ECE 304: Using PSPICE Breakout Parts

Table of Contents

ECE 304: Using PSPICE Breakout Parts

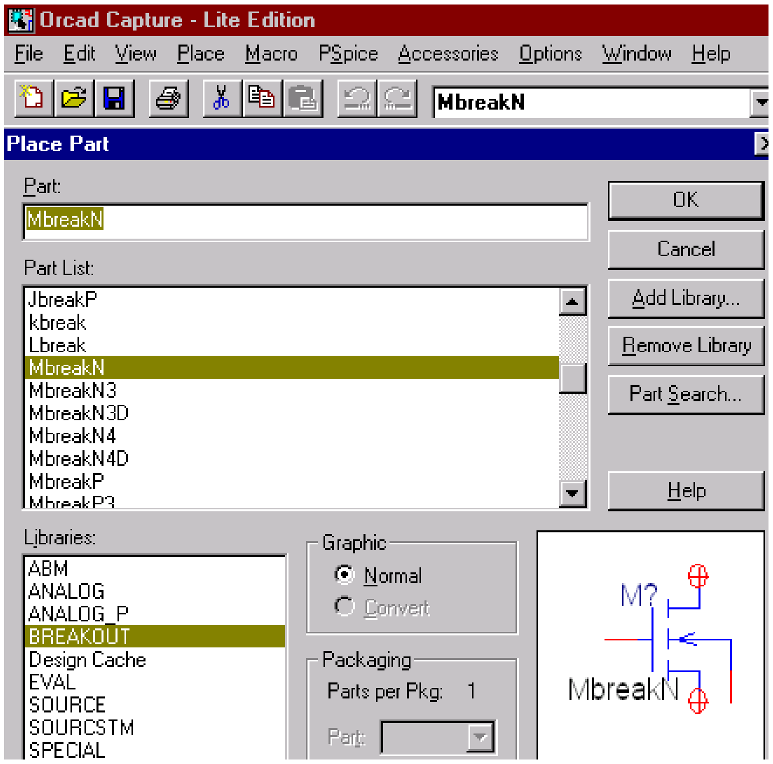
Background

Often PSPICE does not have the part we want. For example, in the power amplifier part of the course, we need models for the power bipolar transistors that are not available in PSPICE. Also, we often want to use simple models for thinking about circuits, for example, we may want a simplified bipolar transistor with infinite output resistance (infinite Early voltage) and most of the other model parameters eliminated.

A major category of examples where custom devices are needed is MOSFET's. PSPICE has only power MOSFET's in its parts library, so we have to build our own MOSFET's.

Approach

Following Herniter, §7.B, p. 403, we use PLACE/PART and select the BREAKOUT LIBRARY.



Part: MbreakN

Part List:
JbreakP
kbreak
Lbreak
MbreakN
MbreakN3
MbreakN3D
MbreakN4
MbreakN4D
MbreakP
MbreakP3

Libraries:
ABM
ANALOG
ANALOG_P
BREAKOUT
Design Cache
EVAL
SOURCE
SOURCSTM
SPECIAL

Graphic: [✓ Normal] Convert

Packaging: Parts per Pkg: 1

Part:

FIGURE 1:

Screen dump showing selection of the breakout library part of the n-channel MOSFET

For an n-channel MOSFET, we choose the part MbreakN. Pasting it on the schematic, we highlight it (left click) and select EDIT/PSPICE MODEL.

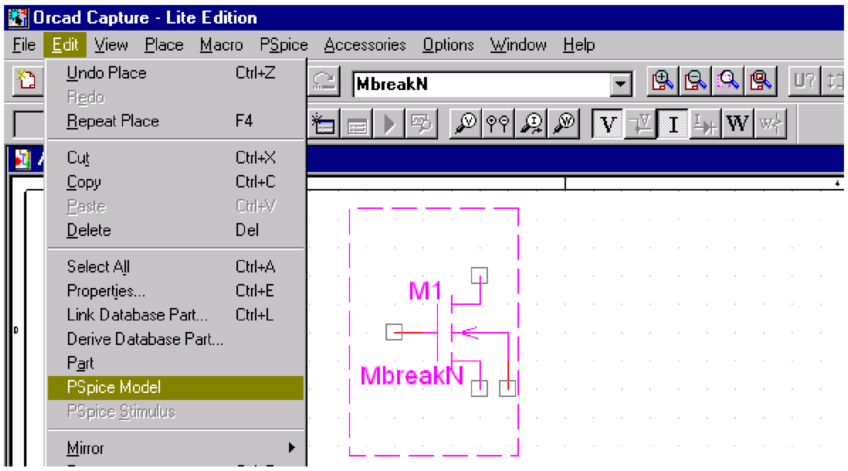
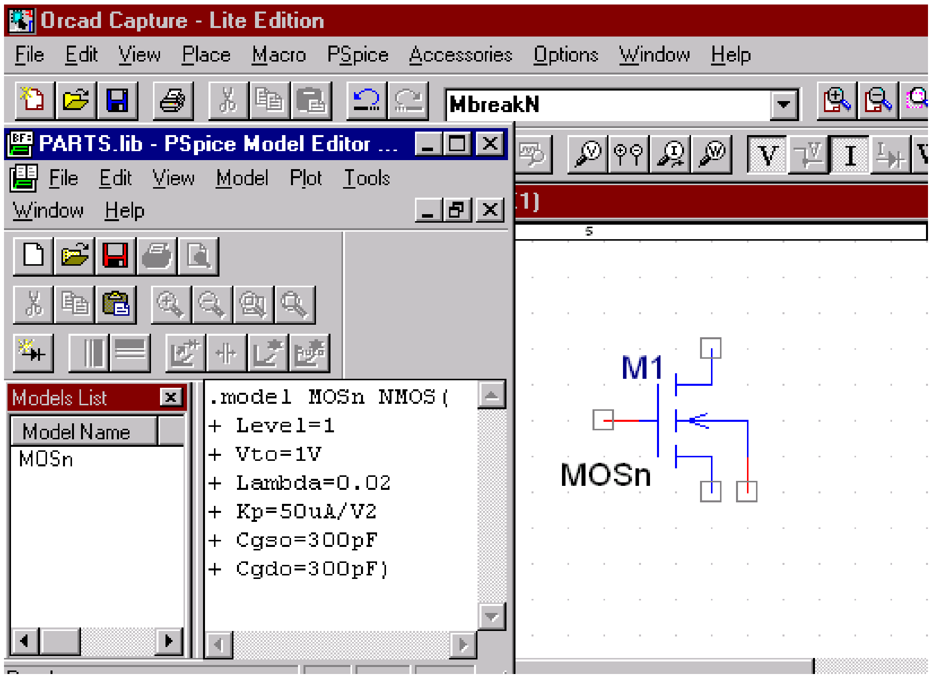


FIGURE 2:

Screen dump showing menu selection to edit the MOSFET ●model statement

The editor then opens, and we type in the ●model statement that we want to use, as shown in Figure 3. Also as shown in Figure 3, every new line in the ●model statement must have a line continuation '+' sign or it will be ignored. It's a good idea to use lower case letters in the symbols to avoid confusing a zero ‘0’ with an 'O'.

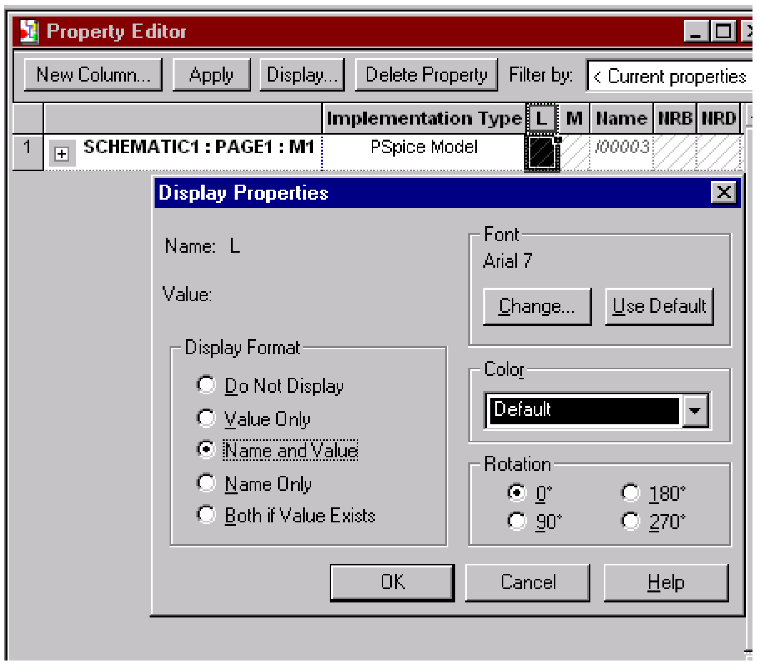

Orcad Capture - Lite Edition
File Edit View Place Macro PSpice Accessories Options Window Help
MbreakN
PARTS.lib - PSpice Model Editor ...
File Edit View Model Plot Tools
Window Help
Models List
Model Name
MOSn
M1
MOSn

```
.model MOSn NMOS(
+ Level=1
+ Vto=1V
+ Lambda=0.02
+ Kp=50uA/V2
+ Cgso=300pF
+ Cgdo=300pF)
```

FIGURE 3:

Screen dump showing the ●model editor with the ●model statement filled in. After selecting FILE/SAVE, the name of the part changes from MbreakN to name entered on the ●model statement, MOSn in this case. The part name also appears on the schematic, as shown here.

Next, we have to set the length and width of the MOSFET. Double-click on the MOSFET symbol to obtain the PROPERTY EDITOR menu. Scroll horizontally to the length variable L and highlight both L and the square underneath.


Property Editor

New Column... | Apply | Display... | Delete Property | Filter by: < Current properties

		Implementation Type	L	M	Name	NRB	NRD
1	SCHEMATIC1 : PAGE1 : M1	PSpice Model			I00003		

Display Properties

Name: L

Value:

Display Format: Do Not Display / Value Only / [✓ Name and Value] / Name Only / Both if Value Exists

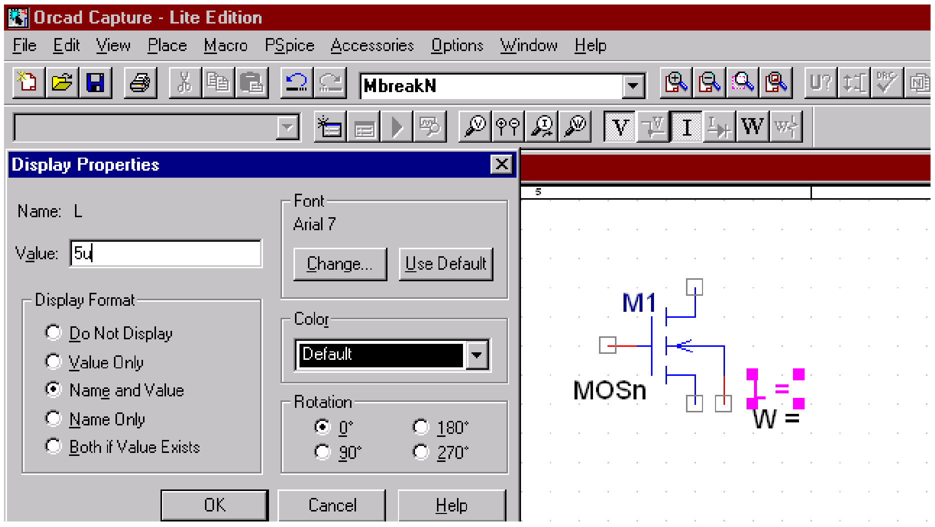
Font: Arial 7

Color: Default

Rotation: [✓ 0°] / 90° / 180° / 270°

FIGURE 4:
Screen dump showing setting of the display properties for the length parameter.

Clicking on the DISPLAY tab brings down the DISPLAY PROPERTIES menu. Select NAME AND VALUE, as shown in Figure 4 and click OK. Then do the same for the width parameter W. On the schematic the variables L= and W= will appear. Double clicking on 'L=', we obtain another DISPLAY PROPERTIES menu allowing us to set the value of L, as shown in Figure 5.


Orcad Capture - Lite Edition

Display Properties

Name: L

Value: 5u

Display Format: Do Not Display / Value Only / [✓ Name and Value] / Name Only / Both if Value Exists

Font: Arial 7

Color: Default

Rotation: [✓ 0°] / 90° / 180° / 270°

FIGURE 5:
Screen dump showing how to set the MOSFET length parameter L.

Also shown in Figure 5 is the view on the schematic of the highlighted 'L='. Your job is to reproduce the MOSFET current mirror shown in Figure 6.

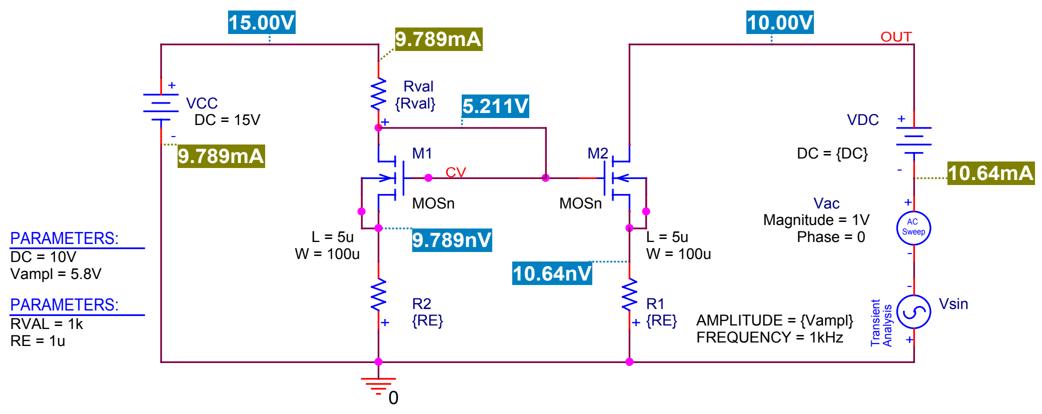


FIGURE 6:
A MOSFET current mirror built with the breakout MOSFET part MOSn.

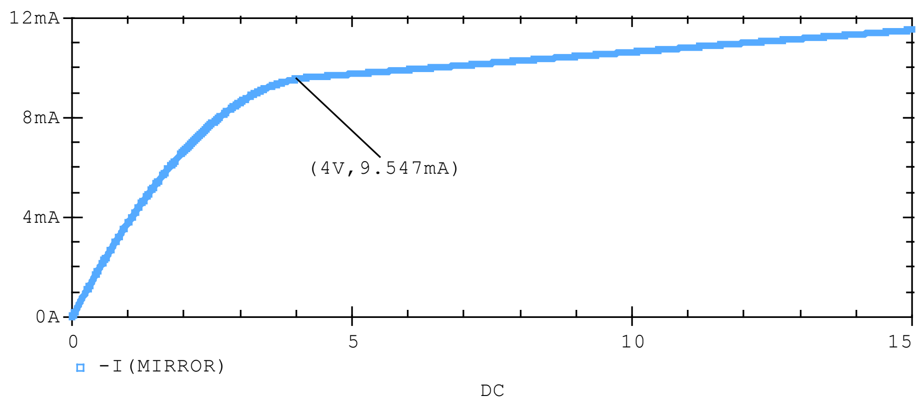


FIGURE 7:
DC *I-V* characteristic of MOSFET current mirror of Figure 6.

Using the Breakout Part in Another Schematic

Once you have created a new part like the MOSFET with model MOSn, you may want to use it in another new schematic. There are at least two ways to do this.

The most straightforward way is to recreate the part in the new schematic by copying the •model statement of the part in the old schematic using EDIT/PSPICE MODEL. Then a new breakout part is pasted into the new schematic, and the copied •model listing is pasted into this breakout part.

Another simple way is to create the new circuit except for the breakout parts. As an example, suppose the circuit we started with is the current mirror of Figure 6 and we want to create a common source amplifier using the same MOSFET. We create the amplifier circuit except for the MOSFET. Then we open the current mirror project and place the two •OPJ files side by side as in Figure 8.

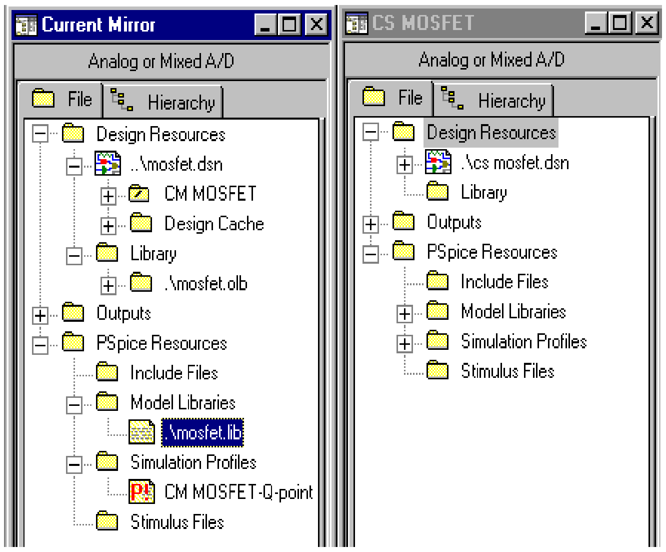


FIGURE 8:

The two •OPJ files side by side. The library \mosfet.lib is highlighted in the CM project.

Under MODEL LIBRARIES in the Current Mirror project is the library file \mosfet.lib. We copy this file using EDIT/COPY on the TOOLBAR and move the mouse over to MODEL LIBRARIES in the Common Source (CS) Amplifier Project, where we paste the file. The CS Amplifier Project now can read the •model parameters for MOSFET model MOSn. Next, copy the MOSFET from the Current Mirror schematic and paste it into the CS Amplifier schematic. The part library now can be found and the amplifier circuit will simulate without error messages. A problem with this approach is that the CS Amplifier project doesn't have its own copy of the model MOSn. Rather, it just knows how to find it. So, should you erase the Current Mirror project later, when it has outlived its usefulness, the CS Amplifier project won't work anymore, a nasty surprise I call the domino effect.

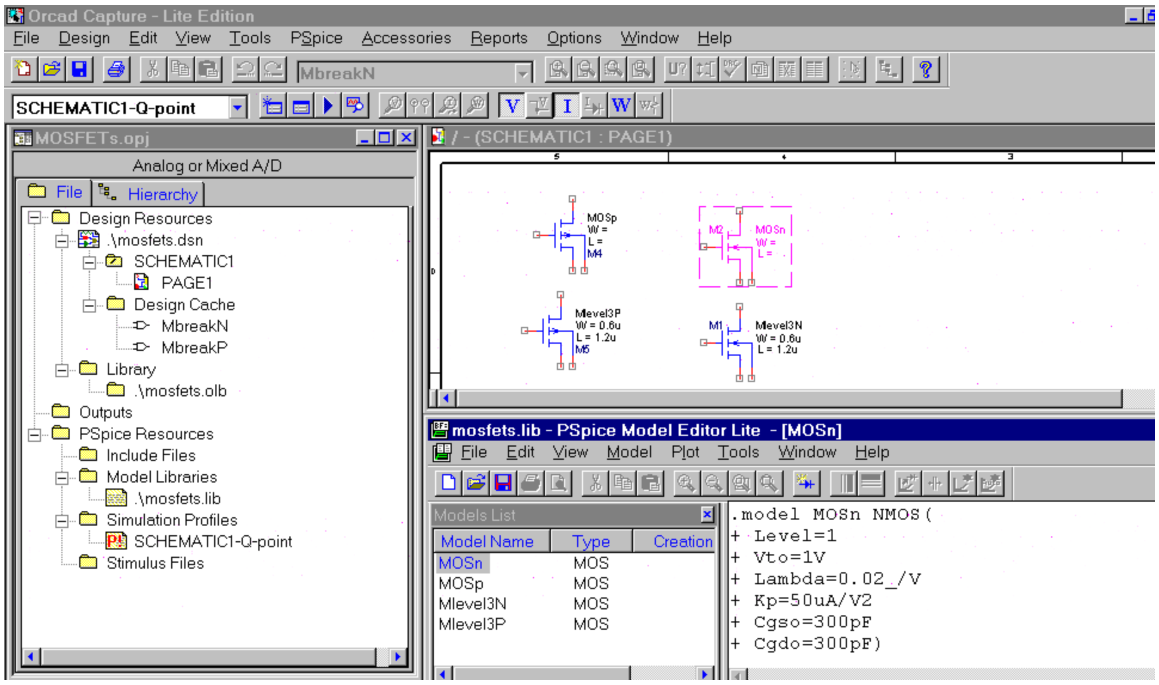
A more reliable method is to inventory our parts in a special "parts library schematic". We are unlikely to erase this schematic by accident later on, so we can remove any projects we have no use for without any domino effect.

A "Parts-storage" Schematic

We want to make a project that is simply a library of custom made MOSFETs, say MOSFETs.opj as shown in Figure 9. This schematic is not a circuit, it is just a bunch of pasted MOSFETs, each with its custom •model statement. Compare with Herniter §7.B, pp. 403-407. Then you can direct all your projects using custom MOSFETs back to MOSFETs•lib of this project, or use ADD AS GLOBAL (explained later) so it is automatically included.

Figure 9 shows a MOSFET part-storage schematic. You can put any variation of a MOSFET you like in such a library. In this example, instead of the standard Level 1 MOSFET model, two of these MOSFETs use a Level 3 MOSFET model taken from a student fabrication project using MOSIS. There is no denying the convenience of the part-storage approach.

It is advisable to change the permissions on the MOSFETs.lib file to READ ONLY using the PROPERTIES menu when you right click on the file in the FILE EXPLORER. This choice prevents the program from overwriting your library, which otherwise can happen in ways you might neither imagine nor control.


Orcad Capture - Lite Edition
File Design Edit View Tools PSpice Accessories Reports Options Window Help
MbreakN
SCHEMATIC1-Q-point
MOSFETs.opj
Analog or Mixed A/D
File Hierarchy
/ - (SCHEMATIC1 : PAGE1)
MOSp W = L = M4
M2 MOSn W = L =
Mlevel3P W = 0.6u L = 1.2u M5
M1 Mlevel3N W = 0.6u L = 1.2u
mosfets.lib - PSpice Model Editor Lite - [MOSn]
File Edit View Model Plot Tools Window Help
Models List

- Design Resources
  - .\mosfets.dsn
    - SCHEMATIC1
      - PAGE1
    - Design Cache
      - MbreakN
      - MbreakP
  - Library
    - .\mosfets.olb
- Outputs
- PSpice Resources
  - Include Files
  - Model Libraries
    - .\mosfets.lib
  - Simulation Profiles
    - SCHEMATIC1-Q-point
  - Stimulus Files

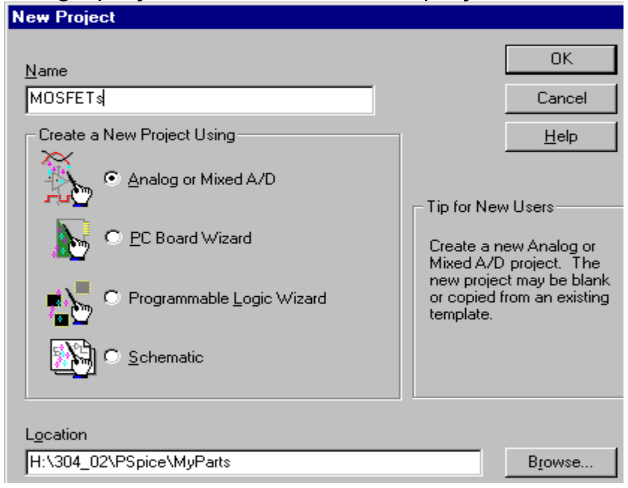
Model Name	Type	Creation
MOSn	MOS	
MOSp	MOS	
Mlevel3N	MOS	
Mlevel3P	MOS	

```
.model MOSn NMOS (
+ Level=1
+ Vto=1V
+ Lambda=0.02_/V
+ Kp=50uA/V2
+ Cgso=300pF
+ Cgdo=300pF)
```

FIGURE 9:

A parts-storage schematic with four MOSFET models stored in MOSFETs.opj with a library named …\mosfets.lib. A library is limited to 15 parts.

To set up the parts storage project, we create a new project as shown in Figure 10.



Name: MOSFETs

Create a New Project Using
- (●) Analog or Mixed A/D
- ( ) PC Board Wizard
- ( ) Programmable Logic Wizard
- ( ) Schematic

Tip for New Users

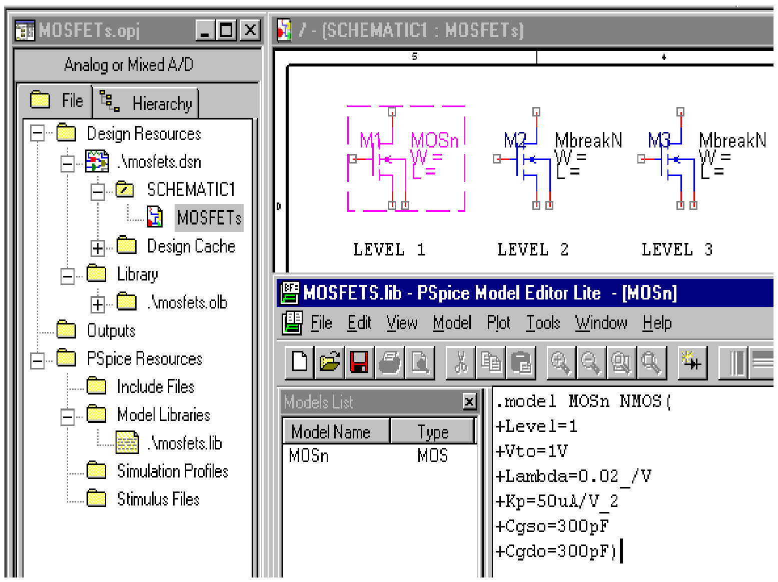
Create a new Analog or Mixed A/D project. The new project may be blank or copied from an existing template.

Location: H:\304_02\PSpice\MyParts

FIGURE 10:

Setting up a parts storage project MOSFETs in a folder called MyParts with path H:\304_02\PSPICE\MyParts

Then we create our parts, for example, by pasting breakout part MbreakN and editing it with the EDIT/PSPICE MODEL menu. See Figure 11.



Model Name	Type
MOSn	MOS

```
.model MOSn NMOS(
+Level=1
+Vto=1V
+Lambda=0.02_/V
+Kp=50uA/V_2
+Cgso=300pF
+Cgdo=300pF)
```

FIGURE 11:
Filling in the part descriptions using the EDIT/PSPICE MODEL menu

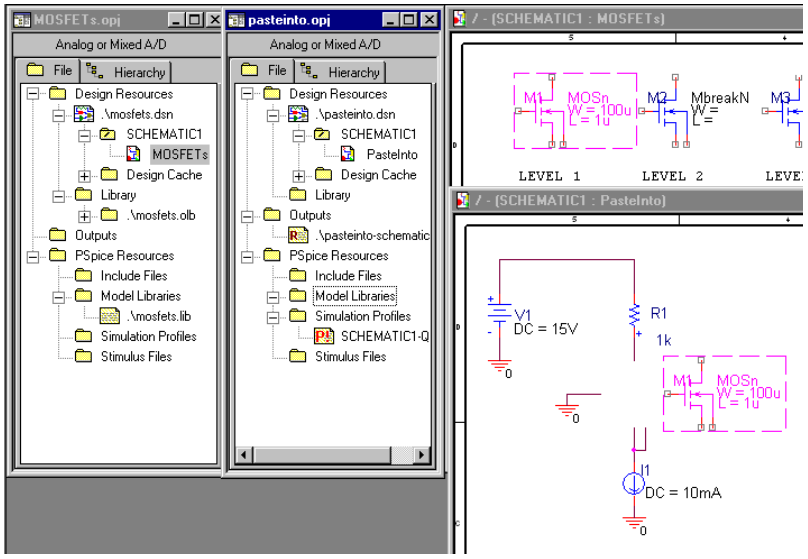
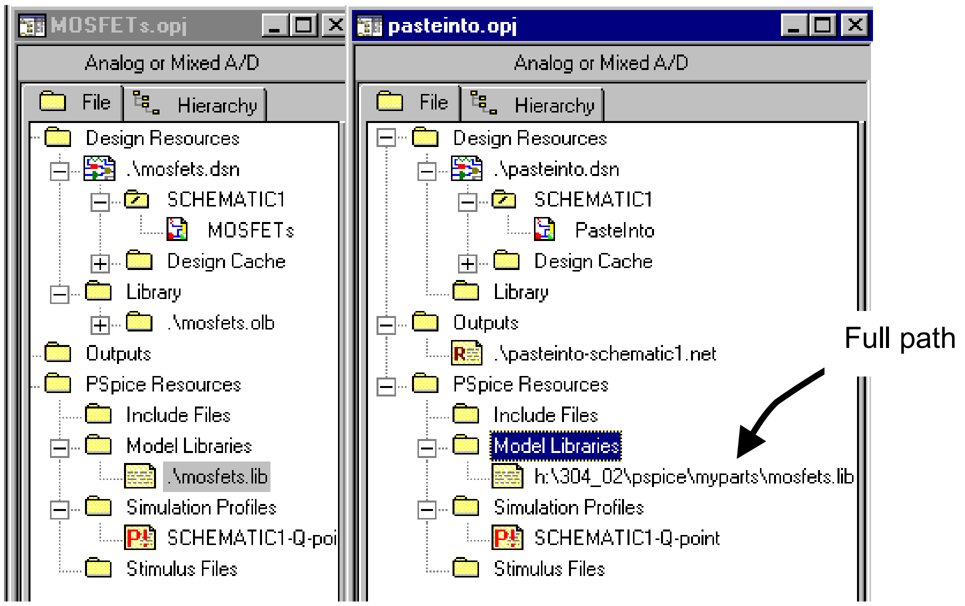


FIGURE 12:
Pasting the part MOSn copied from the parts storage project into the PasteInto project

To use the parts storage project in a new schematic, say the Paste Into project, copy the part in the storage project and paste it on the new schematic. See Figure 12. Next we have to add the path for the part description to the PasteInto project. We set the two •OPJ files side by side, then highlight …\mosfets.lib in the MOSFETs storage project, go to the TOOLBAR and select EDIT/COPY, move the mouse to MODEL LIBRARIES in the PasteInto project, EDIT/PASTE, and like magic, the model file appears with the full path name. See Figure 13.


MOSFETs.opj

Analog or Mixed A/D

File Hierarchy

```
Design Resources
  .\mosfets.dsn
    SCHEMATIC1
      MOSFETs
    Design Cache
  Library
    .\mosfets.olb
Outputs
PSpice Resources
  Include Files
  Model Libraries
    .\mosfets.lib
  Simulation Profiles
    SCHEMATIC1-Q-poi
  Stimulus Files
```

pasteinto.opj

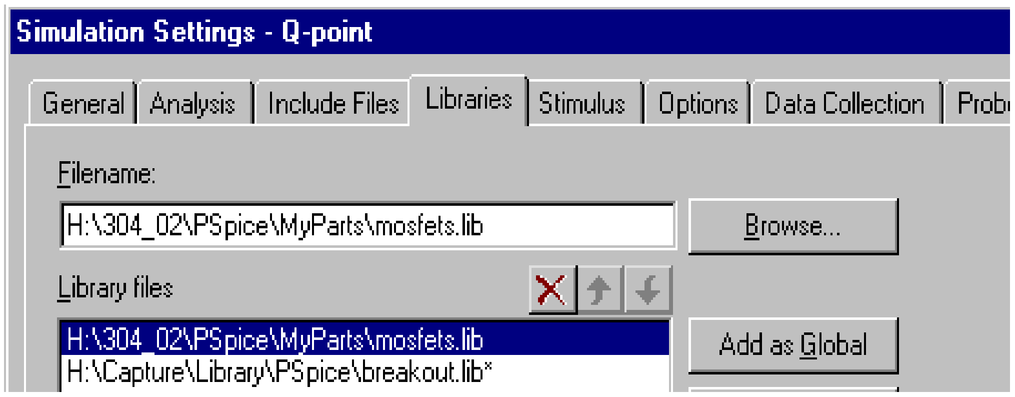
Analog or Mixed A/D

File Hierarchy

```
Design Resources
  .\pasteinto.dsn
    SCHEMATIC1
      PasteInto
    Design Cache
  Library
Outputs
  .\pasteinto-schematic1.net
PSpice Resources
  Include Files
  Model Libraries
    h:\304_02\pspice\myparts\mosfets.lib
  Simulation Profiles
    SCHEMATIC1-Q-point
  Stimulus Files
```

FIGURE 13:
The two •OPJ files after …\mosfets.lib is copied into MODEL LIBRARIES of the PasteInto project

Now set up a SIMULATION PROFILE, say a Q-point analysis. If we now look at the PSPICE/EDIT SIMULATION PROFILE menu and select the LIBRARIES tab (see Figure 14), we find that the MyParts\MOSFETs.lib has no * after its name. That is, it has been added to this design only, and not as a library available to every project. We can make MyParts\MOSFETs.lib available to all projects by pressing ADD AS GLOBAL in the menu of Figure 14. Then it automatically appears in the library list for all projects and we don't have to go through this procedure.


Simulation Settings - Q-point

Filename: H:\304_02\PSpice\MyParts\mosfets.lib

Library files:

```
H:\304_02\PSpice\MyParts\mosfets.lib
H:\Capture\Library\PSpice\breakout.lib*
```

FIGURE 14:
The LIBRARIES tab in PSPICE/EDIT SIMULATION PROFILE

Multiple Substitutions of Breakout Parts

Suppose we want to look at a current mirror with a number of different bipolar transistors. As an example, consider the current mirror in Figure 15.

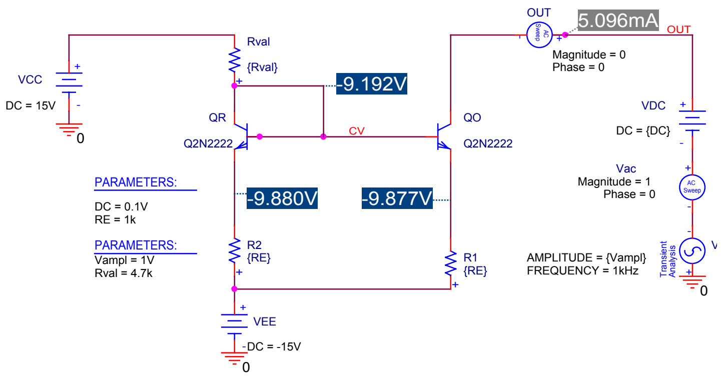


FIGURE 15:

Current mirror schematic using Q2N2222 transistors, reference transistor Q_R on the left, output transistor Q_O on the right.

For example, we may want to look at the effects of different β's in the two transistors Q_O and Q_R. We decide to make Q_R a fixed β-value and vary the β in Q_O. Then we will do the reverse: fix Q_O and vary Q_R. For the fixed transistor, we want to look at two cases: a low β and a high β. We replace Q_R with a breakout transistor QbreakN. The breakout transistor is to be one of two types: a high- and a low-β type, modeled by changing the BF model parameter in the Q2N2222. An efficient way to implement all these swaps is to create a parts-storage schematic with both types. The schematic is not a working circuit, just a collection of parts.

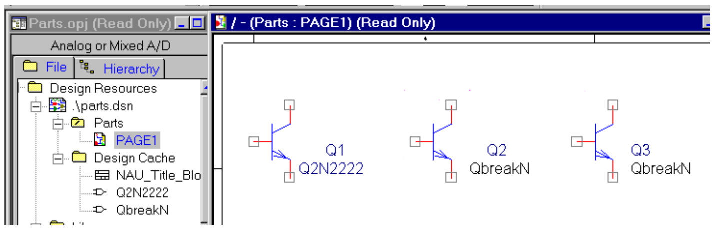
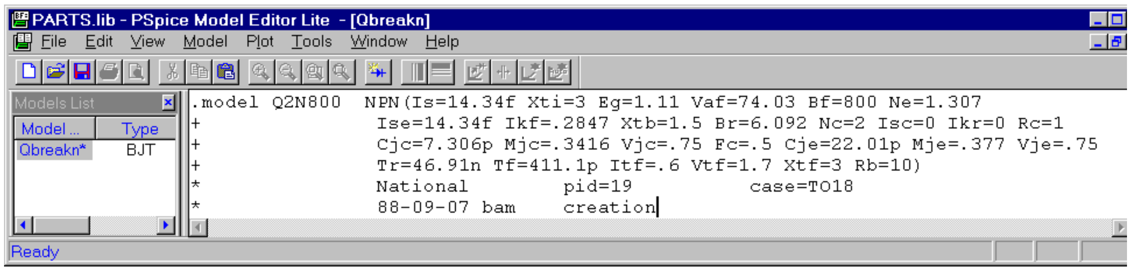


FIGURE 16:

A parts-storage project *Parts•opj* with a Q2N2222 and two QbreakN breakout NPN bipolars modified as shown below. The file is labeled Read Only because it has been set this way to avoid overwriting, also described below.



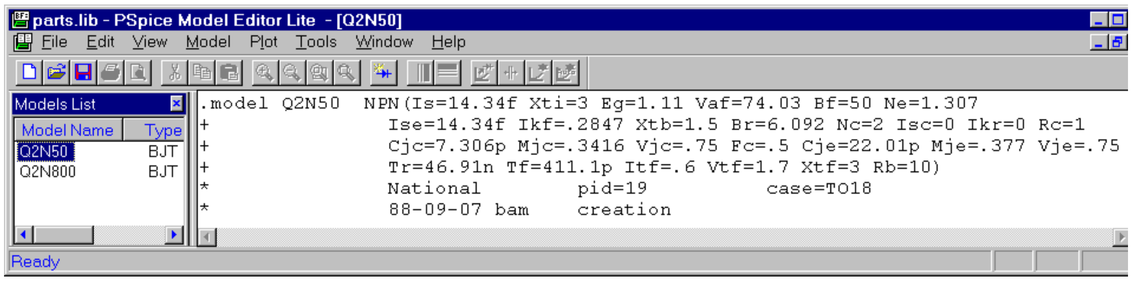
PARTS.lib - PSpice Model Editor Lite - [Qbreakn]
File Edit View Model Plot Tools Window Help
Models List
Model ...
Type
Qbreakn*
BJT
Ready

```
.model Q2N800  NPN(Is=14.34f Xti=3 Eg=1.11 Vaf=74.03 Bf=800 Ne=1.307
+              Ise=14.34f Ikf=.2847 Xtb=1.5 Br=6.092 Nc=2 Isc=0 Ikr=0 Rc=1
+              Cjc=7.306p Mjc=.3416 Vjc=.75 Fc=.5 Cje=22.01p Mje=.377 Vje=.75
+              Tr=46.91n Tf=411.1p Itf=.6 Vtf=1.7 Xtf=3 Rb=10)
*              National        pid=19          case=TO18
*              88-09-07 bam    creation
```

FIGURE 17:

The PSPICE model editor with the custom breakout part Q2N800, which is the same as the Q2N2222 except BF = 800. After FILE/SAVE the part is listed under MODEL LIST.

Figure 17 shows the set-up for the high-β Q2N222, named for convenience as Q2N800. Figure 18 is the listing for the low-β Q2N2222.


parts.lib - PSpice Model Editor Lite - [Q2N50]
File Edit View Model Plot Tools Window Help
Models List
Model Name Type
Q2N50 BJT
Q2N800 BJT
Ready

```
.model Q2N50   NPN(Is=14.34f Xti=3 Eg=1.11 Vaf=74.03 Bf=50 Ne=1.307
+                Ise=14.34f Ikf=.2847 Xtb=1.5 Br=6.092 Nc=2 Isc=0 Ikr=0 Rc=1
+                Cjc=7.306p Mjc=.3416 Vjc=.75 Fc=.5 Cje=22.01p Mje=.377 Vje=.75
+                Tr=46.91n Tf=411.1p Itf=.6 Vtf=1.7 Xtf=3 Rb=10)
*                National        pid=19          case=TO18
*                88-09-07 bam    creation
```

FIGURE 18:

The PSPICE model editor for custom breakout part Q2N50 with BF=50

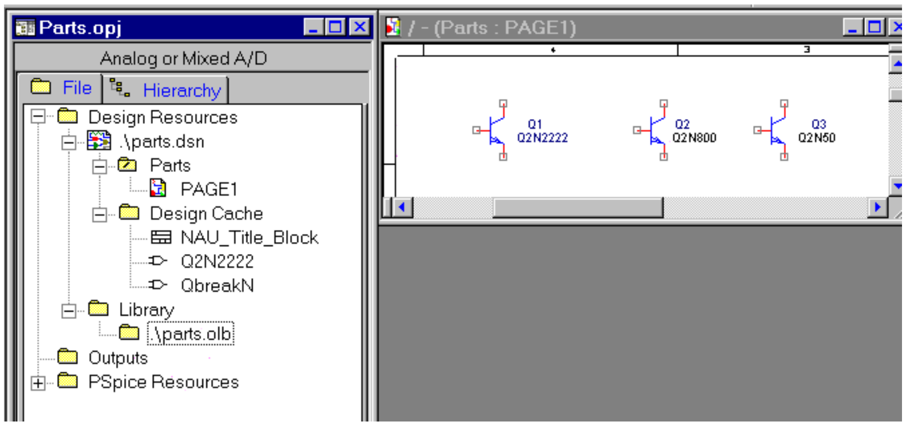


FIGURE 19:

The parts schematic and •opj file. The schematic has been renamed "Parts".

CAN'T FIND THE LIBRARY?

In some cases, Orcad saves files without asking where you want them[1]. So, your files may be somewhere in your file system, maybe where you were working last. If EDIT/PSPICE MODEL cannot find the *Parts.lib* file, or the OUTPUT FILE warns you of a missing library, possibly it was saved somewhere else. You can look for it using the Search feature of the File Explorer.

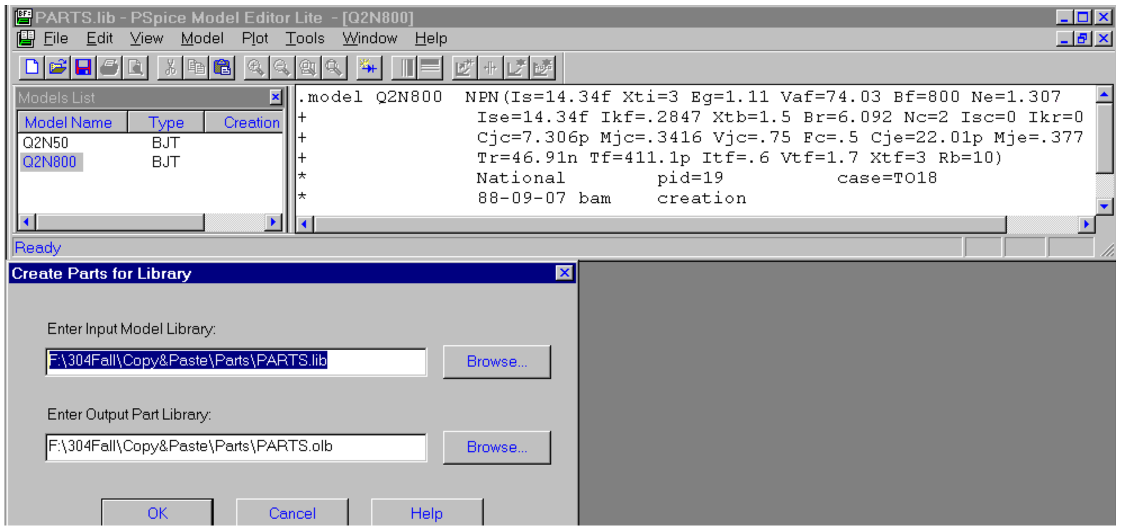
If you just can't find the *Parts.lib* library, delete all the *Parts* files and start over.

MAKING THE •OLB LIBRARY

Besides the *Parts.lib* library, we need a *Parts.olb* library. It might also have to be tracked down with the File Explorer. However, if you have the *Parts.lib* library you can make the *Parts.olb* library as follows.

Highlight a part in the *Parts* schematic. Then using EDIT/PSPICE MODEL, select the menu FILE/CAPTURE PARTS. The CREATE PARTS FOR LIBRARY menu appears, as shown in Figure 20. Fill in the location of the *Parts.lib* library using the BROWSE button. Then hit OK.

[1] To keep track, it is best to use the SAVE AS option so you see where things are going. This problem can originate from failure to fill in the path in the FILE LOCATION box at the bottom of the NEW PROJECT menu. Then the default path is the last entry you made there.



PARTS.lib - PSpice Model Editor Lite - [Q2N800]

File Edit View Model Plot Tools Window Help

Models List

Model Name	Type	Creation
Q2N50	BJT	
Q2N800	BJT	

```
.model Q2N800   NPN(Is=14.34f Xti=3 Eg=1.11 Vaf=74.03 Bf=800 Ne=1.307
+                Ise=14.34f Ikf=.2847 Xtb=1.5 Br=6.092 Nc=2 Isc=0 Ikr=0
+                Cjc=7.306p Mjc=.3416 Vjc=.75 Fc=.5 Cje=22.01p Mje=.377
+                Tr=46.91n Tf=411.1p Itf=.6 Vtf=1.7 Xtf=3 Rb=10)
*                National        pid=19         case=TO18
*                88-09-07 bam    creation
```

Ready

Create Parts for Library

Enter Input Model Library: F:\304Fall\Copy&Paste\Parts\PARTS.lib

Enter Output Part Library: F:\304Fall\Copy&Paste\Parts\PARTS.olb

FIGURE 20:

Creation of the *Parts•olb* model library using the PSPICE model editor with FILE/CAPTURE PARTS. The *Parts.lib* library is filled in using the BROWSE button.

SETTING THE PARTS FILES TO READ ONLY

Using the File Explorer to right click on the file icons, select the PROPERTIES listing and set all the *Parts* files to READ ONLY. Otherwise, the automatic saving features of Orcad are likely to overwrite your parts, and you will have to make the library over again. We now describe how to use this *Parts* library to substitute parts by using the DESIGN CACHE. The reasons for using this procedure are discussed later. (back to Figure 16)

Substituting parts from the Parts library using the DESIGN CACHE

With the library available, we return to the current mirror schematic. We replace the fixed BF transistor with an NPN breakout part, QbreakN.

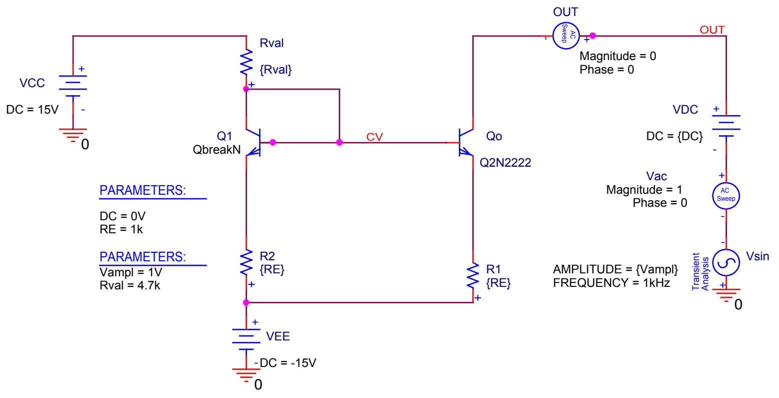


FIGURE 21:

Current mirror with breakout transistor QbreakN substituted for the original Q2N2222.

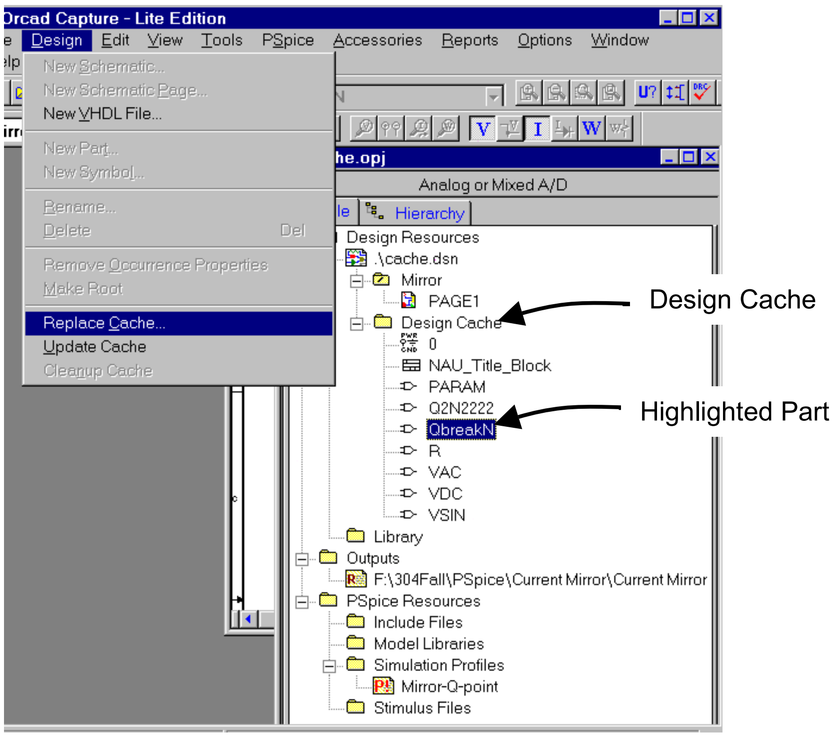
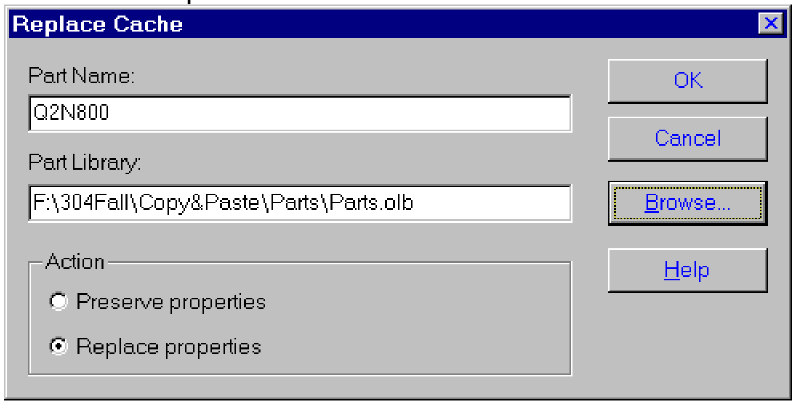


FIGURE 22:
Preparing to replace the breakout part with a library part from PARTS•OLB

Figure 22 shows how parts are introduced into the current mirror from the Parts library. The breakout part QbreakN is highlighted in the DESIGN CACHE of the •opj listing. Under the DESIGN menu, select REPLACE CACHE. This menu is filled in as shown in Figure 23. The REPLACE PROPERTIES option is selected.

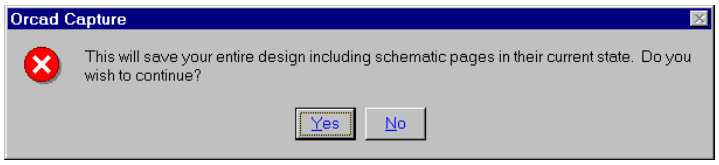

Replace Cache

Part Name: Q2N800

Part Library: F:\304Fall\Copy&Paste\Parts\Parts.olb

Action
Preserve properties [✓ Replace properties]

FIGURE 23:
The REPLACE CACHE menu filled out with the name of the part we want to substitute and the path to the *Parts•olb* library found using the BROWSE button


Orcad Capture

This will save your entire design including schematic pages in their current state. Do you wish to continue?

FIGURE 24:
Warning menu that appears when OK is clicked. Select YES.

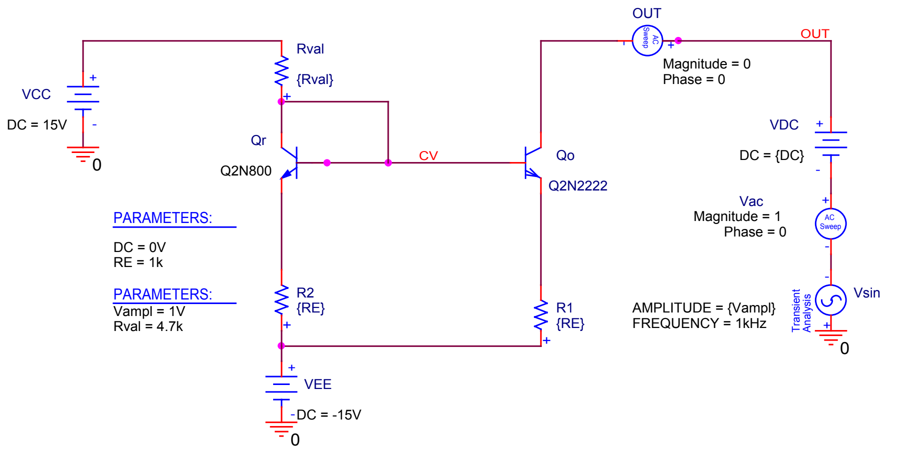
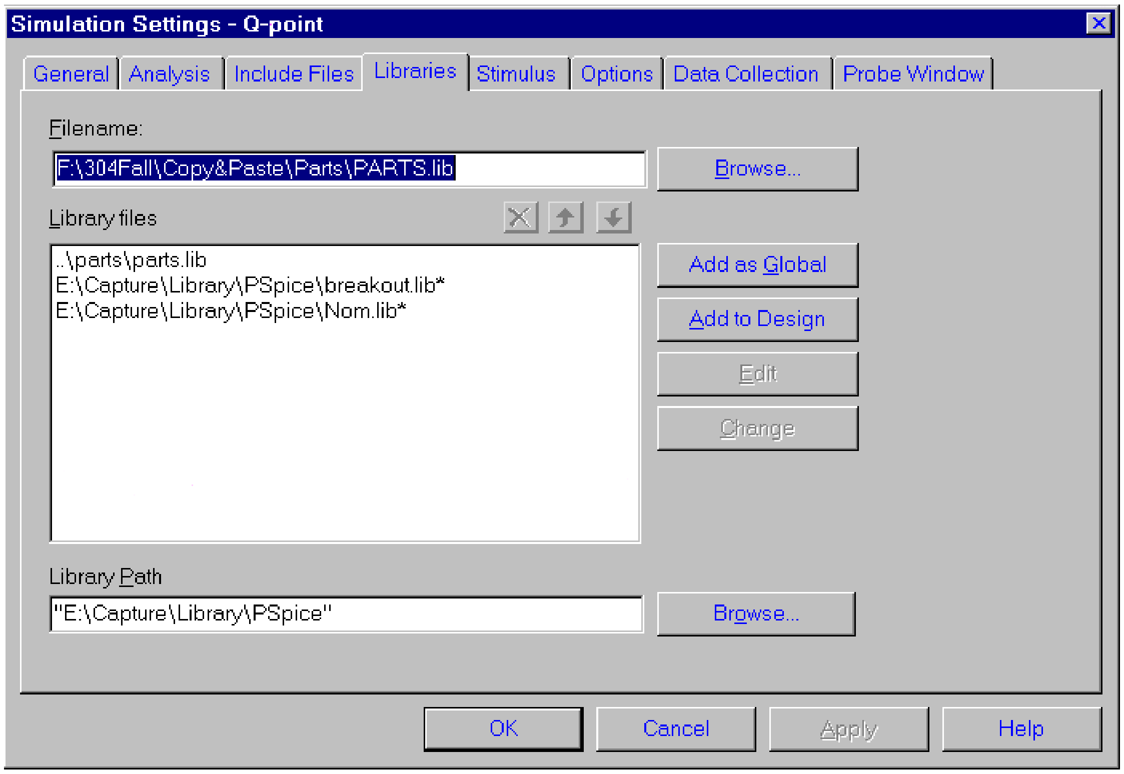


FIGURE 25:

New schematic. Using REPLACE CACHE, the breakout part QbreakN has been replaced with the Q2N800 part from *Parts.olb.*

Figure 25 seems to show that the part has been successfully inserted. However, EDIT/PSPICE MODEL says the part cannot be found! We have to add the *Parts.lib* library to the available library listing. Using PSPICE/EDIT SIMULATION PROFILE, we obtain the SIMULATIONS SETTINGS menu in Figure 26. Using the BROWSE button we find the library and select ADD TO DESIGN. Now, highlighting the part in the current mirror schematic and selecting EDIT/PSPICE MODEL finds the part listing for the Q2N800. The part really is there.



Filename:
F:\304Fall\Copy&Paste\Parts\PARTS.lib

Library files
..\parts\parts.lib
E:\Capture\Library\PSpice\breakout.lib*
E:\Capture\Library\PSpice\Nom.lib*

Library Path
"E:\Capture\Library\PSpice"

FIGURE 26:

Adding the *Parts.lib* to the library list for the current mirror.

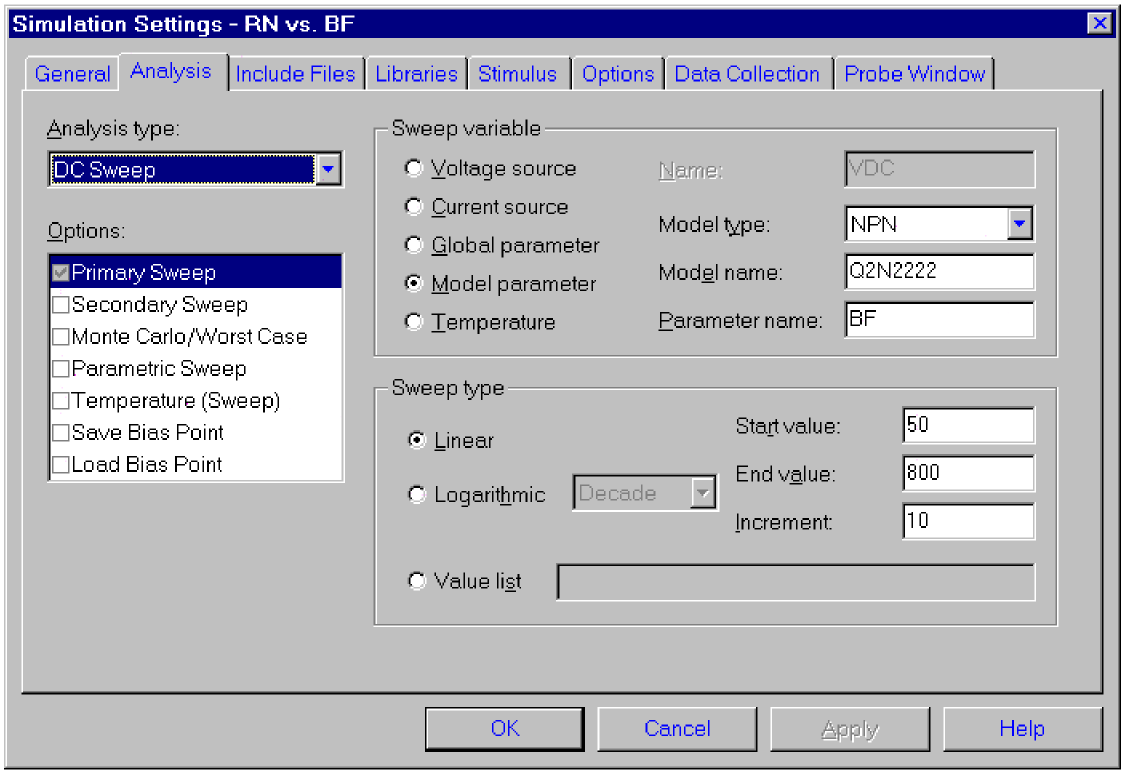
COMMENTS ON THE DESIGN CACHE PROCEDURE

The above process is only a little more complicated than copying the part in the Parts schematic, pasting it in the Mirror schematic, and adding *Parts.lib* to the library files list. If only one part replacement is to be made, it might not be worthwhile. However, for several replacements, the overhead is amortized over all the replacements and it's worthwhile. And, bear in mind, sometimes you underestimate the required number of replacements.

Less obvious, but more significant, is that multiple replacements often cause the simulator to lose track of the library path. Sometimes Orcad simply loses the library and the simulation is aborted. Sometimes Orcad simply refuses to use the newly inserted part, and instead uses the previous part. A warning is that the Q-point transistor data in the OUTPUT FILE lists the old model name. Often nothing will fix it, except starting over. Extremely annoying! Using the DESIGN CACHE is the safest way to keep the program on track, and to maintain the library paths so everything works. (back to Cache)

FINISHING THE CURRENT MIRROR EXAMPLE

Using the schematic of Figure 21, we now can look at the effect of BF on the mirror properties, for example, the output resistance. In PSPICE, model parameters for parts in the library *eval.lib* can be swept in DC sweep, but model parameters in the library *breakout.lib* cannot. By making the output transistor a Q2N2222 and entering its name in the MODEL NAME box in the menu of Figure 27, we enable the sweep of its BF parameter, while the BF of the breakout transistor doesn't change.



Simulation Settings - RN vs. BF

Analysis type: DC Sweep

Options:
- ☑ Primary Sweep
- ☐ Secondary Sweep
- ☐ Monte Carlo/Worst Case
- ☐ Parametric Sweep
- ☐ Temperature (Sweep)
- ☐ Save Bias Point
- ☐ Load Bias Point

Sweep variable: Voltage source [✓ Model parameter] Current source Global parameter Temperature

Name: VDC
Model type: NPN
Model name: Q2N2222
Parameter name: BF

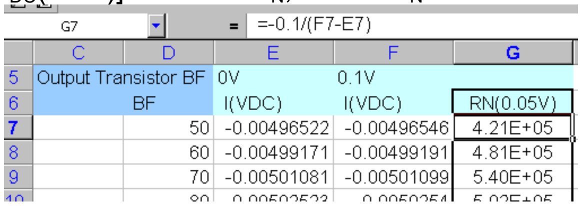
Sweep type: [✓ Linear] Logarithmic (Decade) Value list

Start value: 50
End value: 800
Increment: 10

FIGURE 27:
Setting up the sweep of the model parameter in the Q2N2222 output transistor

With the DC sweep of Figure 27, we can find the mirror current at one DC bias voltage, say DC=0V. Then this data is pasted into EXCEL. The simulation is run again at

DC=0.1V, and this data is pasted into EXCEL. Then the derivative is estimated as $dI_{DC}/dV_{DC} \approx [I_{DC}(0V) - I_{DC}(0.1V)]/0.1V \approx 1/R_N$, where R_N = Norton resistance.



G7 = =-0.1/(F7-E7)

	C	D	E	F	G
5	Output Transistor BF		0V	0.1V	
6		BF	I(VDC)	I(VDC)	RN(0.05V)
7		50	-0.00496522	-0.00496546	4.21E+05
8		60	-0.00499171	-0.00499191	4.81E+05
9		70	-0.00501081	-0.00501099	5.40E+05
10		80	-0.00502523	-0.0050254	5.92E+05

FIGURE 28:

Section of spreadsheet showing the formula used for the Norton resistance R_N. The columns I(VDC) are PSPICE output pasted into the spreadsheet.

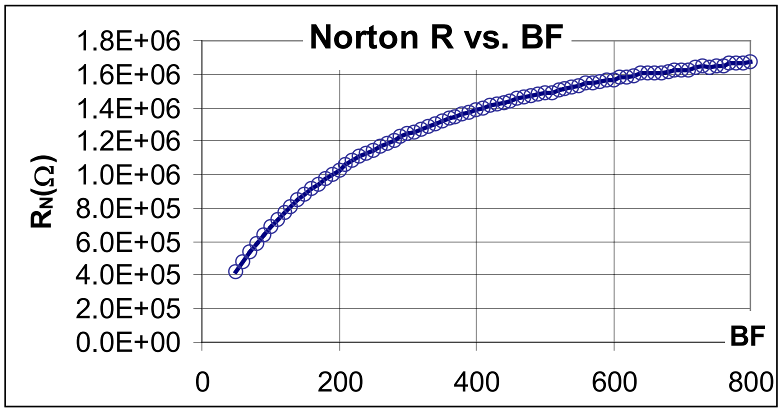


FIGURE 29:

Norton resistance *vs.* Q2N2222 parameter BF at a DC bias of DC≈ (0V+0.1V)/2 = 0.05V

You might expect that PSPICE could generate such a plot using multiple sweeps, but when a <u>model</u> parameter is swept, the secondary sweeps are disabled in PROBE. So, using EXCEL is the best way to get Figure 29.

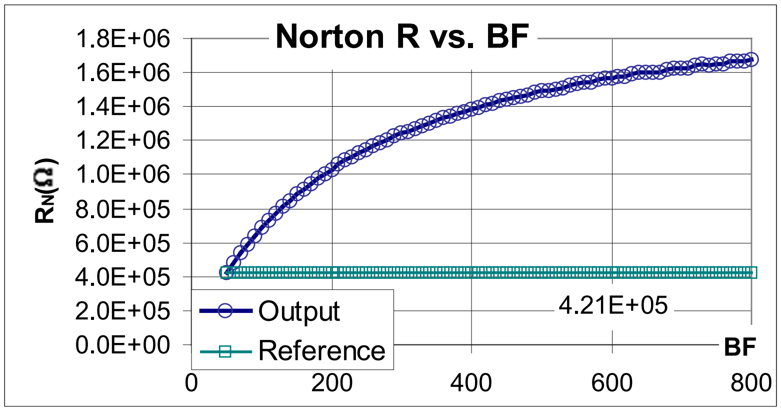


FIGURE 30:

Figure 29 with an added curve to show the Norton resistance when BF of the reference transistor QR is varied and the output transistor is Q2N50, the low-β transistor. The two curves agree for BF=50.

Figure 30 shows the result of the simulation when a Q2N2222 is used as the reference transistor and the output transistor is the low-β Q2N50. Evidently, the β-value of the reference transistor is immaterial: the Norton resistance is not dependent on the reference transistor's β-value.

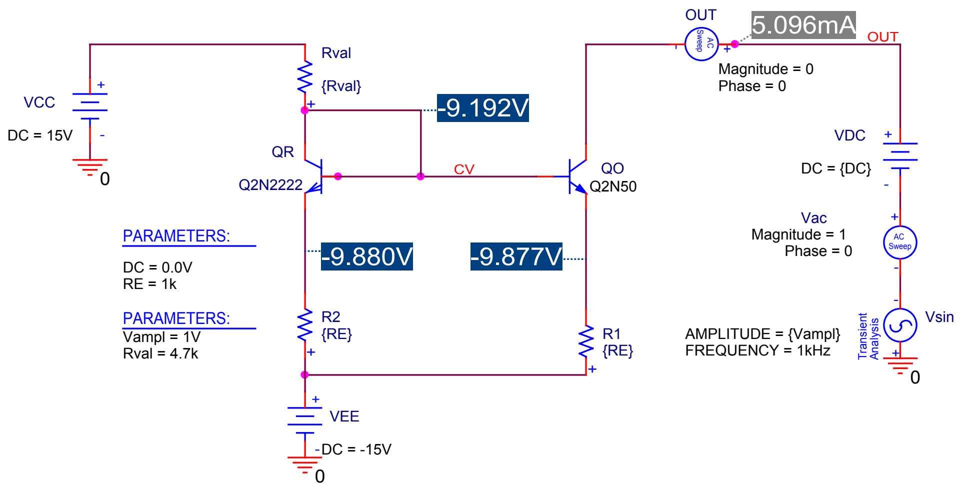


FIGURE 31:

Schematic with the transistors rearranged. The DESIGN CACHE was replaced using REPLACE CACHE (see Figure 23) following the insertion of a breakout part QbreakN for Q_O and a Q2N2222 for Q_R.