. In the simplest case a scale factor is introduced for each gate in the circuit. Macromodels for MOS logic gates were presented by Matson [2][3]. A gate level delay model for CMOS logic gates is presented in [1], which is based on the analytical solutions by Hedenstierna and Jeppson for the delay of long channel CMOS inverters driven by input voltage ramps.

Gate level modeling has the advantage of reduced computation effort. However, optimizing scale factors in gate level algorithms might introduce a certain suboptimality in the resulting sets

of transistor sizes, since transistors cannot be sized individually according to their positions in the pull-up or pull-down branch of a logic gate. In contrast, transistor level optimization algorithms allow for position dependent sizing which is important for high performance circuits [4][5].

## 1.2 Transistor Level Models

There are several approaches modeling the signal delay at transistor levels. Among them are equivalent inverter method [15], RC-tree formulas [22] and several other methods on delay modeling of series-connected MOSFET structures (SCMS) [10][11].

a- Equivalent inverter method

With equivalent inverter method, a gate is collapsed into an equivalent inverter in order to facilitate the delay calculation. The procedure can be described as follows. Two paralleled connected transistors with respective transconductances $\beta_1$ and $\beta_2$ are replaced with a single transistor $\beta_{equ} = \beta_1 + \beta_2$. And two series-connected transistors are substituted by an equivalent transistor with transconductance $\frac{1}{\beta_{equ}} = \frac{1}{\beta_1} + \frac{1}{\beta_2}$. The delay of a CMOS circuit is based on the delay of the equivalent inverter. Since all the internal capacitances are lumped to the output node, delay may well be over-estimated. Also, this method ignores the fact that transistors operate in different modes, i.e. saturation or linear mode.

b- Switch-Level RC models

Most transistor optimization algorithms use RC delay model [5]-[8]. RC (resistance-capacitance) delay modeling techniques represent transistors as a resistance discharging or charging a capacitance. A variety of timing models have been developed to estimate the delay of logic gates, using the switch behavior of the transistors. There are simple RC delay, Penfield-Rubenstein model and Penfield-Rubenstein slope model.

In the simple RC model the total resistance of the pull-up or pull-down path is calculated and all the capacitance of nodes involved in switching are lumped onto the output of the gate. This

may result in pessimistic estimation on the delay because it assumes that all the internal capacitance has to be discharged or charged to switch the gate.

The Penfield-Rubenstein model was developed to calculate signal delays in generalized RC trees. Signal delay though MOS interconnect lines with fan-out may be obtained by tightening the upper and lower bounds for the step response of RC trees. For a group of transistors in series, this formulation simplifies to Elmore delay for an RC ladder, which is

$$t_d = \sum_i R_i C_i$$

where $R_i$ is the summed resistance from point $i$ to power or ground and $C_i$ is the capacitance at point $i$. This model is improved by Penfield-Rubenstein slope delay model which takes into account the rise or fall time of the input waveform. The Penfield-Rubenstein slope delay model combines the slope model with the Penfield-Rubenstein delay model. The slope model defines the intrinsic rise time (or fall time) as the rise time that would occur if the input was driven by a step function. The actual input rise time is then divided by this value to reach a rise-time-ratio, which indicates the degree to which the switched transistor is turned on.

Such delay models requires tables of transistor resistance values from which to calculate delays. Resistance has to be calculated for a variety of transistor widths and rise and fall times. The RC-delay models deviate from SPICE simulations by 10 to 20 percent for long-channel circuits [9]. However, the performance of the model is poor in submicron regime, particular for SCMS's such as NAND and NOR gates [10], which results from neglecting the non-linearities of the MOS-transistors. The inaccuracies might yield suboptimal set of transistor sizings during circuit optimization.

c- Analytical delay model for domino CMOS circuits

An analytical delay model for long-channel domino CMOS circuits is presented in [11]. By modeling the discharging voltage waveform at the output node as a piece-wise linear function of time, closed-form delay formulas are derived to estimate the delay of domino CMOS circuits

wherein a multitude of n-channel transistors form a series connection. Nine possible cases are classified in modeling the discharging voltage wave-form. Operation modes (saturation or linear) of the transistors in the series-connected pull-down network are taken into considerations. The model provides less than 10% error as compared to SPICE simulation over the entire design space for long-channel domino CMOS circuits modules [11]. However, no issue is addressed on the short-channel effect in submicron design.

d- Delay analysis of series-connected MOSFETs in submicron regime

A delay model for MOSFET circuits in submicron regime is proposed by Sakurai and Newton [10]. Based on their *nth* power law short-channel MOSFET model, the delay of a CMOS inverter can be calculated for its ramp input. The delay of series-connected MOSFET structures may therefore be estimated from its delay ratio over inverter, i. e. (delay of NAND/NOR)/(delay of inverter). This approach provides a method of delay estimation for series-connected MOSFET structures in submicron regime, such as NAND and NOR gates. However, there may be difficulties in including input waveform effect for SCMS.

## 1.3 Objectives of Our Work

As the trend of the VLSI technology goes into deep submicron, a good delay model not only for long-channel MOSFETs circuits but also for short-channel circuits is greatly in need. Our work proposes a new analytical delay model that overcomes the limitations described above. The new analytical delay model is based on a piece-wise linear voltage waveform model that closely estimates the output waveforms during the switching of a complex CMOS gate. The waveforms are determined analytically, which permits fast estimation and maximum speed-up for the simulation.

An analytical DC model of short-channel MOS device [12] is used in deriving the delay for short-channel circuits. Different factors that affect the delay of the circuits are taken into consider-

ation. These factors are input rise and fall time, the position of the trigger input that causes the discharging (or charging) of the output node, the node capacitances, fan-out and the operation modes (saturation, linear, cut-off) of transistors in the delay path. The new analytical model shows a good agreement with SPICE for different circuits and a significant speed-up over SPICE can be achieved.

Based on the analytical delay model and a new circuit collapsing technique, a circuit optimization program is developed. The program reads in the circuit description in BLIF format from the output of a logic optimizer - SIS [23]: an interactive multi-level logic synthesis and minimization system, analyzes the performance, finds the worst delay path and sizes the circuit according to it speed, power and area constraints. For each topology a set of circuits are obtained by varying the transistor widths along the longest delay path. The set of circuits with all the different topologies and transistor sizings form the solution space. For each circuit, several criteria (attributes) are computed analytically: delay ($T$), rise and fall time, dynamic power, area ($A$), $AT$, $AT^2$, and power-delay product. The performance data of circuits in the solution space form a decision matrix which is fed in a Multiple Attribute Decision Making process (MADM) [14] to find the optimal circuit implementation which meets all the design criteria.

This work intends to provide designers first-hand information on delay, area and power consumption and their trade-off on a set of design options without intensive circuit simulations. It is aiming to help designer find the optimum design among different circuit topologies and transistor sizings for CMOS circuits, especially in submicron region.

The thesis is organized as follows. Chapter 2 describes the procedures and considerations to analyze a complex CMOS circuit with multiple stages. Chapter 3 introduces the delay models for both inverter and non-inverter stages (Sum of Product stage) in submicron region. The short-channel effect and the factors that affects the delays are discussed. A long-channel delay model is given in Chapter 4. In Chapter 5, we compare the simulation results of our short-channel model

with that of SPICE level 3 simulator on circuits with various complexity. In Chapter 6, simulation results of the long-channel model on both pull-down and pull-up network are compared with SPICE level 3 simulation. Chapter 7 describes a circuit optimization program based on the new analytical delay models. Chapter 8 concludes the thesis work.

# CHAPTER 2 Performance Analysis in Large CMOS Circuits

## 2.1 Introduction

In this chapter, we present the method of handling performance analysis for a complex combinational CMOS circuit with multiple stages. The CMOS circuit is decomposed into stages and each SOP (Sum of Products) stage is collapsed into an equivalent gate. Performance analysis is carried out on each stage by an event-driven simulator. The step input response is obtained for each SOP stage, which is later adjusted to the slope input response along the worst delay path. The output waveform of each collapsed stage is modeled by a piece-wise linear function of time. Circuit attributes such as signal path delay, power consumption and area are calculated.

## 2.2 General Approach

A large CMOS circuit can be considered as a collection of CMOS stages. A stage consists of nodes and transistors forming an electrical path from a strong signal source to some other nodes. In circuits without pass gates the stage decomposition is static and corresponds to logic gates [9]. Here we only consider combinational CMOS circuits without pass transistors.

The timing analysis of the circuit, based on event driven simulation, is performed at the stage level. It uses a value-independent approach, which makes it different from simulation where a specific set of input signals is applied to a circuit. The objective of timing analysis, is to find the possible input combination which results in worst-case behavior.

Given a general CMOS circuit, we first divide it into stages - inverter stages and SOP (non-inverter) stages. Each stage is associated with a unique output and assigned two numbers - stage number (denoted as output node number) and stage level number. Level numbers are assigned so that no gate in stage $i$ is driven by a gate in stage level $j$, where $j \geq i$. The level of a stage is the

same as the level of its output and equal to its highest input level plus one. Dividing a circuit into stages and processing them according to the order of their level numbers ensure that each input of a stage are known at the time when the stage is processed. A unique stage number is assigned to each stage so that even if two stages have the same level (driven by same stage), they can be distinguished by their different stage numbers. As an example of level assignment, let us consider the carry look-ahead adder in Fig.2.1.

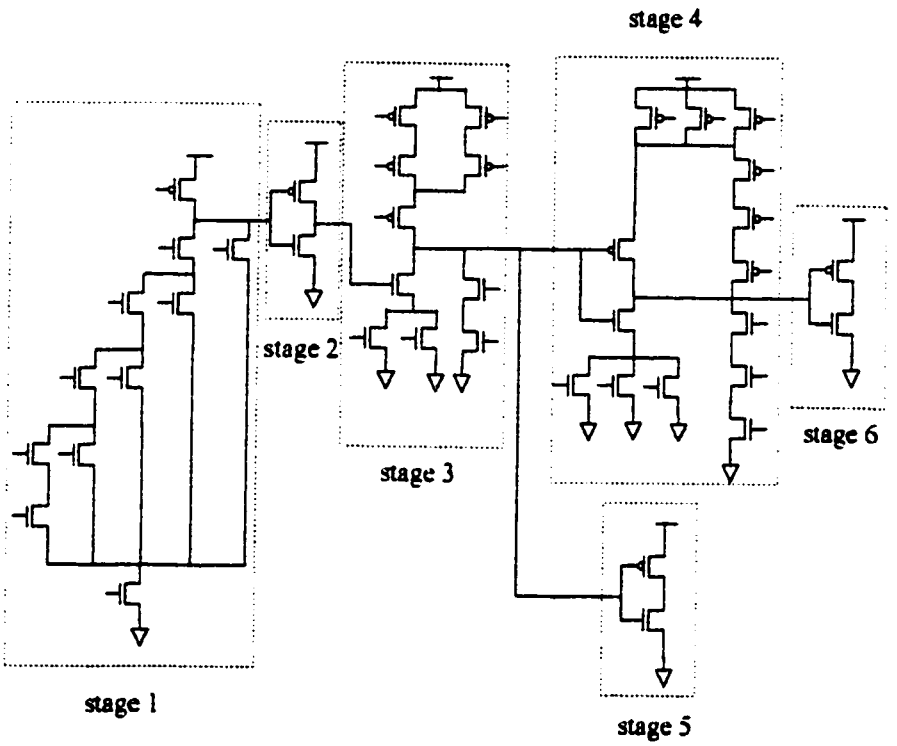


**Fig. 2.1: Carry look-ahead circuit.**

The circuit is divided into 6 stages as shown in Fig.2.2. The stages are numbered from 1 to 6. All the primary inputs are level 0 and the level of each stage is equal to its highest level of input plus one. Although stage 4 and 5 both are of level 5, their stage numbers are different.

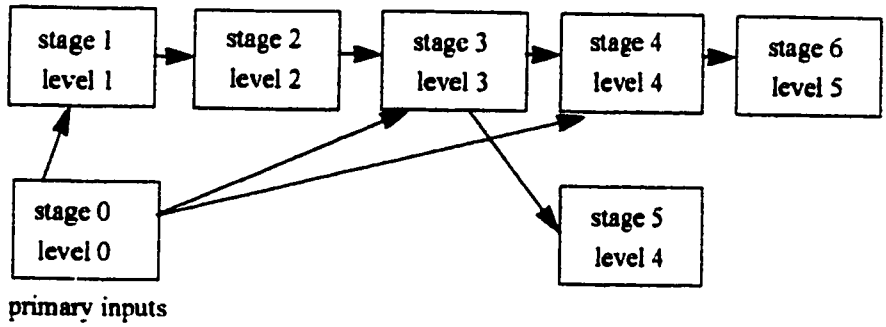


**Fig. 2.2: Stage assignment for carry-look ahead circuit.**

The procedures of the design optimization oriented performance analysis on a complex CMOS circuit are illustrated as follows.

Step 1: The circuit is divided into stages which are assigned with stage and level numbers.

Step 2: Each non-inverter stage is collapsed into a simpler gate.

Step 3: Calculate the step input response for each non-inverter stage.

Step 4: The input vectors are scheduled in an event pool.

Step 5: An event is picked up from the event pool and the output node waveform for each affected stage is determined based on a detailed piece-wise linear model that corresponds to stage input conditions and circuit characteristics. For a stage with multiple fan-ins, the worst delay fan-in node is used in delay calculation.

Step 6: A new event is added into the event pool and step 5 is repeated until the event pool is empty (when the primary output is reached).

Step 7: Find the primary output node with the worst delay. Acquire the longest signal delay path by tracking back its worst fan-in nodes stage by stage.

Step 8: Size the transistors along the worst delay path.

Step 9: Repeat step 3 to 8 until the area constraint is met.

Step 10: Obtain circuit performance data for design optimization.

To illustrate the procedures, consider the circuit in Fig.2.1 after stage assignment. The arrows in Fig.2.2 represent the directions of data flow. The waveforms of step input response are first obtained for stage 1, 3, and 4. Step input delay is estimated in each stage for the worst-case input combination. For a given input vector, stage 1 is processed first. The delay is determined based on its step input response and the voltage slope of the input vector. The result is put in an event pool which will allow the processing of stage 2. For a stage with multiple fan-ins, the fan-in node with the worst delay is chosen as the input of the stage. The output of stage 2 is therefore selected as the trigger input of stage 3. It is put in the event pool and allows stage 3 to be processed. Similarly, the procedures are repeated for the rest of the circuit until the event pool is empty and the analysis is done for all the stages.

Longest signal delay path is obtained by finding the worst delay output node and tracking back its worst fan-in nodes. For the circuit in Fig.2.1, the comparison of the propagation delay between path of stage 1, 2, 3, 4 to 6 and that of stage 1, 2, 3, 4 will reveal the longest delay path. Transistor sizing will be performed with a given sizing step along the longest delay path. Performance parameters are then re-evaluated for the sized circuit. This procedure repeats until the area constraint for a given circuit is met.

## 2.3 Collapsing a Complex CMOS Gate

Collapsing a complex CMOS gate into an equivalent simpler circuit is an effective approach to reduce the simulation time, however, at the cost of losing information on the circuits. To achieve a good accuracy, the collapsed circuit should preserve the delay, power (charge conservation) and current waveforms [13].

### 2.3.1 General Approach

The first step in collapsing a circuit is to find its trigger input. The first transistor that creates a path to ground (voltage source) is selected as the trigger transistor. The position and the input slope of the trigger transistor have considerable impact on signal delay. We assume that both trigger inputs are the same for the pull-up and pull-down network. Due to the large number of possible circuit configurations, we collapse the circuit based on its worst case scenario. The procedures for collapsing the pull-down network in a stage are shown below. Procedures for pull-up network is symmetrical.

1- Determine the trigger transistor (assuming that the triggers for both the pull-up and pull-down network are the same).

2- Determine the longest "on" path from the output node to the ground as shown in Fig.2.3. The pull-down network in each stage is therefore represented by a series-connected NMOS transistors.

3- The series-connected pull-down network obtained from previous step are further collapsed into two transistors in the pull-down networks as shown in Fig.2.4 (a) and (b) according to their operation modes.

We take the carry look-ahead adder in Fig.2.1 as an example. Following the procedures described above, the six-stage adder may be collapsed to a simpler circuit as shown in Fig.2.5.

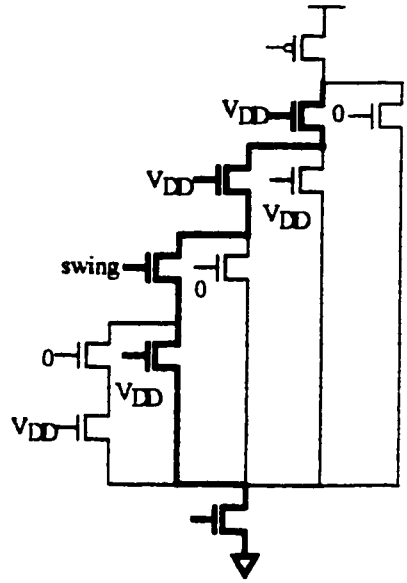


**Fig. 2.3: A complex CMOS gate.**

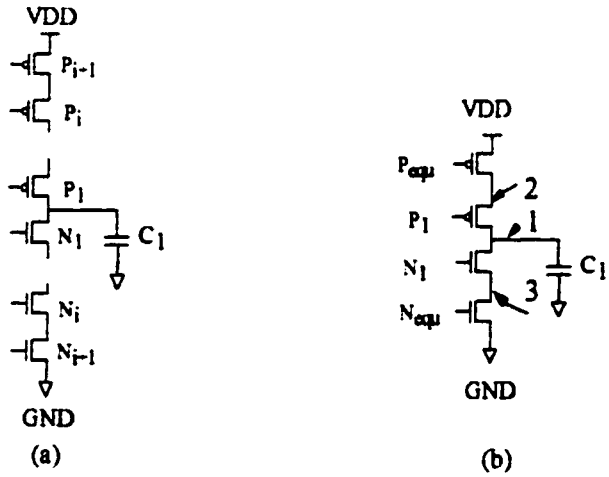


**Fig. 2.4: (a) CMOS gate collapsed to series-connected pull-up and pull-down network. (b) Series-connected pull-up and pull-down are further collapsed to four-transistor network.**

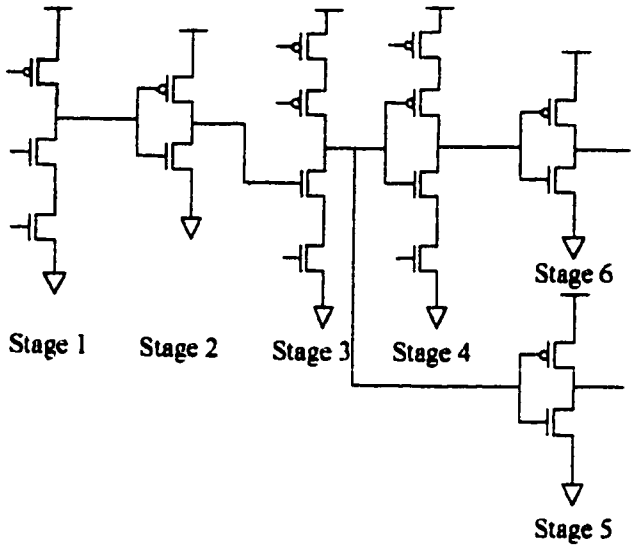


**Fig. 2.5: The Collapsed carry look-ahead circuits after four steps.**

The reason we collapse the series-connected pull-down network into two transistors as shown in Fig.2.4 (b) is that the top most transistor operates in different mode other than the rest of the transistors in the pull-down network when the gate is switched on. When the pull-down (pull-up) path is switched on $N_1$ ($P_1$) transistor is in saturation while the rest operate in linear region until the output node drops (rises) one threshold below $V_{DD}$ (one threshold above zero).

### 2.3.2 Delay Consideration

A straight forward delay comparison between a series-connected MOS network and an equivalent transistor is presented in [10], where a delay degradation factor $F_D$ is defined as the ratio of the delay of a SCMS to that of a single MOSFET and can be calculated as

$$F_D = \frac{I_D}{I_{D_N}}$$

$I_{D_N}$ and $I_D$ are the drain currents of a SCMS and a single MOSFET respectively.

For the case where the input is a step input and load capacitance is large, an RC model predicts that N series-connected MOSFET's would show approximately N times the delay compared

with a single MOSFET. For a long-channel MOSFET without body effect, the drain current $I_D$ in the linear mode can be decomposed into $f(V_D) - f(V_S)$, where $V_D$ and $V_S$ are the drain and source potential respectively and $f(V) = \beta\left[(V_{DD} - V_{T0})V - \frac{1}{2}V^2\right]$.

For the series-connected MOS transistors in Fig.2.6, neglecting the node capacitance current, we have:

$$I_{D1} = I_{D2} = \ldots = I_{Di} = \ldots = I_{Dn} \tag{2.1}$$

Substitute $f(V_D) - f(V_S)$ in (2.1), following equations are obtained,

$$I_{D1} = f(V_1) - f(V_2)$$

$$I_{D2} = f(V_2) - f(V_3)$$

$$\ldots$$

$$I_{Di} = f(V_i) - f(V_{i-1})$$

$$\ldots$$

$$I_{Dn} = f(V_n) - f(V_0)$$

Summing up the equations, we have

$$NI_{DN} = f(V_1) = I_D \tag{2.2}$$

which leads to $F_D = \frac{I_D}{I_{DN}} = N$.

From this simplified case we may conclude that the delay of $n$ series-connected MOS transistors in long-channel region is approximately $n$ times of the delay of a single transistor. It is clear that in this case, we can replace the n series-connected transistors by a equivalent transistor $N_{equ}$ with the equivalent transconductance $\beta_{equ} = \frac{\beta}{n}$ and still get the same delay.

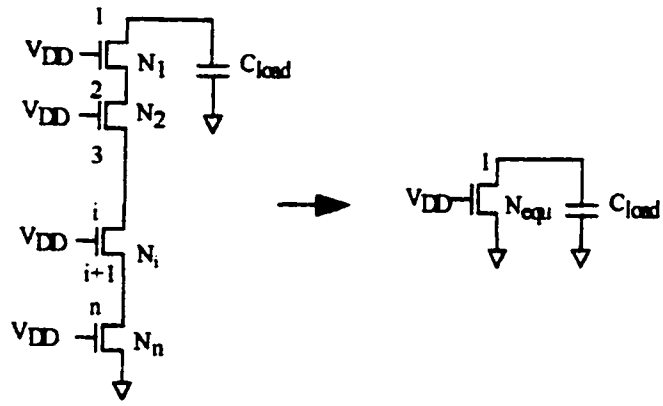


**Fig. 2.6: Series-connected MOSFETs.**

However, this relation does not hold for short-channel transistor. The delay ratio for short-channel transistors is given in [10]:

$$F_D = \frac{delay_{SCMS}}{delay_{inverter}} \approx 1 + \frac{1}{2} N \frac{V_{dsat}}{1 - V_T} (1 + \gamma)\ (N - 1) \tag{2.3}$$

where $V_{dsat}$ is the drain saturation voltage when $V_{GS} = V_{DD}$

$V_T$ is the threshold voltage

$N$ is empirical factor which describes the short-channel effect

$\gamma$ is the body effect coefficient

(2.3) indicates that for a SCMS in submicron region, the SCMS can be replaced by a equivalent transistor as far as delay is concerned. The transconductance of the equivalent transistor is given as

$$\beta_{equ} \approx \frac{\beta_{SCMS}}{1 + \frac{1}{2} N \frac{V_{dsat}}{1 - V_T} 1 + \gamma N - 1} \tag{2.4}$$

In the case of a small output load capacitance, an RC model predicts that $n$ series-connected

MOS transistors with small output load will have a delay of $n^2$ times the delay of a single MOS-FET. However, the real situation is more favorable to the SCMS in short-channel regime, as shown in [10]. The physical interpretation of the improvement of delay degradation factor in sub-micron region is that the velocity saturation is less severe in SCMS because of reduced $V_{DS}$ for each MOS transistor. The current improvement resulted from the smaller $V_{DS}$ thus overcomes the degradation induced by the body effect. For both large and small loads, (2.4) is an acceptable approximation to model the equivalent transconductance for a SCMS in long-channel and short-channel regime.

### 2.3.3 Charge Consideration

The switching of a CMOS circuit reflects actually a process of charging or discharging the output node capacitance. The charge stored in the circuit could therefore affect the performance of the circuit. To accurately model the original circuit, the collapsed equivalent circuit must hold the same charge stored.

The charge stored in a SCMS before switching may be obtained as

$$Q|_{t=0} = \sum C_i V_i$$

where $C_i$ is the node capacitance and $V_i$ is the node voltage. In order to model the original circuit, the collapsed equivalent circuit must have the same charge as that of the original circuit, i.e. the initial voltage in the equivalent circuit must be

$$V(0) = \frac{\sum C_i V_i\Big|_{t=0}}{C_{equ}} \tag{2.5}$$

For the equivalent circuit satisfying (2.5), the average dynamic power dissipation is proven to be the same as that of the original circuit [13]. This can be shown as follows.

$$P_{avg} = \frac{1}{T}\int_0^T V_{DD} I dt$$

Since $V_{DD}$ is a constant, the above equation can be rewritten as

$$P_{avg} = \frac{V_{DD}}{T}\int_0^T Idt = \frac{V_{DD}}{T}Q = \alpha Q \tag{2.6}$$

where $Q$ is the charge stored in the circuit. It is clear from (2.6) that the dynamic power dissipation of a CMOS circuit with a DC power supply is proportional to the charge stored in the circuit. For two circuits with the same switching frequency and same charge storage, their average dynamic power dissipation is the same.

The initial charge stored in the circuit depends on the position of the trigger transistor in a circuit. For example, in the SCMS shown in Fig.2.6, if transistor $N_i$ is the trigger transistor, all the nodes above $N_i$ are charged to $V_{DD}$-$V_T$ except the output node and all the nodes below the trigger transistor is discharged to 0v.

### 2.3.4 Supply Current Consideration

The supply current flows in a CMOS gate only during the transition of the output node. Fig.2.7 shows the currents during the switching of a CMOS gate. The trigger input is the transistor that causes the charging or discharging of the output node. $C_n$ and $C_p$ represent the node capacitances for the pull-up and the pull-down network.

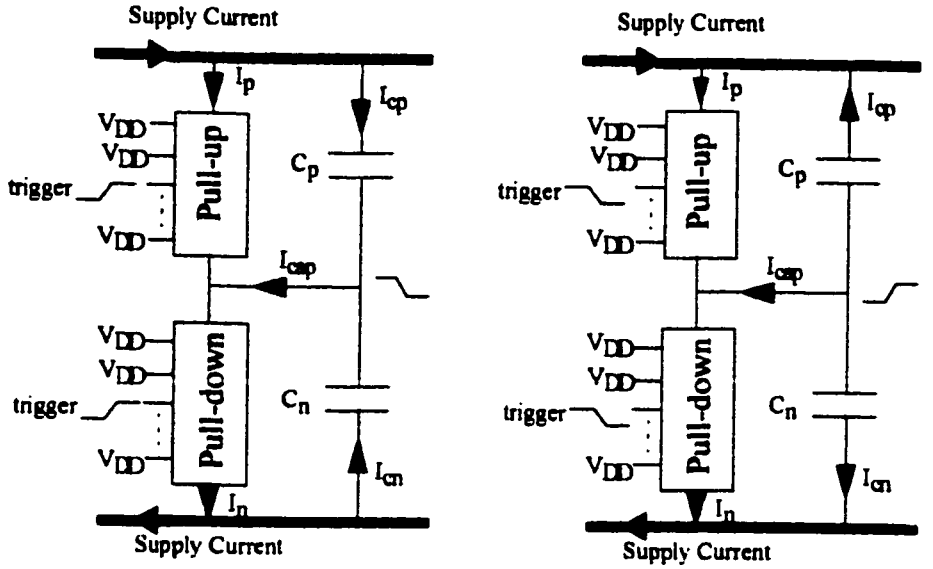


**Fig. 2.7: Current in a stage during switching.**

The supply current for a high to low transition can be written as

$$I_{supply} = I_p + I_{C_p}$$

Since $I_{cap} = (C_n + C_p)\frac{dV_{out}}{dt}$, $I_{C_p} = C_p\frac{dV_{out}}{dt}$ and $I_p = I_{short}$, we have

$$I_{supply} = I_{short} + \frac{C_p}{(C_n + C_p)} I_{cap} \tag{2.7}$$

In the same way, we can obtain the supply current for low to high transition

$$I_{supply} = I_{short} + \frac{C_n}{(C_n + C_p)} I_{cap} \tag{2.8}$$

Equations (2.7) and (2.8) show that the supply current can be derived from the capacitive current and the short-circuit current. Supply current also depends on the estimation of $C_n$ and $C_p$ $C_n$ and $C_p$ represent the contribution of the different node capacitances in the gate. Short-circuit cur-

rent depends on the load and input rise time [20]. It is clear from above analysis, that if we have the information on load, input rise time and the contribution of the different node capacitance passed over to the collapsed circuit, we will have approximately the same supply current as the original circuit.

### 2.3.5 Body Effect in Series-Connected MOS Transistors

For series-connected MOS transistors, the body effect is non-negligible. As presented in [11], we take into account the back-gate bias effect. The maximum threshold voltage at the pinch-off point is given in [11],

$$V_{Tn}^{*} = V_{Tn0} + \gamma_n \left\{ \sqrt{\frac{\gamma_n^2}{4} - V_{Tn0} + V_g + \gamma_n \sqrt{2\phi_F} + 2\phi_F} - \sqrt{2\phi_F} - \frac{\gamma_n}{2} \right\} \tag{2.9}$$

where $V_{Tn0}$ is the threshold voltage without body effect, $\gamma_n$ is the body effect coefficient, $\phi_F$ is the equilibrium Fermi level potential and $V_g$ is the gate-to-substrate voltage. When transistors are operating in the linear region, the threshold voltage is approximately by the average of the minimum and maximum threshold voltages,

$$\overline{V_{Tn}} = \frac{V_{Tn0} + V_{Tn}^{*}}{2} \tag{2.10}$$

For transistors operating in saturation, the threshold voltage is,

$$V_{Tnsat} = \alpha V_{Tn0}^{*} \tag{2.11}$$

where $\alpha$ is a correlation factor ranging from 0.5 to 1.0.

## 2.4 Conclusion

In this chapter, we have discussed the general procedures to analyze a complex CMOS circuit with multiple stages. Several considerations on delay, supply current, charge and body-effect are discussed when collapsing a complex circuit into a simpler circuit. In the next chapter, we will discuss the delay analysis in a single stage (inverter or sum of product stage) in submicron region. New delay models on both non-inverter stage and inverter stage are proposed.

# CHAPTER 3 Delay Analysis for Short-Channel MOSFETs

## 3.1 Introduction

As discussed earlier, our approach to delay estimation for a complex CMOS circuit is based on an event-driven simulator. For each SOP stage, the output waveform is modeled by a piece-wise linear function of time. In this chapter we present the analytical delay models for short-channel MOSFETs. Short-channel effects are discussed first in SECTION 3.2 since they have great impact on the modeling. The large signal model used in the analysis is presented in SECTION 3.3. SECTION 3.4 to SECTION 3.6 present the modeling scheme for series-connected MOSFETs structures. Model description is given in SECTION 3.4. In SECTION 3.5, we present the circuit analysis for step input. Delay with ramp input effect is discussed in SECTION 3.6. The analysis on inverter delay is presented in SECTION 3.7. Since for a complex circuit, the output of one stage may become the input of the next stage, effective rise (fall) time for an output waveform is required. SECTION 3.8 discusses the matching between the response of the effective ramp waveform and that of the real input waveform.

## 3.2 Short-Channel Effects

When channel length reaches the micron-range or below, it becomes comparable to other device parameters such as the depth of drain and source junctions, and the width of their depletion regions [16]. Such a device is called a short-channel transistor, in contrast to the long-channel devices. For a long-channel device, it is assumed that all current flows on the surface of the silicon and the electrical fields are oriented along that plane [17]. However, these assumptions no longer hold for short-channel devices. We consider several of the short-channel effects here when deriving our delay model for short-channel CMOS circuits.

### 3.2.1 Velocity Saturation and Mobility Degradation

The electrical characteristics of short-channel devices deviate considerably from those of long-channel devices. The most important reasons for this difference are the velocity saturation and the mobility degradation [17]. In long-channel devices, the velocity of the carriers is proportional to the electrical field, $v = \mu_{eff}E$, where the effective carrier mobility $\mu_{eff}$ is assumed to be a constant. For short-channel devices, $\mu_{eff}$ is no longer a constant but instead a function of the transverse field $E_t$ [17].When the electrical field along the channel reaches a critical value $E_c$, the velocity of the carriers tends to saturate. Fig.3.1 and Fig.3.2 show effect of electrical field on electron velocity and mobility.

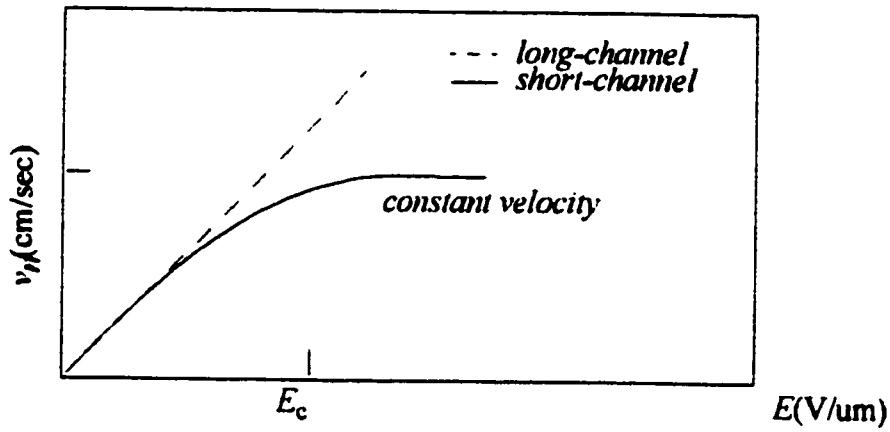


**Fig. 3.1: Velocity saturation for short-channel device.**

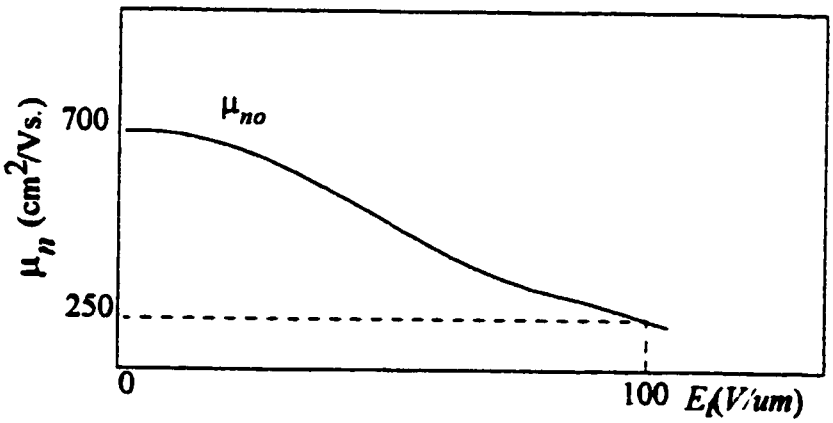


**Fig. 3.2: Mobility degradation.**

Taking into account velocity saturation and mobility degradation, we have [17]

$$\mu_{eff} = \frac{\mu_0}{1 + \theta (V_{gs} - V_T)} \tag{3.1}$$

where $\mu_0$ is the low field bulk mobility, $\theta$ is a mobility degradation factor.

### 3.2.2 Derivation of $V_{dsat}$

In long-channel MOSFET devices, saturation voltage is approximated by

$$V_{dsat} = V_{gs} - V_{Th} \tag{3.2}$$

However, in short-channel MOS device, (3.2) no longer works. It is given in [19] as,

$$V_{dsat} = (1 - \kappa)(V_{gs} - V_{Th}) \tag{3.3}$$

where

$$\kappa = \frac{1}{1 + \frac{E_c L_e}{V_{gs} - V_{Th}}} \tag{3.4}$$

$$E_c = \frac{2 v_{esat}}{\mu_{eff}} \tag{3.5}$$

$E_c$ is the critical field at which the carriers are velocity saturated, and $v_{esat}$ is the saturation velocity. $L_e$ is electrical channel length and is given by [19],

$$L_e = L_{eff} - \chi_d \tag{3.6}$$

where $L_{eff}$ is the effective channel length, and $\chi_d$ is the drain depletion width into the channel in saturation region. $\chi_d$ is a function of $V_{ds} - V_{dsat}$ and may be calculated as,

$$\chi_d = \frac{1}{A}\ln\frac{A(V_{ds}-V_{dsat})+E_d}{E_c} \tag{3.7}$$

where

$$\frac{E_d}{E_c} = \sqrt{1+\left[\frac{A(V_{ds}-V_{dsat})}{E_c}\right]^2}$$

$A$ is a semi-empirical constant.

Fig.3.3 shows the saturation voltage versus $V_{gs}-V_{Th}$. We can see that for a fixed channel length, saturation voltage is almost a linear function of $V_{gs}-V_{Th}$, which indicate that $\kappa$ may be approximated by a constant. Since $0<\kappa<1$, we can also see from (3.3) that saturation voltage in short-channel is smaller than that in long-channel under the same $V_{gs}-V_{Th}$, which indicates a wider saturation region in short-channel DC characteristics. As the channel length is reduced, the value of the saturation voltage deviates from the first order theory. It can be seen that the shorter the channel is, the more $V_{dsat}$ deviates from its low field value.

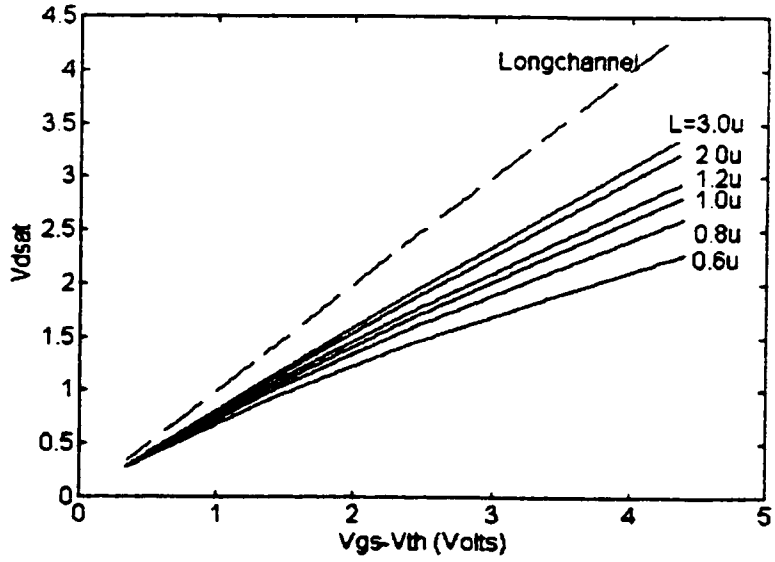


**Fig. 3.3: Saturation voltage vs. $V_{gs}$-$V_{th}$ for NMOS device in CMOSIS5 process.**

### 3.2.3 Threshold Variation

The threshold voltage for a long-channel device is only a function of the manufacturing technology and the applied body bias $V_{SB}$ [16]. As the device dimensions are reduced, however, the threshold potential becomes a function of $L$, $W$, and $V_{ds}$ [18]. It can be seen from Fig.3.4, that the threshold voltage is a function of effective channel length. Measured under $V_{ds}$ = 5.0v, for channel length of *3* micron to *0.6* micron the difference between maximum and minimum threshold voltage is about *0.06*v, i.e. $\Delta V_T < 0.1$. To simplify the model calculation, we will not take into consideration the threshold variation effect in our short-channel delay model. We only take into consideration the body effect discussed earlier in CHAPTER 2.

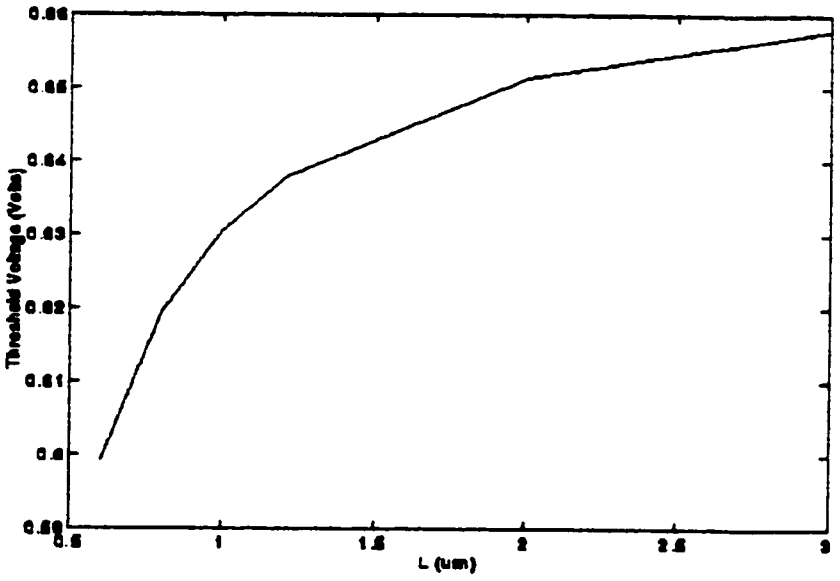


**Fig. 3.4: The threshold voltage as a function of $L$, measured at $V_{ds}$ = 5v.**

### 3.2.4 Applicability to Short-Channel Regime

How to decide whether a MOS device belongs to short-channel or long-channel regime? There is no restriction imposed on the device channel length $L$. However, from (3.3), we can see that as $\kappa \ll 1$, $V_{dsat}$ is close to the saturation voltage in a long-channel device. Thus, we consider

a device long-channel device when it satisfies the following equation [18],

$$\frac{V_{gs}-V_{Th}}{E_c L_{eff}} \ll 1$$

which leads to

$$\kappa \approx \frac{V_{gs}-V_{Th}}{E_c L_{eff}}$$

The regime for the so called "long-channel" length can now be quantified by setting $\frac{(V_{gs}-V_{Th})}{E_c L_{eff}} \leq 0.1$. For example, for long-channel NMOS devices where $\mu_{eff} \approx 530 cm^2/V \cdot s$, $v_{esat} = 10^5 m/s$, and $V_{gs}-V_{Th} = 1$, $L_{eff}$ should be greater than 2.65 $\mu m$.

### 3.3 Large-Signal Model for Short-Channel MOS devices

Taking into consideration the short-channel effects discussed above, we use a large signal model for MOS devices presented in [19]:

$$I_D = \begin{cases} \dfrac{\mu_{eff} C_{ox} W_{eff}}{L_{eff}} \dfrac{1}{1+\dfrac{V_{ds}}{E_c L_{eff}}} \left(V_{gs}-V_{Th}-\frac{1}{2}V_{ds}\right) V_{ds} & for\ V_{ds} \leq V_{dsat} \\ \kappa v_{esat} C_{ox} W_{eff} (V_{gs}-V_{Th}) & for\ V_{ds} \geq V_{dsat} \end{cases} \tag{3.8}$$

where $C_{ox}$ is the gate capacitance per unit area, $W_{eff}$ and $L_{eff}$ are the effective channel width and length respectively and all the other parameters are the same as in previous sections.

In order to extract the semi-empirical parameters such as $v_{esat}$ and $A$, we perform model fitting to SPICE for the *0.6* $\mu m$ channel-length device using CMOSIS5 processing technology. Fig.3.5 and Fig.3.6 show the fitting curves of the large signal model to SPICE.

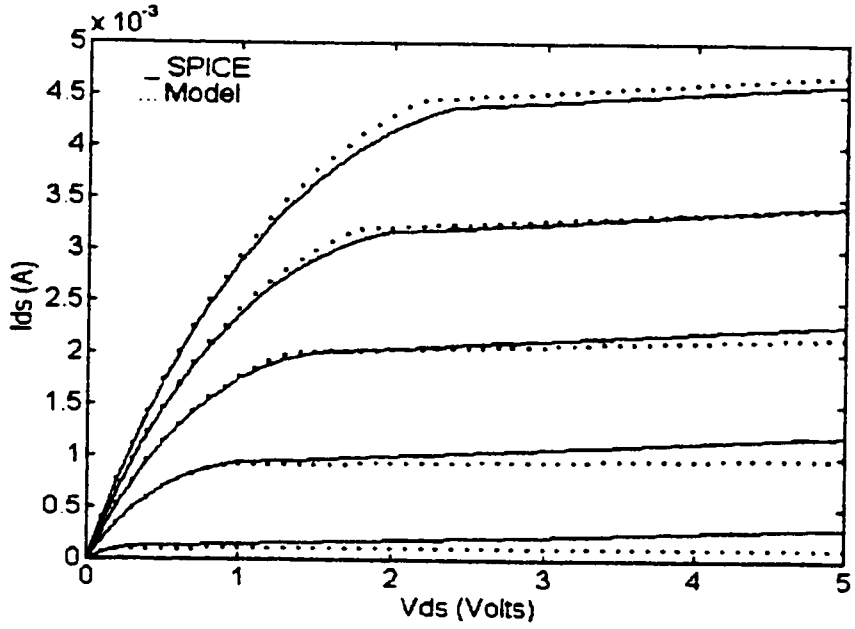


**Fig. 3.5: Transistor DC model fitting to SPICE for NMOS device with $L = 0.6um$.**

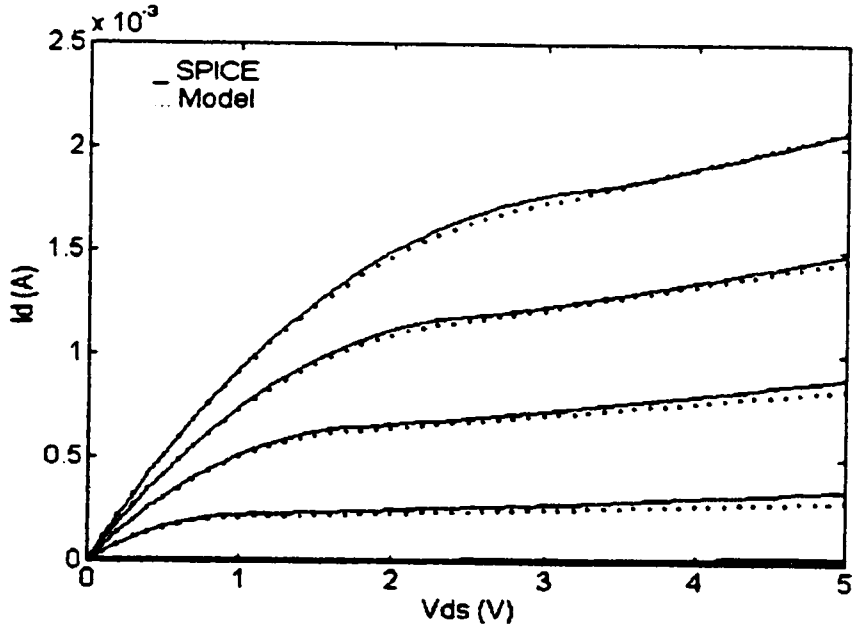


**Fig. 3.6: Transistor DC model fitting to SPICE for PMOS device with $L = 0.6um$.**

A good agreement is seen between the DC model and SPICE simulation results when

$v_{esat} = 10^5 m/s$ for both NMOS and PMOS devices and $A \approx 0.35 \times 10^8$. Channel modulation factors $\lambda_n = 0.02$ and $\lambda_p = 0.1$ are used for NMOS and PMOS devices respectively.

### 3.4 Model Description

As presented in SECTION 2.3, in order to estimate the delay of a complex CMOS gate, the pull-down and pull-up are collapsed to series-connected MOSFET structures shown in Fig.3.7. Assume that the output in Fig.3.7 is switched from high to low by a trigger input and all the other input are kept high. We also assume that the trigger is a step input. The effect of a ramp input will be discussed later. Before the switching of the trigger transistor, all the node capacitances above the trigger are charged to $V_{DD} - V_T$ and all the node capacitances below the trigger are discharged to ground. When the trigger transistor is switched on, $N_1$ starts in the saturation mode, all transistors from $N_2$ to $N_n$ are in the linear region.

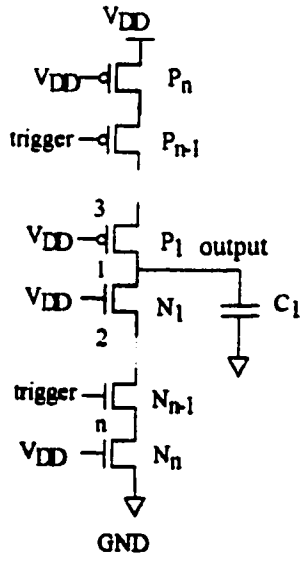


**Fig. 3.7: Series-connected pull-down and pull-up networks.**

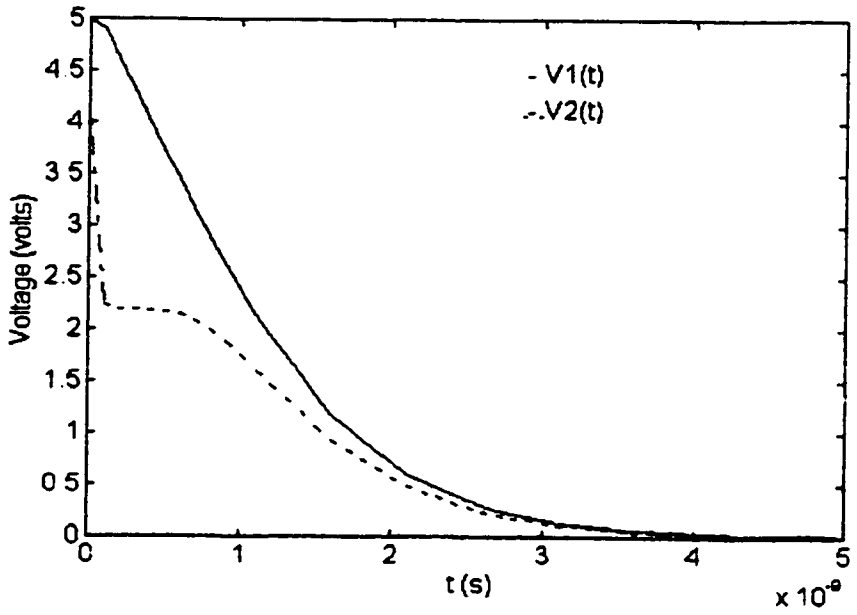


**Fig. 3.8: Voltage responses at the node 1 and 2 in pull-down network in Fig.3.7.**

Fig.3.8 shows two critical waveforms at nodes 1 and 2 in the pull-down network in Fig.3.7. It can be seen that when the trigger input switches from low to high, there is a quick initial charge distribution. $V_2(t)$ quickly reaches to a plateau value and stays there until $V_1(t)$ gets out of its saturation. It then follows $V_1(t)$ to ground. The time that $V_2(t)$ stays in its plateau value depends on the capacitance at the output node. Symmetric analysis also applies to the pull-up network. The voltage response of nodes 1 and 3 in Fig.3.7 in the pull-up network is illustrated in Fig.3.9.

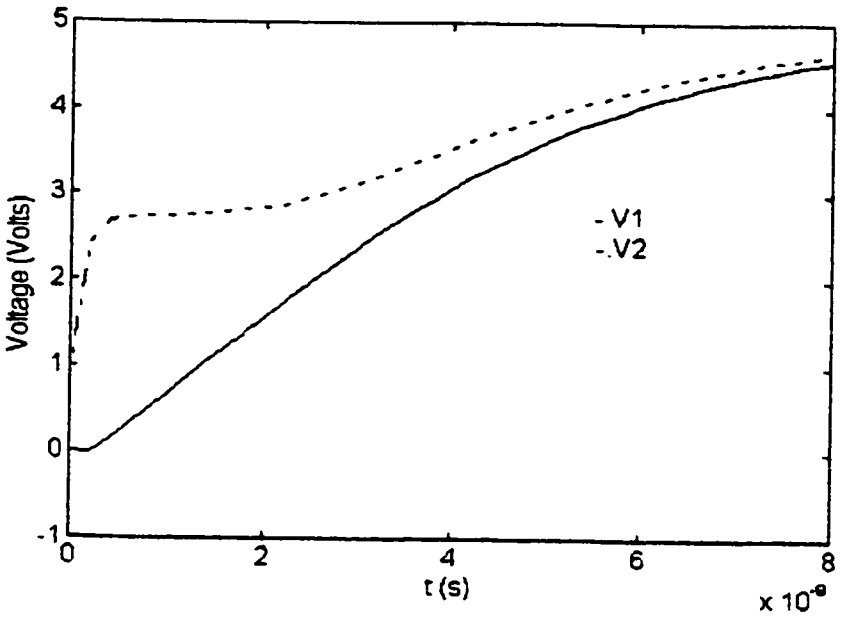


**Fig. 3.9: Voltage at node 1 and 3 for the pull-up network in Fig.3.7.**

As shown in Fig.3.8 and Fig.3.9, it is clear that we can model the charging or discharging waveform at node 1 by a piece-wise linear function of time with up to three segments after $V_3(t)$ or $V_2(t)$ reaches its plateau value. Fig.3.10 shows the three segments that model the output voltage of node 1.

Segment I represents the period where all the internal nodes $N_2$ to $N_n$ have been charged or discharged to their plateau value. It takes into account the time when the internal nodes settle down at their plateau value while $N_1$ is still in saturation region. Segment II is the period when $N_1$ gets out its saturation region and $V_1(t)$ is greater than zero. All the transistors operate in linear region in this segment. In segment 3, $V_1(t)$ is discharged to ground with all the other internal nodes.

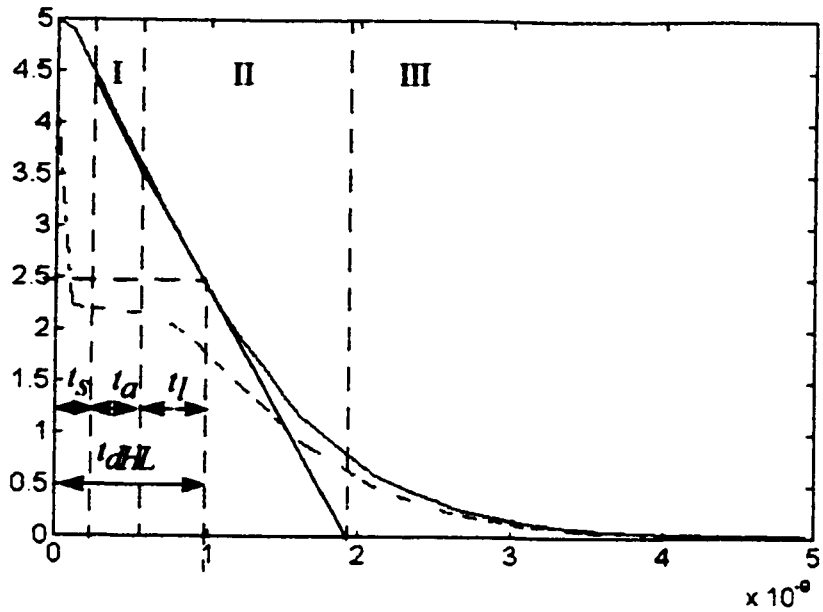


**Fig. 3.10: A piecewise model for output.**

Using the three segments analyzed above, we can divide the propagation delay into three components: $t_s$ - the time for $V_2$ to reach its plateau value, $t_a$ - the time between $t_s$ and the time when the output gets out of its saturation, and $t_l$ - the period between the time when the output gets out of its saturation and the time it drops to $0.5V_{DD}$.

From the above analysis, it is clear that we can replace the transistors from node $N_2$ to $N_n$ by an equivalent transistor $N_{equ}$, due to the fact that they all operate in linear region. The pull-down network can thus be collapsed into two series-connected transistors. Using the same principle, pull-up can be collapsed into two series-connected PMOS transistors as well as shown in Fig.3.11.

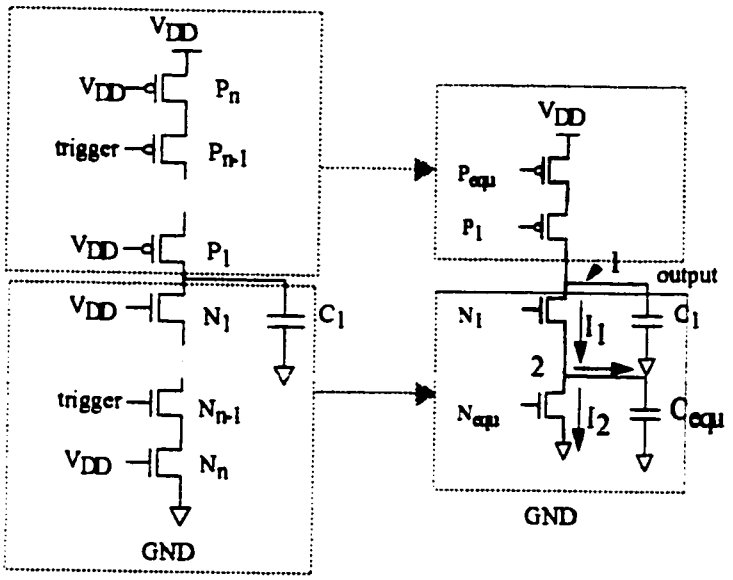


**Fig. 3.11: Simplified and equivalent circuit.**

## 3.5 Circuit Analysis for Step Input

In this section, an analytical delay model is proposed for a series-connected MOSFET structure in short-channel region. Similar approach has been used in [11] to estimate the delay in domino-logic circuits for long-channel devices.

As can be seen from Fig.3.10, there is a plateau region of $V_2(t)$ in segment I where the topmost transistor is in saturation while the rest are in linear region. The transistors from $N_2$ to $N_n$ can thus be approximated by an equivalent transistor $N_{equ}$ as shown in Fig.3.11 with its equivalent channel length $n$-$1$ times of $N_2$. The node capacitance $C_{equ}$ is the lumped diffusion capacitance for transistors $N_2$ to $N_n$. When lumping the node capacitance, the equivalent circuit must have the same charge as the original circuit, i.e.

$$\sum_{2}^{n} C_i V_i(0) = C_{equ} V_2(0)$$

Thus, the initial voltage at node 2 can be set at,

$$V_2(0) = \frac{\sum_{2}^{n} C_i V_i(0)}{C_{equ}} \tag{3.9}$$

To simplify the analysis, we only consider the case that output goes from high to low. The low to high case will be symmetrical. Assume that the inputs are step inputs. The ramp input effect will be discussed later.

As explained earlier, $N_1$ is in saturation when the transistors in Fig.3.11 are switched on. The time $N_1$ stays in saturation depends on the discharging speed of the output node capacitance. Depending on the value of output capacitance, there may exist two scenarios:

Case 1 - $N_1$ is still in saturation when the internal nodes (node 2 to n, Fig.3.7) reach their plateau value.

Case 2 - $N_1$ gets out of the saturation region before the internal nodes reach their plateau value.

### 3.5.1 Case 1 Study - N1 stays in saturation until all internal nodes reach their plateau value

Fig.3.10 shows a typical voltage response for this case, where $N_1$ stays in saturation when the internal nodes reach their plateau value. It generally applies to circuits with large load capacitances compared to the internal node capacitances. The output voltage waveform can be modeled in three segments.

Segment I: $N_1$ is in its saturation region when the internal nodes settle to their plateau voltage. The duration of the segment is $t_a$

Segment II: All transistors are in their linear region and $V_1(t)$ is greater than zero.

Segment III: $V_1(t)$ becomes zero.

We will analyze the three segments in detail to show how the delay model is derived.

### 3.5.1.1 Segment I: Calculation of Plateau Value $V_{2ss}$ and $t_S$

Consider the simplified circuit in Fig.3.11. Segment I starts at the time when the internal node 2 reaches its plateau value and ends when $N_1$ is out of saturation region. We will calculate the plateau value $V_{2ss}$, and the starting point of segment I - $t_S$.

As discussed in SECTION 3.3, $\kappa$ may be approximated by a constant for devices with fixed channel length. $\mu_{eff}$, $E_c$ for $N_{equ}$ may also be considered close to constants when $V_{gs} = V_{DD}$ for a step input. Hence, we may simplify (3.8) into (3.10),

$$I_D = \begin{cases} K_2 \dfrac{1}{1 + \dfrac{V_{ds}}{K_3}} \left( V_{gs} - V_{Th} - \frac{1}{2} V_{ds} \right) V_{ds} & for \ V_{ds} \le V_{dsat} \\ K_1 (V_{gs} - V_{Th}) & for \ V_{ds} \ge V_{dsat} \end{cases} \tag{3.10}$$

where

$$K_2 = \frac{\mu_{eff} C_{ox} W_{eff}}{L_{eff}} \tag{3.11}$$

$$K_1 = \kappa v_{esat} C_{ox} W_{eff} \tag{3.12}$$

$$K_3 = E_c L_{eff} \tag{3.13}$$

- **Plateau value $V_{2ss}$**

When node voltage $V_2(t)$ is at its plateau value, no current flows in or out of $C_{eq2}$ we can write

$$I_{d1} = I_{d2}$$

Replacing $I_{d1}$ and $I_{d2}$ by their expression in (3.10), we have the following equation,

$$K_1(V_e - V_2(t)) = \frac{K_2}{1 + \frac{V_2(t)}{K_3}}\left(V_c - \frac{V_2(t)}{2}\right)V_2(t) \tag{3.14}$$

Solving (3.14) for $V_{2ss}$ we obtain,

$$V_{2ss} = (-K_1 V_e + K_1K_3 + K_2K_3V_c \pm (K_1^2V_e^2 + 2K_1^2K_3V_e - 2K_1K_2K_3V_cV_e + K_1^2K_3^2 \tag{3.15}$$
$$+ 2K_1K_2K_3^2V_c + K_2^2K_3^2V_c^2 - 2K_1K_2K_3^2V_e)^{\frac{1}{2}}/(-2K_1 + K_2K_3)$$

**- $t_S$ - the time when $V_2(t)$ reaches its plateau value**

When $N_1$ is in its saturation and $N_{equ}$ is in linear region, we have the following equations

$$C_{equ}\frac{dV_2}{dt} = I_{d1} - I_{d2} \tag{3.16}$$

$$C_{equ}\frac{dV_2}{dt} = K_1(V_e - V_2) - K_2\frac{1}{1 + \frac{V_2}{K_3}}\left(V_c - \frac{1}{2}V_2\right)V_2 \tag{3.17}$$

Modifying (3.17), we have

$$dt = \frac{C_{equ}(K_3 + V_2)}{aV_2^2 + bV_2 + c}dV_2 = \frac{\frac{C_{equ}}{a}(K_3 + V_2)}{(V_2 - V_{2ss1})(V_2 - V_{2ss2})}dV_2 \tag{3.18}$$

where

$$a = \frac{1}{2}K_1K_3 - K_1$$

$$b = K_1V_e - K_2K_3V_c - K_1K_3$$

$$c = K_1K_3V_e$$

$$V_{2ss1}, V_{2ss2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Since $V_2(t)$ cannot reach $V_{2ss}$ within a finite time, we choose the $0.95V_{2ss}$ level to calculate $t_s$. Integrating (3.18) with $t$ from 0 to $t_s$, $V_2(t)$ from $V_2(0)$ to $0.95V_{2ss}$, we obtain

$$t_s = \frac{C_{equ}}{a(V_{2ss1} - V_{2ss2})}\left[(K_3 + V_{2ss1}) \ln\left|\frac{0.95V_{2ss} - V_{2ss1}}{V_2(0) - V_{2ss1}}\right|\right. \tag{3.19}$$
$$\left. - (K_3 + V_{2ss2}) \ln\left|\frac{0.95V_{2ss} - V_{2ss2}}{V_2(0) - V_{2ss2}}\right|\right]$$

**- Calculation of $V_1(t_s)$**

$V_1(t_s)$ is the voltage of node 1 when node 2 settles to its plateau voltage. In order to determine $V_1(t_s)$, we examine the following equations,

$$C_{equ}\frac{dV_2}{dt} = I_{d1} - I_{d2} = K_1(V_e - V_2) - K_2\frac{1}{1 + \frac{V_2}{K_3}}\left(V_c - \frac{1}{2}V_2\right)V_2 \tag{3.20}$$

$$C_1\frac{dV_1}{dt} = -I_{d1} = -K_1(V_e - V_2) \tag{3.21}$$

Dividing (3.21) by (3.20), we get,

$$\frac{C_1}{C_{equ}}\frac{dV_1}{dV_2} = \frac{-K_1(V_e - V_2)}{K_1(V_e - V_2) - \frac{K_2K_3}{K_3 + V_2}\left(V_c - \frac{1}{2}V_2\right)V_2} \tag{3.22}$$

Solving the equation (3.22), we obtain,

$$V_1(t_s) = A\left[B_1 \ln\left|\frac{0.95V_{2ss} - V_{2ss1}}{V_2(0) - V_{2ss1}}\right| + B_2 \ln\left|\frac{0.95V_{2ss} - V_{2ss2}}{V_2(0) - V_{2ss2}}\right| - 0.95V_{2ss} + V_2(0)\right] + V_{DD} \quad (3.23)$$

where

$$A = \frac{-K_1 C_{equ}}{C_1 a}$$

$$B_1 = \frac{V_e K_3 - (K_3 - V_e + V_{2ss1}) V_{2ss1}}{V_{2ss1} - V_{2ss2}}$$

$$B_2 = \frac{V_e K_3 - (K_3 - V_e + V_{2ss2}) V_{2ss2}}{V_{2ss2} - V_{2ss1}}$$

**- Slope of $V_1(t)$**

Since the internal nodes settle to their plateau value in segment I, $V_2(t)$ remains almost constant and transistor $N_1$ acts actually as a current source. The voltage at node 1 discharges at a constant rate of $\frac{dV_1(t)}{dt}$. $V_1(t)$ at segment I can thus be modeled by a linear function of time. The slope of $V_1(t)$ in segment I may be calculated as

$$m_I = \left|\frac{dV_1(t)}{dt}\right| = \frac{K_1(V_e - V_{2ss})}{C_1} \quad (3.24)$$

### 3.5.1.2 Segment II: $N_1$ is out of saturation and $V_1(t)$ is greater than zero

In segment II, $N_1$ is out of its saturation region and all the transistors operate in linear region discharging towards ground. We will calculate the saturation voltage of $V_1(t)$, and the discharging rate of $V_1(t)$.

**- $V_s$ - value of $V_1(t)$ when $N_1$ gets out of saturation**

As discussed in SECTION 3.2.2, the saturation voltage for $N_1$ is,

$$V_{dsat} = (1-\kappa)(V_{gs}-V_{Th})$$

We denote $V_S$ as the voltage of $V_1(t)$ that separates segment I and II. When $N_1$ is getting out of its saturation region, we have,

$$V_{ds} = V_s - V_2 = V_{dsat}$$

It leads to

$$V_s = V_2 + (1-\kappa)(V_{gs}-V_{Th}) = (1-\kappa)(V_{DD}-V_{Th}) + \kappa V_2 \tag{3.25}$$

We take $V_2 \approx V_{2ss}$, and obtain

$$V_s = (1-\kappa)(V_{DD}-V_{Th}) + \kappa V_{2ss} \tag{3.26}$$

We may model $V_1(t)$ in this segment as a liner function of time with slope $m_{II}$ as,

$$m_{II} = \left|\frac{dV_1(t)}{dt}\right| = \frac{K_2}{C_1}\left(\frac{1}{1+\frac{V_{ds}}{K_3}}\right)\left(V_c - V_{2ss} - \frac{1}{2}V_{ds}\right)V_{ds} \tag{3.27}$$

$K_2$ and $K_3$ can be obtained from (3.11) and (3.13) respectively by setting $V_{gs} = V_{DD} - V_{2ss}$. At the beginning of segment II, the voltage across the drain-source of each transistor can be approximated by $V_{ds} = \frac{V_{2ss}}{n-1}$, where $n$ is the number of transistors in the series-connected pull-down network, The voltage across $N_1$ may be approximated by $V_{ds} \approx V_s - V_{2ss}$. We thus take $V_{ds}$ in (3.27) as,

$$V_{ds} = \frac{1}{2}\left(\frac{V_{2ss}}{n-1} + V_s - V_{2ss}\right) \tag{3.28}$$

### 3.5.1.3 Expression of $V_1(t)$ and Propagation Delay

As shown in Fig.3.10, $V_1(t)$ can be modeled by three piece-wise linear segments beyond the

time point $t_S$:

$$V_1(t) = \begin{cases} V_1(t_s) - m_I t & for\ t_s \le t < t_s + t_a \quad (Segment\ I) \\ V_s - m_{II} t & for\ t_s + t_a \le t < t_s + t_l \quad (Segment\ II) \\ 0 & for\ t_s + t_a + t_l \le t \quad (Segment\ III) \end{cases} \tag{3.29}$$

where $V_1(t_S)$ is the output node voltage at time $t_S$, $V_S$ the saturation voltage of the output, and $m_I$ $m_{II}$ are the slopes of $V_1(t)$ in segment I and segment II respectively.

With the output waveform of $V_1(t)$ known, we can calculate the propagation delay of a CMOS gate. Propagation delay is denoted as the time difference between input transition and the 50% output level. In our step input case, it is the time when output reaches $\frac{1}{2}V_{DD}$ and can be calculated as

$$t_{dHL} = t_s + \frac{V_s - V_1(t_s)}{m_I} + \frac{0.5V_{DD} - V_s}{m_{II}} \tag{3.30}$$

#### 3.5.1.4 Validity of Case 1 Model

In order to use the analytical model of case 1, all internal nodes should settle to a constant voltage while $N_1$ is still in the saturation region. This condition is satisfied when

$$V_1(t_s) \ge V_s \tag{3.31}$$

where $V_S$ is the saturation voltage of $V_1(t)$ as shown in SECTION 3.5.1.2, and $V_1(t_S)$ is the voltage of output node evaluated at the time when the internal nodes reach their plateau value.

### 3.5.2 Case 2 Study: $N_1$ out of saturation before the internal nodes settle to their plateau value

With reference to Fig.3.11, in case 2, the top transistor $N_1$ leaves the saturation region before the internal nodes could reach their plateau value. This case applies to circuits with relatively

small load capacitances compared to the internal node capacitances. Fig.3.12 shows a typical case 2 circuit, where segment I has been reduced to zero. The top transistor $N_1$ leaves the saturation region when $V_1(t)$ drops below its saturation voltage, and all the transistors are in the linear region. Hence the output response in case 2 may be modeled by only two segments: segment II - when all the transistors are in linear region and $V_1(t)$ is greater than zero, and segment III - the output node of $N_1$ is discharged to zero.

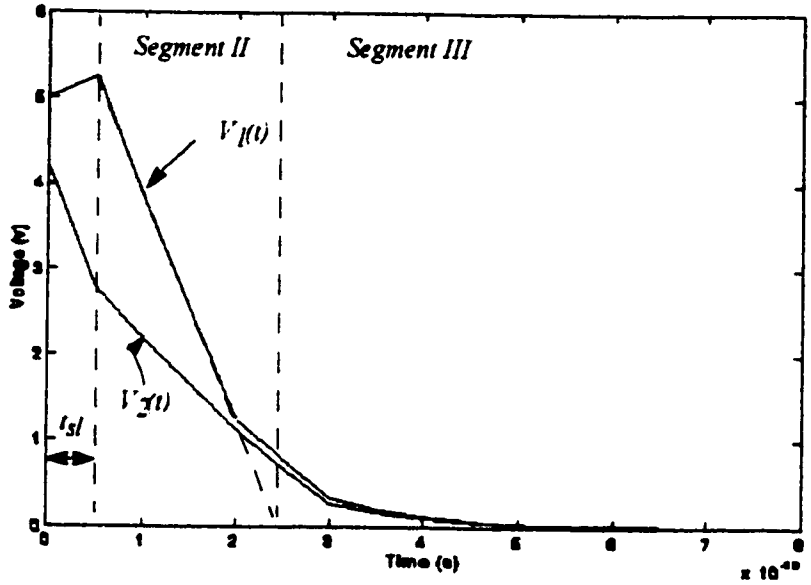


**Fig. 3.12: Output voltage of case 2.**

**- Segment II: $t_{s1}$ - starting point of Segment II**

In case 2, segment II starts at $t_{s1}$ - time when $N_1$ gets out of its saturation mode, and node 2 in Fig.3.11 has not yet reached its plateau value. $V_1(t)$ is given by (3.23) at $t = t_{s1}$,

$$V_1(t_{s1}) = A\left[B_1 \ln\left|\frac{V_2(t_{s1}) - V_{2ss1}}{V_2(0) - V_{2ss1}}\right| + B_2 \ln\left|\frac{V_2(t_{s1}) - V_{2ss2}}{V_2(0) - V_{2ss2}}\right| - V_2(t_{s1}) + V_2(0)\right] + V_{DD} = V_s \tag{3.32}$$

However, (3.32) can not lead to a closed-form solution for $t_{sl}$. From the earlier discussion in SECTION 3.5.1, we may determine $t_{sl}$ from (3.19) as

$$t_{sl} = \frac{C_{equ}}{a(V_{2ss1} - V_{2ss2})}\left[(K_3 + V_{2ss1})\ln\left|\frac{V_2(t_{sl}) - V_{2ss1}}{V_2(0) - V_{2ss1}}\right|\right. \tag{3.33}$$
$$\left. - (K_3 + V_{2ss2})\ln\left|\frac{V_2(t_{sl}) - V_{2ss2}}{V_2(0) - V_{2ss2}}\right|\right]$$

Observing that $V_2(t) \geq V_{2ss}$ for $t \leq t_s$ as shown in Fig.3.8, and knowing that $t_{sl}$ in case 2 is smaller than $t_s$ in case 1, we may obtain $V_2(t_{sl})$ as

$$V_2(t_{s1}) = V_{2ss} + V_{step} \tag{3.34}$$

where $V_{step} = \frac{V_{2ss}}{10}$. Substitute (3.34) in (3.32) and check if the condition

$$V_1(t_{s1}) \geq V_s \tag{3.35}$$

is satisfied. If not, take $V_{2ss} = V_2(t_{s1})$, and repeat (3.34) until (3.35) is satisfied. $t_{sl}$ can be obtained from (3.33). The maximum number of iterations is 10.

### 3.5.3 Summary of Delay Analysis Procedure for Step Input

From the above discussion, we can see that the delay analysis for step input may be classified into two cases: case 1 - $N_1$ in saturation while the other internal nodes settle to their plateau value, and case 2 - $N_1$ out of saturation before the internal nodes settle to their plateau value. The delay analysis procedures for case 1 and case 2 are summarized in Fig.3.13.

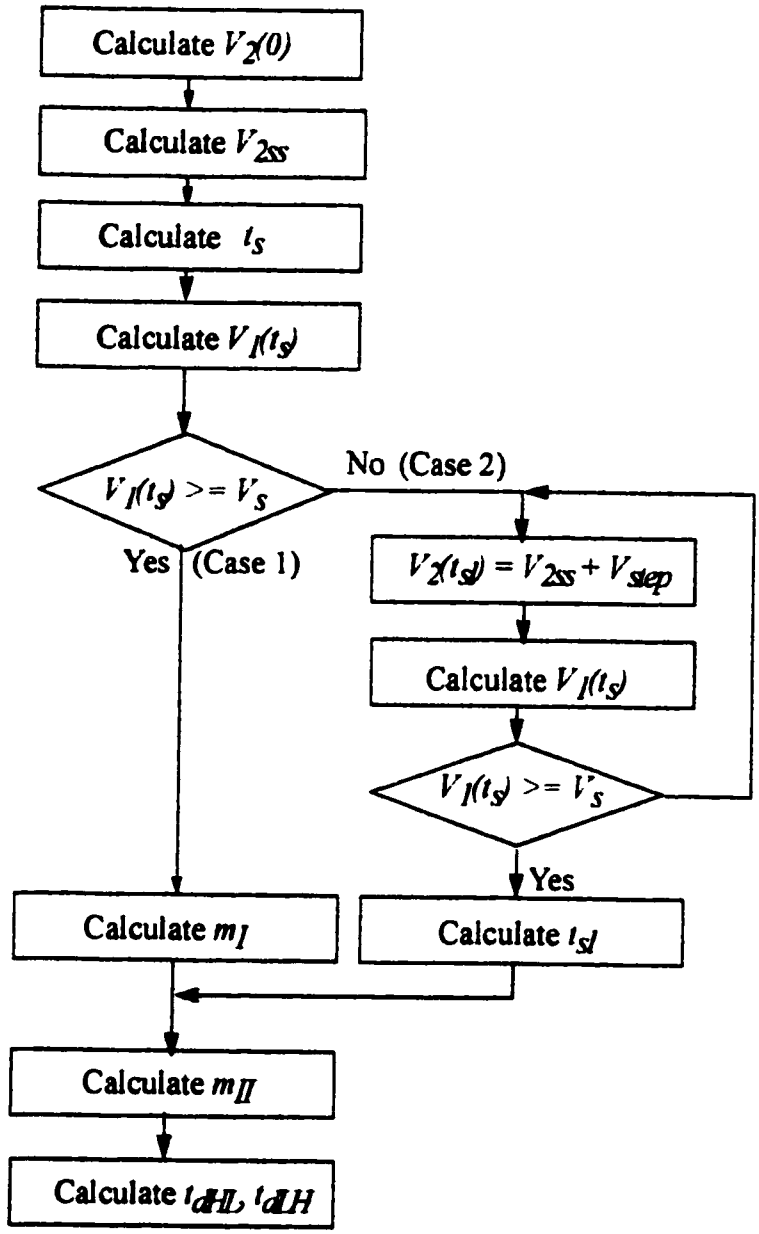


**Fig. 3.13: Flow-chart of the analytical delay model for step input.**

## 3.6 Delay Analysis for Ramp Input

The slope of an input waveform may account for up to 30 percent of the gate delay [1]. When the input rises or falls rapidly, the delay of the charge or discharge path is determined by the rate at which the transistors in the path can charge or discharge the capacitors in the tree [15]. When

the input changes slowly, it will contribute to the output delay. Four output waveforms correspond to input signals with different rise time from 10 to 40*ns* are shown in Fig.3.14.

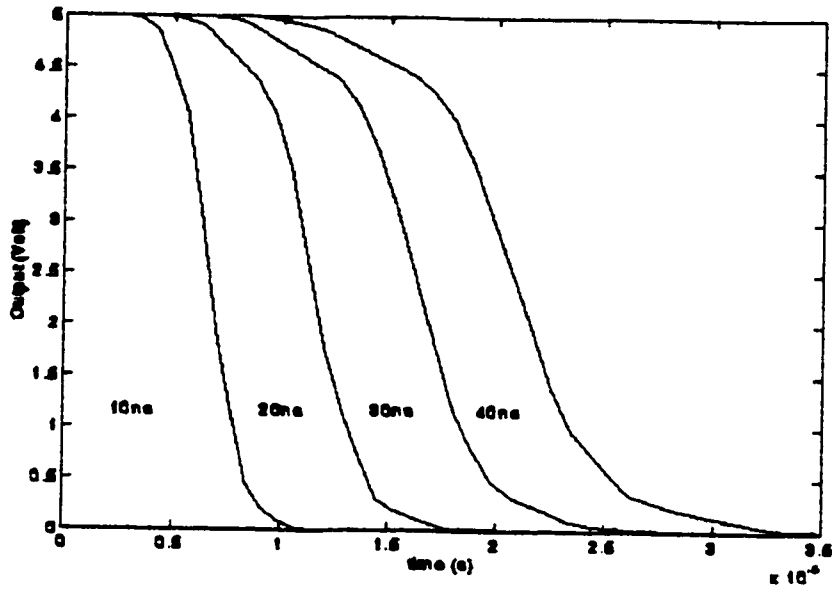


**Fig. 3.14: Output wave-forms with different input signal rise time.**

As illustrated in [11], the input slope effect may be accounted by Fig.3.15. When the input voltage is a ramp signal with rise time $t_r$, the device is completely shut off until the input voltage rise to the threshold voltage $V_{Th}$. The device will draw full-scale current $I_f$ when the input rises to $V_{DD}$. In between, the current may be modeled by a function of time as shown in Fig.3.15. For a circuit with step inputs, the total charge $Q_t$ discharged to ground in the time period $t_d$ can be expressed as $Q_t = I_f \times t_d$. For the same circuit with ramp inputs, the same amount of charge $Q_t$ still needs to be discharged to ground, however, with a longer time period $t_d'$. Assume that the amount of charge $Q_r$ is discharged during the input rise time.

$$Q_r = A_r I_f (t_r - t_{th}) \tag{3.36}$$

where $t_{th} = \frac{V_{Th}}{V_{DD}} t_r$ and $A_r$ is an empirical factor to determine the amount of charge discharged

to ground for the period of $t_p$. For example, as shown in Fig.3.15 (d), the area under the three curves represents the amount of charge discharged to ground during $t_p$. Curve 1 shows a linear relationship between current and time, where $A_r = 0.5$. $A_r > 0.5$ and $A_r < 0.5$ are for the non-linear curve 2 and 3 respectively.

As show in Fig.3.15 (d), we have

$$Q_t = I_f \times t_d = Q_r + I_f(t_d' - t_r) \tag{3.37}$$

Substituting (3.36) in (3.37), we have

$$t_d' = t_d + (1 - A_r)\, t_r + A_r \frac{V_{Th}}{V_{DD}} t_r \tag{3.38}$$

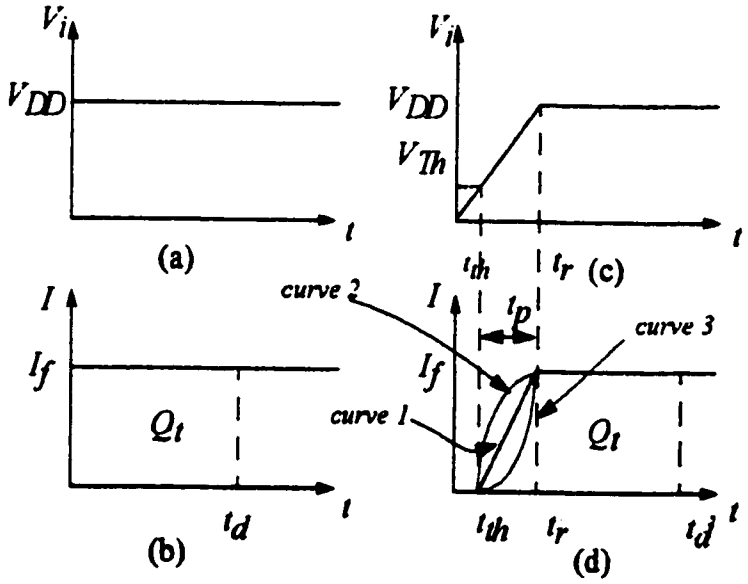


**Fig. 3.15: Input Signals and their corresponding current waveforms.**

## 3.7 Delay Analysis for Inverter

We have performed the circuit analysis for a CMOS gate with series-connected MOSFETs

structure in previous sections. Using the short-channel DC model in (3.8), delay formula for a CMOS inverter can be derived. We analyze the case when output goes from high to low. The case of low to high is symmetrical.

An approximation of input waveform is used in analyzing a CMOS inverter delay in [10] to simplify the calculation. We will use similar approximation method in our analysis.

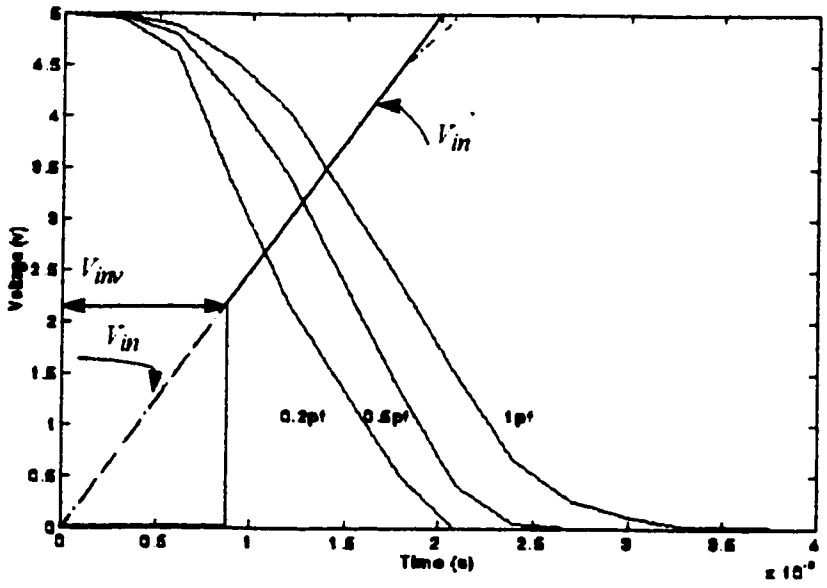


**Fig. 3.16: Approximating input waveform.**

The input waveform approximation method is shown in Fig.3.16, where the ramp input waveform $V_{in}$ is approximated by $V_{in}'$ which stays at zero voltage until the ramp input goes across the logic threshold $V_{inv}$[10]. $V_{in}'$ rises up at that point and coincides with the ramp input waveform $V_{in}$ thereafter. The logic threshold corresponds to the state where $V_{out} = \frac{1}{2}V_{DD}$.

For the very fast input, the ramp input becomes a step function and $V_{in}'$ also becomes the step function and current through PMOS can be completely ignored. For the extremely slow input, the output changes abruptly and comes down to $\frac{1}{2}V_{DD}$ when the input goes across the logic threshold voltage. The inverter with the approximated input waveform shows the same delay as that of the

original input. We analyze the situations for fast input and slow input separately.

### 3.7.1 Fast Input Case

For the fast input case, the current through PMOS transistor is negligible. We may thus approximate a CMOS inverter by a NMOS transistor as shown in Fig.3.17.

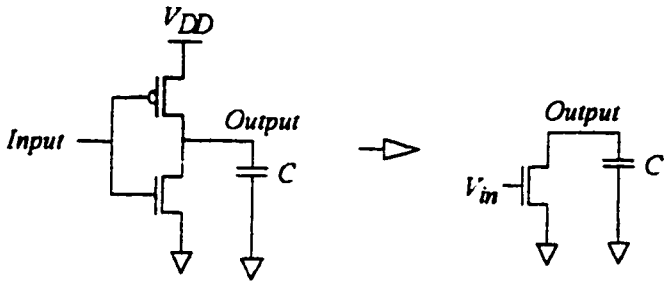


**Fig. 3.17: Equivalent circuit when output goes from high to low.**

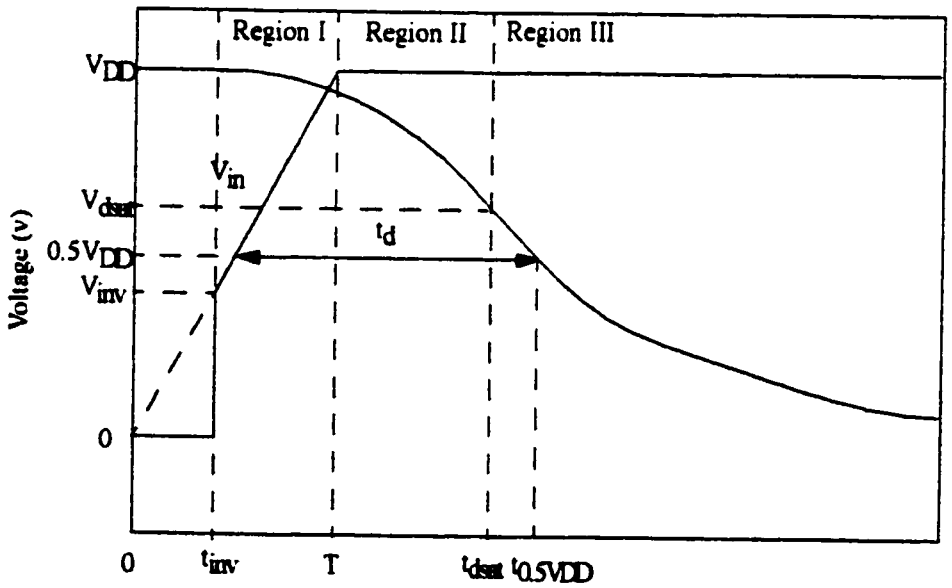


**Fig. 3.18: Input/output waveform of fast input.**

A very fast input case is shown in Fig.3.18, where the NMOS has not gone out of saturation

when the input reaches $V_{DD}$ We may therefore divide the input/output waveform into three operating regions.

As shown in Fig.3.18, region I represents the period of time from $t_{inv}$ to the time when the input reaches $V_{DD}$ $t_{inv}$ is the time when input reaches the logic threshold. The NMOS transistor operates in its saturation region during this period. After the input reaches its final value, the NMOS stays in saturation until the output drops below its saturation voltage. Region II represents the period between $T$ and $t_{dsat}$ - the time before the output reaches its saturation voltage. Region III is the time period after the output reaches the saturation voltage of the NMOS transistor. We will analysis the three operating regions of a CMOS inverter in the following sections.

### 3.7.1.1 Region I: Time before input reaches $V_{DD}$

In region I, the NMOS transistor is in saturation region and the input is a linear function of time. Neglecting the current through PMOS transistor, with reference to Fig.3.17, we have

$$C\frac{dV}{dt} = -K_1 (V_{gs} - V_T) \tag{3.39}$$

where $C$ is the load capacitance and $K_1 = \kappa v_{esat} C_{ox} W_{eff}$. $V_{gs}$ is a function of time. From (3.4), we can see that $\kappa$ and $K_1$ are also functions of time. The differential equation can be solved with the initial condition of $V = V_{DD}$ at $t = t_{inv}$. We thus have

$$V = V_{DD} - \frac{T}{CV_{DD}b^3}\left( b(v_{in} - V_{inv})\left(-a + \frac{1}{2}b(v_{in} + V_{inv} - 2V_T)\right) + a^2 \ln\frac{a + bv_{in} - V_T}{a + bV_{inv} - V_T}\right) \tag{3.40}$$

where

$$a = \frac{2L_{eff}}{\mu_0 C_{ox} W_{eff}}$$

$$b = \frac{2v_{esat}L_{eff}\theta + \mu_0}{v_{esat}C_{ox}W_{eff}}$$

$v_{in}$ and $V_{inv}$ are the input voltages at time $t$ and $t_{inv}$ respectively. $T$ is the input rise time.

### 3.7.1.2 Region II: Time before output reaches $V_{dsat}$

Input waveform reaches $V_{DD}$ in this region, thus $V_{gs} = V_{DD}$ κ and $K_1$ may therefore be considered as constants. Solving the differential equation (3.39), we get the output voltage in region II.

$$V = -\frac{K_1}{C}(V_{DD} - V_T)(t - T) + V_1 \tag{3.41}$$

where $T$ is the input rise time, $V_1$ is the output voltage in region I when $V_{gs} = V_{DD}$ At the end of region II, output drops to $V_{dsat}$ - saturation voltage of the NMOS transistor when $V_{gs} = V_{DD}$ Substituting $V_{dsat}$ into (3.41), we have

$$t_{dsat} = \frac{V_{dsat} - V_1}{-\frac{K_1}{C}(V_{DD} - V_T)} + T \tag{3.42}$$

### 3.7.1.3 Region III: Time after output drops to $V_{dsat}$

In region III, the NMOS transistor of the inverter is operating in the linear region. The differential equation at the output node is,

$$C\frac{dV}{dt} = -\frac{K_2}{1 + \frac{V}{K_3}}\left(V_{gs} - V_T - \frac{1}{2}V\right)V \tag{3.43}$$

Solving (3.43) with the initial condition $V = V_{dsat}$ at $t = t_{dsat}$, we have $t = f(V)$ as shown in (3.44).

$$t = f(V) = t_{dsat} + \frac{2C}{K_2 K_3}\left[-\frac{K_3}{2V_c}\ln\left(\frac{V}{V_{dsat}}\right) + \frac{K_3 + 2V_c}{2V_c}\ln\left(\frac{V - 2V_c}{V_{dsat} - V_{dsat}}\right)\right] \quad (3.44)$$

Delay $t_d$ may therefore be obtained as

$$t_d = f(V = \frac{1}{2}V_{DD}) - \frac{T}{2} \quad (3.45)$$

### 3.7.2 Slow Input Case

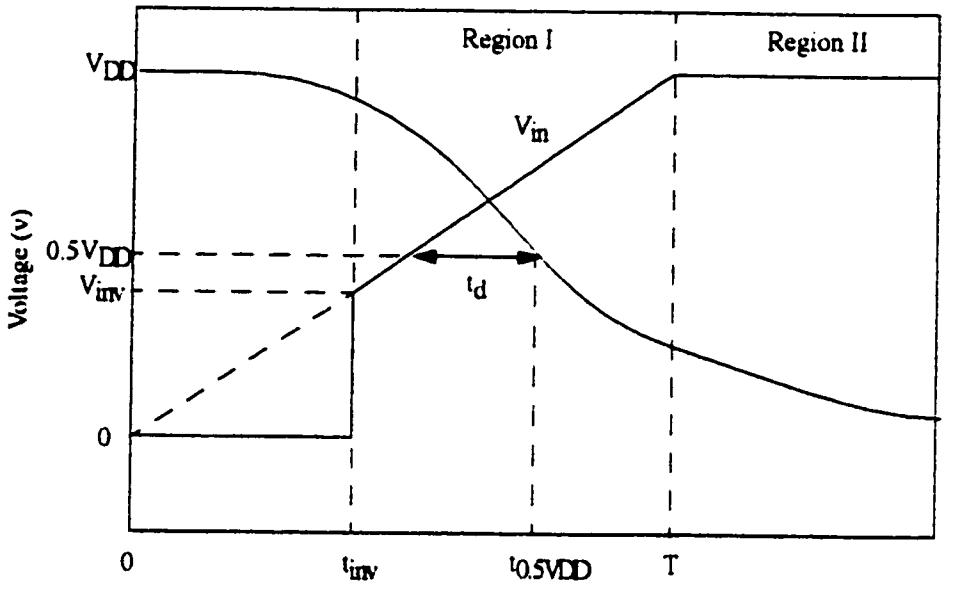


**Fig. 3.19: Input/output waveform for slow input case.**

When the input is very slow, the output reaches $\frac{1}{2}V_{DD}$ before input arrives at $V_{DD}$ Fig.3.19 shows a very slow input case, where the output crosses $\frac{1}{2}V_{DD}$ in region I. In this case, (3.39) is valid. Since the output near logic threshold changes abruptly with a small change in input, we approximate $V_{in}$ as a constant around $V_{inv}$ to simplify the analysis. If

$$V_{gs} \approx V_{inv} \tag{3.46}$$

we have

$$t_d = \frac{1}{s}\left(\sqrt{\frac{V_{DD}Cs}{K_1} + (V_{inv} - V_T)^2} + V_T\right) - \frac{T}{2} \tag{3.47}$$

where

$$s = \frac{V_{DD}}{T}$$

$$K_1 = \kappa v_{esat} C_{ox} W_{eff}$$

$$\kappa = \frac{1}{1 + \frac{E_c L_e}{V_{inv} - V_T}}$$

$$L_e = L_{eff} - \chi_d$$

$\chi_d$ is the same as in (3.7).

### 3.7.3 Case Validity

As shown in SECTION 3.7.1 and SECTION 3.7.2, for a fast input case, the input reaches its final value before the NMOS transistor gets out of its saturation region; while for a slow input case, the output crosses $\frac{1}{2}V_{DD}$ before the input reaches its final value. Thus, for an input to be considered fast input, the output voltage in region I for $V_{in} = V_{DD}$ has to satisfy the following condition,

$$V_1\big|_{V_{in} = V_{DD}} \geq \frac{1}{2}V_{DD} \tag{3.48}$$

### 3.7.4 Logic Threshold $V_{inv}$

When PMOS and NMOS transistor both are in saturation region, we have

$$I_{dn} = -I_{dp} \tag{3.49}$$

If we denote logic threshold as $V_{inv}$ and rewrite (3.49), we have

$$K_{1n}(V_{inv} - V_{Tn}) = K_{1p}(V_{DD} - V_{inv} + V_{Tp}) \tag{3.50}$$

where $K_{In}$ and $K_{Ip}$ are functions of $V_{inv}$. They are given as follows,

$$K_{1n} = \frac{v_{esatn} C_{ox} W_{effn}}{1 + \frac{E_{cn} L_{effn}}{V_{inv} - V_{Tn}}} \tag{3.51}$$

$$K_{1p} = \frac{v_{esatp} C_{ox} W_{effp}}{1 + \frac{E_{cp} L_{effp}}{V_{DD} - V_{inv} + V_{Tp}}} \tag{3.52}$$

$$E_{cn} = \frac{2v_{esatn}}{\mu_{0n}}[1 + \theta_n(V_{inv} - V_{Tn})]$$

$$E_{cp} = \frac{2v_{esatp}}{\mu_{0p}}\left[1 + \theta_p\left(V_{DD} - V_{inv} + V_{Tp}\right)\right]$$

Solving (3.50), we find that $V_{inv}$ is a root of a cubic equation shown as follows,

$$aV^3 + bV^2 + cV + d = 0 \tag{3.53}$$

where

$$a = B_n + AB_p$$

$$b = C_n - B_n(2V_c + V_{Tn}) - A[C_p + B_p(V_c + 2V_{Tn})]$$

$$c = V_c\left(V_c B_n - 2C_n + 2B_n V_{Tn}\right) + V_{Tn} A\,(B_p V_{Tn} + 2C_p + 2B_p V_c)$$

$$d = (C_n - B_n V_{Tn})\,V_c^2 - A V_{Tn}^2 (C_p + B_p V_c)$$

$$B_n = 1 + E_n L_n \theta_n, \qquad B_p = 1 + E_p L_p \theta_p$$

$$A = \frac{v_{esatn} W_n}{v_{esatp} W_p}, \qquad C_n = E_n L_n, \qquad C_p = E_p L_p$$

### 3.7.5 Summary of Delay Calculation for Inverter

The delay calculation procedures for an inverter can be summarized as follows.

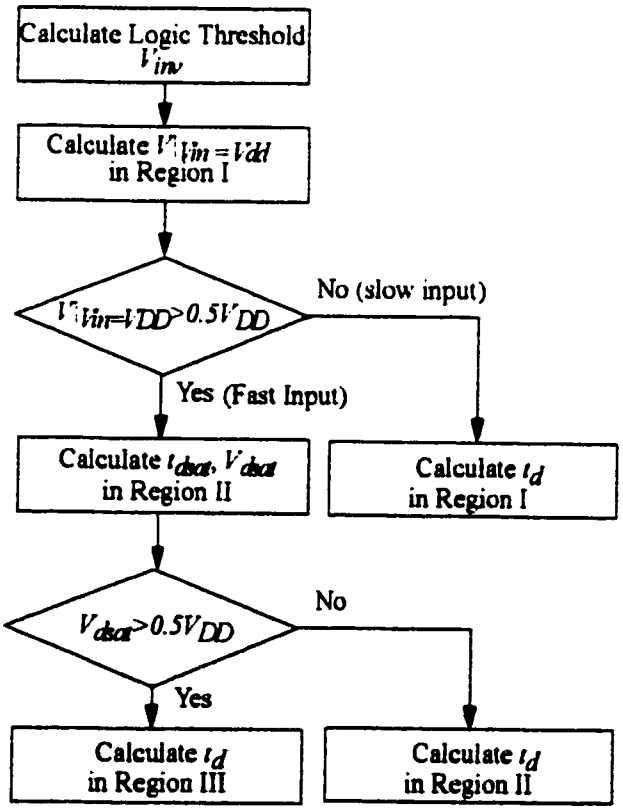


**Fig. 3.20: Delay calculation procedures for an inverter.**

### 3.8 Matching Between Ramp Response and Response of Input waveform

In CMOS data paths, the input waveform of a gate at position *n* is not a ramp input but the output response of the preceding gate *n-1*. Hence, to use the above delay analytical models for calculating gate delays in signal paths, a ramp waveform with effective rise time (or fall time) is required. The response of the effective ramp waveform has to match the response of the real input waveform.

As presented in [1], the output rise (fall) time can be approximated by its derivative at the half-$V_{DD}$ point. Fig.3.21 shows the SPICE output voltage $V_O$ of an inverter in a chain of identical inverters for typical loading conditions. $V_O$ can be characterized as a quasi-linear ramp with exponential tails. When $V_O$ is taken as an input, the details of the waveform below $V_T$ (above $V_{DD}$-$V_T$) can be disregarded since the relevant transistor in the driven gate will be off during that region.

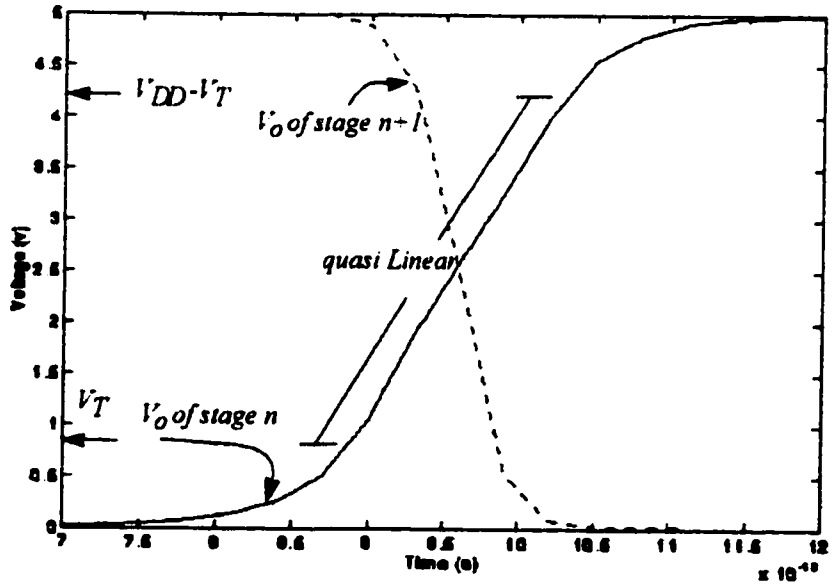


**Fig. 3.21: Output voltage of an inverter driven by an identical inverter in a chain of inverters.**

Hence, the effective signal rise time may be calculated by using the slope of $V_O$ at mid-level voltage,

$$t_{eff} = V_{DD}\left(\frac{dV_o}{dt}\right)^{-1}\Bigg|_{V_o=\frac{1}{2}V_{DD}} \tag{3.54}$$

(3.54) can be generalized to the arbitrary logic gates. As shown in SECTION 3.6, for input signals with different slope, the output waveform is preserved and the waveform displacement is linearly proportional to its input signal slope. Fig.3.22 shows two output waveforms of the same circuit. $V_1$ is the output response of a step input and $V_2$ is the output of a ramp input. They have approximately the same derivatives at $\frac{1}{2}V_{DD}$. Knowing the derivative at $\frac{1}{2}V_{DD}$ level and the delay of the step input response, the fall time of $V_2$ may be calculated as

$$t_{fall} = V_{DD}\left(\frac{dV_2}{dt}\right)^{-1}\Bigg|_{V_o=\frac{1}{2}V_{DD}}$$

with an offset,

$$t_{offset} = t_{0.5V_{DD}} - \frac{1}{2}V_{DD}\left(\frac{dV_2}{dt}\right)^{-1}\Bigg|_{V_o=\frac{1}{2}V_{DD}} \tag{3.55}$$

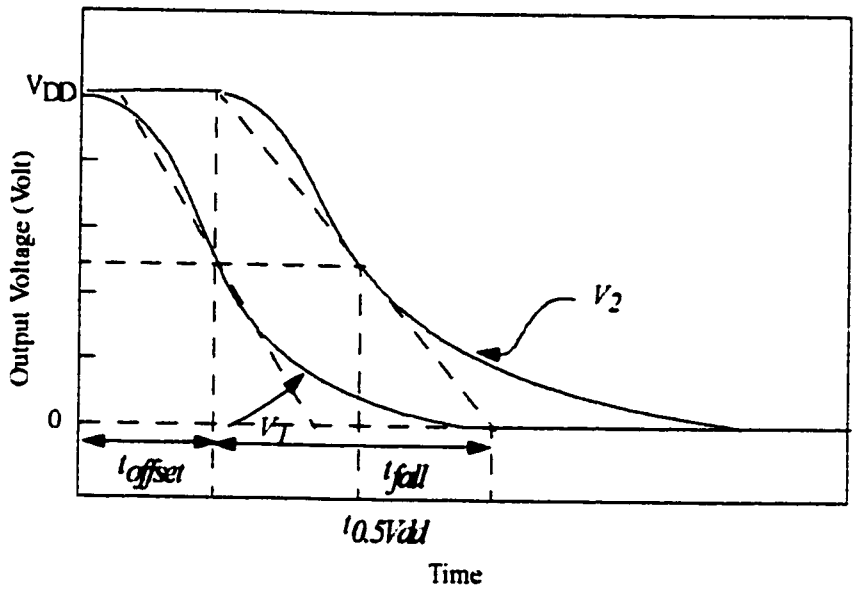


**Fig. 3.22: Effective ramp waveforms for output signals with different inputs.**

## 3.9 Conclusion

In this chapter, we have discussed the short-channel effect, presented the analytical delay models for short-channel MOSFETs for both inverter and non-inverter stages. Ramp input effect are compensated base on the step input response for non-inverter stage. The matching between the response of the effective ramp waveform and that of the real input waveform is also discussed. In the next chapter, we will discuss delay analysis for long-channel MOSFET circuits.

# CHAPTER 4 Delay Analysis for Long-Channel MOSFETs

## 4.1 Introduction

In this chapter we present the analytical delay model for long-channel MOSFETs using similar approaches as shown in CHAPTER 3. For long-channel MOSFETs, we simplify the analysis by neglecting the short-channel effects. A piece-wise linear model for the output waveform is derived for step input. The effect of ramp input is considered in a similar way as shown in CHAPTER 3.

## 4.2 Circuit Analysis for a Step Input

The analysis of series-connected long-channel devices is similar to the approach used for short-channel devices. The case studies discussed in SECTION 3.5 apply to long-channel devices. In this section, we analyze the step input response for long-channel devices using the modeling strategy presented in SECTION 3.5.

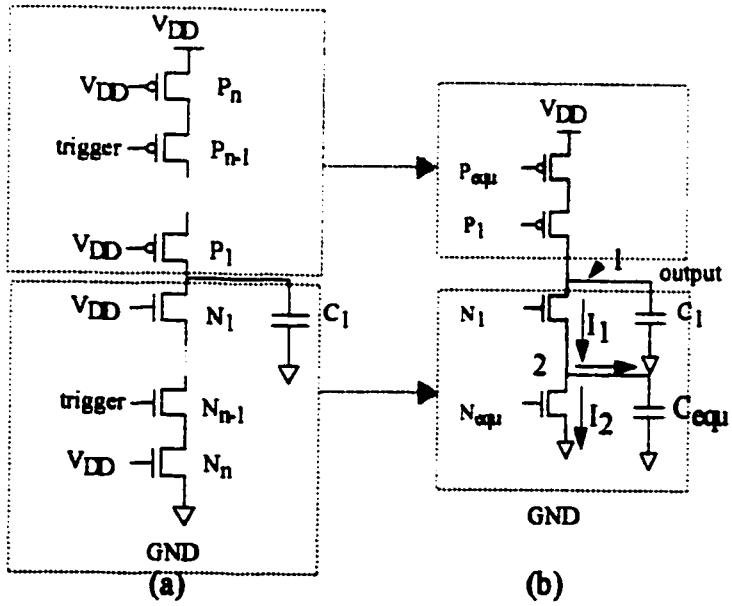


**Fig. 4.1: Simplified and equivalent circuit for delay estimation in long-channel devices.**

### 4.2.1 Case 1 Study - $N_1$ stays in saturation until all internal nodes reach their plateau value

The same modeling strategy used for short-channel analysis applies to the long-channel devices. The circuit of Fig.4.1 (a) is simplified to Fig.4.1 (b). We analyze the equivalent circuit in Fig.4.1(b) in the three segments defined in SECTION 3.5.

#### 4.2.1.1 Segment I: Calculation of Plateau Value $V_{2ss}$ and $t_s$

**- Plateau Value $V_{2ss}$**

When $N_{equ}$ reaches its plateau value, the current discharging its node capacitance is approximately equal to zero, i.e., $\frac{dV_2}{dt} \approx 0$. The currents $I_{d1}$ and $I_{d2}$ in Fig.4.1 are given by,

$$I_{d1} = \frac{\beta_1}{2}\left(V_{DD} - V_{Tnsat} - V_2(t)\right)^2 \tag{4.1}$$

$$I_{d2} = \beta_{equ}\left[\left(V_{DD} - \overline{V_{Tn}}\right)V_2(t) - \frac{1}{2}V_2(t)^2\right] \tag{4.2}$$

Since $I_{d1} \approx I_{d2}$, by equalizing (4.2) and (4.1) we get the plateau value $V_{2ss}$ at node 2. $V_{2ss}$ is given in [11],

$$V_{2ss} = \frac{\Omega - \sqrt{\Omega^2 - n(n+1)}}{n+1} \tag{4.3}$$

where $\Omega = nV_e + \left(V_{DD} - \overline{V_{Tn}}\right)$, $n = \frac{\beta_1}{\beta_{equ}}$ and $V_e = V_{DD} - V_{Tnsat}$.

**- $t_s$: time when $V_2(t)$ reaches $V_{2ss}$**

When $V_2(t)$ reaches its plateau value with $N_1$ in saturation region, we have the following equation from Fig.4.1,

$$\frac{\beta_1}{2}\left[V_{DD} - V_2(t) - V_{Tnsat}\right]^2 = C_2\frac{dV_2(t)}{dt} + \beta_{equ}\left[\left(V_{DD} + \overline{V_{Tn}}\right)V_2(t) + \frac{1}{2}V_2^2(t)\right] \tag{4.4}$$

The solution of (4.4) is given by [11] as,

$$V_2(t) = \frac{a_2 - a_1 e^{B-At}}{1 - e^{B-At}} \tag{4.5}$$

where

$$a_1, a_2 = \frac{nV_e + V_c}{n+1} \pm \sqrt{\left[\left(\frac{nV_e + V_c}{n+1}\right)^2 - \frac{nV_e^2}{n+1}\right]} = V_h \pm V_f$$

$$V_c = V_{DD} - V_{Tnsat}$$

$$V_h = \frac{nV_e + V_c}{n+1}$$

$$V_f = \sqrt{\left[\left(\frac{nV_e + V_c}{n+1}\right)^2 - \frac{nV_e^2}{n+1}\right]}$$

$$B = \ln\left(\frac{V_2(0) - a_2}{V_2(0) - a_1}\right), \qquad A = \frac{\beta_{equ}(n+1)V_f}{nC_{equ}}$$

$n$ is denoted the same as in (4.3). We can calculate $t_s$ now by resolving $t$ from (4.5). Since $V_2(t)$ cannot reach $V_{2ss}$ within a finite time, we choose the $0.95V_{2ss}$ as a level to calculate $t_s$,

$$t_s = \frac{C_{equ}}{\beta_{equ}V_f}\left(B - \ln\left|\frac{0.95V_{2ss} - a_2}{0.95V_{2ss} - a_1}\right|\right) \tag{4.6}$$

**- Calculation of $V_1(t_s)$**

$V_1(t_s)$ is the output node voltage of $N_1$ at $t_s$. Referring to Fig.4.1, we have the following equations,

$$C_1\frac{dV_1(t)}{dt} = -I_{d1} = -\frac{\beta_1}{2}[V_e - V_2(t)]^2 \tag{4.7}$$

$$C_{equ}\frac{dV_2(t)}{dt} = I_{d1} - I_{d2} = \frac{\beta_1}{2}[V_e - V_2(t)]^2 - \beta_{equ}\left(V_2(t)V_c - \frac{1}{2}V_2^2(t)\right) \tag{4.8}$$

We cannot derive a closed-form equation for $V_1(t)$ using (4.7). However, we can express $V_1(t)$ as a function of $V_2(t)$ with (4.7)/(4.8), i.e.,

$$V_1(t) = f(V_2(t)) \tag{4.9}$$

Thus, we have $V_1(t_s) = f(V_{2ss})$ by setting $t = t_s$. The derivation of $V_1(t_s)$ is given in Appendix 1.

- $m_I$: The discharging rate of $V_1(t)$ in segment I

Since node 2 settles to its plateau value $V_{2ss}$ in segment I, $V_2(t)$ is almost a constant in this segment. $N_1$ acts as a current source. The voltage at node 1 discharges at a constant rate. The current for transistor $N_1$ in segment 1 is

$$I_{d1} = \frac{\beta_1}{2}[V_e - V_{2ss}]^2 = -C_1\frac{dV_1(t)}{dt} \tag{4.10}$$

The slope of $V_1(t)$ in segment I can be calculated as,

$$m_I = \left|\frac{dV_1(t)}{dt}\right| = \frac{\beta_1(V_e - V_{2ss})^2}{2C_1} \tag{4.11}$$

#### 4.2.1.2 Segment II: All the transistors operate in linear region and $V_1(t)$ is greater than zero

At the end of segment I, when $V_1(t)$ drops to one threshold below $V_{DD}$, $N_1$ is about to get out of its saturation region and enter segment II. The voltage of $V_1(t)$ for $N_1$ to get out of saturation is,

$$V_s = V_{DD} - V_{Tn1} \tag{4.12}$$

$V_{Tn1}$ is the threshold voltage of $N_1$.

All the transistors operate in linear region in segment II and discharge towards ground. The current through $N_1$ can be expressed as

$$I_{d1} = \beta_1 \left[ (V_c - V_2(t)) V_{ds} - \frac{V_{ds}^2}{2} \right] = -C_1 \frac{dV_1(t)}{dt} \tag{4.13}$$

As discussed in SECTION 3.5.1.3 for short-channel devices, $V_1(t)$ may be modeled in this segment as a linear function of time with slope $m_{II}$ as,

$$m_{II} = \left| \frac{dV_1(t)}{dt} \right| = \frac{\beta_1}{C_1} \left[ (V_c - V_{2ss}) V_{ds} - \frac{1}{2} V_{ds}^2 \right] \tag{4.14}$$

$V_{ds}$ may be derived in the same way as shown in SECTION 3.5.1.3

#### 4.2.1.3 Expression for $V_1(t)$ and Propagation Delay

Summarizing the above analysis, beyond the time point $t_s$, $V_1(t)$ can be modeled by three piece-wise linear segments [11] as shown in Fig.3.10,

$$V_1(t) = \begin{cases} V_1(t_s) - m_I t & for\ t_s \le t < t_s + t_a \quad (Segment\ I) \\ V_s - m_{II} t & for\ t_s + t_a \le t \le t_s + t_l \quad (Segment\ II) \\ 0 & for\ t > t_s + t_a + t_l \quad (Segment\ III) \end{cases} \tag{4.15}$$

For our step input case, the high to low propagation delay may be calculated as,

$$t_{dHL} = t_s + \frac{V_s - V_1(t_s)}{m_I} + \frac{0.5V_{DD} - V_s}{m_{II}} \tag{4.16}$$

The low to high propagation delay can be derived by analyzing the pull-up network with the same method.

#### 4.2.1.4 Validity of Case 1 Model

For the analytical model of case 1 to be valid, all internal nodes should settle to a constant voltage while $N_1$ is still in the saturation region. This condition is satisfied when

$$V_1(t_s) \geq V_s \tag{4.17}$$

where $V_S$ is the saturation voltage of $V_f(t)$ and is given as

$$V_s = V_{DD} - V_{Tnsat} \tag{4.18}$$

$V_{Tnsat}$ is the threshold of $N_1$ in saturation region.

### 4.2.2 Case 2 Study: $N_1$ out of saturation before the internal nodes settle to their plateau value

As mentioned in SECTION 3.5, the time $N_1$ stays in saturation depends on the discharging speed of the load capacitance. When the output node capacitance is relatively small compared to the internal node capacitance, $N_1$ may be out of saturation before $N_{equ}$ can settle at its plateau value. Hence the output response in case 2 can be modeled by only two segments: segment II - when all the transistors operate in linear region and $V_f(t)$ is greater than zero, and segment III - when $V_f(t)$ is discharged to ground.

#### 4.2.2.1 Segment II: $t_{sl}$ - starting point of segment II

Segment 2 starts at $t_{sl}$ - time when $N_1$ is getting out of the saturation region. Thus, $V_1(t)$ at $t_{sl}$ is,

$$V_1(t_{s1}) = f(V_2(t_{sl})) = V_{DD} - V_{Tnsat} \tag{4.19}$$

However, (4.19) cannot lead to a closed-form solution of $t_{sl}$. Knowing that $t_{sl}$ is smaller than $t_s$ in case 1, we may determine $t_{sl}$ by,

$$t_{s1} = t_s - t_{step} \tag{4.20}$$

where $t_{step} = \frac{t_s}{10}$. The condition

$$V_1(t_{s1}) \geq V_{DD} - V_{Tnsat} \tag{4.21}$$

is checked for $t_{S1}$. If it is not satisfied, take $t_S$ as $t_{S1}$ and repeat the above procedures until (4.21) is satisfied. The procedure is summarized in Fig.4.2.

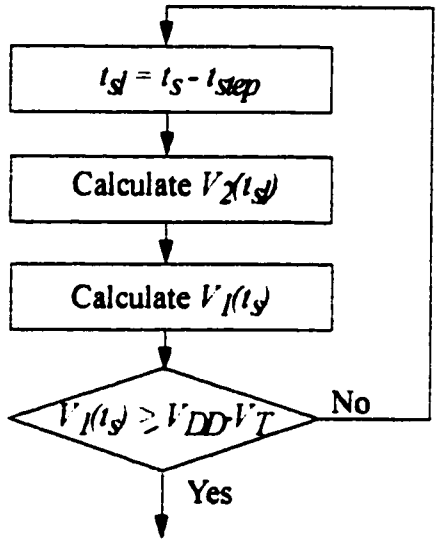


**Fig. 4.2: Case 2 flow-chart.**

The maximum number of iteration is 10 and we can achieve a good accuracy since $V_1(t)$ is a smooth function of $t$.

#### 4.2.2.2 Summary of Delay Analysis for Step Input Response

Given a collapsed equivalent long-channel circuit as shown in Fig.4.1, delay analysis for step input response can be done using the procedures described in the previous sections. The propagation delay can be calculated following the same flow-chart as shown in Fig.3.13 with a small vari-

ation in case 2.

## 4.3 Delay Analysis for Ramp Input

The modeling strategy for a ramp input is similar to that in SECTION 3.5. The delay with ramp input effect is determined by the step input response, the slope of the ramp input and the empirical parameter $A_R$ as shown in (3.38).

## 4.4 Conclusion

In this chapter, we have discussed the analytical delay model for long-channel circuits. A piece-wise linear model is derived for step input. The effect of ramp input is considered in a similar way as in short-channel circuits. In the next two chapters, we will verify the analytical delay models both for short-channel and long-channel circuits by comparing the simulation results between our model and the SPICE simulator.

# CHAPTER 5 Comparison of Delay Analytical Model with SPICE for Short-Channel MOSFETs

## 5.1 Introduction

In order to verify the accuracy of our delay model, we simulate circuits with various complexity. Simulation results are compared between the model and SPICE Level 3 simulator.

The delay of a CMOS circuit depends on circuit parameters such as the load capacitance, the transistor switched, transistor size, node capacitance and input rise time. The switching speed of a CMOS gate is limited by the time taken to charge and discharge the load capacitance. The position of the trigger transistor, on the other hand, affects the initial charge stored in the circuit. By varying these parameters, we can quantify the difference between our model and SPICE, thus prove the validity of the model.

In SECTION 5.2, output waveform comparisons between SPICE and the model on NAND gates are presented. Delay comparisons are performed for NAND gates under different circuit parameters. In SECTION 5.3, the performance of NOR gates is analyzed to verify the model on pull-up networks. Compensation factor for small load capacitance and adjustment factor for slope input effect are given and summarized in SECTION 5.2 and SECTION 5.3 respectively. The analytical delay model for inverters is verified in SECTION 5.4. Comparisons are performed under a wide range of input rise (fall) time and load capacitance. Time-domain output waveforms are also presented for different input rise (fall) time. In SECTION 5.5, we compare the outputs of a 1-bit full adder circuit under different input conditions. A 4-bit carry look-ahead adder is analyzed in SECTION 5.6. Performance comparisons between the model and SPICE are done under different circuit parameters. In SECTION 5.7, the output waveform of a parity generator is compared with that of SPICE. The results from the model show a good agreement with SPICE simulation for the circuits described. An overall deviation is less than 10 percent as compared with SPICE.

## 5.2 NAND Gates

As illustrated in SECTION 2.3, a complex gate may be collapsed into a series-connected pull-down and pull-up MOSFET structure. To reduce the effort of calculation, the series-connected network is further collapsed into a 4-transistor circuit as shown in Fig.3.11 according to the operating characteristics of the transistors. A CMOS NAND gate is a typical gate with a series-connected pull-down MOSFET network, which makes it a perfect case in proving the validity of our collapsing scheme. In this section, we compare our model with SPICE on NAND gates. We choose the channel length of the transistors $L = 0.6\mu m$. CMOSIS5 Level 3 process parameters are used both in the model and in SPICE simulation.

### 5.2.1 5-Input NAND Gate

A 5-input NAND gate is shown in Fig.5.1 We perform simulations under different load capacitance, trigger position and input slope. Simulation results from the model are compared with those from SPICE.

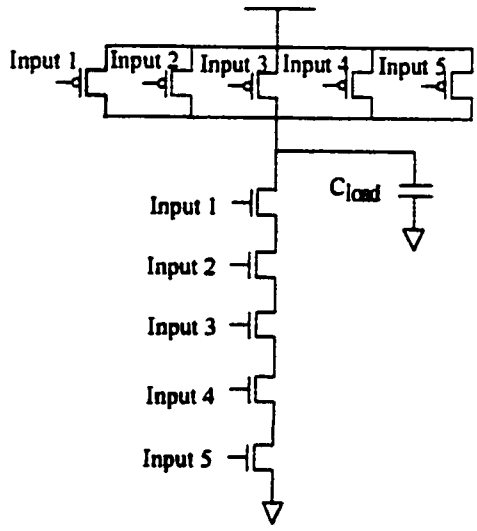


**Fig. 5.1: 5-input NAND gate.**

### 5.2.1.1 Output Waveform for Step Input

As presented in CHAPTER 3, the output waveform of a step input may be modeled by a piece-wise linear function of time. We compare the output waveforms of the 5-input NAND gate from our model with SPICE simulation results. Input 4 is taken as the trigger input with input rise time *1ps* to model the step input. Load capacitance is varying at $C_{load}$ = *50fF, 100fF, 200fF, 500fF* and *1pF*. The comparison is shown in Fig.5.2.

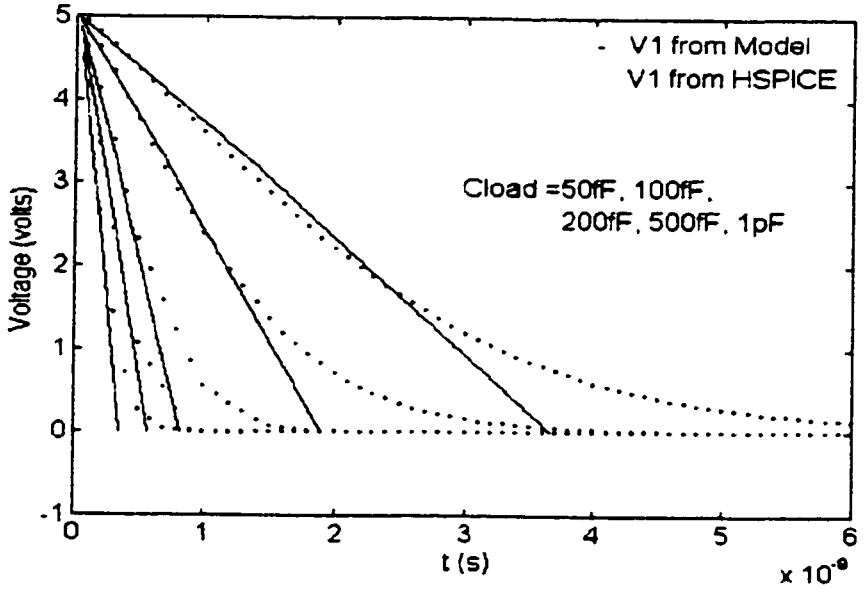


**Fig. 5.2: Comparison of output wave-forms between SPICE and analytical model for different load Capacitance.**

Fig.5.2 shows a very close approximation of our model to SPICE simulation in a very large range of load capacitance. This will enable a close ramp input approximation for the next circuit stage in a large CMOS circuit.

### 5.2.1.2 Effect of Load Capacitance

The effect of the load capacitance on the delay of the circuit is shown by varying load capacitance at *10fF, 30fF, 50fF, 100fF* and *200fF*. Transistor 4 is switched with an input rise time *1ps*.

All the other inputs are kept high. The model and SPICE simulation results are shown in Fig.5.3. It can be seen that the delay increases linearly with the load capacitance. The deviation between model and SPICE is within 8 percent.

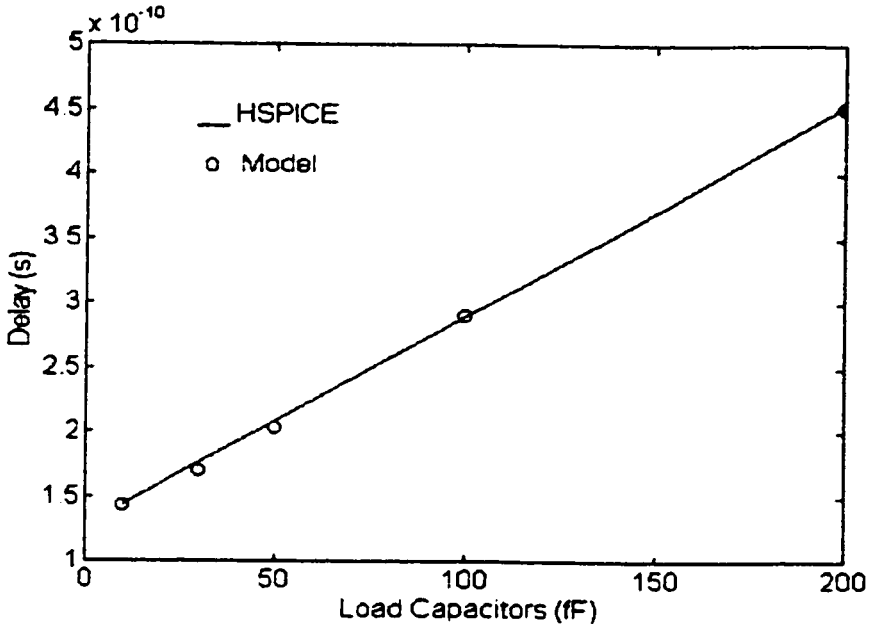


**Fig. 5.3: Delay vs. load capacitance for 5-input NAND gate.**

### 5.2.1.3 Trigger Effect

The delay of the circuit depends on the position of its trigger input which determines the initial charge stored in the internal nodes before switching. Assume that input $i$ is the trigger input in the 5-input NAND and all the other inputs are kept high. If all the transistors are of the same size, the initial charge stored in the circuit before switching is

$$Q_i = C_i(i-1)\,(V_{DD} - V_T) \tag{5.1}$$

$C_i$ is the node capacitance.

The delay of the step input response of the 5-input NAND gate with different trigger inputs and load capacitance is plotted in Fig.5.4. It can be seen that the circuit with trigger input closer to

the output node is faster, which coincides with (5.1). Fig.5.4 shows a good agreement between SPICE and our model.

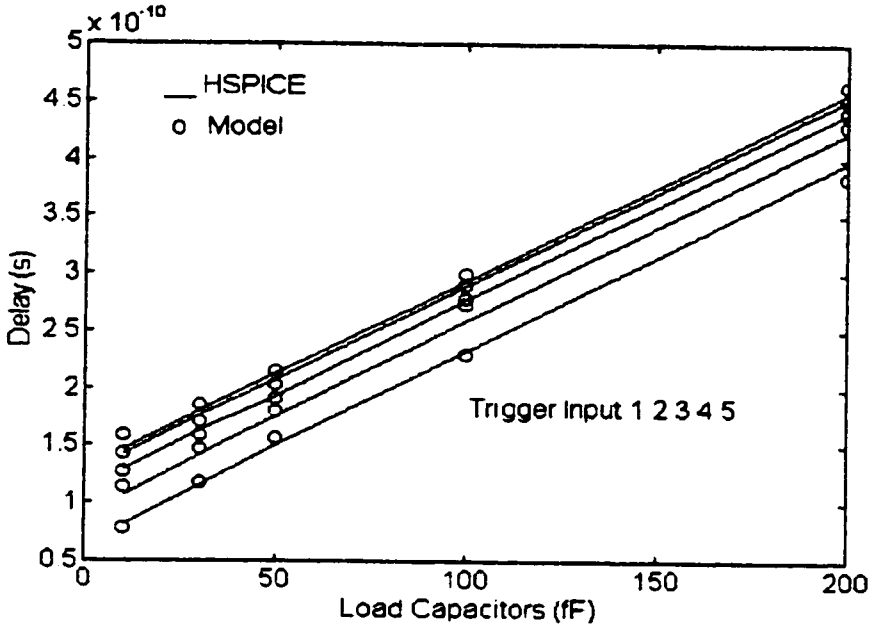


**Fig. 5.4: Delay vs. load capacitance for different trigger inputs.**

### 5.2.1.4 Ramp Input Effect

Fig.5.5 shows the input rise time effect on the delay of the 5-input NAND gate with rise time up to *1ns*. Circuit delay with input rise time ranging from *2ns* to *20ns* is shown in Fig.5.6. To avoid graphical clutter, $t = t_{dHL} + \frac{t_r}{2}$ is plotted instead of $t_{dHL}$. The size of the transistors is $W$= *4.8 um* and $L$ =*0.6um*. Load capacitance is *30fF, 50fF* and *100fF* respectively and trigger input is input 4. Maximum deviation from SPICE simulation is 10 percent.

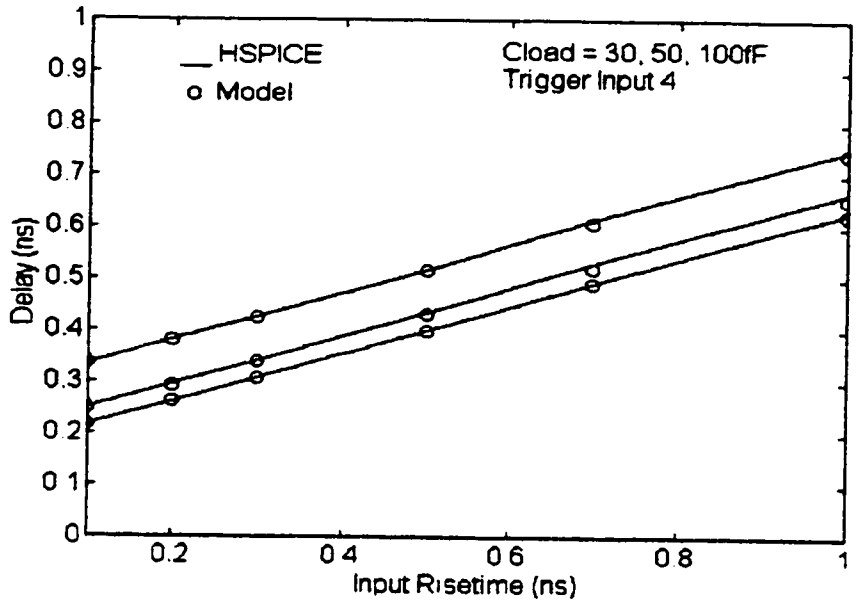


**Fig. 5.5: Effect of ramp input for rise time up to *1ns*.**

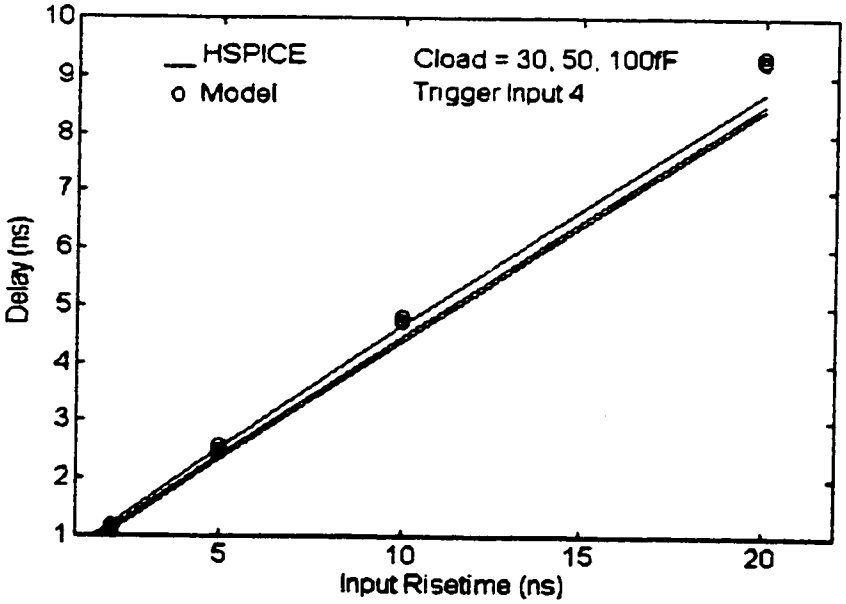


**Fig. 5.6: Effect of ramp input for rise time up to *20ns*.**

### 5.2.2 8-Input NAND Gate

As observed in [10], the delay ratio of a series-connected MOSFET structure over a single MOS transistor is much smaller in short-channel than in long-channel region. This result encourages more extensive use of NAND/NOR/complex gates, cascade voltage switch logic and hot-carrier resistant logic in the submicron circuit design [10]. If the maximum number of series-connected MOSFETs was considered to be five in $2\mu m$ designs, the number may be increased to seven or even eight for submicron circuit design. In this section, delay estimation is performed for an 8-input NAND gate. Simulation results are compared with SPICE. A good agreement is seen between the model and SPICE.

#### 5.2.2.1 Output Waveform for Step Input

We compare the step input response of an 8-input NAND gate between the model and SPICE. The output waveforms are shown in Fig.5.7. Load capacitance varies at *20fF, 50fF, 100fF, 500fF* and *1pF*. Transistor width is 4.8 $\mu m$. Delay deviation between the model and SPICE is within 10 percent.

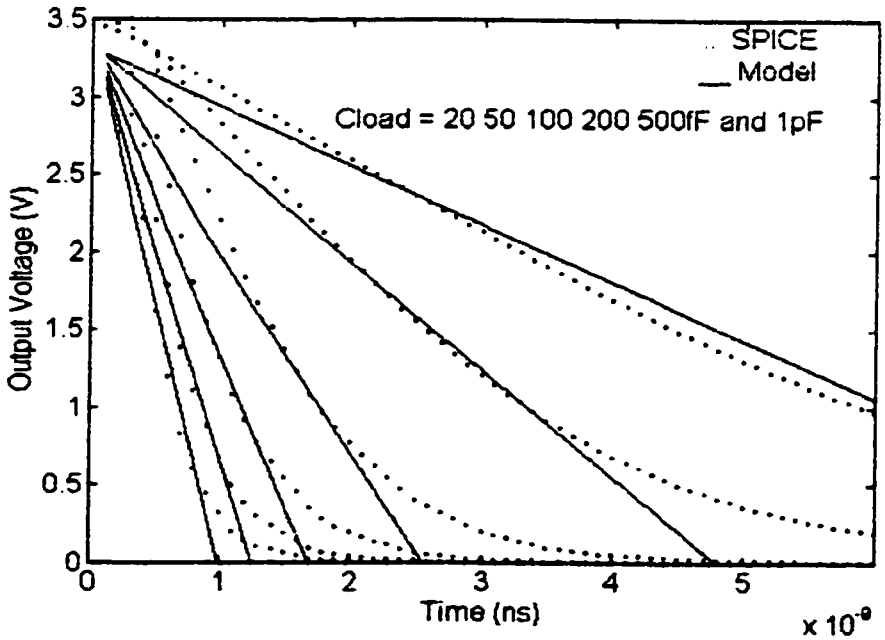


**Fig. 5.7: Step input response of an 8-input NAND gate.**

### 5.2.2.2 Effect of Load Capacitance

Fig.5.8 shows the delay of the step input response of an 8-input NAND gate under different load capacitance and transistor width. The load capacitance varies from *20fF, 50fF, 100fF, 200fF, 500fF* to *1pF*. The transistor width ranges at *1.2um, 2.4um, 4.8um, 9.6um and 14.4um*. It is clear that the increment of load capacitance results in less delay increment in larger transistors than in smaller ones. This indicates a larger driving ability in larger gates. A good agreement is seen between the model and SPICE.

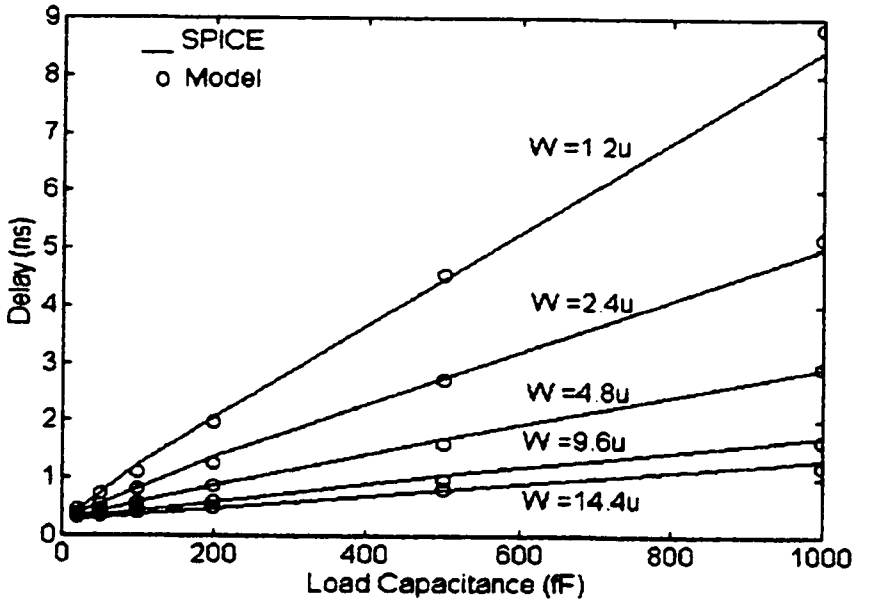


**Fig. 5.8: Delay vs. load capacitance for an 8-input NAND gate with transistor width ranges from *1.2um to 14.4um*.**

### 5.2.2.3 Ramp Input Effect

Fig.5.9 shows signal delays obtained with our model and SPICE for input rise time of *0, 500ps, 700ps, 1ns, 2ns*, and *5ns* respectively. The transistor width is $W = 4.8um$ and load capacitor varies at *20fF, 50fF, 100FF, 200fF, 500fF* and *1pF*.

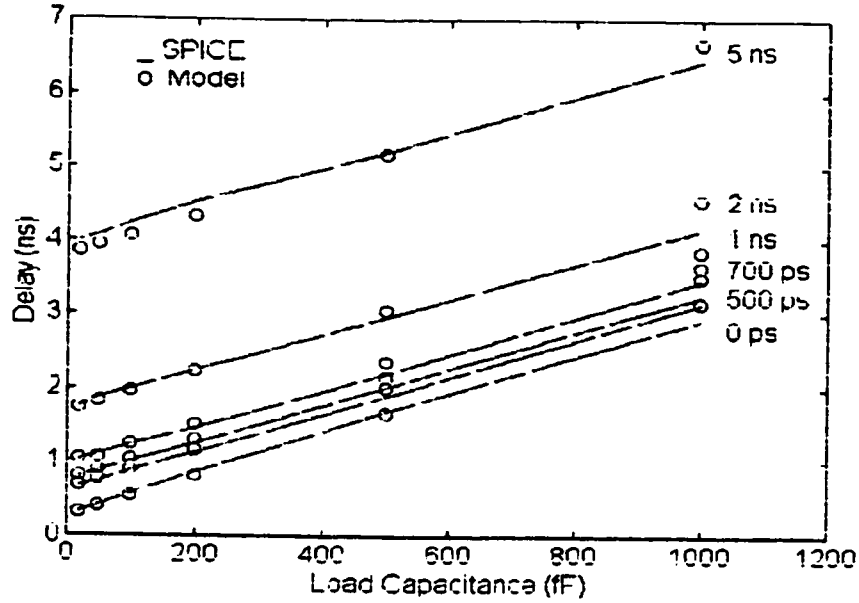


**Fig. 5.9: Delay of an 8-input NAND gate (*W= 4.8um L=0.6um*) for ramp inputs with rise time $t_r$ =*0, 500ps, 700ps, 1ns, 2ns* and *5ns*. Load capacitance ranges from *20fF* to *1pF*.**

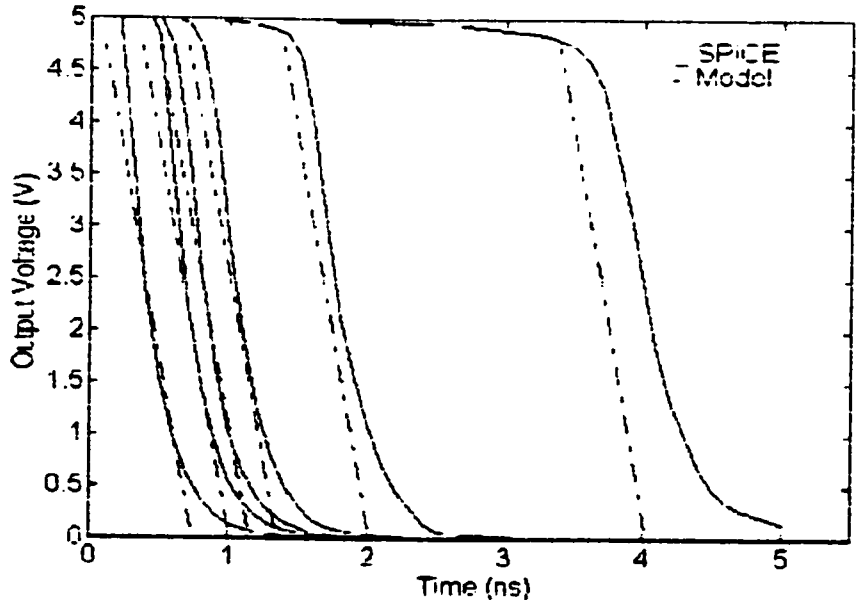


**Fig. 5.10: Output waveforms for an 8-input NAND with input rise time $t_r$ = *100ps, 500ps, 700ps, 1ns, 2ns* and *5ns*, load capacitance *20fF*.**

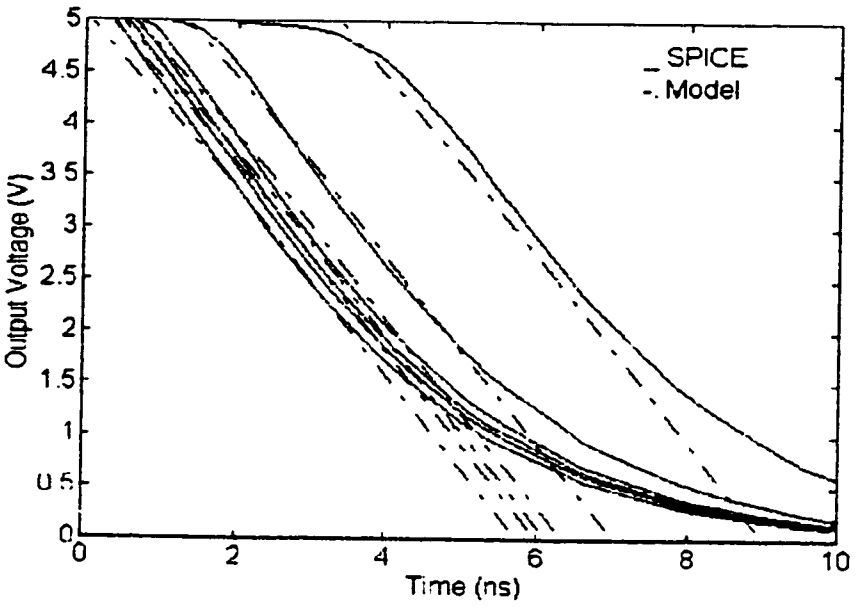


**Fig. 5.11: Output waveforms for an 8-input NAND gate with input rise time at *100ps, 500ps, 700ps, 1ns, 2ns,* and *5ns*, load capacitance *1pF*.**

As illustrated in CHAPTER 3, the derivative of a output waveform at half $V_{DD}$ level has to be modeled accurately since it determines the rise time (fall time) of the next stage in a complex CMOS circuit. Fig.5.10 and Fig.5.11 show the output waveforms with input rise time at *100ps, 500ps, 1ns, 2ns, and 5ns*. Load capacitance is *20fF* and *1pF* respectively. It can be seen that the outputs obtained with the piece-wise linear model fit well with those from SPICE. The maximum delay difference is within 10 percent.

### 5.2.3 Slope Input Effect Adjustment Factor and Load Capacitance Compensation Factor for Series-connected Pull-down MOSFET Network

In order to accurately model the performance of a series-connected pull-down MOSFET network, CMOS NAND gates with different number of inputs are simulated. Slope input effect adjustment factors are obtained for the pull-down series with different numbers of transistors. A non-linear delay characteristic has been observed when load capacitance is small compared to the

node capacitance. In order to compensate this non-linearity, load capacitance compensation factors are introduced, which are acquired by simulations on NAND gates with different number of inputs.

### 5.2.3.1 Slope Input Adjustment Factor

As illustrated in SECTION 3.6, an empirical adjustment factor is needed in calculating the slope input effect. The factor reflects the amount of the charge discharged to ground during input rise time. To obtain the factor for series-connected MOSFET pull-down networks, we perform simulations on NAND gates with number of inputs varying from 2 to 8. The result is summarized in Table 1.

**TABLE 1. Slope input adjustment parameter for NAND gates**

| | NAND gates | | | | | | |
|---|---|---|---|---|---|---|---|
| | **2-input** | **3-input** | **4-input** | **5-input** | **6-input** | **7-inputs** | **8-inputs** |
| Slope Input Adjustment Factor ($A_r$) | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |

### 5.2.3.2 Load Capacitance Compensation Factor

When load capacitance is small compared to the internal node capacitance in a series-connected MOSFET network, the delay of the output is not a linear function of the load capacitance. We denote the internal node capacitance as $C_i$, and

$$C_i = C_{dn} + C_{sn}$$

$C_{dn}$ and $C_{sn}$ are the drain and source capacitance respectively of a NMOS transistor. If we denote load capacitance compensation factor as $A_{cap}$, the output load capacitance may be adjusted as

$$C_L = C_{load}\left(1 + \frac{A_{cap}}{C_{load}} C_i\right)$$

The capacitance compensation factors is summarized in Table 2 for NAND gates with number of inputs ranging from 2 to 8.

**TABLE 2. Compensation factor for load capacitance.**

| | NAND gates | | | | | | |
|---|---|---|---|---|---|---|---|
| | 2-input | 3-input | 4-input | 5-input | 6-input | 7-input | 8-input |
| Compensation factor ($A_{cap}$) | 2 | 3 | 4 | 5 | 5 | 6 | 7 |

## 5.3 NOR gates

In order to estimate the performance of pull-up network, in this section, we compare our model to SPICE on NOR gates. Simulations are performed for NOR gates with different number of inputs. Slope input adjustment factor and load capacitance compensation factor are estimated for series-connected MOSFET pull-up network with different number of transistors.

### 5.3.1 5-Input NOR gate

Fig.5.12 shows the 5-input NOR gate. The transistor channel length is *0.6* microns. The comparison is done under different load capacitances, trigger positions and input fall time.

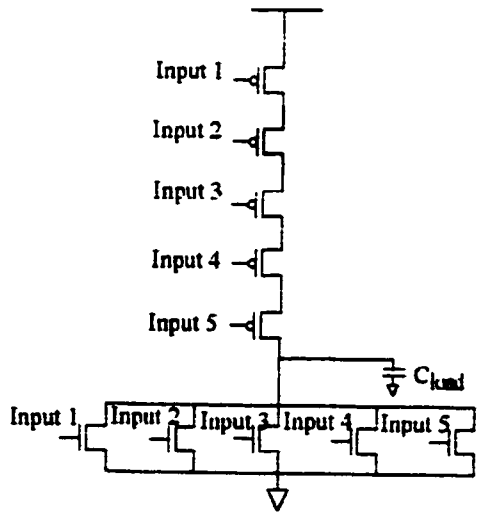


**Fig. 5.12: 5-input NOR gate**

### 5.3.1.1 Load Capacitance Effect

Fig.5.13 shows the delay of a 5-input NOR gate with $W = 4.8\mu m$ and $L = 0.6\mu m$. The load capacitance varies at *1fF 50fF 100fF 200fF 500fF 1pF.* Trigger inputs are 1 and 5 respectively. A good agreement is seen between the model and SPICE.

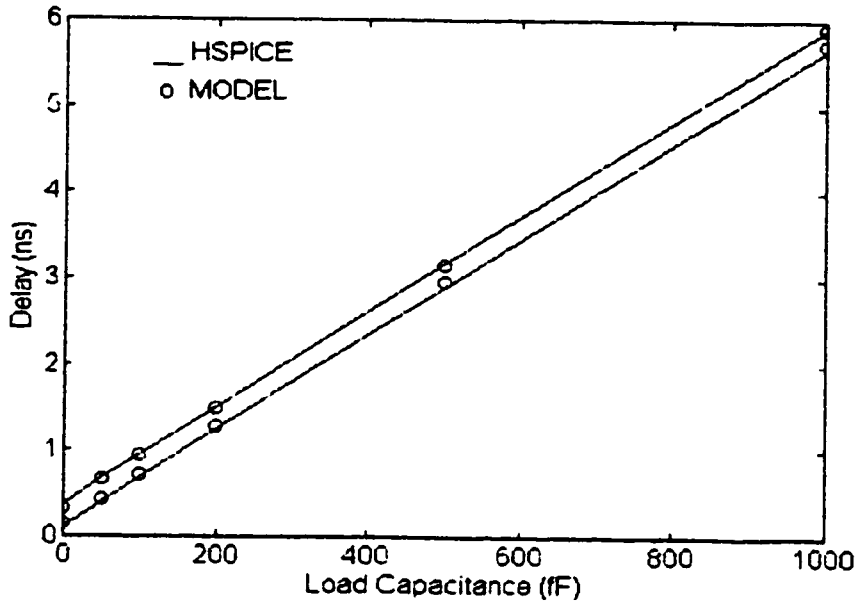


**Fig. 5.13: Delay for 5-input NOR gate with different load capacitance.**

### 5.3.1.2 Effect of Trigger Position

The trigger position effect on the 5-input NOR gate is given in Fig.5.14 for load capacitance ranging from *50fF* to *200fF*. The trigger inputs are input 1, 2, 3, 4, and 5 respectively. It can be seen that for a step input, the closer is the trigger input to the output, the faster the output response.

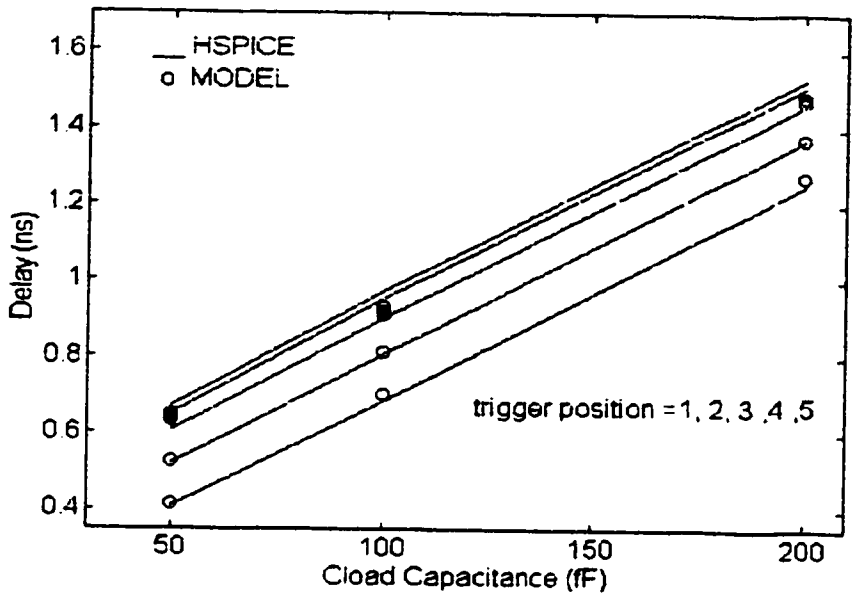


**Fig. 5.14: Effect of trigger position on 5-input NOR.**

### 5.3.1.3 Ramp Input Effect

Fig.5.15 and Fig.5.16 show the delay of the 5-input NOR gate for trigger input 1 and 5 respectively with input fall time ranging at *0.1ns 0.5ns 1ns 2ns 5ns 10ns* and *20ns*. The load capacitance varies from *50fF* to *200fF*.

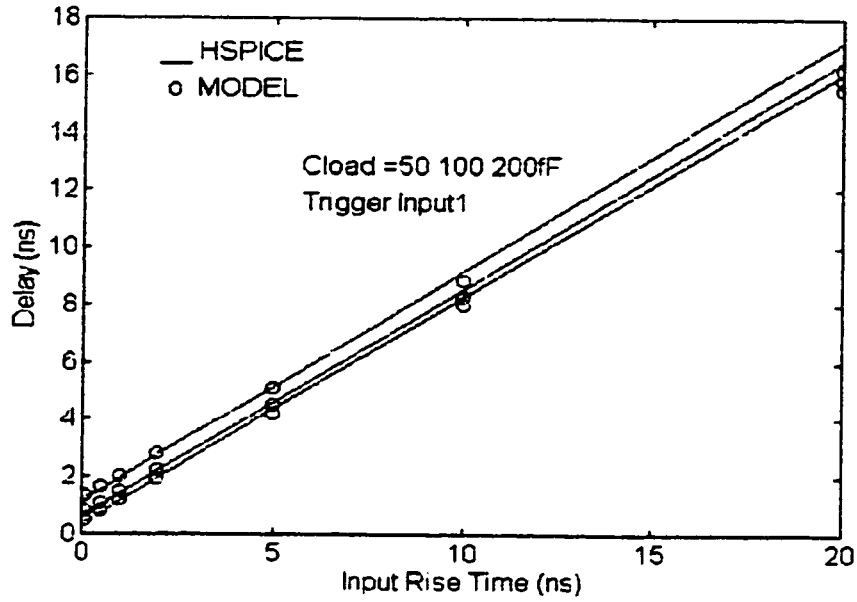


**Fig. 5.15: Input rise time effect on 5-input NOR gate for trigger input 1.**

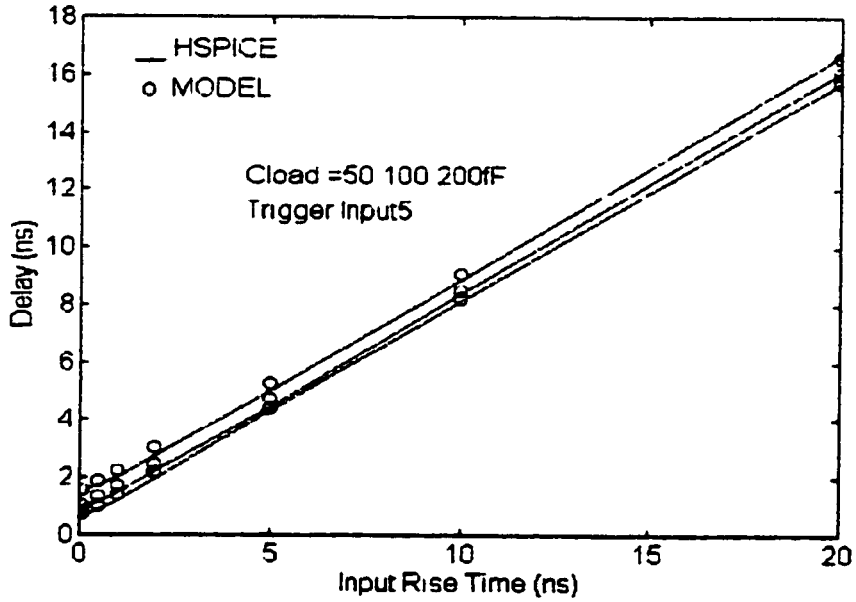


**Fig. 5.16: Input rise time effect on 5-input NOR gate for trigger input 5.**

### 5.3.2 Slope Input Adjustment Factor and Load Capacitance Compensation Factor for Series-connected Pull-up Network

NOR gates with wide range of load capacitance and input fall time are simulated and compared with the model. Slope input adjustment factor and load capacitance compensation factor for series-connected MOSFET pull-up networks are obtained by simulations on NOR gates with the number of inputs ranging from 2 to 8. With the acquired slope input adjustment factor and load capacitance compensation factor, the overall difference of the model from SPICE is within 10 percent for the pull-up networks. The slope input adjustment factor and load capacitance compensation factor for series-connected MOSFET pull-up networks are listed in Table 3.

**TABLE 3. Slope input adjustment factor and load capacitance compensation factor for series-connected pull-up network.**

| | NOR gates | | | | | | |
|---|---|---|---|---|---|---|---|
| | 2-input | 3-input | 4-input | 5-input | 6-input | 7-input | 8-input |
| Slope Input Adjustment Factor | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |
| Load Capacitance Compensation Factor | 1.0 | 1.0 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |

## 5.4 Inverter

As illustrated in SECTION 3.7, we may calculate the delay of an inverter by approximating the waveforms of ramp inputs. Fast input case and slow input case are classified according to the value of output voltage when the input reaches its final value. In this section, we simulate CMOS inverters under different circuit parameters such as transistor size, load capacitance, input rise or fall time. Simulation results from SPICE are compared with the calculated delay for both high to low and low to high case.

### 5.4.1 Inverter Switching from High to Low

The output waveforms of an inverter driving a same size inverter under different input rise time are shown in Fig.5.17. The size of the inverters is $W = 4.8\mu m$ and $L = 0.6\mu m$. The input rise time is *0.1ns, 0.2ns, 0.5ns, 1ns, 2ns, 5ns* and *10ns* respectively.

Fig.5.18 shows the high to low delay of the inverter in a wide range of input rise (fall) time and load capacitance. Input rise time varies at *0, 0.1ns, 0.5ns, 0.7ns, 1ns, 2ns*, and *5ns*. Load capacitance is taken as *0, 20fF, 50fF, 100fF, 200fF, 500fF* and *1pF* respectively. The delay is calculated as $t_{delay} = t_{dHL} + \frac{T}{2}$ to avoid graphic clutter, where $T$ is input rise time. A large variation occurs for step input response of inverters with small capacitance due to the non-linear characteristic of MOSFET device. However, a good agreement is seen between SPICE and model for input with non-zero rise time. Difference percentage is within 10 percent for the given range of load capacitance.

Fig.5.19 shows the delay of an inverter versus input rise time with $W = 14.4\mu m$ and $L = 0.6\mu m$. Load capacitance is *100fF*. An excellent agreement is shown between SPICE and the model. The maximum difference is less than 3 percent.

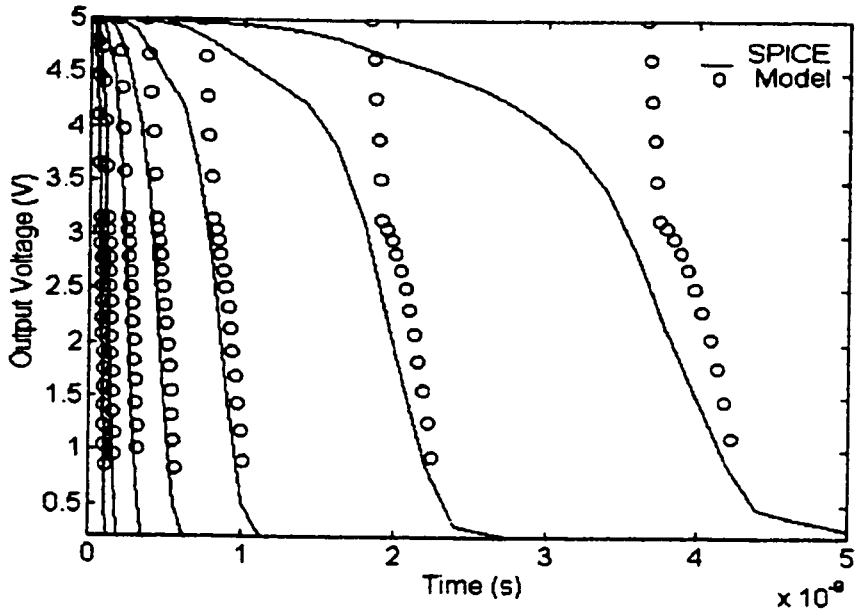


Fig. 5.17: Output waveforms of an inverter driving another same size inverter, $W = 4.8um$.

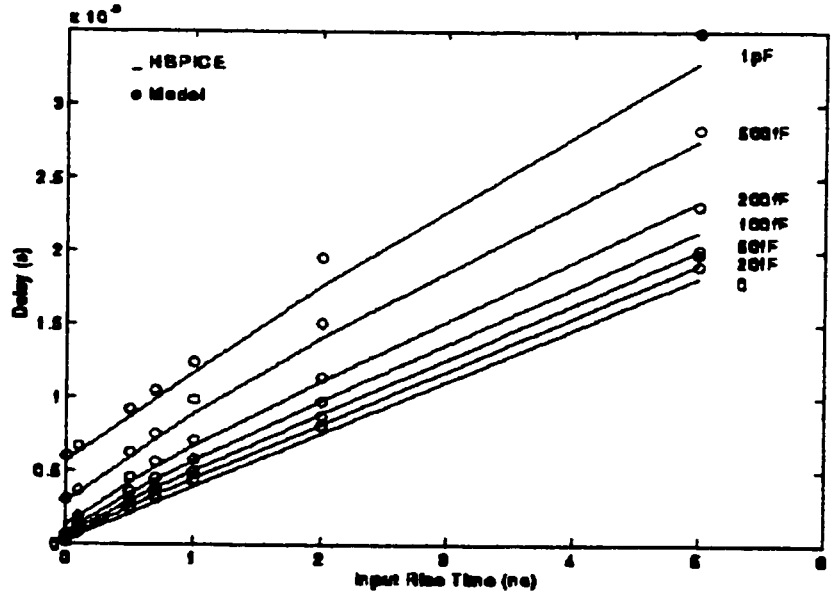


Fig. 5.18: Delay for inverter switching from high to low, $W = 4.8um$, $t_{delay} = t_{dHL} + T/2$.

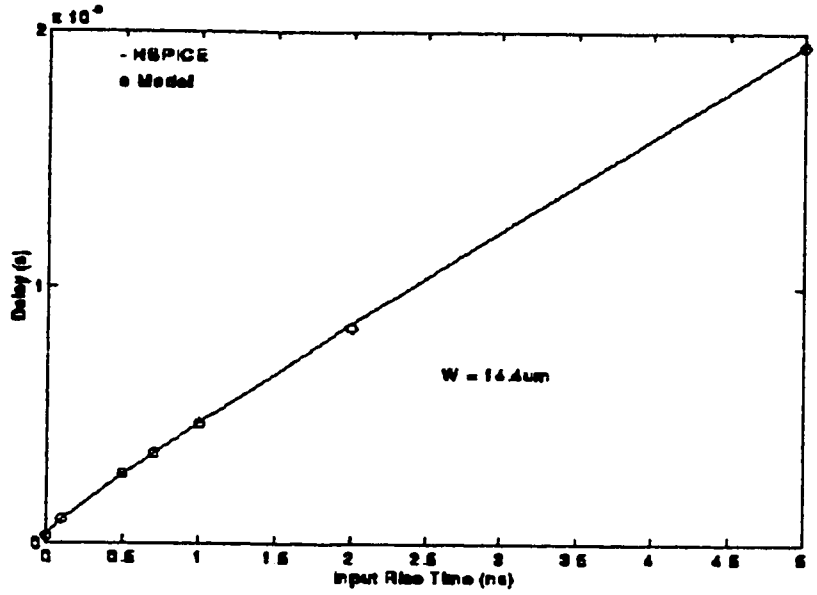


Fig. 5.19: Delay vs. input rise time for $W = 14.4um$, $t_{delay} = t_{dHL} + T/2$.

### 5.4.2 Inverter Switching from Low to High.

In the case that an inverter switches from low to high, the pull-up transistor is on. Fig.5.20 shows the output waveforms of an inverter driving another same size inverter for input fall time of *100ps, 200ps, 500ps, 1ns, 2ns, 5ns* and *10ns* respectively. The size of the transistors is $W = 4.8\mu m$. The waveforms obtained from the model show a good agreement with those from SPICE in a wide range of input fall time.

The delay of step input responses of the inverter versus load capacitance is illustrated in Fig.5.21. For load capacitance larger than *20fF*, the deviation is within 6 percent. Fig.5.22 shows the delay of the inverter versus input fall time for load capacitance of *0, 100fF and 500fF*. Input fall time varies at *0, 0.1ns, 0.5ns, 0.7ns, 1ns, 2ns*, and *5ns*. The maximum percentage difference between the model and SPICE is less than 8 percent.

The delays of an inverter with the size of $W = 14.4um$ under different circuit parameters are shown in Fig.5.23 and Fig.5.24. Fig.5.23 shows the delay versus load capacitance for step input

waveform. The load capacitance varies in a wide range of *0* to *1pF*. A good agreement can be seen between calculated delays and simulation results. With *100fF* load capacitance and the input fall time varying between *100ps* and *5ns*, we have the curve of delay versus input fall time as shown in Fig.5.24. The maximum deviation is less than 6 percent.

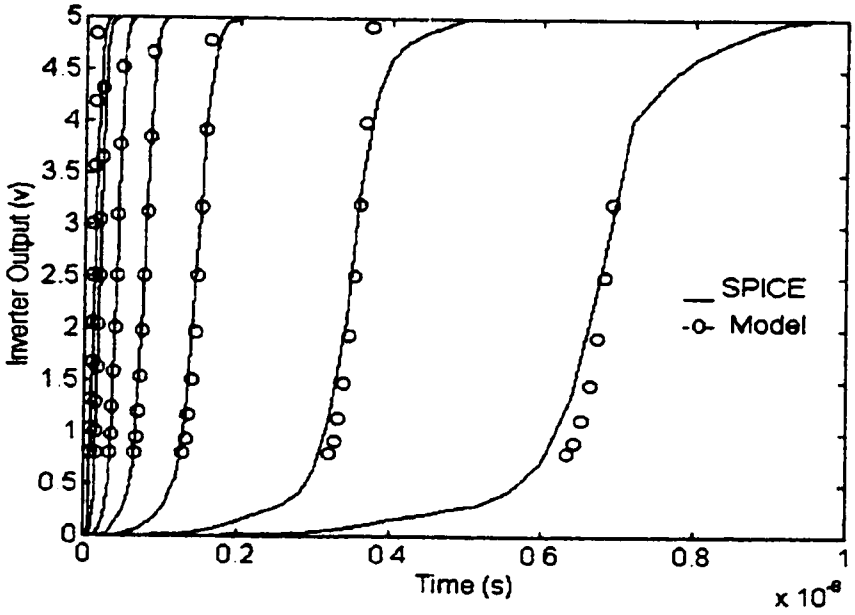


**Fig. 5.20: Output waveforms for inverter driving another same sized inverter. Inverter switches from low to high with input fall time *100ps*, *200ps*, *500ps*, *1ns*, *2ns*, *5ns*, and *10ns* respectively.**

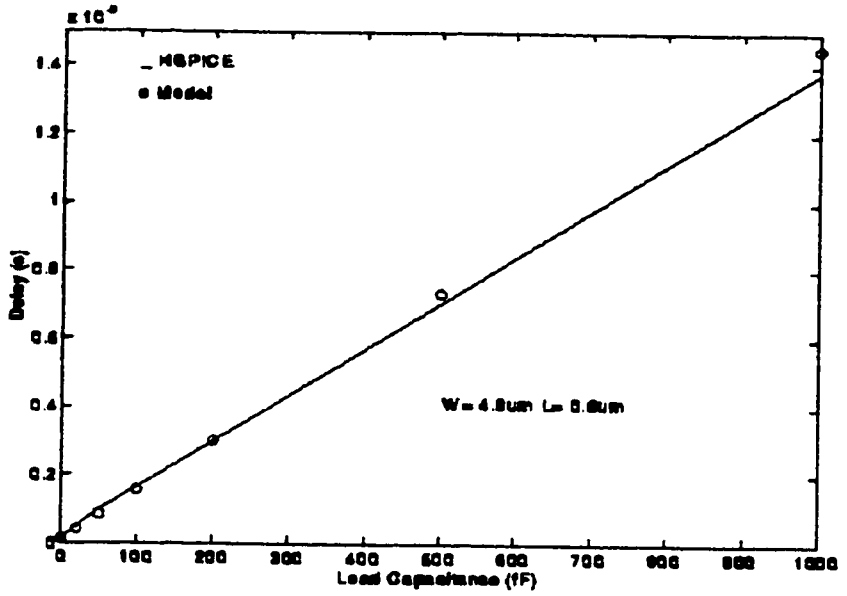


Fig. 5.21: Delay vs. load capacitance for step input, *W=4.8um and L=0.6um.*

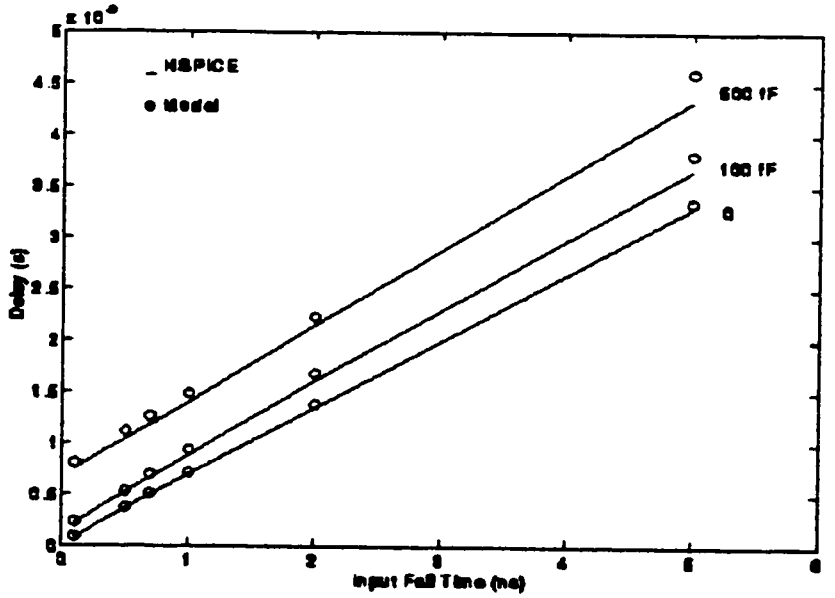


Fig. 5.22: Delay vs. input fall time for inverter switching from high to low, load capacitance *0, 100fF* and *500fF*, $t_{delay} = t_{dHL} + T/2$.

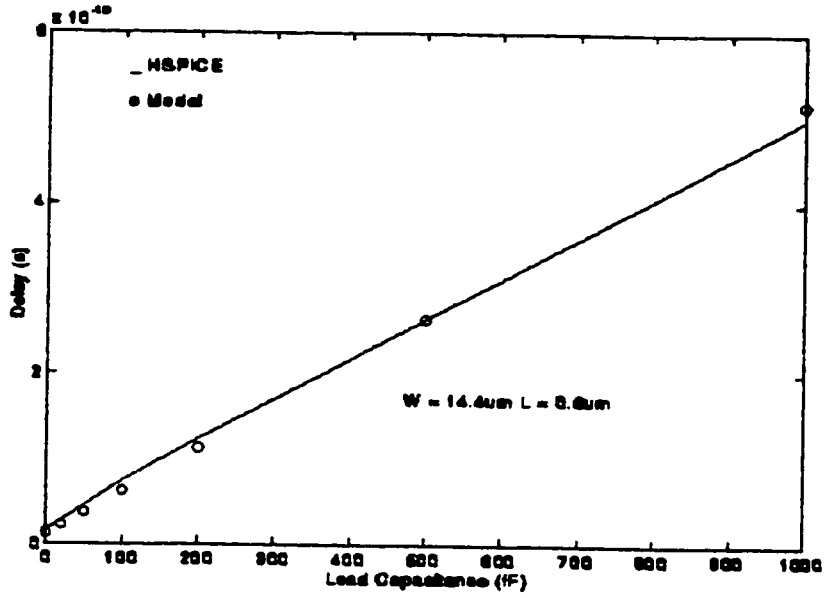


Fig. 5.23: Delay vs. load capacitance for step input, *W= 14.4um.*

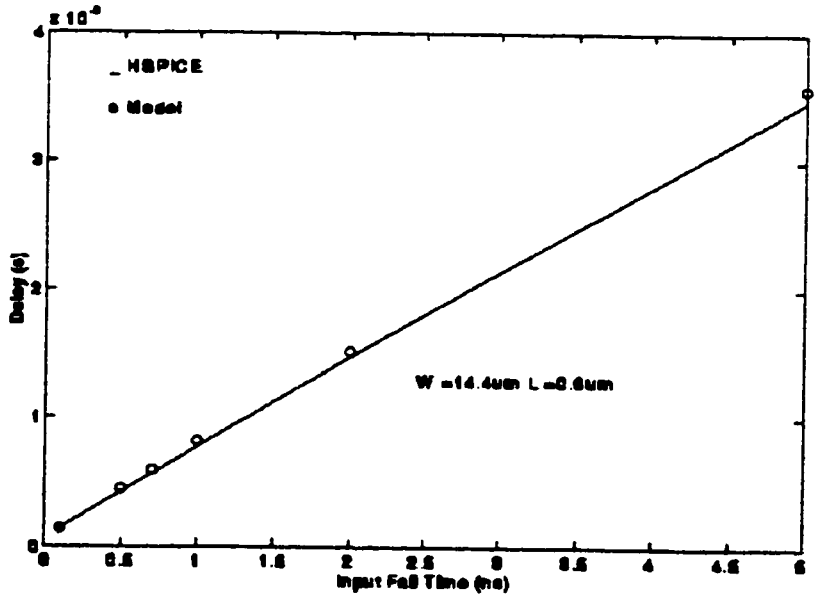


Fig. 5.24: Delay vs. input fall time for inverter with *100fF* load capacitance, *W=14.4um.*

## 5.5 Full Adder

In order to verify the developed model on circuits of different topologies and complexity, we test our model on a full adder circuit shown in Fig.5.25 under two different input conditions. The node capacitance are estimated from the gate, source and drain capacitances of the transistors. The circuit is divided into stages and an event-driven simulation is performed.

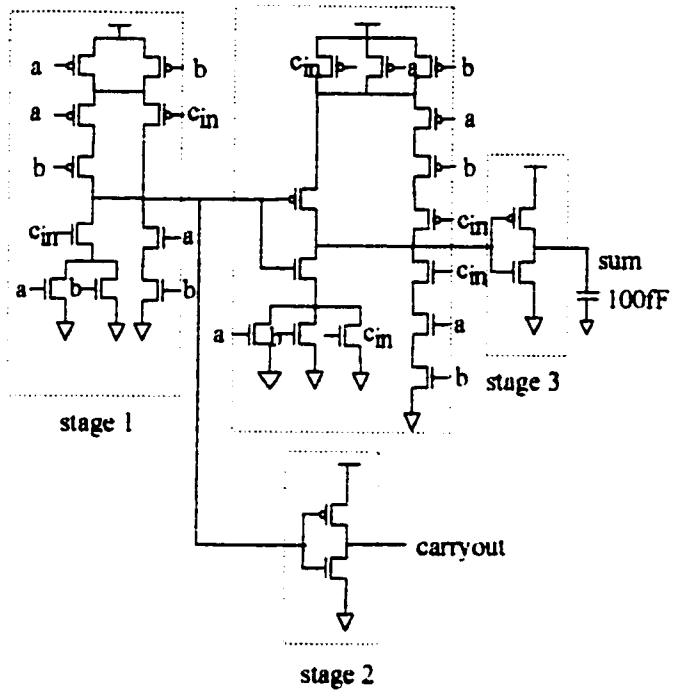


**Fig. 5.25: A 1-bit full adder circuit.**

The size of the transistors are $W = 4.8um$, $L = 0.6um$. The circuit is simulated with two sets of input combinations shown in Table 4. For the input combination in case 1, the adder may be collapsed into the circuit shown in Fig.5.26. Simulation results in Fig.5.27 and Fig.5.28 show a close approximation to SPICE.

**TABLE 4. Two sets of input combination for 1-bit full adder.**

| | a | b | $c_{in}$ | $\overline{\text{carryout}}$ | sum |
|---|---|---|---|---|---|
| Case 1 | 1 | 1 | 0 | 0 | 1 |
| | 0 | 0 | 1 | 1 | 0 |
| Case 2 | 0 | 0 | 0 | 1 | 0 |
| | 0 | 0 | 1 | 1 | 1 |

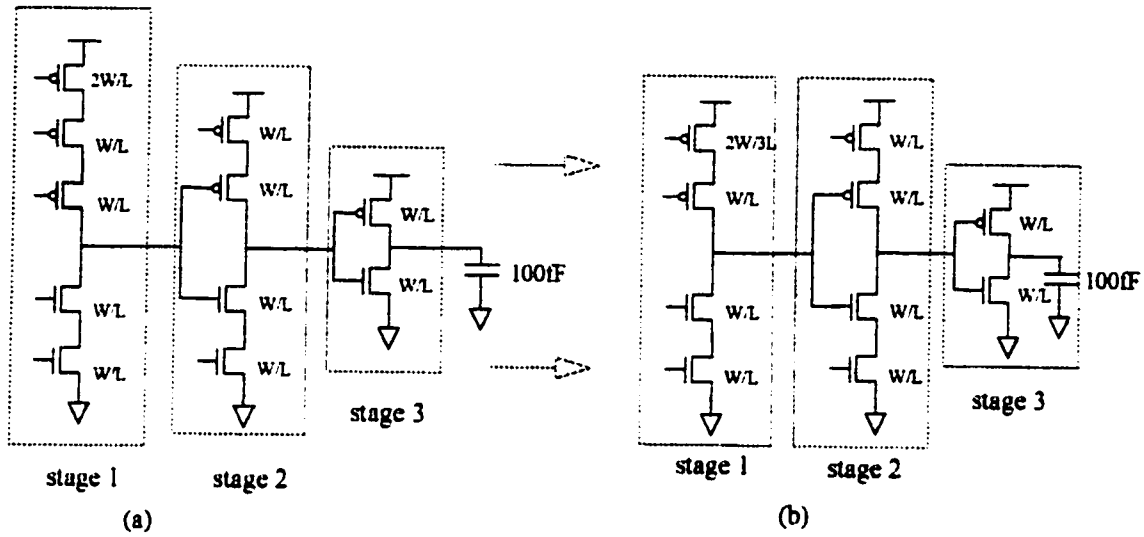


**Fig. 5.26: Collapsing scheme for 1-bit full adder circuit for case 1.**

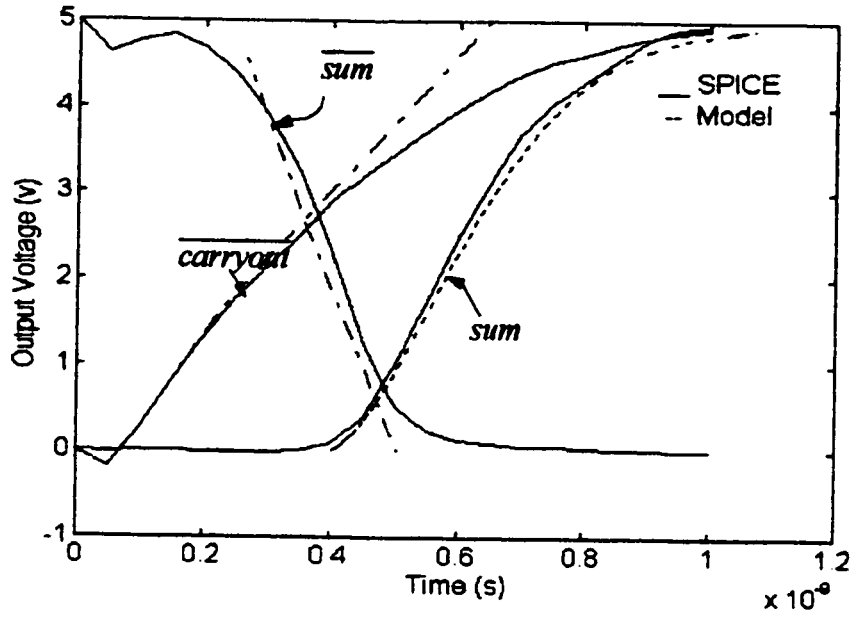


**Fig. 5.27: Case 1 output waveform for 1-bit full adder.**

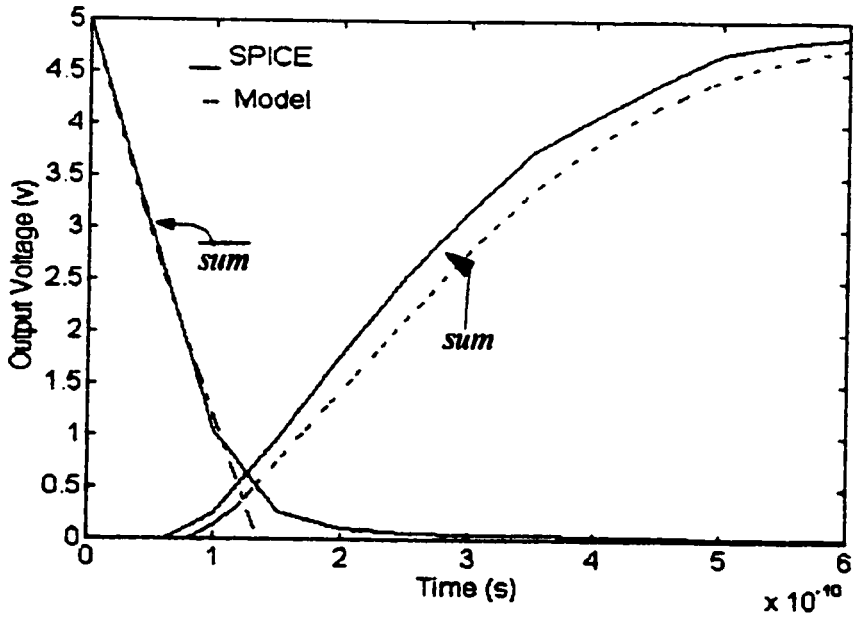


**Fig. 5.28: Case 2 output waveform for 1-bit full adder.**

## 5.6 4-bit Carry Look-ahead Adder

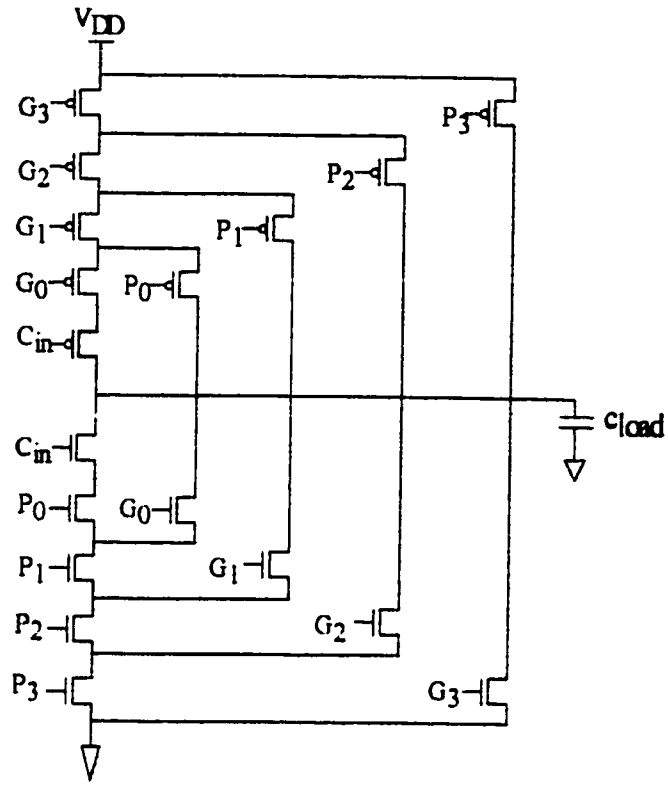


**Fig. 5.29: 4-bit carry look-ahead adder.**

A 4-bit carry look-ahead circuit is shown in Fig.5.29. Transistor sizes are $W = 4.8\mu m$, $L = 0.6\mu m$.To simplify simulation, we consider the worst delay case which occurs if all series transistors are switching simultaneously and if only one of the parallel devices is activated.

Simulation is done under different load capacitance and input rise time. The high to low delay of the step input response versus different load capacitance is shown in Fig.5.30. Fig.5.31 and Fig.5.32 show the delay versus input rise or fall time under different load capacitance. To avoid graphic clutter, the delay time is calculated as $t_{delay} = t_{dHL} + \frac{T}{2}$. The maximum difference between our model and SPICE is within 10 percent.

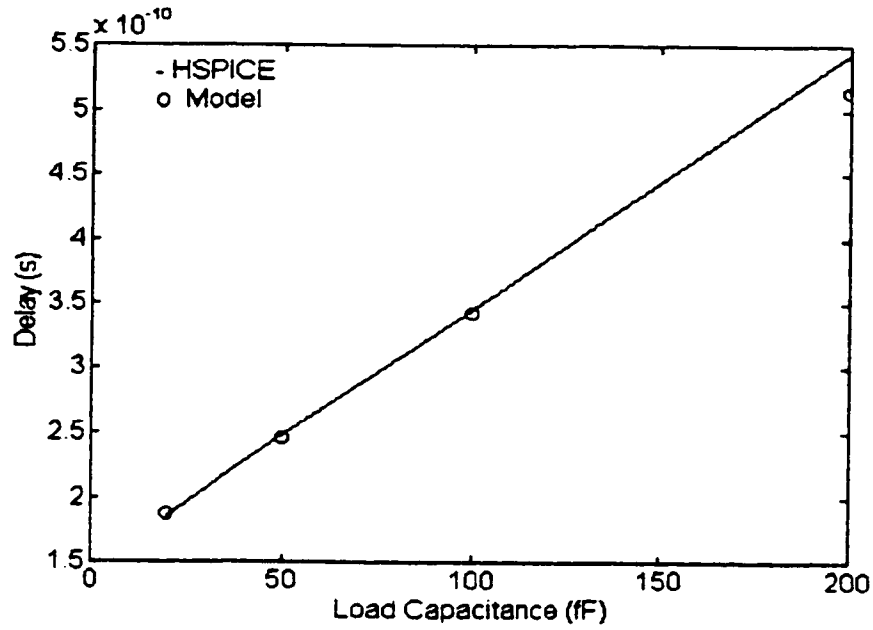


**Fig. 5.30: Delay (high to low) of step input response vs. load capacitance for 4-bit carry look ahead adder.**

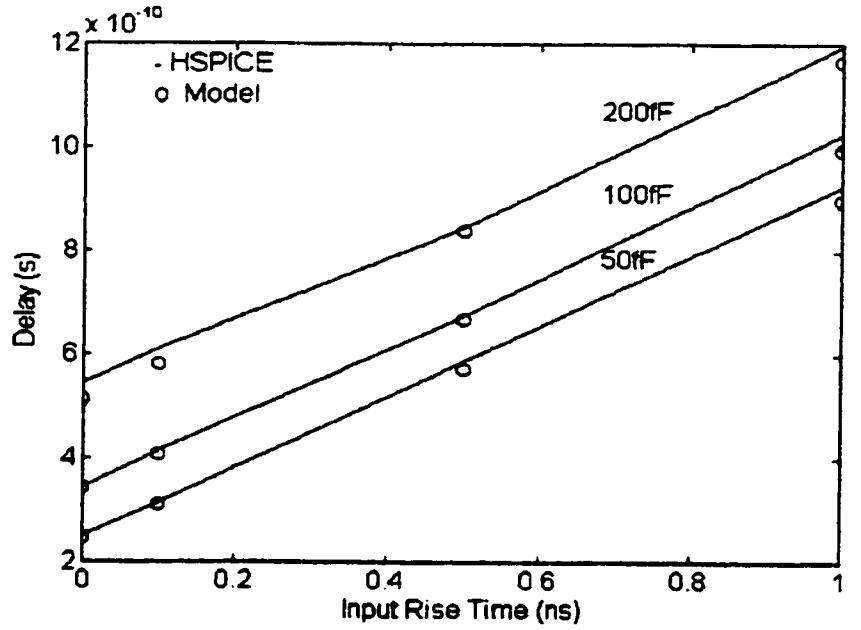


**Fig. 5.31: Delay (high to low) vs. input rise time for 4-bit carry look ahead adder with different load capacitance, $t_{delay} = t_{dHL} + T/2$.**

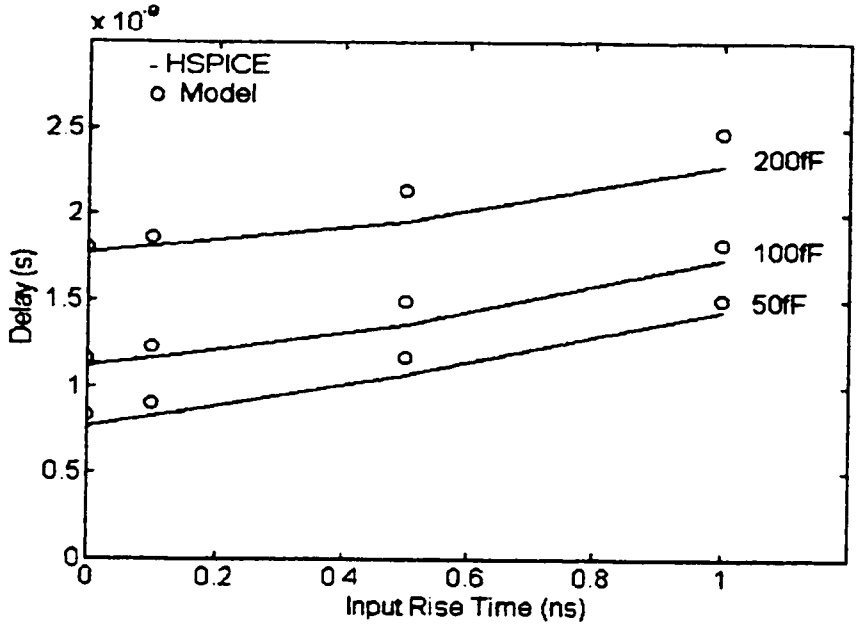


**Fig. 5.32: Delay (low to high) vs. input fall time for 4-bit carry look ahead adder with different load capacitance, $t_{delay} = t_{dHL} + T/2$.**

## 5.7 Parity Generator

A comparison between our model and SPICE is performed for a parity generator circuit shown in Fig.5.33. The topology of the circuit is different from previously tested circuits since it is not complementary CMOS logic. The high to low and low to high output waveforms are shown in Fig.5.34. Transistor width is *4.8 um*. Simulations are done with *50fF* load capacitance for input combinations shown below

| Output (Z) | a | b | c | d |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

The delay of the circuit obtained from the model is within 10 percent difference from SPICE

simulation.

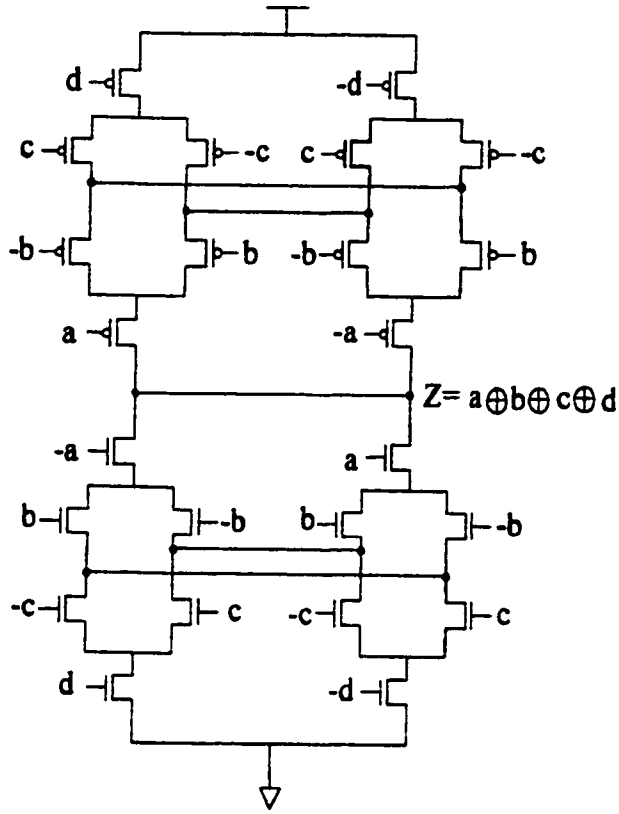


**Fig. 5.33: Parity generator circuit.**

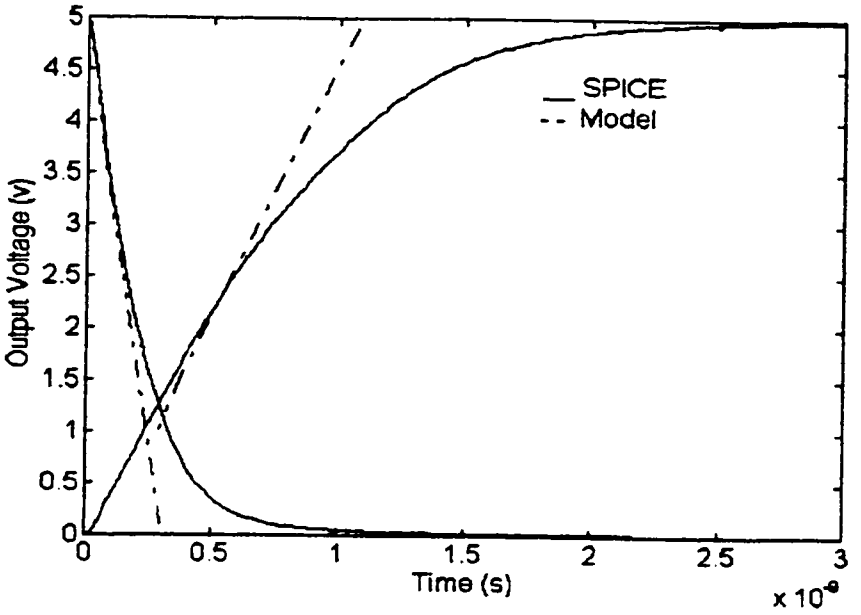


**Fig. 5.34: Output waveforms of parity generator in Fig.5.33.**

## 5.8 Discussion

We have presented comparisons between SPICE and our model for various CMOS circuits of different topology and complexity under wide range of circuit parameters. Slope input adjustment factor and load capacitance compensation factor are obtained by simulations on series-connected MOSFET network. An overall percentage difference between our model and SPICE is within 10 percent. In the next chapter, we will compare our model with SPICE in long-channel devices.

# CHAPTER 6 Comparison of Analytical Delay Model with SPICE for Long-Channel MOSFETs

## 6.1 Introduction

In this chapter, we simulate circuits with series-connected MOSFET pull-down and pull-up networks in long-channel regime. One of the important procedures in collapsing a complex CMOS gate into an equivalent circuit is to collapse a series-connected MOSFET pull-down and pull-up network into a 4-transistor circuit which is feasible for analysis using the analytical delay model. A NAND and a NOR gate are good examples to prove the validity of the collapsing scheme.

As illustrated in SECTION 3.2.4, there is no clear boundary between short-channel and long-channel devices. We consider devices satisfying the following equation

$$\frac{(V_{gs} - V_{Th})}{E_c L_{eff}} \leq 0.1$$

long-channel devices. SPICE level 3 CMOS4s process parameters are used in simulation, and the channel length is taken as 3 micrometers. The delay of a circuit depends on its load capacitance, the node capacitance, the trigger transistor, and input waveform. By varying these parameters, we quantify the difference between our model and SPICE in long-channel region, and thereby prove its validity. Simulation results obtained from the long-channel analytical delay model are compared with those from SPICE Level 3 and a good agreement is achieved.

## 6.2 5-Input NAND Gate

In this section we compare our analytical model with SPICE for a 5-input nand gate as shown in Fig.6.1. The transistor channel width is *8 um*.

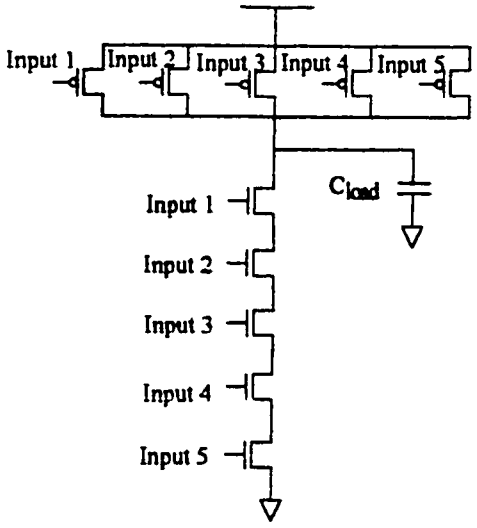


**Fig. 6.1: 5-input NAND gate.**

### 6.2.1 Output Waveform for Step Input

The output waveform of a series-connected MOSFET pull-down network with step inputs can be approximated by a linear piece-wise model as illustrated in SECTION 4.2. Fig.6.2 shows the comparison of time-domain output waveforms of the 5-input NAND gate between the model and SPICE. The load capacitance is varying at *1fF, 50fF, 200fF, 500fF* and *1pF*. Transistor $N_4$ is triggered by a step input while all the other inputs are kept high. A close approximation of the model to SPICE can be observed.

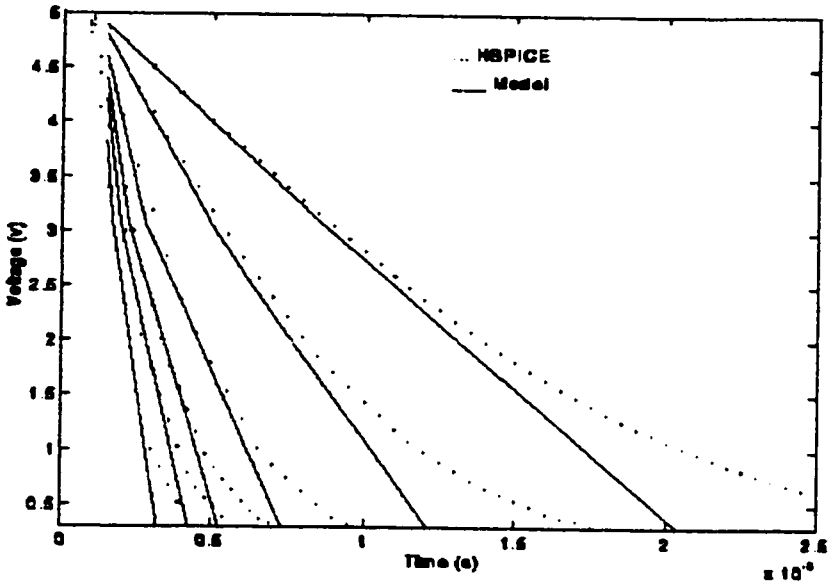


Fig. 6.2: Comparison of output wave-forms between the delay analytical model and SPICE. Load capacitance at the output node ranges from *1 fF, 50 fF, 200 fF, 500 fF* to *1 pF*.

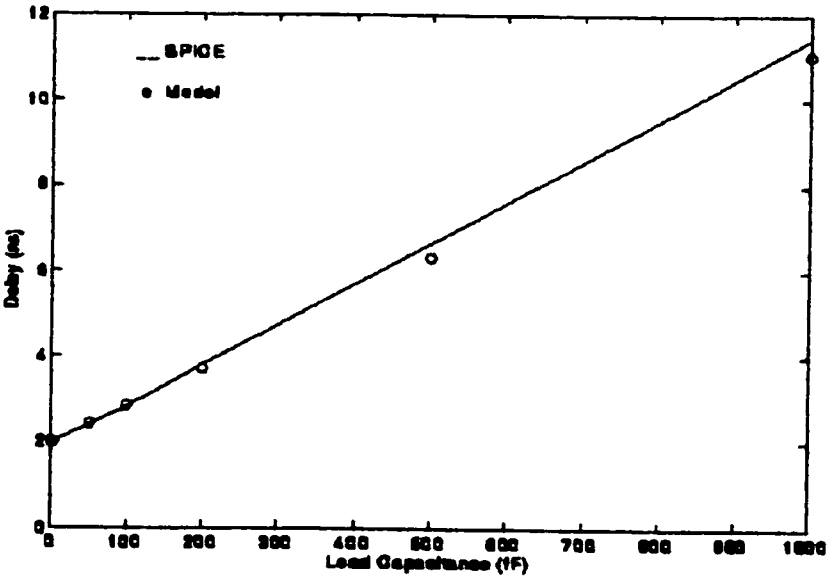


Fig. 6.3: Delay of step input response vs. load capacitance for 5-input NAND.

### 6.2.2 Effect of the Load Capacitance

The effect of load capacitance on delay of the 5-input NAND gate is shown in Fig.6.3. With load capacitance varying from *1fF, 50fF, 200fF, 500fF* to *1pF*, the delay increases linearly with the load capacitance. The comparison of the delay between the model and SPICE is given in Table 5. The simulation results show a good agreement between the model and SPICE. The maximum deviation between model and SPICE simulation is less than 6 percent.

**TABLE 5. Propagation delay of step input response for 5-input NAND gate under difference load capacitance.**

| $C_{load}$ (fF) | SPICE(ns) | Model (ns) | Difference Percentage (%) |
|---|---|---|---|
| 1 | 1.99 | 2.0 | 0.5 |
| 50 | 2.41 | 2.42 | 0.4 |
| 100 | 2.82 | 2.85 | 1.1 |
| 200 | 3.81 | 3.72 | -2.4 |
| 500 | 6.66 | 6.32 | -5.1 |
| 1000 | 11.45 | 11.05 | -3.5 |

### 6.2.3 Ramp Input Effect

As discussed in SECTION 3.6, the slope of input waveform has a profound impact on the delay of the output. It accounts up to30 percents of the total gate delay [1] and thus it is necessary to incorporate the input waveform dependences. Taking input 4 as trigger input, and keeping all the other inputs high, we observe the input slope effect on output delay by varying the input rise time from *0.1ns* to *10ns*. The load capacitance is *100fF*. Comparisons on high to low propagation delay are given in Table 6. A good agreement is seen between the model and SPICE. The maximum percentage difference is less than 10 percent.

**TABLE 6. Input Ramp Effect of 5-input NAND with load capacitance 100fF.**

| Input Rise Time (ns) | $t_{dHL}$ from SPICE3 (ns) | $t_{dHL}$ From Model (ns) | Percentage of Difference (%) |
|---|---|---|---|
| 0.1 | 3.09 | 2.91 | -5.8 |
| 0.2 | 3.11 | 2.93 | -5.8 |
| 0.3 | 3.12 | 3.13 | -2.9 |
| 0.5 | 3.15 | 3.15 | 0 |
| 0.7 | 3.18 | 3.26 | 2.5 |
| 0.8 | 3.18 | 2.92 | -8.1 |
| 1 | 3.22 | 2.94 | -8.7 |
| 2 | 3.3 | 3 | -9.1 |
| 5 | 3.5 | 3.24 | -7.4 |
| 10 | 3.84 | 3.62 | -5.7 |

Fig.6.4 shows the high to low propagation delays of the 5-input NAND with different load capacitance $C_{load}$ = $50fF$, $100fF$, and $200fF$. Input 4 is selected as the trigger input with input rise time varying from *100ps* to *1ns*. The delay for slow input rise time is plotted in Fig.6.5. To avoid graphic clutter we take $t_{delay} = t_{dHL} + \frac{T}{2}$, where $T$ is the input rise time.

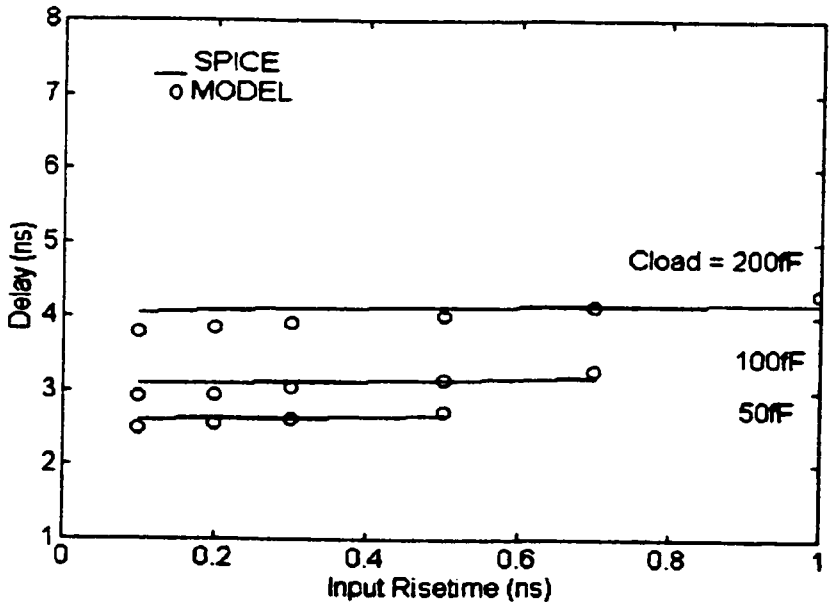


**Fig. 6.4: Delay vs. input rise time for fast input with load capacitance *50fF, 100fF* and *200fF*.**

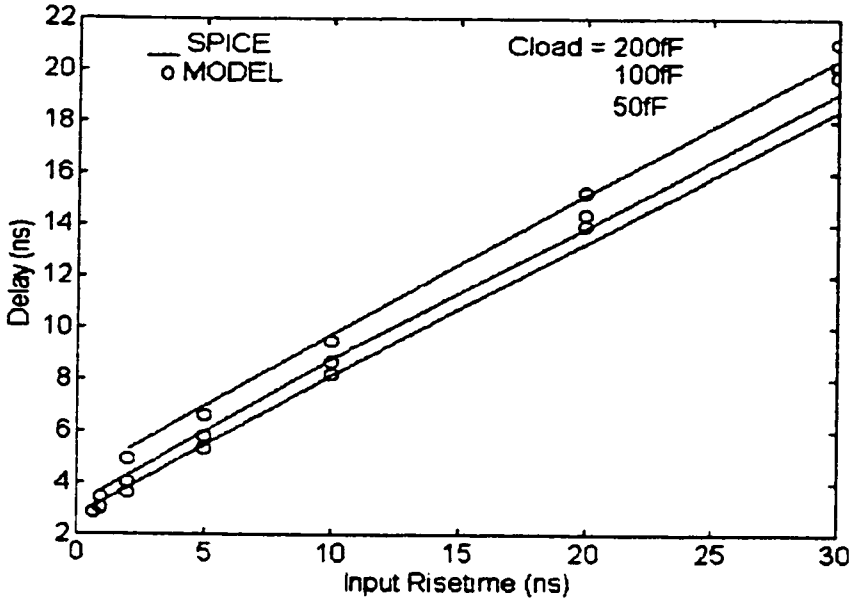


**Fig. 6.5: The input slope effect for slow and very slow input for load capacitance $C_{load}$ = *50 fF, 100 fF* and *200 fF*, $t_{delay} = t_{dHL} + T/2$.**

### 6.2.4 Effect of the Trigger Position

Fig.6.6 shows the delay versus load capacitance for different trigger inputs (trigger input 1 to 5). Assume step input and one trigger input at a time. The other inputs are kept high. As can be seen from Fig.6.6, the delay of the circuit increases linearly with the load capacitance. There is a step difference for different transistors switched, since some node capacitance will be charged or discharged depending on which transistor is switched. The closer is the trigger to the output node, the faster the circuit for a step input.

Fig.6.7, Fig.6.8 and Fig.6.9 show the delay versus input rise time for different trigger inputs with load capacitance 50fF, 100fF, and 200fF respectively. The delay is plotted as $t_{delay} = t_{dHL} + \frac{T}{2}$. We can observe that as the input rise time increases the step difference of delay for different trigger inputs decreases and at certain input rise time, the delays are the same. When the input rise time passes over the equal delay point, the closer is the trigger input to the

output, the larger the delay. The simulation results of our delay model match well with those of SPICE for input rise time less than *10 ns*.

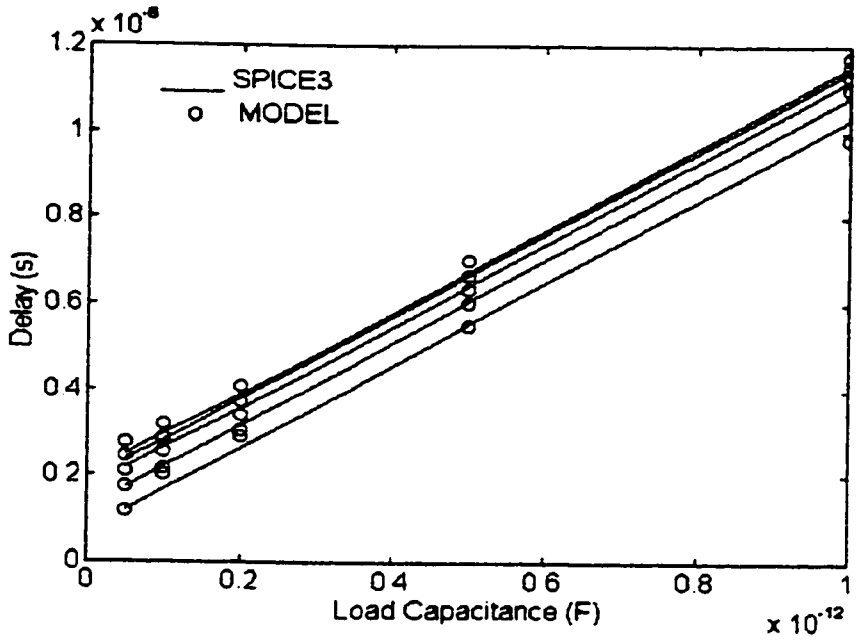


**Fig. 6.6: Effect of trigger position for step input.**

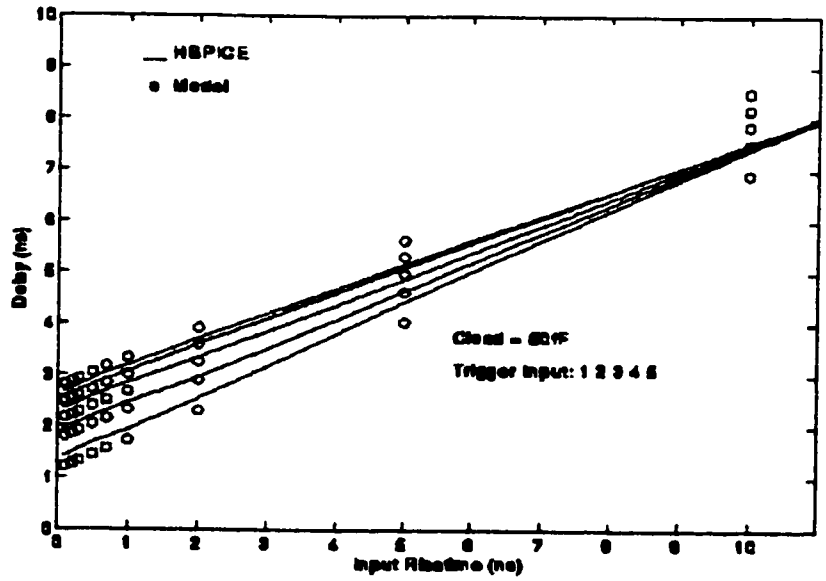


**Fig. 6.7: Delay vs. input rise time for different trigger inputs, load capacitance *50fF*, $t_{delay} = t_{dHL} + T/2$.**

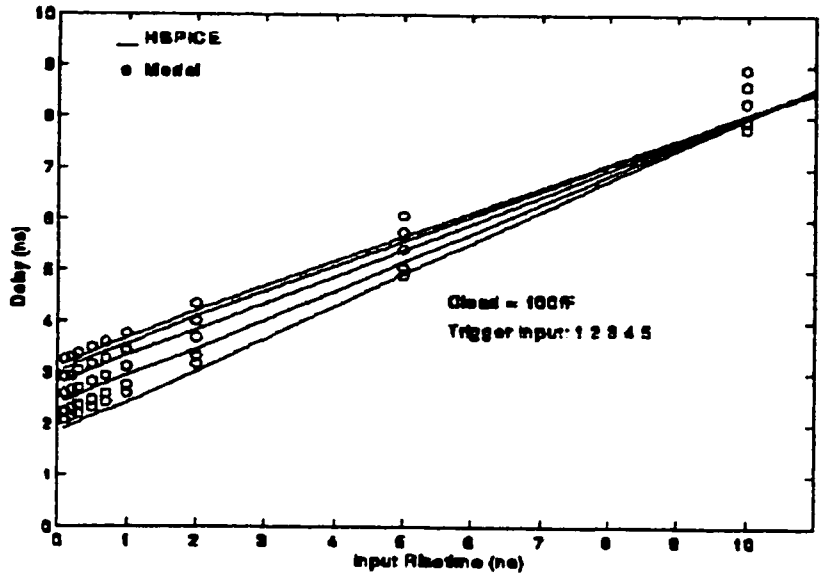


Fig. 6.8: Delay vs. input rise time for different trigger inputs, load capacitance *100fF*, $t_{delay} = t_{dHL} + T/2$.

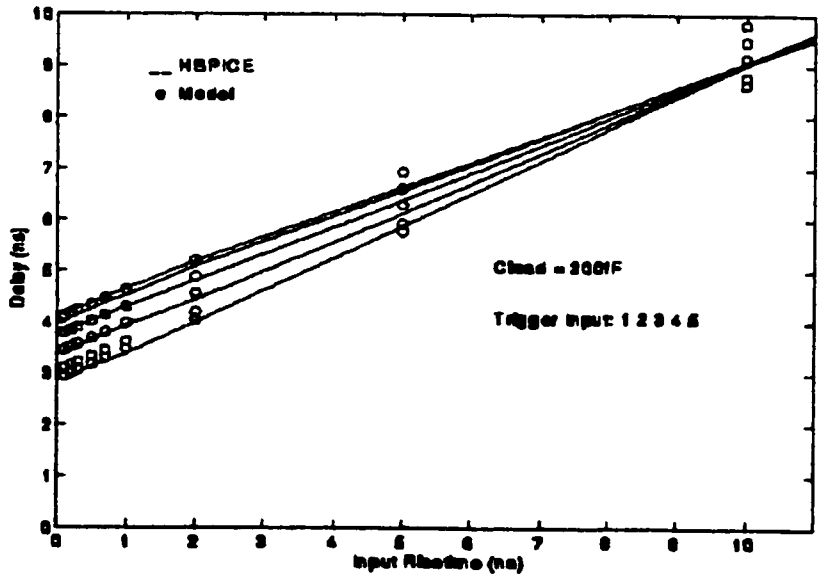


Fig. 6.9: Delay vs. input rise time for different trigger inputs, load capacitance *200fF*, $t_{delay} = t_{dHL} + T/2$.

## 6.3 5-Input NOR gate

In this section, we examine the pull-up network by applying our model to a 5-input NOR gate shown in Fig.6.10. The channel width is 8 microns. We compare the simulation results with SPICE. Delay under different load capacitance, trigger positions and input rise time is examined. The results show a good agreement between the model and SPICE.

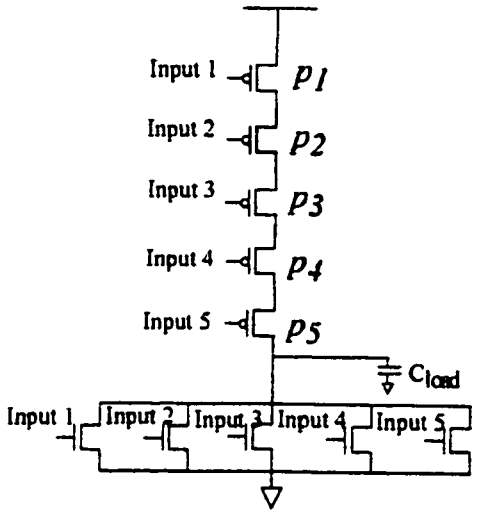


**Fig. 6.10: 5-input NOR gate**

### 6.3.1 Effect of Load Capacitance

The effect of load capacitance on delay is shown by varying load capacitance from *1 fF* to *1pF*. Transistor $P_4$ is switched by a step input and all the other inputs are kept low. As shown in Fig.6.11, the delay of the 5-input NOR gate increases linearly with the load capacitance. The maximum percentage difference between the model and SPICE is less than 7 percent.

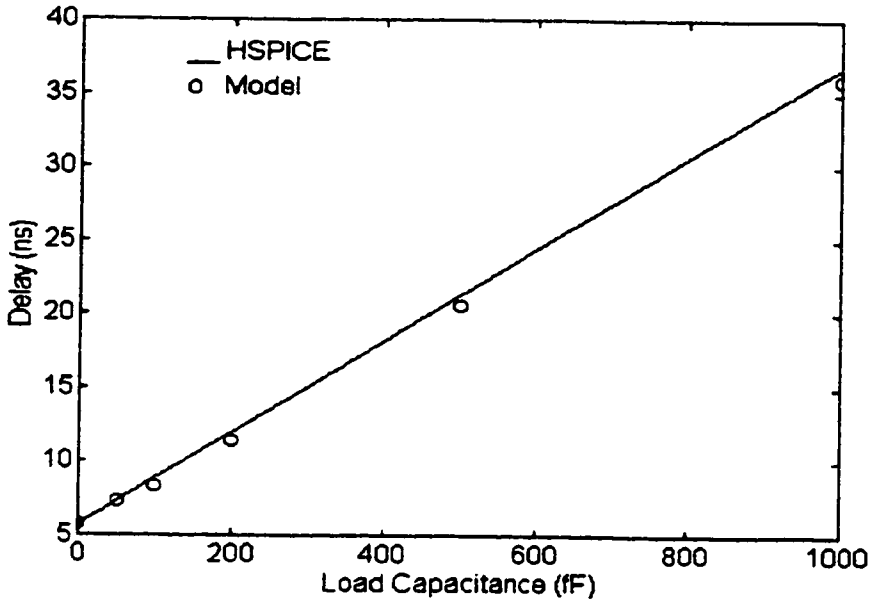


**Fig. 6.11: Load Capacitance effect on 5-input NOR gate for step input, trigger input 4.**

### 6.3.2 Effect of Trigger Position

The effect of trigger position on delay is examined with load capacitance at *50fF, 100fF* and *200fF* respectively. Assume step input and one trigger input at a time. Other inputs are kept low. Fig.6.12 shows the delay versus load capacitance for different trigger position. There is step difference of the delay with different trigger inputs. The closer is the trigger to the output node, the faster the circuit. The difference of simulation results between SPICE and model is less than 10 percent.

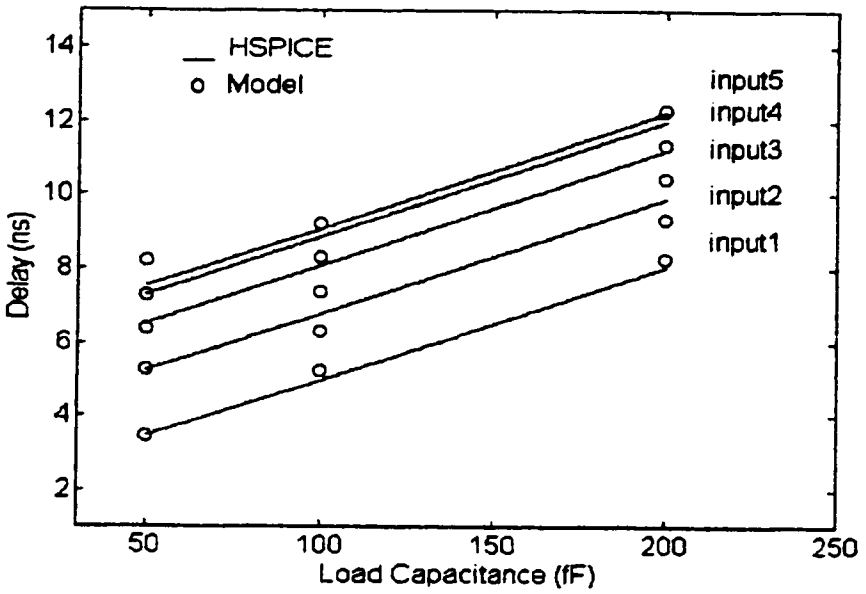


**Fig. 6.12: Delay vs. load capacitance for different trigger input. The input is step input.**

### 6.3.3 Effect of Input Rise Time

The effect of input fall time on delay of the 5-input NOR with different trigger inputs is shown in Fig.6.12, Fig.6.13, Fig.6.14 and Fig.6.15 respectively. Load capacitance for the 5-input NOR gate varies at *50fF, 100fF* and *200fF*. Input fall time ranges from *0.1ns* to *10ns*. The delay time is plotted as $t_{delay} = t_{dLH} + \frac{T}{2}$. Note that the faster is the input fall time, the larger the difference of delay between different trigger inputs. It also can be seen that the closer is the trigger input to the output the faster the circuit. The maximum difference between SPICE and the model is less than 15 percent, which occurs when input fall time is about *10ns*.

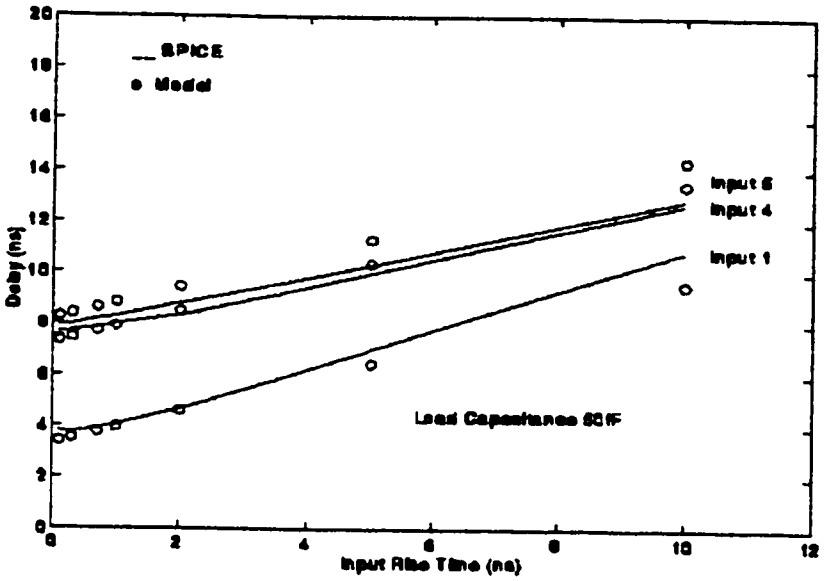


Fig. 6.13: Delay vs. input fall time for different trigger input, load capacitance *50fF.*

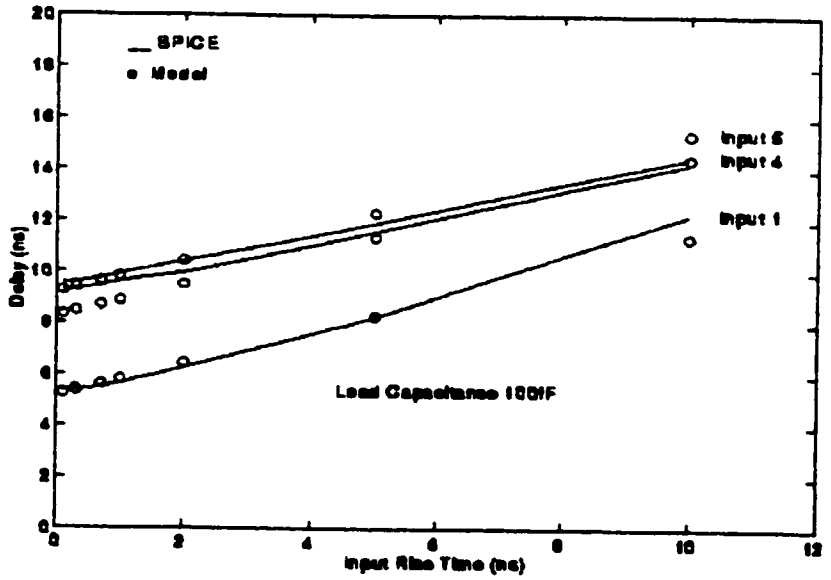


Fig. 6.14: Delay vs. input fall time for different trigger input, load capacitance *100fF.*

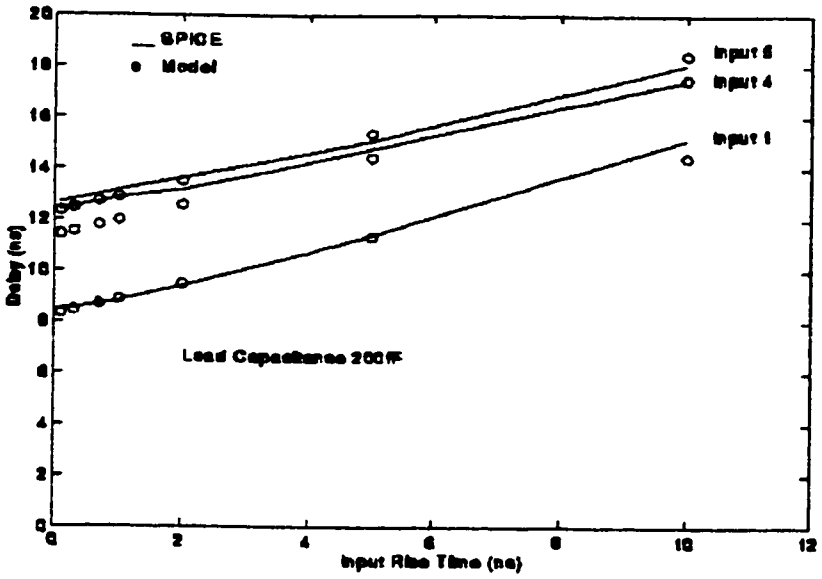


**Fig. 6.15: Delay vs. input fall time for different trigger input, load capacitance *200fF*.**

## 6.4 Discussion

In this chapter, we have compared the model and SPICE simulation on long-channel MOSFET circuits. A 5-input NAND gate and 5-input NOR are taken as examples to examine the validity of the model on series-connected pull-down and pull-up MOSFET network in long-channel regime. The overall results are in accord with SPICE. The load capacitance compensation factor and slope effect adjustment factor are obtained using the same method as shown in short-channel devices.

# CHAPTER 7 Circuit Analysis and Optimization Program

## 7.1 Introduction

In CHAPTER 5 and CHAPTER 6, we have proved the validity of the analytical delay model for combinational CMOS circuits in both short-channel and long-channel regime. CMOS circuits with various complexity are simulated under different circuit parameters, which show a good agreement between the model and SPICE. Based on the transistor level analytical delay model, a circuit analysis and optimization program is developed. The objective of the program is to help designers choose the optimal circuit topology through worst delay path sizing without going through extensive circuit simulations.

For a given multiple-output Boolean expression, several CMOS circuit implementations may be possible. Each implementation represents a different circuit topology and transistor sizing. For each circuit topology, different sizing may result in different performance. Finding the best circuit implementation which meets all the design criteria may become a tedious task for designers. Our program provides a solution to minimizing the effort of designer on finding an optimal circuit implementation. The program reads in the Boolean expression in BLIF format from SIS: a logic synthesis and minimization system where multiple-level implementations of a function is explored. Each Boolean expression is realized in the program as a complementary CMOS circuit. The program analyzes the performance of each circuit of different topology, sizes the longest delay path using the area constraints. Several attributes are computed analytically for each implementation - delay ($T$), rise and fall time, dynamic power, area ($A$), $AT$, $AT^2$ and power-delay product. These circuit attributes forms a decision matrix which is fed to a Multiple Attribute Decision Making (MADM) process to find the optimal circuit topology and implementation according to the design criteria.

## 7.2 Circuit Implementation

A given logic function may be implemented by a number of possible CMOS structures. The chosen structure, along with the technology models, determines the circuit performance - delay, power consumption, and area, etc. In our program, we implement circuits using complementary CMOS.

The topology of each circuit can be represented by its Boolean expression. Each factorized form of the multiple-output Boolean expression represents a different circuit topology. A logic function may have several factorized Boolean expressions. Two different Boolean expressions for a full adder are shown in Fig.7.1, where $A$, $B$, $C_i$ are the primary inputs, *sum* and $C_O$ are the primary outputs, and $D$ and $F$ are intermediate variables.

| | |
|---|---|
| $sum = D\overline{C_o} + ABC_i$ | $sum = C_i\overline{C_o} + \overline{C_o}F + ABC_i$ |
| $C_o = AB + AC_i + BC_i$ | $C_o = AB + C_iF$ |
| $D = A + B + C$ | $F = A + B$ |
| Expression 1 | Expression 2 |

**Fig. 7.1: Boolean expressions for a full-adder.**

The complementary CMOS implementations for expression 1 and 2 are shown in Fig.7.2 and Fig.7.4 respectively in the form of sum of products. Each circuit can be divided into modules or stages with different levels. Fig.7.3 and Fig.7.5 show the stage assignment of the circuits using the strategy illustrated in SECTION 2.3.

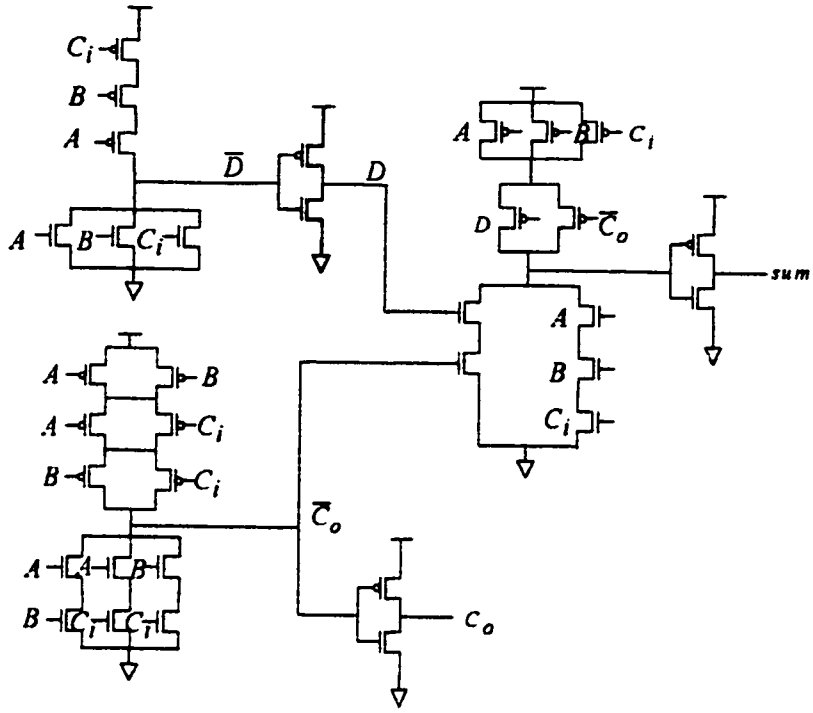


**Fig. 7.2: CMOS representation of expression 1.**

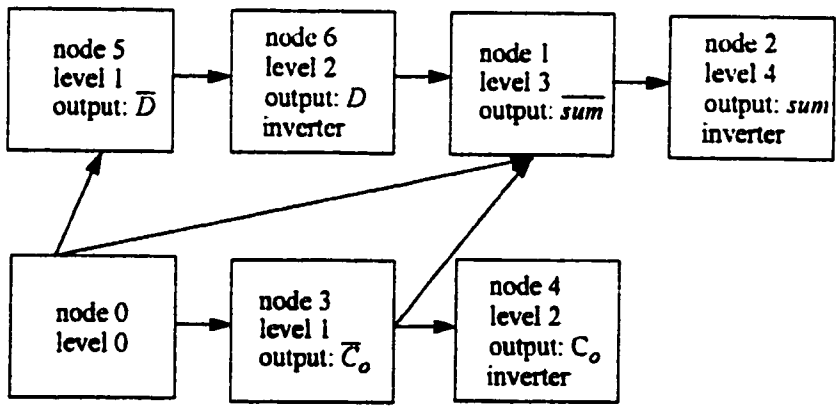


**Fig. 7.3: Stage assignment of Fig.7.2.**

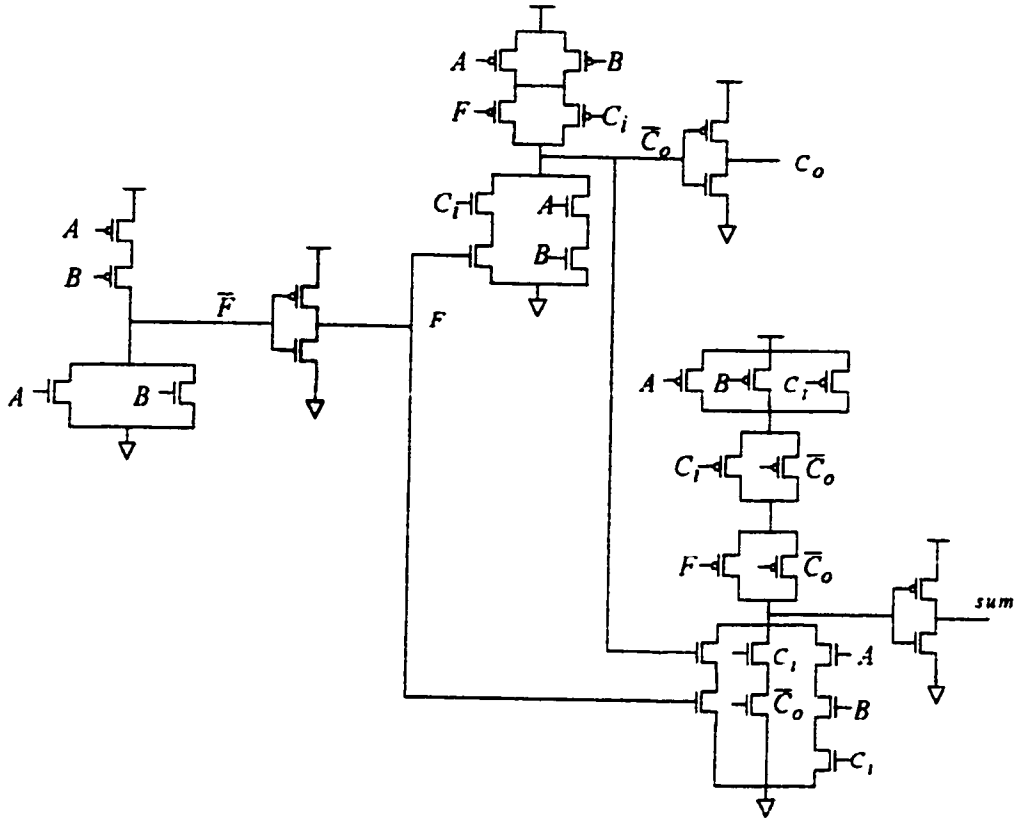


**Fig. 7.4: CMOS representation of expression 2.**

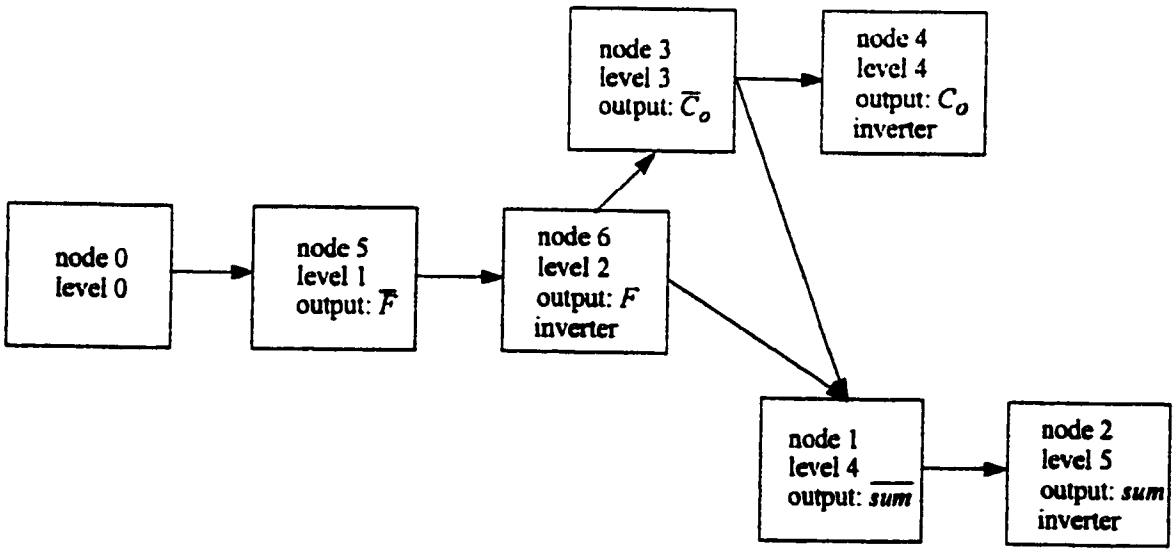


**Fig. 7.5: Stage assignment of Fig.7.4.**

## 7.3 Circuit Collapsing

For delay time optimization, worst-case delay has to be considered. Maximum delay occurs when all series transistors are switching simultaneously and if only one of the parallel devices is activated [1]. We collapse the circuit according to the worst-case scenario in the circuit analysis and optimization program.

The longest path in the pull-up and pull-down network is considered for the worst-case delay performance. In the pull-up network section, only the transistors with the smallest $W_p$ is considered to be on. Similarly in the pull-down network, only the branch with the lowest $W_{eqn}$ is considered to be on. All the other transistors in the pull-up and pull-down network are considered to be off. This approach guarantees the worst path for charging and discharging the load capacitance at the output node. However, it may also over-estimate the delay since the combination of ON transistors may not be realistic. Fig.7.6 shows the worst delay path for collapsing in one stage of the full adder circuit.

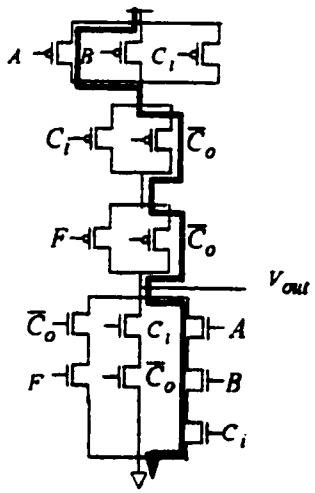


**Fig. 7.6: Worst delay path for collapsing.**

## 7.4 Circuit Simulation

Circuit analysis is performed on the stages of a collapsed CMOS circuit using the analytical delay models illustrated in CHAPTER 3 and CHAPTER 4 for inverter and non-inverter stages in both short-channel and long-channel. The procedures are described in SECTION 2.3. Several circuit attributes are computed.

- Area

Although chip size depends qualitatively on the placement of the subcells, total active area is a common measure for chip area in transistor sizing algorithms [1]. The active area increases linearly with the gate widths in the circuit. It has to be adjusted each time an optimization sizing is performed.

- Delay

The high to low delay $t_{dHL}$ and low to high $t_{dLH}$ are calculated as the difference between 50% $V_{DD}$ level of the input and output. Each stage in the circuit has its own delay. The delay figure at each primary output represents the cumulative delay of the worst path leading to that output. The worst delay node is selected by comparing the delay at all the primary output nodes.

- Rise time and Fall time

The effective rise time and fall time of a stage is calculated using the slope of the output voltage $V_O$ at half $V_{DD}$ point as illustrated in SECTION 3.8.

- Dynamic Power Consumption

To estimate the dynamic power dissipation $P_d$ for a fixed operating frequency $f$ of a CMOS circuit, all capacitances in the data paths of the circuit are summed up, i. e.

$$P_d = \sum C_L V_{DD}^2 f$$

where we take $f = \frac{4}{t_r + t_f}$, $t_r$ and $t_f$ are the rise and fall time of the worst delay output node.

- AT

The delay used in calculating the area-delay product is obtained from the average of the two delays at the worst delay node, i.e. $T = \frac{t_{dHL} + t_{dLH}}{2}$.

Other circuit attributes computed also include $AT^2$ - the area -delay$^2$ product and $P_dT$ - the power-delay product.

### 7.5 Selection of Longest Delay Path and Transistor Sizing

The actual delay of a circuit is defined to be the delay of its longest sensitizable path which is bounded by that of its longest path. Therefore, the performance of a circuit can be optimized if all its long paths can be shortened not longer than a given value $\tau$ [24]. To determine the longest delay path of the network, the worst primary output node is found first and added to the delay path. All the fan-ins of the worst delay output node are then examined. The fan-in with the worst delay is chosen as the next node in the worst delay path. This process is repeated recursively until stage 0 is reached.

Many techniques can be used to improve the performance of a circuit. At the structural level, the internal constructs of gates and their interconnections in a circuit are modified to improve circuit performance. At the topological level, performance-driven placement of gates and performance-driven routing of wires are aimed at minimizing the delay of the longest paths. At the physical level, techniques of transistor sizing and buffering are used to improve gate speed while the topology of the whole circuit is retained [24]. In our program, we assume that the circuits are first optimized at topological level by a logic synthesis and minimization system - SIS. Transistor sizing technique is then used to further improve circuit performance by reducing the delay on the longest delay path.

In our approach, the widths of all the transistors on the worst delay path are sized by a factor of $W_{min}$. The circuits are re-collapsed and capacitances for the whole network are readjusted after

the sizing. Circuit analysis and simulation are performed on the sized network and a new set of attributes are computed. This process is continued until the area constraint is met. Note that for different simulation runs, different delay paths have to be optimized. This is due to the fact that sizing the transistors on one signal path may slow down another, which results from the capacitive loading effects of the path interactions.

## 7.6 Multiple Attribute Decision Making (MADM) Process

### 7.6.1 Decision Matrix

The basis of circuit design is making trade-offs among competing factors or attributes such as chip area, propagation delay and power consumption. The multiple attribute decision making problem can be expressed in a decision matrix format. A decision matrix $D$ is $n$ by $m$ matrix containing $m$ alternatives associated with $n$ attributes,

$$D = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1j} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2j} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & x_{i2} & \cdots & x_{ij} & \cdots & x_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mj} & \cdots & x_{mn} \end{bmatrix}$$

where $x_{ij}$ is the *ith* alternative with respect to the *jth* attribute. The *ith* alternative in the decision matrix is denoted as $A_i$, and

$$A_i = (x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots x_{in})$$

### 7.6.2 Algorithm

The Technique for Order Preference by Similarity to the Ideal Solution (TOPSIS) is developed in [25] to choose the optimum alternative using the decision matrix D. The algorithm is

based on the concept that the chosen alternative should have the shortest distance from the ideal solution and the farthest from the negative ideal solution.

Since all the circuit attributes may not be of equal importance, a weight matrix $W = (w_1, w_2, \ldots, w_j \ldots w_n)$ is introduced, where $\sum_{j=1}^{n} w_j = 1$. If we denote the normalized decision matrix as R where an element $r_{ij}$ is calculated as

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}} \tag{7.1}$$

the normalized weighted decision matrix may be obtained as

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1j} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2j} & \cdots & v_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ v_{i1} & v_{i2} & \cdots & v_{ij} & \cdots & v_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ v_{m1} & v_{m2} & \cdots & v_{mj} & \cdots & v_{mn} \end{bmatrix} = \begin{bmatrix} w_1 r_{11} & w_2 r_{12} & \cdots & w_j r_{1j} & \cdots & w_n r_{1n} \\ w_1 r_{21} & w_2 r_{22} & \cdots & w_j r_{2j} & \cdots & w_n r_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ w_1 r_{i1} & w_2 r_{i2} & \cdots & w_j r_{ij} & \cdots & w_n r_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ w_1 r_{m1} & w_2 r_{m2} & \cdots & w_j r_{mj} & \cdots & w_n r_{mn} \end{bmatrix} \tag{7.2}$$

Let the two alternatives $A^I$ and $A^{NI}$ be defined as the most preferable alternative (ideal solution) and the least preferable alternative (negative-ideal) solution respectively,

$$A^I = \{v_1^I, v_2^I, \ldots, v_n^I\} \tag{7.3}$$

$$A^{NI} = \{v_1^{NI}, v_2^{NI}, \ldots, v_n^{NI}\} \tag{7.4}$$

The distance of each alternative from the ideal solution and the negative-ideal solution is given by [25],

$$S_i^I = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^I\right)^2}\Bigg|_{i=1,2,\ldots,m} \tag{7.5}$$

$$S_i^{NI} = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^{NI}\right)^2}\Bigg|_{i=1,2,\ldots,m} \tag{7.6}$$

where $v_{ij}$ is the element in the weighted normalized decision matrix. The closeness of an alternative to the ideal solution is measured by

$$C_i^I = \frac{S_i^{NI}}{S_i^{NI} + S_i^I} \tag{7.7}$$

It is clear that $0 \le C_i^I \le 1$, $i = 1, 2, \ldots, m$. When $A_i = A^I$, $C_i^I = 1$ and $C_i^I = 0$ if $A_i = A^{NI}$. The more $C_i$ approaches to 1, the closer an alternative $A_i$ is to the ideal solution. A set of alternatives can now be ranked according to the descending order of $C_i^I$. The alternative with $C_i^I$ closest to unity is the alternative closest to the ideal solution and therefore that alternative should be chosen as the optimum between all the other alternatives in the solution space given the attribute weights.

### 7.6.3 Solution Space

In our program, the solution space is generated from the circuit attributes from different factorized forms of the Boolean expression and different transistor sizings. The circuit closest to the ideal-solution is chosen as the best alternative. This alternative may not be the absolute optimum solution, however, it is the optimum alternative in the solution space.

## 7.7 Implementation in C

### 7.7.1 Data Structure

Each input BLIF file can be translated into a data structure shown in Fig.7.7 [26].

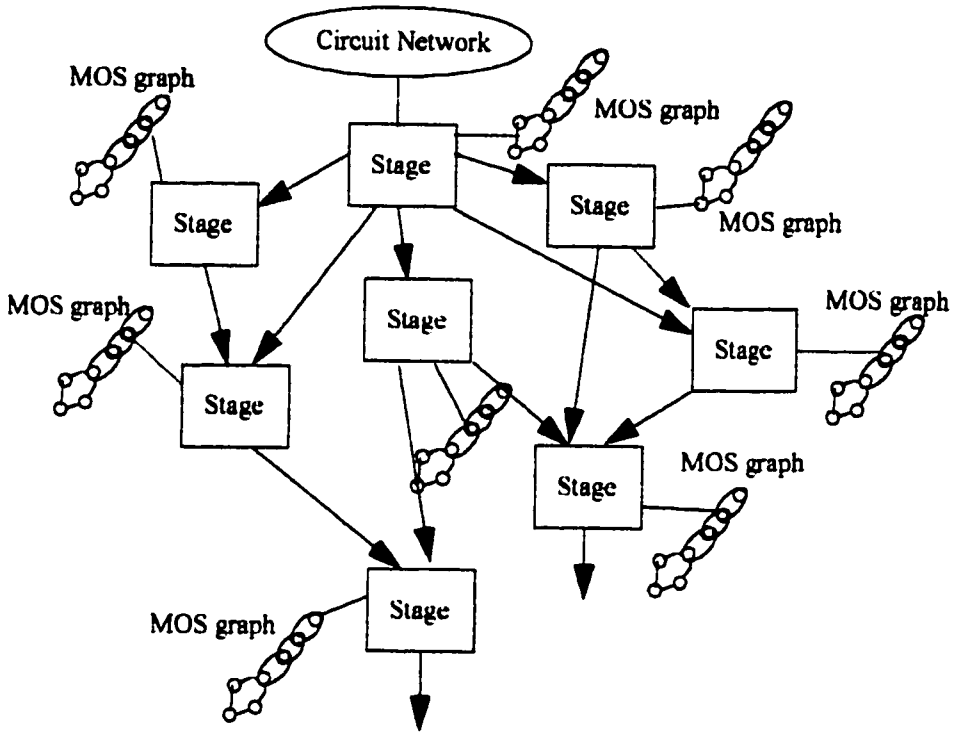


**Fig. 7.7: Data Structure.**

The Circuit Network contains circuit attributes and general data for all the stages in the circuit network. These include,

- BLIF file model name
- primary I/O
- worst output
- worst path
- number of stages in the network
- total area
- dynamic power

- total number of transistors
- threshold in saturation region
- threshold in linear region

A circuit network can be divided into stages. The data for each sum of products section of the circuit are grouped into a stage. A stage contains

- list of I/O
- list of fan-in and fan-out
- type of stage: inverter or sum-of-products
- MOS structure
- stage level and node number
- longest path to ground
- transistor size
- gate, drain, output node and load capacitance
- input waveform and output waveform
- worst input node

Each stage has its MOS graph which is a representation of the actual NMOS and PMOS transistor structures. A MOS transistor may be represented by an edge which contains source node, destination node, gate variable and the state of the variable in term of inverted or non-inverted input.

### 7.7.2 Flow Chart

The flow chart of the program is shown in Fig.7.8 (a). It starts by reading the design parameters and mapping the BLIF file into circuit network. In each sum of product stage, a longest path from power supply to ground is found and collapsed into a four-transistor network. Circuit simulation is done for each stage and the longest delay path to output is found.

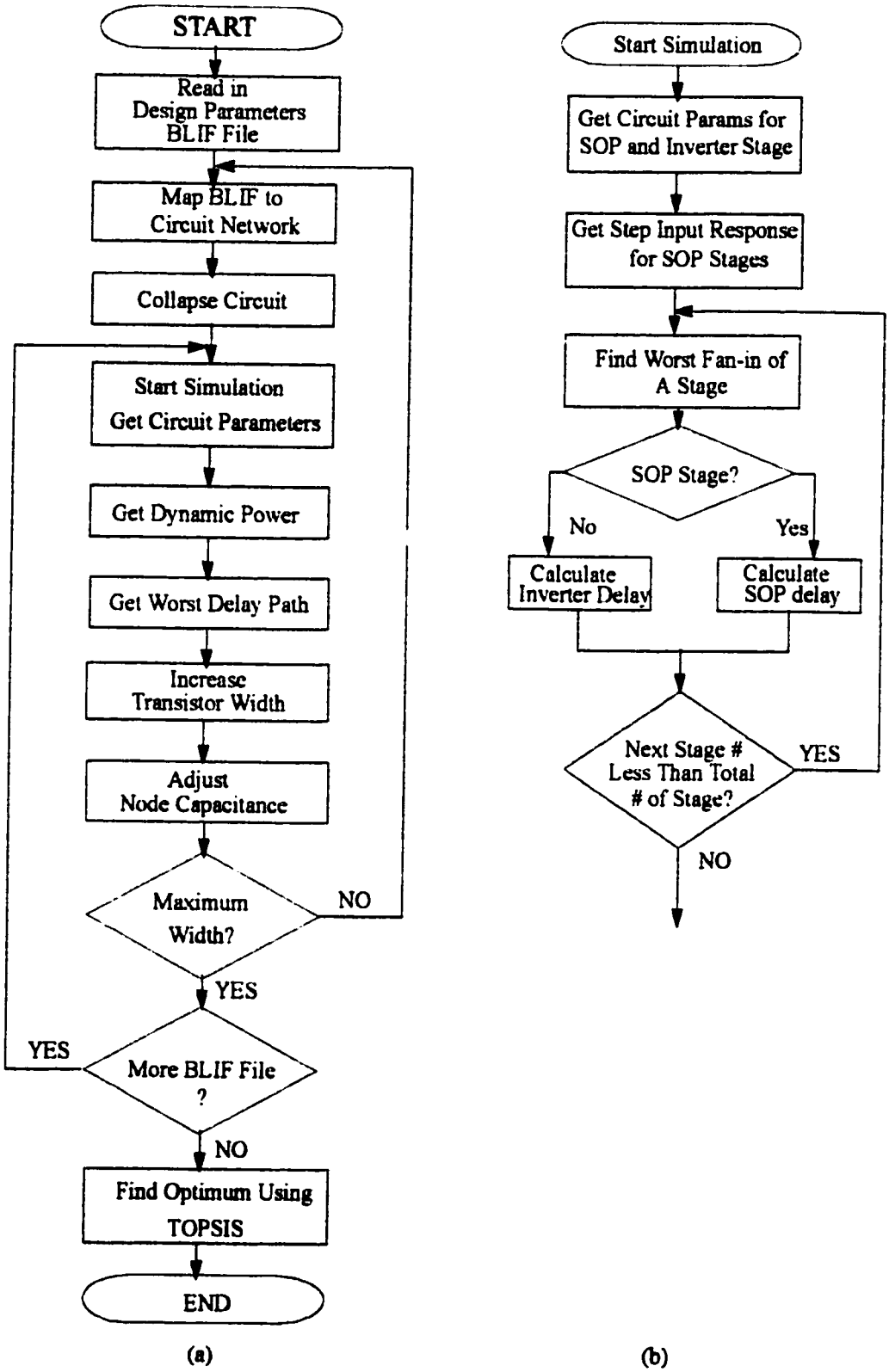


**Fig. 7.8: (a) Flow chart of program. (b) Flow chart of start simulation block.**

Sizing the transistor width along the longest delay path, we repeat above procedures to estimate circuit performance until the area criteria is met. A solution space is formed using simulation data on different circuit topologies represented by different BLIF files. An optimum design is obtained using TOPSIS. Fig.7.8 shows a detailed flow chart for the Start Simulation block. General circuit parameters for each SOP and inverter stage are first calculated. Step input response is obtained for each SOP stage. Starting from level 1 stage where all the inputs of the stage are primary inputs, delay with input ramp effect is calculated. For each successive stage, a worst fan-in node is found by comparing the delays at the output. This output node may become the fan-in of the next stage and the procedures go on until the primary output is reached.

## 7.8 Simulation Results

In this section, some simulation results from the program are presented. The simulations are performed on the two factorized forms of the adder. Circuit 1 refers to the topology in Fig.7.2 and circuit 2 refers to the topology in Fig.7.4. For each circuit three simulation iterations were performed. Two load capacitances are use: *100fF*, and *500fF*.

**TABLE 7. Simulation result for a full adder with 100fF load capacitance.**

| Circuit | Area $\mu m^2$ | $t_{dHL}$ (ns) | $t_{dLH}$ (ns) | $t_{pf}$ (ns) | $P_d$ (mW/ GHz) | trans count | AT $\mu m^2 \cdot ns$ | $AT^2$ $\mu m^2 \cdot ns^2$ | $P_dT$ (mW/ GHz.ns) | literal count |
|---|---|---|---|---|---|---|---|---|---|---|
| 1a | 293.8 | 0.731 | 0.377 | 0.142 | 9.3 | 34 | 162.7 | 90.2 | 5.15 | 6 |
| 1b | 397.4 | 0.582 | 0.449 | 0.095 | 10.6 | 34 | 204.9 | 105.6 | 5.5 | 6 |
| 1c | 501.1 | 0.511 | 0.527 | 0.078 | 11.9 | 34 | 260.0 | 134.9 | 6.2 | 6 |
| 2a | 276.5 | 0.685 | 0.604 | 0.143 | 10.5 | 32 | 178.2 | 114.8 | 6.75 | 6 |
| 2b | 406.1 | 0.528 | 0.720 | 0.095 | 12.8 | 32 | 253.3 | 158.0 | 8 | 6 |
| 2c | 535.7 | 0.593 | 0.530 | 0.079 | 15.1 | 32 | 300.6 | 168.6 | 8.5 | 6 |

Table 7 presents the simulation results for the full adder circuit with topologies shown in Fig.7.2 and Fig.7.4. The load capacitance at the output node is *100fF*. If we take the weight matrix

as [0.05 0.1 0.1 0.35 0.1 0.05 0.05 0.05 0.1 0.05] associated with each attribute, the order of the circuit close to the optimum solution is *1c, 1b, 2c, 2b, 1a, 2a*. From the result we can see that under this weight matrix, circuit 1 performs better than circuit 2 although it has more transistor count. It can also be seen that transistor sizing has important impact on the performance of the circuits. Although circuit *1c* and *1b* outperform circuit 2, their un-sized version has worst performance than the sized circuit *2c* and *2b*.

**TABLE 8. Simulation result for a full adder with load capacitance 500fF.**

| Circuit | Area $\mu m^2$ | $t_{dHL}$ (ns) | $t_{dLH}$ (ns) | $t_{rf}$ (ns) | $P_d$ (mW/ GHz) | transc ount | $AT$ $\mu m^2 \cdot ns$ | $AT^2$ $\mu m^2 \cdot ns^2$ | $P_dT$ (mW/ GHz.ns) | literal count |
|---|---|---|---|---|---|---|---|---|---|---|
| 1a | 293.8 | 1.308 | 0.514 | 0.304 | 29.3 | 34 | 267.6 | 243.8 | 26.69 | 6 |
| 1b | 397.4 | 0.881 | 0.610 | 0.199 | 30.6 | 34 | 296.4 | 221.0 | 22.82 | 6 |
| 1c | 501.1 | 0.892 | 0.551 | 0.304 | 31.9 | 34 | 361.6 | 261.0 | 23.02 | 6 |
| 2a | 276.5 | 1.262 | 0.950 | 0.670 | 30.5 | 32 | 305.8 | 338.2 | 33.7 | 6 |
| 2b | 406.1 | 0.827 | 0.897 | 0.199 | 32.8 | 32 | 350.0 | 301.7 | 28.3 | 6 |
| 2c | 535.7 | 0.891 | 0.769 | 0.353 | 35.1 | 32 | 444.6 | 369.0 | 29.1 | 6 |

The simulation results for the full adder circuit with load capacitance *500fF* are shown in Table 8. With the same weight matrix, the order of the circuit close to the optimum solution is *1b, 2b, 2a, 1c, 1a* and *2c*, which is different from the circuit with *100fF* load capacitance. Depending on different load capacitance, the topology and sizing of the optimum circuit may vary.

The output files generated by the program on circuit 1 and 2 with *100fF* load capacitance are in Appendix B.

## 7.9 Conclusion

In this chapter, we can see that our circuit simulation and optimization program lays the ground for a more complete and accurate simulation/optimization tool at circuit level. The software package starts from reading a Boolean description of a circuit. Circuits with different topol-

ogies can be analyzed and sized. The optimum design may be chosen using the MADM technique among different circuit topologies and transistor sizings.

# CHAPTER 8 Conclusion

Optimizing a design at circuit level is a tedious task which means time consuming circuit simulation on designs with different circuit topologies and transistor sizings. Finer structure sizes in up-to-date technologies allow for integrating more and more functions on a single chip, which on the other hand, makes design optimization almost impossible without proper design automation tools.

Accurate modeling of signal path delays is of particular importance in optimizing high-speed integrated circuits. Although circuit level simulators like SPICE produce accurate and detailed delay information, analytical delay models are required in general because of the time consuming computation in SPICE simulations. The reported delay models for submicron regime design have limitations and they may introduce suboptimality in circuit optimization. In our work, an accurate analytical delay model is developed to estimate the circuit performance while at the meantime significantly reduces the simulation time. An event-driven simulation is carried out on complex CMOS circuits at stage level. The worst delay path are found for each circuit and transistor sizings along the path are performed. Circuit attributes which characterize the performance of the designs with different implementations are obtained. They make up a decision matrix. The optimum circuit design is found through the MADM process given a decision weight matrix.

## 8.1 Collapsing Technique

A new collapsing strategy is presented to model the delay of a circuit. A complex CMOS circuit is divided into stages. Each non-inverter stage is modeled by two pull-down and pull-up transistors according to their operating characteristics. Maximum information is passed to the collapsed circuit by preserving the same initial circuit charge, node capacitance and input waveform. A collapsing scheme for worst delay scenario is used in our design optimization program to

reduce simulation time and achieve circuit optimization. The proposed technique is appropriate for complementary CMOS logic, but also gives good estimation for a general CMOS structure such as the parity generator circuit shown in SECTION 5.7.

## 8.2 Analytical Delay Models

Analytical delay models in both short-channel and long-channel are presented. For a non-inverter stage, a detailed piece-wise linear model for output waveform is derived. Circuit performance are estimated based on the model. Different factors that affect the delay of the circuit are taken into account. The effects of the circuit topology, node capacitances, trigger position are considered. Input slope and short-channel effects are modeled. The delay modeling based on the worst case scenario is used in our circuit optimization program. For an inverter stage in submicron regime, we use an equivalent input waveform to model the ramp input. Both slow input and fast input are modeled. The delay analysis is simplified and a good accuracy on delay estimation is achieved.

Our approach compares favorably with SPICE Level 3 on the delay estimation. Detailed comparisons between our model and SPICE are performed on different circuits. The model shows an overall less than 10 percent deviation from SPICE. The event driven simulation achieves a significant speed-up compared to SPICE.

## 8.3 Circuit Simulation and Optimization Program

A circuit simulation and optimization program is developed based on the analytical delay models. The program reads in circuit descriptions in Boolean expressions and maps the descriptions into complementary CMOS implementations. Stage assignments are performed so that an event driven simulation can be carried out. Each non-inverter stage of the circuits is collapsed into an equivalent four-transistor structure. Circuit analysis and simulation are performed on the col-

lapsed equivalent circuits. A longest delay path from primary input to output is found. Transistor sizing is carried out on the longest delay path until the maximum area constraint is reached. For each sizing cycle, the circuit is re-collapsed and the longest delay path is re-selected. The attributes of circuits with different topologies and sizings are fed into a MADM program where an optimum design is acquired for certain design weight matrix. A full adder circuit with two implementation topologies and different load capacitance is tested through the optimization program. Optimum designs are selected in no time among 12 design options. The order of the circuit close to the optimum design is also given. This work lays ground for the development of a complete and accurate design automation tool at circuit level.

### 8.4 Conclusion

Combining the proposed collapsing technique with the analytical delay models, a CMOS circuit simulation and optimization program is developed. The objective of the program is to provide designer first hand information on selecting circuit topology and transistor sizing without extensive and expensive simulations. The work may be extended to general CMOS circuits with pass logics in the future work.

# Reference


[1] B. Hoppe, G. Neuendorf, D. Shmitt-landsiedel, and W. Specks, "Optimzation of high-speed CMOS logic circuits with analytical models for signal delay, chip area, and dynamic power dissipation," IEEE Trans. on Computer-Aided Design. Vol. 9, No. 3, March 1990.

[2] M. D. Matson and L.A. Glasser, "Macromodeling and optimization of digital MOS VLSI circuits," IEEE Trans. Computer-Aided Design, vol. CAD-5, pp. 659-678, Oct. 1986.

[3] M. D. Maison, "Macromodeling of digital MOS VLSI cirucits," in Proc. 22nd Design Automation Conf., pp. 144-151, 1985.

[4] J. Shyu, A. Sangiovanni-Vincentelli, J. P. Fishburn and A. E. Dunlop, "Optimization-based transistor sizing," IEEE J. Solid-State Circuits, vol. 23, pp. 400-409, Apr. 1988.

[5] J. P. Fishburn and A. E. Dunlop, "TILOS: A posynomial programming approach to transistor sizing, " in Proc. 1985 IEEE Conf. Computer-Aided Automation Conf., pp. 326-328, 1985.

[6] L. A. Glasser and L. P. J. Hoyte, "Delay and power optimization in VLSI circuits," in Proc. 21st Design Automation Conf., pp.529-535, 1984.

[7] K. S. Hedlund, "Aesop: A tool for automated transistor sizing," in Proc. 24th Design Automation Conf., 114-120, 1987.

[8] D. P. Marple and A. E. Gammal, "Optimal selection of transistor sizes in digital MOS VLSI circuits," in Advanced Research in VLSI Proc. 1987 Standard Conf., Stanford, CA, pp. 151-172, 1987.

[9] J. K. Ousterhout, "A switch-level timing verifier for digital MOS VLSI," IEEE Transactions on Computer-Aided Design, vol. CAD-4, No. 3, July 1985.

[10] T. Sakurai and A. R. Newton, "Delay analysis of series-connected MOSFET circiuits,"

IEEE Journal of Solid-State Circuits, vol. 26, No. 2, February 1991.

[11] S. M. Kang and H.Y. Chen, "A global delay model for domino CMOS circuits with application to transistor sizing," International Journal of Circuit Theory and Applications, vol. 18, p. 289-306, 1990.

[12] K.Toh, P. Ko and R.G. Meyer, "An engineering model for short-channel MOS devices," IEEE Journal of Solid Circuits, vol. 23, p. 950-958, August 1988.

[13] F. Rouatbi, "Supply current estimation in CMOS VLSI circuits," MA. Sc thesis, Concordia University, Montreal, Canada, June 1992.

[14] C. Huang and K. S. Yoon, Multiple Attribute Decision Making, Methods and Applications, A state of the Art Survey, Springer Verlag.

[15] N.H.E. Weste, K. Eshraghian, Principle of CMOS VLSI design: A System Perspective. Addison-Wesley, p. 216, 1993.

[16] J. M. Rabaey, Digital Integrated Circuits, A Design Perspective, Pentice-Hall, Inc. p. 50-55, 1996.

[17] C. G. Sodini, P. Ko, and J. L. Moll, "The effect of high fields on MOS device and circuit performance," IEEE Tran. on Electron Devices, vol. ED-31, NO. 10, p. 1386-1393, October 1984.

[18] J. E. Chung, M. Jeng, J. E. Moon, P. Ko, and C. Hu, "Performance and reliability design issue for deep-submicronmeter MOSFET's," IEEE Tran. on Electron Devices, vol. 38. No. 3, p. 545-553, 1991.

[19] K. Toh, P. Ko, and R. G. Meyer, "An Engineering model for short-channel MOS device," IEEE Journal of Solid-State Circuits, vol. 23, No. 4, August, 1988.

[20] N. Hedenstierna and K. O. Jeppson, "CMOS circuit speed and buffer optimization," IEEE trans. on CAD, vol. CAD-6, No. 2, March 1987.

[21] A. I. Kayssi, K. A. Sakallah, and T. N. Mudge, "The impact of signal Transition time on

path delay computation," IEEE Trans. on Circuits and Systems - II: Analog and Digital Signal Processing, vol. 40, No. 5, May 1993.

[22] J. Rubinstein, P. Penfield, and M. A. Horowitz, "Signal delay in RC tree Networks," IEEE Trans. on CAD, vol. CAD-2, No. 3, July 1983.

[23] R. K. Brayton, R. Rudell, A. Sangiovanni-Vincentelli and A. R. Wang, "MIS: a multiple-level logic optimization system," IEEE Transactions on Computer-Aided Design, vol. CAD-6, No. 6, November 1987.

[24] H. C. Chen, D. Du and L. Liu, "Critical path selection for performance optimization", IEEE Trans. on CAD, vol. 12, No. 2, pp. 185-195, February 1993.

[25] C. Huang and K. S. Yoon, Multiple Attribute Decision Making, Method and Applications, A State of the Art Survey, Springer Verlag.

[26] K. Kechichian, "Optimizing CMOS combinational circuits using mulitple attribute decision making techniques", A Project Report for the Degree of Master of Engineering in Concordia University, May 1994.

## APPENDIX A Derivation of $V_1(t_s)$

In segment I of case 1, $N_1$ is in saturation and $N_{equ}$ is in linear region. The node equations of the equivalent circuit in Fig.4.1 can be written as follows,

$$C_1\frac{dV_1(t)}{dt} = -I_{d1} = -\frac{\beta_1}{2}[V_e - V_2(t)]^2 \qquad \text{(EQ. A-1)}$$

$$C_{equ}\frac{dV_2(t)}{dt} = I_{d1} - \beta_{equ}\left(V_2(t)V_c - \frac{1}{2}V_2^2(t)\right) \qquad \text{(EQ. A-2)}$$

We can not derive a closed-form equation for $V_1(t)$ using EQ.A-1. However, substituting $I_{dl}$ in EQ.A-2, and performing EQ.A-1/EQ.A-2, we have,

$$\frac{dV_1}{dV_2} = \frac{A(V_e - V_2)^2}{(V_2 - a_1)(V_2 - a_2)} \qquad \text{(EQ. A-3)}$$

where

$$A = -\frac{C_2 n}{C_1(n+1)}$$

$$n = \frac{\beta_1}{\beta_{equ}}$$

$$a_1, a_2 = \frac{nV_e + V_c}{n+1} \pm \left[\left(\frac{nV_e + V_c}{n+1}\right)^2 - \frac{nV_c^2}{n+1}\right]^{\frac{1}{2}}$$

The closed-form solution for $V_1(t)$ is,

$$V_1(t) = A\left(\left(\frac{V_e - a_1}{a_1 - a_2}\right)^2 \ln\left|\frac{V_2(t) - a_1}{V_2(0) - a_1}\right| - \left(\frac{V_e - a_2}{a_1 - a_2}\right)^2 \ln\left|\frac{V_2(t) - a_2}{V_2(0) - a_1}\right| + V_2(t) - V_2(0)\right] + V_{DD} \qquad \text{(EQ. A-4)}$$

If we take $V_2(t) = V_{2ss}$, we have $V_1(t_s)$ as,

$$V_1(t) = A\left(\left(\frac{V_e - a_1}{a_1 - a_2}\right)^2 \ln\left|\frac{V_{2ss} - a_1}{V_2(0) - a_1}\right| - \left(\frac{V_e - a_2}{a_1 - a_2}\right)^2 \ln\left|\frac{V_{2ss} - a_2}{V_2(0) - a_1}\right| + V_{2ss} - V_2(0)\right] + V_{DD} \qquad \text{(EQ. A-5)}$$

## APPENDIX B Simulation Results from The Program

The following is the output results of the program for a full adder circuit with load capacitance 100fF. Circuit 1 refers to the circuit topology shown in Fig.7.2. Circuit 2 refers to the topology in Fig.7.4. For each circuit, three width increment iterations are performed. The worst output node and worst delay path are given in each circuit. Each stage is represented by stage node number and level number. Its inputs and output are given. Fan-in and fan-out node are listed. The output waveform of each stage is characterized by its rise or fall time, offset, cumulative delay and its worst fan-in node.

Circuit 1a

```
worst output node 4
worst path
(1,1)
(2,2)
(5,3)
(3,4)
(4,5)
---------------------------------------
node nb=1   level=1
inputs :a b
output : f
fanin (node nb,level):
(0,0)
fanout (node nb,level):
(2,2)
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=3.083112e-014
rise time=2.000000e-011   offset = 0.000000e+000   td=2.391239e-011  from node 0
fall time=0.000000e+000  offset = 0.000000e+000   td=1.283263e-010  from node 0
---------------------------------------

---------------------------------------
node nb=2   level=2
inverter
inputs :
output : f
fanin (node nb,level):
(1,1)
fanout (node nb,level):
(3,4)
(5,3)
```

```
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=4.937256e-014
fall time=9.896685e-012  offset = 0.000000e+000   td=7.856942e-011  from node 1
rise time=2.517438e-010  offset = 2.454373e-012   td=1.817701e-010  from node 1
-------------------------------------

-------------------------------------
node nb=5   level=3
inputs :f cin a b
output : cout
fanin (node nb,level):
(2,2)
(0,0)
fanout (node nb,level):
(3,4)
(6,4)
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=7.804513e-014
rise time=4.323812e-011  offset = 0.000000e+000   td=1.144685e-010  from node 2
fall time=8.705485e-011  offset = 2.454373e-012   td=3.299724e-010  from node 2
-------------------------------------

-------------------------------------
node nb=3   level=4
inputs :f cin a cout b
output : sum
fanin (node nb,level):
(5,3)
(2,2)
(0,0)
fanout (node nb,level):
(4,5)
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=4.461768e-014
fall time=1.601643e-010  offset = 3.438631e-011   td=4.892432e-010  from node 5
rise time=5.316798e-010  offset = 6.413254e-011   td=4.920720e-010  from node 5
-------------------------------------

-------------------------------------
node nb=6   level=4
inverter
inputs :
output : cout
fanin (node nb,level):
(5,3)
fanout (node nb,level):
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=1.079726e-013
fall time=1.601643e-010  offset = 3.438631e-011   td=3.096706e-010  from node 5
rise time=5.316798e-010  offset = 6.413254e-011   td=4.474026e-010  from node 5
-------------------------------------

-------------------------------------
node nb=4   level=5
```

```
inverter
inputs :
output : sum
fanin (node nb,level):
(3,4)
fanout (node nb,level):
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=1.079726e-013
rise time=6.191546e-010  offset = 1.796659e-010   td=6.035729e-010  from node 3
fall time=1.527369e-010  offset = 4.157035e-010   td=6.854402e-010  from node 3
---------------------------------------
Circuit 1b

worst output node 4
worst path
(1,1)
(2,2)
(5,3)
(3,4)
(4,5)
---------------------------------------
node nb=1    level=1
inputs :a b
output : f
fanin (node nb,level):
(0,0)
fanout (node nb,level):
(2,2)
wn =4.800000e-006  wp =9.600000e-006
Cin =3.105000e-014  Cout=4.664832e-014
rise time=2.000000e-011   offset = 0.000000e+000   td=3.313507e-011  from node 0
fall time=0.000000e+000  offset = 0.000000e+000   td=1.114545e-010  from node 0
---------------------------------------

---------------------------------------
node nb=2    level=2
inverter
inputs :
output : f
fanin (node nb,level):
(1,1)
fanout (node nb,level):
(3,4)
(5,3)
wn =9.600000e-006  wp =4.800000e-006
Cin =3.105000e-014  Cout=7.217112e-014
fall time=1.497395e-011  offset = 0.000000e+000   td=1.156313e-010  from node 1
rise time=2.126380e-010  offset = 5.135491e-012   td=1.542602e-010  from node 1
---------------------------------------

---------------------------------------
node nb=5    level=3
inputs :f cin a b
output : cout
```

```
fanin (node nb,level):
(2,2)
(0,0)
fanout (node nb,level):
(3,4)
(6,4)
wn =4.800000e-006  wp =9.600000e-006
Cin =3.105000e-014  Cout=1.096795e-013
rise time=6.320401e-011  offset = 0.000000e+000   td=1.580365e-010  from node 2
fall time=1.028216e-010  offset = 5.135491e-012   td=2.890564e-010  from node 2
--------------------------------------

--------------------------------------
node nb=3   level=4
inputs :f cin a cout b
output : sum
fanin (node nb,level):
(5,3)
(2,2)
(0,0)
fanout (node nb,level):
(4,5)
wn =9.600000e-006  wp =4.800000e-006
Cin =3.105000e-014  Cout=6.126336e-014
fall time=2.186177e-010  offset = 4.872765e-011   td=5.930371e-010  from node 5
rise time=4.062904e-010  offset = 8.591123e-011   td=4.163772e-010  from node 5
--------------------------------------

--------------------------------------
node nb=6   level=4
inverter
inputs :
output : cout
fanin (node nb,level):
(5,3)
fanout (node nb,level):
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=1.079726e-013
fall time=2.186177e-010  offset = 4.872765e-011   td=3.676706e-010  from node 5
rise time=4.062904e-010  offset = 8.591123e-011   td=4.081453e-010  from node 5
--------------------------------------

--------------------------------------
node nb=4   level=5
inverter
inputs :
output : sum
fanin (node nb,level):
(3,4)
fanout (node nb,level):
wn =4.800000e-006  wp =9.600000e-006
Cin =3.105000e-014  Cout=1.134398e-013
rise time=6.325121e-010  offset = 2.767811e-010   td=7.198262e-010  from node 3
fall time=1.198783e-010  offset = 3.564381e-010   td=5.277265e-010  from node 3
```

```
---------------------------------------
Circuit 1c

worst output node 4
 worst path
(1,1)
(2,2)
(5,3)
(3,4)
(4,5)
---------------------------------------
node nb=1   level=1
inputs :a b
output : f
fanin (node nb,level):
(0,0)
fanout (node nb,level):
(2,2)
wn =9.600000e-006  wp =9.600000e-006
Cin =4.140000e-014  Cout=6.119544e-014
rise time=2.000000e-011   offset = 0.000000e+000   td=2.442775e-011  from node 0
fall time=0.000000e+000  offset = 0.000000e+000   td=1.308529e-010  from node 0
---------------------------------------

---------------------------------------
node nb=2   level=2
inverter
inputs :
output : f
fanin (node nb,level):
(1,1)
fanout (node nb,level):
(3,4)
(5,3)
wn =9.600000e-006  wp =9.600000e-006
Cin =4.140000e-014  Cout=9.833833e-014
fall time=1.022481e-011  offset = 0.000000e+000   td=8.109656e-011  from node 1
rise time=2.573315e-010  offset = 2.187118e-012   td=1.863634e-010  from node 1
---------------------------------------

---------------------------------------
node nb=5   level=3
inputs :f cin a b
output : cout
fanin (node nb,level):
(2,2)
(0,0)
fanout (node nb,level):
(3,4)
(6,4)
wn =9.600000e-006  wp =9.600000e-006
Cin =4.140000e-014  Cout=1.345766e-013
rise time=4.462554e-011  offset = 0.000000e+000   td=1.079448e-010  from node 2
fall time=9.000166e-011  offset = 2.187118e-012   td=3.107043e-010  from node 2
```

```
---------------------------------------

---------------------------------------
node nb=3   level=4
inputs :f cin a cout b
output : sum
fanin (node nb,level):
(5,3)
(2,2)
(0,0)
fanout (node nb,level):
(4,5)
wn =9.600000e-006  wp =9.600000e-006
Cin =4.140000e-014  Cout=8.801497e-014
fall time=1.458141e-010  offset = 3.503769e-011   td=4.856334e-010  from node 5
rise time=4.827825e-010  offset = 6.931307e-011   td=4.675109e-010  from node 5
---------------------------------------

---------------------------------------
node nb=6   level=4
inverter
inputs :
output : cout
fanin (node nb,level):
(5,3)
fanout (node nb,level):
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=1.079726e-013
fall time=1.458141e-010  offset = 3.503769e-011   td=2.996038e-010  from node 5
rise time=4.827825e-010  offset = 6.931307e-011   td=4.292339e-010  from node 5
---------------------------------------

---------------------------------------
node nb=4   level=5
inverter
inputs :
output : sum
fanin (node nb,level):
(3,4)
fanout (node nb,level):
wn =9.600000e-006  wp =9.600000e-006
Cin =4.140000e-014  Cout=1.155383e-013
rise time=6.307478e-010  offset = 1.702595e-010   td=5.296544e-010  from node 3
fall time=1.568220e-010  offset = 3.890999e-010   td=5.925116e-010  from node 3
---------------------------------------

Circuit 2a

worst output node 4
 worst path
(1,1)
(2,2)
(5,3)
(3,4)
```

```
(4,5)
---------------------------------------
node nb=1   level=1
inputs :a b
output : f
fanin (node nb,level):
(0,0)
fanout (node nb,level):
(2,2)
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=3.083112e-014
rise time=2.000000e-011  offset = 0.000000e+000   td=2.391239e-011  from node 0
fall time=0.000000e+000  offset = 0.000000e+000   td=1.283263e-010  from node 0
---------------------------------------

---------------------------------------
node nb=2   level=2
inverter
inputs :
output : f
fanin (node nb,level):
(1,1)
fanout (node nb,level):
(3,4)
(5,3)
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=4.937256e-014
fall time=9.896685e-012  offset = 0.000000e+000   td=7.856942e-011  from node 1
rise time=2.517438e-010  offset = 2.454373e-012   td=1.817701e-010  from node 1
---------------------------------------

---------------------------------------
node nb=5   level=3
inputs :f cin a b
output : cout
fanin (node nb,level):
(2,2)
(0,0)
fanout (node nb,level):
(3,4)
(6,4)
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=7.804513e-014
rise time=4.323812e-011  offset = 0.000000e+000   td=1.144685e-010  from node 2
fall time=8.705485e-011  offset = 2.454373e-012   td=3.299724e-010  from node 2
---------------------------------------

---------------------------------------
node nb=3   level=4
inputs :f cin a cout b
output : sum
fanin (node nb,level):
(5,3)
(2,2)
```

```
(0,0)
fanout (node nb,level):
(4,5)
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=4.461768e-014
fall time=1.601643e-010  offset = 3.438631e-011   td=4.892432e-010  from node 5
rise time=5.316798e-010  offset = 6.413254e-011   td=4.920720e-010  from node 5
---------------------------------------

---------------------------------------
node nb=6   level=4
inverter
inputs :
output : cout
fanin (node nb,level):
(5,3)
fanout (node nb,level):
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=5.079726e-013
fall time=1.601643e-010  offset = 3.438631e-011   td=8.863277e-010  from node 5
rise time=5.316798e-010  offset = 6.413254e-011   td=5.843329e-010  from node 5
---------------------------------------

---------------------------------------
node nb=4   level=5
inverter
inputs :
output : sum
fanin (node nb,level):
(3,4)
fanout (node nb,level):
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=5.079726e-013
rise time=6.191546e-010  offset = 1.796659e-010   td=9.498974e-010  from node 3
fall time=1.527369e-010  offset = 4.157035e-010   td=1.262097e-009  from node 3
---------------------------------------
Circuit 2b

worst output node 4
 worst path
(1,1)
(2,2)
(5,3)
(3,4)
(4,5)
---------------------------------------
node nb=1   level=1
inputs :a b
output : f
fanin (node nb,level):
(0,0)
fanout (node nb,level):
(2,2)
wn =4.800000e-006  wp =9.600000e-006
```

```
Cin =3.105000e-014  Cout=4.664832e-014
rise time=2.000000e-011  offset = 0.000000e+000   td=3.313507e-011  from node 0
fall time=0.000000e+000  offset = 0.000000e+000   td=1.114545e-010  from node 0
---------------------------------------

---------------------------------------
node nb=2   level=2
inverter
inputs :
output : f
fanin (node nb,level):
(1,1)
fanout (node nb,level):
(3,4)
(5,3)
wn =9.600000e-006  wp =4.800000e-006
Cin =3.105000e-014  Cout=7.217112e-014
fall time=1.497395e-011  offset = 0.000000e+000   td=1.156313e-010  from node 1
rise time=2.126380e-010  offset = 5.135491e-012   td=1.542602e-010  from node 1
---------------------------------------

---------------------------------------
node nb=5   level=3
inputs :f cin a b
output : cout
fanin (node nb,level):
(2,2)
(0,0)
fanout (node nb,level):
(3,4)
(6,4)
wn =4.800000e-006  wp =9.600000e-006
Cin =3.105000e-014  Cout=1.096795e-013
rise time=6.320401e-011  offset = 0.000000e+000   td=1.580365e-010  from node 2
fall time=1.028216e-010  offset = 5.135491e-012   td=2.890564e-010  from node 2
---------------------------------------

---------------------------------------
node nb=3   level=4
inputs :f cin a cout b
output : sum
fanin (node nb,level):
(5,3)
(2,2)
(0,0)
fanout (node nb,level):
(4,5)
wn =9.600000e-006  wp =4.800000e-006
Cin =3.105000e-014  Cout=6.126336e-014
fall time=2.186177e-010  offset = 4.872765e-011   td=5.930371e-010  from node 5
rise time=4.062904e-010  offset = 8.591123e-011   td=4.163772e-010  from node 5
---------------------------------------

---------------------------------------
```

```
node nb=6   level=4
inverter
inputs :
output : cout
fanin (node nb,level):
(5,3)
fanout (node nb,level):
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=5.079726e-013
fall time=2.186177e-010  offset = 4.872765e-011   td=9.443278e-010  from node 5
rise time=4.062904e-010  offset = 8.591123e-011   td=5.427900e-010  from node 5
--------------------------------------

--------------------------------------
node nb=4   level=5
inverter
inputs :
output : sum
fanin (node nb,level):
(3,4)
fanout (node nb,level):
wn =4.800000e-006  wp =9.600000e-006
Cin =3.105000e-014  Cout=5.134398e-013
rise time=6.325121e-010  offset = 2.767811e-010   td=8.972442e-010  from node 3
fall time=1.198783e-010  offset = 3.564381e-010   td=8.265380e-010  from node 3
--------------------------------------
Circuit 2c

worst output node 4
worst path
(1,1)
(2,2)
(5,3)
(3,4)
(4,5)
--------------------------------------
node nb=1   level=1
inputs :a b
output : f
fanin (node nb,level):
(0,0)
fanout (node nb,level):
(2,2)
wn =9.600000e-006  wp =9.600000e-006
Cin =4.140000e-014  Cout=6.119544e-014
rise time=2.000000e-011  offset = 0.000000e+000   td=2.442775e-011  from node 0
fall time=0.000000e+000  offset = 0.000000e+000   td=1.308529e-010  from node 0
--------------------------------------

--------------------------------------
node nb=2   level=2
inverter
inputs :
output : f
```

```
fanin (node nb,level):
(1,1)
fanout (node nb,level):
(3,4)
(5,3)
wn =9.600000e-006  wp =9.600000e-006
Cin =4.140000e-014  Cout=9.833833e-014
fall time=1.022481e-011  offset = 0.000000e+000   td=8.109656e-011  from node 1
rise time=2.573315e-010  offset = 2.187118e-012   td=1.863634e-010  from node 1
---------------------------------------

---------------------------------------
node nb=5   level=3
inputs :f cin a b
output : cout
fanin (node nb,level):
(2,2)
(0,0)
fanout (node nb,level):
(3,4)
(6,4)
wn =9.600000e-006  wp =9.600000e-006
Cin =4.140000e-014  Cout=1.345766e-013
rise time=4.462554e-011  offset = 0.000000e+000   td=1.079448e-010  from node 2
fall time=9.000166e-011  offset = 2.187118e-012   td=3.107043e-010  from node 2
---------------------------------------

---------------------------------------
node nb=3   level=4
inputs :f cin a cout b
output : sum
fanin (node nb,level):
(5,3)
(2,2)
(0,0)
fanout (node nb,level):
(4,5)
wn =9.600000e-006  wp =9.600000e-006
Cin =4.140000e-014  Cout=8.801497e-014
fall time=1.458141e-010  offset = 3.503769e-011   td=4.856334e-010  from node 5
rise time=4.827825e-010  offset = 6.931307e-011   td=4.675109e-010  from node 5
---------------------------------------

---------------------------------------
node nb=6   level=4
inverter
inputs :
output : cout
fanin (node nb,level):
(5,3)
fanout (node nb,level):
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=5.079726e-013
fall time=1.458141e-010  offset = 3.503769e-011   td=8.762610e-010  from node 5
```

```
rise time=4.827825e-010  offset = 6.931307e-011   td=5.648203e-010  from node 5
----------------------------------------

----------------------------------------
node nb=4   level=5
inverter
inputs :
output : sum
fanin (node nb,level):
(3,4)
fanout (node nb,level):
wn =9.600000e-006  wp =9.600000e-006
Cin =4.140000e-014  Cout=5.155383e-013
rise time=6.307478e-010  offset = 1.702595e-010   td=7.686040e-010  from node 3
fall time=1.568220e-010  offset = 3.890999e-010   td=8.913231e-010  from node 3
----------------------------------------
Circuit 2a

worst output node 4
worst path
(5,1)
(3,3)
(4,4)
----------------------------------------
node nb=1   level=1
inputs :cin a b
output : d
fanin (node nb,level):
(0,0)
fanout (node nb,level):
(2,2)
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=3.298968e-014
rise time=2.000000e-011  offset = 0.000000e+000   td=2.423030e-011  from node 0
fall time=0.000000e+000  offset = 0.000000e+000   td=2.242541e-010  from node 0
----------------------------------------

----------------------------------------
node nb=5   level=1
inputs :cin a b
output : cout
fanin (node nb,level):
(0,0)
fanout (node nb,level):
(3,3)
(6,2)
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=5.950368e-014
rise time=0.000000e+000  offset = 0.000000e+000   td=6.835455e-011  from node 0
fall time=0.000000e+000  offset = 0.000000e+000   td=3.780342e-010  from node 0
----------------------------------------

----------------------------------------
node nb=2   level=2
```

```
inverter
inputs :
output : d
fanin (node nb,level):
(1,1)
fanout (node nb,level):
(3,3)
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=2.867256e-014
fall time=1.058958e-011  offset = 0.000000e+000   td=4.924493e-011  from node 1
rise time=3.871683e-010  offset = 3.066994e-011   td=2.370839e-010  from node 1
--------------------------------------

--------------------------------------
node nb=6   level=2
inverter
inputs :
output : cout
fanin (node nb,level):
(5,1)
fanout (node nb,level):
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=1.079726e-013
fall time=1.259038e-010  offset = 5.402629e-012   td=2.550978e-010  from node 5
rise time=6.172425e-010  offset = 6.941291e-011   td=4.924451e-010  from node 5
--------------------------------------

--------------------------------------
node nb=3   level=3
inputs :d cin a cout b
output : sum
fanin (node nb,level):
(5,1)
(2,2)
(0,0)
fanout (node nb,level):
(4,4)
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=3.664512e-014
fall time=1.259038e-010  offset = 5.402629e-012   td=2.609252e-010  from node 5
rise time=6.172425e-010  offset = 6.941291e-011   td=5.429824e-010  from node 5
--------------------------------------

--------------------------------------
node nb=4   level=4
inverter
inputs :
output : sum
fanin (node nb,level):
(3,3)
fanout (node nb,level):
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=1.079726e-013
rise time=2.862156e-010  offset = 1.178174e-010   td=3.768810e-010  from node 3
```

```
fall time=1.315848e-010  offset = 4.771900e-010   td=7.311283e-010  from node 3
---------------------------------------
Circuit 2b

worst output node 4
worst path
(5,1)
(3,3)
(4,4)
---------------------------------------
node nb=1    level=1
inputs :cin a b
output : d
fanin (node nb,level):
(0,0)
fanout (node nb,level):
(2,2)
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=3.298968e-014
rise time=2.000000e-011  offset = 0.000000e+000   td=2.423030e-011  from node 0
fall time=0.000000e+000  offset = 0.000000e+000   td=2.242541e-010  from node 0
---------------------------------------

---------------------------------------
node nb=5    level=1
inputs :cin a b
output : cout
fanin (node nb,level):
(0,0)
fanout (node nb,level):
(3,3)
(6,2)
wn =4.800000e-006  wp =9.600000e-006
Cin =3.105000e-014  Cout=8.078809e-014
rise time=0.000000e+000  offset = 0.000000e+000   td=8.890244e-011  from node 0
fall time=0.000000e+000  offset = 0.000000e+000   td=3.255378e-010  from node 0
---------------------------------------

---------------------------------------
node nb=2    level=2
inverter
inputs :
output : d
fanin (node nb,level):
(1,1)
fanout (node nb,level):
(3,3)
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=3.902256e-014
fall time=1.058958e-011  offset = 0.000000e+000   td=6.416593e-011  from node 1
rise time=3.871683e-010  offset = 3.066994e-011   td=2.546676e-010  from node 1
---------------------------------------

---------------------------------------
```

```
node nb=6   level=2
inverter
inputs :
output : cout
fanin (node nb,level):
(5,1)
fanout (node nb,level):
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=1.079726e-013
fall time=1.652327e-010  offset = 6.286076e-012   td=2.853559e-010  from node 5
rise time=5.895247e-010  offset = 3.077546e-011   td=4.410630e-010  from node 5
---------------------------------------

---------------------------------------
node nb=3   level=3
inputs :d cin a cout b
output : sum
fanin (node nb,level):
(5,1)
(2,2)
(0,0)
fanout (node nb,level):
(4,4)
wn =9.600000e-006  wp =4.800000e-006
Cin =3.105000e-014  Cout=5.119224e-014
fall time=1.652327e-010  offset = 6.286076e-012   td=3.264846e-010  from node 5
rise time=5.895247e-010  offset = 3.077546e-011   td=4.740988e-010  from node 5
---------------------------------------

---------------------------------------
node nb=4   level=4
inverter
inputs :
output : sum
fanin (node nb,level):
(3,3)
fanout (node nb,level):
wn =4.800000e-006  wp =9.600000e-006
Cin =3.105000e-014  Cout=1.134398e-013
rise time=3.724667e-010  offset = 1.402513e-010   td=4.487601e-010  from node 3
fall time=1.059702e-010  offset = 4.211137e-010   td=5.823613e-010  from node 3
---------------------------------------

worst output node 4
worst path
(5,1)
(3,3)
(4,4)
---------------------------------------
node nb=1   level=1
inputs :cin a b
output : d
fanin (node nb,level):
(0,0)
```

```
fanout (node nb,level):
(2,2)
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=3.298968e-014
rise time=2.000000e-011  offset = 0.000000e+000   td=2.423030e-011  from node 0
fall time=0.000000e+000  offset = 0.000000e+000   td=2.242541e-010  from node 0
----------------------------------------

----------------------------------------
node nb=5   level=1
inputs :cin a b
output : cout
fanin (node nb,level):
(0,0)
fanout (node nb,level):
(3,3)
(6,2)
wn =4.800000e-006  wp =1.440000e-005
Cin =4.140000e-014  Cout=1.020725e-013
rise time=0.000000e+000  offset = 0.000000e+000   td=1.094503e-010  from node 0
fall time=0.000000e+000  offset = 0.000000e+000   td=2.967251e-010  from node 0
----------------------------------------

----------------------------------------
node nb=2   level=2
inverter
inputs :
output : d
fanin (node nb,level):
(1,1)
fanout (node nb,level):
(3,3)
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=4.937256e-014
fall time=1.058958e-011  offset = 0.000000e+000   td=7.908694e-011  from node 1
rise time=3.871683e-010  offset = 3.066994e-011   td=2.706233e-010  from node 1
----------------------------------------

----------------------------------------
node nb=6   level=2
inverter
inputs :
output : cout
fanin (node nb,level):
(5,1)
fanout (node nb,level):
wn =4.800000e-006  wp =4.800000e-006
Cin =2.070000e-014  Cout=1.079726e-013
fall time=2.045616e-010  offset = 7.169524e-012   td=3.156140e-010  from node 5
rise time=5.218612e-010  offset = 3.579448e-011   td=4.144163e-010  from node 5
----------------------------------------

----------------------------------------
node nb=3   level=3
```

```
inputs :d cin a cout b
output : sum
fanin (node nb,level):
(5,1)
(2,2)
(0,0)
fanout (node nb,level):
(4,4)
wn =1.440000e-005  wp =4.800000e-006
Cin =4.140000e-014  Cout=6.573936e-014
fall time=2.045616e-010  offset = 7.169524e-012   td=3.916248e-010  from node 5
rise time=5.218612e-010  offset = 3.579448e-011   td=4.306875e-010  from node 5
---------------------------------------

---------------------------------------
node nb=4    level=4
inverter
inputs :
output : sum
fanin (node nb,level):
(3,3)
fanout (node nb,level):
wn =4.800000e-006  wp =1.440000e-005
Cin =4.140000e-014  Cout=1.189070e-013
rise time=4.587179e-010  offset = 1.622659e-010   td=5.265506e-010  from node 3
fall time=9.695836e-011  offset = 3.822083e-010   td=5.109568e-010  from node 3
---------------------------------------
```