

A CLASS FOR COMPUTER-AIDED DESIGN AND MEASUREMENT OF MICROSTRIP CIRCUITS

*David Rutledge, Richard Compton, Wyman Williams
Zorana Popović, and Kent Potter*

Division of Engineering and Applied Science
California Institute of Technology
Pasadena, CA 91125

Abstract

We are developing an innovative microwave-integrated-circuits class at Caltech that uses personal computers to design and analyze microstrip circuits, and to make the artwork and measurements. We have written an interactive microwave computer-aided-design program, *Puff*, that is quite fast at laying out and analyzing microstrip circuits. The students fabricate a different circuit each week and test them on a network analyzer controlled by a personal computer. The fabrication and measurement process is streamlined so that it all can be done in a single three-hour lab period.

Introduction

Two years ago, we started a new microwave course at Caltech. There are several good reasons for developing a new way of teaching microwaves. One is that the traditional microwave engineering class is mathematically oriented. Today, however, students spend much of their time studying digital electronics, and they need to be motivated to learn the mathematics of microwave circuits. We have tried to make a class that would get the students excited about microwaves, and motivate them to learn the mathematics. In addition, personal computers can be used in learning about the circuits. Microstrip substrates and gallium-arsenide FET transistors are now readily available, so that it is reasonable to make these circuits in a teaching lab. In the class, the instructor gives a lecture about a new circuit at the beginning of the week, and then assigns a homework problem to discuss at a second lecture in the middle of the week. Then the students use personal computers to design and analyze the circuit, and to make the artwork on a dot-matrix printer. At the end of the week, the students etch the circuit, measure the scattering parameters, and compare the results with the computer analysis. A similar approach has been used in a lower-level integrated-circuits class at Caltech [1]. There are currently eight different labs. We begin with a lab that evaluates coax-to-microstrip transitions, and then build branch-line and rat-race couplers, and a low-pass filter. The next two labs are based on coupled lines: a directional coupler and a band-pass filter. Then the students build microstrip patch antennas and match them to the $50\text{-}\Omega$ microstrip transmission line. Finally they build a mixer and a low-noise FET amplifier.

Puff

There are difficulties in using commercial CAD software in a university. It is hard to prevent unauthorized use and copying of these valuable programs, and at the same time to make it convenient to use them on any machine on a campus. Second, the source code and algorithms are usually proprietary, and this makes it difficult for the student to understand what the program is doing. Finally, these are large, powerful programs with thick manuals. This means that the students spend as much time learning to use the program as in learning the physics of microwave circuits. These difficulties made us decide to write our own program, called *Puff* after the magic dragon in the folk song made popular by Peter, Paul, and Mary. *Puff* runs on an IBM PC, AT or compatible. It uses the subnetwork-growth method to calculate scattering coefficients [2]. We have found students' comments extremely important in telling us which way to go. The two most important factors are:

1. *Ease of use*: The students must be able to use the program without constantly referring to a manual.
2. *Crash proof*: The students become extremely frustrated when the program hangs up, because they often do not know whether they did something wrong or if there is a bug in the program.

The students were not very concerned about the speed of execution, the range of features in the program, or the accuracy of the microstrip models. The program needs to be adequate for doing the problems that the student will encounter in the lab or in projects, but beyond this point, the bells and whistles just get in the way. In particular, an optimization routine may be counterproductive, since one of the goals of the class is to get the students to tweak the designs on their own. In the same way, including the effects of parasitic discontinuities may hide the effects from the students when they make measurements. We found it instructive for the students to take these discontinuities into account after the measurement by having *Puff* adjust the line lengths until the theory and experiment agree.

To make the program simple to use, *Puff* has only one screen display. There are three different windows on the screen: a *Circuit* layout, a *Parts* list, and *Plot* figures. You can move from window to window by pushing the function keys. Once you are in a particular window, there are about only about eight commands that can be used in that window, and they are shown on a command list on the screen. The keys for the commands in the other windows are locked out. We found it useful to severely limit the number of commands that can be used at any one time. If there are too many keys that do something, the students freeze up and retreat to the manual, and this really slows things down. A typical screen dump is shown in Figure 1.

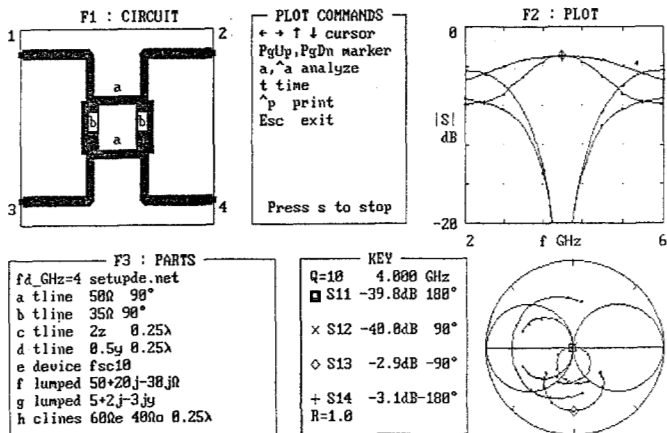


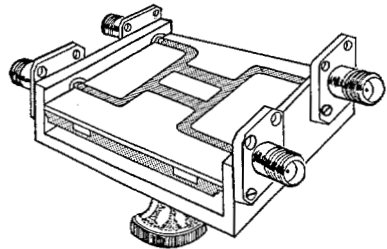
Figure 1. A screen dump for *Puff* that shows a branch-line coupler. The *Circuit* layout is on the top left, the *Parts* list is on the bottom left, and the *Plot* graphs are on the right. The currently active commands are shown at top center.

In the *Circuit* window, when you push a cursor key, *Puff* draws a part from the *Parts* list, oriented in the direction of that key. We have found it dramatically easier to lay out the circuit this way than to make a circuit configuration file. The number of errors is reduced since the circuit on the screen looks like the circuit that will be made. It is hard to get a direction or a connection wrong, or miss the impedance by a factor of ten if you can see it on the screen. The *Parts* list contains the different things that the student might want to put in the circuit: transmission-line sections, coupled-line sections, lumped elements, and user-specified two-ports and three-ports. In the *Plot* window, one can push a key to make a Smith-chart plot, a magnitude plot, and a time-domain plot for the circuit on the screen at any time. Typically this takes 10 seconds.

Fabrication

We have worked hard to make the circuit fabrication process quick. It currently takes 45 minutes. When the design is complete, *Puff* makes the artwork on a standard IBM *Proprinter*. This takes one minute. The printer makes two passes, and achieves a resolution of 130 dots per inch. Then we shrink the artwork 5:1 onto 2.5"-square glass plates, using a Pentax 6×7 camera. The plate photography takes 15 minutes, the photoresist processing 15 minutes, and the copper etching 10 minutes. The students measure their linewidths with a microscope, and the dimensions are accurate to one or two mils. The students make their circuits on 1"-square, *Duroid* substrates with a dielectric constant of 10.2. They mount the circuits on a brass block, as shown in Figure 2.

Figure 2. Brass block for mounting the circuits. The thumbscrew on the bottom locks the circuit into place. The brass plate underneath the circuit flexes to push against the sides to make a ground connection. The VSWR of the coax-to-microstrip transition is typically 1.1 or less for frequencies less than 6 GHz.



Measurements

For the measurements we have developed an inexpensive new analyzer that we call a *sampled-line network analyzer*. It belongs to the six-port family, in which complex impedance measurements are made from power measurements [3]. The circuit for the sampled-line analyzer is particularly simple, consisting of a section of microstrip transmission line with shunt diodes. Different sets of diodes are used for different frequency ranges. The diode outputs go to an A-D board in the PC. The present analyzer has five diodes and works from 0.5 GHz to 5.5 GHz. This is a broad enough bandwidth to allow us to make time-domain plots by Fourier transforming the frequency data. *Puff* makes and plots measurements on the same graphs that it uses for the analysis, so that the students can compare theory and experiment. In addition, *Puff* can line-stretch the data and measure multi-port circuits.

Acknowledgements

We appreciate the suggestions of our students, who have been enthusiastic in spite of teething problems with the software and the network analyzer. We appreciate discussions with Professor Neville Luhmann at UCLA, who has started a more ambitious two-term course with a similar philosophy. We are grateful for the donation of a sweep oscillator and a noise-figure meter, arranged by Weldon Jackson at Hewlett Packard, for *Duroid* substrates, arranged by Sharon Aspden at the Rogers Corporation, and for personal computers from IBM. The National Science Foundation supported *Puff*'s development, and Caltech's Program in Advanced Technology supported the sampled-line network analyzer work.

References

- [1] D. B. Rutledge, "A freshman class in solid-state devices and integrated circuits," *IEEE Trans. on Education*, E-29, pp. 136-141, 1986.
- [2] K. C. Gupta, R. Garg, and R. Chadha, *Computer-aided design of microwave integrated circuits*, Artech House, Dedham, Massachusetts, 1981.
- [3] G. F. Engen, "The six-port reflectometer: an alternative microwave network analyzer," *IEEE Trans. Microwave Theory Tech.*, MTT-25, p. 1075, 1977.