

## **LAB 3: DC Simulations and Circuit Modeling**

Overview - This chapter introduces parametric subnetworks: how to create and use them in hierarchical designs. Beginning with a device model, the lowest level subnetwork will also contain packaging parasitics to better model the device behavior. Also, a test template will be used to simulate curve tracer responses from which a bias network can be computed, built, and checked. The circuit in this lab exercise will be the foundation of the amplifier that will be used for the other lab exercises in this course.

### **OBJECTIVES**

- Model a generic BJT with parasitics and save it as a sub circuit.
- Set up and run numerous DC simulations to determine performance.
- Calculate bias resistor values in the data display.
- Build a biased network based on the DC simulations.
- Test the biased network.



## **Table of Contents**

1. Create a new project: amp_1900. ....	3
2. Set up a generic BJT symbol and model card. ....	3
3. Add parasitics and connectors to the circuit.....	6
4. View the default symbol. ....	7
5. Define the symbol and artwork in Design Parameters. ....	7
6. Use a curve tracer template to test the bjt_pkg sub-circuit. ....	9
7. Modify the template Parameter Sweep. ....	11
8. Simulate at beta=100 and 160. ....	11
9. Open a new design and view the project in the Main window.....	12
10. Set up and simulate the dc bias parameter sweep.....	14
11. Calculate bias values Rb and Rc for a grounded-emitter circuit. ....	15
12. Set up the biased network. ....	16
13. Simulate and annotate the DC solution. ....	17
14. OPTIONAL: Sweep Component Model Temperatures.....	18

## PROCEDURE

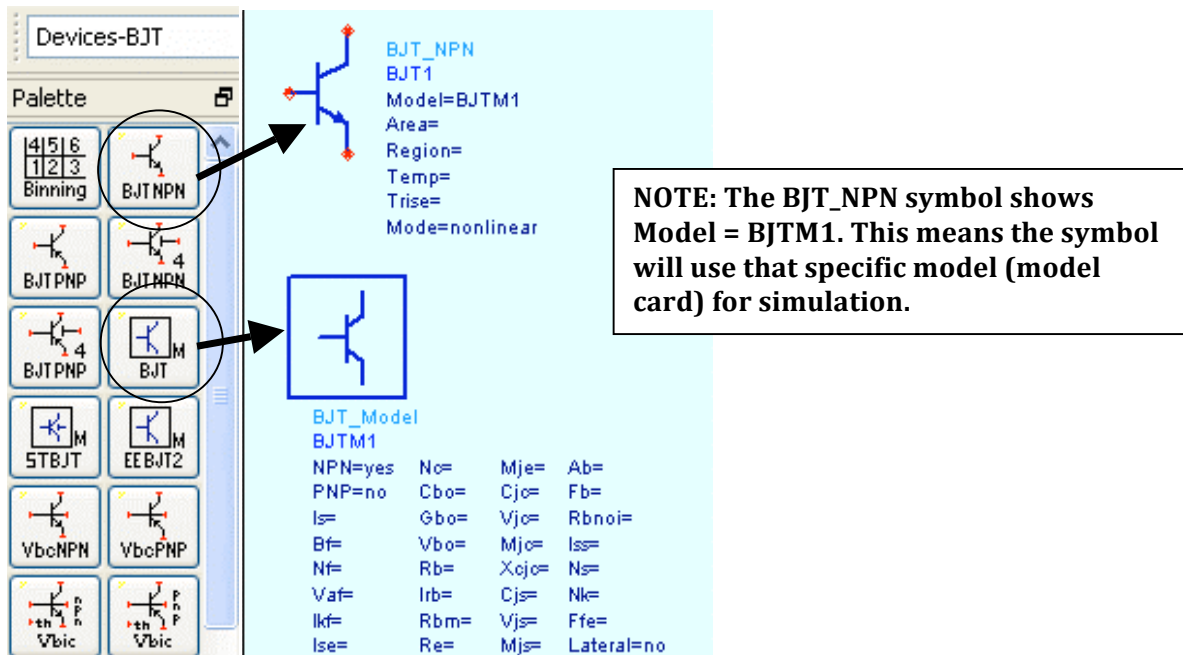
The circuit you build for this lab exercise will be used as the lower level sub-circuit for all of the amplifier labs to follow.

### 1. Create a new project: amp\_1900.

- a. If you have not already created this project, do it now. Then, in this new project, **amp\_1900**, open a new schematic window and save it with the name: **bjt\_pkg**. Also, set any desired Options > Preferences.

### 2. Set up a generic BJT symbol and model card.

- a. In the schematic window, select the palette: **Devices-BJT**. Select the **BJT\_NPN** device shown here and insert it onto the schematic.
- b. Insert the **BJT\_Model** (model card) shown here.

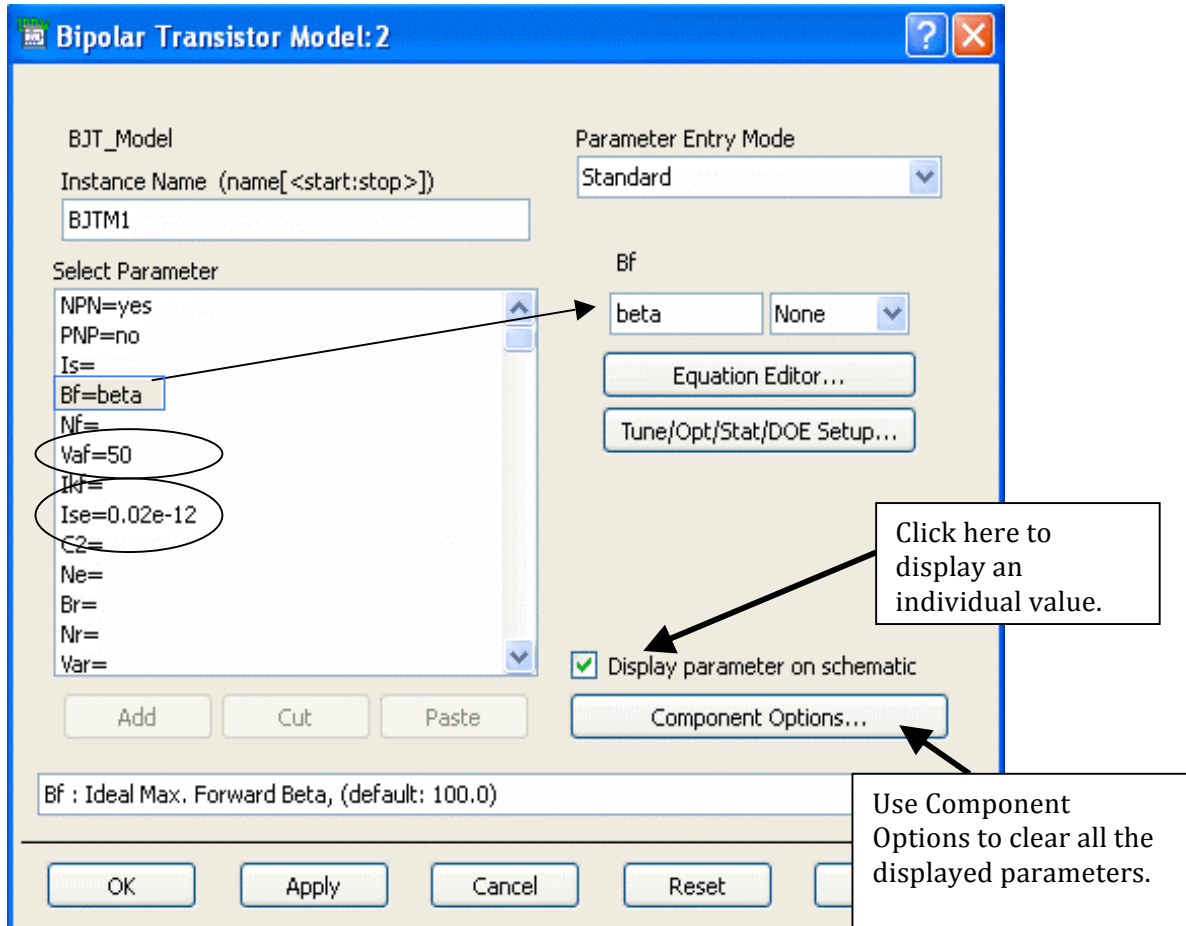


- c. Double click on the **BJT\_Model** card. When the dialog appears, click **Component Options** and in the next dialog, click **Clear All** for parameter visibility – then click **Apply**. This will remove the Gummel-Poon parameter list from the schematic. Keep this dialog open.

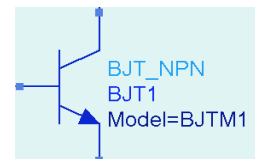
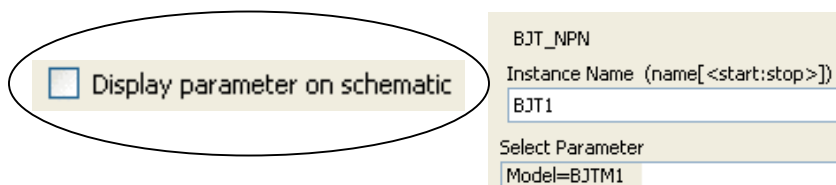
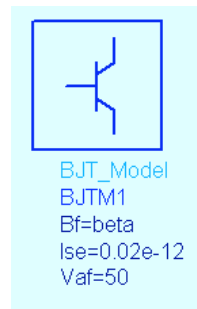
**NOTE on Binning:** You can insert multiple model cards and use the BinModel component to vary the model device variations. These variations can be parameters such as temperature, length or width. The Binning component allows you to create a matrix that references the desired models.

Lab 3: DC Simulations

- d. Next, in **BJT\_Model** dialog, select the **Bf** parameter and type in the word **beta** as shown here. Also, click the small box: **Display parameter on schematic** for **Bf** only and then click **Apply**. Beta is now a parameter of this circuit – later on you will tune it like a variable.



- e. Set **Vaf** (Forward Early Voltage) = **50** and display it.
- f. Set **Ise** (E-B leakage) = **0.02e-12**, and display it. Then close the dialog with **OK**. The device now has some more realistic parameters.
- g. For the **BJT\_NPN** device symbol, remove the unwanted display parameters (Area, Region, Temp and Mode) by unchecking the box. This box. This will make the schematic less crowded with text.

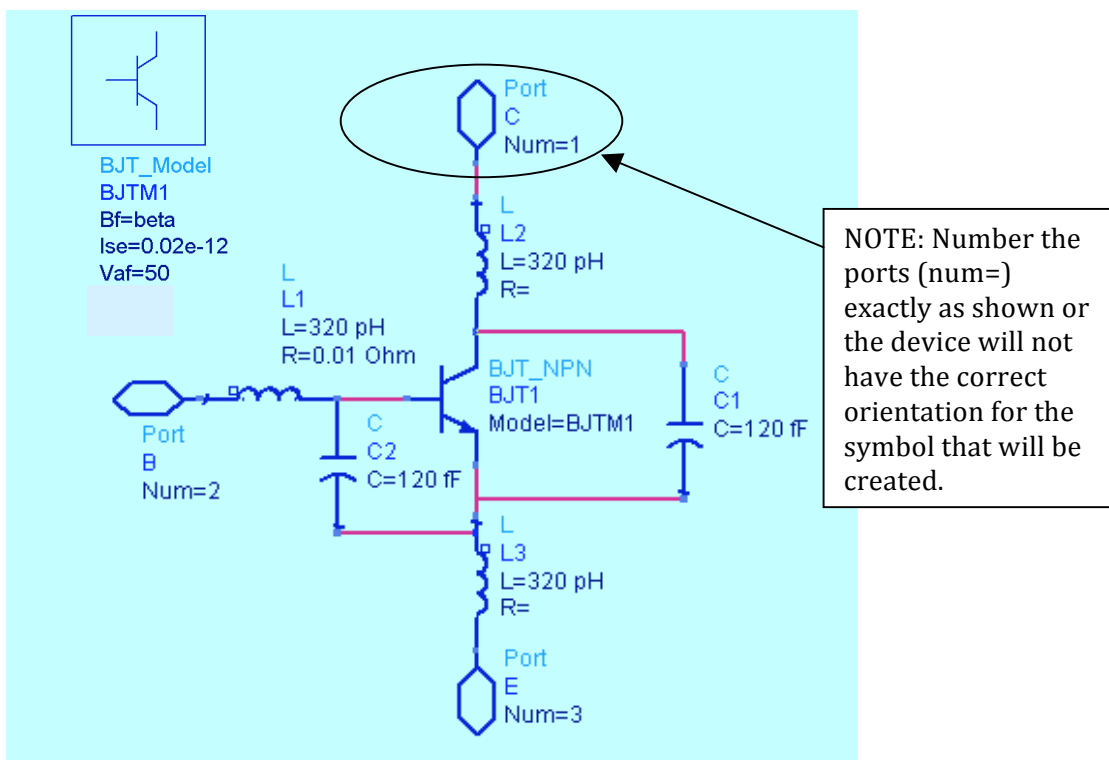
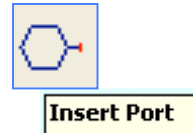


## Lab 3: DC Simulations

### 3. Add parasitics and connectors to the circuit.

The picture shown here is the completed sub-circuit with connectors and parasitics. Remember to use the **rotate icon** for orientation of the components as you insert them. Here are the steps:

- Insert lumped L and C components: Insert three lead inductors of **320 pH** each and two junction capacitors of **120 fF** each. Be sure to use the correct units (pico and femto) or your circuit will not have the correct response. Tip: type L or C in component history to get the components onto your cursor without using the palette.
- Add some resistance **R= 0.01 ohms** to the base lead inductor and display the desired component values as shown.
- Insert port connectors: Click the port connector icon (shown here) and **insert the connectors exactly in this order**: 1) collector, collector, 2) base, 3) emitter. You must do this so that the connectors have the exact same pin configuration as the ADS BJT BJT symbol.
- Edit the port names as show here: change P1 to **C**, change P2 to **B**, and change P3 to **E**.
- Clean up the schematic: Position the components so that the schematic is organized – this is good practice. To move component text, press the **F5** key, **select the component**, and position the text using the cursor.

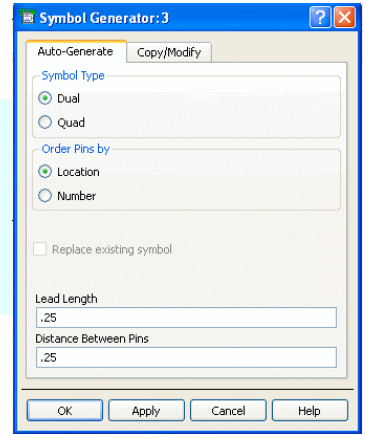
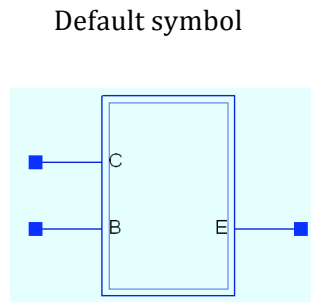


4. View the default symbol.

There are three ways to create a symbol for a circuit: 1) use the default symbol, 2) draw a symbol, or 3) use a built-in symbol. For this lab you will use a built-in BJT symbol. The following steps show how to do this:

- a. To see the default symbol, click: **View > Create/Edit Schematic Symbol**. When the Symbol Generator dialog appears, click **OK** and the default symbol will appear.

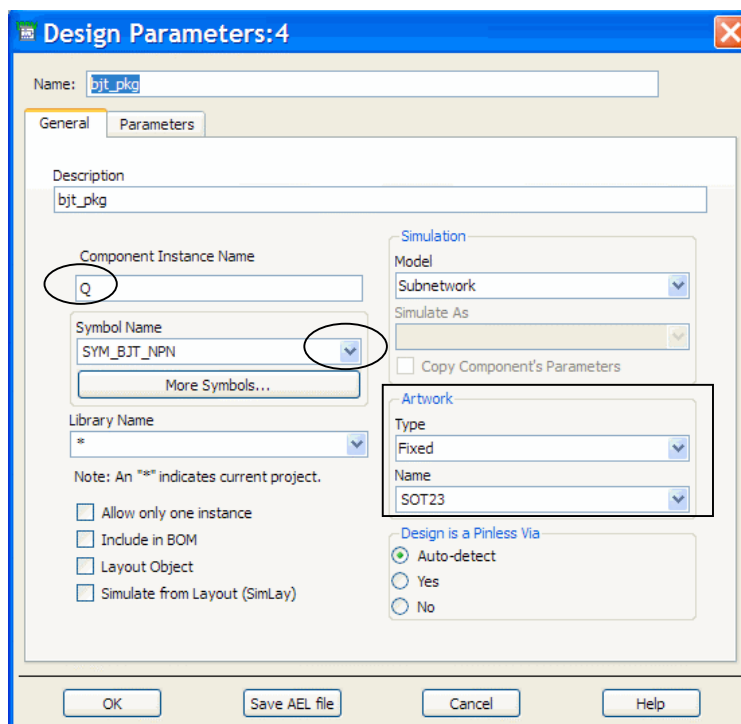
- b. Next, a box or rectangle with three three ports is generated: default symbol. However, delete this symbol using the commands: **Select > Select All**. Then click the the **trash can** icon or delete key.



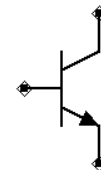
- c. Return to the schematic - click: **View** click: **View > Create/Edit Schematic**.

5. Define the symbol and artwork in Design Parameters.

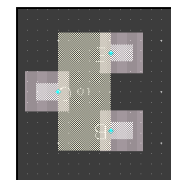
- a. Click **File> Design Parameters** and the dialog appears
- b. In the **General tab**, make these changes: 1) change the Component Instance Name to **Q**, 2) change the Symbol Name to **SYM\_BJT\_NPN** by clicking the arrow and selecting it (this is the built-in symbol), 3) in the



Built-in symbol:  
SYM\_BJT\_NPN



SOT23 fixed artwork



### Lab 3: DC Simulations

Artwork field, select **Fixed** and **SOT23** as shown here.

- c. Click **Save AEL File** to write these changes but do not close this dialog yet because you still need to set other parameters.
- d. Go to the **Parameters tab**. In the Parameter Name area, type in **beta** and assign a default value of **100** by clicking the **Add** button. Be sure the box to **Display** the parameter is checked as shown here. Click the **OK** button to save the new definitions and dismiss the dialog.

The parameter *beta* is now recognized as a variable of this circuit. Its value can now be passed (assigned) from an upper level hierarchy when you use *bjt\_pkg* as a sub-circuit.

NOTES: multiplicity\_M  
You can define multiple components in parallel. You can also copy parameters from another device or file.

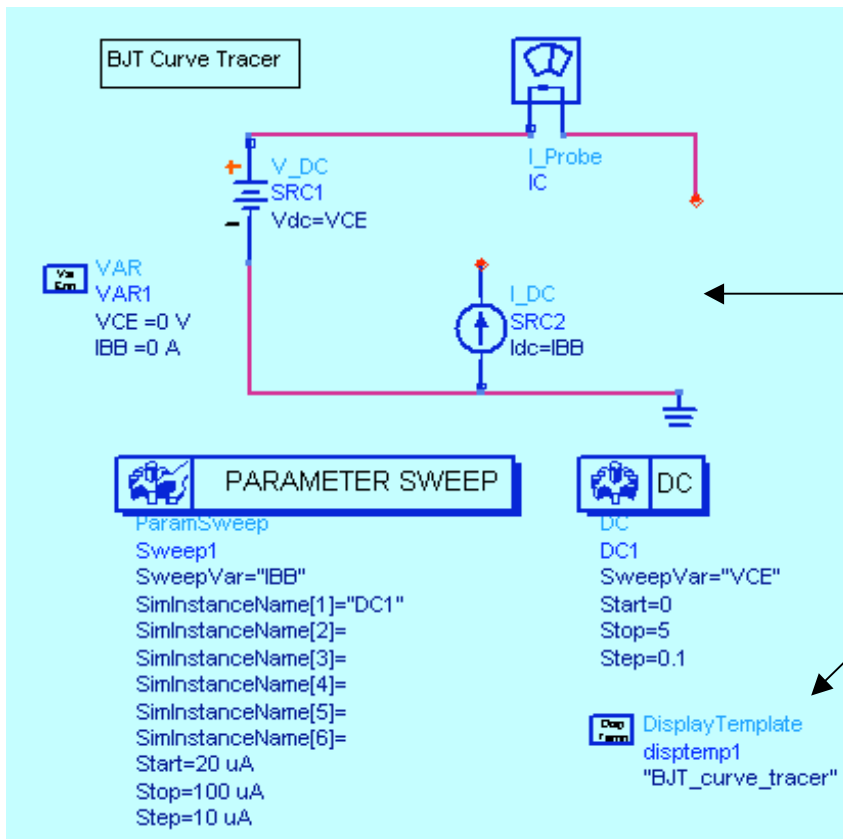
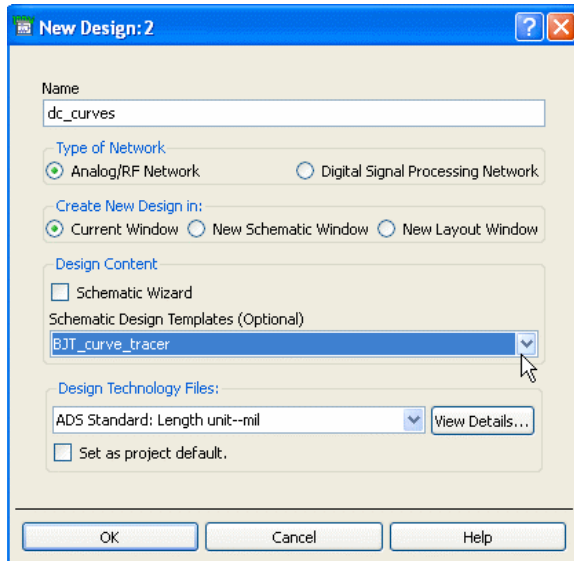
- e. In the schematic window, **Save** the design (click the icon shown here) to make sure all your work creating this sub-circuit will not be lost. In the next steps, you will see how the Design Parameters will be used.





6. Use a curve tracer template to test the bjt\_pkg sub-circuit.

- a. In the current schematic of the bjt\_pkg, click **File > New Design**. When the dialog appears, type in the name: **dc\_curves** and select the **BJT\_curve\_tracer** template as shown. Click **OK** and a new schematic will be created with the template, ready to insert bjt\_pkg.



This is where you will connect the device (bjt\_pkg) in the next few steps.

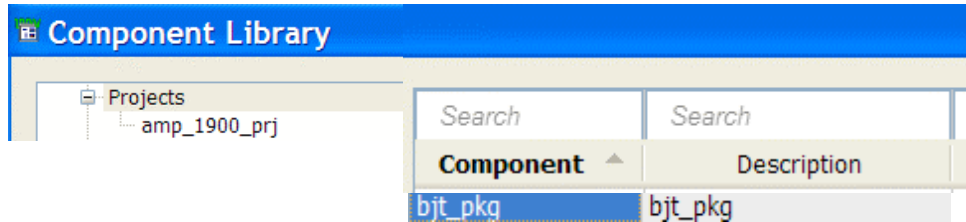
This is a data display template - it will automatically plot the curves.

### Lab 3: DC Simulations

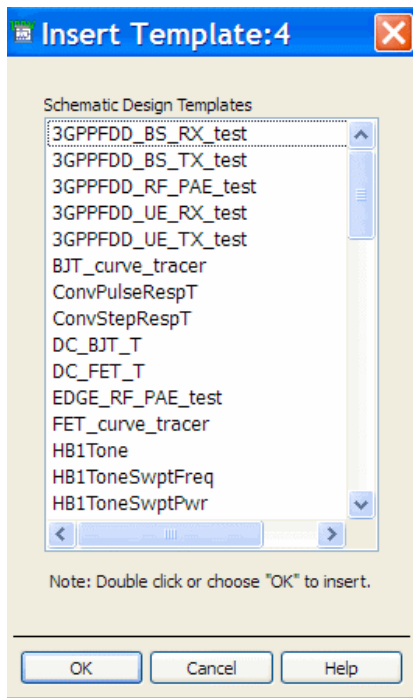
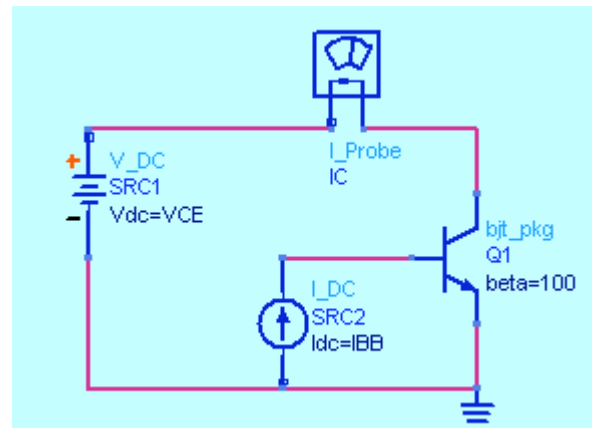
b. Save the design and then click the **Component Library** icon (shown here).



c. When the dialog opens, select **Projects > Amp 1900** and then click and drag the **bjt\_pkg** into the schematic and insert it as shown here. Every circuit that you build will be available in the project library as a sub-circuit.



d. Connect the **bjt\_pkg** component as shown. You may have to adjust the wires and text (F5) to make it look good. Also, you can now **close the library window** and save the dc\_curves design again - it is good practice to save designs often.



**NOTE on templates:** You can also insert templates using the schematic window command: **Insert > Template**. Many templates have pre-defined values, node names (wire labels), and variables. Therefore, you may have to make modifications to fit your circuit. Also, many of these templates have data display templates that automatically plot the data in a pre-defined format. These same data display templates are available in the data display window. In general, using templates is very efficient and timesaving if you know how to use ADS.

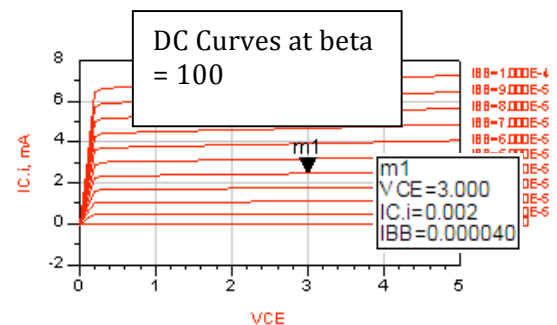
7. Modify the template Parameter Sweep.

- a. Change the Parameter Sweep **IBB** values to: 0 uA to 100 uA in 10 uA steps as shown here. Do not change the DC simulation controller default settings for sweeping VCE – they are OK. Notice that the VAR1 variables (VCE=0 and IBB=0) do not require modification because they are only required to initialize (declare) the variable for the simulator.

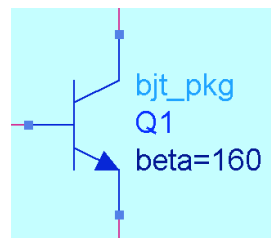
The screenshot shows the simulation controller configuration. On the left, the 'PARAMETER SWEEP' section is expanded to show 'Sweep1' with 'SweepVar="IBB"'. Below it, six 'SimInstanceName' fields are listed, all pointing to 'DC1'. The 'Start' is set to 0 uA, 'Stop' to 100 uA, and 'Step' to 10 uA. On the right, the 'DC' section shows 'DC1' with 'SweepVar="VCE"', 'Start=0', 'Stop=5', and 'Step=0.1'. A 'VarEqn' box contains 'VAR1', 'VCE = 0 V', and 'IBB = 0 A'. Three callout boxes provide context: 'Only one variable can be swept in the controller.' points to the DC SweepVar; 'Parameter sweep used for multiple variable sweeps. Note that DC1 is the name (SimInstance Name) of the simulation controller.' points to the DC1 instance names; and 'VarEqn is required to initialize' points to the VarEqn box.

8. Simulate at beta=100 and 160.

- a. **Simulate** (F7) with **Beta = 100**. After the simulation is finished, the data display will appear with the curve tracer results (data (data display template)). Try moving the marker marker and watch the updated values appear.
- b. **Simulate** again with **Beta = 160** by changing the value directly on the schematic. You should see the updated values.



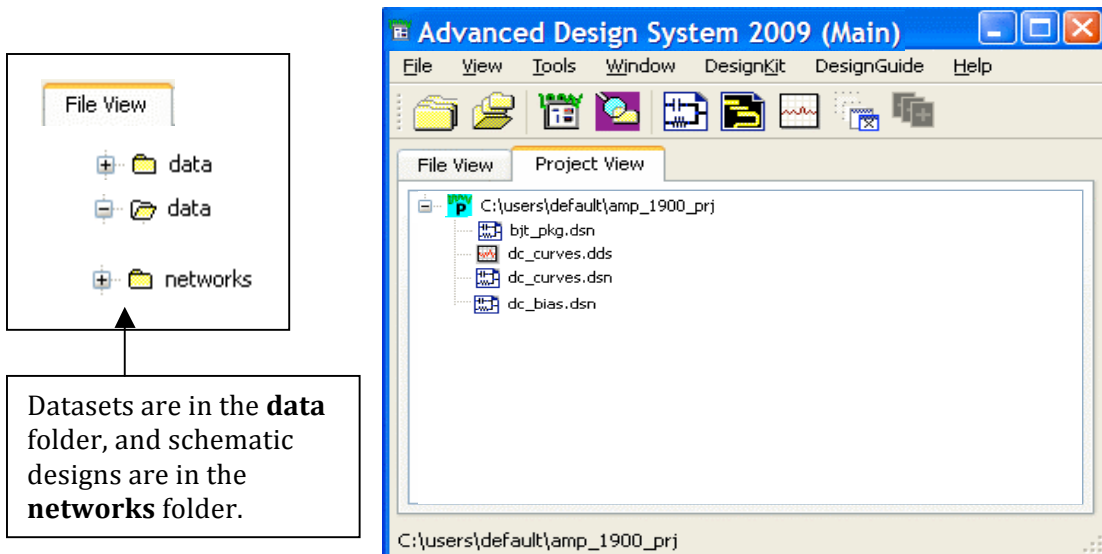
- c. Verify the values for beta = 160 and VCE = 3V, where IBB=40 uA and IC=3 mA with about 10mW of consumed power. If not, check the design.



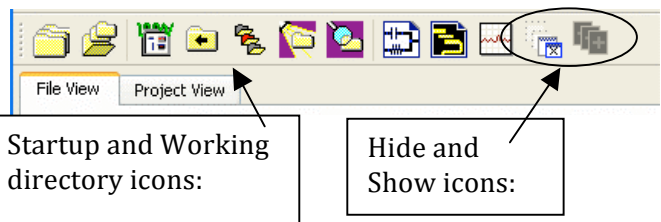
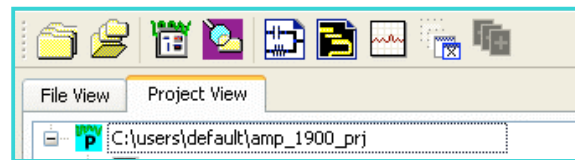
Move Marker m1 to update values below:		Device Power Consumption at m1 bias point, Watts
VCE		
3.000		0.010

9. Open a new design and view the project in the Main window

- a. Save the current schematic: **dc\_curves**. In the same window, create a new design (without a template) named: **dc\_bias**. Then save the design by clicking on the **Save Current Design** icon (shown here) so that it is written into the ADS database.
- b. Also, save the Data Display.
- c. Go to the ADS Main window and verify that your project has three designs and the data display as shown in the Project View.
- d. Also, look at the File View to see the Data directory where the simulation datasets are written. Click on the + to expand the directory folder and – to close it. And then look in the networks directory folder – this is where the schematic designs (.dsn) are written.



- e. Click on the File View and then the the Project View tabs - notice that the that the toolbar icons change.
- f. In the File View tab mode, click the the icons for Startup and Working directory to directory to see how they work. You work. You can always browse the the files of any directory and come come back to your working directory, such as amp\_1900. amp\_1900.



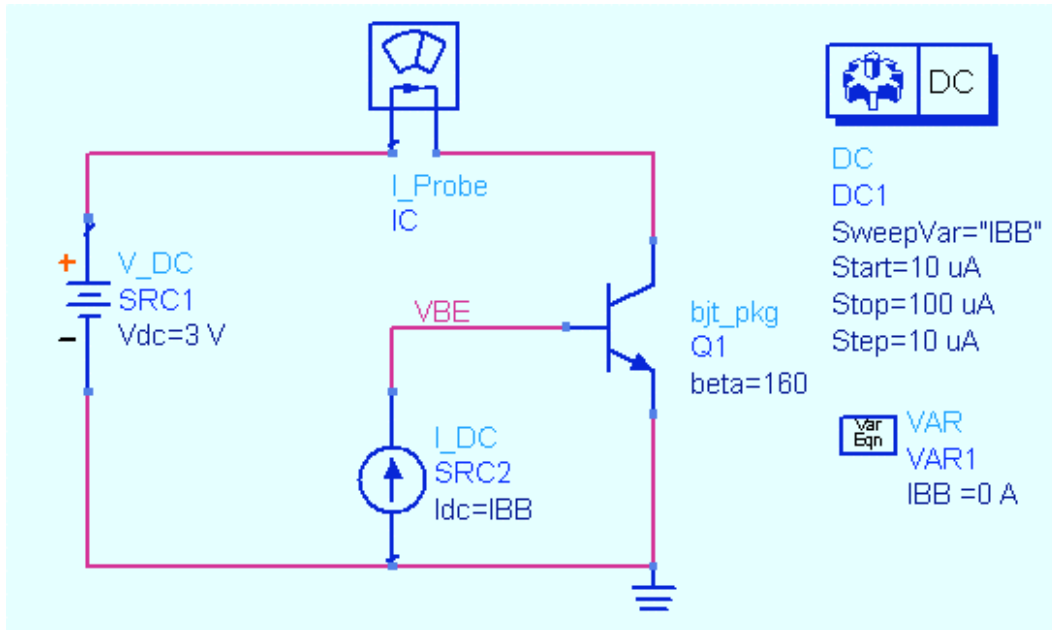
- g. Also try the Hide and Show icons.



**10. Set up and simulate the dc bias parameter sweep.**

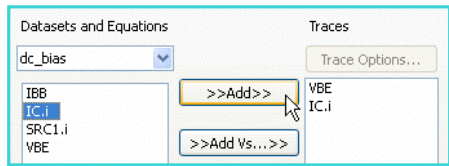
**NOTE on parameter sweeps:** If only one variable is swept, you can use the Simulation controller (Sweep tab). However, if more than one parameter is swept, a Parameter sweep component is required as in the templates you have just used. In general, all simulation controllers allow you to sweep only one parameter (variable).

BUILD the circuit without a template - the steps follow:



- Insert the **bjt\_pkg** using library icon or the component history. Now **push into** the bjt\_pkg and click **File > Design Parameters**. Reset the beta parameter default to 160, pop out and **delete** the bjt\_pkg and **reinsert** it – beta is now 160 whenever you use the modeled circuit.
- From the Probe components palette or component history, insert an **I\_Probe** and rename it **IC** instead of I\_Probe1 as shown here. You can also get the current from the simulation by setting the DC controller Output tab for Pin Currents but the probe is easy to use in this case.
- From the **Sources-Frequency domain palette** or using component history, insert a dc supply and current source and set their values as shown: **Vdc = 3 V** and **Idc = IBB** as shown here.
- Wire the components together and add the ground (ground icon).
- Insert a **DC simulation controller (DC)**. Edit (double click) the controller and go to the **Sweep tab** and assign: **IBB: 10 uA to 100 uA in 10 uA steps** (be sure to use uA units because the sweep tab is generic). Then go to the **Display tab** and check the settings to be displayed as shown. Then click **Apply** and **OK**.

- f. Insert a **VAR** (click icon) variable equation. Use the cursor on the screen to set **IBB=0 A** to initialize (declare) the variable to be swept.
- g. Insert a wire label **VBE** at the base. The voltages at that node will appear in the dataset for use in calculating bias resistor values.
- h. **Simulate** and **plot the data**. When the data display opens, insert a list of **VBE** and **IC.i** only. Because you swept IBB to get these values, IBB will automatically be included.



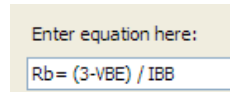
IBB	IC.i	VBE
1.000E-5	599.8uA	754.8mV
2.000E-5	1.430mA	777.1mV
3.000E-5	2.349mA	789.9mV
4.000E-5	3.325mA	798.8mV
5.000E-5	4.341mA	805.7mV
6.000E-5	5.389mA	811.3mV

**NOTE on results:** As you can see, with 3 volts across the device, 40 uA of base current results in about 799mV across the base-emitter junction, with about 3.3 mA of collector current. If you want, draw a box around the values at 40 uA IBB.

- i. Save the design and data display.

### 11. Calculate bias values Rb and Rc for a grounded-emitter circuit.

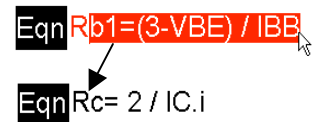
- a. In the data display, insert an equation and type: **Rb = (3 - VBE) / IBB**.



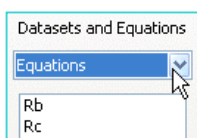
**Eqn**  $Rb = (3 - VBE) / IBB$

- b. Select the Rb Eqn and use the keyboard **Ctrl C** and **Ctrl V** to copy/paste it – it will become Rb1.

- c. Highlight the Rb1 equation as shown and change it to become Rc: **Rc = 2 / IC.i**. The total DC supply will be 5 volts. Therefore, with 3 volts VCE, 2 volts remain for the collector resistor.



- d. Insert a new **List** and scroll down to the **Equations** menu (shown here), and add **Rb** and **Rc**. Then edit both column headings on the list with a bracketed **[3]** as shown. This references the 40uA IBB using its index value [3]. Index values begin at zero: 0, 1, 2, 3, etc. You can also use Plot Options to add a label to the list as shown:



IBB	Rb [3]	Rc
1.000E-5	224518.376	3334.23
2.000E-5	111142.843	1398.88
3.000E-5	73669.708	851.34
4.000E-5	55029.037	601.50
5.000E-5	43886.113	460.68
6.000E-5	36479.142	371.12

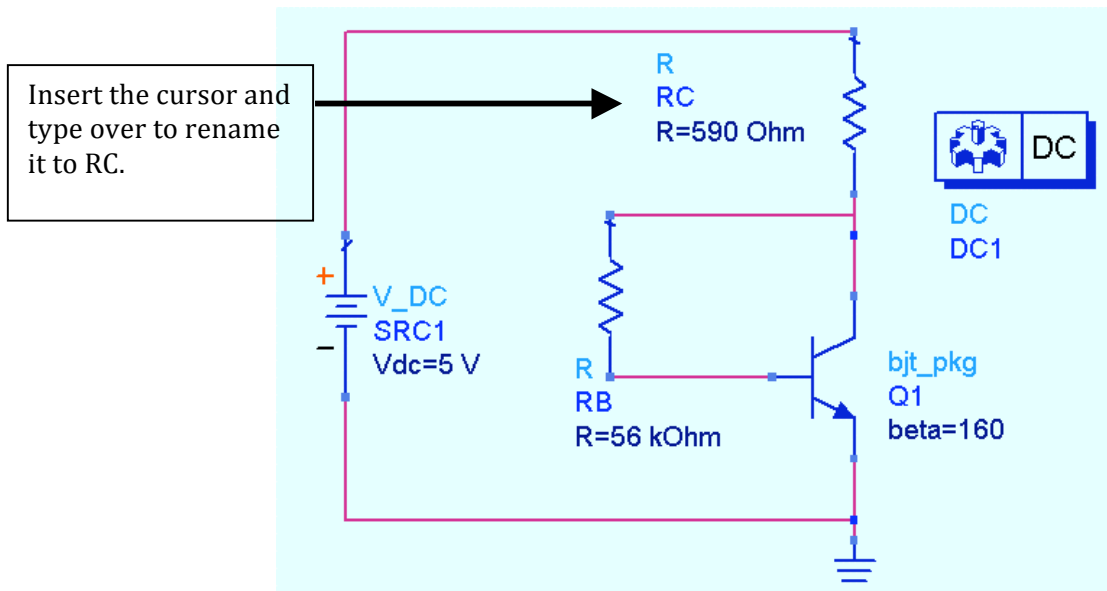
Bias Resistor Values @ 40 uA IBB

Rb[3]	Rc[3]
55029.037	601.500

## 12. Set up the biased network.

Now that you have the calculated bias resistor values, you can test the bias network.

- Save** the design (dc\_bias) with a new name: **dc\_net**. By this time, you should know how to do this (File > Save Design As). Also, save and close the dc\_bias data display.
- Modify the design as shown. Begin by deleting the current source IBB, the I\_Probe, and the Var Eqn.
- Go to the **Lumped Components** palette or use component history, **R**, to insert **resistors** for the base (**56 kOhm**) and collector (**590 Ohm**) resistors as shown. You may need to use **rotate** to insert it correctly.
- Change the instance R names to **RC** and **RB** as shown.



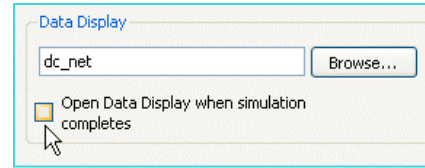
**NOTE on components with artwork:** Later on (after the last lab), you can easily and quickly change to lumped components with artwork by changing the component name – for example, change R to R\_Pad1, C to C\_Pad1, L to L\_Pad1, etc. Then you can create a layout of the schematic. For now, use lumped without artwork.

- Set the V\_DC supply: **Vdc = 5 V**. Wire the circuit and organize it.
- Delete the DC simulation controller and put a new one in its place – this is faster and more efficient than removing the sweep settings. Because there is no sweep, you do not have to set anything to check DC values.

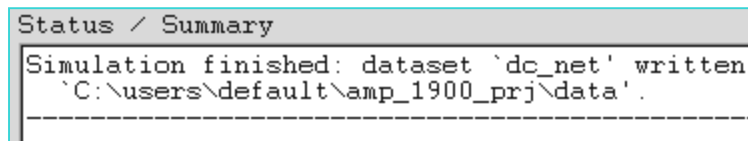


13. Simulate and annotate the DC solution.

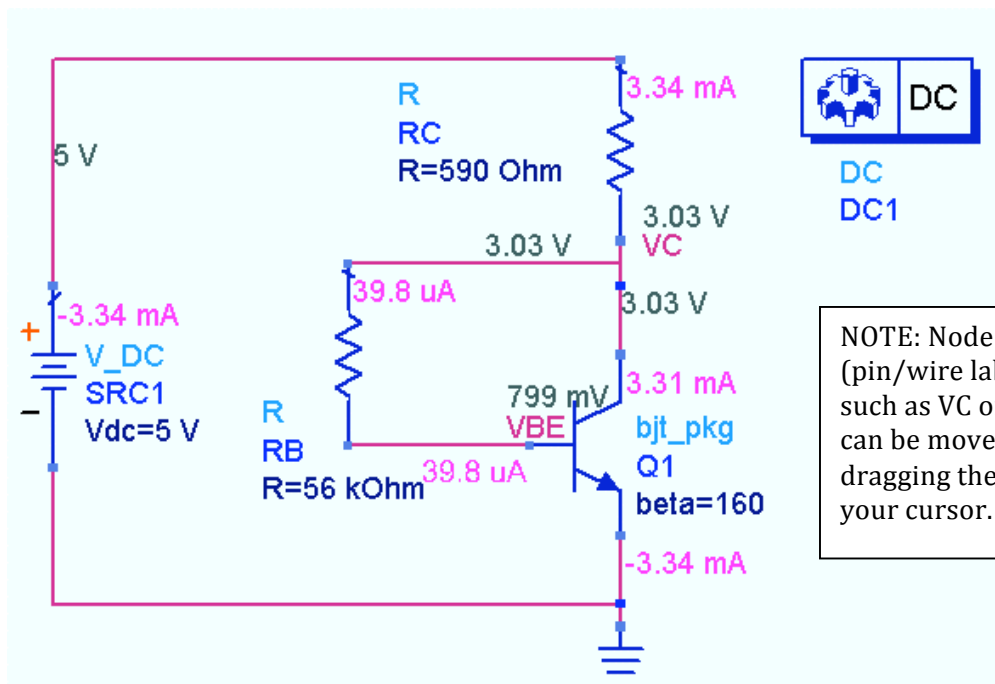
- a. Use the **Simulate > Simulation Setup** (or Hot Key Hot Key “S” if you have set it) and **uncheck** the box to open data display after simulation.



- b. Click **Simulate > Simulate**, or on a PC try the **F7** keyboard key, to run the simulation. The dataset name will be the same as the schematic – this is the default. You can verify this by reading the status window:



- c. Annotate the current and voltage by clicking on the menu command: **Simulate > Annotate DC Solution**. If necessary, move components or component text (F5 key) to clearly see the values of voltage and current. Be sure that you have the same values shown here. If not, check your work, including the sub-circuit.

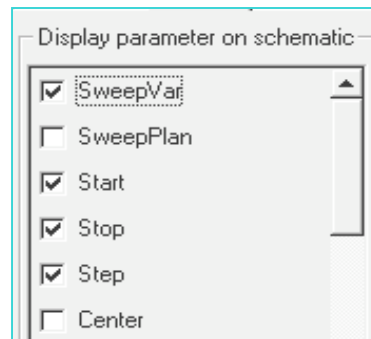
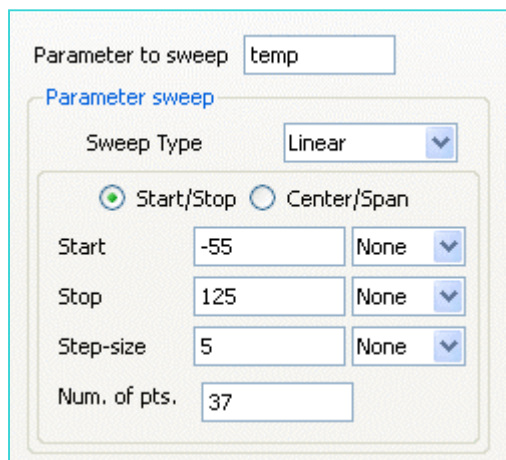


### Lab 3: DC Simulations

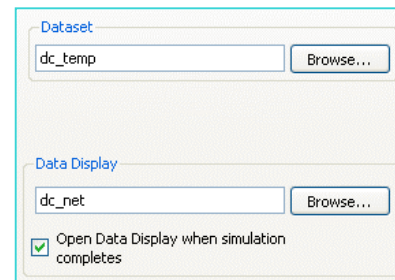
- d. Clear the annotation, click: **Simulate > Clear DC Annotation** and then **Save** all you work. Close all windows if nor doing the optional steps that follow.

#### 14. OPTIONAL: Sweep Component Model Temperatures

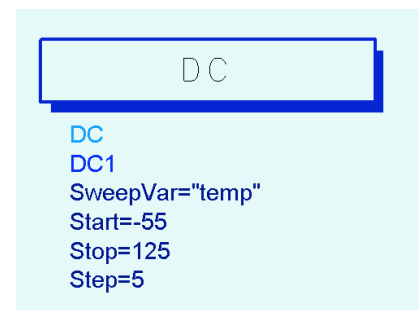
- a. Edit the **DC controller** – select it and click the edit icon.
- b. In the **Sweep tab**, enter the ADS global variable **temp** (default is Celsius) and enter the sweep range: **-55 to 125** with step size = **5**. Also, in the **Display tab**, click the boxes to display the annotation on the controller – click **Apply** to see it and **OK** to dismiss the dialog.



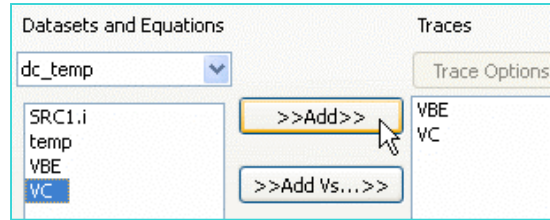
- c. Insert **VC** and **VBE** node / pin labels.
- d. Set the simulation dataset name to **dc\_temp**, and check the box to open the data display dc\_net. Click **Apply** and then **Simulate**.



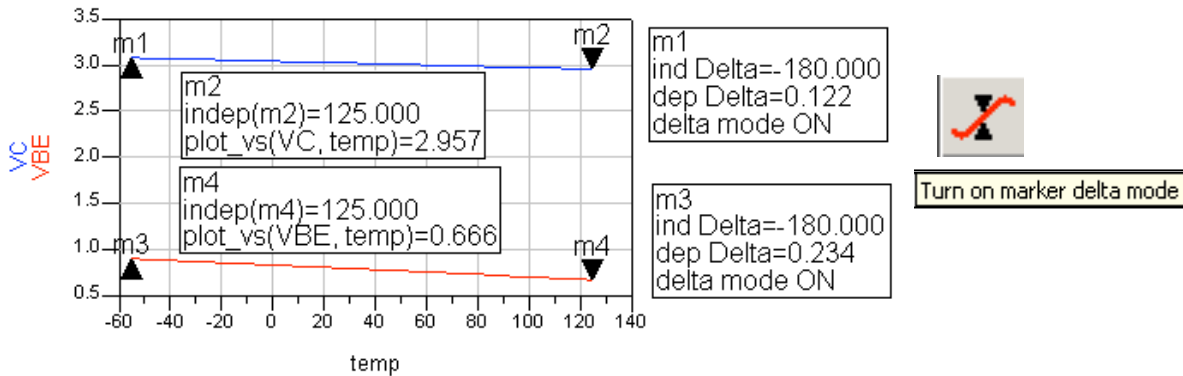
NOTE: The Trise parameter (temperature rise over over ambient) can be specified on individual components.



- e. Plot **VC** and **VBE** in a rectangular rectangular plot. These will be vs. be vs. temp which is the swept variable. You could also use the Add the Add Vs function but it is not not necessary when there is one one swept variable.



- f. Put two markers on each trace in **delta mode** to see the change in voltage as temperature changes. The plot should look like the one shown here: collector voltage decreases at almost one-half the rate of VBE as the temperature increases. You can use this temperature sweep method (sweeping the global variable “temp”) for any ADS simulation.



EXTRA EXERCISES:

1. Plot current (probe: IC.i) vs. temperature using a probe. Or, try setting up a passed parameter for temperature (Temp = 25 in the options controller). The Options controller is in every simulation palette and can be used to set convergence for DC and constant simulation temperature.
2. Replace the Gummel-Poon model card with another model (Mextram) and resimulate. Afterward, compare the results.
3. Insert the template **SP\_NWA\_T** which generates S-parameters for a bjt at all the specified bias points. Try using this with your bjt\_pkg.

## Lab 3: DC Simulations