EMA Design Automation®

PSpice Essentials
Version 17.4

EMA Education Services
Classroom, on-site, and eLearning

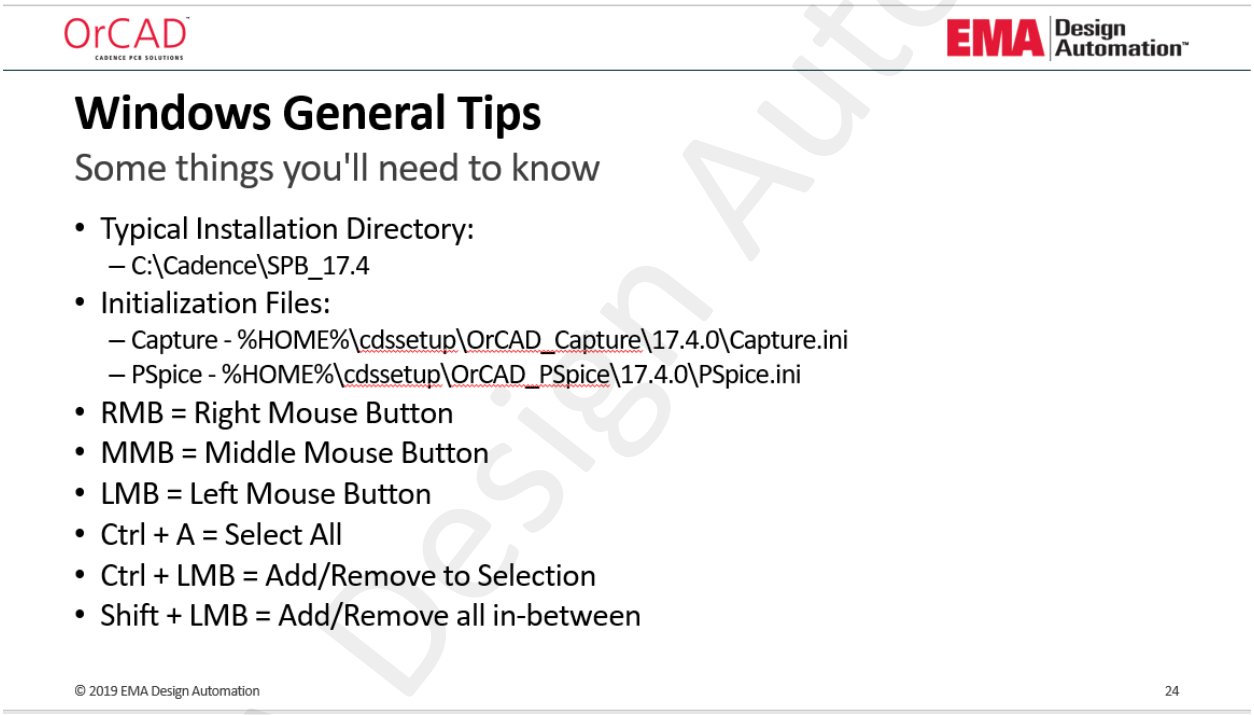
Table of Contents

Lesson 1. Building a Design for Simulation

Lesson Objectives

- PSpice documentation and Learning Tutorial

Windows General Tips



Cadence Tools Overview

OrCAD Capture is tightly integrated with PSpice for simulation and OrCAD and Allegro PCB Editor for board design.

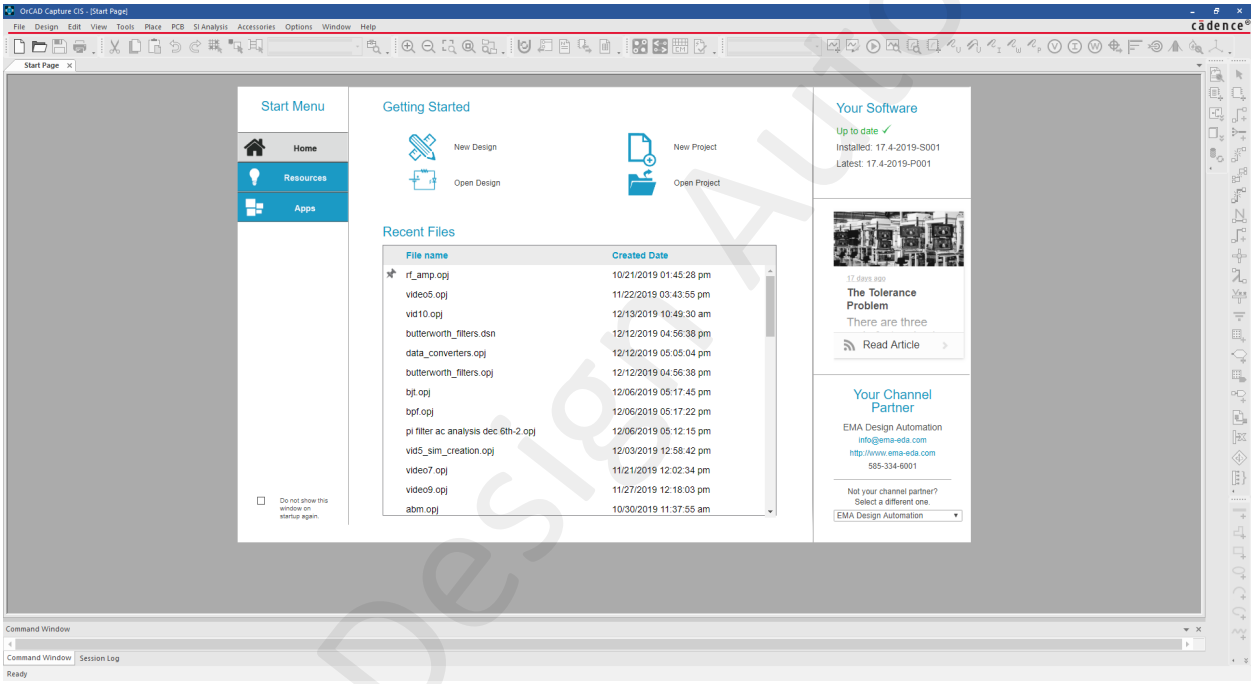
OrCAD Capture can also use the Cadence OrCAD Component Information System (Cadence OrCAD Capture CIS) to integrate your board-level design with existing in-house part procurement and manufacturing databases.

Starting OrCAD Capture

To Start OrCAD Capture application, select ***Start > All Programs > Cadence Release 17.4-2019 > Capture CIS 17.4***

The OrCAD Capture "Start Page"

When you start the OrCAD Capture software, the main session window appears as shown in the following graphic.



You can start a New Project (or Design), Open an existing Project (or Design), Open recent files, Open Learning Resources and OrCAD Apps from Start Page.

In the upper, right corner, the current OrCAD version you are running will be displayed. Below that you will see technology articles by cadence.

OrCAD Help

There are several ways to get help with PSpice and Capture tools.

- From Help menu inside PSpice or Capture tools

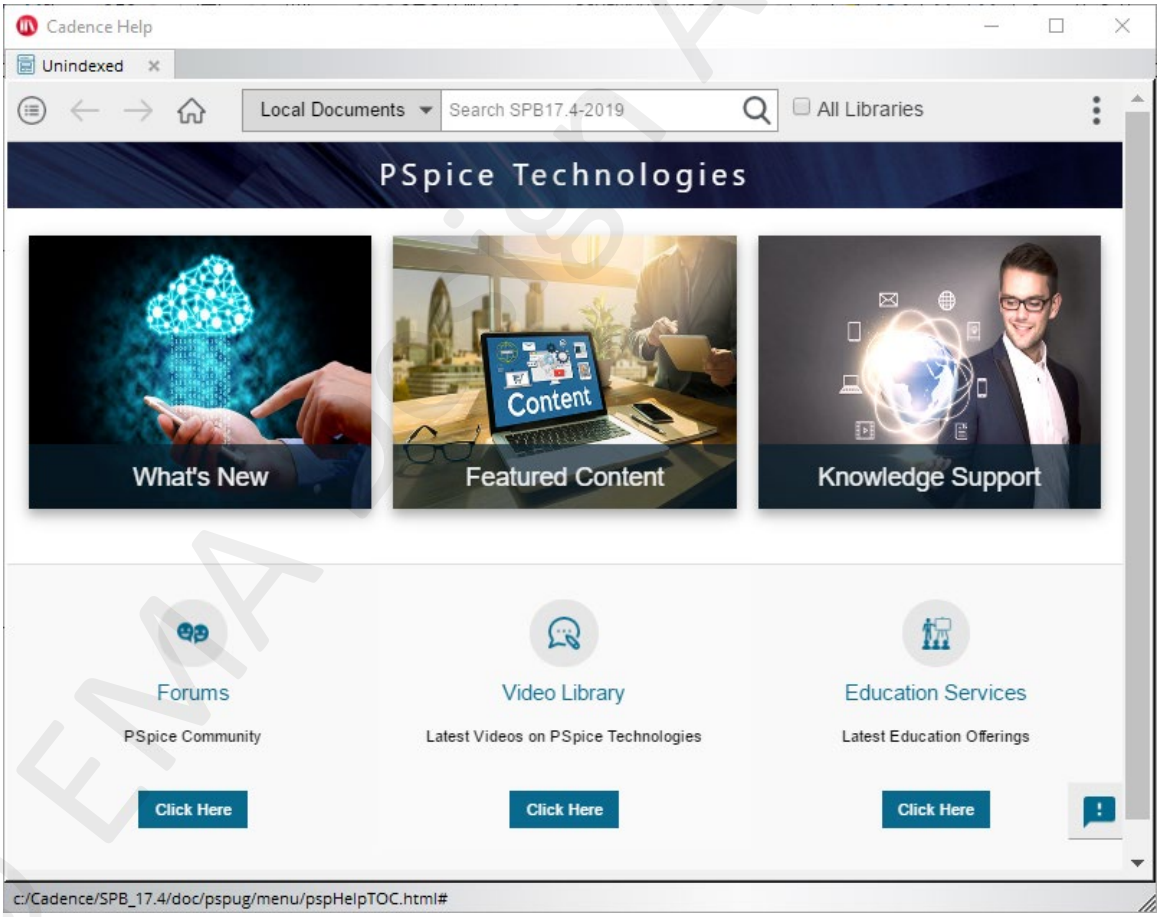
- ***Start > Cadence Help 17.4-2019 > Cadence Help 17.4*** Context sensitive help is available by clicking 'Help' button that can be found in different windows
- http://support.ema-eda.com/ EMA Resource Center is the online support resource for OrCAD tools

Each Cadence tool has HTML based manuals that you can access by choosing ***Help > Documentation*** from respective tool.

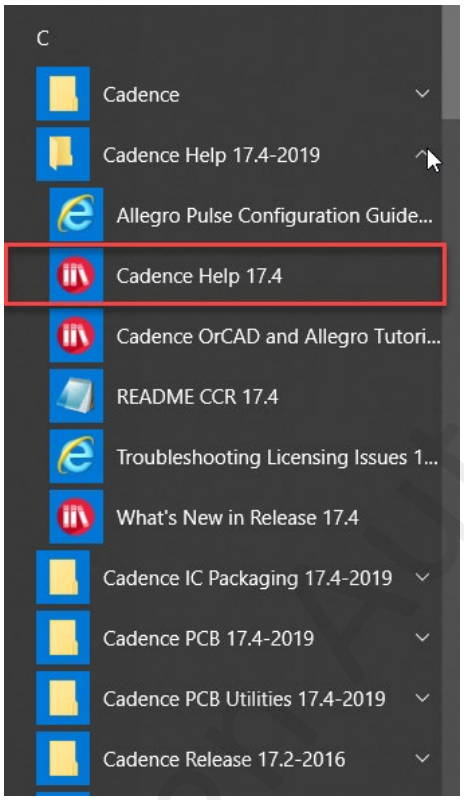
You can access PSpice User's Guide and PSpice Reference Guide from ***Help > Documentation*** menu in PSpice.

PSpice User's Guide is a comprehensive guide for understanding and using the features available in PSpice that designers can use to simulate the design.

PSpice Reference Guide describes PSpice simulation controls, analysis specifications, PSpice devices, model parameters and device equations.

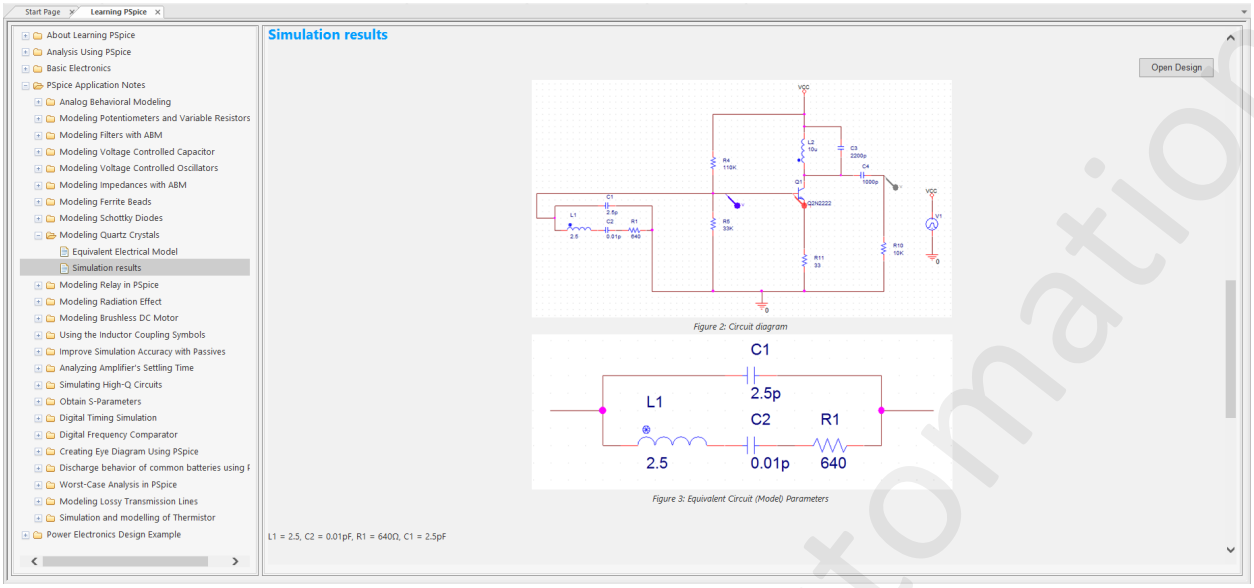


Cadence Help system can be accessed by choosing Start > Cadence Help 17.4-2019 > Cadence Help 17.4.



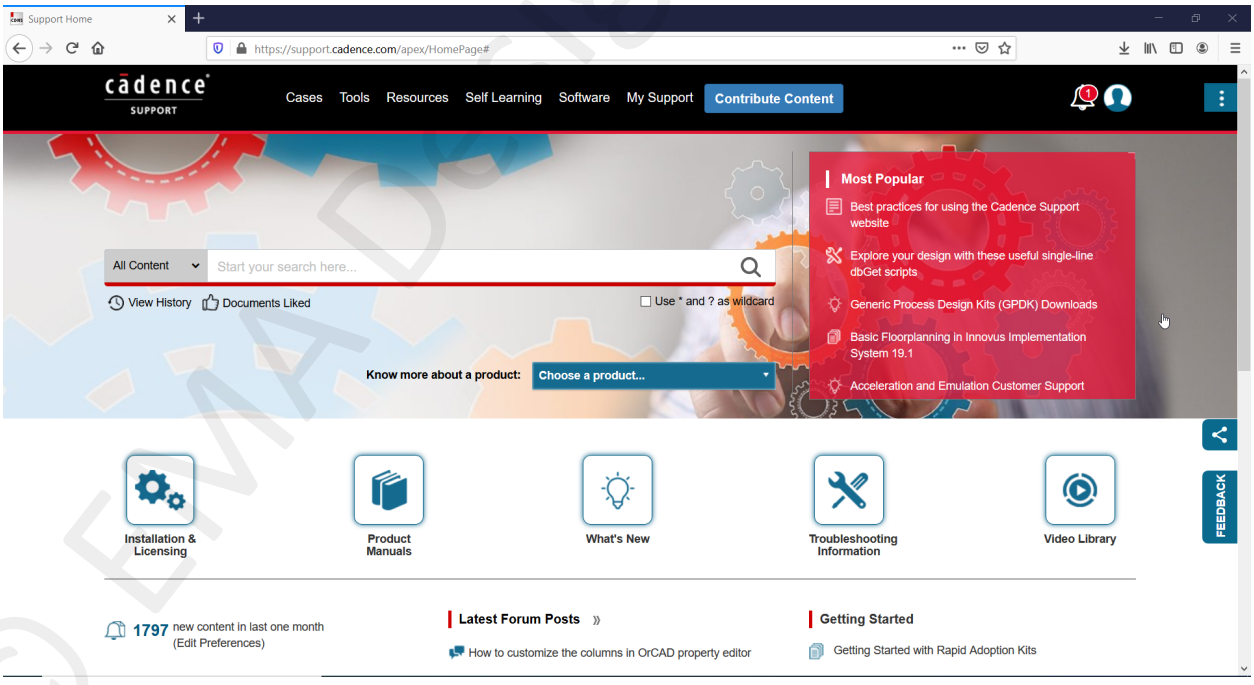
With the Cadence Help, users can access, view, and search the help documentation for all Cadence products installed on the machine. The Documentation Browser panel displays the documentation categorized based on product type.

There is also a PSpice tutorial built into OrCAD Capture. PSpice tutorial covers topics ranging from basic theorems to very advanced topics. This tutorial also includes a wide variety of PSpice Application notes and ready-to-simulate circuits. To access PSpice tutorial choose ***Help > Learning PSpice*** in Capture. User can click on *Open Design* button to access the associated design and then run the simulation.



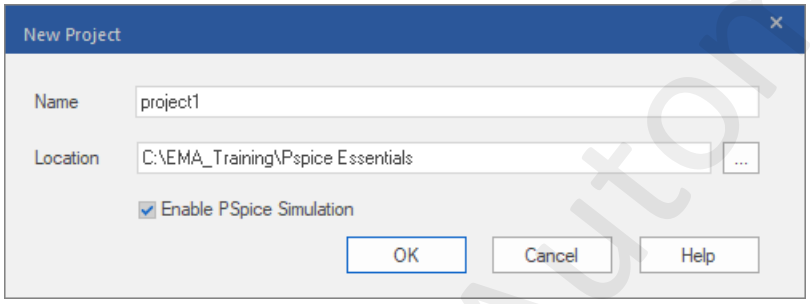
The EMA Resource Center (ERC) is your online support resource for OrCAD products and can be found at http://support.ema-eda.com/. This password protected site contains application notes, product downloads and software service packs.

User can also use Cadence® Online Support site at http://support.cadence.com/ for product related solutions and application notes.



Creating a New Project

Each design created has a *project* associated with it. A project is a collection of files that make up a circuit description (in one or more configurations), its associated simulation characteristics, and any other related derivative information that can be linked with it (unique part libraries, model libraries, etc.). All projects are characterized by a 'project file' appearing at the root of the project location as a *<project_name>.opj* file and contains a collection of pointers to all other files associated with your design.

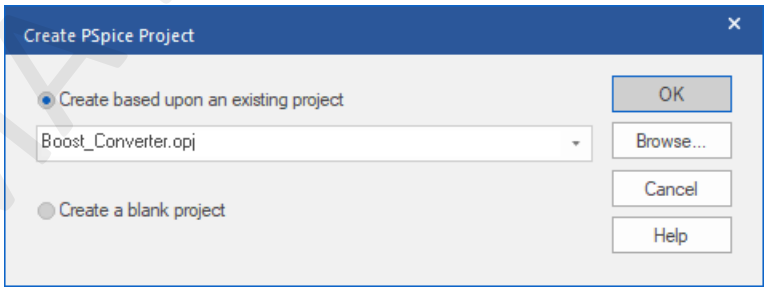


In order to simulate a design in a project, the check box "enable Pspice Simulation" MUST be checked. This will ensure the PSpice A/D menu and toolbars are activated in OrCAD Capture. If the design is not PSpice enabled, it will not simulate.

Since parts used in PSpice A/D project types can have footprint information, PSpice A/D designs may be used in all Cadence OrCAD and Allegro® PCB Editor tools, or other layout programs.

Project Templates

Project templates allow you to create a new project based on a predefined project, or any existing project that you may have. You can also elect to create a blank project.



There are 42 predefined projects.

Empty projects contain a single schematic folder with a single schematic page. Hierarchical projects contain two schematic folders with a single page each. The *empty_aa*, *hierarchical_aa*,

and *simple _aa* projects are setup for Advanced Analysis such as Monte Carlo, and/or Smoke (stress analysis). The rest of the project templates cover basic electronic circuits and SMPS topologies. These project templates cover the range of analog, digital, and mixed signal designs.

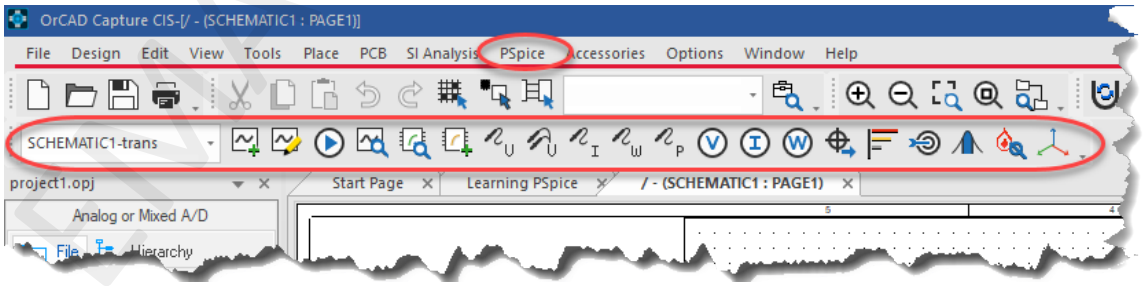
Most templates are pre-configured with a minimum set of part libraries. Templates listing *all_libs* in their names have all of the PSpice A/D-ready part libraries configured. You may add and remove part libraries from any project.

The option to base a new project on an existing project functions as a ***File > Save Project As…*** menu command. It pulls in all aspects of the existing design including simulation profiles, model libraries, and stimulus files. In essence, this allows you to quickly make a copy of an existing project. This is useful if you have a design at a working stage, but would like to try some "what-if" experiments without losing the original working design.

Note: Most (if not all) of the templates configure a number of part libraries for use within the project. One should keep in mind that it's not necessary to retain all "loaded" libraries for a project. In fact, you don't need to "load" any library for a PSpice project. Loading a library pertains to having the library loaded up and configured in the Capture Project page (you'll see a list of libraries under the Library folder). While loading libraries may be convenient, they can cause a very large "system memory footprint" for the project if the number of loaded libraries is significant. This can sometimes cause havoc in machine performance. It's best to not load a library unless you plan on making edits to its contents. You can still access any part and in the system and place it without loading the respective library.

The PSpice Project

You can verify that you are working in a PSpice A/D project by looking for the PSpice menu and the PSpice A/D toolbar. The PSpice A/D toolbar is usually located below the main OrCAD Capture toolbar as shown in the below graphic.



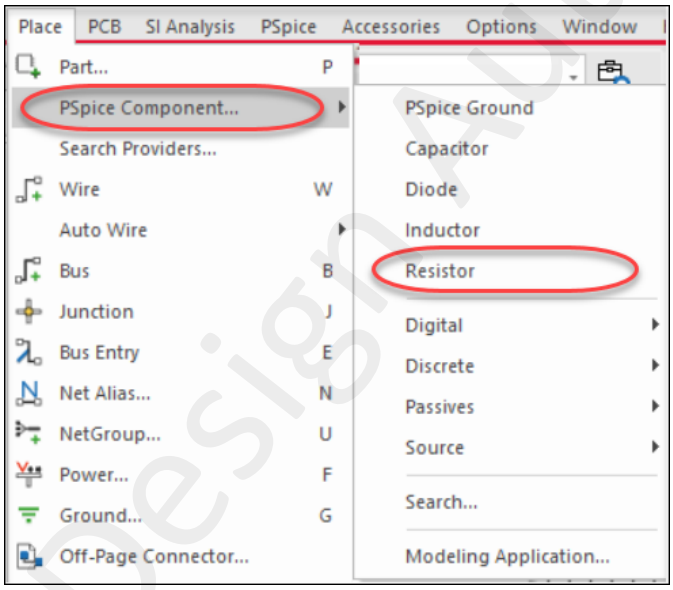
Note: Not every icon in the PSpice A/D toolbar will be active when first starting a blank PSpice project, as most of the icon functions are tied to a Simulation Profile (more on this later).

If you have already entered a lot of information into the design before realizing that you are not in a PSpice A/D project you can still save your work and convert it into a PSpice A/D project. To do this simply create a simulation. Before the simulation window opens, capture will prompt you to select a license that includes the pspice option.

Placing Parts

As of OrCAD Capture release 16.6, placing parts for PSpice designs is much simpler. For this and all following labs we'll abide by the new QuickPlace method for placing PSpice parts into our designs unless otherwise prohibited. If there is interest in the legacy part placement flow (using the Part Browser), see the Appendix for a discussion of that method.

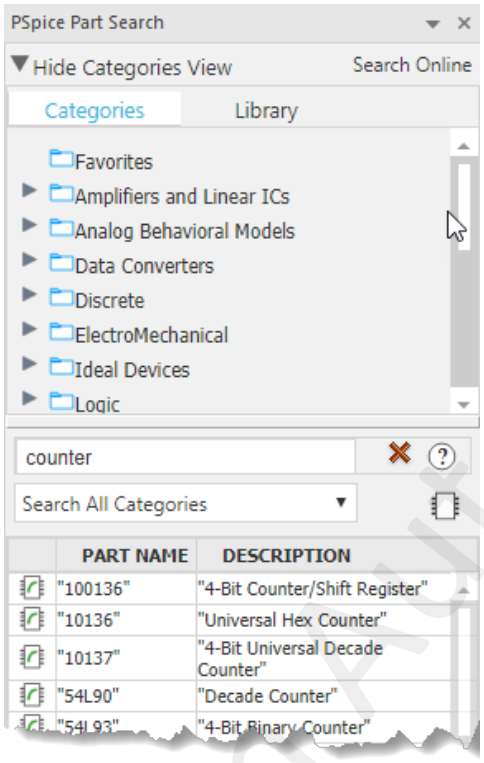
PSpice part selection is accomplished via the *Place > PSpice Component…* menu command.



Notice that the most common PSpice components are listed at the top of the sub-menu selection list, followed by a set of categorized component sub-menus, a more powerful search function, and TCL based Model Applications for modeling common 'non-ideal' parts (among others) very quickly. No longer is there a need to configure part libraries when using the QuickPlace method as this flow automatically draws parts from the correct simulation libraries.

As of OrCAD Capture release 17.4 the Pspice search and Pspice Modeling Application are available via icons in the pspice simulation toolbar

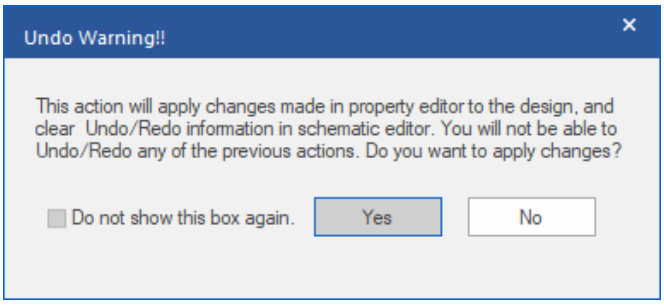
To locate a part by function or part specific name, simply use the *Search* feature for QuickPlace.



You can search by category (as shown for a search for all counters), by library, by part name, partial name with wild cards, among others. Searches can also be restricted to specific categories.

Undo Warning

Some edits that you perform will require the undo cache to be cleared, which will prompt you with a warning panel. To avoid having to constantly click the *Yes* button in this *Undo Warning!!* panel, select the *Do not show this box again* check box before selecting *Yes*. This will clear the Undo cache without asking you when running certain commands.



Lesson 1, Lab 1: Browsing PSpice Tutorials

Some Important Information

The password to your training machines is **train**

The paths to the software and lab files on the training machines are located here:

<Software_Installation>

- C:\Apps\SPB_17.4\

<Labs_Directory>

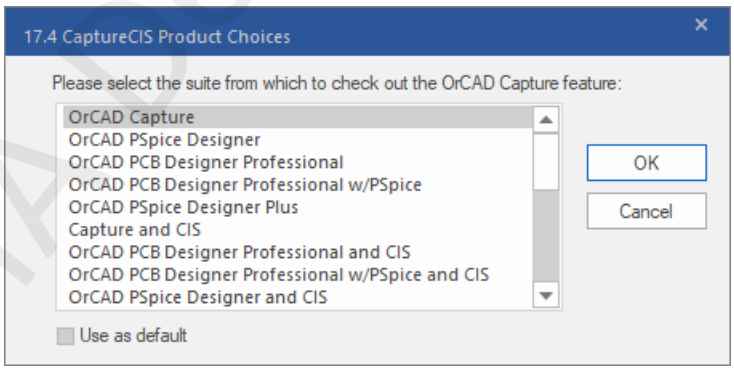
- C:\EMA_Training\PSpiceEssentials\

Opening OrCAD Capture

1. To start OrCAD Capture and begin your first PSpice project, select ***Start > Programs > Cadence PCB 17.4-2019 > Capture CIS 17.4***

OrCAD Capture will respond by prompting for an appropriate license. Depending on what your machine is licensed for, there may be several PSpice license options to choose from.

2. Pick either OrCAD PSpice Designer, OrCAD PCB Designer Professional w/PSpice, or OrCAD PSpice Designer Plus (basically, any license with “PSpice” in it)



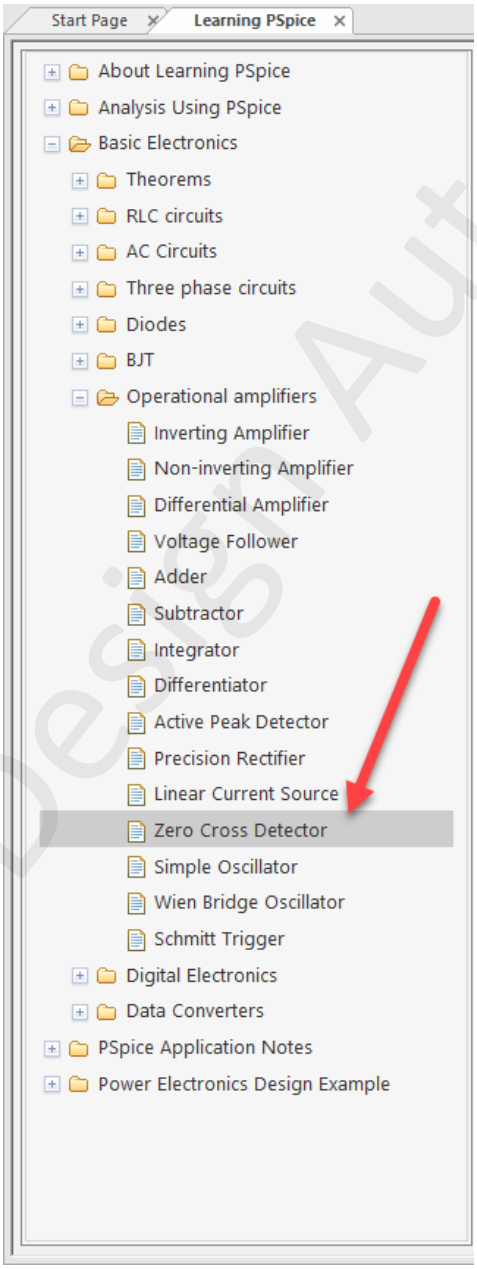
At this point, OrCAD Capture will open presenting (in most cases) the Start Page. This is really a control center of sorts for your Capture projects.

Opening PSpice Tutorial

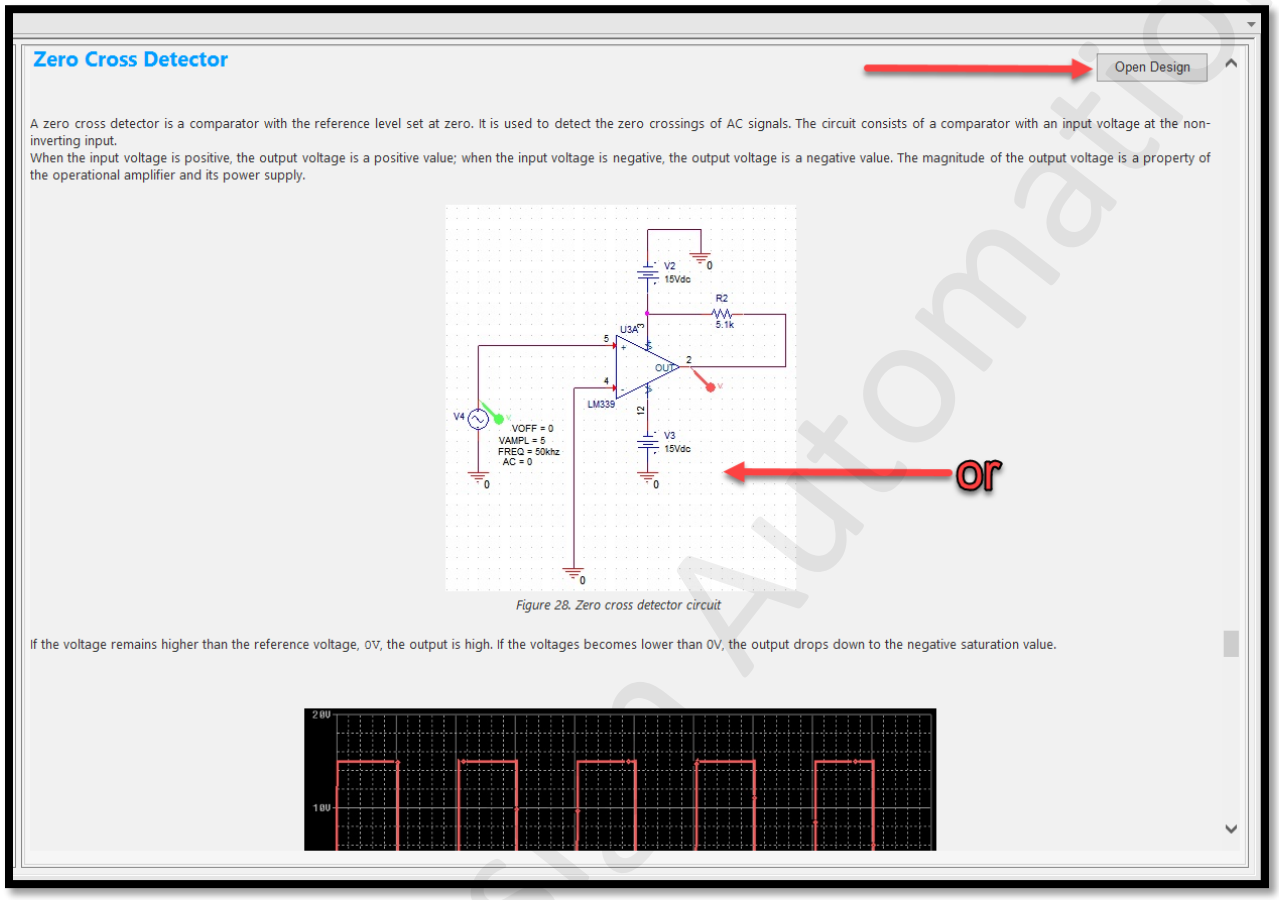
1. Choose ***Help > Learning PSpice***

This will open Learning PSpice tab inside Capture.

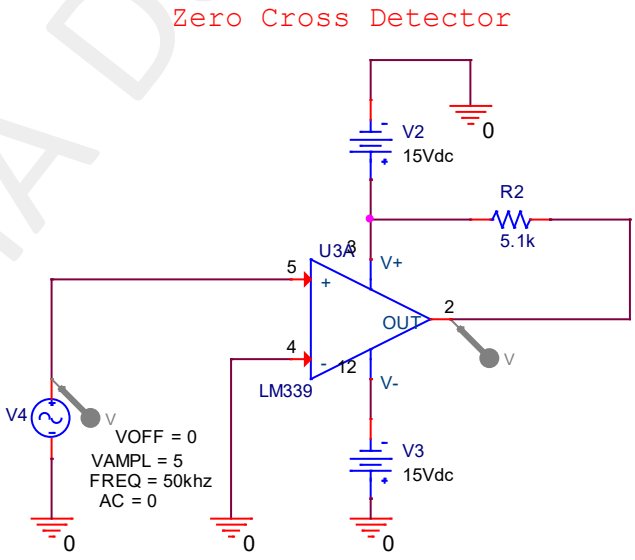
2. In the Topics panel (Left hand side) browse to ***Basic Electronics > Operation Amplifier > Zero Cross Detector***



3. Click the **Open Design** button in the top-right corner. Alternately you can click on the circuit image to open the design in Capture



This will open the design in Capture.

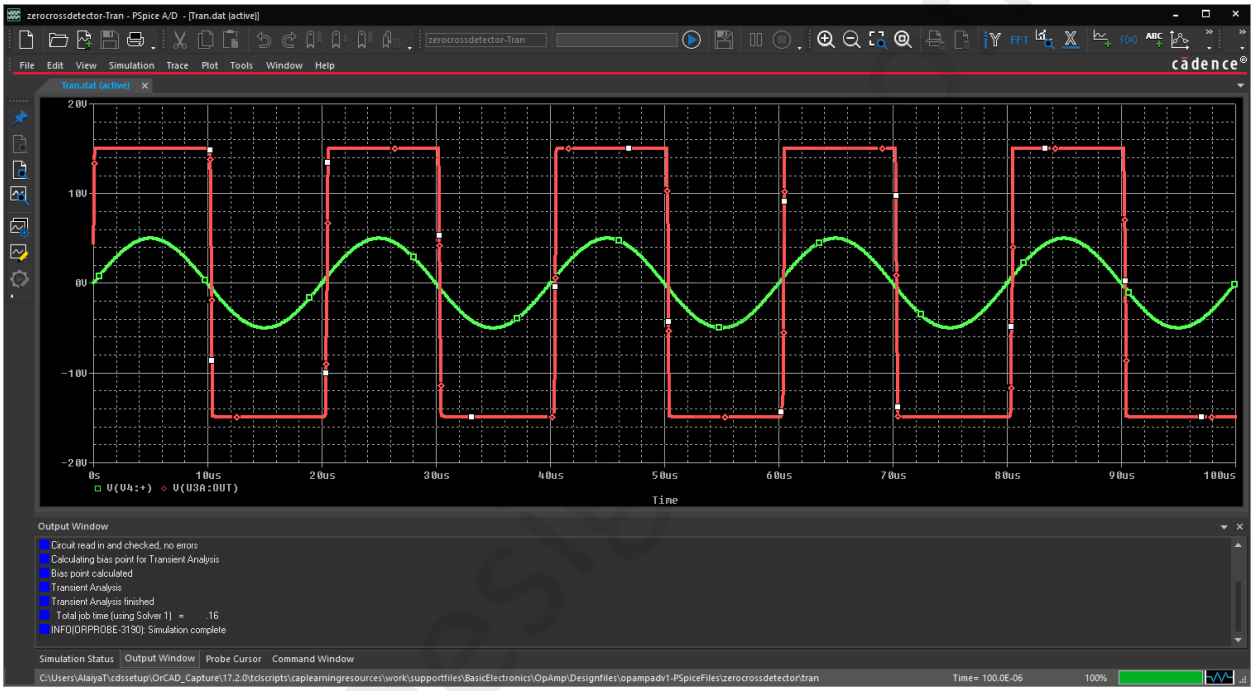


Running PSpice Simulation on Zero Cross Detector Design

1. Choose ***PSpice > Run*** to run the simulation on the design. You can also press F11.

The netlister and DRC automatically run and the Probe window opens with waveforms displayed. This is a zero cross detector and if the input voltage is higher that 0V, the output is high.

If the traces are not being displayed in PSpice, go back to Capture and double click on the greyed-out voltage probes to 'activate' them.



2. Close PSpice waveform window
3. In Capture, close Schematic page by choosing ***File > Close*** and ***No*** to discard all changes.
4. Close Project by choosing ***File > Close*** and ***No*** to discard all changes
5. Close 'Learning PSpice' tab by choosing ***File > Close***

Lesson 1, Lab 2: Searching Help

Lab Objective

- Understand how to get into OrCAD Help and how to search through the documentation

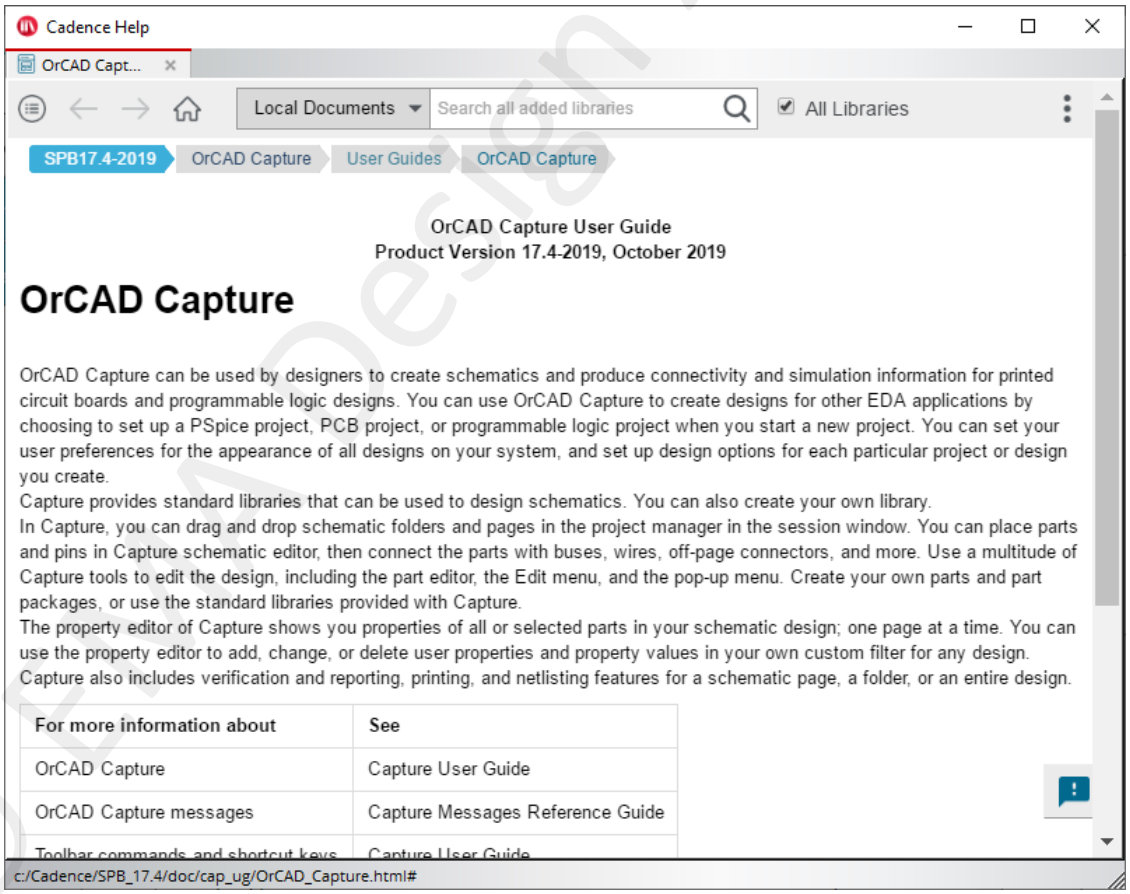
Opening Help

1. In Capture, go to ***Help > OrCAD Capture Help*** or press ***F1*** or the Help ( ) icon

This brings up Cadence Help which has inside it help for all the tools that Cadence offers. When brought up from inside Capture like we just did, the OrCAD Capture general help is displayed.

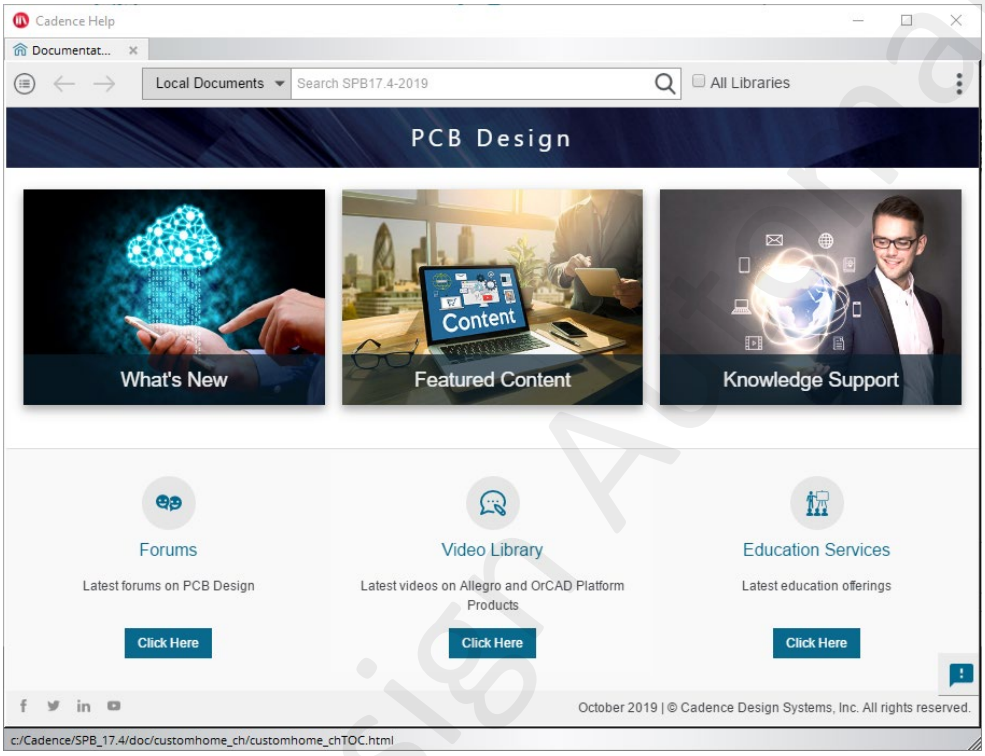
On this page itself, there's not really a lot of useful information or links to more information. There are only two things that you can do from here.

2. Select the ***Home*** icon ( )



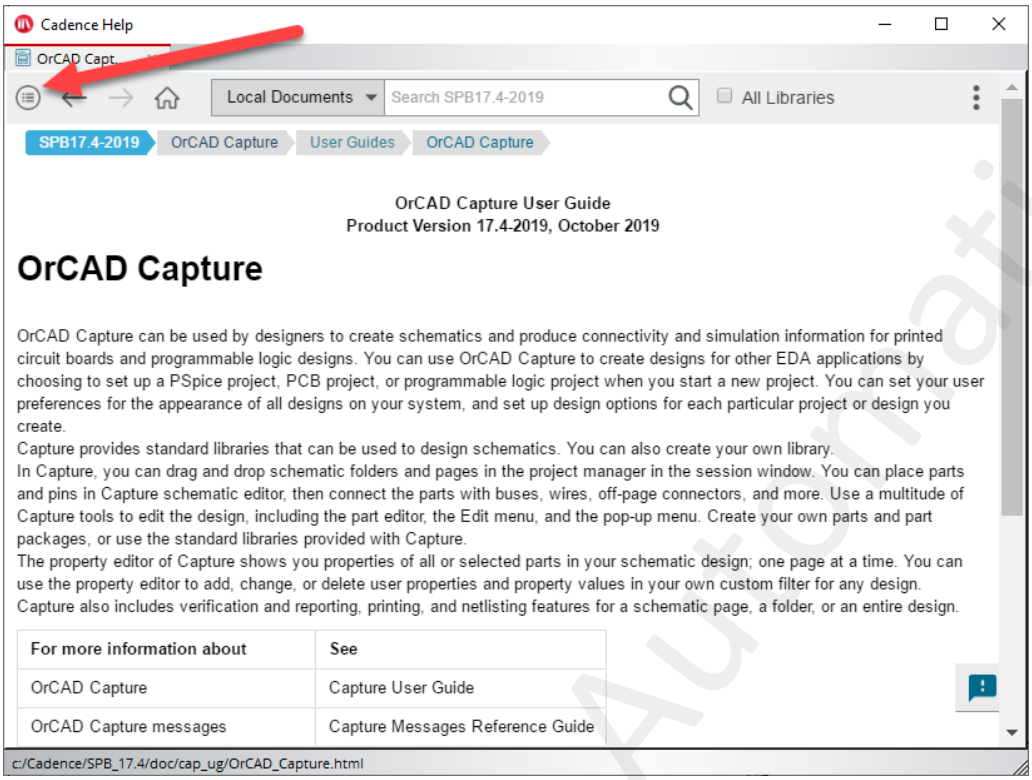
Your screen may look slightly different than this but there are a couple of categories to look at:

- Unfortunately for us on these training machines, this whole page is links to online support at Cadence Online Support (COS) and since we don't necessarily have an account there, we won't check that out at this time but there is good content if you want to check it out on your own at a later time with access to a COS account.

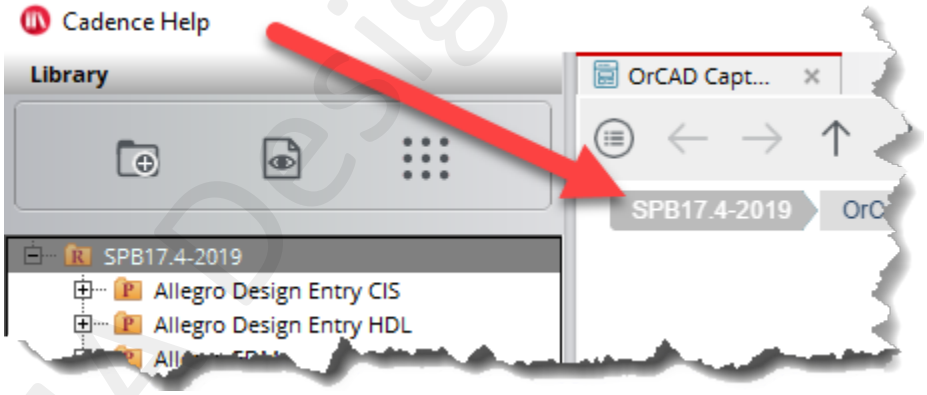


Documentation Browser

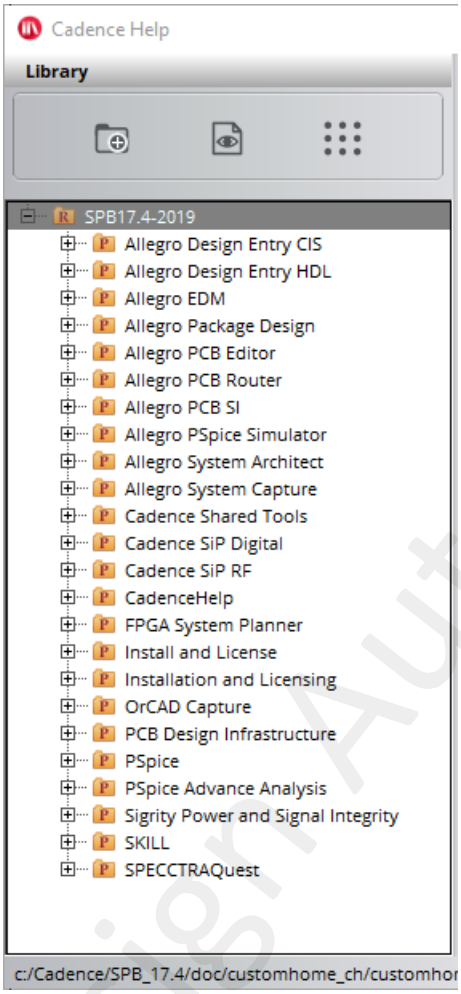
1. On the upper left, click on the ***Show Documentation Browser*** icon ( )



Alternatively, you can click the tab showing the present content to open the Documentation Browser.

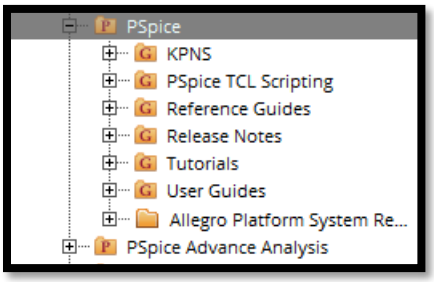


This will open the Documentation Browser on the left



You can see that there is local Help installed for every tool that cadence offers in their SPB (Silicon-Package-Board) software but we're not really interested in seeing everything, we're focused on PSpice.

2. Expand the PSpice folder to see what's all under there

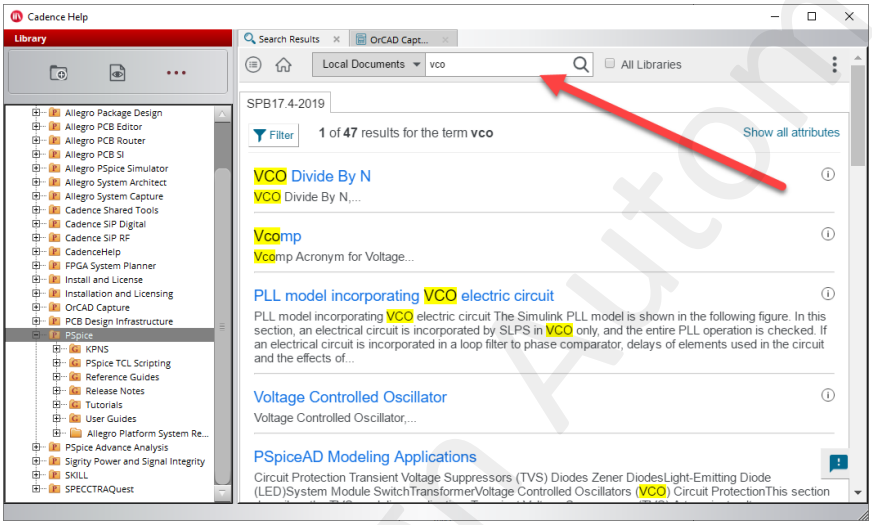


There are a lot of useful documents but for a new user, it's tough to know what information will be in what subfolder. We'd like to be able to search through all the folders.

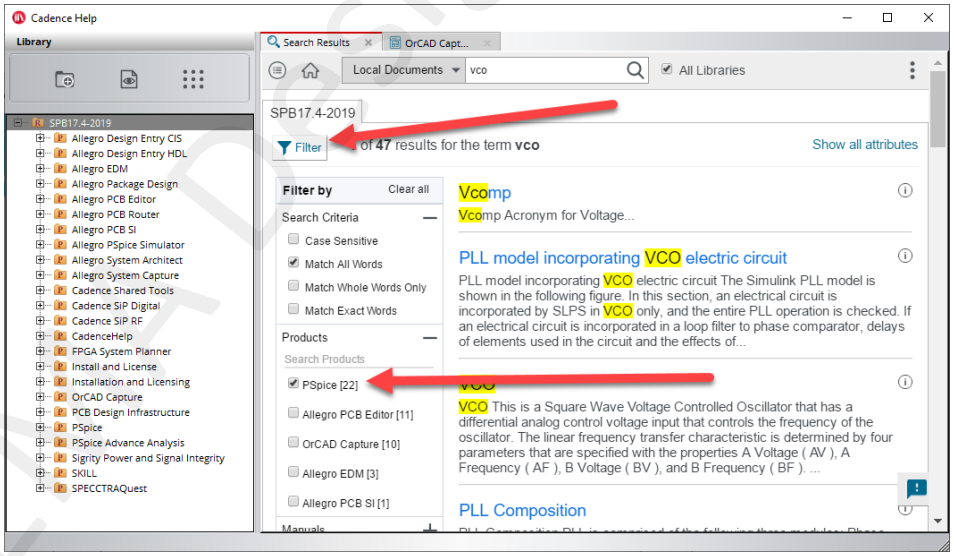
Search Documentation Browser

1. In the upper center of the browser, type in something you'd like to know more about, in this case, we'll search for **vco** to learn about PSpice's Voltage Controlled Oscillator

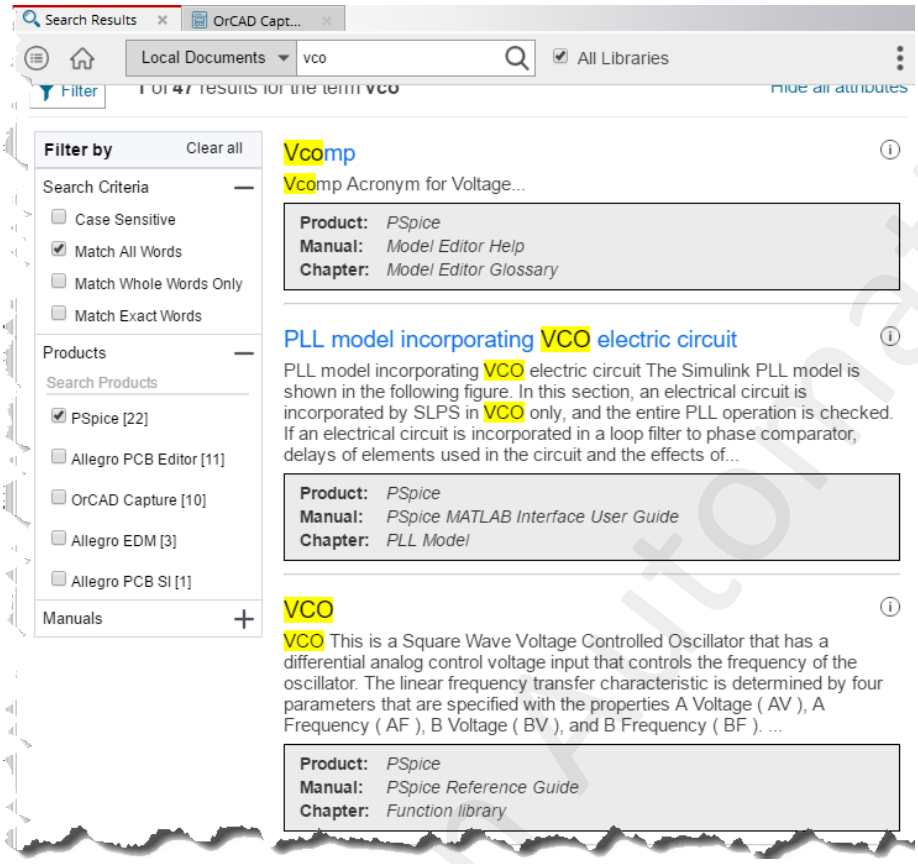
Here's the important and tricky part. If you just hit the search icon to the right of that, it'll search all Cadence documents for VCO which isn't what we want, we want to just search the PSpice documents, so hit the search icon.



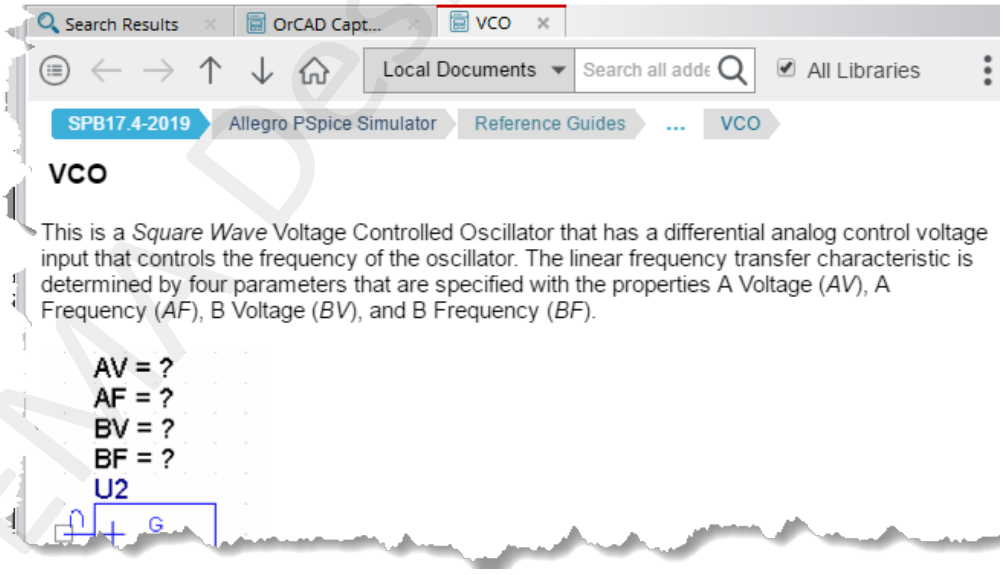
2. Then choose filter and select products > PSpice



This will filter all results to show *PSpice* Help items for the search term VCO and give you results in the right hand side.



3. The third item is VCO in the PSpice Reference Guide, click on this link to learn about the VCO part from within the PSpice Reference Guide

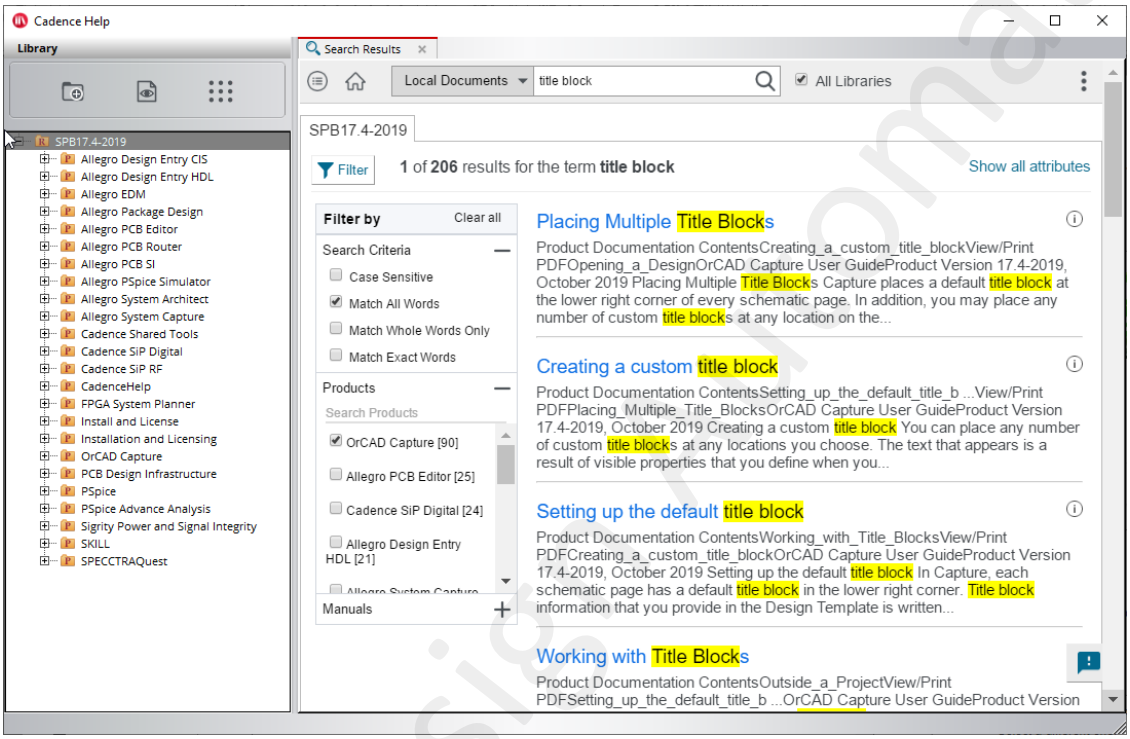


Using this method, you can quickly and easily search through all of PSpice help for whatever you would like to learn about.

Additional Exercise

See if you can figure out how to search the ***OrCAD Capture*** help for information on how to make a ***title block***.

Here's a clue on what it will look like:



Lesson 1, Lab 3: Creating an RC Circuit

Lab Objectives

After you complete this lab you will be able to:

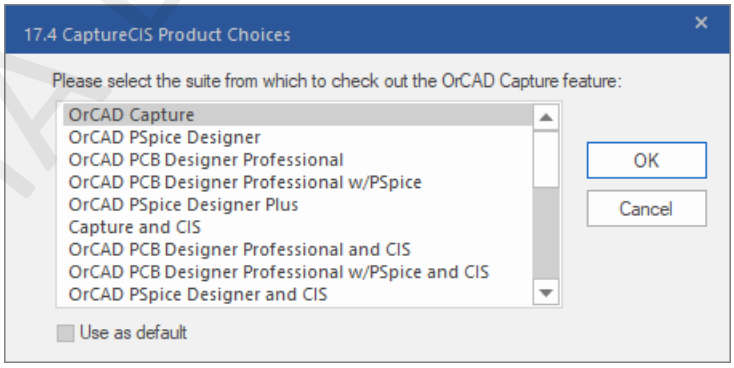
- Create a new PSpice A/D project
- Place parts
- Place wires
- Place power and ground symbols
- Assign reference designators
- Label nets
- Assign voltage values

Creating a New PSpice A/D Project

This lab begins with the assumption that Capture is not currently running. If it is you can just skip to step 3 below.

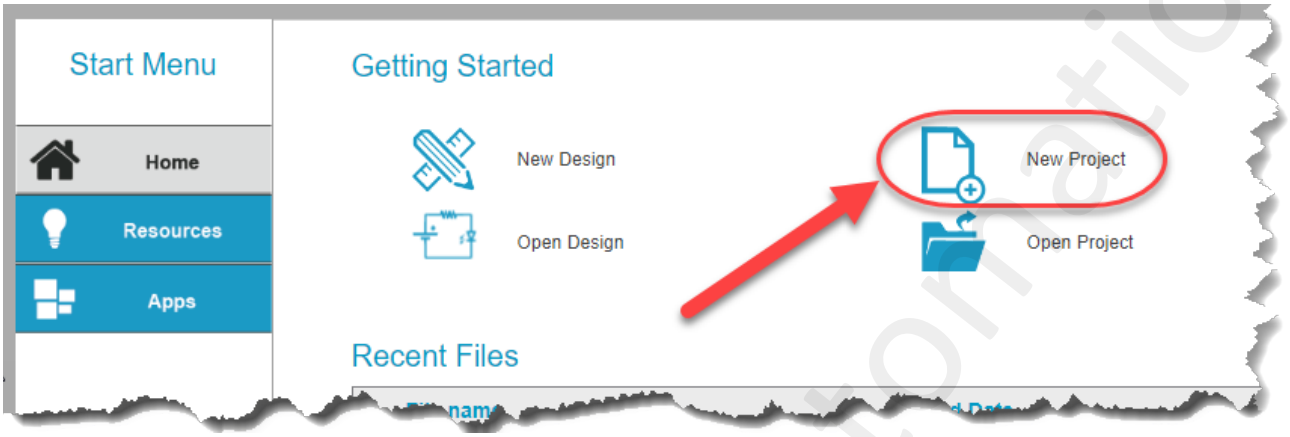
1. To start OrCAD Capture and begin your first PSpice project, select ***Start > Programs > Cadence PCB 17.4-2019 > Capture CIS 17.4***.

2. OrCAD Capture will respond by prompting for an appropriate license. Depending on what your machine is licensed for, there may be several PSpice license options to choose from. Pick either *OrCAD PSpice Designer, OrCAD PCB Designer Professional w/PSpice,* or *OrCAD PSpice Designer Plus* (basically, any license with "PSpice" in it).



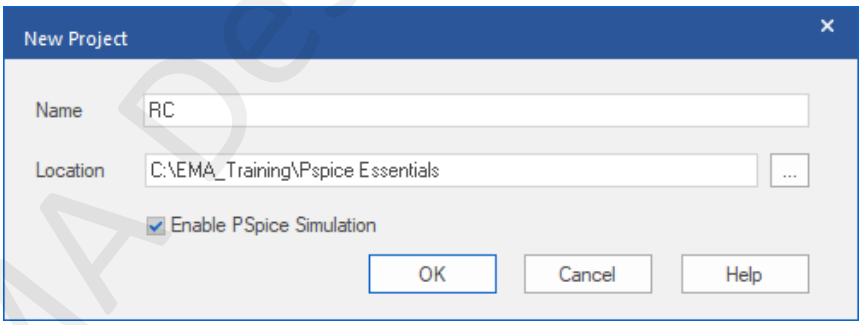
At this point, OrCAD Capture will open presenting (in most cases) the Start Page. This is really a control center of sorts for your Capture projects.

3. Choose the ***New Project*** button under the *Getting Started* Section of the *Start Page* to launch the *New Project Wizard*. Alternatively, you can select *File > New Project* from the pull-down menu at the top of the Capture window.



4. This will bring up the *New Project Wizard.* Fill out the following items:

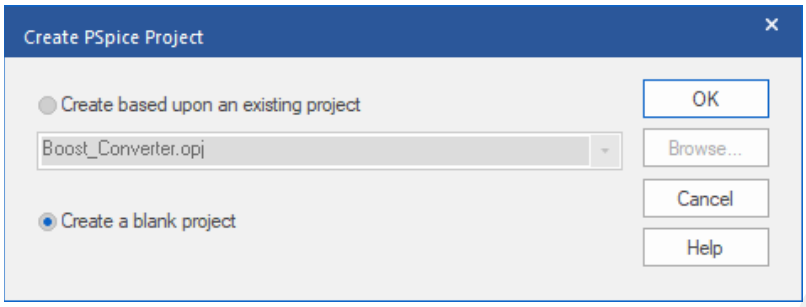
   - Enter **RC** in the ***Name*** field.
   - Next to the *Location* field click the ***Browse*** button, and navigate to the `C:\EMA_Training\PSpiceEssentials\Designs` directory. Your new project will be created and saved in this directory.
   - Check the "*Enable Pspice Simulation"* box (Required for PSpice designs).
   - Select the ***OK*** button when completed.



5. The *Create PSpice Project* panel will come up. This is where you can choose a pre-set template to base the project on, an existing project, or just run with a blank project. Project templates for the training machines are located in:

   - `C:\Apps\SPB_17.4\tools\capture\templates\pspice`

6. For our purposes we'll just create a blank project, so select the *Create a blank project* radio button and select *OK*.



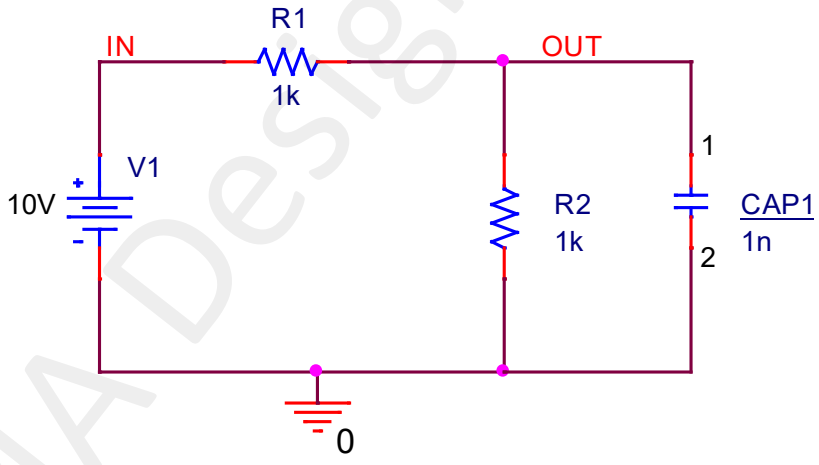
7. This will open the Project Manager

Notice that a new project entitled *RC* has been created (*RC.opj* tab) with a project design file (*RC.dsn*) and has configured the relevant structure for a PSpice A/D design.

8. Expand the *RC.dsn* entry and the *SCHEMATIC1* folder within it to show the initial schematic page. Either double-click the item name or click the plus sign to the left of the item to expand and display the item contents.

9. Double-click ***PAGE1*** to open the schematic drawing window. You should see a schematic page containing only a title block (created by default *Design Template*).

As you complete the remaining steps in this lab, you will be creating the RC circuit shown below:



Placing Parts (QuickPlace method)

To get started with our circuit design, we will begin by placing all our necessary components.

To Place the Resistors:

1. Choose *Place > PSpice Component > Resistor* from the pull-down menu. You will now see a resistor attached to your mouse pointer in a horizontal position.

2. Place *R1* first by just clicking the left mouse button to place the part in the desired location.

3. After placing the first horizontal resistor, press the <*R*> hotkey to rotate the part and place the vertically oriented resistor *R2* to the right of *R1*.

Place the Capacitor

4. Choose *Place > PSpice Component > Capacitor* from the pull-down menu.

5. Press the <*R*> hotkey to rotate the part, and place to the right of *R2*.

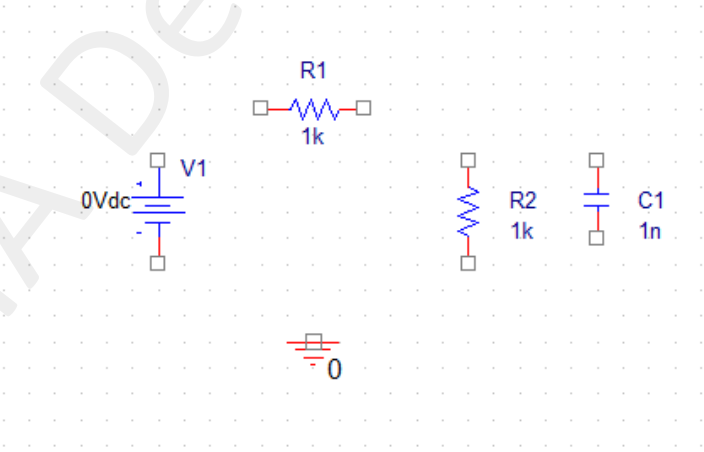
Place the Ground

6. Choose *Place > PSpice Component > PSpice Ground* from the pull-down menu and place the symbol somewhat below *R1* at the bottom of the circuit.

Place the Voltage Source

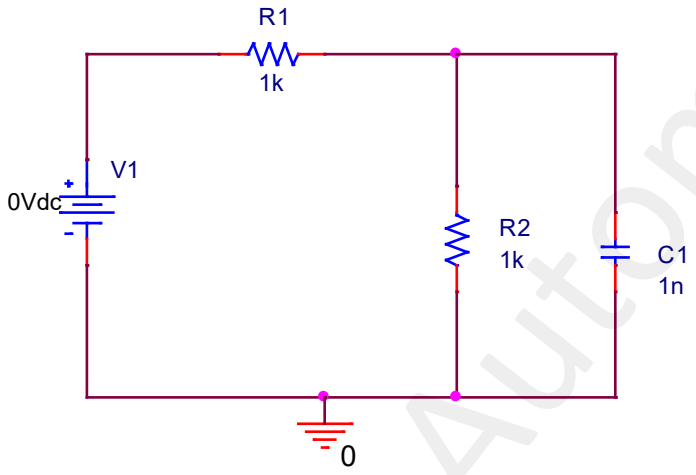
7. Choose *Place > PSpice Component > Source > Voltage Sources > DC* from the pull-down menu, and place the part to the left of *R1*, about the same level as *R2* and *C1*.

You should now have something that looks like this:



Connecting the Components

1. Choose either ***Place > Wire*** from the pull-down menu**,** press the **<*W*>** hotkey, or click the *Place Wire* toolbar icon ( ), and then wire together the components as shown in the graphic below (similar to the graphic provided at the beginning of this lab).



If you make a mistake drawing wires or placing components simply select the item and press the *Delete* key to remove it. You may also click the item to select it, and then drag it to the correct location. Wires can be stretched in OrCAD Capture by selecting the segment and then dragging from either end.

Assigning the Capacitor Reference Designator

1. Double-click the capacitor's reference designator *C1*.
2. Change the ***Part Reference*** property to the value ***CAP1*** and select ***OK***.

This will put an underline on the reference designator indicating that you have changed the reference designator manually. Capture will treat this change as if you intend to have that as the reference designator value so it will not be overwritten when doing an annotation.

Labeling Nets

1. Choose either ***Place > Net Alias*** from the pull-down menu**,** or press the **<*N*>** hotkey, or select the *Place Net Alias* toolbar button ( ).
2. Type **IN** in the ***Alias*** name field and select ***OK***.

3. This will attach an alias box to the mouse pointer, where you can then place the alias on the net between V1 and R1.

4. Repeat steps 1-3 to label the node between *R1* and *CAP1* as ***OUT***.

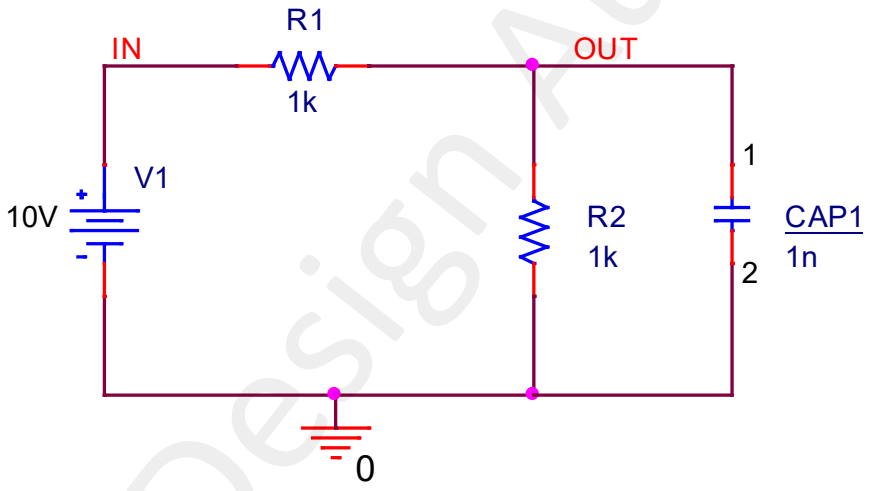
5. Press *<esc>* to end the mode.

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** and select ***OK***.

You should now have a circuit that looks like this (same graphic provided earlier in the lab). Your schematic will not show the pins on the capacitor, but they've been added here for you to show directional connectivity that will be useful in later labs.



Saving the File

1. Choose ***File > Save*** or click the Save toolbar icon ( ).

Lesson 2. Bias Point Analysis

Capture Keyboard Shortcuts

OrCAD CADENCE PCB SOLUTIONS

EMA Design Automation

Capture Keyboard Shortcuts

Cheat Sheet

- R – Rotate
- H – Mirror Horizontally
- V – Mirror Vertically
- C – Re-center (hold for pan)
- I – Zoom In
- O – Zoom Out

- F4 – Repeat Command
- F5 – Redraw
- F6 – Cross Hairs

- Ctrl + Drag – Copy Component
- Alt + Drag – Rip Component
- Ctrl + I – Selection Filter

- Middle Mouse Button scroll
  - Default – up & down
  - + Shift – left & right
  - + Ctrl – zoom in & out

 38

Capture and PSpice Keyboard Shortcuts

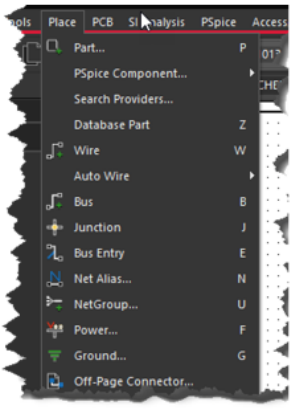
OrCAD CADENCE PCB SOLUTIONS

EMA Design Automation

Capture & PSpice Keyboard Shortcuts

Cheat Sheet

- P – Place Part
- Z – Place Database Part
- W – Place Wire
- B – Place Bus
- J – Place Junction
- E – Place Bus Entry
- N – Place Net Alias
- U – Net Group
- F – Place Power
- G – Place Ground
- Shift + R – Independent Sources
- F11 – Run Simulation



 39

Setting up a Simulation Profile

The simulation you create in this lesson calculates the bias point of a circuit using the voltages provided by the sources and any initial conditions set on the devices or nodes in the circuit.

As described in the lesson topics that follow, the first step in performing a design simulation is to create one or more `simulation profiles`.

Each simulation is defined and enabled through a simulation profile, which contains information necessary to run a single design analysis. Each PSpice A/D design must have at least one or more simulation profile associated to it.

The profile may contain:

- Simulation commands and options
- Included files – usually a circuit fragment to be included in the netlist
- Library statements – any custom model libraries that will need to be referenced during simulation
- Stimulus file references – any stimulus libraries created in the Stimulus Editor
- Probe window display preferences – details when Probe will start and how it will look when it does

Spice Algorithm

PSpice uses Kirchhoff's current law which states that the sum of all currents at any node equals zero as the basis for its matrix creation

The Simulation Settings Dialog Box

Simulation profiles are set up through the Simulation Settings dialog box.

The ***Simulation Settings*** dialog has six sections listed as tabs across the top of the dialog. As shown in the graphic above, the section tabs include: General, Analysis, Configuration Files, Options, Data Collection, and Probe Windows.

The sub-sections that follow describe the contents of each tab.

General Tab

In this tab you can name the profile, change the input design file, specify the output file name, specify the data file name, and enter notes about the profile or simulation that you will be running.

Analysis Tab

Select the ***Analysis*** tab to select the type of simulation you want to run. The four basic simulation types available include:

- Transient – time domain
- DC Sweep
- AC Sweep – small signal frequency analysis
- Bias Point

Combining a basic simulation with one or more optional analyses, such as Parametric or Monte Carlo can further enhance any of the first three analyses.

As you select each different analysis type, option settings available in the dialog box change that are pertinent to the type of analysis you choose.

Configuration Files Tab

The ***Configuration Files*** tab combines model libraries, include files, and stimulus libraries.

Option settings within the Configuration Files display provide the following three ways to configure model libraries, include files, and stimulus libraries:

- Local configuration to a profile
- Local configuration to a design
- Global configuration where the model library is accessible to all designs

Files with global configuration have a world icon in front of their name in the list of configured libraries.

The library file `Nom.lib` is automatically configured when the program is installed. It is a master list of all the installed PSpice A/D model libraries. `Nom.lib` is a simple text file that you can edit. You might want to edit it to add custom libraries that will be used in all of your future projects. If you do add libraries to `nom.lib`, do so only after they are stable. Every time any

library is edited, a new index file is generated. Since nom.lib is a master list of all libraries, this can take a considerable amount of time.

Libraries and include files, which contain process models and parameters, may have different simulation profiles created for different process corners. In earlier application versions library and include files could only be configured for an entire design, (all profiles), or globally. Because stimulus files can now be unique per profile, a variety of test inputs can be contained in corresponding simulation profiles.

UI changes include three categories (Stimulus, Libraries and Include Files) accessed from a single tab (Configuration Files). There are icons displayed which change depending on whether the file has been linked globally or locally to the profile or design.

Global information resides in the PSpice.ini file. Design level information resides in the `<projectname>.opj` file while profile information resides in the <ProfileName>_profile.inc file. For example, `RC_profile.inc` is located in the ***<ProfileName> folder (- Bias)*** of the new directory structure.

Options Tab

Four sections are included within this tab: Analog simulation; Analog Advanced; Gate-level simulation; and Output File. Dialog options that display within each dialog box vary.

Analog Simulation – This dialog box allows access to all of the tolerances and settings used by the simulator. **Reltol**, **ITL1**, **ITL2** and **ITL4** are the most commonly used options in this section since they can be used to solve a majority of transient and bias convergence problems. This section also contains the Autoconverge options.These will be discussed in more detail in the chapter on convergence.

Analog Advanced – Advanced simulation settings for bias point and transient analysis.

Gate-level Simulation – This dialog box allows you to set the digital simulation options. The most commonly used option in this dialog is the one to initialize all flip-flops in a design at the start of a simulation.

Output File – This dialog box allows control of what is printed to the output file. This does not affect the data file used to display results in the Probe window. Many of these options are helpful when trouble shooting simulation errors.

Data Collection Tab

In this dialog you can control how much and which data is saved to the data file that is used to examine the simulation results in the Probe window. This is most useful when you want to

restrict the amount of data saved during a simulation either because of the resulting file size or because of simulation speed.

Probe Windows Tab

In this dialog you control when the Probe window will be opened (during simulation or after) and what will be displayed when it is opened.

Running a Bias Point Simulation

The bias point simulation calculates the bias point based on DC source values and any initial conditions set on the circuit. If there are transient sources present in the circuit the bias point calculation will use their value as defined at time=0.

If you have not netlisted the design before running the simulation, the netlister automatically runs when simulation begins. Because netlisting automatically verifies compliance to design rules, a PSpice A/D netlist is created if the design rule check (DRC) does not find errors. You can view the resulting netlist by using the ***PSpice > View Netlist*** menu command.

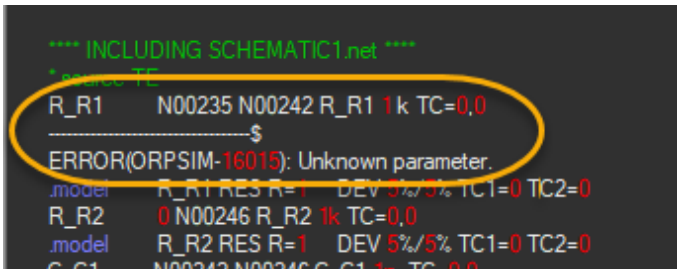
Examine the Output File and Bias Display

There are two ways to view the results of a bias point simulation. You can view the contents of an output file. You can also view the results directly on the schematic.

Reviewing the Output File

Although the Bias point simulation does not produce a data file for use in the Probe window, the results of a simulation are included in an output file. You can view the output file by choosing ***PSpice > View Output File***. Output file syntax is presented in easy to read colors that highlight the different groups such as text, numbers, comments, expressions, operator, and keyword. You can also specify your own color schemes for the output file by editing the `SpiceSyntax.ini` file in the ***C:\Apps\SPB_17.4\tools\pspice*** directory. Simulation errors are included within this file.

The graphic that follows shows a partial list of an output file that shows the bias voltages for the circuit. This output file has an error because there is a blank space between 1 and k. Notice how the error is flagged with the '$' symbol beneath the cause of the error.



Reviewing Results from the Schematic

In addition to being able to view the results of a bias point analysis from an output file, you can also view results directly from the schematic page. You can enable the voltage and/or current bias display for all nets and devices, or for just selected ones by choosing ***PSpice > Bias Points*** and then choosing the desired menu item. You can also toggle the bias display by clicking the bias toolbar icons.

Voltage labels are placed on each net after simulation. Bias voltage is measured with respect to ground, which is always zero.

Current labels are placed on one pin of all two-pin devices and on all pins of devices with more than two pins. The current shown is measured as flowing into the pin to which the current display is attached. When the current through the capacitor is zero, a current display is not attached.

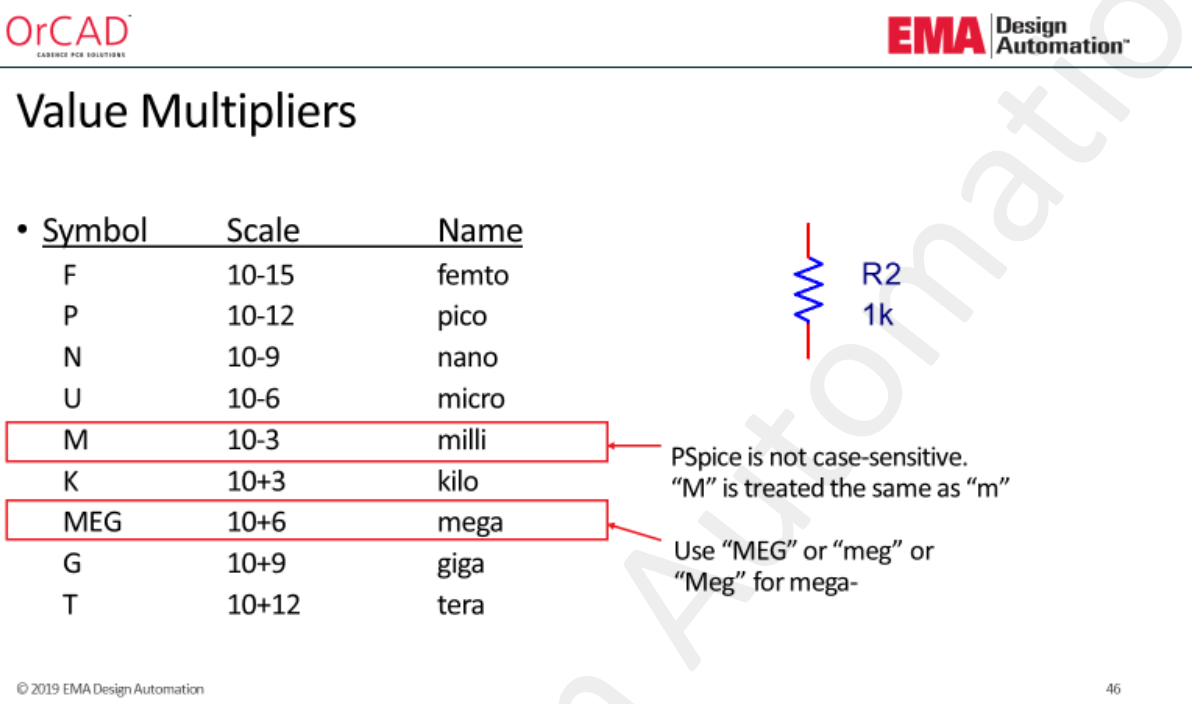
Dissipated power labels are placed on each discrete PSpice A/D device. Dissipated power labels will not be shown for subcircuits or digital devices.

In the graphic below, notice how the voltage at net OUT is half the voltage at net IN in the annotated schematic. The voltage difference is recorded in the output file as well as the annotated schematic. This agrees with the voltage divider principal that applies to our RC circuit:

$$V_{CAP1}=R_2 / (R_1+ R2) * V1$$

Also notice how the current through V1 is negative. By convention, PSpice A/D measures the current through a two terminal device as flowing into the first terminal and out of the second. For voltage sources, current is measured from the + terminal to the – terminal which is opposite to the positive current flow convention. In the above graphic, the current flowing into R1 is 5.0mA and is flowing from left to right.

Value Multipliers



PSpice A/D assumes base unit of measurement for all design objects. For example, ohms are the assumed unit of measurement for resistors. Volts are assumed for voltage sources, farads for capacitors, and miles for transmission lines.

PSpice A/D supports multiplier suffixes for device values. The following table lists the available suffixes.

Rules for Multiplier Suffixes

PSpice A/D is not case sensitive and will interpret "m" to be the same as "M".

Do not use a space between the number and its multiplier suffix. Since PSpice A/D uses white space as a delimiter when netlisting, spaces cause an error or invalid results.

Except for **MEG** and **MIL**, PSpice A/D looks only at the first letter following a number so it is possible to have a capacitor value of .47uF.

PSpice A/D always assumes base units for each device type: Resistors are measured in Ohms, capacitors in Farads, voltage sources in volts, etc.

Numeric vales may be represented in scientific notation. In other words, a resistor with a value of one thousand ohms may be represented as either 1k or 1E+3.

Lesson 2, Lab 1: Simulating a Bias Point

Lab Objectives

- Configure a Bias Point Analysis
- Run the analysis
- Examine the results in the output file
- Examine the results using bias displays

Creating a Bias Point Simulation Profile

1. If not already open, open the RC design.
2. Create a new profile (***PSpice > New Simulation Profile*** or  icon)
3. Enter **RC_bias** for the name of the profile.
4. Click the ***Create*** button.

The ***Simulation Settings*** dialog box opens.

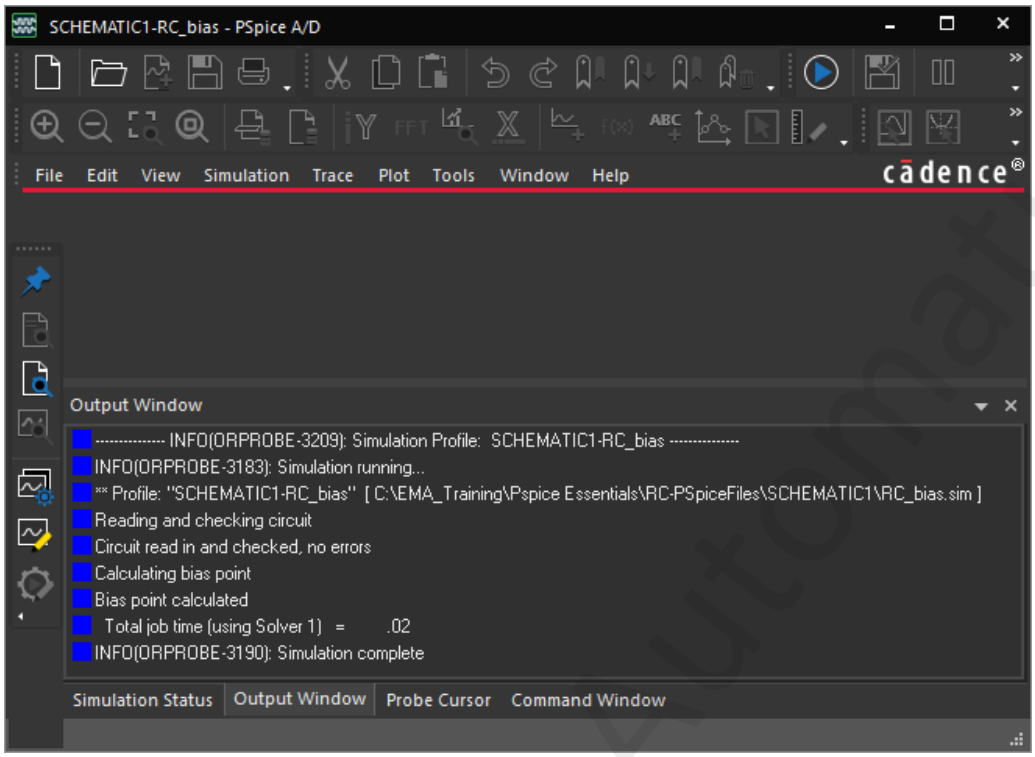
5. Click the ***Analysis*** tab (should be the tab you're on).
6. Choose the ***Bias Point*** option from the Analysis Type drop-list menu.
7. Click ***OK***.

The Simulation Setting dialog box closes and your display returns to your schematic drawing.

Running the Simulation

1. Choose ***PSpice > Run*** or click the ***Run*** toolbar icon. 

The netlister and DRC automatically run and the Probe window opens as shown in the graphic that follows.



PSpice A/D does not generate Probe data for bias analysis. Consequently, the Probe window display shown in the above graphic is empty.

To view results from the output file:

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.

1. From the active schematic page, choose the ***PSpice > View Output File*** menu to examine simulation results in the output file.

Or from the Probe window, choose the View > Output File menu.

Your results should look similar to the text that follows.

```
**** 01/14/20 13:06:04 ******* PSpice 17.4.0 (Nov 2018) ******* ID# 0 ********

 ** Profile: "SCHEMATIC1-RC_bias"  [ C:\EMA_Training\Pspice Essentials\RC-PSpiceFiles\SCHEMATIC1\RC_bias.sim ]


 ****     CIRCUIT DESCRIPTION


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



** Creating circuit file "RC_bias.cir"
** WARNING: THIS AUTOMATICALLY GENERATED FILE MAY BE OVERWRITTEN BY SUBSEQUENT SIMULATIONS

*Libraries:
* Profile Libraries :
* Local Libraries :
* From [PSPICE NETLIST] section of C:\Users\AlaiyaT\cdssetup\OrCAD_PSpice\17.4.0\PSpice.ini file:
.lib "nom.lib"

*Analysis directives:
.OPTIONS ADVCONV
.PROBE64 V(alias(*)) I(alias(*)) W(alias(*)) D(alias(*)) NOISE(alias(*))
.INC "..\SCHEMATIC1.net"
```

2. Close the output file. (***File > Close***)

To examine simulation results from the schematic:

1. Choose ***PSpice > Bias Points > Enable*** to turn on the voltage and current bias displays (if it is not already)

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

3. Select any of the nets in the schematic and then click on the ***Toggle Voltages*** toolbar icon.  This enables you to toggle the voltage display of a single node.

Optional Lab Exercise

1. Edit the schematic and remove the ground symbol.

2. Simulate the circuit again.

What errors do you get? Why?

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, a non-‘0’ ground or an isolated node between two capacitors.

3. Replace the ground symbol after completing this exercise.

Lesson 3. DC Sweep Analysis

DC Sweep Settings and Options

As you configured the bias point analysis through the simulation profile, the DC Sweep analysis is also configured through the simulation profile and Simulation Settings dialog box which is available through the ***PSpice > New Simulation Profile*** or the ***PSpice > Edit Simulation Profile*** menus.

You can setup the simulation by either editing an existing profile or by creating a new profile. If you edit and save an existing profile, the modified settings overwrite the old settings. Consequently, the previous analysis is no longer be available. If, however, you want both analysis types to be available, then you should create a new profile.

When you select the ***Analysis*** tab and the ***DC Sweep*** option in the Simulation Settings dialog box, dialog box options vary with the selection of different analysis types. Fields available for editing are dependent on the options, variables, and sweep types you chose. For example, if you are sweeping a source, then **Model Name** and **Model Type** become unavailable for editing.

Data Collection

Before simulating, select options under the ***Data Collection*** tab of the simulation profile.

By default, all voltages, currents, quiescent power dissipation, and digital states are saved. These settings save all of the data for every node and device in the circuit, even those nested deep inside subcircuits. This has the advantage of making every possible piece of data available to the Probe Window. A very large data file may result from the completed analysis.

Selecting ***All but Internal Subcircuit*** restricts the simulator from saving data at the internal nodes of subcircuits. This helps reduce the size of the data file when design details of subcircuits are unnecessary.

Selecting ***At Markers Only*** also helps reduce the size of the data file. The disadvantage is that if you run a long simulation and need data from a node or device without a marker, you need to place a marker and then re-run the simulation from the beginning.

Selecting ***None*** causes the simulation to run without saving data to the data file. This is useful when you are solving convergence problems and don't want to waste time writing data to your hard drive.

Checking ***Save Data in CDSF Format*** saves the simulation data in an ASCII rather than binary file. This is helpful if you wish to export the simulation data to another application.

Always check to be sure that options within the **Probe Window** and **Data Collection** tabs are configured before running your simulating. These settings cannot be applied to data after a simulation. Few things are more frustrating than running a two-hour simulation only to find that data has not been saved.

Extracting Simulation Data in ASCII Format

To easily get the ASCII probe data without by using the ***Save as CSDF Format*** check box:

1. Run the simulation as normal.
2. Examine the traces of interest in Probe.
3. Below the X axis, select the names of the traces for which you want to extract data. Hold the **<Shift>** key while clicking to select multiple traces.
4. Use ***Edit > Copy*** or ***<CTRL+C>*** to copy the traces to the paste buffer.
5. Open the text editor, word processor, or spreadsheet of choice and paste in the data.

The ASCII text data is pasted instead of the probe display.

Copying the Probe Display into a Document

To paste the entire Probe display into your favorite word processor:

1. Use ***Window > Copy to Clipboard***.
2. In the resulting dialog, check the selection option `Make window and plot backgrounds transparent`.
3. Select the format of the foreground as desired.
4. Go to your favorite word processor and paste in the image.

The imported image has a white background with either colored or black traces that correspond to your selected foreground options.

Examining Results in Probe

The ***Probe Window*** tab of the simulation profile allows you to specify when to open the ***Probe Window*** and what to have displayed when it is opened.

Choosing ***PSpice > New Simulation Profile*** or ***PSpice > Edit Profile*** and then selecting the ***Probe Window*** tab opens the setup dialog for the Probe window.

You can specify that the ***Probe Window*** is opened whenever the profile is opened. You can also specify whether the ***Probe Window*** is opened after or during a simulation (marching waveforms). Having Probe open during a simulation is useful when you want to check the validity of a long simulation early into the run.

The ***Last Plot*** option allows you to restore the settings from the last probe display but using the new simulation data. This is useful if you have a complicated Probe display setup such as multiple axes with custom scales and then find that you need to re-simulate. This option allows you to apply the setup to the new simulation data.

Selecting ***Nothing*** opens the Probe window without displaying traces. You can then add the desired traces by hand.

Setting up Sources

Every PSpice A/D source symbol has a DC property. If you are going to perform a DC sweep analysis, then every source in the design should have a value specified for its DC property. If no value is specified, zero is assigned by default. If you configure the analysis to sweep a voltage or current source, then the value of the source will be set by the simulation profile during simulation. The value of the source seen on the schematic is merely a place holder which is ignored.

The most common DC sources to use for a DC Sweep analysis are the VDC (voltage source) and the IDC (current source). Both symbols have a default value of zero, which is displayed next to the placed symbol. To change the value, you can double-click the visible value and enter the correct value into the Display Properties dialog box that opens.

You may also select the symbol, right click and choose ***Edit Properties…*** to gain access to the Display Properties dialog box. This may be necessary when the attribute value is not visible in the design.

The DC property on a source symbol is **not** used as a DC offset during an AC analysis. It is used only for a DC Sweep analysis or as a default value for transient sources if other parameters are not defined.

Markers

Markers are used as a display shortcut to traces in the ***Probe Window***. For a DC Sweep analysis, only the basic markers will be available for you to place in the design. Basic markers include voltage level, differential voltage between two nodes, current into a pin on a two or three pin PSpice A/D device (i.e. not usable on parts that reference a subcircuit), and dissipated power markers for PSpice A/D devices.

Advanced markers allow you to display real and imaginary parts of voltages and currents as well as decibel values. They are available, however, only if you have previously created an AC Sweep analysis simulation profile. Advanced markers will be used in a future lesson.

Placing Basic Markers

Markers can be placed by choosing the ***PSpice > Markers*** menu or by clicking on the appropriate toolbar button. An image of the marker will be attached to the cursor.

Place the point of the marker on a node (for voltage markers), the pin hot spot (for current markers), or the body of a part (for power markers) and click to place the marker. As when placing any other part or symbol in OrCAD Capture, a copy remains attached to the cursor. You can place additional markers by left-clicking. Press ***<esc>*** or click your right mouse button to display the pop up menu and then select the ***End Mode*** menu option to exit the placement mode.

Voltage differential markers are always placed in pairs. The first marker placed will be the positive node and the second will be the reference node. That is, if the first marker is placed on a node that is 10V and the second is placed on a node that is 5V, then the value will be 5V. If the markers are reversed, the value will be -5V.

Be aware that if you place both current and voltage markers on a design and have Probe set up to display markers when it is opened, then you will have both voltages and currents on the same Y-axis. This usually results in the current traces appearing as flat lines since their scale is frequently a couple orders of magnitude smaller than the scale for voltage traces. The larger scale takes precedence on a Probe plot.

Customizing the Probe Display

The Probe window is the graphical waveform viewer. You can think of it as an electronic oscilloscope. Probe allows you to view simple voltages and currents for any node in the circuit as well as quiescent power dissipation for PSpice A/D devices. You may also view group delay

and the real and imaginary parts of voltages and currents (AC Analysis only). Probe can also perform complex mathematical calculations.

You may customize the probe display by right-clicking on traces, trace symbols below the x-axis, or on the axes themselves and then selecting the properties command. You may change the trace color and probe background and foreground color by choosing the ***Tool > Options*** and then selecting the ***Color Settings*** tab.

Editing the Probe display is especially useful if you are preparing an overhead slide for a presentation. Many of the newer printers have such fine resolution that any colored line is nearly invisible when printed at one pixel wide. By changing the line width, you can overcome this.

Multiple Y-axes

You may often want to view traces with widely different scales in the Probe window. For example, you might want to see both voltages and currents at the same time. Probe allows up to three y-axes per plot.

To add an axis, choose ***Plot > Add Y Axis***. A new axis will be added to the right of the existing axis and will be designated as the active axis. The active axis will be marked with a **>>** near the bottom left. A new axis will not have a scale applied to it until the first trace is added to it. The active axis will be the one to which new traces are added or to which cursors will be applied. The picture above shows an example of a Probe plot with a voltage and current axis. The axis for current is the active axis.

You may also move an axis to the right side of the plot. Choose ***Plot > Axis Settings*** and then click the ***Y Axis*** tab. Click in the ***Y Axis Number*** pull-down field and select the axis that you want to move. In the ***Axis Position*** area click the ***Left*** or ***Right*** radio button to pace the axis on the left or right edge of the plot.

Click the ***X Axis*** tab to open dialog that enables you to:

- Set the displayed data range.
- Set the range of data that Probe functions consider. This is useful when making a Fourier plot since you will want an integral number of cycles or perhaps data from only the last portion of the simulation after the circuit has settled.
- Specify whether the scale should be Linear or Logarithmic.
- Enable the Fourier mode or Performance Analysis mode.
- Change the X axis variable.

Clicking the ***Y Axis*** tab opens a dialog box that enables you to:

- Set the Data Range that will be displayed.
- Specify if the scale should be Linear or Logarithmic.
- Change the axis number and title.

Clicking the ***X Grid*** tab opens a dialog box that enables you to:

- Set the major grid spacing.
- Choose the way in which the grids will be drawn (lines verses dots or **+** 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 will be drawn
- Choose to display tick marks inside the plot edge.

Clicking the ***Y Grid*** tab opens a dialog box that enables you to do the same things that we saw above for the X Grid

Archiving a Project

If you have a PSpice project that you want to archive, send to a colleague or outside the company, it's best to use Archive Project under the file menu as this grabs all the necessary files from PSpice and puts them into a self-contained zip file that when extracted will contain everything needed to duplicate the simulation.

Lesson 3, Lab 1: Performing a DC Sweep

Lab Objectives

- Configure a DC Sweep profile
- Run the analysis
- Examine the results in Probe
- Use markers to display traces in Probe

Setting Up the Simulation

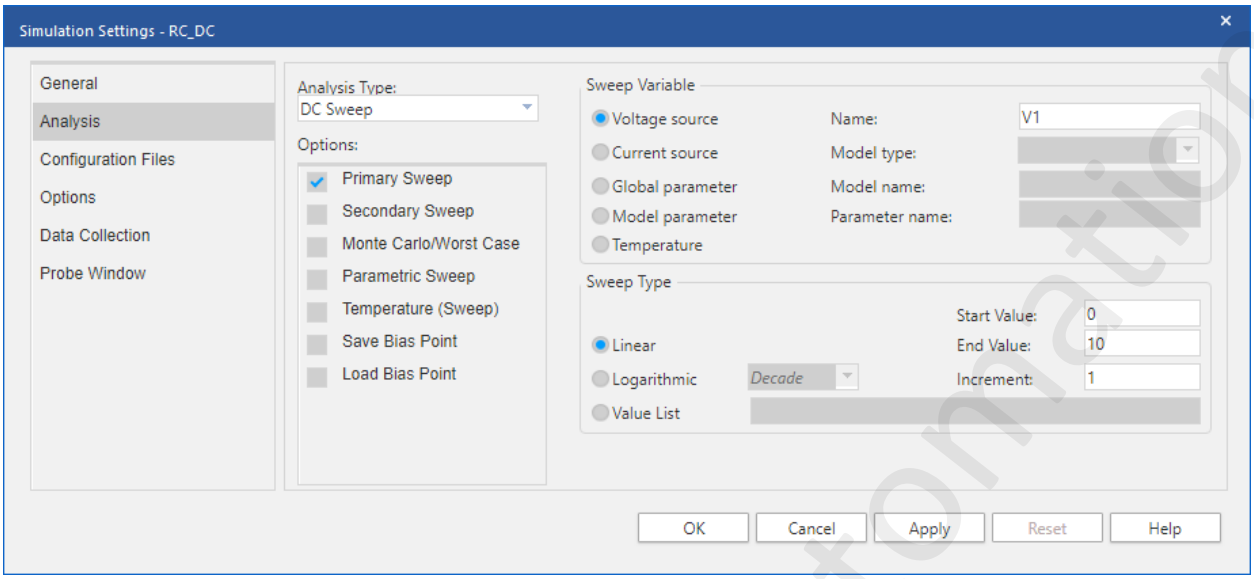
You can visually verify the voltage divider principle by performing a sweep of the DC voltage source and graphically displaying data in the Probe window. In this lab, you will set up a DC Sweep analysis of the source V1 from 0V to 10V in 1V increments.

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

1. Create a new profile (***PSpice > New Simulation Profile*** or  icon)
2. Name the new simulation **RC_DC**.
3. Click **Create**.

The Simulation Settings dialog box opens.

4. Click on the ***Analysis*** tab.
5. From the ***Analysis Type*** drop down choose ***DC Sweep***.
6. In the ***Options*** section only ***Primary Sweep*** should be highlighted and checked.
7. In the ***Sweep Variable*** section select ***Voltage Source***.
8. Type **V1** in the ***Name*** field.
9. In the ***Sweep type*** section check ***Linear***.
10. Enter **0** in the ***Start value*** field.
11. Enter **10** in the ***End Value*** field.
12. Enter **1** in the ***Increment*** field.



Setting Up Data Collection and Probe Windows

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

Data collection settings for voltages, currents, power, digital, and noise display.

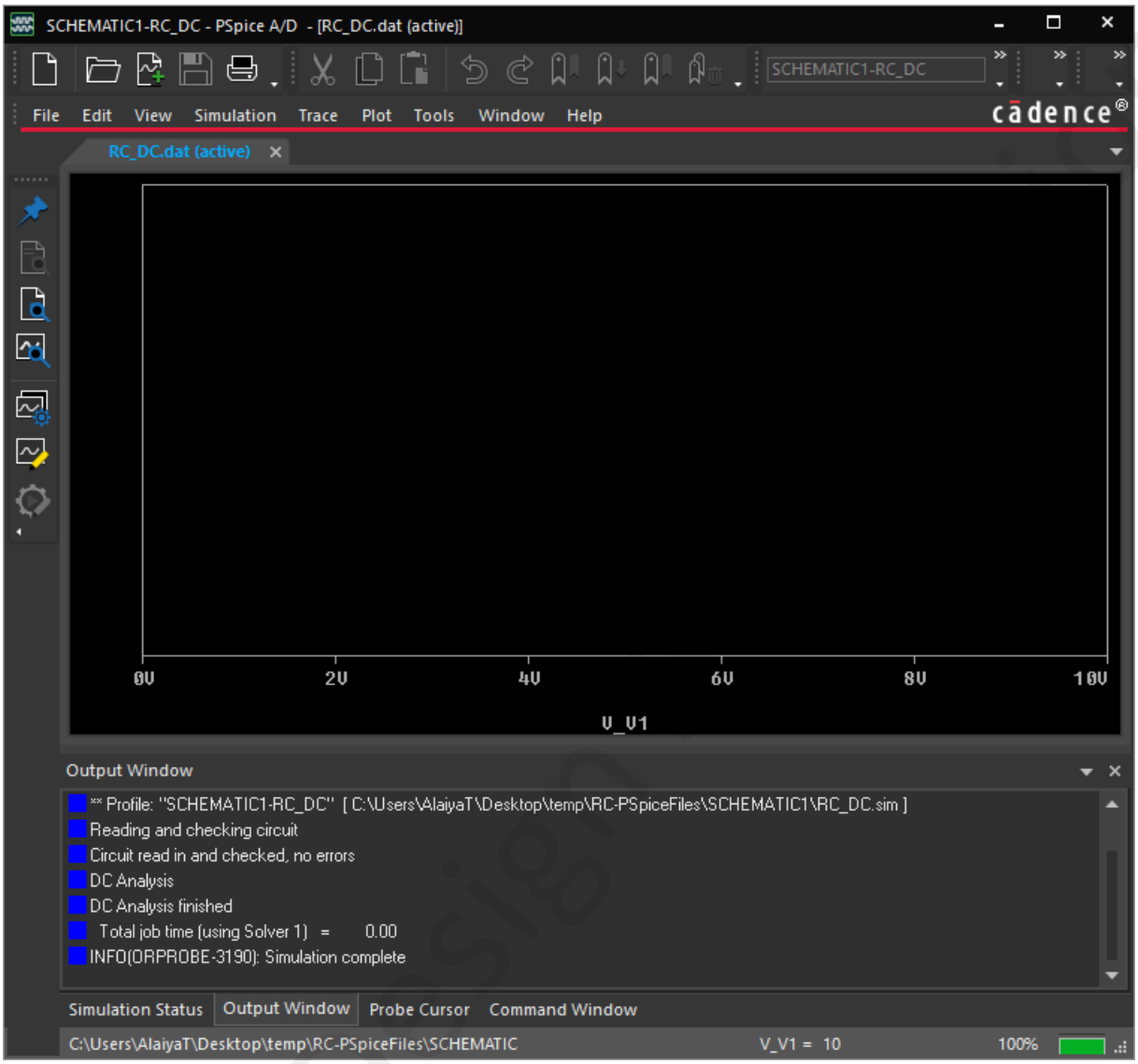
2. From the drop-list menus of each option, select ***All.***
3. Click the ***Probe Windows*** tab.
4. Check the option, ***Display the Probe Window after simulation has completed***.
5. Check the option, ***Show all markers on open schematics***.
6. Click ***OK*** to close the ***Simulation Settings*** dialog box.
7. From the schematic view, run the simulation (***PSpice > Run***, icon or F11 key)

The system automatically generates the netlist.

When the simulation run is complete the Probe windows opens.

Analyzing the Results in the Probe Window

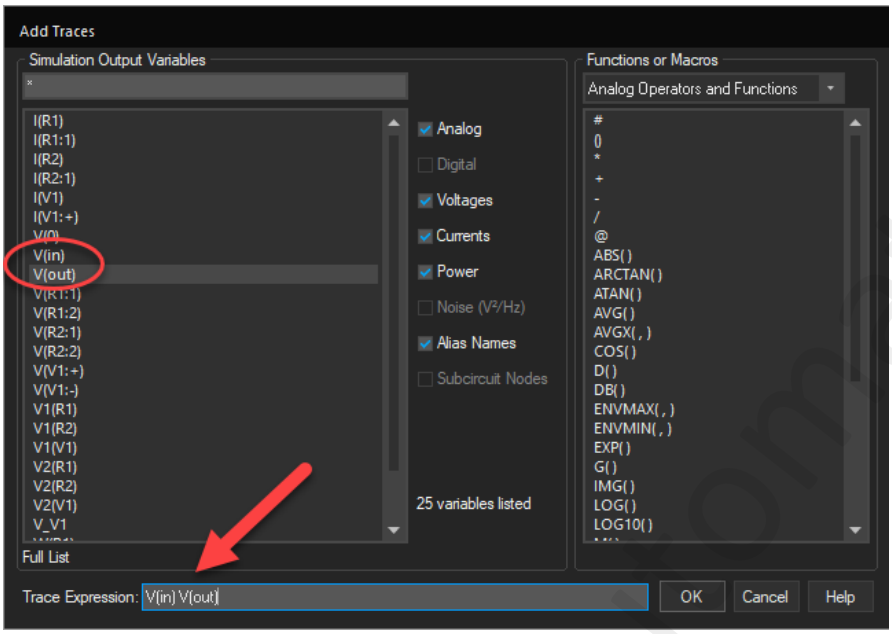
When the Probe window appears, you are presented with a blank plot screen. If you have used markers in the design, then you will see the traces for those markers already displayed.



Plotting Traces from the Menu

1. From the ***PSpice A/D*** window, choose ***Trace > Add Trace*** or click on the ***Add Trace*** toolbar icon. 

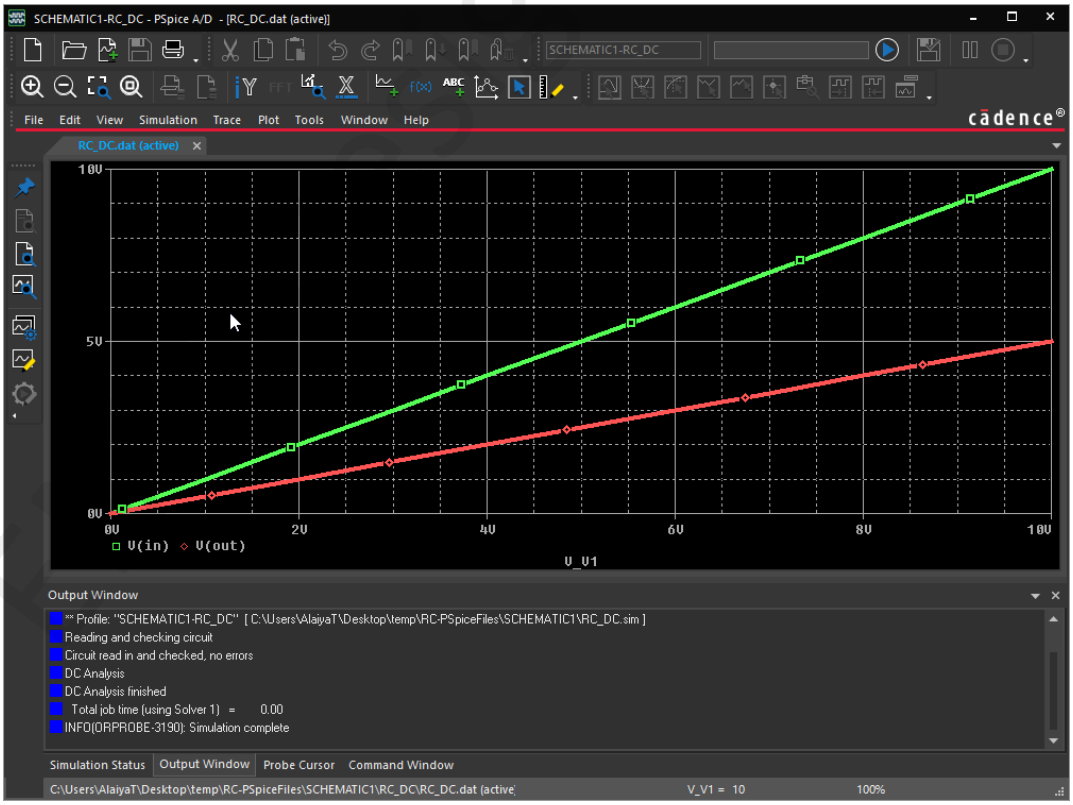
The Add Traces dialog opens.



2. Select ***V(IN)*** and ***V(OUT)***, which are listed in the ***Add Trace*** list box.

This adds the selected traces to the Trace Expression field at the bottom of the window. All traces listed in the Trace Expression field will be displayed at the same time.

3. Click ***OK*** to graphically display the traces as shown below.



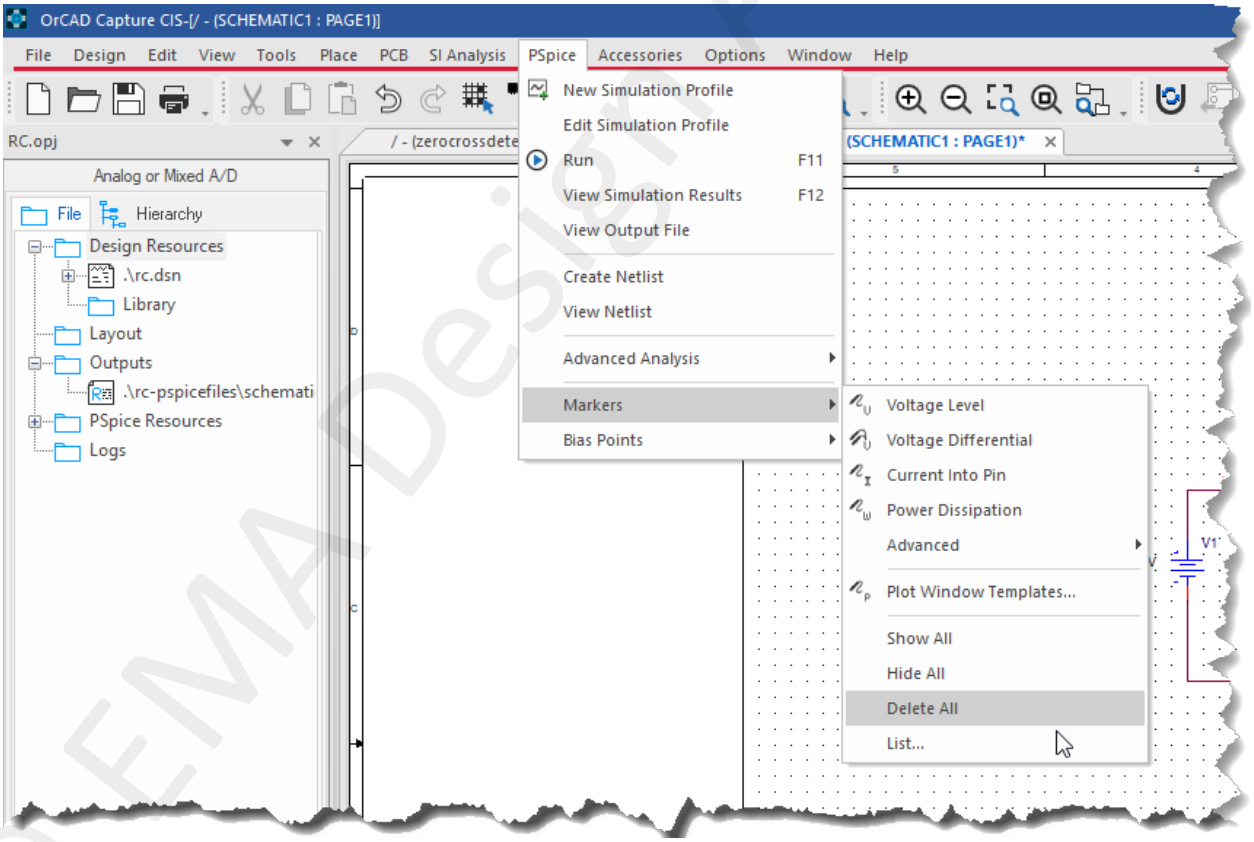
Plotting Traces Using Markers

To see how markers display traces in the Probe window, delete the two displayed traces and add them back in using markers as described in the steps that follow.

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

The color of the trace name changes to red indicating it is now selected.

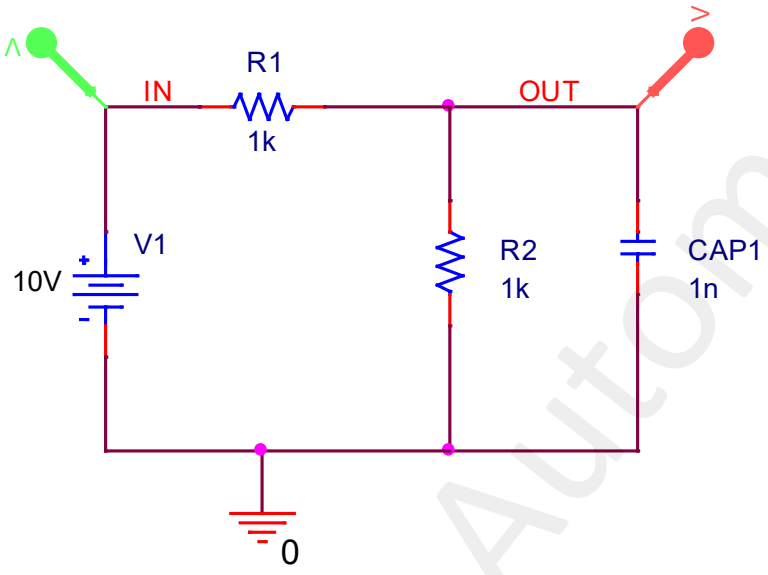
2. Hold the ***<Shift>*** key and select ***V(OUT)***.
3. Press ***<Delete>*** to remove both traces from the display.
4. Press ***<Alt+Tab>*** or click on the OrCAD Capture task bar button to switch back to the OrCAD Capture schematic.
5. Choose the ***PSpice > Markers > Voltage Level*** menu commands or use the 'Voltage/Level Marker' icon 



The Voltage Level marker attaches to your cursor.

6. Position your cursor on node ***V(IN)*** and click your left mouse button to place the marker. You can use the ***<R>*** key to rotate the markers.

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



8. Press ***<esc>*** to exit the placement mode.

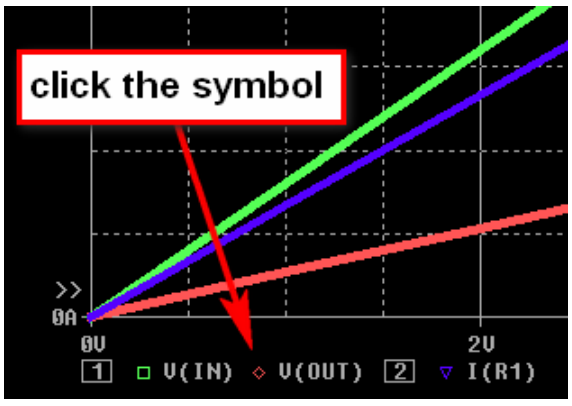
9. Return to the ***Probe*** window to view the graphic display of the traces you just added.

Customizing the Probe Display

You can customize the appearance of the Probe window to a great extent. Trace colors can be changed, grid spacing can be modified, trace symbols can be changed or displayed.

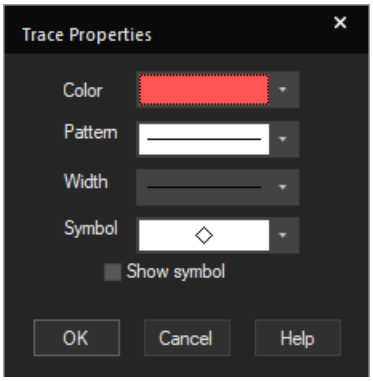
1. From the ***Probe*** window, click a trace using your right mouse button.

Note that you can also RMB on its symbol located below the X axis



2. From the pop-up menu that displays, choose the ***Trace Properties*** menu option.

The Trace Properties dialog box opens.



3. Change the color setting of the trace, the pattern of the line, the thickness of the line, the symbol denoting the trace, and whether or not to display the symbol on the trace itself.

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

Customizing the Default Settings

You can customize the PSpice window to have traces come up with only certain colors and at certain thicknesses.

1. From the ***Probe*** window, select **Tools > Options**

The Probe Settings dialog box opens.

2. On the ***General*** tab, change the '***Default Trace Width***' to 4 (legal values from 1-thinnest to 7-thickest)

3. Select the ***Color Settings*** tab and notice the ability to change the ***Background*** color from the default Black as well as the ***Foreground*** color and the list of available ***Trace Colors***.

Feel free to change these colors around to something that you like, if you want to go back to the stock colors, just select the ***Reset*** button at the bottom.

Create an Additional Y Axis

There are times when you can have a collection of traces displayed and it doesn't make sense to have them all on the same axis. In this case, you can create either an additional plot or an additional axis and move the desired traces to a new location.

1. In the ***Probe*** window, select ***Trace > Add Trace*** and add the trace ***I(R1)*** to the Trace Expression by single clicking on it in the Simulation Output Variables list.

Notice that the displayed current appears to be level at 0. It is, in fact, something larger than zero but when bunched on the same axis as the much larger voltage values, it doesn't show much variation. Let's create another y-axis and move the current trace over to it.

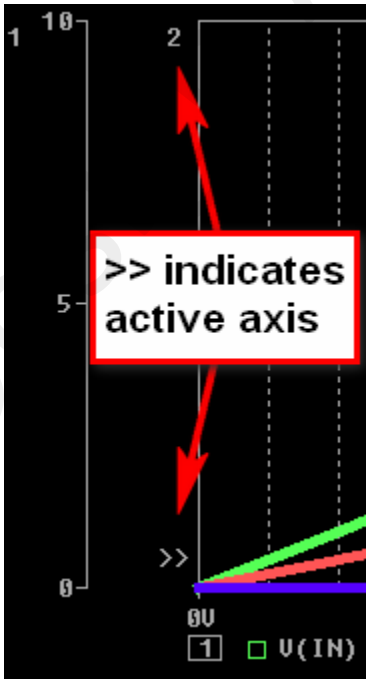
2. Choose ***Plot > Add Y-Axis*** to add an additional axis.

This makes an additional axis named '2' with nothing on it, you can see that the original axis is now named '1' and the trace names at the bottom are now to the right of the number '1' indicating that they belong to the primary axis.

3. A simple way to move the current trace over is to ***select the Trace Name*** of ***I(R1)*** – it will turn red.

4. Hit ***Ctrl+X*** on your keyboard to cut the trace out – it will be removed

5. ***Select the secondary axis*** by clicking on it, you should notice that the >> points to the secondary axis indicating that it is active

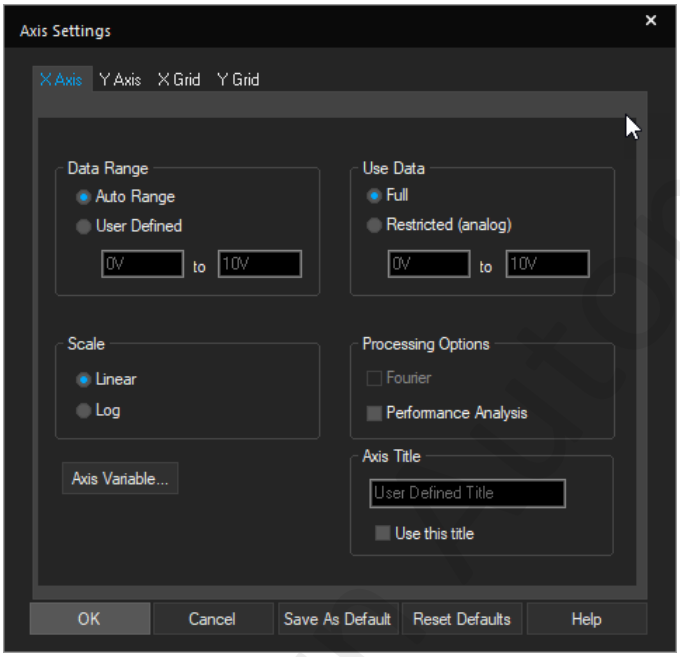


6. Hit ***Ctrl+V*** to paste the previously cut current trace onto the secondary axis

You will notice that the new trace goes right on top of the V(IN) trace, so it doesn't really show up. This is because the lines are at the same slope and the scales are set that they overlap perfectly. To fix this, we can modify an axis slightly so that both traces will be visible in the next section

Changing Axis and Grid Settings

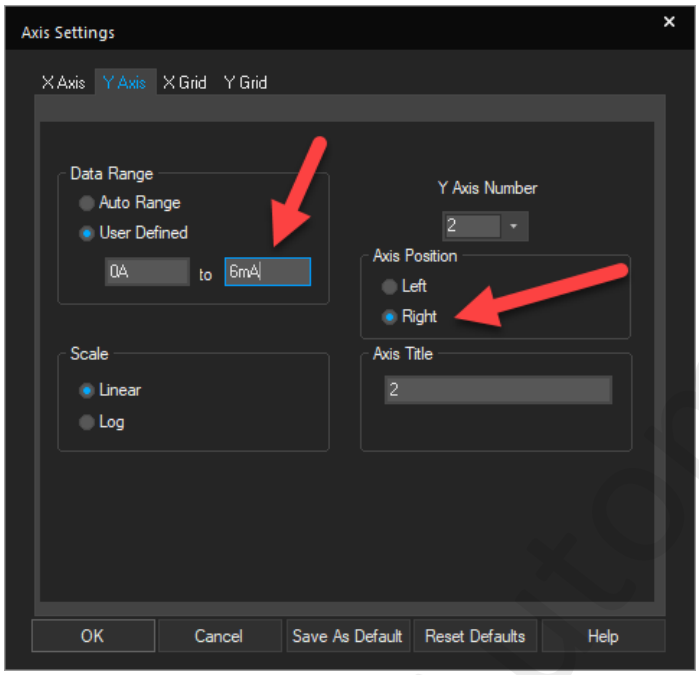
1. To edit axis and grid settings, right-click on any of the grid lines and choose ***Settings*** from the menu presented or, choose the menu commands ***Plot > Axis Settings***.



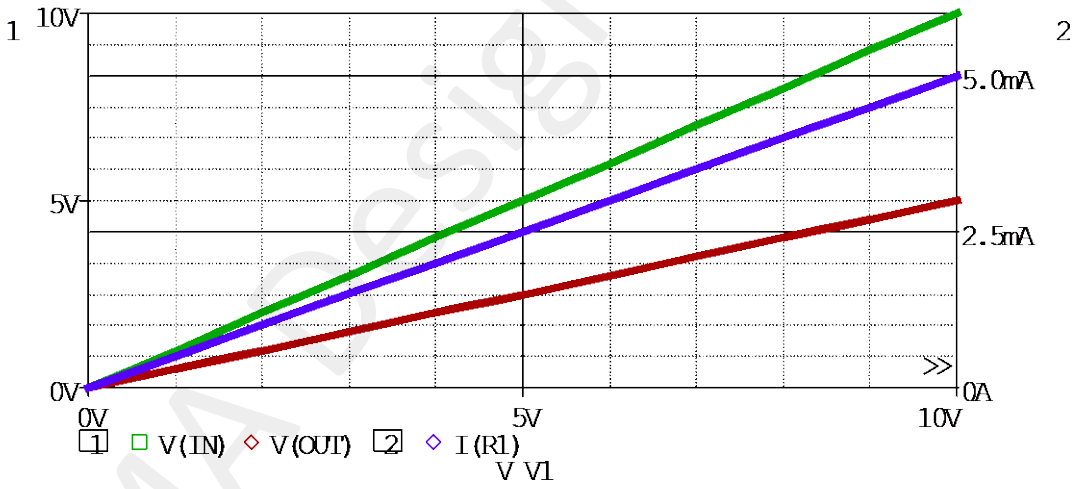
2. Select the ***Y Axis*** tab.

3. Under ***Data Range***, change it to ***User Defined*** leaving the min value and changing the ***max to 6mA***

4. In the ***Y Axis*** pulldown box, ***select axis 2*** and then click on the ***Right*** radio button to move axis 2 to the right edge of the plot.



This will be the result:



Optional Lab Exercise

- Delete the current trace I(R1) off of the secondary axis
- Delete the secondary axis by selecting it and choosing ***Plot > Delete Y Axis***
- Add a New Plot (***Plot > Add Plot to Window***) and re-add the I(R1) trace to it

Lesson 3, Lab 2: Creating Project Archive

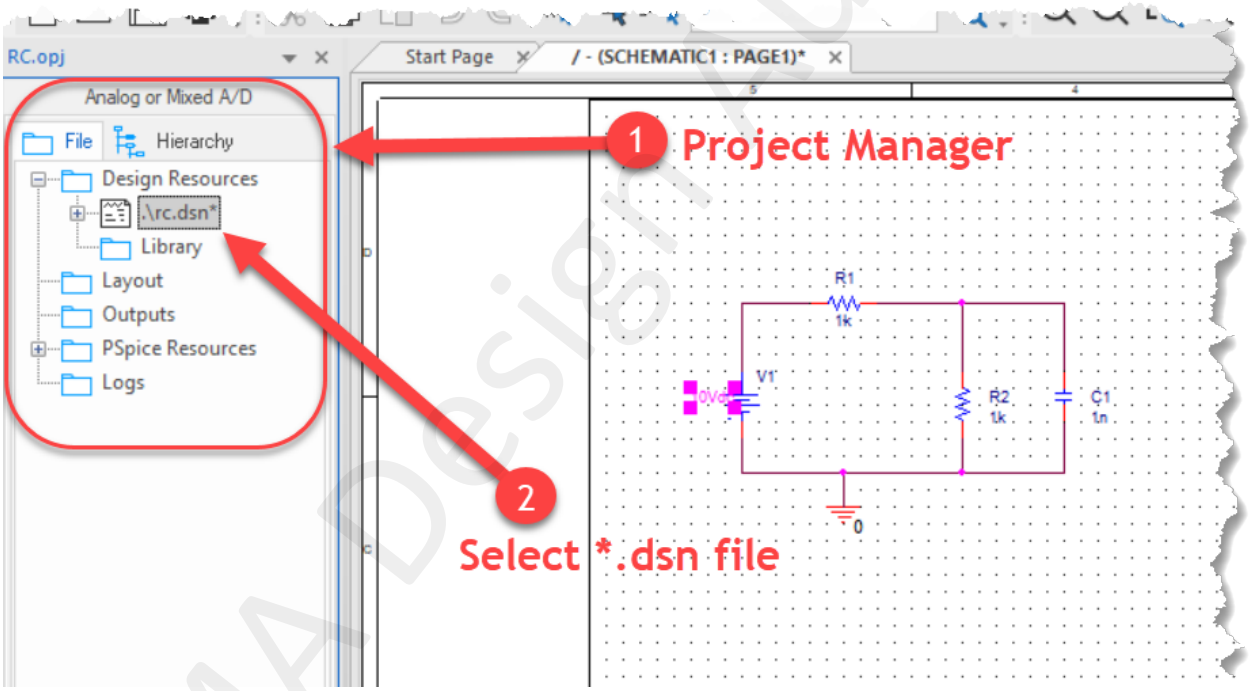
Lab Objectives

After completing this lab, you will be able to:

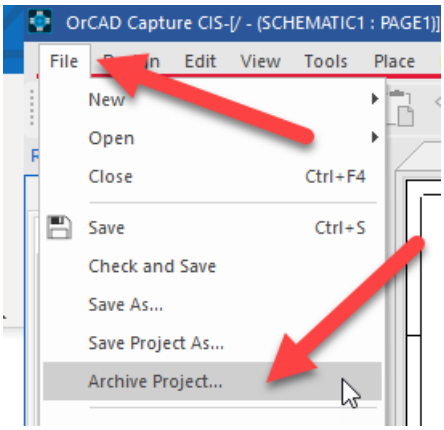
- Create a single .zip file that contains everything necessary to run your PSpice simulation

Invoke the Archive Command

1. In Capture, go to the **Project Manager** window
2. Select the ***.dsn** file

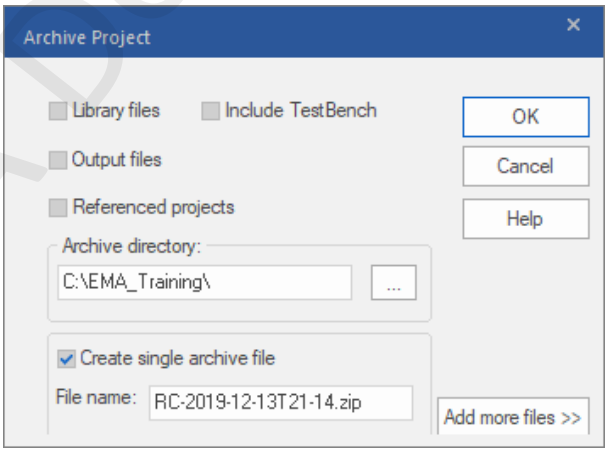


3. ***File > Archive Project…***



You have a number of options for files to include in the project archive.

- ***Library Files***: If you have custom parts in your project not from the default PSpice libraries and want to include them in your archive
- ***Output Files***: If you want to include the generated files that get created by PSpice during a simulation – these can be big files and can be regenerated with all the other files included in the archive, so their inclusion is not recommended
- ***Referenced Projects***: If your project uses hierarchy that points to external designs, you'll want to include them in the archive
- ***Include TestBench***: if you've made a TestBench for your simulation to run in and want to include it
- ***Archive Directory***: The directory where you'd like the archive to be created – note that it can **NOT** be in the same location as the present project
- ***Create Single Archive File***: If you'd like to put everything into a single zip file for easier management



4. **Browse** to an Archive Directory that is different from your present project directory, an example of a directory that you could make would be `C:\EMA_TRAINING\PSpiceEssentials\Archive`

5. **Check off** 'Create single archive file', Hit ***OK***

6. **Navigate** in Windows Explorer to the **Archive Directory** that you made in step 4

Notice the .zip file in there

7. **Open** up the .**zip file** to see the contents

This is a great way to send design files to another user because everything will behave exactly the same way on the unzipped PSpice simulation as it did on the original with no modifications needed at all. It includes all custom libraries (if used and checked) and all the simulation profiles with the configurations all working correctly after being unzipped.

8. **Close** the .**zip file** and **Windows Explorer**

Lesson 4. AC Sweep Analysis

What Does AC Sweep Analysis Do?

For an AC sweep analysis, PSpice A/D calculates the small-signal response of the circuit (linearized around the bias point) when sweeping one or more sources over a range of frequencies. Outputs include voltages and currents with the magnitude and phase, which you can use to obtain Bode plots.

AC Sweep is a linear small signal analysis. It does not simulate nonlinear effects like clipping. For example, if you put 1V into a circuit with a gain of 100, the output will be 100V.

What is Clipping?

In this example, you can see that the transient response will correctly take into account clipping. Clipping means that PSpice identifies that the Opamp only has 15V supply rails and can therefore not generate an output voltage larger than that, even if it would like to so it 'clips' itself at the voltage rail voltage.

AC analysis does things differently; it takes the small signal response and observes the behavior of the circuit. It then extrapolates that behavior at small signals and uses it for large signals as well which can lead to results that are not possible with the supplies in the design.

AC Sweep Simulation Setup

By default, an AC sweep analysis sweeps every AC source in the circuit through a specified range of frequencies. All AC sources are swept together. For example, if you have three AC sources in the circuit and are running a simulation from 10 Hz to 100 MHz, then all three sources are set at 10Hz at the same time and 100MHz. During an AC sweep, all design sources are always analyzed in unison.

To set up an AC sweep simulation, you need to configure your sources and define settings in the Simulation Settings dialog box.

Configuring Sources

AC Sweep analysis requires at least one AC source in the design. At least one source must have a value specified for the source's AC MAG or AC property. If there are multiple AC sources configured, then all of them are swept together through the same frequencies.

Each AC source symbol is assigned an AC Mag and AC Phase property. Magnitude is the peak value with the phase measured in degrees. If the phase is not specified, a zero value is assumed.

The older transient sources that we will see in future labs have a DC and an AC property on them so that the same symbol can be used for all the different analyses.

The DC property is used only for the DC Sweep analysis or as the default value for a transient analysis if no transient information had been provided.

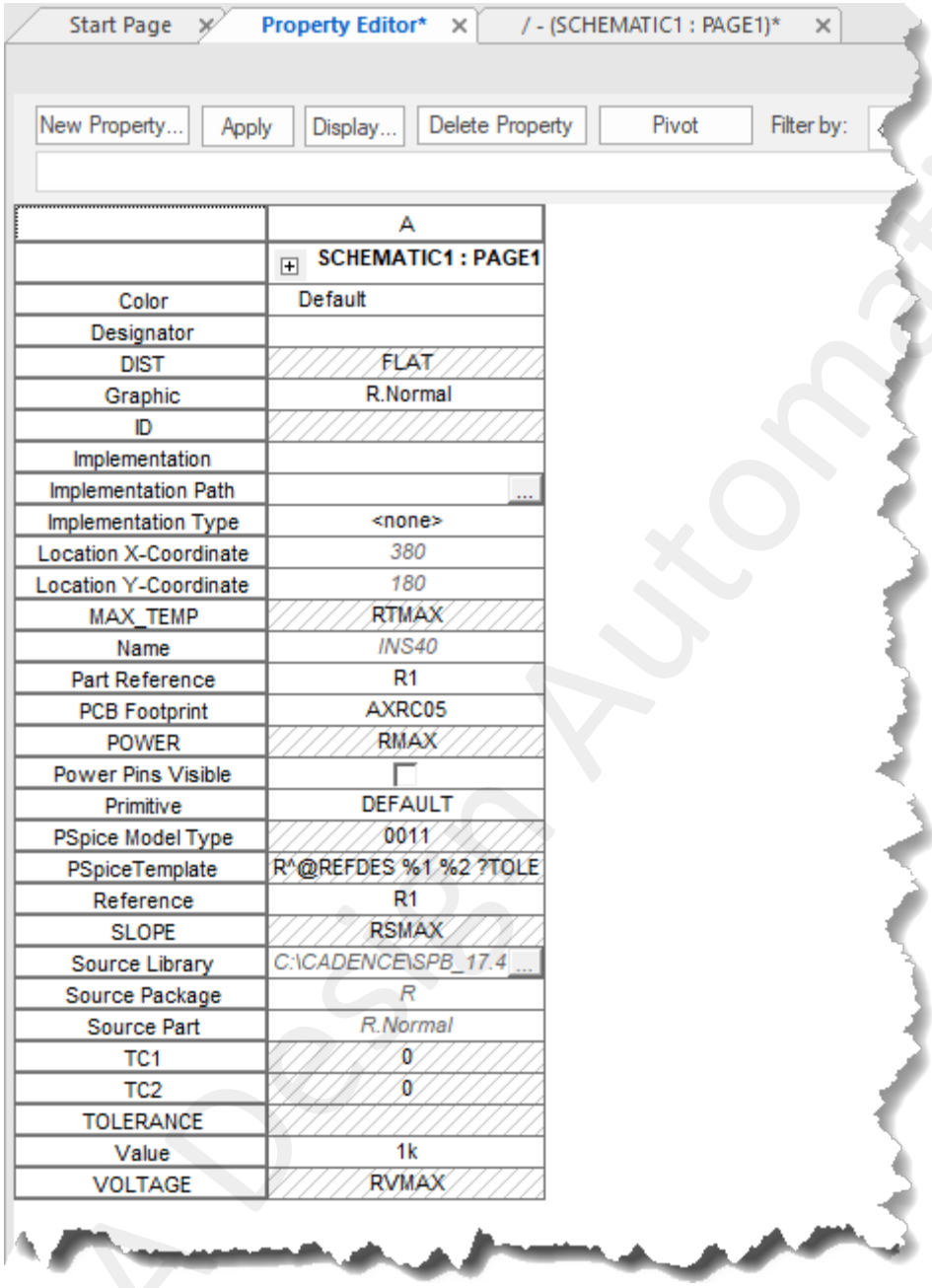
In the transient source symbols the AC property is for use only during an AC Sweep. The format for the property value is Magnitude and Phase with the values separated by a space. If no phase is specified, it is assumed to be 0 degrees.

Defining a Source

1. From the schematic, click the source to highlight the symbol.

2. Click your right mouse button to display the Property Editor Spreadsheet.

An example of the Property Editor is shown in the following graphic.



	A
	SCHEMATIC1 : PAGE1
Color	Default
Designator	
DIST	FLAT
Graphic	R.Normal
ID	
Implementation	
Implementation Path	
Implementation Type	<none>
Location X-Coordinate	380
Location Y-Coordinate	180
MAX_TEMP	RTMAX
Name	INS40
Part Reference	R1
PCB Footprint	AXRC05
POWER	RMAX
Power Pins Visible	
Primitive	DEFAULT
PSpice Model Type	0011
PSpiceTemplate	R^@REFDES %1 %2 ?TOLE
Reference	R1
SLOPE	RSMAX
Source Library	C:\CADENCE\SPB_17.4
Source Package	R
Source Part	R.Normal
TC1	0
TC2	0
TOLERANCE	
Value	1k
VOLTAGE	RVMAX

1. Select the ***ACMAG*** property from the list of properties.
2. Place your cursor within the ***ACMAG*** cell and modify the value as desired.
3. Also enter a value for AC voltage or current value for the source.
4. If applicable, enter a value for a phase shift.
5. Click ***OK*** to close the Property Editor

6. Verify the update from the schematic view.

An alternate way to edit the VAC value is to double click the VAC or IAC value and then modify the value in the Display Properties dialog box that opens.

Setting up the AC Sweep Simulation Profile

The AC sweep simulation analysis is configured through the Simulation Setting dialog box in much the same way as the bias point analysis and the DC sweep analysis introduced in Lessons 2 and 3.

When you setup the AC sweep profile, you can specify the analysis type as being either linear, octave, or decade. You also define the starting frequency, ending frequency, and the number of points in the range.

Defining settings for your AC sweep profile:

1. To open the Simulation Profile dialog box, select the ***PSpice > New Simulation Profile*** or ***PSpice > Edit Simulation Profile*** menu commands.

2. Click on the ***Analysis*** tab if it is not already active.

3. Choose ***AC Sweep/Noise*** as ***Analysis type***.

4. Select whether the sweep will be linear or logarithmic (octave or decade).

5. Enter the Starting and ending frequencies of the range to be swept.

6. Place the cursor in the ***Total Points*** field and type in the desired number of data points. For a linear sweep, the number of points will be equal to the number of frequencies swept. For logarithmic it will be the number of points per octave or decade.

7. Configure and verify the ***Data Collection*** and ***Probe Window*** settings.

8. Click ***OK***.

Notice that you do not have the choice of the variable that will be swept like you did in the setup for the DC Sweep Analysis.

To run the simulation (***PSpice > Run***,  icon or F11 key)

Trace Names

Examples and their meaning

II(R13)	Imaginary part of the current through R13
IGG(m3)	Group delay of M3's gate current
IR(VIN)	Real part if I (current) through VIN
IAG(T2)	Group delay of current at port A of T2
V(2,3)	Magnitude of complex voltage across nodes 2 & 3
VDB(R1)	Db magnitude of V across R1
VBEP(Q3)	Phase of base-emitter V at Q3
VM(2)	Magnitude of V at node 2

Probe Function and its Description

ABS(x)	|x|
SGN(x) +1	(if x>0), 0(if x=0), -1(if x<0)
SQRT(x)	x1/2
EXP(x)	e x
LOG(x)	ln(x)
LOG10(x)	log(x)
M(x)	magnitude of x
P(x)	phase of x (degrees)
R(x)	Real part of x
IMG(x)	Imaginary part of x
G(x)	Group delay of x (seconds)

PWR(x,y)	|x|y
SIN(x)	sin(x)
COS(x)	cos(x)
TAN(x)	tan(x)
ATAN(x)	tan-1
ARCTAN(x)	tan-1
d(x)	Derivative of x with respect to the x-axis variable
s(x)	Integral of x over the range of the x-axis variable**
AVG(x)	running average of x over the range of the x-axis variable
AVGX(x,d)	running average of x from X_axis_value(x)-d to X_axis_value(x)
RMS(x)	Running RMS average of x over the range of the x-axis variable
DB(x)	Magnitude in decibels of x
MIN(x)	Min. of the real part of x
MAX(x)	Max. of the real part of x

**In PSpice A/D, this function is called SDT(x).

Functions

You can add a complex trace using either the available suffixes or functions. Use of either method yields similar results. However, some Probe functions such as ABS(trace_name) do not have defined suffixes.

Lesson 4, Lab 1: Performing an AC Sweep Analysis

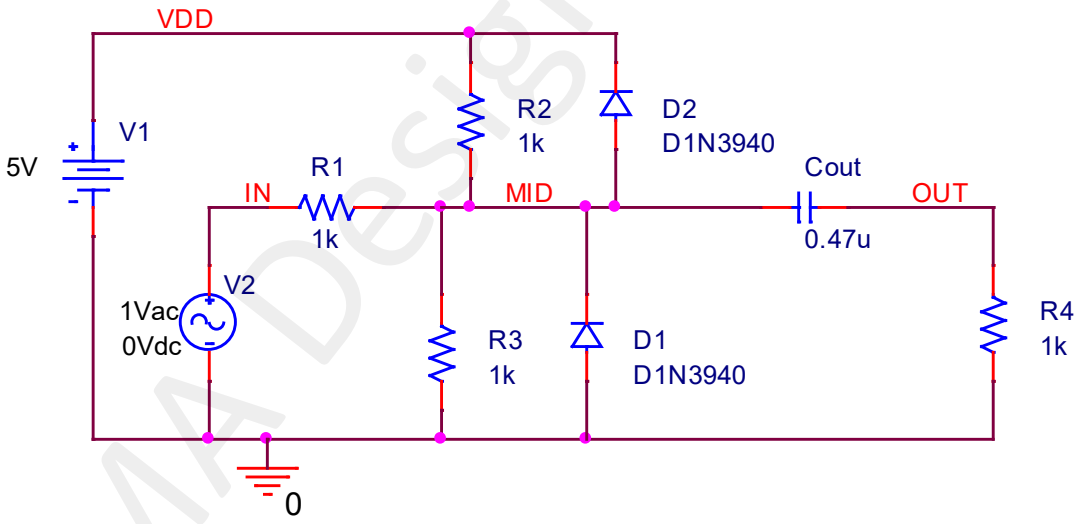
Lab Objectives

- Configure an AC Sweep Analysis
- Run the analysis
- Configure an AC source
- Examine the results in Probe
- Use complex trace variables

Creating a New PSpice A/D Project

1. Create a new blank PSpice A/D project called **clipper** and save it in the `C:\EMA_Training\PSpiceEssentials` directory

2. Draw the circuit as shown below. Use the following parts and symbols: **R**, **C**, **VDD**, **VAC**, **D1N3940**, and **0** ground



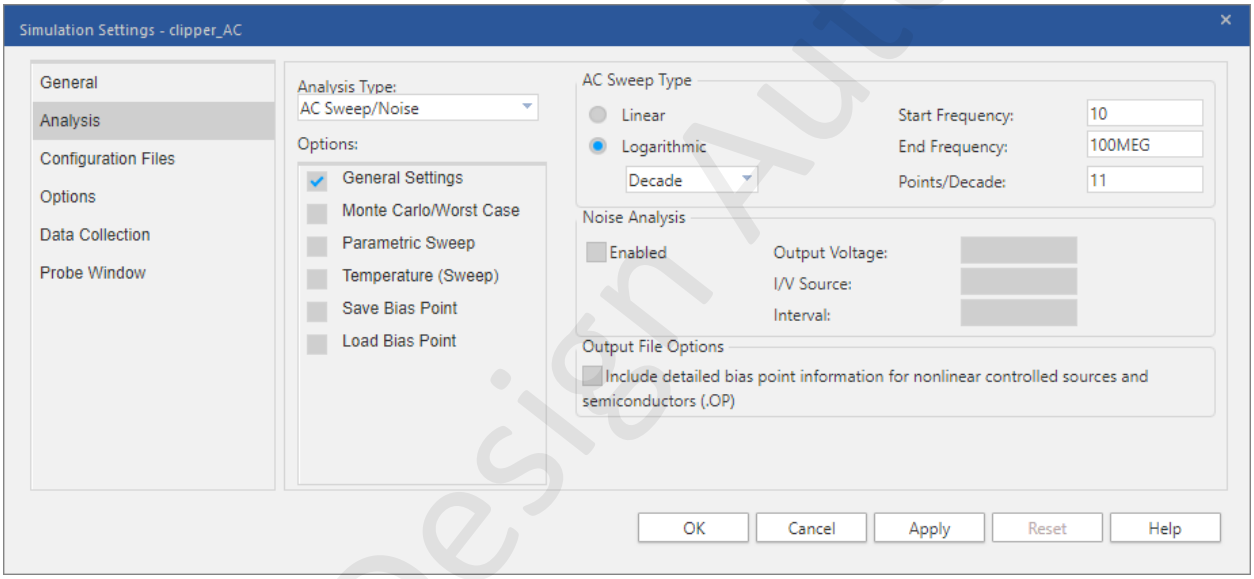
3. Change the reference designators as shown for ***COUT***. The other reference designators do not need to match the picture

Note that if you want a capacitor symbol with **pins** showing, take the C part from the ANALOG_P (the pin at the end means that the pins are showing) PSpice library instead of the ANALOG PSpice library.

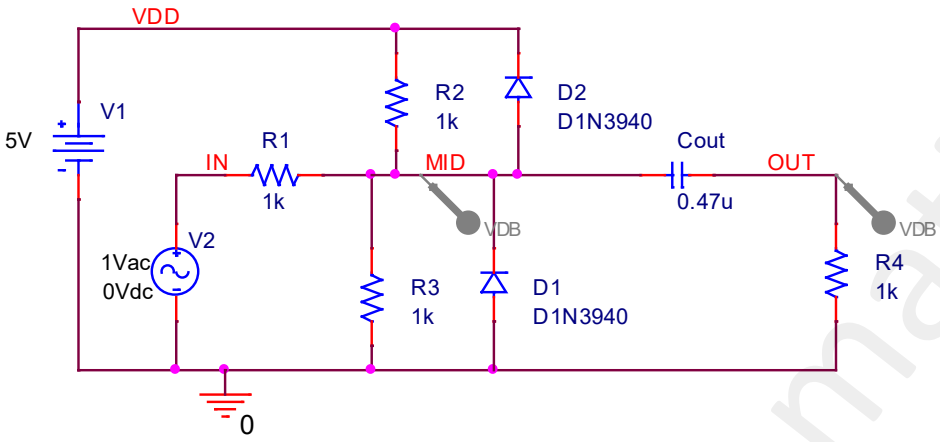
4. Edit the values of the DC ***source***
5. Change the value of the ***capacitor*** as needed
6. Add an alias to the nets for **VDD**, **IN**, **MID** and **OUT**.

Setting Up and Starting an AC Sweep Analysis

1. Create a new profile (***PSpice > New Simulation Profile*** or  icon) called ***ac***.
2. In the ***Analysis*** tab, select ***AC Sweep/Noise*** as the simulation type.
3. Set up the simulation parameters as seen in the graphic below.



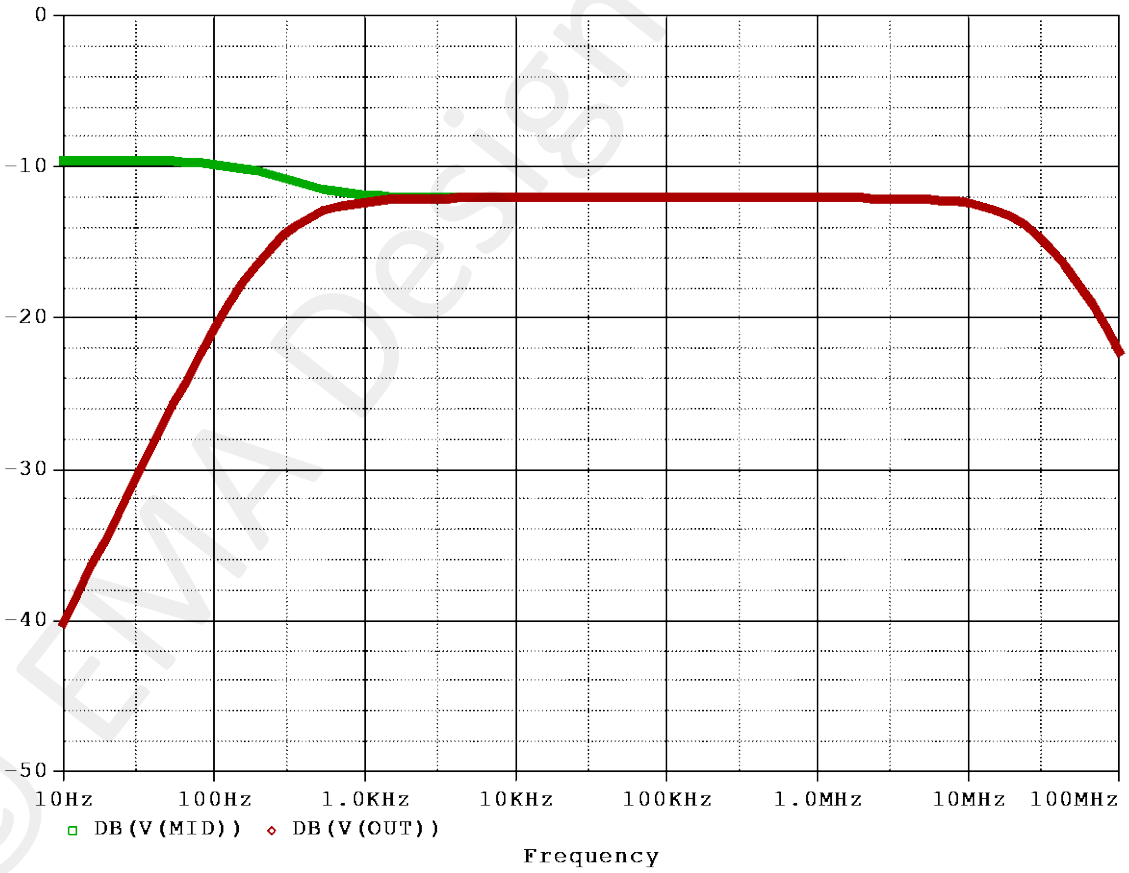
4. Click ***OK*** to save your simulation settings, close the Simulation Settings dialog box, and return to the schematic view.
5. Place VDB markers at **MID** and **OUT** by using ***PSpice > Markers > Advanced > dB Magnitude of Voltage***



Running the Simulation and Viewing Results

1. Choose ***PSpice > Run*** or click the ***Run PSpice*** toolbar button.

After the simulation has successfully completed, results automatically display within the Probe window.



Probe displays the dB magnitude (20log10) of the voltage at the marked nets, MID and OUT, as shown. VDB(MID) has a lowpass response due to the diode capacitance to ground. The output capacitance and the load resistor act as a highpass filter, so the overall response illustrated by VDB(OUT), is a bandpass response. Because the AC Sweep is a linear analysis and the input voltage was set to 1V, the output voltage equals the gain (or the attenuation) of the circuit.

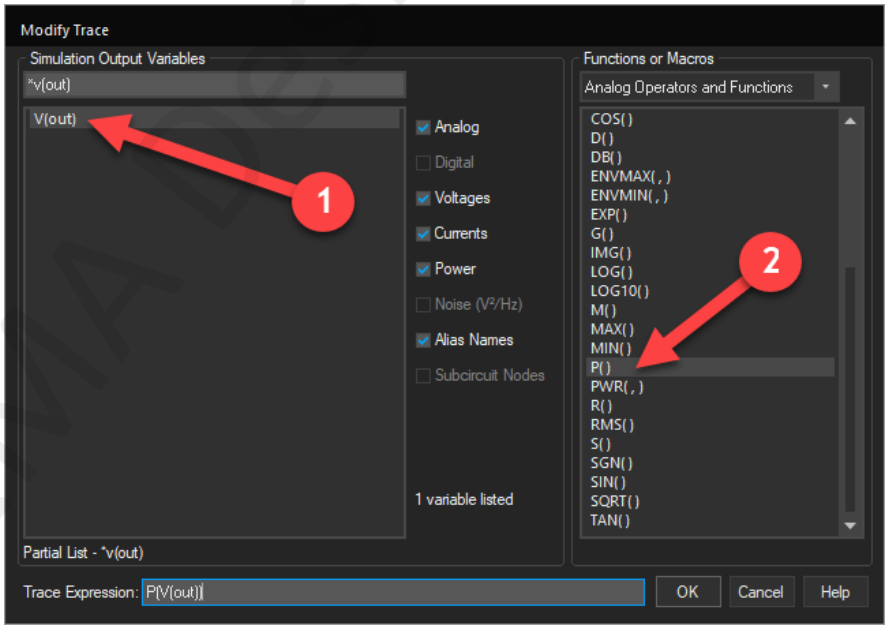
Displaying a Bode Plot

1. Click on the name of the trace ***DB(V(MID)),*** hold the ***shift*** key and subsequently select the **DB(V(OUT))** trace located below the X axis of the Probe window.
2. Press ***<Delete>*** to remove both traces.
3. Choose the ***Plot > Add Plot to Window*** menu item.
4. Go back to Capture and double click on the grayed VDB marker on the ***OUT*** node, the corresponding ***DB(V(OUT))*** waveform should appear on the top plot in Probe
5. In Probe, select the bottom plot by clicking on it

The SEL>> should appear at the lower left of the lower plot

6. Choose the ***Trace > Add*** menu, use the ***Add Trace*** icon  or RMB and select ***Add Trace.***

The Add Traces dialog box opens.

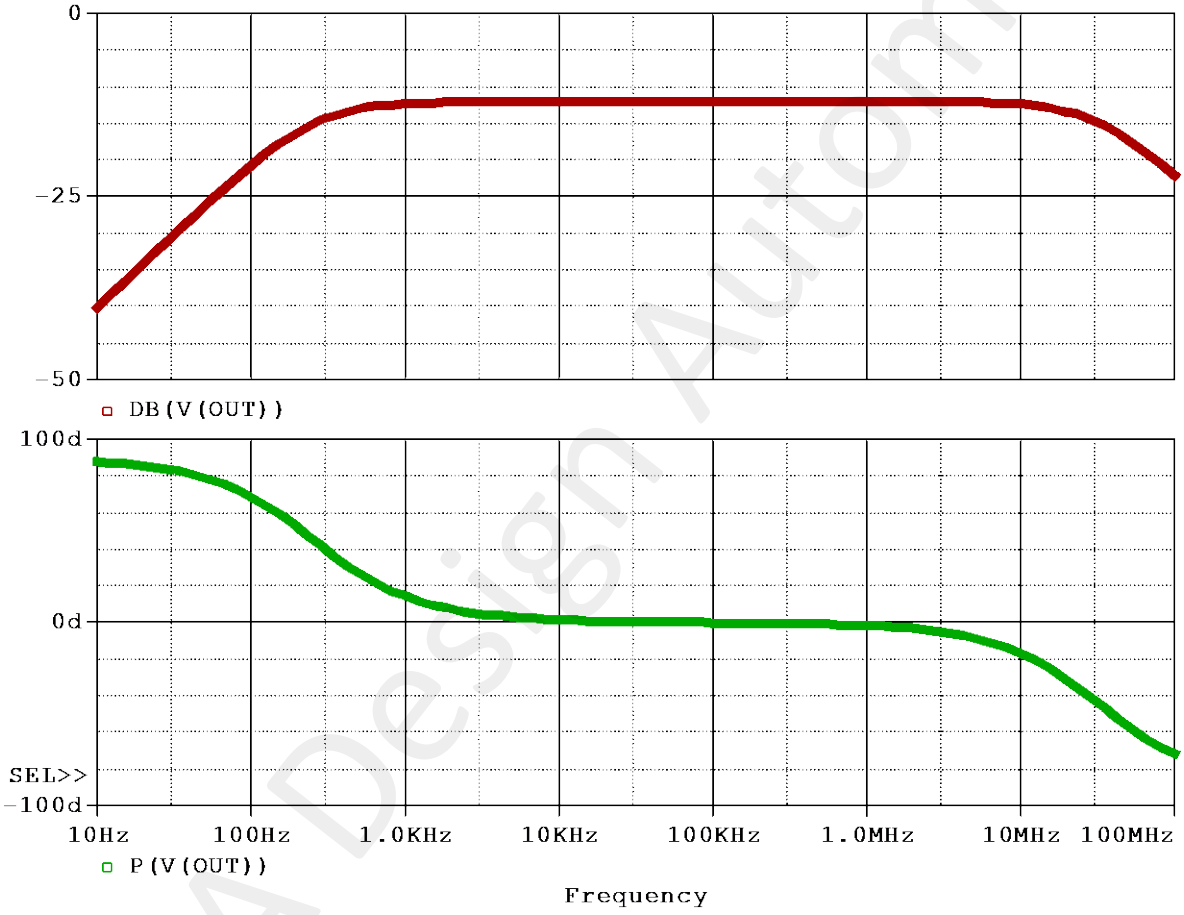


7. On the right hand side, select ***P()*** to indicate that you wish to measure the Phase of the Variable you'll select next.

8. Select ***V(OUT)*** from the list of simulation output variables. Use the vertical scroll bar if necessary (you can also search this list using the the top left search bar)

9. Notice in the Trace Expression at the bottom that you will have ***P(V(OUT))***

10. Click ***OK*** or Press ***<ENTER>***



This shows us the BODE plot for the OUT node where the gain in dB is shown on the top plot and the phase (in degrees) of the OUT node is shown on the bottom plot

Lesson 4, Lab 2: Using Alternate Connections

Lab Objectives

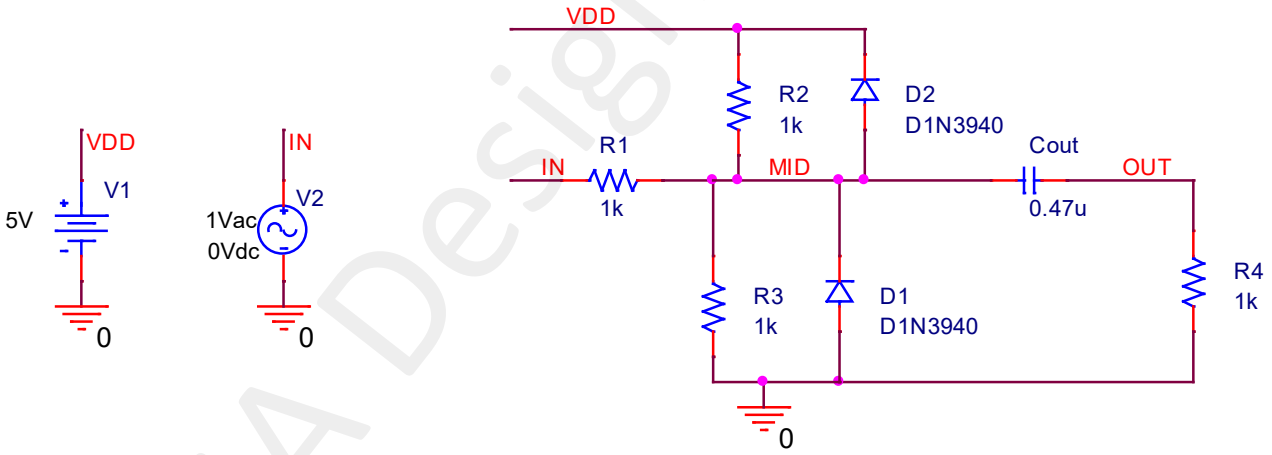
- Discover some alternative ways of making connections in the schematic

Modify the Schematic

There are a variety of ways to make connections in Capture aside from direct connections. Sometimes it makes the schematic a little cleaner and less busy when the sources are tucked away in a different area of the schematic page. On a single page design, you can do that very easily by simply naming the nets the same.

1. Modify your design to pull the voltage sources away from the schematic as seen in this picture.

> Selecting and dragging a component while holding down the Alt key is a good way of ripping up the circuitry and not having the wired connections keep rubberbanding to stay connected.



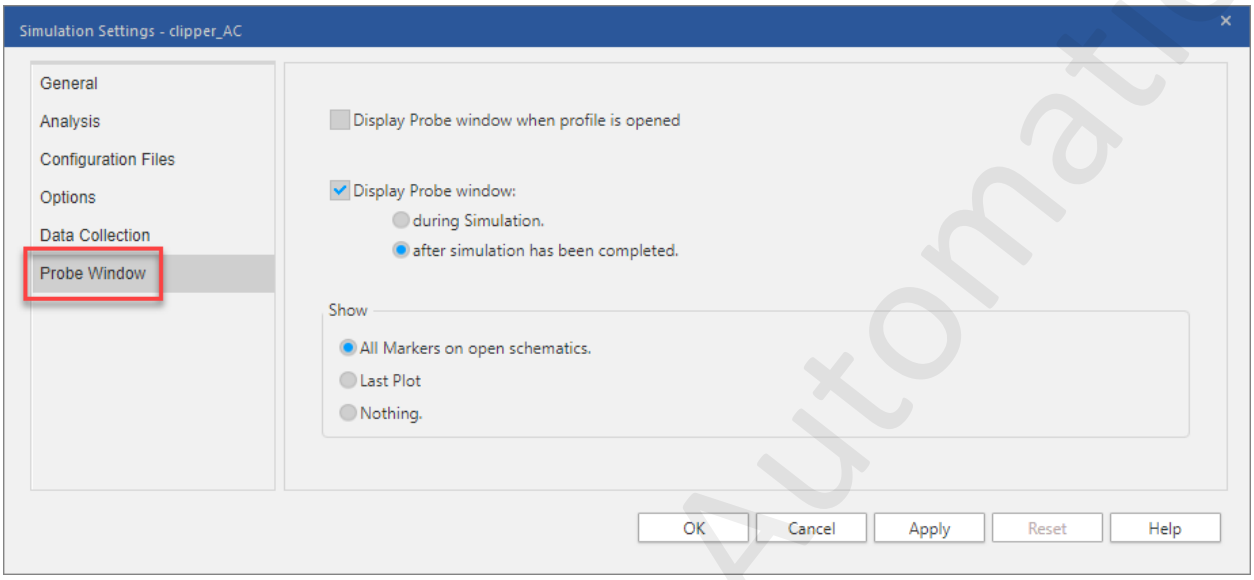
2. Make sure to have the nets named the same thing (VDD and IN) on both ends where they should be wirelessly connected

Saving the Plot Formatting

Before running the simulation, we want to keep the result formatting that we had done previously so we don't have to setup all the waveforms and plots again.

1. To keep the formatting of the last simulation, edit the PSpice simulation profile (***PSpice > Edit Simulation Profile***)

2. Go to the Probe Window tab at the end and in the Show section, choose **Last plot**

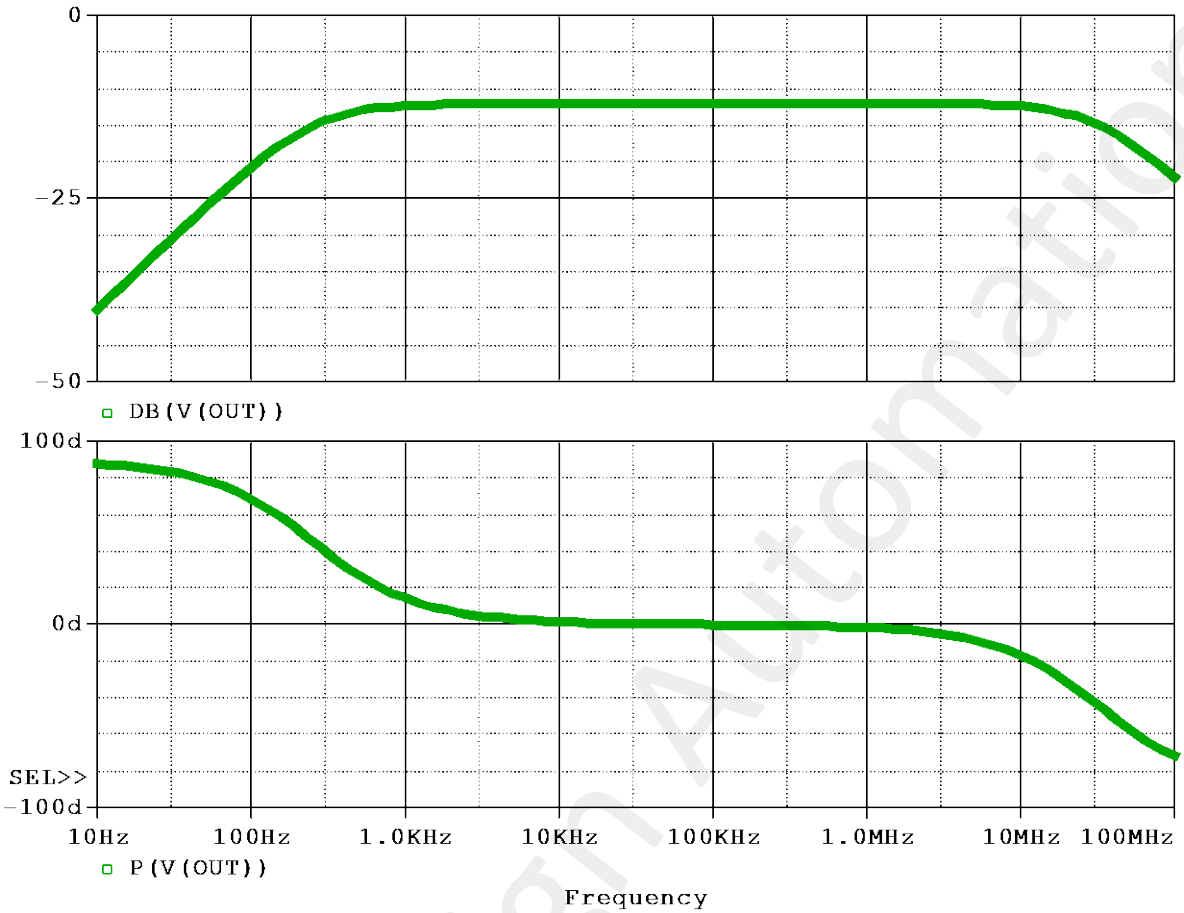


This will keep the simulation display in Probe the same as it was the last time it was run

3. Hit **OK**

4. Choose ***PSpice > Run*** or click the ***Run PSpice*** toolbar button.

The Probe window should open and display the same waveforms that you had previously.

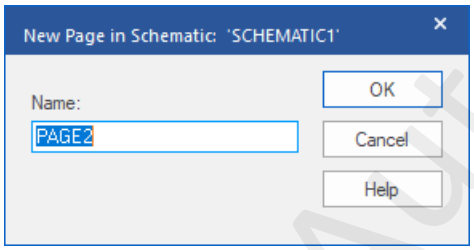


Lesson 4, Lab 3: Optional: Create Additional Page

We want to learn about using Off Page Connectors in this exercise. Off Page Connectors give no real benefit in a single page design like we have here but if you need to branch out and use additional pages, Off Page Connectors become necessary.

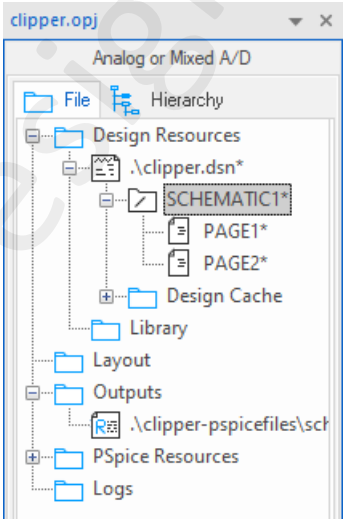
1. In Capture, go to the Project Manager and **right mouse button** on the SCHEMATIC1 folder then choose **New Page**

2. Call the page **PAGE2** (which is the default)

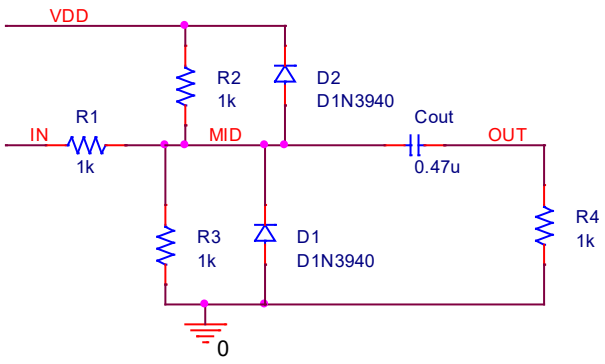


3. Hit **OK**

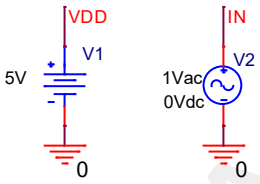
You will now have 2 pages under the SCHEMATIC1 folder of your design, like this:



4. On PAGE1 **cut out** (Ctrl+X) the **voltage sources** and **paste** them anywhere on PAGE2 (Ctrl+V) so page 1 now looks like this:



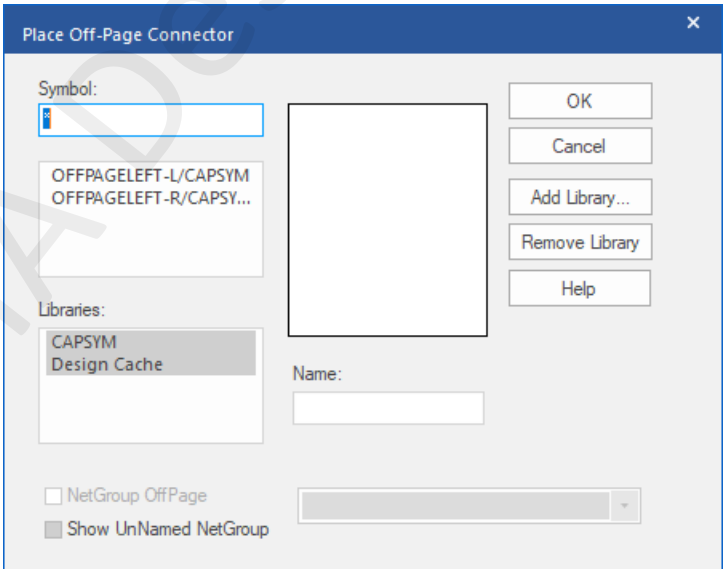
And PAGE2 looks like this:



Add Off Page Connectors

Add some off page connectors on both PAGE1 and PAGE2

1. In Capture, go to ***Place > Off Page Connector…*** or select the  icon to open the Place Off-Page Connector dialog



2. If no libraries besides Design Cache are present in the lower left corner, you can add one by selecting the **Add Library…** button on the right side and browsing to the **capsym** library in the **C:\Apps\SPB_17.4\tools\capture\library** directory

3. Once the capsym library is configured, you can double click on either the **OFFPAGELEFT-L** or the **OFFPAGELEFT-R** symbols to place them on your schematic.

The difference between the two symbols is only graphical, they will both function exactly the same. OFFPAGELEFT-L has the connection on the pointy end of the double arrows, OFFPAGELEFT-R has the connection on the other side of the double arrows, like this:

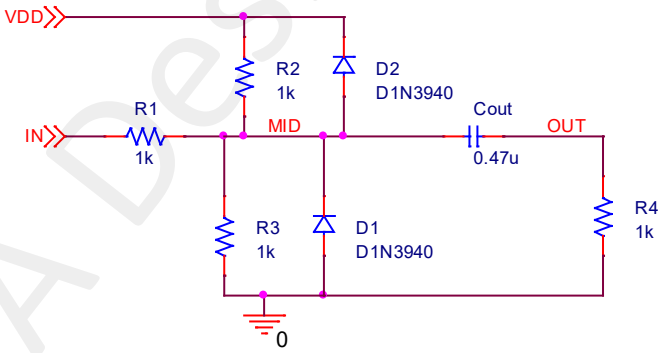
4. Place 2 Off Page Connectors of your choice on both PAGE1 and PAGE2

5. Rename them on each page to VDD and IN by double clicking on the name and typing in the new name

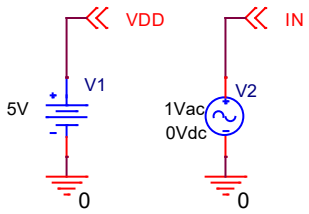
6. Connect them up to the nets that they should be connected to

7. Delete the name of the net that used to define the net name

The Schematics should now look like this, PAGE1:



And PAGE2:



1. In Capture, choose ***PSpice > Run*** or click the ***Run PSpice*** toolbar button. 

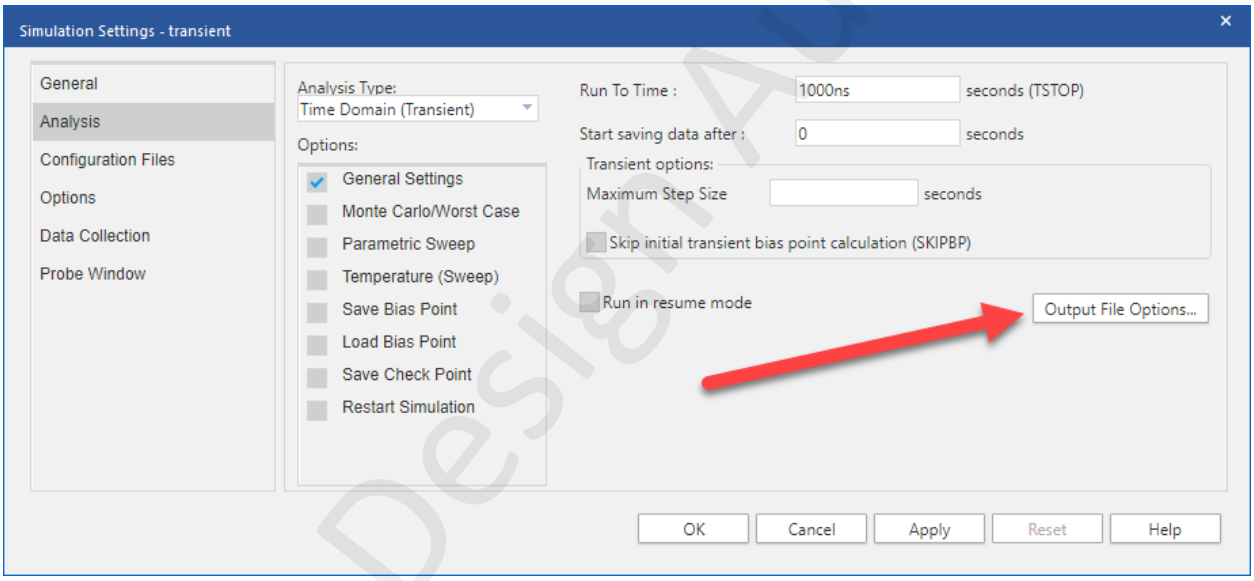
Verify that the results are the same as they were previously even though your sources are now on another page.

Lesson 5. Transient Analysis

What is Transient Analysis?

Transient analysis is a time domain simulation. It measures circuit response in the time domain. Results viewed in Probe resemble the waveform when probing the circuit with an oscilloscope or logic analyzer. You can perform transient analysis on both analog and digital circuits. Time is the swept variable on the x-axis while voltage and/or current is on the y-axis.

Setting up a Simulation



The transient analysis dialog box contains several different options. Two windows are used to set up the transient analysis. The first window shown in the graphic above establishes settings for the Run to time, Start saving data after, and the Maximum step size.

The list below explains dialog box options:

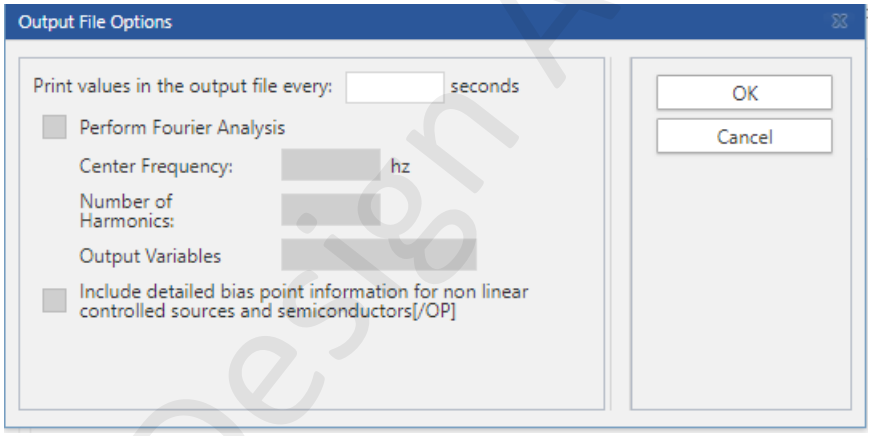
Run to Time - Every transient simulation begins at time = 0 and runs to time = TSTOP. It is not possible to stop a simulation, change the circuit, and continue the simulation from the time at which it was paused. You can pause a simulation, open the Probe window to look at the data and continue the simulation from the time at which it was paused.

Start saving data after - This was formerly known as the No Print Delay. This option allows you to specify a time period, starting at TIME=0 (the beginning of the simulation), during which no data will be saved. This is useful when your circuit has a long settling time and you wish to save data only after the circuit has reached steady state.

Maximum Step Size - This was formerly known as the Step Ceiling. This option is used to limit the internal step size that the simulator may take. It is very useful for solving many convergence problems. However, setting a maximum step size that is too small can cause a simulation to take much longer to simulate and can produce an enormous data file. So, use it with some care.

Skip initial transient bias point calculation - This causes the simulation to not calculate a transient bias solution. This is sometimes used when you have a circuit that is bi-stable with more than one possible initial solution, as in the case of an oscillator.

The second dialog box used to set up a transient analysis is the Transient **Output File Options** dialog box, which includes the Print Delay, Fourier, and the Detailed Bias Point configuration.



There are additional output file settings available for transient analysis:

Print Values to Output File - This option determines how often data is written to the output file. It is used in conjunction with the print and plot symbols. It does not affect how the data is saved to the Probe data file. This option is used only for cases where you want a tabular output in the output file. It has not been used with any great frequency since Probe was introduced and no longer requires a default value to run a simulation.

Perform Fourier Analysis - This will enable Fourier analysis. This option is not used too often since Probe has a built in FFT function.

Include detailed bias… - This will print to the output file a detailed report of the transient bias solution. The report will have the same information as a .OP statement.

A note on bias point calculation: Normally there is a bias point calculation performed separately from any other simulation. This is the .OP command. Another bias calculation is performed at the beginning of the transient analysis. The difference between the two is that the transient bias uses the TIME = 0 value for the source rather than the DC value (a single source can have an AC, DC, and a transient component). For the .OP bias calculation, the DC value is used. For the transient bias solution, the value of the source will be whatever that source is at TIME = 0.

Scheduling Option Values

YYou can schedule runtime settings. This allows you to schedule the values of most of the transient tolerances and the maximum transient step size. The advantage of scheduling these options is that you no longer are forced to use the same value for an option throughout the entire simulation.

In certain situations, you may want to predefine a set of values for a parameter and schedule these values to take effect at various time intervals during a long simulation. For instance, you may want to use a smaller time step value during periods where the input stimulus changes rapidly, but otherwise use a larger value.

The following runtime parameters can be changed at scheduled times during a simulation. Note that these only apply to transient analysis; you cannot interact with other analysis types.

- RELTOL
- ABSTOL
- VNTOL
- GMIN
- ITL4
- Maximum Step Size

You can schedule RELTOL, ABSTOL, VNTOL, GMIN, and ITL4 parameter changes by entering them either in the Simulation Profile or in a text file using the new expression SCHEDULE, and then including that file in the simulation profile settings.

The expression SCHEDULE is a piecewise constant function (from time x forward use y) and takes the form:

```
{SCHEDULE(x1,y1,x2,y2…xn,yn)}
```

where x is the time value, which must be > 0, and y is the value of the associated parameter. You must include an entry for time=0.

When used with the .OPTION command, the syntax is as follows:

```
.OPTIONS <Parameter Name>={SCHEDULE(<time-value>, <parameter
value>, <time-value>, <parameter value>, …)}
```

For example:

```
.OPTIONS RELTOL={SCHEDULE(0s,.001,2s,.005)}
```

For example:

```
.OPTIONS RELTOL={SCHEDULE(0s,.001,2s,.005)}
```

This example indicates that RELTOL should have a value of 0.001 from time 0 up to time 2s, and a value of 0.005 from time 2s and beyond (that is: RELTOL=.001 for t, where $0 \leq t < 2s$, and RELTOL=.005 for t, where t / 2s).

The next example changes RELTOL from .001 to .01 between 1ms and 1.5ms. The SCHEDULE command is added in place of a value for the parameter.

```
{SCHEDULE (0, .001, 1m, .01, 1.5m, .001)}
```

The SCHEDULE command may be used in a text file as:

```
.OPTIONS parameter_name = {SCHEDULE (t1, V1, t2, v2...tn, vn)}
```

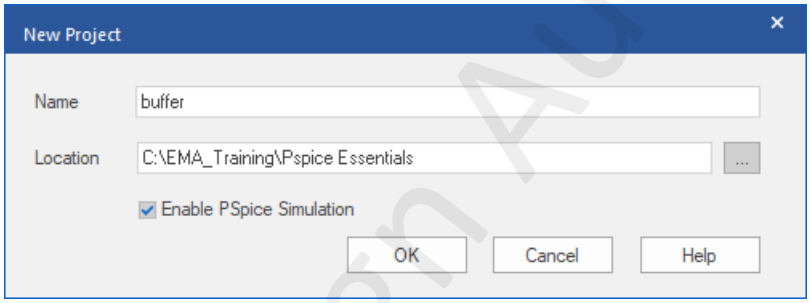
Lesson 5, Lab 1: Transient Analysis on buffer.dsn

Lab Objectives

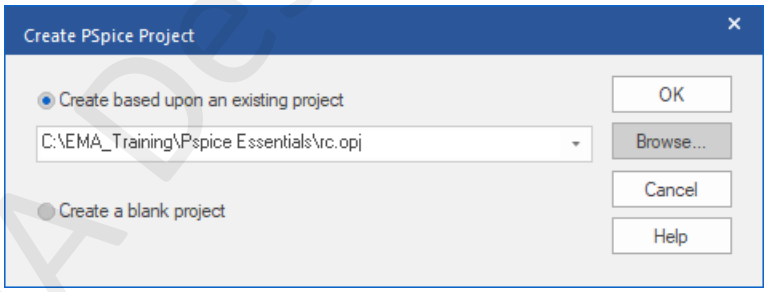
- After completing this lab, you will be able to run a transient analysis on the design `buffer.dsn`.

Creating a New Project Based on an Existing one

1. Create a new project, `buffer.opj` in the `C:\EMA_Training\PSpiceEssentials\Designs` location and **check the box** labeled **"Enable Pspice Simulation"**



2. Click on the ***OK*** button in the ***New Project*** dialog to display the ***Create PSpice Project*** dialog



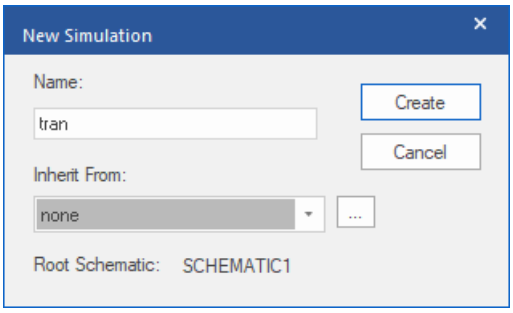
3. Click ***Browse*** and navigate to your `RC.opj` project from lesson 1

4. Hit `OK`

Create a New Simulation Profile

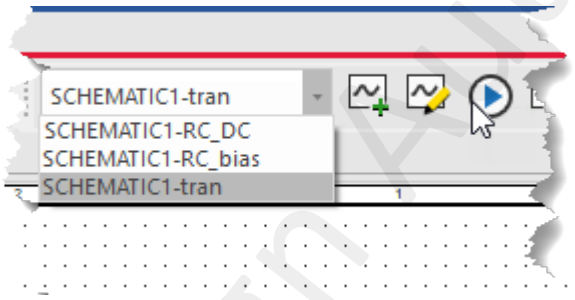
1. In Capture, Create a new profile (***PSpice > New Simulation Profile*** or  icon)

2. In the name field, give it the name `tran`



3. Hit `Create` on the created New Simulation window

4. Notice that from the Active Profile window in the upper left, we have some choices of profile



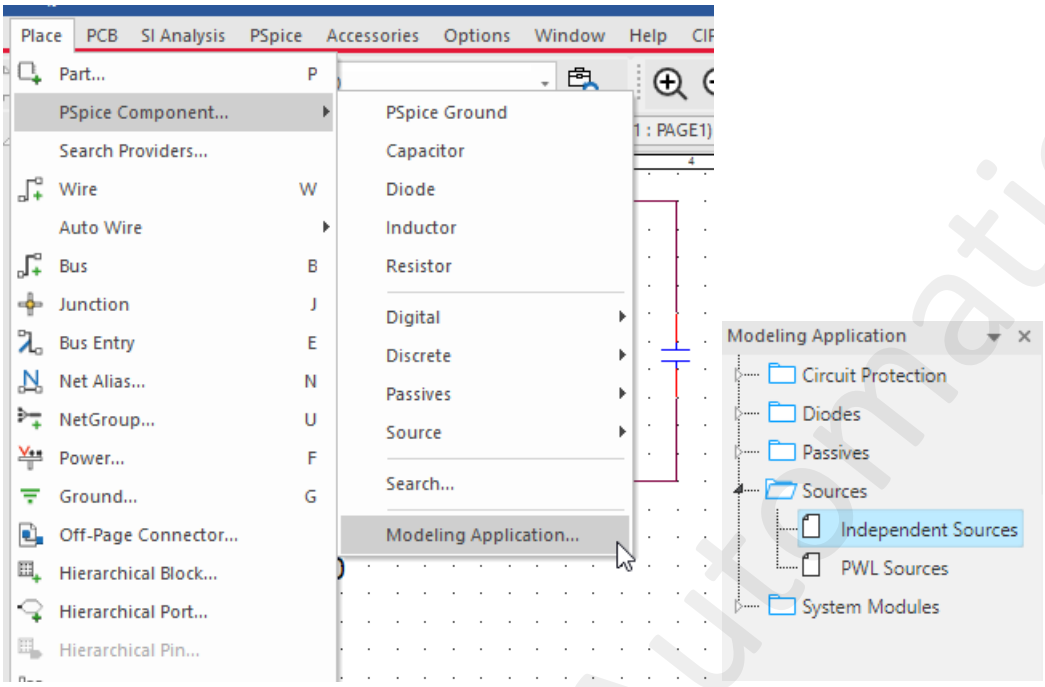
5. Leave tran selected

Add new Pulse Stimulus

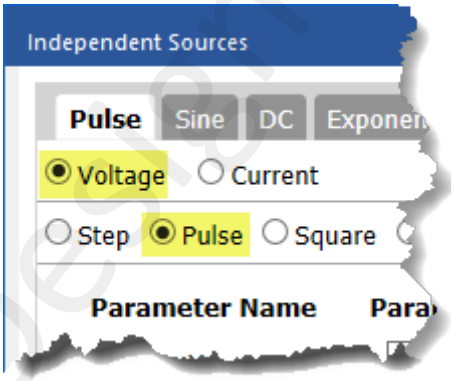
1. In the schematic, delete the DC source

2. Activate the modeling application by hitting the icon  or choose *Place > PSpice Component… > Modeling Application…*

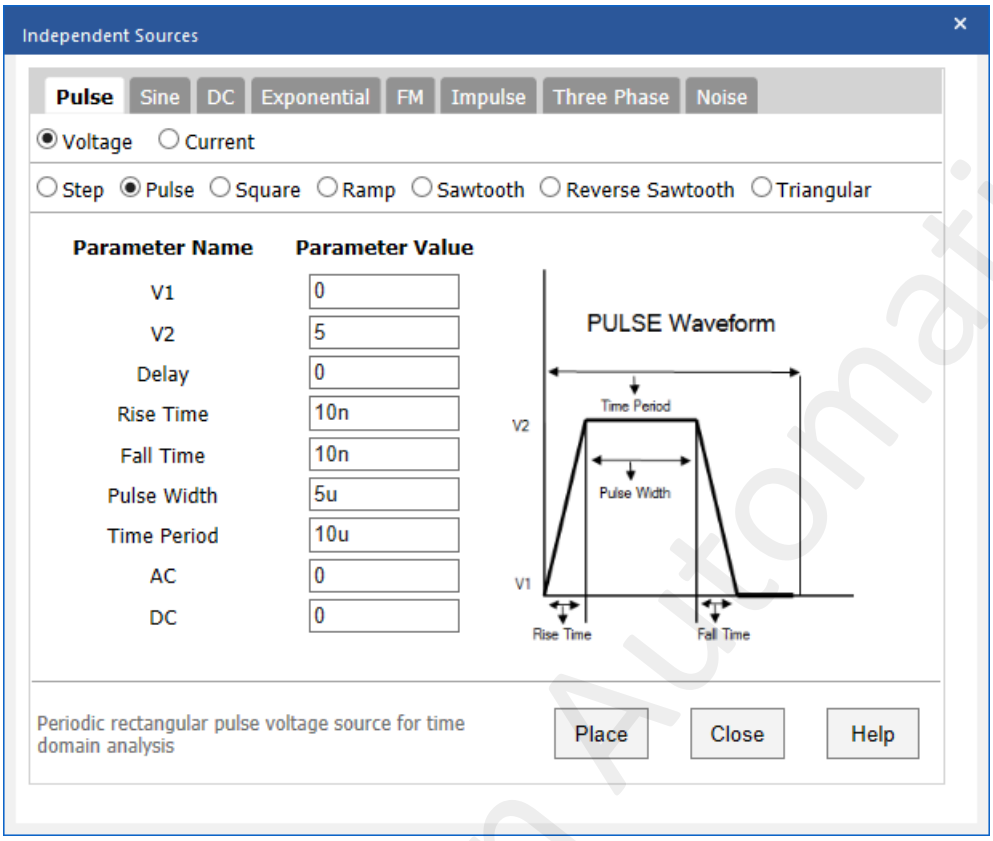
3. On the right side of the screen the modeling application window will open. Choose *sources > Independent Sources*



4. On the Pulse tab, leave the first radio selection as Voltage and change the second one to Pulse



5. Enter the values as shown in the graphic that follows

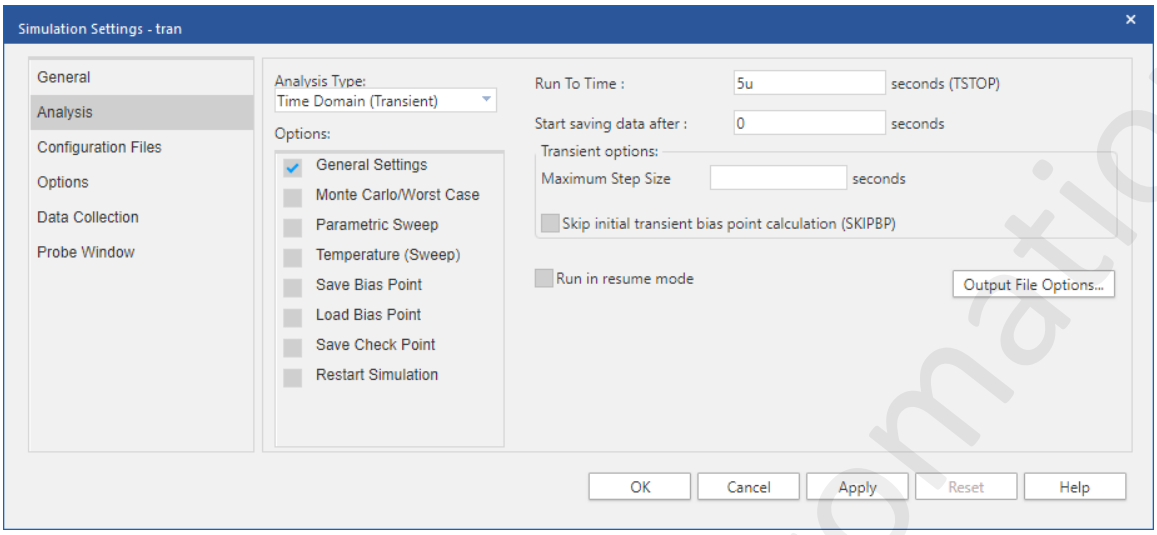


6. Choose Place and drop the symbol on your schematic

Configure the Simulation Profile

On the schematic page…

1. Use *PSpice > Edit Simulation Profile* or the  icon to edit the ***tran*** simulation profile to have a transient analysis with a final time of ***5u***



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

3. Run the simulation (***PSpice > Run***, icon or F11 key)

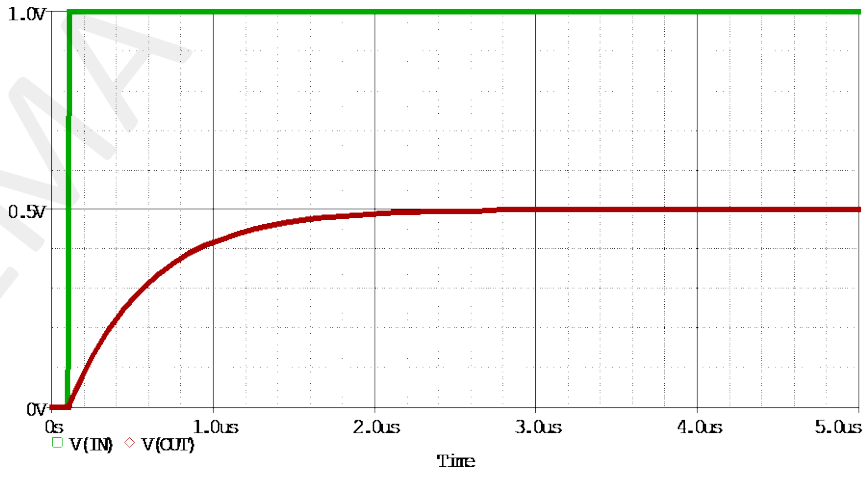
4. If you do not already have voltage markers on the IN and OUT nets, go back to the schematic and add them now.

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.

5. View the results of the simulation in the Probe window.

6. Save the design file

The waveforms illustrated in the next graphic shows how the voltage across the parallel resistor and the capacitor components ramps up to its steady state value.



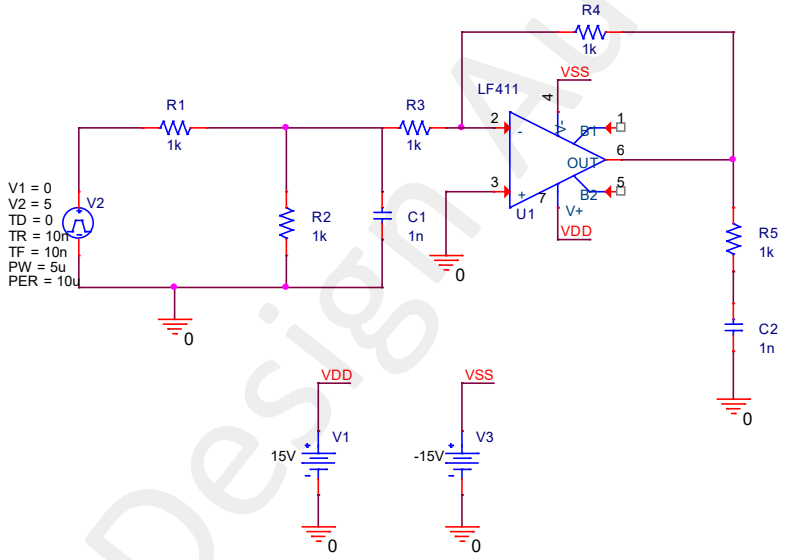
Lesson 5, Lab 2: Transient on BUFFER.DSN

Lab Objectives

- After completing this lab, you will be able to run a transient analysis on the design `Buffer.dsn`.

Editing the Design File

1. Continue using `Buffer.dsn` from the previous lab and add the opamp, resistors and voltage sources so that it looks like this. There are some notes below to aid you should you require assistance finding the parts or symbols.



The additional parts needed are **LF411** (there are a few to choose from, use the one from the `opamp` library), and **VDC**. You will need to edit the value of some reference designators and component values.

You will need to mirror the LF411 vertically to have the – pin on the top, select the part and hit the **V** key to do this

Delete the node name OUT next to CAP1

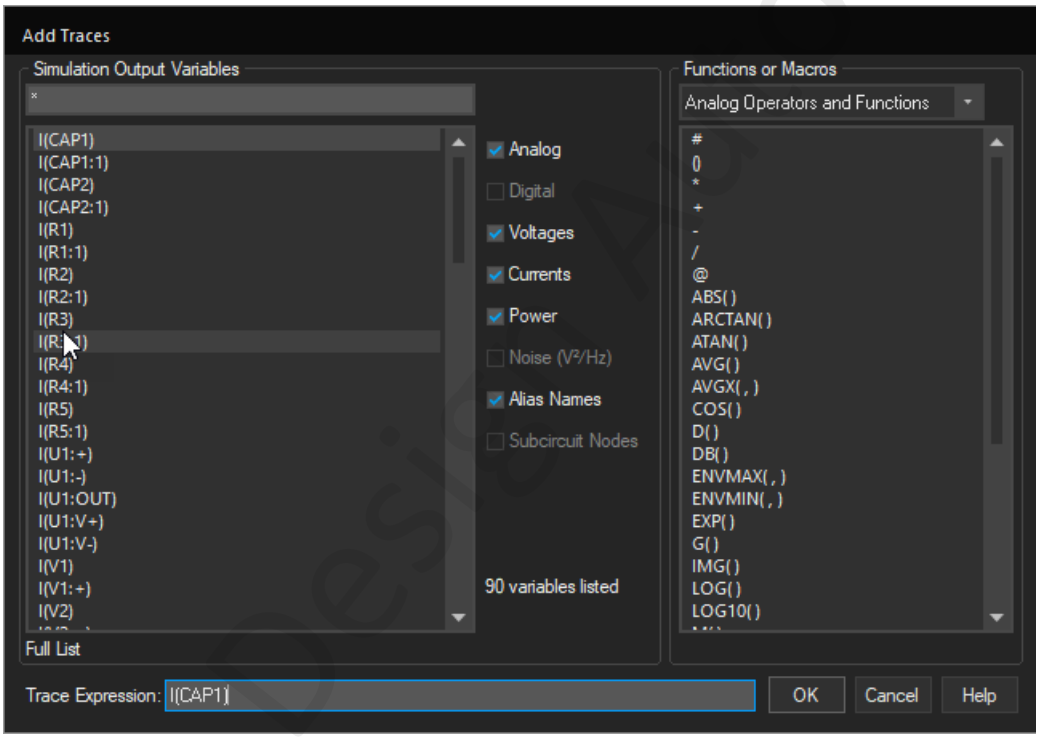
The VDD and VSS are connected from the DC Power source to the Opamp power pins by virtue of the nets having the same name and being on the same schematic page; that should be all that's needed to make the connection.

Verifying Simulation Set Up

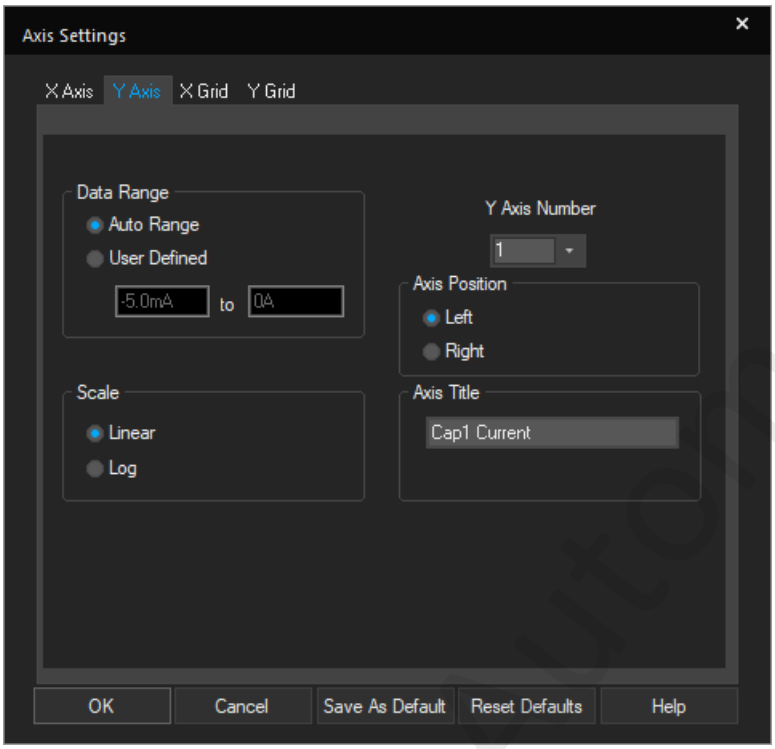
1. Verify settings for the transient analysis profile
2. Run the simulation (***PSpice > Run***, icon or F11 key)

Plotting the Y Axis

1. From the Probe window, click the ***Add Trace*** toolbar icon to open the ***Add Traces*** window.

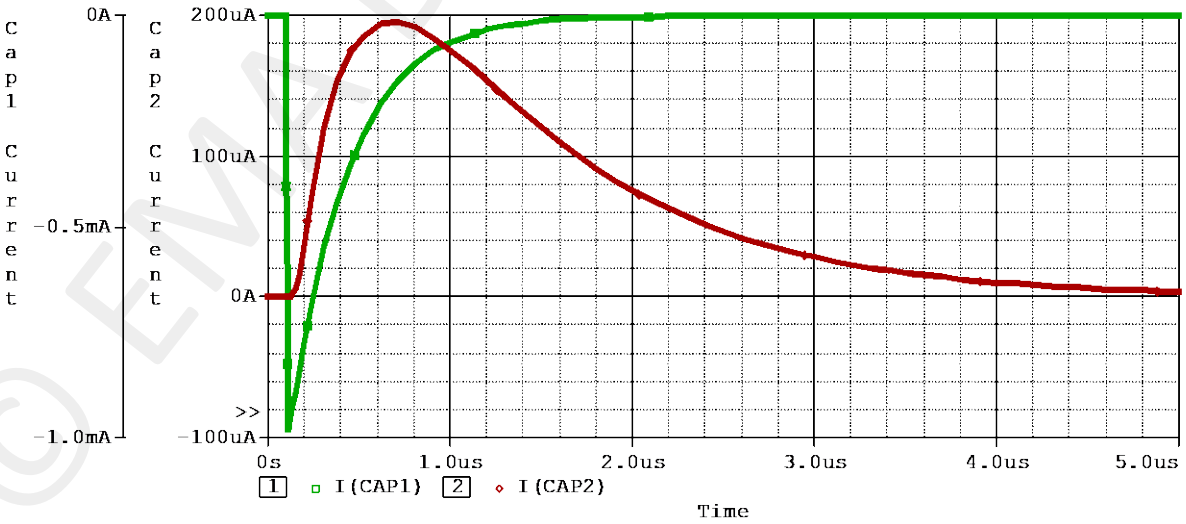


2. Select ***I(CAP1)*** and then press ***OK***.
3. From the ***Probe window***, choose ***Plot > Add Y Axis*** to add the second Y axis.
4. Using the ***Add Traces*** dialog box, add trace ***I(CAP2)***.
5. From ***Probe***, double-click the Y-axis label ***1*** to open the ***Axis Settings*** dialog.



6. Enter **Cap1 Current** in ***Axis Title*** field.
7. Select ***2*** from the **Y-Axis Number** drop down list.
8. Enter **Cap2 Current** in the ***Axis Title*** field.
9. Click ***OK*** to save changes and close the dialog.

The Probe display resembles the following graphic.



The first Y-axis is moved to the far left. The “>>” symbol points to the current axis. Any activities to change the title, range, or other settings are applied to the current axis. Click the desired axis to make it the current axis.

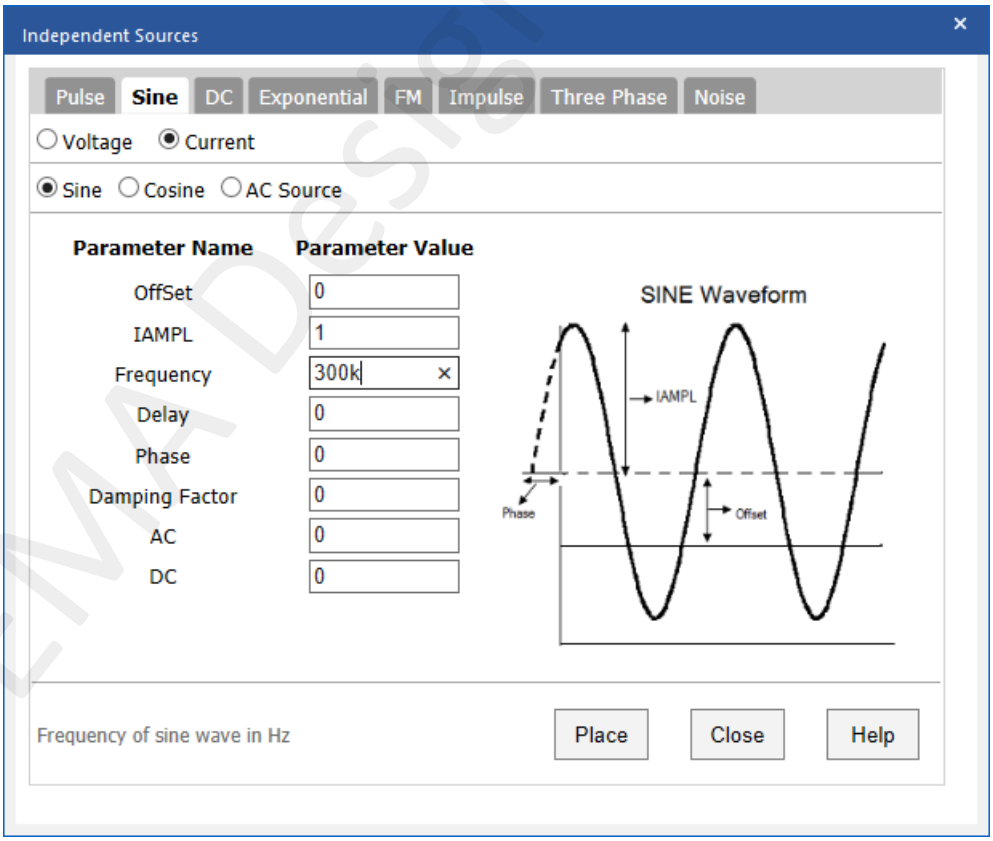
Lesson 5, Lab 3: Resolution

Lab Objectives

- After completing this lab, you will be able to increase resolution on waveforms with insufficient sampling.

Create a New Project

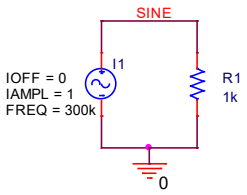
1. Create a new PSpice project based on a blank project called `sine_wave_resolution`
2. Hit Place > PSpice Component > Modeling Application to bring up the Independent Sources dialog
3. Choose the **Sine** tab at the top, then the **Current** radio button and leave the **Sine** radio button selected
4. Set the **Offset to 0**, the **IAMPL to 1** and the **Frequency to 300k**



5. Hit **Place** and put the source on the schematic

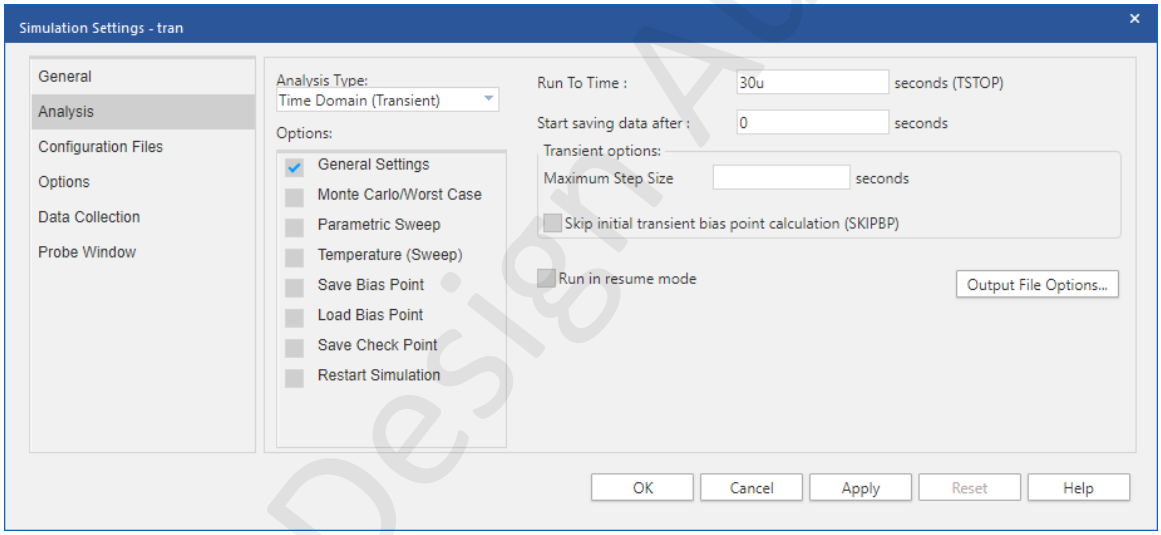
6. Add a **resistor** and the **PSpice Ground** to the schematic

7. Wire the parts together and name the net off the current source `SINE`, like this:



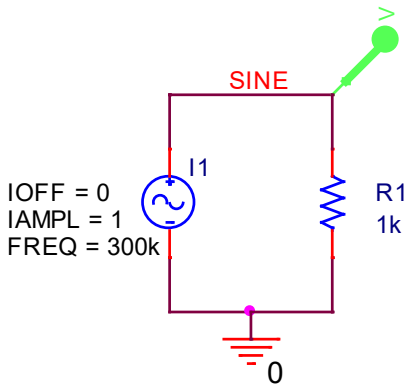
Create a New Profile

1. Create a **new simulation profile** called `tran` and set the **Run to time** for **30u**



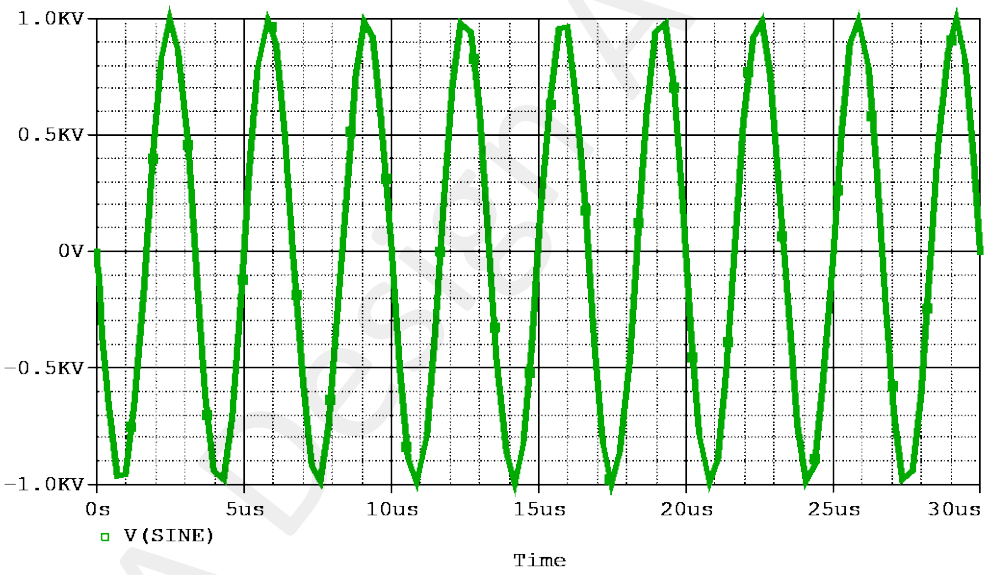
2. Click ***OK*** to close and save the simulation profile.

3. Add a voltage marker to the positive side of the current source, like this (you can hit the **R** key to rotate the probe):



4. Run the simulation (***PSpice > Run***, icon or F11 key)

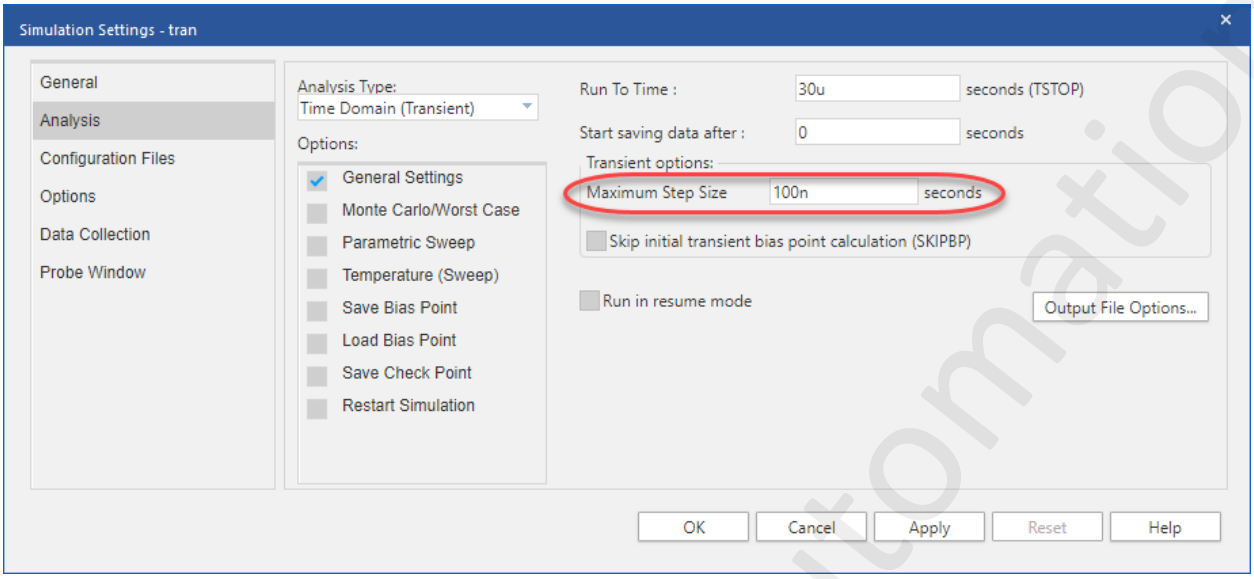
Although the simulation runs successfully, the resulting waveform is very rough. A smooth sine wave was expected.



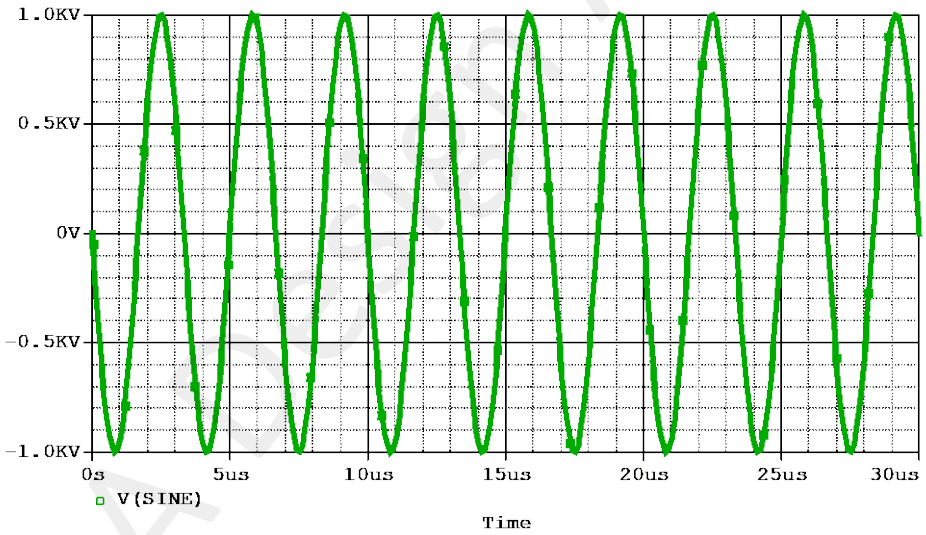
Cleaning the Results Up

PSpice takes approximately 50 simulation sample points from beginning to end so with a Run to Time of 30us that's a simulation sample point approximately every 600ns. We want the simulator to sample more often than that to 'clean up' the results

1. Edit the simulation profile and set ***Maximum Step Size*** to ***100n***.



2. Rerun the simulation and examine the resulting waveform in the Probe Window.



The sine wave is now smooth. The resolution of the waveform display was increased by reducing the simulator's maximum step size.

3. In Capture, choose ***File > Save*** or click the Save toolbar icon ( )

Lesson 5, Lab 4: Scheduling Runtime Settings

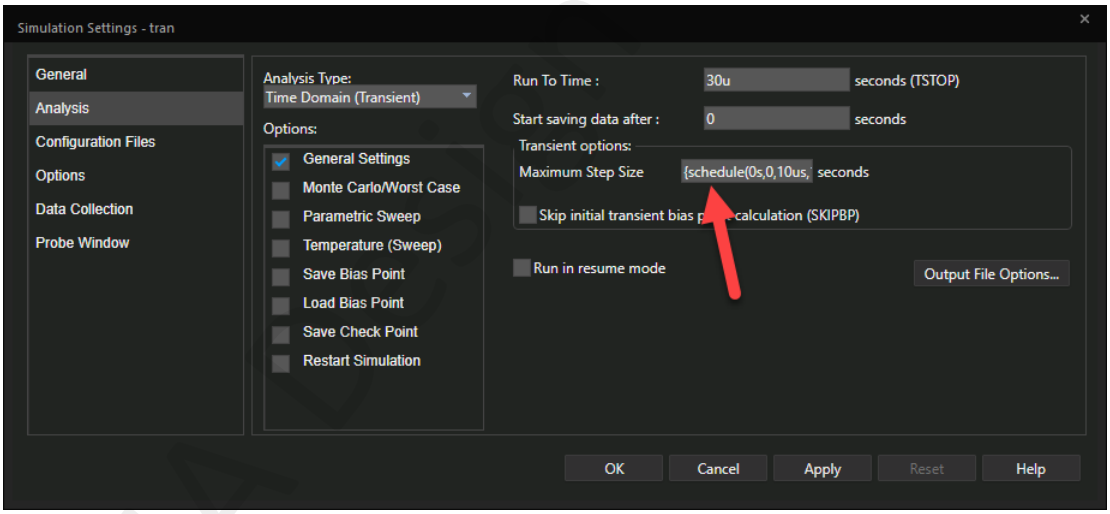
Lab Objectives

- In this lab we will look at **scheduling** PSpice A/D runtime settings. This allows you to schedule the values for various runtime options before running a simulation.
- After completing this lab, you will be able to edit runtime settings.

Opening the sine_wave_resolution circuit

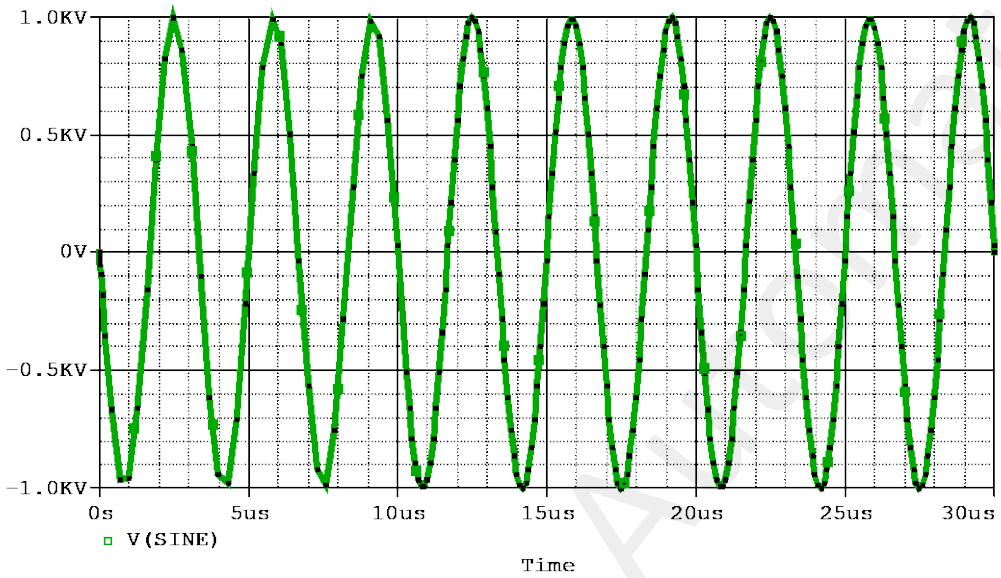
1. If you have closed the `sine_wave_resolution` circuit, reopen it
2. Edit the simulation profile and enter **{schedule(0s,0,10us,100n)}** for the ***Maximum Step Size*** field

This schedules the maximum step size to change from a system generated value (zero) to 100n at time 10us.



3. Click ***OK*** to close and save the simulation profile
4. Run the simulation (***PSpice > Run***, icon or F11 key)
5. Examine V(SINE) in the Probe window. You should see the resolution of the trace change at 10us. Before 10us the trace should be rough. After 10us the trace should be a smooth sine wave

6. Toggle the Mark Data Points icon  to visualize where PSpice took simulation data points



Lesson 5, Lab 5: Making Screenshots

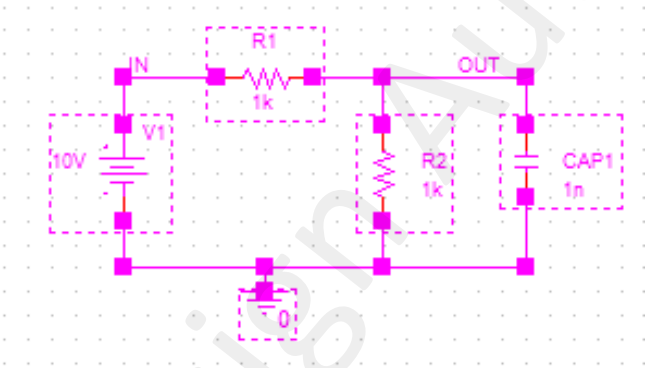
Lab Objectives

After completing this lab, you will be able to:

- Take a screenshot of your Capture schematic into MS Word

Copy the Source Graphics

7. In Capture, **area select** around the schematic contents in Capture to **highlight everything**



8. Hit **Ctrl+C** on your keyboard or else Right Mouse Button (**RMB**) and choose **Copy**

9. **Open MS Word** and create a new **blank** document

If MS Word isn't available, use any word processing tool that is available; if there's nothing available that suits, you can skip this lab.

10. Paste the contents of your clipboard in using **Ctrl+V** or **RMB** and choose **Paste**

Note that you can enlarge the picture with no negative effects on the quality of the image. That's why this is the preferred way of making screenshots over doing screen captures.

11. Close **MS Word**

Lesson 6. Modeling Application

PSpice File Types

The four most important file types to remember are DSN, OPJ, OLB, and LIB.

Modeling App Templates

The Modeling Application facilitates the quick design of PSpice components for simulation without leaving the Capture environment. Once the template is set up with the necessary inputs, the modeling applications will provide a symbol and PSpice model. The modeling app provides templates including transient voltage supressors, sources, parasitic passives, diodes, switches, transformers and voltage-controlled oscillators. In 17.4 this tool can be easily accessed from the schematic toolbar.

When a new device is placed in the schematic, a master index library file is automatically configured at the design-level in the currently active simulation profile for some of the models. This ensures that the project is ready for simulation and the newly created model library is available to PSpice for simulation. The library, PSpiceModelApps_Include.lib, is created and all models for the design that require a library are stored in this library. This library file is placed in the PSpiceModelApps folder under the <Projectname>-PSpicefiles folder.

Each new unique model instance is added to this library file, which is included at design- level. This makes the models for newly developed devices available for all simulation profiles under that project.

Sources

The modeling application supports the following types of sources are supported along with the wave types:

- Pulse: Step, Pulse, Square, Ramp, Sawtooth, Reverse Sawtooth, and Triangular
- Sine: Sine, Cosine, and AC Source
- DC: Ideal DC and DC
- Exponential
- Single Frequency FM

- Impulse: 1.2/50 uSec, 4/10 uSec,4/20 uSec, 8/20 uSec, 10/350 uSec, and 10/1000 uSec.
- Three Phase (Only voltage): DELTA and STAR configurations.
- Noise: DC, Sine, Exponential, and Random Noise
- Piecewise Linear (PWL Sources)

Parasitic Passives

To add a resistance in series and in parallel to a capacitor, specify the equivalent series resistance (ESR) and parallel resistor (RP) values in ohm. To add inductive parasitics, specify either one of the series inductance (ESL) or the self-resonant frequency (SRF) mentioned by the manufacturer. The application calculates the other value based on the following equation, and displays the result in a grayed field:

$$ESL = \frac{1}{\left((2\pi.SRF)^2 C\right)}$$

Modeling of inductors is important for analog applications, especially those involving a wide range of frequency or high DC currents, such as EMI and DC filters, respectively. You can model a non-ideal inductor using the DC series resistance Rdc, inductance, and the self-resonant frequency (SRF) of the inductor to represent the non- linear characteristics. Based on these inputs, the application calculates the parasitic capacitance value for the inductor:

$$C_p = \frac{1}{((2\pi SRF)^2 \times L) + (R_{dc} \times (2\pi SRF))}$$

Using a Text File as a Source

Use a text file with Time and Voltage pairs as a library source where the `FILE` property should have a value of the path to where the text file is located. The pairs may be comma or space separated on either one or multiple lines of the files. These files are often generated from some third party software or digital scopes.

You can also use `VPWL_F_RE_FOREVER` to repeat a text file infinitely or the `VPWL_F_RE_N_TIMES` part to repeat the period a defined number of times managed by the `REPEAT_VALUE` property.

Lesson 6, Lab 1: Creating a Sine Wave Source

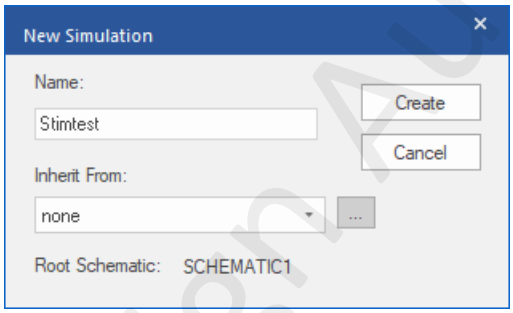
Lab Objectives

- After completing this lab, you will be able to create a 60Hz sine wave source.

Placing the Sinusoidal Source

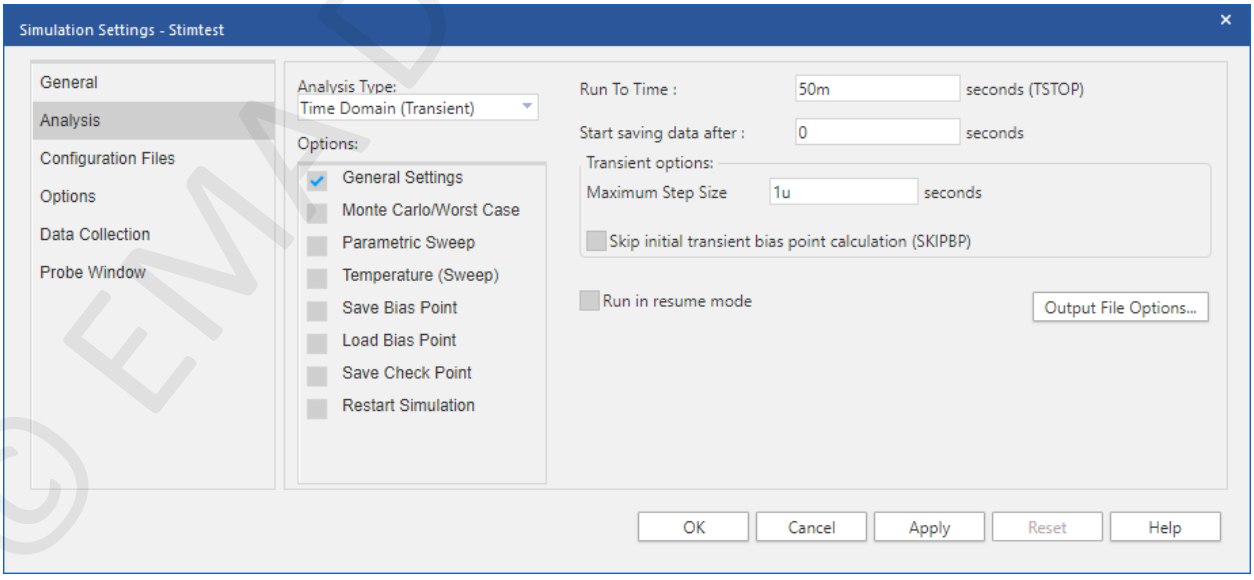
12. In OrCAD Capture, create a project called **Stimtest**

13. Create a new profile (***PSpice > New Simulation Profile*** or  icon) called **stimtest**. Hit **Create**



As shown in the graphic that follows, suggested simulation settings include:

- Run to time = 50m seconds
- Maximum step size = 1u seconds



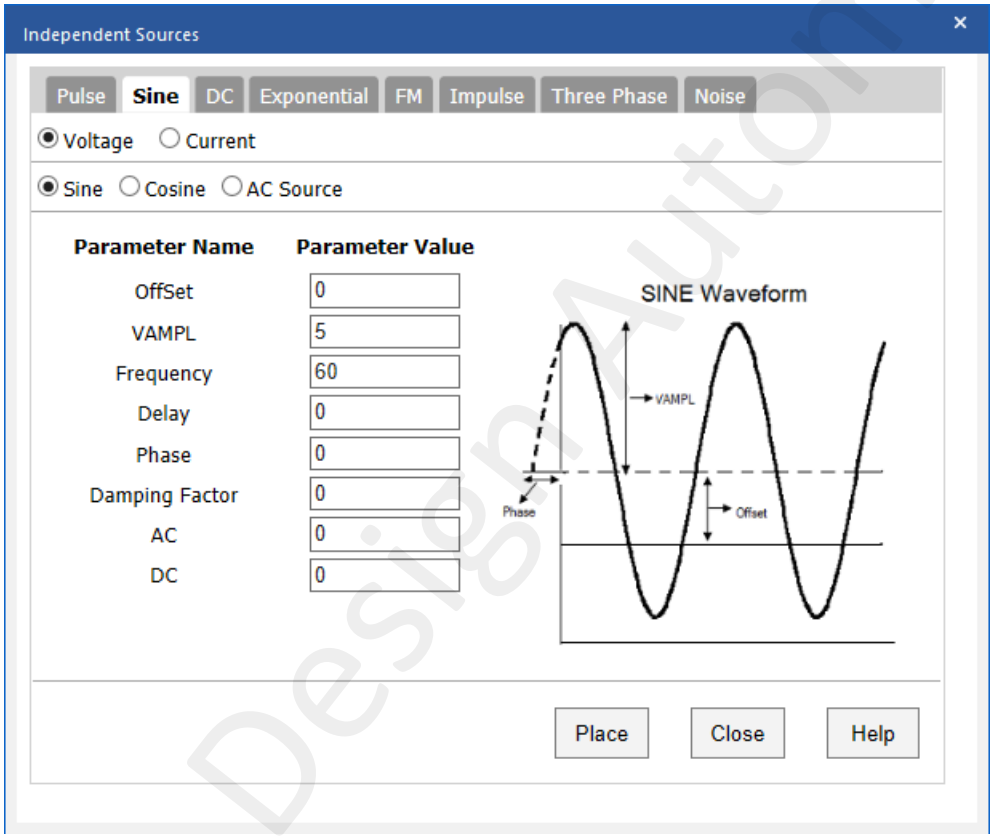
14. Hit **OK**

15. Access the Modeling Applications with Place > PSpice Component > Modeling Application and select Sources > Independent Sources

A dialog box opens that enables you to select the type of stimulus you want to create.

16. Check the ***Sine*** tab and select **Voltage**.

17. In the resulting dialog, enter **0** for ***Offset*** value, **5** for ***VAMPL***, and **60** for ***Frequency***



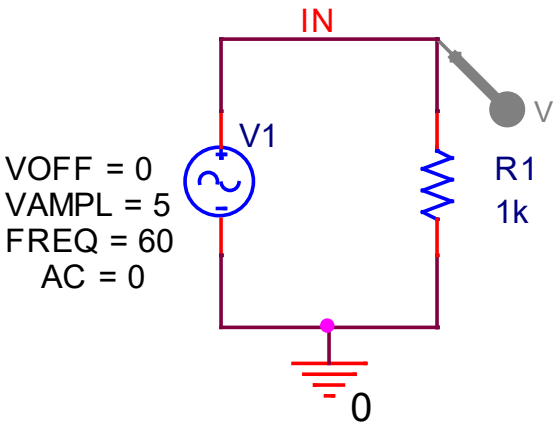
Note that amplitude is peak amplitude not RMS.

18. Click ***Place***

Notice the 60Hz sine sin wave source is now attached to your cursor.

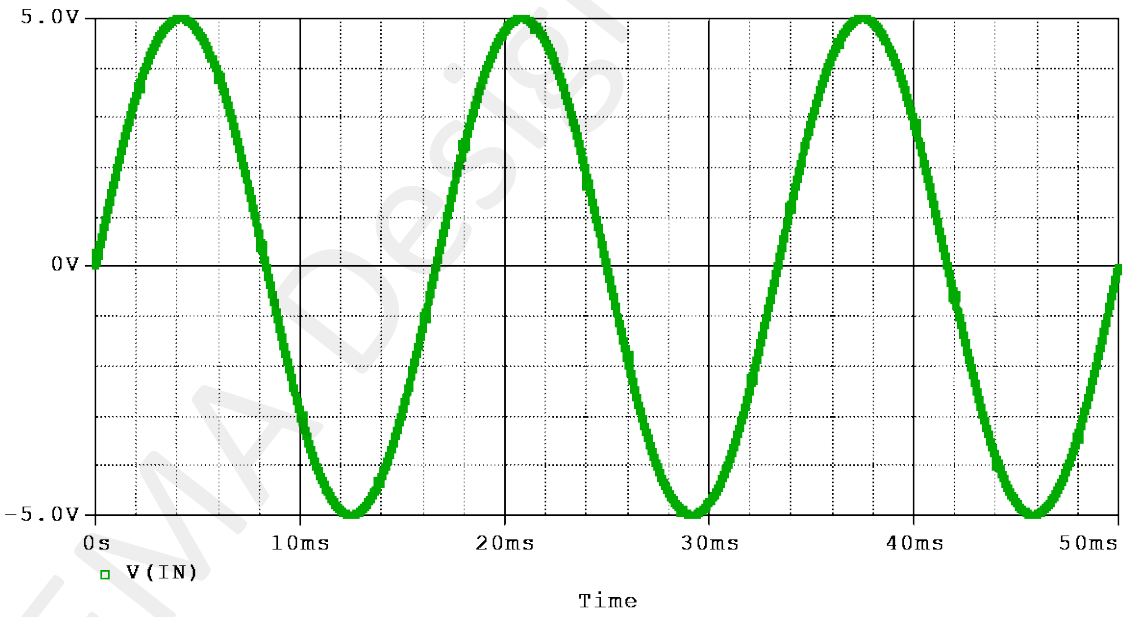
19. Place the source symbol on the schematic and wire it with a 1k resistor and ground. Add a voltage marker to the schematic (at the non-ground node)

20. Alias the Node IN



Running the Simulation

1. Run the simulation (***PSpice > Run***, icon or F11 key)
2. Verify the results in the Probe window



Lesson 6, Lab 2: Creating a Piecewise Linear Source

Lab Objectives

- Create a piece-wise linear source
- This lab is a continuation of the previous.

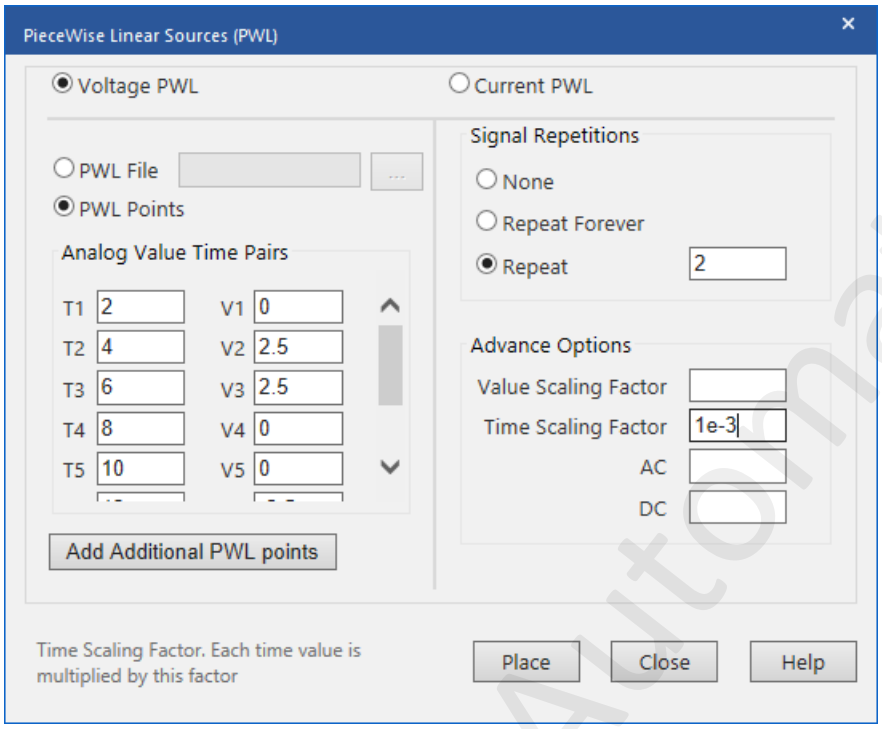
Creating a Piece-wise Linear Source

1. Return to the page 1 schematic of ***Stimtest.opj*** and remove the sinusoidal source
2. From the Place > PSpice Component… > Modeling Application… or choose icon 
3. Choose Sources > PWL Sources
4. Select the **Voltage PWL** radio button
5. For the ***PWL Points***, enter:
    - T1: 2 V1: 0
    - T2: 4 V2: 2.5
    - T3: 6 V3: 2.5
    - T4: 8 V4: 0
    - T5: 10 V5: 0

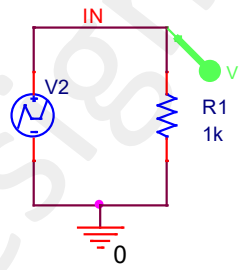
You may need to click on Add Additional PWL points to create new rows of data to fill in here and on remaining rows.

    - T6: 12 V6: -2.5
    - T7: 14 V7: -2.5
    - T8: 16 V8: 0

6. In the Signal Repetitions section, click on the **Repeat** radio button and fill in a value of **2** to repeat this waveform a total of 2 times
7. In Advance Options (lower right corner) enter a Time Scaling Factor of **1e-3** to change the time to milliseconds instead of seconds

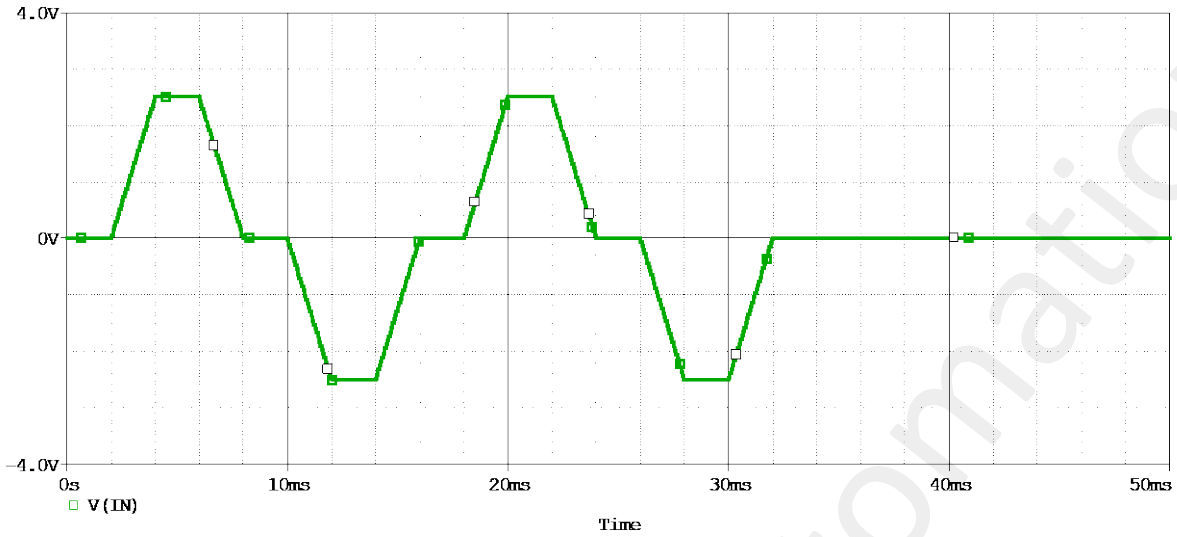


8. Click ***Place*** and place the PWL source into the test circuit



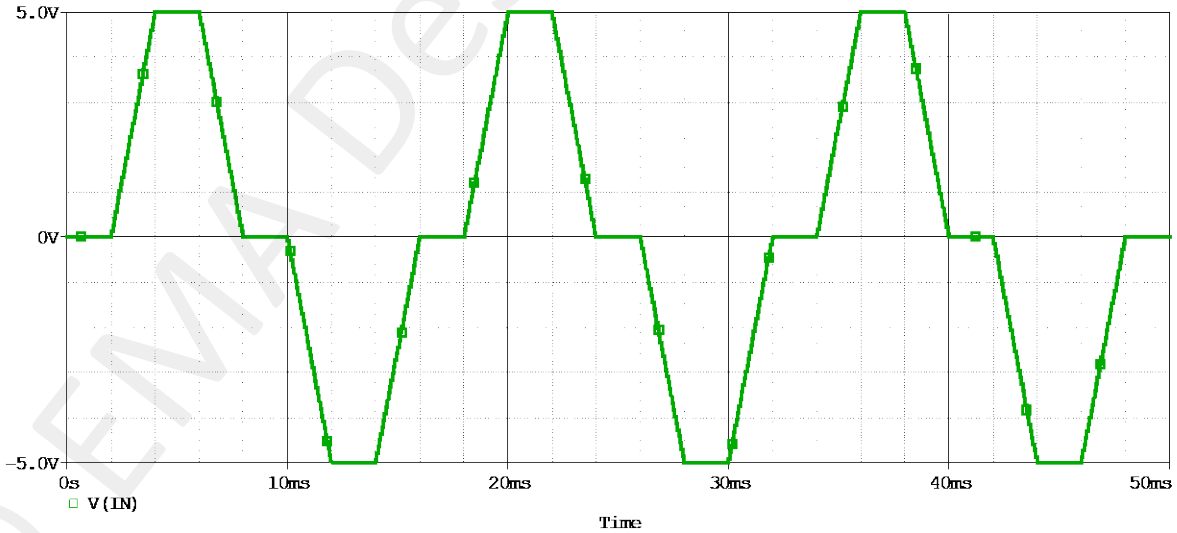
Verifying the Results

1. Run the simulation (***PSpice > Run***, icon or F11 key)
2. View the output.

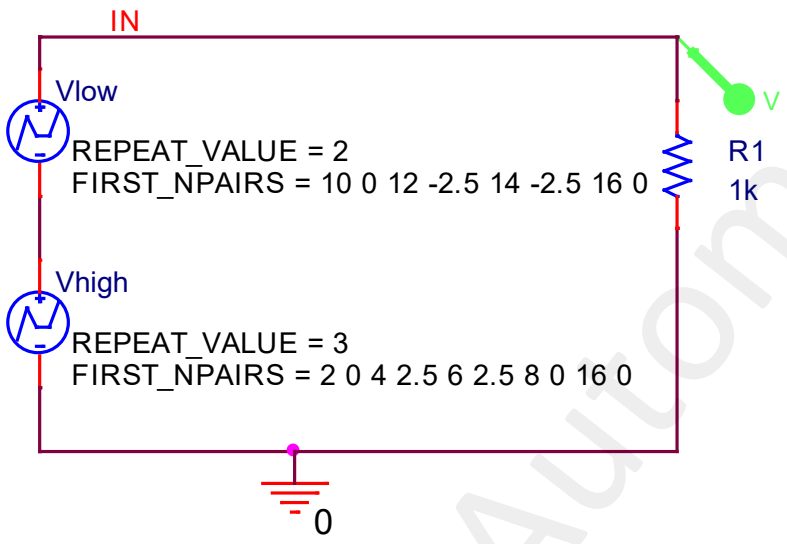


Optional Exercises

- Change the PWL Source to repeat 3 times instead of 2 (modify the REPEAT_VALUE property on the VPWL source)
- Scale the Voltage multiplier to be doubled using the VSF property on the VPWL part
- Create the same PWL source from a text file
- How could you create a voltage source that went high a different amount of times than it went low? Say it went high three times and low twice, but you'd like that to be easily configurable, like this:



Feel free to come up with your own but if you want a suggestion, one possible solution is this (you can use the Modeling Application PWL Source to make this part instead of manipulating the properties manually):



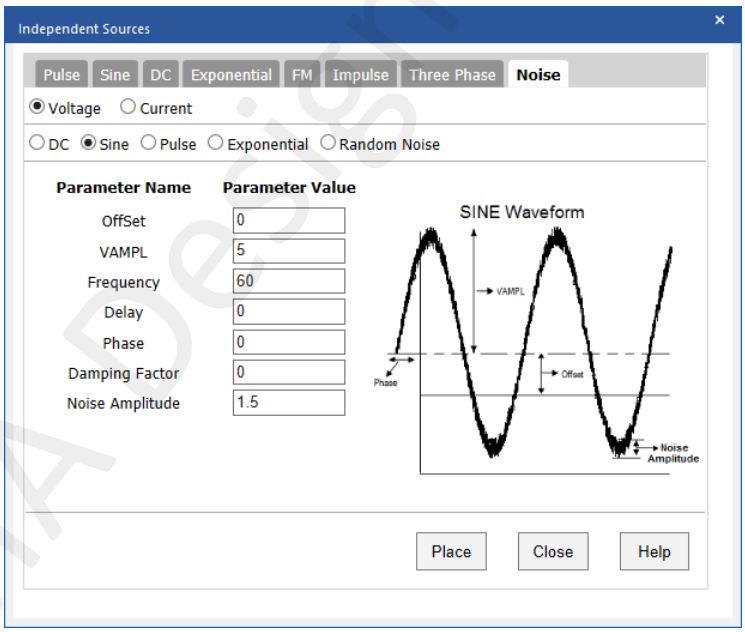
Lesson 6, Lab 3: Creating a Random Noise Source

Lab Objectives

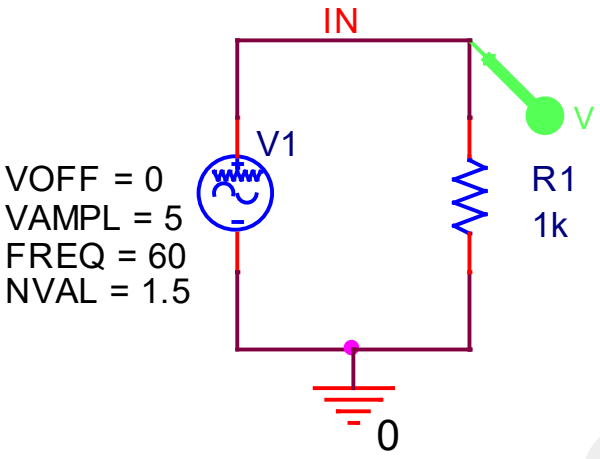
- Create a 60Hz sinusoidal source with random noise
- This lab is a continuation of the previous

Creating a Random Noise Source

1. Return to the page 1 schematic of ***Stimtest.opj*** and remove the PWL source
2. From the Modeling Application menu choose Sources > Independent Sources
3. Check the ***Noise*** tab and select **Voltage**
4. Select the ***Sine*** radio button
5. Enter **0** for ***Offset*** value, **5** for ***VAMPL***, **60** for ***Frequency***, and ***1.5*** for ***Noise Amplitude***

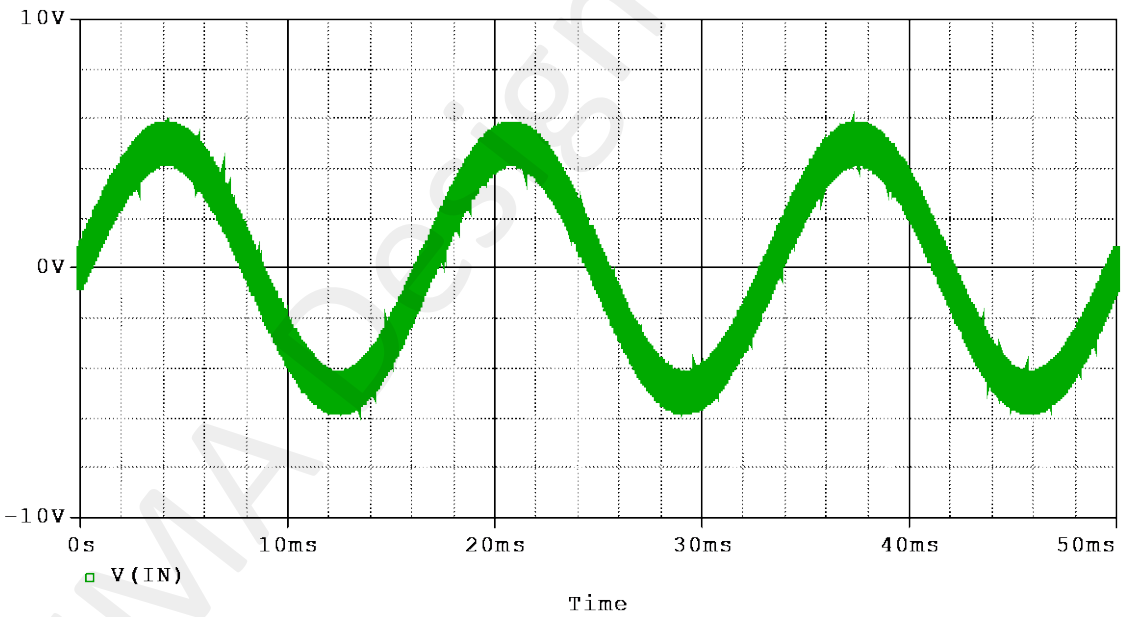


6. Click ***Place*** and place the PWL source into the test circuit



Verifying the Results

1. Run the simulation (***PSpice > Run***, icon or F11 key)
2. View the output



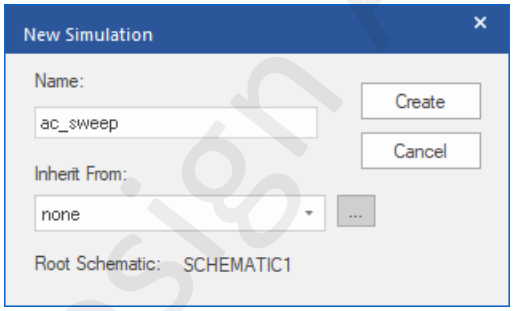
Lesson 6, Lab 4: Creating a Non-Ideal Capacitor

Lab Objectives

- Create a non-ideal capacitor
- Observe the resonance in an AC sweep
- Creating a Piece-wise Linear Source

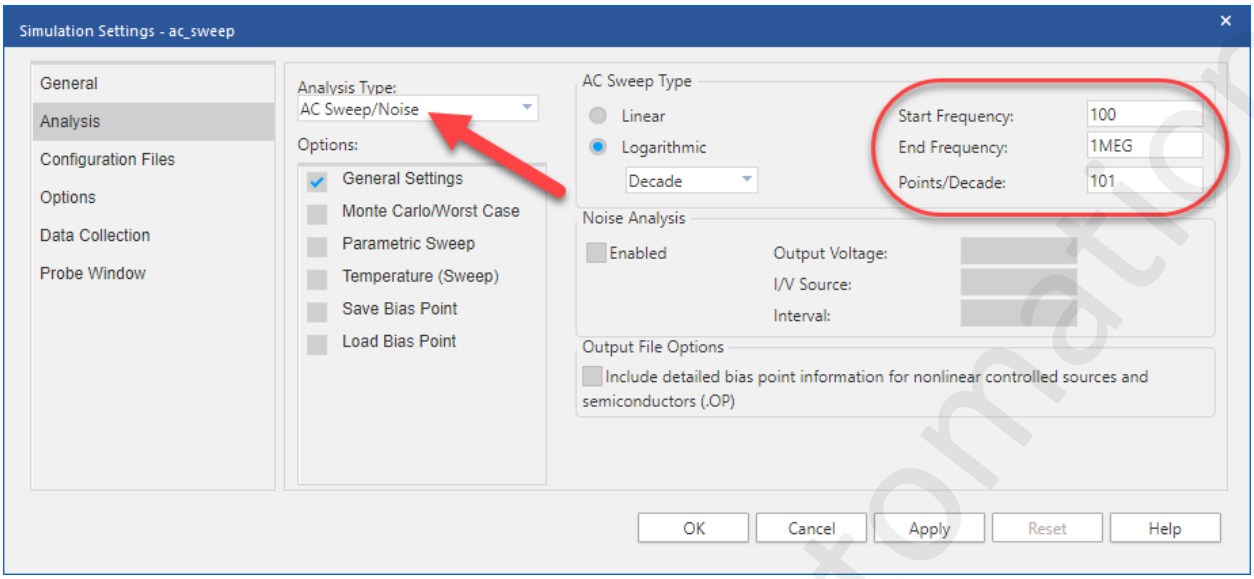
Create a New AC Sweep Project

1. In OrCAD Capture, create a new project called **NonIdealCap** based off a blank project
2. Create a new profile (***PSpice > New Simulation Profile*** or  icon) called **ac_sweep**
3. Hit **Create**



As shown in the graphic that follows, suggested simulation settings include:

- **AC Sweep/Noise** Analysis Type
- Logarithmic Sweep from 100 to 1MEG
- Points per Decade = 101



4. Hit **OK**

Create a Non-Ideal Capacitor

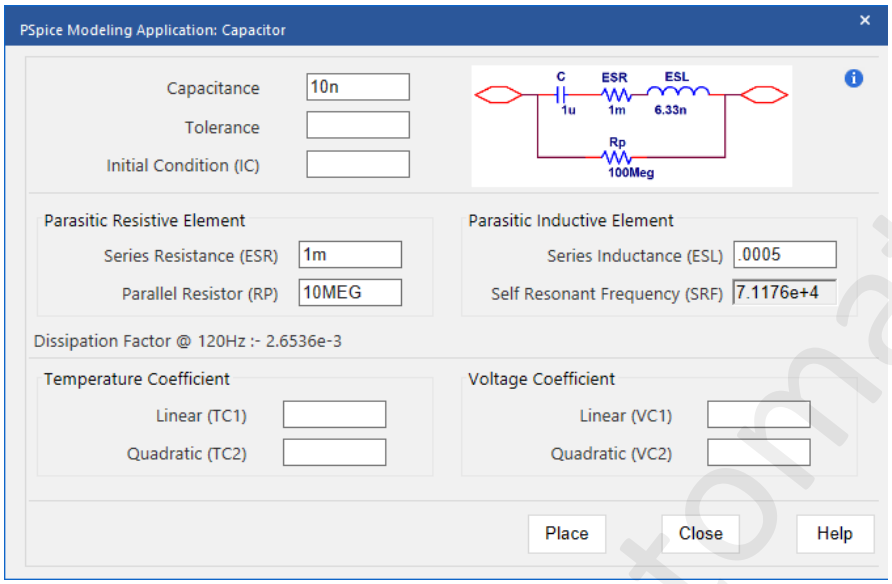
1. Access the Modeling Applications *(**Place > PSpice Component > Modeling Application**)* or press icon and select ***Passives > Capacitor***

A dialog box opens that enables you to specify the parasitic behavior of the capacitor.

2. In the resulting dialog, enter **10n** for ***Capacitance***, **1m** for ***Series Resistance***, and **10MEG** for ***Parallel Resistance***

3. Enter **0.0005** for ***Series Inductance***

Note that the ***Self Resonant Frequency*** will be automatically calculated from these other values to be **~71kHz.**



Note that as you fill in each value, the modeling app will provide you a description of that property.

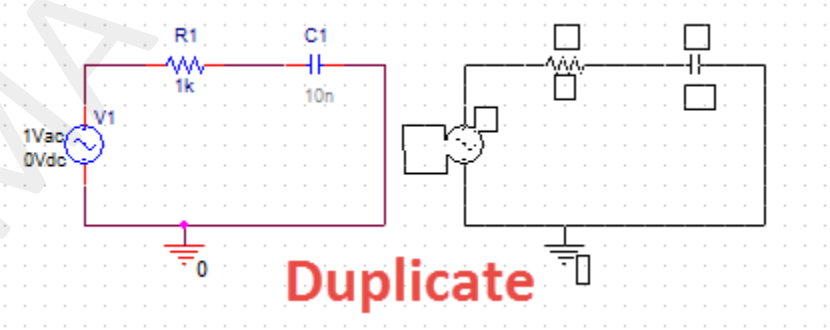
4. Click ***Place***

Notice the capacitor is now attached to your cursor

5. Place the capacitor on the schematic

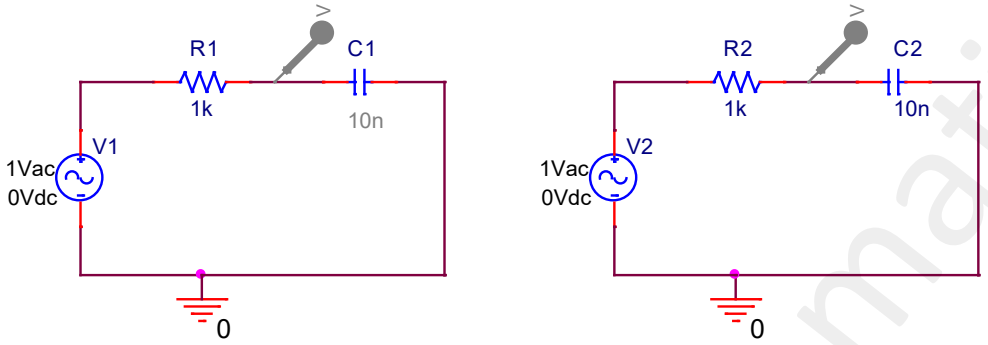
Create a Test Circuit

1. Wire the capacitor in series with a 1k resistor, a VAC source and tie in ground
2. Select everything and make a copy to duplicate the circuit; you can use **Edit->Copy** and then **Edit->Place** to paste the copied circuit or **Ctrl+C** and **Ctrl+V**



3. Add a voltage marker to both circuits (between the resistor and capacitor)

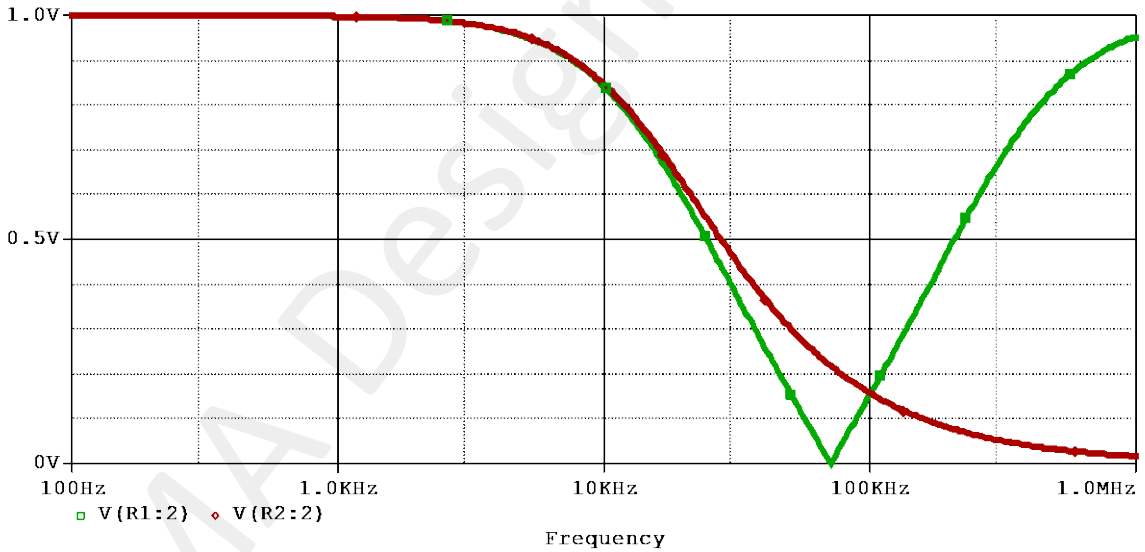
4. Delete the duplicated non-ideal capacitor C2 and add in an ideal capacitor and change its value to **10n**



Verifying the Results

1. Run the simulation (***PSpice > Run***, icon or F11 key)

2. View the output and confirm that the resonant frequency of the capacitor is at 70kHz



Note the difference between the circuit with a non-ideal capacitor (green trace) and one where an ideal capacitor is used (red) – it's a pretty big deal at higher frequencies!

Additional Exercises

- Create a non-ideal inductor using the modeling app

Lesson 7. Resolving Simulation Errors

Diagnosing Simulation Problems

Find the line with the ERROR in it

Look on the line **above** for the $

On the line above that, at the location of the $ is what the ERROR is referring to

Common Circuit Errors and Solutions

Common causes and solutions of circuit errors:

1. Unmodeled pins:

- ERROR – Less than two connections at node <name>.
- SOLUTION – Make certain that the electrical connection has been made to at least two pins on the node.

2. Missing ground:

- ERROR – Node <name> is floating.
- SOLUTION – Check that there is a PSpice A/D ground in the circuit. A PSpice A/D ground will label the node as **0**.

3. Missing DC path to ground:

- ERROR – Node <name> is floating.
- SOLUTION – Check that the node is not isolated from ground (zero) in DC conditions. You may need to add a large resistor to ground for nets that are isolated.

4. Voltage source or inductor loop:

- ERROR – Voltage source or inductor loop.
- SOLUTION – Add a small series resistance.

Be sure to notice the difference between the floating node error and the less than two connections at node xxx error. The first means that there is no DC path to ground for the listed

node. The second error means that there is a wire connected to a device pin without connection to another pin.

A floating node error may be caused by:

- Missing ground
- A node isolated from DC current because it is between two capacitors
- A node isolated from DC ground because of an Analog Behavioral Modeling device. ABM devices do not actually pass a current during simulation. For now, think of them as black boxes whose output is based on an equation using their inputs.

The less than two connections error may be caused by:

- Missing the connection point on a pin when the snap to grid function is disabled
- Unintentional use of a non-PSpice A/D part or symbol

Multiple Bias Points

It is possible to have multiple bias points for a given circuit. However, only one of these points is usually valid for the desired circuit operation. It is possible to provide PSpice A/D with an initial "guess" of what a node voltage should be for some or all nodes in a circuit or to force a node voltage.

The NODESET symbols allow you to specify a starting point for the simulator to use in calculating the initial bias solution. Nodesets are released after the beginning of the calculations and will not affect the remainder of the simulation.

IC symbols will force a voltage or current condition on a node or through a device. These are released at the end of the bias calculation.

Both Nodeset and IC symbols can be found in the Special library. Both have a single pin and two-pin version. Single pin symbols are used to set the value on a single wire. The two-pin versions are used to set the voltage level across a device or between two nodes.

Simulation Accuracy

The accuracy of a simulation is controlled by several tolerance variables with the primary one being RELTOL. RELTOL is the relative tolerance of node voltages and branch currents in the circuit. The default value for RELTOL is .001 (.1%). To force the simulator to calculate a more accurate solution, tighten or reduce RELTOL. However, the tighter the accuracy becomes, the more likely it is to encounter convergence errors.

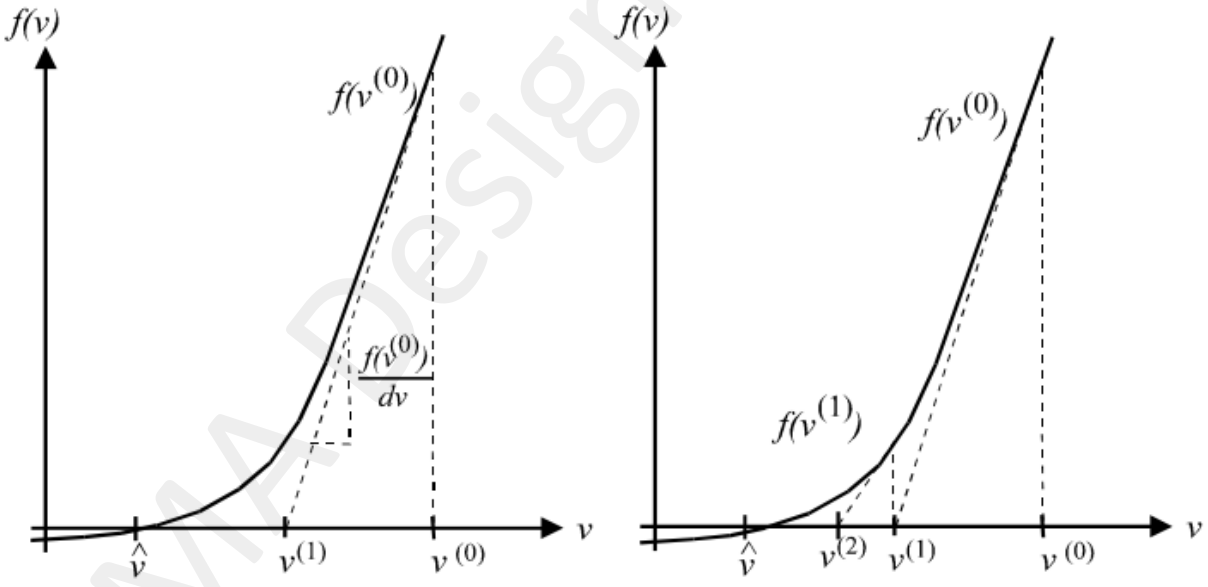
If the waveforms in Probe lack resolution and appear to be rough, specify a maximum step size for the transient analysis that provides an adequate number of data points or a smaller maximum step size in the transient simulation profile. This forces the simulator to take more data samples during the simulation which may slow the simulation.

The step size that the simulator uses is affected by such things as the speed of sources in the circuit, length of transmission lines, and if nothing else affects it, will be set to 1/50th of the final time specified in the simulation profile.

Finding Solutions

PSpice A/D finds the solution to the polynomial equations using a Newton-Raphson method. In essence, start with a solution to the polynomial equations. Take a step and find a second solution and compare it to the first. If the second solution is within RELTOL of the previous solution, accept it as valid and move on.

Newton-Raphson



The graphic above is a representation of how PSpice A/D determines a solution.

Error Tolerances/Convergence Criteria

Normally you do not need to change these tolerance parameters in order to get convergence. However, if you do, be very careful since changing them may allow the simulation to succeed, but it might be giving invalid results if you have loosened (increased) the value too much. The main exception to this rule is for simulating circuits that have currents greater than 1000 amps. In that case you may need to increase RELTOL to shift the dynamic range of the simulator.

PSpice A/D Runtime Settings

As of PSpice A/D release 9.2, you can pause a transient simulation, change certain run time parameters, and then continue the simulation using the new parameter values from the point at which the simulation was paused. The new parameter values are not stored in the simulation profile but are kept in the output file so that you can refer to them later.

The parameters that you can change are:

- Reltol
- Abstol
- Vntol
- Gmin
- Tstop
- Tmax
- Itl1
- Itl2
- Itl4

To Change Runtime Parameters:

1. Pause the current simulation.
2. Choose ***Simulation > Edit Runtime Settings***.
3. Enter new values in the desired fields or check the box to ***Use Original Value***.
4. Click ***OK*** and ***Resume Simulation*** to resume the simulation using the new parameters.

Extending a Simulation

When you enter a value for TSTOP in the RunFor text box on the simulation toolbar in PSpice A/D, you can control the simulation in various ways:

If the RunFor text box is blank, the original value for Final Time in the Simulation Profile will be used. The simulation will run to completion and then stop.

If you enter a value in the RunFor text box, the RunFor value will override any value for TSTOP in the Simulation Profile. The simulation will run until it reaches the RunFor value and then pause.

If the simulation is paused before RunFor (TSTOP) is reached, you can enter a new value in the RunFor text box, then click the Run toolbar button. PSpice A/D will resume the simulation, run for the additional time specified (RunFor), and then pause again.

If you pause a simulation and want to restart it using the original value for TSTOP in the Simulation Profile, clear the RunFor text box and press the Pause button again.

Troubleshooting DC and Bias Point Convergence Failure

There are many steps that can be taken to correct DC and bias point convergence failure. The suggestions included in this section are organized in the same order as the presentation slides of the PSpice A/D training class.

- Examine the non-convergence error message in the .out file to further localize the problem. Examine the previous/current iteration voltage printout, and note which voltages, currents, and devices failed to converge. Voltages and currents that reach 1E10 are clear indicators of problem areas.
- Check the direction of independent and dependent current sources. A convergence problem may result when current is forced backward through diodes or pn junctions.
- Check for diodes that should be forward biased in the reported voltages for the last iteration. Use NODESETs or add a shunt resistor to keep the diodes forward biased.
- Try using GMIN stepping.
- Try raising ITL1 to 500, RELTOL to 0.003, and using NODESETs.
- Look for large currents for the problem devices and currents. A current or voltage that reaches +/-1E10 indicates that the maximum value has been reached. These locations are often the starting point of non-convergence. For power circuits where currents exceed 1000A, it will be necessary to increase ABSTOL, as the default (1E-12) is set for IC currents.
- In a DC sweep, consider increasing ITL2 to 500, and using a smaller step size.

- If a particular model is suspect, isolate it in a test circuit to attempt to generate similar DC curves. If a single device shows the problem, look for unrealistic model parameters. Try simplifying the model to obtain better convergence.
- Try using NODESETs to help PSpice A/D find a stable solution.
- Divide the circuit into smaller pieces and simulate them individually.
- Check for singularities in ABM expressions. In particular, look for denominators which contain circuit variables. Try adding a small offset to the denominator to prevent it from becoming zero.

Troubleshooting Transient Convergence Failure

There are many steps that can be taken to correct transient convergence failure. The suggestions included in this section are also organized in the same order as the presentation slides of the PSpice A/D training class.

- Examine the simulator messages to look for clues as to why the problem occurs.
- When specifying nonlinear device model parameters, be sure that a complete capacitance model is specified. Do not use “ideal” models which do not have the device parasitics modeled.

Most error messages issued by PSpice A/D are fairly descriptive and are a good place to start the troubleshooting process.

The second point is a problem when you are using a default model. This was more of an issue a few years ago when the breakout library used only default parameters for all of its models. Now breakout has minimum values necessary to converge in most uses.

The breakout library contains a default model for every intrinsic PSpice A/D device. Parts from breakout are commonly used as a starting point for creating a new model or in cases where a generic device of a certain type is needed for a simple functional simulation.

- Be sure to give source and drain areas for all MOSFETs, so that the junction capacitors and overlap capacitors are modeled.
- Use an ESR with ideal inductors. Use the Maximum Step Size to limit large time step predictions.
- Try modeling the quality factor of large inductors by placing a resistor in parallel with them. $R=2\pi f_{ro}L$ where f_{ro} is the Frequency roll off of the inductor.
- Increase the ITL4 option to 40 or 100 to increase the possible number of iterations at each time step.
- Loosen the absolute tolerances for power circuits. This is most important for ABSTOL since its default is 1pA. As a rule of thumb, the absolute tolerances should not be more

than 9 orders of magnitude smaller than typical signals present in the circuit. Failure to converge at the first time step is an indication of a dynamic range problem.

- As a last resort, try different values for the Maximum Step size or Step Ceiling in hopes that the simulator will jump over a convergence problem.

A good portion of convergence errors can be solved using one or any combination of the three following things:

Set RELTOL to **.01**. A one percent relative tolerance is usually accurate enough for most circuits. If you are simulating a circuit that has currents in the pico-amp range, then you may want to be a little more careful about changing RELTOL so drastically.

- Side-effects:
  - The simulation will be less accurate
  - The simulation will usually run faster.

Set ITL4 to 40 or 100. If you need to increase ITL4 to more than 100 to get a circuit to converge, then it usually indicates a problem in the circuit or at least suggests that a closer look should be taken.

- Side-effects:
  - The simulation may take longer to run since you are allowing the simulator to make more attempts at solving the equations before stopping
  - May increase the size of the .dat file. This is because the simulator may take smaller steps and thus collect more data.

Set a maximum Step size in the transient simulation profile.

- Side-effects:
  - This will increase the simulation time since you are forcing the simulator to take a smaller step than it would normally.
  - This will increase the size of the .dat file since you are forcing the simulator into a smaller resolution.

Using Auto-Convergence

Auto-converge allows PSpice to automatically adjust the ITL1, ITL2, ITL4, RELTOL, ABSTOL, VNTOL and PIVTOL tolerances to find a solution with the smallest possible disruption to your results

Available from the Simulation Settings > Options Tab, choose the AutoConverge button.

Multi-Core Support

It is not recommended to use all your available cores as that will bring the remaining applications on your system to a halt. Generally, it's a good idea to use half of your cores for larger simulations.

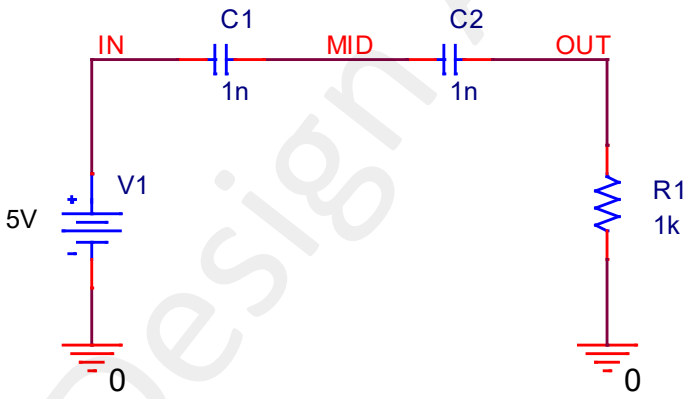
Lesson 7, Lab 1: Floating Node

Lab Objectives

- After completing this lab, you will be able to solve floating node errors.

Create a New Project

1. Create a new PSpice project based off of a blank project called `noDCpath` and save in in the *C:\EMA_Training\PSpiceEssentials\Designs* folder
2. Add these parts: DC Voltage Source, Capacitors (x2), Resistor and PSpice Ground
3. Give the DC Voltage Source a value of 5V
4. Name the nodes in the circuit IN, MID and OUT, like this:



5. Create a new profile (***PSpice > New Simulation Profile*** or icon) called `bias`
6. Change the **Analysis Type** in the Simulation Profile to **Bias Point**, Hit **OK**

Identifying Errors

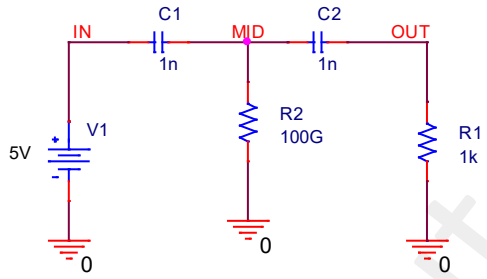
1. Run the simulation (***PSpice > Run***, icon or F11 key)
2. Observe the errors

```
ERROR(ORPSIM-15142): Node MID is floating
```


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) or 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 PSpice ground



2. Re-simulate the circuit.

The simulation now runs without errors.

Lesson 7, Lab 2: Convergence Error

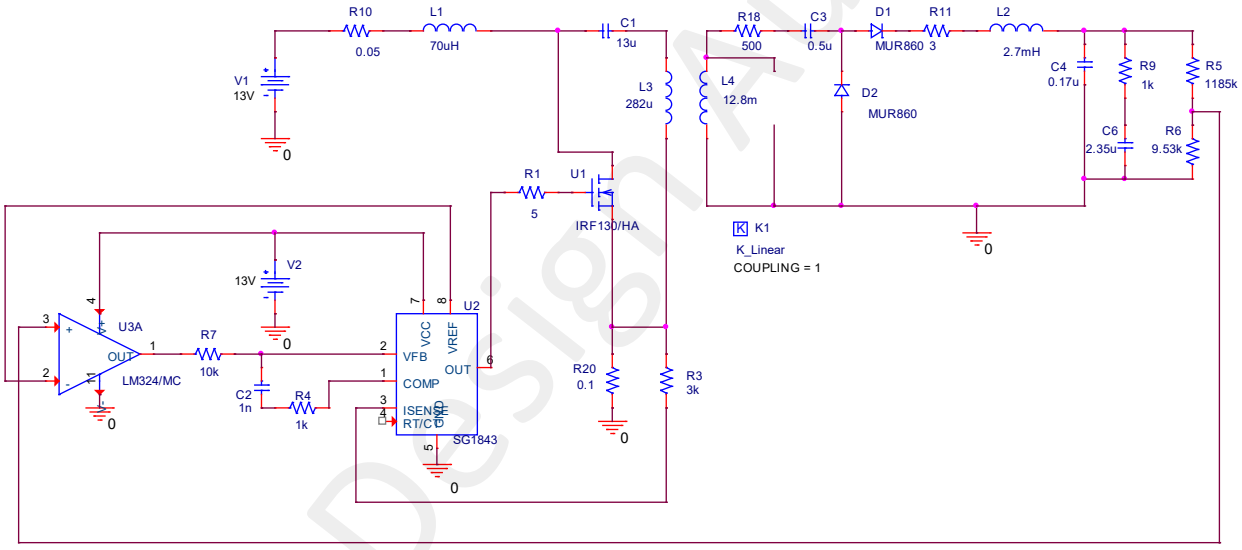
Lab Objectives

- After completing this lab, you will be able to solve a convergence error using AutoConverge.

Open the Project and Run the Simulation

1. Open the project `ConvergeError.opj` from the *C:\EMA_Training\PSpiceEssentials\Designs* folder

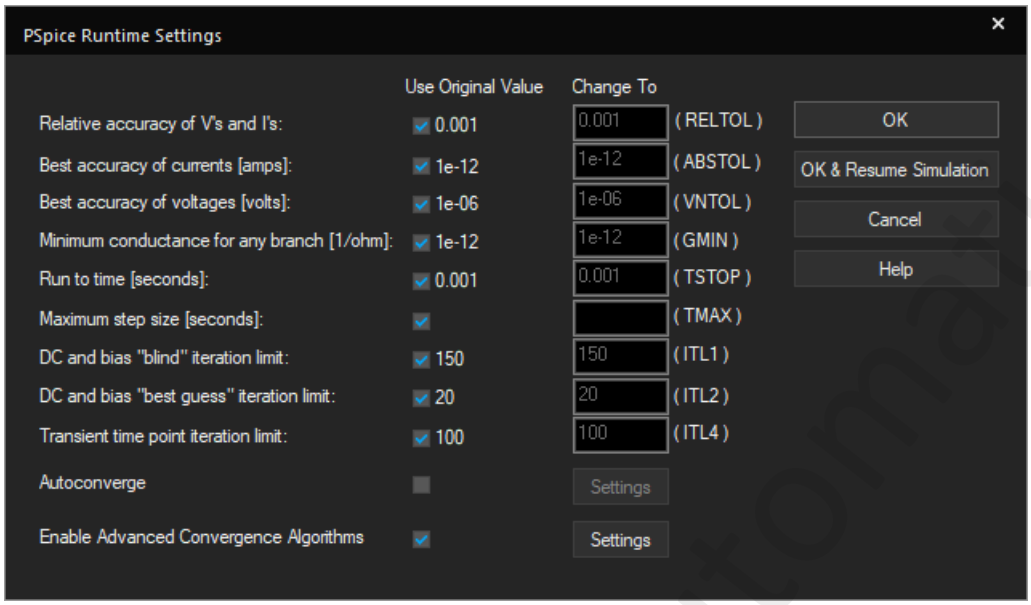
2. Examine the flyback convertor schematic provided in the next graphic.

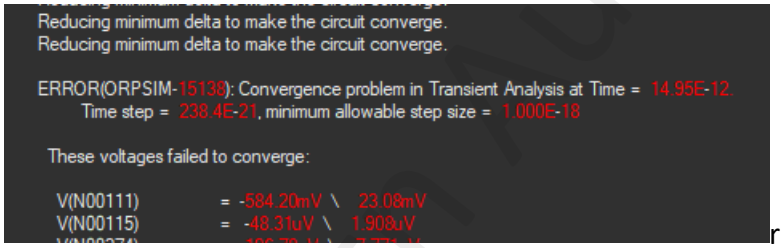


3. Choose the ***PSpice > Edit Simulation Profile*** menu commands to open the ***Simulation Settings*** dialog box.

4. Review the profile settings Run the simulation (*PSpice > Run*, icon or F11 key)

The PSpice Runtime Settings dialog box and a convergence error similar to the one that follows opens as the simulation pauses.





The exact error message you receive may vary slightly.

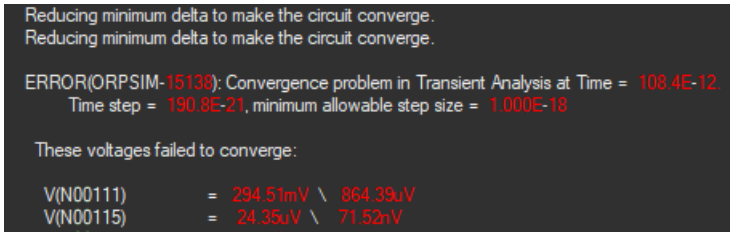
Correcting Simulation Errors

1. Cancel the ***PSpice Runtime Settings*** dialog. Close the ***Probe*** window and stop the paused simulation when asked.

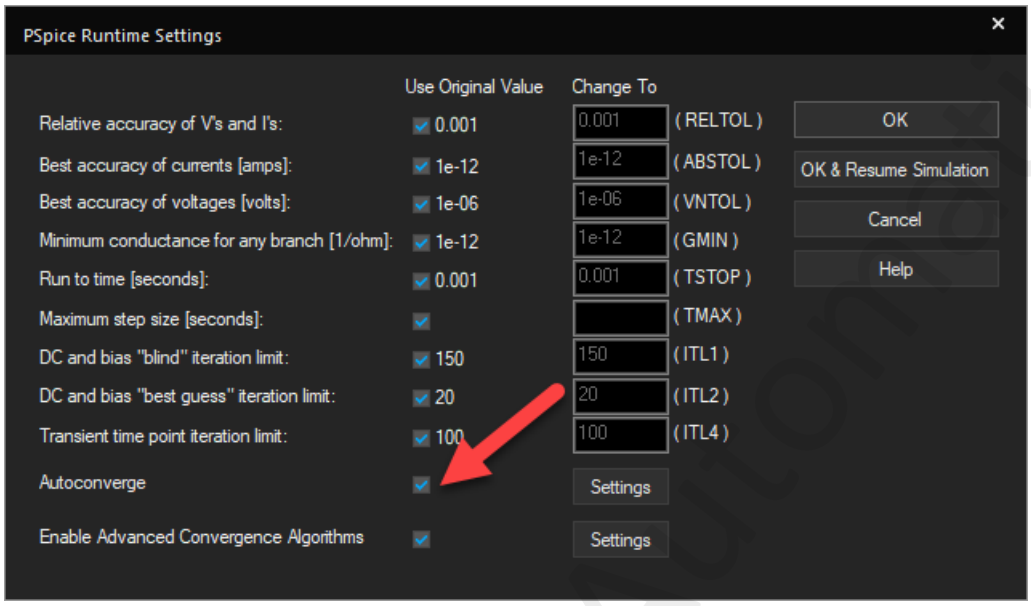
2. Put a 1k snubbing resistor in parallel with L4.

3. Run the simulation again (***PSpice > Run***,  icon or F11 key)

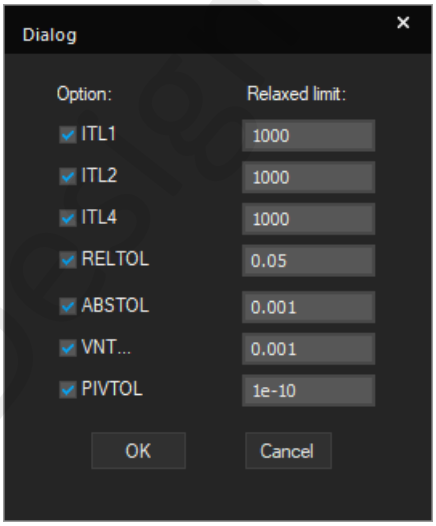
Although the error message changes, the simulation pauses again, and errors or problems still exist.



4. When the PSpice Runtime Settings dialog opens up indicating that you have a convergence error, **click** on the **checkbox** to turn on **AutoConverge**



5. Select the Settings button in the same column to bring up the settings for AutoConverge

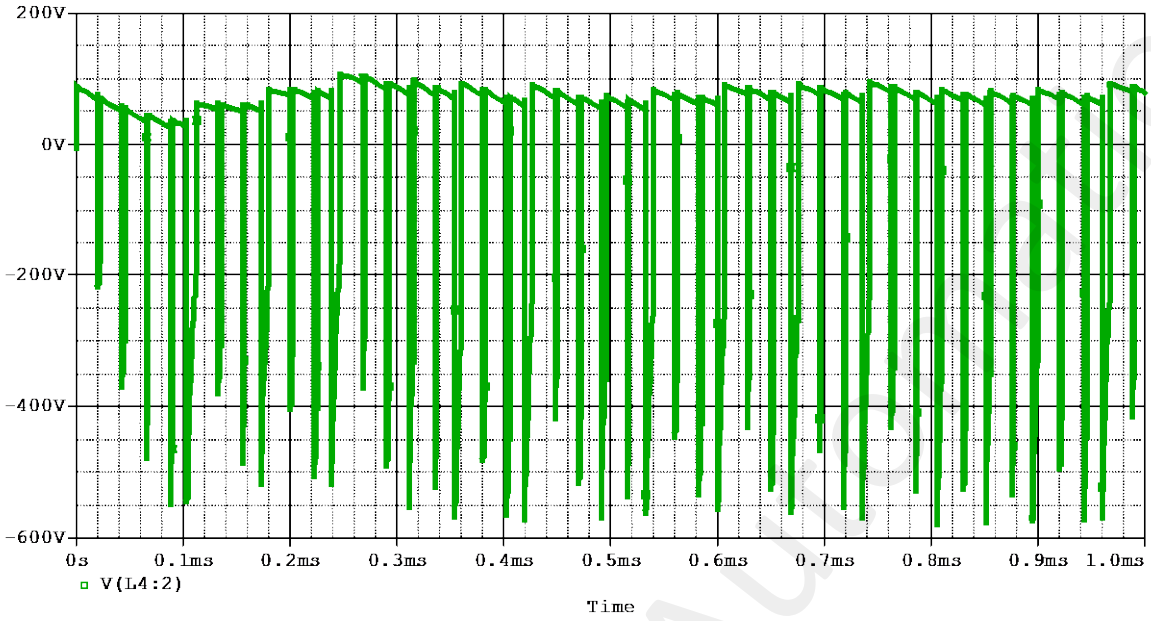


These numbers indicate a relaxed limitation that can be used when the simulator runs into convergence issues (like it has now). The simulator will not throw the current limitations out the window and use these ones but will instead chip away slowly at the nominal values to try to achieve convergence but not take up any tolerance space with these variables that it doesn't need.

6. Hit **OK** on the Dialog window

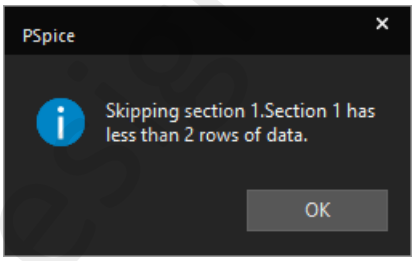
7. Hit **OK & Resume Simulation** on the PSpice Runtime Settings window

The simulation should now progress, and you will be able to see results:



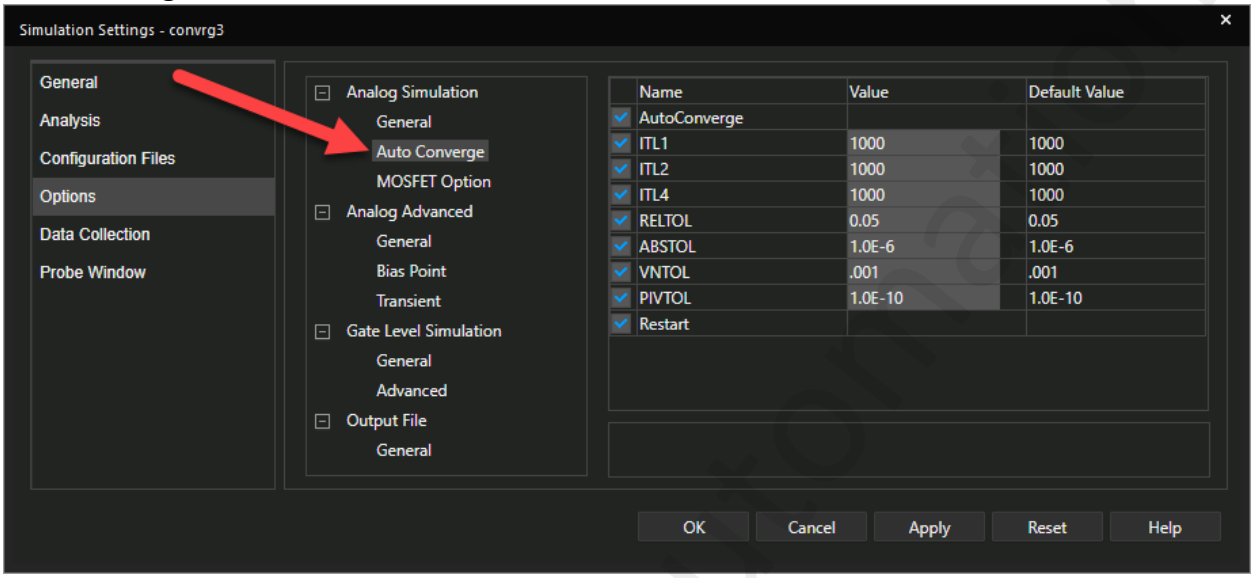
If you get an error here:

```
Skipping section 1.Section 1 has less than 2 rows of data.
```



1. Hit **OK** and **Close** out of PSpice

2. In Capture, **Edit the Simulation** Profile, go to the **Options** Tab and check **AutoConverge…**



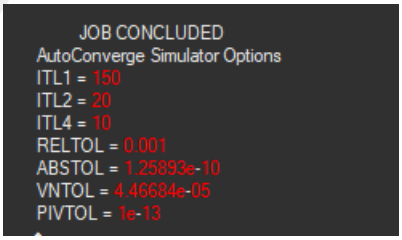
3. Hit **OK** to close out of the Edit Simulation Profile and **re-run** the simulation

This is fantastic that we get to see results graphically instead of having to pore over the output data and try to determine where the problem may lie and how to fix it. Speaking of the output data, we do likely want to see what the tolerance values it ended up using were.

Looking at the Output Data

1. In PSpice, choose *View > Output File*

2. Scroll to the very bottom and that a look at the **AutoConverge Simulator Options** section



This tells you what tolerance values it ended up using to obtain the results that you see on the screen. You will now need to evaluate whether the tolerances that it used to complete the simulation are acceptable to you or not.

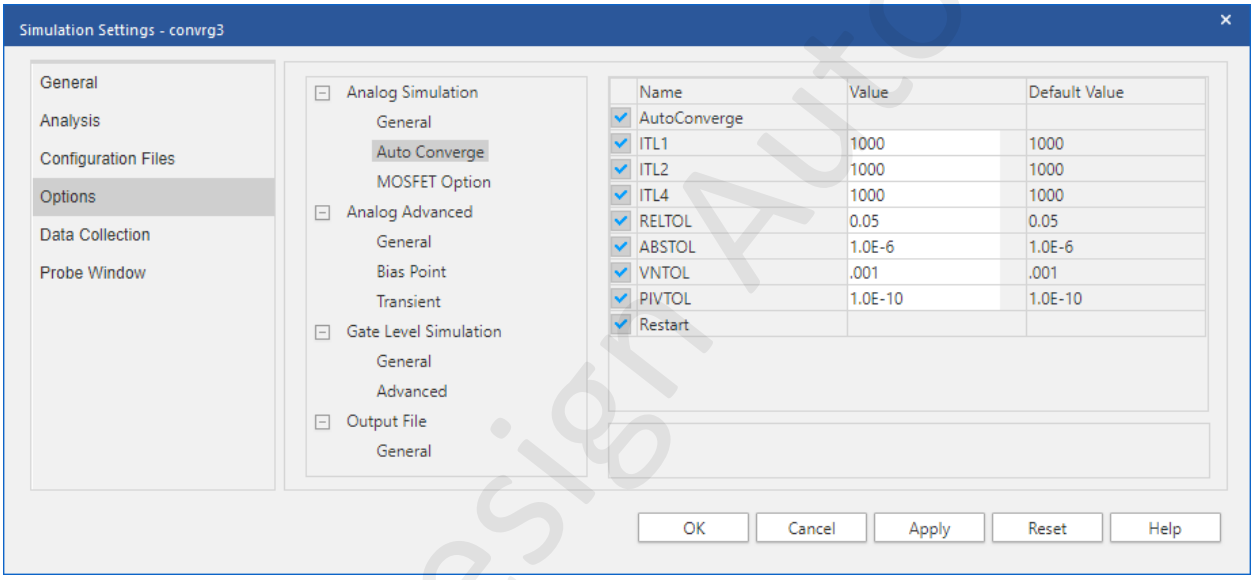
Turn on AutoConverge by Default

If you got the Skipping Section error previously, you already did this.

If you want the simulator to use AutoConverge every time for a particular design, you can turn it on in the simulation profile. This is useful if you know that a circuit requires the AutoConverge functionality to complete but you don't want to have to click the button to turn it on after every simulation that is run.

1. Edit the simulation profile

2. In the **Options** tab click the **AutoConverge** button.



3. Check the **AutoConverge** check box.

When you run the simulation with AutoConvergence set, PSpice A/D initially runs using the normal values for the specified simulation time. However, if the simulation does not converge, PSpice A/D automatically changes the values within the relaxed limit of the parameters selected in the AutoConvergence Options dialog box.

If convergence is not achieved at the end of the complete run, PSpice A/D starts simulation with an initial relaxed value if the Restart option is selected.

4. Click **OK** in the **AutoConverge Options** dialog.

5. Click **OK** in the **Simulation Settings** dialog.

6. Rerun the simulation. It should now run to completion without giving you a pop up window indicating that it failed to converge

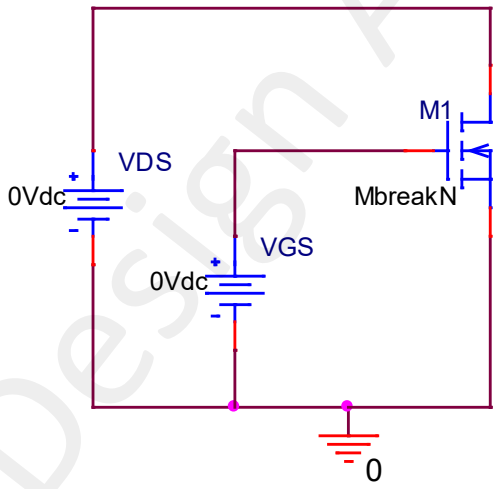
Lesson 7, Lab 3: Discontinuities

Lab Objectives

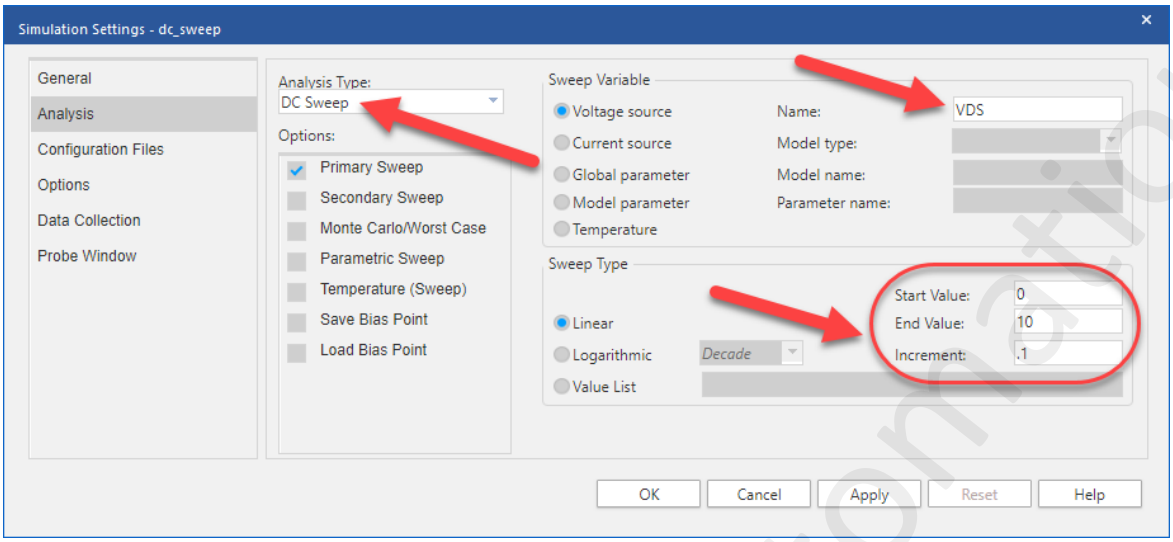
- After you complete this lab you will be able to examine possible model discontinuities.

Create new Project

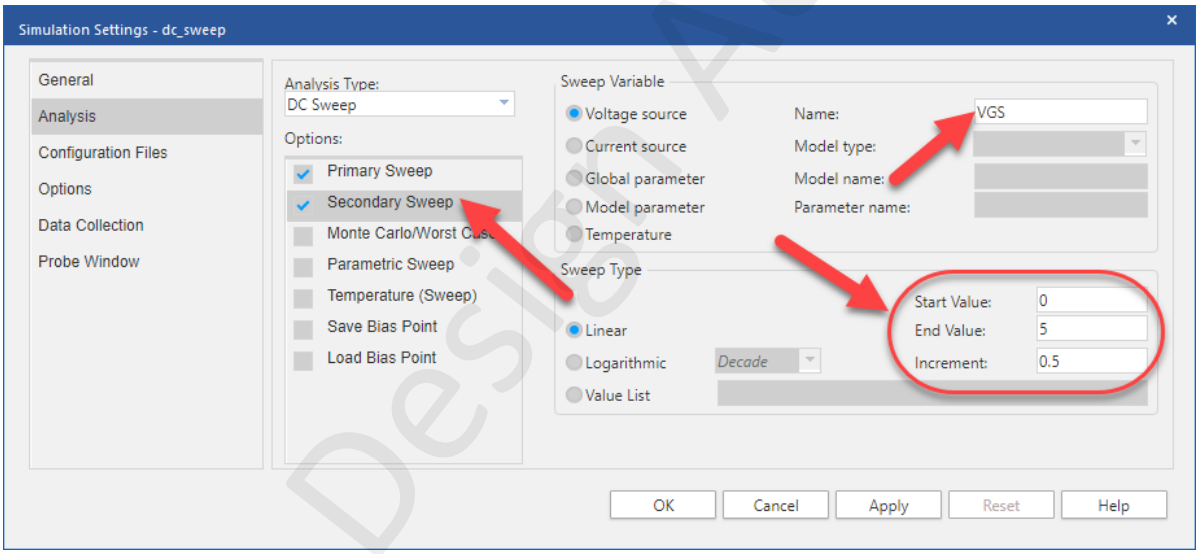
1. Create a new PSpice project called `mosfet_test` based off a blank project
2. Place 2 DC Voltage Parts, the MBreakN3 part and PSpice Ground
3. Rename the reference designators for the two voltage parts to VDS and VGS
4. Wire up the design like this:



5. Create a new profile (***PSpice > New Simulation Profile*** or  icon) called `dc_sweep`
6. For the Analysis Type, drop down **DC Sweep** and for the Primary Sweep, sweep the voltage source **VDS** from **0V** to **10V** in increments of **0.1V**



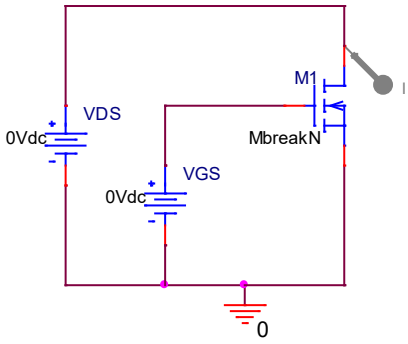
7. In the Simulation Profile, also check off the **Secondary Sweep** and make the name of the voltage source being swept **VGS** and the range from **0V** to **5V** in **0.5V** increments



8. Hit **OK** to close the Simulation Profile window

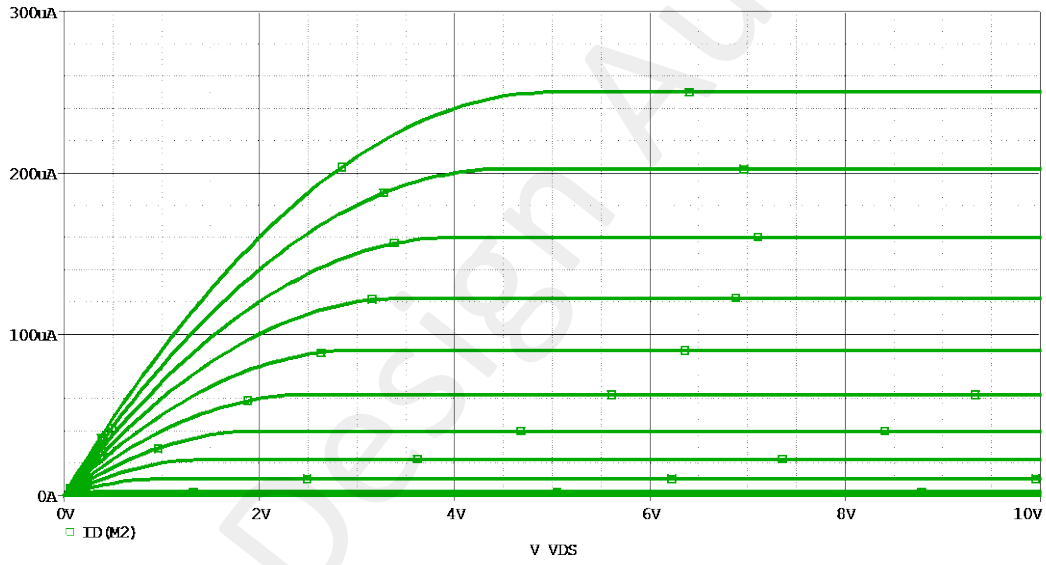
Run the Project

1. On the schematic, Add a current probe to the drain pin on the M1 MOSFET



2. Run the simulation (***PSpice > Run***, icon or F11 key)

What this shows us in the I-V curves for this MOSFET and is a great way to test your part to verify that the behavior that you see here matches what your datasheet says your part should do.



Check for Rapid Transitions

If we want to detect signals that have rapid transitions, we want to measure the slope of the traces and check that for high values. We can do this easily in PSpice by taking a look at the derivative.

1. In the Probe window add a new plot window (*Plot > Add Plot to Window*)

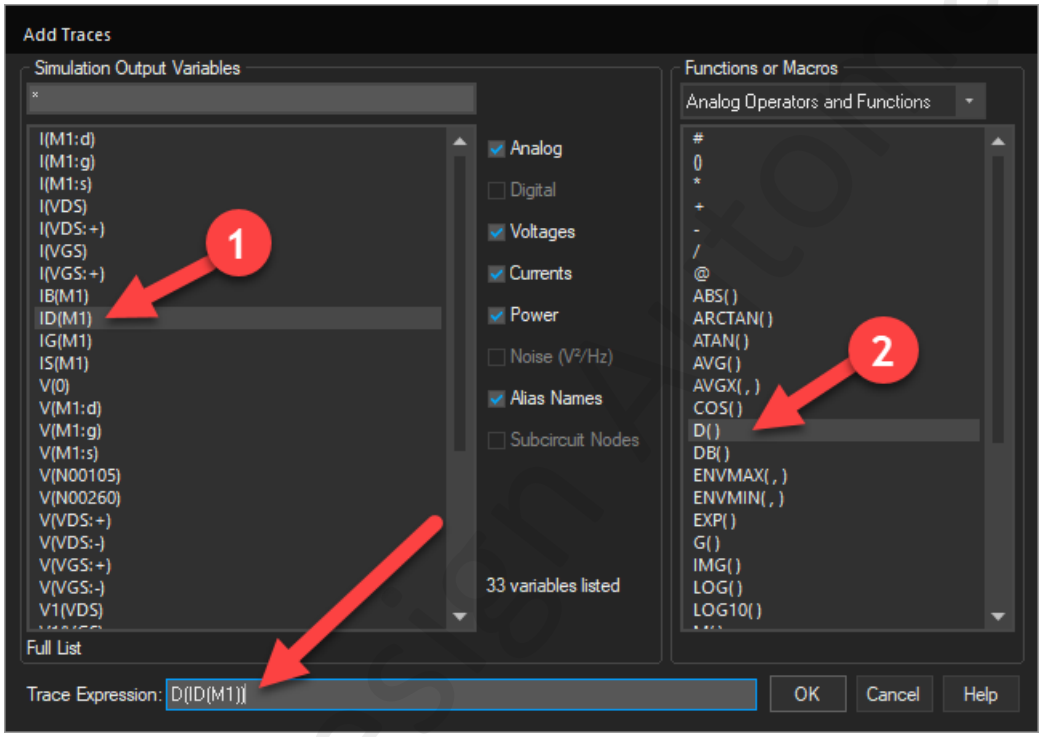
Add a new trace on the new plot to show the derivative of the current through the drain pin of M1

2. ***Trace > Add Trace*** to bring up Add Traces dialog

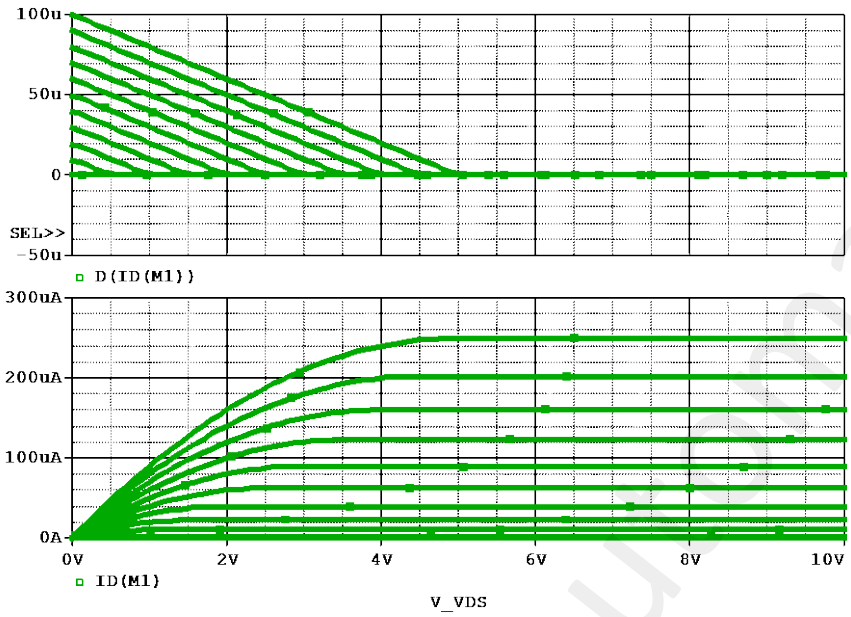
3. On the right hand side of the window in the Functions or Macros section, **left mouse button click** (LMB) on **`D()`** which is used to calculate the derivative

4. Without doing anything else, LMB on the name of the signal that you want to show the derivative for on the left hand side which is ID(M1) in our case (current through the drain pin of M1)

5. The ***Trace Expression*** field at the bottom should now read `D(ID(M1))`



6. Looking at the new trace, notice the large changes in the derivative along each curve at the lower voltages. These types of transitions can be a source for convergence errors.



Optional: Test a different MOSFET model

The model that was tested above was a very simply modeled MOSFET. If you want to try something a bit more complex, try replacing the MBREAKN3 part with a MOSFET (Place > PSpice Component… > Search…) from the Discrete > MOSFET folder and see how the results change.

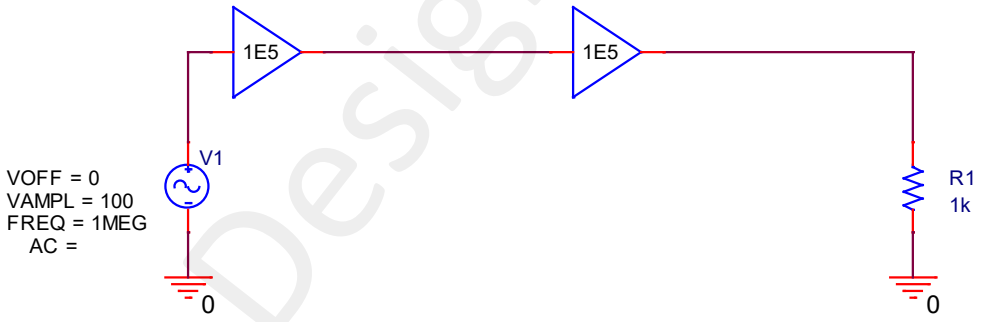
Lesson 7, Lab 4: No Limit

Lab Objectives

- After you complete this lab you will be able to solve limit problems.

Create a New Project

1. Create a new PSpice project called `NoLimit` based on a blank project
2. Place **Sine Wave Voltage Source**, a **Resistor**, **PSpice Ground** and two parts called **GAIN** (there are more than one GAIN parts that come up but we're using the first one which is from the `abm` source library)
3. Make the Sine Wave Voltage Source **100V** in Amplitude and with a frequency of **1MEGHz**
4. Change the **GAIN** block to have an amplification factor of **1E5** instead of 1E3
5. Wire them together like this:



What we've done here is made a sine wave voltage source that is being amplified by a LOT. This isn't very realistic to have such radical amplification without any feedback controls or limiting but it is going to show us a new convergence error in PSpice due to the vast dynamic range between the smallest and largest voltages being solved for.

6. Create a new simulation profile called `tran` with a **Run to Time** of **10us**

Run the Simulation and Examine Results

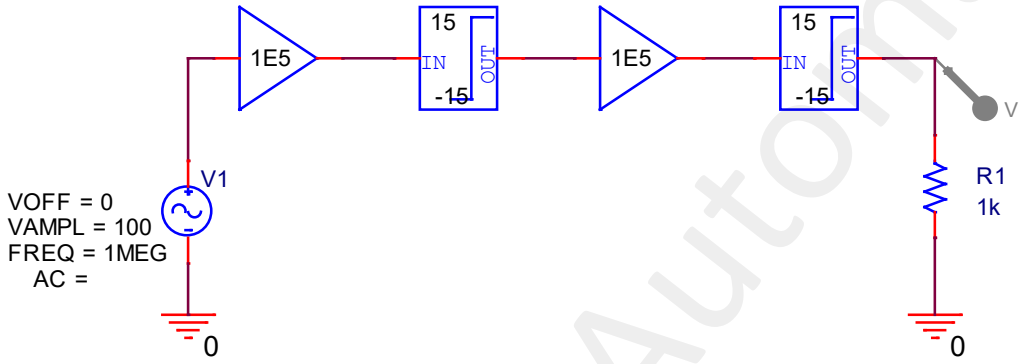
1. Run the simulation (***PSpice > Run***, icon or F11 key)

A convergence error occurs where a source does not converge.

2. **Cancel** the simulation, **close PSpice** and return to Capture

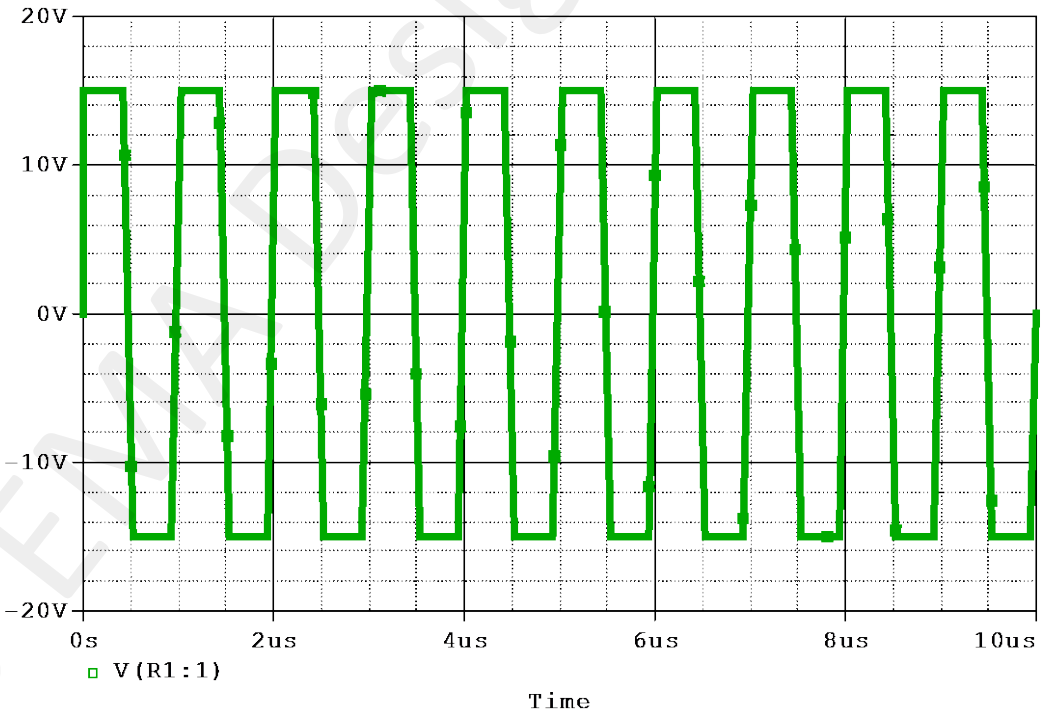
3. Add a device called **LIMIT** after each of the controlled sources

4. Modify the upper and lower limits from 0 and 10 to **-15** and **15** for each occurrence of the limit block. You can do this double-clicking on the properties directly on the schematic and changing the value.



The limit devices crops anything above the high value and below the low value to reduce the dynamic range of the overall simulation to something more reasonable.

5. Rerun the simulation. Observe the voltage across the load resistor, R1



The unlimited controlled sources sharply increase and have a huge dynamic range. This is a common situation when using ABM (Analog Behavioral Model) devices and a potential source of convergence problems.

6. Save and close all open projects if not done so already.

Lesson 8. Transformers

Linear Transformers

Linear transformers are created in OrCAD Capture and PSpice A/D using multiple occurrences of the common inductor symbol L and the K_Linear symbol. Both symbols are located in the Analog symbol library. Transformers can also be created in PrCAD Pspice by using the Modeling Application.

To create a linear transformer:

1. Draw a schematic and assign reference designators to the inductors. (Reference designators may be automatically annotated as you place the inductors).

Reference designator values assigned to the coupled inductors Li fields are (i=1,2,…6). Be sure to specify the value for the Coupling property.

> Linear transformers are based on the SPICE coupling device K and must couple two or more inductors. Values of the inductors are measured in Henry's.

2. Place a copy of the K_Linear symbol anywhere on the schematic and edit its properties.

Below is an example of a PSpice A/D netlist for a simple linear transformer circuit.

```
I1 1 0 AC 1mA
L1 1 0 10uH
L2 2 0 10uH
R2 2 0 .1
K12 L1 L2 1
```

Linear Transformer Circuit

To complete the circuit using the "dot" convention, place a "dot" on the pin 1 of each inductor.

The polarity of the current is determined by the order of the nodes in the inductors, not by the order of the inductors in the coupling statement.

The coupling value is the coefficient of mutual coupling and must have a value between 0 and 1. The equation for coupling is:

```
Coupling = Mij/(Li*Lj)1/2
where:
Li and Lj are a coupled-pair of inductors
Mij is the mutual inductance between Li and Lj
```

For transformers of normal geometry, the coupling is usually 1. Values less than one occur for air core transformers when the coils do not completely overlap.

The linear branch relation for transient analysis is:

```
Vi = Li * dIi/dt + Mij * dIj/dt +Mik *dIk/dt + …
```

Non-linear Transformers

PSpice A/D supports the simulation of non-linear transformers. A transformer becomes non-linear if the coupling statement contains a reference to a core model. Other than a reference to a model, creating a non-linear transformer is no different than creating a linear one.

There are four behavioral differences between a linear and non-linear transformer:

- The magnetic core's B-H characteristics are analyzed using the Jiles-Atherton and Tibrizi core models.
- The inductors become windings, so the value on the inductor symbol is the number of turns.
- The list of coupled inductors may number only one.
- A model statement is required to define the referenced core model.

Non-linear transformers are created using the Kbreak symbol to couple inductors in the design. The inductor is a winding whose value equals the number of turns. The non-linear transformer may couple a single inductor.

Below is a netlist example of a non-linear transformer:

```
L1 5 9 20; inductor with 20 turns
K1 L1 1 K528T500_3C8; Ferroxcube toroid core
L2 3 8 15; Primary winding with 15 turns
L3 4 6 45; secondary winding with 45 turns
K2 L2 L3 1 K528T500_3C8; another core (not the same as K1)
```

Core models can be found in the magnetic library.

Coupling Inductors

The above circuit shows the settings for a non-linear transformer.

For more information on the PSpice A/D format of the Jiles-Atherton model please see the PSpice Reference Guide.

For more information on the Jiles-Atherton model please see D.C. Jiles and D.L. Atherton, "Theory of ferromagnetic hysteresis," Journal of Magnetism and Magnetic Materials, 61, 48 (1986).

Also see PSpice A/D application note PSPA021L: Using Coupled Inductors and Inductor Cores. This is available at http:\\www.pspice.com in the Help Online section.

Core Simulation

The B-H curve shows the relationship between magnetic flux density B and magnetic field strength H for a magnetic core. The B-H curve is often provided by the manufacturer's datasheet.

Hysteresis Loss

The area inside the B-H loop represents the energy loss as the core is magnetized in each direction in an AC power supply circuit. This energy loss is called the Hysteresis Core Loss and should be minimized.

Lesson 8, Lab 1: Linear Transformer

Lab Objectives

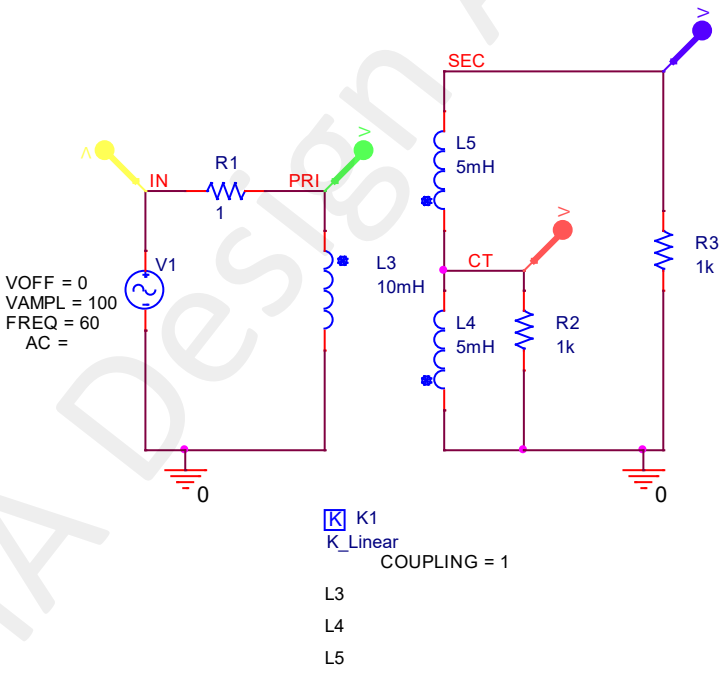
- In this lab you will create a center-tapped linear transformer.

Constructing the Circuit

1. Create a **new project** called `K_Linear` based off a blank project
2. Draw and wire the circuit shown below using the **R**, **L**, **0**, **VSIN**, and **K_Linear** symbols.

For Linear Transformers, the inductors are specified in Henry's, later, they'll be specified in turns.

3. Configure the ***VSIN*** to have an ***offset*** of ***0***, ***Vampl = 100*** and ***Freq = 60***.

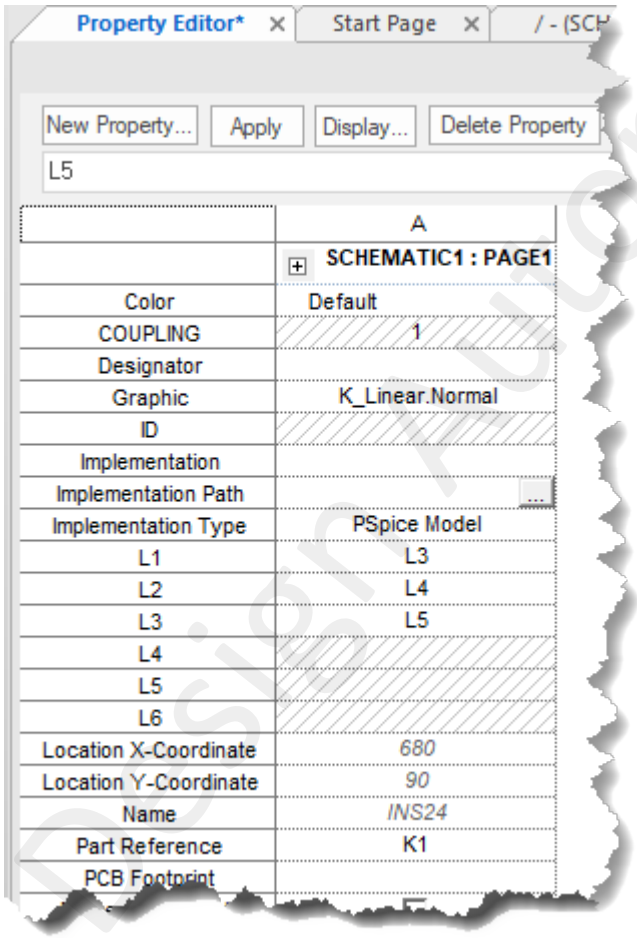


Configuring the Coupling Symbol

1. Double-click on the ***K_Linear*** symbol to edit its properties.

Specify the inductors to be coupled by entering their reference designators into the L1 – L6 fields. L1 – L6 will contain the reference designators of the inductors to be coupled, not the values of the inductors. The inductor values will still be specified on each individual inductor, as normal.

2. Type **L3** in the ***L1*** property, **L4** in the ***L2*** property, and **L5** in the ***L3*** property (based on your reference designators matching the original circuit).



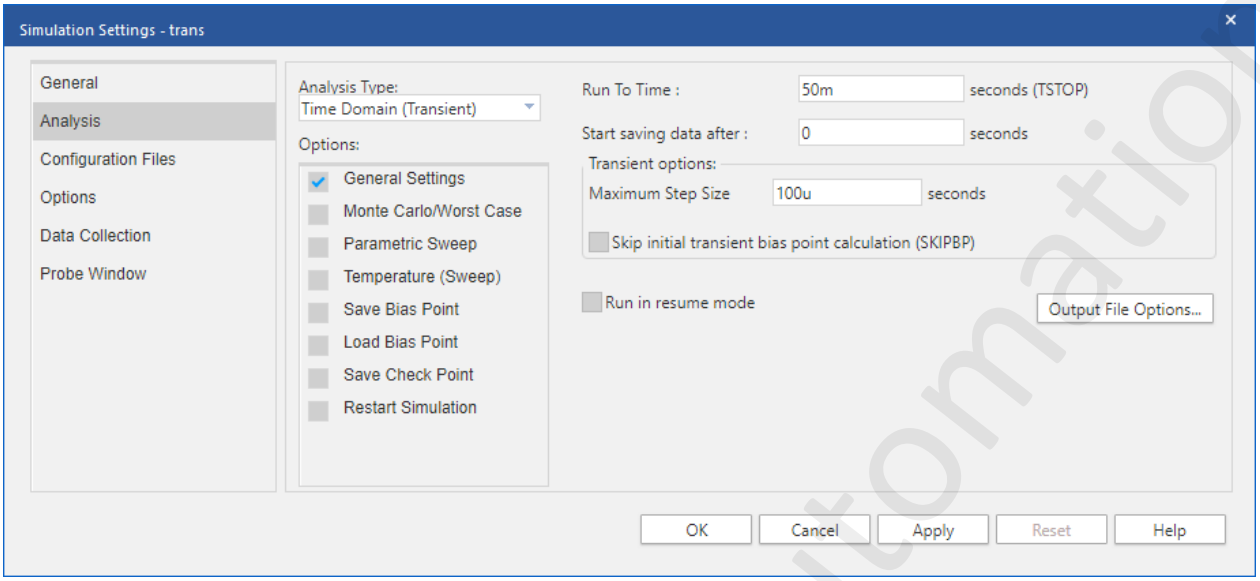
3. If desired, make L1-L3 visible by selecting property values L3 through L5, choosing Display… and picking Value Only.

You may do this through either the Property Editor or the Display Properties dialog box.

4. Close the ***Property Editor*** and return to the schematic view.

Configuring the Simulation

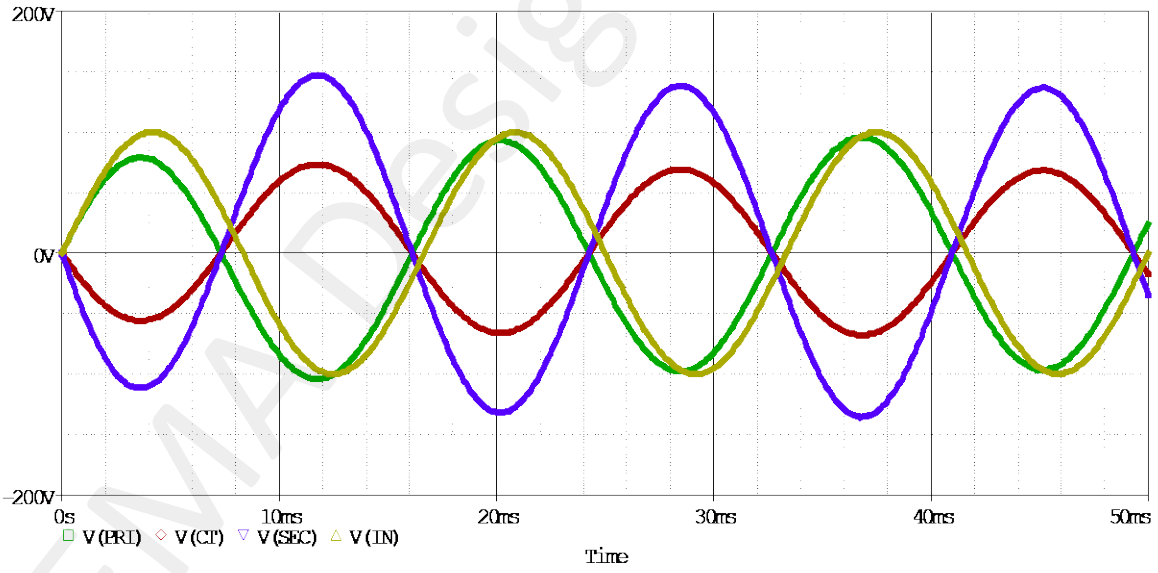
1. Configure a transient analysis with a ***Run to time*** of ***50m*** and a ***Maximum step size*** of ***100u***.



2. Click ***OK*** to close the ***Simulation Settings*** dialog.

3. Run the simulation (***PSpice > Run***, icon or F11 key)

4. Examine the results in the Probe window. The following figure shows the expected results.



From the cursor display you can see that the value on the center-tapped node is exactly half of that on the end.

It may seem odd that the secondary voltage is higher than the primary voltage but that is because the inductors are measured in Henry's instead of turns.

The Voltage Ratio (VR) is equal to the Turns Ratio (TR) which is equal to the square root of the Inductance Ratio (IR).

```
VR = TR = SQRT(IR)
```

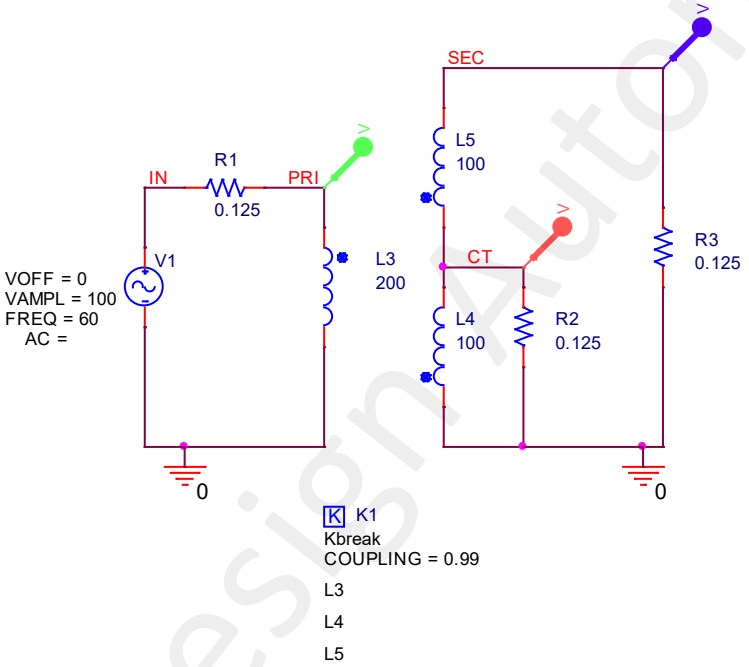
At the center tap, the IR is 10mH to 5mH or 2:1, the square root of that is sqrt(2) which yields the Turns Ratio and the Voltage Ratio. To see it from the secondary's perspective, invert the ratio to get 1/sqrt(2) or 0.707 at the center tapped secondary compared to the primary which is what we are seeing. The full secondary is double the center tap, so its voltage is double the center tap as we're seeing in the simulation results.

Lesson 8, Lab 2: Non-linear Transformers

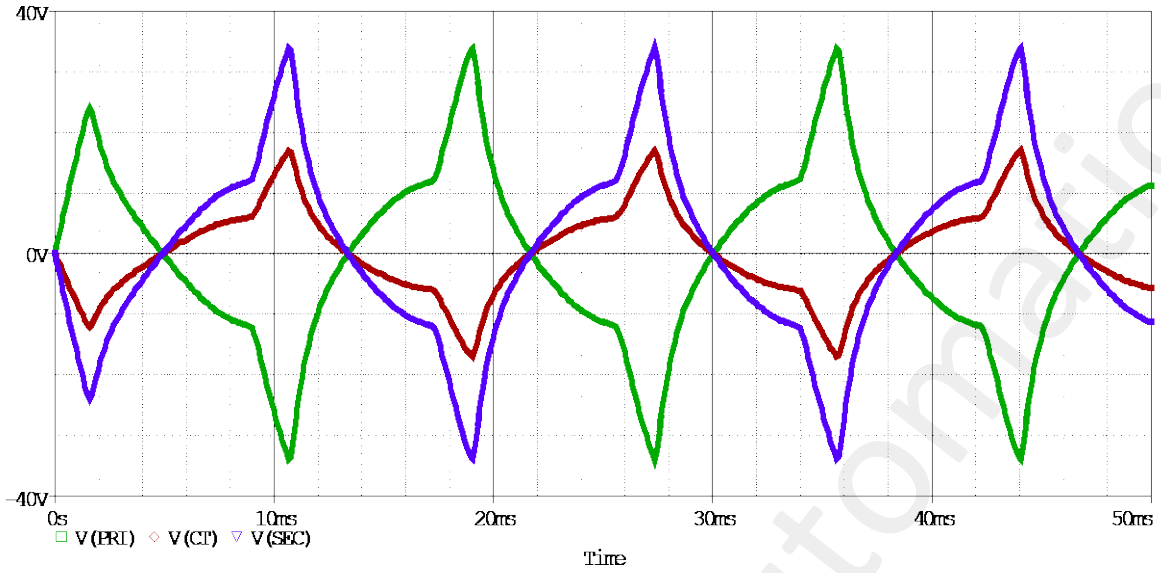
Lab Objectives

- In this lab you will create and simulate a nonlinear transformer.

Create a Non-linear Transformer



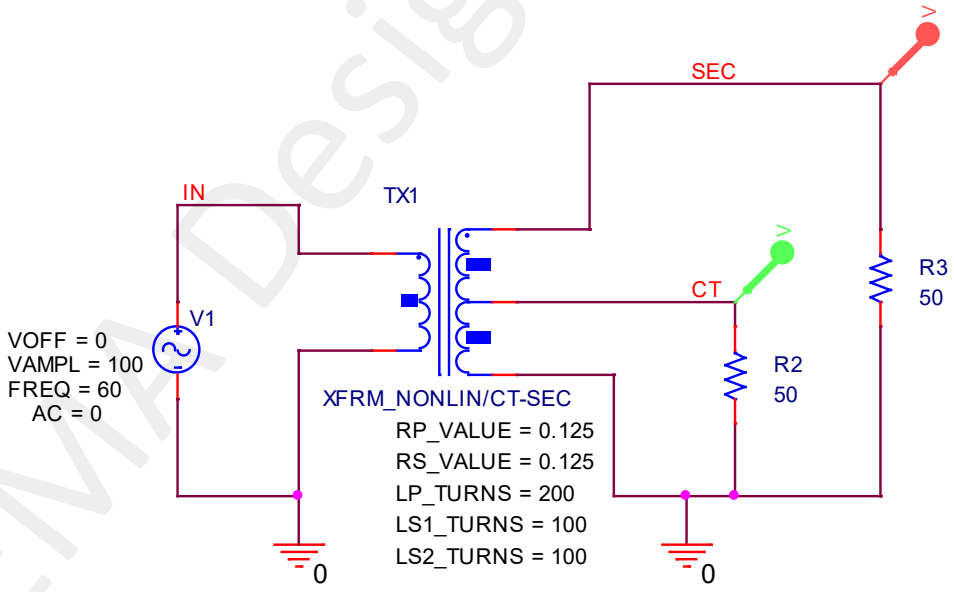
1. Using the same circuit as in the last lab, replace the K_Linear symbol with ***Kbreak***.
2. Configure the L1-L3 properties as they were for the K_Linear (***L1=L3, L2=L4, L3=L5***). You may need to flip the inductor symbols to have the correct orientation.
3. For ***Coupling*** type ***.99***.
4. Change the values of R1, R2, and R3 to ***.125***.
5. Change the values of the inductors to ***200***, ***100*** and ***100*** for L3, L4 and L5, respectively.
6. Run the simulation (***PSpice > Run***, icon or F11 key)
7. Examine the results in the Probe window. You should now see that the core model is saturating.



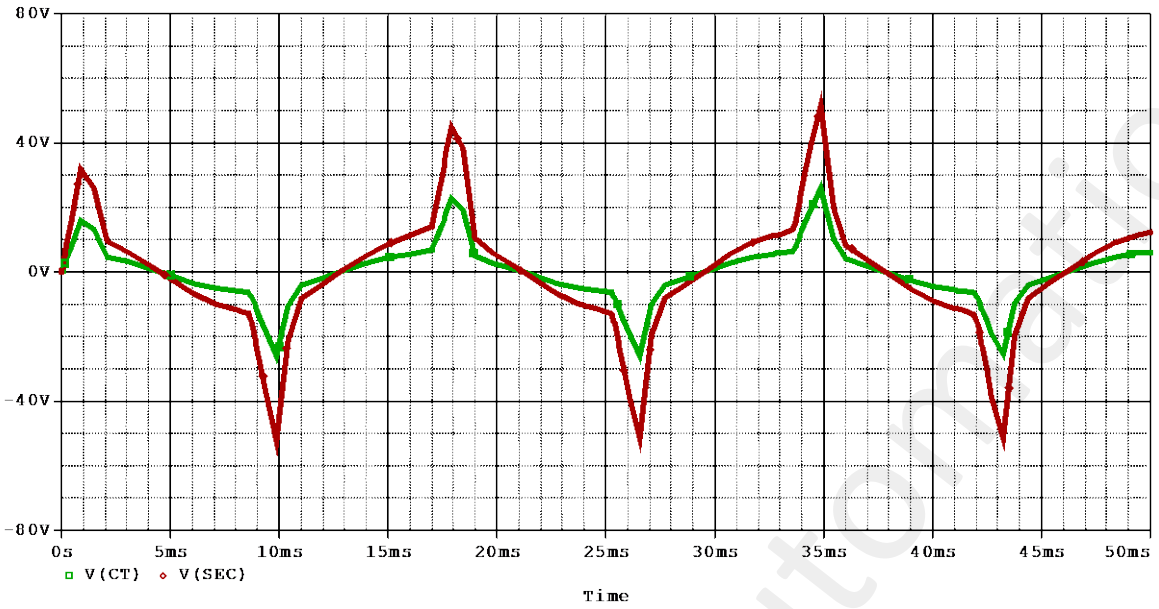
Optional Exercise: Using Pspice Breakout Library

There are specific center-tapped symbols added to the 9.2 breakout library; Xfrm_Nonlin/Ct-Pri, Xfrm_Nonlin/Ct-Pri/Sec, and Xfrm_Nonlin/Ct-Sec.

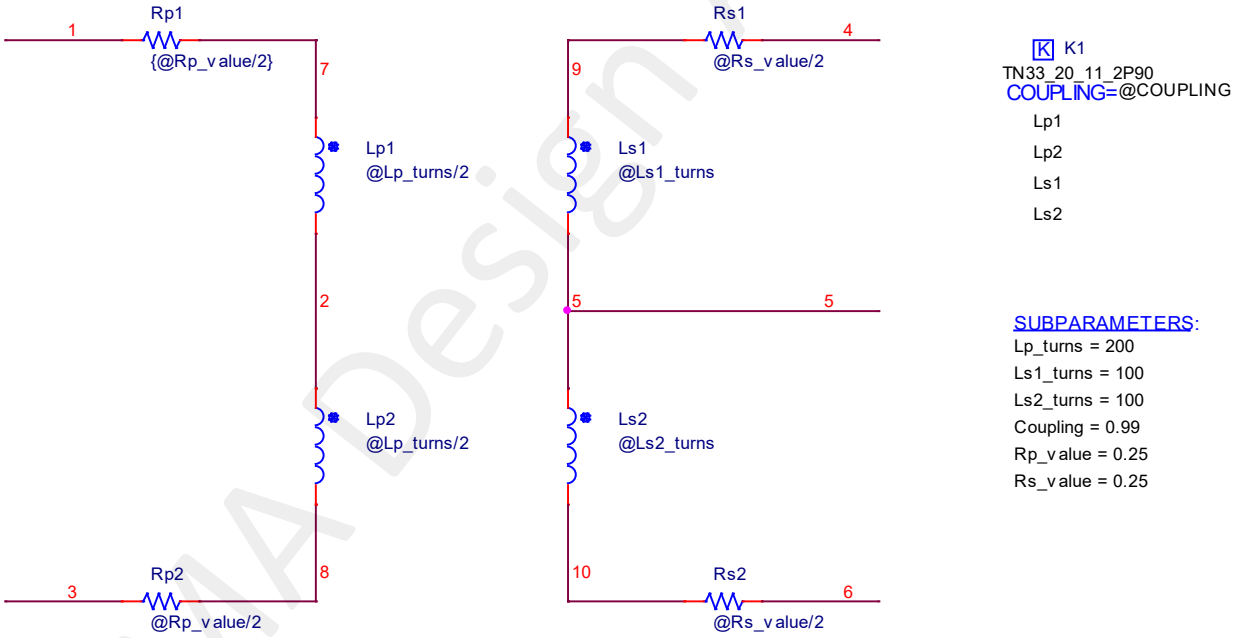
Modify the above circuit using XFRM_NONLIN/CT-SEC.



Here are the results:



Examine the subcircuit definition (**RMB** on the part in Capture and choose **Edit PSpice Model**). It is the text equivalent of this circuit:

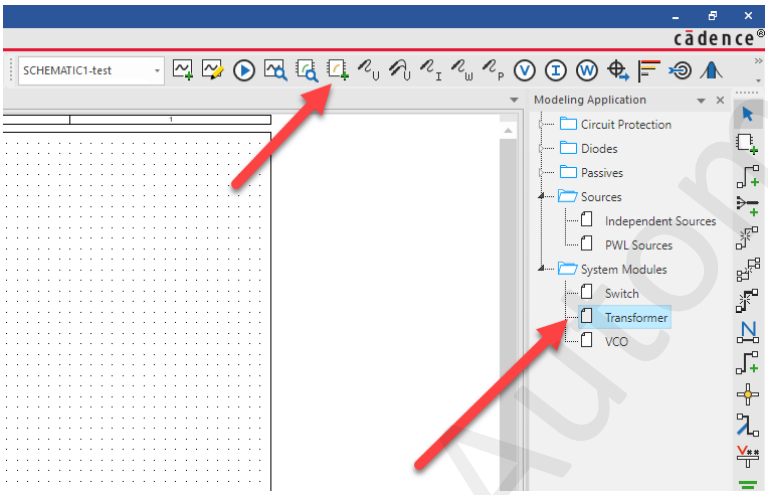


This schematic graphic was generated by hand (to aid in understanding) based on the subcircuit and cannot be automatically created by any tool inside of PSpice.

The benefit of placing all your parts individually instead of as a complete part like this are that you can more easily modify the internal workings of the part such as the polarity of the windings and the core itself.

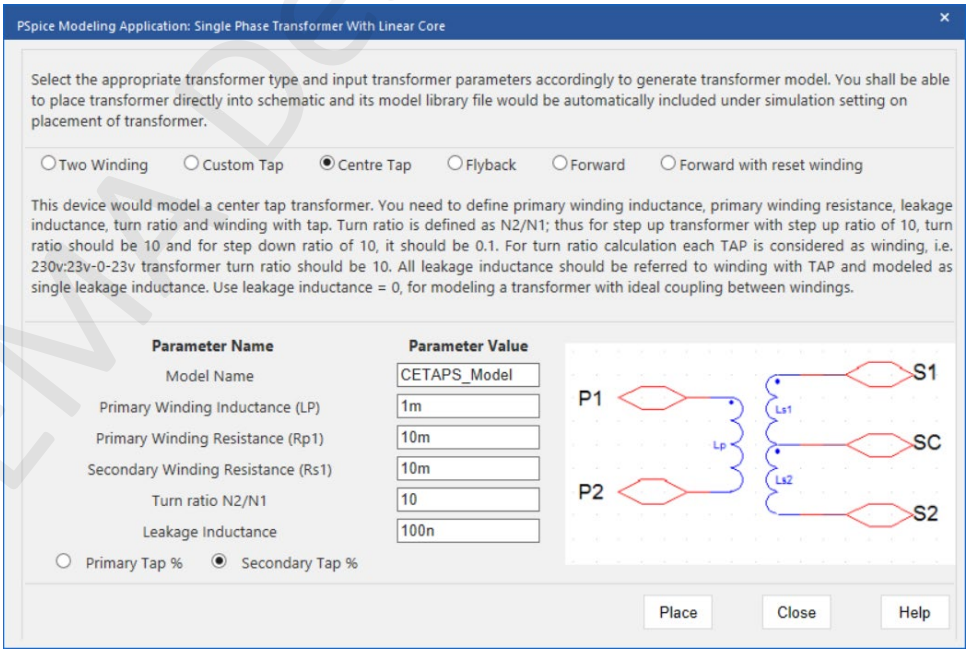
Optional Exercise: Using Modeling Application

The Modeling Application contains several types of transformers. The modeling application can be accessed from Place>Pspice Component>Modeling Application or by pressing the Modeling Application Icon. The Modeling Application panel will open.



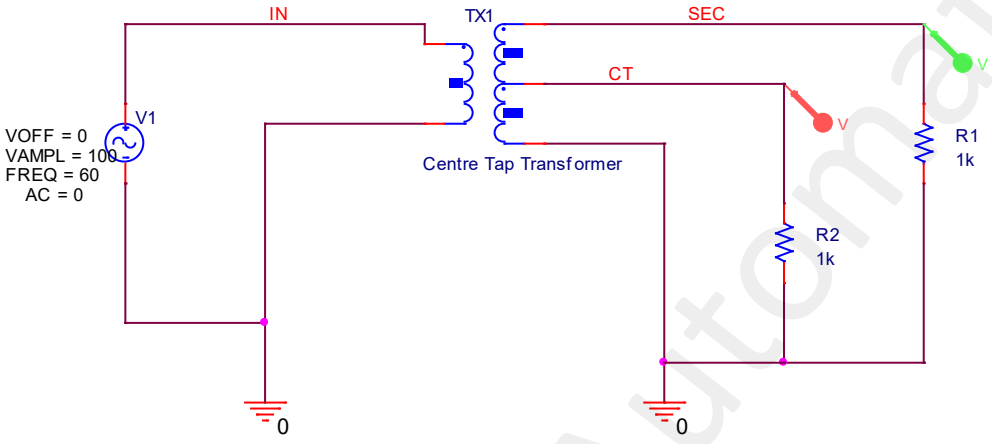
The folders can be expanded to see the types of models contained. Under the system Modules folder, there is a transformer option. Click **Transformer.** A new window will pop containing several different types of transformers; Two Winding, Custom Tap, Centre Tap, Flyback, Forward and Forward with reset winding. Once the type of transformer is selected enter the desired parameters.

In this lab we will use the default settings listed in the picture below.

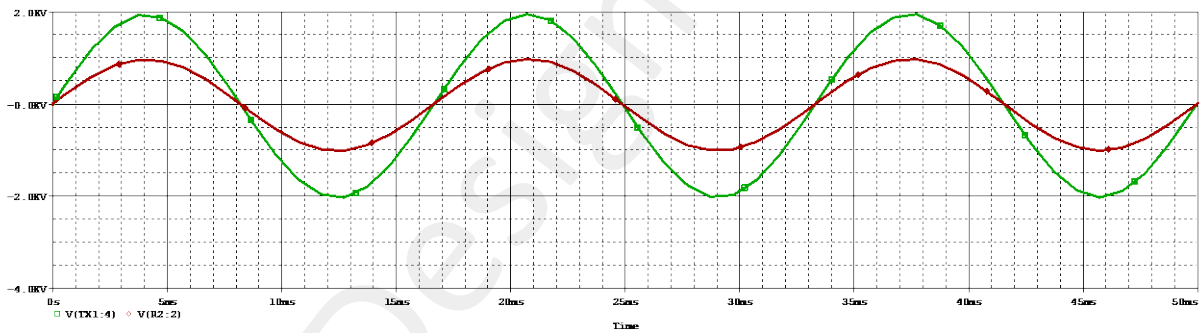


Once the desired parameters are enterered, press Place. The Modeling Application will generate a symbol with the inputted parameters.

Modify the previous circuit and place your new symbol in your schematic. Run your transient analysis.



Here are the results:



Lesson 8, Lab 3: B-H Curve and Hysteresis Loss

Lab Objectives

In this lab you will characterize a magnetic core "P11_7_I_3C85" in a buck converter design by plotting the B-H curve and calculating the hysteresis core loss.

Create a new Buck Converter Project

If you're running behind, feel free to open and run the completed version of this schematic from the `Solutions` folder and skip ahead a few pages to the **Plot the B-H Curve** section. If you're on a good pace, feel free to go through the exercise of putting this all together.

1. Create a **new PSpice project** called `buck_converter` based off of a blank project

2. **Place these parts**:

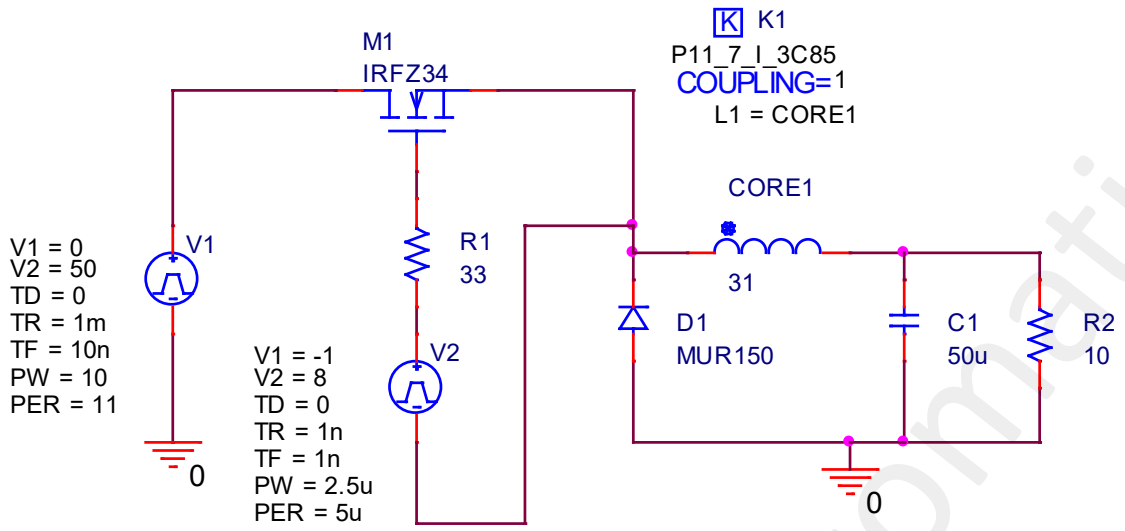
- 2 Voltage Pulse Parts
- 2 Resistors, 1 Inductor, 1 Capacitor
- IRFZ34 Power MOSFET (there are 2, we'll take the one from the powermos library)
- MUR150 fast recovery diode (there are 2, we'll take the one from the diode library)
- P11_7_I_3C85 core model
- PSpice Ground

3. **Rename** the **reference designator** for the **Inductor** from L1 to **CORE1**

4. **Edit** the **L1** property on the **K1** part to be equal to **CORE1** and **Display** it on the schematic

This couples the CORE1 inductor part, whose value is in turns to the K1 coupling part.

5. **Wire up the design** and **modify** the **component values** and **voltage sources** to equal this:



The Core GAP property has not been specified on the default model that we're using so we're going to edit the model to put this data in. From the datasheet we can see that this value is 500um. PSpice will want this in units of cm which will be 0.05cm.

GRADE	A_L (nH)	μ_e	AIR GAP (μm)
3C81	63 ±3%	≈ 43	≈ 500

6. **Select** the **K1** part by selecting the box around the letter K and Right Mouse Button Click (**RMB**) to select **Edit PSpice Model**

7. After the line that contains PATH, **add another line** with `GAP=0.05`, it should look like this:

```
* P11_7_I_3C85 CORE model
* created using Model Editor release 9.1 on 06/25/99 at 11:50
* Model Editor is a Cadence Design Systems product.
.MODEL P11_7_I_3C85 CORE
+ MS=379.49E3
+ A=23.138
+ C=78.870E-3
+ K=18.787
+ AREA=.19
+ PATH=1.6300
+ GAP=0.05
```

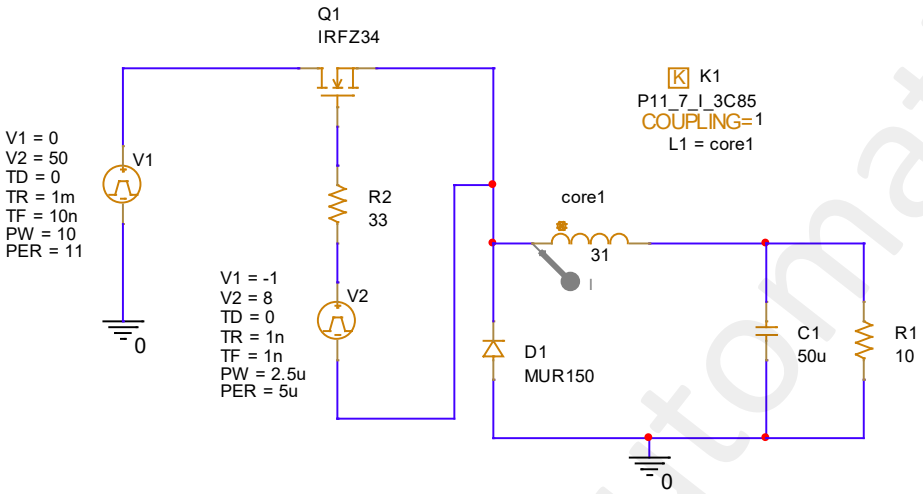
8. **Save** the model and **close** the PSpice Model Editor

Run the Simulation

1. **Examine** the design and how it acts as a buck converter

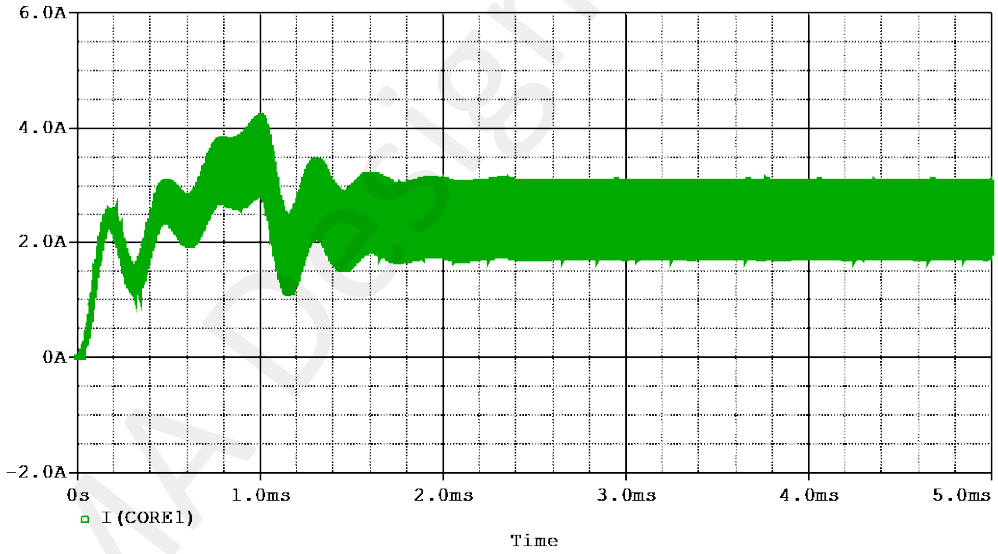
2. Create a **new simulation profile** called `trans` that runs a **transient** simulation to **5ms**

3. Put a **current probe** on the left pin of the CORE1 part



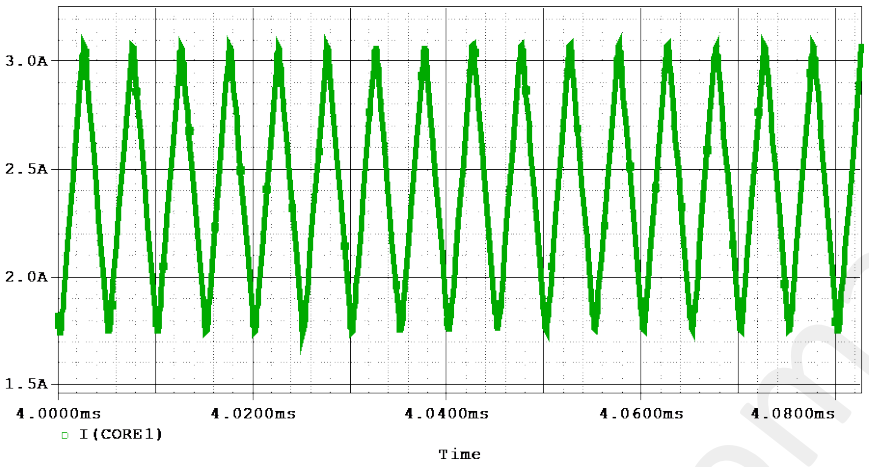
4. Run the simulation (***PSpice > Run***, icon or F11 key)

5. **Look** at the inductor current **I(CORE1)**. Note that steady state is reached at ~4ms.



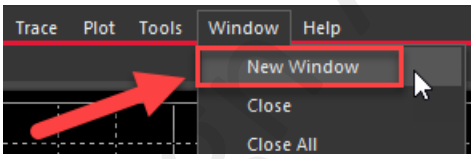
6. **Zoom after 4ms** to see the inductor current closely

Notice that the core is not saturating. There is no flattening (clipping) observed in the inductor ripples. This is the necessary condition for meaningful calculations of core-loss.



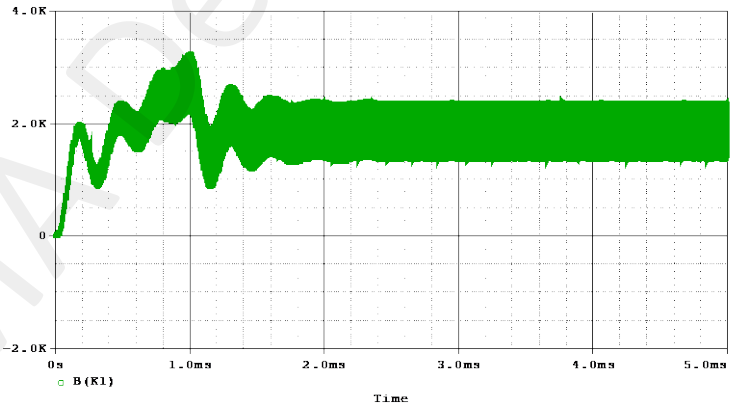
Plot the B-H Curve

1. We are going to plot the B-H hysteresis curve from our transient simulation data. **Create a new window** (becomes a tab at the bottom) for this plot.

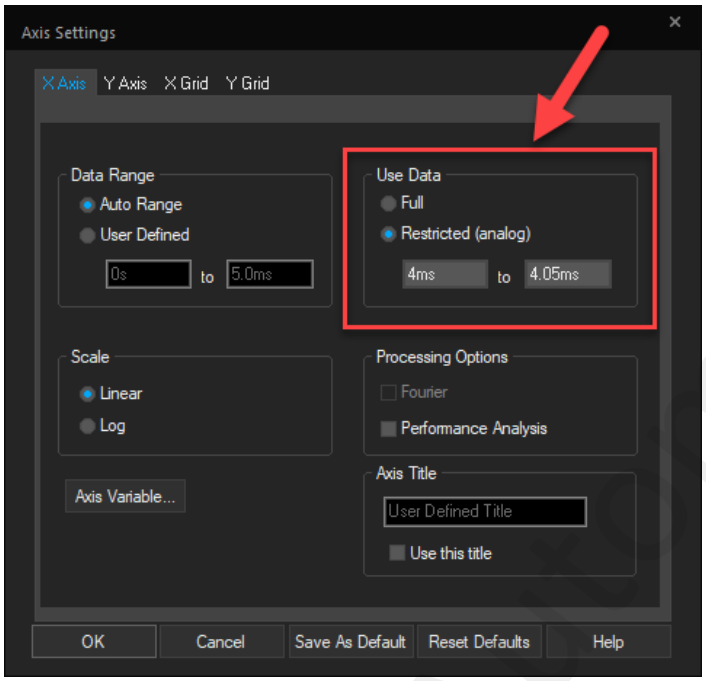


2. Go to ***Trace > Add Trace***. In the Add Traces dialog box, click on **B(K1)** and hit **OK**

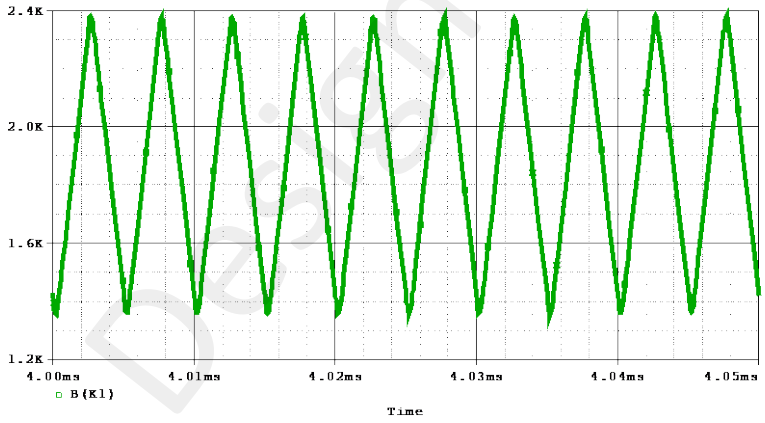
This plots B(K1) on the Y-axis with time on X-axis.



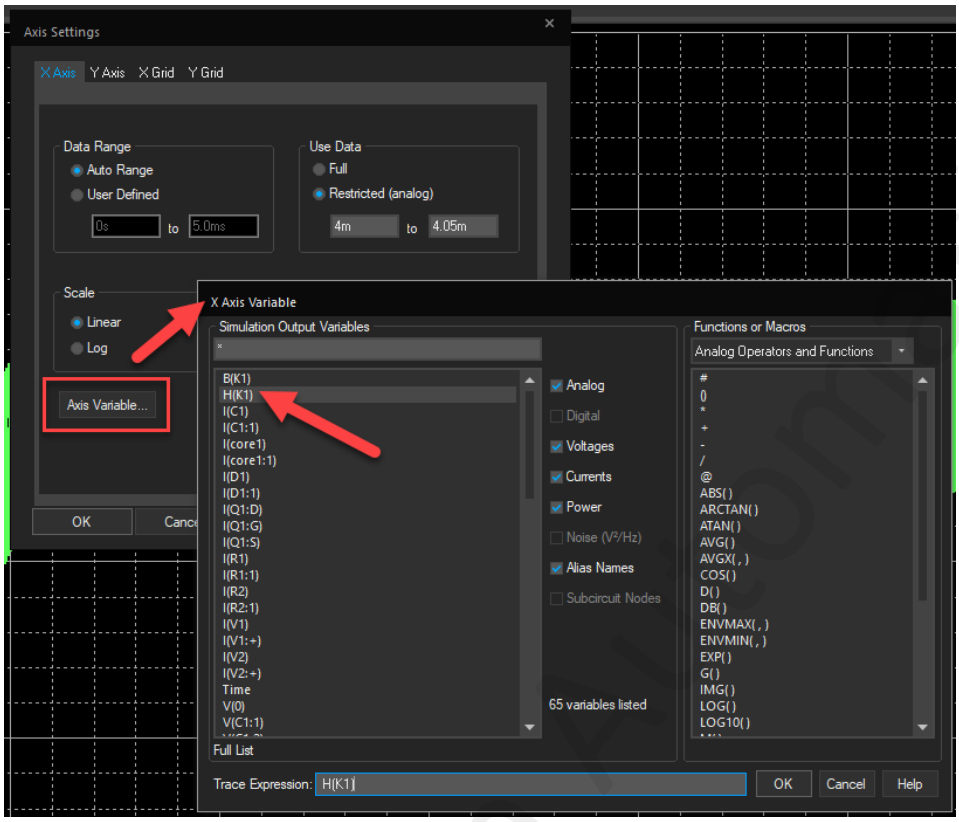
3. In Probe, go to ***Plot >Axis Settings***. Now we want to **restrict** the **X-axis range** so we can see in more detail a few cycles of B(K1). This is done on X-axis settings as shown below. Restrict it between **4m** and **4.05m**.



4. Click **OK** to close the Axis-setting window. B(K1) is plotted as below.

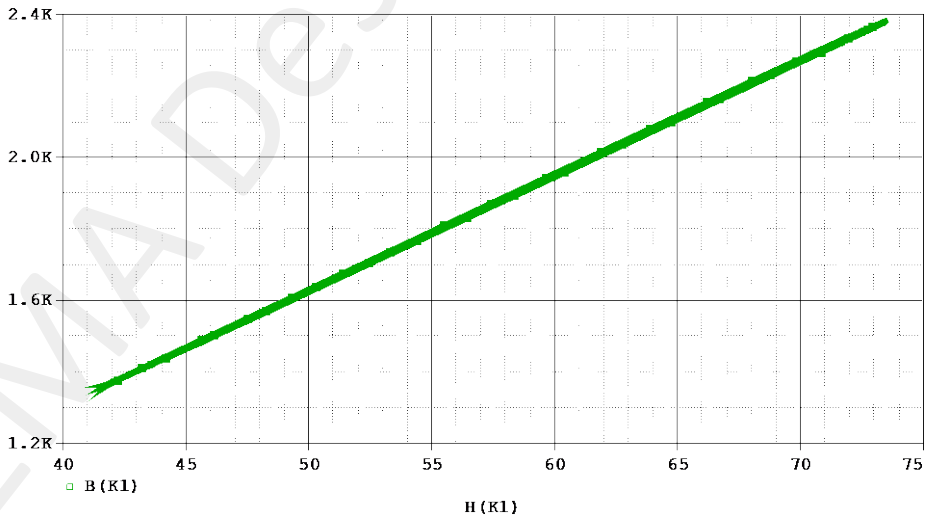


5. Going back to the **Axis Settings**, we now change the X-axis variable from Time to **H(K1)**. Click on the **Axis Variable button** in the bottom left of the window to set the magnetic field strength H as the x-axis variable.



6. Click **OK** to close the Axis Settings window

You should see the plot of the B-H curve with B on Y-axis and H on X-axis.



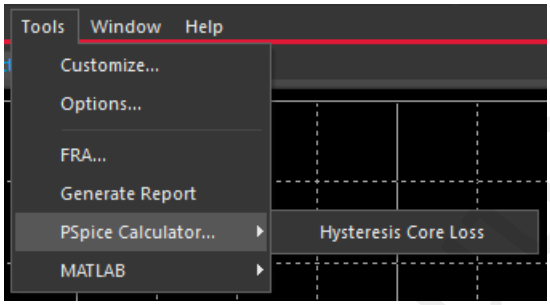
This B-H is fairly narrow, meaning that there should not be much hysteresis core loss as the core charges from negative to positive saturation and back. We will calculate this loss value in the next exercise.

Calculate the Hysteresis Core Loss

We will continue working off the previous design and simulation data.

PSpice has a built-in application to solve for the hysteresis core loss by solving for the area inside the B-H curve

1. To launch this, go to ***Tools > PSpice Calculator…>Hysteresis Core Loss***



The calculator tool will automatically detect the cores present in the design as well as the inductor frequency

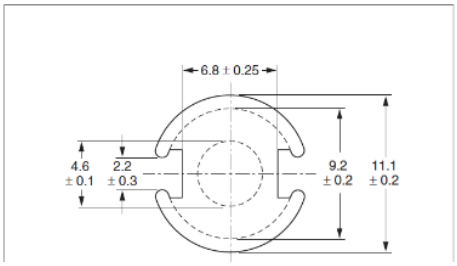
2. Set the **start time** to be **4m** as steady state was not reached before that
3. You can enter the **End time** as **4.05m** but it's not critical
4. We use the manufacturer's datasheet to set the **Core Volume** to be **309**

P cores and accessories P11/7/I

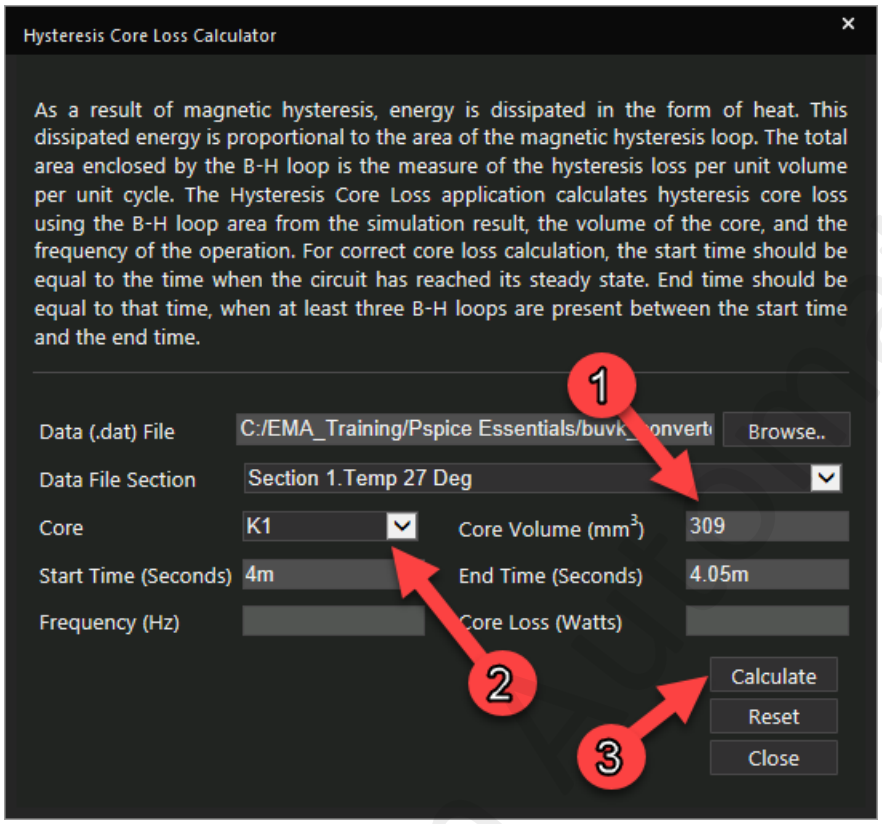
CORE SETS

Effective core parameters

SYMBOL	PARAMETER	VALUE	UNIT
Σ(l/A)	core factor (C1)	0.860	mm^{-1}
V_e	effective volume	309	mm^3
l_e	effective length	16.3	mm
A_e	effective area	19.0	mm^2
A_{min}	minimum area	13.7	mm^2
m	mass of set	≈ 1.9	g



5. Now with all the necessary information provided, Hit the **Calculate** button



6. We see that the hysteresis core loss is calculated to be **0.0972 Watts**

Notice the application automatically calculates the frequency of the inductor current and loop area of hysteresis B-H curve to solve for the total core loss. We see that the frequency is ~200KHz in steady state. If you look at schematic, the pulse switching period is 5us which gives a frequency of 200KHz. Thus the application is correctly calculating the frequency.

We can see that we're in the range of core loss value compared with the datasheet; to be closer, the core needs to be operated at the same conditions as mentioned in the datasheet for core-loss. Typically, these conditions are controlled by the B value and frequency value.

Lesson 9. Parametric Analysis

What Does Parametric Analysis Do?

Parametric analysis is limited to sweeping a single parameter at a time. You can sweep a variable based on a regular progression of values (linear, octave, or decade) or by entering a simple list of values that the parameter is to take on.

A parametric analysis requires at least one other basic analysis type such as AC Sweep, DC Sweep, or transient to be configured and enabled. For each of the simulations enabled, a parametric analysis will cause the simulation to be run multiple times with the new parameter value being applied with each execution. You can think of it as a nested loop. The outer loop is the parametric analysis stepping the variable while the inner loop is the basic analysis type using that variable value.

For now, we will be sweeping a global parameter. Later you will learn how to modify model parameters and then set up the analysis to use them in a parametric or DC Sweep.

Preparing a Design for Parametric Analysis

There are three steps to defining and using a global parameter in PSpice A/D:

1. Define the parameter by using the PARAM symbol which is located in the Special library.
2. Substitute the parameter value with a circuit device.
3. Set up a parametric analysis that sweeps the parameter.

You can define the new global parameter by editing the properties of the PARAM symbol and adding a new property. The button text will change based on how the spreadsheet has been pivoted. Define the new property's name and a default value. The name can be anything that uses the characters that are valid for PSpice A/D. The value will be a default value since the simulation setup will usually be sweeping the value during a simulation.

You can parameterize a device value by replacing the value with a variable or an equation enclosed by curly braces. The braces tell PSpice A/D that the expression or variable must be evaluated at the time of simulation. The expression can be a simple variable like we are using in this lab or can be an involved equation using multiple variables, predefined PSpice A/D functions, or even functions that you have defined using the `.FUNC` command.

Parametric Setup

Parametric set up consists of the following steps:

1. Choose and configure one of the three basic simulation: AC Sweep, DC Sweep, or transient.

2. After setting up the basic simulation, click on the ***Parametric Sweep*** checkbox in the ***Options*** sections of the Simulation Profile.

3. Select the Swept variable type.

The swept variable can be an independent voltage source, an independent current source, a global parameter, a model parameter, or temperature. Dialog box options available vary with your selection. For example, if you select a global variable, then you need to supply a value for the Parameter Name field. That value would be the name of the variable created using the PARAM symbol.

4. Select the sweep type.

The Sweep type offers you a choice of linear, logarithmic, or a value list. If logarithmic is selected, then you will also need to select from the pulldown box whether the simulation will be done in octaves or decades. The Increment field also changes if logarithmic is selected to reflect either points per decade or points per octave in place of increment for a linear sweep.

5. Enter the name of the variable to be swept.

6. Enter the start, end, and increment values.

Viewing Parametric Results in Probe

After the simulation has completed you may add traces in the Probe Window. Instead of seeing a single waveform as in past simulations, you will now be presented with a family of curves. Each trace represents one complete analysis with the chosen variable set to one of its values.

To find the value of the swept parameter for any given run, right-click any trace on the plot or any trace symbol below the x-axis. Then select **Trace Information** from the pop-up menu to view the parameter value for that run.

It is also possible to view only selected traces from the family of curves generated during a parametric analysis. As seen in the picture that follows, you may add a suffix to the trace name to specify a particular trace to view.

The graphic shows traces I(CAP2)@1, I(CAP2)@25, and difference between them. The Trace Expression for each is; I(CAP2)@1, I(CAP2)@25, and I(CAP2)@25-I(CAP2)@1. The number after the "@" symbol refers to the simulation run number, not the parameter's value during any simulation run.

Using Cursors

Each window open in Probe has two available cursors. These cursors act much like the cursors on an oscilloscope or logic analyzer. You can add cursors across multiple traces and plots. The cursor window displays the X and Y values for the different traces. Cursor window also displays maximum, minimum, average and X and Y differences between the two cursors.

The Cursor Max and Cursor Min buttons will snap automatically to the highest and lowest point on the waveform respectively. The Cursor Peak and Cursor Trough buttons are location and direction sensitive, meaning, when you select that button it will start looking for a peak or trough at your cursor location and moving in the direction that you last travelled. This helps you to find local minimum and maximum without snapping to the absolute largest or smallest point (like Max and Min do). These are explained a little more further below.

The first cursor is controlled by clicking and dragging the left mouse button. The second cursor is controlled by clicking and dragging the right mouse button. To assign a cursor to a trace either left-click (for cursor one) or right-click (for cursor two) on the symbol that appears next to the trace name below the X axis.

The toolbar located at the top of the window has several predefined search commands that the cursors can use. These toolbar icons, from left to right perform the following commands:

1. Find the next peak on the trace. A peak is defined as a point that has three points on either side of it with a lesser Y value.
2. Find the next trough on the trace. A trough is defined as a point that has three points on either side of it with a greater Y value.
3. Find the inflection point of the trace.
4. Find the minimum point on the trace.
5. Find the maximum point on the trace.
6. Move the cursor to the next data point on the trace.
7. Enter a custom search command.

8. Find the next transition. This is used only for digital traces.
9. Find the previous transition. This is used only for digital traces.
10. Mark the cursor location.

When you select a search command, the system acts upon the last cursor touched. The search will be executed in the direction that the selected cursor was last moved.

Lesson 9, Lab 1: Parametric Analysis

Lab Objectives

- 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

Create and Define Cval Part

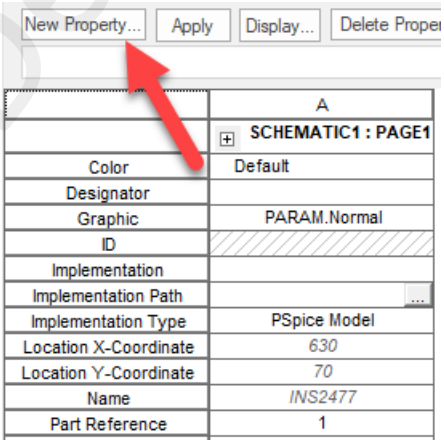
1. Open the ***schematic page*** of the ***Buffer*** project from Lab 5

If you didn't complete Lab 5 or want to start from something new, use the design in the Solutions folder.

2. Place the **Param** part anywhere on the schematic (use Place > PSpice Component… > Search… to find it)

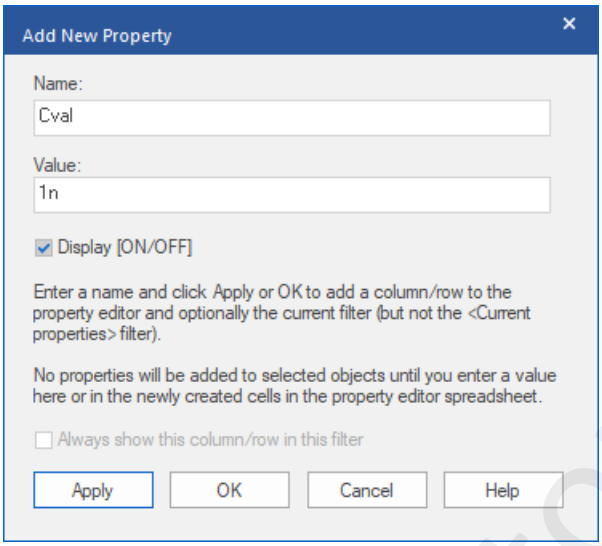
3. Highlight the **Param** symbol and open the Property Editor dialog by hitting **Ctrl+E**

4. Click the ***New Property*** button in the upper left

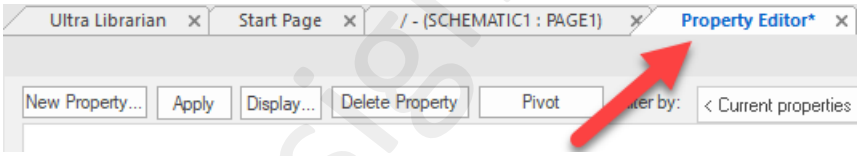


5. Add a property whose value is **Cval**

6. Set the ***Value*** of Cval to ***1n*** and click ***OK***



7. **Select** the **Cval** property in the list of properties
8. Check the ***Display*** [ON/OFF] checkbox
9. Check the ***Name and Value*** radio button, Hit **OK**
10. Close the property spreadsheet by clicking on the X in the ***Property Editor*** tab

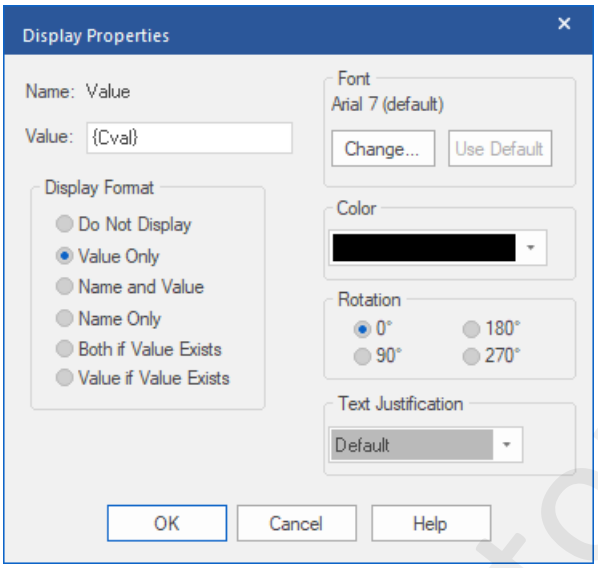


Notice on the schematic that the PARAM part now has Cval underneath it defined as 1n.

This defines the global parameter CVAL, assigns it a default value of 1n, and sets the display to show both the property name and value. At this point it is still not being used by any device and would have no effect on simulation.

Using a Global Parameter as a Property Value

1. Double-click the ***value*** of ***CAP2***. (It is currently ***1n***.)
2. Type **{Cval}** as the new value (the curly braces are required here)



3. Click ***OK***

Any time a model parameter or component value (as in discrete components like resistors, caps, and inductors) is to be assigned at the time of simulation, it is enclosed in curly braces **{ }**. This tells PSpice A/D 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, PSpice A/D will look for a global variable called CVAL and will substitute the current value of CVAL for the value of the capacitor.

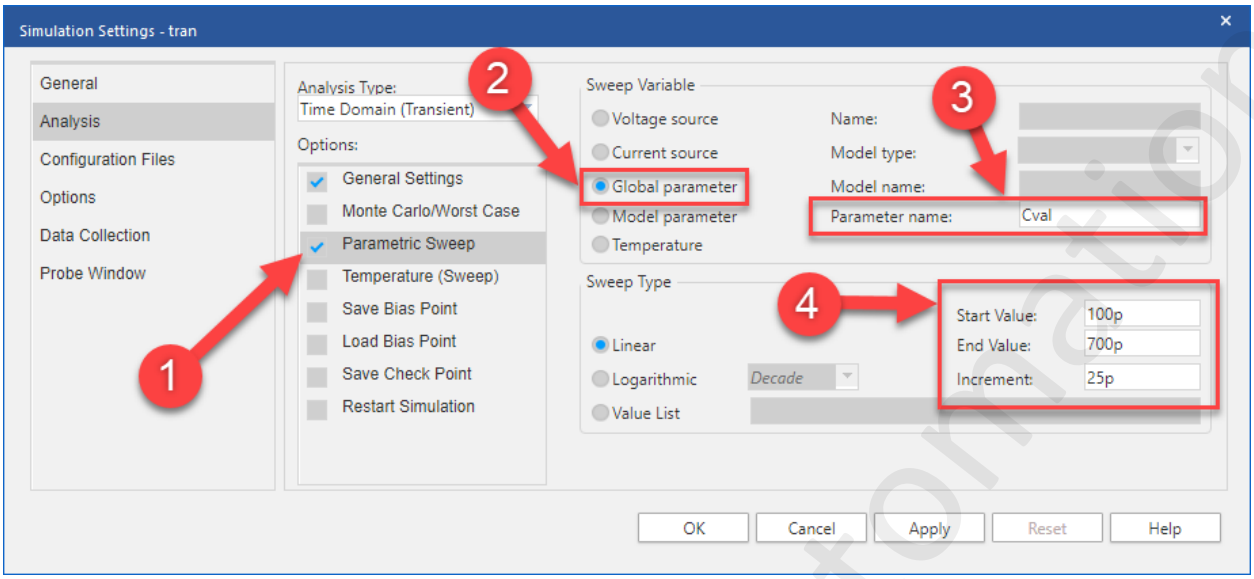
4. **Edit** the **tran** simulation profile or create a new one
5. Confirm that ***Time Domain (Transient)*** is the selected analysis type
6. In the ***Analysis*** tab click the ***Parametric Sweep*** option
7. Check the Global Parameter radio button

This will highlight all of the necessary fields.

8. Type **Cval** in the ***Parameter Name*** field

Note that it is not enclosed in curly braces.

9. Check the ***Linear*** radio button in the Sweep type section of the dialog
10. Type **100p** for the ***Start Value***
11. Type **700p** for the **End Value**
12. The ***Increment*** will be **25p**

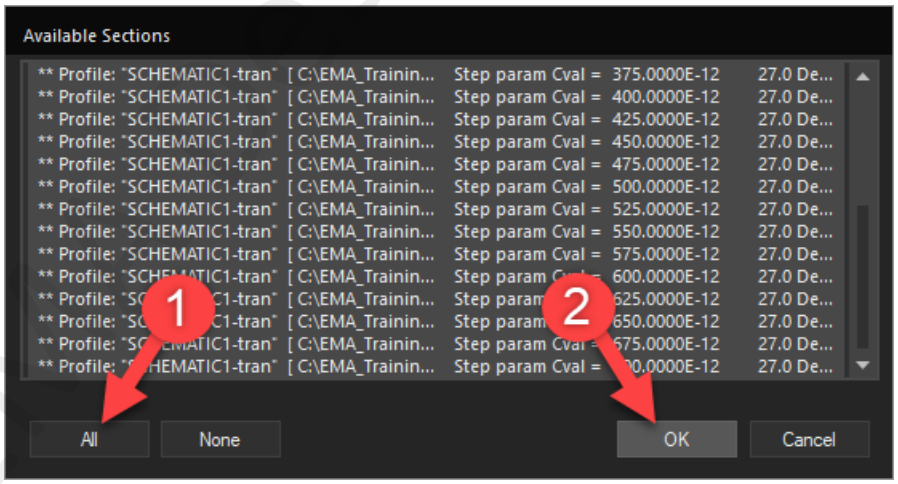


13. Click ***OK*** to save the changes and close the dialog

14. Run the analysis

Viewing the Output in the Probe Window

Since we have 25 runs of the transient analysis, each with a different value of CAP2, we are presented with the Available Sections screen listing all 25 runs and the value of CVAL for each. We are given the option to select which runs we wish to view. Having all selected is the default. Click **OK** to accept the selections.

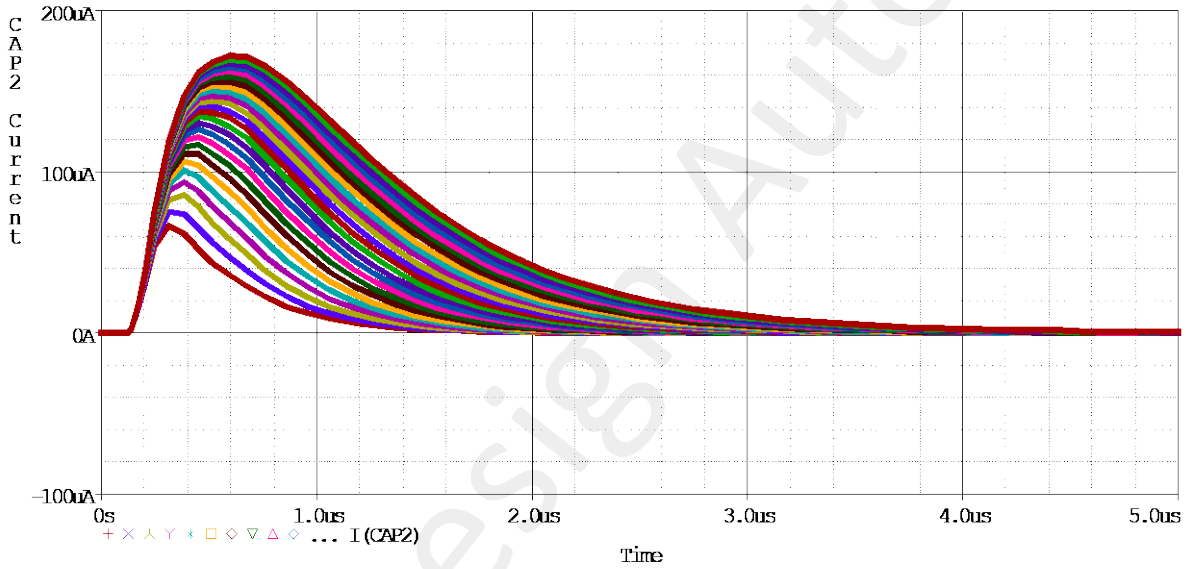


To View all 25 Traces:

1. Choose ***Trace > Add Trace*** or use the ***Add Trace*** toolbar button. 
2. Select ***I(CAP2)*** from the list of traces
3. Click ***OK***

All 25 traces will be displayed.

If there had been a current marker on the input pin of CAP2, then the same family of curves would have been displayed.



Now let's 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.

4. Click the name ***I(CAP2)*** below the X axis and press ***<Delete>*** to remove all 25 traces
5. Click the ***Add Trace*** toolbar button. 
6. Type **`I(CAP2)@25 I(CAP2)@1`** in the trace command box separated by a space

You can avoid typing most of the text by clicking on the ***I(CAP2)*** twice from the trace list and then editing the Trace Expression field.

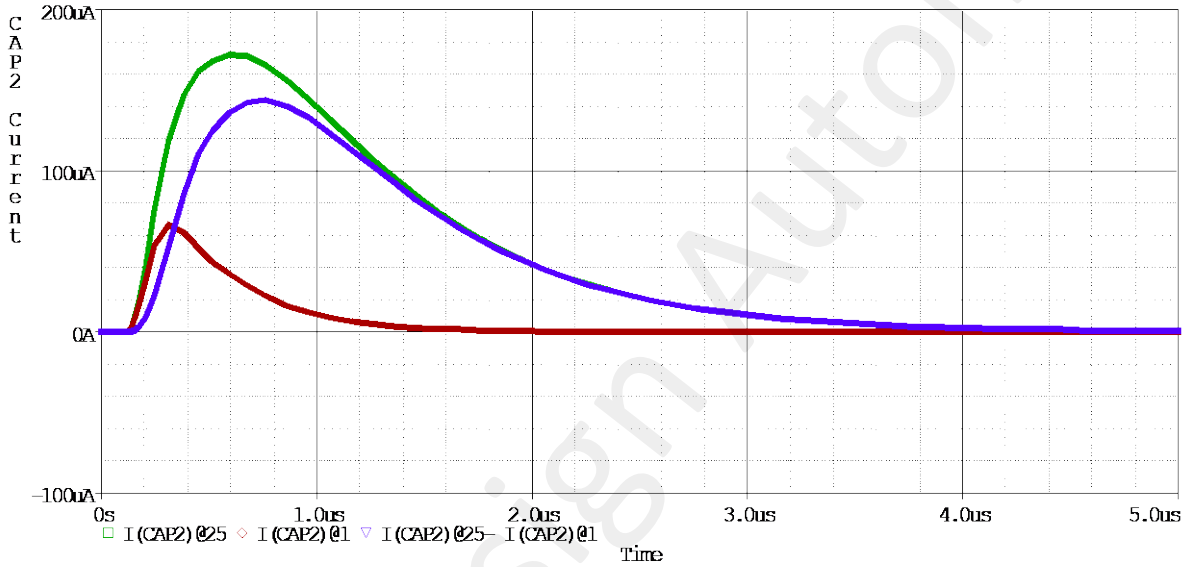
7. Click ***OK***

Notice the difference at the peak value. To quantify the change, you will 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.

8. Click the ***Add Trace*** toolbar button. 

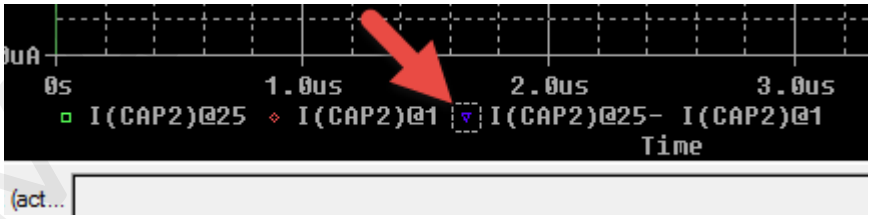
9. Type the waveform expression **`I(CAP2)@25 – I(CAP2)@1`** in the Trace Expression field

10. Click ***OK***



11. Click the ***Toggle Cursor*** toolbar button. 

12. Left click on the trace symbol in the legend representing the waveform expression I(CAP2)@25 – I(CAP2)@1



13. Click on the ***Cursor Peak*** toolbar button. 

14. Expand the Cursor window spreadsheet to see all the data, as below

	Trace Color	Trace Name	Y1	Y2	Y1 - Y2		Y1(Cursor1) - Y2(Cursor2)		143.746u		
		X Values	762.712n	0.000	762.712n		Y1 - Y1(Cursor1)	Y2 - Y2(Cursor2)	Max Y	Min Y	Avg Y
	CURSOR 2	I(CAP2)@25	166.034u	-3.2079f	166.034u		22.288u	0.000	166.034u	-3.2079f	83.017u
		I(CAP2)@1	22.288u	-3.1817f	22.288u		-121.458u	26.20E-18	22.288u	-3.1817f	11.144u
	CURSOR 1	I(CAP2)@25- I(CAP2)@1	143.746u	-26.19E-18	143.746u		0.000	3.1817f	143.746u	-26.19E-18	71.873u

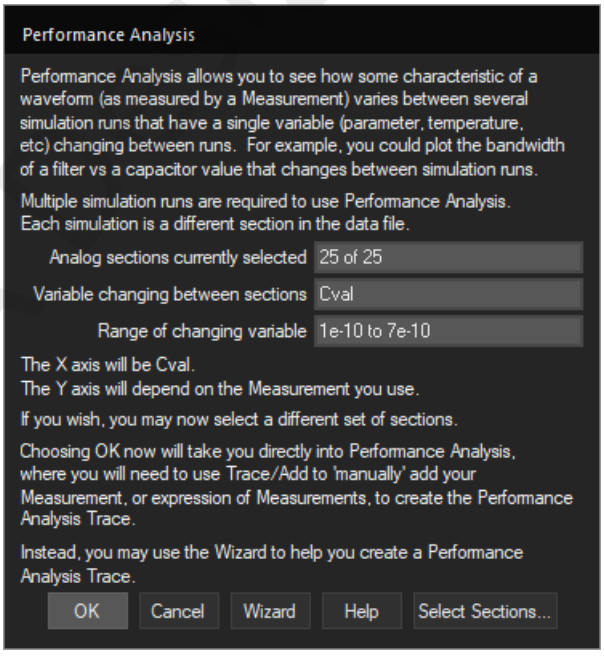
Notice that the Y value in the cursor box is 143.746u. This tells you that when CAP2 is set to 700pF, the current through CAP2 is maximum 144uA larger than when CAP2 is set to 100pF (at time 762.712ns, specifically). Your exact values may vary slightly.

Running Performance Analysis

Probe provides a powerful technique for comparing characteristics of a family of waveforms. This technique, called Performance Analysis or Measurements, uses the search commands to define functions that detect points on each curve in the family. Once these functions have been defined in an external file, they can be applied to a family of waveforms through the standard Add Trace command, and produce traces that are a function of the variable that changed within the family. In this example, it would be possible to apply a function using our search command to detect peak current for each waveform, then plot the peak versus the stepped CAP2 capacitance.

Refer to the PSpice A/D User's Guide, the Probe online help, or the performance analysis wizard in Probe for additional help.

1. In Probe, choose ***Trace > Performance Analysis*** to launch the Performance Analysis tool

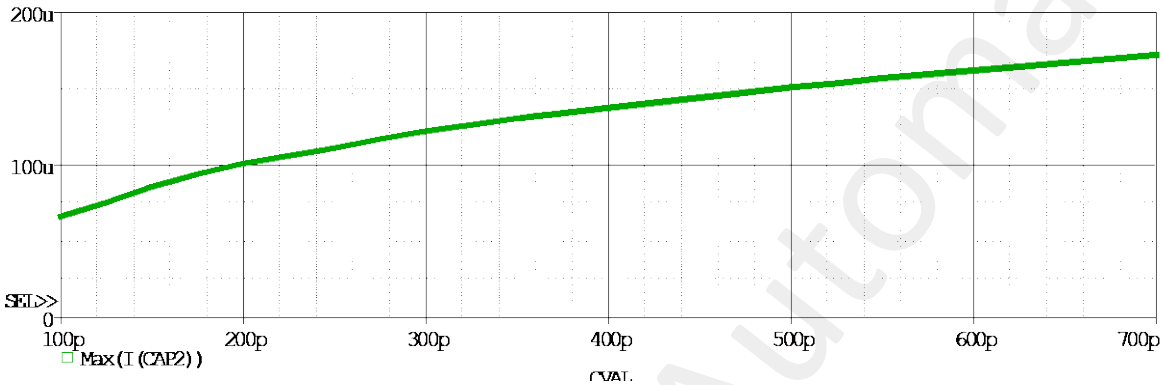


2. Hit the ***Wizard*** button and then ***Next***

3. On Step 2, select **Max** and then ***Next***

4. On Step 3, type or browse for the name ***I(CAP2)*** to generate a plot based on the maximum Y value for each trace in the family of curves given that the value of I(CAP2) is the variable being swept. Select ***Next*** Twice or ***Finish***

5. Your final graph should resemble the graphic that follows



Forward

This shows what the maximum value of current is (Y-axis) as a function of the capacitor value that is being swept.

> What approximate Capacitor value do you want to use if you want the peak of the current through the capacitor to be 100uA?

Note that if you want to find the exact position when y is 100uA, you can use the search function; remember that it's location sensitive:

```
Search forward level (100u)
```

If you want to search for and locate an exact x value, you can do so with this search function:

```
Search x value (400p)
```

Lesson 10. Linking PSpice Models to Capture

Available Libraries

OrCAD Capture is shipped with two basic types of libraries; simulation and layout. During a default installation, these two libraries are installed in the following directories:

- C:\Apps\SPB_17.4\tools\capture\library\
- C:\Apps\SPB_17.4\tools\capture\library\pspice\

Parts from the `capture/library` folder are not specially designed for simulation. Parts from the `capture/library/pspice` folder, however, are designed for simulation as well as layout. If you plan to simulate a design, be sure to use parts from the PSpice directory. By default, if you are working in a PSpice A/D project, the place part function automatically points to default simulation libraries.

If you use non-simulation parts in a design you try to simulate, you will receive an error that says there is no PSpice A/D Template for the parts. If you get this type of error, delete the parts and replace them with parts from the PSpice library. If the parts do not exist in the current libraries, you either need to download the models from the part vendor or create them manually.

Placing a Part

Each of these three methods opens the Place Parts flyaway tab shown in the graphic below. This flyaway tab is a graphical parts browser.

Part Browser

Part field - Located in the top, left column of the Place Part window. To locate a part, you can either enter the part name into this field or browse to locate the part library containing the part.

Part list - Lists all parts in the selected libraries panel. In the previous figure, notice that part LM324, an opamp, is in the library Opamp.olb.

Part names in this list are displayed alphabetically for all of the selected part libraries in the list. You can limit the number of parts displayed in the browse list by selecting or deselecting libraries. For example, if you are drawing a digital design that will use only 7400 parts, you can deselect all other libraries and leave only the desired 7400 libraries selected.

Libraries list - Lists all libraries configured for use with your project. Removing a library from this list does not remove the library file from your hard drive.

Graphical part display - Shows a picture of how the part when placed on your schematic page. Beneath the part graphics in the picture above are two small icons. The icon on the left means that the selected part has an associated PSpice A/D model. The icon on the right indicates that there exists a PCB footprint for the part.

OK button - Attaches the symbol to your cursor in anticipation of placement and then closes the Place Part dialog.

Add Library - Opens a browse window so you can select additional libraries you want to configure.

Remove Library - Removes the selected library configuration.

Filter - Allows you to limit the search and display function to only parts representing a PSpice A/D model, an HDL model, or to parts containing packaging information.

To exit the part placement mode, either press the <esc> key on your keyboard. You can also click your right mouse button to display the pop up menu and select ***End Mode***.

Wild Card Searches

You can perform wild card searches while in the part browser. For example, if you want to find all devices whose name contains 4000 and any trailing characters, such as A, B, or xxx, you would enter ***4000*** and press ***Enter***. A question mark can be used to match a single character.

As previously stated, when adding a library to the list of configured libraries, be sure that to reference the `C:\Apps\SPB_17.4\tools\capture\library\pspice` directory which contains the libraries specifically designed for simulation.

A common mistake is to use resistors, capacitors and inductors from Discrete.olb located in C:\Apps\SPB_17.4\tools\capture\library rather than using parts from Analog.olb which is located in C:\Apps\SPB_17.4\tools\capture\library\pspice.

Model Creation

These four steps are the foundation of using PSpice models in the OrCAD Capture environment.

Step 1: Import PSpice Model

The first step is to acquire a model; in the case of this lesson, we will get it from the internet.

The only concerns about bringing a part in from the internet are to:

- Make sure that you save the text directly into a notepad type tool to ensure that no non-ASCII text gets populated into the PSpice model file. Avoid word processing tools like MS Word or WordPad
- If you save the file directly from the manufacturer's site, make sure that you change the extension to .lib. SPICE libraries can come with many different file extensions – PSpice only uses ones ending in .lib though.

Step 2a: Link to OrCAD Capture Symbol

If you have a model device and want a Capture symbol made automatically, this is the way to make your part. Acceptable part types are:

- Diode
- Bipolar Transistor
- Magnetic Core
- Insertion Gate Bipolar Transistor
- Junction FET
- Operational Amplifier
- MOSFET
- Voltage Regulator, Voltage Comparator, Voltage Reference
- Darlington Transistor

Step 2b: Link to Custom OrCAD Capture Symbol

If you have a part that is subcircuit based (X) you will want to make your part using this method. It allows you to assign your PSpice model to a Capture symbol of your choosing instead of simply having it attached to a rectangle. The benefit of this is that your part is much easier to understand and use in a schematic with graphics that make sense to you.

Step 3: Place the OrCAD Capture Symbol

In order to use your new part in your design, you'll need to place it using Place Part in Capture. If you've put the part into a new library, you will need to add the library to the configured library list so that the part can be found.

Step 4: Configure PSpice Library

This is commonly the most forgotten step.

Since the design is the source for linking to ALL externally referenced files (including PSpice libraries); we'll need to tell the design where the library file from step 1 resides on your PC. Failure to do this step will result in errors in simulation about parts being undefined.

Testing Models

A critical step in getting new parts to work in Capture with PSpice is actually checking to see if they do what you think they should do. The best way to do this is to place them in a circuit where you know what the output should be (based on datasheet or bench measurement curves) and verify that the PSpice output gives you results that match.

Lesson 10, Lab 1: Linking a PSpice Model to a Capture Symbol

Lab Objective

- In this lab you will learn the process of associating PSpice models downloaded from a website to OrCAD Capture symbols and making them ready for use in a circuit.

Linking a manufacturer model to an existing Capture symbol

Semiconductor manufactures post spice models in their website. Before we can use those models in PSpice simulation we need to associate them with a Capture Symbol.

1. In Windows Explorer, browse to the *C:\EMA_Training\PSpiceEssentials* folder and locate MAX4412.FAM file. This is the PSpice model file of MAX4412 opamp downloaded from Maxim website (http://www.maximintegrated.com/)

Semiconductor manufacturers may use a different file extension for their PSpice model files. LIB is the recommended file extension for PSpice model files.

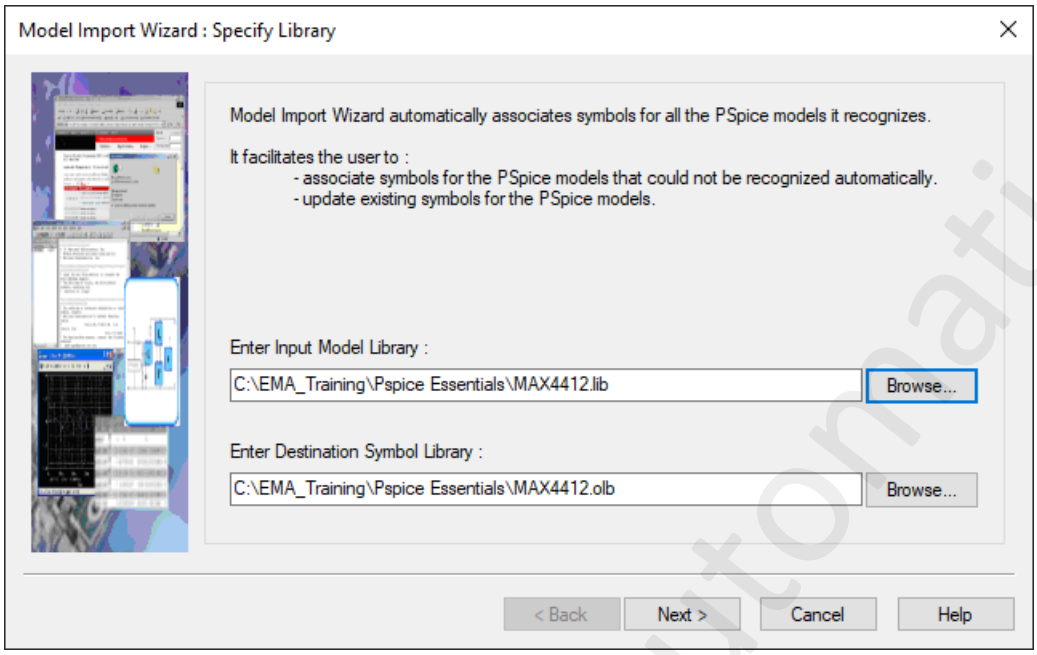
2. Right-click on MAX4412.FAM file and from the pop-up menu, select *Rename*. Change the file extension to .LIB (New file name will be MAX4412.LIB but you may not see the file extension if that is turned off in your machine)

3. Run the Model Editor by ***Start > All Programs > Cadence PCB Utilities 17.4-2019 > Model Editor 17.4***

4. Choose **File > Model Import Wizard [Capture]…**

5. In the Enter Input Model Library, browse to the **MAX4412.lib** file

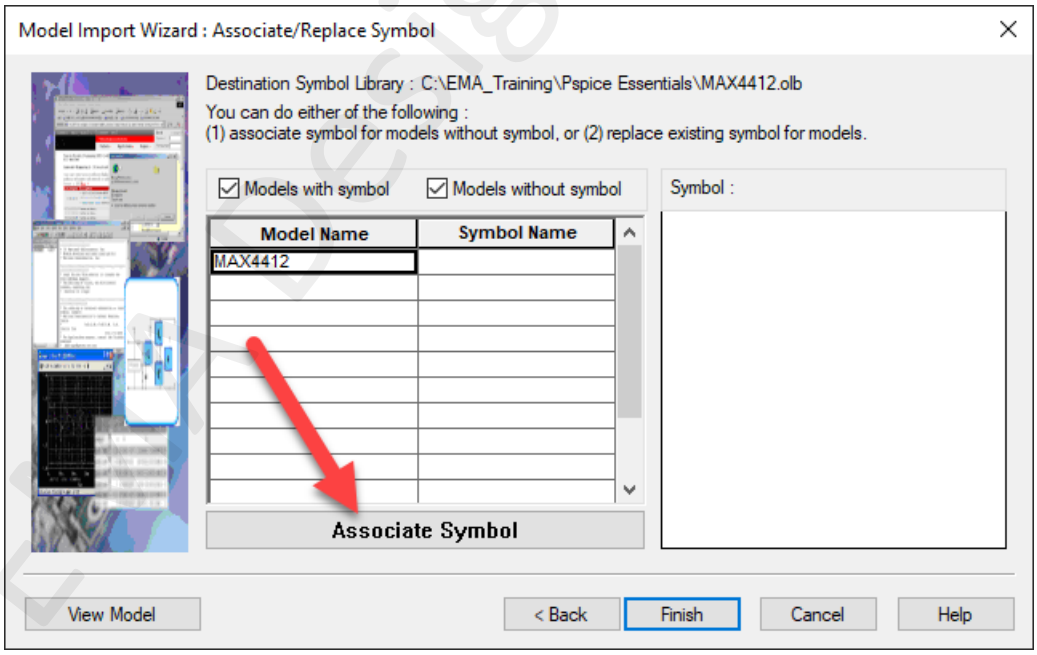
The Enter Destination Symbol Library path should fill in with MAX4412.olb in the same directory as the .lib file which is fine.



6. Click Next

We will be associating MAX4412 Model with an Operational Amplifier symbol included in standard OrCAD library.

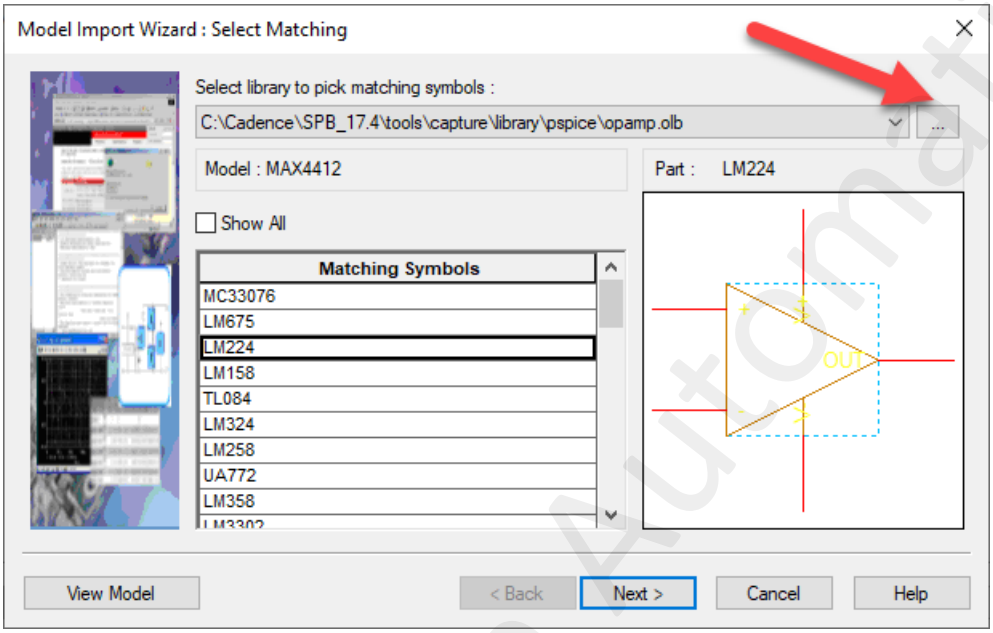
7. Click on '***Associate Symbol'*** button in the Associate/Replace Window



There are several Opamp symbols included in the OrCAD software. For the purpose of this lab, we will be using a matching symbol from

`C:\Apps\SPB_17.4\tools\Capture\Library\PSpice\Opamp.olb`

8. In 'Model Import Wizard : Select Matching' Window, *Select library to matching symbols*: section, **browse** to C:\Apps\SPB_17.4\tools\capture\library\pspice\opamp.olb

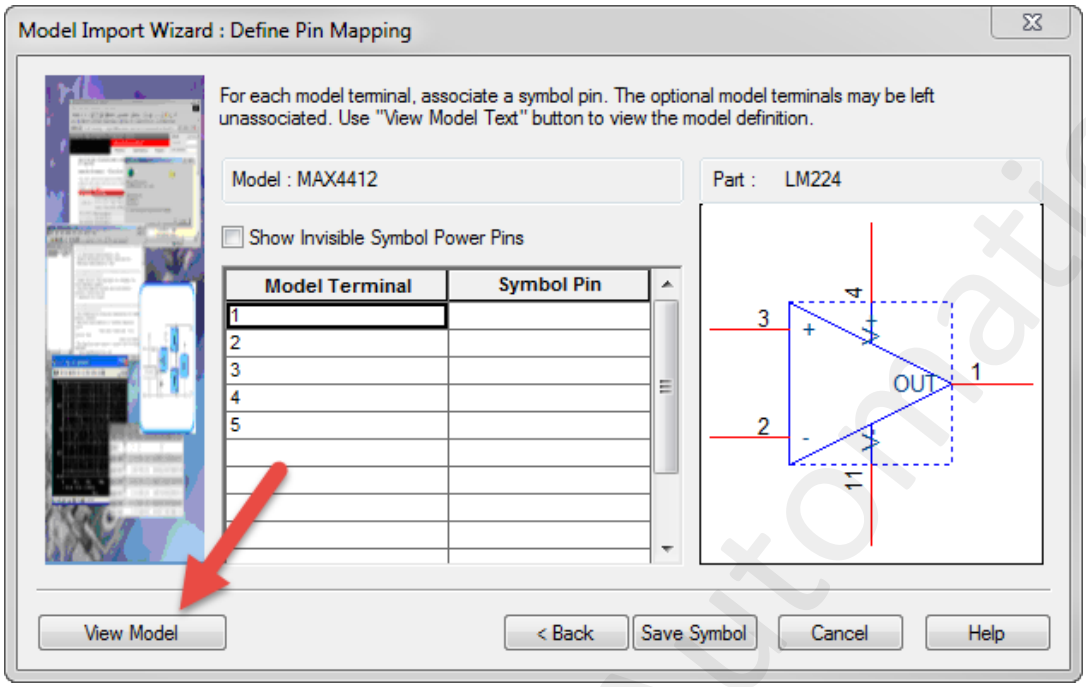


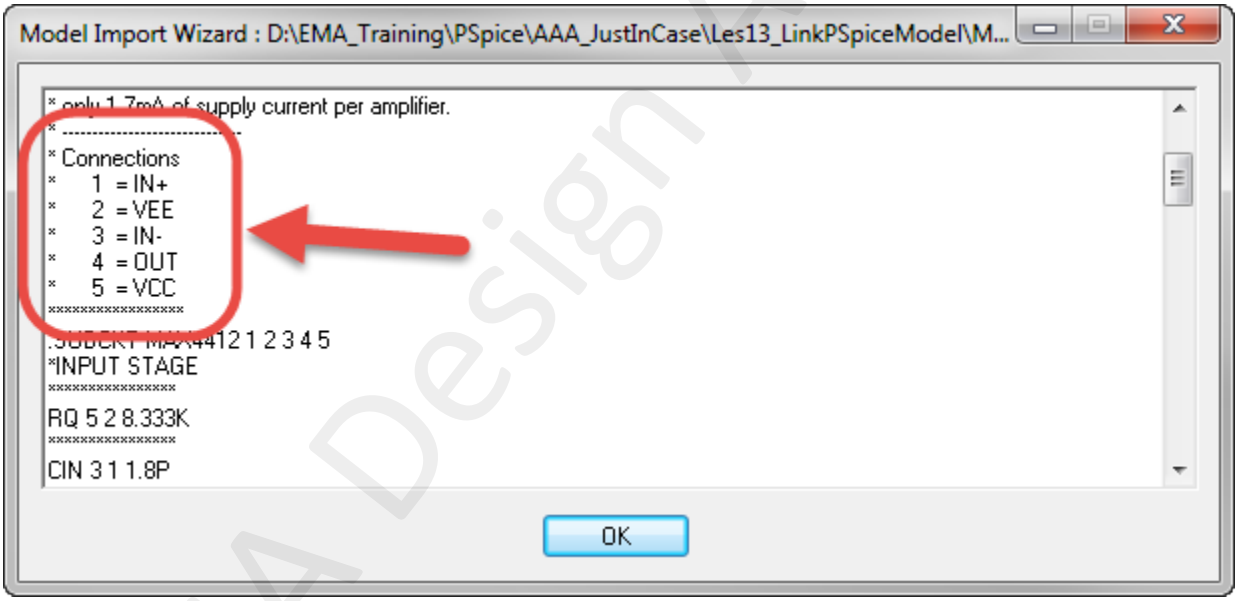
Note that the path in the screenshot is not the same as the one you will use on the training machine. The matching symbol's list is generated based on model definition of the model selected by the user.

9. Scroll through the list of Matching Symbols and pick a symbol of your choice (for the lab screenshots, we're using **LM224**)

10. Click **Next**

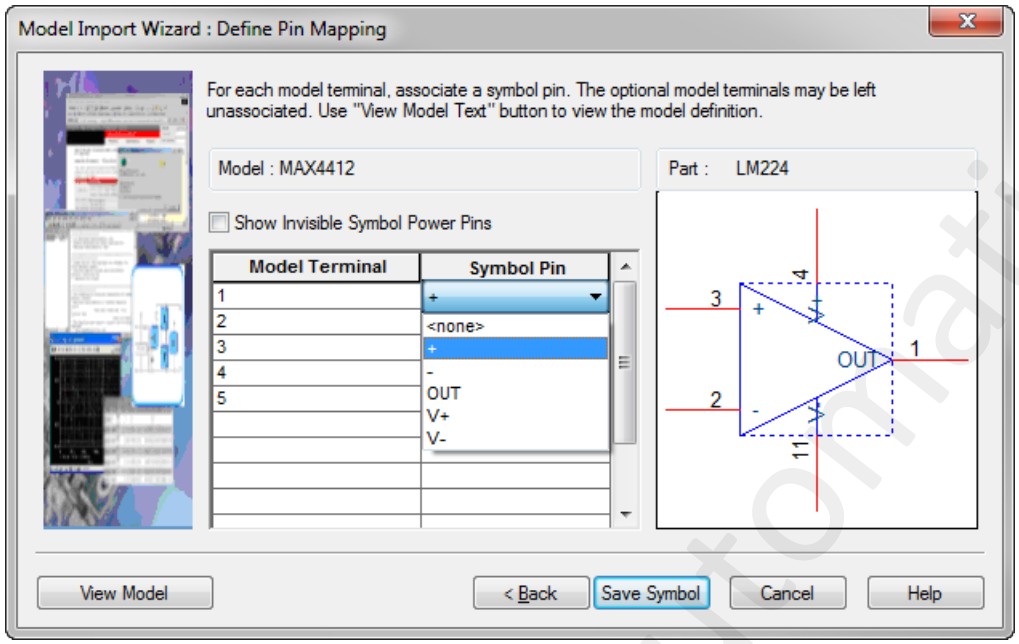
11. In 'Model Import Wizard : Define Pin Mapping' window, associate each model terminal with a symbol pin; since we're not sure what pins 1 through 5 should be connected to we'll click **View Model** to view the model terminal pin definitions





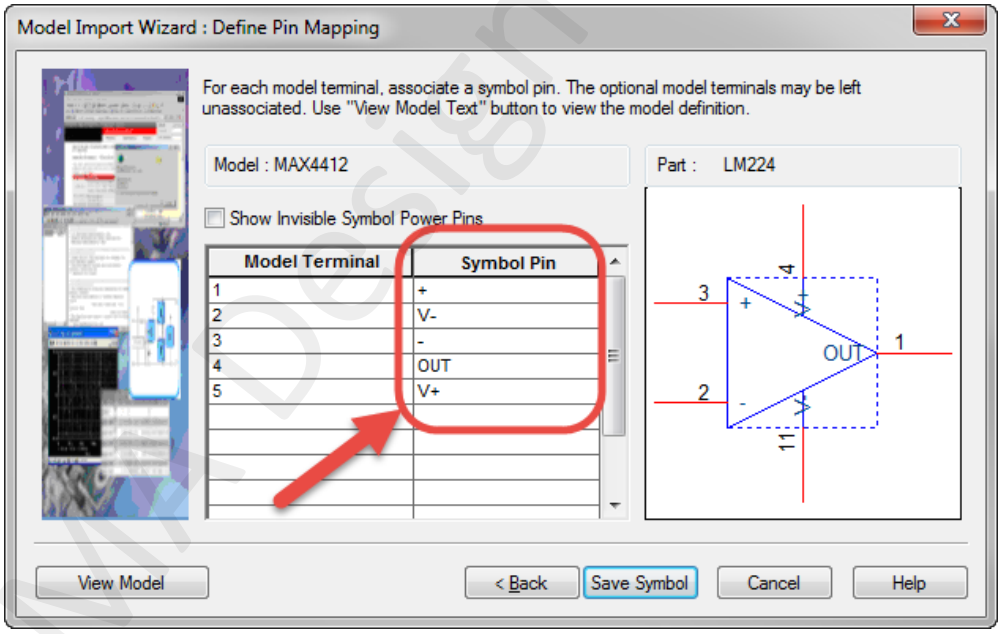
In the Model Text, note that terminal 1 is IN+ pin (Non-inverting input pin). We need to map this terminal to matching symbol pin.

12. Select **'+'** from the Symbol pin drop-down list against Model terminal 1



13. Map the remaining Model terminal to rest of the symbol pins

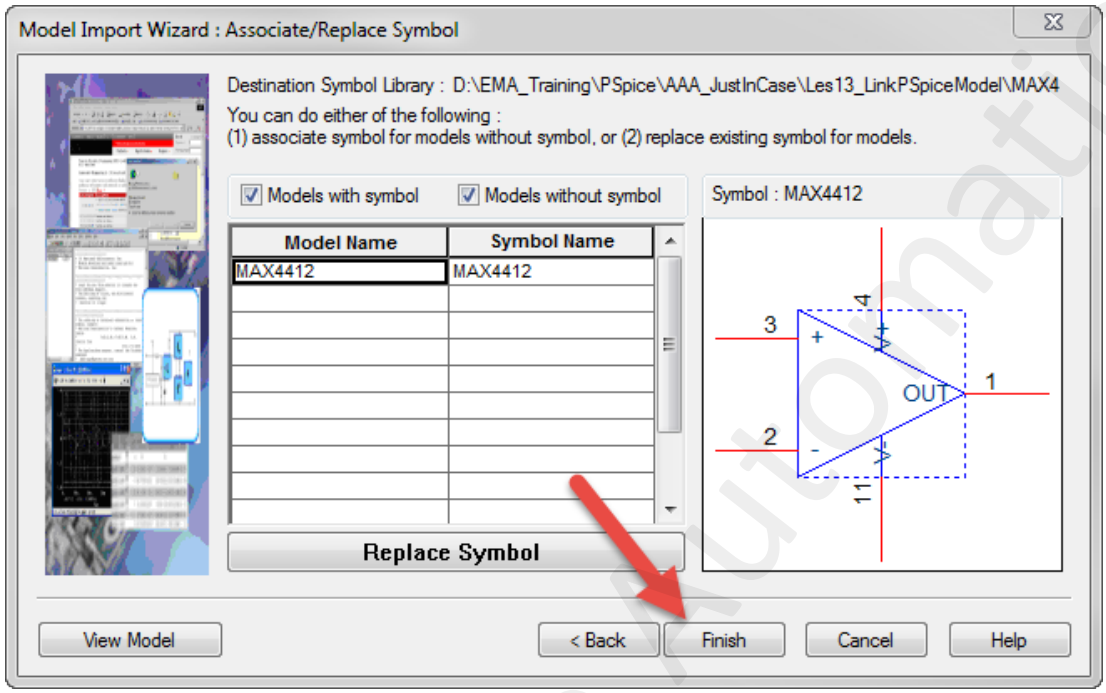
The final mapping window should look like this:



14. Click on **Save** Symbol button

Capture copies the base symbol to destination OLB file (C:\EMA_Training\PSpiceEssentials\MAX4412.olb) and rename the symbol name to MAX4412 (subckt name)

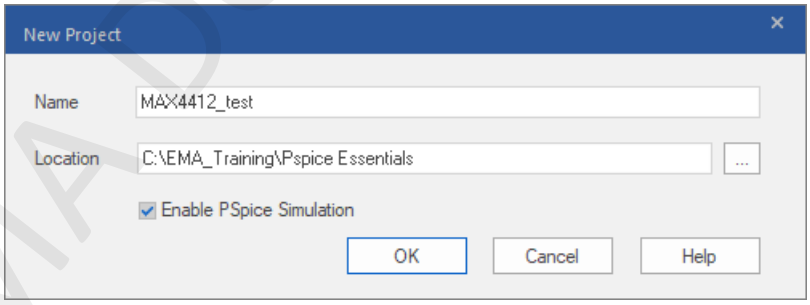
15. Click on **Finish** button to finish the Symbol association process and click **OK** to close the Status Dialog



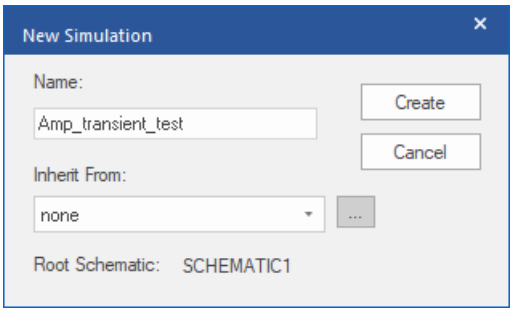
16. Close PSpice Model Editor

Configuring New Parts and Models

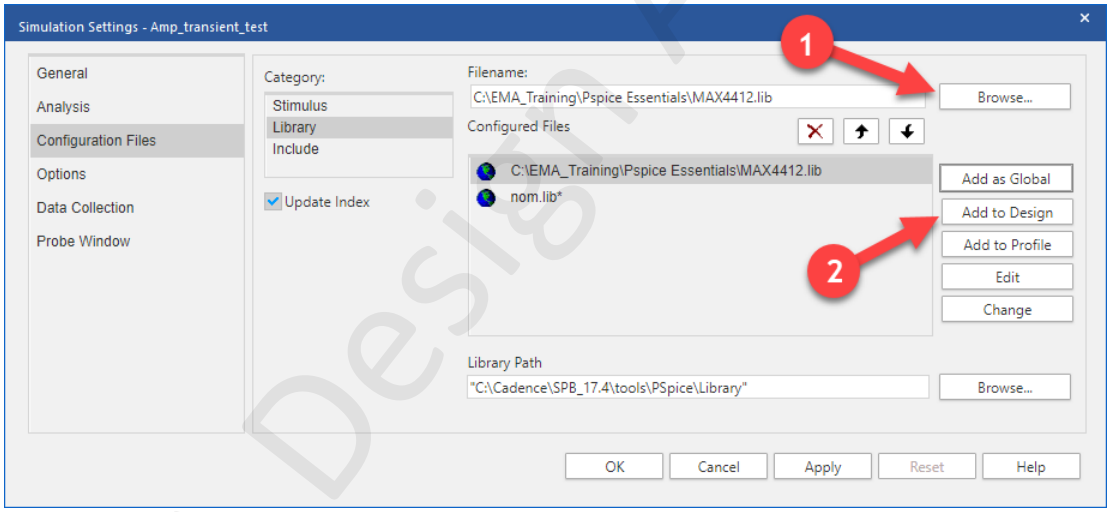
1. Create a **new project** called ***MAX4412_test*** in the C:\EMA_Training\PSpiceEssentials directory and base it on a blank project.



2. Create a simulation profile called ***Amp_transient_test***.

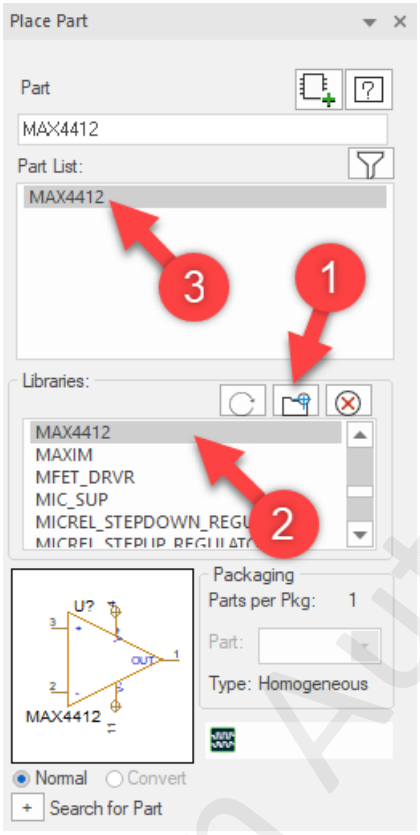


3. Click the ***Configuration Files*** tab.
4. Click ***Library*** in the ***Category*** section.
5. Click ***Browse*** in the upper right hand corner and navigate to the C:\EMA_Training\PSpiceEssentials directory.
6. Double-click on MAX4412.lib to select it and place it in the Filename field.
7. Click ***Add to Design*** to make MAX4412.lib visible to the current design.



You could have clicked Add as Global to make it visible to every design that is opened or created.

8. Click ***OK*** to close the dialog.
9. Open the ***Place Part*** dialog using your preferred method.
10. Click the Add Library icon  and navigate to the C:\EMA_Training\PSpiceEssentails directory.
11. Double-click on MAX4412.olb to configure the library.

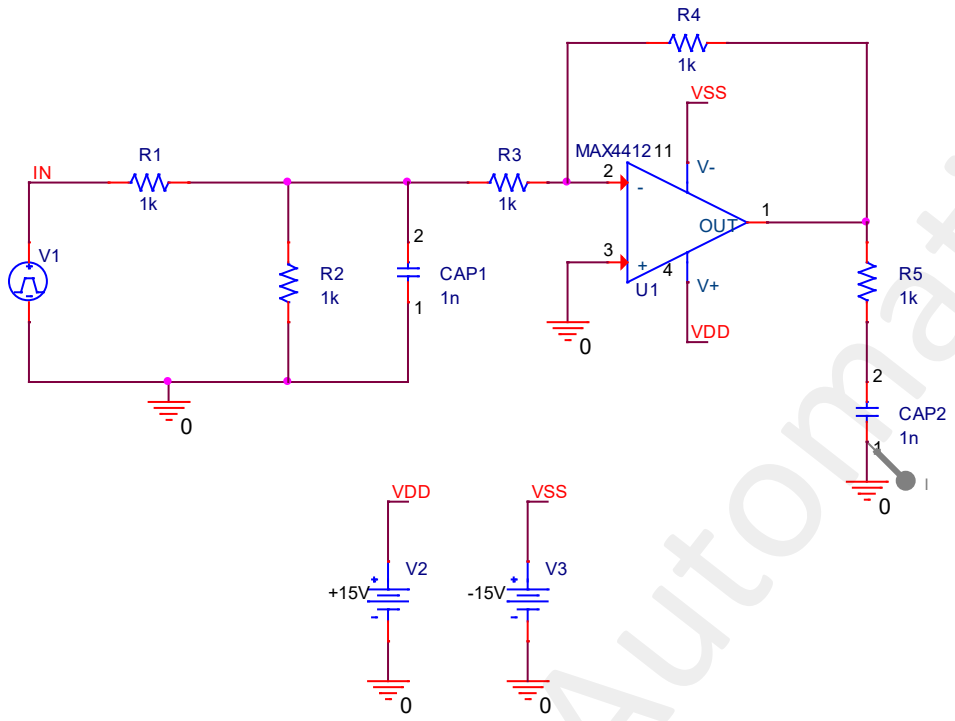


12. Select the ***MAX4412*** part and place an occurrence of it on the schematic page.

Your new model is now configured. It has an associated graphic and is ready for use in a design.

Optional Exercise

Create some circuitry around the freshly placed opamp to verify that it does work as intended. To expedite this process, you can use the completed files from Lab 5 (create a New Project and base it on this completed lab) and swap out the opamp. If you don't have the completed lab 5, you can use the one from the Solutions folder.



Lesson 11. Editing a Model

Creating a Model with the PSpice A/D Model Editor

It is sometimes necessary to edit an existing PSpice A/D model. This might be because you have a device that is similar to an existing model and changing one or two parameters will yield a model for the new device. Or, you may need to add tolerances to a model parameter in order to perform a Monte Carlo or Worst Case analysis.

There are several ways to edit a model. Since all of the model libraries are ASCII text files, you can edit the files in the text editor of your choice. However, OrCAD Capture provides an interface to the Model Editor. If the Model Editor recognizes the model type, then you can edit the model graphically. If the model type is not recognized, the Model Editor will open an integrated text editor for editing the model by hand. Both of these methods automatically take care of configuring the new libraries without over-writing the original files.

Recognized PSpice A/D Model Types

Other devices may be edited or created in the Model Editor, but must be done in the text mode.

Text Mode

1. Click to select the symbol of the model that you want to edit.
2. Choose ***Edit > PSpice Model*** from the menus.
3. Edit the model parameters or device specifications and extract the model parameters.
4. Save the file and return to the design.

This creates a model library file that has the same base name as the profile that was active when the Model Editor was invoked. It will also automatically configure the model library in the simulation profile at the profile level. If you want the library to be global, simply change it in the simulation profile.

Graphical Mode

The Model Editor screen consists of the following four parts:

1. The parts list for the current library
2. The specifications for the current device
3. The graph of the specification
4. The model parameter table

Device Specification

The device specifications can each be accessed via a tab below the graph. The device specification section allows you to input new values for the data curve of that specification. For example, if you wanted to change the forward current of the diode shown, you could enter values from the manufacturer's data sheet for the Vfwd and Ifwd. After entering the data points, you then extract the parameters by Choosing ***Tools > Extract Parameters*** or clicking the toolbar button.

Model Parameters

The model parameter section displays the extracted PSpice A/D model parameters. Those model parameters with their Active field checked are associated with the current parameter tab. Those parameters with their Fixed field checked will not be updated if the model parameters are extracted.

A minimum and maximum value is also displayed for each of the model parameters. These values are especially important if you decide to edit the model parameters directly instead of extracting them from device specifications.

Be aware that the curves shown are for the current specification only and does not take into account the other model parameters that can interact with it during a simulation. For this reason, it is always advisable to simulate a new model in a test circuit where its performance can be evaluated before using it in a simulation.

Local and Global Model Libraries

Library files may be configured with the scope of: local to a profile, local to a design, and globally. The graphic that follows shows settings within the ***Configuration Files*** tab of the ***Simulation Settings*** dialog box. In this graphic, the simulation is configured to reference the zener model library.

When a model is edited within OrCAD Capture, the Model Editor automatically configures it as local to a profile in the simulation profile.

When a library is globally configured, a reference to it is written to the initialization file. It is then visible to every project opened or created. A global library is listed in the simulation profile with an asterisk following its name.

A local library is configured only in the current project. It is still a file that is written to the hard drive, so you may configure it manually in other projects if necessary.

Referencing a Model

Most library parts are commercially available devices whose PSpice A/D models are well defined and tested. Because of this, editing such a device model is expected to be a rare occurrence. However, there are situations in which you might want to edit a standard model. If you do, be aware that the part name displayed on the schematic is not actually the model name referenced by the symbol. It is the name of the part. The ***Implementation*** property on a part is actually the link to the PSpice A/D model.

Parts found in the Breakout library are intended to be edited. In most cases they reference a default model that has only the minimum model parameters required for simulation. The property displayed on the schematic for a breakout part is the implementation property. For this reason, it is usually recommended that you use breakout devices when editing or creating a model.

If you are going to use a new model and part for more than testing purposes or if you are going to use the model and part in more than one design, you need to create a permanent symbol for the device.

The graphic that follows summarizes differences between parts from the regular PSpice A/D library and breakout parts. Breakout parts display the Implementation property while regular parts display the Value property.

Breakout Devices

Use the breakout devices to ensure that the Implementation property, which is the real model being used for the part, is being displayed as opposed to the Value property which may be meaningless.

Extracted Parameters

Model parameters and specification parameters found in a manufacturer's data sheet often have similar names. Consequently, careful attention is required when you edit a model.

For example, BF is a transistor model parameter that refers to forward beta. And, you can probably find a specification on the data sheet for the transistor for forward beta. The difference is that the data sheet specification is a measured parameter while the PSpice A/D model parameter is a number that has been calculated to work with all of the other model parameters in the model to simulate the desired behavior. The two numbers can vary several orders of magnitude from one another. The diode model parameter Bv (breakdown voltage), however, typically corresponds exactly to the zener voltage on the data sheet.

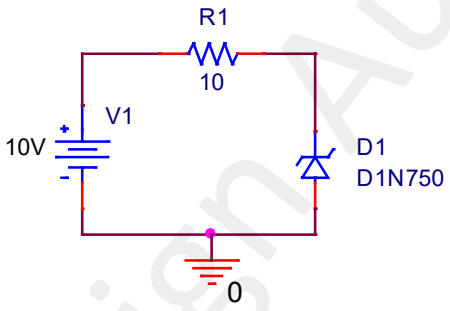
Lesson 11, Lab 1: Editing a Model

Lab Objectives

- Edit a zener diode model and simulate

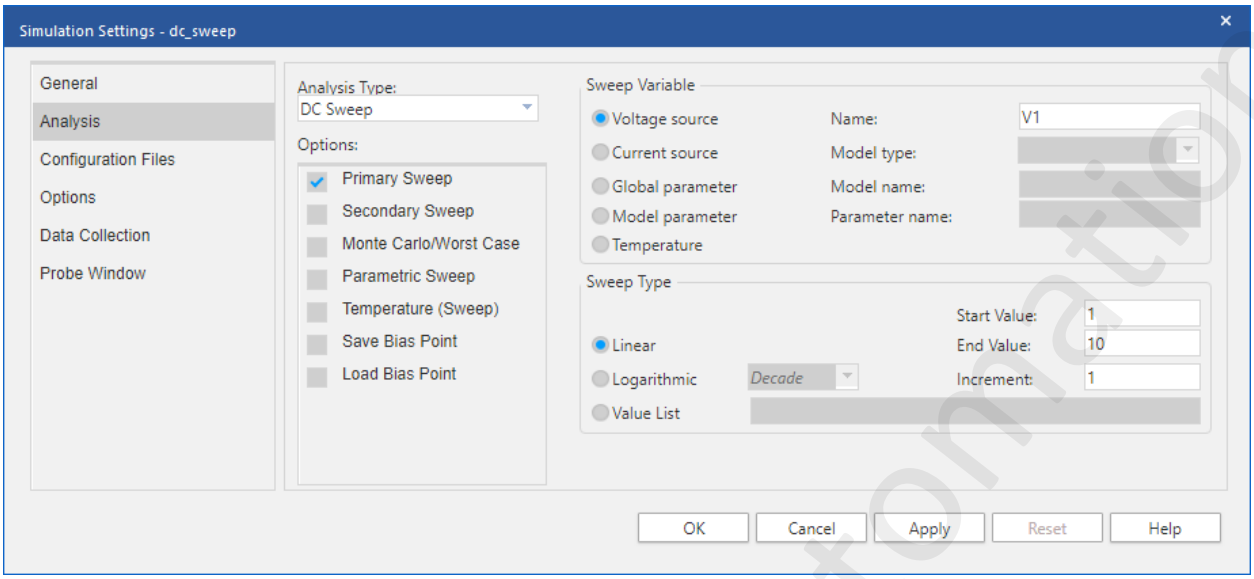
Creating Project Zener.opj

1. Create a new, PSpice A/D project
2. Name the project `Zener.opj` and base it off a blank project
3. Using the parts **VDC**, **0**, **R**, and **D1N750,** create the circuit pictured below

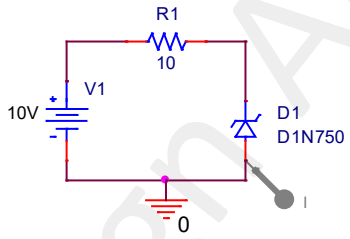


4. Set up a new simulation profile called dc_sweep
5. Set it to perform a DC sweep analysis that sweeps the voltage of V1 from 1V to 10V in 1V steps

The graphic that follows shows the suggested profile settings for this simulation.

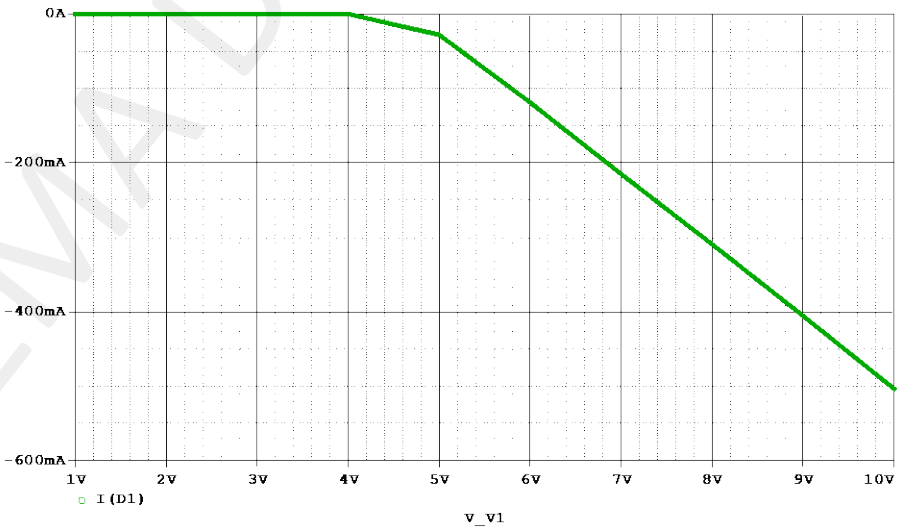


6. Place a current marker on the diode



7. Run the simulation (***PSpice > Run***, icon or F11 key)

8. Examine the results in the Probe window



Notice the diode begins conducting between 4 and 5 volts.

9. Click the diode to select it

10. Choose ***Edit > PSpice Model***

Or, click your right mouse button to display the pop-up menu and choose the Edit PSpice Model to start the Model Editor.

Find Bv in the list of parameters, it will be equal to 4.7.

11. Change the value of **Bv** to **8.0**

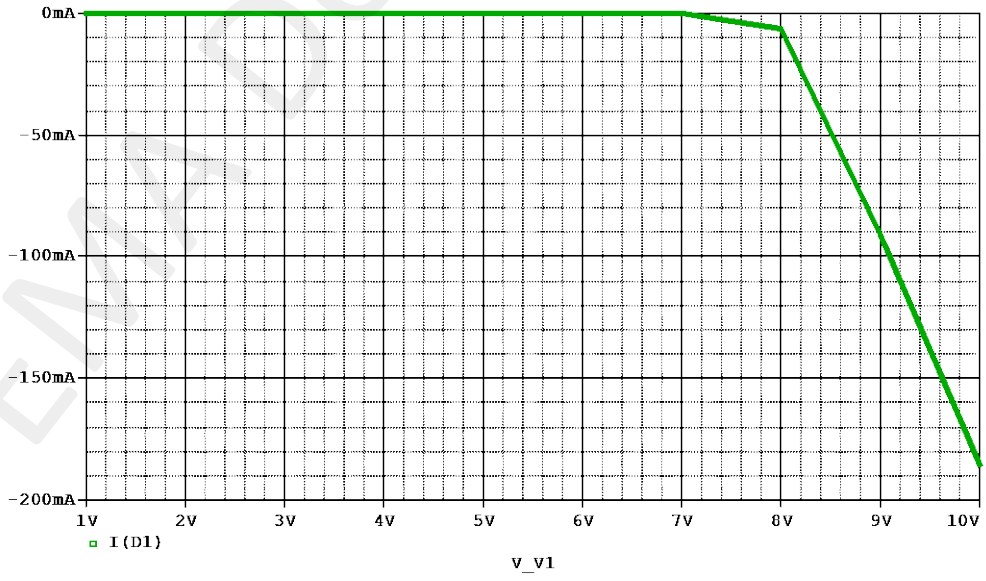
```
.model D1N750     D(Is=880.5E-18 Rs=.25 Ikf=0 N=1 Xti=3 Eg=1.11 Cjo=175p M=.5516
+           Vj=.75 Fc=.5 Isr=1.859n Nr=2 Bv=8.0 Ibv=20.245m Nbv=1.6989
+           Ibvl=1.9556m Nbvl=14.976 Tbvl=-21.277u)
*           Motorola    pid=1N750    case=DO-35
*           89-9-18 gjg
*           Vz = 4.7 @ 20mA, Zz = 300 @ 1mA, Zz = 12.5 @ 5mA, Zz =2.6  @ 20mA
```

12. ***Save*** the file, ***close*** the PSpice Model Editor

13. Return to OrCAD Capture

Running the Simulation

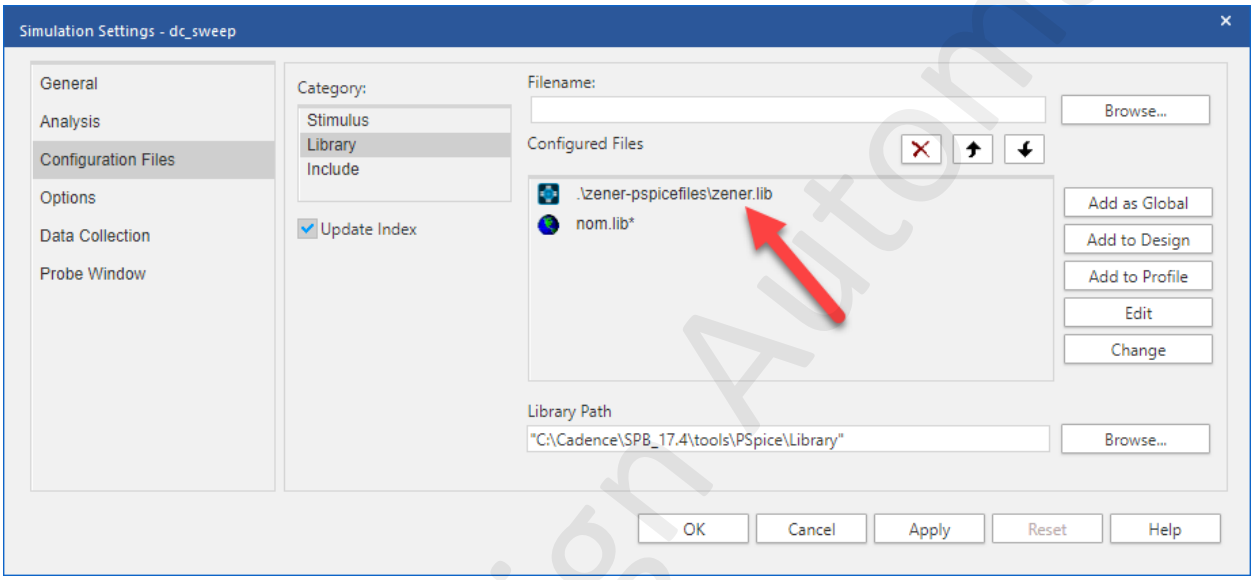
1. Rerun the simulation and examine the results. You will see that the diode begins conducting at about 8V



Checking the Libraries

By doing this, you did not change the installed library's value of BV for this part, PSpice automatically made a copy of the library and made the change to the now locally configured library.

Verify this by checking into the Configured Libraries in the Simulation Settings



Lesson 11, Lab 2: Creating a Model from a Datasheet

Lab Objectives

- Extract a diode model from a datasheet and simulate

Background

A Diode is a two terminal component which allows an electric current to pass in one direction (called the diode's forward direction), while blocking current in the opposite direction (the reverse direction). A diode's behavior is given by its current-voltage characteristics (I-V graph) as given below.

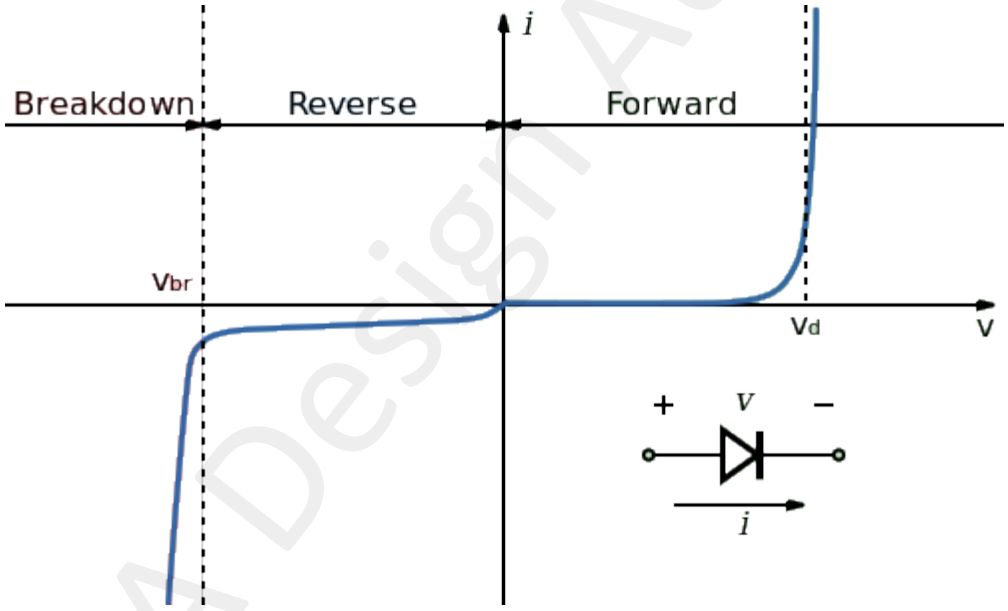


Image from https://en.wikipedia.org/wiki/Breakdown_voltage

Diode Model Parameters

Most of the Diode model parameters are dependent on the material properties of diode such as IS (saturation current), Eg (Bandgap voltage) etc. These parameters cannot be directly obtained from part datasheet. These can be extracted using Model Editor from characteristic curves given in datasheets. This extraction method is described in next section.

Some model parameters which have to be entered directly from datasheets are:

- BV (Breakdown Voltage): This is the reverse breakdown voltage at which reverse current increases abruptly.
- IBV (Reverse Breakdown knee current): This is the current at Reverse breakdown knee voltage.

These parameters can be obtained directly from datasheets as shown below.

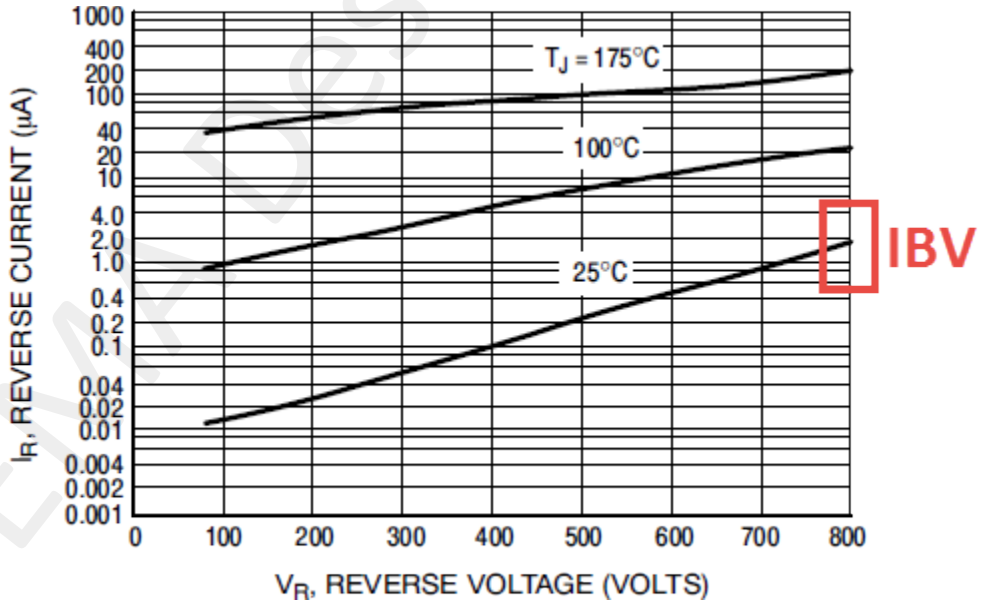
BV (Breakdown Voltage):

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Peak Repetitive Reverse Voltage Working Peak Reverse Voltage DC Blocking Voltage MURS480E	V_{RRM} V_{RWM} V_R	800	V
Average Rectified Forward Current	$I_{F(AV)}$	4.0 @ T_L=110°C	A
Non-Repetitive Peak Surge Current (Surge Applied at Rated Load Conditions Halfwave, Single Phase, 60 Hz)	I_{FSM}	70	A
Operating Junction and Storage Temperature Range	T_J, T_{stg}	−65 to +175	°C

BV

IBV (Reverse Breakdown knee current):



To get details of other parameters, refer to the PSpice Reference Guide.

Gathering Information to Model a Diode

We're going to build a model for a standard diode MURS480ET3G. Here's a summarized snapshot of its datasheet which is available as in your training folder as MURS480E.PDF

Maximum Ratings:

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Peak Repetitive Reverse Voltage Working Peak Reverse Voltage DC Blocking Voltage MURS480E	V_{RRM} V_{RWM} V_R	800	V
Average Rectified Forward Current	$I_{F(AV)}$	4.0 @ T_L=110°C	A
Non-Repetitive Peak Surge Current (Surge Applied at Rated Load Conditions Halfwave, Single Phase, 60 Hz)	I_{FSM}	70	A
Operating Junction and Storage Temperature Range	T_J, T_{stg}	-65 to +175	°C

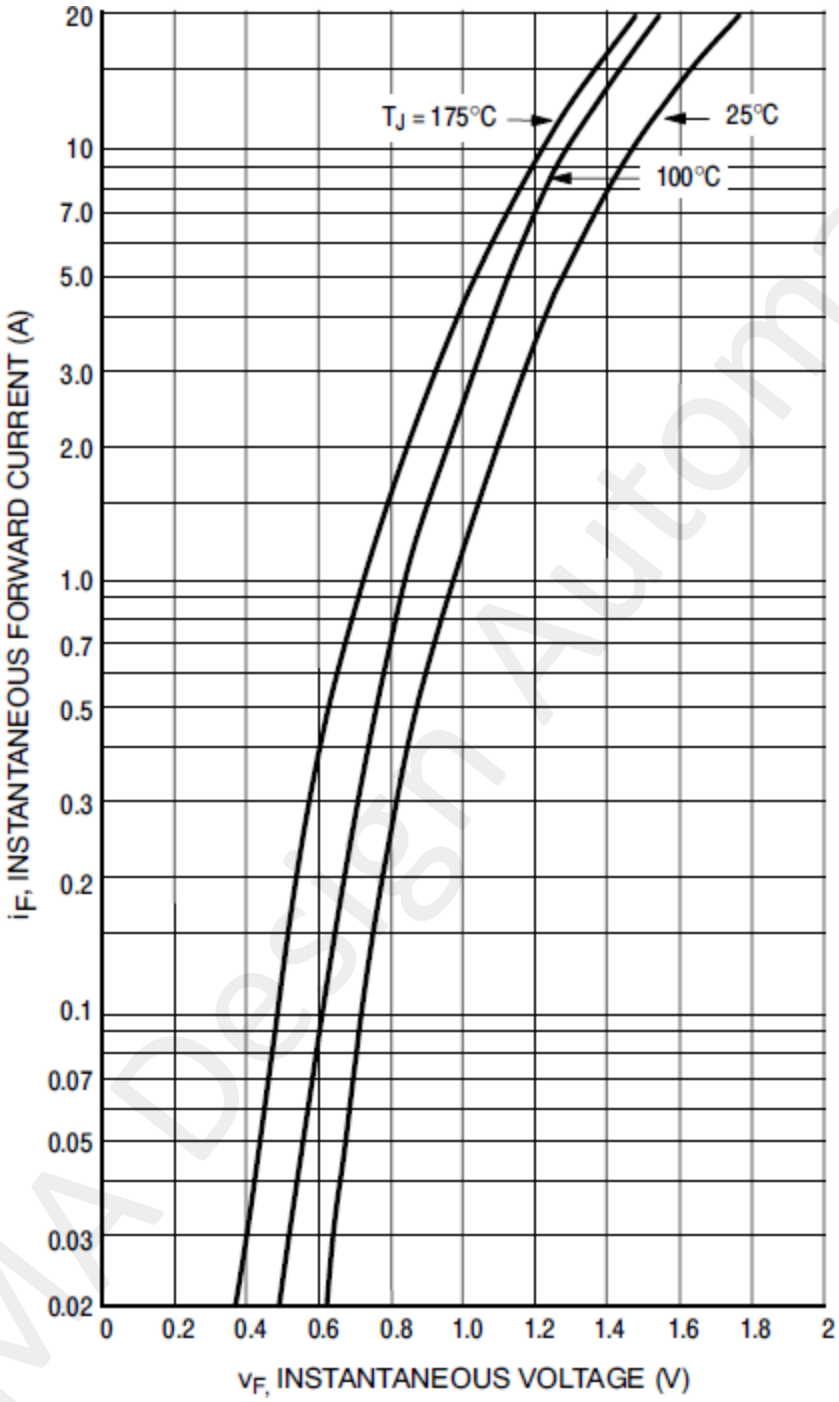
Electrical Characteristics:

ELECTRICAL CHARACTERISTICS

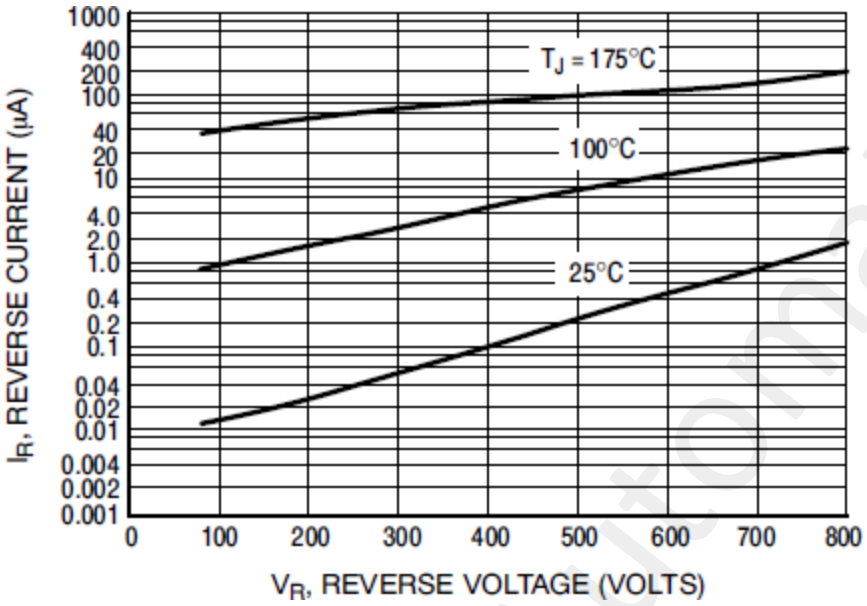
Characteristic	Symbol	Max	Unit
Maximum Instantaneous Forward Voltage (Note 1) (i_F = 3.0 Amps, T_J = 150°C) (i_F = 3.0 Amps, T_J = 25°C) (i_F = 4.0 Amps, T_J = 25°C)	v_F	 1.53 1.75 1.85	V
Maximum Instantaneous Reverse Current (Note 1) (Rated dc Voltage, T_J = 150°C) (Rated dc Voltage, T_J = 25°C)	i_R	 900 25	μA
Maximum Reverse Recovery Time (I_F = 1.0 A, di/dt = 50 A/μs) (I_F = 0.5 A, i_R = 1.0 A, I_{REC} = 0.25 A)	t_{rr}	100 75	ns
Maximum Forward Recovery Time (I_F = 1.0 Amp, di/dt = 100 Amp/μs, Recovery to 1.0 V)	t_{fr}	75	ns
Controlled Avalanche Energy	W_{AVAL}	20	mJ

trr

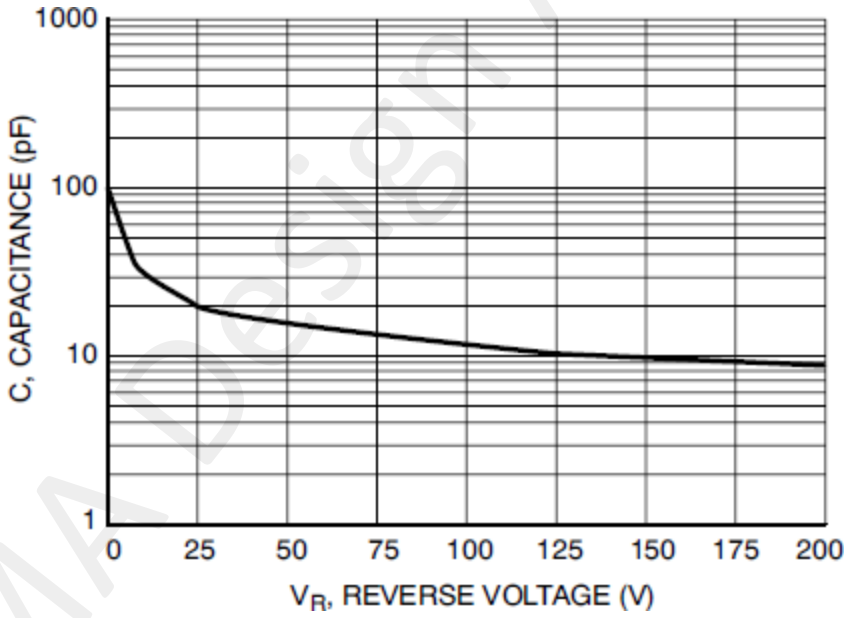
Vf vs. If



Vr vs. Ir



Vr. vs. C



As discussed above, most of the diode parameters are material specific so we use their characteristic curves to extract them.

In other words, parameters in a Diode SPICE model do not usually correspond to the parameters listed in datasheets. To create a SPICE model, you need to extract the characteristics of the actual device and translate that to the SPICE model parameters.

Create a New Library and Model

1. Open the PSpice Model Editor by going to ***Start > All Programs > Cadence PCB Utilities 17.4-2019 > PSpice Model Editor 17.4***

2. Choose ***File > New*** to start a new library

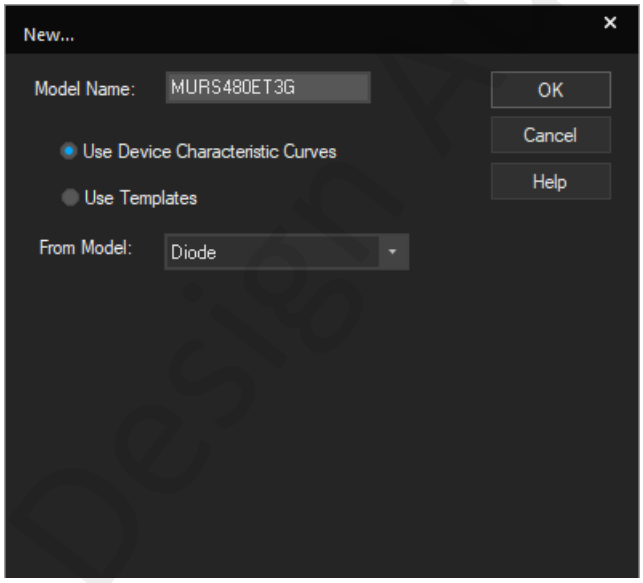
You will name the library when you save it at the end.

3. Choose ***Model > New…*** to begin creating a new model

4. Type `MURS480ET3G` in the Model Name field

5. Click the ***Use Device Characteristic Curves*** option

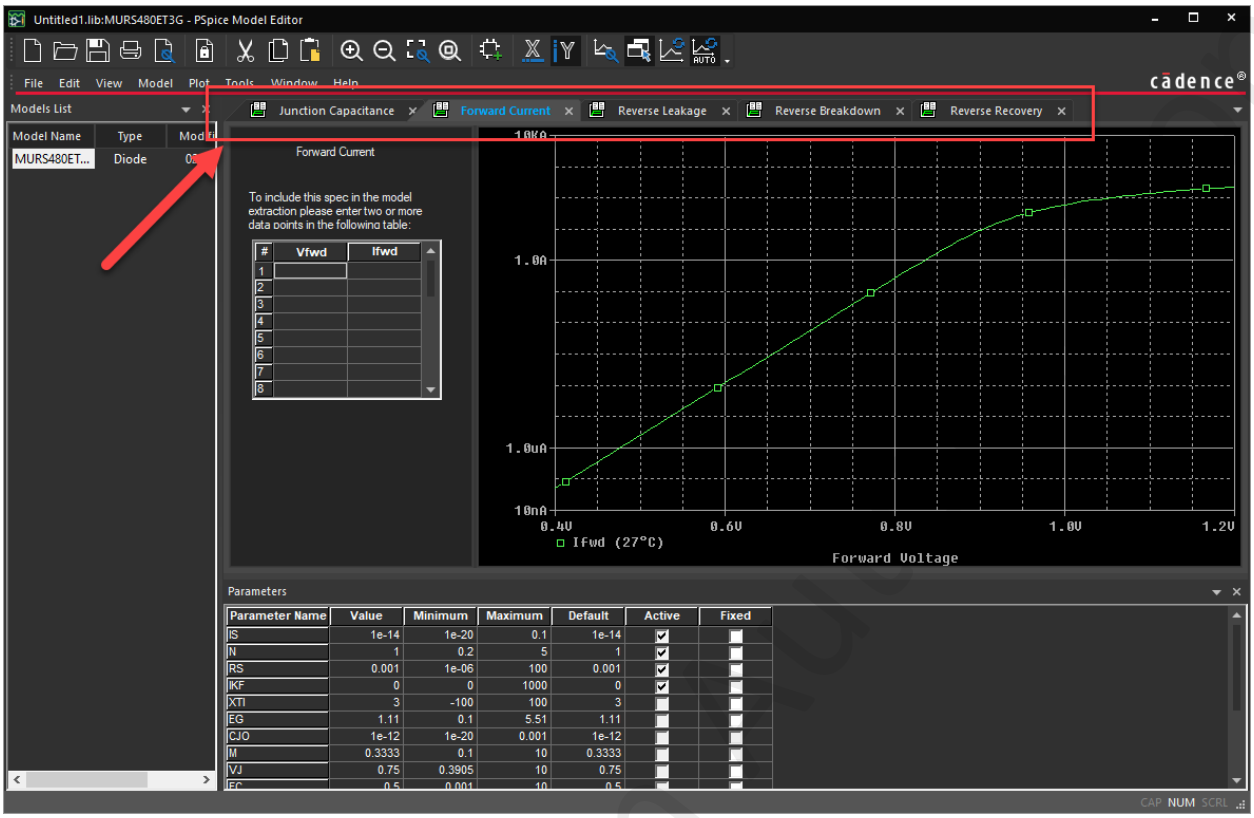
6. Check that ***Diode*** is the selected in the From Model drop down



7. Hit **OK**

The Model Editor opens. The Model Editor includes the following five tabs:

- Junction Capacitance
- Forward Current
- Reverse Leakage
- Reverse Breakdown
- Reverse Recovery

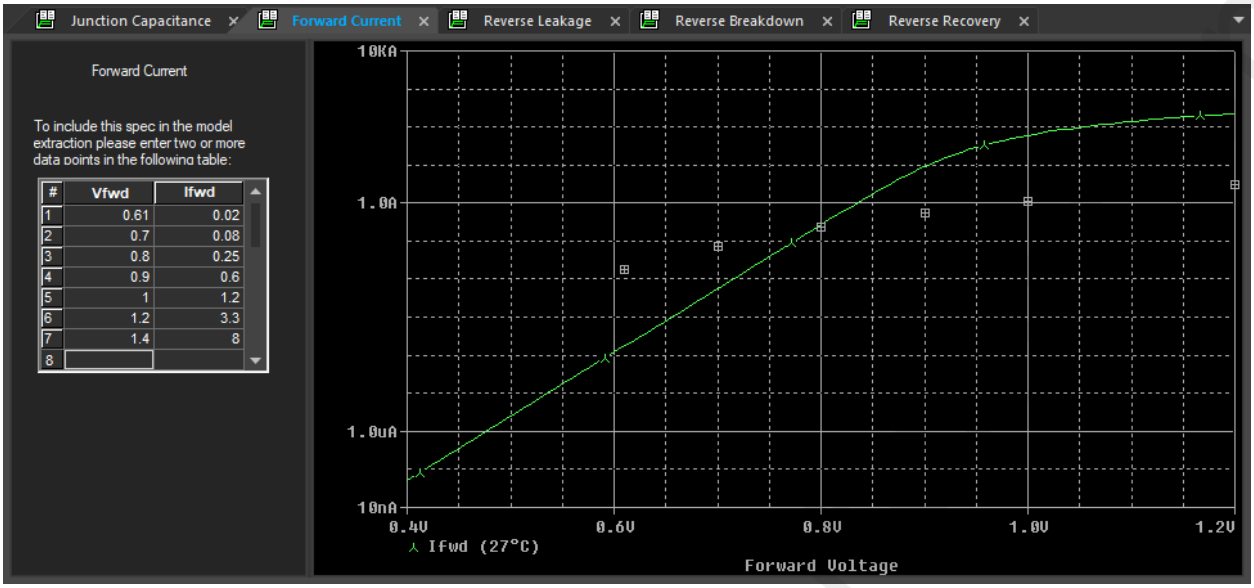


To define your new model in this exercise, you will enter new data values for each tab category. The sample data chart is provided above for each tab. After reviewing a sample data chart, you will be ready to input values.

Extract Diode Characteristics

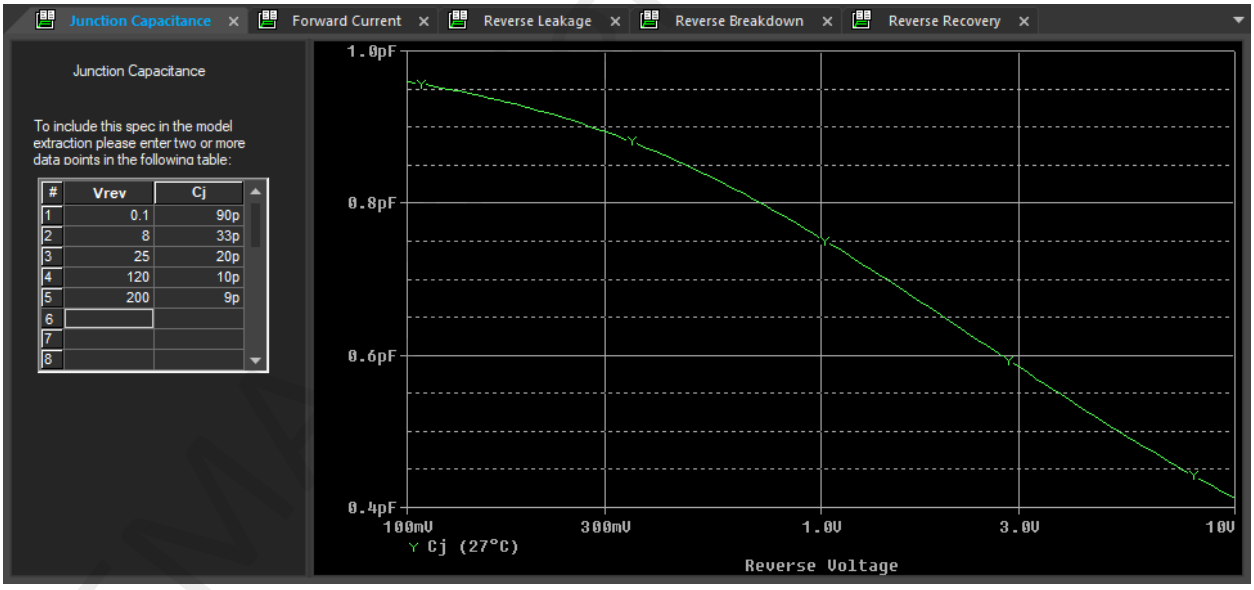
For the first three characteristics, enter data points from their corresponding characteristic curves in datasheet. Accuracy depends on the number of data points used. We will be using the curves printed in the “Gathering Information to Model a Diode” section above.

1. Taking information at 25C from the Vf vs. If curve and plotting the points into the Forward Current spreadsheet, we will end up with something like this:



2. After entering all of the points for Forward Current, click the Junction Capacitance tab

3. Taking the information from the Vr vs. C plot and entering it into the spreadsheet will yield something like this:

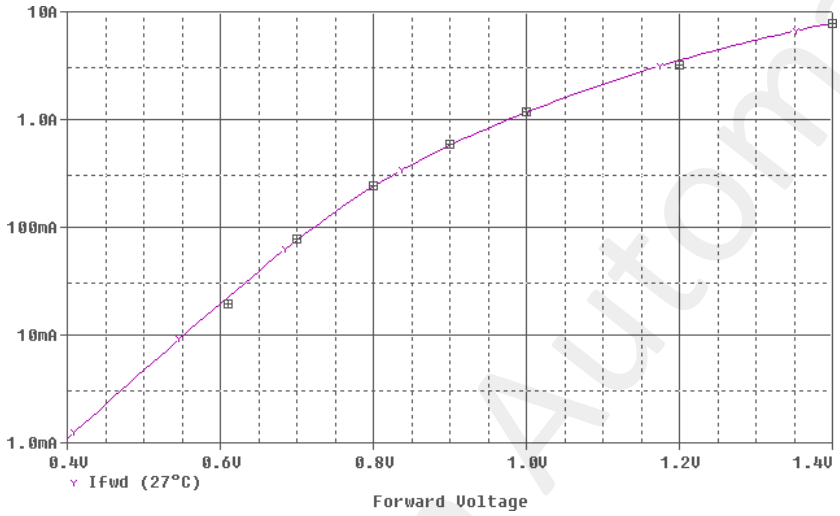


The Reverse Leakage doesn't extract and match very well on this particular part so we're going to skip that tab and assume that the reverse leakage current will not be well modeled or simulated

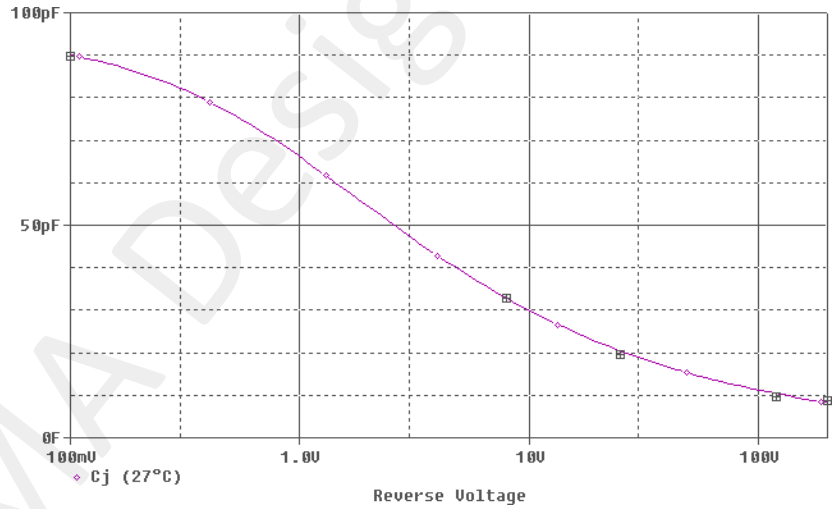
4. We haven't seen the traces move yet; to have the traces move to match the points that we've entered choose **Tools > Extract Parameters** or hit the icon

5. Verify that the traces go through the entered points on the Forward Current and Junction Capacitance tabs

Forward Current:



Junction Capacitance:



Entering Reverse Characteristics

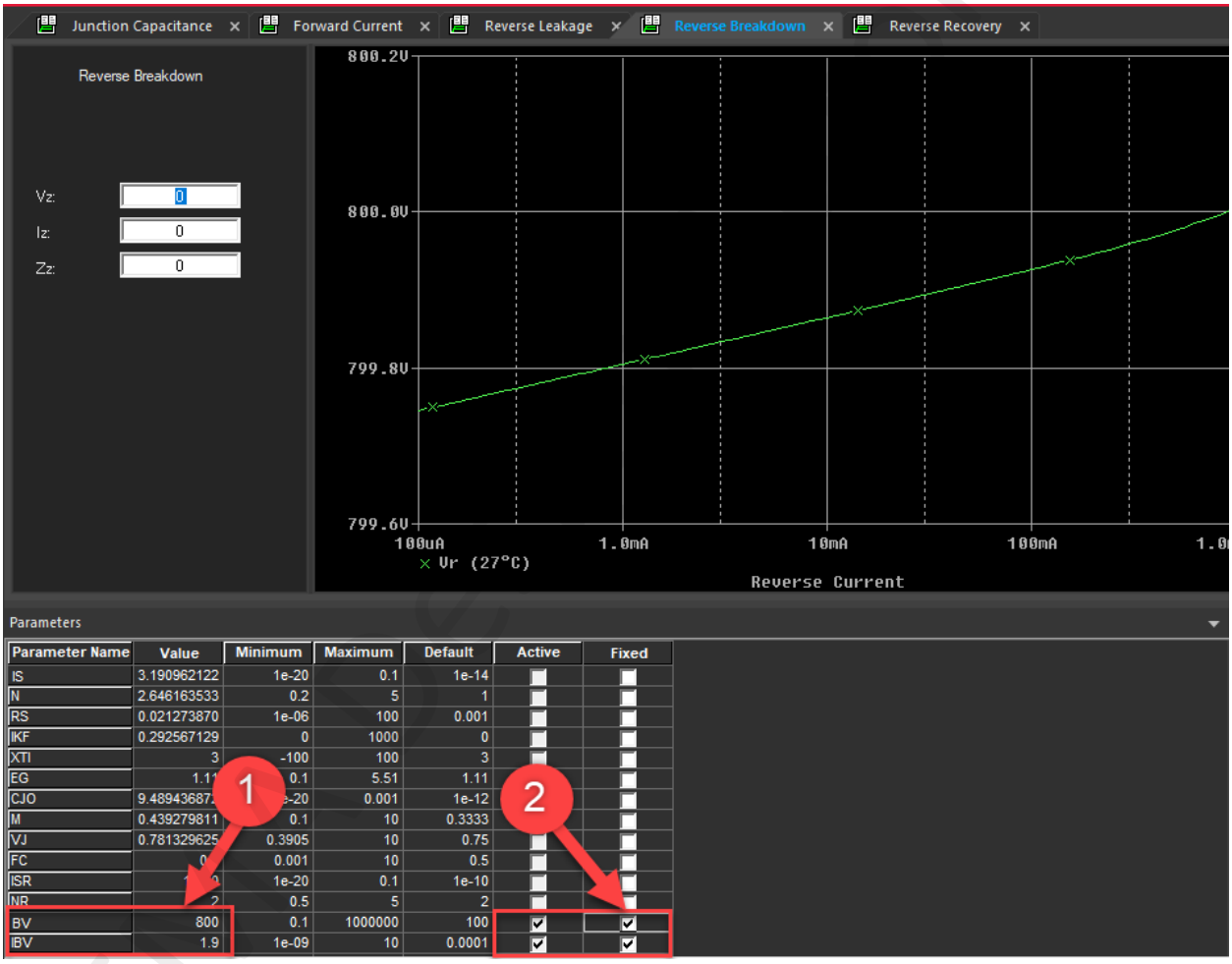
Since we are modeling a normal diode which is operated below breakdown unlike zener, we would skip the reverse breakdown characteristics. We will directly enter values of BV and IBV directly from datasheet, into the parameters columns.

In the Reverse Breakdown tab there are 3 parameters that are defined as:

- VZ – nominal Zener voltage @ IZ
- IZ – nominal Zener current @ VZ
- ZZ – Zener impedance @ IZ and VZ

Zz is not shown on the data sheet. You would normally contact the part vendor for the data.

1. Click on the Reverse Breakdown tab and enter the Breakdown voltage and the current at the breakdown voltage directly into the model, like this (should IBV=1.9uA):



2. It's important to also hit the Fixed box to the right of those two lines so the model editor does not attempt to change those 2 values

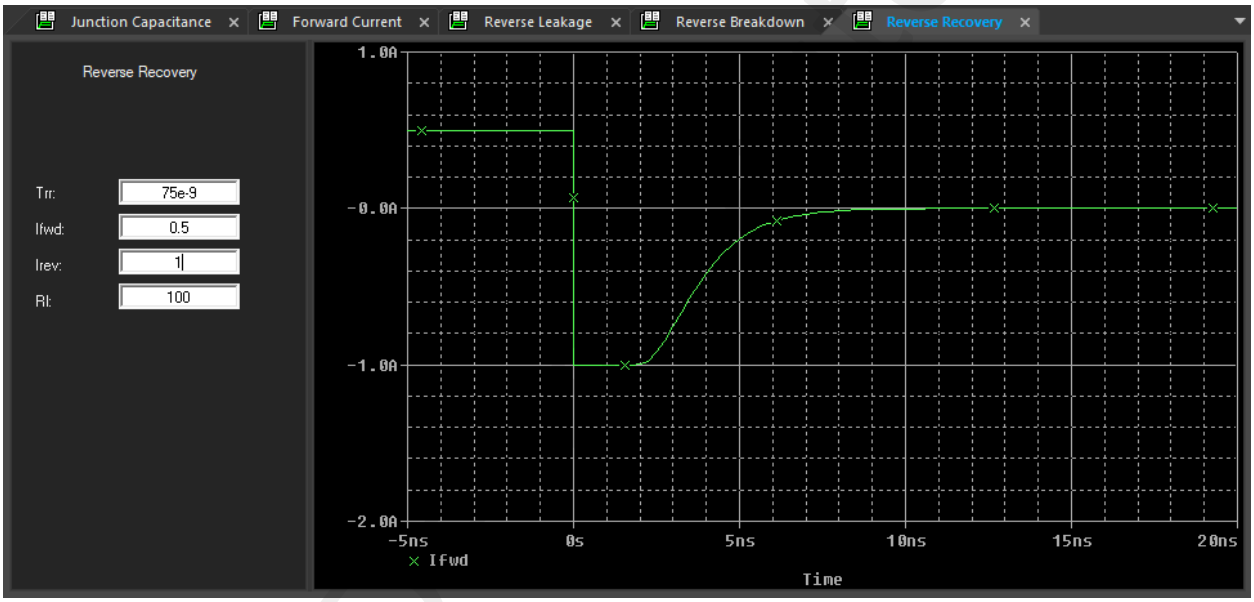
For reverse recovery curve, enter value of trr, If and Ir from Electrical Characteristics table, as given in datasheet. Most of the test fixtures have their load resistance at 100 ohms, so we can

leave that at its default of 100. If we want to use different load enter the same value in Rl column.

In the Reverse Recovery tab there are 4 parameters that are defined as:

- Trr = Reverse recovery time
- Ifwd = forward current (before switching)
- Irev = initial reverse current
- Rl = Load Resistance (total load of test fixture)

3. Click on the Reverse Recovery tab and fill in the parameters from the Electrical Characteristics table from the datasheet like this:



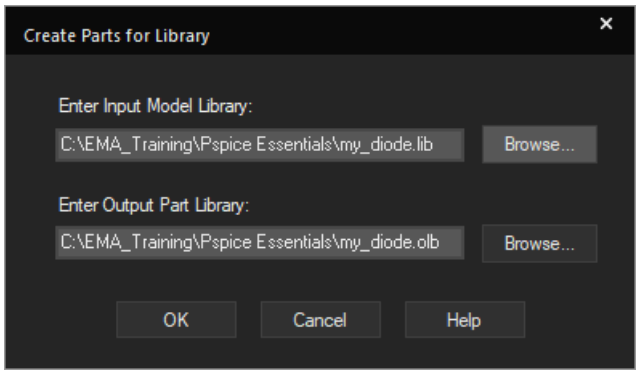
4. We haven't seen the traces move yet; to have the traces move to match the points that we've entered choose **Tools > Extract Parameters** or hit the  icon

5. Choose **File > Save As** and save the file as **MY_DIODE.lib** in the C:\EMA_Training\PSpiceEssentials\ directory

Create a Symbol

1. Choose **File > Export to Capture Part Library…**

2. Ensure that the MY_DIODE.lib is in the Input Model Library at the top and MY_DIODE.olb is in the Output Part Library at the bottom



3. Click **OK** to create the Part Library and then **OK** to close the status dialog box

4. **Close** the PSpice Model Editor

You just created a Capture graphical part library with a part that you can place in Capture that will call out this MURS480ET3G model that was just created, let's place the diode into a schematic now to test it.

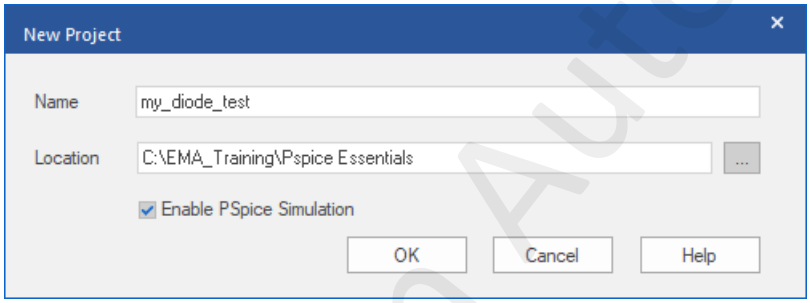
Lesson 11, Lab 3: Testing the Forward Current

Lab Objectives

- Put the recently created diode model into a test circuit to verify forward current

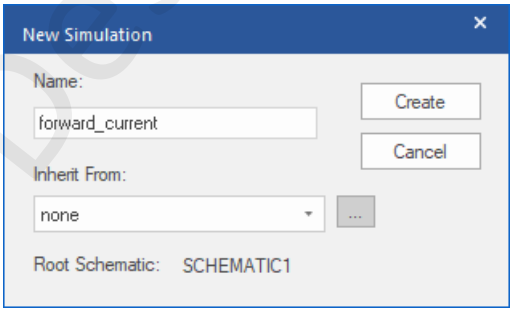
Configuring New Parts and Models

1. Create a new project, call it `my_diode_test` and save it in the C:\EMA_Training\PSpiceEssentials directory, **Check** *Enable Pspice simulation*, Hit **OK**



2. Base it off a blank project

3. Create a simulation profile (PSpice > New Simulation Profile or  icon) called **forward_current**, Hit **Create**



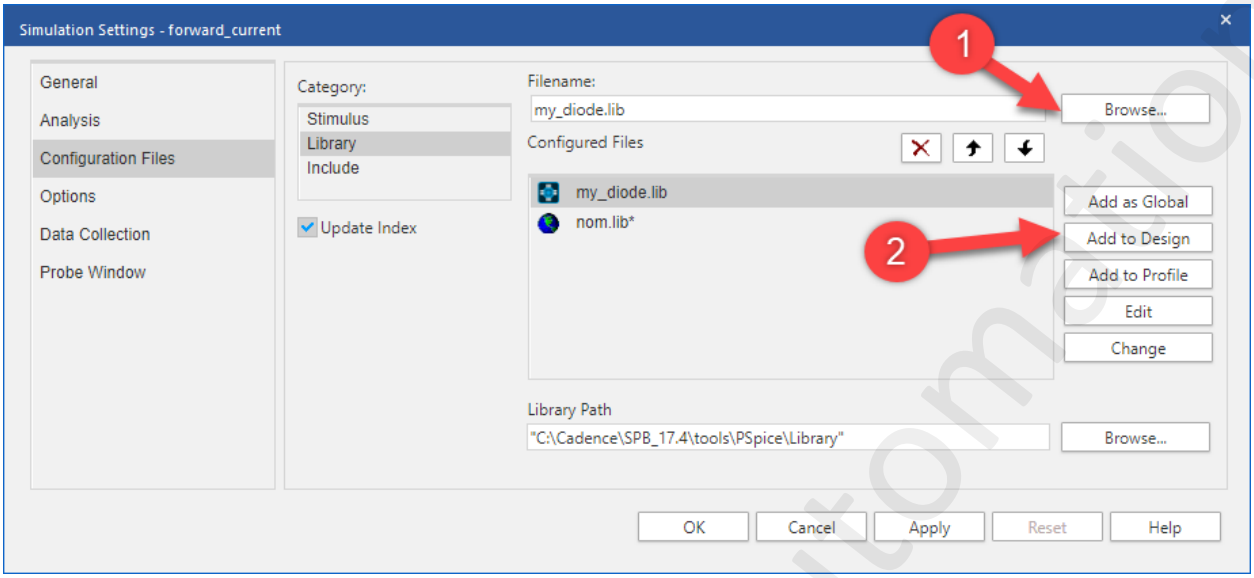
4. On the Simulation Settings window, click the Configuration Files tab

5. Click Library in the Category section

6. Click Browse in the upper right hand corner and navigate to the training directory

7. Double-click on MY_DIODE.lib to select it and place it in the Filename field

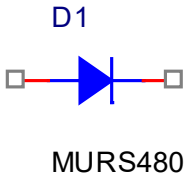
8. Click Add to Design to make MY_DIODE.lib visible to the current design



You could have clicked Add as Global to make it visible to every design that is opened or created.

Notice in the example above, the absolute path is removed so that if you move this project into a different directory, the pathing stays relative. If you would like to have the pathing be relative, instead of browsing to a location with the Browse button, just type my_diode.lib and "Add to Design" and it will hook it up for you.

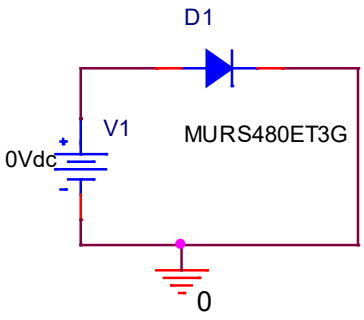
9. Click OK to close the dialog
10. Open the Place Part dialog using your preferred method
11. Click the Add Library icon and navigate to the training directory
12. Double-click on MY_DIODE.OLB to configure the library
13. Select the MURS480ET3G part and place an occurrence of it on the schematic page



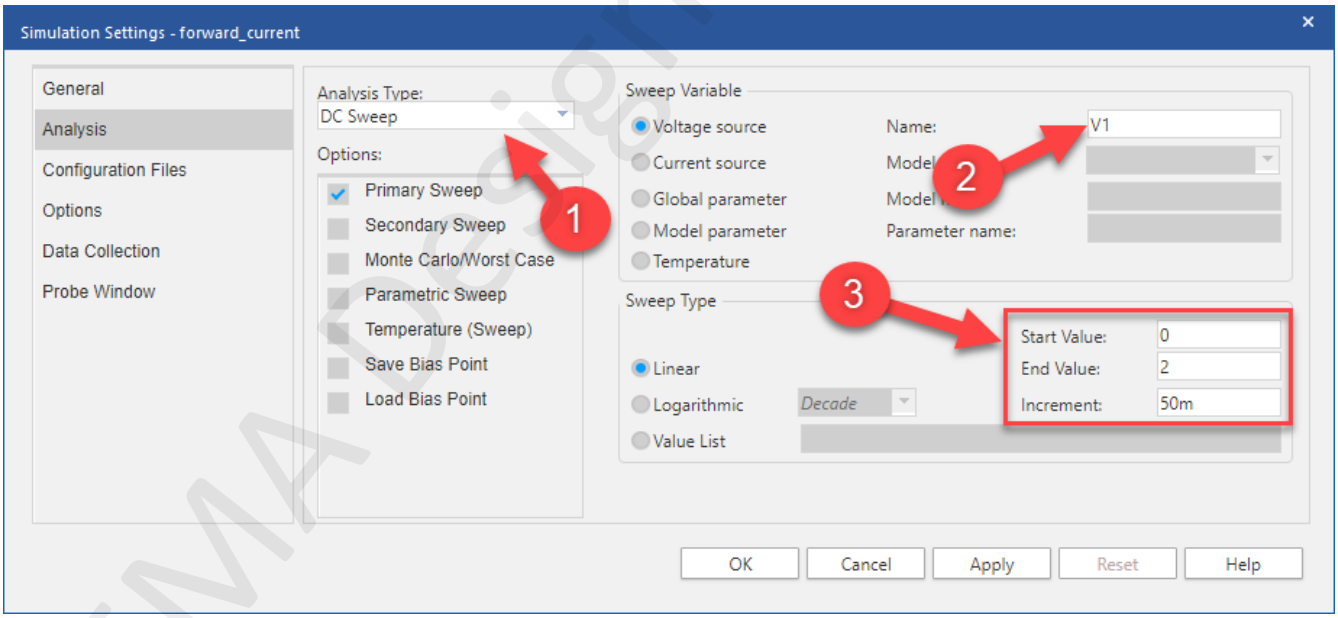
Your new model is now placed and configured. It has an associated graphic and is ready for use in a test design.

Test the Forward Characteristics

1. Place a **VDC** part and a **PSpice 0 ground** and hook them up like this (with the MURS480ET3G diode that was already placed):



2. Edit the forward_current simulation profile so that the Analysis Type is DC Sweep
3. Make the Sweep Variable a Voltage Source whose Name is V1
4. Ensure the Sweep Type is Linear that starts at 0, ends at 2 and increments by 50m

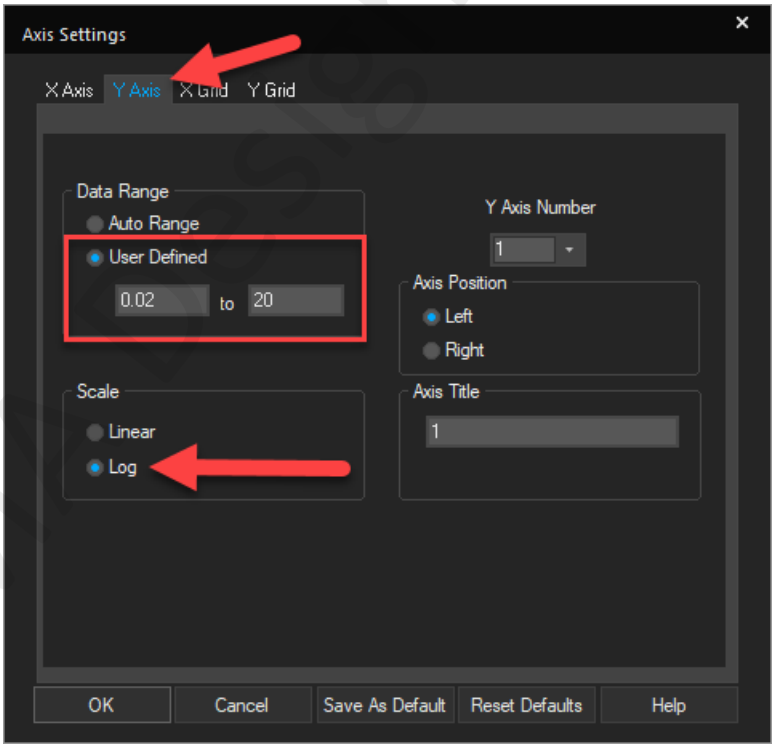


5. Hit OK
6. Run the simulation by choosing PSpice > Run, hitting F11 or selecting the icon

Setup the Display

The PSpice Probe window will open with a black background and the x-axis showing the Voltage of V1 ranging from 0 to 2 volts.

1. Add the new trace by choosing Trace > Add Trace…, hitting the Insert key or selecting the  icon
2. Select I(D1) from the Simulation Output Variables list to populate the Trace Expression at the bottom
3. Hit OK
4. Change the axis settings by choosing Plot > Axis Settings…
5. Go to the Y Axis tab at the top
6. Change the Scale from Linear to Log
7. Change the Data Range from Auto Range to User Defined with a minimum of 0.02 and a maximum of 20 to match the plot in the datasheet

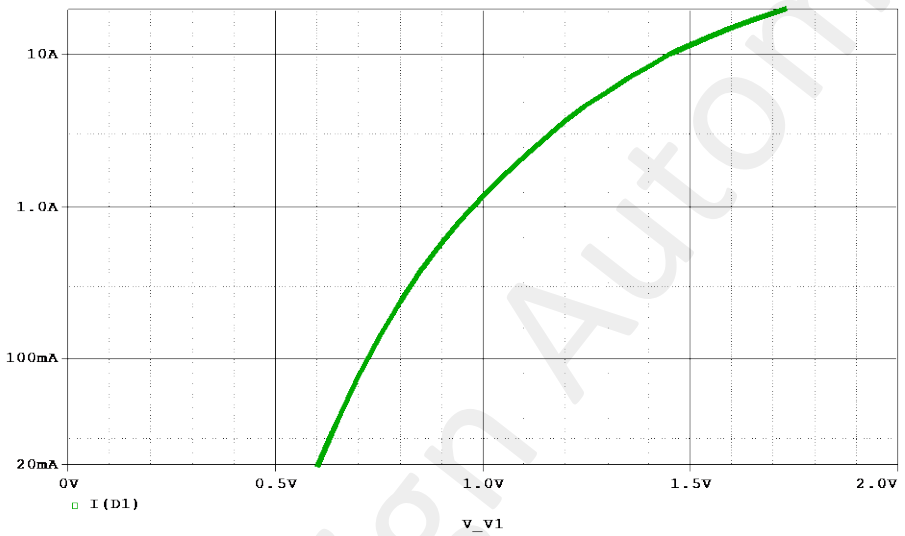


8. Hit OK

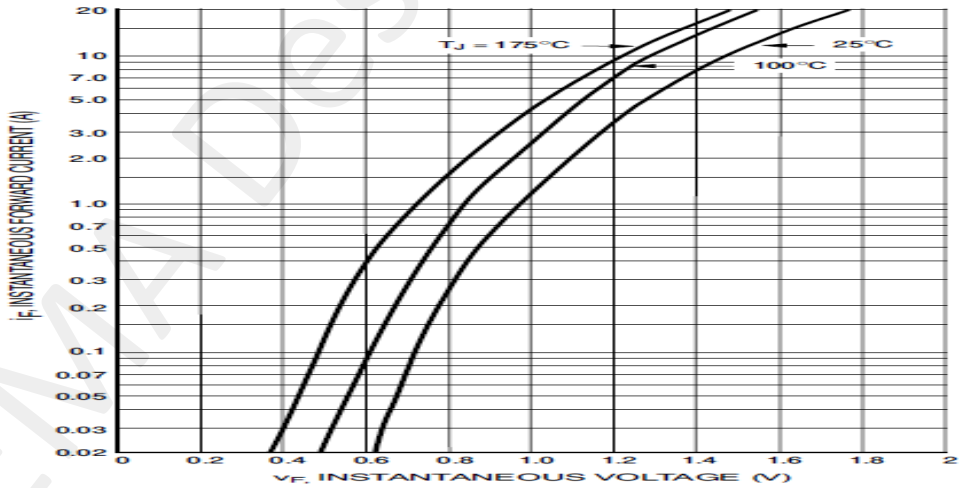
Note that you can also quickly change the scale from linear to log and back by toggling the icon on the Probe taskbar

Verify the Forward Current

1. Check the display of the forward current against the datasheet to ensure that they match



Here's a copy of the datasheet curve again (squished vertically to show matching better):



Lesson 11, Lab 4: Testing the Reverse Recovery

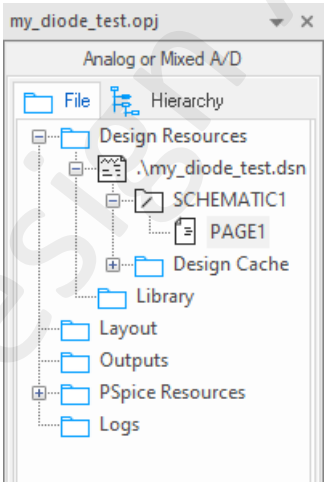
Lab Objectives

- Create a new simulate-able schematic within the same design file
- Verify that Reverse Recovery characteristics

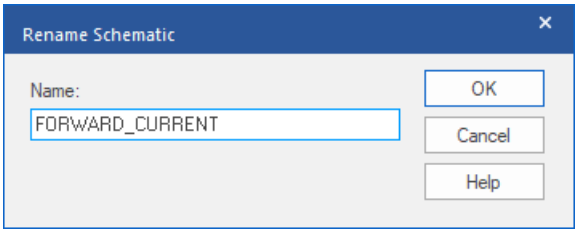
Managing the Schematic Folders

In this section we're going to create a new schematic folder in the design file so that we will ultimately have two simulate-able schematics within one design file. You could just as well create a new design file but this is a clever way of reducing the amount of design files that you will need and is used often by Cadence in their sample designs so it will be good to understand it.

1. Go to the Project Manager which should look something like this



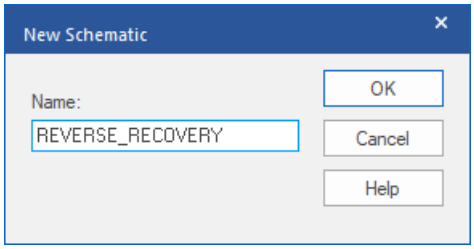
2. Right Mouse Button (RMB) on the SCHEMATIC1 folder and choose Rename to change the Schematic's name to FORWARD_CURRENT



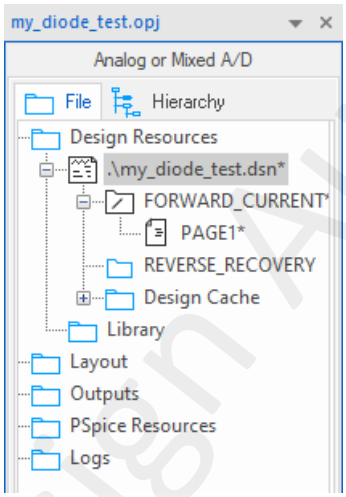
This is not necessary but will make the schematics a little bit easier to understand

3. Right Mouse Button (RMB) on the dsn file itself and choose New Schematic…

4. Give it the name REVERSE_RECOVERY



The project should now look like this:



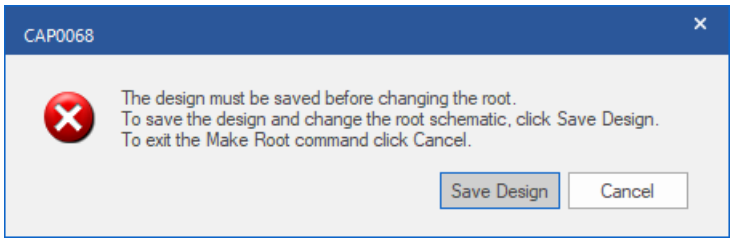
5. RMB on the REVERSE_RECOVERY schematic and choose New Page, the default name of PAGE1 is fine, Hit OK

> We have now made space in our design file for a second schematic to be created and used but in order to simulate it, we need to promote it to the ROOT so the netlister starts in that folder

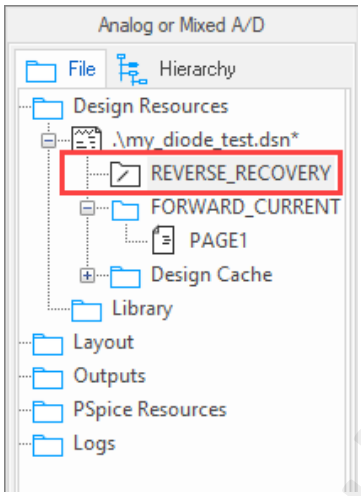
Make Schematic the ROOT

1. RMB on the REVERSE_RECOVERY Schematic and choose Make Root

2. You may get an error that you haven't saved your design yet, if so, select Save Design



3. Notice that REVERSE_RECOVERY is now the topmost folder and it has a tiny slash through the yellow graphic folder on the left indicating that it is the ROOT

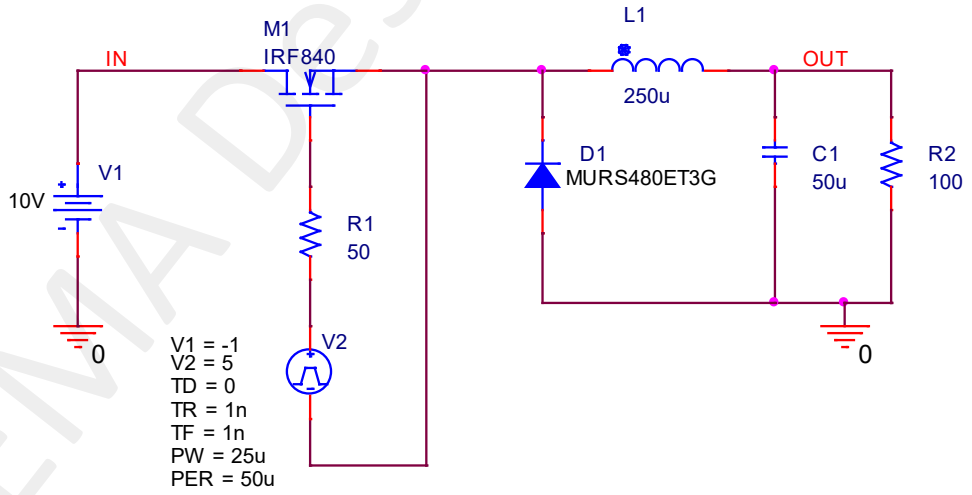


4. Double click on PAGE1 under the REVERSE_RECOVERY folder to start editing that page

Create a schematic to test the Reverse Recovery

To obtain Reverse recovery time, we need to carry out a transient analysis for a time of at least twice the on time of the pulse source. The value of the supply voltage (V1) should be such that it does not drive diode beyond breakdown.

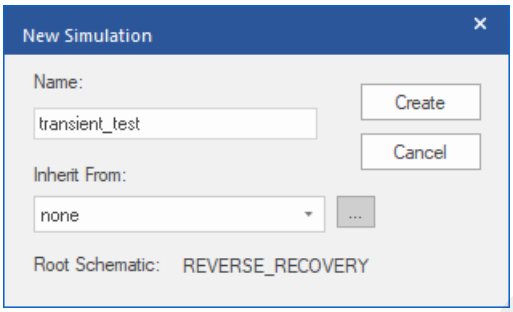
1. Draw a schematic like this



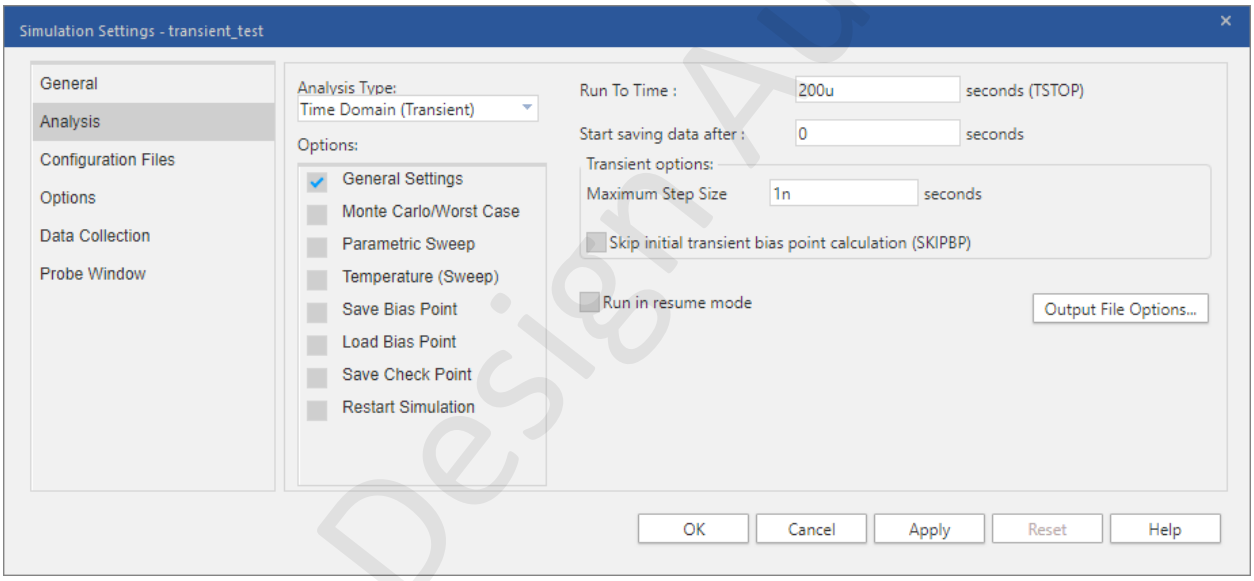
2. The parts are **VDC**, **IRF840** (from the PWRMOS.OLB library in the C:\Apps\SPB_17.4\tools\capture\library\pspice\ directory), **R**, **VPULSE**, **L**, **C** and our diode, **MURS480ET3G**

Since we don't have a simulation profile anymore (because there are none tied to this new schematic folder), we need to create a new one.

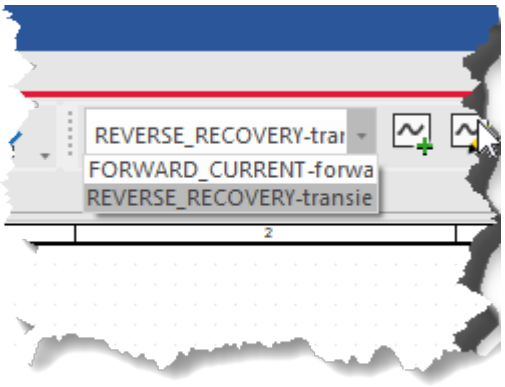
3. Create a new simulation profile (PSpice > New Simulation Profile) called transient_test



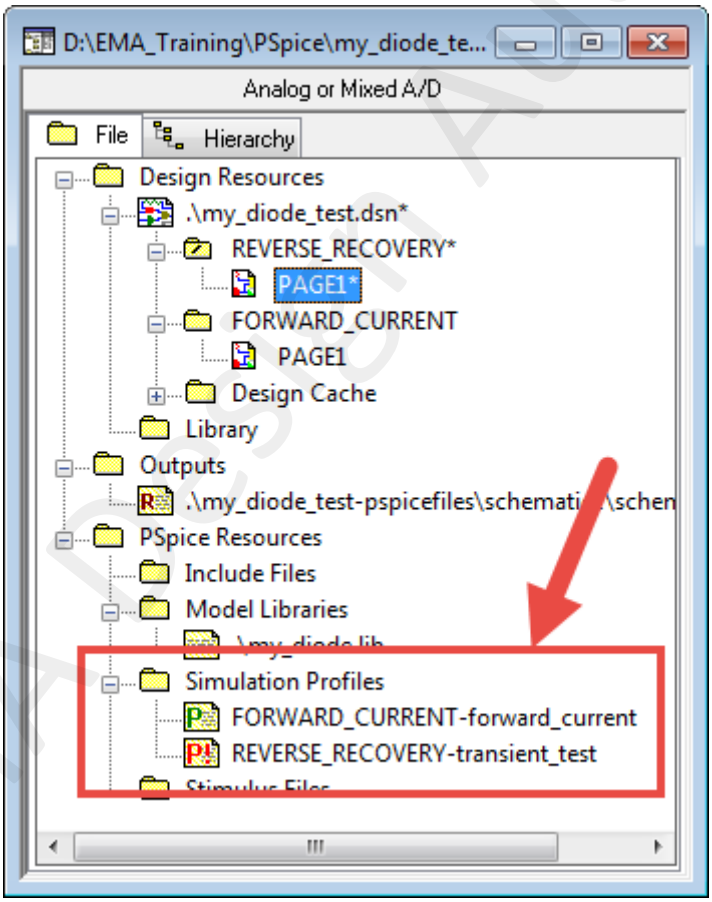
4. Set the Run to Time at 200u and the Maximum step size to 1n, Hit OK



5. Notice that you now have two different simulation profiles in the top left corner and they each point to a different Schematic folder as the root which means that by toggling the simulation profiles, you will effectively toggle which Schematic folder is the ROOT



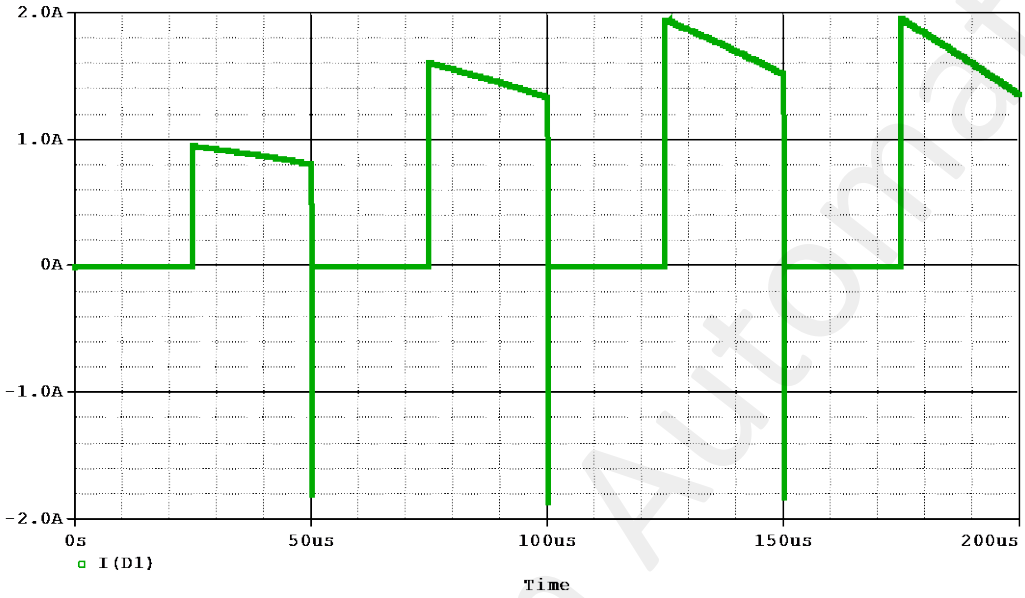
The name is somewhat truncated in the drop-down view so you won't be able to read the whole name. If you want to see the whole name, go to the Project Manager and in the PSpice Resources at the bottom you can see all the different Simulation Profiles. You can even make an inactive simulation profile active by RMB on it and choosing Make Active.



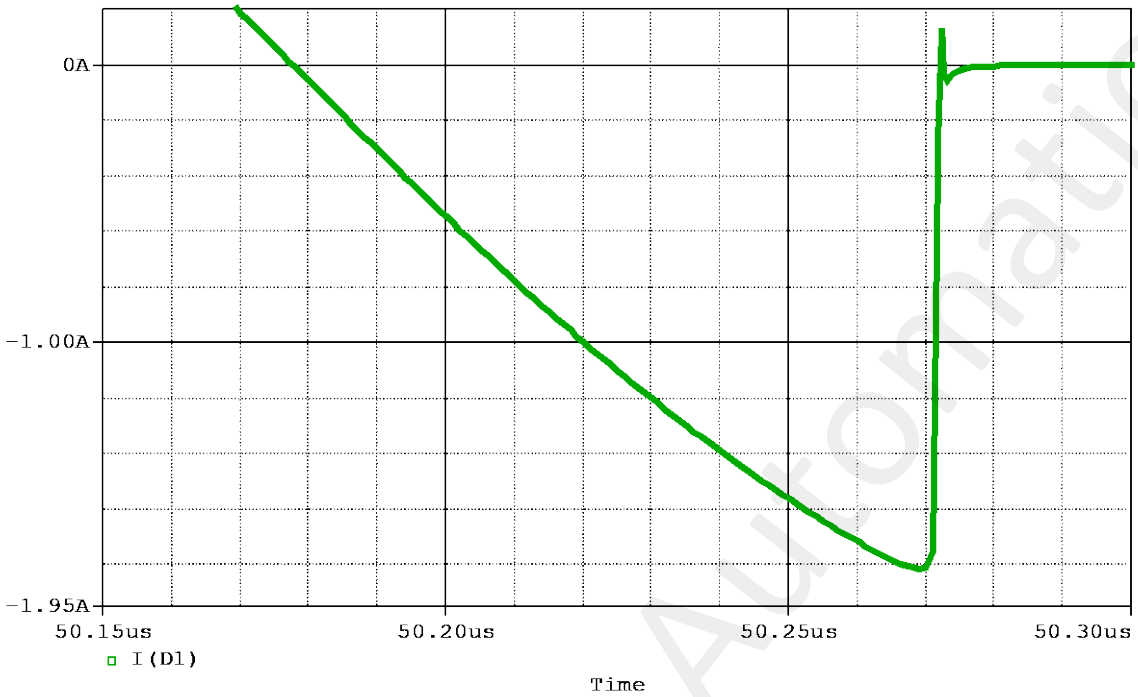
6. Run the Simulation by choosing PSpice > Run, hitting F11 or selecting the icon

Verify the Results

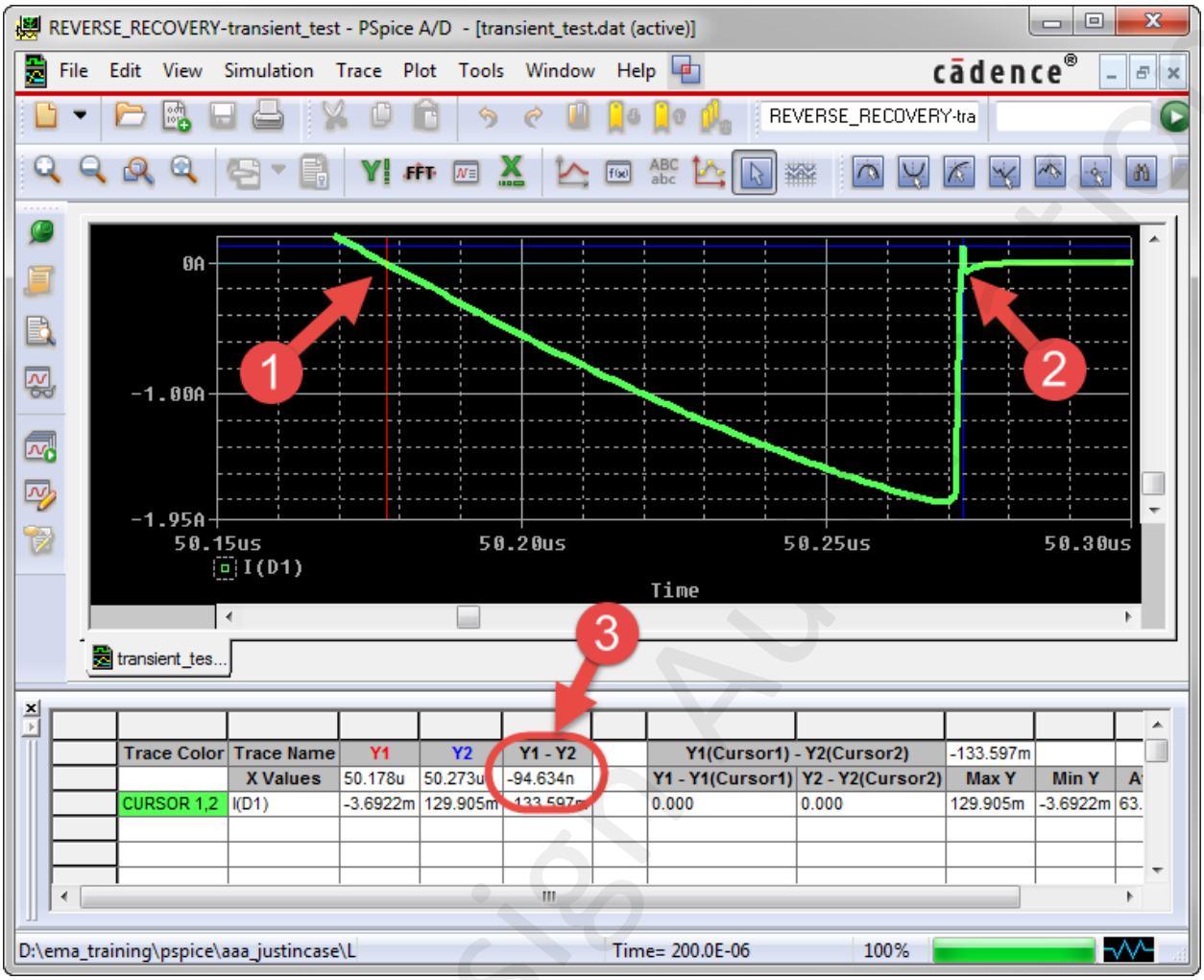
1. In Probe, Add Trace I(D1)



2. Zoom in tightly on the time around 50us where the current briefly reverses using the Zoom In and Zoom Area commands



3. Invoke the cursors  and place cursor 1 where the current crosses the 0A threshold in the negative direction and cursor 2 approximately where it crosses the 0A threshold in the positive direction to measure the time difference between those two points



4. In this screenshot it comes out to approximately 95ns

This value of trr obtained from simulation result (95n) does not match the datasheet exactly (75n), as SPICE models the snap recovery characteristics.

> If you'd like to, you can fiddle with the TT parameter in the model to get a trr value that works better as their relationship is closely tied.

Lesson 12. Monte Carlo Analysis

What Does Monte Carlo Analysis Do?

Monte Carlo Analysis causes the simulator to run multiple runs of the specified simulation (AC Sweep, DC Sweep, or Transient) are performed while parameters are varied. The variance is performed according to a specified distribution, which is UNIFORM by default. This provides statistical data on the impact of a device parameter's variance.

Options Within the Simulation Settings Dialog Box

Analysis Type - Monte Carlo analysis must be run with one of three basic simulations. You cannot perform a Monte Carl analysis by itself. After configuring the basic analysis (AC Sweep, DC Sweep, or Transient) click the Monte Carlo option to access the setup dialog and click the Monte Carlo radio button to activate the Monte Carlo settings.

Output Variable - The output variable is the point in the circuit at which the collating functions are applied. The output variable is either a voltage at a node or the current through an independent current or voltage source.

Number of runs - This is the number of times that the basic simulation runs. The current version has a limit of 10,000 runs.

Use Distribution - This specifies the distribution that is used when applying the tolerances to the devices. PSpice A/D has two distributions built in: Uniform and Gaussian. Uniform is the default distribution.

> You may define your own distributions by clicking the Distributions... button. Please see the PSpice Reference Guide for details on creating custom distribution function.

Random Number Seed - Like every other computerized random number generator, the randomness in Monte Carlo analysis is actually pseudo-random. There is a predefined function that generates a series of numbers based on the initial seed value. Monte Carlo has a possible range of random seeds of odd numbers from 1 to 32,767. If no seed is specified, a default value of 17533 will be used.

Save Data From - Allows you to control what data is saved from the analysis. Choices are: None, All, First, Every, Runs(list). For the last three options you will also need to specify a Run variable. For Example, selecting Every and entering 10 for run saves the data from every 10th run.

Distributions to Choose From

There are three types of Monte Carlo distributions to choose from

- Uniform – equal probability of a component value landing anywhere in the tolerance range
- Gaussian – Bell curve based probability with more likelihood of a component's value being near the nominal value and less as it gets away from nominal. Note that the total range is 3X the tolerance of your component
- GaussUser – Same Bell curve based probability with customizable spread from 1 to 9. A GaussUser value of 1 means that the 3 sigma point will be equal to your tolerance, a GaussUser value of 2 means that the 3 sigma point will be equal to 2x tolerance, etc. The recommended value to use is 1.

You can also create your own distribution if you would like to create something not available here.

Device and Lot Tolerances

Device and Lot tolerances are available for each model parameter. Device tolerances, denoted as DEV within the model statement, cause components that share the same model to vary independently of one another.

Lot tolerances, denoted by LOT in the model statement, cause components that share a common model to vary together.

It is possible to use both tolerances together on a model parameter. In this case, the tolerances add together. If you have the following model:

```
.MODEL CRES RES(R=1 LOT=30% DEV=2%)
```

For each run, R is first assigned a LOT variation between +/- 30%. Each resistor is then assigned a DEV variation between +/- 2% in addition to the LOT variation. Therefore, RC1 and RC2 (both referencing the CRES model) could be as much as 32% away from their nominal value. In any one run, however, they cannot be further than +/- 2% away from their new nominal value, for a total of 4% from each other.

Breakout Components

When applying tolerances to resistors, inductors and capacitors in a design, it is sometimes necessary to use the Rbreak, Lbreak, and Cbreak parts instead of the normal R, L, and C parts.

The normal R, L, and C parts do not reference a model that is accessible to the user while the breakout parts do.

If you are applying only a device tolerance to the components, then you can use the standard parts and edit the Tolerance property. If a resistor is to have a DEV tolerance of 1%, then you can set the value of the Tolerance property to “1%”. The percent sign is required for correct interpretation by the PSpice A/D netlister.

If you are using LOT or LOT and DEV tolerances, you must use the breakout parts and edit the PSpice A/D model.

Simple Monte Carlo Example

The LOT property gives you the ability to add skew to your entire distribution. If used, the LOT property would likely be very small compared to the DEV property.

In this example, for the DEV=10% column, this is the model that was used:

```
.model RMODEL RES (R=1 DEV=10%)
```

For the LOT=10% column, this is the model that was used:

```
.model RMODEL RES (R=1 LOT=10%)
```

Configuring the Analysis

Monte Carlo

Configuring a Monte Carlo analysis requires setting up the output variable, use ditributions, the collating function to use, and optionally a threshold, range and desired direction of the worst-case.

To select Monte Carlo analysis:

1. Select the number of runs
2. Specify an output variable. This is either a voltage at a net or a current through an independent voltage or current source.
3. Select a Use distribution type from the drop down menu
4. Choose random seed

5. Optionally, have the data from selected runs saved by choosing all or scpecifying which runs you would like to see.

Viewing Output File, Probe Plot, and Histograms

The report class that runs varies with different options listed in the command line. Monte Carlo analysis generates the following three classes of reports:

- Model parameter values for each run (that is, the values with the tolerances applied)
- Waveforms for each run, as a function of specifying data collection or by specifying the output variable in the analysis setup
- Summary of all the runs using collating functions

Output is saved to the data file for use by the Probe graphical waveform analyzer. The Probe window offers a special facility through the Performance Analysis feature which can be used to produce histograms of the derived Monte Carlo data.

Lesson 12, Lab 1: Performing a Monte Carlo Analysis

Lab Objectives

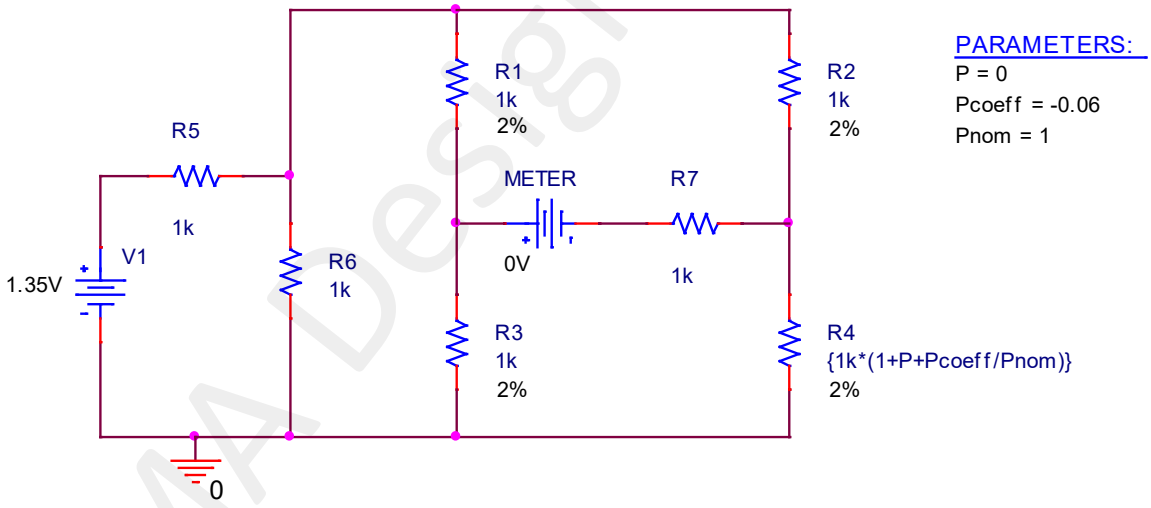
- Configure and run a Monte Carlo analysis
- Examine the results in Probe
- Use Performance Analysis to examine the results

Creating the Pressure Project

1. Create a new project called `Pressure.opj`

2. Create the circuit shown in the graphic below using the **Vdc**, **R**, **PSpice Ground**, and **Param** parts

There are steps outlined below to get all the details correct.



3. Edit the properties of the ***Vdc*** part located in the bridge and set its ***DC Value*** to ***0*** and its ***Refdes*** to ***Meter***

A zero-volt voltage source is commonly used as a current meter in PSpice A/D.

4. Hold **Ctrl** and click ***R1, R2, R3,*** and ***R4*** to select the four resistors

5. Right Mouse Button (**RMB**) on any of the four resistor graphics and **Edit Properties…** open the ***Property Editor***

You should see the properties for the four resistors listed.

6. Set the ***Tolerance*** property for the four resistors to be **2%**. This gives a device tolerance of 2% for the resistance of these four parts

POWER	RMAX	RMAX	RMAX	RMAX
SLOPE	RSMAX	RSMAX	RSMAX	RSMAX
TOLERANCE	2%	2%	2%	2%
VOLTAGE	RVMAX	RVMAX	RVMAX	RVMAX

You can **Display** the tolerance property on the schematic if you would like to.

7. **Close** the Property Editor

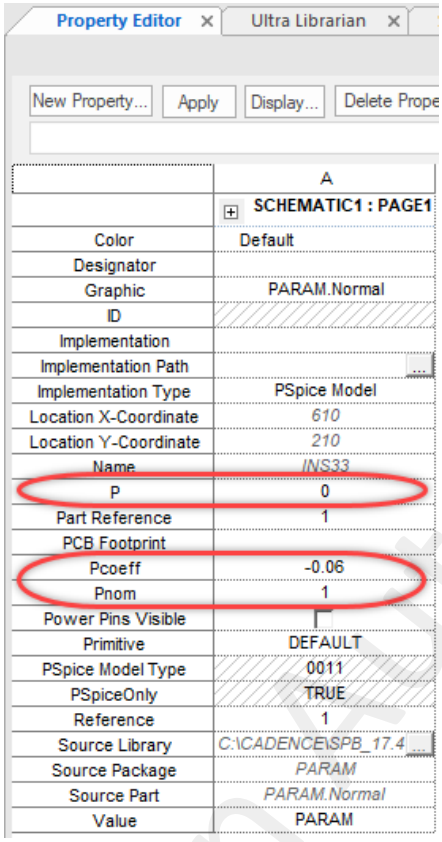
8. After placing the Parameter symbol, highlight the symbol and open the ***Property Editor*** dialog

9. Click the ***New Property*** button

You will use each new row or column you create to define a new Param property in the Property Editor spreadsheet.

10. Add the new properties

- ***Pnom = 1***
- ***P = 0***
- ***Pcoeff = -0.06***



11. To display Param properties and values on the schematic page:

- Highlight the value cell (either Pnom, P, or Pcoeff) in the Property Editor.
- Click the ***Display*** button in the ***Property Editor***
- Select the ***Name and Value*** display option in the ***Display Properties*** dialog box and click ***OK***.

The new Parametric property and value display on the schematic page.

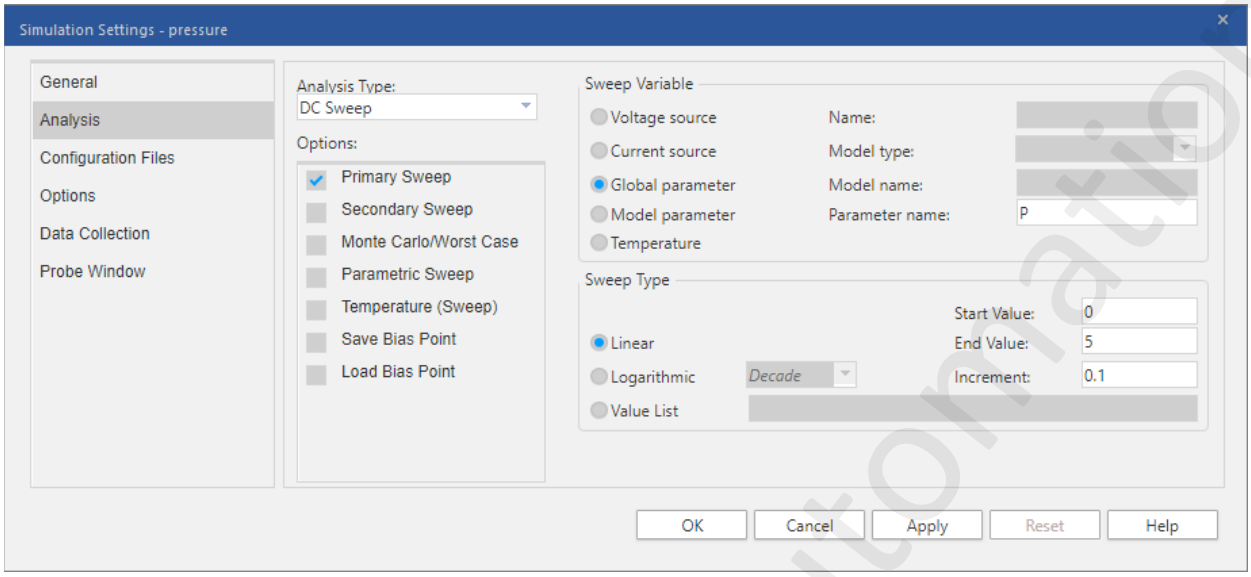
Now, we assign the resistance of ***R4*** to depend on these parameters.

12. Edit the value of ***R4*** to **{1K*(1+P+Pcoeff/Pnom)}**

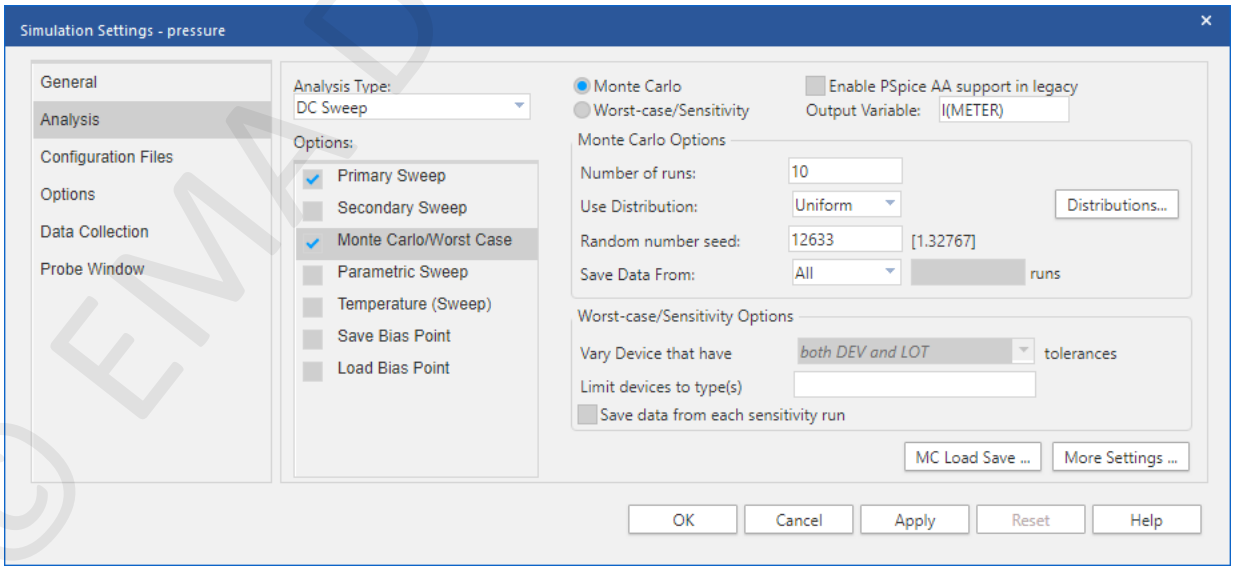
Setting Up the Analysis

First, set up a DC Sweep analysis before sweeping the value of pressure, which is defined as P.

1. Create a new profile (***PSpice > New Simulation Profile*** or  icon) called `Pressure`
2. Configure a DC Sweep analysis of a global parameter **P** sweeping from 0 to 5 in 0.1 increments



3. Check the ***Monte Carlo/Worst Case*** option
4. Click on the Monte Carlo radio button
5. Set the ***Output Variable*** to **I(Meter)**
6. Set the ***Number of runs*** to **10**
7. Set the ***Use distribution*** to ***Uniform*** by selecting it from the pull-down menu box
8. Set the ***Random seed number*** to **12633**
9. Set ***Save data from*** to ***All*** by selecting it from the pull-down menu



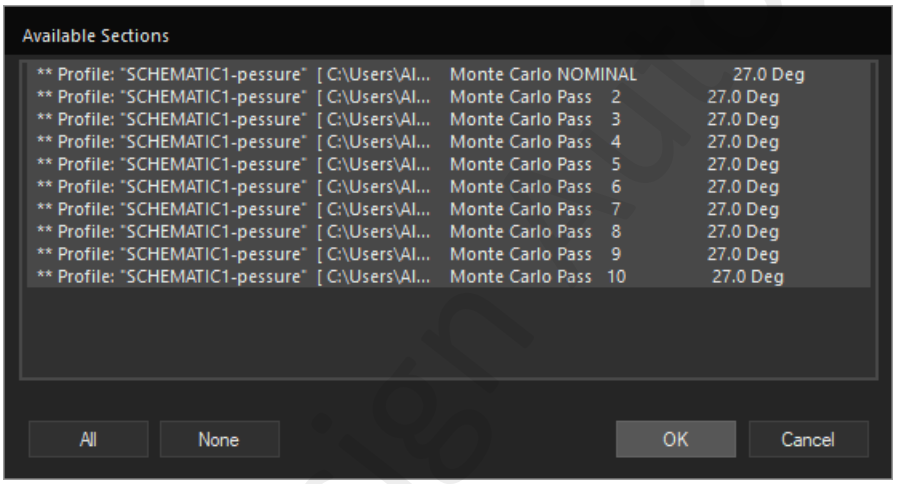
10. Click ***OK*** again to save and close the simulation profile

11. Run the simulation by clicking the toolbar button

Examining the Results

Since a Monte Carlo analysis was performed, you are asked to select which data sections you want to load into the Probe window. By default, all are selected.

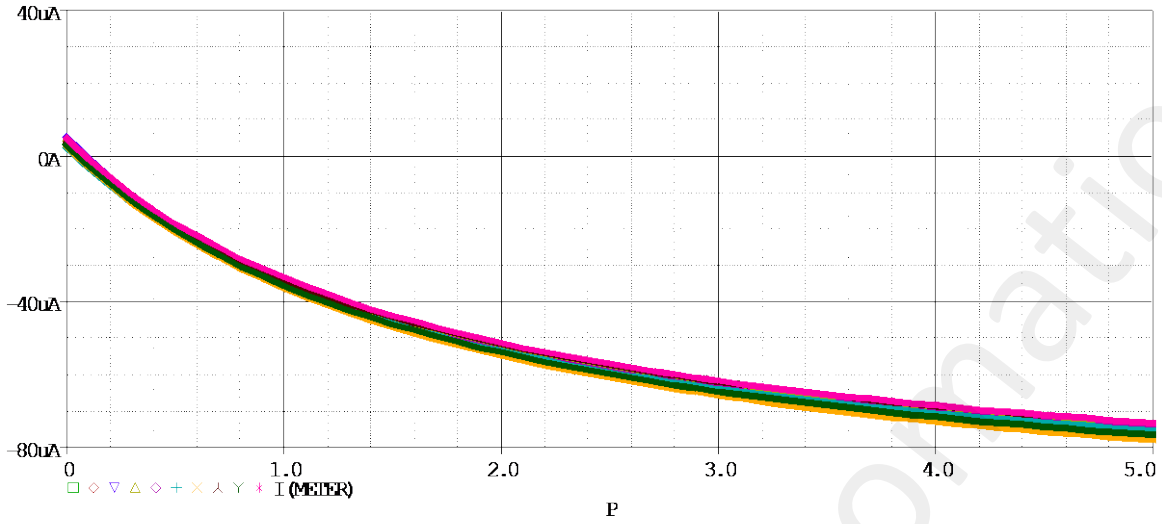
1. Hit **OK** to accept all Available Sections



2. When the Probe window opens after simulation, click the ***Add Trace*** toolbar button. 

3. Select ***I(Meter)***.

4. Click ***OK*** to add the trace.



You can also look at the sorted results in the output file.

Below is a partial listing of the output file, to get there, go to ***View > Output File***

```
Mean Deviation =  14.7930E-09
Sigma          =  15.7980E-06

 RUN              MAX DEVIATION FROM NOMINAL

Pass  9      22.4220E-06 (1.42 sigma) higher at P =  5
             ( 95.996% of Nominal)

Pass  8      20.0850E-06 (1.27 sigma) lower at P =  0
             ( 22.27 % of Nominal)

Pass  3      17.8620E-06 (1.13 sigma) lower at P =  3.9
             ( 103.44% of Nominal)

Pass  5      15.4150E-06 ( .98 sigma) higher at P =  .8
             ( 92.997% of Nominal)

Pass  4      15.2260E-06 ( .96 sigma) higher at P =  5
             ( 97.281% of Nominal)

Pass  7      15.0880E-06 ( .96 sigma) higher at P =  0
             ( 158.39% of Nominal)

Pass  2      12.9200E-06 ( .82 sigma) lower at P =  5
             ( 102.31% of Nominal)

Pass 10      10.8110E-06 ( .68 sigma) lower at P =  0
             ( 58.163% of Nominal)

Pass  6      6.3404E-06 ( .40 sigma) lower at P =  5
             ( 101.13% of Nominal)
```


Lesson 12, Lab 2: Viewing Monte Carlo Histograms

Lab Objectives

- Create and modify Monte Carlo histograms.

Preparing the Circuit

A typical application of Monte Carlo analysis is predicting yields on production runs of a circuit board. Probe can be used to display data derived from Monte Carlo waveform families as histograms, which is part of Probe's performance analysis feature.

This Chebyshev filter is designed to have a 10kHz center frequency and a 1.5 kHz bandwidth. The components were rounded to the nearest available 1% resistor and 5% capacitor value. In our analysis, we are concerned with how the bandwidth and the center frequency vary when 1% resistors and 5% capacitors are used in the circuit.

Setting up the Analysis

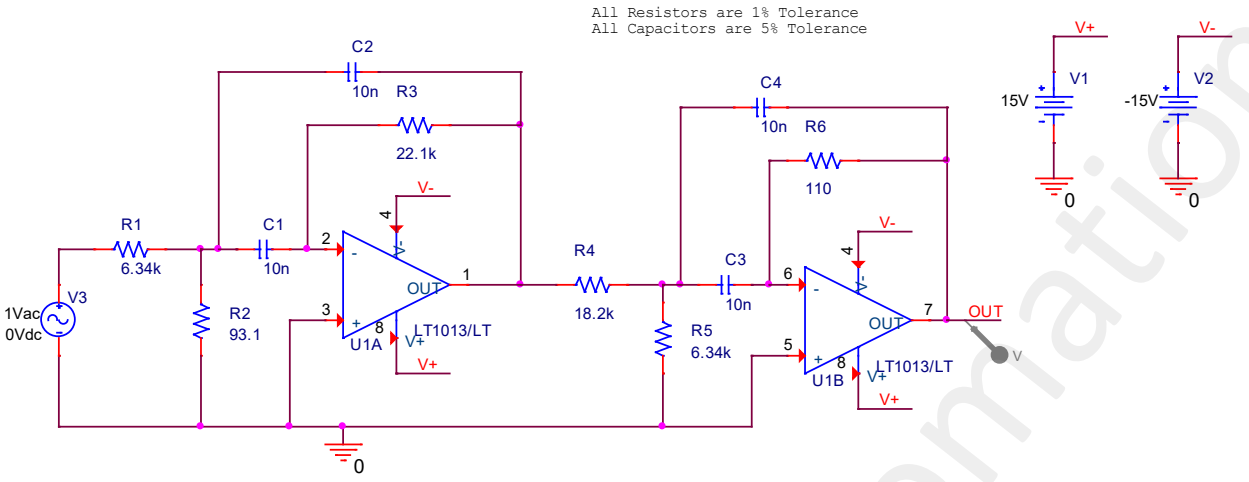
To analyze the filter, we will set up both an AC analysis and a Monte Carlo analysis. The AC analysis sweeps 51 points per decade from 100Hz to 1MHz. The Monte Carlo analysis is set to take 100 runs. The analysis type is AC and the output variable that we are interested in is V(OUT).

The data file can become quite large when running a Monte Carlo analysis. Since you are only interested in the output of the filter, you can place a voltage marker on the output of the filter, then set up the simulation to collect data only at the marked node.

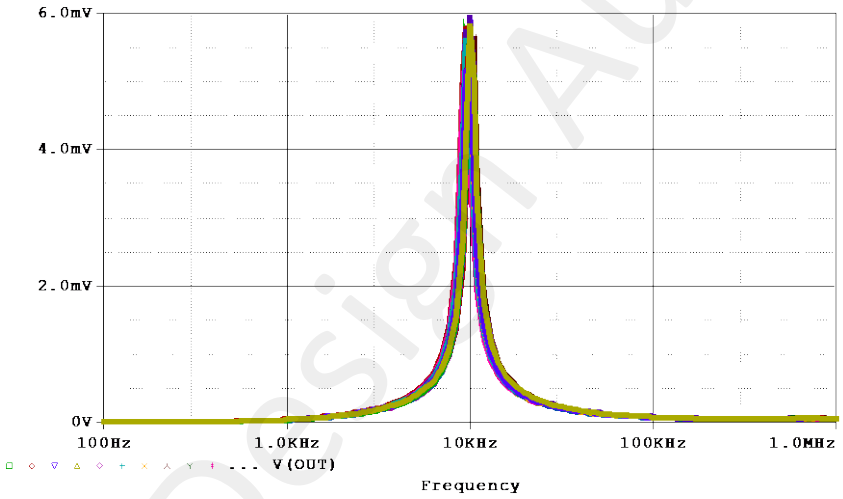
Creating the Histogram

This circuit is not hard to capture, just time consuming so feel free to open up the completed drawing instead of drawing it all.

1. Open the file `Lab2_Cheby.opj` from the `Solutons\Les12_MonteCarlo` folder



2. Run the Analysis



3. Click the ***Performance Analysis*** toolbar button.  (next to the FFT button)

4. Click the ***Add Trace*** toolbar button. 

The following 4 steps describe one way of quickly generating the desired trace expression. You may also just type in the expression as documented below:

5. Select the `Bandwidth(1,db_level)` goal function from the Functions or Macros section

6. Without moving the insert cursor, click on the V(out) trace name

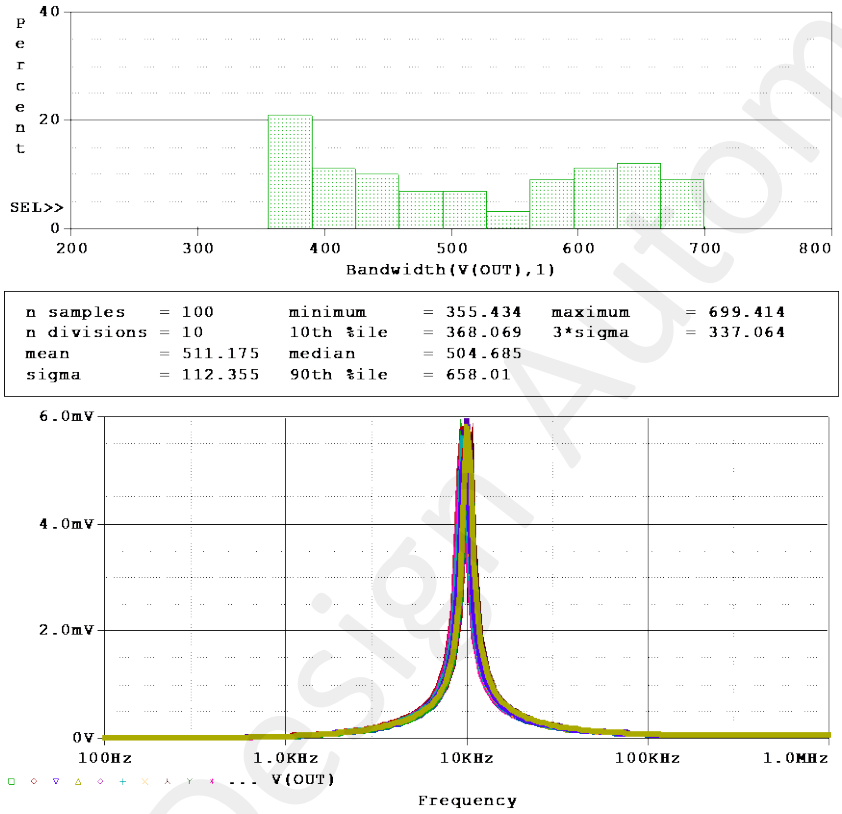
7. Without moving the insert cursor, type **1**

8. Move the cursor to after the V and insert **db**

The trace name is now `Bandwidth(Vdb(OUT),1)`.

9. Click ***OK*** to view the histogram

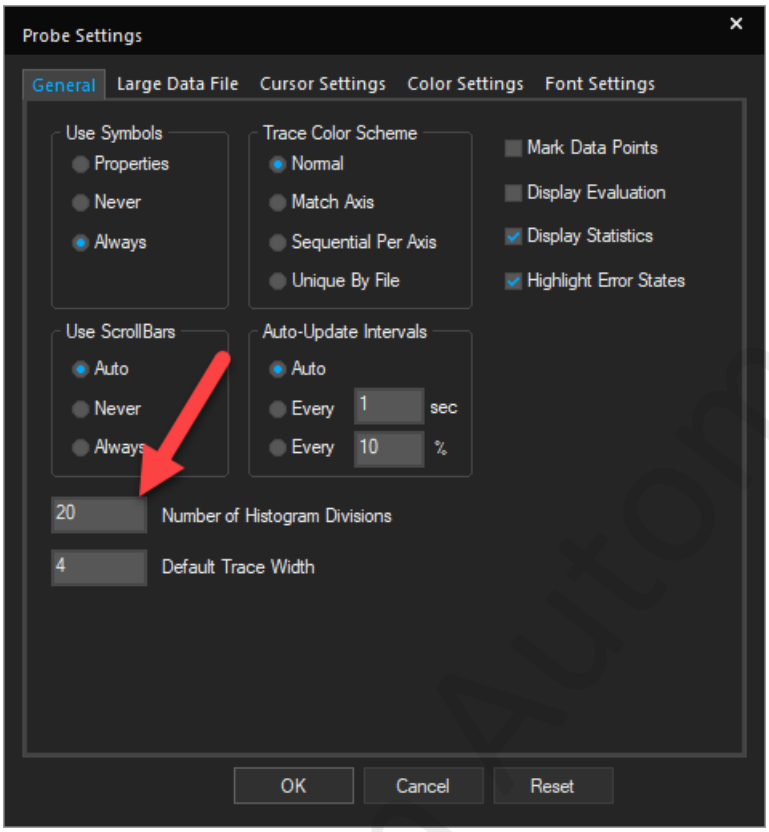
Since a Monte Carlo analysis is random, your results may vary slightly from those shown in the following graphic.



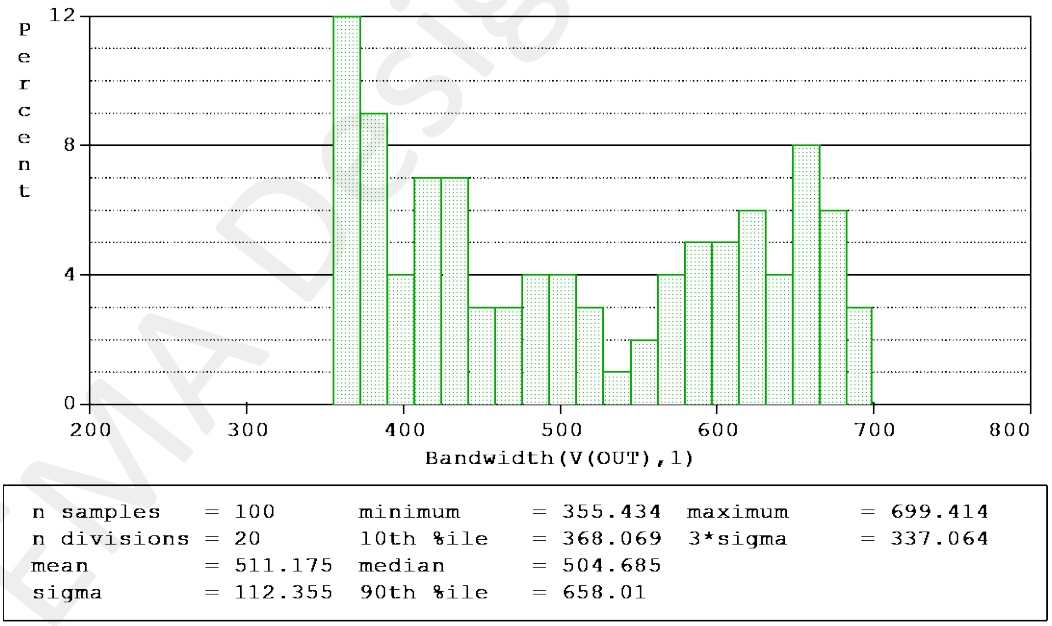
This shows you the bandwidth at 1dB down from center that you can expect to see based on the 1% resistor and 5% capacitor tolerances for 100 random simulations.

Changing the Number of Histogram Divisions

1. Choose ***Tools > Options*** and enter **20** in the Number of Histogram Divisions text box



2. Click ***OK***



The statistics for the histogram are displayed by default. However, this may be turned off by choosing **Tools > Options** and then unchecking Display Statistics.

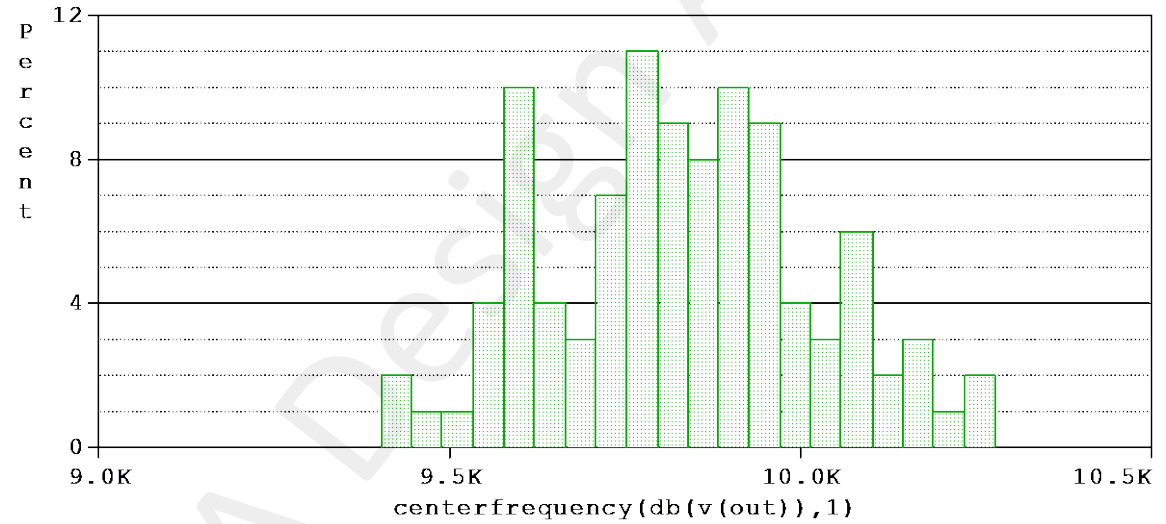
Viewing the Center Frequency Distribution of the Filter

1. Click the ***Add > Trace*** toolbar button
2. Select the ***CenterFrequency(1, db_level)*** goal function by clicking on it
3. Select ***V(out)*** by clicking on it
4. Without moving the insert cursor, type **1**
5. Move the cursor to after the V and insert **db**

The Trace Command box should read CenterFrequency(Vdb(OUT),1).

6. Click ***OK*** to view the histogram

The new histogram replaces the old one. To view both simultaneously, choose ***Plot > Add Plot*** before adding a new trace. The histogram for center frequency is shown in the graphic that follows.



n samples	= 100	minimum	= 9402.02	maximum	= 10277.2
n divisions	= 20	10th %ile	= 9583.57	3*sigma	= 562.033
mean	= 9830.77	median	= 9819.37		
sigma	= 187.344	90th %ile	= 10074		

These results show you the variation in center frequency that you will get with 1% resistors and 5% capacitors. If this is within your specifications, you are complete but if not, you can further reduce your tolerances and determine what values you will need to obtain an adequate yield.

Additional Exercise

- Experiment with creating histograms using the CenterFrequency and Bandwidth functions; try changing the dB level from 1 to 3.
- Experiment with other Performance Analysis functions.

Lesson 13. Worst Case Analysis

What is Worst Case Analysis?

Worst-Case analysis causes the simulator to perform sensitivity and Worst-Case analyses of the circuit. Multiple runs of the specified simulation are performed while parameters are varied. Unlike Monte Carlo analysis, only one parameter is varied per run. During each run, PSpice A/D calculates the sensitivity of the declared output variable to the variance in the modified parameter. Once all of the sensitivities have been calculated, a final simulation is performed with all parameters assigned their maximum deviation from the nominal based on their individual sensitivity runs.

What is Sensitivity Analysis?

For Worst-Case analysis, you must define what you consider to be the measure of worst-case. There are five collating functions from which to choose the definition of worst-case:

- YMAX finds the greatest difference in each waveform from the nominal run.
- MAX finds the maximum value of each waveform.
- MIN finds the minimum value for each waveform.
- RISE_EDGE<value> finds the first occurrence of the waveform crossing above the threshold value. This function assumes that there will be at least one point that lies below the specified value followed by at least one above.
- FALL_EDGE<value> finds the first occurrence of the waveform crossing below the threshold value. This assumes that there is at least one point above the value followed by at least one point below.

Assigning Tolerances

In order to perform a Worst-Case analysis, you must assign tolerances to devices in the circuit for which you want the values to vary. These are assigned in the `.MODEL` statement in the same way they are assigned for a Monte Carlo run.

Device Tolerance

When a device tolerance is applied to a model parameter, every device that references that model varies independently. For example, if we have the following resistor model:

```
.MODEL My_res RES (R=1 DEV=2%)
```

every resistor in the circuit that references the My_res model varies randomly from one another by a maximum of 4% (a range of +2% to –2% from the nominal value)

Simple Worst Case Example

The Worst-Case Algorithm

When a Worst-Case analysis is performed on a circuit, the following procedure is followed:

7. A simulation is performed with all variables in the circuit set to their nominal values.

8. A series of sensitivity analyses are performed. A single variable is adjusted in the positive direction by RELTOL (default is .1%) and the effect on the output variable, as measured by the collating function, is noted. Another simulation is performed adjusting the same parameter in the negative direction by RELTOL and the effect on the output variable, as measured by the collating function is noted. The variable is returned to its nominal value and the simulation proceeds using the next variable until all variables with tolerances have been tested.

9. A final simulation is performed with all variables taken to the limit of their specified tolerance in the direction that gave the worst result as measured by the collating function.

> When sensitivity analyses are performed, only the LOT tolerance is used. But when the final simulation is performed, the value of the varied parameter is adjusted by DEV + LOT.
>
> This algorithm does not take into account interdependence between variables. In other words, the simulation does not explore the effects one device has on another. It is entirely possible that even though moving R1 in the positive direction and moving R2 in the negative direction gave the greatest deviations in the sensitivity runs, when both are applied together they may cancel one another out rather than adding.

Configuring the Worst Case Analysis

Configuring a Worst-Case analysis requires setting up the output variable, Worst-Case/Sensitivity Options, the collating function to use, and optionally a threshold, range and desired direction of the worst-case.

To select Worst-Case/Sensitivity analysis:

1. Specify an output variable. This is either a voltage at a net or a current through an independent voltage or current source.

2. Select whether you want to vary devices with only a LOT tolerance, only a DEV tolerance, or devices with both DEV and LOT tolerances specified.

3. Optionally, list the types of PSpice A/D devices to which you want Worst-Case applied. You list the devices by specifying their SPICE symbol. For example, “R” for resistors, “Q” for bipolar transistors, etc.

4. Optionally, have the data from each sensitivity run saved by checking the box. If you don’t check this box, then the sensitivity data is omitted from the output file.

Clicking the More Settings button opens the following dialog.

5. Select the collating function that defines the worst-case for your analysis.

6. If you chose a collating function that has the option of specifying a threshold, type in the value for that threshold.

7. If you want the function applied to a range, then enter the range. The range refers to the portion of the swept variable (x-axis in the Probe window) to which the function is to be applied.

8. Chose if you want the worst-case direction to be HI or LO. As you might expect, the MAX and MIN functions ignore this setting and report the maximum or minimum.

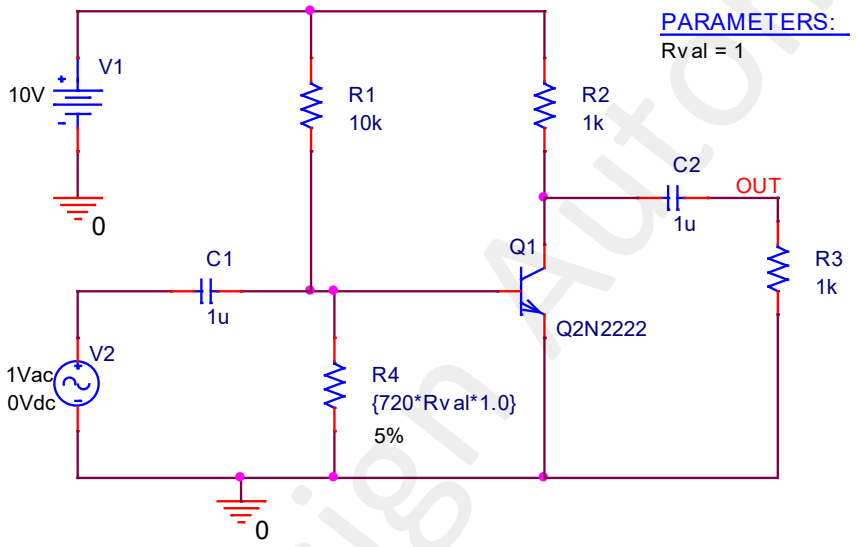
9. Click the check box for saving parameter values to the output file if you would like to have them available. Otherwise, they will not be saved.

Lesson 13, Lab 1: Working Worst Case

Lab Objectives

- Look at examples where the Worst Case is correctly calculated

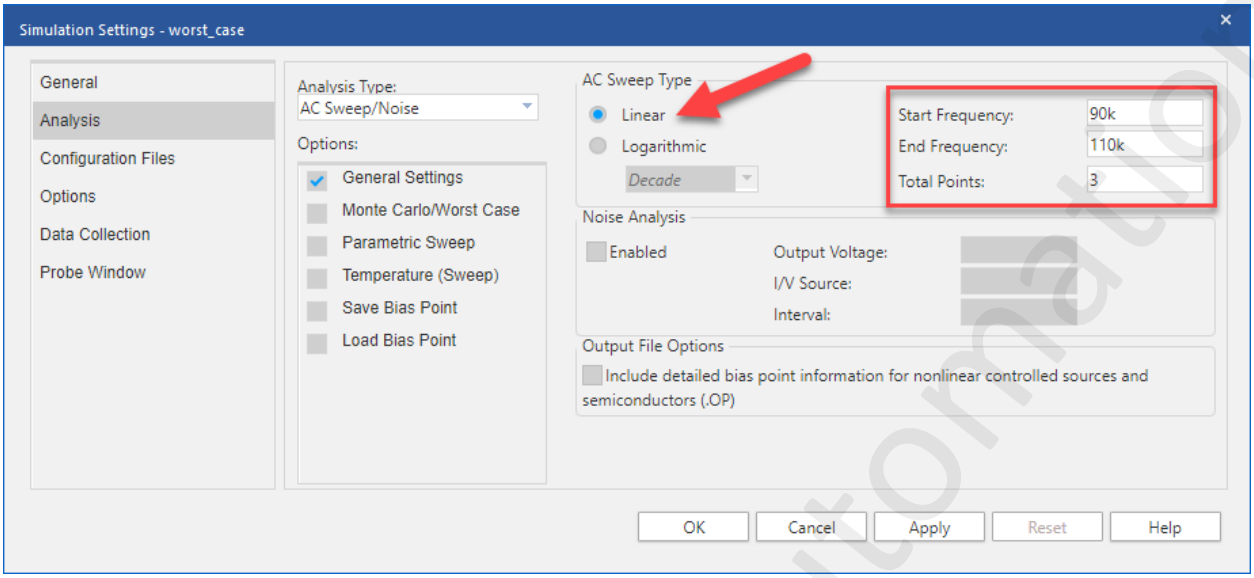
Creating the Circuit



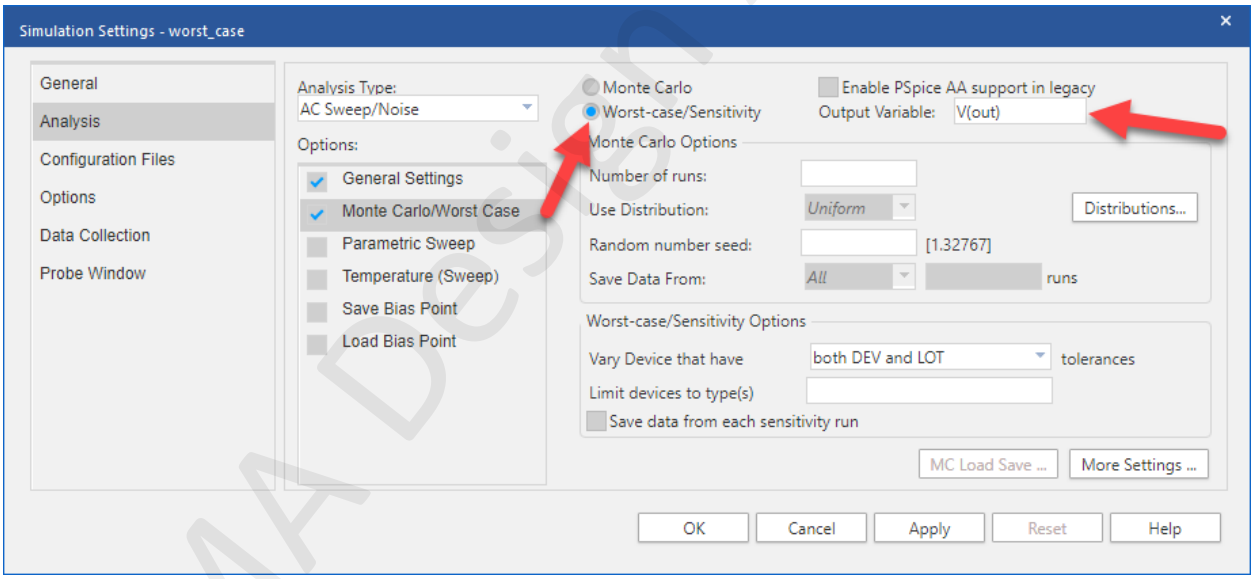
1. Create a new design called `Worst_Case` based on an empty PSpice Schematic
2. Enter the circuit as shown above. The circuit uses the parts: **VDC, VAC, R, C, Q2N2222** (from the bipolar library) and **PARAM**
3. R4 has a VALUE of **{720*Rval*1.0}**, its **TOLERANCE** is set to **5%** and displayed on the schematic. No other components have any TOLERANCE value set
4. **C1** and **C2** are **1u** and **R1** is **10k**

Create a new Simulation Profile to find the Worst Case

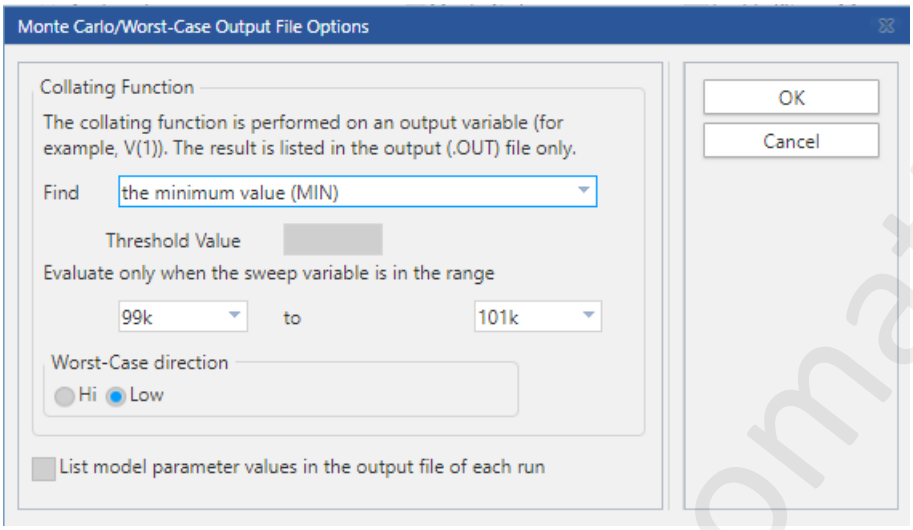
1. Create a new profile (***PSpice > New Simulation Profile*** or icon) called **worst_case**
2. For Analysis Type, select AC Sweep/Noise with the values shown below:



3. Select the Monte Carlo/Worst Case toggle button and fill that resulting window with the values shown below:

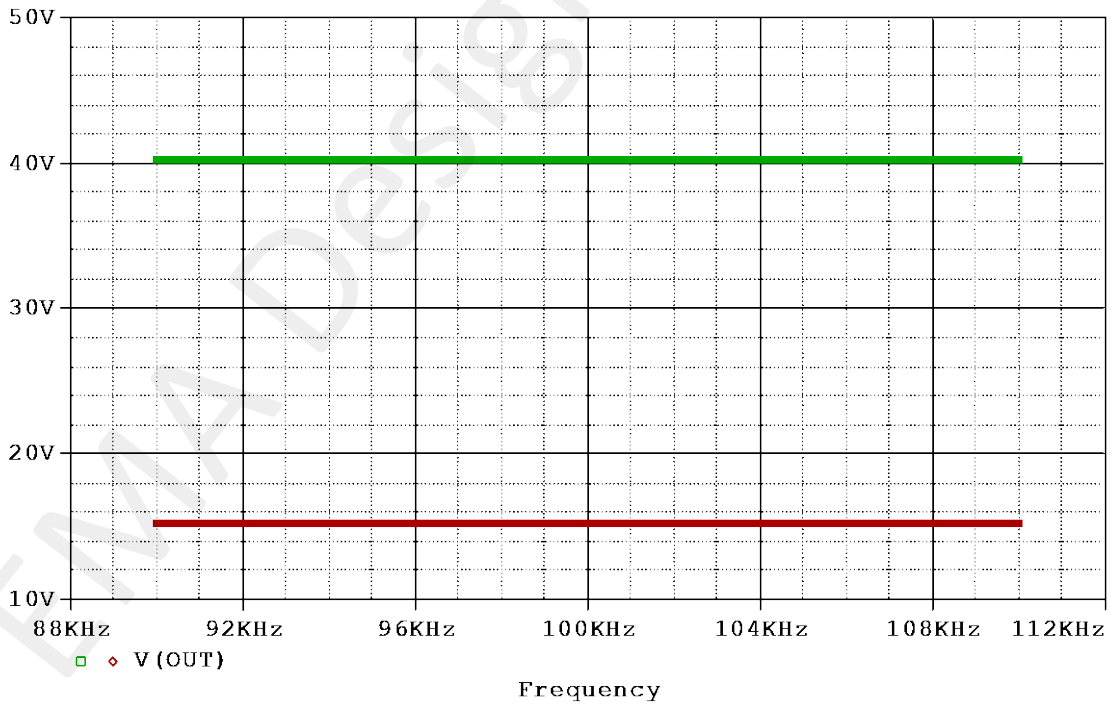


4. Click on the More Settings… button and configure the additional settings
5. Choose to Find **the minimum value (MIN)**
6. Type in **99k** and **101k** into the "Evaluate only when the sweep variable is in the range" drop downs as shown below:
7. Change the worst case direction to **Low**



What this should do is run a Worst Case Analysis with the intention of finding the minimum value of the voltage at the OUT node at 100KHz

8. Hit **OK** twice to close the dialogs
9. **Run** the Simulation and choose **All** again for the Available Sections, Hit **OK**
10. Add Trace V(OUT)



What we're being shown is the nominal value of V(OUT) of close to 40V (green trace) and the worst case value in the minimum direction of V(OUT) of close to 15V (red trace). Keep these numbers in mind for the next section.

In the next steps, we want to verify that these worst case findings are correct.

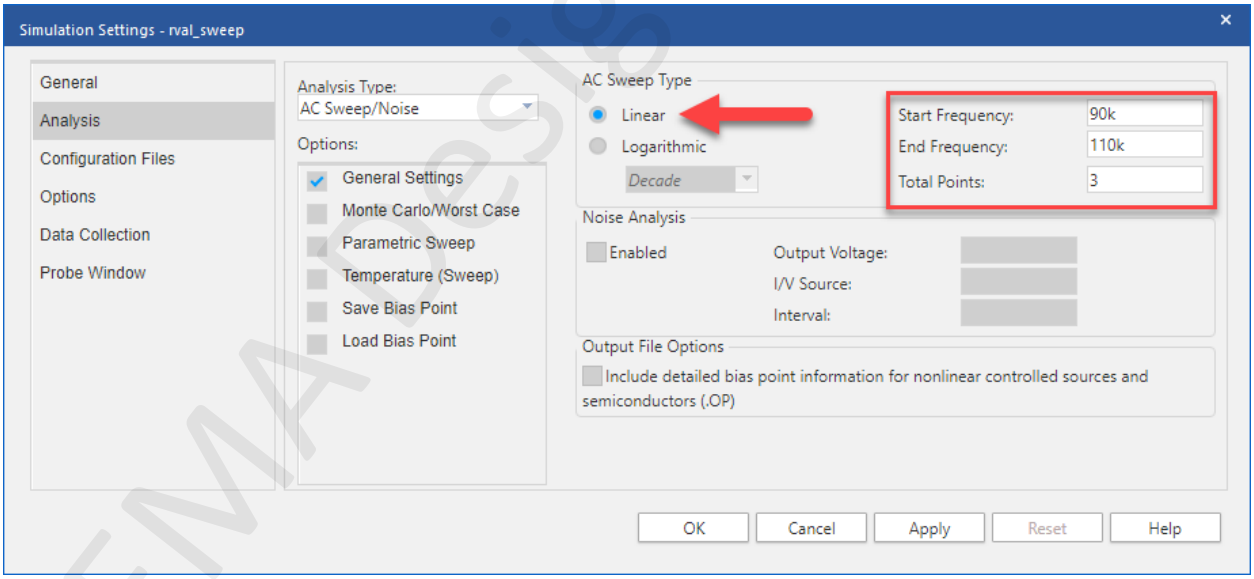
Creating a New Simulation Profile to Sweep RVAL

By limiting the number of components with a tolerance to just one (R4), we have allowed ourselves an easy way to verify the worst case analysis. What we want to do is verify what the worst case analysis results ***should be*** by sweeping through the complete range of possible values that R4 could be given its tolerance range.

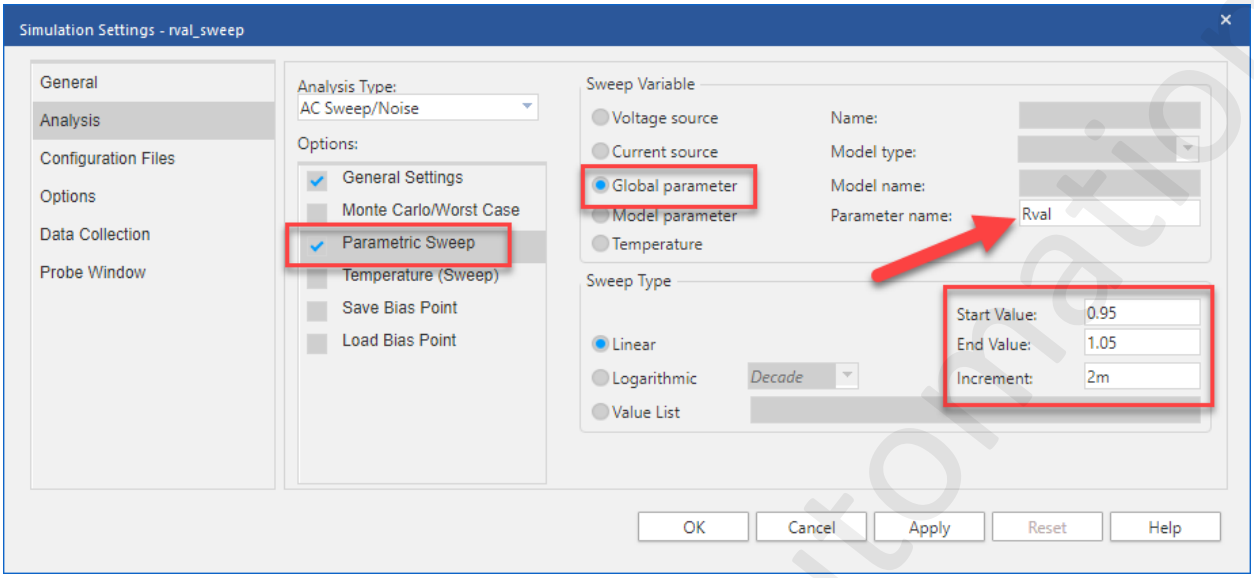
To do that, we're simply going to sweep R4 from 95% to 105% of its nominal value and check to see what the worst case results are and at which value of R4 they occur. We will then verify these results by doing a worst case analysis.

1. Create a new profile (***PSpice > New Simulation Profile*** or  icon) called ***rval_sweep***

2. For Analysis Type, select ***AC Sweep/Noise*** with the same values as before (shown below):



3. Select the ***Parametric Sweep*** toggle button and fill that resulting window with the values shown below:



This will allow our R4 resistor to be swept from 95% to 105% of its 720 Ohm value simulating its overall range of possible values.

4. Hit ***OK*** to close the dialog.

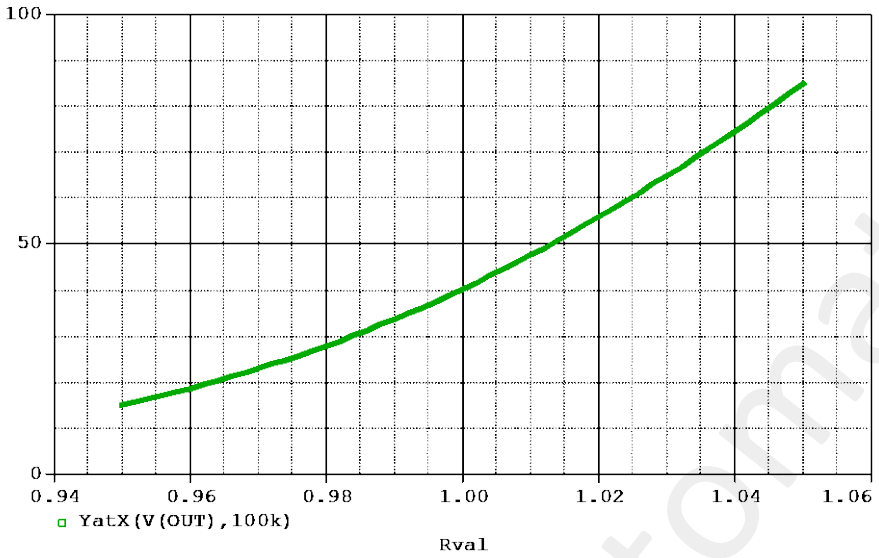
5. ***Run*** the Simulation and choose ***All of the Available Sections*** for display

The x-axis will be displaying Frequency but we want the x-axis to be the Rval parameter as it is being swept. To change the axis, we need to move into Performance Analysis mode.

6. You can change this by selecting the ***Performance Analysis button***  (which is next to the FFT button). This should change your x axis from Frequency to Rval.

7. On our graph we would like to see the voltage at the OUT node of our circuit at 100k as we sweep through our Rval parameter. To do this, add a trace (***Trace > Add Trace***) and type **`YatX(V(OUT), 100k)`**

That should give you results that look like this:



What this is telling us is that when Rval is at its nominal value (1.00), the output will be approximately 40V and the worst case in the low direction for voltage will occur when Rval is set to its lowest value of 0.95 (or maximum negative tolerance) where it will be about 15V.

Good news! This corresponds to what we saw above when we ran the WC analysis.

* Optional Exercise

- Using cursors, can you find the exact voltage values at Rval = 1 and Rval = 0.95? See next line for answers and the method.
- 40.286V and 15.169V respectively; Toggle Cursors  then use the Cursor Search feature  and the syntax **search forward x value (1)** to find the Rval = 1 value.

Lesson 13, Lab 2: Misled Worst Case

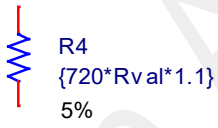
Lab Objectives

- Look at an example where the Worst Case analysis is misled and does not find the true worst case

Editing the Resistor Part

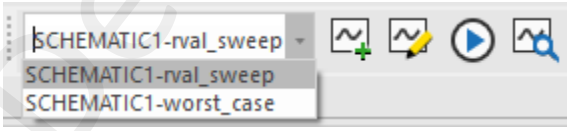
Using the same design and simulation profiles that we set up in the previous lab, we want to now tweak the R4 part to drive the BJT into saturation and eliminate the gain.

1. We want to change the nominal value of R4 by +10% which you can do by changing the value to {720*Rval*1.1} or {792*Rval}

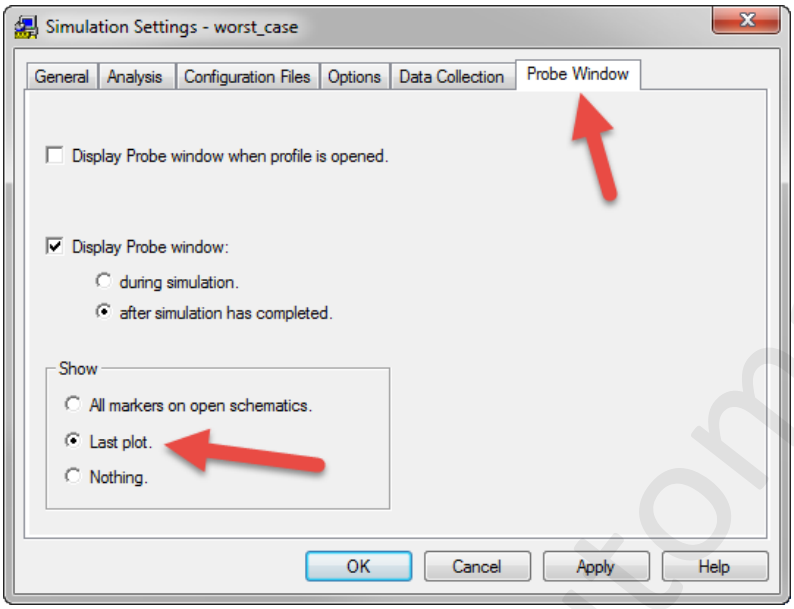


Run the Worst Case Analysis

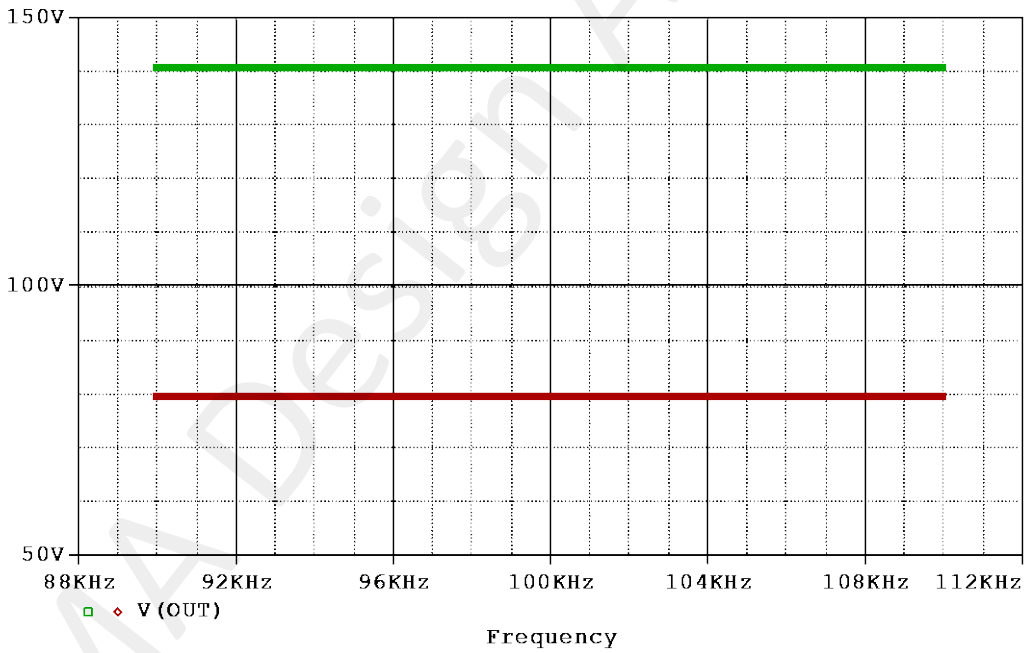
1. In Capture, change the simulation profile back to ***worst_case*** (upper simulation drop down)



2. Before running the simulation, we would like to edit the simulation profile to keep the results that we had displayed last time. To do that, go to ***PSpice > Edit Simulation Profile > Probe Window*** tab and under the ***Show*** section select '***Last Plot***'



3. ***Run*** the simulation keeping all the ***Available Sections***. Your results should look like this:



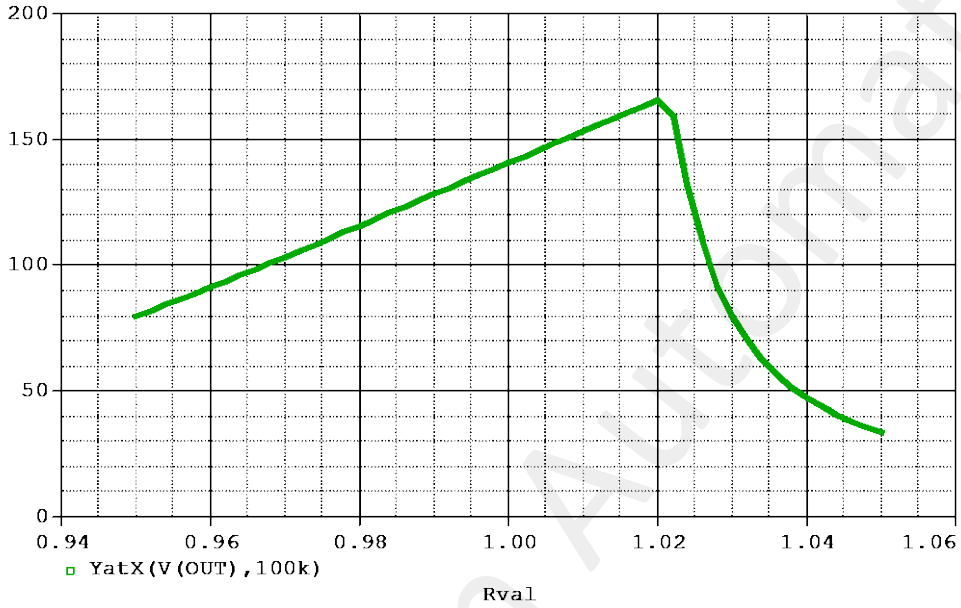
What we're seeing is the nominal value of V(OUT) of close to 140V (green trace) and the worst case value in the minimum direction of V(OUT) of close to 80V (red trace). Keep these numbers in mind for the next section where we'll sweep the resistor value across its tolerance range.

Run the Rval Parametric Sweep

1. In Capture, change the simulation profile back to ***rval_sweep***

2. Before running the simulation, change the ***simulation profile*** to keep the format of the plot the last time we ran the simulation by choosing '***Last Plot***'. Hit ***OK*** to close the dialog

3. ***Run*** the simulation keeping all the ***Available Sections***. Your results should look like this:



Notice now at the Nominal value of Rval the voltage at the OUT node is now about 140V and the very worst case appears to be about 35V going in the positive direction for Rval (on the right side).

What We Are Seeing

The interesting thing that's happening here is that a sensitivity analysis for the component will reveal that as the component goes up a little bit, the voltage also goes up. Since we're trying to find the minimum voltage value it will then conclude that for this part, it must set the component to the most negative tolerance value which will yield a misled worst case result of about 80V (on the leftmost side).

When we swept the R4 part we saw that the minimum worst case voltage was about 35V but when we ran the worst case analysis, it found a worst case of 80V instead. This was because the initial sensitivity analysis indicated that the tolerance would need to be set to the minimum value to get the minimum worst case which was not correct. This confirms what we suspected when we ran the Rval sweep previously and shows how the Worst Case analysis can sometimes be misled.

Lesson 14. Analog Behavioral Modeling

Analog Behavioral Modeling Defined

The Analog Behavioral Modeling (ABM) feature provided in PSpice A/D allows for flexible descriptions of the behavior of electronic components in terms of a transfer function or a look-up table of values. PSpice A/D interpolates linearly between the points on the table. A mathematical relationship is used to describe a circuit segment, so the segment does not need to be either modeled discretely or designed component-by-component.

Where to Find ABM Devices

ABM symbols are in `abm.olb`. The library can actually be thought of as two libraries. The first section contains the control system devices and the second section contains the symbols for the more traditional SPICE controlled sources.

Types of ABM Devices

The two types of ABM devices are control system devices and PSpice A/D equivalent devices.

Control system devices are defined with the reference voltage preset to ground so that each controlling input and output is represented by a single pin in the symbol.

PSpice A/D equivalent devices reflect the structure of the PSpice A/D E and G device types that respond to differential input and have differential output.

Using ABM Devices

Generally, an ABM symbol is placed on the schematic page and then its ***EXPRESSION*** attribute is modified to fit the current need. The Evalue symbol is pictured below. Its ***EXPRESSION*** attribute is ***V(%IN+,%IN-)***. If left as is, the output of the device will be the voltage differential seen across the two input pins IN+ and IN-. Using ABM Device by Reference.

One way to use an ABM device is to connect the controlling nets to its input pins.

Using ABM Devices by Reference

Another way to use the ABM devices is to reference a node voltage, a voltage differential between two nodes, or the current through a device located somewhere else in the design.

Voltage Controlled Resistor Example

The voltage waveform on the left is used as a control signal for an ABM device which varies the current to emulate a resistor varying with the control signal.

Phase Locked Loop Example

Creating a Phase Locked Loop can be done much more quickly with ABM components than with transistor level ones. The simulation time is also much shorter as fewer details are taken in to account.

The graph on the right is showing the error which seems random at the beginning and the end but in the center it is predictable meaning that the output waveform is matching the phase of the input waveform.

Behavioral Assertions

Stop your simulation the second it exceeds some pre-defined threshold, thus eliminating the wasted simulation time that would follow.

In this example, the simulation is stopped immediately after the output voltage exceeds 21V saving us the remaining simulation time that would be used only to find out that the results were not acceptable.

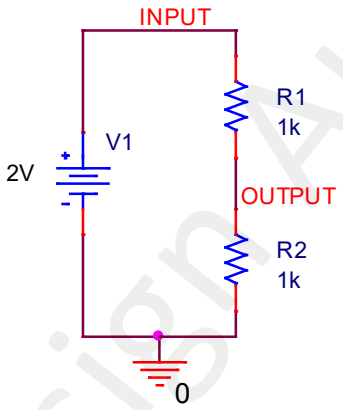
Lesson 14, Lab 1: Voltage Controlled Resistor

Lab Objectives

- Create a circuit with a resistor that is variable based on a voltage source

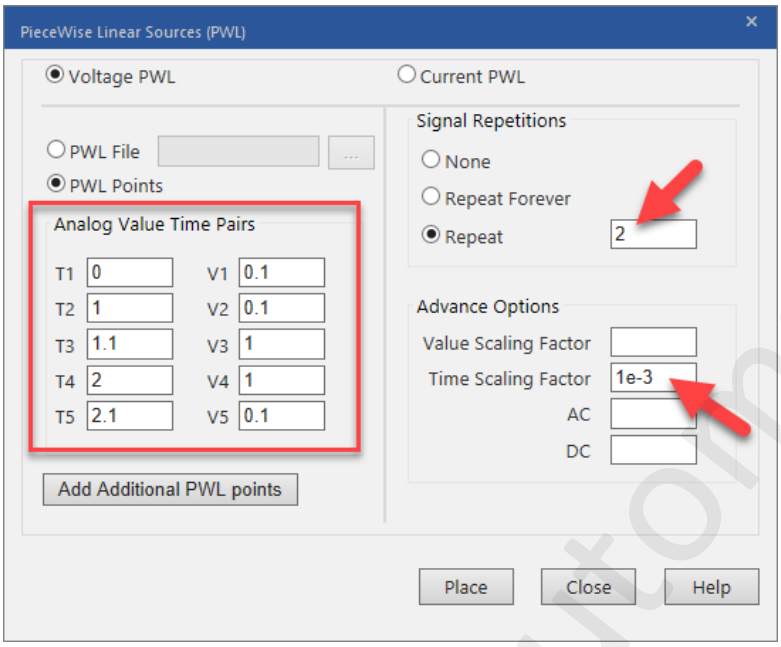
Placing and Defining Parts

1. Create a new PSpice design called `abm_vcr`
2. Place a Resistor and DC Voltage source and name the nets to get a circuit like this:

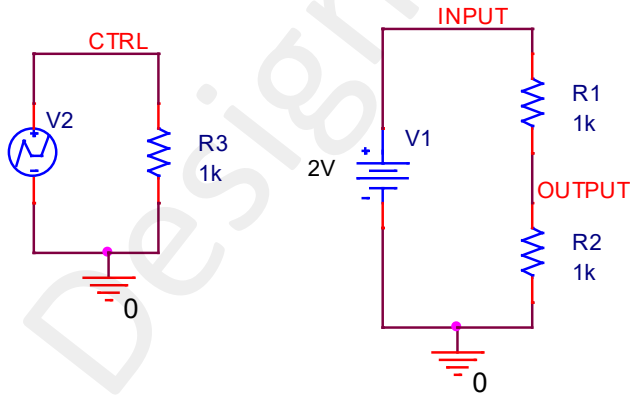


This is the framework for the design that we want to make, we would like to have the resistor R2 be a variable resistor whose value changes based on a controlled voltage source.

3. Go to **Place > PSpice Component > Modeling Application… > Sources > PWL Sources** to bring up the PWL Sources dialog
4. Fill in the PWL points to match the screenshot below, you can also make it repeat twice as shown if you would like to see the effect



5. Hit Place and put the VPWL part on the schematic
6. Put a load across the VPWL part and ground the loop

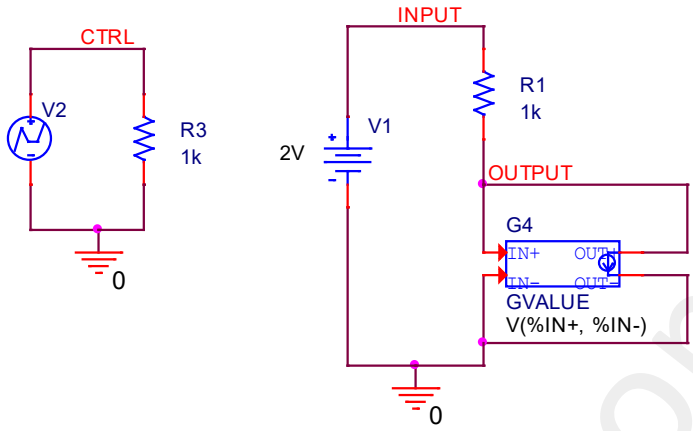


This is the starting point of our circuit, we'll now replace the R2 part with a resistor that looks to the V(CTRL) as its reference.

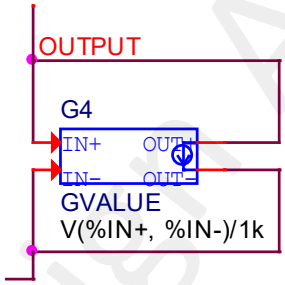
Using a Voltage Controlled Current Source

1. Delete the R2 part from your schematic
2. Using the PSpice Part Search (Place > PSpice Component… > Search…), search for and place the GVALUE part where R2 was

3. Wire it in like this:



4. Double click on the V(%IN+, %IN-) value on the schematic (it's the EXPR property) and edit it to have a '/1k' at the end, like this:

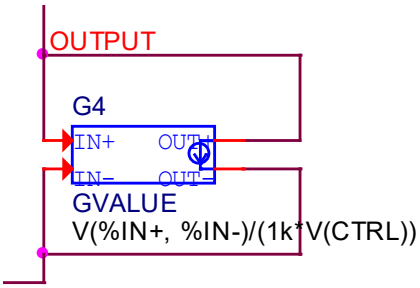


This now means that the current coming out of the device is equal to the voltage difference at the input pins divided by 1k (which is a placeholder for the default resistance). Since I=V/R and we're setting the R (at 1k) and the voltage is across our resistor, the current being generated is going to be the correct current for a resistor of 1k so in a roundabout way, we've made a (somewhat complicated) resistor. We'll see why we did this in the next section.

Tying in the Control Voltage

1. Edit the EXPR property again by double clicking on the displayed value of V(%IN+, %IN-)/1k

2. Modify the denominator of the EXPR property to be (1k*V(CTRL)), like this:



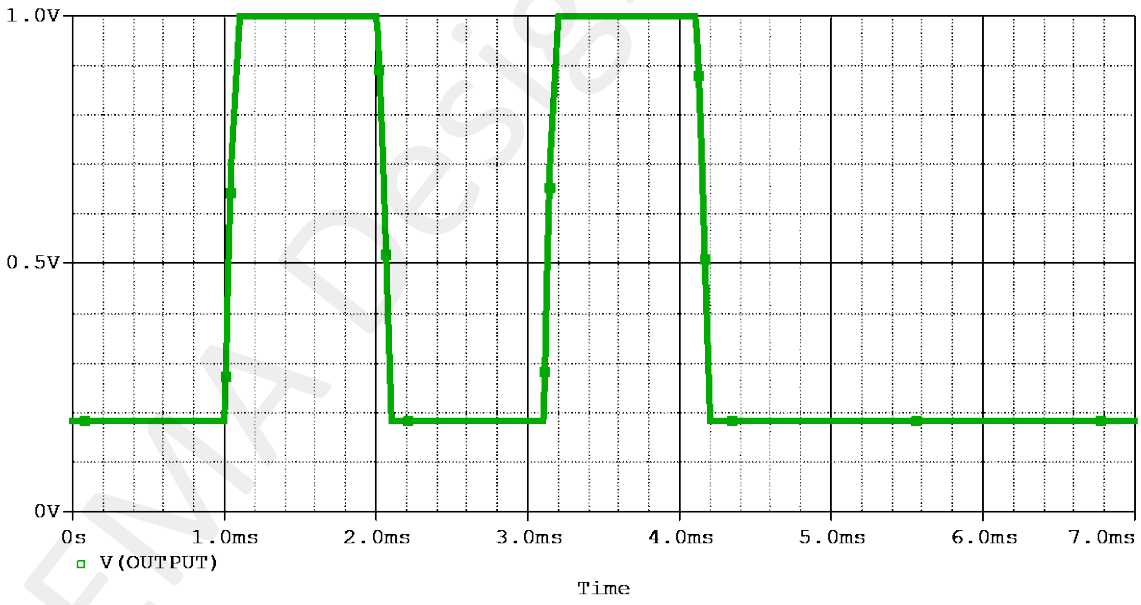
Multiplying the control voltage onto the default resistance value of 1k should effectively swing it between 100 ohms and 1k.

Simulate to Verify

1. Make a new simulation profile, name it **tran** like and perform a transient simulation with a run to time of 7ms

2. Run the simulation (***PSpice > Run***, icon or F11 key)

3. Verify your results:



Note that when the control voltage is high (1V), the resistance is 1k and the output is 1V (half of the 2V input). When the control voltage is low (0.1V), the resistance is 100 ohms and the output is 0.1818V (or 100ohm/(100ohm+1000ohm)*2V) so it is working as we expect it to.

Optional Questions

1. When the VPWL part was made, why was it important that the voltage value didn't go all the way down to 0?

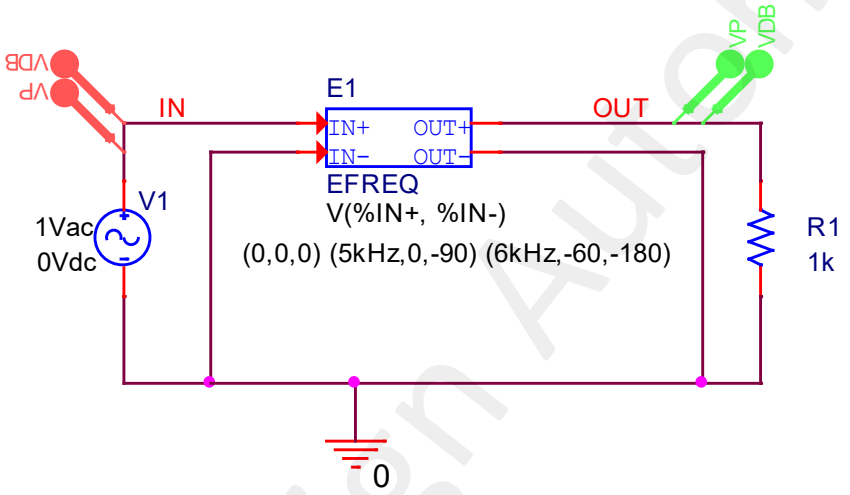
2. What types of control voltage sources would you be limited to with this setup?

Lesson 14, Lab 2: Frequency Table

Lab Objectives

- Create a lowpass filter using an Efreq part.

Placing and Defining Parts



This circuit is a simple lowpass filter implemented using 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 lowpass filter with a response of 1 (0 dB) for frequencies below 5kHz and a response of 1mV (-60dB) above 6kHz. The phase lags linearly with frequency.

1. Create a new project called EFREQ and base it off a blank project
2. Place and connect the symbols needed in the circuit, which are **VAC**, **0**, **EFREQ**, and **R**
3. Set the voltage for ***V1*** to ***1V***.
4. Double-click on the ***EFREQ*** symbol.
5. Select on the ***TABLE*** property.
6. Edit the value for the ***TABLE*** property to read *(frequency, gain (dB), phase)*:

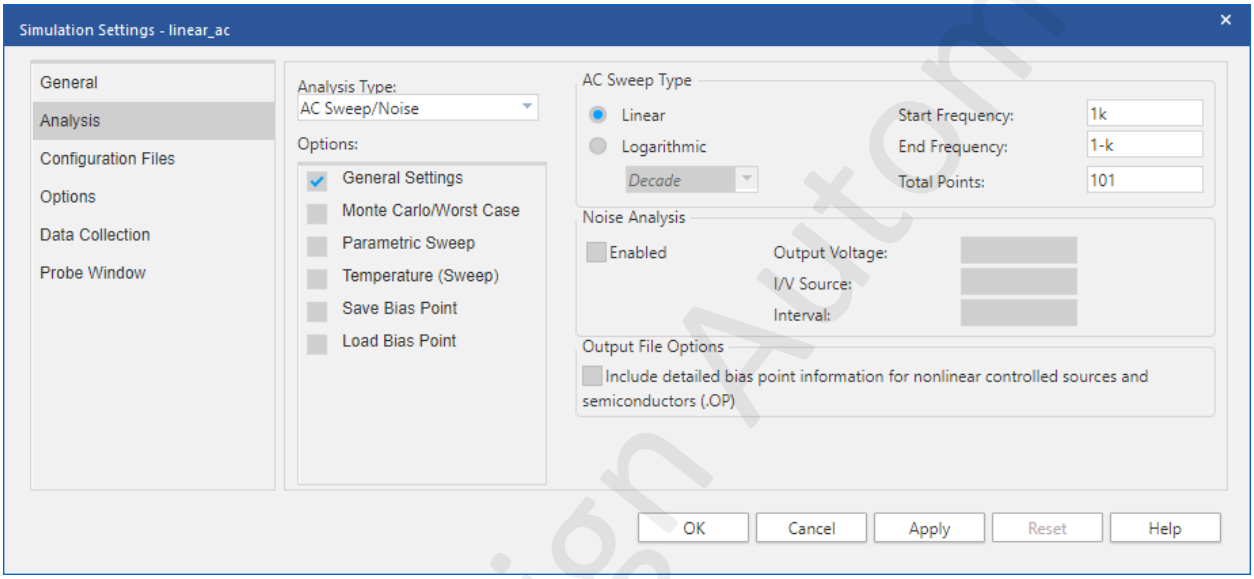
```
(0,0,0) (5kHz,0,-90) (6kHz, -60,-180)
```

7. Click the ***Display*** button.

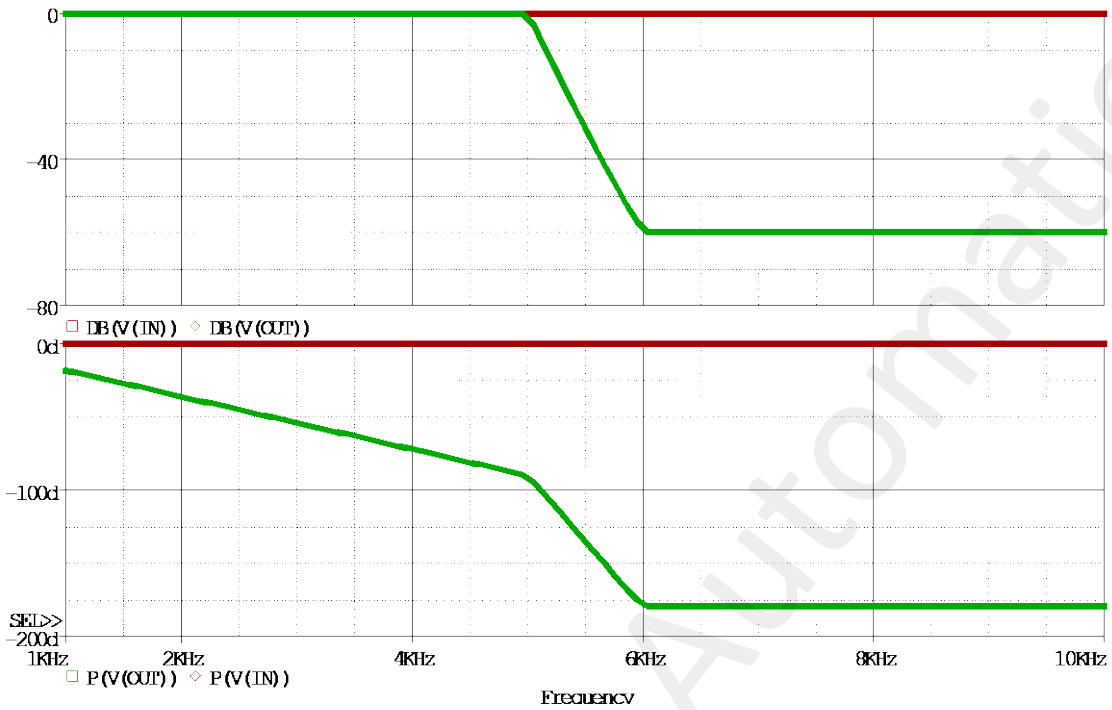
8. Check the ***Value Only*** radio button to display the Value of the TABLE property on the schematic.

Configure Simulation

1. Set up new simulation profile called `linear_ac`
2. Set it as a ***Linear AC simulation*** with ***101*** points, starting at ***1k*** and ending at ***10k***.



3. Click the ***OK*** button.
4. Choose ***PSpice > Marker > Advanced***.
5. Select ***dB Magnitude of Voltage*** from the list.
6. Place the marker on node ***IN*** and ***OUT***.
7. Repeat on the ***IN*** and ***OUT*** nodes for the ***Phase of Voltage*** Advanced Marker
8. Run the simulation (***PSpice > Run***, icon or F11 key)
9. Create a ***New Plot*** and ***move*** the ***DB*** measurements up to it (Cut and paste works)
10. Double Click on the ***X axis*** to change the Data Range to between ***1kHz and 10kHz***



You can see that the response is as expected. Below 5k the response is 0dB and above 6k the response is -60db. The phase also behaves as we described it.

Lesson 14, Lab 3: Voltage-Controlled Oscillator

Lab Objectives

- Create a voltage-controlled oscillator using Evalue and Gvalue parts.

Background Math

This circuit is a Voltage-controlled oscillator. The top portion of the circuit is a simple time domain function for a sinusoidal source. It normally has the equation of sin((twopi*fc*time)+phi), but here the constant phi has been replaced with a function of a controlling source.

$$y(t) = \sin(2\pi f_c t + \phi(t))$$

The instantaneous frequency is given by the time derivative of total phase:

$$2\pi f_{inst} = 2\pi f_c + \phi'(t)$$

The relationship between ϕ and the frequency deviation:

$$f_d = f_{inst} - f$$

is

$$\phi(t) = \int 2\pi f_d(t) dt$$

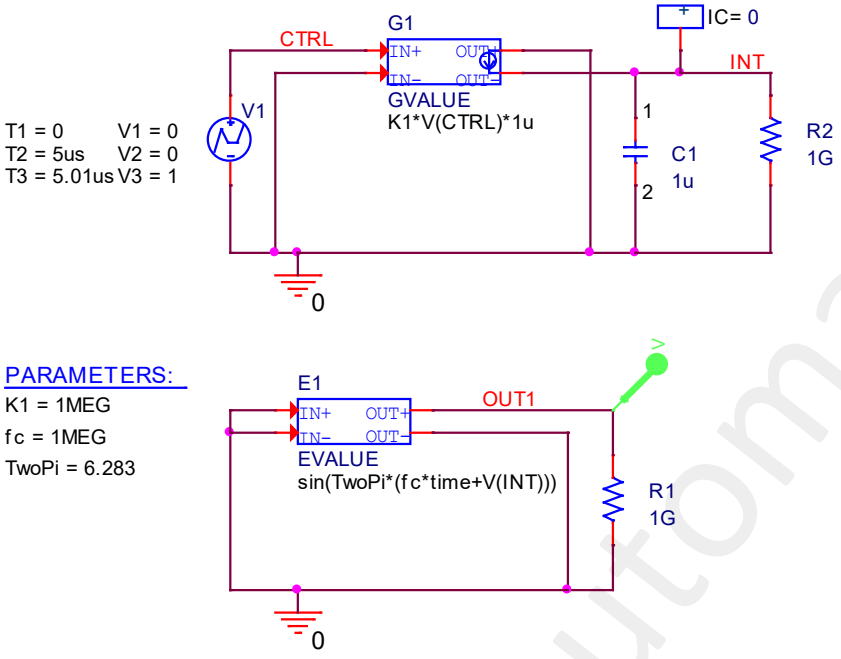
For a linear VCO we want fd to be proportional to the controlling voltage: v_{ctrl} so

$$\phi(t) = 2\pi k_1 \int v_{ctrl}(t) dt$$

where k_1 is in Hertz/volt.

Creating a new design

1. Create a new project called `VCO.opj`.

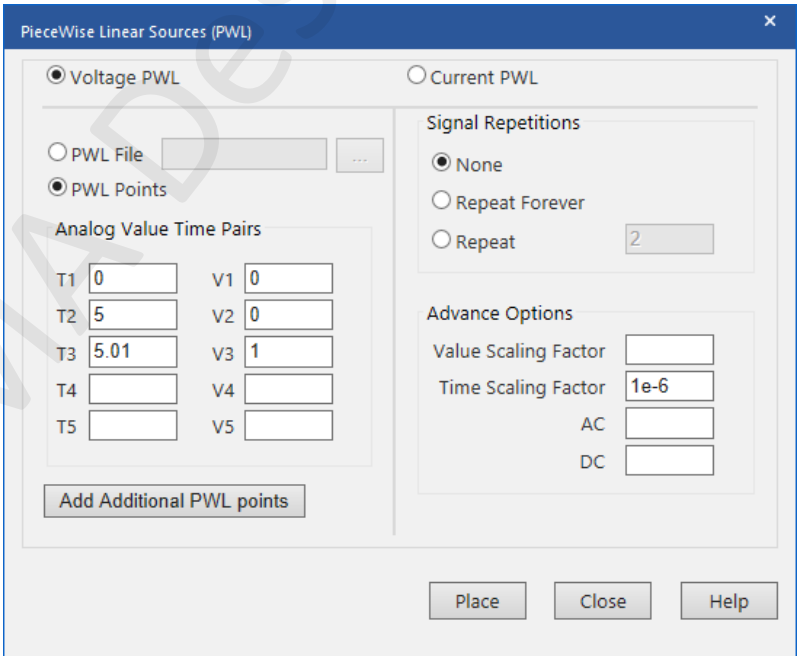


2. Enter the circuit shown above. Use the **Evalue**, **Gvalue**, **IC1**, **R**, **C**, **VPWL** (Use the Modeling App), and **Param** parts.

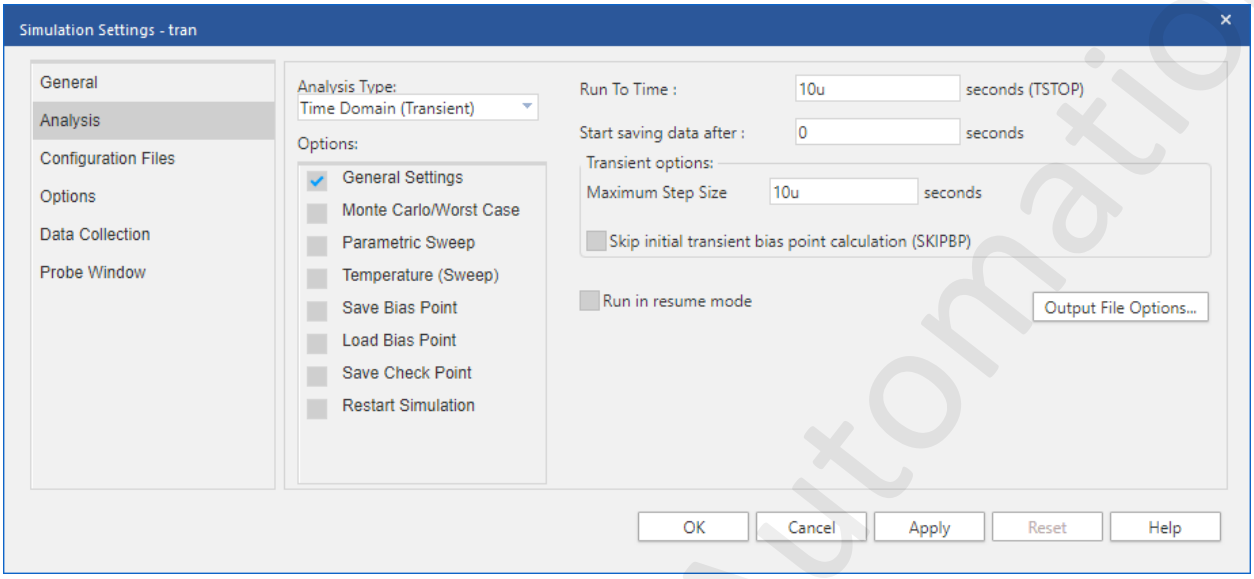
3. The value of the EVALUE part is `sin(TwoPi*(fc*time+V(INT)))`

4. The value of the GVALUE part is `K1*V(CTRL)*1u`

5. ***Create the VPWL*** so it has these time points ***(0s,0V) (5us,0V) (5.01us,1V)***

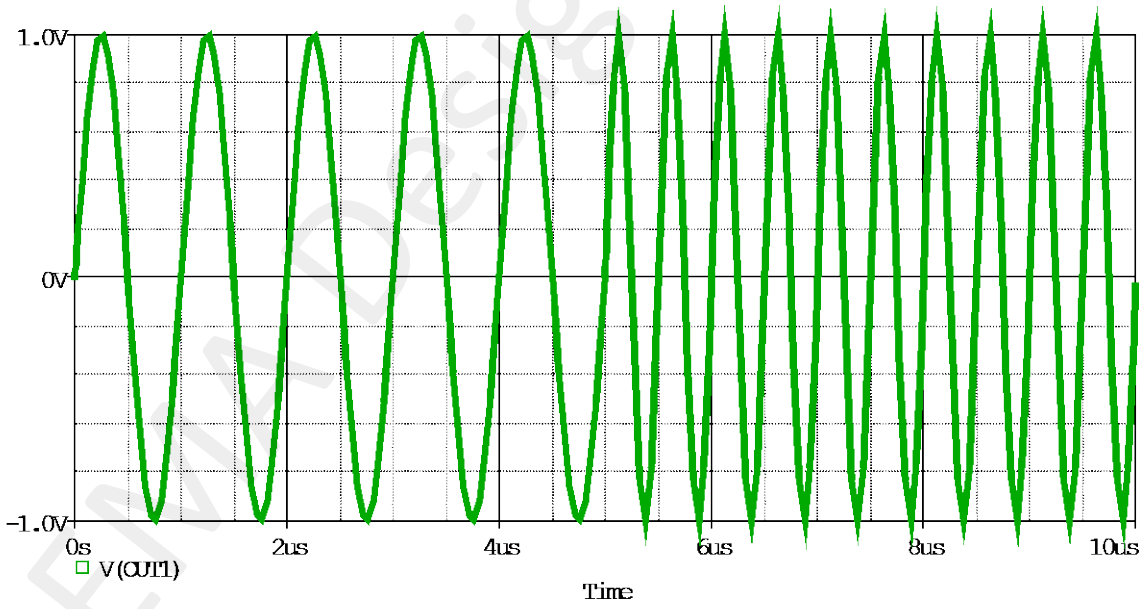


6. Create a simulation profile called **`tran`** defined as a ***transient analysis*** with the ***final time of 10us*** and a ***maximum step of 50ns***



7. **Run** the simulation and examine the results

You should see that the output of the VCO is 1 MHz for the first 5us and 2 MHz for the next 5us.

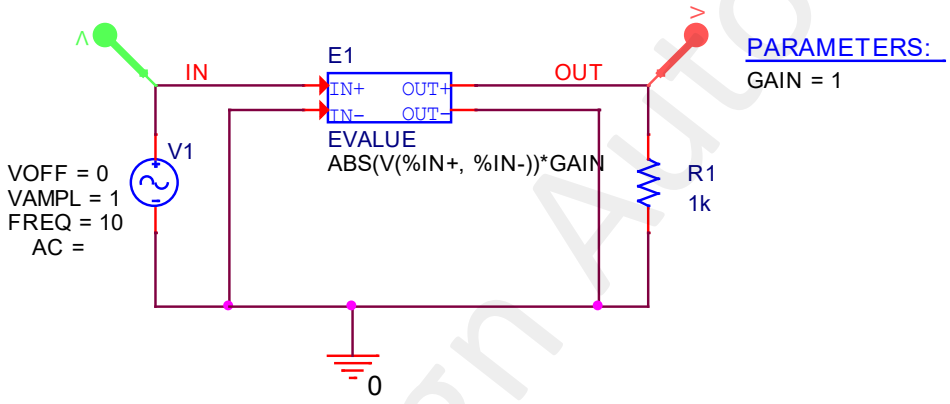


Lesson 14, Lab 4: Optional: ABM Evalue with Wired Inputs

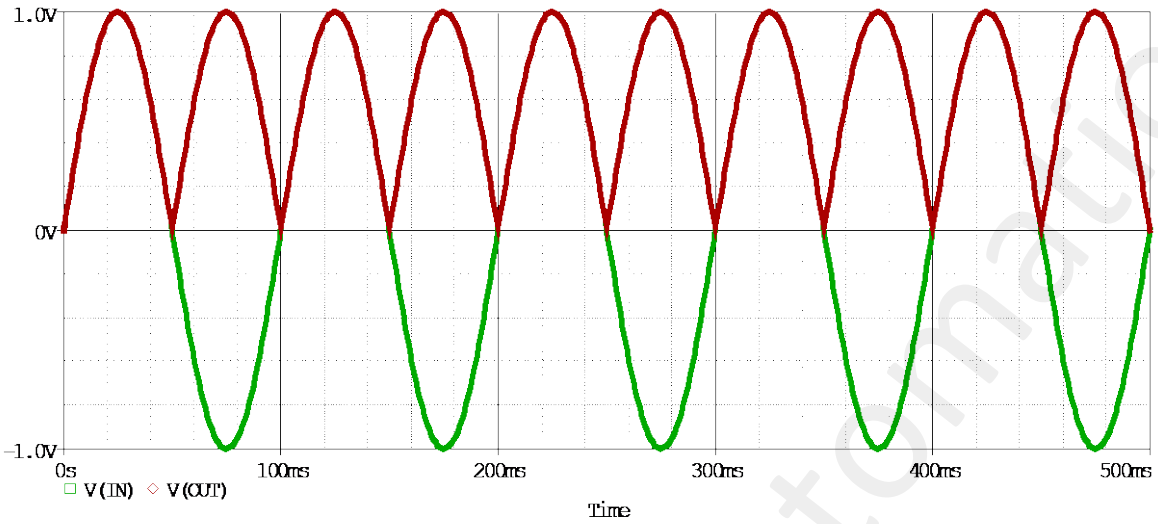
Lab Objectives

- Create a rectifier using an Evalue part.

Define Symbol Properties



1. Create a new project called **EVALUE** and base it off a blank project
2. Place and connect the symbols for the circuit above: **VSIN**, **0**, **Evalue**, **R**, and **PARAM**
3. On the **PARAM** part, create a new property called **GAIN** and give it a value of **1**, display it on the schematic
4. Edit the sine wave to have **0 offset, 1V amplitude, and 10Hz frequency**, the AC property can remain blank
5. Double-click the default expression for the Evalue device **(V(%IN+,%IN-)**) and change it to **ABS(V(%IN+,%IN-))*GAIN**.
6. Set up a ***transient simulation*** with ***0.5s Final Time*** and ***1ms Maximum Step Size***.
7. Run the simulation and observe the output.



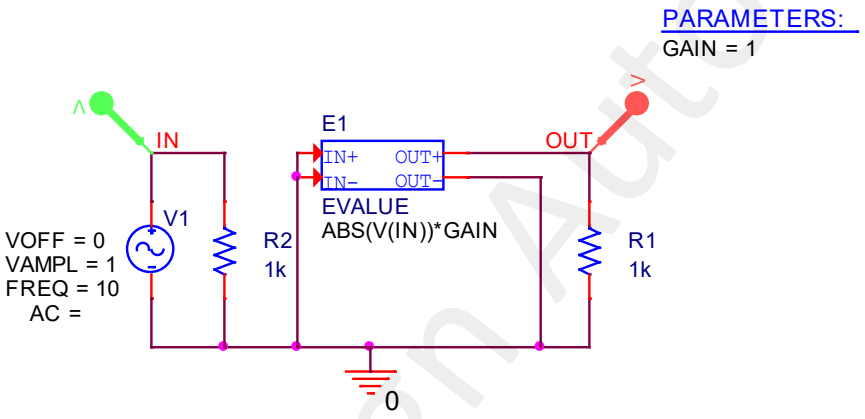
8. Change the gain and re-run the simulation. If you want to make the gain an expression you must enclose it in curly braces. For example, GAIN ={2*5}.

Lesson 14, Lab 5: Optional: ABM Evalue by Reference

Lab Objectives

- Edit the circuit so that the ABM device is connected by reference.

Editing the Design



It is possible to reference the voltage at a node, a voltage differential between two nodes, and the current through a device in the expression of an ABM device. In this example you will see that even though there are no inputs connected to the ABM symbol, it is still functional. This is extremely useful if the output of the device is dependent upon more than just a differential voltage. This allows the Evalue and Gvalue to become powerful, flexible, and have a more general application.

1. Continue working on the previous design
2. Add resistor **R2** and connect as shown above
3. Disconnect pin IN+ of Evalue from the source by selecting and deleting the wire
4. Short the IN+ and IN- pin of E1 to ground
5. Assign a net alias of ***IN*** to the node between V1+ and R2
6. Edit the ***EXPRESSION*** property of ***E1*** to reference the voltage at node VIN instead of the voltage seen across the two input pins of E1. The value of the expression will be ***ABS(V(IN))*GAIN***.

7. Run the simulation (***PSpice > Run***,  icon or F11 key) and view the results.

The results are the same for both circuits although the referenced voltage is not wired directly to the controlled source.

Additional Exercise

- Try running a parametric sweep to sweep the value of the GAIN parameter from 1 to 10

Appendix

Topics covered in Appendix I:

Adding New Parts

Creating Symbols for Simulation Models

Subcicuit from the Buffer Circuit

Creating a Part with the Wizard

Topics covered in Appendix II:

Digital and Mixed Circuit Analysis

Placing Digital Parts

Wiring Digital Parts

Digital Power and Ground

Viewing Digital Results

Defining a Stimulus

Configuring Stimulus files

Mixed Simulation

Appendix II Lab 1: Creating Digital Stimulus

Appendix II Lab 2: Creating an ADC (mixed signal) design

Appendix II Lab 3: Adding a Bus to our ADC Design

Appendix II Lab 4: Adding a DAC to our ADC Design

Appendix II Lab 5: Modifying the Digital Timing Model

Appendix II Lab 6: Magnetics Part Editor

Appendix I

Adding New Parts

Types of Simulation Models

Model or subcircuit information necessary to run a simulation is stored using either one of two possible methods. It is either stored in a simulation library or as a symbol reference to an existing schematic.

In the first method, circuit data is stored in a simulation library which is formatted as ASCII text within the library file. (This file has an `.lib` extension.) A symbol references the model or subcircuit definition through the symbol ***Implementation*** property.

In the second method a symbol references an existing schematic which is known as a subschematic. This is usually referred to as a hierarchical symbol. When the design is simulated the netlister descends the hierarchy and includes components from the subschematics.

There are advantages and disadvantages to both methods. The advantage of storing the information as a subcircuit definition is space. A collection of subcircuit definitions takes up considerably less hard disk space than a collection of hierarchical schematics. The disadvantage is that, without a schematic to look at, editing a large subcircuit may be difficult.

Editing underlying definitions of hierarchical symbols is very easy and advantageous since the schematic is much easier to edit than the netlist. However, greater storage space requirements as well as additional library maintenance are necessary because a collection of separate files with attached absolute paths are involved.

Creating Subcircuits from Schematics

When designs used for a subcircuit definition are initially simulated alone, sources on the design exist that are not part of the subcircuit definition. For example, opamps do not include their own power supplies. Likewise, a subcircuit definition for an opamp should not include a power supply.

To resolve such an issue, first remove sources that are external to the device and replace them with an interface or global port. Also remove loads from the design since they will be replaced by the rest of the top level circuit.

Hierarchical ports should be connected to nodes intended for a wired connection. For instance, in the buffer circuit shown in the above graphic, replace V1 with a hierarchical port. Then connect a hierarchical connector to the output of U1, and pins V+ and V- on the opamp.

Creating the Subcircuit

The netlister generates a file that has the same name as the design with a .net extension. To use this, you will need to either rename the file to have a ".lib" extension, or cut and paste the contents of the file into an existing model library. The name of the resulting subcircuit is derived from the name of the root schematic folder. You may use any text editor or word processor to do this. The only restriction is that the file must be saved as text.

The normal procedure is to configure the subcircuit in a file of its own until it has been tested. After the testing is completed, then the subcircuit is usually added to an existing library of custom model definitions.

Differences Between Models and Subcircuits

Despite its name, a model library can contain a mixture of model statements and subcircuit definitions.

Creating Symbols for Simulation Models

There are two ways to create a symbol for an existing simulation model. The first method uses the Part Creation Wizard in the Model Editor. The second method creates a part from scratch in the Part Editor.

Part Creation Wizard

To use the ***Part Creation Wizard*** in the ***Model Editor***, run the ***Model Editor*** and use the ***File > Export to Capture Part Library*** menu commands. A dialog will come up which allows you to browse to find the model library for which you want to create parts. By default, the wizard creates a part library with the same base name as the model file upon which it is based. The ***Part Creation Wizard*** is most useful when creating a symbol for an existing simulation model or subcircuit definition. The Part Wizard creates all necessary pins as well as generating the PSpiceTemplate, Implementation Type, and Implementation properties.

Creating a Symbol from Scratch

You may create symbols by hand using the Part Editor available in OrCAD Capture. If you choose to do it this way, you are responsible for assigning pin names as well as generating the PSpiceTemplate, Implementation Type, and Implementation properties.

The Part Creation Wizard in the Model Editor creates parts for an entire library of models or subcircuits. If the device is defined as a .model statement, then the resulting symbols look like the standard symbols for those devices. If the device is defined as a .subckt, then the symbols created are rectangles with pins. You can go into the part library later and edit the graphics, as desired.

Creating Symbol Libraries

You can use the ***File > New > Library*** menu command to create a library with the default name `libraryX.olb` where X is an integer. This library may be created from either the OrCAD Capture or Project Manager window. If done from the OrCAD Capture window, the library may be added to either the current project or to a new project. In both cases, the resulting library file is written to the hard drive for availability in other projects.

To rename the `libraryX.olb` file, right-click the library file name and use the ***Save As*** command from the pop-up menu. The library is saved in the desired location and with the desired name.

Creating a New Part in a Library

To add a new part to the library:

8. Highlight the library in the Project Manager.
9. Click your right mouse button to display the pop up menu.
10. Choose New Part to display the New Part Properties dialog box.
11. Define the name of the new part and click OK.

The Part Editor opens. The Part Editor can be used to create or edit existing parts and symbols.

PSpice Template

The ***PSpiceTemplate*** property is the template used to generate the part netlists.

> Spelling of the PSpiceTemplate must be exact. There is no space between words PSpice and Template.

The slide above shows a fragment from the buffer circuit created in a previous lab and the resulting netlist entry.

The first character of the template will always be the PSpice A/D device designator. In this case it is an X since we are dealing with a subcircuit definition for the `opamp`.

The carat "^" symbols instruct the netlister to insert a hierarchical path, if there is one. When the design is flat, an underscore is inserted.

The first at "@" symbol instructs the netlister to search for a part property with the same name as the text following the symbol and then insert that property's value, if it has one. In this case the netlister will look for a property called ***REFDES*** and inserts U5 into the netlist.

Next come pin definitions. The percent sign instructs the netlister to search for a pin with the name that follows the percent sign and write the netname connected to that pin into the netlist. In the example above this results in the nets 0, IN, VDD, VSS, and OUT being written to the netlist.

```
The second @ command instructs the netlister to search for a
property called Model on the part. Here's where there is a
little hand waving going on behind the scenes. Most of the
PSpice A/D symbols currently in the library were converted
from a previous format where the property Model was used in
place of Implementation. The netlister actually looks for the
Implementation property and uses its value. The template
functions perfectly well if you were to use @implementation in
place of @model.
```

There are additional special characters that may be used in the template.

12. ?variable |…| writes everything between the pipe symbols to the netlist if the variable has a defined value.

13. ?variable |…||…| also writes everything between the pipe symbols to the netlist if it is an "if, then, else" construct. If the variable has a defined value, then everything between the first delimiters is written to the netlist. If it has no defined value, then it writes everything between the second set.

14. ~variable |...| means if the variable does not have a defined value then the netlister write**s** everything between the delimiters to the netlist.

15. &variable means that if the variable does not have a value then the remainder of the template is ignored.

Commands may be nested one within another. For a complete listing of the special characters and examples of their use, please see the **PSpice A/D Users Guide**.

Configuring Symbol and Model Libraries

Once you have new model and part libraries you must configure them so that OrCAD Capture and PSpice A/D can find them. Keep in mind that there are two libraries that must be configured to complete the task and that each one has its own method of configuration.

New part libraries (`.olb`) are configured in the ***Place > Part*** dialog by clicking the ***Add Library*** button.

New model libraries (`.lib`) are configured in the simulation profile under the ***Configuration Files*** tab.

Be sure to store custom libraries separately from the library folders included with the Cadence software application. This helps avoid situations where an old software installation is deleted to make room for a new installation and custom libraries are lost from the installation directory.

Appx I. Lab 1 Subcircuit from the Buffer Circuit

Lab Objectives

- Convert a schematic to a subcircuit

Create Buffer2 Project

1. Create a new project called `Buffer2.opj` and base it on `Buffer.opj` from Lab 9
2. Delete V1.
3. Remove the entire piece of circuitry that includes V2 and V3.
4. Delete the load resistor and capacitor at the output of the LF411.

5. Delete the Parameters part

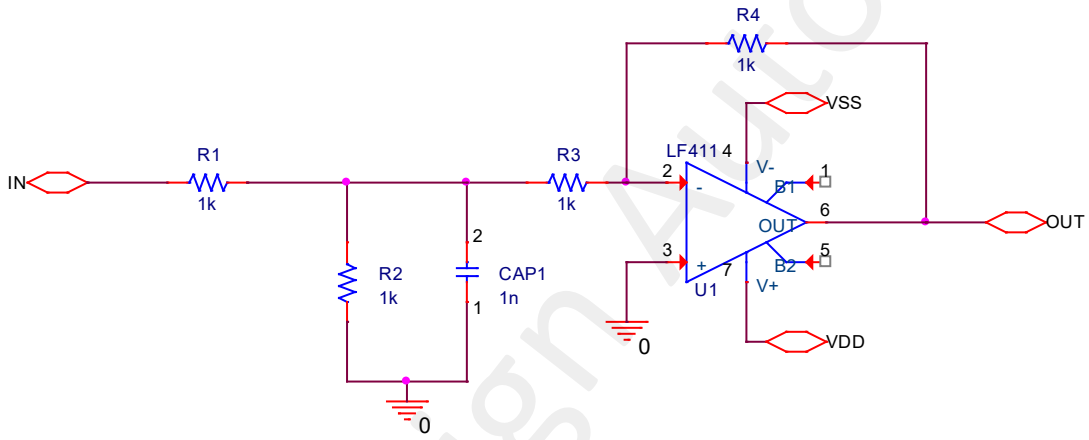
6. Add hierarchical ports (from Place > Hierarchical Port) **portboth-L** on the V+ and V- pins of the LF411.

portboth-L is in the `CAPSYM` library located at C:\Apps\SPB_17.4\tools\Capture\Library

7. Add a hierarchical port **portboth-L** to the output node of the LF411. Add a hierarchical port **portboth-R** to the input left of R1

8. Double-click on each hierarchical port and label it as shown in the picture.

9. Remove redundant net names (IN, VSS, VDD)

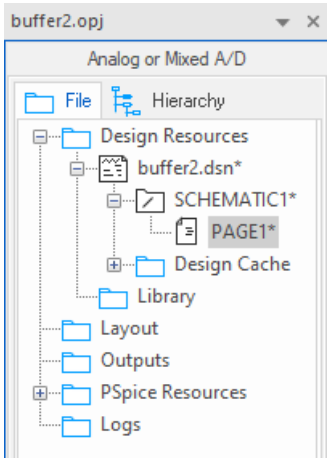


10. Save the design.

Rename Schematic Folder to Create Subcircuit

1. Click on the ***Project Manager*** toolbar button and select the `buffer2.dsn` file.

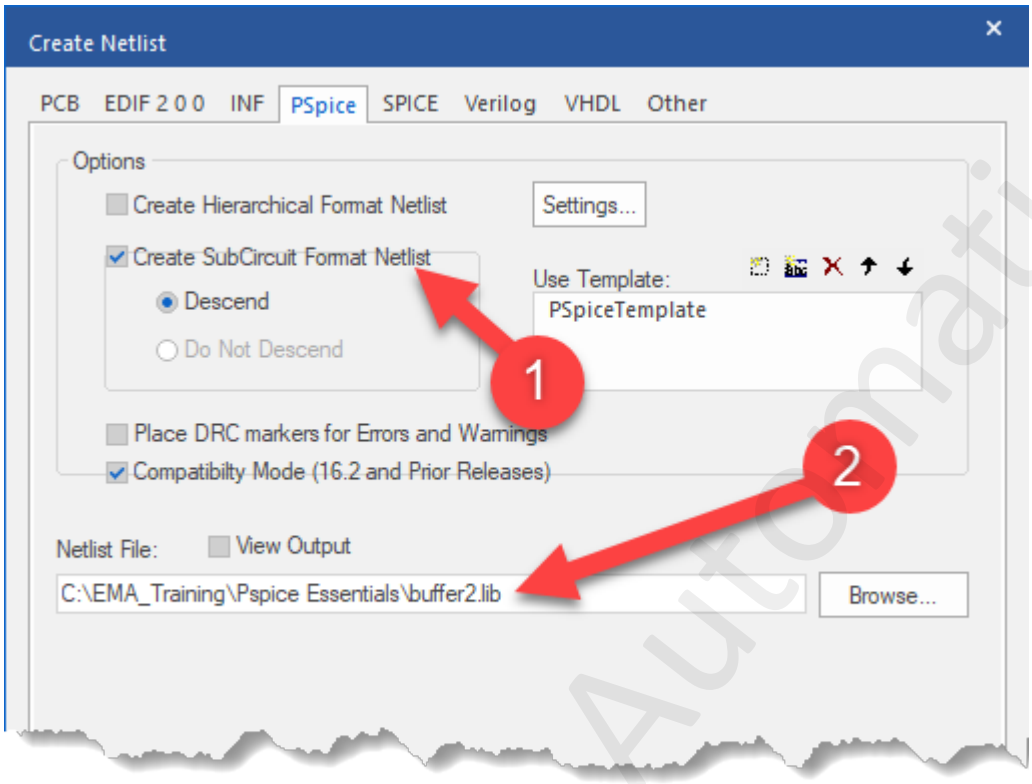
2. Change the name of the schematic folder from ***SCHEMATIC1*** to `BUFFER`.



The name of the subcircuit is created from the schematic name. If you do not change it, the resulting subcircuit will be called schematic1.

Create Netlist

1. **Select buffer2.dsn** in the Project Manager
2. Choose ***Tools > Create Netlist***.
3. Click on the ***PSpice*** tab.
4. Check the ***Create Subcircuit Format Netlist*** check box.
5. Change the path of the library file to your training directory so that you can find it later.



6. Click ***OK***.

Subcircuit Listing

1. View the resulting subcircuit by clicking on the ***Outputs*** folder located within the OrCAD Capture Project Manager and selecting the library file.

Note that your listing may vary slightly from the one shown below by pin order and node names.

```
* source BUFFER2
.SUBCKT BUFFER IN OUT VDD VSS
R_R1          IN N00129  1k TC=0,0
R_R2          0 N00129  1k TC=0,0
C_CAP1          0 N00129  1n  TC=0,0
R_R3          N00129 N00877  1k TC=0,0
R_R4          N00877 OUT  1k TC=0,0
X_U1          0 N00877 VDD VSS OUT LF411
.ENDS
```

2. Close the **buffer2.lib** file

Appx I. Lab 2 Creating a Part with the Wizard

Lab Objectives

- Use the Symbol Creation Wizard
- Create a part for the subcircuit

Create Library Part

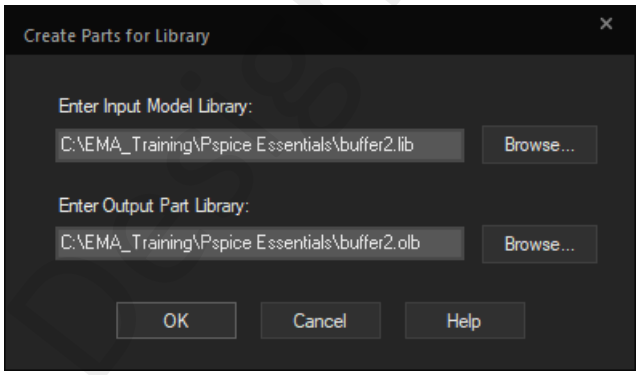
1. Run the Model Editor by ***Start > All Programs > Cadence PCB Utilities 17.4-2019 > Model Editor***.

The Model Editor opens.

2. Choose ***File > Export to Capture Part Library***

3. Select Browse and navigate to the `buffer.lib` file.

This file should be saved in a folder located at C:\EMA_Training\PSpiceEssentials\Designs



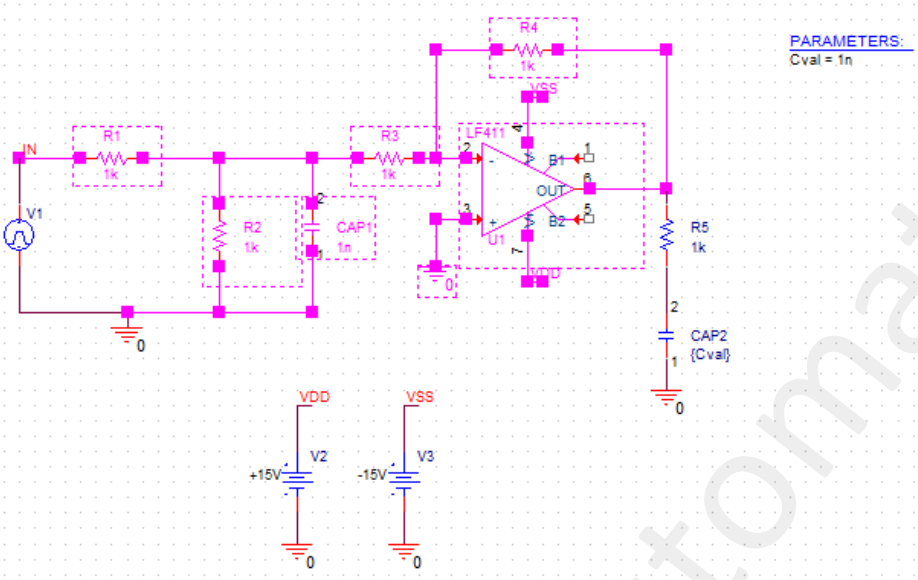
4. Click ***OK*** to create the part. Hit ***OK*** to close the information window. You may close the PSpice Model Editor.

View the Part

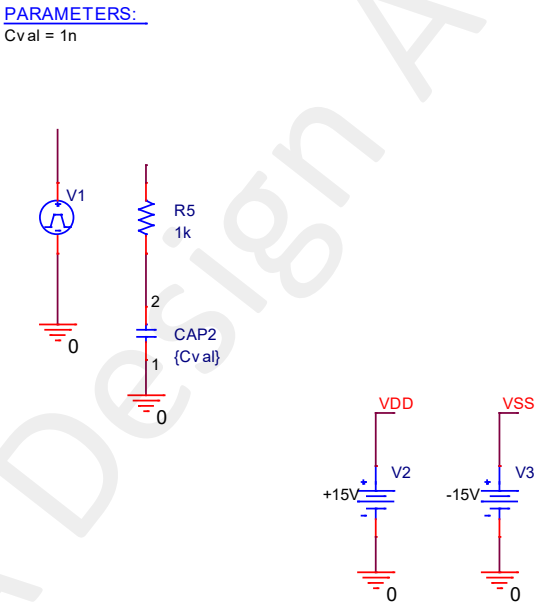
1. Return to OrCAD Capture.

2. Create a new project called `Buffer3.opj` based again off the buffer.olb file from Lesson 9

3. **Delete** all the component except the Voltage Sources, the RC Load and the Param part



4. Move the RC Load to the left



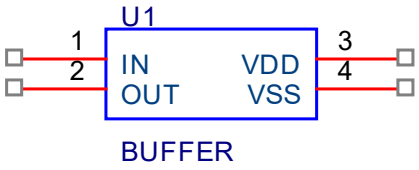
5. Choose ***Place > Part***.

6. Click the ***Add Library*** icon and configure `Buffer2.olb`.

Be sure to use the one from the training directory. There is a buffer.olb file in the regular PSpice library folder, but that is not the one you want to use.

7. Find ***Buffer*** in the list of parts and double-click on it.

8. Place an occurrence of the part.

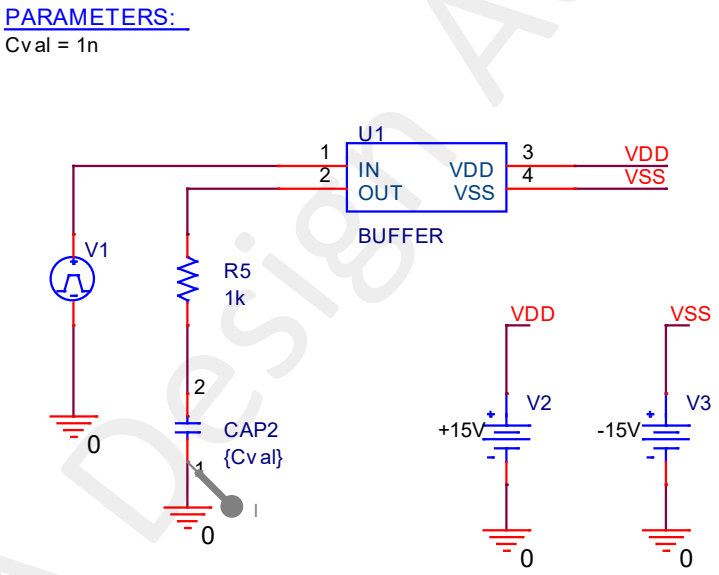


Using the New Symbol

1. Wire up the newly placed buffer part to the source, load and supply voltages

2. Add a current probe to pin 1 of CAP1

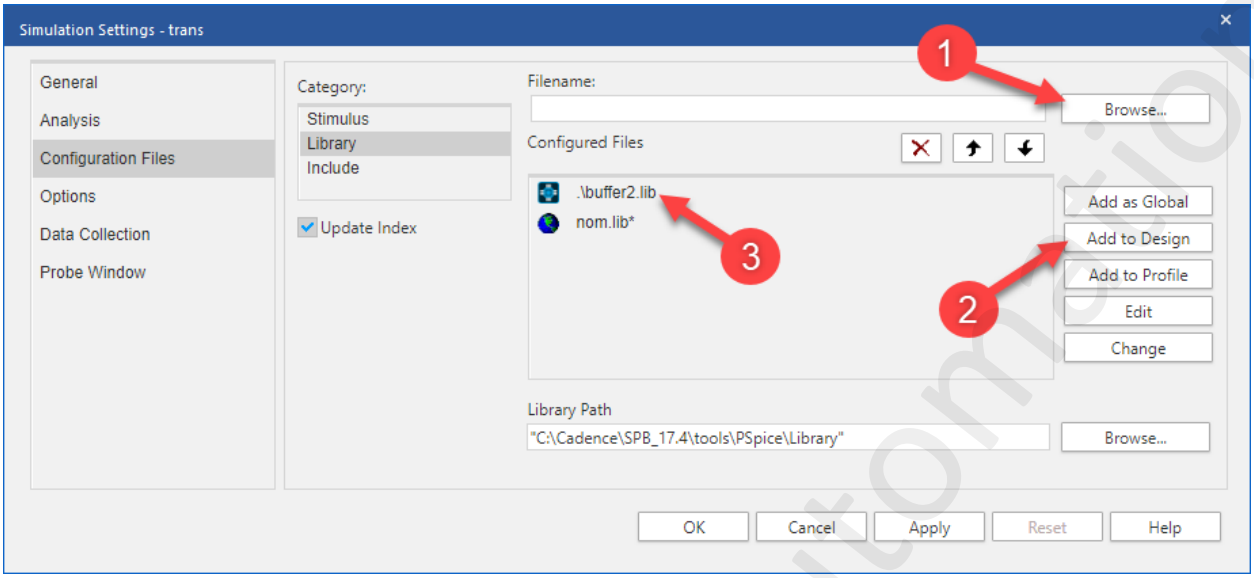
If your pin numbers are not graphically displayed, you can hover your mouse over them and a tech tip will pop up letting you know what the number is. If you want to use a capacitor that displays the pin number graphically, use the C part from the ANALOG_P.olb library.



3. Verify the simulation profile runs for 5us

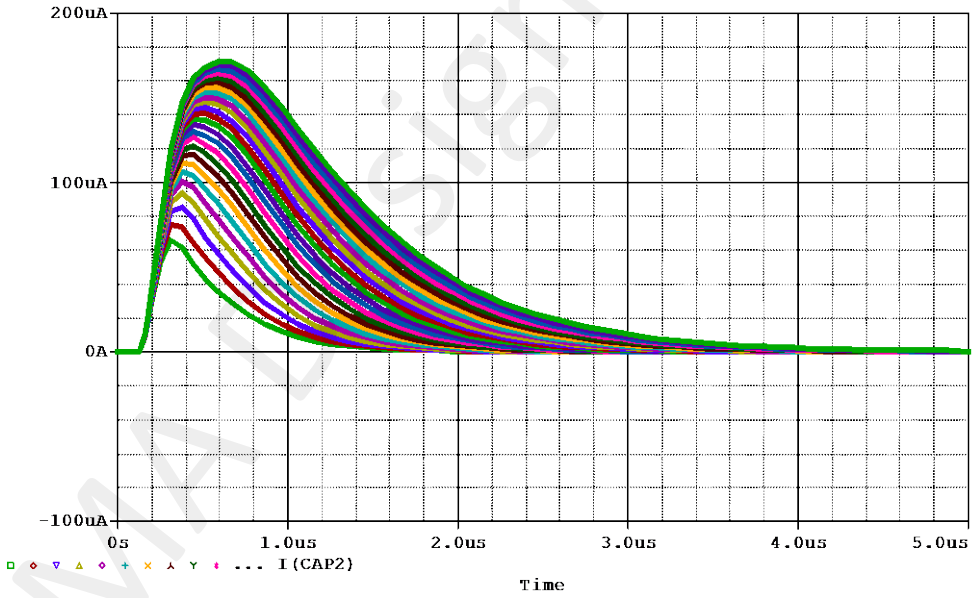
4. In the simulation profile, under the ***Configuration Files*** tab and with ***Library*** selected in the ***Category*** section, add `Buffer2.lib` to the design.

This tells the simulator where to find the PSpice library intelligence for the buffer subcircuit part U1



5. Click ***OK*** to close the simulation profile dialog.

6. Run the analysis and examine the result in the Probe window. It should look like the results seen in Transient chapter, which are reproduced in the following graphic.



With this, we were able to show how to turn a section of circuitry into a reusable circuit block that can be used and placed as an individual component that represents all the underlying technology.

Appx I Lab 3: Creating a Part from Scratch

Lab Objectives

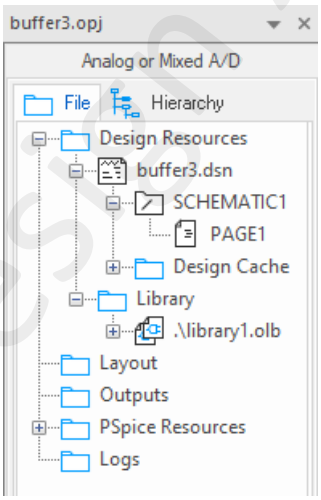
- Create a new four pin part from scratch for our buffer circuit
- Configure libraries

Creating a New Library File

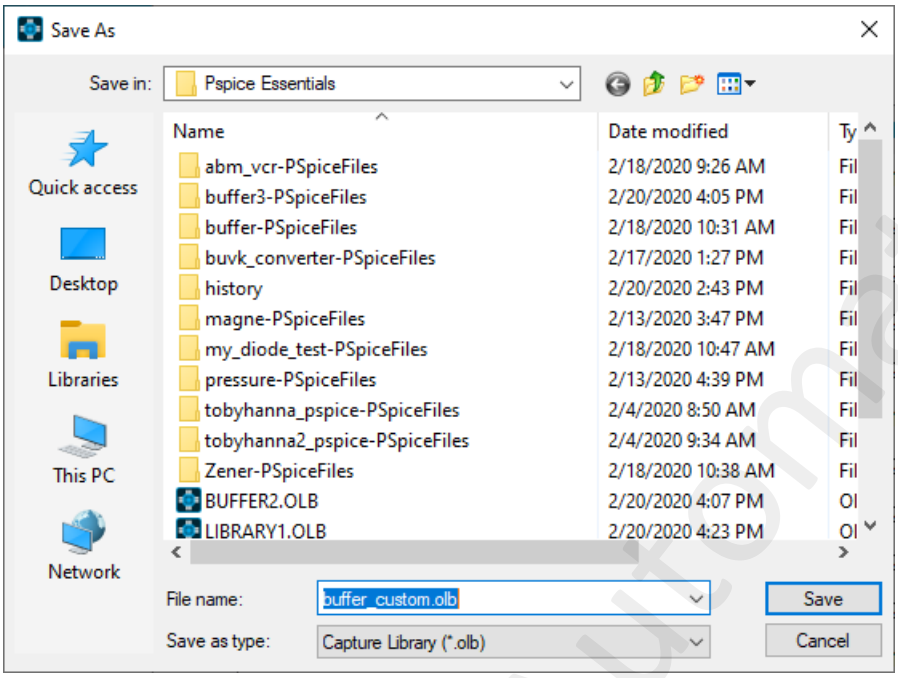
To complete the steps that follow, continue using the `Buffer3` project you created in the previous lesson.

1. From the Project Manager window select the ***File > New > Library*** menu commands to create a new library file.

The new library named .\library1.olb is listed within the Project Manager window as shown in the graphic that follows.

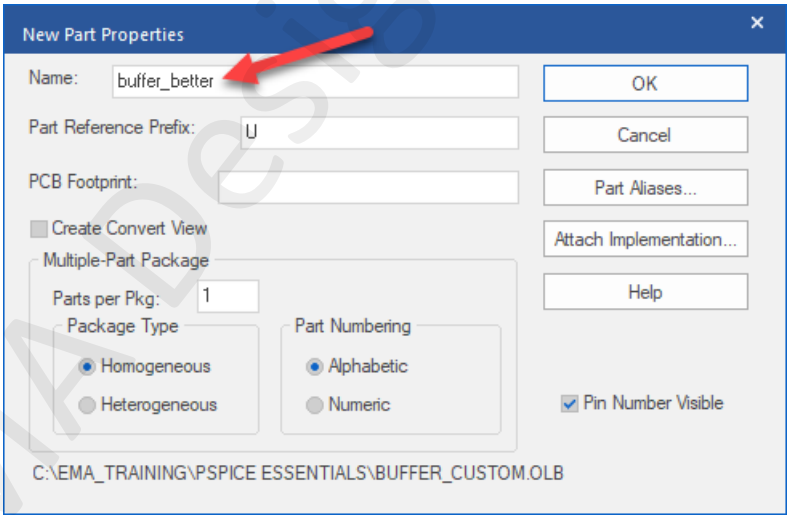


2. Highlight `.\library1.olb` and click your right mouse button.
3. Select ***Save As*** from the pop up menu.
4. Name the library `buffer_custom.olb` and save it in the training directory.

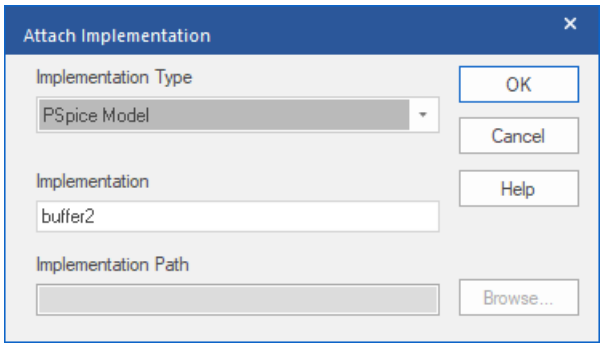


5. Highlight `buffer_custom.olb` and click your right mouse button. Select ***New Part*** from the pop-up menu.

6. Name the new part `buffer_better` and configure the properties as shown in the picture below.



7. Click the ***Attach Implementation*** button and configure the resulting dialog as seen in the following picture.



This tells the symbol that it is going to be associated with an existing PSpice Model named buffer which we made a default symbol for in the last lab.

8. Click ***OK*** to accept the settings and ***OK*** again to open the part editor.

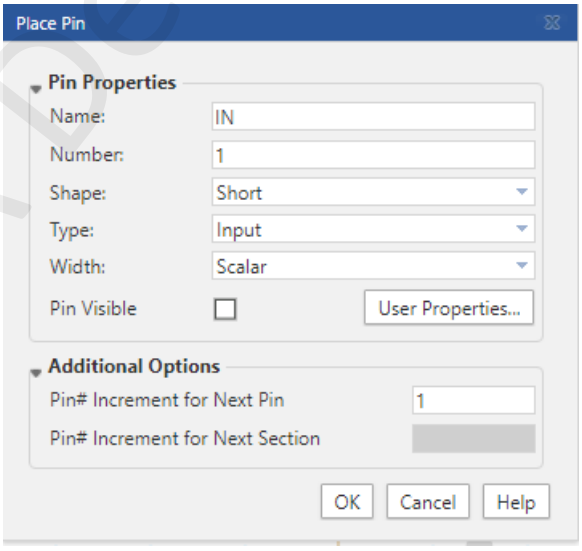
Drawing the Graphics

1. Choose ***Place > Line*** or the toolbar button  to draw the body of the part. The width and length of the part is six grid spaces.

2. Click the ***Place Pin*** toolbar icon. 

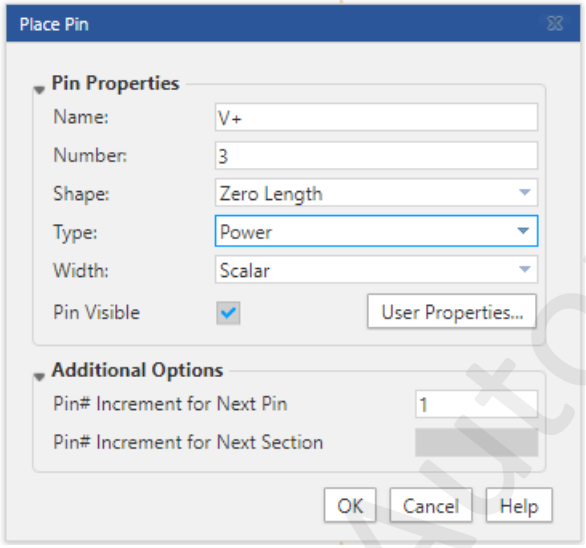
The Place Pin dialog box opens.

3. Define pin number 1 with the name IN and defined as an Input pin, as shown in the graphic that follows.



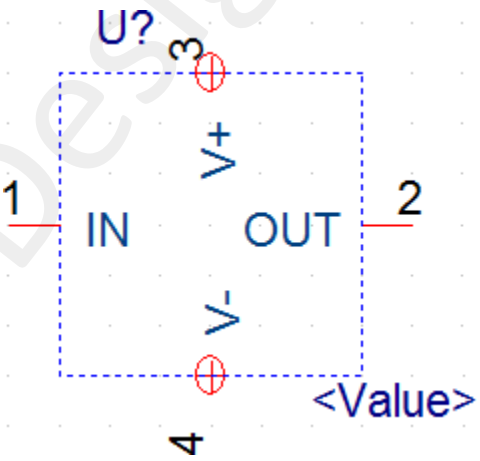
4. Place four pins in total around the body

- Define pin number 1 as a short input pin, named IN (as detailed above)
- Define pin number 2 as a short output pin named OUT
- Define pin number 3 as a zero length power pin named V+ (make Pin Visible)



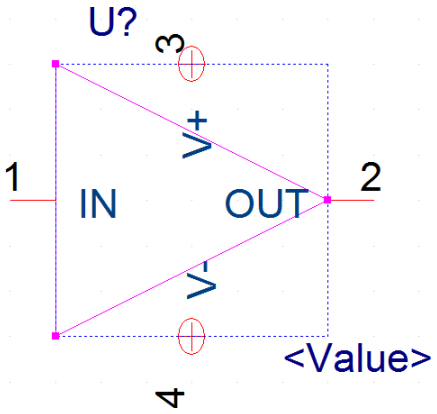
- Define pin number 4 as a zero length power pin named V- (make Pin Visible)

If mistakes are made, you may modify a pin definition by double clicking the pin and changing the values in the Pin Properties dialog box or RMB on a blank section and *Edit pins* to edit all pins on the part.



5. Draw a Polyline ( ) and make a right pointing triangle

Hold down the <SHIFT> key when making a corner to allow the next line to be drawn at any angle (instead of a forced 90-degree angle)



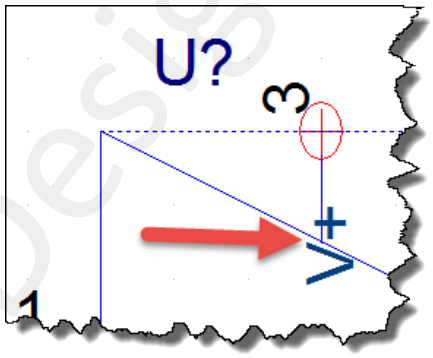
To draw the vertical lines from the power pin to the edge of the triangle, we'll need to carefully turn the snap to grid off.

6. Start the vertical line ( ) from the top boundary edge at the V+ pin location

Since you can't make a nice clean graphical connection at the top edge of the graphic due to the grid snapping, we need to turn the grid off while in this mode.

7. Move the mouse up and toggle off the snap to grid feature.

8. Finish drawing the vertical line by making the connection

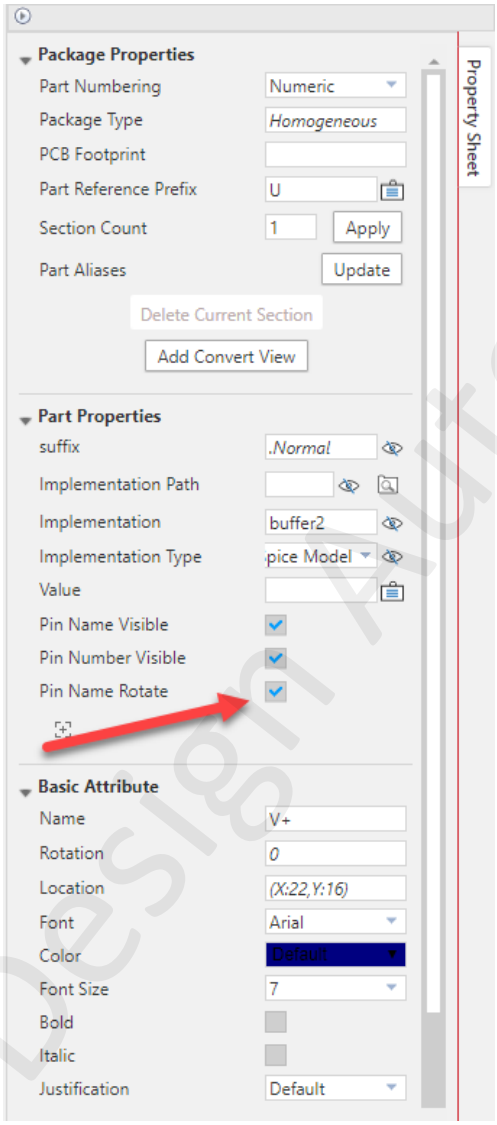


9. Turn the Snap to Grid back on ( )

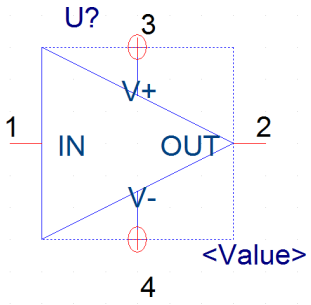
10. Repeat for the bottom vertical line by starting at the bottom pin with the snap to grid **on**, turning the snap to grid **off** mid-way, finishing the connection and turning the snap to grid back **on**

11. Make sure the snap to grid feature is back on

If you want the Voltage Pins and Names to not be rotated, you can turn their rotation off by going to the Property sheet on the right of the screen, then go to the Part Properties section and uncheck the Pin Names check box. You can also press “R” on the keyboard to rotate names.



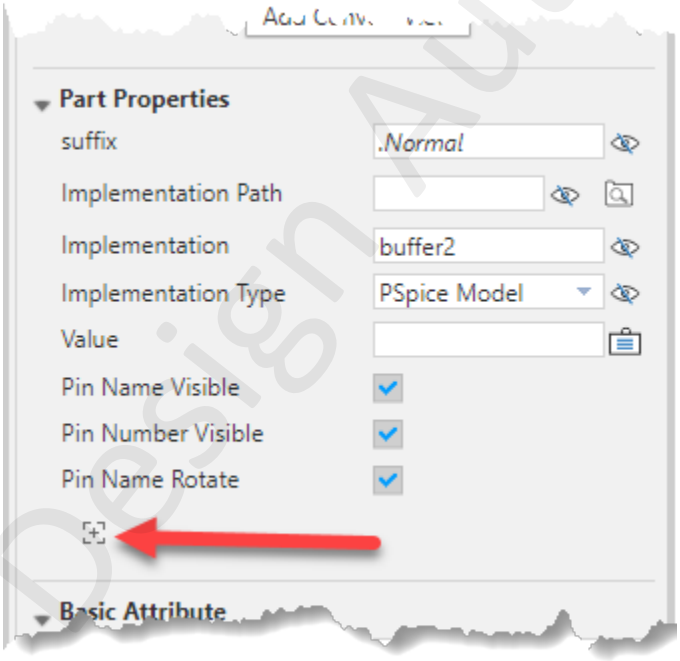
The result:



Editing the Properties

1. On the right of the screen expand ***Property Sheet*** to open the ***Properties*** dialog.

2. Click the ***New*** button to bring up the ***New Property*** dialog.

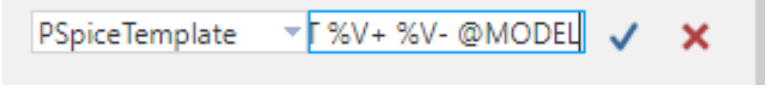


3. Type **PSpiceTemplate** in the ***Name*** field

Note that there is no space between PSpice and Template.

4. Type the following text in the ***Value*** field. Note that there are spaces before and after each of the pin references. Then click the check mark.

```
X^@REFDES %IN %OUT %V+ %V- @MODEL
```



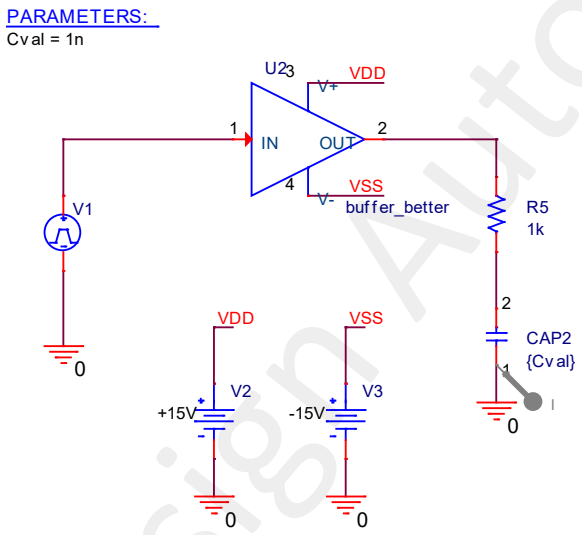
This specifies the order in which the pins will be netlisted – this order needs to correspond with the buffer library file that we generated in the previous lesson exercise.

You can also click **Associate Pspice Model** at the bottom of the Property Sheet.

5. ***Save*** and ***Close*** the library file.

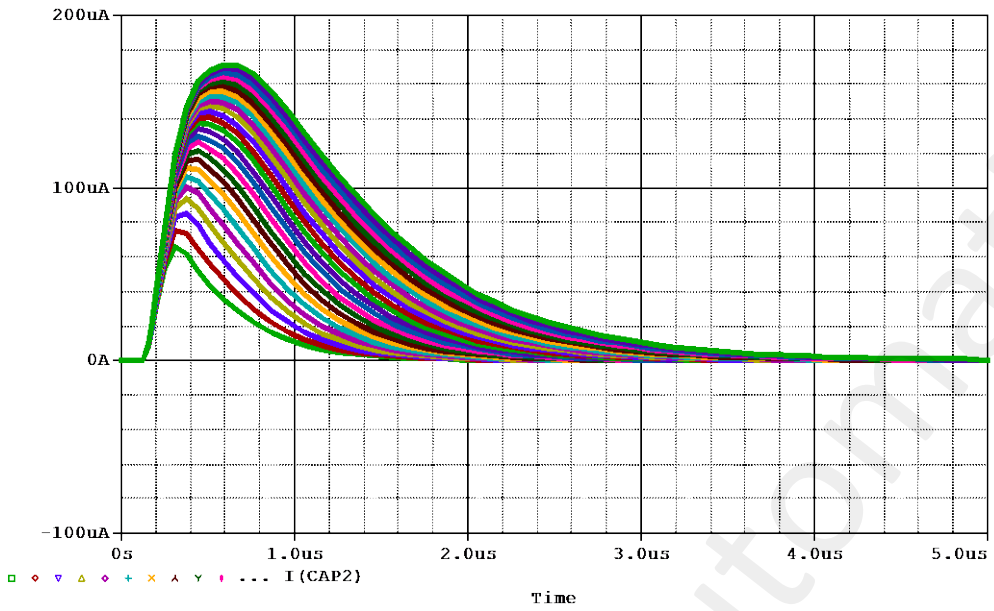
Using the Part in a Design

1. Go back to the open `buffer3.opj` design and replace the buffer block with your `buffer_better`



2. Run the simulation (***PSpice > Run***, icon or F11 key) which is a transient run to 5us

3. Examine the results in the Probe window.



What we did was make a custom graphic for our PSpice subcircuit to make hooking it up and understanding what it does on the schematic a little easier. The results as we can see are unchanged.

Additional Exercise

Take a look at the 3 PSpice properties for the `R` part in the `Analog` library, what are they?

- Implementation = _____
- Implementation Type = _____
- PSpiceTemplate = you don't need to write it all out

How can this be a PSpice part and have no values for Implementation and Implementation Type? The secret is in the PSpiceTemplate, pasted here:

```
R^@REFDES %1 %2 ?TOLERANCE|R^@REFDES| @VALUE
TC=@TC1,@TC2 ?TOLERANCE|\n.model R^@REFDES RES R=1
DEV=@TOLERANCE% TC1=@TC1 TC2=@TC2|
```

Study that PSpiceTemplate for a bit to see if you can understand it at all given what you've learned about how they work

Here's an example netlist containing a resistor that has no tolerance defined and one that does.

Resistor without tolerance property

```
R_R1 IN OUT 1k TC=0,0
```

Resistor with tolerance property

```
R_R1 IN OUT R_R1 1k TC=0,0
.model R_R1 RES R=1 DEV=5% TC1=0 TC2=0
```

You can see how the PSpiceTemplate was sneaky and made a complete underlying model in the netlist based on the existence of a TOLERANCE property (TOLERANCE needs to be defined in a model). This circumvents the need for an Implementation and Implementation Type property which would otherwise be needed to link to an externally configured model.

Appx II Digital and Mixed Circuit Analysis

How Digital Components Are Modeled

PSpice A/D has a complete listing of primitive digital parts from which all other digital parts are built. The primitives include gates (AND, OR, NAND, NOR, etc.), flip-flops (RS, JK, D), buffers, and inverters. Also included are various Logic, Pin Delay, and I/O devices for use with mixed analog/digital simulations.

All digital models are built as sub-circuits. However, they are not built from transistor models. Rather, they are built as event driven models that will only evaluate an output response when there is a qualified input occurrence. This greatly speeds up simulations over systems that model digital devices at the transistor level since each digital event in the simulator occurs much less frequently than an analog time step evaluation. Numerous analog time steps must occur before a digital input node has a qualified threshold crossing to affect a digital event change.

Each digital part is modeled with its core function calling an individual "timing model" and a relevant "I/O model". The timing model is unique to each device type and determines output switching times, propagation delays, and input requirements such as setup and hold. The I/O model is specific to each "family" of digital devices and is based on the underlying technology characteristics to determine interface loading characteristics, drive strength, and power supply specifics.

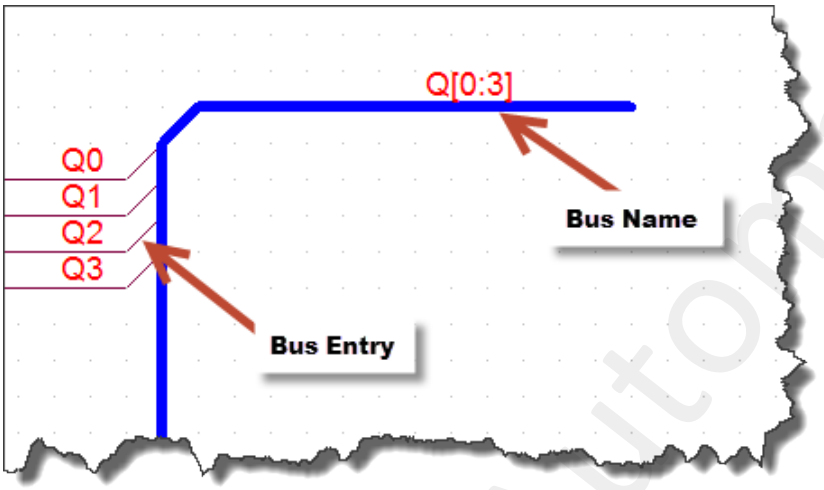
Placing Digital Parts

Placing a digital part on a schematic is done exactly the same way that an analog symbol is placed. Select the ***Place > PSpice Component…*** **command** and either select from the *Digital* sub-menu listing of most common parts, or *Search* for a specific part name or type using the search function. Of course, you will need to configure the appropriate digital model libraries for simulation (more on that later).

Buses and Bus Entries

Buses can be used to carry like named signals across a page, between pages, and even between levels of hierarchy. All buses must be named appropriately, and each wire making up the bus must be named as well. Connectivity within the bus is maintained by the name mechanism, not through the graphical connections.

Unlike electrical wires in PSpice A/D, buses carry no electrical information. System connectivity is passed through buses by the name assigned to the bus. In the below graphic for example, the bus is named *Q[0-3]*. This indicates that the bus name is *Q* and it is four bits wide. The name could also have been given as *Q[0:3]*. Either naming convention is valid.



Note that the bits are enclosed in square brackets, not parentheses, and each wire has a "bus entry" symbol indicating bus membership. The net aliases given to the wires associated with the bus will be *Q0*, *Q1*, *Q2*, and *Q3*. It is not necessary to use square brackets when naming individual wires.

Wiring Digital Parts

Digital symbols have slightly different rules governing their connectivity. With analog parts, every pin must be connected before simulation is enabled. Digital parts, however, may have their output pins unconnected and still simulate successfully. If you want to leave a digital pin unconnected or indicate that you intentionally left a digital pin unconnected, then you may connect it to the ***No Connect*** symbol. Place a ***No Connect*** by using the ***Place > NoConnect*** menu command or by using the *No Connect* toolbar icon ( ).

Digital Power and Ground

All TTL family symbols are automatically connected to global digital power and global digital ground nodes (*$G_DPWR* and *$G_DGND*), through invisible pins on the symbol. ECL devices have invisible power and ground nodes to accommodate VDD, VTT, VCC1, and VCC2. CD4000 devices have invisible pins for VDD and VSS. For any of these devices, you will not need to wire them to power and ground if using standard supply power.

Tying a Digital Pin HI or LO

There are special symbols for tying a digital input HI or LO. Instead of using a ground or voltage source, use the ***$D_HI*** or ***$D_LO*** symbol. These symbols will tie the nodes connected to them to the global digital signals *$D_HI* or *$D_LO*. These symbols are in the standard `source.olb` symbol library.

Access these symbols through the ***Place > Power*** menu command or by using the *Place Power* toolbar icon ().

Pullups and Pulldowns

If you need to 'pull' digital output pins high or low, use any of the ***PULLUP*** or ***PULLDOWN*** symbols (supplied in `dig_misc.olb)` instead of tying the pin through a resistor to power or ground. This avoids having the program insert unnecessary A/D interfaces.

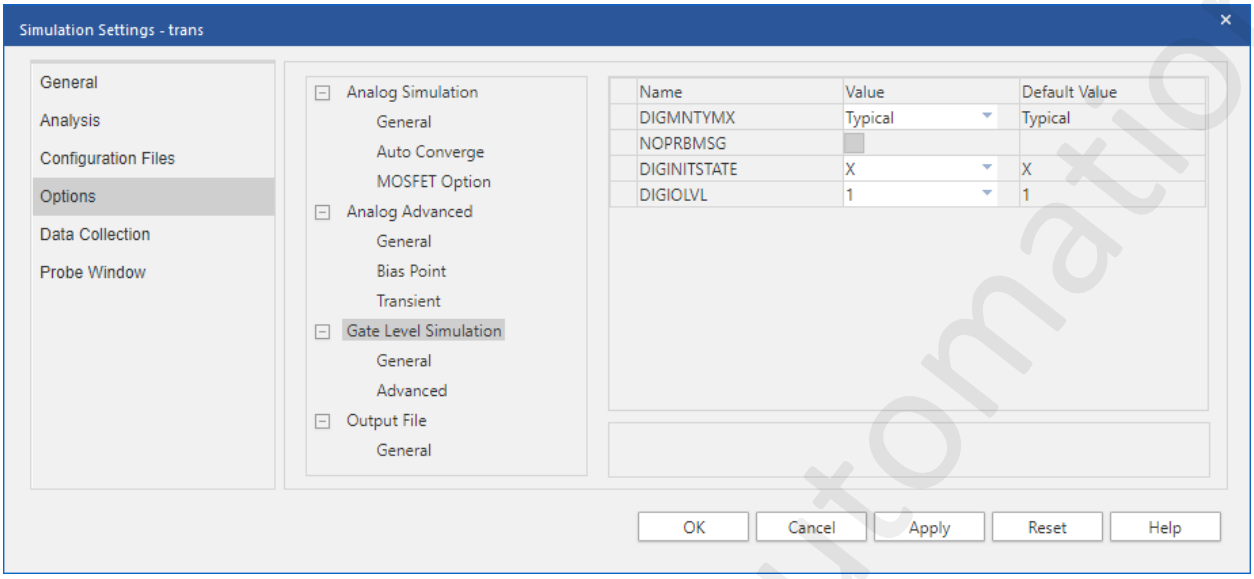
To quickly access these part symbols, use the *Place > PSpice Component…* menu command, and execute a *Search* for **Pullup** or **Pulldown**. You'll see that there are several symbols with predefined resistor values as well as generic ones. For any of the pullups or pulldowns you can set the resistance value by double-clicking on the value and changing it as you would for a regular resistor.

Setting up Digital Simulations

As stated earlier, digital simulation is event driven and is therefore restricted to transient simulations only. To execute a digital simulation, set up a transient analysis just as you did when simulating an analog circuit.

In addition to setting up the transient, you may need to set up some digital simulation options such as flip-flop initialization, timing, or I/O modeling settings. Access the digital simulation options in the *Simulation Profile* dialog by clicking on the ***Options*** tab. In the ***Category*** field select Gate-level Simulation.

Timing Mode	DIGMNTYMX	Typical
Initialize all flip-flops	DIGINITSTATE	X
Default I/O level	DIGIOLVL	1



Generally, the only item you will need to change in this dialog is in ***Initialize all flip-flops to:*** field where you can initialize all flip-flops to known state. The PSpice simulator uses ideal mathematical models for devices and assumes no noise is present. If you were to connect a flip-flop on a testbench, there would be enough noise in the circuit to set the flip-flop to a stable state (0 or 1). Flip-flop states at the beginning of a digital simulation is unknown by default since the simulator has no way of knowing what state you expect the devices to start in. Devices such as counters and shift-registers are modeled using flip-flops, so you may need to set this option even if you do not see any flip-flops in your design.

Viewing Digital Results

Digital traces are displayed in Probe similar to analog traces. Markers can be used on the Capture canvas (*Voltage Marker*) to automatically display the logic output in Probe, just like analog voltage markers. You can also add the digital traces after Probe has started by selecting the *Trace > Add Trace* menu command, or by clicking on the ***Add Trace*** toolbar icon ().

All digital traces in Probe are displayed at the top of the Probe plot, and have a state rather than a voltage level. Probe can display High, Low, Rising, Falling, Unknown, and Tristate. High and Low states are drawn in green as a single line. Rising and falling are drawn in yellow with a double line in the middle. Unknown is drawn as double red lines. Tristate is drawn as triple blue lines.

Buses

Buses can also be displayed in Probe and are drawn as double green lines with the bus value shown between the lines. Their values can be displayed in binary, octal, decimal, or hex format. Individual digital signals can be combined in Probe to be displayed as a bus. Bus format is:

```
{digital_signals_list};display_name;radix
```

or

```
{bus_prefix[msb:lsb]};display_name;radix
```

Example:

```
{Q2 Q1 Q0};MyBus;O    or  {Q2,Q1,Q0};MyBus;O
```

The line above specifies a 3-bit bus whose high order bit is the digital value at node *Q2*. On the Probe plot, Probe names the bus signal *MyBus* and values appear in octal notation. See the Probe Help for more detailed information on bus syntax.

Transient Sources

The sources used for a transient analysis are the same as those available in the Stimulus Editor. For an analog source: Piece-wise Linear, Sinusoidal, Pulse, SFFM, and exponential. Digital sources are: digital clock, signals, and busses. All of these sources can be generated using the Stimulus Editor sources.

There are source symbols in the source library for transient analysis that do not use the Stimulus Editor: Vsin, VSFFM, Vpulse, Vexp, and VPWL. Each of these sources simply have the necessary properties on them to generate the desired stimulus.

Stimulus Editor

These stimuli are stored in an ASCII file and are not a part of the design file. Instead, the file can be configured in the design so that it is read at the time of simulation. This is especially useful when a group of engineers are working on a project that share common stimuli such as clocks, reset signals, and other control signals.

Vstim, Istim, Digstim Symbols

From OrCAD Capture you can use the three stimulus symbols to access the Stimulus Editor (StmEd). The type of symbol you choose will vary with the type of stimulus you wish to create.

There are three special symbols for use with the Stimulus Editor:

1. **Vstim--**StmEd voltage sources
2. **Istim**--StmEd current sources.
3. **Digstim**--StmEd digital sources

These three stimulus sources are located in the library `Sourcstm.olb` which is located in the PSpice directory with other PSpice A/D part libraries.

It is important to use the correct stimulus symbol since the resulting netlist format for each symbol is different. For example, if you try to use the digstim symbol for an analog stimulus, you will get an error during simulation.

Defining a Stimulus

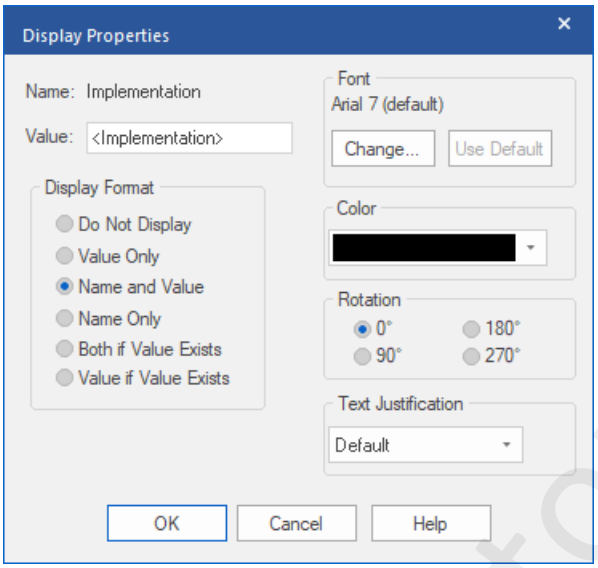
After you place a StmEd symbol, such as VSTIM from the SOURCSTM PSpice library, the stimulus is defined through the value assigned to the Implementation property. This property links a part or symbol to a simulation model, a schematic file, or in this case, the stimulus definition.

In the graphic above, the VSTIM symbol points to the analog stimulus named sin_60hz.

To Define a Stimulus:

1. From the schematic, double click the text ***Implementation*** included with the symbol.

The Display Properties dialog box opens.



2. Type text to define the value field.

As you type in a name value, remember, case is not important. Blank spaces are not permitted in name or value assignments.

3. To minimize visible text on the schematic page, double click the name value. Then choose the ***Value Only*** setting in the ***Display Properties*** dialog box.

This change is a cosmetic change that does not impact the simulation.

Accessing the Stimulus Editor

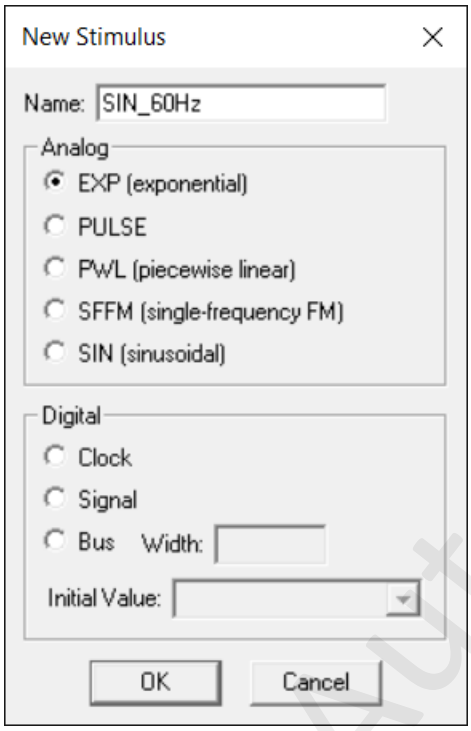
The Stimulus Editor creates transient analog and digital transient sources. The steps that follow explain how to access the Stimulus Editor from OrCAD Capture.

1. Select a stimulus source.

The symbol is highlighted.

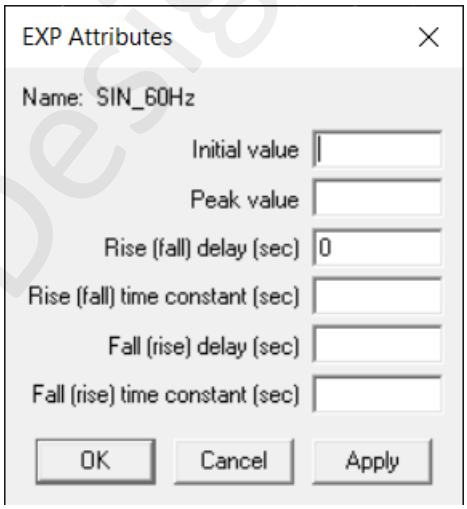
2. Select the ***Edit > PSpice Stimulus*** menu or click your right mouse button to display the pop up menu. Then select ***Edit PSpice Stimulus***.

The following dialog box opens.



3. Specify the desired type of stimulus and click the ***OK*** button.

A dialog box that has options that corresponds to your selection opens. The graphic that follows is the dialog box that opens when you choose the EXP (exponential) option.



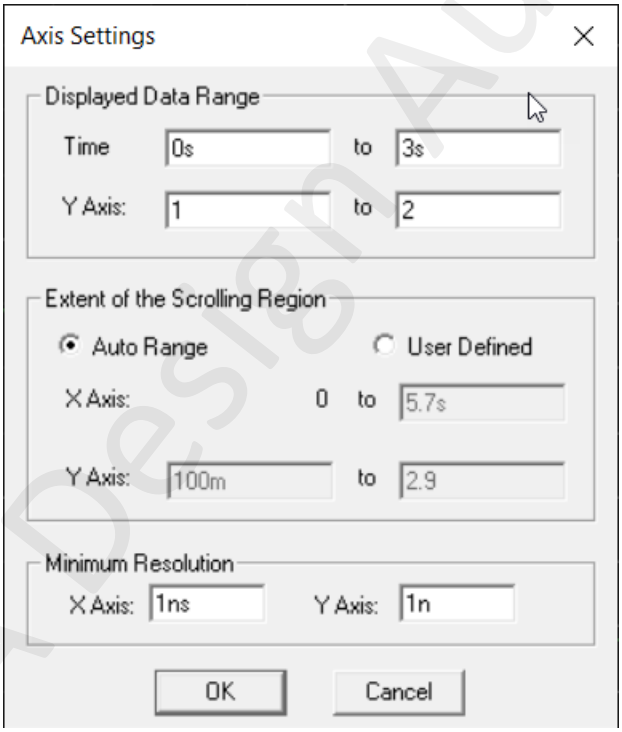
Creating and Editing Additional Stimuli

Once you are working in the ***Stimulus Editor*** you can add as many stimuli as you want by choosing ***Stimulus > New***.

You can also edit an existing stimulus. If the stimulus is displayed you can double-click on it to make it the active stimulus and then proceed to change its attributes, add and remove points, or make any other changes that you want. When you save the file, you will save the changes made to the stimulus.

Configuring the Display

It is possible to edit display parameters of the StmEd to either facilitate trace editing or alter visibility properties. For instance, if you are creating a Piece-wise linear (PWL) stimulus that moves in .5V increments at 10ms intervals, it is very helpful to set the display to have a minimum resolution of .5V and 10ms. This makes it very easy to place new points since the cursor will snap to grid at those intervals. You can access the axis settings dialog as shown in the graphic on the following page by choosing ***Plot > Axis Settings*** or by clicking on the appropriate toolbar button. 



Changing Colors in the Stimulus Editor

Your colors in the Stimulus Editor will likely be black background and green trace color. If you would like to change the default colors, save your file, close Stimulus Editor, open the pspice.ini file (C:\Apps\SPB_17.4\tools\pspice) and modify the colors in this section:

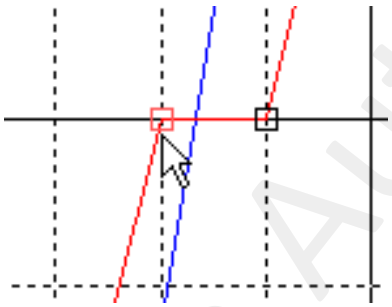
```
[Stimulus Editor DISPLAY COLORS]
```

```
NUMTRACECOLORS=6
BACKGROUND=BRIGHTWHITE
FOREGROUND=BLACK
TRACE_1=BLUE
```

Re-Open the Stimulus Editor and the new colors should be present

Correcting Mistakes

If you have made any placement errors, click on the vertex using the arrow cursor and then drag it to the correct location.



There are several ways to correct mistakes made when creating a PWL stimulus. You can move points, delete points, and add points (or vertices).

To Move a Point

1. Click on the desired vertex to select it. It will turn red to show that it is selected.
2. Click and drag the point to the new location.

The new location is constrained by the points to either side of it as well as the minimum resolution of the X and Y axes.

To Delete a Point

1. Click on the vertex to select it.
2. Press ***<Delete>*** to remove it.

To Add a Point

1. Click on the ***Add New Point*** toolbar button or select the stimulus name and select ***Edit > Add***. 

2. Using the Pencil  , click to add new points.

New vertices are constrained by the minimum X and Y resolution set.

3. Right-click to end placement mode.

Configuring Stimulus Files

A stimulus library is a file that contains one or more stimulus definitions. You might have a library of sine waves, PWL sources, or all the clock and control signals for your current project. Stimulus libraries are configured in the simulation profile dialog under the Configuration Files tab. Cadence does not ship any stimulus libraries with PSpice A/D, but you can create your own as needed.

As with model libraries, you can configure stimulus libraries locally to a profile, locally to a design, or globally. Local libraries are visible to only the current design. Global libraries are visible to every design opened or created on your system.

When you create a stimulus and save the file, it is automatically configured as a local library in your simulation profile. The library will have the same base name as the profile that was current when you accessed the Stimulus Editor but will have the .stl (stimulus library) extension. You can make the library global by selecting it in the simulation profile and clicking the ***Add as Global*** button.

Review of Adding a Stimulus

There are three things to keep in mind when using a stimulus from the Stimulus Editor:

1. A stimulus is stored in a stimulus library file, just as a simulation model is stored in a model library file. The name of the library is usually based on the name of the profile in which it was created but contains one or more stimuli with their own unique names that are usually not the same as the library name. For example, a stimulus library named clipper.stl might contain stimuli named Clock, Reset, 60Hz_sine, and Pulse_5mHz.

2. The stimulus library must be configured in the simulation profile under the Configuration Files tab so that it is visible to PSpice A/D during simulation. When PSpice A/D encounters a Stimulus Editor source in a design it will look through all of the configured stimulus libraries for a stimulus with the name listed in the symbol's implementation property.

3. You must use one of the Stimulus Editor symbols in the design in order to link to the stimulus definition. The implementation property on the symbol will be set to the name of the stimulus that you want to reference.

Mixed Simulation

Entering a schematic with both analog and digital devices is no different than entering a design of analog only or digital only components. All parts and symbols are placed on the design and wired together with wires or buses. The simulation is set up once for the entire design. If there are flip-flops in use, you will also need to set up the flip-flop initialization.

Viewing the Results in the Probe Window

The Probe display will be slightly different when running a mixed-signal simulation, split into analog and digital sections. The digital section will always be at the top of the screen while the analog signals will be at the lower portion. You can set the relative size of the digital display by selecting *Plot > Digital Size…* menu command, and then enter a percentage of the screen that the digital section may use. You can also drag the bottom border of the digital section to re-size it.

Both the analog and digital traces share a common time axis. During simulation all analog voltages and currents and all digital states are calculated at each time step. There is no separate analog or digital simulation when running a mixed signal design.

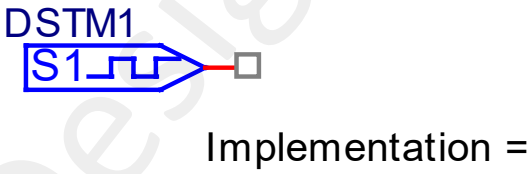
Appx II, Lab 1 Creating Digital Stimulus

Lab Objectives

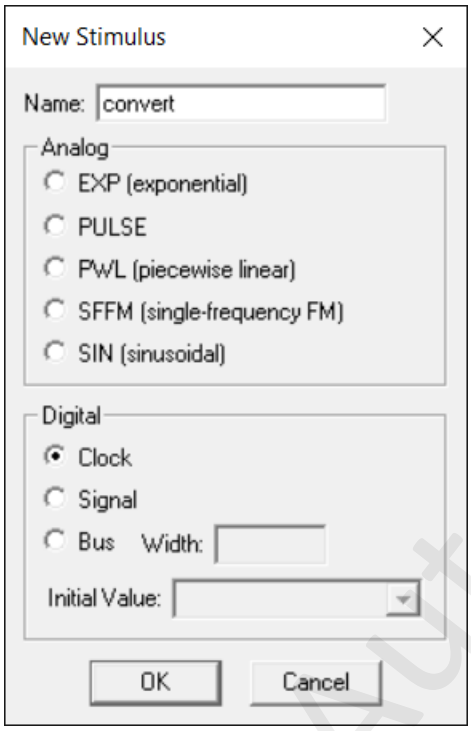
- Create a digital stimulus for use in a digital design

Creating the Stimulus

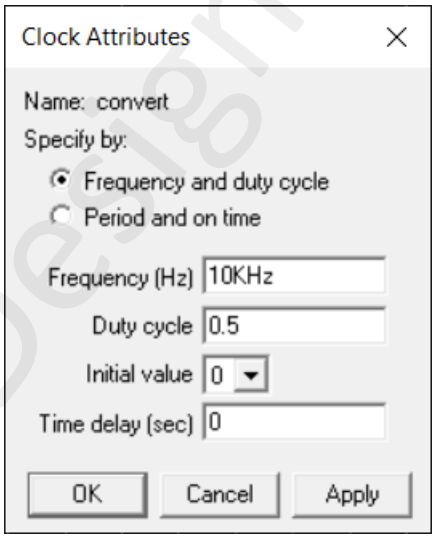
1. Start a new project in Capture, name it ***MixedSignal***
2. In the newly created schematic page, place a *DigStim1* part to create access to the *Stimulus Editor*
    - Select *Place* > *PSpice Component...* > *Search*
    - Enter **DigStim1** in the *Search Field* and select the <*Return*> key
    - In the *Results* area, select the *DigStim1* (at the top of the list), then double-click on the entry to attach to the cursor and place one instance in the schematic page



    - Select *DSTM1*, and then with a RMB, select *Edit PSpice Stimulus*. You will then be taken to the *Stimulus Editor* where you can define your digital stimulus for this design
3. In the *New Stimulus* panel, enter ***Convert*** for the name, and select the *Clock* radio button to create a digital clock type and select *OK*



4. In the next panel, enter ***10k*** for the clock frequency and select ***OK***



This is what we will be using to stimulate our digital design. The *Stimulus Editor* will then draw the clock

5. Select *File > Save*, or select *Save* icon from the toolbar (). You will be prompted to "update" Capture schematic. Select *Yes* to update the *Implementation* field for *DSTM1* (points the part to the waveform name you just created), and will add the stimulus file

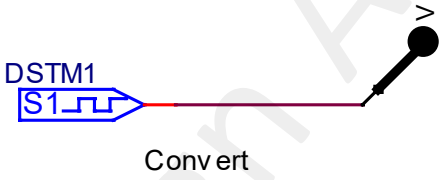
to the simulation profile (once you create it) so PSpice can find the proper referenced stimulus

6. You can either exit the *Stimulus Editor* or minimize the window for later use if desired

Verifying Stimulus operation

1. Back in the schematic page, place a *Voltage Probe* on the output of *DSTM1*

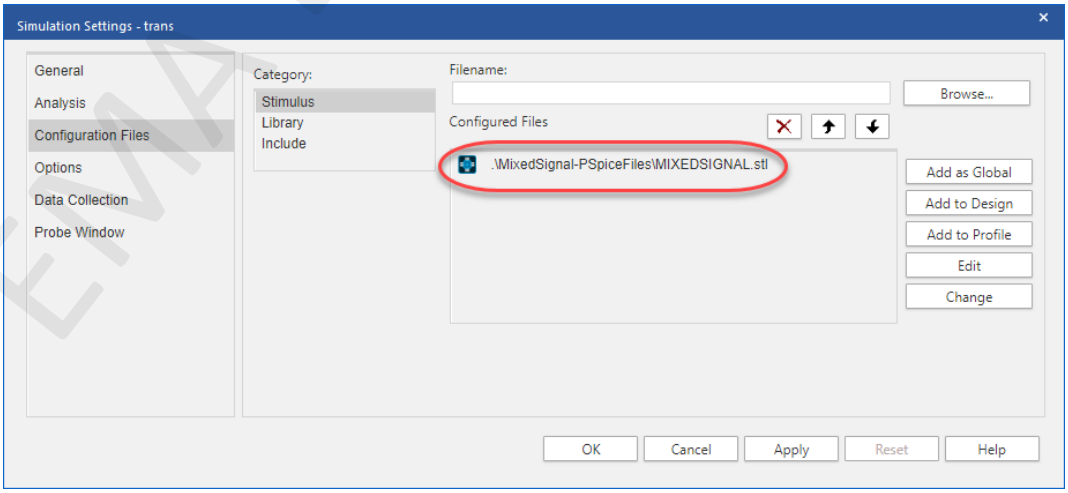
If desired, you can edit the displayed *Implementation* property of *DSTM1* and select to just display the *Value* to clean up the schematic a bit.



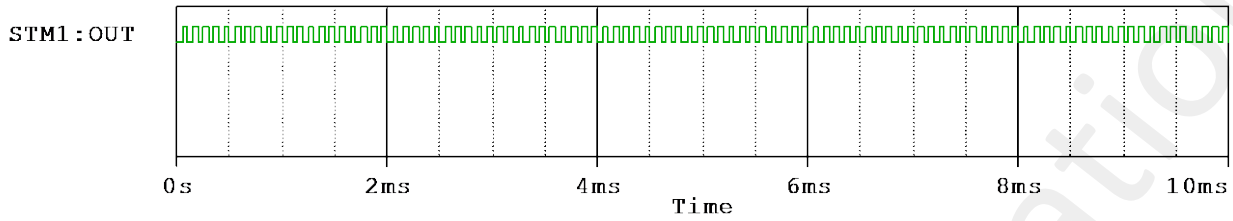
2. Create a simulation profile named **Trans** for a transient simulation to run for ***10ms***

3. Select the *Configuration Files* tab to see that the project has automatically configured the stimulus file you just created

PSpice MUST know where to find the stimulus files containing the user defined stimulus for a simulation.



4. Execute the simulation and observe that the stimulus is generating a 10KHz clock



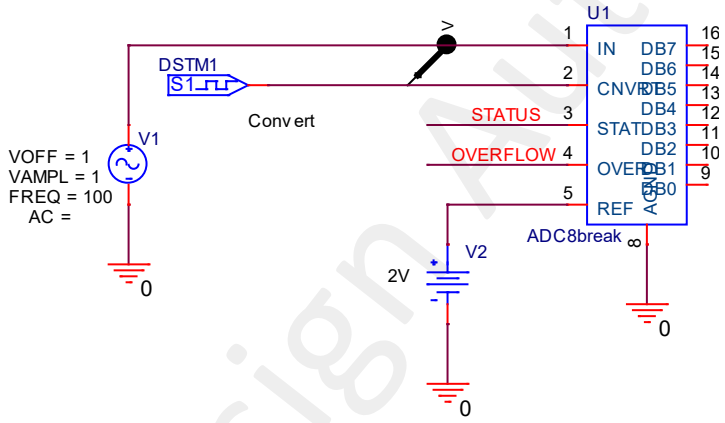
Notice the display configuration has no vertical axis. In purely digital designs, Probe will display digital signals with only state change information in the vertical axis.

Appx II Lab 2 Creating an ADC (mixed signal) design

Lab Objectives

- Create an A-D converter design
- Use the stimulus generated from lab 1 to drive an ADC
- Configure and run a mixed signal simulation
- Examine the results in Probe

Creating the Circuit



1. Using the *DSTM1* created in the last lab, add on to the circuit as above using the ***ADC8break* part** (*Place > PSpice Component… > Digital > ADC > 8bit*). This is an A-D converter with an 8-bit output (one of the digital primitives available). Pin descriptions are:

    - IN – Analog input
    - CNVR – Convert control. A high on this pin will trigger a conversion and present data on the outputs
    - STAT – Output to indicate device status. Delay control is available
    - OVER – Output to indicate out-of-range situations for either below AGND pin or above REF pin. Delay control is available
    - REF – Reference voltage input for comparing voltage at the IN pin. As voltage at this pin approach IN pin, the output approaches max value
    - AGND – Analog reference pin (device ground reference)

- DB0-DB7 – Digital output. Will present 8-bit digital value equally distributed between REF and AGND analog values. Delay control is available

2. Place the net aliases as shown for easy signal identification in Probe. Values for the stimulus devices are:
   - V1 – Sine wave source, 100Hz @ 1V amplitude, with 1V of DC offset
   - V2 – DC voltage source of 2V for reference (allows inputs signals of 0-2V)
   - DSTM1 – Stimulus file created in the last lab

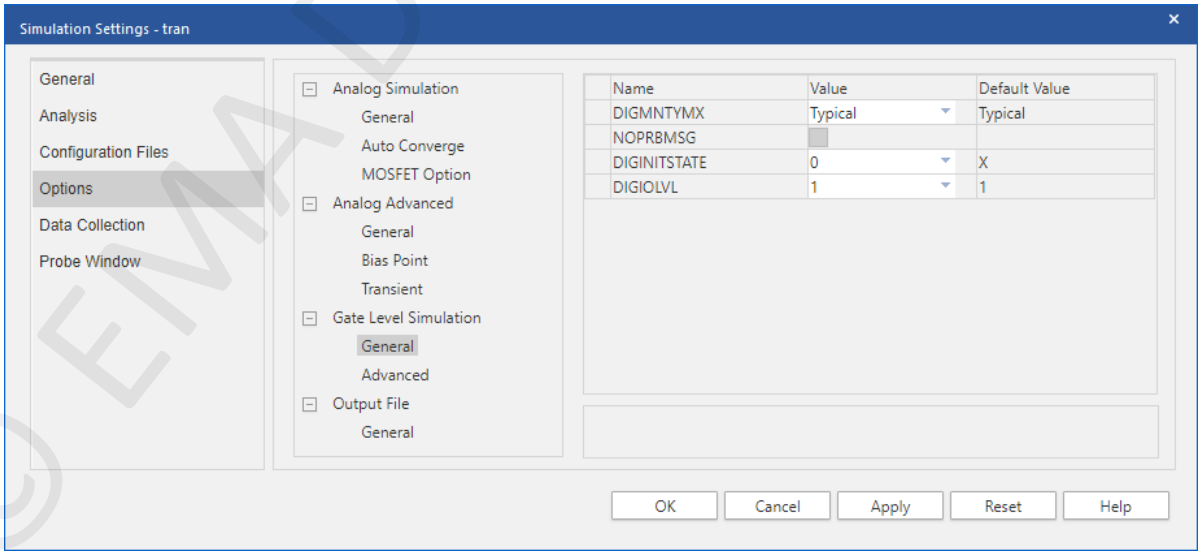
Executing the Simulation

We'll use the same transient simulation profile as was used to test our stimulus file results (Transient simulation to 10ms).

1. Open the simulation profile for editing

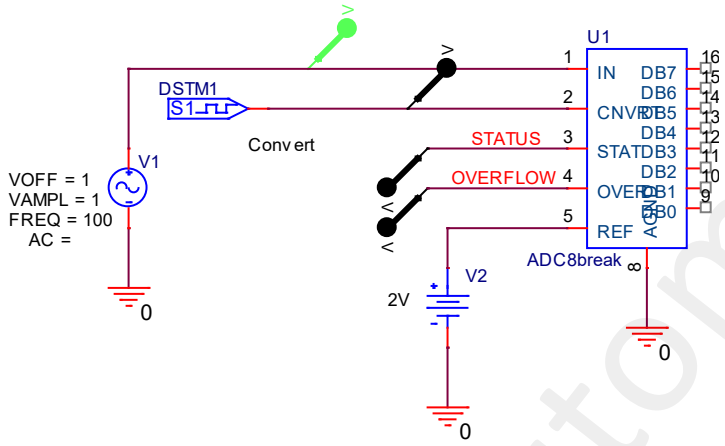
2. Select the *Options* tab, and the *Gate-level Simulation* category to initialize the flip-flops in the device to "**0**" for the start of the simulation

Timing Mode	DIGMNTYMX	Typical
Initialize all flip-flops	DIGINITSTATE	0
Default I/O level	DIGIOLVL	1



3. Select the *OK* button to close the profile

4. Place voltage probes on the circuit as indicated below:

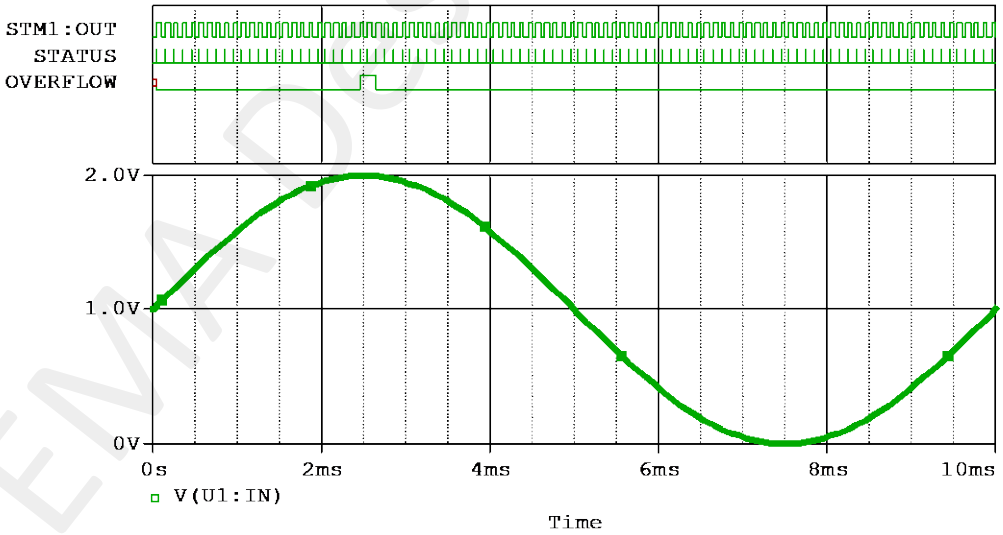


5. Save the file

6. Run the simulation (*PSpice > Run*, or icon in toolbar, or F11 key)

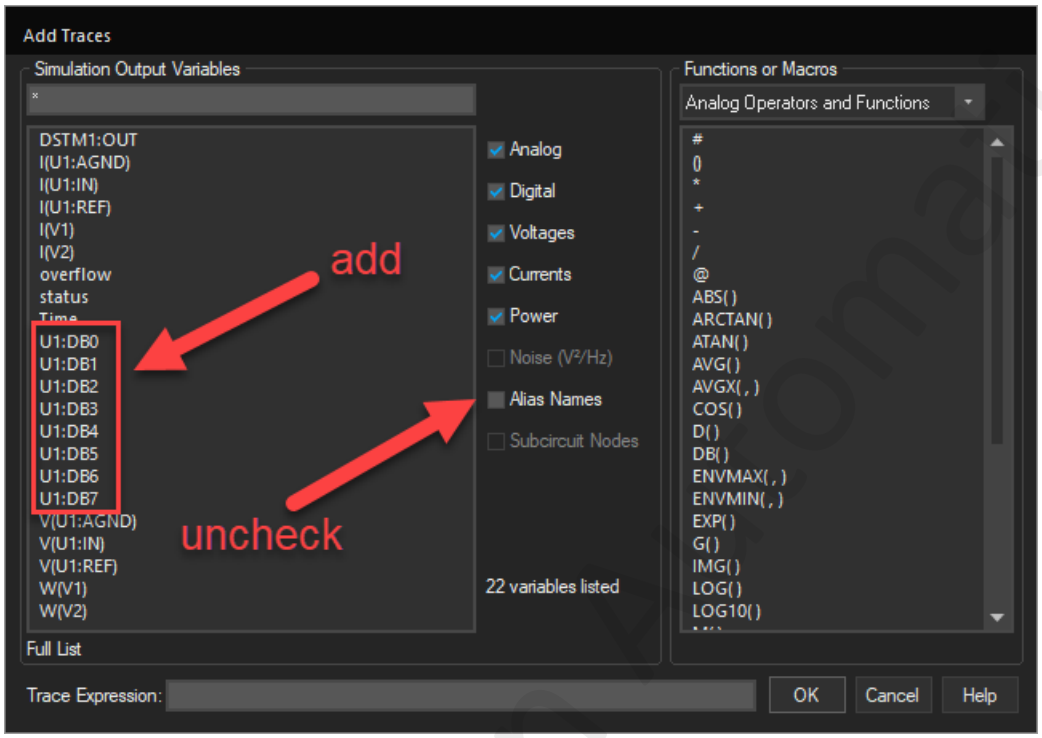
Viewing the Results in the Probe Window

1. When the simulation finishes, you should see 1 cycle of the analog waveform in the Probe window



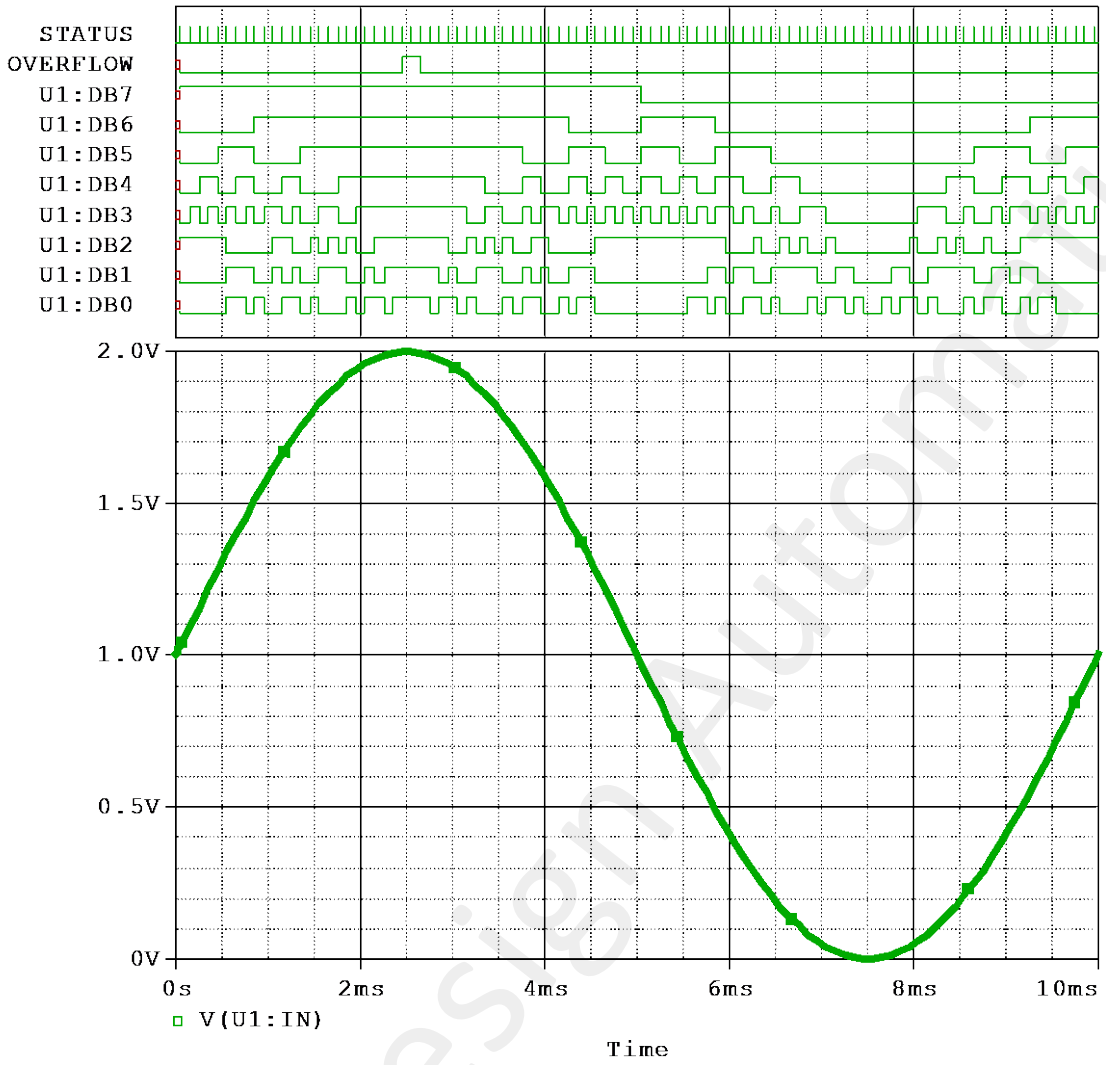
2. Select ***Add Trace*** toolbar icon ( )

3. To clean up the large list, uncheck the *Alias Names* checkbox in the middle of the window



4. In the *Add Trace* panel, select *U1:DB7, U1:DB6, U1:DB5, U1:DB4, U1:DB3, U1:DB2, U1:DB1, U1:DB0* from the list to put each individual output signal in the display, and select *OK*

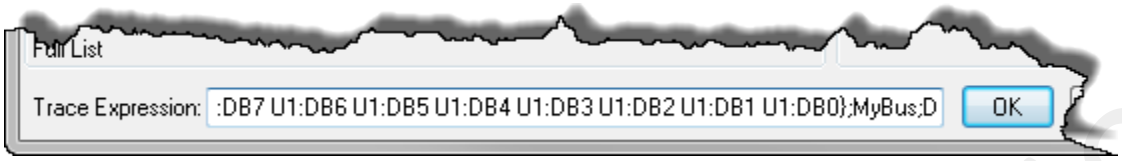
5. You should see something like the display pictured below. We can see that with every rising edge of our stimulus feeding the *Convert* pin, the ADC outputs are presenting new data. Notice that once the input signal begins to equal or exceed the reference voltage (2v in this case), the output reaches 255 and the *Overflow* node goes high



We can verify that the circuit is working as expected with this display, but there is a better way of looking at the results. Instead of looking at individual bits of our outputs, we can look at the outputs as a decimal bus.

6. Click the ***Add Trace*** toolbar icon ()

7. In the *Trace Expression* field, enter an open curly brace '{', then sequentially select the output bits from the list of available nodes (*U1:DB7* through *U1:DB0* , in that order), followed by the closing curly brace. Then add the remaining bus name and radix separated by semi-colons. The Trace expression field should look like this when complete:

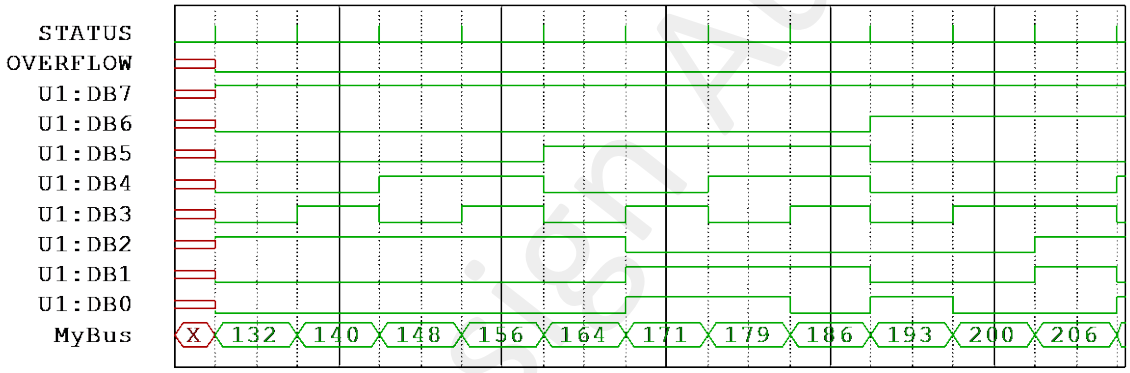



```
{ U1:DB7 U1:DB6 U1:DB5 U1:DB4 U1:DB3 U1:DB2 U1:DB1
U1:DB0};MyBus;D
```

Enclosing the signal names in curly braces tells Probe that this is a bus definition. *MyBus* will be the displayed name for the bus, and the "*D*" at the end specifies the display format, which is decimal (base 10). Alternatively, you can use 'H' for hexadecimal (base 16), 'B' for binary (base 2) and 'O' for octal (base 8). Note the radix is NOT case sensitive.

8. Select ***OK*** to display the bus.

9. Click on the Zoom In toolbar icon ( ) to increase the resolution



The bus is added at the bottom of the display. You can compare the bus values with the individual traces above. If you see a bus value of "*", this means that the bus segment is not wide enough to display the value. **Note:** Since this is a 'primitive' digital device, there are no flip-flops that are contained within sub-circuit to actually initialize in the design, therefore, there is a period of time from the *T=0* to the first rising edge on the *Convert* pin where the outputs are unknown.

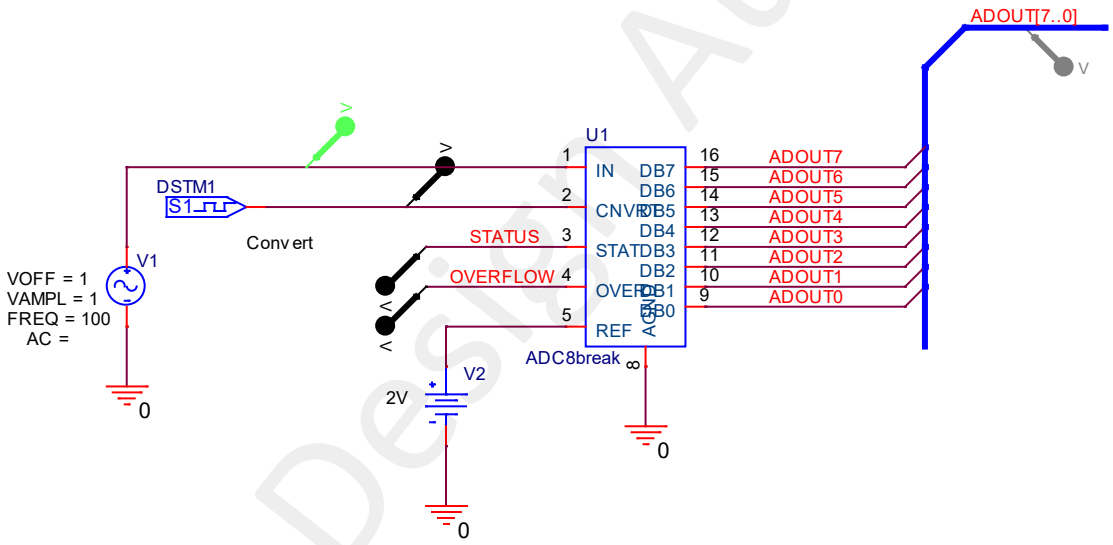
Appx II Lab 3: Adding a Bus to our ADC design

Lab Objectives

- Add a bus to our design
- Run the simulation and observe bus in Probe

Placing the Bus

1. Using the Design made in the previous lab, select the ***Place Bus*** toolbar icon ( ), and draw a bus as seen in the schematic below. You can make angled lines on the bus by holding the SHIFT key as you click a corner on it



2. Label the bus **`ADout[7:0]`**

Connecting the Bus

1. Connect the *DBx* outputs of *U1* to the bus. Use the *Place > Bus Entry (E)* command or use the *Place Bus Entry* icon ( ) from the toolbar to get the angled lines into the bus
2. Label the wires *ADout0* through *ADout7* from bottom to top in the same order as shown above

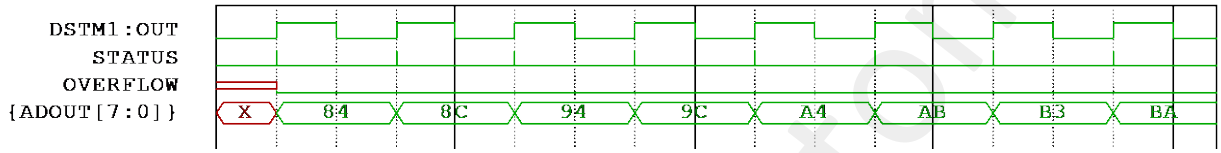
Hint: You can use the auto-incrementing feature to speed up the process

3. Place a *Voltage Marker* on the bus

Running the Simulation

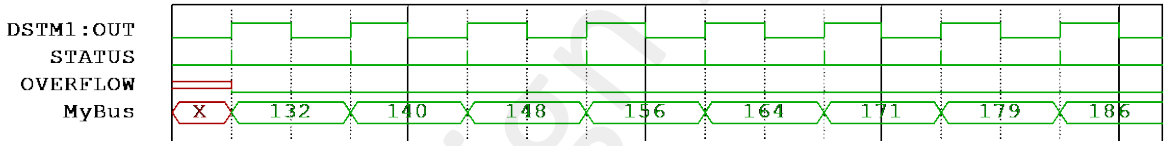
1. Run the simulation (*PSpice > Run*, icon or F11 key)

2. *Zoom* in so that you can see the values for the bus. The default radix is hex



3. Double-click the name of the bus

4. Change the *Trace Expression* to read **{ADOUT[7:0]};MyBus;d**. The bus name has now changed to MyBus and the values are now displayed in decimal format



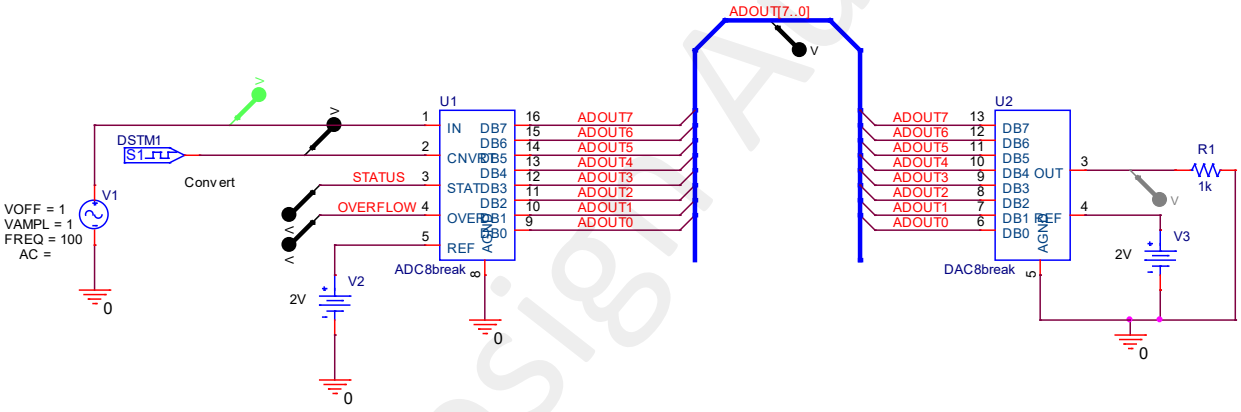
Appx II Lab 4: Adding a DAC onto our ADC design

Lab Objectives

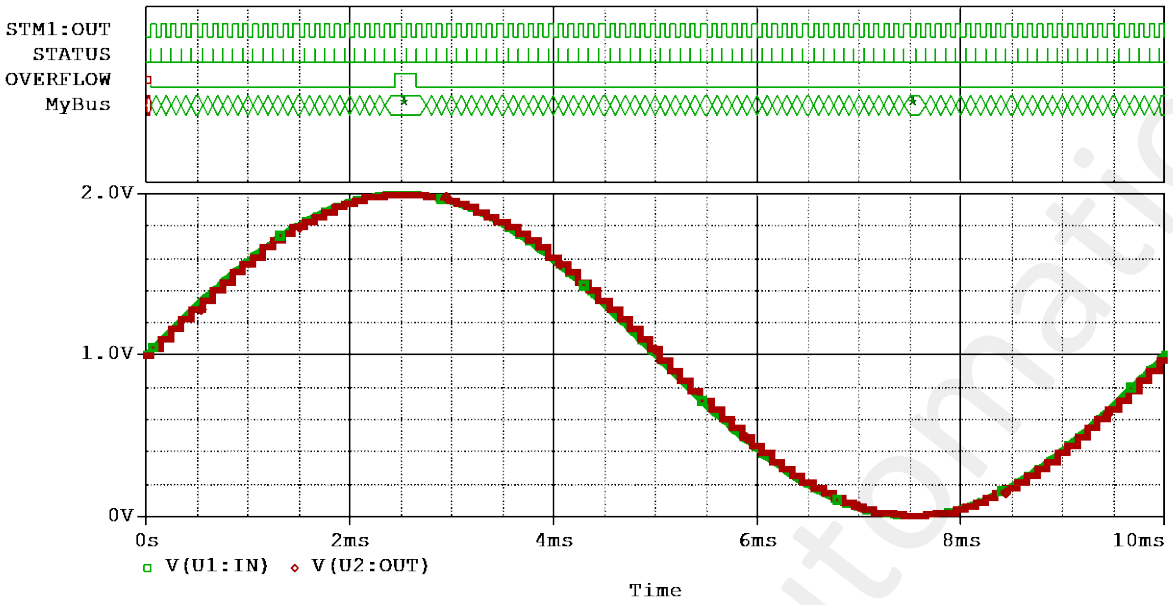
- Add a D-A converter on to our ADC design from previous lab
- Simulate the mixed A/D circuit
- View the results in Probe

Adding the DAC part

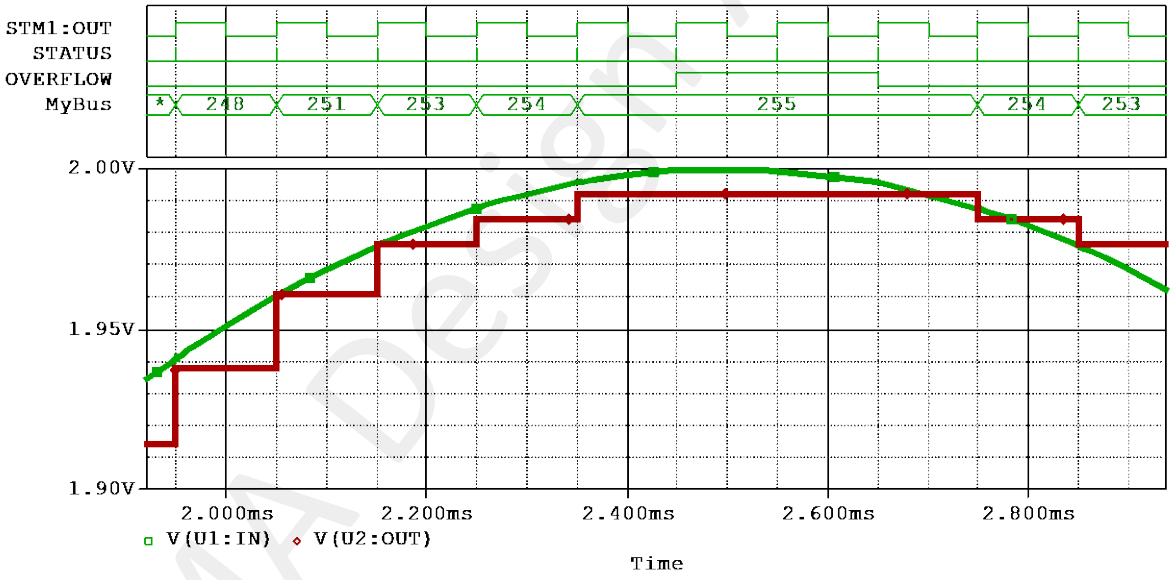
1. Using the ADC design from the previous lab, add the *DAC8break* part to the design to re-generate our original analog sine wave input



   - Select *Place > PSpice Component… > Digital > DAC > 8bit* command to place the DAC
   - Add the extra *VDC* part as a 2V voltage reference and a load resistor as shown
2. Run the analysis
3. Examine the output in the Probe window



You can zoom in on the waveform and see the stepping details of the newly created analog waveform based off of the 8 bits of digital signals sent through the bus.



Appx II Lab 5: Optional: Modifying the Digital Timing Model

Lab Objectives

- Manipulate Digital Timing model to add delays to ADC

Available Timing Parameters

This digital device in this model is just a primitive but it does consist of the same digital model structure of all other digital parts in PSpice A/D. This optional section illustrates how to alter the timing model of this *ADC8break* part.

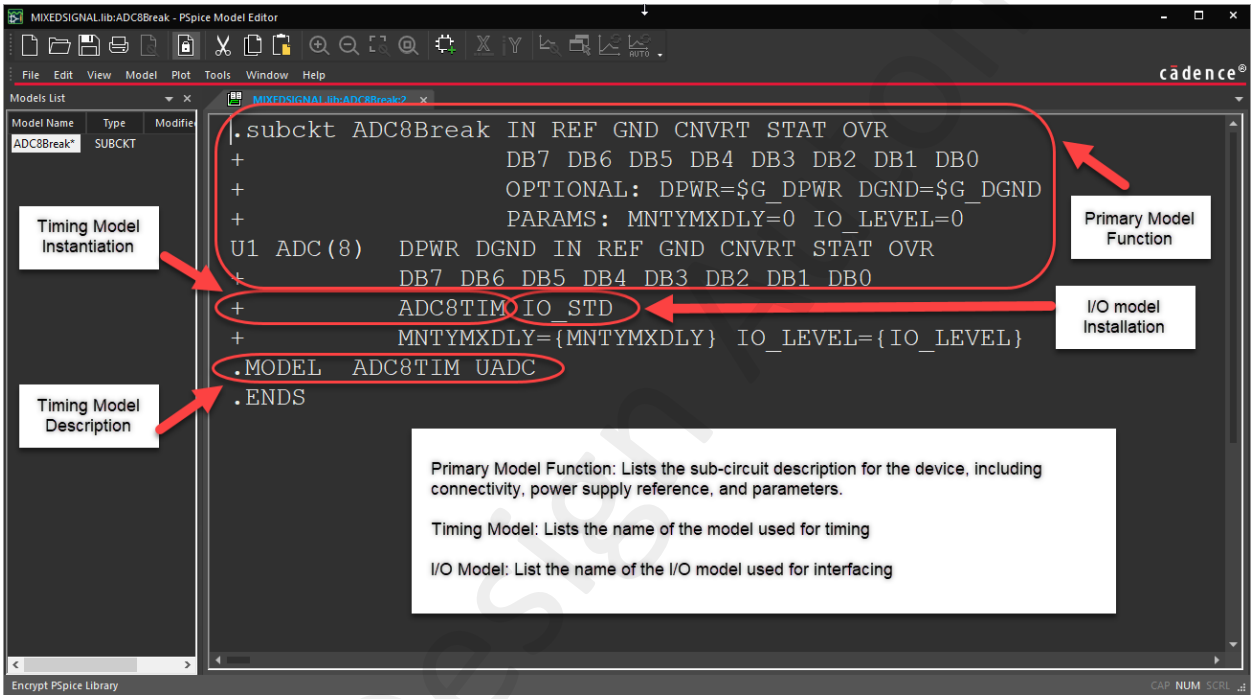
Below is a chart listing the three main timing parameters available for the *ADC8break* device, each having a minimum (MN), typical (TY), and maximum (MX) parameter. By default, all the timing associated with this part is set to *0s* which gives *immediate* response characteristics for the part (you may have noticed this already).

Model Parameter	Description	Units	Default Value
TPCSMN	propagation delay: rising edge of convert to rising edge of status, min	sec	0
TPCSTY	propagation delay: rising edge of convert to rising edge of status, typ	sec	0
TPCSMX	propagation delay: rising edge of convert to rising edge of status, max	sec	0
TPDSMN	propagation delay: data valid to falling edge of status, min	sec	0
TPDSTY	propagation delay: data valid to falling edge of status, typ	sec	0
TPDSMX	propagation delay: data valid to falling edge of status, max	sec	0
TPSDMN	propagation delay: rising edge of status to data valid, min	sec	0
TPSDTY	propagation delay: rising edge of status to data valid, typ	sec	0
TPSDMX	propagation delay: rising edge of status to data valid, max	sec	0

The model parameters listed above are applied to the digital model that controls the behavior of the *ADC8break* part used in this lab. Follow the steps below to apply *typical* timing values to this model (*minimum* and *maximum* values can be applied analogously).

Accessing and modifying the digital model

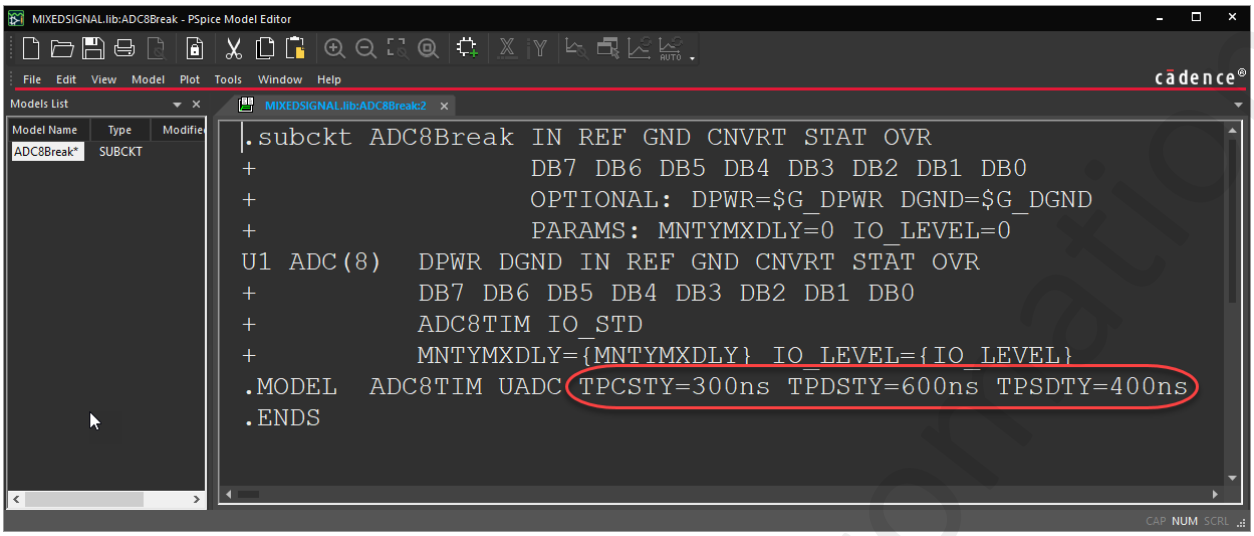
1. In the Capture schematic, select *U1* and with a RMB, select the *Edit PSpice Model* command. You should see the Model Editor tool open up with the *ADC8break* part model opened up and ready for editing (see image below)



2. The timing information for the part will need to be added to the *.MODEL* statement. Add the following text to the end of the *.MODEL* statement:

`TPCSTY=300ns TPDSTY=600ns TPSDTY=400ns`

so that the statement now looks like this:

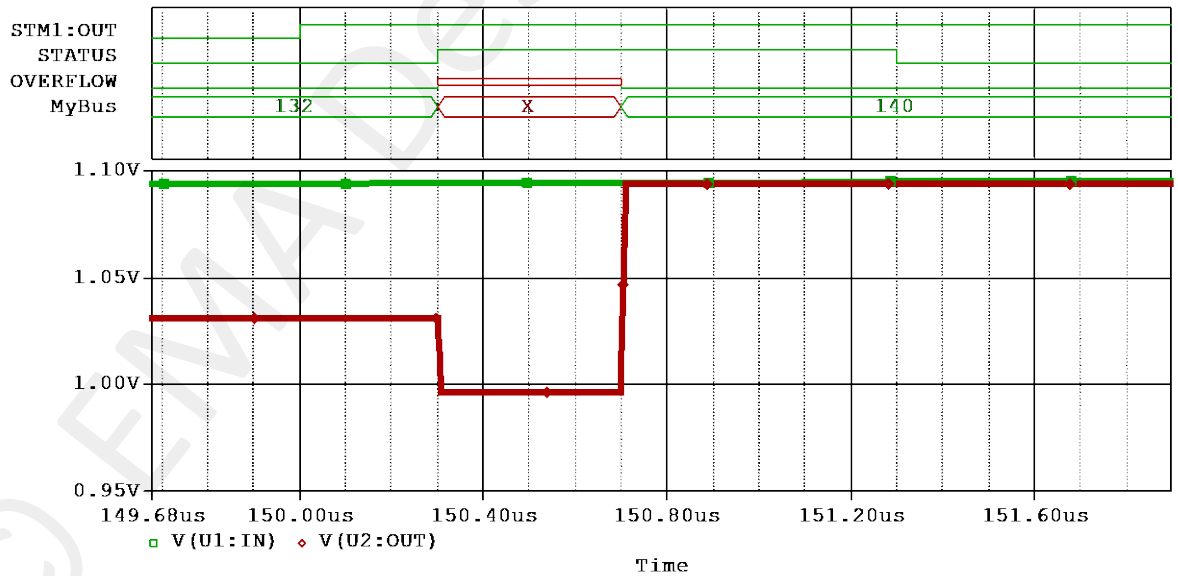


3. Save the file in the Model Editor to create a local copy of the model library for the digital part. This will also automatically configure the local model library so that PSpice will use this library reference instead of the default digital library from which the *ADC8break* was originally referenced

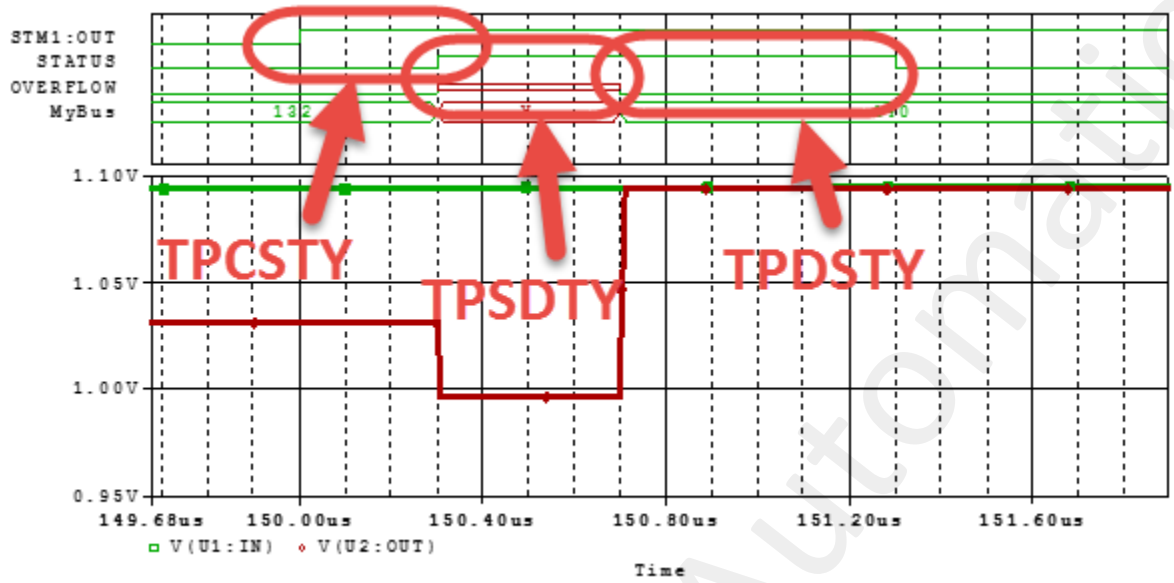
Analyzing the Results

1. Re-run the simulation (*PSpice > Run*, icon or F11 key)

2. *Zoom* in tight so that you can see the delays present between the outputs of the *ADC8break* part. You should see something that looks like the below screen shot.

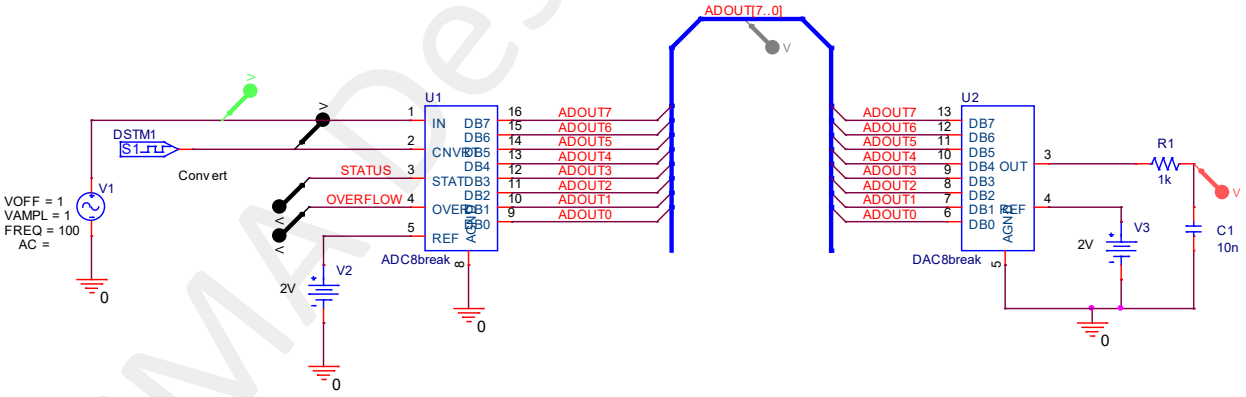


3. Try to identify the different delays that you're seeing and how they correspond to the waveforms above

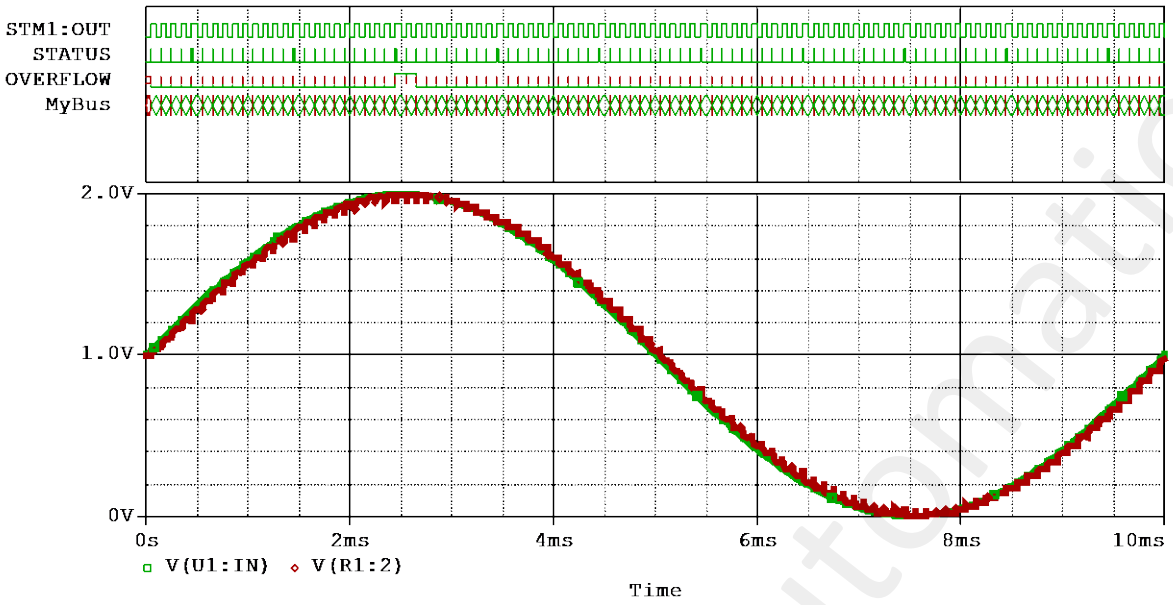


Cleaning up the Transitions

There are a number of good ways to clean up the transition spikes that we're seeing due to the effect of the timing becoming non-ideal for the ADC. The simplest way would be to add a high frequency filtering cap on the output and moving the probe to the other side of the resistor:



This appears to clean things up pretty well.



If you wanted to engineer something a little more elegant, you sure could but we'll leave it at this for the time being.

Additional Example Files

Open and run any of the other MIXSIM projects (`OSC.opj, FSK.opj, DSAD.opj,` or `Motordrv.opj`). The path to the MIXSIM projects:

```
C:\Apps\SPB_17.4\tools\pspice\capture_samples\mixsim
```

You could also open some good PSpice example files on digital modeling in the ***Help > Learning PSpice*** tool and inside the *PSpice Application Notes* folder (such as Digital Timing Simulation, Digital Frequency Comparator) or inside the *Basic Electronics* folder (such as Digital Electronics and Data Converters)

Appendix II Lab 6: Magnetics Part Editor

Creating a Model with the PSpice Magnetic Parts Editor

The Magnetic Parts Editor is a tool that will facilitate the design of magnetic components, including inductors and transformers. The tool provides templates to reduce manual calculations of magnetic parameters for choosing the needed core, windings, and insulation material.

Topologies Supported by Magnetic Parts Editor

The first step in using the Magnetic Parts Editor wizard is to select the type of magnetic component you wish to build. The following components can be chosen to be designed using the wizard:

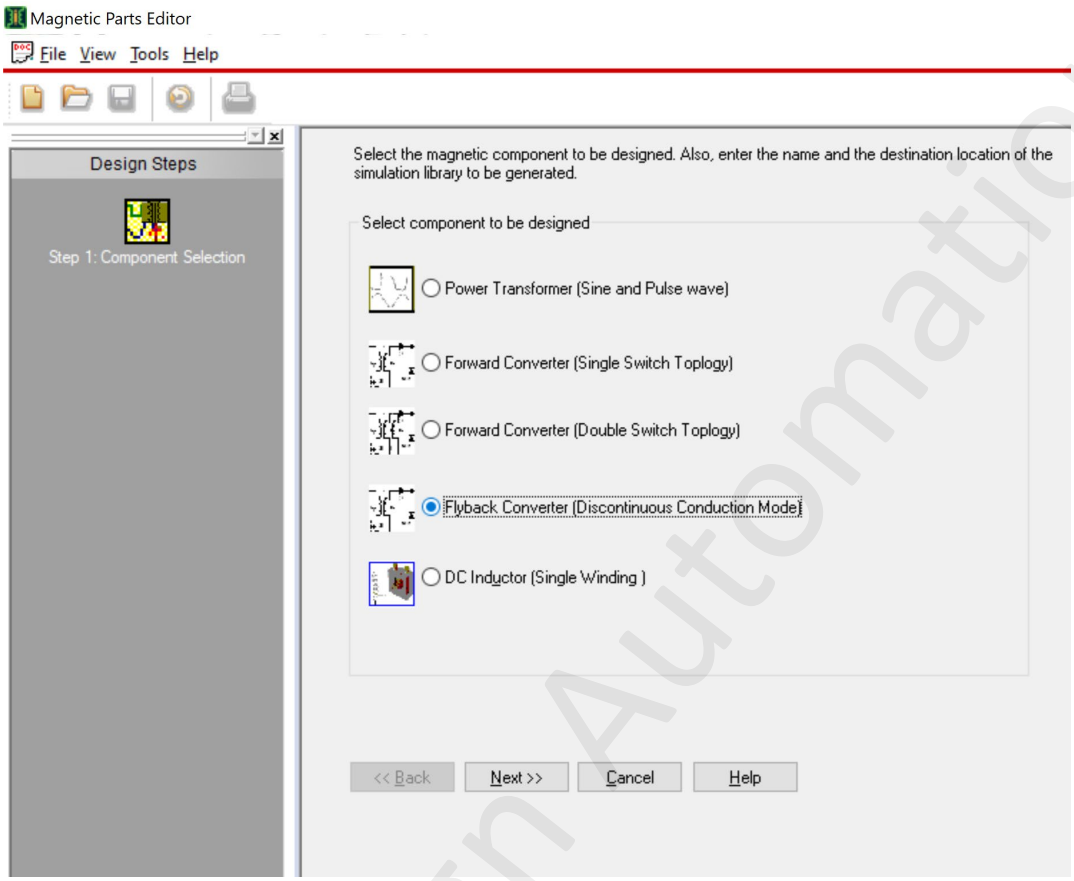
- Power Transformers
- Forward Converters (single and forward switch)
- Flyback Converters (operating in discontinuous conduction mode)
- DC Inductors (single winding operating in continuous conduction mode)

Building a Flyback Converter

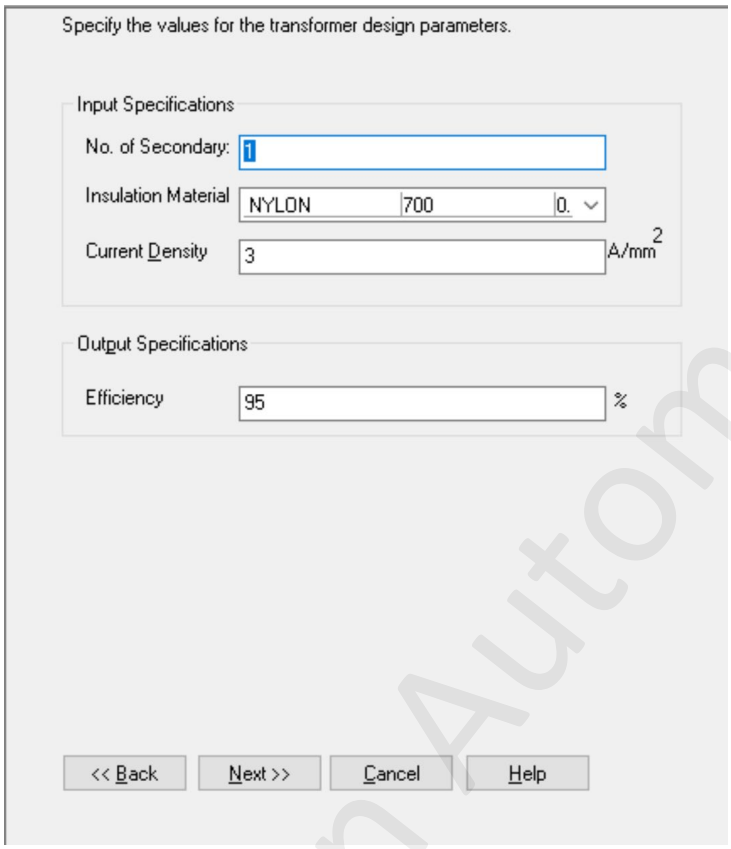
4. Launch the Magnetic Parts Editor by ***Start > All Programs > Cadence PCB Utilities 17.4-2019 > PSpice Magnetic Parts Editor 17.4***

5. Choose ***File > New*** from the menus.

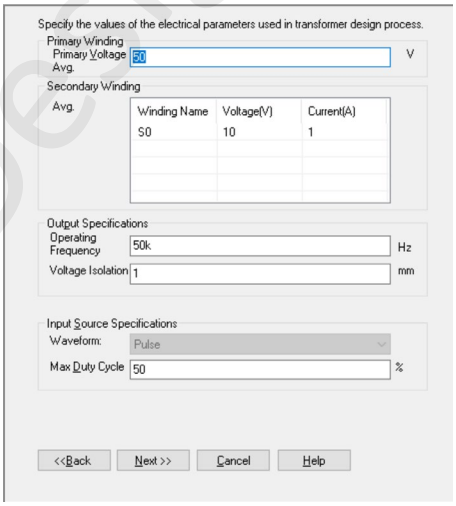
6. You should now see the list of support magnetic components. Select Flyback Converter (Discontinuous Conduction Mode).



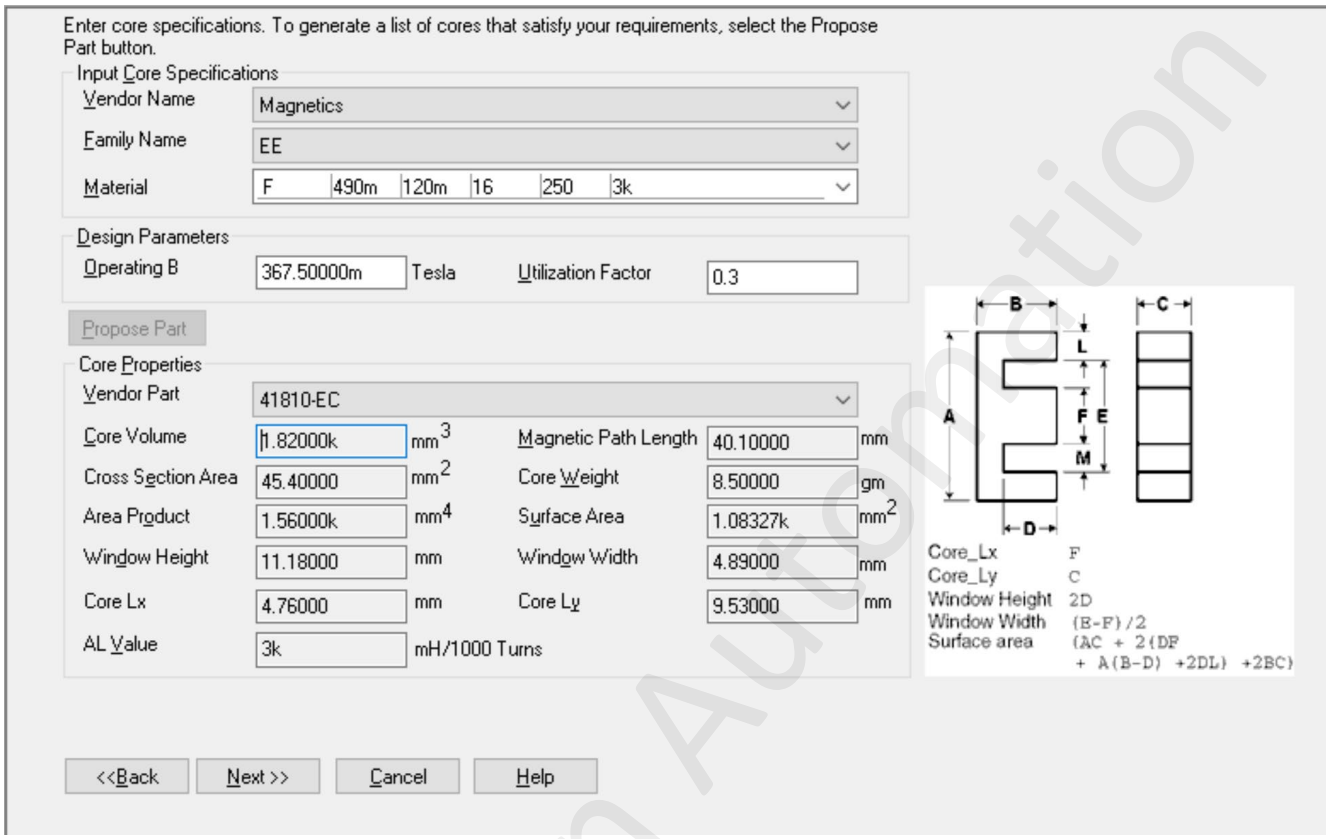
7. Click Next. Select only 1 secondary winding with nylon material and the default current density and output efficiency.



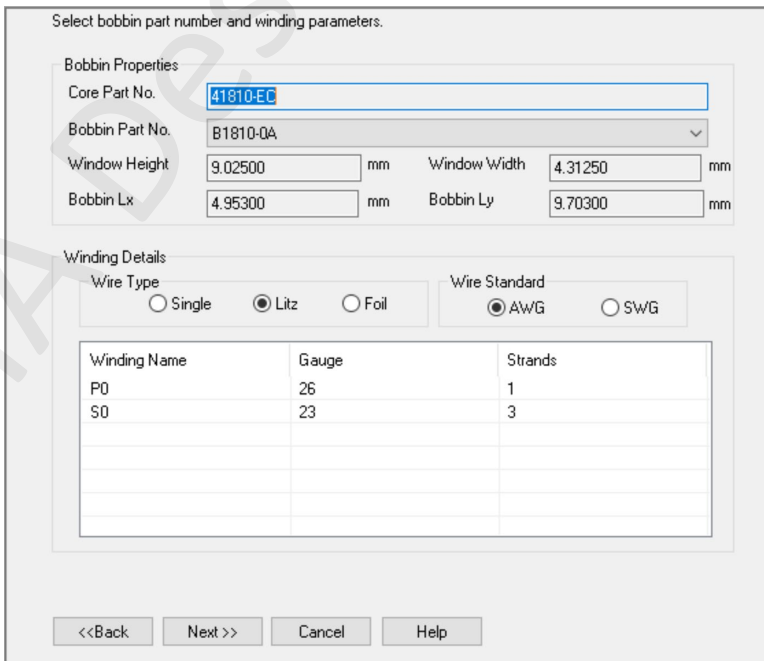
5. Click Next and now we specify our desired primary voltage of 50V and secondary voltage of 10V. Leave the other values as their defaults.



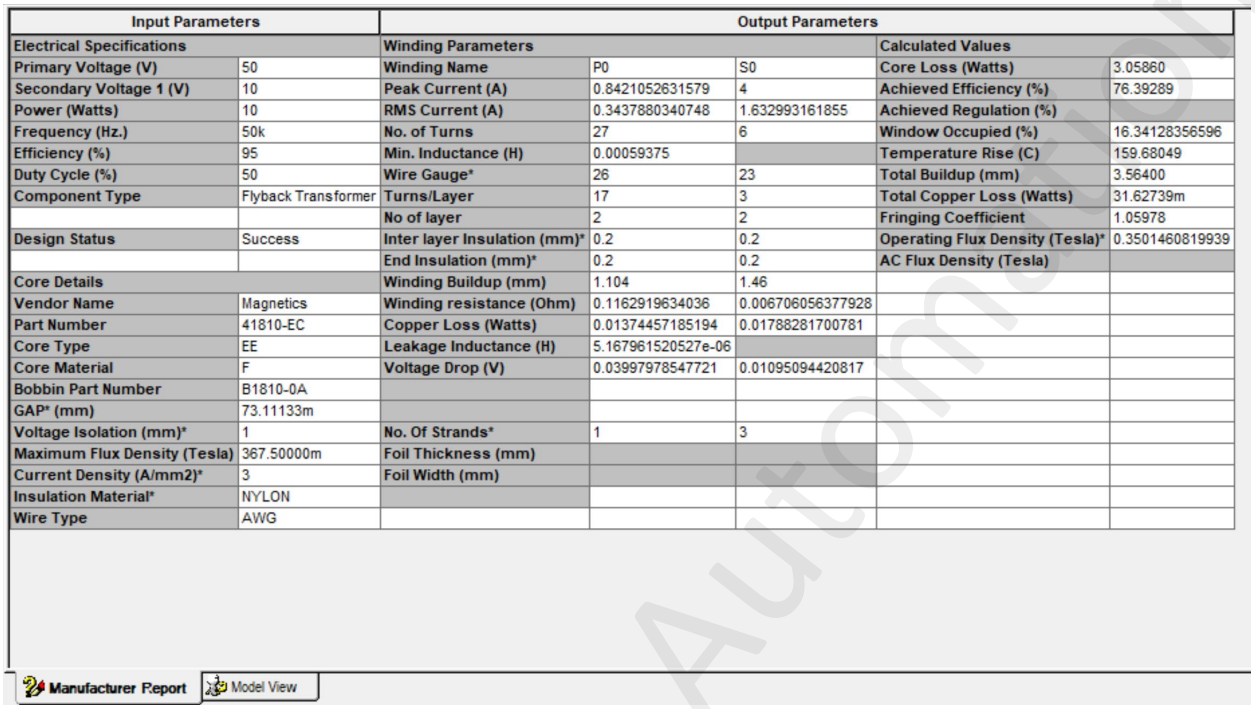
6. With our specifications, we can now ask the wizard to propose a magnetic core from the EE family. Click the dropdown for Material and select the F material. Click Propose Part and you should see the vendor part 41810-C proposed as the desired magnetic core. Click Next.



7. Now we can see the proposed bobbin part B1810-0A for our core. Select Litz for the winding details and leave the provided PO gauge of 26 and S0 gauge of 23 with 3 strands.



8. Click Next to see a final summary of the specifications as well as the model itself.



Input Parameters		Output Parameters				
Electrical Specifications		Winding Parameters			Calculated Values	
Primary Voltage (V)	50	Winding Name	P0	S0	Core Loss (Watts)	3.05860
Secondary Voltage 1 (V)	10	Peak Current (A)	0.8421052631579	4	Achieved Efficiency (%)	76.39289
Power (Watts)	10	RMS Current (A)	0.3437880340748	1.632993161855	Achieved Regulation (%)	
Frequency (Hz.)	50k	No. of Turns	27	6	Window Occupied (%)	16.34128356596
Efficiency (%)	95	Min. Inductance (H)	0.00059375		Temperature Rise (C)	159.68049
Duty Cycle (%)	50	Wire Gauge*	26	23	Total Buildup (mm)	3.56400
Component Type	Flyback Transformer	Turns/Layer	17	3	Total Copper Loss (Watts)	31.62739m
		No of layer	2	2	Fringing Coefficient	1.05978
Design Status	Success	Inter layer Insulation (mm)*	0.2	0.2	Operating Flux Density (Tesla)*	0.3501460819939
		End Insulation (mm)*	0.2	0.2	AC Flux Density (Tesla)	
Core Details		Winding Buildup (mm)	1.104	1.46		
Vendor Name	Magnetics	Winding resistance (Ohm)	0.1162919634036	0.006706056377928		
Part Number	41810-EC	Copper Loss (Watts)	0.01374457185194	0.01788281700781		
Core Type	EE	Leakage Inductance (H)	5.167961520527e-06			
Core Material	F	Voltage Drop (V)	0.03997978547721	0.01095094420817		
Bobbin Part Number	B1810-0A					
GAP* (mm)	73.11133m					
Voltage Isolation (mm)*	1	No. Of Strands*	1	3		
Maximum Flux Density (Tesla)	367.50000m	Foil Thickness (mm)				
Current Density (A/mm2)*	3	Foil Width (mm)				
Insulation Material*	NYLON					
Wire Type	AWG					

Manufacturer Report | Model View

We have now created a PSpice model for a flyback converter with a real magnetic core and real bobbin. This new subcircuit model can now be mapped to a symbol using the Model Editor and then simulated on a schematic.