

Puff, an Interactive Microwave Computer Aided Design Program for Personal Computers

Richard C. Compton, Wyman L. Williams and David B. Rutledge

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

Abstract

We will demonstrate a CAD program designed for the lay out and analysis of microstrip circuits. The program runs on the IBM PC or AT. Circuit elements are selected from a parts list and drawn on the screen using the cursor keys. The analysis may then be performed directly from the screen drawing. *Puff* has been used by microwave students in classes at the California Institute of Technology and the University of California at Los Angeles.

Introduction

There are several difficulties in using commercially available microwave CAD software in an educational environment. These programs are copy protected which makes them hard to distribute onto a large number of personal computers on a campus and at the same time prevent unauthorized use and copying. Second, the source code and algorithms for these programs are proprietary, so that it is not possible to determine how the programs operate or how to modify them for your own purposes. Finally, these are big, powerful programs with thick manuals, which means that a lot of time is spent learning to use the program rather than learning about microwave circuits. We have written a program, called *Puff* after the magic dragon in the folk song, which overcomes many of these problems. *Puff* uses several unique tools to provide the user with a fast and easy-to-use microwave CAD package. The code is written in *Turbo Pascal*, a popular inexpensive compiler for IBM personal computers, and can be easily modified.

Puff

Figure 1 shows how the screen is organized. There are three main window blocks that are simultaneously displayed: a *Circuit* window, a *Plot* window, and a *Parts* list. You can move from window to window by pushing one of the function keys (F1...F3). Once you are in a particular window, there are about ten commands that can be used in that window, and they are shown on a command window in the top center of the screen. The number of commands in each window is kept deliberately small. If there are too many commands, we found that the user starts to freeze up and retreat to the manual.

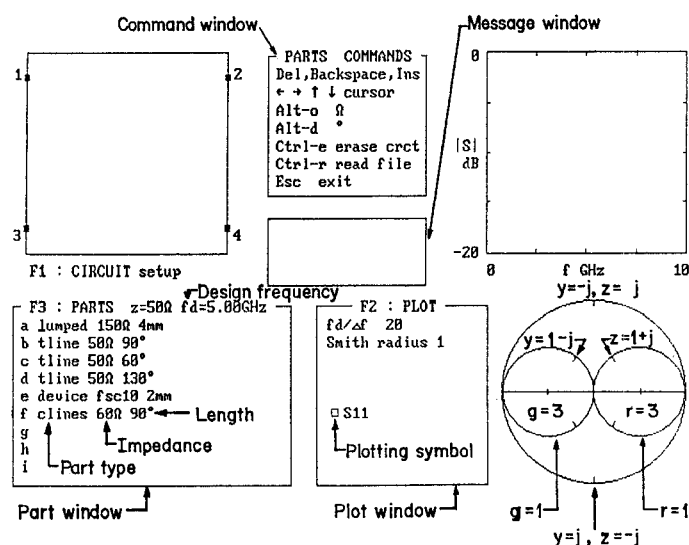


Figure 1. *Puff* Enhanced Graphics Adaptor screen dump, obtained from a dot-matrix printer.

The *Parts* list contains the different things that the user might want to put in the circuit: transmission-line sections, coupled-line sections, lumped elements, and user-specified devices like transistors. Parts may be added or changed using a screen editor built into *Puff*. When you push a cursor key in the *Circuit* window, *Puff* draws a particular element from the *Parts* list, oriented in the direction of that key. We have found it dramatically easier to draw the circuit this way rather than to describe the circuit with a text file. The circuit drawing provides information about the relative size and spacing of the components and helps with the lay out and positioning of the various elements. The number of errors is greatly reduced since the circuit on the screen is a scale version of the circuit that will be made. You cannot get a direction or a connection wrong, or miss the impedance by a factor of ten if you can see it on the screen.

One can push the **p** key to plot the circuit behaviour at any time. *Puff* displays the *s*-parameters, at each frequency, as they are calculated and then joins the points using a spline interpolation. A plot typically takes 5 seconds on an IBM AT. The *s*-parameters are displayed on a Smith chart and a rectangular dB plot. The plotting variables may be changed, to vary the plotting scales and *s*-parameters that are calculated, by typing directly over existing values. Circuit performance can be optimized by varying any of the parameters in the parts list. Each time a part is altered *Puff* redraws the circuit to take into account changes in a part's dimensions. The redraw is performed by stepping through a linked-list of key strokes that were originally used to lay out the circuit. A moveable marker allows you to read off the calculated values at different frequencies. The frequency response may then be fast fourier transformed to provide time-domain information. The time-domain plots are interesting, because they demonstrate the reflections associated with impedance discontinuities. When the design is complete, *Puff* can produce a 5:1 circuit mask on a standard IBM graphics printer. The printer makes two passes, and achieves a resolution of about 150 dots per inch.

Figure 2. shows a *Puff* lay out for a simple 4 GHz low noise FET amplifier circuit. The transistor behaviour is obtained from a file of *s*-parameters taken from the manufacturers data sheet. The stubs at the input provide noise matching while the output stubs provide a conjugate power match. *Puff* has also been used to design oscillator circuits by using a feedback loop around the transistor. In addition, *Puff* has been used to lay out and analyze branch-line and rat-race couplers, low-pass filters, and coupled-line bandpass filters.

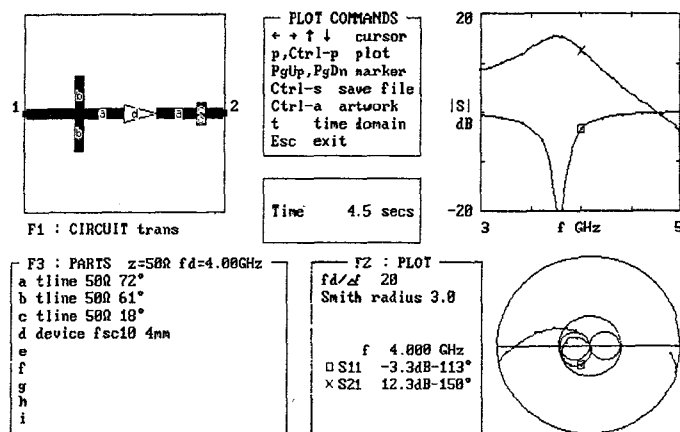


Figure 2. *Puff* lay out showing an FET amplifier circuit. Each of the elements in the *Circuit* drawing (top left) are identified by a single letter corresponding to a component in the *Parts* list. Element d is a Fujitsu fsc10 transistor. The Smith chart and rectangular plot show s_{11} , s_{21} .

Algorithms

The *s*-parameters for each of the circuit elements are combined using the subnetwork growth technique[1,2]. In this method connectors between two networks are joined and the *s*-parameters associated with the neighboring networks are recalculated using one of two simple formulas depending of whether the connectors belong to two different networks or the same network. These formulas may be derived by applying Mason's rule [3] to signal flow graph's for the two cases (Fig 3.)[4]. When all the connectors are joined the analysis is complete. The circuit is stored by *Puff* as a linked-lists of networks and connections. The dynamic nature of these lists allows them to grow when adding parts to the circuit and collapse when the analysis takes place.

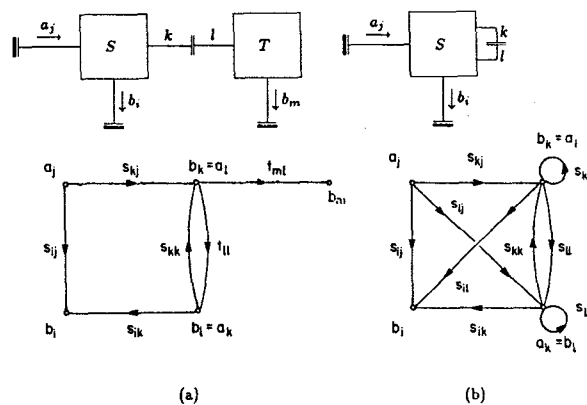


Figure 3. Block diagrams and the corresponding signal-flow graphs for the two basic operations in the subnetwork growth method: joining connectors that belong to (a) different networks or (b) the same network.

Acknowledgements

The authors wish to acknowledge assistance provided by Professor N. C. Luhmann Jr. at the University of California, Los Angeles. This work was supported by the National Science Foundation. Mr Williams acknowledges the support of the Fannie and John Hertz Foundation.

References

- [1] G. Filipsson, "A new general computer algorithm for *s*-matrix calculation of Interconnected multiports," *Proc. 11th Euro. Microwave Conf.*, pp. 700-704, 1981.
- [2] K. C. Gupta, R. Garg, and R. Chadha, *Computer-aided design of microwave integrated circuits*, Artech House, Dedham, Massachusetts, 1981.
- [3] S. J. Mason, "Feedback Theory—Further Properties of Signal Flow Graphs," *Proceedings of the IRE*, **44**, pp. 920-926, July, 1956.
- [4] R. C. Compton and D. B. Rutledge, "Microwave Computer Aided Design and Education using Personal Computers," to be submitted to the *MTT Transactions Special Issue on Computer Aided Design*.