CIRCUIT SIMULATION FOR MIXED-SIGNAL ANALOG/DIGITAL CIRCUITS

by

Premal Buch

Memorandum No. UCB/ERL M95/78

5 May 1995

CIRCUIT SIMULATION FOR MIXED-SIGNAL ANALOG/DIGITAL CIRCUITS

by

Premal Buch

Memorandum No. UCB/ERL M95/78

5 May 1995

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Circuit Simulation for Mixed-Signal Analog/Digital Circuits

Abstract

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. We present techniques to efficiently handle the simulation of mixed signal analog/digital circuits, and SYMPHONY, a fast mixed-signal simulator which embodies them.

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 Extended 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. The simulator is implemented in an event-driven framework with local and global clocks for even management. A set of benchmark results are presented on a suit of BiMOS circuits.

A transistor level power estimator which exploits algorithms for fast circuit simulation to compute the power dissipation of CMOS circuits is also presented. The proposed approach uses stepwise equivalent conductance and piecewise linear waveform approximation. The power estimator has been implemented in the SWEC framework. Experimental results indicate that SWEC can obtain a substantial speed-up over HSPICE (and handle circuits that HSPICE cannot) while maintaining an accuracy of within 5-7%. Benchmark results on a suite of industry circuits are presented.

Contents

Acknowledgments

I would like to thank my advisor Prof. Ernest Kuh for his constant support, guidance and encouragement during the course of this research. I would also like to thank Prof. Richard Newton for his feedback on this work. Thanks also to Dr. Shen Lin for introducing me to the SWEC project and Dr. Vijay Nagasamy and Prof. Qingjian Yu for guiding me on different parts of this work.

I was fortunate to have David Xue, Amit Narayan, Sunil Khatri, Renu Mehra, Rajeev Ranjan, Rajeev Murgai and Sriram Krishnan to share the ups and downs in my years here. I would also like to thank all the members of my research group - Charles Hough, Narasimha Bhat, Julie Hu, Dr. Kamal Chaudhary, Minshine Shih, Dr. Henrik Esbensen and Janet Wong.

This work was performed while the author was supported by the SRC under grant DC-324-018.

Chapter 1

Introduction

1.1 Motivation

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 [42][55]. Since the basic framework is still that of a slow simulator, this is not very efficient for large circuits which are mostly digital. Electrical-Logic simulation [27][28] and Analytic macromodeling [9] are some other efficient approaches 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.

In this work, we present SYMPHONY [6], a mixed signal circuit simulator. 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 Extended 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. The simulator is implemented in an event-driven framework with local and global clocks for even management. A set of benchmark results are presented on a suit of BiMOS circuits.

With the growing emphasis on low power design, there is a need for power estimation tools which provide feedback to the designer about the power performance of a design. While power estimation can be useful at all stages of the design process, transistor level power simulation is required to obtain accurate estimates of the power dissipation before going for the layout and implementation stage.

The most direct approach of obtaining the power dissipation of a circuit is to use

circuit simulators such as HSPICE [21] to simulate the design. While offering good accuracy, this approach suffers from a drawback of large runtimes. Furthermore, due to numerical accuracy issues, the standard approach to power simulation modifies the circuit description by adding powermeters (controlled current sources and parallel RC circuits) in order to monitor power waveforms without affecting the transient simulation [24].

We present a transistor level power estimator [7] which exploits algorithms for fast circuit simulation to compute the power dissipation of CMOS circuits. The proposed approach uses Stepwise Equivalent Conductance and Piecewise Linear Waveform approximation. The power estimator has been implemented in the SWEC framework. Experimental results indicate that SWEC can obtain a substantial speed-up over HSPICE (and handle circuits that HSPICE cannot) while maintaining an accuracy of within 5-7% in all cases in the benchmark suite of industry circuits.

1.2 Organization

This report is organized as follows:

In Chapter 2 we give the historical background and an overview of the circuit simulation problem. Chapter 3 describes the Stepwise Equivalence Conductance approach proposed in [31].

Chapter 4 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. Benchmark results for the pure vanilla SYMPHONY are presented.

Chapter 5 describes the dynamic partitioning techniques used to exploit the latency and multi-rate behavior of BiMOS circuits and its implementation in SYMPHONY. Benchmark results are presented for BiMOS circuits.

In Chapter 6, 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. Experimental results on a suite of digital bipolar circuits from MCNC benchmarks are presented.

Chapter 7 introduces the power estimation problem and the transistor level power simulation in context of low power design. A transistor level power estimator implemented in the SWEC framework is described. Results comparing the accuracy and speed of the proposed estimator with HSPICE are presented on a set of large industry circuits from LSI Logic Corporation.

Chapter 8 concludes with a summary of results and suggested directions of future work.

Chapter 2

Previous Work

2.1 Historical Background

The circuit simulation problem consists of solving a set of nonlinear algebraic equations of the form [15]

$$\mathcal{F}(\mathbf{V}(t)) + \mathbf{C}\dot{\mathbf{V}}(t) = \mathbf{I_S}(t) \qquad \text{(EQ 2.1)}$$

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, $\boldsymbol{C}$ is the constant capacitance matrix, and $\mathbf{I_s}(t)$ is a vector of inputs.

Two of the earliest and most widely used circuit simulators are SPICE, developed at UC Berkeley [34], and ASTAP, developed at IBM [54]. These simulators can be used to perform a wide variety of analyses, including DC, AC, transient, pole-zero, and noise analysis. Of these, transient analysis is one of the most commonly used for circuit design, and it is the most computationally expensive. SPICE and ASTAP use the straightforward traditional algorithms classified as *direct methods*, to solve nonlinear systems of equations for transient analysis. These methods are accurate, stiffly stable but slow.

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.

2.2 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 the following:

- Use of a circuit representation such as the Modified Nodal Approach [20] to construct the matrices and vectors describing the circuit behavior.
- Numerical integration of the differential equations by a stiffly stable integration method, such as the Backward Euler method [18][33][44].
- Linearization of the algebraic equations by Newton-Raphson iterations, and
- Sparse Gaussian elimination or LU decomposition to solve the system of linear equations.

As the circuit size increases, these methods become inefficient. Even when using sparse matrix techniques, solution time grows super-linearly with the problem size. Solution time for an $n \times n$ matrix tends to grow as n^{β}, where $1.1 < \beta < 1.5$. Also, since the differential equations describing the circuits are usually stiff, i.e. they contain widely separated eigenvalues corresponding to wide ranging time constants in the circuits, and since numerical integrations of all the equations is carried out simultaneously, the maximum allowable time-steps become very small. Circuit simulators exploiting this time sparsity or latency by using device-level [34] or block-level bypass [43][53][57] schemes have been implemented to reduce the runtimes. For practical purposes through, direct methods based simulators cannot handle large industrial circuits.

2.3 Simplified Numerical Algorithms Based Methods

The direct approach of SPICE has been modified to avoid the large number of Newton-Raphson iterations, to maximize time steps used, or to exploit circuit's latency and multirate behavior. These approaches include (1) the Relaxation approach, (2) the Waveform Relaxation approach, (3) the Semi-Implicit Integration approach, (4) the Exponential Integration, and (5) the Asymptotic Waveform Evaluation method.

Relaxation methods can be used for the solution of (EQ 2.1) in a number of ways [38], In all cases, their principal advantages stem from the fact that they do not require the direct solution of a large system of linear equations and from the fact that they permit the simulator to exploit latency efficiently. The two most common methods used in electric simulation are the Gauss-Jacobi method and the Gauss-Seidel method [51].

Linear relaxation methods replace the direct methods for the solution of the linear systems of equations (the last stage in the solution process in the Section 2.2) while non-linear relaxation methods use relaxation at the non-linear equation solution level to augment the Newton-Raphson method. The rate of convergence of the non-linear relaxation methods is only linear while it is quadratic for Newton-Raphson based direct methods. The gain in efficiency in relaxation methods is obtained due to the fact that each iteration of a relaxation method involves solving a set of decoupled non-linear equations, each in one unknown, while direct methods require the solution of a set of simultaneous non-linear equations. SPLICE [29][36] is an example of a simulator employing one of these classes of methods.

Waveform Relaxation approach applies relaxation techniques at the differential equation level. This is an analogue of the Gauss-Seidel technique for solving nonlinear algebraic equations. However, the unknowns here are waveforms (elements of a function space). Waveform Relaxation methods have been implemented in simulators like

RELAX2 [30].

The efficiency of the above approaches is determined by the speed of convergence, which heavily depends on the coupling between nodes. Thus, they can be very slow for tightly coupled circuits (*e.g.* circuits with strong negative feedbacks). Also, the ordering of nodes during the solving processes becomes very important. This can be handled optimally by using selective trace, in which the order in which node voltages are updated becomes a function of the signals flowing in the network. The timing error in this case can be restricted to one time-step [38].

To avoid being trapped in the lengthy iteration process, the semi-implicit integration method has been proposed and applied in the simulators like MOTIS [10][16]. These conjecture that there is a small enough time-step to obtain the exact solution in one Newton-Raphson iteration. The nonlinear devices of a circuit are linearized using the node voltages at the previous time-point and then the linearized circuit is integrated. However, to maintain the desired accuracy in most cases very small time steps need to be used, and that unfortunately degrades the efficiency.

The exponential integration approach represents node voltage as piecewise exponential. This approach has been implemented in XPsim [3]. It can be proved that this allows a larger time-step from the error control considerations. However, this approach is neither absolutely stable nor stiffly stable.

Asymptotic Waveform Evaluation method implemented in AWEsim [22] 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 non-linear elements exist in the circuit.

ILLIADS [46] proposes an approach based on a generic circuit primitive and the

exact analytic solution of its non-linear 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 of approximating the *i-v* characteristics of non-linear devices by piecewise linear or stepwise constant curves have been proposed to speed-up the simulation. The increase of efficiency in these methods is due to avoiding time consuming Newton-Raphson iterations and avoiding the model evaluation of non-linear devices.

The effective linear conductance model has been used in timing analyzers Crystal [39], TV [23], and Rsim [48]. For the whole transition, these timing analyzers replace every MOS transistor by an effective conductor and use the RC-tree [41] approach to estimate the first order timing information of the analyzed circuit. This type of analysis cannot handle analog waveforms. Also, there is no mathematically rigorous way of determining the effective conductance, and thus high accuracy cannot be guaranteed.

DIANA [2] uses threshold functions and boolean-controlled network elements to macro-model circuit blocks and mix logic and timing level simulation. This approach can have convergence problems which are solved using heuristics. Electrical-Logic Simulation (ELogic) is a relaxation-based switch-level simulation technique implemented in ELOGIC [28], which solves for the time required for a node to make a certain voltage change rather than solving for a node voltage at a given time-point. Since the number and/or values of the voltage states that can be specified may vary, ELOGIC provides for a continuous accuracy-speed trade-off, using three ELogic algorithms, ELogic-1, ELogic-2, and ELogic-3. This techniques is in general quite accurate and fast. Since discrete voltage states are used, there may be self-oscillations when the exact solution of a node voltage is

not sufficiently close to one of the voltage states. A similar output oscillation called inter-active-oscillation, may occur due to the interaction of a node and its fanin nodes. This can cost unnecessary CPU time and sometimes degrades the waveform accuracy [27].

The Newton-Raphson iterations can be avoided if we approximate the *i-v* curve of each non-linear device by piecewise linear segments. The Katzenelson algorithm [25] has been proposed to perform the piecewise linear version of numerical integration. Based on the state of the non-linear device, appropriate piecewise linear segment can be used to for each device to write the linear system of circuit equations at each time-point. Based on the Katzenelson algorithm, if a device changes state at a time-point, than we need to iterate using the updated circuit equations until we converge. Because of this, this algorithm can lose its efficiency if many devices with piecewise linear characteristics are present. Also, this approach is no longer consistent because the local truncation error will not vanish even when a very small integration time-step is used. The piecewise linear simulator PLATO [47] was implemented using this approach.

Another approach to simplify the device models is to use stepwise constant functions to approximate *i-v* characteristic of each device. This was implemented in SPECS2 [55]. While this approach is efficient (since the currents are constant, no linear/non-linear system of equations nee be solved), it cannot be applied to circuits with floating capacitors or inductors, which is a very strong restriction.

SWEC [31] uses a stepwise equivalent conductance model to perform fast, accurate transient simulation of large MOS circuits. We describe SWEC in detail in Chapter 3.

Chapter 3

The Stepwise Equivalent Conductance Approach

Stepwise Equivalence Conductance approach was first proposed in [31] to exploit 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. This technique is consistent, absolutely stable and convergent. 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.

3.1 Circuit Transformation

The Stepwise Equivalence Conductance circuit simulation makes the following two assumptions regarding the simulated circuits:

- Every node in the circuit has a nonzero capacitance to ground. This assumption does not place any restriction on the simulated circuits because practically every

node in a real circuit has a parasitic capacitance to ground.

- Every nonlinear device in the circuit has a unique current path. Examples of these kinds of devices are MOS, JFET, and diodes.

Assuming for the sake of simplicity, that there are no inductors and only constant capacitors in the simulated circuit, the KCL nodal equations for the simulated circuit will be of the form

$$\mathcal{F}(\mathbf{V}(t)) + \mathbf{C}\dot{\mathbf{V}}(t) = \mathbf{I_S}(t) \qquad \text{(EQ 3.1)}$$

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, $\boldsymbol{C}$ is the constant capacitance matrix, and $\mathbf{I_s}(t)$ is a vector of inputs. Since every node is assumed to have nonzero grounded capacitance, $\boldsymbol{C}$ is diagonally dominant. If $\mathcal{F}$ is nonlinear, then the implicit integration of (EQ 3.1) for each time-step involves solving a system of nonlinear equations. Computationally expensive Newton-Raphson iterations are generally needed to find the solutions.

The unique path assumption of nonlinear devices implies that the simulated circuit can be treated as an equivalent circuit with two-terminal resistive elements only. Specifically, the i-v characteristic of every nonlinear device at each time point can be characterized by its instantaneous equivalent conductance $G(t)$ defined as the ration of I and V across the two terminals of the current path evaluated at the time instant[1].

Then, (EQ 1) can be transformed to a linear time variant system below without any loss of accuracy.

$$\boldsymbol{G}(t)\,\mathbf{V}(t) + \mathbf{C}\dot{\mathbf{V}}(t) = \mathbf{I_S}(t) \qquad \text{(EQ 3.2)}$$

$\boldsymbol{G}(t)$ represents the instantaneous equivalent conductance matrix for every branch in

1. $G(t)$ is set to zero if $V(t) = 0$. The situation where $I(t) \neq 0$ when $V(t) = 0$ is practically impossible.

the circuit at time t. $\boldsymbol{G}(t)$ will satisfy the following relation

$$\boldsymbol{G}(t)\,\mathbf{V}(t) = \mathcal{F}(\mathbf{V}(t)) \qquad \text{(EQ 3.3)}$$

for every time instant t.

Instead of solving for $\mathbf{V}(t)$ of (EQ 1) directly, the Stepwise Equivalent Conductance circuit simulation solves for the $\mathbf{V}(t)$ of (EQ 2) and uses it as a solution for (EQ 1). An efficient integration scheme for (EQ 2) is used and no nonlinear equations need be solved.

The sufficient condition for the uniqueness of the solution of (EQ 1) is that $\dot{\mathbf{V}}$ is Lipschitz continuous at time t. If $\boldsymbol{G}(t)$ is known beforehand, than the uniqueness of the solution of (EQ 1) implies that the solution of (EQ 2) will be the same as (EQ 1). However, during the process of solving (EQ 2), for every time t, the function $\boldsymbol{G}$ is known only up to t. The following can be proved [32] about the solution of (EQ 2) in spite of this lack of information.

Theorem 3.1 : *If $\mathcal{F}(.)$ and $\boldsymbol{I}_s(.)$ of* (EQ 1) *are continuously differentiable then the solution of* (EQ 2) *will be exactly the same as that of* (EQ 1).

According Theorem 3.1, although the function $\boldsymbol{G}$ is not known after the current time t, the time derivatives of $\boldsymbol{G}$ at t up to the infinite order are known and thus it is possible to uniquely determine $\boldsymbol{G}$ for a small interval beyond t. This can be in turn used to determine $\mathbf{V}$ for that small interval. The sufficient condition of Theorem 3.1 is that $\mathcal{F}$ of (EQ 1) be continuously differentiable, which is rather restrictive because it excludes piecewise continuous i-v characteristics. To relax this restriction for the purpose of the numerical integration

- small time steps are used only when any piecewise characteristic in the circuit undergoes two different operation regions, and

- an absolutely stable integration scheme is employed.

Then, even though the sufficient condition is not satisfied strictly, the numerical solution of (EQ 2) can still yield very good accuracy.

3.2 The Stepwise Equivalent Conductance Integration Algorithm

For the integration of each time step, it is assumed that the equivalent conductances of the time-varying conductors remain constant during the time step. Therefore, for the calculation purpose, the circuits consists of only linear constant elements. The constant value assumed for each time-varying conductor can be determined to yield the necessary accuracy.

$G(t)$ can be expanded in a Taylor series around $t = t_n$. Retaining only the first two terms,

$$\left[G(t_n) + \dot{G}(t_n)\,(t-t_n) + \frac{1}{2}\ddot{G}(t_n)\,(t-t_n)^2 + \ldots\right]\mathbf{V}(t) + \mathbf{C}\dot{\mathbf{V}}(t) = \mathbf{I}_S(t) \quad \text{(EQ 3.4)}$$

This can then be further approximated as,

$$\mathcal{G}\mathbf{V}(t) + \mathbf{C}\dot{\mathbf{V}}(t) = \mathbf{I}_S(t) \quad \text{(EQ 3.5)}$$

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

$$\mathcal{G} = \mathbf{G}(t_n) + \dot{\mathbf{G}}(t_n)\frac{h_n}{2} \quad \text{(EQ 3.6)}$$

To solve for $\mathbf{V}(t_{n+1})$ from a given $\mathbf{V}(t_n)$ an error is introduced which is proven in [32] to be

$$-C^{-1}\ddot{G}(t_n)\,\mathbf{V}(t_n)\left(\frac{h_n^3}{6}\right) - C^{-1}\dot{G}(t_n)\,(C^{-1}G(t_n)\,\mathbf{V}(t_n) + \dot{\mathbf{V}}(t_n))\left(\frac{h_n^3}{12}\right) \quad \text{(EQ 3.7)}$$

By using the trapezoidal rule of integration,

$$\mathbf{V_{n+1}} = \left[\mathcal{G} + \frac{2}{h_n}C\right]^{-1}\left(\frac{2}{h_n}C\mathbf{V_n} + C\dot{\mathbf{V}}_\mathbf{n} + \mathbf{I_{s_{n+1}}}\right) \qquad \text{(EQ 3.8)}$$

This lead to the total local truncation error for the integration from t_n to t_{n+1} of the order $O(h^3)$. The method is therefore consistent with respect to the local truncation error, and since the integration scheme, trapezoidal rule, is absolutely stable, we have demonstrated the convergence of the algorithm.

3.3 Time Step Selection

The local truncation error for each integration will be equal to the error given in (EQ 3.7) plus the error introduced from the trapezoidal rule approximation of $\dot{\mathbf{V}}(t_{n+1})$. Therefore, given the error criterion on the local truncation error at t_n, the necessary time step h_n can be determined exactly. A variable time step integration can be implemented. However, (EQ 3.7) is very complicated. Determining h_n involved several matrix operations. It would be impractical to perform the matrix operations at every time point. Therefore, a simpler scheme of using h_n is used. By using two parameters, a voltage error ΔV and a relative error ε, the following can be derived: For each conductance G_i and for each node voltage V_j, if h_n meets the constraints imposed on (EQ 3.9), then the norm of the error introduced in (EQ 3.7) will be less than $\frac{\varepsilon}{3}\Delta V$ [32].

$$\left(\left|\frac{\ddot{G}_i(t_n)\,h_n^2}{G_i(t_n)}\right|\right) \le \varepsilon \qquad \left(\left|\frac{\ddot{G}_i(t_n)\,h_n^2}{G_i(t_n)}\right|\right) \le \varepsilon \qquad \forall \text{ device } i$$
$$h_n\left|\dot{V}_j(t_n)\right| \le \Delta V \qquad \forall \text{ node } j \qquad \text{(EQ 3.9)}$$

The advantage of this is that the computation of determining a time step meeting all the constraints in (EQ 3.9) is linear in terms of nodes or devices in the circuity, while

the computation needed to solve (EQ 3.7) is of the cubic order.

SWEC [31] is a fast and efficient timing simulator for digital CMOS VLSI circuits, which has been implemented based on the concepts above. To further speed up the simulation, SWEC first decomposes the circuit into weakly coupled subcircuits and applies the Stepwise Equivalent Conductance technique to each of the subcircuits. In addition, SWEC exploits another special property of CMOS circuits, that is, the voltage waveform can be modeled with piecewise-linear segments connected between regions with smooth curves. Thus, the voltage waveforms of the outputs from the CMOS gates behave like straight line segments most of the time. Because of this property, larger time steps can be used. To handle feedback inside the circuits and to further exploit the latency and multirate behavior of MOS circuits, a special event driven mechanism based on the piecewise linearity of waveforms has also been developed and built inside SWEC.

Chapter 4

SYMPHONY: A Mixed Signal Circuit Simulator

SYMPHONY [6] 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 non-linear device conductances and PieceWise Linear voltage waveforms. The pure vanilla simulator is implemented in an event-driven framework with local and global clocks for even management. In later chapters we describe some other techniques implemented in SYMPHONY to achieve additional speed-up in simulation.

4.1 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.

During transient simulation, at each time-point the system of non-linear time-variant circuit equations is given by

$$\mathcal{F}(\mathbf{V}(t)) + \mathbf{C}\dot{\mathbf{V}}(t) = \mathbf{I_S}(t) \qquad \text{(EQ 4.1)}$$

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, $\boldsymbol{C}$ is the constant capacitance matrix, and $\mathbf{I_S}(t)$ is a vector of inputs.

From Chapter 3, the Stepwise Equivalent Conductance approach can be used transform the system of (EQ 4.1) in to a linear time variant system below without any loss of accuracy.

$$\boldsymbol{G}(t)\,\mathbf{V}(t) + \mathbf{C}\dot{\mathbf{V}}(t) = \mathbf{I_S}(t) \qquad \text{(EQ 4.2)}$$

$\boldsymbol{G}(t)$ represents the instantaneous equivalent conductance matrix for every branch in the circuit at time t. $\boldsymbol{G}(t)$ will satisfy the following relation

$$\boldsymbol{G}(t)\,\mathbf{V}(t) = \mathcal{F}(\mathbf{V}(t)) \qquad \text{(EQ 4.3)}$$

for every time instant t.

Since $\boldsymbol{G}(t) = \mathcal{F}(V(t))/V(t)$, 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 4.4) (reproduced from Chapter 3).

$$\left(\left|\frac{\ddot{G}_i(t_n)\,h_n^2}{G_i(t_n)}\right|\right) \le \varepsilon \qquad \left(\left|\frac{\ddot{G}_i(t_n)\,h_n^2}{G_i(t_n)}\right|\right) \le \varepsilon \qquad \forall \text{ device } i$$
$$h_n\left|\dot{V}_j(t_n)\right| \le \Delta V \qquad \forall \text{ node } j \qquad \text{(EQ 4.4)}$$

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. Since this approach is well known [34], we do not go into its details here.

4.2 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.

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.

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 [32] that this approach is valid in presence of global feedback also.

4.3 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. 6 for digital subcircuits and the LTE based error control formula for analog subcircuits.

For a MOS 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 need to be (re)scheduled for that time.

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. Thus, there cannot be any guarantee that no extra breakpoints will be introduced before the predicted next event. In Fig. 4.1 the voltage at node n_a (in R_a) can be determined only up to time t_{an}.

In case of digital subcircuits, there is no backtracking, i.e. between the current event time and the next predicted event time, there cannot be another event. Thus, in this

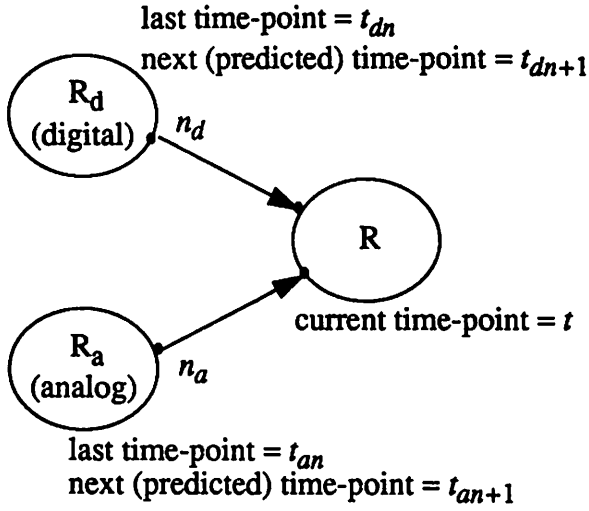

Fig. 4.1 Event Scheduling

case, it is guaranteed that no extra breakpoints will be introduced (causing rescheduling) before the predicted next event. e.g., in Fig. 4.1, 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 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 event time under evaluation does not increase monotonically. Each region has a local event queue and a local clock. The local clock represents the latest 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 wave-

forms till this timepoint, and no waveform information earlier than this time need be maintained.

4.4 Experimental Results

SYMPHONY was tested on a set of BiMOS benchmark circuits. The performance comparison of SYMPHONY against SPICE3 is presented in Table 1. It was found that SYMPHONY yields good 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 SPICE 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 instead of expecting the user to provide an aggressive partitioning scheme, and can be improved by user provided partitioning). The reported Spice3e runtimes are the best obtained after experimenting with various parameters relating to accuracy and convergence criteria [37][38].

Circuit	# nodes	SPICE3 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 4.1: Results on benchmark circuits

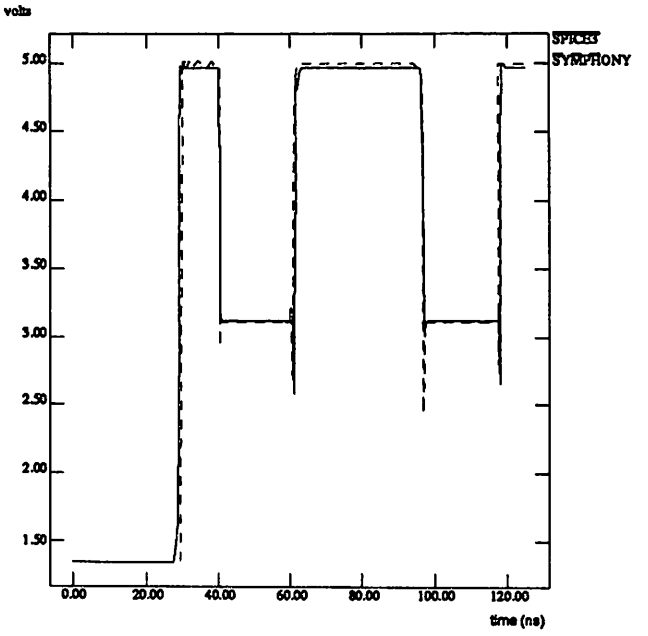

Fig. 4.2 BiCMOS Adder: sum bit

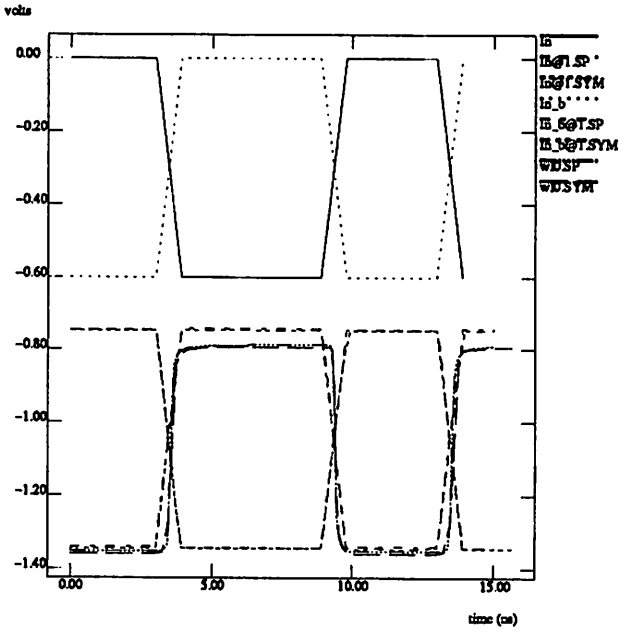

Fig. 4.3 BiCMOS Regfile1: writelines

Chapter 5

Dynamic Partitioning for Simulation of BiMOS Circuits

5.1 Introduction and Motivation

Circuit partitioning to improve the speed-up simulation has been known for a while. SPLICE [28][36] used selective trace for node ordering in a relaxation based framework. The concept of channel connected components was used in [5], and has been applied 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. It is our

aim to define a partitioning technique that exploits the properties of Bipolar circuits to speed-up the simulation.

There are several works in the area of dynamic partitioning for transient circuit simulation. The selective trace scheme of SPLICE [28][36], a relaxation based simulator, schedules nodes when any fanin node voltage changes. [4] proposes circuit partitioning for I^2L bipolar circuits relying on the fact that as in MOS, the reverse transmission through gates is generally sufficiently small to be neglected, thus allowing the base region to be analyzed independently of the collector/emitter regions. [45] proposes a dynamic partitioning technique to partition the non-linear equation solution phase for bipolar circuits. In [52], an explicit solution method is used to solve the linearized nodal equations, with separate dynamic partitioning management for MOS circuits. Recently, [58] proposed an approach to selective dynamic regionization for specific MOS memory structures. None of these works address the area of dynamic partitioning for BiMOS circuits in an event-driven framework.

5.2 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 tak-

ing 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 voltage 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.

5.3 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 timepoints.

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.

5.4 Experimental Results

SYMPHONY was tested on a set of BiMOS benchmark circuits. The performance comparison of SYMPHONY against SPICE3 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, with the same level of accuracy in the results. 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 5.1: Results on benchmark circuits

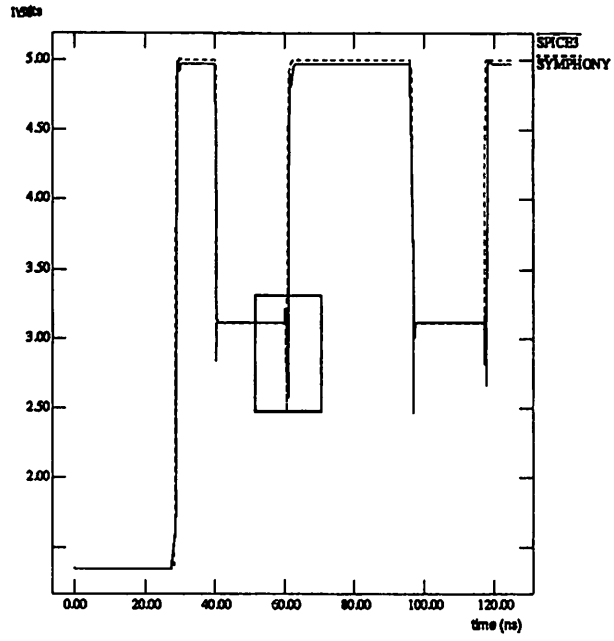

Fig. 5.1 BiCMOS Adder: sum bit

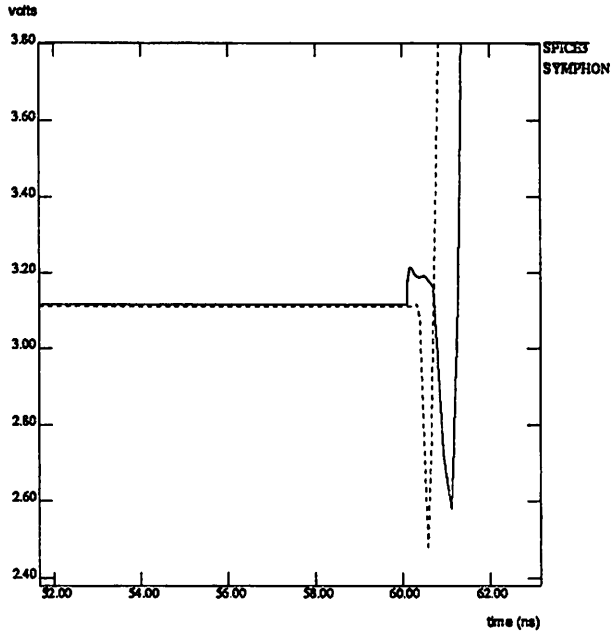

Fig. 5.2 BiCMOS Adder: sum bit
(zoom of worst discrepancy)

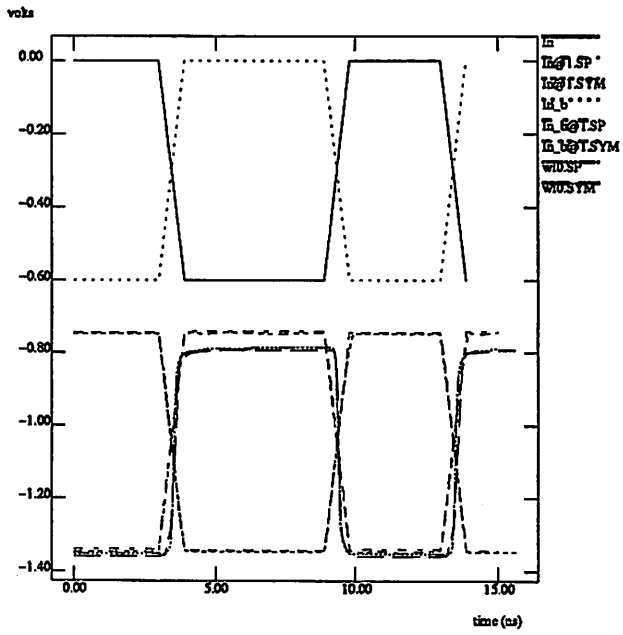

Fig. 5.3 BiCMOS Regfile1: writelines

Chapter 6

Simulation of Digital Bipolar Circuits using Piecewise Linear Approximation

6.1 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 the SWEC approach, 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. The main difference of this approach from the SWEC approach is that since the linear segments are generated a priori, and are much larger than the approximations generated by SWEC dynamically, the number of such events is much smaller, hence making this approach more efficient.

6.2 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 [11][17][25][49][55][56], 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

$$\|f-p\|_\infty = \max_{a \leq x \leq b} |f(x) - p(x)| \qquad \text{(EQ 6.1)}$$

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

$$\|f-\hat{p}_n\|_\infty = \min_{p \in \pi_n} \|f-p\|_\infty \qquad \text{(EQ 6.2)}$$

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 $\le n$.

$$\text{dist}_\infty(f, \pi_n) = \min_{p \in \pi_n} \|f-p\|_\infty \qquad \text{(EQ 6.3)}$$

Theorem 6.1 : *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*

$$(-1)^i [f(x_i) - p(x_i)] = \varepsilon \|f-p\|_\infty \qquad i = 0, \ldots, n+1 \qquad \text{(EQ 6.4)}$$

with ε = signum[$f(x_0)$-$p(x_0)$].

A proof of this theorem can be found in [40].

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 6.2 : *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*

$$\text{dist}_\infty(f, \pi_n) \le \|f-p_n\|_\infty \le (1 + \|\Lambda_n\|_\infty)\, \text{dist}_\infty(f, \pi_n) \qquad \text{(EQ 6.5)}$$

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

$$\Lambda_n(x) = \sum_{i=0}^{n} \left| \prod_{j=0, j \ne i}^{n} \frac{x-x_j}{x_i-x_j} \right| \qquad \text{(EQ 6.6)}$$

A proof of this theorem can be found in [13]. 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

$$x_i = \frac{1}{2}\left[a+b+(a-b)\,(\cos\frac{2i+1}{2n+2}\pi)/(\cos\frac{\pi}{2n+2})\right] \qquad i = 0,\ldots, n \qquad \text{(EQ 6.7)}$$

It can be shown that the $\|\Lambda_n^e\|_\infty$, corresponding to the *expanded Chebyshev points is* within 2% of the smallest possible value of $\|\Lambda_n\|_\infty$ for all n. From Theorem 6.2 and by computing the values of $\|\Lambda_n^e\|_\infty$, it can be shown that for $n \le 47$, the error in the polynomial interpolating f(x) at the *expanded Chebyshev points* is never bigger than 4 times the best possible error $\text{dist}_\infty(f, \pi_n)$ (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 $\|\Lambda_n^u\|_\infty \ge e^{n/2}$, which grows very rapidly with n).

We use the e*xpanded 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.

6.3 Simulation with PWL Device Models in the SYMPHONY Framework

From Chapter 3, 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 Figure 6.1) 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 prob-

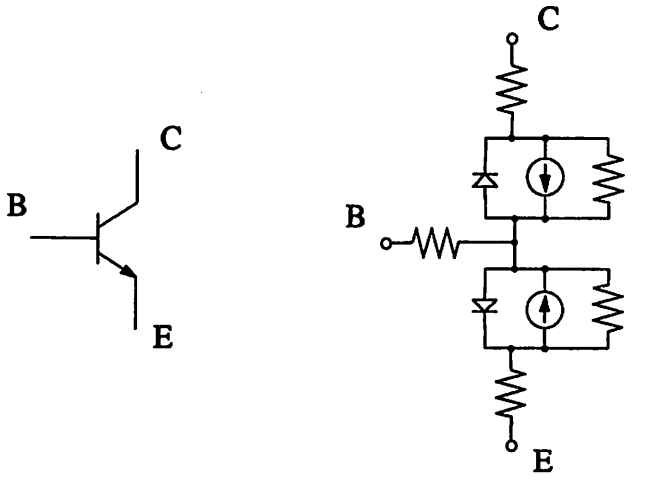

Figure 6.1. The static Ebers-Moll model, including second-order effects

lem 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 3.5, 3.6))

$$\mathcal{G}\mathbf{V}(t) + \mathbf{C}\dot{\mathbf{V}}(t) = \mathbf{I_S}(t) \qquad \text{(EQ 6.8)}$$

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

$$\mathcal{G} = \mathbf{G}(t_n) + \dot{\mathbf{G}}(t_n)\frac{h_n}{2} \qquad \text{(EQ 6.9)}$$

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 6.8) 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 3.9))

$$\left(\left|\frac{\ddot{G}_i(t_n)\,h_n^2}{G_i(t_n)}\right|\right) \le \varepsilon \qquad \left(\left|\frac{\ddot{G}_i(t_n)\,h_n^2}{G_i(t_n)}\right|\right) \le \varepsilon \qquad \forall \text{ device } i$$
$$h_n\left|\dot{V}_j(t_n)\right| \le \Delta V \qquad \forall \text{ node } j \tag{EQ 6.10}$$

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:

$$h_n \le \begin{cases} \dfrac{V_{upper} - V(t_n)}{\dot{V}(t_n)} & \text{if } \dot{V}(t_n) > 0 \\ \dfrac{V_{lower} - V(t_n)}{\dot{V}(t_n)} & \text{if } \dot{V}(t_n) < 0 \end{cases} \tag{EQ 6.11}$$

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 $\dot{V}(t_{n+1})$.

6.4 Experimental Results

The PWL approach described above has been implemented in the SWEC framework. We use SPICE3e along with this program to compare benchmark results.

For the purpose of this benchmarking, we used MCNC benchmark circuits. Table 1 shows the runtime results for these circuits. The results were obtained on a DEC 5100/25 platform with a 24 Mbyte memory. The second and third columns in the table correspond to the runtimes reported by SWEC and SPICE3e respectively (SPICE3e could not complete on the ring oscillator on our platform. MCNC documentation report a runtime of 260 seconds for this circuit but does not specify the hardware platform on which this result was obtained). Figures 1-5 show some of the node voltage waveforms for each of the benchmark circuit, demonstrating that except at sharp breakpoints, the PWL characteristic based program matches well with the SPICE3e output.

Circuit	# nodes	PWL	Spice3e
Bipolar Inverter	5	0.067	0.300
BiCMOS Inverter	8	0.300	0.800
bjtinv	37	2.100	4.200
bjtff	170	19.600	41.800
ring11	99	38.433	-*

Table 6.1: Benchmark Results (MCNC Circuits) (* could not converge)

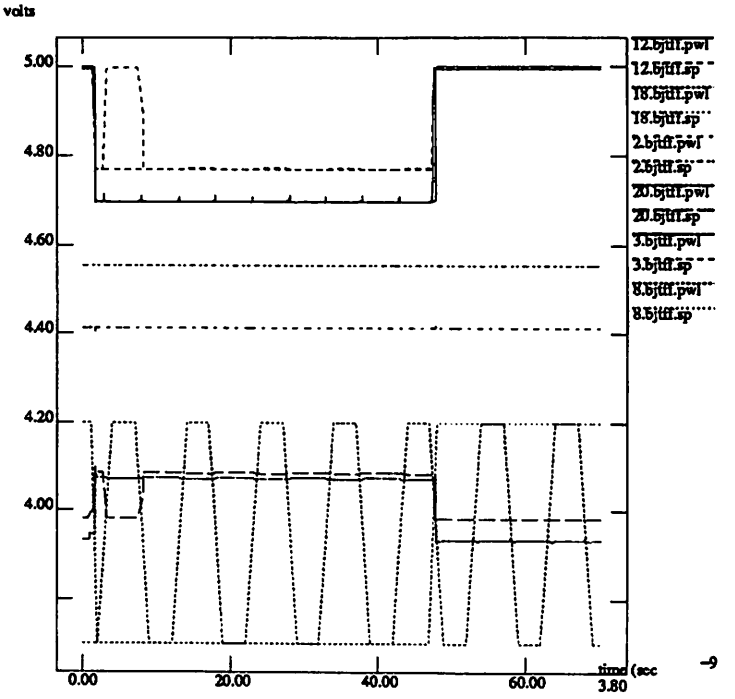

Figure 6.1. bjtff (MCNC Benchmark)

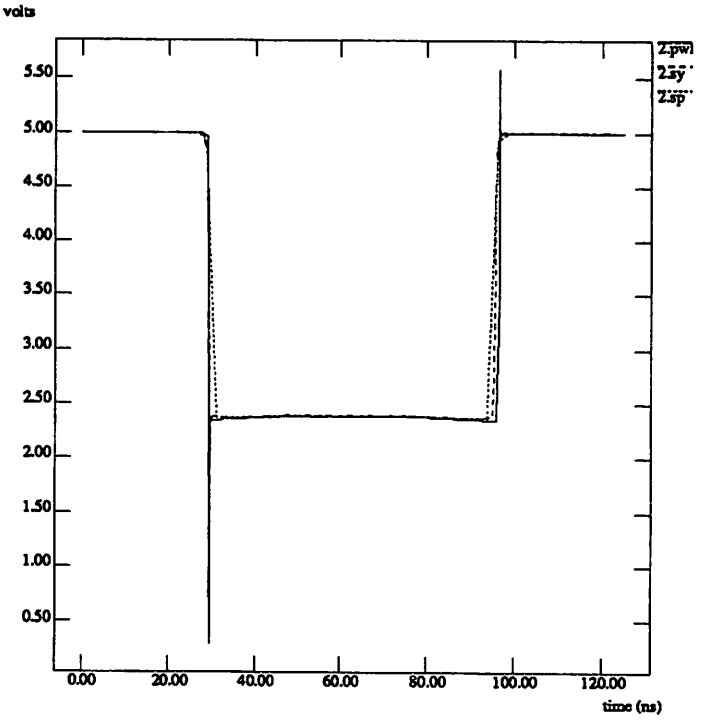

Figure 6.2. Bipolar Inverter

Chapter 7

Power Estimation of CMOS Circuits

7.1 Introduction and Motivation

Power minimization is becoming very important for a number of reasons ranging from an increasing demand for portable computing and telecommunication equipment, increasing clock frequencies, to advances in process technology that enables the integration of extremely large number of densely packed devices on a single chip. Minimizing power dissipation of chips has an impact not only on energy savings, but also helps create more reliable chips. Although designers have several techniques at their disposal to minimize power, there is little or no help in terms of tools to assist in analyzing and evaluating the effectiveness of various decisions during the design process.

A challenging problem in this context is how to efficiently obtain power estimates which meet the accuracy and the run-time constraints of the designer. Several approaches have been proposed to compute or estimate power dissipation, each with a different accuracy/run-time trade-off. These approaches can be classified into three broad categories: statistical/empirical techniques [29], probabilistic techniques [26][35][50], and circuit

simulation based techniques [12][14][24].

The main advantages of the probabilistic techniques is their short runtimes and input-independence. The probabilistic techniques use a stochastic model of logic signals of a circuit and propagate the probabilities of logic values through the combinational logic modules in order to compute the average switching rate of the circuit. This measure is in turn, is used to obtain the average power consumption of the circuit. It can potentially be accurate; however, for high accuracy, the spatial and temporal correlation between internal node values must be modeled. As this proves to be expensive, most approaches trade off accuracy for speed, resulting in highly unacceptable estimates at times.

Another approach is to make use of various statistical measures of the circuit. This approach is the most crude of all, despite its advantage in speed. It reads a description of the design, compiles various statistical measures, and calculates the power consumption based on these measures. The main use of this method is to obtain rough estimates of power dissipation at early stages of the design.

Circuit simulators such as HSPICE [21] still provide the most direct and accurate approach for computing power dissipation. While offering good accuracy, HSPICE suffers from a drawback of limited capacity and large run times. This makes HSPICE impractical for all but the smallest of circuits consisting of around a few hundred gates. The popular solution thus is to separately use point tools like Powermill [14] for power estimation and HSPICE for the simulation and verification of the circuit in parts.

In this work we propose the use of a circuit simulation tool SWEC, that alleviates some of the problems faced by tools such as HSPICE, thus making it realistically feasible to perform power estimation along with simulation of the entire design. SWEC uses Stepwise Equivalence Conductance [31] and piecewise linear waveform approximations for fast and accurate circuit simulation. These techniques prove to be very well suited for efficient power estimation as well. In the following sections, we describe the power estima-

tion problem, provide a brief description of SWEC, the power estimator implemented within the SWEC framework and present experimental results on industrial circuits, which show that this tool can be effectively used for power estimation of large CMOS circuits.

7.2 Problem Description

By power estimation we refer to the problem of estimating the average power dissipation of a circuit. In CMOS circuits, there are two components that contribute to power dissipation [8]: static dissipation (due to leakage current) and dynamic dissipation (due to switching transient current and charging and discharging of load capacitance).

In most CMOS ASICs the contribution due to static dissipation is small compared to dynamic dissipation. The static power dissipation P_s of a circuit is given by the equation:

$$P_s = \sum_{i}^{n} I_l \times V_{dd} \qquad \text{(EQ 7.1)}$$

where I_l is the leakage current of the device (gate), V_{dd} is the supply voltage and n is the number of devices in the circuits.

The dynamic power dissipation P_{di} for a logic gate is given by the equation:

$$P_{di} = \frac{1}{2} \cdot C_{Li} \cdot V_{dd}^2 \cdot \frac{N_{ti}}{T} \qquad \text{(EQ 7.2)}$$

where C_{Li} is the output load capacitance on the gate i, V_{dd} is the supply voltage, T is the clock cycle and N_{ti} is the number of switching transitions per clock cycle for gate i. The dynamic power dissipation P_d of a circuit with n gates is given by the summation:

$$P_d = \frac{V_{dd}^2}{2T} \cdot \sum_{i=1}^{n} C_{Li} \cdot N_{ti} \qquad \text{(EQ 7.3)}$$

The total power dissipated by the circuit is the sum of the two components; static and dynamic dissipation.

$$P_{total} = P_s + P_d \qquad \text{(EQ 7.4)}$$

The most accurate and straightforward approach to power estimation is by simulation: perform a circuit simulation of the design and monitor the current waveform. While this estimate accounts for all types of power dissipation, the main drawback of this approach is the very high runtime. At the transistor level, the problem thus reduces to performing efficient transient simulation of the circuits in a *power-estimation friendly* way, i.e. using simulation techniques that are also suited to computing power estimation with minimum overhead.

7.3 Power Estimation using SWEC

The Stepwise Equivalence Conductance and the piecewise linear waveform approximation are ideally suited for efficient power computation. The power can be measured directly by monitoring the conductance and the voltage waveform during each time-step. Using (5), The power dissipated in each device during a time step h_n (from t_n to t_{n+1}) is given by

$$P_d = \frac{1}{h_n} \int_{t_n}^{t_{n+1}} V(t)\mathcal{F}(V(t))\,dt \qquad \text{(EQ 7.5)}$$

Recall that, during each time-step, each nonlinear device conductance is approximated by an equivalent conductance $\mathcal{G}$ (Eq. 6-10). Thus, (11) can be simplified as:

$$P_d = \frac{1}{h_n} \int_{t_n}^{t_{n+1}} \mathcal{G} \cdot V^2(t)dt \qquad \text{(EQ 7.6)}$$

Since the voltage waveforms are piecewise linear, *dV/dt* is a constant for a given time step. Thus, the power dissipated in each device from t_n to t_{n+1} can be obtained by simply computing the area under the power waveform curve given by:

$$P_d = \frac{\mathcal{G}}{h_n}\int_0^{h_n}\left[V(t_n) + t \cdot \frac{dV}{dt}\Big|_{h_n}\right]^2 dt \qquad \text{(EQ 7.7)}$$

The power in capacitors can be computed similarly. Specifically, one can directly measure the power consumption during simulation using the piece-wise linearity property of the waveforms in SWEC. For each event during the course of a simulation, we perform the following calculations. Suppose an event changes the voltage across a capacitor C_i from v_0 at time t_0 to v_1 at time t_1. Then, the power dissipated from t_0 to t_1 is given by:

$$P_c = \frac{1}{h_n} \cdot C_i \cdot V_{avg} \cdot \frac{dV}{dt} \qquad \text{(EQ 7.8)}$$

where V_{avg} is the average value of v_0 and v_1. *dV/dt* is a constant as before. Inductors in the circuit are handled similarly, with the inductor current (computed for transient simulation using the modified nodal analysis) as the controlling variable.

Then we update the average power up to the time t_1 as follows.

$$P_{t_1} = \frac{P_{t_0} \cdot t_0 + P_{t_1 - t_0} \cdot (t_1 - t_0)}{t_1} \qquad \text{(EQ 7.9)}$$

where $P_{t_1 - t_0}$ denotes the power consumed from t_0 to t_1 and P_{t_i} denotes the power consumed from $t = 0$ to t_i. We perform this calculation for every event of the simulation, and finally sum up the power dissipated at every node to obtain the power consumption of the circuit.

7.4 Experimental Results

Circuits	Size (cell units)	MOS devices	Capacitors	Description
mux2b16	208	214	134	16 bit 2-to-1 mux
cla16	993	1200	500	16 bit carry look ahead adder
mult8	1276	2691	1008	8 bit wallace tree multiplier
mult16	5320	9778	3148	16 bit wallace tree multiplier
multp16	6344	11314	3922	16 bit pipelined multiplier

Table 7.1: Circuits used in the experiment

The power estimator described in the previous section has been implemented in the SWEC framework [7]. We use HSPICE along with this program to compare benchmark results.

For the purpose of this benchmarking, we used industrial circuits obtained from LSI Logic Corporation. The netlists were extracted from the layout of real designs generated in the design synthesis environment of LSI Logic using their ASIC cell libraries and submicron devices. These circuits range in size from 200 to 6000 cell units in the LSI technology. These netlists were then used along with input stimuli to serve as data for both HSPICE and SWEC. The circuits used in the experiments are listed in Table 1.

Table 2 shows the power measurement results for these circuits. The results were obtained on a DEC 5100/125 platform with a 96 Mbyte memory. The second and third columns in the table correspond to the power dissipation results reported by SWEC and HSPICE respectively. The last column shows the percentage error in the SWEC measurement as compared to HSPICE. It can be seen that for the first 4 circuits the absolute percentage errors range from 1.9% to 10.2%. Note that HSPICE was not able to handle the last two circuits (multp16 - 6344 cell units and mult16 - 5320 cell units) due to memory limitations and/or CPU time constraints. SWEC successfully completed simulation in all

Circuits	SWEC	HSPICE	% Error
mux2b16	0.698	0.685	1.90
cla16	1.430	1.540	-7.14
mult8	12.254	11.483	6.71
mult16	54.547	51.580	5.74
multp16	40.125	*	-

Table 7.2: Results of Power Measurements (in mW) (* indicates HSPICE could not complete)

Circuits	SWEC	HSPICE	Speed up
mux2b16	9.12	166.19	19
cla16	51.75	2017.07	39
mult8	229.82	13089.65	57
mult16	936.00	80500.20	86
multp16	907.78	*	-

Table 7.3: Comparisons of Run Times (in seconds) (* indicates HSPICE could not complete)

examples we ran.

Table 3 shows the speed up achieved by SWEC as compared to HSPICE. For the circuits that HSPICE was able to handle, the speed-up ranged from 18.22 for a 200 cell unit design to 56.96 for a 1276 cell unit design. Note that the speed-up increases as the circuit size grows.

Chapter 8

Conclusions and Future Work

SYMPHONY, an efficient mixed signal simulator was presented. 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 achieves additional speed-up by modeling the voltages by Piecewise Linear waveforms. Device characteristics of bipolar elements in digital subcircuits are modeled by a Piecewise Linear approximation using the Extended 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. The simulator is implemented in an event-driven framework with local and global clocks for event management. A set of benchmark results were presented on a suit of BiMOS circuits. Work on further improving the performance of SYMPHONY is under progress.

We also presented an approach to power estimation using SWEC. The proposed method exploits the stepwise equivalent conductance approximation to efficiently compute power dissipation while speeding up the transient simulation process. Based on the results presented, we believe that SWEC can effectively replace HSPICE when a relatively accurate power estimate is desired without the large run times.

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] G. Arnout, and H. De Mann, "The Use of Threshold Functions and Boolean Controlled Network Elements for Macromodeling of LSI Circuits," *IEEE Journal of Solid-State Circuits*, Vol. SC-13, pp. 326-332, June 1978.

[3] 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.

[4] G. R. Boyle, "Simulation of Integrated Injection Logic," *Ph.D. thesis, University of California, Berkeley*, 1978.

[5] 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.

[6] P. Buch, and E. S. Kuh, "Mixed Signal Circuit Simulation of BiCMOS Analog/Digital Circuits," Submitted to *the IEEE International Conference on Computer-Aided Design*, Nov. 1995.

[7] P. Buch, S. Lin, V. Nagasamy, and E. S. Kuh, "Techniques for Fast Circuit Simulation Applied to Power Estimation of CMOS Circuits," *Proceedings of the International Symposium on Low Power Design*, pp. 135-138, April 1995.

[8] A. P. Chandrakashan, S. Sheng, and R. Brodersen, "Low Power CMOS Digital Design," *IEEE Transactions on Solid-State Circuits*, Vol. 27, No. 4, pp. 473-483, April 1992.

[9] 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, Nov. 1992.

[10] 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.

[11] 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.

[12] M. A. Cirit, "Estimating Dynamic Power Consumption of CMOS Circuits," *IEEE International Conference on Computer-Aided Design*, pp. 534-537, Nov. 1987.

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

[14] A.-C. Deng, "Power Analysis for CMOS/BiCMOS Circuits," *Proceedings of the International Workshop on Low Power Design*, 1994.

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

[16] 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.

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

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

[19] 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.

[20] 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.

[21] *HSPICE Version H92 User's Manual*, Meta-Software Inc., Campbell, CA.

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

[23] 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.

[24] S. M. Kang, "Accurate Estimation of Power Dissipation in VLSI Circuits," *IEEE Journal of Solid-State Circuits*, Vol. SC-21, pp. 889-891, Oct. 1986.

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

[26] K. Keutzer, and P. Vanbekbergen, "The Impact of CAD on the Design of Low Power Digital Circuits" In *Proceedings of the 1994 IEEE Symposium on Low Power Electronics*, pp 42-45, San Diego, CA, Oct 10-12, 1994.

[27] Y. H. Kim, "ELOGIC: A Relaxation-Based Switch-Level Simulation Technique," *MS Project Report, University of California, Berkeley*, 1985.

[28] Y. H. Kim, J. Kleckner, R. Saleh, and A. R. Newton, "Electrical-Logic Simulation," *Proceedings of the IEEE International Conference on Computer-Aided Design*, Nov. 1984.

[29] 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.

[30] P. E. Landman, and J. M. Rabaey, "Power Estimation for High Level Synthesis", *Proceedings of EDAC-EUROASIC '93*, Paris, France, pp. 361-366, February 1993

[31] 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.

[32] 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.

[33] S. Lin, "Circuit Simulation for VLSI and Electronic Packaging Design," *Ph.D. thesis, University of California, Berkeley*, 1993.

[34] 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.

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

[36] F. Najm, "Estimating Power Dissipation in VLSI Circuits", *Technical Report, University of Illinois at Urbana-Champaign*, May 1994.

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

[38] A. R. Newton, and D. O. Pederson, "Analysis Time, Accuracy and Memory Requirement Trade-offs in SPICE2," *Proceedings of the Asilomar Conference*, 1975.

[39] 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.

[40] 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.

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

1984.

[42] J. Rubinstein, Jr., P. Penfield, and M. Horowitz, "Signal Delay in RC Tree Networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 2, No. 7, pp. 202-211, July 1983.

[43] 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.

[44] 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.

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

[46] P. Saviz, "Circuit Simulation by Hierarchical, Waveform Relaxation," *Ph.D. thesis, Columbia University*, 1990.

[47] 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.

[48] 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.

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

[50] 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.

[51] C-Y. Tsui, M. Pedram, and A. M. Despain, "Efficient Estimation of Dynamic Power Con-

sumption under a Real Delay Model," *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 224-228, Nov. 1993.

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

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] K. Yamamura, "Piecewise Linear Analysis of Non-linear 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

[58] 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.

[59] 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.