# OrCad Capture

# Release 15.7

Robert J. Hofinger
Purdue University
1/15/08

# Table of Contents

# Part I - Capture

## 1. Introduction to PSpice



## 2. Getting Started with Orcad Capture CIS, Release 15.7



## 3. DC Simulations



## 4. AC Simulations



## 5. Digital Simulations



## 6. Components

# Part II - Allegro

## 1. Introduction to OrCad PCB Editor

## 1: Introduction to PSpice

In the past, students traditionally verified their laboratory electronic circuits by building them on breadboards and measuring the various nodes with the appropriate laboratory equipment. By using a computer simulation program, such as PSpice, students can obtain results before they come to lab. Hence the laboratory experiments become reinforcement to the subject matter at hand.

The use of a computer simulation program allows the student to easily subject the circuit to various stimuli (such as input signals and power supply variations) and to see the results in either a tabular format or plotted out graphically using PSpice's post processor called Probe.

## An Outline of PSpice

PSpice simulates the behavior of electronic circuits on a digital computer and tries to emulate both the signal generators and measurement equipment such as multimeters, oscilloscopes, curve tracers, and frequency spectrum analyzers.

## Types of Analysis Performed by PSpice

PSpice is a general-purpose circuit simulator capable of performing four main types of analysis: Bias Point, DC Sweep, AC Sweep/Noise, and Time Domain (transient).

Bias Point
The **Bias Point** analysis is the starting point for all analysis. In this mode, the simulator calculates the DC operating point of the circuit. Options include calculating the detailed bias points for all non-linear controlled sources and semiconductors (.OP), performing sensitivity analysis (.SENS), and calculating the small signal DC gain. (.TF)

DC Sweep
The **DC Sweep** analysis varies a voltage source over a range of voltages in an assigned number of increments in a linear or logarithmic fashion.

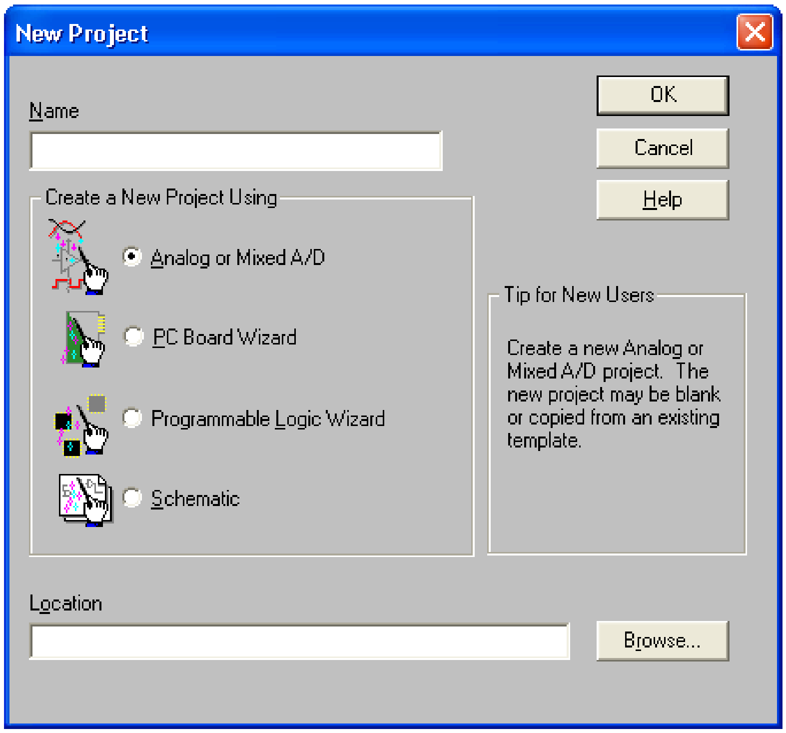
AC Sweep/Noise
The **AC Sweep/Noise** analysis varies the operating frequency in a linear or logarithmic manner. It linearizes the circuit around the DC operating point and then calculates the network variables as functions of frequency. The start and stop frequencies as well as the number of points can be assigned. Spice will compute the effective noise voltage spectral density that appears at the *Output Voltage* node because of internal noise sources (.NOISE). In this analysis the detailed bias points for all non-linear controlled sources and semiconductors (.OP) can also be performed.

Time Domain (transient)
The **Time Domain (transient)** analysis is probably the most popular analysis. In this mode, you can plot the various outputs as a function of time. The starting and ending times for the various plots can be input. The accuracy (smoothness) of the output plots can also be controlled by regulating the maximum (time) step size.

## 2: Getting Started with Orcad Capture CIS, Release 15.7

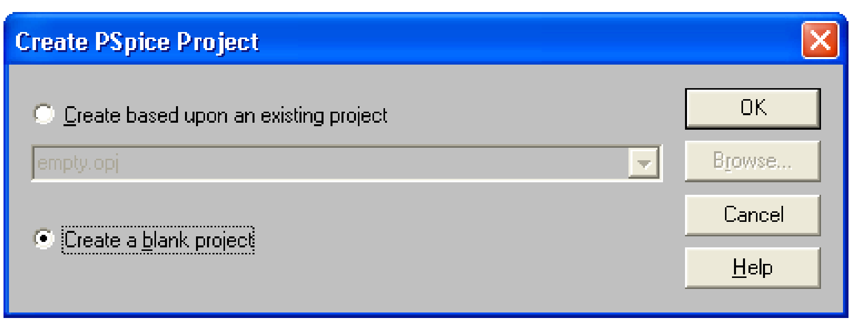
You start a new project (program) by going to the **File** menu in the upper left corner, then **New**, and then **Project**.

The following screen will appear. Be sure that the **Analog or Mixed A/D** button is activated. (see figure below) Change it if necessary. This is **VERY** important.

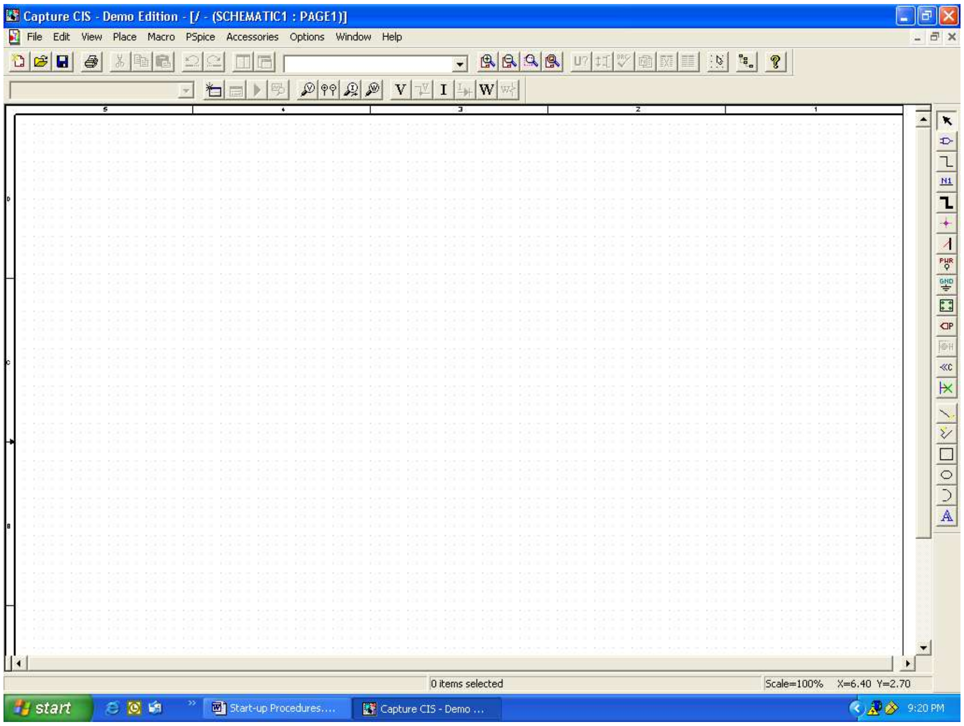



You will need to fill in the top line Name with a file name (use Start-Up Example) and then the bottom line **Location** with the path name. This is the directory where you will be storing your "Project".

Now the following screen will appear. Since you are starting a new project, change the button settings as shown below. Activate the **Create a blank project** button and left-click OK.

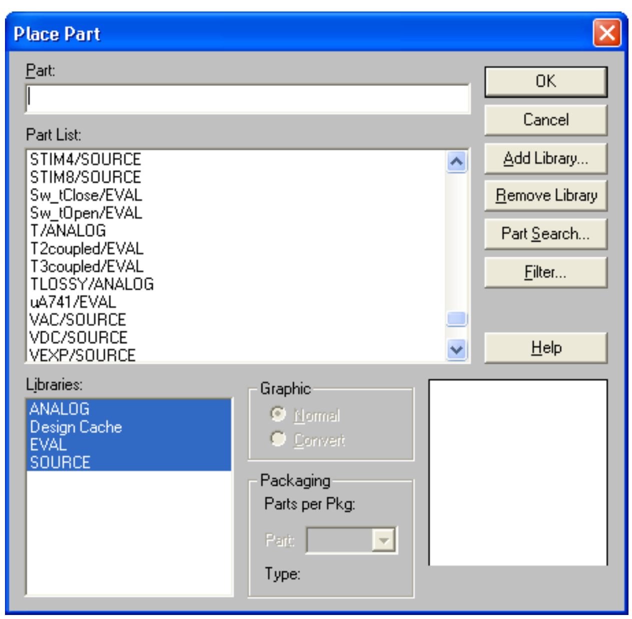


Now you should come up to a blank schematic entry screen.




You can now start adding components and symbols to your schematic, by using the **Place**, **Part…** menu sequence, or the special icon (the uppermost one) on the right hand toolbar.

The following screen will appear.

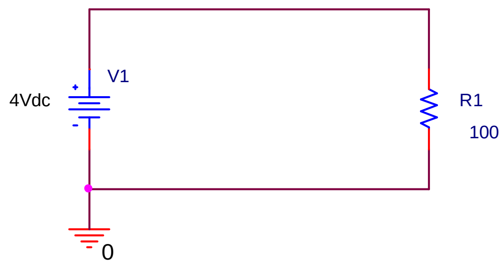


If all of the Libraries shown do not appear on your screen, and they probably won't, go to **Add Library**. There you will find a list of available libraries. For this first example, you will need the analog.olb, the eval.olb, and the source.olb libraries. Add them now.

**Note:** that only parts from the Libraries that are highlighted are shown in the **Part List** window.

At this time, highlight all of the libraries. Then start entering your parts. When you have found the required part, either by entering its name in the **Part** window or by highlighting its name in the **Part List** window, left-click OK to place the part onto the schematic. You can continue left-clicking to place multiple copies of the same part or right click to end this selection.

Practice now by entering the schematic shown below. Change the default values and orientations to those shown below.




To change a value, or a reference, highlight the appropriate value (left-click) and then double left-clicking. When you have added the resistor (R), and the power supply (Vdc) symbols, enter the ground symbol labeled "0", which is located in the "..../PSpice/source.olb" library. Recall that every circuit has to have a node "0". Left-click Apply and close the page.

You can rotate parts by highlighting the part (left-click) and bringing up the part menu (right click), or by pressing the "r" key on the keyboard. See **PSpice Component Layout** description on the next page.

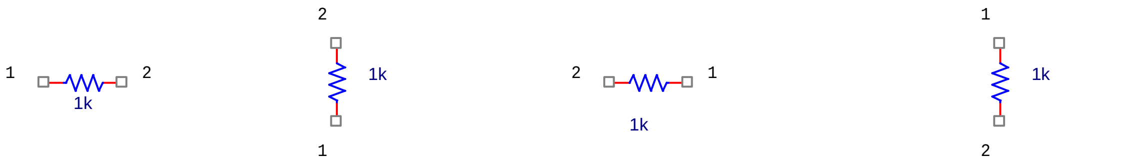
Now its time to add the connecting wires.

Use the **Place, Wire** menu sequence or the icon on the right hand side toolbar. (second one from the top) Connecting wires requires that you drag the "cross hair" over the end of the part and left-click. This "solders" one end of the wire. Drag the wire to another connecting point and left-click again. You have now "soldered" the other end.

You are now ready to simulate your circuit.

## PSpice Component Layout

All two leaded passive components have an implied “1” end and a “2” end.  Whenever you place a component, it takes a default position, for example, a resistor, capacitor, or inductor will take a default position with its “1” end to the left as shown in (a).  A component may be rotated by activating it, then right-clicking and selecting **Rotate**, or by typing the letter “r” (see example b). Each rotation moves the component counterclockwise by 90º.  To get the “1” end facing up, you must rotate the component 3 times from its default position as indicated in (c)



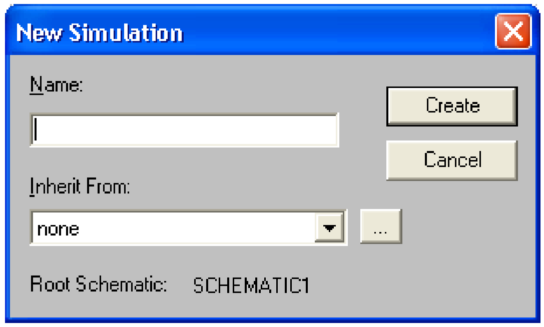

(a) Default    (b) First rotation (90 deg)    (c) Second rotation (90 deg)    (d) Third rotation (90 deg)

PSpice utilizes the implied “1” and “2” ends for its handling of current directions and voltage polarities, for example, it represents current as entering a device from its “1” end and leaving its “2” end, and it represents a positive voltage at its “1” end with respect to its “2” end.  Knowing about component layout is important when you are viewing your results in Probe and especially important when setting up initial conditions.  For example if you set a capacitor’s initial voltage to 10V, PSpice will place 10 volts across the capacitor with its “1” end positive with respect to its “2” end.

If you have placed the device in your circuit upside down, its polarity will be reversed from what you expect.  If this happens, disconnect its wiring, rotate it 2 times (to get it to the desired direction) rewire, and reset any initial conditions.

## DC Bias Simulation

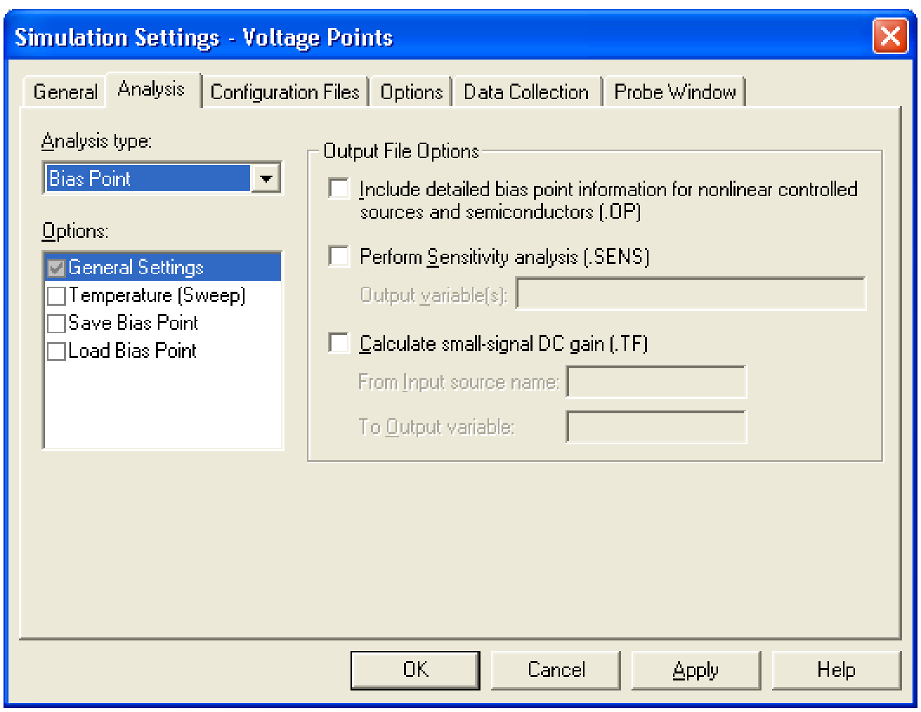
To start the simulation process, open the **PSpice** menu. The first choice available is **New Simulation Profile**. Left-click on it and the following window will appear.




Give the New Simulation a **Name**. For now use *"Voltage Points"*

*Note: You could have done the same thing by left-clicking on the*  *button on the toolbar.*

Left-click Create and the next screen will appear

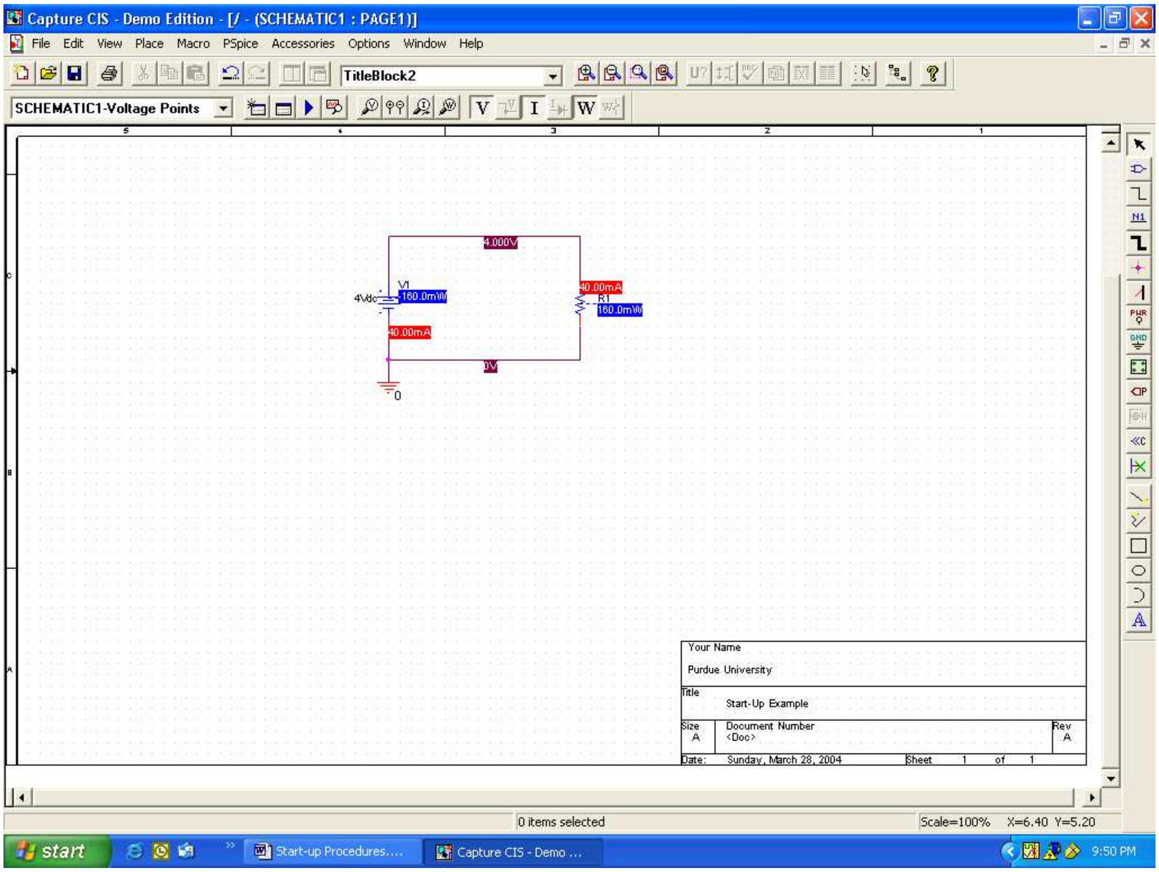



For a DC analysis, select the **Bias Point** setting in the **Analysis type:** window. Since we do not need that process in this part of our example, go to the **Probe Window** tab, uncheck the box next to the **Display Probe Window** setting and then left-click OK.

Now you are ready to **Run** a simulation.

Go to the **PSpice** tab and select **Run**.

The simulation window will appear. When the simulation has completed, close this window and the schematic will reappear. When the V, I, and W tool buttons are activated, the results of the voltage, the current , and the power dissipated in that component will be shown. The tool buttons alongside the V, I, and W buttons allow you to alternately toggle a highlighted value OFF and ON



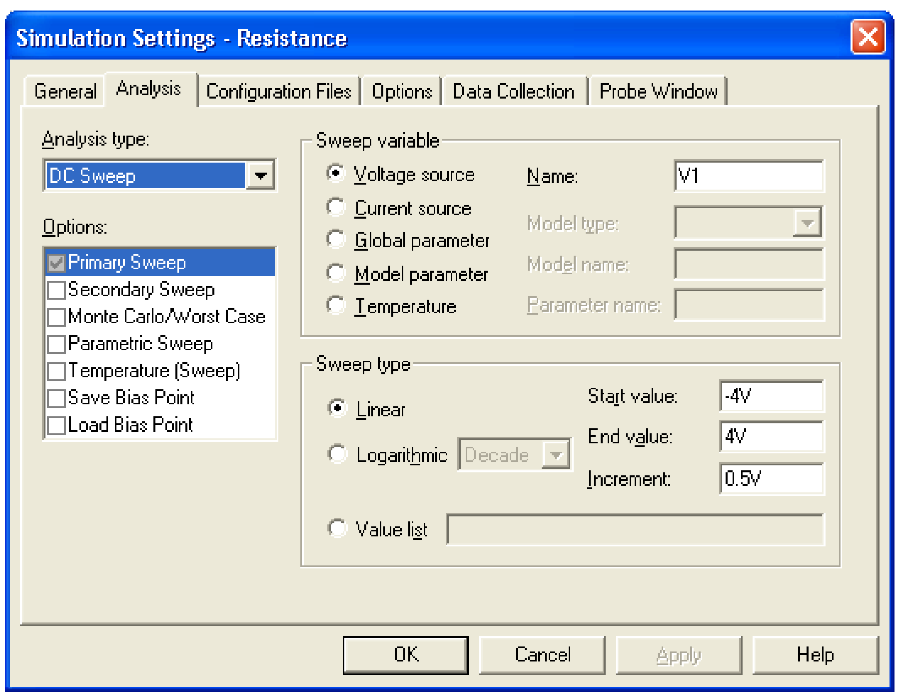
## Linear Resistance

In this segment you will plot out the current vs. voltage characteristics of a linear resistor.

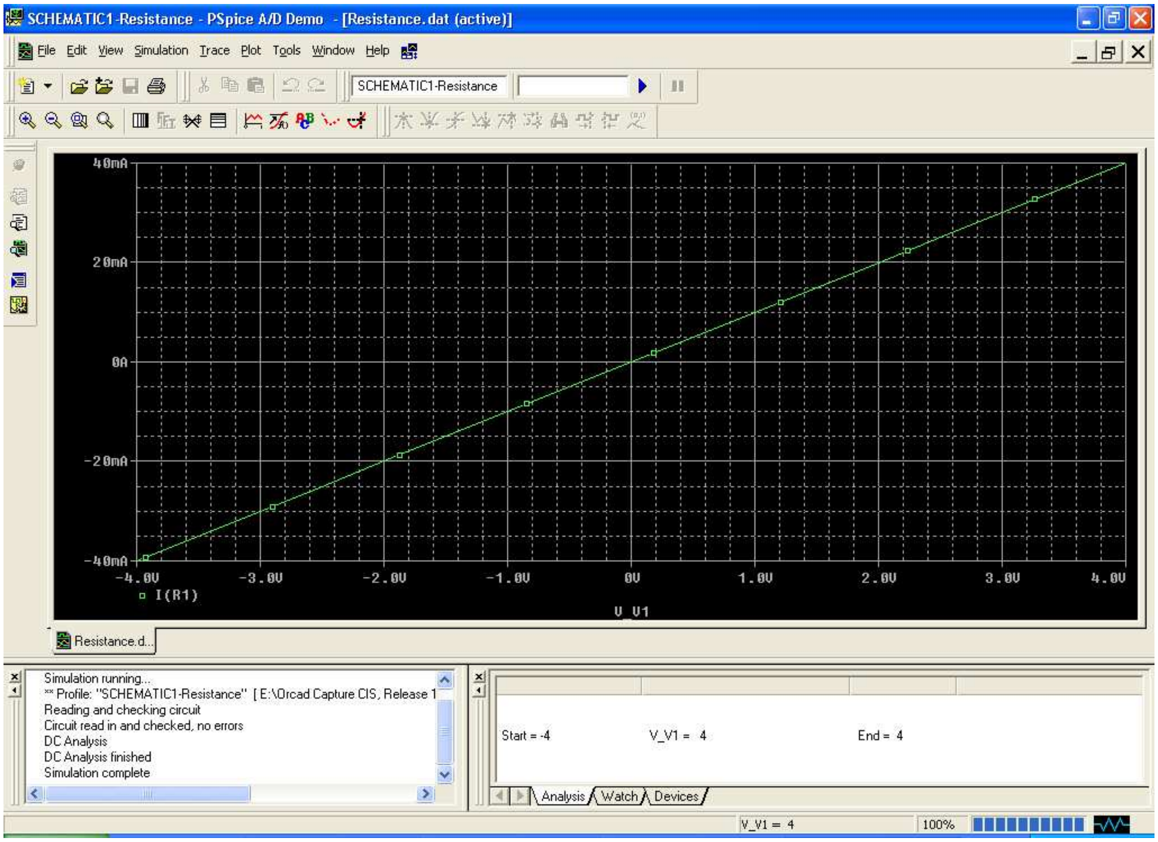
The resistor network is repeated below.


As before, go to the **PSpice** tab, or the  button, and create a **New Simulation Profile**. This time give it the **Name** "*Resistance*"

The simulation profile setup for a DC Sweep is shown below.




Click OK to apply the values and close this window. Now re-run PSpice. When the circuit is finished simulating, the Probe window will appear. The X-AXIS will already be set up with a scale of -4V to +4V in increments of 1V. Left-click on the **Trace** menu or on the toolbar menu. This brings up the **Add Traces** window. Highlight and left-click I(R1) to add it to the **Trace Expression** line at the bottom of the screen. Left-clicking $\boxed{\text{OK}}$ brings up the plot of the resistor current vs. the resistor voltage.

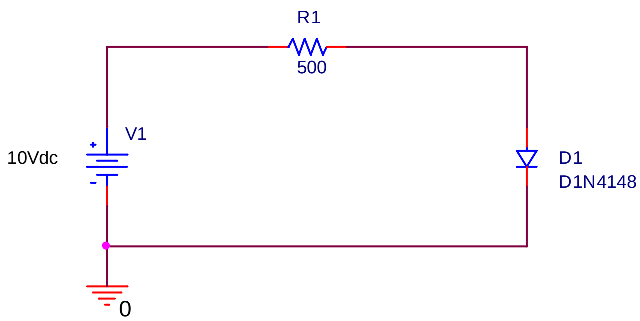



If your resistor "curve" is negative that means your resistor is "backwards" in the circuit. Disconnect the resistor, rotate it 180° and reconnect the wires. Re-simulate the circuit to get an image similar to that shown above.

## Non-Linear Resistance

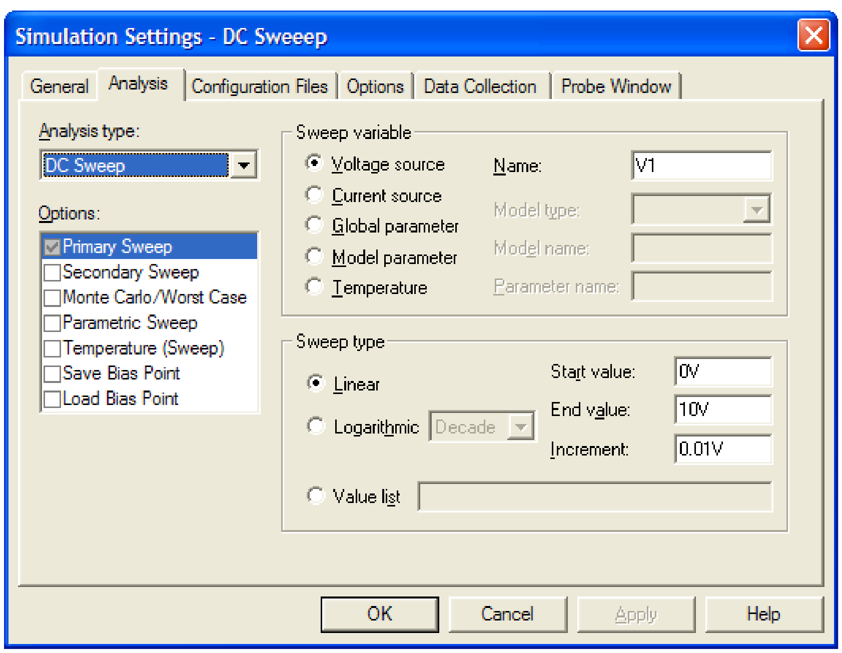
In this segment you will plot out the current vs. voltage characteristics of a non-linear resistive device, specifically the D1N4148 diode.

The diode network is shown below.

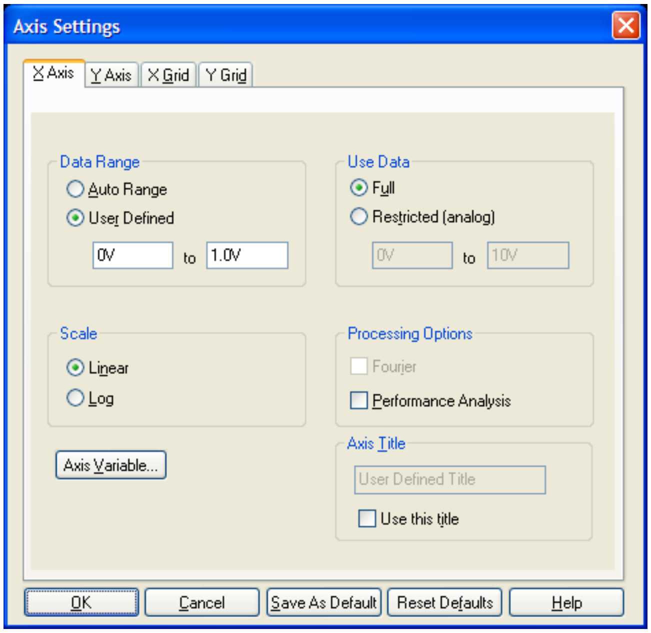



Re-wire your circuit, change the values of the components, and add a 1N4148 diode.

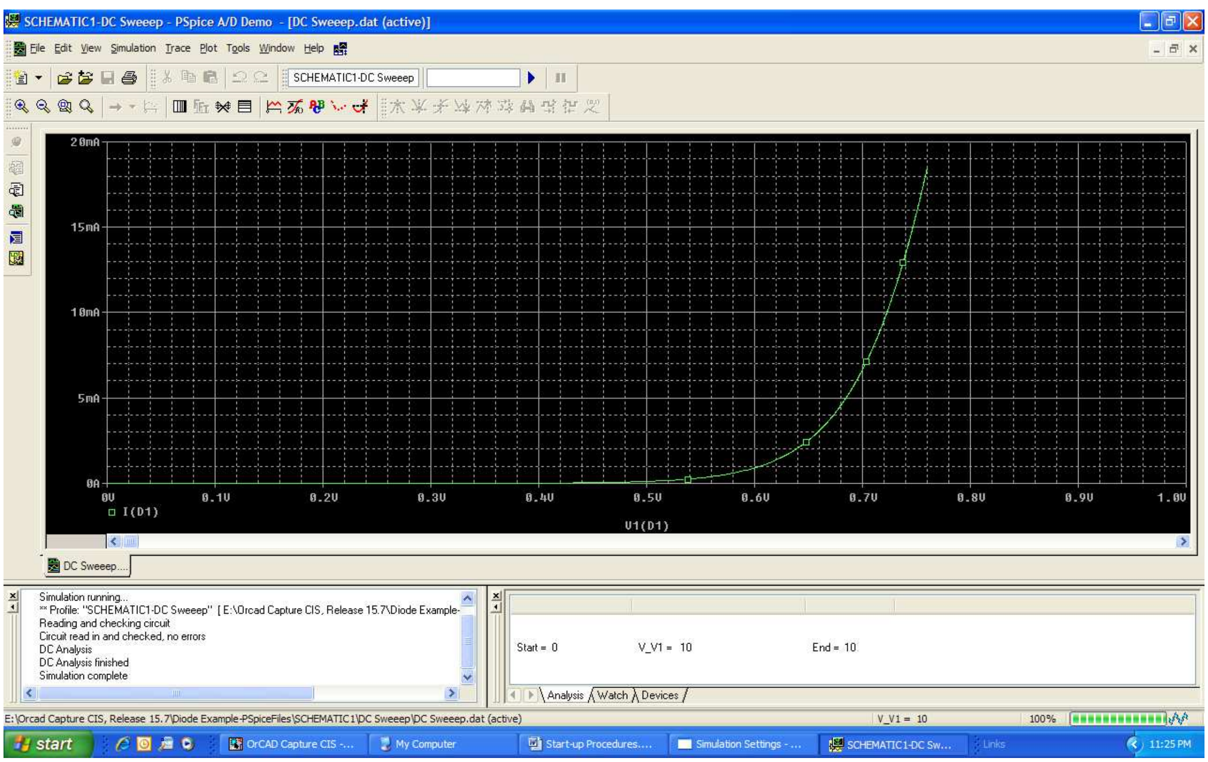
Again create a **New Simulation Profile**. The simulation profile set up for a DC Sweep is shown below.




Click OK to close this window and run PSpice. When the circuit is finished simulating, the Probe window will appear. At that point, you will want to change the x-axis from V_V1 to V1(D1). Pull down the Plot menu and click on the Axis Settings… option. This will bring up the following menu.




Click on the Axis Variable button and then choose V1(D1). After you have chosen the x-axis the above window will reappear. Now set the Data Range to User Defined and adjust the settings to what you prefer. In the above example 0V to 1.0V was selected. When you close this window, you can now select your Y-axis trace. Use the I(D1) selection to plot the $I_D$ vs. $V_D$ diode characteristics.

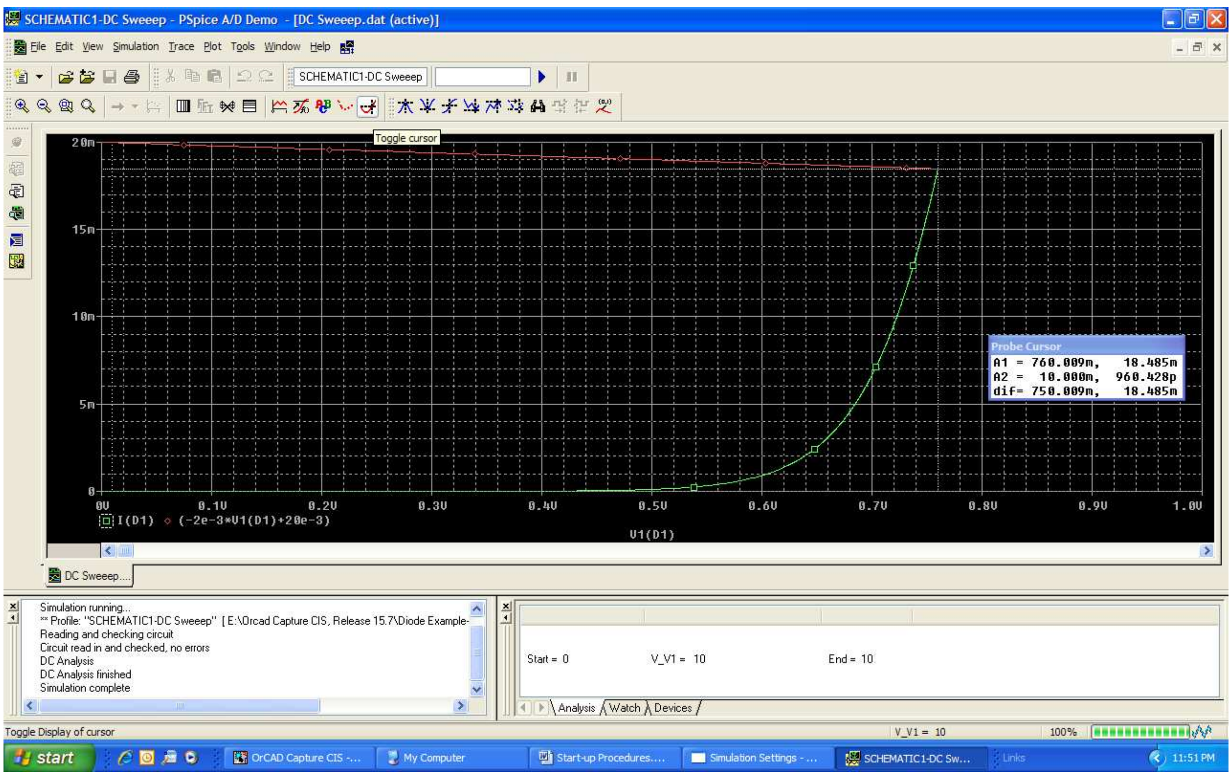


## Operating Point

In the previous section we used the variable voltage supply, V1, to plot the characteristics of a 1N4148 diode. Let us now add the load line to the above plot to find the operating point, commonly called the "Q" point of the circuit.

Bring down the **Trace** menu and click on the **Add trace…** , or click the **Add Trace** icon on the toolbar. This brings up the **Add Traces** window. In the Trace Expression box, using the double brackets from the Analog Operators and Functions section on the right side of the window, include the following expression [-2e-3*V1[D1]+20e-3]. Left Clicking OK adds the second straight line, the load line, to the plot.

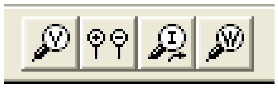
Activating the Toggle cursor Icon on the toolbar brings up the Probe cursor, along with the data point window. When the cursors are placed on the point of crossing of the two plots, the results can be seen on the first row. In this example V1(D1) = 760 mV and I(D1) = 18.485 mA.



## Markers

In the two previous examples we chose the parameters to be displayed on the Y-Axis in the **Add Traces** window.

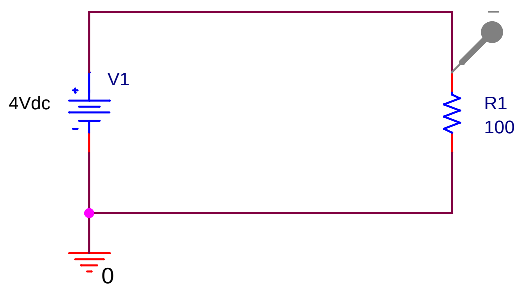
In **Probe**, you have the ability to preset which parameters you want to display. On the **Action toolbar**, there are four toolbar buttons that can be used for this purpose.



They are, from left to right: Voltage/Level Marker
Voltage Differential Marker(s)
Current Marker
Power Dissipation Marker

Placing one, or more, of these **Markers** on your schematic before running the Simulation automatically sets up the **Probe** display.

Below is an example of adding the **Current Marker**.



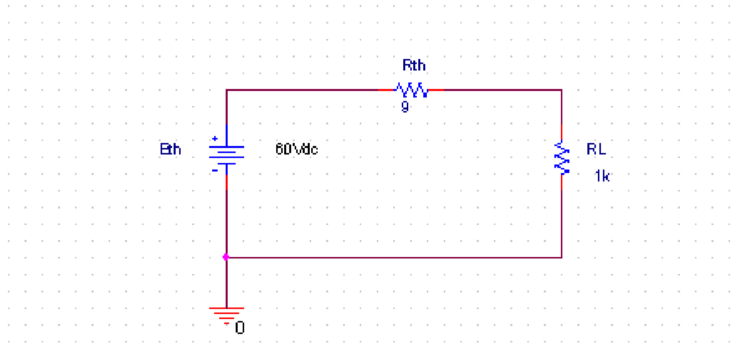

Now when you run your simulation, the Probe display with the result(s) will automatically appear.

## Parametric DC Sweep

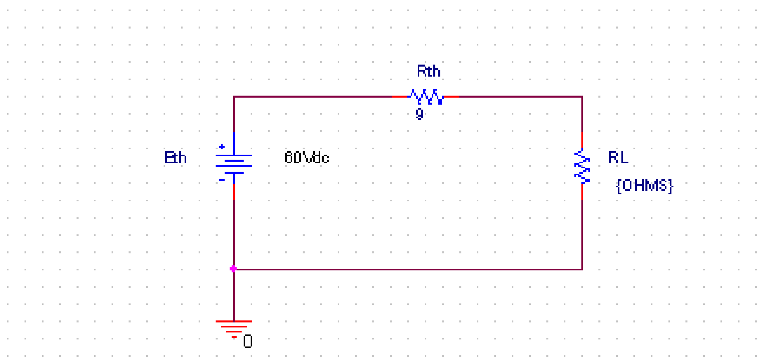
As an example of performing a parametric DC sweep, that is, varying the parameter or value of a component, let's find the power dissipated in a resistor as the value of that resistor varies and plot the results.

You will recognize that this is the Maximum Power Transfer curve.

Starting with the basic schematic:

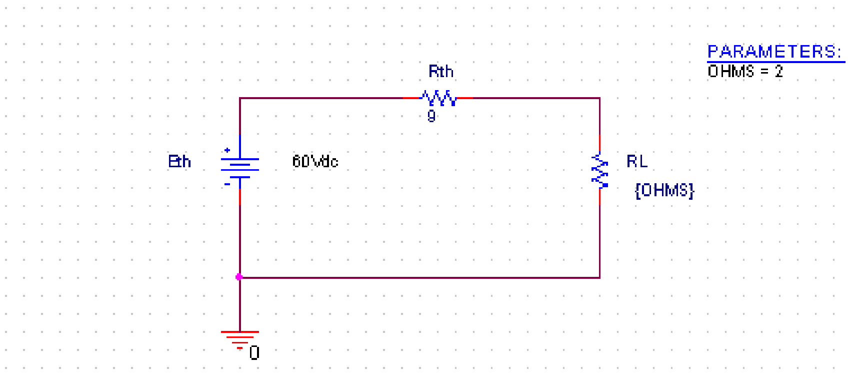



In order to do a sweep of the values of resistor RL, you need to change its value from a fixed number to a variable text. Say we call it {OHMS}

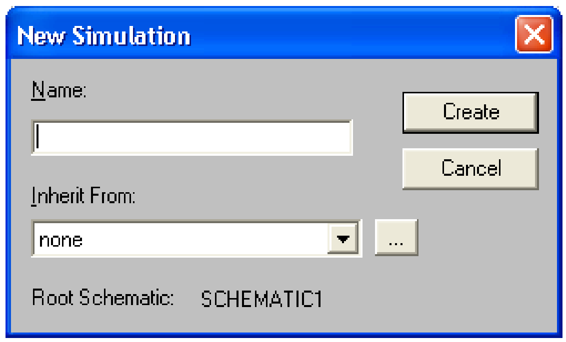



Note: the text OHMS is enclosed in curly brackets { }. OrCad will treat this as a parameter.

Now go to the Place Part menu, under the **SPECIAL** library, select a "part" called **PARAM**. If this library is not on your list, you will have to add it using the Add Library... button. It is called **special.olb** in the **pspice** folder. Now place this selected "part"**(PARAM)** near your "variable" load resistor. Double left-clicking on it, brings up the **Property Editor**. Left-click on the New Column button give the variable a name, in this case we are using the variable name OHMS, and a value (one that is reasonable for your design). This value will be used as the default condition if you choose to do another type of simulation on this design. Use the value 2. Apply it and then highlight this new column. Left-click Display... and then activate the button labeled **Name and Value**. When you close out the [Property Editor] your schematic should look like this:

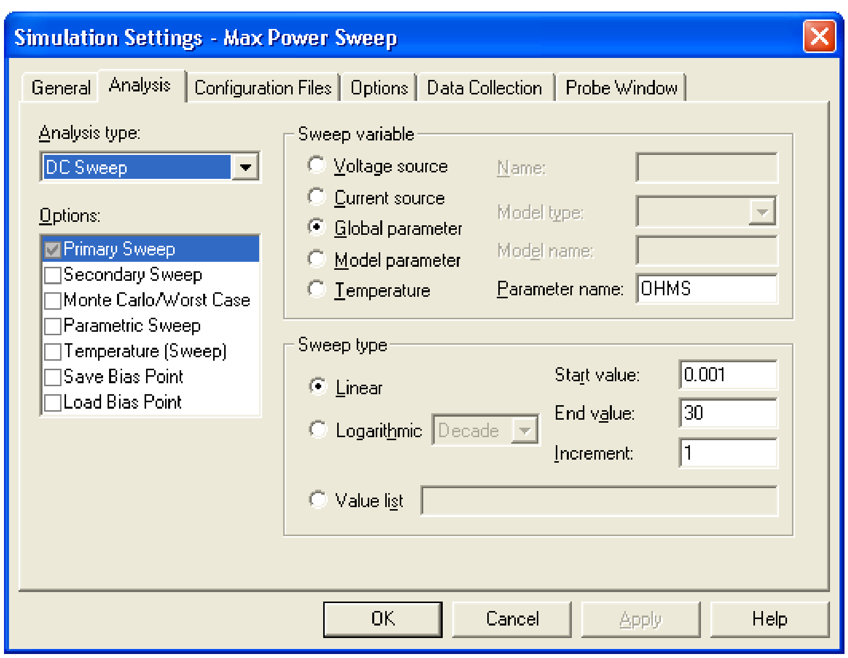



To start the simulation process, open the **PSpice** menu. The first choice available is the **New Simulation** Profile. Left-click on it and the following window will appear.



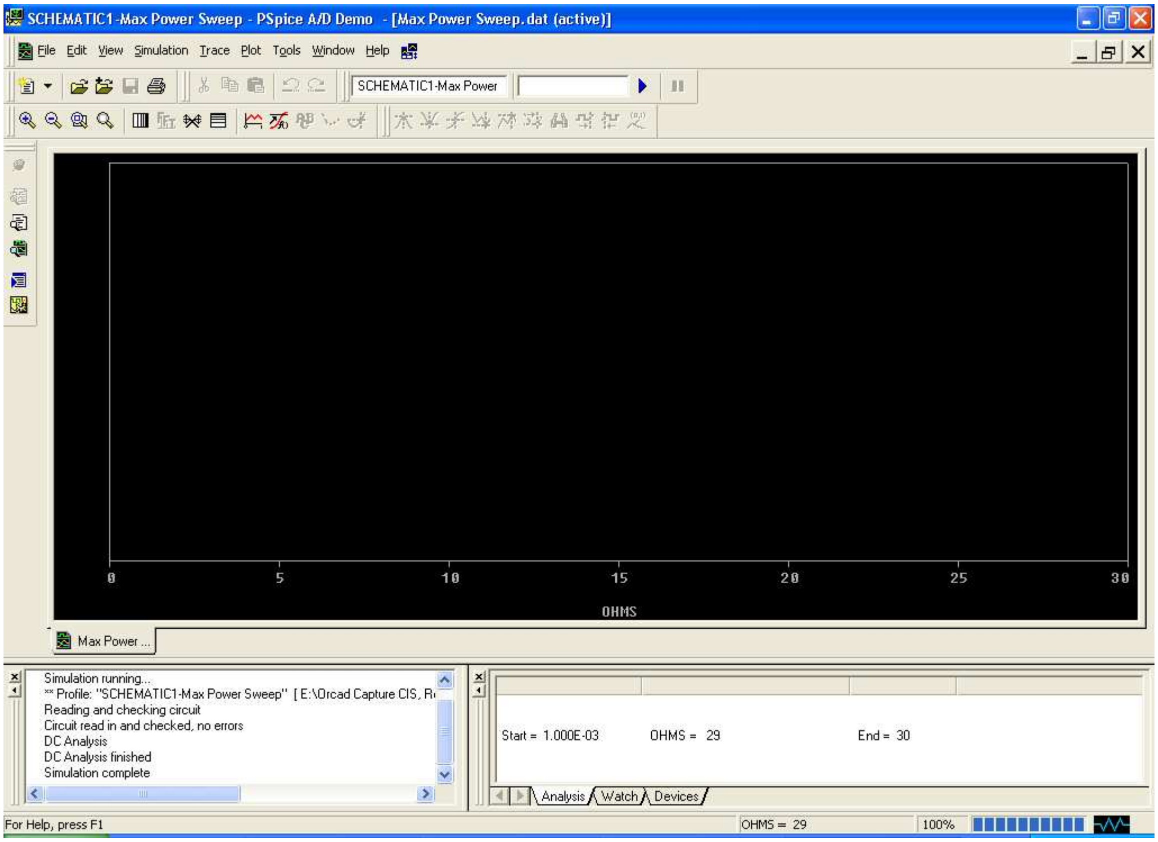

Give the New Simulation a **Name**. For now, use "*Max Power Sweep*".

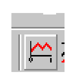
Left-click Create and the next screen will appear



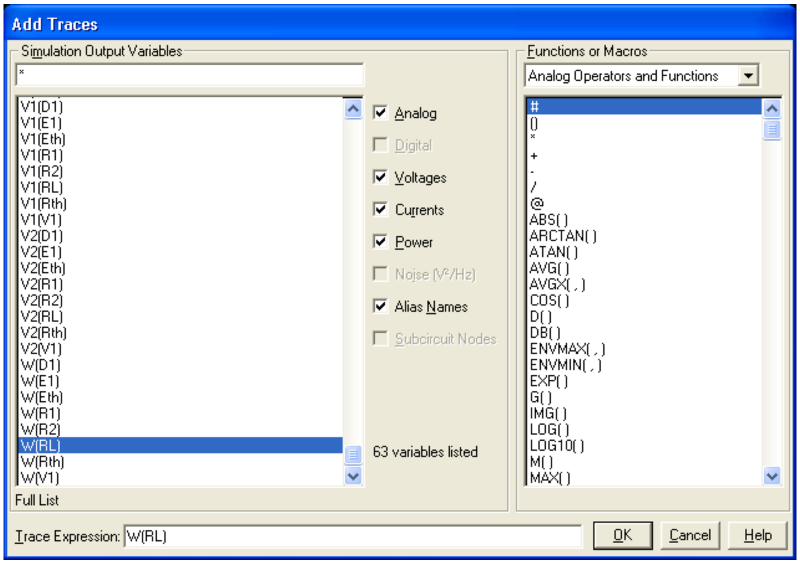
Activate the **Global parameter** button and add the name of your parameter, without the curly brackets, that you are sweeping. In this case it is OHMS. The sweep type will be **Linear**. Since you will want to sweep the resistor value from 0 Ω's to 30 Ω's in steps of 1 Ω you will enter these values in the appropriate places. Note that you could not enter a **Start value** of 0 Ω's since this would result in a divide by zero, which is illegal. If you try it you will get an error message. Go try it, run **PSpice** and see the results. Instead, you can use 0.001 Ω, a small number close enough to 0.

If everything was entered correctly, the simulation will run without any errors, and the following screen will appear:

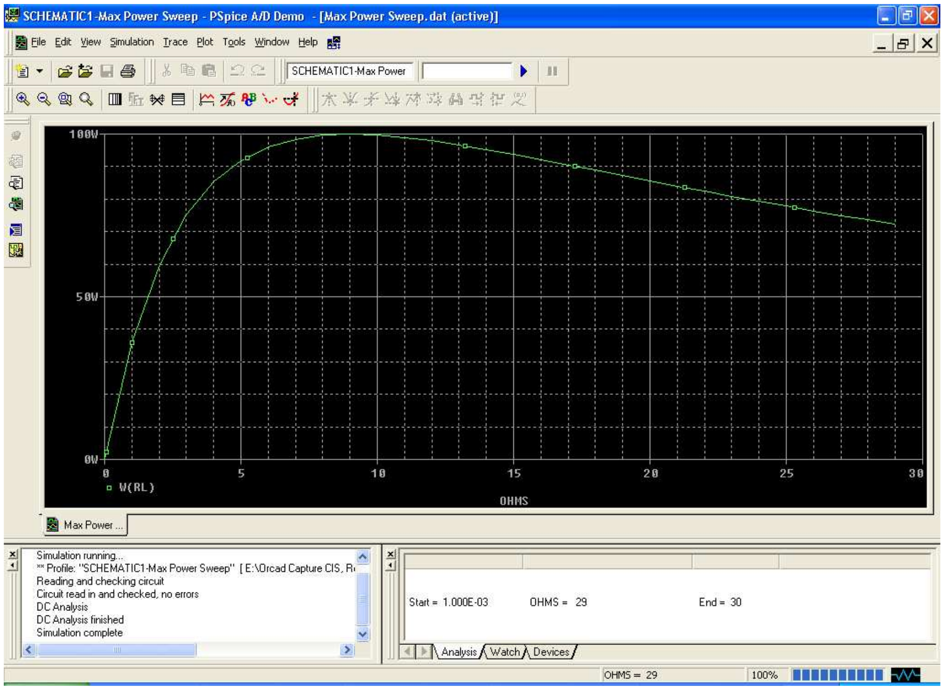



Add a trace by left-clicking on the **Trace** menu or on the toolbar button: 

This brings up the **Add Traces** menu. Since you are looking for the power dissipated in a resistor as a function of its value, highlight the W(RL).



Left-clicking $\boxed{\text{OK}}$ brings up the plot on the power vs. OHMS graph.

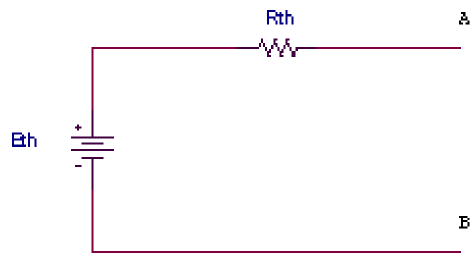


To confirm your result:

Left-click the **Toggle cursor** toolbar button:  Then left-click the **Cursor Max** toolbar button. (the leftmost one) The cursor will move over to the maximum point on the curve and in the **Probe Cursor** window, the result A1 = 9.0010 (9.0 Ω) and 100.000 (100 watts) will appear.

## Thévenin and Norton Equivalents

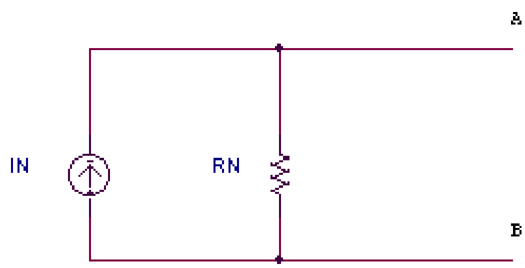
You can use the Bias Point simulation method to find the Thévenin and Norton equivalent circuits. As you recall, the Thévenin's Theorem states that any two-terminal bilateral dc network can be replaced by an equivalent circuit consisting of an ideal voltage source and a series resistance, as shown below.




**Thévenin Equivalent**

The voltage source $E_{th}$ is also the "open circuit" voltage or $E_{oc}$. The resistance $R_{th}$ is also the internal resistance of the circuit $R_{int}$.

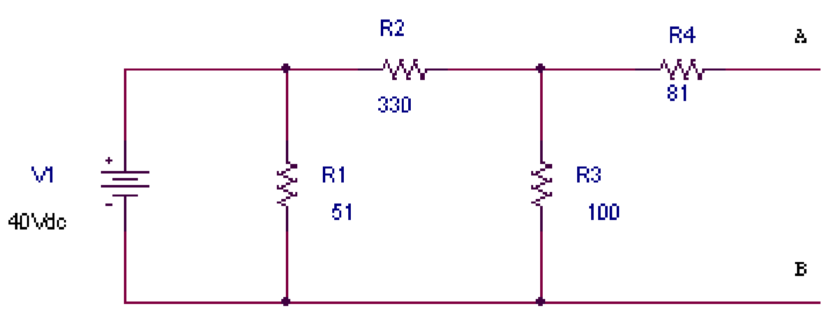
The Norton Equivalent circuit consists of an ideal current source and a series resistance.



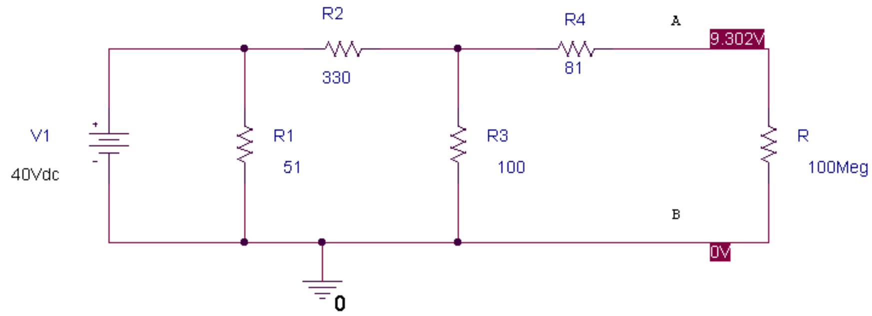

**Norton Equivalent**

The current source $I_N$ is also the "short circuit" current or $I_{SC}$ The resistance $R_N$ is also the internal resistance of the circuit $R_{int}$ (as above)

With this in mind, you need to solve, i.e., run the OrCad circuit simulation, PSpice, three times; once to find the "open circuit" voltage, once to find the "short circuit" current, and a third time to find the "internal resistance". Using the following circuit as an example

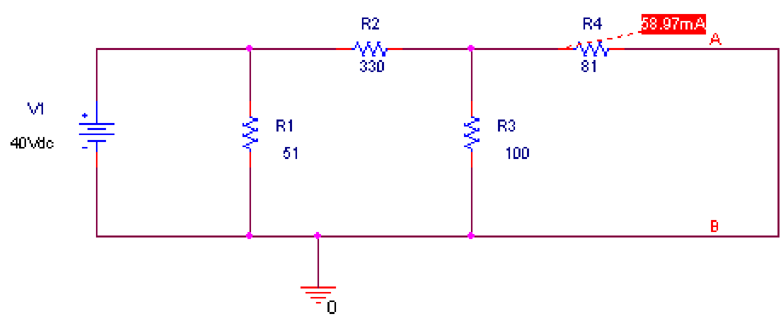


First, solving for the "open circuit" voltage, leads to $V_{OC}$ = 9.302 volts.

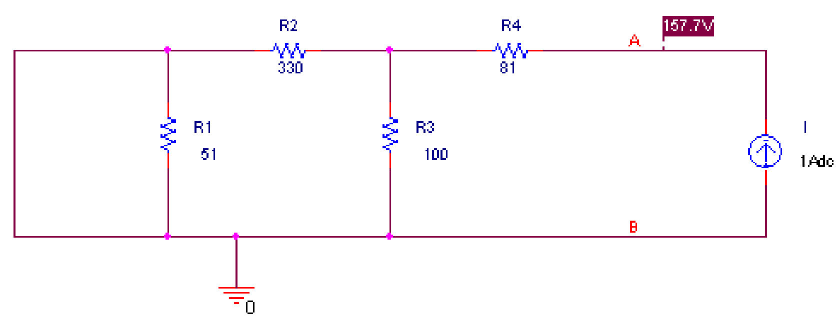



Note: the placement of the 100Meg Ω resistor across the output A-B. Since PSpice does not simulate properly with a "floating" component, you can simulate an open connection by inserting a very large resistor.

Second, solving for the "short circuit" current, leads to $I_{SC}$ = 58.97 mA.




Third, to solve for the "internal resistance", since PSpice cannot measure resistances directly, you have to do it indirectly. First set all internal sources to zero, i.e. voltage sources to "shorts" and current sources to "opens". Now connect a current source of 1 Adc to points A-B, the resultant voltage at points A-B will be the "internal resistance". In this example, $R_{int}$ = 157.7 Ω.




This confirms our previous simulations since $R_{int} = R_{th} = \dfrac{V_{OC}}{I_{SC}} = \dfrac{9.302\,\text{V}}{58.97\,\text{mA}} = 157.7\,\Omega$

## AC Inputs

Shown below are the five types of AC inputs, along with sample inputs:

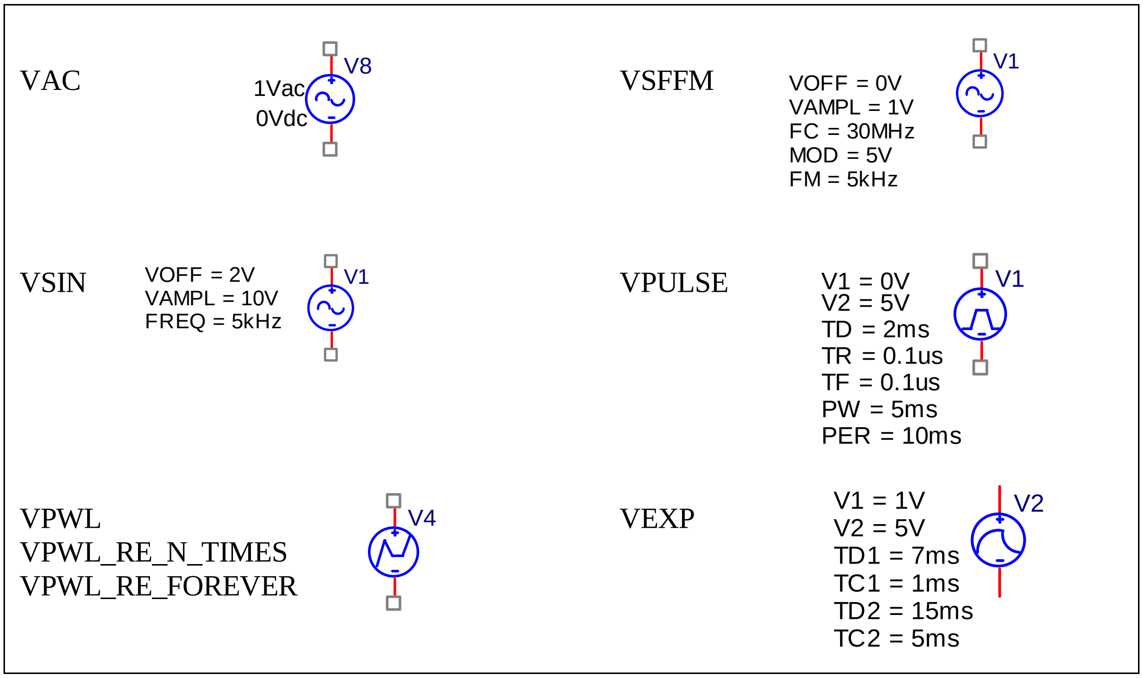



**TABLE OF AC INPUT STIMULI**

VAC - a sine wave input used for AC Sweep Analysis; that is for viewing an output as a function of frequency.

VSIN - a sine wave input used for transient analysis; that is for viewing an output as a function of time.

VSFFM - creates a Single-Frequency Frequency Modulation source.

VPULSE - creates a rectangular repetitive pulse having the parameters specified.

VPWL - an pulse input where you can create your own type of single pulse or triangular waveform. Used in a transient analysis simulation. Points on the piecewise linear source are entered in the ***Property Editor***.

VPWL_RE_N_TIMES - is the VPWL repeated N times.

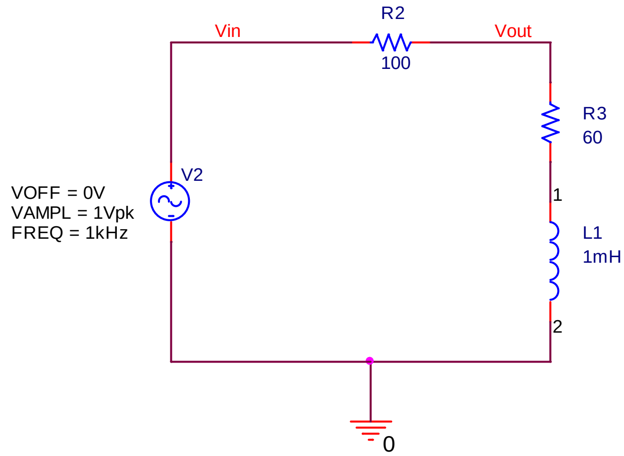
VPWL_RE_FOREVER - is the VPWL run continuously.

VEXP - Creates a single exponential waveform.

## Time Domain (Transient Analysis)

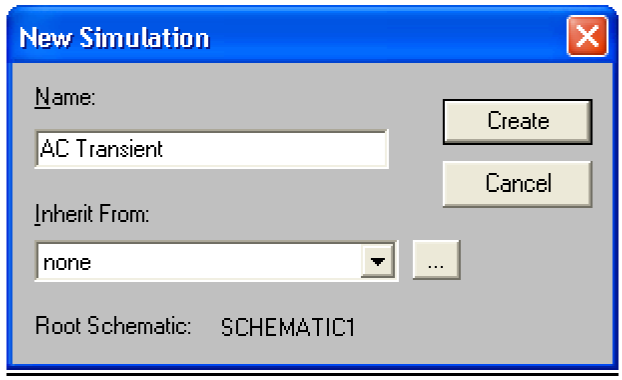
In the transient analysis mode simulation, you will be displaying the amplitude of an ac signal as a function of time.

The example series RL circuit, being driven by an ac source, is shown below. Use the **VSIN** component part for the voltage source V2.

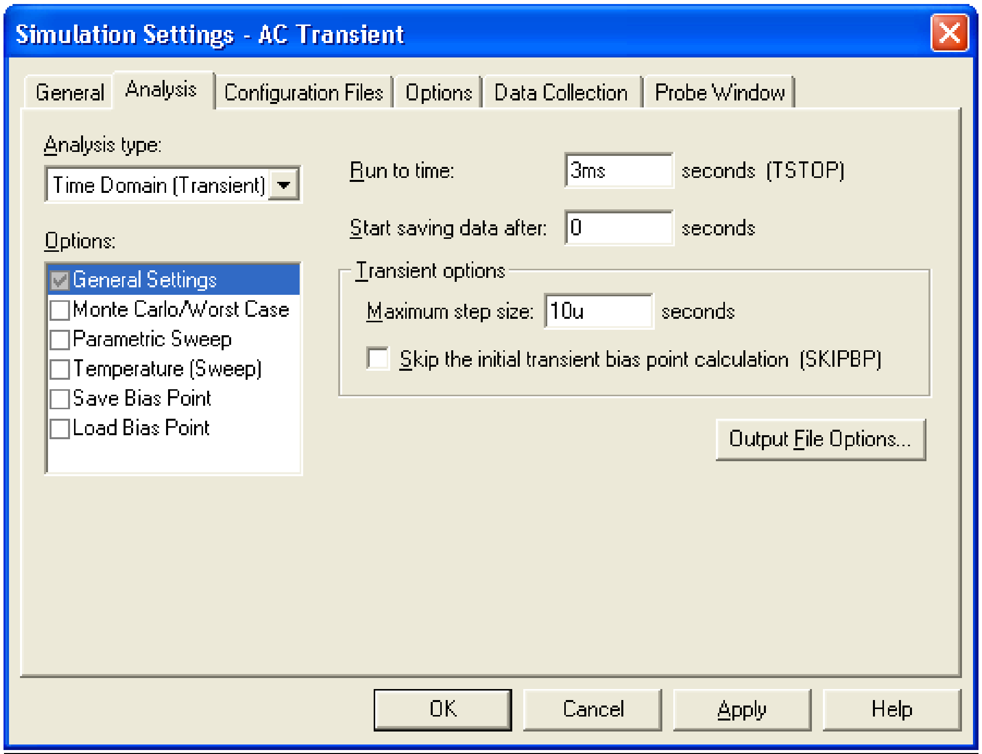



Note, any time you simulate a circuit with an inductor, it is a good practice to include a series resistance. The inductor model is an ideal model with no series resistance. When OrCad performs any type of AC simulation, it first does a DC Bias simulation to calculate the quiescent (DC) operating points. An inductor with no series resistance, i.e. a DC short circuit, could result in erroneous results.

Before we can simulate, we must create a Simulation profile. As before, open the **PSPice** menu tab, and create a **New Simulation Profile**. This time give it the **Name** *"AC Transient"*



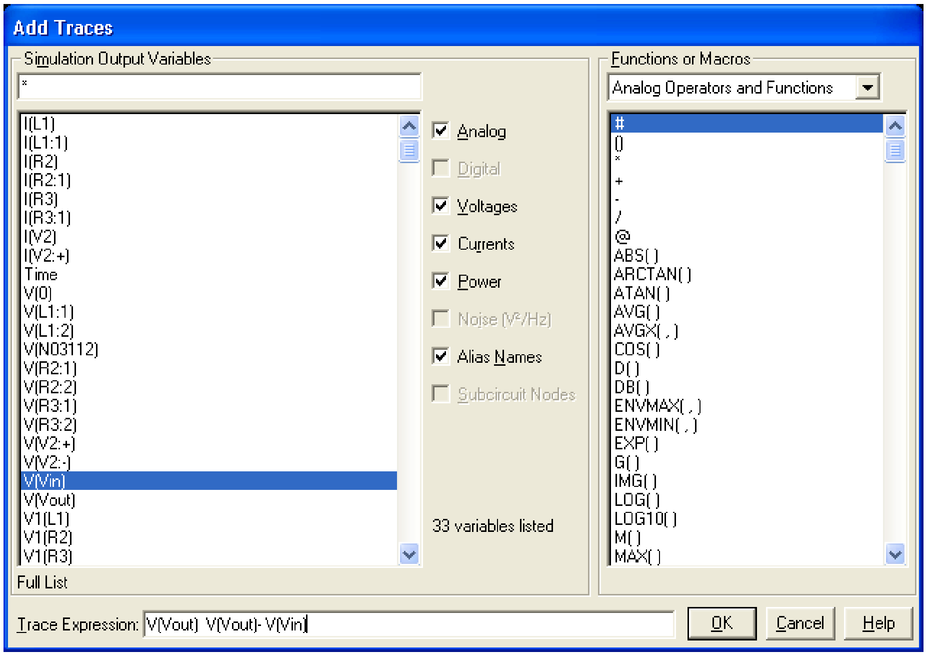
The simulation profile setup for an AC Transient analysis is shown below.



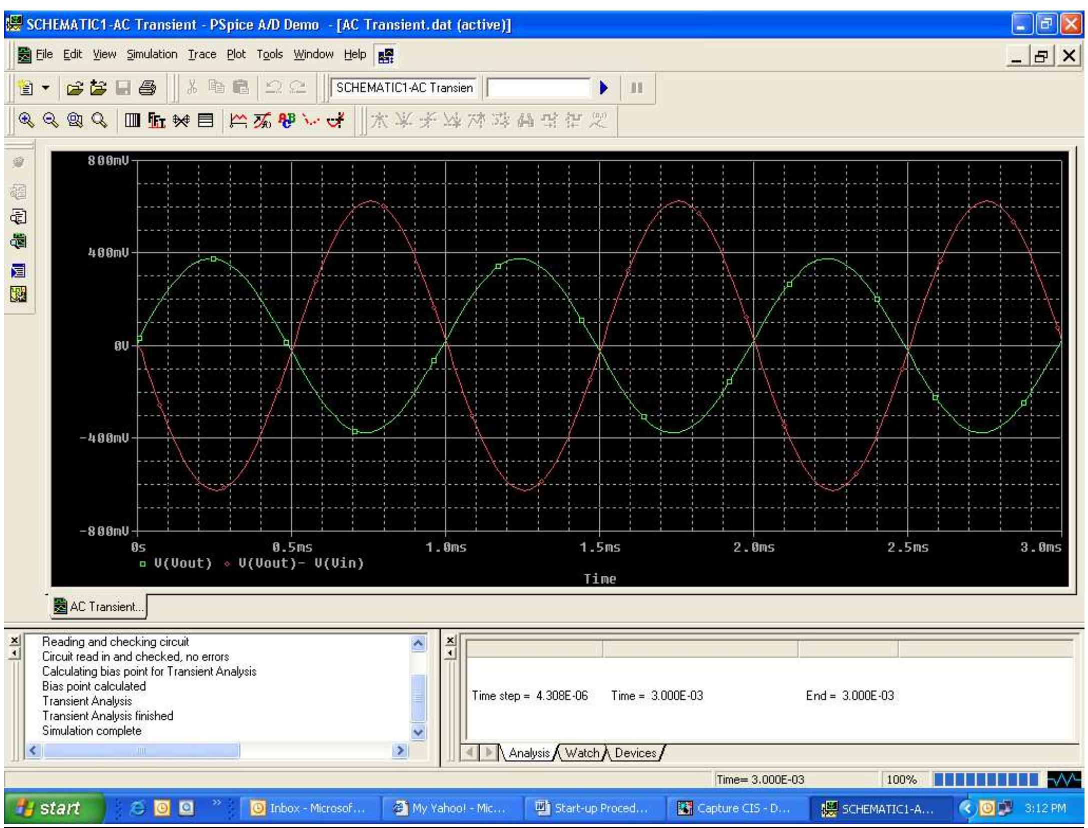

We will be using the General Settings option. Since the AC generator was set to a frequency of 1kHz, I have set the **Run to time** to 3msec to display 3 complete cycles. Also I have chosen to display approximately 100 points per cycle, therefore I have set the **Maximum step size** (under the **Transient options**) to 10usec.

Click OK to apply the values and close this window. Now run PSpice. When the circuit is finished simulating, an empty Probe window will appear. The X-AXIS will already be set up with a time scale set to 3msec in increments of 0.5msec. Left-click on the **Trace** menu or the shortcut on the toolbar menu. This brings up the **Add Traces** window. Highlight and left-click the outputs that you would like to have displayed. I have chosen V(Vout), the voltage across inductor L1 and its series resistance R3, and V(Vout) - V(Vin), the voltage drop across resistor R2, to add it to the **Trace Expression** line at the bottom of the screen.

Note that I have created the second expression by doing a subtraction operation on V(Vout) and V(Vin).



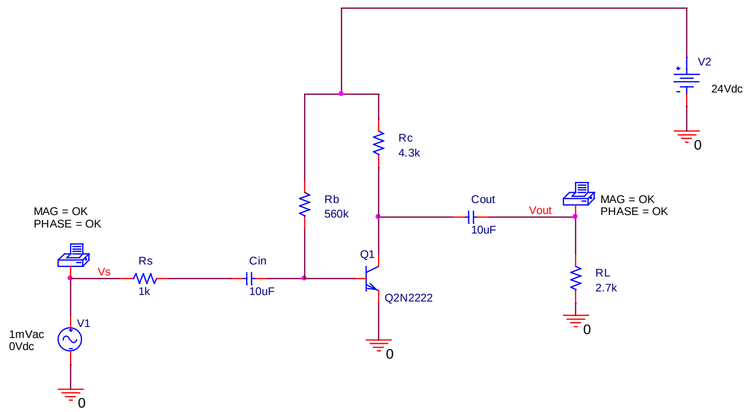

Left-clicking $\boxed{\text{OK}}$ brings up the plot of both voltage drops as a function of time..



## AC Sweep Analysis

In this mode you will be looking at the ac signals and using the results to determine the ac gain of the circuit.

You will use the circuit shown in the following schematic.




For the voltage source V1, use the **VAC** part. The **VSIN** part is used for the **Transient Analysis** simulations.

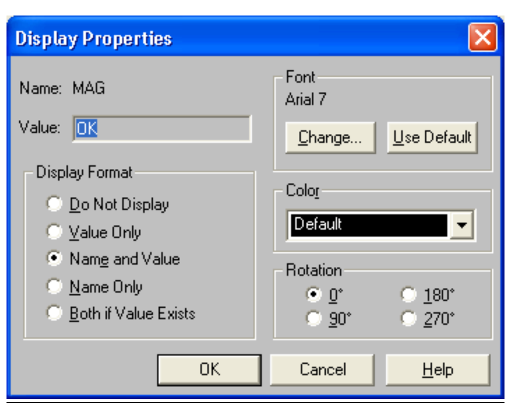
Note the use of a new symbol:  In the **Part** menu, this is listed as **VPRINT1 (**in the **SPECIAL** library). This symbol is used to tell PSpice that you wish to include some characteristics of this node in your output file.

Double-left clicking on the symbol will bring up the **Properties Editor** screen.

Under the column designated AC enter **OK**. Under the columns designated **MAG** and **PHASE,** enter **OK**. Continue entering OK under any other characteristic that you wish to examine and record. You don't have to enter OK as I have. Entering anything will activate the property.

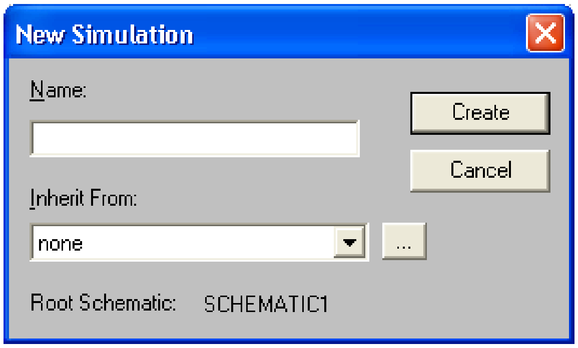
If you wish to display the name and/or the value of the characteristic on the schematic, as shown on the input node Vs, highlight the column and press the Display… button.

The following screen will appear:

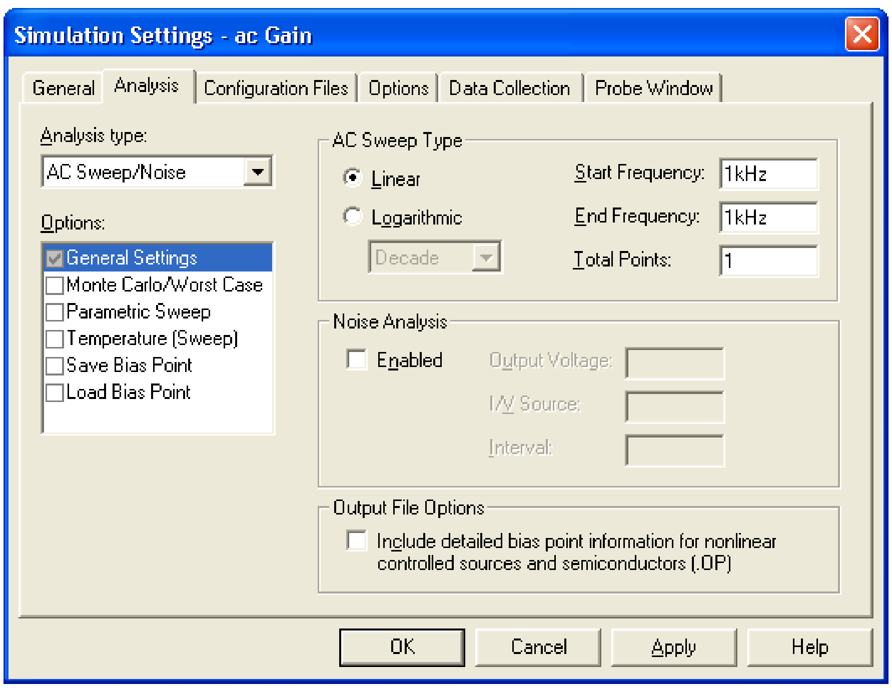


Here you can make your choices for displaying the symbol properties.

To start the simulation process, open the **PSpice** menu. The first choice available is **New Simulation Profile**. Left-click on it and the following window will appear.




Give the New Simulation a **Name**. For this example, use "*ac Gain*". Left-click Create and the next screen will appear




Choose the **AC Sweep/Noise** under the **Analysis type** option. Since in this example we will be looking at only one frequency, (i.e. 1kHz) set the **AC Sweep Type** as shown above. Under the **Probe Window** uncheck the **Display Probe Window** setting.

Run the simulation and open the output file (either using the shortcut icon on the left side or under **PSpice**, and **View Output File**). At the end of the output file you will find the **AC Analysis** results, a portion of which is shown below.

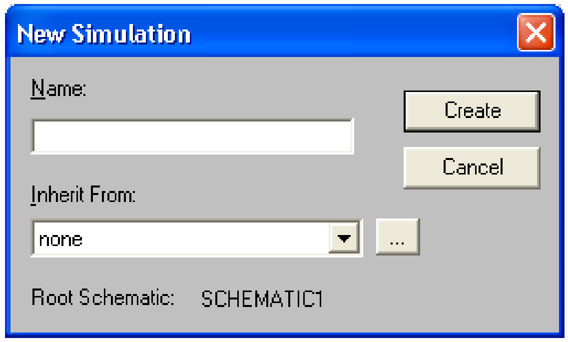
```
  ****     AC ANALYSIS                  TEMPERATURE =   27.000 DEG C

******************************************************************************

 FREQ        VM(VOUT)    VP(VOUT)

  1.000E+03   2.798E-03      -1.789E+02


**** 12/31/06 23:08:44 ******** PSpice Lite (July 2006) ******* ID# 10813 ****

 ** Profile: "SCHEMATIC1-ac Gain"  [ F:\Orcad Capture CIS, Release 15.7\ac example-pspicefiles\schematic1\ac gain.sim]


 ****     AC ANALYSIS                  TEMPERATURE =   27.000 DEG C


******************************************************************************

 FREQ      VM(VS)     VP(VS)


  1.000E+03   1.000E-03   0.000E+00
```

In this example, the frequency (FREQ), magnitude (VM) and phase (VP) are displayed for the output signal node (Vout) and the input signal node (Vs).

Recall that for your reports, you can modify, add to, or delete anything from the output file, just as you would with a text file (which it is). Hence we could calculate the ac gain and include it in the simulation report.

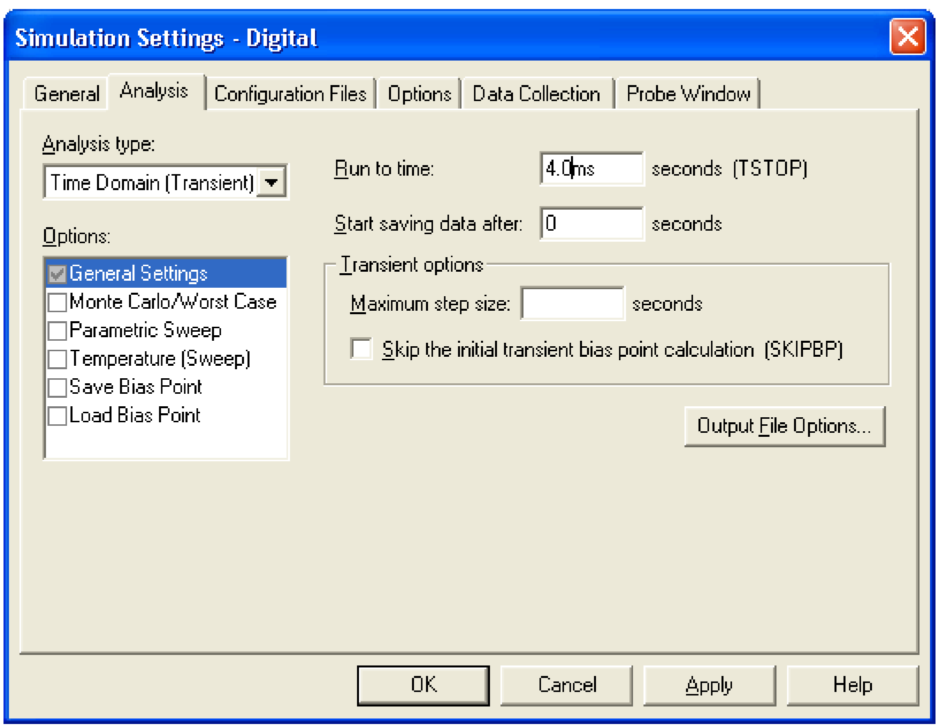
## **Digital Simulations**

After creating your new project, (as was described in page 1) we can now start the digital simulation process. Open the **PSpice** menu. The first choice available is **New Simulation Profile**. Left-click on it and the following window will appear.



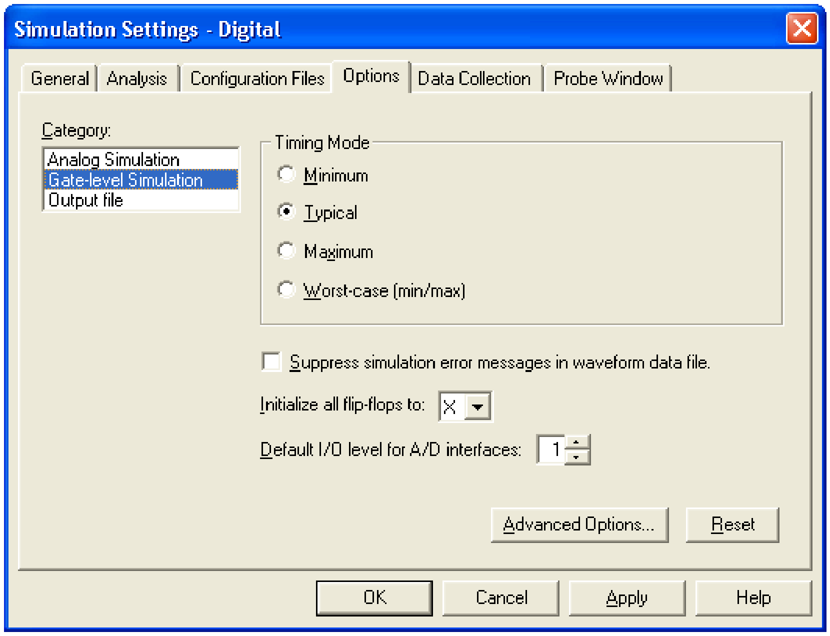

Give the New Simulation a **Name**. For now we can use *"Digital"*.

Left-click Create and the next screen will appear

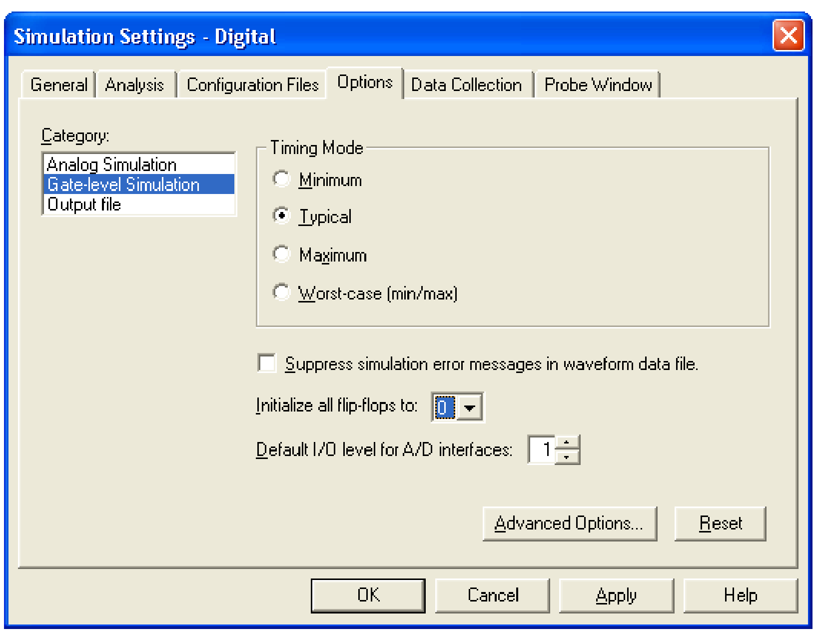



Notice that the Run to time: and the Start saving data after: times were set to 4.0 ms and 0 seconds respectfully. This is because we will run our simulation from t = 0 to t = 4.0 ms.

Next go to the Options tab and set the Category: to Gate-level Simulation.




When simulating digital circuits that contain flip-flops or their derivatives, or circuits that contain both analog components and digital flip-flops or their derivatives, you need to initialize the data storage devices. In OrCad Capture, you do this by accessing the Options tab under the Simulation Settings menu and then highlighting the Gate-level Simulation in the Category window. An example is shown below.




On the line labeled **Initialize all flip-flops to**: change the options from "X" (logic don't care) to either a "0" (logic Zero) or a "1" (logic ONE).

## Use of Digital Input Stimuli

OrCad Capture has 6 types of digital input stimuli:

Digital “HI”, Digital “LO”, Digital Clock, Voltage Pulse (an analog pulse generator), Digital Stimulus, and File Stimulus.

.

| | | | |
|---|---|---|---|
| Digital "HI"<br>$D_HI/SOURCE | HI | Digital "LO"<br>$D_LO/SOURCE | LO |
| Digital Clock<br>DigClock/SOURCE | OFFTIME = .5uS<br>ONTIME = .5uS<br>DELAY =<br>STARTVAL = 0<br>OPPVAL = 1<br>DSTM2<br>CLK | Voltage "Pulse"<br>VPULSE/SOURCE | V1 =<br>V2 =<br>TD =<br>TR =<br>TF =<br>PW =<br>PER =<br>V1 |
| Digital Stimulus<br>STIMX/SOURCE<br>X =1,4,8,16 | DSTM<br>S1 | File Stimulus<br>FileStimX/SOURCE<br>X = 1,2,4,8,16,32 | DSTM3<br>F1<br>FILENAME =<br>SIGNAME = |

**TABLE OF DIGITAL INPUT STIMULI**

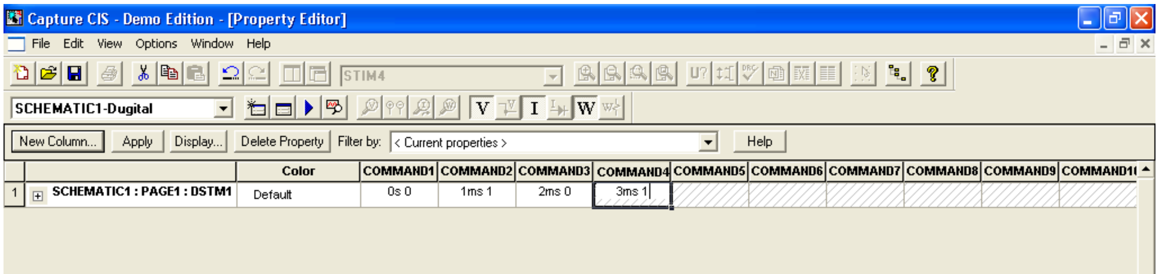
***Digital “HI”*** and ***Digital “LO”*** are constant digital “1” and digital “0” inputs and are found in the PWR icon on the right hand side toolbar.

The rest of the Digital Input Stimuli are found in the **Place** menu and the **Part** selection or in the Place part icon.

***Digital Clock*** **-** a digital “1” and digital “0” clock generator whose parameters can be input and changed from the logic diagram.

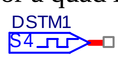
***Voltage “Pulse”*** **-** an analog “one time” voltage pulse generator that also has parameters that can be input and changed from the logic diagram.

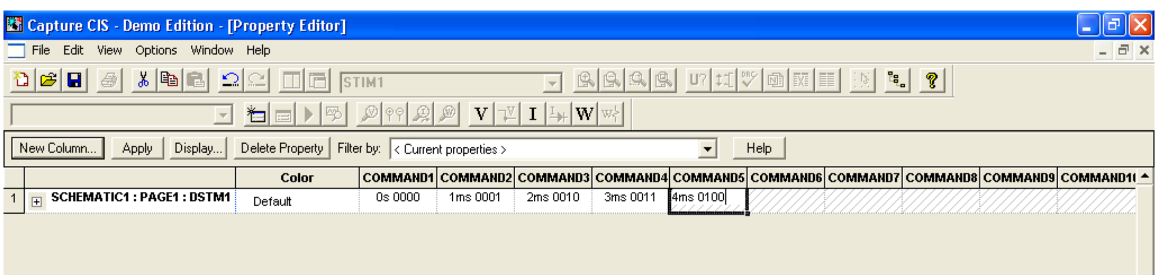
***Digital Stimulus*** **-** a digital input whose wave shape is specified in the Property Editor. Shown below is the Property Editor for a single line Digital Stimulus - STIM1/SOURCE.

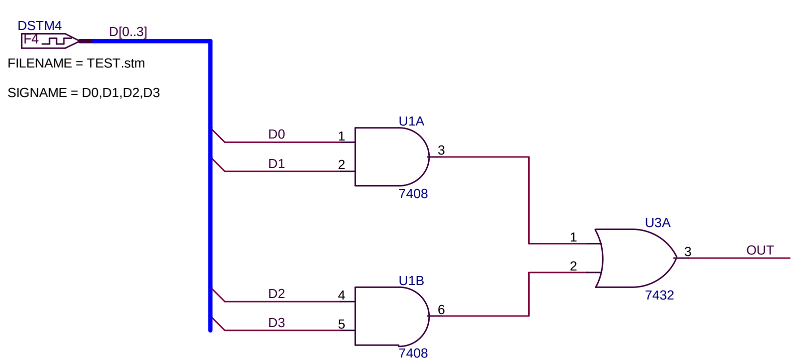

Note in the example above COMMAND 1 is 0s 0, COMMAND 2 is 1ms 1, COMMAND 3 is 2ms 0, and COMMAND 4 is 3ms 1. As you can guess this means that at 0s, the starting time, the digital input is logic "0", at 1ms the logic input is logic "1", at 2ms the logic input is logic "0", and at 4ms the logic input is logic "1". These times are in absolute values. They could also have been written as COMMAND 1 is 0s 0, COMMAND 2 is +1ms 1, COMMAND 3 is +1ms 0, and COMMAND 4 is +1ms 1 - relative time values.

For multiple inputs, a series of digital inputs can be specified using the STIM4, STIM8, and STIM16 inputs. Shown below is the Property Editor for a quad line Digital Stimulus - STIM4/SOURCE




***File Stimulus*** **-** a means of inputting large amounts of digital inputs via a text file. In the following example a four (4) input file stimulus is connected to two **AND** gates via a bus wire (to be described in the next section).



In this example, the digital signal transitions for signals D0, D1, D2, and D3 are specified in the file, named “TEST.stm” that was created by a text editor such as Notepad. **DO NOT** use a word processor.

This file must be stored in your project under the “Simulation Profile” folder that you have previously created. In the example shown on the previous pages, it would have been **Digital**.

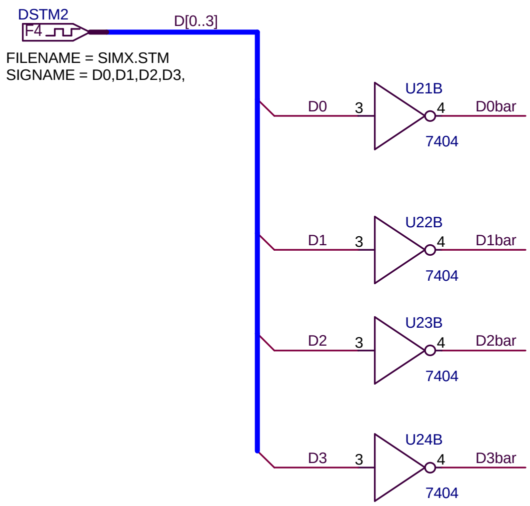
LISTING OF FILE “TEST.stm”

```
D3,D2,D1,D0          ; Header, listing the signal names (in order)
                     ; This header must be placed on the first line
0ms    0000          ; Comments are placed after the ;
1ms    0001
2ms    0010
3ms    0011
4ms    0100
5ms    0101
6ms    0110
7ms    0111
8ms    1000
9ms    1001
10ms   1010
11ms   1011
12ms   1100
13ms   1101
14ms   1110
15ms   1111
```

## Use of Bus Wires

A scalar wire, the one that we have been using all along, can carry only a single signal. A bus wire can carry multiple signals. This can cut down on the number of wires on your schematic, thereby making the schematic easier to read.

Looking at the example block diagram below




You can see that we have multiple signals coming from a single wire. This is the bus wire.

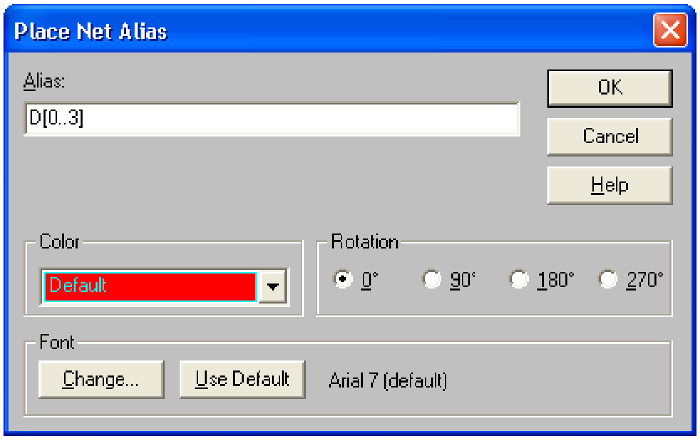
Along with that, I have used a multiple FileStim input source, in this case a FileStim4 source. The four means that it can inject 4 signals simultaneously. There are also 2 input, 8 input, 16 input, and 32 input versions in the OrCad Capture source library.

The FileStim text file is created and labeled on the "FILENAME = " line in the same way. But now, on the "SIGNAME = " line, multiple signal names can be entered. As shown in the example above, the letter designation can be anything but the suffix following it must be a number, and <u>IN SEQUENCE</u>. Typically it is the numerical sequence of signals available.

At the signal exit points, notice the use of "BUS ENTRY" lines. A bus entry is used to tie a signal to a bus. The advantage of using bus entries instead of wires is that two bus entries can be connected at the same point on a bus without connecting the signals. If two wires are run directly to a bus at the same location, the signals are connected.

The nodes at the gates are marked with the appropriate **Net Alias**. If you start with the lowest numbered node, for example D0, place it on the appropriate wire and then continue placing the rest of the nodes in sequence, the **Net Alias** name will increase automatically.

Now one more thing needs to be done. The bus wire must be labeled with the names of the signals. In this case, D0 to D3. Using the **Place** menu and the **Net Alias** selection, create the label of the bus wire as shown below.



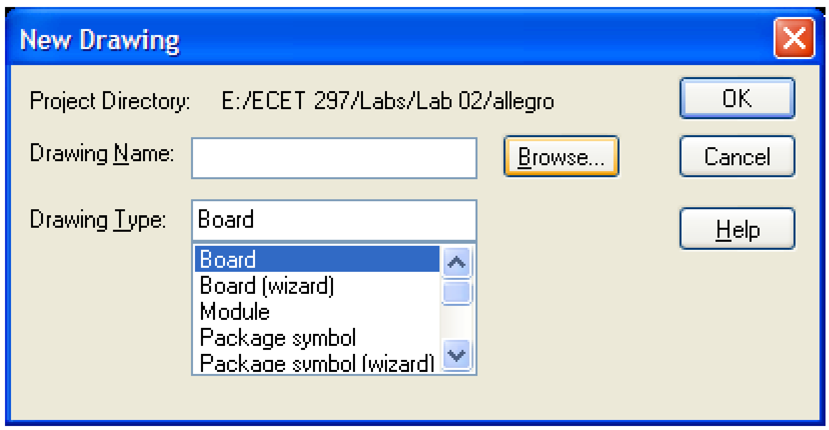

Place the **Net Alias** name on the bus wire and you're all set.

## 1: OrCad PCB Editor 15.7

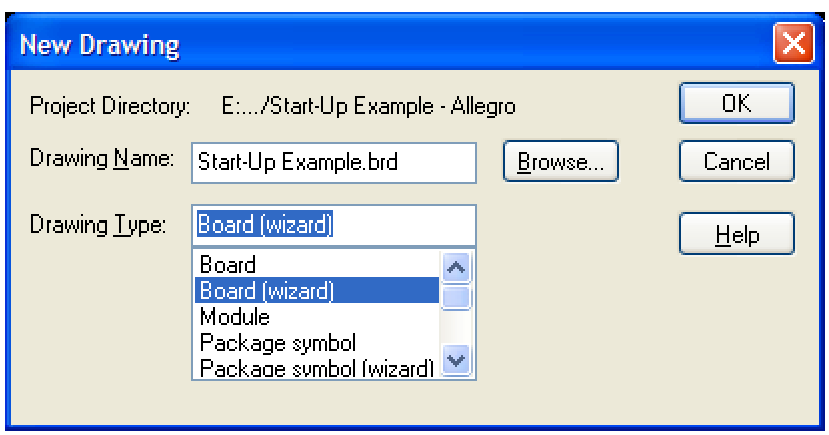
OrCad PCB Editor 15.7, formally called Layout, is based on the Cadence Allegro layout program. It is a powerful circuit board layout tool that has all the automated functions one needs to complete the layout of a multilayer printed circuit board.

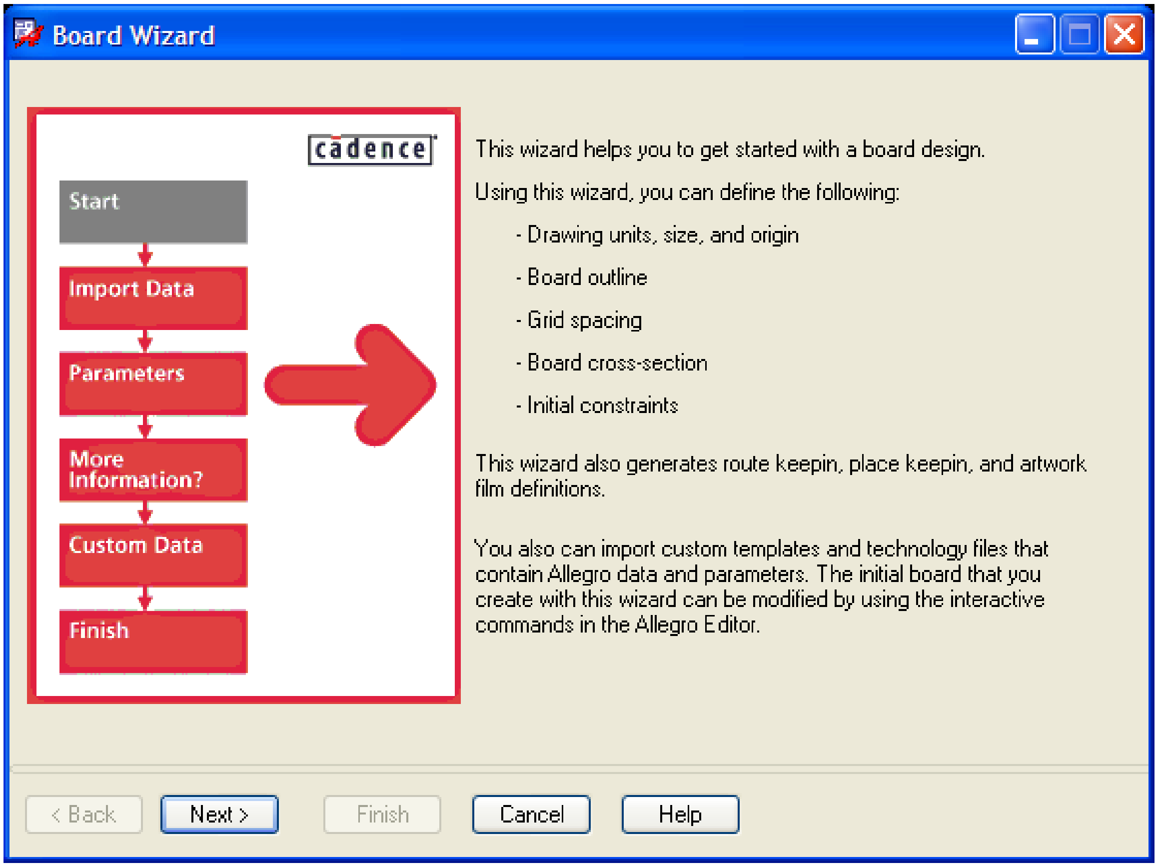
You start a new project (program) by going to the **File** menu in the upper left corner, then **New**.

The following screen will appear:




We will begin our session by opening a blank file. Let us call it Start-Up Example to be located in my Start-Up Example–Allegro directory and highlighting the Board (wizard).






Follow the six steps in the board wizard to complete the PC board initialization.

Now it is time to place the components on the board