AN ABSTRACT OF THE THESIS OF

Taras Dudar for the degree of Master of Science in Electrical and Computer
Engineering presented on December 11, 2002.
Title: Algorithms and Simulators for Coupled Device/Circuit Simulation.

Redacted for Privacy

Abstract approved: _____

Kartikeya Mayaram

Algorithms and simulators comprised of SPICE3 as a circuit level simulator
and two device simulators EOFLOW and PROPHET for accurate simulation of
new types of devices are presented in this thesis.

An integration of EOFLOW with SPICE3 creates a capability for efficient
simulation of a system containing interconnected electroosmotic flow channels
together with control electronics. Using this simulator, an accurate simulation of a
complex interconnection of channels has been performed. In addition, various flow
control schemes have been evaluated for their effectiveness.

Coupling of PROPHET and SPICE3 allows for the simulation of accurate
semiconductor device models. This capability is necessary for critical RF and
analog applications. The coupled SPICE3-HB-PROPHET simulator incorporates
the harmonic balance algorithm for large-signal frequency domain analysis.
Applications of this analysis are demonstrated in the noise coupling between
devices sharing the same silicon substrate.

Algorithms and Simulators for Coupled Device/Circuit Simulation

by

Taras Dudar

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented December 11, 2002
Commencement June 2003

Master of Science thesis of <u>Taras Dudar</u> presented on <u>December 11, 2002</u>

APPROVED:

Redacted for Privacy

Major Professor, representing Electrical and Computer Engineering

Redacted for Privacy

Head of the Department of Electrical and Computer Engineering

Redacted for Privacy

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for Privacy

Taras Dudar, Author

ACKNOWLEDGEMENT

I take this opportunity to express my gratitude and appreciation towards people who assisted me during my schooling at Oregon State University (OSU) in general and while working on my research in particular.

To begin with, I would like to say a "Big Thank You" to Dr. Oleg Mikulchenko, my advisor at National Technical University of Ukraine "Kyiv Polytechnic Institute", who initiated my engineering career in computer-aided design, having introduced me to the subject over seven years ago. He has always inspired me to work hard thus stimulating my academic growth. It is owing to him that I came to the OSU and was able to complete my Master's program.

My sincere thanks goes to Dr. Kartikeya Mayaram, my research advisor at OSU for having given me an opportunity to come to OSU, providing me with a research topic and giving tremendous support to accomplish my goal. I also took several of his classes, which due to Prof. Mayaram's unique and exceptional teaching methods supplied me with crucial theoretical knowledge, essential for my research work.

Speaking about the research, I would like to acknowledge Dr. Oleg Mikulchenko again to whom I am very grateful for the initial development of the project on "Algorithms and Simulators for Coupled Device/Circuit Simulation" and for equipping me with all the knowledge needed to continue it.

This work would not have been possible without contacts and interactions with researchers from the University of Illinois at Urbana Champaign and Stanford University. My true thanks goes to Dr. Daniel Yergeau – a researcher from Stanford University for his continued help with all issues related to SPICE3-HB-PROPHET coupling development and to Rui Qiao – a Ph.D. student from the University of Illinois at Urbana-Champaign for his help in SPICE3-EOFLOW coupling development.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

To my family

ALGORITHMS AND SIMULATORS FOR COUPLED DEVICE/CIRCUIT SIMULATION

1   INTRODUCTION

Significant improvements and growth in semiconductor technology in the last two decades have created the possibility for using new types of devices in systems. Existing devices have been shrunk to such small sizes and pushed to such high frequencies of operation that old models no longer work. Decreased time to market demands very accurate and fast simulation capabilities. In the case when an analytical or compact model for a device does not exist (simply because the device is new) or when the existing model cannot be used because the device is pushed to its limits in frequency and size, the only solution is to use a numerical model to describe device behavior. Since numerical models are the core of device simulators, the need for simulation of circuits containing devices (described as numerical models) has spurred the development of algorithms for coupled device/circuit simulation [1-3].

1.1   Summary of Previous Work

Development of coupled device/circuit simulators started more then twenty years ago with the simulator MEDUSA [1]. MEDUSA enabled mixed-level simulations by combining circuit and device simulators. MEDUSA included a one-dimensional device simulator for bipolar junction transistor (BJTs) and a quasi-two-dimensional simulator for metal oxide semiconductor field effect transistor (MOSFETs) [2].

In the late 1980's, research work was focused on a mixed-level device and circuit simulator CODECS [3], an integration of a device simulator into SPICE3 [4]. CODECS can be used to simulate circuits containing diodes, MOSFETs and BJTs that were described using two-dimensional numerical models. Various simulation algorithms were studied and implemented for effective coupling of numerical devices with a circuit simulator. A modified two-level Newton scheme for the dc operating point analysis proved to be the most successful. The advantage was that it required a fewer number of circuit-level iterations. Full-block-LU decomposition algorithms were used for transient analysis. The combination of both algorithms provided reasonable convergence and run-time performance.

In the early 1990's, three-dimensional (3D) device simulators were coupled to the circuit simulator SPICE3. The use of mixed-level circuit and 3D-device simulation emerged from the three-dimensional nature of problems. A single-event upset (SEU) phenomenon in static random access memory (SRAM) cells is one such case [5]. A two-carrier 3D-device simulator SIERRA [6] and the circuit simulator SPICE3 (with an architecture similar to CODECS) were used to analyze these phenomena in detail [7]. Iterative solution techniques were used at the device level and this necessitated the development of new algorithms.

With the emergence of radio frequency (RF) integrated circuits (ICs) in the early 1990's the need for RF circuit simulators was identified. Since conventional SPICE3 analyses did not satisfy the needs of RF circuit designers, several techniques were developed for the efficient and accurate simulation of RF ICs. Frequency-domain methods, time-domain methods and a combination of them (called hybrid or mixed frequency-time methods) are three of these techniques, that are widely used for steady-state analysis of circuits with widely different time constants [8].

Faster and accurate simulation of RF ICs can be achieved by using periodic time-domain steady-state analysis, multitone harmonic balance (HB), mixed frequency-time-domain methods and envelope methods. These techniques also are

necessary for nonlinear and cyclostationary noise analysis simulation in RF circuits.

The simulation of RF circuits has to be done in the periodic steady state. This allows for accurate simulation of different circuit characteristics such as distortion, power, frequency, noise and transfer functions. The time-domain steady-state method in the context of coupled device and circuit simulation has been implemented recently [9]. This method combined with accurate numerical models for diodes, BJTs and MOSFETs provides accurate and reliable steady-state simulation for RF circuits.

The harmonic balance method, a frequency-domain approach, in the context of coupled simulation solves problems that the time domain methods are unable to solve, e.g., quasiperiodic steady state problems (mixers) [10]. In [10] the HB method was implemented at the circuit level. The results obtained using this method are accurate and efficient due to the use of circuit-level waveforms [11].

A coupled device/circuit simulator has been extensively used in the integrated circuit community. The applications include RF circuit simulation, simulation of SEU effects in SRAMs, power devices simulations, and validation of nonquasistatic MOSFET models [12]. Recent publications show that coupled device/circuit simulations are extremely important not only in the field of RF circuit design but also in micro electro mechanical systems (MEMs) [13], piezoelectric [14] and sensor [15] simulations.

Verification of microfluidic systems is another recent application of coupled device/circuit simulators. Large-signal analyses of these systems are performed efficiently by such simulators. Construction and validation of macromodels is another issue being accomplished by the coupled device/circuit simulators. One such simulator for microfluidic applications has been described in [13]. NEKTAR [16], a microfluidic simulator, and SPICE3, a circuit simulator, have been coupled in [13]. The coupled simulator supports compact models for the electronic components and macromodels and numerical models for microfluidic devices. A

complete microfluidic system with associated control electronics can be simulated provided suitable boundary conditions are set in the fluid solver. A new time-stepping technique designed especially for coupled domain problems with an emphasis on microfluidics was also proposed in [13]. This technique was taken as the basis for the development of the circuit/microfluidic simulator presented in this work.

1.2   Contributions of This Work

This work examines the framework for coupled device and circuit simulation for two very diverse applications. A general framework is used to implement simulators for electroosmotic flows and advanced semiconductor devices. Coupling algorithms have been developed and the applications of these simulators have been demonstrated. Specifically, EOFLOW and SPICE3 were coupled for simulation of microfluidic devices, and PROPHET and SPICE3 were combined for the simulation of semiconductor devices.

The SPICE3-EOFLOW coupling allows the simulation of DC and transient analysis of circuits containing microfluidic channels and control electronics to control fluid behavior. Biological chips, in which electronic circuitry and interconnected channels are used for mixing and separation of different kinds of fluids, can be readily simulated.

The SPICE3-HB-PROPHET simulator provides DC, AC and HB analyses for circuits containing semiconductor devices described by numerical models. Large-signal response of circuits where substrate coupling noise mixes with a signal and significantly alters circuit performance can be determined. The transient analysis has been previously implemented for PROPHET [17] and therefore was not considered in this work.

1.3   Thesis Organization

The thesis is organized as follows. In Chapter 1, an introduction to the problem of coupled device/circuit simulation is described.

Chapter 2 covers the theoretical background of a generic coupled device/circuit simulator. Comparison of two coupling approaches – the full-system solution approach versus the two-level Newton approach is provided. Device stamping techniques for different analyses are also described along with a general-purpose framework for coupled device and circuit simulation.

The SPICE3/EOFLOW coupling is described in Chapter 3. An introduction to microfluidic systems simulation is given followed by a brief description of EOFLOW. The coupling of this solver with SPICE3 including appropriate boundary conditions are then described. Validation of the simulation results, flow control examples, and a complex simulation example are also provided.

Chapter 4 provides a description of the SPICE3/PROPHET coupling. An introduction to systems containing numerical devices and coupling algorithms is given. Validation of the simulation algorithms for DC and HB analyses and simulation results are also presented.

Chapter 5 summarizes this work and defines possible future work.

2 THEORETICAL BACKGROUND

2.1 Device Simulations Basics

Coupled device/circuit simulators are an integration of circuit simulators and device solvers. In general, a device simulator (device solver) is a program to solve the specified system of partial differential equations (PDEs) since any device can be described by a system of PDEs. There are several techniques to solve a system of PDEs based on the space and time discretization converting them into a finite number of nonlinear differential-algebraic equations. Usually device solvers use finite difference or finite element discretization techniques.

If the device is divided into nonoverlapping rectangular regions for two dimensions (2D) or rectangular solids for three dimensions (3D) by grid lines parallel to the x, y and z axes, the discretized equations are assembled at each grid node by approximating spatial derivatives with difference expressions. For the rectangular mesh shown in the Figure 2.1 the derivatives of a function f(x, y) at the grid point (i, j) are [2]:

$$\left(\frac{\partial f}{\partial x}\right)_{i,j} = \frac{f(x_{i+1}, y_j) - f(x_{i-1}, y_j)}{h_i + h_{i-1}} \tag{2.1}$$

$$\left(\frac{\partial f}{\partial y}\right)_{i,j} = \frac{f(x_i, y_{j+1}) - f(x_i, y_{j-1})}{k_j + k_{j-1}} \tag{2.2}$$
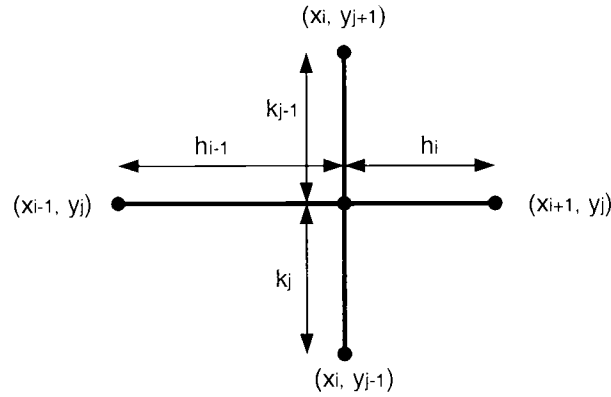
Figure 2.1 Grid for finite difference discretization in two space dimensions.

In the finite element method, the device is divided into a number of nonoverlapping elements (usually of triangular or rectangular shape). With appropriate boundary conditions, a system of nonlinear equations is obtained that can be solved by applying a Newton's method.

For electrical device simulation, device solvers deal with the system of PDEs based on the Poisson's equation and the electron and hole current continuity equations [2]:

$$\nabla \varepsilon E = q(N_D - N_A + p - n) \tag{2.3}$$

$$\frac{1}{q}\nabla J_n = \frac{\partial n}{\partial t} - (G - R) \tag{2.4}$$

$$\frac{1}{q}\nabla J_p = \frac{\partial p}{\partial t} + (G - R) \tag{2.5}$$

where:

$$E = -\nabla \psi \qquad (2.6)$$

$$J_n = -q\mu_n n \nabla \psi + qD_n \nabla n \qquad (2.7)$$

$$J_p = -q\mu_p p \nabla \psi - qD_p \nabla p \qquad (2.8)$$

and:

ε = dielectric constant of the material (F cm^{-1})

q = electronic charge (C)

ψ = electrostatic potential (V)

n (p) = electron (hole) concentration (cm^{-3})

E = electric field (V cm^{-1})

J_n (J_p) = electron (hole) current density (A cm^{-2})

μ_n (μ_p) = electron (hole) mobility (cm^2 V^{-1} s^{-1})

D_n (D_p) = electron (hole) diffusivity (cm^2 s^{-1})

N_D (N_A) = donor (acceptor) concentration (cm^{-3})

G = net generation rate (cm^3 s^{-1})

R = net recombination rate (cm^3 s^{-1})

For microfluidic devices the essential part is the solution of the Navier-Stokes equations. The transient Navier-Stokes equations with the electric force term are given as [18]:

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u = v\nabla^2 u - \frac{1}{\rho}\nabla p + \frac{1}{\rho}F \qquad (2.9)$$

$$\nabla \cdot u = 0 \qquad (2.10)$$

where:

u = velocity vector (m/s)

t = time (s)

v = kinematic viscosity (m^2/s)

ρ = density (kg/m^3)

p = pressure (Pa)

F = electric force per unit mass (N/kg)

Equations (2.3)-(2.10) can be solved using the finite difference or finite element method described above.

2.2 Two-Level Newton Approach vs. Full-Newton Approach

There are two fundamental approaches for solving coupled systems the full-Newton method and the two-level Newton method. The full-Newton algorithm is an application of the Newton method to the system of equations obtained by combining the device- and circuit-level equations. The two-level Newton method solves the device- and circuit-level equations separately. This method provides better convergence for DC analysis [12, 19]. From a computational cost point of view, the full-Newton approach is better because it does not require several device-level Newton iterations for each circuit-level Newton iteration. A good initial guess is required for the full-Newton method for convergence. Therefore, this method is robust for transient analysis [12] where the previous time point values can be used as a good initial guess. For DC and HB analyses, a good initial guess cannot be easily obtained. For this reason, a two-level Newton algorithm has to be used.

2.3   Device Stamping

Device stamping is the procedure for adding the contribution of a particular device to a circuit matrix and the right-hand-side (RHS) vector. One can think of a general two terminal device as a parallel combination of a nonlinear resistor and a nonlinear capacitor as shown in Figure 2.2. The nonlinear resistor represents the conductive part of the device and the nonlinear capacitor represents the susceptive part.
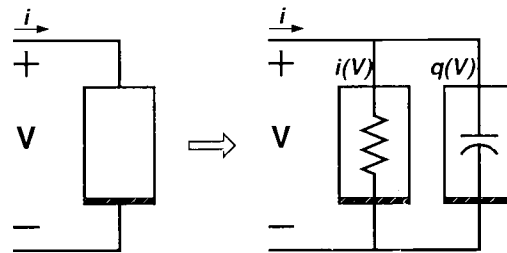
Figure 2.2 Representation of a general nonlinear device as a parallel combination of a nonlinear resistor and a nonlinear capacitor.

For DC analysis, the capacitor is an open circuit and any two terminal device can be represented simply as a nonlinear resistor. For every Newton iteration the equivalent circuit representation also known as the companion network for the nonlinear resistor is shown in Figure 2.3.
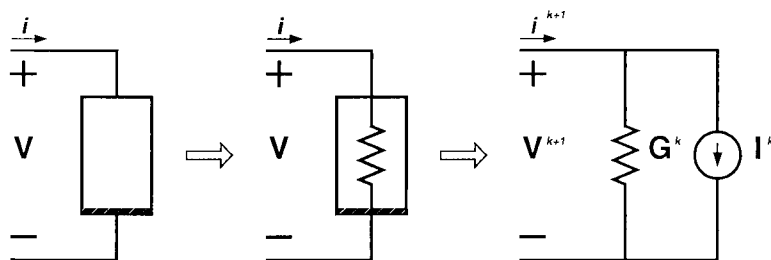
Figure 2.3 Nonlinear device companion network for DC analysis at the *k-th* Newton iteration.

In Figure 2.3,

$$G^k = \frac{\partial i}{\partial V}\bigg|_{V^k} \quad \text{and} \quad I^k = i\left(V^k\right) - G^k V^k \tag{2.11}$$

where k is the Newton iteration count.

For AC analysis, the nonlinear resistor and capacitor are represented by their linearized models at the operating point. The companion model for AC analysis is shown in Figure 2.4.
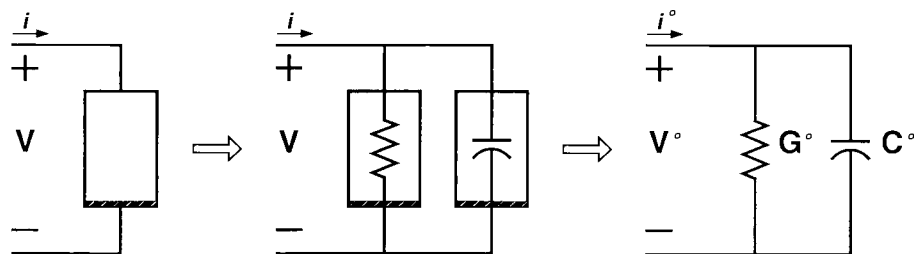
Figure 2.4 Nonlinear device companion network for AC analysis.

In Figure 2.4, V^o and i^o are the operating point voltage and current, respectively, and G^o and C^o are the small-signal conductance and capacitance, respectively of the device at the operating point.

$$G^o = \frac{\partial i}{\partial V}\bigg|_{V^o} \quad \text{and} \quad C^o = \frac{\partial q}{\partial V}\bigg|_{V^o} \tag{2.12}$$

For the harmonic balance (HB) method, the companion model is similar to that for AC analysis. The only difference is that instead of the operating point voltage V^o we apply a bias voltage V^b to calculate G^b and C^b for every sampled bias point. The current through the device should be also calculated for this bias voltage. The companion model for HB analysis is shown in Figure 2.5.
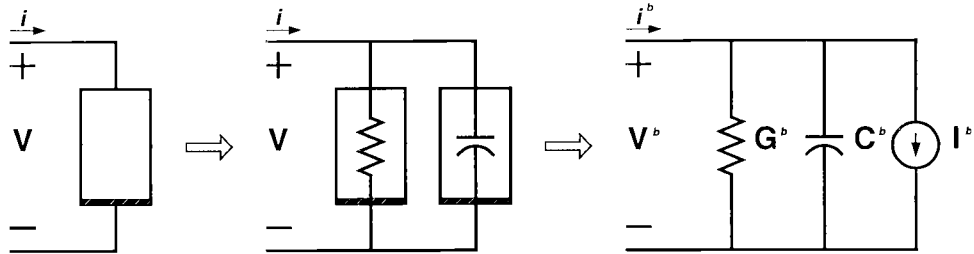
Figure 2.5 Nonlinear device companion network for HB analysis.

In Figure 2.5, V^b is a biasing voltage, G^b the small-signal conductance and C^b the small-signal capacitance of the device at the biasing point.

$$G^b = \frac{\partial i}{\partial V}\bigg|_{V^b} \quad \text{and} \quad C^b = \frac{\partial q}{\partial V}\bigg|_{V^b} \tag{2.13}$$

For transient analysis, we also have to consider the time discretization as shown in Figure 2.6.
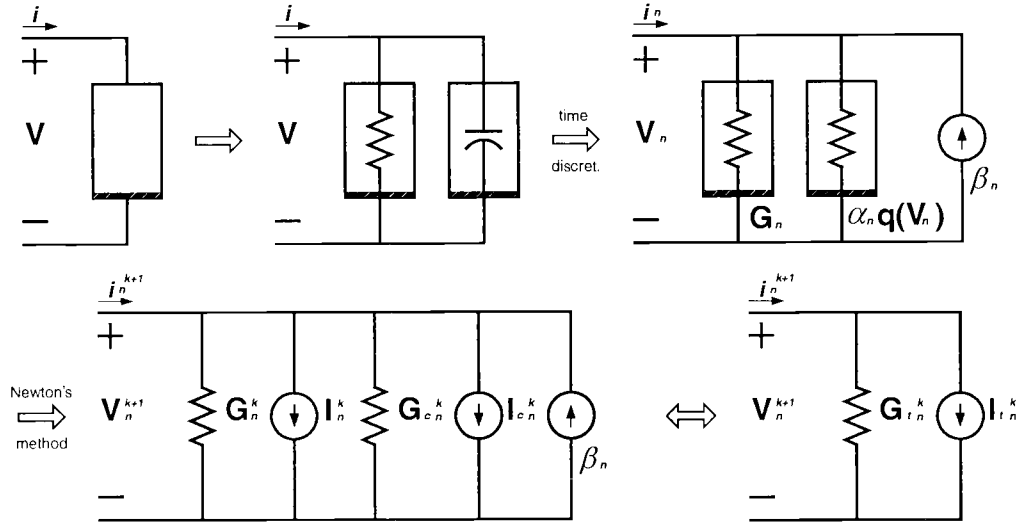
Figure 2.6 Nonlinear device companion network for transient analysis. Newton's method is used for the solution of the nonlinear equations.

In Figure 2.6, the subscript n denotes the n-th time point and superscript k denotes the k-th Newton-Raphson (NR) iteration. V_n and i_n are the voltage across the device and the current through the device, respectively, at the n-th time point. V_n^{k+1} and i_n^{k+1} are the voltage across and the current through the device at the n-th time point for the $k+1$-th NR iteration. G_n and G_n^k are the device conductances for V_n and V_n^k voltages across the device, respectively.

$$G_n = G(V_n) = \frac{\partial i}{\partial V}\bigg|_{V_n} \quad \text{and} \quad G_n^k = G(V_n^k) = \frac{\partial i}{\partial V}\bigg|_{V_n^k} \tag{2.14}$$

α_n and β_n are coefficients of the integration method for the n-th time point. $q(V)$ is the charge for the nonlinear capacitor as a function of the voltage and $q(V_n)$ is its value for voltage V_n.

$G_{cn}{}^k$ is the conductance for the nonlinear capacitor in the companion circuit.

$$G_{cn}^k = \alpha_n \left. \frac{\partial q}{\partial V} \right|_{V_n^k} \tag{2.15}$$

$I_n{}^k$ and $I_{cn}{}^k$ are currents due to the resistor and capacitor nonlinearities, respectively.

$$I_n^k = i\left(V_n^k\right) - G_n^k V_n^k = i\left(V_n^k\right) - V_n^k \left. \frac{\partial i}{\partial V} \right|_{V_n^k} \tag{2.16}$$

$$I_{cn}^k = \alpha_n q(V_n^k) - G_{cn}^k V_n^k = \alpha_n q(V_n^k) - \alpha_n V_n^k \left. \frac{\partial q}{\partial V} \right|_{V_n^k} \tag{2.17}$$

$G_{tn}{}^k$ and $I_{tn}{}^k$ are, respectively, the total conductance and the intercept on the current axis of the linearized device model for a voltage V_n^k, as shown in Figure 2.7.
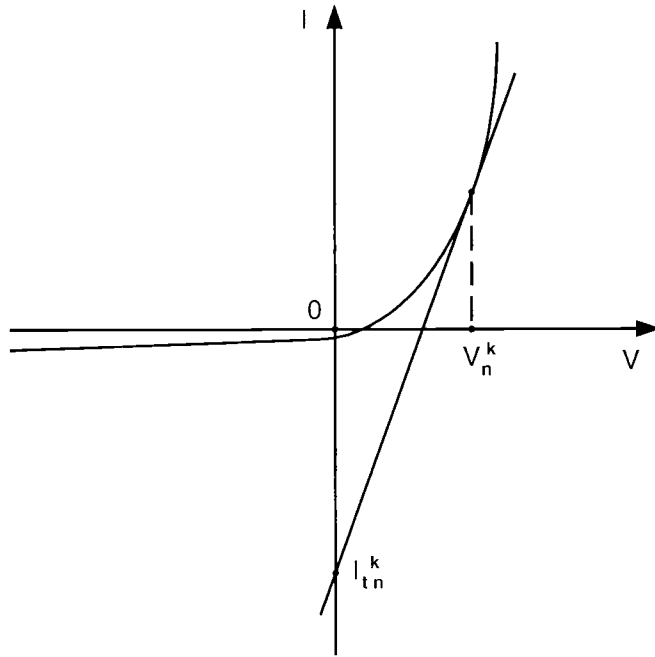
Figure 2.7 Linearized device model for voltage V_n^k.

$$G_{tn}^k = G_n^k + G_{cn}^k = \left.\frac{\partial i}{\partial V}\right|_{V_n^k} + \alpha_n \left.\frac{\partial q}{\partial V}\right|_{V_n^k} \tag{2.18}$$

$$I_{tn}^k = I_n^k + I_{cn}^k - \beta_n =$$

$$= i\left(V_n^k\right) + \alpha_n q(V_n^k) - \beta_n - V_n^k \left.\frac{\partial i}{\partial V}\right|_{V_n^k} - \alpha_n V_n^k \left.\frac{\partial q}{\partial V}\right|_{V_n^k} = \tag{2.19}$$

$$= i\left(V_n^k\right) + \alpha_n q(V_n^k) - \beta_n - G_{tn}^k V_n^k$$

Next, the calculation of all these variables for the numerical device is described. Assume that for a general device, one can apply a terminal voltage, V_n, observe the current through the device, i_n, and perform a transient analysis. The

incremental conductance, $\dfrac{\Delta i}{\Delta V}$, can be obtained at every time point t_n by applying a perturbation ΔV_n to the voltage V_n (resulting in $V_n + \Delta V_n$) and calculating the corresponding current, $i_n + \Delta i_n$.

Since integration is involved during transient analysis within the device,

$$i_n = i(V_n) + \alpha_n q(V_n) - \beta_n \tag{2.20}$$

Therefore, for every NR iteration,

$$\left.\frac{\partial i_n^k}{\partial V_n^k}\right|_{V_n^k} = \left.\frac{\partial i}{\partial V}\right|_{V_n^k} + \alpha_n \left.\frac{\partial q}{\partial V}\right|_{V_n^k} = \left.\frac{\Delta i_n^k}{\Delta V_n^k}\right|_{\Delta V_n^k \to 0} \cong \frac{\Delta i_n^k}{\Delta V_n^k} \tag{2.21}$$

These are all the equations that are required for calculating the values of the conductance and the current for the companion model. In cases where the device simulator can provide the derivatives required in Equation (2.21), the approximation of derivatives in Equations (2.22) is not necessary and this can save some computational effort.

$$G_{tn}^k \cong \frac{\Delta i_n^k}{\Delta V_n^k} \quad \text{and} \quad I_{tn}^k = i_n^k - G_{tn}^k V_n^k \tag{2.22}$$

Recall that $V_n^k = V_{+n}^k - V_{-n}^k$, where V_{+n}^k and V_{-n}^k are the nodal voltages at the positive and negative nodes of the device, respectively, for the n-th time point and the k-th NR iteration. The device contribution to the SPICE3 sparse matrix and the RHS vector is

$$
\begin{array}{cc}
 & N_+ \qquad N_- \\
\begin{array}{c} \\ N_+ \\ \\ N_- \\ \\ \end{array}
\begin{bmatrix}
\cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & +G_{tn}^k & \cdots & -G_{tn}^k & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & -G_{tn}^k & \cdots & +G_{tn}^k & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots
\end{bmatrix}
\begin{bmatrix}
V_{+n}^{k+1} \\ \\ \\ V_{-n}^{k+1} \\ \\
\end{bmatrix}
=
\begin{bmatrix}
-I_{tn}^k \\ \\ \\ +I_{tn}^k \\ \\
\end{bmatrix}
\end{array}
\qquad (2.23)
$$

Also, $i_n^k = i_{+n}^k = -i_{-n}^k$, where i_{+n}^k and i_{-n}^k are the currents flowing into the positive and negative nodes of the device, respectively, for the n-th time point and the k-th NR iteration.

Based on the above analysis and equations the overall companion network for a device is given by the circuit shown in Figure 2.8.
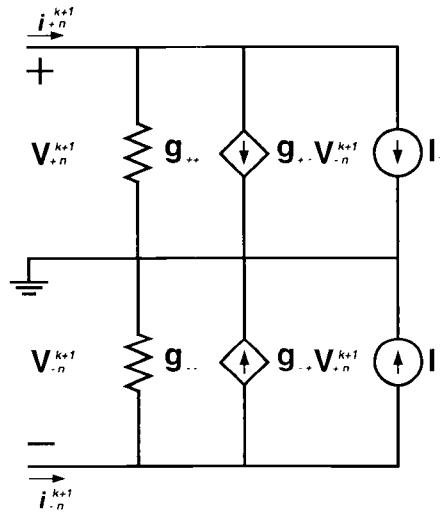
Figure 2.8 Ground referenced companion network for a nonlinear two-terminal device for transient analysis.

In Figure 2.8,

$$g_{++} = \frac{\partial i_{+n}^{k}}{\partial V_{+n}^{k}} = \frac{\partial i_{n}^{k}}{\partial V_{n}^{k}}\frac{\partial V_{n}^{k}}{\partial V_{+n}^{k}} = G_{tn}^{k}; \quad g_{+-} = \frac{\partial i_{+n}^{k}}{\partial V_{-n}^{k}} = \frac{\partial i_{n}^{k}}{\partial V_{n}^{k}}\frac{\partial V_{n}^{k}}{\partial V_{-n}^{k}} = -G_{tn}^{k};$$

$$\text{(2.24)}$$

$$g_{-+} = \frac{\partial i_{-n}^{k}}{\partial V_{+n}^{k}} = \frac{-\partial i_{n}^{k}}{\partial V_{n}^{k}}\frac{\partial V_{n}^{k}}{\partial V_{+n}^{k}} = -G_{tn}^{k}; \quad g_{--} = \frac{\partial i_{-n}^{k}}{\partial V_{-n}^{k}} = \frac{-\partial i_{n}^{k}}{\partial V_{n}^{k}}\frac{\partial V_{n}^{k}}{\partial V_{-n}^{k}} = G_{tn}^{k};$$

and,

$$I_{+} = i_{n+}^{k} - g_{++}V_{+n}^{k} - g_{+-}V_{-n}^{k}; \quad I_{-} = i_{n-}^{k} - g_{-+}V_{+n}^{k} - g_{--}V_{-n}^{k} \quad \text{(2.25)}$$

For generality one can think of a resistor as a voltage-controlled current source (VCCS) in which the controlling voltage is the voltage across the resistor. Therefore, one can represent a general two terminal device for transient analysis as shown in Figure 2.9.
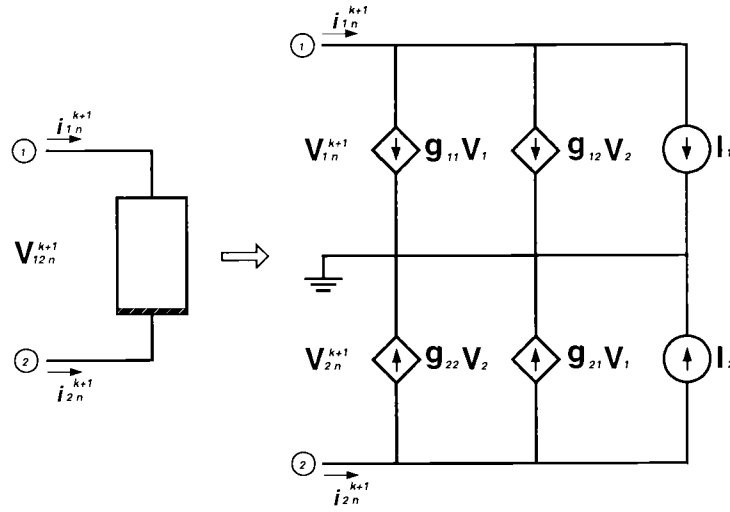
Figure 2.9 Ground referenced companion network for a nonlinear two-terminal
device for transient analysis.

In Figure 2.9,

$$g_{ij} = \frac{\partial i_{in}^{k}}{\partial V_{jn}^{k}} \cong \frac{\Delta i_{in}^{k}}{\Delta V_{jn}^{k}}, \quad I_{i} = i_{in}^{k} - \sum_{j=1}^{2} g_{ij} V_{jn}^{k} \quad \text{and} \quad V_{i} = V_{in}^{k+1} \tag{2.26}$$

for $i, j = 1, 2$.

This idea can be extended for an N-terminal device. As can be observed
from Equation (2.19), the only information required for stamping a device
contribution into the SPICE3 sparse matrix and the RHS vector are the derivatives
of the current with respect to the terminal voltages and currents through the device
for any particular time point and applied voltages. For example, a three terminal
device can be represented as shown in Figure 2.10.
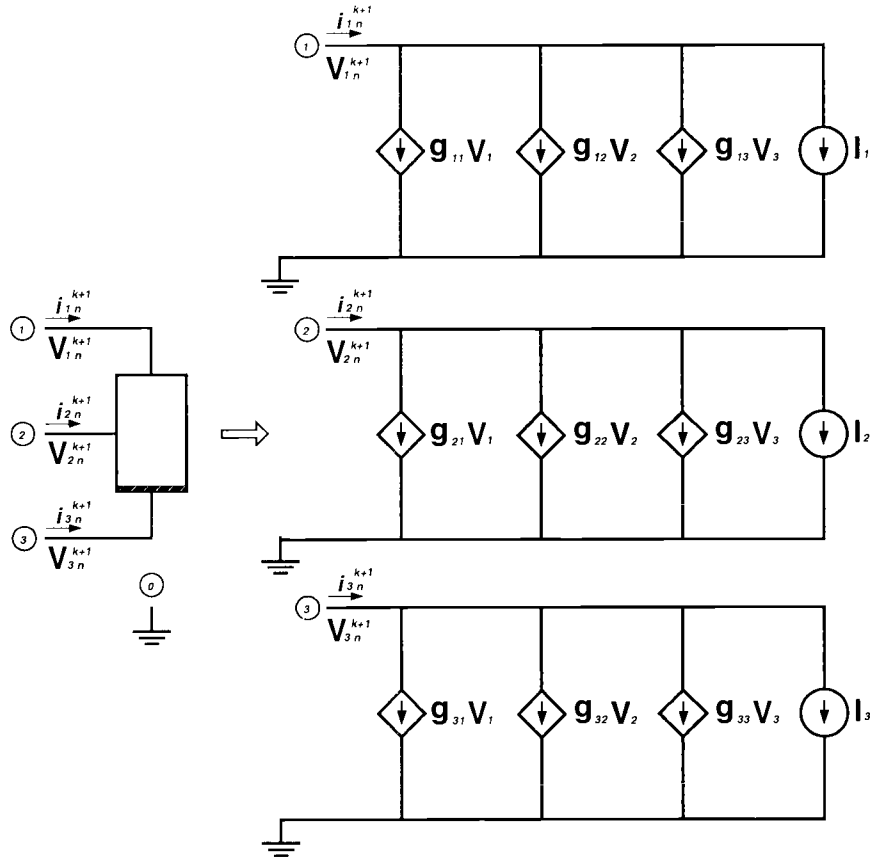
Figure 2.10 Companion network for a three-terminal device for transient analysis.

In Figure 2.10, for $i, j = 1, 2, 3$,

$$g_{ij} = \frac{\partial i_{in}^{k}}{\partial V_{jn}^{k}} \cong \frac{\Delta i_{in}^{k}}{\Delta V_{jn}^{k}}, \quad I_i = i_{in}^{k} - \sum_{j=1}^{3} g_{ij} V_{jn}^{k} \quad \text{and} \quad V_i = V_{in}^{k+1} \tag{2.27}$$

In summary, at the *n-th* time point and the *k-th* NR iteration any N-terminal device can be described by a set of equations (Equation (2.28)). These equations represent a linearized device model at this time point.

$$i_{in}^{k+1} = i_{in}^{k} + \sum_{j=1}^{N} g_{ij}\left(V_{jn}^{k+1} - V_{jn}^{k}\right), \quad i = 1, 2, ..., N \qquad (2.28)$$

For any device since

$$\sum_{i=1}^{N} i_{in}^{k+1} = \sum_{i=1}^{N} i_{in}^{k} = 0 \qquad (2.29)$$

we have

$$\sum_{i=1}^{N}\sum_{j=1}^{N} g_{ij}\left(V_{jn}^{k+1} - V_{jn}^{k}\right) = 0 = \sum_{j=1}^{N}\left[\left(V_{jn}^{k+1} - V_{jn}^{k}\right)\sum_{i=1}^{N} g_{ij}\right] \qquad (2.30)$$

The g_{ij} $(i, j = 1, 2, ..., N)$ pertain to a single device. Changes in other circuit elements (even if g_{ij} are the same) will be reflected in the overall solution of the system. Hence, $\Delta V_{jn}^{k+1} = V_{jn}^{k+1} - V_{jn}^{k}$ which are obtained from a solution of the whole system can be different for the same set of g_{ij}. Therefore, we have

$$\sum_{i=1}^{N} g_{ij} = 0, \quad j = 1, 2, ..., N \qquad (2.31)$$

The condition in Equation (2.31) is fundamental for any device stamp. It is very important that the g_{ij} parameters obtained from device simulators satisfy this condition. If this condition is not satisfied then there is an error in the calculation of the g_{ij} parameters.

2.4 A General-purpose Framework for Coupled Device and Circuit Simulation

Based on the ideas described in Section 2.3 it is now possible to describe a general-purpose framework for coupled device and circuit simulation. Any analytical device model provides conductances between each pair of device terminals and currents into all the terminals. These are stamped into the circuit matrix and the RHS vector at every analysis step. If a device simulator can provide the same information, then it is possible to integrate the device simulator with a circuit simulator. In this case, a device simulator can be treated as a regular device model in the circuit simulator. This basic concept is illustrated in Figure 2.11.
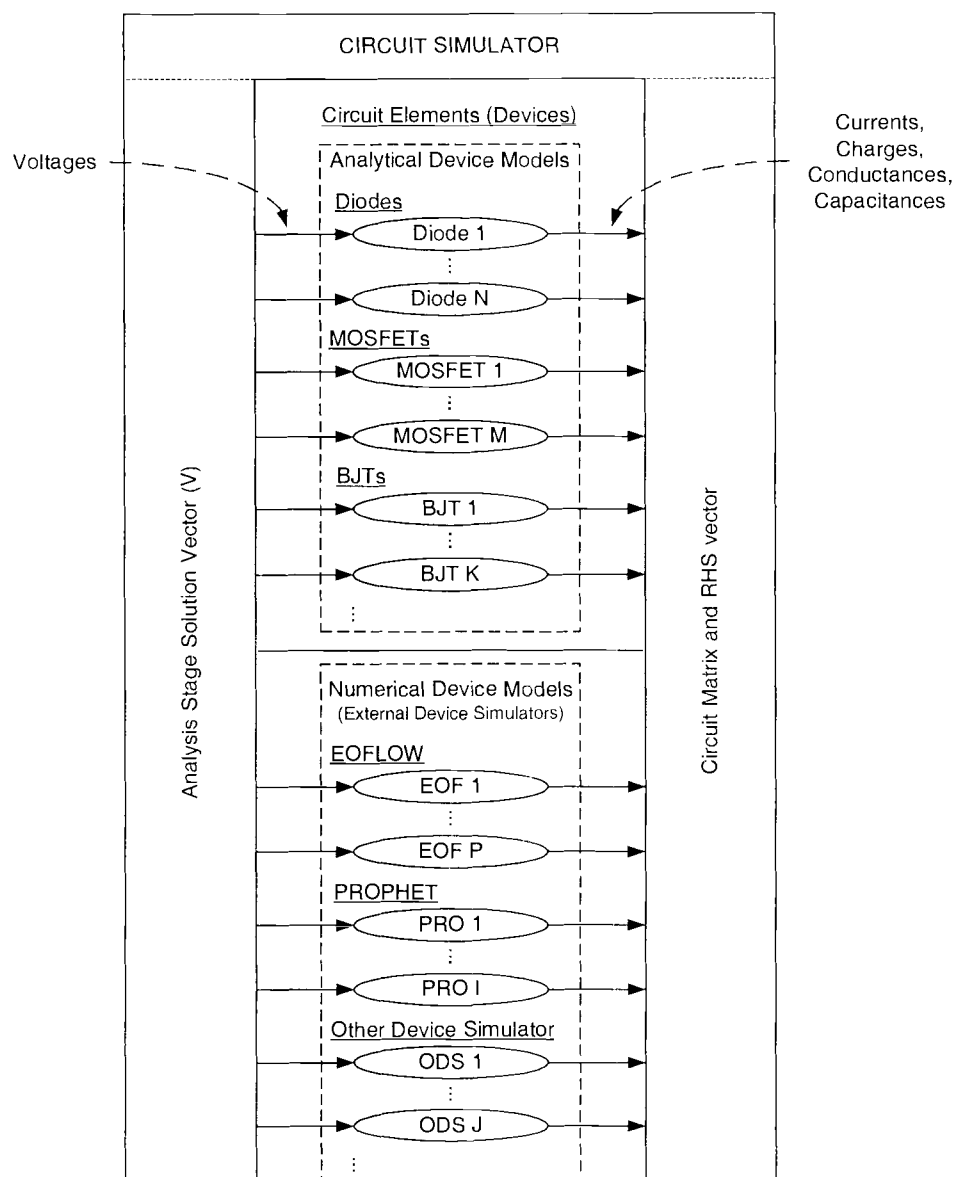
Figure 2.11 General architecture of a coupled device/circuit simulator. This simulator supports multiple device simulators.

One implementation for coupled simulation is shown in Figure 2.12. Here the two simulators communicate through data files instead of direct integration as

in [2]. SPICE3 supplies the simulation point information to the device simulator by modifying the device simulator's input file. After this the device simulator results (currents, conductances, and capacitors) are passed to the SPICE3 circuit matrix and the RHS vector through a data file. This approach can be used for coupling SPICE3 with any device simulator.
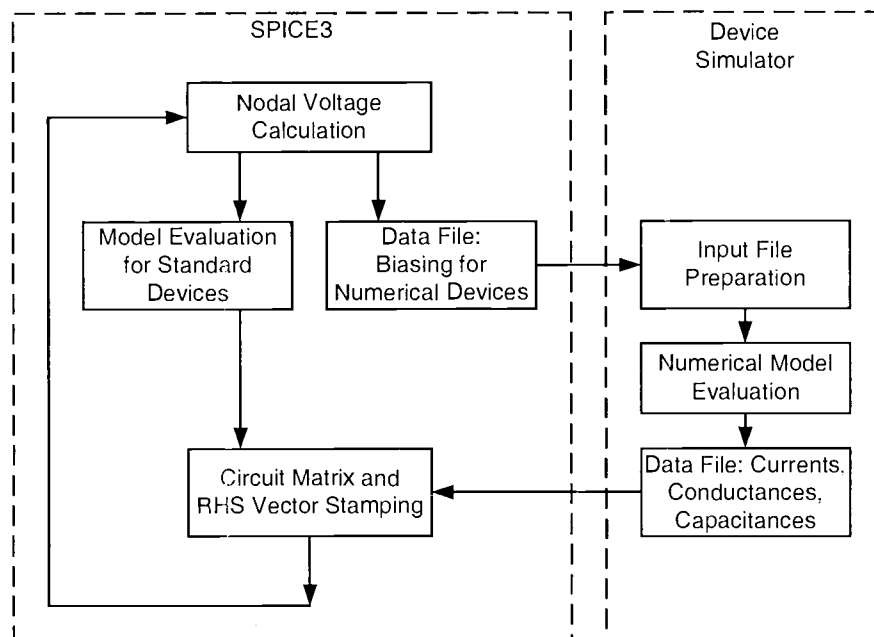
Figure 2.12 Interfacing SPICE3 with any device simulator through data files.

Based on the information provided in Sections 2.1 – 2.4 the data transfer between the circuit and device simulators can by summarized for each analysis.

2.4.1 DC analysis

The circuit simulator provides the operating point information (node voltages) to the device simulator and receives currents through the device and values of the derivatives of these currents with respect to voltages (small-signal conductances) from the device simulator. The block diagram for the coupling in DC analysis is shown in Figure 2.13.

$$\begin{bmatrix} I_1 \\ I_2 \\ \cdots \\ I_n \end{bmatrix}, \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{bmatrix}$$

Circuit Simulator

Device Simulator

$$\begin{bmatrix} V_1 \\ V_2 \\ \cdots \\ V_n \end{bmatrix}$$
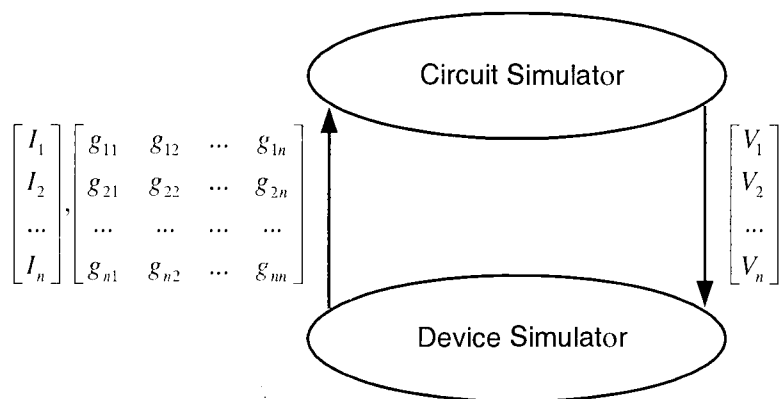
Figure 2.13 DC analysis data transfer.

2.4.2 AC analysis

For AC analysis, the circuit simulator provides the operating point information to the device simulator and obtains small-signal conductances and capacitances from the device simulator for this operating point as shown in Figure 2.14. If the small-signal parameters are frequency dependent, they should be recalculated for all the frequencies.

$$\begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{bmatrix} \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \cdots \\ V_n \end{bmatrix}, \omega$$
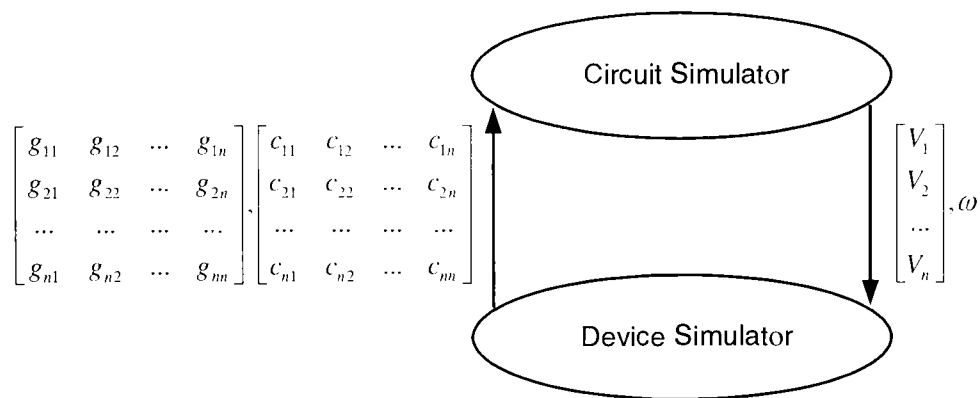
Figure 2.14 AC analysis data transfer.

2.4.3 HB Analysis

For HB analysis, the circuit simulator provides the bias point information to the device simulator. The device simulator supplies the circuit simulator the values of the small-signal conductances and capacitances and currents through the device for this bias point as shown in Figure 2.15.

$$\begin{bmatrix} I_1 \\ I_2 \\ \cdots \\ I_n \end{bmatrix}, \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{bmatrix}, \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \cdots \\ V_n \end{bmatrix}$$
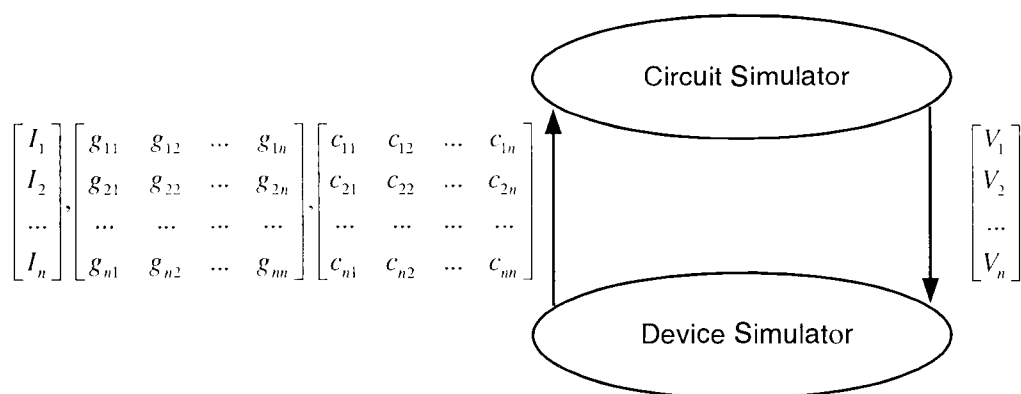
Figure 2.15 HB analysis data transfer.

2.4.4 Transient Analysis

Depending on how transient analysis is implemented in the device simulator there are several ways of coupling the circuit simulator and device simulator for transient analysis. If the device simulator is capable of providing values of the currents through the device, small-signal conductances, charges, and small-signal capacitances at each time step and each Newton iteration, the data transfer between the circuit and device simulator is shown in Figure 2.16.

$$
\begin{bmatrix} I_1 \\ I_2 \\ \dots \\ I_n \end{bmatrix}, \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \dots & \dots & \dots & \dots \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{bmatrix}
$$

$$
\begin{bmatrix} q_1 \\ q_2 \\ \dots \\ q_n \end{bmatrix}, \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}
$$

Circuit Simulator

$$
\begin{bmatrix} V_1 \\ V_2 \\ \dots \\ V_n \end{bmatrix}, time
$$
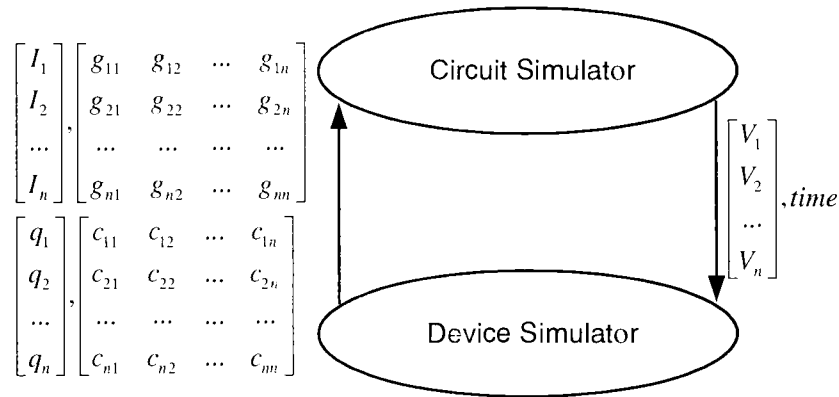
Device Simulator

Figure 2.16 Transient analysis data transfer when the device simulator provides values of the currents through the device, small-signal conductances, charges, and small-signal capacitances at each time step and each Newton iteration.

Some device simulators only provide currents through the device terminals for each time step and each Newton iteration. The data transfer between the circuit and device simulators for this case is shown in Figure 2.17.

$$\begin{bmatrix} i_1 \\ i_2 \\ \cdots \\ i_n \end{bmatrix}, \begin{bmatrix} \dfrac{\Delta i_1}{\Delta V_1} & \dfrac{\Delta i_1}{\Delta V_2} & \cdots & \dfrac{\Delta i_1}{\Delta V_n} \\ \dfrac{\Delta i_2}{\Delta V_1} & \dfrac{\Delta i_2}{\Delta V_2} & \cdots & \dfrac{\Delta i_2}{\Delta V_n} \\ \cdots & \cdots & \cdots & \cdots \\ \dfrac{\Delta i_n}{\Delta V_1} & \dfrac{\Delta i_n}{\Delta V_2} & \cdots & \dfrac{\Delta i_n}{\Delta V_n} \end{bmatrix}$$

Circuit Simulator

Device Simulator

$$\begin{bmatrix} V_1 \\ V_2 \\ \cdots \\ V_n \end{bmatrix}, time$$
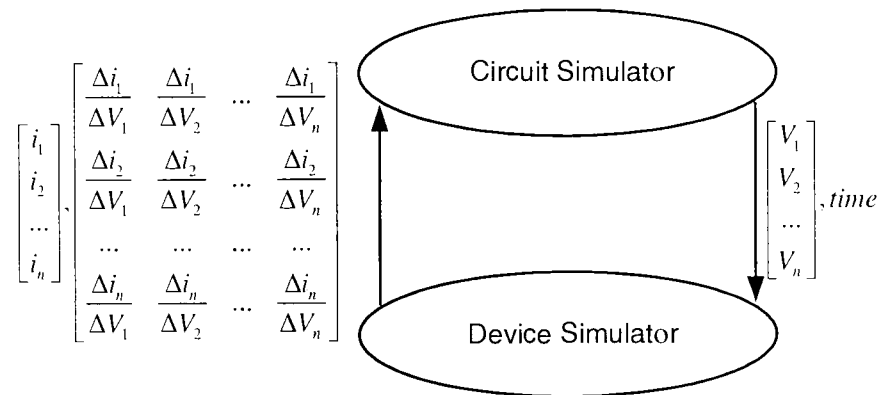
Figure 2.17 Transient analysis data transfer when the device simulator provides only currents through the device terminals for each time step and each Newton iteration.

3 COUPLED SPICE3/EOFLOW SIMULATOR

3.1 Introduction

An interest in microfluidics emerged in the late 1980s with experiments of U.S., Canadian and Swiss scientists, who moved chemical solutions through networks of microfabricated channels etched on a sheet of glass [20]. Although the first microfluidic devices were rather simple, the present day devices and microfluidic systems can be very complex and contain thousands of elements. Areas of applications include biomedical analysis, environmental monitoring, automotive applications and process control [21]. A unique application is the use of electrophoresis chips for high-speed DNA genotyping as described in [22]. In this case, an interconnection of electroosmotic devices in a capillary array electrophoresis chip is used to demonstrate the rapid analysis of biological samples.

In any complex design problem, there is a need for computational prototyping tools whereby devices or systems can be simulated before fabrication. Although the state-of-the-art CAD tools for integrated circuit design are very advanced, such tools for microfluidic systems are still in their infancy. Recently, there has been a focus on simulation of microfluidic systems using coupled circuit and flow simulators [13]. In this chapter, a coupled circuit and electroosmotic flow solver that allows accurate simulation of a complex interconnect of channels is presented. In addition, several examples of how the simulator can be effective in designing control strategies for electroosmotic flow are demonstrated.

3.2 Electroosmotic Flow Solver – EOFLOW

EOFLOW is a new simulation program for electroosmotic fluid transfer as described in [23]. Three basic microfluidic structures: straight-, T- and cross-channels (shown in Figure 3.1) can be simulated in EOFLOW.
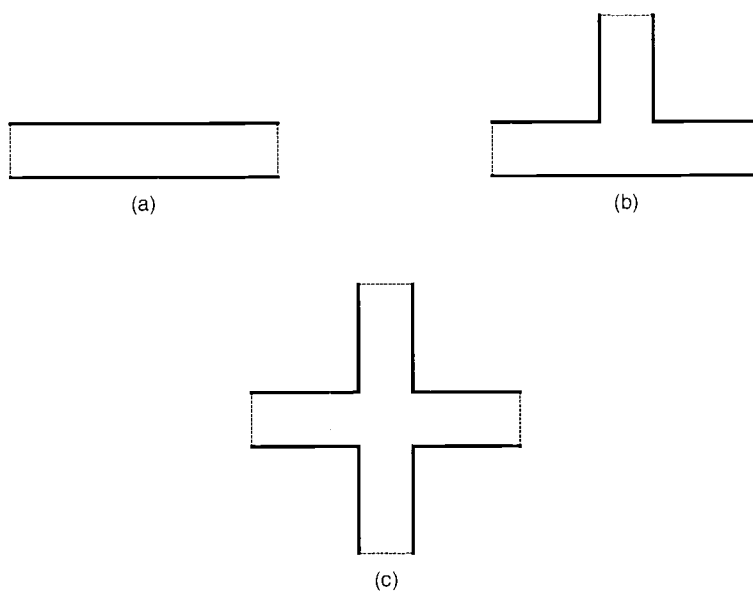
Figure 3.1 Types of channels used in EOFLOW.
(a) straight channel, (b) T-channel, (c) cross channel.

A combination of these can be used to construct complicated structures. Transient responses of these devices (distribution of potentials within a device and the flow rate through the outlets with time) can be simulated for any voltage excitations applied at the outlets. EOFLOW uses a semi-implicit integration method with fixed time steps during transient simulation.

3.3 Coupled Simulation with SPICE3

SPICE3 [4] is a very well established tool for simulation of large systems. It solves a system of nonlinear differential algebraic equations. Although initially developed for electrical circuit simulations, SPICE3 is also suitable for microfluidic system simulations. In flow problems, satisfying the conservation laws with appropriate boundary conditions results in a set of equations similar to that seen in electrical circuits. For this reason SPICE3 is an excellent framework for simulating different types of systems.

The accuracy of a simulator depends on the quality of models used for the devices. For this reason coupling of the electrical simulator with detailed physics based device simulators results in an accurate simulator. Furthermore, an interconnection of devices can be simulated in conjunction with the control electronics. This is an important aspect of coupling EOFLOW with SPICE3. The following subsections address the boundary conditions and the coupling algorithm.

3.3.1 Boundary conditions

The set of boundary conditions for an example of three interconnected cross channel microfluidic devices is shown in Figure 3.2.
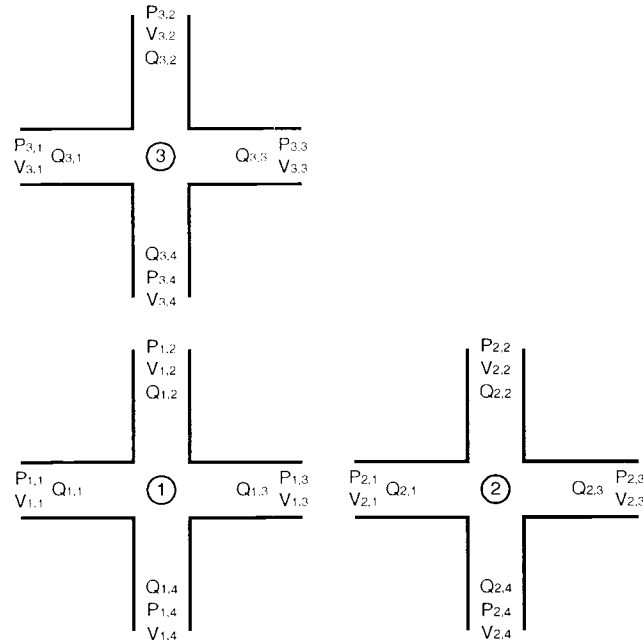
Figure 3.2 Boundary conditions that have to be matched for an interconnection of channels.

As can be observed from Figure 3.2 at each of the interconnection nodes the flow rates, voltages, and, fluid pressures of two devices should be identical. Referring to the example in Figure 3.2 the following conditions should be satisfied:

$$V_{1,3} = V_{2,1}, \quad V_{1,2} = V_{3,4}; \quad Q_{1,3} = Q_{2,1}, \quad Q_{1,2} = Q_{3,4}$$
$$P_{1,3} = P_{2,1}, \quad P_{1,2} = P_{3,4}$$

(3.1)

where V, Q and P are the potential at the in/outlet, the flow rate through the in/outlet and the pressure at the in/outlet, respectively. SPICE3 is used to force these device constraints during a simulation.

3.3.2 Coupling EOFLOW to SPICE3

EOFLOW is embedded in SPICE3 as a subroutine as shown in Figure 3.3. SPICE3 provides the electrical simulation engine and the framework for interconnecting several devices. The microfluidic devices are treated as a model in the SPICE3 framework.
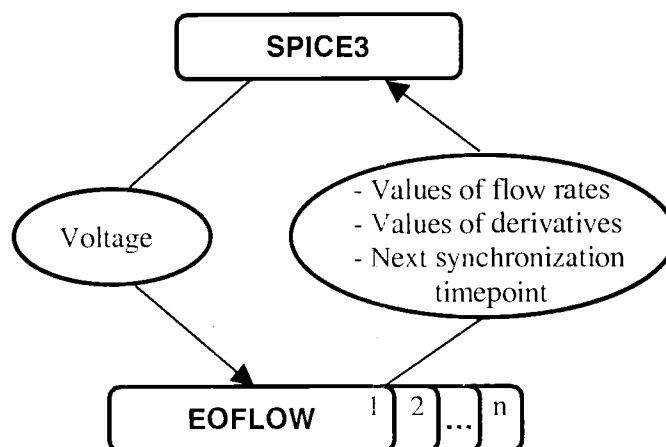
Figure 3.3 Simulator block diagram.

Each distributed model of a fluidic device calls EOFLOW. The time stepping scheme used has been described in [13]. Since EOFLOW runs with larger time steps than SPICE3, synchronization timepoints are determined by EOFLOW and transferred to the SPICE3 numerical engine. At every synchronized timepoint SPICE3 supplies the terminal voltages to EOFLOW and in return obtains the values of the derivatives of the flow rate through the ith outlet with respect to the voltage applied to the jth outlet. Symbolically, $g_{ij} = \partial Q_i / \partial V_j$, $i, j = 1, ..., 4$ (where Q_i, $i = 1, ..., 4$ is the flow rate through the ith outlet, and V_j, $j = 1, ..., 4$ is the voltage applied to the jth outlet) for every microfluidic element. The SPICE3

transient analysis lumped element representation of a cross channel is shown in Figure 3.4. The dependence of the flow rates on the applied voltages is modeled as voltage controlled current sources. The g_{ij} are the parameters for these controlled sources.
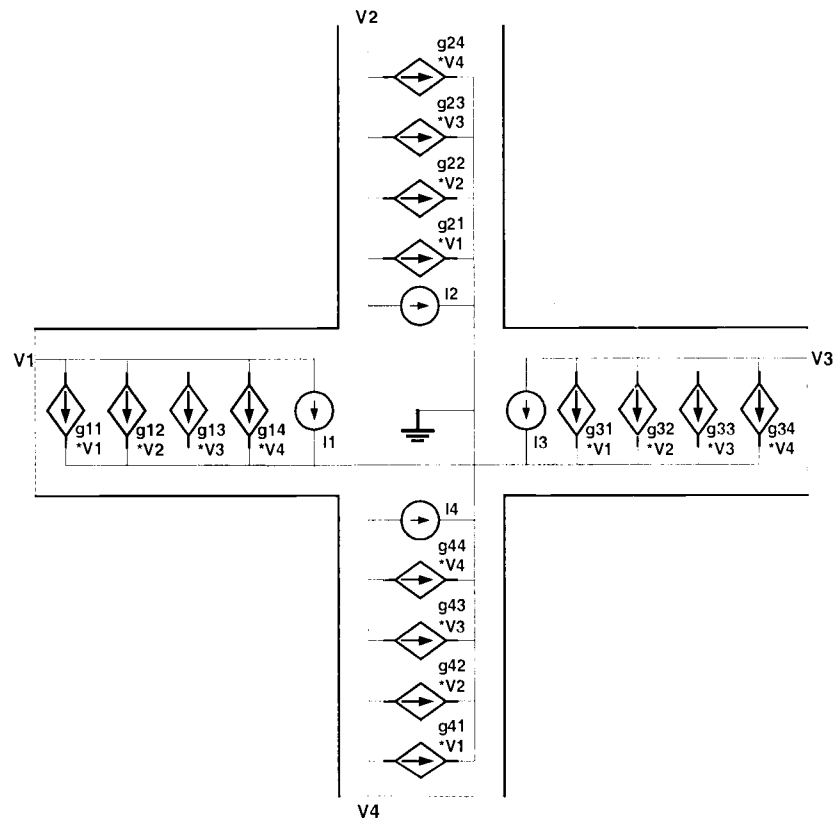
Figure 3.4 SPICE3 model for cross channel.

The interface with SPICE3 depends on the time integration algorithm used in the physical device solver. In this case, EOFLOW uses a semi-implicit time marching method. An implicit scheme is used for the viscous term in the Navier-Stokes equation [18], [23] and an explicit scheme is used for the convective term.

Because of this integration method a system of linear equations is solved at each time step instead of a system of nonlinear equations. As a result the values of the derivatives (g_{ij}, $i,j = 1,...,4$) are constant at a particular time point regardless of the applied voltages. Therefore, every iteration of the Newton method uses these values without the need for recalculations. Moreover, the values of the g_{ij} change slowly in time, which allows their reuse during time stepping. In every N time steps the values are recalculated and compared with the values that have been previously used. If the error is less than 0.1% the value of N is doubled. If the value of the error is between 0.1% and 1% the value of N is kept the same. Finally, if the value of error is more than 1%, the value of N is divided by two. This allows for SPICE3-EOFLOW coupling to be computationally efficient.

3.4 Simulation Validation

Before presenting the simulation results we validate the coupled simulator by using a standalone version of EOFLOW. These results are compared with the ones obtained from SPICE3-EOFLOW coupling. A simulation loop was created around EOFLOW using the bisection method to find the solution. This is a search method in which the solution is searched for iteratively. An example of how the bisection method is used is shown in Figure 3.5. Two initial points at x_1 and x_2 are chosen from which the zero crossing of the function $f(x)$ is obtained at x_6.

Figure 3.5 Bisection method convergence example.

Since the bisection method does not require any knowledge about derivatives, its implementation is very easy. However, the performance of this method is very slow and hence it is used only for verifying the results. The simulated structure is shown in Figure 3.6.

Figure 3.6 Reference simulation structure.

Different voltages were applied to the various outlets and the voltage at the interconnection node of the two cross channels was calculated. The resulting voltage, Vx, obtained from the SPICE3-EOFLOW coupled simulator and EOFLOW with a bisection loop is presented in Figure 3.7. It can be observed from Figure 3.7 that the simulation results are in good agreement. The trade-off of accuracy versus speed of simulation was also evaluated for this example. If at every time point the values of the derivatives and flow rates are recalculated, the two methods are in exact agreement but the simulation is time consuming. There is a difference at time zero because the two methods used different initial conditions. Reuse of the values of the flow rates and derivatives improves the speed of calculation significantly as shown in Table 3.1. However, there is a difference in the simulated results due to an error incurred by reusing the values of the flow rates and derivatives from previous time steps. This error can be controlled by changing the criterion for reuse of the previous time point values. The computational cost for each of these simulations is summarized in Table 3.1. It can be seen that the SPICE3-EOFLOW simulation without data reuse is approximately three times faster then the bisection method. With data reuse the computational time reduces by an order of magnitude.
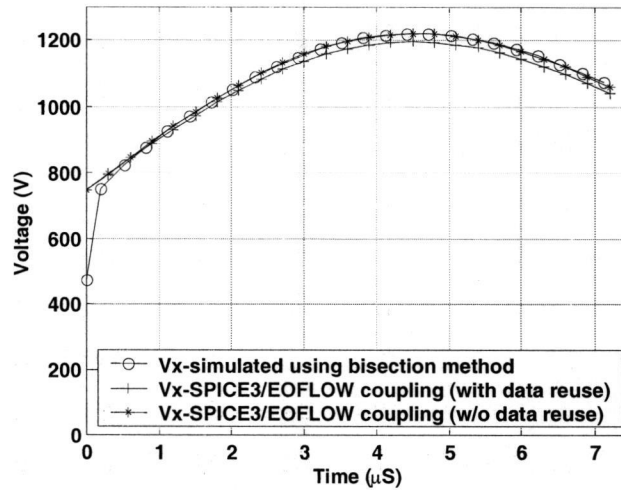
Figure 3.7 Voltage *Vx* obtained from different methods.

Table 3.1 Computational cost for different simulation methods.

Method	Time
Bisection	15 hours 52 min
SPICE3-EOFLOW (without data reuse)	5 hours 26 min
SPICE3-EOFLOW (with data reuse)	12 min 27 sec

3.5   Simulation Results

In this section, we apply the coupled SPICE3-EOFLOW simulator to different types of problems. We demonstrate the usefulness of the simulator in determining an appropriate flow control strategy. Also, a complex interconnection of channels is simulated to show the versatility of this simulator.

3.5.1 *Control systems for flow*

Microfluidic systems consisting of cross channel pipes and control electronics (Figures 3.8 and 3.9) were simulated. The control loop is intended to reduce the vertical velocity so that flow spreading can be avoided. Zero flow rates are obtained through the vertical channels by automatically adjusting the voltages applied to the vertical channels. The block diagram of the system with a proportional type controller is shown in Figure 3.8, whereas an integral type of controller is shown in Figure 3.9.
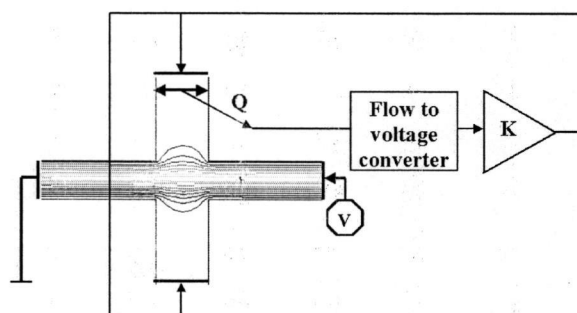
Figure 3.8 Block diagram of the proportional control system.
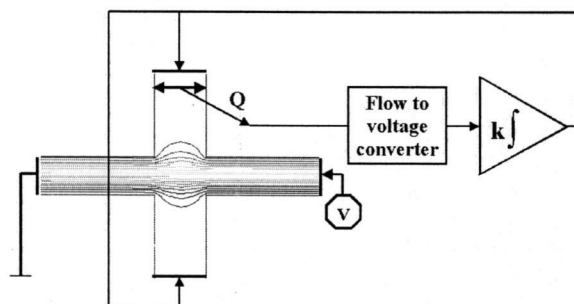
Figure 3.9 Block diagram of the proportional-integral control system.

The dynamic behavior of the controlled potentials and flow rates and the streamlines, for these systems are shown in Figures 3.10, 3.11, 3.12 and 3.13, respectively.
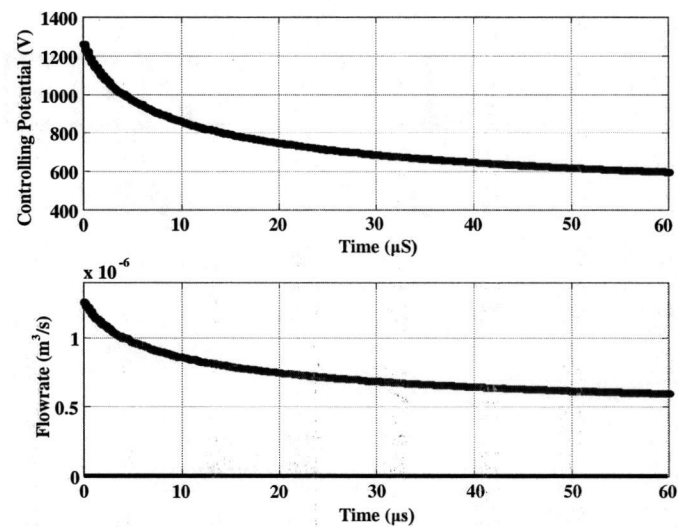
Figure 3.10 Dynamic behavior of controlled potentials and flow rate for the electroosmotic system with proportional control.
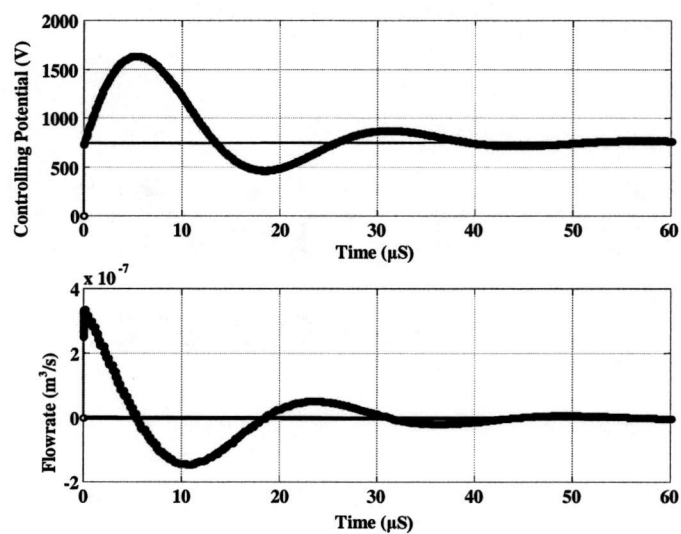
Figure 3.11 Dynamic behavior of controlled potentials and flow rate for the electroosmotic system with proportional-integral control.
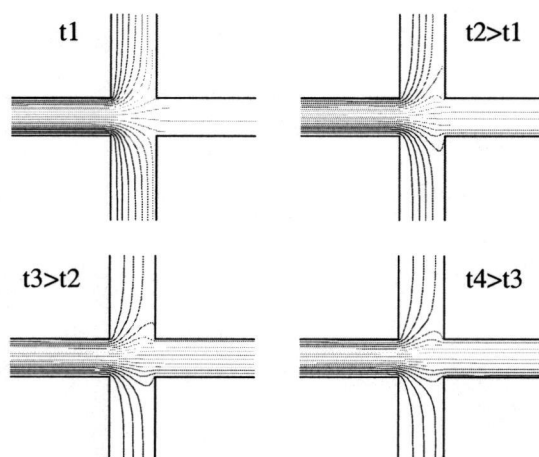
Figure 3.12 Dynamic behavior of streamlines for the electroosmotic system with proportional control.
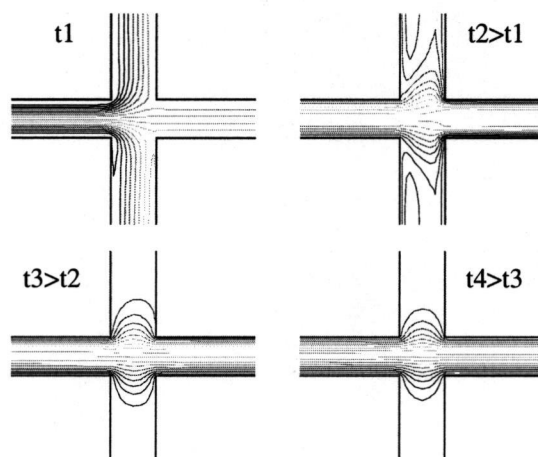
Figure 3.13 Dynamic behavior of streamlines for the electroosmotic system with proportional-integral control.

As shown in Figures 3.10 and 3.12, it is not possible to stabilize the flow rate at the given level in a reasonable amount of time when using a proportional control system. For an integral control system, the flow rate can be stabilized in less than $100\,\mu S$ as shown in Figures 3.11 and 3.13. Thus, an integral control system provides a better performance for this application.

3.5.2 Complex system simulation results

As an example of a complex system, consider a mesh of 10×10 interconnected cross channel pipes as shown in Figure 3.14. Voltage sources with different amplitudes and frequencies were connected to each outlet.

Point where potential has been checked

Top labels (left to right):
250 + 950*sin(2*pi*9.5e4)V
250 + 850*sin(2*pi*8.5e4)V
250 + 750*sin(2*pi*7.5e4)V
250 + 650*sin(2*pi*6.5e4)V
250 + 550*sin(2*pi*5.5e4)V
250 + 450*sin(2*pi*4.5e4)V
250 + 350*sin(2*pi*3.5e4)V
250 + 250*sin(2*pi*2.5e4)V
250 + 150*sin(2*pi*1.5e4)V
250 + 50*sin(2*pi*0.5e4)V

Left labels (top to bottom):
500+100*sin(2*pi*1e4)V
500+200*sin(2*pi*2e4)V
500+300*sin(2*pi*3e4)V
500+400*sin(2*pi*4e4)V
500+500*sin(2*pi*5e4)V
500+600*sin(2*pi*6e4)V
500+700*sin(2*pi*7e4)V
500+800*sin(2*pi*8e4)V
500+900*sin(2*pi*9e4)V
500+1e3*sin(2*pi*1e5)V

Right labels (top to bottom):
1e3*sin(2*pi*1e5)V
900*sin(2*pi*9e4)V
800*sin(2*pi*8e4)V
700*sin(2*pi*7e4)V
600*sin(2*pi*6e4)V
500*sin(2*pi*5e4)V
400*sin(2*pi*4e4)V
300*sin(2*pi*3e4)V
200*sin(2*pi*2e4)V
100*sin(2*pi*1e4)V

Bottom labels (left to right):
50*sin(2*pi*0.5e4)V
150*sin(2*pi*1.5e4)V
250*sin(2*pi*2.5e4)V
350*sin(2*pi*3.5e4)V
450*sin(2*pi*4.5e4)V
550*sin(2*pi*5.5e4)V
650*sin(2*pi*6.5e4)V
750*sin(2*pi*7.5e4)V
850*sin(2*pi*8.5e4)V
950*sin(2*pi*9.5e4)V

Figure 3.14 Simulation of a 10 x 10 cross channel mesh with voltages applied at the outlets.

This example shows that the voltage at any interconnection point can be readily simulated. Furthermore, the computational efficiency of the coupling algorithms when a large number of interconnected microfluidic devices need to be simulated is demonstrated. The response is shown at a point near the center of the mesh in Figure 3.15 for one period of the output waveform. Figure 3.16 shows the frequency-domain spectrum of the output waveform in Figure 3.15. The resulting

output signal contains harmonics corresponding to the input frequencies as expected.

Figure 3.15 One period of the output signal.

Figure 3.16 Spectrum of the output signal.

The computational time is an important issue for this type of simulation. One EOFLOW calculation for a device with 2370 nodes takes about one minute on a Sun Solaris workstation with an UltraSparc 440 MHz processor. At every time point each device requires four simulations for the cross channel to calculate the derivative values. For a transient simulation up to 300 μS each device is evaluated at approximately 1000 tim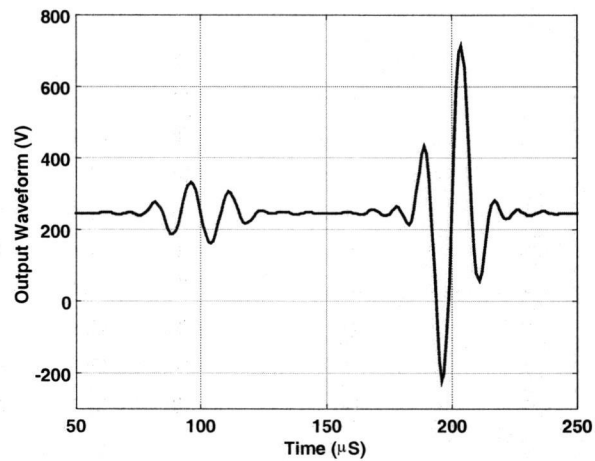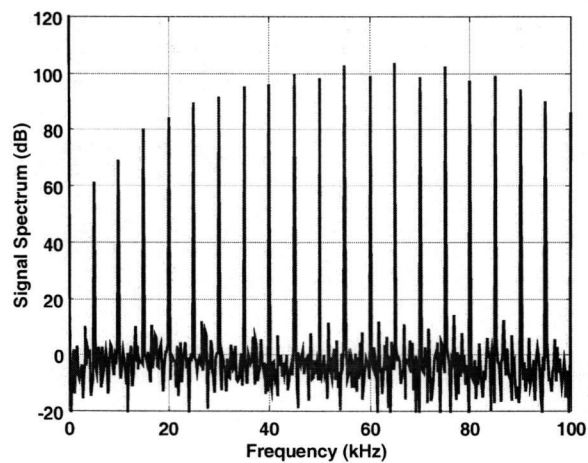e points. Based on this approximation the simulation would require 100*1000*4 = 400000 min $\approx$ 9 months. Taking into account that the derivatives and flow rates change slowly with time and reusing previously calculated values helps reduce this time to two days. The simulation time could be decreased further by evaluating the devices in parallel. In this case, the 100 devices can be simulated simultaneously on different machines, whereby the total simulation time can be reduced to a few hours.

3.6   Summary

A new approach for the simulation of electroosmotic systems has been developed. SPICE3 is combined with a flow solver EOFLOW to enable simulation of a complex network of interconnected channels. Furthermore, various electronic control options for flow can be evaluated. A system consisting of one hundred microfluidic elements was simulated to demonstrate the versatility of this simulator.

4   COUPLED SPICE3/PROPHET SIMULATOR

4.1   Introduction

As the frequency of operation of circuits increases, designers need to use distributed element models for accurate simulation. Lumped element models may not be adequate for the proper design of next generation radio frequency (RF) circuits. As the electromagnetic wave length becomes comparable to the component dimensions, the distributed nature of the devices becomes important. Therefore, the integration of device-level solvers into circuit simulators becomes critical.

PROPHET [24] is a recently developed tool for the solution of partial differential equations (PDEs) and can be used for simulations of IC devices and processes. PROPHET can be used in the same way as conventional technology computer-aided design (TCAD) tools are used [25]. The tool works in one, two, or three spatial dimensions of the simulation domain and is suitable for distributed element modeling. All model coefficients and material parameters are contained in a database library that can be modified by a user. Use of PROPHET as a regular SPICE3 device model allows for the accurate simulation of RF integrated circuits. PROPHET is a relatively new tool and, therefore, requires further improvement. Adaptive meshing and error control, efficient and robust numerical algorithms, calibration and material characterization are some of the PROPHET enhancements that will take place in the future.

A significant amount of research has already been done in the field of coupled device/circuit simulation [3], [12], [10], [26]. While developing tools and algorithms for coupled device/circuit simulations, the most common problem encountered is the computational cost. Usually devices are described by a large system of partial differential equations that is difficult to solve, resulting in a long simulation time even for small circuits. Moreover, during a coupled simulation, a

large number of devices can be used and the model evaluation routine for each of them is called several times. This makes the accurate simulation of such circuits almost impossible in a reasonable amount of time. Therefore, the development of an efficient algorithm for coupling can reduce computational cost and this continues to be a challenging research problem. Transient analysis is the most time consuming simulation, which suggests that more efficient coupling algorithms for transient analysis are essential. The periodic steady state (PSS) analysis methods are much faster than transient analysis for periodically driven circuits typically encountered in RF applications. The harmonic balance method is one such PSS analysis method. Therefore, the work in this chapter focuses on an implementation of the harmonic balance method for coupled device/circuit simulations. In addition, DC and AC analyses were also implemented.

4.2   Coupling Description

The node voltages required by PROPHET for numerical model evaluation are calculated by SPICE3 and passed to PROPHET by modifying the input file. Then SPICE3 calls PROPHET using the system command. PROPHET calculates currents through every node and conductances and capacitances between each pair of device nodes, which are needed for stamping a device into the SPICE3 circuit matrix. These parameters are passed to SPICE3 through a data file. SPICE3 calculates the node voltages at the next Newton iteration. Such a technique is common for coupling tasks and can be used for interfacing any device simulator with SPICE3. The SPICE3-HB-PROPHET coupling is shown in Figure 4.1. This technique can be used for all three analyses that have been implemented: DC, AC, and HB. For DC analysis, capacitances are not used. For AC analysis, the device is evaluated at the operating point voltages for different frequencies. For HB analysis, the device is evaluated for the same frequency but at different bias voltages. The

device evaluation procedure takes two steps. First, DC analysis is used to calculate the currents through all the terminals. Second, AC analysis is used to calculate the device small-signal parameters (small-signal conductances and capacitances). To improve the speed of the calculations, PROPHET uses a zero-frequency AC analysis as described in [27]. Although, capacitances are not used during DC analysis the values of conductances are still required. Therefore, a PROPHET AC analysis is used for all these analyses.
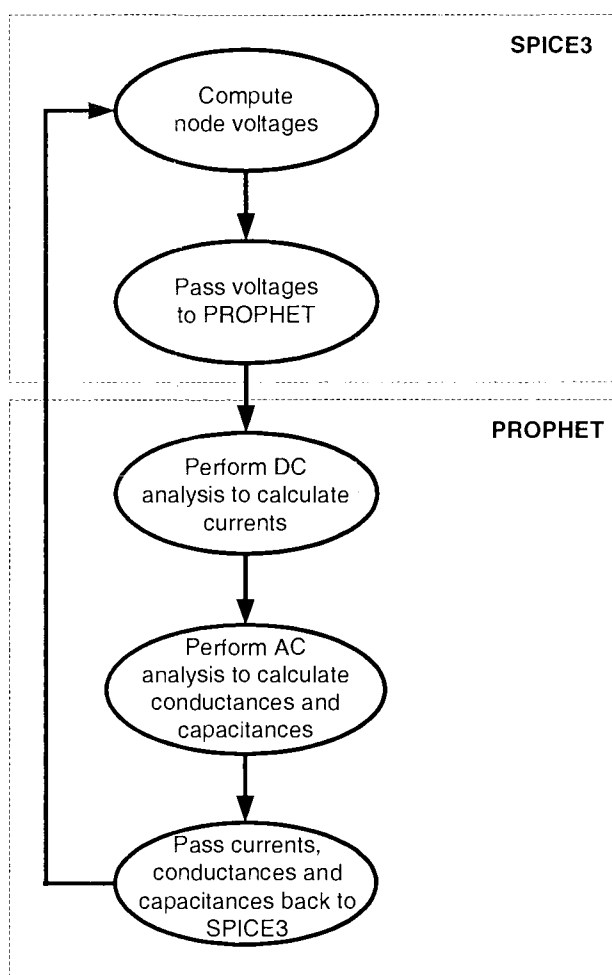

Figure 4.1 The coupling between SPICE3 and PROPHET.

4.3   Simulation Validation

4.3.1   DC simulation validation

To validate whether the numerical PROPHET model is implemented correctly it is necessary to compare the results from the SPICE3-HB-PROPHET coupled simulations with the results that are obtained in an alternate way. For DC analysis, it is possible to simulate circuits containing a transistor with resistors connected to the terminals within PROPHET. Therefore, the results for the same circuit can be compared from SPICE3-HB-PROPHET coupled simulation and from PROPHET directly.

A simple common-source amplifier shown in Figure 4.2 was chosen as a test circuit to validate the coupling algorithm. This simple circuit can also be simulated directly in PROPHET. The nonlinearity in this circuit makes it possible to check the Newton's loop in the SPICE3-HB-PROPHET coupled simulation. The transistor in the test circuit is described as a numerical device using the PROPHET input file.
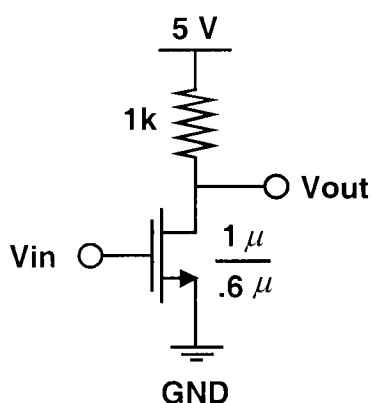
Figure 4.2 Common-source amplifier that was used for DC simulation validation.

The resulting input-output characteristic of the circuit in Figure 4.2 is shown in Figure 4.3. As can be seen the two transfer curves overlap which demonstrates that the coupling algorithm has been implemented correctly and gives the correct result. The gate voltage Vin was changed from 0 to 5 V with a step of 0.1 V and a total of fifty DC points were evaluated. During SPICE3-HB-PROPHET DC analysis, the option to restart from the previous condition was used for device simulation. The device has four terminals with 1116 nodes resulting in an average time of 25 sec for one device evaluation. In average, 2.4 device evaluations were necessary per DC point, which resulted in a total number of 120 device evaluations and a total computational time of around 50 minutes. On the other hand, the total time for direct PROPHET simulation was about 10 min with 50 iterations on a Sun Solaris workstation with an UltraSparc 440 MHz processor. This results in average of 12 sec for one DC point evaluation, unlike the previous 25 sec, because an AC calculation is not required for direct PROPHET DC simulation.
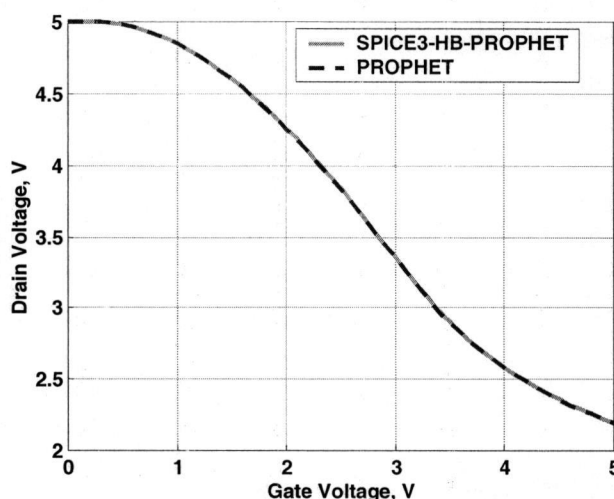
Figure 4.3 Input-output transfer characteristic for the common-source amplifier from the SPICE3-HB-PROPHET coupled simulation and direct PROPHET simulation.

4.3.2 HB simulation validation

HB simulation can be performed using the CODECS-HB [10] simulator. CODECS-HB is essentially SPICE3, extended to perform frequency-domain steady-state simulations with 2D numerical device models for diodes, BJTs and MOSFETs. The best way to verify a harmonic balance simulation result is to compare it with a transient simulation result for a test circuit. PROPHET has the capability to do a transient analysis for an entire circuit including PROPHET device descriptions and a SPICE3 circuit description. A full-Newton solution approach is used. A comparison can be made between the coupled device/circuit HB simulation results and those obtained from PROPHET directly. The common-source amplifier shown in Figure 4.4 was chosen as a test circuit. The resulting output waveform for the circuit in Figure 4.4 is shown in Figure 4.5.
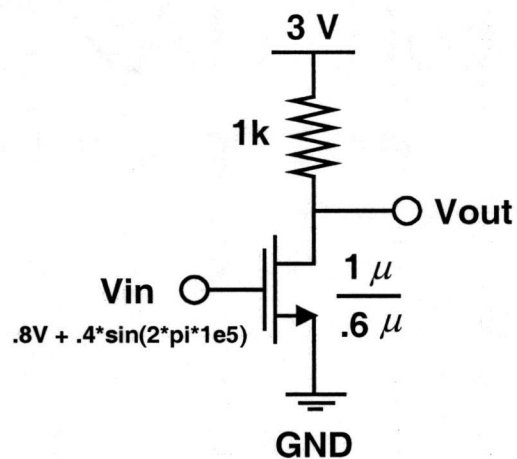
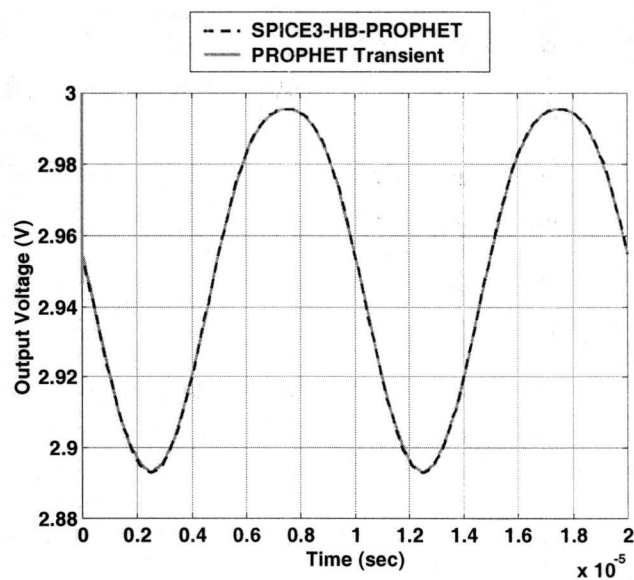Figure 4.4 Common-source amplifier that was used for HB simulation validation.

Figure 4.5 Output waveform obtained from the direct transient simulation in PROPHET and the harmonic balance method in SPICE3-HB-PROPHET.

As can be seen from Figure 4.5 the two waveforms overlap which demonstrates that the coupling algorithm has been correctly implemented.

4.4   Simulation Results

4.4.1   DC simulation results

The simple inverter circuit, shown in Figure 4.6, was simulated in SPICE3-HB-PROPHET and directly in PROPHET to demonstrate the simulation of multiple coupled devices.
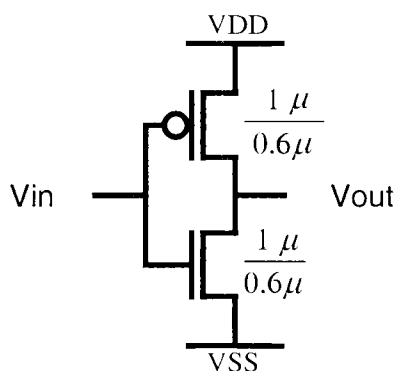
Figure 4.6 Schematic of a test inverter circuit.

The resulting input-output characteristic is shown in Figure 4.7. As can be seen from Figure 4.7 the two curves overlap as in Figure 4.5 which again demonstrates that the coupling algorithm has been correctly implemented. Although. a limiting routine was not used. the simulation of this circuit did not exhibit any convergence problems because PROPHET was able to converge even for unreasonable terminal voltages (up to 20 V).

However, a voltage limiting routine should always be used. The voltage during a Newton iteration can be large such that the device does not converge. In such a case voltage limiting is essential. The primary problem with implementing a limiting scheme is that PROPHET has been implemented in SPICE3 as a general device. This general device can have anywhere from two to ten nodes (the number of nodes can be further extended if necessary) and can have a p-n junction between any two nodes. Because of this, a routine needs to be implemented in PROPHET to provide SPICE3 with the information on exactly which voltages should be limited. Therefore, when this becomes necessary, it would be easier to make modifications at the PROPHET level rather than in SPICE3.
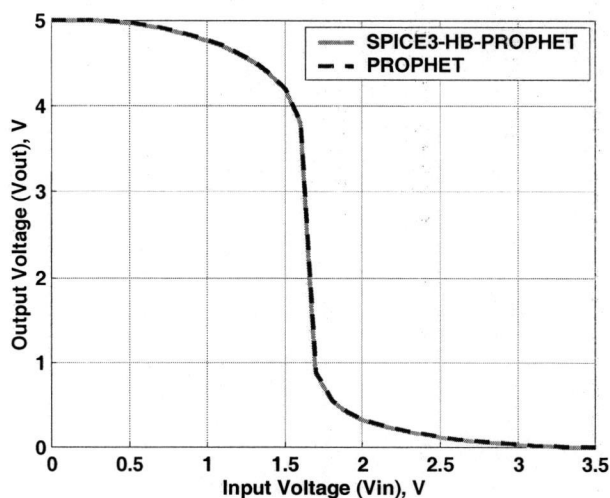
Figure 4.7 Inverter input-output transfer characteristic.

4.4.2   HB simulations results

The harmonic balance method, which is part of the combined SPICE3-HB-PROPHET simulator, is different for amplifier, mixer and oscillator circuits. The HB simulation results for these types of circuits are summarized below.

4.4.2.1   Amplifier simulation results

First, the harmonic balance method was employed for the simple common-source (CS) amplifier circuit shown in Figure 4.8. The time-domain result for this simulation is shown in Figure 4.9.

Figure 4.8 Common-source amplifier circuit.

SPICE3-HB-PROPHET Result
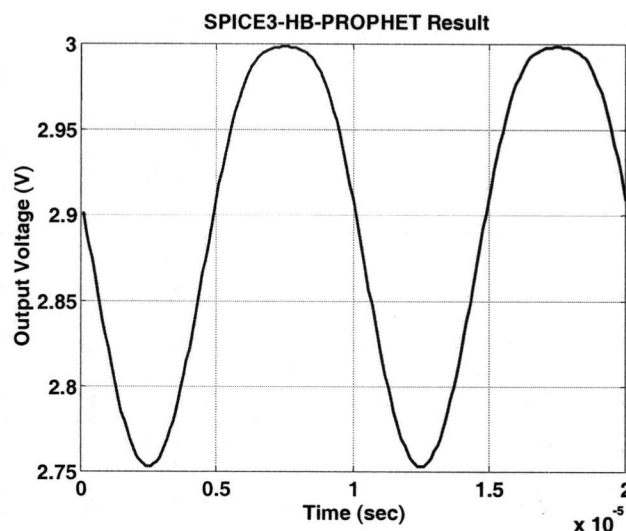
Figure 4.9 CS amplifier output voltage in the time-domain.

As can be seen from Figure 4.9, the upper portions of the sine wave show a flattening. This distortion of the waveform can be explained if one considers the frequency-domain representation of the output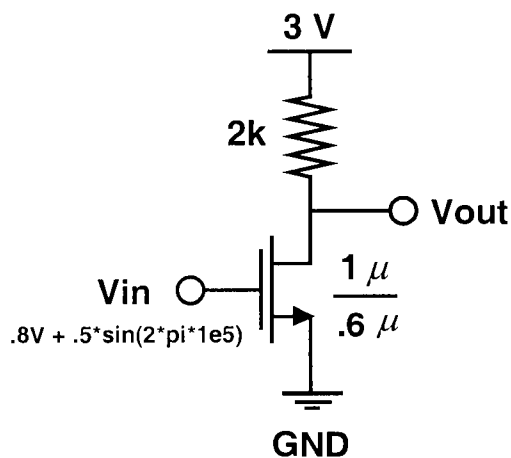 voltage as shown in Figure 4.10. Since the aim of the experiment was to see how the circuit solution converges in the presence of considerable harmonics, the MOSFET is biased with a gate voltage of only 0.8V. This is insufficient to strongly invert the transistor and therefore it operates in a region bordering the strong and weak inversion regimes. There is considerable distortion in the output, as seen in the large number of harmonics (both even and odd) in Figure 4.10. In addition, from the DC transfer characteristics of an inverter, it is evident that when a circuit is biased with a low input DC voltage, the output tends to approach the positive supply rail. This effect is reflected in the clipping and flattening of the upper portions of the output.
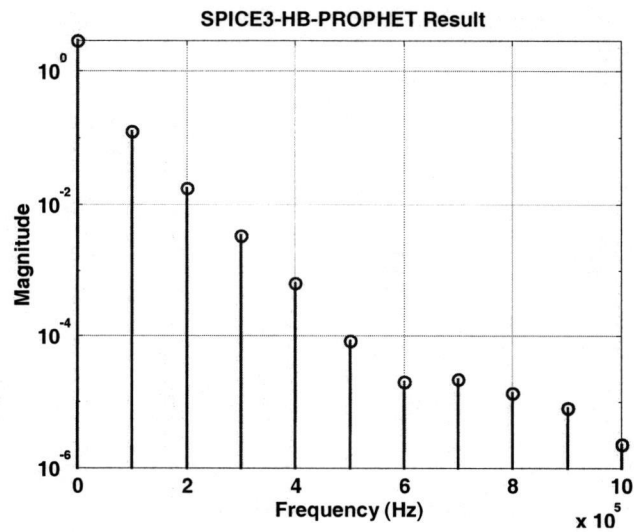
Figure 4.10 Frequency domain representation of the CS amplifier output signal.

4.4.2.2 Mixer simulation results

The modified common-source amplifier circuit shown in Figure 4.11 was used to check the mixing behavior in the presence of two input signals and investigate convergence of the harmonic balance analysis for a mixer. The result of this simulation is shown in Figure 4.12.
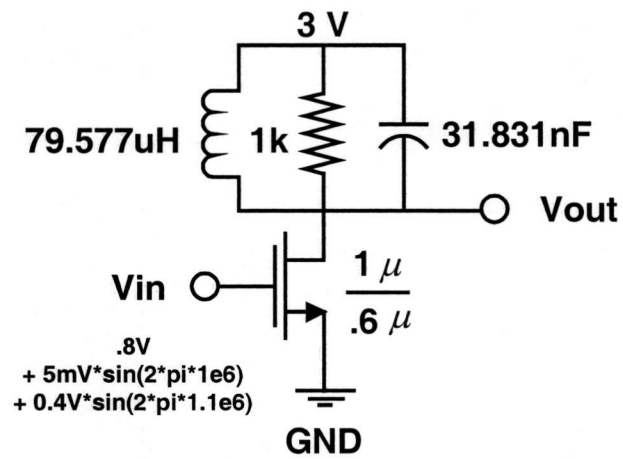
Figure 4.11 Common-source mixer circuit.
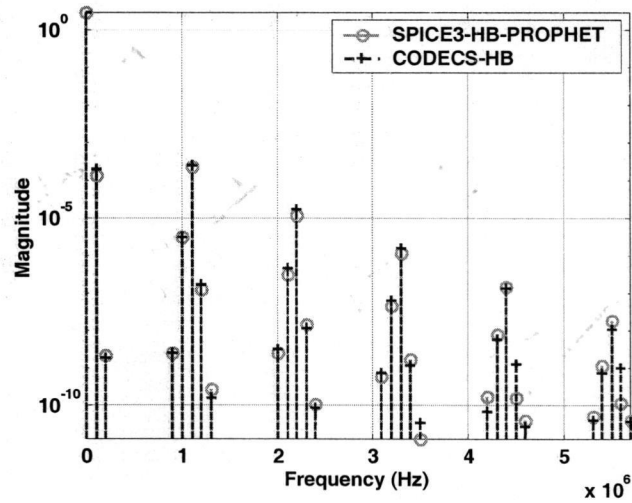
Figure 4.12 Frequency-domain representation of the common-source mixer output voltage.

As can be seen from Figure 4.12 the tested circuit exhibits a significant mixing behavior. Had there been no mixing, the output frequency spectrum would have shown only the harmonics of each tone. However, the intermodulation terms that appear on either side of each harmonic are a direct result of this mixing. On closer inspection of the intermodulation side bands, it must be noted that they are not of the same magnitude as expected. This is due to the fact that the harmonics are not at the resonance frequency of the tank. This results in a filtering action that translates to side-bands of unequal magnitude. To validate these results the same circuit was simulated in the CODECS-HB simulator using HB analysis. As shown in Figure 4.12 the results match each other taking into consideration the differences in the physical model descriptions for CODECS-HB and PROPHET.

Both circuits, Figure 4.8 and Figure 4.11, use transistors described as numerical models in PROPHET (Appendix A) as shown in Figure 4.13.
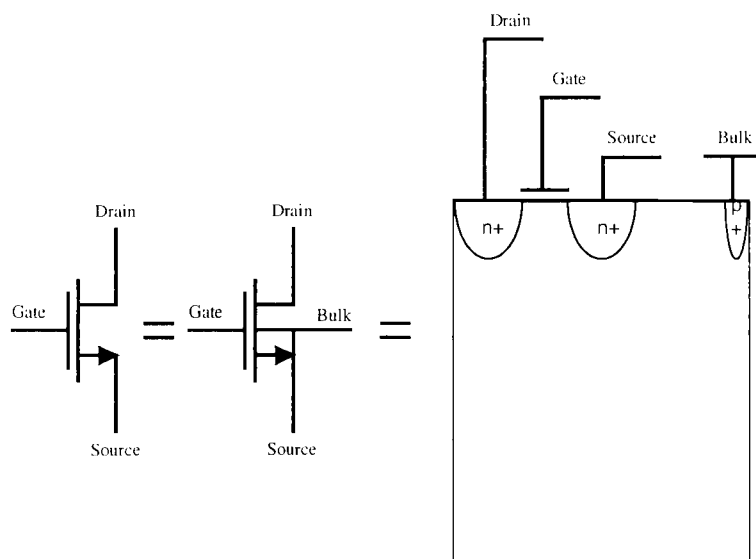
Figure 4.13 PROPHET transistor structure used in simulations.

To investigate mixing for a more complicated problem, two-device structures (Appendix C) connected in separate common-source configurations (Figure 4.14) were created. This is typical of a substrate-coupling problem.
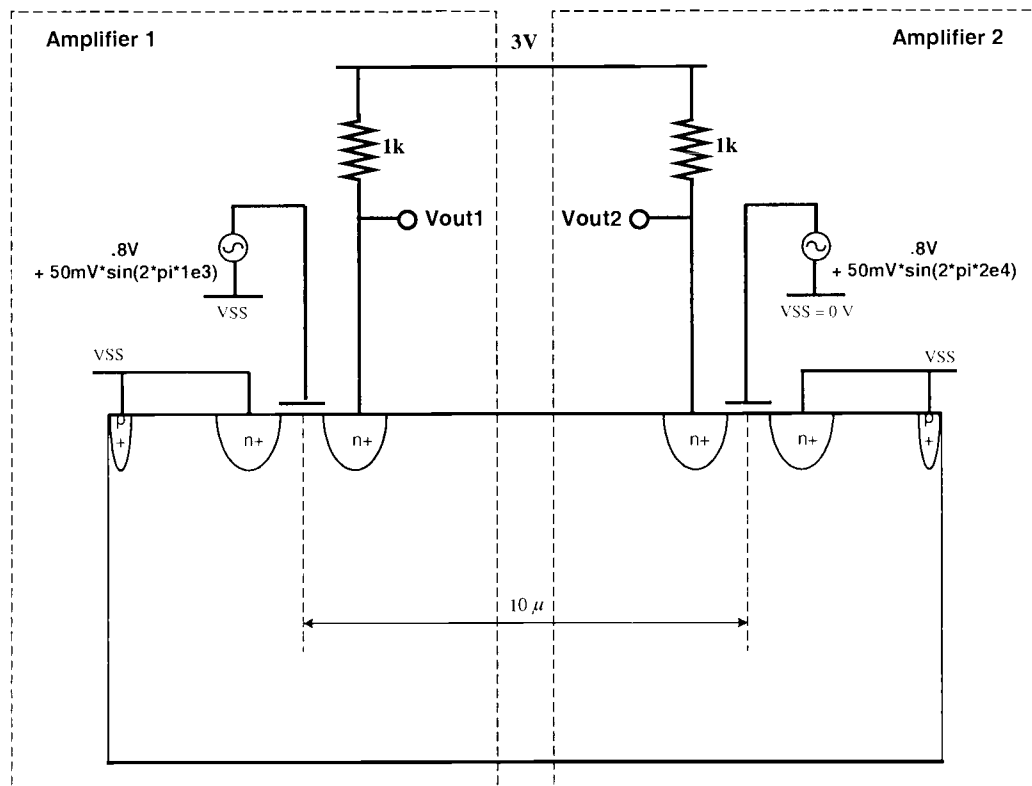
Figure 4.14 Two common-source amplifier structure.

As can be observed from the results of the simulation presented in Figure 4.15 and Figure 4.16 the separation of 10μm between the two structures and a 0.05V input signal does not result in significant mixing. Five harmonics for each input signal were assumed which resulted in 121 timepoints for one HB iteration. Depending on the bias conditions this circuit converges in five to eight HB

iterations. For certain bias voltages, however, the HB analysis does not converge. The possible reason for this might be that the HB algorithm implemented in CODECS-HB is not very robust. Sometimes it does not converge even when analytical models are used. Based on the discussion with Yutao Hu [28], CODECS-HB does not use any kind of continuation or homotopy method for enhancing the convergence of the Newton method. The implementation of such methods could solve the convergence problem for harmonic balance method in CODECS-HB.

The calculation of conductances and capacitances required by harmonic balance method is done using a zero frequency AC analysis in PROPHET. One such analysis for the eight terminal device with 5180 nodes takes about seven minutes on a Sun Solaris workstation with an UltraSparc 440 MHz processor. Taking into account the time for the DC analysis performed before each AC analysis, the net time for a complete AC simulation might vary anywhere from eight to ten minutes. Assuming that eight harmonic balance iterations are required for obtaining the solution and that the initial DC analysis for the circuit requires ten Newton iterations, the total simulation time = 10*10+121*10*8 = 9780 min = 163 hours $\approx$ one week. One way to decrease the computational time is to reduce the number of nodes in the device. Reducing the number of nodes in the PROPHET device to 2520 results in an approximate time of three minutes for one complete AC simulation thus reducing the total simulation time to around two days. But reducing the number of device nodes results in a loss of accuracy and is, therefore, not recommended. Another possible technique to decrease computational time would be to improve PROPHET's AC analysis. The AC simulation in another device simulator MEDICI [29] for a similar device with 5180 nodes takes about fifteen times less computational time than PROPHET. If the AC analysis of PROPHET can be made as efficient as that of MEDICI, then the simulation would take about half a day rather than a week. MEDICI was, however, not used in this work because it is only a two-dimensional simulator and also because its DC analysis is not as robust as that of PROPHET.
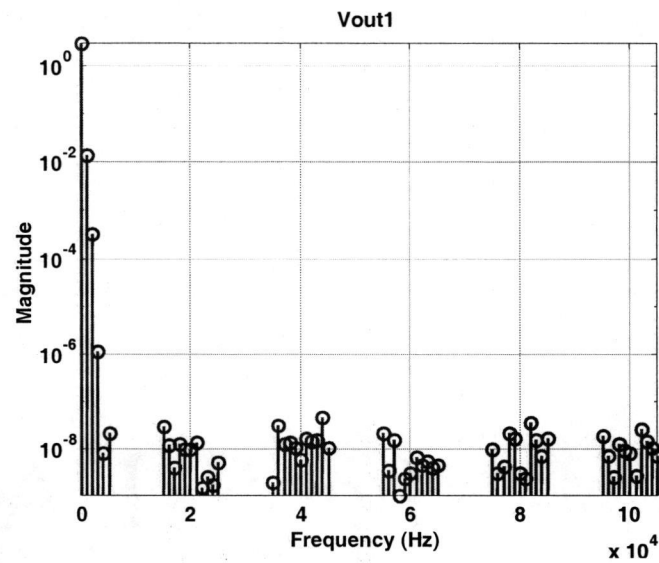
Figure 4.15 Frequency-domain spectrum of the first CS amplifier output (Vout1).
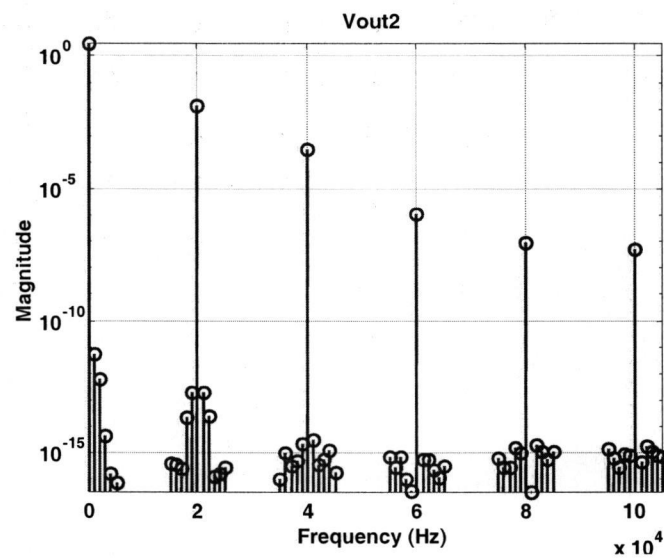
Figure 4.16 Frequency domain spectrum of the second CS amplifier output (Vout2).

4.4.2.3   Oscillator simulation results

A ring oscillator circuit shown in Figure 4.17 was simulated to explore convergence of the harmonic balance method for oscillators.
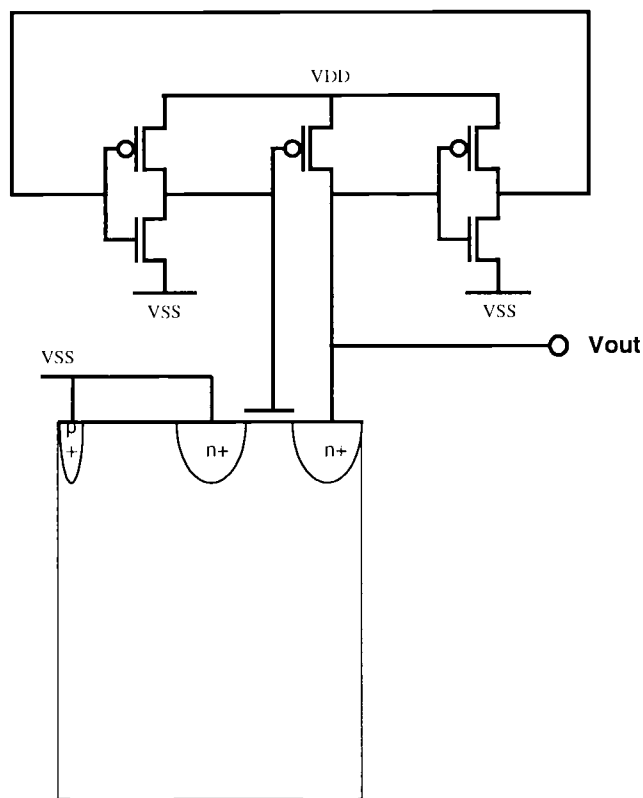
Figure 4.17 Ring oscillator circuit.

The transient analysis has convergence problems for this circuit in PROPHET. Therefore, it is not possible to compare harmonic balance analysis results with those from a transient analysis for this circuit. Due to this reason, a harmonic balance simulation in SPICE3-HB-PROPHET for the oscillator circuit

shown in Figure 4.17 was compared with harmonic balance simulation and transient simulation in CODECS-HB. For this purpose, the same transistor structure was created in both PROPHET and CODECS-HB (Appendix A and Appendix B, respectively). The results of these simulations are grouped in the Figure 4.18 and Figure 4.19 for easy comparison.
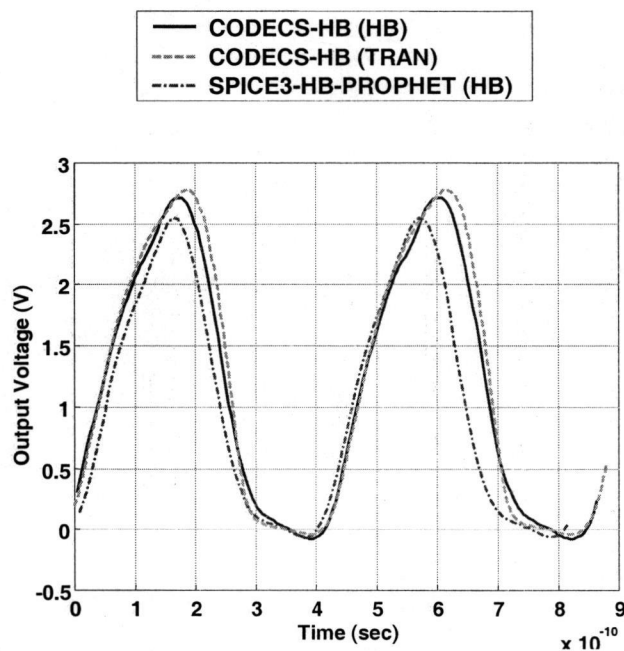
Figure 4.18 Oscillator output voltage waveform.
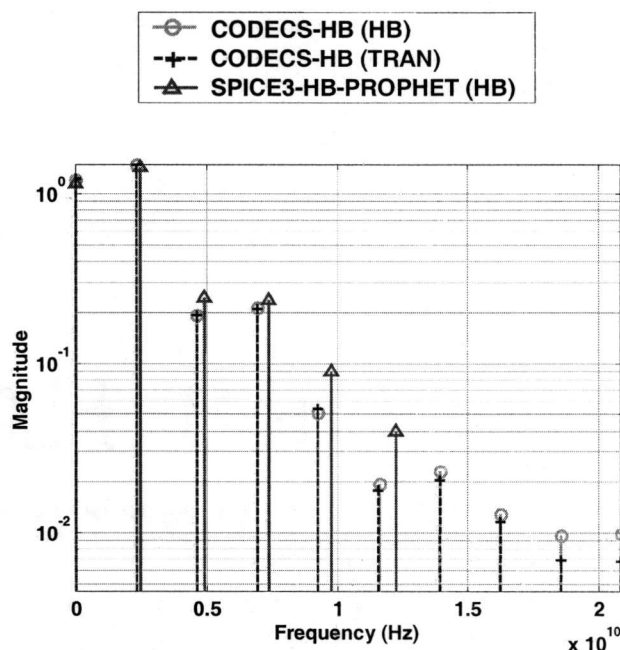
Figure 4.19 Oscillator output voltage in the frequency domain.

As can be observed from simulation results shown in Figures 4.18 – 4.19 there are some differences. The difference between CODECS-HB results obtained from transient and harmonic balance analyses can be explained due to the fact that harmonic balance analysis implemented in CODECS-HB is a quasi-static analysis. Since the f_T of the transistor is only 20 GHz, this results in an error in the calculation of high frequency harmonics. The difference between CODECS-HB and SPICE3-HB-PROPHET results can be explained due to the differences in the device and model descriptions for CODECS-HB and PROPHET.

Table 4.1 summarizes the computational time for some of the simulations described in this chapter. All the simulations were performed on a Sun Solaris workstation with an UltraSparc 440 MHz processor.

Table 4.1 Summary of computational time for different simulations.

Circuit	Simulator	Analysis	
		HB	Transient
Amplifier	PROPHET	N/A	18 min
	SPICE3-HB-PROPHET	29 min	N/A
Mixer	PROPHET	N/A	60 hours 6 min
	SPICE3-HB-PROPHET	14 hours 4 min	N/A
	CODECS-HB	4 hours 25 min	258 hours
Oscillator	SPICE3-HB-PROPHET	58 hours	N/A
	CODECS-HB	48 hours 50 min	10 hours 39 min

4.5 Summary

PROPHET, a computer program for the solution of sets of partial differential equations, was integrated in SPICE3 as a new device model. The designer can perform DC, AC and HB analyses for circuits containing PROPHET devices described as numerical models. This capability is necessary as the frequencies used in circuits make the wavelength comparable to the physical size of the circuit elements and distributed effects become important.

The common-source amplifier circuit with the transistor described by its numerical model was simulated in the SPICE3-HB-PROPHET coupled simulator and then directly in PROPHET for verification of the method. DC and HB analyses were verified. DC analysis results from SPICE3-HB-PROPHET and PROPHET were compared with each other while HB results from SPICE3-HB-PROPHET were compared with the PROPHET transient analysis. In both cases, very close agreement was achieved which demonstrates that the coupling algorithm has been correctly implemented.

After this. the application of HB analysis was checked for three different types of circuits: amplifiers. mixers and oscillators. The influence of substrate coupling through the silicon substrate was simulated as an example of a mixer circuit. The results for a 10 μm separation do not show significant mixing. The results of HB analysis for an oscillator circuit were also compared with results from HB and transient analyses using CODECS-HB simulator.

5   CONCLUSIONS

Two simulators, EOFLOW and PROPHET, have been integrated into the generic framework of SPICE3. The new enhanced SPICE3 now has a capability to simulate systems that include two new device types.

EOFLOW integrated with SPICE3 allows efficient simulations of complex microfluidic systems containing control circuitry. Such a simulator provides accurate simulation of a complex system of interconnected channels. In addition, different types of control systems can be investigated for their performance.

The integration of PROPHET into SPICE3 enables accurate simulations of RF circuits. The incorporation of the HB method in this coupled simulator also allows for prediction of substrate coupling and its mixing with the signal of interest for analog and RF applications. Additionally, simulation of different circuit characteristics such as distortion, power, frequency, noise and transfer functions can be performed.

Future work should focus on adding other devices to SPICE3 or enabling other analyses such as pole-zero, Fourier, and sensitivity. The improvement of convergence in HB analysis in SPICE3-HB-PROPHET is essential to make the coupled simulator more robust. New coupling algorithms also need to be developed for additional performance improvements.

BIBLIOGRAPHY

[1] W. L. Engl, R. Laur, and H. K. Dirks, "MEDUSA – a simulator for modular circuits," *IEEE Trans. Computer-Aided Design*, vol. 1, pp. 85-93, Apr. 1982.

[2] K. Mayaram, "CODECS: A mixed-level circuit and device simulator," *Memo No. UCB/ERL M88/77*, Electronics Research Laboratory, University of California, Berkeley, Nov. 1988.

[3] K. Mayaram and D. O. Pederson, "CODECS: A mixed-level device and circuit simulator," *IEEE Int. Conf. Computer-Aided Design*, pp. 112-115, Nov. 1988.

[4] T. L. Quarles, "The SPICE3 implementation guide," *Memo No. UCB/ERL M89/44*, Electronics Research Laboratory, University of California, Berkeley, Apr. 1989.

[5] K. Mayaram, J. Chern, and P. Yang, "Algorithms for transient three-dimensional mixed-level circuit and device simulation," *IEEE Trans. Computer-Aided Design*, vol. 12, pp. 1726-1733, Nov. 1993.

[6] J. H. Chern, J. T. Maeda, L. A. Arledge, and P. Yang, "SIERRA: a 3-D device simulator for reliability modeling," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 516-527, May 1989.

[7] K. Mayaram, P. Yang, and J. Chern, "Transient three-dimensional mixed-level circuit and device simulation: algorithms and applications," *IEEE Int. Conf. on Computer-Aided Design*, pp. 112-115, Nov. 1991.

[8] K. S. Kundert, "Introduction to RF simulation and its application," *IEEE Journal of Solid-State Circuits*, vol. 34, pp. 1298-1319, Sept. 1999.

[9] Y. Hu and K. Mayaram, "Periodic steady-state analysis for coupled device and circuit simulation," *Proc. Int. Conf. Simulation of Semiconductor Processes and Devices (SISPAD)*, pp. 90-93, Sept. 2000.

[10] Y. Hu and K. Mayaram, "Harmonic balance analysis for coupled device and circuit simulation," *Radio and Wireless Conference*, pp. 137-140, Aug. 2001.

[11] Y. Hu and K. Mayaram, "An efficient algorithm for large-signal frequency domain coupled device and circuit simulation," *ISCAS 2002*, May 2002.

[12] K. Mayaram and D. O. Pederson, "Coupling algorithms for mixed-level circuit and device simulation," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 1003-1012, Aug. 1992.

[13] R. M. Kirby, G. E. Karniadakis, O. Mikulchenko, and K. Mayaram, "An integrated simulator for coupled domain problems in MEMS," *IEEE/ASME Journal of Microelectromechanical Systems*, vol. 10, pp. 379-391, Sept. 2001.

[14] J. S. Wang, D. F. Ostergaard, "A finite element-electric circuit coupled simulation method for piezoelectric transducer," *Proc. 1999 IEEE Ultrasonics Symp.*, vol. 2, pp. 1105-1108, 1999.

[15] A. Schroth, T. Blochwitz, G. Gerlach, "Simulation of a complex sensor system using coupled simulation programs," *The 8th Int. Conf. on Solid-State Sensors and Actuators and Eurosensors IX*, vol. 2, pp. 33-35, 1995.

[16] R. M. Kirby, T. C. Warburton, S. J. Sherwin, A. Beskok and G. E. Karniadakis, "The Nektar code: Dynamic simulations without remeshing", *Proc. of the 2nd Int. Conf. on Computational Technologies for Fluid/Thermal/Chemical Systems with Industrial Applications*, Aug. 1999.

[17] D. W. Yergeau, Private Communication, 2002.

[18] M. J. Mitchell, R. Qiao, and N. R. Aluru, "Meshless analysis of steady-state electro-osmotic transport," *IEEE/ASME Journal of Microelectromechanical Systems*, vol. 9, pp. 435-449, Dec. 2000.

[19] F. M. Rotella, "Mixed circuit and device simulation for analysis, design, and optimization of opto-electronic, radio frequency, and high speed semiconductor devices," Ph.D. Dissertation, Stanford University, Apr. 2000.

[20] S. F. Brown, "Good-bye, test tubes. Hello, labs-on-a-chip," *Fortune*, October 11, 1999.

[21] A. Rasmussen and M. E. Zaghloul, "In the flow with MEMS," *IEEE Circuits and Devices Magazine*, vol. 14, pp. 12-25, July 1998.

[22] A. T. Woolley, G. F. Sensabaugh, and R. A. Mathies, "High-speed DNA genotyping using microfabricated capillary array electrophoresis chips," *Anal. Chem.*, vol. 69, pp. 2181-2186, June 1997.

[23] R. Qiao and N. R. Aluru, "Mixed-domain and reduced-order modeling of electroosmotic transport in bio-MEMS," *Proc. 2000 IEEE/ACM Int. Workshop on Behavioral Modeling and Simulation*, pp. 51-56. Oct. 2000.

[24] http://www-tcad.stanford.edu/~prophet/

[25] R. W. Dutton, E. C. Kan, D. W. Yergeau, Z. Yu, and C. Rafferty, "Next-generation TCAD tools--supporting rapid prototyping of new models and numerics," *1997 NASA Device Modeling Workshop*, Aug. 1997.

[26] L. M. Silveira, M. Kamon, and J. White, "Algorithms for coupled transient simulation of circuits and complicated 3-D packaging," *IEEE Trans. Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging*, vol. 18, pp. 92-98, Feb. 1995.

[27] Z. Yu, D. Chen, L. So, R. W. Dutton, S. Beebe, R. J. G. Goossens, and F. Rotella, "PISCES-2ET and its application subsystems," Integrated Circuits Laboratory, Stanford University, 1994. (http://gloworm.stanford.edu/tcad/programs/pisces_2et_and_apps.pdf)

[28] Y. Hu, Private Communication, 2002.

[29] MEDICI, Version 2000.2.0, Avant! Corporation, 2000.

APPENDICES

APPENDIX A 0.5μm process transistor structure used for PROPHET

include(silicon_poisson)

include(silicon_dd)

\#

\# MOSFET (NMOS) with 0.5u process

\#

grid xloc=0.000,0.010,0.05,0.11,0.15,0.22,0.32,1.0,2.0,3.0,7.0

+   xdel=0.002,0.002,0.04,0.04,0.02,0.02,0.10,0.4,0.4,1.0,1.0

+   yloc=-2.5,-2.0,-1.34,-0.66,0.00,1.05,1.45,2.05,2.45,3.50

+   ydel=0.25,0.25,0.660,0.660,0.40,0.40,0.06,0.06,0.40,0.40

+   dim=2

field set=net1   val="-1e17*gbox(X,0,0,0.3)*gbox(Y,-2.5,3.5,0.05)"

field set=net2   val="-1e18*gbox(X,6,7,1.5)*gbox(Y,-2.5,3.5,0.05)"

field set=ns1   val="2e20*gbox(X,0,0,0.07)*gbox(Y,0,1.46,0.0003)"

field set=nd1   val="2e20*gbox(X,0,0,0.07)*gbox(Y,2.04,3.5,0.0003)"

field set=nb1   val="-1e20*gbox(X,0,0,0.074)*gbox(Y,-2.5,-2,0.0003)"

field set=netdope val="-5e14+ns1+nd1+net1+net2+nb1"

field set=netdope val=0 material=oxide

deposit mat=oxide start=1.45 end=2.05 thick=0.01 xdel=0.005

\#source1

boundary xmin=0   xmax=0   ymin=0   ymax=1.0   name=C3

\#drain1

boundary xmin=0   xmax=0   ymin=2.5 ymax=3.5   name=C1

```
#gate1
boundary xmin=-0.01   xmax=-0.01    ymin=1.45 ymax=2.05  name=C2
#bulk1
boundary xmin=0       xmax=0        ymin=-2.5 ymax=-2   name=C4


field set=tl val=300
field set=edge val=1


# start it up
bias initial system=silicon_poisson temper=27


# resolve with 2carriers
bias system=silicon_dd
```

APPENDIX B 0.5μm process transistor structure used for CODECS

* MOSFET model for substrate coupling (Gaussian doping)

* y-axis mesh is changed: the fine mesh between 0 and 0.01 is moved.

* x-axis uses larger mesh

.model mod3 numos

+ concmob fieldmob

+ xmesh 1 0

+ xmesh 8 1.45

+ xmesh 10 1.47

+ xmesh 20 2.03

+ xmesh 22 2.05

+ xmesh 29 3.5

+ xmesh 39 5.5

+ xmesh 44 6.5

+ ymesh 1 -0.01

+ ymesh 4 0

+ ymesh 6 0.02

+ ymesh 12 0.14

+ ymesh 22 0.24

+ ymesh 27 0.8

+ ymesh 33 2.0

+ ymesh 43 6.0

+ unif -5e14 0 6.5 0 6

+ gimp -1e17 0.3 0 0.167 1 0 6.5 0 0

+ gimp -1e18 1.5 6 0.033 1 0 6.5 6 6

+ gimp 2e20 0.07 0 0.0043 1 0 1.46 0 0

+ gimp 2e20 0.07 0 0.0043 1 2.04 3.5 0 0

+ gimp -1e20 0.074 0 1.0 1 5.5 6.5 0 0

+ oxide 8 22 1 4

+ silicon 1 44 4 43

+ contact 24 29 4 4

+ contact 8 22 1 1

+ contact 1 6 4 4

+ contact 40 43 4 4

APPENDIX C 0.5μm process two transistor structure used for PROPHET

```
include(silicon_poisson)
include(silicon_dd)
#
# two mosfets (NMOS) with 0.5u process
#
grid xloc=0.000,0.010,0.05,0.11,0.15,0.22,0.32,1.0,2.0,3.0,7.0
+    xdel=0.002,0.002,0.04,0.04,0.02,0.02,0.10,0.4,0.4,1.0,1.0
+ yloc=-2.5,-2.0,-1.34,-0.66,0.00,1.05,1.45,2.05,2.45,3.50,4.5,9.0,10.0,11.05,11.45,
12.05,12.45,13.5,14.16,14.84,15.5,16.0
+     ydel=0.25,0.25,0.660,0.660,0.40,0.40,0.06,0.06,0.40,0.40,1.0,1.0,0.40,0.400,
0.060, 0.060,0.400,0.40,0.660,0.660,0.25,0.25
+    dim=2


field set=net1    val="-1e17*gbox(X,0,0,0.3)*gbox(Y,-2.5,16,0.05)"
field set=net2    val="-1e18*gbox(X,6,7,1.5)*gbox(Y,-2.5,16,0.05)"


field set=ns1     val="2e20*gbox(X,0,0,0.07)*gbox(Y,0,1.46,0.0003)"
field set=ns2     val="2e20*gbox(X,0,0,0.07)*gbox(Y,10,11.46,0.0003)"


field set=nd1     val="2e20*gbox(X,0,0,0.07)*gbox(Y,2.04,3.5,0.0003)"
field set=nd2     val="2e20*gbox(X,0,0,0.07)*gbox(Y,12.04,13.5,0.0003)"


field set=nb1     val="-1e20*gbox(X,0,0,0.074)*gbox(Y,-2.5,-2,0.0003)"
field set=nb2     val="-1e20*gbox(X,0,0,0.074)*gbox(Y,15.5,16,0.0003)"
field set=netdope val="-5e14+ns1+nd1+ns2+nd2+net1+net2+nb1+nb2"
field set=netdope val=0 material=oxide
```

```
deposit mat=oxide start=1.45  end=2.05  thick=0.01 xdel=0.005
deposit mat=oxide start=11.45 end=12.05 thick=0.01 xdel=0.005


#source1
boundary xmin=0      xmax=0       ymin=0   ymax=1.0  name=C3
#drain1
boundary xmin=0      xmax=0       ymin=2.5 ymax=3.5   name=C1
#gate1
boundary xmin=-0.01   xmax=-0.01    ymin=1.45 ymax=2.05  name=C2
#bulk1
boundary xmin=0      xmax=0       ymin=-2.5 ymax=-2   name=C4


#source2
boundary xmin=0      xmax=0       ymin=12.5  ymax=13.5 name=C7
#drain2
boundary xmin=0      xmax=0       ymin=10   ymax=11   name=C5
#gate2
boundary xmin=-0.01   xmax=-0.01    ymin=11.45 ymax=12.05 name=C6
#bulk2
boundary xmin=0      xmax=0       ymin=15.5  ymax=16    name=C8


field set=tl val=300
field set=edge val=1


# start it up
bias initial system=silicon_poisson temper=27


# resolve with 2carriers
bias system=silicon_dd
```