Electronic Circuit and System Simulation Methods

Lawrence T. Pillage
Ronald A. Rohrer
Chandramouli Visweswariah

McGraw-Hill, Inc.
New York San Francisco Washington, D.C. Auckland Bogotá
Caracas Lisbon London Madrid Mexico City Milan
Montreal New Delhi San Juan Singapore
Sydney Tokyo Toronto

Library of Congress Cataloging-in-Publication Data

Pillage, Lawrence.
Electronic circuit and system simulation methods / Lawrence Pillage, Ronald A. Rohrer, Chandramouli Visweswariah.
p. cm.
Includes index.
ISBN 0-07-050169-6
1. Linear integrated circuits—Computer simulation. 2. SPICE (Computer file) I. Rohrer, Ronald A. II. Visweswariah, Chandramouli. III. Title.
TK7874.P52 1994
621.3815'01'1353—dc20 94-24429
CIP

1 2 3 4 5 6 7 8 9 0 DOC/DOC 9 0 9 8 7 6 5 4

ISBN 0-07-050169-6

The sponsoring editor for this book was Stephen S. Chapman.

Printed and bound by R. R. Donnelley & Sons Company.

Other McGraw-Hill Books of Interest

Handbooks

CHEN • *Computer Engineering Handbook*

COOMBS • *Printed Circuits Handbook*

COOMBS • *Electronic Instruments Handbook*, Second Edition

DI GIACOMO • *Digital Bus Handbook*

FINK AND CHRISTIANSEN • *Electronics Engineers' Handbook*

HARPER • *Electronic Packaging and Interconnection Handbook*

JURAN AND GRYNA • *Juran's Quality Control Handbook*

RORABAUGH • *Digital Filter Designer's Handbook*

TUMA • *Engineering Mathematics Handbook*

WAYNANT • *Electro-Optics Handbook*

WILLIAMS AND TAYLOR • *Electronic Filter Design Handbook*

Other

ANTOGNETTI • *Power Integrated Circuits*

BEST • *Phase-Locked Loops*, Second Edition

BUCHANAN • *CMOS/TTL Digital Systems Design*

BUCHANAN • *BiCMOS/CMOS Systems Design*

BYERS • *Printed Circuit Board Design with Microcomputers*

DAUGHERTY • *Analog to Digital Conversion*

ELLIOTT • *Integrated Circuits Fabrication Technology*

HECHT • *The Laser Guidebook*

KIELKOWSKI • *Inside SPICE*

LICARI • *Multichip Module Design*

MASSOBRIO AND ANTOGNETTI • *Semiconductor Device Modeling with SPICE*, Second Edition

SZE • *VLSI Technology*

TABAK • *Advanced Microprocessors*, Second Edition

TSUI • *LSI/VLSI Testability Design*

WATERS • *Active Filter Design*

WOBSCHALL • *Circuit Design for Electronic Instrumentation*

WALKER • *Optical Engineering Fundamentals*

To order or receive additional information on these or any other McGraw-Hill titles, in the United States please call 1-800-822-8158. In other countries, contact your local McGraw-Hill representative.

BC14BCZ

Table of Contents

Preface

The field of circuit simulation has seen some exciting developments ever since the advent of integrated circuits. Modern integrated circuits continually challenge circuit simulation algorithms and implementations with the verification problems they pose. What makes circuit simulation unique is its multi-disciplinary nature. It is an intertwined set of concepts borrowed and adapted from mathematics, circuit theory, graph theory, physics, device modeling, electrical engineering, and software development. Although there is much active research in the subject, this book attempts to clearly explain some of the fundamentals of circuit simulation, on which most modern techniques are based. Some of the more recent advances are covered in the book, too.

This book evolved from our teaching and research activities over the years. We are indebted to all those who invented the concepts and techniques described in the book and to those who wrote earlier books on the subject.

This book would not have been possible without the collaboration, cooperation, and help of many colleagues, students, and friends. While it is not possible to mention all of them, we would particularly like to thank our spouses Leah Pillage, Casey Jones, and Patricia Buchanan for their constant support and encouragement; Catherine Rapinett for typing early drafts of the manuscript; all of the graduate students from the simulation courses at Carnegie Mellon University and the University of Texas at Austin for their critique of the notes; and David Ling, Ellen Yoffa, and Bill Joyner at the IBM T. J. Watson Research Center for their support and encouragement. We also thank Steve Chapman and Jim Halston at McGraw Hill for an outstanding job.

L.T.P., R.A.R., and C.V.

Electronic Circuit and System Simulation Methods

Chapter 1 *Introduction to Circuit Simulation*

The simulation techniques introduced in the 1950s to analyze circuits with tens of transistors are substantially the same as those used today to analyze circuits with tens of thousands of transistors. With the availability of powerful computers and the advent of workstations, circuit simulators can run much faster today than they did then. But these programs are often called upon to verify the potential performance of the next generation of workstations and parallel processors, and with each new generation of computers the underlying circuitry becomes ever more complex. Due to this increasing complexity, circuit simulation as a pre-manufacturing design verification strategy is barely keeping up with the demands that are being placed upon it. And in many areas of application it has even fallen behind, being supplemented by less precise switch- and logic-level simulation. But such reduced-precision simulation strategies may only work reliably when design styles are restricted. In the fierce competition for faster and smaller circuits, more often than not, presumed restrictions on design styles are bent or broken. In the final analysis then, only circuit simulation is trusted -- sometimes more so than it should be -- to verify the essential electrical-system behavior. As a result, huge computing resources are assigned to the circuit simulation task, which is often the bottleneck in the design process.

This chapter provides an overview of circuit simulation techniques. In particular, it is an introduction to some of the algorithms and methodologies used in the industry standard circuit simulator SPICE [Nagel71, Nagel75]. The techniques introduced in this chapter will be covered in greater detail in the chapters that follow. In addition, alternatives to SPICE and various nontraditional simulation techniques will be introduced in subsequent chapters.

1.1 Traditional Circuit Simulation

Before introducing circuit simulation techniques it may be helpful to first discuss some of the capabilities of a simulator such as SPICE. Most general purpose circuit simulation programs provide the following capabilities:

- Linear dc analysis to evaluate the dc currents and dc voltages for a lumped, linear, time-invariant circuit.
- Nonlinear dc analysis to obtain the quiescent operating point of a circuit which contains nonlinear elements, such as transistors.
- Linear ac analysis to obtain the frequency response of a lumped, linear, time invariant circuit.
- Small signal ac analysis to obtain the frequency domain response of a circuit by replacing nonlinear elements with their linearized equivalents computed from the quiescent operating point.
- Linear transient analysis to determine the time domain response of a circuit to various input waveforms starting with the initial conditions obtained from the linear dc analysis.
- Large signal transient analysis to obtain the time domain response of a circuit which contains nonlinear elements, such as transistors. The time domain responses are determined by considering the various input waveforms starting with the initial conditions obtained from the nonlinear dc analysis.

In addition to the modes listed above, circuit simulators provide a variety of other functions, such as pole/zero analysis and noise analysis. There are also special purpose simulation programs for thermal analysis, switched capacitor circuit analysis, and so on. By far the most common way in which the above modes are used is a nonlinear dc analysis to establish the quiescent point at the start of the simulation, followed by a large signal transient analysis. The analyses listed above, which form the core for most general purpose circuit simulators, also share a common analysis core in that they all rely on the linear dc analysis algorithm. Therefore, to consider how a circuit simulator provides these capabilities we start with the foundations of linear dc analysis.

1.2 Linear, Time-Invariant Circuits

Consider a simple linear time-invariant circuit comprised of the most basic two-terminal elements: resistors, ideal independent current and voltage sources. The branch relations for these circuit elements are shown in Figure 1.1. The current-voltage relationships for any of these elements can be abstracted in terms of a generic two-terminal element or a topological branch as shown in Figure 1.2. In order to treat all types of branches in a consistent and systematic manner, we use *associated reference directions*.

Associated Reference Directions: The positive (+) reference for the branch voltage (v_b) is at the tail of the branch current (i_b) reference arrow, and the negative (-) reference for the branch voltage is at the head of the branch current reference arrow.

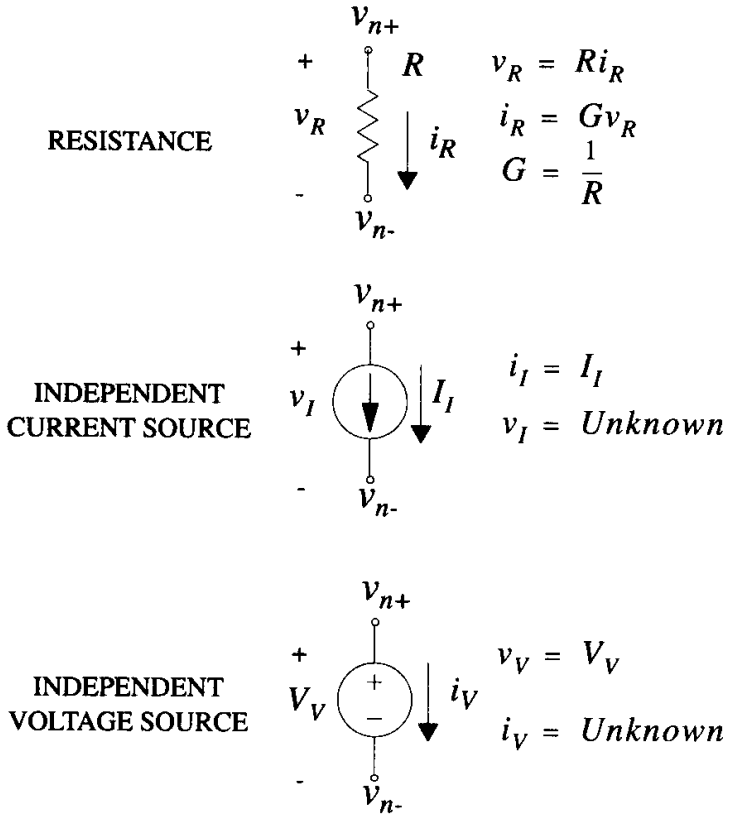

Figure 1.1 Branch relations for some two-terminal circuit elements.

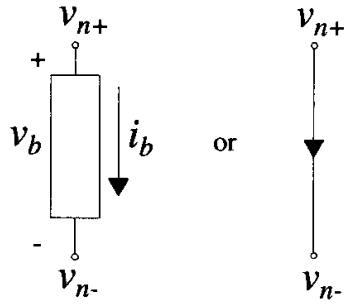

Figure 1.2 Generic two-terminal element and a topological branch.

The (positive referenced) branch voltage drop is in the same direction as the (positive referenced) branch current. Using the associated reference directions, we can compute instantaneous branch power as follows:

- $p_b = v_b i_b > 0$ implies power is being delivered to the element from the rest of the circuit.
- $p_b = v_b i_b < 0$ implies power is being delivered to the rest of the circuit from the element.

Using the topological branch model in Figure 1.2 and the associated reference direction conventions, Kirchhoff's Laws of Interconnection are easily defined:

Kirchhoff's Voltage Law (KVL): Every circuit node has a unique voltage (with respect to the ground or datum node, which is 0 volts by convention). The voltage (drop) across a branch, v_b, is equal to the difference between the (positive and negative referenced) voltages of the nodes on which it is incident.

$$v_b = v_{n+} - v_{n-} \tag{1.2.1}$$

Kirchhoff's Current Law (KCL): The (algebraic) sum of all of the currents flowing out of (or into) any circuit node is zero.

An example of the application of KCL at a node is shown in Figure 1.3. The branch relations along with the KCL expression for node N result in the following equation in terms of the node voltage variables:

$$\frac{v_N - v_M}{R_3} + \frac{v_N - v_K}{R_5} + \frac{v_N - v_R}{R_{10}} + \frac{v_N - v_S}{R_{64}} + I_6 - I_{49} = 0 \tag{1.2.2}$$

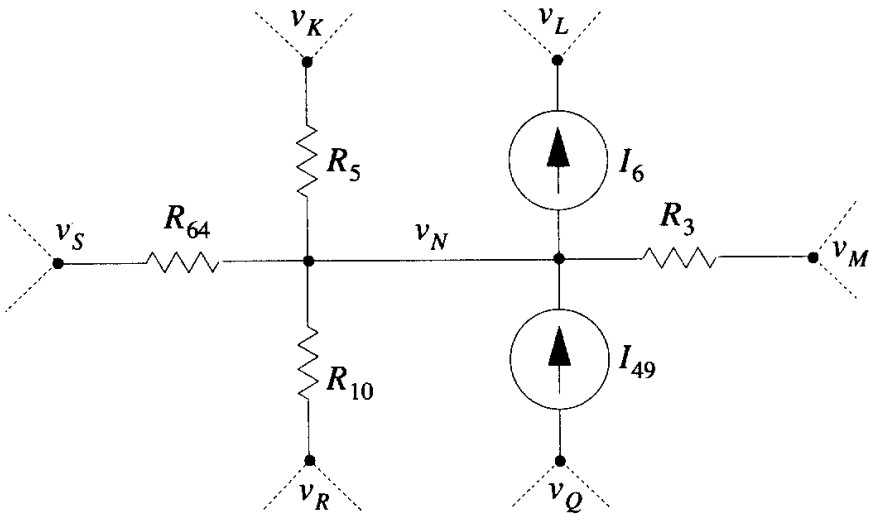

Figure 1.3 KCL at the N^{th} node.

1.3 Nodal Analysis

Consider a b branch, $(n+1)$ node circuit (in the rest of this book, we will consider $(n+1)$ node circuits so that the number of non-datum nodes will always be n). Writing a

KCL equation in terms of node voltages, as described above, for every non-datum node, leads to n nodal equations in terms of n non-datum node voltages. The *datum* node is also called the *ground* or *reference* node, and usually is taken to be at a potential of zero volts. We can write programs to formulate and then solve such sets of equations in general.

For example, consider the analysis of the dc circuit shown in Figure 1.4. This circuit has four nodes not counting the ground node; therefore, $n = 4$. Consider for now that all of the resistances are $1\,\Omega$ and the two current sources have values of 1A. By inspection we can write the KCL expression for the n non-datum nodes resulting in the following set of nodal equations:

$$\begin{aligned}
&\frac{v_1 - v_0}{R_2} + \frac{v_1 - v_2}{R_3} - I_1 = 0 && \rightarrow \quad 2v_1 - v_2 = 1 \\
&\frac{v_2 - v_1}{R_3} + \frac{v_2 - v_0}{R_4} + \frac{v_2 - v_3}{R_5} = 0 && \rightarrow \quad -v_1 + 3v_2 - v_3 = 0 \\
&\frac{v_3 - v_2}{R_5} + \frac{v_3 - v_0}{R_6} + \frac{v_3 - v_4}{R_7} = 0 && \rightarrow \quad -v_2 + 3v_3 - v_4 = 0 \\
&\frac{v_4 - v_3}{R_7} + \frac{v_4 - v_0}{R_8} - I_9 = 0 && \rightarrow \quad -v_3 + 2v_4 = 1
\end{aligned} \tag{1.3.1}$$

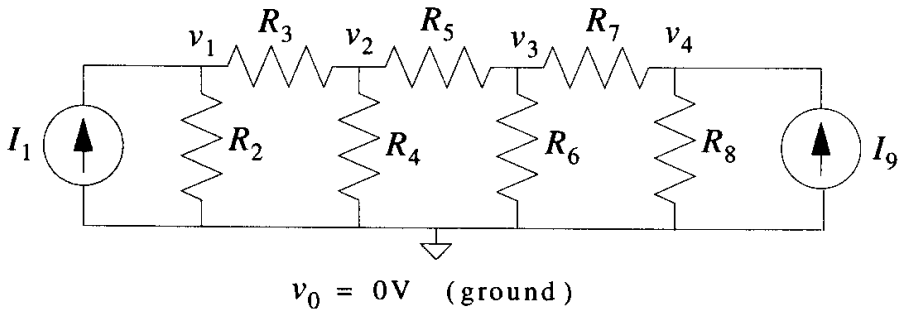

Figure 1.4 Resistor ladder circuit example.

These n nodal equations can be expressed in matrix form

$$\boldsymbol{Yv} = \boldsymbol{J} \tag{1.3.2}$$

where $\boldsymbol{Y}$ is the $n \times n$ nodal admittance matrix, $\boldsymbol{J}$ is the $n \times 1$ vector of current source inputs and $\boldsymbol{v}$ is the $n \times 1$ vector of node voltages which are being sought. Writing the set of nodal equations by applying KCL at each node is not, however, the most efficient means of formulation for a software program.

1.4 Nodal Admittance Equation Stamps

In general, the circuit being analyzed is described for a circuit simulation program in terms of a netlist file. Even computer-aided engineering (CAE) tools which include schematic capture will first convert the graphical representation into a netlist description. Figure 1.5 is a simple netlist format (similar to the SPICE language) which describes the circuit shown in Figure 1.4.

Branch type/ name	From node	To node	Value
I1	0	1	1.0
R2	1	0	1.0
R3	1	2	1.0
R4	2	0	1.0
R5	2	3	1.0
R6	3	0	1.0
R7	3	4	1.0
R8	4	0	1.0
I9	0	4	1.0

Figure 1.5 A netlist representation of the circuit shown in Figure 1.4.

For the nodal admittance (matrix) equations, the elements in the netlist contribute terms in a procedural manner. Branches 2 through 8, the resistors, contribute terms to the $\boldsymbol{Y}$ matrix while branches 1 and 9, the independent current sources, contribute to the $\boldsymbol{J}$ vector.

From the input list these contributions can be characterized on a branch-by-branch basis in terms of *matrix stamps* (sometimes called *element stencils* or just *stencils*). To explain how stamps work, we will assume that the circuit nodes are consecutively numbered. Of course this is not the case in general, however, unique alphanumeric node names in the user-specified netlist are easily mapped into consecutively numbered nodes internal to the circuit simulator.

The resistor matrix-stamp is shown in Figure 1.6(a). For a resistor of value R_k from node i to node j as shown in Figure 1.7(a), a *positive* conductance value $(1/R_k)$ is *added* to matrix locations (i, i) and (j, j) while a *negative* value $(-1/R_k)$ is *added* to matrix locations (i, j) and (j, i) . The stamp for independent current sources is shown in Figure 1.6(b). For a current source branch of value I_k directed from node i to node j, as shown in Figure 1.7(b) a *negative* current source value is *added* to the i^{th} entry of the $\boldsymbol{J}$ vector and a *positive* value is *added* to the j^{th} entry. We postpone the treatment of the independent voltage source stamp until Chapter 2 where we discuss Modified Nodal Analysis.

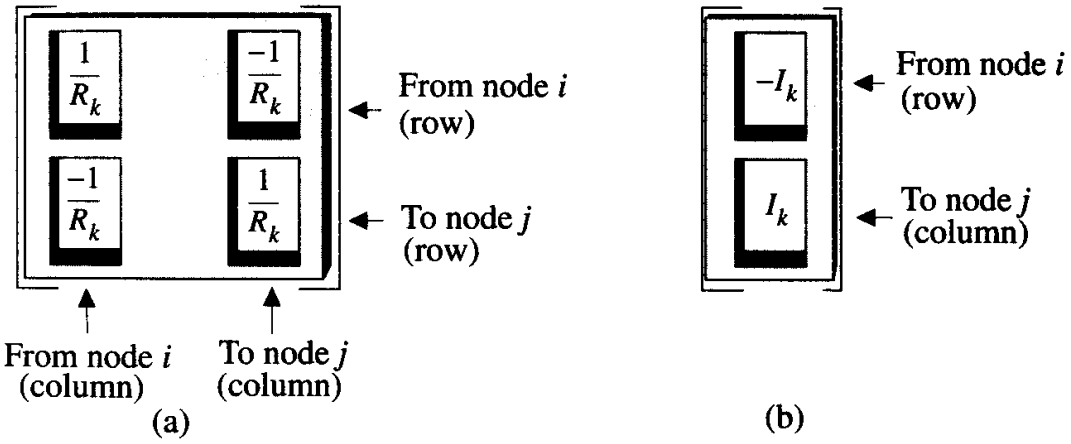

Figure 1.6 Element stamps: (a) For a resistor of value R_k connected *from* node i *to* node j; (b) For an independent current source of value I_k connected *from* node i *to* node j.

Figure 1.7 (a) A resistor connected *from* node i *to* node j, and (b) a current source connected *from* node i *to* node j.

We can obtain the overall set of nodal admittance equations by *stamping in* the branch contributions on an element-by-element basis. To understand the stamp concept, note that the current entering node i and leaving node j due to I_k adds two terms to the $\boldsymbol{J}$ vector. For example, the first line of the netlist in Figure 1.5 is stamped into the nodal matrix equations as follows:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} -I_1 \\ I_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{1.4.1}$$

Note that this section builds up the equations the way a computer program would. The intermediate forms of the equations (1.4.1), (1.4.4), and (1.4.5) are not mathematically valid equations, but the final set of equations (1.4.6) and (1.4.7) are complete. For resistors (see Figure 1.7(a)), the current is expressed in terms of the node voltages on the left hand side. For example, the current *leaving* node i due to resistor R_k is

$$\frac{1}{R_k}(v_i - v_j) \tag{1.4.2}$$

and the current *leaving* node j through resistor R_k is

$$\frac{1}{R_k}(v_j - v_i) \tag{1.4.3}$$

Therefore, stamping in the second element (R_2) from the netlist in Figure 1.5 results in

$$\begin{bmatrix} \frac{1}{R_2} & -\frac{1}{R_2} & 0 & 0 & 0 \\ -\frac{1}{R_2} & \frac{1}{R_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} -I_1 \\ I_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{1.4.4}$$