Tutorial

2

SAN JOSE STATE UNIVERSITY

Electrical Engineering Department

Bottom up IC design-flow Using an Analog Leaf Cell

IC DESIGN GROUP SAN JOSE STATE UNIVERSITY

A tutorial guide for using CDS tools for IC design in Leaf Cell

Shao Ng
Eric Basham
David W. Parent
Electrical Engineering, SJSU
One Washington Square
San Jose, CA 95192-0084
Phone 408.924.3963 • Fax 408.924.2925

Table of Contents

LIST OF FIGURES:

Acknowledgements:

This tutorial is based on the NCSU design kit. For more information, see http://www.ece.ncsu.edu/cadence/CDK.html.

This tutorial also follows the design flow used by WPI at http://vlsi.wpi.edu/cadence/.

The Leaf Cell is designed and layout by Eric Basham.

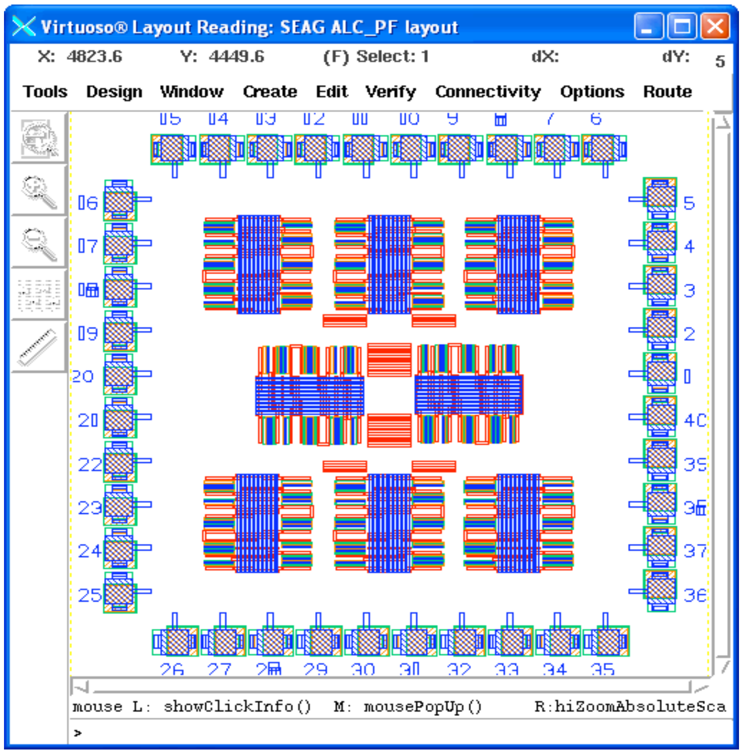
Chapter I

Chapter I: Introduction

Full custom IC design is very time consuming. In order for students to complete the whole custom IC design flow--design, fabrication, and testing within a regular semester class, we had design the Analog-leaf-cell. The ALC is a sea-of-gates semi-custom IC design approach that reduces design and fabrication time. The design time is reduced because the transistors are pre-placed and the novice designer only has to route one metal layer, and fabrication time is reduced because the transistors are pre-fabricated and only one metal layer has to be processed to complete the manufacture of an IC design. In this tutorial, we will show you the design of the Leaf-cell as well as a tutorial for simple analog design.

1. The Leaf Cell:

The leaf cell is designed to facilitate analog circuit design for Junior/Senior students who already have taken an introduction to current design and a solid state physics course in our Cadence IC design lab. It is intended as a teaching tool to introduce the student to the design, fabrication and test cycle without requiring extensive fabrication and layout resources. Additionally, general concepts of reconfigurable devices may be addressed during the design cycle.


Virtuoso® Layout Reading: SEAG ALC_PF layout
Tools Design Window Create Edit Verify Connectivity Options Route
15 14 13 12 11 10 9 8 7 6
16 17 18 19 20 21 22 23 24 25
5 4 3 2 1 40 39 38 37 36
26 27 28 29 30 31 32 33 34 35

```
X: 4823.6    Y: 4449.6    (F) Select: 1    dX:    dY: 5
mouse L: showClickInfo()   M: mousePopUp()   R:hiZoomAbsoluteSca
>
```

Figure 1: Leaf Cell in Pad Frame

There are Leaf Cells in a Pad Frame (Figure 1). Figure 2 shows one laid out Leaf Cell. There are 8 N transistors (bottom) and 8 P transistors (top). They are laid out in a predetermined, but not connected manner in order to guide your design and reduce the level of required layout for a functional design. The blue lines are the metal connectors and the red lines are poly connections.

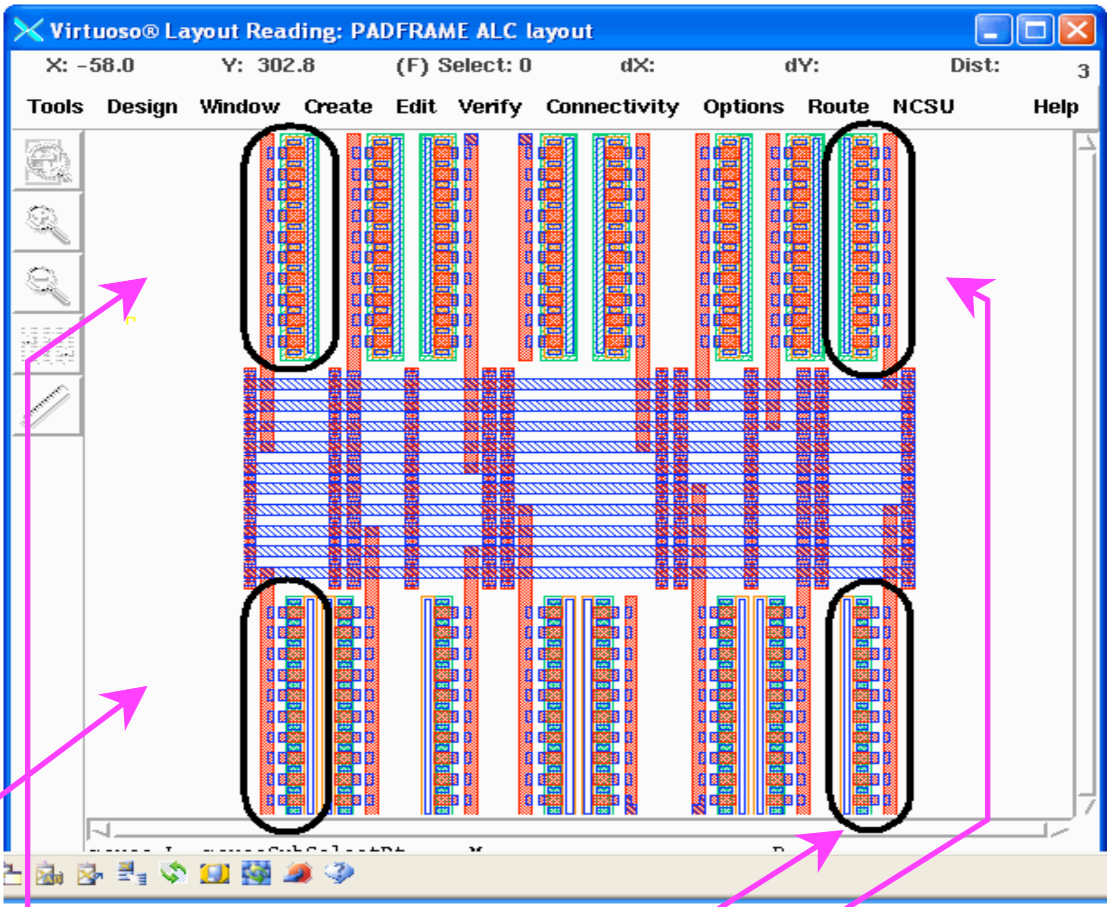


Figure 2: Leaf Cell block

Accordingly, Figure 3 shows the blank wiring diagram (bubble schematic) of a block of Leaf Cell.

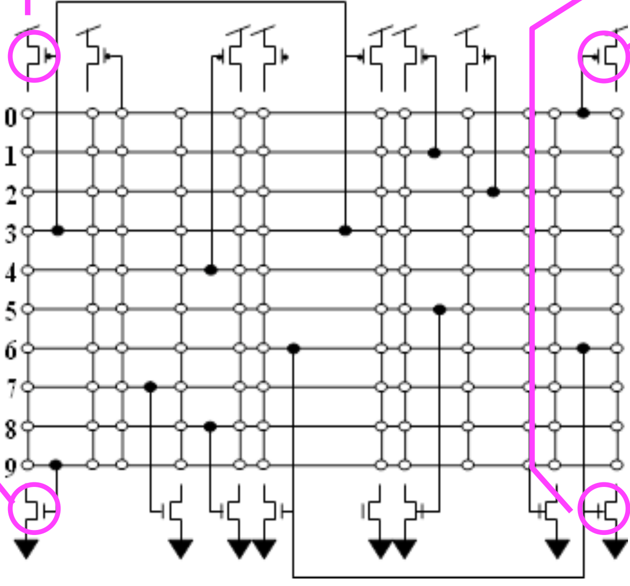
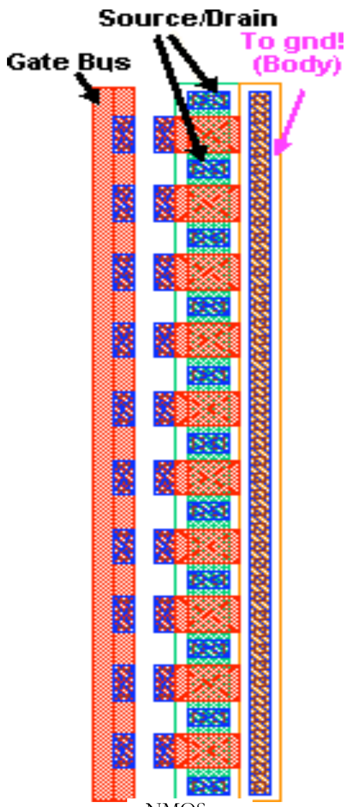


Figure 3: Bubble Schematic

Each empty bubble represents a connection that can be made between the wires that run vertically and the wires that run horizontally; each solid bubble is a hard wired connection. Transistor gates are connected to an individual wire running horizontally by a short wire running vertically. Since there are two sets of transistors with gates connected to the same horizontal bus, there are 5 horizontal bus lines for P transistors (0-4) and 5 horizontal bus lines for N \transistors (5-9). As part of the design process you will have to determine which horizontal busses are inputs, outputs, power and ground.

How to size:



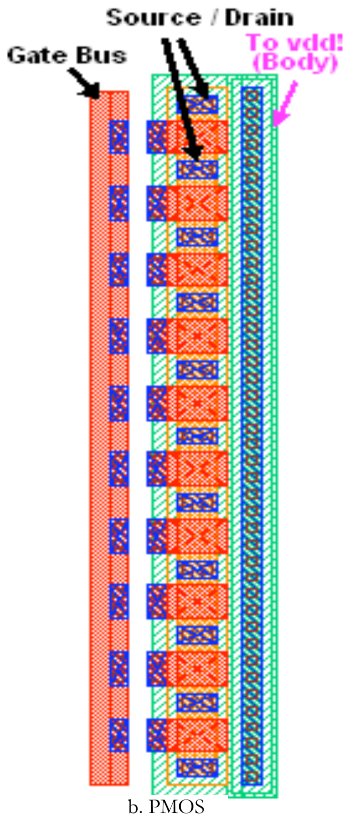
a. NMOS

Figure 4: A Single Transistor

Figure 4 shows a bank of 10 transistors with W=L=6.4 μm that can be configured in series or parallel. The contacts are predefined and covered with metal to allow you to make connections between the poly layers. The metal bus on the left is Source and the right bus is the drain.

Sizing is really as simple as shorting the metal contact connections together to make either transistor in parallel or transistors in series.

Figure 5 shows some examples of how to perform sizing on PMOSs. In this case all the transistors' sources are connected to Vdd (PMOS) and the outputs are the drain bus on the left, but this is not the only method of connection. The thin green lines show the transistors, the thick blue lines show the metal connections you will add.



b. PMOS

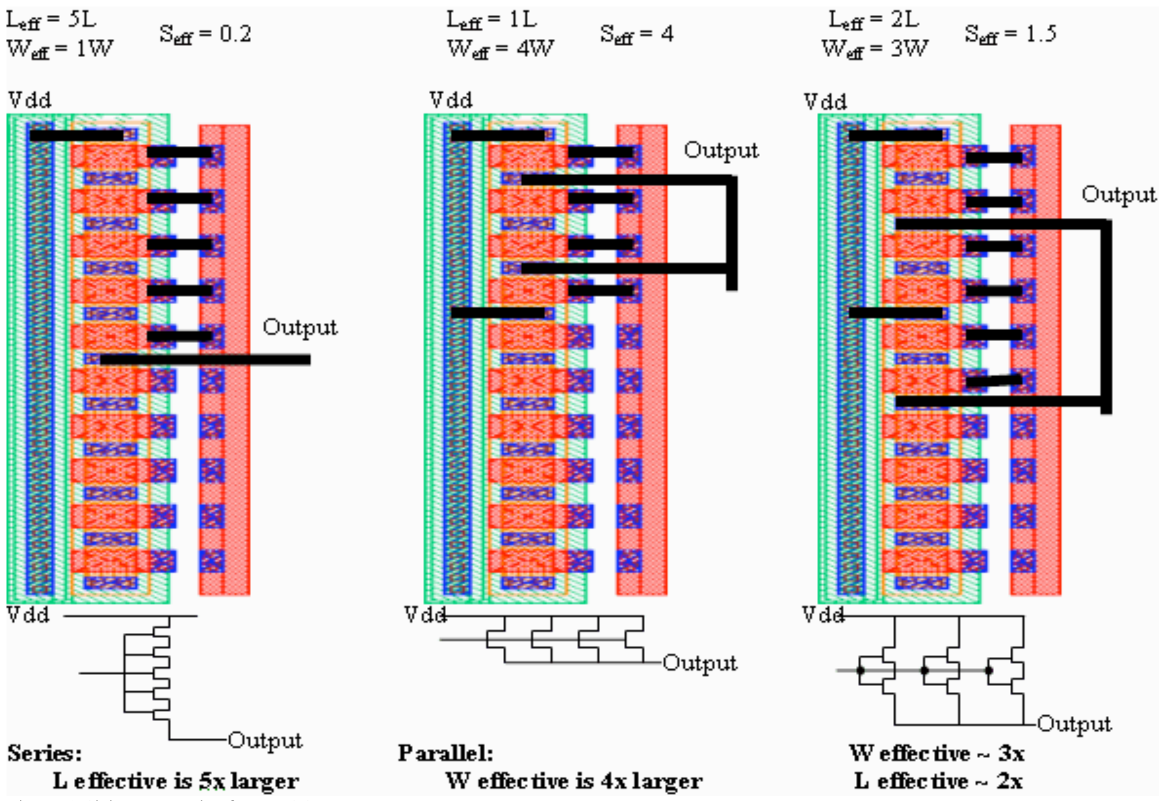


Figure 5: Sizing Examples for PMOSs.

2. Leaf Cell Parameters:

Since the Leaf Cell is designed for analog, the minimum sizes of W and L of both the PMOS and NMOS are designed to be the same, which is **6.4** micro meter. In this way, the ratio S = W / L can be easily determined and then laid out. Table 1 lists some of the important parameters of the Leaf Cell for reference.

Table 1: Leaf Cell Parameters (Tech File: AMI 1.6):

Parameter (unit)	Value
W_N Minimum (m)	$6.4\text{x}10^{-6}$
L_N Minimum (m)	$6.4\text{x}10^{-6}$
TOX (m)	$3.04\text{x}10^{-8}$
μ_N (cm^2/Vs)	640.04
CGDO (F/m)	$2.7\text{x}10^{-10}$
CGSO (F/m)	$2.7\text{x}10^{-10}$
CJSW (F/m)	$1.465\text{x}10^{-10}$
CJ (F/m^2)	$2.806\text{x}10^{-4}$
VT (V)	0.58
Length of Source or Drain (m)	$4.8\text{x}10^{-6}$
K' / 2 (uA/V^2)	36.95

Parameter (unit)	Value
W_P Minimum (m)	$6.4\text{x}10^{-6}$
L_P Minimum (m)	$6.4\text{x}10^{-6}$
TOX (m)	$3.04\text{x}10^{-8}$
μ_P (cm^2/Vs)	268.45
CGDO (F/m)	$2.7\text{x}10^{-10}$
CGSO (F/m)	$2.7\text{x}10^{-10}$
CJSW (F/m)	$1.464\text{x}10^{-10}$
CJ (F/m^2)	$2.960\text{x}10^{-4}$
VT (V)	-0.806
Length of Source or Drain (cm)	$4.8\text{x}10^{-6}$
K' / 2 (uA/V^2)	-15.50

3. Design Flow

The design flow for using the Leaf Cell deviates slightly from the normal bottom-up IC design flow in that the transistors have already been laid out in a predetermined pattern. Only metal1 needs to be drawn upon the Leaf Cell to finish any designs. This is called a semi-custom approach.

Initial Design

In the initial design phase, you decide the specification of the circuit. You then size the widths and lengths of the transistors by ratios as needed for your analog design. You would also draw the circuit schematic out on paper and develop the test vectors to prove that your design will work.

Schematic Capture

You then enter in the schematic and symbol for your circuit, and create a test bench for simulation. You simulate the circuit to make sure it functions properly, and it meets the time and power specifications. Since analytical equations are not as accurate as spice simulations, you might need to change the widths (in this case, we change the S ration) of the PMOS and NMOS transistors for you to meet your specifications.

Pre - Layout

Once the schematic has met your specifications, you use the leaf cell bubble schematic to plan how to wire the circuit together. This is the critical step because you need to look at the leaf cell tree and design which N/P MOS you want to use, as well as the joints that need to be connected.

Layout

Referring to your bubble graph plan, in Cadence layout tools you draw out how the circuit would look under a microscope. These pictures are used to make the photolithography masks that are used to define your circuit on silicon. In this design flow, you may only add metal1.

Design Rule Checking

Once the circuit is laid out, you have to complete a design rule check to make sure that the circuit will not have yield problems when it is fabricated.

Circuit Extraction

Once you have laid out a circuit you need to extract its electrical properties to make sure that you drew the correct functionality and to estimate the parasitic resistances and capacitances that degrade circuit performance. The extracted view can be sent to the simulator.

Layout versus Schematic

Once you have an extracted view of your circuit, you need to run a layout versus schematic check. This makes sure that the electrical properties of your schematic match those of your extracted view. This is faster than running all your test vectors on the extracted view that you did on the schematic view.

Post Extraction Simulation

Once you make sure that the circuits are equivalent, you run the simulation again using the extracted view. This will take into account parasitic caps. You might have to change the widths of your transistors slightly to match your specification. The problem at this point is that you have to change the drawing now to change the transistor and then re-extract the circuit to finally meet your spec. Then you have to go back and change your schematic as well. The better job you do at predicting circuit performance at the initial design stage and schematic capture stage, the less re-work you have to do at the layout stage.

Fabrication and Test

For the final step, fabrication test, the leaf cell can be fabricated first in the lab prior to the completion of the students' designs. After the students designs are completed, the last mask, which is only the mask of the wire connection, can then be finished up for testing.

Once the circuit comes back from fabrication, you test it using the test vectors you developed earlier. If the circuit does not meet specification you have to start the process at the beginning using the feedback you received from the actual devices that were made to modify your design.

4. Applications:

In the following two Chapters, we will go through two semi-custom analog designs, a simple current mirror array for a current steered DAC, and a simple OTA (Operational Transconductance Amplifier).

Design 1: Current Mirror:

Current mirrors are widely used in analog integrated circuits. They are used as biasing elements to minimize the dependencies of circuit performance to the variation of power supply and temperature; they are also used as load devices for amplifier stages to result in high voltage gain at low supply voltage. Ideally, the output current of a current mirror is equal to the input current multiplied by a desired current gain. Is it true in practice? Unfortunately, you will notice it is not true in the simulation part of the tutorial.

Figure 6 shows a 4-bit current steered DAC. It has a current mirrors that have binary weights assigned to each current mirror (used as a source) corresponding to the position of the bit. Therefore, a 4 bit DAC would have 4 current mirrors with the current equals to its bit position times the reference current. A step by step semi-custom design of this current mirror will be shown in Chapter 2.

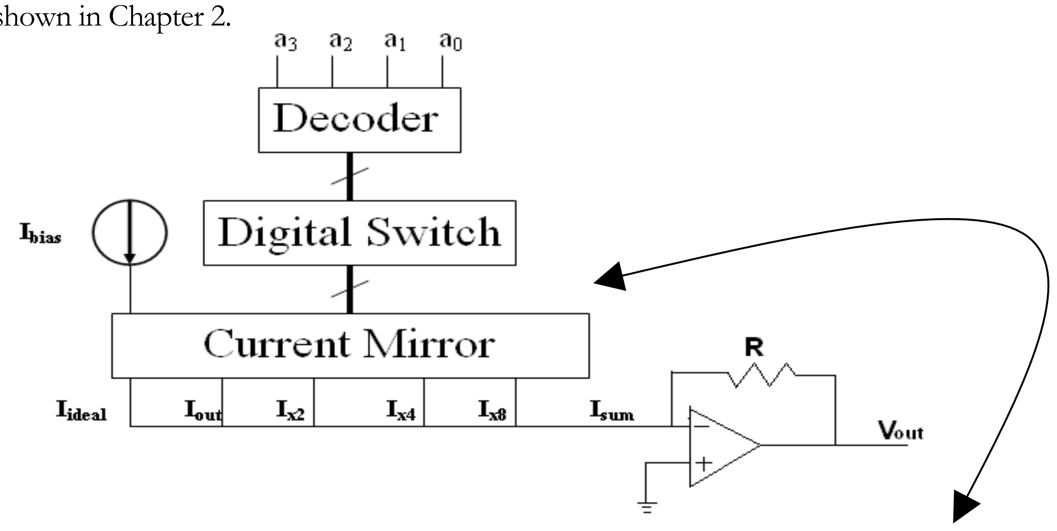
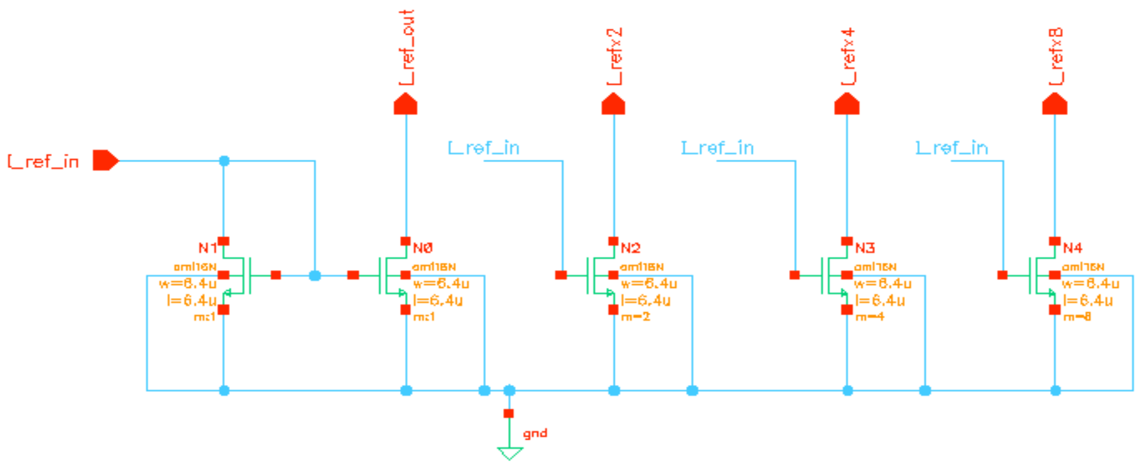


Figure 6: A simple current mirror array for a current steered DAC.



Design 2: OTA:

The operational transconductance amplifier (OTA) is basically an op-amp without an output buffer, so that it can only drive capacitive loads. OTA can also be defined as an amplifier where all nodes are low impedance except the input and output nodes.

OTAs are very useful in CMOS analog circuit design. There are two practical concerns when designing an OTA for filter applications. One is the input signal amplitude. Large signals can cause the OTA gain to become nonlinear. Another is the parasitic input/output capacitances. The external capacitance should be large compared to the input/output parasitic therefore it limits the maximum frequency of the filter and it causes amplitude or phase error which can usually be tuned out with proper selection of I_{BIAS}.

Figure 7 shows a simple OTA, and we will see how it works out in Chapter 3.

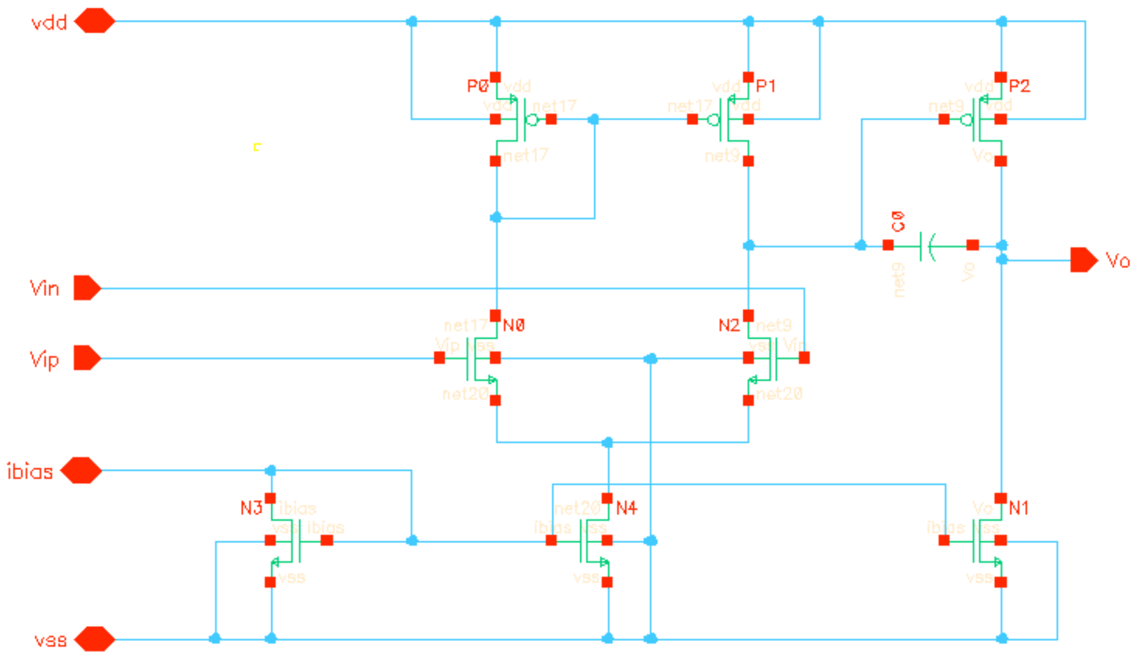


Figure 7: A simple OTA (Operational Transconductance Amplifier).

Chapter
2

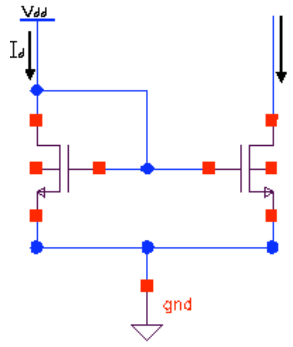
Chapter 2: A Current Mirror Tutorial

Section 1: Initial Design

Design specification:

The steered currents need to be within 5% of the multiples of the reference current. Vdd is 2.5V.

Hand Calculation:



In this case, since the ratios of W and L are integers, and Vdd is fixed, the only element we can control is Id. In order to have the steered currents within 5% error, we need to do some calculation to pick the right I_{bias}.

For CMOS, the current equation is:

$$I_d = \frac{\mu_n C_{ox}}{2} \bullet \frac{W}{L} \bullet \left(V_{gs} - V_t\right)^2$$

Plugging in the parameters, we have Id = 79.52858μA. Therefore, we choose Id=80 μA, plugging into the following equation:

$$V_{dd} = V_{gs} = V_t + \sqrt{\frac{\text{Id}}{\text{K}' \bullet \text{W/L}}}$$, we have:

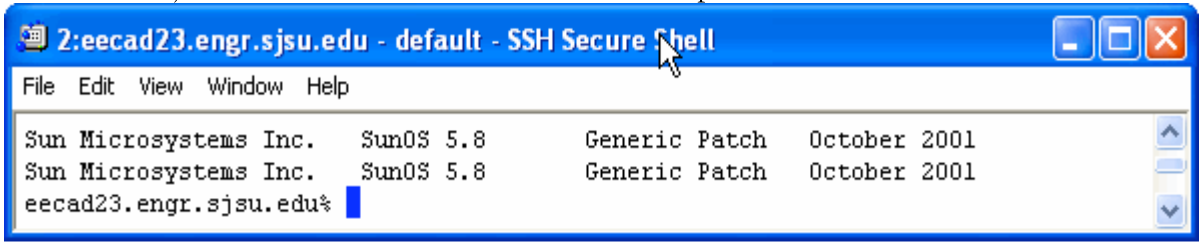
W/L	1	2	4	8
Id (μA)	80	160	320	640
Vgs	2.505653	2.505653	2.505653	2.505653
error	0.23%	0.23%	0.23%	0.23%

Getting started with the Tool:

Open the Terminal:

Your account should have the all paths set to run the software. You need only to log in and start a terminal. There should be a terminal icon on your Common Desktop Environment. (If you can't find it, you can right click the mouse and look for *Terminal* to open it.)

Just double click on it and a command window will appear (Figure 8.). You type commands in this window just like an MSDOS command line except that the commands are different.

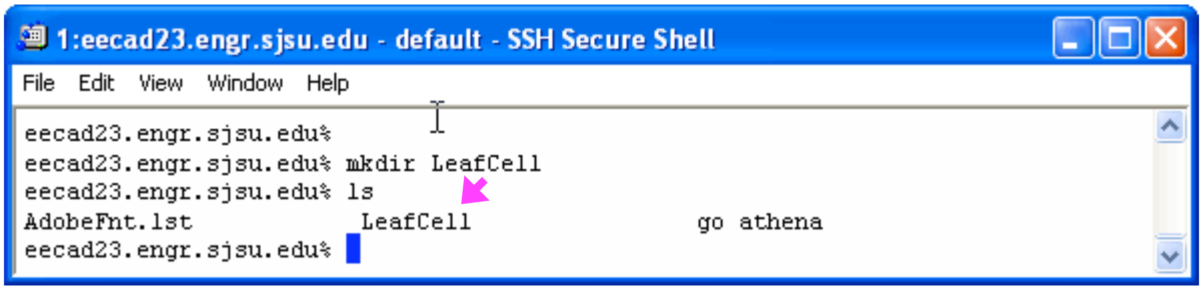

2:eecad23.engr.sjsu.edu - default - SSH Secure Shell
File Edit View Window Help

```
Sun Microsystems Inc.   SunOS 5.8       Generic Patch   October 2001
Sun Microsystems Inc.   SunOS 5.8       Generic Patch   October 2001
eecad23.engr.sjsu.edu%
```

Figure 8: The terminal

Create a Project Directory:

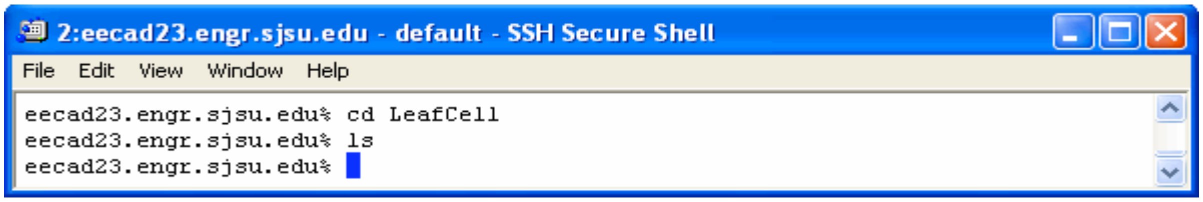
To do this, type the command *mkdir LeafCell* at the command line (Figure 9). This command makes a directory LeafCell in your home directory. **You only have to create this directory once!** Type the command *ls*. You should see a directory with the name just created as in Figure 9. The command *ls* lists out the content of your present working directory.


1:eecad23.engr.sjsu.edu - default - SSH Secure Shell
File Edit View Window Help

```
eecad23.engr.sjsu.edu%
eecad23.engr.sjsu.edu% mkdir LeafCell
eecad23.engr.sjsu.edu% ls
AdobeFnt.lst             LeafCell                 go athena
eecad23.engr.sjsu.edu%
```

Figure 9: Making a directory and listing the contents of a directory.

In order to start the CDS tools so that your project files are available, you need to go into your project directory before starting the tools. Type in the command *cd LeafCell* at the command line. This changes your present working directory to *LeafCell*. If you type the command *ls*, you should see no files inside that directory (Figure 10.).

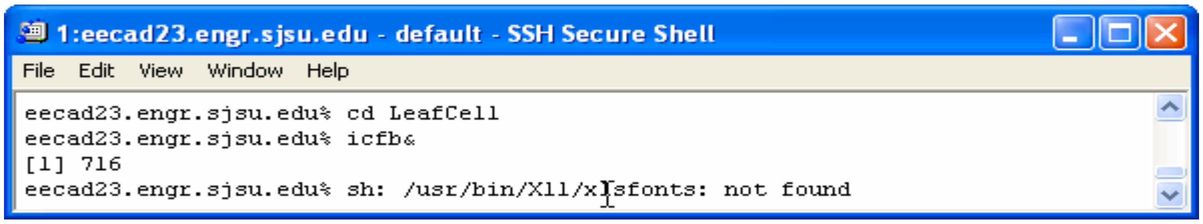

2:eecad23.engr.sjsu.edu - default - SSH Secure Shell
File Edit View Window Help

```
eecad23.engr.sjsu.edu% cd LeafCell
eecad23.engr.sjsu.edu% ls
eecad23.engr.sjsu.edu%
```

Figure 10: Changing your present working directory.

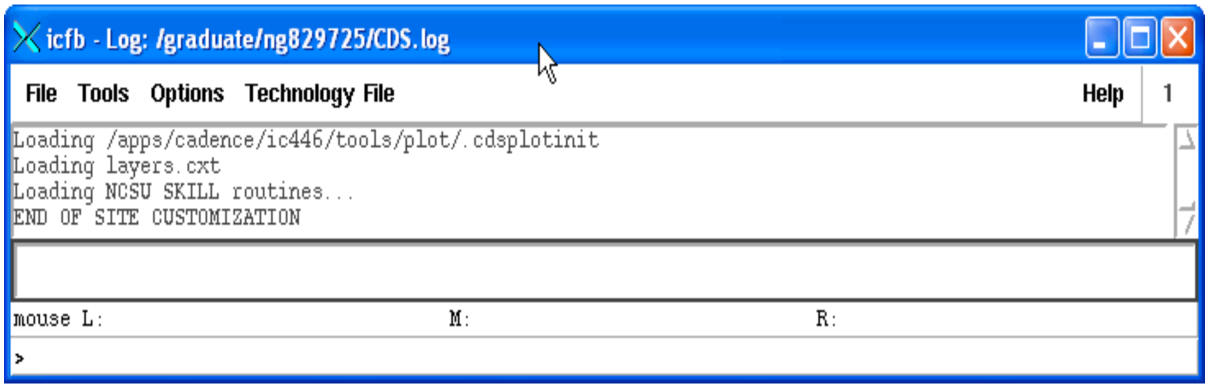
Start CDS Tools:

Type in the command *icfb &*. You should see messages similar to Figure 11. After some time, the *CIW* (Command Interface Window) will pop up (Figure 12). Once the CIW come up, you will not need to use the command line.


1:eecad23.engr.sjsu.edu - default - SSH Secure Shell
File Edit View Window Help

```
eecad23.engr.sjsu.edu% cd LeafCell
eecad23.engr.sjsu.edu% icfb&
[1] 716
eecad23.engr.sjsu.edu% sh: /usr/bin/X11/xIsfonts: not found
```

Figure 11 : Starting CDS tools (icfb---IC Front to Back).


icfb - Log: /graduate/ng829725/CDS.log
File Tools Options Technology File
Help 1

```
Loading /apps/cadence/ic446/tools/plot/.cdsplotinit
Loading layers.cxt
Loading NCSU SKILL routines...
END OF SITE CUSTOMIZATION
```

```
mouse L:                M:                R:
>
```

Figure 12: The CIW window.

Summary of Starting up CDS Tools:

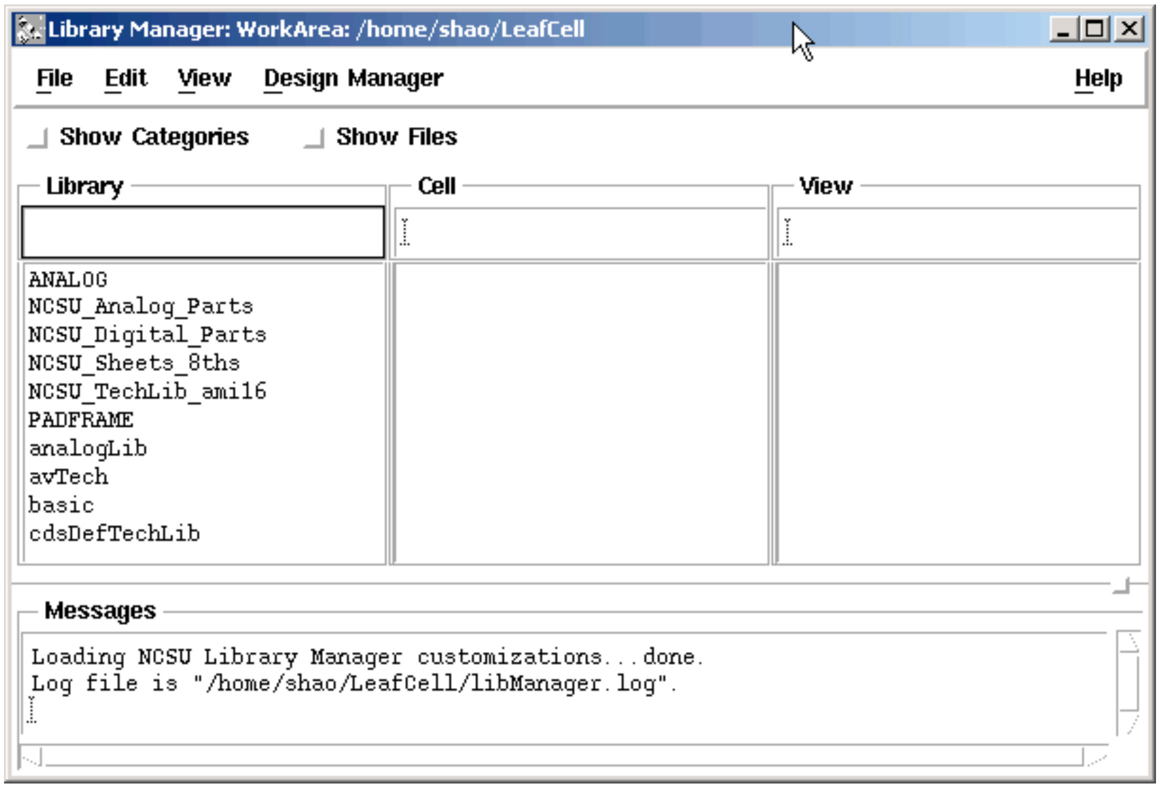
Now that you have created your project directory, whenever you want to work on the design in that directory you just have to:

1. Log in
2. Start a terminal
3. Type in the command**, *cd LeafCell*** (or the name you selected for your project)
4. Type in the command, ***icfb &***.

Section 2: Getting started with Schematic Capture and Spice Simulation

Design entry through schematic capture:

We will be using the NCSU design kit which automatically starts the library manager (Figure 13.) You should see several NCSU libraries (*NCSU_Analog_Parts*, *NCSU_Digital_Parts*, and *NCSU_Sheet_8ths*), as well as libraries named *ANALOG, analogLib, avTech, basic*, *cdsDefTechLib*, and PADFRAME



```
Loading NCSU Library Manager customizations...done.
Log file is "/home/shao/LeafCell/libManager.log".
```

Figure 13: Library Manager

Creating a new library:

We need to create a project library and attach a technology library to it.

To create your new project library:

1. Go to the *icfb* window (or library manager window), and go to: *File... New...Library* (Figure 14). A pop-up window should appear.

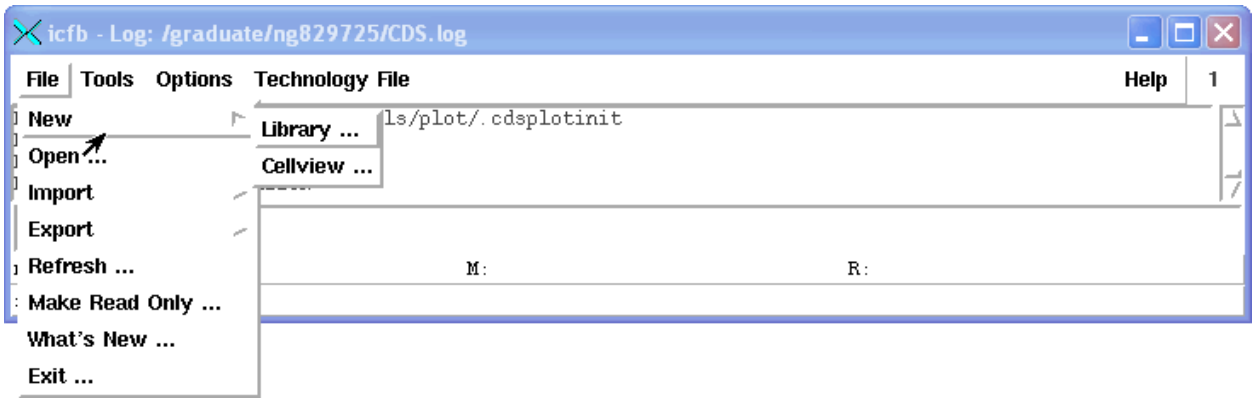
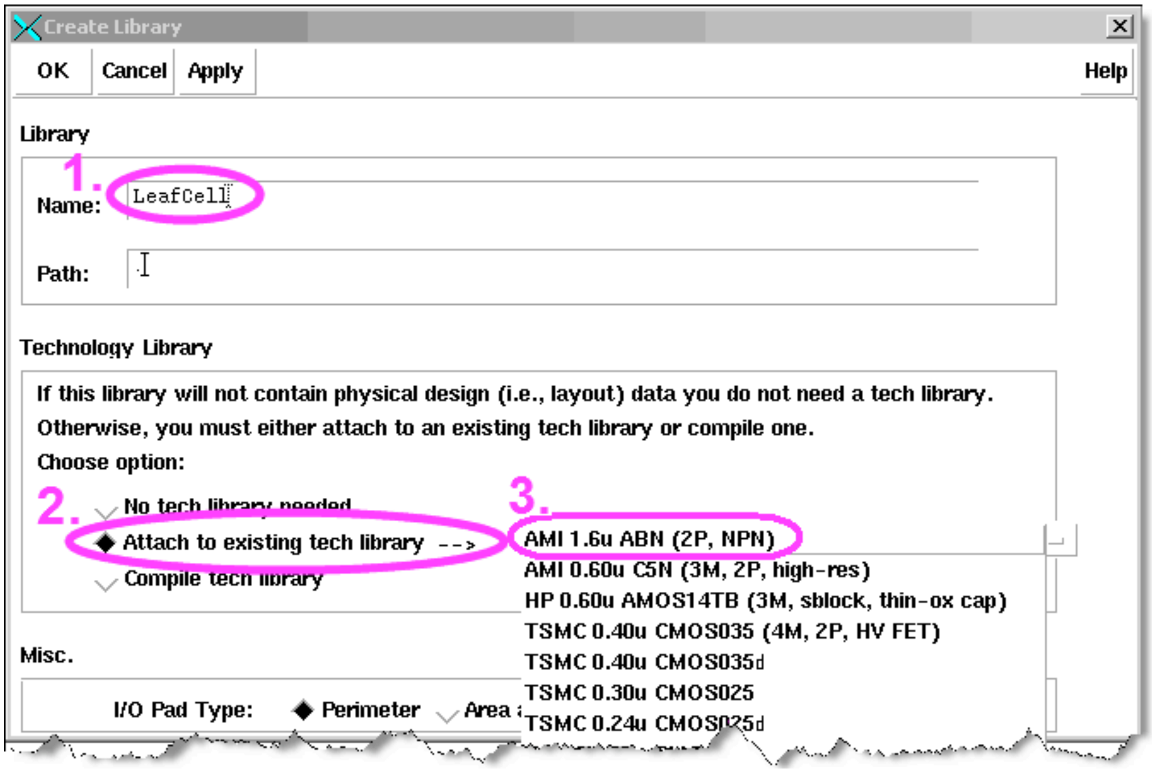


Figure 14: Where to go for the new project directory.

2. Fill out the form exactly according to Figure 15. Make sure to click on *attach tech library*.


Create Library
OK
Cancel
Apply
Help
Library
1.
Name:
LeafCell
Path:
Technology Library
If this library will not contain physical design (i.e., layout) data you do not need a tech library.
Otherwise, you must either attach to an existing tech library or compile one.
Choose option:
2.
No tech library needed
Attach to existing tech library -->
Compile tech library
3.
AMI 1.6u ABN (2P, NPN)
AMI 0.60u C5N (3M, 2P, high-res)
HP 0.60u AMOS14TB (3M, sblock, thin-ox cap)
TSMC 0.40u CMOS035 (4M, 2P, HV FET)
TSMC 0.40u CMOS035d
TSMC 0.30u CMOS025
TSMC 0.24u CMOS025d
Misc.
I/O Pad Type:
Perimeter
Area

Library Name: LeafCell

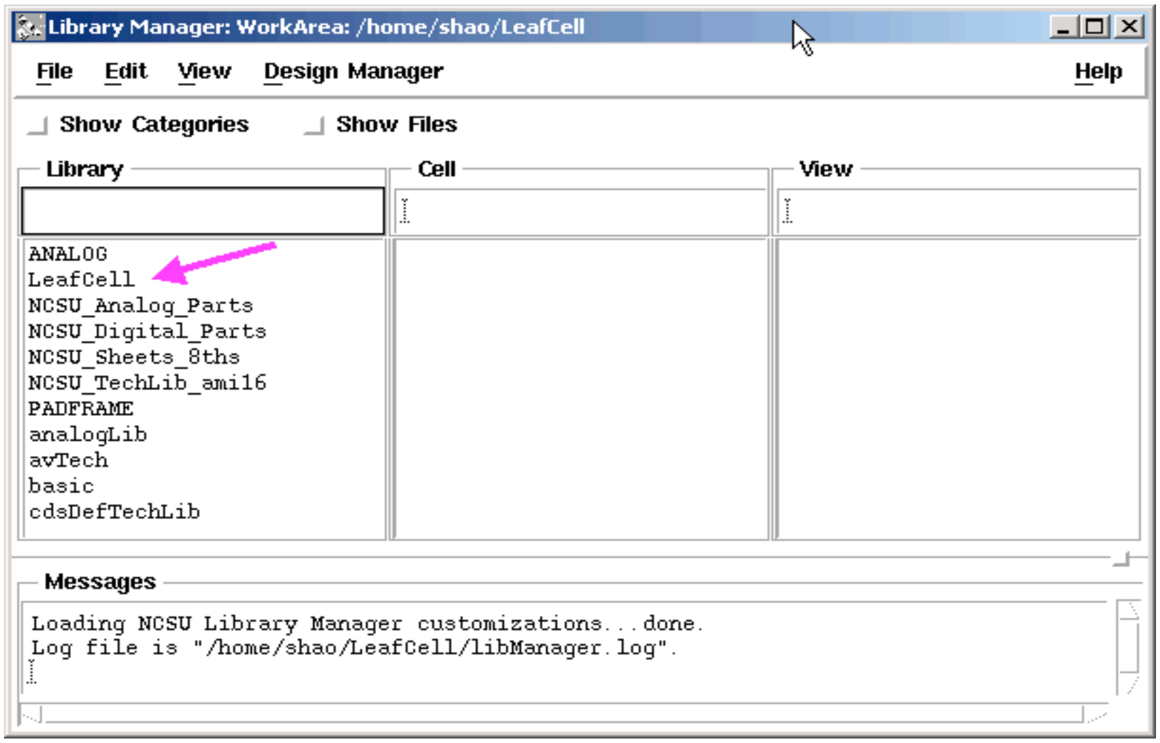
Technology Library: If this library will not contain physical design (i.e., layout) data you do not need a tech library. Otherwise, you must either attach to an existing tech library or compile one.

Choose option: No tech library needed / [✓ Attach to existing tech library] --> [✓ AMI 1.6u ABN (2P, NPN)] / Compile tech library

I/O Pad Type: [✓ Perimeter] Area

Figure 15: Creating a Library

3. Find the "*AMI 1.6u ABN (2P, NPN)*" tech library, choose it and click *OK*.
4. Your library manager should now show your new project directory as in Figure 16.



```
ANALOG
LeafCell
NCSU_Analog_Parts
NCSU_Digital_Parts
NCSU_Sheets_8ths
NCSU_TechLib_ami16
PADFRAME
analogLib
avTech
basic
cdsDefTechLib
```

```
Loading NCSU Library Manager customizations...done.
Log file is "/home/shao/LeafCell/libManager.log".
```

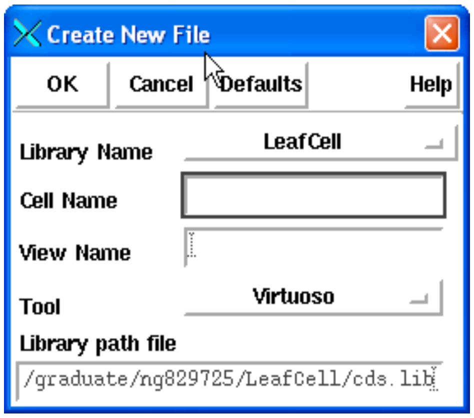
Figure 16: Updated library manager showing LeafCell project directory.

Creating a schematic view:

After creating the library LeafCell, we can now start adding views into it. Each design will have different views. For example, one set of views includes a schematic, symbol, layout, and extracted views. There can be more depending on the application.

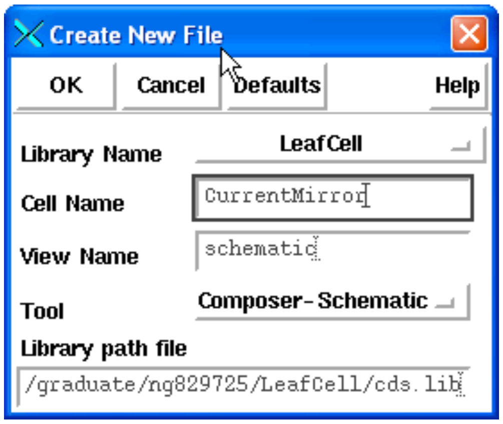
To create a schematic view:

1. Go to the library manager window, click at the *LeafCell* library, and then go to: F*ile… New… Cell View*. A pop-up like Figure 17 should appear.



Library Name	LeafCell
Cell Name	
View Name	
Tool	Virtuoso
Library path file	/graduate/ng829725/LeafCell/cds.lib

Figure 17: Pop-up: Creating a new cell view



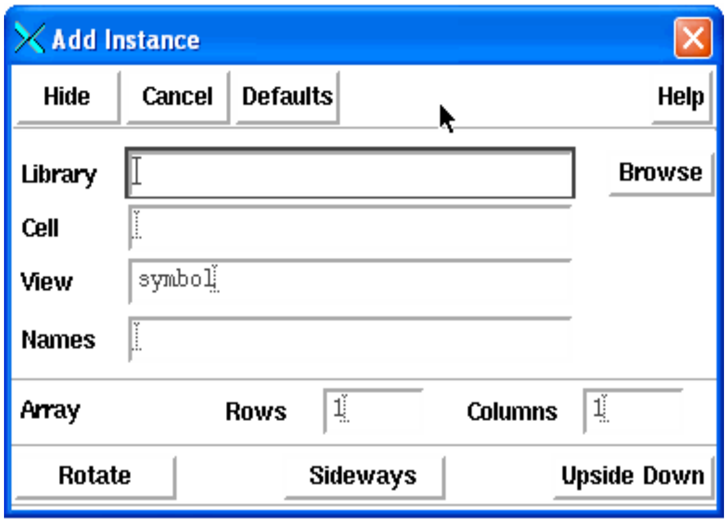
Library Name	LeafCell
Cell Name	CurrentMirror
View Name	schematic
Tool	Composer-Schematic
Library path file	/graduate/ng829725/LeafCell/cds.lib

Figure 18: Creating a new schematic view

2. Fill out the form exactly according to Figure 18.
3. Click *OK*. The *Virtuoso Schematic Editing* window should appear.

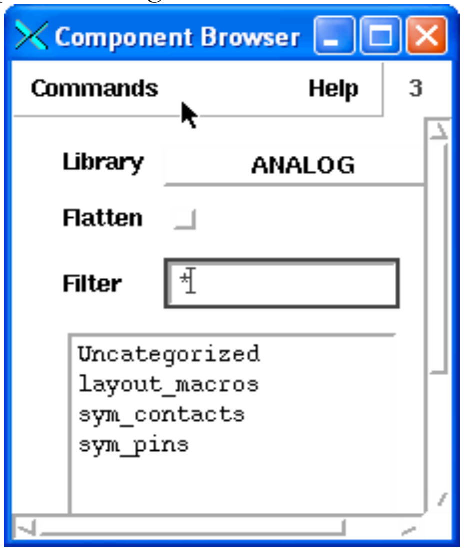
Adding the Transistors:

To add instance to the schematic, go to A*dd… Instance* in the schematic window, or press the letter 'i' key from your keyboard. The *Add Instance* window appears as in Figure 19.



Library	
Cell	
View	symbol
Names	
Array	Rows 1 Columns 1

Figure 19: Adding an instance



Library	ANALOG
Flatten	
Filter	*

```
Uncategorized
layout_macros
sym_contacts
sym_pins
```

Figure 20: Browsing for components

Click on *browse* to graphically get components. A pop-up like Figure 20 should appear.

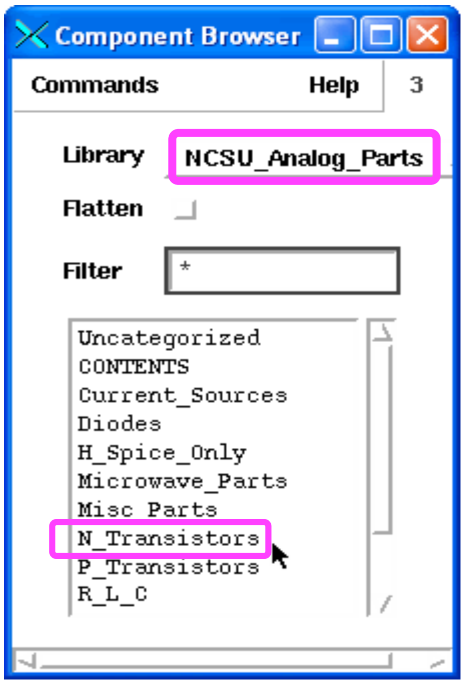


Figure 21: Choosing the right folders

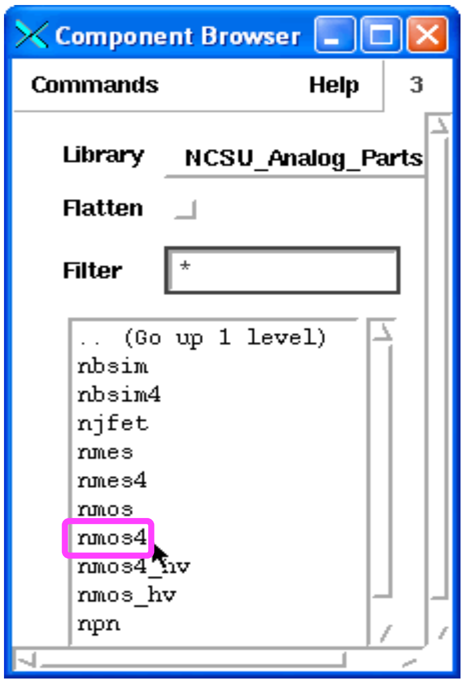
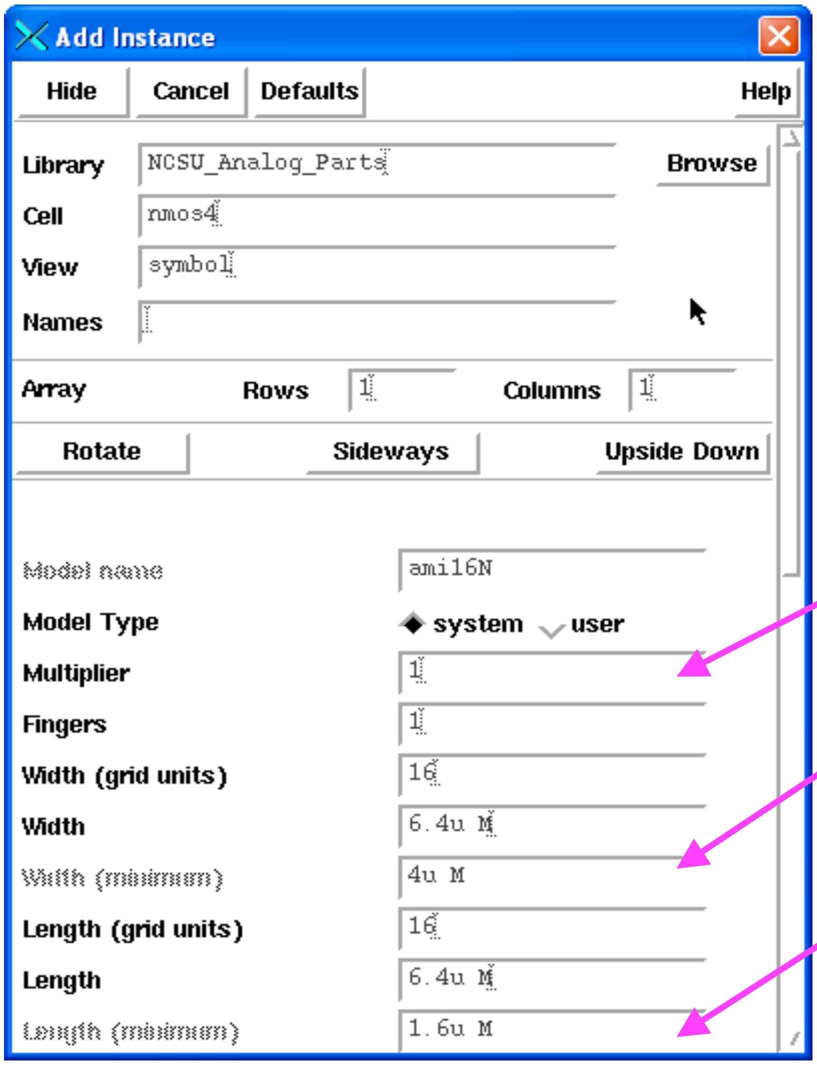


Figure 22: Getting the right nmos.

To get the parts we need, change the library to *NCSU_Analog_Parts* (Figure 21).

Click on *N_transistors* and the pop-up should look like Figure 22.

Click on *nmos4*. The A*dd Instance* window will then change, as in Figure 23.


Add Instance

Hide | Cancel | Defaults | Help

Library: NCSU_Analog_Parts
Cell: nmos4
View: symbol
Names:

Array — Rows: 1 — Columns: 1

Rotate | Sideways | Upside Down

Model name: ami16N
Model Type: ◆ system ✓ user
Multiplier: 1
Fingers: 1
Width (grid units): 16
Width: 6.4u M
Width (minimum): 4u M
Length (grid units): 16
Length: 6.4u M
Length (minimum): 1.6u M

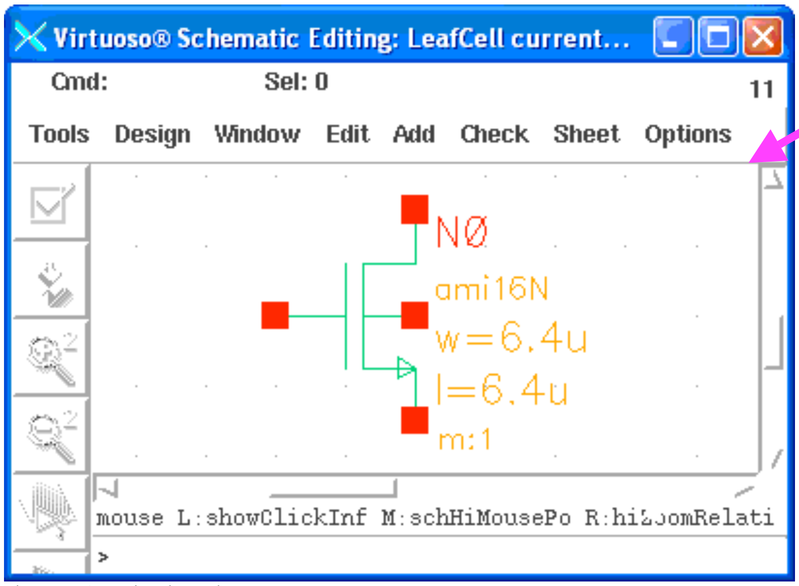
Number filled here indicates the number of transistors (with indicated size in this form) placing one next to another, sharing source or drain.
In this case, we use:
1 for I_ref_in
I_refx1,
2 for I_refx2,
4 for I_refx4,
so on and so forth.

Minimum transistor width for our attached tech library

Minimum transistor length for our attached tech library

Figure 23: Specifying the nmos needed.

Make sure you use the right sizes and multiplier, and then stamp it down as in Figure 24.


Virtuoso® Schematic Editing: LeafCell current...
Cmd:
Sel: 0
11
Tools
Design
Window
Edit
Add
Check
Sheet
Options
NØ
ami16N
w=6.4u
l=6.4u
m:1
mouse L:showClickInf M:schHiMousePo R:hiZoomRelati
>

Turned **Sideways.**
--To **Rotate**, or turn the object **Sideways** or **Upside Down**, go back to the *Add Instance form*, and click on the appropriate button in the middle of the form.

Figure 24: Stamping down the nmos.

After adjusting the sizes and number of multiplier for different transistors, place the transistors into the schematic as in Figure 25.

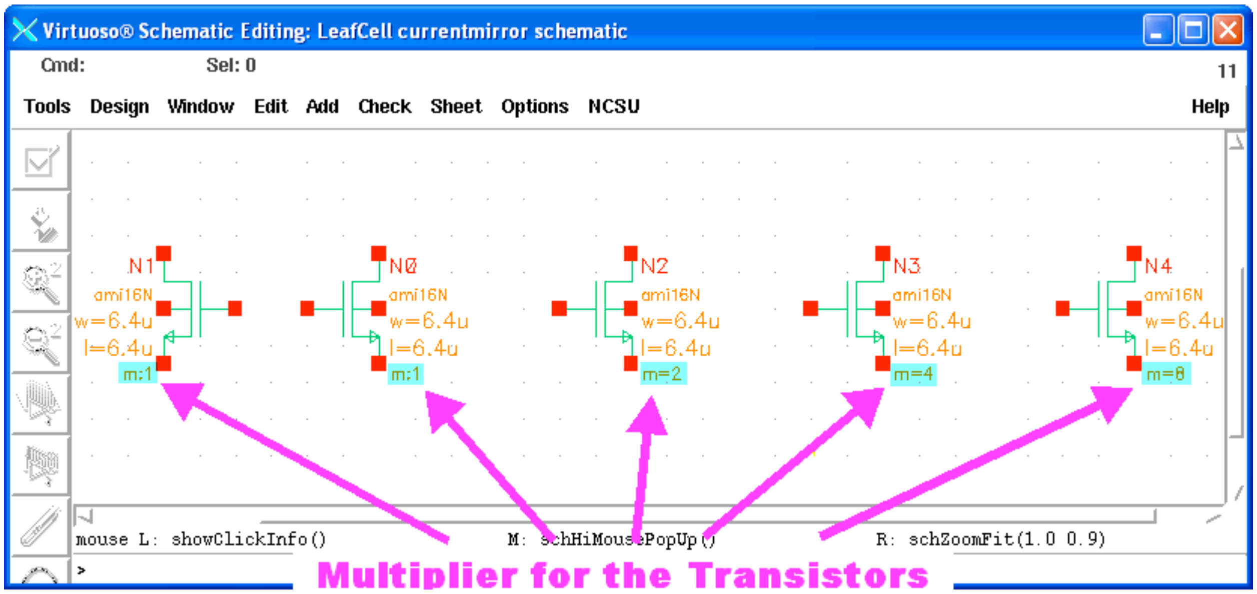


Figure 25: Placing the transistors in schematic

If you make a mistake and need to get out of adding instance mode, press the *esc* key from your keyboard, and start from the *Add Instance* window again.

Adding the gnd net:

Open the *Add Instance* window, and then the *Component Browser*. Click into the *Supply net* instead, and then you will see the *gnd* as in Figure 26.

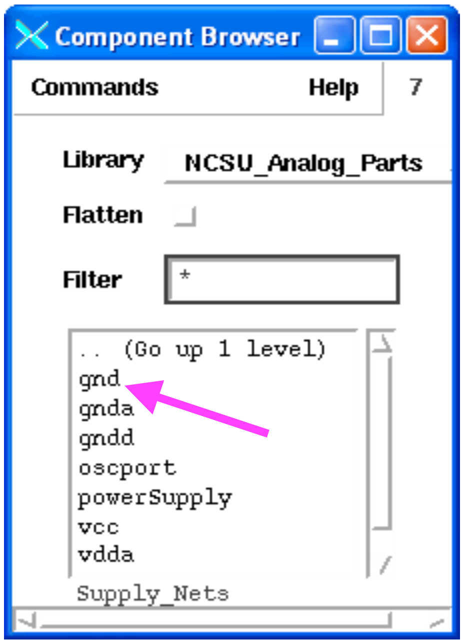


Figure 26: Getting the instance "gnd"

Note: The items in supply nets are actually global signals. Global signal are automatically given pins. This makes our symbols cleaner because we only show logic ports.

Adding Pins:

To add the input and output pins, go to A*dd... Pins* and a pop up like Figure 27 should appear. Fill it out exactly like Figure 27. The pins names must match the pins names in the symbol view we are going to create later. If they do not match (directionally or namely), the design, check & save routine will fail.

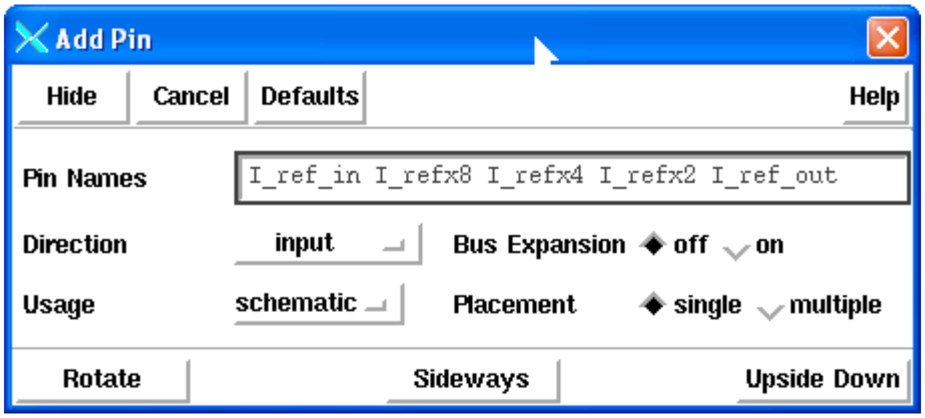

Pin Names: I_ref_in I_refx8 I_refx4 I_refx2 I_ref_out
Direction: input
Usage: schematic
Bus Expansion: ◆ off ✓ on
Placement: ◆ single ✓ multiple

Figure 27: The *Add Pin* Form.

Stamp down the input pin *I_ref_in*. Right click the mouse to rotate the pin as you want. Then go back to the form, change the *direction* from ***input*** to ***output*** for the rest of the pins in the *Add Pin* form as in Figure 28.

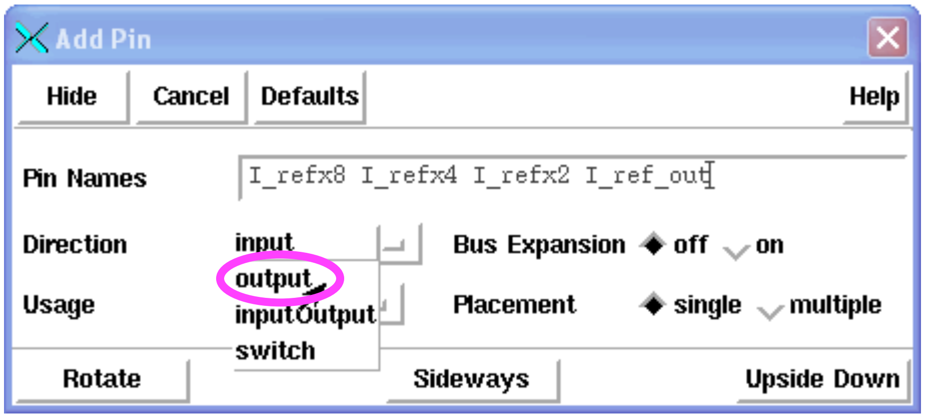

Pin Names: I_refx8 I_refx4 I_refx2 I_ref_out
Direction: input
Bus Expansion: ◆ off ✓ on
Placement: ◆ single ✓ multiple

Figure 28: Changing direction for the Y pin.

Stamp them down. The schematic will then look like Figure 29.

Adding wires:

What is left is to wire up the transistors.

To add a wire, press *w* from keyboard. The wire will snap to place at the proper ports of each device. Wire it up like Figure 30.

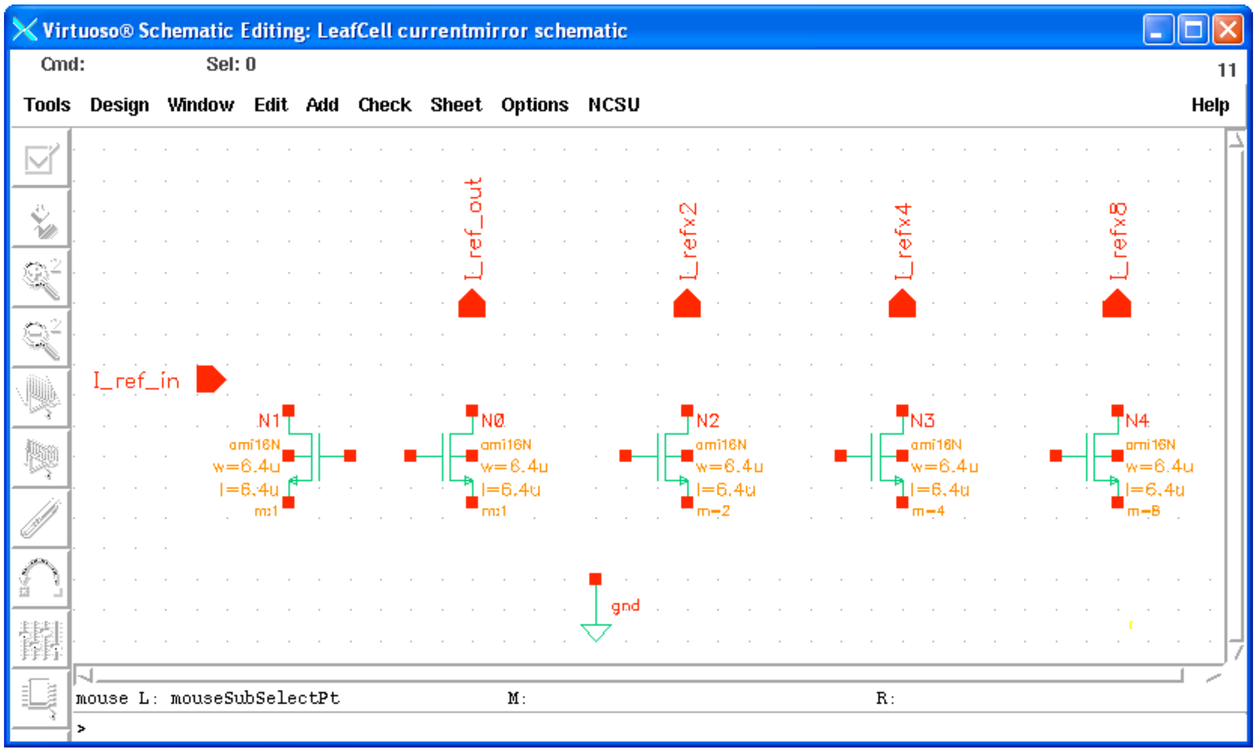


Figure 29: Stamping down all the instances

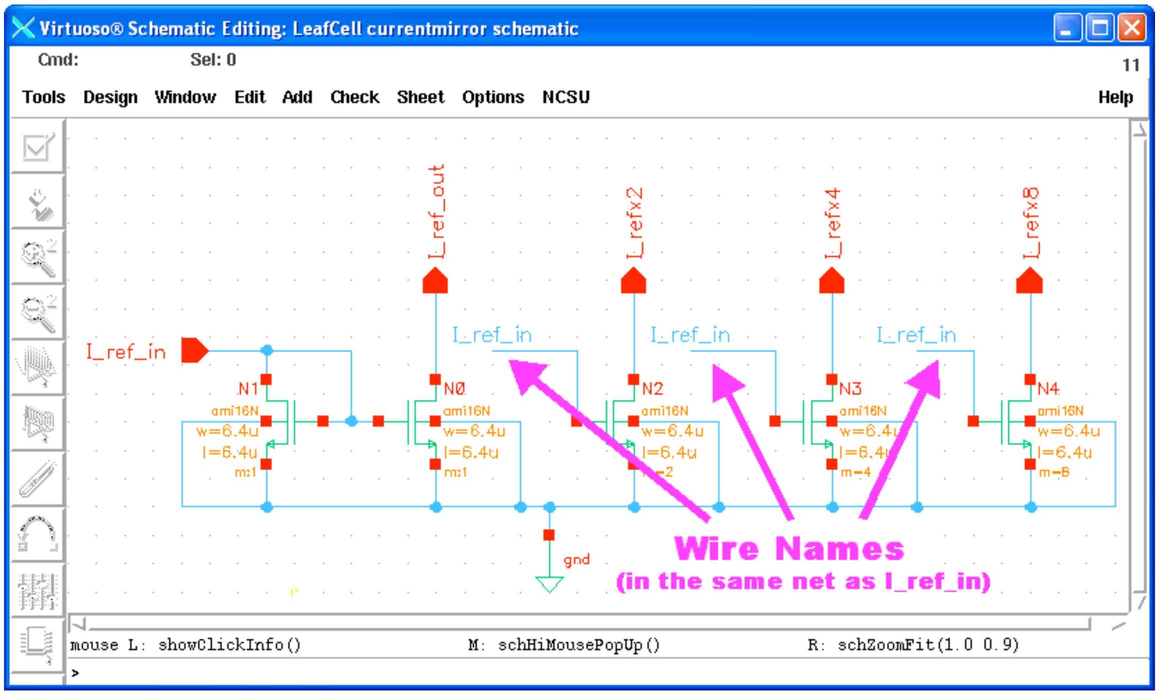
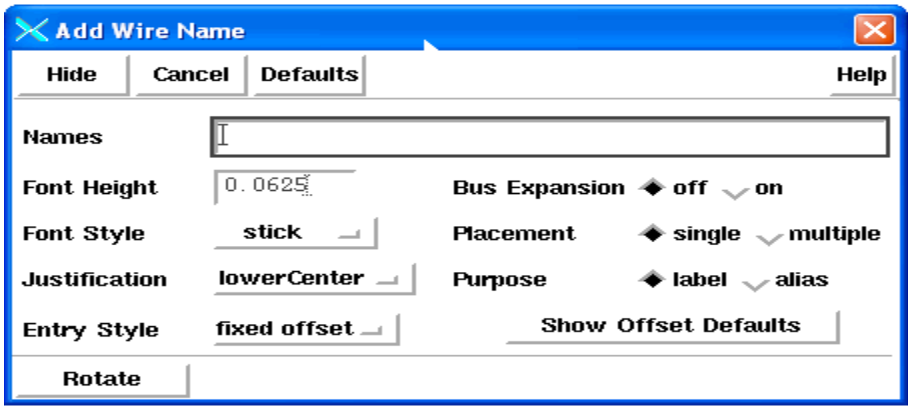


Figure 30: The completed schematic.

To add wire names, go to *Add…Wire Name…*, the *Add Wire Name* window will pop up as in Figure 31.



Names:
Font Height: 0.0625
Bus Expansion: ◆ off ◇ on
Font Style: stick
Placement: ◆ single ◇ multiple
Justification: lowerCenter
Purpose: ◆ label ◇ alias
Entry Style: fixed offset
Show Offset Defaults

Figure 31: Adding the wire names.

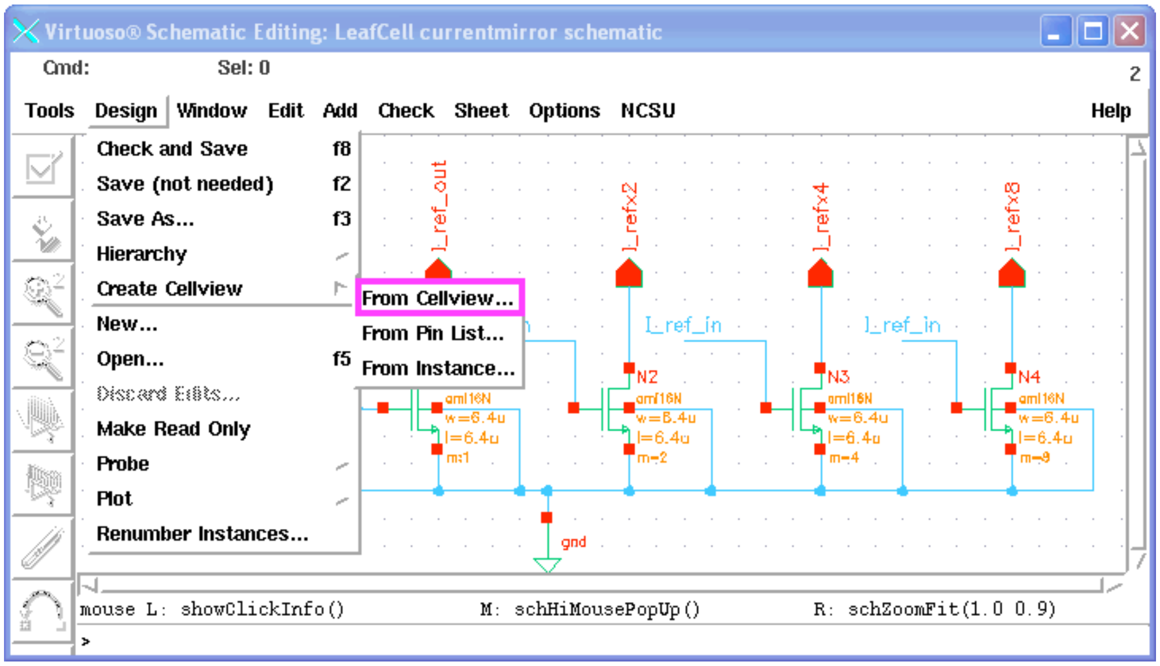
Fill in the names, and then stamp down on top of the wire with that coordinate name, as shown in Figure 30 above.

To save and check your schematic for errors, go to *Design... Check & Save*. Any errors will be highlighted in the schematic window. If so, it is usually something not connected, or a port name is wrong. Fix these error(s) before continuing.

Creating a symbol view:

There are many ways to create a symbolic view. The easiest way is to let the system create it from the schematic view. To create a symbol view:

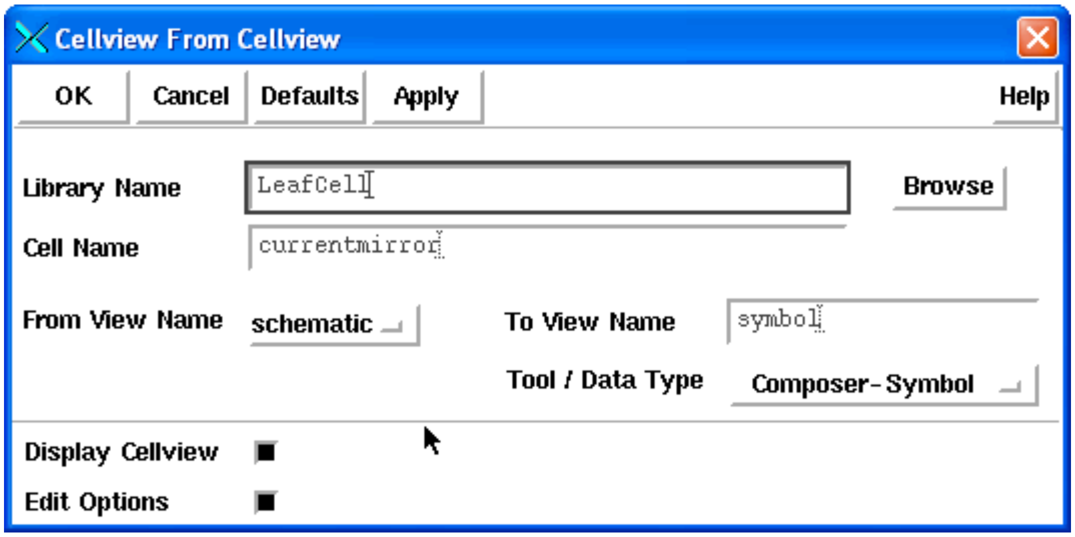
1. Go to the *schematic* window and go to *Design…Create Cellview…From Cellview…* (Figur32).



```
mouse L: showClickInfo()          M: schHiMousePopUp()          R: schZoomFit(1.0 0.9)
>
```

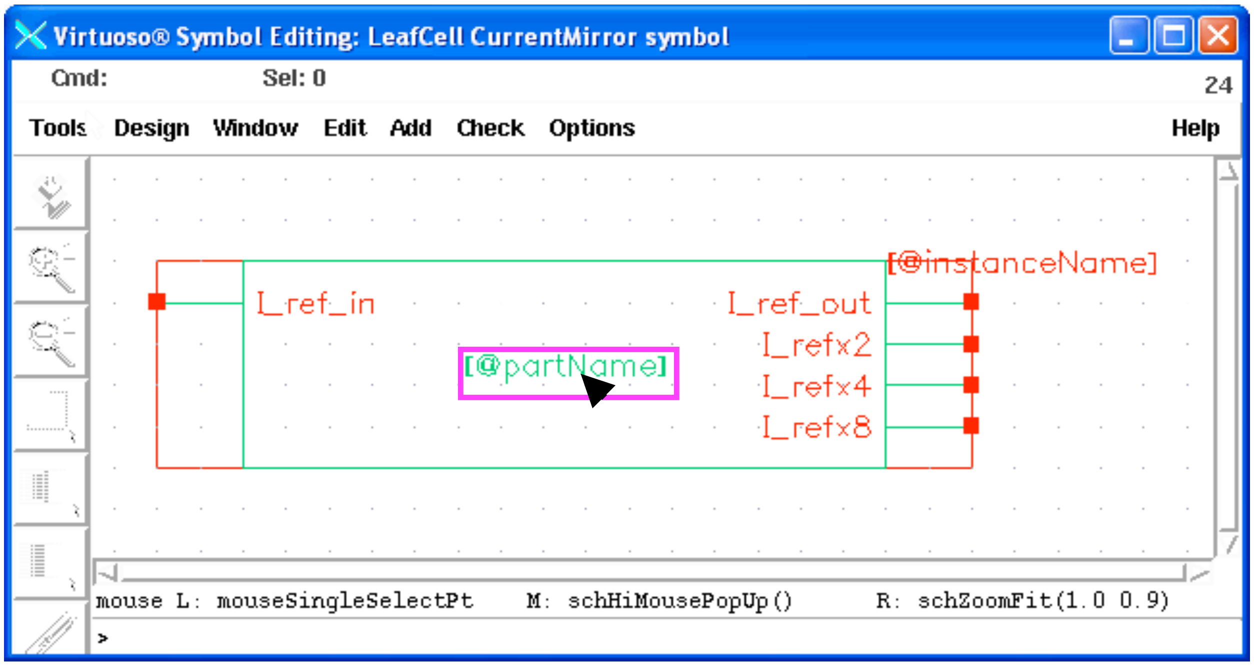
Figure 32: Where to go for creating symbol from schematic.

2. The *Cellview From Cellview* window will pop-up as in Figure 33. Click *OK*. The *Virtuoso Symbol Editing* window should appear with the symbol made by the system (Figure 34).



Library Name: LeafCell
Cell Name: currentmirror
From View Name: schematic
To View Name: symbol
Tool / Data Type: Composer-Symbol
Display Cellview ■
Edit Options ■

Figure 33: The Cellview From Cellview Window.



```
mouse L: mouseSingleSelectPt     M: schHiMousePopUp()       R: schZoomFit(1.0 0.9)
>
```

Figure 34: CurrentMirror Symbol created by the system.

To change the part name, click at "[@partName]", then go to menu: Edit...Properties…Objects…, or use the keyboard to input "q".

Then you can change the *Label* from "[@partName]" back to "CurrentMirror" from the *Properties* form.

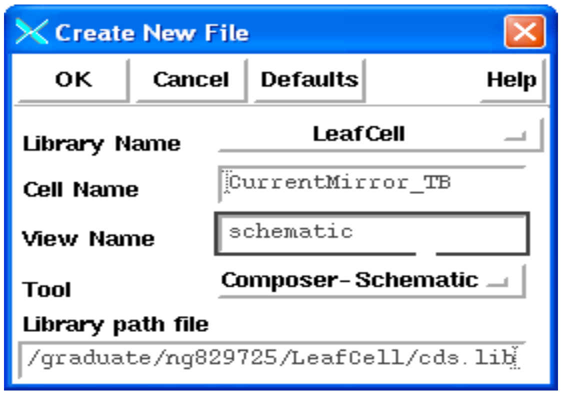
Save and Close the Virtuoso Symbol Editor.

Creating a test bench:

Now that you have created a schematic and symbolic view of the CurrentMirror, it is time to create a test bench. This will be a virtual test bench, but it will have to have a power supply net, input vectors, and the testing object-the CurrentMirror- to perform correctly.

Creating the Test Bench:

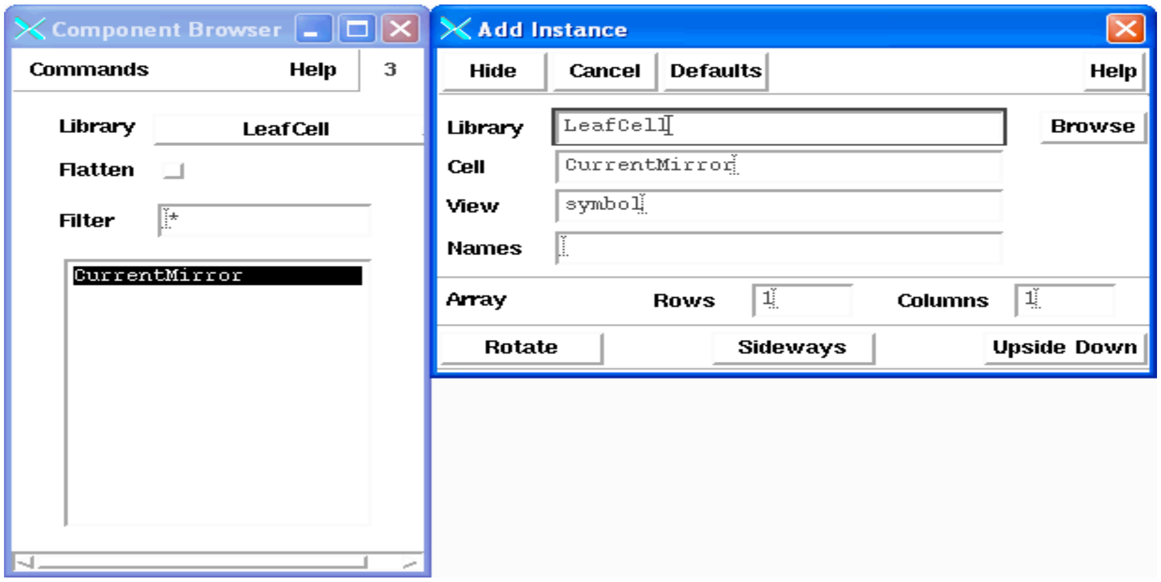
In the library manager, go to *File… New… Cell View.* Fill out the pop-up window exactly like Figure 35 and click on *OK.* The *schematic editor* will appear.



Library Name: LeafCell
Cell Name: CurrentMirror_TB
View Name: schematic
Tool: Composer-Schematic
Library path file: /graduate/ng829725/LeafCell/cds.lib

Figure 35: Creating a test bench.

In the *schematic editor* window, press *i* in your keyboard or go to *Add... Instance.* Click on *Browse*, select the *LeafCell* library as in Figure 36, (click on *Flatten* if there are many instances under the library), select *CurrentMirror* and stamp it down in your test bench schematic (Figure 37).



Library: LeafCell
Flatten
Filter: *
CurrentMirror

Library: LeafCell
Cell: CurrentMirror
View: symbol
Names:
Array Rows: 1 Columns: 1

Figure 36: Adding Current Mirror into the test bench.

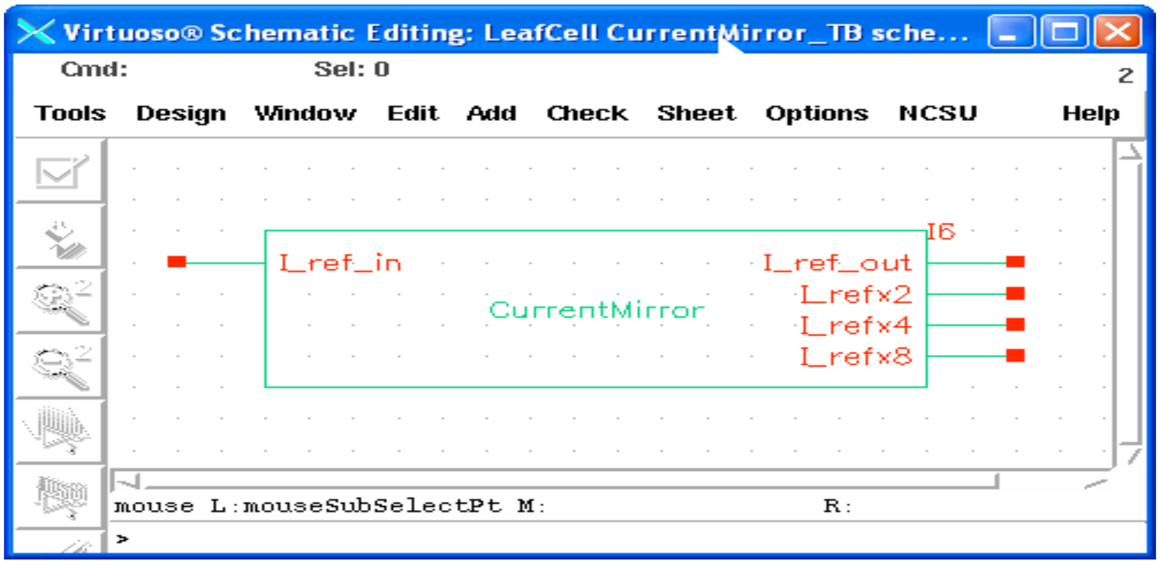


Figure 37: Stamping down the Current Mirror. into the test bench.

While still in add instance mode add the *gnd* and *vdd* global symbols just like you did when creating the schematic for the CurrentMirror (Figure 38).

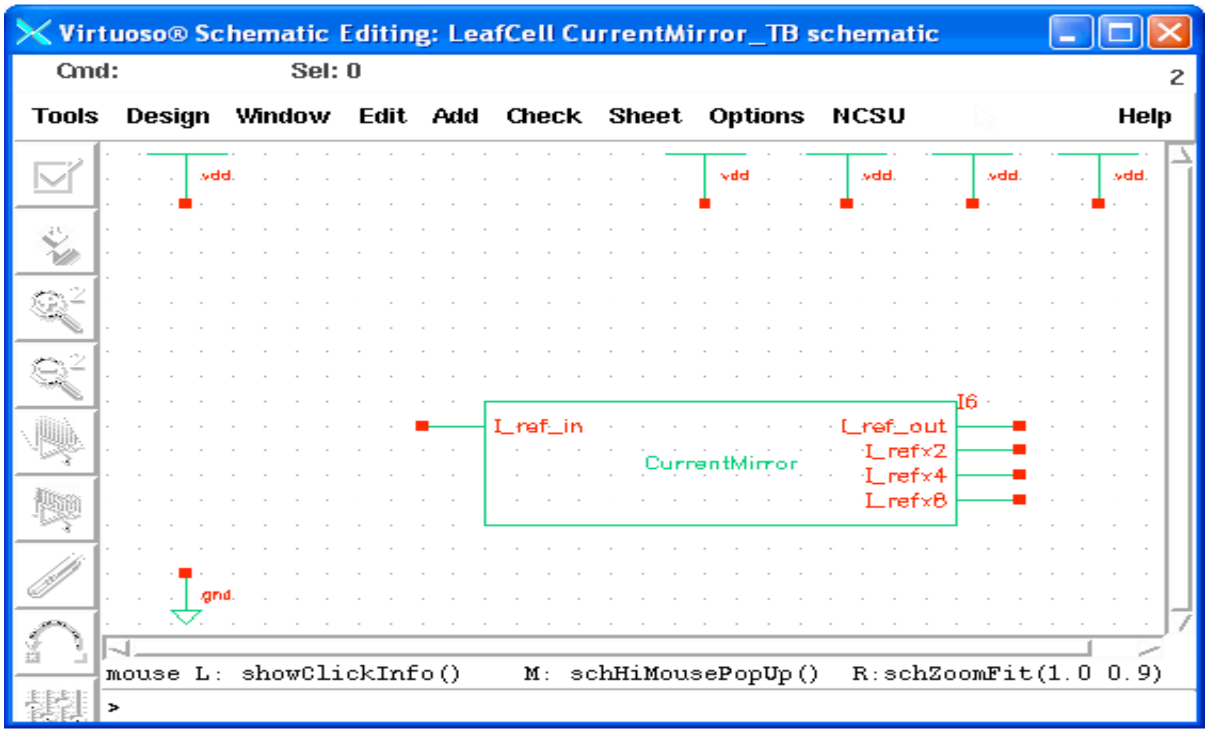
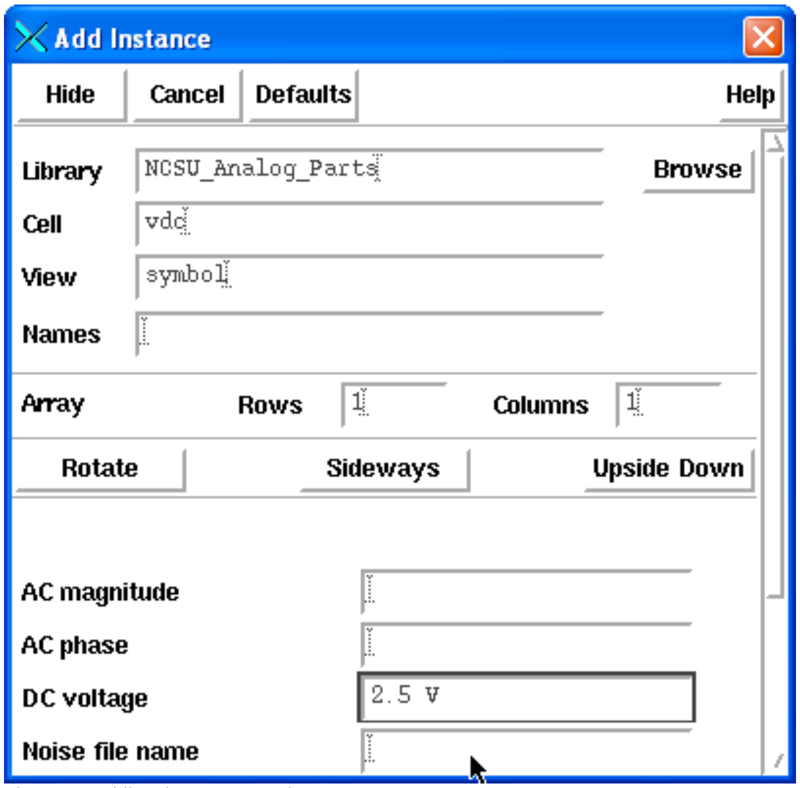


Figure 38: Stamping Vdd & Gnd into the test bench.

Add the power supply (*vdc* from voltage sources in *NCSU_Analog_Parts*) according to Figure 39. Make sure set the DC voltage to 2.5 volts. Stamp it down into the schematic (Figure 40).


Add Instance

Hide | Cancel | Defaults | Help

Library NCSU_Analog_Parts Browse
Cell vdc
View symbol
Names

Array Rows 1 Columns 1

Rotate | Sideways | Upside Down

AC magnitude
AC phase
DC voltage 2.5 V
Noise file name

Figure 39: Adding the power supply

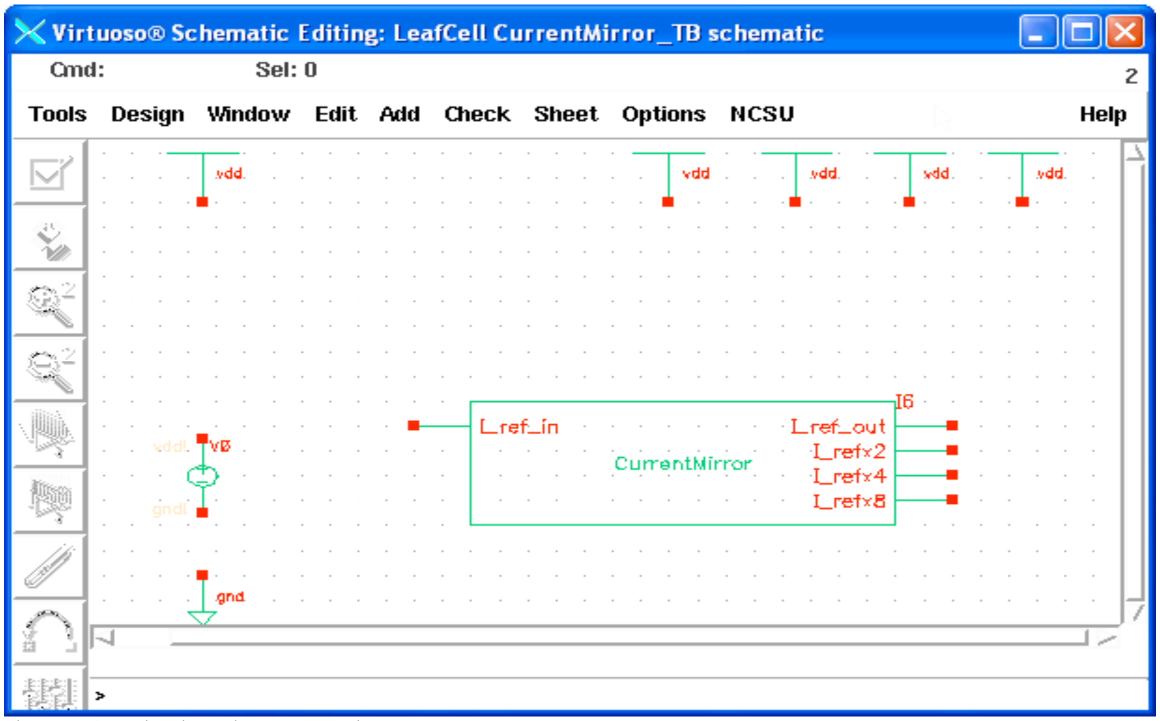
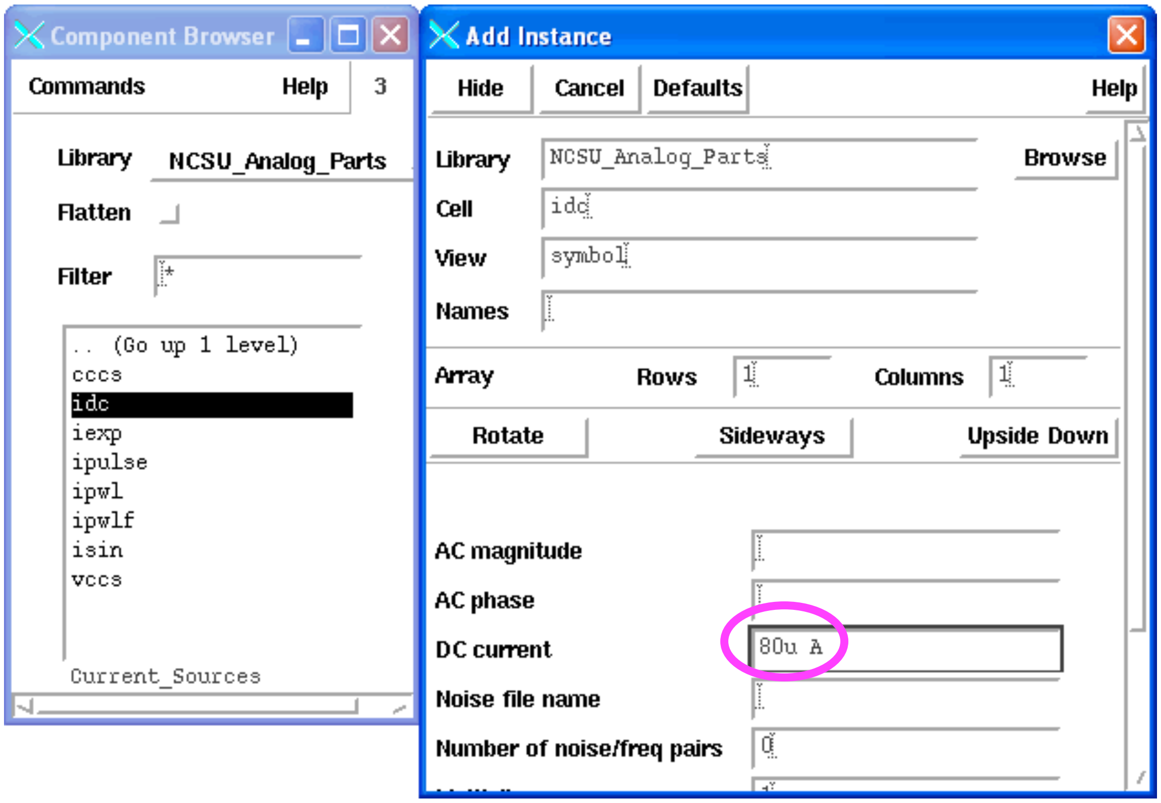

Virtuoso® Schematic Editing: LeafCell CurrentMirror_TB schematic
Cmd: Sel: 0
2
Tools Design Window Edit Add Check Sheet Options NCSU Help
vdd
vdd
vdd
vdd
vdd
I6
I_ref_in
I_ref_out
I_refx2
I_refx4
I_refx8
CurrentMirror
V0
vdd
gnd
gnd

Figure 40: Stamping down the power supply

Then we need to add a current source for the I_ref_in signal. In order to get the instance *idc*, we Browse into library *NCSU_Analog_Parts*, then *Current_Sources* (Figure 41). Make sure to put in the value for the dc current (in this case, we put in 80u, and make sure there is no space between them) (Figure 41). Stamp it down as in Figure 42.


Component Browser

Commands Help 3

Library NCSU_Analog_Parts

Flatten

Filter *

.. (Go up 1 level)
cccs
idc
iexp
ipulse
ipwl
ipwlf
isin
vccs

Current_Sources

Add Instance

Hide Cancel Defaults Help

Library NCSU_Analog_Parts Browse

Cell idc

View symbol

Names

Array Rows 1 Columns 1

Rotate Sideways Upside Down

AC magnitude

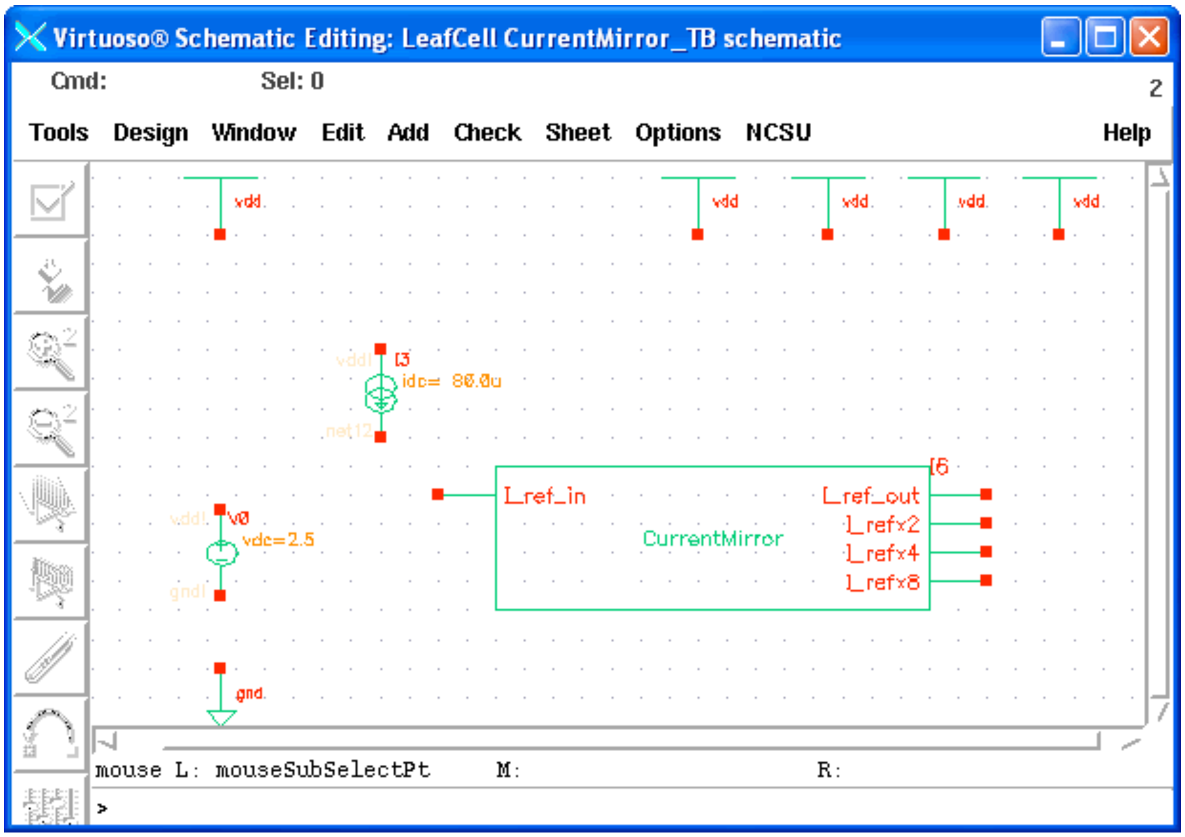
AC phase

DC current 80u A

Noise file name

Number of noise/freq pairs 0

Figure 41: Adding the current source.


Virtuoso® Schematic Editing: LeafCell CurrentMirror_TB schematic

Cmd: Sel: 0 2

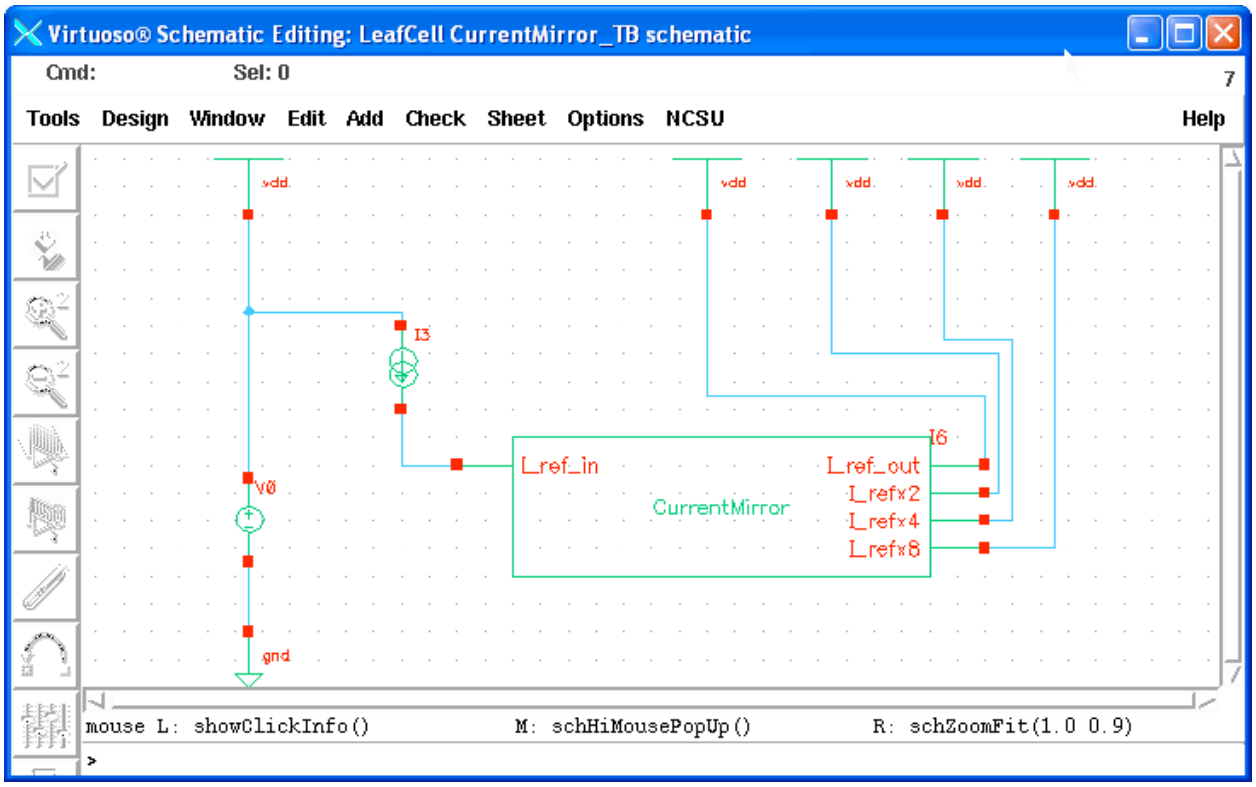
Tools Design Window Edit Add Check Sheet Options NCSU Help

mouse L: mouseSubSelectPt M: R:

>

Figure 42: Stamping down the current source.

Now we have all the instances we need. The next thing to do is to wire them up as in Figure 43.


Virtuoso® Schematic Editing: LeafCell CurrentMirror_TB schematic
Cmd:
Sel: 0
Tools Design Window Edit Add Check Sheet Options NCSU Help
vdd
I3
V0
gnd
I6
Lref_in
CurrentMirror
Lref_out
Lrefx2
Lrefx4
Lrefx8

```
mouse L: showClickInfo()          M: schHiMousePopUp()          R: schZoomFit(1.0 0.9)
>
```

Figure 43: The completed test bench

Press the *esc* key to get out of add instance mode. If you want to move the symbol around, press the ***m*** key for move, and press the ***esc*** key to get out of it.

Go to *Design... Check & Save* to check for errors.

You are now ready to simulate the Current Mirror!

Simulation in Spectre Spice using the Affirma environment:

Now that you have created a schematic and symbolic view of the Current Mirror as well as a test bench for testing it, it is ready to run the Affirma Analog environment tool. We are simulating an analog circuit in a physics based environment that solves for voltages and currents over time.

To simulate your circuit:

From the *CurrentMirror_TB* schematic window, go to *Tools... Analog Environment.* A pop-up window like Figure 44 should appear. You will need to customize your environment the first time you run *Affirma*.

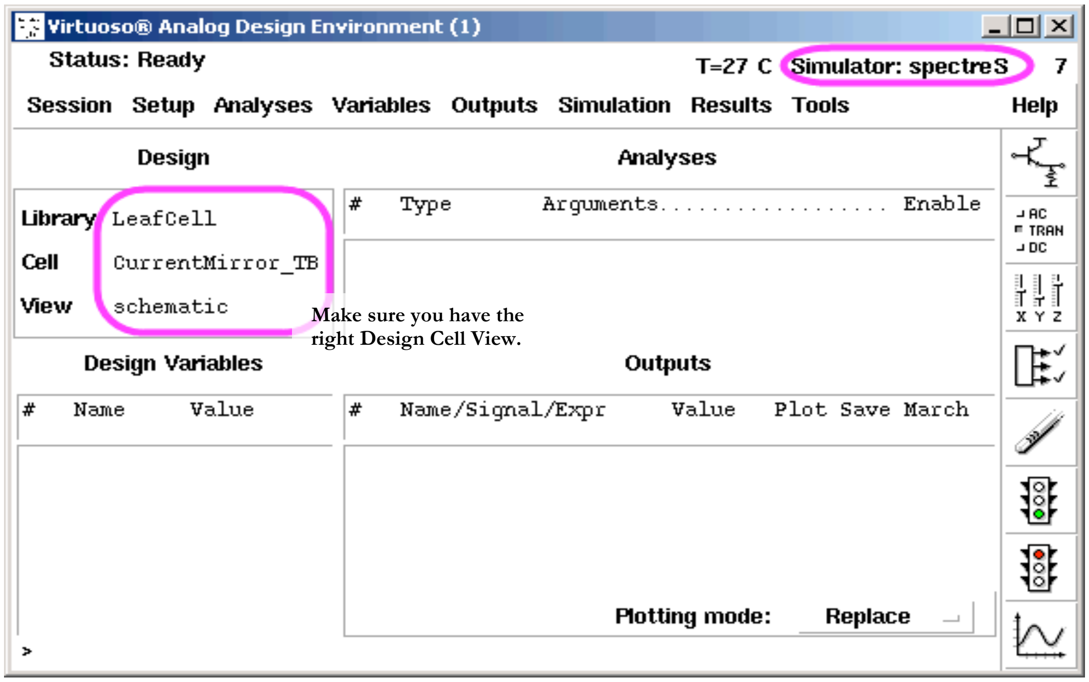
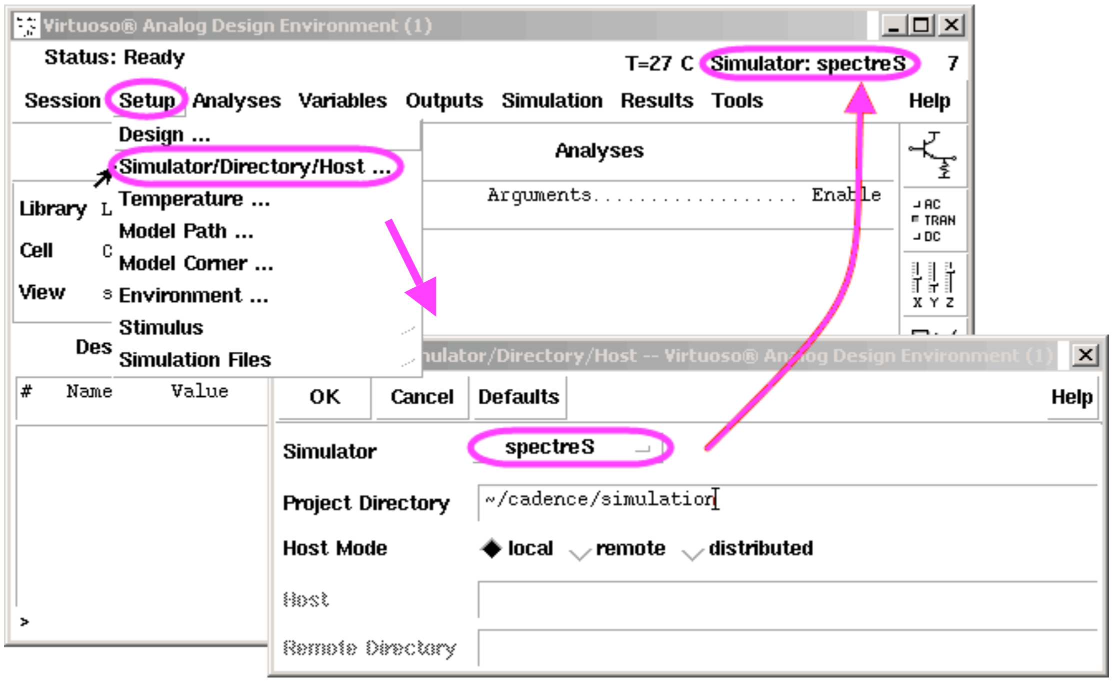


Figure 44: Affrima Analog Environemt

You will be using the Spectre Spice simulator, which is a spice like simulator, using BSIM3 model decks, but the underlying algorithms are different. These algorithms are transparent to the user.

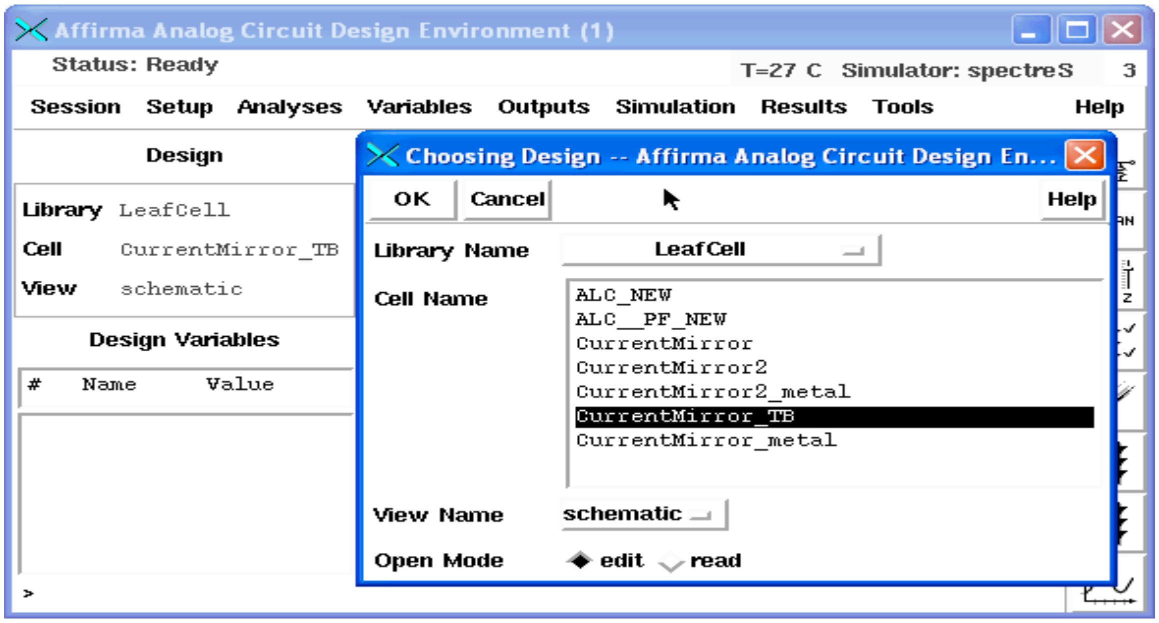
Go to *Setup...Simulator/Directory/Host*. A pop-up like Figure 45 should appear. Make sure the Simulator is set to *SpectreS*, and you have the right project directory. Click *OK* when done.



Simulator: spectreS
Project Directory: ~/cadence/simulation
Host Mode: [✓ local] remote distributed
Host
Remote Directory

Figure 45: Simulator/Directory/Host.

If the Design is not shown, go to Setup…Design…, and choose the right design to simulate (figure 46).


Affirma Analog Circuit Design Environment (1)
Status: Ready
T=27 C Simulator: spectreS 3
Session Setup Analyses Variables Outputs Simulation Results Tools Help
Design
Design Variables
# Name Value
Choosing Design -- Affirma Analog Circuit Design En…
OK Cancel Help

Library LeafCell
Cell CurrentMirror_TB
View schematic

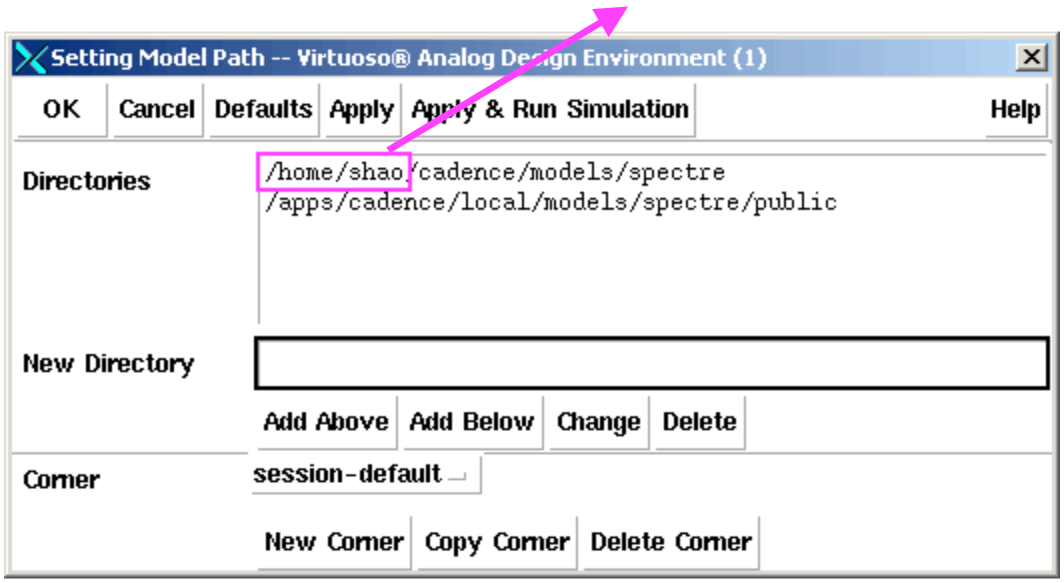
Library Name: LeafCell

Cell Name:
```
ALC_NEW
ALC__PF_NEW
CurrentMirror
CurrentMirror2
CurrentMirror2_metal
CurrentMirror_TB
CurrentMirror_metal
```
View Name: schematic

Open Mode: [✓ edit] read

Figure 46: Choosing the right design to simulate..

Go to *Setup... Model Path* and a pop-up like Figure 47 should appear. Make sure you have input these model paths in the form, except the *home directory* may differ.

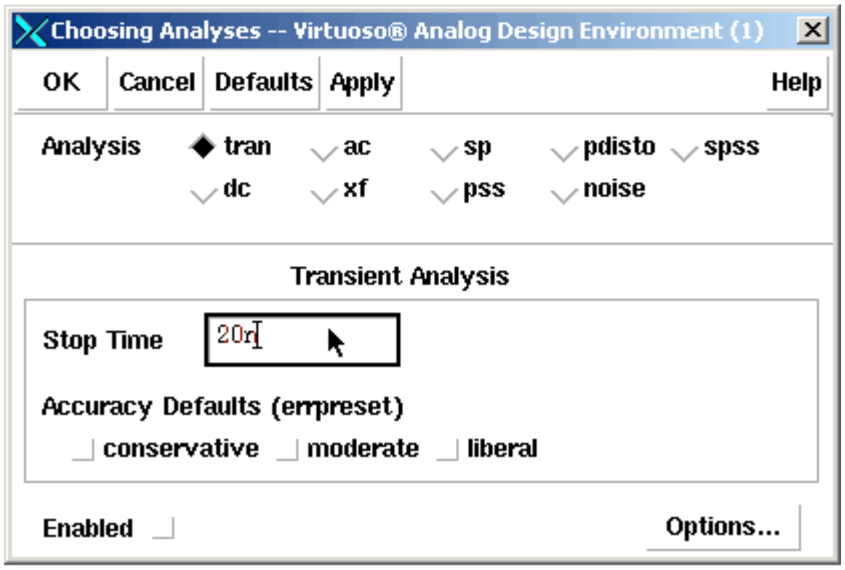

Setting Model Path -- Virtuoso® Analog Design Environment (1)
OK Cancel Defaults Apply Apply & Run Simulation Help
Add Above Add Below Change Delete
New Corner Copy Corner Delete Corner

Directories
```
/home/shao/cadence/models/spectre
/apps/cadence/local/models/spectre/public
```
New Directory

Corner: session-default

Figure 47: Setting up the model path.

Now you need to set up what kind of analysis you will run. Go to *Analyses... Choose* and fill out the pop-up according to Figure 48.



Choosing Analyses -- Virtuoso® Analog Design Environment (1)

OK | Cancel | Defaults | Apply | Help

Analysis: [✓ tran] ac sp pdisto spss dc xf pss noise

Transient Analysis

Stop Time: 20n

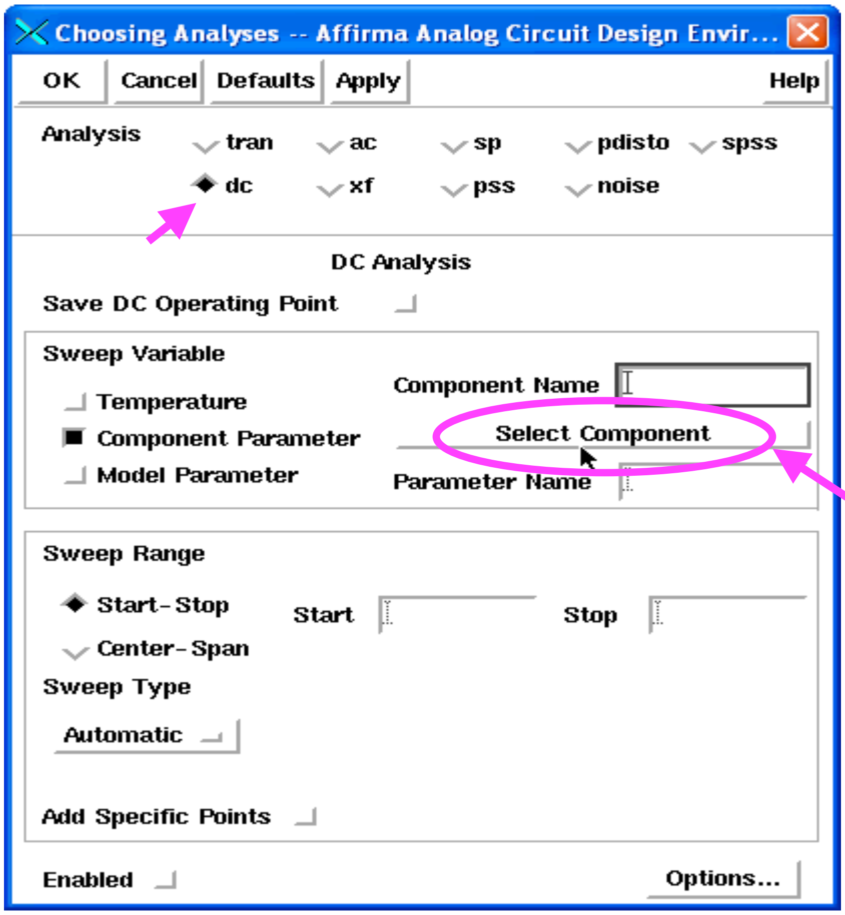
Accuracy Defaults (errpreset): conservative, moderate, liberal

Enabled

Options...

Figure 48: Setting up the transient analysis

While still in the *Chososing Analyses Form*, click on Analysis *dc*. Then Figure 49 will show.



Choosing Analyses -- Affirma Analog Circuit Design Envir...

OK | Cancel | Defaults | Apply | Help

Analysis: tran ac sp pdisto spss [✓ dc] xf pss noise

DC Analysis

Save DC Operating Point

Sweep Variable: Temperature, [✓ Component Parameter], Model Parameter

Component Name:

Select Component

Parameter Name:

Sweep Range: [✓ Start-Stop], Center-Span — Start: Stop:

Sweep Type: Automatic

Add Specific Points

Enabled

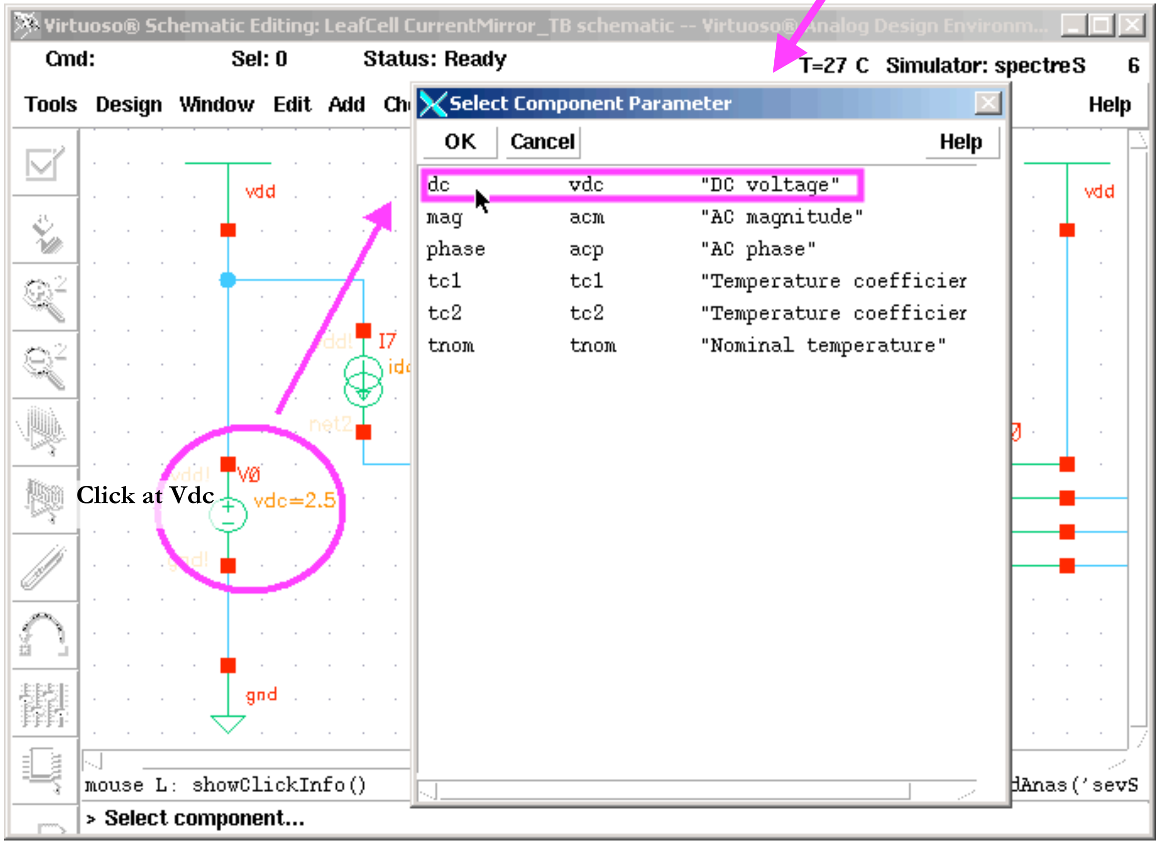
Options...

Click on it, and then the test bench schematic will pop-up for you to select the right component.

Note: Schematic window will pop-up only if it has not been closed. If it is not opened yet, you have to open it manually to select the component.

Figure 49: choosing the dc analysis

Click on *Select Component*, then the CurrentMirror_TB Schematic window will pop in front, point your cursor at the *Voltaget source* and click it, then the *Select Component Parameter* form shows. Choose the *"DC voltage"* and then click *OK*.

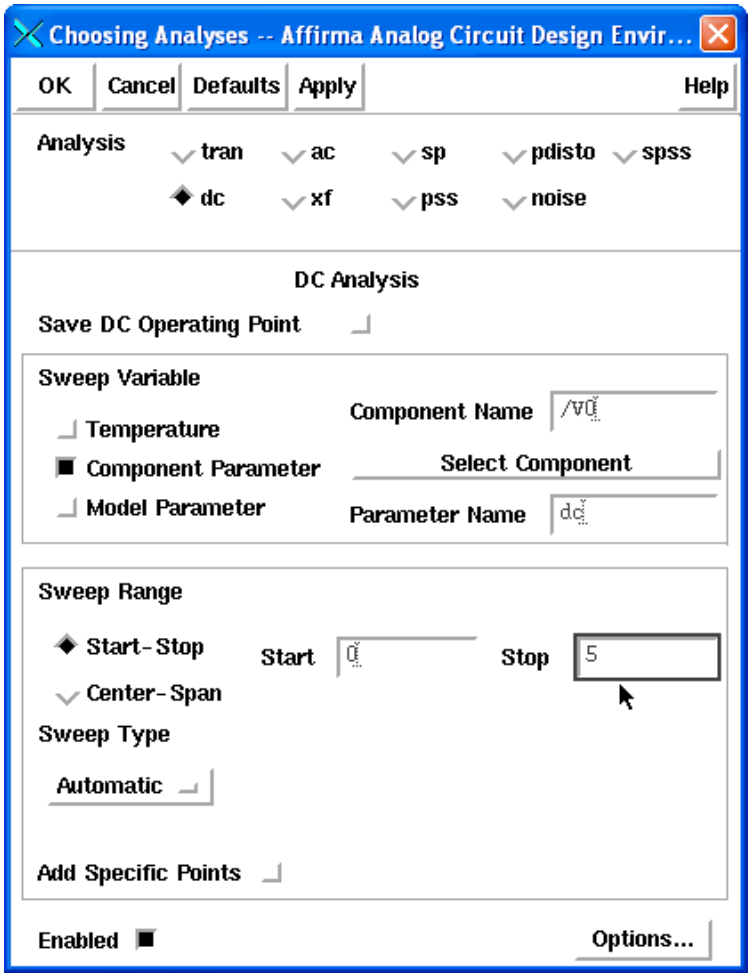


```
dc          vdc         "DC voltage"
mag         acm         "AC magnitude"
phase       acp         "AC phase"
tc1         tc1         "Temperature coefficier
tc2         tc2         "Temperature coefficier
tnom        tnom        "Nominal temperature"
```

Figure 50: Setting up the dc analysis

Fill out the rest of the form as in Figure 51, and click *OK*. Make sure you set the right range of the start-stop point for the current source.

Then the *Affirma Analog Circuit Design Environment* window will look as in Figure 52.



Save DC Operating Point

Sweep Variable

- Temperature
- [✓ Component Parameter]
- Model Parameter

Component Name: /V0

Parameter Name: dc

Sweep Range

[✓ Start-Stop] Center-Span

Start: 0

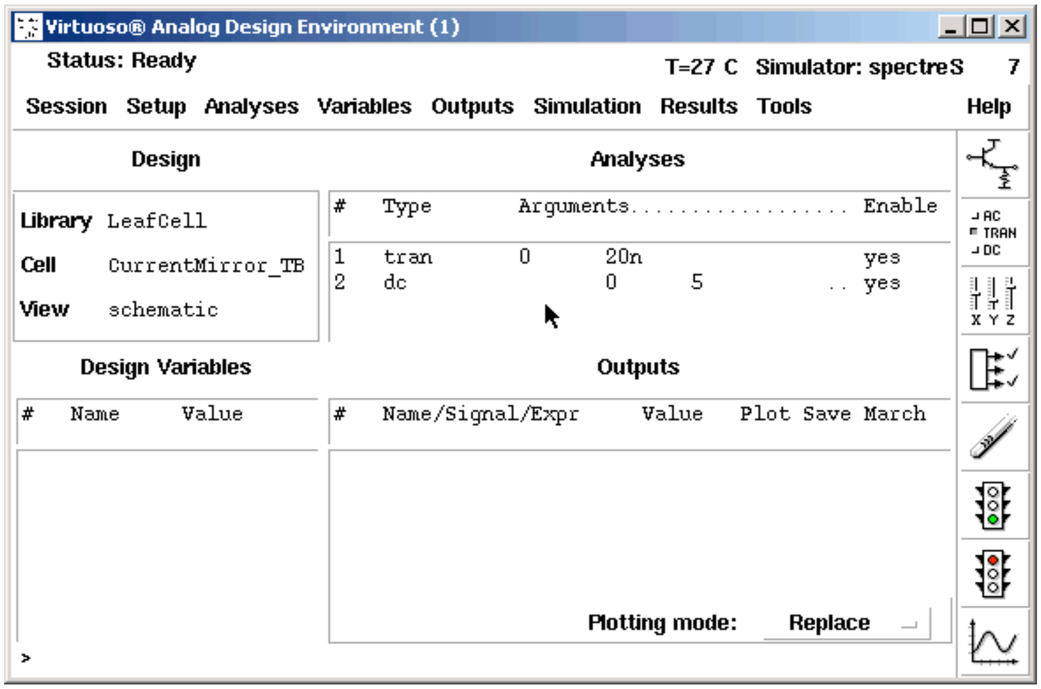
Stop: 5

Sweep Type: Automatic

Add Specific Points

Enabled [✓]

Figure 51: Setting up for the rest of the dc analysis



Library LeafCell

Cell CurrentMirror_TB

View schematic

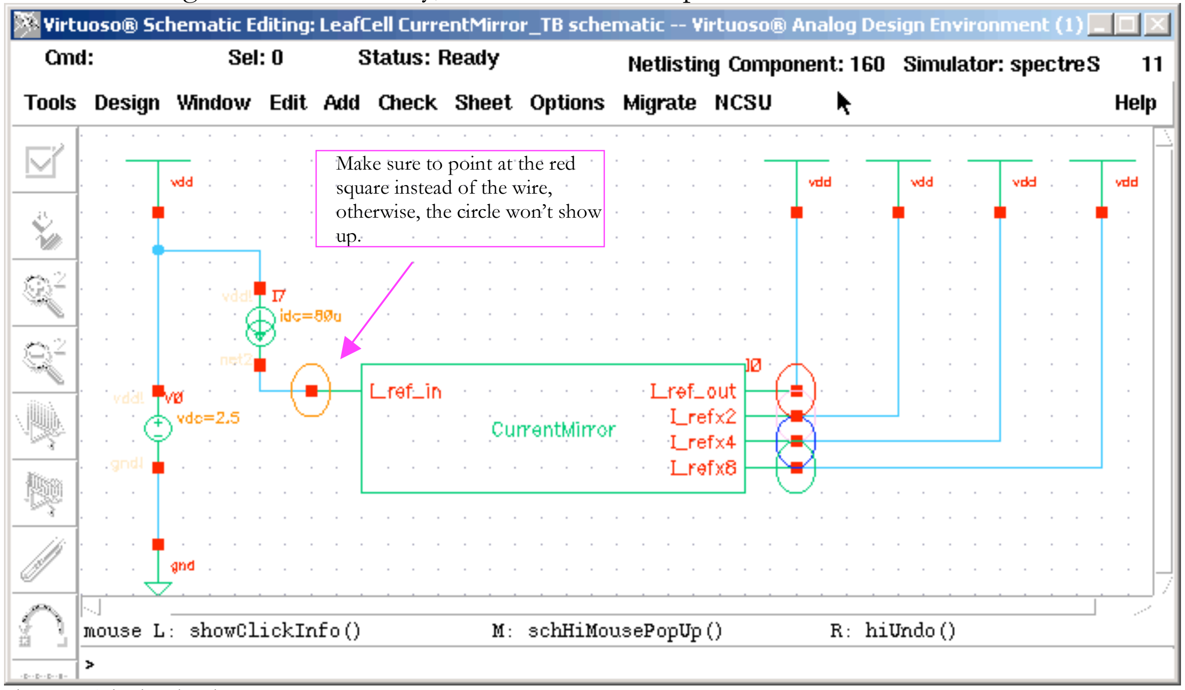
```
#   Type        Arguments.................  Enable
1   tran        0        20n                yes
2   dc                   0        5      .. yes
```

```
#   Name        Value
```

```
#   Name/Signal/Expr      Value    Plot Save March
```

Figure 52: Affirma window after setting up the analyses.

Now you need to choose which vectors you wish to plot out. Go to *Outputs...To Be Plotted... Select On Schematic*. Then the *Currentmirror_TB schematic* window will pop on top of all other windows. Click on the input and output pins of the CurrentMirror. The pins will be circled as in Figure 53. In this way, the current will be probed.

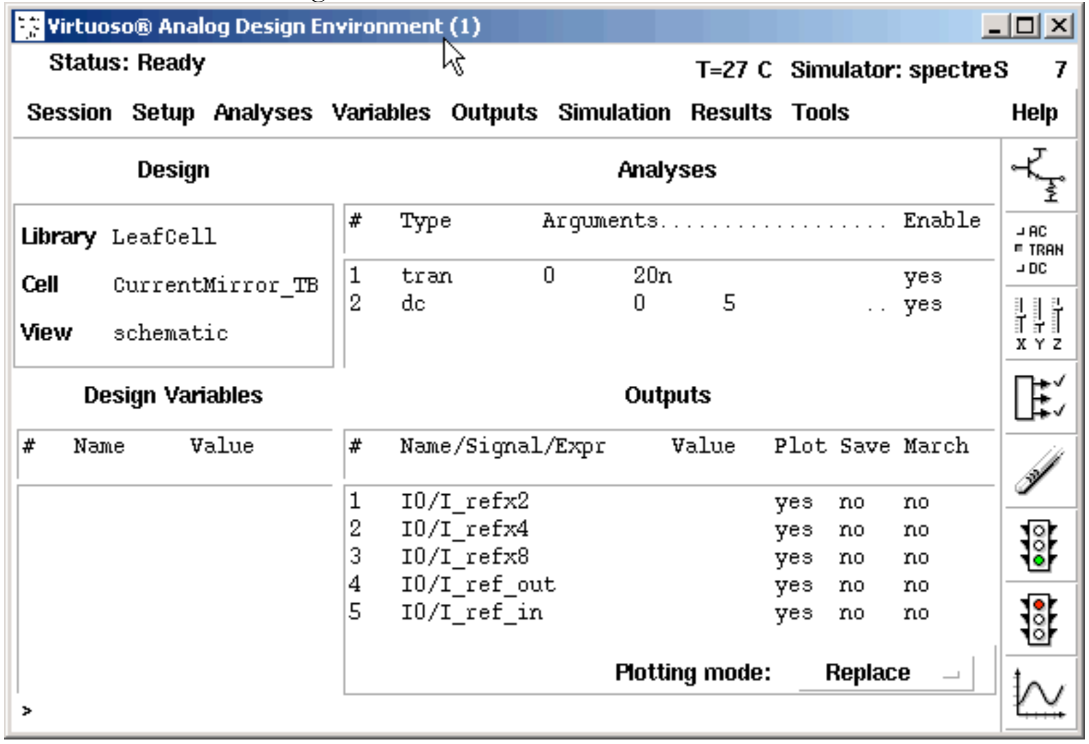

Virtuoso® Schematic Editing: LeafCell CurrentMirror_TB schematic -- Virtuoso® Analog Design Environment (1)
Cmd:
Sel: 0
Status: Ready
Netlisting Component: 160
Simulator: spectreS
11
Tools Design Window Edit Add Check Sheet Options Migrate NCSU Help
vdd
I7
idc=80u
V0
vdc=2.5
gnd
I_ref_in
CurrentMirror
I_ref_out
I_refx2
I_refx4
I_refx8
I0

Make sure to point at the red square instead of the wire, otherwise, the circle won't show up.

```
mouse L: showClickInfo()          M: schHiMousePopUp()          R: hiUndo()
>
```

Figure 53: Selecting the pins.

Press the *esc* key to get out of selection mode. Your *Affirma* pop-up should look like Figure 54, but the net names might be different.


Virtuoso® Analog Design Environment (1)
Status: Ready
T=27 C
Simulator: spectreS
7
Session Setup Analyses Variables Outputs Simulation Results Tools Help
Design
Analyses
Design Variables
Outputs
AC
TRAN
DC
X Y Z

Library	LeafCell
Cell	CurrentMirror_TB
View	schematic

#	Type	Arguments.................			Enable
1	tran	0	20n		yes
2	dc		0	5 ..	yes

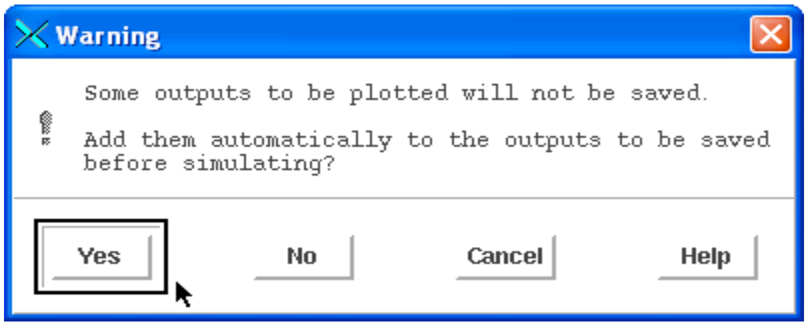
#	Name	Value

#	Name/Signal/Expr	Value	Plot	Save	March
1	I0/I_refx2		yes	no	no
2	I0/I_refx4		yes	no	no
3	I0/I_refx8		yes	no	no
4	I0/I_ref_out		yes	no	no
5	I0/I_ref_in		yes	no	no

Plotting mode: Replace

Figure 54: a complete Affirma window.

Go to Simulation... Run. A *Warning* window will pop-up for saving the outputs (Figure 55). Just click on *Yes* and let the simulator run. The CIW and the Affirma pop-up will show you the process it goes through to simulate your circuit. If all went well, a plot like Figure 56 should appear.



Some outputs to be plotted will not be saved.

Add them automatically to the outputs to be saved before simulating?

Figure 55: Yes to Warning.

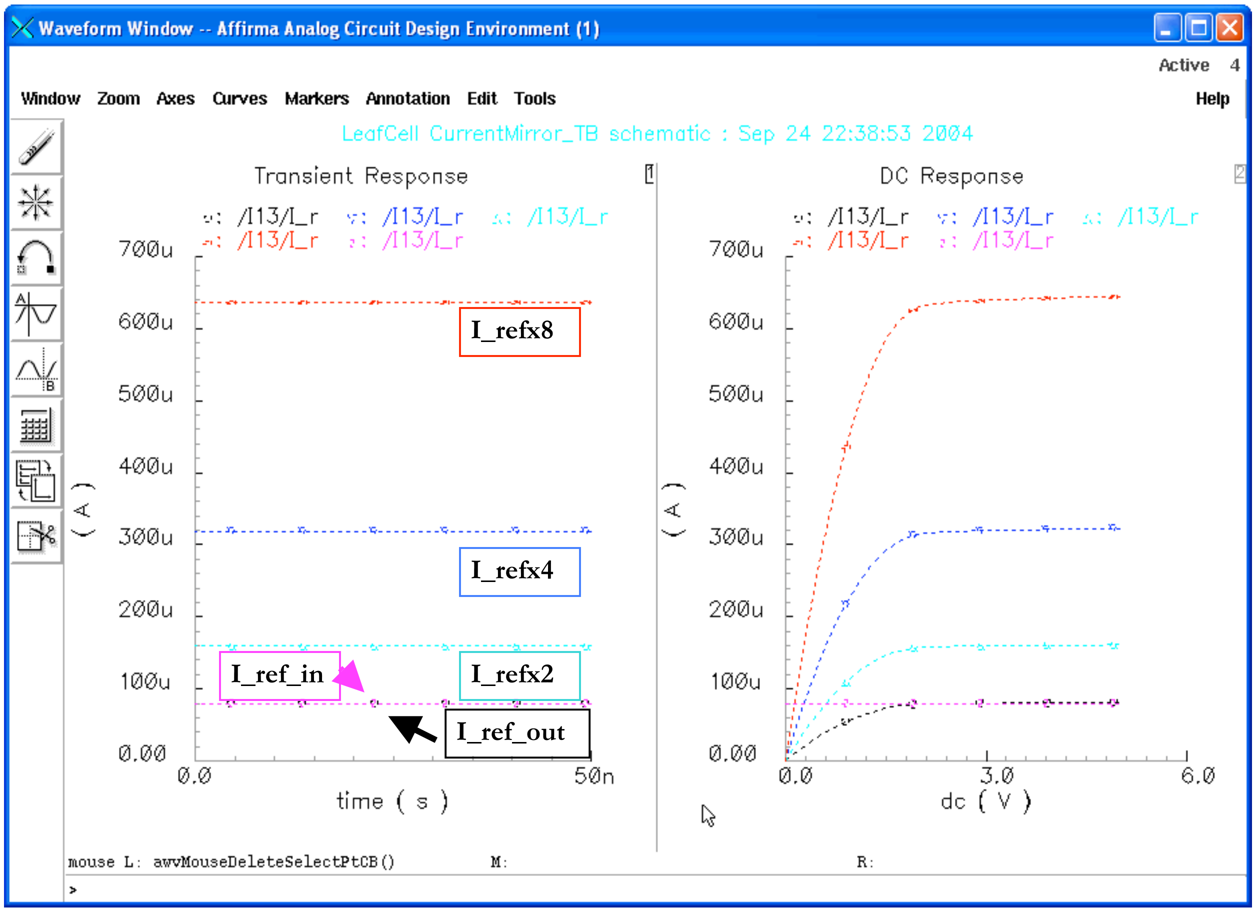
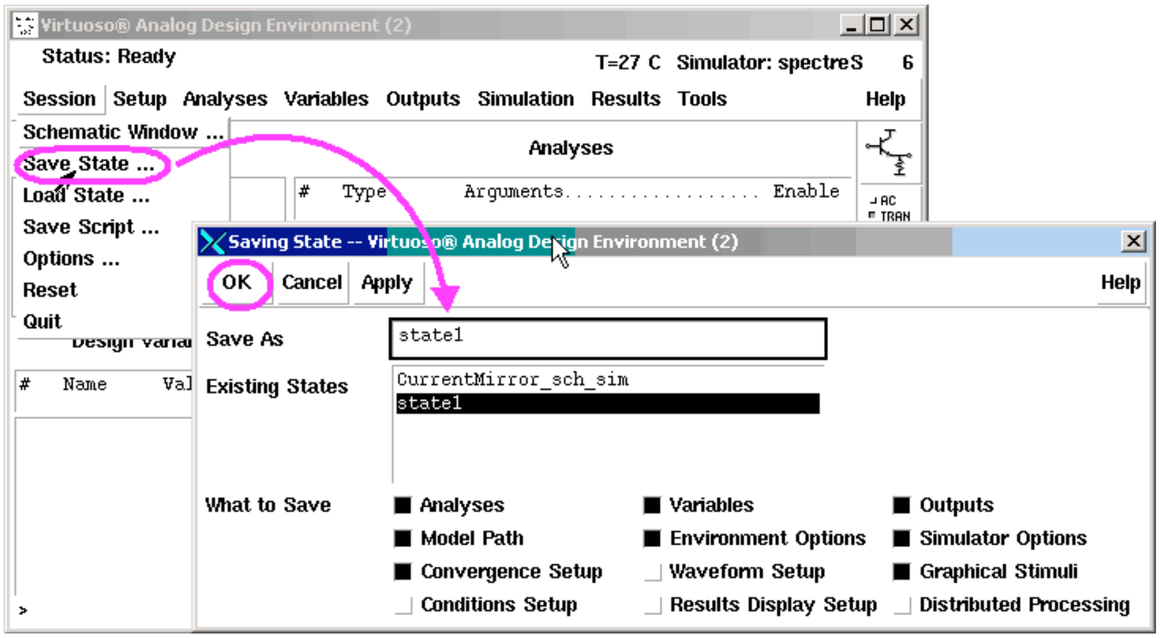


Figure 56: Output Graphic.

From the Transient Response, we see that the I_ref_in and I_ref_out are almost aligning but not exactly (I_ref_in is 80mA, and I_ref_out is 79.59mA), same as in the dc response. They start getting closer as the V_{gs} is getting higher. Can you tell why it is like that?

If the simulations worked, go back to the *Analog Design Environment* window and choose *Session…Save State …* to save the simulation setup as in the figure below:


Virtuoso® Analog Design Environment (2)
Status: Ready
T=27 C Simulator: spectreS 6
Session Setup Analyses Variables Outputs Simulation Results Tools Help
Schematic Window ...
Save State ...
Load State ...
Save Script ...
Options ...
Reset
Quit
Analyses
# Type Arguments.................. Enable
AC
TRAN
Design Varia
# Name Val
Saving State -- Virtuoso® Analog Design Environment (2)
OK
Cancel
Apply
Help

Save As	state1		
Existing States	CurrentMirror_sch_sim		
	state1		
What to Save	■ Analyses	■ Variables	■ Outputs
	■ Model Path	■ Environment Options	■ Simulator Options
	■ Convergence Setup	□ Waveform Setup	■ Graphical Stimuli
	□ Conditions Setup	□ Results Display Setup	□ Distributed Processing

Also go to CIW and choose *Options... Save defaults* and click *OK* on the pop-up.

Section 3: Layout

Pre-Layout of the CurrentMirror

This is a critical step. Once this step is done, the actual layout step will be very easy. Here is what we need to do:

Sizing:

First, we look at the S ratio of our transistors. For I_ref_in and I_ref_out, we use the minimum sizes, which S is 1. Then for I_refx2, we use S = 2 for double the current. For the same purpose, we use S = 4 for I_refx4, and S = 8 for I_refx8.

Then we look at the bubble schematic to pick which transistor leaf we are going to use.

Bubble Schematic:

Use different markers to fill in the joints and wires. Figure 57 shows the complete fill_in_the_blank of the bubble schematic.

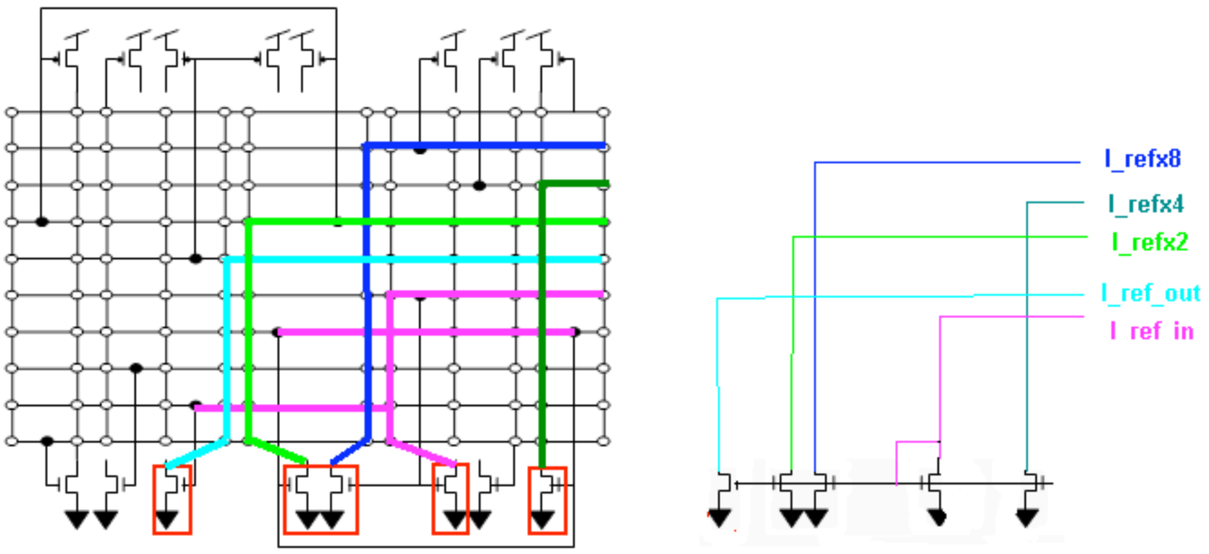
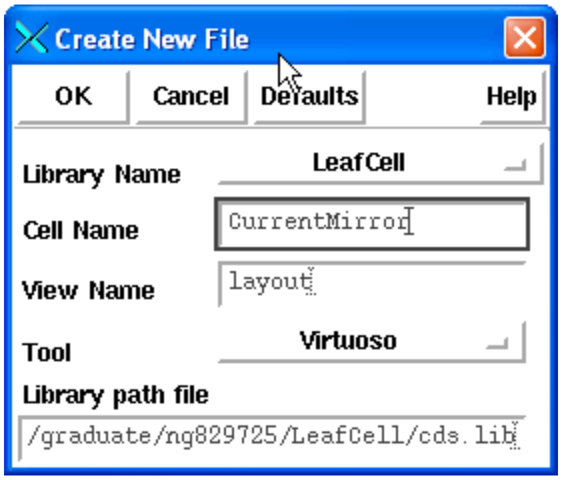


Figure 57: Bubble schematic of the CurrentMirror.

After completing the bubble schematic, we may then go to the actual layout.

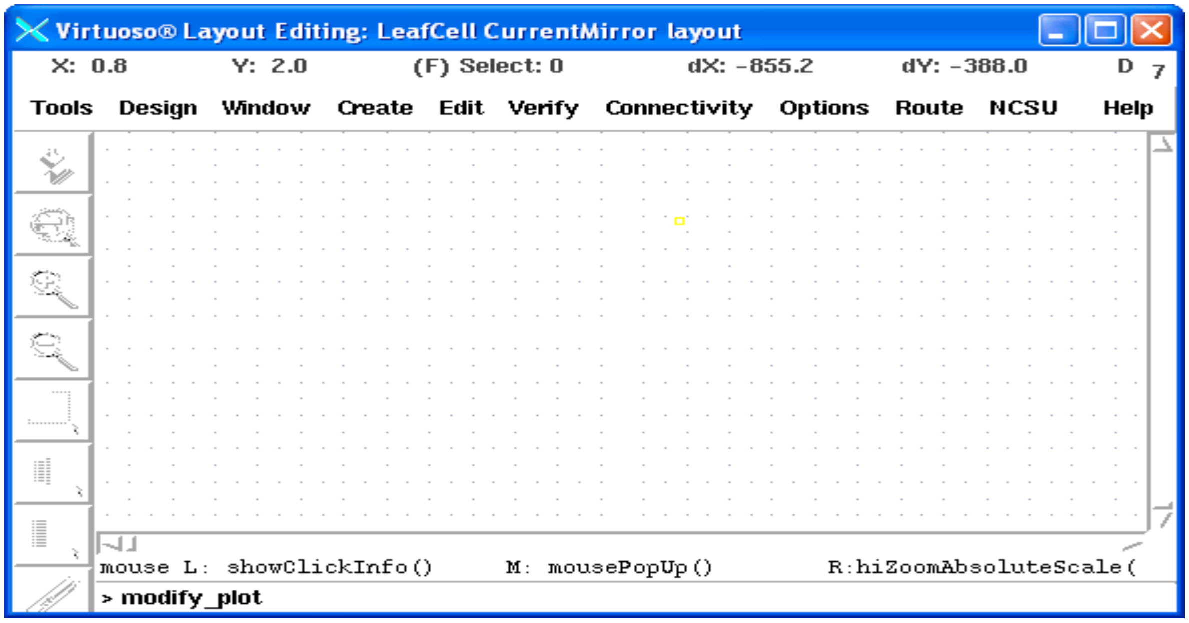
Layout in Leaf Cell:

First we need to create a layout view of our CurrentMirror. Go to the *Library Manager* and create a new cell view according to Figure 58.



Library Name: LeafCell
Cell Name: CurrentMirror
View Name: layout
Tool: Virtuoso
Library path file: /graduate/ng829725/LeafCell/cds.lib

Figure 58: Creating a layout view



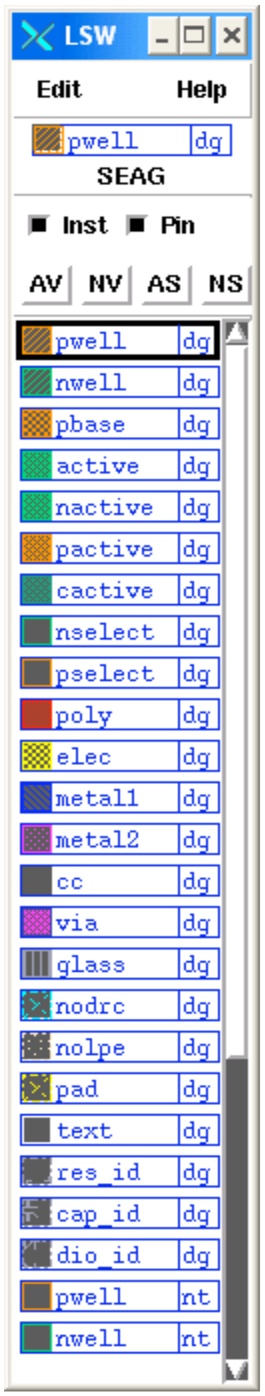
X: 0.8 Y: 2.0 (F) Select: 0 dX: -855.2 dY: -388.0 D 7

```
mouse L: showClickInfo()      M: mousePopUp()      R:hiZoomAbsoluteScale(
> modify_plot
```

Figure 59: Layout Editor

Figure 59 shows the layout editor that you used in the cell design tutorial.

Notice that the layers are slightly different in the *LSW* (Fig 60). This is the SpartanN process.


LSW

Edit Help

pwell dg

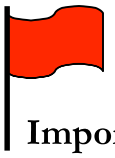
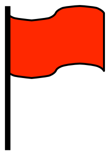
SEAG

Inst Pin

AV NV AS NS

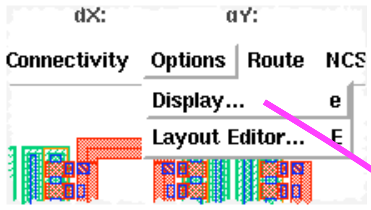
pwell	dg
nwell	dg
pbase	dg
active	dg
nactive	dg
pactive	dg
cactive	dg
nselect	dg
pselect	dg
poly	dg
elec	dg
metal1	dg
metal2	dg
cc	dg
via	dg
glass	dg
nodrc	dg
nolpe	dg
pad	dg
text	dg
res_id	dg
cap_id	dg
dio_id	dg
pwell	nt
nwell	nt

Figure 60: LSW

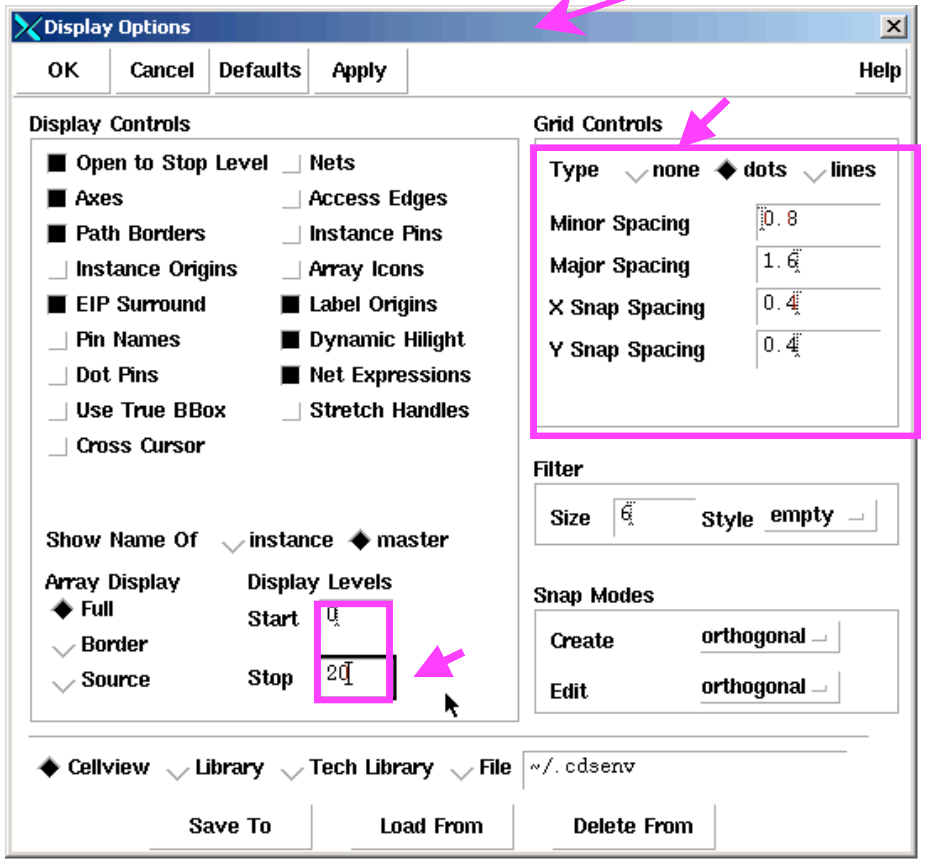



Important Notes:

Make sure you check your *Display Options* every time you open a *Layout Editor* to avoid placing and aligning problems.



From the Layout Editor, click on Options…Display.


Display Options

OK Cancel Defaults Apply Help

Display Controls

- [✓] Open to Stop Level
- [✓] Axes
- [✓] Path Borders
- [ ] Instance Origins
- [✓] EIP Surround
- [ ] Pin Names
- [ ] Dot Pins
- [ ] Use True BBox
- [ ] Cross Cursor
- [ ] Nets
- [ ] Access Edges
- [ ] Instance Pins
- [ ] Array Icons
- [✓] Label Origins
- [✓] Dynamic Hilight
- [✓] Net Expressions
- [ ] Stretch Handles

Show Name Of: instance [✓ master]

Array Display: [✓ Full] Border Source

Display Levels: Start 0, Stop 20

Grid Controls

Type: none [✓ dots] lines

Minor Spacing	0.8
Major Spacing	1.6
X Snap Spacing	0.4
Y Snap Spacing	0.4

Filter

Size 6 Style empty

Snap Modes

Create orthogonal

Edit orthogonal

[✓ Cellview] Library Tech Library File ~/.cdsenv

Save To Load From Delete From

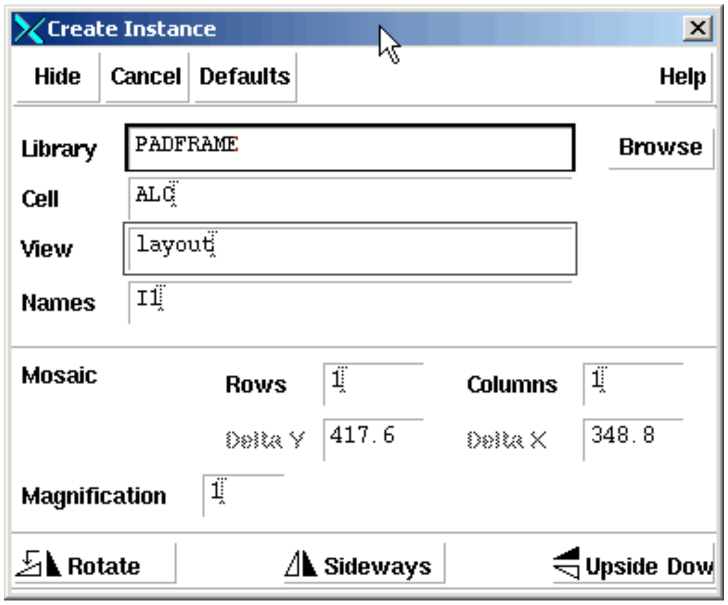
Figure 61: Setting Display Options

To set the display so that all the layers will appear, go to *Options... Display* in the layout editor. Set the pop-up according to Figure 61 and click ok.

Also, make sure you set the *Grid Controls* as in Figure 61 to minimize the gird errors while doing the DRC.

Routing the Leaf Cell Block:

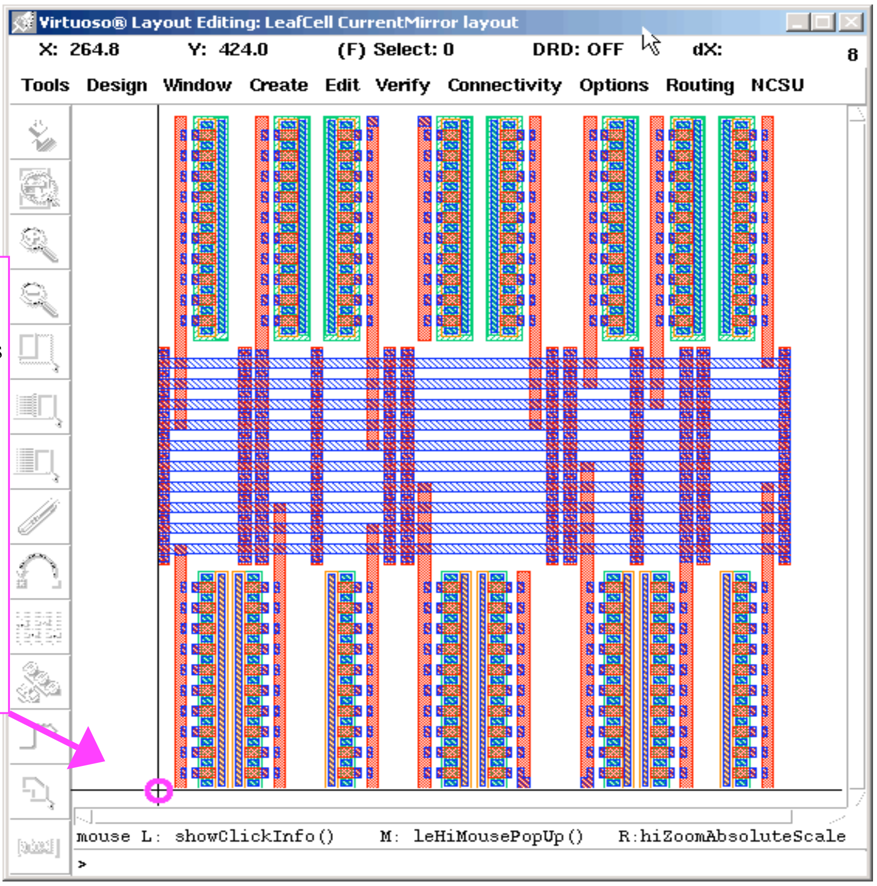
Go to *Create…Instance…*, or press *i* in the *Layout Editor*, and then fill out the form as in Figure 62. The Leaf Cell Block instance is called *ALC* in the *PADFRAME* library.



Library	PADFRAME
Cell	ALC
View	layout
Names	I1
Mosaic	
Rows	1
Columns	1
Delta Y	417.6
Delta X	348.8
Magnification	1

Figure 62: Adding the Leaf Cell Block into the layout.

Stamp the ALC down into the Layout Editor (Figure 63).



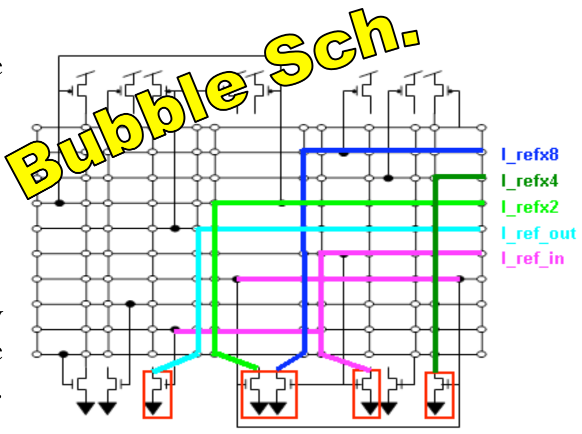
```
mouse L: showClickInfo()     M: leHiMousePopUp()     R:hiZoomAbsoluteScale
>
```

Use Reference Point!

While stamping down instances, use the origin/cross point for the reference point. It is very important for minimizing grid errors, which will give you a big headache.

Make sure you zoom into the origin to double check if the instance is exactly aligned horizontally and vertically. If not, move the instance until it is.

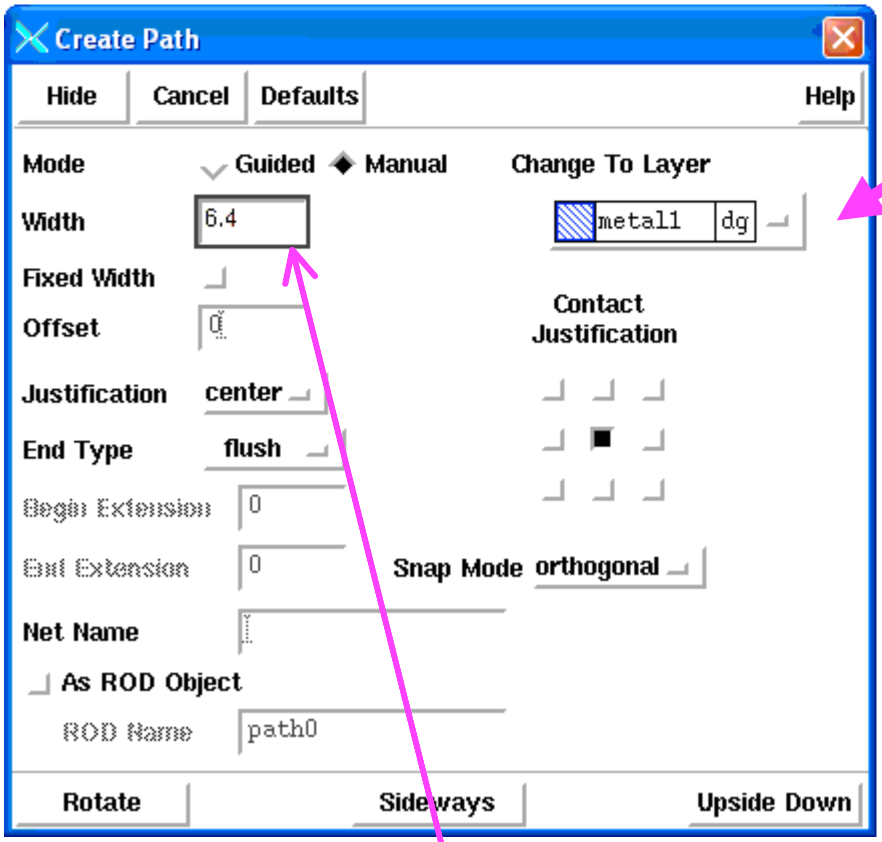
Figure 63: stamp down the ALC



Now, referring to the bubble schematic we did in the pre-layout, we start to layout the metals.

1. Zoom in to the left most cells we are using (referring to the bubble schematic, it is the one for *I_ref_out*) as in Figure 66.

2. Find and click on *Metal1* in the *LSW* window (Figure 64 shows part of the LSW window). Come back to the layout window, go to *Create….Path*. The *Create Path* window will show up (Figure 65).


Create Path

Hide | Cancel | Defaults | Help

Mode: Guided [✓ Manual]
Width: 6.4
Fixed Width
Offset: 0
Justification: center
End Type: flush
Begin Extension: 0
End Extension: 0
Net Name:
As ROD Object
ROD Name: path0
Change To Layer: metal1 dg
Contact Justification
Snap Mode: orthogonal

Rotate | Sideways | Upside Down

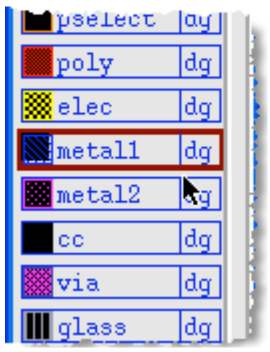


Figure 64: Choosing Metal1 in the LWS window

Figure 65: Create path

3. Fill in the width of the path to be ***6.4*** in the form (Figure 65). **Note:** we choose the width to be as large as possible so that the path can cover as wide as the contacts to minimize the resistance.

4. To connect the gates, the poly layers of signal path, point the mouse at the starting location of the path you are going to create, click once, then move the mouse at the ending location of the path and double click to finish the path (Figure 66).

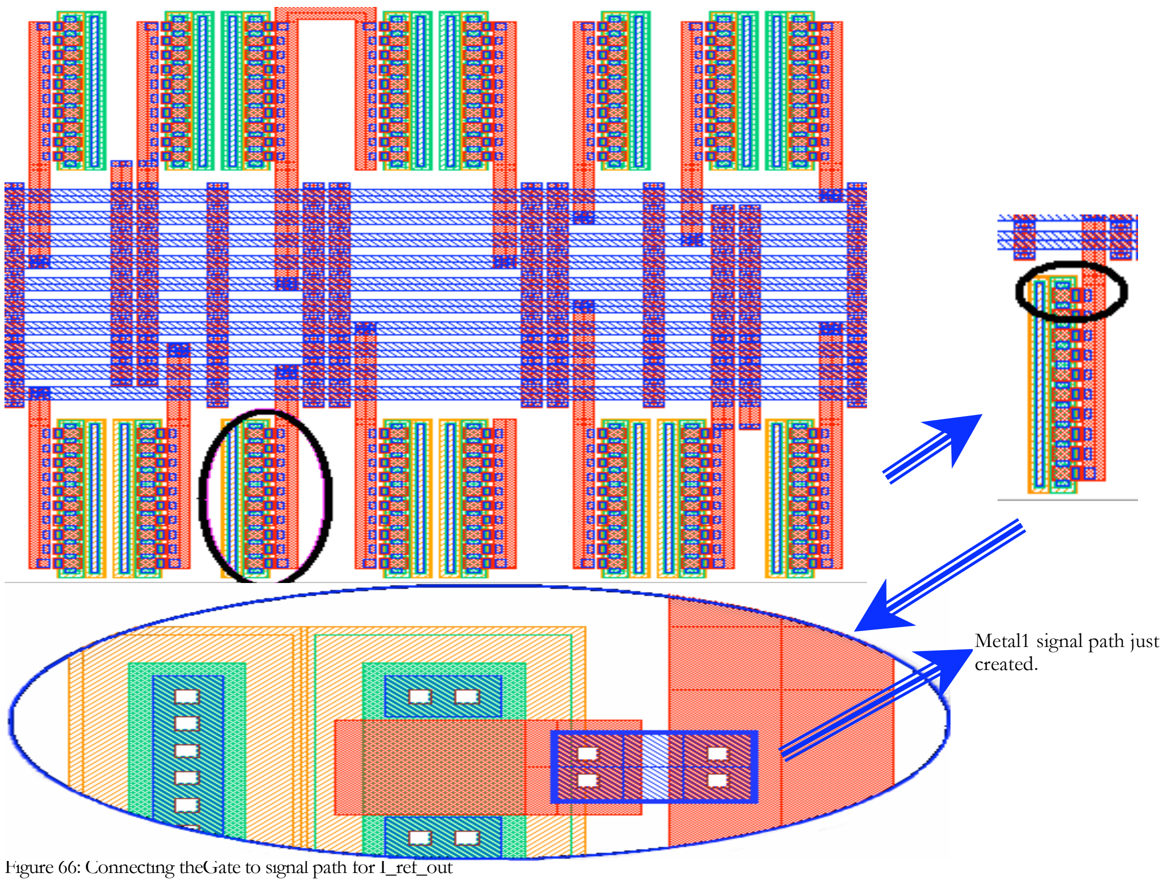
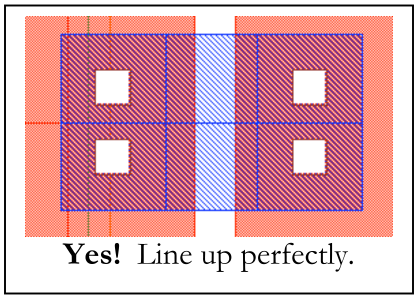


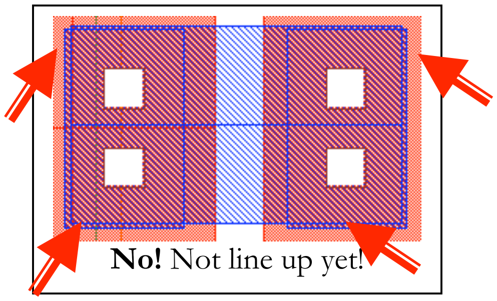
Figure 66: Connecting theGate to signal path for I_ref_out

Important Notes:

For every metal layer you put down, make sure it is aligned with the horizontal and vertical boundaries. If you have the right display setup as mentioned earlier, and careful with every metal layer you put down, you can get rid of the headache of going back to them for grid errors later after the DRC.



Yes! Line up perfectly.



No! Not line up yet!

5. Connect the *source* as well as *drain* using *path width* to be **3.2** (Figure 67). **Note:** For this connection, we use a narrower path width of **3.2** because if we still use width 6.4, it will overlap with the poly layers and violate the design rules of minimum separation between the same layers.

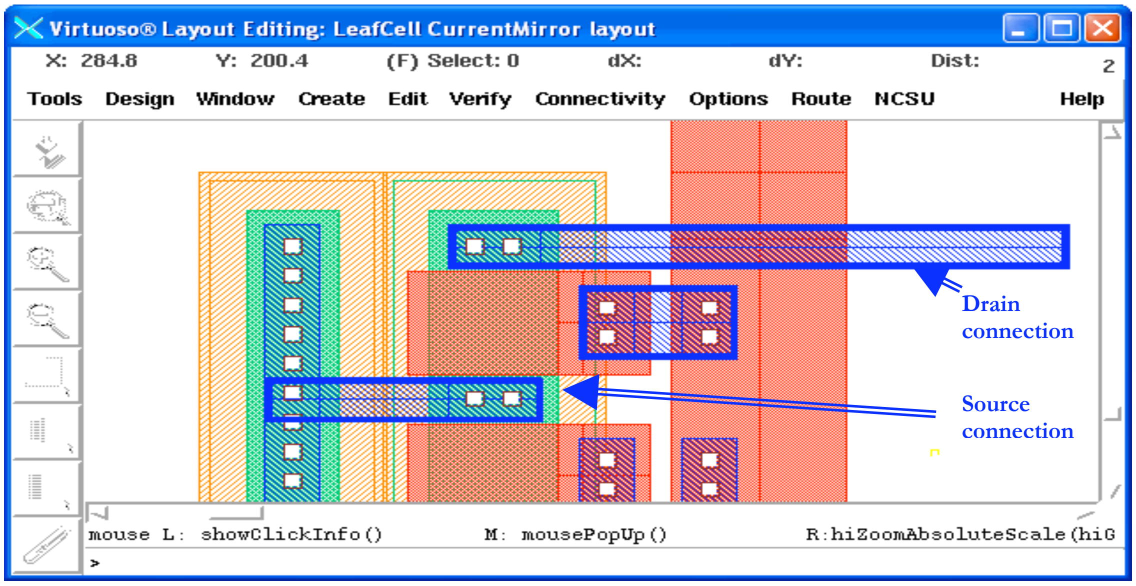


Figure 67: Connecting the Body to Source, Drain to Body.

6. Now we need to ground the transistors we aren't using. To do so, create paths of width ***6.4*** (same width as used in the connection of the signal path over poly), connect the gate to the source (Figure 68) so that when we ground the source later, we also ground the un-used transistors.

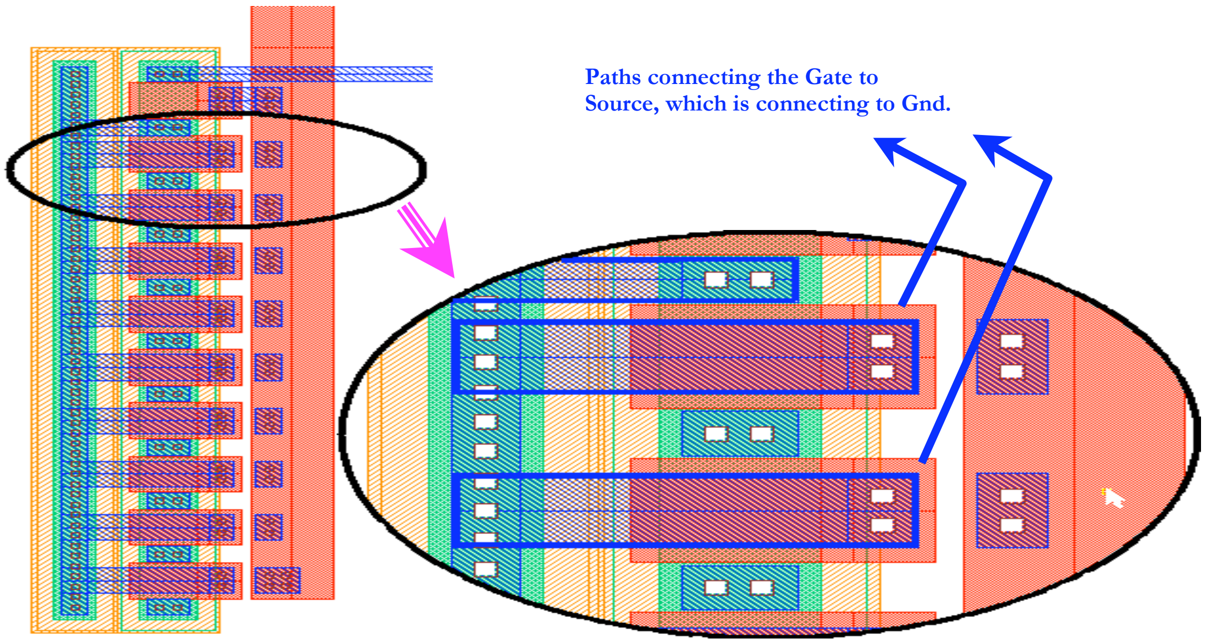

Paths connecting the Gate to Source, which is connecting to Gnd.

Figure 68: Grounding the un-used transistors (Shorting the gate to gnd!).

Why we need the paths for grounding the unused nmos?

The LVS may pass without those grounding paths sometimes. However, we must keep them because if we leave them floating, they will act like an antenna picking up unwanted signals, which is noise, so that the circuit might not work.

7. Now we've finished the inner connections of the NMOS leaf for I_ref_out, then we zoom into the next leaf, which is the one for I_refx2 according to the Bubble schematic.

8. Repeat steps 2 to 6 to create paths connecting the transistors for I_refx2. Make sure they are connected in parallel (as shown in Ch1) in order to have S = 2. The finished one should look like Figure 69.

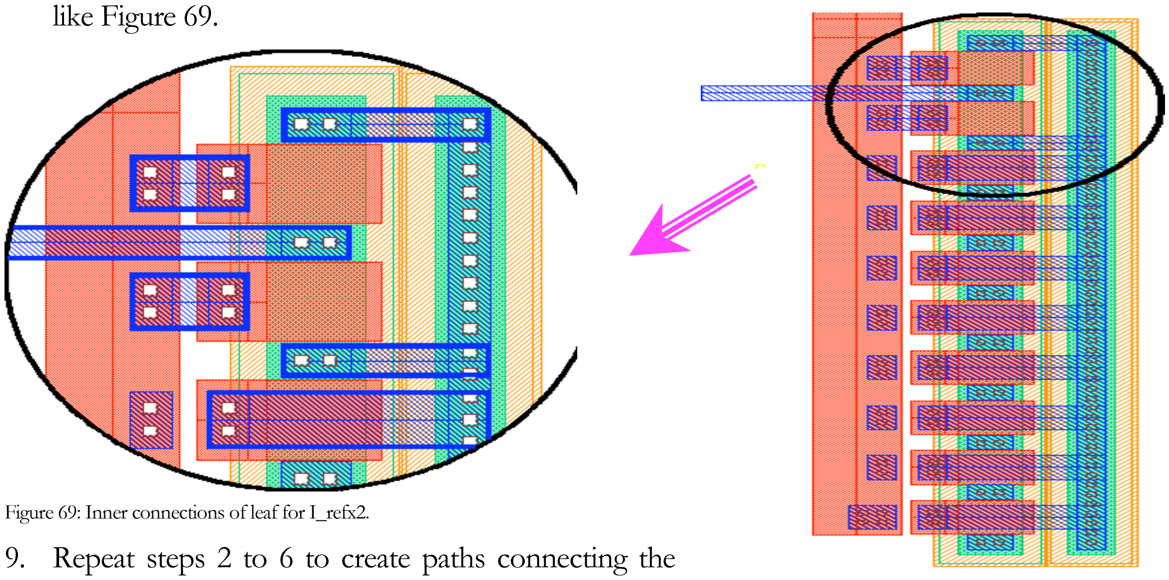


Figure 69: Inner connections of leaf for I_refx2.

9. Repeat steps 2 to 6 to create paths connecting the transistors for I_refx8. Make sure they are connected in parallel (as shown in Ch1) in order to have S = 8. The finished one should look like Figure 70.

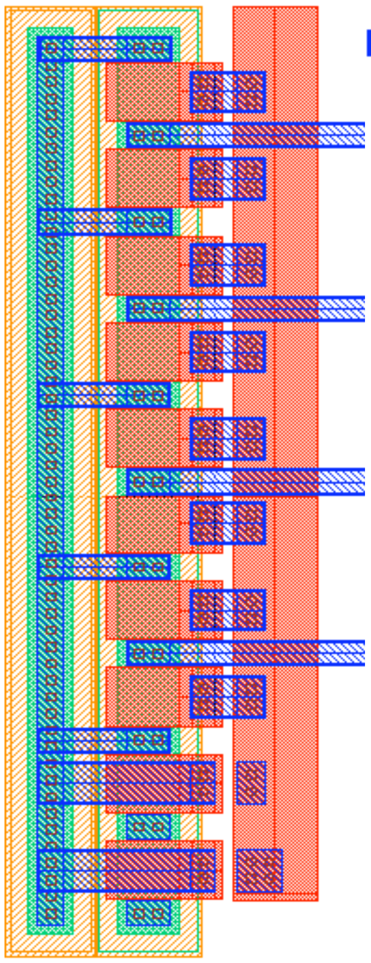


Figure 70: Inner connection for leaf I_refx8.

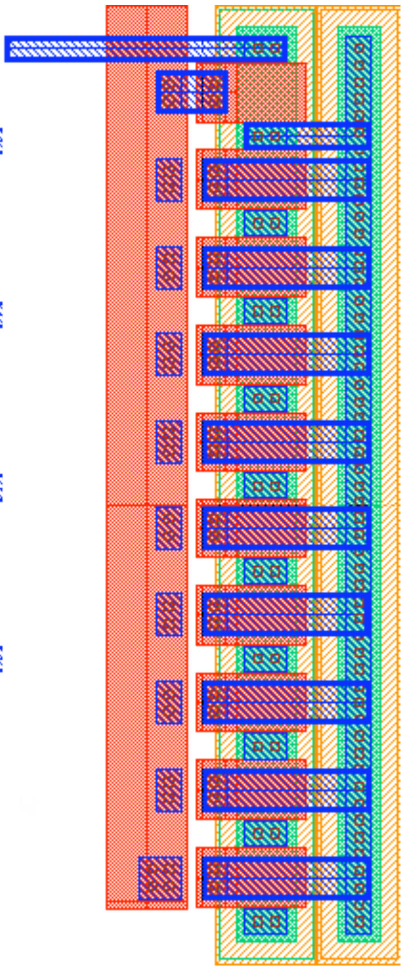


Figure 71: Inner connection for leaf I_ref_in

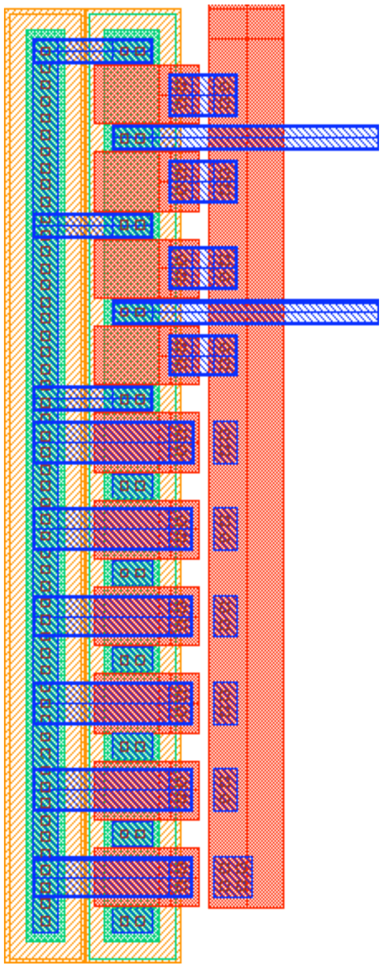
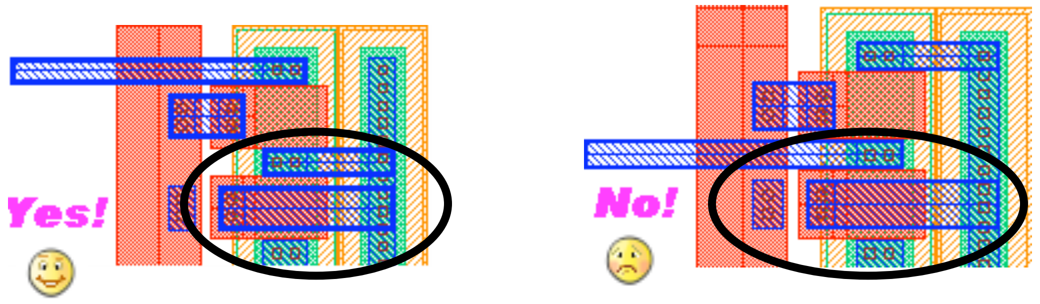


Figure 72: Inner connection for leaf I_refx4

10. Repeat steps 2 to 6 to create paths connecting the transistors for *I_ref_in*. The finished one should look like Figure 71.

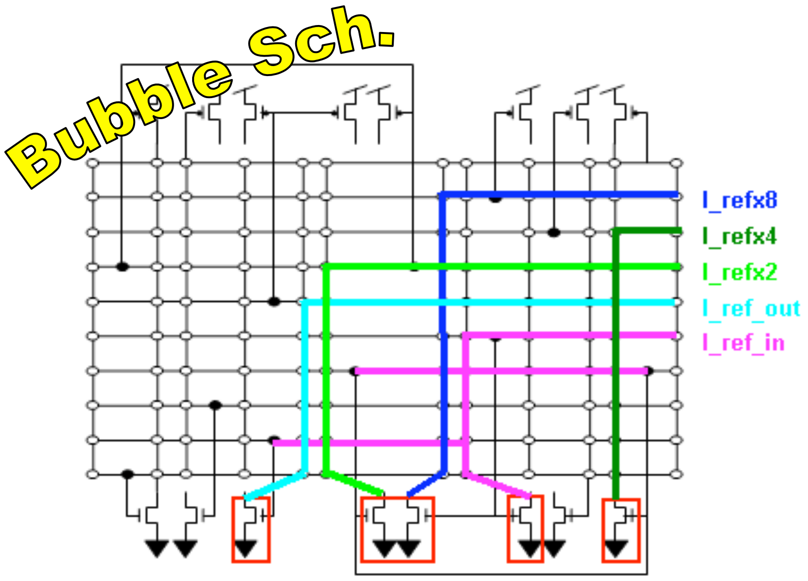
Important Notes:

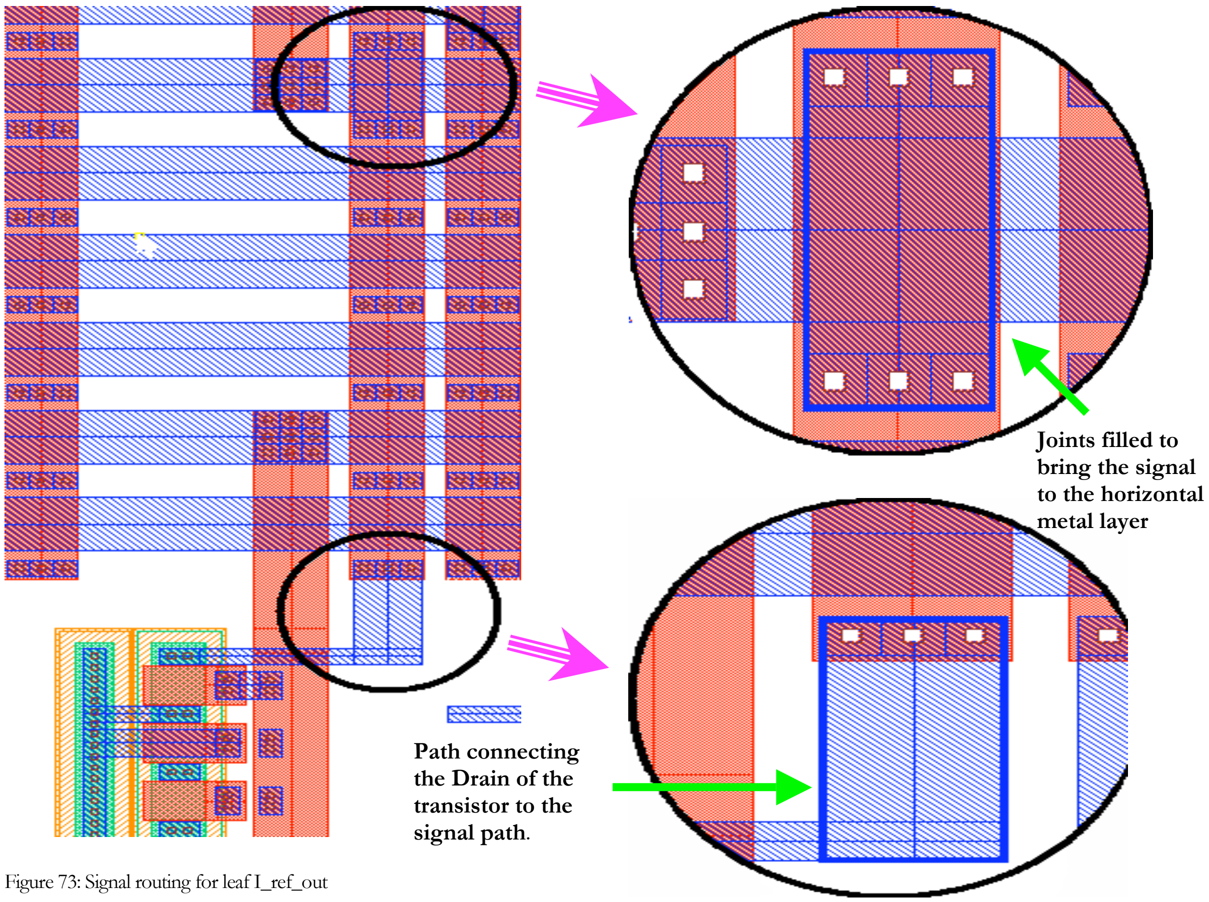
The S/D of the transistor right next to the *grounding path* has to be connected to gnd! instead of the *signal path*(with potential level higher than gnd!). Otherwise, the LVS won't pass later.



11. Repeat steps 2 to 6 to create paths connecting the transistors for I_refx4. Make sure they are connected in parallel (as shown in Ch1) in order to have S = 4. The finished one should look like Figure 72.

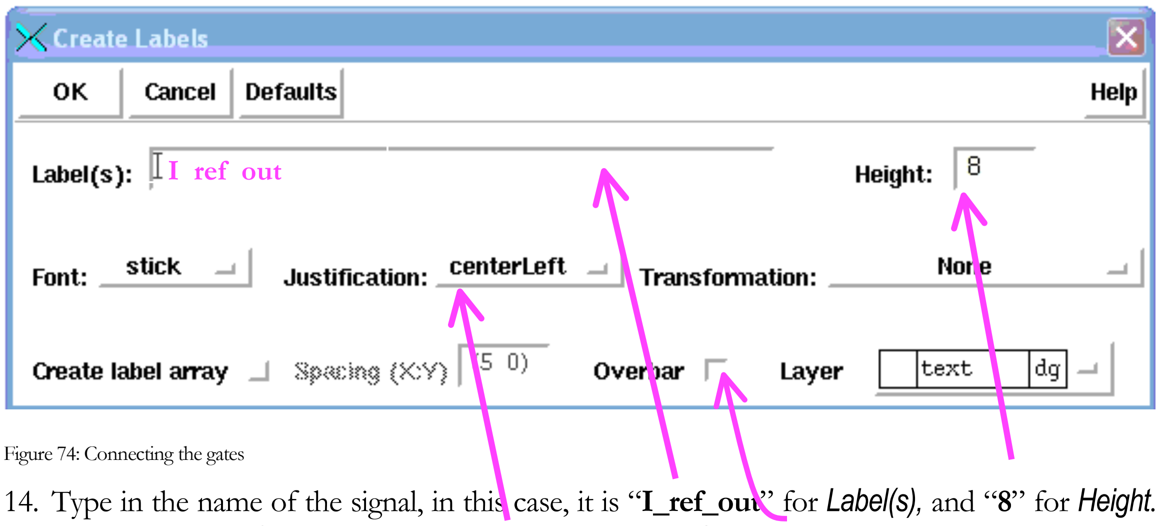
12. After the transistors are connected with proper sizes, we need to connect the leaves together. Again, we start from the very left, which referring to the Bubble Schematic is signal I_ref_out. Create path with width of **6.4,** connect the Drain to the joints as in Figure 73.




Joints filled to bring the signal to the horizontal metal layer
Path connecting the Drain of the transistor to the signal path.
Figure 73: Signal routing for leaf I_ref_out

It is very helpful to label the path after you are done with each signal path so that it will be easier while routing the signal paths outside the block.

13. Go to *Create…Label…*, a form as in Figure 74 will appear.



Label(s): I ref out
Height: 8
Font: stick
Justification: centerLeft
Transformation: None
Create label array
Spacing (X:Y) 5 0)
Overbar
Layer text dg

Figure 74: Connecting the gates

14. Type in the name of the signal, in this case, it is "**I_ref_out**" for *Label(s)*, and "**8**" for *Height*. Also, set the *Justification* to be **centerLeft** , and set the *Overbar* off in this case.

15. Click on *OK* and then stamp it right next to the **6th** row of metal line from the bottom as in Figure 75.

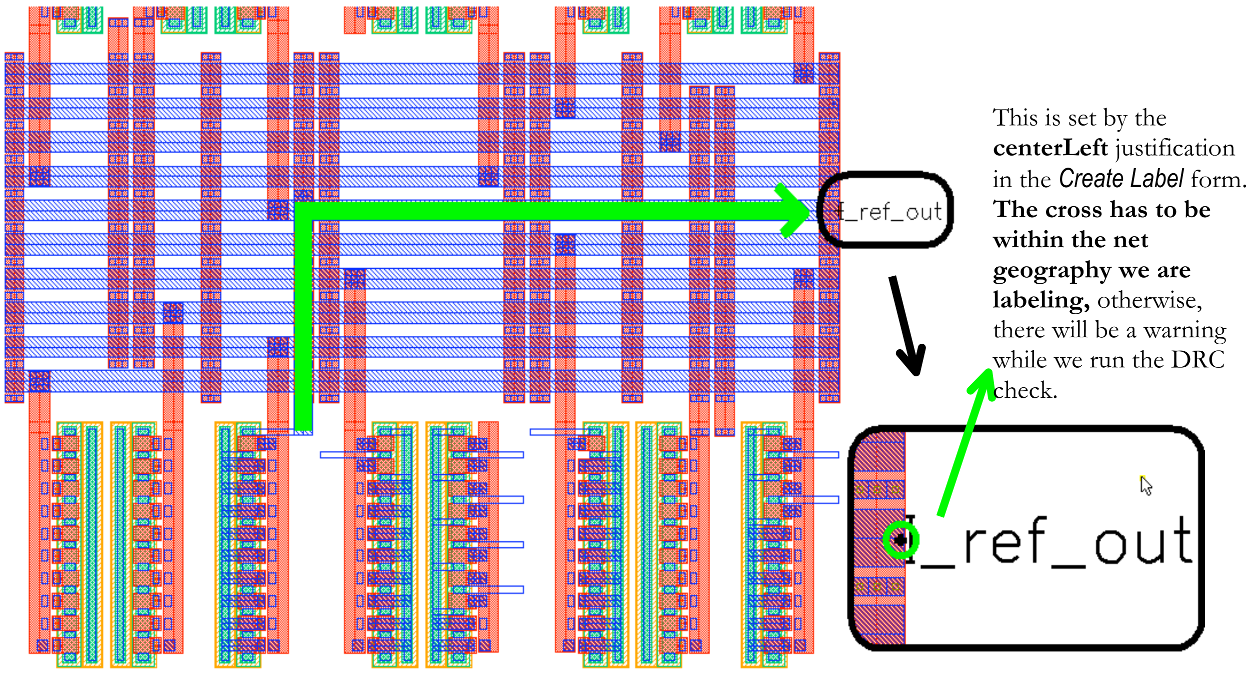

I_ref_out
I_ref_out
This is set by the **centerLeft** justification in the *Create Label* form. **The cross has to be within the net geography we are labeling,** otherwise, there will be a warning while we run the DRC check.

Figure 75: Creating label for signal I_ref_out

16. Repeat step 12 to 15 for the next leaf, which according to the Bubble Schematic is **I_refx2** (Figure 76). Remember to create a label right next to the **7th** row of metal line for it (Figure 76).

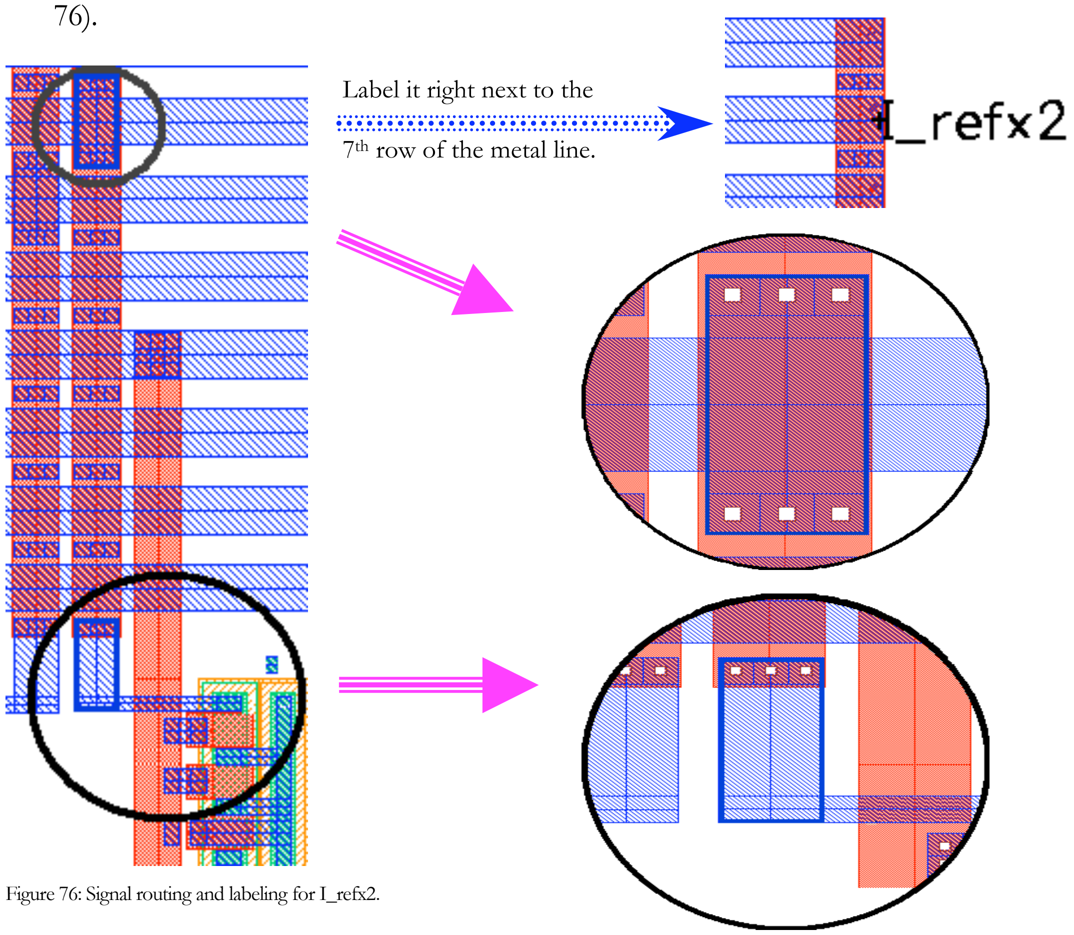


Figure 76: Signal routing and labeling for I_refx2.

17. Repeat step 12 to 15 for the next leaf, which according to the Bubble Schematic is **I_refx8** (Figure 77). Remember to create a label right next to the **9th** row of metal line (from the bottom) for it (Figure 77).

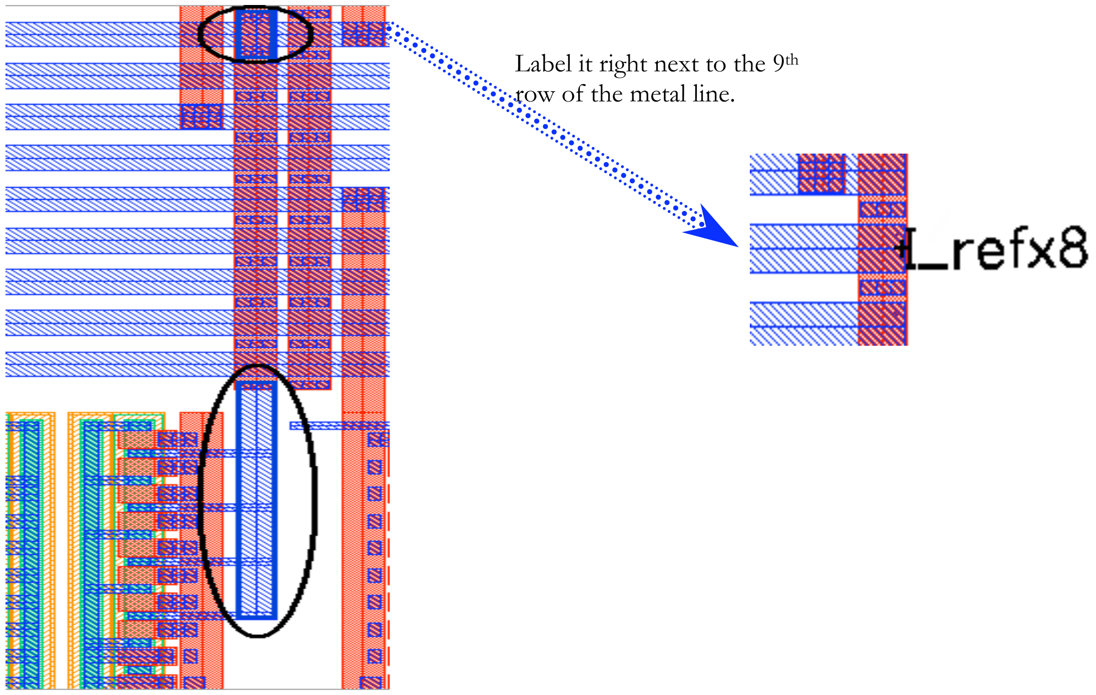


Figure 77: Signal routing and labeling for I_refx8.

18. Repeat step 12 to 15 for the next leaf, which according to the Bubble Schematic is **I_ref_in** (Figure 78). Remember to create a label right next to the **5th** row of metal line (from the bottom) for it (Figure 78).

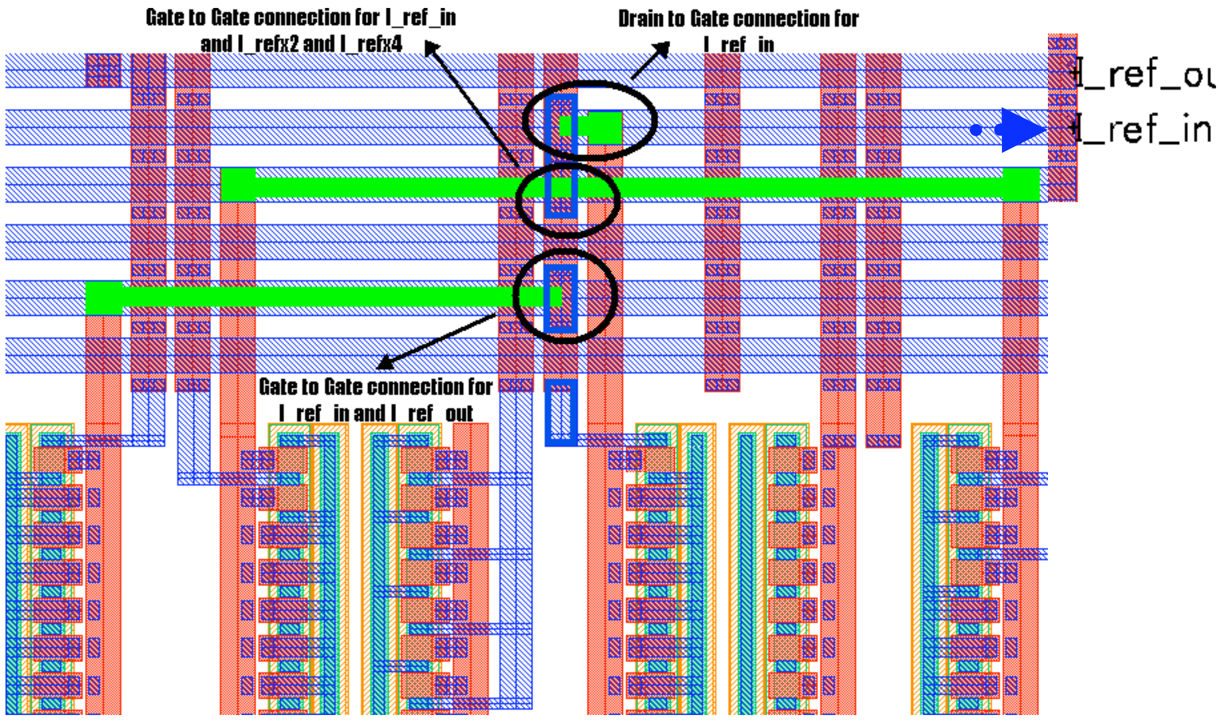

Gate to Gate connection for I_ref_in and I_refx2 and I_refx4
Drain to Gate connection for I_ref_in
I_ref_ou
I_ref_in
Gate to Gate connection for I_ref_in and I_ref_out

Figure 78: Signal routing and labeling for I_ref_in.

19. Also, make sure to make all the internal connections between **I_ref_in** and the gates of the transistors for the output currents, especially the gate connection between **I_refx8** and **I_ref_in** as in Figure79 a.

20. Repeat step 12 to 15 for the last leaf, which according to the Bubble Schematic is **I_refx4** (Figure 79). Remember to create a label right next to the **8th** row of metal line (from the bottom) for it (Figure 79).

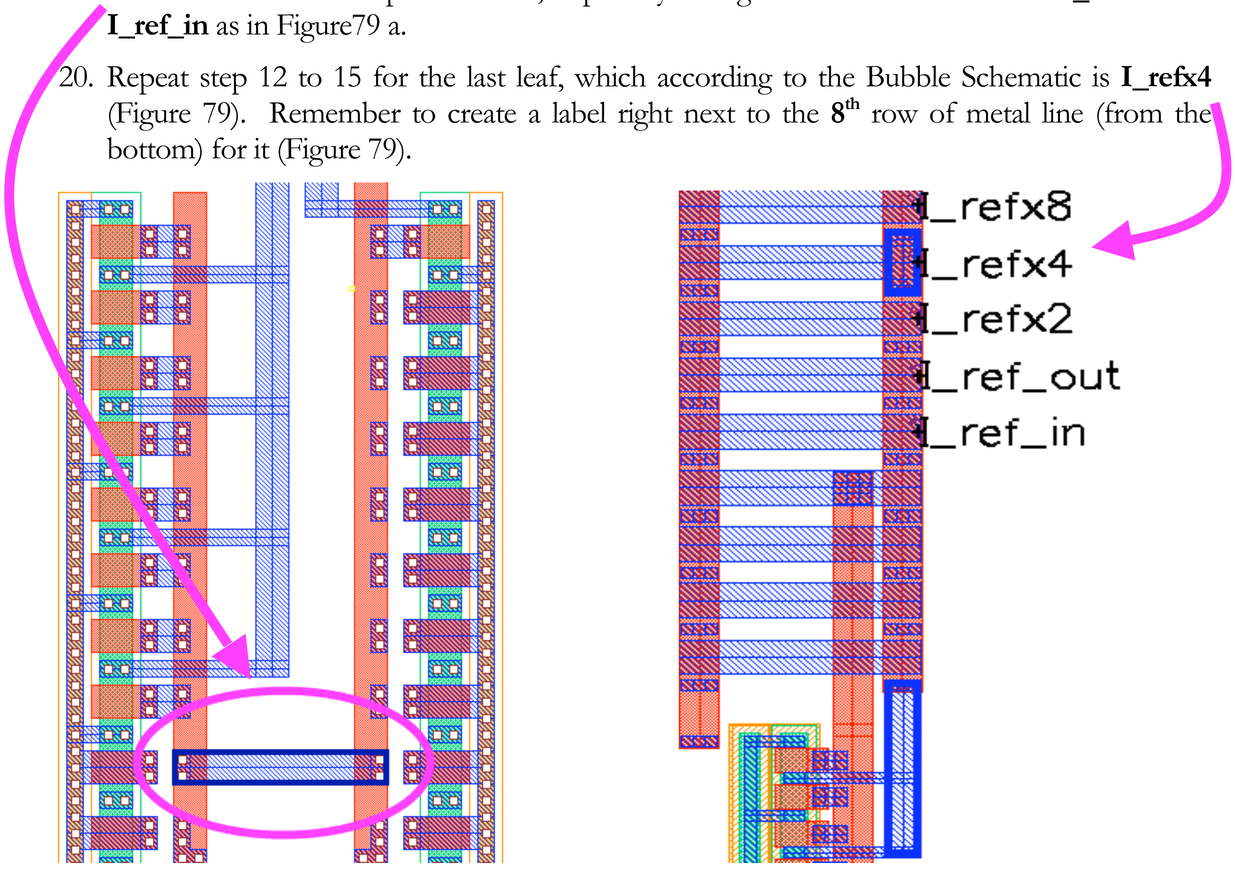


Figure 79: a) Gate connection between I_refx8 and I_ref_in. b) Signal routing and labeling for I_refx4.

Now the connections between the transistors are done. It is time to connect the *gnd* net.

21. Zoom to the bottom of the source of the transistors as shown in Figure 80. Connect them together and label them *gnd* as well as in Figure 81.

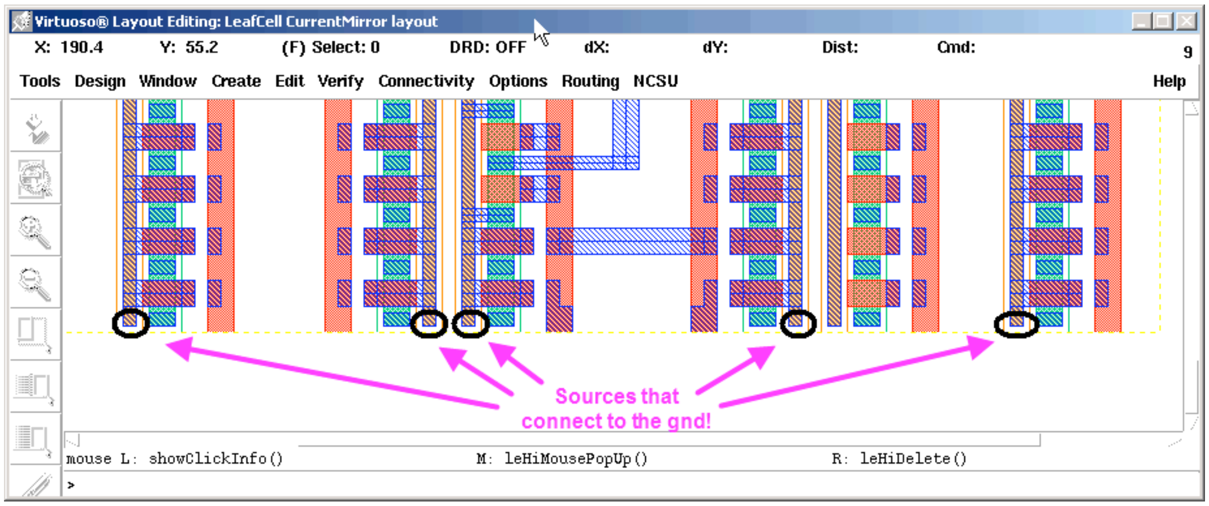

Virtuoso® Layout Editing: LeafCell CurrentMirror layout
X: 190.4 Y: 55.2 (F) Select: 0 DRD: OFF dX: dY: Dist: Cmd: 9
Tools Design Window Create Edit Verify Connectivity Options Routing NCSU Help
Sources that connect to the gnd!
mouse L: showClickInfo() M: leHiMousePopUp() R: leHiDelete()
>

Figure 80: Zooming into the sources needed to be connected to gnd!.

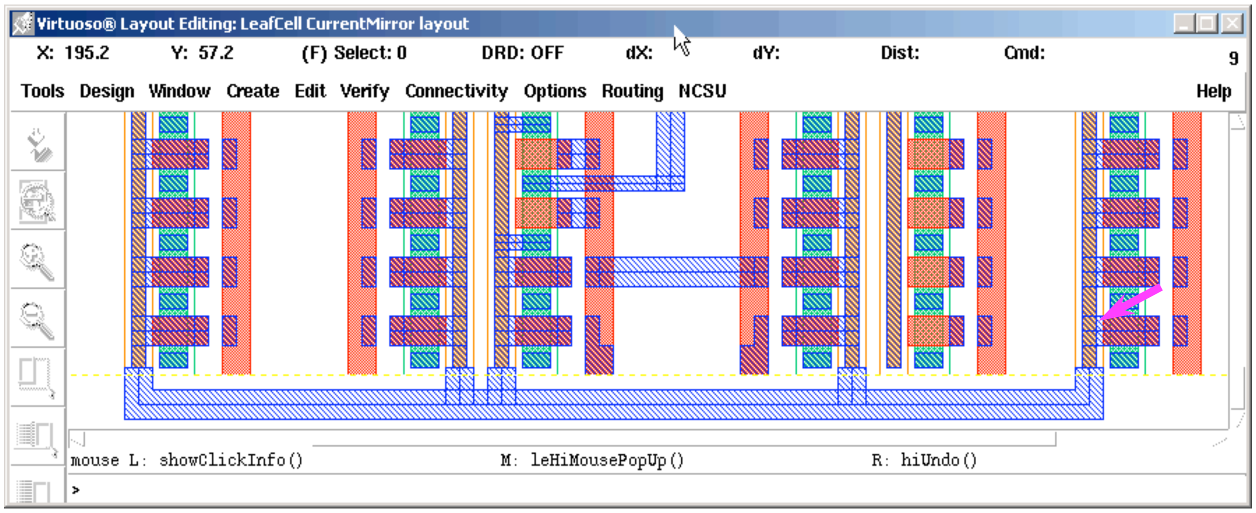
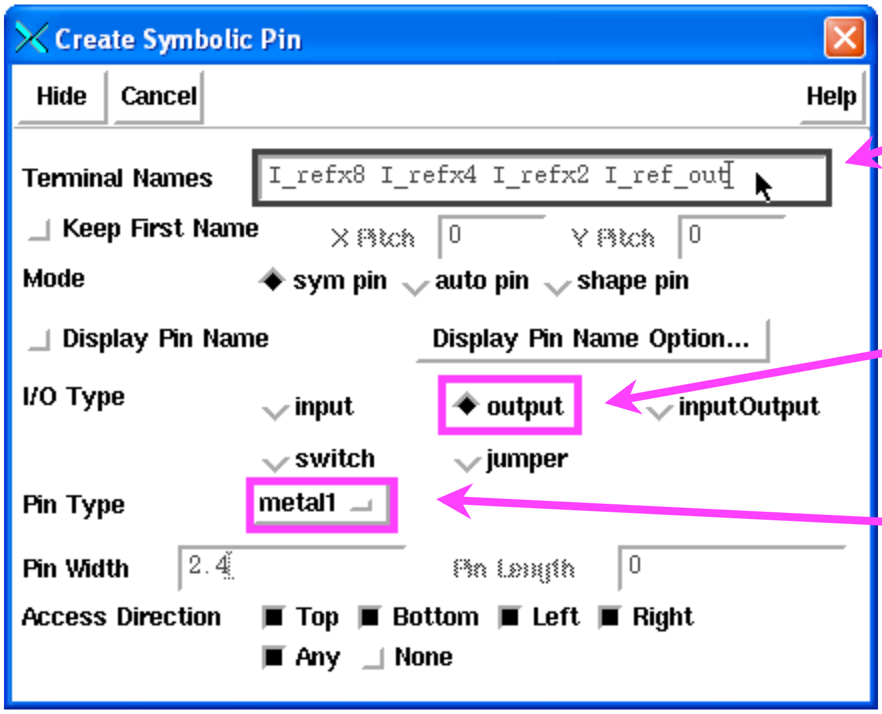

Virtuoso® Layout Editing: LeafCell CurrentMirror layout
X: 195.2 Y: 57.2 (F) Select: 0 DRD: OFF dX: dY: Dist: Cmd: 9
Tools Design Window Create Edit Verify Connectivity Options Routing NCSU Help
mouse L: showClickInfo() M: leHiMousePopUp() R: hiUndo()

Figure 81: Wiring the *gnd* net and label it "gnd!"

All the metal wirings are now done. Our labels don't associate with anything physical in the system. To let the system know which net is which, we need to insert the pins to point out the nets with the names we labeled previously.

22. To Create Input/Output Pins: go to *Create…Pin…* fill in the form as in Figure 82 for all the outputs. **Note: make sure the Pin type is METAL 1.**


Create Symbolic Pin

Hide Cancel Help

Terminal Names: I_refx8 I_refx4 I_refx2 I_ref_out

Keep First Name [ ] X Pitch 0 Y Pitch 0

Mode: [✓ sym pin] auto pin shape pin

Display Pin Name [ ] Display Pin Name Option...

I/O Type: input [✓ output] inputOutput switch jumper

Pin Type: metal1

Pin Width: 2.4 Pin Length: 0

Access Direction: ■ Top ■ Bottom ■ Left ■ Right ■ Any ☐ None

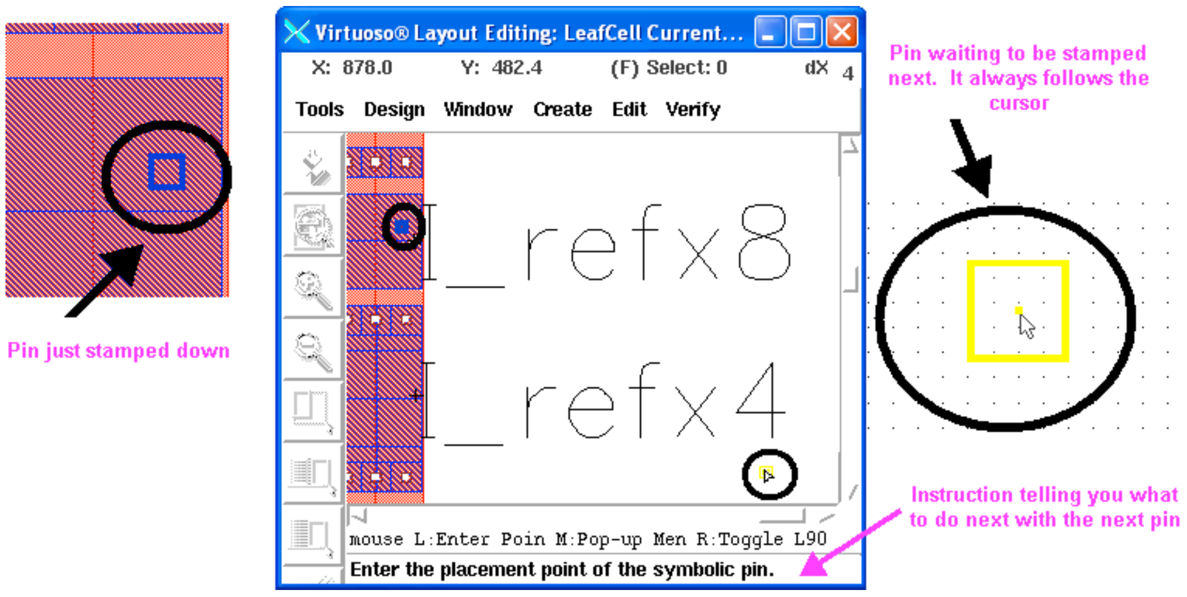
Names for all outputs. (Or you can put in all terminal names, but make sure to change the I/O types while stamping down the pins.

Make sure you check the right *I/O Type* for the terminal names.

Make sure the *Pin Type* to be. **metal1**

Figure 82: Adding the output pins.

23. Click once to stamp the first pin down (Figure 83), which is terminal I_refx8.

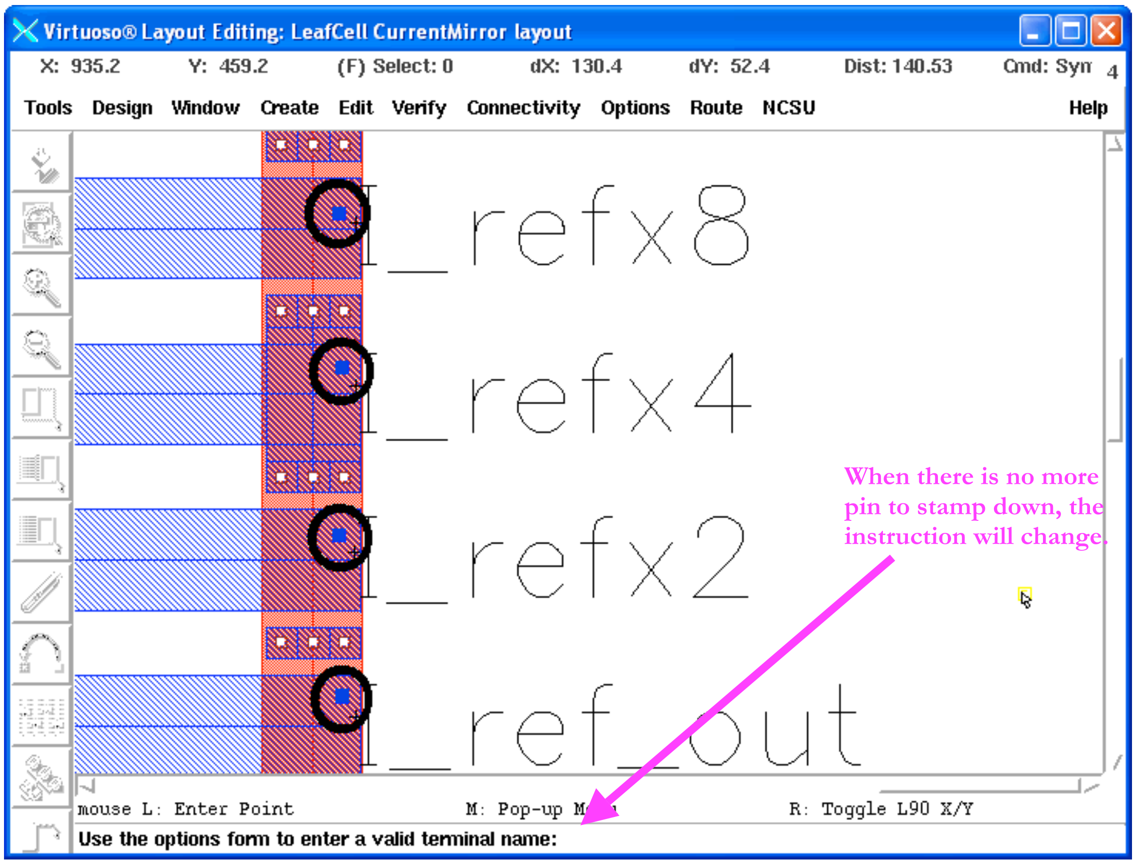

Pin just stamped down
Virtuoso® Layout Editing: LeafCell Current...
X: 878.0 Y: 482.4 (F) Select: 0 dX 4
Tools Design Window Create Edit Verify
I_refx8
I_refx4
Pin waiting to be stamped next. It always follows the cursor
Instruction telling you what to do next with the next pin

mouse L:Enter Poin M:Pop-up Men R:Toggle L90

Enter the placement point of the symbolic pin.

Figure 83: Stamping the output pin *I_refx8*.

24. Stamp down the rest of the output pins in the same manner (Figure 84).

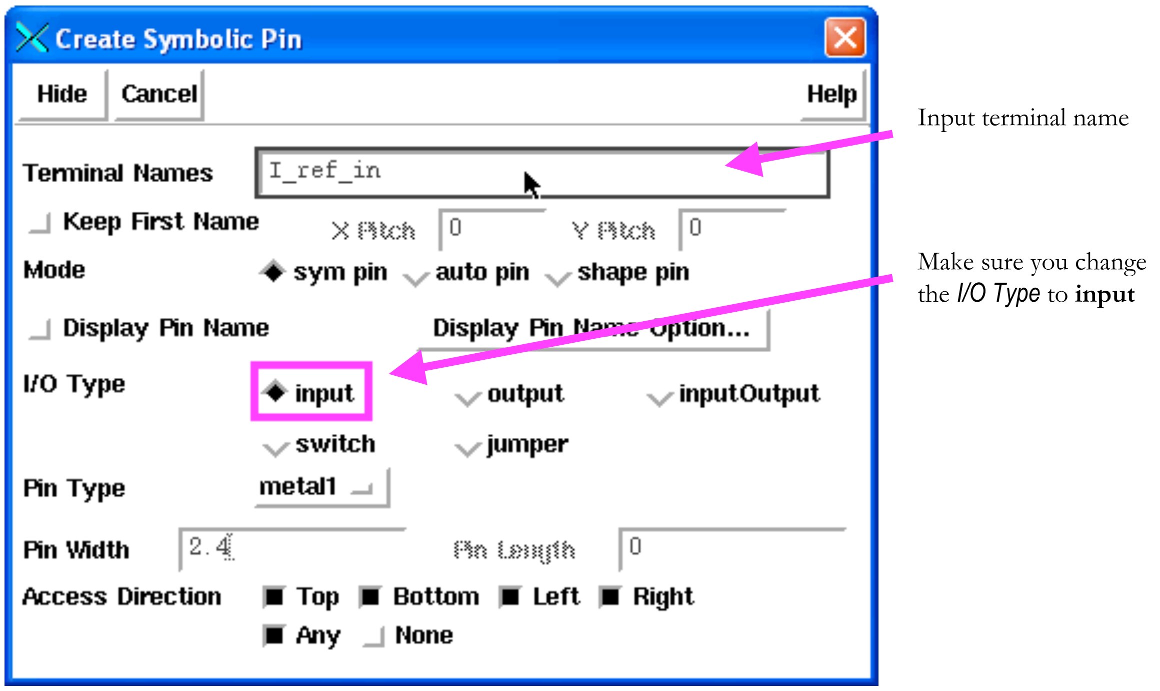

Virtuoso® Layout Editing: LeafCell CurrentMirror layout
X: 935.2 Y: 459.2 (F) Select: 0 dX: 130.4 dY: 52.4 Dist: 140.53 Cmd: Syn 4
Tools Design Window Create Edit Verify Connectivity Options Route NCSU Help
I_refx8
I_refx4
I_refx2
I_ref_out
When there is no more pin to stamp down, the instruction will change.

mouse L: Enter Point M: Pop-up M R: Toggle L90 X/Y

Use the options form to enter a valid terminal name:

Figure 84: Stamping the rest of output pins.

25. Go back to the *Create Symbolic Pin* form; fill in the input pin name, I_ref_in (Figure 85), and stamp it down as in Figure 86.


Create Symbolic Pin

Hide Cancel Help

Terminal Names I_ref_in

Keep First Name X Pitch 0 Y Pitch 0

Mode [✓ sym pin] auto pin shape pin

Display Pin Name Display Pin Name Option...

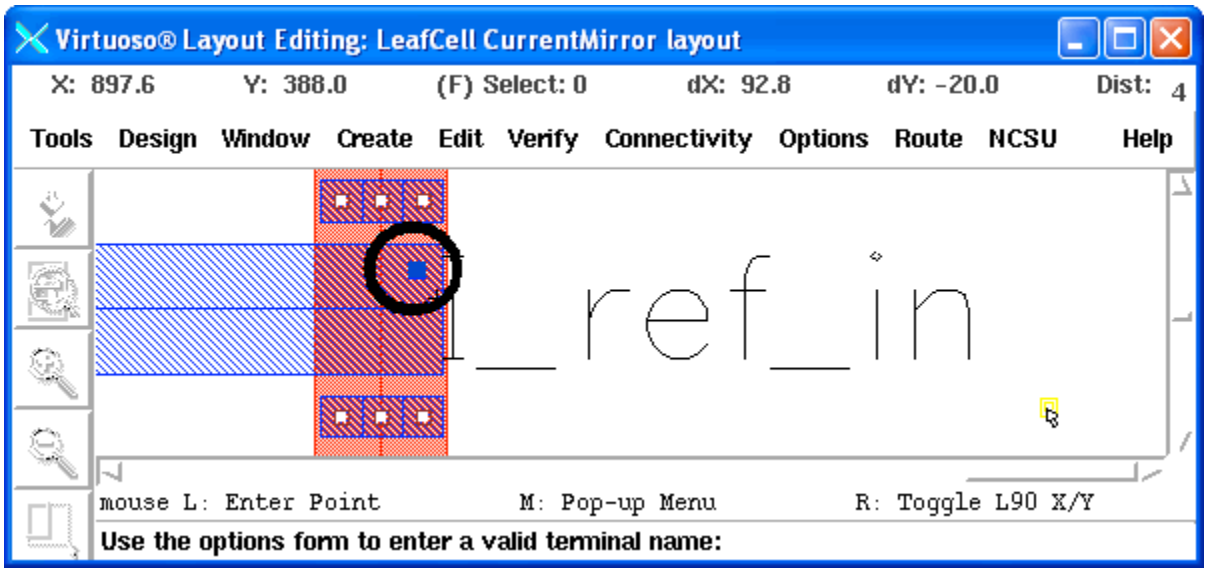
I/O Type [✓ input] output inputOutput switch jumper

Pin Type metal1

Pin Width 2.4 Pin Length 0

Access Direction ■ Top ■ Bottom ■ Left ■ Right ■ Any None

Figure 85: Adding the input pin.


Virtuoso® Layout Editing: LeafCell CurrentMirror layout

X: 897.6 Y: 388.0 (F) Select: 0 dX: 92.8 dY: -20.0 Dist: 4

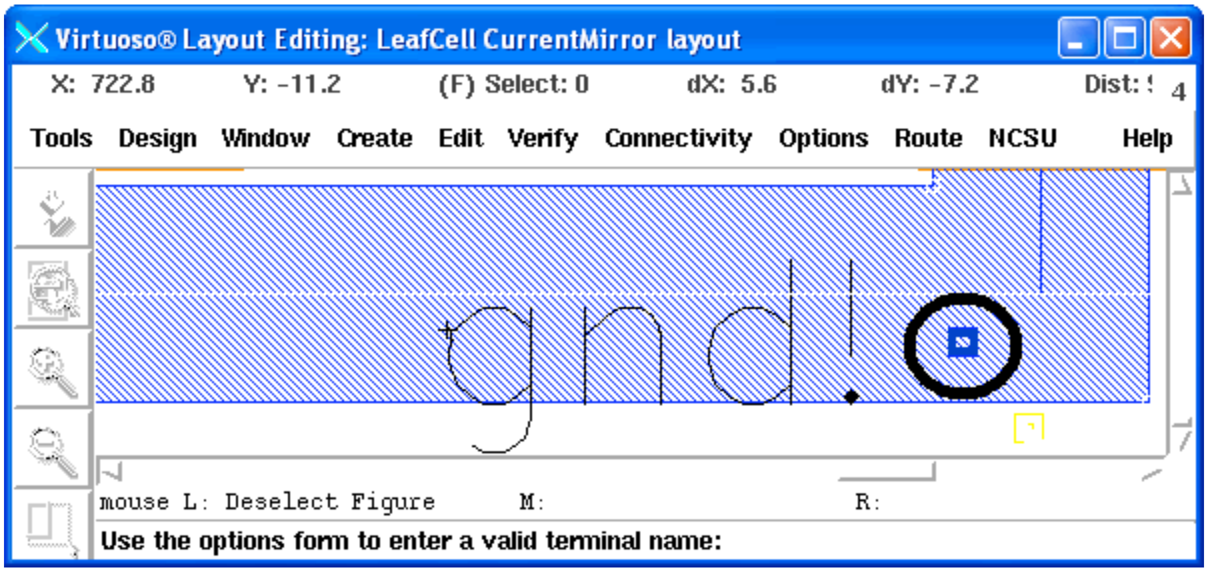
Tools Design Window Create Edit Verify Connectivity Options Route NCSU Help
I_ref_in

mouse L: Enter Point M: Pop-up Menu R: Toggle L90 X/Y

Use the options form to enter a valid terminal name:

Figure 86: Stamping down the input pin.

26. Go back to the *Create Symbolic Pin* form; fill in the *Terminal Name* as "**gnd!**"; make sure the *I/O type* is **inputOutput**, and then stamp it down as in Figure 87. Click on ESC to exit the adding pin mode.



mouse L: Deselect Figure M: R:

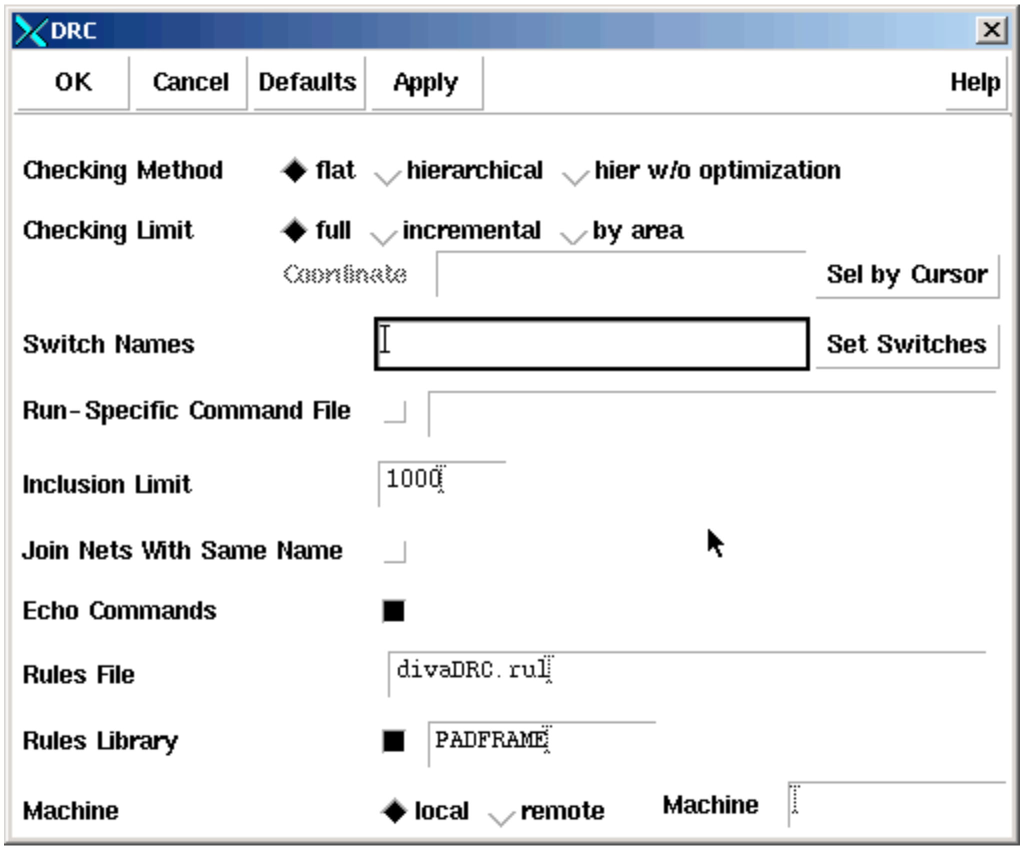
Use the options form to enter a valid terminal name:

Figure 87: Stamping down the ground pin.

Now, we've complete the layout. Go to *Design….Save* the layout. Then it is ready for DRC.

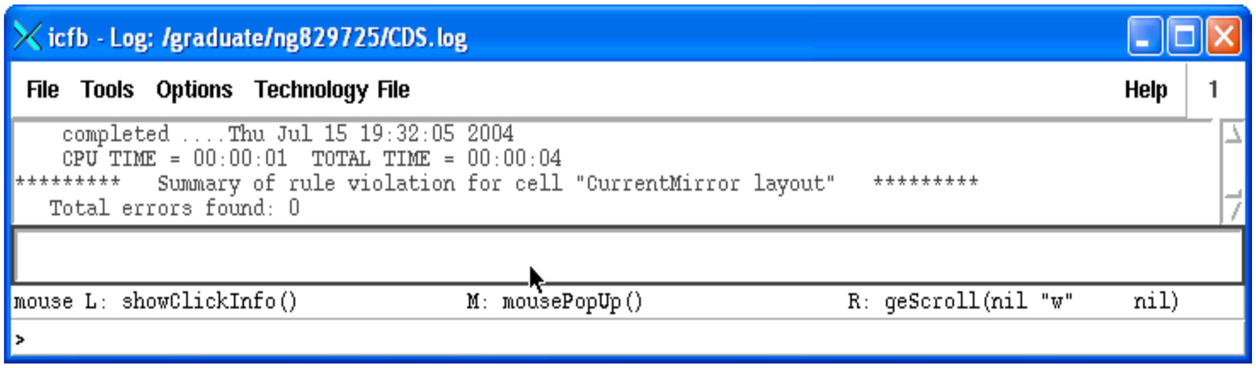
Design Rule Checking (DRC)

Go to *Verify... DRC* and a pop-up like Figure 88 should appear. You just have to click *OK* and it should run with no errors like in Figure 89.



Checking Method: [✓ flat] hierarchical hier w/o optimization
Checking Limit: [✓ full] incremental by area
Coordinate
Switch Names
Run-Specific Command File
Inclusion Limit: 1000
Join Nets With Same Name
Echo Commands: ■
Rules File: divaDRC.rul
Rules Library: ■ PADFRAME
Machine: [✓ local] remote Machine

Figure 88: Running DRC



```
    completed ....Thu Jul 15 19:32:05 2004
    CPU TIME = 00:00:01  TOTAL TIME = 00:00:04
*********   Summary of rule violation for cell "CurrentMirror layout"   *********
   Total errors found: 0
```

```
mouse L: showClickInfo()             M: mousePopUp()              R: geScroll(nil "w"      nil)
>
```

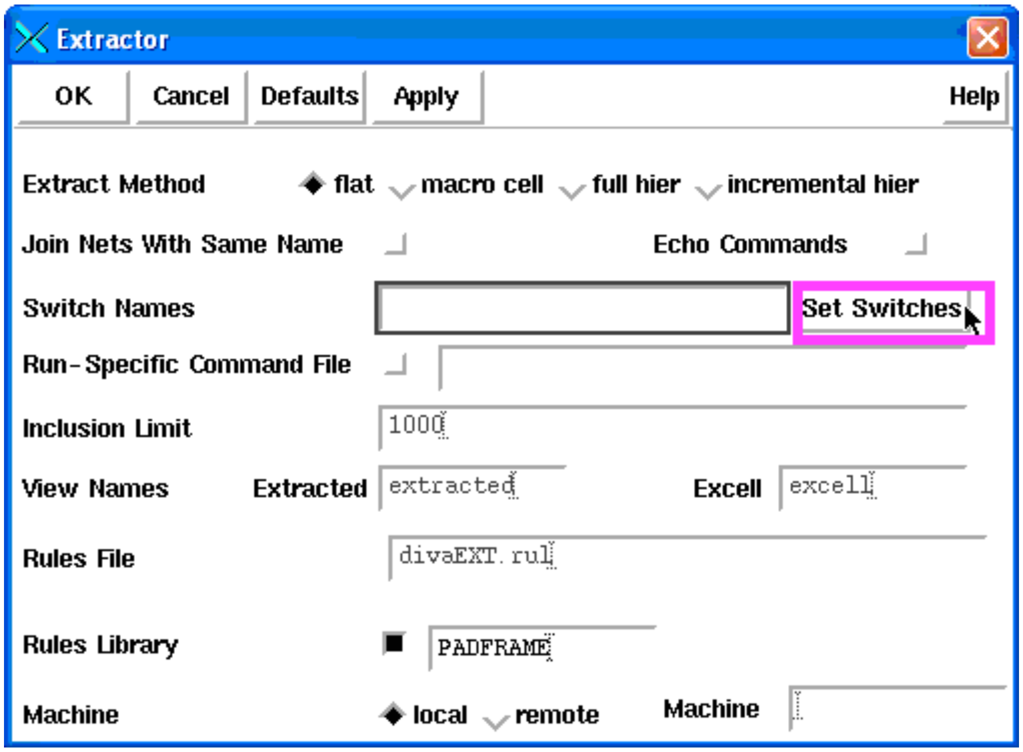
Figure 89: No DRC Errors.

If you have errors, fix them according to the AMI16 design rules with the method you learned in the cell design tutorial. For more information about how to fix errors occurred during running the CDS tools, you may check on FAQ 1 on Cadence Tools in Prof. Parent's website. Once the circuit has no DRC errors, **save** your work again.

Circuit Extraction

Now that your circuit is laid out with no DRC errors, it is time to check if it is an electrical equivalent of your current mirror schematic.

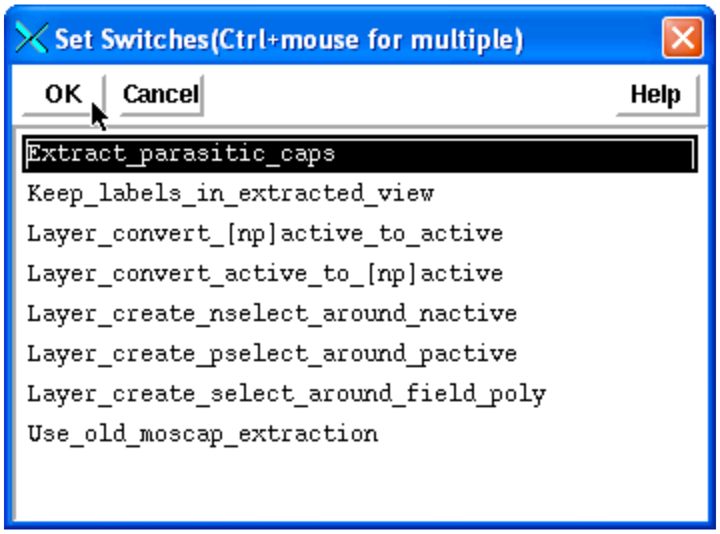
Go to *Verify... Extract* and a pop-up like Figure 90 should appear.



Extract Method: ◆ flat, macro cell, full hier, incremental hier
Join Nets With Same Name
Echo Commands
Switch Names
Run-Specific Command File
Inclusion Limit: 1000
View Names — Extracted: extracted — Excell: excell
Rules File: divaEXT.rul
Rules Library: ■ PADFRAME
Machine: ◆ local, remote — Machine

Figure 90: Running the Extractor.

Click on the set switches button in the extractor pop-up and a list of choice should appear like in Figure 91.

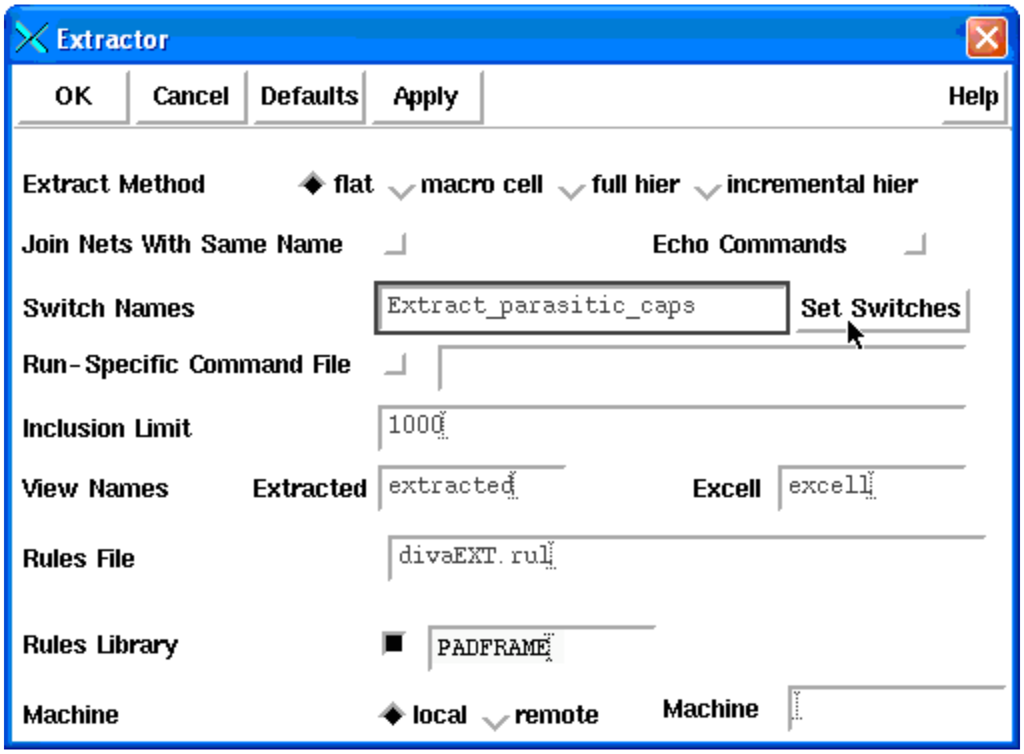


```
Extract_parasitic_caps
Keep_labels_in_extracted_view
Layer_convert_[np]active_to_active
Layer_convert_active_to_[np]active
Layer_create_nselect_around_nactive
Layer_create_pselect_around_pactive
Layer_create_select_around_field_poly
Use_old_moscap_extraction
```

Figure 91: Selecting Extract parasitic capacitances.

Click on *Extract_parasitic_caps* and press *OK*.

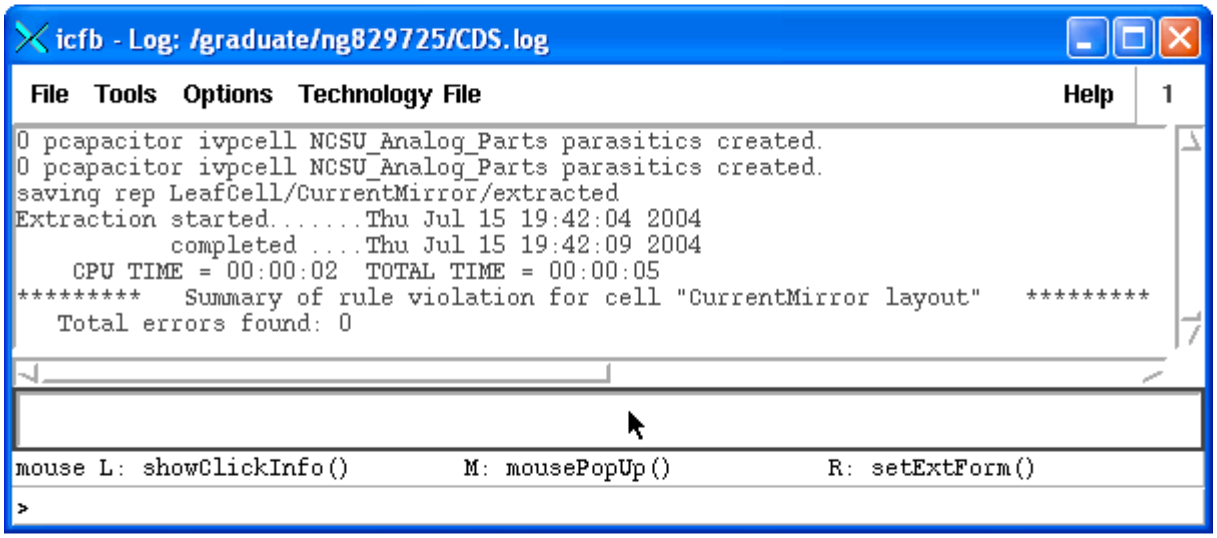
Your Extractor Form should look like Figure 92.



Extract Method [✓ flat] macro cell full hier incremental hier
Join Nets With Same Name
Echo Commands
Switch Names: Extract_parasitic_caps
Set Switches
Run-Specific Command File
Inclusion Limit: 1000
View Names Extracted: extracted
Excell: excell
Rules File: divaEXT.rul
Rules Library [✓] PADFRAME
Machine [✓ local] remote
Machine:

Figure 92: Extractor all set to go.

Click *ok* to run the extractor. The *CIW* should give no errors as in Figure 93.



```
0 pcapacitor ivpcell NCSU_Analog_Parts parasitics created.
0 pcapacitor ivpcell NCSU_Analog_Parts parasitics created.
saving rep LeafCell/CurrentMirror/extracted
Extraction started.......Thu Jul 15 19:42:04 2004
           completed ....Thu Jul 15 19:42:09 2004
    CPU TIME = 00:00:02  TOTAL TIME = 00:00:05
*********   Summary of rule violation for cell "CurrentMirror layout"   *********
   Total errors found: 0
```

```
mouse L: showClickInfo()          M: mousePopUp()          R: setExtForm()
>
```

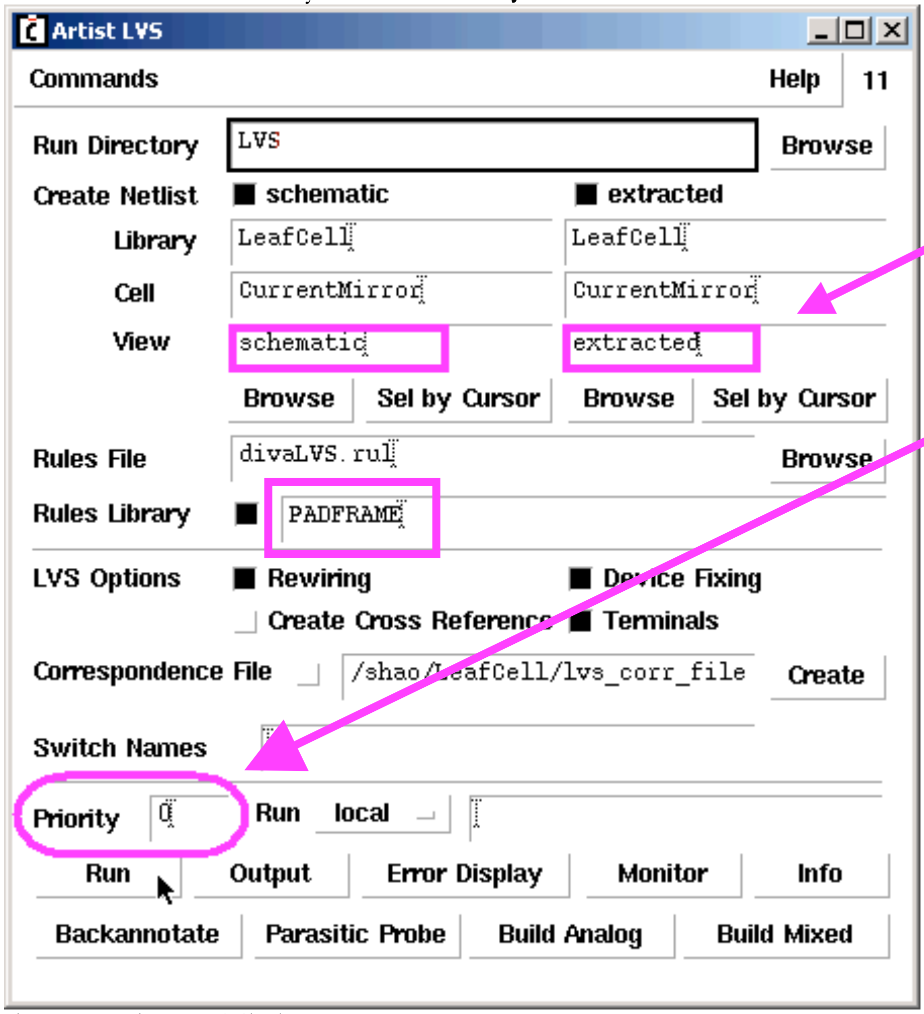
Figure 93: Extracted current mirror with no errors.

Layout versus Schematic (LVS)

An LVS check makes sure that the circuit you laid out is equivalent to the one you entered into your schematic.

To run an LVS check, go to *Verify... LVS* and a pop-up should appear like Figure 96.

Use the *Browse* button to select which schematic and which extracted file you are going to check for equivalence. The pop-up should be filled out just like Figure 94. Make sure to have PADFRAME as your Rules Library. Click *Run* to start.



Run Directory: LVS
Create Netlist: ■ schematic ■ extracted
Library: LeafCell | LeafCell
Cell: CurrentMirror | CurrentMirror
View: schematic | extracted
Rules File: divaLVS.rul
Rules Library: ■ PADFRAME
LVS Options: ■ Rewiring ■ Device Fixing ☐ Create Cross Reference ■ Terminals
Correspondence File: ☐ /shao/LeafCell/lvs_corr_file
Switch Names:
Priority: 0 Run: local

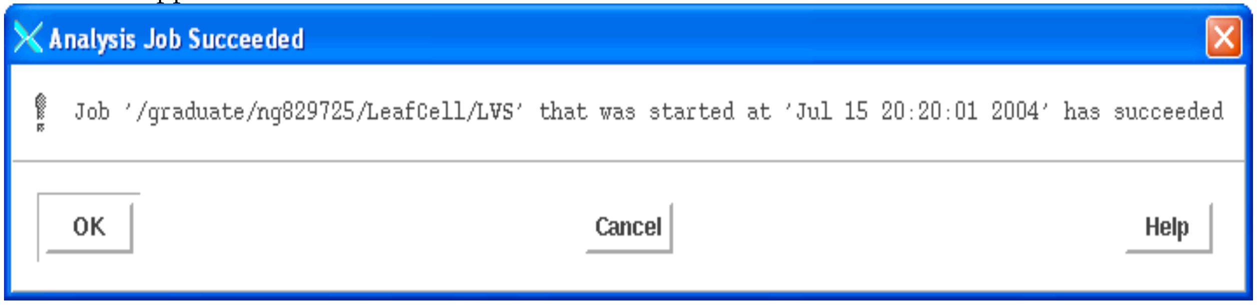
Make sure you have the right views of the current mirror.

If you have a large circuit to check, set the *Priority* to a higher number, so it will not suck up system resources too much. You can leave it *0* (highest priority) for now.

NOTE: If you think that having your job running at the highest priority will get your job finished quickly, think again. You could crash the system if too many users use too high a priority on an LVS job!

Figure 94: Running an LVS Check

This will take several minutes. When the LVS check is succeeded, a pop-up like Figure 95 should appear.



Job '/graduate/ng829725/LeafCell/LVS' that was started at 'Jul 15 20:20:01 2004' has succeeded

Figure 95: Successful completion of an LVS check.

Analysis Job Succeeded **doesn't** mean that the LVS check is passed. To make sure the LVS is passed or not, go back to the *LVS* Form and click on *Output* (Figure 96). The output file will pop up (Figure 97), listing the result of the comparison between the schematic and the extracted layout views.

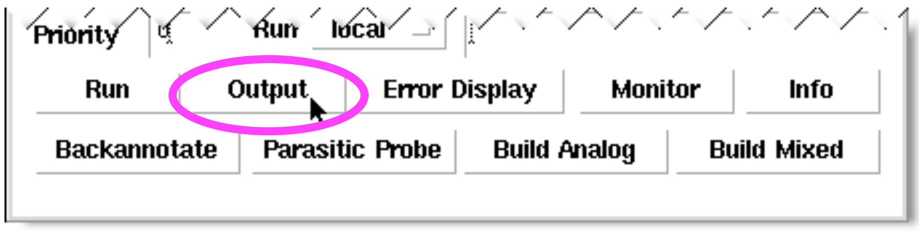
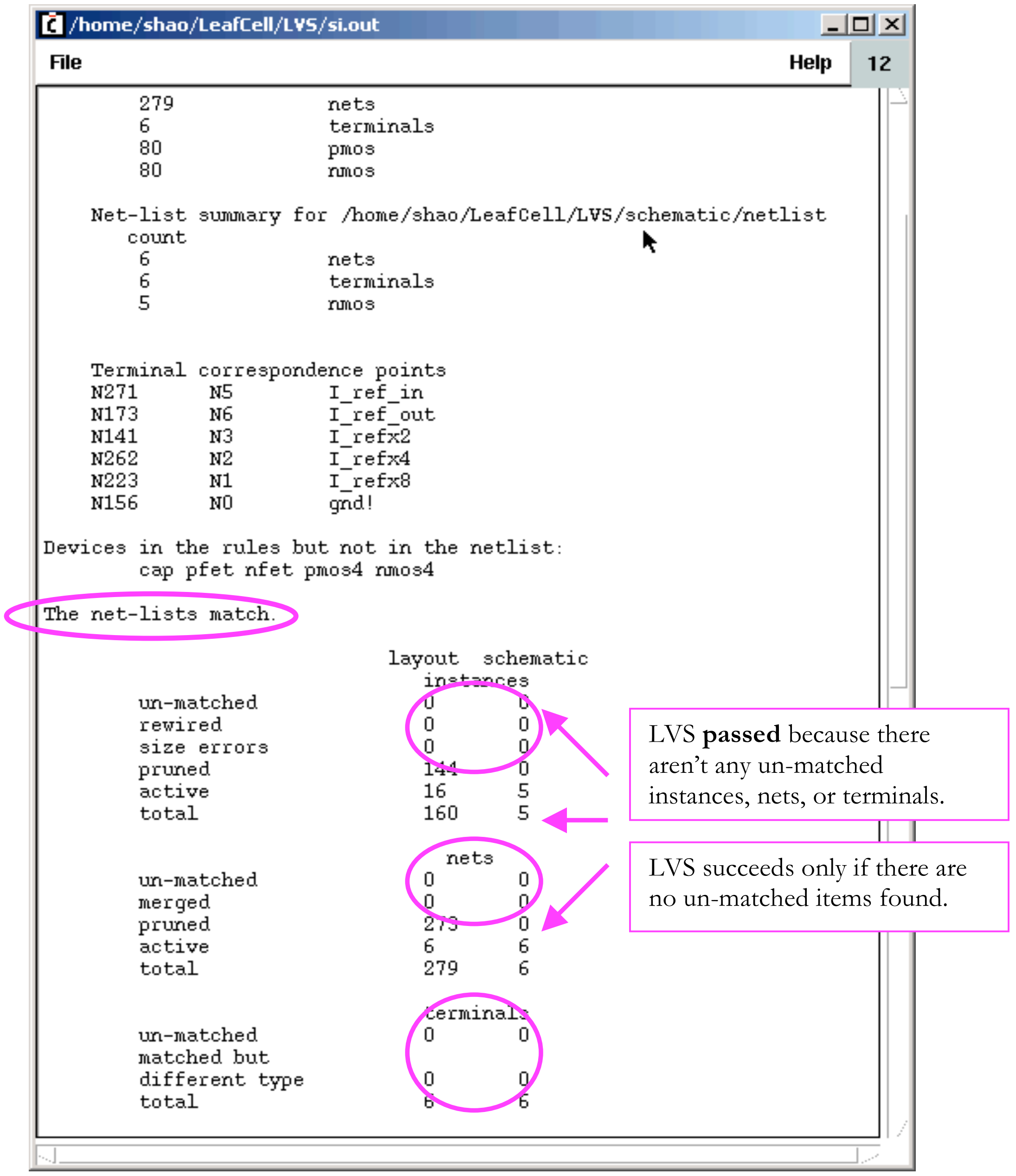


Figure 96: Output the LVS information..



```
        279              nets
        6                terminals
        80               pmos
        80               nmos

    Net-list summary for /home/shao/LeafCell/LVS/schematic/netlist
       count
        6                nets
        6                terminals
        5                nmos


    Terminal correspondence points
    N271      N5         I_ref_in
    N173      N6         I_ref_out
    N141      N3         I_refx2
    N262      N2         I_refx4
    N223      N1         I_refx8
    N156      N0         gnd!

Devices in the rules but not in the netlist:
        cap pfet nfet pmos4 nmos4

The net-lists match.

                              layout  schematic
                                 instances
        un-matched               0       0
        rewired                  0       0
        size errors              0       0
        pruned                   144     0
        active                   16      5
        total                    160     5

                                   nets
        un-matched               0       0
        merged                   0       0
        pruned                   273     0
        active                   6       6
        total                    279     6

                                 terminals
        un-matched               0       0
        matched but
        different type           0       0
        total                    6       6
```

LVS **passed** because there aren't any un-matched instances, nets, or terminals.

LVS succeeds only if there are no un-matched items found.

Figure 97: Output of the LVS.

What should be done if the LVS doesn't pass?

The first thing we need to do is to go back to the schematic view and the layout view to see if the wiring is wrong. **Make sure the grounded nmos is not right next to a signal path with potential higher than the gnd!** (Figure 98)

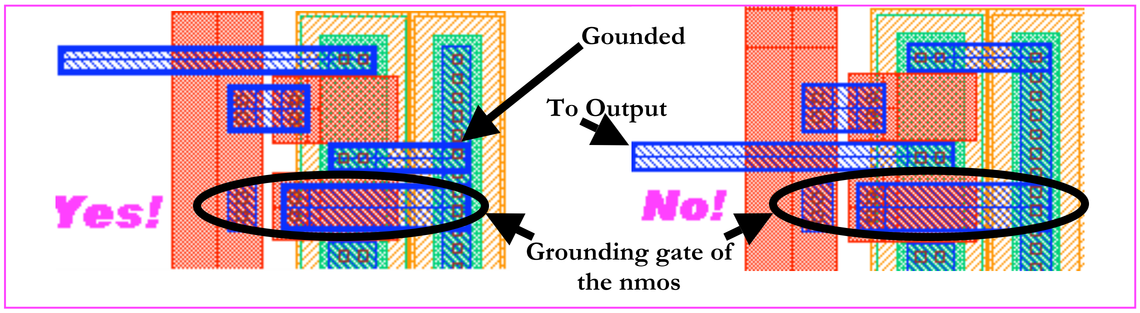


Figure 98: The right way to short NMOS to ground.

Alternative Note: On the other hand, this means if we are doing a project using pmos shorted to vdd!, the shorted pmos should not be place right next to a singal path with potential lower than vdd! (Figure 99)

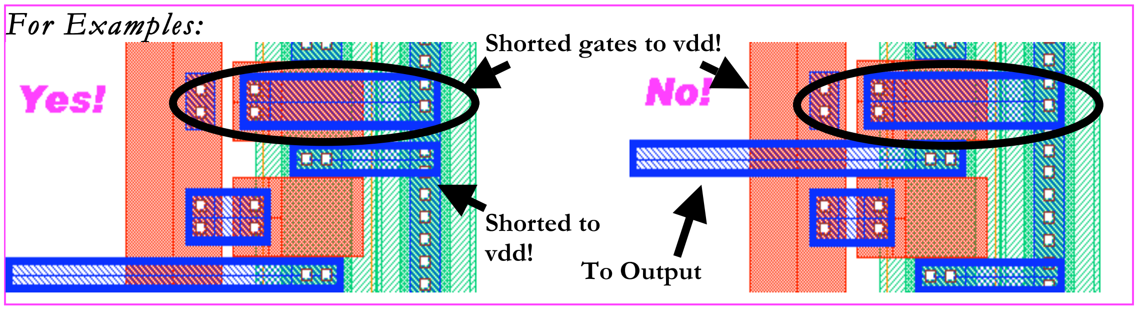


Figure 99: The right way to short PMOS to vdd.

If nothing wrong is found, we can go back to the LVS form. Click at *Info*, and then the *Display Run Information* window is shown (Figure 100). From this information window, we can click on whatever is available to see if we can get any clue of what was going wrong.

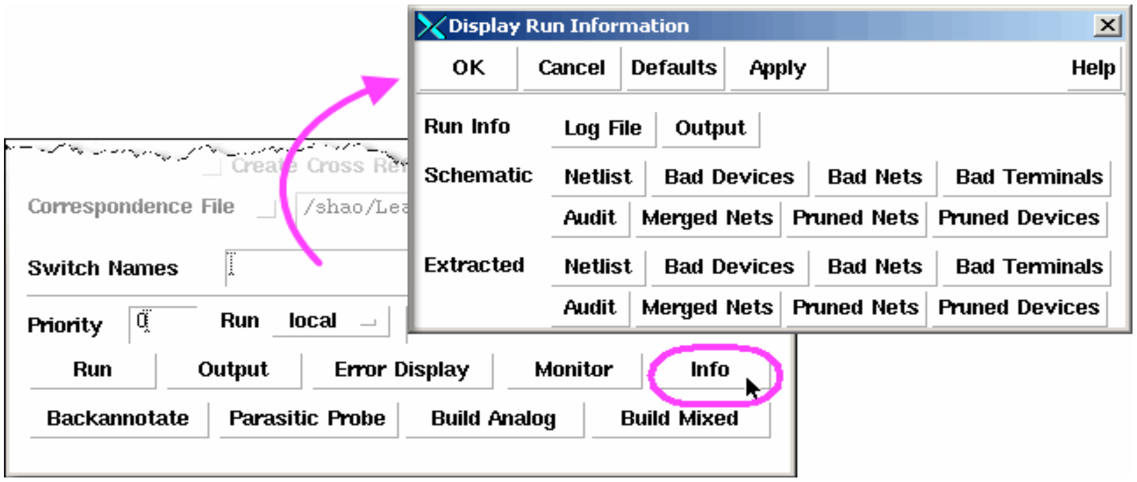


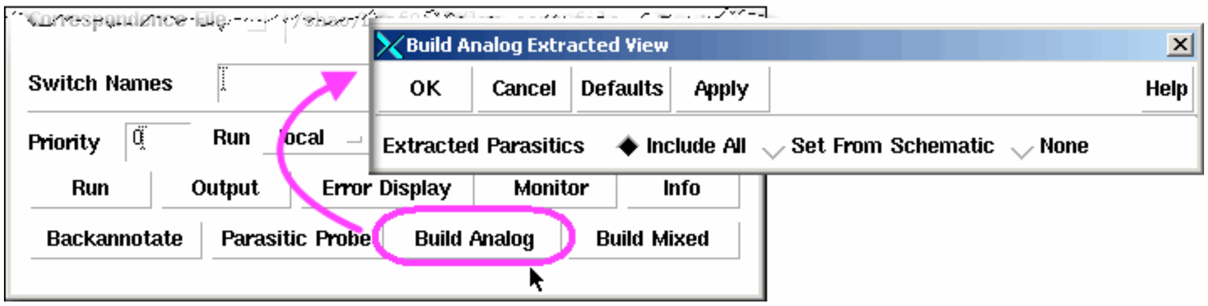
Figure 100: Display Run Information for debugging.

If you try all these and still can't find what was going wrong, then you should have someone to check it for you, or go to the TAs for help.

Build Analog Extracted View:

In order to prepare for the analog post extraction simulation, the Analog_Extracted view should be created.

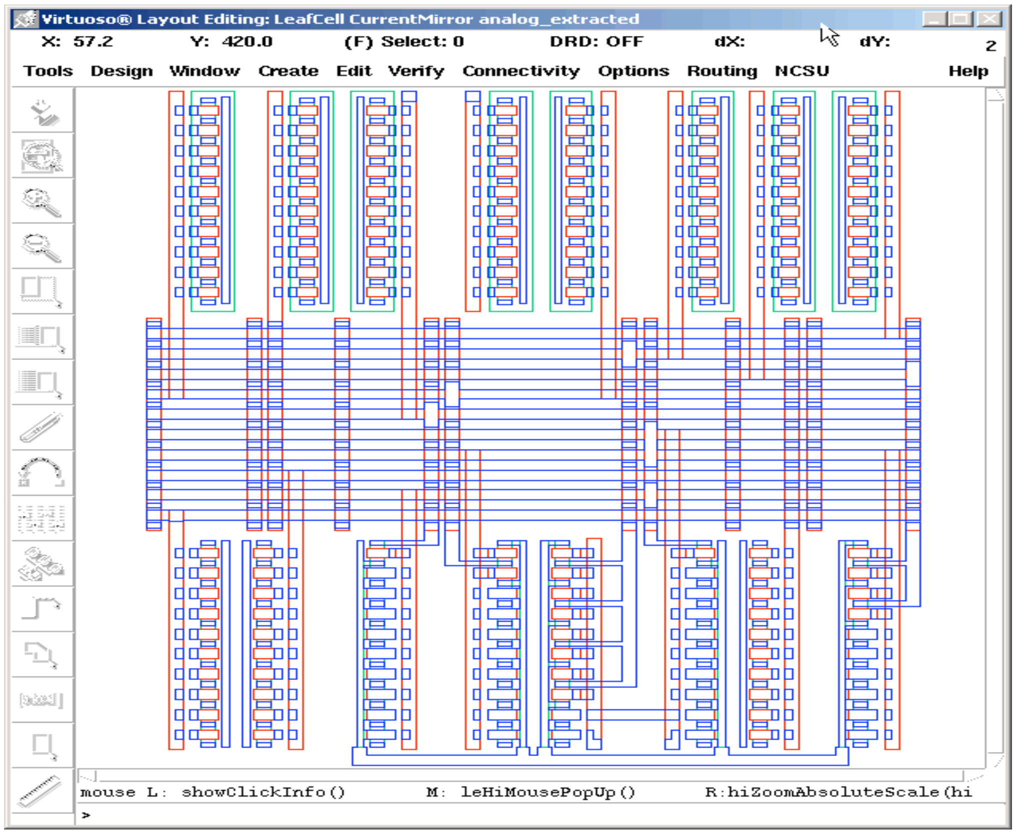
Go back to the *Artist LVS* form, find and click at the **Build Analog** botton, and then click **OK** in the popup window. The Analog_extracted view will be created by the system (Figure 101).



Extracted Parasitics ◆ Include All Set From Schematic None

Figure 101: Build Analog Extracted..

To see how your Analog_Extracted current mirror looks like, go back to the library manager and select to open the **analog_extracted** view. Your extracted view should look like Figure 102.



X: 57.2 Y: 420.0 (F) Select: 0 DRD: OFF dX: dY: 2

mouse L: showClickInfo() M: leHiMousePopUp() R:hiZoomAbsoluteScale(hi

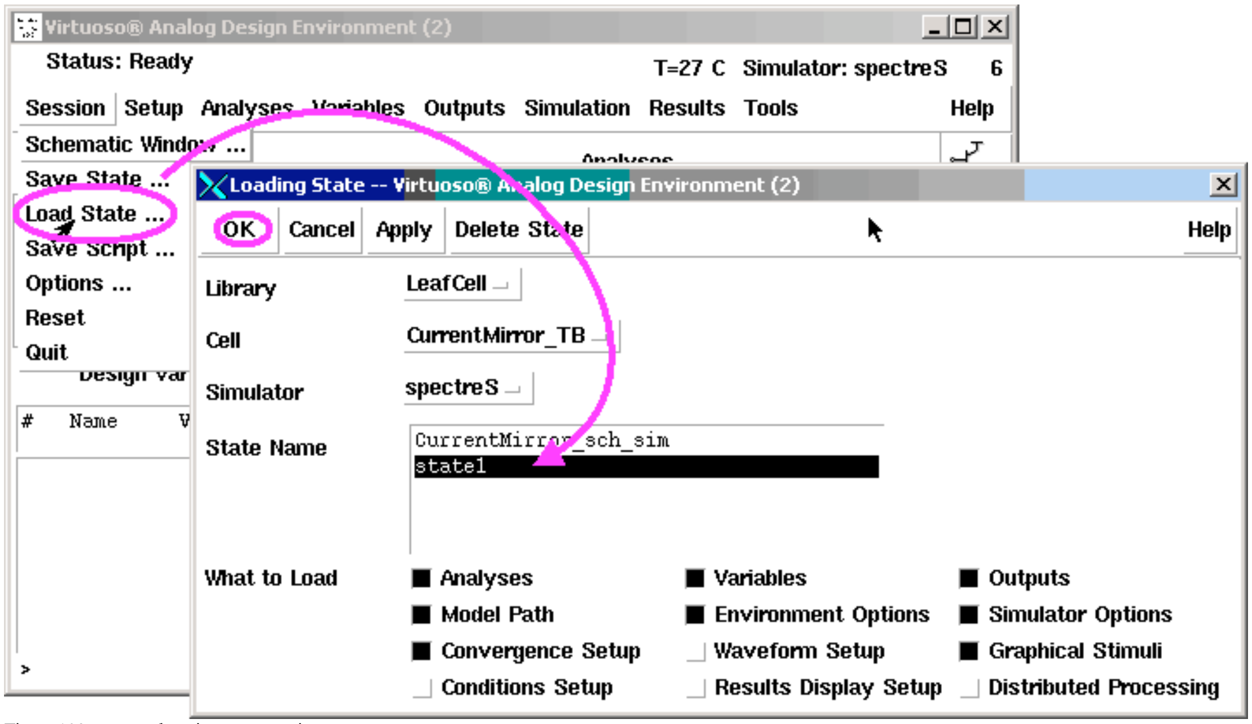
Figure 102: Extracted view of the current mirror.

Post Extraction Simulation

In post extraction simulation, you verify that your circuit still meets your specification with the additions of parasitic capacitances. For example, in a schematic, where two wires that are not connected, there is no transfer of AC voltage or current; while in a layout, where two metal lines are close but not connected, there is a capacitance between the two that that sometimes will cause cross talk. Post extraction simulation will show these errors. On the other hand, post extraction is another way to check out layout errors which sneaked through the verification processes.

To perform a post extraction simulation:

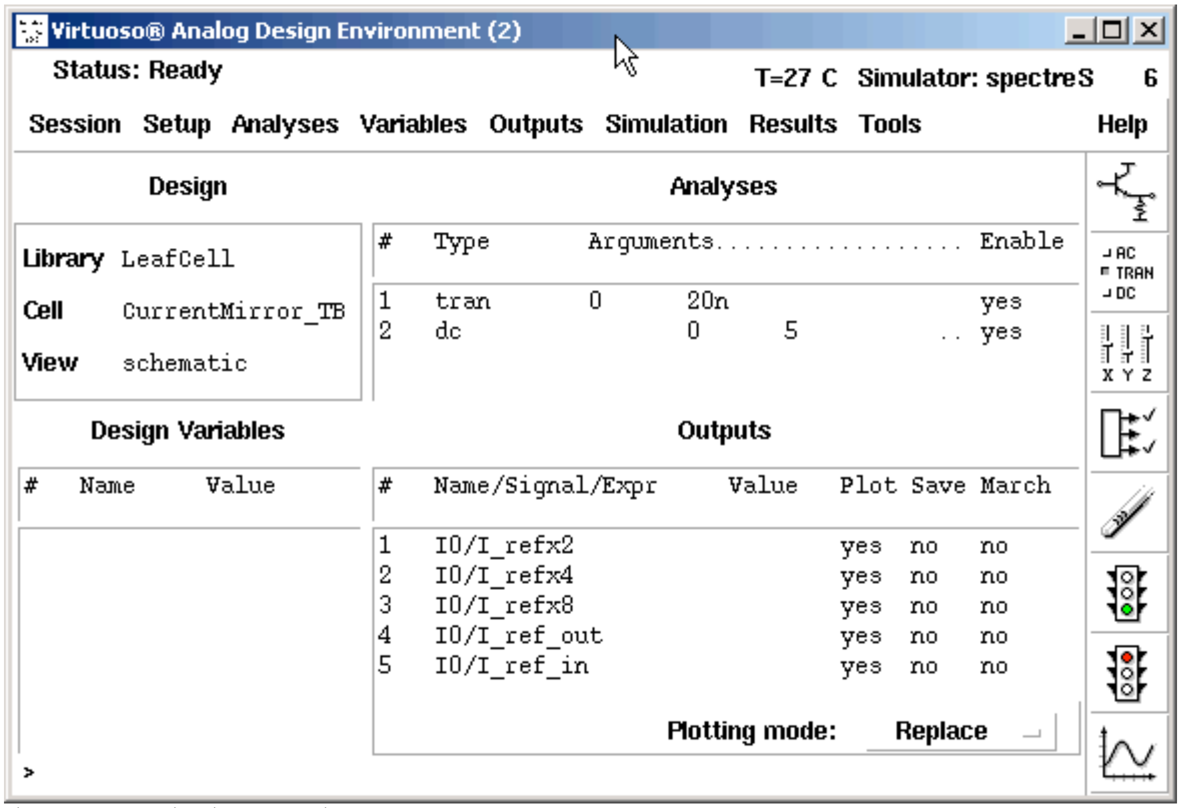
1. Open up the *currentmirror_TB* schematic window,
2. Click on *Tools…Analog Environment…*to start the *Affirma Analog environment*.


Virtuoso® Analog Design Environment (2)
Status: Ready
T=27 C Simulator: spectreS 6
Session Setup Analyses Variables Outputs Simulation Results Tools Help
Schematic Window ...
Save State ...
Load State ...
Save Script ...
Options ...
Reset
Quit
Loading State -- Virtuoso® Analog Design Environment (2)
OK Cancel Apply Delete State Help

Library LeafCell
Cell CurrentMirror_TB
Simulator spectreS
State Name CurrentMirror_sch_sim / state1
What to Load ■ Analyses ■ Variables ■ Outputs ■ Model Path ■ Environment Options ■ Simulator Options ■ Convergence Setup □ Waveform Setup ■ Graphical Stimuli □ Conditions Setup □ Results Display Setup □ Distributed Processing

Figure 103: extracted environment setting.

3. Go to Session…Load State…, select the state1 which you save earlier, and then click OK (Figure 103). The Affirma will look as in Figure 104.


Virtuoso® Analog Design Environment (2)
Status: Ready
T=27 C Simulator: spectreS 6
Session Setup Analyses Variables Outputs Simulation Results Tools Help
Design
Analyses
Design Variables
Outputs
AC
TRAN
DC
X Y Z
Plotting mode: Replace

Library LeafCell
Cell CurrentMirror_TB
View schematic

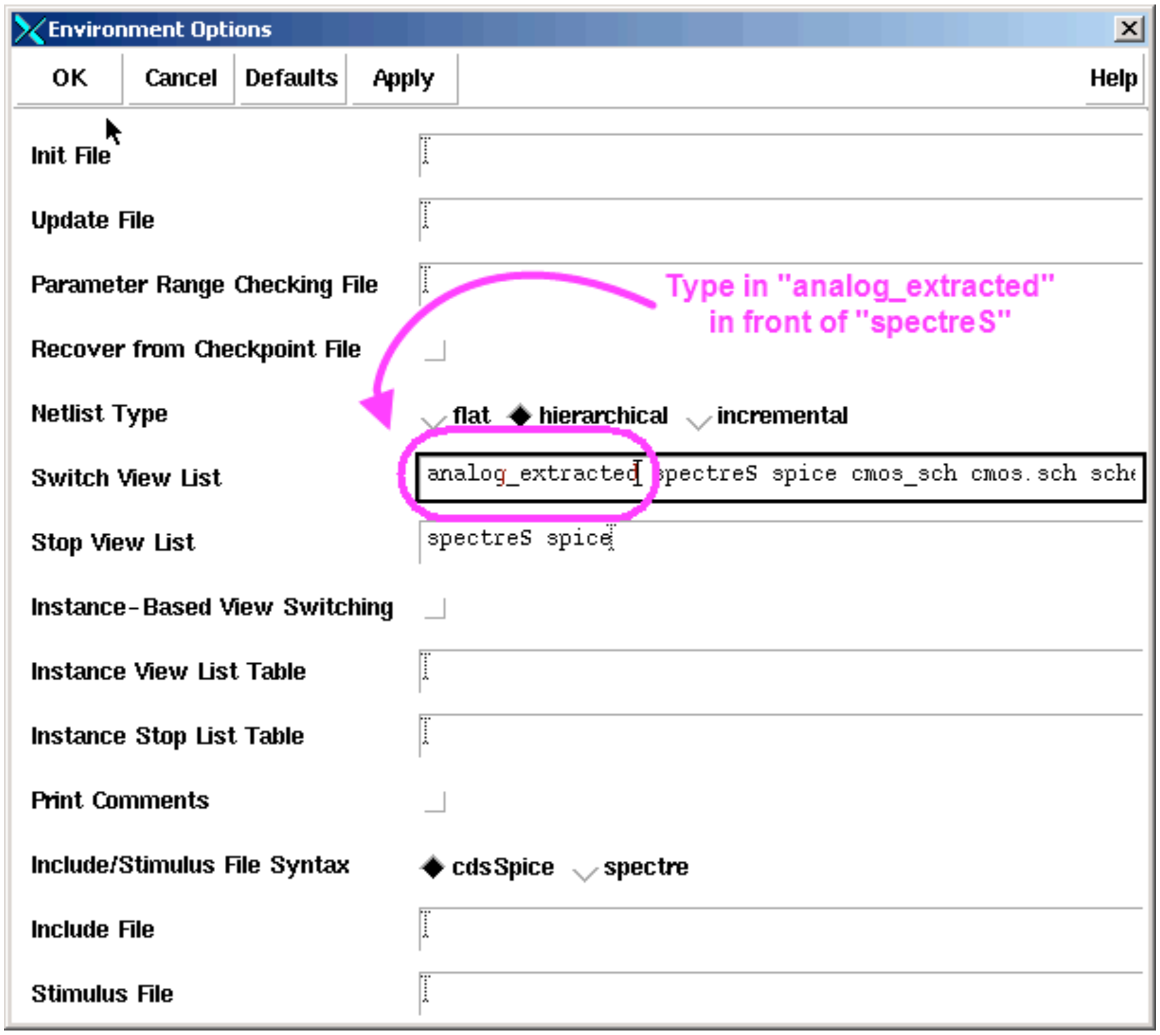
```
#   Type      Arguments.................. Enable
1   tran      0       20n                 yes
2   dc                0       5        .. yes
```

```
#   Name      Value
```

```
#   Name/Signal/Expr     Value   Plot Save March
1   I0/I_refx2                   yes  no   no
2   I0/I_refx4                   yes  no   no
3   I0/I_refx8                   yes  no   no
4   I0/I_ref_out                 yes  no   no
5   I0/I_ref_in                  yes  no   no
```

Figure 104: extracted environment setting.

4. Go to *Setup... Environment*. A pop-up like Figure 105 should appear.
5. Type in the word **"analog_extracted"** as shown in Figure 105 in front of "**spectraS**"
6. Click OK, the finished one should look like Figure 104 again.



Init File

Update File

Parameter Range Checking File

Recover from Checkpoint File

Netlist Type: flat [✓ hierarchical] incremental

Switch View List: analog_extracted spectreS spice cmos_sch cmos.sch sch

Stop View List: spectreS spice

Instance-Based View Switching

Instance View List Table

Instance Stop List Table

Print Comments

Include/Stimulus File Syntax: [✓ cdsSpice] spectre

Include File

Stimulus File

Figure 105: Adding the extracted environment.

7. Go to Simulation…Run… to start the simulation. This will take a while. A *Graphics* window will pop up showing the simulation results (Figure 106).
8. Go to *Markers…Vertical Markers* to put them at 2 and 5V at the signal I_ref_out. You can find the readings at the bottom right of the *Waveform Window* as well as its slope.

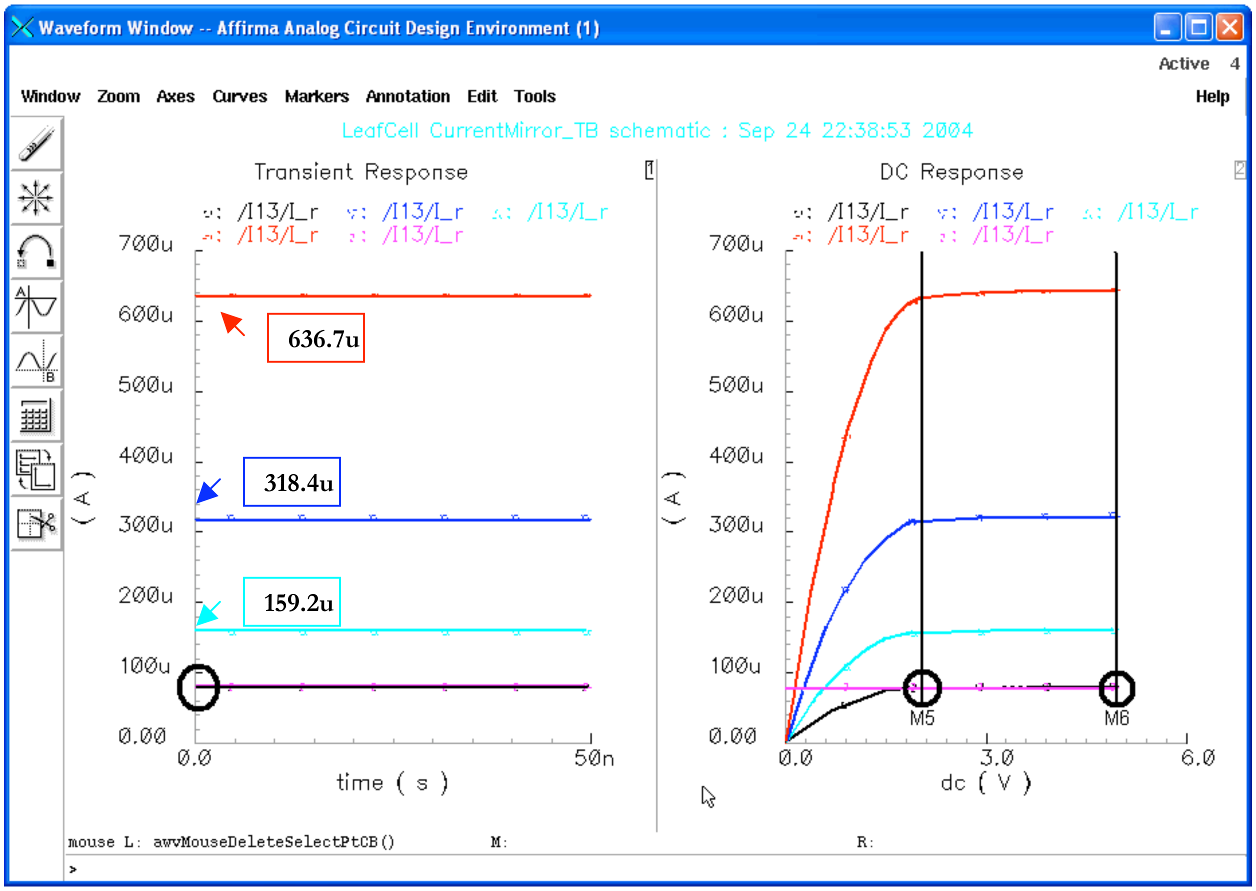
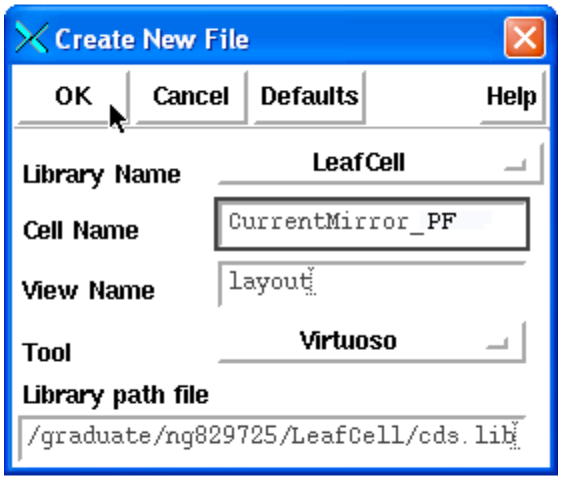

Waveform Window -- Affirma Analog Circuit Design Environment (1)
Active 4
Window Zoom Axes Curves Markers Annotation Edit Tools Help
LeafCell CurrentMirror_TB schematic : Sep 24 22:38:53 2004
Transient Response
DC Response
/I13/I_r
636.7u
318.4u
159.2u
time (s)
dc (V)
(A)
M5
M6
mouse L: awvMouseDeleteSelectPtCB() M: R:

Figure 106: Post Extracted Simulation results.

Put in the Pad frame and Extraction Simulation:

After it is tested through simulation for both schematic and layout (extracted layout), it is time to put it into a pad frame. What you need to do is as follow:

1. Create and CurrentMirror_PF Layout view (Figure 107).



Create New File

OK Cancel Defaults Help

Library Name: LeafCell

Cell Name: CurrentMirror_PF

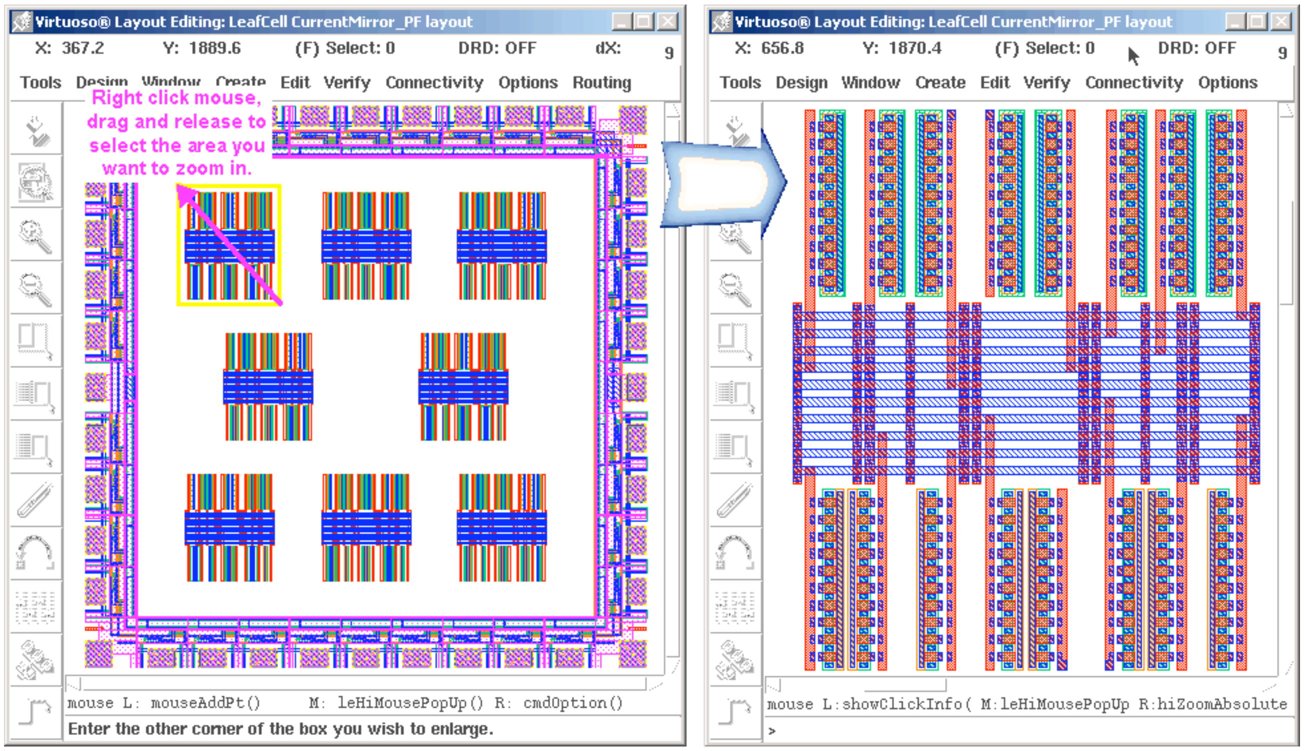
View Name: layout

Tool: Virtuoso

Library path file: /graduate/ng829725/LeafCell/cds.lib

Figure 107: Getting the metal only.

2. Go to *Create…Instance…* or just click at the *"i"* from the keyboard to add an instance called ALC_PF from library PADFRAME (Figure 108). Leave the window open.



Virtuoso® Layout Editing: LeafCell CurrentMirror_PF layout

X: 367.2 Y: 1889.6 (F) Select: 0 DRD: OFF dX: 9

Tools Design Window Create Edit Verify Connectivity Options Routing

Right click mouse, drag and release to select the area you want to zoom in.

mouse L: mouseAddPt() M: leHiMousePopUp() R: cmdOption()

Enter the other corner of the box you wish to enlarge.

Virtuoso® Layout Editing: LeafCell CurrentMirror_PF layout

X: 656.8 Y: 1870.4 (F) Select: 0 DRD: OFF 9

Tools Design Window Create Edit Verify Connectivity Options

mouse L:showClickInfo(M:leHiMousePopUp R:hiZoomAbsolute

>

Figure 108: Add instance ALC_PF, and then zoom in to one of the leaf cell.

3. Leave the window open.

4. For security reason, we make a secondary copy of the layout for the CurrentMirror (Figure 109).

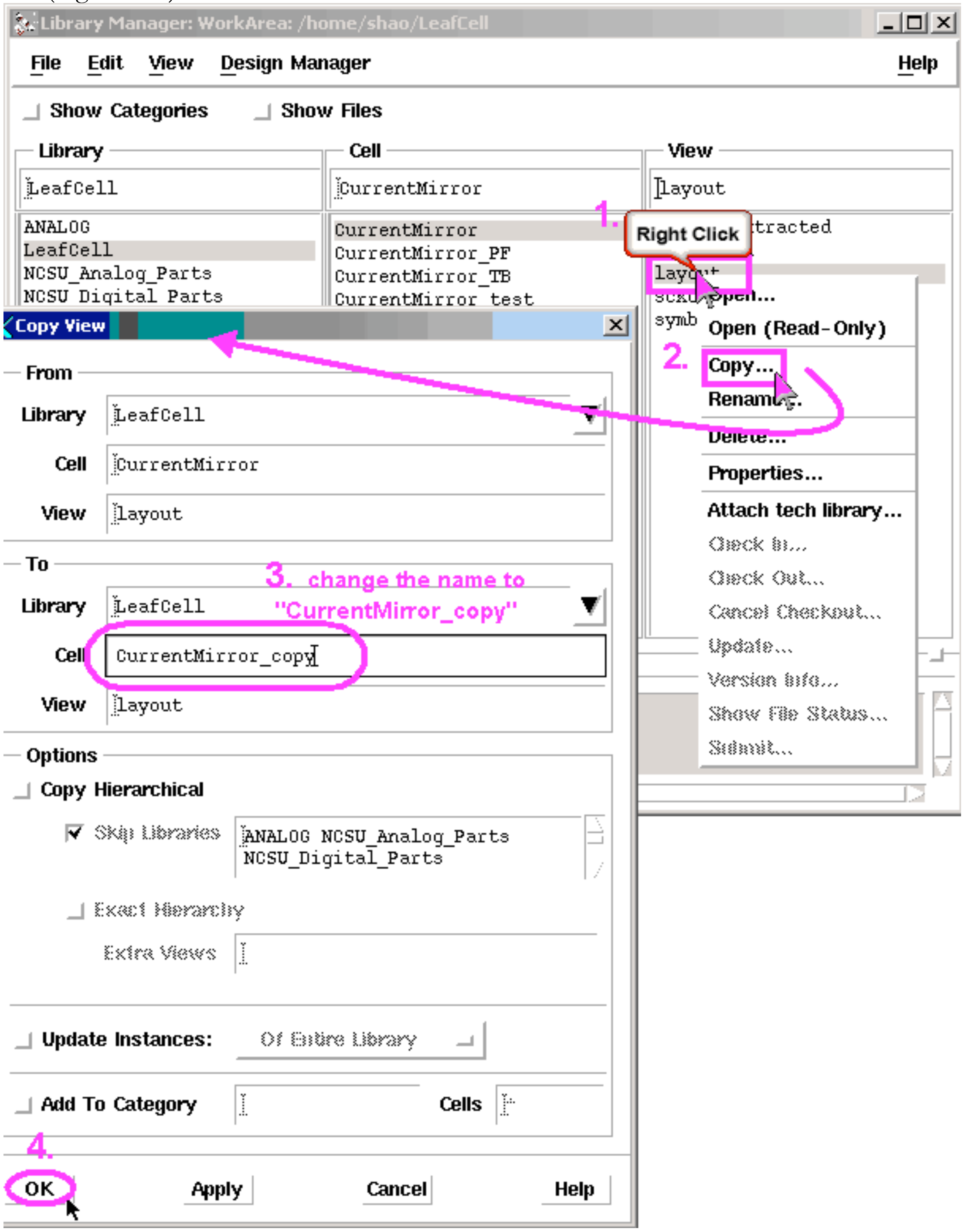


Figure 109: Make a secondary copy of layout view for CurrentMirror.

5. Open CurrentMirror_copy Layout from the Library Manager.

6. Carefully select the ALC, **not the wiring metals**, and then delete the ALC (Figure 110).

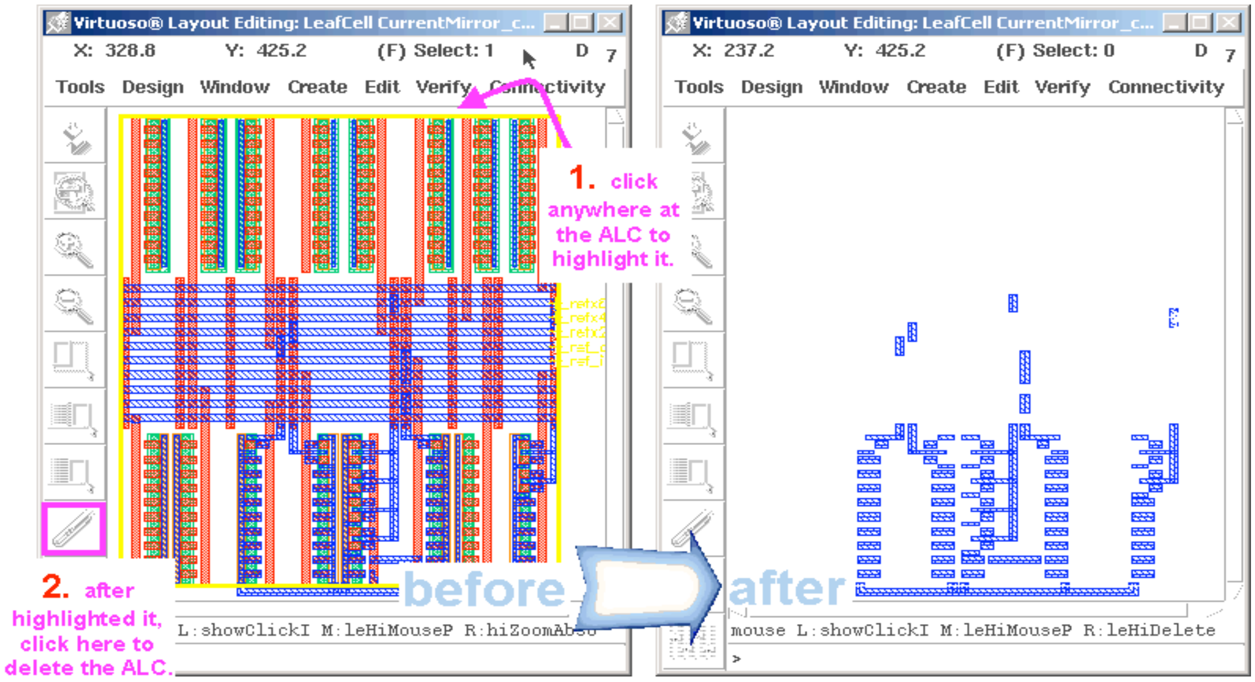

Virtuoso® Layout Editing: LeafCell CurrentMirror_c...
X: 328.8 Y: 425.2 (F) Select: 1 D 7
Tools Design Window Create Edit Verify Connectivity
1. click anywhere at the ALC to highlight it.
2. after highlighted it, click here to delete the ALC.
before
L:showClickI M:leHiMouseP R:hiZoomAbso
X: 237.2 Y: 425.2 (F) Select: 0 D 7
after
mouse L:showClickI M:leHiMouseP R:leHiDelete

Figure 110: Steps for selecting the metals from the leafcell layout.

7. Select all the leftovers. While they are all hightlighted, click the copy button. It will ask you to select a reference point (Figure 111).

   Decide your reference, click at your reference point. Move your mouse to the CurrentMirror_PF Layout window, and then point your mouse and click at the same position as your chosen reference (Figure 111).

   Note: You can always right click and drag anytime to select areas you want to zoom in.

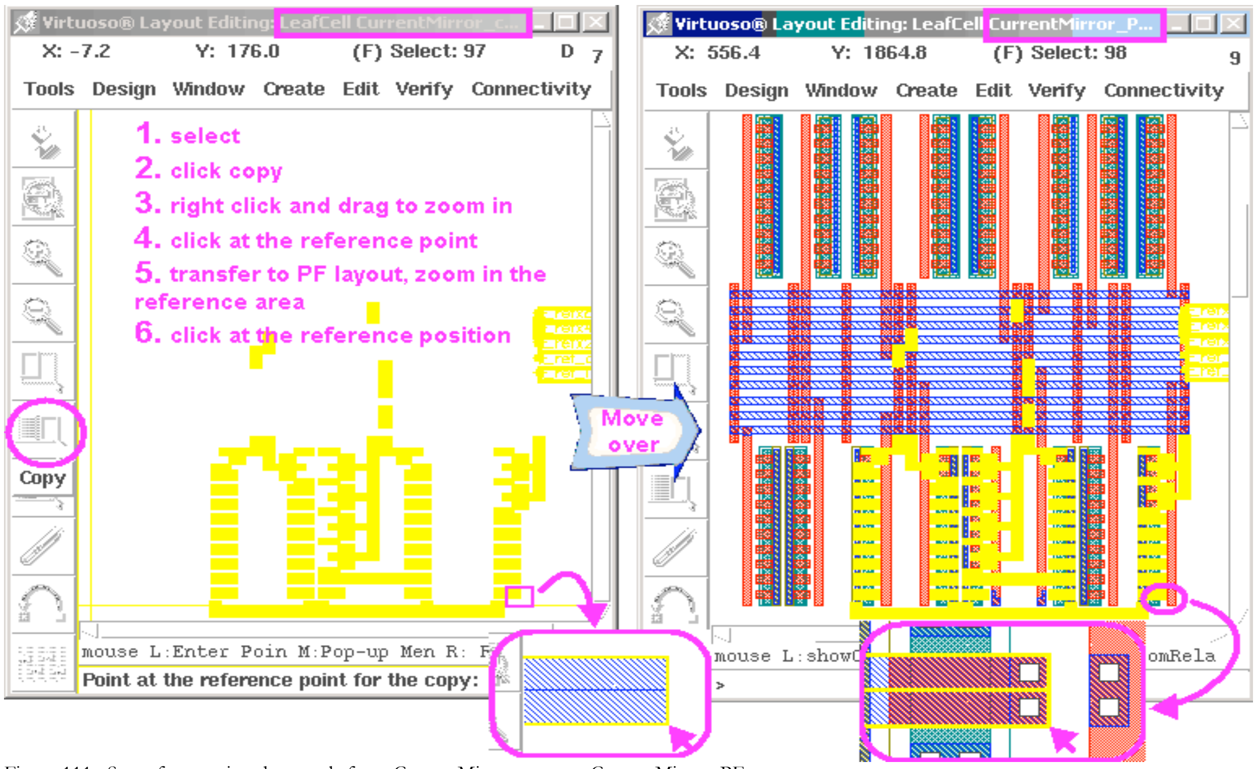
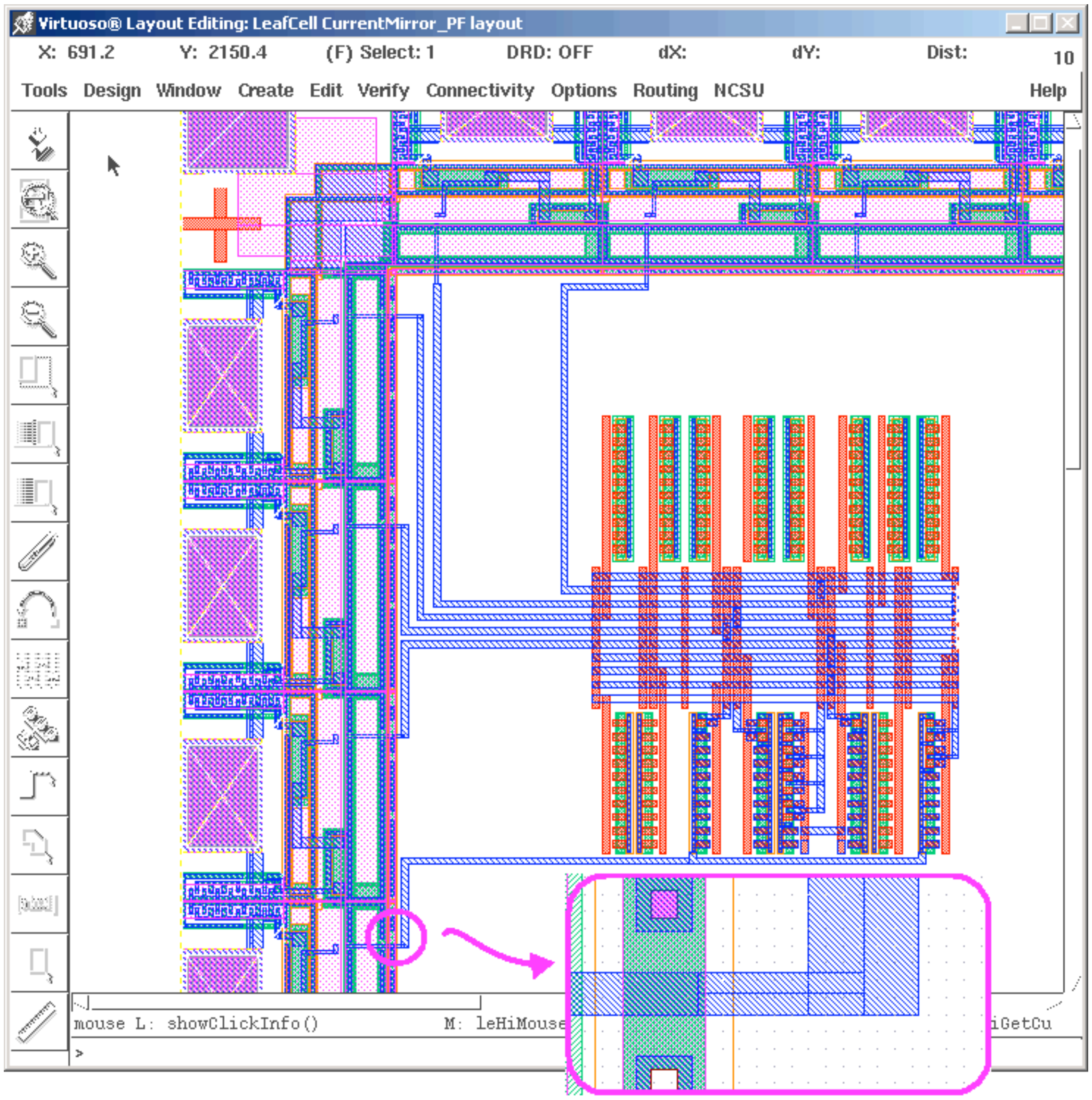

Virtuoso® Layout Editing: LeafCell CurrentMirror_c...
X: -7.2 Y: 176.0 (F) Select: 97 D 7
Tools Design Window Create Edit Verify Connectivity
1. select
2. click copy
3. right click and drag to zoom in
4. click at the reference point
5. transfer to PF layout, zoom in the reference area
6. click at the reference position
Copy
Move over
mouse L:Enter Poin M:Pop-up Men R:
Point at the reference point for the copy:
Virtuoso® Layout Editing: LeafCell CurrentMirror_P...
X: 556.4 Y: 1864.8 (F) Select: 98 9
mouse L:show
omRela

Figure 111: Steps for copying the metals from CurrentMirror_copy to CurrentMirror_PF.

8. *ESC* from the copy mode and close the CurrentMirror_copy Layout window. When it ask you if you want to save the cell view, click *NO*.

9. In the CurrentMirror_PF Layout window, create a metal path with width **6.4** (we use 6.4 because it is as wide as the interconnection of the cells, but you can always choose the width for your design) to connect the *I/O pins* and *gnd!* to the pad. (Figure 112)

   Note: You need to adjust the path width to 2.4 when approaching the pads.

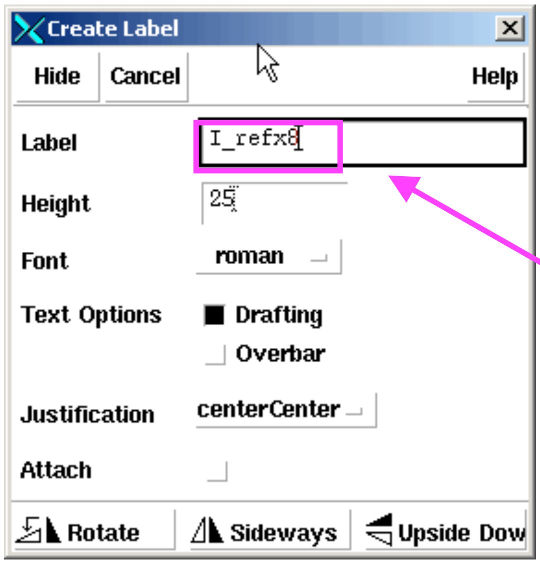


```
mouse L: showClickInfo()          M: leHiMouse    iGetCu
>
```

Figure 112: Connecting the I/O pins and gnd! to the pads.

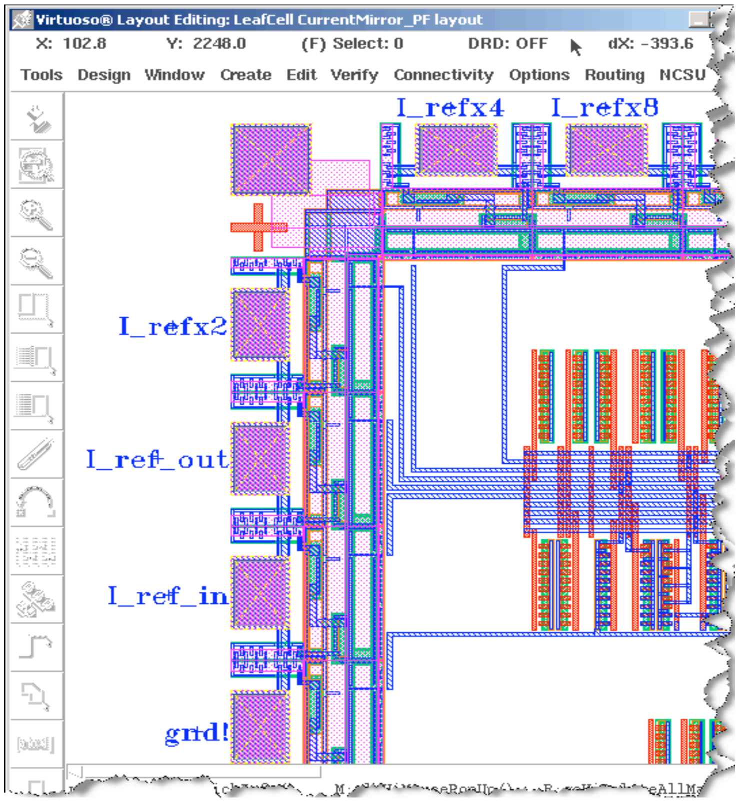
The next step (Step 10) is optional; however, I recommend it because it helps you a lot while you are testing the chip. While testing, you may know which pad is for which pin directly under the telescope without going back to your documentation.

10. Select Metal1 from the LSW window.
11. Go to *Create…Label…*, and then fill in the pin names and stamp them down right next to the pad numbers one by one. The completely labeled PF is as in Figure 113.



Label: I_refx8
Height: 25
Font: roman
Text Options: ■ Drafting
Overbar
Justification: centerCenter
Attach

1. Type in "I_refx8",
Stamp it down above 2nd pad to the right from upper left corner;
2. Type in "I_refx4",
Stamp it down above 1st pad to the right from upper left corner;
3. Type in "I_refx2",
Stamp it down above 1st pad down from upper left corner;
4. Type in "I_ref_out",
Stamp it down above 2nd pad down from upper left corner;
5. Type in "I_ref_in",
Stamp it down above 3rd pad down from upper left corner;
6. Type in "gnd!"
Stamp it down above 4th pad down from upper left corner;



Use the proper "Justification" to make sure that the plus sign is in the pad.

Figure 113: Labeling the pads.

12. Save the layout.

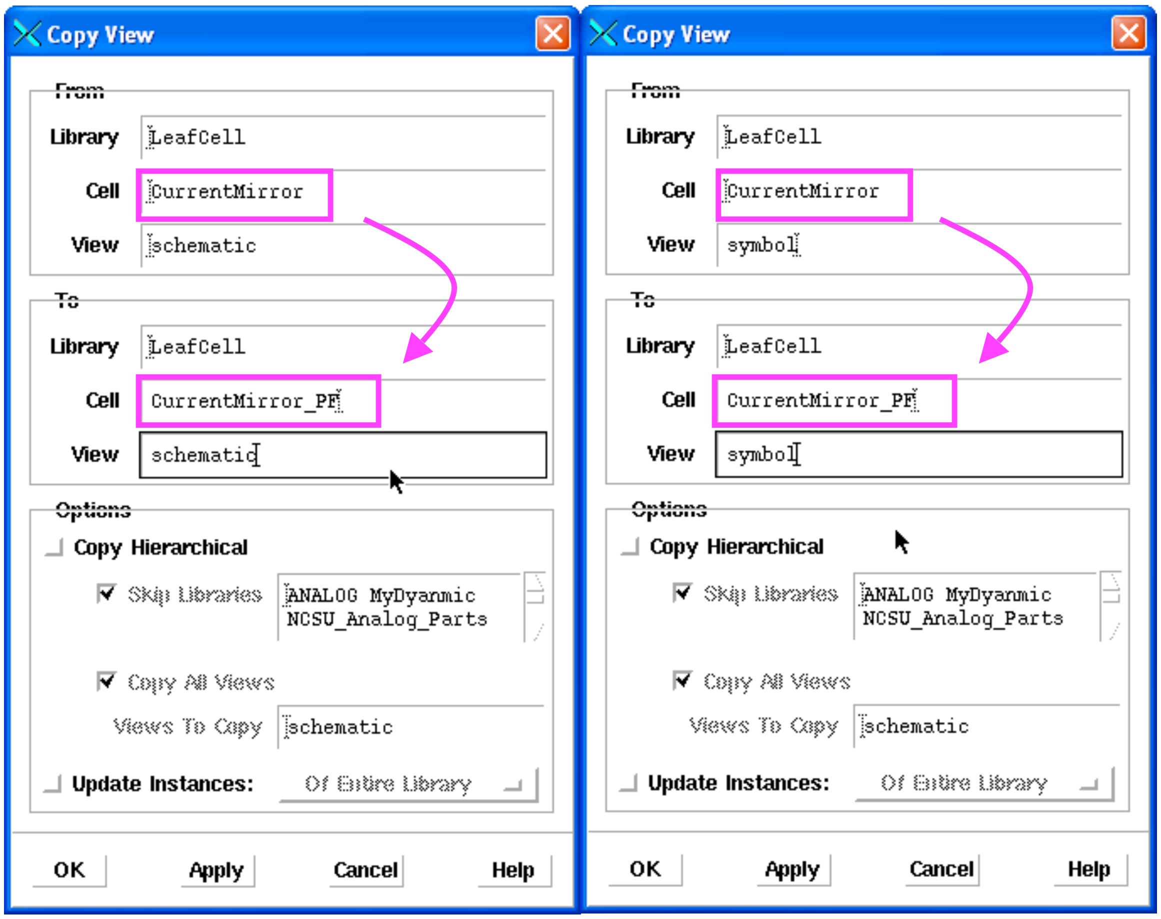
13. *Run* the *DRC*. There should be **217** errors afater DRC check, no more and no less. If there are more than 217, then you need to use your debugging skills to fix it.

Now the PF is done. The next steps are to extract it, run LVS, and then build analog, which are the same progresses as you do after you are done with a layout.

Before running the LVS, we need to create the schematic and symbol view for the PF.

14. Copy the symbol view and schematic view from cell ***CurrentMirror*** to cell ***CurrentMirror_PF***.

    To do so, go to the *Library Manager*, **make sure you select the view you are copying first;** then go to the menu, select *Edit…Copy…*, and then fill out the form as in Figure 114. **If the *Copy Problem Form* pops up, choose *Fix Errors*, then *OK*.** The copied views will show up in the Library Manager window as in Figure 115.



Copy View

From
Library: LeafCell
Cell: CurrentMirror
View: schematic

To
Library: LeafCell
Cell: CurrentMirror_PF
View: schematic

Options
Copy Hierarchical
[✓] Skip Libraries: ANALOG MyDyanmic NCSU_Analog_Parts
[✓] Copy All Views
Views To Copy: schematic
Update Instances: Of Entire Library

OK Apply Cancel Help

Copy View

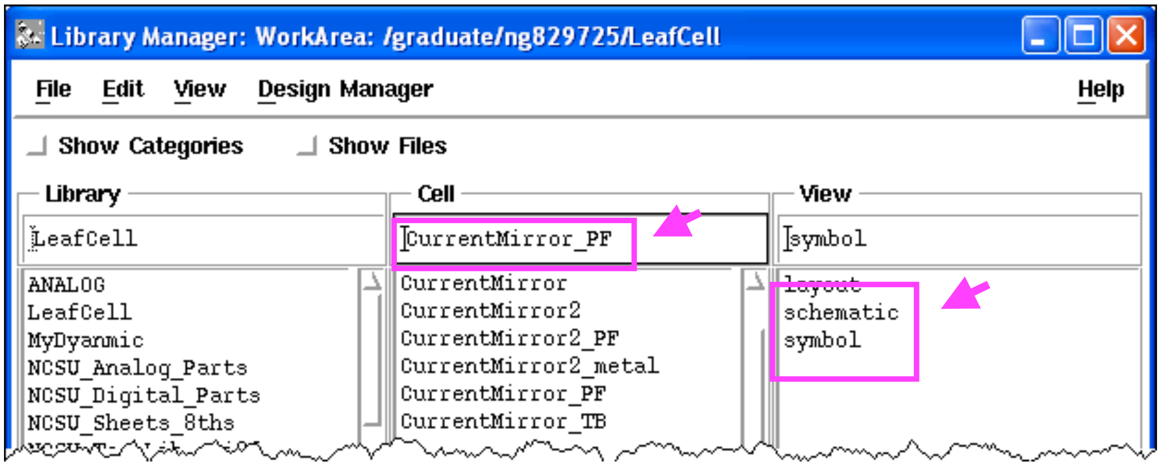
From
Library: LeafCell
Cell: CurrentMirror
View: symbol

To
Library: LeafCell
Cell: CurrentMirror_PF
View: symbol

Options
Copy Hierarchical
[✓] Skip Libraries: ANALOG MyDyanmic NCSU_Analog_Parts
[✓] Copy All Views
Views To Copy: schematic
Update Instances: Of Entire Library

OK Apply Cancel Help

Figure 114: Copying the schematic view and symbol view of the current mirror from cell *CurrentMirror* to cell *CurrentMirror_PF*.



Library Manager: WorkArea: /graduate/ng829725/LeafCell

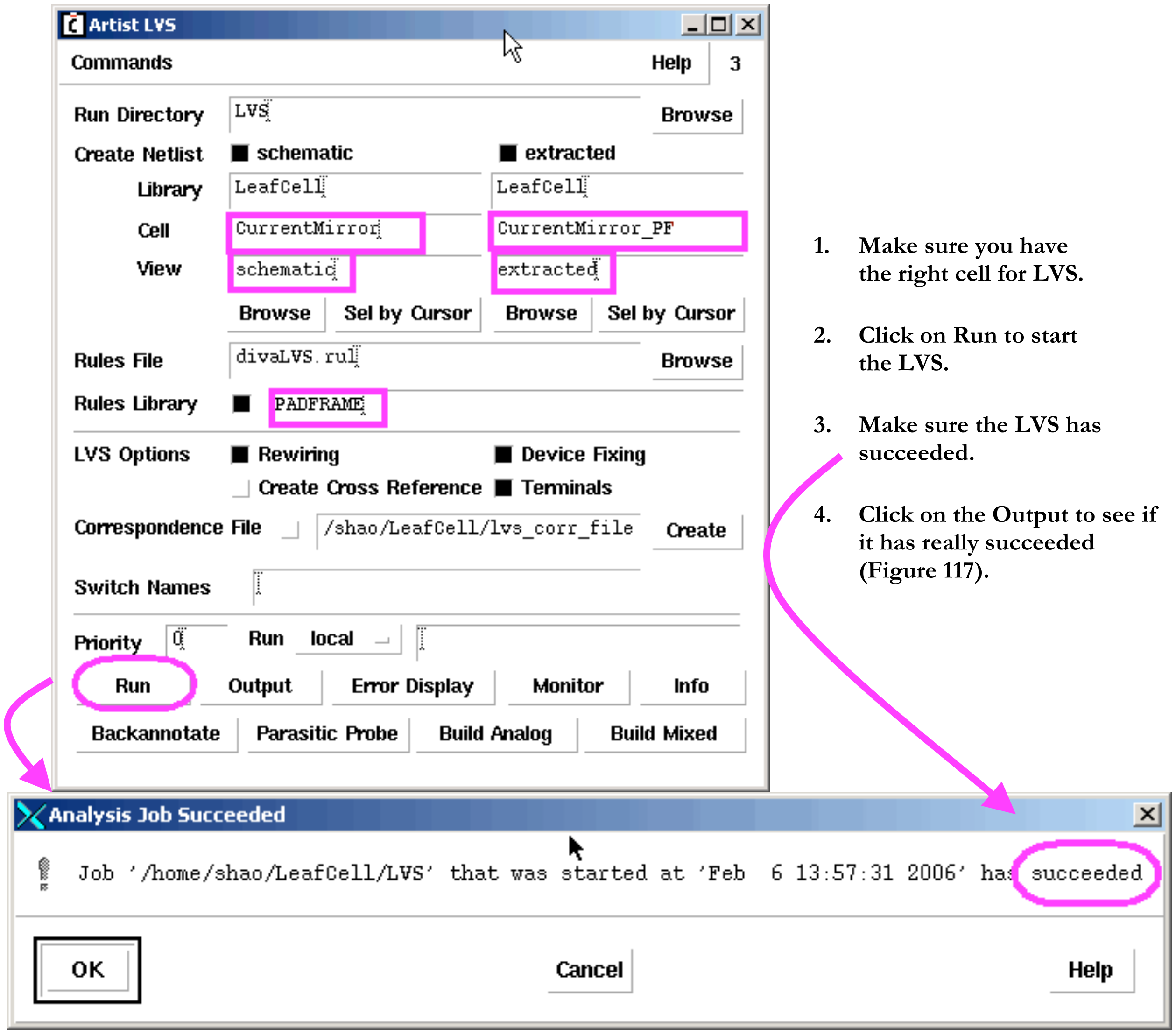
Library: LeafCell — Cell: CurrentMirror_PF — View: symbol

```
ANALOG                 CurrentMirror            layout
LeafCell               CurrentMirror2           schematic
MyDyanmic              CurrentMirror2_PF        symbol
NCSU_Analog_Parts      CurrentMirror2_metal
NCSU_Digital_Parts     CurrentMirror_PF
NCSU_Sheets_8ths       CurrentMirror_TB
```

Figure 115: Library Manager window after copying.

15. *Extract* the *CurrentMirror_PF*.

16. Run LVS for your PF (Figure 116).



Artist LVS

Run Directory: LVS

Create Netlist: ■ schematic ■ extracted

	schematic	extracted
Library	LeafCell	LeafCell
Cell	CurrentMirror	CurrentMirror_PF
View	schematic	extracted

Rules File: divaLVS.rul

Rules Library: ■ PADFRAME

LVS Options: ■ Rewiring ■ Device Fixing
Create Cross Reference ■ Terminals

Correspondence File: /shao/LeafCell/lvs_corr_file

Switch Names:

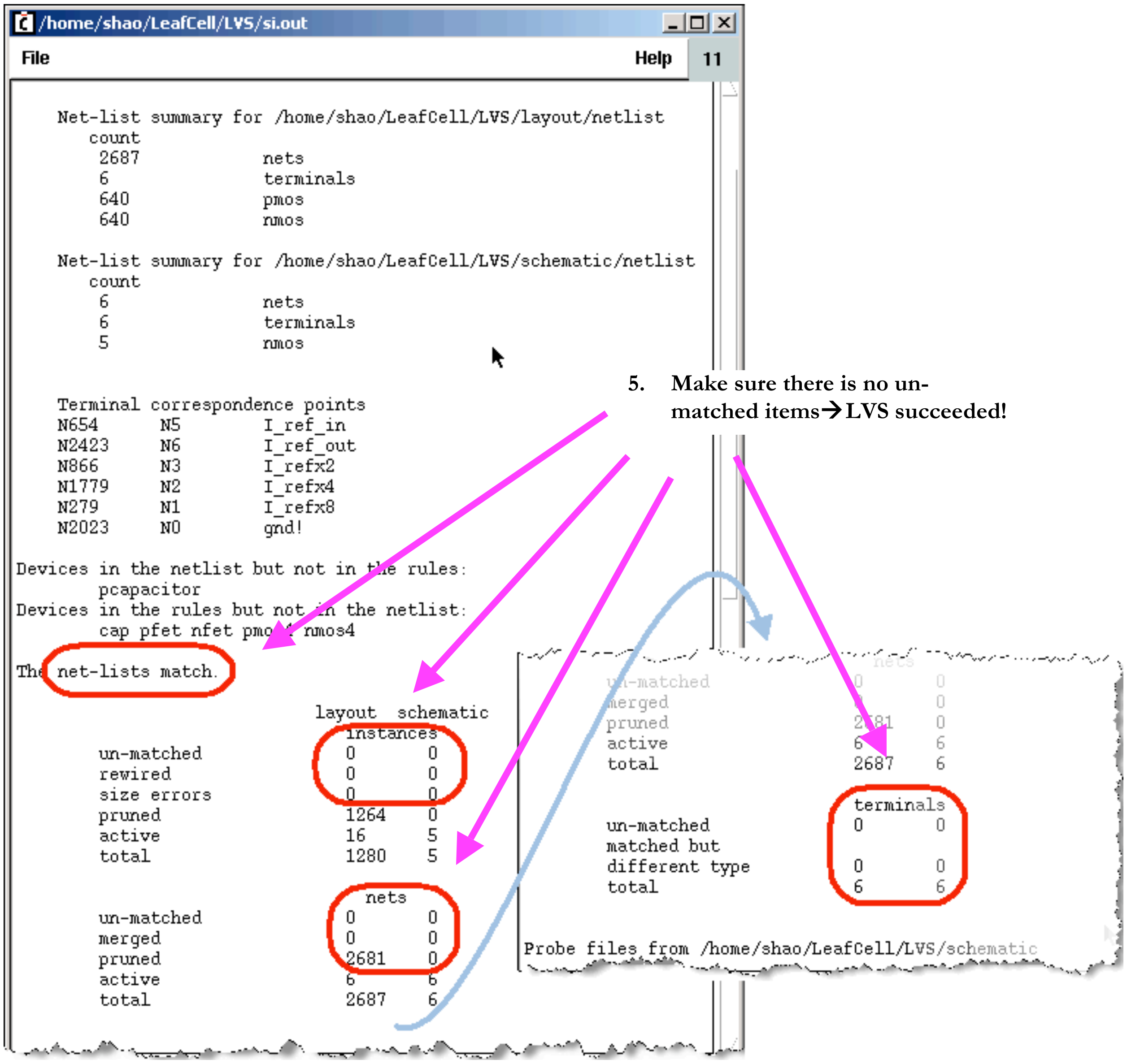
Priority: 0 Run: local

Analysis Job Succeeded

```
Job '/home/shao/LeafCell/LVS' that was started at 'Feb  6 13:57:31 2006' has succeeded
```

1. **Make sure you have the right cell for LVS.**
2. **Click on Run to start the LVS.**
3. **Make sure the LVS has succeeded.**
4. **Click on the Output to see if it has really succeeded (Figure 117).**

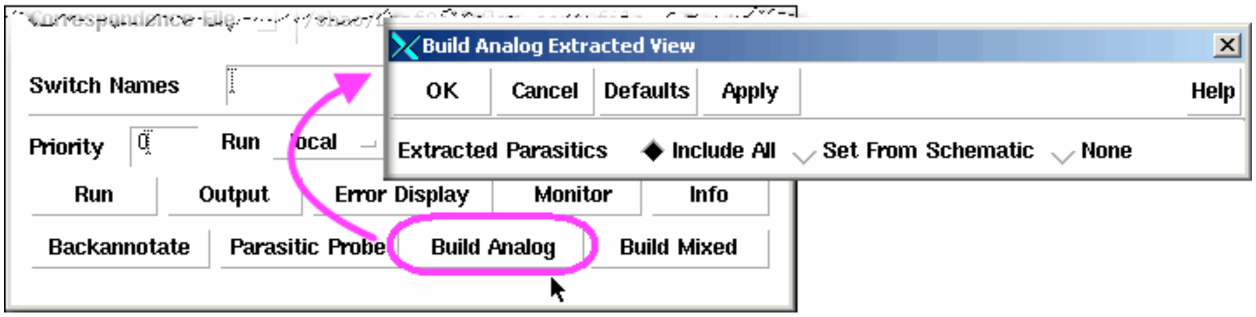
Figure 116: Running LVS for PF.


/home/shao/LeafCell/LVS/si.out
File
Help
11
5. Make sure there is no un-matched items→LVS succeeded!

```
   Net-list summary for /home/shao/LeafCell/LVS/layout/netlist
      count
       2687            nets
       6               terminals
       640             pmos
       640             nmos

   Net-list summary for /home/shao/LeafCell/LVS/schematic/netlist
      count
       6               nets
       6               terminals
       5               nmos


   Terminal correspondence points
   N654      N5        I_ref_in
   N2423     N6        I_ref_out
   N866      N3        I_refx2
   N1779     N2        I_refx4
   N279      N1        I_refx8
   N2023     N0        gnd!

Devices in the netlist but not in the rules:
        pcapacitor
Devices in the rules but not in the netlist:
        cap pfet nfet pmos4 nmos4

The net-lists match.

                                layout  schematic
                                   instances
       un-matched                  0       0
       rewired                     0       0
       size errors                 0       0
       pruned                      1264    0
       active                      16      5
       total                       1280    5

                                      nets
       un-matched                  0       0
       merged                      0       0
       pruned                      2681    0
       active                      6       6
       total                       2687    6

                                     terminals
       un-matched                  0       0
       matched but
       different type              0       0
       total                       6       6

Probe files from /home/shao/LeafCell/LVS/schematic
```

Figure 117: Running LVS for the PF.

17. After the Net-lists match, go back to the LVS form to ***Build Analog***.(Figure 118)



Extracted Parasitics [✓ Include All] Set From Schematic None

Figure 118: Build Analog for the PF

Now it is time to simulate the current mirror in the PF. Before doing this, we need to change the symbol of the current mirror in the test bench.

18. Open the *CurrentMirror_TB* schematic. Replace the part *CurrentMirror* with **CurrentMirror_PF** as in Figure 119.

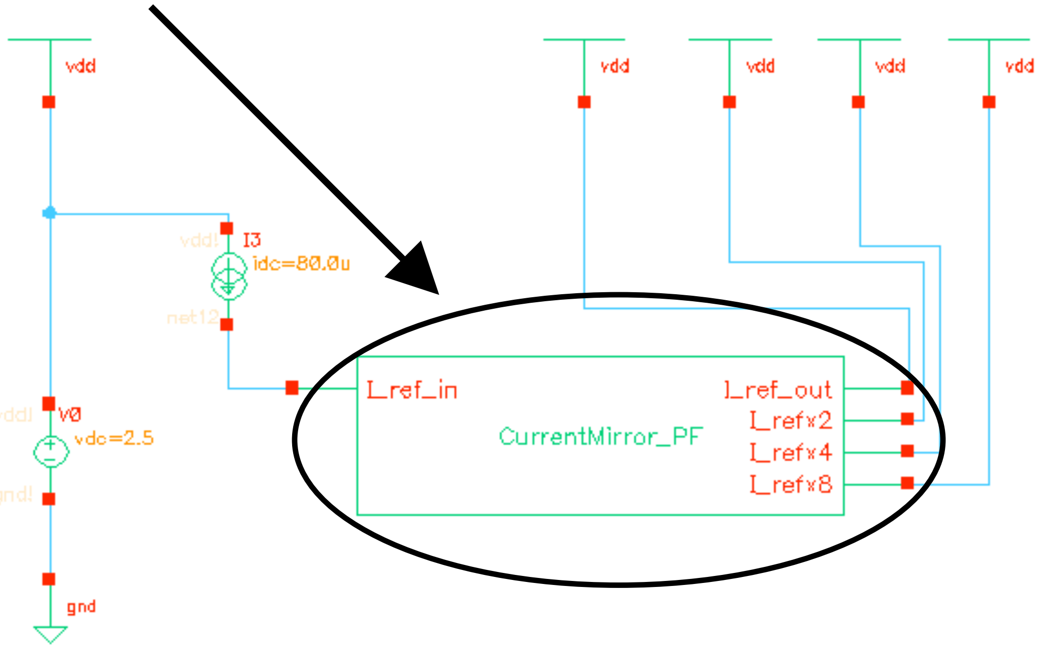
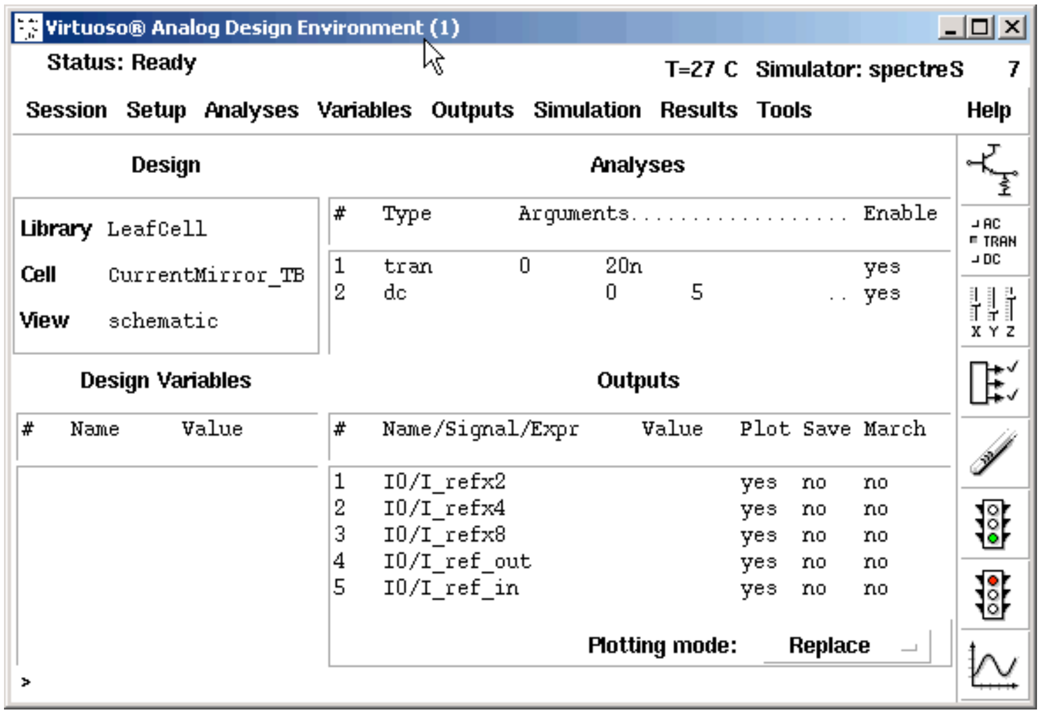


Figure 119: Replacing the CurrentMirror in the test bench with CurrentMirror_PF.

19. Now open the Analog Environment window (Figure 120) again. Go to ***Session…Load State*** to load ***State1***, which is the one we save earlier in the tutorial.



Design

Library	LeafCell
Cell	CurrentMirror_TB
View	schematic

Analyses

#	Type	Arguments.................			Enable
1	tran	0	20n		yes
2	dc		0	5 ..	yes

Design Variables

#	Name	Value

Outputs

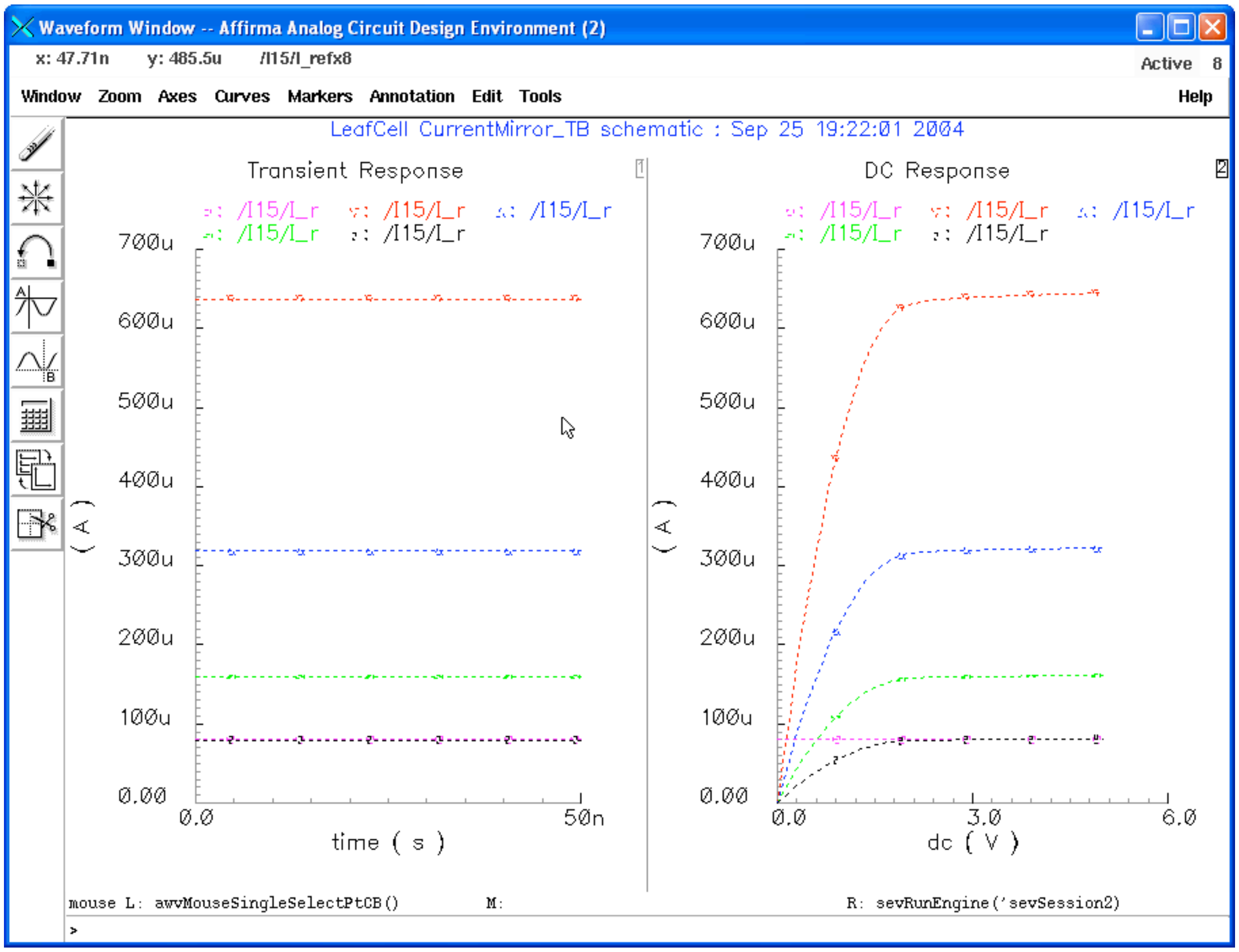
#	Name/Signal/Expr	Value	Plot	Save	March
1	I0/I_refx2		yes	no	no
2	I0/I_refx4		yes	no	no
3	I0/I_refx8		yes	no	no
4	I0/I_ref_out		yes	no	no
5	I0/I_ref_in		yes	no	no

Plotting mode: Replace

>

Figure 120: Setting up the Affirma to simulate the PF.

19. Run the simulation, and then the result should look like Figure 121.


Waveform Window -- Affirma Analog Circuit Design Environment (2)
x: 47.71n
y: 485.5u
/I15/I_refx8
Active 8
Window Zoom Axes Curves Markers Annotation Edit Tools Help
LeafCell CurrentMirror_TB schematic : Sep 25 19:22:01 2004
Transient Response
DC Response
/I15/I_r
700u
600u
500u
400u
300u
200u
100u
0.00
(A)
0.0
50n
time (s)
3.0
6.0
dc (V)

```
mouse L: awvMouseSingleSelectPtCB()          M:          R: sevRunEngine('sevSession2)
>
```

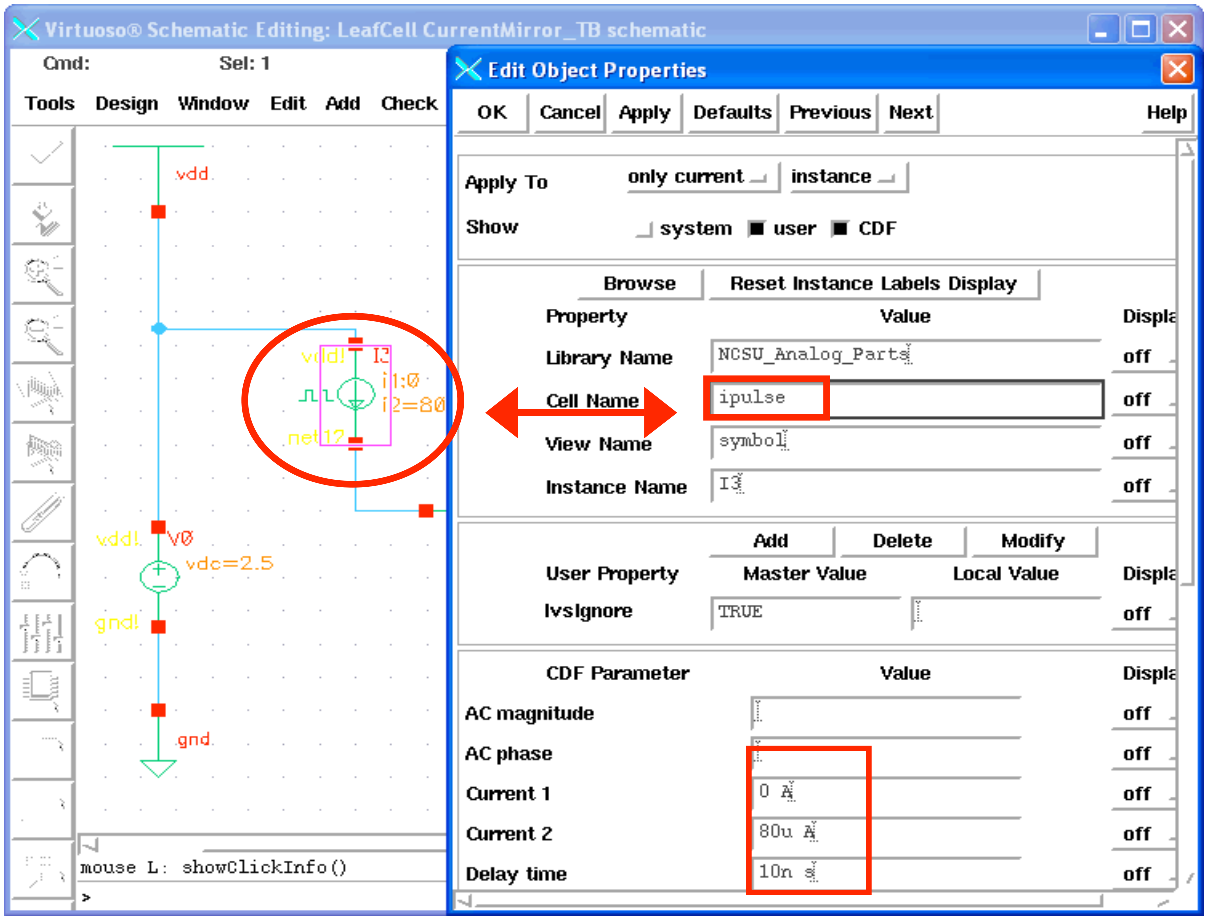
Figure 121: Setting up the Affirma to simulate the PF.

The results are within 1% error. Therefore, the CurrentMirror design is done.

Simulation for the Slew Rate:

In order to find out how the current mirror reacts with loads as the reference input current switches, we need to do a step transient responds. Before we start the simulation, we need to make some changes in the test bench.

1. Go to CurrentMirror_TB, click at the current source (Figure 122), keying "Q" to bring out the *property editor*, and then replace the part **idc** with **ipulse**. Set it up as in Figure 122. After the editing, click OK to conform.


Virtuoso® Schematic Editing: LeafCell CurrentMirror_TB schematic
Cmd: Sel: 1
Tools Design Window Edit Add Check
vdd
vdd! I3
i1:0
i2=80
net12
vdd! V0
vdc=2.5
gnd!
gnd
Edit Object Properties
OK Cancel Apply Defaults Previous Next Help

Apply To: only current, instance
Show: system, ■ user, ■ CDF

Browse | Reset Instance Labels Display

Property	Value	Displa
Library Name	NCSU_Analog_Parts	off
Cell Name	ipulse	off
View Name	symbol	off
Instance Name	I3	off

Add | Delete | Modify

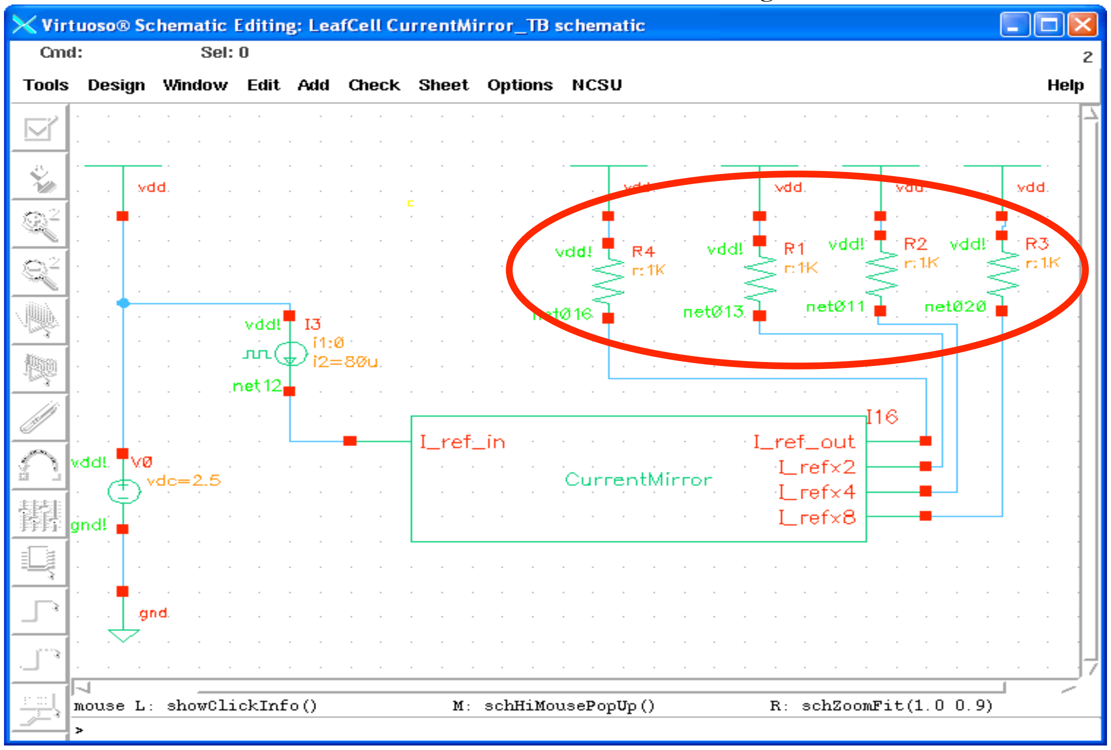
User Property	Master Value	Local Value	Displa
lvsIgnore	TRUE		off

CDF Parameter	Value	Displa
AC magnitude		off
AC phase		off
Current 1	0 A	off
Current 2	80u A	off
Delay time	10n s	off

```
mouse L: showClickInfo()
>
```

Figure 122: Editing the current source.

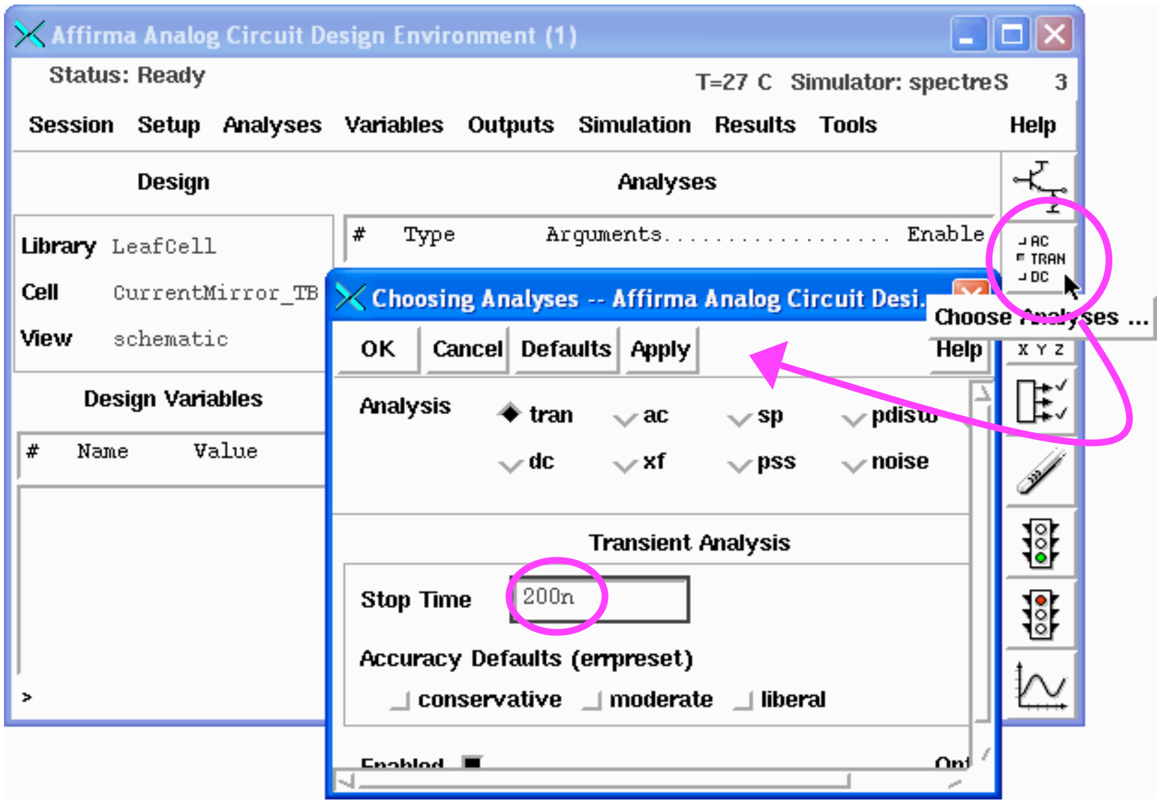
2. Add the 1K resistor loads to the current mirror as in Figure 123.



```
mouse L: showClickInfo()          M: schHiMousePopUp()          R: schZoomFit(1.0 0.9)
>
```

Figure 123: Adding in the load resistors.

3. Check and Save the test bench.
4. Go to Tools to open the *Analog Environment*. Make sure to set up the *Simulator/Directory/Host* as well as the *Design*.
5. Click to open the Choose Analysis Form (Figure 124.) Set it up as in Figure 124.



Affirma Analog Circuit Design Environment (1)
Status: Ready
T=27 C Simulator: spectreS 3
Session Setup Analyses Variables Outputs Simulation Results Tools Help
Design
Analyses
Design Variables
Choosing Analyses -- Affirma Analog Circuit Desi...
OK Cancel Defaults Apply Help
Choose Analyses ...
Analysis tran ac sp pdisto dc xf pss noise
Transient Analysis
Accuracy Defaults (errpreset)
conservative moderate liberal
Enabled

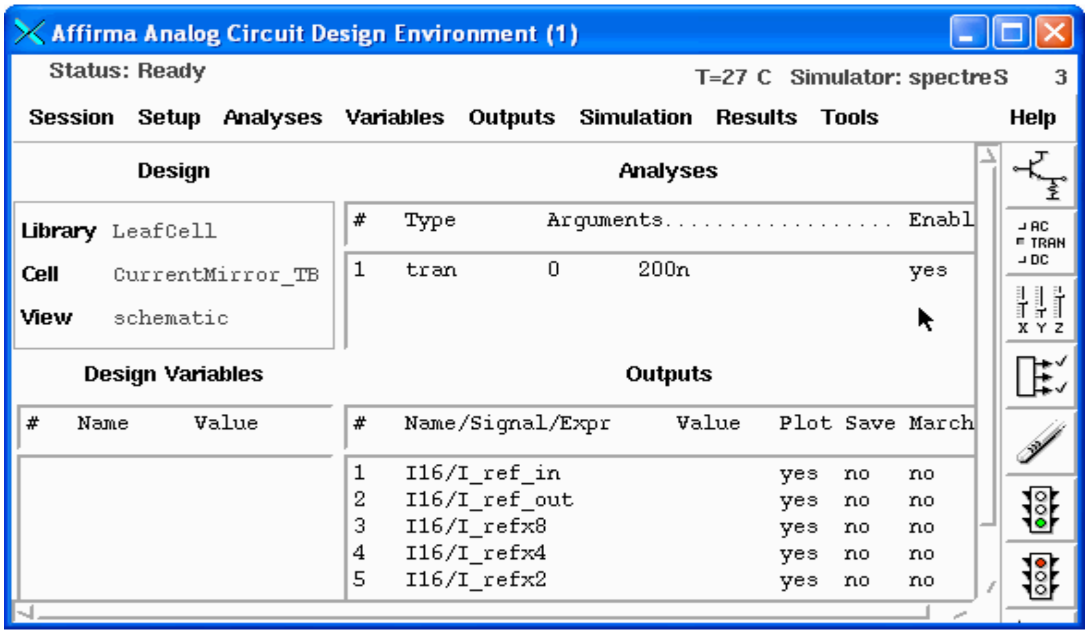
Library LeafCell
Cell CurrentMirror_TB
View schematic

Analysis: [✓ tran]

Stop Time 200n

Figure 124: Setting up the Affirma for simulation.

6. Select the input and outputs of the current mirror to be the Outputs…To Be Plotted… (Figure 125).



Affirma Analog Circuit Design Environment (1)
Status: Ready
T=27 C Simulator: spectreS 3
Session Setup Analyses Variables Outputs Simulation Results Tools Help
Design
Analyses
Design Variables
Outputs

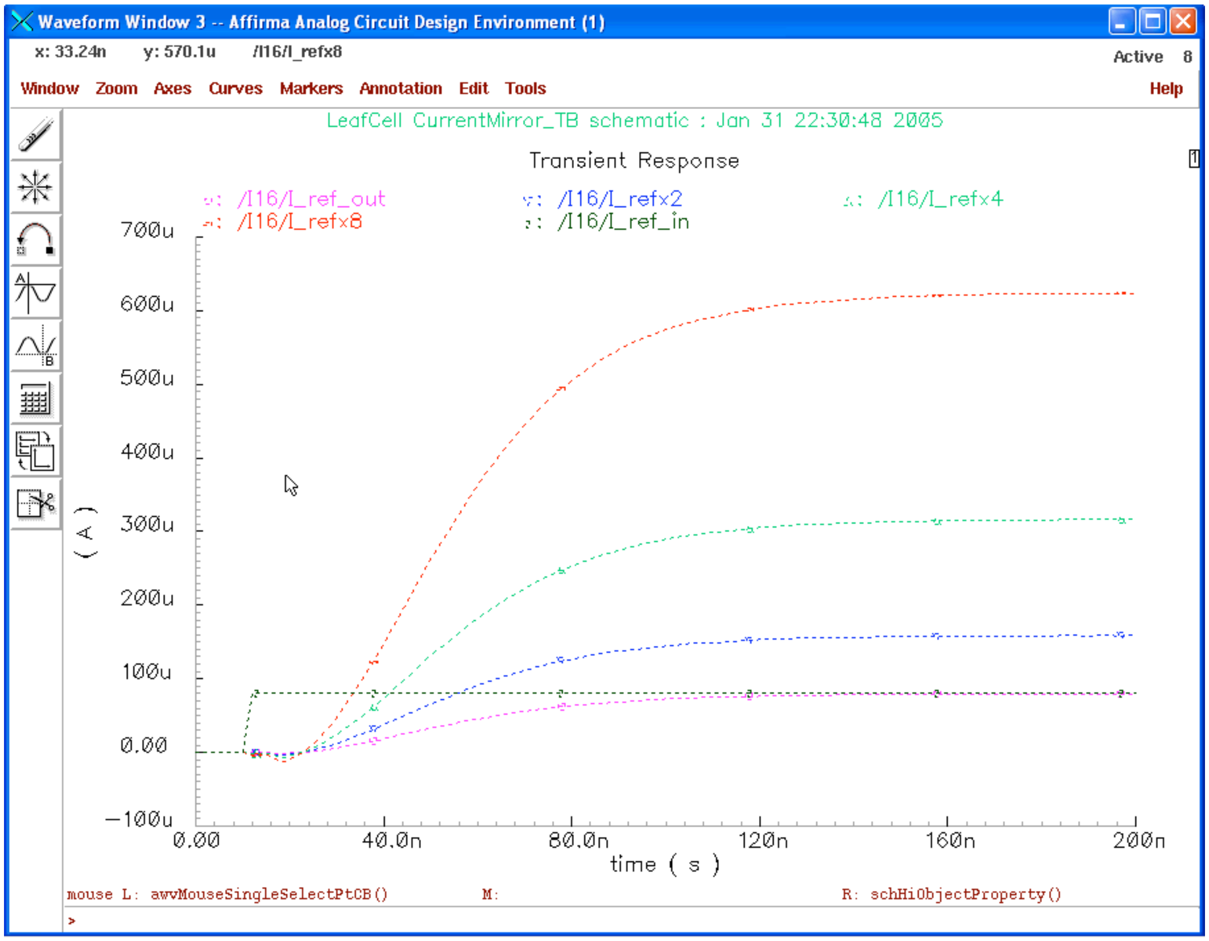
Library LeafCell
Cell CurrentMirror_TB
View schematic

#	Type	Arguments.................		Enabl
1	tran	0	200n	yes

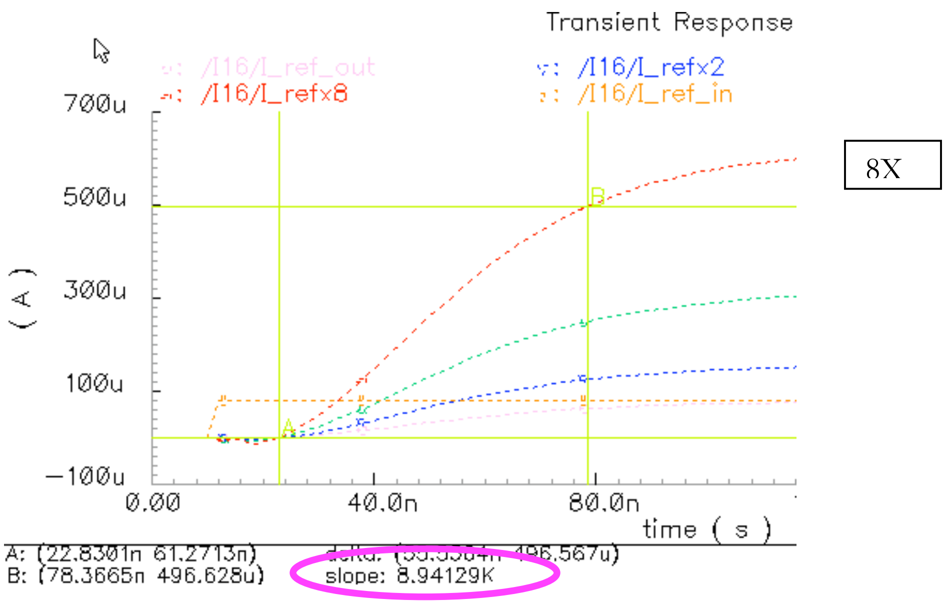
#	Name/Signal/Expr	Value	Plot	Save	March
1	I16/I_ref_in		yes	no	no
2	I16/I_ref_out		yes	no	no
3	I16/I_refx8		yes	no	no
4	I16/I_refx4		yes	no	no
5	I16/I_refx2		yes	no	no

Figure 125: Setting up the Affirma for simulation.

7. Run the simulation. The resulting wave form will be shown as in Figure 126.

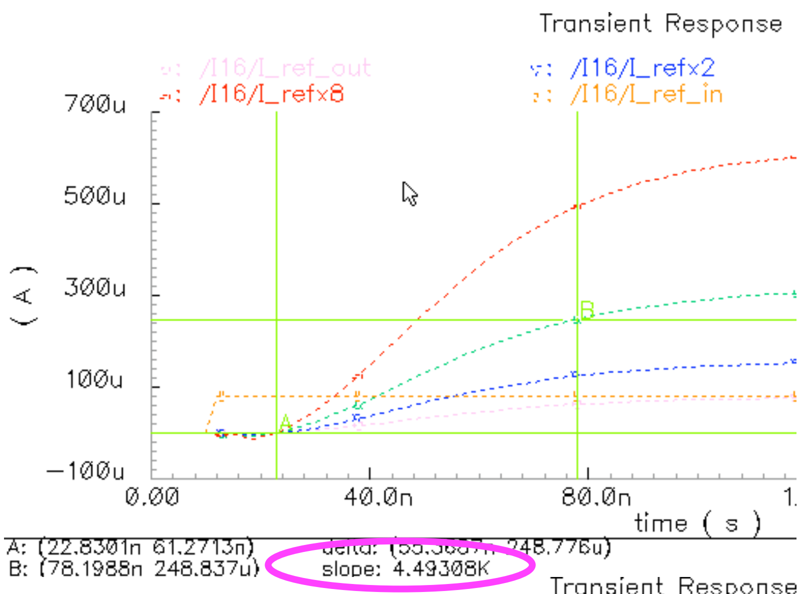

Waveform Window 3 -- Affirma Analog Circuit Design Environment (1)
x: 33.24n y: 570.1u /I16/I_refx8
Active 8
Window Zoom Axes Curves Markers Annotation Edit Tools Help
LeafCell CurrentMirror_TB schematic : Jan 31 22:30:48 2005
Transient Response
/I16/I_ref_out
/I16/I_refx8
/I16/I_refx2
/I16/I_ref_in
/I16/I_refx4
700u
600u
500u
400u
300u
200u
100u
0.00
−100u
(A)
0.00
40.0n
80.0n
120n
160n
200n
time (s)
mouse L: awvMouseSingleSelectPtCB() M: R: schHiObjectProperty()
>
Figure 126: Step transient response waveform.

8. To obtain the slew rate for the current responses, use Crosshair Marker A and B (Figure 127). The slope shown at the bottom is slower than what you are going to have, because the spice parameters have changed.



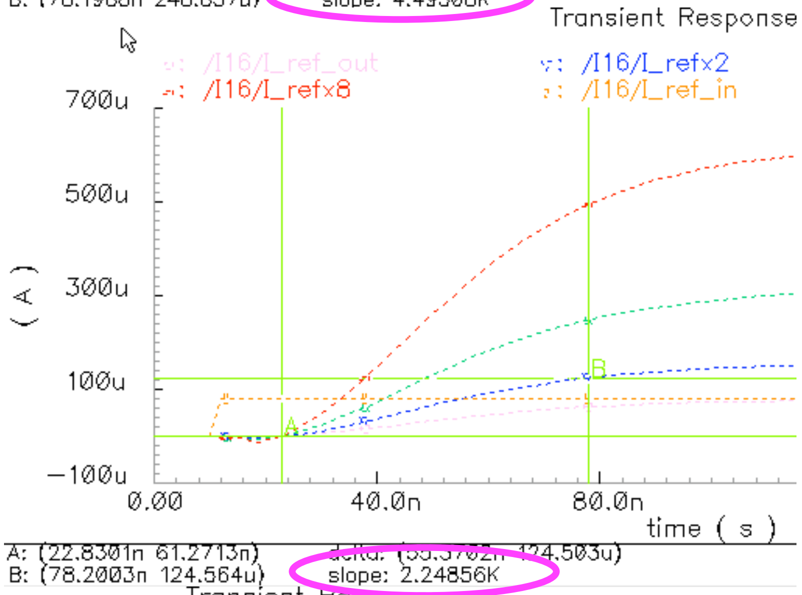
```
A: (22.8301n 61.2713n)      delta: (55.5364n 496.567u)
B: (78.3665n 496.628u)      slope: 8.94129K
```

4X

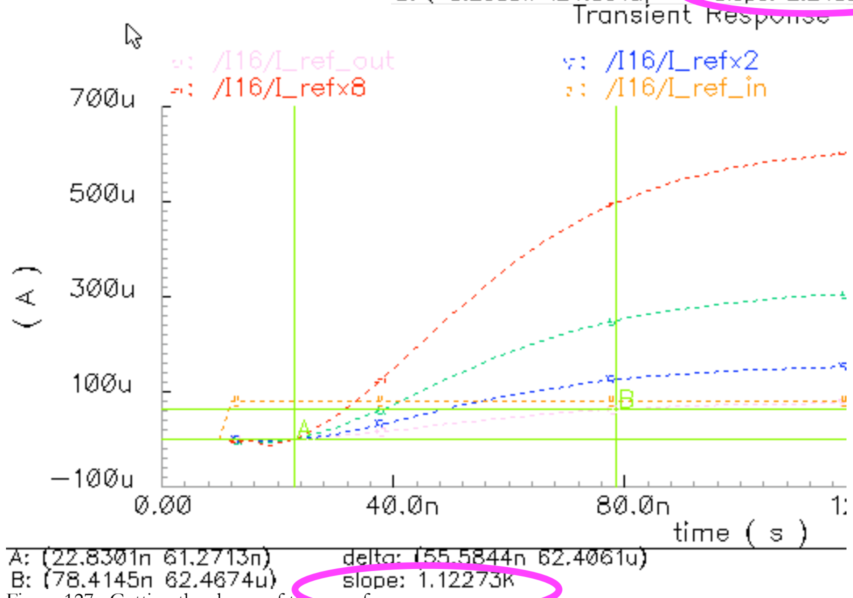


```
A: (22.8301n 61.2713n)     delta: (55.3687n 248.776u)
B: (78.1988n 248.837u)     slope: 4.49308K
```

2X



```
A: (22.8301n 61.2713n)     delta: (55.3702n 124.503u)
B: (78.2003n 124.564u)     slope: 2.24856K
```



1X

```
A: (22.8301n 61.2713n)     delta: (55.5844n 62.4061u)
B: (78.4145n 62.4674u)     slope: 1.12273K
```

Figure 127: Getting the slopes of the waveform

After Thoughts (Ask yourself & Try yourself):

1. If we zoom close into the waveform, right after the reference current switches, the outputs jump to the other direction before they run in the same direction as the reference (Figure 128). Why?

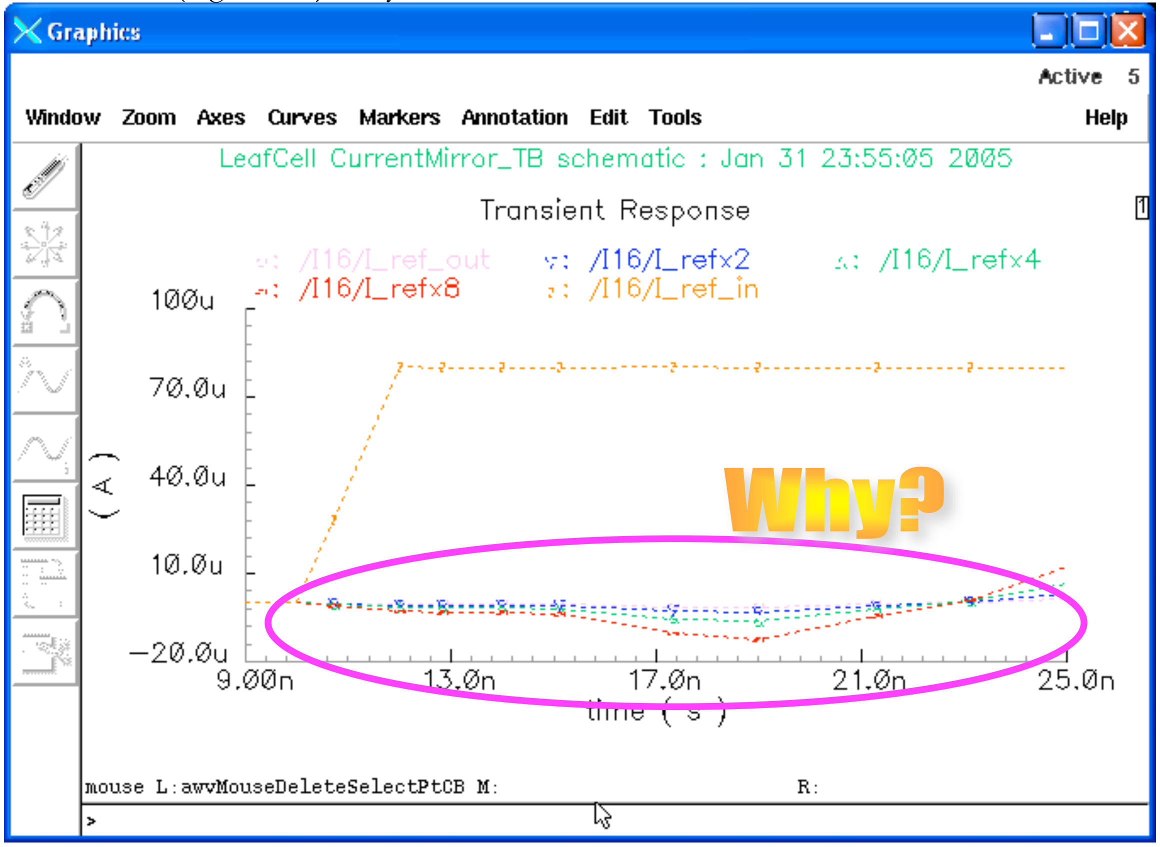
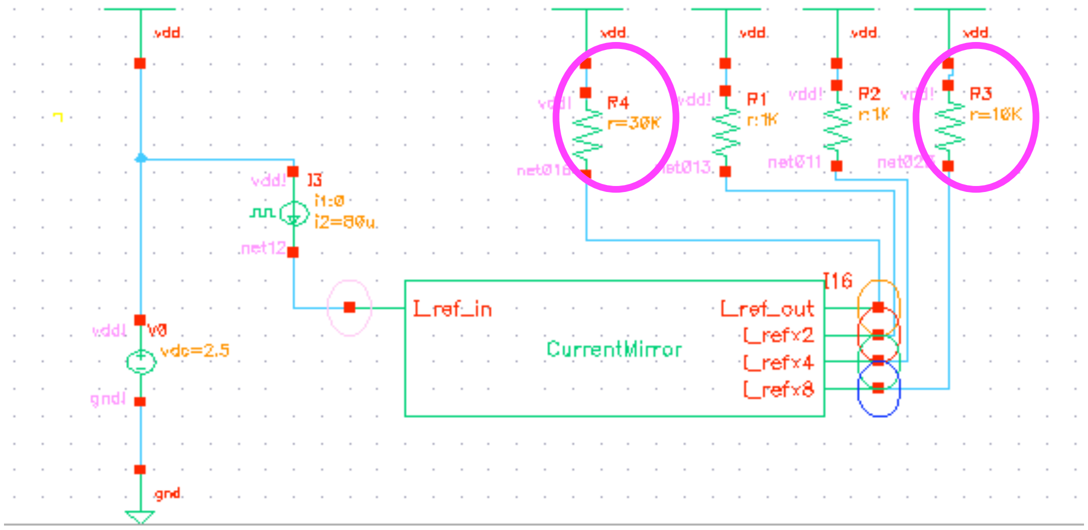


Figure 128: Dipping in the other direction

2. What happen if we change the load? How will the current mirror reacts?

For example, if we change the resistance load for I_ref_out to be 30K and the one for I_ref_x8 to be 10K, the resulting waveform differs a lot (Figure 129)



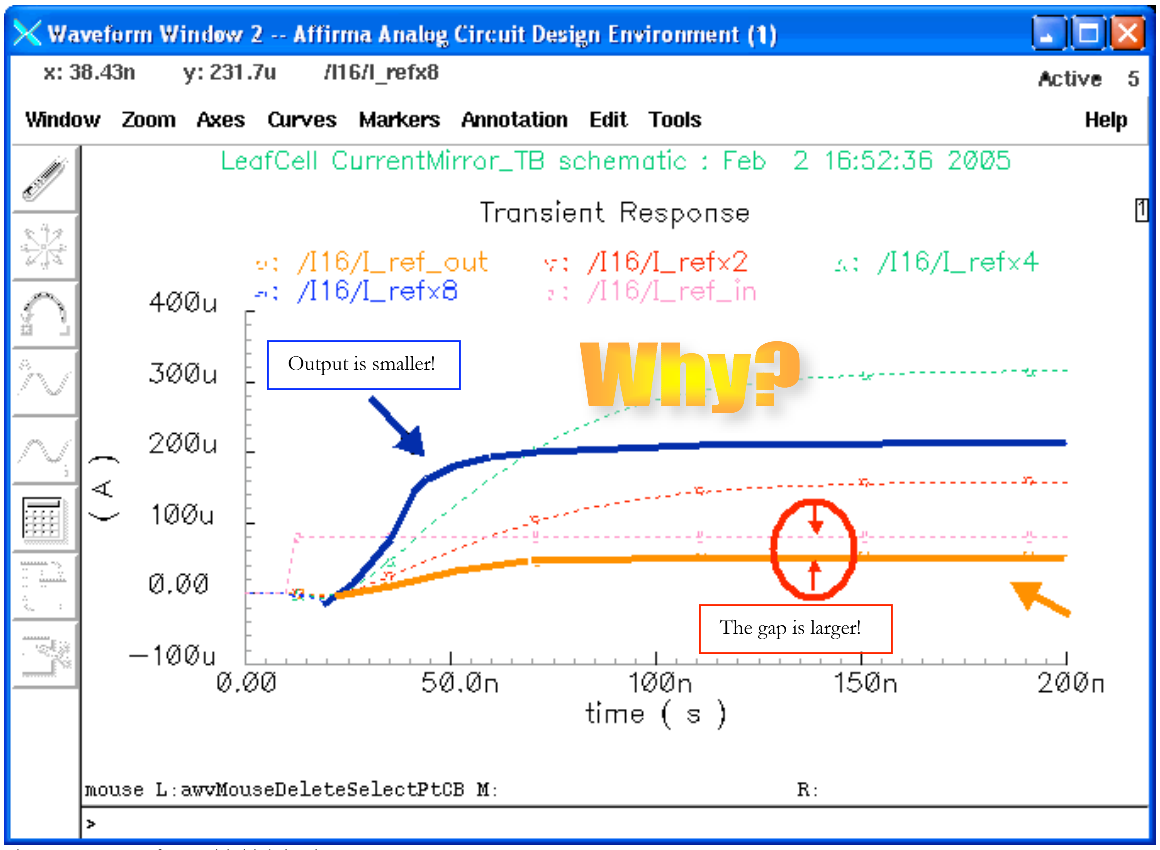

Waveform Window 2 -- Affirma Analog Circuit Design Environment (1)
x: 38.43n y: 231.7u /I16/I_refx8
Active 5
Window Zoom Axes Curves Markers Annotation Edit Tools Help
LeafCell CurrentMirror_TB schematic : Feb 2 16:52:36 2005
Transient Response
/I16/I_ref_out
/I16/I_refx2
/I16/I_refx4
/I16/I_refx8
/I16/I_ref_in
400u
300u
200u
100u
0.00
−100u
(A)
0.00
50.0n
100n
150n
200n
time (s)
Output is smaller!
Why?
The gap is larger!
mouse L:awvMouseDeleteSelectPtCB M: R:
>

Figure 129: Waveform with high loads.

If you zoom in to take a closer look at the wave, you will see some kind of osilation (Figure130). Why is that happening?

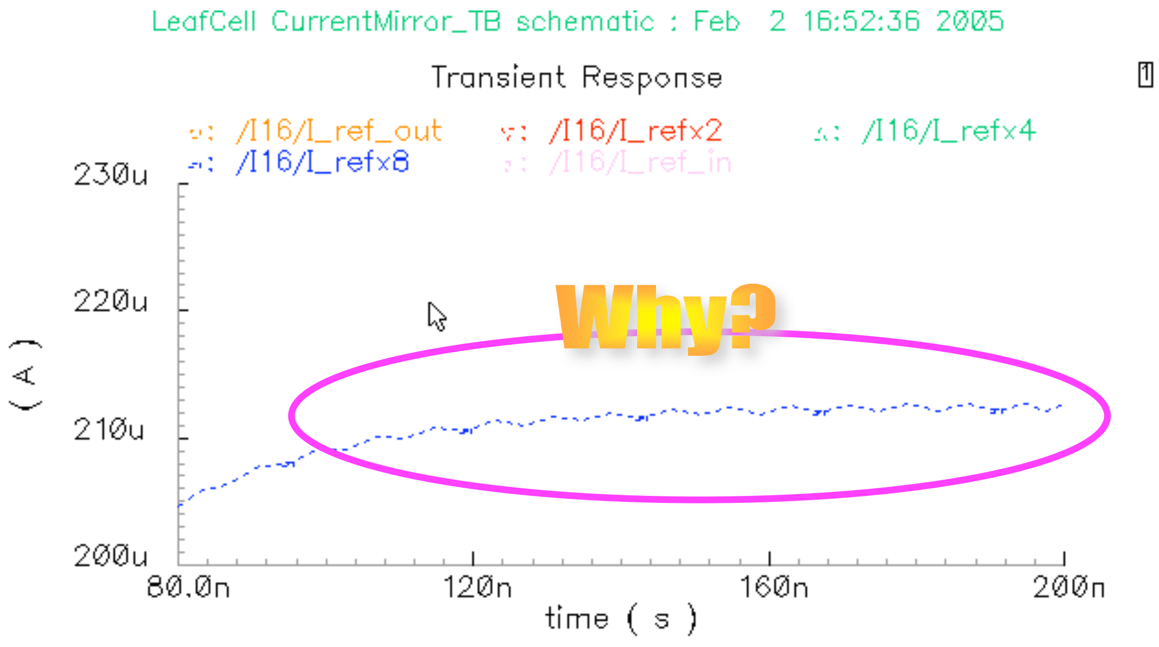
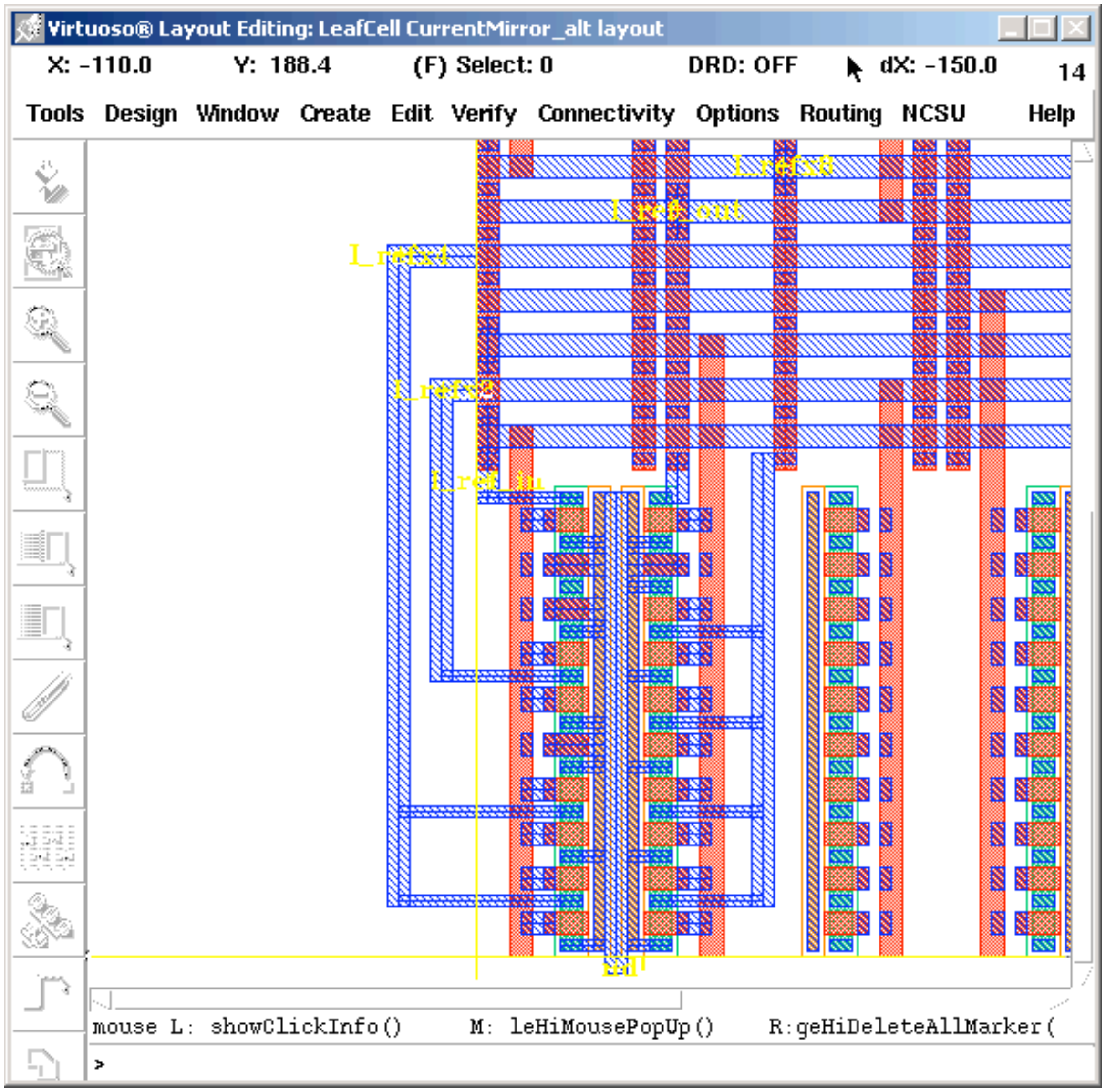


Figure 130: Why it oscilates?

Just try to play around with it, you'll find more questions. ^v^

Additional Way for the Same Design:

There is always more than one way to do designs. For this current mirror, you can lay it out in one leaf as in figure125. Can you figure out the input and outputs? Can you also tell which transistor is grounded?

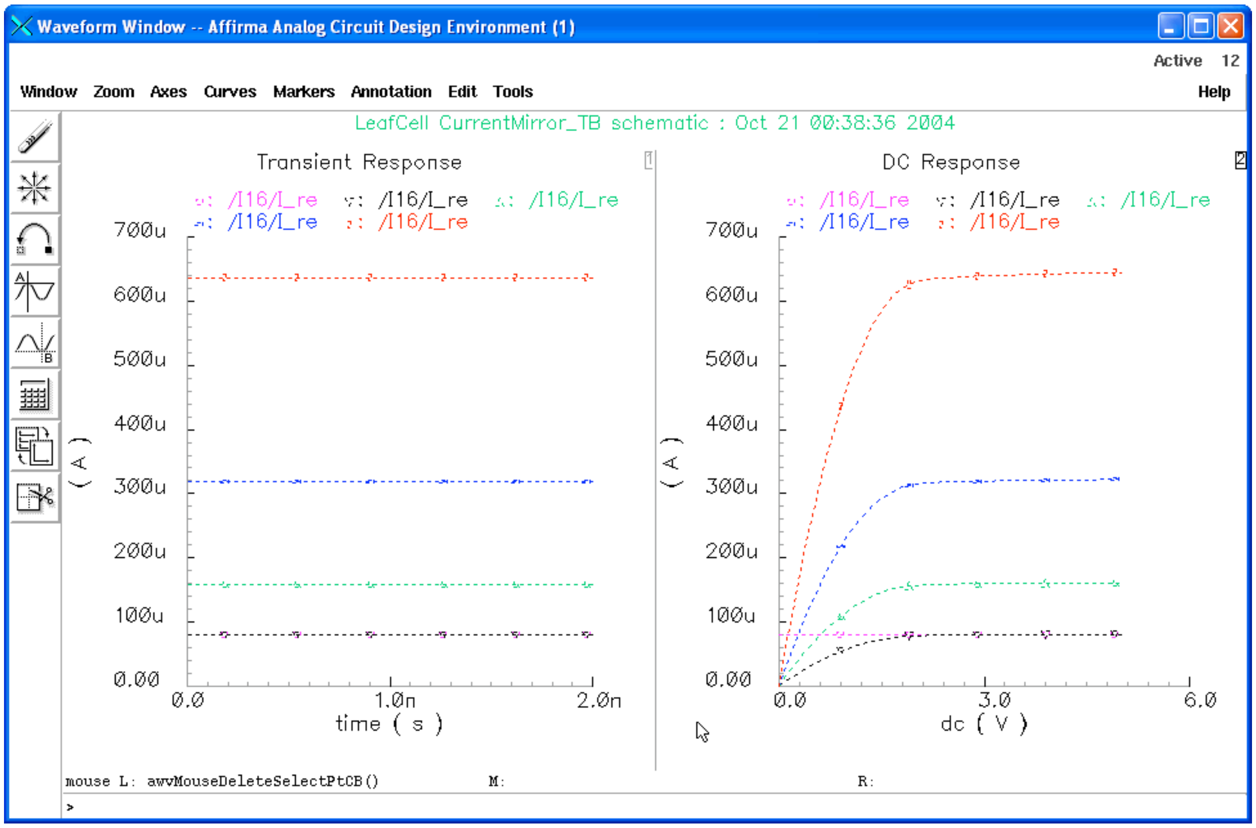


```
Virtuoso® Layout Editing: LeafCell CurrentMirror_alt layout
X: -110.0    Y: 188.4    (F) Select: 0    DRD: OFF    dX: -150.0    14
mouse L: showClickInfo()    M: leHiMousePopUp()    R:geHiDeleteAllMarker(
>
```

Figure 131: Another way to layout the CurrentMirror.

After the same design steps of DRC, LVS, and build analog, the simulation results are shown in Figure 132.

It works!


Waveform Window -- Affirma Analog Circuit Design Environment (1)
Active 12
Window
Zoom
Axes
Curves
Markers
Annotation
Edit
Tools
Help
LeafCell CurrentMirror_TB schematic : Oct 21 00:38:36 2004
Transient Response
DC Response
/I16/I_re
(A)
time (s)
dc (V)

```
mouse L: awvMouseDeleteSelectPtCB()          M:                    R:
>
```

Figure 132: Simulation waveform for the CurrentMirror laid out in figure 125.

Chapter 3: Design of a OTA

Section 1: Initial Design

Design specification:

I=80uA; Vdd=5V; Vss=0V; Cl=5pF GB>=1MHz.

Hand Calculation:

Some equations to be used:

$$C_c = .22(C_L)$$

$$I_5 = C_c * SR$$

$$S_3 = (W/L)_3 = \frac{I_5}{K_3'[V_{dd} - V_{in(max)} + V_{t3} + V_{tn}]^2}$$

$$S_4 = S_3$$

$$p_3 \approx \frac{-g_{m3}}{2C_{gs3}} = \frac{-\sqrt{2K_p' S_3 I_3}}{2(.667)W_3 L_3 C_{ox}}$$

To make sure the mirror pole P3 is greater than the dominant pole defined the GB

$$g_{m1} = GB * C_c$$

$$S_1 = S_2 = \frac{G^2{}_{m1}}{2K_n' I_1}$$

$$V_{in(min)} = V_{gs5} + V_{t5}$$

$$V_{ds5} = V_{in(min)} - Vss - \sqrt{(I_5/\beta_1)} - V_{t1}$$

$$S5 = \frac{2I_5}{K_5'(V_{ds5})^2}$$

$$g_{m6} = 2.2(g_{m2})(C_L/C_c)$$

$$S_6 = S_4(g_{m6}/g_{m4})$$

or

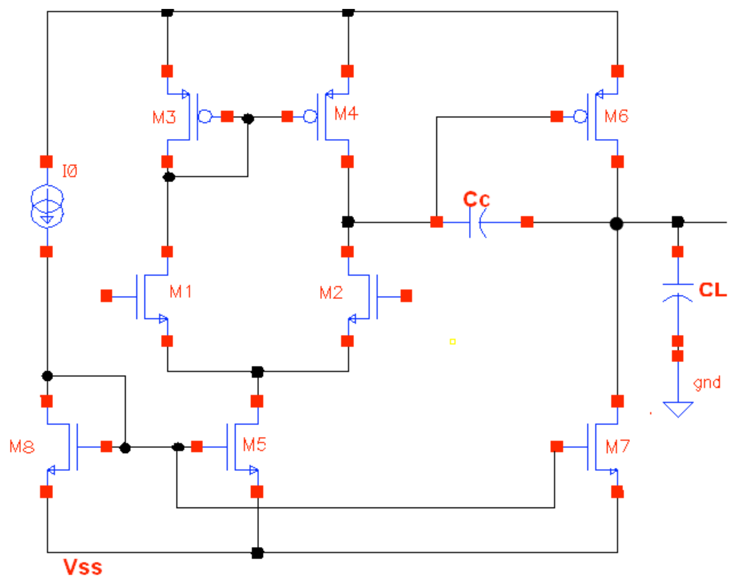
$$S_6 = \frac{g_{m6}}{K_6' V_{ds6(sat)}} \text{(depend on trade-offs)}$$

$$I_6 = \frac{g^2{}_{m6}}{2K_6' S_6}$$

$$S_7 = S_5(I_6/I_5)$$

$$A_v = \frac{2g_{m2}g_{m6}}{I_5(\lambda_2+\lambda_3)(\lambda_6+\lambda_7)}$$

$$P_{diss} = (I_6 + I_5)(V_{dd} + |V_{ss}|)$$



One example for the transistor sizes:

The equations set in the previous page shows the step by step calculation for the OTA. However, in this design, instead of starting from sizing the active load (M3 & M4) in the circuit, I started from sizing the smallest transistors, which are M1 and M2 in the above OTA structure. It is because if we don't start from the smallest transistors, by following the above calculation order, the ration of the W/L will be less than one for the smallest transistors. In another words, according to the ALC templates, we need to connect the basic transistor units in series, resulting the effective length to be much bigger than the minimum channel length. We don't want that! Therefore, I assume **$(W/L)_{1,2}$** to be **6.4□m/6.4□m**, which is the minimum transistor unit size.

The calculated results are shown as follows:

Tansistor:	1	2	3	4	5	6	7	8
S	1	1	3	3	8	40	30	8
W/L	6.4/6.4	6.4/6.4	19.2/6.4	19.2/6.4	51.2/6.4	256/6.4	192/6.4	51.2/6.4
Cc = 1.2pF								

The resulted schematic will be in Figure 133.

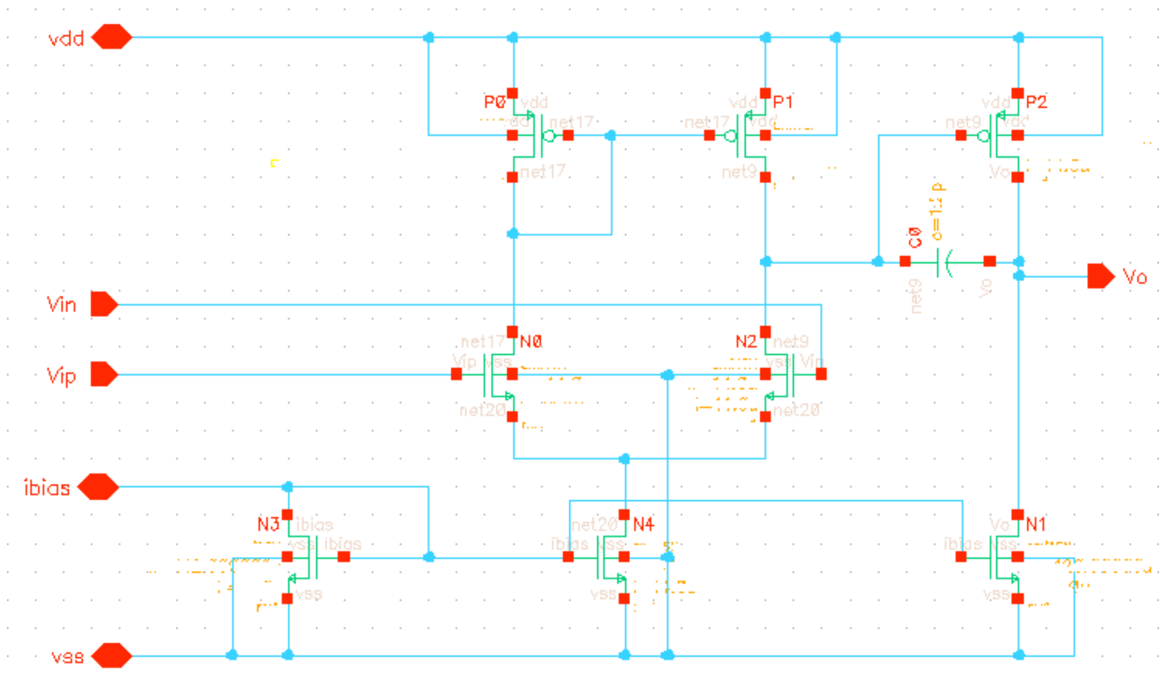
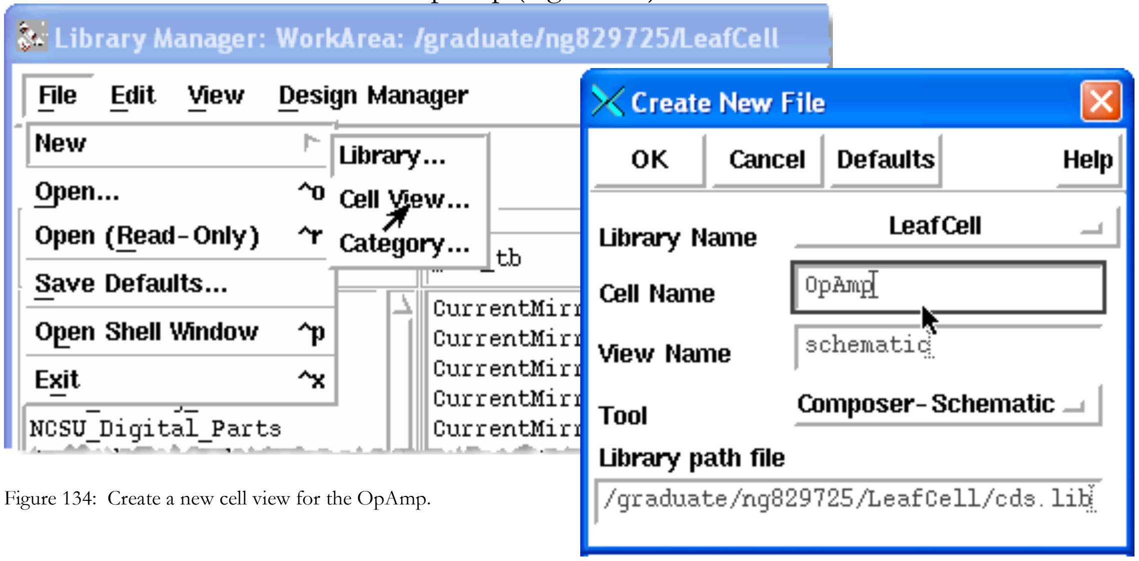


Figure 133: Parameter sizing for the OTA.

Section 2: Getting started with Schematic Capture and Spice Simulation

Creating the schematic view:

1. Create a new cell view for the OpAmp (Figure 134)


Library Manager: WorkArea: /graduate/ng829725/LeafCell

File Edit View Design Manager

New ▸ Library... Cell View... Category...
Open... ^o
Open (Read-Only) ^r
Save Defaults...
Open Shell Window ^p
Exit ^x

NCSU_Digital_Parts

CurrentMirr
CurrentMirr
CurrentMirr
CurrentMirr
CurrentMirr

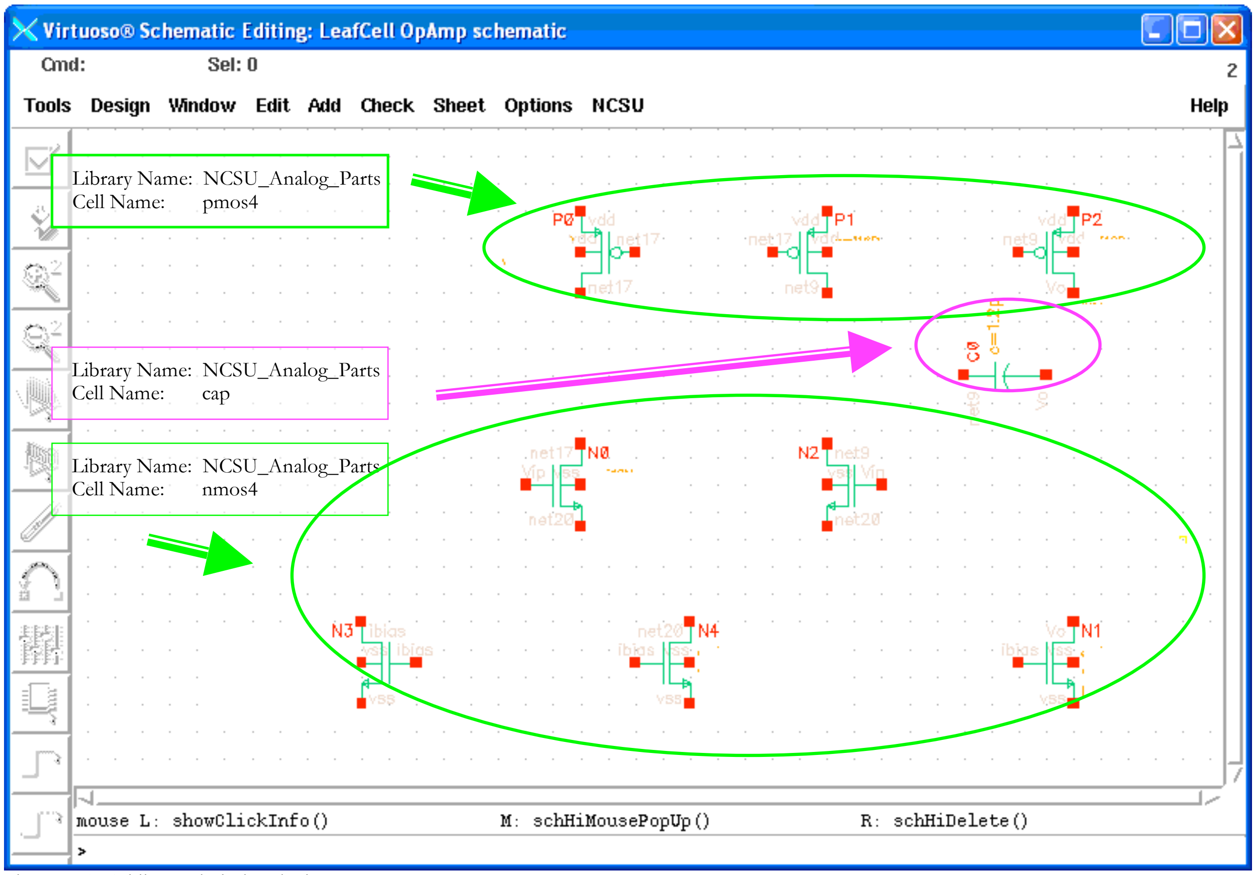
Create New File

OK Cancel Defaults Help

Library Name: LeafCell
Cell Name: OpAmp
View Name: schematic
Tool: Composer-Schematic
Library path file: /graduate/ng829725/LeafCell/cds.lib

Figure 134: Create a new cell view for the OpAmp.

2. Adding Instances and place them correspondingly(Figure 135)


Virtuoso® Schematic Editing: LeafCell OpAmp schematic

Cmd: Sel: 0 2

Tools Design Window Edit Add Check Sheet Options NCSU Help

Library Name: NCSU_Analog_Parts
Cell Name: pmos4

Library Name: NCSU_Analog_Parts
Cell Name: cap

Library Name: NCSU_Analog_Parts
Cell Name: nmos4

```
mouse L: showClickInfo()          M: schHiMousePopUp()          R: schHiDelete()
>
```

Figure 135: Adding and placing the instances.

3. Adding the input and output pins(Figure 136):

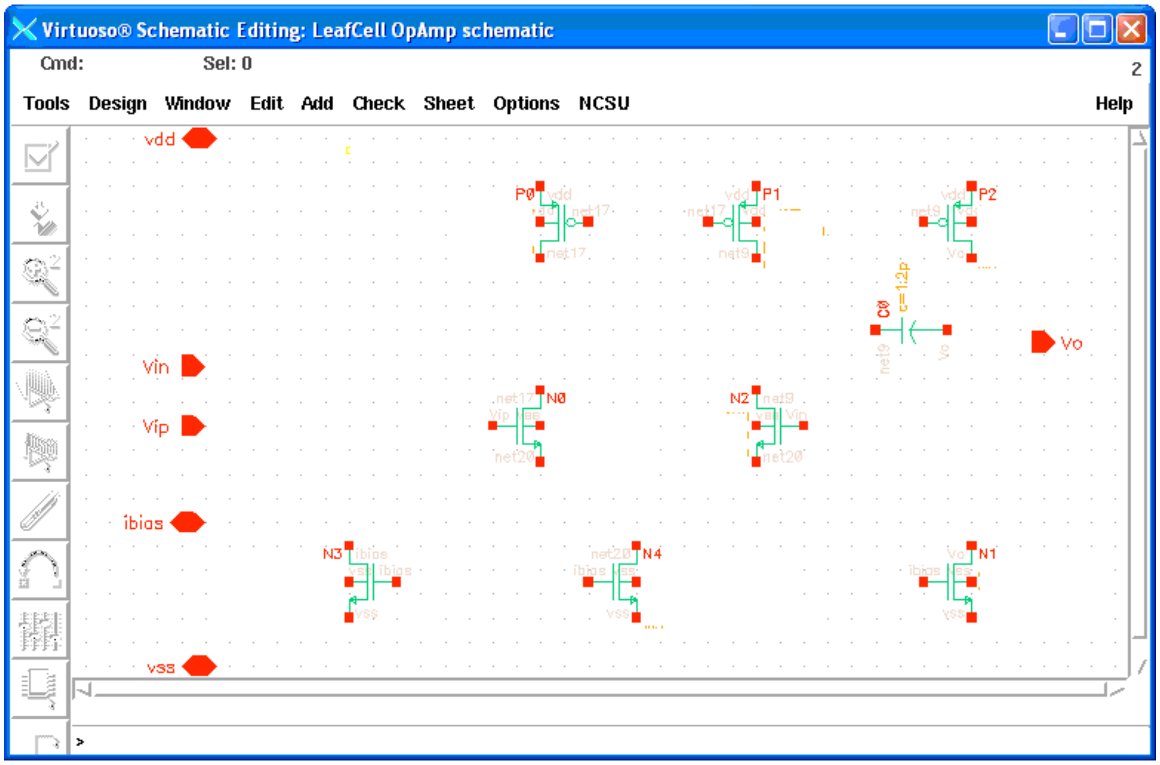


Figure 136: Adding and placing the instances.

4. Adding wires(Figure 137):

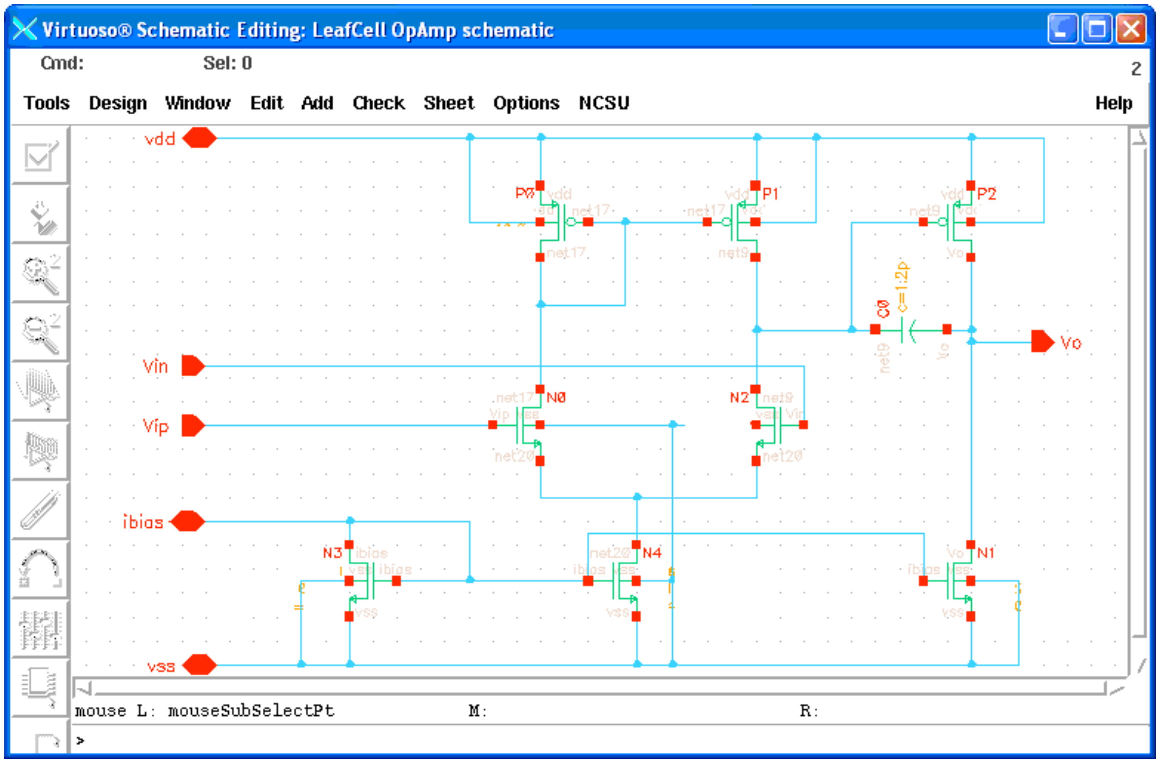
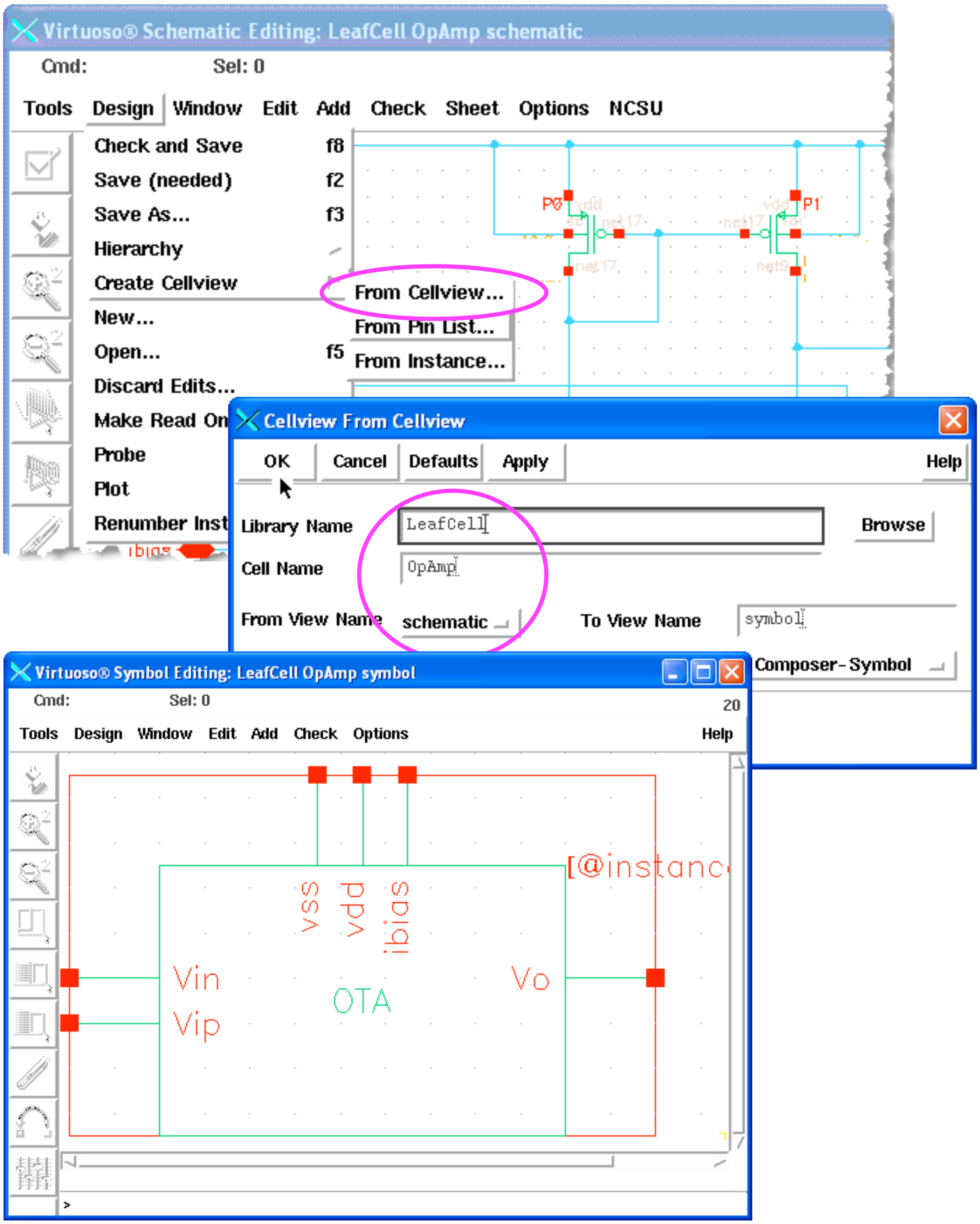


Figure 137: Wiring up the OTA.

Creating a symbol view:

5. Creating the symbol view from the schematic cell view (Figure 138):



Library Name: LeafCell

Cell Name: OpAmp

From View Name: schematic

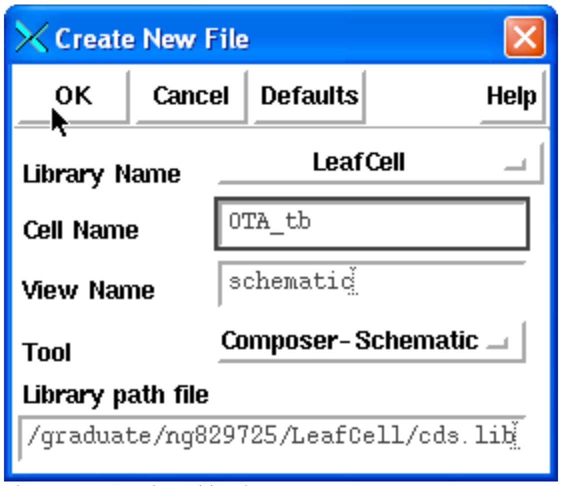
To View Name: symbol

Composer-Symbol

Figure 138: Creating the symbol view from schematic cell view.

Creating a test bench:

7. Create a new cell view for the test bench, named OTA_tb (Figure 139):

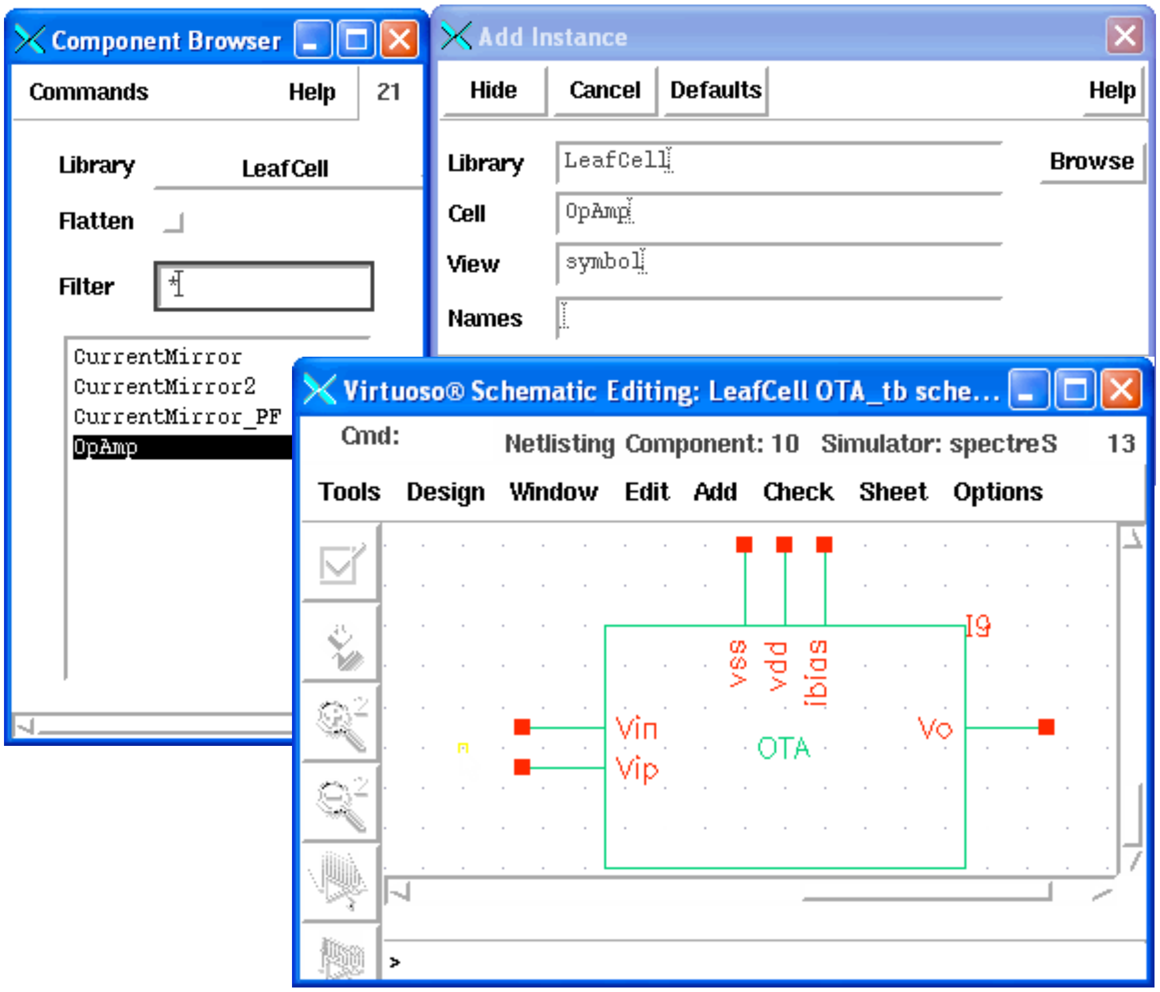

Create New File

OK Cancel Defaults Help

Library Name	LeafCell
Cell Name	OTA_tb
View Name	schematic
Tool	Composer-Schematic
Library path file	/graduate/ng829725/LeafCell/cds.lib

Figure 139: Creating OTA_tb.

8. Inserting the OpAmp into the testbench (Figure 140)


Component Browser

Commands Help 21

Library	LeafCell
Flatten	
Filter	*

CurrentMirror
CurrentMirror2
CurrentMirror_PF
OpAmp

Add Instance

Hide Cancel Defaults Help

Library	LeafCell
Cell	OpAmp
View	symbol
Names	

Virtuoso® Schematic Editing: LeafCell OTA_tb sche...

Cmd: Netlisting Component: 10 Simulator: spectreS 13

Tools Design Window Edit Add Check Sheet Options

Figure 140: Adding the OpAmp and stamping it down into the OTA test bench.

9. Adding CL, Ibias, and other power supplies to complete the test bench (Figure 141)

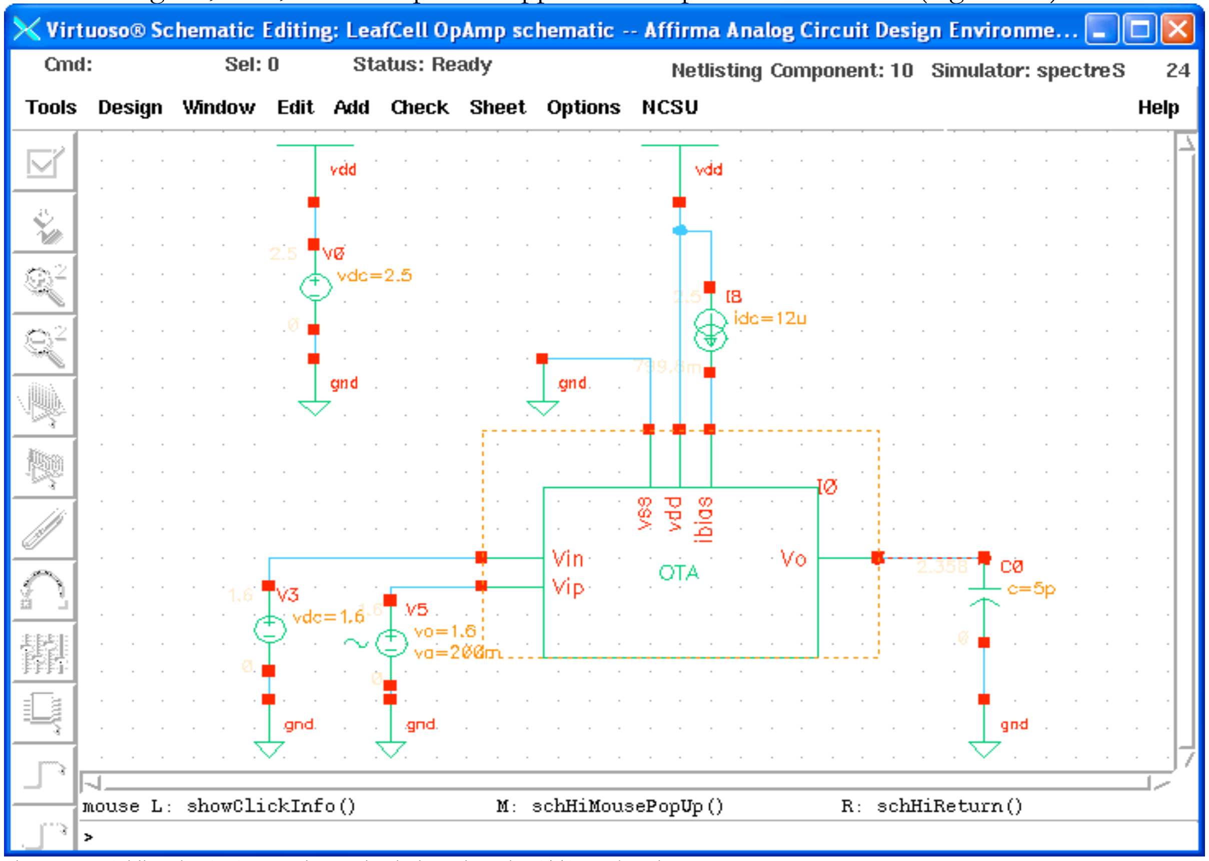
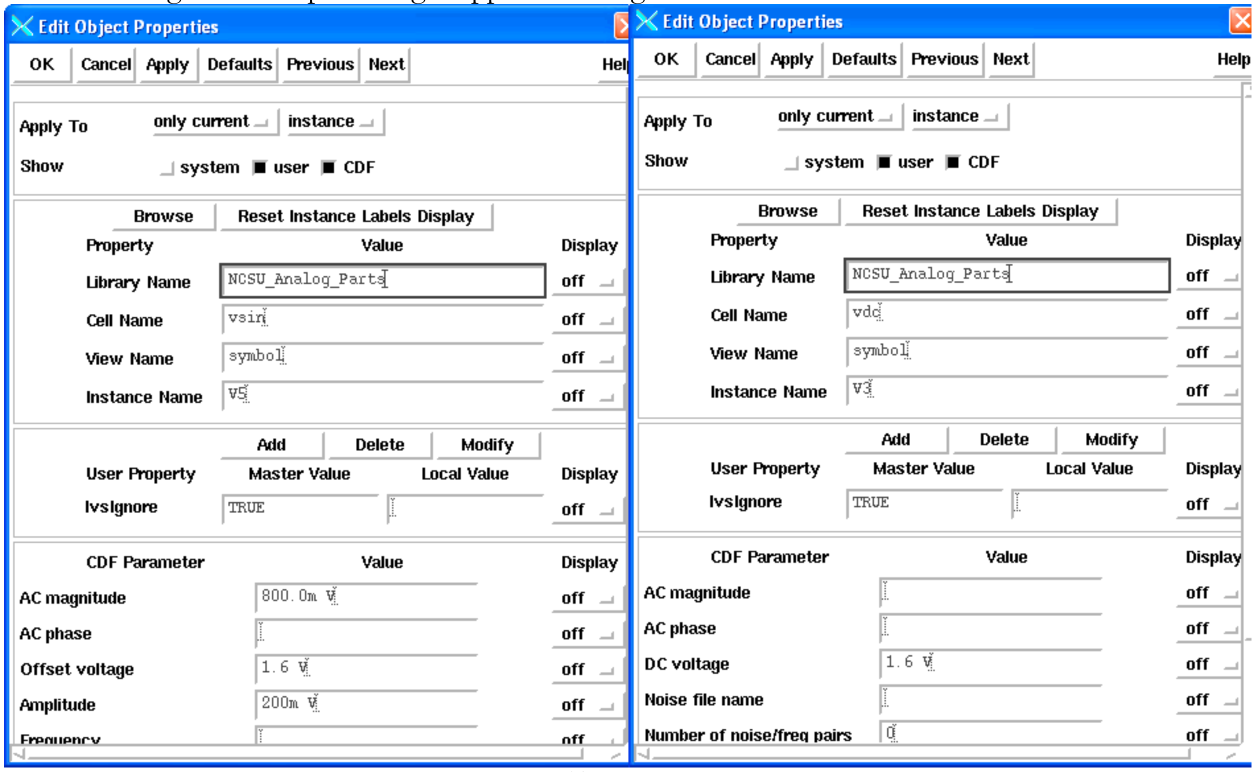


Figure 141: Adding the OpAmp and stamping it down into the OTA test bench.

10. Setting the two input voltage supplies as in Figure 142.



Apply To: only current, instance
Show: system, user ■, CDF ■

Property	Value	Display
Library Name	NCSU_Analog_Parts	off
Cell Name	vsin	off
View Name	symbol	off
Instance Name	V5	off

User Property	Master Value	Local Value	Display
lvsIgnore	TRUE		off

CDF Parameter	Value	Display
AC magnitude	800.0m V	off
AC phase		off
Offset voltage	1.6 V	off
Amplitude	200m V	off
Frequency		off

Apply To: only current, instance
Show: system, user ■, CDF ■

Property	Value	Display
Library Name	NCSU_Analog_Parts	off
Cell Name	vdc	off
View Name	symbol	off
Instance Name	V3	off

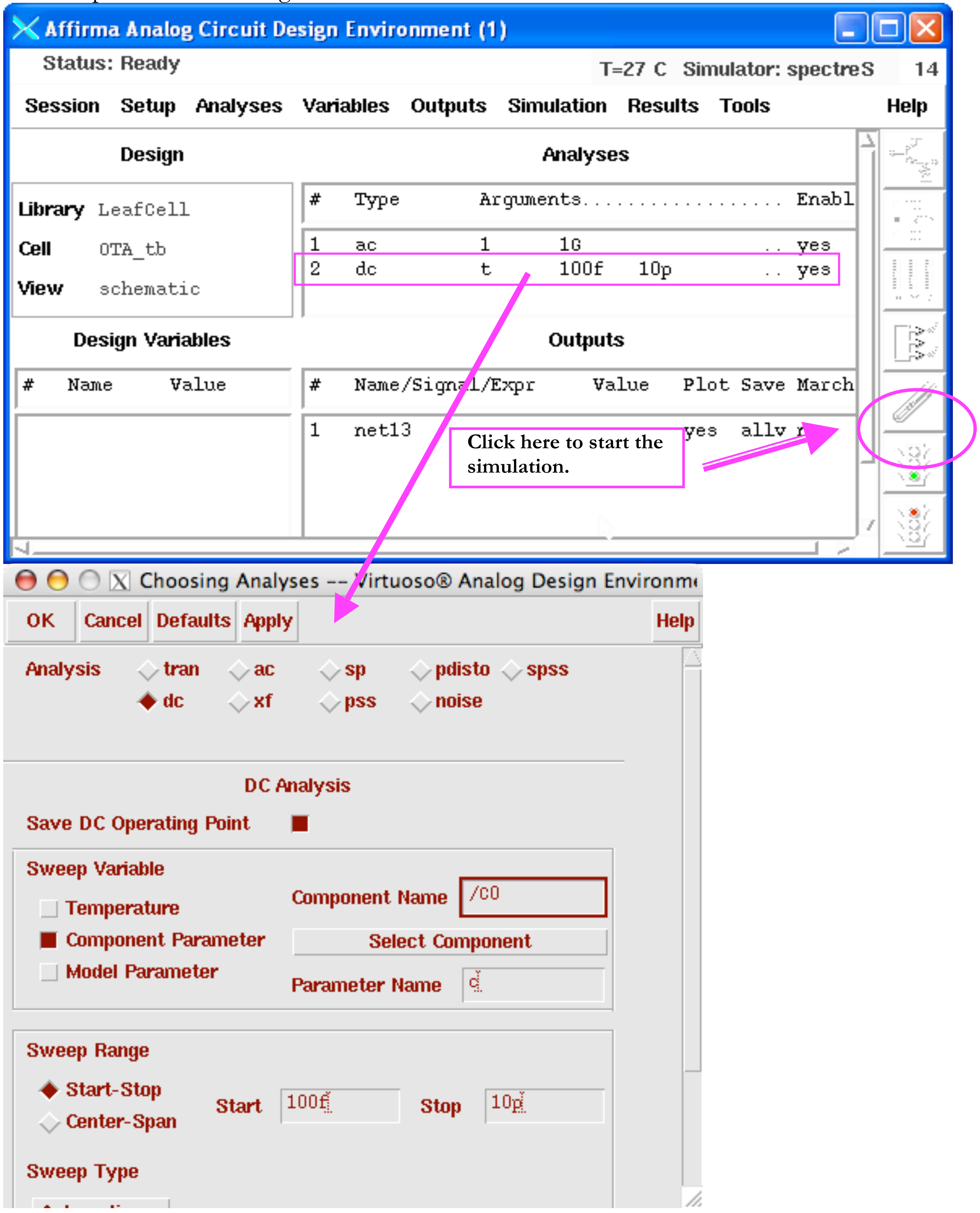
User Property	Master Value	Local Value	Display
lvsIgnore	TRUE		off

CDF Parameter	Value	Display
AC magnitude		off
AC phase		off
DC voltage	1.6 V	off
Noise file name		off
Number of noise/freq pairs	0	off

Figure 142: Setting up the Vsin and Vdc for the two inputs to the OTA.

Simulation in Spectre Spice using the Affirma environment:

11. From the *OTA_tb* schematic window, go to *Tools... Analog Environment* to open the form. Set up the form as in Figure 143.


Affirma Analog Circuit Design Environment (1)
Status: Ready
T=27 C Simulator: spectreS
14
Session Setup Analyses Variables Outputs Simulation Results Tools Help
Choosing Analyses -- Virtuoso® Analog Design Environme
OK Cancel Defaults Apply Help
Click here to start the simulation.

Design

Library	LeafCell
Cell	OTA_tb
View	schematic

Analyses

#	Type	Arguments.................	Enabl
1	ac	1 1G ..	yes
2	dc	t 100f 10p ..	yes

Design Variables

#	Name	Value

Outputs

#	Name/Signal/Expr	Value	Plot	Save	March
1	net13		yes	allv	r

Analysis: tran, ac, sp, pdisto, spss, [✓ dc], xf, pss, noise

DC Analysis

Save DC Operating Point ■

Sweep Variable

- ☐ Temperature
- ■ Component Parameter
- ☐ Model Parameter

Component Name: /C0

Select Component

Parameter Name: c

Sweep Range

- ◆ Start-Stop
- ◇ Center-Span

Start: 100f   Stop: 10p

Sweep Type

Figure 143: setting up the Affirma Form for simulation.

We need to set up the AC analysis for Bode Plots, and the DC analysis to look at the operating points of the transistors.

The simulation waveform will appear after the simulation is done (Figure 144).

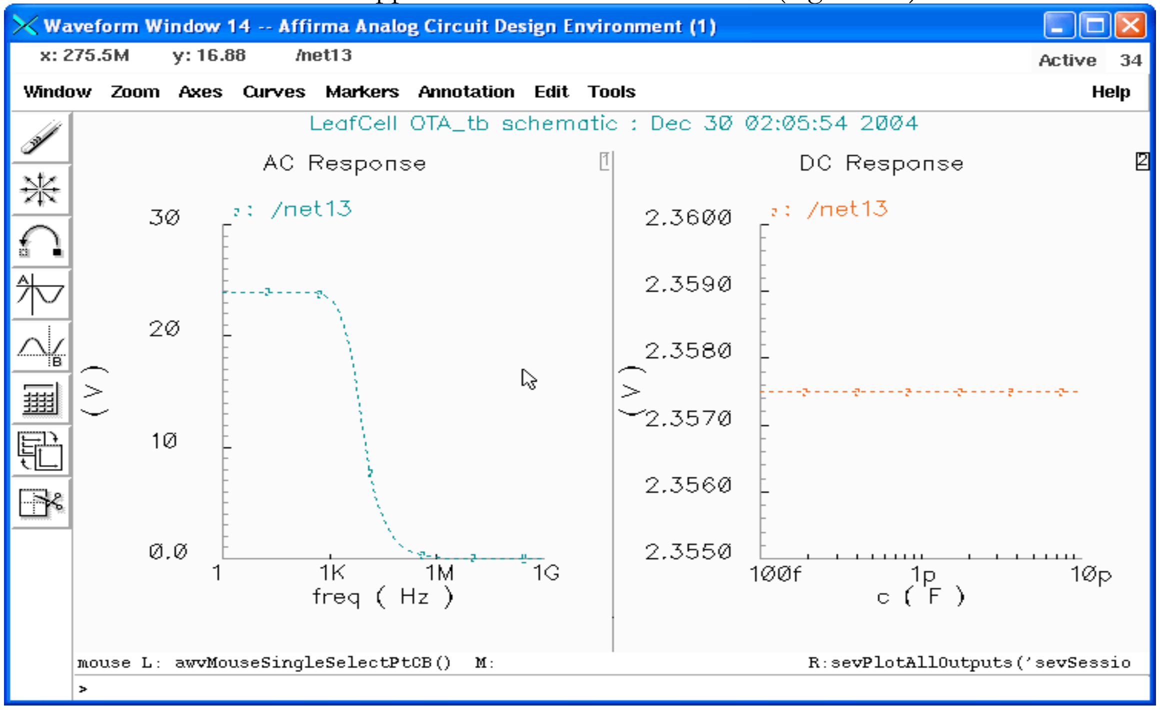
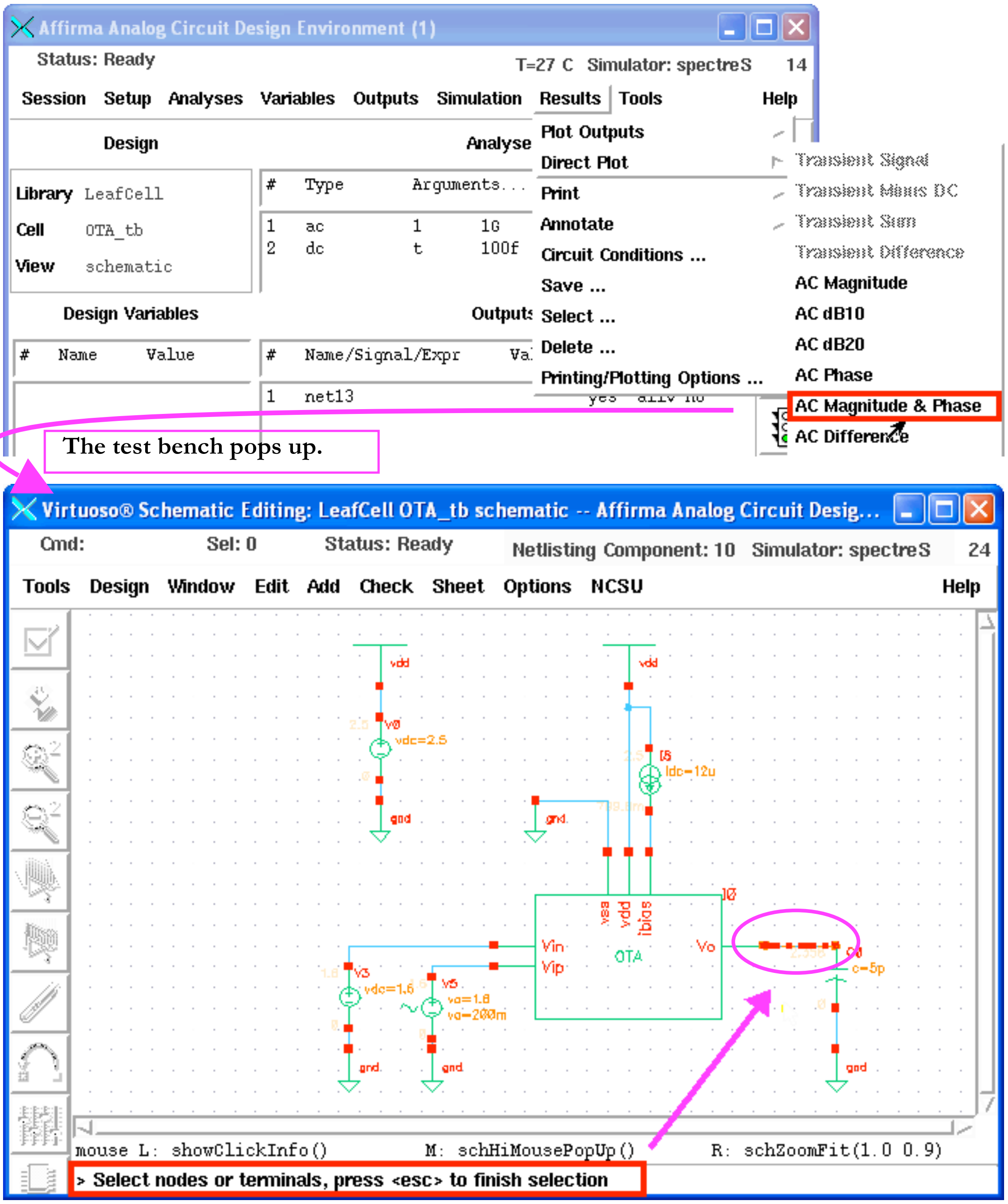


Figure 144: setting up the Affirma Form for simulation.

12. To plug the Bode Plot, go back to the *Affirma Analog Environment* form, click at *Results…Direct Plot…AC Magnitude & Phase* (Figure 145).


Affirma Analog Circuit Design Environment (1)
Status: Ready
T=27 C Simulator: spectreS 14
Session Setup Analyses Variables Outputs Simulation Results Tools Help
Design
Analyse
Library LeafCell
Cell OTA_tb
View schematic
Design Variables
Outputs
Plot Outputs
Direct Plot
Print
Annotate
Circuit Conditions ...
Save ...
Select ...
Delete ...
Printing/Plotting Options ...
Transient Signal
Transient Minus DC
Transient Sum
Transient Difference
AC Magnitude
AC dB10
AC dB20
AC Phase
AC Magnitude & Phase
AC Difference
Virtuoso® Schematic Editing: LeafCell OTA_tb schematic -- Affirma Analog Circuit Desig...
Cmd: Sel: 0 Status: Ready Netlisting Component: 10 Simulator: spectreS 24
Tools Design Window Edit Add Check Sheet Options NCSU Help
vdd
V0
vdc=2.5
gnd
I5
idc=12u
OTA
Vin
Vip
Vo
vss
ibias
I0
C0
c=5p
V3
vdc=1.6
V5
va=1.8
vd=200m

#	Type	Arguments...	
1	ac	1	1G
2	dc	t	100f

#	Name	Value

#	Name/Signal/Expr	Va
1	net13	

The test bench pops up.

```
mouse L: showClickInfo()        M: schHiMousePopUp()        R: schZoomFit(1.0 0.9)
> Select nodes or terminals, press <esc> to finish selection
```

Figure 145: Plotting the Bode Plot for the output terminal.

13. Select the output terminals, then press *Esc*, the result waveform will show (Figure 146.) The Bode Plot is shown in *section 2* in the *Waveform Window*. However, by using *Crosshair Marker A*, we find that the -3dB point is at ~5 KHz instead of 4MHz in our Spec. That is, the GB spec is not met.

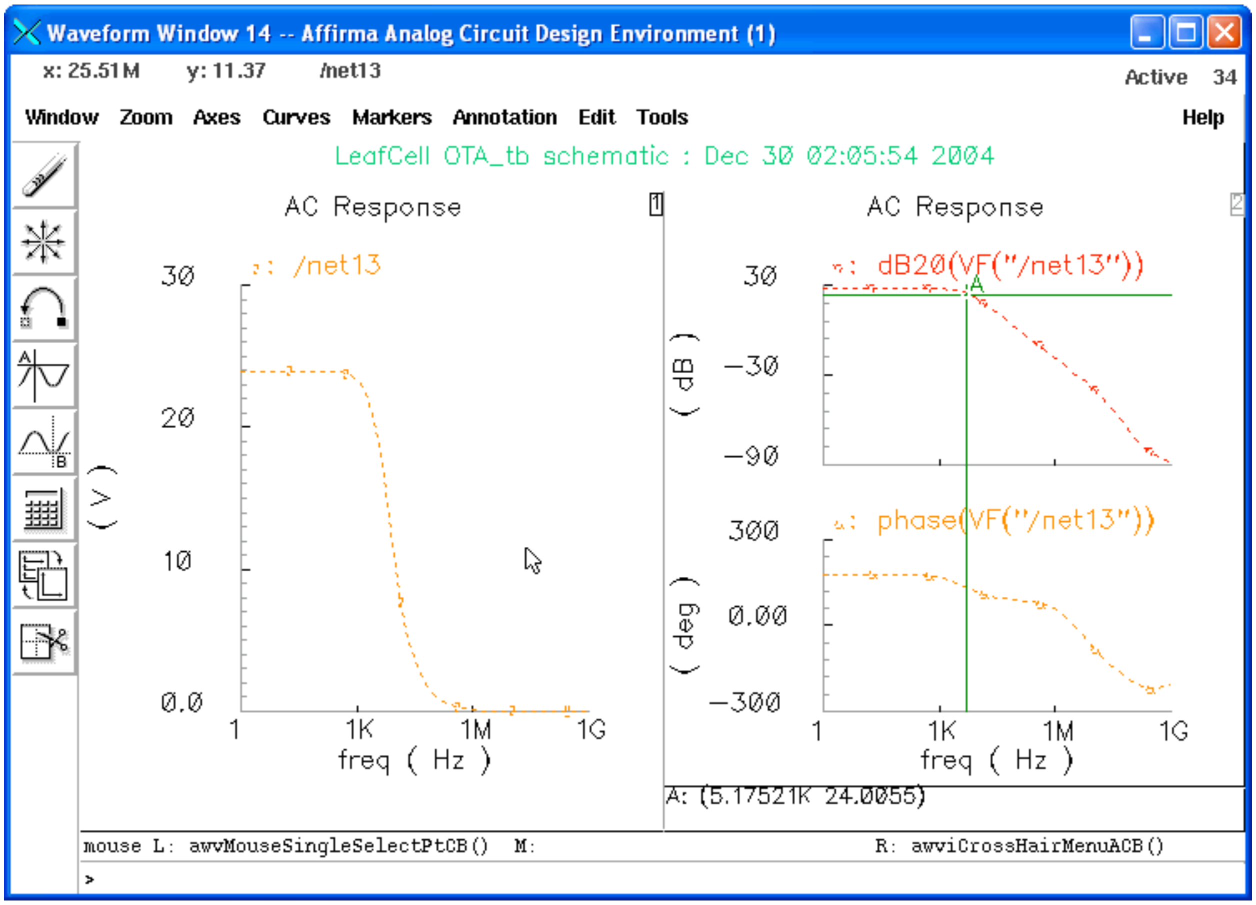
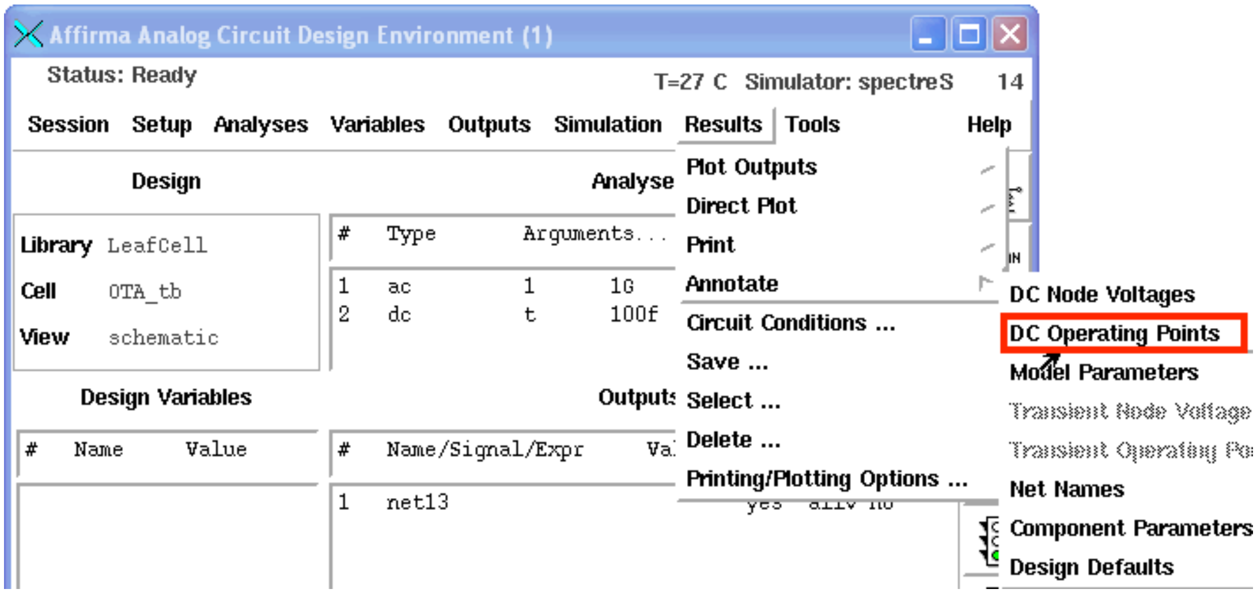

Waveform Window 14 -- Affirma Analog Circuit Design Environment (1)
x: 25.51M y: 11.37 /net13
Active 34
Window Zoom Axes Curves Markers Annotation Edit Tools Help
LeafCell OTA_tb schematic : Dec 30 02:05:54 2004
AC Response
/net13
dB20(VF("/net13"))
phase(VF("/net13"))
freq (Hz)
A: (5.17521K 24.0055)
mouse L: awvMouseSingleSelectPtCB() M: R: awviCrossHairMenuACB()

Figure 146: Bode Plot is shown in section 2 in the waveform.

14. In order to find out if the transistors are working, we can check their operating points. In the *Affirma Analog Environment* form, click at *Results…Annotate…DC Operating Points…*,(Figure 147).



Design

Library	LeafCell
Cell	OTA_tb
View	schematic

Analyses

#	Type	Arguments...	
1	ac	1	1G
2	dc	t	100f

Design Variables

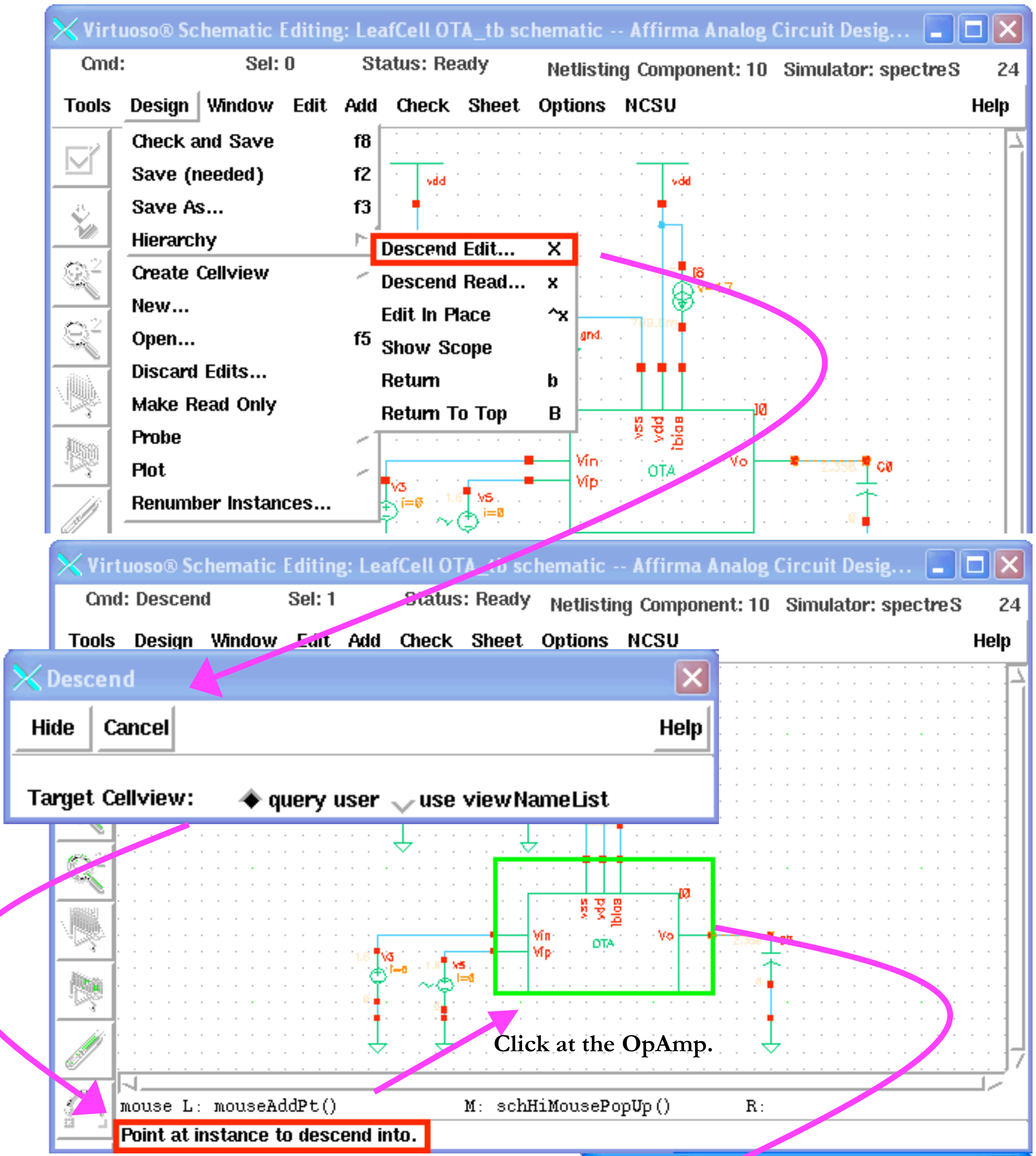
#	Name	Value

Outputs

#	Name/Signal/Expr	Va
1	net13	

Figure 147: Setting up to show the DC operating points for the transistors.

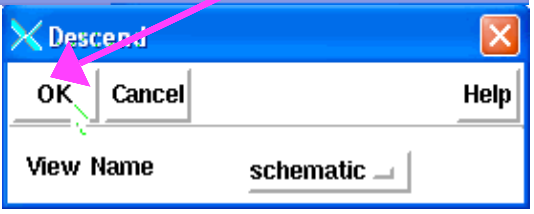
15. The OTA_tb schematic will then pop up. Click at *Design…Hierarchy…Descend Edit…* (Figure 148). The *Descend Window* will then pop up, asking you to point at the instance that you want to descend edit (Figure 148).



mouse L: mouseAddPt()   M: schHiMousePopUp()   R:

Point at instance to descend into.

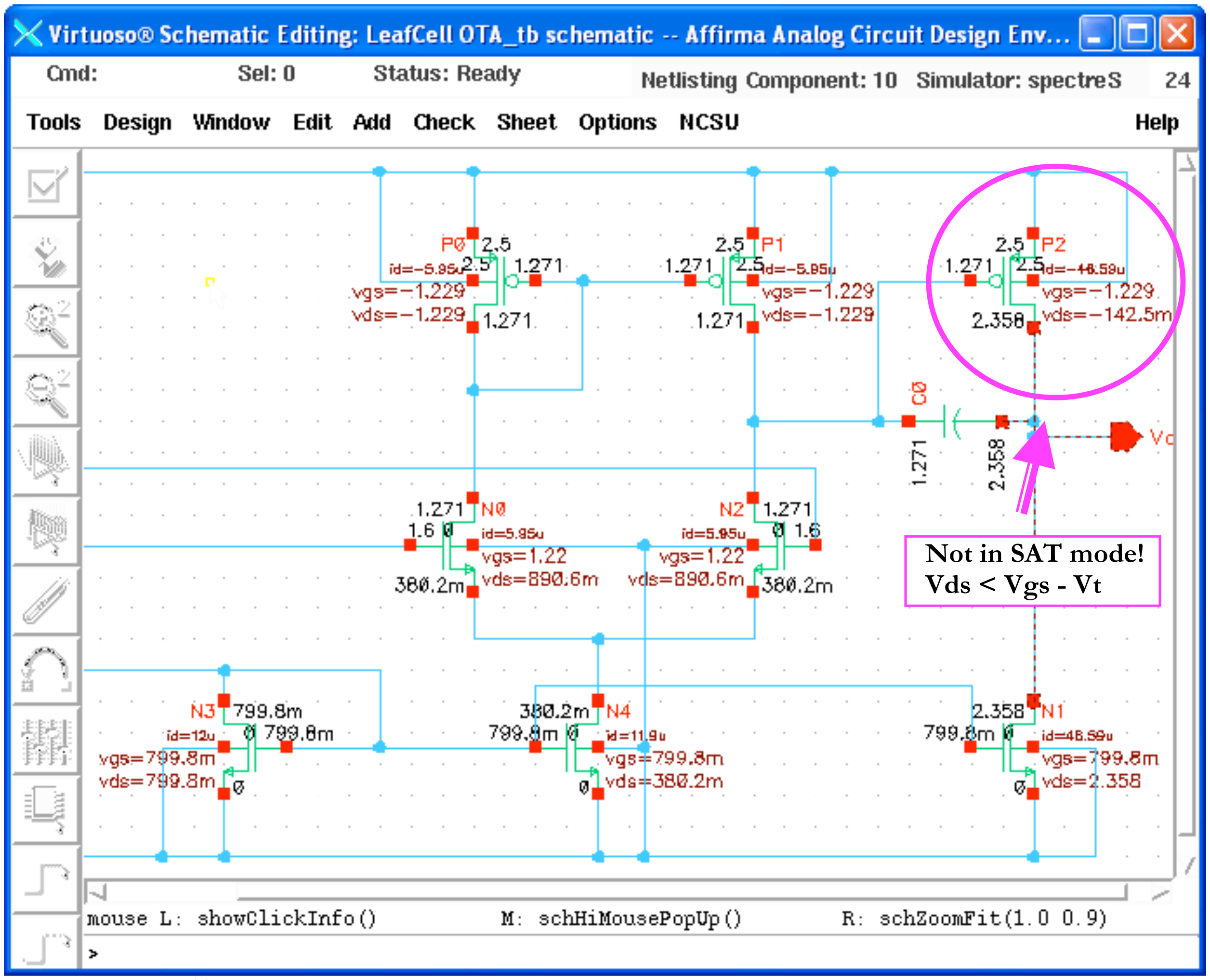
Figure 148: Descend into the OpAmp for transistor operating points.



View Name: schematic

16. Click at the *OTA*, then the OTA will be highlighted, and another *Descend window* (Figure 151) pops up asking you for the correct view. Make sure it is the schematic view and

then click *OK*. Go back to the Affirma Analog Environment form, click at Results…Direct Plot…AC Magnitude & Phase one more time (Figure 147). The schematic view of the OTA will then be shown with all the operating points for the transistors (Figure 149).



Virtuoso® Schematic Editing: LeafCell OTA_tb schematic -- Affirma Analog Circuit Design Env...
Cmd:
Sel: 0
Status: Ready
Netlisting Component: 10
Simulator: spectreS
24
Tools Design Window Edit Add Check Sheet Options NCSU
Help
P0 2.5 id=-5.95u 2.5 1.271 vgs=-1.229 vds=-1.229 1.271
P1 2.5 1.271 2.5 id=-5.95u vgs=-1.229 vds=-1.229 1.271
P2 2.5 1.271 2.5 id=-46.59u vgs=-1.229 vds=-142.5m 2.358
C0
1.271
2.358
Vo
N0 1.271 1.6 0 id=5.95u vgs=1.22 vds=890.6m 380.2m
N2 1.271 0 1.6 id=5.95u vgs=1.22 vds=890.6m 380.2m
Not in SAT mode!
Vds < Vgs - Vt
N3 799.8m 0 799.8m id=12u vgs=799.8m vds=799.8m 0
N4 380.2m 799.8m 0 id=11.9u vgs=799.8m vds=380.2m 0
N1 2.358 799.8m 0 id=46.59u vgs=799.8m vds=2.358 0

```
mouse L: showClickInfo()          M: schHiMousePopUp()          R: schZoomFit(1.0 0.9)
>
```

Figure 149: Descend OTA schematic view with transistor operating points.

By looking at the operating point, we see that the OpAmp we didn't choose the right Iref. You can easily correct it by choosing a much smaller Iref. (1u to 10uA works for me. How about yours?)

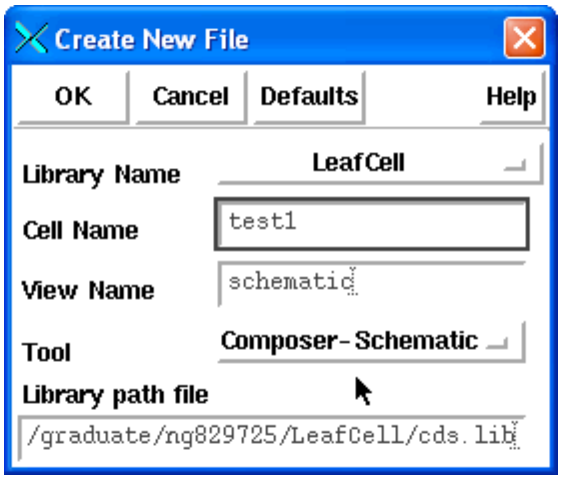
Appendix A: How to extract K'?

We need to go back to the very beginning. Start from the calculation again. Since we used the values for K'n and K'p from the *MOSIS PARAMETRIC TEST RESULTS* when we did the first calculation, we can start from extracting K'n and K'p ourselves to have more accurate K' values.

1. Create a test schematic called *test1* (Figure 150.)

2. Create the test schematic for NMOS as in Figure 151. *Check and Save* it.

3. Set up the Analog Environment as in Figure 152. Then run it. Simulation results are shown in Figure 153.



Library Name: LeafCell
Cell Name: test1
View Name: schematic
Tool: Composer-Schematic
Library path file: /graduate/ng829725/LeafCell/cds.lib

Figure 150: Creating the test1 schematic to extract k'n.

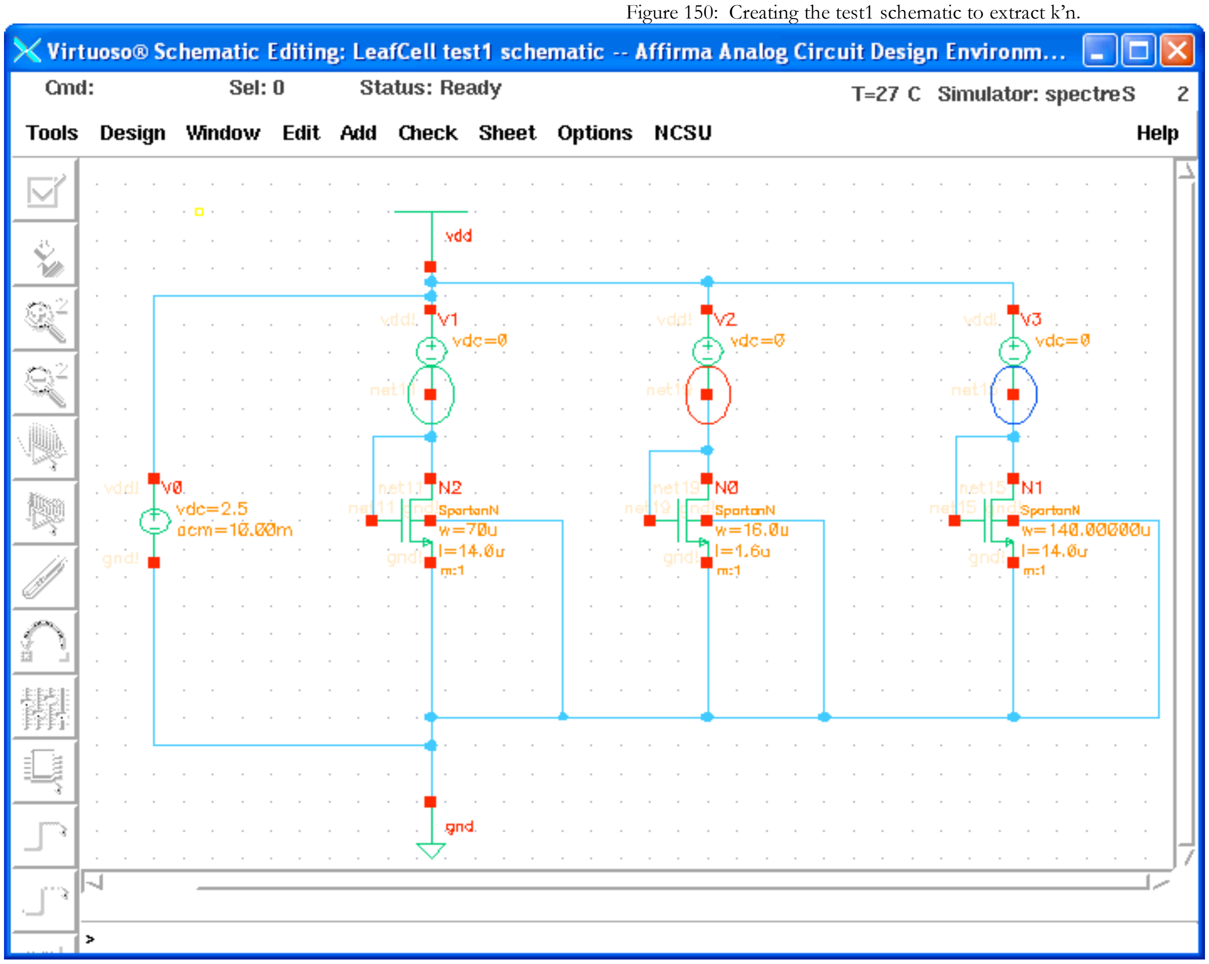
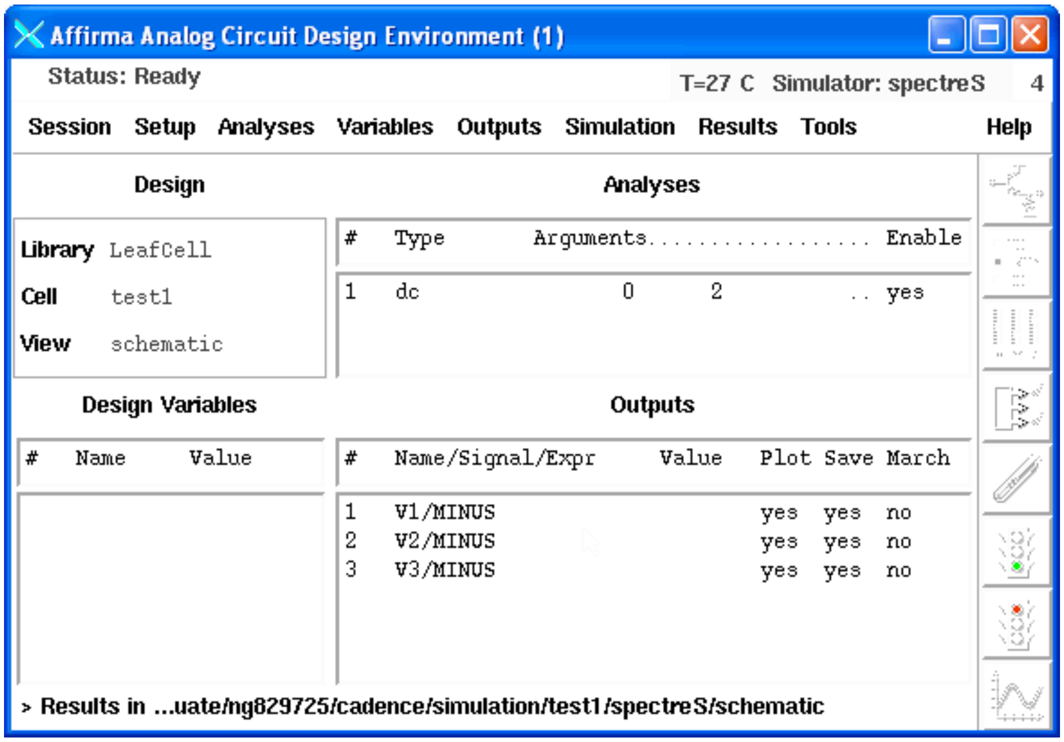


Figure 151: test1 schematic.



Affirma Analog Circuit Design Environment (1)

Status: Ready T=27 C Simulator: spectreS 4

Session Setup Analyses Variables Outputs Simulation Results Tools Help

Design

Library	LeafCell
Cell	test1
View	schematic

Analyses

#	Type	Arguments.................		Enable
1	dc	0	2 ..	yes

Design Variables

#	Name	Value

Outputs

#	Name/Signal/Expr	Value	Plot	Save	March
1	V1/MINUS		yes	yes	no
2	V2/MINUS		yes	yes	no
3	V3/MINUS		yes	yes	no

> Results in ...uate/ng829725/cadence/simulation/test1/spectreS/schematic

Figure 152: Setting up for simulation for test1.

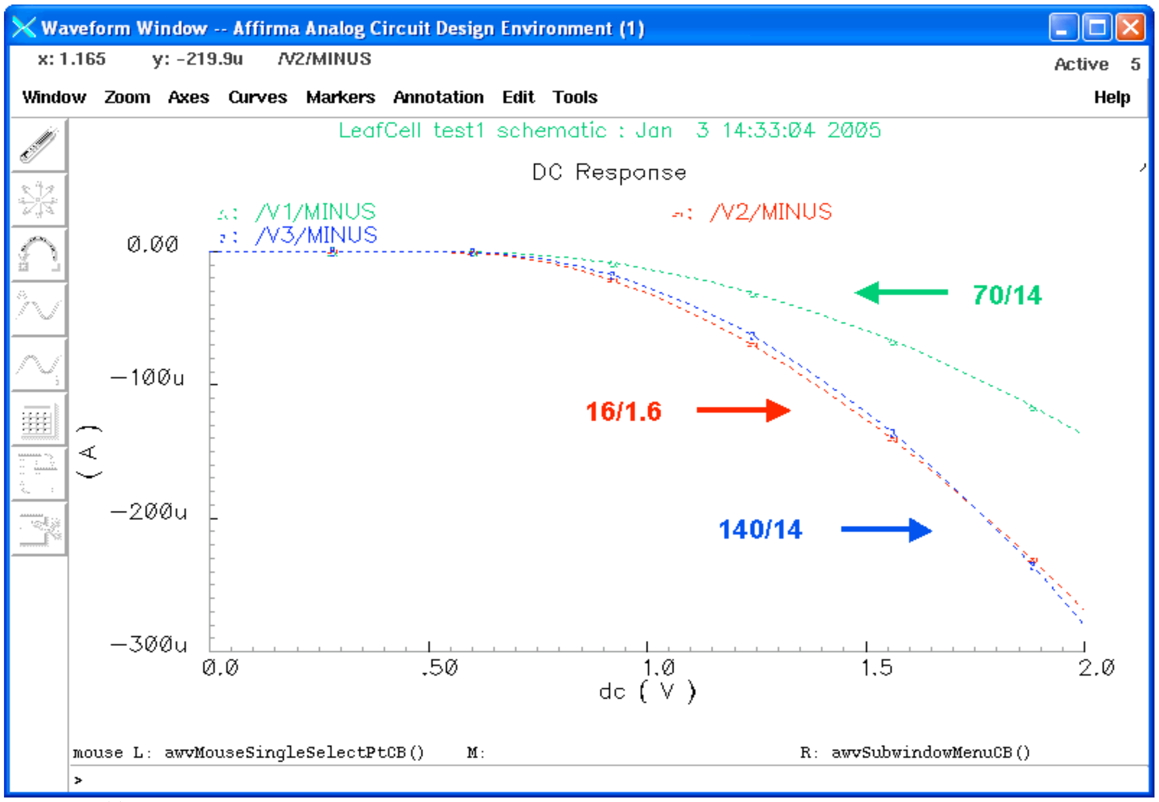


Figure 153: Test1 simulation waveform.

4. Goto *Markers…Vertical Marker…*; Create markers as in Figure 154; Click at *Display Intercept Data.* Data form will show up as in Figure 155.

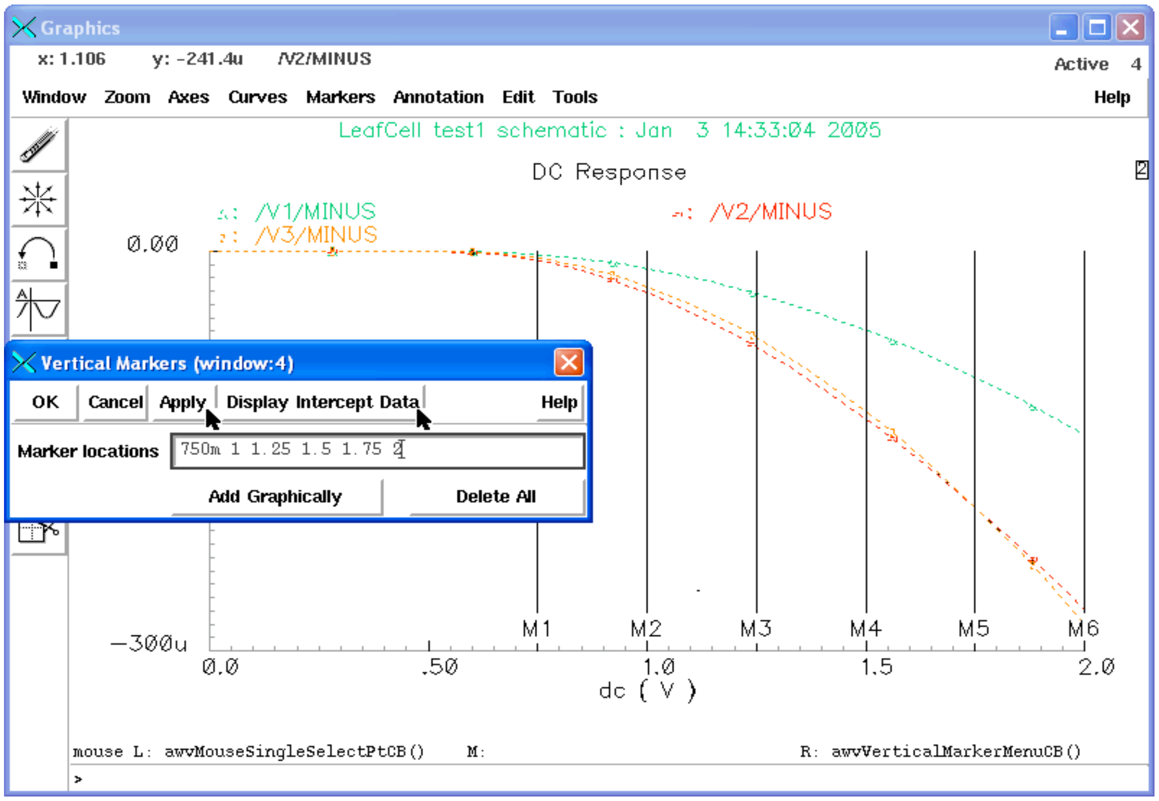
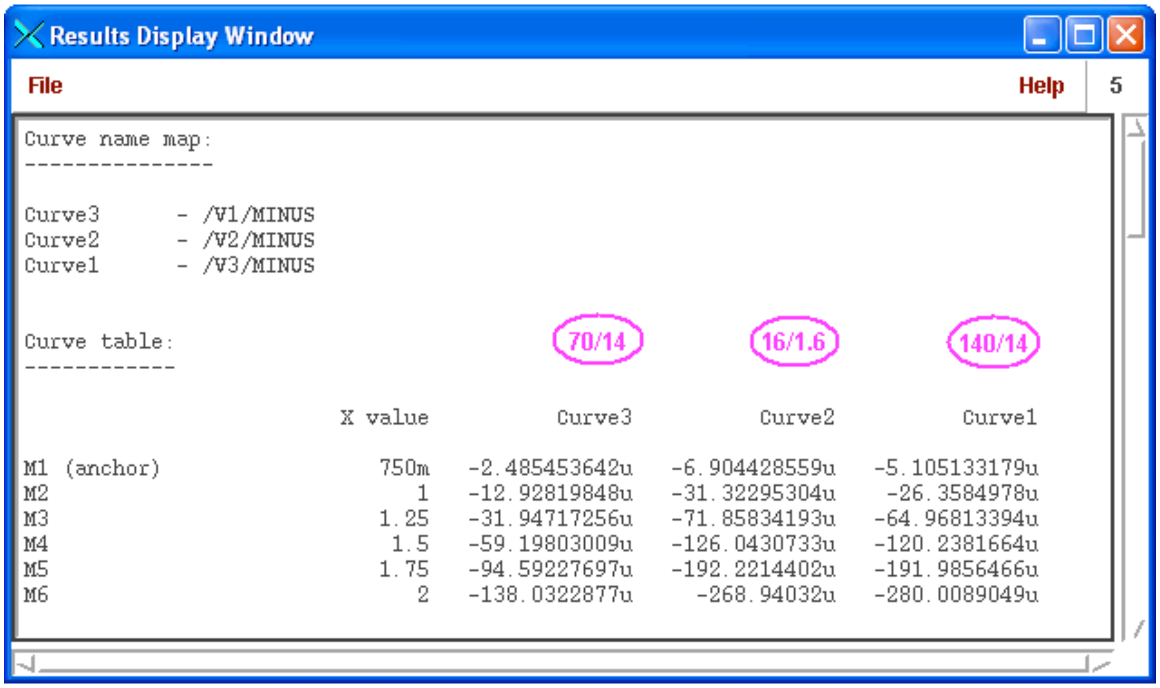

Graphics
x: 1.106 y: -241.4u /V2/MINUS
Active 4
Window Zoom Axes Curves Markers Annotation Edit Tools Help
LeafCell test1 schematic : Jan 3 14:33:04 2005
DC Response
/V1/MINUS
/V3/MINUS
/V2/MINUS
0.00
-300u
0.0
.50
1.0
1.5
2.0
dc (V)
M1
M2
M3
M4
M5
M6
Vertical Markers (window:4)
OK
Cancel
Apply
Display Intercept Data
Help
Marker locations
750m 1 1.25 1.5 1.75 2
Add Graphically
Delete All
mouse L: awvMouseSingleSelectPtCB() M: R: awvVerticalMarkerMenuCB()

Figure 154: Creating vertical markers for intercepting datas.



```
Curve name map:
---------------

Curve3      - /V1/MINUS
Curve2      - /V2/MINUS
Curve1      - /V3/MINUS


Curve table:
------------

                       X value        Curve3         Curve2         Curve1

M1  (anchor)              750m   -2.485453642u  -6.904428559u  -5.105133179u
M2                           1   -12.92819848u  -31.32295304u   -26.3584978u
M3                        1.25   -31.94717256u  -71.85834193u  -64.96813394u
M4                         1.5   -59.19803009u  -126.0430733u  -120.2381664u
M5                        1.75   -94.59227697u  -192.2214402u  -191.9856466u
M6                           2   -138.0322877u    -268.94032u  -280.0089049u
```

Figure 155: Creating vertical markers for intercepting datas.

5. From the data charts, you will be able to find out the K' values you need.