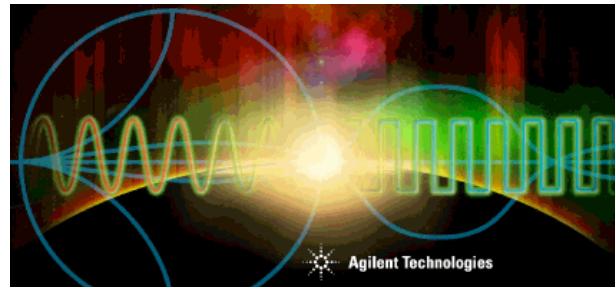# Topic 3:

# DC Simulations and Sub-circuit Modeling

# DC Simulation

You get steady-state DC voltages and currents according to Ohm's Law: V= IR

• Capacitors = treated as ideal open circuits

• Inductors = treated as ideal short circuits

• Topology check: dc path to ground (may add T ohms)

• DC convergence occurs when 2 conditions are met:
    Voltage change at each iteration is zero and
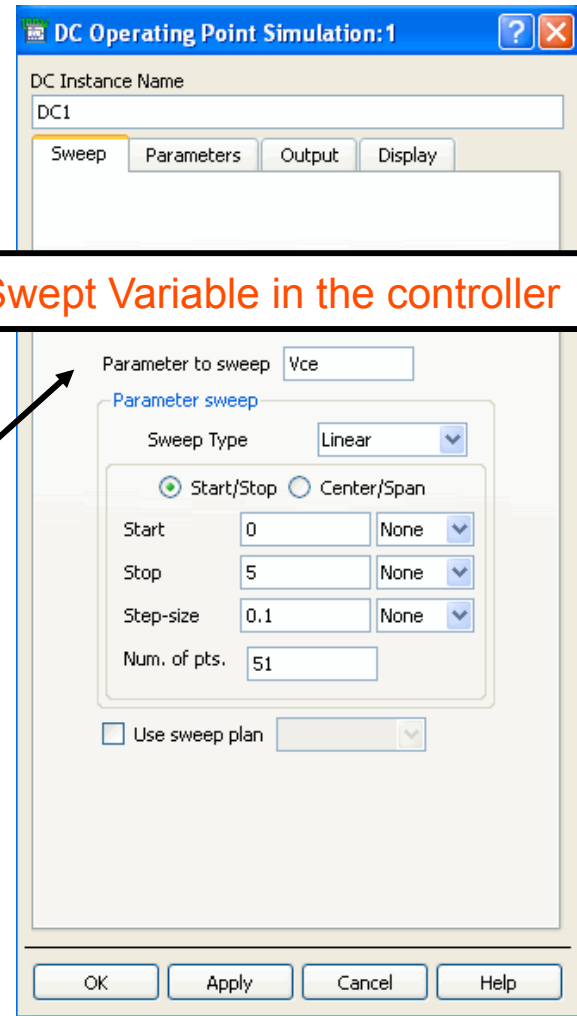    Kirchoff's Law is satisfied: sum of node current = 0.

# DC simulation controller
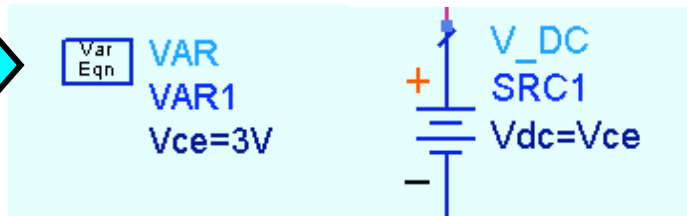
## Palette and editor (dialog box)
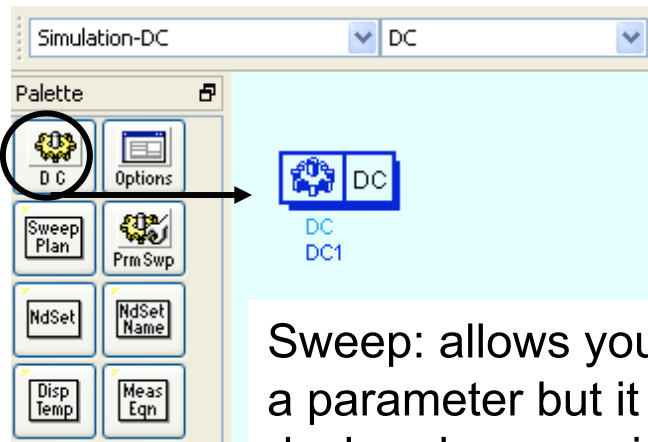


Sweep: allows you to sweep a parameter but it must be declared as a variable. Note this dialog entry automatically puts quotes on the controller (screen) display.

**VAR**

(variable equation)



Swept Variable in the controller

More on DC...

# DC Parameters and Device Values

Values (detailed or brief) are available after simulation on schematic:

**Advanced**: rarely used - and only for convergence issues: increase Max Delta V or iterations.

Brief

Detailed

Back Annotation...

# Schematic Annotation of DC values
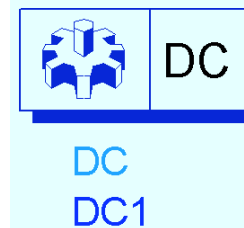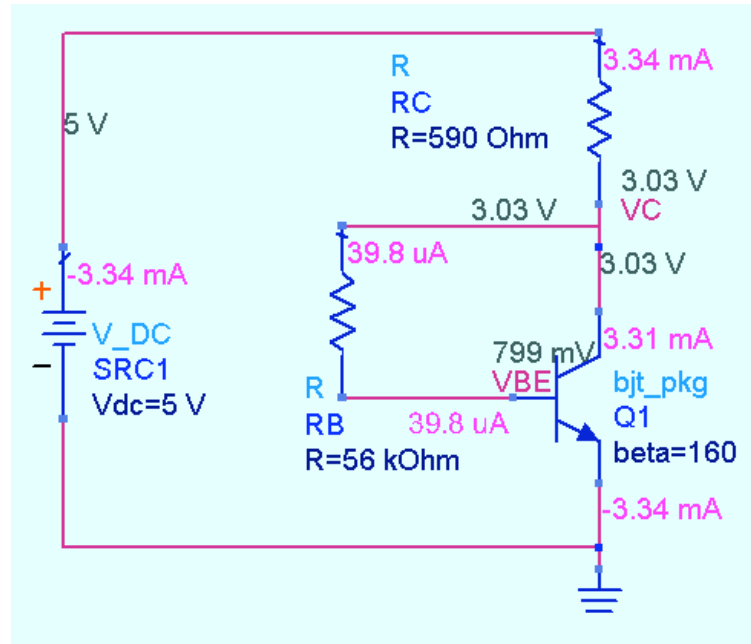
Immediately after DC simulation, click: **Simulate > Annotate DC Solution**.
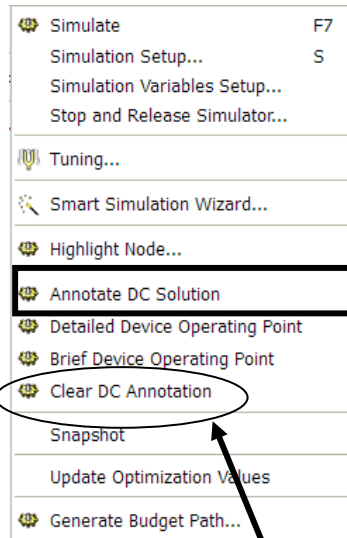
**Simulate >**

No controller settings necessary!



DC
DC1

Minus sign used for current flowing out of a connection. Otherwise, current flows into a connection or device.

DC Simulation Controller is required in all simulations if you want DC annotation.

Clear it here

**Next, named nodes...**

# Wire/Pin Labels (node names) in Schematic

To label a node, use the icon: 

- Type in the name, point and click.
- **You get node voltage in the <u>dataset</u>.**
- Use these in equations: dBm (Vout).
- Connect two pins by name - without a wire.
- Move and edit the label names (attributes).
- You can also create busses.
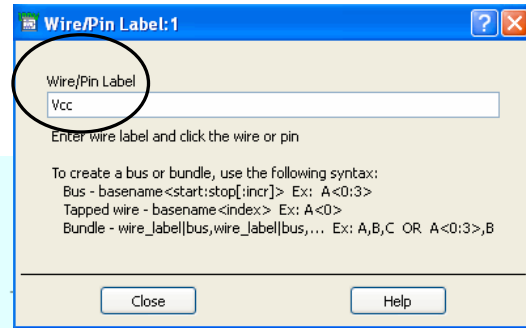
To remove a label, use the icon:

With a blank (no name), click on the node.

Or  Edit > Wire/Pin Label >
        Remove Wire/Pin Label
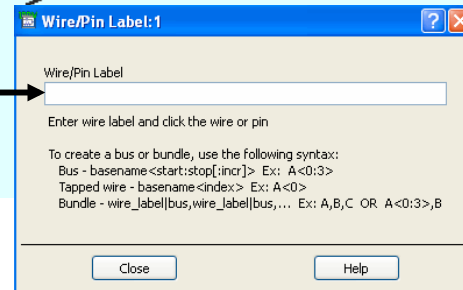
To edit the label, double click it,

Or use the command: Edit > Wire/Pin Label Attributes

For busses, see:
**examples/Tutorial/wire_bus_prj**
It is a documented example.
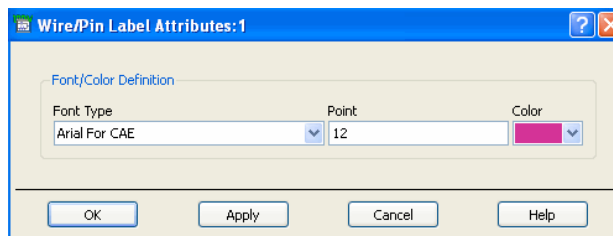
Next, VARS...

# Variable Equations: VAR

The **VAR** is a declaration (initialization).

Click: 
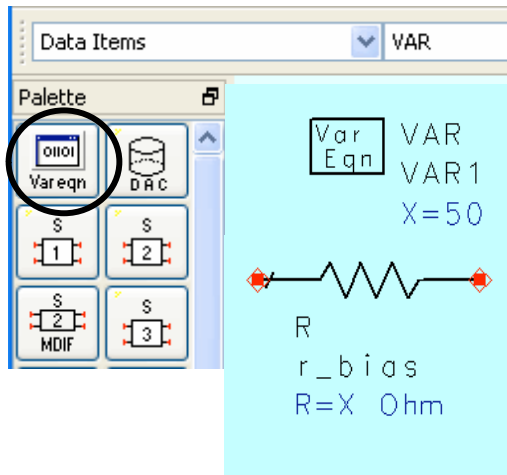


**TIP: Add dummy (X,Y,Z) variables and then edit them on-screen.**
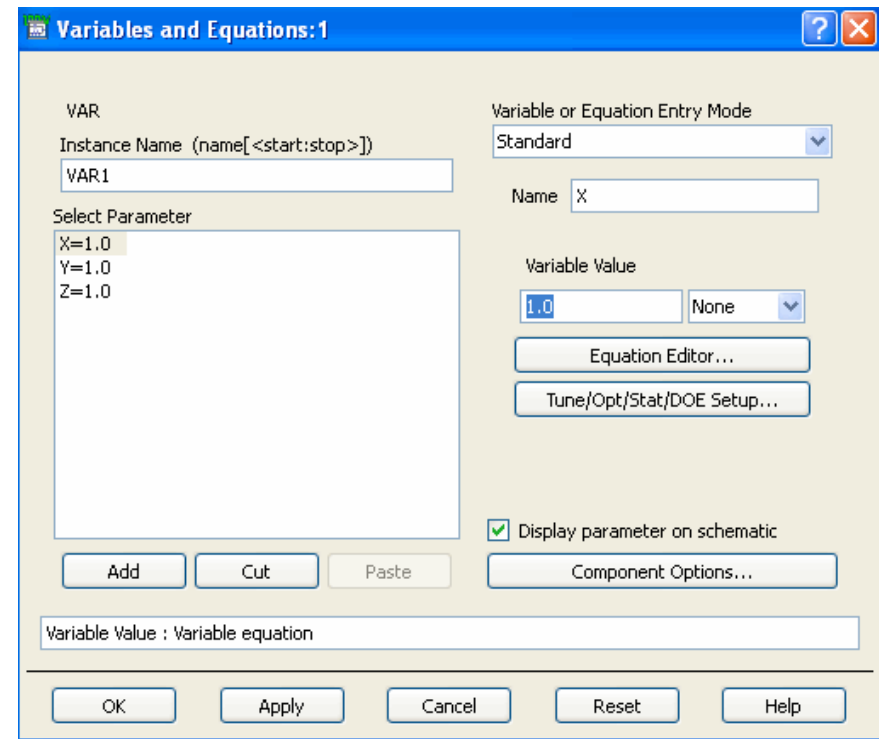
**Component parameters can be assigned to a variable: VAR.**

**VARS can be used with optimization, parameter sweeps, and other applications!**

Next, models…

# ADS and Analog Models

## In ADS, devices are modeled in various ways:

pf_mwt_MWT_671HP_19931015
A1

**Vendor library parts**: these are ready for simulation if you have the LICENSE for the library.

Specify the file and specify the desired symbol.

**Tools**

HSPICE Compatibility Component Wizard...
Netlist Export
Spice Model Generator
User-Compiled Model

- ⊞ HF Diode Library
- ⊟ Measurement Based SMT Pas...
  - muRata Manufacturing Ca...
  - muRata Manufacturing Ind...
- ⊟ Microwave Transistor Library
  - chip BJTs (No Layout)
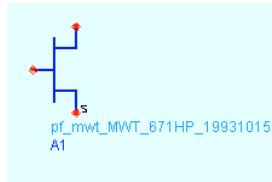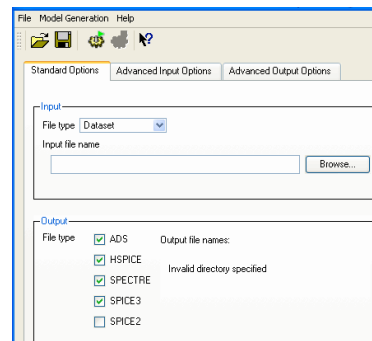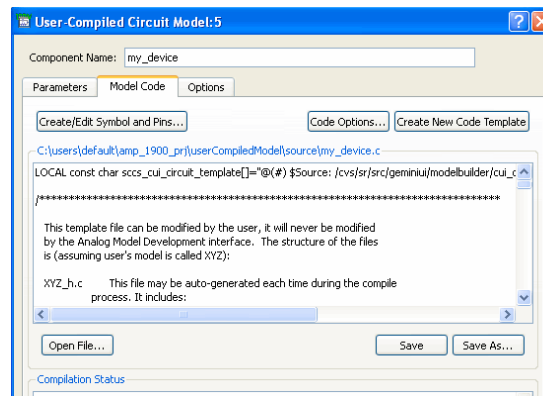  - chip GaAs FETs (No Layout)
  - chip HEMTs (No Layout) Pa...
- ⊟ RF Passive SMT Library
  - SMT Capacitor
  - SMT Inductor
  - SMT Resistor
- ⊟ RF Transistor Library
  - Packaged BJTs
  - Packaged GaAs FETs
  - Packaged Power MOSFETs
- ⊟ S Parameter Library (No Layout)
  - Agilent Technologies I (No ...
  - Agilent Technologies II (No...
  - Alpha Industries (No Layout)
  - Fujitsu (No Layout)
  - Harris (No Layout)
  - Litton (No Layout)
  - Mitsubishi (No Layout)
  - Motorola (No Layout)
  - Mwt (No Layout)
  - NEC (No Layout)
  - Phillips (No Layout)
  - Raytheon (No Layout)
  - Siemens (No Layout)
  - Sony (No Layout)
  - Toshiba (No Layout)
- ⊞ System Library

Download Libraries...

File Model Generation Help

Standard Options | Advanced Input Options | Advanced Output Options

Input
File type  Dataset
Input file name
Browse...

Output
File type  ☑ ADS       Output file names:
          ☑ HSPICE
          ☑ SPECTRE    Invalid directory specified
          ☑ SPICE3
          ☐ SPICE2

**Spice Model Generator:** Output data from ADS simulations as lumped equivalent spice models.

User-Compiled Circuit Model:5

Component Name: my_device

Parameters | Model Code | Options

Create/Edit Symbol and Pins...    Code Options...  Create New Code Template

C:\users\default\amp_1900_prj\userCompiledModel\source\my_device.c

LOCAL const char sccs_cui_circuit_template[]="@(#) $Source: /cvs/sr/src/geminiui/modelbuilder/cui_c

/*********************************************************************

This template file can be modified by the user, it will never be modified
by the Analog Model Development interface. The structure of the files
is (assuming user's model is called XYZ):

XYZ_h.c        This file may be auto-generated each time during the compile
               process. It includes:

Open File...                          Save    Save As...

Compilation Status

**User compiled models**: you write the code and use the ADS interface from schematic (Tools > User Compiled Model) to compile the model in ADS for simulation.
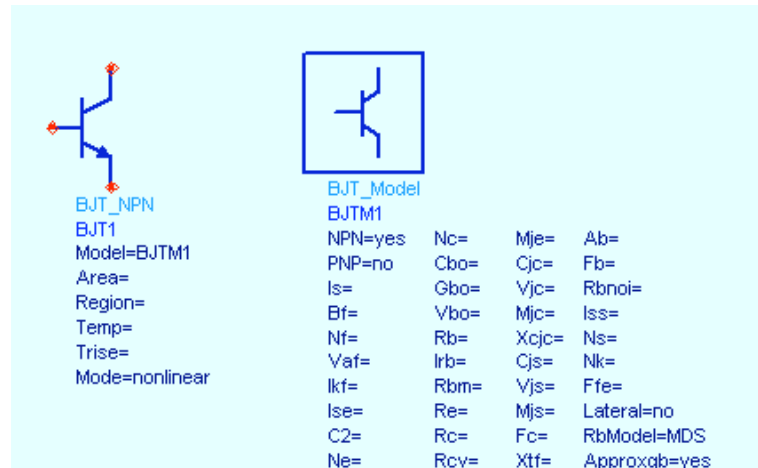
More models...

# more on Analog Models…

**Model Cards**: use a built-in symbol and model card which lists all the parameters that you can modify.  This example is a BJT (Gummel-Poon) model.

You will use this method in the lab exercise!



**Verilog-A**: ADS supports Verilog-A  (LRM version 2.2 compliant)

• Place your Verilog-A code (text file) in: PROJECT_DIR/veriloga for the current project or in $HOME/hpeesof/veriloga for all projects.

• Add a symbol and bitmap (similar to user compiled model).

• The Verilog-A device will have all the functionality of other ADS models.

• For more information see the examples in:

    `$HPEESOF_DIR/examples/Verilog-A/Tutorial_prj`

Next, wiring...

# Wiring and Moving components

- Use the wire icon or connect pins = wired
- Point and click to snap to grid
- Drag a wired component and it stays wired.
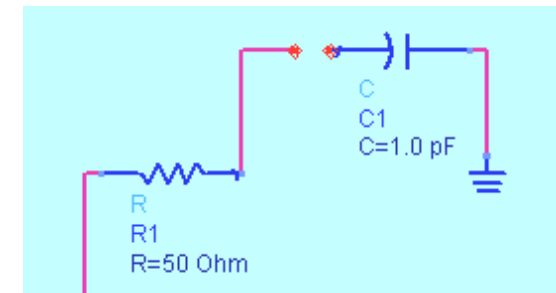- Wire color can be changed: Options > Layers
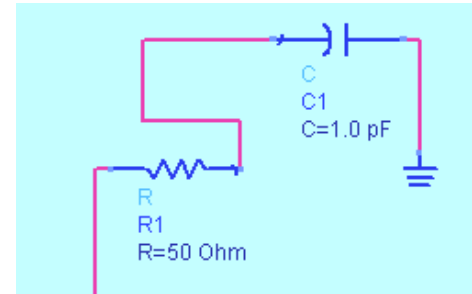
**Options**

| | | |
|---|---|---|
| | Snap Enabled | Ctrl+E |
| | Pin Snap | |
| | Vertex Snap | |
| | Reroute entire wire attached to moved component | |
| | Route around component text | |
| | Route around component symbol | |

**Red pin = not connected!**

**Activate/Deactivate (short) toggle icons:**

**Edit > Move**

**Mirror Icons:**
**X and Y mirrors**

**Rotation icon:**
 90 degrees default
(change in preferences)

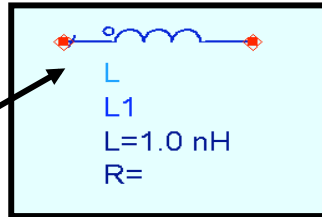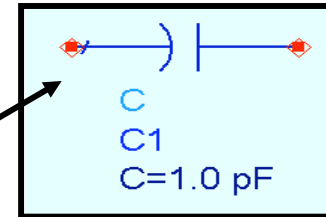| | | |
|---|---|---|
| | Move Using Reference | Ctrl+M |
| | Move Edge | |
| | Move Relative... | |
| | Move Disconnect | |
| | Move To Layer... | |
| | Move Wire Endpoint | Ctrl+Shift+M |
| | Move Wire/Pin Label | |
| | Move Component Text | F5 |

Useful ⟶
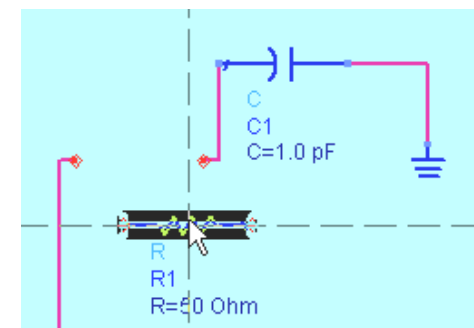
Next, symbols, names...

# Symbols, units, names, case sensitivity
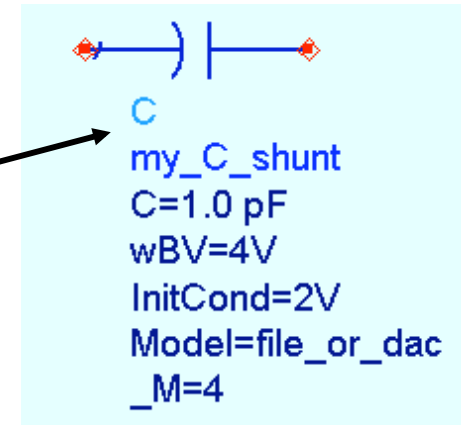


Circle for mutual inductance:

Slash for pin# 1 (layout):



## Example of on-screen control:

**C**   (component type): changes the component

**C1**  (instance name): rename it: c_shunt

**C=**  (parameter): a number (unit) or valid variable.



## Case Sensitivity:

**UNIX** is always case sensitive
**PC** is usually case sensitive (Linux)

• **Exception: Windows - insert R - after the first insert, it will recognize either r or R.**
• **But m=milli, M=mega, V=volts, and VARS are case sensitive all of the time!**

QUIZ: Is this valid?

C
coupling_c
C = x

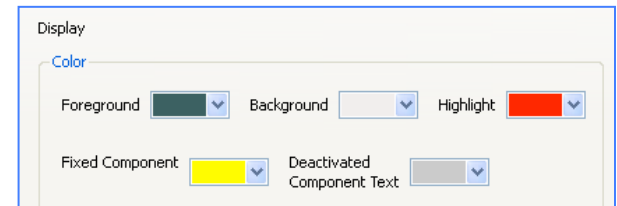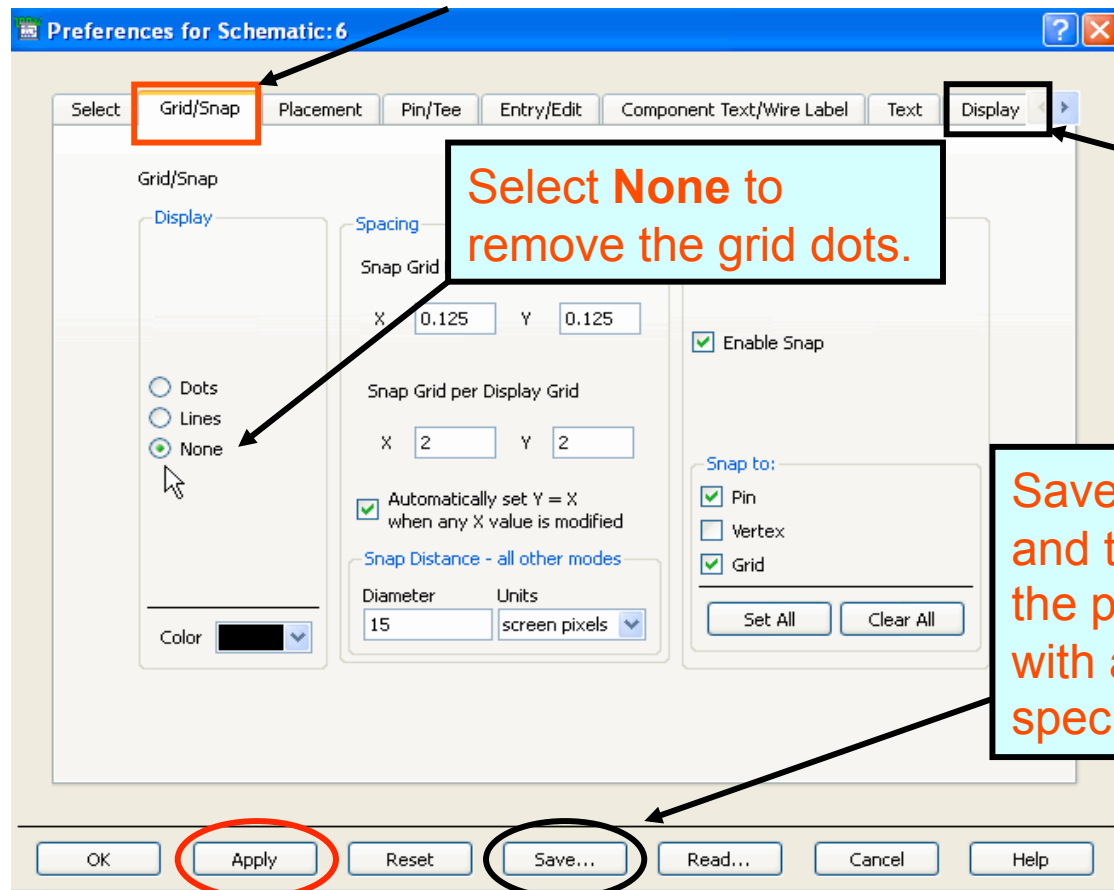Next, preferences... ➡

Answer: YES, if x is a valid VAR: x=1pF

# Schematic Preferences are available

Click: **Options > Preferences**

and select the desired TAB:



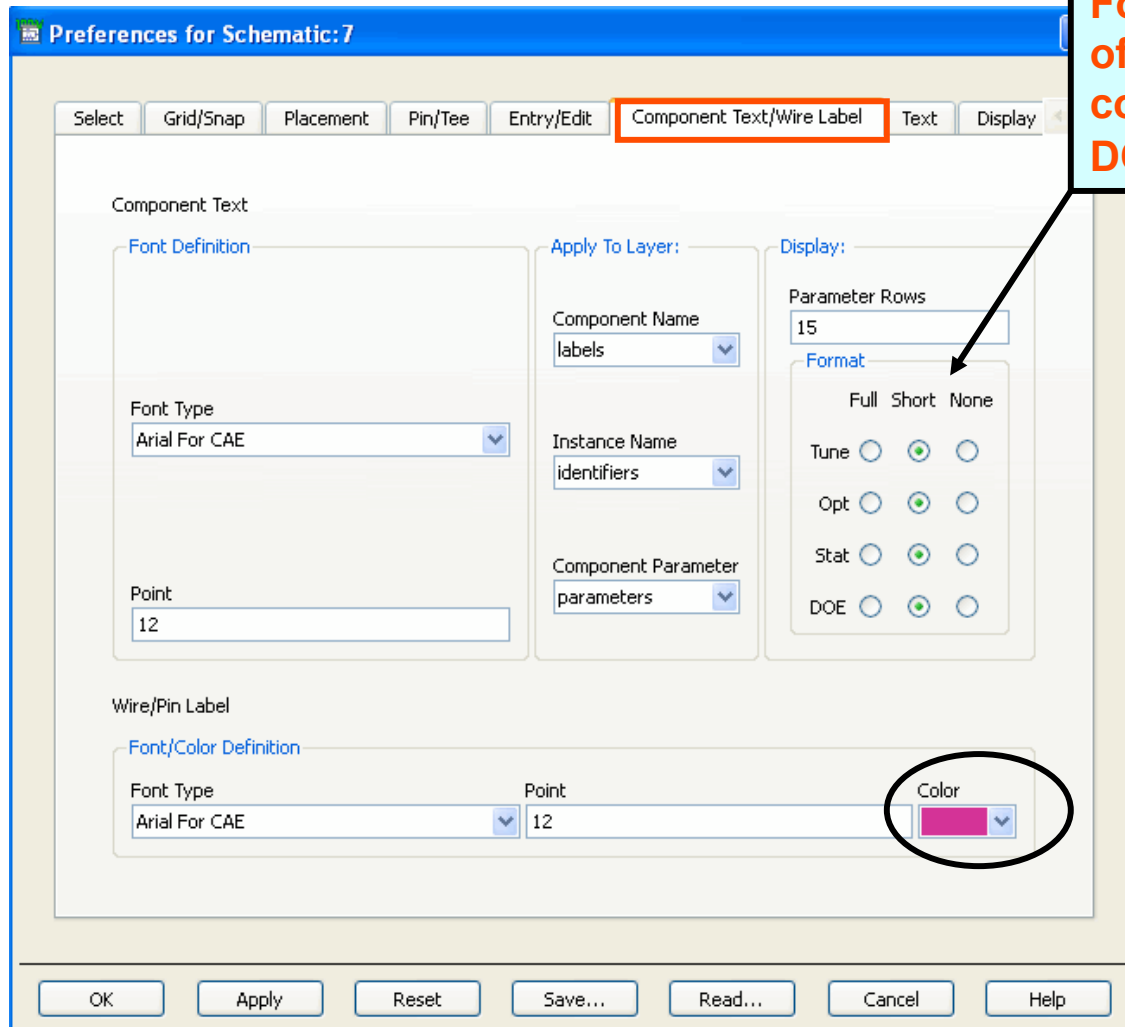Select **None** to remove the grid dots.

Go to the Display tab to set a different color.

Save settings to schematic.prf and they apply to all designs in the project. Or save/read them with another name (.prf) for specific designs.
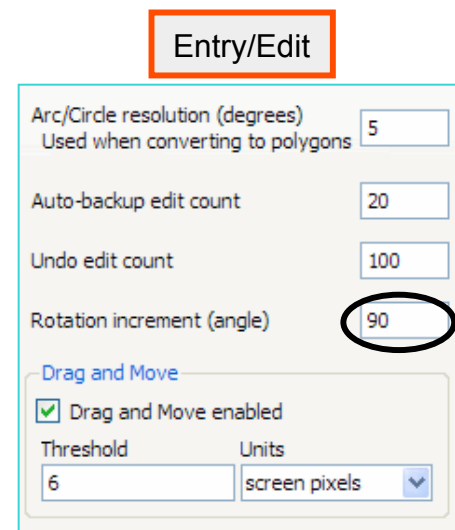
schematic.prf

# More Schematic Preferences

**Format: determines the amount of text used on schematic for component tuning, opt, stats, DOE.**



**NOTE: Set wire color in: Options > Layers.**

Entry/Edit

These are only some of the Entry choices.

Next,
Hot Key Exercise...

# ADS default Hot Keys

**Pre-configured keys**:
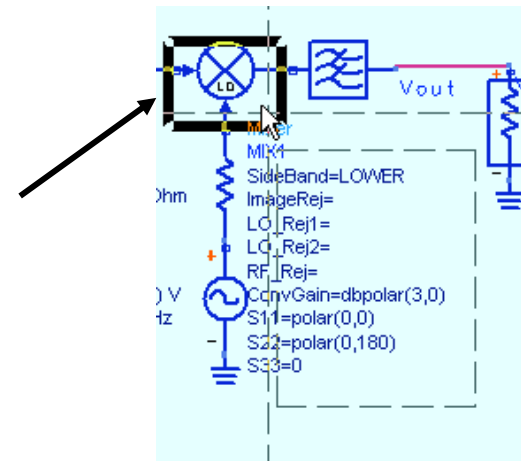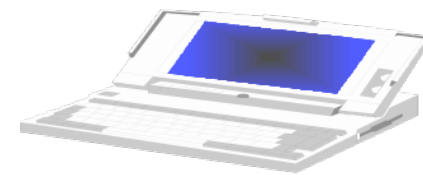
F7 = Simulate

F5 = Move Component Text

**Next to commands:**

| | | |
|---|---|---|
| ✂ Cut | | Ctrl+X |
| 📋 Copy | | Ctrl+C |
| 📋 Paste | | Ctrl+V |

**Try this now if ADS is running:** click the **F5** key, and select a component like the Mixer – then move the  cursor and the text will follow inside the dashed box…

Hot Keys are global for all projects!

**If you don't like mouse clicks, then use your keyboard: HOT KEYS! Hot keys are global for all projects.**

Mixer
MIX4
SideBand=LOWER
ImageRej=
LO_Rej1=
LO_Rej2=
RF_Rej=
ConvGain=dbpolar(3,0)
S11=polar(0,0)
S22=polar(0,180)
S33=0

Vout

# Setting up Hot Keys for all projects (global)

STEP 1: Create a new project: **amp_1900** (you will use this for the next few labs).
STEP 2: Open a new schematic and click **Tools > Hot Key / Toolbar Configuration**

STEP 3: Set up a hot key for the **Simulation Setup:** ⟶

a. Select the command.
b. Type in the letter: **s** (not case sensitive).
c. Click: **Assign** & **Apply**.
d. Press the S key to verify it works:
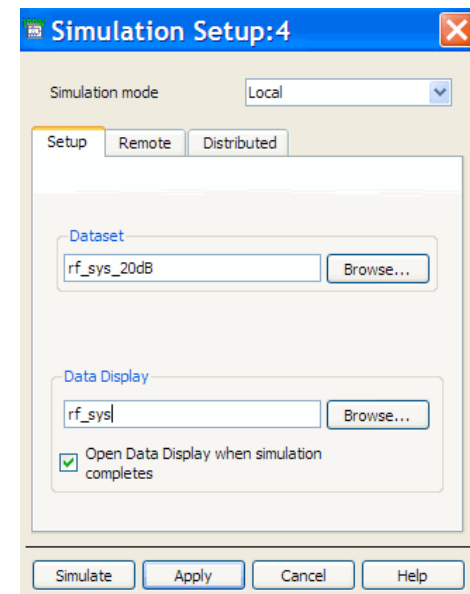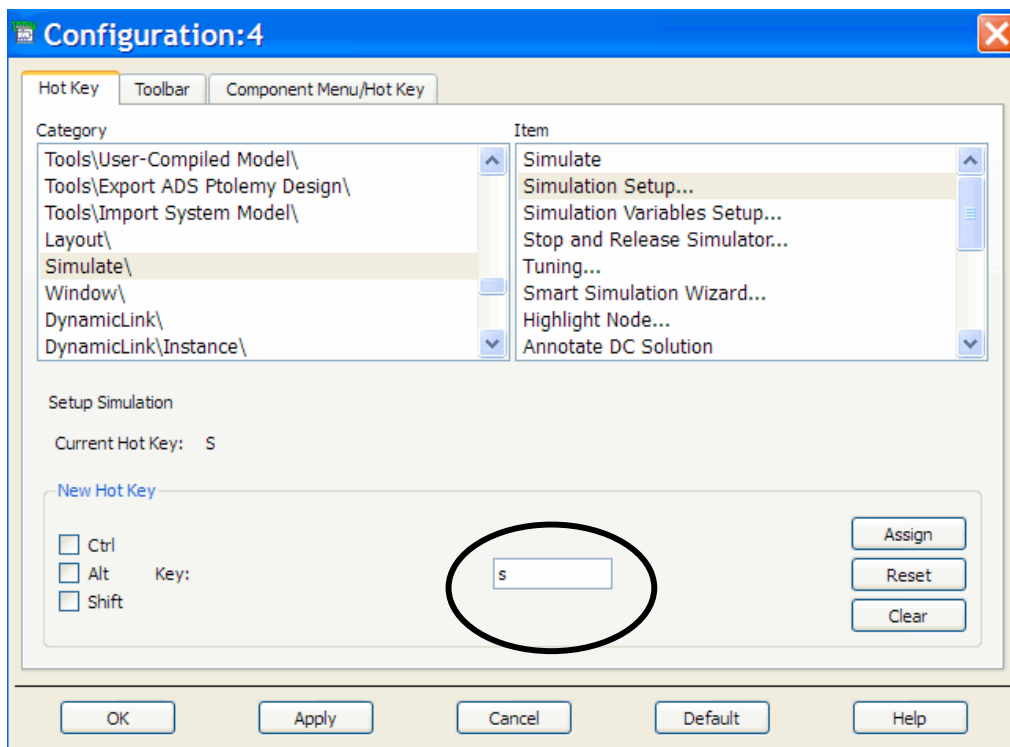


Also recommended: X for Edit > Move > Move and Disconnect
You can also use R for rotate instead of the default: Ctrl + R
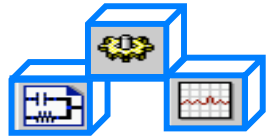
**What the lab is about ...**

# Lab 3:

# DC Simulations and modeling the sub-circuit

# Steps in the Design Process: labs 2-9

**You are here:**

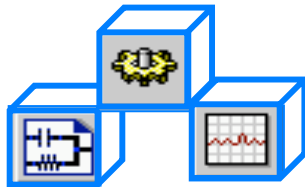Now its time to model the transistor as a sub-circuit and create a bias network that will be the core of the amplifier.

- Design the RF sys behavioral model receiver
- Test conversion gain, spectrum, etc.
- **Start amp_1900 design – subckt parasitics**
- **Simulate amp DC conditions & bias network**
- Simulate amp AC response - verify gain
- Test for noise contributions
- Simulate amp S-parameter response
- Create a matching topology
- Optimize the amp in & out matching networks
- Filter design – lumped 200MHz LPF
- Filter design – microstrip 1900 MHz BPF
- Transient and Momentum filter analysis
- Amp spectrum, delivered power, Zin - HB
- Test amp comp, distortion, two-tone, TOI
- CE basics for spectrum and baseband
- CE for amp_1900 with GSM source
- Replace amp and filters in rf_sys receiver
- Test conversion gain, NF, swept LO power
- Final CDMA system test CE with fancy DDS
- Co-simulation of behavioral system

# Start with some Amplifier specifications...

## AMP with max gain & low noise:

Available voltage: 5 volts
Device: Generic NPN - BJT (Gummel-Poon)
Collector current: about 3.25 mA
Frequency: RF = 1900 MHz
Gain:  > 15 dB (or much more with this model)
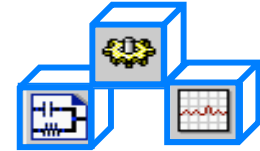50 ohm match: input and output

Later labs:  matching and testing the AMP for TOI, distortion, noise, compression, GSM & CDMA modulation response, and more.

**Filters:**  later labs -1900 MHz BPF (input) and LPF for the IF (output)

**YOUR JOB**:  Build, test, and refine the circuits to meet specifications.

Now – it is time to print out the lab exercise and then watch the lab video before you start:

Lab 1: Circuit Simulation Fundamentals

Lab 2: System Simulation Fundamentals

Exercise ➞ **Lab 3: DC Simulation and sub-circuit modeling**

Lab 4: AC Simulation and Noise

Lab 5: S-parameter Simulation and Optimization

Lab 6: Filters: Design Guide, Transient, Momentum, DAC

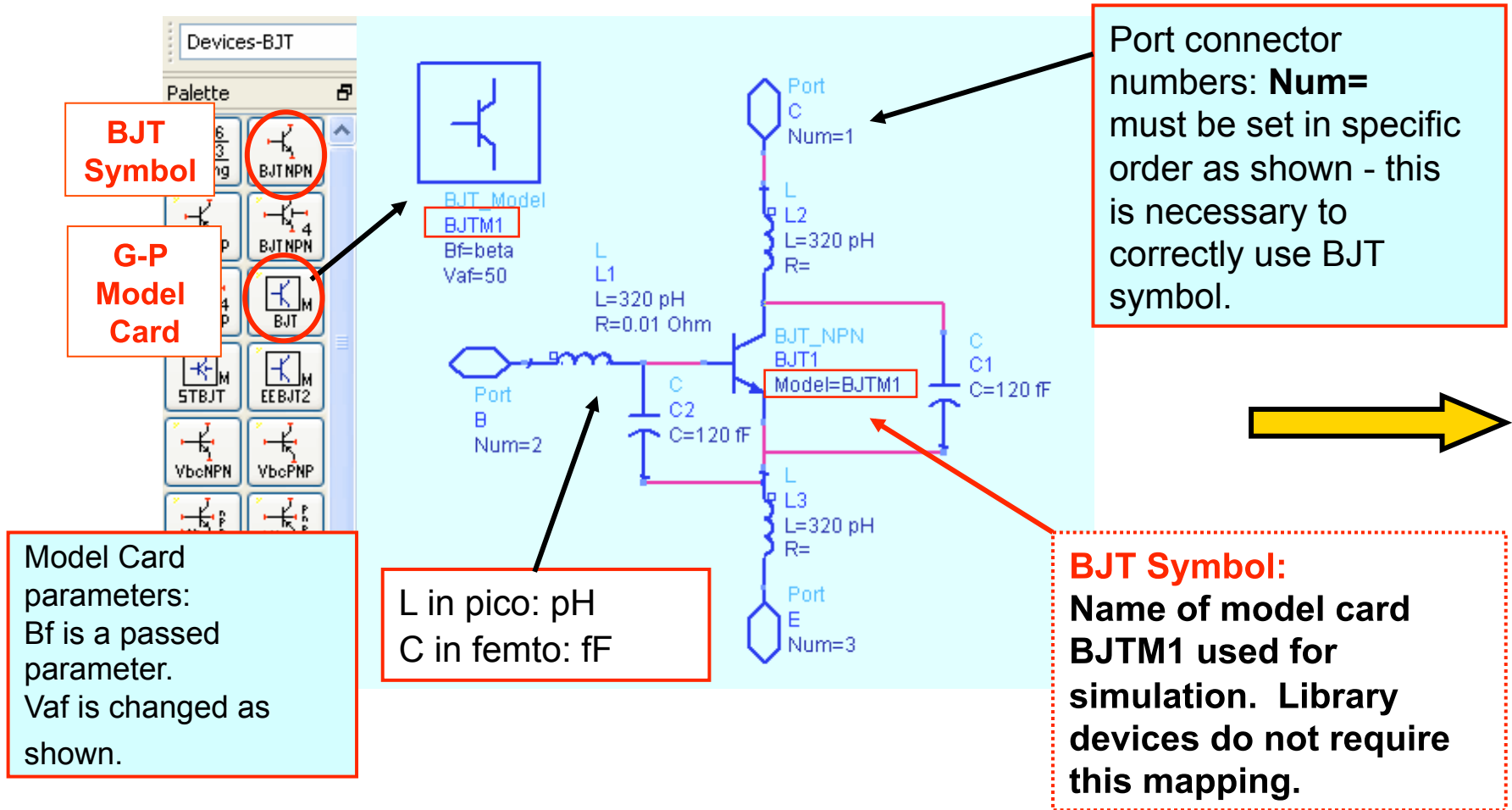Lab 7: Harmonic Balance Simulation

Lab 8: Circuit Envelope Simulation

Lab 9: Final Circuit and System Simulations

**Again, be sure to watch the lab video before doing the step-by-step exercise…**
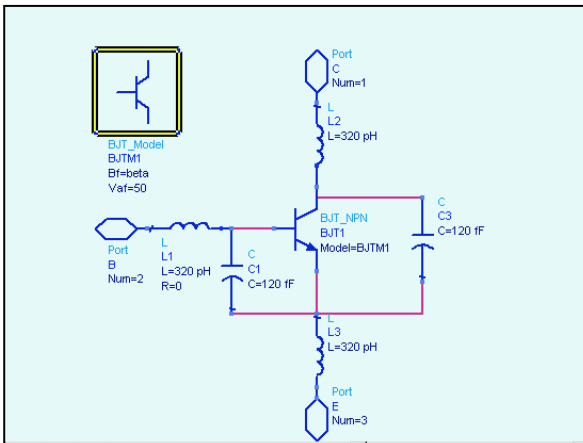
# BJT device with package parasitics

G-P Model Card, BJT symbol, parasitics, and port connectors:



Port connector numbers: **Num=** must be set in specific order as shown - this is necessary to correctly use BJT symbol.

Model Card parameters:
Bf is a passed parameter.
Vaf is changed as shown.

L in pico: pH
C in femto: fF

**BJT Symbol:**
**Name of model card BJTM1 used for simulation. Library devices do not require this mapping.**

# Viewing and creating a schematic symbol



**Schematic view**
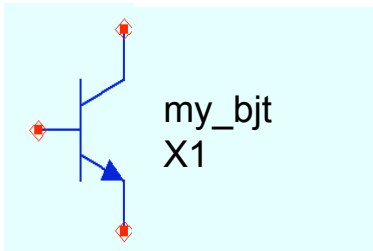
**Symbol view**
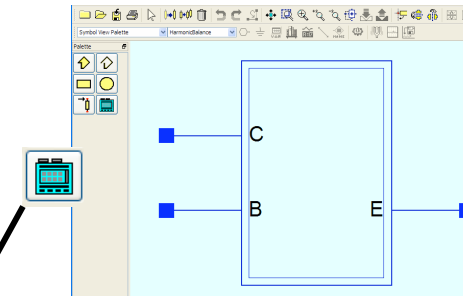
NOTE: Default symbol would have pins 1, 2, 3 if not renamed E, B, C.

Generate, draw, or copy/ modify a symbol:

my_bjt
X1

Or, you can specify the symbol using Design Parameters…

# Design Parameters for a schematic

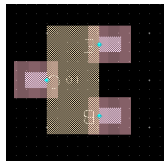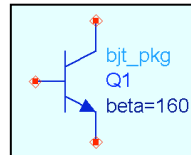Click: File > Design / Parameters to set parameters for your design:



- Description for library annotation
- Component Instance name: **Q**
- Symbol: **SYM_BJT_NPN**
- Passed parameter for Bf = **beta**

Artwork: Specify a layout type and name.

Symbol can display specific values.

You can copy parameters from other library models.

**Save the design and it is ready to use as a sub-circuit:**
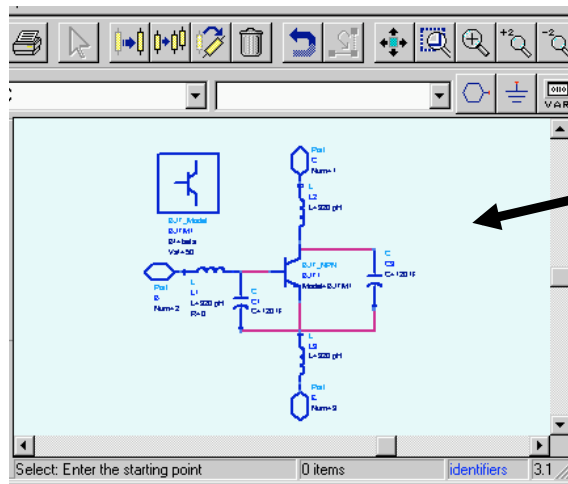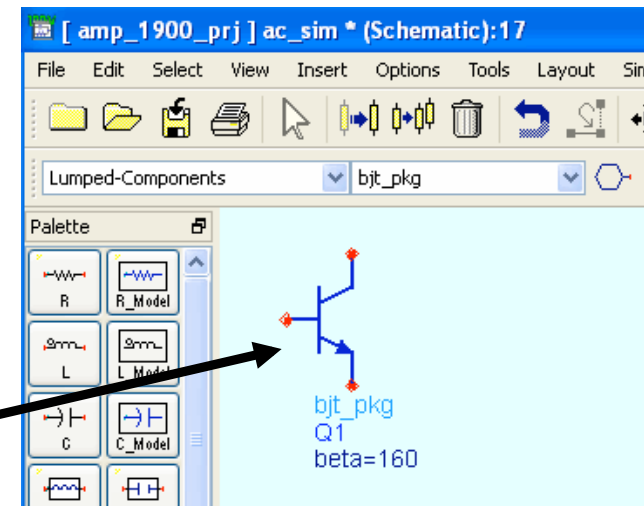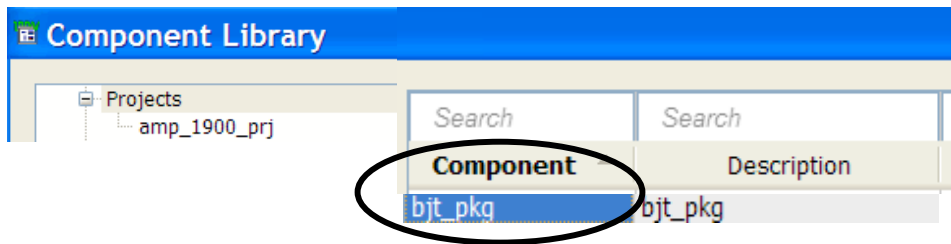
# Insert the model in a new schematic



Insert the sub-circuit from the library.
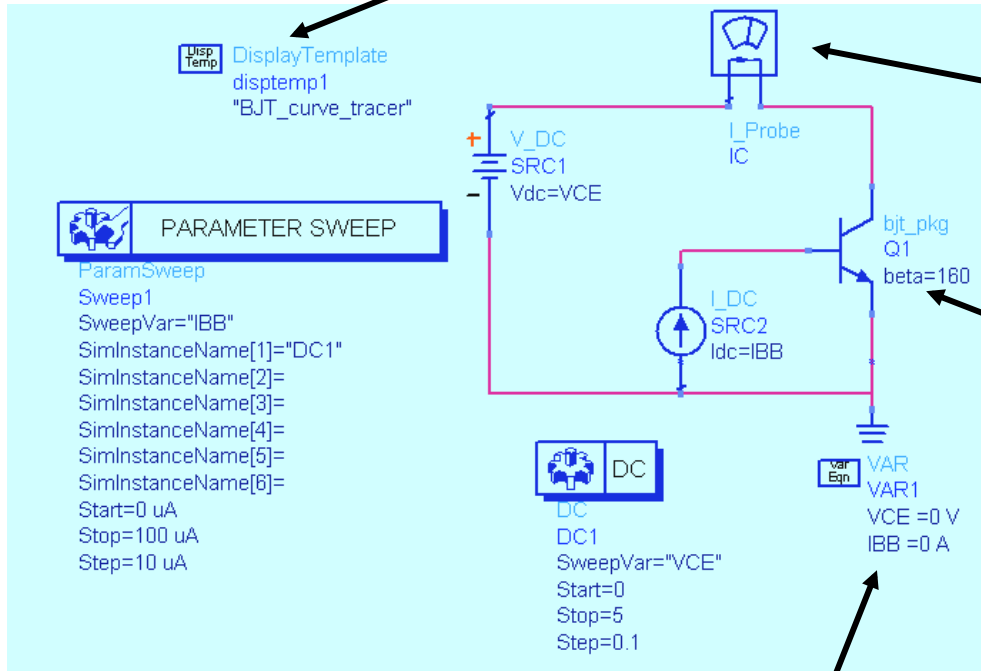
Design parameters follow the sub-circuit: Q1, beta, etc.

ICONS: Push into and Pop out of the hierarchy.

# Set up a DC curve sweep with a template

This template also has a data display template.

Design Content
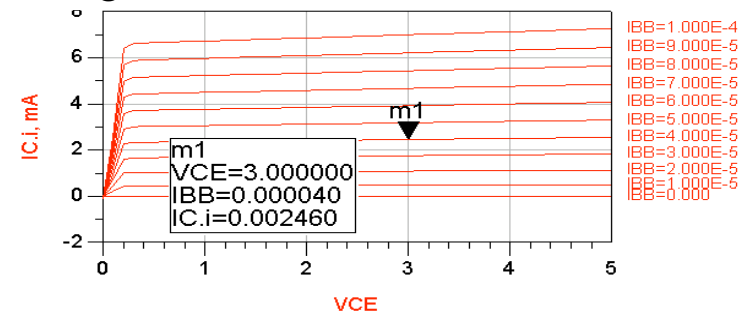
☐ Schematic Wizard

Schematic Design Templates (Optional)

BJT_curve_tracer

**Probe**: gives collector current in dataset – but pin currents can also be used (set in controller Output tab).

☑ Pin Currents

Your model (**bjt_pkg**) with annotation and passed parameter: **beta**.

Disp
Temp  DisplayTemplate
      disptemp1
      "BJT_curve_tracer"

+  V_DC
   SRC1
–  Vdc=VCE

I_Probe
IC

bjt_pkg
Q1
beta=160

PARAMETER SWEEP

ParamSweep
Sweep1
SweepVar="IBB"
SimInstanceName[1]="DC1"
SimInstanceName[2]=
SimInstanceName[3]=
SimInstanceName[4]=
SimInstanceName[5]=
SimInstanceName[6]=
Start=0 uA
Stop=100 uA
Step=10 uA

I_DC
SRC2
Idc=IBB

DC
DC1
SweepVar="VCE"
Start=0
Stop=5
Step=0.1

Var
Eqn  VAR
     VAR1
     VCE =0 V
     IBB =0 A

Initialized VARs: VCE=0V & IBB=0A

NOTE: DC controller sweeps the X-axis and the Parameter Sweep, sweeps the Y-axis.

## Data Display template gives curve tracer results:

IC.i, mA

m1
VCE=3.000000
IBB=0.000040
IC.i=0.002460

m1

IBB=1.000E-4
IBB=9.000E-5
IBB=8.000E-5
IBB=7.000E-5
IBB=6.000E-5
IBB=5.000E-5
IBB=4.000E-5
IBB=3.000E-5
IBB=2.000E-5
IBB=1.000E-5
IBB=0.000

VCE

# Calculate resistor values and test bias network

Setup a new sweep: then use voltage and current to calculate R values for the DC specs for Ib and Ic.
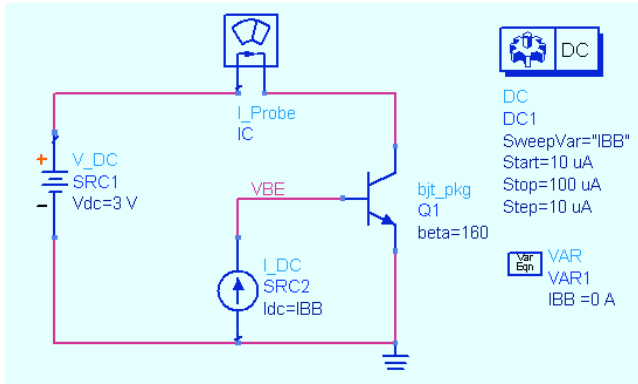
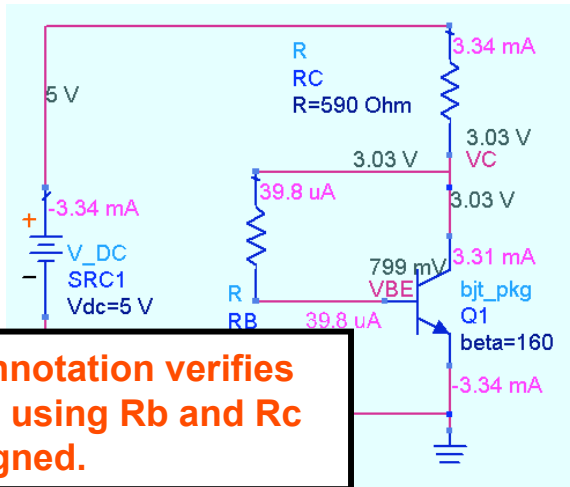| IBB | IC.i | VBE |
|---|---|---|
| 1.000E-5 | 599.8uA | 754.8mV |
| 2.000E-5 | 1.430mA | 777.1mV |
| 3.000E-5 | 2.349mA | 789.9mV |
| 4.000E-5 | 3.325mA | 798.8mV |
| 5.000E-5 | 4.341mA | 805.7mV |
| 6.000E-5 | 5.389mA | 811.3mV |



I_Probe
IC

V_DC
SRC1
Vdc=3 V

VBE

bjt_pkg
Q1
beta=160

I_DC
SRC2
Idc=IBB

DC
DC1
SweepVar="IBB"
Start=10 uA
Stop=100 uA
Step=10 uA

VAR
VAR1
IBB =0 A

Eqn Rb= (3-VBE) / IBB

Eqn Rc= 2 / IC.i

Bias Resistor Values @ 40 uA IBB

| Rb[3] | Rc[3] |
|---|---|
| 55029.037 | 601.500 |



5 V

R
RC
R=590 Ohm

3.34 mA

3.03 V
VC

3.03 V

-3.34 mA

39.8 uA

3.03 V

V_DC
SRC1
Vdc=5 V

R
RB

799 mV
VBE

39.8 uA

3.31 mA

bjt_pkg
Q1
beta=160

-3.34 mA

**Back Annotation verifies DC bias using Rb and Rc as designed.**

**OPTIONAL: Sweep Temperature**



m1
temp=-55.000
plot_vs(VC, temp)=3.079

m2
ind Delta=200.000
dep Delta=-0.122
delta mode ON

m3
temp=-55.000
plot_vs(VBE, temp)=0.899

Start the lab now!

Agilent Technologies