

PART 11

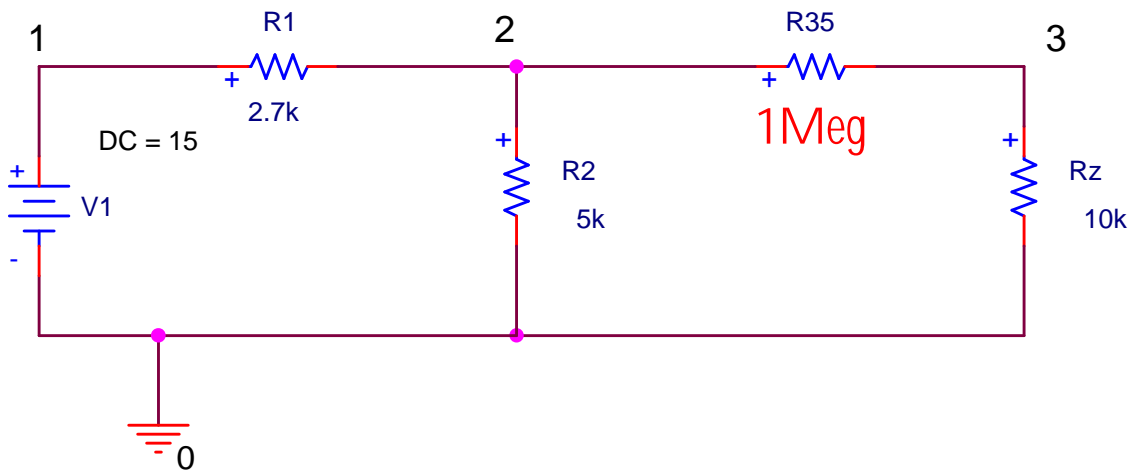
Creating and Using Netlists

For this entire text, we have created circuits schematically and then run simulations. Behind the scenes, Capture generates a netlist from the circuit schematic and then PSpice runs the netlist to simulate the circuit. Before the Schematic Capture front end was added to the PSpice suite of programs, users would write a netlist manually and then simulate the circuit by having PSpice simulate the circuit described by the netlist. Netlists are actually easier to use than drawing a schematic when you have fairly simple circuits. You can easily create a short netlist and then run PSpice. Netlists become cumbersome for large circuits or when you need to reference sub circuits such as OP-AMPS where you have to know the order of the calling nodes in order to use the sub circuit.

In this chapter, we will choose a few of the circuits from most chapters to illustrate how to create netlists. Although you may never write a netlist, it is useful to know how to read netlists because during some simulation, you may have an error that you cannot figure out. In desperation, you may look at the netlist created by Capture and find the error. Netlist errors are rare, but they do occur. Also, if you create your own sub circuits for use with Orcad Capture, you may have to create your own netlist for the sub circuit and then create a graphic symbol that references the sub circuit.

11.A. Resistive Circuit Nodal Analysis

In our first example, we will simulate the resistive circuit of Section 3.A. We will create a netlist and run the PSpice simulation. The first thing we need to do is draw the circuit and then give all nodes a label. Ground is always given the node label 0 (the number zero). Other nodes can be labeled with numbers or given text labels. I will label the circuit as shown:



We have labeled the nodes as 1, 2, and 3, and ground is always labeled as node 0.

Next, we need to create the netlist for the circuit elements. Voltage source names must start with the letter V. Thus, our name of V1 is ok. The netlist line for a voltage source specifies the name of the source, the node for the positive terminal, the node for the negative terminal, and then the value of the source. The line for our source is

```
V1 1 0 DC 15
```

This line specifies that voltage source V1 is connected between nodes 1 and 0, node 1 is the positive voltage terminal, and it is a 15 volt DC source.

Resistors names must start with the letter R. All of our resistor names start with the letter R, so we will use the names on the schematic. In the netlist line for a resistor, you specify the resistor name, the positive reference node, the negative reference node, and then the value. Since a resistor is not polarized, from a circuit operation point of view, the order in which we specify the nodes do not effect the operation of the circuit. However, if we ask for the current through the resistor, we need to know which direction PSpice considers positive. When you create a netlist for the resistor, the first node is the positive voltage reference and the second node is the negative reference. Once we know the polarity for voltage, we know that the positive direction for current always enters the positive voltage reference, as shown in Figure 11-1. We can now specify the netlist line for R1.

```
R1 1 2 2.7K
```

This line states that R1 is connected between nodes 1 and 2 and its value is 2.7 k Ω . Since node 1

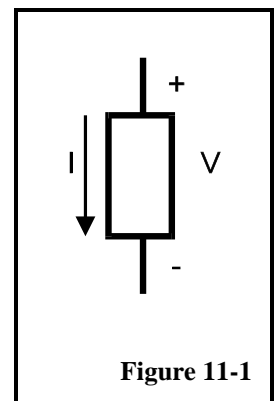
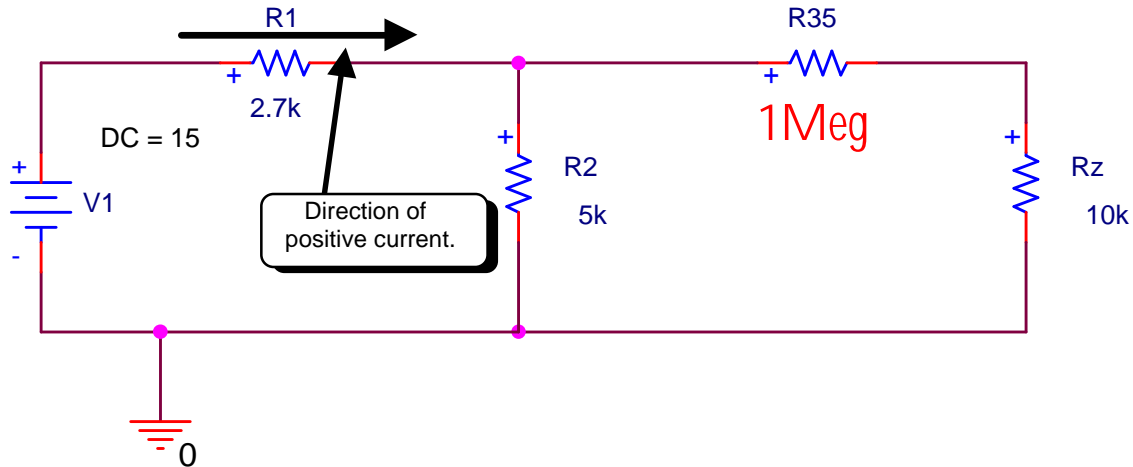


Figure 11-1

was first, it is the positive reference and the definition of positive current for R1 is from node 1 to node 2, or from left to right:



Now that we know which direction PSpice interprets as positive, when we ask for the current through R1, which would be specified as I(R1), we know how to interpret the answer.

The remainder of the circuit components are resistors, and we know how to write the netlist for those:

R2 2 0 5k

R35 2 3 1MEG

RZ 3 0 10k

The netlist states that R2 is connected between nodes 2 and ground (0) and its value is 5 k Ω , R35 is connected between nodes 2 and 3 and its value is 1 M Ω , and RZ is connected between nodes 3 and ground and its value is 10 k Ω .

The DC nodal analysis (or bias point) is always run, so we do not need to specify a simulation for this circuit. The last line we need to add is the .end statement. The complete netlist is shown below:

Bias Point Solution of a Resistive Circuit

*

* This is file Book Circuits\Netlists\section 11A.cir\

*

V1 1 0 DC 15

R1 1 2 2.7K

R2 2 0 5k

R35 2 3 1MEG

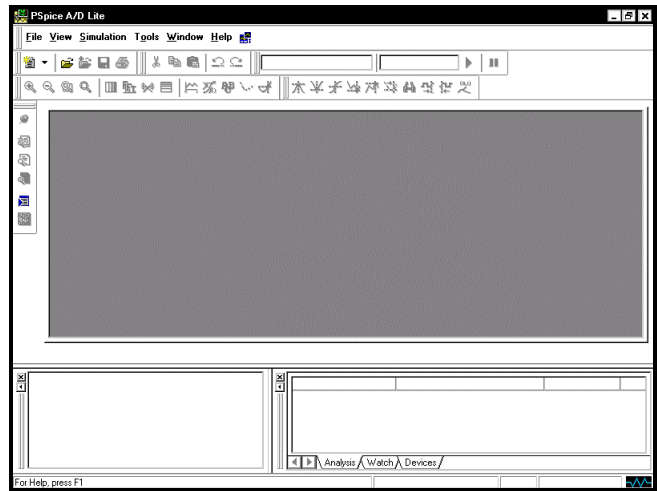
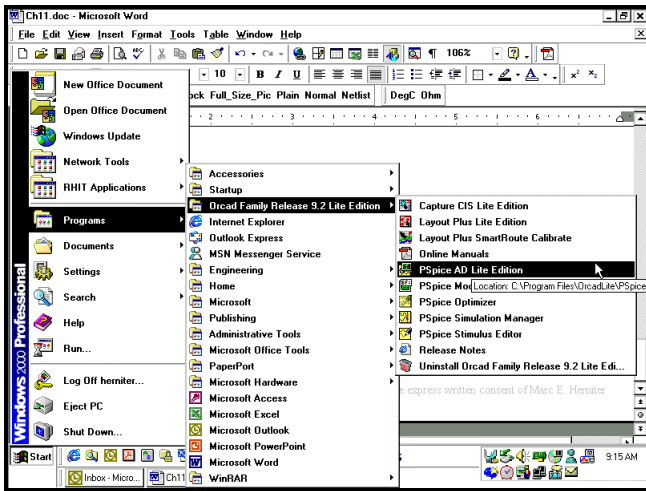
RZ 3 0 10k

.end

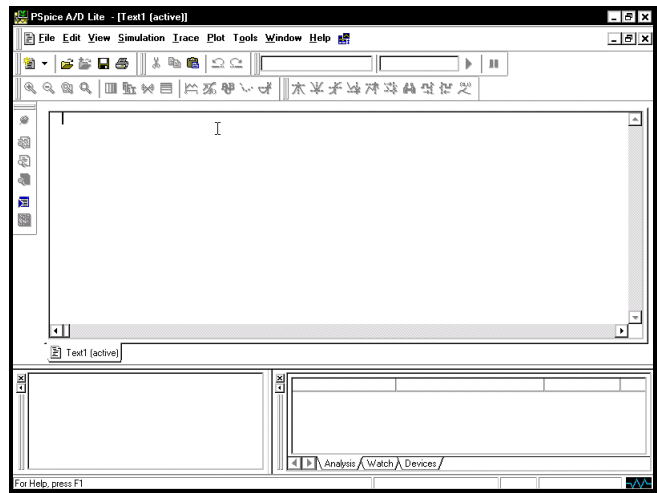
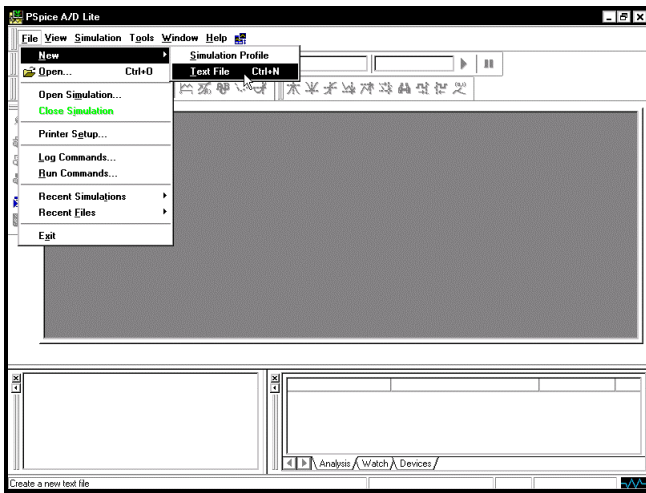
The first line of the file is the title and is required. If you forget to place the title in the first line, and instead put a line that describes a circuit component in the first line, that line will still be used as the title, and the component will not be used in the simulation. Also, any lines where the first character is an asterisk (*) are comments and ignored by PSpice.

The next thing we need to do is to create this netlist in a text file. You can use any text editor you wish or you can use the text editor included with PSpice. The file we create must use the “.cir” extension. If you use the Windows Notepad, the default extension is “.txt” and you may need to rename your file. We will use the text editor that comes with PSpice.

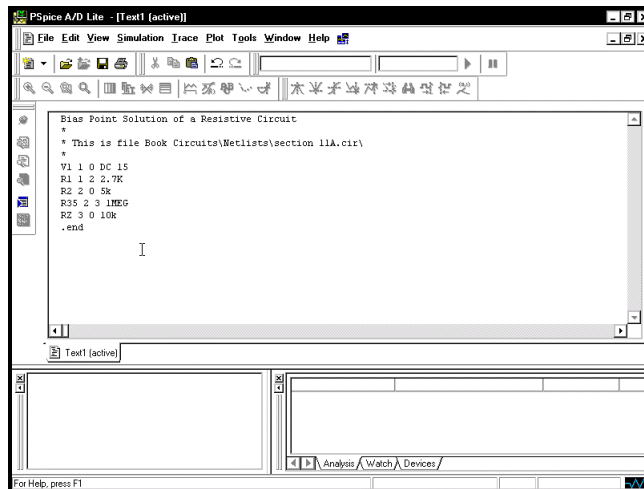
To run PSpice, click on the **Start** menu, select **Programs**, and then look in the **Orcad Family Release 9.2 Lite Edition** group and select **PSpice AD Lite Edition**:



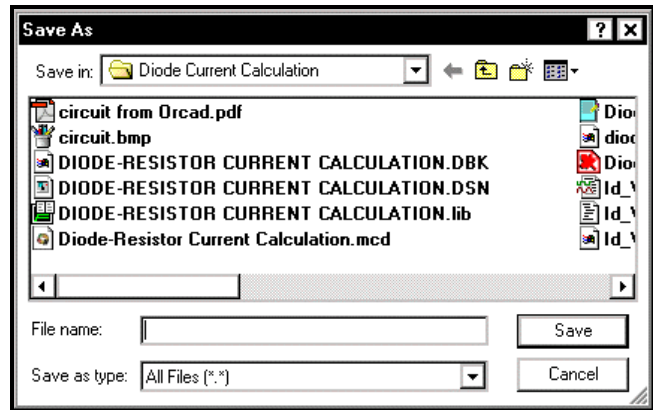
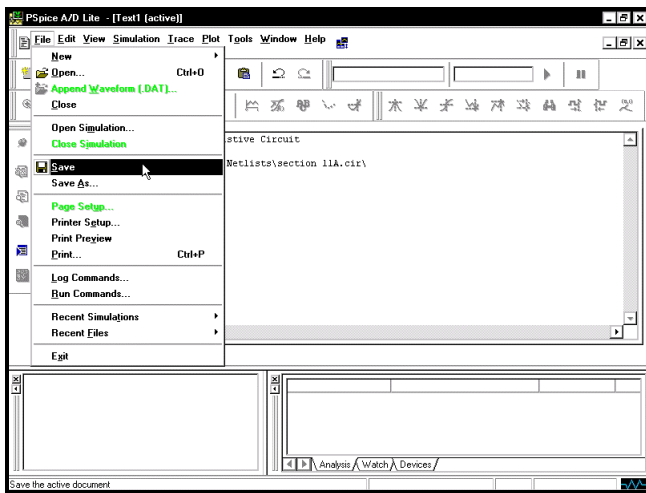
Select **File**, **New**, and then **Text File** from the menus:



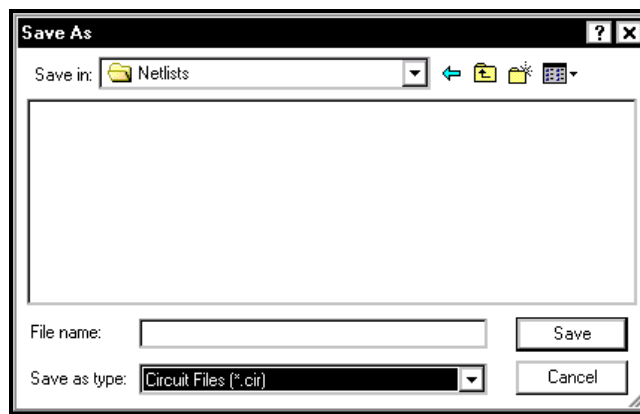
Enter the netlist:



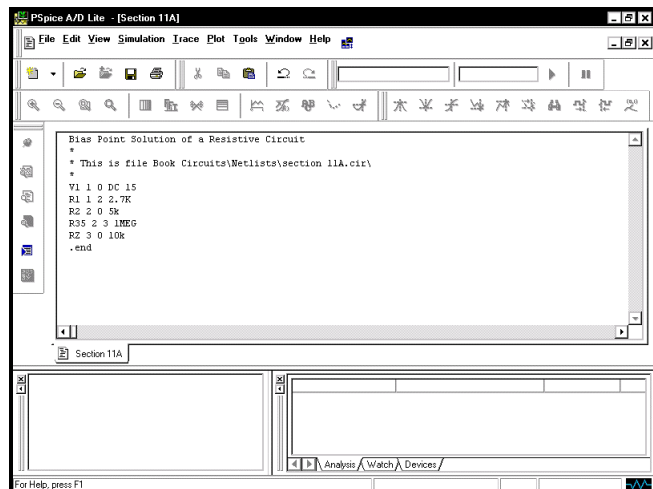
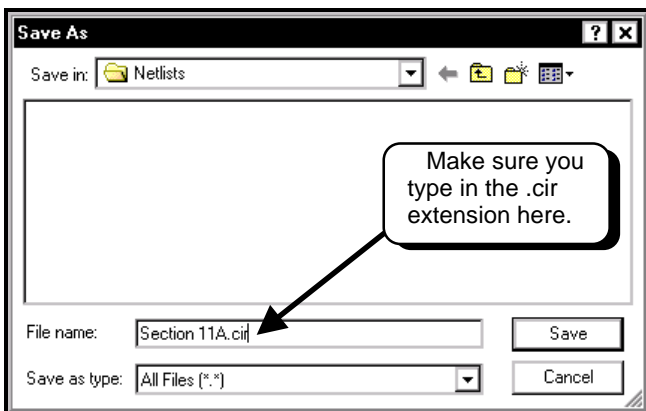
We must now save the file. Select **File** and then **Save** from the PSpice menus:



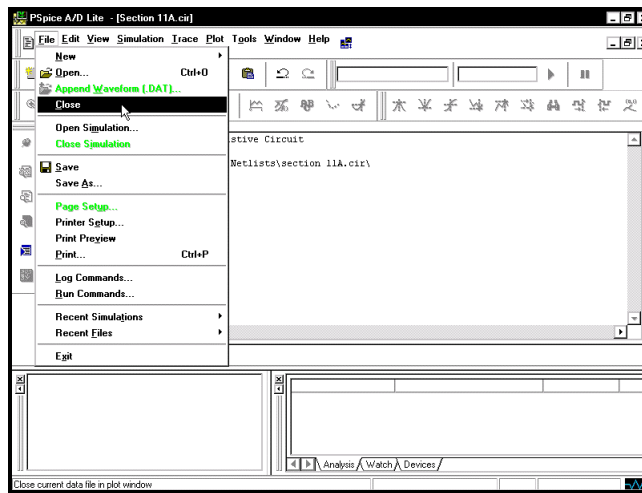
Change the directory to the one in which you want to save the file and also change the file type to .cir:



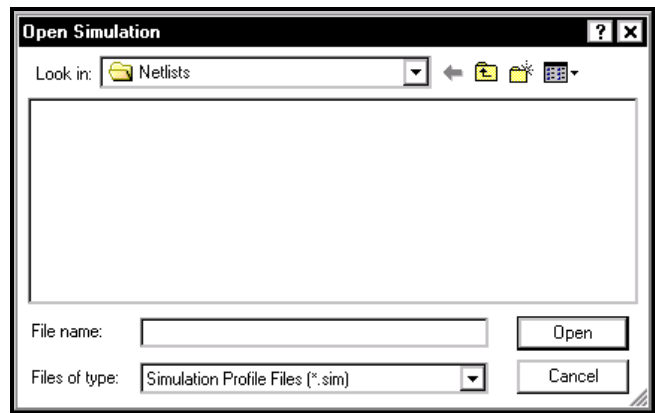
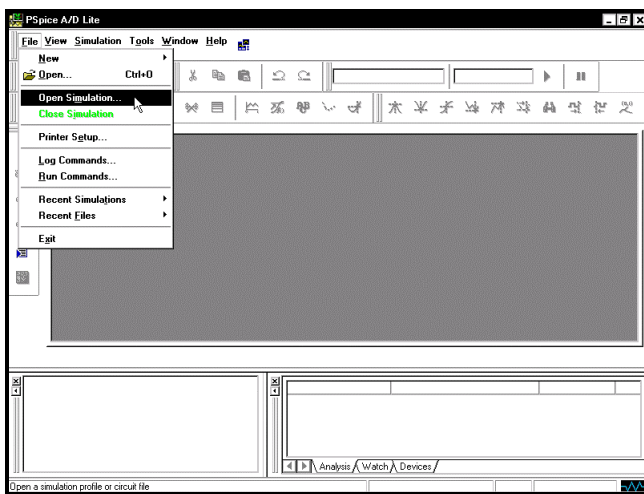
Specify a name for the netlist and then click the **Save** button:



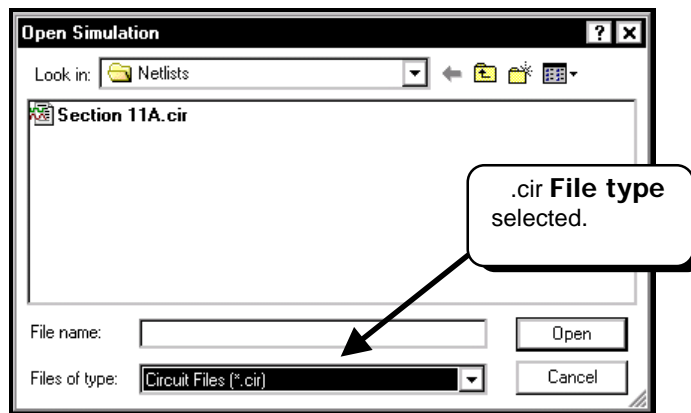
Note that you must type in the “.cir” extension when you specify the file name. We must now close the file and run the simulation. Select **File** and then **Close** to close the text file:



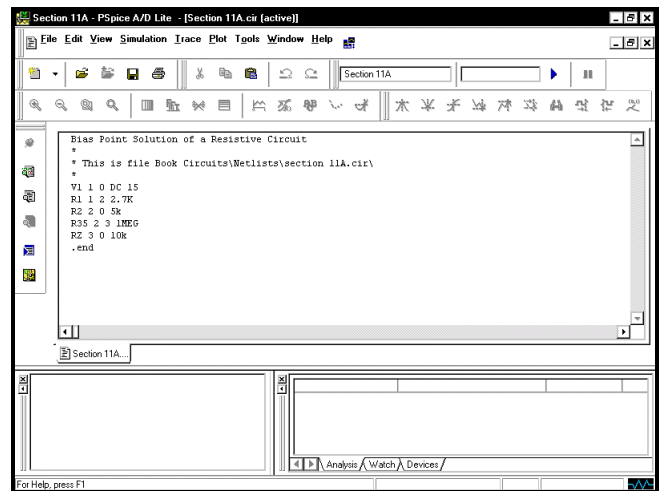
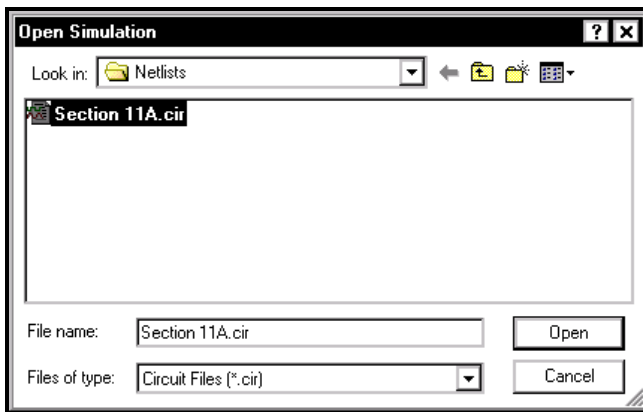
Select **File** and then **Open Simulation**:



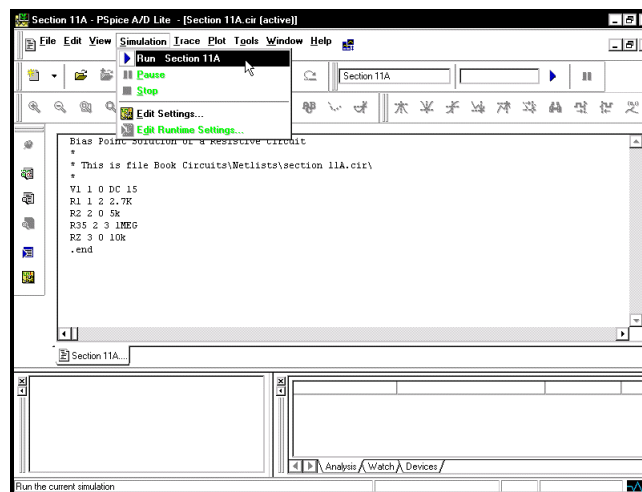
Select the **.cir** File type:



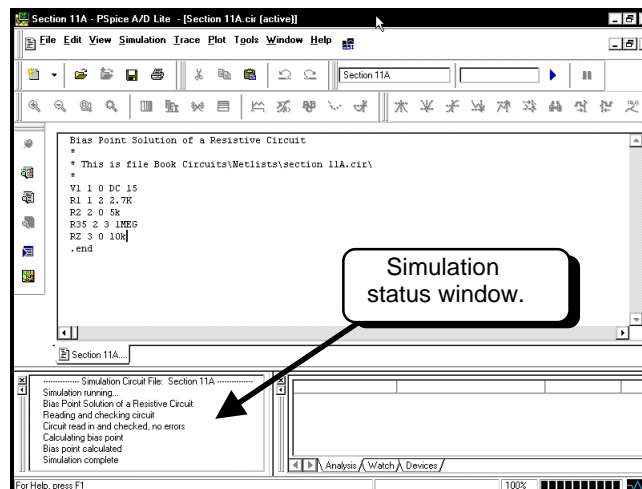
Select the circuit file you wish to simulate and then click the **Open** button:



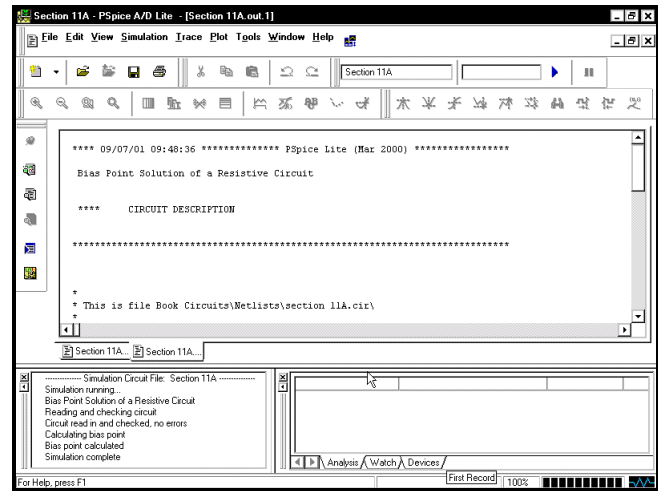
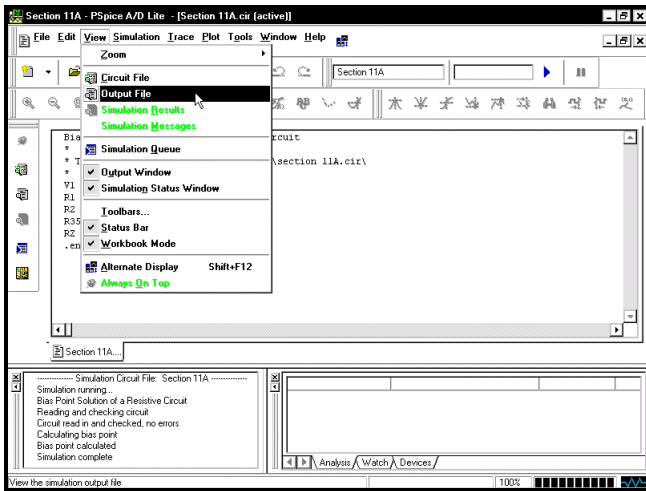
We can now run the simulation. Select **Simulation** and then **Run** from the menus:



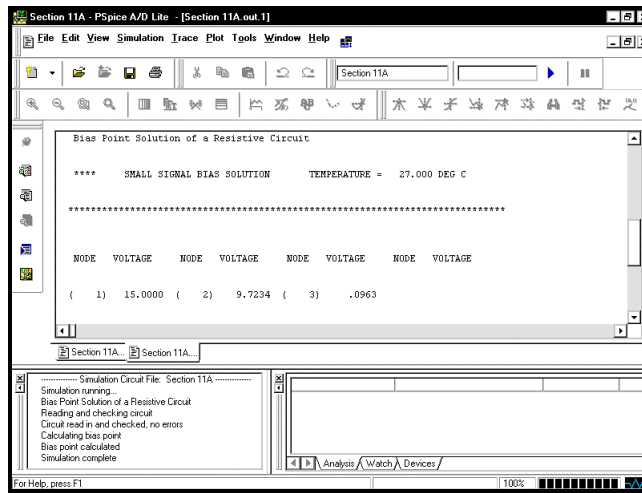
When the simulation runs, information will be displayed in the simulation status window:



My window states that the simulation completed successfully. If you have any errors, you would be notified in this window as well. The results for this simulation are stored in the output file. If your circuit has errors, they will be listed in the output file as well. We will now look at the output file. Select **View** and the **Output** file:



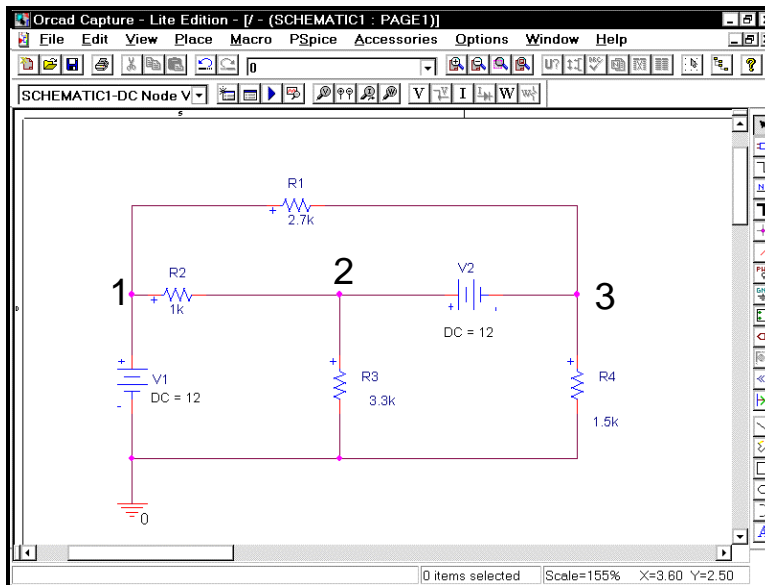
The results of the bias point analysis are near the end of the file, so scroll the window until you see the text shown below:



The results show that node 1 is at **15.0000** volts relative to ground, node 2 is at **9.7234** volts relative to ground, and node 3 is at **.0963** volts relative to ground.

Select **File** and then **Close** to close the output file, and then select **File** and then **Close Simulation** to unload the netlist.

EXERCISE 11-1: Find the DC node voltages for the circuit below:



SOLUTION: Create the netlist and run the simulation. Node numbers are assigned as shown above. The netlist is:

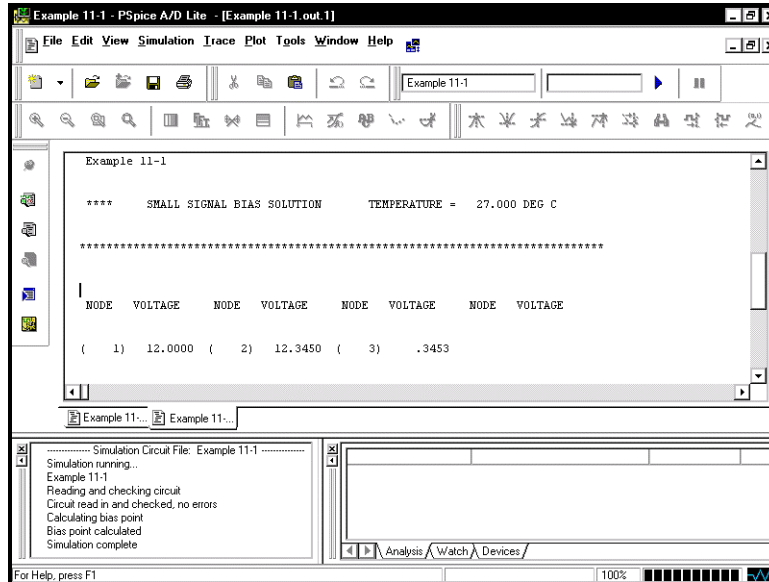
Example 11-1

```

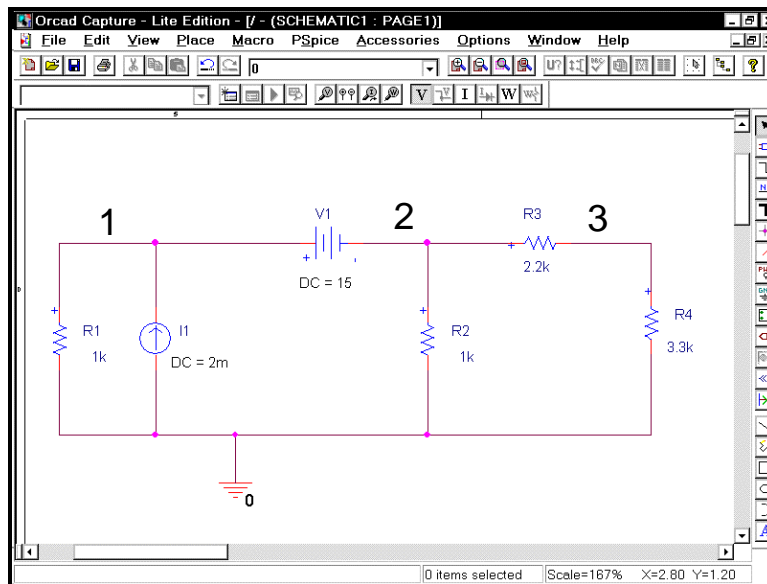
V1 1 0 DC 12
R1 1 3 2.7K
R2 1 2 1k
R3 2 0 3.3k
R4 3 0 1.5k
V2 2 3 DC 12
.end

```

Run the simulation and view the output file:



EXERCISE 11-2: Find the DC node voltages for the circuit below:



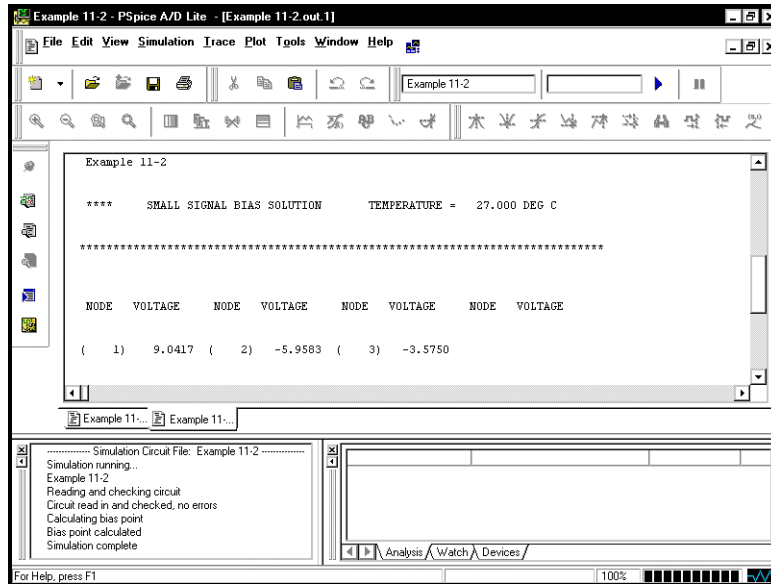
HINT: DC current sources must start with the letter I. The current enters the first node and leaves the second node.

SOLUTION: Create the netlist and run the simulation. Node numbers are assigned as shown above. The netlist is:

Example 11-2

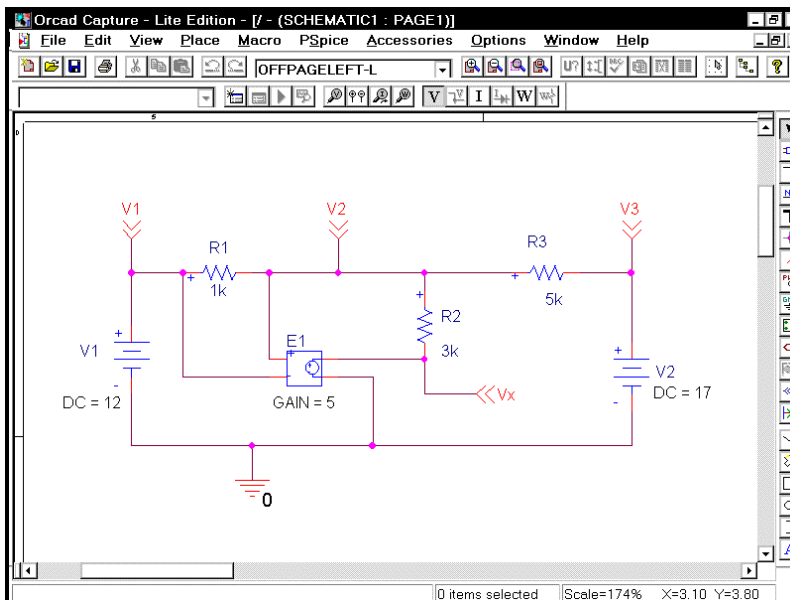
```

I1 0 1 DC 2m
R1 1 0 1k
V1 1 2 DC 15
R2 2 0 1k
R3 2 3 2.2k
R4 3 0 3.3k
.end
    
```



11.B. Nodal Analysis with Dependent Sources

Next we will redo the circuit of Section 3.B using netlists. The circuit is shown below:



The nodes of this circuit have been given text labels, and we will use the same labels here. Note that the nodes are named V1, V2, V3, and Vx. The new component in this circuit is a voltage dependent voltage source. The syntax for the netlist line of this type of source is:

1. The name must start with the letter E. Thus, in our circuit E1 is a good choice.
2. The first two nodes specify the nodes to which the source is connected. The first node is the positive reference, the second node is the negative reference.
3. The second two nodes sense the voltage. The first node is the positive reference, the second node is the negative reference.
4. The last thing specified is the gain of the dependent source.

For our circuit , the netlist line for E1 is

E1 Vx 0 V2 V1 5

The source is named E1. The source is connected between node Vx and ground. It senses the voltage between nodes V2 and V1, with V2 being the positive reference. The gain of the source is 5. The remainder of the components have been used previously. The complete netlist is

Nodal Analysis with Dependent Sources

*

* This is file Book Circuits\Netlists\Section 11B.cir

*

E1 Vx 0 V2 V1 5

V1 V1 0 DC 12

R1 V1 V2 1k

R2 V2 Vx 3k

R3 V2 V3 5k

V2 V3 0 DC 17

.end

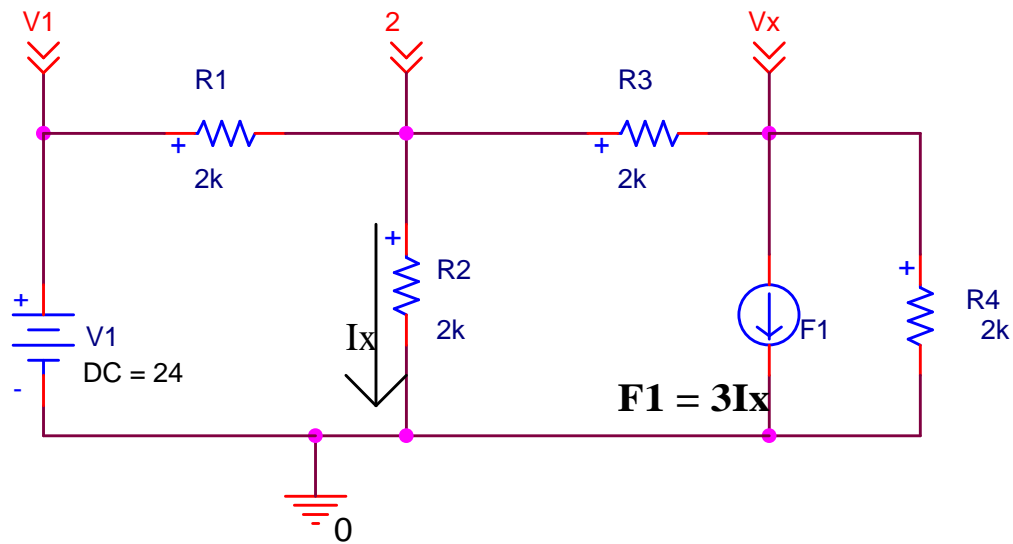
We can create the netlist using the same procedure shown in section 11.A. The results from the output file are have been copied and pasted below:

Nodal Analysis with Dependent Sources

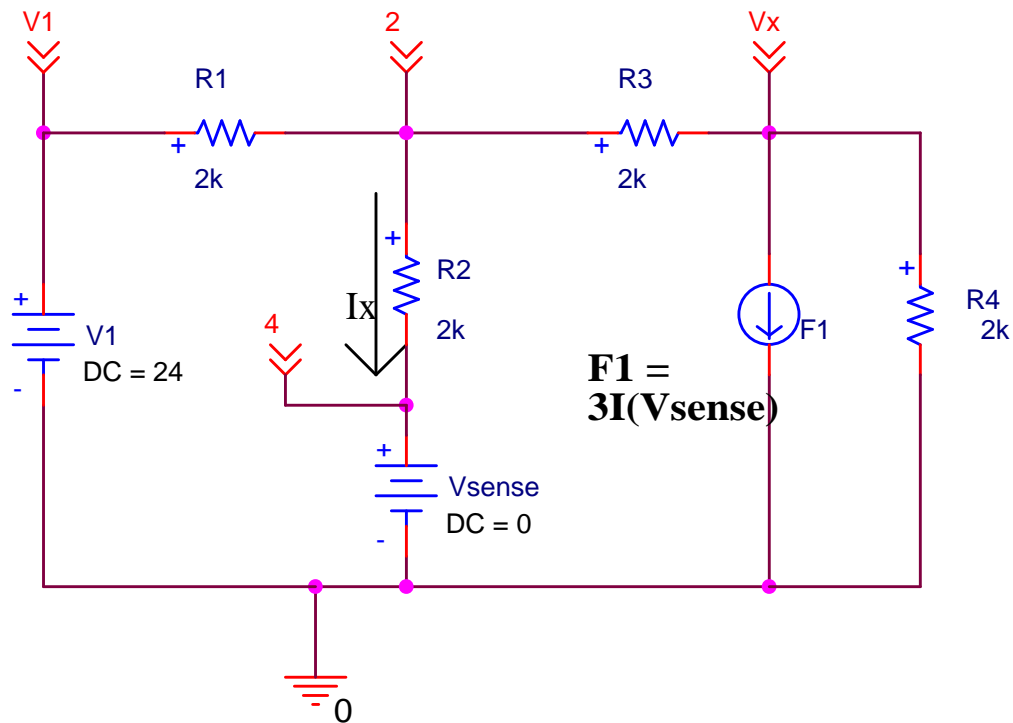
**** SMALL SIGNAL BIAS SOLUTION TEMPERATURE = 27.000 DEG C

NODE	VOLTAGE	NODE	VOLTAGE	NODE	VOLTAGE	NODE	VOLTAGE
(V1)	12.0000	(V2)	34.5000	(V3)	17.0000	(Vx)	112.5000

EXERCISE 11-3: Find the DC node voltages for the circuit below:



HINT: F1 is a current-controlled current source. The current through this source is 3 times the current I_x . To solve this circuit we need to be able to sense the current I_x . This is done by placing a zero-volt source in series with the element in which we want to measure the current. The circuit is modified as shown below:



Since Vsense has a voltage of zero, it does not affect the circuit. However, PSpice can only sense currents through voltage sources. Note that we have labeled the node between R2 and Vsense as node number 4. Since Vsense has a voltage of zero, the voltage of node 4 should also be zero volts. The syntax for a current controlled current source is:

1. The name must start with the letter F. Thus, in our circuit F1 is a good choice.
2. The first two nodes specify the nodes to which the source is connected. Current enters the source at the first node and leaves the source at the second.
3. The name of the voltage source through which current is measured.
4. The last thing specified is the gain of the dependent source.

The netlist for the circuit is

Example 11-3

*

```

F1 Vx 0 Vsense 3
V1 V1 0 DC 24
R1 V1 2 2k
R2 2 4 2k
Vsense 4 0 DC 0
R3 2 Vx 2k
R4 Vx 0 2k
.end

```

The results from the output file are shown below:

```

**** 09/07/01 11:48:18 ***** PSpice Lite (Mar 2000) *****

```

Example 11-3

```

****      SMALL SIGNAL BIAS SOLUTION      TEMPERATURE = 27.000 DEG C

```

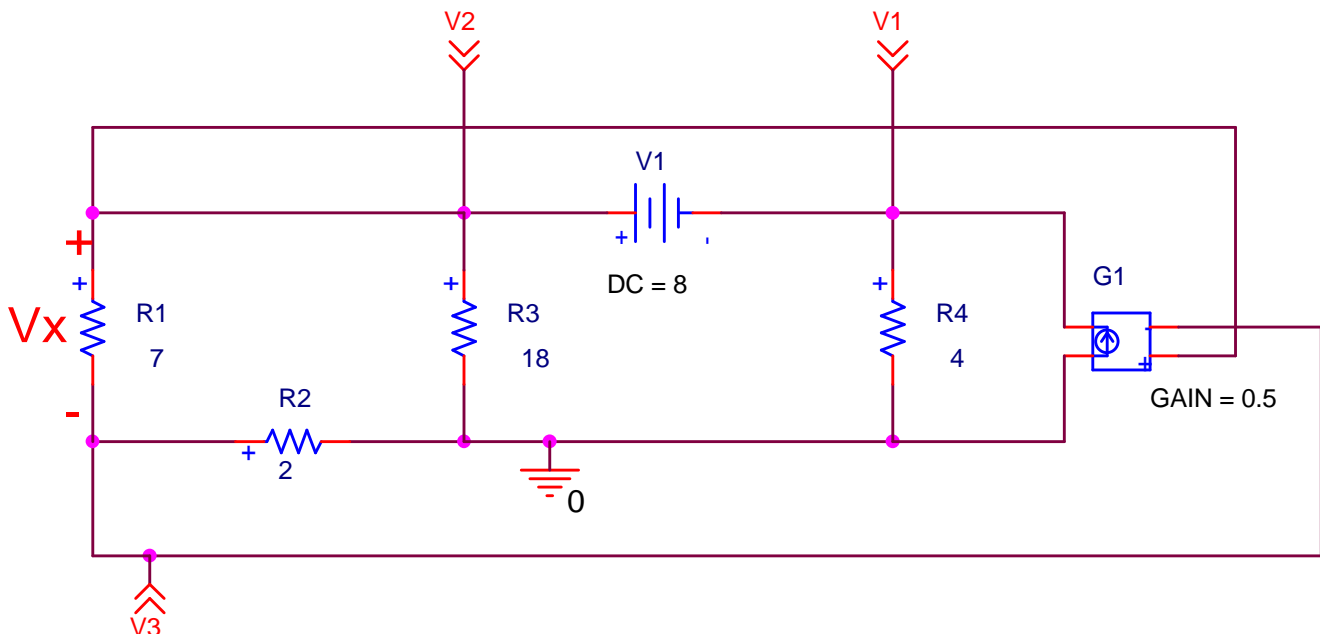
```

*****

```

NODE	VOLTAGE	NODE	VOLTAGE	NODE	VOLTAGE	NODE	VOLTAGE
(2)	6.0000	(4)	0.0000	(V1)	24.0000	(Vx)	-6.0000

EXERCISE 11-4: Find the DC node voltages for the circuit below:



HINT: **G1** is a voltage-controlled current source. The current through G1 is 0.5 times the voltage V_x . Note that V_x is the voltage at node V2 minus the voltage at node V3. It is not necessary to add the text V_x to your circuit. The syntax for this source is:

1. The name must start with the letter G. Thus, in our circuit G1 is a good choice.
2. The first two nodes specify the nodes to which the source is connected. Current enters the source at the first node and leaves the source at the second.
3. The second two nodes sense the voltage. The first node is the positive reference, the second node is the negative reference.
4. The last thing specified is the gain of the dependent source.

SOLUTION: The netlist for the circuit is:

Example 11-4

*

```
G1 0 V1 V2 V3 0.5
R1 V2 V3 7
R2 V3 0 2
R3 V2 0 18
R4 V1 0 4
V1 V2 V1 DC 8
.end
```

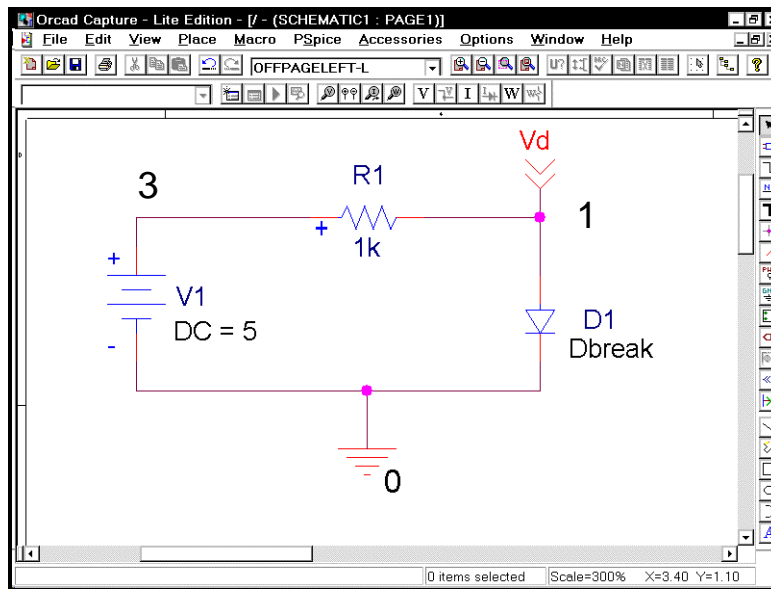
The results from the output file are:

Example 11-4

```
****          SMALL SIGNAL BIAS SOLUTION          TEMPERATURE = 27.000 DEG C
*****
NODE   VOLTAGE   NODE   VOLTAGE   NODE   VOLTAGE   NODE   VOLTAGE
(  V1)  64.0000  (  V2)  72.0000  (  V3)  16.0000
```

11.C. Diode Current and Voltage

Next we will redo the circuit of Section 3.C using netlists. The circuit is shown below:



We have assigned the node between V1 and R1 the label 3. In the original circuit, the node between R1 and D1 was called Vd. In this circuit, we need to use a .print statement which requires nodes to be labeled with numbers. Thus, we will label the node between R1 and D1 as node number 1.

When we use netlists, we do not need to use a part called Dbreak. This part is used to make the creating of diode models easier in the schematic capture front end. However, to create the simple diode model we need for this circuit, all we do is add a .model line in netlist.

The syntax of the netlist line for a diode is:

1. The name of the part in the schematic, D1 in this example.
2. The anode node of the diode.
3. The cathode node of the diode.
4. The name of the diode model. In this example, the model name will be Dx.

The netlist line for this diode is

D1 1 0 Dx

Note that the diode model must be defined in the netlist or somewhere in a library specified by the netlist. Here we will create a line in the netlist that defines the model. For this model, all we need to define is the parameter Is. The model line for this model is:

.model Dx D (Is=1e-15)

Although the bias point simulation calculates currents and voltages for all elements in the circuit, all currents are not displayed in the output file. To display this information, we need to run a DC simulation at a single point. To do this, we will use the list version of the DC statement:

.DC V1 LIST 5

This line specifies that a DC analysis should run and that V1 should be set to a single value of 5 volts. Although this is redundant because V1 is already specified in the circuit, we need to run a DC analysis to view the results, and for a DC analysis we need to specify a source or a temperature.

The last thing we need to do is print the results of the analysis. We do this with a .print command.

.print DC V(1) I(D1)

This line states to print the information from the DC analysis, and specifies the information to print. In this case we are printing the voltage a node 1 relative to ground and the current through D1.

The complete netlist for this circuit is

Diode Voltage and Current

```

D1 1 0 Dx
V1 3 0 DC 5
R1 3 1 1K
.model Dx D (Is=1e-15)
.DC V1 LIST 5
.print DC V(1) I(D1)
.end

```

Create the netlist and run PSpice as we did in the previous examples. The results are saved in the output file:

```

**** 09/09/01 15:22:01 ***** PSpice Lite (Mar 2000) *****
Diode Voltage and Current
****          DC TRANSFER CURVES          TEMPERATURE =  27.000 DEG C
*****
V1          V(1)          I(D1)
5.000E+00   7.521E-01   4.249E-03
          JOB CONCLUDED
          TOTAL JOB TIME          .02

```

We see that for $V1 = 5$ volts, the diode voltage (the voltage between node 1 and ground) is 0.7521 volts and the current through the diode is 4.249 mA. This result is slightly different than the result from the schematic simulation in section 3.C because, in that simulation, we specified the diode series resistance as 0.1 ohm.