CGTA: Current Gain-based Timing Analysis for Logic Cells

Shahin Nazarian   Massoud Pedram   Emre Tuncer   Tao Lin

EE-Systems Dept., University of Southern California
Los Angeles, CA 90089

Magma Design Automation
Santa Clara, CA 95054

{shahin , pedram@usc.edu}

{emre , tao@magma-da.com}

ABSTRACT

This paper introduces a new current-based cell timing analyzer, called CGTA, which has a higher performance than existing logic cell timing analysis tools. CGTA relies on a compact lookup table storing the output current gain (sensitivity) of every logic cell as a function of its input voltage and output load. The current gain values are subsequently used by the timing calculator to produce the output current value as a function of the applied input voltage. This current and the output load then uniquely determine the output voltage value. Therefore, CGTA is capable of efficiently and accurately computing the output voltage waveform of a logic cell, which has been subjected to an arbitrary noisy input voltage waveform. Experimental results are presented to assess the quality of CGTA compared to other existing approaches.

1. INTRODUCTION

As the layout geometries in recent technologies scales down, the increase in the package density and operational frequency aggravates the noise sources. To check whether a noise source can create erroneous outputs, the circuit should be analyzed using a timing analysis tool. Input pattern dependent circuit-level timing analysis with tools such as Spice, is very accurate, but requires significant computational resources, which makes this approach impractical for large VLSI circuits. Logic-level timing analysis tools such as static or statistical static timing analysis tools are used as efficient alternatives with an acceptable level of accuracy.

Delay models for both interconnect lines and cells are required to perform timing analysis. 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 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 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 made the researchers produce excellent interconnect delay models. However the conventional 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 the logic cell timing analysis when noise is present.

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.

The interesting fact about the shape of the waveform is that 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. Hence the shape of the waveform should be considered more effectively. We re-state the problem 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.

As the silicon technology is driven to nanometer, conventional voltage-based lookup tables are nearing the end of their useful life. In [3]-[4] the common voltage-based cell timing analyzers are reviewed and their shortcomings are highlighted. In addition to being inefficient in accurately considering the impact of the shape of the noisy waveform, 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.

Current-based has been shown to be more accurate than voltage-based logic cell timing analysis [5]-[7]. In fact some industrial current-based timing analyzers, such as CCSM and ECSM are already in use [8]. Existing current-based approaches may still exhibit large variations from Spice simulation when presented with complex interconnect models or non-monotonic input voltage waveforms. Their complexity is a barrier to apply them in novel design tools.

In this paper, we present CGTA, a current gain-based timing analysis tool for logic cells. The gain (sensitivity) of output current to input voltage is defined as the derivative of output current waveform to the input voltage waveform. The gain is then used to accurately model the impact of the shape of the input voltage waveform on the output current waveform and eventually the voltage waveform. To respond to the more general problem, CGTA is able to directly build the output waveforms without the need for creating an equivalent input waveform as is done by conventional techniques. CGTA is simple and efficient to implement. More precisely, the application of the current sensitivity factor in delay calculation brings together the accuracy of a current-based cell modeling and the efficiency of a voltage-based cell modeling. It will be shown that CGTA can result in a very efficient, yet accurate timing analysis compared to cell timing analysis using existing current-based methods.

The remainder of this paper is organized as follows. In section 2 we review the previous logic cell timing analysis techniques including current-based ones. Section 3 describes CGTA. Section 4 and 5 review the experimental results and conclusions respectively.

2. BACKGROUND

Most of today's logic cell timing analysis techniques 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 review the various voltage-based cell timing analyzers and discusses 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 timing analysis technique is the Blade in [5]. Blade 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 in [5] 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 output switching is computed while considering the miller effect between input and output nodes as well as internal parasitic effects. These effects are modeled by capacitors which are calculated through a series of transient Spice-based simulations. Additionally, I-V tables are generated in the same fashion as the ones in Blade model.

Alternatively, cells under the crosstalk-induced pulse (glitch) attack are studied in [7] 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. The DC and transient cell characterization steps and large number of required iterations for the aforementioned techniques are too complex to be utilized in existing CAD tools and flows.

3. CGTA

This section describes CGTA, a new current-based cell delay modeling for the purpose of timing analysis. The key innovation in CGTA originates from its modeling of the output current signal as a function of the input voltage signal. Therefore, we substitute the I-V lookup tables of existing current-based cell timing analyzers with a simpler computational model, while maintaining the accuracy. Unlike the voltage-based methods that first need to find an equivalent linear input waveform, our model directly builds the output voltage waveform.

We define the *current gain, ρ_c,* as the derivative of the output current to the input voltage. Each cell is pre-characterized with a 2-D lookup table with input voltage and effective output capacitance as the input keys and ρ_c as its returned value. Output current waveform is computed by using the lookup table gain information and performing Taylor series expansion. Having the

output current waveform the output voltage waveform can be computed considering the load.

3.1 Intuition Behind Current Gain Utilization
As described in 2.2, the characterization steps in the existing current-based cell timing analyzers are quite involved. The 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. These voltages must then be swept during the DC characterization step in order to fill in the I-V lookup tables and compute the parasitic capacitances. Also a series of transient simulations should be performed during which voltage transition are applied to input and output pins. To resolve this issue, we notice that the output voltage of a cell is a function of the input voltage, the parasitic effects, output load, and supply voltage, V_{dd}. Considering the parasitic capacitive values as constant [6], the output voltage may be replaced by the input voltage and output load values. Therefore, having the output load, the output current can be written as a function of input voltage. This is why ρ_c is defined as the sensitivity of the output current to input voltage.

3.2 CGTA Model
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 $I_{gain}(K \times L)$ where K and L denote the number of input voltage levels and effective capacitance values, respectively. I_{gain} contains $\rho_c(V_{in}^i, C_{eff}^j)$ which is simply the derivative of the cell's output current, i_{out} with respect to its input voltage at voltage value V_{in}^i when the cell output is connected to an effective load with a value of C_{eff}^j :

$$\rho_c(V_{in}^i, C_{eff}^j) = \rho_c(i, j) = \frac{\Delta i_{out}}{\Delta v_{in}} \Big|_{v_{in}(t)=V_{in}^i} \qquad (1)$$

ρ_c quantitatively shows how sensitive the output current is to the input voltage, at a certain input voltage value and for a certain effective output capacitance value. The $\rho_c(i,j)$ value is stored in row *i* and column *j* of the I_{gain} lookup table. Figure 1 depicts an example of such a lookup table. The I_{gain} tables are created per pair of input and output pins by a series of transient Spice-based 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.

It is shown later in this section that CGTA is able to consider arbitrary loads including simple capacitive, RC-π, or more complex interconnect RC models.
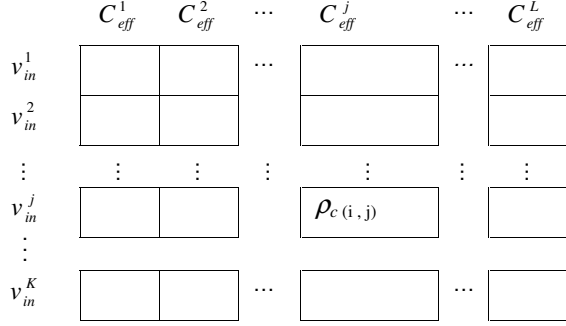
Figure 1. $I_{gain}(K \times L)$, the cell current gain lookup table used in CGTA.

However since ρ_c is a function of the output load, 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. Note that the effective capacitance calculation is done only for the purpose of obtaining ρ_c values from the I_{gain} lookup table. Note that when calculating the output voltage, we use the actual load e.g., an RC-π model shown in Figure 2.

The input voltage waveform, v_{in}, is represented by a time-indexed voltage array, i.e., by using P equidistant sample points $(t_0, \ldots, t_{P-1}.)$ CGTA takes this data and uses the I_{gain} table to find ρ_c values for each point. Figure 3 illustrates ρ_c for a typical crosstalk-induced noisy waveform that was generated from a part of an industrial design.

(a)

(b)
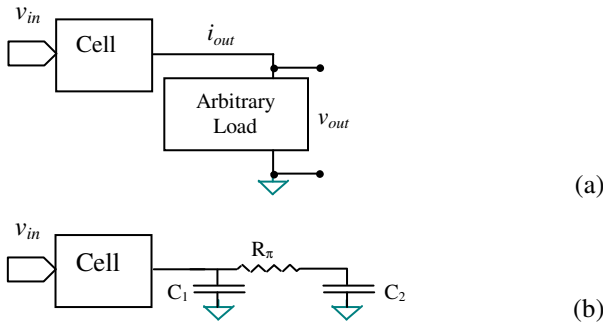
Figure 2. i_{out} is calculated as a function of v_{in} and ρ_c. (a) v_{out} in terms of i_{out} and output load, (b) RC-π modeling of the load.
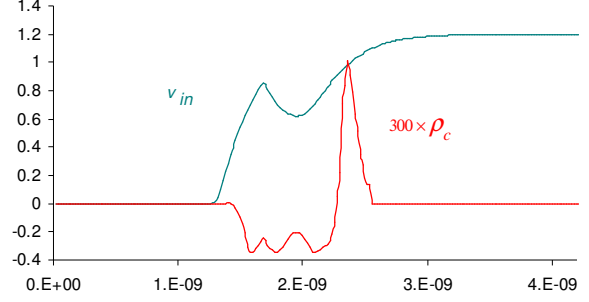
Figure 3. ρ_c for a typical crosstalk-induced noisy waveform. $300 \times \rho_c$ is illustrated for visibility purposes.

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 $(t_0, \ldots, t_{P-1}.)$ CGTA constructs the output waveforms by reporting the output current and voltage levels at P equidistant points. Therefore, it is easy to see that CGTA 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. (The goal of this work is to develop CGTA as a cell timing analysis technique; therefore, calculation of the interconnect delay is outside of the scope of the present paper.) 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. In practice, we have considered a sampling time intervals of at least 50ps, e.g., for an input waveform with a rise time of 1ns, at least 20 sampling points are used.

CGTA builds an equivalent output current waveform in response to the noisy input voltage waveform, v_{in}, using the truncated Taylor series expansion of i_{out}:

$$i_{out}(t_{k+1}) = i_{out}(t_k) + \rho_c(t_k) \cdot \{v_{in}(t_{k+1}) - v_{in}(t_k)\} + \frac{1}{2} \cdot \frac{\Delta \rho_c}{\Delta v_{in}}(t_k) \cdot \{v_{in}(t_{k+1}) - v_{in}(t_k)\}^2 \quad (2)$$

where $i_{out}(t_0)$ is initialized to zero. $\rho_c(t_k)$ is a shorthand notation for $\rho_c(v_{in}(t_k))$. 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 (2).

As pointed out $\rho_c(v_{in}(t_k))$ is found from the I_{gain} table (if necessary using interpolation.) $\frac{\Delta \rho_c}{\Delta v_{in}}(t_k)$ is found from the I_{gain} table and using Equation (3):

$$\frac{\Delta \rho_c}{\Delta v_{in}}(t_k) = \frac{\Delta \rho_c(t_k)}{\Delta v_{in}(t_k)} = \frac{\Delta \rho_c(v_{in}(t_k))}{\Delta v_{in}(t_k)} \qquad (3)$$

$\frac{\Delta \rho_c}{\Delta v_{in}}(t_k)$ is defined to be zero if the input voltage does not change from the previous sampling point to the current one, i.e., $\Delta v_{in}(t_k)=0$.

A Padé approximation can be used to calculate the output current, instead of the Taylor series expansion of Equation (2). 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 the first two terms of the Taylor series to find the output current provides sufficient accuracy, yet it is much more efficient than using the Padé approximation. This makes Equation (2) 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 based on the arbitrary load connected to the output. Figure 4 illustrates the equivalent output current waveform and also the resulting output voltage waveform for the noisy input waveform of Figure 3. Both output current and voltage waveforms by CGTA closely match their respective actual waveforms generated by Hspice [10].

The underlying principle of our approach to handle the compound cells (i.e., multi-stage cells, for example an OR gate) is similar to that described in [5]. We repeat the characterization process for each logic function (NOR function and the NOT function.) Therefore two runs of CGTA calculation steps are required for output waveform computation of an OR gate.

Each cell exhibits a kind of low pass filtering effect, which prunes our certain amount of input noise. This is not considered in current-based approaches in general.
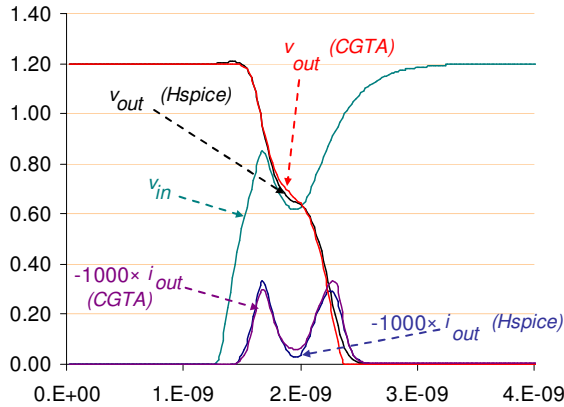
Figure 4. The actual and equivalent waveforms for CGTA. -1000×current waveforms are illustrated for visibility purposes.

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 CGTA calculator.

4. EXPERIMENTAL RESULTS

CGTA was written 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 CGTA 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-π structure as well as capacitive only loads. The size of I_{gain} for CGTA 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 in CGTA to generate the results in this paper. Compared with Hspice, the generated output voltage waveforms by CGTA matched the Hspice with 1-3% error. Figure 4 shows a comparison for an example of such output waveforms for CGTA with Hspice. The key advantage is that efficiency has been obtained without compromising the accuracy compared to other cell timing analysis techniques.

The accuracy of CGTA is next demonstrated on realistic 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 1000μm coupled interconnect lines running parallel to one another with a total distributed coupling value of 100fF. 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 100fF total coupling and a victim, all of which are 500μm long. We maintained a fixed offset of -100ps between signal arrival time of the 1st and 2nd aggressor line inputs, while sweeping that of the 2nd aggressor line input arrival time. The two aggressor inputs have slews 200ps, and 400ps, respectively. Configuration III contains three aggressor lines, each with 50fF total distributed coupling and 300μm long. The victim line is 500μm long. We maintained a fixed offset of -50 between the arrival times of 1st and 3rd aggressor line inputs and -100 between those of 2nd and 3rd. The arrival time of the 3rd 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 existing voltage-based techniques and CGTA compared to Hspice. The cell delays were calculated as the difference between the $0.5V_{dd}$ crossing point of the output waveform and that of the input waveform. In terms of percentage errors, the average and maximum errors for CGTA compared to Hspice are about 1% and 3%, respectively. The average run-time of CGTA output waveform computation for a typical logic cell is less than 100μsec.

Table 1. Absolute errors in calculated delays vs. Spice simulation results for different timing analysis tools

| Method | Delay Error (*ps*) = \|Delay(Hspice) – Delay(Method)\| | | | | | |
| | Configuration I | | Configuration II | | Configuration III | |
	Max	Avg	Max	Avg	Max	Avg
Noiseless Point-based	81.3	29.3	134.2	48.5	153.4	55.3
Noisy Point-based	82.7	24.5	144.5	51.3	151.6	56.4
Least Square Fitting (LSF)	75.1	30.9	110.8	45.4	124.6	49.4
Elmore-based [3]	82.3	14.5	145.3	33.4	166.3	35.3
Weighted LSF [4]	42.4	10.3	49.3	17.4	48.5	15.6
SDP+Elmore-based [3]	39.5	8.6	46.8	12.8	45.6	11.7
CGTA	11.4	3.7	11.8	3.9	11.9	3.9

5. CONCLUSION

Conventional logic cell timing analysis tools approximate a noisy input waveform by an equivalent saturated ramp input, which then enables them to utilize standard delay look-up tables to report the cell timing information as a function of the input ramp slew and the output load. These techniques can result in significant error because the output waveform corresponding to the actual noisy input tends to be quite different from the one that is produced by assuming a saturated ramp input. Current-based cell timing analyzers have been proposed as an accurate alternative to these voltage-based timing analyzers. Unfortunately, the existing current-based techniques tend to be CPU-intensive in terms of the DC and transient simulations needed to pre-characterize the logic cell output current as a function of its input and output voltages and to calculate the values of internal logic cell capacitances. This paper presented CGTA, an accurate and efficient current-based cell timing analyzer which overcomes the aforementioned shortcoming. In particular, CGTA uses a compact table lookup whereby the output current gain (sensitivity) of a logic cell is pre-characterized as a function of its input voltage and

output load. The current gain values are then used as part of a highly efficient timing calculator to provide the output voltage waveform as the logic cell is presented with an arbitrary noisy input voltage waveform. Experimental results demonstrate the high accuracy of CGTA and its efficiency.

REFERENCES

[1] J. Qian, S. Pullela, 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/MWAUnderstandingECSMandCCSM.html.

[9] http://www.mathworld.com.

[10] "Hspice: The golden standard for Accurate Circuit Simulation," http://www.synopsys.com/products/mixedsignal/hspice/hspice.html.