Preface

Circuit simulation is an important tool used when designing analog electronic circuits. Circuit simulators allow new designs to be evaluated quickly and at considerably less expense than the only other alternative, fabrication. Simulation is very heavily used, particularly in the design of analog integrated circuits, with almost all circuits being simulated before being fabricated. It is also rapidly becoming more popular for the design of board level analog circuits. However, even though circuit simulators have been available for over 20 years, getting a circuit simulator to converge and give an accurate answer is still considered an art. There is considerable folklore on how to use simulators successfully; however, much of it is based on the idiosyncrasies of particular simulators or tricks that are based largely on luck. For example, one large electronics company found that convergence difficulties sometimes disappeared when the input file was reorganized and wrote a program that designers could use to randomly shuffle the simulator input file.

The benefits and risks inherent in the use of a circuit simulator can be compared with that of an automobile. Cars are very useful, and some might say essential, and despite continuing improvements, they can still fail to get you where you are going, either because of collisions, break downs, or you simply get lost. Though, through knowledge of where you are going, how to safely operate the car, and how to properly maintain it, you can greatly increase your chances. As with automobiles, simulators are not completely reliable. Despite decades of improvements, simulators still occasionally fail to either converge or compute an accurate result.

It is impossible to write a circuit simulator that always computes accurate answers. Even attempting to write a simulator that produced accurate results in virtually all cases would result in the simulator being so conservative that it would be much too slow on circuits that do not have accuracy problems. Thus, circuit simulators are designed to work well in most cases, and it is up to the designer to be knowledgeable enough to recognize any problems that occur and correct them. Generally, circuit designers understand their circuits well enough to recognize inaccurate results, however they often do not know the best way to overcome problems. They usually rely on a trial-and-error approach to solve a problem, and often there are as many errors as trials.

How This Book is Unique

The motivation for writing this book came when I was asked to give a presentation to designers on these topics. I went to the bookstore to get material from the available simulation books. What I found was three types of books. Most books were targeted towards neophytes and were little more than users guides for various commercial simulators. They talked about simple things such as how to create a netlist and how to plot waveforms. The second type of book gave detailed descriptions of the semiconductor models provided in most simulators. The last type of book delved very deeply into simulation algorithms and was targeted towards people who write circuit simulators. None of the books addressed problems that an average designer might have with the simulator itself, which is my intention with this book.

This book is not an introduction to circuit simulators. I assume that you are already adept at the mechanics of operating a simulator and are interested in understanding how it works and increasing the skill and sophistication with which you use it. For an introduction to circuit simulators and SPICE, see Vladimirescu's *The SPICE Book* [vladimirescu94].

Why You Should Read This Book

This book is designed to take you from being a reactive user to being a proactive user. A reactive user is one that runs the simulator and hopes that nothing goes wrong. If something does go wrong, various remedies are pulled from a bag of tricks one at a time with little understanding with the hope that one solves the problem. If not, either the circuit is redesigned to avoid the problem or the simulator is not used. The proactive user anticipates problems. When one occurs, the user knows why it occurred and just what to do to resolve it. The reactive user is controlled by the simulator, whereas the proactive user controls the simulator.

I expect that by reading this book you will gain the following:

1. A basic understanding of how circuits simulators compute their results.
2. An understanding of what kind of errors occur with circuit simulators and how to recognize these errors.
3. General guidelines on how to improve the accuracy of the solution.
4. General guidelines on how to encourage the simulator to converge.
5. An understanding of what kind of errors are expected when simulating particular classes of circuits.
6. Know the meaning of the simulator's key convergence and error control parameters, such as `trtol`, `chgtol`, `gmin`, and etc.
7. Can relate simulator error messages to what actually when wrong and what should be done to fix it.

In addition, the book describes many nonobvious applications of circuit simulators.

What Is In This Book

This book presents one person's suggestions on getting a simulator to behave, based on 20 years of experience, both as a user and as a developer of circuit simulators. This book is intended to be a practical guide for circuit designers that routinely use circuit simulators. I describe problems that commonly occur and give suggestions on how to solve these problems. These issues are examined in the context of the circuit simulators SPICE2 [spice2] and SPICE3 [spice3], because they are so pervasive, and Spectre[1], a new simulator designed to address many of these problems. In fact, Spectre was modified several times in order to address issues that were uncovered while writing this book. Even though I focus on these three simulators, this book should be useful to the user of any circuit simulator.

In this book, I introduce the algorithms used by a circuit simulator with an emphasis on how convergence problems occur and how the algorithms influence the error in the results. Techniques for resolving convergence difficulties and controlling error are presented. In addition, I use example circuits, such as oscillators or switched-capacitor circuits, to illustrate the simulation problems that are inherent with certain types of circuits or to show how to make challenging measurements. The presentation is geared toward describing how designers can recognize and control common simulation problems.

Chapter 1 introduces circuit simulation, both from a historical and a computational perspective. Chapter 2 presents DC analysis and discusses the issue of convergence in depth. Convergence is important for both DC and transient analysis. Chapter 3 covers the small signals analyses, such as AC and noise analysis. Chapter 4 discusses transient analysis, with considerable attention being paid to the issue of accuracy. Chapter 5 talks about Fourier analysis. Finally in Appendix A the various simulator options are described.

The structure of each of the chapters on simulator analyses is similar. They start with an introduction, then delve into the theory or mathematical underpinnings of the analysis. Enough theory is

[1] Spectre^{TM} is a registered trademark of Cadence Design Systems of San Jose, California.

given for you to understand the basic operation of the analysis as well as its characteristics. Examples given in this section illustrate issues that result directly from the underlying mathematics. After the theory, the practical details or heuristics of the implementation are presented. Again, the focus is on presenting those details that directly result in accuracy or convergence issues. Examples given in this section illustrate the issues that result from the heuristics present in simulators. Finally, the chapter finishes up with a presentation of some of the various ways in which the analysis can be used. The intention is to present real-world nontrivial applications that illustrate how to make important measurements or how to resolve some thorny issues.

A Word About Netlists Forgive me for using Spectre netlists rather than SPICE netlists for the examples given in this book. Spectre's parameterized subcircuits makes these examples much more powerful and more self-explanatory. The Spectre language is similar to the SPICE language, so you should be able to translate these netlists without too much difficulty, as long as your simulator provides the needed components. Brief documentation for Spectre's netlist language is given in Appendix B.

Acknowledgements

I would like to deeply thank Jacob White and Alberto Sangiovanni-Vincentelli, who taught me the fine art of circuit simulation. I would also like to acknowledge the contributions of David Root of Hewlett-Packard and Mike Tu, Don Webber, and James Spoto of Cadence Design Systems. Dave helped me better understand the charge-conservation issue. Mike pointed out some of the subtle issues involved in loop-gain calculations. Don (along with Jacob) helped me work through and understand many of the issues presented in this book. And finally, James provided me with the environment and the freedom that allowed me research and write this book.

Thanks to Karin Freuler of Cadence and Mark Williams of Harris Semiconductor for volunteering to edit large pieces of this book. I

would also like to acknowledge the many circuit designers that contributed to this book, either intentionally or innocently.

Finally, special thanks go to my family, Mary, Kale, and Kara, for being so patient and supportive during the time it took to write this book.

One of the pleasures in writing a book is the opportunity afforded the author to improve both the breadth and the depth of his own understanding of a subject. I hope that in trying to convey the understanding of what I have learned while writing this book, that I am also able to share with you some of that pleasure.

Ken Kundert
March 16, 1995
Los Altos, California