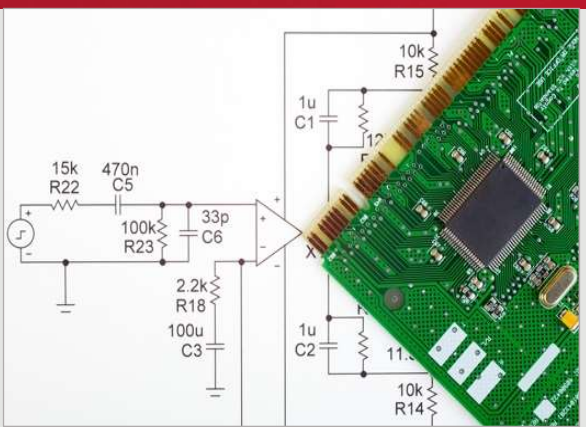
cadence

Mixed-Signal Simulation with PSpice for TI 2020®
Simulation Labs



This lab manual uses hands-on exercises for both new and moderately experienced users of Cadence® PSpice® to maximize the performance of their circuits.

Who will this benefit?

"Engineers and managers dealing with PCB designs, hardware, SPICE models, systems design, FPGA designs, RF circuits, analog/mixed or digital simulations, EMI /EMC analyses, or electronics in general."

Quickly Create Schematic for PSpice Simulation

Learn how to quickly design your circuit idea in the OrCAD workspace & get it ready for PSpice simulation to see how it will behave in real world

Import 3rd party models in your design

Take advantage of the ability to bring in any third-party vendor model from internet into PSpice

Learn Different Types of Analysis Techniques

Learn when to use and how to configure and run the following analyses:

- Transient, AC Sweep, DC Sweep Analysis
- Parametric & Performance Analysis
- Temperature
- Noise

Quickly Solving Convergence errors

Learn how to resolve the most common Spice circuit convergence errors with PSpice and use Auto Convergence to converge a simulation automatically

Software Product Version 17.4 - 2019

cadence

Table of Contents

Mixed-Signal Simulation with TI - PSpice® 2020

Starting with the Basics

Going a Little Advanced

Starting with the Basics

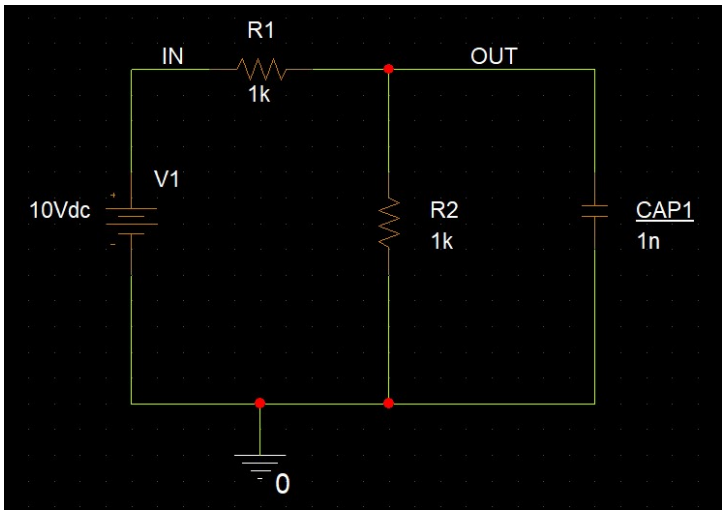
Module 1: Creating an OrCAD Capture Design for PSpice Simulation

Lab 1-1 Creating an RC Circuit

Objective: To enter and prepare a design for Cadence® PSpice® simulation

---

In this lab, you will create a new project and add libraries to it so you can place parts on a schematic. You will also draw wires to make connections and set up the source for the circuit and save your design. The RC circuit you will create in this lab is shown below,

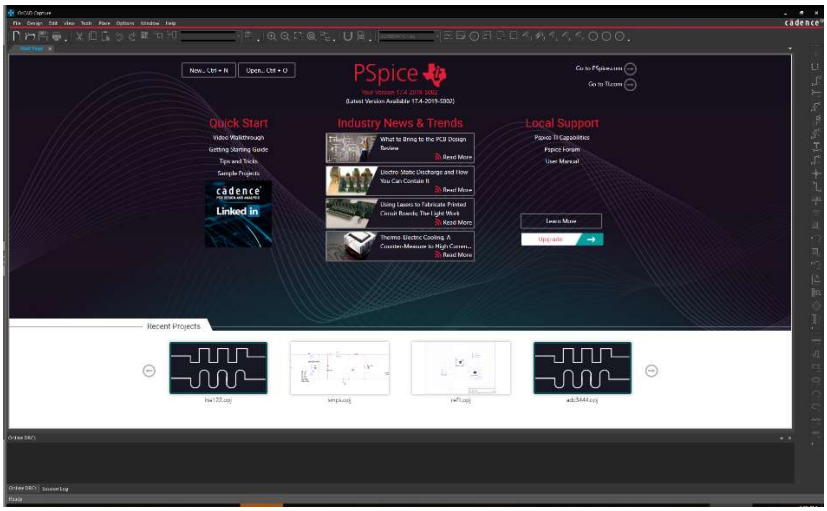


Creating a New PSpice A/D Project

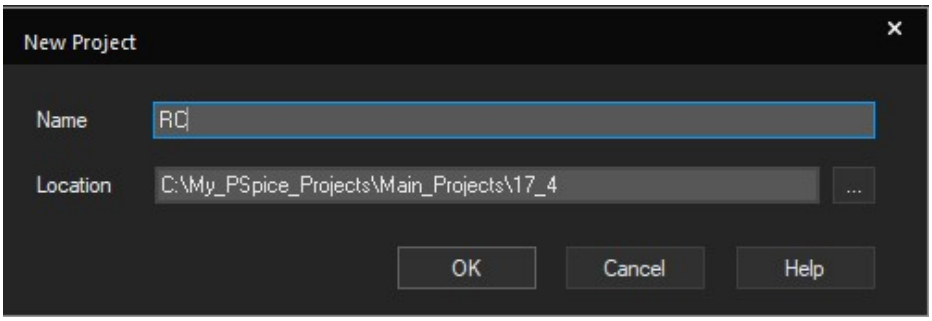
1. Choose **Start – Programs – Cadence PCB 17.4-2019 – Capture CIS 17.4**

   You can search "Capture" using the search button from task bar and open Capture CIS 17.4 quickly

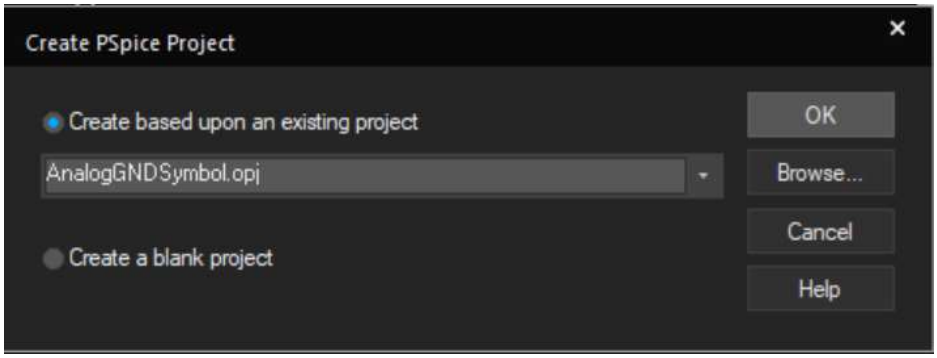
   This opens the OrCAD Capture Start page



2. Choose **File – New – Project** to launch the New Project window.

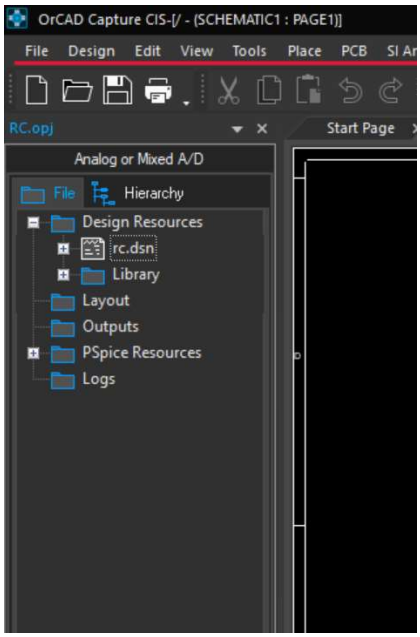


3. Specify project details as shown in the image and click OK.



4. Choose **empty_all_libs.opj** from the pull-down menu and click **OK**.

5. This option generates a new project, which has a single schematic folder, a single page, and is configured to four PSpice A/D libraries.

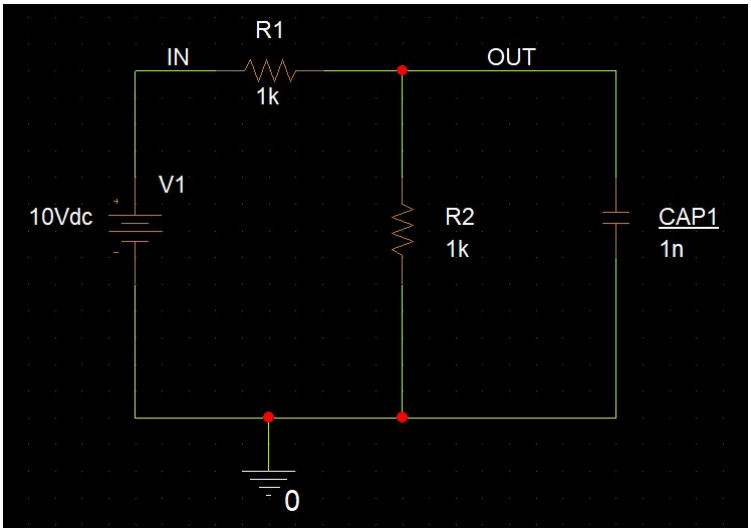


6. Double-click **RC.dsn** or click the plus sign (+) to the left of the design name to expand the display of design contents.

7. **Double**-click **Schematic1** or click the plus sign to the left the schematic name to display folder pages.

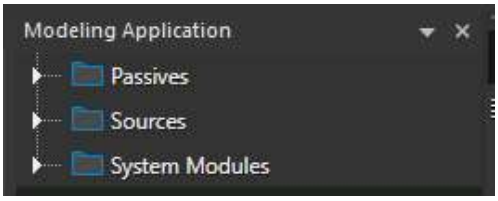
8. **Double**-click **Page1** to open the drawing window.

   To create the final RC circuit (shown below), complete the remaining steps in this lab

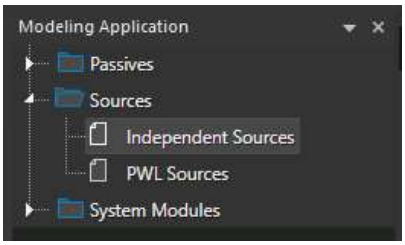


Placing the Voltage Source

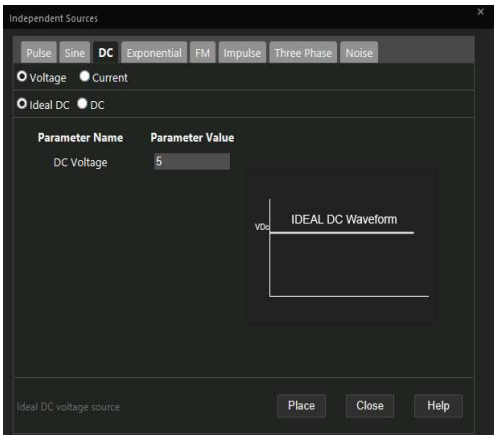
1. Go to **Place – PSpice Component – Modelling Application**. Modelling application tab opens on the right side as shown below,



2. Expand Sources & select Independent sources from the options,



3. A dialog box to select various Independent sources opens. From the option, select an ideal DC voltage source. Selections shown below,



4. Place the source on the schematic page. Press **Esc** to end the placement mode

Placing the Resistors

1. Go to **Place – PSpice Component**, select **Resistor option** and place it on the schematic

2. Press **R** to rotate the resistor image attached to the cursor

3. Use Ctrl+C - Ctrl+V to copy paste the resistor & place another one on the schematic.

4. Press **Esc** to end placement mode

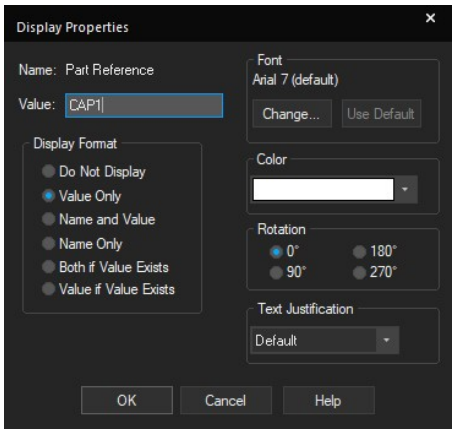
Placing the Capacitor

1. Go to **Place – PSpice Component** again, select **Capacitor option** and place it on the schematic

2. Press **R** three times to rotate the capacitor image attached to the cursor.

3. ***Assigning Reference Designator***

   Double - click on name of the capacitor (C1) and rename it to CAP1. Click OK

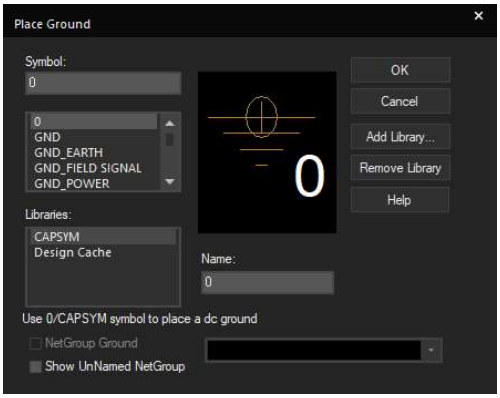


4. Press **Esc** to end the placement mode.

Placing the Analog Ground Symbol

1. Choose **Place – Ground,** a dialog box as shown below comes up.
   Select **CAPSYM** from libraries and select symbol **0** from the list above

   Or click the place ground icon ( ) from the draw graphical toolbar



2. Click **OK** to attach the symbol to the cursor and return to the schematic page.

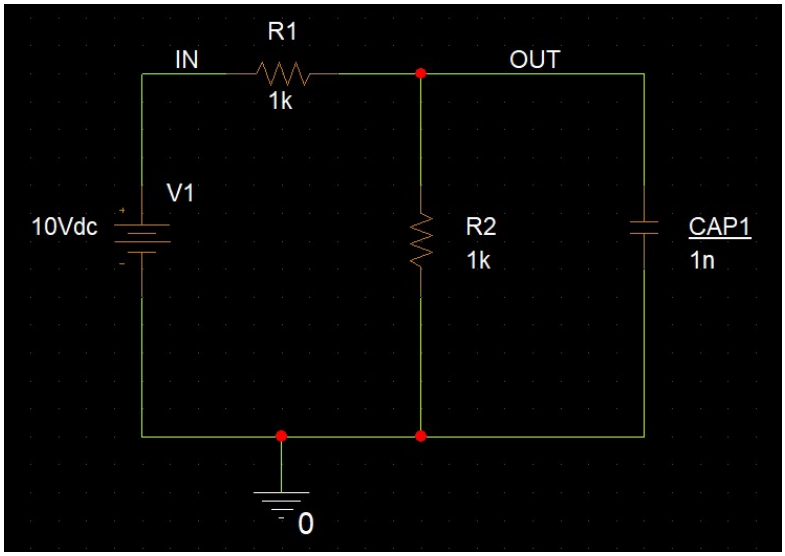
3. Click to place the ground symbol.

4. Press **Esc** to end placement mode

Connecting the Components

1. Go to **Place – Wire** *or* press **W** *or* click the **Place** Wire icon ( )

2. Wire together the components as shown in the following figure.



If you make a mistake while drawing wires or placing components, select the object and press **Delete** to remove it. You can also click the object, hold down the left mouse button, and drag it to the correct location. To stretch wires in Capture, select the segment, and then click and drag the end of the segment.

Labeling Nets

1. Choose **Place – Net Alias** or press **N** or click the icon

2. Specify **IN** in the **Alias** name field.

3. Click **OK**.

4. Place the alias on the net between V1 and R1.

5. Repeat steps 1–4 to label the node between R1 and CAP1 as **OUT**.

6. Press **Esc** to end the placement mode.

Assigning a DC Value of 10V to the Voltage Source

1. From the schematic page, **double**-click the 0Vdc text shown next to the source.
2. Change the DC value to **10V**.
3. Click **OK**.

[Tip: Similarly, you can change values of any other components of your circuit by simply double clicking on the value of that component]

Saving the File

1. Go to **File -Save** or click the save document icon () from the file toolbar

Lab Summary

In this lab, you performed the following tasks:

- Created a new project
- Added libraries
- Placed parts
- Drew wires

Module 2: DC Bias Point Analysis

Lab 2-1 Simulating a DC Bias Point

Objective: To determine the DC Bias values for a circuit.

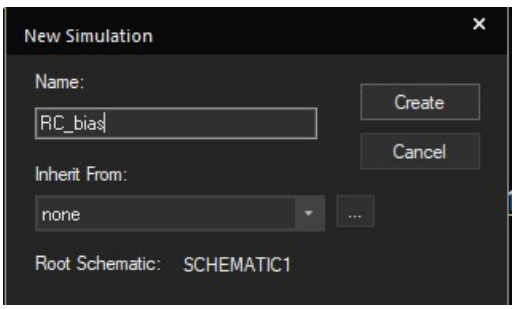
[***Concept Refresh***: DC Operating Point Analysis calculates the behavior of a circuit when a DC voltage or current is applied to it. The result of this analysis is generally referred as the bias point or quiescent point, Q-point. In most cases, the results of the DC Operating Point Analysis are intermediate values for further analysis]

Creating a Bias Point Simulation Profile

1. Use the same design from Lab 1-1. RC.opj

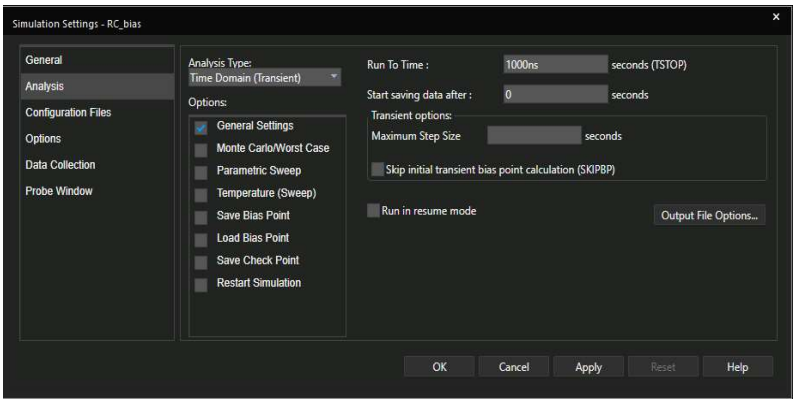
2. Choose **PSpice – New Simulation Profile** or click the New Simulation Profile icon.


3. Specify **RC_bias** as the name of the simulation profile.

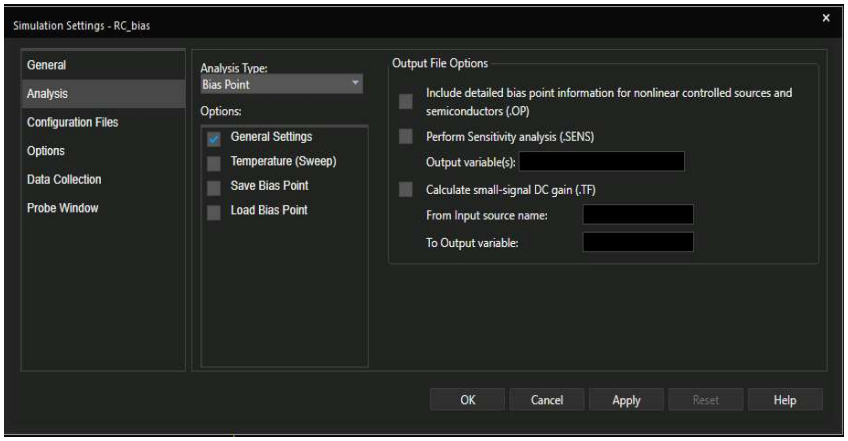


4. Click the **Create** button.

   The Simulation Settings dialog box opens.



5. Choose the Bias Point option from the **Analysis Type drop-down list**



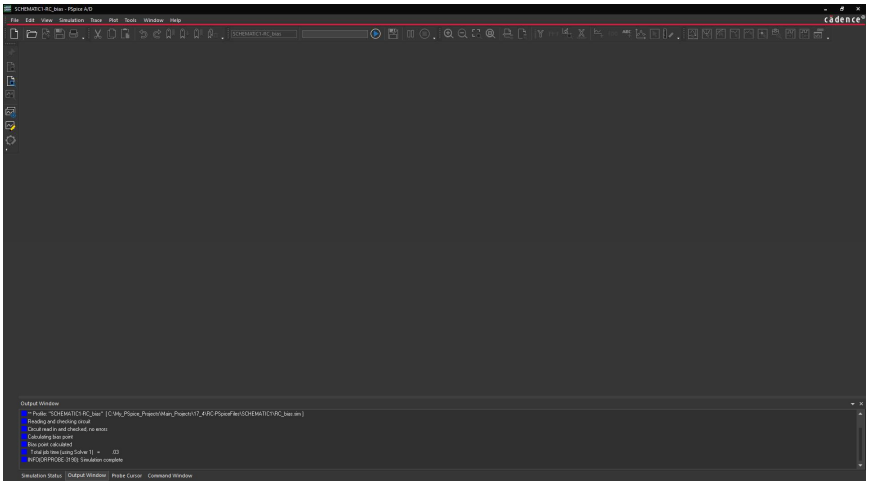
6. Click **OK**.

   The Simulation Setting dialog box closes, and your display returns to your schematic page

Running the Simulation

1. Choose **PSpice – Run**, *or* click the **Run** icon ( )

   The netlist creation process runs in the background. It automatically checks for Design Rule Check (DRC) errors in the background. After successful netlist creation without any DRC errors, the Probe window opens as shown in the following figure,



PSpice A/D simulator does *not* generate Probe data for bias analysis, so this Probe window is empty.

You can view bias point simulation results from either the output file generated during simulation or from the bias display available from the schematic page.

Viewing Results from the Output File

1. From the schematic page, choose **PSpice – View Output File** to examine simulation results in the output file or from the Probe window, choose **View – Output**

```
15: ** Creating circuit file "RC_bias.cir"
16: ** WARNING: THIS AUTOMATICALLY GENERATED FILE MAY BE OVERWRITTEN BY SUBSEQUENT SIMULATIONS
17:
18: *Libraries:
19: * Profile Libraries :
20: * Local Libraries :
21: * From [PSPICE NETLIST] section of C:\Users\ronaks\AppData\Roaming\SPB_Data\cdssetup\OrCAD_PSpice\17.4.0\PSpice.ini file:
22: .lib "nom.lib"
23:
24: *Analysis directives:
25: .OPTIONS ADVCONV
26: .PROBE64 V(alias(*)) I(alias(*)) W(alias(*)) D(alias(*)) NOISE(alias(*))
27: .INC "..\SCHEMATIC1.net"
28:
29:
30:
31: **** INCLUDING SCHEMATIC1.net ****
32: * source RC
33: R_R1          IN OUT   1k TC=0,0
34: R_R2          0 OUT   1k TC=0,0
35: C_CAP1          0 OUT   1n  TC=0,0
36: V_V1          IN 0 10Vdc
37:
38: **** RESUMING RC_bias.cir ****
39: .END
40:
41: **** 10/24/19 15:03:51 ******* PSpice 17.4.0 (Nov 2018) ******* ID# 0 ********
42:
43:  ** Profile: "SCHEMATIC1-RC_bias"  [ C:\My_PSpice_Projects\Main_Projects\17_4\RC-PSpiceFiles\SCHEMATIC1\RC_bias.sim ]
44:
45:
46:  ****     SMALL SIGNAL BIAS SOLUTION       TEMPERATURE =   27.000 DEG C
47:
48:
49: ********************************************************************************
50:
51:
52:
53:  NODE   VOLTAGE     NODE   VOLTAGE     NODE   VOLTAGE     NODE   VOLTAGE
54:
55:
56: (   IN)   10.0000  (  OUT)    5.0000
57:
```

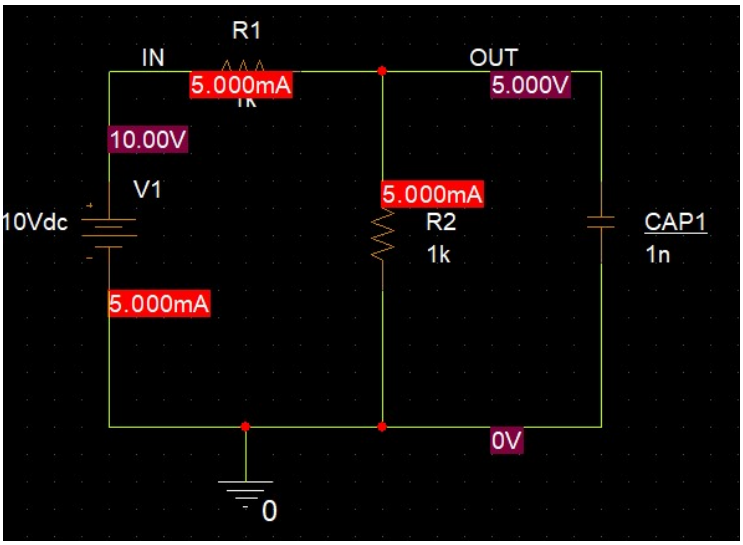
2. Close the output file.

Examining DC Bias Results from the Schematic

1. Choose **PSpice – Bias Points – Enable** to toggle on the voltage and current bias displays if they are not already displayed.

2. Toggle on and off the bias display toolbar icons to alter and control the view of results output.

Select any of the nets in the schematic, and then click the **Toggle Voltages** icon (). This enables you to toggle the voltage display of the circuit



[When you are designing an analog circuit, you first check BIAS conditions; DC Analysis. This tells you what happens if you simply turned the circuit on]

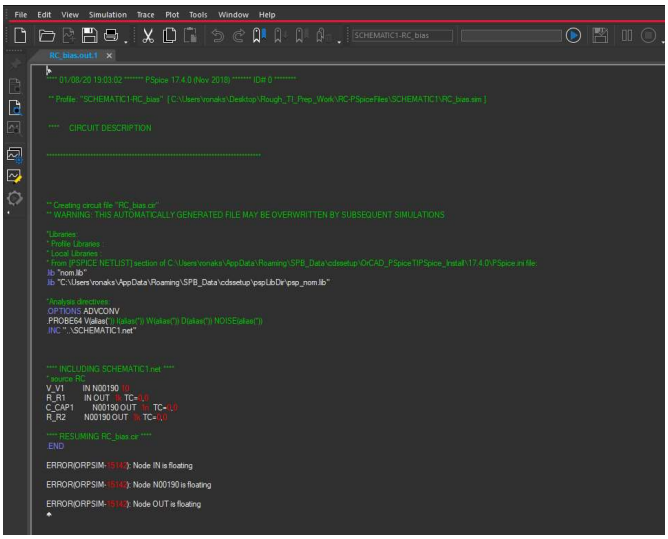
Additional Lab Exercise

1. Edit the schematic and remove the ground symbol.

2. Simulate the circuit again.

   What errors do you get and why?

   Answer: The error generated is probably the most common error that PSpice A/D users encounter. All SPICE-based simulators require that every analog node have a DC path to ground. This error is generally caused by either a missing ground symbol or an isolated node between two capacitors.



3. Replace the ground symbol after completing this exercise.

Lab Summary

In this lab, you performed the following tasks:

- Configured a DC bias point analysis
- Ran a DC Bias Point simulation
- Examined the results in the output file; used bias displays

Module 3: DC Sweep Analysis

Lab 3-1 Running a DC Sweep

Objective: To configure and run a DC sweep simulation and examine the results in the PSpice Probe window.

[***Concept Refresh***: DC Sweep Analysis is used to calculate a circuits' bias point over a range of values. This procedure allows you to simulate a circuit many times, sweeping the DC values within a predetermined range. You can control the source values by choosing the start and stop values and the increment for the DC range]

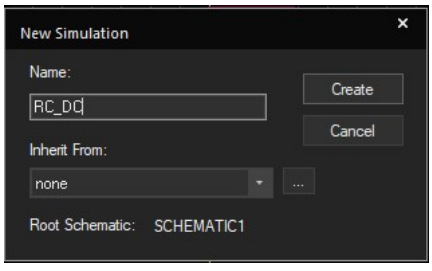
In this lab, you create a DC sweep simulation profile and then configure your DC source to sweep its value. You then run the sweep simulation and view the results in the Probe window. In the Probe window, you use markers and configure multiple axes.

Setting Up the Simulation

To complete the steps in this lab, continue using the RC project you created in Modules 1 and 2.

1. Choose **PSpice – New Simulation Profile** or click the **New Simulation Profile** icon ( )

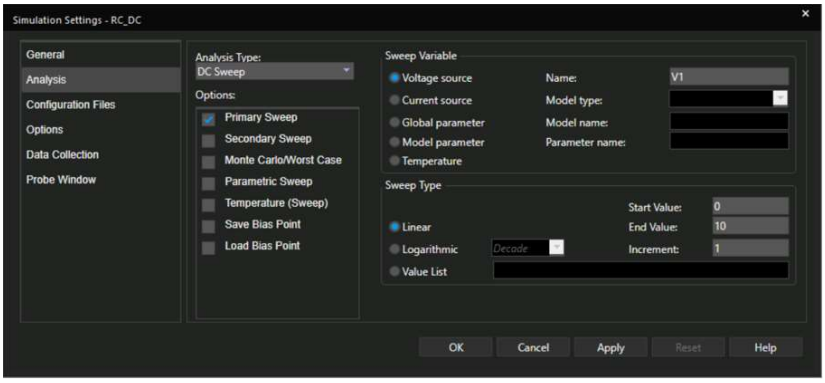
2. Name the new simulation **RC_DC**.



3. Click **Create.** The simulation settings dialog box opens.

4. Click the **Analysis** tab.

5. From the **Analysis Type** drop-down list, select **DC Sweep**.



6. In the **Options** list, make sure that only **Primary Sweep** is selected.

7. In the **Sweep Variable** section, select **Voltage source**.

8. Enter **V1** in the Name field.

9. In the **Sweep Type** section, select **Linear**.

10. Enter **0** in the Start Value field.

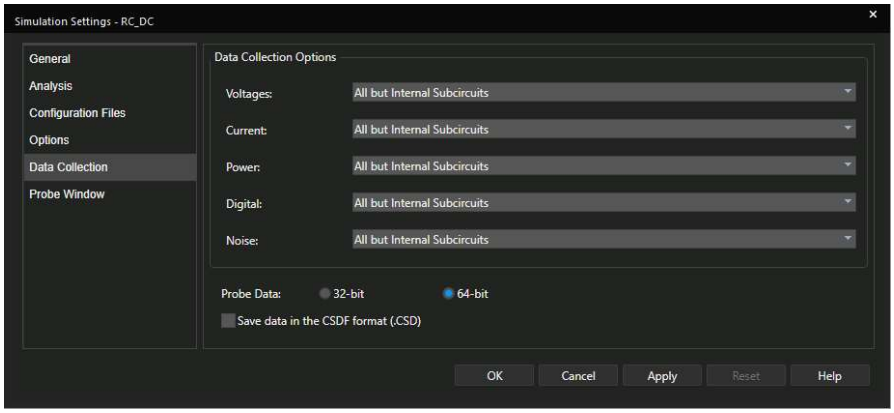
11. Enter **10** in the End Value field.

12. Enter **1** in the Increment field. Do not close the simulation profile tab

Setting Up Data Collection and Probe Windows

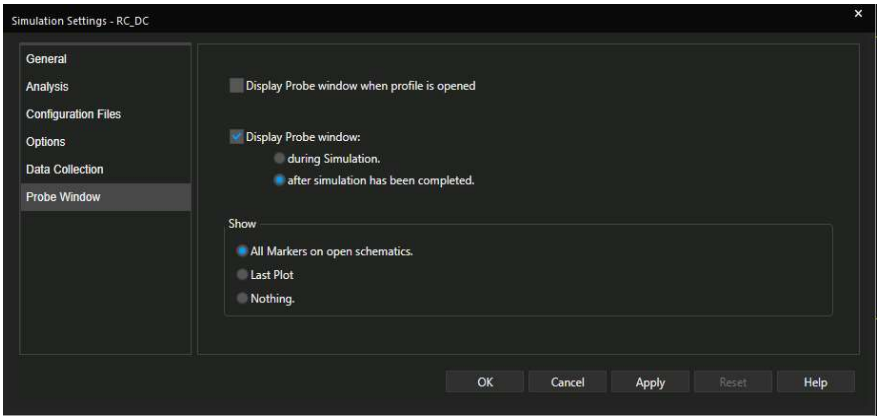
1. Click the **Data Collection** tab.

   The data collection settings for voltages, currents, power, digital, and noise display are displayed.



2. From the drop-down menus of each option, select **All.**

3. Click the **Probe Window** tab.



4. Select the **Display the Probe Window - after simulation has completed** option (shown above)

5. From the **Show section, select All Markers on open schematic.**

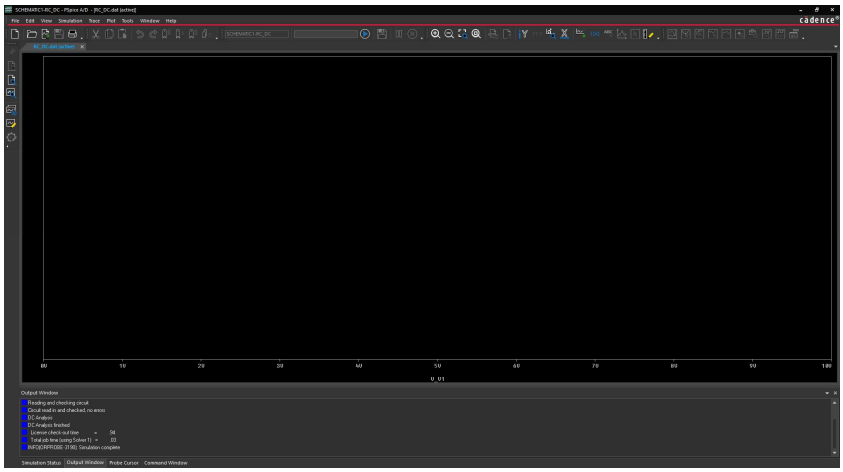
6. Click **OK** to close the Simulation Settings dialog box.

7. Choose **PSpice – Run** *or* click the **Run** icon ( ) to run the simulation.

   Capture automatically generates the netlist. When the simulation run is complete, the Probe window opens.

Analyzing the Results in the Probe Window

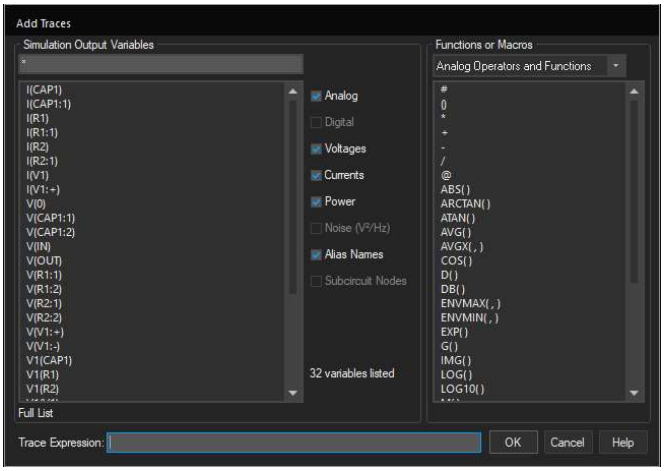
When the Probe window opens, you see a blank plot screen. If you have used markers in the design, the traces for those markers are displayed.



Plotting Traces from the Menu

1. From the Probe window, choose **Trace – Add Trace** or click the **Add Trace** icon ().
   The **Add Traces** dialog box opens.

2. Select **V(IN)** and **V(OUT)**, which are listed in the **Simulation Output Variables** list.
   This adds the selected traces to the Trace Expression field. All traces listed in the Trace Expression field are displayed at the same time.



[The DC Sweep analysis generates output like that of a curve tracer. It performs a series of Operating Point analyses, modifying the voltage of a selected source in pre-defined steps, to give a DC transfer curve.]

Plotting Traces with Markers – Quick & Easy

To see how you can display waveforms in the Probe window by placing markers, we will first delete the two displayed traces on Probe window and add them again with markers.

1. Click the trace name, **V(IN)** located below the X-axis.

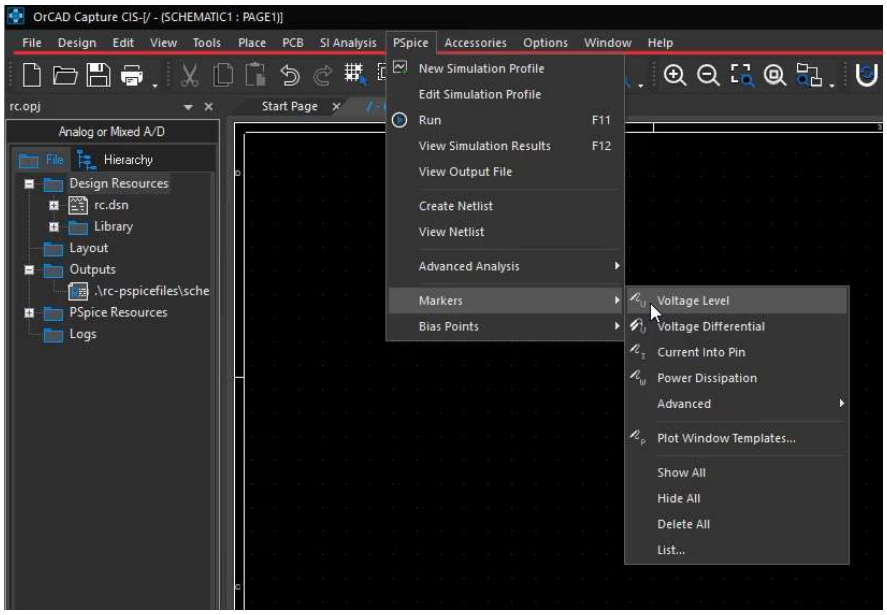
   The color of the trace name changes, indicating that it is selected now.

2. Press **Shift** and select **V(OUT)**.

3. Press **Delete** to remove both traces from the display.

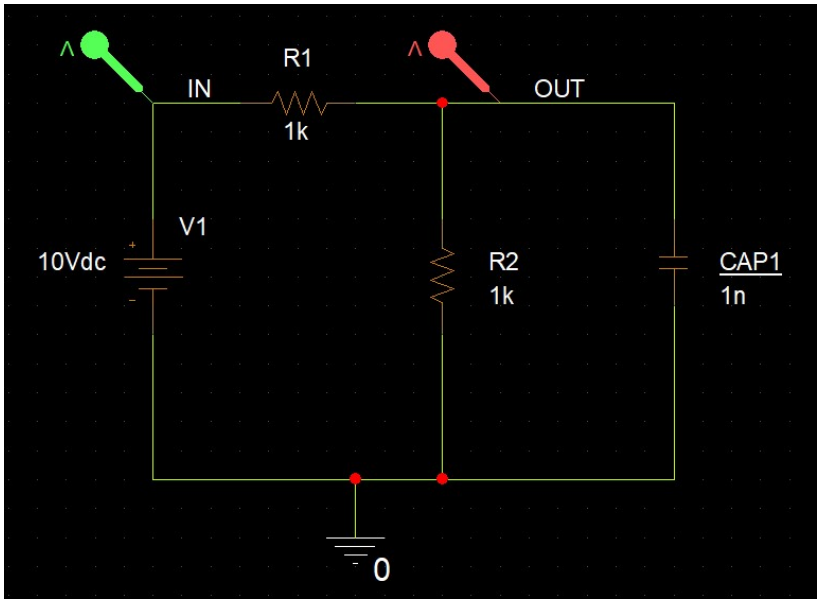
4. Press **Alt+Tab** or click the OrCAD® Capture icon ( ) from the taskbar to switch back to the schematic.

5. Choose the **PSpice – Markers – Voltage Level** or select the voltage marker icon ( )



6. Position your cursor on the node, **V(IN)** and click the left mouse button to place the marker.

7. Click the **V(OUT)** node to place a second marker.



8. Press **Esc** to exit the placement mode.

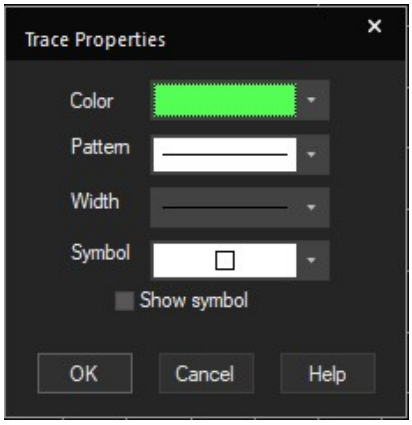
9. Return to the Probe window to view the graphic display of the traces you just added.

Customizing the Probe Display

You can customize the appearance of the Probe window. You can change the trace colors, modify the grid spacing, and change or display the trace symbols.

1. From the Probe window, **right**-click a trace and select **Trace Property**.

   The Trace Properties dialog box opens.

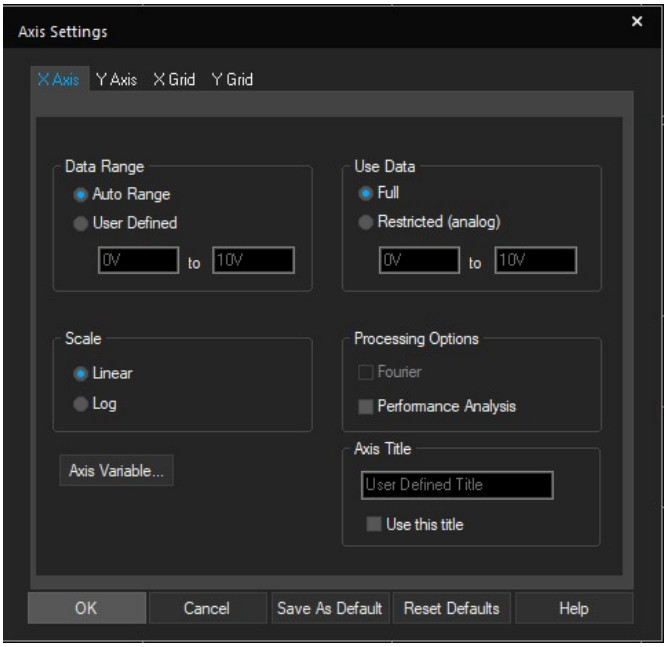


2. Change the color setting of the trace, the pattern of the line, the thickness of the line, the symbol denoting the trace, and whether to display the symbol on the trace itself.

3. Click **OK** to save your changes and close the Trace Properties dialog box.

Changing Axis and Grid Settings

1. To edit axis and grid settings, **right**-click any of the grid lines and select **Settings o***r* choose **Plot – Axis Settings**. The Axis Settings dialog box opens.



2. Click the **X Axis** tab to do the following:
   - Set the displayed data range.
   - Set the range of data that the Probe functions consider.
   - Specify whether the scale should be linear or logarithmic.
   - Change the X-axis variable.

3. Click the **Y Axis** tab to do the following:
   - Set the data range that will be displayed.
   - Specify if the scale should be linear or logarithmic.
   - Change the axis title.
   - Change the axis number.

4. Click the **X Grid** or **Y Grid** tab to do the following:
   - Set the major grid spacing.
   - Choose the way in which the grids are drawn (lines verses dots or plus sign (+) symbols).
   - Choose to display numbers below the axis.
   - Choose to display tick marks inside the plot edge.

- Chose the minor grid spacing.
- Chose the way the grid is drawn.
- Choose to display tick marks inside the plot edge.

5. Click **Cancel** in the Axis Settings dialog box.

6. Choose **Plot – Add Y Axis**.

   A second Y axis is added to the Probe output.

7. Choose **Plot** – **Axis Settings**.

8. Click the **Y Axis** tab.

9. In the **Y Axis** drop-down list, select **axis 2**, and then click the **Right** radio button to move axis 2 to the right edge of the plot.

10. Click **OK**

Lab Summary

In this lab, you performed the following tasks:

- Set up a DC sweep simulation
- Set up data collection and Probe windows
- Analyzed the results in the Probe window
- Plotted traces from the menu
- Plotted traces using markers
- Customized the Probe display
- Changed axis and grid settings

Module 4: AC Sweep Analysis

Lab 4-1 Running an AC Sweep Analysis

Objective: **To configure an AC sweep profile, AC source, run the analysis, describe the results in Probe, and use complex trace variables.**

[*Concept Refresh*: *Understanding purpose of AC sweep analysis for a circuit*

When you are designing an analog circuit, you first check Bias conditions, using DC Analysis. This tells you what would happen if you simply turned the circuit on and applied no signal to it. Then, you perform an AC analysis to figure out frequency response of the circuit.

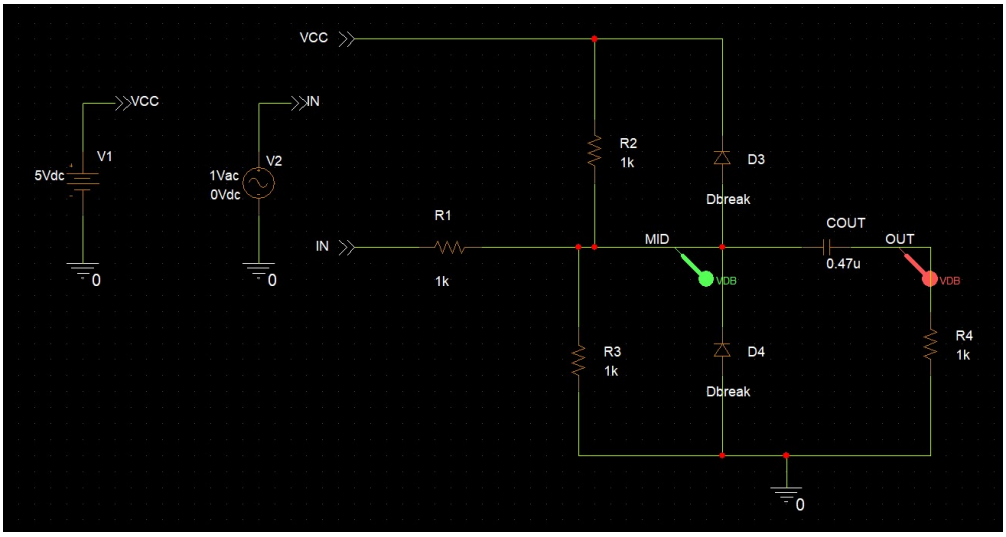
Let's take an example of a microphone amplifier: After figuring out the bias conditions, you know the voltage of each Node in the circuit. But, then what? You will apply a signal to it. The node voltages will vary around the biasing, that is DC points. So, the small signal response of the circuit will be highly dependent on the DC bias points. Around those bias points, responses could be simplified as linear, because, the changes are so small, thereby allowing you to analyze AC response in a simple way.

The AC analysis allows you to figure out what happens to your circuit when you apply well-behaving AC signals into its input (e.g., a 10mVpp sine wave without noise] Source

Creating a New PSpice A/D Project

1. Create a new PSpice® A/D project called **Clipper**.

2. Draw the circuit as shown in the below figure. Use the following parts and symbols: **R, C, VDC, VAC, Dbreak, 0**, and **Off-page connector**. The icon for an off-page connecter looks like ()



3. Change the reference designators as shown for **COUT**. The other reference designators do not need to match those in the figure.

4. Edit the values of the sources.

5. Change the values of the capacitors and resistors per the figure.

6. Label the **off-page connectors** as shown.

7. Add an alias to the net for **MID** & **OUT**

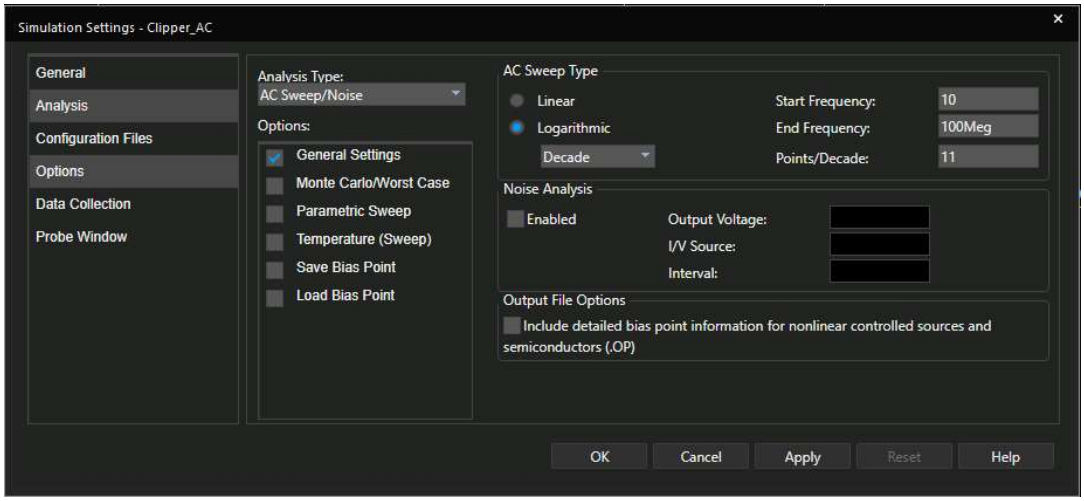
Setting Up and Starting an AC Sweep Analysis

1. Create a new simulation profile called **Clipper_AC**.

2. Click the **Analysis** tab.

3. Select **AC Sweep/Noise** as the simulation type.

4. Set up the simulation parameters as shown below.



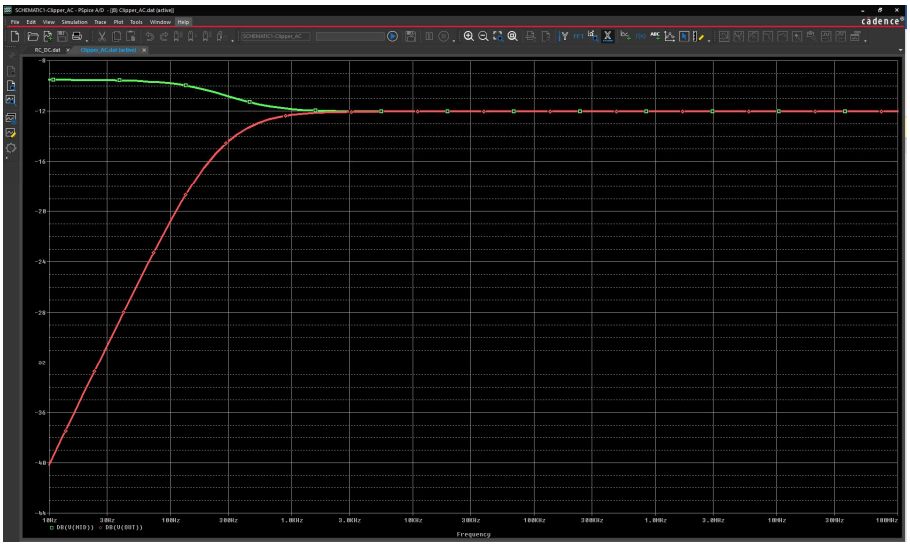
5. Set the **Data Collection** and **Probe Window** parameters as desired.

6. Click **OK** to save your simulation settings, close the Simulation Settings dialog box, and return to the schematic view.

7. Go to **PSpice – Markers – Advanced** – **dB Magnitude of Voltage** to place markers on the **MID** and **OUT** net of the schematic**.**

Running the Simulation and Viewing Results

1. Choose **PSpice – Run** *or* click the **Run PSpice** icon ( )
2. After the simulation has successfully completed, results are displayed in the Probe window as shown below.



Probe displays the dB magnitude of the voltage at the marked nets, *Out* and *Mid,* as shown.

Because the AC sweep is a linear analysis and the input voltage is set to 1V, the output

voltage equals the gain (or the attenuation) of the circuit.

Displaying a Bode Plot in PSpice

[***Concept Refresh:*** Bode plots are a very useful way to represent the gain and phase of a system as a function of frequency. This is referred to as the frequency domain behavior of a system

It is usually a combination of a Bode magnitude plot, expressing the magnitude (usually in decibels) of the frequency response, and a Bode phase plot, expressing the phase shift]

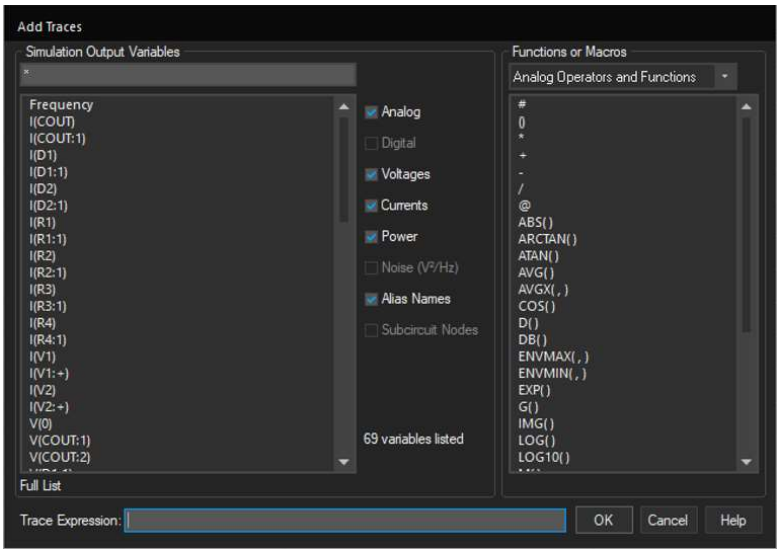
1. Click the symbol for the trace **Vdb(Mid)** located below the x axis of the Probe window.

2. Press **Delete** to remove the Vdb(Mid) trace.

3. Go to **Plot – Add Y Axis** menu.

4. Choose the **Trace – Add Trace** menu, *or* click the **Add Trace** icon ( )

   The **Add Traces** dialog box opens as shown below,



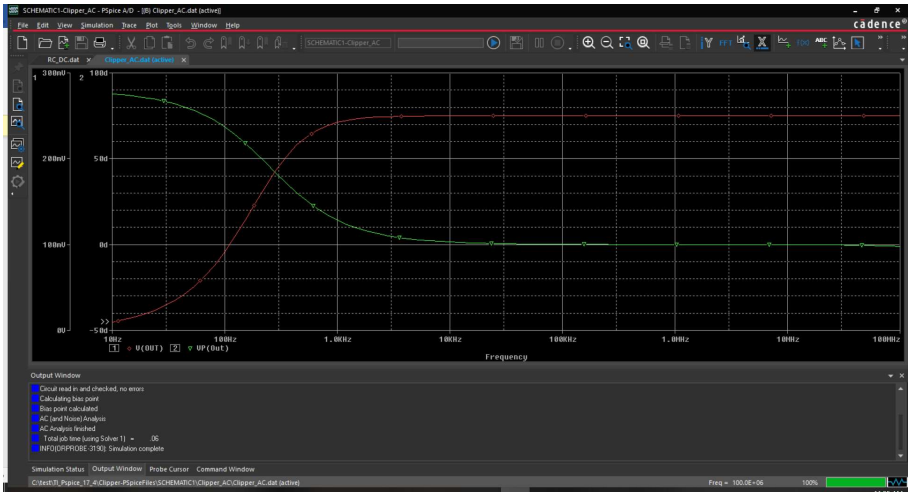
5. Select **V(OUT)** from the list of simulation output variables.

6. Click the **Trace Expression** field to place the cursor in it.

7. Edit the trace name to be **VP(Out)**. Press **Enter**.

8. Click on the trace **DB(V(OUT)).** Click the **Trace Expression** field to place the cursor in it.

9. Edit the trace name to be **V(OUT)**. Press **Enter**.



10. Close the results window and save the Clipper design.

Lab Summary

In this lab, you performed the following tasks:

- Configured an AC Sweep profile
- Ran the analysis
- Configured an AC source
- Examined the results in Probe
- Used complex trace variables

End of Lab

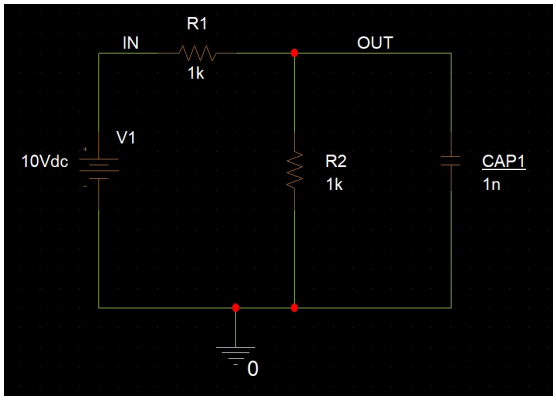
Module 5: Transient Analysis

Lab 5-1 Running Transient Analysis on RC.opj

Objective: To run a transient analysis on the design RC.opj

[In circuit analysis, the aim is to study the circuit completely including its behavior during different states, such as steady and transient. In steady state, a system behaves normally when everything is fine with the circuit. But if any fault occurs/ any input is given to the circuit suddenly/ if any input is removed from the circuit, then for a very small time the circuit goes in *transient state*. Basically, if any change occurs in the circuit then it goes in transient mode. Generally, transients last for very short duration. But it is very important to study that small duration of time. In that small instant of time - current or voltage may rise or drop to a certain value. If that happens, then our circuit must sustain those conditions. Thus, we perform transient analysis on the system.

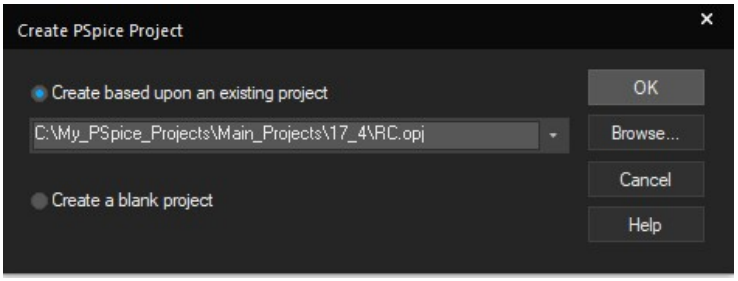
In this lab, you will perform a transient analysis on **RC design** used in previous labs, but with a few changes



Running a Transient Analysis on RC.opj circuit

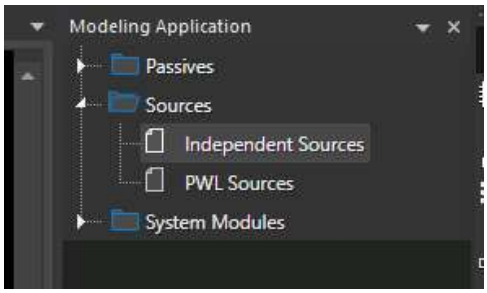
1. Create a new analog or mixed A/D project, ***rc2.opj,*** and save it in the location with other projects used in this training.

2. Create ***rc2.opj*** based on ***rc.opj*** existing project (shown below)

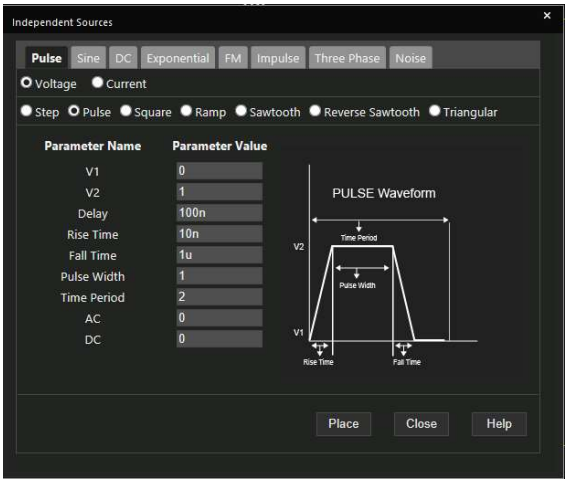


3. We have to replace the DC source with a **Pulse** source.

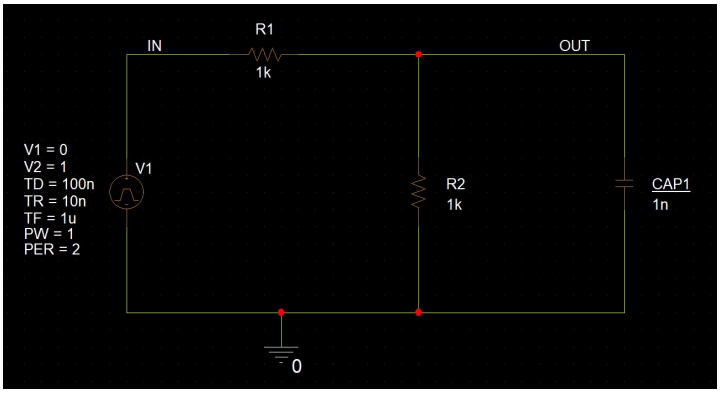
4. Go to **Place – PSpice Component – Modelling Application**. Select Independent source option under sources



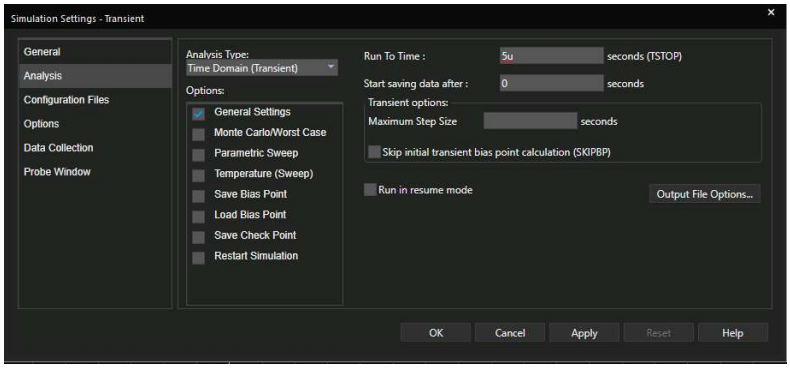
5. Make the selection of a voltage pulse source and enter the below values,



6. Click **Place** to place the pulse source instead of the **DC** source as shown below,



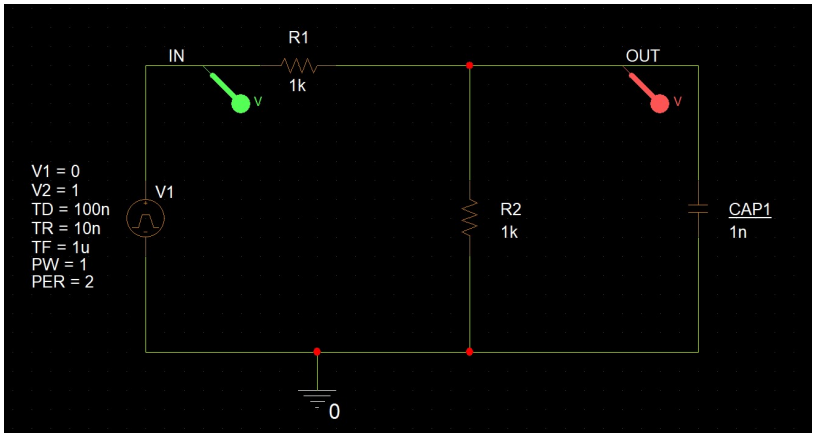
7. From the schematic page, create a RC simulation profile to configure a transient analysis with a final time of **5u.**



8. Click **OK** to save the profile.

9. Click the **Run** icon to run the simulation.

10. If there are no voltage markers on the IN and OUT nets, go back to the schematic and add them now.

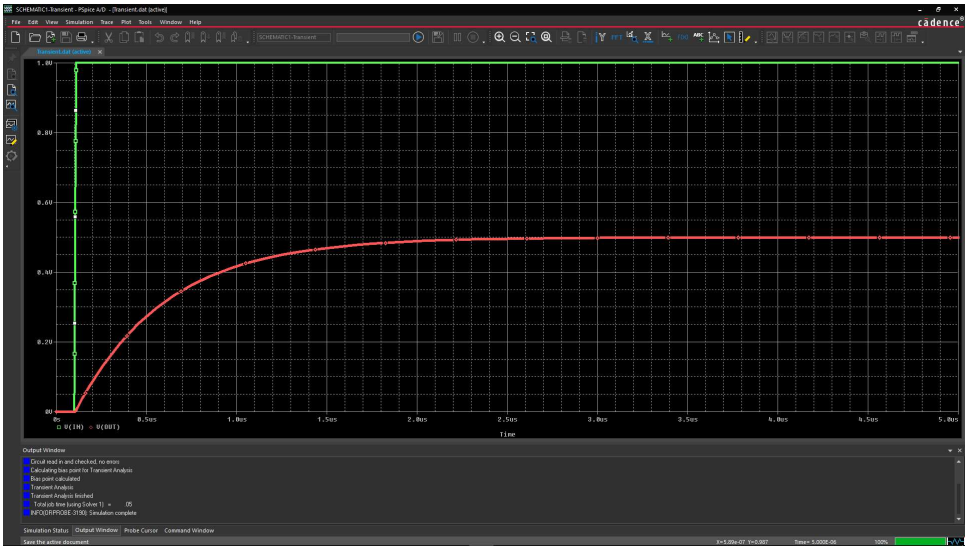


Note: Instead of adding markers to the schematic, you can alternatively add traces (V(IN) and V(OUT)) in the Probe window using the Add Trace toolbar icon.

11. Review the results of the simulation in the Probe window.

12. Save the design file

The waveforms in the following figure show how the voltage across the parallel resistor and the capacitor components ramp up to its steady state value.



Lab Summary

In this lab, you performed the following task:

- Performed transient analysis by changing the source in **RC.opj** design

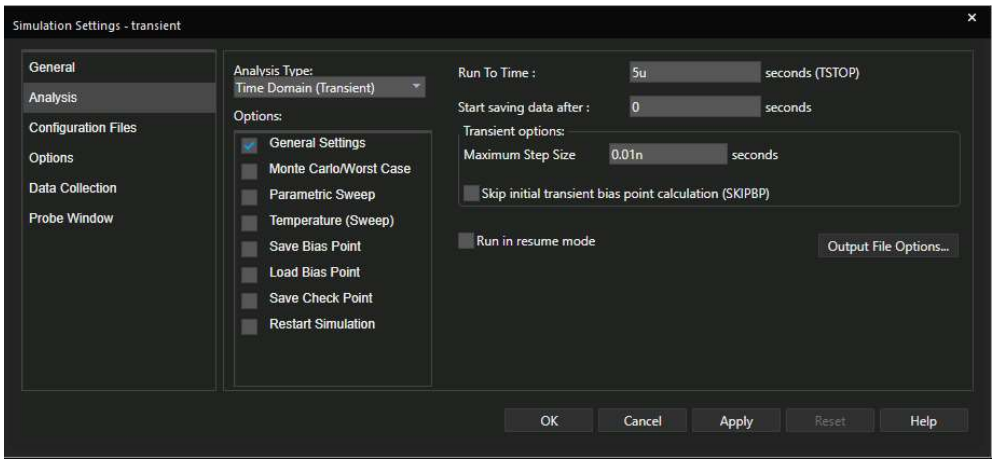
End of Lab

Lab 5-2 Extending a Simulation

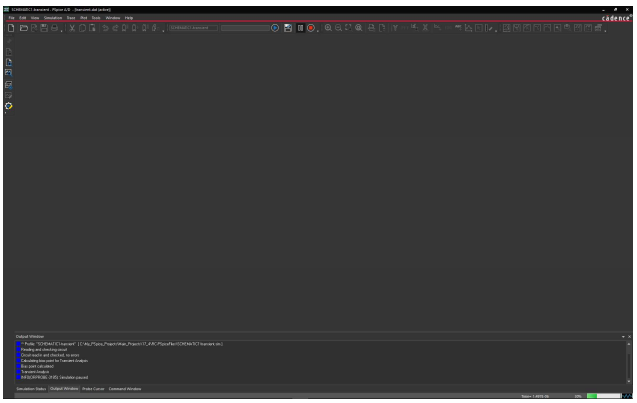
Objective: **Learn how to extend an ongoing PSpice simulation and change runtime parameters during the process.**

Rerunning the RC Transient Simulation

1. Update the simulation profile with the same runtime of 5us and a **Max Step Size of .01ns**. Ensure the stimulus file is configured in the profile. Click OK and run the simulation.

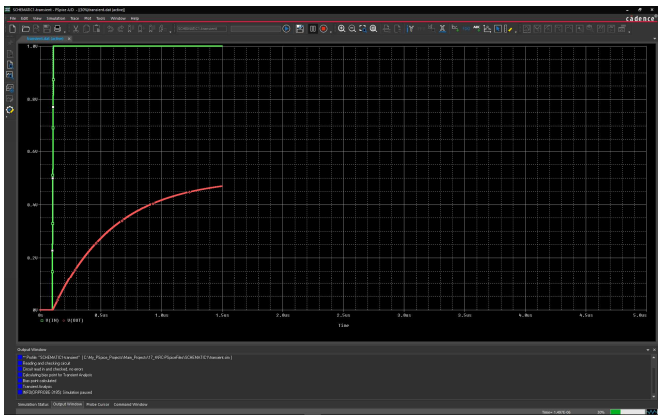


2. When the Probe Window opens pause the simulation at **about 30%.** If the simulation is already completed you can run the simulation again from the Probe window. Use the Pause icon to pause the simulation



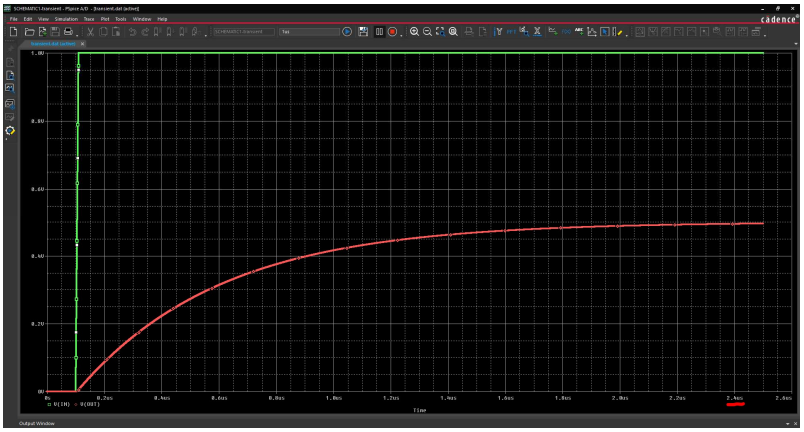
3. This pauses the simulation. Click the View Simulation Results icon ( ), add traces VIN & VOUT to the Probe window.

   The traces are displayed up to the point where the simulation was paused



4. Enter **1us in the Run For field** and then click the ( ) icon to continue the simulation.

The simulation will continue running for another 1us and then pause again. You can click the Run icon multiple times and the simulation will continue for another 1us or what every value you enter in the Run For field.



5. Stop the simulation by clicking the **red stop icon ( )** to the right of the Pause icon.

Lab Summary

In this lab, you performed the following tasks:

- Created a transient simulation profile
- Configured a transient source
- Ran the analysis and examined analysis results in Probe

End of Lab

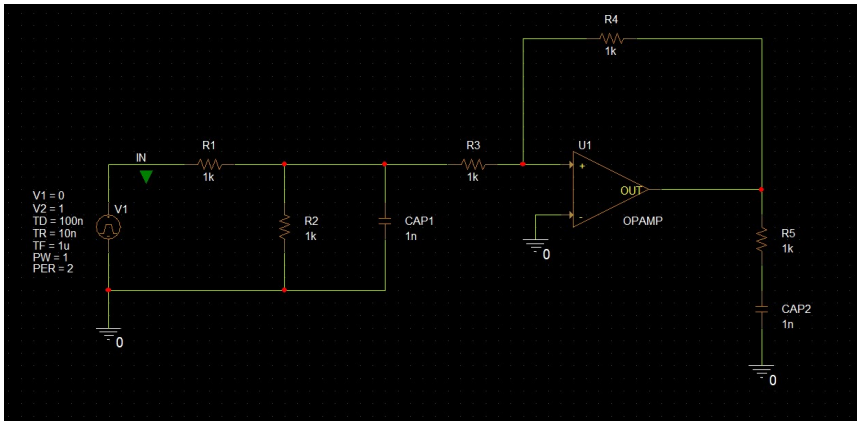
Lab 5-3 Learning Unique Checkpoint Re-Start Feature in PSpice

Objective: Viewing example to demonstrate and understand the usage of Save Check Points and Restart Simulation in a PSpice project.

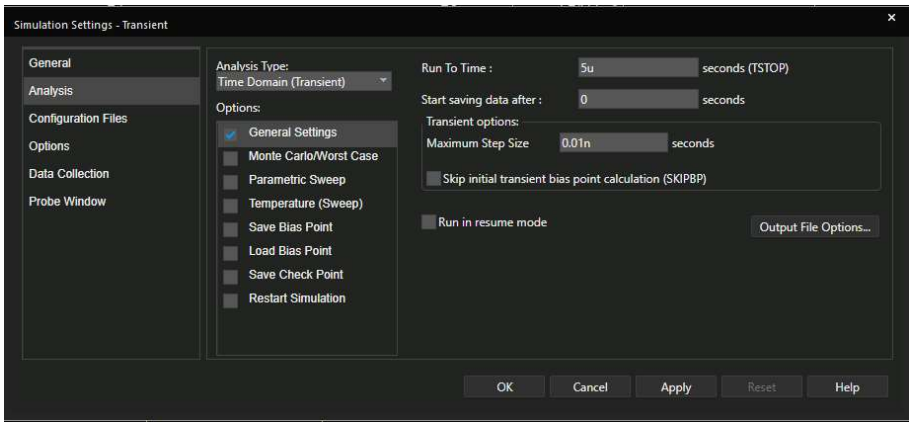
[Often it is necessary to re-run a simulation with different component parameters to get a better result, or to avoid convergence problems. If a simulation takes a lot of time, then it is useful to start the simulation from a specific steady point/state, especially if you are running a long simulation. The following example demonstrates the necessary steps to save time]

Using Checkpoint Restart on the Design Buffer2.opj

1. Open **Buffer2.opj** circuit (shown below)



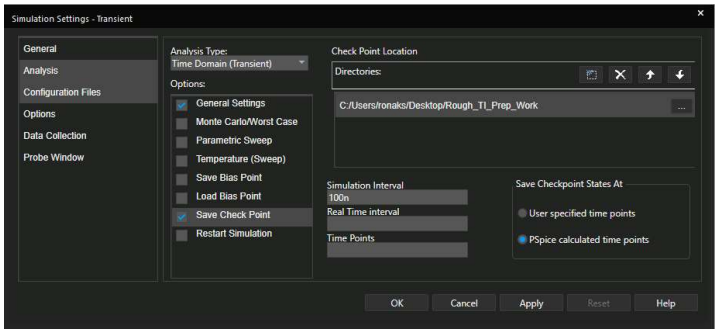
2. Choose **PSpice – Edit Simulation Profile** to edit the current profile & verify the simulation settings.



3. Select the **Save Check Points option**.

   a. Enter **100n** for Simulation Interval.

b. Select **PSpice calculated time points**. Click OK.



4. Choose **File – Save** in the OrCAD® Capture window.

5. Run the simulation.

You have saved the PSpice calculated checkpoints. The checkpoints are used to restart a simulation from one of the checkpoints. This is especially useful for long simulations.

You might want to restart a simulation from a saved checkpoint after changing the design. You can change component and parameter values, simulation, checkpoint restart, and data save options before restarting a simulation.

You cannot add or remove components, change device name or order in the circuit file, or change initial condition of device before restarting a simulation.

6. From the Probe window, click the **Add Trace** icon ( ).

7. Select **I(CAP1)**. Press **OK**.

8. From Probe, choose **Plot – Add Y Axis** to add the second Y-axis.

9. In the Add Traces dialog box, add trace **I(CAP2)**.

10. From the Probe window, choose **File – Save As**. Save the waveform file as **original.dat**.

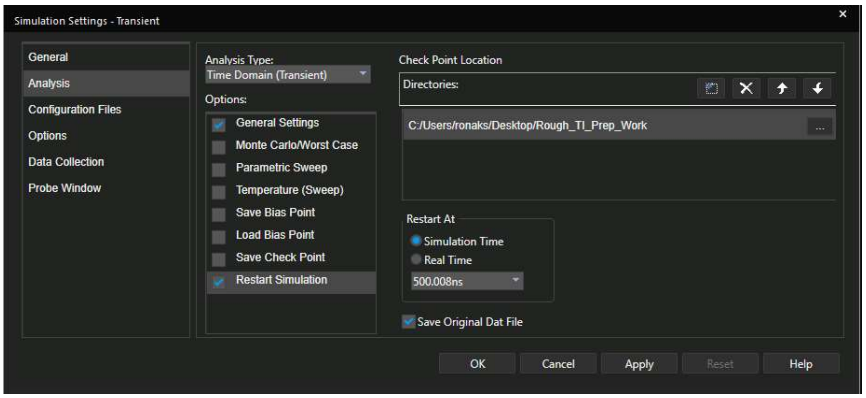
Running a Simulation with Checkpoints

1. Go back to the Capture workspace. Choose **PSpice – Edit Simulation Profile** to edit the simulation profile.

   a. Deselect the **Save Check Points** option.

   b. Select the **Restart Simulation** option.

   c. Select **500.008us** from the **Restart At** pull-down menu. Click **OK**.

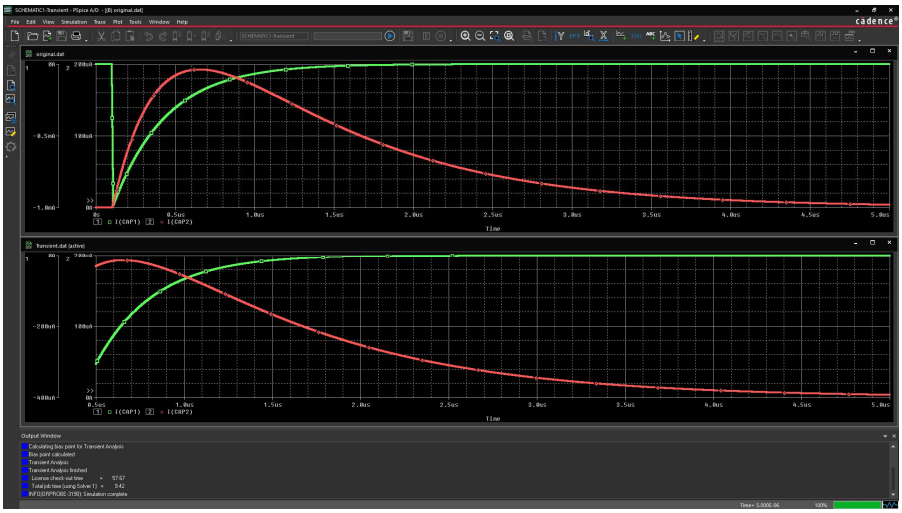


2. Run the simulation.

   Notice that the simulation started at **500ns i.e. 0.5 us** and runs to the specified runtime.

Comparing Results

1. From the Probe window, choose **File – Open**. Double-click **original.dat**.
2. Add traces **I(CAP1) & I(CAP2)** following the same procedure as before
3. From the Probe window, choose **Window – Tile Horizontally**.



Lab Summary

In this lab, you performed the following tasks:

- Specified your checkpoints
- Specified your restart point
- Ran the analysis and examined analysis results in Probe

End of Lab

Module 6: Parametric Analysis

Lab 6-1 Perform Parametric Analysis in PSpice

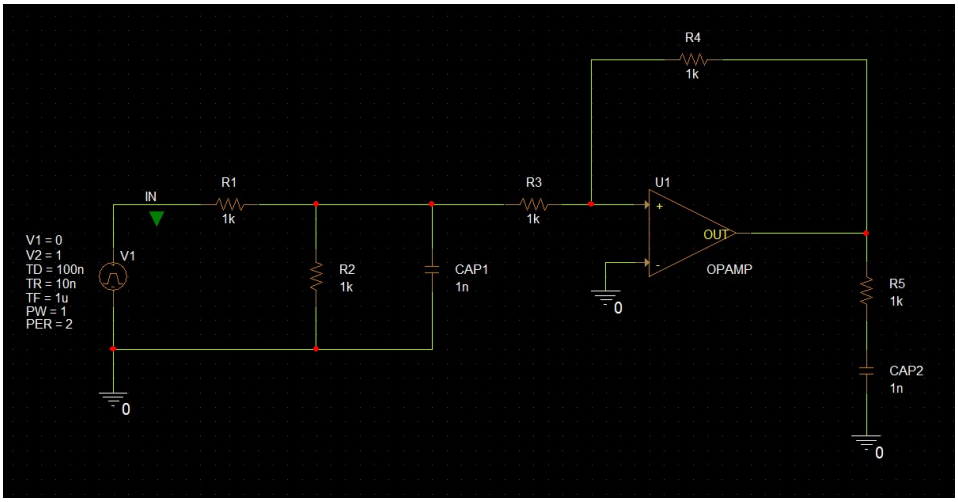
Objective: To define a global parameter, configure a parametric analysis, run the simulation and describe the results in Probe.

In this lab, you

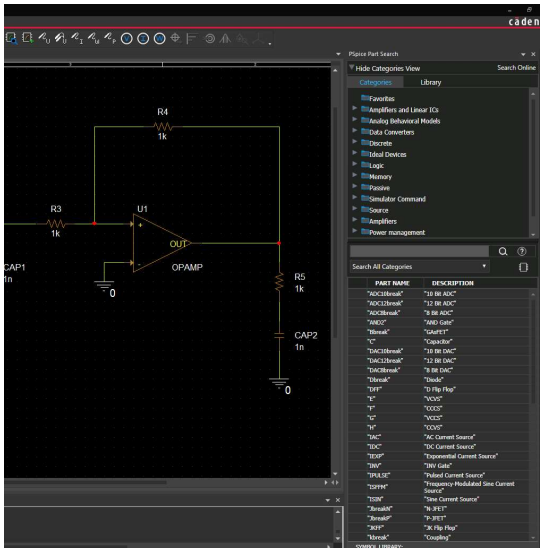
- Define a global parameter
- Configure a parametric analysis
- Run the simulation
- Examine the results in Probe
- Use cursors to examine the results
- Use Performance Analysis to examine the results

Creating and Defining CVAL Part

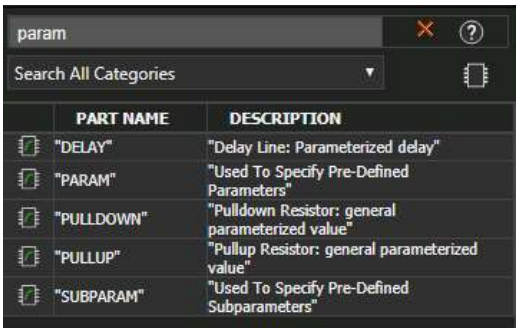
1. Open **buffer2.opj** project.



2. Go to **Place – PSpice Component – Search**. PSpice part search window like the one shown below opens,



3. Search for **Param** in the part search field

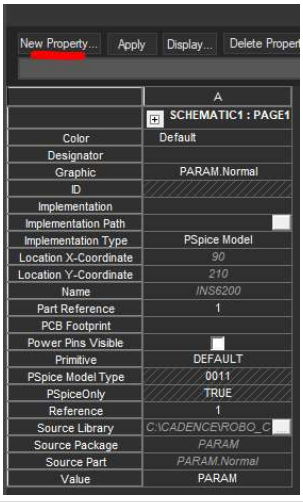


4. Double click on **Param** and place it anywhere on the schematic page.

5. Press **Esc** to end the placement mode.

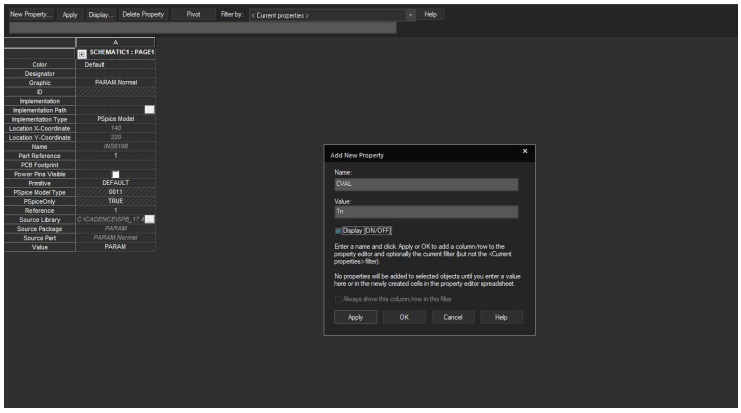
6. Select the Parameter symbol located on the schematic. Open the Property Editor.

7. Click the **New Property** button.

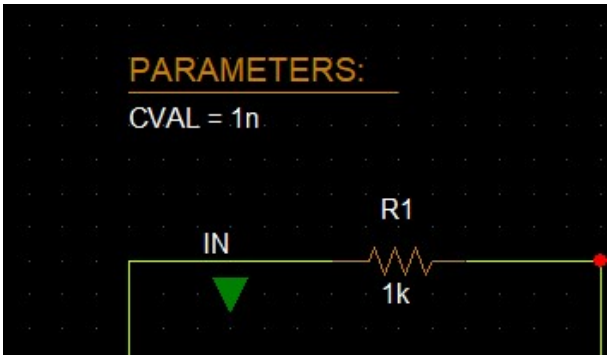


8. Add a **CVAL** property.

9. Set the **value** of CVAL to **1n**. **Enable** Display [ON/OFF]

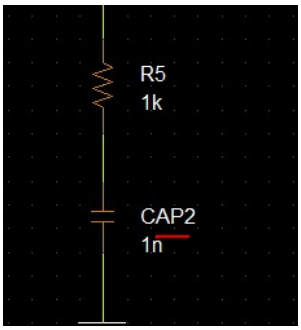


10. Click OK. Close the **Property Editor** window once the property is added



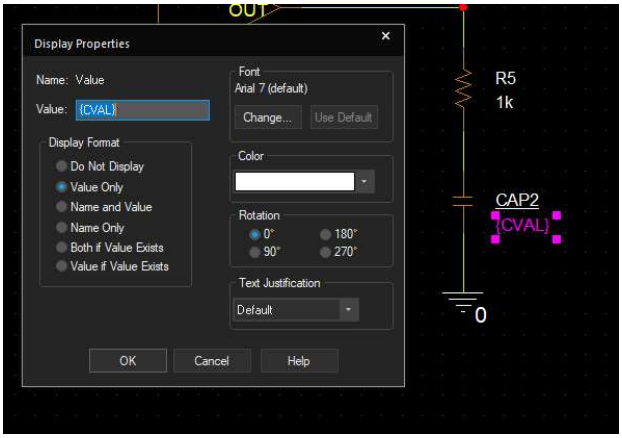
Using a Global Parameter as a Property Value

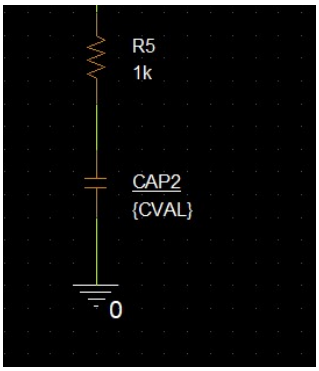
1. **Double**-click the value of **CAP2** (currently **1n**).



2. Enter **{CVAL}** as the new value.

   Note: The curly brackets are required here.



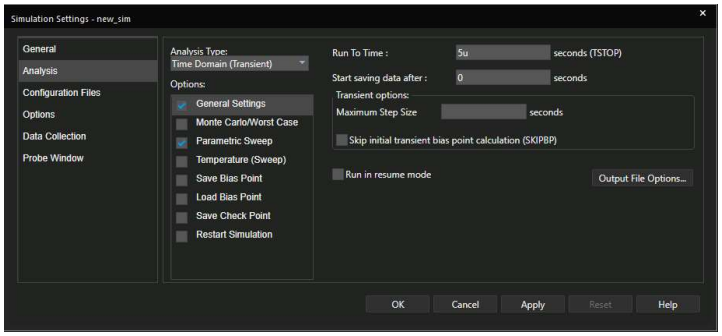


3. Click **OK**.

   [Any time a model parameter or component value (as in discrete components such as resistors, caps, and inductors) is to be assigned at the time of simulation, it is enclosed in curly brackets ({ }). The curly brackets tell PSpice to evaluate the variable or expression rather than using some preset value. For instance, if CAP2 from the last example is given a value of {CVAL} instead of 1n, then PSpice A/D looks for a global variable called *CVAL* and substitutes the current value of CVAL for the value of the capacitor]

4. Create a new simulation profile.

5. On the **Analysis** tab, set the transient analysis settings as shown below,



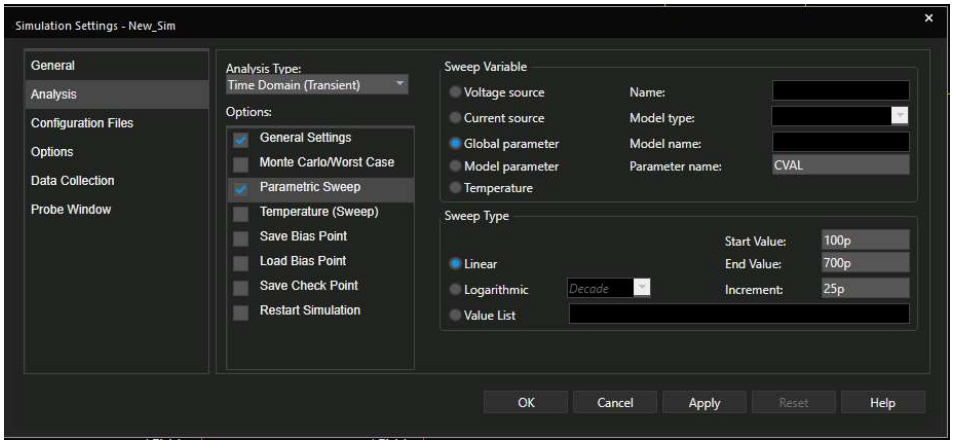
6. Select the **Parametric Sweep** option. Click on **Global Parameter** radio button, which selects all the necessary fields.

7. Enter **CVAL** in the **Parameter Name** field. Notice that it is *not* enclosed in curly brackets.

8. Select the **Linear** radio button in the Sweep type section of the dialog box.

9. Enter **100p** for parameter value in the **Start Value** field. Enter **700p** for the **End Value**. Set the **increment** to **25p**.

10. Make sure that **Time Domain (Transient)** is the selected analysis type.

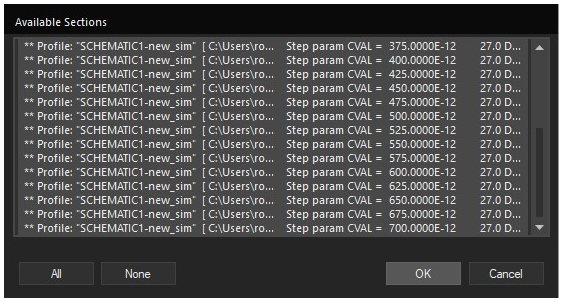


11. Click **OK** to save the changes and close the dialog box.

12. Run the analysis.

Viewing the Output in the Probe Window

Because we have 25 runs of the transient analysis, each with a different value of CAP2, we are presented with the available sections window that lists all 25 runs and the value of CVAL for each. We are given the option to select which runs we wish to view. All are selected by default.



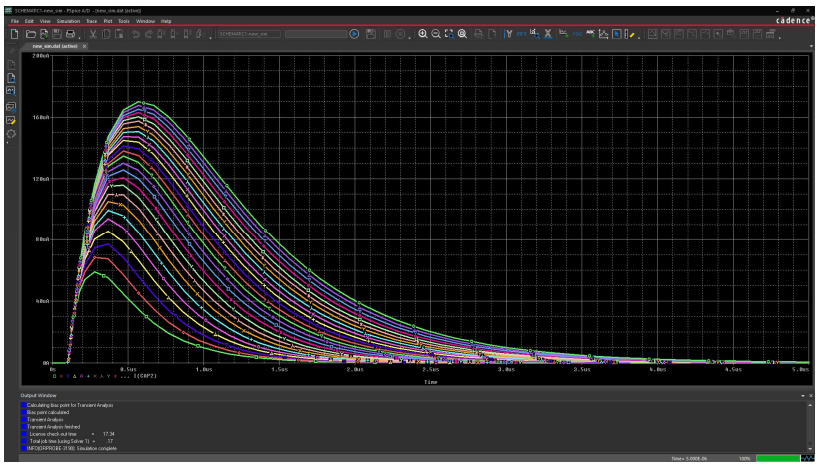
1. Click **OK** to accept the selections.

2. Choose **Trace – Add Trace or** click the **Add Trace** button 

3. Select **I(CAP2)** from the list of traces.

4. Click **OK**.

   All 25 traces are displayed. If there is a current marker on the input pin of CAP2, then the same family of curves is displayed, as well.



Working with Cursors in the Probe Window:

Aim: To compare the current through CAP2 for the first run, where CAP2 is set to 100p, and the last run where is it set to 700p

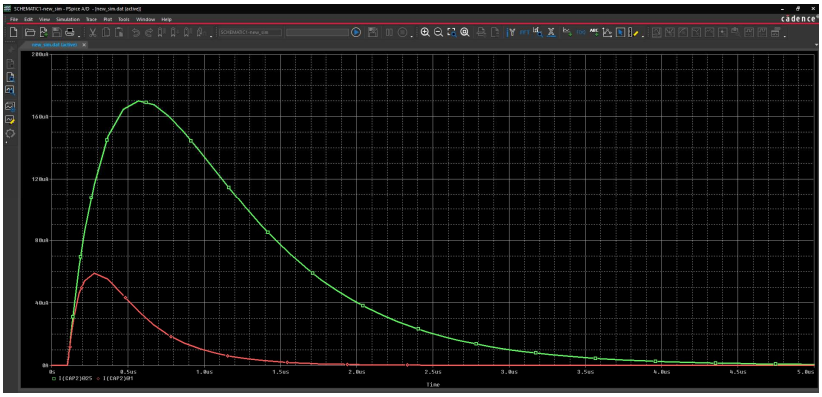
1. Click **I(CAP2)** below the X-axis, and press **Delete** to remove all 25 traces.

2. Click the **Add Trace** button.

3. Enter **I(CAP2)@25 I(CAP2)@1** in the trace command box separated by a space. To avoid entering most of the text, **double**-click **I(CAP2)** from the trace list and then edit the command box field.

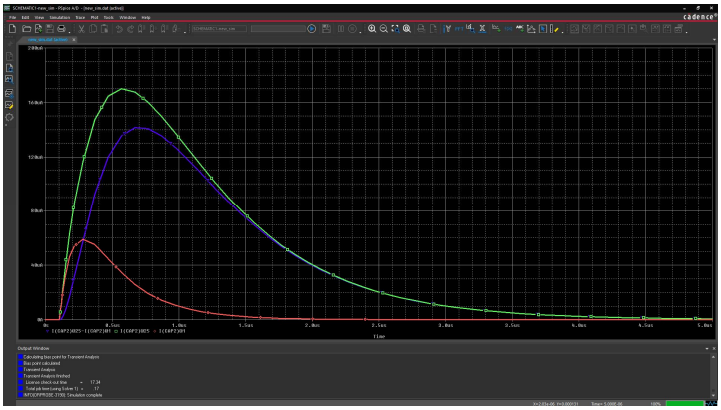
4. Click **OK** in the Add Traces form.

   Notice the difference at the peak value. To quantify the change, plot the difference of the waveforms for runs 25 and 1, and then use the search commands available in Probe to find the exact peak.



5. Click the **Add Trace** button.

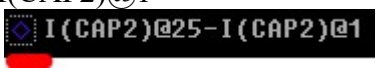
6. Enter the waveform expression **I(CAP2)@25 – I(CAP2)@1** in the Trace Command field. Click **OK**.



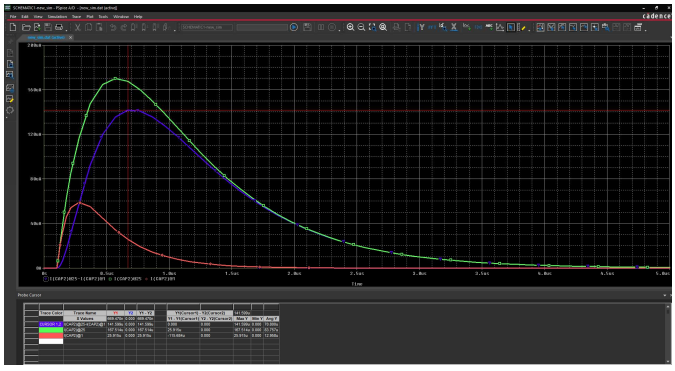
Reading Cursor Measurements

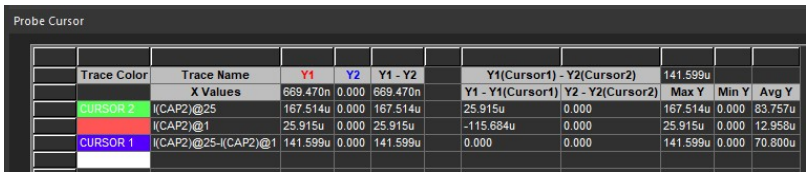
1. Click the **Toggle Cursor** icon ( ) symbolor go to **Trace - Cursor - Display**

2. Click the trace symbol in the **legend** that represents the waveform expression I(CAP2)@25 – I(CAP2)@1



3. Click the Cursor Max icon ( )





The previous figure shows the screen after the pointer is placed at the peak. Notice that the Y value in the cursor box is **141.599u i.e. approximately 142u**. This tells you that when CAP2 is set to 700 pF, the current through CAP2 is 144 uA larger than when CAP2 is set to 100 pF.

Lab Summary

In this lab, you performed the following tasks:

- Defined a global parameter
- Configured a parametric analysis
- Ran the simulation
- Examined the results in Probe
- Used cursors to examine the results

End of Lab

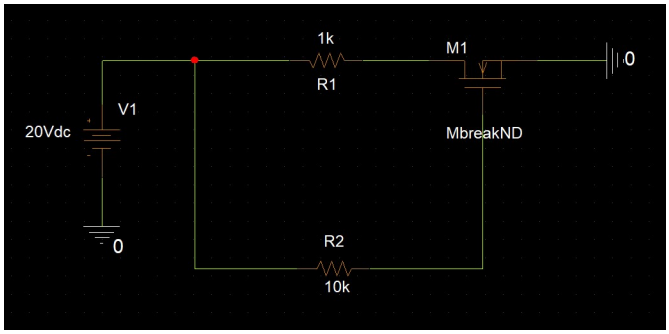
Module 7: Temperature Sweep Analysis

Lab 7-1 Temperature Sweep analysis with PSpice

Objective: **Learn to perform a Temperature Sweep on a circuit in PSpice**

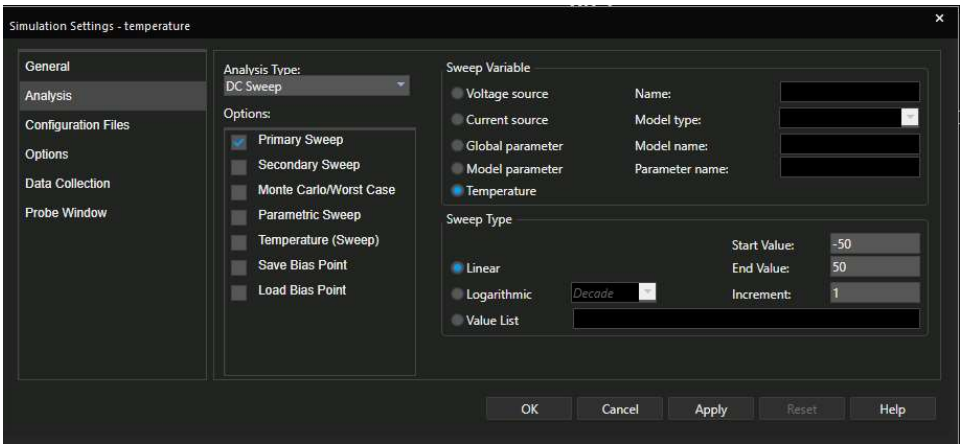
Example Circuit

1. Open **temperature.opj**

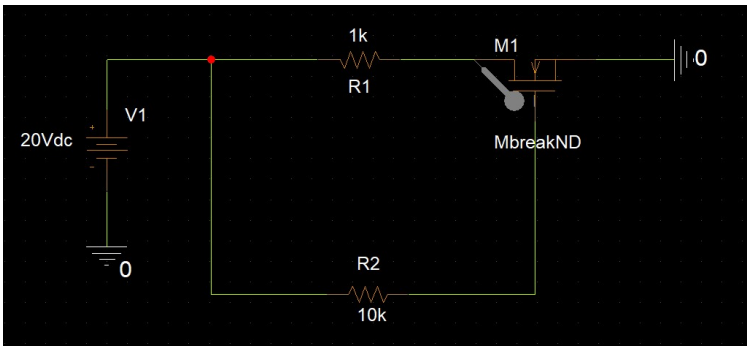


Configuring a Simulation Profile

1. Create a simulation profile to perform a DC sweep of temperature from -50°C to 50°C in 1° increments. The following shows the simulation settings.



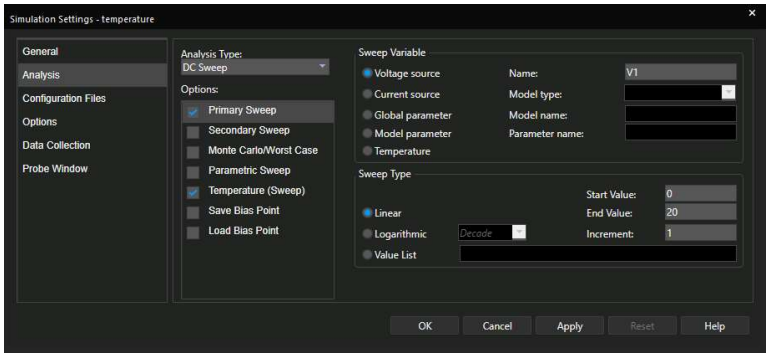
2. Add a current marker as shown in the figure below,



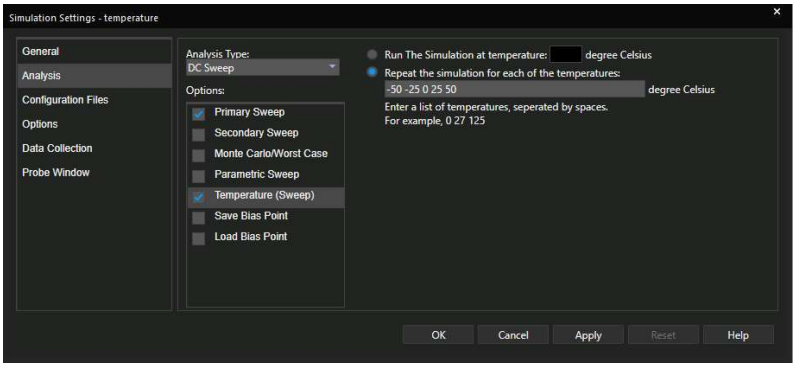
3. Run the simulation and examine the output.

Performing temperature as a secondary sweep

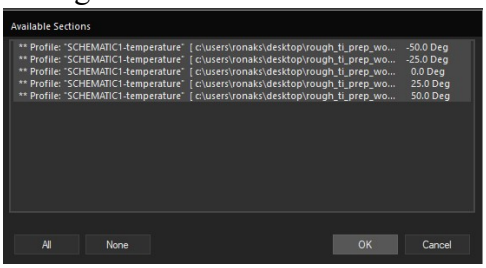
1. Edit the simulation settings for DC primary sweep as shown below



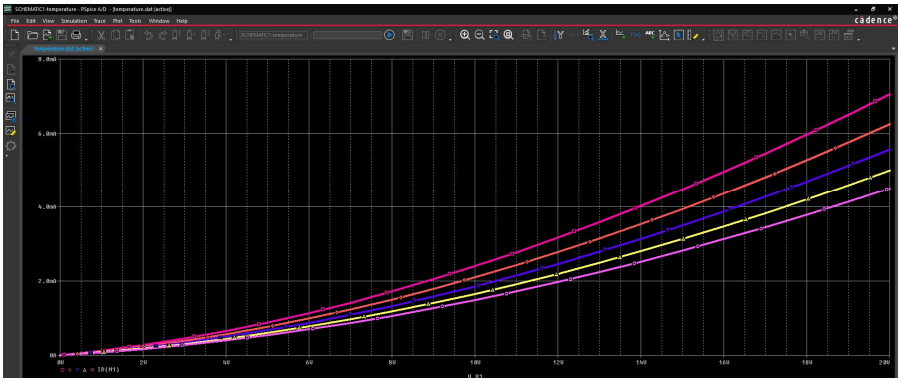
2. Set the temperature sweep to run specifically at temperatures -50°C, -25°C, 0°C, 25°C, -50°C



3. Click **OK** to save simulation settings and run the simulation.



4. Select all the available sections as shown above and click OK.



Here, we see one response for each of the different temperatures that we selected.

Lab Summary

In this lab, you

- Learned to perform temperature sweep in PSpice

Going a Little Advanced

Module 8: Making a Circuit Non-Ideal (Adding Parasitics)

Lab 8-1 Modelling non-ideal inductors and simulating a circuit with PSpice

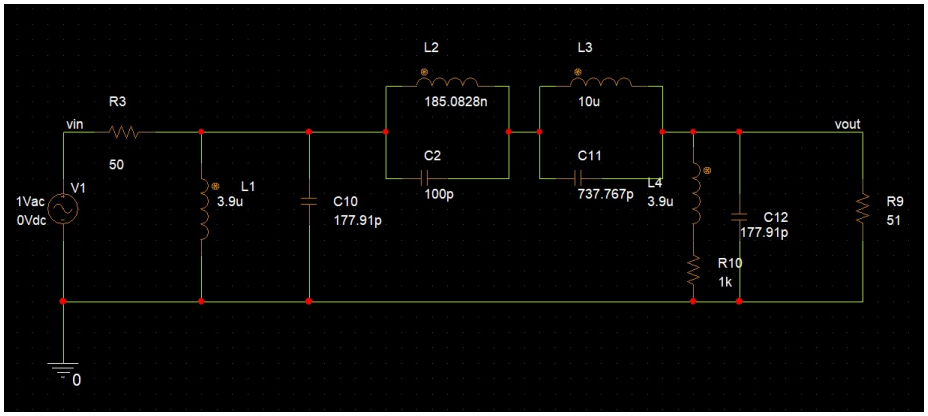
Objective: Learn to use the modelling app in PSpice to add parasitics to circuit components on the fly

---

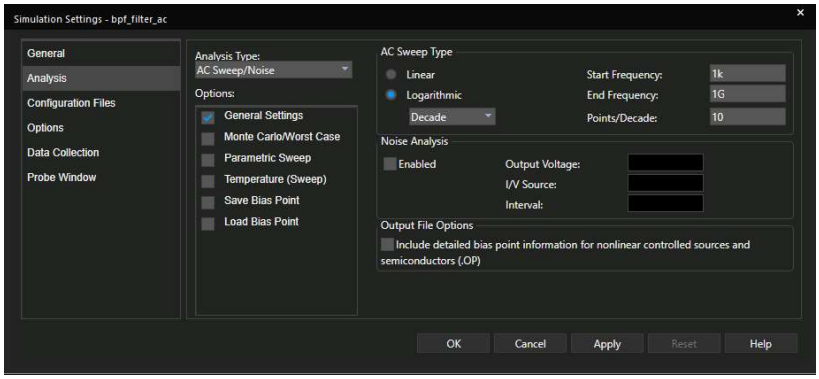
PSpice allows you to model selected non-ideal components in your circuit on the fly via the Modelling App. To see a list of the components you can model non-ideally, go to – **Place > PSpice Component > Modeling Application**. We will model non-ideal inductors in this module.

Example Band Pass Filter Circuit:

1. Open **PSpice_Lab_2.opj** circuit



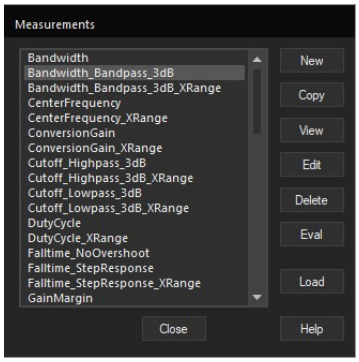
2. Review the AC simulation settings,



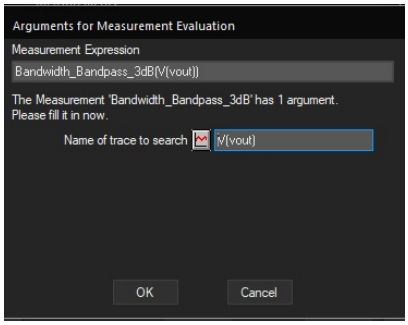
3. Run PSpice simulation. Go to **View - Measurement results** and note the simulation results,

Evaluate	Measurement	Value
☑	Bandwidth_Bandpass_3dB(V(vout))	23.67267meg
☑	CenterFrequency(V(vout),3)	15.05661meg
☑	Cutoff_Highpass_3dB(V(vout))	3.22028meg
☑	Cutoff_Lowpass_3dB(V(vout))	26.89295meg

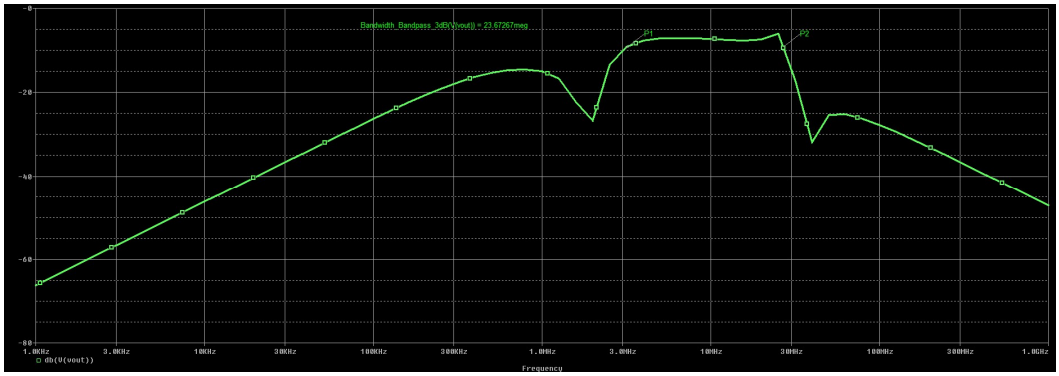
4. Go to **Trace – Measurements**. Select **Bandwidth_Bandpass_3dB** from the list and click on **Eval**.



5. Search for the trace **(V(vout))** and select it,



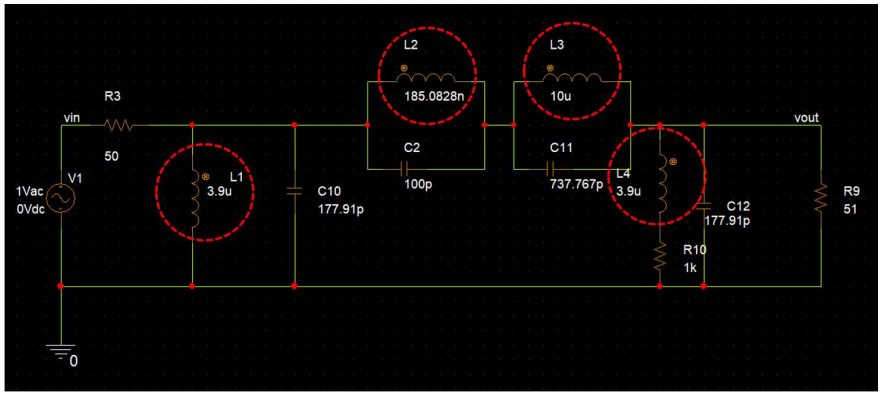
6. A graph will be displayed as shown in the figure below, with a **bandwidth value = 26.99862 meg**



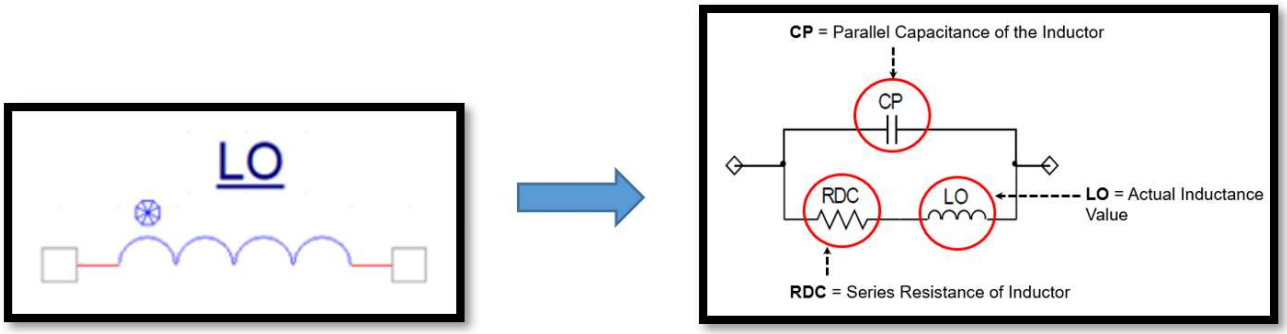
7. Go back to the schematic. Now, lets replace the ideal inductors with their non-ideal counterparts:

   The PSpice modeling app enables a designer to quickly model a non-ideal inductor device using component datasheet. We will use datasheet 1, 2 and 3 for the filter inductors L1, L2, L3, L4 as highlighted in the figure below to make them non-ideal.

   (Note, that the appropriate datasheets of these Inductors have already been found for you in this lab)

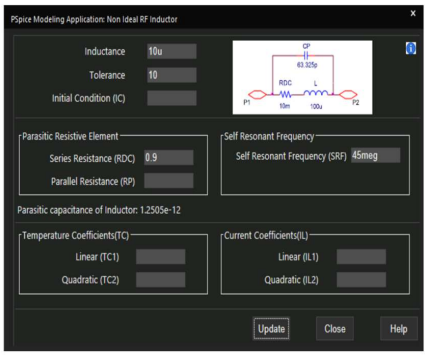


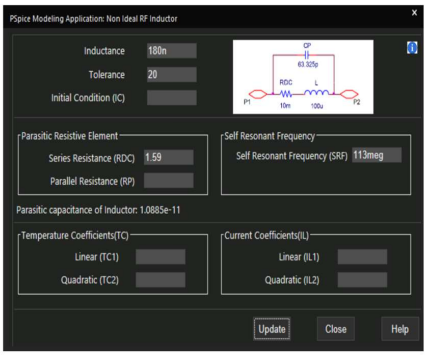
8. Replacing ideal inductors with non-ideal counterparts:

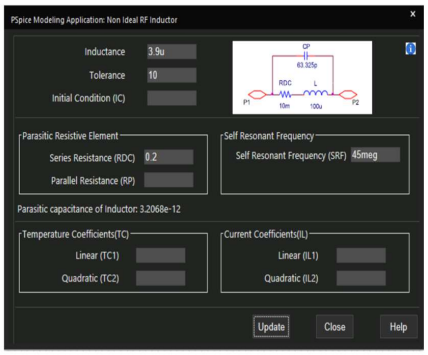


   Open the modelling app., go to **Place > PSpice component > Modelling Application > Passives > Inductor**

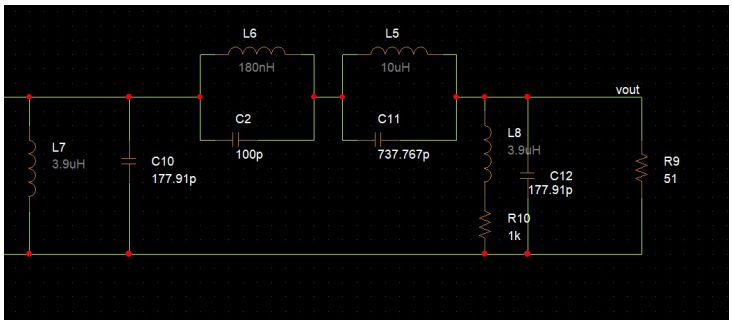
9. Place 3 inductors using modelling apps. with the values as shown below,







10. The circuit schematic should look like the figure below,



11. We have now replaced all the ideal inductor components with their non-ideal counterparts. Run AC analysis simulation again and observe the change in measurement results from the previous results.

	Evaluate	Measurement	Value
	☑	Bandwidth_Bandpass_3dB(V(vout))	23.51299meg
	☑	CenterFrequency(V(vout),3)	14.77247meg
	☑	Cutoff_Highpass_3dB(V(vout))	3.01598meg
	☑	Cutoff_Lowpass_3dB(V(vout))	26.52897meg

The change in measurement results become increasingly important & evident when this filter is working in conjunction with a load, say RF amplifier. Thus, taking into consideration the component parasitics is very important.

Lab Summary

In this lab, you

- Learned to use the modelling app and added parasitics to the inductors in the circuit. We further examined the change in measurement results after the parasitics were added.

End of Lab

Module 9: Simulating a Text Netlist in PSpice

Lab 9-1 Creating a Text Netlist

Objective: **To create a circuit file in a text editor, simulate the circuit file in PSpice A/D, and describe the results in Probe.**

[A SPICE netlist is a text-based representation of a circuit.

Viewing the netlist helps you to learn about SPICE syntax and simulation. It can also help in identifying simulation errors and convergence issues]

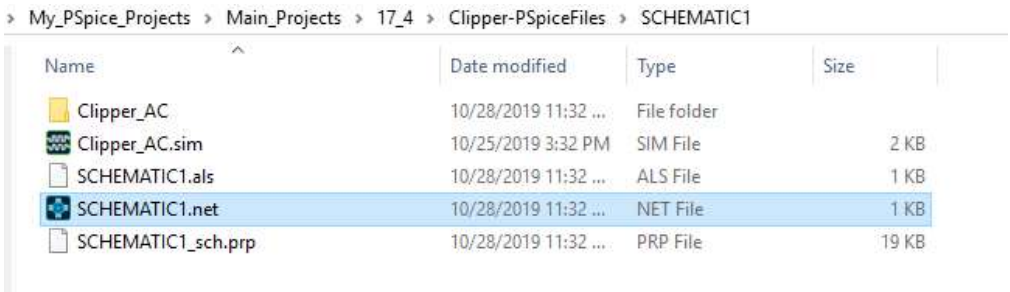
In this lab, you will create a circuit file in a text editor and then simulate the it with PSpice Simulator . Later, you will examine the waveform results

Creating a PSpice Netlist

You can manually create a circuit netlist and simulate it with the PSpice simulator. Rather than have you spend the time entering the text that describes a circuit, you can use the netlist created by Capture for the clipper circuit created in the previous lab.

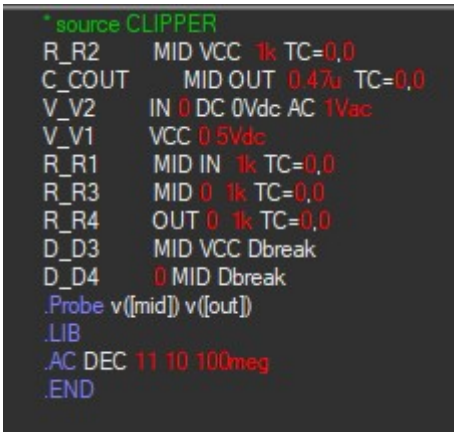
1. From PSpice A/D, choose **File – Open-Text File**

2. Go to the **Schematic1.net** file in your Clipper project directory and select it



3. After the Schematic1.net file is opened the netlist will look like as shown below. Copy this netlist by pressing **Ctrl+C**

4. As shown in the figure above, add all the statements of **.PROBE, .LIB, .AC, .END** at the end of the netlist



[Add a .PROBE statement that contains no voltage or current statements. This Probe statement writes only those output variables specified to the data file to restrict the size of the data file (a maximum of 8 variables can be listed). This illustrates how to specify a node that has a name rather than a number. The square brackets force the interpretation of names to mean node names

If you set PROBE/CSDF, this creates a data file in a text format with the Common Simulation Data File (CSDF) format, not a binary format. This format is used for transfers between different computer families. CSDF files are larger than regular text files.

Add a library call for *nom.lib*. (You can use a LIB command with no arguments because *nom.lib* is assumed.)

The LIB statement indicates which libraries are to be searched for the models used. The absence of a library name tells the simulator to use the *nom.lib*. This library file is the master library file and contains references to all parts initially installed with the tool]

Add an AC sweep analysis specification with the following format:

```
.AC DEC 11 10 100meg
```

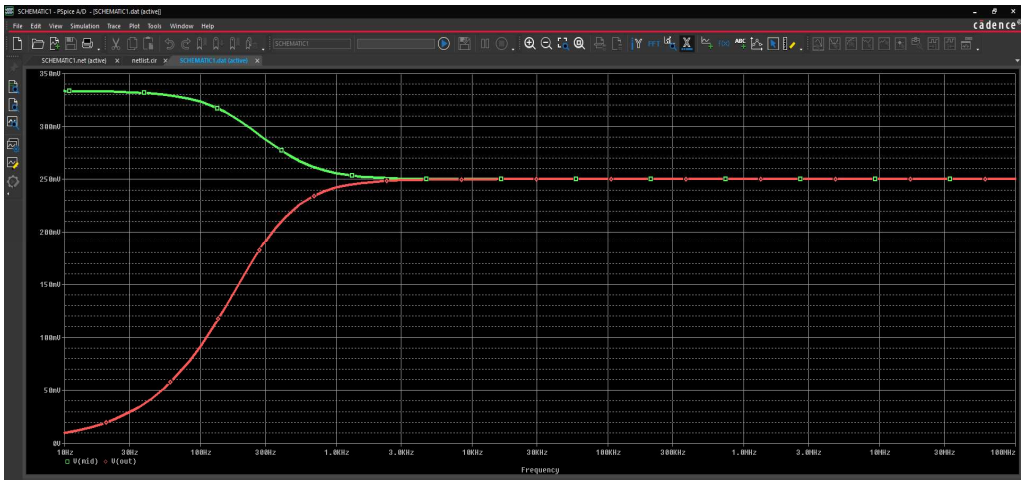
The .AC sweep analysis specification defines the simulation to be performed.

End the file with an END statement. The END statement ends the circuit definition]

5. Save the file as *CLIP_NL2.CIR*.

Running PSpice A/D with the Netlist

1. Choose **Simulation – Run** *or* click the **Simulate** icon to start the simulation. Add traces of V(mid) and V(out)



Lab Summary

In this lab, you performed the following tasks:

- Created a circuit file in a text editor
- Simulated the circuit file with TI-PSpice
- Viewed the results in Probe

Module 10: Resolving Convergence Errors

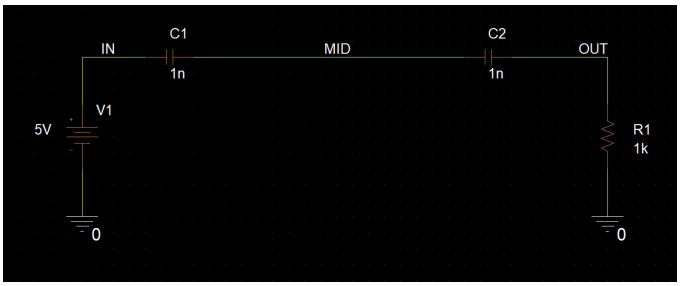
Lab 10-1: After completing this lab you will be able to solve floating node errors in PSpice

[All circuit nodes in any Spice circuit must have a DC path to "0". Else, there is a DRC that gets flagged when the netlist gets created.

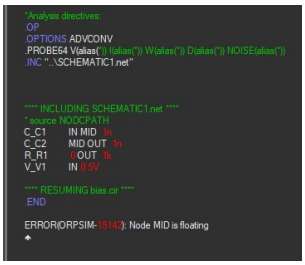
Since you have a GND net, make sure to place a PSpice specific "0" symbol.]

Using nodcpath Project:

1. Open the project called **nodcpath.opj**



2. Run the simulation and observe the errors.



A floating node error always indicates that there is no DC path to ground for the named node. That can be caused by either a missing ground (or a ground that is not named 0), by a node being isolated from ground by capacitors or other devices that do not conduct DC current.

Correcting Errors:

1. Add a 100G resistor between MID and ground.

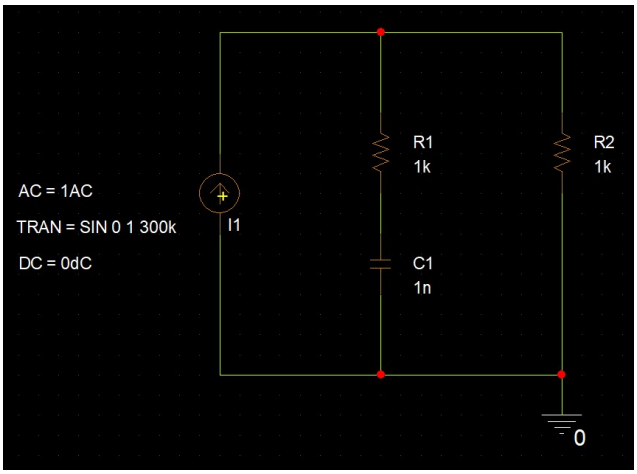
2. Re-simulate the circuit. The simulation should now run successfully

End of Lab

Lab 10-2 Enhancing Waveform Resolution

Objective: After completing this lab, you will be able to solve resolution problems

Using stpclg Project

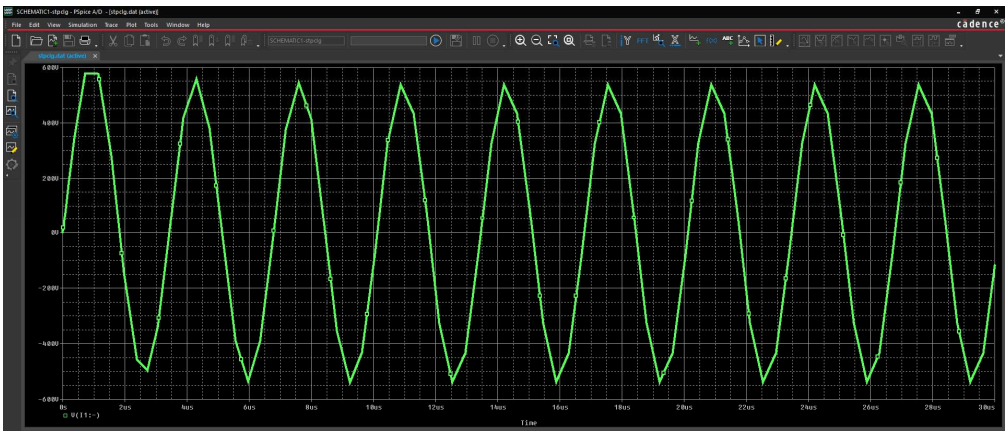


Setting Up and Running Simulation

1. Create a simulation profile and name it ***stpclg***. Set up a transient simulation with a final time of **30u**.

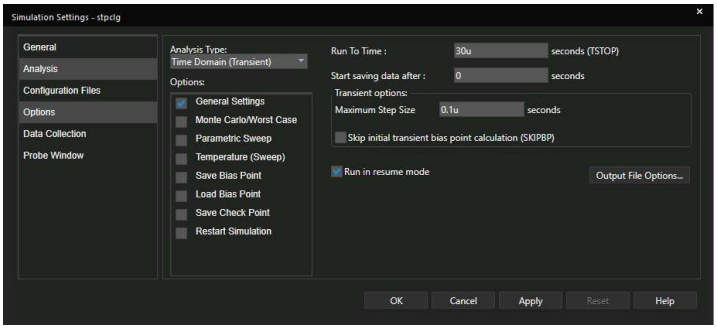
2. Simulate the design and examine the result in the Probe window. Plot the **V(I1:-)** waveform.

   Although the simulation runs successfully, the resulting waveform is very rough. A smooth sine wave was expected.

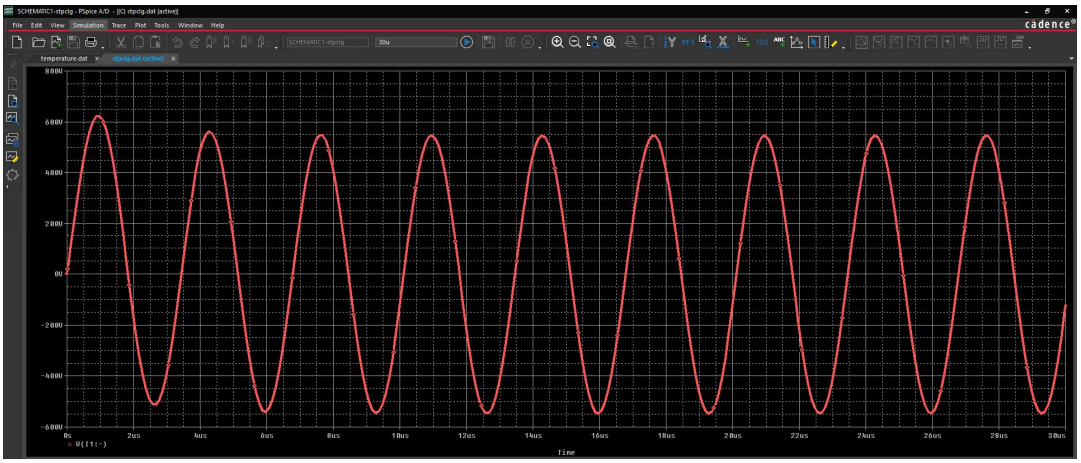


Correcting Errors

1. Edit the simulation profile. Set **Maximum Step Size** to **.1u**.



2. Rerun the simulation. Examine the resulting waveform in the Probe window. The sine wave is now smooth. The resolution of the displayed waveform is increased by setting a maximum step size, that the simulator can handle.



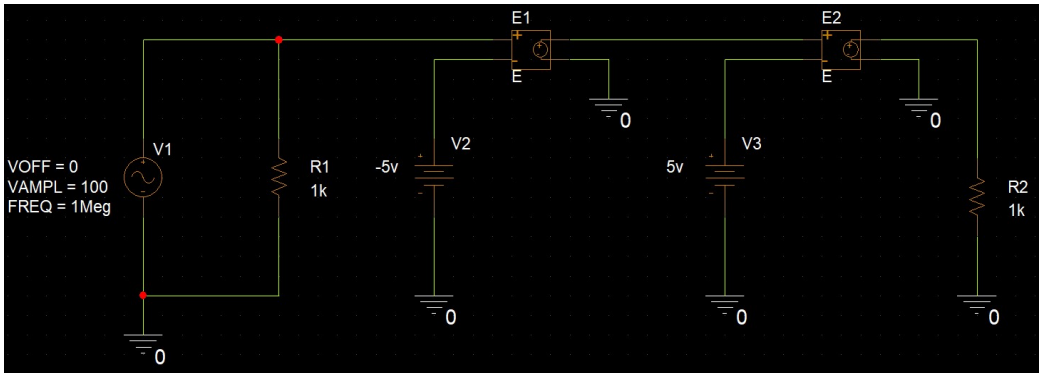
End of Lab

Lab 10-3 Solving Voltage & Current Limit Problems in PSpice to achieve convergence

Objective: To solve a current & voltage limit problem taking example of a circuit

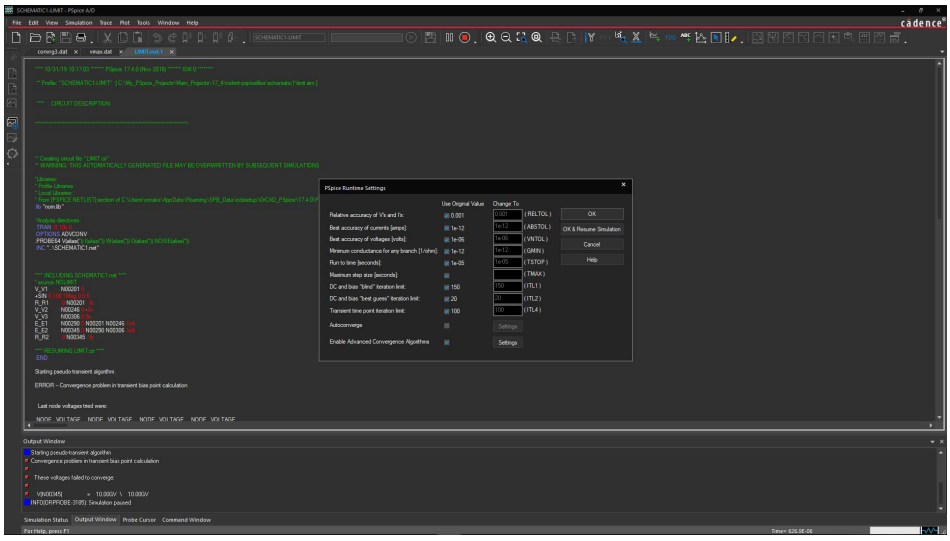
Example Circuit

1. Open the ***Nolimit.opj*** project.

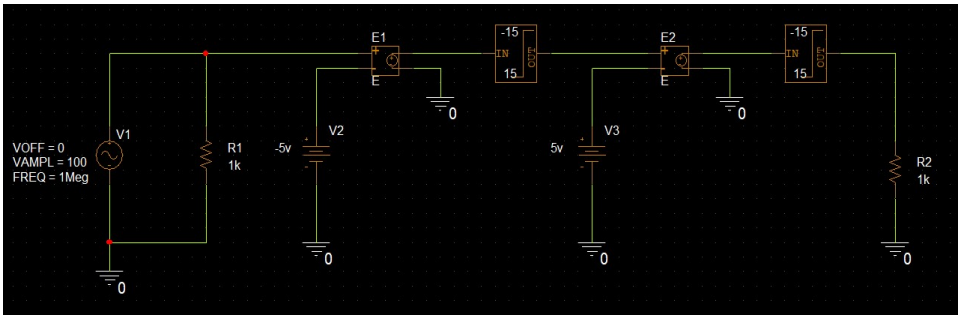


2. Examine the circuit and the simulation profile. The source is a 1 MHz sine wave with an amplitude of 100V. We are running a 10us transient analysis. Each of the controlled sources has a gain of 1E6.

3. Run the simulation. A convergence error occurs where a source does not converge



Read the error message: Voltages and currents are limited to the range +/- 1e10. This is especially important to remember when simulating with Analog Behavioral Modelling (ABM) devices. To limit the output of E1 and E2, **add LIMIT device as shown in figure below**.



4. Find a **LIMIT part** in the ABM library. Add the limit device after each of the controlled sources. The limits should be **–15 to 15** for each occurrence of the limit block.

	A
	SCHEMATIC1 : PAGE1
Color	Default
Designator	
Graphic	LIMIT.Normal
HI	15
ID	
Implementation	
Implementation Path	
Implementation Type	PSpice Model
LO	-15
Location X-Coordinate	510
Location Y-Coordinate	160
Name	INS1189
Part Reference	LIMIT1
PCB Footprint	
Power Pins Visible	
Primitive	DEFAULT
PSpiceOnly	TRUE
PSpiceTemplate	E^@REFDES %OUT 0 VALU
Reference	LIMIT1
Source Library	C:\PSPICE_TN\PSPICE_
Source Package	LIMIT
Source Part	LIMIT.Normal
Value	LIMIT

5. Add a voltage Probe at the output node to view the waveform in Probe.

6. **Rerun** the simulation

   The unlimited controlled sources sharply increase. This is a common situation when using ABM devices.

7. Close all open projects.

In this lab, you

- Solved a limit problem

End of Lab

Module 11: Measurements with PSpice Probe using Performance Analysis

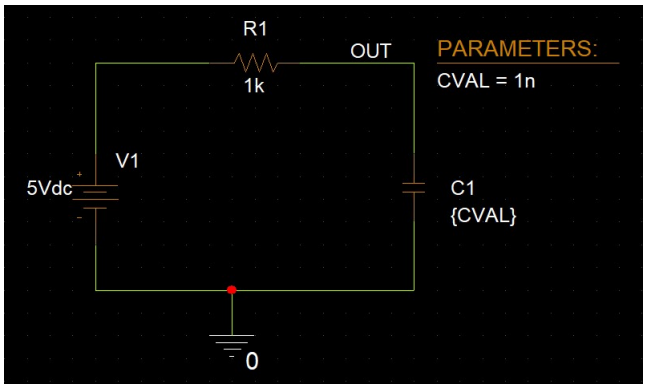
Lab 11-1 Using the Rise-Time Measurement

Objective: To use a simple RC circuit to demonstrate the rise-time goal function

---

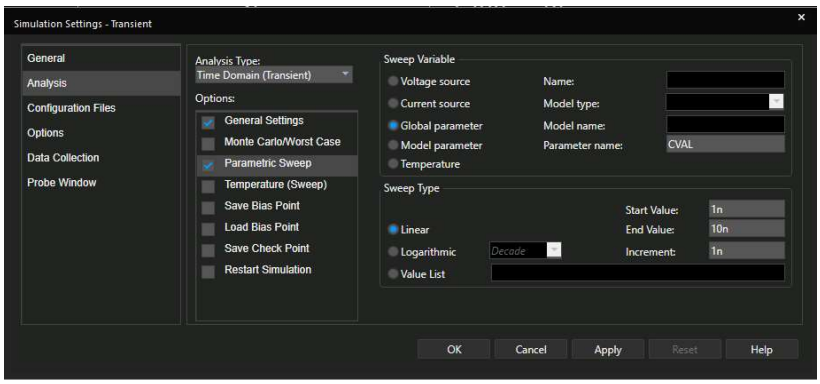
[Probe is very useful as a waveform viewer to check the output of a simulation. In addition, it has a Performance Analysis feature, which allows it to be much more versatile and useful in evaluating the behavior of a design, especially in a multi-section simulation. Multi-section simulations are simulations that perform a variable sweep such as a parametric or Monte Carlo analysis]

1. Create a new PSpice project. Name it ***Goals.opj***. Create the following design:

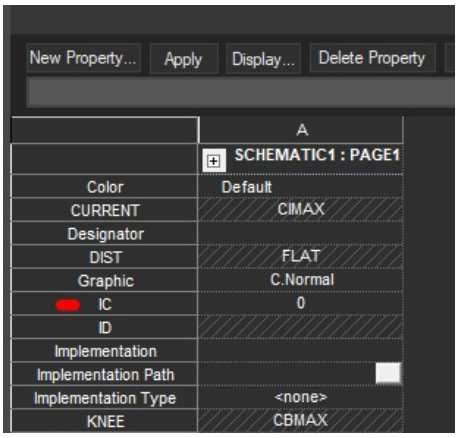


Create a Transient Analysis Profile

1. Configure a transient analysis running to **10us** and a parametric analysis that sweeps CVAL from **1n** to **10n** in **1n** increments.



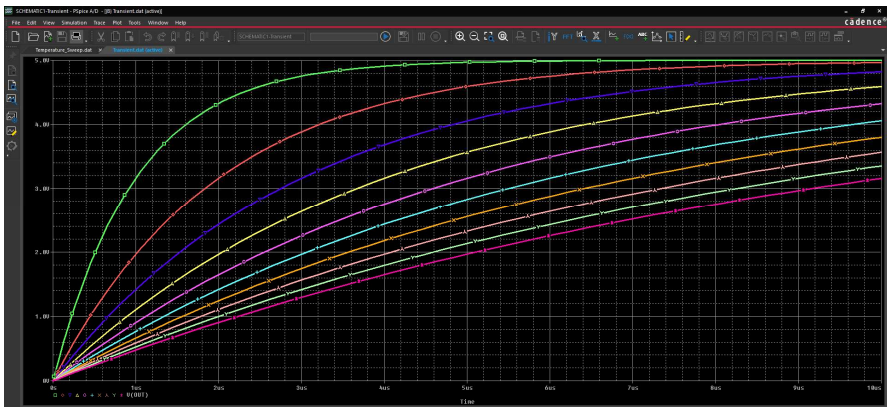
2. Set the Initial Condition (**IC)** attribute to **0** on the capacitor. If you do not set an initial voltage across the capacitor, then the capacitor is seen as an open circuit during the bias point calculation. V(OUT) is set to 5V, which then is used as the starting point for the transient simulation.



Running the Simulation

1. Run the simulation.

2. When Probe opens, click **OK** and then display the trace V(OUT).



Configuring and Running Performance Analysis

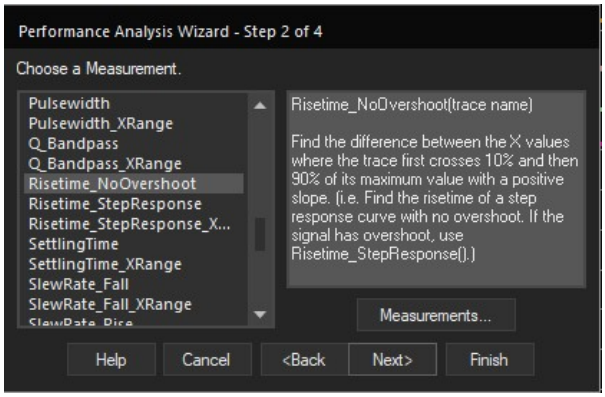
1. Choose **Trace – Performance Analysis**.

   The Performance Analysis dialog box opens showing 10 data sections available, the parameter CVAL as the swept variable with values ranging from 1nF to 10nF.

2. Click the **Wizard** button to run the Performance Analysis wizard.

3. In the Performance Analysis Wizard – step 1 of 4 dialog box, click **Next** to continue through the wizard.

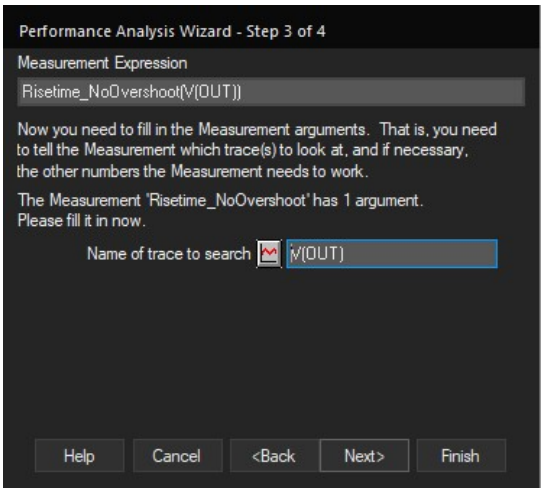
4. In the Performance Analysis Wizard – step 2 of 4 dialog box, scroll down the list of goal functions on the left and select **RISETIME_ No Overshoot** to provide a detailed description of the goal function (on the right)



5. Click **Next**.

6. In the Performance Analysis Wizard – step 3 of 4 dialog box, click the **Add Trace (**  **)** button to display the list of available traces.

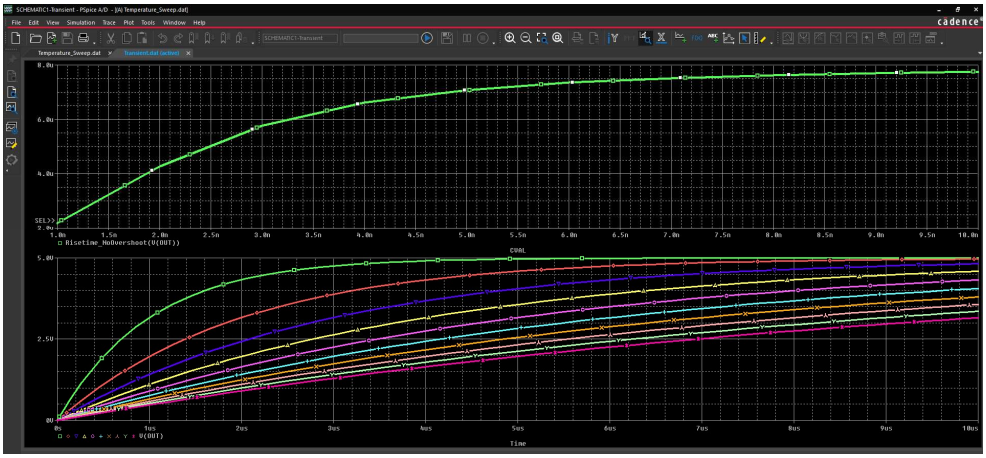
7. Select **V(OUT).** Click **OK**.



8. Click **Next**.

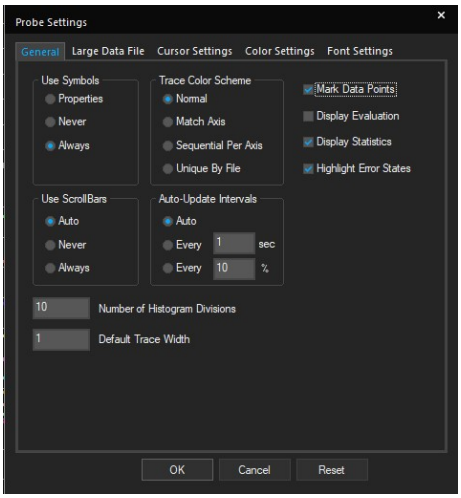
   Probe tests the goal function on the first data section to see if the goal function is valid for this trace. It also marks data points 1 and 2.

9. Click either **Next** or **Finish** to complete the wizard and display the results of the goal function.



The rise time increases as the value of the capacitor increases.

If you want to see the actual data points used to create the curve, choose **Tools – Options.** Select the **Mark Data Points** option.



Lab Summary

In this lab, you performed the following tasks:

- Used a simple RC circuit to demonstrate the use of the risetime goal function

End of Lab

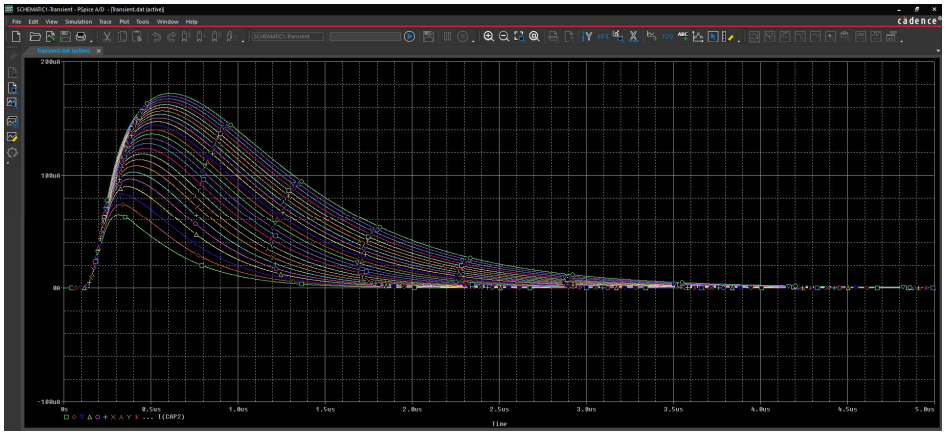
Lab 11-2 Create an expression in PSpice to measure Pulse Width

Objective: To create a measurement function that measures the pulse width of the output of a buffer circuit.

Example Circuit: Buffer Project

1. Open the **Buffer2.opj** project.

2. Open the Probe window and open the .dat file from the project directory. Because the parametric simulation data was created earlier, you do not need to rerun the simulation.



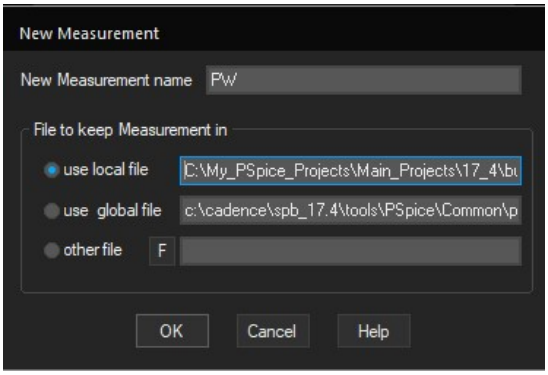
Aim: You want to create a goal function that measures the pulse width of each of the above traces. Define the pulse width to be the difference between the X-values of the trace measured at the 50% of each maximum Y-value

Defining the Measurement

- The name is PW
- It takes a trace name as the single argument
- The marked point expression is x2-x1 (greater X value minus the lesser X value).
- There need to be two search functions. The first searches for the 50% point of the trace on a positive slope; the second searches for the 50% point on the negative slope

1. In Probe, choose **Trace – Measurements**.

2. Click **New** in the Goal Functions dialog box.



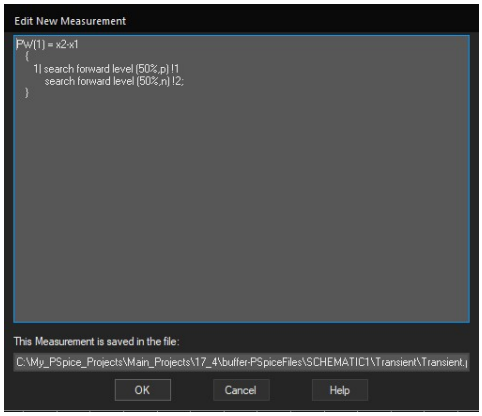
3. Enter **PW** in the **New Measurement Name** field.

4. Click the **Use Local File** radio button.

   This stores the new goal function in a PRB file that has the same name as the active profile.

5. Click **OK**.

6. In the Edit New Measurement dialog box, enter the marked expression as **x2-x1**.



7. In the Edit New Measurement dialog box, enter the marked expression as **x2-x1**.

   a. Change the search function to the following:

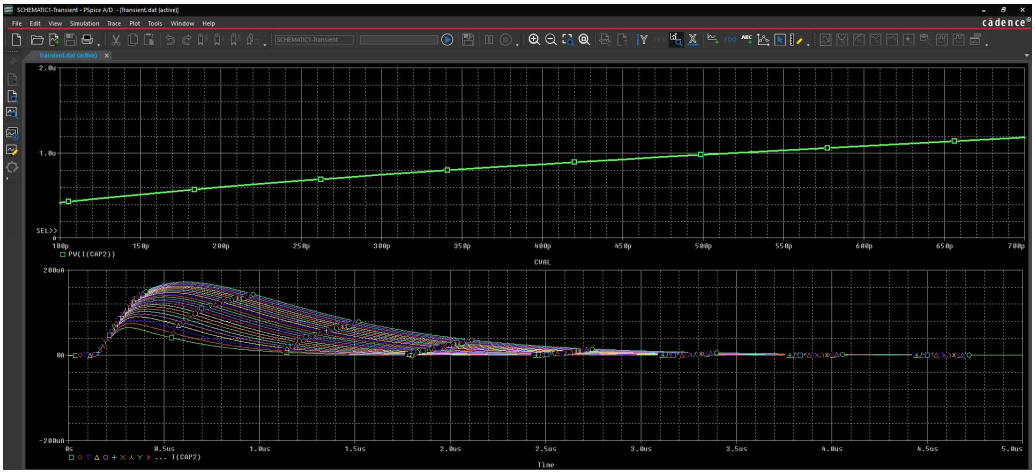
```
1| search forward level
(50%,p) !1 search forward
level (50%,n) !2;
```

Note: You must always place a semicolon at the end of the function.

8. Click **OK** to save the function and close the dialog box.

Using the New Goal Function

1. Click the **Performance Analysis** icon ()
2. Click the **Trace Add** button.
3. Scroll down the list of goal functions and select **PW(1)**.
4. Click the trace name **I(CAP2)** to place it in the argument.
5. Click **OK** to add the trace.



We used a simple example, but the above steps are the same for more difficult functions.

1. Decide what you want to measure.
2. Examine the waveforms you must determine, which points on the trace are needed to calculate the measured value.
3. Compose the search commands to find and mark the desired points.
4. Use the marked points in the marked point expressions to calculate the final value for the trace.
5. Test the search commands and measurements.

Lab Summary

In this lab, you performed the following tasks:

- Created a measurement function to measure the pulse width of the output of the Buffer circuit that you created earlier

Module 12: Importing 3rd Party Vendor Models from Web in PSpice

Lab 12-1 Understand the process of using 3rd party vendor models in PSpice

Objective: Learn how you can quickly download a model from Internet & use it in PSpice simulation

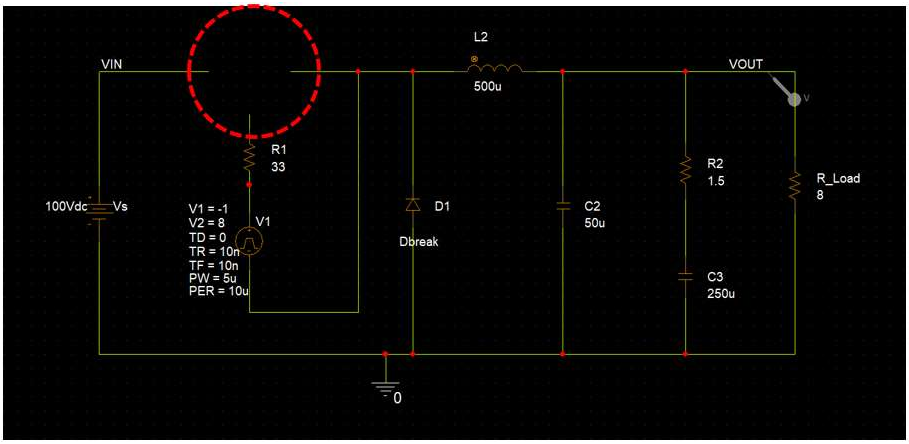
---

[PSpice is the only Spice simulator that allows you to download any component right from the web into your design environment from any third-party model vendor and use it in a design to simulate]

Simple plug & play steps to use any model from Internet in PSpice simulation

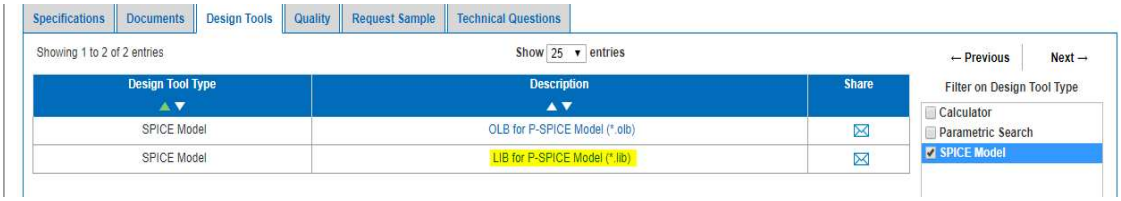
1. Open **model_import.opj**

2. Review the circuit schematic of the project,



3. Notice that we have left a space open for a component to be connected. We will be connecting a power MOSFET here

4. Visit this webpage - https://www.vishay.com/mosfets/list/product-91070/tab/designtools-ppg/. Scroll down and download the PSpice model for IRF840

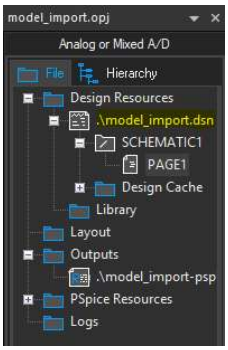


5. You will notice that a text file opens in other tab. Copy the text content and paste it in a notepad file. Save the notepad file as **pwr_mos.lib** and save it in a desired location.

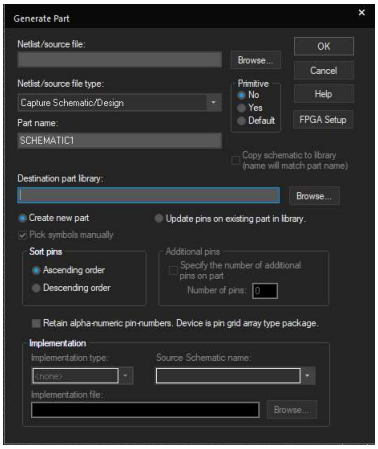
The content of the .lib file of the component contains the necessary PSpice netlist information required for it to work with PSpice.

6. But, along with a .lib file – we need a schematic symbol that represents this component on the schematic. You can quickly create a schematic symbol for a component and associate the .lib file to it using **Generate Part** functionality

7. Click on **.dsn file** from the project manager window as shown below,

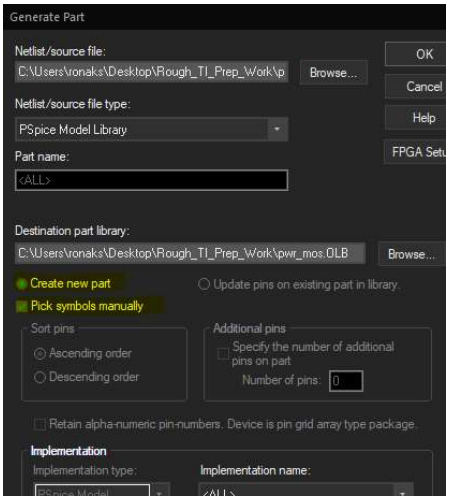


8. Go to **Tools – Generate Part.** A window like the one shown below opens,

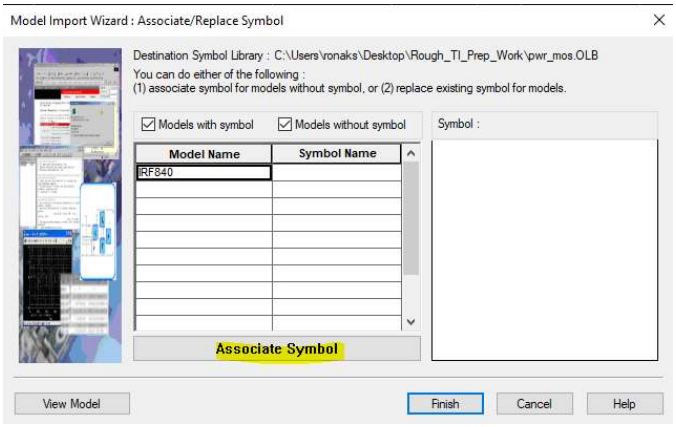


9. In the **netlist/source file option**, browse for the netlist file (pwr_mos.lib) we previously saved and select it
   The **destination part library path** of the part is selected automatically as the same one of the .lib file

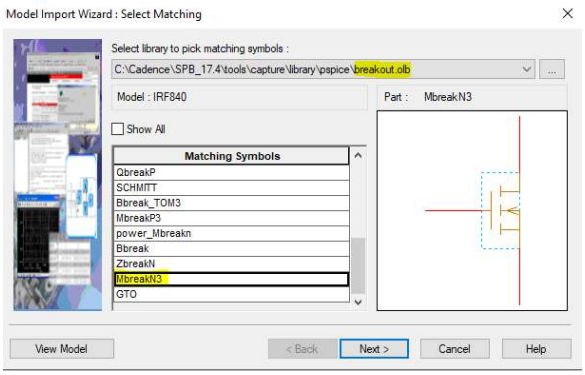
10. Select **Create new part & Pick symbol manually** options as shown below,



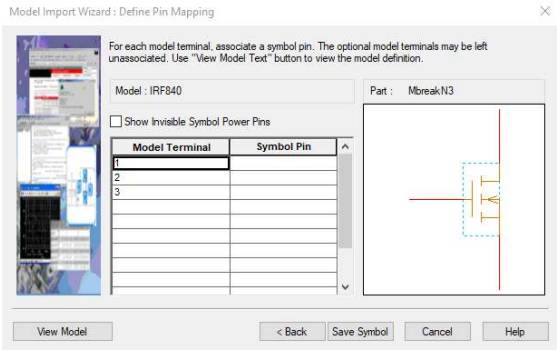
11. Click **OK.** This will open a model import wizard where you can associate a symbol with the model



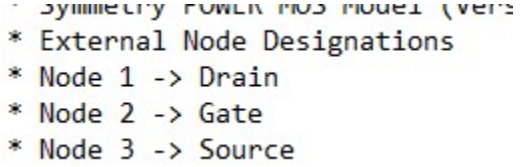
12. Click on **Associate Symbol** option. Select a library part from PSpice installation to which you want the new part to resemble. Go to the location of **breakout.olb** in your library directory and select it. From the list of suggested matching symbols, select **MbreakN3**



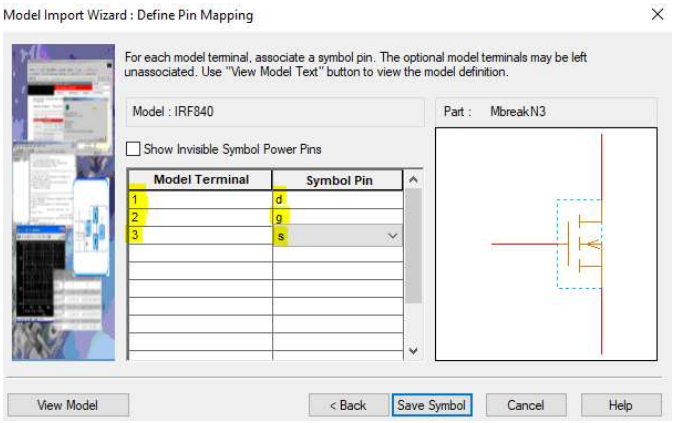
13. Click on **next**. A window pops up asking you to define the pin mapping for your new component.



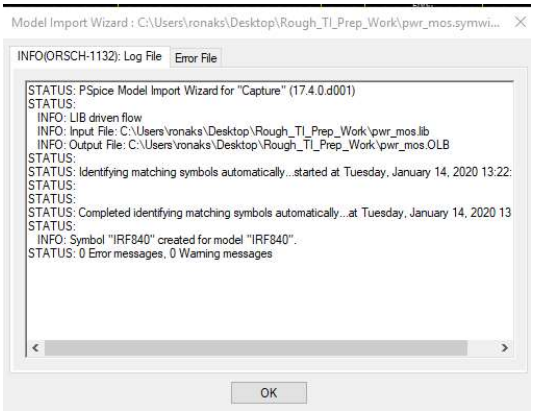
Per the information present in the netlist of the component (text file that we downloaded from internet), note the pin mapping



14. Assign the symbol pins to model terminals 1,2 & 3 accordingly. Verify that your model terminal & symbol pin mapping looks the same as shown in figure below,

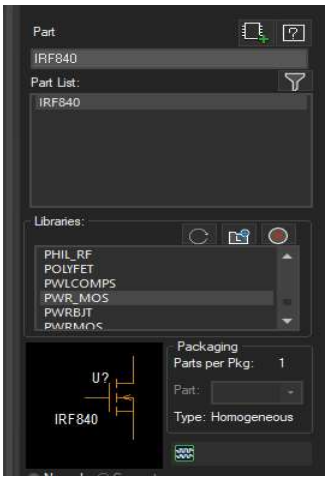


15. Click on save symbol and then finish.



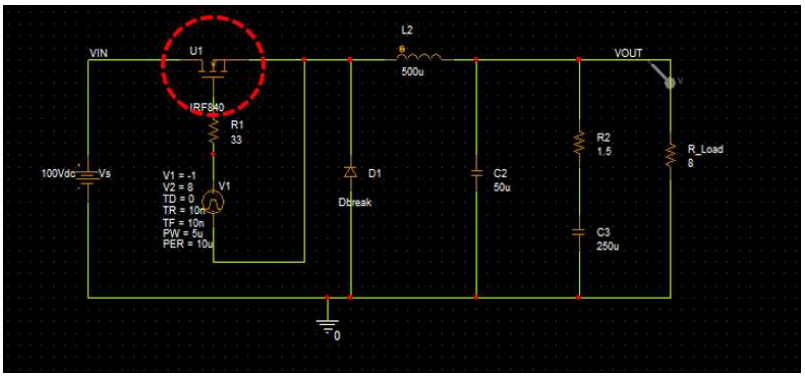
You have successfully created a symbol based on the downloaded library

16. Go back to Capture window. Select the add library icon and browse to the schematic symbol .olb file we just created.



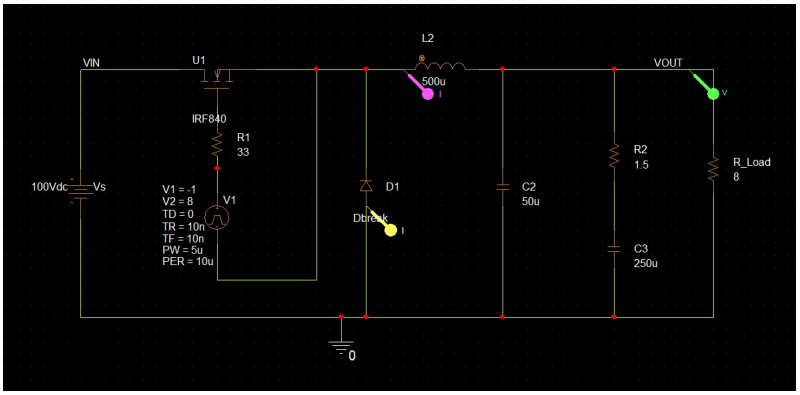
17. Once you load up the .olb file, the part **IRF840** shows up in the list

18. You can quickly double click on it and place it on the schematic.

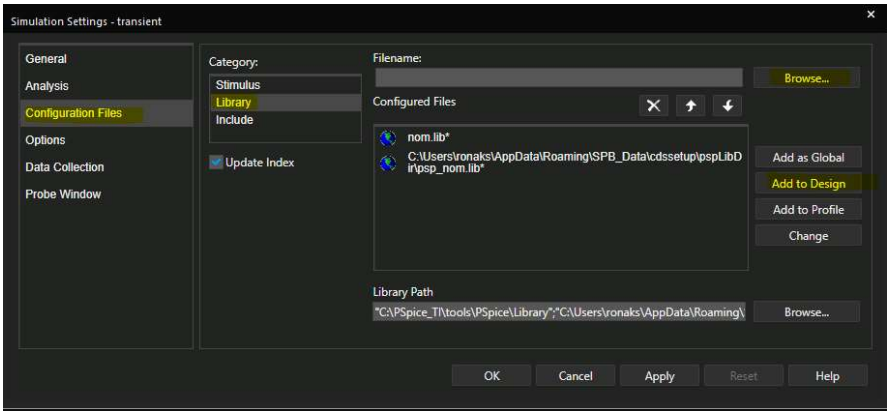


You have now successfully placed the downloaded part on the schematic for PSpice simulation.

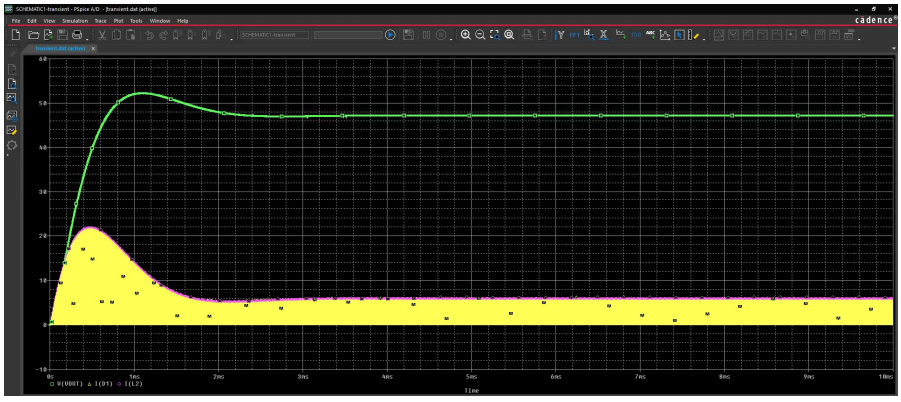
19. Add voltage & current markers on the schematic as shown in figure below



20. Go to **Edit Simulation Profile – Configuration Files – Library**. Browse for the **pwr_mos.lib** library and it to the design



21. Run the simulation. Below are the results of the simulation,



Keep the circuit open as it will be used in the next lab.

Lab Summary

In this lab, you

- Learnt the process to bring in third party vendor models to use in PSpice simulation

End of Lab

Lab 12-2 Understanding when soft landing occurs in TI - PSpice

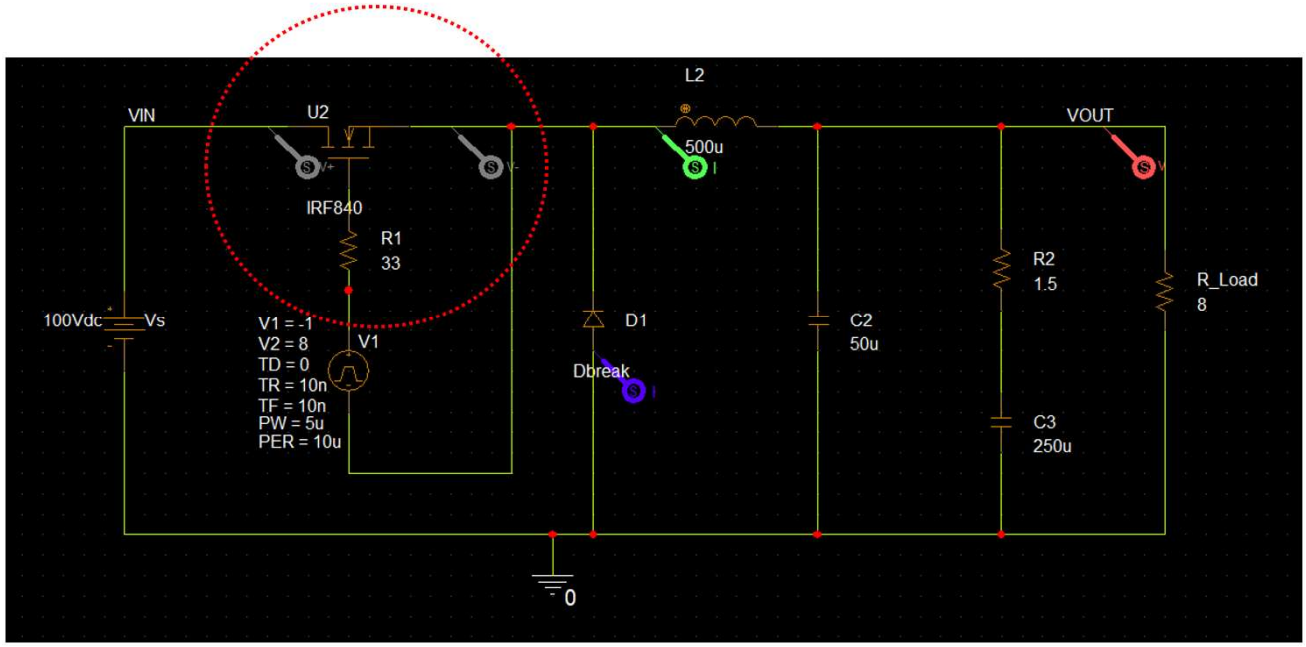
Objective: To run transient analysis on a buck converter used in the last lab, but probe more than 3 signals in the circuit at the same time

---

Note, that in the previous lab we probed **3 signals**, in this lab we will try to probe more than 3 signals at a time in the same circuit and see what happens

Re-simulating model_import.opj

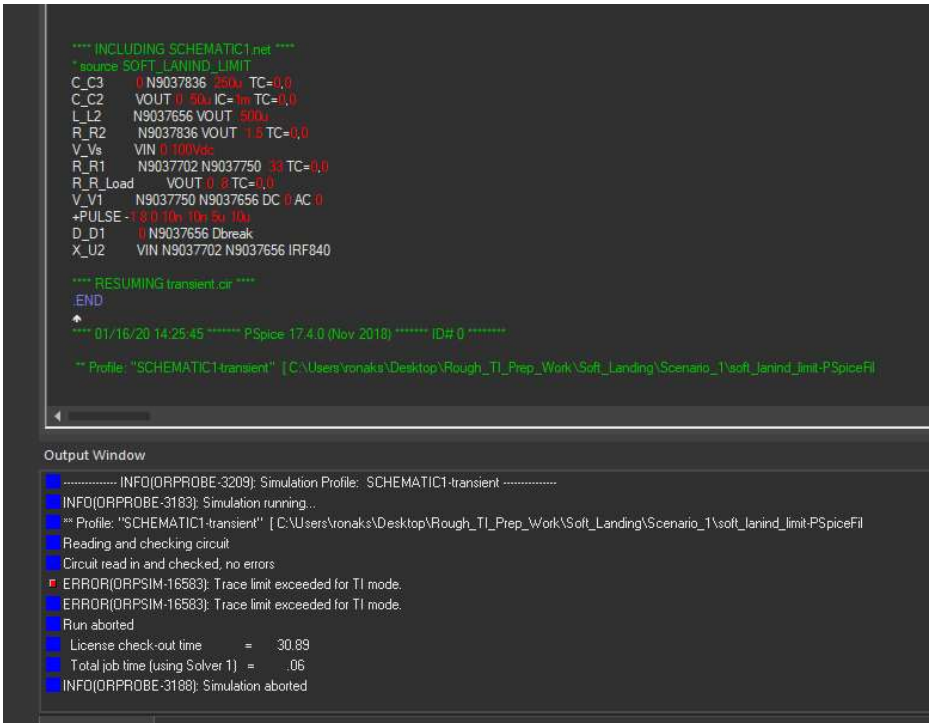
1. Position 4 markers on the schematic as shown below

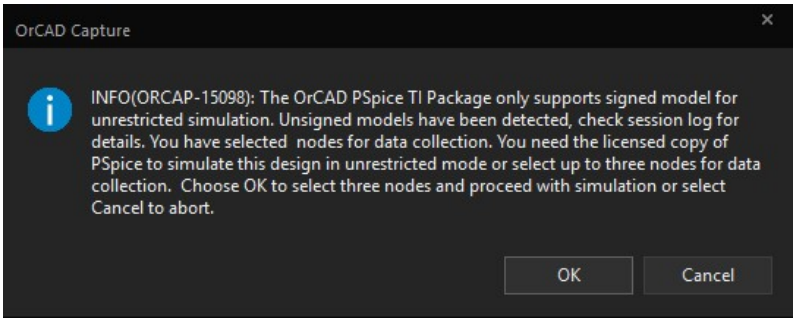


   Note that we added additional voltage differential marker across the MOSFET, which makes it a total of 4 markers on the schematic.

2. Simulate the circuit by clicking on run icon ( )

3. Notice the error that comes up as shown below,

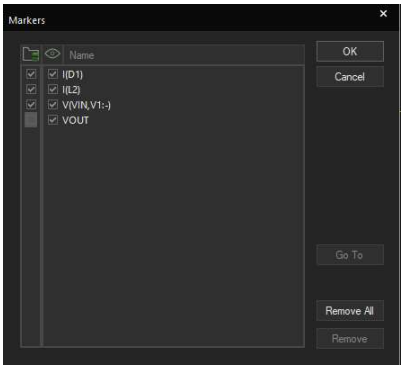




It says that the trace limit has exceeded for TI mode.

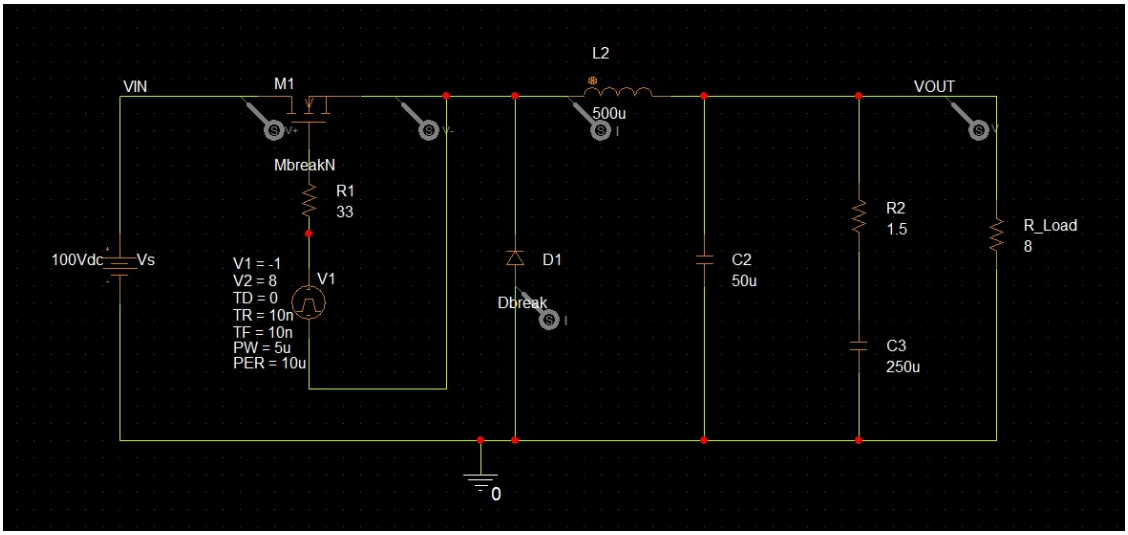
Thus, a TI-PSpice user will be able to plot maximum of 3 signals at a time when any 3rd party model/s is used in the schematic. The software will show the above error if this condition is not met. We term this as 'Soft-Landing'

4. Although, PSpice will give user an option to pick any 3 signals to probe at a time per their convenience.



Additional Exercise

1. Replace the component IRF840 with MbreakN3 component and try plotting the same 4 signals at the same time



You will notice that the simulation goes through successfully this time and no error shows up. This is due to the reason that all the parts used in the circuit are from the inbuilt PSpice libraries that come with installation and none of the models are imported from web.

Lab Summary

In this lab, you

- Understood when soft-landing occurs in TI-PSpice.

End of Lab

Module 13: Creating Hierarchical Blocks & Symbols

Lab 13-1 Easily create Hierarchical Blocks for use in PSpice Simulation

Objective: Learn how to create Hierarchical Schematic blocks for PSpice Simulation

---

[Why are Hierarchical Blocks so useful?

- There are fewer net name conflict issues as the connectivity is derived through ports on a block.
- Constraints can be captured while the logic is being defined. It is easy to replicate a block and easy to make changes at the block level.

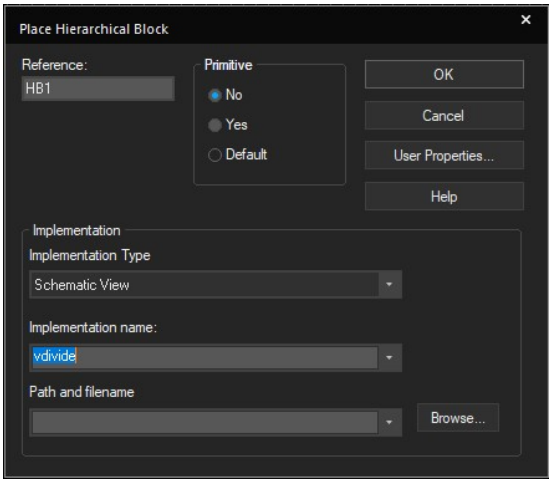
Read a collateral here to understand the differences between using Flat vs Hierarchical approaches for schematic design]

Creating the Top-Level Design (Hierarchical Block)

1. Start a new PSpice A/D project. Name it **Divider.*opj***.

2. On the new schematic page, choose **Place – Hierarchical Block** *or* click the **Place Block** button.

   The Place Hierarchical Block dialog box opens.



3. Enter **HB1** in the **Reference** field.

4. In the **Primitive** section, select **No**.

5. In the **Implementation Type** pull-down box, select **Schematic View**.

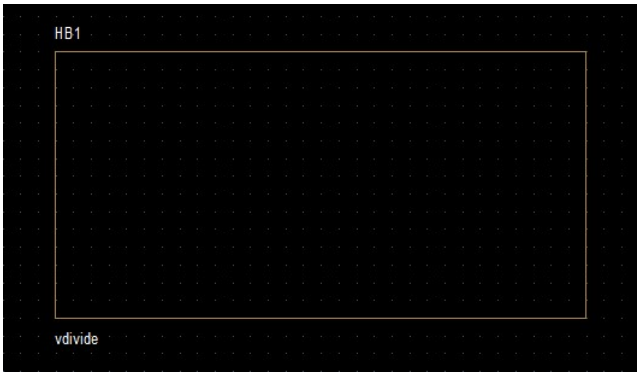
6. In the **Implementation name** field, enter **vdivide**.

7. Leave the **Implementation Path** field blank.

8. Click **OK**.

9. Click and drag the mouse to draw the block on the schematic page.

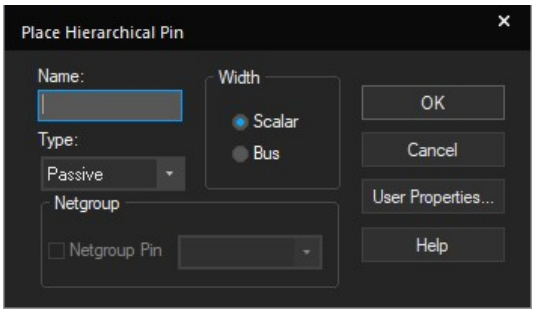
   Note: If necessary, you can resize the block by dragging a corner.



10. If the block is not selected, click it to **select it.**

11. Choose **Place – Hierarchical Pin**

    The Place Hierarchical Pin dialog box opens.

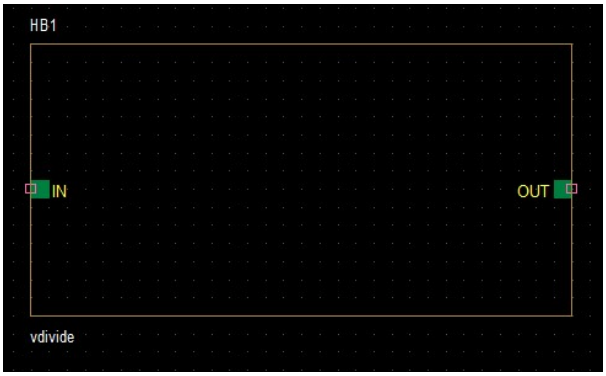


12. Name the pin **IN**. Click **OK**.

    A pin symbol is attached to your pointer.

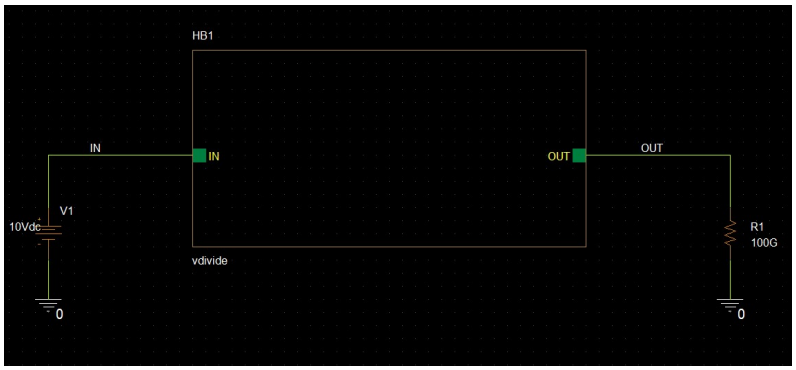
13. Place the pin on the left side of the block.

14. Place another pin on the right side of the block. Name it **OUT**.

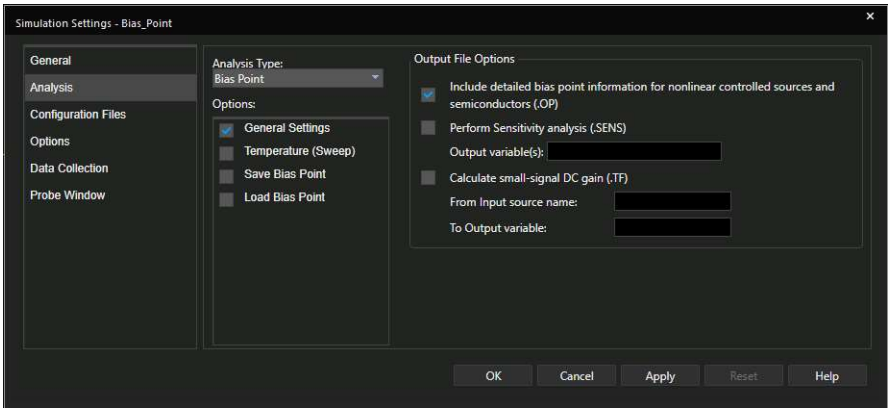


15. Complete the top-level design as shown in the following figure. Use the **VDC** and **R** parts, and ground symbol of your choice.

    Note: Change the ground name to **0** if you do *not* use the 0 (zero) symbol.



16. Configure a bias point analysis in a new simulation profile.

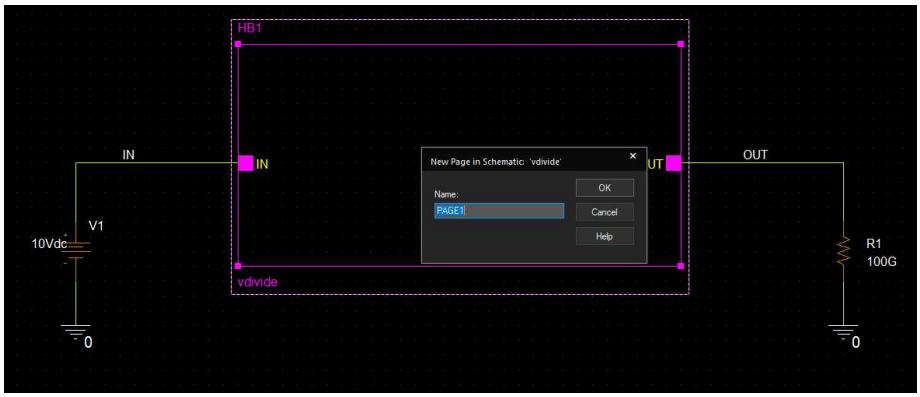


Creating the Lower-Level Design

1. Select the block.

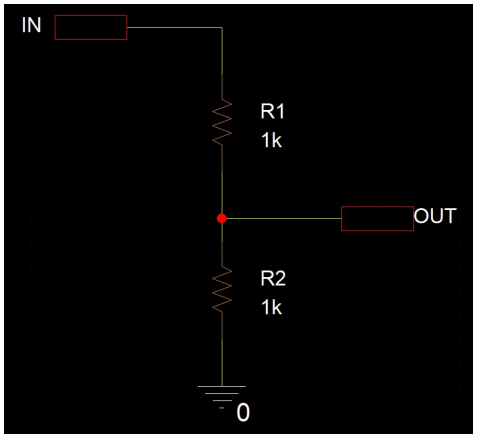
2. **Right**-click and select **Descend Hierarchy**. **Alternate:** Choose **View – Descend Hierarchy**

3. (Optional) Change the new page name.



4. Click **OK**.

5. Use the two ports on the page, two **R** parts, and any ground symbol to create the schematic shown here.



If you use a symbol other than **0** ground, **double**-click the symbol and rename it **0**.

6. Save the file.

7. Choose **View – Ascend Hierarchy** or press **Shift+A** to return to the top-level design.

Simulating and Viewing Results

1. Simulate the design.

2. Go back to the Capture Schematic window and enable voltage and current bias voltage displays 

Lab Summary

In this lab, you

- Created a hierarchical block & successfully performed simulation

End of Lab

Module 14: Analog Behavioral Modelling with PSpice

Lab 14-1 Create Low Pass Filter using Behavioral Modelling

Objective: To create a low-pass filter with an EFREQ part.

[Behavioral Modeling is the process of developing a model for a device or a system component representing the behavior rather than from a microscopic description. You can use Behavioral Modeling in the domain of analog simulation to model new device types and for black-box modeling of complex systems.

LPF using Frequency Look Up Table

This circuit is a simple low-pass filter implemented with a frequency look-up table. The input to the frequency look-up table is the voltage at node IN. The output is the output on node OUT.

The table values describe a low-pass filter with a response of 1 (0 dB) for frequencies below 5 kHz and a response of .001 (-60 dB) above 6 kHz. The phase lags linearly with frequency. The delay (phase wrap) is necessary so that the impulse response is causal, that is, so the impulse response has no significant real components before time zero.

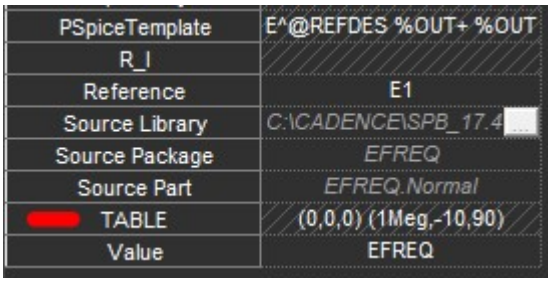
Steps

1. Create a new project named **Lowpass**.
2. Place and connect the symbols needed in the circuit: **VAC**, **0**, **EFREQ**, and **R**.
3. Set the voltage for V1 to **1V**.

4. **Double**-click the **EFREQ** symbol and select the **TABLE** property.

5. Edit the value for the **TABLE** property as the following:

```
(0,0,0) (5KHz,0,-5760) (6KHz, -60,-6912)
```

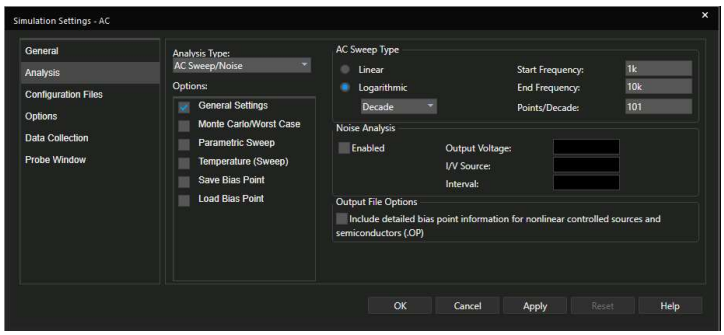


6. Click the **Display** button.

7. Click the **Value Only** radio button.

8. Click **OK** to close the Display Properties form.

9. Set up a logarithmic AC simulation with **101** points, starting at **1k** and ending at **10k**.

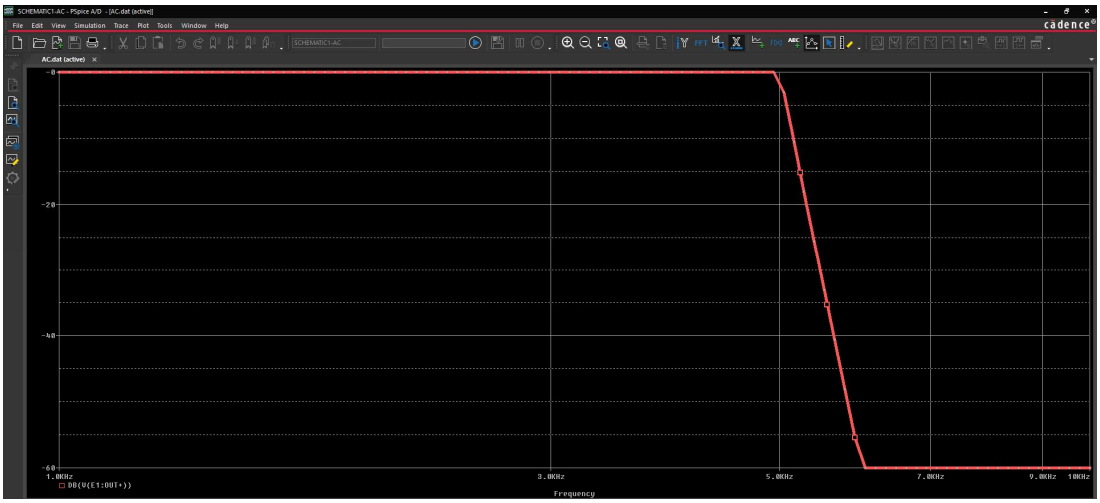


10. Click **OK**.

11. Choose **PSpice – Markers – Advanced** – **dB Magnitude of Voltage** from the text menu.

12. Place the marker on node **OUT**.

13. Run the simulation. Examine the results in the Probe window.



You can see that the response is as expected. Below 5k the response is 0 dB and above 6k the response is -60 db.

Lab Summary

In this lab, you

- Created a low-pass filter using an EFREQ part

End of Lab

Module 15: Simulating Digital & Mixed Signal Circuits with PSpice

Lab 15-1 Digital Design Simulation with PSpice

Objective: To create a digital design, configure and run a digital simulation, and describe the results in Probe

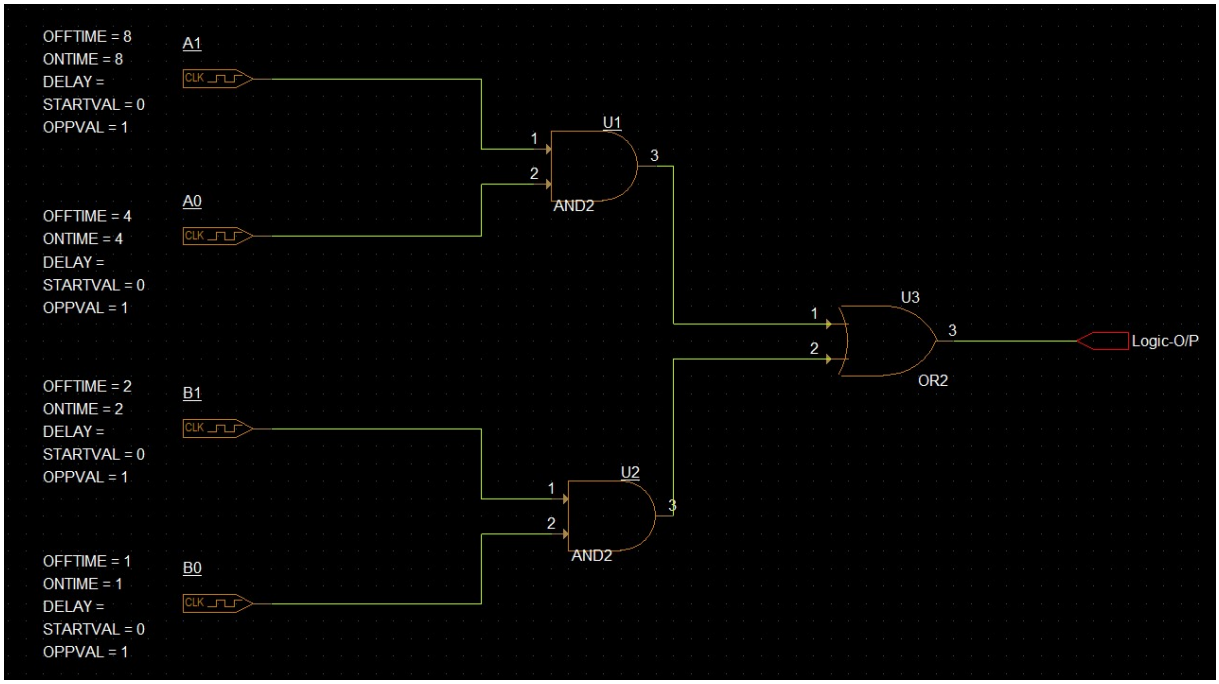
---

In this lab, you create a digital design by

- Configuring and running a digital simulation
- Examining the results in Probe

Creating a Digital Circuit

1. Create a new project called **Digital_Simulation** as shown below. This circuit represents the logic formula Y=A1A0 + B1B0.



2. Create this circuit with the following parts: **AND2**, **OR2**, and **DigCLOCK**
3. Select AND2, OR2 parts from **Place – PSpice Component – Digital – Gates.**

4. To select Digital clock source DigCLOCK for our circuit, click on the icon to place part ( )

   Select DigCLOCK part from the source library. (Add all the inbuilt libraries present with installation by clicking on the icon for adding libraries ( ) if needed

5. Edit the values of digital clock stimulus as shown in the circuit diagram. Match the reference designators of the components per the figure too.

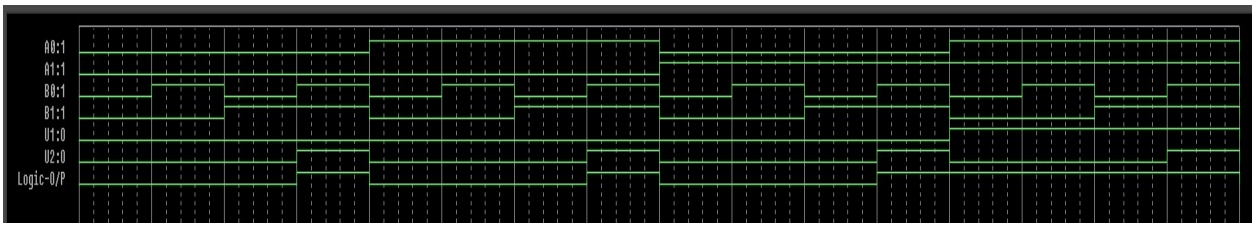
Creating the Simulation Profile

1. Define a new simulation profile with a transient analysis set to **run to time of 16 seconds**.
2. Click **Apply**. Click **OK**. Save the file.
3. Run the simulation.

Viewing the Results in the Probe Window

1. Click the **Add Trace** button.
2. Click each of these signals: **A0:1, A1:1, B0:1, B1:1, U1:O, U2:O, Logic-O/P**

   The output should look as shown below,



   Verify the logic operation of our circuit: Y=A1A0 + B1B0.

Lab Summary

In this lab, you performed the following tasks:

- Created a digital design
- Configured and ran a digital simulation
- Examined the results in Probe

End of Lab

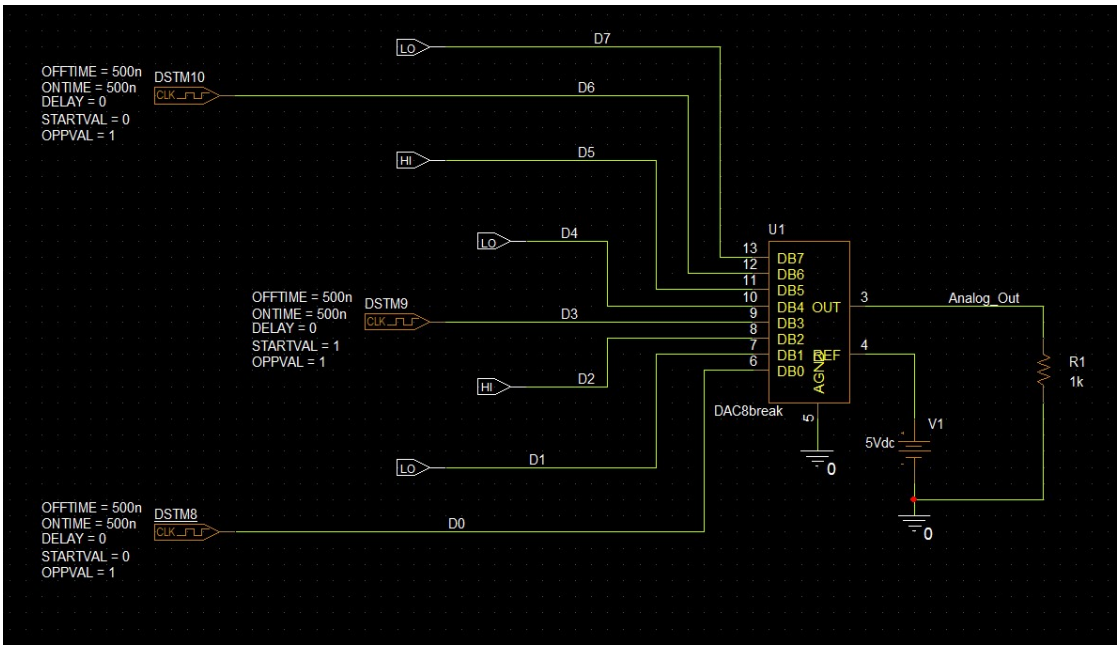
Lab 15-2 Mixed Signal Circuit Design with PSpice

Objective: To open an existing mixed signal digital to analog converter circuit and simulate it to view the results

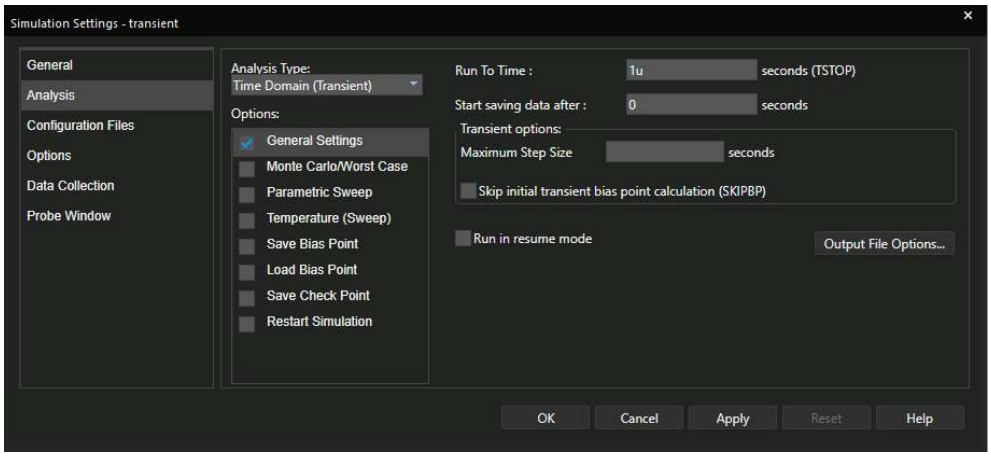
---

Opening the mixed signal circuit

1. Open **mixed_signal_3.opj** project. This circuit represents a Digital-to-Analog converter.



2. Examine the simulation profile where a transient analysis is set to run to 1us,

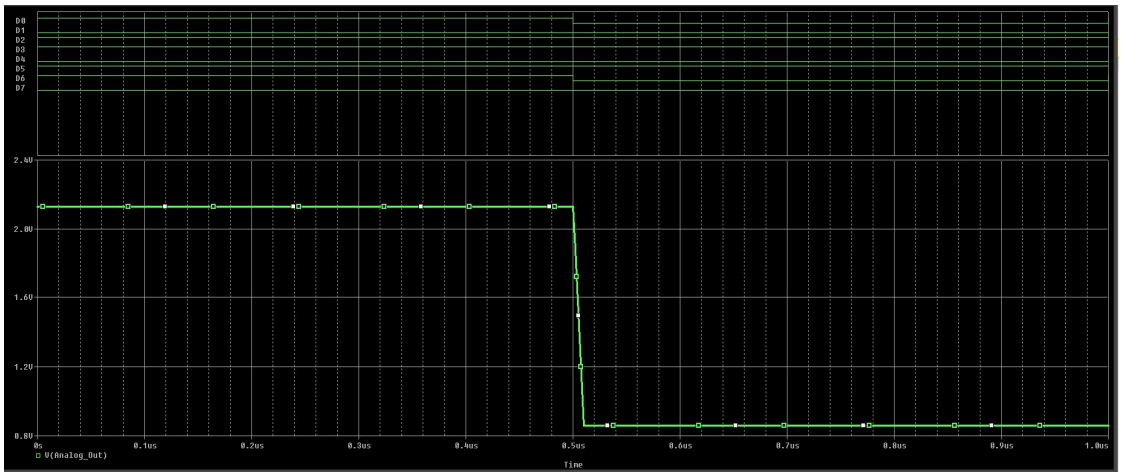


3. Run the analysis.

Viewing the Results in the Probe Window

1. Click the **Add Trace** button.
2. Click each of these signals: **D0, D1, D2, D3, D4, D5, D6, D7, V(Analog_Out)**

   The output should look as shown below,



Examine the output in the Probe window. Notice that the analog and digital signals share a common x axis for time. This allows you to examine the effects of both the analog and digital portions of the circuit concurrently.

Lab Summary

In this lab, you performed the following tasks:

- Opened an existing mixed A/D circuit
- Simulated the mixed A/D circuit
- Viewed the results in Probe

End of Lab

Module 16: Noise Analysis with PSpice

[What is noise analysis?

When running a noise analysis, PSpice calculates and reports the following for each frequency specified for the AC Sweep/Noise analysis:

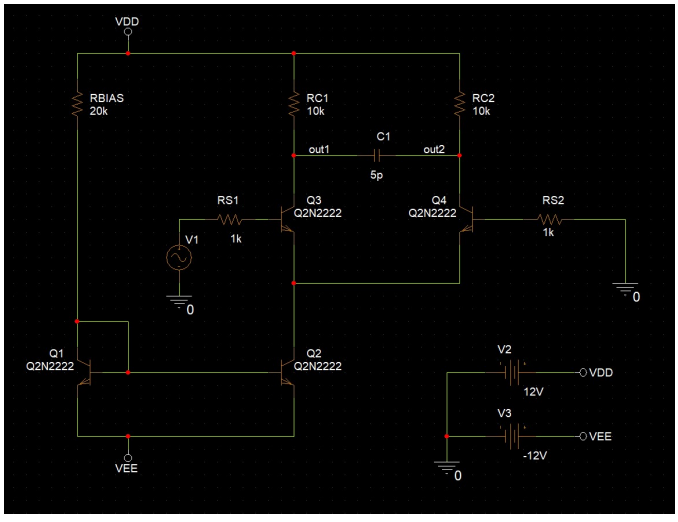
1. Device noise, which is the noise contribution propagated to the specified output net from every resistor and semiconductor device in the circuit; for semiconductor devices, the device noise is also broken down into constituent noise contributions where applicable
   Example: Diodes have separate noise contributions from thermal, shot, and flicker noise.

2. Total output – RMS sum of all the device contributions propagated to a specified output net
   Input noise - equivalent noise that would be needed at the input source to generate the calculated output noise in an ideal (noiseless) circuit]

Lab 16-1 Running Noise Analysis

Objective: To configure and run Noise Analysis on a circuit.

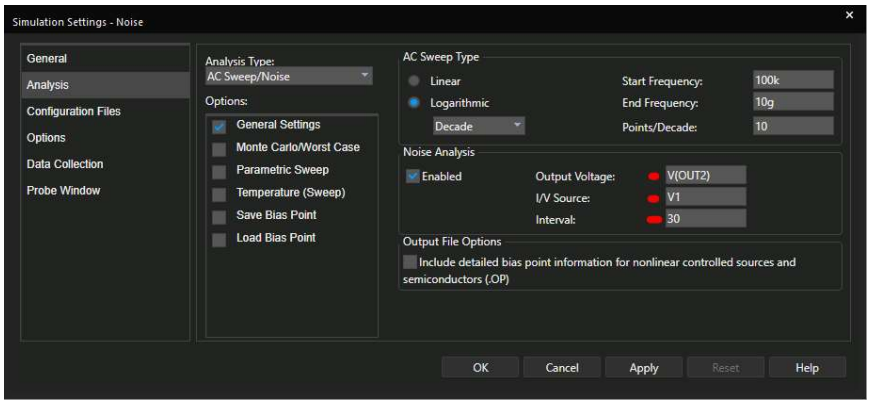
Configuring and Running Noise Analysis



1. Open **Example.opj**.

2. Create a new simulation profile. Name it **Noise**.

3. Enable **Noise Analysis.** Configure the settings as shown in the dialog box below,



[Noise Analysis parameters:

Output Voltage: A voltage output variable of the form V(node, [node]) where you want the total output noise calculated.

I/V Source: The name of an independent current or voltage source where you want the equivalent input noise calculated.

Interval: An integer n designating that at every nth frequency, you want to see a table printed in the PSpice output file (.OUT) showing the individual contributions of all of the circuit's noise generators to the total noise]

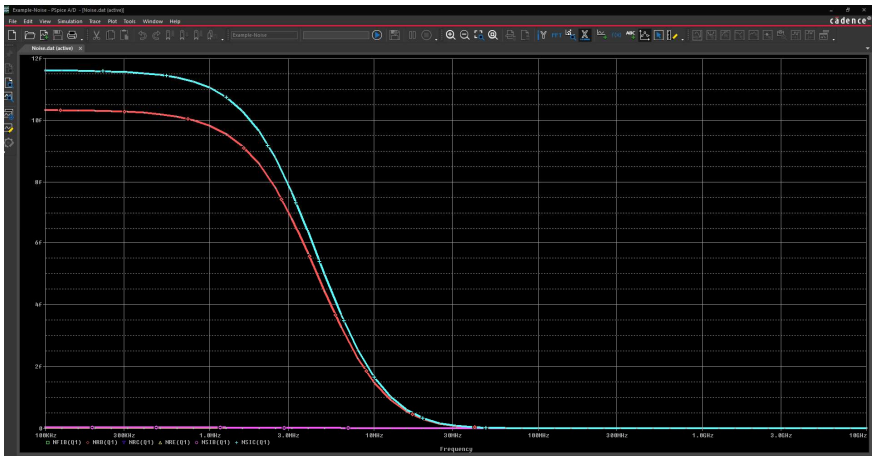
4. Click **OK** to save the profile. Run the simulation.

Examining the Results in Probe

The following output variable formats are used in PSpice to view traces for device noise contributions and total input or output noise at every frequency in the analysis

To view this...	Use this output variable...
Flicker noise for a device	NFID(*device_name*) NFIB(*device_name*)
Shot noise for a device	NSID(*device_name*) NSIB(*device_name*) NSIC(*device_name*)
Thermal noise for the RB, RC, RD, RE, RG, or RS constituent of a device, respectively	NRB(*device_name*) NRC(*device_name*) NRD(*device_name*) NRE(*device_name*) NRG(*device_name*) NRS(*device_name*)
Thermal noise generated by equivalent resistances in the output of a digital device	NRLO(*device_name*) NRHI(*device_name*)
Total noise for a device	NTOT(*device_name*)
Total output noise for the circuit	NTOT(ONOISE)
RMS-summed output noise for the circuit	V(ONOISE)
Equivalent input noise for the circuit	V(INOISE)

1. In the Probe window, add the traces **NFIB(Q1), NRB(Q1), NRC(Q1), NRE(Q1), NSIB(Q1),** and **NSIC(Q1)**. In the output file, this shows the individual noise contributions from the Q1 device.



2. Choose **View – Output File** to display the tables generated in the output file for each noise source.

Lab Summary

In this lab, you performed the following tasks:

- Ran a Noise Analysis

End of Lab