WAVSTAN: Waveform based Variational Static Timing Analysis

Saurabh K Tiwary
Cadence Berkeley Labs
Berkeley, CA, USA
stiwary@cadence.com

Joel R Phillips
Cadence Berkeley Labs
Berkeley, CA, USA
jrp@cadence.com

***Abstract*— We present a waveform based variational static timing analysis methodology. It is a timing paradigm that lies midway between convention static delay approximations and full dynamic (SPICE-level) analysis. The core idea is to break the modulation of waveforms processed by a circuit into two parts: (a) non-linear circuit elements *e.g.*, transistors, diodes etc. and (b) linear elements: transmission line, RLC network etc. The non-linear and linear parts of the circuit are then solved using a combination of current-source modeling, model order reduction methodology, perturbation analysis and learning-based Galerkin methods which helps us get SPICE-like accuracies. The proposed method is potentially as robust and 10-20X faster than current-source based gate modeling methodologies.**

I. Introduction

Static timing analysis (STA) is the current method of choice for analyzing and verifying the timing constraints in digital ASICs. Static timing methodologies are at root based on a simple abstraction of the waveforms at different nodes in any given circuit – modeling the waveforms with two numbers, a slew and a delay – which are propagated in a conservative way through the circuit graph. This methodology is "vectorless" and avoids the expense of dynamic simulation [12] [8] [11] to estimate timing. Due to the phenomenal advancement in drawing extremely small transistor dimensions and the push for aggressive design styles to achieve better chip functionality, the voltage waveforms commonly observed in modern chips differ significantly from the assumptions of STA. We often observe non-idealities in the waveforms that are not captured by a conventional static timing analysis tool, for example, overshoots, spikes, ringing etc.

Tables for traditional table-lookup gate level models have to be generated for different loading conditions to accurately predict the circuit behavior in real designs. It is often quite difficult to exhaustively characterize the cells for all possible loading conditions. Thus, approximate equivalent capacitance models are generated for the interconnects [10] to do a look-up in the table of the gate models. Due to these approximations and the intrinsic limitations of the models, the generated output waveforms do not match the real circuit behavior. The problem is more severe in the case of non-linear loading conditions as is the case when other gates are on the fan-out of the driver. The problem is exacerbated due to large changes in transistor and interconnect characteristics in the presence of process variations.

Recently, the research efforts have focused on two key aspects of the problem:

- modeling the gate behavior in the presence of non-ideal waveforms, bias and loading conditions and
- efficiently simulating linear interconnect networks.

The most successful amongst the gate models have been the current-source models [7] [4]. They represent the output current and impedance of a logic gate as a function of input voltage in tabular form. As for the linear interconnect network, there are fairly mature techniques for applying model order reduction ideas to solve the resulting large linear circuit equations efficiently [9]. Together, this strategy results in a substantial improvement over table-lookup based strategies, but there are still some drawbacks to the approach. Once the models for the gates and the interconnect network are generated, they are simulated using time-step integration techniques [14] which can lead to long analysis times. In the presence of large variations in the manufacturing process, the problem of extracting these models becomes more complicated as one has to extract the model information not only for the nominal condition but also for different settings of the process parameters resulting in huge model sizes. Finally, in most static timers, the more detailed waveform information is discarded after a single stage; only the delay/slew numbers is retained.

We present an efficient algorithm for generating accurate and robust models of logic gates incorporating that is also suited for incorporating variational information. We also present a scheme for efficient simulation of these models along with reduced interconnect networks resulting in a variational static timing analysis framework. Section 2 presents the basic background on static timing analysis and some of the common approaches. Section 3 describes the key pieces of our algorithm including variation aware model generation for the logic gates and their co-simulation with interconnects. Section 4 presents our experimental results where we demonstrate the efficacy of the proposed methodology using simulation examples of non-ideal waveforms. The time taken for generating these waveforms is potentially 10-20X faster than the current-source models. Finally, Section 5 offers concluding remarks.

978-3-9810801-2-4/DATE07 © 2007 EDAA

II. Background: Static Timing Analysis

A. Problem

The problem of static timing analysis (STA) can be described as follows. We have a digital circuit (Fig. 1) that has inputs IP_i for i=1,2,..,I and outputs OP_j for j=1,2,..,J. The waveforms at these terminals can be represented as $wv(IP_i)$ or $wv(OP_j)$.The circuit has C_k circuit blocks for k=1,2,..,K connected by interconnect networks IN_l for j=1,2,..,L. The problem is to find information about the characteristics of the waveform at the output OP_j e.g. delay, rise/fall-time, overshoot etc. as a function of a particular set of input waveforms $\{wv(IP_1)$,$wv(IP_2)$,..,$wv(IP_I)\}$. The information about the output waveforms is to be obtained in the presence of global parameter variations for the underlying transistor models.

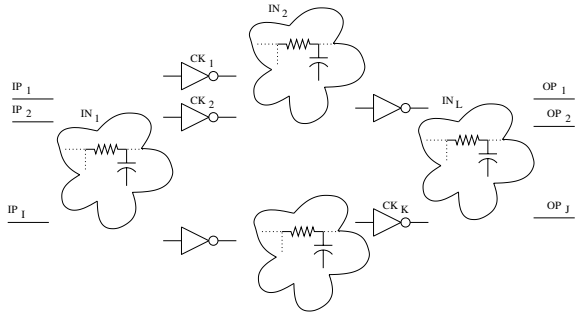

Fig. 1. Representation of a digital circuit with logic gates and interconnect network

B. Previously Proposed Solutions

In traditional STA methodologies, logic levels and corresponding valid transitions between them are propagated from one stage (gate) to another. A voltage transition at a particular circuit node has the topology of a saturated ramp. However, conventional STA simplifies the transition by approximating it with a *linear* saturated ramp. Accordingly, the signal transitions are modeled using just two parameters: arrival time (AT) and slew. Earlier table-lookup delay models were motivated by this abstraction, where these two parameters would be tabulated as a function of input slew and an output load capacitance. These models can have significant inaccuracies as transitions in modern designs are quite different from a linear ramp. Also, seemingly insignificant details of the waveform can affect the delay of the next stages by appreciable amounts. Moreover, interconnects are often very resistive which means that they can hardly be modeled by a single capacitance.

Significant progress has been made in addressing these problems resulting from the underlying assumptions and approximations of transition STA. The simplest, straightforward and most accurate solution is to do a full SPICE [11] analysis of the complete circuit. As this is often seen as computational impractical, the alternative approach is fine grained modeling of the digital gates. That is, instead of characterizing a waveform with a delay and slew number, waveforms are represented using a host of different values like, overshoot, spikes, decay number (time taken to reach from 10% to 1% of logic value) etc. apart from the regular metrics of delay and slew (time taken to reach from 10% to 90% of Vdd). These methods are *ad hoc* at best. They lack a methodical strategy for characterization of the waveforms.

The approach that seems the most promising at handling this problem models the input and output interfaces of digital gates along with applying model order reduction (MOR) techniques for efficiently simulating large, linear interconnect networks, e.g. BLADE [4]. However, most of these methods use a lot of "black magic" to make the models work. Also, they suffer from slow simulation speeds because of the SPICE-like time-step integration techniques involved in solving the reduced circuit equations. To date, variation aware modeling support is also missing or is, at best, *ad hoc* in such methodologies.

We seek to solve these problems using a method we term as *waveform pushing*. In this approach, a given input waveform is sequentially "pushed" through the various interconnects and logic networks of the circuit till it reaches the output. Complete waveforms are solved for at each stage of waveform pushing. This is different from the standard static timing analysis (STA), because in our case, we keep track of the actual shape of the waveform with respect to time, while in STA, only numbers related to certain (hopefully) key waveform characteristics *e.g.*, delay, rise/fall-time *etc.* are moved around across the gates and interconnects. Thus, we seek to use the complete waveforms during fast simulation of our digital circuits to get an accurate behavior of the circuit under a particular test condition.

III. Modeling Methodology

Our waveform based variational static timing analysis methodology uses four basic numerical techniques. First, perturbation analysis allows us to pre-characterize not only the performance of a single, fixed gate instance, but the entire range of behavior of a gate topology. A novel Schur-transform-based model order reduction technique accelerates treatment of parasitic components in the compressed representation. A combination of time-varying modeling and Schur-complement manipulations allow us to rigorously obtain models of I/O loading effects. Finally, a combination of waveform compression and Galerkin approximation allows us to pass a highly compressed representation of the full waveform through a complete circuit analysis. In this section, we present a brief description of each of these methods.

A. Perturbation Based Variation Aware Modeling

We start with efficient representation of the state vectors as a fuction of *small* variations in some circuit parameter (λ). The state vector is composed of the voltage at the capacitive nodes and current through inductors in a circuit. The parameter (λ) could by any process parameter *e.g.*, vth, tox etc. or parameters of the input waveform *e.g.*, rise-time, or other parameters like the supply voltage or the W, L of transistors. This functional representation is done using perturbation analysis [5] [13] of the underlying circuit equations.

SPICE solves a standard circuit simulation problem by casting it into an equation of the form:

$$\frac{dq(x,t)}{dt} + i(x,t) = 0 \tag{1}$$

Here, x is the state-vector of system. For sake of simplicity, we would be dropping the 't' parameter. We could rewrite the equation as a function of the underlying parameter λ.

$$\dot{q}(x,\lambda) + i(x,\lambda) = 0 \tag{2}$$

If we expand i and q around some fixed x_0 and λ_0, we can rewrite Equation 2 in the form of Equation 3 after some massaging of the terms.

$$\frac{d}{dt}\left[\frac{dq}{dx}\bigg|_{x_0}\Delta x + \frac{dq}{d\lambda}\bigg|_{\lambda_0}\Delta\lambda\right] + \frac{di}{dx}\bigg|_{x_0}\Delta x + \frac{di}{d\lambda}\bigg|_{\lambda_0}\Delta\lambda = 0 \tag{3}$$

We can now discretize the equation at time points t^n and t^{n+1}. Taking the backward Euler method as an exmaple, and with some algebraic manipulations, we can get a recursive relation for obtaining Δx^{n+1} as a function of variations in lambda ($\Delta\lambda$).

$$\Delta x^{n+1} = K_x^{n+1}\Delta x^n - K_\lambda^{n+1}\Delta\lambda \tag{4}$$

If we use a linear approximation to compute the change in initial state of the system using the equation

$$\Delta x_{t^0} = \frac{\partial x}{\partial \lambda}\Delta\lambda \tag{5}$$

Then, we can rewrite Eqn (4) as

$$\Delta x^{n+1} = K_{lin}^{n+1}\Delta\lambda \tag{6}$$

For small variations in λ, Eqn (6) can give the state-vector for the circuit at all time points using a simple scalar multiplication. This, in essence, means that once we have pre-characterized a circuit, we don't need to re-simulate it for getting the output waveforms when circuit simulation conditions (λs) are varied (for example, shape of waveform, model parameter of transistors).

B. Gate Modeling Via Schur Complements

The Norton equivalent components for the input/output of a circuit (Figure 2) can be obtained by computing *schur complements* [6] of the detailed, time-varying circuit equations. The basic idea is to rewrite the state-space equation in terms of just one variable (the input or output node). The scalar elements in that equation would automatically give the norton equivalent circuit components.

For a non linear circuit, we start with its *approximate* linear representation to compute the schur complement and the equivalent input/output model for the circuit. We start with the charge-current equation used by SPICE during circuit simulation.

$$\dot{q} + i = 0 \tag{7}$$

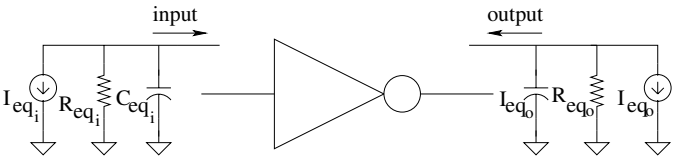

Fig. 2. Norton equivalent model of a logic gate. Note that all elements (R,C,I) in the resulting model are time varying.

Jacobians of charge and current give the capacitance and conductance matrices, respectively, at a particular point in the state-space(v_0).

$$\frac{dq}{dv}\bigg|_{v_0} = C \qquad \& \qquad \frac{di}{dv}\bigg|_{v_0} = G \tag{8}$$

The Jacobians obtained during SPICE simulation at *each* time-step are used to construct an approximation of the non linear circuit composed of only *linear* elements. The linear circuit has the same number of nodes and voltage sources as the non-linear parent circuit. The capacitance and conductance values of the linear network are such that the resulting capacitance and conductance matrices of the linear network matches that of the non-linear circuit at each time-point (linearization point) in its state-space. The voltage values for the linear circuit are forced to take the same values as their corresponding nodes in the non-linear network by introducing new *pseudo* current sources at the different nodes. The values of these current sources are chosen in such a way so as to balance the resulting state-space equation for the linear circuit (Eqn (9)).

$$C\dot{v} + Gv + I = 0 \tag{9}$$

Thus, we would have a different linearized approximation of the non-linear circuit at each simulated time step. For this linear network, we now compute the equivalent output model. The values of current, capacitance and conductance elements of this generated model would be *time varying*.

The equivalent output model for the linear network is obtained by rewriting Eqn (9) as Eqn (10).

$$\begin{aligned} \Rightarrow (sC + G)v &= -I \\ \Rightarrow Yv &= -I \end{aligned} \tag{10}$$

The last equation can be rewritten by breaking the variable v into two parts: $v = [v_1 v_2]^T$ where v_1 is the state-variable corresponding to the output node of the circuit and v_2 is the set of *all* other state-variables. Thus, the state-space equation can be rewritten in terms of these variables:

$$\begin{aligned} Y_{11}v_1 + Y_{12}v_2 &= I_1 \\ Y_{21}v_1 + Y_{22}v_2 &= I_2 \end{aligned} \tag{11}$$

Substituting the variable v_2 in the equations gives

$$\begin{aligned} (Y_{11} - Y_{12}Y_{22}^{-1}Y_{21})v_1 &= I_1 - Y_{12}Y_{22}^{-1}I_2 \\ \Rightarrow Y_{eq}v_1 &= I_{eq} \end{aligned} \tag{12}$$

Rearranging the terms, we get the equivalent input/output conductance, capacitance and current source values (note that these expressions hold on a per-timepoint basis):

$$\begin{aligned} C_{eq} =& C_{11} - C_{12}G_{22}^{-1}G_{21} - G_{12}G_{22}^{-1}C_{21} \\ &+ G_{12}G_{22}^{-1}C_{22}G_{22}^{-1}G_{21} \\ G_{eq} =& G_{11} - G_{12}G_{22}^{-1}G_{21} \\ I_{eq} =& I_1 - G_{12}G_{22}^{-1}I_2 \\ &- s(C_{12}G_{22}^{-1} - G_{12}G_{22}^{-1}C_{22}G_{22}^{-1})I_2 \end{aligned} \tag{13}$$

Note that in Eqn (13), I_{eq} has a first order term in 's'. The way this term of the current source is actually computed is by taking a numerical derivative of I_2 with respect to time. Experimental results have shown that this first order term is very important for the model to function correctly. Perturbation method described in the previous subsection could be easily employed to obtain the equivalent conductance in the form $G = G_{eq} + G_\lambda \Delta\lambda$. Similar parameter dependent components for I and C can also be extracted. Due to space limitations, we are not presenting the detailed methodology.

A similar strategy can be used to compute the equivalent input model for the non linear circuit. Usually, the input equivalent model for a non linear digital gate has only capacitive term. However, due to the particular method of construction of our approximate linear network for the non-linear circuit, the generated model has a current source, as well, to account for the pseudo current sources used to satisfy Eqn (9).

C. Model Order Reduction for Interconnects

Parasitic extraction of interconnects results in large number of linear passive components. It is very expensive to directly solve the resulting system of equations. The standard approach to reduce the cost is to use model order reduction (MOR) techniques [9] to compress these equations. If we have a state-space equation of order N such that the state vector x is of length N as shown in Eqn (14)

$$\begin{aligned} C\dot{x} + Gx &= Bu \\ y &= Lx \end{aligned} \tag{14}$$

We can obtain a reduction matrix V [9] of size $N \times q$ such that $x \simeq Vz$. Thus, Equation(14) can be rewritten using the variable z as

$$\begin{aligned} \tilde{C}\dot{z} + \tilde{G}z &= \tilde{B}u \\ y &= \tilde{L}z \end{aligned} \tag{15}$$

where, $\tilde{C} = V^T C V$, $\tilde{G} = V^T G V$, $\tilde{B} = V^T B$ and $\tilde{L} = LV$. The order of the resulting system of equations (Eqn (15)) is q where $q \ll N$ and hence, Eqn (15) is very easy to solve.

D. Combining Linear MOR and Time Varying Gate Model

Once we have the equivalent time varying model for the logic gates and the reduced order models for the linear interconnect network, we need to solve a combined set of state-space equations incorporating the two. For concreteness, we take the example shown in Figure 3. The RC network represented by the box consists of a large number of R, L and C elements. This linear interconnect network is interfaced with some non-linear circuit elements which are represented by their equivalent time varying norton models. The driver represented with its output model is shown on the left side of the interconnect network and the fan-out load represented by the equivalent input model is shown on the right.

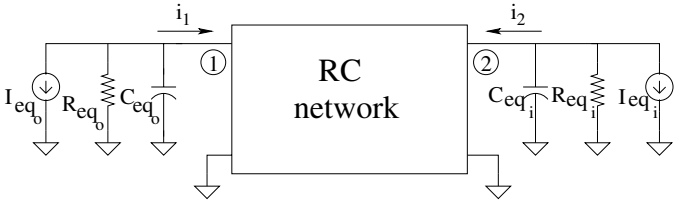

Fig. 3. An linear interconnect network coupled with a time varying equivalent model for a nonlinear circuit

In order to solve this circuit, we need to interface the two circuit blocks: linear network and time varying equivalent model using the current and voltage values at the nodes 1 and 2. Thus, we could rewrite the state-space equation for the whole circuit as:

$$\begin{aligned} I_{th_o} + \frac{v_1}{R_{th_o}} + C_{th_o}\dot{v_1} + i_1 &= 0 \\ I_{th_i} + \frac{v_2}{R_{th_i}} + C_{th_i}\dot{v_2} + i_2 &= 0 \\ v_1 = \begin{bmatrix} 1 & 0 & 0 & \cdot & \cdot & 0 \end{bmatrix} v &= b_1^T v = \tilde{b_1^T} z \\ v_2 = \begin{bmatrix} 0 & 1 & 0 & \cdot & \cdot & 0 \end{bmatrix} v &= b_1^T v = \tilde{b_2^T} z \\ \tilde{C}\dot{z} + \tilde{G}z + \tilde{b_1}i_1 + \tilde{b_2}i_2 &= 0 \end{aligned} \tag{16}$$

New sets of matrix equations can be generated for Eqn (16) from the reduced order state-space equation for the interconnect network by adding 4 new variables viz. v_1, v_2, i_1 and i_2 and their corresponding 4 new equations. The resulting set of equations can then be solved either directly or through the method proposed in the next subsection.

E. Efficient Differential Equation Solution Using Galerkin Method and Schur Transform

The Galerkin method [3] is a standard spectral technique to solve a system of equations of the form

$$\mathbf{L}(y(x)) + f(x) = 0 \tag{17}$$

where $\mathbf{L}$ is any linear differential operator. We start with assuming that our solution is in the form

$$y(x) \approx u(x) = \sum_{j=1}^{N} c_j \phi_j(x) \tag{18}$$

where $\phi_j(x)$ are a set of basis vectors for j=1,2,..,N. We define the residue of Eqn (17) as

$$r(x) = \mathbf{L}(u(x)) + f(x) \tag{19}$$

For obtaining the coefficients c_j we try making the residue orthogonal to the set of basis vectors which results in a set of N equations that can be solved to get the N coefficients.

$$\int_a^b \phi_i(x)[\mathbf{L}(u(x)) + f(x)]dx = 0 \qquad i = 1, .., N \tag{20}$$

As can be observed, Eqn (16) is of the form of Eqn (17). Thus, Galerkin method can easily be employed to solve the combined set of equations for the circuit of Figure 3. However, we can further simplify the computation of the solution by applying the *Schur transform* [6].

We start with the assumption that the solution of the state-space equation (Eqn. 16) for all the state-vectors can be written as a linear combination of a suitably chosen set of basis vectors.

$$\begin{aligned} v_i &\simeq \sum_{k=1}^{q} c_k^i \phi_k(x) \\ &= \Phi C^i \end{aligned} \tag{21}$$

where $\Phi = \begin{bmatrix}\phi_1 & \cdot & \cdot & \phi_q\end{bmatrix}$ and $C^i = \begin{bmatrix}c_1^i & \cdot & \cdot & c_q^i\end{bmatrix}^T$ We can apply the Galerkin method to solve the system of equations given in Eqn (16). If the basis function is a vector consisting of voltage values at different time points, we can do the integration with respect to a basis function ϕ_k (Eqn (20)) by taking a dot product instead of integrating the equation analytically. Rewriting our initial differential equation (Equation 16) in the form of Equation 20 by integrating with respect to all the basis vectors and then rearranging the terms, we get the following set of equations:

$$(C\otimes\Delta+G\otimes I_q)(I_N\otimes S^{-1})\begin{bmatrix} c_1^1 \\ c_1^q \\ \cdot \\ c_N^1 \\ c_N^q \end{bmatrix} + (I_N\otimes S^{-1})\begin{bmatrix} \phi_1^T I_1 \\ \phi_1^T I_N \\ \cdot \\ \phi_q^T I_1 \\ \phi_q^T I_N \end{bmatrix} = 0 \tag{22}$$

where S and Δ are obtained by the eigen value decomposition of $[\Phi^T \dot{\Phi}]$.

$$[\Phi^T \dot{\Phi}]S = S\Delta \tag{23}$$

Here, $\dot{\Phi}$ is the time derivative of Φ and I_N is an identity matrix of size N. Another set of reordering of these equation would give us a huge linear matrix equation of the form of Equation 24. (The terms in the braces represent the size of the matrices and vectors.) However, only the block diagonal elements of the matrices would have non-zero elements which can be very easy to evaluate.

$$A(Nq \times Nq)C(Nq \times 1) = b(Nq \times 1) \tag{24}$$

Here C is a column vector consisting all the C^i's, N is the number of variables in Equation 16 and q is the number of basis vectors. For perturbation models, the same set of equations appear in the form

$$A(Nq \times Nq)C(Nq \times 1) = b(Nq \times 1) + \Delta\lambda b'(Nq \times 1) \tag{25}$$

These set of equations can also be solved very efficiently to extract the coefficients of the basis vectors that satisfy the differential equation of Equation 16. Thus, using Equation 21, we can obtain the voltage waveforms at all the nodes in the circuit. We can repeat the same methodology for different driver-interconnect-load blocks and hence perform the static timing analysis on any given circuit.

IV. Experimental Results

We have implemented the whole flow as described in Section III into our tool that interacts with a SPICE engine that supports BSIM3 [1] device models.

A. Model Generation

Non-linear transistor circuit blocks are replaced by their equivalent norton model using the Schur Complement method (section III-B). The resulting model is a parallel combination of time-varying current source, conductance and capacitance elements. For each gate, an equivalent input and output model is created. These simplified models have the variational information included in them using the perturbation based ideas of Section III-A. In the initial implementation of these ideas, perturbation models have been generated with the slew of the input waveform as the underlying parameter (λ). However, the strategy is generic enough to be easily extended to incorporate other parameter variations like transistor parameters eg. V_{th}, t_{ox} etc. thereby, having probable applications in statistical static timing analysis.

The linear RC network is reduced to a simplified form by using model order reduction techniques (section III-C). The simplified models for linear and non-linear parts of the circuit are then finally co-simulated based on the discussions in section III-D. Efficient computation of the resulting set of equations is done by using the Galerkin method (section III-E).

B. Basis Vectors for Galerkin Method

The basis vectors for use in Galerkin method (section III-E) is computed by first generating a set of possible output waveforms for the non-linear circuit. This is done by varying the load capacitance. The underlying idea is that for a particular circuit instance, the waveforms generated at the output could differ only because of the shape (slew in our example) of input waveform and load at the output of the circuit. Singular value decomposition (SVD) is performed on this generated set of output waveforms. The resulting eigenvectors corresponding to the most significant eigenvalues above a fixed threshold are used as a basis for the voltage waveforms during solution of the circuit equations. This method provides us with a nice trade-off between accuracy and simulation speed-up.

C. Simulation Results

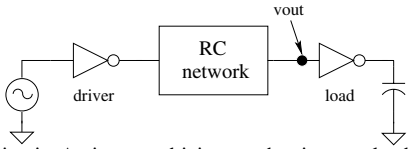

Fig. 4. Test circuit: An inverter driving another inverter load with a 17 node RC interconnect network between them.

We use the circuit set-up shown in Figure 4 as the test case for comparing the efficacy of our modeling approach against full SPICE simulation. In our experiments, we used one cycle of a waveform with period of 2ns and logic transitions at 0.5ns and 1.2ns as input stimulii with different values of rise time. The input and output models for the driver were extracted using a stand-alone inverter driving a single, fixed capacitive load. Figure 5 shows the input equivalent time varying capacitance for the inverter circuit. As can be seen from the plot, Miller multiplication results in large values of input capacitances during the signal transition periods. Since, the models are extracted from underlying SPICE level circuit

equations, this information about equivalent input/output capacitance and conductance of the circuit can also be used by the circuit designers as a design tool to tune their circuit's behavior.

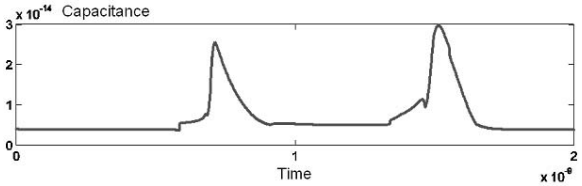

Fig. 5. Time varying input equivalent capacitance for the inverter.

After generating the model for the non-linear gates and the interconnect network, we finally had a 7^{th} order state-space equation (3 from reducing the RC network of order 17 and 4 from section III-D for the gate models). Figure 6 compares the model and output response for a cascaded driver-load set of inverters connected through an interconnect network as shown in Figure 4. As can be seen from the plot, there is a close match between the SPICE and model output even for large deviations of rise time with respect to the nominal (300ps) where the model was trained. For *extremely* large variations from the nominal, the model can be re-trained for different "nominal" values for better accuracy. Figure 7 compares the output waveform produced by SPICE and the model at the input of the load gate of Figure 4. However, this time we trained and tested the model for "spikey" input waveforms as shown in Figure 7. We again see a close match between the model and SPICE outputs.

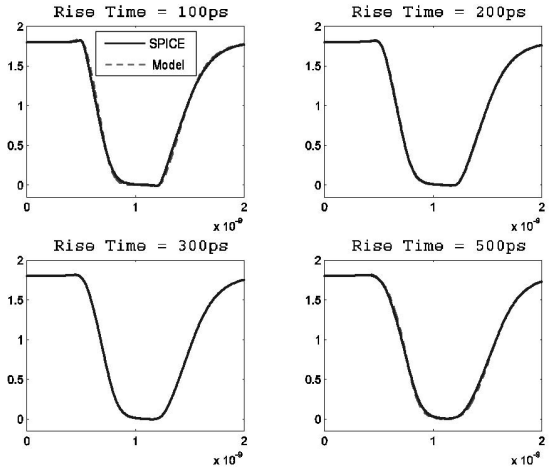

Fig. 6. SPICE and model output for circuit of Figure 4 with different values of rise time of input waveform when the model was only trained for a fixed input rise time (λ_0) = 300ps in the presence of just a single fixed load capacitance.

D. Speed-up Analysis

The modeling scheme presented in this work generates the equivalent models for the logic gates of equal, if not lower, complexity compared to the ones produced using the current-source models [7] [4]. The handling of the interconnect network is the same in both the cases. However, the waveform generation in the current-source models is done using a full time step integration while in our proposed methodology, we need to just compute the coefficients of the basis vectors. (This does however mean a small extra model generation effort in performing the SVD computation). In our experiments, a full time-step integration required evaluations at 50-100 time-steps for simulating a single cycle while in our implementation, we just needed 7 matrix solves. Thus, our proposed methodology is approximately an order of magnitude faster than other similar methods.

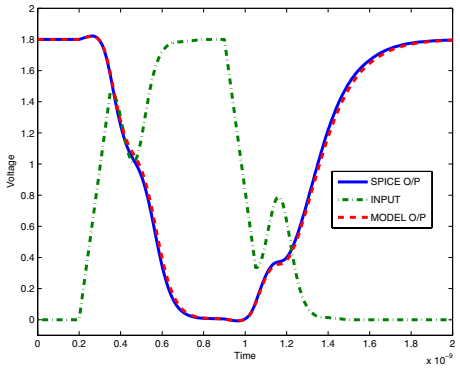

Fig. 7. SPICE and model output for a spiky input waveform at the input of the loading gate of Figure 4.

V. Conclusion

We have presented a novel static timing analysis methodology. The proposed method accounts for non-linear gate capacitance effects, non-linear output driver resistance effects, dynamic effects associated with capacitance on internal cell nodes and multi-stage loading and Miller capacitance effects. Also, it captures the non-ideal waveform shape effects through SPICE-like transient, but fast simulation and the information about the output waveforms is obtained in the presence of parameter variations of the underlying transistor and interconnect models. Our methodology is potentially $10-20X$ faster than similar current-source models. We thus believe that our work presents a platform for incorporating these ideas to perform accurate, yet fast static timing analysis.

References

[1] BSIM3 transistor models. http://www-device.eecs.berkeley.edu/ bsim3.

[2] Ultrasim simulator from Cadence. http://www.cadence.com.

[3] Z. Chen. *Finite Element Methods and Their Applications*. Springer '05.

[4] J. F. Croix and D. F. Wong. Blade and razor: cell and interconnect delay analysis using current-based models. In *DAC '03*, pages 386–389, 2003.

[5] M. H. Holmes. *Introduction to Perturbation Methods*. Springer-Verlag, New York, 1995.

[6] L. W. Johnson, R. D. Riess, and J. T. Arnold. *Introduction to Linear Algebra*. Addison Wesley, 2001.

[7] I. Keller, K. Tseng, and N. Verghese. A robust cell-level crosstalk delay change analysis. In *ICCAD '04*, pages 147–154, 2004.

[8] Y. H. Kim, S. H. Hwang, and A. R. Newton. Electrical-logic simulation and its applications. In *IEEE TCAD*, pages 8–22, 1988.

[9] A. Odabasioglu, M. Celik, and L. Pileggi. PRIMA: Passive reduced-order interconnect macromodeling algorithm. In *TCAD, Vol 17, No 8*, pages 645–654, 1998.

[10] J. Qian, S. Pullela, and L. T. Pillage. Modeling the effective capacitance for the RC interconnect of CMOS gates. In *IEEE TCAD*, pages 1526–1555, 1994.

[11] T.Quarles. The SPICE3 implementation guide. In *UCB/ERL M89/44*, April 1989.

[12] L. Vidigal, S. Nassif, and S. Director. CINNAMON: Coupled integration and nodal analysis of mos networks. In *Proc. 23rd DAC*, 1986.

[13] Z. Wang, R. Murgai, and J. Roychowdhury. Adamin: automated, accurate macromodeling of digital aggressors for power and ground supply noise prediction. In *TCAD*, pages 56–64, 2005.

[14] K. Brenan and S. Campbell and L. Petzold, *Numerical solution of initial-value problems in differential-algebraic equations*. Elsevier 1989.