An Accurate and Efficient Gate Level Delay Calculator for MOS Circuits

*Foong-Charn Chang ** (201) 582 - 5248*
Chin-Fu Chen (201) 582 - 5144
Prasad Subramaniam (201) 582 - 2472

AT&T Bell Laboratories
600 Mountain Avenue, Murray Hill, NJ 07974

ABSTRACT

This paper describes an accurate and efficient gate level delay calculator that automatically characterizes and computes the gate delays of MOS circuits. The high accuracy is attributed to a sophisticated delay model, which includes an accurate representation of the waveform, a consistent and meaningful definition of delay, a consideration of waveform slope effects at both the input and output of a gate, and an innovative approach for handling transmission gate circuits. Meanwhile, the high efficient delay characterization is accomplished through a fast timing simulation technique instead of using a circuit simulation or a timing simulation technique, a theorem to reduce a two-dimensional delay table into a scaled one-dimensional table, and an incremental characterization process.

The delay calculator has been used in a production timing analyzer and a production multiple delay simulator since 1986. The results show that the multiple delay simulator performs 5000 times faster than a SPICE-like circuit simulator at only 15% cost of accuracy. Gate delay models, delay characterization, and practical examples are presented in this paper.

1. Introduction

Delay calculation is the most important process for a timing analyzer and a multiple delay simulator. In timing analysis, a set of critical paths that fail to satisfy timing constraints are identified based on the gate delays calculated by the delay calculator. The accuracy of this analysis is directly related to the gate delay accuracy. For multiple delay simulation, simulation results are generated based on logic levels and fixed rise and fall discrete delays. Although the speed of the multiple delay simulator is equivalent to that of a logic simulator [7,8], its accuracy is totally dependent on the gate delay accuracy. Furthermore, the overhead of delay calculation could be significant if the delay calculator is not efficient enough.

Delay calculation consists of two phases, namely, a generic delay characterization and a more specific delay computation. The first phase involves a detailed characterization of each cell in the circuit based on a pre-derived delay model. Subsequently, the second phase calculates the actual gate delays using the first phase results and the actual loading of the circuit. Since most logic simulators do not have a built-in electrical level simulation capability, all standard cells in a library are pre-characterized using a SPICE-like stand alone circuit simulator. While this method is extremely efficient due to pre-characterization, it is difficult to model various environmental factors such as temperature, VDD, and circuit loading. In addition, it is almost impossible to extend this method to model variable size cells used in a full-custom design.

To resolve these limitations, a in-circuit characterization technique that accounts for the actual loading, environments, and cell sizes has been derived. This technique can be easily implemented in a mixed-mode simulation environment[2], since the logic simulator shares the same data structures as a timing or a E-LOGIC like fast timing simulator[6,14]. Therefore, gate delays can be automatically calculated by these detailed simulators which consider the I-V curves, the loading capacitances, as well as the effect of waveform slopes at both the input and output of the gate.

For delay characterization purposes, MOS circuits are divided into three broad categories:

(a) complex driver-load gates, (e.g. Figure 1)

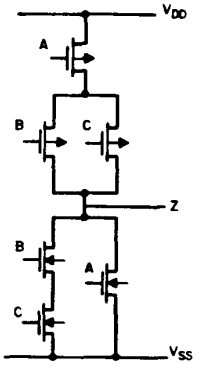

Figure 1. A CMOS driver-load gate.

This is a multiple-input single-output device, which consists of two clusters of transistors: the first cluster, i.e. "load", connects the output to V_{DD} ; the second cluster, i.e. "driver" connects the output to V_{SS} [2].

(b) subcircuits with driver-load gates and unidirectional transmission gates (e.g. Figure 2)

** Primary Author to contact

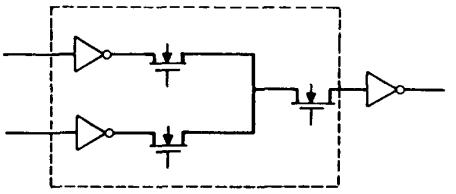

Figure 2. A subcircuit containing driver-load gates and unidirectional transmission gates.

(c) subcircuits with driver-load gates and bidirectional transmission gates (e.g. Figure 3)

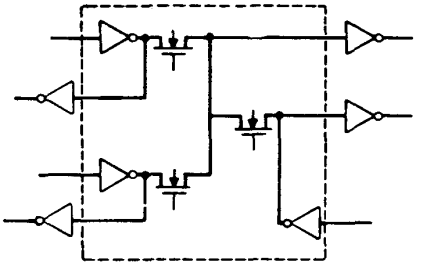

Figure 3. A subcircuit containing driver-load gates and bidirectional transmission gates.

This paper describes an accurate delay model and an efficient characterization technique for categories (a) and (b). The delay model is a slope-capacitance oriented model which can further be reduced into a one-dimensional model to achieve space and time efficiency. For delay characterization, a technique based on the fast timing simulation algorithm will be described. Practical examples are used to illustrate the speed/accuracy tradeoff between the chosen fast timing and the full timing characterization techniques. Since the delay of category (c) is state dependent, it is not possible to simulate such circuits accurately using a multiple delay simulator. Therefore, in a mixed mode simulation environment, such type of circuits are automatically set to timing mode to assure the simulation accuracy.

2. Delay Model

2.1 Definition of Delay

To develop a delay model, a consistent and meaningful definition of delay is necessary. The most popular delay definition, provided by Elmore [10], can be inconsistent in MOS circuits because it occasionally gives rise to negative delays. The following definition is used to alleviate this problem: **Delay** *is measured as the difference between the time the output signal crosses a certain threshold voltage and the time the input signal crosses that same threshold voltage.* In Figure 4, the delay T_D is $T_o - T_i$, where T_i and T_o are the times when the input and output voltage respectively cross the threshold voltage V_T.

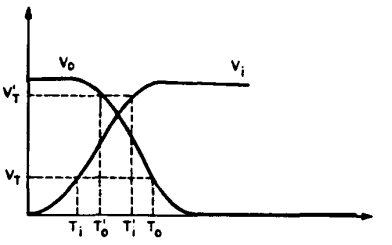

Figure 4. Waveforms illustrating the effect of threshold voltage on the delay of a gate.

2.2 Threshold Voltage V_T

The choice of the threshold voltage V_T is critical for a meaningful delay, especially when the input waveform changes very slowly. For example, in Figure 4, chosing V'_T as the threshold voltage, results in a negative delay because T'_o is earlier than T'_i. This is physically not very meaningful. A proper choice for the threshold voltage is the unity gain point of the dc transfer characteristic of the gate [11]. With this choice, as long as the input waveform is monotonic, the delay is always positive irrespective of how slowly the input changes [12]. The unity gain point is determined for each type of gate from its dc transfer characteristics. For multiple input gates, the unity gain point is determined with respect to each input, thus providing a more accurate delay model.

Some driver-load gates do not have a dc operating point, as in the case of the push-pull inverter shown in Figure 5. For such a device, the threshold voltage is determined as the smallest voltage that must be applied to the input (largest if the input is falling) in order to form a conducting path between the output and V_{SS} or V_{DD}.

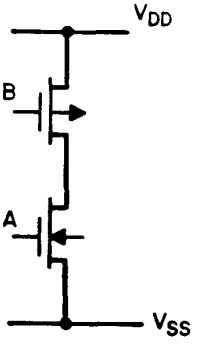

Figure 5. Push-pull inverter circuit. DC transfer curves do not exist for any of the two inputs.

For transmission gates, there are two types of transitions at the output. In the first case, the data input is stable, and the control input is switched. In this case, the threshold voltage is determined as described for the push-pull inverter. In the second case, the gate has already been turned on by the control, and the output transition is due to a transition at the data input. For this case, the threshold voltage is simply chosen as $\frac{V_{DD}}{2}$.

2.3 Representation of Waveform Slope

The delay is measured as a function of the output capacitance and the input waveform slope as the delay parameters. The dependence of delay on the input waveform slope enables us to obtain fairly accurate delays for extremely slowly changing inputs. The input waveform slope can be represented by a single parameter, T_{12}, if we approximate the waveform by a linear segment followed by an exponential tail as shown in Eqn.(1) for a rising signal and Eqn.(2) for a falling signal. T_{12} is the time spent by the signal between 40% and 60% of the steady state value. This approximation has been found to work well for most MOS digital circuits.

$$f = \begin{cases} \dfrac{0.2t}{T_{12}} & t < 3T_{12} \\ 1-0.4e^{-\frac{(t-3T_{12})}{2T_{12}}} & t \geq 3T_{12} \end{cases} \quad (1)$$

$$f = \begin{cases} 1-\dfrac{0.2t}{T_{12}} & t < 3T_{12} \\ 0.4e^{-\frac{(t-3T_{12})}{2T_{12}}} & t \geq 3T_{12} \end{cases} \quad (2)$$

2.4 Equations for gate delays and slopes

The output delay T_D and output slope T_{12_o} of a gate is modeled as a function of two parameters, the capacitance C_o and the input waveform slope T_{12_i}. e.g.

$$T_D = f_D\,(C_o, T_{12i}) \quad (3)$$

$$T_{12o} = f_{12}\,(C_o, T_{12i}) \quad (4)$$

It can be shown that these two-dimensional functions can be simplified into one-dimensional functions if only one input of a gate changes at a time. e.g.

$$T_D = C_o g_D\left(\frac{C_o}{T_{12i}}\right), \quad T_{12i} \neq 0 \quad (5)$$

$$T_{12o} = C_o g_{12}\left(\frac{C_o}{T_{12i}}\right), \quad T_{12i} \neq 0 \quad (6)$$

Equations(5) and (6) indicate that if we scale the input waveform slope T_{12_i} by the output capacitance C_o, then the scaled delay $\frac{T_D}{C_o}$ and the scaled output waveform slope $\frac{T_{12_o}}{C_o}$ are one-dimensional functions of the scaled waveform slope at the input $\frac{T_{12_i}}{C_o}$. This theorem not only saves tremendous storage, it also speeds up the characterization significantly [12].

3. Implementation of a Gate Level Delay Calculator

The delay calculation takes place in two phases. In the first phase, the delay T_D and the output slope T_{12_o} are tabulated for every type of gate for each input of that gate. The gate type distinguishes a gate not only by function, but also by transistor sizes in the gate. The capacitance range of each table is determined by the largest and the smallest capacitance that is driven by that type of gate. In the second phase, a quadratic spline interpolation and linear extrapolation are used to compute the actual delays.

3.1 First Phase - Delay Characterization

Since the delay between any input and output pair is a function of input states and is characterized statically, a sensitizing input pattern needs to be determined. For fully complementary CMOS and depletion load NMOS gates, the sensitizing pattern is generated to produce the largest delay. However, for non-complementary driver-load gates, (because it is quite difficult to find such a pattern), a random pattern is applied at the inputs of the gate. Hence, for such gates, the delay under actual operating conditions may be larger than that characterized.

After choosing a sensitizing pattern, the output slope T_{12_o} and the delay T_D are tabulated by performing a timing or fast timing simulation on the gate under a range of loading capacitances and input slopes based on Eqns (5) and (6). Since this characterization process is repeated for both every input and every gate type, the first phase of delay characterization can be quite time consuming, depending on the number of different gate types in the circuit and the complexity of each gate type. However, it is done only once during the lifetime of a circuit. During the course of a design, only those gate types that are newly introduced and those signal capacitances which have fallen outside the previous range will be recharacterized. This incremental capability saves the user a significant amount of CPU time. Furthermore, using a fast timing simulation technique speeds up the delay characterization by a factor of 15 over the full timing technique at only a minimum cost of accuracy. More details will be described in section 4.

3.2 Second Phase - Delay Computation

In the second phase of delay characterization, the delays for each individual gate are computed. First, the slope of the waveform, i.e., T_{12}, is estimated at each node in the circuit by applying a step function at the input that controls that node. For every multiple-input gate, the largest T_{12} value due to the various inputs is chosen to represent the slope at that output. Then this information is used with the type and capacitance information of the gate to compute the delays.

The second phase of the delay characterization has several valuable distinctions. The first arises due to the different threshold voltages that are used to characterize each type of gate. Consider the example in Figure 6, which consists of one gate driving another.

The first gate has a delay of T_{D_1}, which is measured using a threshold voltage V_{T_1}, and the second gate has a delay T_{D_2}, which is measured using the threshold voltage V_{T_2}. Because of the difference in the threshold voltages, the time T_L which the signal takes to change from V_{T_1} to V_{T_2} must be accounted for. The values of T_L (i.e. the latent time) depends on the threshold voltages, V_{T_1} and V_{T_2}, and the rising or falling of the waveforms .

Secondly, the delay modeling of the gates that drive the transmission gates is purely empirical since the driving gate delay depends on whether the transmission gate is turned on or off. Because it is not possible to know the state of the transmission gate during the delay characterization phase, some approximations have to be made. The delay T_1 is first calculated as a function of the output capacitance C_1 assuming that the transmission gate is off. Then, a second delay T_2 is calculated as a function of

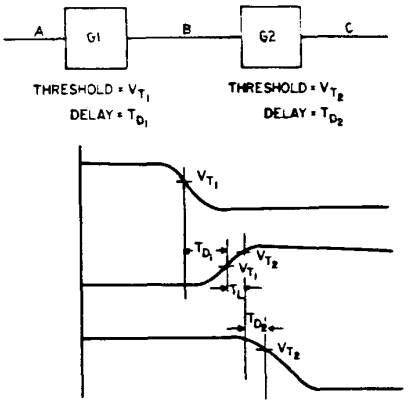

Figure 6. The delays for the gates G_1 and G_2 do not account for the latent time T_L.

C_1+C_2, C_2 being the load driven by the transmission gate as shown in Figure 7. This is an approximation of the delay when the transmission gate is on.

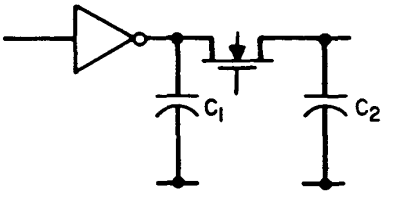

Figure 7. Gate driving a transmission gate. The delay depends on whether the transmission gate is on or off.

The delay of the driving gate is assumed to be T_1. However, the difference between these delays, i.e., T_2–T_1 is added to the transmission delay of the transmission gate to compensate for the approximation made in the delay of the driving gate. In general, this approximation is valid if $C_1 \gg C_2$.

Finally, delays computed for TTL compatible output buffers need special attention to be accurate. As described earlier, the multiple delay simulator computes the delays based on the unity gain point of the dc transfer curves of a driver-load gate. For output buffers, the threshold voltages are determined from the threshold voltage of the transistors that constitute the push-pull buffer. If the buffer is designed to be TTL compatible, this threshold voltage will be much higher than 2.4 V, which represents the logic 1 value in TTL. Thus, the delays computed by the simulator will tend to be quite large for these buffers especially under large load conditions. To simulate these more accurately, a command for delay calculation has been implemented to allow the user to set the threshold of output gates to an arbitrary value. This allows for accurate delay modeling of TTL compatible output buffers.

4. Fast Timing Simulation for Delay Characterization

The full timing simulation employs a implicit integration method and an iterative technique to solve a set of nonlinear differential algebraic equations [5]. It is quite expensive when used to characterize the delays in a VLSI circuit for subsequent multiple-delay simulation. Recently, a fast timing algorithm was developed for the MOTIS3 simulation system [6]. This algorithm is used to rapidly compute the delays with little sacrifice in accuracy (see Figure 8).

START:
  TIME = TIME + ΔT
  generate input voltage
  get branch current I_{net}
  $\Delta H = C\Delta V / I_{net}$
  if(ΔH > *Max Time Step) then*
    get V_{out} *by interpolation*
    ΔT = *Max Time Step*
  else
    $V_{out} = V_{out} + \Delta V$
    $\Delta T = \Delta H$
  end if
  if(V_{out} > *threshold voltages) then*
    calculate T_{12_o} *and* T_D
    STOP
  else
    go to START
  end if

Figure 8. Fast delay calculation algorithm using fast timing simulation.

The delay is determined in terms of discrete time-steps ΔH which are obtained from a forward Euler prediction as

$$\Delta H = \frac{C\Delta V}{I_{net}}$$

where C is the output load, and I_{net} is the net output current. The accuracy of the delay characterization is governed by the incremental voltage step ΔV and the maximum time-step . When the output switches slowly, the ΔH computed using the above formula may be greater than the maximum time-step specified. In this case, the time-step is clamped to the maximum value, and the output voltage V_{out} is updated using interpolation. As soon as the output voltage crosses the appropriate threshold voltages, T_D and T_{12_o} are determined.

To evaluate the accuracy of the fast delay algorithm, a VLSI circuit containing 10314 transistors was simulated using both the regular and the fast delay characterization methods. Figure 9 shows a histogram of the deviations in the delays computed for this circuit. Over half the delays are identical, and a significant part of the rest are within 3%. Furthermore, a majority of the deviations are on the conservative side. This example illustrates that characterization of the delay using the fast timing algorithm has a minor impact on the accuracy of the delays. The speed performance of several production circuits on a VAX 8650 running under the VMS operation system is summarized in Table 1. The comparison of the speed is made between the delay calculation, that uses **full timing** simulation to characterize the delays, and the delay calculation, that uses **fast timing** simulation to characterize the delays. The speed up factors shown in table 1 range from 10 to 20.

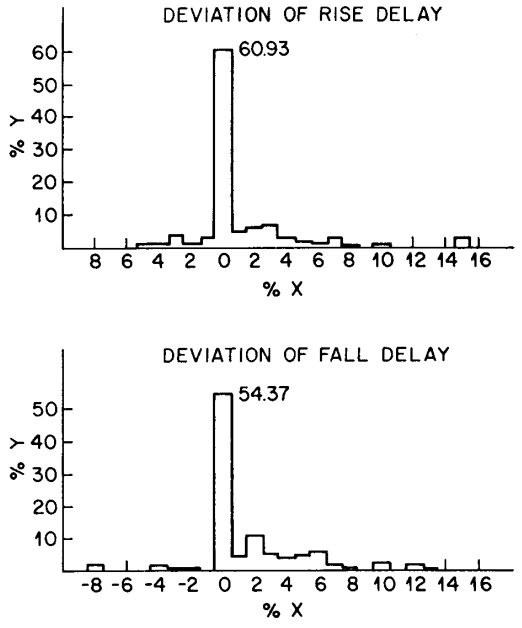

Figure 9. Error distribution between the fast timing and the full timing delay characterization for a 10314 transistor circuit. Positive error means fast delay calculation is more pessimistic than the full-timing.

Circuit	# of Trs	Timing	Fast Timing	Speed up Factor
A	10314	441.8 sec	29.7 sec	14.9
B	31875	3423.9	223.7	15.3
C	18803	3357.8	117.3	28.6
D	23839	1569.5	77.1	20.3
F	24690	525.5	25.9	20.3

Table 1. Speed comparison of two different delay characterization techniques

5. Validation of the Delay Model

This section illustrates the accuracy of the delay calculator used in a multiple delay simulator with a four phase clock generator and AT&T polycells.

5.1 Four Phase Clock Generator

This circuit generates four clocks, *MCK*, $\overline{MCK}$, *SCK* and $\overline{SCK}$ from a single clock input *CLKI*. The delay of each one of these clock edges was measured with respect to the input clock. Table 2 demonstrates the accuracy of the multiple delay simulation as compared to the MOTIS3 timing simulation and the ADVICE circuit simulation [13].

5.2 AT&T CMOS Standard Cell Library Characterization

To further validate the accuracy of the delay calculator, all the cells in the AT&T CMOS standard cell library were simulated using the multiple delay simulator. The results were compared to the results of the timing simulation. The cells were divided into three groups: combinational circuits, flip-flops, and buffers.

	ADVICE	MOTIS3 timing	MOTIS3 multiple delay
Rising edge at *CLKI*			
CLKI→*MCK*	27.8 ns	28.2 ns	34.0 ns
CLKI→$\overline{MCK}$	30.2 ns	31.0 ns	38.6 ns
CLKI→*SCK*	13.9 ns	13.8 ns	16.8 ns
CLKI→$\overline{SCK}$	17.0 ns	16.8 ns	20.8 ns
Falling edge at *CLKI*			
CLKI→*MCK*	21.0 ns	20.4 ns	24.2 ns
CLKI→$\overline{MCK}$	20.7 ns	20.0 ns	22.8 ns
CLKI→*SCK*	34.5 ns	34.6 ns	41.4 ns
CLKI→$\overline{SCK}$	34.3 ns	33.8 ns	40.8 ns

Table 2. Comparison of delays for the four phase clock generator

Multiple input combinational cells were connected to behave like inverters and loaded to drive ten standard inverters. The Q and $\overline{Q}$ of the flip-flops, the input buffers, and the input stages of the bidirectional buffers have similar loads. The output buffers and the output stages of bidirectional buffers were loaded with a capacitance of 100pF. Since the output buffers are designed to be TTL compatible, the threshold voltage was adjusted to 0.4V to measure the fall delay and to 2.4V to measure the rise delay.

The waveforms at the output of each standard cell were monitored for both the timing and the multiple delay simulations. The time of transition of the multiple delay simulation for a signal was compared with the time the same signal crossing 0.4 V_{DD} in the timing simulation. This number was chosen, because it represented the unity gain point of a standard inverter in the library. Figure 10 shows a histogram of the deviation of the multiple delay simulation from the timing cross-over point. >From the histogram, it can be seen that most of the multiple delay transitions occur within ± 2ns of the timing transition. A majority of the deviations are on the conservative side.

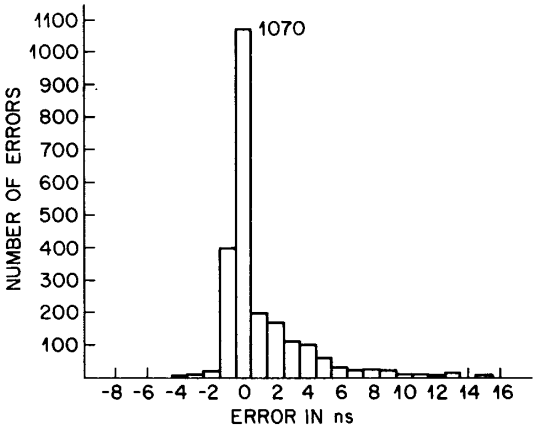

Figure 10. Distribution of error incurred by multiple delay simulation when compared with timing simulation for cells in the CMOS standard cell library. A positive error indicates that the delay computed by the multiple delay simulator is larger than that computed by timing.

6. Conclusion and future research

An efficient and accurate delay calculator has been described in this paper. This in-circuit delay calculator is integrated with a production timing analyzer and a multiple delay simulator. Several MOS chips have been verified and the results are very favorable showing an accuracy of about 15% when compared to a conventional circuit simulation. The speed of the multiple delay simulator is equivalent to a logic simulator, and is about three orders of magnitude faster than a SPICE-like circuit simulator. The overhead of delay characterization is minimum because a fast timing algorithm is used. Furthermore, the incremental delay characterization feature described reduces this overhead as much as possible.

It is useful to extend the model and techniques described here to handle a driver-load gate driving a bidirectional transmission gate. To improve simulation accuracy, the delay can also be calculated dynamically during simulation by carrying the slope information.

Acknowledgements

Several people have contributed to the development of delay calculator. Some early work was done by Ming-Tang Yin. Special thanks are due to Dan Hogerty and Suthat Narakornpichit who provided software tools that helped debug the program. The assistance of several circuit designers at AT&T Bell Laboratories who provided the circuits that were used to validate the accuracy of the multiple delay models is gratefully acknowledged. Finally, the active participation of several MOTIS3 users, particularly, Chong-Hoc Hao and Chris Daley helped uncover many problems with the delay model and provided valuable feedback during its refinement.

REFERENCES

[1] H.N.Nham and A.K.Bose, 'A Multiple Delay Simulator for MOS LSI Circuits', *Proceedings of the 17th Design Automation Conference,Minneapolis,* Minnesota, pp. 610-617, June 1980.

[2] C.F.Chen, C-Y.Lo, H.N.Nham, and Prasad Subramaniam, 'The Second Generation MOTIS Mixed Mode Simulator', *Proceedings of the 21st Design Automation Conference*, Albuquerque, New Mexico, pp. 10-17, June 1984.

[3] B.R.Chawla, H.K.Gummel, and P.Kozak, 'MOTIS - An MOS Timing Simulator', *IEEE Transactions on Circuits and Systems*, **Vol. CAS-22**, pp. 901-910, December 1975.

[4] S.P.Fan, M.Y.Hsueh, A.R.Newton, and D.O.Pederson, 'MOTIS-C: A New Circuit Simulator for MOS LSI Circuits' *Proceedings of the International Symposium on Circuits and Systems*, pp. 700-703, April 1977.

[5] C.F.Chen and Prasad Subramaniam, 'The Second Generation MOTIS Timing Simulator - An Efficient and Accurate Approach for General MOS Circuits', *Proceeding of the International Symposium on Circuits and Systems*, pp. 538-542, Montreal, May 1984.

[6] David Tsao, Chin-Fu Chen, 'A "Fast Timing" Simulation for Digital MOS Circuits', *IEEE Internation Conference on Computer-Aid Design*, pp. 185-187, November 1985.

[7] S.A.Szygenda and E.W.Thompson,'Digital Logic Simulation in a Time-Based, Table-Driven Environment, Part I , "Design Verification"', *Computer*, **Vol. 8**, pp. 24-36, March 1975.

[8] H.Y.Chang, G.W.Smith, and R.B.Walford, 'LAMP: System Description', *Bell System Technical Journal*, **Vol. 53, No. 8**, pp. 1431-1449, October 1974.

[9] Prasad Subramaniam, 'Modeling MOS VLSI Circuits for Transient Analysis', *IEEE Journal of Solid State Circuits*, **Vol. SC-21, No. 2**, pp. 276-285, April 1986.

[10] W.C.Elmore, 'The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers', *Journal of Applied Physics*, **Vol. 19, No. 1**, pp. 55-63, January 1948.

[11] D.J.Pilling, P.F.Ordnung, and D.Heald, 'Time Delays in LSI Circuits', *Proceedings of the IEEE International Symposium on Circuit Theory*, pp. 311-315, April 1972.

[12] Prasad Subramaniam, 'Delay Characterization of MOS Transistor Circuits', to appear.

[13] L.W. Nagel, 'ADVICE for Circuit Simulation', Proceedings of ISCAS 1980, Houston.

[14] Seung H. Hwang, Young H. Kim, and A.R. Newton, 'An Accurate Delay Modeling Technique for Switch-Level Timing Verification ', *Proceedings of the 23rd Design Automation Conference*, Las Vegas, Nevada , pp. 227-233, June 1986.