SYMPHONY: AN EFFICIENT MIXED SIGNAL CIRCUIT SIMULATOR

by

Premal Buch and Ernest S. Kuh

Memorandum No. UCB/ERL M95/24

10 April 1995

SYMPHONY: AN EFFICIENT MIXED SIGNAL CIRCUIT SIMULATOR

by

Premal Buch and Ernest S. Kuh

Memorandum No. UCB/ERL M95/24

10 April 1995

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

SYMPHONY: An Efficient Mixed Signal Circuit Simulator[†]

Abstract

With the growing use of mixed signal design techniques, there is a need for tools to simulate mixed analog/digital designs. Along with CMOS, there is also increasing use of bipolar technology in today's high performance designs. To our knowledge, the issue of simulation of mixed analog/digital designs with bipolar devices has not been addressed before. In this work, we present SYMPHONY, a mixed signal simulator that fills this void using techniques specifically targeted at exploiting characteristics of such circuits.

SYMPHONY combines a fast simulator for digital subcircuits with a traditional nonlinear solver *a la* SPICE for the analog subcircuits.

Dynamic partitioning for BiMOS circuits in an event-driven framework, and a new PWL model for bipolar device characteristics using expanded Chebyshev points, are the two main contribution of this work. Dynamic circuit partitioning is used to fully exploit the latency and multirate behavior of the circuit. The PWL model allows us to speed-up simulation by using simple models for bipolar devices in digital parts of the design. These two techniques are combined in SYMPHONY to dynamically identify subcircuits with potentially analog or digital behavior.

A set of benchmark results are presented on a suit of BiMOS circuits.

[†] This work was supported by a grant from SRC contract 95-DC-324.

1 Introduction

Advances in IC technology and on the manufacturing and packaging front have made mixed signal design with analog and digital circuitry on the same chip a reality. The growing interest in mixed signal circuit design has also given rise to a new set of challenges to CAD tools. To verify the functionality of the design at the transistor and layout level, circuit simulation is still one of the most important tools. While it is always possible to simulate the entire transistor level circuit with time trusted tools *a la* SPICE, for large state-of-the-art circuits this has long ceased to be a feasible option. There has been some work in this area, but by and large the standard solution to the problem of mixed signal circuit simulation is to use fast tools (which trade-off accuracy for speed) for digital parts of the circuit, more accurate simulators for the analog parts and hope that there are no problems when both are put together in the same design. There are some simulation frameworks that provide more refined interfaces between fast, event-driven digital simulators and time point driven analog simulation [1] but this approach is valid only for circuit with little or no feedback from analog to digital subcircuits (and vice versa). Another approach to mixed signal simulation is to take an analog simulator and try to incorporate digital circuit simulation techniques into it [26][38]. Since the basic framework is still that of a slow simulator, this is not very efficient for large circuits which are mostly digital. Analytic macromodeling [4] is another efficient approach to this problem.

One problem not addressed in these works is that of circuit partitioning to identify analog and digital blocks in the design. Some industrial tools use a library based approach to identify such subcircuits. Most of the tools expect user input to resolve this problem. While this may be acceptable (though inconvenient!) in the top-down phase of the design, it cannot be used to verify flattened circuits extracted from the layout.

With the push towards faster and faster designs, there is a growing use of bipolar devices in digital designs. Most of the research in recent years has been concentrated on developing fast circuit simulators for CMOS circuits. These simulators either cannot handle BiMOS circuits or perform poorly on them since they do not exploit any of the special characteristics of these circuits.

In this work, we present SYMPHONY, a mixed signal circuit simulator which uses special techniques targeted to address the above issues. SYMPHONY combines a fast simulator for digital circuits with a traditional nonlinear solver *a la* SPICE for the analog subcircuits. The digital simulator uses Stepwise Equivalence Conductance to model nonlinear device conductances and Piecewise Linear voltage waveforms. Device characteristics of bipolar elements in digital subcircuits are modeled by a Piecewise Linear approximation using the expanded Chebyshev Points, such that the worst case approximation error is minimized. Dynamic circuit partitioning is used to fully exploit the latency and multirate behavior of the circuit and dynamically identify subcircuits with analog and digital behaviors. The simulator is implemented in an event-driven framework with local and global clocks for event management.

This paper is organized as follows: In Section 2 we give the historical background and an overview of the circuit simulation problem. We briefly review the merits-demerits of some of the alternative techniques available to us for the digital and analog simulation engines of a mixed signal simulator.

Section 3 introduces SYMPHONY, a mixed signal simulator developed as a part of this work. We describe the simulator framework and details of event management and circuit partitioning in SYMPHONY. Section 4 describes the dynamic partitioning techniques used to exploit the latency and multi-rate behavior of BiMOS circuits and its implementation in SYMPHONY. In Section 5, we present a new PWL approximation model for bipolar device characteristics. Theoretical results on the accuracy of the model are presented along with the implementation details in the SYMPHONY framework for digital subcircuits containing bipolar devices.

In Section 6 we present some experimental results on a set of BiCMOS circuits and Section 7 concludes with a summary of this work.

2 Previous Work

2.1 Historical Background

Assuming for the sake of simplicity, that there are no inductors and only constant capacitors in the circuit, the circuit simulation problem consists of solving a set of nonlinear equations of the form [8]

$$\mathcal{F}(\mathbf{V}(t)) + \mathbf{C}\dot{\mathbf{V}}(t) = \mathbf{I}_S(t) \tag{1}$$

where $\mathbf{V}(t)$ is the node voltage vector, $\mathcal{F}(.)$ is a vector function of $\mathbf{V}(t)$ with its i-th entry representing the total current flowing out of node i through resistive devices, C is the constant capacitance matrix, and $\mathbf{I}_s(t)$ is the vector of inputs.

The problem of mixed signal simulation consists of simulating analog and digital subcircuits present in one design. As mentioned earlier, this involves using algorithms and techniques that trade-off the requirements of high performance for digital subcircuits and high accuracy for analog subcircuits. In the rest of the section, we describe some of the available techniques at our disposal, including slow but accurate direct methods and other approaches using simplified numerical algorithms/device models to gain speed at a slight compromise of accuracy.

2.2 Direct Methods

The direct methods use traditional algorithms to solve nonlinear systems of equations. The steps involved in transient analysis using the direct methods are: construct the circuit matrices using representations like the Modified Nodal Approach [13]; numerical integration of the differential equations by a stiffly stable method [11][20][28]; linearization of the algebraic equations by Newton-Raphson iterations; and sparse Gaussian elimination/LU decomposition to solve the system of linear equations.

Two of the earliest and most widely used circuit simulators SPICE [21], and ASTAP [37] use direct methods. These methods are accurate, stiffly stable but slow (solution time for an $n \times n$ matrix tends to grow as n^{β}, where $1.1 < \beta < 1.5$). Circuit simulators exploiting this time sparsity or latency by using device-level [21] or block-level bypass [27][36][40] schemes have been implemented to reduce the runtimes. For practical purposes through, direct methods based simulators cannot handle large industrial circuits.

All different algorithms proposed to improve the efficiency of circuit simulation are broadly based on two approaches: simplify the numerical algorithms and simplify the device models.

2.3 Simplified Numerical Algorithms Based Methods

Several methods modify the direct approach of SPICE to avoid the large number of Newton-Raphson iterations, to maximize time steps used, or to exploit the circuit's latency and multirate behavior.

There have been several works using relaxation methods [23] at different levels of the solution process (SPLICE [17][22], RELAX2 [18]). The efficiency of these approaches is determined by the speed of convergence, which heavily depends on the coupling between nodes and their ordering during solving. Thus, they can be very slow for tightly coupled circuits.

The semi-implicit integration method (MOTIS [5][9]) avoids lengthy iteration process. However, accuracy considerations usually constrain the time-step to be small, which degrades the efficiency.

The exponential integration approach (XPsim [2]) represents node voltage as piecewise exponential. While this allows larger time-steps, it is neither absolutely nor stiffly stable.

Asymptotic Waveform Evaluation method (AWEsim [14]) provides a generalized approach to linear RLC circuit response approximations. The transient response is approximated by matching the initial boundary conditions and the first $2q$-1 moments of the exact response to a lower order q-pole model. The efficiency of this approach is however impaired if the circuit has nonlinear elements.

ILLIADS [30] proposes an approach based on a generic circuit primitive and the exact analytic solution of its nonlinear state equation, *i.e.* the Ricatti equation. ILLIADS has been used to simulate large MOS circuits but cannot handle arbitrary circuits with bipolar elements or lossy lines.

2.4 Simplified Device Models Based Methods

Several approaches using simple device models have been proposed to speed-up the simulation by avoiding time consuming Newton-Raphson iterations and the model evaluation of nonlinear devices.

The effective linear conductance model has been used in timing analyzers Crystal [24], TV [15], and Rsim [32]. This type of analysis cannot handle analog waveforms. Also, there is no rigorous way of determining the effective conductance, and thus high accuracy cannot be guaranteed.

PLATO [31] uses piecewise linear segments to approximate the *i-v* curve of each nonlinear device. The Katzenelson algorithm [16] can be used for numerical integration. This becomes inefficient if many piecewise linear devices are present in the circuit.

SPECS2 [38] uses stepwise constant functions to model *i-v* characteristic of nonlinear devices. While this approach is efficient, it cannot be applied to circuits with floating capacitors or inductors.

Stepwise Equivalence Conductance approach [19] exploits certain characteristics of MOS circuits to speed-up simulation. This approach is based on the use of a stepwise equivalent model of a nonlinear resistive device. When applying the integration to digital MOS circuits, an additional speed-up in the simulation is achieved by taking advantage of the fact that the voltage waveforms can be modeled to a good approximation as piecewise linear functions. We use this approach in SYMPHONY since it provides good speed-ups while maintaining high accuracy and guaranteeing consistency, stability, convergency. In the following, we present a brief review of this approach.

From EQ (1), the equations for the circuit being simulated are

$$\mathcal{F}(\mathbf{V}(t)) + \mathbf{C}\dot{\mathbf{V}}(t) = \mathbf{I}_S(t) \tag{2}$$

This nonlinear system of equations can be transformed to a linear time variant system below without any loss of accuracy.

$$G(t)\mathbf{V}(t) + C\dot{\mathbf{V}}(t) = \mathbf{I}_S(t) \tag{3}$$

$G(t)$ represents the instantaneous equivalent conductance matrix for every branch in the circuit at time t, with $G(t)V(t) = \mathcal{F}(V(t))$ at every time instant t. $G(t)$ can be expanded in a Taylor series around $t{=}t_n$. Retaining only the first two terms, we obtain

$$[G(t_n) + \dot{G}(t_n)\,(t - t_n)]\,\mathbf{V}(t) + \mathbf{C}\dot{\mathbf{V}}(t) = \mathbf{I}_S(t) \tag{4}$$

This can then be further approximated as,

$$\mathcal{G}\mathbf{V}(t) + \mathbf{C}\dot{\mathbf{V}}(t) = \mathbf{I}_{\mathbf{S}}(t) \tag{5}$$

where, for $h_n = t_{n+1} - t_n$,

$$\mathcal{G} = \mathbf{G}(t_n) + \dot{\mathbf{G}}(t_n)\frac{h_n}{2} \tag{6}$$

The system of EQ (5) can be solved using sparse Gaussian elimination or LU decomposition, and no Newton-Raphson iterations are needed.

3 SYMPHONY: A Mixed Signal Circuit Simulator

SYMPHONY is a mixed signal simulator which combines a fast simulation engine for digital circuits with a traditional nonlinear solver *a la* SPICE for the analog subcircuits. The digital simulator uses Stepwise Equivalence Conductance to model nonlinear device conductances and Piecewise Linear voltage waveforms. The pure vanilla simulator is implemented in an event-driven framework with local and global clocks for event management. In later sections we describe some techniques implemented in SYMPHONY to achieve additional speed-up in simulation.

3.1 Simulator Framework

SYMPHONY uses circuit partitioning to identify analog and digital subcircuits in the circuit to be simulated. Digital subcircuits are handled by a fast Stepwise Equivalence Conductance based simulator and the analog subcircuits by a traditional Newton-Raphson iterations based engine.

From Section 2, during transient simulation, the stepwise equivalent conductance approach can be used to transform the system of nonlinear circuit equation of EQ (2) in to a linear time variant system below EQ (5) without any loss of accuracy.

$$G(t)\mathbf{V}(t) + \mathbf{C}\dot{\mathbf{V}}(t) = \mathbf{I}_S(\mathrm{t}) \tag{7}$$

G(*t*) represents the instantaneous equivalent conductance matrix for every branch in the circuit at time *t* with, $G(t)\,\mathrm{V}(t) = \mathcal{F}(\mathrm{V}(t))$ for every time instant *t*.

Since *G*(*t*) = $\mathcal{F}$(V(*t*))/V(*t*), clearly the effective conductance approximation is only suitable for devices whose conductances are not very sensitive to voltage changes. This dictates the time-step selection based on the control of the LTE to less than ε in EQ (8) [19].

$$\begin{gathered}\left(\left|\frac{\ddot{G}_i(t_n)\,h_n^2}{G_i(t_n)}\right|\right) \le \varepsilon \qquad \left(\left|\frac{\ddot{G}_i(t_n)\,h_n^2}{G_i(t_n)}\right|\right) \le \varepsilon \qquad \forall \text{ device } i \\ h_n\left|\dot{V}_j(t_n)\right| \le \Delta V \qquad \forall \text{ node } j\end{gathered} \tag{8}$$

Since mixed signal circuits may have bipolar elements (whose terminal currents are exponential functions of terminal voltages), the maximum allowable time-step can be very small, thus nullifying any speed-up gained by the use of the stepwise equivalent conductance model. Also, in the case of analog subcircuits, very high accuracy is required, which is best provided by traditional, exact simulation techniques.

Hence, for the analog subcircuits (which are obtained by circuit partitioning or user input), we use the traditional Newton-Raphson iteration based approach, briefly described in Section 2.2. Since this approach is well known [21], we do not go into its details here.

3.2 Circuit Partitioning

One of the keys to efficient mixed signal simulation is identifying the analog and digital circuit blocks in the design. This is best done automatically in order to preserve the generality and applicability of the simulator. For this reason, we propose the use of circuit partitioning techniques to separate analog and digital blocks in the circuit. Partitioning is used to identify the numerically sensitive regions containing bipolar elements so that they can be simulated by a Newton-Raphson based approach while the rest of the partitions can be simulated by a fast simulation engine.

The pure vanilla SYMPHONY performs a static partitioning based on the circuit structure at the start of the simulation. In case of MOS elements, the determination of the gate voltage need not depend on the voltage at the source or drain node, as long as there is no other charge transfer path connected between them. Also, if the gate voltage is evaluated prior to the evaluation of the drain and source node voltages, then the gate voltage can be treated as a constant voltage source in determining the source/drain voltages. If this ordering can be maintained during the simulation, solving for the voltage at the gate node can be separated from solving for the voltages at the drain and source nodes, and no iterations are needed between the two solving processes.

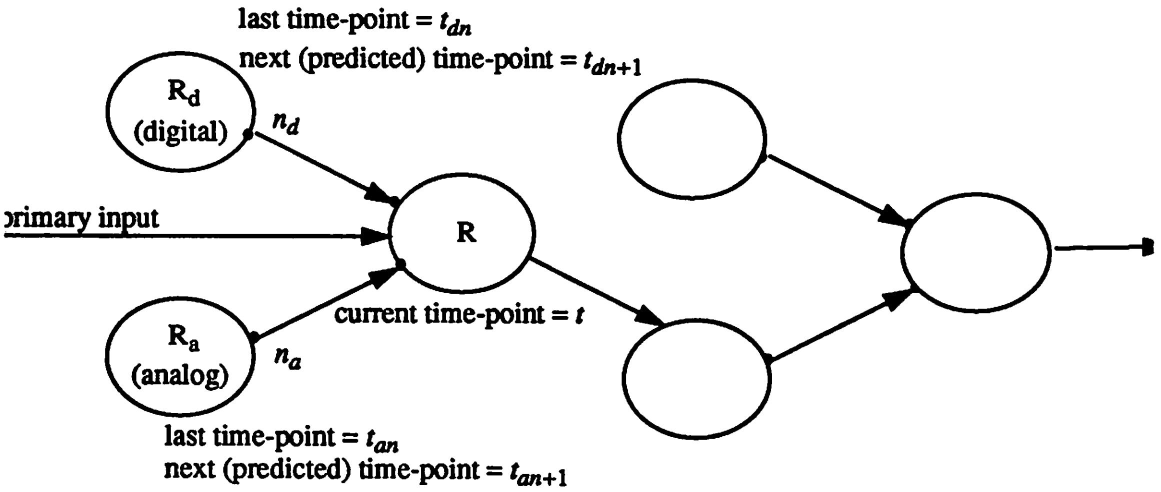
last time-point = t_{dn}
next (predicted) time-point = t_{dn+1}
R_d
(digital)
n_d
ɔrimary input
R
current time-point = t
R_a
(analog)
n_a
last time-point = t_{an}
next (predicted) time-point = t_{an+1}

Fig. 1 Event Scheduling

Thus, the circuit graph with an edges between each electrically connected node, can be transformed to an equivalent graph where gate-source and gate-drain edges corresponding to all MOS elements have been removed. The connected components in this graph are the regions whose simulation can be decoupled. Note that, local feedback is not a problem for tightly coupled nodes in the same region. It can be proved that this approach is valid in presence of global feedback also.

3.3 Event Management

SYMPHONY uses an event-driven mechanism to handle feedback loops in large circuits and further exploit circuit latency. Events are predicted, stored and scheduled in an event queue. The prediction is based on the input slopes and the conductances of MOS transistors given by EQ (8) for digital subcircuits and the standard, LTE based error control formula for analog subcircuits.

For a MOS device spanning two regions, the gate voltage is required to evaluate the regions containing the source/drain regions. This is obtained by curve-fitting based on the gate voltage slope. The same predictor is used for curve-fitting until a timepoint when the change in the gate voltage of a MOS exceeds a threshold. This timepoint is then a breakpoint of the gate voltage waveform and the regions containing the corresponding source/drain nodes have to be (re)scheduled for evaluation for that time. For safe event scheduling, we require that the gate voltage can be computed correctly any time the corresponding source/drain regions are evaluated.

Since the simulation of analog subcircuits is iteration based, the time step may have to be reduced and the event discarded if the simulation at a time point does not converge. Due to this back-tracking in time, we cannot use a predictor to predict (extrapolate) the voltage waveforms. Since interpolation is used to compute the nodal voltage at any time, in Fig. 1 the voltage at node n_a (in R_a) can be safely determined only up to time t_{an}.

In case of digital subcircuits, the time-step is determined such that the current predictor is guaranteed (by theory) to be valid (*i.e.* under the error margin allowed by LTE considerations [19]) until the next predicted event time. *i.e.* between the current event time and the next predicted event time, there cannot be another event. Thus, in Fig. 1, the voltage of node n_d (in R_d) can be consistently predicted (by curve-fitting) for any time up to t_{dn+1}.

In order to safely evaluate a region R at time t, we need the voltages and slopes of all inputs to the region at time t. For the input from a digital region, this requires that $t_{dn+1} > t$. For an input from an analog region, this requires that $t_{an} > t$.

Thus, the time counter (event time of the subcircuit under evaluation) does not increase monotonically. Each region has a local event queue and a local clock. The local clock represents the earliest time the region can be scheduled for evaluation. Each event is placed on the local queue in *wait* state. Once the local clock passes this event time, the event is labeled *active* and transferred to a global queue from which events are scheduled for evaluation in the increasing order of time. Each region also maintains information about its boundary node voltages at previous timepoints, since neighboring regions with slower local clocks may need this information as and when their local clocks advance. A global clock representing the earliest simulation time of all regions is also maintained. All nodes have valid voltage waveforms till this timepoint, and no waveform information earlier than this time need be maintained.

4 Dynamic Partitioning for Simulation of BiCMOS Circuits

4.1 Introduction and Motivation

Circuit partitioning to improve the speed-up simulation has been known for a while. The concept of channel connected components was introduced first in [3], and has been used in various forms since then. This approach is based on the weak coupling between the gate and drain/source of a MOSFET and the unidirectionality of signal propagation. Using this, a circuit can be partitioned *a priori* (statically) based on its structure, and as long as the ordering between the gate and the drain/source node can be maintained during simulation, the solution for the gate voltage can be separated from the drain/source voltage computation.

In case of bipolar circuits, there is a strong coupling between the base and the collector/emitter nodes. The unidirectionality of signal propagation also cannot be assured. Consequently, the circuit partitioning technique based on channel connected components is applicable to MOS circuits only and cannot be used to partition bipolar circuits.

There are several works in the area of dynamic partitioning for transient circuit simulation. [29] proposes a dynamic partitioning technique to partition the nonlinear equation solution phase for bipolar circuits. In [35], an explicit solution method is used to solve the linearized nodal equations, with separate dynamic partitioning management for MOS circuits. Recently, [41] proposed an approach to selective dynamic regionization for specific MOS memory structures. None of these works address the area of dynamic partitioning for arbitrary BiMOS circuits in an event-driven framework.

4.2 Dynamic Partitioning of BiMOS Circuits

A common feature of BiMOS designs is the number of predominantly MOS subcircuits with a few bipolar devices as output drivers etc. These subcircuits usually represent independent functional blocks in the design. Unfortunately, if only static partitioning is used during the simulation of such circuits, bipolar devices can act as a glue between two functionally independent regions, resulting in large regions containing many functional blocks. Since the partitioning is performed statically before the simulation, the circuit behavior under the input stimuli cannot be exploited to take advantage of the latency and multi-rate behavior between these functional blocks.

Dynamic partitioning can be used to speed-up the simulation in such cases by taking advantage of the special behavior of bipolar devices placed in digital MOS subcircuits. These bipolar devices toggle between the *on* and *off* state of the device, with little time spent in the transitions. When the bipolar device is off, there is no coupling between the base and the collector/emitter nodes. This can be exploited to decouple the solution for the base node from the solution for the collector/emitter nodes.

In order to perform the dynamic partitioning with minimum overhead, we use incremental dynamic partitioning, i.e. maintain dynamic partitions by examining the currently evaluated regions and test them to determine if those regions should be split or merged with others. Since the overhead of data management for dynamic partitioning can nullify any potential speed-up, we need to use efficient implementation techniques and data structures.

4.3 Dynamic Partitioning in SYMPHONY

SYMPHONY starts by initially partitioning the circuit at MOSFET gates (if any), and labels the partitions with BJTs as *analog*. For each *analog* subcircuit, a signal flow graph G is constructed by inserting a directed edge between nodes i and j if there is any coupling between nodes i and j in the circuit. This graph is used for further partitioning of the *analog* region.

At each timepoint based on the conductance of each bipolar device in the subcircuit, some of the edges may be *active* (if the bipolar device connecting the two nodes is *on*) or *inactive* (if the bipolar device is *off*). The signal flow graph G may have one or more disconnected components after the removal of all *inactive* edges. Each such disconnected component represents a *dynamic* partition of the analog subcircuit. Since there is no coupling between these partitions, each can be simulated independently with different time-points.

During the simulation process, a bipolar device can change states from *on* to *off* and vice versa. At each such change, the signal flow graph G has to be updated to reflect the deletion/addition of *active* edges. This can result in one dynamic partition being divided in one or more *child* partitions or two dynamic partitions being merged. The first case does not pose a problem since all *child* partitions have the same time-stamp as the original partition. Merging two partitions can however be a problem if both have a different time-stamp. In this case, both regions are synchronized by scheduling at the current simulation time.

5 Simulation of Digital Bipolar Circuits using Piecewise Linear Approximation

5.1 Introduction and Motivation

The Stepwise Equivalence Conductance method is a powerful technique for the simulation of digital MOS circuits. It relies on approximating device characteristics with time-varying linear segments at each instance of simulation. The accuracy of this approach depends on the simulation time interval for which such segments remain valid approximations of the device characteristics. Thus, for devices whose conductances are very sensitive to terminal voltage changes, the time-step is constrained to be very small in order to ensure that the current approximate segment remains valid for the entire time-step. In case of bipolar elements (whose terminal currents are exponential functions of terminal voltages), the maximum allowable time-step is very small. Thus, the Stepwise Equivalence Conductance approach is not very efficient for bipolar circuits.

We propose to approximate the *i-v* characteristics of bipolar devices by piecewise linear segments. Unlike stepwise equivalent conductance, this approximation is done statically, before the simulation. The number of segments used for the approximation is small (typically 3 to 4). Thus, the bipolar device can be replaced by time-varying linear resistors and current sources. The only constraint on the time-step now is that every time a device makes a transition from one segment to another, an event is generated.

5.2 Piecewise Linear Modeling

Obtaining a good approximation to the exponential device characteristics is the most critical issue in the suggested approach. While there has been a lot of work in the area of simulation via the PWL approximation [6][10][16][33][38][39], most of the works assume the existence of a PWL model of a nonlinear device, and just address the issue of simulating networks of PWL devices. We propose a formal method to obtain the breakpoints required for the piecewise linear approximation, such that the worst case approximation error is minimized.

Definition: Let π_n be the set of all polynomials of degree $\leq n$, and $p_n(x) \in \pi_n$ approximate a given function $f(x)$ uniformly well on some interval $a \leq x \leq b$. The error in the approximation of $p_n(x)$ to $f(x)$ is measured by the norm

$$\|f-p\|_\infty = \max_{a \le x \le b} |f(x) - p(x)| \tag{9}$$

Ideally, we would want a best uniform approximation from π_n, that is a polynomial $\hat{p}_n(x)$ of degree $\le n$ for which

$$\|f-\hat{p}_n\|_\infty = \min_{p \in \pi_n} \|f-p\|_\infty \tag{10}$$

We denote the number $\|f-\hat{p}_n\|_\infty$ by $\text{dist}_\infty(f, \pi_n)$ and call it the *uniform distance* on the interval $a \le x \le b$ of f from polynomials of degree $\le n$.

$$\text{dist}_\infty(f, \pi_n) = \min_{p \in \pi_n} \|f-p\|_\infty \tag{11}$$

Theorem 1 : *A function f which is continuous on $a \le x \le b$ has exactly one best uniform approximation on $a \le x \le b$ from π_n. The polynomial $p \in \pi_n$ is the best uniform approximation to f on $a \le x \le b$ if and only if there are n+2 points $a \le x_0 \le \ldots \le x_{n+1} \le b$ so that*

$$(-1)^i [f(x_i) - p(x_i)] = \varepsilon \|f-p\|_\infty \qquad i = 0, \ldots, n+1 \tag{12}$$

with ε = signum[f(x_0)-p(x_0)].

A proof of this theorem can be found in [25]. The construction of a best uniform approximation from π_n is, in general, a nontrivial task. By proper *interpolation, almost best* approximations to nonlinear functions can be obtained with much less computation.

Theorem 2 : *Let $p_n(x) \in \pi_n$, interpolate f(x) at the points $x_0 < x_1 < \ldots < x_n$ in the interval $a \le x \le b$ of interest. Then*

$$\text{dist}_\infty(f, \pi_n) \le \|f-p_n\|_\infty \le (1 + \|\Lambda_n\|_\infty)\, \text{dist}_\infty(f, \pi_n) \tag{13}$$

where $\|\Lambda_n\|_\infty$ is the uniform norm of the Lebesgue function $\Lambda_n(x)$ given by,

$$\Lambda_n(x) = \sum_{i=0}^{n} \left| \prod_{j=0, j \ne i}^{n} \frac{x - x_j}{x_i - x_j} \right| \tag{14}$$

A proof of this theorem can be found in [7]. It is thus desirable to choose the interpolation points $x_0, \ldots, x_n$ in $a \le x \le b$ such a way that $\|\Lambda_n\|_\infty$ be as small as possible. This is *almost* accomplished by the *expanded Chebyshev points* for the interval $a \le x \le b$, given by

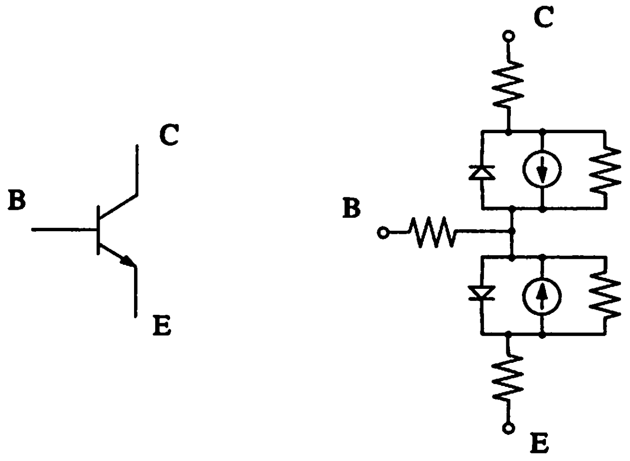

Figure 2. The static Ebers-Moll model, including second-order effects

$$x_i = \frac{1}{2}\left[a+b+(a-b)\,(\cos\frac{2i+1}{2n+2}\pi)\,/\,(\cos\frac{\pi}{2n+2})\right] \qquad i = 0,\ldots,n \qquad (15)$$

It can be shown that the $\left\|\Lambda_n^e\right\|_\infty$, corresponding to the *expanded Chebyshev points is* within 2% of the smallest possible value of $\left\|\Lambda_n^e\right\|_\infty$ for all n. From Theorem 6.2 and by computing the values of $\left\|\Lambda_n^e\right\|_\infty$, it can be shown that for $n \leq 47$, the error in the polynomial interpolating $f(x)$ at the *expanded Chebyshev points* is never bigger than 4 times the best possible error $\mathrm{dist}_\infty(f, \pi_n)$ (obtained by using the best uniform approximation polynomial $\hat{p}_n(x)$), and is normally smaller than that (by contrast, if Λ_n^u denotes the Lebesgue function for a *uniform* spacing of interpolation points, then $\left\|\Lambda_n^u\right\|_\infty \geq e^{n/2}$, which grows very rapidly with n).

We use the *expanded Chebyshev points* as break points for the piecewise linear approximation of the bipolar *i-v* characteristics. *i.e.* at these points, the piecewise linear approximation exactly matches the exponential *i-v* characteristics.

5.3 Simulation with PWL Device Models in the SYMPHONY Framework

From Section 2, the stepwise equivalent conductance approximation is only suitable for devices whose conductances are not very sensitive to voltage changes. Since the bipolar device models (e.g. the Ebers-Molls model in Fig. 2) include diodes, whose current is an exponential function of the terminal voltages, the stepwise equivalent conductance approximation cannot be applied to BiMOS circuits. One solution to this problem is to use the traditional Newton-Raphson iterations based solver for any subcircuit region containing bipolar devices. Unfortunately, this precludes us from exploiting the properties of digital MOS subcircuits, and can introduce avoidable speed-loss for digital subcircuits consisting predominantly of MOS devices with very few bipolar devices (which is the most typical case in digital BiMOS design).

SYMPHONY uses PWL models for the *i-v* characteristics of bipolar devices present in digital subcircuits. For digital subcircuits, after applying the stepwise equivalent conductance based transformation, the circuit equations are of the form (from EQ (5), EQ (6))

$$\mathcal{G}\mathbf{V}(t) + \mathbf{C}\dot{\mathbf{V}}(t) = \mathbf{I}_S(t) \tag{16}$$

where, for $h_n = t_{n+1} - t_n$,

$$\mathcal{G} = \mathbf{G}(t_n) + \dot{\mathbf{G}}(t_n)\frac{h_n}{2} \tag{17}$$

Since at every instant, each diode in the bipolar device model is still modeled by an effective conductance (which is a PWL function of the terminal voltages) the structure of the system of equations from EQ (5) remains unchanged.

We use standard techniques for the solution of PWL circuit networks. At every time-point t_{n+1}, the previous state of the bipolar device at t_n is used as an initial guess for the solution at t_{n+1}. If at the end of the solution process, we find that any device changed its state, we need to reduce the time-step and repeat the process. We iterate until we get a consistent solution (i.e. the initial guess of each device state is consistent with the computed device state).

The time-step selection is now dictated by two types of constraints: to ensure the validity of the stepwise equivalent conductance approximation of the nonlinear devices in the circuit, the time-step is controlled by the LTE criterion given by (from EQ (8))

$$\begin{gathered}\left(\left|\frac{\ddot{G}_i(t_n)\,h_n^2}{G_i(t_n)}\right|\right) \le \varepsilon \qquad \left(\left|\frac{\ddot{G}_i(t_n)\,h_n^2}{G_i(t_n)}\right|\right) \le \varepsilon \qquad \forall \text{ device } i \\ h_n\left|\dot{V}_j(t_n)\right| \le \Delta V \qquad \forall \text{ node } j\end{gathered} \tag{18}$$

In addition to this, the PWL approximation of bipolar device characteristics adds an additional constraint on the time-step. Specifically, the validity of each PWL approximation of a bipolar device also has to be maintained. This is achieved by enforcing that for the V_{be} and V_{bc} of each device the following is satisfied:

$$h_n \leq \begin{cases} \dfrac{V_{upper} - V(t_n)}{\dot{V}(t_n)} & \text{if } \dot{V}(t_n) > 0 \\ \dfrac{V_{lower} - V(t_n)}{\dot{V}(t_n)} & \text{if } \dot{V}(t_n) < 0 \end{cases} \tag{19}$$

Where V_{upper} and V_{lower} are the breakpoints of the current segment of the PWL approximation (or *current state*) of the base-emitter and base-collector diodes of a device. Clearly, if the $V(t_{n+1})$ obtained at the end of the computation lies outside this range, the initial assumption about the state of the corresponding diode is invalid. In this case, a new h_n is computed using these values of $V(t_{n+1})$ and $\dot{V}(t_{n+1})$, and the process repeated until a time-step is accepted by the above post-processing check on $V(t_{n+1})$ and $\dot{V}(t_{n+1})$.

The PWL approximation of bipolar devices can be effectively combined with dynamic partitioning by using the digital simulator with PWL models for subcircuits (previously classified as *analog*) containing only *off* bipolar devices (or devices with gate voltages less than some threshold) and the analog simulator for the rest.

6 Experimental Results

SYMPHONY was tested on a set of BiMOS benchmark circuits. The results were obtained on a DEC 5100/25 platform with a 24 Mbyte memory. The performance comparison of pure vanilla SYMPHONY against SPICE3e is presented in Table 1. It was found that SYMPHONY yields a reasonable speed-up while maintaining very high accuracy.

The amount of speed-up strongly depends on the mix of analog and digital blocks in the design under simulation. SYMPHONY yields ~3x speed-up for circuits with a high percentage of analog subcircuits like the BiCMOS Adder and Regfile while it is more than 20 times faster than SPICE3e for the Counter, a largely CMOS circuit. This is due to two reasons: the analog subcircuits are simulated by a slower simulation engine; and the presence of bipolar transistors results in larger partitions, since the static partitioning is performed across MOS elements only (this is an unavoidable penalty for performing automatic partitioning and can be improved by user provided partitioning).

Circuit	# nodes	SPICE3e cpu time (sec)	SYMPHONY cpu time (sec)
Adder	38	10.80	3.46
Regfile1	610	192.10	68.80
Counter	86	288.80	13.51

Table 1: Results on benchmark circuits

The performance comparison of the SYMPHONY version including dynamic partitioning, against SPICE3e is presented in Table 2, including the data from Table 1 on SYMPHONY without dynamic partitioning. The benchmarks set is same as in the previous case. It was found that SYMPHONY yields and additional 1.5x speed-up by using dynamic partitioning.

Circuit	# nodes	SPICE3e	SYMPHONY	
			No Dyn Part	w/ Dyn Part
Adder	38	10.80 s	3.46 s	2.13 s
Regfile	610	192.10 s	68.80 s	43.20 s

Table 2: Results on benchmark circuits

The PWL approach described above has been implemented in the digital simulator using stepwise equivalent conductance. We use SPICE3e along with this program to compare

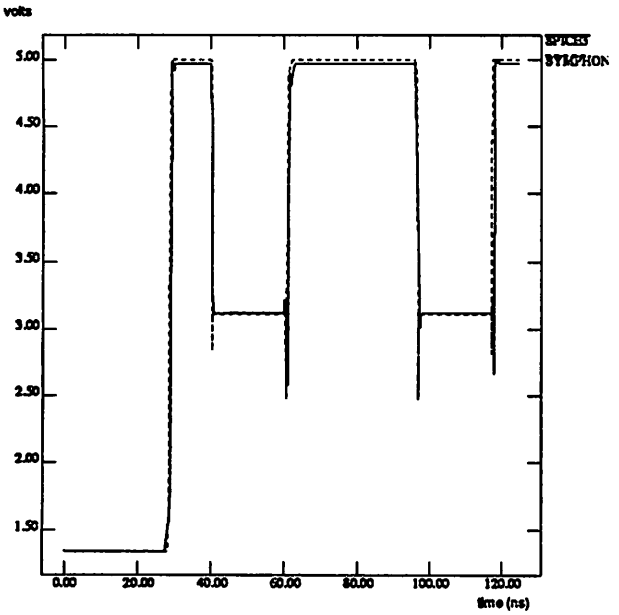

Fig.3 BiCMOS Adder: sum bit

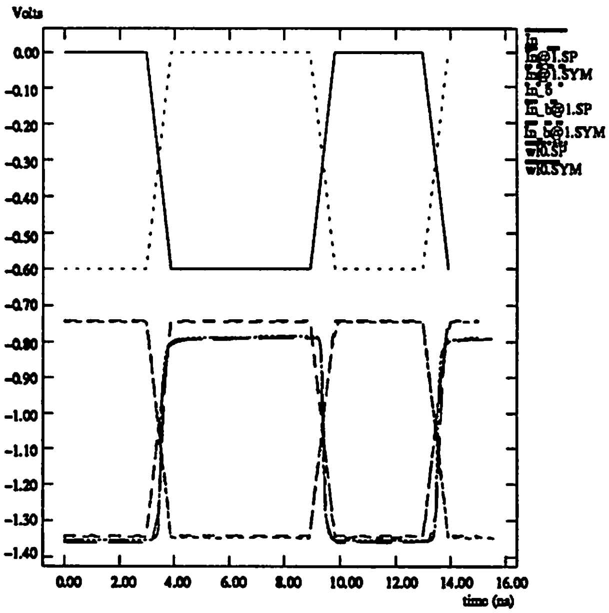

Fig. 4 BiCMOS Regfile: writelines

benchmark results. For the purpose of this benchmarking, we used MCNC benchmark circuits. Table 3 shows the runtime results for these circuits. (SPICE3e could not converge on the ring oscillator circuit during our benchmarking. MCNC documentation reports a runtime of 260 seconds for this circuit but does not specify the hardware platform on which this result was obtained).

Circuit	# nodes	SPICE3e	SYMPHONY
Bipolar Inverter	5	0.300	0.067
BiCMOS Inverter	8	0.800	0.300
bjtinv	37	4.200	2.100
bjtff	170	41.800	19.600
ring11	99	-*	38.433

Table 3: Benchmark Results (MCNC Circuits) (* did not converge)

Circuit	# nodes	SPICE3e	SYMPHONY		
			No Dyn Part	w/ Dyn Part	w/ Dyn Part, PWL
Adder	38	10.80 s	3.46 s	2.13 s	1.76 s
Regfile	610	192.10 s	68.80 s	43.20 s	28.6 s

Table 4: Results on benchmark circuits

Table 4 compares the final version of SYMPHONY including both PWL approximation of bipolar devices and dynamic partitioning against SPICE3e. Again, the benchmarks set is same as in the previous case. It is found that SYMPHONY can get an over all speed-up of ~ 6-7x over SPICE3e. The waveforms are compared in Figures 3 and 4, demonstrating the high accuracy of SYMPHONY

7 Conclusions

The increasing use of mixed signal circuits with analog and digital circuitry on the same chip have created a set of demands that traditional circuit simulators cannot meet very well. As far as we know, there are no efficient mixed signal simulators which can simulate mixed analog/digital designs with bipolar devices. In this work, we present SYMPHONY, a mixed signal simulator that fills this void using techniques specifically targeted at exploiting characteristics of such circuits.

SYMPHONY combines a fast simulator for digital circuits with a traditional nonlinear solver *a la* SPICE for the analog subcircuits. The digital simulator uses stepwise equivalent conductance to model nonlinear device conductances and achieves additional speed-up by modeling the voltages by piecewise linear waveforms.

Dynamic partitioning for BiMOS circuits in an event-driven framework, and a new PWL model for bipolar device characteristics are the two main contribution of this work. Dynamic circuit partitioning is used to fully exploit the latency and multirate behavior of the circuit. The PWL models allow us to speed-up simulation by using simple models for bipolar devices in digital parts of the design. These two techniques are combined to dynamically identify subcircuits with potentially analog or digital behavior. This eliminates the need of user-input to partition the circuit.

A set of benchmark results were presented on a suit of BiMOS circuits. We demonstrate that a speed-up of 6-7x can be obtained for mixed signal circuits with substantially analog components without compromising accuracy. Work on further improving the performance of SYMPHONY is under progress.

References

[1] E. L. Acuna, J. P. Dervenis, A. J. Pagones, F. L. Yang, and R. A. Saleh," Simulation Techniques for Mixed Analog/Digital Circuits," *IEEE Journal of Solid-State Circuits*, Vol. 25, No., pp. 353-363, April 1990.

[2] R. L. Baner, A. Ng, J. Fang, and R. K. Brayton, "XPsim: A MOS VLSI Simulator," *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 66-69, November 1988.

[3] R. E. Bryant, "A Switch-Level Model and Simulator for MOS Digital Systems," *IEEE Transactions on Computers*, Vol. C-33, No. 2, pp. 160-177, February 1984.

[4] Y.-H. Chang, and A. T. Yang, "Analytic Macromodeling and Simulation of Tightly Coupled Mixed Analog-Digital Circuits," *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 244-247, November 1992.

[5] B. R. Chawla, H. K. Gummel, and P. Kozak, "MOTIS - a MOS Timing Simulator," *IEEE Transactions on Circuits and Systems*, Vol. CAS-22, No. 12, pp. 901-909, December 1975.

[6] M. J. Chien, and E. S. Kuh, "Solving Nonlinear Resistive Networks using Piecewise Linear Analysis and Simplical Subdivision," *IEEE Transactions on Circuits and Systems*, Vol. CAS-24, No. 6, pp. 305-317, June 1977.

[7] S. Conte, and C. de Boor, *Elementary Numerical Analysis: An Algorithmic Approach*, McGraw-Hill Book Company, 1980.

[8] C. A. Desoer, and E. S. Kuh, *Basic Circuit Theory*, McGraw Hill, 1969.

[9] S. P. Fan, M. Y. Hsueh, A. R. Newton, and D. O. Pederson, "MOTIS-C: A New Circuit Simulator for MOS LSI Circuits," *Proceedings of the International Symposium on Circuits and Systems*, April 1977.

[10] T. Fujisawa, and E. S. Kuh, "Piecewise Linear Theory of Nonlinear Networks," *SIAM J. Appl. Math.*, Vol. 22, pp. 307-328, March 1972.

[11] C. W. Gear, "The Automatic Integration of Stiff Ordinary Differential Equations," *Proceedings of the IFIP Congress*, Edinburgh, Scotland, 1968.

[12] B. Hamman, and J.-L. Chen, "Data Point Selection for Piecewise Linear Curve Approximation," *Computer-Aided Geometric Design*, Vol. 11, No. 3, pp. 289-301, June 1994.

[13] C. W. Ho, A. E. Ruehli, and P. A. Brennan, "The Modified Nodal Approach to Network Analysis," *IEEE Transactions on Circuits and Systems*, Vol. CAS-22, No. 6, pp. 504-509, June 1975.

[14] X. Huang, V. Raghvan, and R. A. Rohrer, "AWEsim: A Program for Efficient Analysis of Linearized Circuits," *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 534-537, November 1990.

[15] N. P. Jouppi, "Timing Analysis and Performance Improvement of MOS VLSI Designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 6, No. 7, pp. 650-665, July 1987.

[16] J. Katzenelson, "An Algorithm for Solving Nonlinear Resistive Networks," *Bell System Technical Journal*, Vol. 44, No. 8, pp. 1605-1620, October 1965.

[17] J. Kleckner, R. Saleh, and A. R. Newton, "Electric Consistency in Schematic Simulation," *Proceedings of the IEEE International Conference on Circuits and Computers*, pp. 30-34, October 1982.

[18] E. Lelarasmee, and A. L. Sangiovanni-Vincentelli, "RELAX: A Circuit Simulator of Large Scale MOS Integrated Circuits," *Proceedings of the 19th ACM/IEEE Design Automation Conference*, June 1982.

[19] S. Lin, E. S. Kuh, and M. Marek-Sadowska, "Stepwise Equivalent Conductance Circuit Simulation Technique," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 12, No. 5, pp. 672-683, May 1993.

[20] W. Liniger, and R. A. Willoughby, "Efficient Integration Methods for Stiff Systems of Ordinary Differential Equations," *SIAM Journal of Numerical Analysis*, Vol. 7, pp. 47-66, March 1970.

[21] L. W. Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits," *Ph.D. thesis, University of California, Berkeley*, UCB/ERL M520, May 1975.

[22] A. R. Newton, "The Simulation of Large-Scale Integrated Circuits," *IEEE Transactions on Circuits and Systems*, Vol. CAS-26, pp.741-749, September 1979.

[23] A. R. Newton, and A. L. Sangiovanni-Vincentelli, "Relaxation-Based Electrical Simulation," *IEEE Transactions on Electronic Devices*, Vol. ED-30, No. 9, pp. 1184-1207, September 1983.

[24] J. K. Ousterhout, "A Switch-Level Timing Verifier for Digital MOS VLSI," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 4, No. 7, pp. 336-349, July 1985.

[25] J. R. Rice, *The Approximation to Functions*, Vol. 1, 2, Addison-Wesley, Reading, Mass., 1984.

[26] M. Rumsey, and J. Sackett, "An ASIC Methodology for Mixed Analog-Digital Simulation," *Proceedings of the 26th ACM/IEEE Design Automation Conference*, pp. 618-621, June 1989.

[27] K. Sakallah, and S. W. Director, "An Activity-Directed Circuit Simulation Algorithm," *Proceedings of the IEEE International Conference on Circuits and Computers*, pp. 1032-1035, October 1980.

[28] I. W. Sandberg, and H. Shichmann, "Numerical Integration of Stiff Nonlinear Differential Equations," *Bell System Technical Journal*, Vol. 47, No. 4, pp. 511-527, April 1968.

[29] P. Saviz, "Circuit Simulation by Hierarchical, Waveform Relaxation," *PhD. thesis, Columbia University*, 1990.

[30] Y.-H. Shih, Y. Leblebici, and S.-M. Kang, "ILLIADS: A Fast Timing and Reliability Simulator for Digital MOS Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 12, No. 9, pp. 1387-1402, September 1993.

[31] M. T. van Stiphout, J. T. J. van Eijndhoven, and H. W. Buurman, "PLATO: A New Piecewise Linear Simulation Tool," *Proceedings of the European Design Automation Conference*, pp. 235-239, March 1990.

[32] C. J. Terman, "Rsim - A Logic-Level Timing Simulator," *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 437-440, November 1983.

[33] S. Topcu, O. Ocah, A. Atalar, and M. A. Tan, "A Novel Algorithm for DC Analysis of Piecewise Linear Circuits: POPCORN," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol. 41, No. 8, pp. 553-556, August 1994.

[34] J. Varga, *Matrix Iterative Analysis*, Englewood Cliffs, NJ, Prentice-Hall, 1969.

[35] L. Vidigal, S. Nassif, and S. Director, "CINNAMON: Coupled Integration and Nodal Analysis of MOS Networks," *Proceedings of the 23rd ACM/IEEE Design Automation Conference*, pp. 79-185, 1986.

[36] A. Vladimirescu, and D. O. Pederson, "Performance Limits of the CLASSIE Circuit Simulation Program," *Proceedings of the International Symposium on Circuits and Systems*, May 1982.

[37] W. T. Weeks, A. J. Jiminez, G. W. Mahoney, D. Mehta, H. Qassemzadeh, and T. R. Scott, "Algorithms for ASTAP - A Network Analysis Program," *IEEE Transactions on Circuit Theory*, Vol. CT-20, No. 11, pp. 628-634, November 1973.

[38] C. Visweswariah, and R. A. Rohrer, "Piecewise Approximate Circuit Simulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 10, no. 7, pp. 861-870, July 1991.

[39] K. Yamamura, "Piecewise Linear Analysis of Nonlinear Resistive Networks Containing Gummel-Poon Models or Shichmann-Hodges Models," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E77-A, No. 1, pp. 309-316, January 1994

[40] P. Yang, I. N. Hajj, and T. N. Trick, "SLATE: A Circuit Simulation Program with Latency Exploitation and Node Tearing," *Proceedings of the IEEE International Conference on Circuits and Computers*,

October 1980.

[41] M. L. Yu, and B. D. Ackland, "VLSI Timing Simulation with Selective Dynamic Regionization," *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 195-199, November 1994.