

Pspice Tutorial for ELEN 3081

Written by Menachem Gielchinsky

Introduction

revision history: see last page

Welcome to Pspice. Pspice is a program Electrical Engineers use to simulate circuits. In the past an older program known as Hspice was used. Hspice was entirely word based, while Pspice allows you graphically design your circuit. Some of the more adventurous or lazy among you might not read this whole manual. Please don't give up before you have read at least the next four lines.

- (1) Always use a ground.
- (2) Always use the ground named zero, with the zero next to it in its picture. (see Figure9)
- (3) Make sure all your components have different names.

Installation

The first step on your Pspice journey is installing Pspice. This tutorial is meant for OrCad version 9 demo. Your professor probably has a copy of it on courseworks. Once you have downloaded it double click on the file to open it. You might have to unzip it first, make sure you know where you are unzipping it to. Then run setup. If you turn off you antivirus software as the installer tells you to, remember to turn it back on after the installation. You should get a screen like Figure1; make sure that "Capture" is selected.

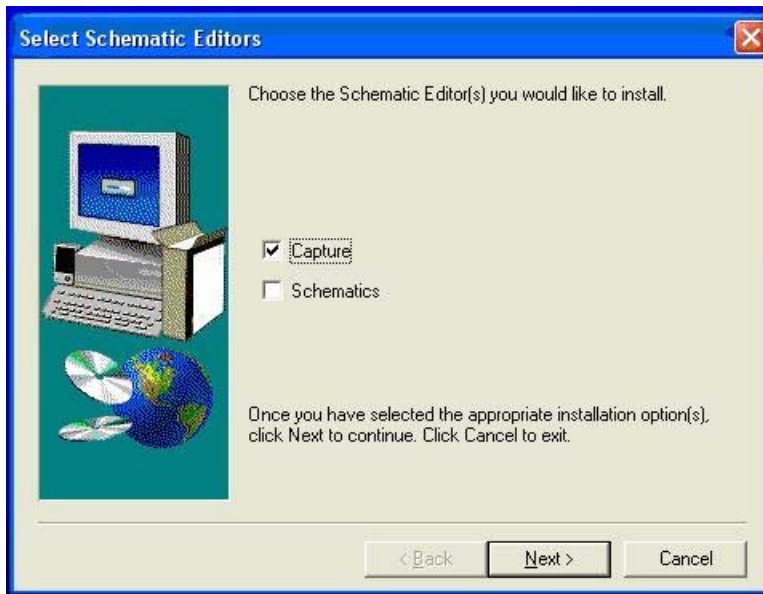


Figure1

If you don't change anything, the program will install to

C:\Program Files\OrCad_Demo.

Running Pspice

Open the start menu and find the “Pspice Student” folder and click on “Capture Student”.



Figure2

When OrCad opens, click on “File” then “New” then “Project” (or click on the little sheet of paper in the top left corner). The screen in Figure3 will popup. I named my project “Bunch of resistors” and put it in the location “C:\stuff” though you are welcomed to name it whatever you want and put it wherever you want, just make sure “Analog or Mixed A/D” is selected. Click ok and in the next popup click on “Create a blank project”.

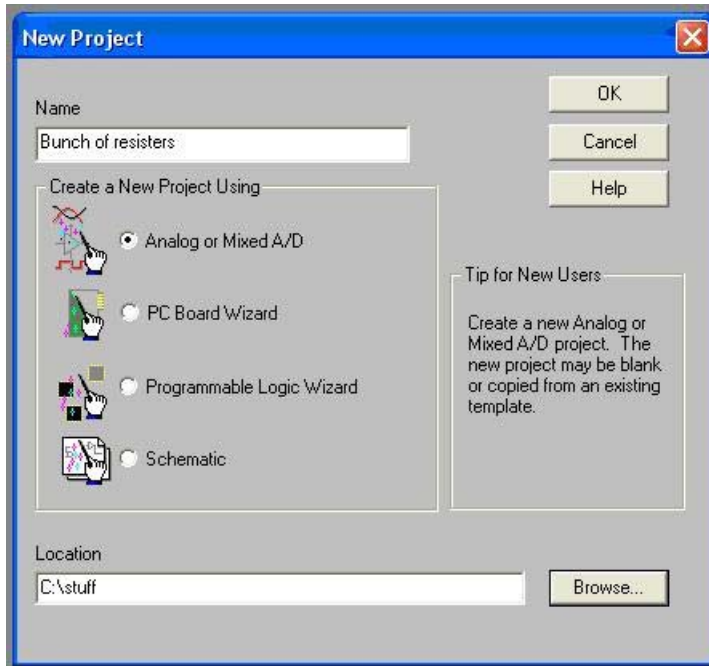


Figure3

Two windows will popup (as seen in Figure4). The one with the dots will be referred to as your schematic. In the explorer window you will notice that you have a folder called SCHEMATIC1 with PAGE1 in it. By default, SCHEMATIC1 is considered by OrCad to be the “root” schematic directory; this means that everything in this directory (just PAGE1 for now) will be included in any simulations you perform. Other folders can be added at the same level as SCHEMATIC1, but these will be ignored unless their contents are placed in SCHEMATIC1 by using a subcircuit (more about this later). You can change the name of the top level schematic, or even assign top level status to another folder you create, but for now we’ll leave the defaults in place.

Your schematic is currently called PAGE1. You can change that by right clicking on SCHEMATIC1 or PAGE1 in the explorer window and clicking on “rename”.

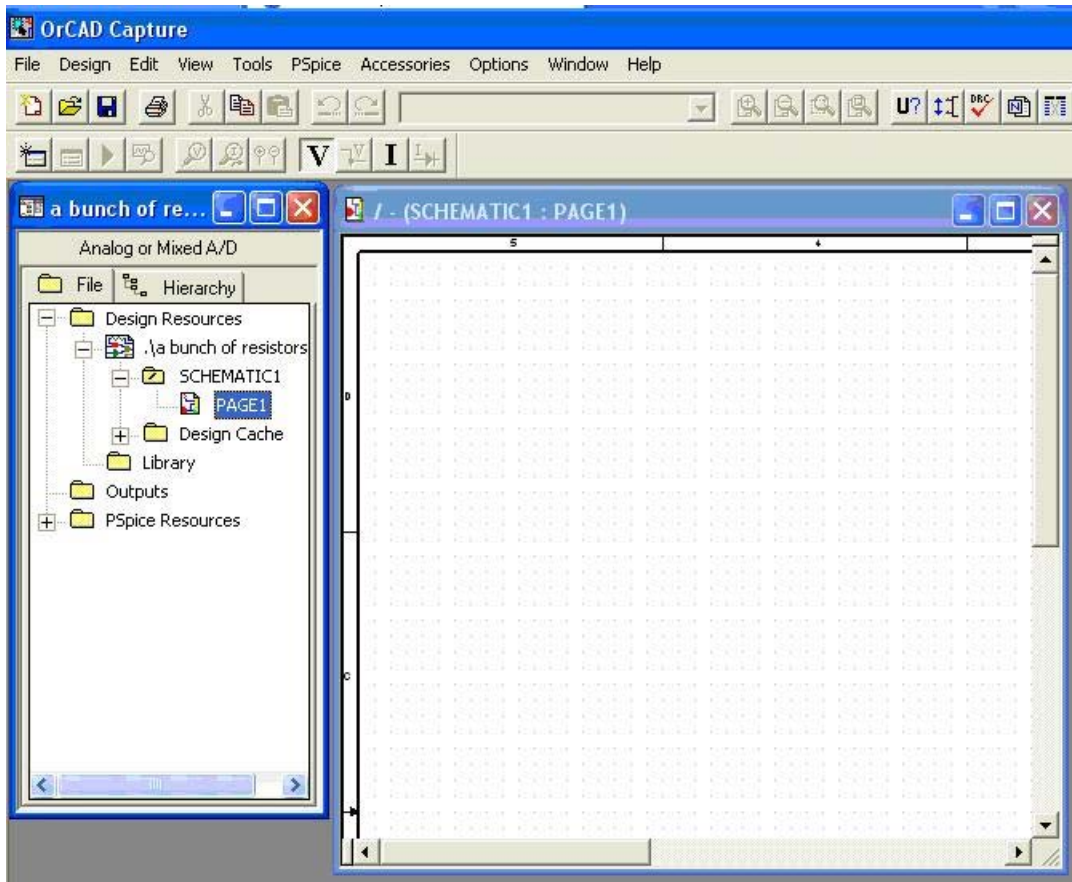


Figure4

Now, click on your schematic. You will see a toolbar appear on the right side of your screen. Click on the icon second from the top (or just press "P"). That is the "Place part" icon. In the lab you keep all your parts in a little plastic box, in Pspice every part is accessed through that icon. Now click on "Add Library" and add "SOURCE" then add "ANALOG". You now have three libraries available to you (Design Cache is parts already in your schematic). You can browse through part lists of individual libraries, or you can browse through all three at once by holding down Ctrl and clicking on all the libraries (see Figure5).

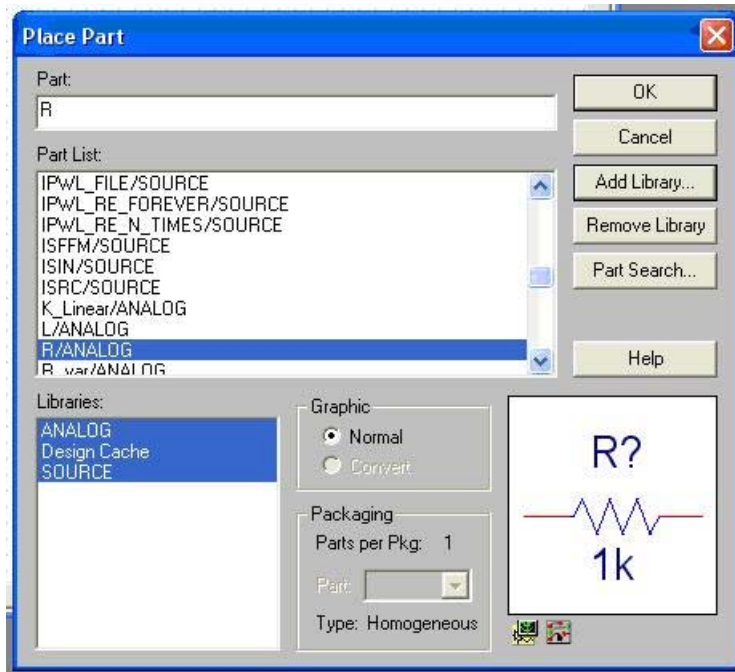


Figure5

Find the standard resistor (named “R” in library “ANALOG”) and click ok. Notice that you now have a resistor attached to your cursor. Press “R” to rotate the resistor (like in Tetris). Left click to place the resistor. After placing the resistor, you still have a resistor attached to your cursor, place three more in the configuration in Figure6 then press “E” (you are now back to the regular cursor). Notice that each resistor has a name (R1, R2...), two boxes, and a standard value of 1k. The two boxes are the contact points.

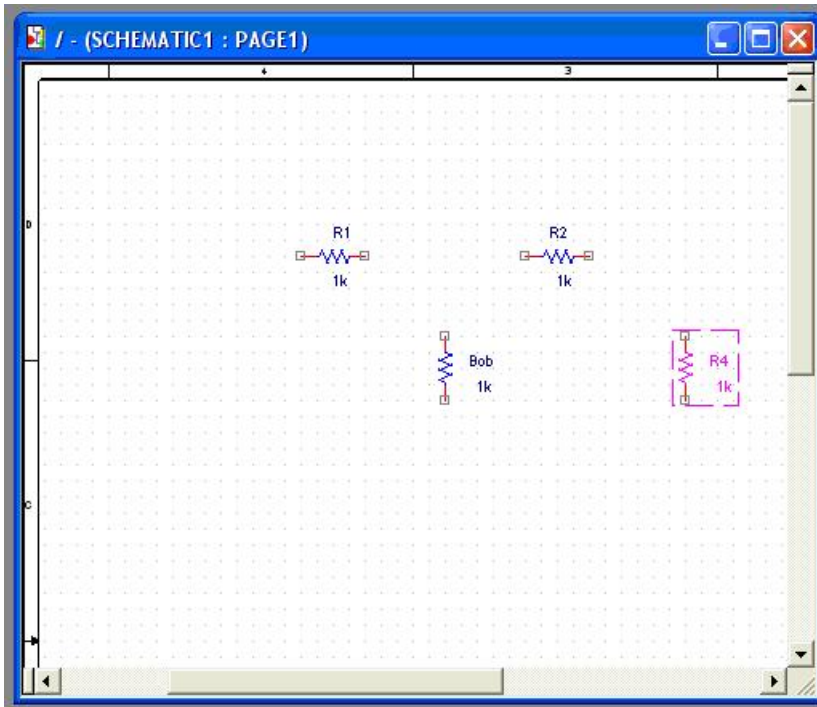


Figure 6

To change a resistor's name or value, double-click on the name or value. You should get a popup like Figure7. If you didn't, you probably clicked on the wrong part of the resistor. Close the incorrect popup and try to click on the name or value of the resistor. I renamed R3 to Bob though you can name it whatever you want, EXCEPT MAKE SURE EACH PART IN YOUR SCHEMATIC HAS A DIFRENT NAME.

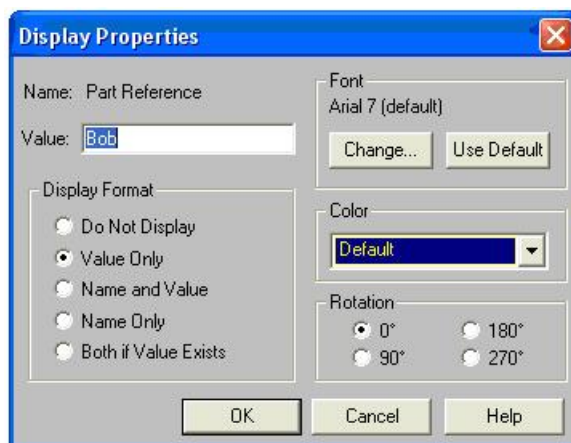


Figure7

Now that you have placed your 4 resistors we will place a DC voltage supply. Click on the "Place part" icon, and in the "SOURCE" library find "Vdc" and place it on the left of your resistors and press E. Now we need wires to connect everything. Click on the third highest icon (or press "W"). Now click on one of the boxes of one of the components to start the wire and then click on a different box to end the wire (a red dot

means a connection will be made there when you click). You want the schematic to look like Figure 8. To make the schematic look nicer sometimes (such as the wire from R4 to Bob) you have to click at each corner to anchor the wire at that position. If you want to delete a wire (or component) you can do that by pressing “E” (to get to the curser) and selecting that component/wire and pressing the “Delete” key.

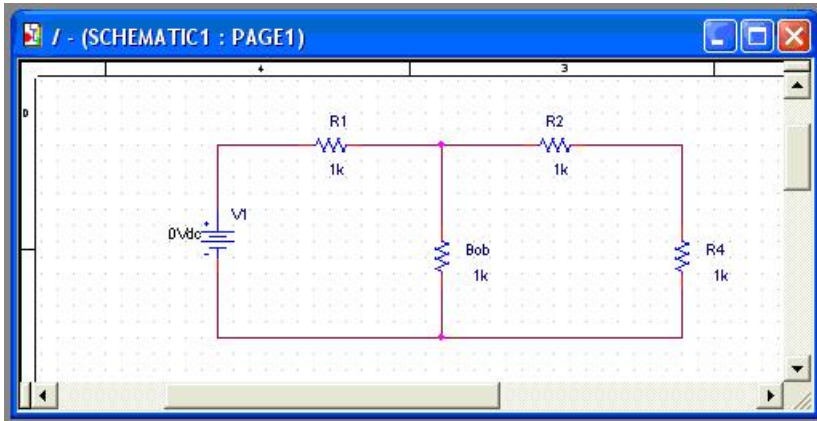


Figure8

V1 is currently at 0Vdc so change it (like we changed the resistor name before) to 10Vdc. Now to add a ground click on the icon that says GND on the toolbar (or press “G”). Then add library “SOURCE”, then choose the ground symbol with the zero next to it like in Figure9.

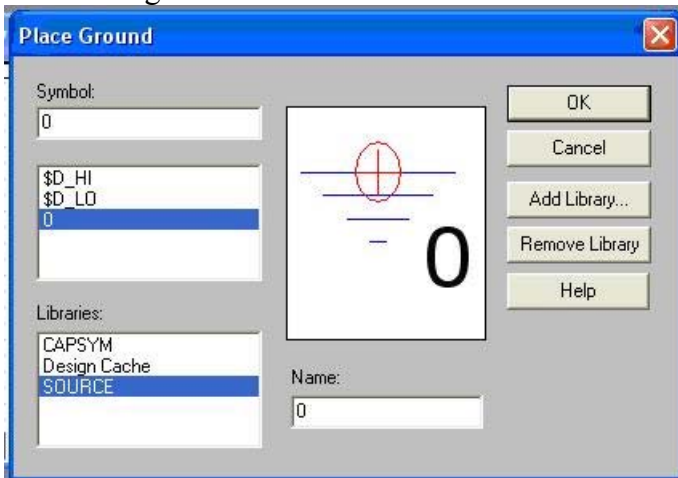


Figure9

IT IS EXTREMELY IMPORTANT THAT YOU USE THE GROUND SYMBOL NAMED ZERO WITH THE ZERO NEXT TO IT OR ELSE NOTHING WILL WORK. YOU ALWAYS NEED A GROUND IN YOUR PROJECTS. FOR THE FIRST WEEK OR SO THIS IS THE MOST COMMON MISTAKE STUDENTS MAKE.

Now your schematic should look like Figure10

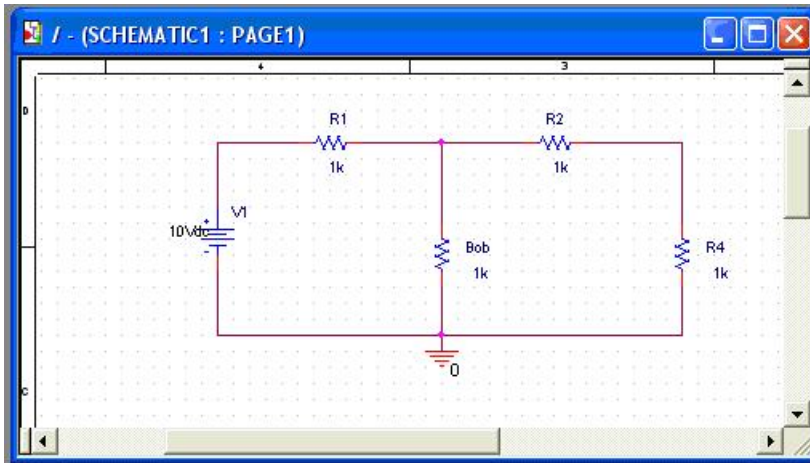


Figure10

Now we will test our circuit. Click on the “Pspice” menu, and click on “New Simulation Profile”. In the popup window, give this simulation a name (such as “test”) and make sure “Inherit From” is on “none”. Click “create”. You will get a popup identical to Figure11, for now ignore everything on it and click ok.

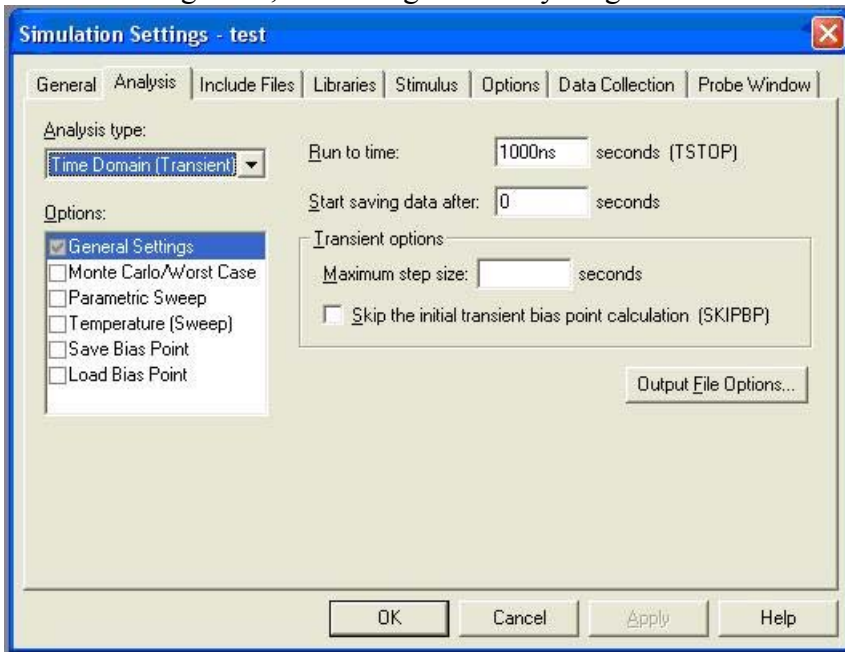


Figure11

Now click on the Run button on the top left corner of the screen (under the save disk). A black screen with some numbers on the bottom will popup, close that. You will get something like Figure12. On the top you have the option to show or hide the voltage (V) and current (I) at each node. Pspice is often helpful when trying to do homework.

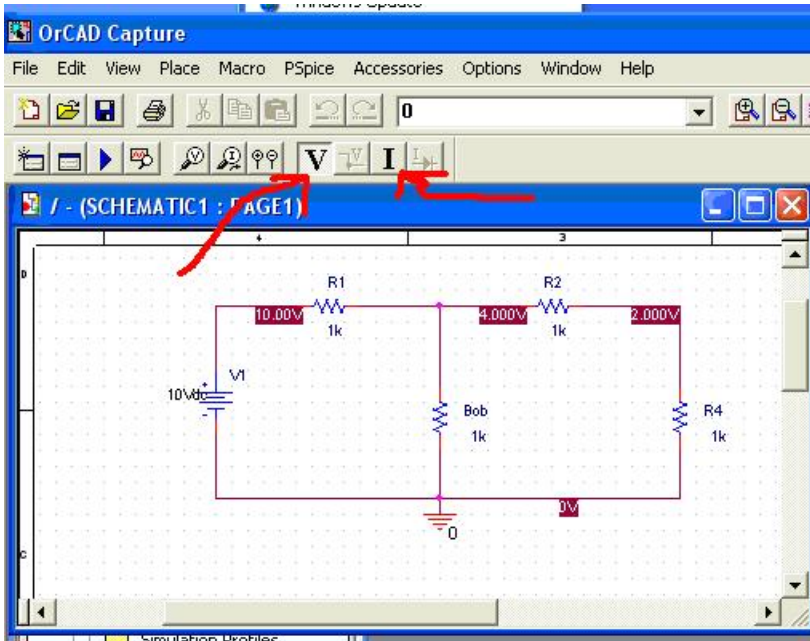


Figure12

Initial conditions on capacitors

Now add a capacitor (named "C" in library "ANALOG") as in Figure13.

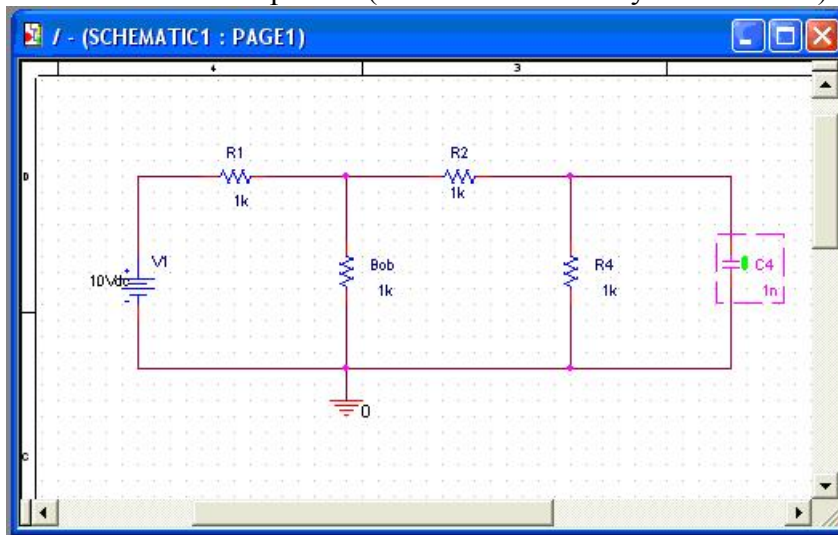


Figure13

There are two ways to add initial conditions to a cap. One method is to double click in the area around the green smug I made in Figure13. If you double-clicked in the right place you will get Figure14.

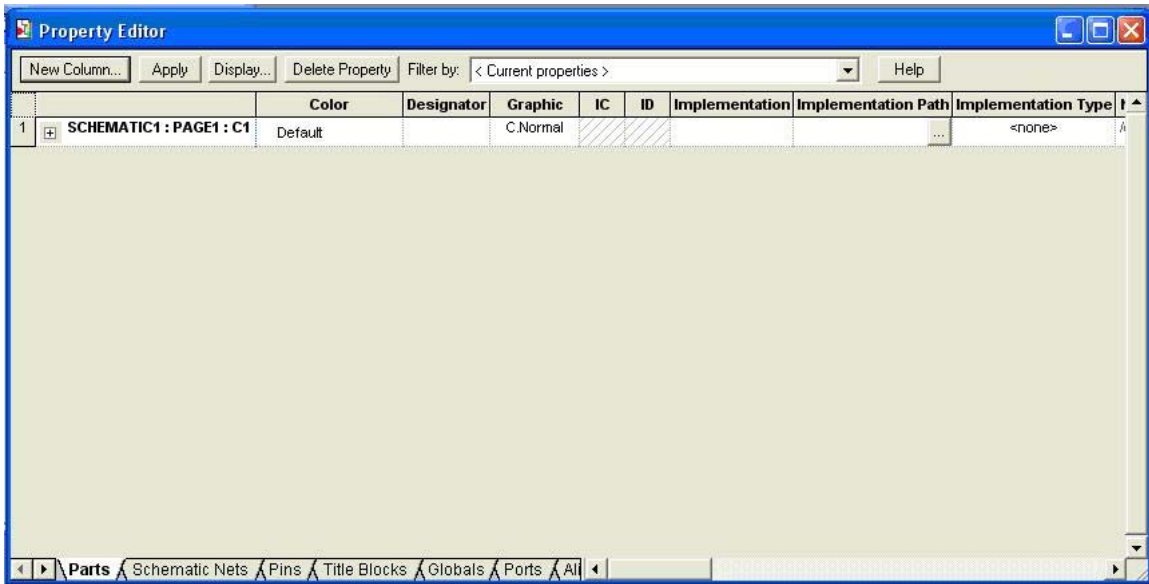


Figure14

In the shaded box under IC (which stands for initial condition) put in some number for the initial voltage (I used 1), press enter, and then close the “property editor”. The other method is by using the IC1 part (if the capacitor is connected to ground) or by using the IC2 part (if the cap is between two non-grounded nodes). These parts can be found in SPECIAL.OLB.

Click on the Pspice menu (on the top next to File, Edit, View) and click on “Edit Simulation Profile”. Figure11 should popup again. Change the Run Time to 10000ns and click “OK”. Click on your schematic, and then click on the thing that looks like a magnifying glass with the “V” in it (Figure15). That is your voltage probe. They can be placed before or after the running your simulation (you can also use the current probe, or the voltage difference probe which are located right next to it).

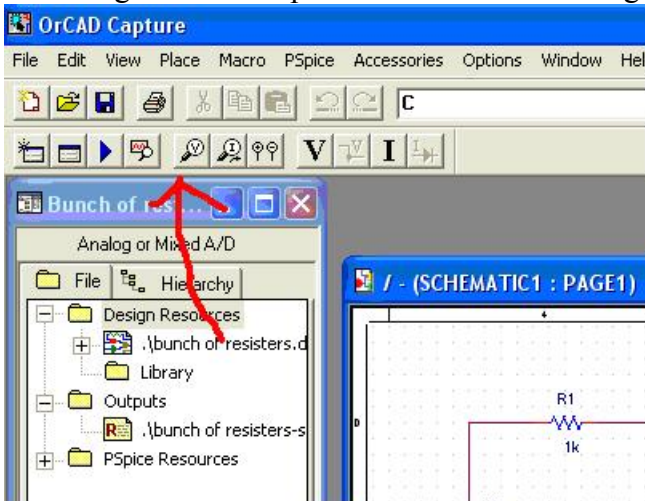


Figure15

Click on a few nodes in your schematic as I did in Figure16, then press “E”, then click the Run button.

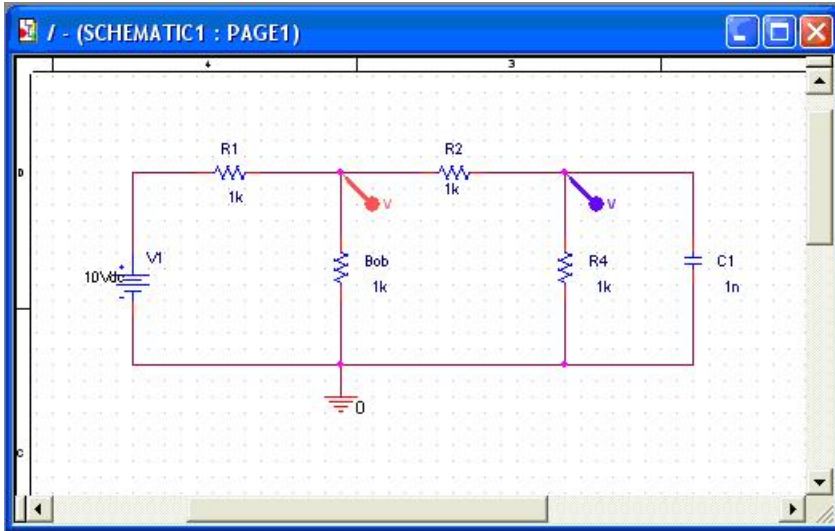


Figure16

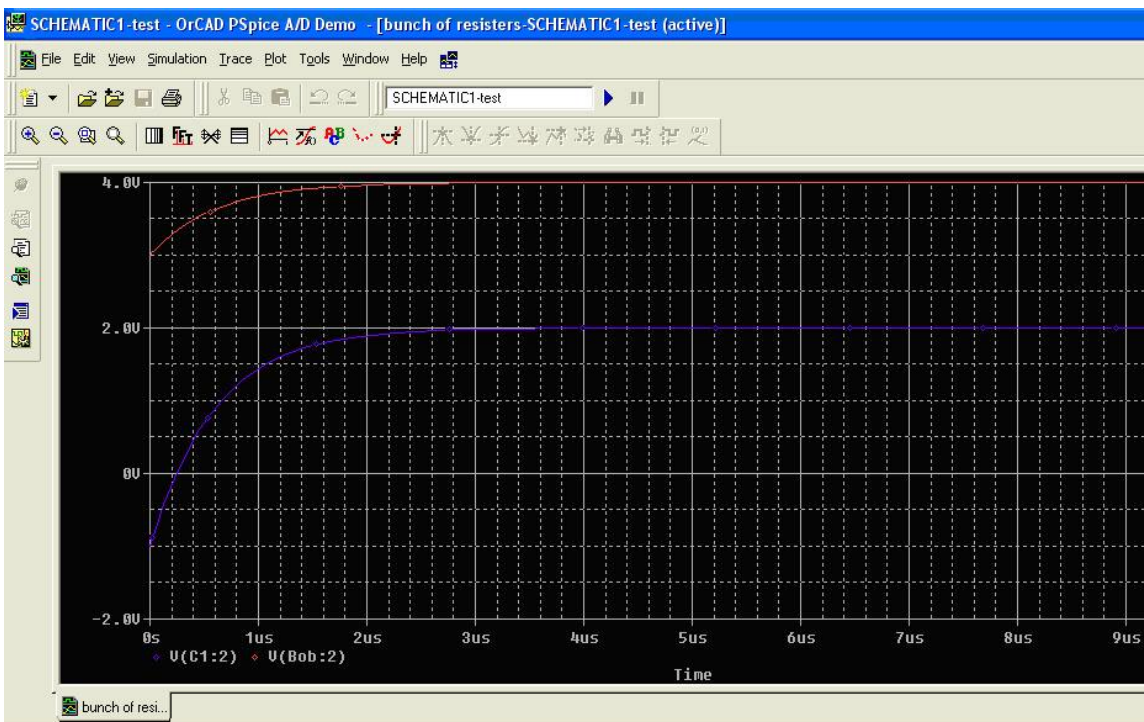


Figure17

You should get Figure17. Notice that although you gave the cap an initial condition of positive 1v, depending on how you orientated the cap, it might be what you would consider $-1v$. If you attempt to copy this picture into word or some other program, it won't always print nicely. Your best bet is to print it directly from here with the “Print” button. If you want the curves to be easily distinguishable when you print them, you can right click on a curve, click properties, and then change the pattern. You can also transfer

these graphs to Excel or Matlab by clicking on the name of the curve you want to copy, press Ctrl-C, then in Excel or Matlab press Ctrl-V. You now have two columns, Time vs. Voltage. Another option is to create a virtual printer in Windows.

- 1) Use the Windows Printer Wizard and choose port "FILE:(print to file)" and printer type "generic MS Publisher Color Printer". This creates a virtual postscript printer that only prints to files.
- 2) In OrCad Capture, do "Print Setup..." and choose this printer.
- 3) Now when you print your schematic it will ask for a file name. Give it the full desired path, and manually append the suffix .ps to the name you choose.
- 4) The file you created can be viewed and converted to other formats using Ghostscript (freely available on the Web).

Parameters/DC Sweeps

Now is your chance to become a Spice-monkey (a Spice-monkey is someone who does not use equations to figure out how to design a circuit, but rather plugs stuff into Spice and uses whatever value works). Delete the capacitor from your schematic. Now change the value of R4 from "1k" to "{r}". The brackets tell Spice that "r" is a variable. Now click on "Place Part" and add library "SPECIAL". Find part "PARAM" and place one on your schematic and the press "E". Double click on the "PARAMETERS" icon you have just placed, and click "New Column". Under name type "r" and under value you can put whatever number you want (I used 1k, but for a sweep it isn't important) (see Figure18). Next click "OK". Then click on the value you just gave "r" (1k in Figure19) and click "Display". Change the display format to "Name and Value" as in Figure19 and click "OK".

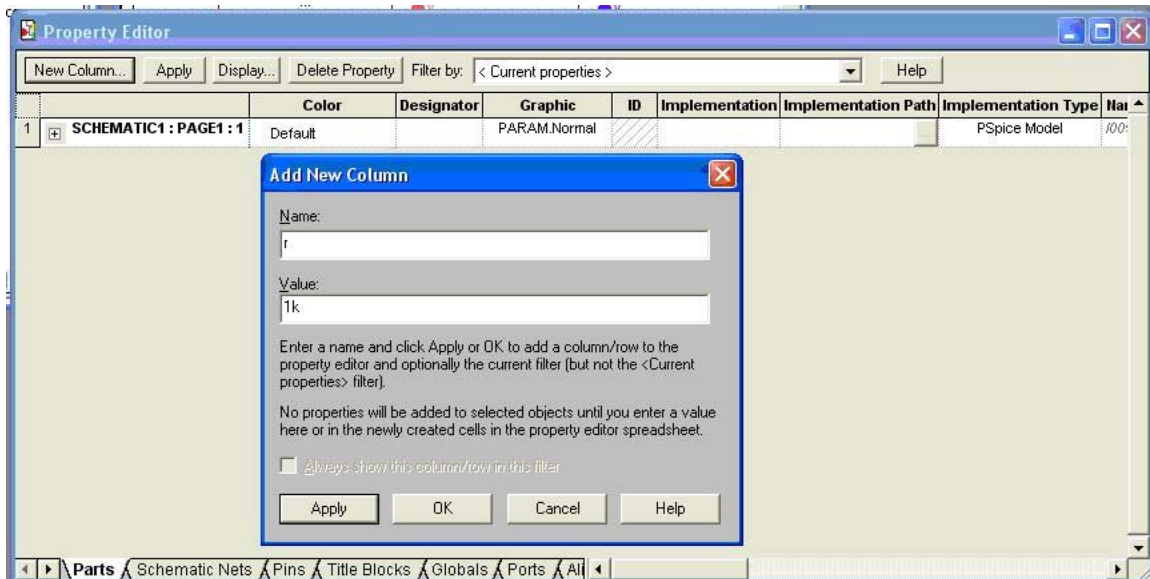


Figure18

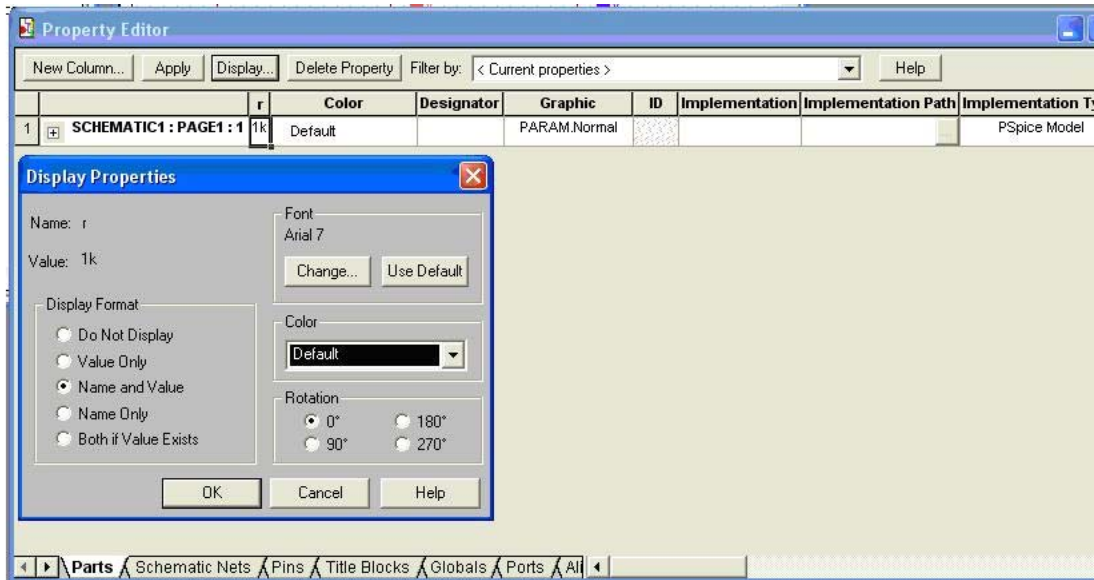


Figure19

You can tell if you did everything right if under “PARAMETERS” (on your schematic) it says “r = 1k” (Figure20). Close the “Property Editor”. Go back to “Edit Simulation Profile” select “DC Sweep” from the “Analysis Type” drop-down menu. Click on “Global Parameter” under “Sweep Variable” and type “r” as the Parameter Name. Choose a start value, end value, and increment. If you want to be really cool, check the box next to “Secondary Sweep” click on Voltage Source, and type in “v1” (the name of the DC voltage source in the schematic) and sweep that also (I did from 1v to 10v with increments of 1v). Run the simulation and you should get curves like Figure21.

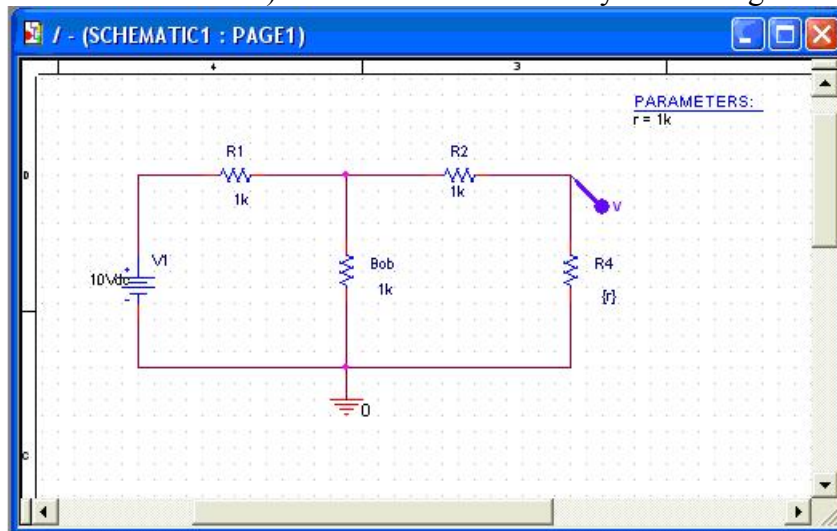


Figure20

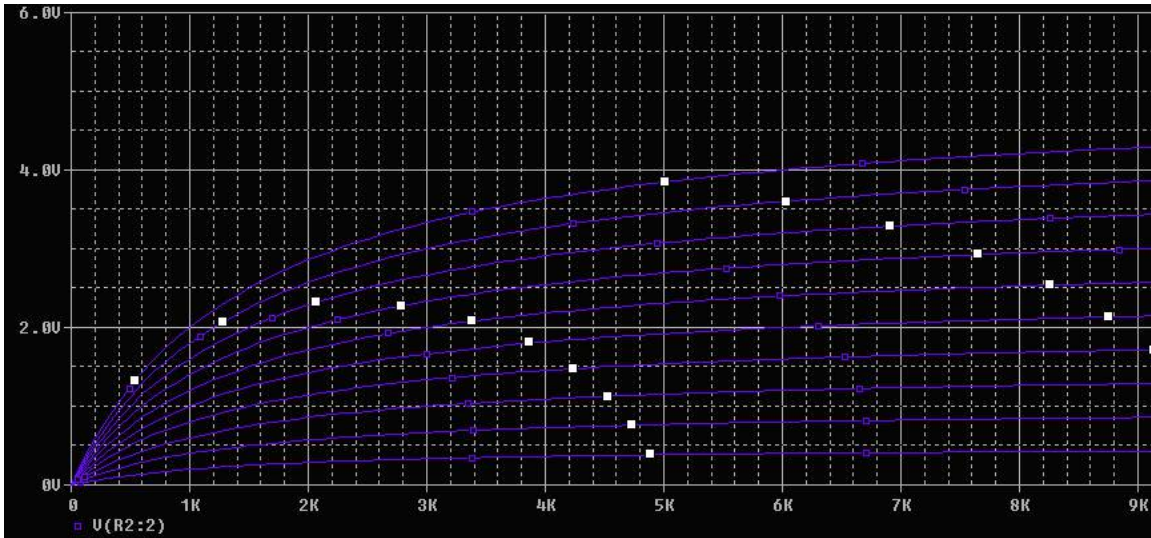


Figure21

You can also do temperature sweeps and current sweeps. When sweeping a source, simply do it the way just shown, but when sweeping a parameter (resistance, capacitance...) make sure to add the parameter as shown.

Controlled Sources

There are four types of controlled sources (all found in the “ANALOG” library).

Name	Pspice Name
Voltage controlled Voltage source	E
Current controlled Current source	F
Voltage controlled Current source	G
Current controlled Current source	H

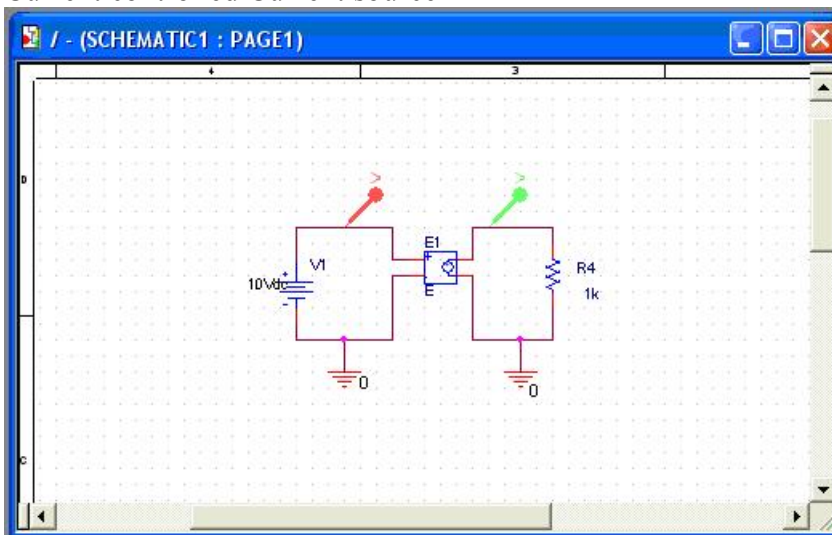


Figure22

Setup a circuit like Figure22. Double click on the VCVS and change the GAIN to 2. Then run a single sweep of the voltage source. The output of the VCVS should be twice the input from the sweep. (If you wanted, you could have left the gain at 1 and used it as a buffer).

AC/Frequency Analysis

Now build the circuit in Figure23. The source is “Vac” in the “Source” library.

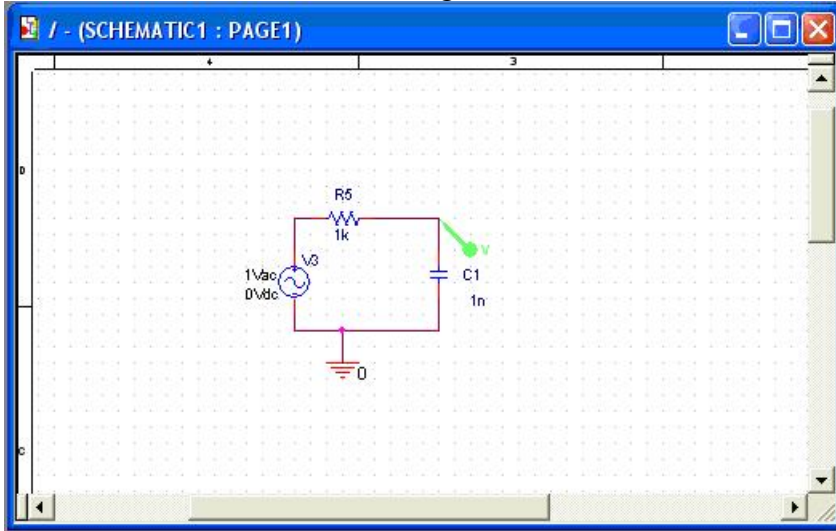


Figure23

It has two values, Vdc which is the DC offset, and Vac, which is the AC voltage. Set those values as in the figure above and place a voltage probe as shown. The ~~Ac~~ AC source does not have a set frequency because it is used to do frequency sweeps. Go to “Edit Simulation Profile” and as your analysis type choose “AC Sweep/Noise”. Choose 1Hz as the start frequency (never put 0Hz in there, it wont work) and 10meg as the end frequency, and use about 1k points. After running the simulation, you should get Figure 24. (If you get something different you might have chosen a logarithmic sweep rather than a linear one – don’t worry, the results are the same, they are just plotted on different scales.)

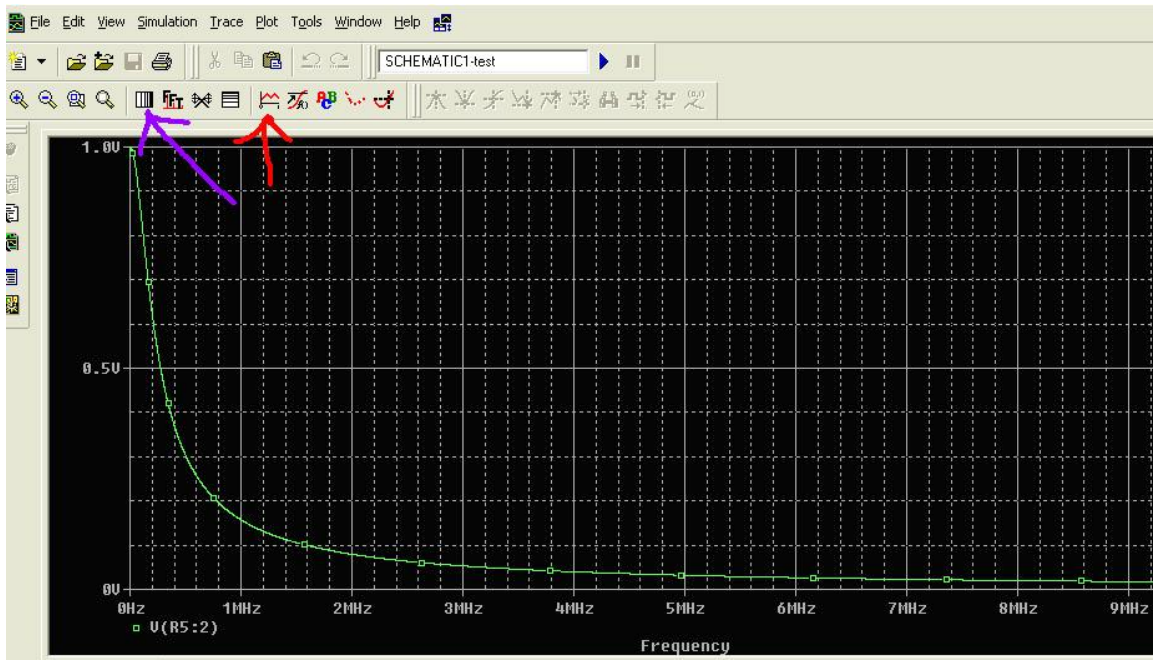


Figure24

Click on the button by the red arrow. A new window will popup. On the left you will see variable names, on the right you will see operators and functions, and on the bottom you will see a box that says “Trace Expressions”. In that box you want to type the formula $DB(V(out)/V(in))$ which will give you the output in decibels (keep reading). For me $V(out)$ was $V(R5:2)$ and $V(in)$ was $V(V3:+)$ so I typed $DB(V(R5:2)/ V(V3:+))$. $V(R5:2)$ means the voltage on the 2nd side of resistor R5, $V(V3:+)$ means the voltage on the positive side of source V3. Try to refer back to your schematic to figure out what is what (you can do that by double-clicking the nodes of interest and noting their names). Once you have your new function graphed, delete the old function by clicking on the name of the old function at the bottom of the graph (such as $V(R5:2)$) and pressing delete. Now click on the button by the purple arrow to change from linear to logarithmic scale. You now have a classic bode plot, which tells you the dB drop for this first order low-pass filter.

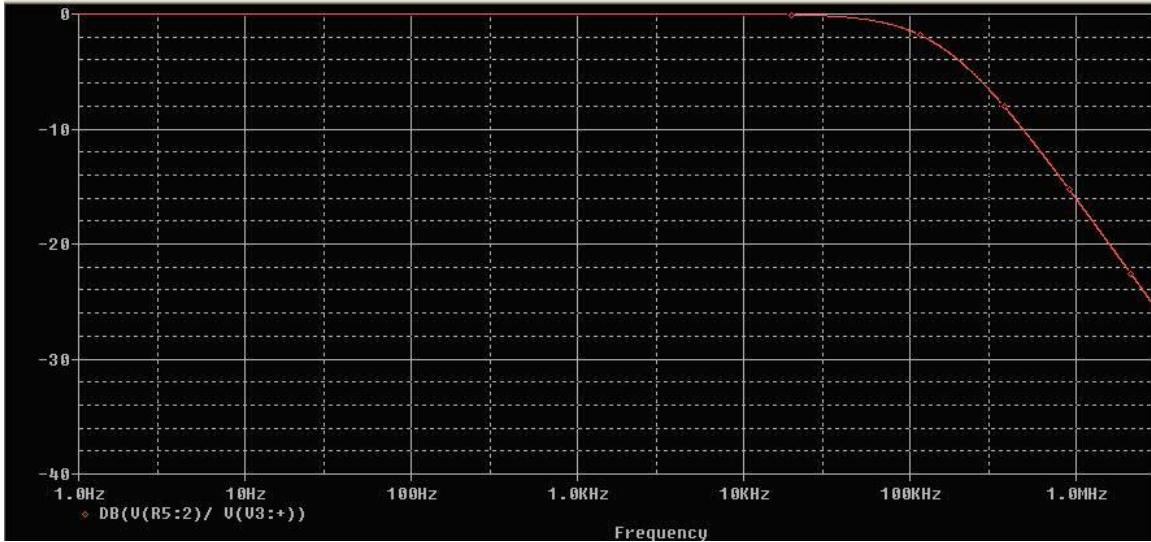


Figure25

AC Sources

There are some cool AC sources in the “SOURCE” library that are used for transient analysis. VSIN outputs a sine wave with a specified frequency, amplitude and DC offset. VPULSE outputs a square or a triangular train of pulses (or something in between as we will explain). VPULSE has a few different settings

V1 = high voltage level

V2 = low voltage level

TD = time delay before VPULSE starts

TR = how long it takes to rise from V2 to V1

TF = how long it takes to fall from V1 to V2

PW = pulse width (i.e. how long the pulse stays at V1)

PER = total period (TF + TR + PW + time spent at V2)

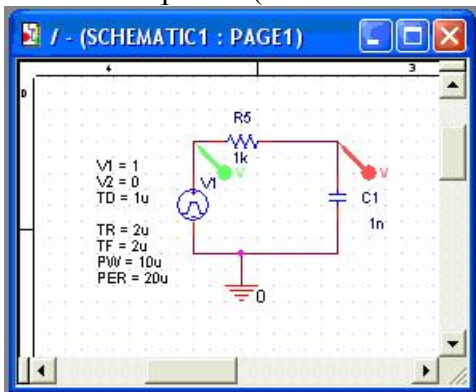


Figure26

Build the circuit in Figure26 and run a transient analysis for 60us (and set the maximum step size to 1us or less). You should get Figure27. If you click on the button by the red arrow, the “Probe Cursor” (i.e. the thing by the blue arrow) will popup. A1 is the

X and Y coordinates of the cursor you control with your left mouse clicker, and A2 is that of the right mouse clicker. Your cursor will only move along points on one of the curves. Currently I am measuring the rise time of the green curve. The time “dif” is about 2u and the voltage “dif” is 1 volt. If you want to have A1 or A2 move along the other curve, simply left or right click the green or red symbol next to that curve’s name at the bottom of the graph.

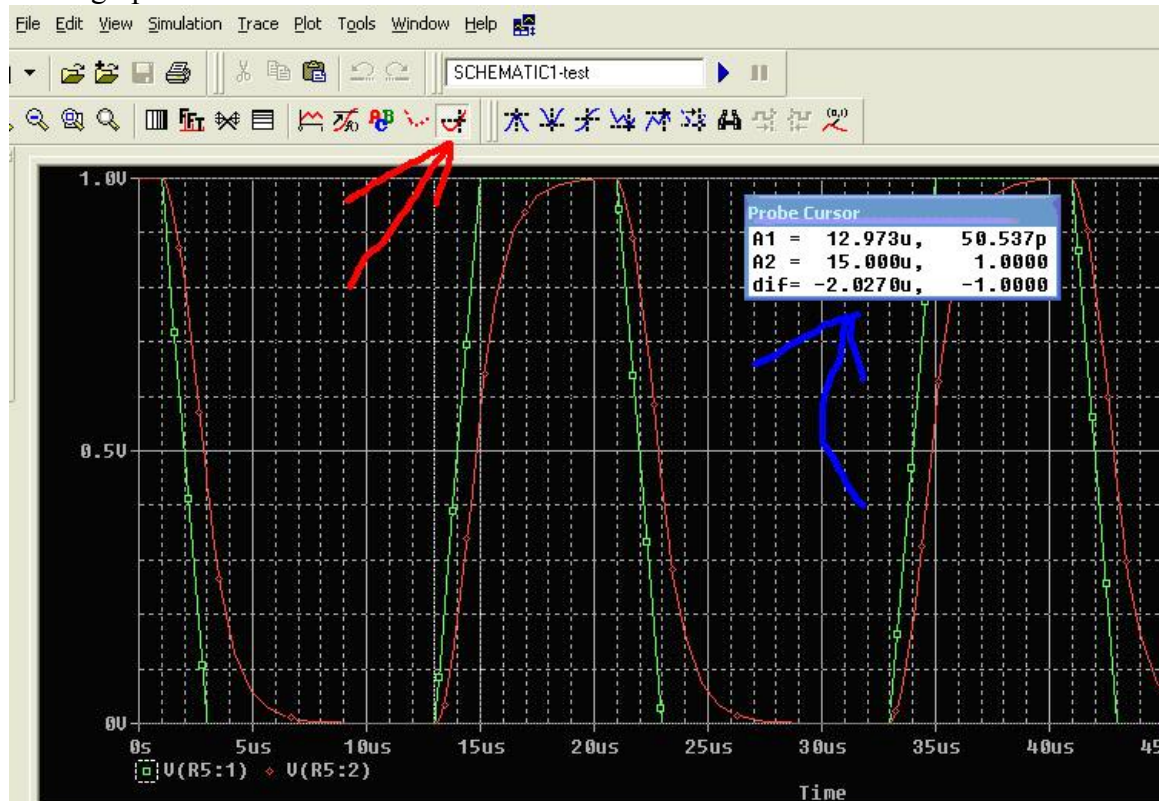


Figure27

Plotting Ideal Functions

You can check simulations against hand calculations by plotting the equations on the same graph as the simulation results. To do this you can (a) place an ideal source on the schematic and plot its output; (b) use “ADD Trace” to add the calculated function, since “Time” is one of the output variables available for manipulation. Each way has a slightly different set of available functions, so between them almost any situation is covered.

Sub-Circuits

Often when building a large circuit, you will first build smaller individual sub-circuits (i.e. amplifiers, filters etc) and then connect them. Usually this is neater and visually easier to understand than building one big circuit. Pspice also allows you to do this. We will now take the schematic we have been using, delete some parts, and convert it into a sub-circuit. We’ll then add a new schematic page to this project, which will

become the main or “top-level” schematic page, and put the sub-circuit and other parts on it. (Instead, we could have added another folder at the level of SCHEMATIC1, copied the sub-circuit elements to it, removed the corresponding elements from SCHEMATIC1, then placed the new sub-circuit on SCHEMATIC1, but the following procedure accomplishes the same thing while illustrating how to re-assign top level status).

Start by deleting the capacitor and probes in Figure26. What is left will be our sub-circuit. OrCad will later create a symbol (a simple box by default) that will represent the circuit, and which can be used on another schematic in place of it; it may even be used multiple times. But to use the sub-circuit, we must define the connection points or “ports”. Click on the “Place Port” symbol (next to the red arrow in Figure28) and find “PORTLEFT-L” and place it as shown in Figure28. Change the port’s name from “PORTLEFT-L” to something like “Vsource”. You can do this by right clicking on it and clicking rename. Next in the explorer window on your left, change the folder name from “SCHEMATIC1” to “source” and change the schematic name from “page1” to “source”. Save, and close the schematic.

For good measure, we will now add another sub-circuit to the project. In your explorer window, right click on “bunch of resistors” (or whatever you named your project) and click on “New Schematic”. Name it something like “network”. Next right click on that folder you just made, click on “New Page” and name it whatever you just named the folder. In the schematic you just made, put down some parts (like resistors and capacitors), and connect them. Add ports. Make sure you have one wire going in on the left and one wire going out on the right like I do in Figure29. Now attach “PORTRIGHT-R” to the left wire, and “PORTLEFT-L” to your right wire, and rename them “in” and “out”. Save and close that schematic.

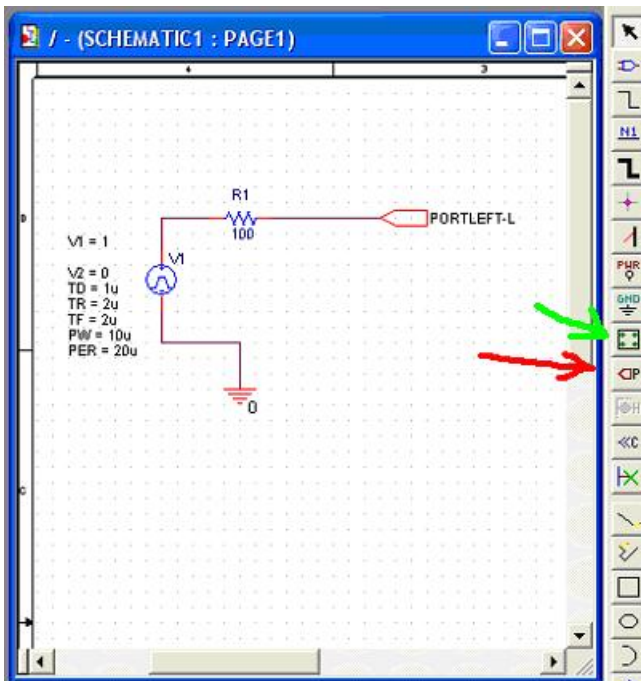


Figure28

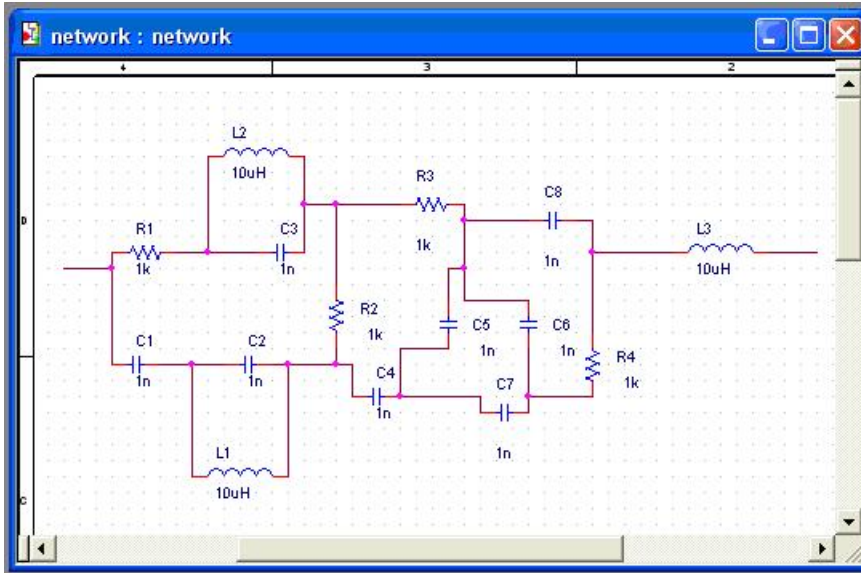


Figure29

It's time to make the top-level schematic, onto which we will place the sub-circuits we've just made. In the explorer window, right-click on your project's name (bunch of resistors) and click on new schematic and name it "main" or something similar. Then right click that folder and create a new page with the same name as that folder. Open up the new page, save it, then right click on the folder "Main" and click on "Make Root". This last step is important, because it tells OrCad that the schematic "main" is the top-level schematic.

Reopen page "Main" and click on the button (Place Hierarchical Part) next to the green arrow in Figure28. In the "Reference" text box, put whatever name you would like to give the sub-circuit when it is placed on the main page. I named it "source1" in the example. Choose "Schematic View" as the "Implementation Type" and under "Implementation name" type the actual name of the sub-circuit created earlier ("source"). Click "Ok". You now have a crosshair, which will be used to define the symbol (a box) for your sub-circuit on the main page. Click and drag your crosshair to make the symbol the size that you desire. You can't resize it once you let go of the mouse button. You should see a box with connection points labeled with the port names you entered earlier. This is your sub-circuit. If you select it (by left clicking on it), you can right click and choose "Descend hierarchy" to look inside the box and see the whole sub-circuit.

Now add the other sub-circuit; this one I named "Network1", and used the sub-circuit "network" as the implementation type. If you want to place more than one, repeat the procedure, but give the new sub-circuit a different name in the "Reference" box. As a short cut, you could copy and paste right from the schematic by clicking on the network you placed and press "ctrl" and "c" to copy, and "ctrl" and "v" to place the part. Just make sure they all have different names. Finally connect the circuit like I have in Figure 30, and run a transient analysis for about 60us.

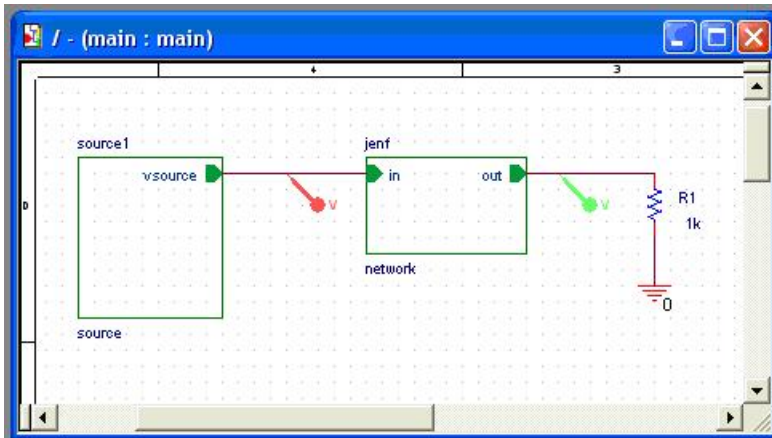


Figure 30

If you make a change in a sub-circuit, make sure to save that sub-circuit before testing the simulation. In some versions of Pspice you will have to use the Synchronize button in the “View” tab at the top of the page. If OrCad or Pspice crashes during these operations and you are having trouble getting everything working again, delete the “Outputs” and “Simulation Profiles” in you explorer window (as shown in Figure 31).

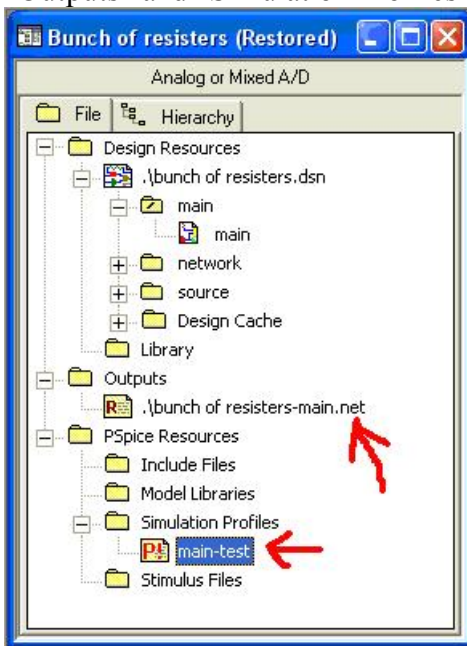


Figure 31

Finally, it is possible to create “parts”, which are sub-circuits with arbitrarily fancy symbols and which can be used in any project, just like the parts that came with OrCad. We leave this as a challenge for the student.

Transistors

OrCad provides the Q2N2222 NPN and Q2N2907A PNP in the "Eval" library. The only MOSFETs provided are the IRF150 and the IRF9140. These are power

MOSFETs and they won't work as fast as the MOSFET you used in your lab. We need to alter the models. To do this first find the folder that contains your libraries. For me it was C:\Program Files\OrCad_Demo\Capture\Library\Pspice. To find this folder easily, go to the "Place Part" in Pspice, and click add library, and click on the drop down menu by "Look in" to figure out the correct path to the "Eval" library. Once you get to that folder, you should see two files named "Eval". Make a copy of both. Now go to the Start menu and click on "Pspice Model Editor Student". Open the copy of "Eval" you just made. Find "IRF150" and double click on it. Comments next to the asterisks are not read, so don't worry about that stuff. Scroll to the bottom and change the bottom to the parameters to be the same as those in Figure 32.

```
* The factors not modeled include:
*   - maximum ratings (eg. high-voltage breakdown),
*   - safe operating area (eg. power dissipation),
*   - latch-up,
*   - noise.
*
* For high-current switching applications, we advise that you include
* series inductance elements, for the source and drain, in your circuit file.
* In doing so, voltage spikes due to di/dt will be modeled. According to the
* 1985 International Rectifier databook, the following case styles have lead
* inductance values of:
*   TO-204 (modified TO-3)  source = 12.5nH      drain = 5.0nH
*   TO-220                  source = 7.5nH       drain = 3.5-4.5nH
* -----
*
*.model m14007n NMOS(Level=1 vto=1.77 Kp=4e-4 gamma=4.1 phi=.65 lambda=1.5e-2
+      is=1e-15 pb=.87
+      tox=1.265e-10
+      rs=3 rd=3)
*      Int'l Rectifier pid=IRFC150      case=TO3
*      88-08-25 bam      creation
```

Figure 32

Save the library and restart Pspice. Now the "IRF150" will act like an m14007n. If you don't want to do all that, you could go on Google and try to find a library that already has the transistor you need. One site that has a few libraries available is <http://www.OrCad.com/community.pspice.models.aspx>

Music Inputs

While some students might be satisfied using standard sine waves as inputs to their circuits, other might be interested in running a real music file in Pspice. This section will describe how to take a .wav file, convert it into a file that Pspice can read, and then add it into Pspice, then convert the output of the simulation into a .wav for listening.

We will be using a program called Sox to do most of this. To get a copy of sox go to <http://sox.sourceforge.net/> to the section called "release information". If you have Windows get the one called "Win32 Binary (Win95/98/NT/XP)". Unzip Sox into a folder on the C drive called sox (meaning the stuff should be put in C:\sox .If for some reason

your hard drive is not the C drive but another letter, replace the C in C:\ with that letter). Sox is run from a text line (i.e. Dos). To get to a text line in Windows, click on the start menu; go to All Programs, Accessories, and then Command Prompt. A black screen with letters will popup. Type in “cd c:\sox” then press enter. Type “dir” to see what is in the Sox folder. There should be six files in there.

Now place a .wav file in the Sox folder (you can do this the way you normally would through Windows). (If your music sample is stereo, convert it to mono for use in spice by typing “sox filename.wav -c 1 filename_mono.wav avg”). Type “sox filename.wav filename.dat”. (Filename is the name of your wav, if your file is named bob, then type bob.wav and bob.dat). You have just created a file. (If Sox is giving you trouble, insert “-V” between “sox” and “filename.wav”, this will show any errors Sox encounters).

Before we use the file we just created we have to modify it a little. Open the .dat with WordPad (or notepad if it is small). At the top you will see something like
; Sample Rate 22050
; Channels 1
followed by two columns. Cut this header out. The left column is time and the right column is the corresponding voltage. Save and exit WordPad. Open Pspice and in the source library find “VPWL_FILE”. Double click on the “<FILE>” and type in the location of your .dat file. Continue by following the listed instructions below.

1) Place a voltage marker on the schematic to select the output variable you wish to hear.
2) Edit simulation profile (“data collection” section) to save only marked variables; this limits the file size to something reasonable.
3) Create a text file in OrCad sim directory (maybe called user_options.inc) and put in this text:

~~options numdgt=10~~

options numdgt=10

4) Edit the simulation profile (“include files” section) to add the file from step (3)
5) Place a “vprint1” symbol on schematic, attached to the node you wish to hear; edit its properties to enable printing of transient analysis results by placing a Y in the TRAN column. **If there is no "TRAN" column, click "New Column..." and add it, with Name="TRAN" and Value="Y".**

6) Edit the simulation profile (“Analysis” section, “Output File Options...” subsection) to “Print values in the output file every: ” 22.675737u seconds. This is 1/44 kHz, the standard .wav sample rate.

7) After running the simulation, edit the .out file in your sim directory to remove everything up to the two columns of numbers (the time and the output variable). Remove anything after the two columns as well.

8) Add these two lines before the columns:

; Sample Rate 44100

; Channels 1

9) Save file as whatever.dat, or as whataever.txt then rename to whatever.dat

10) Convert to wav using the command:

“sox whatever.dat whatever.wav”

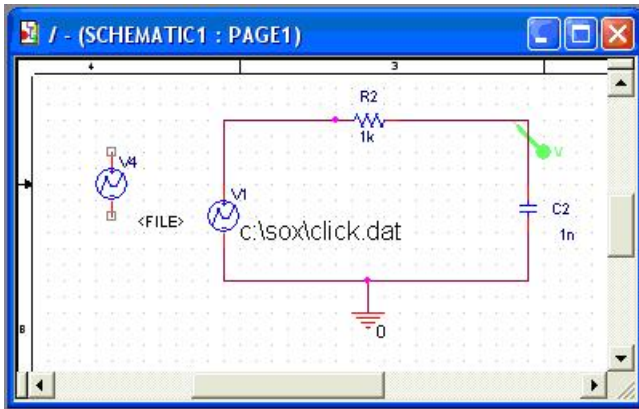


Figure33

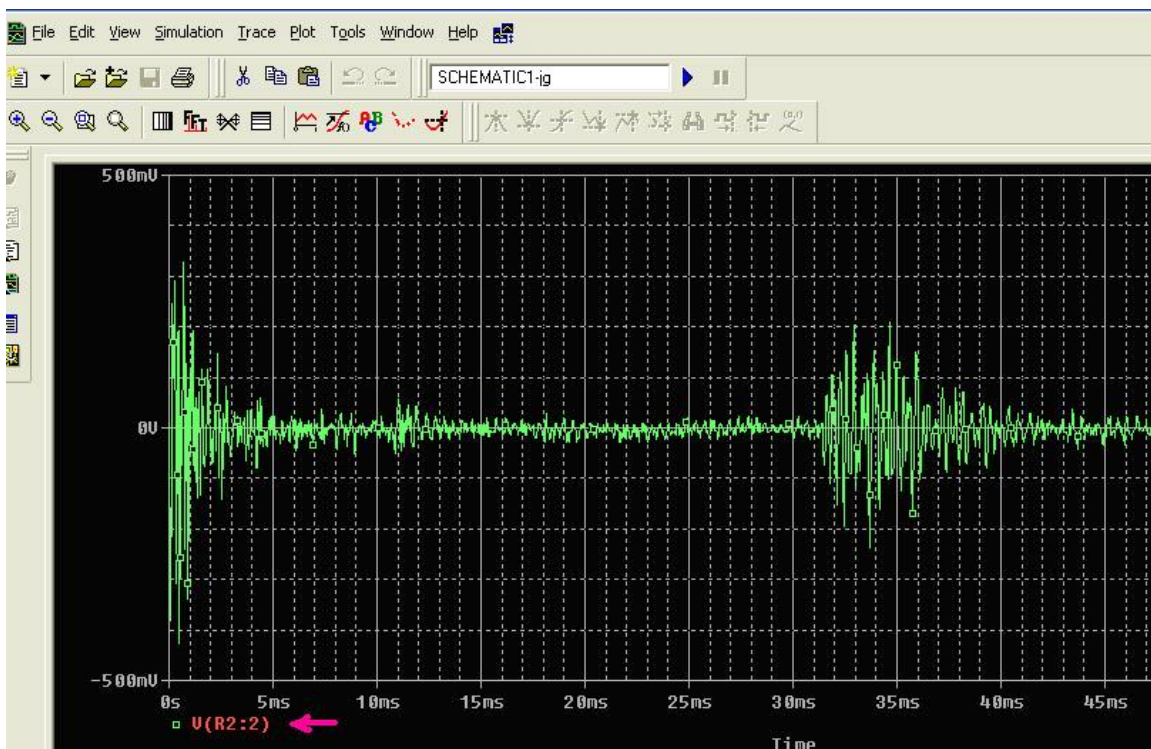


Figure34

Behavioral Sources

We saw some simple controlled sources earlier in the analog library. For user defined controlled sources we will have to use the ABM library. Find the single input single output VCVS named ABM1. You have the ability to write equations using the following operators

Operator	Translation
+	Addition
-	subtraction
*	Multiplication
/	Division
sin(x)	Sine
cos(x)	Cosine
tan(x)	Tangent
atan(x)	Arc (inverse) tangent
exp(x)	Exponential (e^x)
log(x)	Natural logarithm ($\ln(x)$)
sinh(x)	Hyperbolic sine
cosh(x)	Hyperbolic cosine
tanh(x)	Hyperbolic tangent
sqrt(x)	Square root
log10(x)	Logarithm to the base 10
db(x)	Decibel ($20\log_{10}(x)$)
abs(x)	Absolute value
min(x1,x2)	Smallest between x1 and x2
max(x1,x2)	Largest between x1 and x2

On a one input device you variable is V(%IN). If you are having trouble with your device (which I did for all the voltage devices) double click on the device to open the Property editor. Look under the column name Part Reference. If the name has any question marks, delete them.

Revision History

rev A: Sept 26, 2007 (by DV)

- correction to step 3 of "Using Music Inputs" (numdgt)
- one or two typos corrected. "Curser" (sic) still incorrectly spelled
- added instructions for vprint1 Property Editing to add "TRAN"