Constructing Current-Based Gate Models Based on Existing Timing Library

Andrew B. Kahng, Bao Liu and Xu Xu

CSE and ECE Departments, UC San Diego, La Jolla, CA 92093-0114
{abk,bliu,xxu}@cs.ucsd.edu

Abstract

Current-based gate modeling achieves a new level of accuracy in nanoscale design timing and signal integrity analysis. However, to generate current-based gate models requires additional pre-characterization of the gate, e.g., in the form of a new or an extended timing library format. We construct current-based gate models based on the existing Liberty timing library format without further pre-characterization. We present an inverse problem formulation, and propose to solve the problem by quadratic polynomial regression. Our constructed current-based gate models find applications in timing, power, and signal integrity verifications for improved accuracy in library-compatible flows, e.g., to include power supply voltage drop effect in gate delay calculation without further pre-characterization, to calculate gate supply current, etc. Our experimental results show our constructed current-based gate models achieve slightly less accurate results, e.g., within 4.6%*(*8.6%*), than pre-characterized current-based gate models, e.g., within* 4.3%*(*4.4%*), of SPICE results in gate delay calculation for ideal (degraded) power supply voltage, and accurate gate supply current calculation.*

1 Introduction

Performance is a primary VLSI design objective, and timing verification, as is crucial to VLSI design success, has evolved through several generations as VLSI manufacturing process technology evolves into nanoscale domain.

Traditional VLSI design timing verification is based on k-factor lookup tables, e.g., in Liberty timing library format, which give gate delays and output signal transition times based on gate input signal transition times and gate capacitive loads.

In deep sub-micron VLSI designs, on-chip interconnects need to be taken as distributed R(L)C networks. The traditional single-load-capacitor-based k-factor lookup tables adapt to this era thanks to the technique of "effective capacitance" computation [4], which translates a distributed R(L)C load interconnect into an equivalent single lumped load capacitance for gate delay and gate output signal transition time computation.

An effective-capacitance-based gate model includes a voltage source and an output resistor which are computed from the traditional k-factor lookup tables. The voltage source outputs a saturated ramp voltage, while the output resistor captures tail attenuation of the driving point waveform. Certain assumptions in this model hold only for a limited set of VLSI technology nodes, e.g., approximating a signal transition in a saturated ramp, and capturing a gate output resistance as a constant, increasingly deviate from reality as VLSI process technology evolves. Thus effective-capacitance-based gate models can not accurately capture signal integrity effects in nanoscale VLSI designs.

In better accordance with transistor physics, a current-based gate model [3, 6] includes a voltage-controlled current source as well as a voltage-controlled capacitor. A transient analysis at the driving point computes a time domain gate output voltage waveform. Such transient analysis at gate level provides a new level of efficient-accuracy tradeoff between SPICE simulation and gate level timing and signal integrity analysis. As effective-capacitance-based gate level analysis cannot accurately capture nanoscale design phenomena, for example, signal integrity effects which result in increasingly complex signal transition waveforms, transient analysis becomes a necessity, yet efficiency advantage over SPICE simulation still remains with current-based gate models.

Industry is moving in adopting current-based gate delay models, e.g., Cadence Design System's Effective Current Source Model (ECSM) [2], and Synopsys' Composite Current Source (CCS) model. These models need additional pre-characterization to be constructed, for example, Cadence proposed an extension of the Liberty timing library format, which describes gate output voltage waveform and gate intrinsic output capacitance for each input signal slew rate and output net capacitance combination, to help construct a current-based gate model [2]. A complete pre-characterization of a gate output current needs DC simulation, e.g., by sweeping the gate input and output voltages in SPICE simulation [3].

We propose to construct current-based gate models based on existing Liberty timing libraries without any additional pre-characterization. The constructed current-based gate models find applications in existing library-compatible flows for accuracy improvement in timing, power, and signal integrity verifications, e.g., to include supply voltage variation effect in gate delay calculation. This leads to better data efficiency and accuracy than other models, e.g., Scalable Polynomial Delay Model (SPDM), which includes additional coefficients to characterize timing for discrete supply voltages.

We observe that constructing a current-based gate model

Proceedings of the 7th International Symposium on Quality Electronic Design (ISQED'06)
0-7695-2523-7/06 $20.00 © 2006 IEEE

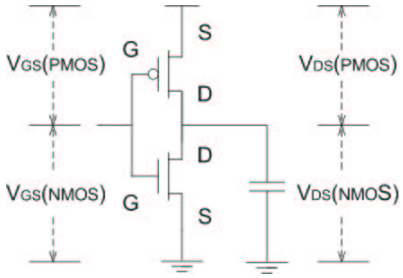

Figure 1. A CMOS inverter and a single load capacitor. Gate output current I_o, input voltage V_i, and output voltage V_o are given by PMOS/NMOS transistor current, port voltages V_{GS} and V_{DS}, respectively.

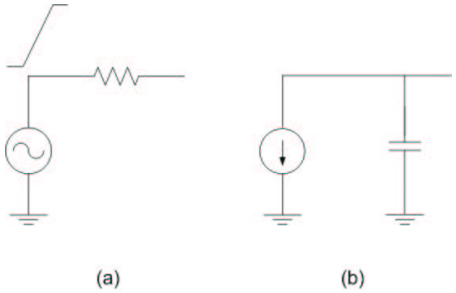

Figure 2. (a) Voltage-based gate model which consists of a voltage source and a resistor; and (b) current-based gate model which consists of a current source and a capacitor.

from an existing timing library is an inverse problem, e.g., relating gate output current time domain integral and load capacitance gives an inhomogeneous Fredholm integral equation system of the first kind [9]. Inverse problem solutions are usually extremely sensitive to variation of input data, and their accuracy and smoothness are difficult to obtain. However, we approximate a gate output current by a quadratic polynomial of gate input and output voltages, and propose to apply a standard least mean square nonlinear regression method for the best approximation of gate output current.

Our experimental results show that our constructed current-based gate models compute gate delays within 4.6%(8.6%) of SPICE results, while pre-characterized current-based gate models compute gate delay with 4.3%(4.4%) of SPICE results, for ideal (degraded) power supply voltage. Our constructed current-based gate models also give accurate gate supply current calculation. In general, our constructed current-based gate models find applications in existing library-compatible flows for accuracy improvement in timing, power, and signal integrity verifications, with slightly less accurate results than pre-characterized current-based gate models.

The rest of the paper is organized as follows. We introduce current-based gate modeling in Section 2, and present an inverse problem formulation in Section 3. We propose to find polynomial regression of a gate output current in Section 4, and apply our constructed model for gate delay with supply voltage variation and supply current calculation in Section 5. We present our experimental results in Section 6 and conclude in Section 7.

2 Current-Based Gate Model

A MOSFET model is typically current-based, for example, the alpha-power law MOSFET model is as follows [11].

$$I_{DS} = \begin{cases} 0 & V_{GS} < V_T \\ \frac{W}{L_{eff}} \frac{P_C}{P_V} (V_{GS} - V_T)^{\alpha/2} & V_{DS} < P_V (V_{GS} - V_T)^{\alpha} \\ \frac{W}{L_{eff}} P_C (V_{GS} - V_T)^{\alpha} & V_{DS} > P_V (V_{GS} - V_T)^{\alpha} \end{cases} \quad (1)$$

where I_{DS} is the source-drain current, V_{DS} the source-drain voltage, V_{GS} the gate-source voltage, V_T the threshold voltage, W the channel width, L the channel length, P_C and P_V are parameters, α is typically between 1 and 2 to capture nanometer transistor effects.

For a simple inverter (Fig. 1), such a transistor model gives a current-based gate model. For a complex gate, an equivalent inverter marcomodel can be constructed for each input combination [7], and gives a similar current-based gate model, e.g., for the worst case input combination for static timing analysis. Such current-based gate models better capture transistor physics and provide significant accuracy improvement compared with voltage-based gate models.

A current-based gate model includes a 2-D lookup table $I_o(V_i, V_o)$ which gives gate output current for a pair of gate input and output voltages, and a voltage-controlled capacitor at the gate output (Fig. 2). A transient analysis is applied to compute the gate output voltage, e.g., at each time step, the gate output current is given by the 2-D lookup table, and the gate output voltage variation is computed, e.g., by a nonlinear solver which applies Newton-Raphson or secant iteration [3]. Algorithm 1 describes the transient analysis process for a current-based gate model.

Algorithm 1: Transient Analysis with a Current-Based Gate Model

Input: Piece-wise linear input V_i, lookup table $I_o(V_i, V_o)$
Output: Piece-wise linear output V_o

1. Reduce load network, e.g., to a Pi-model
2. For each time step t
3. Find $V_i(t)$ and $V_o(t)$
4. Find $I_o(V_i, V_o)$ by table lookup
5. Compute $V_o(t+1)$

Additional pre-characterization is needed by existing approaches to construct a current-based gate model. In Cadence ECSM, current-based gate models are computed based on gate output voltage waveforms for each input signal transition time and load capacitance configuration. In Blade and Razor [3], the current table is constructed by DC simulation, e.g., by sweeping the gate input and output voltages and performing DC analysis in SPICE simulation. In the next section, we present to construct a current-based gate model based on existing Liberty timing library without additional pre-characterization.

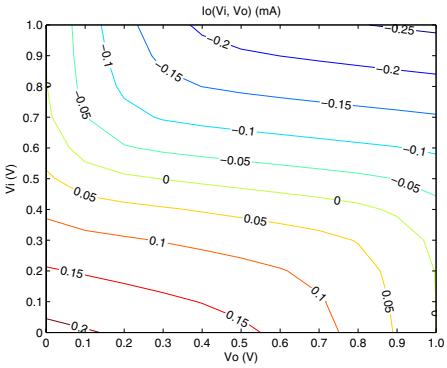

Figure 3. Contour of the output current $I_o(V_i, V_o)$ of a $4\times$ inverter in $90nm$ technology.

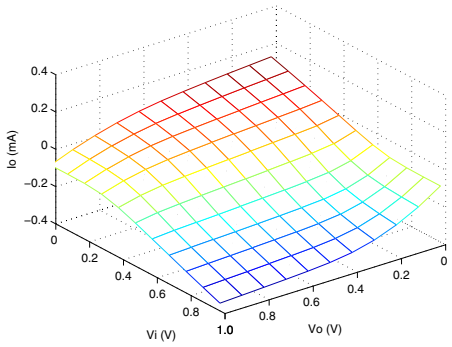

Figure 4. 3-D surface map of the output current $I_o(V_i, V_o)$ of a $4\times$ inverter in $90nm$ technology.

3 Inverse Problem

We study the following problem.

Problem 1 *Given*

1. $\beta_1 =$ *signal slew lower threshold (e.g.,* 0.2*),*
2. $\beta_2 =$ *signal delay threshold (e.g.,* 0.5*),*
3. $\beta_3 =$ *signal slew upper threshold (e.g.,* 0.8*),*
4. $T_{r-in} =$ *input slew rate (from* 0% *to* 100%V_{dd}*),*
5. $C_L =$ *output load capacitance,*
6. $D_g =$ *gate delay, and*
7. $T_{r-out} =$ *output signal slew rate (from* $\beta_1 V_{dd}$ *to* $\beta_3 V_{dd}$*),*

Find a current-based gate model of a 2-D lookup table $I_o(V_i, V_o)$ *and a single intrinsic gate output capacitance* C_g.[1]

This is an inverse problem, i.e., to find an unknown/underlying physical process by a set of measurements [9]. For this problem, the time domain integral of the gate output current, or the gate output charge, is given by the load capacitance times output voltage swing.

$$Q = \int Idt = C\Delta V \tag{2}$$

This gives us inhomogeneous Fredholm integral equations of the first kind (i.e., the unknown function appears only in the integral) [9] for each gate delay or output slew rate in the timing library lookup tables as follows.

$$\begin{aligned} \int_0^{D_g+0.5T_{r-in}} I_o(V_i,V_o)dt &= \beta_2(C_g+C_L)V_{dd} \\ \int_{D_g+0.5T_{r-in}-0.5T_{r-out}}^{D_g+0.5T_{r-in}+0.5T_{r-out}} I_o(V_i,V_o)dt &= (\beta_3-\beta_1)(C_g+C_L)V_{dd} \end{aligned} \tag{3}$$

Straightforwardly, the above integral equations can be translated into differential equations, which are nothing different than normal linear equations. Numerical integration methods, e.g., forward Euler, give accurate results for sufficiently small time steps.[2] At each time step, linear interpolation represents a gate current by a finite number of gate currents in a lookup table. Solving the linear equation system gives the lookup table of gate current $I_o(V_i, V_o)$. However, such solutions are very sensitive to small variations of input data. Small variations do exist as we approximate the gate input and output signal transition waveforms by saturated ramps. This results in extremely poor smoothness of the solutions.

In general, an inverse problem is solved by optimization of an objective $A + \alpha B$, where A is an accuracy measurement, B is a smoothness measurement, and α provides a tradeoff between accuracy and smoothness of the solution [9].

4 Regression of Gate Output Current

Achieving quality solutions of an inverse problem often requires *a priori* information. For this, let us look at the contour and the 3-D surface map of the output current of a $4\times$ inverter in $90nm$ technology (Fig. 3 and Fig. 4).[3]

We approximate a gate output current in a quadratic polynomial of the gate input and output voltages, e.g., for a rising signal transition, as follows.

$$\begin{aligned} I_o(V_i,V_o) &= a_{00}+a_{01}V_o+a_{02}V_o^2 \\ &+ a_{10}V_i+a_{11}V_iV_o+a_{12}V_iV_o^2 \\ &+ a_{20}V_i^2+a_{21}V_i^2V_o+a_{22}V_i^2V_o^2 \end{aligned} \tag{4}$$

The following observations hold for the output current of a gate.

$$\begin{aligned} I_o(0,0) &> 0 \\ I_o(0,V_{dd}) &= 0 \\ I_o(V_{dd},0) &= 0 \end{aligned}$$

[1] More accurate gate models include multiple and/or non-constant intrinsic capacitors [2].

[2] Other numerical integration methods, e.g., trapezoidal, involves solving a nonlinear equation, e.g., by Newton-Raphson or secant method, which do not have a closed form solution, and achieve only negligible accuracy improvement for sufficiently small time steps.

[3] We observe similar contour and 3-D surface map for the output current of a complex gate, which can be macromodeled as an inverter for each input combination [7], except that a complex gate of a non-inverting logic function needs to substitute V_i by $V_{dd} - V_i$.

$$
\begin{aligned}
I_o(V_{dd},V_{dd}) &< 0 \\
\frac{\partial I_o(V_i,V_o)}{\partial V_i} &< 0 \ \forall V_i,V_o \\
\frac{\partial I_o(V_i,V_o)}{\partial V_0} &< 0 \ \forall V_i,V_o
\end{aligned} \tag{5}
$$

We have the following constraints for the polynomial coefficients in (4).

$$
\begin{aligned}
a_{00},a_{20} &> 0 \\
a_{01},a_{02},a_{10} &< 0 \\
a_{00}+a_{01}+a_{02} &= 0 \\
a_{00}+a_{10}+a_{20} &= 0 \\
a_{01}+a_{11}+a_{21} &< 0 \\
a_{10}+a_{11}+a_{12} &< 0 \\
a_{00}+a_{01}+a_{02} &+ \\
a_{10}+a_{11}+a_{12} &+ \\
a_{20}+a_{21}+a_{22} &< 0
\end{aligned} \tag{6}
$$

Now, we have reduced the inverse problem to finding a small number of polynomial coefficients, and these coefficients are in limited ranges, e.g., a_{00} is the gate output current $I_o(0,0)$ with zero gate input and output voltages, which upper bounds the absolute values of a_{01}, a_{02}, a_{10}, and a_{20}. These enable us to apply a simple "try and trial" method to find the coefficients which give the most accurate estimates of gate delay and output signal transition time.

For better efficiency, we apply a standard Least Mean Square (LMS) regression method [5], to explore the solution space of coefficient combinations. We perturb the coefficients in a rotation, and approximate gradients by computing differentiates of the coefficient combinations, e.g., as follows. For each tentative coefficient combination, we realize a $I_o(V_i,V_o)$ table, and compute gate delay for each input signal transition time and load capacitance configuration as in the timing library lookup table. We compare the computed gate delays with the entries in the timing library lookup tables, and compute the sum of square of the gate delay mismatches. If a perturbation of a coefficient reduces the sum of square of gate delay mismatch, we commit the perturbation; otherwise, we perturb the coefficient in the opposite direction. The perturbation process is in an iteration, where the step of perturbation is gradually reduced, until the coefficients are converged.

Algorithm 2 describes this nonlinear regression process.

5 Applications

A current-based gate model finds its application in a wide range of nanoscale VLSI analysis problems, where signal transition waveforms are becoming increasingly complex, traditional ramp approximation deviates from reality, and transient analysis becomes a necessity. For example, complex signal transition waveforms appear with (1) a weak driver of a long interconnect, (2) crosstalk coupling, (3) noise propagation, etc. We present two direct applications of a current-based gate model as follows.

Algorithm 2: Construct a Current-Based Gate Model from a Liberty Timing Library

Input: $\beta_1, \beta_2, \beta_3$, $\{T_{r-in}\}$, $\{C_L\}$, $\{D_g\}$, and $\{T_{r-out}\}$
Output: $I_o(V_i,V_o)$, C_{go}

1. Start with initial polynomial coefficients
2. For each iteration
3. Perturb a coefficient $a_i' = a_i + \delta$
4. Compute sum of square of gate delay mismatch ε
5. If ε reduces, commit perturbation $a_i = a_i + \delta$
6. Otherwise, go opposite direction $a_i = a_i - \delta$
7. Stop if no improvement is available
8. Reduce step δ, go to another iteration
9. Compute $I_o(V_i,V_o)$ and C_{go} from the coefficients

5.1 Gate Delay Calculation with Supply Voltage Variation

With a current-based gate model, a direct application is to include power/ground supply voltage variation effect in gate delay calculation in a library-compatible flow.

For example, consider an inverter (or an equivalent inverter macromodel for a complex gate [7]) (Fig. 1), for a falling input signal transition which turns on the PMOS transistor, a supply voltage degradation ΔV reduces both V_{DS} and V_{GS} of the PMOS transistor by ΔV (while the gate input and output voltages V_i and V_o swing between the ground voltage and the degraded power supply voltage $V_{dd}-\Delta V$). Therefore, the gate input and output voltages need to be increased by ΔV, respectively, in gate output current table lookup.

$$
\begin{aligned}
I_o &= I_o(V_i',V_o') \\
V_i' &= Vi+\Delta V \\
V_o' &= Vo+\Delta V
\end{aligned} \tag{7}
$$

For a rising input signal transition which turns on the NMOS transistor, the gate input and output voltages V_i and V_o do not need to be adjusted in gate output current table lookup, because the ΔV supply voltage degradation does not apply to the NMOS transistor when the PMOS transistor is off (while V_i and V_o swing between 0 and $V_{dd}-\Delta V$).

In case that there is a ground voltage bounce ΔV, V_i and V_o need to be decreased by ΔV in gate current table lookup for a rising input signal transition; and do not need to be adjusted in gate current table lookup for a falling input signal transition.

5.2 Supply Current Calculation

Another application of a current-based gate model is to calculate supply current of the gate. Supply current can be approximated based on a Liberty library, e.g., by approximating a gate supply current in a triangle waveform, and finding the width of the triangle as the signal transition time, and the area of the triangle as the gate power consumption given in a Liberty library. Closed form formulas [8, 10, 12] have been proposed to calculate gate supply current based on a simple device model, e.g., the alpha-power law transistor model [11]. Applying a current-based gate model achieves better accuracy

of supply current estimation, since it avoids the simplifications and the assumptions made in closed-form formulas and does not rely on a closed-form device model [1]. With our constructed current-based gate models, we can directly compute supply current of a gate for improved accuracy.

6 Experiments

We apply our proposed current-based gate model construction and gate delay calculation to a list of library cells with different input slew rate and load configurations in 70*nm* technology from Berkeley Predictive Technology Model (*BPTM*). For each library cell, we generate Liberty timing library lookup tables by running SPICE simulation. We then construct a current-based model based on the Liberty timing library lookup tables by Algorithm 2, and compute the gate output waveforms by applying transient analysis.

6.1 Regression of Gate Output Current

We compare three gate output current regression processes: (1) gate output current regression in our constructed gate models for least mean square gate delay mismatch, (2) quadratic and (3) cubic polynomial regressions by a commercial regression solver *R* of the pre-characterized gate output currents given by SPICE simulation (Table 1). The means and the standard deviations of gate delay mismatch are computed for 90*nm* technology cell instances with 1.0*V* supply voltage, $20fF$, $50fF$, $100fF$, or $1000fF$ load capacitance, and $10ps$, $100ps$, $200ps$, or $500ps$ input signal transition time. We observe that (1) quadratic and cubic polynomial regressions give comparable results in matching a pre-characterized current-based gate model, which supports our choice of order of polynomial regression, and (2) we achieve better gate delay regression by matching gate delays directly than the commercial regression solver which matches gate output current.

6.2 Gate Delay with Supply Voltage Variation

We compare gate delay calculation by (1) our constructed current-based gate models, (2) pre-characterized current-based gate models (without gate output current regression), and (3) SPICE simulation for cell instances in 90*nm* technology (Table 2). In most of the cases, our constructed gate models are slightly less accurate (and even more accurate in some cases) than the pre-characterized gate models. The pre-characterized current-based gate models compute gate delay within 4.3% of SPICE results, our constructed gate models compute gate delay within 4.6% of SPICE results. In the presence of power supply voltage variation, we adjust gate output current by applying (7). The constructed and the pre-characterized models compute gate delay variation within 8.6% and 4.4% of SPICE simulation results, respectively.[4]

6.3 Supply Current

We apply our constructed current-based gate models to supply current calculation, and compare with results from pre-characterized current-based gate models and SPICE simulation. Fig. 5 shows the supply current of a $4\times$ inverter in 90*nm* technology with $10ps$ input signal transition time and $20pF$ load capacitance. Our constructed current-based gate models give accurate supply current estimation which closely resembles estimates from pre-characterized current-based gate models and SPICE simulation.

[4] We expect accuracy improvement with implementation of voltage-controlled intrinsic gate output capacitances.

Table 1. The means (%) and the standard deviations (ps) of gate delay mismatch based on (1) gate output current regression in our constructed current-based gate models for least mean square gate delay mismatch, (2) quadratic, and (3) cubic polynomial regressions of the pre-characterized gate output currents for 90*nm* technology cell instances.

Cell name	Our Constructed		Quadratic Pre-Char.		Cubic Pre-Char.	
	μ (%)	σ (ps)	μ (%)	σ (ps)	μ (%)	σ (ps)
Inv-x4	1.25	5.7	9.73	45.2	11.12	37.8
Inv-x8	1.85	16.9	12.93	27.2	14.67	27.8
Inv-x16	3.36	19.3	17.63	20.6	10.63	25.0
Nand2-x8	1.34	13.2	8.22	24.0	9.59	22.4
Nor2-x4	0.46	4.8	7.06	152.7	5.91	56.5

6.4 Runtime

The runtime for constructing each current-based gate model (which is independent on cell complexity) takes an average of 28.3 seconds in an *i*686 Linux system with a $2.8GHz$ P4 processor and 512*MB* memory; applying a current-based gate model takes an average of 0.13 second for 1000 transient analysis time steps (including program initialization time).

Our current-based gate model construction can be smoothly integrated into an existing library-compatible timing verification flow for improved analysis accuracy, e.g., to include power/ground supply voltage variation effect in gate delay calculation, to calculate gate supply current, etc. In such a flow, the model construction process needs to take place only once for each library cell, and the subsequent analysis runs will be able to apply the constructed current-based gate models for accuracy improvement.

7 Conclusion

Existing current-based gate modeling requires additional pre-characterization in the form of a new or an extended timing library format. We propose to construct current-based gate models from the existing Liberty timing library format without need of additional pre-characterization, for any complex gate based on an inverter macromodel. The constructed current-based gate models enables accuracy improvement in existing library-compatible flows for timing, power, and signal integrity verifications.

We present an inverse problem formulation, and propose to approximate gate output current as a function of gate input and output voltages by quadratic polynomial regression for least mean square of gate delay mismatch. Our experimental results show that our constructed current-based gate models achieve slightly less accurate results, e.g., within

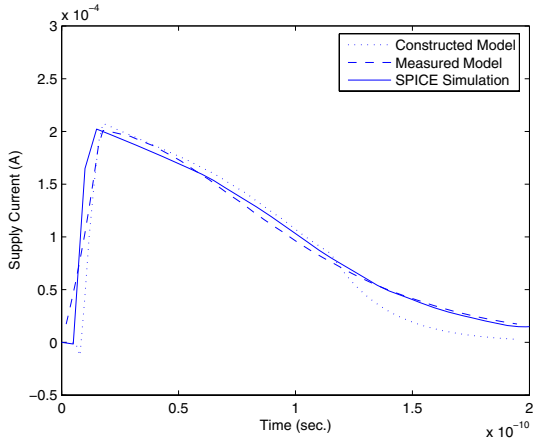

Figure 5. Supply current of an $4\times$ inverter in $90nm$ technology with $10ps$ input signal transition time and $20pF$ load capacitance by (1) our constructed current-based gate model, (2) pre-characterized current-based gate model, and (3) SPICE simulation.

4.6%(8.6%), than pre-characterized current-based gate models, e.g., within 4.3%(4.4%), of SPICE results in gate delay calculation for ideal (degraded) power supply voltage, and accurate gate supply current calculation.

Accuracy improvement of current-based gate model construction includes implementation of more powerful regression solvers, more sophisticated gate models, e.g., with multiple non-constant intrinsic capacitors and resistors.

Acknowledgment

The authors thank Professor Peng Li at TAMU for many helpful discussions on device modeling.

References

[1] E. Acar, R. Arunachalam and S. Nassif, "Predicting Short Circuit Power From Timing Models," *Proc. of the Asia Pacific Design Automation Conference*, 2003.

[2] Cadence Design Systems, Inc., "Effective Current Source Model," version 1.1, 2004.

[3] J. F. Croix and D. F. Wang, "Blade and Razor: Cell and Interconnect Delay Analysis Using Current-Based Models," in *Proc. Design Automation Conference*, 2004.

[4] F. Dartu, N. Menezes, J. Qian and L. Pileggi, "A Gate-Delay Model for High-Speed CMOS Circuits", in *Proc. Design Automation Conference*, 1994, pp. 576-579.

[5] R. Gunst and R. Mason, *Regression Analysis and Its Application: A Data-Oriented Approach,* New York: Macel Dekker Inc., 1980.

[6] A. Korshak and J.-C. Lee, "An Effective Current Source Cell Model for VDSM Delay Calculation," in *Proc. IEEE International Symposium on Quality Electronic Design*, 2001, pp. 296-300.

[7] A. Nabavi-Lishi and N. C. Rumin, "Inverter Models of CMOS Gates for Supply Current and Delay Evaluation," in *IEEE Trans. on Computer-Aided Design*, vol. 13, 1994, pp. 1271-1279.

[8] K. Nose and T. Sakurai, "Analysis and Future Trend of Short-Circuit Power," in *IEEE Trans. Computer-Aided Design,* vol. 19, 2000, pp. 1023-1030.

[9] W. H. Press, S. A. Teukilsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in C - The Art of Scientific Computing*, Second Edition, Cambridge University Press, 2002.

[10] J. L. Rossello and J. Segura, "Charge-Based Analytical Model for the Evaluation of Power Consumption in Submicron CMOS Buffers," in *IEEE Trans. on Computer-Aided Design*, 21(4), 2002, pp. 433-448.

[11] T. Sakurai and A. R. Newton, "Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas," *IEEE Journal of Solid-State Circuits*, 25(2), April 1990.

[12] S. Vemuri and N. Scheinberg, "Short-Circuit Power Dissipation Estimation for CMOS Logic Gates," in *IEEE Trans. on Circuit and Systems-I*, vol. 4, 1994, pp. 762-766.

Table 2. Gate delay calculation with ideal and degraded power supply voltages by (1) our constructed current-based gate model, (2) pre-characterized current-based gate model, and (3) SPICE simulation for $90nm$ technology cell instances.

Cell name	V_{dd} (V)	C_L (fF)	$D_g(1)$ (ps)	$D_g(1)$ (%)	$D_g(2)$ (ps)	$D_g(2)$ (%)	$D_g(3)$ (ps)
Inv-x4	1.0	20.0	54.7	93.0	58.8	100.0	58.8
Inv-x4	1.0	50.0	135.4	97.4	137.2	98.7	139.0
Inv-x4	1.0	100.0	270.1	99.0	271.0	99.3	272.9
Inv-x4	1.0	1000.0	2693.0	101.0	2670.3	100.0	2666.9
Inv-x4	0.9	20.0	74.3	102.0	74.3	102.1	72.8
Inv-x4	0.9	50.0	181.5	105.4	171.8	99.8	172.2
Inv-x4	0.9	100.0	360.1	106.7	333.1	98.7	337.6
Inv-x4	0.9	1000.0	3578.0	108.6	3252.7	98.8	3293.0
Inv-x8	1.0	20.0	31.3	97.8	32.8	102.5	32.0
Inv-x8	1.0	50.0	68.3	94.9	72.9	101.2	72.2
Inv-x8	1.0	100.0	134.0	95.7	140.0	100.6	139.1
Inv-x8	1.0	1000.0	1316.4	98.0	1343.0	100.0	1342.8
Inv-x8	0.9	20.0	40.2	101.2	40.0	100.8	39.7
Inv-x8	0.9	50.0	95.5	106.8	89.0	99.6	89.4
Inv-x8	0.9	100.0	187.0	108.6	170.0	98.7	172.2
Inv-x8	0.9	1000.0	1775.9	106.9	1625.7	97.9	1661.0
Inv-x16	1.0	20.0	18.2	98.4	19.3	104.3	18.5
Inv-x16	1.0	50.0	39.7	102.6	39.4	101.8	38.7
Inv-x16	1.0	100.0	71.5	99.0	72.7	100.0	72.2
Inv-x16	1.0	1000.0	643.6	95.5	674.7	100.0	674.1
Inv-x16	0.9	20.0	24.8	107.8	23.0	100.0	23.0
Inv-x16	0.9	50.0	51.1	106.4	47.8	99.6	48.0
Inv-x16	0.9	100.0	95.3	106.6	88.6	99.1	89.4
Inv-x16	0.9	1000.0	885.5	106.2	816.7	97.8	833.9
Nand2-x8	1.0	20.0	34.5	103.9	32.4	97.6	33.2
Nand2-x8	1.0	50.0	74.0	100.8	72.3	98.5	73.4
Nand2-x8	1.0	100.0	140.6	100.2	139.2	99.2	140.3
Nand2-x8	1.0	1000.0	1332.4	99.5	1341.2	100.1	1339.4
Nand2-x8	0.9	20.0	46.5	107.8	39.2	95.6	41.0
Nand2-x8	0.9	50.0	99.9	106.4	88.9	97.9	90.8
Nand2-x8	0.9	100.0	188.9	106.6	171.2	98.7	173.5
Nand2-x8	0.9	1000.0	1791.2	108.3	1658.8	100.2	1654.5
Nor2-x4	1.0	20.0	120.7	99.7	125.5	103.6	121.1
Nor2-x4	1.0	50.0	284.3	100.1	287.1	101.1	283.9
Nor2-x4	1.0	100.0	558.1	100.5	558.1	100.5	555.1
Nor2-x4	1.0	1000.0	5469.4	101.1	5416.6	100.2	5408.3
Nor2-x4	0.9	20.0	158.0	104.1	156.8	103.3	151.7
Nor2-x4	0.9	50.0	370.8	104.0	360.7	101.2	356.4
Nor2-x4	0.9	100.0	725.5	104.0	678.7	97.3	697.5
Nor2-x4	0.9	1000.0	7104.9	104.6	6542.2	96.4	6788.5