



N0AX

# HANDS-ON RADIO

## Experiment #83 — Circuit Simulation, *Part 1*

This month, we're going to begin a multi-part article to get you started with circuit simulation. The 2010 edition of *The ARRL Handbook* has a brand new chapter on computer aided design (CAD) by Dave Newkirk, W9VES. Many readers will want to try out some of the suggestions and techniques in that chapter, so this seems like a good time to get started with a professional quality circuit simulator. This column will lead you through the process of downloading and installing your very own simulation software, Linear Technology's *LTspiceIV*, which comes at a ham-friendly price — free!<sup>1</sup>

Simulation tools are ubiquitous in the engineering world these days. In fact, it's quite unusual for a product design to be created purely on paper or amid swirls of rosin smoke on the workbench. Circuit designers stay at the computer until they are confident in their creation before picking up a soldering iron. Hobbyists also use simulation tools, but before we start, some general cautions are in order.

### "Here be Dragons!"

That legend prominently indicated *terra incognita* on early maps. As with exploring a new world, simulation is often full of dragons for the beginning circuit designer for whom everything is unfamiliar. Nevertheless, this is no reason to avoid taking the simulation plunge! Amateurs use antenna modeling programs — electromagnetic simulators — to great effect. They have learned to recognize the trickery of a "trouble dragon," unrealistic gain; excessive bandwidth; extreme sensitivity to small changes in frequency, orientation, or size, bizarre impedances and so forth. *Caveat simulator!* By knowing and respecting the limits of the tool, excellent and useful results can be obtained.

Trouble dragons live at the numerical limits of the models and mathematics on which the simulation is based. Circuit simulators

create equations for voltage and current at each point in the circuit. All of the equations are then solved simultaneously for one instant in time, called a *time step*. This data is stored and used as input to solving them all again one time step later. By repeatedly solving, storing and solving again, a numerical picture of the circuit's behavior is built up.

Computers have their limits. The equation variables have limited precision so that every value is rounded by a tiny amount. Time steps, while very short, are still finite, leading to problems if their duration becomes significant with respect to the frequency at which the simulation is performed. The circuit models used to describe the components themselves approximate how a real world component actually behaves. These small cracks are how the trouble dragons get in.

If your design involves very high or very low power, very low noise or very high frequency signals, then you should be using a simulator designed specifically for that type of application. For example, the *Handbook's* CAD chapter uses examples generated by the free SV2 student-version of *Ansoft Designer*.<sup>2</sup> That software package is designed

<sup>2</sup>Ansoft, [www.ansoft.com](http://www.ansoft.com). *Ansoft Designer SV2* is a student version of the full-featured *Ansoft Designer* simulator package for signal analysis and communications applications.

for use at RF and has a number of special tools for use in RF systems.

### The Simulation Cycle

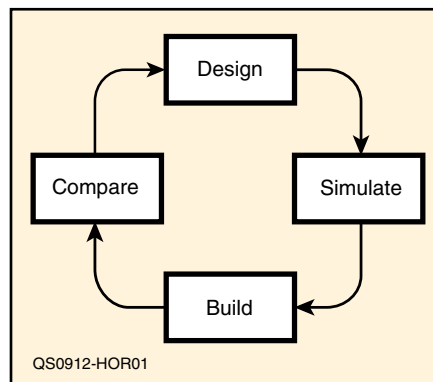
There is no smell of burning resistor or overheated transistor in a simulation. The placement of components on the screen has no effect on the behavior of the circuit, so a high-gain stage whose input is too close to its output will never break into oscillation. The dc power sources are free of ripple and noise. These effects and many more can only be experienced (and remedies learned) by building real circuits.

Figure 1 shows the process by which you really, really learn circuit design — from concept to finished project. The first step is to select a type of circuit and describe what it is supposed to do — the *performance requirements*. For example, an amplifier will need to achieve some level of gain over some frequency range. You may need a certain input impedance and output impedance. Armed with that information, choose a circuit and come up with a preliminary set of component values by using pencil and paper or a computer design tool. This is your *design*.

Next, *simulate* the circuit's performance. If the result satisfies your performance requirements, you can move to the next step. If not, change the circuit in some way (or change your requirements) until you are satisfied.

Now *build* your design as a real world collection of components and verify that the circuit works. This is where the real fun begins as the effects of construction and actual component variation take effect. Are you done? Not yet!

To soak up every bit of design experience and know-how, go back and *compare* your actual measured performance to what the simulator predicted, particularly near the limits of the circuit's function. Look for design sensitivities by substituting different parts or values. If the circuit's behavior diverges from the simulator's predictions, now is the time to take a closer look. You may not be able to say exactly why differences are present, but you'll be aware they exist and that will map a bit more of the unknown coastline.



**Figure 1 — Getting the most out of circuit simulation requires that you compare what the simulator predicts with how the actual circuit behaves.**

<sup>1</sup>Linear Technology, [www.linear.com](http://www.linear.com). *LTspiceIV* is the simulator portion of the switchmode power supply design package, *SwitcherCAD III*. It has been made available for use subject to the restrictions described in the *LTspice User's Guide*.

## Setting Up Your Simulator

Okay, enough philosophy! Start by browsing to Linear Technology's Web site, [www.linear.com/software](http://www.linear.com/software). Download the following three items: *LTspiceIV*, the *LTspice User's Guide*, and *LTspice Getting Started*. Register with Linear Technology in order to receive notices of new versions of the software and other related information.

To run *LTspiceIV* effectively, your PC will need to have at least 128 kB of RAM and at least 200 MB of free hard drive space available as simulations can generate a large amount of data. If the program runs out of space, you'll get an OUT OF MEMORY message. See the FAQ section of the *User's Guide* for information on system requirements. (*Linux* users can run *LTspiceIV* in the emulator software *WINE*.) The file containing *LTspiceIV* is a self-extracting .exe file. Double click on the file to begin the installation process. The process takes only a few moments and installs a shortcut to *LTspiceIV* on your desktop.

You immediately clicked on the shortcut and launched the program, didn't you? Admit it! Okay, so did I! What you need to do now is to open the PDF document *LTspice Getting Started*. For general-purpose use, review pages 14 to 26 that show the simulator's basic operating tools.

## Entering a Circuit

This month, we will just enter a very basic circuit to get used to the controls. Start a new schematic as shown on page 14. (The section "Schematic Capture" in the *LTspice Userguide* will provide additional information.) You should see the *LTspice* toolbar and status bar at the top of the screen, and window tabs at the top left. Under the VIEW menu, left click to turn on a field of guiding dots. (From here on, "click" means "left-click" unless stated otherwise.)

Now you'll create the two resistor voltage divider shown in Figure 2. Click the RESISTOR button in the toolbar and move the cursor into the schematic area. You'll see a black resistor symbol with two blank boxes next to it. Click once to create R1, then move the cursor and click again to create R2. (Creating R2 below R1 will make wiring the circuit a little easier.) Press the ESC (escape) key to turn off the resistor tool.

Move the cursor over R1 and the hand symbol appears. Right-click to open the PROPERTIES window for R1, enter 1000 in the RESISTANCE ( $\Omega$ ) window and click OK. R1 is now shown on the screen as having a value of 1000 ( $\Omega$  assumed). Assign R2 a value of 2200  $\Omega$  in the same way.

Click the WIRE button in the toolbar and a set of crosshairs will appear. Move the center point of the crosshairs over one of the R1 terminals and click. Move the crosshairs to one of the R2 terminals and click again,

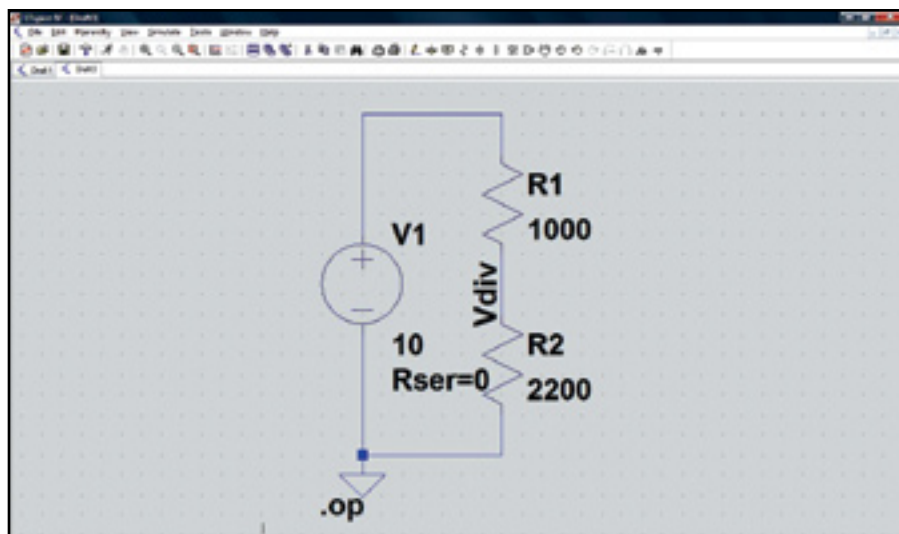


Figure 2 — A simple two resistor voltage divider. Each component is labeled with a designator and value. The ground symbol establishes a reference voltage point for the simulation. The simulator command to compute the circuit's dc operating point is .op.

Table 1  
DC Operating Point

V(n001):	10	Voltage
V(Vdiv):	6.875	Voltage
I(R2):	0.003125	Device_Current
I(R1):	0.003125	Device_Current
I(V1):	-0.003125	Device_Current

then press the ESC key. A blue "wire" now connects R1 and R2.

Click the 'Component' button in the toolbar, scroll right to find the word VOLTAGE and click once to highlight it. The voltage source symbol will appear above the list of components. Click OK, then place the voltage source on the schematic by clicking once at the desired location and press the ESC key. Open the voltage source's PROPERTIES window and assign a value of 10 V for voltage and 0 for series resistance. Use the WIRE tool to connect the positive terminal of the source to the remaining terminal of R1 and the negative source terminal to the remaining terminal of R2.

Circuit simulators require that you identify a specific point to use as a reference voltage. This is what the GROUND symbol means — not that the circuit is necessarily at Earth potential. Click the GROUND button in the toolbar, place a ground symbol near the negative source terminal and connect it to the terminal or to the wire connected to the terminal, which forms a connection dot. You should now have a circuit that looks something like Figure 2.

## Running a Simulation

Each of the connections between com-

ponents is called a *node*. The simulator calculates current through every circuit branch and voltage at every circuit node, assigning each node a label. To make it easier to read, the computer's output labels the nodes. In the divider circuit, label the connection between R1 and R2 as *Vdiv* by clicking on the LABEL NET toolbar button, entering *Vdiv* into the text window, click OK, placing the attachment point (the small box) on the "wire" between R1 and R2, then click once, followed by pressing the ESC key.

Under the SIMULATION menu, click RUN. In the EDIT SIMULATION window, select the DC .op PNT tab. Click OK and two things happen: the label .op appears on the schematic and a window appears with the results of the simulation in Table 1.

Here's what Table 1 is telling you, line by line: V(n001) is the source's 10 V output voltage. V(Vdiv): 6.875 is the voltage at the *Vdiv* label you placed and is equal to  $10 \times 2200 / (1000 + 2200) = 6.875$  V. I(R2) is the 3.125 mA current through R2, which in a simple series circuit like this is also equal in magnitude to the current through the source, I(V1), and R1, I(R1). The current in the source is negative because it flows *out* of the positive terminal. This is the circuit's dc operating point — you've just run your first simulation! Under the FILE menu, click SAVE AS and save the schematic. You can now experiment to your heart's content!

## Further Reading

This would be a good time to join the *LTspice* User's Group at [groups.yahoo.com/group/LTspice](http://groups.yahoo.com/group/LTspice) where you'll find tutorials and other useful information. The CAD chapter of *The 2010 ARRL Handbook* will also start making sense. 