

# Cell Delay Analysis Based on Rate-of-Current Change

Shahin Nazarian

Massoud Pedram

*University of Southern California  
Department of Electrical Engineering, Los Angeles, CA 90089*

[shahin@usc.edu](mailto:shahin@usc.edu)

[pedram@usc.edu](mailto:pedram@usc.edu)

**Abstract** - A cell delay model based on rate-of-current-change is presented, which accounts for the impact of the shape of the noisy waveform on the output voltage waveform. More precisely, a pre-characterized table of time derivatives of the output current as a function of input voltage and output load values is constructed. The data in this table, in combination with the Taylor series expansion of the output current, is utilized to progressively compute the output current waveform, which is then integrated to produce the output voltage waveform. Experimental results show the effectiveness and efficiency of this new delay model.

## 1. Introduction

The drastic down scaling of layout geometries to 90nm and below has resulted in a significant increase in the packing density and the operational frequency of VLSI circuits. An unfortunate side effect of this technology advancement has been the aggravation of noise effects, such as the capacitive crosstalk noise, in VLSI circuits. These is mainly because the metal wires have become narrower and thicker (and in fact longer in the case of global interconnects) and are laid out closer to one another, which in turn increases the capacitive coupling noise. Furthermore, IC manufacturing process variations, device/interconnect aging phenomena, and dynamic circuit parameter changes (such as power plane fluctuations and temperature gradients in the substrate) give rise to a rather significant deviation of the electrical parameters of the circuit components from their designed (nominal) values. This effect can produce excessive timing uncertainty, which in turn requires sophisticated crosstalk-aware delay analysis techniques and tools to overcome it.

Timing analysis is an essential aspect of determining whether a noise source can create a faulty output in a circuit. In particular, the signal arrival times in a circuit can change as a function of the noise that is present in the circuit. Gate-level timing analysis tools such as STA (static timing analysis), and SSTA (statistical static timing analysis) tools are used as efficient alternatives with an acceptable level of accuracy. These tools employ delay models for both interconnect lines and logic cells.

The function of an *interconnect delay model* is to take as input the transient waveform at the near-end of an interconnect line and produce as output, the corresponding waveform at the far-end of the line while accounting for the effect of various noise sources that couple to the line. This process is known as interconnect delay (or timing) analysis. Similarly, the function of a *cell delay model* is to take a noisy input waveform and produce the waveform for the cell output. This process is known as cell delay (or timing) analysis. Conventional timing analysis tools start with arrival time and slope (transition time or slew) at the near-end of an interconnect line and produce the arrival time and slew at the output of a cell that is driven by the far-end of that line.

The fact that the interconnect delay dominates the cell delay in modern VLSI circuits, has drawn attention toward producing faster and more accurate interconnect delay models. However the conventional logic cell delay models have not improved as much and their deficiencies, especially in handling noisy waveforms have been intensified due to recent technology trend. Consequently cell models are one of the main sources of inaccuracy in existing timing analysis tools. The focus of this paper is on cell delay modeling considering noise.

Cell delay is conventionally pre-characterized based on input slew and capacitive output load by using a circuit level timing analyzer such as Spice. Therefore the resulting pre-characterized look-up tables are inherently incompatible with the RC/RLC interconnect loads. This incompatibility is dispelled by finding an effective capacitive load, which is in some way equivalent to the more complex RC [1] or RLC load [2]. An iterative or non-iterative approach may be used to calculate the effective capacitance.

The goal of cell timing analysis is conventionally stated as: Given a noisy waveform at the input of a cell, find an equivalent input voltage waveform that when is applied to the cell generates an output waveform which is as close as possible to the output waveform in terms of its arrival time and slew. As the silicon technology is driven to nanometer, conventional voltage-based lookup tables cannot meet this goal any longer, mainly due to being inefficient in accurately considering the impact of the shape of the noisy waveform. Different voltage waveforms

with identical arrival time and slew at the input of a cell can result in very different propagation delays through that cell. This is because the exact shape of the input voltage waveform can greatly influence the cell output waveform behavior.

Generally speaking, as the crosstalk noise becomes more significant in current technologies, using only a reference point (arrival time) and a constant slope (slew) to convey the timing information for a signal transition adversely impacts the robustness of timing analysis tools. Additionally, the voltage-based timing analysis tools are inefficient in low power design styles that incorporate two or more logic “islands”, each running at a different operating voltage. Traditional library cell characterization that accurately covers a wide range of operating voltages can be prohibitively time consuming. In [3]-[4] the common voltage-based cell timing analyzers are reviewed and their shortcomings are highlighted.

To consider the shape of the waveform more effectively in this work, the problem is re-stated in a more general statement as follows: Given a noisy voltage waveform at the input of a cell, determine the output voltage waveform, which has the minimum error with respect to the actual output waveform. Current-based cell delay modeling has proven to be more successful than voltage-based logic cell timing analysis in achieving this goal [5]-[7]. In fact some industrial current-based timing analyzers, such as ECSM (Effective Current Source Model) and CCSM (Composite Current Source Model) are already in use [8].

In this paper, we present a rate-of-current-change (ROCC) based logic cell timing analyzer, which utilizes a pre-characterized table of the time derivatives of the output current waveform to compute the output current and subsequently the output voltage waveforms. The data in this table, together with the Taylor series expansion of the output current, is utilized to compute the output current waveform in a step-by-step manner. Having computed the output current, the output voltage waveform can be computed based on the output load.

To respond to the aforesaid more general problem, our model is able to directly build the output waveforms without the need for creating an equivalent input waveform as is done by conventional techniques. The characterization and application steps are simple and efficient to implement. Furthermore, the application of pre-characterized ROCC parameter values can accurately model the behavior of a logic cell as it receives a noisy input. Experimental results demonstrate that the ROCC-based delay calculator can accurately capture the impact of the shape of the input voltage waveform on the output current waveform and eventually the voltage waveform.

The remainder of this paper is arranged as follows. In section 2 the previous logic cell delay modeling techniques including the current-based ones are reviewed. Section 3 describes our current ROCC-based cell delay modeling.

Section 4 and 5 explain the experimental results and conclusions respectively.

## 2. Background

The cell delay modeling techniques can be classified into two general groups, voltage-based and current-based ones, according to which of the output current or output voltage they compute. On the other hand, cell delay models may apply lookup tables and/or equations.

Most of today’s logic cell delay models, which are used in integrated circuit design flows, consist of lookup tables or characteristic equations that rely on linear or ramp voltage waveforms and simplified loads as inputs and create linear or ramp voltage waveform approximations as output. Interested reader may refer to references [3]-[4] that extensively explain the various voltage-based cell delay models and their shortcomings and strengths. Two recently developed approaches, i.e., equation-based and current-based techniques, contend to replace voltage-based lookup tables. Both have the ability to better predict nanometer timing across a range of supply voltage [8].

### 2.1 Equation-based techniques

The equation-based timing analyzers generally use a polynomial with multiple coefficients relating timing to a variety of input parameters. The goal is to model delay variation due to environmental factors such as supply voltage and substrate temperature. However, it is difficult to fit the actual non-linear behavior of the timing quantity of interest with a polynomial that has a limited (and relatively small) number of terms.

In practice, the extreme effort to characterize real silicon to the equation-based modeling has made it unpopular. Sophisticated optimization algorithms are required to perform curve fitting of a polynomial to simulation data, and the accuracy and turnaround time of the library creation is limited by the quality of the optimization algorithms.

### 2.2 Current-based techniques

Current-based cell timing analyzers generally base their delay calculations on the amount of current flow into or out of a cell. Current-based cell modeling is much easier to characterize than the equation-based one. Rather than a mathematical abstraction, current-based modeling is a physical model patterned after the actual construction of transistors. It improves delay calculation accuracy by modeling a cell’s output drive as a current source rather than a voltage source. Current sources are more effective at tracking non-linear transistor switching behavior and permit highly accurate modeling of long complex interconnects, which are common in many of today’s largest nanometer low power designs.

One example of a current-based cell delay model is proposed in [7] where cells under the crosstalk-induced pulse (glitch) attack are modeled by using an analytical current model consisting of four parameters, namely a DC current source, a linear resistance, an output capacitance, and the internal delay of the gate.

Another current-based model, called Blade [5], consists of a voltage-controlled current source, an internal capacitance, and a time shift of the output waveform. First  $I_{out}(V_{in}, V_{out})$ , the amount of current sourced by a cell in response to DC voltage levels on the input and output pins of interest, is determined and a lookup table (denoted by the cell I-V table) is created for each cell by sweeping the DC values of input and output voltages and measuring the current sourced by the cell output pin. However, a response exclusively derived from the DC-based I-V table results in an overly optimistic timing analysis as the DC sweep of the input and output ignores the effects of parasitic elements. Therefore a calibration procedure is thus performed to consider the cell parasitic effects. This procedure determines an internal capacitive load which, when applied to the Blade model, results in a transient waveform that matches the shape of a Spice-generated waveform for the cell under identical conditions. Once the waveform shapes have been matched, a time shift is calculated by examining the time difference between the 50% points of the Spice output and the calibrated Blade output. A runtime engine consisting 31×31 I-V lookup tables and a secant iteration-based nonlinear solver is used to compute the output waveforms.

A more complete current-based cell delay technique is presented in [6], where the current drawn by a cell during the output switching is computed while considering the Miller effect between the input and output nodes along with the effect of internal parasitic capacitances. As a result, the current drawn by a cell during output switching is essentially represented by the following equation:

$$i_{out} = I(V_{in}, V_{out}) + C_g(v_{in}, v_{out}) \frac{\Delta v_{out}}{\Delta t} - C_M(v_{in}, v_{out}) \frac{\Delta v_{in}}{\Delta t} \quad (1)$$

The coefficients of the last two terms in Equation (1) capture the current charging an effective capacitance between the cell input and output, i.e., the Miller capacitance,  $C_M$ , and that charging an effective ground capacitance at the output,  $C_o$ . Also  $C_g = C_M + C_o$ .  $C_M$  and  $C_o$  assumed to be constant and calculated through a series of transient simulations with voltage transitions applied at the input and output nodes, during which the current flowing through the output node is measured. A 2-D lookup table similar to the I-V tables of Blade [5] is used to store values of  $I(V_{in}, V_{out})$  which are found through a series of DC simulations using Hspice. The output voltage waveform can be iteratively computed using Equation (1). The cell characterization of this technique is more accurate than the ones in [5],[7] but it is also more complex. Furthermore, assuming constant values parasitic effects tends to reduce the accuracy of the model.

### 3. ROCC-based Cell Delay Model

This section describes our ROCC-based cell delay modeling for the purpose of timing analysis. The key innovation in this model originates from its construction of the output current signal as a function of the input voltage signal. Therefore, we substitute the DC and transient steps of existing current-based cell delay models with a simpler computational model, while maintaining the accuracy. On the other hand, unlike the voltage-based methods which first need to find an equivalent linear input waveform, the ROCC-based delay calculator directly builds the output voltage waveform.

We utilize the instantaneous rate of current change,  $\theta_c$ , i.e., the derivative of the output current with respect to time. Each cell is pre-characterized with a 2-D lookup table with input voltage and effective output capacitance as the input keys and  $\theta_c$  as its returned value. Output current waveform is computed by using the lookup table data in conjunction with Taylor series expansion of the output current at time instance  $t_k$  around its value at time instance  $t_{k-1}$ . Having the output current waveform, the output voltage waveform can be computed considering the load.

#### 3.1 Impetus for our cell delay model

As described in 2.2, the characterization steps in the existing current-based cell timing analyzers are quite involved. Their major source of complexity is due to the fact that both input and output voltages should be considered as input parameters to the cell model. The DC output current and parasitic effects are dependent to both input and output voltages. These voltages must then be swept during the DC characterization step in order to find the DC output current and fill in the I-V lookup tables. However, parasitic capacitances (i.e.,  $C_M$  and  $C_o$ ) are assumed to be constant to simplify the model. It is not clear how valid this assumption is for different cells which are subjected to noisy waveforms of various shapes. The transient simulations required to find the constant values of the parasitic capacitances are another source of complexity.

To resolve the abovementioned shortcomings, we notice that the output voltage of a cell is a function of the input voltage, the parasitic capacitors, output load, and supply voltage,  $V_{dd}$ . For a given load and power supply voltage level, it is reasonable to assume that the output voltage and parasitic capacitances inside the logic cell are only a function of the applied *input voltage waveform*. Consequently, the output current can be written as a function of the input voltage for a certain load. This observation is important since it enables us to calculate the output current and voltage waveforms, starting from a given input voltage waveform through a constructive stepwise approach.

### 3.2 Cell characterization and output waveform computation

Each logic cell in the standard library is pre-characterized with a lookup table, which is used for output voltage calculations of the cell. This table will be referred as  $CC_R(K \times L)$  where  $K$  and  $L$  denote the number of input voltage levels and effective capacitance values, respectively.  $CC_R$  contains  $\theta_c(V_{in}^i, C_{eff}^j)$  which is simply the rate of change for the cell's output current,  $i_{out}$ , with respect to time for input voltage value,  $V_{in}^i$ , when the cell output is connected to an effective load with a value of  $C_{eff}^j$ :

$$\theta_c(V_{in}^i, C_{eff}^j) = \theta_c(i, j) = \frac{\Delta i_{out}}{\Delta t} \Big|_{v_{in}(t)=V_{in}^i} \quad (2)$$

The  $\theta_c(i, j)$  value is stored in row  $i$  and column  $j$  of the  $CC_R$  lookup table. Note that  $CC_R$  tables are created for each pair of input and output pins of the logic cell by a series of transient Hspice simulations, in which noiseless (saturated ramp) input waveforms are applied while the output current change is monitored. This process is repeated for different effective load capacitances.

$\theta_c$  is a function of the output load; therefore, an effective output capacitance is used to model the output of the load. The iterative effective capacitance calculation technique of [1] is used to determine the effective capacitance. Effective capacitance is dependent on the input transition time; therefore, given a noisy waveform, the effective capacitance changes for different regions of the waveform due to different slews. We thus divide the noisy waveform into different parts by doing a piecewise linear approximation of the waveform. Each part of the noisy waveform is approximated by a fixed transition time, and therefore, has its own effective capacitance. It is empirically found that the effective capacitance calculation converges in fewer than 3 iterations. The effective capacitance calculation is done only for the purpose of obtaining  $\theta_c$  values from the  $CC_R$  lookup table. Note that when calculating the output voltage, we use the actual load. The ROCC-based model is able to consider arbitrary loads including simple capacitive, RC- $\pi$ , or more complex interconnect RC models.

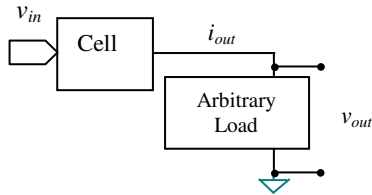


Figure 1.  $i_{out}$  is calculated as a function of  $v_{in}$  and  $\theta_c$ .

The transition time (slew) of the input voltage ramp waveform used for cell characterization can affect  $\theta_c$  values. However, the dependency is weak. Therefore in practice we can do cell characterization for a single input

ramp (with an effective value based on typical waveforms applied to cell.)

The input voltage waveform,  $v_{in}$ , is represented by a time-indexed voltage array, i.e., by using  $P$  equidistant sample points ( $t_0, \dots, t_{P-1}$ ). The cell model takes this data and uses the  $CC_R$  table to find  $\theta_c$  values for each point. Figure 2 depicts the waveform for  $\theta_c$  values of an inverter in our 130nm library under a ramp input (shown in red.)

We assume that the noisy input voltage waveform,  $v_{in}$ , has been characterized by the user (or a timing analysis tool) by having specified the input waveform voltage levels at  $P$  equidistant sample points ( $t_0, \dots, t_{P-1}$ ). The output waveforms are constructed by reporting the output current and voltage levels at the equidistant points. Therefore, it is easy to see that the ROCC-based cell modeling technique can be used as the main delay calculation engine in a timing analysis tool which starts from the primary inputs of the circuits and calculates the voltage waveforms for all intermediate signals and the primary outputs during a linear time traversal of the circuit net list. To detect noise,  $P$  should be selected such that the time between two consecutive sampling points is no larger than one half of the smallest crosstalk noise width. An equivalent output current waveform is then built, in response to the noisy input voltage waveform,  $v_{in}$ , using Taylor series expansion of  $i_{out}$ :

$$i_{out}(t_{k+1}) = i(t_k) + \theta_c(t_{k+1})\Delta t + \frac{1}{2}\theta_c'(t_{k+1})(\Delta t)^2 + \dots + \frac{1}{n!}\theta_c^{(n-1)}(t_{k+1})(\Delta t)^n \quad (3)$$

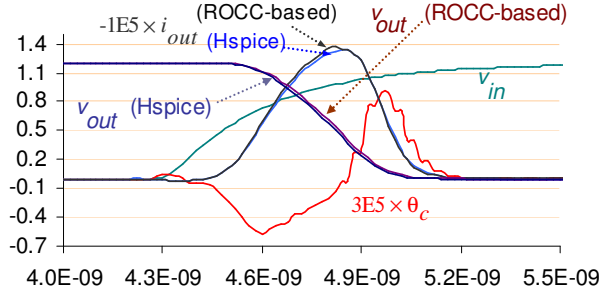
where  $i_{out}(t_0)$  is initialized to zero.  $\theta_c(t_k)$  is a shorthand notation for  $\theta_c(v_{in}(t_k))$ . As pointed out  $\theta_c(v_{in}(t_k))$  is found from the  $CC_R$  table (if necessary using interpolation.)

$\theta_c^{(n-1)}(t_k) = \frac{\Delta \theta_c^{(n-1)}}{\Delta t}(t_k) = \frac{\Delta^n i_{out}}{\Delta t^n}(t_k)$  is the  $n^{\text{th}}$  rate of change of the output current over time which can be calculated directly during the initial library characterization process or can be approximately calculated from the entries in the  $CC_R$  table. In practice  $n=1$  ( $n=2$ ) is sufficient for accurate timing analysis of a logic cell subjected to a noiseless ramp (a noisy input waveform.)  $\Delta t = t_{k+1} - t_k$  is the sampling time. In general, the  $P$  computed output values may not be equidistant. This is undesirable when doing the timing analysis of a logic circuit. To avoid this, a set of  $P$  equidistant points are computed based on weighted average of the two nearest values found from Equation (3).

A Padé approximation can be used to calculate the output current, instead of the Taylor series expansion of Equation (3). Padé approximations are usually superior to Taylor expansions when functions contain poles, because the use of rational functions allows them to be well-represented [9]. However, our experimental results demonstrate that using truncated Taylor series to find the output current provides sufficient accuracy, yet it is much

more efficient than using the Padé approximation. This makes Equation (3) more suitable than an equivalent Padé formula to be used in a logic cell timing analysis tool.

Having calculated the output current, the output voltage can be found for an arbitrary load connected to the output. Figure 2 illustrates the equivalent output current waveform and also the resulting output voltage waveform for a ramp input as well as the actual waveforms generated by Hspice [10].



**Figure 2.** An example of the ROCC-based cell delay model used on a typical ramp input.  $\theta_c$  and  $i_{out}$  have been scaled up to improve visibility.

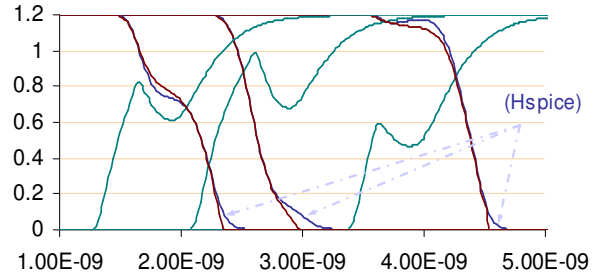
The underlying principle of our approach to handle the compound cells (i.e., multi-stage cells, for example an AND gate) is similar to that described in [5]. We repeat the characterization process for each logic function (NAND function and the NOT function.) Therefore two runs of calculation steps are required for output waveform computation of an AND gate.

Each cell exhibits a kind of low pass filtering effect, which prunes certain amount of input noise. This is not considered in current-based approaches in general. To increase the accuracy, similar to [5], a low pass filter may be used on the noisy input waveforms prior to presenting the waveform to the ROCC-based waveform calculator.

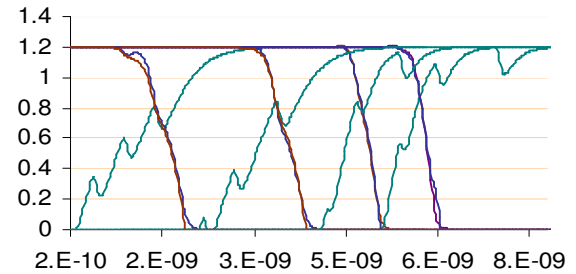
#### 4. Experimental Results

The ROCC-based cell timing analysis was coded in C and compiled under Sun Blade 1000 machine. The cells used in the experiments are from a 130nm, 1.2V production cell library using parasitically extracted netlists. An automated test system was devised to assess the model and compare its delay accuracy and run-time with Hspice. A variety of cells in the production library were tested considering waveforms with a large variety of shapes, from pure ramp to noisy waveforms. The set of experiments included RC- $\pi$  structure as well as capacitive only loads. The size of  $CC_R$  lookup table for each cell was set to (20,5) meaning that 20 input voltage values between 0 and 1.2V and 5 output capacitance values are considered. No low pass filters were used to generate the results in this paper. Compared with Hspice, the output voltage waveforms generated by the ROCC-based delay calculator matched the Hspice with only a 1-3% error.

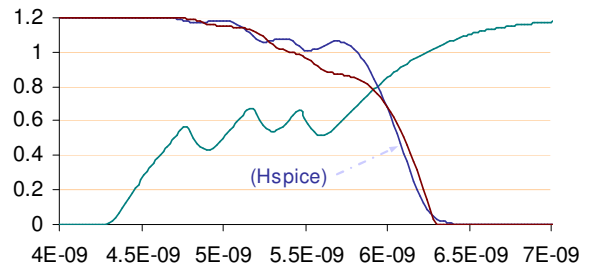
Figure 3 shows comparison with Hspice for some examples of such output waveforms. In this figure, in part (a), the crosstalk-induced noisy input waveforms are generated under single aggressor attack. In parts (b) and (c), the noisy waveform is subjected to three aggressor signal transitions. Therefore there are multiple crosstalk-induced distortions. The equivalent output waveforms generated by our model nearly match the Hspice for waveforms in parts (a) and (b). Part (c) shows an extreme case where the input signal transition to cell is the victim of three strong couplings. To be precise, 200, 200, and 220fF of coupling capacitances exists and the signal transitions on aggressor lines occur close enough to create large crosstalk-induced fluctuations around  $0.5V_{dd}$  level and hence cause multiple  $0.5V_{dd}$  crossing points at the output of the victim. Although the error in  $0.5V_{dd}$  propagation delay value is quite low (less than 1%), it is seen that the equivalent output waveform does not match the Hspice waveform as close as those in parts (a) and (b).



(a)



(b)



(c)

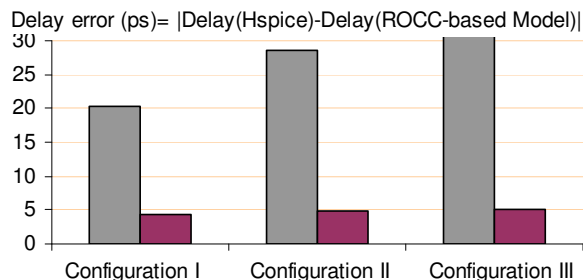
**Figure 3.** The actual and equivalent waveforms by our model for waveforms subjected to (a) one aggressor, as well as (b) and (c) three aggressors

The accuracy of ROCC-based cell model is next demonstrated on some circuit configurations that are part of a large high-performance ASIC design obtained from industry. The circuit configurations appraise our model under different scenarios, i.e., for different number of aggressor lines, interconnect lengths, coupling capacitance values, and input slews to create various noisy waveform shapes. Configuration I is a pair of 1000 $\mu\text{m}$  coupled interconnect lines running parallel to one another with a total distributed coupling value of 200fF. Both aggressor and victim line inputs have a slew of 150ps. For all configurations we set the arrival time and slew (transition time) of the victim line input to 1000ps to 150ps, respectively. For configuration I we swept the arrival time of the aggressor line input from 500 to 1500ps in steps of 5ps. Configuration II includes two aggressor lines each with 200fF total coupling and a victim, all of which are 500 $\mu\text{m}$  long. We maintained a fixed offset of -100ps between signal arrival time of the 1<sup>st</sup> and 2<sup>nd</sup> aggressor line inputs, while sweeping that of the 2<sup>nd</sup> aggressor line input arrival time. The two aggressor inputs have slews 200ps, and 400ps, respectively. Configuration III contains three aggressor lines, with 200fF, 200fF, and 220fF total distributed coupling, respectively, and all 500 $\mu\text{m}$  long. The victim line is also 500 $\mu\text{m}$  long. We maintained a fixed offset of -50 between the arrival times of 1<sup>st</sup> and 3<sup>rd</sup> aggressor line inputs and -100 between those of 2<sup>nd</sup> and 3<sup>rd</sup>. The arrival time of the 3<sup>rd</sup> aggressor line input was then swept from 500 to 1500ps in steps of 5ps. The slews of the three aggressor lines are 200ps, 350ps, and 400ps respectively.

Table 1 shows the maximum and average delay errors of the ROCC-based cell delay model compared to Hspice. The cell delays were calculated as the difference between the 0.5V<sub>dd</sub> crossing point of the output waveform and that of the input waveform. Compared to Hspice and in terms of percentage errors, the average and maximum errors for our model are about 1% and 3%, respectively.

The average run-time of output waveform computation for a typical logic cell is less than 100 $\mu\text{sec}$  for our model.

**Table 1. Absolute errors in calculated delays vs. Spice simulation results for ROCC-based model**



## 5. Conclusions

The ROCC-based logic cell delay analysis model was presented. A pre-characterized table of ROCC, i.e., time derivatives of the cell output current as a function of input voltage and output load values is utilized in combination with the Taylor series expansion to progressively compute the output current waveform. The output voltage is then produced by integrating the output current. Experimental results show the accuracy and efficiency of this new delay model.

## 6. References

- [1] J. Qian, S. Pallela, L. Pillage, "Modeling the effective capacitance for the RC interconnect of CMOS gates," *Transactions On Computer-Aided Design of Integrated Circuits and Systems*, Vol. 13, pp. 1526–1535, 1994.
- [2] K. Agarwal, D. Sylvester, D. Blaauw, "An effective capacitance based driver output model for on-chip RLC interconnects," *Proceedings of Design Automation Conference (DAC)*, pp. 376-381, 2003.
- [3] S. Nazarian, M. Pedram, E. Tuncer, and T. Lin, "Sensitivity-based gate delay propagation in static timing analysis," *Proceedings of International Symposium on Quality of Electronic Designs (ISQED)*, Mar. 2005, pp. 536-541.
- [4] M. Hashimoto, Y. Yamada, H. Onodera, "Equivalent waveform propagation for static timing analysis," *IEEE Trans. Computer-Aided Design of Integ. Circuits & Systems*, Vol. 23, No.4, pp. 498-508, 2004.
- [5] J.F. Croix, D.F. Wong, "Blade and razor: cell and interconnect delay analysis using current-based models," *Proceedings of Design Automation Conference (DAC)*, pp. 386-389, 2003.
- [6] I. Keller, K. Tseng, N. Verghese, "A robust cell-level crosstalk delay change analysis," *Proceedings of International Conference on Computer-Aided Design (ICCAD)*, pp.147-154, 2004.
- [7] A. Korshak, J.C. Lee, "An effective current source cell model for VDSM delay calculation," *Proceedings of International Symposium on Quality of Electronic Designs (ISQED)*, pp. 296–300, 2001.
- [8] P. Buch, W.J. Dai "Understanding ECSM and CCSM," [http://www.magma-da.com/c/@SQJEgjiSi1J\\_6/Pages/MWAUnderstandingECSMan\\_dCCSM.html](http://www.magma-da.com/c/@SQJEgjiSi1J_6/Pages/MWAUnderstandingECSMan_dCCSM.html).
- [9] <http://www.mathworld.com>.
- [10] "Hspice: The golden standard for Accurate Circuit Simulation," <http://www.synopsys.com/products/mixedsignal/hspice/hspice.html>.