Massachusetts Institute of Technology

Masters of Engineering Thesis Proposal

Circuit Simulation with Automatic Protoboard Layout

Author:

Michael Mekonnen

Supervisor:

Dennis Freeman

April 19, 2013

CIRCUIT SIMULATION WITH AUTOMATIC PROTOBOARD LAYOUT

1. INTRODUCTION

In this paper, I will propose the project I plan to complete for my Masters of Engineering Thesis. The project will involve creating a new teaching tool for 6.01, Introduction to Electrical Engineering and Computer Science I. The tool will allow students to draw schematics of circuits and analyze them. Furthermore, it will automatically generate the corresponding protoboard layouts using the schematic drawings.

In Section 2 I will discuss a comparable tool, Circuits Maximus (CMax), that is already available for 6.01 students. In Section 3 I will discuss the motivation for building a new simulation tool and its potential improvements over CMax. In Section 4 I discuss previous work (or lack there of) done related to this new simulation tool. Finally, in Section 5 I go into a little bit of detail regarding the implementation of the tool.

2. BACKGROUND

My focus in this project is the circuits module of 6.01. In this module, students get an introduction to the world of circuits and get hands-on experience through various labs. In a typical circuits lab, students first design a circuit by drawing a schematic drawing of the circuit and discuss their design with a staff member. After they are happy with the

schematic design, they build the circuit using a simulation tool called Circuits Maximus (CMax)[1]. With this tool, students can layout their circuits as if they were laying it out on a protoboard, and test the circuit to make sure that it behaves as expected. This way, if there is a problem with their circuit layout, they can easily debug it with CMax as opposed to debugging it on a physical protoboard. CMax also provides a very safe way of debugging circuit layouts as it is only a simulation. Once the students are satisfied with their observations from CMax, they build their circuits on a physical protoboard and carry out the appropriate experiments.

CMax has been a fantastic resource for 6.01 students. Its introduction made learning circuits significantly easier for many students, especially those that have no prior experience with circuits. In addition to making the lab exercises much more manageable, it provides students with a very handy way to build and analyze circuits at their own leisure.

3. Motivation

While CMax is a fantastic tool, we can imagine a tool that can be even more useful. The most instructive part of the labs that students do in the circuits module is really designing the circuits in the first place, which they do by drawing schematic diagrams on paper. Once they are happy with their schematic diagrams, they proceed to laying them out with CMax. The process of laying out a schematic drawing of a circuit on a protoboard (be it a physical one or on CMax) does not really have very much instructive substance. This process

is essentially solving a puzzle, and has almost nothing to do with the subject matter, which is designing circuits. In fact, when the circuits get complicated and involve several pieces, translating a schematic diagram into a protoboard layout gets to be quite challenging and time consuming. In these situations, students often end up with unpleasant layouts with diagonal and crossing wires (which are often undesired). Students sometimes end up with complicated layouts with so many wires that are very difficult to debug should something go wrong.

Hence, in the best case scenario, students should not have to produce protoboard layouts for their schematic drawings. Indeed they should work out the right schematic drawing of the circuit of interest, but the layout generation should not be part of the learning process. My MEng Thesis project will, thus, be a tool that lets students draw and analyze schematic drawings of circuits and produces the corresponding protoboard layouts automatically. Given the protoboard layouts output by this tool, students can proceed to building the circuits on physical protoboards and carrying out the appropriate experiments.

With this tool, a typical circuits lab would go as follows. First, as before, the students draw schematic drawings of their circuits on paper. Once they have schematic drawings they are happy with, they can directly draw their schematic drawings on the simulation tool. In fact, students my go straight to building the schematic drawings on the simulation tool, bypassing the experimentation on paper. Once they have a schematic drawn, they can analyze it with the tool, discuss it with staff members, and amend it easily and quickly with the user-friendly

graphical user interface of the simulation tool. When they are satisfied with their schematic drawing, they can produce the corresponding protoboard layout simply at the click of a button – this would be the most important advantage of this tool. They can then build the layout on a physical protoboard and carryout experiments with it.

4. Previous Work

As discussed so far, the simulation tool I will build is meant to be an improvement on CMax, and it will be tailored specifically for 6.01. As such, the only directly related previous piece of work is CMax. I have already discussed the motivation behind creating this new simulation tool as an improvement on CMax.

There are several very general circuit simulators available, but none that fit directly into the material presented in 6.01. When it comes to automatic protoboard layout, I was not able to find any tools that do that. On that front, this new simulation tool will be a first.

5. Implementation

In keeping with the implementation of the current tools in 6.01, I plan to build my circuit simulation tool entirely in the Python programming language. This is because I have significant experience developing in Python, and the language is flexible enough to allow any of the things I plan to do in this project.

There are three important pieces to this simulation tool.

5.1. **A user-friendly GUI.** It is important that the GUI be very easy and intuitive to use to allow for quick building of schematics as well as

quick amending of schematics already drawn. The GUI will essentially be composed of two parts. The first is a large (initially empty) area in which the schematics will be drawn. The second is a pallete that provides all the available circuit components. The components will be precisely the components necessary for 6.01 labs, i.e. the ones currently available in CMax. Students will be able to select circuit components from the palette to put in the drawing area, and they will also be able to connect different circuit components using wires. The GUI will also allow easy saving and opening of files, undoing and redoing actions, etc.

I will use the Python Tkinter module to implement this GUI[2].

5.2. **Circuit Analysis.** The tool will provide the ability to analyze circuits by allowing students to probe voltages and currents in the schematics they draw. In fact, CMax already allows students to probe voltage differences, so what the new simulation tool will do will be very similar to what CMax does to analyze circuits.

5.3. **Automatic Protoboard Layout.** This is of course the most important part of the project, and the novel idea put forth in the project. The way I will tackle the automatic layout problem is by formulating it as a search problem. The very general steps to this process are given below, and these steps will be elaborated upon in detail in my final Thesis write-up:

(1) Extract all the pieces from the schematic drawing.
(2) Figure out how the pieces are to be interconnected from the schematic drawing.

(3) Figure out a good way to place the pieces on a protoboard so as to require as little wiring as possible.

(4) Figure out a way to wire up the pieces. This step can be formulated as a search problem, in fact 6.01 provides a great infrastructure for this using the A^* algorithm[3].

References

[1] Circuits Maximus, `http://mit.edu/6.01/www/cmax_docs/cmax_docs.html`.

[2] Python Tkinter module, `http://docs.python.org/2/library/tkinter.html`.

[3] 6.01 Course Notes, Chapter 7: Search, `http://mit.edu/6.01/www/handouts/readings.pdf`.