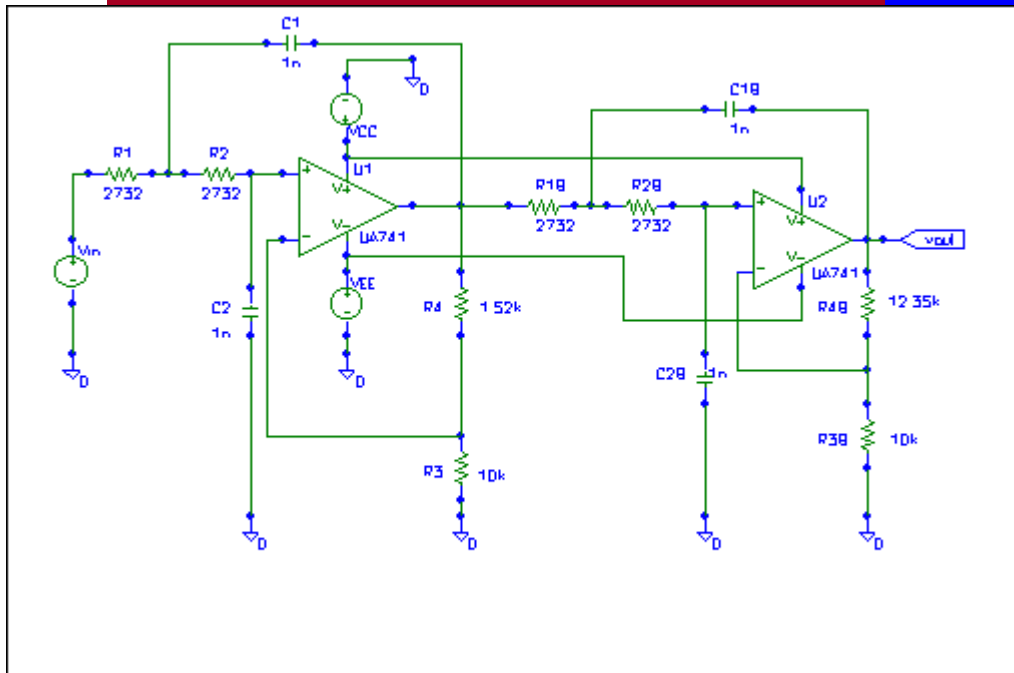


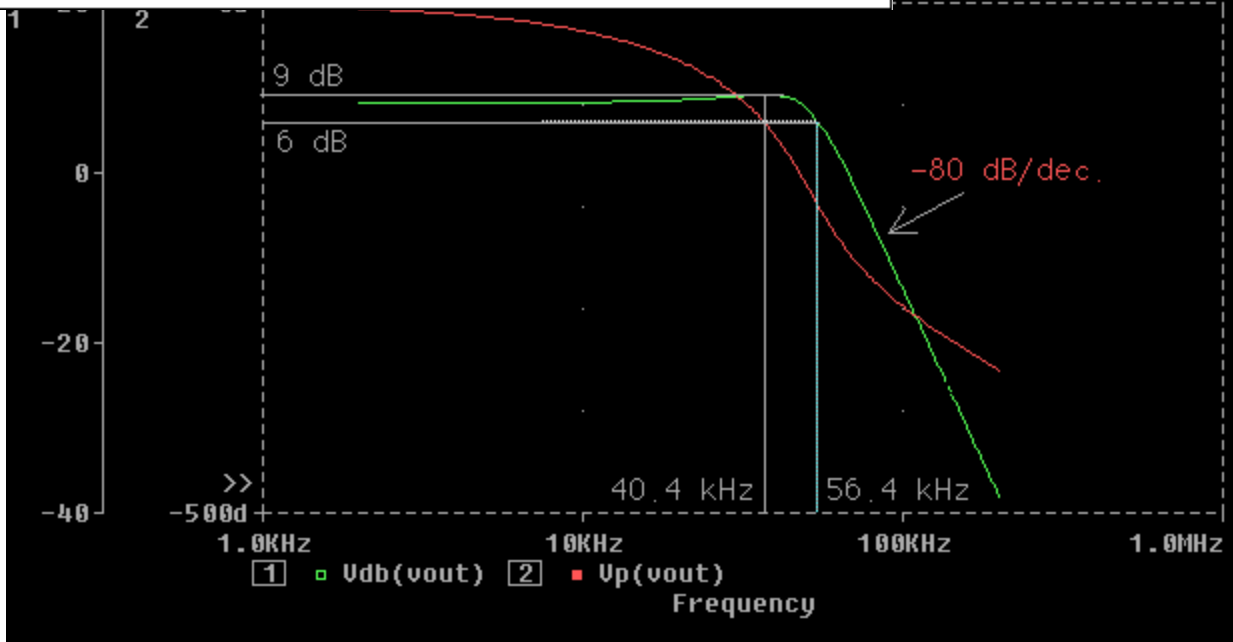
A Tutorial for Schematics

- the PSpice Schematic Capture Utility



* Schematics Netlist

```
X_U1  $N_0002 $N_0003
$N_0004 $N_0005 $N_0001
UA741
V_VCC  $N_0004 0 dc 10
V_VEE  $N_0005 0 dc -10
V_Vin  $N_0006 0 dc 0 ac 1
R_R1  $N_0006 $N_0007 2732
R_R2  $N_0007 $N_0002 2732
R_R4  $N_0003 $N_0001 1.52k
R_R3  0 $N_0003 10k
C_C1  $N_0007 $N_0001 1n
C_C2  0 $N_0002 1n
X_U2  $N_0008 $N_0009
$N_0004 $N_0005 vout UA741
R_R1B  $N_0001 $N_0010 2732
R_R2B  $N_0010 $N_0008
2732
```



Written by W.C. Dillard
Auburn University

Outline/Contents

PART I. PSPICE FUNDAMENTALS

I.	Introductory Remarks	3
	A. Spice, PSpice and <i>Schematics</i>	
	B. Font Conventions	
	C. The Passive Sign Convention and PSpice	
II.	What is <i>Schematics</i> and what can it do for you?	4
III.	Units and Unit Prefixes	4

PART II. CONSTRUCTING AND SIMULATING A DC CIRCUIT

I.	Drawing Circuit Diagrams with <i>Schematics</i>	5
	A. Getting Parts	6
	B. Changing a Part's Name and Attributes	7
	C. Arranging Parts and Pin Numbers	8
	D. Wiring Parts	8
	E. The Analog Ground	9
	F. Saving Schematics	9
II.	Getting Results	10
	A. The Netlist, Arrgh!	
	B. Naming Nodes	10
	C. Displaying Results on the Schematic	11
	D. Printing your Drawing	11
III.	Running PSpice from <i>Schematics</i> and Choosing an Analysis Type	13
	A. Obtaining a Bias Point Detail	14
	B. Obtaining a dc Sweep	15
IV.	Running and Using PROBE	16
	A. Adding Traces	
	B. Using Markers	
	C. Other PROBE Features - Label, Cursor, Copying and Saving	19
V.	Printing your Drawing	12
VI.	Your First Schematic	20

PART III. CONSTRUCTING AND SIMULATING AN AC CIRCUIT

I.	Obtaining an ac Analysis and making Bode Plots	21
	A. ac Sweeps	21
	B. Making Bode Plots	24

PART IV. CONSTRUCTING AND SIMULATING A TRANSIENT CIRCUIT

I.	Constructing and Simulating a Transient Circuit.....	25
II.	Setting Initial Conditions	27

PART V. CONCLUSIONS

PART I. PSPICE FUNDAMENTALS

I. Introductory Remarks

A. Spice, PSpice and Schematics

Spice (Simulation Program with Integrated Circuit Emphasis) was developed at the University of California at Berkeley. It quickly became an industry standard for simulating integrated circuits and is still distributed at low cost (comparatively) to government facilities such as universities. As the electronics industry advanced, several companies began to sell PC and Macintosh compatible versions of Spice. One such company, ORCAD Corporation, produces a PC compatible version called PSpice. In this tutorial, we are using the **Student Version 9.1**, which is free from ORCAD via either CD or download. (Go to ORCAD's website to request the CD or download the software directly.)

In Spice, circuit information such as the names and values of resistors and sources and how they were connected is input using data statements with a specific format. Particulars for every element must be typed in the exact proper order. This makes debugging difficult since you have to know the proper format to recognize the mistakes you might make in formatting! An advancement in PSpice is a program, called *Schematics*, which bypasses the cryptic formatting, allowing you to draw the circuit diagram and assign element values via user-friendly dialog boxes. ORCAD packages *Schematics* in a 'software suite' that includes PSPICE (the simulation engine) and PROBE (a plotting utility). Learning to use *Schematics* and its support programs is what this tutorial is all about

B. Font Conventions

Since we'll be discussing text you might need to type into dialog boxes within *Schematics* as well as text the program itself will return to the screen, the following fonts are adopted. **All text you type in will in this font.** *All text the program returns will be in this font.* Finally, all the text contained in dialog boxes and menus will be in this font.

C. The Passive Sign Convention and PSpice

All currents and voltages in PSpice and *Schematics* obey the passive sign convention shown in Figure 1. The voltage across the element is defined positive at node m with respect to node n . Obviously, the ordering of the nodes is quite important. In fact, we'll return to this concept of node m versus node n several times. If V_{mn} as calculated by PSpice is positive, then PSpice will return a positive number. If the current value returned by PSpice is positive, then current flows in at node m and out of node n . For example, if we ask PSpice its calculations for V_{mn} and I and it said, -4.5 and $2.2E-3$, we know that node n is 4.5 V positive with respect to node m and a current of 2.2 mA flows from m to n . In other words, the element is a source.

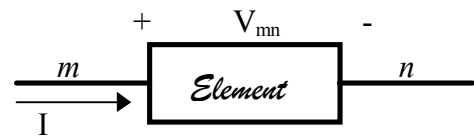


FIGURE 1

II. What is *Schematics* and what can it do for you?

As mentioned in section I, *Schematics* is a program that lets you simulate circuitry using PSpice algorithms without using the Spice data statement formats. For example, consider the case of creating the simple dc voltage source

$$v(t) = 5.5 \text{ Volts}$$

In Spice you create the source by typing the line

VIN 3 7 dc 5.5

Where VIN is the source's name (you make that up), 3 and 7 are the node numbers where the source is connected (which you also make up). Of course, you must remember that the first node, 3, is assumed positive with respect to the second node, 7. The remainder of the statement identifies the source type as dc with a value of 5.5 V. That's a lot to get right and in the right order! In *Schematics*, you just draw the voltage source. A dialog box is used to specify all the above-mentioned attributes by name rather than order of entry. Also, *Schematics* will automatically make up the node numbers for you!

When *Schematics* and PSpice are coupled with the PROBE utility, the total package becomes very impressive. PROBE makes plots. Plots typically have time, frequency, or dc voltage or current for the x-axis variable. The y-axis can be any simple mathematical function of any of the voltages or currents in the circuit under simulation. Also, PROBE has several built-in functions such as taking the magnitude of a voltage or current, finding its dB equivalent or phase!

The evaluation version of *Schematics* is a powerful simulation suite. However, it does have limitations. Circuits can contain no more than 10 transistors, 2 operational amplifiers, and 64 nodes. Also, schematic diagrams are limited to one page.

III. Units and Unit Prefixes

When elements such as resistors and voltage sources are given values, it is convenient to use unit prefixes. PSpice supports the prefixes listed below. Note that the letter must immediately follow the value - no spaces. Also, any text can follow the prefix letter. Finally, PSpice is case insensitive. So, there is no difference between 1 Mohm and 1 mohm in PSpice.

PSpice Unit Prefixes

T -	tera	-	10^{12}	K -	kilo	-	10^3	N -	nano	-	10^{-9}
G -	giga	-	10^9	M -	milli	-	10^{-3}	P -	pico	-	10^{-12}
MEG -	mega	-	10^6	U -	micro	-	10^{-6}	F -	femto	-	10^{-15}

PART II. CONSTRUCTING AND SIMULATING A DC CIRCUIT

I. Drawing Circuit Diagrams with *Schematics*.

Let's draw the simple dc circuit in Figure 2 using *Schematics*. The initial step is to open *Schematics* using the **Start/Programs/Pspice Student/Schematics** sequence of pop-up menus. When *Schematics* opens, your screen will change to the Schematic Editor window in Figure 3. Notice that the window bar across the top informs us that

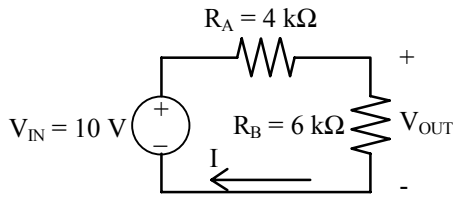


FIGURE 2

this a new schematic. Also, note the Help menu -- it'll get some use. A quick look under the Draw menu, Figure 4, shows the options we'll need to create our schematic -- namely, Get New Part and Wire. Let's get started.

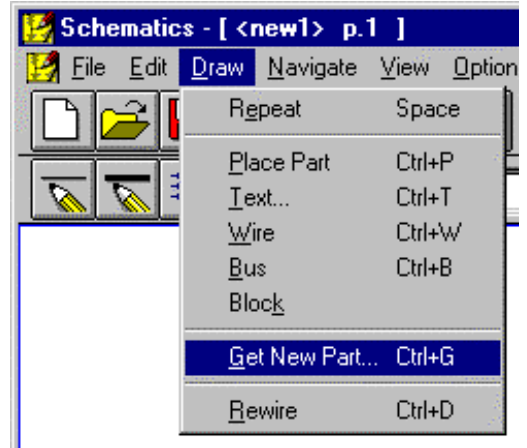


FIGURE 4

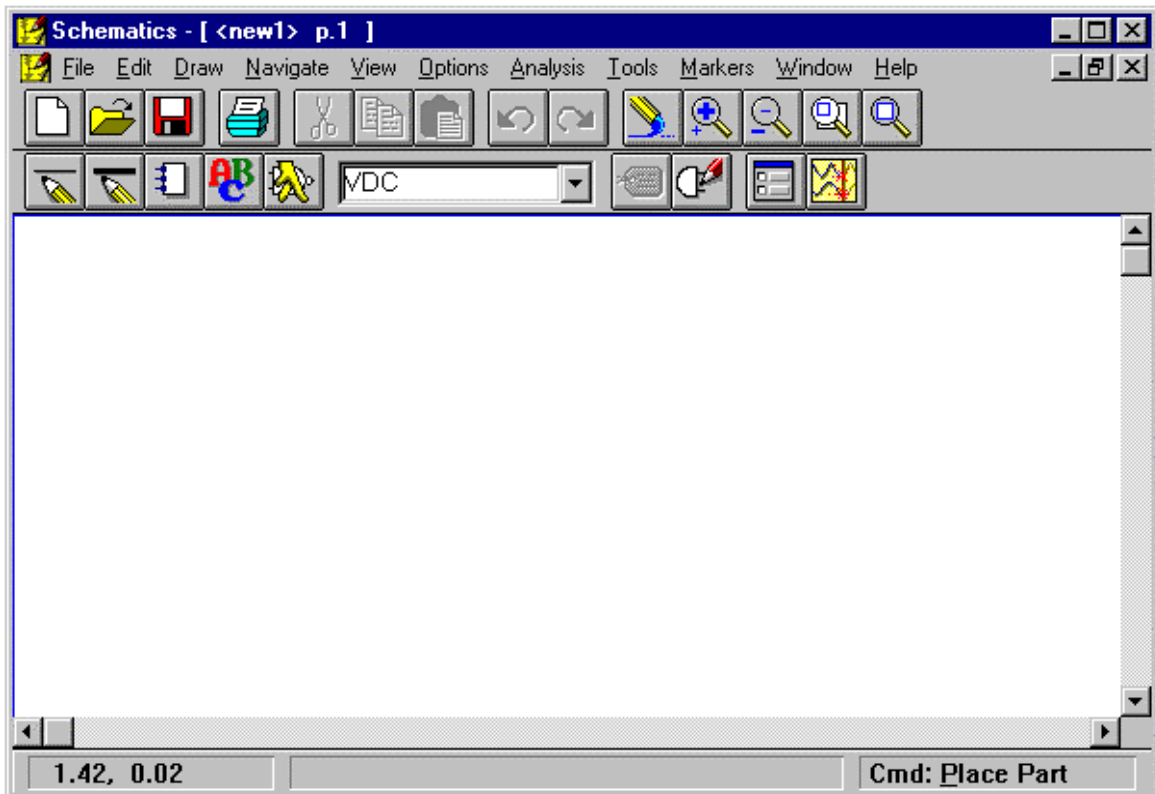


FIGURE 3

A. Getting Parts

First, let's get the voltage source. This is done by left clicking the **D**raw menu and selecting **G**et New Part. The **P**art **B**rowser dialog box in Figure 5 appears listing all the parts in the evaluation version. Since we don't know the *Schematics* name for a dc voltage source, we select **L**ibraries, and the **L**ibrary **B**rowser window in Figure 6 appears listing the various libraries the parts are divided into. For example, resistors, capacitors and inductors are in the ANALOG.slb library. Integrated circuit parts in the EVAL.slb library. Of interest to us is the SOURCE.slb library, which has already been selected in Figure 6. Select the part VDC and OK. The box in Figure 5 should reappear. Select **P**lace & **C**lose and

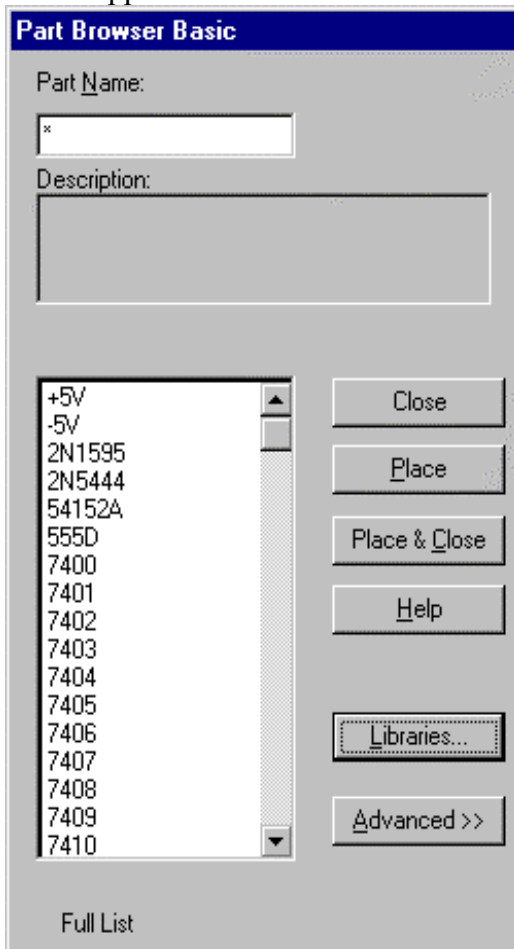


FIGURE 5

you're back to the schematic page of Figure 3 with one difference: the mouse pointer is now a dc voltage source! Just point it any place inside the drawing area and left click **ONCE**. The result is a voltage source in our schematic. Since we need only one source, we click the right mouse button to stop placing sources and the mouse pointer returns to its original status as shown in Figure 7.

Notice that *Schematics* calls the source V1. If we had continued to left click, we would have created V2, V3, etc.. Also, note the grid dots. *Schematics* always aligns whatever you draw to the grid dots even if you place the part between dots. Finally, the two 'leads' extending from the source will be used to make wiring connections to the resistors R_A and R_B . This, 'real world feel' is one feature that makes *Schematics* so popular.

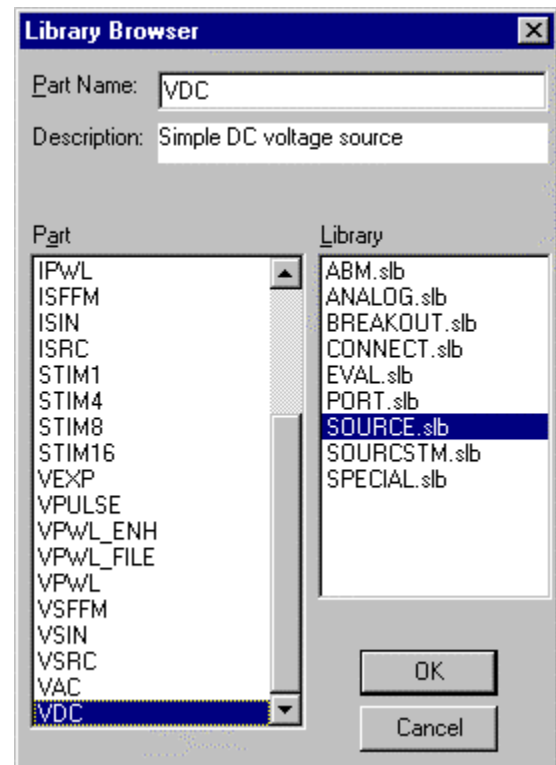


FIGURE 6

Now we'll place the resistors. We repeat the process of Draw/Get New Part except this time, when the Part Browser Basic dialog box appears; we type in **R** and select OK. Now the mouse pointer is a resistor. We left click once to place R1, MOVE THE MOUSE SOME DISTANCE, and left click to place R2. To stop adding resistors, click

once. Moving the mouse between part placements keeps them from stacking atop one another in the diagram. Figure 8 shows the present condition. Although you may not be able to see it in this black and white print, R2 is red meaning it is still selected and may be edited. Note that the resistors are automatically assigned values of 1 k Ω .

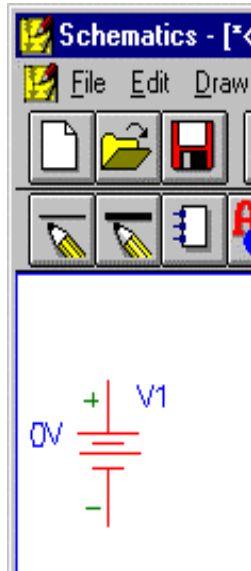


FIGURE 7

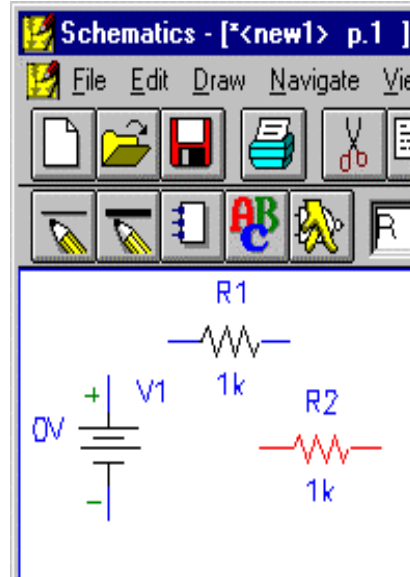


FIGURE 8

B. Changing a Part's Attributes

To change the voltage source's attributes (its name and value), simply double click on the name V1 and the Edit Reference Designator window in Figure 9 appears.

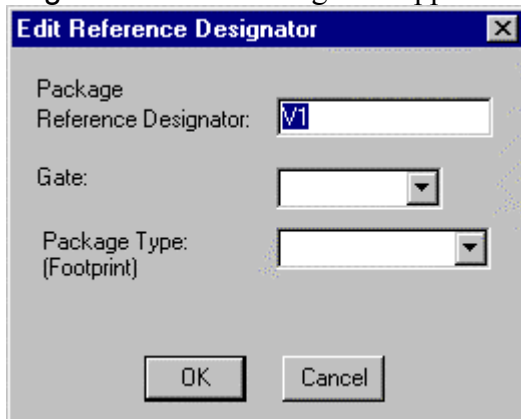


Figure 9

Type in the new name, in this case **Vin**, and select OK. To change the source's value, double click on the value in Figure 9, to open the Attributes window in Figure 10. Enter **10V** and select OK. Do likewise for the resistors, renaming them Ra and Rb and setting their values at 4 k Ω and 6 k Ω respectively. Next, we'll arrange the parts for wiring.

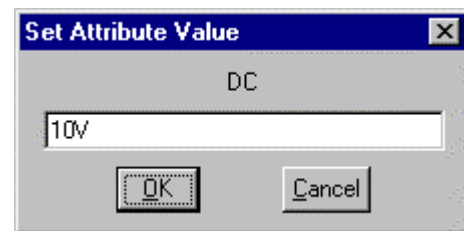


FIGURE 10

C. Arranging Parts and Pin Numbers

Every pin (external connection) on every part in every *Schematics* libraries is numbered. When arranging parts in *Schematics*, it is vital to understand the relationship between the passive sign convention in Figure 1, current directions in PSpice and the part's pin numbers. When a part is placed, it is automatically oriented either vertically or horizontally. As shown in Figure 11, vertically placed parts have pin 1 at the top and pin 2 at the bottom, while horizontally oriented parts have pin 1 on the left and pin 2 on the right.

Currents measured by PSpice flow into pin 1 and out of pin 2.

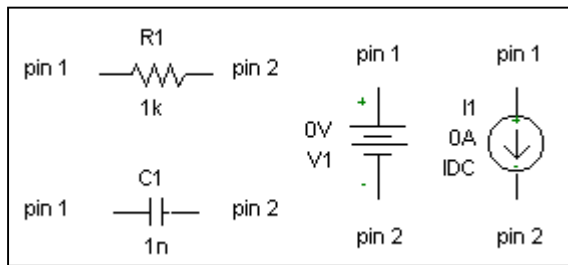


FIGURE 10

Now consider R2 in our circuit in Figure 8. We need to rotate it for appearance sake. Also, the current of interest, I , flows downward through Rb in Figure 2. Selecting Rotate from the Edit menu (or CNTL-R) will spin Rb 90° counterclockwise, putting pin 1 at the bottom. Should we ask for the current in Rb, we will get the negative of I . It is better to rotate Rb 270°, leaving pin 1 at the top. Now the current in Rb as PSpice sees it is the same as I .

Of course, a part can be dragged to a new position. To make our schematic match Figure 2, we'll rotate R2 three times (270° total) and drag it at the right end of the

diagram. A diagram similar to Figure 11 should result.

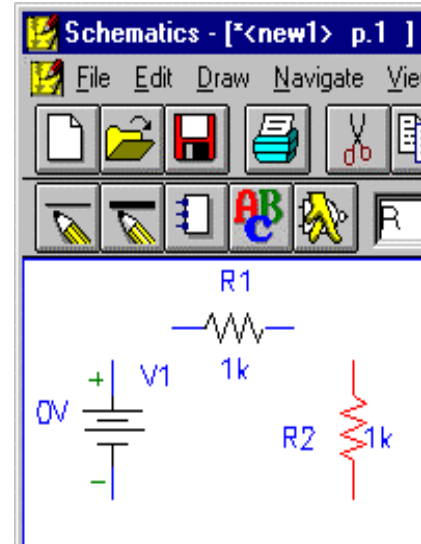


FIGURE 11

D. Wiring Parts

To connect the parts, go to the Draw menu and select Wire. Now the mouse pointer turns into a symbolic pencil. To connect the top of the source to Ra, use the mouse to place the pencil at the end of the wire stub protruding from Vin, click once and release. Next, move the mouse up and over to the left end of Ra. The wire is drawn up and over at 90° angles as the dashed lines in Figure 12. DASHED LINES ARE NOT

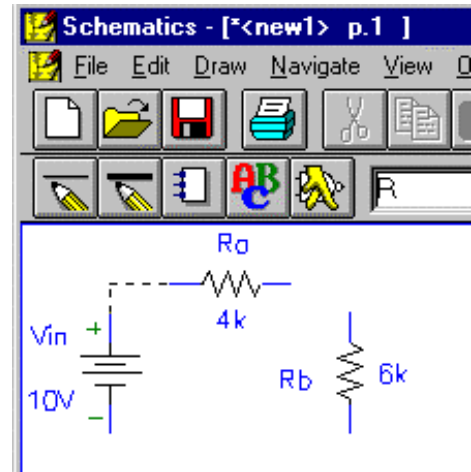


FIGURE 12

YET WIRES! Left click once to complete and 'cut' the connection. The dashed lines become solid and the connection is made as shown in Figure 13. Any left over wire fragments (dashed lines) can be removed by selecting **Redraw** from the **Draw** menu. As for drawing the two remaining wires, here's a *Schematics* shortcut. To reactivate the wiring pencil, if you need to, double right click. This shortcut reactivates whatever the most recent mouse use happened to be. Simply repeat the steps listed above to complete the wiring. Figure 14 shows the wired schematic. A note of caution -- be patient when wiring. This is the area of *Schematics* where you're most likely to err, so be careful.

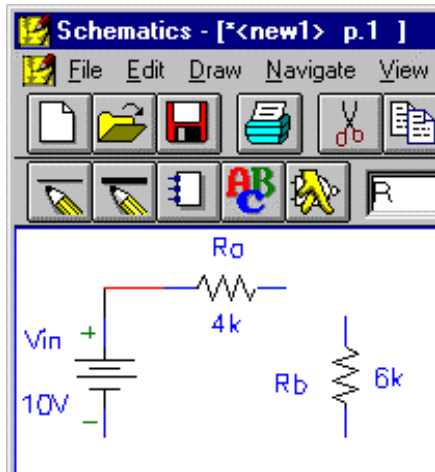


FIGURE 13

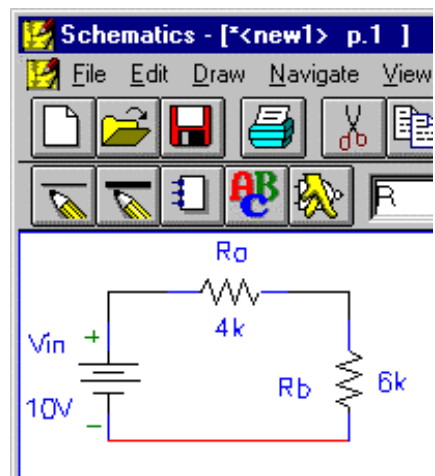


FIGURE 14

E. The Ground

PSpice requires that all schematics have a ground, or reference terminal. The voltage there will be zero and all other node voltages are referenced to it. The part you need is either the analog ground (AGND) or the earth ground (EGND) in the PORT.slb library. In this example, we'll get the AGND part and place it at the bottom of the schematic as shown in Figure 15. If the connection dot doesn't appear in your schematic, simply select the AGND part and move until it touches the bottom wire in the diagram. Your schematic is finished and ready for saving.

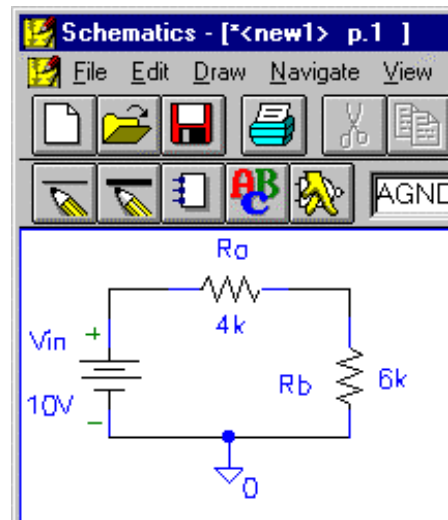


FIGURE 15

F. Saving Schematics

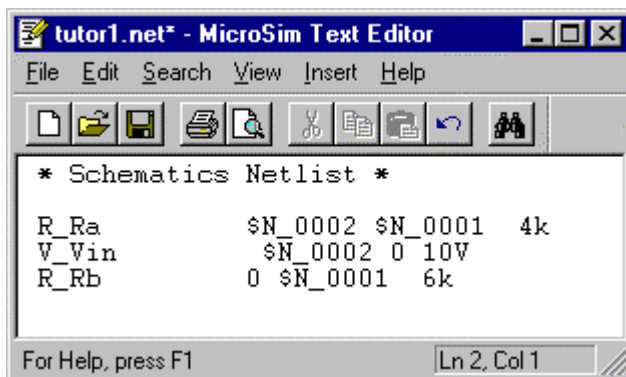
Simply go to **File** menu and select **Save**. PSpice will try to save your schematics in the My Documents directory. It's much better to save your work in a different directory. Schematic files are given the extension `.sch`.

II. Getting Results

A. The Netlist, Arrgh!!

The netlist is the old-fashioned Spice code listing for the circuit you've drawn in *Schematics*. To create the netlist, go to the **Analysis** menu and select **Create Netlist**. At the bottom of the Schematic Editor window, is the either message *Netlist Created* or a dialog box appears informing you of netlist errors. Assuming your netlist is fine; you can view it by returning to the **Analysis** menu and selecting **Examine Netlist**. The Windows utility **Notepad** opens the netlist file shown in Figure 16. Notice that all three of our elements are here along with the proper values. The text \$N_0001 \$N_0002 etc., are the node numbers that *Schematics* made up when it converted our diagram to the netlist. Those of you familiar with Spice will recognize each line of the netlist as nothing more than the proper data statements for dc voltage sources and resistors.

By tracing our way through the node numbers we can assure ourselves that our circuit is properly connected. The source is 10 volts positive at node 2 and with respect to 0. AGND is always node number zero. Ra connects 2 to 1 and Rb finished the loop



```
* Schematics Netlist *
R_Ra      $N_0002 $N_0001  4k
V_Vin     $N_0002 0 10V
R_Rb      0 $N_0001  6k
```

FIGURE 16

between 0 and 1 – it's perfect. Remember, the order of the node numbers is very important! Later on, when we ask *Schematics* for the current through Ra for example, *Schematics* will give us the current from node 2 to node 1. This is consistent with the passive sign convention we discussed earlier. So, while the particular node numbers are not vital, their order is: and we find that order in the netlist.

B. Naming Nodes

As for we *Schematics* users, this is about the only use we have for the netlist. After all, the whole idea of the *Schematics* program is to get away from Spice formats; node numbers and data statements. Here's the rub. We'll eventually want to know the node voltages in our circuit. We need a way to tell *Schematics* which nodes are of interest to us. The old Spice way was to use the node number but we are trying our best to avoid those. *Schematics* allows us to give any node a unique name. Let's call the output node "Vout". Simply double click on the wire at the output node and the dialog in Figure 17 will appear. Type in **Vout** as shown and select OK. The schematic should look like Figure 18. You can drag the Vout label wherever you like.

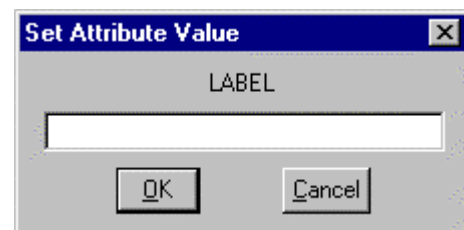


FIGURE 17

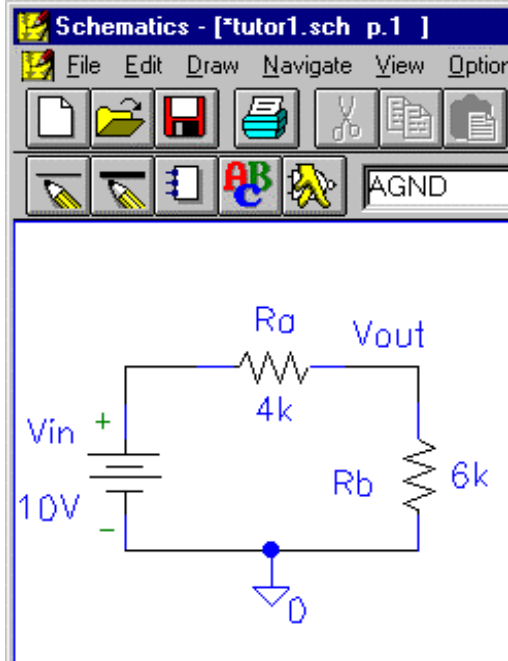


FIGURE 18

C. Displaying Voltages and Currents On the Schematic

In the **Analysis** menu, you'll find the **Display Results on Schematic** option. When enabled, PSpice will print all dc node voltages and branch currents directly on the schematic. This is very convenient since both the circuit and the results are on one screen. To delete an undesired result, click on it and press the Delete key. The final circuit is shown in Figure 19 where results for V_{in} , V_{out} and I are included.

D. Printing your Drawing

First, let's talk about the *Schematics* page and how it gets printed. While in *Schematics*, go to the **Z**OOM menu and select **P**age View. You should see something like Figure 20 where our circuit is quite small. This entire page is what *Schematics* might send to the printer if you

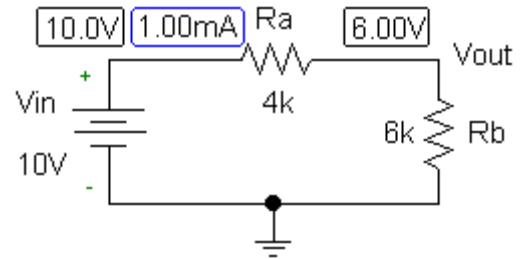


FIGURE 19

blindly select **P**rint in the **F**ile menu -- not very impressive.

Here's an easy way to isolate just the circuit for printing. Start by going to the **C**onfigure menu and selecting **D**isplay Options. When the Display Options dialog box comes up, de-select the **G**rid On check-box to turn off the grid dots and select **O**K. Next, point the mouse pointer above and left of the upper left hand corner of your drawing. Click left and hold. Drag the mouse beyond the lower right edge of the drawing. You should see a box grow as you drag, eventually surrounding your circuit. Go to the **F**ile menu and select **P**rint. The dialog box in Figure 21 appears. Select the options **O**nly Print Selected Area and **U**ser Definable Zoom Factor. For most of the small schematics you will create a scale of 125 to 200% will do fine. Other options are self-explanatory. Finally, select **O**K to print.

To incorporate your schematic into a different program such as **W**ord, draw a box around the diagram as described above then under the **E**dit menu, select **C**opy to Clipboard. You can now use the **P**aste command in **W**ord, **W**ordPerfect, etc. to include your circuit in a report.

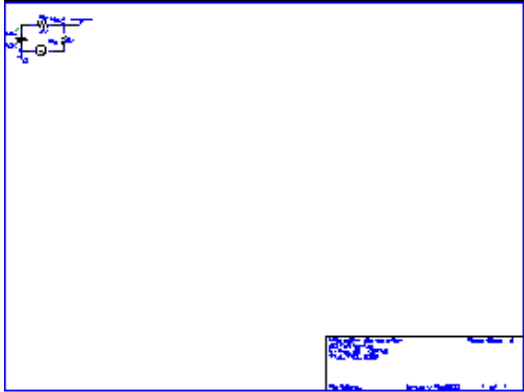


FIGURE 20

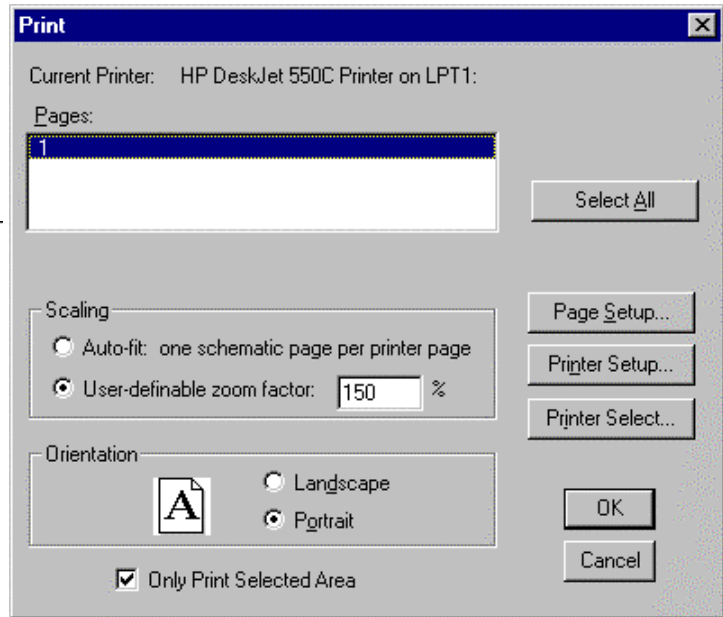


FIGURE 21

III. Running PSpice from Schematics and Choosing an Analysis Type

To simulate the circuit, we must first decide just what kind of analysis we want to perform. To see our options, select **Setup** from the **Analysis** menu. The **Setup** dialog box shown in Figure 22 appears. Choosing an analysis type is a very critical step. Since this dialog box can be a little daunting at first, let's keep it simple for now and just explain the **Bias Point Detail**.

A **Bias Point Detail** is nothing more than a dc analysis of the circuit. Simulation results include all node voltages, the currents through all voltage sources and the total power dissipation. PSpice will put these data, along with the netlist, in a **Notebook** or **Notepad** file called the output file. It has a .out extension. Node voltages and branch currents can still be displayed on the schematic using the **Display results on schematic** option in the **Analysis** menu.

A. Obtaining a Bias Point Detail

Now, back to our example. To perform a bias point detail analysis, make certain it is selected in the **Setup** dialog box in Figure 22 and select **OK**. Next, Select **Simulate** from the **Analysis** menu. If you haven't

already saved your circuit, *Schematics* will ask for a file name. If you've already done this, the **PROBE** window in Figure 23 will open. Of interest right now is the **Output** window, which informs us of simulations errors and completion. In Figure 23, we see there are no errors.

Finally, we're ready to look at the simulation results. However, before doing this, let's quickly solve for V_{out} in our circuit. By voltage division,

$$V_{out} = V_{in} \left[\frac{R_b}{R_b + R_a} \right] = 10 \left[\frac{6}{6 + 4} \right] = 6 \text{ V}$$

Also, the current around the loop should be 1 mA and the total power dissipation should be 10 mW. Let's see if PSpice agrees. Your schematic should look like Figure 24. Note the values for V_{OUT} and I are dead on.

As mentioned, all of the node voltages and voltage source currents are in the output file. Choose **Examine Output** from the **Analysis** menu. You'll see the beginning of a long **Notebook** file. Scrolling down, you should come to a section of the file similar to Figure 25 containing node voltages, voltage source currents and total power dissipation.

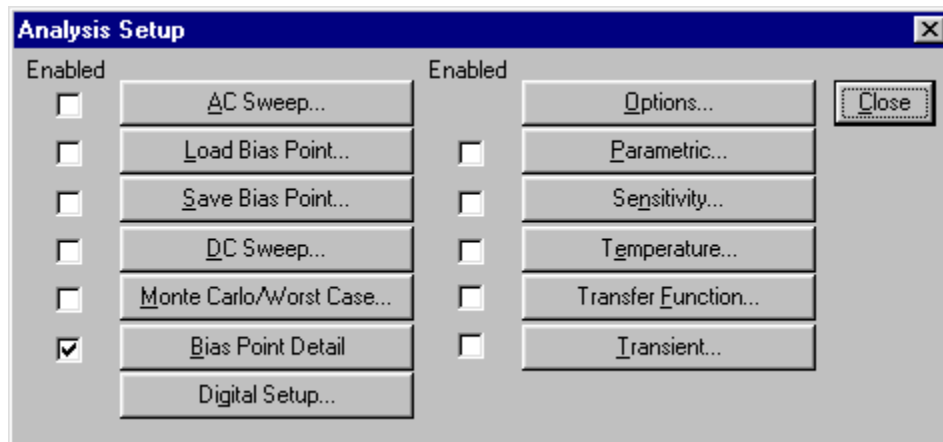


FIGURE 22

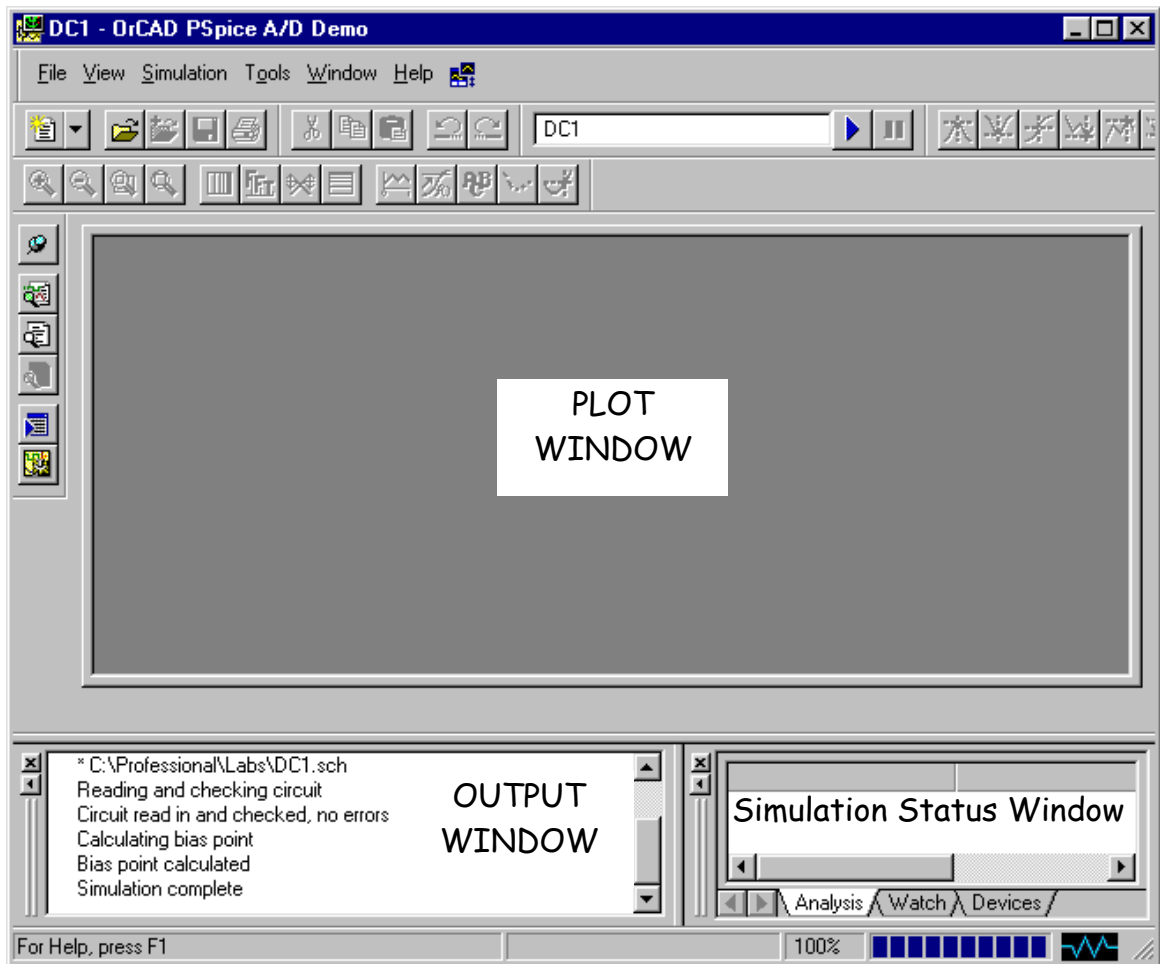


FIGURE 23

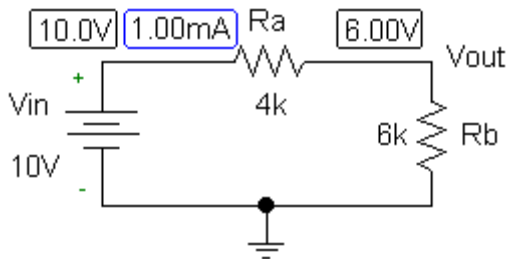


FIGURE 24

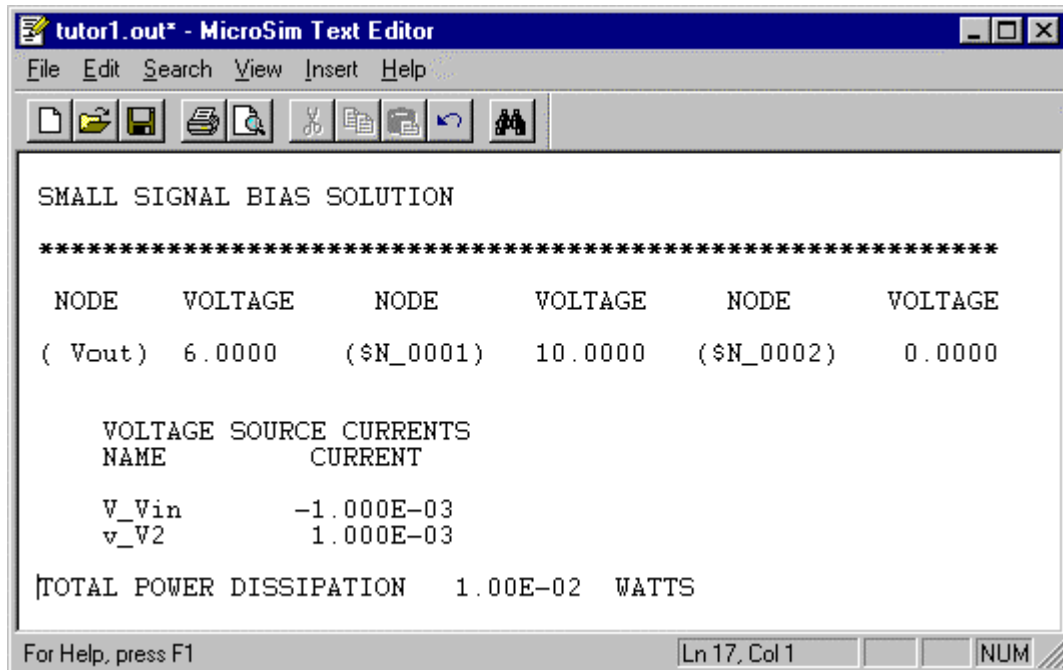


FIGURE 25

We see that indeed V_{OUT} is 6.00 V and the total power dissipation is 10 mW. The current bears closer inspection. PSpice says the current through V_{in} is -1 mA while we say it's +1 mA. Recall from Figure 1 that PSpice uses the passive sign convention wherein positive current enters the positive voltage terminal and exits at the negative terminal. PSpice is telling us that the current flowing top-down through our source (counterclockwise in Figure 2) is -1 mA. This agrees with our calculation of 1 mA flowing clockwise around the loop. Never let a computer fool you.

B. Obtaining a dc Sweep

Suppose we want to determine the effect of varying V_{IN} on the output voltage. We can run a quick dc sweep by enabling **DC Sweep** in the **Setup** dialog box of Figure 22. In a dc sweep, we vary some attribute of an element. It could be the resistance of a

resistor such as R1 or the voltage of V_{IN} . To perform a dc sweep, first be sure the **DC Sweep** checkbox has a check in it. Then, click on the text **DC Sweep**. *Schematics* now asks for the sweep details via the dialog box in Figure 26. Obviously, there's a lot more here than we need to sweep a dc source; so let's concentrate on the basics. We want a **Voltage Source** sweep, so select that. The name of the source is V_{in} , so type that into the **Name:** box. For now, let's choose a linear sweep. As for the range of V_{in} , let's choose -10 to +10 V in 0.1 V increments. After inputting that info, the DC Sweep dialog box should look like Figure 26. Click on **OK**, and you'll return to the **Setup** window in Figure 22. Click **OK** in this window and the analysis setup is finished. Now, choose **Simulate** from the **Analysis** menu and, most likely, something wonderful happens. PSpice simulates the dc sweep and puts you into the PROBE utility.

IV. Running and Using PROBE

A. Adding Traces

As mentioned in the early pages, PROBE does plots. In this case the x-axis will be V_{in} varying - 10 to + 10 V. We will set the y-axis to display V_{out} , the voltage across the 4 k Ω resistor and the current, I. A newly opened PROBE window looks like Figure 27. (Your PROBE graph may have a black background. I have gone to the MSIM.INI file in my Windows directory and changed the color of the PROBE DISPLAY BACKGROUND to BRIGHTWHITE. Also, in the VIEW menu, I've closed the Output and Simulation Status windows. This is just a personal preference.) Note that V_{in} is already the x-axis variable.

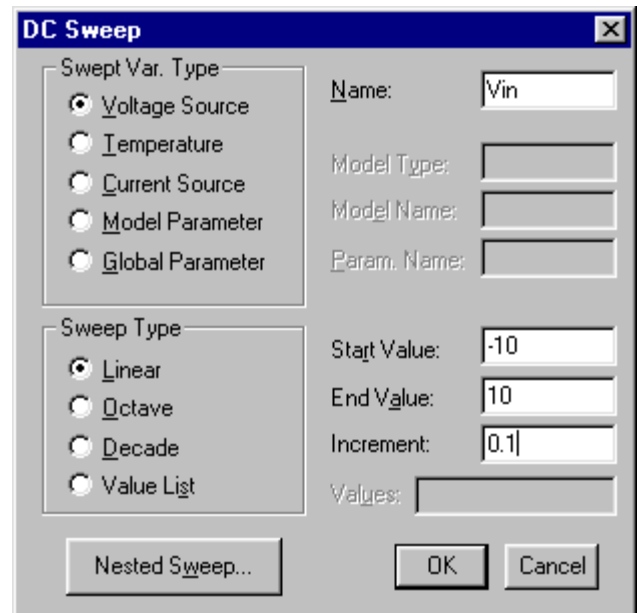


FIGURE 26

To put V_{out} on the y-axis, go to the Trace menu and select Add. The dialog box in Figure 28 appears containing all the node voltage labels and currents through each element. Note that the Alias Names and Subcircuit Nodes categories have been de-selected as they tend to clutter the trace listing. Find and click on $V(V_{out})$. Note that $V(V_{out})$ appears on the Trace Command line. Click OK and the graph changes to that in Figure 29.

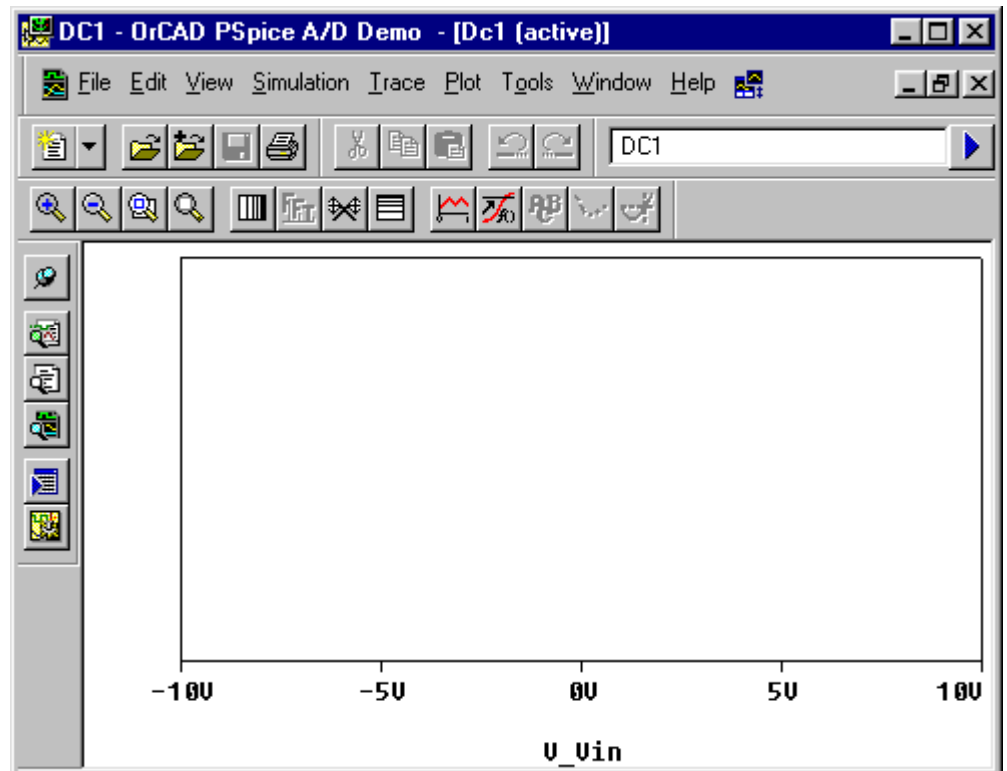


FIGURE 27

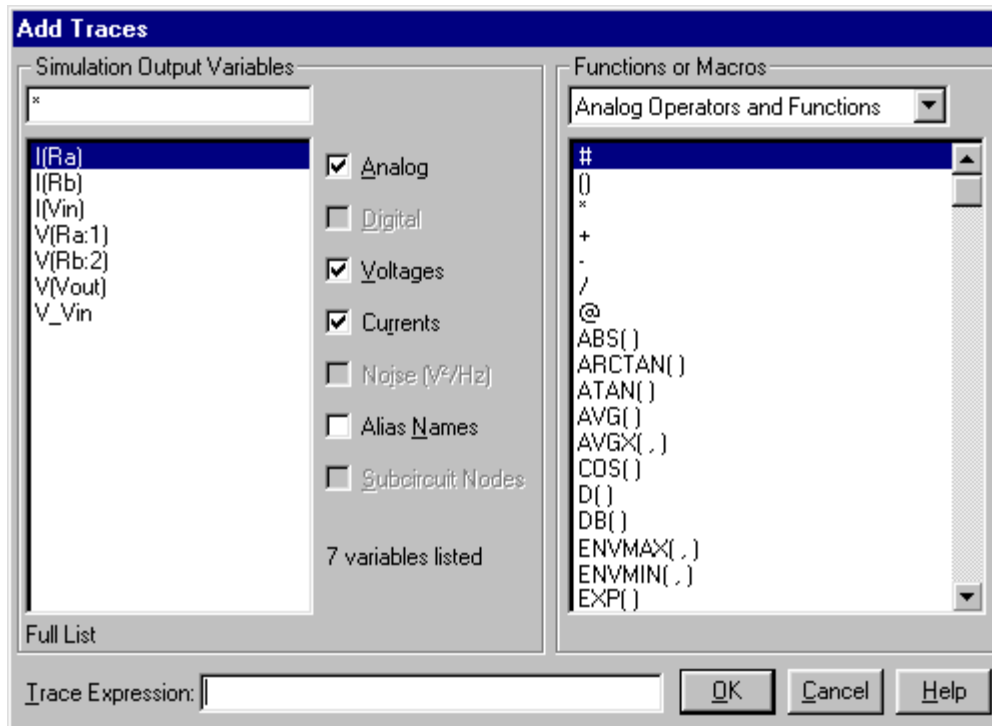


FIGURE 28

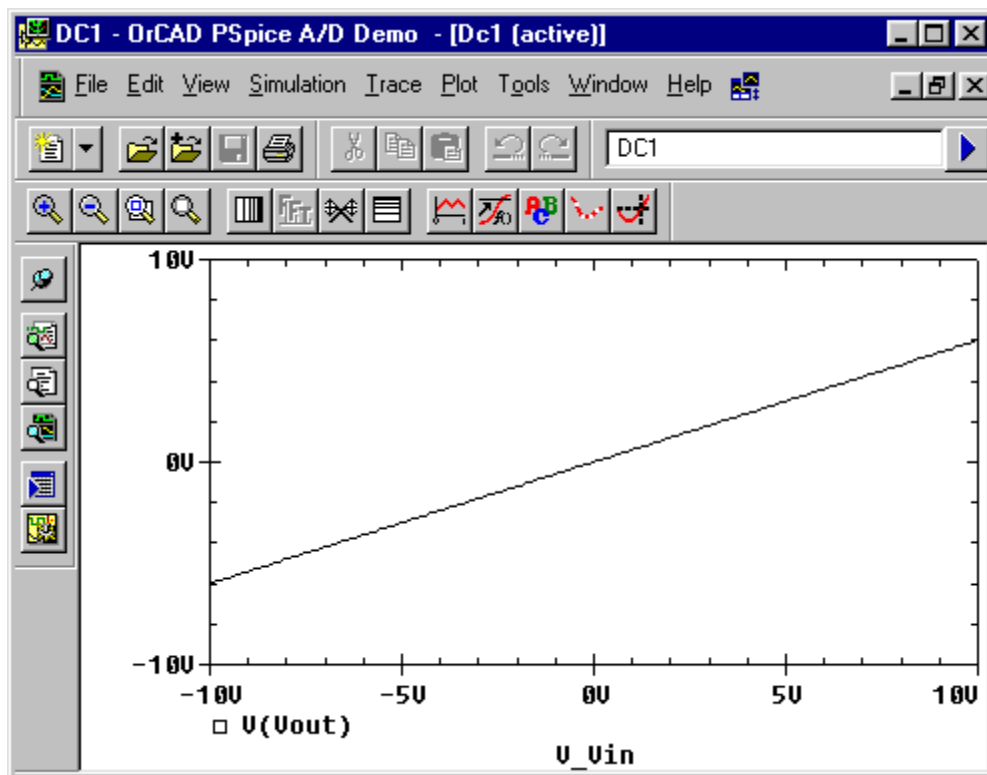


FIGURE 29

Next, we'll add the voltage across the 4 kΩ resistor to the plot. We know that one end of the resistor is at Vin and the other end is at Vout. Let's use that to our advantage. Under the **Trace** menu, choose **Add** again. Now, click on V(Vin). Place the cursor on the Trace Command line after V(Vin) and type a minus sign. Now, click on V(Vout) to finish the function. Click **OK** and the graph in Figure 30 appears showing both resistor voltages where each voltage trace has its own color. Since this manual might be printed in black only, I've labeled the traces.

Let's finish the PROBE intro by plotting the current. We must be careful here. Since the current is in the mA range while the voltages already plotted are in volts, the y-axis scale is thousands of times larger than the

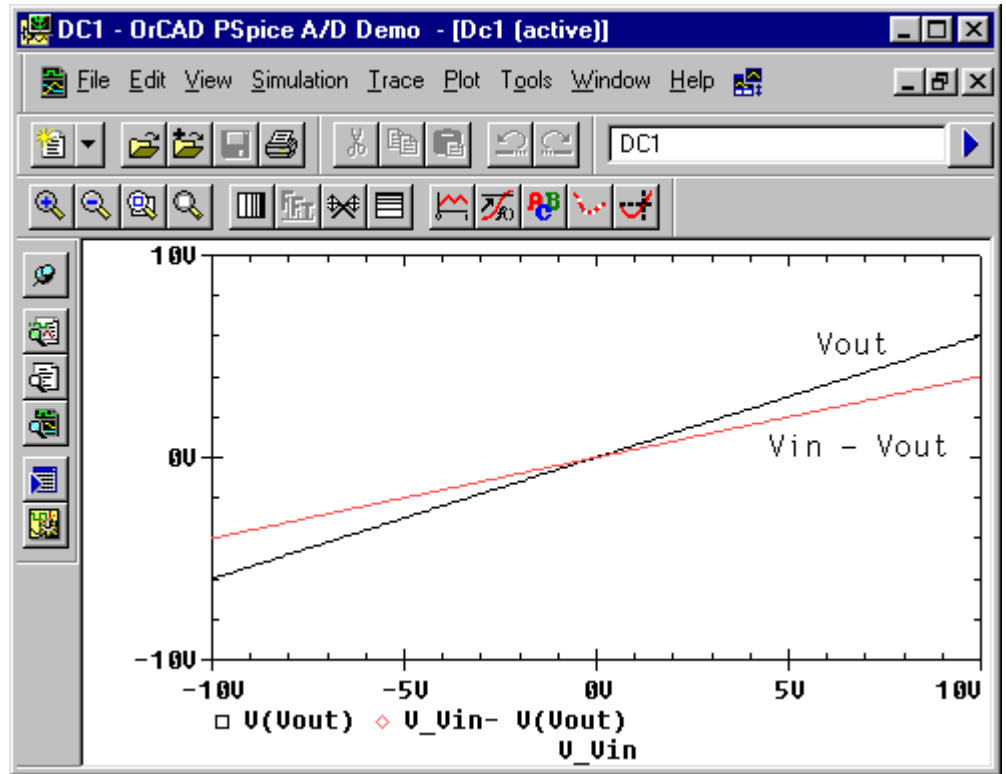


FIGURE 30

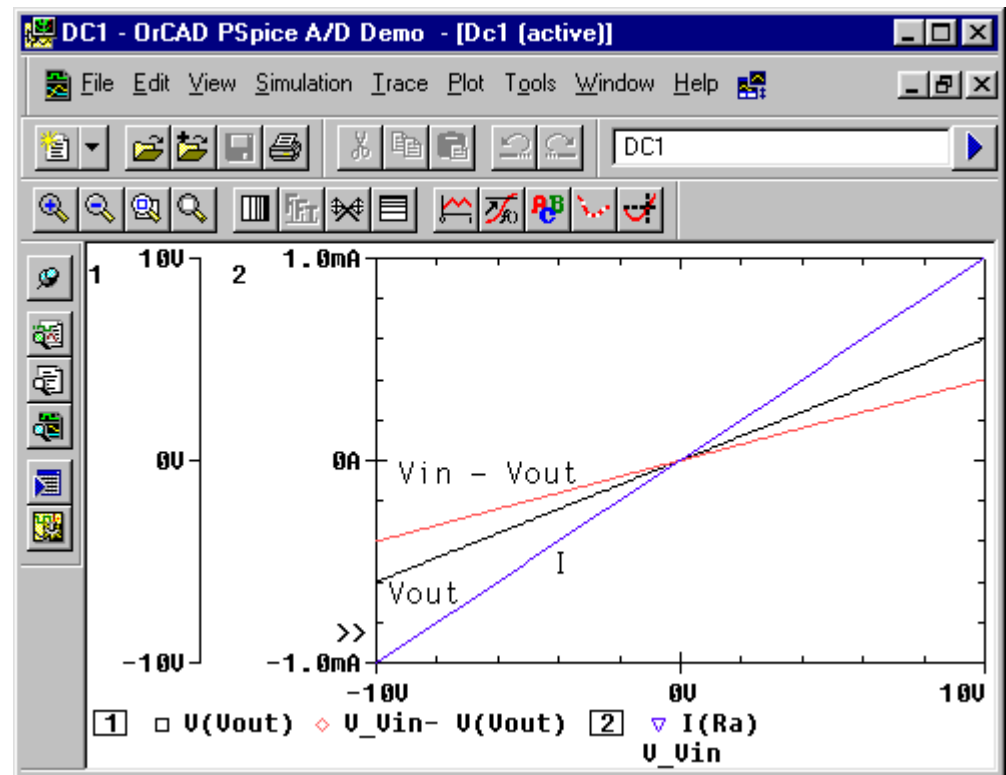


FIGURE 31

current value. We need a new y-axis. You'll find it in the **P**lot menu. Now we have one graph with two y-axes. Select **A**dd from the **T**race menu. In the **A**dd Traces dialog box of Figure 28, choose **I(Ra)** and click **O**K. The PROBE window looks like Figure 31.

B. Markers

An alternate method for adding traces is to use the markers within *Schematics*. In the Marker menu, you'll find markers for **V**oltage/**L**evel, **D**ifferential **V**oltage and **C**urrent into **P**in. Simply choose one and place it on the schematic. Voltage markers are placed at nodes and current markers are placed at pins. PROBE will plot all marked signals automatically. Unfortunately, all markers are plotted on the same y-axis.

C. Other PROBE Features - Label, Cursor, Copy and Save

The PROBE utility has many other features for modifying plots. Under the **P**lot menu are options for setting the ranges and labels for both the x- and y- axes. Under the **E**dit menu, you can give the plot a title. But, the most important features are under the **T**race, **P**lot and **T**ools menus. There you'll find the **L**abel, **C**ursor, **C**opy to Clipboard and **D**isplay Control features. The **L**abel option allows you to annotate your plots with graphics and text as I've done in the PROBE plots here. The **C**ursor is a pointer you control with the arrow keys. It moves along the plot trace displaying the x and y values at its present position. It also has a myriad of other capabilities that I'll leave to you to discover.

The **C**opy to Clipboard command does exactly that; allowing you to then paste the plot into other applications. One

disadvantage of the **C**opy to Clipboard command is that it copies the PROBE window exactly as you see it on the screen. The black background makes the copy less than ideal for inkjet or laser jet printing. An alternative is to alter the PSEVAL.INI in your WINDOWS directory. In it you will find the color schemes for screen displays and printing in both *Schematics* and PROBE. Setting the PROBE DISPLAY background color to BRIGHTWHITE will change the black PROBE graph on your screen to white. This change will show up when you paste the PROBE plot into other programs as well.

Display Control is very important feature. It saves your PROBE plots including any text and labels you have added. We'll save our plot as tutor1. Select **D**isplay Control and the dialog box in Figure 32 appears. Simply enter tutor1 as the name for the plot and select **S**ave. Later if you want to see the plot again, you must open the PSpice file tutor1.sch first, go into PROBE, and select **D**isplay Control. Click on the plot name and select **R**estore.



FIGURE 32

V. Your First Schematic

It's time for you to draw and simulate your first schematic. Using the techniques described in the previous pages, use PSpice to find V_X and I_X in the circuit in Figure 34. Remember the step-by-step procedure.

- 1) Draw the schematic. Enable the **Display Results on Schematics** option to display dc simulation results directly on the schematic. Also, give names to the nodes of interest in your circuit.
- 2) Use **SETUP** to choose the kind of simulation you want to perform.
- 3) Simulate.

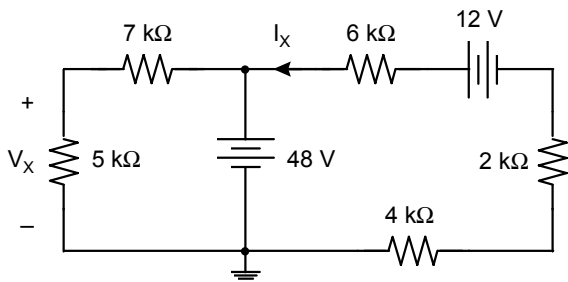
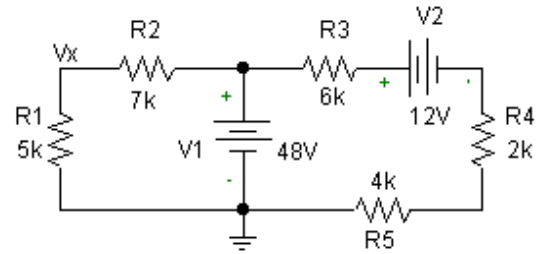


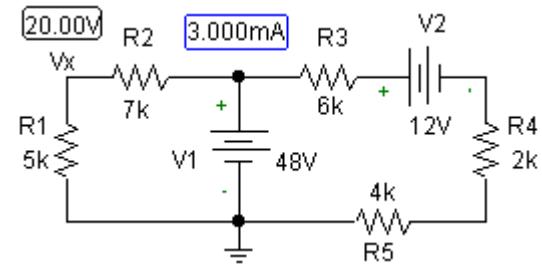
FIGURE 34

Your simulation should yield, $V_X = 20\text{ V}$ and $I_X = -3.0\text{ mA}$. Of course, the negative sign on the current means the direction of flow is opposite its defined direction in Figure 34.

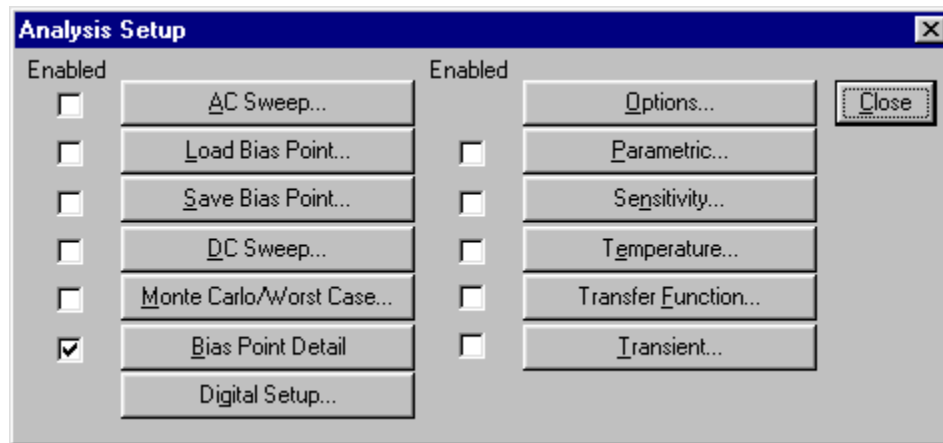
Should you get lost, figures for the schematic, the **SETUP** menu and the simulation results are shown in Figure 35.



(a)



(b)



(c)

FIGURE 35

PART III. CONSTRUCTING AND SIMULATING AN AC CIRCUIT

I. Obtaining ac Sweeps and Bode Plots

A. ac Sweeps

Performing an ac sweep implies you're simulating a frequency dependent circuit -- a circuit containing inductors and/or capacitors and steady-state sinusoidal sources. In other words, the you're working in the frequency domain with phasors and complex impedances. As an example, let's simulate the first-order, low-pass active filter in Figure 36.

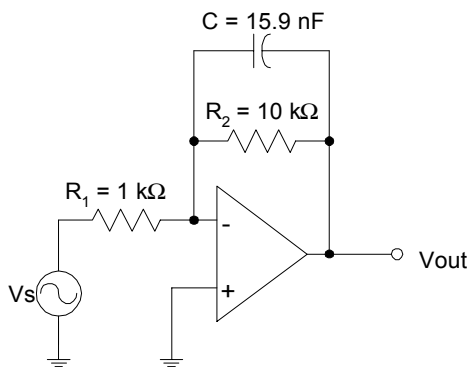


FIGURE 36

A simple op-amp model is shown in Figure 37. It contains three elements: the input resistance, R_{in} , the output resistance, R_{out} and the gain element or dependent voltage source. For a perfect op-amp, R_{in} and the gain, A , are infinite and R_{out} is zero. Here we'll use values $R_{in} = 1 \text{ M}\Omega$, $R_{out} = 50 \Omega$ and $A = 100,000$. Inserting Figure 37 into Figure 36 yields Figure 38. Since every element in Figure 38 is in the PSpice libraries, we're ready to go to *Schematics*. The capacitor part name is *C*, and the voltage dependent voltage source is called *E*; it's in the analog.slb library. After much

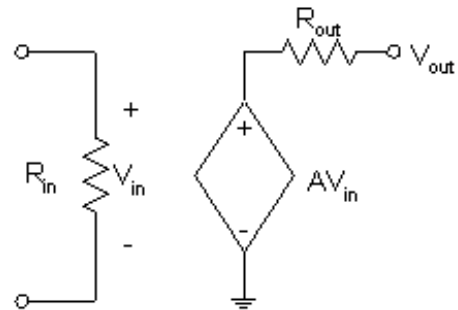


FIGURE 37

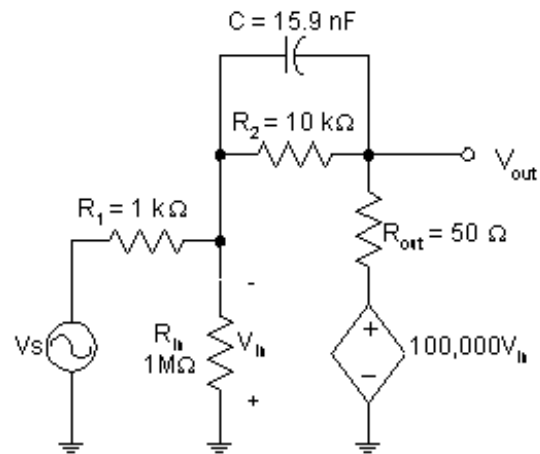


FIGURE 38

Draw/Get New Part/Wire etc., the schematic in Figure 39 appears. Through its Attributes dialog box, Figure 40, the source V_S has been given the values, $DC = 0$ and $AC = 1$. To set the gain to 100,000, double click on the dependent voltage source symbol and, in its Attributes dialog box, select **Gain**. Type **-100000** in the **Value** box, **Save Attribute** and **OK**. The negative sign is required since the voltage the dependent voltage source, *E1*, is sampling has the opposite polarity of V_{in} in Figure 39. One last thing, in Figure 40, note that R_{in} is $1 \text{ meg}\Omega$, not $1 \text{ M}\Omega$. PSpice is case insensitive; so $1 \text{ M}\Omega$ is the same as $1 \text{ m}\Omega = 0.001 \Omega$.

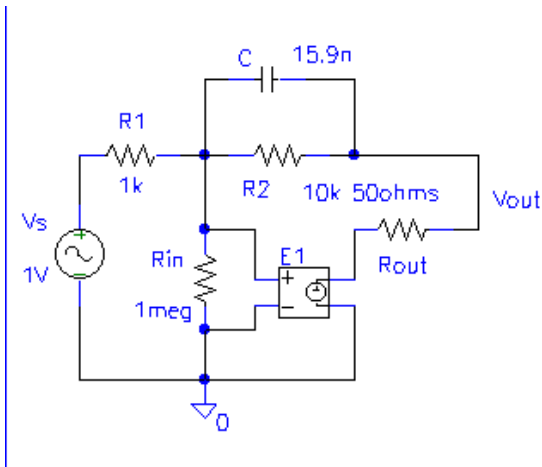


FIGURE 39

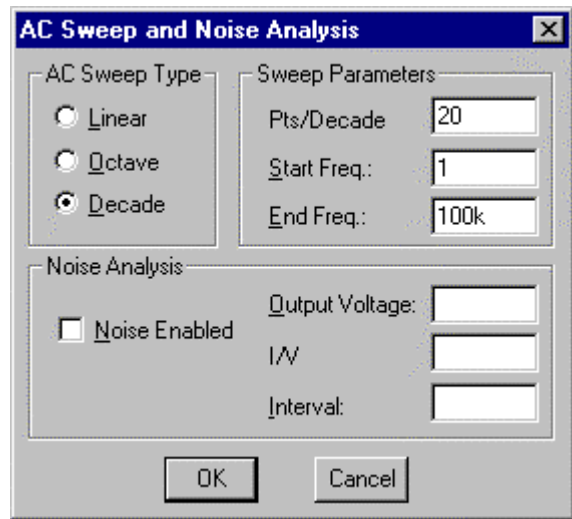


FIGURE 41

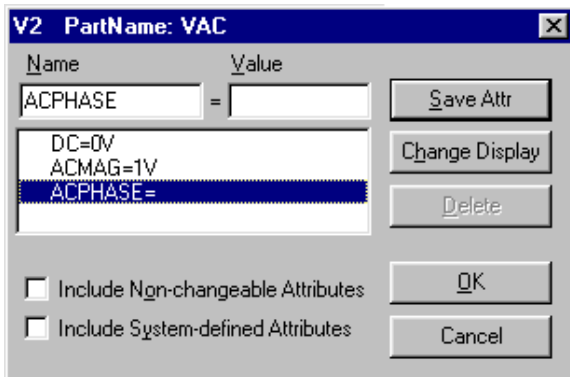


FIGURE 40

To perform the ac sweep, go to Setup under the Analysis menu. When the Setup dialog box in Figure 22 appears, select AC Sweep. The AC Sweep and Noise Analysis dialog box in Figure 41 appears. As in the DC Sweep case, there are lots of options. Since we don't know the frequencies at which the circuit's performance will change. Let's put in a wide frequency range, from 1 Hz to 100 kHz. Additionally, we'll choose Decade rather than Linear in the AC Sweep Type box. This will make a smoother plot. So far we have a frequency range of 5 decades. Choosing 20 points per decade, we'll have 100 data points overall. Once the selections are made, click OK to return to the Setup window. Click on Close to get back to *Schematics*.

Under the Analysis menu, select Simulate and, if the netlist is good, we should eventually see the PROBE window in Figure 42. Note that the x-axis is already frequency ranging from 1 Hz to 100 kHz on a log scale. Now we will plot V_{out} . Since all voltages and currents are phasors with magnitude and phase, we'll discuss how to get *Schematics* to plot both components. First, we'll plot magnitude. Select Add under the Trace menu. When the Add Trace dialog box in Figure 43 appears, select type click OK. You should see Figure 44. The gain is indeed frequency dependent! The gain? We've plotted V_{out} not V_{out}/V_S . Yes, but in the V_S attributes, we chose AC = 1. This means $V_{in} = 1/0^\circ$ V. So, the gain is $V_{out}/V_S = V_{out}$!

To plot the phase of V_{out} , type **P(Vout)** in the Trace Command line. A warning first, since the phase may change dozens of degrees, while the magnitude range is only 0 to 1, we should put magnitude and phase on separate y-axes. In the Plot menu, select Add y-axis. Then adding P(Vout) to the plot will produce Figure 45.

From Figure 45 we see that the gain at low frequencies is about 10. Using the Cursor and Label options under the **Tools** menu, I've marked the frequency at which the gain has decreased by a factor of 0.707. This

frequency is several names: corner frequency, cutoff frequency, half-power frequency and bandwidth. For our circuit, it is 1 kHz.

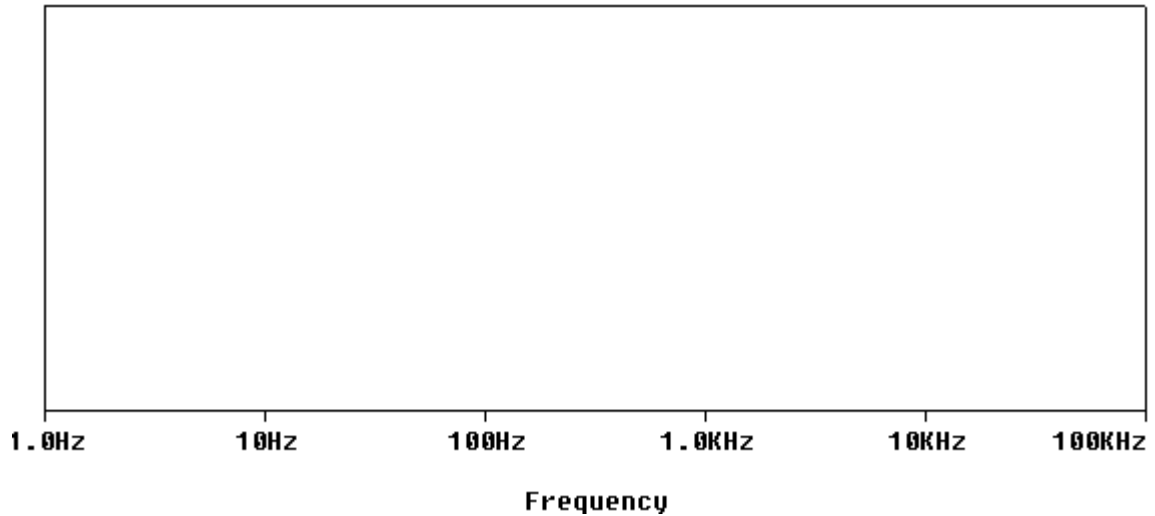


FIGURE 42

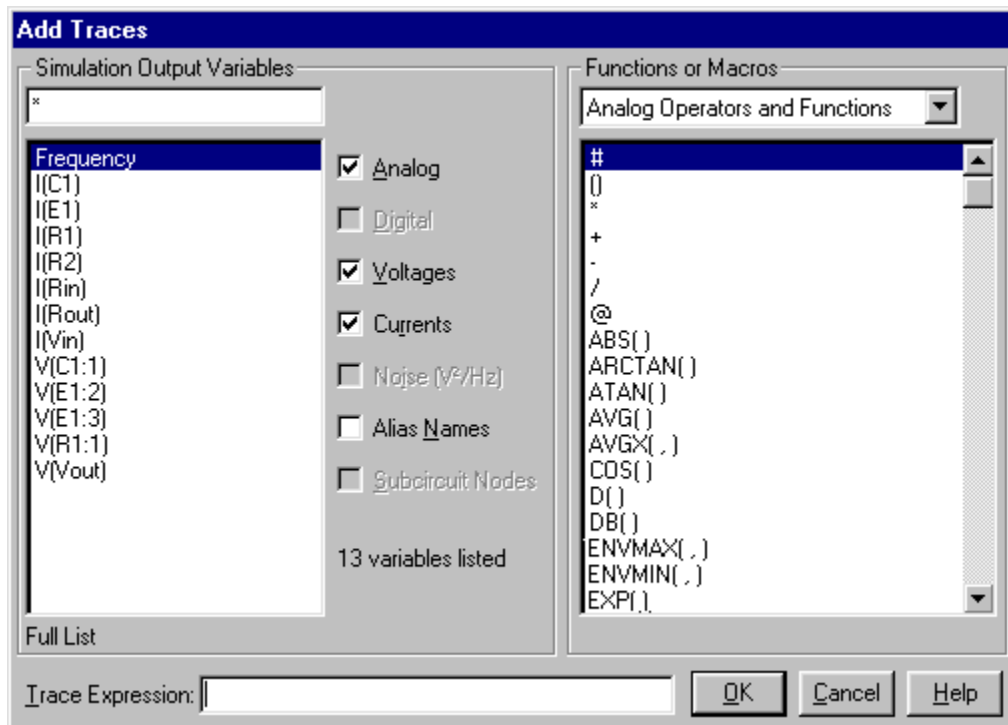


FIGURE 43

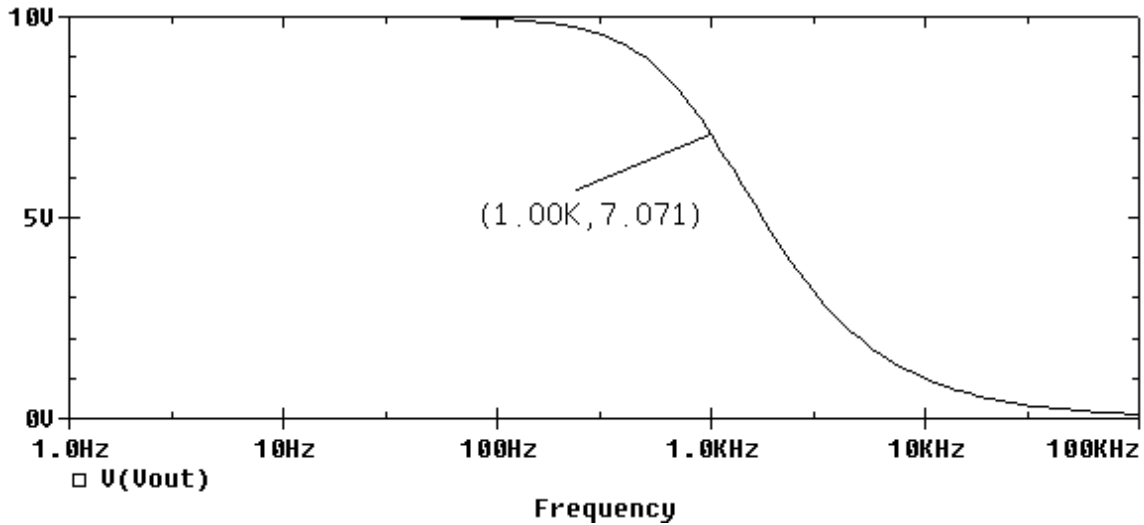


FIGURE 44

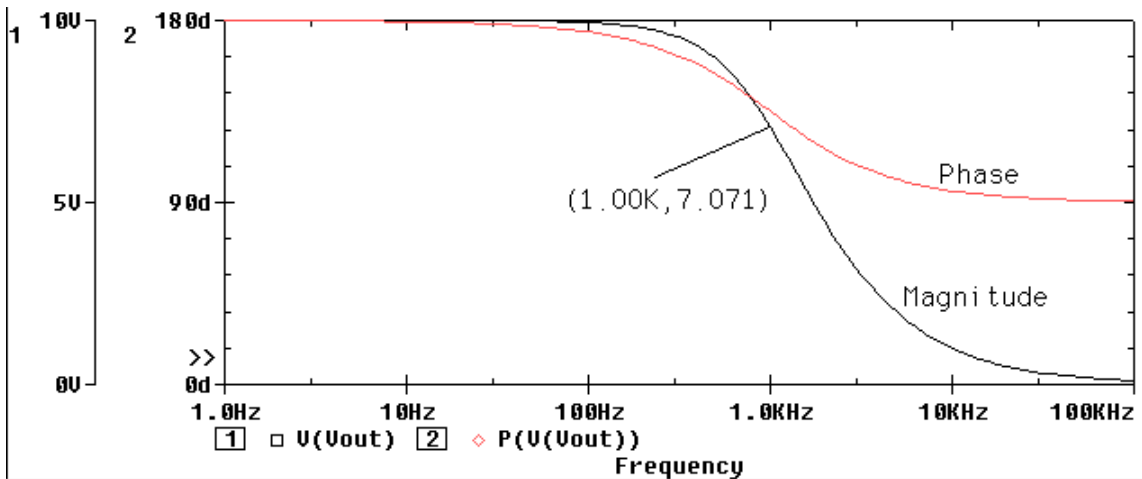


FIGURE 45

B. Making Bode Plots

Notice also that the gain plot in Figure 44 is nonlinear. This makes extracting information from the plot more difficult. For example, at what rate is the gain decreasing? Well, once we go beyond about 100 Hz, the gain does decrease. However, the rate of decrease, or the slope, is not constant. A way to linearize frequency domain graphics is to plot the gain in decibels, or dB. The dB equivalent of a voltage or current gain is

$$\text{number of dB} = 20(\log_{10} A)$$

We invoke the dB in PSpice by adding the trace **DB(Vout)** on the Trace Command line. You'll find the DB operator on the righthand side of the **Add Traces** window in Figure 44. We may as well remove the magnitude trace. To do this, click on the text

M(Vout)

in the lower left corner of the plot. It will become highlighted. Simply press the

Delete key. The PROBE window should look like Figure 46 (I added the labels). This is the Bode Plot! Notice also that the dc gain is +20 dB. Given that its magnitude in Figure 43 was 10, and given the expression for dB in terms of gain, how

could it be otherwise? Finally, the phase, in degrees, starts at 180° and decreases to 90°. Notice that at 1 kHz the gain has decreased by 3 dB from its dc value. Therefore, -3 dB corresponds to multiplication by 0.707. You'll get lots of chances to use that nugget.

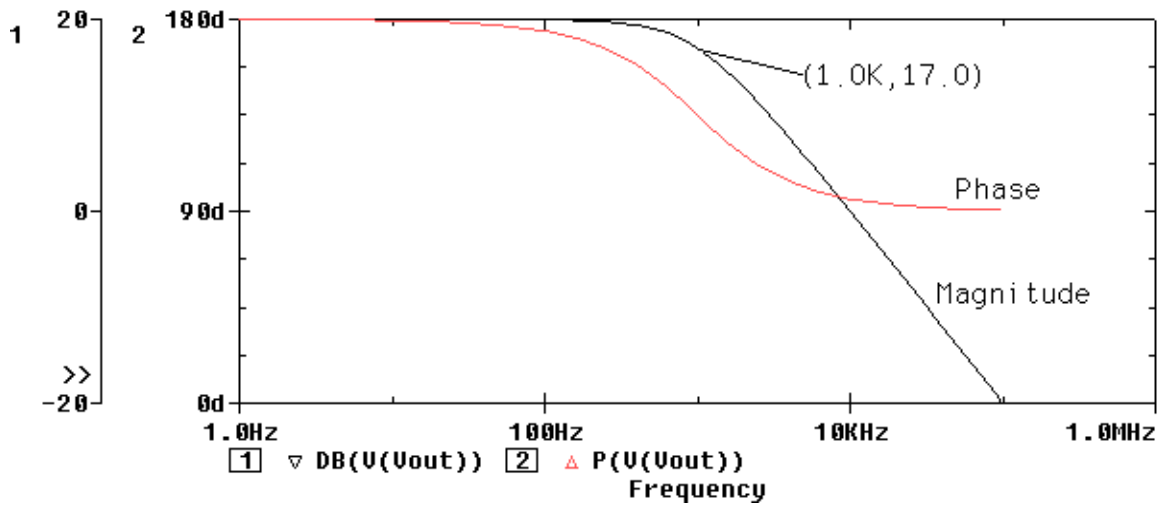


FIGURE 46

PART IV. CONSTRUCTING AND SIMULATING A TRANSIENT CIRCUIT

I. Transient Analysis

Now we get into simulating the time varying characteristics of circuits. A typical example is the unit step response of RC, RL and RLC circuits. Such a circuit is shown in Figure 47, where I_S is a dc current source. Before $t = 0$, the circuit is in steady-state operation. That is to say, the currents and voltages are constant. At $t = 0$, the switch opens and the voltages and currents fluctuate to new steady state values.

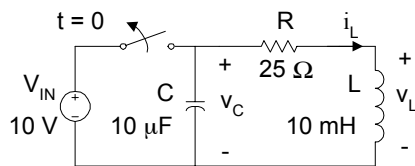


FIGURE 47

To “build” the circuit in *Schematics*, you need switches. In the EVAL library, you will find two switches called Sw_tOpen and Sw_tClose. These switches open and close, respectively, at used defined times. For this simulation, we need the Sw_tOpen switch. After the circuit is finished, the *Schematics* diagram looks like Figure 48. A warning – when you get current sources, *Schematics* places them such that positive current flows down the page. To make Figure 48 look like Figure 47, I had to rotate the current source 180°.

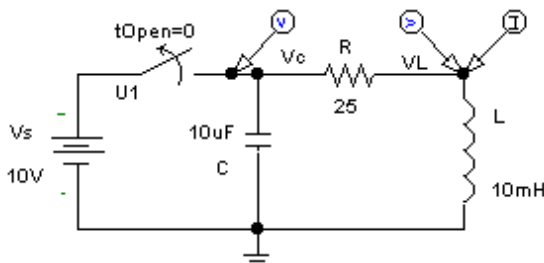


FIGURE 48

To set the switches attributes, double click on the switch part to open its Attribute window, shown in Figure 49. Parameters of interest are

- tOpen – time at which switch opens
- ttran – elapsed time between closed and open conditions
- Rclosed – switch resistance when closed
- Ropen – switch resistance when open

Here’s how the switch works. Beginning at time tOpen and ending at time Topen + ttran, the switch’s equivalent resistance changes linearly from Rclosed to Ropen. It is critical for good simulation results that ttran be much less than the circuit’s time constant, that Rclosed be much less than and Ropen much greater than the resistors in the circuit. The values in Figure 49 meet all of these requirements.

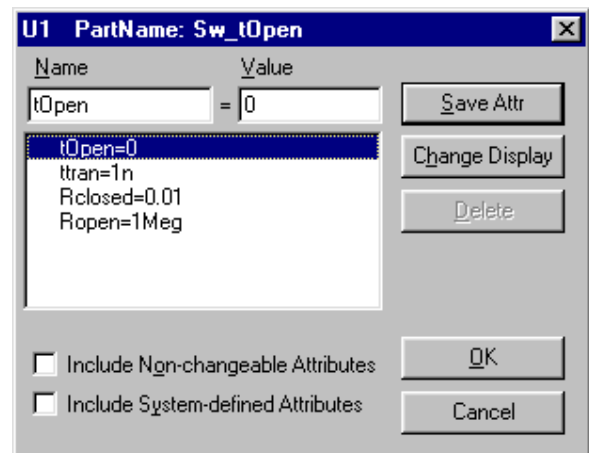


FIGURE 49

There’s one last issue before we assign attributes to the current source -- the initial conditions. How do we get the capacitor and inductor to reach steady-state before $t = 0$? *Schematics* does it for us! You see, before every transient or AC analysis, *Schematics* first finds the dc bias detail whether we ask for it or not. Then, *Schematics* uses these values as the starting point for the transient

simulation. When performing these dc calculations, *Schematics* does two things of interest to us here. First, it treats capacitors as open circuits and inductors as shorts circuits. Second, in transient analyses, *Schematics* uses the current/voltage sources values at $t = 0$. The end result is *Schematics* uses all the proper data to find the values of $v_C(t)$ and $i_L(t)$ just before switching.

Next we select Setup and click on Transient. The Transient dialog box in Figure 50 appears. By default, the simulation starts at 0 seconds. As stated at the outset, we want a 5 ms simulation. A step ceiling of 5 us will yield 1001 data points -- plenty.

Click OK. Back in the Setup window, be sure that the Transient checkbox has a check in it. Click OK. Now, Run PSpice. When the PROBE window appears, just add traces for V(Vout), I(Is) and I(L). (You may want to use 2 y-axes). Our screen looks something like Figure 51. Notice that all voltages and currents have the proper values before switching. Using the Cursor

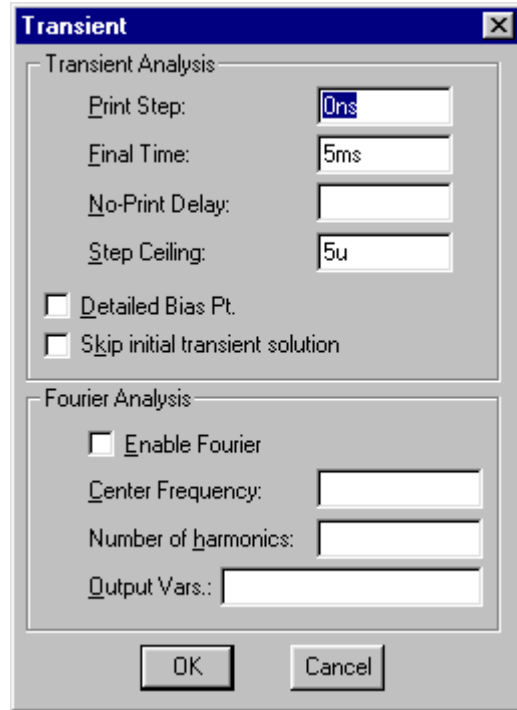


FIGURE 50

tool, the initial conditions are found to be $v_C(0) = 1\text{ V}$ and $i_L(0) = 1\text{ mA}$ -- perfect. The decaying exponential nature of the waveforms is indicative of damped second order systems.

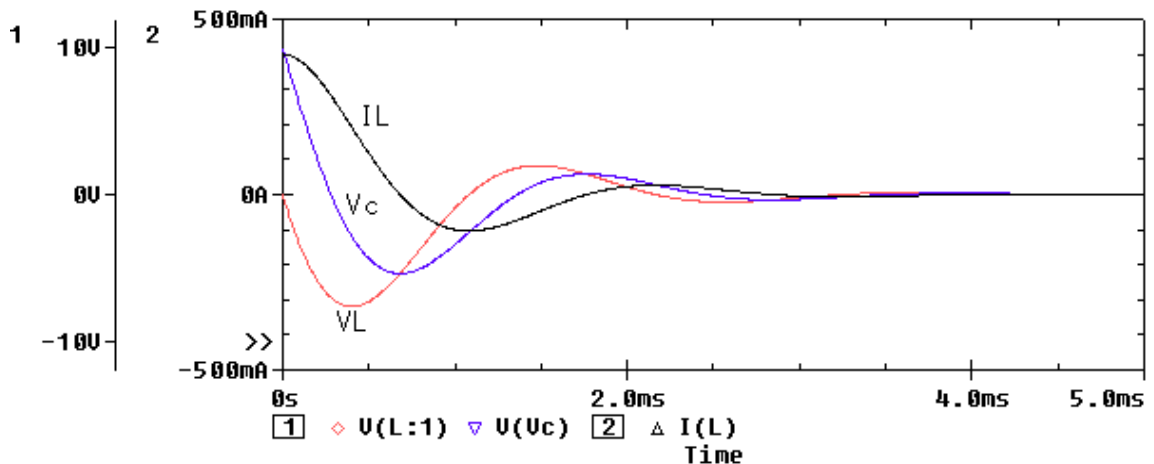


FIGURE 51

B. Setting Initial Conditions

Should you ever need to set the initial voltage across a capacitor or the current through an inductor to a particular value, use the IC option in the Attribute window of that particular component. For example, Figure 52 shows an inductor's attribute window. The IC field is the initial condition of the inductor current. If left blank, PSpice will determine the initial condition as forced by the external circuitry when it performs the Bias Point Detail part of the simulation.

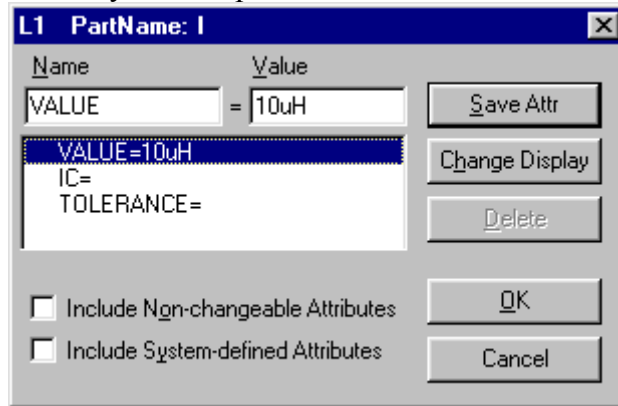


FIGURE 52

PART V. CONCLUSIONS

As you proceed through a *Schematics* simulation, your work goes through several gyrations. There's the diagram you draw, the netlist, the output file, the data for voltages and currents that PROBE uses for plotting, and so on. To keep things organized, the **Design Manager** puts these various components of your simulation in individual files. Each kind of file has its own special three-letter extension. Listed below are the fundamental files created during a simulation along with their extensions and a brief description.

Extension	Description	Creation
.cir	circuit file; contains simulation commands and most of the netlist information.	when you <u>S</u> ave, Run <u>P</u> Spice or <u>C</u> reate Netlist
.dat	binary data file; contains simulation results for use by PROBE	when you <u>S</u> imulate
.net	netlist file; contains the netlist information	when you Run <u>P</u> Spice or <u>C</u> reate Netlist
.out	output file; contains bias point data, netlist, model parameters and error messages	when you <u>S</u> imulate
.sch	schematic file; contains your drawing, parts attributes and <u>S</u> etup choices	when you <u>S</u> ave or Save <u>A</u> s
.prb	PROBE file; contains PROBE plots including labels and cursors.	when you use <u>D</u> isplay Control/Save

As we said at the outset, Schematics is only one part of a software suite called the **DesignLab**. Listed below are brief descriptions for these programs.

Schematics - Used for drawing circuits, creating netlists and invoking PSpice through the Run PSpice command.

PROBE - Used to produce plots of based simulated voltages and currents.

PSpice A/D - This is the actual simulation engine. The A/D suffix means that PSpice can simulate both analog and digital circuits. When we use *Schematics* we use PSpice A/D indirectly through the Run PSpice command.

PISyn - A signal integrity checker. Used in advanced digital simulations.

MSimPCB - A printed circuit board layout program that interfaces with *Schematics*. A package type can be defined for each part in you schematic. Based on your netlist, MSimPCB will generate the board layout.

Optimizer - An optimization program. You input the circuit performance you want to optimize and Optimizer will, based on your netlist, find the best solution.

Stimulus Editor - Basically, it is similar to PROBE in that it displays the voltage and current source waveforms. In addition, we can graphically edit the waveforms. This is very useful in digital simulations.

Full manuals for each of these **Design Manager** components are included on the Evaluation Version 9.1 CD-ROM and download.

We've covered the basics of *Schematics*, and PROBE. At this point, you should be able to navigate you way through simple simulations for dc, dc sweep, ac and transient analyses and produce output plots. If you look through the menus, the Help utility and ORCAD's documentation, you quickly see that *Schematics*, even this Evaluation Version, can do a lot more than we've covered. My advice is to spend some time exploring the program before the eve of your first *Schematics* homework assignment. With a little practice, you'll find yourself simulating complex circuits quickly and accurately.