RC22392 (W0204-022) April 4, 2002
Electrical Engineering

IBM Research Report

TETA: Transistor-Level Waveform Evaluation for Timing Analysis

Emrah Acar, Florentin Dartu and Lawrence T. Pileggi
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

TETA: Transistor-Level Waveform Evaluation for Timing Analysis

Emrah ACAR, Florentin DARTU and Lawrence T. PILEGGI

Abstract*[1]*--Static timing analysis breaks down the longest path problem into waveform analysis of paths of logic stages that are comprised of nonlinear transistors and complex RLC loads. Runtime efficiency is of the utmost importance, however, the waveform evaluation of these logic stages cannot be accelerated via timing simulation algorithms that attempt to exploit temporal or spatial latency since the simulation problem is already a partitioned one. TETA was developed as a general purpose transistor-level waveform evaluation engine for providing accuracy-efficiency trade-offs for these logic-stage waveform evaluation problems that are encountered during timing analysis. Of particular emphasis are the large RC(L) coupled logic stages which present the bottleneck for waveform evaluation along multiple stages of a digital circuit path. TETA applies a novel compaction scheme for the logic-stage transistor clusters and employs a novel nonlinear algebraic solution method to analyze the circuit. Importantly, stability of the waveform evaluation with TETA requires only stable SIMO (single input multi-output) N-port interconnect models that are not necessarily passive. Waveform evaluators that use general transistor and piecewise linear device models require provably passive MIMO (multi-input multi-output) interconnect models that can be extremely inefficient for large coupled N-port problems. Furthermore, the methodology in TETA brings extra efficiency by avoiding extra matrix factorizations and enabling use of device model tables without any loss of accuracy. Complex logic gates and nonlinear capacitors are handled without loss of generality.

***Index Terms*--circuit simulation, timing analysis, transient analysis.**

I. INTRODUCTION

Logic stages are the atomic blocks of digital circuits. Their simulation is of paramount importance in many electronic design automation tasks that include critical path timing analysis, macrocell/block characterization and path optimization. Most digital systems can be efficiently analyzed by partitioning the circuits into logic stages that are solved individually. A typical logic stage includes a set of nonlinear transistors that are dc-connected via source-drain nodes and a linear interconnect load model. Unlike large-scale digital simulation problems, a logic stage does not have spatial and temporal latency which can be exploited via *timing simulation algorithms* to gain significant runtime performance [2,3,4,5,11,15]. For simulation of strongly-coupled transistors and interconnect in logic stages, *traditional circuit simulation*[6] still provides excellent speed-accuracy trade-off .

For deep sub-micron technologies (DSM) the interconnect can have a substantial impact on the delays and overall behavior of the digital signal paths[1]. Some of the logic stages may be strongly coupled via the RC or even the RCL interconnect as well, thereby making the logic-stage behavior a function of the switching activity of the neighboring stages. In such circumstances the logic-stage analyses must incorporate multi-port RCL interconnect models to evaluate the waveforms and timing. In terms of the runtime complexities, it is the logic-stages that are comprised of coupled multi-port interconnects that represent the bottleneck for the overall timing evaluation process. Many timing simulation methods fail to work with a multi-port coupled interconnect load and often attempt to synthesize simple, de-coupled “effective” load models to exploit possible latency. Oversimplification of the interconnect coupling can be very error-prone at certain times risking the fidelity of timing verification.

To manage the complexity of multi-port coupled interconnect significant research has been carried out on reduced order modeling [7,8,18,19,20]. For efficiency, the original RCL interconnect model is replaced by a simpler reduced order model (macro-model). Of particular concern for reduced order modeling of large coupled RCL interconnect is the macromodel stability and passivity. The stability is easily assured by forcing all poles to lie in the left-half plane, but passivity is more difficult to guarantee [8,20]. Passivity is required to ensure stability over all possible impedance models that can be attached to the multiport terminals. While it is possible to ensure that a macromodel is passive[8,20], the price of this passivity is high; namely a multi-input multi-output (MIMO) macromodel for which the number of state variables is equal to the product of the order of the transfer immittance models and the number of ports. In many practical applications this can render the reduced order macromodel to be more complex than the original circuit model.

In this paper we propose TETA, an interconnect-centric approach to waveform evaluation of logic stages. Armed with efficient device and interconnect models, TETA addresses a simplified and accurate timing simulation for multi-port logic stages coupled with large RCL interconnect. Importantly, it solves the passivity dilemma of existing simulation methods by employing a novel approach that enables the safe use of stable but not necessarily passive, reduced order models. In addition, the algorithms in TETA facilitate excellent runtime performance even for transistor dominated circuits and they offer great

[1] This work is supported by in part by the MARCO/DARPA Gigascale Silicon Research Center (http://www.gigascale.org), SRC and Intel Corporation. Their support is gratefully acknowledged. E. Acar is with the IBM Austin Research Labs, 11400 Burnet Rd, Austin, TX 78758 (e-mail: emrah@us.ibm.com). L. Pileggi is with Dept. of ECE, Carnegie Mellon University, 5000 Forbes Ave Pittsburgh, PA 15213 (e-mail: pileggi@ece.cmu.edu). F. Dartu is with Strategic Cad Labs, Intel Corp. 2111 NE 25th St., Hillsboro, OR 97124 (e-mail: florentin.dartu@intel.com). E. Acar was previously with Carnegie Mellon University in preparation of this work.

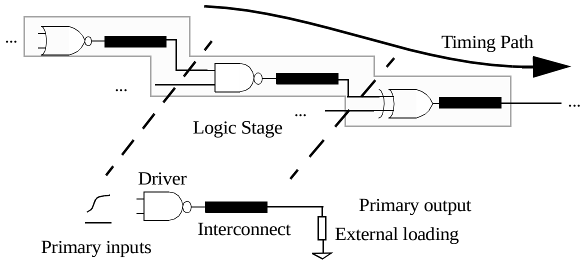

Fig. 1 An example for partitioning a typical timing path into logic stages. The entire circuit is broken into simple logic stages and processed in the order in which the stages are encountered.

opportunities for accuracy-efficiency trade-offs. For example, since the algorithms in TETA allow the use of simplified table models for transistors, the runtime advantage for a logic stage can be significant over SPICE, where the simulation runtime of small to medium size circuits is dominated by the device model calls.

TETA was designed for use as an API waveform evaluation engine that could be used for a myriad of electronic design automation problems. For similar levels of accuracy we compare TETA with SPICE on several benchmarks that are formed in single logic stages and coupled multi-input stages. But we first in section II., review the background on the logic stage extraction, reduced order modeling and general simulation issues. In section III., we outline the cost of existing methods and introduce our new approach. We further describe the numerical methods used in our approach and dissect associated practical issues. Then in section IV., we present results and concluding remarks.

II. BACKGROUND

II.1 Analysis of Logic Stages

The complexity of ICs makes full-chip simulation impractical for functional and timing verification. While the asymptotic order of complexity of accurate circuit simulation is superlinear in terms of the number of circuit elements, most digital circuits display certain features that timing simulators (a more approximate but efficient simulation methods) can exploit for performance. Among these features are temporal and spatial latency. Timing simulators often take advantage of spatial latency by partitioning the original circuit into smaller blocks that can be analyzed more efficiently. They are also enhanced by event-driven or activity-driven methods that exploit temporal latency. For example, to analyze the critical path timing of a digital system, these simulators first divide the path into a chain of logic stages; then analyze each partitioned logic stages with particular input excitations that synthesize the critical signal propagations. However, a single logic stage does not exhibit any temporal and spatial latency. Therefore, timing simulation methods for which performance largely depends on such features generally deliver less accurate results at inferior runtimes when compared to traditional circuit simulation methods for such basic structures.

A typical logic stage includes strongly connected transistors that form a chain of nodes and transistor channels that creates an electrical path from a strong signal source (such as Vdd or Ground) to some output node of interest. It may also include linear circuits for device parasitics. Stages are generally logic gates and the associated interconnect, but may include pass transistors or more general transistor clusters. The fanout load at the output node is included in the logic stage, and is generally modeled as a linear capacitor (C) although the actual load exhibits nonlinear characteristics.

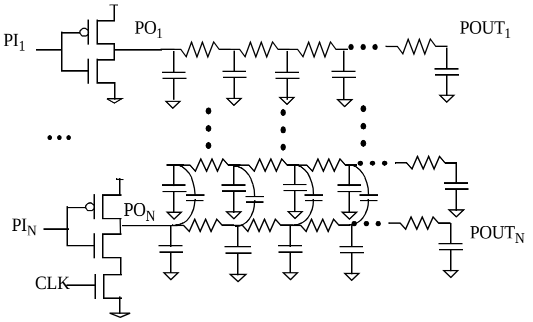

Fig. 2 An example of a N-port logic stage with large multi-port RC coupling: PI: Primary Inputs, PO: Port Output Nodes, POUT: Primary outputs

One of the important consequences of DSM technologies is the dominant interconnect that can also create dominant capacitive/inductive coupling among the wires. This coupling causes interaction of the switching activities between logic stages, and may impose signal degradation as a function of neighboring line switching, thereby complicating any worst/best case delay prediction. To better model the logic stages in the presence of coupling we must employ an *N-port logic stage*. As shown in Figure 2, an *N-port logic stage* consists of *N* different logic stages that are physically coupled via a large RCL interconnect. Detailed and accurate analysis is required for these structures so that timing can be accurately estimated or bounded for design timing signoff.

The presence of large multi-port interconnect models generally complicates any timing simulation methods which assume certain primitive structures for logic stages. Most timing simulation algorithms rely on exploiting the spatial latency of the system, but N-port logic stages do not exhibit such properties. Therefore, such timing simulation methods often assume weak coupling of the actual multi-port interconnect and partially ignore the impact of neighboring signal switching. Usually, they attempt to remove the original interconnect coupling and synthesize “effective” de-coupled load models. This simplification has significant potential to create substantial errors in timing estimation for many DSM circuits. Moreover, it is generally difficult to apply such assumptions without first performing a simulation to assess the impact of the coupling.

II.2 Reduced Order Modeling for N-port interconnect

The direct simulation of the N-port logic stage with large number of RCL elements is a major bottleneck in any timing simulator since evaluating these stages can dominate the overall runtime. To manage the overwhelming complexity of such large stages, several reduced order modeling algorithms have been previously proposed [7,8,10,18,19,20]. The objective of reduced order modeling is to create a smaller size macro-model that accurately represents the complete interconnect system. These macro-models are generally obtained by matching certain characteristics - such as frequency response behavior or moments - of the original linear system. The ever increasing complexity of DSM technologies forces us to favor advanced Krylov subspace [8,19,20] based methods to cope with large coupled multi-port RCL interconnect. Nearly all reduced order modeling methods gener-

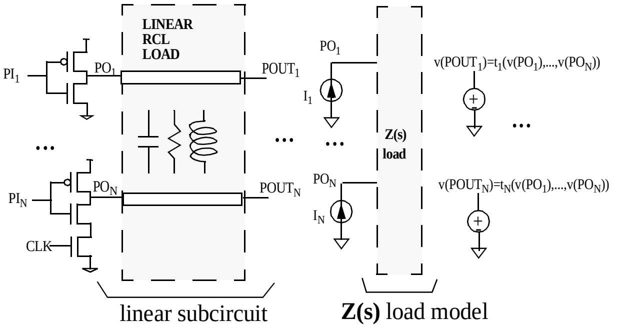

Fig. 3 Illustration of a N-port logic stage with large RCL interconnect load. PO nodes are the port output nodes and POUT nodes are the primary output nodes that propagate the signals to next stages. For the multi-port interconnect, the multi-port impedance model, *Z(s)* is used. Assuming effective ideal current sources at the PO nodes, this model yields $\boldsymbol{V_{PO}}(s) = \boldsymbol{Z}(s)\boldsymbol{I_{PO}}(s)$. The POUT node variables are related to PO nodes via a transfer function by $\boldsymbol{V}_{POUT}(s) = \boldsymbol{T}(s)\boldsymbol{V}_{PO}(s)$

ate frequency domain macro-models that are represented in either pole/residue pairs or in a state space form at significantly less complexity than that of original linear system.

For the N-port logic stage shown in Figure 3, we model the large coupled interconnect by its driving point impedance. Assume that the multi-port RCL load is driven by ideal current sources, we derive the frequency domain relation as $\boldsymbol{V}_{PO}(s) = \boldsymbol{Z}(s)\boldsymbol{I_{PO}}(s)$; where $\boldsymbol{V}_{PO}(s)$ and $\boldsymbol{I}_{PO}(s)$ represent the *N*-dimensional vectors representing the port variables and $\boldsymbol{Z}(s)$ is the $N \times N$ impedance matrix macro-model. The relation between primary output nodes and the port variables can be also formulated via a transfer function, $\boldsymbol{V}_{POUT}(s) = \boldsymbol{T}(s)\boldsymbol{V}_{PO}(s)$. The analysis of primary outputs is essential for propagating the signals to subsequent stages.

With various reduced order modeling methods, each entity of the $\boldsymbol{Z}(s)$ can be modeled in the pole/residue form:

$$z_{ij}(s) = k_{ij0} + \sum_{l=1}^{ord} \frac{k_{ijl}}{s + p_{ijl}}. \tag{1}$$

For simulation efficiency we must assess the complexity of a reduced order model by defining the order of the reduction, P_{order} as the total number of pole/residue pairs in $\boldsymbol{Z}(s)$. When time-domain responses are sought, the $\boldsymbol{Z}(s)$ macro-model is transformed into time-domain or RLCG realizations [9,21,22,23] for simulation with the nonlinear portions of the circuit. Each pole in (1) represents a state variable in this time domain transformation of which the computational cost grows linearly with P_{order}.

Some reduced order model applications require the preservation of the model passivity. The multi-port interconnect is a passive linear system and its macro-model, $\boldsymbol{Z}(s)$ should be passive to ensure overall simulation stability. The passivity of the impedance macro-model is only guaranteed by block version of the reduction algorithms [8,20]. These algorithms, often referred to as multi-input multi-output (MIMO) algorithms, significantly increase the overall computational complexity and the order of approximation. For a general RCL interconnect model, a MIMO algorithm generates *N* pole/residue pairs to match a single moment of the z_{ij}(s). Hence, matching q moments of each z_{ij}(s) yields the order of the reduction to $P_{order} = q \times N^3$. In contrast, single-input multi-output (SIMO) type reduction scheme creates a common set of poles for each column of $\boldsymbol{Z}(s)$, producing only one pole/residue pair for each matched moment of a z_{ij}(s). The SIMO method yields the order of approximation as low as $P_{order} = q \times N^2$ while still matching q moments for each entry. The SIMO method carries fewer pole/residue pairs but the costs of both methods are similarly dominated by $q \times N$ matrix solves. From the simulation point of view however, the SIMO approach is preferred because having fewer pole/residue pairs simplifies the time domain realization. Additionally, while both methods match the same number of moments and have similar accuracy, SIMO cannot guarantee passivity. Clearly this represents a trade-off between the efficiency of the macro-model and the assurance of simulation stability. In this paper, a primary contribution is to present a novel waveform evaluation method that does not necessarily require passive reduced order models and employs more efficiently sized stable macro-models.

The issue of passivity as it leads to instability depends on the interconnect macro-model characteristics, and the nonlinear elements which drive the interconnect. Therefore, to understand the complete problem we must review basic simulation concepts.

II.3 Time Domain Waveform Evaluation

For time-domain analysis, the circuit equations are formulated by a set of nonlinear differential equations via Modified Nodal Analysis (MNA) [16]. Prior to a transient analysis, a dc analysis is applied for solving the initial conditions. Then, for each timepoint, the simulation method applies a combination of 1) A numerical integration method, usually a stiffly stable method such as Backward Euler (BE) or Trapezoidal Approximation (TR), is used to convert the system into a set of nonlinear algebraic equations. This step replaces the energy storage devices with corresponding linear companion models. The circuit is now transformed into a nonlinear resistive form; 2) The transformed nonlinear resistive circuit formulated as a set of nonlinear algebraic equations, is solved by an iterative method. Most simulation engines employ a modified Newton-type method for their good convergence rate and behavior.

The most well known circuit simulation program, SPICE [6], follows this general flow for nonlinear transient analysis. SPICE employs the TR numerical integration scheme and a modified (damped) Newton-Raphson (NR) method for the nonlinear solver as well along with other advanced numerical methods to enhance the runtime performance. Unfortunately, solution via NR will create time-varying impedances at the ports of the interconnect models which when combined with a non-passive model can create an unstable and inaccurate simulation.

II.4 Newton-Raphson method

At the heart of circuit analysis a set of nonlinear algebraic equations is being solved. Often, a variant of the Newton-Raphson (NR) method is employed to solve the nonlinear system iteratively [12,13]. To summarize, the NR iteratively linearizes each nonlinear element and solves the resulting linearized circuit until a convergence condition is achieved. It has a theoretical quadratic rate of convergence with an initial estimate in the vicinity of the exact solution, but this convergence rate can be significantly less in practice when damping methods are applied. In SPICE, and most other simulation engines that employ NR, damping is required to achieve convergence in a reliable manner[6].

The NR linearization is analogous to a first-order Taylor series expansion of the i-v relation and requires the evaluation of the nominal device model and its partial derivative terms. For a two-terminal nonlinear device, such as a diode, it creates a Norton equivalent of which the conductance is the derivative of the device current with respect to branch voltage. The Norton current source completes the i-v characteristic behavior of the nominal device model. For multi-terminal devices, NR linearization produces trans-immittance terms for the branches that are realized with controlled current sources. For a nonlinear device the runtime cost is generally the CPU time required to evaluate the device model and all of the related partial derivatives. With advanced device models having large number of model parameters, this task turns to be very expensive and dominates the overall runtime[11,14] especially for small to medium size circuits. Unfortunately, direct use of table models to speed up the evaluation is less efficcient with the NR based algorithms since they require spline methods for smoothing the partial derivatives.

II.5 The Passivity Bottleneck

As mentioned earlier, traditional circuit simulation methods and other timing simulators require provably passive interconnect macromodels to assure the simulation stability. Essentially the NR based mechanisms used in the simulators demand such requirement. We explain the link between the NR method and passivity using a simple example which is shown in Figure 4. Upon numerical integration, the circuit is transformed into a nonlinear resistive circuit. The NR method linearizes each nonlinear device (MOSFET transistors) and transforms them into Norton equivalents. We apply substitution theorem to lump all the elements in the driver partition into a simple Norton equivalent as shown in Figure 4. In this scenario, the resulting Norton current source (I_{NR}) drives an *effective load* that is made of the interconnection of G_{NR} and the original load. Only a passive load and a passive G_{NR} can guarantee the absolute stability of the *effective load* impedance seen by I_{NR} which brings the overall simulation stability [20]. As mentioned earlier, the passivity is at an expense of the MIMO macro-modeling. In the light of section II.2, stable interconnect macro-models are relatively easier to be generated and facilitated in nonlinear simulation.

The NR based requirements of passive MIMO methods introduces a larger number of state variables for a time-domain waveform analysis. During the analysis of a N-port logic stage with a MIMO macromodel, the total number of the state variables grows super-linearly, and sometimes quadratically with the number of ports. This fact clearly explains why a large N-port logic stage remains a major bottleneck for the simulation and waveform evaluation of all logic stages.

III. THE TETA APPROACH

Our objective is to develop an efficient, high-accuracy waveform evaluation methodology for general N-port logic stages. Prior to presenting our approach, we dissect the major components in existing NR-based timing simulation methods and discuss associated computational complexities.

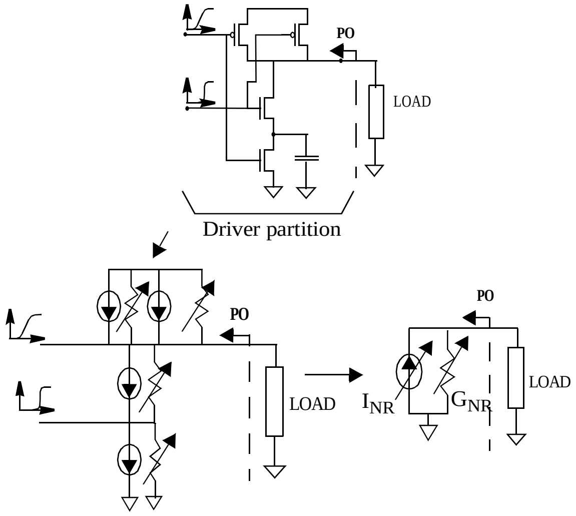

Fig. 4 The NR based macro-modeling of the nonlinear transistors in driver. For each iteration of a particular timepoint, NR based methods linearize the nonlinear devices and produces a nonlinear resistive circuit. Using substitution theorem, these elements can be transformed into a final Norton equivalent at the loading point PO. Possible transconductance terms in the linearization are omitted for simplicity but their presence don't impose any change for the structure of Norton equivalent representation. .

III.1 Dominant Factors of Simulation

The dominant factors of the computational complexity in existing timing simulation methodologies for the N-port logic stages, especially for large *N*, can be listed as:

Nonlinear Device Modeling: The evaluations for nominal and partial terms of the nonlinear device models dominate the total complexity of the simulation. With advanced device models, this evaluation step tends to be very expensive.

Interconnect Modeling: The original N-port interconnect model may be quite large and inefficient for a direct analysis. It may be reduced by an accurate and provably passive MIMO reduced order modeling algorithm. However, MIMO methods are proven to be less efficient inflating the number of state variables in time-domain realization.

Large Linear System Solution: In a typical N-port logic stage, the number of circuit elements and number of state variables associated to load macro-model grow by *N*. Since existing timing simulation methods consider the N-port logic stage as a single entity, the dimension of the associated set of nonlinear equations also grows superlinearly. Therefore, the complexity attributed to the matrix solvers that are used in NR based methods increases quadratically with *N*.

In the development of TETA we focused on finding an algorithm that addresses these three major issues in order to out-perform the existing methodologies. We accomplished this by proposing a novel recipe for timing simulation and waveform evaluation of these logic stages. Our recipe uses an accurate and efficient numerical method that reduces the cost of model evaluation. The proposed numerical method does not strongly depend on certain modeling assumptions and does not induce substantial modeling error. It provides an explicit

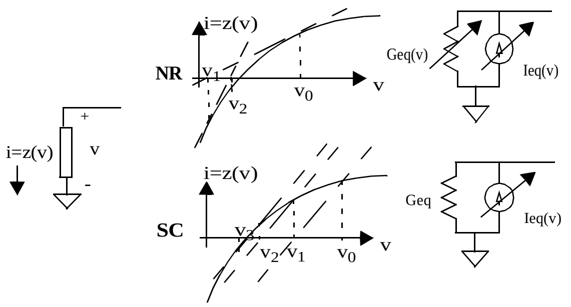

Fig. 5 Successive Chord (SC) and Newton-Raphson (NR) methods: The iterations and corresponding Norton equivalent models. Modeling the nonlinear element at intermediate (i,v) points, the NR method creates a varying-conductance, Geq(v), where the SC employs a fixed conductance Geq.

control of accuracy-efficiency trade-off. Secondly, our simulation method is designed to be compatible with stable but not necessarily passive reduced order models. Bypassing the passivity dilemma brings extra efficiency and flexibility with efficient interconnect modeling. And lastly, noting that the N-port logic stage is considered as a single entity in current simulation methods, its inherent partitioning can be exploited without destroying the coupling. Application of a relaxation method and a novel compaction of nonlinear devices further reduces the problem dimension and gains runtime performance over the timing simulators that use direct methods on the entire N-port logic stage.

III.2 Successive Chords Method in Circuit Simulation

Successive Chords (SC) can be described as a modified Newton iterative nonlinear solution method [12,28]. This nonlinear solution algorithm differs from the NR method by applying constant partial terms in the linearization step. When used in time domain waveform analysis, the SC method linearizes the nonlinear device with constant first order terms making the resulting Norton conductances constant over all iterations. Therefore, at each linearization step, the admittance matrix of the linearized circuit stays constant.

The system in Figure 5 has a nonlinear device whose current is related to its branch voltage via $i = z(v)$. To find the operating point of the open circuit, we solve the nonlinear equation

$$i = z(v) = 0 . \tag{2}$$

At k^{th} iteration of the NR method, the next solution of the branch voltage is found as:

$$v_{k+1} = Geq_k^{-1} Ieq_k \tag{3}$$

where the linearized Norton equivalent is described with $Geq_k = z'(v_k)$ and $Ieq_k = z(v_k) - z'(v_k)v_k$ satisfying the Taylor series expansion around v_k:

$$\begin{aligned} z(v) \cong z(v_k) + z'(v_k)(v - v_k) &= \\ z(v_k) - z'(v_k)v_k + z'(v_k)v &= Geq_k \times v + Ieq_k \end{aligned} . \tag{4}$$

The SC method linearizes the nonlinear device using a constant first order term, z'_0 that follows the scheme $z(v) \cong z(v_k) + z'_0(v - v_k) = z(v_k) - z'_0 v_k + z'_0 v$. This linearization scheme creates a Norton equivalent with $Geq = z'_0$ and $Ieq_k = z(v_k) - z'_0 v_k$. SC's next solution can be found by:

$$v_{k+1} = Geq^{-1} Ieq_k . \tag{5}$$

Observing the constant nature of its conductance, we name this type of Norton equivalent as *semi-Norton equivalent model.* With proper selection of the constant derivative term, the SC method convergences to the exact solution without sacrificing accuracy.

The iterations for both methods are shown in Figure 5. As seen from the figure, the NR method reaches the solution with fewer iterations. The reason is that the SC method has linear rate of convergence, or at most superlinear, compared to the quadratic convergence of NR [28]. However, the cost of one SC iteration is relatively cheaper. As we illustrated in the linearizations, the NR method requires an extra model evaluation of the derivative term, $z'(v)$ that may cost a substantial runtime for complex device models. Both methods require continuity of the nominal device model. However, the NR method additionally requires continuous device model derivatives in order to avoid a divergence. This continuity is usually considered in nonlinear device modeling at an expense of extra parameters and results in more complex equations. Furthermore, the NR method can also perform a slower convergence rate with a step-limiting or damping scheme that controls the solution update [12]. Commercial implementations of the NR method usually use this control for improving the convergence and controlling accuracy.

Application of the SC method for multi-terminal devices is similar. The SC method essentially employs a semi-constant linearization and uses approximate partial terms instead of exact values. For voltage controlled devices, it creates constant conductances and transconductances while matching nominal device models with Norton current sources.

For each timepoint, both the NR and SC methods can solve the circuit composed of Norton equivalents and source elements. The MNA formulation translates the linearized circuit into a linear system $\boldsymbol{Y} \times \boldsymbol{x} = \boldsymbol{b}$. The admittance matrix, $\boldsymbol{Y}$ is formed by all conductance elements and auxiliary branch relations, $\boldsymbol{x}$ is the unknown vector and the $\boldsymbol{b}$ represents the vector of source elements. For each iteration the NR method updates its Norton equivalents, the entries of $\boldsymbol{Y}$ and $\boldsymbol{b}$. However, the SC method updates only the Norton currents and so $\boldsymbol{b}$. By keeping a constant $\boldsymbol{Y}$, the SC method saves significant computation time for matrix formation (also known as stamping) and LU factorization. This advantage becomes very significant for large N-port logic stages.

III.3 MOSFET Models

We propose to model the nonlinear MOSFET devices according to the SC method. The general characteristic equations of a MOSFET device are given as:

$$I_G = 0 \qquad I_D = -I_S = I_{DS} = f(v_{ds}, v_{gs}, v_{sb}) \tag{6}$$

The linearization of (6) around a particular operating point is related to three partial terms $\frac{\partial I_{DS}}{\partial v_{ds}} = g_{ds}$, $\frac{\partial I_{DS}}{\partial v_{gs}} = g_m$ and $\frac{\partial I_{DS}}{\partial v_{sb}} = g_{mb}$. The device stamp, i.e. the contribution to the admittance matrix, is computed using these terms. The linearization also defines a Norton current vector, $\boldsymbol{I_{EQ}}$, that contributes the RHS vector $\boldsymbol{b}$ to match the nominal device model. The SC style linearization uses constant

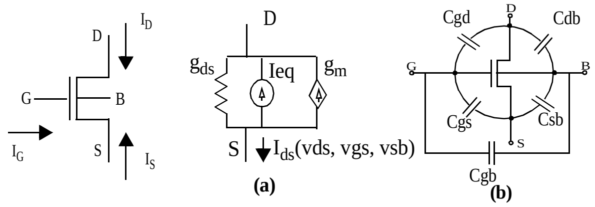

Fig. 6 a) Linearized MOSFET Model: Note that fixed values for g_m and g_{ds} are used in the SC method b) Modeling the capacitances

approximate partial terms for g_{ds} and g_m as shown in Figure 6a. We will approximate g_{mb} as 0, while its partial effect will be considered in the device model.

The selection of approximate terms for g_{ds} and g_m are very critical for convergence of SC method, as these terms directly impact the solution. An optimal selection process should try to find consistent updates within a limited number of iterations. Moreover, the algorithm should avoid the risk of divergence that may be caused by an inconsistent direction. To understand how these terms affect the convergence, we may return to the one-dimensional example given in Figure 5. In that example the convergence of the SC method greatly depends on the selection of the fixed partial term z'_0, i.e. the slope of the chord. Using a steep slope (large z'_0), the SC iterations proceed with small changes in solution updates and thereby take many iterations. In contrast, a flat slope creates larger updates and may cause a divergence if not properly selected. For that example, the average slope value seems to satisfy the convergence and to provide a final solution in a few iterations. For such monotonic concave functions, an average or larger slope will result in a stable convergence behavior.

When applied to relatively smooth, monotonic nonlinear systems with sufficiently close initial guesses, the SC method is quite effective. MOSFET device characteristics are well studied and the device operation is generally classified into several regimes. At cut-off and saturation regions, the device currents have relatively very small sensitivities with respect to terminal voltages. However the device in the linear region is more active and has larger sensitivities. With an analogy to our 1-D observation, the partial terms from linear region represent a more stable update scheme in a SC iteration. Therefore, we select the approximate partial terms g_{ds} and g_m from the linear region. The selection of operating point directly affects the convergence and the robustness of the iterations. Figure 7 shows the runtime comparisons of the prototype with various approximate g_{ds} terms. Based on our observations, we suggest to select the approximate terms based the runtime results of a set of similar benchmark circuits.

In our prototype implementation of TETA the parasitic device capacitances are modeled as in Figure 6b. For simplicity, the gate and channel parasitics are chosen as linear capacitors. A more realistic nonlinear capacitance model is also possible with the SC methodology, but requires a revised linear companion model for the nonlinear capacitance. The nonlinear capacitance can be treated as a two-terminal nonlinear device and its Norton conductance is kept constant for all iterations. The nominal model is satisfied by selecting a compansating Norton current source. An in-depth discussion for nonlinear capacitors can be found in [24,26].

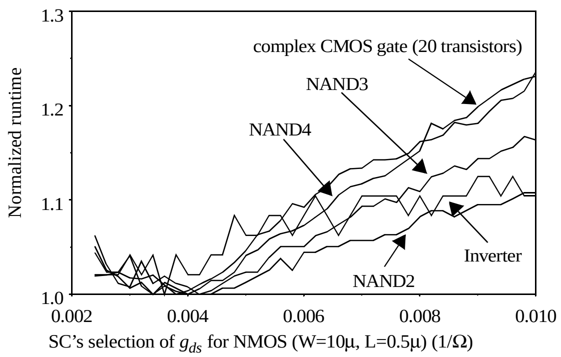

Fig. 7 Normalized runtime for TETA as a function of the NMOS conductance used in the successive chord algorithm.

III.4 Model Tables

The MOSFET device model evaluates the drain-source current, I_{DS} to be incorporated to the right hand side vector, $\boldsymbol{b}$. Many timing simulation algorithms are empowered with simplified device modeling[3,4,5] replacing the complex model equations with piecewise linear models. However, these efficient simpler models come with loss of model accuracy. Since these models are also used to evaluate the partial terms for the NR-based nonlinear solvers, simplified device models have to support the continuity of the nominal and the partial derivative models. On the contrary, the SC method only requires continuous nominal device models. This makes our job significantly easier as we bypass the requirements for the partial terms.

Step or piecewise linear models are very attractive but their generation from actual device models are at a substantial cost and may incur considerable inaccuracy. Model accuracy can be enhanced by more expensive higher order model approximations. We have to note that, one of the disadvantages of event-driven timing simulation methods [3,5] which fully exploit piecewise models is the substantial impact of coarse tables. In our approach, the focus is on evaluating the nominal device model. One of the direct and efficient ways of computing device models is using lookup tables. The table accuracy is adjustable by choosing fine resolutions and more datapoints which do not result in a substantial effect on the algorithm convergence. The lookup-table models also enable us to employ measured data. The direct link between measured data and the timing simulation was not possible in previous methodologies due to the need for explicit evaluation of partial terms.

In TETA we implement MOSFET models in terms of multi-dimensional tables. I_{DS} is modeled as afunction of port voltages $I_{DS}(v_{ds}, v_{gs}, v_{sb})$. We generate a continuous piecewise linear surface using triangular interpolation [17,26] which is very inexpensive compared to the explicit model evaluation. It is very important to observe that the derivative of I_{DS} table-lookup model is not continuous, so these models can be used in conjunction with the SC method, but are not recommended in a NR procedure. It is worth mentioning that I_{DS} scaling with W/L gives us the possibility to keep in memory only one, at most two, table models for each MOSFET type. Additional tables may be necessary to model differently sized devices or other technological attributes such as the narrow channel effect, high threshold voltage.

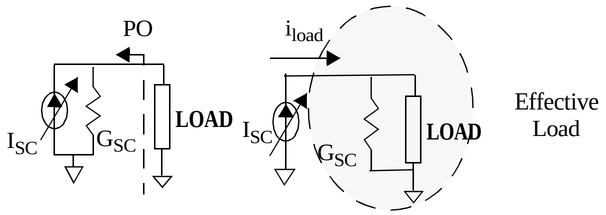

Fig. 8 The construction of the effective load that includes the original load and Norton conductance of the nonlinear driver.

III.5 Incorporating Stable Interconnect Macro-models

The constant linearization scheme applied in the SC method creates a linearized circuit, hence an admittance matrix, $\boldsymbol{Y}$, that remains constant over all nonlinear iterations. As we view the nonlinear portion of the logic stages transformed into the corresponding linearized elements (resistors, controlled sources and independent sources) it is possible to lump all of the linear elements into a final Norton equivalent. As shown in Figure 8, we observe that the Norton conductance, G_{SC}, remains constant with an SC style linearization. G_{SC} is avaliable *a priori* and can be augmented into the original load constructing the *effective load* seen by the current source, I_{SC}. If a reduced order modeling method is performed on the *effective load*, its macro-model can be efficiently used for computing the PO node voltage. While performing these steps, we need the effective load macro-model to be stable since we interface with an ideal source. As a result, this procedure waives the need for provably passive reduced order models. Similar reasoning can be made for the case of N-port logic stages where the original multi-port interconnect and the Norton conductances (G_{sc}'s) make up *effective N-port load*.

Any update on G_{SC} immediately changes the *effective load*, hence $\boldsymbol{Z}(s)$ macro-model needs to be re-calculated. Fortunately, in the SC procedure G_{SC} remains constant for all iterations in a particular timepoint. Moreover, by using a first order numerical integration scheme with a fixed timestep, $\boldsymbol{Y}$ will remain constant for all timepoints. Therefore, $\boldsymbol{Z}(s)$ macro-model needs to be constructed only once with a stable reduced order modeling algorithm. Using a single timestep is extremely effective for problems such as waveform evaluation of logic-stages during timing analysis, since we are only interested in the behavior of the circuit while it is switching, and the nature of these circuits are such that multi-rate behavior is not a problem.

III.6 The Convergence Conditions

The convergence of the Newton methods generally depend on the initial iteration starting point and the properties of the nonlinear function. In addition, the SC method requires the consistency of the approximate partial terms of the admittance matrix $\boldsymbol{Y}$ with the actual ones. If the difference between the approximate and the exact linearization matrices tends to be attenuated during the iterations, the convergence is guaranteed. The detail analysis and conditions for convergence of SC method is given in [27,28]. In our experiences, we find that convergence conditions are satisfied with the selection of the chord models presented here.

III.7 The Algorithm

Setup: Our algorithm assumes the N-port logic stage is clustered into *N* cells compacting the nonlinear devices. The cells are coupled at their PO nodes via multi-port coupled interconnect, as shown in Figure 2. Each cell contains voltage sources (i.e. primary inputs) and a specified output node (PO_i) which drives the interconnect. The far-end node, primary output ($POUT_i$), propogates the signals to subsequent stages.

As a preliminary step for waveform evaluation, some initial computations are required. With a stable numerical integration scheme and a fixed timestep, the storage devices in each cell can be converted to companion models and the nonlinear devices are linearized according to the SC method. The output resistance for each cluster of cells as seen from PO_i nodes ($Rout_i$'s) can be computed via *N* linear matrix solves. The multi-port coupled interconnect RCL load, if any, interconnected with the output resistances ($Rout_i$'s), are crunched with a stable SIMO reduced order modeling algorithm creating the $\boldsymbol{Z}(s)$ *effective load* (*impedance*) *macro-model* and other $\boldsymbol{T}(s)$ transfer function models.

For each timepoint the conductances in the companion models that represent the storage devices and the SC-based semi-Norton conductances for the nonlinear transistors are constant, and stamped into a fixed admittance matrix prior to the simulation. The admittance matrix for each cell, $\boldsymbol{Y}_i$'s can be computed and factored once for all consequent iterations. The output resistance ($Rout_i$) computations are generated as a byproduct of the LU factorizations for solving the transistor clusters.

Simple 1-port case: TETA employs a relaxation solution approach. While the real benefits of this relaxation approach are realized for large N-port logic stages, we first explain main algorithm details for a simple one-port logic stage example. At a particular timepoint (t_0), the circuit is formulated as a nonlinear set of algebraic equations

$$\boldsymbol{f}(\boldsymbol{v}, \boldsymbol{i}, i_{load}, t_0) = \boldsymbol{0} \tag{7}$$

where $\boldsymbol{v}$ and $\boldsymbol{i}$ are the node voltages and the auxiliary branch currents in the cell and i_{load} is the load current that flows in the interconnect load. The load current is a function of the output node voltage at PO that can be formulated as $i_{load} = g(v_{PO}, \boldsymbol{Z}(s))$.

Using an SC based iterative method we apply an inner loop to solve (7) based on an estimate of load current, i_{load}. Starting with initial guesses of $\boldsymbol{v}$ and $\boldsymbol{i}$, the SC method facilitates the constant admittance matrix in linear solves until convergence. During the inner loop steps the semi-Norton current sources vary and are computed based on the recent iterates. Generally, the inner loop terminates when a convergence condition based on the internal node voltages is achieved. Upon this convergence in the outer loop i_{load} is re-evaluated based on the $\boldsymbol{Z}(s)$ realization and latest value of v_{PO}. If the change in i_{load} is significant the algorithm returns to the inner loop with the modified i_{load} estimate; Otherwise the algorithm terminates the outer-loop and reports the final circuit variables (i.e. $\boldsymbol{v}(t_0)$ and $\boldsymbol{i}(t_0)$) for the timepoint. All storage device companion models are updated as well as the time-domain realization of the $\boldsymbol{Z}(s)$ for the next timepoint.

In the outer loop computation the load current and v_{PO} are directly related. As shown in Figure 9, using the time-domain realization, i_{load}

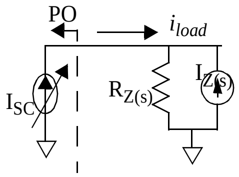

Fig. 9 The relation between v_{PO} and i_{load}. The Z(s) macro-model is realized in Norton equivalent model ($I_{Z(s)}, R_{Z(s)}$). The driver seen from PO is lumped into a semi-Norton current source.

can be computed as

$$i_{load} = (R_{\boldsymbol{Z(s)}})^{-1} v_{PO} - I_{\boldsymbol{Z(s)}} \qquad (8)$$

where $I_{\boldsymbol{Z(s)}}, R_{\boldsymbol{Z(s)}}$ are the Norton equivalent parameters of the time-domain $\boldsymbol{Z}(s)$ realization. Therefore, updating i_{load} essentially results in updating v_{PO} and the change in v_{PO} can be used for the outer loop convergence detection.

The one-port logic stage problem can be directly solved by combining (8) with (7). The direct method makes the outer loop unnecessary, handling the entire circuit in the inner loop. Although it seems more appealing, the direct method introduces severe limitations for large N-port problems. Incorporation of the nontrivial relation between multi-port load currents and v_{POi}'s into the nonlinear circuit formulation raises the problem dimension super-linearly and does not exploit the inherent clustering of the nonlinear devices. Therefore TETA uses a Gauss-Seidel style relaxation method to solve general N-port logic stages.

N-port logic stages: As in the one-port case, we can formulate each cell of the N-port logic stages as a nonlinear function of cell variables ($\boldsymbol{v_i}$, $\boldsymbol{i_i}$) and all load currents:

$$\boldsymbol{f_i}(\boldsymbol{v_i}, \boldsymbol{i_i}, i_{load_1}, i_{load_2}, \ldots, i_{load_N}, t_0) = 0 \qquad i = 1 \ldots N \qquad (9)$$

where the load currents and PO node voltages are related as:

$$\boldsymbol{I_{load}} = \begin{bmatrix} i_{load_1} \\ \ldots \\ i_{load_N} \end{bmatrix} = -\boldsymbol{I_{Zs}} + [\boldsymbol{R_{Z(s)}}]^{-1} \begin{bmatrix} v_{PO_1} \\ \ldots \\ v_{PO_N} \end{bmatrix} . \qquad (10)$$

$$= -\boldsymbol{I_{Z(s)}} + [\boldsymbol{R_{Z(s)}}]^{-1} \boldsymbol{V_{PO}}$$

In (10), $\boldsymbol{R_{Z(s)}}$ and $\boldsymbol{I_{Z(s)}}$ represent the vectors of Norton equivalent parameters in $\boldsymbol{Z}(s)$ time-domain realization.

As we model the entire circuit by N seperate nonlinear set of equations in (9), TETA employs a Gauss-Seidel style relaxation algorithm for each cell. In the inner loop each cell is solved with SC method starting with initial guesses of load currents. Formulated seperately, each cell is decoupled from the others making the nonlinear solver more efficient. For ith cell, the SC method uses corresponding constant admittance matrix, $\boldsymbol{Y_i}$ to solve $\boldsymbol{f_i}(\boldsymbol{v_i}, \boldsymbol{i_i}, \boldsymbol{I_{load}}) = 0$ iteratively. Once it is solved, the impact of v_{POi} on all load currents are evaluated and i_{load}'s are updated. Therefore, in the inner loop iterations of the remaining cells, the latest solution of the ith cell are being reflected in the $\boldsymbol{I_{load}}$ vector. After solving all cells, outer loop checks a convergence criterion based on the changes in the $\boldsymbol{I_{load}}$ vector. Similar to the one-port case, load currents and the PO node voltages are directly related via (10) so that the outer loop may alternatively facilitate a convergence criterion based on the v_{POi} updates. If such convergence is achieved, the algorithm exits the outer loop, reports the circuit variables and updates the companion models of storage devices and the macro-model realization.

Post completion: As TETA calculates the near end PO node voltage waveforms, the far end primary outputs (POUT nodes) can be computed by another time-domain transformation of the transfer function, $\boldsymbol{V_{POUT}}(s) = \boldsymbol{T}(s)\boldsymbol{V_{PO}}(s)$. This transformation is straightforward via previously reported methods [9,22,23].

Summary: TETA facilitates a Gauss-Seidel relaxation algorithm that exploits the inherent compaction of the transistor clusters. A direct nonlinear solution of the entire N-port logic stage superlinearly inflates the problem dimension. Furthermore, in the proposed relaxation algorithm, the inactive cells are usually solved easily in a few iterations thereby exploiting the temporal latency and improving the the runtime performance.

In addition, where existing timing simulators based on NR based nonlinear solvers cannot guarantee simulation stability with stable-only reduced order models, TETA fully exploits the full structure of the effective load and efficiency of SIMO macro-models. After the inner loop of the ith cell, only the ith column of $[\boldsymbol{R_{Z(s)}}]^{-1}$ matrix is used to incrementally update the I_{load} vector avoiding any matrix inversions.

The runtime of a timing simulator is dominated by the model evaluation calls. TETA uses lookup-table device models without trading off the model accuracy. We observed that selecting 20-40 datapoints along the operating voltages (v_{ds}, v_{gs}) and 5-10 sample points for v_{sb} provides extremely accurate device models justifying the storage requirements. For complex device models TETA remarkably outperforms the existing simulation methods by avoiding the model evaluations. TETA's runtime is generally dominated by backward/forward substitutions which has a linear order of complexity. Like all Newton based methods, SC method performs better with a good initial estimate. As our TETA methodology is designed to tackle certain N-port logic stages, we assume that information on such initial conditions are available as part of the timing analysis process. For example, during timing analysis the sensitization conditions are known and applied for analyzing the switching behavior during the delay or waveform calculation step. Our algorithm is summarized in TABLE I.

The performance of our proposed method can be enhanced by several practical and numerical techniques to control and update the associated admittance matrices. The SC method can be enhanced by step-limiting and line-search algorithms that improve the linear convergence rate. The variable timestep schemes can be applied by storing multiple admittance matrices for each cell. Possible minor corrections on the approximate admittance matrices are also easily handled by small-rank updates and other quasi-Newton techniques.

IV. RESULTS

For all examples presented in this section we compare TETA with HSPICE[1] using level 3 MOS models for a 0.5 micron MOSIS technology. For a fair comparison we used the same capacitance models,

[1] HSPICE is a Registered trademark of Meta-Software, Inc.

TABLE I

TETA algorithm for simulating N-port logic stages

```
INITIALIZATION:
Construct Norton equivalents for storage elements and semi-Norton equivalents of the nonlinear devices
Construct the admittance matrices based on MNA formulation ($\mathbf{Y_i}$'s) using Norton equivalents
Calculate the cell output resistances (Rout$_i$'s) based on dt
Construct effective load macro-model $Z(s)$ using stable SIMO methods and time-domain realization ($I_{Z(s)}$,$R_{Z(s)}$)
Construct a stable transfer function macro-model, $T(s)$ between PO and POUT nodes
Set initial conditions for the node voltages

TIME LOOP: for t = t0 : dt : tfinal {
OUTER LOOP:
            do {    Compute load currents $I_{load}$ based on $v_{PO}$ using (10)
                    INNER LOOP:
                    for i = 1 : N {
                        do { Apply SC iterations to solve $f_i(v_i, i_i, I_{load}) = 0$ }
                        while { not a convergence based on $((v_i, i_i))$ vectors }
                        update $I_{load}$ incrementally based on latest $v_{POi}$ using (10)
                    }
            } while { not all $v_{POi}$ values converge }
Report $v_i(t)$ and $i_i(t)$ for the $i^{th}$ cell, $i = 1, 2 .. N$
Solve POUT node voltages using $v_{POi}$ 's and a standard time-domain realization for $T(s)$
Update companion models for storage elements
Update time-domain realizations for $Z(s)$ and $T(s)$
}
```

node voltage tolerances and timestep for both simulators. The CPU times for HSPICE include only the *transient analysis* times as reported in its output. Moreover, we used the option FAST that optimizes HSPICE for further speedup.

Our prototype implementation of TETA is written in C. It is able to handle MOS transistors, resistors, capacitors and general *N*-port $\boldsymbol{Z}(s)$ and $\boldsymbol{T}(s)$ macro-models that can include any combination of coupled RLC components. The MOS transistor i-v models were built as 3-D tables in order to take into account the body effect.

We expect the smallest speed-up advantage of TETA in analysis of simple logic gates without interconnect. In our first experiment we analyzed a 4-input NAND gate driving a purely capacitive load. For TETA the trade-off between speed and accuracy is obtained by modifying the node voltage tolerances for both inner and outer loops. TABLE II presents the run time dependence on outer loop relative and absolute node tolerances, RTOL1 and VTOL1, respectively, as well as inner loop tolerances, RTOL2 and VTOL2. The output waveform generated by TETA for the worst case in terms of accuracy, hence the best in terms of run time, is shown in Figure 10. It is worth noting that for this example TETA offers a remarkable 60x speedup over HSPICE.

TABLE III shows a comparison between TETA and HSPICE for a set of CMOS gates. For this set of examples, nominal capacitance loads are used and MOS models include linear grounded capacitors in both simulators. Also, both simulators used a fixed timestep of 10ps. The use of table models for the MOS I-V characteristics and bypassing matrix stamping and factorization represent the main sources of runtime speedup for TETA in these cases. The results in TABLE III show a linear speed-up performance with the circuit size. This behavior is explained by the linear convergence rate of the successive chord method.To demonstrate the advantage of TETA in the case of simpler device models, we also include the speed up comparisons obtained for level 1 device models for the same process technology. From computational cost perspective, evaluation of a level 1 device model is relatively equal to that of the proposed interpolation model, however TETA provides a significant speedup even for this device model.

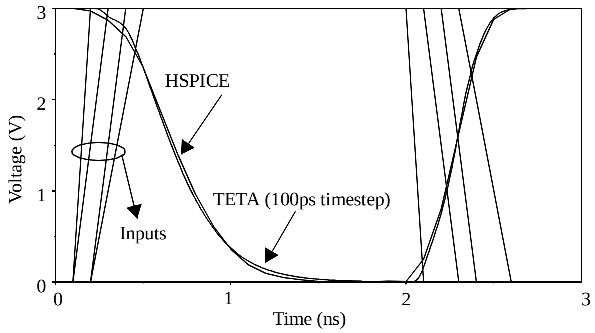

Fig. 10 The effect of tolerance variation on TETA's accuracy. TETA used RTOL1=1%, VTOL1=30mV, RTOL2=30% and VTOL2=300mV.

We have observed runtime speedup performances between 10-100x for other nonlinear circuits which do not include large numbers of circuit elements. The speedup is generally achieved with efficient interpolation-based device modeling, use of SC method avoiding costly matrix operations. Our experiments indicate that the ratio of the total SC iterations over total NR iterations for all timepoints is in the

TABLE II

Run times for a 4-input NAND gate as a function of node voltage tolerances (one pulse transition and 30ps time step)

RTOL1	VTOL1	RTOL2	VTOL2	Run time
1%	30mV	30%	300mV	0.86ms
0.5%	30mV	30%	300mV	0.9ms
0.1%	1mV	30%	300mV	1.7ms
1%	10mV	0.1%	100mV	1.24ms
1%	10mV	1%	300mV	1.05ms

TABLE III

Run time comparisons between TETA and HSPICE for capacitive loads and voltage tolerances of 30mV and 3% (one pulse). Both simulators used a 10ps timestep for which TETA presents no discoverable error.

Example	HSPICE (ms) w/ level 3	TETA (ms) w/level 3	speed-up w/level 3	speed-up w/level 1
Inverter	95.9	1.4	68.5	62.1
NAND2	246.54	4.2	58.7	48.3
NAND3	354.6	6.1	58.1	46.9
NAND4	531.52	8.8	60.4	39.2
Complex CMOS gate (20 transistors)	1630.4	26.2	62.2	36.4

range of 2:1 to 6:1, and the average ratio is around 3:1. This is in part due to smooth nonlinearity of the transient analysis and availability of relatively close initial estimates for each timepoints which may not exist in a general nonlinear DC analysis. We have to note that a SC iteration is very cheaper than that of a NR iteration.

Importantly, TETA yields its best efficiency when simulating large N-port macromodels that represent large coupled RCL circuits. To demonstrate TETA's efficacy for such problems we analyzed a 16-bit bus line with every line capacitively coupled to adjacent lines. The complexity of the linear circuit is modeled by the number of nodes of the linear circuit. HSPICE simulated the flat circuit using a timestep of 30ps. Figure 11b shows the accuracy comparison between HSPICE and TETA using an aggressive timestep of 200ps (for 30ps there is no discoverable error in TETA results). The per transition run times for TETA are 1.44ms and 4.4 ms for timesteps of 200ps and 30ps, respectively. In Figure 11a we present the speed-up over HSPICE as a function of the number of nodes of the linear circuit. Due to the sparsity and the dominant effect of nonlinear model evaluation, the HSPICE simulation times depend only linearly on the number of nodes.

As one example of a stage with non-coupled RLC interconnect, we simulated an inverter driving an industrial clock tree for which the inductive component plays a significant role. The waveforms for the driving point of this clock tree are presented in Fig. 12. HSPICE simulated the full linear circuit containing 160 nodes. TETA simulated a 9th order macromodel having four pairs of complex poles and one real pole. Due to the small time constants presented by the linear circuit it was necessary to simulate the circuit with a time step of 1ps. TETA obtained the results shown in Fig. 12, with a runtime speed-up over HSPICE of 327x.

To check the method for any simulator-induced overshooting due to the integration scheme, it is also useful to test with a very strong driver driving a long, lossless transmission line with an open circuit load when the driver is excited by a fast input signal. For that purpose we simulated a one-port stage with a large inverter driving a RLC line of R=5 ohms, C=200 ff and L=1 uH. The stage is excited by a step input. The RLC line is chopped into two segments and its reduced order model at the fullest order is taken to avoid the differences in the macromodeling step. Figure 13 shows a perfect aggreement between HSPICE and TETA results. Using level 3 device models, the obtained runtime speedup for this example is 56x.

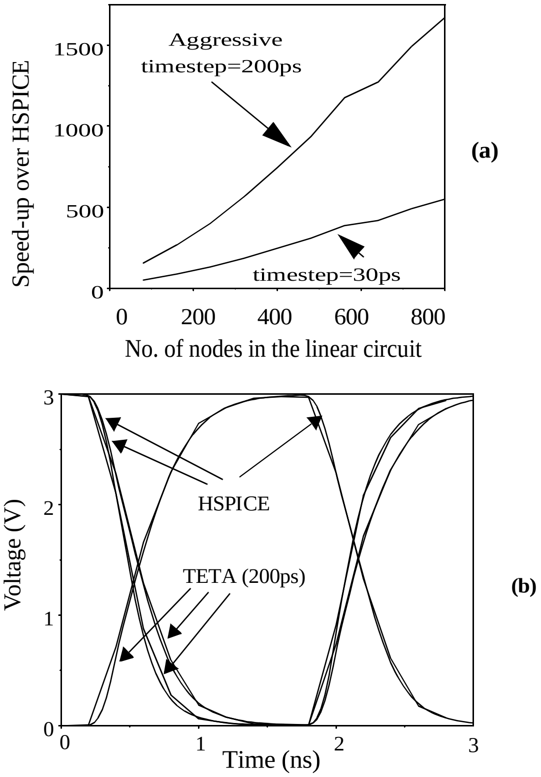

Fig. 11 A 16-bit bus example with every line coupled to the adjacent lines. a) Speed-up as a function of the number of nodes in the linear circuit; b) Response waveform comparison.

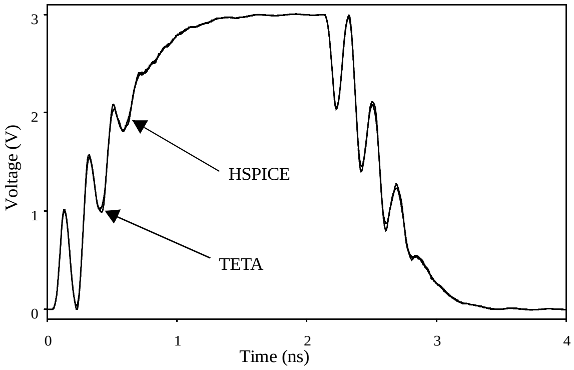

Fig. 12 A clock tree with significant inductive component. HSPICE simulated the full circuit containing 160 nodes while TETA simulated a 9th order macromodel.

The accuracy-efficiency tradeoff in TETA is adjusted via proper setting of the timestep and the table sizes. As we have seen in the

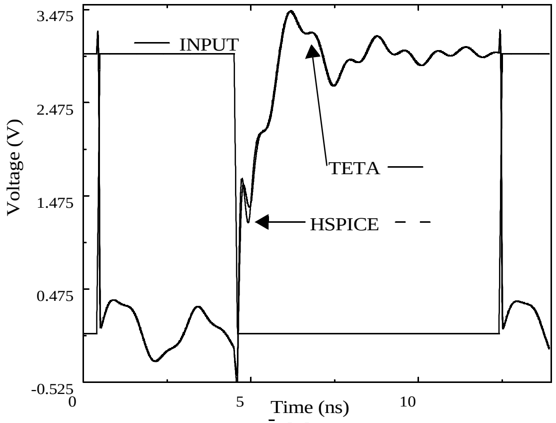

Fig. 13 A strong inverter is excited by a step input source. The inverter is driving a low-loss RLC line of R=5 ohms, C=200 ff and L=1 uH. Exact order is taken in the reduced order model for the RLC line.

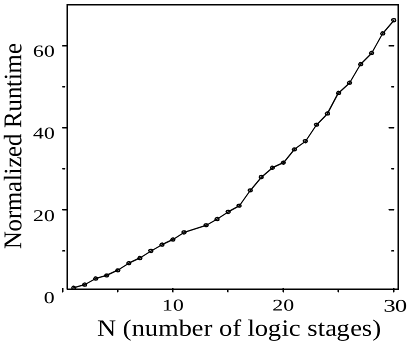

Fig. 14 The runtime profile for a N-port logic stage composed of NOR gates driving a large multi-port RC interconnect with respect to number of logic stages, *N*. RC lines are created by 20 segments with coupling to adjacent lines. *Z(s)* macro-model is created by PRIMA with order 4.

results, the waveforms are computed very accurately even with aggressive timesteps. As a final example, we demonstrate the scalability of the TETA for large N-port logic stages. We experimented an N-port logic stage where *N* NOR gates are driving an N-port coupled RC interconnect. The load macro-model is generated by a 4th order SIMO model with PRIMA [20]. Figure 14 shows the total simulation runtime versus the number of coupled stages. Each stage drives an equal amount of capacitance half of which is coupled to adjacent lines. As seen from the plot, the runtime profile is almost linear with the number of logic stages which is a very important benefit gained by the Gauss-Seidel relaxation algorithm used in TETA.

V. CONCLUSION AND FUTURE WORK

TETA is an interconnect-centric transistor level waveform evaluation engine intended for a myriad of electronic design automation tasks. The overall simulation methodology is built upon a successive chord nonlinear solver that delivers high efficiency for analysis of logic stages coupled with large multi-port interconnect. Admittedly the SC method does not provide all desired convergence properties of the NR method in order to be fully utilized for general analog simulation analysis; but it offers outstanding efficiency and robustness for analyzing digital *N*-port logic stages such as the problems encountered during waveform evaluation in timing analysis. TETA *is not* a full-scale timing simulator, and its purpose is to simulate only relatively small circuits that are formed from partitioning of CMOS digital circuits into strongly connected components.

One practical application of TETA is the computation of the coupling noise on delay. Aggressor alignment with the victim drivers is non-trivial and there is a need for fast and accurate waveform evaluation to determine the worst case. Recently, TETA showed significant advantage in [25] as a part the waveform iteration strategy that computes the delay in the presence of coupling with proper worst-case aggressor alignment.

TETA also offers many promising features that can be utilized in a statistical simulation environment. The approximate linearized device models are less sensitive to parameter variability. Hence a nominal device linearization is applicable for each perturbed device, making a statistical simulation more efficient [26]. Such use of the TETA engine is the current focus of on-going research.

ACKNOWLEDGEMENT

The authors would like to thank the reviewers for their valuable comments. E. Acar would like to thank Dr. Sani Nassif of IBM for numerous productive discussions.

REFERENCES

[1] M. Bohr, “Interconnect scaling - the real limiter to high performance ULSI,” in *Intl. Electronic Device Meeting*, 1995, pp. 241-244.

[2] B. R. Chawla, H. K. Gummel, P. Kozak, “MOTIS - an MOS timing simulator,” *IEEE Trans. on CAS*, vol. CAS-22, pp. 901-910, Dec. 1975.

[3] C. Visweswariah, R. A. Rohrer, “Piecewise approximate circuit simulation,” *IEEE Trans. on CAD*, pp. 861-870, Jul. 1991.

[4] Y. H. Shih, Y. Leblebici, S. M. Kang, “ILLIADS: a fast timing and reliability simulator for digital MOS circuits,” *IEEE Trans. on CAD*, pp. 1387-1402, Sep. 1993.

[5] A. Devgan, R. A. Rohrer, “Adaptively controlled explicit integration,” *IEEE Trans. on CAD*, vol. 13, pp. 746-762, Jun. 1994.

[6] L. W. Nagel, “SPICE2, a computer program to simulate semiconductor circuits,” Tech. Rep. Memo UCB/ERL M520. Univ. of California, Berkeley, May 1975.

[7] C. L. Ratzlaff, L. T. Pillage, “RICE: Rapid interconnect circuit evaluation using AWE,” *IEEE Trans. on CAD*, vol.13, pp. 763-776, Jun. 1994.

[8] K. J. Kerns, “Accurate and stable reduction of RLC networks using split congruence transformations,” Ph.D. thesis, Univ. of Washington, Sep. 1996.

[9] S. Y. Kim, N. Gopal, L. T. Pillage, “Time-domain macromodels for VLSI interconnect analysis,” *IEEE Trans. on CAD*, vol. 13, pp. 1257-1270, Oct. 1994.

[10] D. F. Anastasakis, N. Gopal, S. Y. Kim, L. T. Pillage, “Enhancing the stability of asymptotic waveform evaluation for digital interconnect circuit applications,” *IEEE Trans. on CAD*, vol 13, pp 729-736, Jun. 1994.

[11] R. A. Saleh, A. R. Newton, *Mixed-mode Simulation*, Kluwer Academic Publishers, 1990.

[12] W. J. McCalla, *Fundamentals of Computer-aided Circuit Simulation*, Kluwer Academic Publishers, 1988.

[13] L. T. Pillage, R. A. Rohrer, C. Visweswariah, *Electronic Circuit and System Simulation Methods*, McGraw Hill Inc., 1995.

[14] A. Devgan, “Dynamic timing simulation,” presented at IEEE/ACM Design Automation Conf., 1996.

[15] A. Vincentelli, E. Lelarasmee and A. Ruehli, “The waveform relaxation

method for the time-domain analysis of large scale integrated circuits," *IEEE Trans. Computer Aided Design*, vol 1, pp 131-145, Aug 1982.

[16] C. Ho, A. Ruehli and P. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. on Circuits and Systems*, vol 22, no 6, pp 678-687, 1975.

[17] J. J. Rotman, *An Introduction to Algebraic Topology*, Springer-Verlag, 1988.

[18] L. Pillage and R. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer Aided Design*, vol 9, pp 352-366, Apr. 1990.

[19] P. Feldman and R. Fruend, "Efficient linear circuit analysis by Pade approximation via the Lanczos process," *IEEE Trans. Computer Aided Design*, vol 14, pp 634-649, May 1995.

[20] A. Odabasioglu, M. Celik and L. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *Proc. IEEE Int'l Conf. Computer Aided Design*, pp 58-64, Nov. 1997.

[21] A. Odabasioglu, M. Celik and L. Pileggi, "Practical considerations for passive reduction of RLC circuits," *Proc. IEEE Int'l Conf. Computer Aided Design*, pp 214-219, Nov. 1999.

[22] X. Huang, V. Raghavan and R. A. Rohrer, "AWEsim: A program for the efficient analysis of linear(ized) circuits," *Proc. IEEE Int'l Conf. Computer Aided Design*, pp 534-537, Nov. 1990.

[23] Y. Liu, L. Pileggi and A. J. Strojwas, "ftd: An exact frequency to time domain conversion for reduced order RLC interconnect models," *Proc IEEE/ACM DAC*, pp 87-92, Jun. 1998.

[24] F. Dartu and L. Pileggi, "TETA: Transistor-level engine for timing analysis", *Proc. IEEE/ACM DAC*, pp 595-598, Jun 1998.

[25] P. Gross, R. Arunachalam, K. Rajagopal and L. Pileggi, "Determination of worst-case alignment for delay calculation", *Proc. IEEE Int'l Conf. Computer Aided Design*, pp 212-219, Nov 1998.

[26] E. Acar, "Linear-centric simulation approach for timing analysis," Ph.D. Thesis, Carnegie Mellon University, Nov. 2001.

[27] K. Brenan, S. Campbell and L. Petzold, *Numerical Solution of Initial-Value Problems in Differential Algebraic Equations*, Elsevier, New York, 1989.

[28] J. M. Ortega and W. R. Rheinbolt, *Iterative Solution of Non-linear Equations in Several Variables*, Academic Press, New York, 1970.