Gate Delay Calculation Considering the Crosstalk Capacitances

Soroush Abbaspour and Massoud Pedram

University of Southern California

{sabbaspo, pedram}@usc.edu

Abstract

In this paper, we present a new technique for calculating the output waveform of CMOS drivers for cross-coupled RC loads. The proposed technique is based on an effective capacitance calculation for each driver and an efficient, provably convergent, iteration scheme between the coupled drivers. Our technique can easily handle different input arrival times, transition times, and polarities, and can be extended to multiple cross-coupled drivers in a straightforward manner. Experimental results show that the new technique exhibits high accuracy (less than 4% error in average).

1. Introduction

As we continue to exploit deep submicron (DSM) technologies to design faster and smaller circuits, we must revisit the problem of calculating the gate propagation delay. This is an important design problem that is becoming more involved because of the highly nonlinear behavior of CMOS logic gates in the DSM region. Since interconnect modeling and *RC* model order reductions have advanced significantly over the past several years, it is reasonable to assume that we can accurately extract and model the various *R* and *C* parasitics as well as the capacitive coupling between interconnect lines. However, since we are stuck in a pattern of computing gate delays only when grounded linear capacitors load the gate, we are immediately faced with the problem of load modeling for the purpose of gate delay calculation [15].

Modeling the coupling capacitors is the most difficult step in calculating the load. Sometimes this problem is addressed by modeling the coupling capacitors as elements to ground, with modified values of capacitances. For example, for opposite direction switching of two identical coupled lines, switching at the same instance of time, the coupling capacitance can be accurately modeled as twice the amount of line capacitance to ground. Although such approximation tends to yield pessimistic delay values, in general, it does not provide an upper bound on the delay for many realistic coupling scenarios [15,21].

An example of VLSI routing is depicted in Figure 1. If there are significant coupling capacitances among these lines, then transitions on some subset of lines (aggressors) can affect the output behavior of the remaining lines (victims). Furthermore, because of different signal arrival times, slew rates, and switching behavior (falling or rising) of transitions on the aggressor lines, the output waveforms of the victim lines may vary by a lot [15,21].

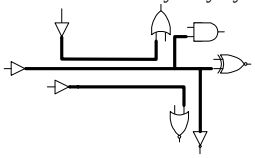

Figure 1: An example VLSI routing scenario

To analyze noise, due to capacitance coupling, one can start with a reduced order coupled interconnect model and calculate the signals on a quiet victim line by superimposing the coupled signals from all other lines. Such a model, while inexact, can provide a reasonable approximation since the nonlinear CMOS gate of the non-switching victim line is behaving like a transistor in its linear region of operation, therefore, is modeled fairly well by a linear resistor. However, when the victim line itself is also switching, the problem becomes much more complex. As the victim line switches, the impedance of its driving gate changes by orders of magnitude, thereby, influencing the amount of coupling voltage. Obviously, such effects can accurately be modeled in SPICE, but due to the large circuit sizes, it is desirable to perform such analyses at the highest possible level of abstraction.

1.1 Gate Delay Calculation for Capacitive Loads

The gate propagation delay is divided into two terms: the intrinsic gate delay and the (external) gate load delay. The intrinsic gate delay is due to the native characteristics of the CMOS devices (e.g., transistors) in the gates/cells. More precisely, it is equal to gate propagation delay under zero load condition. The load delay captures the timing effect of the load on the gate propagation delay.

Figure 2(a) depicts a CMOS gate, which drives a purely capacitive load (C_L), where one of its inputs switches with a signal transition time of T_{in} causing the output of the gate to change. The gate propagation delay is a function of the input transition time and the output load. In commercial ASIC cell libraries, it is possible to characterize various output transition times (e.g. 10%, 50%, and 90%) as a function of the input transition time and output capacitance. i.e.,

$$t_\alpha = f_\alpha(T_{in}, C_L) \quad (1)$$

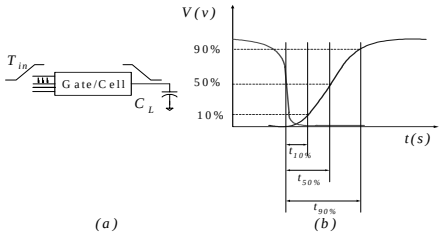

Figure 2: (a) A gate driving a capacitive load, (b) definitions of t_α terms

where α% denotes the percentage of the output transition, t_α is the output delay with respect to the 50% point of the input signal, and f_α is the corresponding delay function. The delay description function can be obtained in various ways. Two common approaches for the gate propagation delay computation are based on (1) the use of a Thevenin equivalent circuit for the driver, which is in turn composed of a voltage source and a series resistance, and (2) the delay tables. The first approach is difficult to deal with and is not as accurate as the second approach, which is currently in wide use especially in the ASIC design flow.

The algorithm for finding the output waveform of a gate is reviewed next. Given: 1) the capacitive load, C_L, 2) the input transition time, T_{in}, 3) the 50% transition point of the input waveform, δ; the output waveform is obtained as follows:

Draw_Output_Waveform (δ, T_{in}, C_L)
1. For α=10%, 50%, and 90% do
   t_α=**Calc_Delay** (δ, T_{in}, C_L, Table(T_{in}, C_L, α))
2. Draw the output waveform according to above data

Calc_Delay (δ, T_{in}, C_L, Table(T_{in},C_L,α))
1. From Table(T_{in},C_L,α) according to T_{in} and C_L, find the 50% input to α% output propagation delay, add δ to this value, and call it t_α
2. Return t_α

In VDSM technologies, we cannot neglect the effect of interconnect resistances of the load. Using the sum of all load capacitances as the capacitive load is a simple, yet quite pessimistic, approximation [7]. A more accurate approximation for an n^{th} order load seen by the gate/cell (i.e., a load with n distributed capacitances to ground) is to use a second order *RC*-π model [3, 5]. Equating the first, second, and third moments of the admittance of the real load with the first, second, and third moments of the *RC*-π load [19], we can find C_1, R_π, and C_2 as shown in Figure 3. It follows that for accurate gate delay calculation, we can use a four-dimensional delay table, where the dimensions are T_{in}, C_1, R_π, and C_2. However, this is costly in terms of storage and computational requirements. Therefore, the "effective capacitance" approach has been proposed [4,8] whereby the *RC*-π load is approximated by an equivalent capacitance. Consequently, it is possible to continue to use a two-dimensional table lookup to calculate the propagation delay to the output quantile point (i.e., 10%, 50%, and 90%).

The crosstalk capacitances affect the output transition time of each node of the load. Indeed, this effect has become more apparent in VDSM technologies

This research was supported in part by NSF under grant no. 9988441

where the coupling capacitances between interconnect lines have become quite important in terms of their relative magnitude compared with the area plus fringing capacitances of these lines. It is thus quite important to directly model the effect of the coupling capacitances on the gate output waveforms. An even more accurate approximation for this case is to consider *n* drivers where each of them drives an *RC*-π load. Obviously, there may be coupling capacitances between the near-end points and far-end points of the *RC*-π load as well as capacitive couplings between the near-end output terminals of each driver and the input terminals of the other drivers. The general load model for two drivers is shown in Figure 4 [3, 5,16, 17].

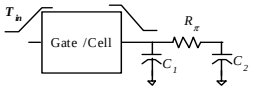

Figure 3: A gate/cell, which drives an *RC*-π calculated load

Given a complicated load with these two types of capacitive couplings (load-to-load and input-to-load couplings), one can use moment matching techniques to model the load and parasitic networks as an *RC* network depicted in Figure 4. In other words, given an extracted netlist, it is possible to calculate the precise values of all the *R* and *C* elements in Figure 4, where capacitances could be positive or negative. This calculation is not the focus of our paper, which starts by assuming that the various *R* and *C* values of the model of Figure 4 are already known. For accurate gate delay calculation, we may start from a multi-dimensional delay table, where the dimensions are the input transition times, the difference between the 50% transition points of the input waveforms, values of capacitances to ground, coupling capacitances, and shielding resistances, etc. However, this is extremely costly (if practical at all) in terms of the storage and computational requirements. Therefore, we ought to develop an alternative approach that allows us to avoid constructing and relying on such tables.

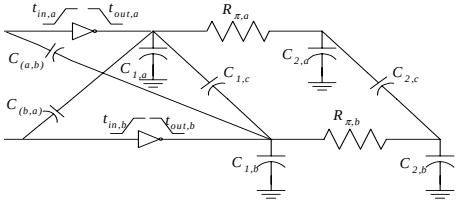

Figure 4: Two cross coupled CMOS gates each driving an RC-π load and exhibiting 1) interconnect coupling and 2) input-output coupling

In this paper, we present an algorithm, which calculates the output waveform of each driver driving a general load considering coupling capacitances in deep submicron technologies as shown in Figure 4. Using the algorithms, we can find the output waveform of each driver where the input transitions can have falling or rising behavior and could have overlap and etc. We show that our algorithm provides accurate results that are in a very good match for some important points of the output waveform of the CMOS drivers (e.g., 10%, 50% and 90%).

The remainder of this paper is organized as follows. Section 2 reviews previous works done to compute effective capacitance and describe our new algorithm to evaluate output waveform for *RC*-π load. Section 3 will solve the problem for the case of two drivers that drives a capacitive load in the presence of coupling capacitance. Section 4 will solve the problem for the case of resistive loads as shown in Figure 4. The results will be shown in section 5 and finally the conclusions are presented in Section 6.

2. Effective Capacitance Calculation

2.1 Prior Work

Many research results have been reported for calculating the interconnect propagation delay. These papers are simulation-based [6,7,9] or rely on analytical derivations [1,11]. Similarly and more recently, a number of research results have been published that focus on the loading effect of the *RC* wires on the gate propagation delay [8,9,12,13].

The effective capacitance is a function with two parameters: (1) characteristics of the output voltage waveform of the driving cell and (2) characteristics of the load, or more precisely, the driving point admittance of interconnects. If two drivers produce the same output waveform when they are faced with the same load, then the two cells are considered to be *equivalent* in terms of calculating C_{eff} [12]. Consider the circuit depicted in Figure 3. If R_π goes to infinity, then the gate/cell will see only C_1 as its load. On the other hand, if R_π goes to zero, then the gate/cell will see $C_{tot}=C_1+C_2$. Therefore, the effective capacitance that the driver experiences, can be written as:

$$C_{eff} = C_1 + kC_2 \quad \text{where} \quad 0 \le k \le 1 \qquad \textbf{(2)}$$

By using a table of circuit simulation results and a pair of two-dimensional delay tables, Macys *et al.* [12] calculated a value for the effective capacitance. In their work, the effective capacitance is a function of the total capacitance in the *RC*-π model (C_{tot}), the gate output slew rate, and the Elmore delay [1] of the load. The authors approximate the *RC*-π load with an effective capacitance such that the output voltage waveforms of the driving cell passes through some critical voltages (e.g., 0 and $0.75V_{dd}$) at the same instances in time. They also normalize the four model parameters (output slew time and three π model parameters) to two parameters and use a table of circuit simulation results to find the effective capacitance by exploiting an iteration-based procedure. However, Macys' approach is not based on any analytical derivation and is very sensitive to the simulation table entries.

Using a two-piece output waveform, Qian *et al.* propose an effective capacitance calculation approach that approximates the output waveform of a single-stage gate [10]. The authors calculate the effective capacitance by equating the charges at the gate output when using the driving-point admittance of the load and using a single effective capacitance as the load. Average charges for both loads models are equated until the gate output voltage reaches the 50% threshold. Qian's effective capacitance approach is costly in terms of its computational calculations and requires a large number of iterations (e.g., 5 to 10 iterations). It also involves empirical equations that assume fast input transitions.

Kahng and Muddu [11,13] propose a number of effective capacitance algorithms. In their latest approach [13], they state that by using the voltage of output pin of the gate/cell, they can find a non-iterative and fast method for calculating the effective capacitance that accurately matches the output waveforms in a range from $0.3V_{dd}$ to $0.6V_{dd}$. However, finding the output transition time (from the complex set of equations that the authors present) can be very costly. Furthermore, in reality, the driver resistance in their model is a function of the output load and input transition time and can thus vary greatly. However, the authors use a single value for the driver resistance corresponding to the case that the driver sees the total capacitances of a load.

In [19] authors calculate an effective capacitance, which approximately matches both 50% propagation delay and output transition time with reasonable accuracy. Their approach is analytical and has good performance. However, their analytical expressions can be applied to low-order circuits and are not suitable for high-order circuits (with more storage elements).

Furthermore, the previous papers do not prove the convergence of their iterative algorithms. In this paper, we propose an optimal lookup-table based effective capacitance algorithm for both speed and accuracy; such that using the capacitances; we can approximate the voltage waveform of the output terminal of the gates, which is less sensitive with respect to lookup table.

2.2 A New Effective Capacitance Calculation Algorithm

Problem Statement: Given is a CMOS driver where its input rise time is T_{in}, and an output load modeled by an *RC*-π circuit which is shown in Figure 6, find the output waveform of the driver.

Due to the shielding effect of the resistance, the effective capacitance can be written as C_1+kC_2 where $0<k<1$. Our algorithm gives an iterative approach for calculating *k*. Consider a unit step voltage source that drives an *RC* circuit in Figure 5(a). The current flowing into the *RC* circuit in Laplace domain is calculated as [18]:

$$I(s) = V(s)Y(s) = \frac{C}{sRC+1} \qquad \textbf{(3)}$$

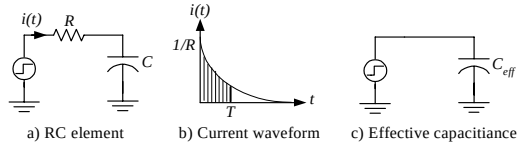

Figure 5: Effective capacitance concepts

The total charge induced into the capacitance up to time *T* is equal to the area, which is shown in Figure 5(b). In this case, it can be calculated as:

$$q(t) = \int_0^T i(t)dt = \int_0^T \frac{1}{R} e^{\left(-t/RC\right)} dt = C(1 - e^{\left(-T/RC\right)}) \quad \textbf{(4)}$$

We can replace the *RC* load with a single effective capacitance and calculate the amount of charge dumped into this capacitance for the same unit step input as shown in Figure 5(c). By matching the charge dumped into this load with Equation (4), we have:

$$C_{eff} = C(1 - e^{\left(-T/RC\right)}) \quad \textbf{(5)}$$

According to Equation (5), C_{eff} depends on the time up to which the charge is matched as well as the *R* and *C* values. The same observation holds for a ramp input [20].

According to the above discussion, the effective capacitance for an *RC*-π model load (depicted in Figure 6) can be written as:

$$C_{eff} = C_1 + (1 - e^{\left(-k\, t_{out}/R_\pi C_2\right)})C_2 \quad \textbf{(6)}$$

where *k* is a dimensionless constant and t_{out} is the gate output transition time. Macys [12] showed that the effective capacitance calculation is a function of only three parameters α, β, and γ, where;

$$\alpha = C_1 \Big/ (C_1 + C_2) \qquad \beta = t_{out} \Big/ R_\pi C_2 \qquad \gamma = C_{eff} \Big/ (C_1 + C_2) \quad \textbf{(7)}$$

In addition, Macys showed that if we provide the table that relates these three variables for each technology, where the table is independent of the input transition time and the gate configuration, the table could be used for any different combinations of load parameters. Therefore;

$$C_{eff} = C_1 + (1 - e^{-\frac{k\, t_{out}}{R_\pi C_2}})C_2 = (C1 + C2)\left[\alpha + \left(1 - e^{-k\beta}\right)(1 - \alpha)\right] \quad \textbf{(8)}$$

which results in

$$\frac{C_{eff}}{C_1 + C_2} = \gamma = \alpha + \left(1 - e^{-k\beta}\right)(1 - \alpha) \quad \textbf{(9)}$$

If we replace the output transition time (t_{out}) by 50% propagation delay we can rewrite the equation as:

$$\frac{C_{eff}}{C_1 + C_2} = \gamma = \alpha + \left(1 - e^{-k_t \beta'}\right)(1 - \alpha) \quad \textbf{(10)}$$

where β' is the ratio between the 50% propagation delay and the $R_\pi C_2$ product, and k_t is a fixed value which can be obtained from a lookup table (compiled from circuit simulation results) and which is constant for the calculated α and β'. Figure 7 reports the k_t values for different α and β' values in a 0.1µm CMOS technology. The figure shows that the table of k_t values is only a function of α and β' and remains constant for three different configurations of the gate and input waveform. It should be noted that k_t is decreasing with respect to α and β' and achieves its maximum at the minimum values of α and β', as seen in Figure 7. The advantage of this approach, with respect to Macy's approach, is that first; we have an equation that helps us understand the behavior of the effective capacitance. Second, by writing the sensitivity function of the output transition time with respect to k_t (for our approach) and γ (for Macys' approach), it can be proven that our approach results in a more stable effective capacitance estimation. More precisely,

$$H_\gamma^{t_{out}} = \frac{\frac{\partial t_{out}}{t_{out}}}{\frac{\partial \gamma}{\gamma}} = \frac{(\partial t_{out})/(t_{out})}{\left(\partial \frac{C_{eff}}{C_1 + C_2}\right)\Big/\left(\frac{C_{eff}}{C_1 + C_2}\right)} = = \frac{\frac{\partial t_{out}}{t_{out}}}{\frac{\partial C_{eff}}{C_{eff}}} = \frac{\partial t_{out}}{\partial C_{eff}} \cdot \frac{C_{eff}}{t_{out}} \quad \textbf{(11)}$$

$$H_{k_t}^{t_{out}} = \frac{\frac{\partial t_{out}}{t_{out}}}{\frac{\partial k_t}{k_t}} = \frac{\partial t_{out}}{\partial k_t} \cdot \frac{k_t}{t_{out}} = \frac{\partial t_{out}}{\partial C_{eff}} \cdot \frac{\partial C_{eff}}{\partial k_t} \cdot \frac{k_t}{C_{eff}} \cdot \frac{C_{eff}}{t_{out}} \quad \textbf{(12)}$$

$$= \frac{\partial t_{out}}{\partial C_{eff}} \cdot \frac{C_{eff}}{t_{out}} \cdot \frac{\partial C_{eff}}{\partial k_t} \cdot \frac{k_t}{C_{eff}} = H_\gamma^{t_{out}} \cdot \frac{\partial C_{eff}}{\partial k_t} \cdot \frac{k_t}{C_{eff}}$$

where: $$coeff = \frac{\partial C_{eff}}{\partial k_t} \cdot \frac{k_t}{C_{eff}} = \frac{{k_t}^2 \beta'(1 - \alpha)e^{-k_t \beta'}}{\alpha + (1 - e^{-k_t \beta'})(1 - \alpha)} \quad \textbf{(13)}$$

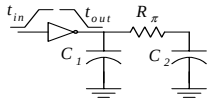

Figure 6: An inverter driving an *RC*-π load

Experimental results show that coefficient ***"coeff"*** is strictly smaller than one. In particular, the results for 30 different cases in 0.1µm technology are tabulated in Figure 8, which show that this coefficient is always less than one. We conclude that our approach is less sensitive compared to Macy's approach. To make the equations hold for 10% output transition point, we ought to derive a new lookup table for the k_t. The same holds for the 30%, 70%, and 90% output percentile points.

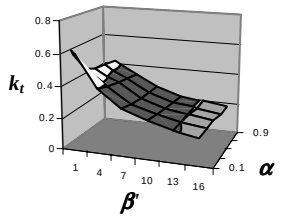

Figure 7: k_t for 50% propagation delay as a function of α and β'

In order to solve the proposed problem in section 5, we need to have the voltage waveform of the far-end capacitance for the *RC*-π load. If we apply a ramp input with rise time T_r to an *RC* load, the output waveform equation would be:

$$Vout(t) = \begin{cases} \frac{V_{dd}\left(RCe^{-t/RC} + t - RC\right)}{T_r} = \frac{V_{dd}RC}{T_r}(e^{-t/RC} + t/RC - 1) & 0 < t \le T_r \\ \frac{V_{dd}\left(RCe^{-t/RC} - RCe^{-(t-T_r)/RC} + T_r\right)}{T_r} = \frac{V_{dd}RC}{T_r}(e^{-t/RC} - e^{-(t-T_r)/RC} + T_r/RC) & t > Tr \end{cases} \quad \textbf{(14)}$$

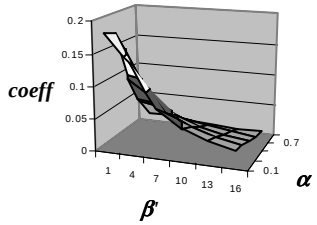

Figure 8: Sensitivity ratio between our approach and Macys' approach versus α and β'

where *t* is the time variable and V_{dd} is the final value of the input voltage waveform as shown in Figure 9. Therefore, to find the delay for the time it takes for the output waveform to reach the α percentile point (t_α), we need to solve the following nonlinear equation:

$$V_{out}(t_\alpha) = \alpha V_{dd} \quad \textbf{(15)}$$

Instead, according to Equation (14), if T_r/RC values of two different circuit configurations are equal, their t_α/RC values are also equal. Therefore, instead of solving the nonlinear Equation (15), we can make a table of delays, where for each T_r/RC, for each α percentile point, we have t_α/RC. For example, suppose we have a circuit where its input transition time is T_r and it drives an *RC* load. We need to find the time that the output waveform reaches its 50% of V_{dd}. Then from its corresponding table in Figure 9, for the T_r/RC, we find t_α/RC and thus, t_{50}.

Our algorithm for calculating the output waveform is as follows. Given the following information for a particular timing path of a cell; the input slew time, T_{in}, the π-load model parameters,(C_1,R_π,C_2), gate propagation delay from 50% transition point of input waveform to α% transition point of the output waveform, *Table(50%-α%)*, the k_t table, *Table*(k_t), we perform the following steps:

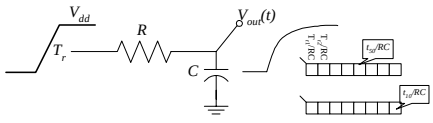

Figure 9: *RC* propagation delay calculation based on Table lookup

Draw_for_*RC*-π_Load (T_{in}, Load Parameters)

1. For α=10, 50, and 90 do
   a. **Find_Transition_Point** (T_{in}, C_1, R_π, C_2, Table (50%-α%),Table(k_t))
2. Draw output waveform according to the results

Find_Transition_Point (t_{in} ,C_1,C_2,R_π,Table(50%-α%), Table(k_t))

1. Guess an initial value for C_{eff}
2. Compute α from the load values
3. Obtain t_α from Table (50%-α%) based on values of C_{eff} and t_{in}
4. Compute β' from t_α and load elements
5. Find k_t from Table (k_t) according to α and β'
6. Calculate C_{eff} from Equation (10)
7. Find the new value of t_α for the obtained C_{eff} from Table(50%-α%)
8. Compare the new t_α with the old t_α
9. If not within acceptable tolerance, then return to step 4 until t_α converges
10. Return t_α

Experimental results demonstrate that this algorithm gives accurate results with fast convergence. Next, we prove that the algorithm converges independently of the initial guess.

Theorem 1: Iterative Equation (10) always converges independently of the initial guess. Furthermore, its solution is unique.

Proof: Per reference [23], the iterative equation for finding the solution to $x=f(x)$ will converge for any initial input and its solution is unique, if

$$\left|\frac{d}{dx} f(x)\right| < 1 \tag{16}$$

In this case, to prove the convergence of the equation, we prove:

$$\left|\frac{d}{dC_{eff}}\left(C_1 + (1-e^{-k_t\beta'})C_2\right)\right| = \left|\frac{d(k_t\beta')}{dC_{eff}}\left(e^{-k_t\beta'}\right)C_2\right| < 1 \tag{17}$$

Because

$$\begin{aligned}\left|\frac{d(k_t\beta')}{dC_{eff}}\left(e^{-k_t\beta'}\right)C_2\right| &< \left|\frac{d(k_t\beta')}{dC_2}\cdot\frac{dC_2}{dC_{eff}}\left(e^{-k_t\beta'}\right)C_2\right| \\ = \left|k_t\beta'\frac{dC_2}{dC_{eff}}\left(e^{-k_t\beta'}\right)\right| &\cong \left|k_t\beta'\left(e^{-k_t\beta'}\right)\right| < 1 \qquad for\ \forall(k_t\beta') > 0\end{aligned} \tag{18}$$

Therefore, the proposed iterative effective capacitance equation always converges to its unique solution, independently of initial value of C_{eff}. ■

3. Crosstalk for Coupled Capacitive Loads

Problem Statement: Two CMOS drivers, a and b, are given where their corresponding input transition times are $t_{in(a)}$ and $t_{in(b)}$ and there is a $\delta=\delta_b-\delta_\alpha$ delay between their input waveforms where δ_α and δ_b denote the 50% transition points of the input waveforms of driver a and b, respectively. Also, the corresponding capacitive loads are C_a and C_b and there is a capacitive coupling between the two output loads with value C_c. Furthermore, the output waveform of drivers a and b are $t_{out(a)}$ and $t_{out(b)}$, respectively. The objective is to find the output waveform of the two drivers. In fact, we must solve a nonlinear equation:

$$S(T_{in}, C_L, C_C, T_{out}) = 0 \tag{19}$$

where;

$$T_{in} = \begin{bmatrix} t_{in(a)} \\ t_{in(b)} \end{bmatrix} \quad T_{out} = \begin{bmatrix} t_{out(a)} \\ t_{out(b)} \end{bmatrix} \quad C_L = \begin{bmatrix} C_a \\ C_b \end{bmatrix} \quad C_C = \begin{bmatrix} C_c \end{bmatrix} \tag{20}$$

This is, however, a difficult undertaking. Thus, we look for a better solution. Many different scenarios could arise under this problem statement. For example, the input voltages may independently have a falling or rising transition; there could be a non-zero positive or negative skew between their 50% input transition times, the slew rates of the two inputs can widely differ, etc. According to circuit theory, we can model the coupling capacitance by a Miller capacitance to ground [21] for each scenario. A simple approximation can be obtained as follows. Taking the circuit in Figure 10 as a two-port network, in order to model the coupling capacitance as an equivalent capacitance to the ground, we suppose that the equivalent circuit has the same current sink and the same voltage waveforms at the output terminals of the drivers. Using this assumption, we have [22]:

$$I_a = -I_b = C_c\frac{d(V_a - V_b)}{dt} = C_{eff,a}\frac{dV_a}{dt} = -C_{eff,b}\frac{dV_b}{dt} \tag{21}$$

To calculate the effective capacitance to ground for driver **a**, by integrating the current over the period from the rising time of the output of driver **a**, to the switching threshold point, t_{th}, we have;

$$\begin{aligned}\Delta Q &= \int_{t_0}^{t_{th}} I_a dt = \int_0^{V_{th}} C_c dV_a - \int_{V_b^0}^{V_b^1} C_c dV_b = \int_0^{V_{th}} C_{eff,a} dV_a \\ \Rightarrow C_{eff,a} &= C_c\left(1 - \frac{\Delta V_b}{V_{th}}\right) \qquad \Delta V_b = V_b^1 - V_b^0\end{aligned} \tag{22}$$

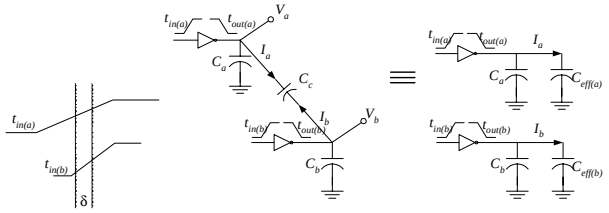

Figure 10: Two gates driving capacitive load with capacitive crosstalk

where ΔV_b is the voltage transition of voltage waveform of the output of the driver **b**, when the voltage waveform of the output of the driver **a** transits from 0 to a voltage threshold at which we are interested in finding the propagation delay. For example, if the effective capacitance to ground of inverter **a** for calculating 50% output transition point is needed, then ΔV_b becomes the output voltage transition of driver **b**, from when the output waveform of driver **a** transits from 0 to its 50% transition point as shown in Figure 12. The same calculation may be performed for driver **b**. Therefore;

$$C_{eff,b} = C_c\left(1 - \Delta V_a / V_{th}\right) \tag{23}$$

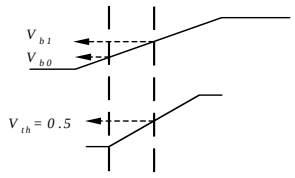

Figure 11: Definition of voltages for Equations (22) and (23)

Therefore, the algorithm is as follows:

Find_Waveforms_Capacitive_Cross_Coupled($\delta_{(a)}$,$\delta_{(b)}$,$t_{in(a)}$, $t_{in(b)}$,C_a,C_b C_c)

1. Guess an initial value for output capacitive load and put it $C_{L(a)}$ and $C_{L(b)}$ (for example $C_{L(a)}=C_a+C_c$)
2. $t_{out,a}$=**Draw_Output_Waveform**($\delta_{(a)}$,$t_{in(a)}$,$C_{L(a)}$)
3. $t_{out,b}$=**Draw_Output_Waveform**($\delta_{(b)}$,$t_{in(b)}$,$C_{L(b)}$)
4. Repeat until the output waveform converges

a. For ($V_{th,a}$,$V_{th,b}$)={(50%,50%),(50%,90%), (90%,50%), (50%,10%), (10%,50%)} do
Find_Output($t_{out,a}$,$t_{out,b}$,$V_{th,a}$,$V_{th,b}$,$C_{L(a)}$, $C_{L(b)}$,C_c)

Find_Output($t_{out,a}$,$t_{out,b}$,$V_{th,a}$, $V_{th,b}$, $C_{L(a)}$, $C_{L(b)}$, C_c)

1. $C_{L(a)} \leftarrow C_{L(a)} + C_c\left(1 - \frac{\Delta V_b}{V_{th,a}}\right)$ And $C_{L(b)} \leftarrow C_{L(b)} + C_c\left(1 - \frac{\Delta V_a}{V_{th,b}}\right)$
2. Update $t_{out,a}$ and $t_{out,b}$
3. If $t_{out,a}$ and $t_{out,b}$ *tolerance* are within acceptable range, then
a. Return $t_{out,a}$ and $t_{out,b}$
4. **Find_Output** ($t_{out,a}$,$t_{out,b}$,$V_{th,a}$,$V_{th,b}$,$C_{L(a)}$,$C_{L(b)}$,C_c)

This algorithm can easily be extended to handle a collection of *N* cross-coupled drivers. In practice, we may encounter cases where the output waveform behaves like the voltage waveform shown in Figure 12. In such a case, we need to generate delay estimates for more percentile points of the output waveform. As shown in Figure 12, if we apply curve fitting for upper points and lower points, we can predict the output waveform of the gate. This technique works fine if the magnitude of distortion does not exceed over some threshold voltages which for each horizontal lines drawn in Figure 12, we have more than one point. At this point we need to prove that the proposed algorithm converges to a unique solution independently of the initial guess for the effective capacitance to ground.

Theorem 2: The "Find_Waveforms_Capacitive_Cross_Coupled**"** algorithm converges to its unique solution independently of the initial guess for the value of the effective capacitance to ground.

Proof: In this algorithm, instead of attempting to solve the rather complicated Equation 19, we solve the problem by an iterative technique, which can be described as follows:

$$\begin{cases} T_{out} = F(T_{in}, C_L + C_{EFF}) \\ C_{EFF} = G(C_C, T_{out}) \end{cases} \Rightarrow T_{out} = F(T_{in}, C_L + G(C_C, T_{out})) \quad \textbf{(24)}$$

To prove that the above equation converges, we use Equation 16. More precisely, we show that [23]:

$$\left| \frac{\partial}{\partial T_{out}} \left(F(T_{in}, C_L + G(C_C, T_{out})) \right) \right| < 1 \quad \textbf{(25)}$$

In addition, we assume the *G* and *F* functions have the following forms:

$$G(C_C, T_{out}) = C_c \begin{bmatrix} 1 - k_{g1} t_{out(a)} \\ 1 - k_{g2} t_{out(b)} \end{bmatrix} \qquad F(T_{in}, C_L + C_{eff}) = \begin{bmatrix} k_{f1}\left(C_a + C_c\left(1 - k_{g1} t_{out(a)}\right)\right) \\ k_{f2}\left(C_b + C_c\left(1 - k_{g2} t_{out(b)}\right)\right) \end{bmatrix} \quad \textbf{(26)}$$

where k_{fi} (i=1,2) is the rate of output transition time change to output load change for the i^{th} driver and k_{gi} (i=1,2) is the ratio of Miller factor changes to output transition time changes. Therefore;

$$\left| \frac{\partial}{\partial T_{out}} \left(F(T_{in}, C_L + G(C_C, T_{out})) \right) \right| = \left| \begin{bmatrix} \frac{\partial F_a}{\partial t_{out(a)}} & \frac{\partial F_a}{\partial t_{out(b)}} \\ \frac{\partial F_b}{\partial t_{out(a)}} & \frac{\partial F_b}{\partial t_{out(b)}} \end{bmatrix} \right| \quad \textbf{(27)}$$

$$= \left| \frac{\partial F_a}{\partial t_{out(a)}} \bullet \frac{\partial F_b}{\partial t_{out(b)}} - \frac{\partial F_a}{\partial t_{out(b)}} \bullet \frac{\partial F_b}{\partial t_{out(a)}} \right| = \left| k_{f1} k_{f2} C_c{}^2 k_{g1} k_{g2} \right|$$

The worst-case value of k_{fi} (i=1,2) is when the drivers are weak, where in 0.1µm technology is in the order of 10^3(*S/F*). Also according to [21], the miller factor could vary from –1 to 3, therefore, for the worst-case in 0.1µm technology, k_{gi} (i=1,2) is in the order of 10^7(*1/S*) and the coupling capacitance is in the order of 10^{-11}(*F*) in the worst-case. Therefore, the condition in Equation (27) always holds and the product is always less than 1, which proves the convergence of the iterative algorithm. ■

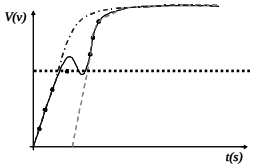

Figure 12: Approximating the output waveform with curve fitting in the case of non-monotone response

4. Crosstalk for Coupled *RC* Loads

Problem Statement: The problem statement is the same as the one in Section 3 except that the load is now the one that is depicted in Figure 13. We are interested in determining the output waveforms at the near ends.

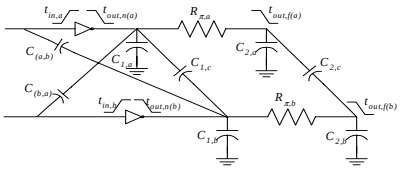

Figure 13: A general format of two gates driving resistive and capacitive loads considering crosstalk

Empirical gate/cell level models remain popular for timing analysis, even for full custom designs. In [14] a gate/cell level modeling methodology was developed which achieves compatibility with *RC* interconnect loading through an "effective capacitance" approximation. Dartu and Pileggi in [15] extended this waveform-based gate model to consider the problem of calculating the delay (and response waveform) when there is a significant amount of coupling. In particular, they present algorithms for obtaining the best and the worst gate delays in the presence of coupling capacitances. In their paper, the authors use a Norton equivalent model for the gates to do the analysis. What we do for this problem is to partition the circuit with the two cross-coupled drivers into two separate sub-circuits, each with a single driver and loaded by an RC-π load. By applying the "*Find_Output*" algorithm of Section 3, we first decouple the drivers and next by applying the "*Draw_for_RCπ_Load*" algorithm of Section 2, we estimate the output waveforms of the two drivers. Finally, we go thru a number of iterations to determine the exact output waveform of each sub-circuit. The algorithm to find the output waveforms is as follows:

Find_Output_Waveforms ($t_{in,a}$,$t_{in,b}$, Load Parameters)
1. Model each coupling capacitance as a capacitance to ground
2. ($t_{out,n(a)}$,$t_{out,f(a)}$)=**Draw_for_RCπ_Load**($t_{in,a}$, Load Parameters)
3. ($t_{out,n(b)}$,$t_{out,f(b)}$)=**Draw_for_RCπ_Load**($t_{in,a}$, Load Parameters)
4. Repeat
   a. For ($V_{th,a}$,$V_{th,b}$)={(50%,50%),(50%,90%), (90%,50%), (50%,10%), (10%,50%)} do
      1. **Update_Voltage_Waveforms**(Voltage Waveforms, $V_{th,a}$, $V_{th,b}$, Load Parameters)
5. Until the output waveforms converges

Update_Voltage_Waveforms (Voltage Waveforms, $V_{th,a}$, $V_{th,b}$, Load Parameters**)**
1. Update equivalent Miller capacitance values by using Equations (22) and (23)
2. **Draw_for_RCπ_Load (**$t_{in,a}$, Load Parameters**)**
3. **Draw_for_RCπ_Load (**$t_{in,b}$, Load Parameters**)**
4. If voltage waveforms are within acceptable tolerance, then return values
5. **Update_Voltage_Waveforms (**Voltage Waveforms, $V_{th,a}$, $V_{th,b}$, Load Parameters**)**

The proof of convergence for this iterative approach is similar to the proof in Section 3. It is omitted here due to space limitation.

5. Experimental Results

We performed a large number of simulations on different circuits in 0.1µm CMOS technology and report the results here. We considered different ranges of coupling capacitances, driver sizes and loads. In Table 1, we compare the results of the algorithm proposed in Section 2, "*Draw_for_RCπ_Load*", with those obtained from Hspice simulations for three different percentile points of the output transition time. The increments for α and β' of the k_t table were taken as 0.1 and 1, respectively. Table 1 shows that our algorithm comes within 1% of Hspice. In Table 2, we compare the results of the algorithm proposed in Section 3, "*Find_Waveforms_Capacitive_Cross_Coupled*", with the results obtained by Hspice. We achieved a mere 3% error for different cases. In Table 3, we applied the algorithm proposed in Section 4 to the complex load configuration of Figure 13 and compared its results with Hspice. Again, we observed a small error (about 6% on average.). These results are reported after 3 iterations. Also, in the experiments, the $k=C_{eff}/(C_1+C_2)$, varies from 0.15 to 0.90.

6. Conclusion

As we go toward VDSM technologies, the effect of interconnect resistance and coupling capacitance must be carefully taken into account. The interconnect resistance reduces the cell delay via shielding the far-end capacitances, whereas, the coupling capacitances increases the gate propagation delay. Gate load delay calculation requires accuracy, and using delay tables is essential for accurate delay calculation for given capacitive load and input transition times. The gate delay can widely vary as a function of the input transition times, driver strengths, and the skew between the transitions, and the output load configurations. In this paper, we presented three efficient iterative algorithms with provable convergence property, which have low computational complexity and result in highly accurate results. To use the delay tables, we approximated the load with an effective capacitance, which is equivalent to the real load in terms of its propagation delay (at 10%, 50%, and 90% percentile points). The algorithms proposed for calculating gate propagation delay results in high accuracy with 6% error on average.

7. 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] J. K. Ousterhout, "A Switch-level Timing Verifier for Digital MOS VLSI," *IEEE Trans. on Computer Aided Design of VLSI Circuits and Systems*, vol. 4 (1985), pp. 336-349.

[3] P. R. O'Brien and T. L. Savarino, "Modeling the Driving-Point Characteristic of Resistive Interconnect for Accurate Delay Estimation," *Proc. of IEEE Int'l Conf. on Computer Aided Design*, 1989, pp. 512-515.

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

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

[6] L. T. Pillage and R. A. Rohrer, "Asymptotic Waveform Evaluation for Timing Analysis," *IEEE Trans. on Computer Aided Design of VLSI Circuits and Systems*, vol. 9 (1990), pp. 352-366.

[7] C. L. Ratzlaff, N. Gopal, and L. T. Pillage, "RICE: Rapid Interconnect Circuit Evaluator," *Proc. of 28th ACM/IEEE Design Automation Conf.*, June 1991, pp. 555-560.

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

[9] M. Sriram and S. M. Kang, "Fast Approximation of the Transient Response of Lossy Transmission Line Trees," *Proc. of 30th ACM/IEEE Design Automation Conf.*, June 1993, pp. 691-696.

[10] J. Qian, S. Pullela, and L. Pillage, "Modeling the "Effective Capacitance" for the RC Interconnect of CMOS gates," *IEEE Trans. on Computer Aided Design of VLSI Circuits and Systems*, vol. 13 (1994), pp. 1526-1535.

[11] A. B. Kahng and S. Muddu,' "Accurate Analytical Delay Models for VLSI Interconnects," *Proc. of IEEE International Symposium on Circuits and Systems*, May 1996.

[12] R. Macys, S. McCormick, "A New Algorithm for Computing the "Effective Capacitance" in Deep Sub-micron Circuits," *Proc. of IEEE Custom Integrated Circuits Conference*, June 1998, pp. 313-316.

[13] A. B. Kahng and S. Muddu,' "Improved Effective Capacitance Computations for Use in Logic and Layout Optimization," *Proc. of the 12th International Conference on VLSI Design*, 1999, Pages: 578 -582.

[14] F. Dartu, N. Menzes, and L.T. Pileggi," Performance computation for pre-characterized CMOS gates with RC loads," *IEEE Transaction on CAD*, vol. 15, pp. 544-553, May 1996

[15] F. Dartu, L.T. Pileggi, "Calculating worst-case gate delays due to dominant capacitance coupling," *Proc. Of Design Automation Conference*, pp 46 –51, June 1997

[16] I.D. Huang, S.K. Gupta, and M.A. Breuer, "Accurate and Efficient Static Timing Analysis with Crosstalk," *Proc. Of International Conference on Computer Design*, pp: 265 –272, 2002

[17] J.Cong, D.Z. Pan, and P.V. Srinivas, "Improved crosstalk modeling for noise constrained interconnect optimization," *Proc of* Asia and South Pacific Design Automation Conference, pp: 373 –378, 2001

[18] C.V. Kashyap, C.J. Alpert, A. Devgan, "An effective capacitance based delay metric for RC interconnect" *Proc. of International Conference on Computer Aided Design*, pp: 229 – 234,2000

[19] S. Abbaspour, M. Pedram, "Calculating the effective capacitance for the RC interconnect in VDSM technologies," *Proc. of Asia and South Pacific Design Automation Conference*, January 2003.

[20] R.Puri, D.S. Kung, A.D. Drumm, "Fast and Accurate Wire Delay Estimation for Physical Synthesis of Large ASICs," *Proc. of GLSVLSI*, 2002

[21] P. Chen, D.A. Kirkpatrick, K. Keutzer, "Miller Factor for Gate-Level Coupling Delay Calculation", *Proc. of ICCAD*, pp 68-74, 2000

[22] J. M. Rabaey, Digital integrated circuits: a design perspective, Upper Saddle River, NJ: Prentice Hall, 1996.

[23] http://mathworld.wolfram.com

Table 1: Simulation results for output waveform evaluation algorithm for RCπ load (proposed in Section 2) for 0.1µm technologies (3 iterations).

Driver and Load Parameters					10% propagation delay (from 50% of input to 10% of output)			50% propagation delay (from 50% of input to 50% of output)			90% propagation delay (from 50% of input to 90% of output)		
T_{in} (pS)	$W_p(\lambda)$/ $W_n(\lambda)$	C_1(fF)	R(Ω)	C_2(fF)	Our App. (*pS*)	Hspice (*pS*)	Error %	Our App. (*pS*)	Hspice (*pS*)	Error%	Our Approach (*pS*)	Hspice (*pS*)	Error%
400	100/50	500	100	900	169	165	2.4	711	703	1.1	1705	1696	0.5
200	150/75	1500	200	400	147	144	2.1	640	635	0.8	1515	1491	1.6
100	50/25	250	80	1000	190	186	2.1	1186	1168	1.5	2960	2890	2.4
100	150/75	1000	500	1250	90	87	3.4	486	476	2.1	1825	1802	1.3
600	80/40	350	150	1000	199	193	3.1	845	834	1.3	2095	2080	0.7
300	200/100	1000	450	1650	102	105	2.8	401	392	2.3	1650	1647	0.1
50	200/100	1500	650	2050	85	84	1.2	461	456	1.1	2047	2037	0.4
250	20/10	450	350	650	426	419	1.7	2586	2572	0.5	6420	6348	1.1
150	160/80	850	1000	1500	86	82	4.9	349	336	3.9	1642	1621	1.3
550	150/75	1300	400	1900	185	181	2.2	721	707	2.0	2560	2495	2.6
350	120/60	1600	500	1500	205	203	0.9	1012	1000	1.2	3145	3097	1.5
450	30/15	500	100	600	149	142	4.9	515	503	2.4	1190	1149	3.7
Avg.							**2.6**			**1.7**			**1.4**

***All inputs are rising.

Table 2: Simulation results for capacitive load considering crosstalk (cf. section 3) for 0.1µm technologies (3 iterations).

Driver A			Driver B			Load Parameters			50% propagation delay of driver A			50% propagation delay of driver B		
T_{in} (ps)	$W_p(\lambda)$/ $W_n(\lambda)$	50% Coord. (pS)	T_{in} (ps)	$W_p(\lambda)$/ $W_n(\lambda)$	50% Coord. (ps)	Ca (fF)	Cb (fF)	Cc (fF)	Our Appoach (*pS*)	Hspice (*pS*)	Err. %	Our Approach (*pS*)	Hspice (*pS*)	Err%
100	100/50	50(R)	200	150/75	150 (F)	500	400	500	645	637	1.3	465	458	1.5
100	100/50	50(R)	200	150/75	250 (F)	500	400	500	570	550	3.6	475	464	2.4
100	100/50	50(F)	200	150/75	150 (F)	500	400	500	281	271	3.7	185	177	4.5
200	150/75	100(F)	350	250/125	750 (F)	1000	900	1500	705	697	1.1	253	241	4.9
200	150/75	100(R)	350	250/125	750 (F)	1000	900	1500	602	595	1.2	681	670	1.6
350	120/60	175(F)	150	80/40	650 (R)	700	800	500	521	505	3.2	995	975	2.1
350	120/60	175(R)	150	80/40	650 (R)	700	800	500	415	405	2.5	402	391	2.8
400	250/125	200(R)	350	120/60	500 (R)	1100	900	1000	341	325	4.9	286	279	2.5
400	250/125	200(F)	350	120/60	900 (R)	1100	900	1000	356	346	2.9	1189	1149	3.5
Avg.											**2.7**			**2.9**

***F: Falling input, R: Rising input

Table 3: Simulation results for general load considering crosstalk (cf. Section 4) for 0.1µm technologies (3 iterations).

Driver A			Driver B			Load Parameters										50% propagation delay of driver A			50% propagation delay of driver B		
T_{in} (ps)	$W_p(\lambda)$/ $W_n(\lambda)$	50% Coor. (pS)	Tin (ps)	$W_p(\lambda)$/ $W_n(\lambda)$	50% Coor. .(pS)	$C_{1,a}$	$R_{1,\pi}$	$C_{2,a}$	$C_{1,b}$	$R_{2,\pi}$	$C_{2,b}$	$C_{1,c}$	$C_{2,c}$	$C_{(a,b)}$	$C_{(b,a)}$	Our App (*pS*)	Hspice (*pS*)	Err (%)	Our App. (*pS*)	Hspic (*pS*)	Err (%)
100	100/50	50(R)	200	150/75	150(F)	300	100	800	800	300	999	400	500	100	150	675	630	7.1	1109	1050	5.6
100	100/50	50(F)	200	150/75	250(F)	600	200	400	500	400	800	450	650	200	350	985	945	4.2	470	442	6.3
200	150/75	100(F)	350	250/125	750(F)	500	300	400	600	200	900	350	250	300	100	717	688	4.2	485	466	4.1
350	120/60	175(F)	150	80/40	650(R)	600	500	500	400	200	500	450	500	350	400	535	512	4.5	1430	1390	2.9
400	250/125	200(R)	350	120/60	500(R)	700	350	900	500	300	600	350	250	450	150	513	489	4.9	522	495	5.4
200	150/75	100(R)	350	250/125	750(F)	800	500	950	450	350	900	250	450	350	300	511	489	4.5	407	391	4.0
400	250/125	200(F)	350	120/60	900(R)	250	150	650	350	125	800	650	350	125	275	312	291	7.2	1345	1296	3.8
Avg.																		**5.2**			**4.6**

***F: Falling input, R: Rising input