PLAWE : A PIECEWISE LINEAR CIRCUIT SIMULATOR USING ASYMPTOTIC WAVEFORM EVALUATION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCE
OF BİLKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Selçuk TOPÇU
July, 1994

PLAWE: A PIECEWISE LINEAR CIRCUIT SIMULATOR USING ASYMPTOTIC WAVEFORM EVALUATION

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BİLKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

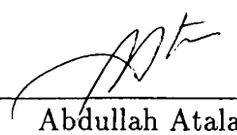
Satılmış Topçu

July 1994

SATILMIŞ TOPÇU

tarafından [illegible]ştır.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

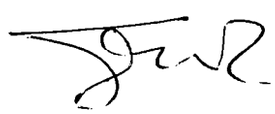


Abdullah Atalar, Ph. D. (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

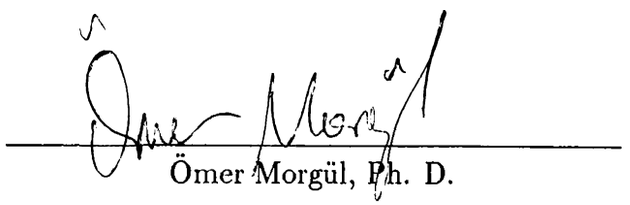
Mehmet A. Tan, Ph. D. (Co-supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.



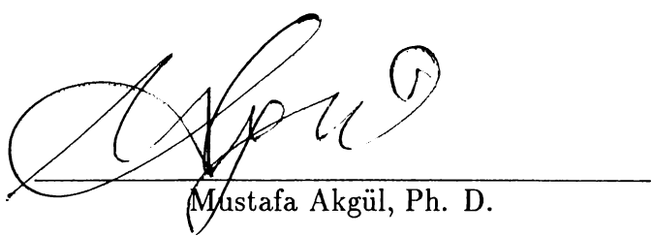
Erol Sezer, Ph. D.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.



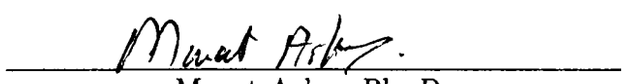
Ömer Morgül, Ph. D.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.



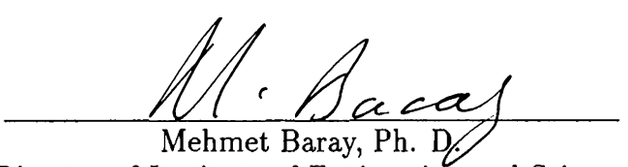
Mustafa Akgül, Ph. D.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.



Murat Aşkar, Ph. D.

Approved for the Institute of Engineering and Science:



Mehmet Baray, Ph. D.
Director of Institute of Engineering and Science

To Leyla and Alper Topçu

Abstract

PLAWE: A PIECEWISE LINEAR CIRCUIT SIMULATOR USING ASYMPTOTIC WAVEFORM EVALUATION

Satılmış Topçu

Ph. D. in Electrical and Electronics Engineering

Supervisors:

Prof. Dr. Abdullah Atalar and Assoc. Prof. Dr. Mehmet A. Tan

July 1994

A new circuit simulation program, PLAWE, is developed for the transient analysis of VLSI circuits. PLAWE uses Asymptotic Waveform Evaluation (AWE) technique, which is a new method to analyze linear(ized) circuits, and Piecewise Linear (PWL) approach for DC representation of nonlinear elements.

AWE employs a form of Padé approximation rather than numerical integration techniques to approximate the response of linear(ized) circuits in either the time or the frequency domain. AWE is typically two or three orders of magnitude faster than traditional simulators in analyzing large linear circuits. However, it can handle only linear(ized) circuits, while the transient analysis problem is generally nonlinear due to the presence of nonlinear devices such as diodes, transistors, etc.. We have applied the AWE technique to the transient simulation of nonlinear circuits by using static PWL models for nonlinear elements. But, finding a good static PWL model which fits well to the actual $i - v$ characteristics of a nonlinear device is not an easy task and in addition, static PWL modelling results in low accuracy. Therefore, we have developed a dynamic PWL modeling technique which uses SPICE models for nonlinear elements to enhance

the accuracy of the simulation while preserving the efficiency gain obtained with AWE. Hence, there is no modelling problem and we can adjust the accuracy level by varying some parameters. If the required level of accuracy is increased, more simulation time is needed. Practical examples are given to illustrate the significant improvement in accuracy. For circuits containing especially weakly nonlinear devices, this method is typically at least one order of magnitude faster than HSPICE.

A fast and convergent iteration method for piecewise-linear analysis of nonlinear resistive circuits is presented. Most of the existing algorithms are applicable only to a limited class of circuits. In general, they are either not convergent or too slow for large circuits. The new algorithm presented in this thesis is much more efficient than the existing ones and can be applied to any piecewise-linear circuit. It is based on the piecewise-linear version of the Newton-Raphson algorithm. As opposed to the Newton-Raphson method, the new algorithm is globally convergent from an arbitrary starting point. It is simple to understand and it can be easily programmed. Some numerical examples are given in order to demonstrate the effectiveness of the presented algorithm in terms of the amount of computation.

Keywords: Circuit simulation, piecewise-linear, DC analysis, transient analysis, AWE, moment matching, Padé approximation.

Özet

PLAWE: ASİMTOTSAL EĞRİ BULMA YÖNTEMİNİ KULLANAN PARÇALI DOĞRUSAL BİR DEVRE BENZETİM YAZILIMI

Satılmış Topçu
Elektrik ve Elektronik Mühendisliği Doktora

Tez Yöneticileri:
Prof. Dr. Abdullah Atalar ve Doç. Dr. Mehmet A. Tan
Temmuz 1994

Çok geniş ölçekte tümleşik (VLSI) devrelerin geçici durum analizinde kullanılmak üzere yeni bir devre benzetim yazılımı, PLAWE, gerçekleştirilmiştir. PLAWE, doğrusal devrelerin analizi için yeni bir yöntem olan asimtotsal eğri bulma (AEB) tekniğini ve doğrusal olmayan elemanların DC gösterimi için parçalı doğrusal (PD) yaklaşımını kullanır.

AEB, doğrusal devrelerin yanıtını zaman veya sıklık alanında bulmak için sayısal tümlev teknikleri yerine bir çeşit Padé yaklaşımını kullanır. AEB, büyük doğrusal devrelerin analizinde geleneksel benzeticilerden tipik olarak yüz veya bin kat daha hızlıdır. Bununla beraber AEB yalnızca doğrusal devreleri kotarabilir. Oysa geçici durum analiz problemi diyot, transistör gibi doğrusal olmayan aygıtların varlığından dolayı genellikle doğrusal değildir. Doğrusal olmayan elemanlar için duruk PD modeller kullanılarak AEB tekniği doğrusal olmayan devrelerin geçici durum benzetiminde uygulanmıştır. Fakat doğrusal olmayan bir aygıtın gerçek $i - v$ karakteristiğine iyi uyan bir duruk PD

model bulmak zordur ve buna ilaveten duruk PD modelleme düşük doğruluk sonucunu verir. Bu yüzden AEB ile elde edilen verimlilik kazancını korumakla beraber benzetim doğruluğunu artırmak için doğrusal olmayan elemanlar yerine SPICE modelleri kullanan bir dinamik PD modelleme tekniği geliştirilmiştir. Böylece modelleme problemi ortadan kaldırılmıştır ve bazı parametreler değiştirilerek doğruluk derecesi ayarlanabilir hale gelmiştir. Eğer istenilen doğruluk derecesi artırılırsa daha çok benzetim zamanına gereksinim duyulmaktadır. Doğruluktaki önemli ilerlemeyi göstermek için pratik örnekler verilmiştir. Özellikle yumuşak huylu doğrusal olmayan aygıtları içeren devreler için, bu yöntem HSPICE ' tan tipik olarak en az on kat daha hızlıdır.

Doğrusal olmayan dirençli devrelerin parçalı doğrusal analizi için hızlı ve yakınsayan bir dürüm yöntemi sunulmuştur. Varolan algoritmaların çoğunluğu yalnızca sınırlı bir devre sınıfına uygulanabilmektedir. Genel olarak bu algoritmalar ya yakınsamamakta yada büyük devreler için çok yavaş kalmaktadırlar. Bu tezde sunulan yeni algoritma varolan algoritmalardan çok daha verimlidir ve herhangi bir parçalı doğrusal devreye uygulanabilmektedir. Bu algoritma Newton-Raphson algoritmasının parçalı doğrusal sürümü üzerine kurulmuştur. Newton-Raphson yönteminin aksine, bu yeni algoritma herhangi bir noktadan başlayarak her zaman yakınsar. Anlaşılması kolaydır ve basit olarak programlanabilir. Sunulan algoritmanın hesaplama miktarı cinsinden verimliliğini göstermek için bazı sayısal örnekler verilmiştir.

Anahtar Sözcükler: Devre benzetimi, parçalı doğrusal, DC analiz, geçici durum analizi, moment eşleme, Padé yaklaşımı.

Acknowledgment

I would like to express my deepest gratitude to Dr. Abdullah Atalar and Dr. Mehmet A. Tan for their supervisions and encouragements in all steps of the development of this work.

I would like to thank Dr. Erol Sezer, Dr. Murat Aşkar, Dr. Mustafa Akgül and Dr. Ömer Mörgül, the members of my jury, for their motivating and directive comments on my research.

I like to acknowledge the financial support of TÜBİTAK through the Science for Stability Programme for presentation of this work in *ISCAS'91*.

It is my pleasure to express my thanks to Ogan Ocalı for his valuable discussions and comments especially about the *popcorn* algorithm. I would also like to thank Cemal T. Dikmen and Murat Alaybeyi for their collaboration in writing the core of the program, PLAWE. Thanks to Mustafa N. Yazgan for providing the *grafix* program to view and print the PLAWE outputs.

A special note of thanks is due to my friends Mustafa Karaman, F. Levent Değertekin, Dilek Çolak, Ayhan Bozkurt and Mustafa Çelik for their moral support and friendship.

Finally, my sincere thanks are due to my family for their continuous moral support throughout my graduate study.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

There are several steps involved in the design of a very large-scale integrated (VLSI) circuit, which may consist of several hundreds of thousands of components, mainly transistors. The circuit designer first obtains a very high-level functional description of the circuit based on the specifications provided by the user. The synthesis, often called the top-down process, translates this high-level description into various levels including the register level, the transistor level etc. and terminates at the physical mask level. This is followed by the design verification, or the bottom-up process, where a simulation tool is used to predict the performance of the circuit which is compared with the user's specifications; thus, completing the so-called design loop. If the performance is not satisfactory, certain changes are made and the whole process is then repeated. The total time spent in the design loop is usually referred to as the turn-around time.

The main objective of a VLSI circuit designer is to obtain designs with a turn-around time as low as possible. Computer-aided design (CAD) tools are used at various steps in the design process to perform tasks which would otherwise take a very long time if they were done by human beings. There is a bottleneck in speeding up the bottom-up design verification process due to the unavailability of a simulation tool that is capable of accurately predicting the performance of an entire VLSI circuit at a reasonable cost. The accuracy of a simulator is important since otherwise the integrated circuit which is

fabricated and tested might turn out to perform rather unsatisfactorily. For large circuits, the speed of simulation is equally important so that the entire circuit can be simulated in a reasonably small amount of computation time.

There are a variety of circuit simulation tools, with different accuracy and speed, which are used in the circuit analysis and design. The accuracy and speed requirements may vary depending upon the size and type of the circuit. The extensive computations and thus very long simulation times are mainly due to the complex nonlinear characteristics of the devices and due to the large number of iterations for computing the transient response in timing simulation. Almost all the existing circuit simulators use iterative methods (e.g. Newton Raphson) to handle nonlinear characteristics and numerical integration methods (e.g. Forward Euler, Backward Euler, Trapezoidal, etc.) to compute the time domain responses of energy storage elements.

Aspects of stability, convergence and hence completion of the job in a successful manner are all important issues for circuit simulation tools. Moreover, the models of new devices resulting from the emerging technologies must be easily put into a simulator. Otherwise, the simulator may become obsolete in a short time.

By the motivation of these facts, a new circuit simulator, PLAWE (Piecewise Linear Asymptotic Waveform Evaluator) has been developed. PLAWE can solve large circuits containing linear energy storage elements and passive and active linear or nonlinear resistive elements. It uses the Asymptotic Waveform Evaluation (AWE) [1] method and the Piecewise Linear (PWL) approach for DC representation of nonlinear devices.

1.1 Simulation Techniques

Most of the existing simulators for integrated circuits can be classified into two distinct categories, namely, *analog simulators* and *digital simulators*. Analog simulators treat an electronic circuit as a continuous dynamical system with the electrical signals such as voltages and currents. Digital simulators, on the other hand, view the circuit as a digital

network with signals occupying discrete states such as low (**0**) and high (**1**). For small circuit blocks where analog voltage levels are critical in evaluating the circuit performance, analog circuit simulators such as SPICE [2] and ASTAP [3] can be used to predict the performance of the circuit very accurately. These are general purpose simulators in that they can handle almost any type of circuit element such as resistors, capacitors, inductors, voltage and current sources (independent and controlled), nonlinear devices (transistors, diodes, etc.) and transmission lines. They can also perform many types of analyses such as DC, AC (or small-signal), noise, and transient analyses. However, as the size of the circuit (i.e. number of components) increases, using these simulators is no longer cost-effective. Several decomposition techniques have been used to speed-up their performance and have resulted in the development of a variety of analog simulators such as SLATE [4], MACRO [5], MOTIS [6], MOTIS-C [7], MOTIS-II [8], PREMOS [9], RELAX [10]–[12], SPLICE [13], DIANA [14], SAMSON [15], IDSIM [16], CINNAMON [17], and SPECS [18].

The existing digital simulators can be further divided into *Boolean gate-level* [19],[20] and *switch-level* [21]–[26] simulators. In the Boolean gate model a circuit consists of a set of logic gates connected by unidirectional memoryless wires. Information is only stored in the feedback paths of sequential circuits. The Boolean gate model, however, cannot describe some of the new technologies currently used in VLSI circuits, especially circuits with MOS transistors. Because, the MOS circuits consist of bidirectional switching elements connected by bidirectional wires with memory due to the interconnect and device capacitances. The switch-level simulators model an MOS circuit as a set of nodes connected by transistor switches. Each node occupies a discrete number of states **0**, **1**, or **X** for the intermediate or unknown state and each switch is either open, closed, or in an intermediate state. Digital simulators, in general, operate at sufficient speeds to test entire VLSI systems, since the circuit behavior is modeled at a logical level rather than a detailed electrical level. However, these simulators do not model the dynamics of the circuits properly and are often useful only in predicting steady-state responses of the circuits. Analog simulators, on the other hand, are fairly accurate in predicting both steady-state and transient responses but they are cost effective only for circuits, with less than a few thousand components.

In many of the VLSI circuits the detail provided by the analog simulators are not required for the entire circuit, but only for some critical areas of the circuit. *Mixed-mode* simulators allow the designer to use a combination of analog simulation and digital simulation, in the same program. These simulators, such as SPLICE [13], DIANA [14], and SAMSON [15], have been observed to realize a one or two order of magnitude reduction in simulation time while still providing a detailed circuit-level analysis where necessary. These simulators, however, work well as long as only small, isolated sections of the circuit need to be simulated as analog circuits. Furthermore, trying to combine analog and digital models in a single program requires rather unsatisfactory approximations at the interfaces. Therefore, unless great care is exercised, a mixed-mode simulator could end up providing the accuracy of a digital simulator at the speed of an analog simulator.

1.2 Piecewise Linear Modeling of Nonlinear Devices

Models describe the device behavior to a circuit simulation program. A device model can be used for a variety of different purposes and ideally it would be convenient to have only one model which can serve all the needs. However, different applications impose different requirements on the model. A completely theoretical model based on the fundamental of physics becomes practically intractable. On the other hand, use of a completely empirical model results in a loss of predictive capabilities. A compromise is usually made in developing models for circuit simulation [27].

The nonlinear device models compute the terminal currents of the device in terms of the terminal voltages. These terminal currents should be continuous functions of the terminal voltages for convergence. Sometimes, it is easier to divide the operating range of the device into different regions so that the model equations can be conveniently formulated. Since different equations are used for each region of operation, it is important to make sure that the terminal currents are continuous across the region boundaries. In

DC analysis, it is possible to encounter wide variations in the terminal voltages. Therefore, it is important to consider the entire voltage range while formulating the model equations even though the device will not encounter these voltages in practical circuits.

During model development, it is important to keep the following paradox in mind.

> More complex models can potentially represent the device characteristics more accurately. But, it is more difficult to extract all the model parameters for such complex models in a computationally efficient manner. In addition, if the model parameters are not specified properly, the device characteristics will not be reproduced accurately.

In PLAWE, piecewise linear models are used to characterize the nonlinear devices since one of the major goals is to finish the simulation in a reasonably short time. The main reason behind the choice of PWL approximation is that it results in a set of linear equations and hence the iterative solutions of nonlinear equations are avoided. So, the time complexity is decreased and the convergence in DC analysis is guaranteed. Since AWE can be applied only to linear(ized) circuits, the PWL approach can make usage of the AWE technique in the simulation of nonlinear circuits. The user can create his/her own device models for nonlinear elements. Hence, the PWL approximation enables the user to determine the trade-off between the speed and accuracy of the simulation.

PLAWE employs PWL models which can be built at various levels of accuracy to describe two and three terminal nonlinear devices. A two-terminal nonlinear device is represented by a linear equation in terms of its branch voltage and current as

$$av + bi + c = 0$$

in every region of the (v, i) plane. Similarly, a three-terminal nonlinear device model consists of a set of regions in the (v_1, v_2, i_1, i_2) space. Every region is represented in terms of two linear terminal equations of the form

$$a_{1k}v_1 + a_{2k}v_2 + b_{1k}i_1 + b_{2k}i_2 + c_k = 0, \quad k = 1, 2$$

It should be noted that the accuracy of the PWL model for a nonlinear device depends on the number of regions into which the terminal equations are linearized. However, as

the number of regions increases, the complexity of the analysis may increase dramatically. Nevertheless, it is observed that, with the inclusion of constant grounded capacitances, models of diodes and transistors with only a few segments (2 segments for diodes and 4 segments for MOSFET's) yield quite good results for timing analysis.

1.3 Formulation of Circuit Equations

In PLAWE, Sparse Tableau Analysis (STA) method [28] is used to formulate the circuit equations. In STA method, with nonlinear devices replaced by linear equivalents, the circuit is described with a large and sparse matrix equation, involving the Kirchoff's Current Law (KCL), Kirchoff's Voltage Law (KVL) and the Branch Constitutive Equations (BCE). A KCL equation is written in terms of branch currents for each node. A KVL equation relating a branch voltage to its node voltages is written for each branch. Finally, a branch constitutive equation is written for each branch in terms of its branch voltage and current. Collectively, these equations may be written as follows:

$$\begin{bmatrix} \boldsymbol{I} & \mathbf{0} & -\boldsymbol{A}^T \\ \mathbf{0} & \boldsymbol{A} & \mathbf{0} \\ \boldsymbol{G}_l & \boldsymbol{R}_l & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_b \\ \boldsymbol{I}_b \\ \boldsymbol{V}_n \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{w}_l \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{w} \end{bmatrix} \begin{matrix} \text{KVL} \\ \text{KCL} \\ \text{BCE} \end{matrix} \tag{1.1}$$

or in compact form

$$\boldsymbol{T}_l\, \boldsymbol{x} + \boldsymbol{y}_l = \boldsymbol{y} \tag{1.2}$$

where $\boldsymbol{V}_b$, $\boldsymbol{I}_b$, and $\boldsymbol{V}_n$ are branch voltages, branch currents, and node voltages, respectively. The vectors $\boldsymbol{y}_l$ and $\boldsymbol{y}$ denote the equivalent sources due to linearization of nonlinear elements and the independent sources, respectively. The subscript l denotes the *linear region* in which the circuit operates. The size of STA matrix is $(2b + n)$ where b is the number of branches and n is the number of ungrounded nodes in the circuit.

Chapter 2

PIECEWISE LINEAR MODELS

A simulator that utilizes piecewise linear models can achieve simulations of arbitrary accuracy because piecewise linear models can be made to conform to nonlinear device characteristics with arbitrary precision by simply adding regions of linearity. However, as pointed out in the preceding chapter, complex models degrade the efficiency of the simulator. Simpler models yield more efficient simulations, and there are strong incentives to use models that are as simple as possible.

Several restrictions can be placed upon the piecewise linear models in order to preserve the efficiency of simulation. First, the number of piecewise linear regions must be small. This is necessary to keep the principle advantage of the AWE technique: the ability to take large time steps. Secondly, only linear capacitors are allowed in the models. Nonlinear capacitors must be modeled using equivalent linear capacitors. Although it may be possible to approximate nonlinear capacitors with piecewise linear capacitors, this was not explored. Finally, the DC coupling from the gate (base) to the source and drain (emitter and collector) for an MOS (bipolar) transistor can be restricted to be unidirectional. This facilitates circuit partitioning which is not explored in this thesis. After a brief discussion of the representation of piecewise linear models, this chapter describes simple MOS and bipolar models and simulations are given to demonstrate the capabilities of these models.

2.1 Piecewise Linear Representation

A piecewise linear device is represented by a collection of linear circuits, each of which represents the linearized behavior of the device for a particular region of operation. Each region of operation is a *polytope* [29],[30] described by a conjunction of linear inequalities in terms of the device's terminal voltages. Each inequality represents a *hyperplane* boundary. For example, a description of the MOS switched resistor model which is just a particular piecewise linear model is shown in Fig. 2.1 and Fig. 2.2. The electrical behavior of the device in each of its two regions is modeled by the circuits in Fig. 2.1 (a) and (b). In Fig. 2.2, the region to the right of the plane labeled $V_{gs} = V_t$ is the polytope corresponding to the *on* region while the region to the left of the plane is the polytope corresponding to the *off* region. More general models may have circuits consisting of interconnections of linear circuit elements. Additionally, they may have more than two regions of linearity.

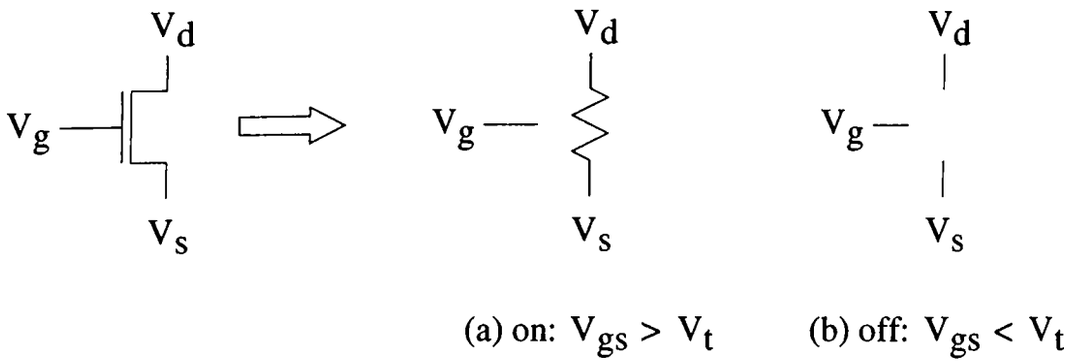


Figure 2.1: The MOS switched resistor model.

In general, a device may have n terminals. Without loss of generality, we can concentrate on the voltage-controlled devices. Consider the n dimensional space defined by the voltages at those terminals: $\{v_1, v_2, \ldots, v_n\}$. Then the set of points whose coordinates satisfy a given linear equation in those voltages:

$$a_0 + a_1 v_1 + a_2 v_2 + \cdots + a_n v_n = 0 \tag{2.1}$$

defines a hyperplane in the n dimensional space. The hyperplane is simply the multidimensional generalization of the familiar three dimensional plane. Like planes,

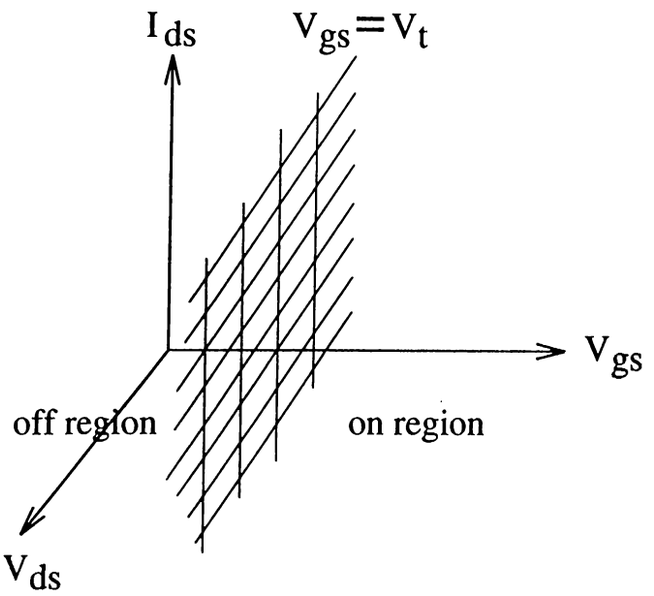


Figure 2.2: The regions of operation for MOS switched resistor model.

hyperplanes partition space. Points that lie on one side of the hyperplane have coordinates that satisfy the inequality:

$$a_0 + a_1 v_1 + a_2 v_2 + \cdots + a_n v_n < 0 \tag{2.2}$$

while points on the opposite side satisfy:

$$a_0 + a_1 v_1 + a_2 v_2 + \cdots + a_n v_n > 0 \tag{2.3}$$

The polytope is the multidimensional generalization of the *polyhedron.* While a polyhedron is a region bounded by planes in three dimensional space, a polytope is a region bounded by hyperplanes in n dimensional space. Equations 2.2 and 2.3 suggest that a polytope can be specified by a conjunction of linear inequalities:

$$\begin{aligned} a_0 + a_1 v_1 + a_2 v_2 + \cdots + a_n v_n &> 0 \\ b_0 + b_1 v_1 + b_2 v_2 + \cdots + b_n v_n &> 0 \\ c_0 + c_1 v_1 + c_2 v_2 + \cdots + c_n v_n &> 0 \\ &\vdots \end{aligned} \tag{2.4}$$

Each inequality bounds the region by a hyperplane.

2.2 MOS Transistor Model

2.2.1 4-segment MOS model

The regions of operations for the 4-segment model are plotted in Fig. 2.3. This model has 4 regions of operation corresponding to cut-off, forward saturation, linear, and reverse saturation states of an MOS transistor. Although there are four regions, any given region is bounded by only two hyperplanes. This is particularly important as far as the efficiency is concerned because the effort required to detect whether a piecewise linear device changes region is proportional to the number of boundaries of the current region.

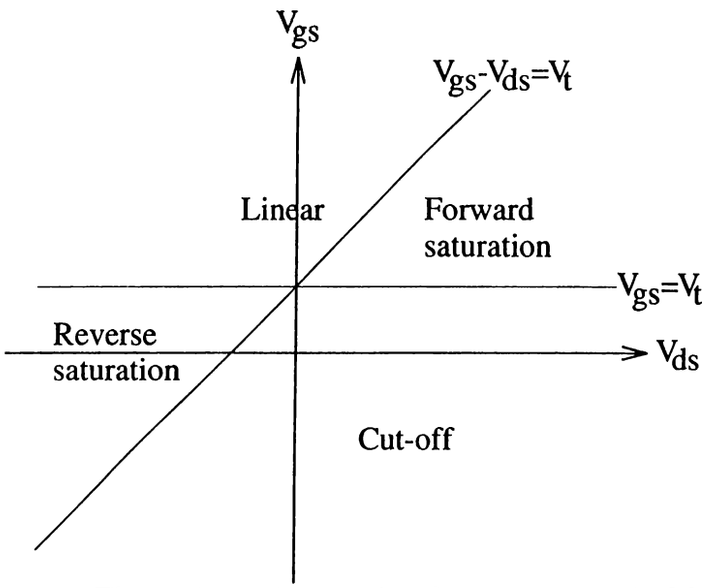


Figure 2.3: The regions of operation for 4-segment nMOS model.

The resulting 4-segment MOS model is depicted in Fig. 2.4. In the *cut-off* region the MOS transistor is replaced by an open circuit. Hence, the drain-to-source current in this region is zero. In the *forward saturation* region the MOS transistor is modeled by a linear voltage controlled current source. The transconductance is set by the parameter g_m, while the gate-to-source voltage for which the current source delivers zero current is V_t. In the *linear* region, which contains both the forward linear and reverse linear regions, the MOS transistor is modeled by a conductance, g_{ds}. In the *reverse saturation* region

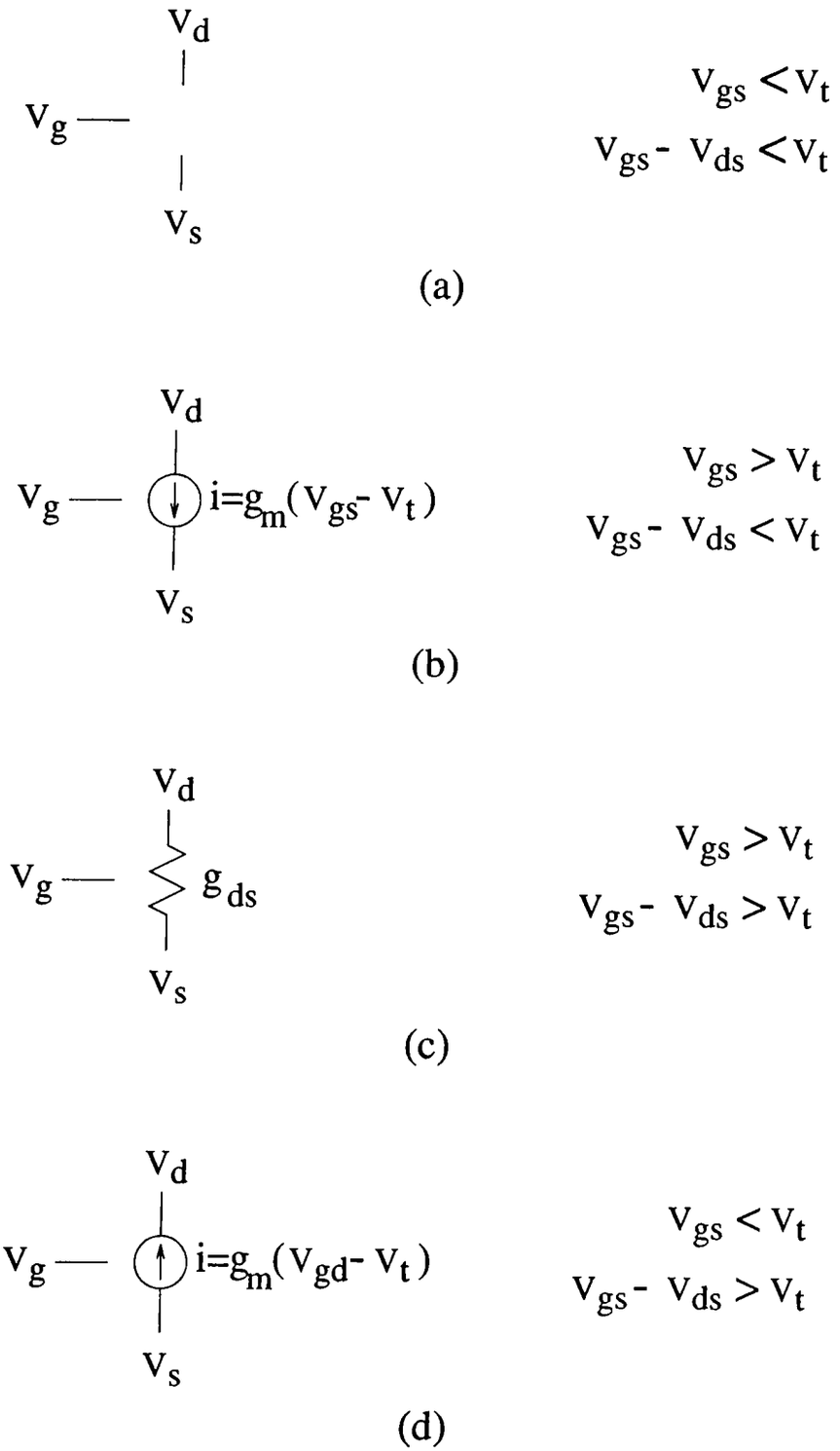


Figure 2.4: The 4-segment PWL nMOS model: (a) cut-off region (b) forward saturation region (c) linear region (d) reverse saturation region.

the MOS transistor is modeled by a linear voltage controlled current source. The current source is controlled by the gate-to-drain voltage V_{gd} since the drain-to-source voltage, V_{ds}, is negative in this region. The inequalities in Fig. 2.4 define the hyperplane boundaries for each region of operation. Consequently, the piecewise linear equations describing the behavior of an nMOS device in the four regions are:

$$I_{ds} = \begin{cases} 0 & \text{Cut-off: } V_{gs} < V_t \ , \ V_{gs} - V_{ds} < V_t \\ g_m(V_{gs} - V_t) & \text{Forward saturation: } 0 < V_{gs} - V_t < V_{ds} \\ g_{ds}V_{ds} & \text{Linear: } V_{gs} > V_t \ , \ V_{gs} - V_{ds} > V_t \\ -g_m(V_{gd} - V_t) & \text{Reverse saturation: } V_{ds} < V_{gs} - V_t < 0 \end{cases} \tag{2.5}$$

The same MOS device equations also apply to the pMOS device. The only difference is the change in the sign associated with the voltages and drain-to-source current. It is seen from Eqn. (2.5) that on the boundary between the forward saturation and linear regions (i.e., $V_{ds} = V_{gs} - V_t$), the drain-to-source current is $I_{ds} = g_m V_{ds} = g_{ds} V_{ds}$. Therefore, the conductance g_{ds} must be chosen to be equal to the transconductance g_m for preserving the continuity of I_{ds}. In order to illustrate the ability of 4-segment piecewise linear model to match the nonlinear I-V characteristics of an MOS transistor, the three dimensional visualizations of the nonlinear and piecewise linear MOS characteristics are given in Fig. 2.5 and in Fig. 2.6, respectively. In these figures, the z-axis which is not depicted, represents the drain-to-source current (I_{ds}) of the transistor.

In a VLSI circuit every transistor may have a different physical geometry, i.e. their channel widths (W) and channel lengths (L) may be different from each other. We know that the channel current is dependent on the device geometry. This means that the width and length of the channel must be taken into account while calculating the drain-to-source current. For this purpose, the value of I_{ds} calculated from (2.5) is multiplied by the (W/L) ratio, which depends on the actual geometry of the device.

In addition to the linear circuit elements in the 4-segment model given in the Fig. 2.4, a very small conductance, *gmin*, is placed in parallel with the drain to source nodes of the MOS device. This conductance is used to eliminate the potential problems which may occur due to the open-circuit equivalent of off transistors (Fig. 2.4 (a)). It also enhances

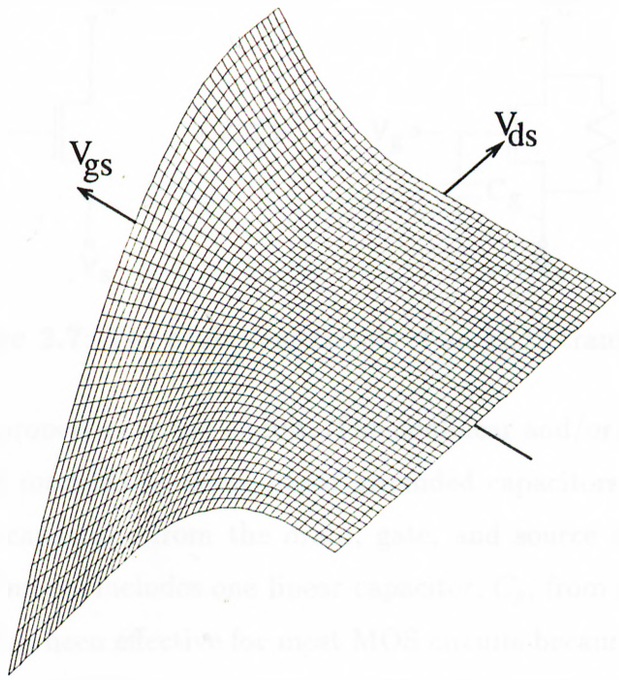


Figure 2.5: Three dimensional plot of nonlinear MOS I-V characteristics.

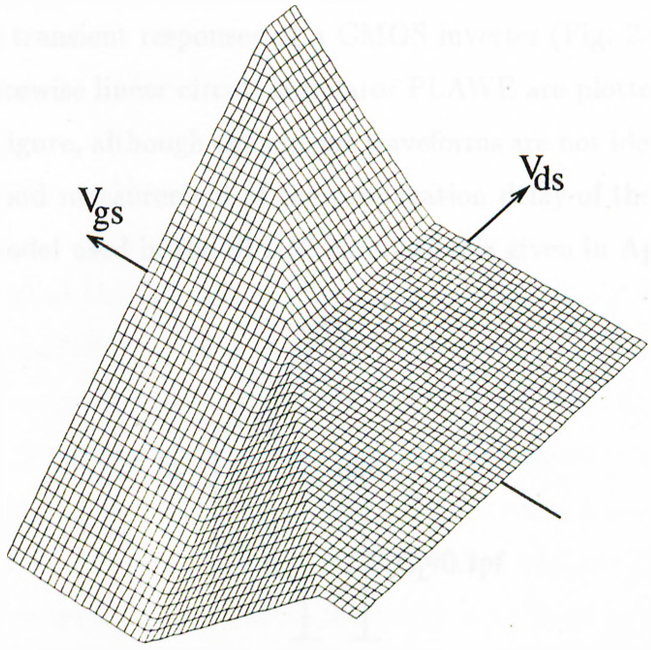


Figure 2.6: Three dimensional plot of PWL MOS I-V characteristics.

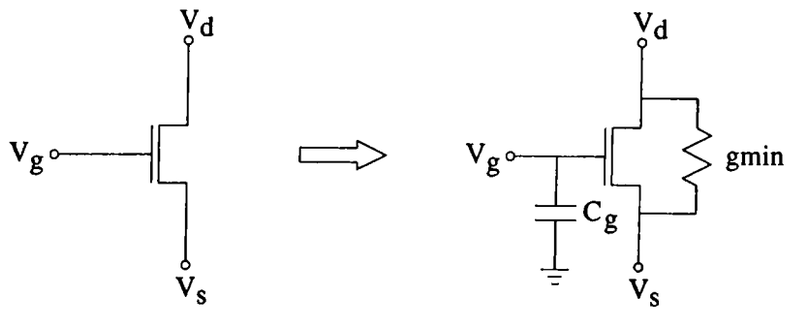


Figure 2.7: The equivalent circuit of an MOS transistor.

the DC convergence properties of the circuit. The nonlinear and/or floating capacitances of the MOS device is modeled by using linear grounded capacitors. It may be possible to include grounded capacitors from the drain, gate, and source nodes to the ground. The 4-segment MOS model includes one linear capacitor, C_g, from gate node to ground. This approximation has been effective for most MOS circuits because the majority of the capacitances is connected to ground. Therefore, the MOS transistors are replaced by an equivalent circuit shown in the Fig. 2.7.

In order to demonstrate the ability of 4-segment model to duplicate the nonlinear MOS characteristics, the transient responses of a CMOS inverter (Fig. 2.8) obtained by using SPICE and our piecewise linear circuit simulator PLAWE are plotted in the Fig. 2.9. As it is seen from this figure, although the output waveforms are not identical, the 4-segment model provides a good measurement of the propagation delay of the inverter to the 50% point. The MOS model used in the SPICE simulation is given in Appendix.

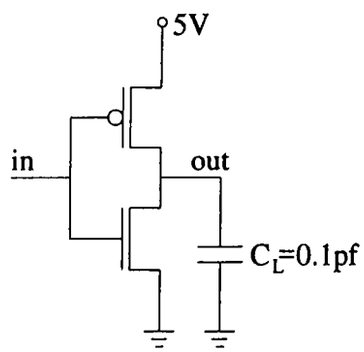


Figure 2.8: The CMOS inverter

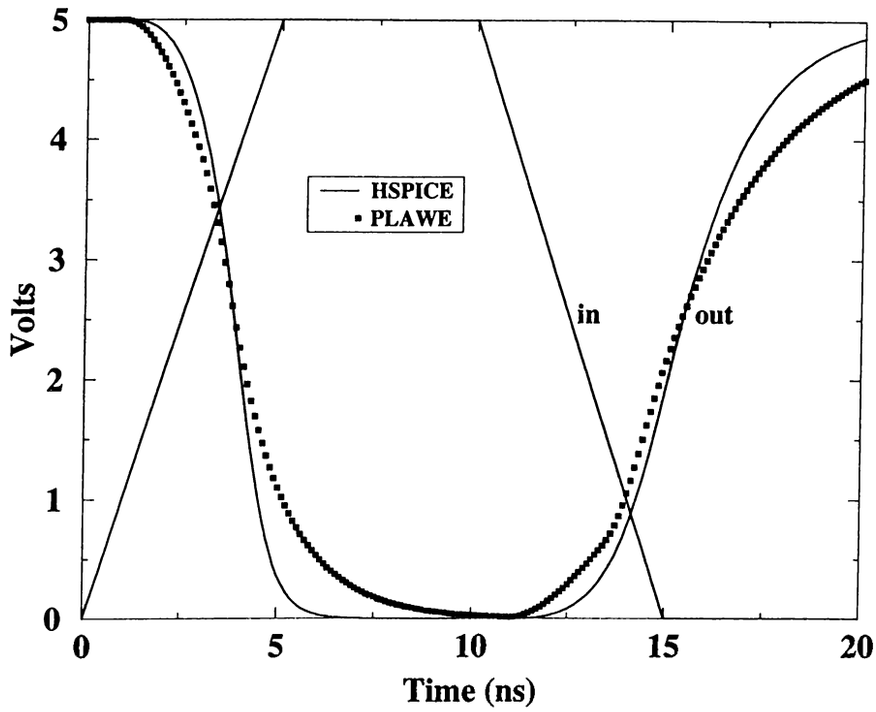


Figure 2.9: Transient response of a CMOS inverter by using SPICE model and 4-segment PWL MOS model ($V_t = 1.0$V, $gmin = 10^{-8}$, for ntype transistor: $g_m = 7.25 \times 10^{-5}$ and for ptype transistor: $g_m = 3.5 \times 10^{-5}$).

2.2.2 9-segment MOS model

As stated previously, accuracy of the simulation can be increased arbitrarily by simply adding regions of linearity to the piecewise linear models of nonlinear devices. Therefore, we extracted the 9-segment PWL MOS model which matches the nonlinear I-V characteristics of MOS transistor better than the 4-segment model. The regions of operations for the 9-segment model are plotted in Fig. 2.10. Note that there are 4 regions for the linear state, 2 regions each for the forward and reverse saturation states, and 1 region for the cut-off state of an MOS transistor. The piecewise linear equations describing the behavior of an nMOS device in the 9-segment MOS model are given in the Table 2.1. Similar to the 4-segment model, the MOS transistor is replaced by the equivalent circuit shown in the Fig. 2.7 and the current (I_{ds}) calculated from the Table 2.1 is multiplied by the (W/L) ratio to include the effect of the device geometry.

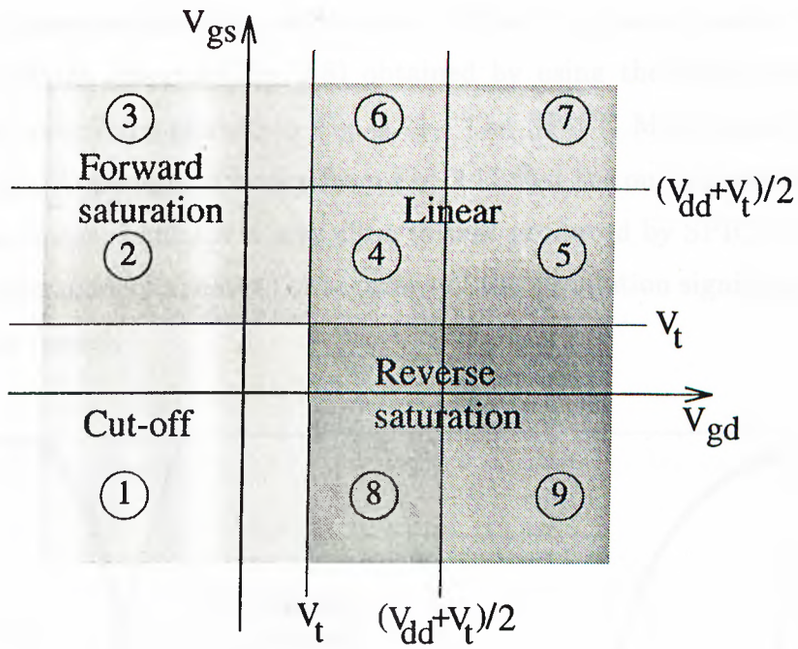


Figure 2.10: The regions of operation for 9-segment MOS model.

	Region	Equation	Boundaries
Cut-off	Region 1	$I_{ds} = 0$	$V_{gs} < V_t$ $V_{gd} < V_t$
Forward saturation	Region 2	$I_{ds} = g_m(V_{gs} - V_t)$	$V_t < V_{gs} < (V_{dd} + V_t)/2$ $V_{gd} < V_t$
	Region 3	$I_{ds} = g_m(3V_{gs} - V_{dd} - 2V_t)$	$V_{gs} > (V_{dd} + V_t)/2$ $V_{gd} < V_t$
Linear	Region 4	$I_{ds} = g_m(V_{gs} - V_{gd})$	$V_t < V_{gs} < (V_{dd} + V_t)/2$ $V_t < V_{gd} < (V_{dd} + V_t)/2$
	Region 5	$I_{ds} = g_m(V_{gs} - 3V_{gd} + V_{dd} + V_t)$	$V_t < V_{gs} < (V_{dd} + V_t)/2$ $V_{gd} > (V_{dd} + V_t)/2$
	Region 6	$I_{ds} = g_m(3V_{gs} - V_{gd} - V_{dd} - V_t)$	$V_{gs} > (V_{dd} + V_t)/2$ $V_t < V_{gd} < (V_{dd} + V_t)/2$
	Region 7	$I_{ds} = 3g_m(V_{gs} - V_{gd})$	$V_{gs} > (V_{dd} + V_t)/2$ $V_{gd} > (V_{dd} + V_t)/2$
Reverse saturation	Region 8	$I_{ds} = -g_m(V_{gd} - V_t)$	$V_{gs} < V_t$ $V_t < V_{gd} < (V_{dd} + V_t)/2$
	Region 9	$I_{ds} = -g_m(3V_{gd} - V_{dd} - 2V_t)$	$V_{gs} < V_t$ $V_{gd} > (V_{dd} + V_t)/2$

Table 2.1: The piecewise linear equations for 9-segment MOS model.

In order to demonstrate the performance of the 9-segment model, the transient responses of a CMOS inverter (Fig. 2.8) obtained by using the SPICE model and the piecewise linear model are plotted in Fig. 2.11. The SPICE MOS model used for this example is given in Appendix. It is seen from Fig. 2.11 that the output waveform produced by the piecewise linear simulator is very close to that produced by SPICE. In addition to this, the 9-segment model improves the accuracy of the simulation significantly compared to the 4-segment model.

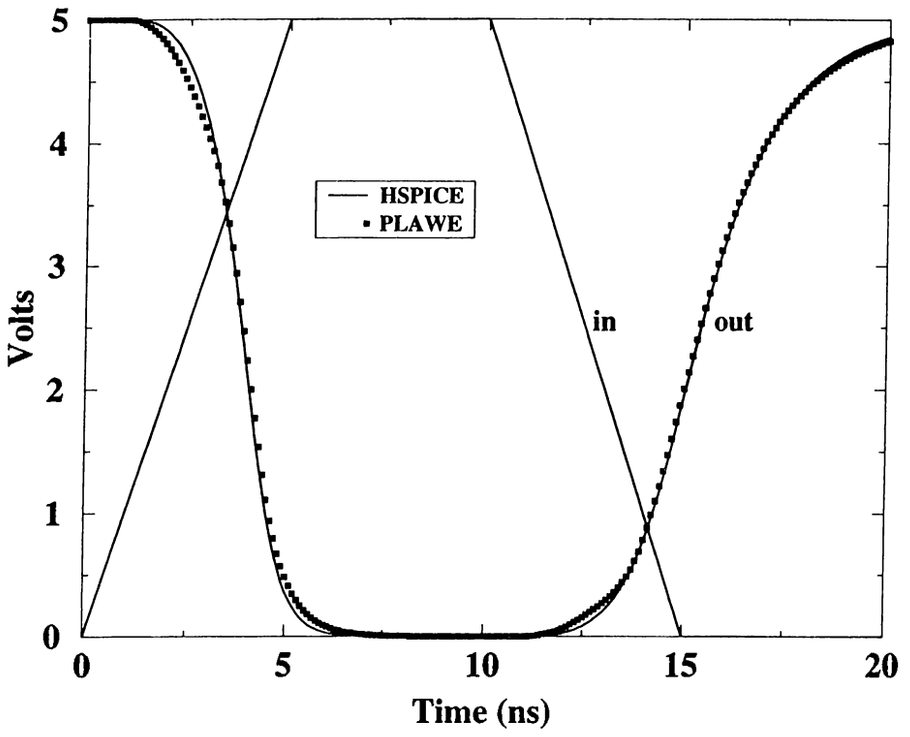


Figure 2.11: Transient response of a CMOS inverter by using SPICE model and 9-segment PWL MOS model (V_{dd} = 5V, V_t = 1.0V, $gmin = 10^{-8}$, for ntype transistor: $g_m = 5.56 \times 10^{-5}$ and for ptype transistor: $g_m = 1.94 \times 10^{-5}$).

2.3 Bipolar Transistor Model

The piecewise linear bipolar transistor model is extracted using the well-known Ebers-Moll model [31] which is shown in the Fig. 2.12. The terminal currents I_E and I_C can be expressed in terms of the diode currents as

$$\begin{aligned} I_E &= I_{DE} - \alpha_R I_{DC} \\ I_C &= \alpha_F I_{DE} - I_{DC} \end{aligned} \tag{2.6}$$

The terminal current I_B can be obtained by Kirchoff's current law as $I_B = I_E - I_C$.

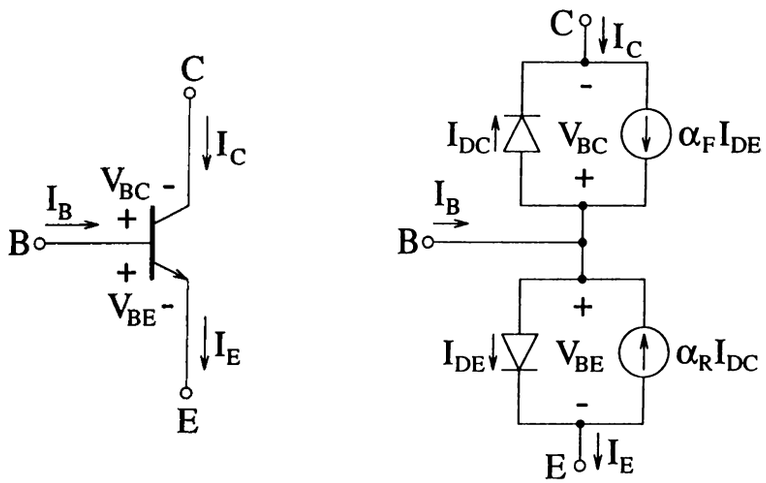


Figure 2.12: The Ebers-Moll model for an *npn* bipolar transistor.

Bipolar transistor has four possible modes of operation listed in the Table 2.2 as a function of the bias that is applied to the emitter and collector junctions. Therefore, by using a simple *on/off* model for the diodes in the Fig. 2.12, we can divide the operating region of a bipolar transistor into four linear regions as shown in the Fig. 2.13. Then, the piecewise linear equations for the diode currents I_{DE} and I_{DC} can be written as

$$\begin{aligned} I_{DE} &= g_{ej} V_{BE} + I_{eoj} \\ I_{DC} &= g_{cj} V_{BC} + I_{coj} \end{aligned} \tag{2.7}$$

for each linear region of operation ($j = 1, 2, 3, 4$).

Emitter junction	Collector junction	Mode of operation
Reverse	Reverse	Cutoff
Forward	Reverse	Forward active
Forward	Forward	Saturation
Reverse	Forward	Reverse active

Table 2.2: The modes of operation for the bipolar transistor.

Now let us combine Eqns. (2.6) and (2.7), and we have the piecewise linear equations for the terminal currents I_E and I_C.

$$\begin{aligned} I_E &= g_{ej}\, V_{BE} - \alpha_R\, g_{cj}\, V_{BC} + (I_{eoj} - \alpha_R\, I_{coj}) \\ I_C &= \alpha_F\, g_{ej}\, V_{BE} - g_{cj}\, V_{BC} + (\alpha_F\, I_{eoj} - I_{coj}) \quad \text{for } j = 1, 2, \ldots \end{aligned} \tag{2.8}$$

Note that if each diode in the Ebers-Moll model is modeled with n piecewise linear segments, then our bipolar model will have n^2 regions of operation. So, we can improve the accuracy of bipolar model by simply adding linear segments into the diode models.

We model the nonlinear floating capacitances of the bipolar device using equivalent linear grounded capacitors. For this purpose, we include a linear capacitor from base-to-ground in the bipolar model as shown in the Fig. 2.14. In addition, a very small

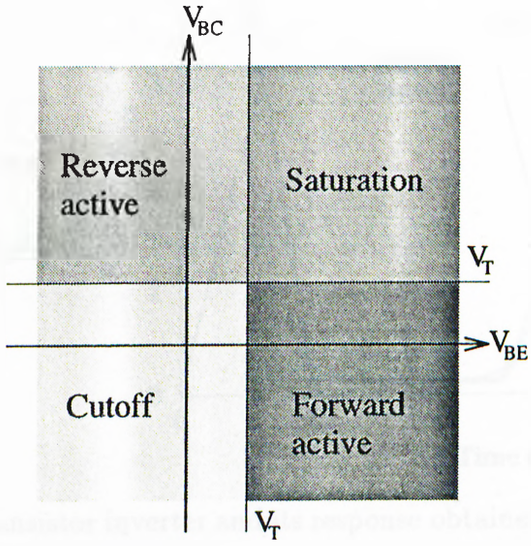


Figure 2.13: The regions of operation for a bipolar transistor.

conductance *gmin* is placed in parallel with both the base-emitter and base-collector junctions (Fig. 2.14).

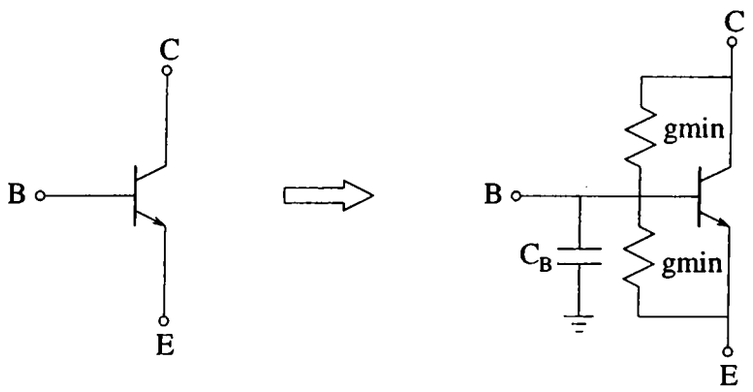


Figure 2.14: The equivalent circuit of a bipolar transistor.

The Fig. 2.15 depicts a bipolar inverter and compares the response of the circuit using piecewise linear and SPICE models. The match between waveforms is quite good especially considering the simplicity of the piecewise linear model.

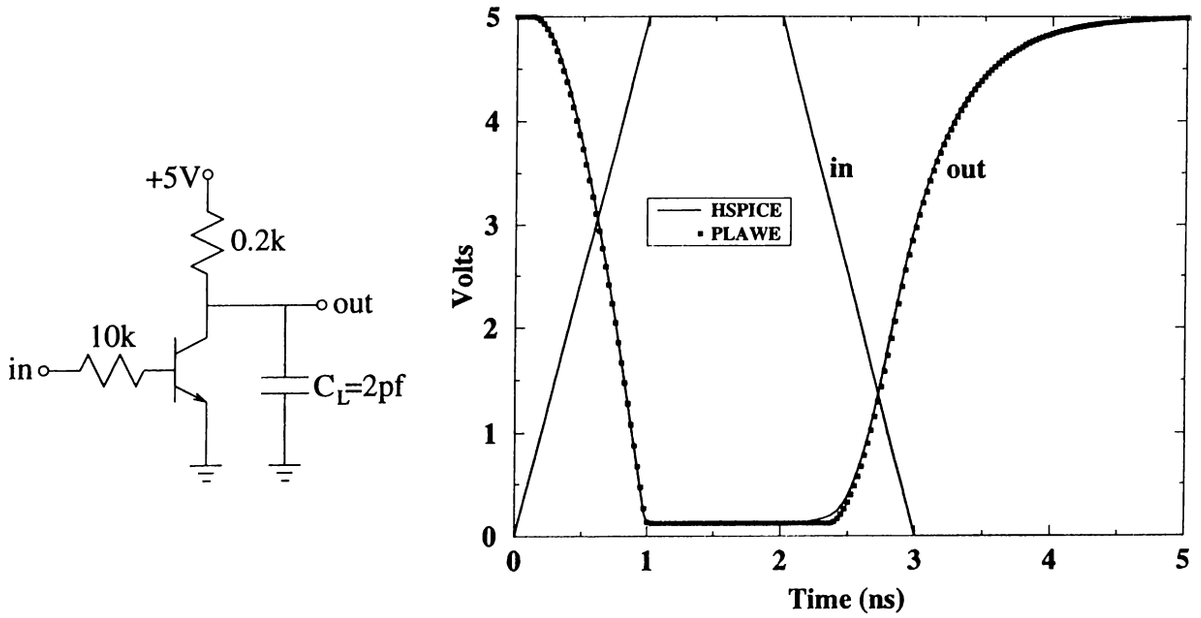


Figure 2.15: Bipolar transistor inverter and its response obtained by using SPICE model and piecewise linear model.

An ECL inverter [31] and its transient response obtained by using piecewise linear and SPICE models are given in the Fig. 2.16. The inverting and noninverting outputs of the ECL inverter are loaded with $C_L = 0.1$pf. As it is seen from Fig. 2.16, output of the piecewise linear simulator fits very well with the SPICE outputs. The SPICE BJT model used for the bjt inverter and ECL inverter examples is given in Appendix.

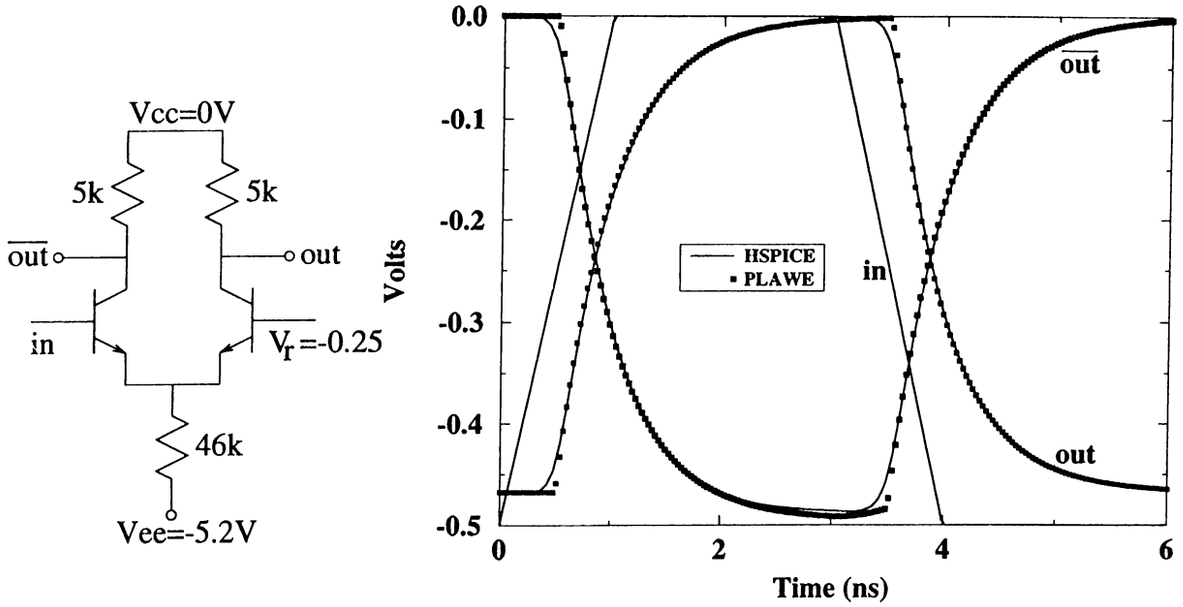


Figure 2.16: An ECL inverter and its response obtained by using SPICE model and piecewise linear model.

Chapter 3

DC ANALYSIS

Finding the "operating point" or "DC solution" of a circuit is usually the first step in the analysis of nonlinear circuits. It is the basis for DC sweep and usually provides the initial conditions for transient analysis and a DC operating point for ac analysis. DC analysis is important not only from a circuit theoretic point of view, but even more so from a computational point of view. Indeed, an essential part of most nonlinear transient analysis computer programs for solving dynamic nonlinear networks is a DC analysis subprogram. In the transient analysis, the DC network, obtained by replacing the capacitors by equivalent voltage sources and the inductors by equivalent current sources, usually possesses a unique solution. In other situations, a DC resistive nonlinear network obtained by open-circuiting all capacitors and short-circuiting all inductors may actually possess several distinct solutions. DC analysis involves determining node voltages for given values of DC sources and it is equivalent to the solution of nonlinear algebraic systems of equations. A well-known technique for solving systems of nonlinear equations is the Newton-Raphson iteration scheme. For well-behaved problems, this method converges rapidly. However, it may, under certain circumstances, diverge or oscillate about a solution.

In many cases, the equations describing the nonlinearities are not known in functional form and only tables of measured values are given. In other cases, the functions may

be known but are very complicated and it is convenient to replace the nonlinearities by piecewise linear equations to take advantage of the linearity. PLAWE uses piecewise linear models to describe the nonlinear devices. For DC nonlinear circuit simulation, PWL analysis is attractive because it can provide important features such as convergence in DC analysis, computational efficiency, and a clearly defined accuracy/speed trade-off.

A nonlinear resistive circuit can be described by

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{y} \tag{3.1}$$

where $\boldsymbol{g}(\cdot)$ is a continuous nonlinear mapping from $\boldsymbol{R}^n$ into itself, $\boldsymbol{x}$ is a point in $\boldsymbol{R}^n$ and represents a set of chosen circuit variables and $\boldsymbol{y}$ is an arbitrary point in $\boldsymbol{R}^n$ which represents the inputs to the circuit. Various methods are available and many computer programs exist for the solution of (3.1). These methods can be classified into two major groups. One is based on an iterative algorithm which is applied directly to the nonlinear circuit equations. The well-known method in this group is the Newton-Raphson method [32]–[35]. The second group is based on the technique of piecewise-linear (PWL) approximation and PWL analysis which has been investigated by many researchers due to its computational efficiency [36]–[48].

In the PWL analysis of nonlinear circuits, the operating region of every nonlinear element is divided into a finite number of segments, and the nonlinear mapping $\boldsymbol{g}(\cdot)$ is approximated by a continuous PWL mapping $\boldsymbol{f}(\cdot)$, that is linear on each segment. As a result of this approximation, the space $\boldsymbol{R}^n$ is divided into N linear regions bounded by hyperplanes. In general, N is a very large number. Then, the system of PWL equations

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{y} \tag{3.2}$$

can be expressed by the following set of linear simultaneous equations

$$\boldsymbol{A}_l\,\boldsymbol{x} + \boldsymbol{w}_l = \boldsymbol{y}, \qquad \text{for } \sigma_l, \qquad l = 1, 2, \cdots, N \tag{3.3}$$

where $\boldsymbol{A}_l$ is a constant $n \times n$ matrix (called Jacobian matrix for convenience) and $\boldsymbol{w}_l$ is a constant n-vector. They characterize the circuit in linear region σ_l. Several methods

have been developed for solving (3.3). Some of the existing methods can only be applied to a restricted class of PWL resistive circuits with a unique solution or with topological limitations [40]–[44]. These restrictions are usually imposed in order to obtain a more efficient algorithm. To find all solutions of (3.2), one may solve n linear simultaneous equations in (3.3) for each of N linear regions to find $\boldsymbol{x}_l$ and decide whether $\boldsymbol{x}_l$ lies within the considered linear region, σ_l. If $\boldsymbol{x}_l$ lies within σ_l, it is a valid solution. This method is conceptually simple and finds all existing solutions, but it is computationally complex. Hence, the major issue in PWL analysis is the reduction of complexity. Recently, a number of authors have proposed various methods [36]–[39] to decrease the number of linear regions, N, by a sign test. This test gives a necessary and sufficient condition on the existence of a solution in a given linear region. One of these methods [36] requires more than $O(Nn^2)$ multiplications. Moreover, the sign test is not a simple procedure. A more efficient method is proposed in [37]. Nishi [38] has proposed a method in which the number of multiplications required to find all solutions of (3.3) is $O(Nn)$. Although the method developed in [39] seems to be the best, it is computationally impractical for large PWL circuits containing several thousands of elements. For instance, if the circuit contains 1000 MOS transistors each of which is modeled with 4 segments, then there are 4^{1000} (approximately 10^{600}) linear regions. If the sign test requires at least one multiplication for each linear region, it will take much more than billions of years on today's supercomputers to find the solutions by using these methods.

In this chapter, we present a new algorithm [49], which we call *popcorn*, shown to be more efficient than the existing algorithms of the same generality. This algorithm is globally convergent for a general class of PWL resistive circuits with no restrictions. It is simple and can be easily programmed. In Section 3.1, the method of PWL analysis of nonlinear resistive circuits is reviewed and the Katzenelson algorithm is described. In Section 3.2, the piecewise linear version of the Newton-Raphson algorithm is explained. The POPCORN algorithm, which is particularly geared toward the analysis of large PWL circuits, is presented in Section 3.3. Some numerical examples are given in Section 3.4 to illustrate the effectiveness of the POPCORN algorithm compared to Katzenelson and PWL Newton-Raphson algorithms.

3.1 Katzenelson Algorithm

In PWL analysis, the well-known technique due to Katzenelson [40] has been originally applied to the circuits with two-terminal elements which are strictly monotonic. The PWL approach was further extended to include the resistive circuits of much broader class [43]–[48]. In particular, Fujisawa and Kuh [44] have shown that the Katzenelson's algorithm can be applied to (3.2) and it always converges to a solution as long as the equation has a unique solution. Fujisawa, Kuh, and Ohtsuki [45] have shown that if all the Jacobian matrix determinants $\det \boldsymbol{A}_l$, $l = 1, 2, \cdots, N$ in (3.3) have the same sign, then there exists at least one solution to the equation $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{y}$ and the algorithm also converges. This property is referred to as *the sign condition.* This restriction of the sign condition was later removed in the generalized Katzenelson's method [46],[48].

Previously, PLAWE was using the *Katzenelson algorithm* to find the operating segments of nonlinear elements and compute the DC solution. This algorithm computes the solution for a given input in an iterative manner starting from a valid solution for an arbitrary input. A brief description of the Katzenelson Algorithm is as follows [40],[43]. To determine the solution, we first choose an initial guess $\boldsymbol{x}_0$ in, say, linear region σ_0. We rewrite the Eqn. (3.3) here for convenience

$$\boldsymbol{A}_0 \boldsymbol{x} + \boldsymbol{w}_0 = \boldsymbol{y}, \qquad \text{for } \sigma_0. \tag{3.4}$$

Substituting $\boldsymbol{x}_0$ into (3.4), we obtain

$$\boldsymbol{A}_0 \boldsymbol{x}_0 + \boldsymbol{w}_0 = \boldsymbol{y}_0 \tag{3.5}$$

where $\boldsymbol{y}_0$ is called the initial source vector. Combining (3.4) and (3.5), we have

$$\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{A}_0^{-1}(\boldsymbol{y} - \boldsymbol{y}_0). \tag{3.6}$$

If it happens that the calculated $\boldsymbol{x}$ in (3.6) lies within the linear region σ_0, then $\boldsymbol{x}$ is the correct solution. Usually, it is not, and we proceed with the following iteration. To determine the next starting point, we connect $\boldsymbol{x}_0$ and the calculated $\boldsymbol{x}$ by a straight line.

The next starting point, $\boldsymbol{x}_1$ is the intersection of this straight line and the boundary of linear region σ_0 where we started. Thus

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \lambda_0 \boldsymbol{A}_0^{-1}(\boldsymbol{y} - \boldsymbol{y}_0) \tag{3.7}$$

where λ_0 is a scalar parameter in the range of $0 < \lambda_0 \leq 1$. Katzenelson has proposed that λ_0 may be selected such that the operating point of *only one nonlinear element* goes to the boundary of its present segment. Assuming that $\boldsymbol{x}_1$ lies on the common boundaries of linear regions σ_0 and σ_1, the piecewise linear model of the circuit corresponding to the linear region σ_1 must be used in the next iteration. The point $\boldsymbol{x}_2$ is determined in a similar manner by evaluating $\boldsymbol{x}$ from (3.5) and (3.6) but using $\boldsymbol{x}_1$, $\boldsymbol{A}_1$ and $\boldsymbol{w}_1$ instead of $\boldsymbol{x}_0$, $\boldsymbol{A}_0$ and $\boldsymbol{w}_0$, respectively. Note that in calculating $\boldsymbol{x}_2$ from the new starting point $\boldsymbol{x}_1$, the trajectory always move away from linear region σ_0. This is because of the fact that for circuits with a unique solution the determinant of the Jacobian matrix does not change sign from one linear region to another. Thus, every linear region in the circuit variable space is entered at most once, and this guarantees the convergence of the iteration process for circuits with a unique solution.

This process continues until a solution is reached. The number of required iterations depends heavily on the distance of the starting point from the correct solution. However, finding a good initial guess is a rather difficult task, particularly for large-sized circuits. Furthermore, the Katzenelson algorithm has no guarantee of convergence for circuits with multiple solutions such as flip-flops. In fact, Katzenelson has proved that his algorithm is convergent only for the networks which consist of 2-terminal elements [40].

3.2 PWL Newton-Raphson Algorithm

Newton-Raphson technique is a well-known method for the computer-aided analysis of general nonlinear (i.e., not necessarily PWL) resistive circuits. It is widely used for solving the systems of differentiable nonlinear simultaneous equations. There exists also a piecewise linear version of the Newton-Raphson method [33]. A brief description of the

PWL Newton-Raphson method may be presented as follows. To begin with, an initial linear region, say σ_0, is chosen, in which the circuit is characterized by the equation

$$\boldsymbol{A}_0 \boldsymbol{x} + \boldsymbol{w}_0 = \boldsymbol{y}. \tag{3.8}$$

Solving (3.8) for $\boldsymbol{x}$, we obtain

$$\boldsymbol{x}_1 = \boldsymbol{A}_0^{-1}(\boldsymbol{y} - \boldsymbol{w}_0). \tag{3.9}$$

Observe that the value of $\boldsymbol{x}_1$ depends on the initial linear region σ_0. Hence, the value of $\boldsymbol{x}_1$ can now be used to identify the next linear region, σ_1. In general, assuming that $\boldsymbol{x}_n$ lies in the linear region σ_n, the next point $\boldsymbol{x}_{n+1}$ is calculated from

$$\boldsymbol{x}_{n+1} = \boldsymbol{A}_n^{-1}(\boldsymbol{y} - \boldsymbol{w}_n) \tag{3.10}$$

where $\boldsymbol{A}_n$ and $\boldsymbol{w}_n$ are both defined in linear region σ_n. This iteration process continues until the solution $\boldsymbol{x}_{n+1}$ lies within the linear region σ_n. However, it is well-known that for the continuous case the Newton-Raphson iteration may not converge depending on the initial guess. The same situation may occur in PWL case while applying (3.10), if the initial linear region σ_0 is not close enough to the linear region of the solution. The divergence can be in the form of a cyclic repetition of two or more virtual linear regions. We have observed that the PWL version of the Newton-Raphson method may not converge for some circuits, particularly with multiple solutions. We have tested this method 100 times on a 128-bit shift register circuit which contains 2580 MOS transistors using different initial linear regions. It has converged in only 22 trials, however, the convergence speed was rather high. Hence, the PWL Newton-Raphson method does not guarantee convergence, but if it does converge, it is extremely fast.

3.3 The *POPCORN* Algorithm

We have developed a new algorithm by modifying the PWL Newton-Raphson method to avoid its major drawback, i.e., divergence. Before we present the final version of the algorithm with convergence guarantee, let us give the first version as follows:

1. Initially, choose an arbitrary linear region, let's say, $\sigma_k = \{a_{k,1}, a_{k,2}, \ldots, a_{k,m}\}$ where $a_{k,j}$ represents the segment for the j th element in the k th iteration. Set $k = 0$.

2. Compute $\boldsymbol{x}_{k+1}$ from

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k^{-1}(\boldsymbol{y} - \boldsymbol{w}_k)$$

   Check if $\boldsymbol{x}_{k+1}$ lies in σ_k. If so, then STOP; $\boldsymbol{x}_{k+1}$ is the solution. Otherwise, CONTINUE.

3. Let $\boldsymbol{x}_{k+1}$ lies in the linear region $\sigma'_{k+1} = \{a'_{k+1,1}, a'_{k+1,2}, \ldots, a'_{k+1,m}\}$. The new linear region $\sigma_{k+1} = \{a_{k+1,1}, a_{k+1,2}, \ldots, a_{k+1,m}\}$ is chosen as follows: For $j = 1, 2, \cdots, m$
   If $a'_{k+1,j} = a_{k,j}$ then $a_{k+1,j} = a'_{k+1,j}$.
   If $a'_{k+1,j} \neq a_{k,j}$ then

$$a_{k+1,j} = \begin{cases} a'_{k+1,j} & \text{with probability } 1-p \\ \text{Any other segment} & \text{with probability } p \end{cases}, \quad 0 < p < 1$$

4. Set $k = k + 1$. Go to step 2.

As it is seen, this algorithm accepts the linear region chosen by the PWL Newton-Raphson method most of the time. If the solution found for a nonlinear device satisfies the limits of its assumed segment, it is kept as it is in the next iteration. But, if the solution does not satisfy the assumed segment, the segment in which the present solution lies is chosen with a high probability $(1-p)$. With a small probability (p), any other segment is chosen. Here, the other segments are chosen with equal likelihood. The segment selection procedure for a nonlinear device is independent of the other nonlinear devices. Note that, if $p = 0$ then the algorithm becomes identical to the PWL Newton-Raphson algorithm. Although the random feature of this algorithm seems to prevent the cyclic repetition of the iteration process, we have found a counterexample circuit with no convergence. That circuit, shown in Fig. 3.1, contains two voltage-controlled voltage sources and two tunnel diodes modeled by 3 PWL segments. This circuit has a unique solution, but the algorithm described above cannot find the solution. It must be noted that both the PWL Newton-Raphson and the Katzenelson algorithms fail for this circuit, unless the initial linear region happens to be the correct one.

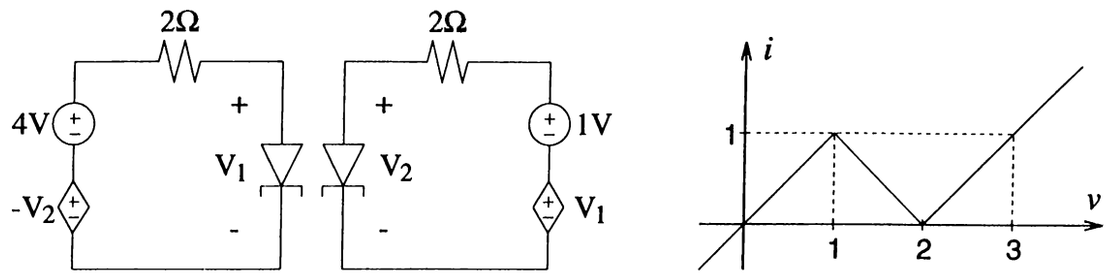


Figure 3.1: Tunnel diode circuit and the $i - v$ characteristics of the tunnel diodes.

Then, we have overcome the convergence problem by making a small modification in the third step of the algorithm described above. Consequently, the final version of the third step of the POPCORN algorithm is given below.

3. Let $\boldsymbol{x}_{k+1}$ lies in the linear region $\sigma'_{k+1} = \{a'_{k+1,1}, a'_{k+1,2}, \ldots, a'_{k+1,m}\}$. The new linear region $\sigma_{k+1} = \{a_{k+1,1}, a_{k+1,2}, \ldots, a_{k+1,m}\}$ is chosen as follows: For $j = 1, 2, \cdots, m$
   If $a'_{k+1,j} = a_{k,j}$ then

$$a_{k+1,j} = \begin{cases} a'_{k+1,j} & \text{with probability } 1-q \\ \text{Any other segment} & \text{with probability } q \quad , \quad 0 < q < 1 \end{cases}$$

   If $a'_{k+1,j} \neq a_{k,j}$ then

$$a_{k+1,j} = \begin{cases} a'_{k+1,j} & \text{with probability } 1-p \\ \text{Any other segment} & \text{with probability } p \quad , \quad 0 < p < 1 \end{cases}$$

With this modification, a segment may not be chosen, albeit with a very small probability q, for the next iteration, even though the present solution satisfies limits of the segment.

The POPCORN algorithm assures the convergence for any initial guess since the algorithm tries all of the linear regions eventually, until it converges. Having such a feature, it resembles the well-known simulated annealing algorithm without cooling [50]. The convergence proof is trivial, since the probability of visiting the linear region containing the solution is non-zero. In the worst case, the algorithm visits all of the linear regions and convergence is always assured. This simple proof does not tell us how fast the algorithm converges, it merely shows that it converges eventually. The important

point is that the POPCORN algorithm uses PWL Newton-Raphson algorithm as the basis. In each iteration, the PWL Newton-Raphson method selects a new linear region to be used in the next iteration and our algorithm makes a random perturbation on that linear region by means of the parameters p and q. This perturbation technique prevents the algorithm from going into a cyclic repetition of virtual linear regions.

The POPCORN algorithm is convergent for any value of p and q as long as they are non-zero. Obviously, these parameters should be appropriately selected to improve the speed of the algorithm. We have made many experiments for different type and size of circuits by changing the values of p and q. The results obtained are very encouraging as can be concluded from the numerical examples given in the following section.

3.4 Numerical Examples

We have implemented the POPCORN algorithm in C programming language and analyzed various CMOS, ECL and analog bipolar circuits. Some of these circuits have multiple solutions. Let us describe the example circuits briefly. **Counter** is a combinational circuit which finds the number of one's in a 128-bit input and it has 4616 MOS transistors. The circuit **lfsr** is a linear feedback shift register which produces pseudo-random binary numbers and it contains 2662 MOS transistors. **Sh128** is a 128-bit shift register circuit consisting of master-slave flip-flops and it has 2580 MOS transistors. **Pgen** is a pulse generating circuit having 1678 MOS transistors in it. **Addcs** circuit is a carry-select adder which contains 770 MOS transistors. The circuit **rsync** is used to produce a synchronization pulse and it has 500 MOS transistors. **Add18** is an 18-bit adder circuit containing 414 MOS transistors. **Status** is a 5-bit register circuit which can be loaded in series or in parallel and it has 122 MOS transistors. **Sh5** is a 5-bit shift register circuit which contains 102 MOS transistors. These circuits are solved thousands of times using the POPCORN algorithm to select the values of p and q. The results given below are obtained using approximately 6500 hours of CPU time on a number of SUN *Sparc*-2+ workstations.

3.4.1 Choosing Parameter p

The MOS transistors are modeled with 4 PWL segments representing the cutoff, linear, forward and reverse saturation states. Let us define $\bar{q}$ as equal to q times the number of nonlinear elements in a given circuit. First, we have changed the value of parameter p while $\bar{q}$ is kept constant at 0.005. The simulation results of the algorithm for the example circuits are shown in Fig. 3.2. The horizontal axis denotes the value of p and the vertical axis shows the average number of iterations, $\bar{k}$, required to find the solution. Note that, due to the random nature of our algorithm, the number of iterations required for the same circuit may be different for different runs. The plots in Fig. 3.2 are obtained by taking the mean of more than 200 simulation results for every circuit at chosen values of p and $\bar{q}$. We have observed that the mean value does not change more than five percent after 200 simulations has been performed. As it is seen from Fig. 3.2, for combinational

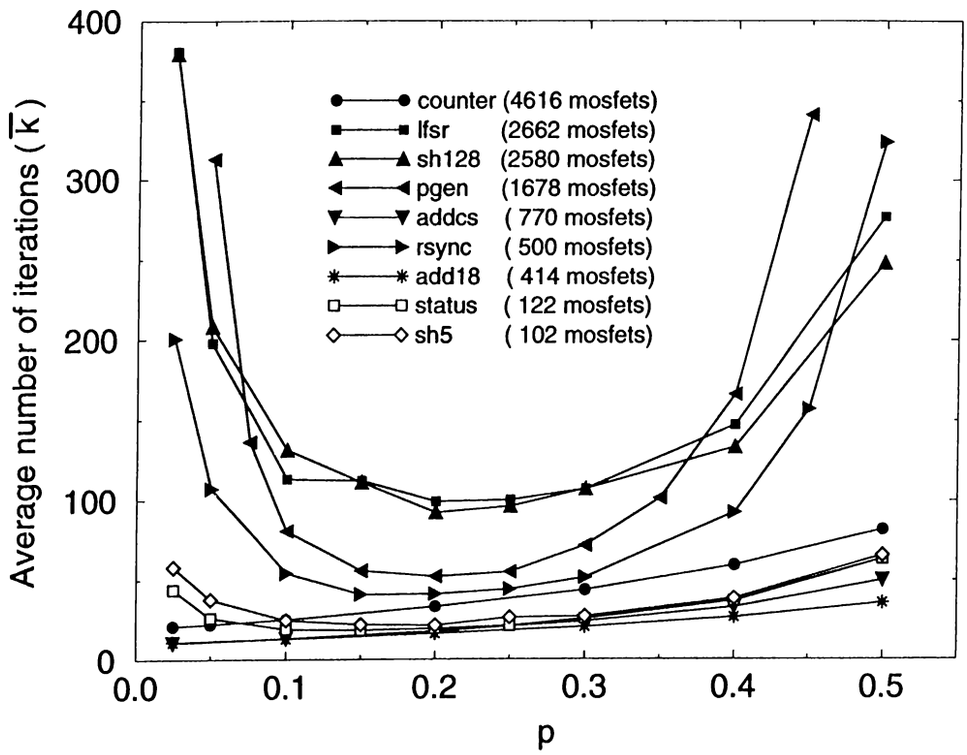


Figure 3.2: Average number of iterations required for the POPCORN algorithm as a function of p using $\bar{q} = 0.005$ and 4-segment PWL MOSFET model.

circuits such as **counter**, **addcs**, and **add18**, $\bar{k}$ increases monotonically with p. For other circuits, $\bar{k}$ reaches a minimum around $p = 0.2$ and it increases sharply as the value of p goes to 0 or 0.5. For 4-segment PWL MOSFET model, the parameter p can be safely set to a value between 0.1 and 0.3.

The standard deviation in the required number of iterations is smaller than half of the mean in the range $0.1 \leq p \leq 0.3$. Hence, the number of iterations required for a given circuit does not change significantly in different trials. For instance, the shift register circuit **sh128** is simulated 200 times using the parameter values $p = 0.2$ and $\bar{q} = 0.005$. The mean and standard deviation of the required number of iterations in 200 simulations are 92 and 45, respectively. To give more detail, Fig. 3.3 shows a histogram obtained from these simulations for **sh128**. In Fig. 3.3, the number of iterations required to find the solution varies between 20 and 198.

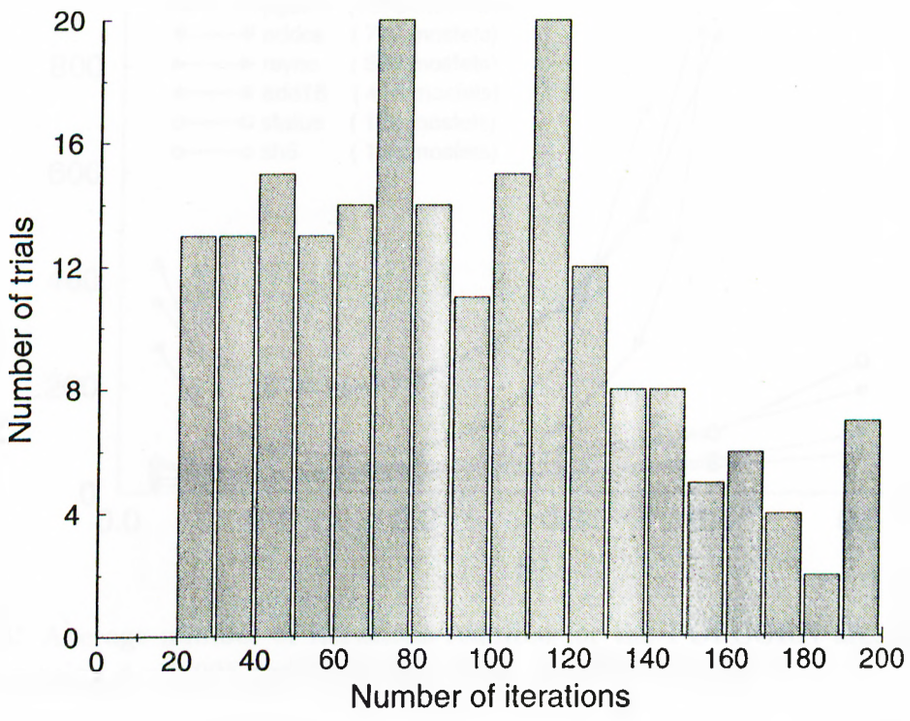


Figure 3.3: The histogram of the number of iterations required in 200 simulation trials for **sh128** circuit using $p = 0.2$ and $\bar{q} = 0.005$.

We have investigated the effect of the number of segments in the PWL MOS model on the selection of the parameter p. For this purpose, we have analyzed the example circuits using 9-segment PWL model for MOS transistors. In the 9-segment model, there are 4 segments in the linear region, and 2 segments each in the forward and reverse saturation regions. The average of more than 200 simulation results for each circuit is given in Fig. 3.4. It is observed that the results for both 4-segment and 9-segment models have similar characteristics. The number of iterations is approximately doubled for 9-segment model. As it is seen from Fig. 3.4, for 9-segment PWL MOSFET model, the minimum occurs around $p = 0.15$ and the parameter p can be set to a value between 0.05 and 0.25.

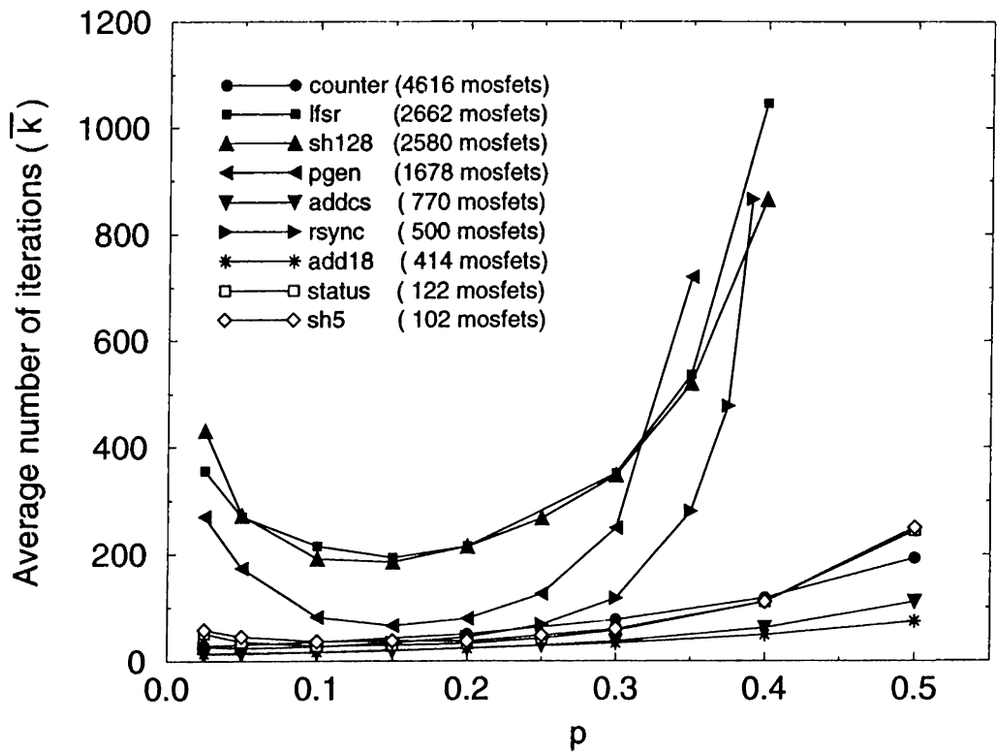


Figure 3.4: Average number of iterations required for the POPCORN algorithm as a function of p using $\bar{q} = 0.005$ and 9-segment PWL MOSFET model.

3.4.2 Choosing Parameter $\bar{q}$

We have selected the value of parameter p in a range where the speed performance of the algorithm is sufficiently good. Now, we have to find a suitable value for parameter $\bar{q}$ which is defined to be q times the number of nonlinear elements in the circuit. To do this, we have analyzed the example circuits by setting $p = 0.2$ and changing the value of $\bar{q}$. Fig. 3.5 shows $\bar{k}$ as a function of $\bar{q}$ using 4-segment PWL MOSFET model. As it can be seen from Fig. 3.5, for all of the circuits except for the tunnel diode circuit, $\bar{k}$ increases as $\bar{q}$ approaches unity. The tunnel diode circuit, however, needs a $\bar{q}$ value close to unity to converge quickly. Therefore, a compromise value of the parameter $\bar{q}$ can be chosen between 0.02 and 0.5. The standard deviation is not larger than half of the mean value in this range.

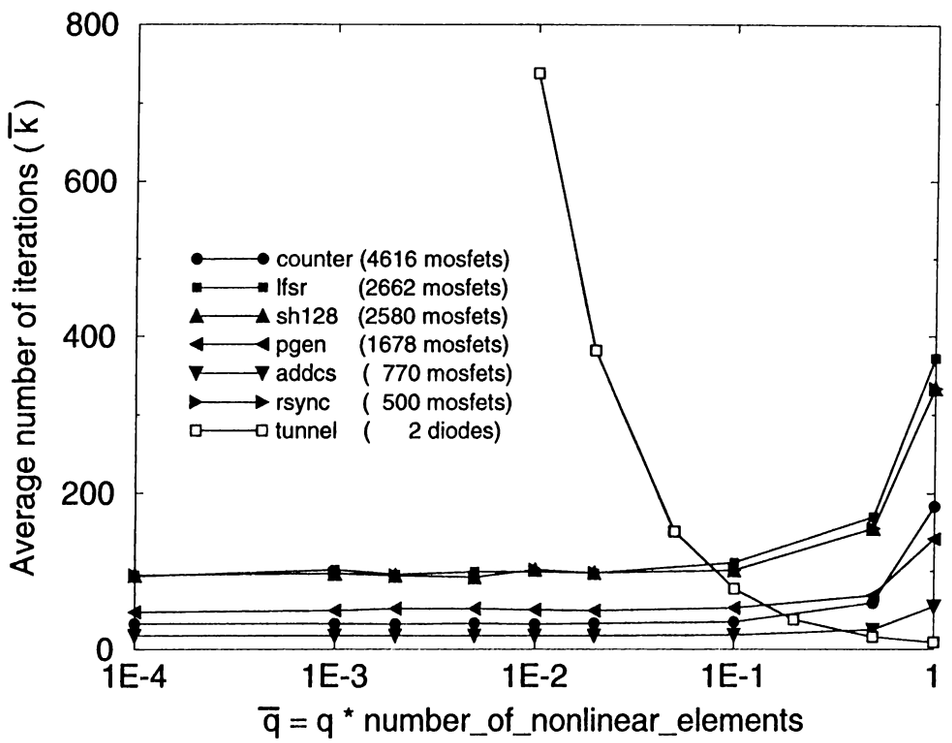


Figure 3.5: Average number of iterations required for the POPCORN algorithm as a function of $\bar{q}$ using $p = 0.2$ and 4-segment PWL MOSFET model.

We have chosen some example circuits to make a performance comparison between the POPCORN, PWL Newton-Raphson and Katzenelson algorithms. First, we have used the circuits **rsync** and **pgen** which have multiple solutions. We have set $p = 0.2$ and $\bar{q} = 0.1$ in the POPCORN algorithm. The results for these circuits are given in Fig. 3.6 and Fig. 3.7. The vertical axis in these figures represents the number of transistors which could not find the correct segment at the corresponding iteration. When this number becomes zero, it means that the solution is found. It is seen from Fig. 3.6 and Fig. 3.7 that the PWL Newton-Raphson and Katzenelson algorithms fail in finding any of the multiple solutions. However, the POPCORN algorithm has converged to one of the solutions in each trial.

Second, we have chosen several CMOS, ECL, and analog bipolar circuits which have a unique DC solution. Table 3.1 gives the number of iterations required to solve these circuits by using three different algorithms. The **control** circuit produces some read/write signals in a correlator chip. **Opamp** is a noninverting amplifier circuit containing a 741 operational amplifier [51]. The MOS transistors and BJT's are modeled with 4 segments while the diodes are modeled using 2 segments. In the POPCORN algorithm, the parameters are chosen to be $p = 0.2$ and $\bar{q} = 0.1$. As it is seen from Table 3.1, the speed of the POPCORN algorithm is close to the speed of the PWL Newton-Raphson method. However, the Katzenelson algorithm is relatively slow compared to the POPCORN algorithm.

Circuit	# nonlinear elements	NUMBER OF ITERATIONS		
		PWL NR	POPCORN	Katzenelson
Control (CMOS)	176 mosfet	6	9	73
Add18 (CMOS)	414 mosfet	11	18	146
Addcs (CMOS)	770 mosfet	10	17	74
Counter (CMOS)	4616 mosfet	20	33	565
4-bit FA (ECL)	102 bjt, 34 diode	12	35	186
Opamp (bipolar)	26 bjt	17	281	not converged

Table 3.1: Number of iterations taken by PWL Newton-Raphson (NR), POPCORN, and Katzenelson algorithms to solve various CMOS, ECL and analog bipolar circuits.

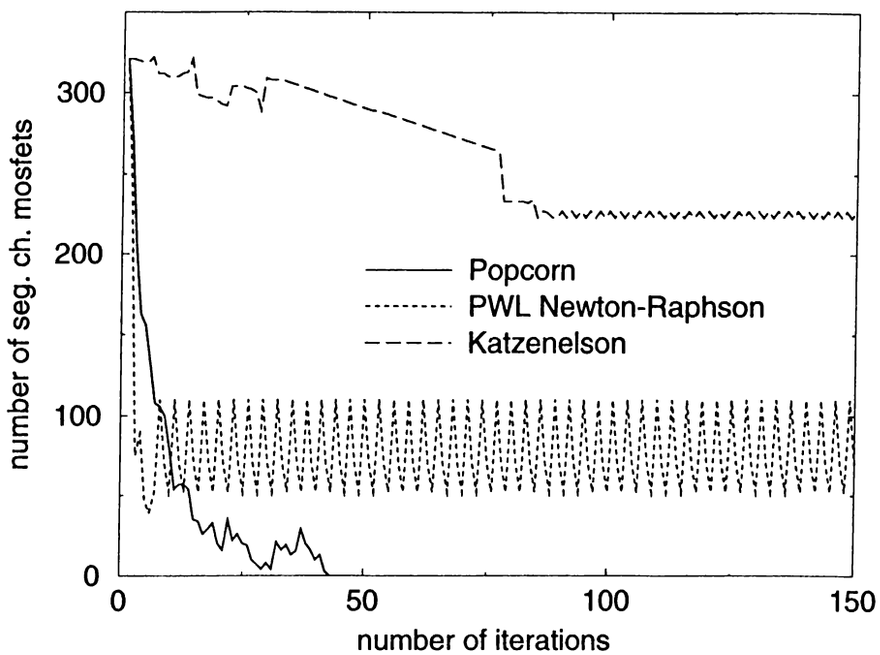


Figure 3.6: The results of the POPCORN, PWL Newton-Raphson, and the Katzenelson algorithms for the circuit **rsync** with 500 MOS transistors.

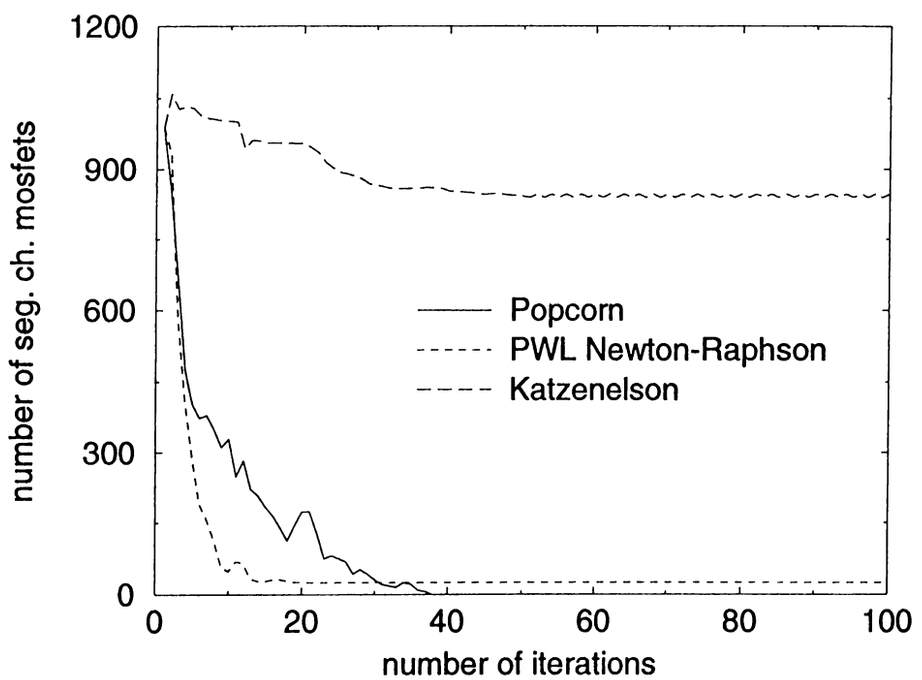


Figure 3.7: The results of the POPCORN, PWL Newton-Raphson, and the Katzenelson algorithms for the circuit **pgen** with 1678 MOS transistors.

Chapter 4

ASYMPTOTIC WAVEFORM EVALUATION (AWE)

AWE is a technique for approximating the time domain responses of linear circuits with multiple step and ramp input signals and unrestricted non-equilibrium initial conditions [1],[52],[53]. Although it is a recently proposed method, AWE is shown to be effective in the time domain analysis of large linear(ized) circuits [54] and it has been the subject of many papers [55] - [68]. It has been applied to the analysis of linear interconnect [55]-[57], pole-zero analysis [58]-[60], as well as nonlinear transient simulation [61]-[64]. A survey of all these research works and the evolution of AWE can be found in [53]. AWE uses a form of Padé approximation [69]-[73] rather than numerical integration techniques to find the transient response of a linear circuit in terms of dominant poles and residues. To do this, it matches the initial boundary condition and the first $2q-1$ moments of the actual response to a lower order q-pole model. AWE is most conveniently explained in terms of the differential state equations for a lumped, linear, time-invariant circuit:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{4.1}$$

where $\boldsymbol{x}$ is the vector of state variables and $\boldsymbol{u}$ is the input excitation vector.

Suppose that the particular input is a combination of step and ramp signals in terms of the constant vectors $\boldsymbol{u}_0$ and $\boldsymbol{u}_1$ as

$$\boldsymbol{u}_p(t) = \boldsymbol{u}_0 + \boldsymbol{u}_1 t, \qquad \text{for } t \geq 0 \tag{4.2}$$

which is a common test signal in timing analysis. In general the form of $\boldsymbol{u}_p(t)$ need not be confined to such signals, but this simple class of input excitations is adequate for the investigation of propagation delay and rise/fall time effects. For this particular input, the dynamical system described by (4.1) has the particular solution

$$\boldsymbol{x}_p(t) = -\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{u}_0 - \boldsymbol{A}^{-2}\boldsymbol{B}\boldsymbol{u}_1 - \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{u}_1 t, \qquad t \geq 0. \tag{4.3}$$

For the existence of this particular solution, the $\boldsymbol{A}$-matrix must be nonsingular. That is, the circuit to be solved must have a unique and well-defined solution when all of its capacitances are open-circuited and all of its inductances are short-circuited. The terms $-\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{u}_0$ and $-\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{u}_1$ in (4.3), are the steady-state (i.e., capacitors open-circuited, inductors short-circuited) solutions of the circuit corresponding to the dc inputs $\boldsymbol{u}_0$ and $\boldsymbol{u}_1$, respectively. To calculate $-\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{u}_1$, if we substitute $\dot{\boldsymbol{x}} = \boldsymbol{0}$ and $\boldsymbol{u} = \boldsymbol{u}_1$ into (4.1),

$$\boldsymbol{0} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}_1 \tag{4.4}$$

we conclude with $\boldsymbol{x} = -\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{u}_1$. In terms of circuit variables, replace all of the capacitors with zero valued current sources (i.e. open circuit) and all of the inductors with zero valued voltage sources (i.e. short circuit) and apply a source of value $\boldsymbol{u}_1$ from the input. Then, after solving the circuit, measure the voltages across the capacitors and currents through the inductors. Similarly, to calculate $-\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{u}_0 - \boldsymbol{A}^{-2}\boldsymbol{B}\boldsymbol{u}_1$, we substitute $\dot{\boldsymbol{x}} = -\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{u}_1$ and $\boldsymbol{u} = \boldsymbol{u}_0$ into (4.1),

$$-\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{u}_1 = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}_0 \tag{4.5}$$

and solving for $\boldsymbol{x}$ we obtain $\boldsymbol{x} = -\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{u}_0 - \boldsymbol{A}^{-2}\boldsymbol{B}\boldsymbol{u}_1$.

Now, we need the homogeneous equation to complete the solution of (4.1),

$$\dot{\boldsymbol{x}}_h(t) = \boldsymbol{A}\boldsymbol{x}_h(t) \tag{4.6}$$

with the initial condition

$$\boldsymbol{x}_h(0) = \boldsymbol{x}_0 + \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{u}_0 + \boldsymbol{A}^{-2}\boldsymbol{B}\boldsymbol{u}_1. \tag{4.7}$$

AWE finds an approximation to the solution of (4.6) for i-th state variable as

$$\hat{x}_{h_i}(t) = \sum_{l=1}^{q} k_l e^{p_l t} \tag{4.8}$$

where q is the approximation order which is usually much smaller than order of the circuit, p_l's are the dominant and possibly complex approximate poles for $\hat{x}_{h_i}$ and k_l's are their corresponding residues. The order of approximation, q, and the dominant poles p_l's can be different for each state variable. In obtaining the approximate poles and residues, AWE matches the initial condition and the first $(2q-1)$ integral moments of $\hat{x}_{h_i}$ to those of x_{h_i}. The integral moments are computed recursively as

$$\boldsymbol{m}_{-1} = -\boldsymbol{x}_h(0) \tag{4.9}$$

$$\boldsymbol{m}_{k+1} = \boldsymbol{A}^{-1}\boldsymbol{m}_k\,, \quad \text{for } k = (-1, 0, \cdots, 2q-2) \tag{4.10}$$

In order to compute integral moment $\boldsymbol{m}_{k+1}$, we must multiply $\boldsymbol{m}_k$ with $\boldsymbol{A}^{-1}$. To do this, setting $\boldsymbol{u}(t) = \boldsymbol{0}$ in (4.1) we obtain $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$. Now place $\dot{\boldsymbol{x}} = \boldsymbol{m}_k$ into (4.1) and thus we conclude with

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{m}_k \tag{4.11}$$

In terms of circuit variables, $\boldsymbol{u}(t) = \boldsymbol{0}$ means that all the input sources are killed and $\dot{\boldsymbol{x}} = \boldsymbol{m}_k$ means that the capacitors are replaced with current sources of value $C_i \times [\boldsymbol{m}_k]_i$ and the inductors are replaced with voltage sources of value $L_i \times [\boldsymbol{m}_k]_i$. Then, measure the voltages across the capacitors and currents through the inductors which are the next integral moments. It should be noted that to calculate the integral moments from the recursion formula (4.10), we do not have to invert the energy storage matrix $\boldsymbol{A}$. Instead, we find the DC solution of a circuit obtained by replacing the capacitors and inductors with current and voltage sources, respectively. So, just one LU-factorization and a few Forward and Backward Substitutions (FBS) [74] are sufficient to do this.

An important obstacle with AWE is that it may produce unstable poles even though the circuit is stable [52],[68], which is also a major problem for Padé approximation [73],[75]–[79]. For this reason, we use derivative moments in addition to integral moments in PLAWE. This does not guarantee the stability of AWE but in general, the usage of derivative moments may be useful to obtain stable approximations. The derivative moments are computed recursively as

$$\begin{aligned} \boldsymbol{m}_{-1} &= -\boldsymbol{x}_h(0) \\ \boldsymbol{m}_{k-1} &= \boldsymbol{A}\boldsymbol{m}_k \ , \quad \text{for } k = (-1, -2, \cdots, -2q) \end{aligned} \tag{4.12}$$

In order to find the derivative moment $\boldsymbol{m}_{k-1}$, we need to perform a multiplication of $\boldsymbol{m}_k$ by $\boldsymbol{A}$. This is done by setting $\boldsymbol{u}(t) = \boldsymbol{0}$ and replacing $\boldsymbol{x} = \boldsymbol{m}_k$ in (4.1). Similarly, as in the case of integral moments, we conclude with

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{m}_k \tag{4.13}$$

In other words, kill all the input sources and replace the capacitors and inductors with voltage and current sources of value $[\boldsymbol{m}_k]_i$, respectively. After solving the circuit, divide the currents through capacitors and the voltages across inductors by C_i and L_i, respectively. They are the next derivative moments. That is, we do not need to compute the matrix $\boldsymbol{A}$ to obtain the derivative moments since we calculate each of them by solving a DC circuit which requires only one FBS.

The relation between the moments and the poles and residues for i-th state variable is given as:

$$\begin{aligned} -\left(k_1 p_1^{2q-1} + k_2 p_2^{2q-1} + \cdots + k_q p_q^{2q-1}\right) &= [\boldsymbol{m}_{-2q}]_i \\ \\ -\left(k_1 p_1 + k_2 p_2 + \cdots + k_q p_q\right) &= [\boldsymbol{m}_{-2}]_i \\ -\left(k_1 + k_2 + \cdots + k_q\right) &= [\boldsymbol{m}_{-1}]_i \\ -\left(\frac{k_1}{p_1} + \frac{k_2}{p_2} + \cdots + \frac{k_q}{p_q}\right) &= [\boldsymbol{m}_0]_i \end{aligned}$$

$$-\left(\frac{k_1}{p_1^{2q-1}}+\frac{k_2}{p_2^{2q-1}}+\cdots+\frac{k_q}{p_q^{2q-1}}\right) = [\boldsymbol{m}_{2q-2}]_i \tag{4.14}$$

where $(\boldsymbol{m}_{-2q}, \boldsymbol{m}_{-2q+1}, \cdots, \boldsymbol{m}_{-2})$ are the derivative moments, $(\boldsymbol{m}_0, \boldsymbol{m}_1, \cdots, \boldsymbol{m}_{2q-2})$ are the integral moments and $\boldsymbol{m}_{-1}$ is the initial condition. It is seen that there are $2q$ unknowns (i.e. q poles and q residues) in (4.14). Therefore, we use only $2q$ equations out of those given in (4.14). The calculation of the poles and the residues given the moments is explained in detail in [1].

Note that the integral moments give information about the integrals of the actual response $x_{h_i}(t)$. They correspond to the coefficients of the Taylor series expansion of the Laplace Transform of the actual response, $X_{h_i}(s)$, around $s = 0$. The derivative moments, on the other hand, are the derivatives of the actual response at $t = 0$, and correspond to the coefficients of the expansion of $X_{h_i}(s)$ in s^{-1}. Combining the derivative moments with the integral moments does not change the procedure used by AWE to calculate the approximate poles and residues [80].

The important point is that in the AWE technique, the cost of finding the particular solution (4.3) and all of the integral or derivative moments is one and only one LU-factorization and several FBS's. In fact, the efficiency of AWE arises from this point.

Chapter 5

TRANSIENT ANALYSIS WITH STATIC PWL MODELING

In transient analysis, PLAWE uses Asymptotic Waveform Evaluation (AWE) [1],[53] instead of the numerical integration techniques. AWE can handle only linear(ized) circuits, while the time domain analysis problem generally is concerned with the nonlinear circuits. Therefore, PLAWE utilizes PWL approach to exploit the efficiency of AWE in the transient simulation of nonlinear circuits. For this purpose, we have developed two different piecewise linear modeling schemes, which we call *static PWL modeling* and *dynamic PWL modeling*, for nonlinear elements such as diodes, transistors, etc.. In static PWL modeling scheme, the user defined PWL models for the nonlinear devices are not changed during the simulation. In dynamic PWL modeling scheme, PWL models for nonlinear elements are changed automatically by means of an error criterion during the simulation. Hence, it is aimed that the accuracy of the simulation can be improved by linearizing the nonlinear devices about their present operating points. The transient analysis using static PWL models is presented in this chapter while the transient analysis using dynamic PWL models will be described in the next chapter.

In transient analysis, PLAWE computes the time domain response of a given

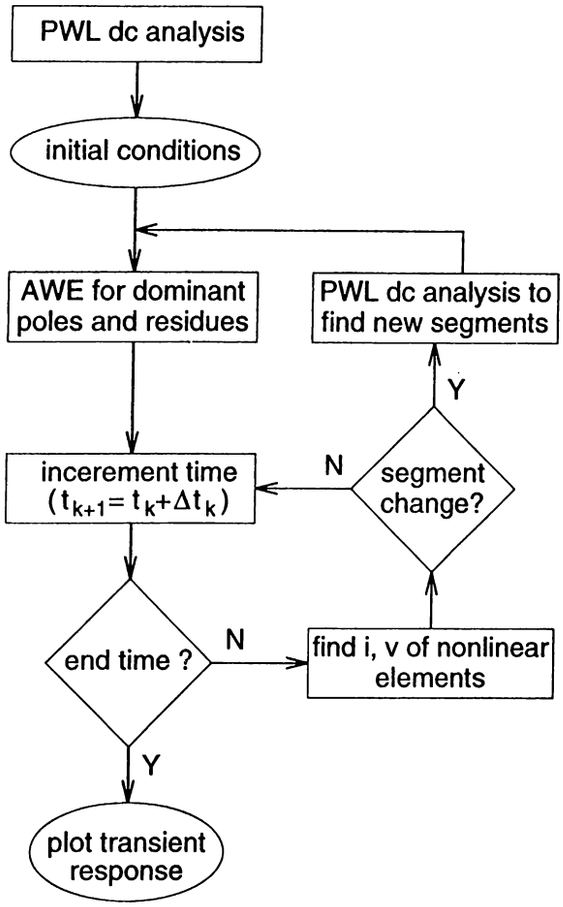


Figure 5.1: Flowchart of the transient analysis with static PWL modeling.

circuit over a user specified time interval $(0, T)$. The transient solution is determined computationally by dividing the time interval $(0, T)$ into discrete time points $(0, t_1, t_2, \cdots, t_n, \cdots, T)$. In PLAWE, any independent source can be assigned a time-dependent value for transient analysis. Ideal step changes in the time-dependent sources are also handled.

PLAWE follows the steps outlined in the flowchart shown in the Fig. 5.1 for transient analysis. A PWL dc analysis is performed at $t = 0$ to obtain an appropriate set of initial conditions for the capacitors and inductors as well as to obtain the initial segments of PWL devices prior to the transient analysis. The time domain responses of energy storage elements are calculated by using the AWE technique. Using AWE, we obtain approximate

analytic expressions for capacitor voltages and inductor currents, in the form of

$$x(t) = c + \sum_{i=1}^{q} k_i e^{p_i t} \tag{5.1}$$

at $t = t_k$. These approximations are valid over a time interval in which all PWL elements continue to satisfy their operating segments found at $t = t_k$. We increment the time by internal time steps, Δt_k, where $t_{k+1} = t_k + \Delta t_k$ for $k = 1, 2, \cdots$ and hence, we successively visit the time points t_1, t_2, $t_3, \cdots$, etc.. The expressions (5.1) found for energy storage elements are evaluated at each time point t_k and the values obtained are used in a mere substitution to calculate the voltages and currents of other elements in the circuit. Here, the important point is that we proceed through time by Δt_k's making only one FBS at each time point t_k until a PWL element crosses its boundary. Thus, we gain an advantage with respect to the conventional circuit simulators such as SPICE which performs a Newton-Raphson iteration process at every time point t_k and this process requires at least one but usually more than one LU-factorization. When a segment change occurs in a PWL element at $t = t_n$, the capacitor voltages and inductor currents at time t_n become the initial conditions and they are used for a new PWL dc analysis which finds the new segments of PWL devices to be used for $t \geq t_n$. Then, a new AWE is performed at $t = t_n$ using the initial conditions of energy storage elements and the new segments for PWL devices. We proceed with discrete time steps in the same way until another PWL element goes to a new segment and the whole process is repeated up to the end time point T.

A similar procedure must be followed when there is a change in the value of an independent source at time t_n. We evaluate the approximate expressions found for energy storage elements and solve the circuit at time t_n^-. A new dc analysis is performed at time t_n^+, and a new AWE is carried out to be used for $t \geq t_n$. If the independent source value has a non-zero rise time instead of an ideal step change, then the dc analysis at time t_n^+ is not required. In PWL dc analysis, we can use the previous operating point as the initial guess. This saves a lot of computation in dc analysis to find the new operating point.

The selection of internal time step Δt_k in transient analysis is a critical issue from the computation time efficiency standpoint. If the time step is chosen too small, then too many unnecessary computations (FBS) must be performed. Conversely, too large

time steps may cause large errors if there exist high frequency poles with large residues. Another drawback of the large time step is that we may skip an overshoot or a spike of the waveform which may possibly cause a segment change in a PWL device. Therefore, PLAWE dynamically calculates the internal time step at each time point t_k in the transient analysis. In this calculation, we consider primarily the rate of change of the most rapidly changing capacitor voltage or inductor current at $t = t_k$. In general, the internal time step changes continuously during the simulation. As a result of this dynamic selection of the time step, when there are rapid voltage or current variations, the simulator takes small time steps over the time axis.

Another facility of PLAWE is that the user can observe dynamically the operating segments of PWL devices. This facility is a lot of help to the user and we believe that PLAWE provides instructive feedback from this point of view since the solution style is very similar to the hand calculation.

5.1 Examples

In this section, some results obtained by using PLAWE are presented to demonstrate its accuracy and efficiency. The program leaves the accuracy speed trade-off to the user by providing a number of options. The user can define his/her own PWL models for nonlinear devices. In addition, the order of approximation and the number of derivative moments used in AWE are important parameters that affect the accuracy. Input file for PLAWE is the same as SPICE input file, except the model card and the options card.

In order to make a comparison on the accuracy of our simulation results, we have simulated some example circuits by using PLAWE and HSPICE[‡] [81] and the results of both simulators are plotted in the same figure for each example. Throughout the examples, we will denote the waveforms obtained by using HSPICE with solid lines and the waveforms produced by PLAWE with dotted lines.

[‡]HSPICE version H92 is a trademark of Meta-Software, Inc.

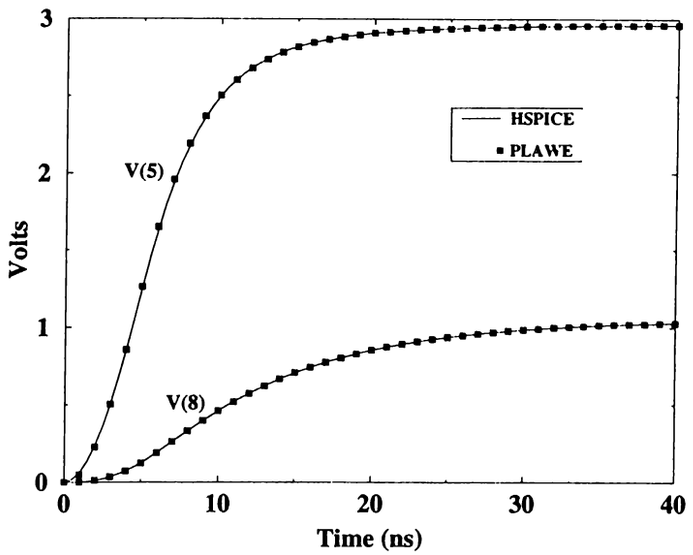


Figure 5.2: Transient analysis results for a large linear RC tree with 9076 elements.

The first example is a large linear RC tree containing 9076 elements. The transient responses observed at the output nodes of this circuit are given in Fig. 5.2. It is seen that PLAWE results overlap with the SPICE results very well.

The second example shown in Fig. 5.3 is a diode transmission gate containing 4 diodes which are modeled with 2 PWL segments. Two pulses with +5V and −5V peaks are applied to the input of this circuit and the resultant output voltage waveform is depicted in Fig. 5.4. As it is seen, PLAWE predicts the output voltage fairly accurately.

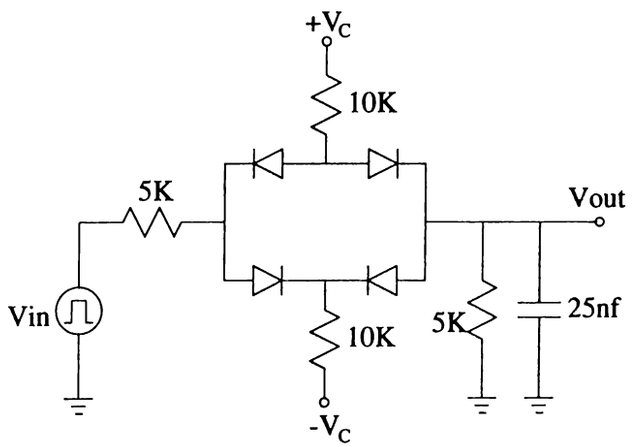


Figure 5.3: A diode transmission gate.

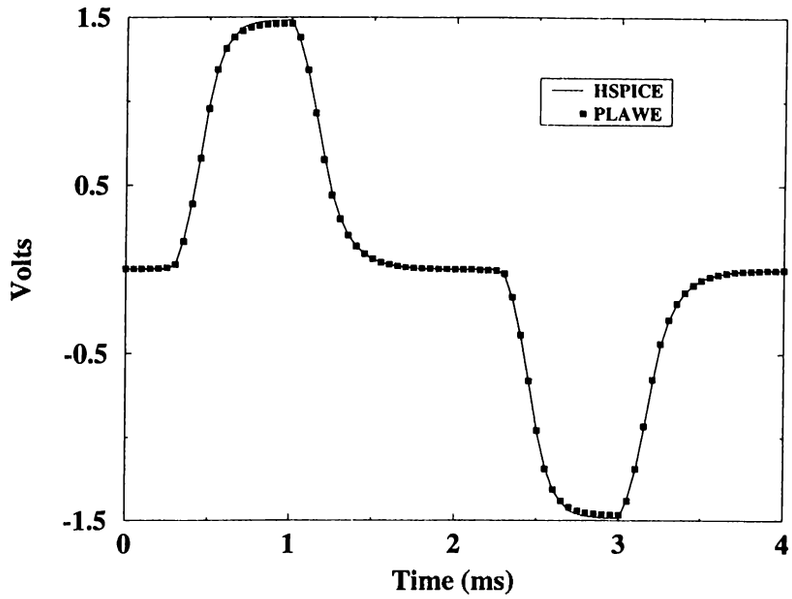


Figure 5.4: Output voltage waveform for the diode transmission gate.

The third example is an ECL EX-OR gate which contains 30 bipolar transistors and 10 diodes. The diodes and BJT's are modeled by using 2 and 4 PWL segments, respectively. The transient response of this circuit is presented in Fig. 5.5.

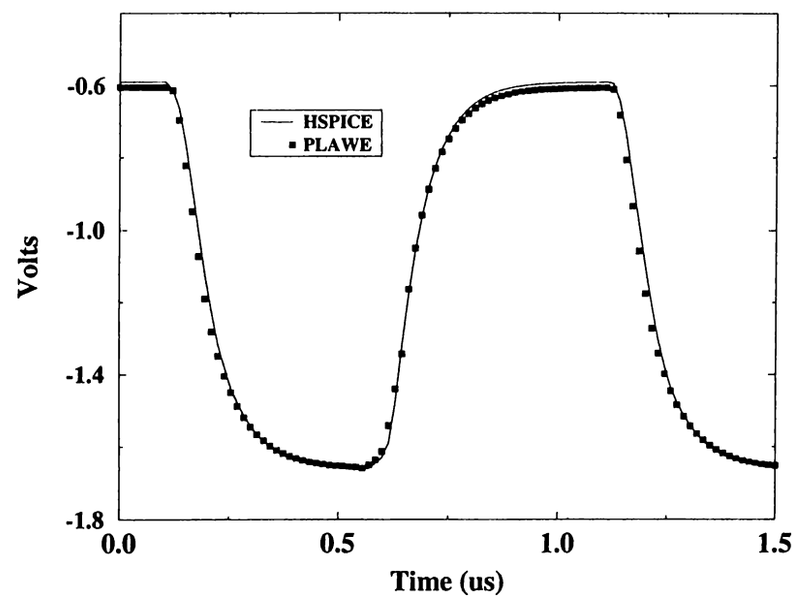


Figure 5.5: Transient response of an ECL EX-OR gate.

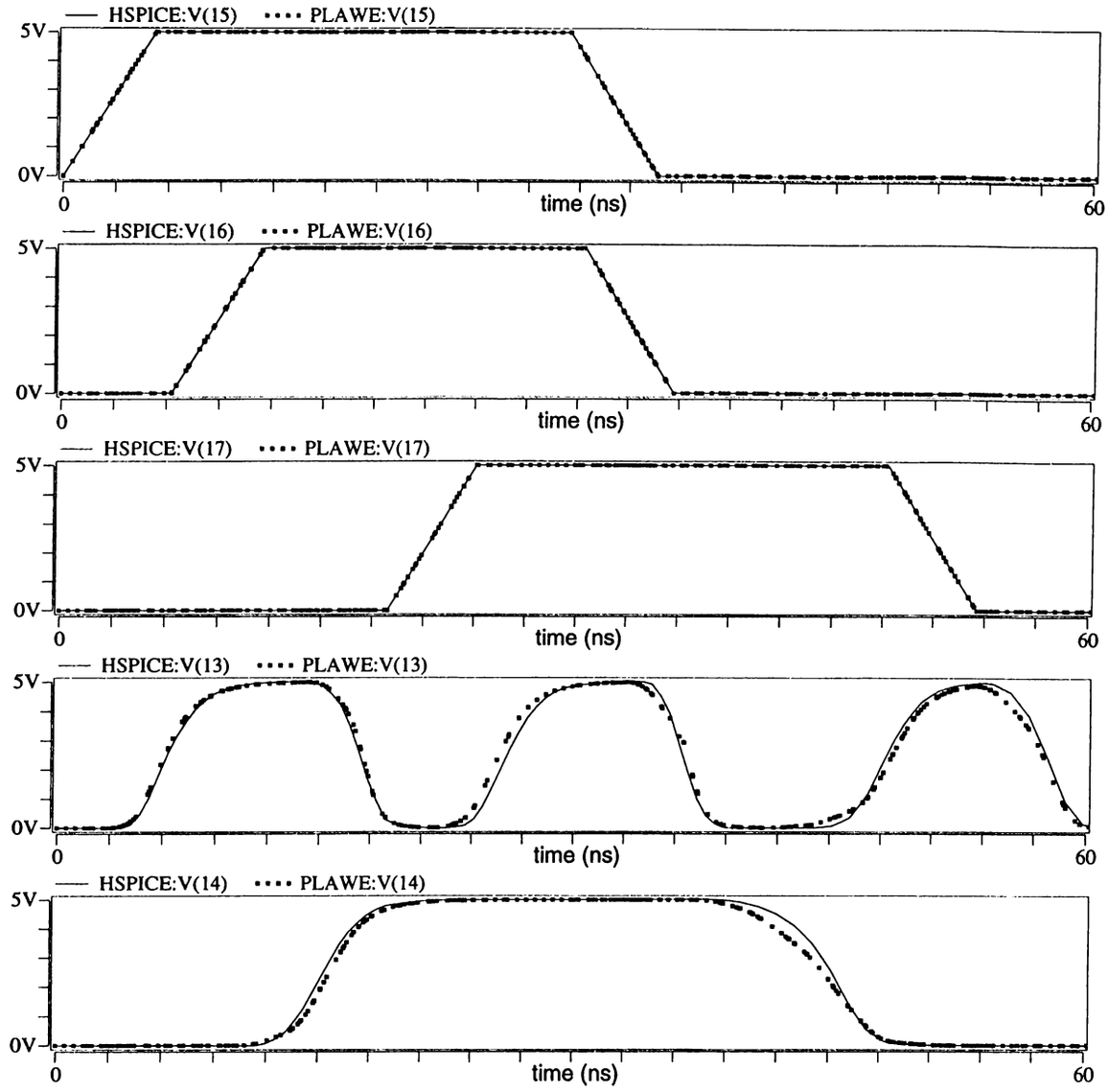


Figure 5.6: The input and output waveforms of the CMOS full-adder circuit.

In the following example circuits, both ntype and ptype MOS transistors are modeled with 4 PWL segments representing the cut-off, linear, forward saturation, and reverse saturation states. The piecewise linear equations and the linear gate-to-ground capacitance C_g in the PWL MOS model is multiplied by the W/L ratio for every transistor.

The fourth example is a CMOS full-adder circuit which contains 28 MOS transistors. The transient simulation results for this circuit are given in Fig. 5.6. In this figure, first

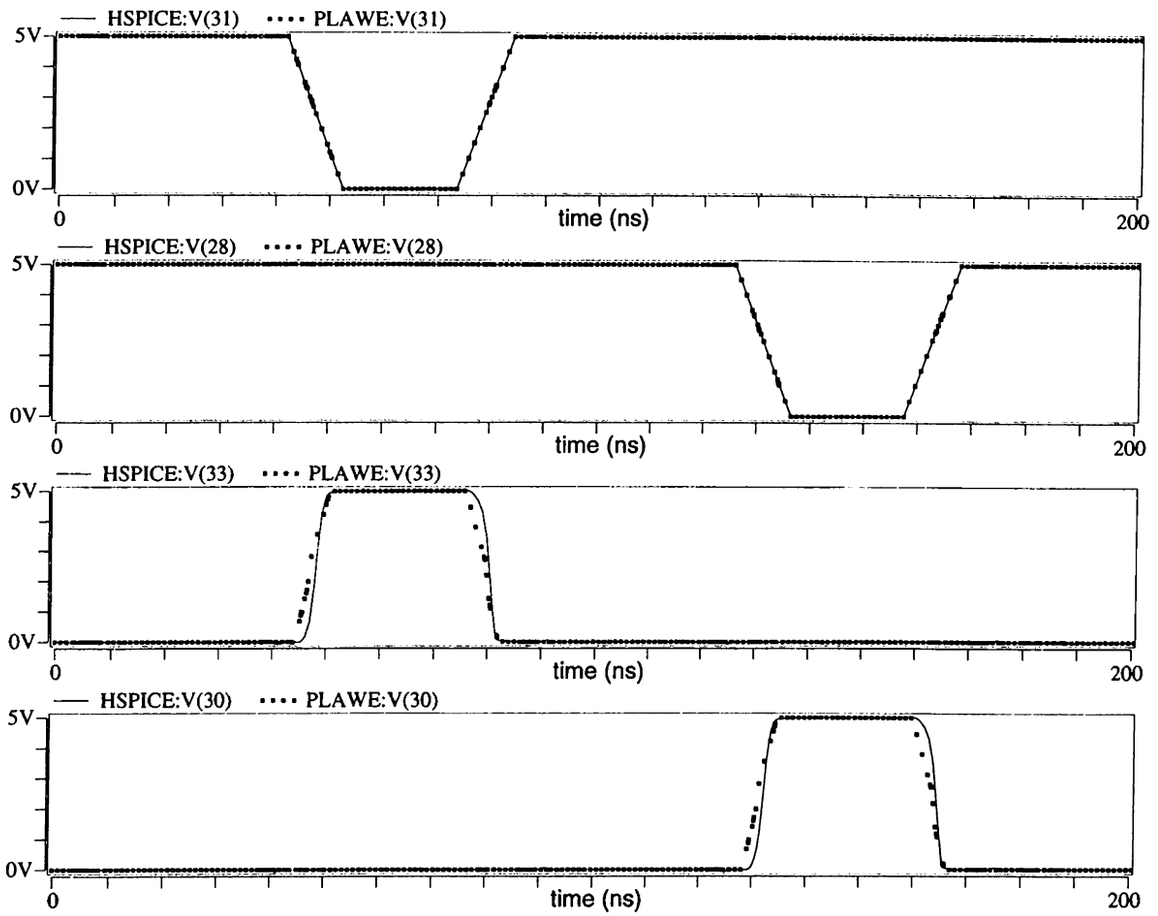


Figure 5.7: Transient simulation results for the address decoder circuit.

three waveforms from the top are the input signals to the circuit, the fourth and the fifth are the SUM and the CARRY outputs, respectively. Propagation delay between the input and output nodes can be observed from the output waveforms. As it is seen from the Fig. 5.6, the simulation results of PLAWE is quite close to those of HSPICE.

The fifth example is an address decoder circuit which consists of 56 MOS transistors. The transient analysis results obtained by using PLAWE and HSPICE for this circuit are given in Fig. 5.7. In this figure, first two waveforms from the top show the input signals and the last two ones show the outputs. As it is seen in the Fig. 5.7, PLAWE simulation results fit very well to the HSPICE results.

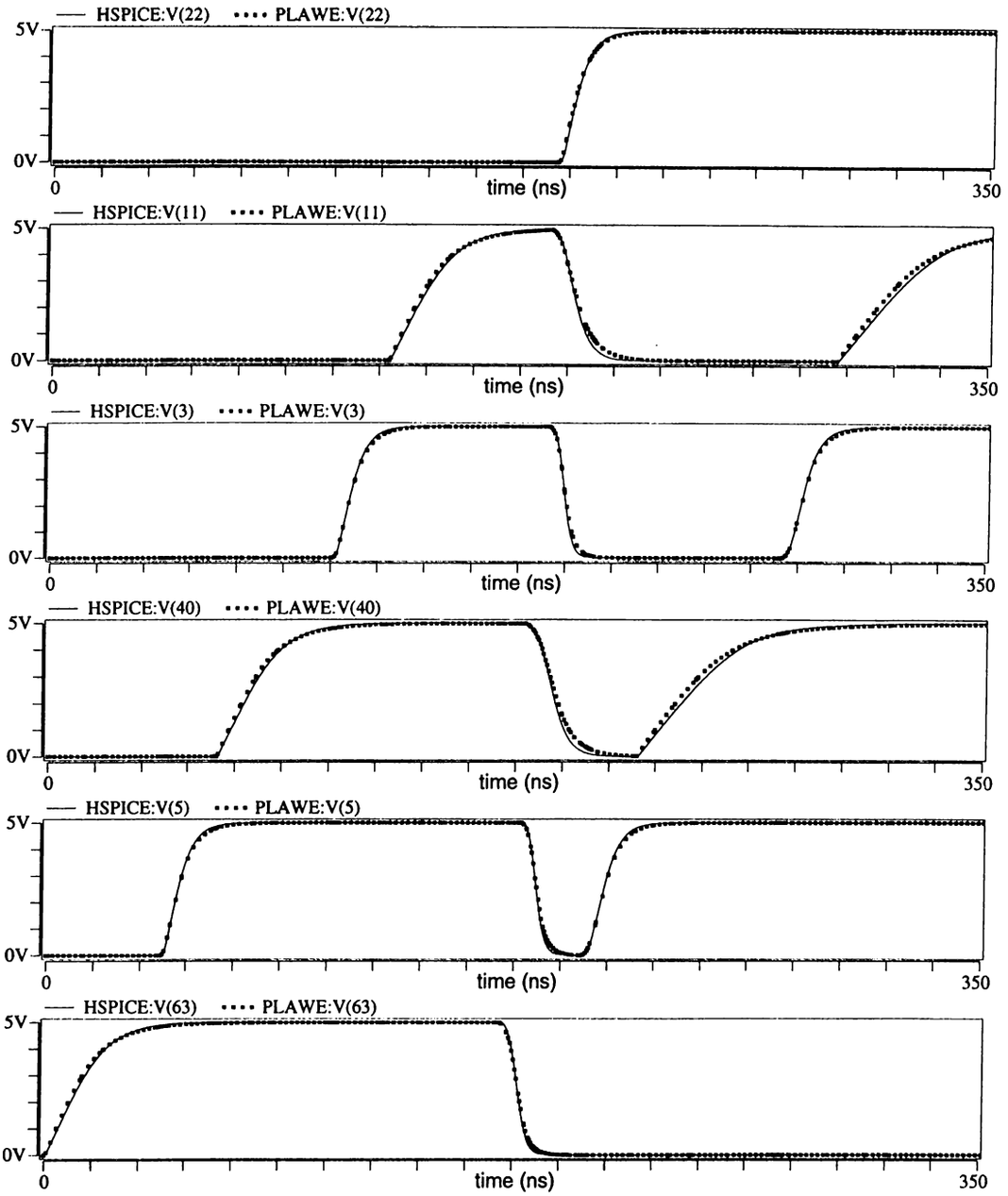


Figure 5.8: Output waveforms of the 5-bit adder circuit.

A 5-bit adder circuit which has 114 MOS transistors is used as the sixth example. Fig. 5.8 shows the transient simulation results for this example. The waveforms plotted in the Fig. 5.8 are observed at the output nodes of the adder. As it is seen, the simulation results of both simulators overlap almost exactly.

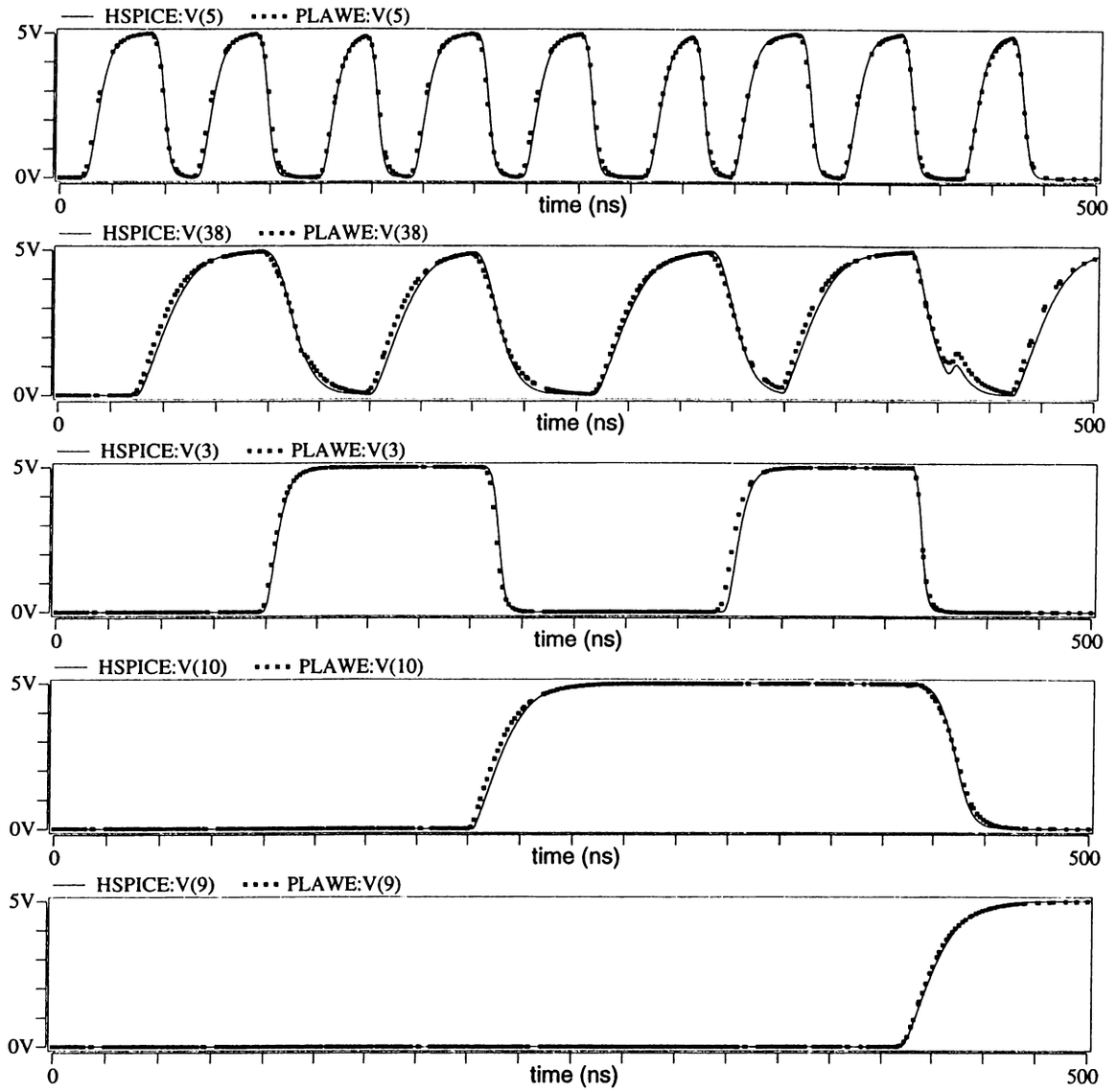


Figure 5.9: Output waveforms for the 18-bit adder circuit.

In Fig. 5.9, we present the transient analysis results for an 18-bit adder circuit which contains 414 MOS transistors. Notice that the waveform showing the voltage of node 38 makes a small ripple towards the end of simulation which is also captured by PLAWE. In this example, PLAWE and HSPICE again produce almost identical results.

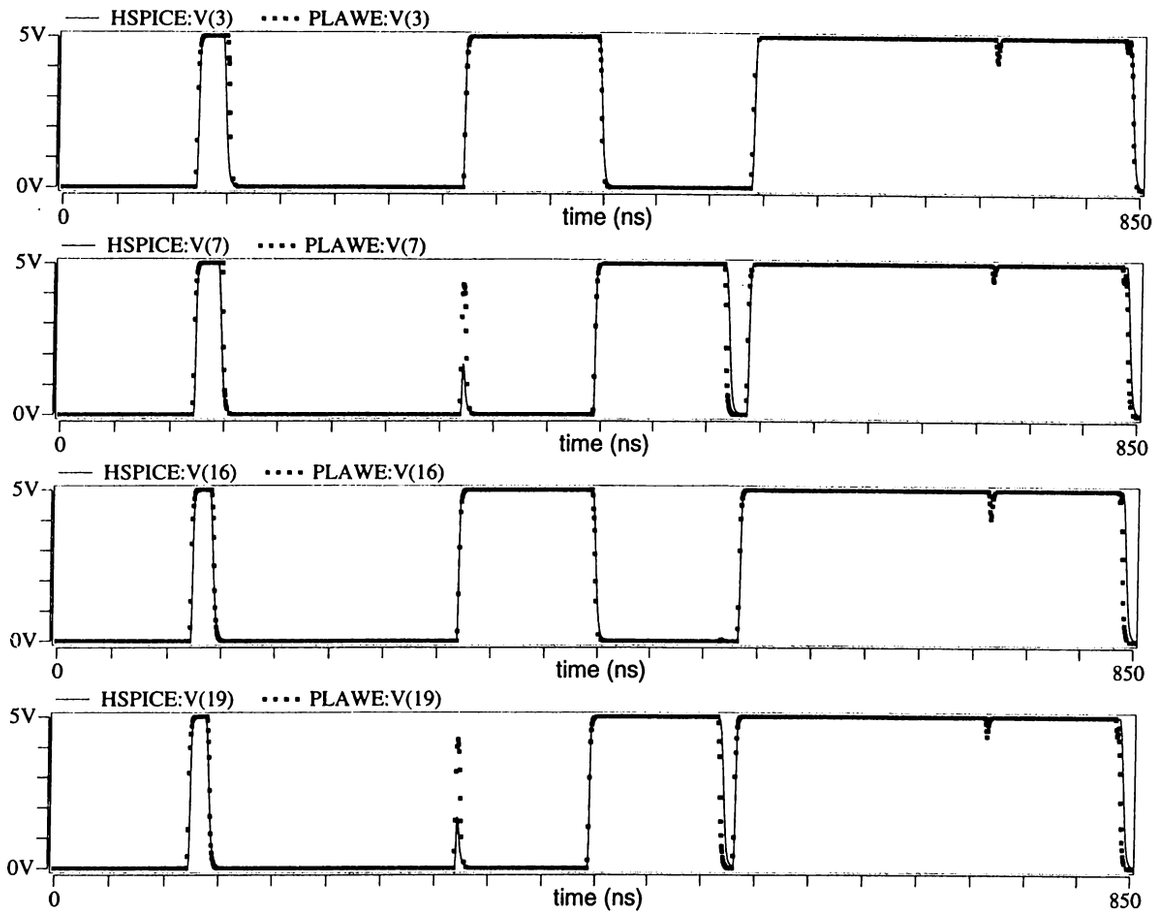


Figure 5.10: Transient response of a carry-select adder containing 770 MOS transistors.

The eighth example is a carry-select adder circuit with 770 MOS transistors. Simulation results produced by PLAWE and HSPICE are shown in Fig. 5.10. This circuit is somewhat hard to solve since it produces some spikes in the output nodes. In spite of this stiffness, both HSPICE and PLAWE detects all those spikes.

For the last example, we have simulated a 128-bit shift register circuit [82] consisting of 128 master-slave (MS) flip-flops. This circuit contains a total of 2580 MOS transistors and its simulation results are given in Fig. 5.11. It is observed that the results produced by PLAWE are fairly good considering the simplicity of the PWL MOS model. In the Fig. 5.11, the first waveform from the top shows the input signal to a powerful clock driver which supplies clock lines to all MS flip-flops. The second waveform is viewed at

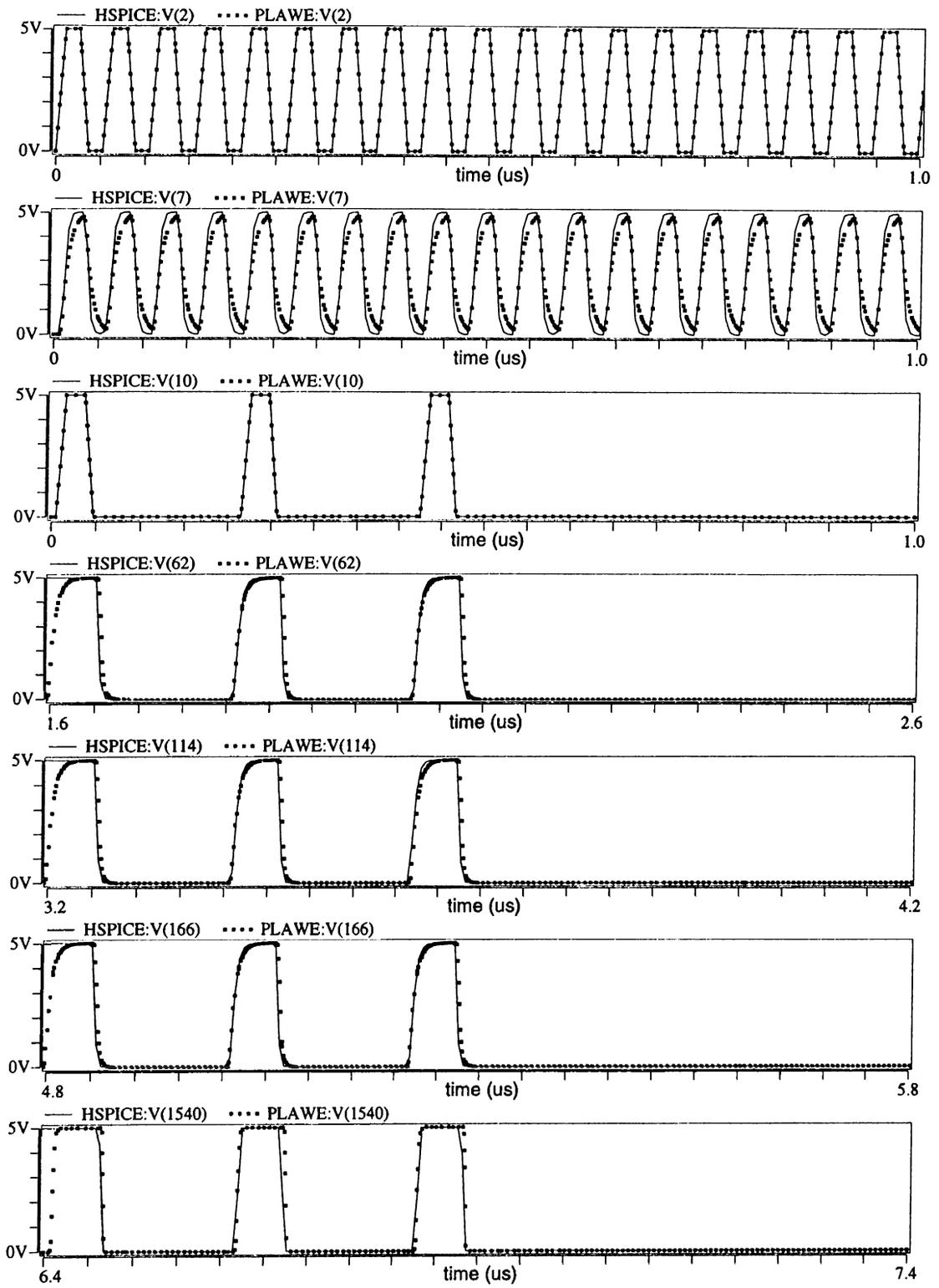


Figure 5.11: Transient analysis results for the 128-bit shift register circuit.

the output of this clock driver. The third one is applied to data input of the first MS flip-flop stage. The fourth, fifth, sixth, and seventh waveforms are observed at the outputs of stages 32, 64, 96, and 128, respectively. It is seen that the PLAWE and HSPICE results for these stages does not coincide initially. This is due to the fact that the shift register circuit has multiple dc solutions at $t = 0$ and our simulator finds a different, but surely a valid, dc operating point prior to the beginning of transient analysis. Therefore, the difference between PLAWE and HSPICE simulations continues until the data input signal propagates up to those stages and then the difference is removed.

These examples demonstrate that the accuracy performance of PLAWE in transient analysis is considerably good. Although we use very simple PWL models for nonlinear devices, the accuracy of the simulation results is rather promising. Now, we evaluate the computational efficiency of PLAWE for the examples given above. Table 5.1 below gives the CPU times taken by both PLAWE and HSPICE for each example circuit. All CPU times are for a SUN *Sparc*-2+ Workstation.

Circuit	# elements	# nodes	CPU TIME	
			PLAWE	HSPICE
Bigrc	9076 RC	5476	107s	171s
Dtrgate	4 diode	7	0.8s	1.4s
Ex-or	30 bjt, 10 diode	80	9.3s	10.9s
Full-Adder	28 mosfet	18	7.3s	9.9s
Decoder	56 mosfet	40	27s	62s
Add5	114 mosfet	68	21s	113s
Add18	414 mosfet	226	484s	1451s
Addcs	770 mosfet	405	1863s	1559s
Shift128	2580 mosfet	1544	57.1h	31.3h

Table 5.1: Run time comparisons between PLAWE and HSPICE.

As it is seen from Table 5.1, for small and intermediate sized circuits, PLAWE is faster than HSPICE. However, when the circuits get larger, HSPICE becomes faster than PLAWE. It seems that the computational complexity of PLAWE is somewhat larger than that of HSPICE. Therefore, we need a new sparse matrix solver which should be faster

than the present one. In addition, Modified Nodal Analysis (MNA) can be used instead of Sparse Tableau Analysis (STA) to formulate the linear circuit equations. We know that the size of STA matrix is $(2b + n)$ while the size of MNA matrix is (n) where b is the number of branches and n is the number of nondatum nodes in a circuit. We have observed that the STA matrix is approximately 5 to 7 times larger than the MNA matrix. It has been shown that the cost of LU factorization is $O(n^{1.4-1.7})$ for sparse matrices that typically arise in circuit simulation [53]. Hence, if we use MNA rather than STA to describe the circuit equations, then the speed of PLAWE will increase approximately one order of magnitude. Another way of reducing the complexity of PLAWE may be to utilize the circuit partitioning technique which leads to the ability to analyze smaller subcircuits separately and to combine them efficiently to obtain an overall analysis of a large circuit.

Chapter 6

TRANSIENT ANALYSIS WITH DYNAMIC PWL MODELING

Previous attempts to apply AWE to the transient analysis of nonlinear circuits [29],[61] solved this problem by using static PWL models for nonlinear elements as described in the Chapter 5. However, finding a good PWL model which fits well to the actual $i - v$ characteristics of a nonlinear device is not an easy task and additionally, static PWL modelling results in low accuracy. The method presented in this chapter uses SPICE models for nonlinear elements to enhance the accuracy of the simulation while preserving the efficiency gain obtained with AWE. Hence, there is no modelling problem and we can adjust the accuracy level by varying some parameters. If the required level of accuracy is increased, more simulation time is needed. Some practical examples are given to illustrate significant improvement in accuracy. For circuits containing especially weakly nonlinear devices, this method is typically at least one order of magnitude faster than HSPICE.

Using the given SPICE models, our method can extract a linear equivalent for every nonlinear element about their bias points. For each nonlinear element, it can also calculate the error caused by the linear equivalent while the operating point moves to any arbitrary direction. We have a simple error criterion used for this purpose which is explained in Section 6.1.2. When the calculated error exceeds some user specified tolerance limits at

any time, the new linear equivalents are produced for all nonlinear elements about their present operating points. Note that if the error limits are decreased, linear equivalents of the nonlinear elements are renewed more frequently as we proceed over time and so accuracy of the simulation will increase.

We describe the method and explain the extraction of linear equivalent circuits from SPICE models of nonlinear elements in the next section. Then in Section 6.2, some examples are provided to illustrate the efficiency and the accuracy of our method.

6.1 The Method

Our method is a new approach using the AWE technique to find the time domain responses of nonlinear circuits containing diodes, transistors, etc.. The flowchart of the new method is given in Fig. 6.1. We first determine the dc operating point by using the Newton-Raphson iteration [28],[32]. This step is the first nonlinear dc analysis which gives the initial conditions of the circuit. Now, we can obtain linearized equivalents for all nonlinear elements about their computed operating points. For a diode, linearization step is simply replacing it by a Norton equivalent which represents the tangent approximation to its $i - v$ curve about the presumed operating point. Linearization of an MOS transistor is explained in the Section 6.1.1. After the linearization step, we have a linearized equivalent circuit for the actual nonlinear circuit. Thus, we can use AWE to find the transient behavior of the energy storage elements in the circuit. AWE yields approximate analytic expressions in the form of (5.1) for all capacitor voltages and inductor currents. These expressions will be valid as long as the linear equivalent circuit is not changed. Then, we increment the time by an amount of internal time step Δt_k and solve the linear circuit equations to find the branch currents and branch voltages of nonlinear elements. This step is necessary to find the new operating points of the nonlinear elements at $t_{k+1} = t_k + \Delta t_k$ and it costs one Forward and Backward Substitution (FBS). At time t_{k+1}, we must check whether the linear equivalent of every nonlinear element duplicates the nonlinear $i - v$ characteristics of the device sufficiently well or not. If at least one of

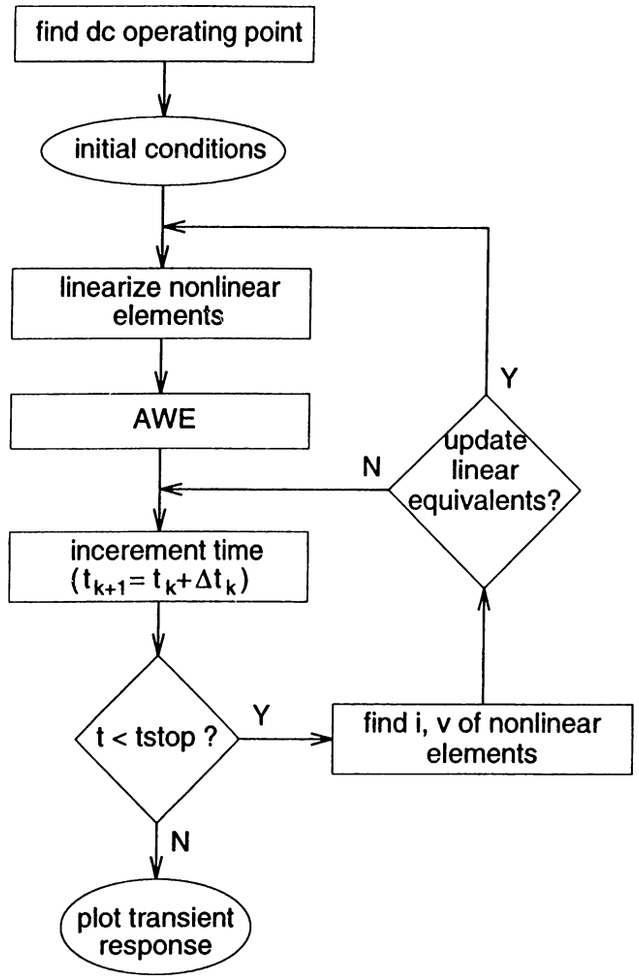


Figure 6.1: Flowchart of the transient analysis with dynamic PWL modeling.

the nonlinear elements has an error greater than a user specified threshold, then we go to the linearization step and create a new linear equivalent for each nonlinear element about their operating points found at $t = t_{k+1}$. However, if none of the linear equivalents used for the nonlinear elements produces an error greater than the threshold value, then we continue in the inner loop by incrementing the time. This decision step is explained in detail in the Section 6.1.2. The procedure above is repeated until the end of the simulation is reached.

6.1.1 Linearization of an MOS Transistor

For simplicity, we have used the Level 1 MOSFET model of the SPICE [81] which represents the basic device characteristics including the body effect and the channel length modulation [27]. The DC drain-to-source current (i_{ds}) in the Level 1 MOS model is determined as follows:

Cutoff Region : $v_{gs} \leq v_t$

$$i_{ds} = 0 \tag{6.1}$$

Linear region : $0 < v_{ds} < v_{gs} - v_t$

$$i_{ds} = \beta \left(1 + LAMBDA \cdot v_{ds}\right) \left(v_{gs} - v_t - \frac{v_{ds}}{2}\right) v_{ds} \tag{6.2}$$

Saturation Region : $0 < v_{gs} - v_t \leq v_{ds}$

$$i_{ds} = \frac{\beta}{2} \left(1 + LAMBDA \cdot v_{ds}\right) \left(v_{gs} - v_t\right)^2 \tag{6.3}$$

where

$$\beta = KP \left(\frac{W}{L}\right). \tag{6.4}$$

The threshold voltage is calculated as follows:

$$v_t = \begin{cases} VTO + GAMMA \left(\sqrt{PHI + v_{sb}} - \sqrt{PHI}\right) & \text{if } v_{sb} \geq 0 \\ VTO + GAMMA \left(0.5 \frac{v_{sb}}{\sqrt{PHI}}\right) & \text{if } v_{sb} < 0 \end{cases} \tag{6.5}$$

where $LAMBDA$, KP, VTO, $GAMMA$, and PHI are the SPICE MOS model parameters [81]. The parameters W and L represent the width and length of an MOS transistor, respectively.

The linear DC equivalent circuit for an n-type MOSFET is given in Fig. 6.2. As it is seen, the transistor is modeled by a voltage controlled current source shunted by a resistor and a constant current source. In this linearized model, drain-to-source current is calculated as follows:

$$I_{ds} = g_m v_{gs} + g_d v_{ds} + I_0 \tag{6.6}$$

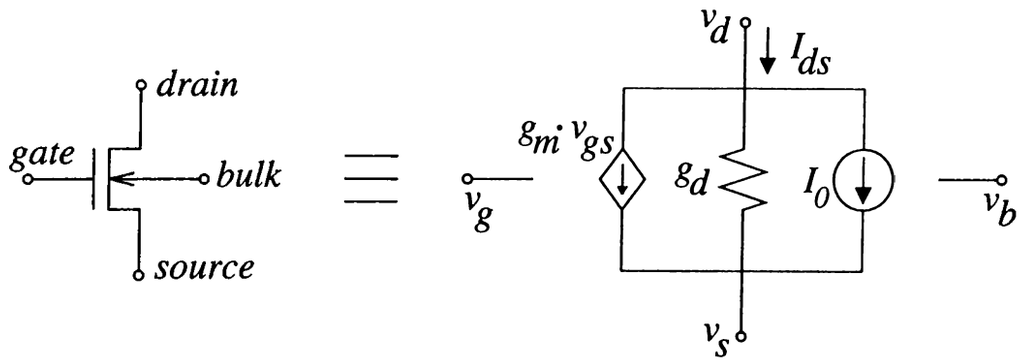


Figure 6.2: DC equivalent circuit used in transient analysis for an n-type MOSFET.

where

$$g_m = \frac{\partial(i_{ds})}{\partial(v_{gs})} \tag{6.7}$$

$$g_d = \frac{\partial(i_{ds})}{\partial(v_{ds})} \tag{6.8}$$

$$I_0 = i_{ds} - g_m v_{gs} - g_d v_{ds}. \tag{6.9}$$

The partial derivatives g_m and g_d are called the transconductance and conductance, respectively and they are calculated in each operating region of the transistor as follows:

Cutoff Region :

$$g_m = g_d = 0 \tag{6.10}$$

Linear region :

$$g_m = \beta\,(1 + LAMBDA \cdot v_{ds})\,v_{ds} \tag{6.11}$$

$$g_d = \beta\,(v_{gs} - v_t - v_{ds} + 2 \cdot LAMBDA \cdot v_{ds} \cdot (v_{gs} - v_t - 0.75\,v_{ds})) \tag{6.12}$$

Saturation Region :

$$g_m = \beta\,(1 + LAMBDA \cdot v_{ds})\,(v_{gs} - v_t) \tag{6.13}$$

$$g_d = \frac{\beta}{2} \cdot LAMBDA \cdot (v_{gs} - v_t)^2 \tag{6.14}$$

6.1.2 Deciding to Update the Linear Equivalents for Nonlinear Elements

As shown in Fig. 6.1, after every time increment, we must decide for each nonlinear element whether its linear equivalent should be updated or not. This decision is made by finding the difference between the actual $i - v$ characteristics of the device and the operating point computed from the linear equivalent. If this difference is greater than a user defined threshold value, then a new linear equivalent is created for every nonlinear element. For an MOS transistor, the difference mentioned above is equal to

$$\delta i = |i_{ds} - I_{ds}| \tag{6.15}$$

where i_{ds} and I_{ds} are calculated from the SPICE MOS Level 1 model and the linear equivalent, respectively using the branch voltages v_{gs}, v_{ds}, and v_{sb}. The difference calculation for a diode is shown schematically in Fig. 6.3. As it is seen in Fig. 6.3, the diode has been linearized about $v_d = v_0$ and it is replaced by the Norton equivalent

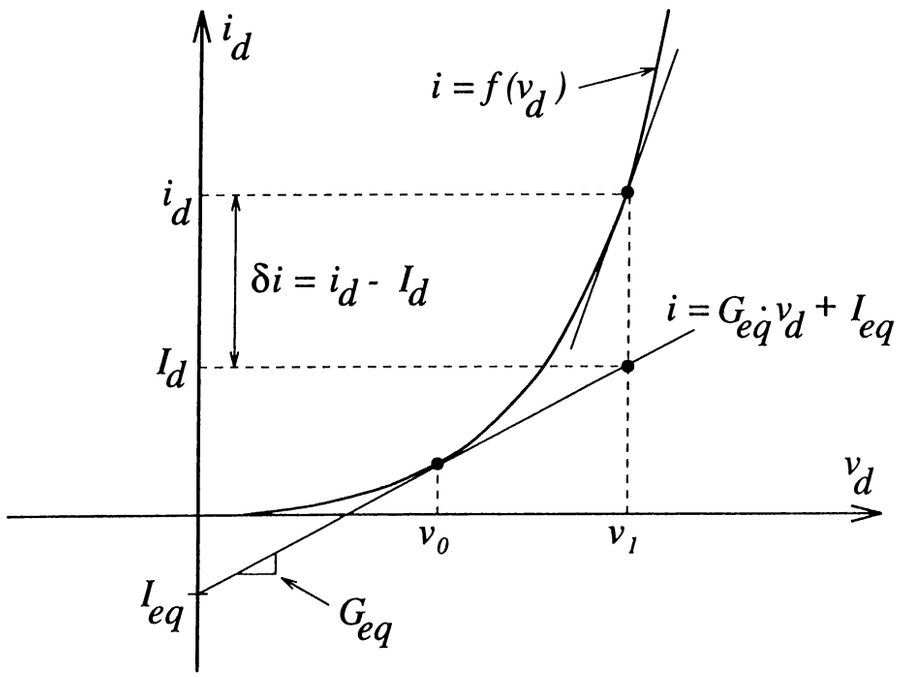


Figure 6.3: Error calculation for a diode equivalent circuit.

which consists of a current source of value I_{eq} shunted by a conductance G_{eq}. In this case, when the diode branch voltage, v_d, becomes equal to v_1, the difference between the linear segment and the nonlinear $i - v$ characteristics is $\delta i = i_d - I_d$. If the value of δi is greater than a user specified error tolerance limit, then a new linearization must be done for the diode at $v_d = v_1$. Note that, to measure the error caused by the linear equivalent, we use the difference in currents instead of voltages because, the calculation of the difference in currents is easier than calculation of the difference in voltages, especially for three terminal elements such as MOS transistors.

6.2 Examples

In order to illustrate the accuracy performance of our method, we have chosen some example circuits. The first example which is an opamp circuit [51] with unity gain feedback is shown in Fig. 6.4. The transistor schematic of the opamp is given in Fig. 6.5. The bulk connections of the MOS transistors and the capacitors from each node to ground are not shown in Fig. 6.5 for simplicity. A SPICE input description of the opamp circuit is given in Appendix. We have used a pulse of small amplitude for the input voltage in order to ensure that the opamp has a weakly nonlinear behavior. That is, operating points of the transistors does not change very much due to small input variations.

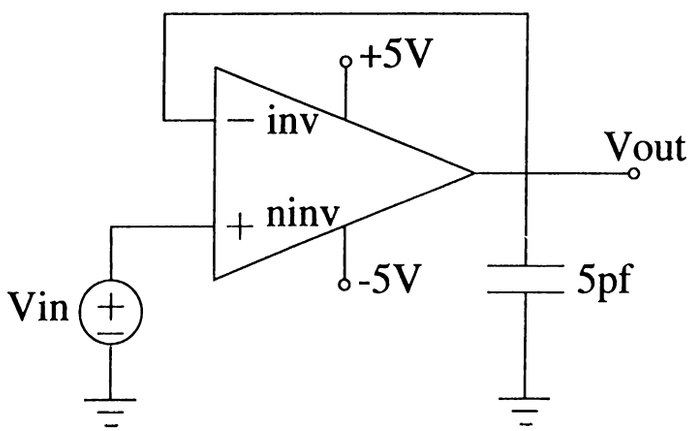


Figure 6.4: Opamp circuit with unity gain feedback.

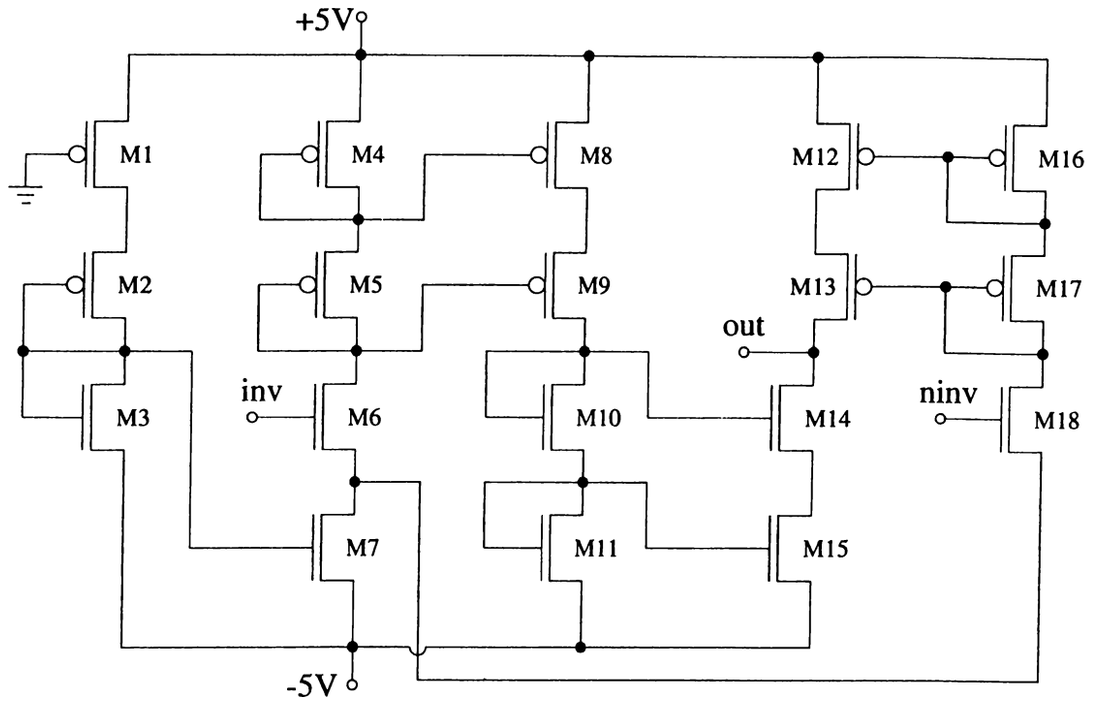


Figure 6.5: Schematic of the operational amplifier in transistor level.

We have simulated this example circuit by using our method, HSPICE[‡] [81], and SPICE3 [83] with different error tolerance limits. First of all, a reference result assumed to be very accurate is obtained by means of HSPICE using very tight error limits and very small internal time step. Then, we have assumed this result to be the exact response of the circuit and all other simulation outputs are compared with this result to estimate their accuracy. The error in a simulation output is calculated by finding the average of absolute differences with respect to the exact response at every time points where the output waveforms are printed.

$$\text{Average absolute difference} = \frac{1}{N}\sum_{k=1}^{N} |v_{exact}(t_k) - v_c(t_k)| \tag{6.16}$$

where $v_{exact}(t)$ and $v_c(t)$ are the exact and calculated responses of the circuit. For all simulations, N is chosen as 1000. Consequently, the accuracy versus number of timepoints for our method, HSPICE, and SPICE3 is plotted in Fig. 6.6 for the opamp circuit. Here, the horizontal axis denotes the number of time points that the simulator needs to take

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

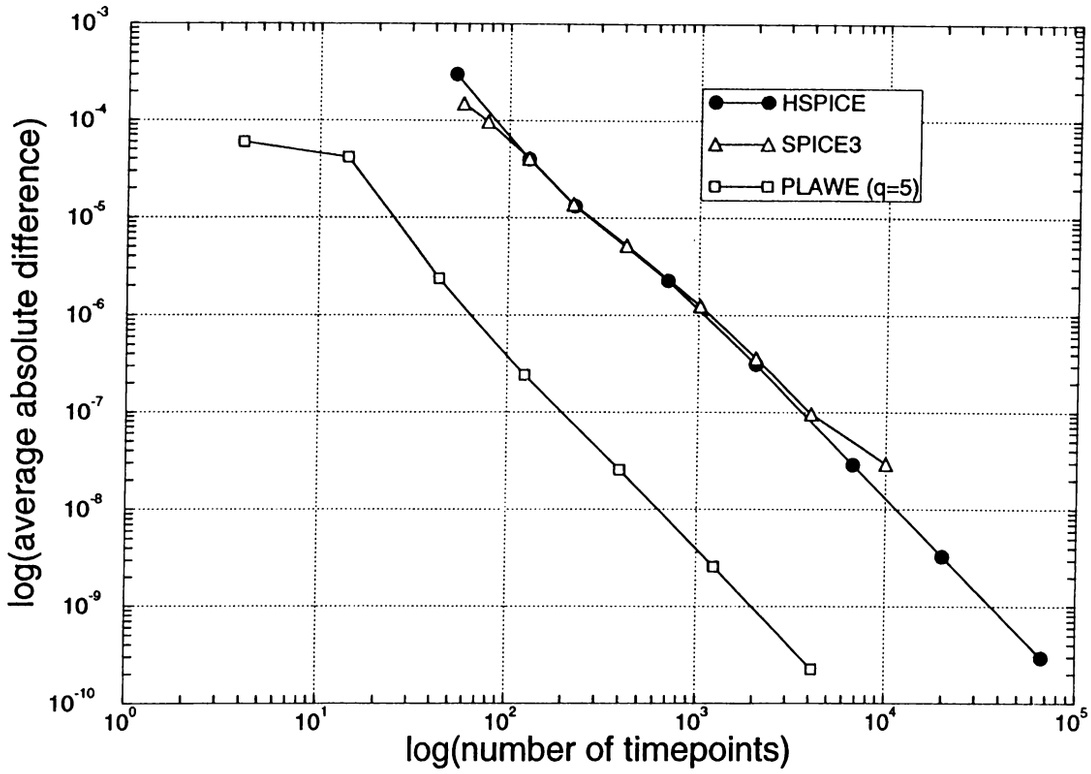


Figure 6.6: Accuracy comparison between our method and SPICE for opamp circuit.

in order to preserve the corresponding accuracy. At every point, HSPICE or SPICE3 performs a Newton-Raphson iteration while our method performs, in addition to Newton-Raphson, an AWE which costs one LU-decomposition and a few Forward and Backward Substitutions (FBS). This means that, for the same number of time points, our method spends approximately 3/2 times of the cpu seconds spent by HSPICE or SPICE3. We have set the order of approximation, q, to 5 in AWE for this example.

It is observed from Fig. 6.6 that HSPICE and SPICE3 have the same accuracy versus speed graphs since both of them are using trapezoidal integration algorithm in the transient analysis. It is seen that our method can produce transient responses which are accurate up to 9 significant digits and it requires approximately 1/20 of the number of iterations needed by HSPICE to provide the same accuracy. If the user agrees to obtain less accurate results such as having an error about 10^{-4}, this ratio becomes 1/30. Thus,

our method is approximately 20 times faster than HSPICE or SPICE3 when the accuracy is less important than the simulation time.

Our second example given in Fig. 6.7 is a small RC tree driven by a CMOS inverter. This circuit is chosen as an example to explore the effect of inserting nonlinear elements to a linear circuit for which the AWE technique provides very accurate results, efficiently. Again, by using HSPICE, we have obtained a reference result which is assumed to be extremely close to the exact result. Then, we have simulated the example circuit using our method and HSPICE by changing the error tolerance parameters. These simulation results are compared with the reference result to estimate their accuracy levels. Thus, we have plotted the graph in Fig. 6.8 which shows the accuracy versus number of timepoints required by each simulator. It is observed that if the desired accuracy is low, our method is several times faster than HSPICE. However, when the accuracy is increased, both of the simulators need approximately equal number of timepoints to provide the same accuracy.

As the third example shown in Fig. 6.9, we have inserted additional MOS transistors into the second example to increase the number of nonlinear elements in the circuit. In a similar way as in the previous example, we have obtained the graph of accuracy versus number of timepoints required by our simulator and HSPICE. The resultant graph is shown in Fig. 6.10. It can be observed from Fig. 6.8 and Fig. 6.10 that increasing the number of nonlinear elements inserted to a linear circuit causes a degradation in the speed performance of our method. Because, the overall nonlinearity of the circuit is increased by additional MOS transistors and due to this modification, we have to update the linear equivalents for the nonlinear elements more frequently as we proceed over time.

These examples show that if the desired accuracy is low, our method is approximately 20 times faster than HSPICE for weakly nonlinear circuits and when the nonlinearity of the circuit is increased, it becomes several times faster than HSPICE. Actually, an accuracy corresponding to an error about 1% is usually acceptable for most of the digital circuits and this much accuracy is generally higher than the one which can be obtained by using static PWL modeling technique. In addition, the speed of our method can be increased by using a better internal time step control algorithm.

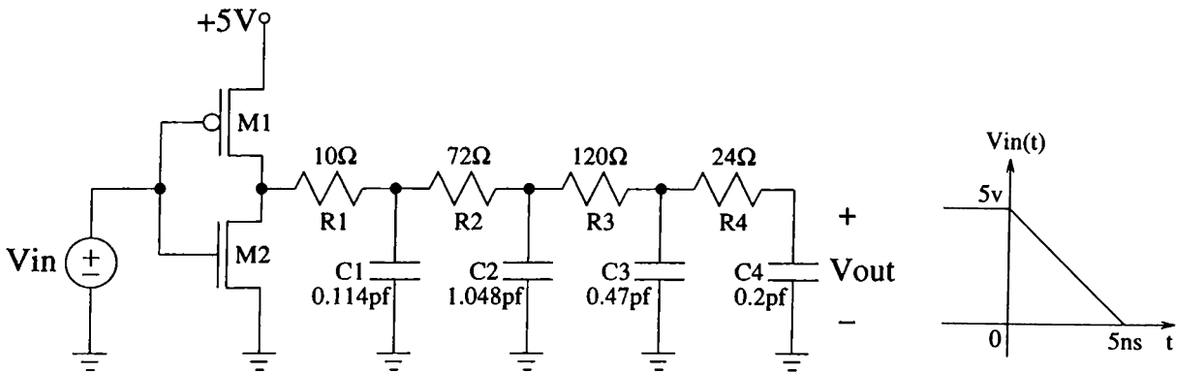


Figure 6.7: RC tree driven by a CMOS inverter and the input voltage function.

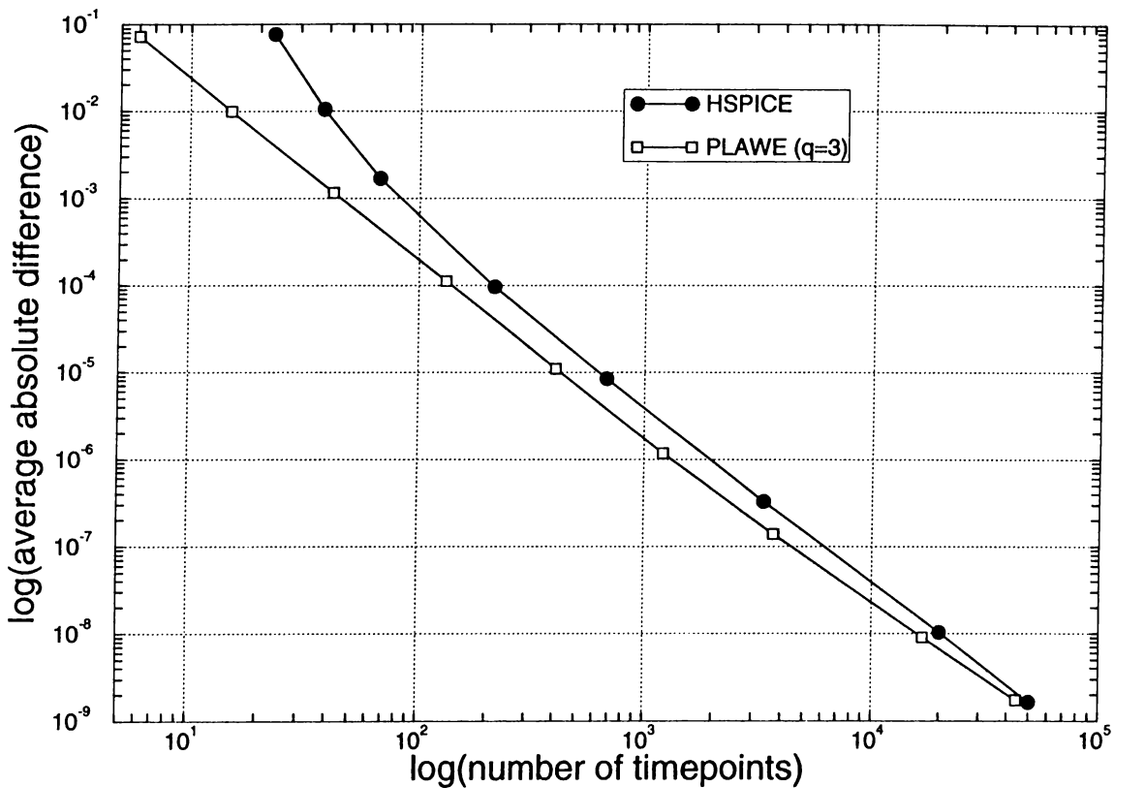


Figure 6.8: Accuracy vs speed graphs for PLAWE and HSPICE in the second example.

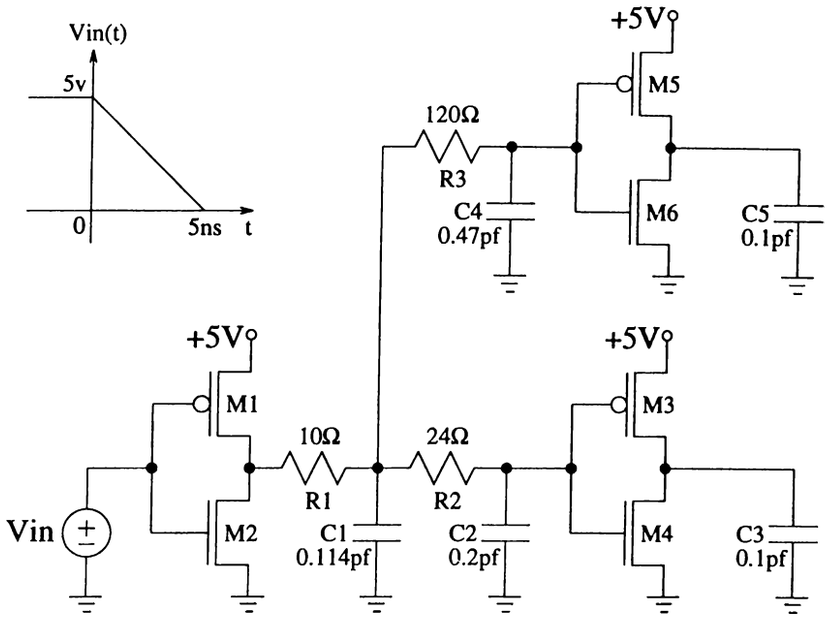


Figure 6.9: Two CMOS inverters driven by the same inverter.

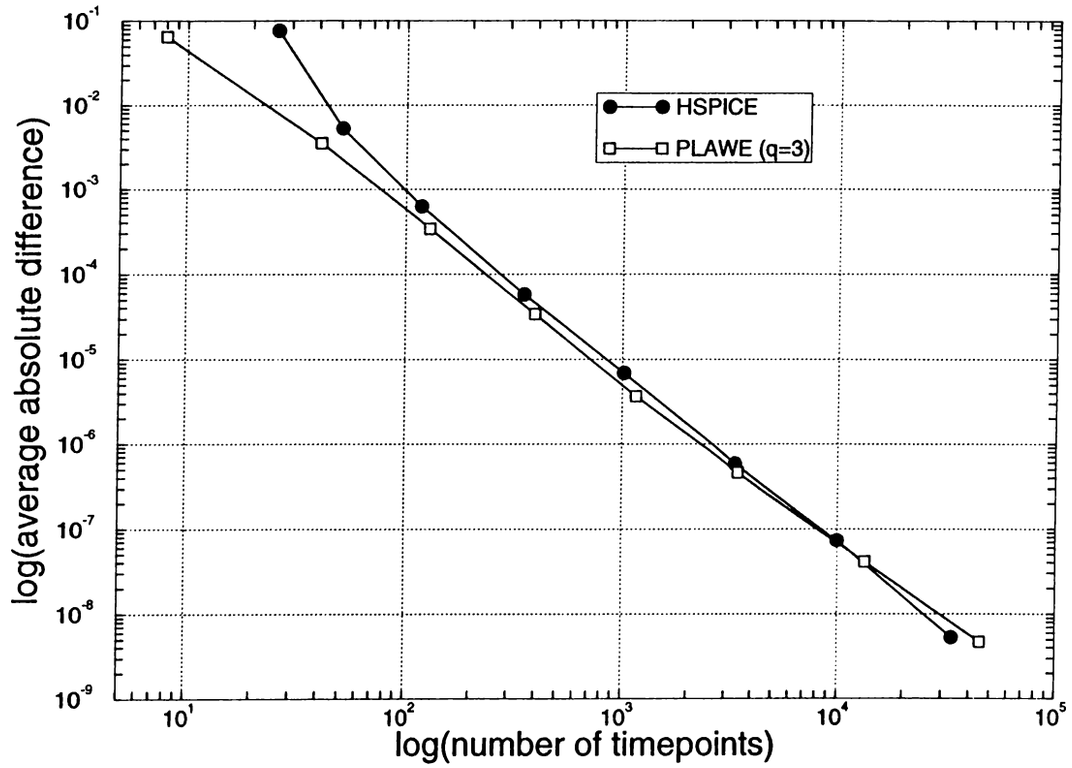


Figure 6.10: Accuracy vs speed graphs for PLAWE and HSPICE in the third example.

Chapter 7

CONCLUSION AND FUTURE WORK

An efficient algorithm for finding DC solutions of *large* PWL resistive circuits is developed. The algorithm is an extension of the piecewise-linear version of the well-known Newton-Raphson method. The main feature of our approach is to insert some randomness into the PWL Newton-Raphson method to guarantee convergence without sacrificing from the speed. The degree of randomness in the algorithm is controlled by the parameters p and q. Using the large number of simulations on the practical example circuits, we have obtained appropriate values for these parameters to yield the best performance of the algorithm. The efficiency of the algorithm has also been examined with respect to the number of PWL segments used to model nonlinear elements. In the case of multiple DC solutions, the algorithm reaches to one of the solutions in each trial. A software implementation of the algorithm has been done and it is used in the DC analysis part of the circuit simulation program PLAWE. This algorithm can also be adapted to the continuous case by modifying the Newton-Raphson algorithm.

A new method is proposed to apply the AWE technique to the time domain analysis of nonlinear circuits. The existing approaches which addressed to solve this problem have utilized the static PWL modeling for nonlinear elements. However, these methods

have two major drawbacks: 1) Finding good PWL models for nonlinear elements is a difficult problem, 2) Static PWL approximation produces less accurate simulation results. Our method overcomes these disadvantages since it uses the SPICE models for nonlinear elements and it can produce very accurate results. By means of a few error tolerance parameters, accuracy level of the simulation can be adjusted by the user. Another advantage of our method is that its implementation is very easy. We have presented some examples to show the efficiency and the accuracy performance of the method. The method is capable of providing an accuracy of 10^{-9} which can not be obtained by static PWL modeling approach. It is observed from the examples that, the method is more efficient in the analysis of weakly nonlinear circuits. As the nonlinearity of a circuit is increased by inserting additional nonlinear elements, the efficiency of the method begins to decrease. Unfortunately, we can say that for large circuits having many number of nonlinear elements, it may take more CPU time than taken by HSPICE to preserve the same accuracy level. This means that it is better to agree to have less accurate results such as comparable to 10^{-2} for the simulation of large nonlinear circuits. Because these results are still more accurate than the results supplied by the static PWL approach.

Briefly, a new circuit simulation tool, PLAWE, has been developed using the AWE technique and the PWL approach. PLAWE has been implemented in C Programming language by writing more than 13 000 lines of code and it runs on SUN Workstations under UNIX operating system. Computational efficiency of the simulator is not optimized yet, but its speed performance is sufficiently good. Some of the results we obtained are:

- The *Popcorn* algorithm guarantees convergence for PWL DC analysis as well as it is computationally efficient as shown by the example circuits.
- Accuracy speed trade-off is achieved by user defined PWL models for nonlinear elements.
- Both static and dynamic PWL modeling techniques has been examined and it is shown that the dynamic PWL modeling can produce very accurate results while it is computationally efficient only for weakly nonlinear circuits.

- Simple static PWL models with a few segments for nonlinear devices give quite good results in the transient analysis.
- Instability problem in AWE can be sometimes overcome by using a combination of the derivative and the integral moments but this does not guarantee the stability of AWE approximations in all cases.

Suggested future work can be outlined as follows:

- Modified Nodal Analysis (MNA) method should be used instead of Sparse Tableau Analysis (STA) method to describe the linear circuit equations. Because, MNA computes less unknown circuit variables and hence its computational complexity is less than that of STA.
- It is believed that transient simulation time can be reduced significantly upon the partitioning of a large circuit into several smaller subcircuits. So an algorithm exploiting circuit partitioning can be developed and implemented in PLAWE.
- The internal parasitic capacitances of the nonlinear devices can be modeled by piecewise linear capacitances in each region of operation instead of constant grounded capacitances. This may increase the accuracy of simulation.
- The capacitor-voltage source loops and inductor-current source cutsets must be handled appropriately to ensure reliability of the simulator.

APPENDIX

SPICE MOS Level-1 model:

```
.MODEL ntype nmos (LEVEL=1 VTO=1.0 KP=57E-6
+GAMMA=0.3 PHI=0.7 LAMBDA=0.05 CGBO=0 CGSO=0
+CGDO=0 IS=0 CAPOP=5 ACM=0)
*
.MODEL ptype pmos (LEVEL=1 VTO=-1.0 KP=17E-6
+GAMMA=0.5 PHI=0.69 LAMBDA=0.04 CGBO=0 CGSO=0
+CGDO=0 IS=0 CAPOP=5 ACM=0)
```

SPICE BJT model:

```
.model npn npn (is=1e-14 bf=100 br=1.0 rb=0 rc=0 re=0 cje=0 vje=0.8
+ mje=0.5 cjc=0 vjc=0.7 mjc=0.5 cjs=0 vjs=0.6 mjs=0.5 rbm=0 irb=0)
```

A complete SPICE input deck for the opamp circuit mentioned in Chapter 6 is given below. The SPICE Level 1 model parameters for ntype and ptype MOS transistors are given above.

```
*** opamp circuit with unity gain feedback
m1      3      0    1    1  ptype w=12.0u l=24.0u
m2      4      4    3    3  ptype w=12.0u l=12.0u
m3      4      4    2    2  ntype w=12.0u l=12.0u
m4      5      5    1    1  ptype w=24.0u l=2.4u
m5      6      6    5    5  ptype w=24.0u l=2.4u
m6      6    out    7    7  ntype w=60.0u l=2.4u
m7      7      4    2    2  ntype w=12.0u l=12.0u
```

```
m8     8     5   1   1  ptype w=24.0u l=2.4u
m9     9     6   8   8  ptype w=24.0u l=2.4u
m10    9     9  10  10  ntype w=24.0u l=2.4u
m11   10    10   2   2  ntype w=24.0u l=2.4u
m12   11    13   1   1  ptype w=48.0u l=2.4u
m13  out    14  11  11  ptype w=48.0u l=2.4u
m14  out     9  12  12  ntype w=48.0u l=2.4u
m15   12    10   2   2  ntype w=48.0u l=2.4u
m16   13    13   1   1  ptype w=24.0u l=2.4u
m17   14    14  13  13  ptype w=24.0u l=2.4u
m18   14  ninv   7   7  ntype w=60.0u l=2.4u
c1     3  0  20.76ff
c2     4  0  46.58ff
c3     5  0  6.52ff
c4     6  0  50.87ff
c5     7  0  59.93ff
c6     8  0  3.29ff
c7     9  0  42.90ff
c8    10  0  11.88ff
c9    11  0  3.29ff
c10   12  0  2.69ff
c11   13  0  6.52ff
c12   14  0  51.23ff
cload out 0  5pf

vdd 1 0 dc  5v
vss 2 0 dc -5v
vin ninv 0 pulse(0v 0.1v 0ns 1ns 1ns 100ns 200ns)

.tran 200ps 200ns
.print tran v(out)
.end
```

Bibliography

[1] L. T. Pillage and R. A. Rohrer, "Asymptotic Waveform Evaluation for Timing Analysis," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 352–366, Apr. 1990.

[2] L.W. Nagel,. "SPICE2: A Computer Program to Simulate Semiconductor Circuits,". Tech. Rep. #ERL-M520, University of California, Berkeley, May 1975.

[3] W.T. Weeks, A.J. Jimenez, G.W. Mahoney, D. Mehta, H. Qassemzadeh, and T.R. Scott, "Algorithms for ASTAP-A Network Analysis Program," *IEEE Trans. Circuit Theory*, vol. CT-20, pp. 628–634, Nov. 1973.

[4] P. Yang, I.N. Hajj, and T.N. Trick, "SLATE: A Circuit Simulation Program with Latency Exploitation and Node Tearing," in *Proc. of the IEEE Int. Conf. on Circuits and Computers*, pp. 353–355, Oct. 1980.

[5] N.B.G. Rabbat, A.L. Sangiovanni-Vincentelli, and H.Y. Hsieh, "A Multi-level Newton Algorithm with Macromodelling and Latency for the Analysis of Large-Scale Nonlinear Circuits in the Time Domain," *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 733–741, Sept. 1979.

[6] B.R. Chawla, H.K. Gummel, and P. Kozak, "MOTIS - An MOS Timing Simulator," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 901–910, Dec. 1975.

[7] S.P. Fan, M.Y. Hsueh, A.R. Newton, and D.O. Pederson, "MOTIS-C: A New Circuit Simulator for MOS LSI Circuits," in *Proc. of the IEEE Int. Symp. on Circuits and Systems*, pp. 700–703, Apr. 1977.

[8] C.F. Chen, C.Y. Lo, H.N. Nham, and P. Subramaniam, "The Second Generation MOTIS Mixed-Mode Simulator," in *Proc. of 21st Design Automation Conference*, pp. 10–17, June 1984.

[9] Y.P. Wei, I.N. Hajj, and T.N. Trick, "A Prediction-Relaxation Based Simulator for MOS circuits," in *Proc. of the IEEE Int. Conf. on Circuits and Computers*, Sept. 1982.

[10] E. Lelarasmee, A.E. Ruehli, and A.L. Sangiovanni-Vincentelli, "The Waveform Relaxation Method for the Time Domain Analysis of Large Scale Integrated Circuits," *IEEE Trans. Computer-Aided Design*, vol. CAD-1, pp. 131–145, July 1982.

[11] J. White and A.L. Sangiovanni-Vincentelli, "RELAX2 : A Modified Waveform Relaxation Approach to the Simulation of MOS Digital Circuits," in *Proc. of the IEEE Int. Symp. on Circuits and Systems*, pp. 756–759, May 1983.

[12] J. White and A.L. Sangiovanni-Vincentelli, "RELAX2.1 : A Waveform Relaxation Based Circuit Simulation Program," in *Proc. of the IEEE Custom Integrated Circuits Conf.*, pp. 232–236, May 1984.

[13] A.R. Newton, "The Simulation of Large-Scale Integrated Circuits," *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 741–749, Sept. 1979.

[14] G. Arnout and H. De Man, "The use of Threshold Functions and Boolean-Controlled Network Elements for Macromodelling of LSI Circuits," *IEEE J. Solid State Circuits*, vol. SC-13, pp. 326–332, June 1978.

[15] K. Sakallah and S.W. Director, "An Activity Directed Circuit Simulation Algorithm," in *Proc. of the IEEE Int. Conf. on Circuits and Computers*, pp. 356–360, Oct. 1980.

[16] V.B. Rao, D.V. Overhauser, T.N. Trick, and I.N. Hajj. *Switch-Level Timing Simulation of MOS VLSI Circuits.* Kluwer Academic Publishers, Boston, 1989.

[17] L.M. Vidigal, S.R. Nassif, and S.W. Director, "CINNAMON: Coupled Integration and Nodal Analysis of MOS Networks," in *Proc. of 1986 Design Automation Conf.*, pp. 179–185, June 1986.

[18] A.J. de Geus, "SPECS : Simulation Program for Electronic Circuits and Systems," in *Proc. of the IEEE Int. Symp. on Circuits and Systems*, pp. 534–537, May 1984.

[19] S.A. Szygenda, "TEGAS2 - Anatomy of a general Purpose Test Generation and Simulation System for Digital Logic," in *Proc. of 9th ACM Design Automation Workshop*, June 1972.

[20] J. Jephson, R. McQuarrie, and R. Vogelsberg, "A Three Value Computer Design Verification System," *IBM Systems J.*, vol. 8, pp. 178–188, 1969.

[21] S.A. Szygenda and E.W. Thompson, "Modelling and Digital Simulation for Design Verification and Diagnosis," *IEEE Trans. Computers*, vol. C-25, pp. 1242–1253, Dec. 1976.

[22] P. Wilcox, "Digital Logic Simulation at the Gate and Functional Level," in *Proc. of IEEE Design Automation Conf.*, pp. 242–248, 1979.

[23] R.E. Bryant, "An Algorithm for MOS Logic Simulation," *LAMBDA magazine*, vol. 1, pp. 46–53, 1980.

[24] R.E. Bryant, "A Switch-level Model and Simulator for MOS Digital Circuits," *IEEE Trans. Computers*, vol. C-33, pp. 160–177, Feb. 1984.

[25] R.H. Bryd, G.D. Hachtel, M.R. Lightner, and M.H. Heydemann, "Switch Level Simulation : Models, Theory and Algorithms," in A.L. Sangiovanni-Vincentelli, ed., *Advances in Computer-Aided Engineering Design.* Jai Press, 1985.

[26] V. Ramachandran, "An Improved Switch-level Simulator for MOS Circuits," in *Proc. of 20th Design Automation Conf.*, pp. 293–299, June 1983.

[27] D.A. Divekar. *FET Modeling for Circuit Simulation.* Kluwer Academic Publishers, Boston, 1988.

[28] J. Vlach and K. Singhal. *Computer Methods for Circuit Analysis and Design.* Van Nostrand Reinhold Company, New York, 1983.

[29] R. Kao. "Piecewise Linear Models for Switch-Level Simulation,". Tech. Rep. #CSL-TR-92-532, Stanford University, Stanford, June 1992.

[30] W. M. G. van Bokhoven "Piecewise linear analysis and simulation," in A. E. Ruehli, ed., *Circuit Analysis, Simulation and Design, Part 2*, chapter 10. Elsevier Science Publishers B.V., North Holland, 1987.

[31] D.A. Hodges and H.G. Jackson. *Analysis and Design of Digital Integrated Circuits.* McGraw-Hill, Singapore, 1988.

[32] W.J. McCalla. *Fundamentals of Computer-Aided Circuit Simulation.* Kluwer Academic Publishers, Boston, 1988.

[33] L.O. Chua and P.M. Lin. *Computer Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques.* NJ: Prentice Hall, Englewood Cliffs, 1975.

[34] F.H. Branin, JR. and H.H. Wang, "A Fast Reliable Iteration Method for dc Analysis of Nonlinear Networks," *Proc. IEEE*, vol. 55, pp. 1819–1826, Nov. 1967.

[35] H.R. Yeager and R.W. Dutton, "Improvement in Norm-Reducing Newton Methods for Circuit Simulation," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 538–546, May 1989.

[36] L.O. Chua and R.L.P. Ying, "Finding All Solutions of Piecewise-Linear Circuits," *Int. J. Circuit Theory Appl.*, vol. 10, pp. 201–229, July 1982.

[37] Q. Huang and R.W. Liu, "A Simple Algorithm for Finding All Solutions of Piecewise-Linear Networks," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 600–609, Apr. 1989.

[38] T. Nishi, "An Efficient Method to Find All Solutions of Piecewise-Linear Resistive Circuits," in *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 2052–2055, May 1989.

[39] K. Yamamura and M. Ochiai, "An Efficient Algorithm for Finding All Solutions of Piecewise-Linear Resistive Circuits," *IEEE Trans. Circuits Syst.-I*, vol. 39, pp. 213–221, Mar. 1992.

[40] J. Katzenelson, "An Algorithm for Solving Nonlinear Resistor Networks," *Bell Syst. Tech. J.*, vol. 44, pp. 1605–1620, Oct. 1965.

[41] L.O. Chua, "Efficient Computer Algorithms for Piecewise-Linear Analysis of Resistive Nonlinear Networks," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 73–85, Jan. 1971.

[42] T. Ohtsuki and N. Yoshida, "DC Analysis of Nonlinear Networks Based on Generalized Piecewise-Linear Characterization," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 146–152, Jan. 1971.

[43] E.S. Kuh and I.N. Hajj, "Nonlinear Circuit Theory: Resistive Networks," *Proc. IEEE*, vol. 59, pp. 340–355, Mar. 1971.

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

[45] T. Fujisawa, E.S. Kuh, and T. Ohtsuki, "A Sparse Matrix Method for Analysis of Piecewise-Linear Resistive Networks," *IEEE Trans. Circuit Theory*, vol. CT-19, pp. 571–584, Nov. 1972.

[46] T. Ohtsuki, T. Fujisawa, and S. Kumagai, "Existence Theorems and a Solution Algorithm for Piecewise-Linear Resistive Networks," *SIAM J. Math. Anal.*, vol. 8, pp. 69–99, Feb. 1977.

[47] M.J. Chien and E.S. Kuh, "Solving Nonlinear Resistive Networks Using Piecewise-Linear Analysis and Simplicial Subdivision," *IEEE Trans. Circuits Syst.*, vol. CAS-24, pp. 305–317, June 1977.

[48] S.M. Lee and K.S. Chao, "Multiple Solutions of Piecewise-Linear Resistive Networks," *IEEE Trans. Circuits Syst.*, vol. CAS-30, pp. 84–89, Feb. 1983.

[49] S. Topçu, O. Ocalı, A. Atalar, and M.A. Tan, "A Novel Algorithm for DC Analysis of Piecewise Linear Circuits: POPCORN," *IEEE Trans. Circuits Syst.-I*, Accepted for publication.

[50] P.J.M. van Laarhoven and E.H.L. Aarts. *Simulated Annealing: Theory and Applications.* D. Reidel, Lancaster, 1987.

[51] P.R. Gray and R.G. Meyer, eds. *Analysis and Design of Analog Integrated Circuits*, chapter 6. John Wiley, 1983.

[52] P.K. Chan, "Comments on Asymptotic Waveform Evaluation for Timing Analysis," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 1078–1079, Aug. 1991.

[53] V. Raghavan, R.A. Rohrer, L.T. Pillage, J.Y. Lee, J.E. Bracken, and M.M. Alaybeyi, "AWE-Inspired," in *Proc. of the IEEE Custom Integrated Circuits Conf.*, pp. 18.1.1–18.1.8, May 1993.

[54] X. Huang, V. Raghavan, and R.A. Rohrer, "AWEsim: A Program for the Efficient Analysis of Linear(ized) circuits," in *Technical Digest of the IEEE Int. Conf. on Computer-Aided Design*, pp. 534–537, Nov. 1990.

[55] T.K. Tang, M. Nakhla, and R. Griffith, "Analysis of Lossy Multiconductor Transmission Lines Using the Asymptotic Waveform Evaluation Technique," *IEEE Trans. on Microwave Theory and Techniques*, vol. MTT-39, Dec. 1991.

[56] T.K. Tang and M. Nakhla, "Analysis of High Speed VLSI Interconnects Using the Asymptotic Waveform Evaluation Technique," *IEEE Trans. Computer-Aided Design*, vol. CAD-11, pp. 341–352, Mar. 1992.

[57] J.E. Bracken, V. Raghavan, and R. A. Rohrer, "Simulating Distributed Elements with Asymptotic Waveform Evaluation (AWE)," in *Proc. of the IEEE Int. Microwave Symposium*, June 1992.

[58] L.T. Pillage, C.M. Wolf, and R.A. Rohrer, "Dominant Pole(s)/Zero(s) Analysis for Analog Circuit Design," in *Proc. of the IEEE Custom Integrated Circuits Conf.*, pp. 21.3.1–21.3.4, May 1989.

[59] J.Y. Lee, X. Huang, and R. A. Rohrer, "Pole and Zero Sensitivity Calculation in Asymptotic Waveform Evaluation," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 586–597, May 1992.

[60] J.Y. Lee and R. A. Rohrer, "AWEsymbolic: Compiled Circuit Analysis Using Asymptotic Waveform Evaluation," in *Proc. of the 29th ACM/IEEE Design Automation Conf.*, pp. 213–218, 1992.

[61] C. T. Dikmen, M. M. Alaybeyi, S. Topçu, A. Atalar, E. Sezer, M. A. Tan, and R. A. Rohrer, "Piecewise Linear Asymptotic Waveform Evaluation for Transient Simulation of Electronic Circuits," in *Proc. of the IEEE Int. Symp. on Circuits and Systems*, pp. 854–857, June 1991.

[62] R. Kao and M. Horowitz, "Efficient Moment Based Timing Analysis for Variable Accuracy Switch Level Simulation," in *Proc. of the 28th ACM/IEEE Design Automation Conf.*, 1991.

[63] M.M. Alaybeyi, J.E. Bracken, J.Y. Lee, V. Raghavan, R.J. Trihy, and R. A. Rohrer, "Analysis of MCM's Using Asymptotic Waveform Evaluation (AWE)," in *Proc. of the IEEE Multi-Chip Module Conf.*, Mar. 1992.

[64] V. Raghavan, J.E. Bracken, and R. A. Rohrer, "AWEspice: A General Tool for the Accurate and Efficient Simulation of Interconnect Problems," in *Proc. of the 29th ACM/IEEE Design Automation Conf.*, pp. 87–92, 1992.

[65] M.M. Alaybeyi, J.E. Bracken, J.Y. Lee, V. Raghavan, R.J. Trihy, and R. A. Rohrer, "Exploiting Partitioning in Asymptotic Waveform Evaluation (AWE)," in *Proc. of the IEEE Custom Integrated Circuits Conf.*, 1992.

[66] M.M. Alaybeyi, J.Y. Lee, and R. A. Rohrer, "Numerical Integration and Asymptotic Waveform Evaluation," in *Technical Digest of the IEEE Int. Conf. on Computer-Aided Design*, 1992.

[67] J.E. Bracken, V. Raghavan, and R. A. Rohrer, "Combining Asymptotic Waveform Evaluation and the Method of Characteristics," in *Technical Digest of the IEEE Int. Conf. on Computer-Aided Design*, 1992.

[68] D.F. Anastasakis, N. Gopal, S.Y. Kim, and L.T. Pillage, "On the Stability of Moment-Matching Approximations in Asymptotic Waveform Evaluation," in *Proc. of the 29th ACM/IEEE Design Automation Conf.*, pp. 207–212, 1992.

[69] C.F. Chen and L.S. Shieh, "A Novel Approach to Linear Model Simplification," *Int. J. Control*, vol. 8, pp. 561–570, 1968.

[70] V. Zakian, "Simplification of Linear Time-Invariant Systems by Moment Approximants," *Int. J. Control*, vol. 18, pp. 455–460, 1973.

[71] M.J. Bosley, H.W. Kropholler, and F.P. Lees, "On the Relation Between the Continued Fraction Expansion and Moments Matching Methods of Model Reduction," *Int. J. Control*, vol. 18, pp. 461–474, 1973.

[72] G.A. Baker, Jr. *Essentials of Padé Approximants.* Academic Press, 1975.

[73] X. Huang. *Padé Approximation of Linear(ized) Circuit Responses.* PhD thesis, Carnegie Mellon University, Nov. 1990.

[74] M.M. Alaybeyi. "BUSTLE: A New Circuit Simulation Tool,". Master's thesis, Bilkent University, Ankara, Turkey, July 1990.

[75] Y. Shamash, "Stable Reduced-Order Models Using Padé-Type Approximations," *IEEE Trans. Automatic Control*, pp. 615–616, Oct. 1974.

[76] Y. Shamash, "Model Reduction Using the Routh Stability Criterion and the Padé Approximation Techniques," *Int. J. Control*, vol. 21(3), pp. 475–584, 1975.

[77] J. Pal, "Stable Reduced-Order Padé Approximants Using the Routh-Hurwitz Array," *Electronic Letters*, vol. 15, pp. 225–226, 1979.

[78] F.J. Alexandro, Jr., "Stable Partial Padé Aproximations for Reduced-Order Transfer Functions," *IEEE Trans. Automatic Control*, vol. AC-29, pp. 159–161, Feb. 1984.

[79] H. Xiheng, "FF-Padé Method of Model Reduction in Frequency Domain," *IEEE Trans. Automatic Control*, vol. AC-32, pp. 243–246, Mar. 1987.

[80] C.T. Dikmen. "BUSTLE: A New Circuit Simulation Tool Using Asymptotic Waveform Evaluation and Piecewise Linear Approach,". Master's thesis, Bilkent University, Ankara, Turkey, July 1990.

[81] Meta-Software, Inc., Campbell, U.S.A. *HSPICE User's Manual*, 1992.

[82] S. Topçu. "Design and Testing of a Microprocessor Compatible 128-Bit Correlator Chip,". Master's thesis, Bilkent University, Ankara, Turkey, July 1989.

[83] T.L. Quarles. "SPICE3 Version 3C1 Users Guide,". Memorandum No. UCB/ERL M89/46, University of California, Berkeley, Apr. 1989.

Vita

Satılmış Topçu was born in Ankara, Turkey, in 1964. He received his B.Sc. degree from Middle East Technical University, Ankara, Turkey, and M.Sc. degree from Bilkent University, Ankara, Turkey, in 1986 and 1989, respectively, both in electrical and electronics engineering. His current research interests include design, testing and simulation of VLSI circuits.