SPICE

A Guide to Circuit Simulation and Analysis Using PSpice®

PAUL W. TUINENGA

073

73.7507
T-25

SPICE
A Guide to Circuit Simulation and Analysis Using PSpice®

PAUL W. TUINENGA
Microsim Corporation

中国科学院电子学研究所
1991 .7.
图书館藏

PRENTICE HALL, *Englewood Cliffs, New Jersey 07632*

9150064

9150064

Library of Congress Cataloging-in-Publication Data

Tuinenga, Paul W.
SPICE: a guide to circuit simulation and analysis using PSpice/
Paul W. Tuinenga.
p. cm.
Bibliography: p.
ISBN 0-13-834607-0
1. SPICE (Computer program) I. Title.
TK454.T85 1988 88-2393
621.319'2—dc 19 CIP

Editorial/production supervision and
interior design: *Carolyn Fellows* and *Joseph Scordato*
Cover design: ***Lundgren Graphics***
Manufacturing buyer: *Mary Noonan* and *Sallye Scott*

IBM®PC is a registered trademark of International Business Machines Corporation. PSpice® is a registered trademark of MicroSim Corporation.

© 1988 by Prentice-Hall, Inc.
A Division of Simon & Schuster
Englewood Cliffs, New Jersey 07632

ED76 / 1

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

ISBN 0-13-834607-0

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall Hispanoamericana, S.A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Simon & Schuster Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

Preface

SPICE, from the University of California, at Berkeley, is the *de facto* world standard for analog circuit simulation. PSpice®, from MicroSim Corporation, is one of the many commercial derivatives of SPICE, and the first to be available on the IBM® personal computer. PSpice quickly became popular, and our customers found that there are no "how to" references for using SPICE, or its derivatives, the way there are references for database, spreadsheet, and word processing programs. This book's goal is to help users of PSpice, and many other SPICE-derived simulators, to access and use the features of the simulator for their work. Many of these features are only hinted at in other references, notes, or advice from other users, which are the traditional means of help you get for SPICE.

Beyond the syntax and semantics of the SPICE-standard input file, this book also demonstrates the use of PSpice for electrical engineering applications. There is a lot you can do with a circuit simulator that corresponds to what you might do on the lab-bench, as well as simulated measurements that go beyond what is possible with lab equipment.

The background required for using PSpice includes the following:

> This book assumes that you have a passing acquaintance with electrical or electronic circuits. This may come from formal study of basic electronic components (for instance, do you know what resistors and capacitors are, and how they react to electrical stimulation) and network analysis (for instance, do you recall Kirchhoff's laws). Or perhaps you picked up a working knowledge from your job, or as a hobby. If you have a circuit whose operation you basically understand, and you want to simulate this circuit to check the details of operation, you probably know enough to understand PSpice.

This book also assumes that you are able to operate the computer that will run the simulator, as well as create the input for the simulator. Perhaps a friend, or another kind person, will help you with this.

Many people made this book possible, Foremost, I thank my family for their patience while my preoccupation with this book kept me away from them. Also, I thank the people at MicroSim for their encouragement. Finally, I thank the users of PSpice whose many questions made the need for this book so obvious.

Paul W. Tuinenga

ADDITIONAL ITEMS AVAILABLE

PSpice is available for the IBM-compatible personal computers, including the newer PS/2 series. PSpice is also available for workstations, minicomputers, and mainframes. Check with MicroSim for product availability. This book and the Student Version PSpice software are available in several formats:

1. A paperback edition with software included in the book (834614).
2. A paperback edition which does not include the software (834606).
3. The software is available separately in quantity from Prentice-Hall in an IBM® PC-compatible version (834630) and a Macintosh II® compatible version (834622).

To order the software separately in quantity, please see the order card which is included in this book.

此为试读,需要完整PDF请访问: www.ertongbook.com

Introduction

PSpice helps you simulate your electrical circuit designs **before** you build them. This lets you decide if changes are needed, without touching any hardware. PSpice also helps you check your design **after** you think it is complete. This lets you decide if the circuit will work correctly outside your office, in the real world, and have good production yield. In short, PSpice is a simulated "lab bench" on which you create test circuits and make measurements (PSpice will not design the circuit for you).

The practical way to check an electrical circuit is to build it. However, by the early 1970s, the components which were connected on an integrated circuit had become much smaller than individual discrete components. Physical effects that were negligible for normal circuits, such as a stereo amplifier, became important for these microcircuits. So the circuits could not be assembled from components in the lab and give the correct test results; the circuit had to be either (i) physically built, which is expensive and time consuming, or (ii) carefully simulated using a computer program. This is why the acronym, SPICE, stands for *Simulation Program with Integrated Circuit Emphasis*.

WHAT IS PSPICE?

PSpice is a member of the SPICE "family" of circuit simulators, all of which derive from the SPICE2 circuit simulator developed at the University of California Berkeley, during the mid-1970s. SPICE2 evolved from the original SPICE program, which, as it turns out, evolved from another simulator called CANCER that was

developed in the early 1970s. Tremendous effort over this relatively short time created a simulator whose algorithms are robust, powerful, and general; SPICE2 quickly became an industry standard tool. Since this development was supported using public funds the software is "in the public domain," which means it may be freely used by U.S. citizens. The software is improved by U.C. Berkeley to the extent that it supports further research work. For example, SPICE3 is a "redesigned" implementation of the SPICE2 program that fits into U.C. Berkeley's computer-aided design (CAD) research program. SPICE3 is not better than SPICE2, in the way that SPICE2 was an advance over the original SPICE program; rather, it is designed to be a module in the U.C. Berkeley CAD research system. Neither SPICE2 nor SPICE3 is supported by U.C. Berkeley in the way commercial products are, nor does U.C. Berkeley provide consulting services. This led to commercial versions of SPICE which have the kind of support industrial customers require. Also, many companies have an in-house version of SPICE that has modifications to suit particular needs.

PSpice, which uses the same algorithms as SPICE2 (and conforms to its input syntax), shares this emphasis on microcircuit technology. However, the electrical concepts are general and are useful for all sizes of circuits (for example, power generation grids) and a wide range of applications. For instance, the simulator has no concept of large or small circuits; microvolts or megavolts are "just numbers" to PSpice. As long as PSpice is able to solve your circuit matrix, it will do so. This makes PSpice "technology independent" and generally useful. On the other hand, no assumptions are made about how the circuit should behave; for example, PSpice is not concerned that 0.03-watts output power does not make for a very loud stereo amplifier. You have to look at the results to see if they make sense in your application.

For discrete circuits (circuit made of individual parts assembled on a circuit board) PSpice has a variety of uses. Like the integrated circuit designs mentioned before, your designs are pressed for schedule time, budget expense, and manufacturing yield. With PSpice you can

Check a circuit idea before building a breadboard (even before ordering the parts).

Try out ideal, or "blue sky," operation by using ideal components to isolate limiting effects in your design.

Make simulated test measurements which are

- difficult (due to electrical noise or circuit loading),
- inconvenient (special test equipment is unavailable), or
- unwise (the test circuit would destroy itself).

Simulate a circuit many times with component variations to check what percentage will pass "final test," and find which combinations give the "worst case" results.

Once you become familiar with PSpice, you will find that it can substitute for most (but not all) of your breadboard work. Like any new tool, experience is required to get the most benefit from it.

The PSpice control statements (or "language," if you prefer) are easy to learn and use. These statements, which are collected in the file which is read by the simulator (the file is called a "circuit file"), are usually self-contained and may be understood without referring to any other statements. Moreover, each statement has so little interaction with other statements that they have the same meaning regardless of context. So the language of PSpice is easy to learn because you can focus on each statement type, master it, then move on to the next. Also, as you will see, you will not need to know many statement types to get started. Most of your difficulty will probably come from learning and operating your computer system.

OTHER SPICE-BASED PROGRAMS

The commercially supported versions of SPICE2 fall into three groups:

The original group of mainframe-based versions, including HSPICE from Meta-Software, IG-SPICE from A.B. Associates, and I-SPICE from NCSS timesharing. HSPICE focuses on the needs of the integrated circuit designer with special device model support; Meta-Software also distributes RAD-SPICE which, as you might guess, simulates the operation of circuits subjected to ionizing radiation. I-SPICE and IG-SPICE focus on "interactive" circuit simulation and graphics output (which was an innovation for mainframe users). Precise from Electronic Engineering Software is a more recent addition to this group.

The IBM-PC based programs (besides PSpice), including AllSpice from Acotech, IS-SPICE from Intusoft, and Z-SPICE from Z-Tech. These are very similar to SPICE2; in most cases no changes have been made—even to correct errors (such as convergence problems). Without serious support for the simulator, these programs fall into a more "hobbyist" class of product. However, interesting additions include pre-processors or shell programs to manage input and provide "interactive" control, as well as post-processors for refining the normal SPICE output.

Advanced programs, with "innards" that are significantly overhauled, or entirely new, but adhering to the U.C. Berkeley standards for circuit description, including SPICE-Plus from Analog Design Tools, DSPICE from Daisy Systems, and PSpice from MicroSim. Many additions and improvements are available from these products. Also, these advanced simulators have options to extend simulation capabilities and interpret results.

The "growth" portion of the analog circuit simulation business is the last group, especially with the rapidly expanding market for engineering workstations. While U.C. Berkeley remains at the forefront of computer-aided tools for engineering, as

a practical matter the complete simulation products will come from industry. Most, if not all, of the techniques you will see in this text are applicable to these products.

ORGANIZATION OF THIS BOOK

This book adopts a "graduated example" (which some call a "tutorial") approach to learning about circuit simulation. It is tempting to load new software and try some examples, well before reading the instructions, so we will channel this urge toward learning about the simulator. We will start by building a simple circuit, make some DC "measurements," and move on. The biggest hurdle seems to be running a simulator the first few times. After that you start to focus on the electronics you are simulating and how best to measure what you want to discover.

The details of the semiconductor models are described later. These models are independent of the methods for using the simulator. In fact, we will do without them entirely in the examples.

An abridged summary of the control statements and device descriptions are at the end of the book. In fact, these appear as appendices because this information is not the *raison d'être* of the book, and is applicable to the SPICE-like simulators in only a general sense. You should try to obtain a detailed guide to the simulator you will be using.

Not "everything you ever wanted to know about SPICE . . . ." is in this book. Missing are some topics that I had planned to include, as well as additional depth of coverage of the topics that do appear. These fell victim to the schedule for completing the text. Additional topics, without doubt, will become obvious from reader comments. Perhaps in the next edition. . . .

k_{ij}	Electromechanical coupling coefficient.
$k(f)$	Desired frequency characteristic.
K	Thermal conductivity. A constant.
L	Parallel inductance. Optical path length.
M	Receiving sensitivity.
N	Near field range.
P	Pressure. Proportion. Amplitude.
$P(t)$	Pressure wave. Profile of laser pulse.
Q	Q value.
r	(Transducer) radius. Radial position.
R	Load. Range. Reflectivity ratio.
s	Elastic compliance.
S	Transmitting sensitivity. Specific thermal capacity.
S_{ij}	Strain.
t	Time. Thickness. Spatial separation of pulses. Pulse length.
T	Temperature.
T_{ij}	Stress.
u	Particle velocity. Particle displacement.
$U(t)$	Received voltage pulse.
v	Volume.
V	Voltage.
w	Pulse width. Half width of beam.
W	Figure of merit for energy trapped in a disc.
x	Off-axis distance.
X_E	Reactance of electrical current.
Y_T, Y_R	Transmitter and receiver efficiency parameters.
z	Depth.
Z	Acoustic impedance.
α	Attenuation coefficient. Coefficient of linear expansion.
γ	Angle of deviation from axis.
δ	Skin depth.
ε	Dieletric coupling constant.
θ	Beam angle.
λ	Ultrasonic wavelength.
Λ	Optical wavelength.
μ	Permeability. Particle displacement.
μ_0	Permeability of free space.
$\Delta\mu$	Sound-induced refractive index variation.
ξ	Efficiency.
ρ	Density.
σ	Conductivity.
ϕ	Angle.
ω	Angular frequency.

Contents

此为试读,需要完整PDF请访问: www.ertongbook.com

CHAPTER 1

Getting Started

Let us begin with a quick circuit to introduce you to running PSpice. This will show you the basics of a circuit simulation without getting complicated by rules, details, exceptions, and so on, and quickly get to a successful result. Later, we will get to the wide range of features and ways of combining these to express complex circuit functions.

Sometimes the examples will omit features of the simulator intentionally to concentrate on a particular topic. These features are necessary for normal use, and we will get to them in due course. The examples are brief, to demonstrate an idea and there is the danger that, in not explaining somewhat unrelated items, they may mislead you. If this happens it was not intended, but is just a problem with this approach.

Sometimes the examples will repeat some of what was done already.

1.1 A SMALL CIRCUIT

The best way to learn a circuit simulator is to "do" simulations. Usually you start with a small circuit that you know, by inspection, will work.

Running this simulation requires several basic accomplishments. It requires that you (i) create the input file, or "circuit file" (although some call it a "program" for the simulator), (ii) run the simulator (without errors), (iii) find where the output went, and (iv) inspect the output.

9150064

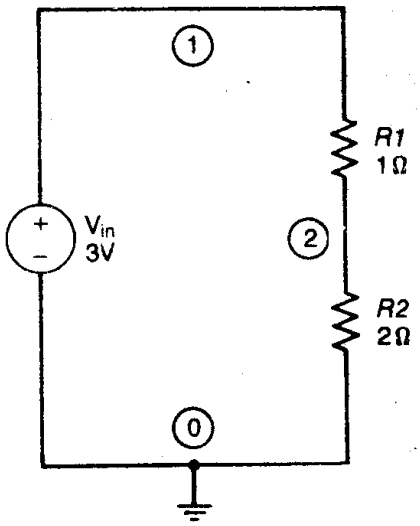

Figure 1.1 Schematic for small-circuit example.

In PSpice, the circuit file to simulate this circuit is

```
* Resistor divider circuit
VIN 1 0 3.0volt
R1  1 2 1.0ohm
R2  2 0 2.0ohm
.END
```

How you run this situation will depend on the system you are using. You will need to learn to use a text editor to create the input file. Then you will run PSpice, specifying the input file you created. If everything works, PSpice will read your input file called, for example, TEST.CIR, and place the results in an output file called TEST.OUT. The same text editor you used for creating the input file can also be used to inspect the output file. This output file may also be directed, by you, to a printer.

Exercise 1.1.1

Create and run this simulation on your system. Look at the output file. Did the printout show (correctly) node 2's voltage as 2 volts? Experiment by changing the circuit file—leave out, or add, something and see what errors PSpice will check.

Now to describe, and explain the circuit file. PSpice always expects the first line of the circuit file to be a title line. You can leave it blank, but circuit description can not start until the second line of the file. The examples in this book will sometimes start the title line with a "*" (which also indicates a comment line) by force of habit on my part. This is not necessary. What is necessary is the last line, ".END", which completes the description of the entire circuit including any simulation controls. You use ".END" because PSpice will let you start another, completely different, circuit simulation right after ".END". Between the first and last line, the circuit file may be in almost any order.

此为试读,需要完整PDF请访问: www.ertongbook.com