New Efficient Algorithms for Computing Effective Capacitance*

Andrew B. Kahng and Sudhakar Muddu

Silicon Graphics, Inc., Mountain View, CA 94039

UCLA Computer Science Dept., Los Angeles, CA 90095-1596

Abstract

We describe a novel iterationless approach for computing the effective capacitance of an interconnect load at a driving gate output. Our new approach is considerably faster than previous methods for computing effective capacitance, with little or no loss of accuracy. Thus, the approach is suitable within the analysis loop for performance-driven iterative layout optimization. After reviewing previous gate load models and effective capacitance approximations, we separately derive our method for the cases of step and ramp waveform at the gate output, and note ongoing extensions for the case of complex gates (e.g., channel-connected components). Experimental results using the new effective capacitance approach show that our resulting delay estimates are quite accurate – within 15% of HSPICE-computed delays on data corresponding to an 0.25μm microprocessor design.

1 Introduction

With interconnect delays dominating overall path delays for deep-submicron integrated circuits, heuristics for logic synthesis and layout optimization must accurately model interconnect effects. In synthesis and floorplanning, pre-layout delay estimation capability is needed. In (post-layout) timing analysis, existing accurate delay estimates are not efficient enough to be used in the typical incremental synthesis/layout/in-place optimization loop or during performance-driven area routing. In either context, accurate estimations of gate delay and rise time, which are required for a number of signal integrity and reliability checks, depend closely on an accurate model for the driving point admittance of a load interconnect tree at the output of a gate.

In this paper, we propose a new iterationless approach for computing the effective capacitance of an interconnect load at a driving gate output. This new method is considerably faster than previous methods for computing effective capacitance, with little or no loss of accuracy. Thus, it is applicable within the analysis loop for performance-driven iterative layout optimization. We begin our discussion in Section 2 with a review of gate load models; our previously proposed open-ended Π model [2] is of particular interest since it affords a path to linear-time estimation. Section 3 reviews previous effective capacitance approaches, including the approach of McCormick and the approach of Pillage et al. Section 4 then develops our new effective capacitance method for the cases of step and ramp waveform at the gate output (our current work addresses extensions to the case of complex gates, e.g., channel-connected components). Experimental results in Section 5 using the commercial HSPICE simulator on data corresponding to a recent 0.25μm high-end microprocessor project confirm the accuracy of our new methods.

*Author to whom correspondence should be addressed: Dr. Sudhakar Muddu, Silicon Graphics, Inc., muddu@mti.sgi.com.

2 Review of Gate Load Models

With narrower deep-submicron interconnect geometries, the resistive component of the gate load is comparable to or larger than the gate output resistance: the gate does not "see" all of the capacitance loading since the metal resistance "shields" some capacitance. The resistance shielding effect is very significant for deep-submicron technologies. For example, if we increase the interconnect resistance of the load and keep the gate output resistance constant, the total gate delay at the output will *decrease* since the interconnect resistance will tend to shield some of the load capacitance. In this case, while the total gate delay decreases, the increase in interconnect resistance would increase the interconnect propagation delay.

Various *load models* have been proposed for modeling the driving point admittance at the gate output. Gate delays are estimated using these models either through the delay table methodology or through an explicit *simulation* of the gate with the given load model. This section summarizes the range of existing gate load models.

2.1 Driver Modeling

We first review current methodology for modeling the driver itself. We express the *total gate delay* (D_{AB} in Figure 1) as the sum of *intrinsic gate delay* and *gate load delay*:

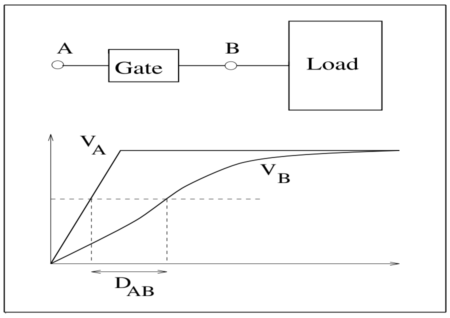

Figure 1: Total gate delay viewed as the sum of intrinsic gate delay and gate load delay.

- *Intrinsic gate delay* is delay due to physical devices (e.g., transistors) in the gate. Intrinsic gate delay can be thought of as total gate delay with infinite load at the output.
- *Gate load delay* is the delay due to the load connected to the output of the gate.

Three basic approaches have been proposed in the literature to model a driver in computing the driver delay. (1) The first approach models the driver as a Thevenin equivalent circuit with an effective linear resistor driven by a voltage source. (2) The second approach characterizes the behavior of a driver using relevant parameters such as input slew time(s) and output load capacitance. (3) The third approach uses piecewise-linear device models to represent a nonlinear device.

In the Thevenin model the effective driver resistance value depends on the input slew time, loading capacitance, and driver size. One can always use a resistor of fixed value (R_{eff}) to model a driver by selecting an appropriate load capacitance to match the 50% threshold delay. A more accurate model, called the slope model, uses a one-dimensional table to compute the effective driver resistance based on the ratio of input slew time and output slew time [7].

In current design practice, pre-characterized total gate delay or gate load delay for various load values is stored for each gate/cell in the library in *delay table* format. The intrinsic gate delay is similarly known for each gate/cell in the library.[1] There are important methodology questions associated with the delay table approach. For example, delays and slew times may be obtained (e.g., during characterization using a circuit simulator) by loading a given gate with a *discrete load capacitor* and then varying the load capacitance and input slew time. But in actual layouts, the gate output is connected via interconnects to other gate inputs. Modeling the load at the gate as a single load capacitor may work well for technologies and designs where the area of interconnect at the gate output is small or the interconnect parameters are not dominant compared to gate parameters. However, with submicron technologies the interconnect resistance, capacitance, and inductance must be considered in the delay table characterization (inductance effects will definitely be an issue in the next process generation).

2.2 The O'Brien/Savarino Π Model

The simplest approximation of the driving point admittance of a load interconnect tree is the total capacitance of the tree (C_{tot}). This is a pessimistic first-order approximation [12].[2] For deep-submicron technologies or MCM interconnects, the total interconnect resistance is large and comparable to the driver output resistance; actual delay is much smaller than that derived from the lumped capacitance model because the interconnect resistance shields the load capacitance seen by the gate driver. Another simple method – approximating the load tree by a single lumped *RC* segment model with resistance and capacitance equal to the total interconnect resistance (R_{tot}) and capacitance (C_{tot}) – is optimistic because the total interconnect resistance is lumped together and shields the total capacitance.[3]

O'Brien and Savarino [8, 9] proposed using a one-segment Π model to approximate the load at the gate output while still considering resistance shielding effects. Their model approximates the load interconnect at the gate by matching the first three moments of the driving point admittance of the interconnect load. The disadvantage of the Π model is that delay tables need to be expanded to four dimensions: slew time of the input voltage, along with the three Π model parameters R_1, C_1, C_2.

Let the driving point admittance at the gate output (X) be represented by $Y_L(s) = \sum_{i=1}^{\infty} A_i s^i = sA_1 + s^2A_2 + s^3A_3 + \ldots$. The parameters of the equivalent circuit are obtained by matching the first three moments of the admittance with corresponding coefficients of the driving point admittance of the Π load model in Figure 2, i.e.,

$$R_1 = \frac{-A_3^2}{A_2^3} \quad C_1 = A_1 - \frac{A_2^2}{A_3} \quad C_2 = \frac{A_2^2}{A_3}\,. \tag{1}$$

[1]Delay tables specify the total gate delay and the output slew rate (rise and/or fall time) for each gate in the library. There exists at least one pair of tables (one table for delay, one table for slew rate) for each gate in the library. Typically, delay tables are developed/characterized as functions of only the input slew time (rise or fall time) and a single capacitance value which represents the effect of the load. This delay table format is equivalent to the so-called empirical "k-factor" formulas for delay and output rise time. Standard industry delay calculators use 2-dimensional tables for delay and output slew rate of gates. Synopsys [14] logic optimization is perhaps the most prominent exemplar of a tool methodology that uses such a format (for characterizing delays during logic synthesis).

[2]C_{tot} includes the load capacitance at the leaves. Coupling effects may be taken into consideration by including their effect in the total capacitance.

[3]The lumped capacitance and lumped *RC* models are referred to as, e.g., *Wire Load Model1* and *Wire Load Model2* in Synopsys manuals [14]. Similar lumped models are available with other industry timing analysis tools.

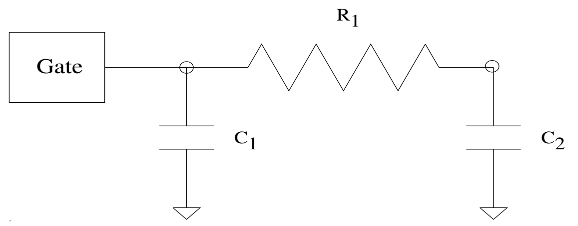

Figure 2: One-segment Π model for matching the first three moments of the driving point admittance of a load interconnect tree.

2.3 Open-ended *RC* Π model

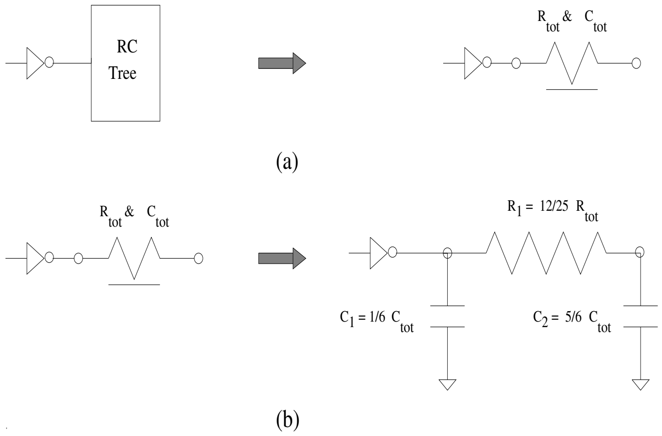

Figure 3: An open-ended *RC* line to capture an *RC* interconnect tree, and the *RC* Π model.

In [2], we proposed a new one-segment *RC* Π model, with *predetermined* parameter values that depend only on the total resistance and total capacitance, to model the driving point admittance of a distributed *RC* interconnect tree. This "open-ended *RC* " model approximates the entire interconnect tree by an equivalent open-ended *RC* line whose resistance and capacitance are equal to the total interconnect resistance and capacitance, as shown in Figure 3(a). By using an open-ended *RC* line to approximate the entire tree, the distributed nature of the load interconnect is still considered in the calculation of model parameters (i.e., the resistance of the open-ended line shields part of the load capacitance from the gate driver). At the same time, there is a substantial gain in efficiency because the moments of the driving point admittance are obtained without recursive tree traversal (indeed, the required linear time complexity of computing the model is no greater than that for computing the simple lumped-capacitance model).

As shown in Figure 3(b), the parameters for the open-ended Π model are

$$R_1 = \frac{12R_{tot}}{25}, \quad C_1 = \frac{C_{tot}}{6}, \quad \text{and} \quad C_2 = \frac{5C_{tot}}{6} \tag{2}$$

The open-ended Π model can be extended to *RLC* networks as shown in Figure 4.

3 Effective Capacitance Approaches

If the designer must account for all possible combinations of C_1, C_2, R_1 in the Π model, a very large lookup table or highly complex k-factor equations (along with an expensive characterization process) would be required. Thus, the effective capacitance model was proposed [6, 13] to allow the delay table approach to be used with a single load capacitance (approximating the load at gate output) in the lookup. Two approaches in the literature afford such an *effective capacitance*: (i) the method of

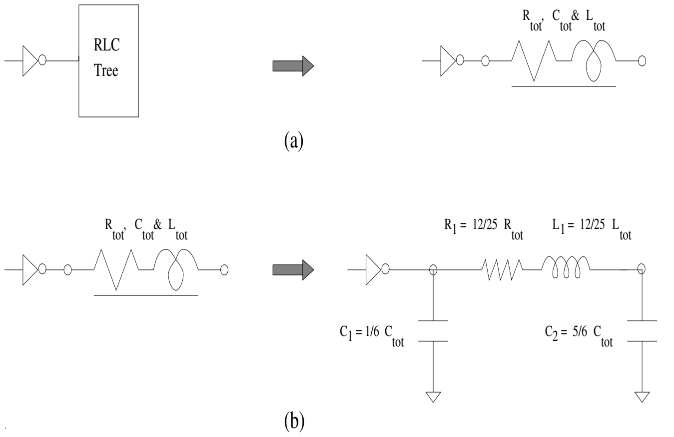

Figure 4: An open-ended *RLC* line model to capture an *RLC* interconnect tree, and the *RLC* Π model.

McCormick [6], and (ii) the method of Pillage et al. [11, 13]. Both of these approaches are iterative in nature and are computationally expensive in the context of delay calculation, static timing analysis, etc. for iterative layout or performance optimization.

McCormick's Effective Capacitance Model. In McCormick's approach, the effective capacitance is obtained as a function of total capacitance (C_{tot}), a step input capacitance (C_{step}) which depends on the gate output slew rate, and the Elmore delay of the load. The step input capacitance C_{step} is chosen to approximate the load admittance Y_L such that the output voltage waveform of the cell (for a step input) passes through the endpoints of some critical output voltage range - e.g., the 0% and 75% points in [6] - at identical times under both C_{step} and Y_L loading. The effective load capacitance is computed as follows:

- Model each cell with an equivalent circuit consisting of a step input source ($v_S(t)$) and a linear source resistance (R_S). Compute the response $v_{out}(t)$ at the cell output by replacing the cell with this simple equivalent circuit and modeling the load at the output by its driving point admittance (Y_L). By approximating the driving point admittance Y_L to a few coefficients of s, the time-domain response $v_{out}(t)$ at the gate output can be obtained.
- A *step input capacitance* C_{step} is chosen to approximate the load admittance Y_L for a step input such that the output voltage $v_{out}(t)$ has matched values at the $t = 0$ and $t = t_{th}$ threshold points, for load Y_L and for capacitive load C_{step}.
- The effective capacitance is computed in the range between step input capacitance C_{step} and total load capacitance C_{tot}. This is because when the inverter response (or slew rate) under no load is fast, the gate will not see all the load capacitance; when the inverter response is slow, the gate will end up charging all the load capacitance. The range between C_{step} and C_{tot} is enforced by computing the Elmore delay of the gate load and the slew rate of the cell under a no-load condition.

Pillage et al.'s Effective Capacitance Model. Pillage et al. [11, 13] calculate an effective capacitance by equating (i) the current at the gate output with driving-point admittance as the load, and (ii) the current at the gate output with a single effective capacitor as the load. It is difficult to obtain a single effective capacitance that will exactly match the actual load in terms of current at the gate output, at all times t. Hence, *average currents* for both models are equated over some period of time, say, until the gate output voltage reaches the 50% threshold, i.e.,

$$\frac{1}{T_D}\int_0^{T_D} I_Y(t)dt = \frac{1}{T_D}\int_0^{T_D} I_C(t)dt$$

where T_D is the time over which the currents are averaged, $I_Y(t)$ is the current at the gate output with driving-point admittance as the load, and $I_C(t)$ is the current at the gate output with the effective capacitance as the load. From transform domain analysis, we have

$$I_Y(s) = Y_L V_{out}(s) \quad \text{and} \quad I_C(s) = sC_{eff}V_{out}(s)$$

Since the current at the gate output is a function of gate output voltage, usually a particular waveform must be assumed for gate output voltage.

Pillage et al. approximate the load at the gate output using O'Brien and Savarino's [8, 9] one-segment Π model. It is for purposes of making the Π model compatible with k-factor delay formulas (i.e., delay tables) that the load must be approximated by some effective capacitance. The effective capacitance is a function of Π model parameters and the gate output threshold delays used in the average current computation above. The gate output threshold delays are in turn expressed in terms of gate delay time and gate output rise time, leading to an iterative computation of the effective capacitance [13, 11].

The disadvantage of this approach, particularly for incremental layout synthesis applications, is its high time complexity due to the iterative nature of the effective capacitance algorithm. Usually, it requires anywhere from 5 to 10 iterations before an accurate capacitance value is obtained. In addition, the computation of the Π model at the gate output requires significant resources: although the moment computation is linear, calculating the first three moments for each gate load requires three traversals of the interconnect tree and can be expensive for large instances.[4]

4 New Methods for Computing Effective Capacitance

The previous effective capacitance approaches require costly moment computations for entire load, and also require empirical equations for the delay and output slew times [13]. Obtaning empirical equations for a modern process technology is itself extremely difficult. Furthermore, the empirical equations in [13, 11] assume fast input transitions.

To address these limitations of existing approaches, we propose two new and simple, yet accurate, techniques. We model the load at the gate output with a simple open-ended *RC* Π model, which eliminates any need for moment computations at the gate output. We then model the gate with a Thevenin equivalent circuit to compute the closed-form equation for the voltage response at the gate output. The value for effective capacitance can be obtained by matching the delay/slew time for the Π model response with the single capacitance model. We now give details of the effective capacitance computation under both step and ramp input configurations of the driver model.

4.1 Effective Capacitance Under Step Input

We model the gate with an equivalent circuit consisting of a step input source ($v_S(t)$) and a linear source resistance (R_S), as shown in Figure (5). We propose to use a Π model for the *RC* (*RLC*) network at the output of a gate to estimate the driving point admittance. In particular, we use the open-ended *RC* Π model [2] cited above for the load at the gate output, because the Π model parameters are now functions of total interconnect capacitance C_{tot} and resistance R_{tot}. Under these conditions, we compute the gate output response analytically using the source resistance R_S with step input source and with the load modeled as a Π model.

[4] In practice, this method accurately predicts the output response only up to 50% threshold voltage; the response waveform tail beyond the 50% threshold is very inaccurately estimated. A possible explanation for the inaccuracy in predicting the slowly decaying tail of the response is that the CMOS gate behaves like a resistor and a single exponential function is used for response under C_{eff} model. To correct this inaccuracy, [11] proposed a two-piece gate output waveform approximation within their effective capacitance method. The first part of the gate output response is estimated using the method of [13], in which the CMOS gate is modeled using a combination of quadratic and linear functions. A resistive model for the CMOS gate is then used to capture the remaining portion of the response.

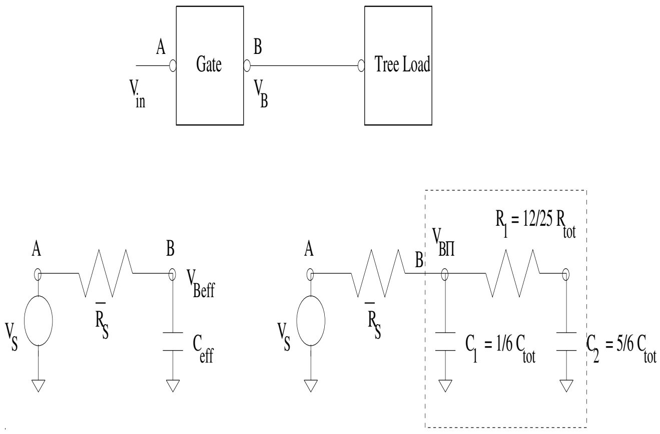

Figure 5: A Thevenin equivalent model of step input source, and a driver resistance for the gate. The load at the gate output is modeled in two different ways: (i) using an effective capacitance model, or (ii) using a higher-order model such as a Π model.

The response at the gate output B in transform domain is given by

$$\begin{aligned} V_B(s) &= V_S(s)\frac{(1+sR_1C_2)}{1+s(R_SC_1+R_SC_2+R_1C_2)+s^2R_SR_1C_1C_2} \\ &= \frac{V_0}{s}\left[1+\frac{(1+S_1R_1C_2)}{s_1b_2(s_1-s_2)}\frac{1}{(s-s_1)}+\frac{(1+S_2R_1C_2)}{s_2b_2(s_2-s_1)}\frac{1}{(s-s_2)}\right] \end{aligned} \tag{3}$$

where

$$\begin{aligned} b_1 &= R_S(C_1+C_2)+R_1C_2 \\ b_2 &= R_SR_1C_1C_2 \\ s_{1,2} &= \frac{-b_1\pm\sqrt{b_1^2-4b_2}}{2b_2} \end{aligned}$$

Depending on the sign of $b_1^2-4b_2$, the poles of the transfer function can be either real or complex; the time-domain response is computed separately for each case. Substituting for the coefficients b_1 and b_2 into the condition, we obtain

$$b_1^2-4b_2 = R_S^2(C_1+C_2)^2+R_1^2C_2^2+2R_SR_1C_2(C_2-C_1) \tag{4}$$

For most practical cases the value of C_2 is greater than C_1, which is true for the open-ended RC Π model. Hence, we study the case of real poles only, with the voltage at gate output B being

$$v_B(t) = V_0\left[1+\frac{(1+S_1R_1C_2)}{s_1b_2(s_1-s_2)}e^{s_1t}+\frac{(1+S_2R_1C_2)}{s_2b_2(s_2-s_1)}e^{s_2t}\right] \tag{5}$$

We use the above response to compute the slew time (or delay with respect to input signal) at any user specified (e.g. 50%) threshold voltage. Assume T_{out}^{Π} is the slew time at the gate output for 50% threshold voltage. If we model the load at the gate output with a single capacitance (C_{step}) then the slew time for such model is given by $k_1R_SC_{step}$, where $k_1 = \ln(\frac{1}{1-v_{thd}})$ is a constant associated with the given threshold voltage v_{thd}. Comparing the above slew times, the single step input capacitance (we refer to this capacitance as C_{step} because of the step input source model) can be computed as

$$C_{step} = \frac{T_{out}^{\Pi}}{k_1R_S} \tag{6}$$

Finally, the effective capacitance is computed in the range between step input capacitance C_{step} and total load capacitance C_{tot}. This is because when the inverter response (or slew rate) under no load is fast the gate will not see all the load capacitance, but when the inverter response is slow the gate will end up charging all the load capacitance. The range between C_{step} and C_{tot} is enforced by computing (i) the load delay D_{LD} of the gate with C_{tot} as load, and (ii) the slew rate D_{NL} of the gate under a no-load condition, then setting

$$C_{eff} = C_{step}+(C_{tot}-C_{step})\frac{1}{1+D_{LD}/D_{NL}} \tag{7}$$

This implies that $C_{eff}\approx C_{step}$ if $D_{LD}/D_{NL}\gg 1$ and $C_{eff}\approx C_{tot}$ if $D_{LD}/D_{NL}\ll 1$.

Algorithm Template: Given the following information for a cell:

- the Π model parameters $(R_1, C_1, C2)$
- the characterized output delay table for the cell

Perform these steps for effective capacitance computation:

1. Compute Π model parameters using either Equation (1) or Equation (2)
2. Compute T_{out}^{Π} by solving for the voltage response at the cell output according to Equation (5)
3. Compute C_{step} from T_{out}^{Π} and R_S using Equation (6)
4. Use the characterized cell delay table and obtain the delay D_{LD} with C_{tot} as load and the delay D_{NL} with no load
5. Compute C_{eff} using Equation (7)

4.2 Effective Capacitance Under Ramp Input

In this case, the driver is again modeled with a Thevenin equivalent circuit, i.e., a source ramp input with rise time T_R and a series source resistance R_S. We again use the open-ended RC Π model to efficiently approximate the entire load at the gate output. The voltage at the gate output (X) in the transform domain is

$$\begin{aligned} V_X(s) &= V_{in}(s)\frac{(1+sR_1C_2)}{1+s(R_SC_1+R_SC_2+R_1C_2)+s^2R_SR_1C_1C_2} \\ &= \frac{V_0(1-e^{-sT_R})}{T_R}\left[\frac{1}{s^2}-\frac{R_S(C_1+C_2)}{s}\right. \\ &\quad \left.+\frac{(1+s_1R_1C_2)}{b_2s_1^2(s_1-s_2)}\frac{1}{(s-s_1)}-\frac{(1+s_2R_1C_2)}{b_2s_2^2(s_1-s_2)}\frac{1}{(s-s_2)}\right] \end{aligned}$$

where $b_1 = R_S(C_1+C_2)+R_1C_2$, $b_2 = R_SR_1C_1C_2$, and $s_{1,2} = \frac{-b_1\pm\sqrt{b_1^2-4b_2}}{2b_2}$. The time-domain response for $t\le T_R$ is

$$\begin{aligned} v_X(t) &= \frac{V_0}{T_R}\left[t-R_S(C_1+C_2)+\frac{(1+s_1R_1C_2)}{b_2s_1^2(s_1-s_2)}e^{s_1t}-\frac{(1+s_2R_1C_2)}{b_2s_2^2(s_1-s_2)}e^{s_2t}\right] \\ &\quad \text{for } t\le T_R \\ &= \frac{V_0}{T_R}\left[T_R+\frac{(1+s_1R_1C_2)(1-e^{-T_Rs_1})}{b_2s_1^2(s_1-s_2)}e^{s_1t}\right. \\ &\quad \left.-\frac{(1+s_2R_1C_2)(1-e^{-T_Rs_2})}{b_2s_2^2(s_1-s_2)}e^{s_2t}\right] \text{for } t>T_R \end{aligned} \tag{8}$$

Depending on the sign of $(b_1^2-4b_2)$ the poles will be either real or complex. Since the quantity $(b_1^2-4b_2)$ from Equation (4) is always positive, we discuss the case of real poles only.

From the response $v_B(t)$, we may compute the slew time T_{out}^{Π} at the user-specified (e.g. 50%) threshold voltage. Now the effective step input capacitance (we again refer to this effective capacitance as C_{ramp} because of the ramp input source model) can be computed as

$$C_{ramp} = \frac{T_{out}^{\Pi}}{k_2 R_S}$$

where the constant k_2 is given by

$$\begin{aligned} k_2 &= \left| \ln\left(\frac{1}{1 + \frac{v_{th} T_R}{R_S C_{tot}} - \frac{T_{RD1}}{R_S C_{tot}}} \right) \right| \qquad \text{for } t \le T_R \\ &= \left| \ln\left(\frac{R_S C_{tot}}{T_R} \cdot \frac{(e^{\frac{1}{R_S C_{tot}/T_R}} - 1)}{(1 - v_{th})} \right) \right| \text{for } t > T_R \qquad (9) \end{aligned}$$

Finally, the effective capacitance is computed in the range between step input capacitance C_{step} and total load capacitance C_{tot} just as in the previous case, i.e., by computing D_{LD} of the gate with C_{tot} as load and D_{NL} of the gate with no load, then setting

$$C_{eff} = C_{ramp} + (C_{tot} - C_{ramp}) \frac{1}{1 + D_{LD}/D_{NL}} \qquad (10)$$

Again, $C_{eff} \approx C_{ramp}$ if $D_{LD}/D_{NL} \gg 1$ and $C_{eff} \approx C_{tot}$ if $D_{LD}/D_{NL} \ll 1$. We use the above-described algorithm template while replacing the step input equations with corresponding ramp input equations.

We believe that our effective capacitance approach works well for delay estimates at threshold voltages between 30% to 60%. The tail end of the response for effective capacitance model could deviate significantly from the actual response [2]. Furthermore, the above proposed methods provide accurate estimation of delay estimates, but do not provide accurate output waveforms that may be needed to derive the output waveforms of downstream gates.

Note that computing the driver resistance R_S is typically quite difficult for complex gate structures, e.g., channel-connected components. For such instances, we are currently testing an alternate approach that is based on our effective capacitance algorithm above: the key difference is that we apply an iteration over R_S values until we arrive at a correct value.

5 Experimental Results

In this section we present a preliminary set of experimental results. Our experiments use the latest 0.25μm CMOS process parameters (corresponding to a recent high-end microprocessor design project) to model our gates and interconnects. We use standard CMOS inverters of various sizes to determine the accuracy of our effective capacitance computation. The experimental configuration consists of two inverters connected in series. We apply a ramp input with slew time 400ps to the first inverter. The load at the second inverter is an *RC* tree of varying topology. The size of the first inverter is fixed at $W_p/W_n = 100/50\mu m$. HSPICE delays are computed by simulating the chain of inverters with different second inverter sizes and different load *RC* trees. We use a static timing tool, within which we have implemented our effective capacitance approach, to compute the delays. Table 1 shows a comparison of delay estimates. Using the new effective capacitance approach, our delay estimates are consistently within 15% of the HSPICE-computed delays.

6 Conclusions

We have proposed a new iterationless approach for effective capacitance computation in both the step input and ramp input regimes. Our new approach is considerably faster than previous methods for computing effective capacitance, with HSPICE simulations confirming that little or no loss of accuracy is incurred. Thus, we believe our technique will enable accurate delay analysis within a tight synthesis-analysis loop, e.g., for performance-driven incremental layout optimization. Our recent work has embedded this calculation in a production performance verification flow for high-end microprocessor design; ongoing work extends the approach to complex gate structures as well as applications within the realm of iterative layout and circuit performance optimization.

INV Size (W_p/W_n)μm	HSPICE Delay (ps)	Estimated Delay (ps)	Total Parasitics R/C
24/12	130	110	260 Ω / 0.50pF
100/50	80	90	260 Ω / 0.50pF
100/50	115	125	710 Ω / 1.40pF
80/40	130	120	150 Ω / 0.40pF
80/40	140	160	300 Ω / 0.80pF
200/100	55	65	1000 Ω / 1.40pF

Table 1: Comparison of HSPICE delays and estimated delays output by our in-house timing tool. The timing tool uses the C_{eff} approach described above to compute load at the inverter output.

REFERENCES

[1] W. C. Elmore, "The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers", *Journal of Applied Physics* 19, Jan. 1948, pp. 55-63.

[2] A. B. Kahng and S. Muddu, "Efficient Gate Delay Modeling for Large Interconnect Loads", *Proc. IEEE Multi-Chip Module Conf.*, Feb. 1996 (Submitted to IEEE Trans. on CAD).

[3] A. B. Kahng and S. Muddu,' "Accurate Analytical Delay Models for VLSI Interconnects", *IEEE Intl. Symposium on Circuits and Systems*, Atlanta, May 1996.

[4] A. B. Kahng, K. Masuko and S. Muddu, "Analytical Delay Models for VLSI Interconnects Under Ramp Input", *IEEE ICCAD*, Nov. 1996.

[5] A. B. Kahng and S. Muddu, "A Glossary on Analysis and Modeling of VLSI Interconnections", *Manuscript*, February 1996.

[6] S. P. McCormick, "Modeling and Simulation of VLSI Interconnections with Moments", *PhD Thesis*, MIT, June 1989.

[7] J. K. Ousterhout, "A Switch-level Timing Verifier for Digital MOS VLSI", *IEEE Trans. on CAD*, July 1985, pp. 336-349.

[8] P. R. O'Brien and T. L. Savarino, "Modeling the Driving-Point Characteristic of Resistive Interconnect for Accurate Delay Estimation", *Proc. IEEE ICCAD*, 1989, pp. 512-515.

[9] P. R. O'Brien and T. L. Savarino, "Efficient On-Chip Delay Estimation for Leaky Models of Multiple-Source Nets", *Proc. IEEE Custom Integrated Circuits Conf.*, 1990, pp. 9.6.1-9.6.4.

[10] L. T. Pillage and R. A. Rohrer, "Asymptotic Waveform Evaluation for Timing Analysis", *IEEE Trans. on CAD* 9, Apr. 1990, pp.352-366.

[11] J. Qian, S. Pullela and L. Pillage, "Modeling the "Effective Capacitance" for the *RC* Interconnect of CMOS gates", *IEEE Trans. on CAD*, December 1994, pp. 1526-1535.

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

[13] C. Ratzlaff, S. Pullela and L. Pillage, "Modeling the *RC* Interconnect effects in a Hierarchical Timing Analyzer", *Proc. IEEE Custom Integrated Circuits Conference*, May 1992, pp. 15.6.1-15.6.4.

[14] Synopsys, "Design Compiler Family Reference Manual", version 3.3a, March 1995.