

## [Cadence Design System Tutorials from CMOSedu.com](#) ([Return](#))

### Tutorial 1 - Layout and simulation of a resistive voltage divider

This tutorial will introduce you to Cadence 6.1 for chip design, layout, and simulation. To demonstrate the operation of Cadence we'll set it up for

use with [ON's C5](#) process (formerly AMI's [C5 process](#)) and fabrication through [MOSIS](#).

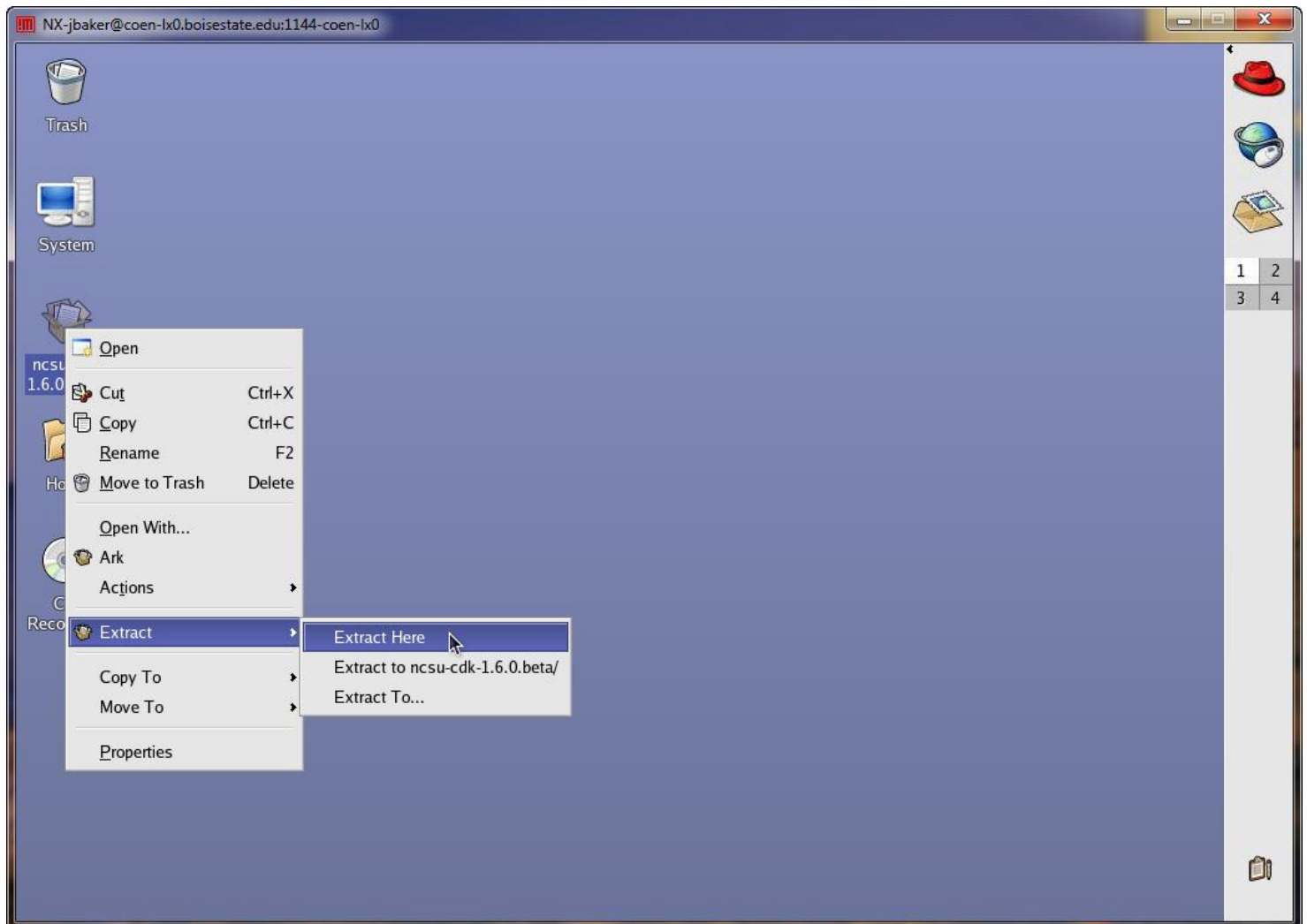
This tutorial uses the MOSIS scalable CMOS (SCMOS) [submicron design rules](#).

This tutorial uses a remote desktop to connect to the server.

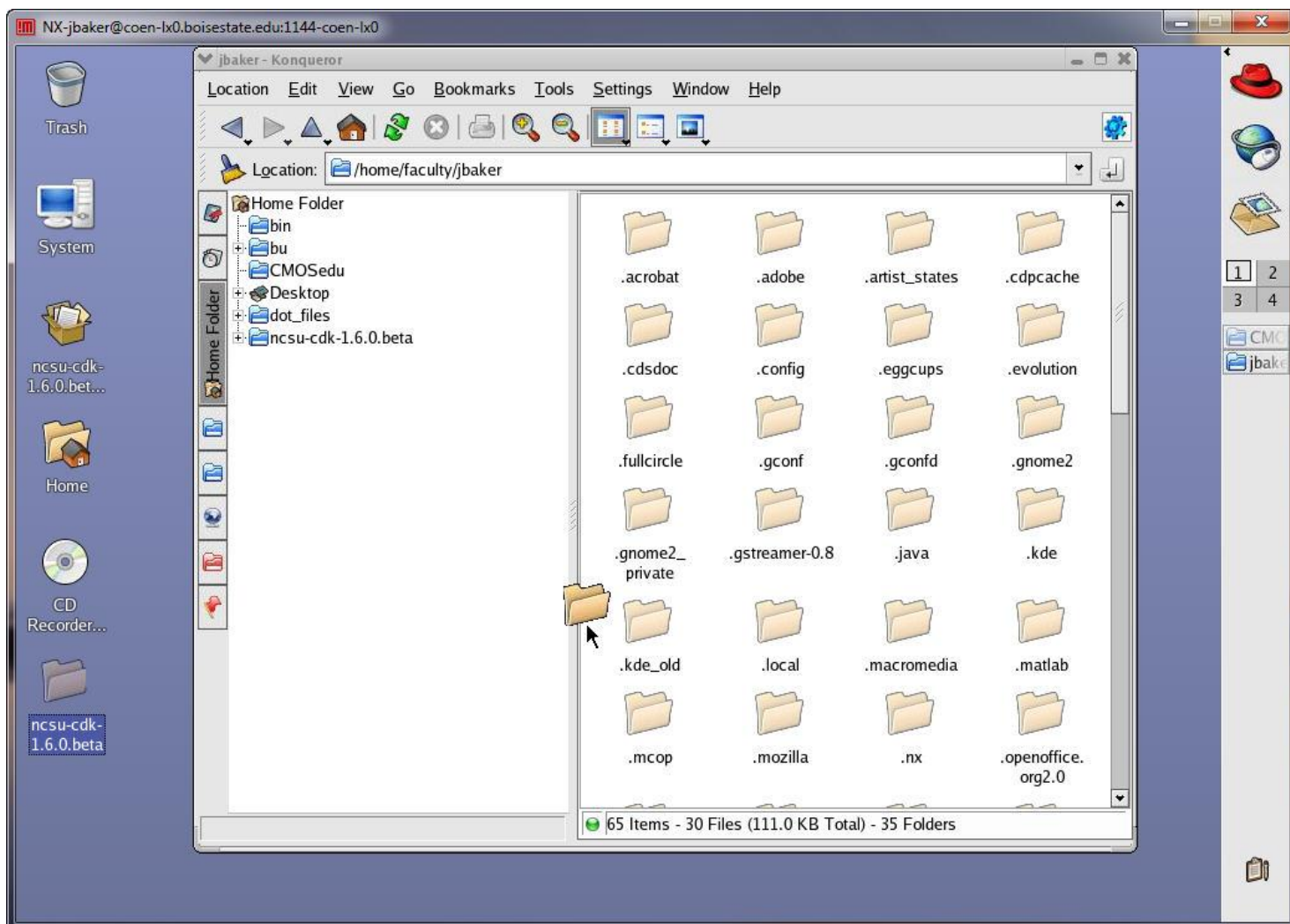
Here are links for help installing a [remote desktop](#) and Xterm ([written](#) and [video](#)).

The operation of Cadence isn't dependent on which method is used to log into the server.

To begin, download the NCSU Cadence Design Kit (CDK) version 1.6.0 beta from [http://www.eda.ncsu.edu/wiki/NCSU\\_CDK](http://www.eda.ncsu.edu/wiki/NCSU_CDK) to your desktop and extract the download's contents by right+clicking on the icon as seen below.



Next move the extracted folder (**not** the downloaded tar.gz archive), ncsu-cdk-1.6.0.beta, to your home folder (directory), \$HOME (my home directory is /home/faculty/jbaker, this path and \$HOME are equivalent), see below.



In your home directory open the `.bashrc` file (ensure that you can [view](#) hidden files and assuming that you are using the Bash shell) and add the following lines

```
export SPECTRE_DEFAULTS=-E
export CDS_Netlisting_Mode=Analog
export CDS_LOAD_ENV=CWDElseHome
export CDK_DIR=$HOME/ncsu-cdk-1.6.0.beta
```

When finished, in a terminal window in your home directory, type the command `source ~/.bashrc` (period, space, period bashrc) to re-source the `.bashrc` file.

Note that if Cadence IC5.141 isn't installed on your system then you likely won't need to add the last line seen above to your `.bashrc`.

Now, make a working directory in your home account called **CMOSedu**

Copy everything in the directory `$HOME/ncsu-cdk-1.6.0.beta/cdssetup` into `$HOME/CMOSedu`

In the working directory **CMOSedu**, rename `cdsinit`, `simrc`, and `cdsenv` to `.cdsinit`, `.simrc`, and `.cdsenv` (add a period)

In **CMOSedu** open the file `cds.lib` and add the following lines to point to the built-in Cadence libraries. Note that the path seen below may be different for your installation of Cadence so **verify** it's correct before adding the lines.

```
DEFINE analogLib /usr/cadence/IC615/tools.lnx86/dfII/etc/cdslib/artist/analogLib
DEFINE functional /usr/cadence/IC615/tools.lnx86/dfII/etc/cdslib/artist/functional
DEFINE sbaLib /usr/cadence/IC615/tools.lnx86/dfII/etc/cdslib/artist/sbaLib
```

We will use Spectre (Cadence's spice) for the simulations in these tutorials so add

`envSetVal("asimenv.startup" "simulator" 'string "spectre")` to the bottom of your `.cdsinit` file in your home directory (later after you've started and exited Cadence).

This makes Spectre the default simulator.

Additional helpful hints for modifying the `.cdsinit` are found [here](#).

Finally, in the directory `$HOME/ncsu-cdk-1.6.0.beta/lib/NCSU_TechLib_ami06` delete the files `divaDRC.rul`, `divaEXT.rul`, and `divaLVS.rul`.

Save [diva\\_rul\\_files.zip](#) to your desktop.

Extract the files in this zip to your desktop.

Move the extracted files (`divaDRC.rul`, `divaEXT.rul`, and `divaLVS.rul`) into `$HOME/ncsu-cdk-1.6.0.beta/lib/NCSU_TechLib_ami06` in the deleted files' places.

The deleted files are locked and point to the wrong place (a bug in the beta version that will be fixed in the final release).

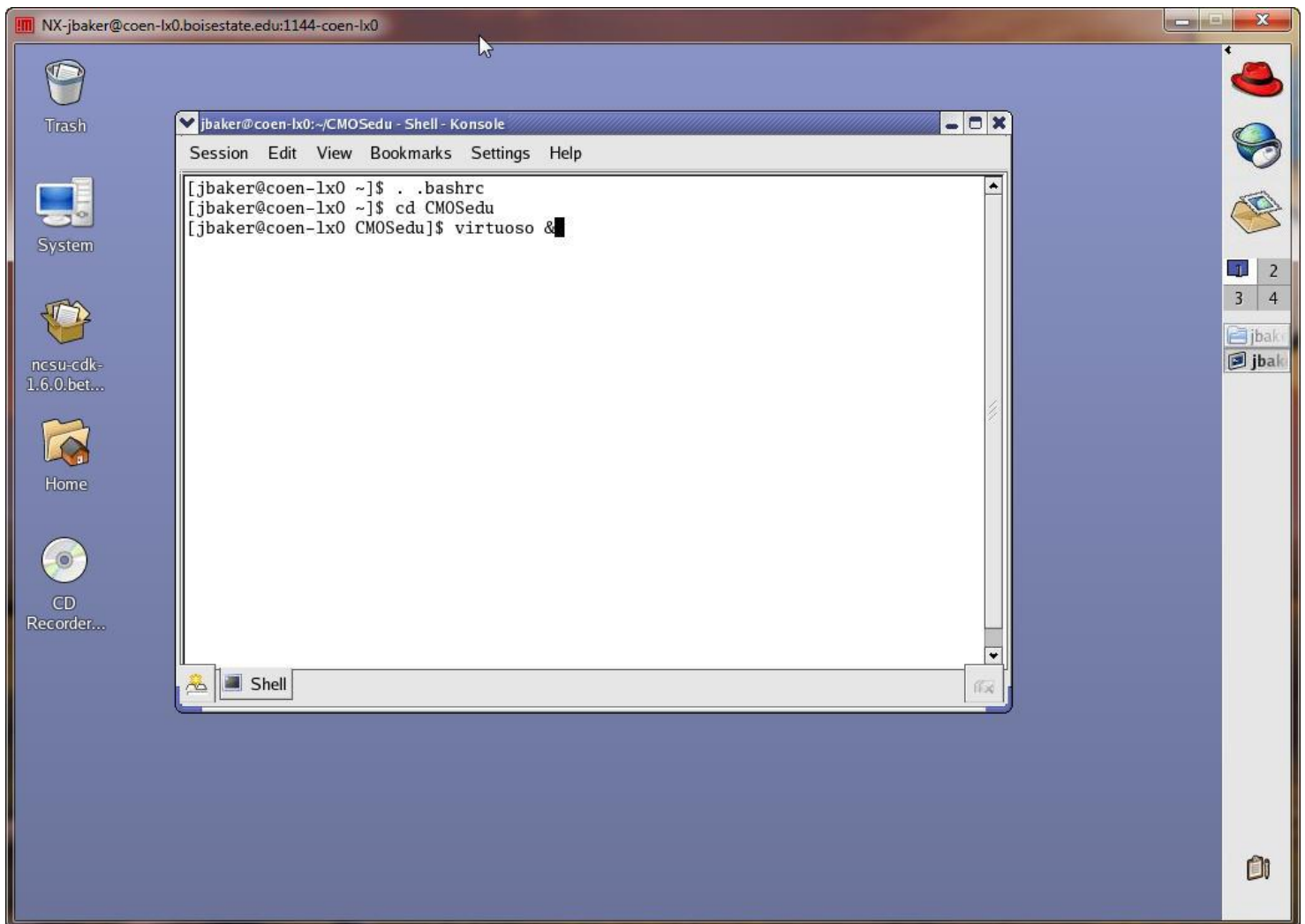
Note that these steps are important if you want to DRC, Extract, and LVS your layouts!

We are now ready to start Cadence and design a chip using [ON's C5](#) process and the [MOSIS](#) scalable CMOS design rules (lambda of 300 nm and a technology

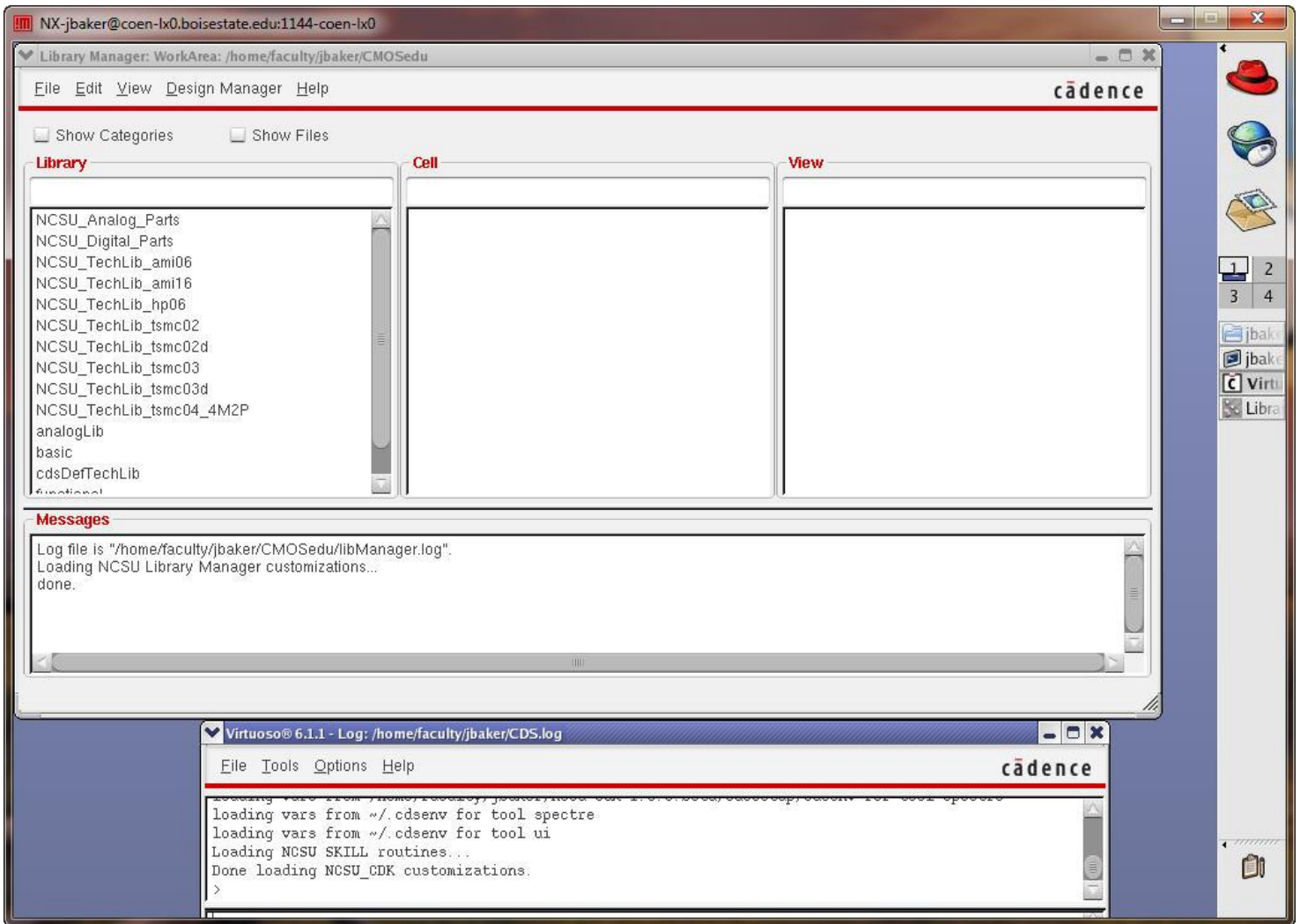
code of `SCMOS_SUBM`, see [here](#) for help on submitting a chip to MOSIS).

Open a terminal window and change directories to `CMOSedu` (the Unix command is `cd CMOSedu`). To start Cadence's Virtuoso editing tool

type, in the same terminal window, `virtuoso &` (adding `&` runs the process in the background allowing you to continue using the open terminal window) as seen below.

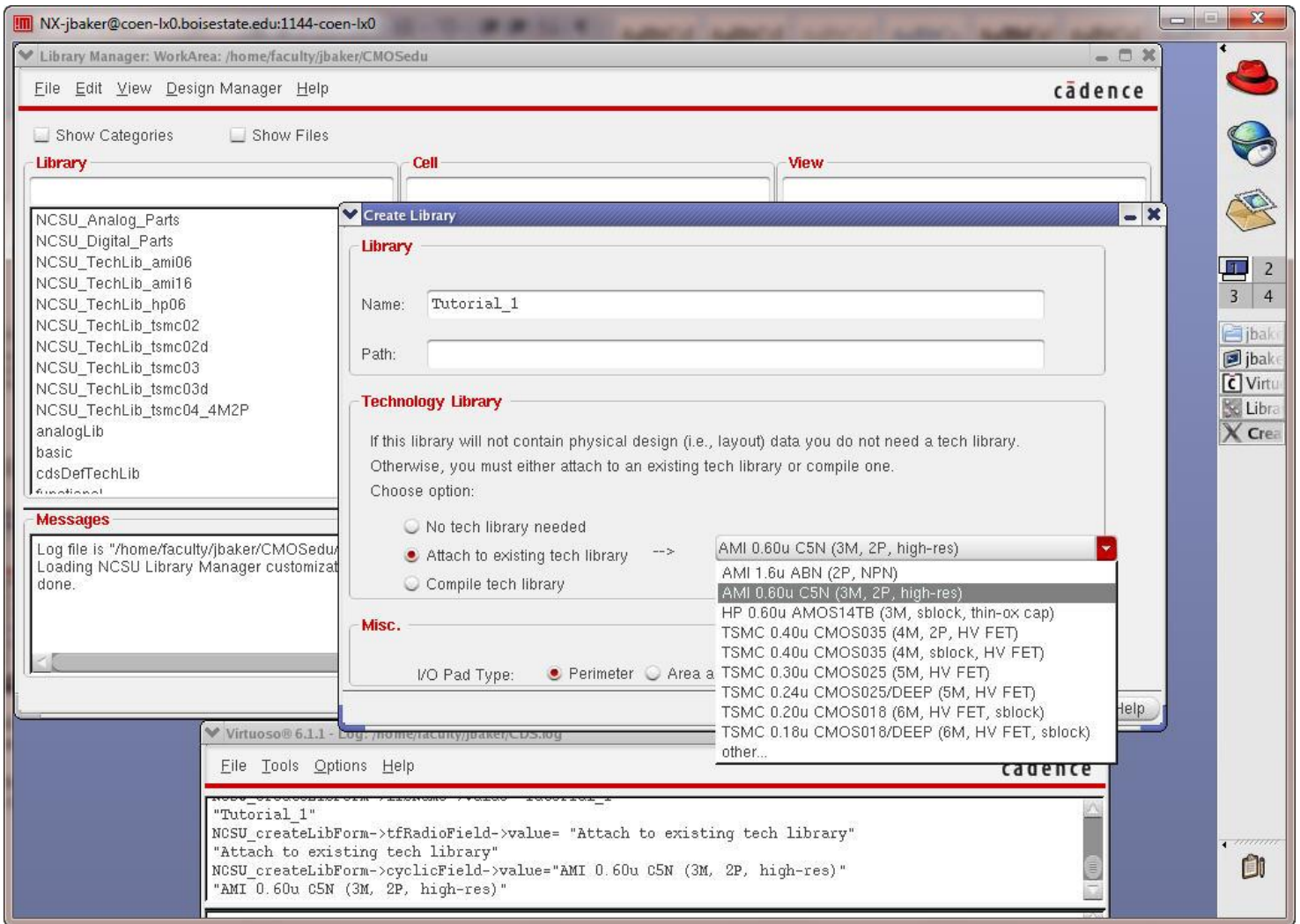


After starting Virtuoso, and re-sizing windows, the following should appear

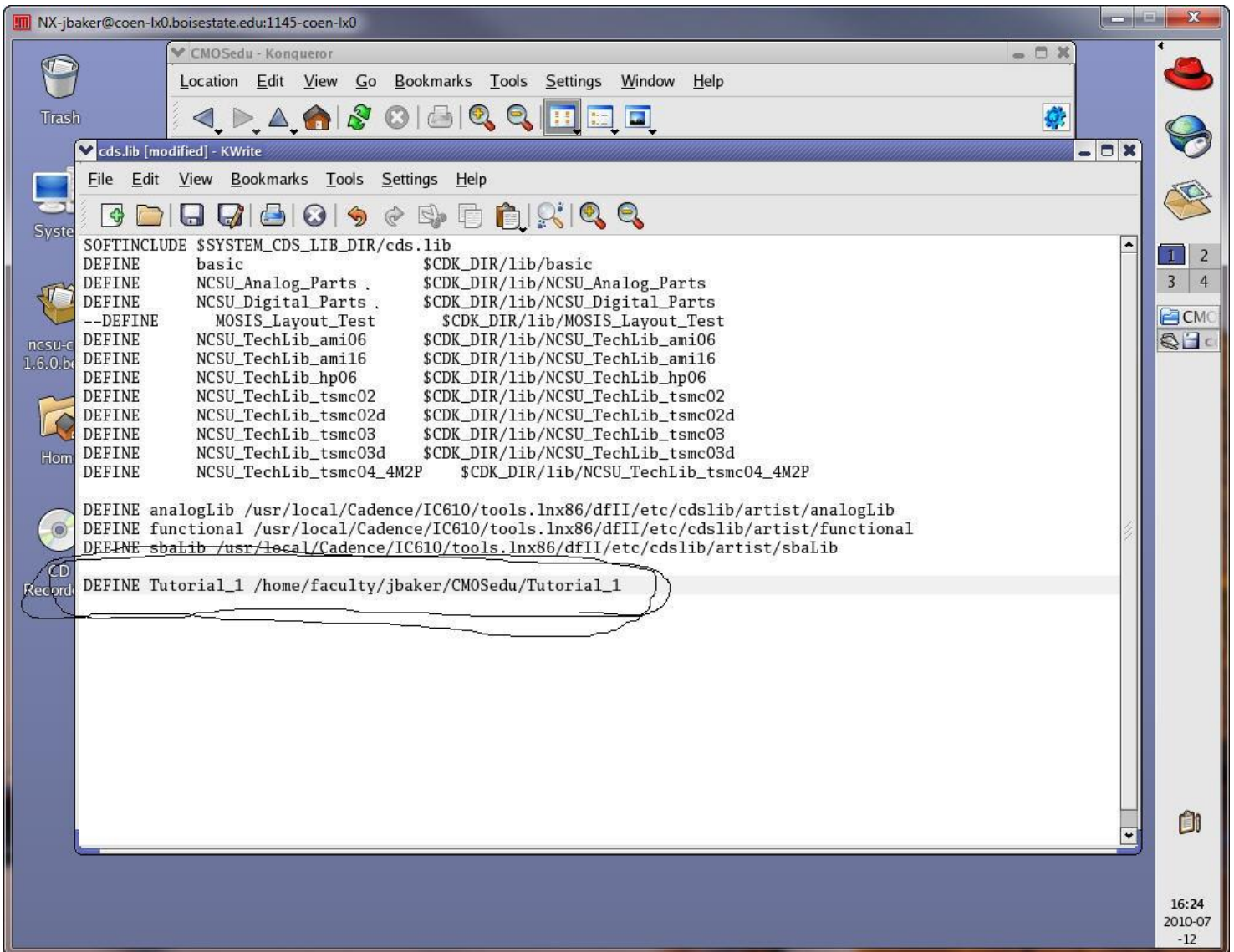


The bottom window is called the Command Interpreter Window or CIW. We need to keep the CIW visible since it tells us what the tools are doing. The other window is the Library Manager. If this window isn't open, or you close it, it can be opened in the CIW using Tools -> Library Manager.

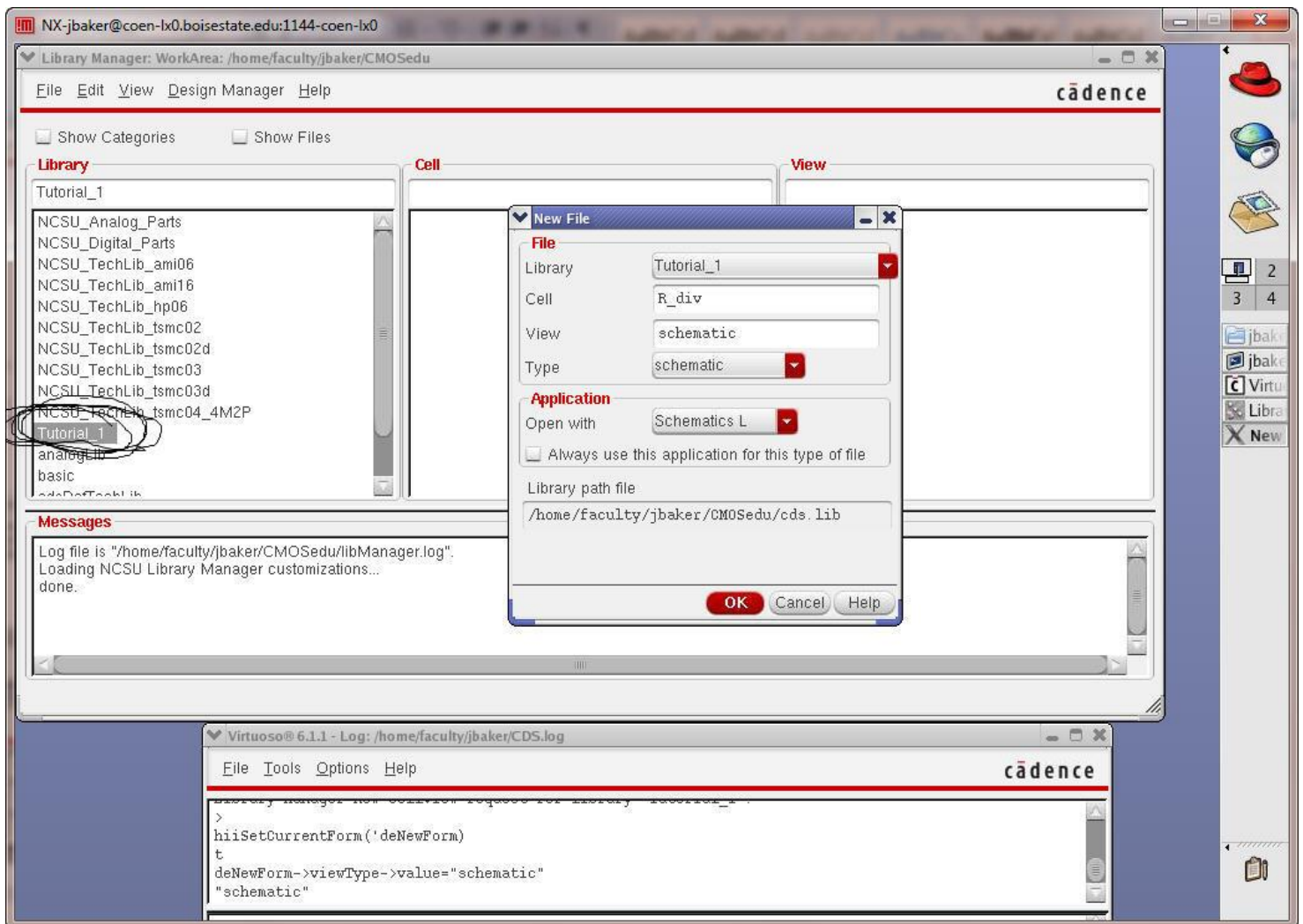
Next let's create a new library by going to, in the Library Manager, File -> New -> Library. The window to create this library, after pressing OK, may be behind other windows so bring it to the front. Call the tutorial Tutorial\_1 and attach the AMI 0.60u C5N process (remember AMI semiconductor is now On semiconductor) as seen below.



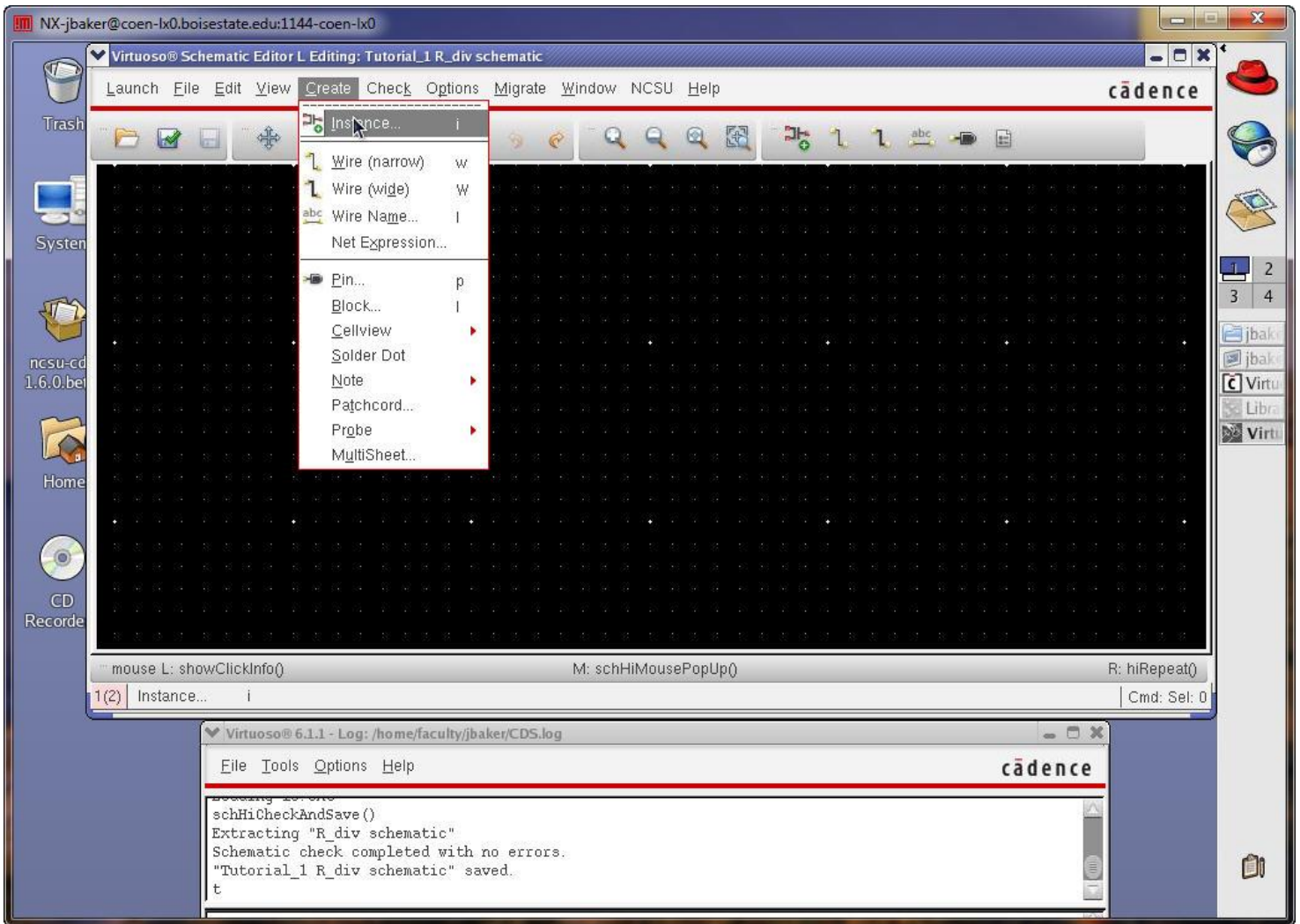
Let's pause for a moment and look at the `cds.lib` file in the `CMOSedu` directory. As seen below when a library is created a definition line is added to the `cds.lib` file. If we wanted to simulate the IC61 book examples found at `CMOSedu.com` we simply unzip, for example, `Ch1_IC61.zip` into the `CMOSedu` directory and add `DEFINE Ch1_IC61 $HOME/CMOSedu/Ch1_IC61` to the `cds.lib` (remembering `$HOME = /home/faculty/jbaker` or my home directory).



Next select the Tutorial\_1 library in the Library Manager and then the menu items File -> New -> Cell View and enter the information seen below. Again note that the window may be behind some other window.

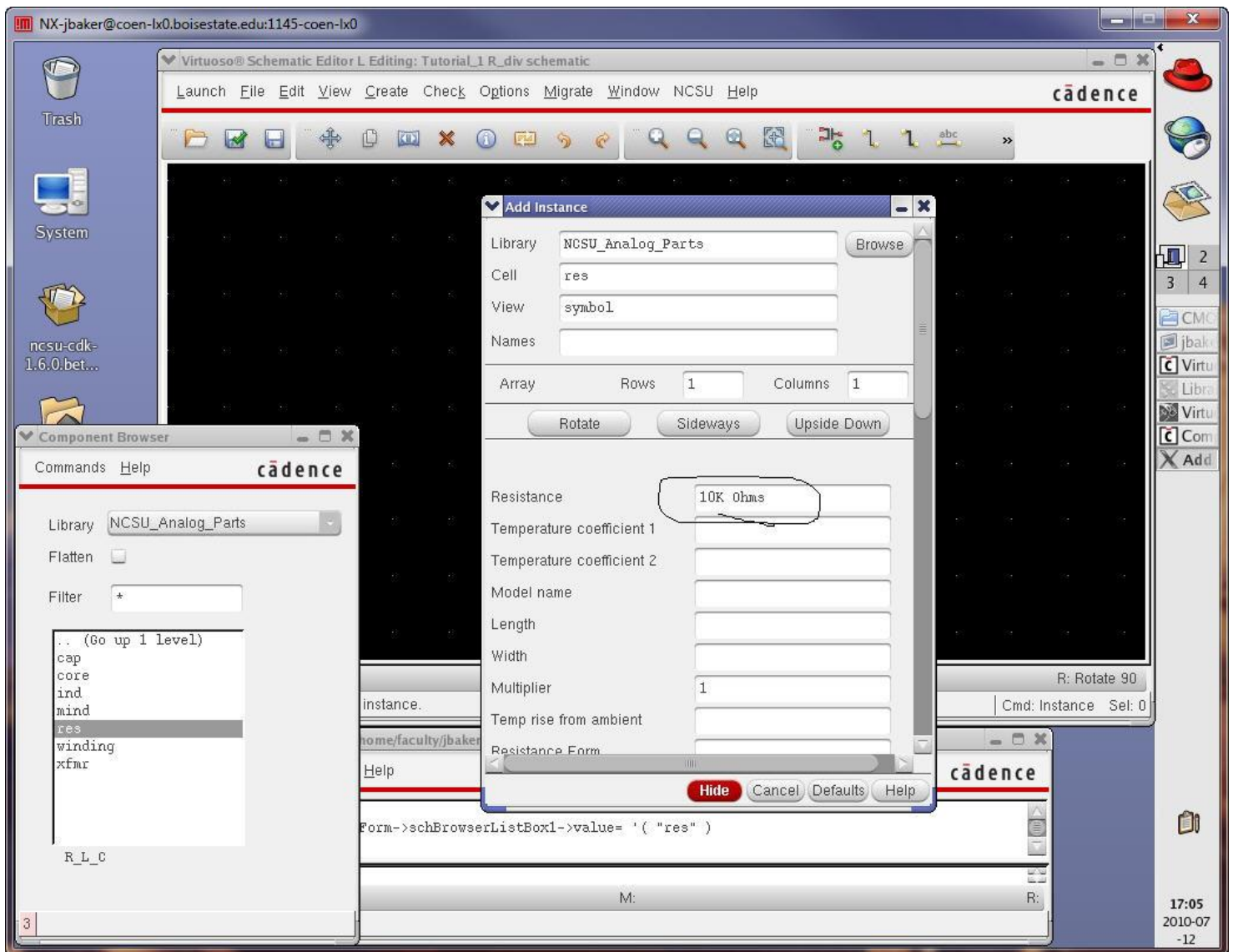


After selecting OK and resizing the window we can add a component (an instance) by going to *Create -> Instance* (or just pressing the Bindkey *i* or use the menu item above the drawing display) as seen below.



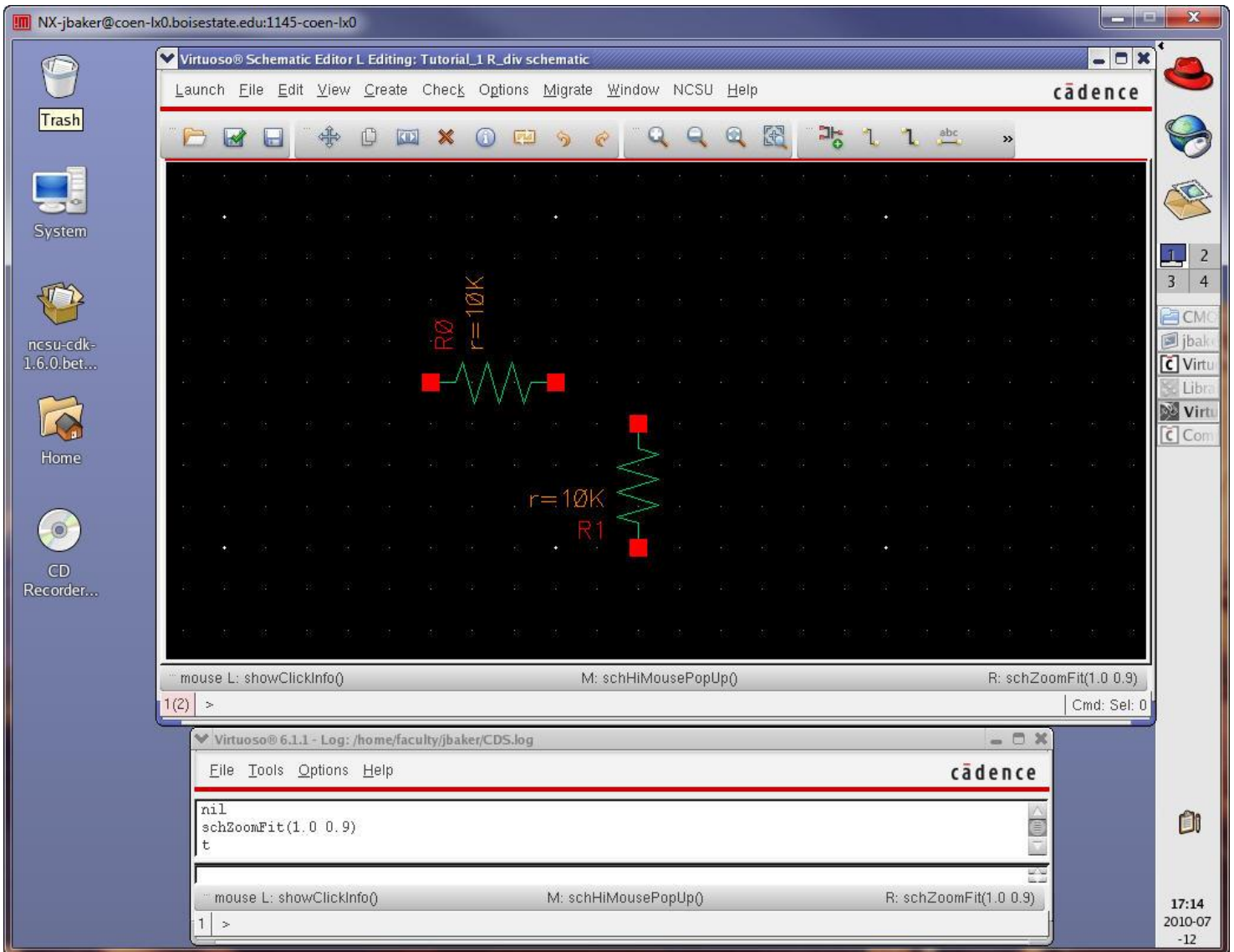
After re-sizing the window, selecting NCSU\_Analog\_Parts (in the Component Browser window), R\_L\_C, and res the following appears. Set the resistance value to 10k as seen below.





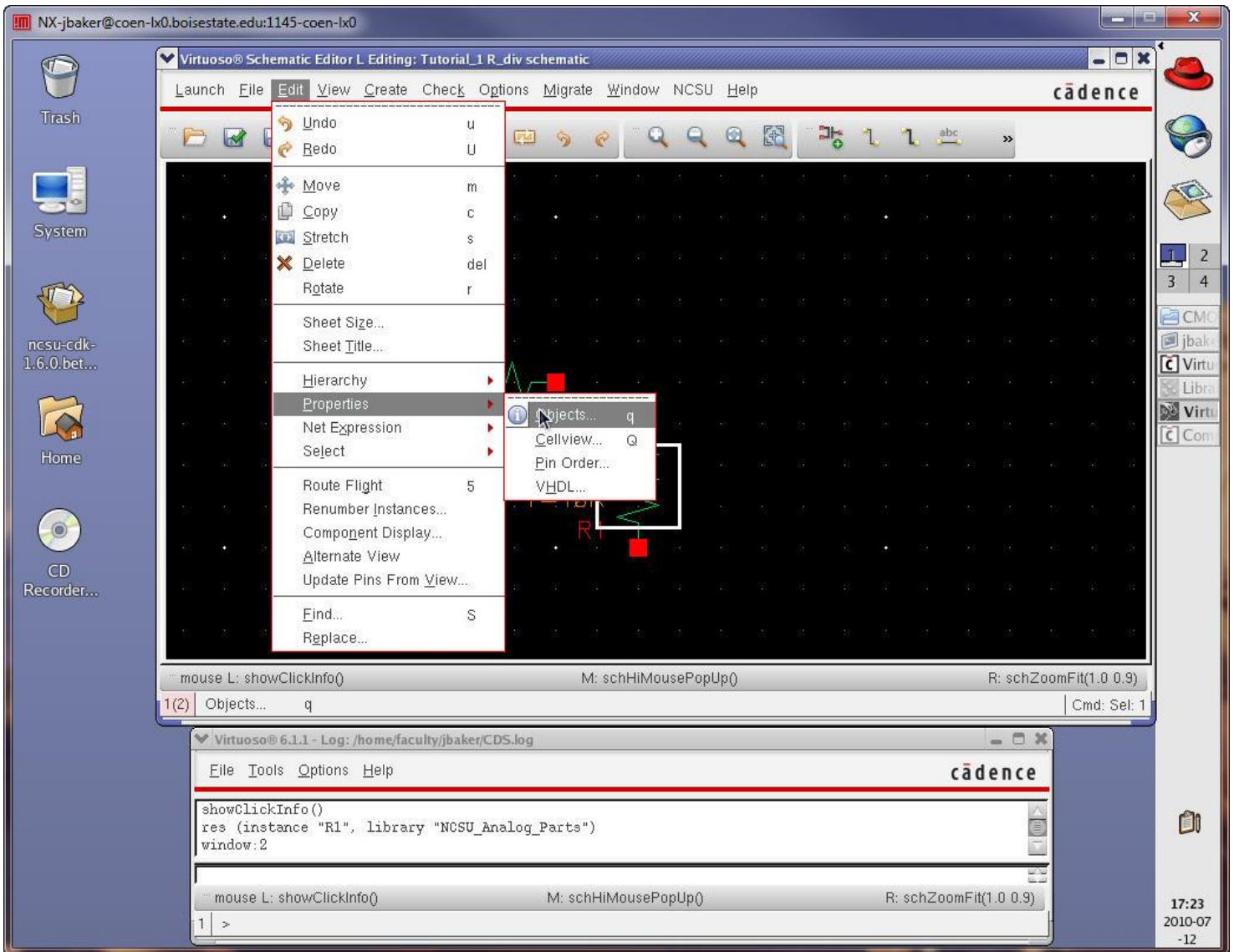
Hide the Add Instance window and minimize the Component Browser to the task bar. Add the two resistors as seen below. Right clicking the mouse button rotates the symbol. Pressing Esc leaves the "Add Instance" mode.

Note that the Bindkey **f** fits the display. A listing of the Bindkeys is found [here](#).



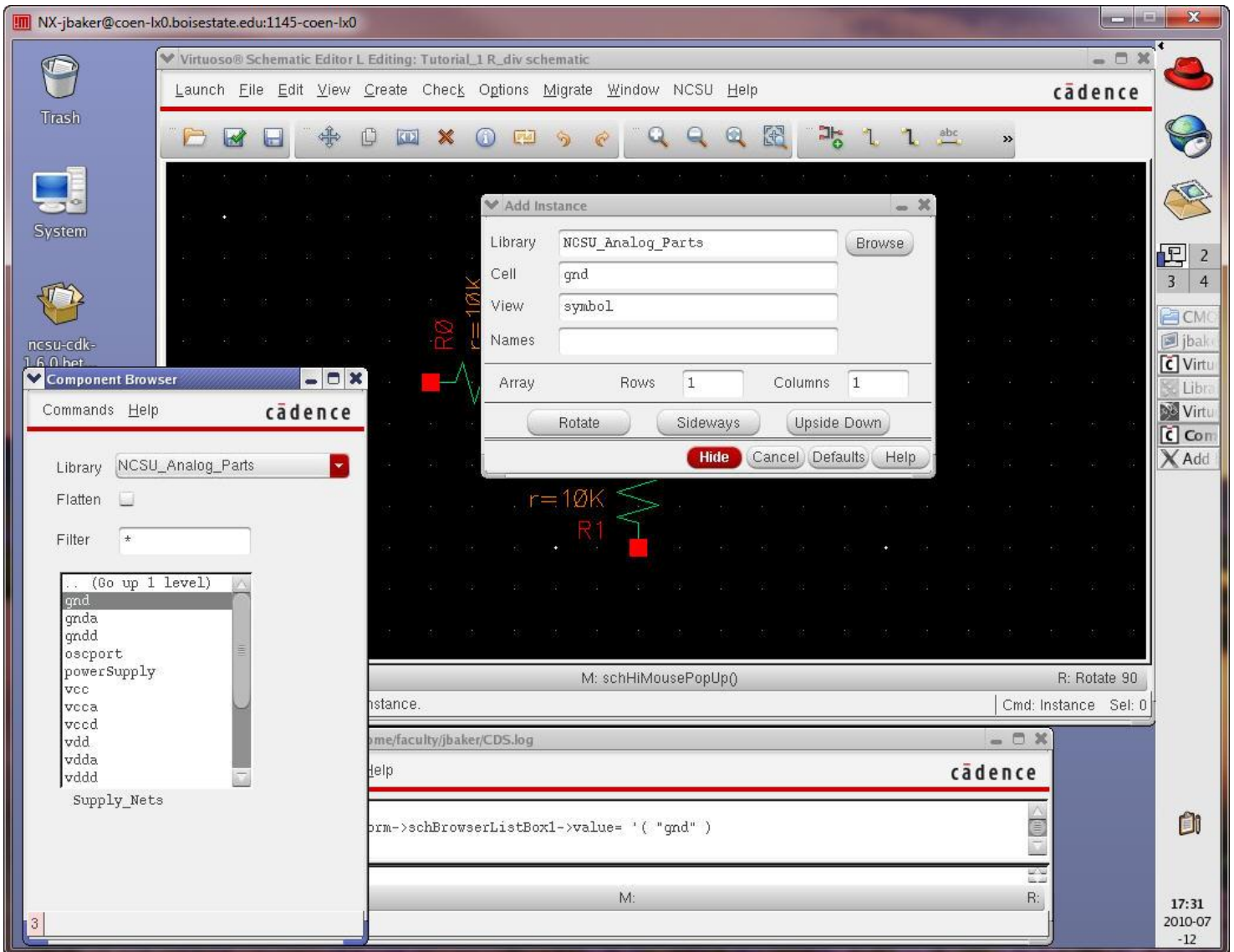
To change the resistor's value select the resistor and use Edit -> Properties -> Objects (or just use the Bindkey q) as seen below. We'll use this command often.

Click your mouse in the drawing area and press Esc a few times so that no commands are active.

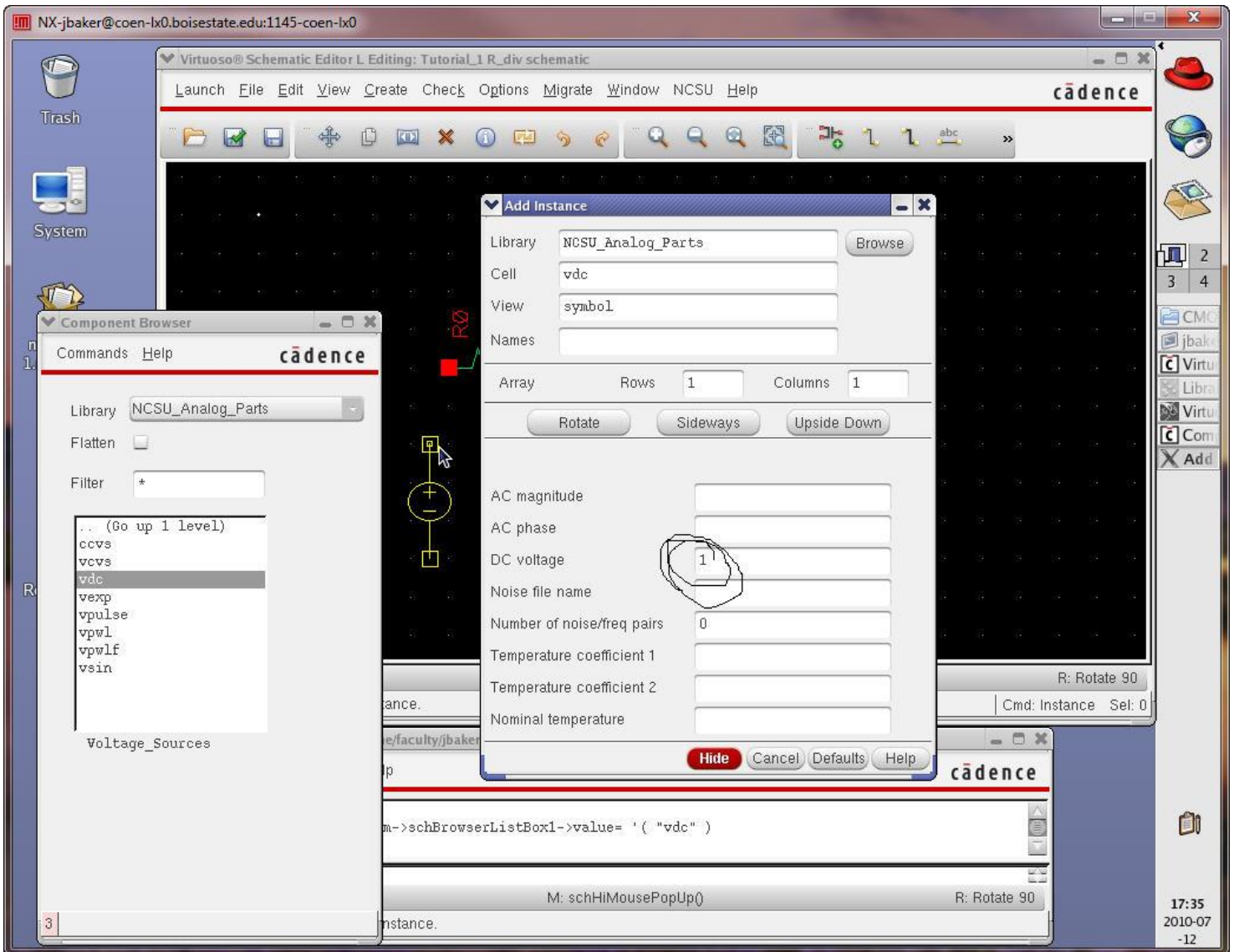


Next add ground to the schematic by pressing **i** (add instance) and maximizing the Component Browser, selecting Supply\_Nets and gnd as seen below.

If you know the name and Library of the instance you want to add you can type them into the fields directly in the Add Instance window.



Add the ground symbol and then add a 1-V DC source, symbol name of vdc under Voltage\_Sources as seen below.



After placing the symbol we need to wire the circuit together. This can be done by using the Bindkey **w** (for wire) or the menu item as seen below.

The screenshot displays the Cadence Virtuoso Schematic Editor interface. The main window shows a schematic diagram of a circuit with the following components and labels:

- A DC voltage source labeled  $V0$  with  $vdc=1$ .
- A resistor labeled  $R0$  with  $r=10K$ .
- A resistor labeled  $R1$  with  $r=10K$ .
- A ground connection labeled  $gnd$ .

The interface includes a menu bar (Launch, File, Edit, View, Create, Check, Options, Migrate, Window, NCSU, Help) and a toolbar. A status bar at the bottom of the schematic editor shows the command: `1(2) Point at starting point for the router or snap to diamond using the "s" key.`

Below the schematic editor, a log window titled "Virtuoso@ 6.1.1 - Log: /home/faculty/jbaker/CDS.log" is open, showing the following log entries:

```
t
schZoomFit(1.0 0.9)
t
```

The log window also has a status bar with the command: `1 Point at starting point for the router or snap to diamond using the "s" key.`

The system tray on the right shows the time as 17:41 on 2010-07-12.

It's often useful to label wires with signal names. Using the Bindkey I (lowercase L) or the menu item enables naming wires. Let's do this as seen below.

The screenshot displays the Cadence Virtuoso Schematic Editor interface. The main window shows a schematic diagram of a circuit with the following components and labels:

- A DC voltage source labeled  $V0$  with  $vdc=1$ .
- An input terminal labeled  $in$ .
- A resistor labeled  $R0$  with  $r=10K$ .
- An output terminal labeled  $out$ .
- A resistor labeled  $R1$  with  $r=10K$ .
- A ground terminal labeled  $gnd$ .

The circuit is connected as follows: The  $in$  terminal is connected to the positive terminal of  $V0$ . The positive terminal of  $V0$  is also connected to resistor  $R0$ . The other end of  $R0$  is connected to the  $out$  terminal. The  $out$  terminal is also connected to resistor  $R1$ . The other end of  $R1$  is connected to the  $gnd$  terminal. The negative terminal of  $V0$  is also connected to  $gnd$ .

Below the schematic, a status bar shows the command: `1(2) Use the options form to supply the wire names.` and `Cmd: Wire Name Sel: 0`.

A log window titled "Virtuoso@ 6.1.1 - Log: /home/faculty/jbaker/CDS.log" is open at the bottom, showing the following log entries:

```

t
mouseAddPt()
t
mouse L: mouseAddPt()      M: schHiMousePopUp()      R: Rotate 90
1 Use the options form to supply the wire names.

```

The system clock in the bottom right corner shows 17:47 on 2010-07-12.

We are about ready to simulate the operation of this circuit. Let's do a "Check and Save" first. If we have edited the schematic and try to simulate without checking and saving first the simulation will fail.

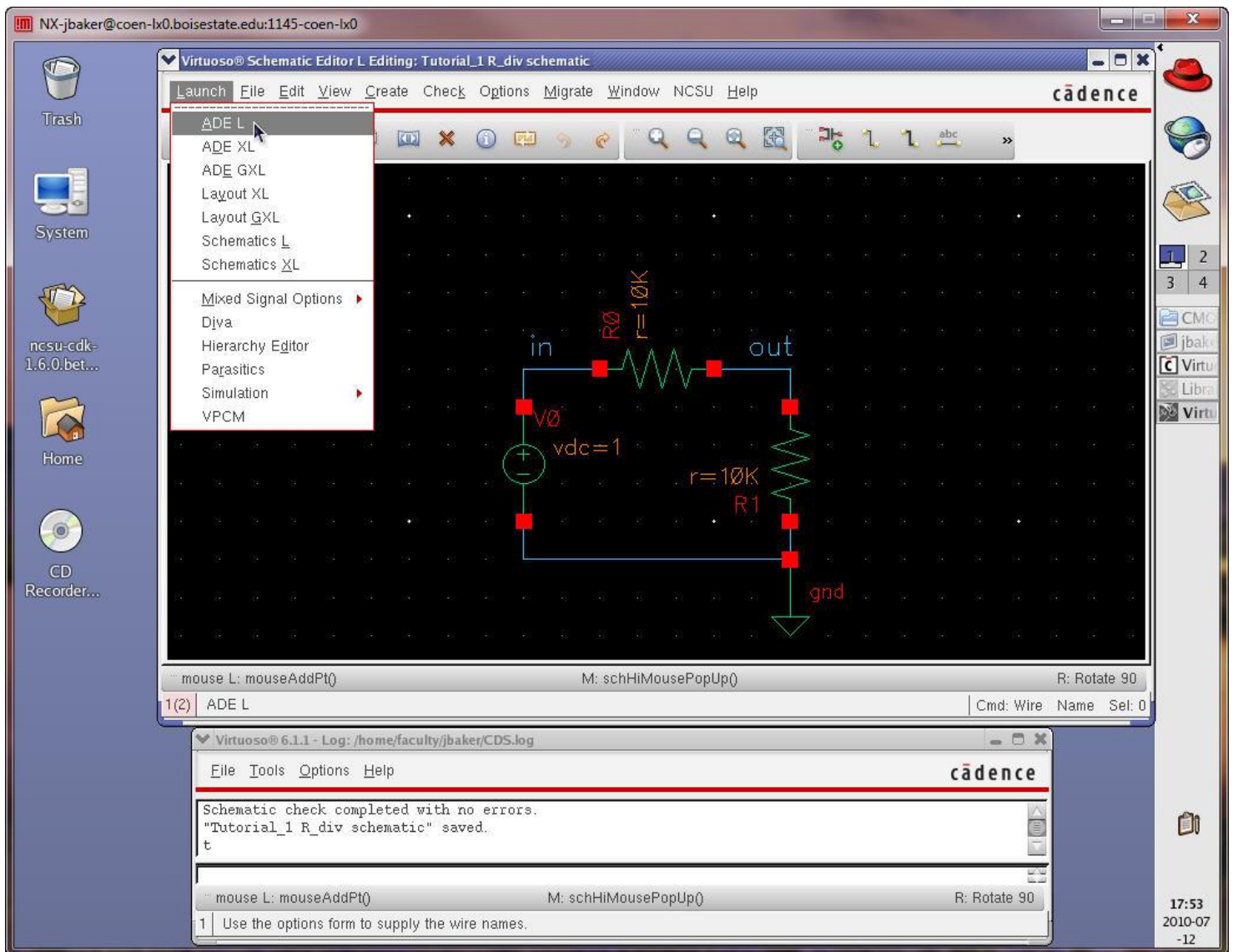
The screenshot displays the Cadence Virtuoso Schematic Editor interface. The main window shows a circuit schematic with a voltage source labeled  $v_{d0}$  and  $v_{dc}=1$ , connected to a resistor  $R0$  with value  $r=10K$ . The output is labeled  $out$ . A second resistor  $R1$  with value  $r=10K$  is connected to ground ( $gnd$ ). The input is labeled  $in$ . A tooltip for the "Check and Save" icon is visible. Below the schematic, a log window titled "Virtuoso@ 6.1.1 - Log: /home/faculty/jbaker/CDS.log" displays the following text:

```
File Tools Options Help
-----
Schematic check completed with no errors.
"Tutorial_1_R_div schematic" saved.
t
mouse L: mouseAddPt() M: schHiMousePopUp() R: Rotate 90
1 Use the options form to supply the wire names.
```

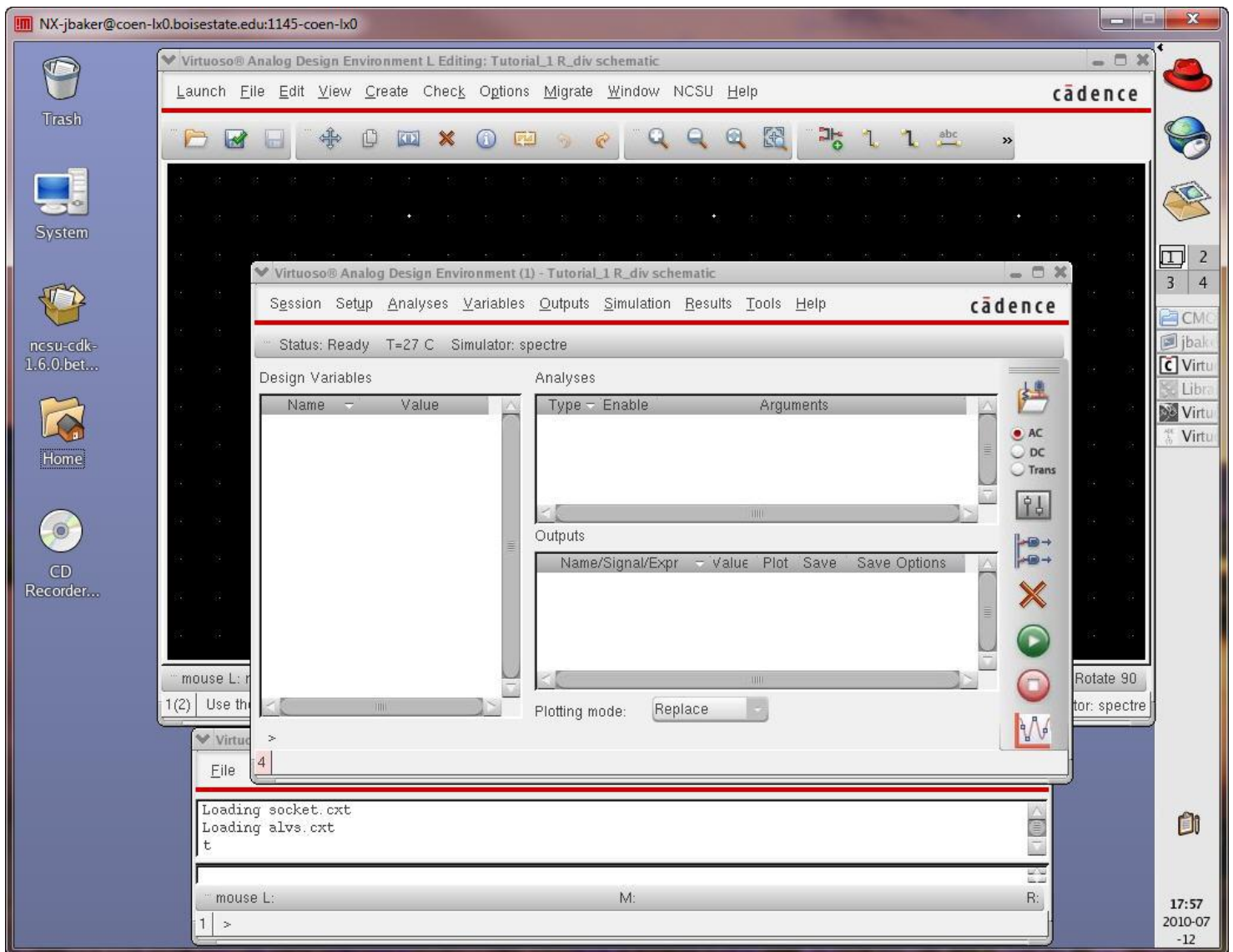
The system tray in the bottom right corner shows the time 17:49, date 2010-07, and page number -12.

To simulate with Spectre (Cadence's SPICE simulator) go to the menu Launch → ADE L as seen below.

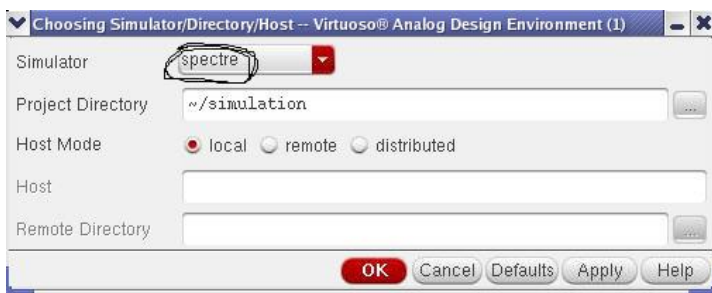




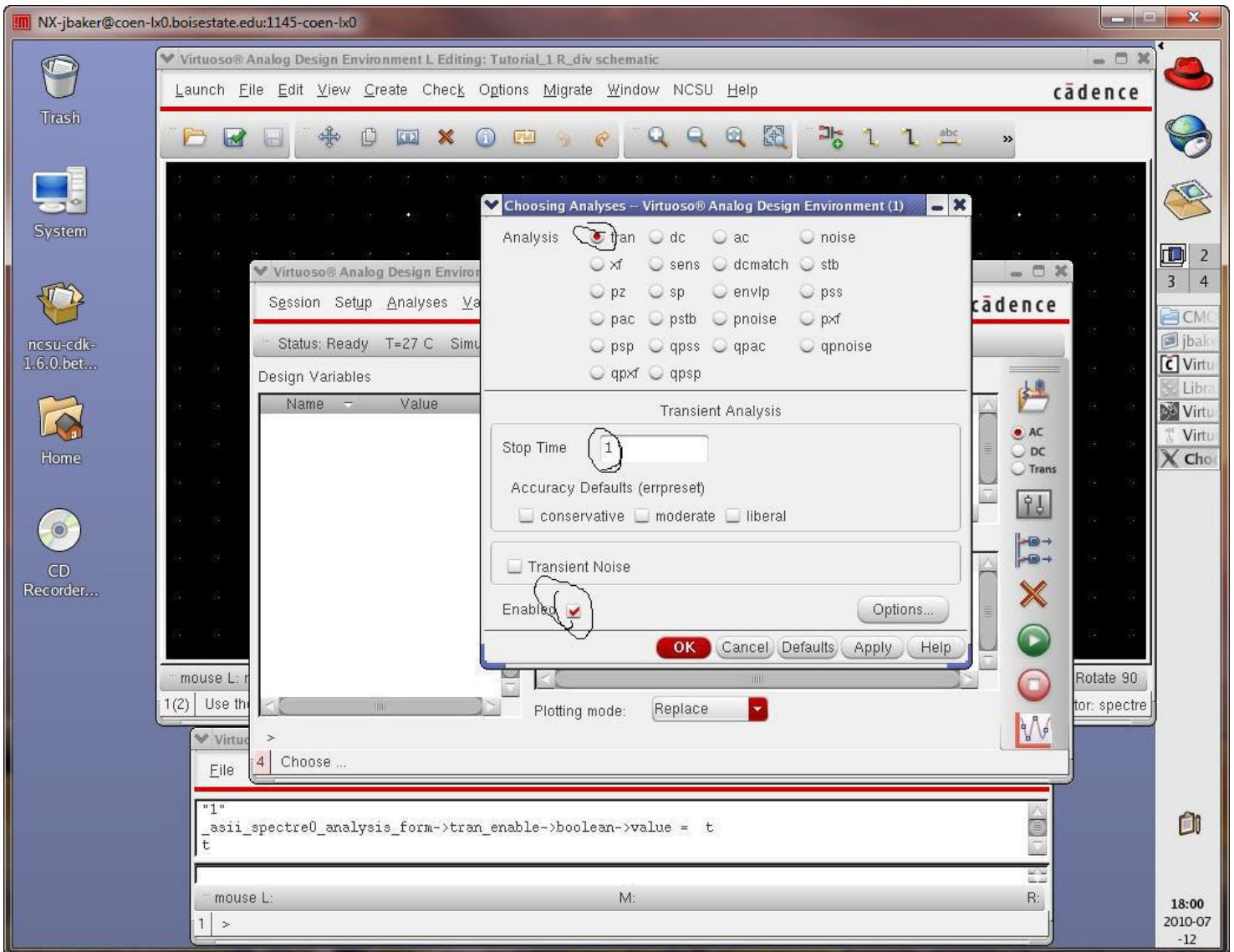
The Virtuoso Analog Design Environment (ADE) window should appear as seen below.



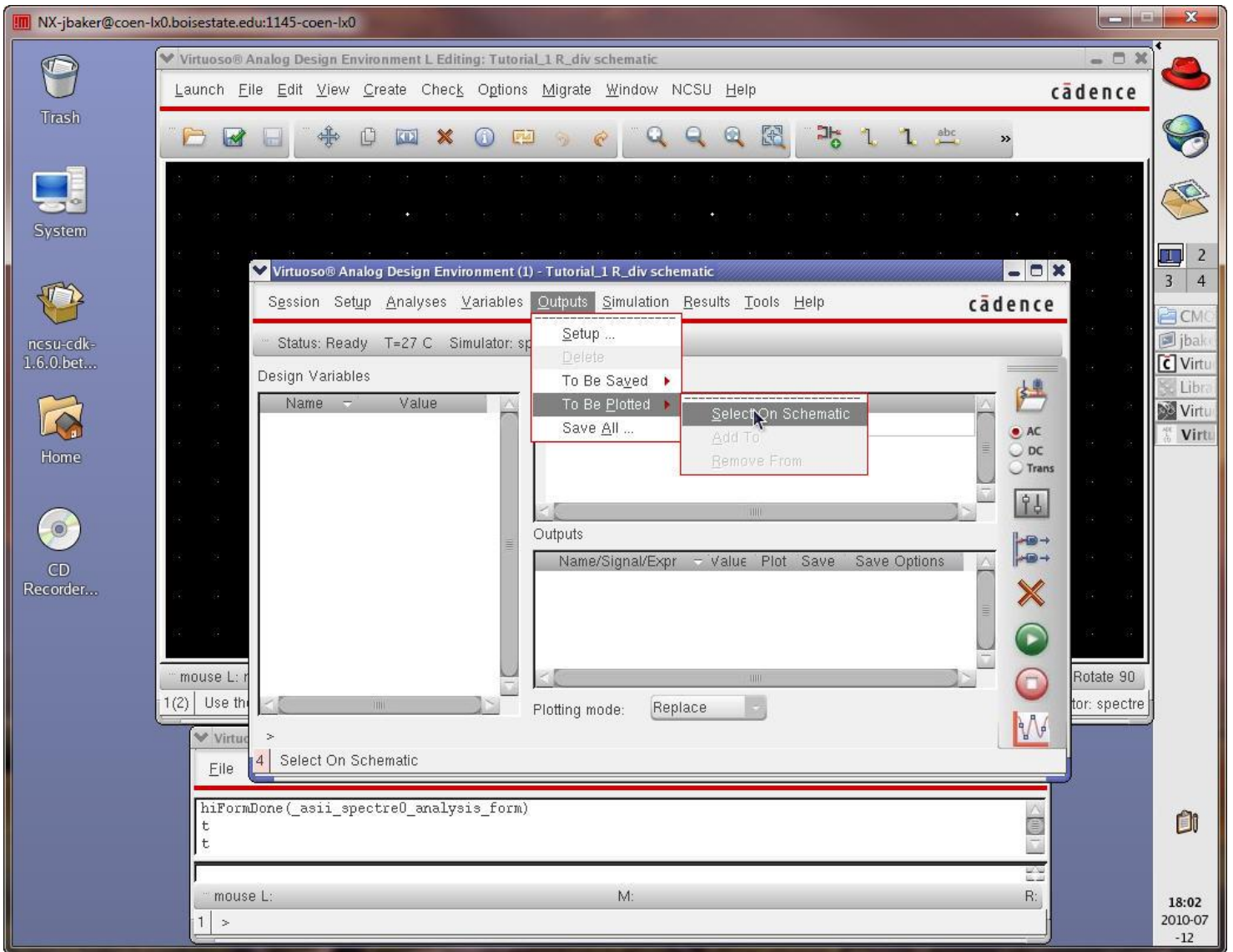
Use the menu in the ADE and go to Setup -> Simulator/Directory/Host to verify the simulator is set to Spectre. We'll assume throughout these tutorials that Spectre is used. If it's not the default then please re-visit above instructions.



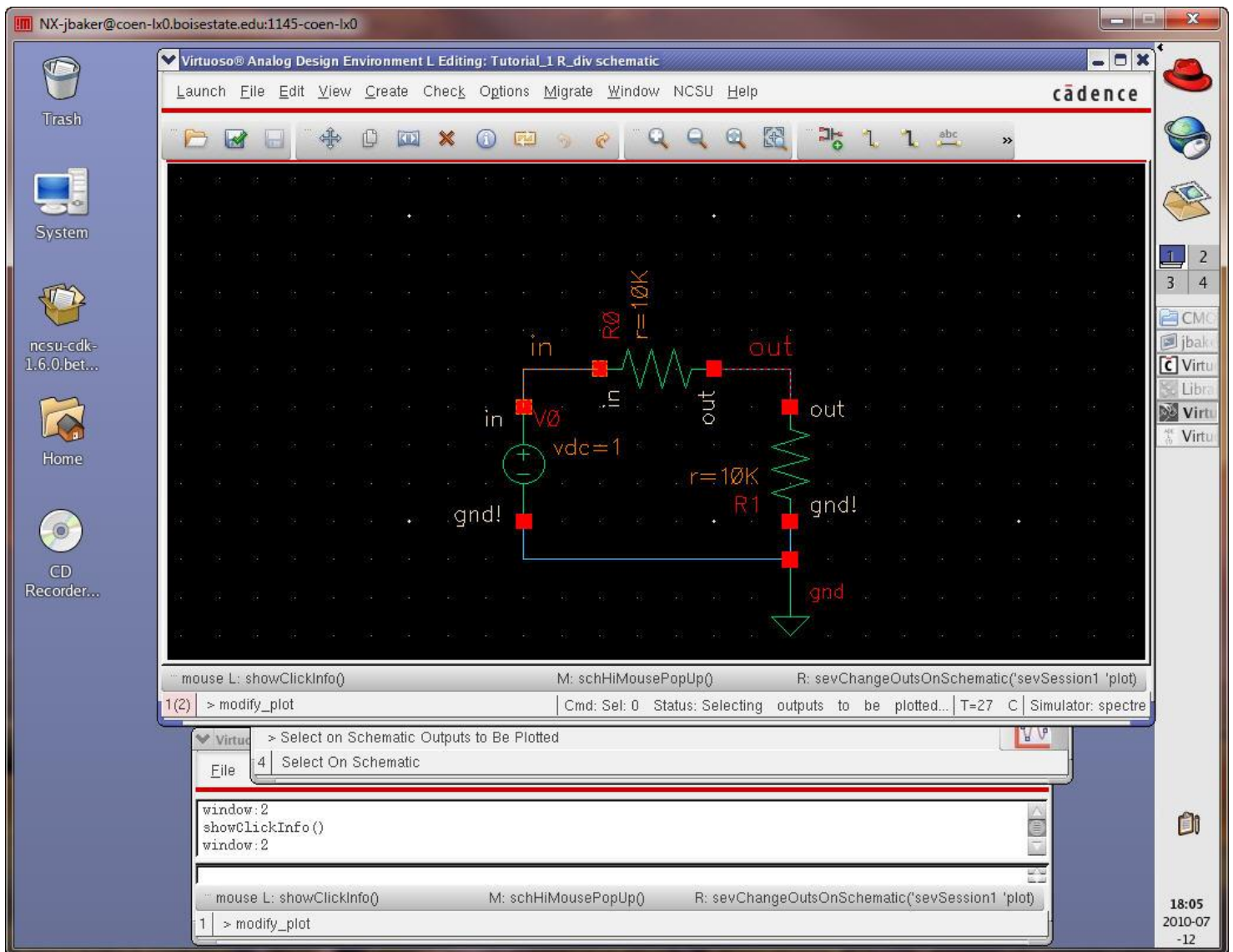
Next go to Analyses -> Choose and select a transient analysis (tran), a stop time of 1 second, and Enabled as seen below.



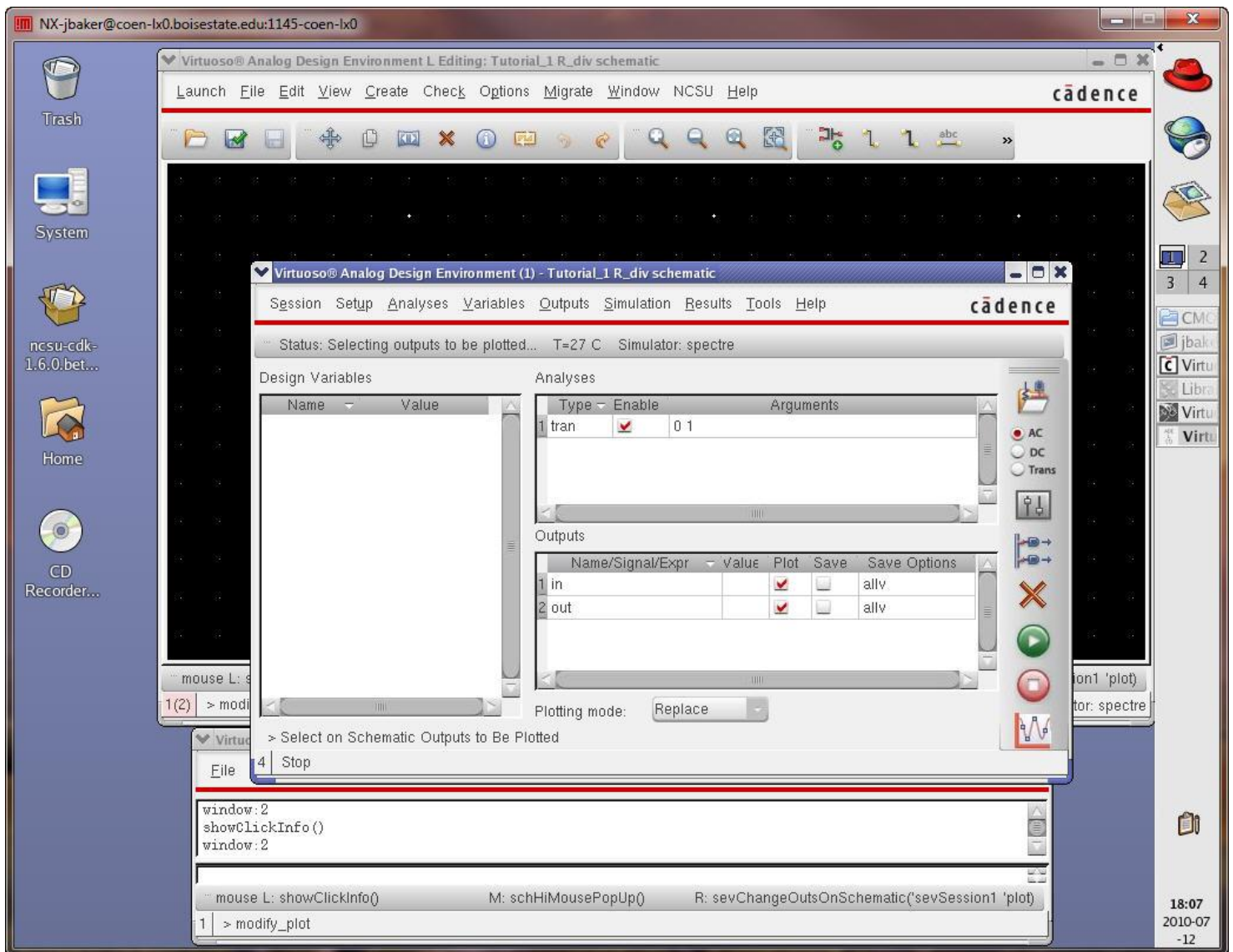
Next we need to select the signals we want to plot. Follow the menu items seen below selecting "Select On Schematic".



Select the wires as seen below.

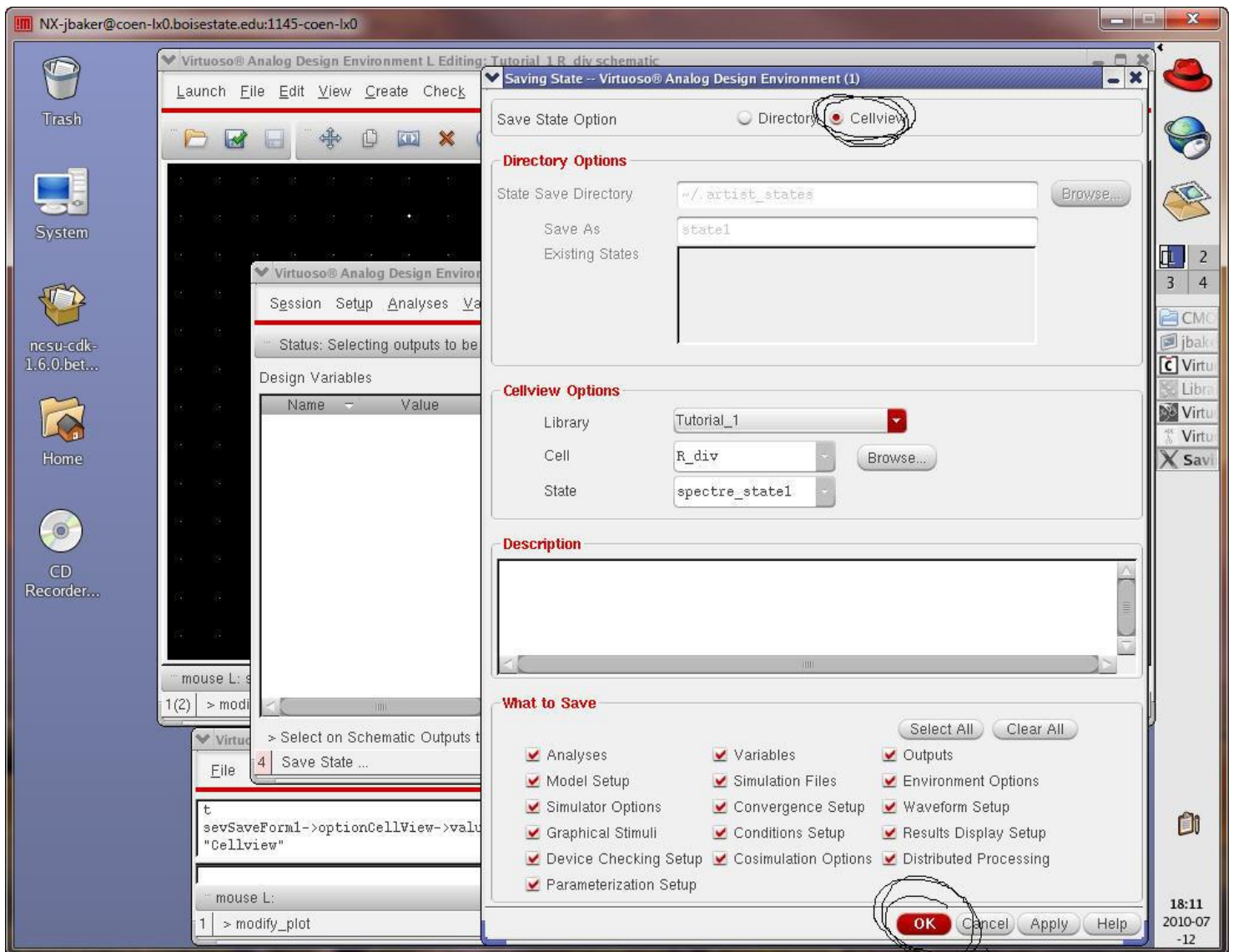


The ADE should now look like the following (bring the window to the front via the task bar).



We can now simulate the schematic by pressing the green "Netlist and Run" button. However, before we do this let's save this information so that next time we want to simulate this circuit we don't have to go through these steps again.

In the ADE window use the menu items Session -> Save State and select Cellview then OK as seen below.



To load this state we select, in the ADE window, Session -> Load State and Cellview then OK. This is important to save time and used throughout the examples from CMOSedu.com.

Now pressing the green button and running the simulation results in the following.

The screenshot displays the Cadence Virtuoso Analog Design Environment. The terminal window on the left shows the results of a transient analysis:

```

tran: time = 91 ms      (9.1 %), step = 20 ms      (2 %)
tran: time = 131 ms    (13.1 %), step = 20 ms      (2 %)
tran: time = 191 ms    (19.1 %), step = 20 ms      (2 %)
tran: time = 231 ms    (23.1 %), step = 20 ms      (2 %)
tran: time = 291 ms    (29.1 %), step = 20 ms      (2 %)
tran: time = 331 ms    (33.1 %), step = 20 ms      (2 %)
tran: time = 391 ms    (39.1 %), step = 20 ms      (2 %)
tran: time = 431 ms    (43.1 %), step = 20 ms      (2 %)
tran: time = 491 ms    (49.1 %), step = 20 ms      (2 %)
tran: time = 531 ms    (53.1 %), step = 20 ms      (2 %)
tran: time = 591 ms    (59.1 %), step = 20 ms      (2 %)
tran: time = 631 ms    (63.1 %), step = 20 ms      (2 %)
tran: time = 691 ms    (69.1 %), step = 20 ms      (2 %)
tran: time = 731 ms    (73.1 %), step = 20 ms      (2 %)
tran: time = 791 ms    (79.1 %), step = 20 ms      (2 %)
tran: time = 831 ms    (83.1 %), step = 20 ms      (2 %)
tran: time = 891 ms    (89.1 %), step = 20 ms      (2 %)
tran: time = 931 ms    (93.1 %), step = 20 ms      (2 %)
tran: time = 985.5 ms  (98.6 %), step = 14.5 ms    (1.45)
Number of accepted tran steps = 54.
Accumulated tran full load time = 0 s.
  Accumulated tran full component evaluation time = 0 s.
  Accumulated tran full preload time = 0 s.
  Accumulated tran full merge time = 0 s.
Accumulated tran residue-only load time = 0 s.
  Accumulated tran residue-only component evaluation time = 0 s.
  Accumulated tran residue-only preload time = 0 s.
  Accumulated tran residue-only merge time = 0 s.
Accumulated tran factor time = 0 s.
Accumulated tran solve time = 0 s.
Accumulated tran output time = 0 s.
Initial condition solution time = 0 s.
Intrinsic tran analysis time = 10 ms.
Total time required for tran analysis `tran' was 10 ms.

finalTimeOP: writing operating point information to rawfile.
modelParameter: writing model parameter values to rawfile.
element: writing instance parameter values to rawfile.
outputParameter: writing output parameter values to rawfile.
designParamVals: writing netlist parameters to rawfile.
primitives: writing primitives to rawfile.
subckts: writing subcircuits to rawfile.

```

The plot window on the right, titled "Transient Response", shows a graph of voltage  $V(V)$  versus time (s). The y-axis ranges from 0.4 to 1.1, and the x-axis ranges from 0.0 to 1.0. Two traces are visible:  $/in$  (red) and  $/out$  (black). Both traces show a step response, with  $/in$  stepping from approximately 0.5 to 1.0 and  $/out$  stepping from approximately 0.4 to 0.9 at  $t = 0$ .

Which is, hopefully, what the reader expected :-)

Let's close the ADE and go back to the schematic.



The screenshot displays the Cadence Virtuoso Schematic Editor interface. The main window shows a schematic diagram of a circuit. The circuit consists of a voltage source labeled  $vdc=1$  connected to a resistor labeled  $R0$  with a value of  $r=10K$ . The output of this resistor is connected to another resistor labeled  $R1$  with a value of  $r=10K$ , which is connected to ground. The input and output nodes are labeled  $in$  and  $out$ . The schematic is set against a black background with a grid.

Below the schematic, a command window is open, showing the following text:

```

Virtuoso@ 6.1.1 - Log: /home/faculty/jbaker/CDS.log
File Tools Options Help
-----
hiCloseWindow(window(4))
t
-----
mouse L: showClickInfo()      M: schHiMousePopUp()      R: sevNetlistAndRun("sevSession1")
1(2) >
-----

```

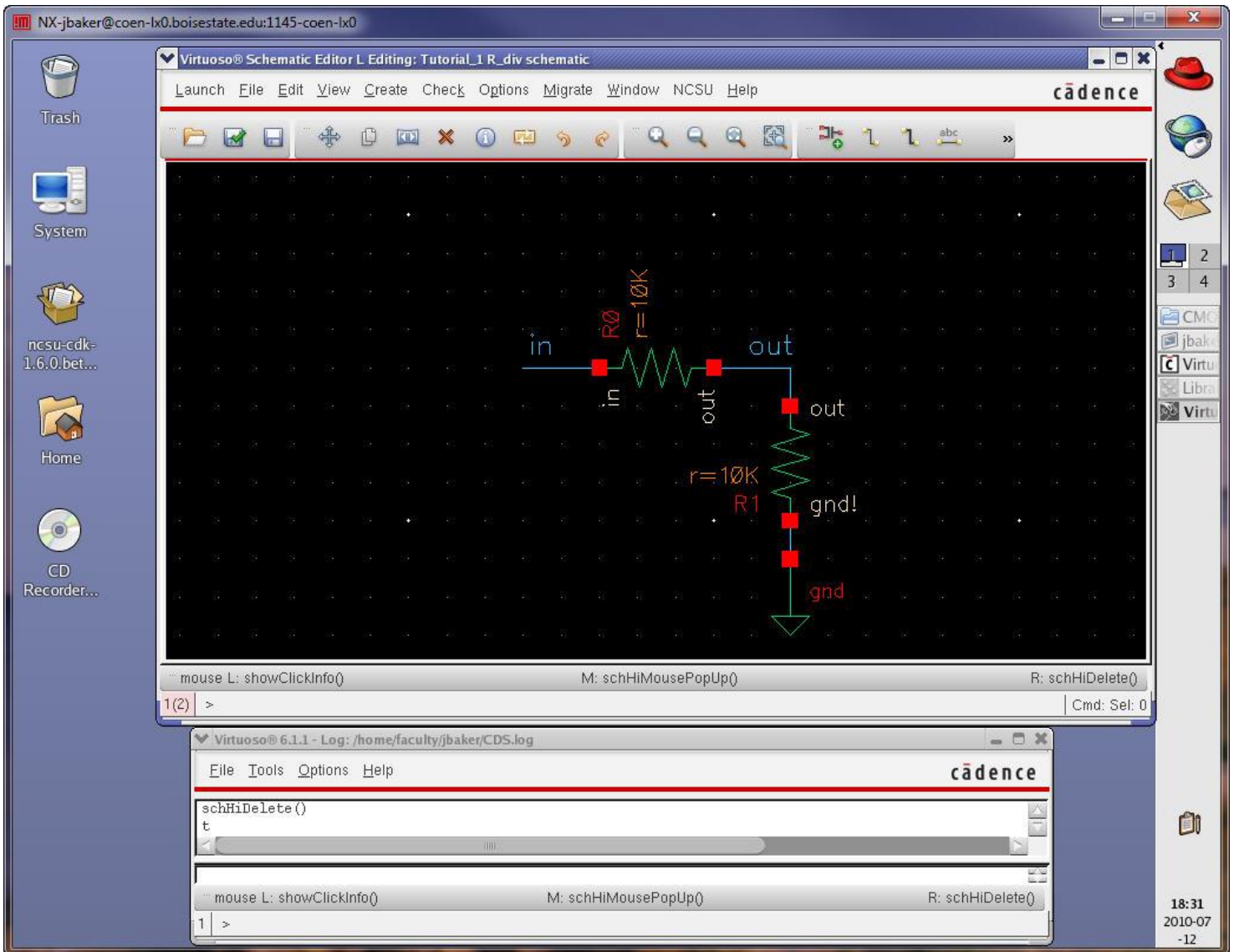
The system tray on the right side of the window shows the date and time: 18:25 2010-07 -12.

While this is a general purpose schematic let's make it more useful as a cell in an integrated circuit. Select, and delete, the bottom wire and voltage source.

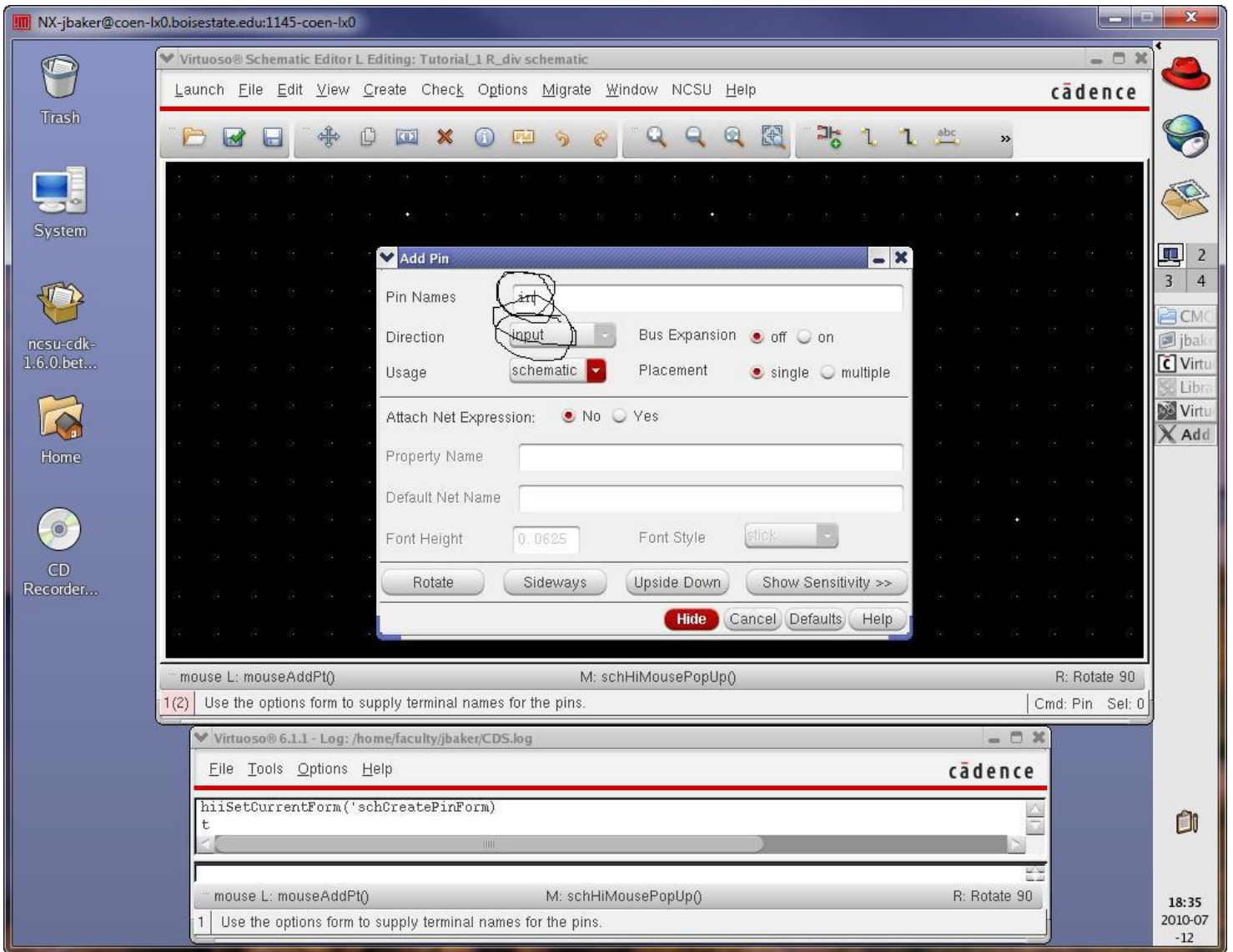
Note that once you select and delete one wire or component Virtuoso is in the delete mode so all you have to do is click on the next item you want to delete.

To exit this mode press Esc. To undo an action press u.

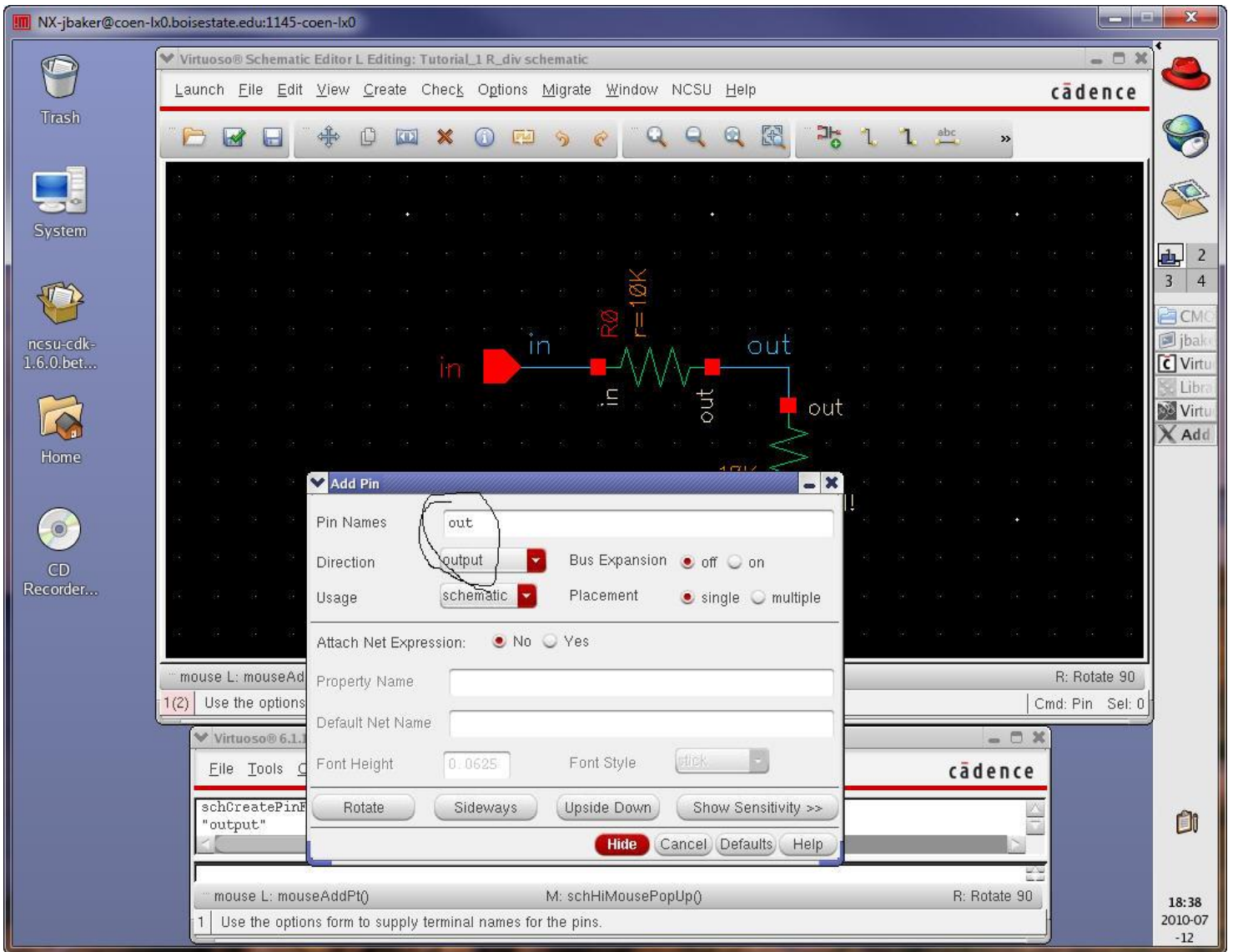
Ensure you have the following schematic.



Let's add pins to the schematic using the menu Create -> Pin (or the Bindkey p). Pins are what is used to connect the schematic symbol, which will make shortly, up in a higher-ranking schematic view. Add an input pin called in (so it matches the wire name, useful but not necessary).



And an output pin.



Adding a wire to the pin and "Checking and Saving" results in the following.

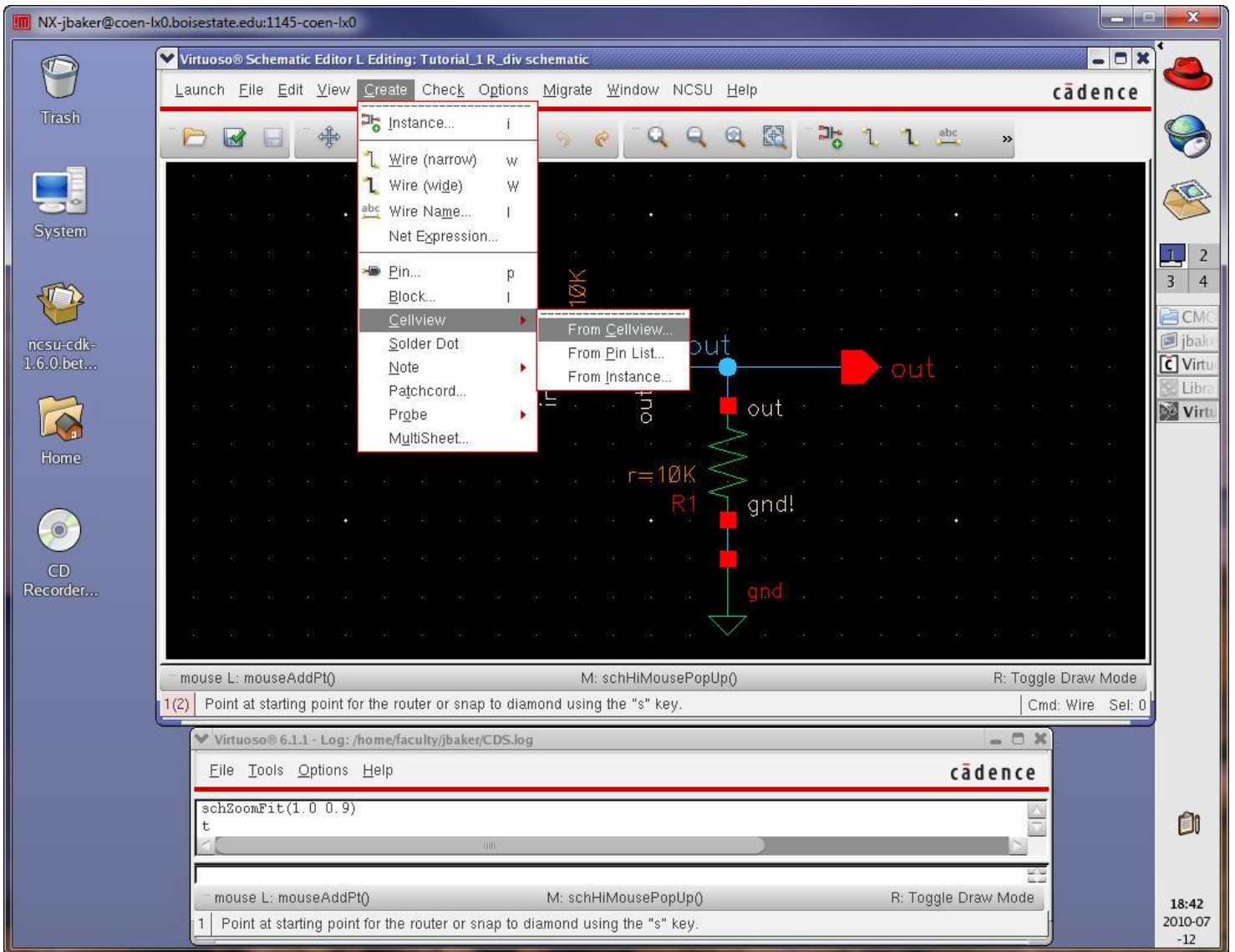
The screenshot displays the Cadence Virtuoso Schematic Editor interface. The main window shows a schematic diagram of a circuit. The circuit consists of an input terminal labeled 'in' connected to a red buffer symbol. The output of the buffer is connected to a resistor labeled 'R0' with a value of  $r=10K$ . The output of this resistor is connected to a blue node labeled 'out'. This node is also connected to another resistor labeled 'R1' with a value of  $r=10K$ , which is connected to a ground symbol labeled 'gnd!'. The output of this second resistor is connected to another red buffer symbol, which has an output terminal labeled 'out'.

Below the schematic editor, a command window is open, showing the following text:

```
Virtuoso@ 6.1.1 - Log: /home/faculty/jbaker/CDS.log  
File Tools Options Help  
schZoomFit(1.0 0.9)  
t  
mouse L: mouseAddPt() M: schHiMousePopUp() R: Toggle Draw Mode  
1 Point at starting point for the router or snap to diamond using the "s" key. Cmd: Wire Sel: 0
```

The system tray on the right shows the time as 18:40, the date as 2010-07, and the page number as -12.

Let's create a symbol for this schematic following the steps seen below.



The screenshot displays the Cadence Virtuoso Schematic Editor interface. The main window is titled "Virtuoso® Schematic Editor L Editing: Tutorial\_1\_R\_div schematic". A "Cellview From Cellview" dialog box is open, showing the following settings:

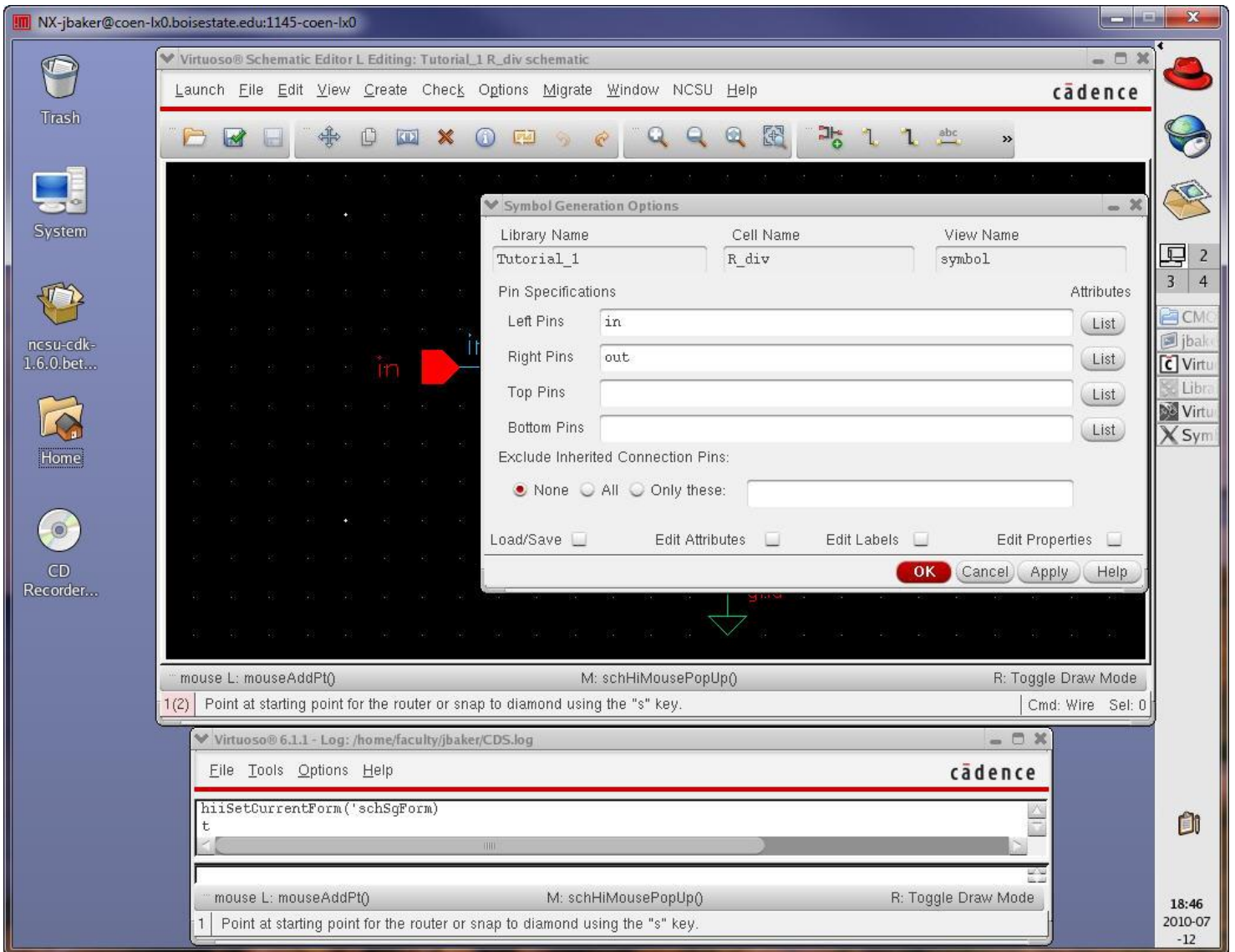
- Library Name: Tutorial\_1
- Cell Name: R\_div
- From View Name: schematic
- To View Name: symbol
- Tool / Data Type: schematicSymbol
- Display Cellview:
- Edit Options:

The dialog box has buttons for OK, Cancel, Defaults, Apply, and Help. In the background, a schematic diagram is visible, showing a red resistor labeled "R1" connected to a red triangle labeled "gnd!". A red arrow labeled "in" points to the left side of the resistor. The status bar at the bottom of the editor shows "1(2) Point at starting point for the router or snap to diamond using the "s" key." and "Cmd: Wire Sel: 0".

Below the main editor window, a "Virtuoso® 6.1.1 - Log: /home/faculty/jbaker/CDS.log" window is open, displaying the following log entry:

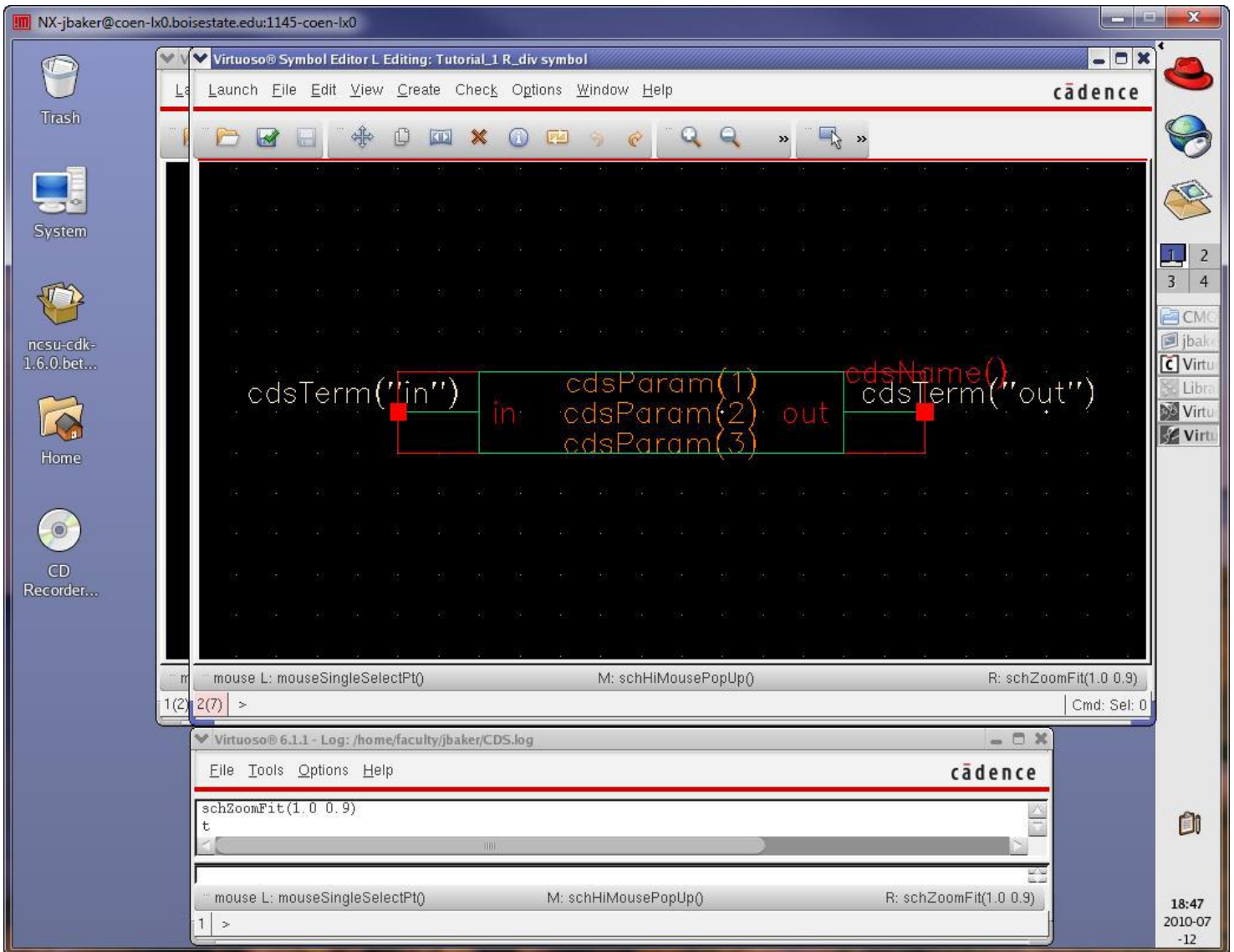
```
hiiSetCurrentForm('schViewToViewForm')
t
```

The log window also shows the same status bar information as the main editor window.

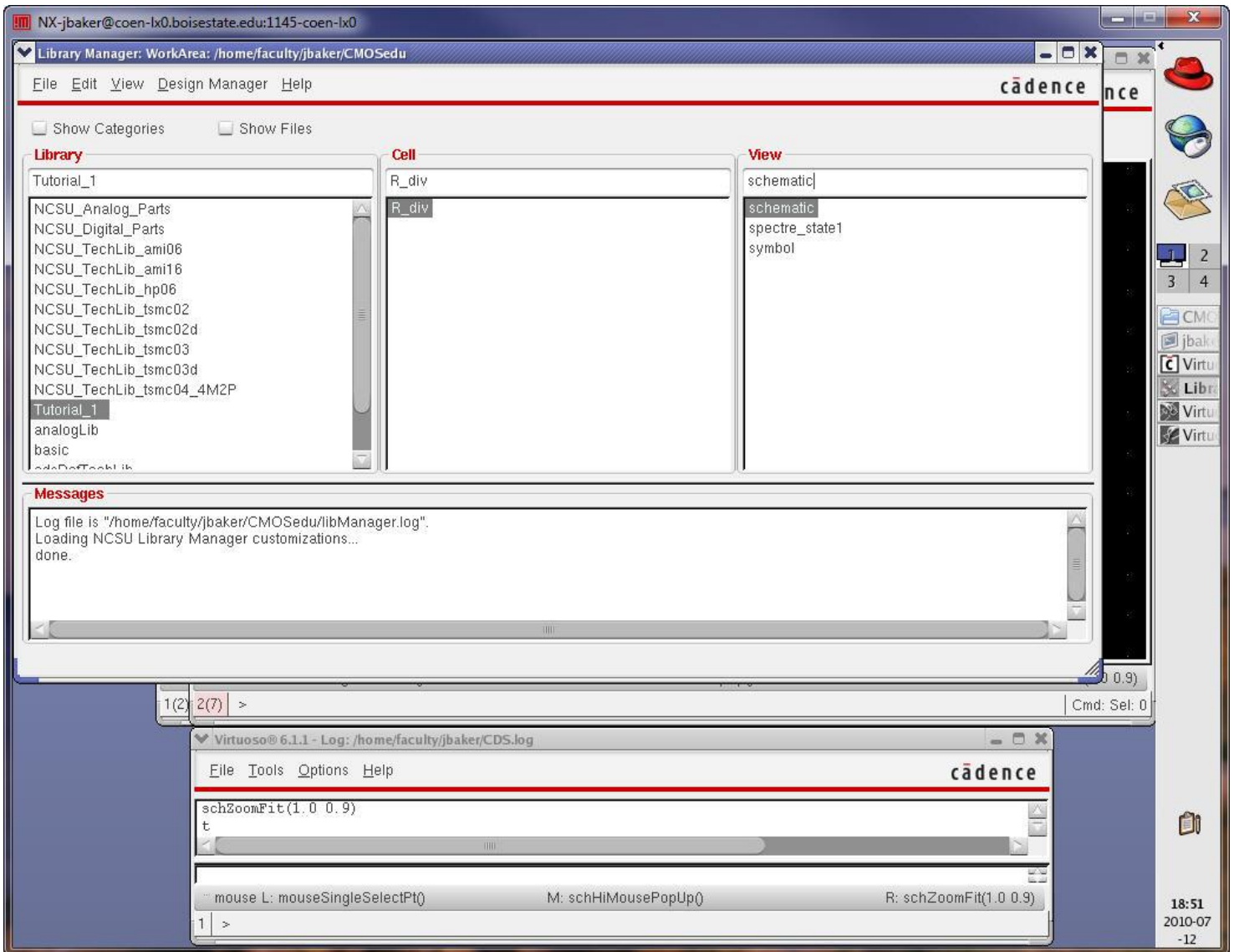


And finally,

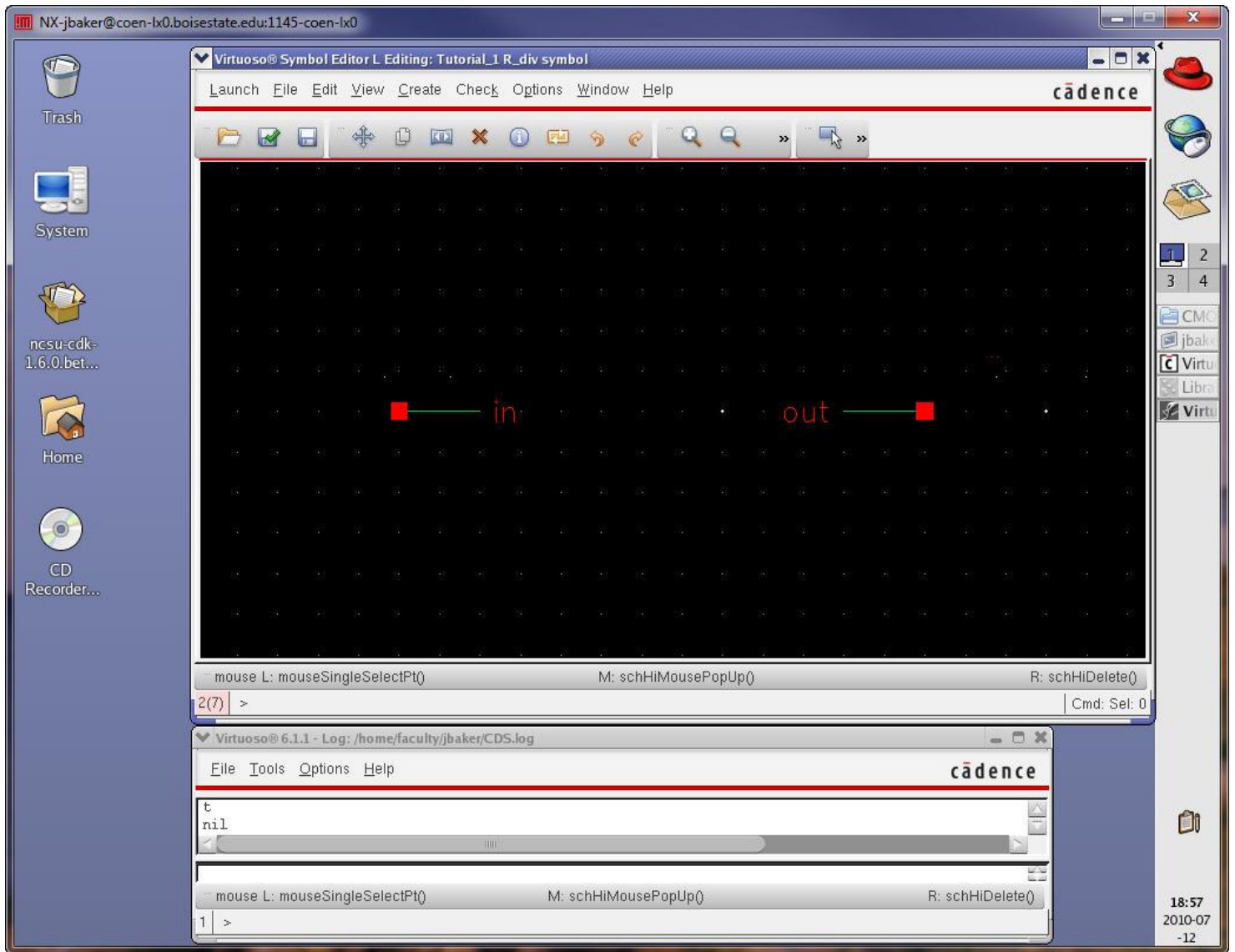




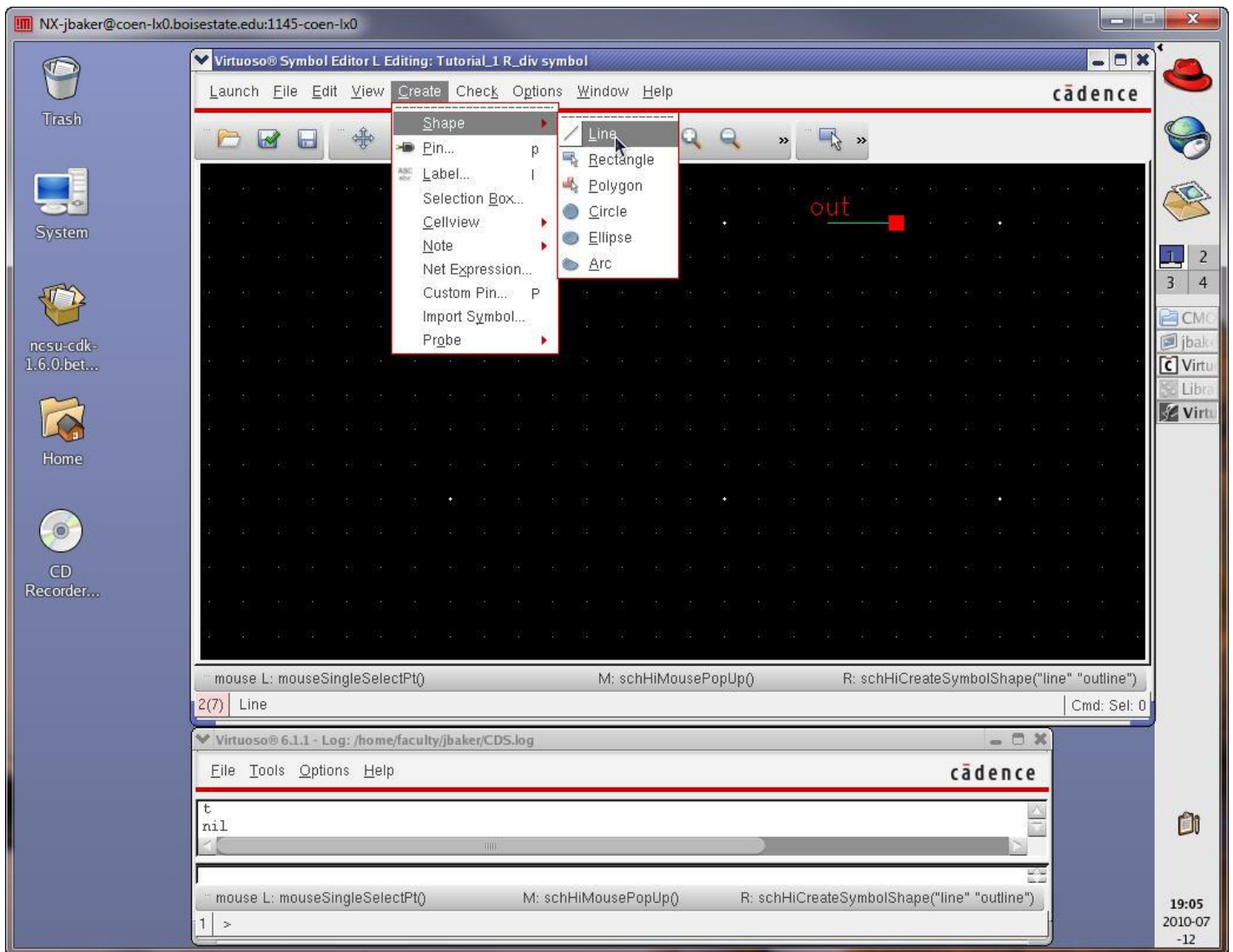
If we look at the Library Manager we see three Cell Views for the R\_div group remembering that the spectre\_state1 Cell View contains our simulation parameters which we load when the ADE is started as mentioned above.



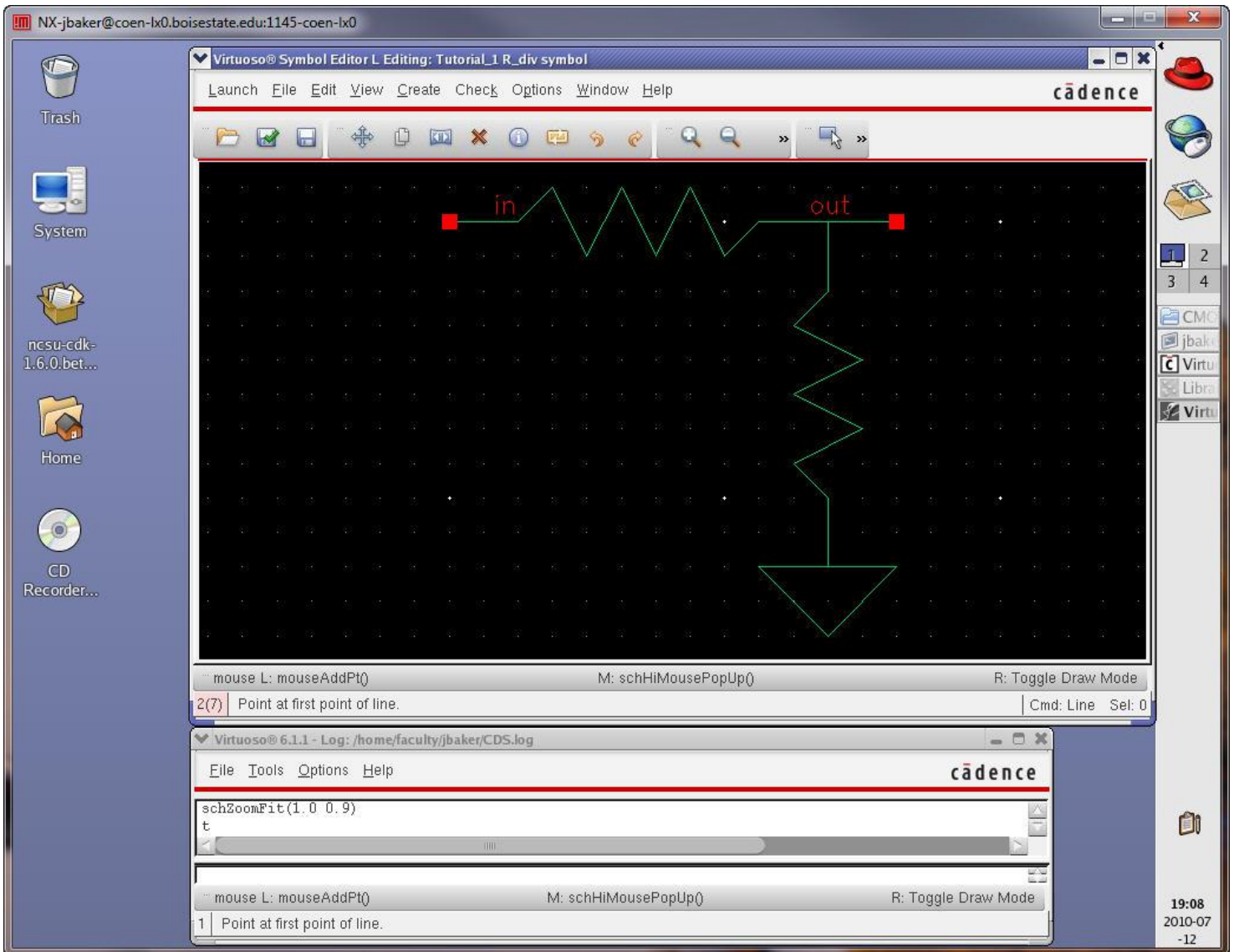
Let's delete everything in the symbol view except for the following. Remember that you can always use the undo command (u) if you accidentally delete something.



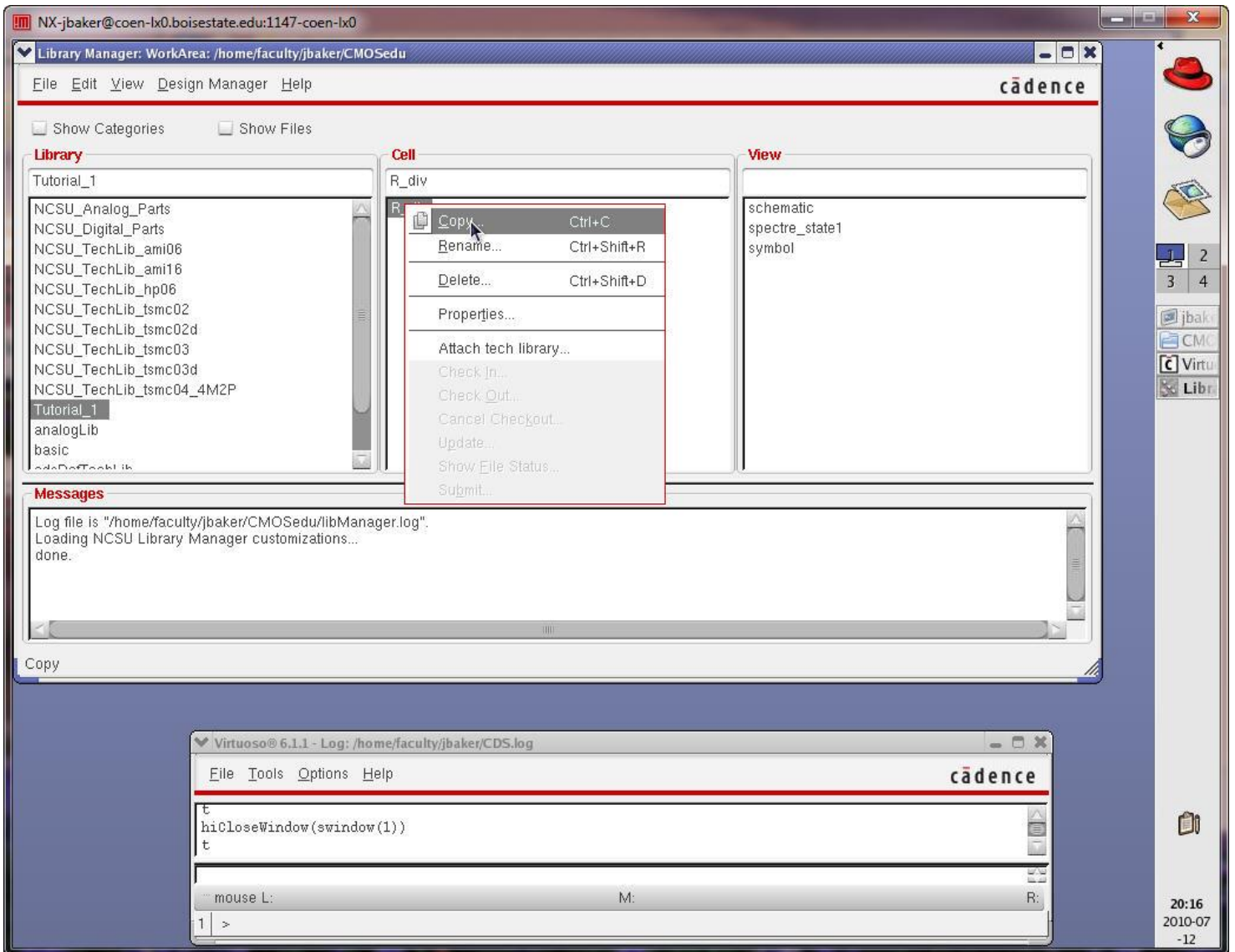
Move the text and then add a line.

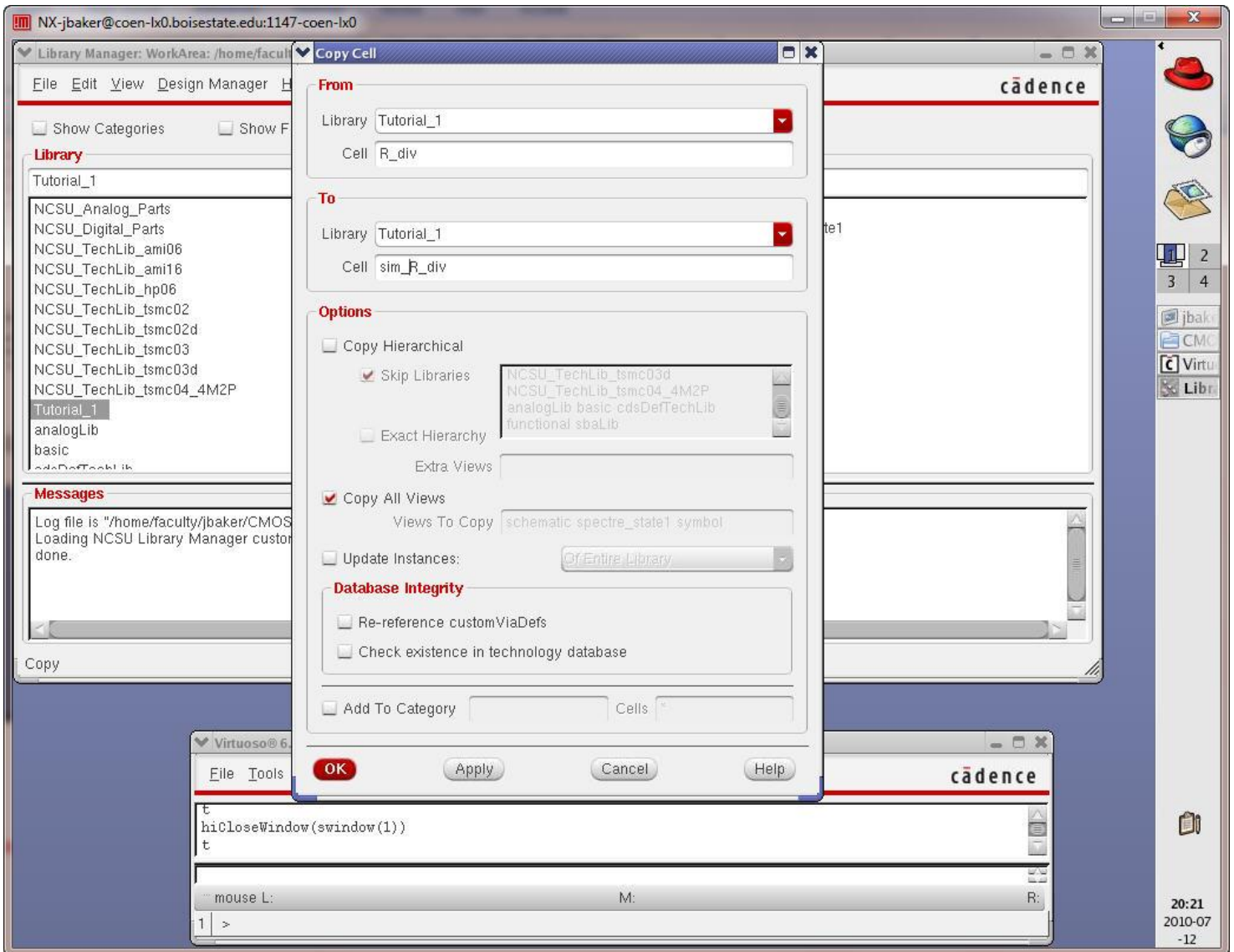


Draw the symbol seen below. Note that **Z** (zoom out by 2) and **f** (fit) can be useful when drawing the symbol.

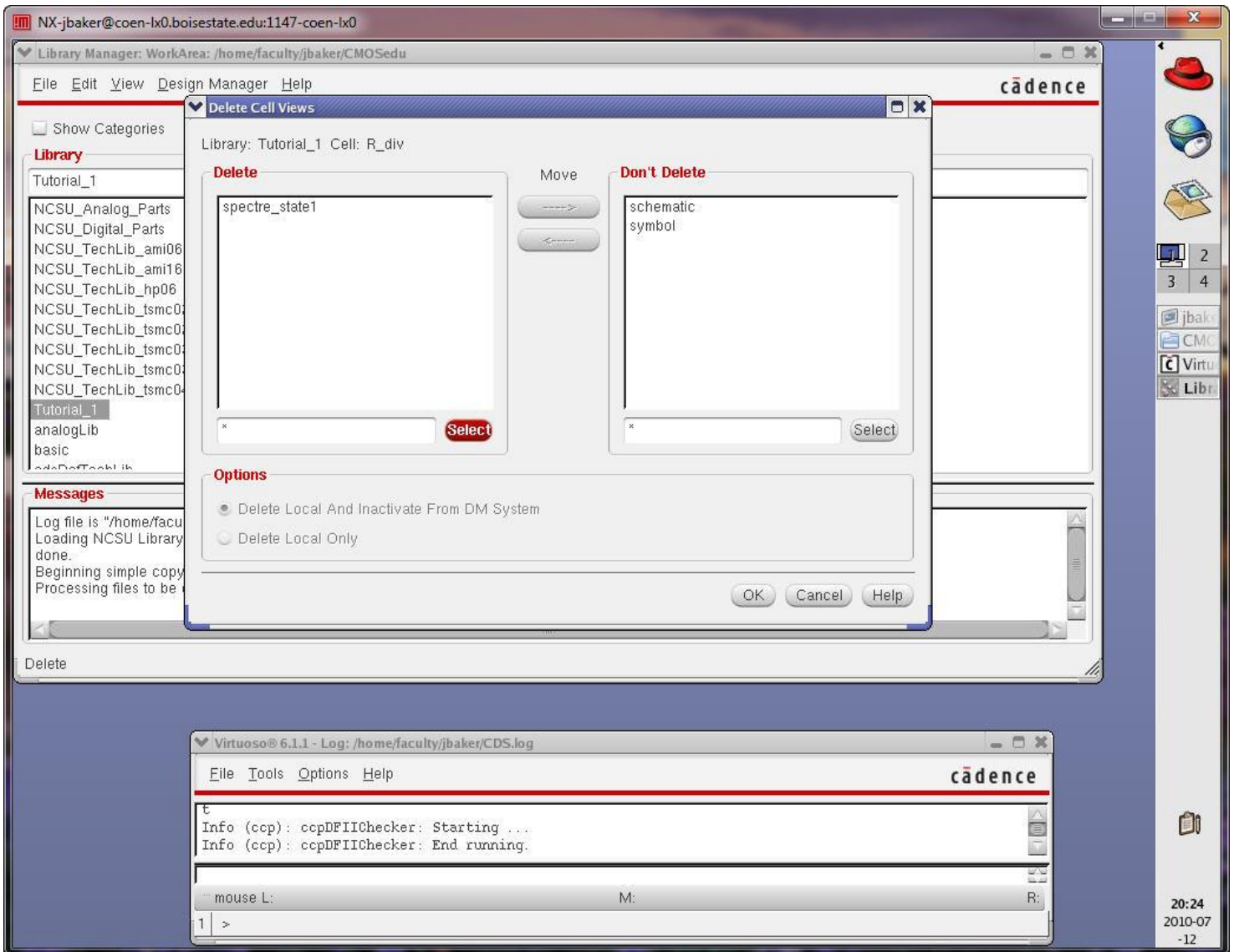


When finished Check and Save the symbol view. Close both the symbol and schematic views of the R\_div cell. Before doing the layout for the R\_div cell let's simulate the cell's operation again. To speed things up copy R\_div into another cell called sim\_R\_div (right click on the cell name).



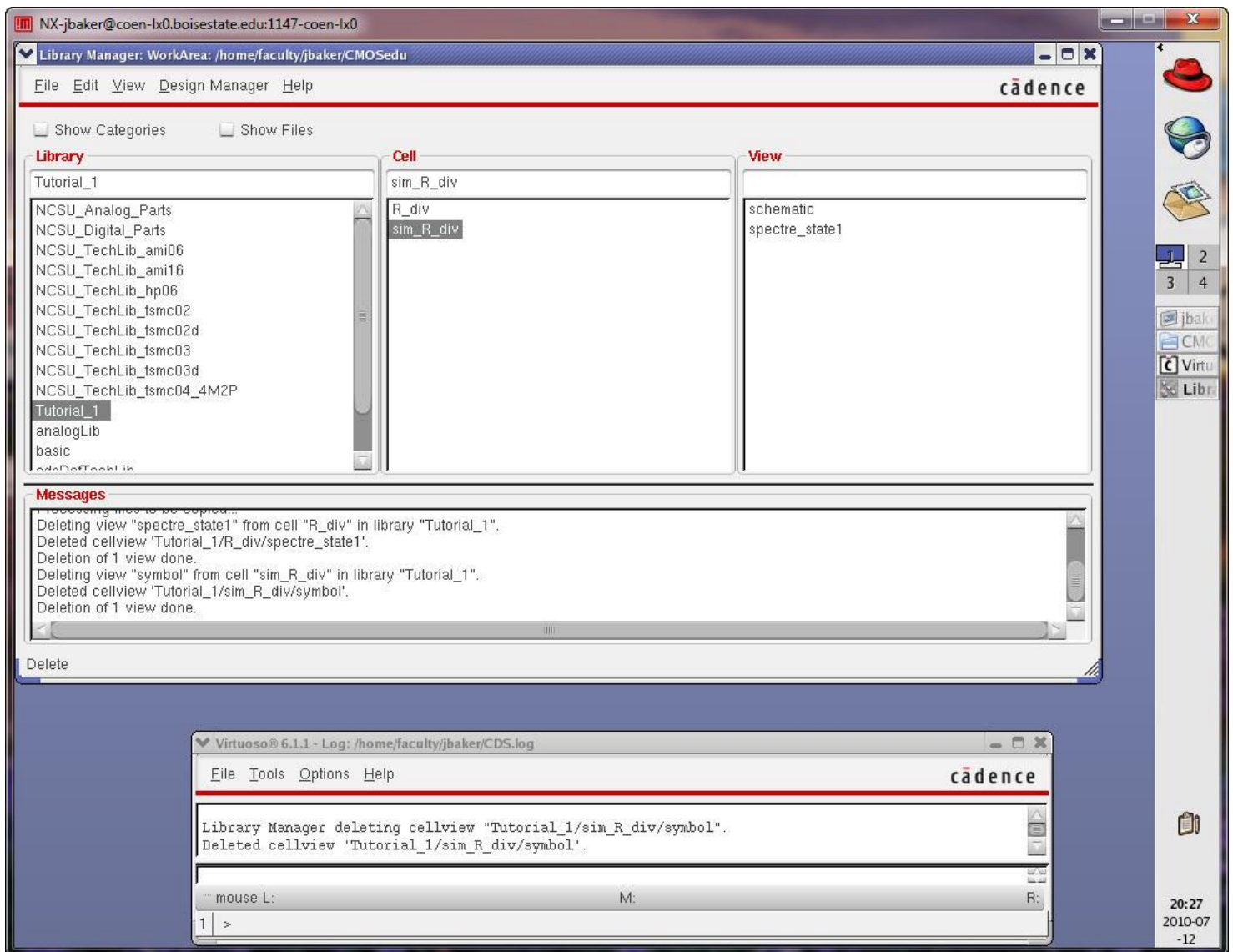


After hitting OK delete the spectre\_saved1 in the R\_div cell by right clicking on the name in the View category.

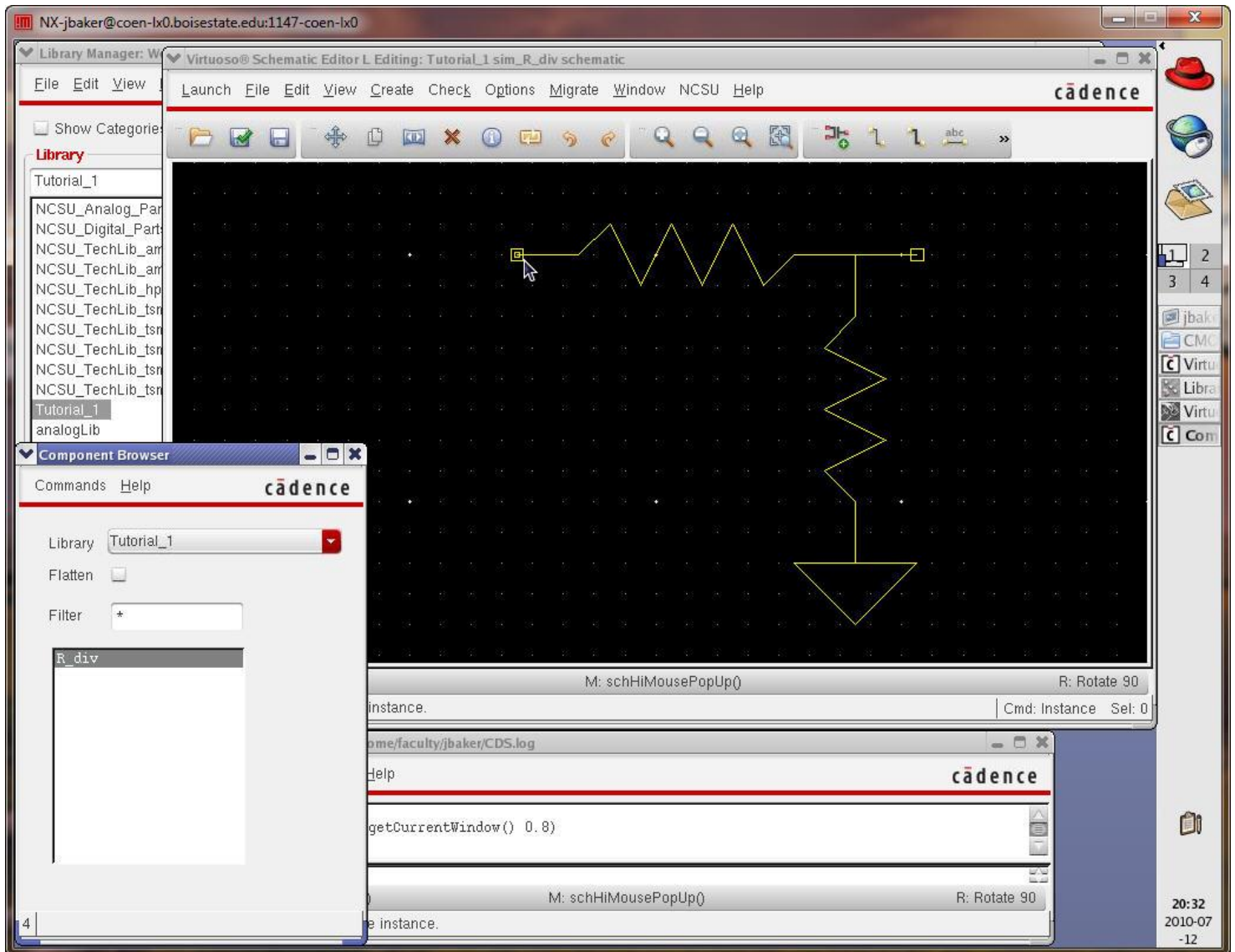


After hitting OK and Yes you are sure delete the symbol view in the sim\_R\_div cell.





Next open the schematic view of the sim\_R\_div cell and delete everything in the cell.  
 Press i to add the symbol for R\_div in the Tutorial\_1 library.  
 After you place the cell hit Esc to leave the place instance command mode.



Next add wires, wire names, and the input voltage like we did above.  
Set the input voltage to 1 V DC.

The screenshot displays the Cadence Virtuoso Schematic Editor interface. The main window shows a schematic diagram of a circuit. The circuit includes a DC voltage source labeled  $V_0$  with a value of  $vdc=1$ , connected to an input terminal labeled  $in$ . The circuit consists of two resistors connected in series. The output terminal is labeled  $out$ . The ground symbol is labeled  $gnd$ .

The console window at the bottom shows the following output:

```

t
showClickInfo()
window:3

mouse L: showClickInfo()      M: schHiMousePopUp()      R: schHiObjectProperty()
2(3) >
Cmd: Sel: 0

Virtuoso@ 6.1.1 - Log: /home/faculty/jbaker/CDS.log
File Tools Options Help
cādence

t
showClickInfo()
window:3

mouse L: showClickInfo()      M: schHiMousePopUp()      R: schHiObjectProperty()
1 >

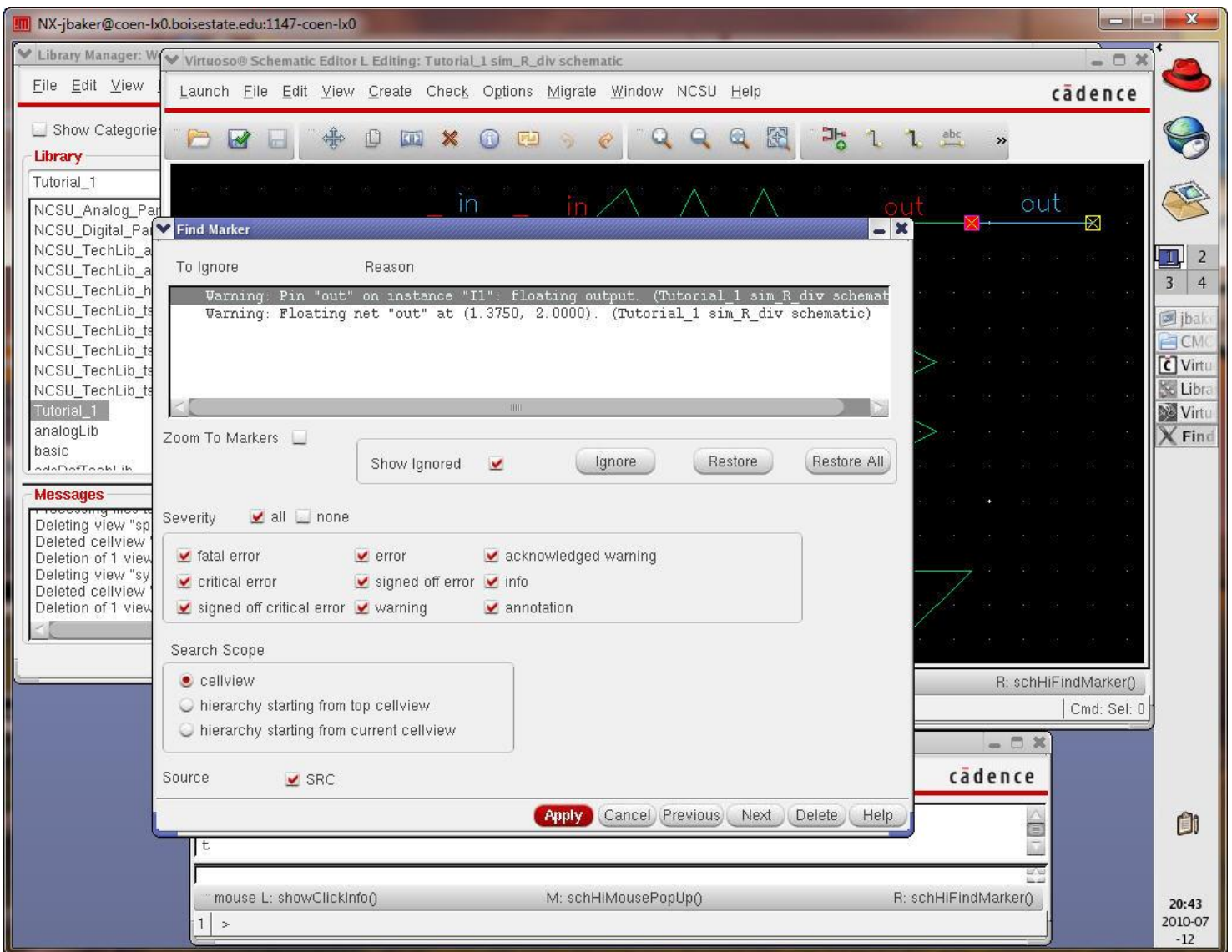
```

The console output indicates that two warnings were generated during the simulation process.

When finished Check and Save the schematic.  
Note we get two warnings.

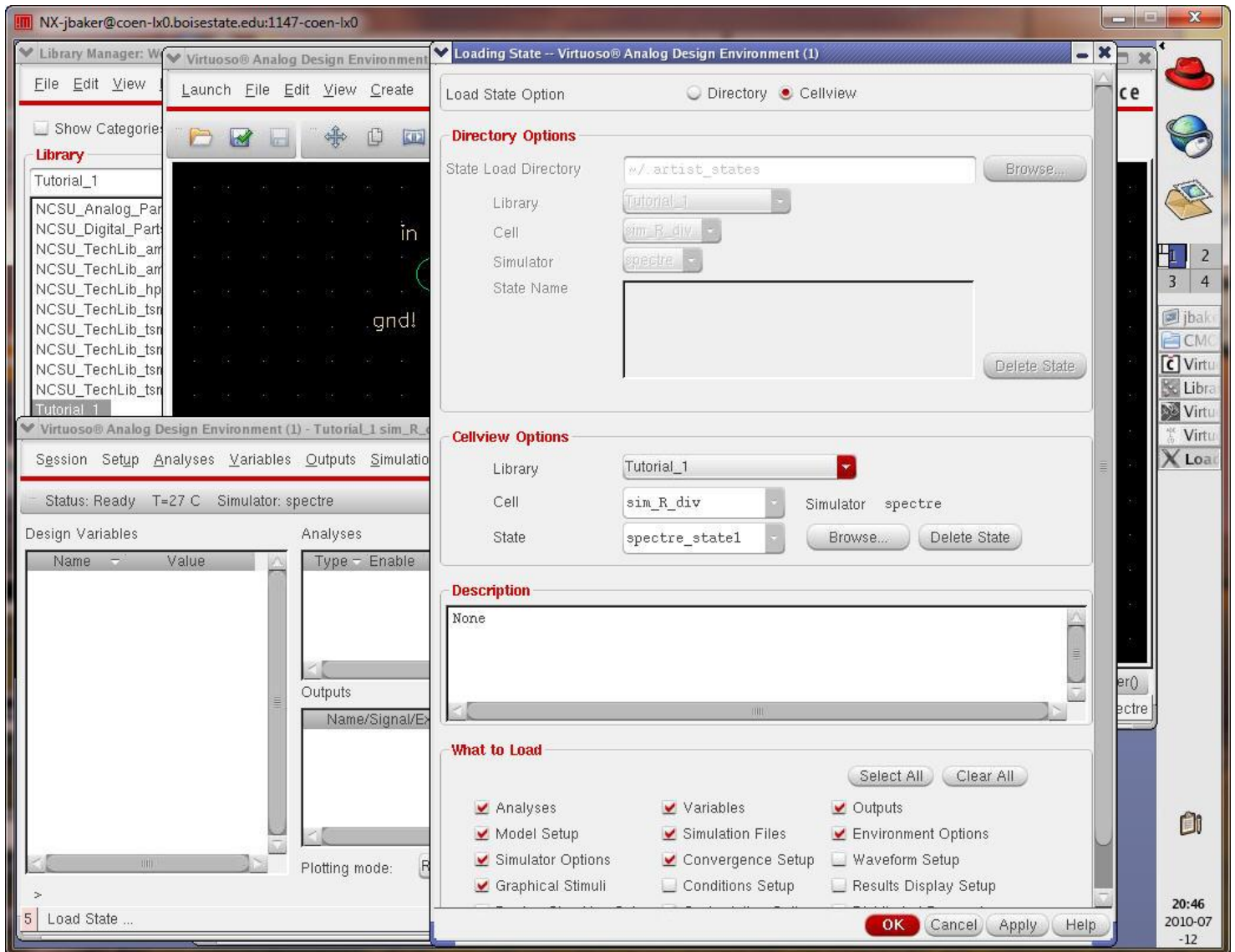
The screenshot shows the Cadence Virtuoso Schematic Editor interface. The main window displays a schematic diagram with a voltage source labeled 'vdc=1' connected to a network of resistors. The input nodes are labeled 'in' and the output nodes are labeled 'out'. A floating wire, also labeled 'out', is visible on the right side of the schematic. A 'Schematic Check' dialog box is open in the center, displaying a warning icon and the message: 'There were 0 errors and 2 warnings found.' Below the dialog box, the command line shows 'mouse L: showClickInfo()' and 'M: schHiMousePopUp()'. At the bottom of the screen, a log window titled 'Virtuoso@ 6.1.1 - Log: /home/faculty/jbaker/CDS.log' displays the following warning: 'Warning: Floating net "out" at (1.3750, 2.0000). There were 0 errors and 2 warnings found in "Tutorial\_1 sim\_R\_div schematic".' The system clock in the bottom right corner shows 20:40 on 2010-07-12.

The warnings are from the floating wire named out.  
Go to the menu items Check -> Find Marker to see the details.



We are okay with this wire floating so press ignore twice and then close the Find Marker window. Check and Save the schematic again. There should be no warnings or errors.

Now let's simulate this circuit. Launch the ADE and then load the state (Cellview...important).



While the type of analyses will be remembered the outputs to be plotted won't be remembered. Use the menu items Outputs -> To Be Plotted -> Select on Schematic to select the in and out wire nodes. Next use the menu items Session -> Save State to save the state. Hitting the green button to Netlist and Simulate starts the Spectre simulation.

The screenshot displays the Cadence Virtuoso Analog Design Environment. The left window shows the simulation results for a transient analysis (tran). The right window shows a schematic diagram with two output nodes labeled 'out' and a corresponding plot window titled 'Transient Response' showing a square wave signal over time.

**Simulation Statistics:**

tran: time	(%)	step	(%)
91 ms	9.1 %	20 ms	(2 %)
131 ms	13.1 %	20 ms	(2 %)
191 ms	19.1 %	20 ms	(2 %)
231 ms	23.1 %	20 ms	(2 %)
291 ms	29.1 %	20 ms	(2 %)
331 ms	33.1 %	20 ms	(2 %)
391 ms	39.1 %	20 ms	(2 %)
431 ms	43.1 %	20 ms	(2 %)
491 ms	49.1 %	20 ms	(2 %)
531 ms	53.1 %	20 ms	(2 %)
591 ms	59.1 %	20 ms	(2 %)
631 ms	63.1 %	20 ms	(2 %)
691 ms	69.1 %	20 ms	(2 %)
731 ms	73.1 %	20 ms	(2 %)
791 ms	79.1 %	20 ms	(2 %)
831 ms	83.1 %	20 ms	(2 %)
891 ms	89.1 %	20 ms	(2 %)
931 ms	93.1 %	20 ms	(2 %)
985.5 ms	98.6 %	14.5 ms	(1.45 %)

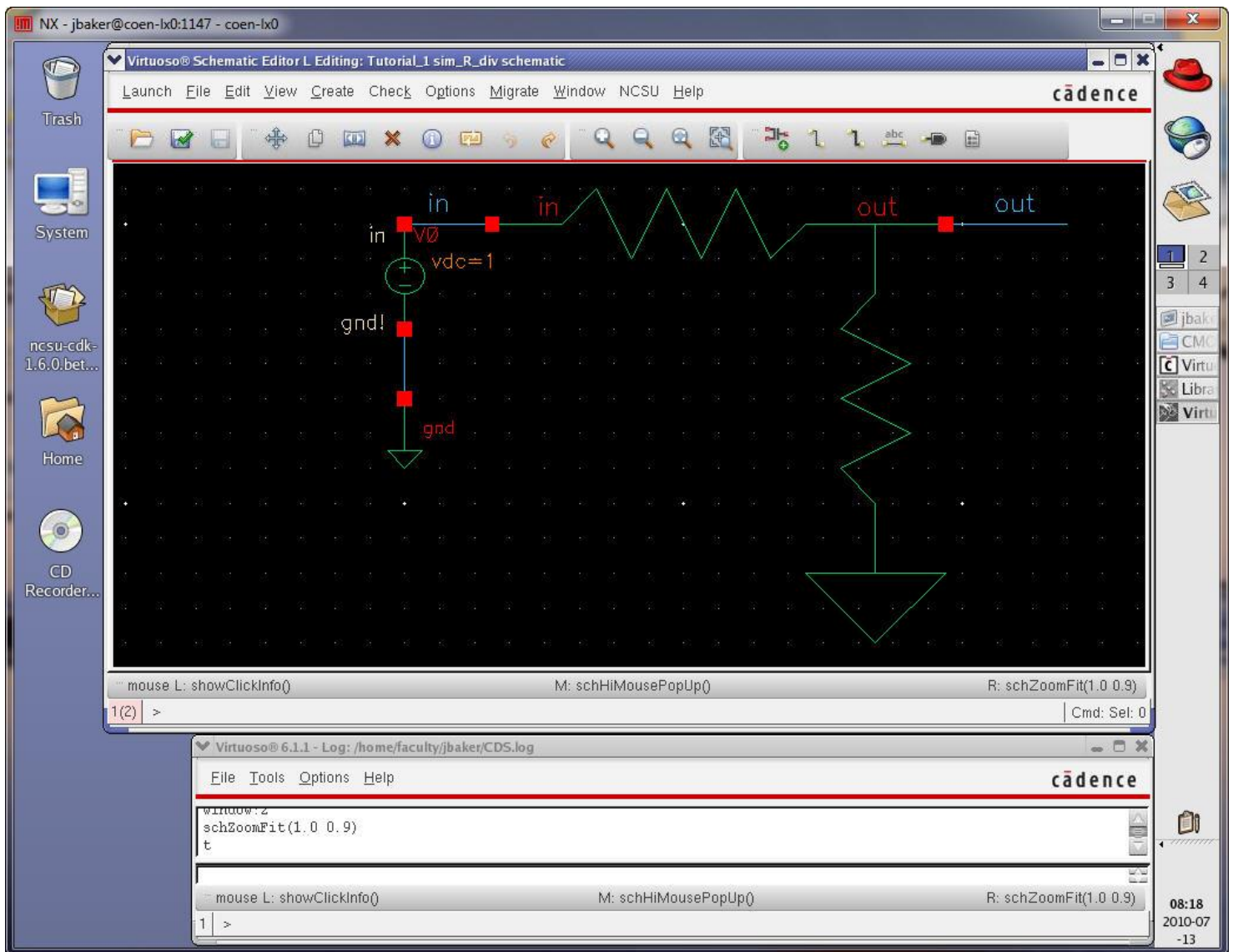
Number of accepted tran steps = 54.  
Accumulated tran full load time = 0 s.  
Accumulated tran full component evaluation time = 0 s.  
Accumulated tran full preload time = 0 s.  
Accumulated tran full merge time = 0 s.  
Accumulated tran residue-only load time = 0 s.  
Accumulated tran residue-only component evaluation time = 0 s.  
Accumulated tran residue-only preload time = 0 s.  
Accumulated tran residue-only merge time = 0 s.  
Accumulated tran factor time = 0 s.  
Accumulated tran solve time = 0 s.  
Accumulated tran output time = 0 s.  
Initial condition solution time = 0 s.  
Intrinsic tran analysis time = 0 s.  
Total time required for tran analysis 'tran' was 0 s.

finalTimeOP: writing operating point information to rawfile.  
modelParameter: writing model parameter values to rawfile.  
element: writing instance parameter values to rawfile.  
outputParameter: writing output parameter values to rawfile.  
designParamVals: writing netlist parameters to rawfile.  
primitives: writing primitives to rawfile.  
subckts: writing subcircuits to rawfile.

**Transient Response Plot:**

The plot shows the transient response of the circuit. The y-axis is labeled 'V (V)' and ranges from 4.0 to 1.1. The x-axis is labeled 'time (s)' and ranges from 0.0 to 1.0. The plot shows a square wave signal with a period of approximately 0.5 seconds. The signal is labeled '/in' (red) and '/out' (black). The plot is titled 'Transient Response' and has a legend showing '/in' (red) and '/out' (black). The plot is zoomed in to show the signal between 0.55062ms and 500mV.

Close the ADE.



Before moving on to the layout lets cover one last item, that is, descending through the hierarchy.

Select the R\_div symbol (left+click the mouse button on the symbol) and follow the menu items seen below (or just press **X**).

The contents of the symbol can be edited in the current tab (window) or in a new window. Select current tab and press OK.

Note, also, that you can return back up in the hierarchy by pressing **b** or the menu item seen below (but shaded since we are already at the top).

Descend down into the R\_div schematic and then back up to the sim\_R\_div schematic to get some experience with the commands.



The screenshot shows the Cadence Virtuoso Schematic Editor interface. The main window displays a schematic diagram of a resistive divider circuit. The circuit consists of a green zigzag line representing a resistor, with an input terminal labeled 'in' on the left and two output terminals labeled 'out' on the right. A menu is open over the circuit, showing options like 'Descend Edit...', 'Descend Read...', 'Edit In Place', and 'Show Scope'. The status bar at the bottom shows '1(2) Descend Edit... X'.

The interface includes a menu bar with 'Launch', 'File', 'Edit', 'View', 'Create', 'Check', 'Options', 'Migrate', 'Window', 'NCSU', and 'Help'. The 'Edit' menu is open, showing options like 'Undo', 'Redo', 'Move', 'Copy', 'Stretch', 'Delete', 'Rotate', 'Sheet Size...', 'Sheet Title...', 'Hierarchy', 'Properties', 'Net Expression', 'Select', 'Route Flight', 'Renummer Instances...', 'Component Display...', 'Alternate View', 'Update Pins From View...', 'Find...', and 'Replace...'. The 'Hierarchy' menu is also open, showing options like 'Descend Edit...', 'Descend Read...', 'Edit In Place', 'Show Scope', 'Return', and 'Return To Top'.

The status bar at the bottom shows 'mouse L: showClickInfo()', 'M: schHiMousePopUp()', and 'R: schZoomFit(1.0 0.9)'. The command line shows '1(2) Descend Edit... X'.

The bottom window shows the Virtuoso 6.1.1 log file, with the following content:

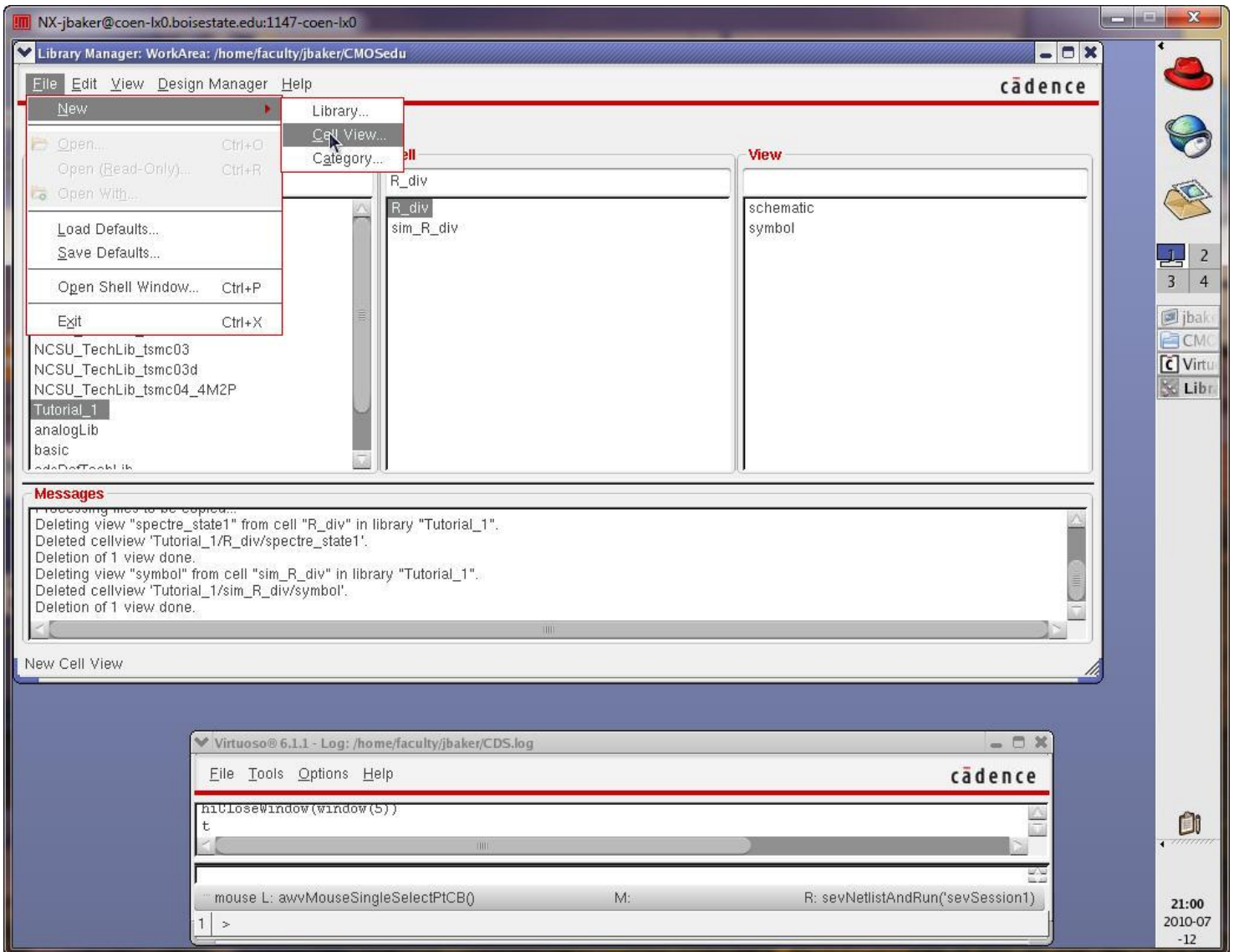
```

File Tools Options Help
showClickInfo()
R_div (instance "I1", library "Tutorial_1")
window:2
mouse L: showClickInfo()
M: schHiMousePopUp()
R: schZoomFit(1.0 0.9)
1 >

```

The system tray on the right shows the date and time: 08:21 2010-07 -13.

We are now ready to lay out the resistive divider.  
To begin make a new Cell View for the layout of R\_div.



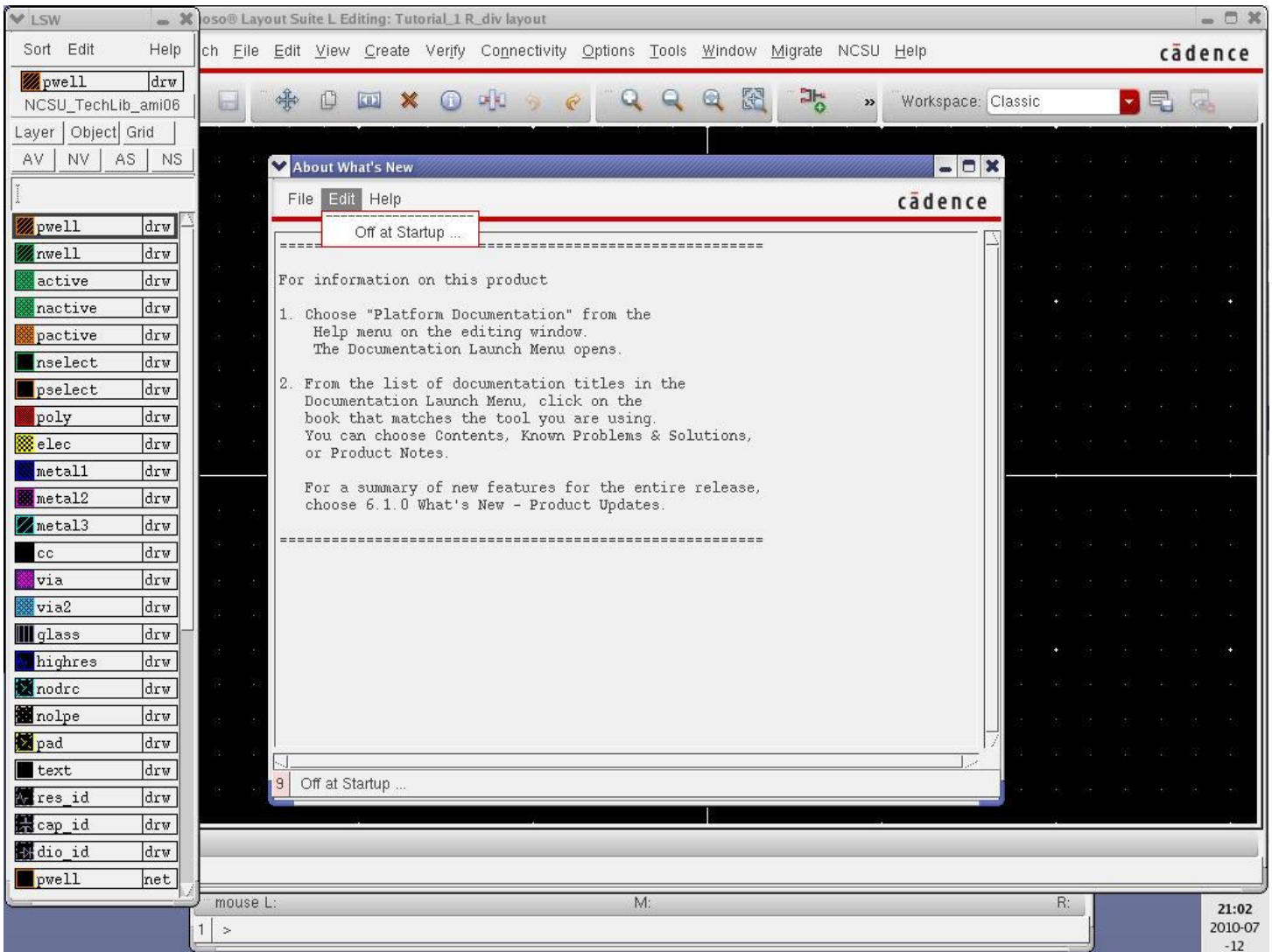
After creating the layout cell view the following will appear.

Select Off at Startup to ensure you don't see this What's New information each time you start-up a layout session.

Note the Layout Selection Window (LSW) that allows you to select specific layers when doing layout.

In the top of the LSW you can select AV (all layers visible) or NV (only the layer selected is visible). After making this selection follow it by re-drawing the layout window (View -> Redraw) to see the results. AS and NS are used in a similar manner to allow selecting or not selecting specific layers.

\*\*\*Note, below, that we use the **drw** layers to draw layouts not the **net** layers, see the top layer is "pwell drw" and the bottom layer is "pwell net." It's easy to draw shapes, accidentally, using a net layer so if you have an issue with a layout you think should be fine check to see that you haven't done this.\*\*\*



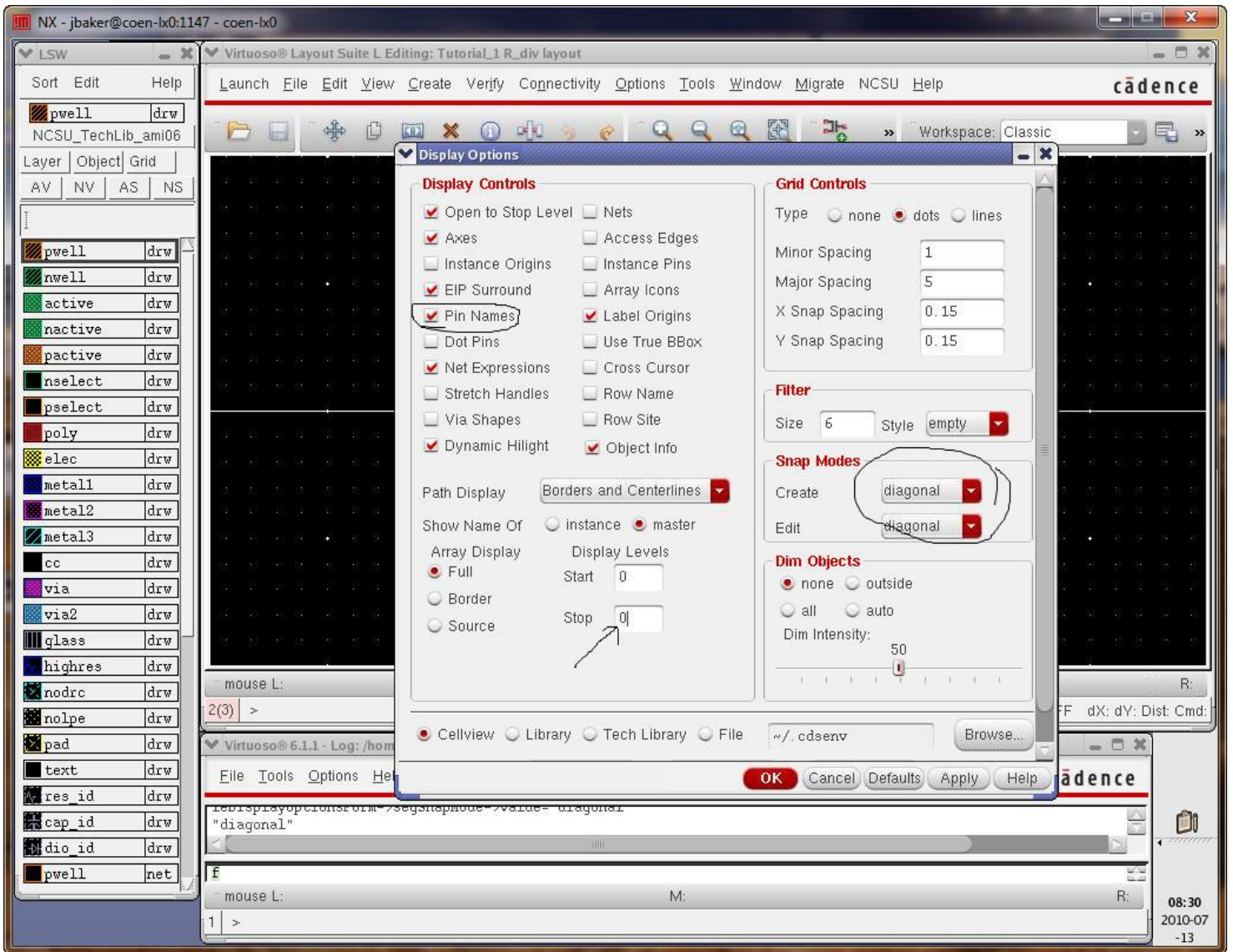
Next go to the menu items Options -> Display (or just press e).

Set the display so that Pin Names are shown.

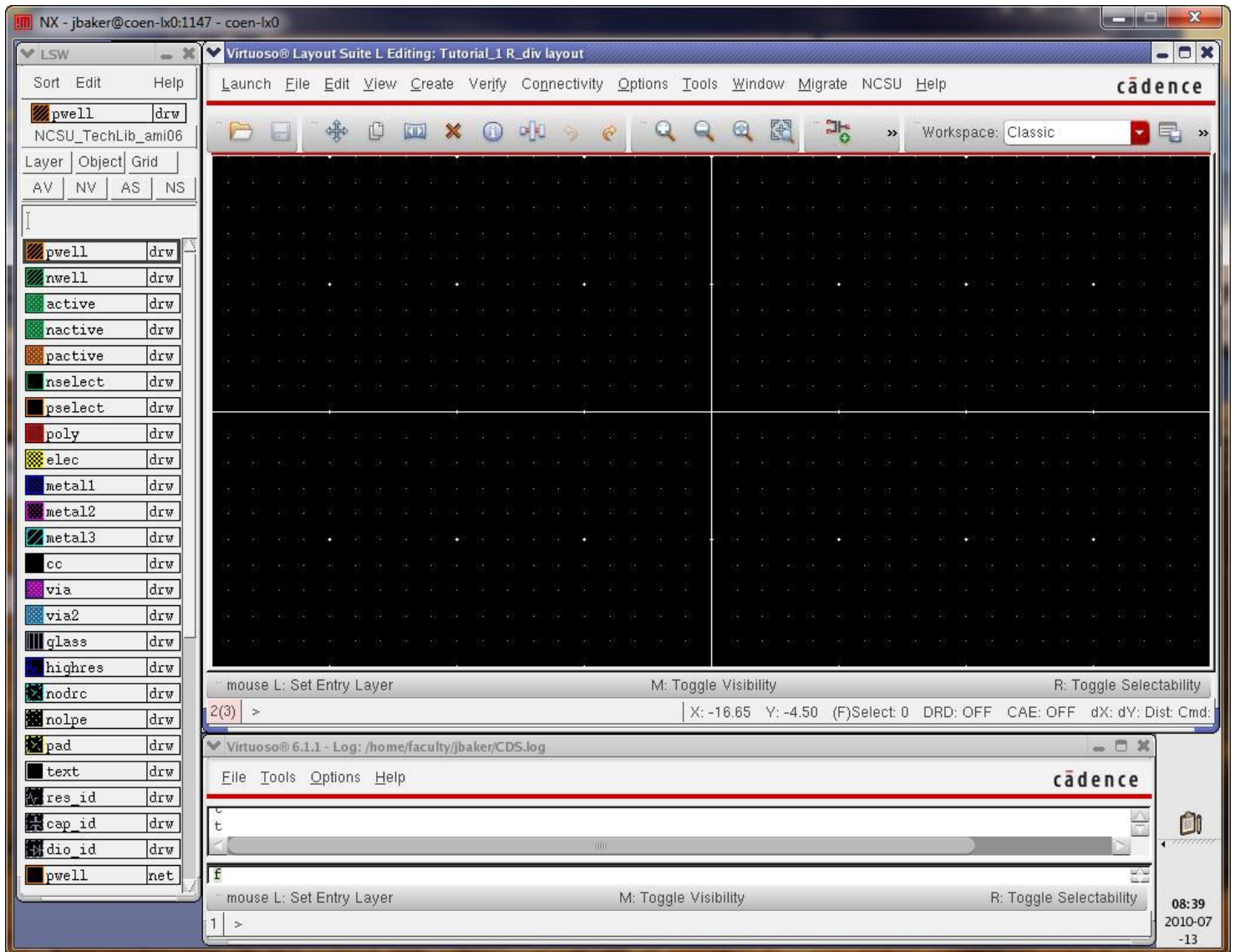
Change the Snap Modes to diagonal.

Note that the Display Levels has no depth, that is, the layout of a cell placed in another cell will show as an outline. To see the contents we need to increase

the Stop level. We'll do this later just so we can see this window again and what an outline of a cell looks like.



We are now ready to draw some shapes.



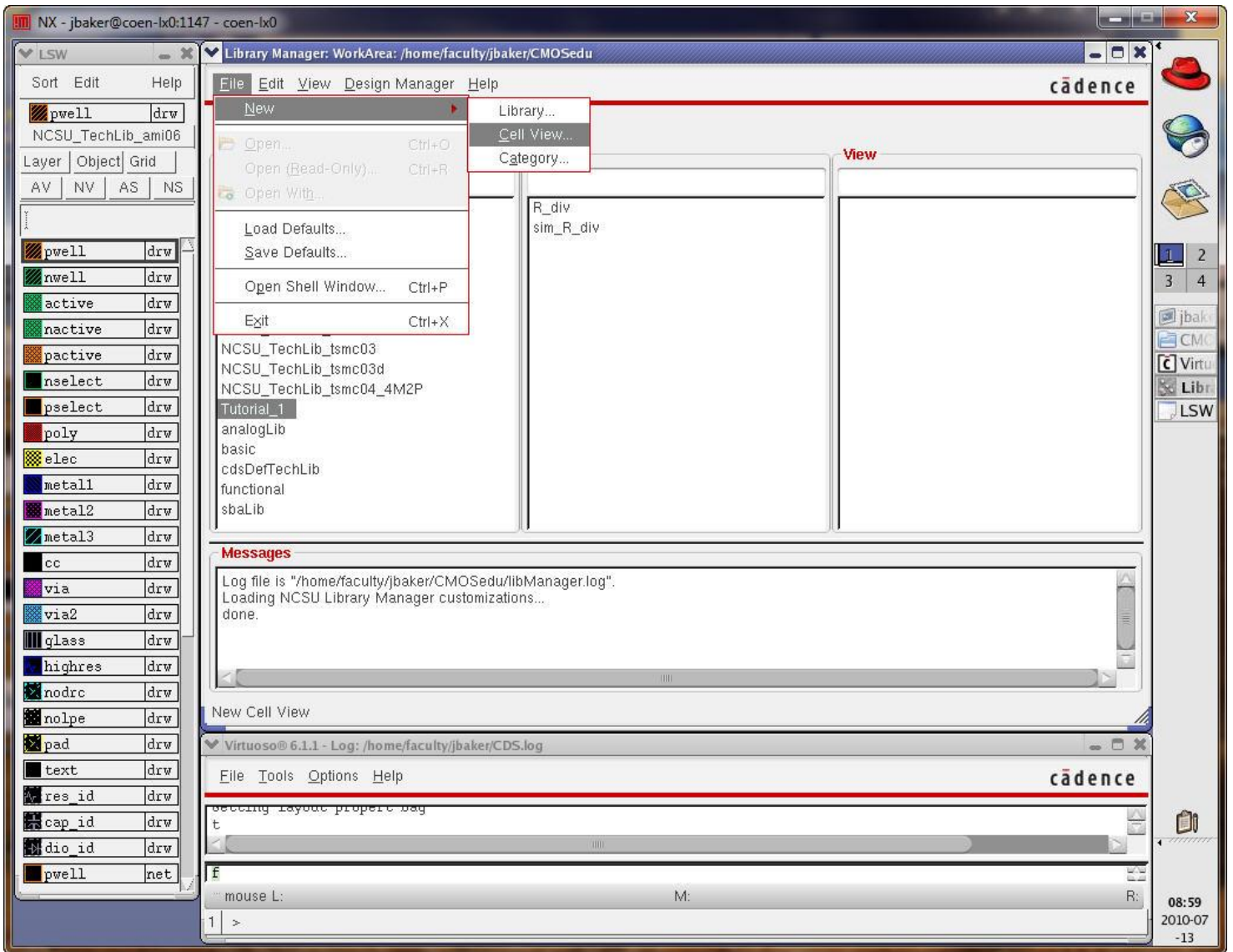
When we started drawing the schematic for the R\_div cell the first thing we did was press i and place the symbol for a resistor. Here we don't have a layout for the resistor so we need to create one!

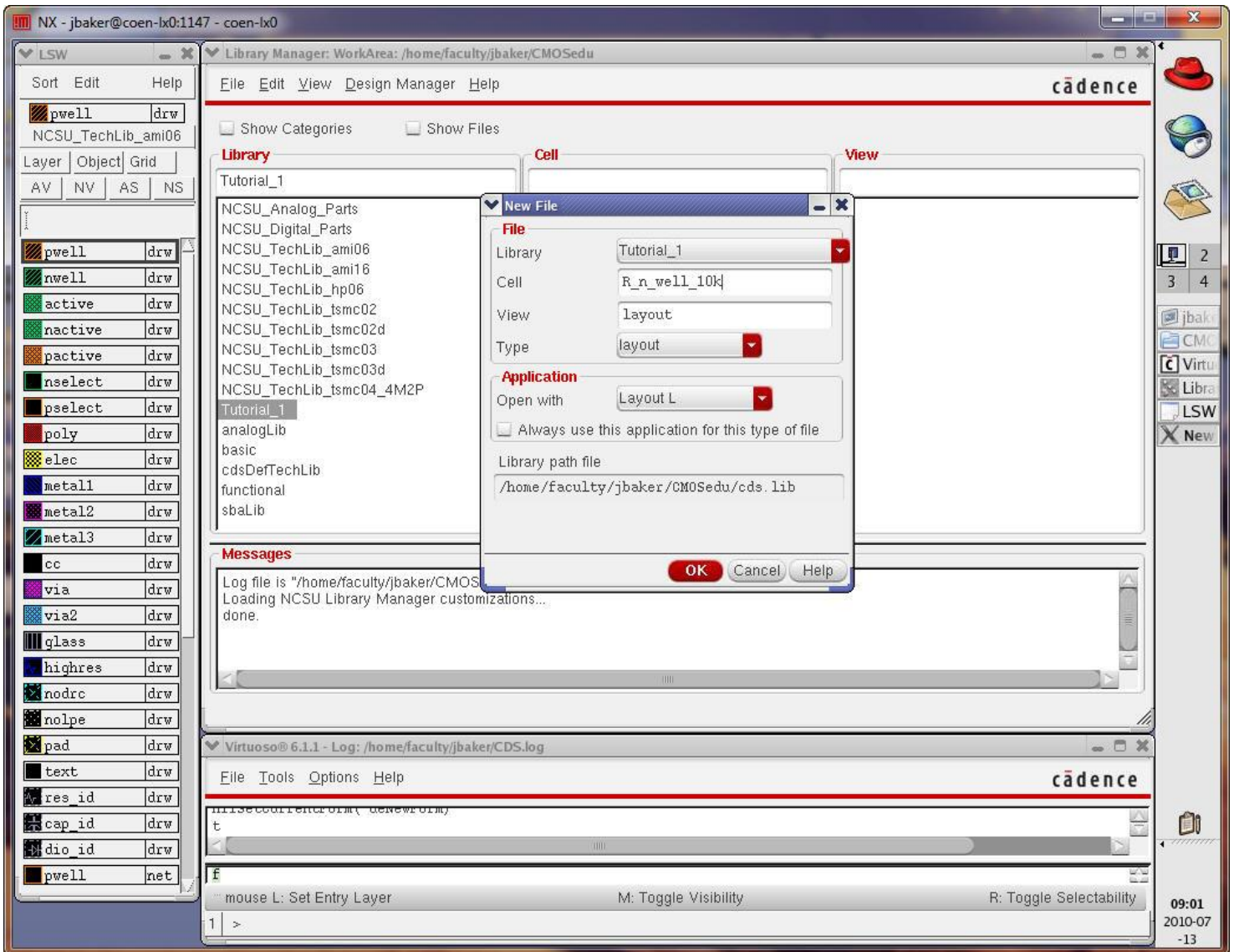
We'll use the n-well layer for the 10k resistor.

The sheet resistance of n-well in the C5 process is roughly 800 ohms.

The minimum width of n-well is 12 lambda (3.6 microns since lambda here is 300 nm) so let's make a 10k resistor using a width of 4.5 um and a length 56 um.

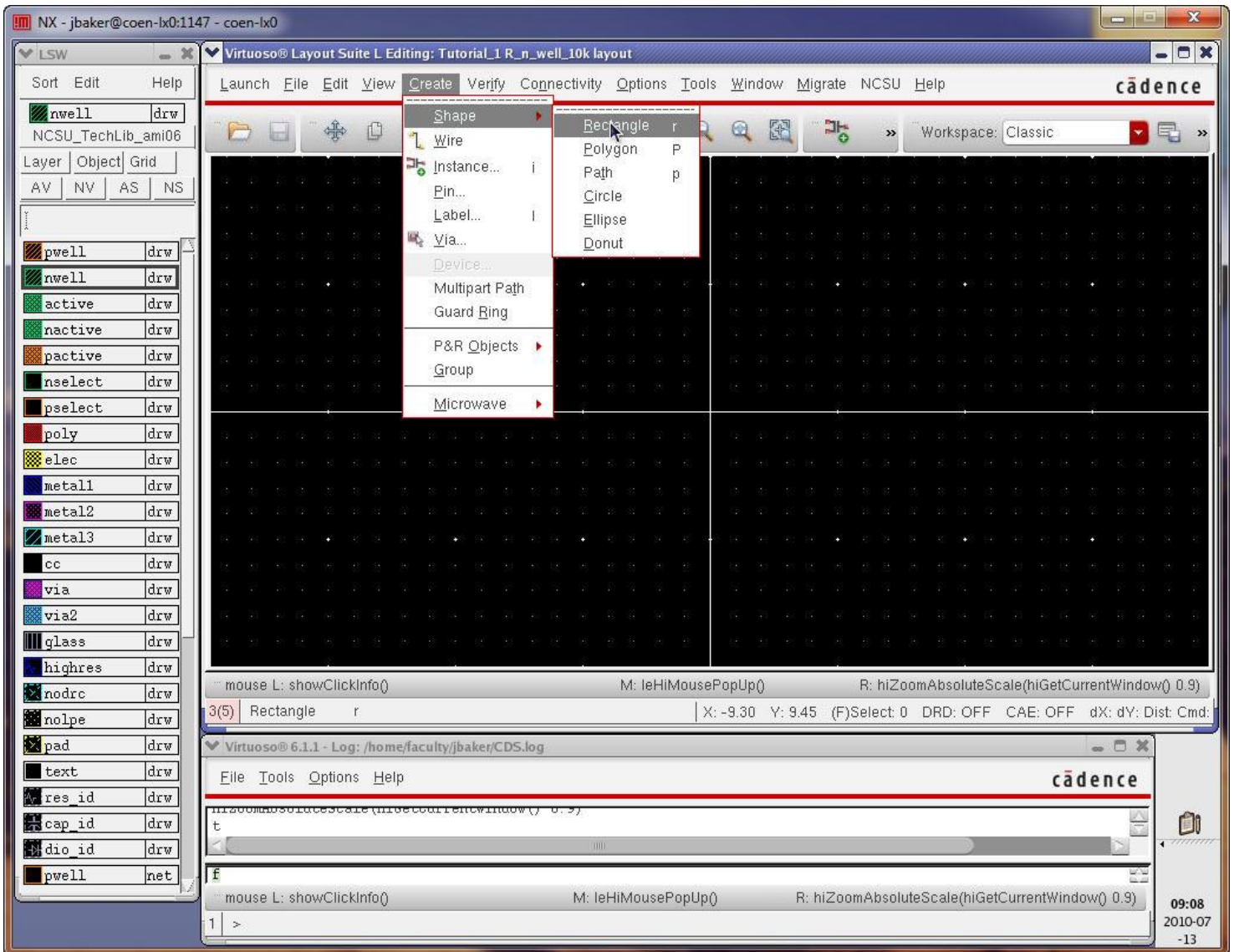
Create a cell (layout view) called R\_n\_well\_10k.





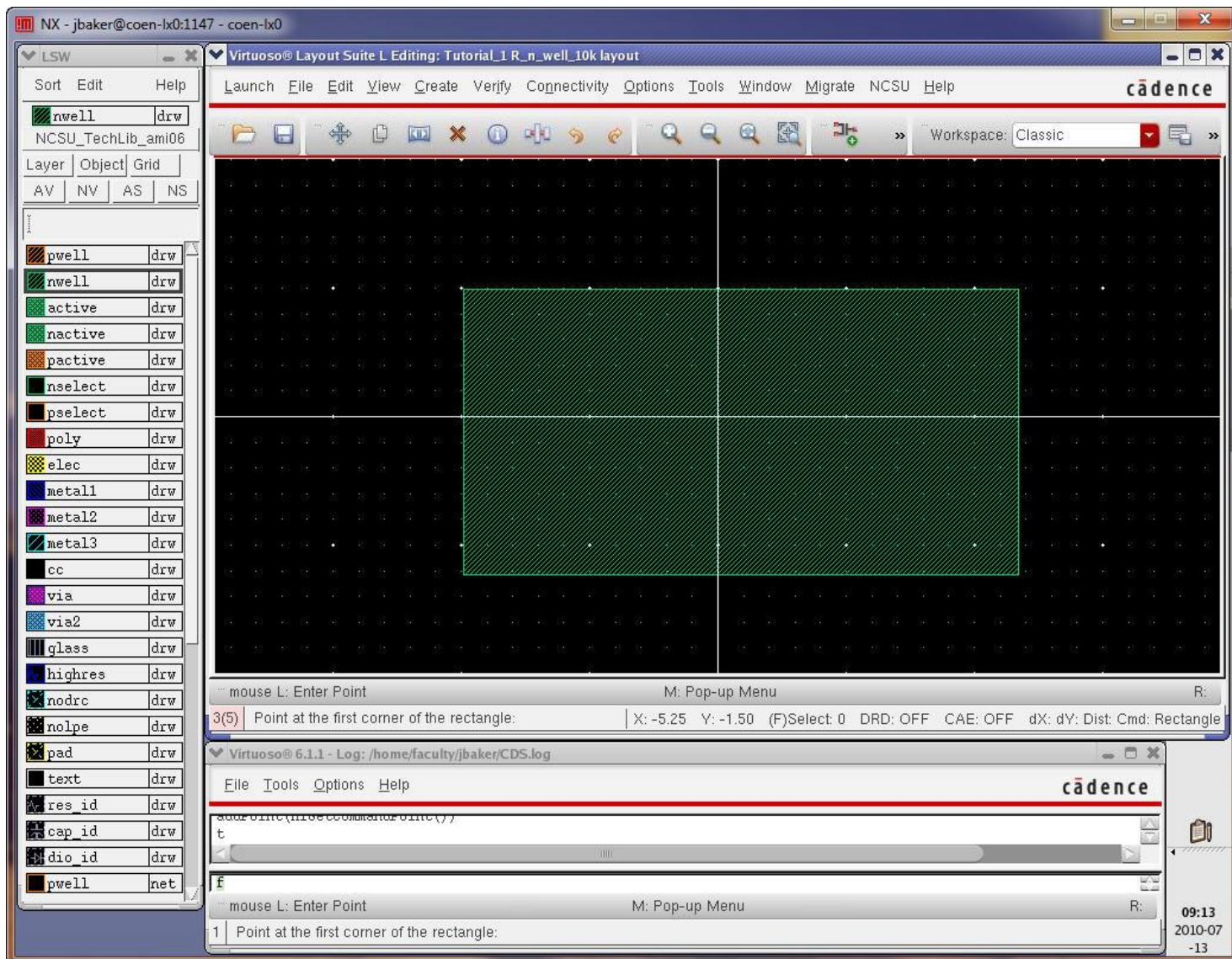
Select n-well in the Layer Selection Window (LSW).

Next create a rectangle (this will be the resistor with a width of 4.5 um and a length of 56 um).



At this point don't worry about the size. Click once to start drawing the rectangle then, after moving the mouse, click again to finish the drawing.

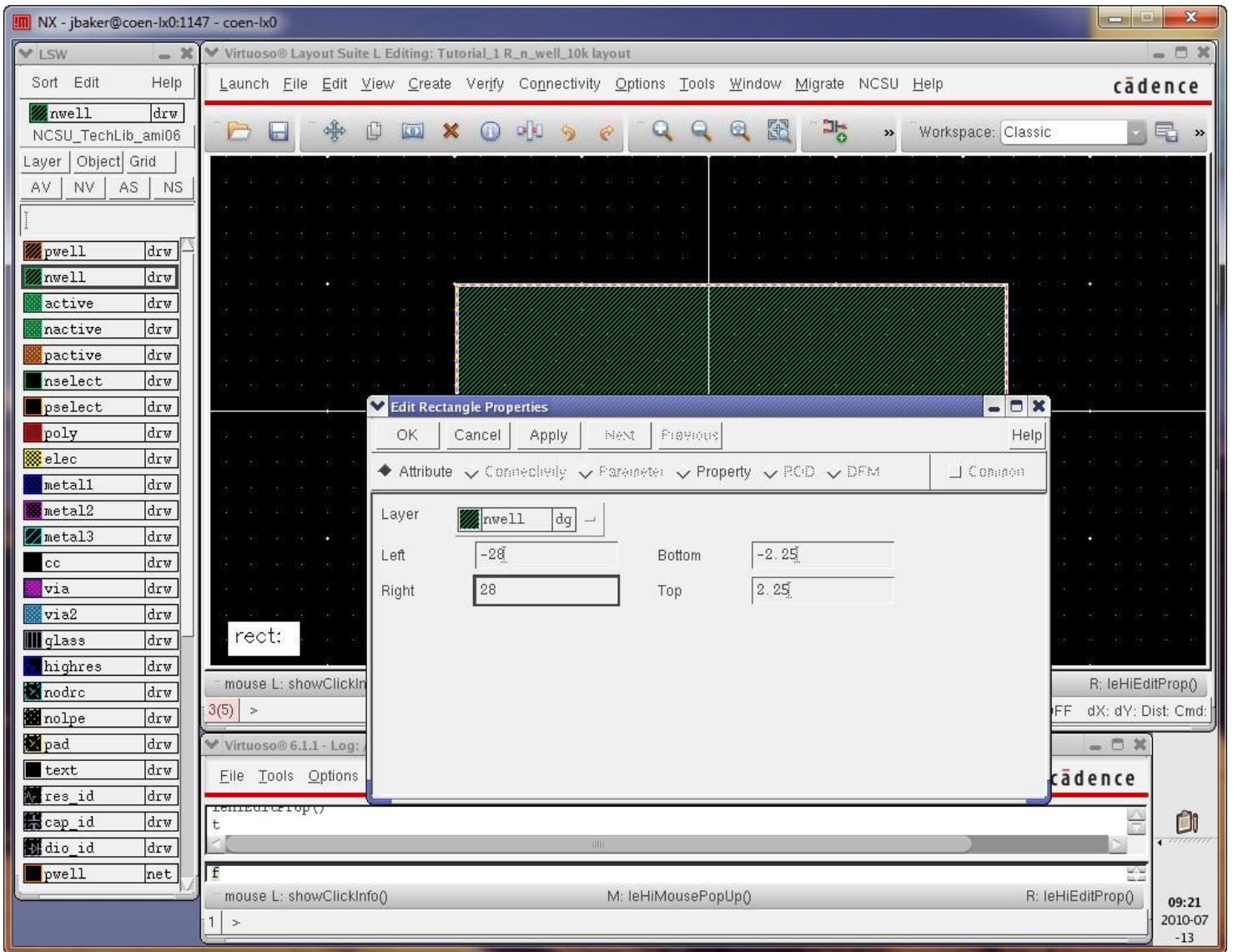




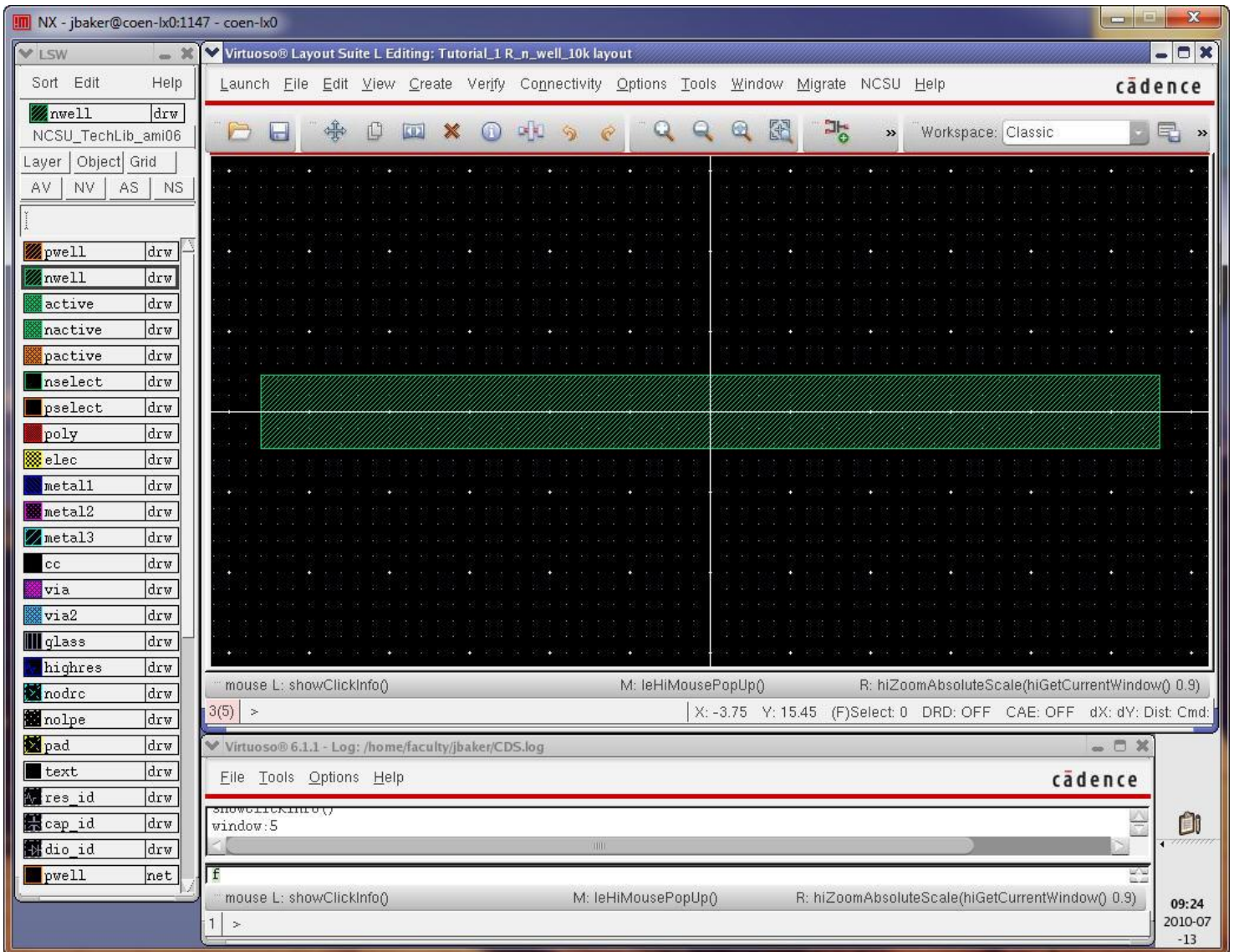
To exit the Create Rectangle mode press Esc (or Virtuoso will continue drawing rectangles).

Next select the rectangle and press q (Edit -> Basic -> Properties).

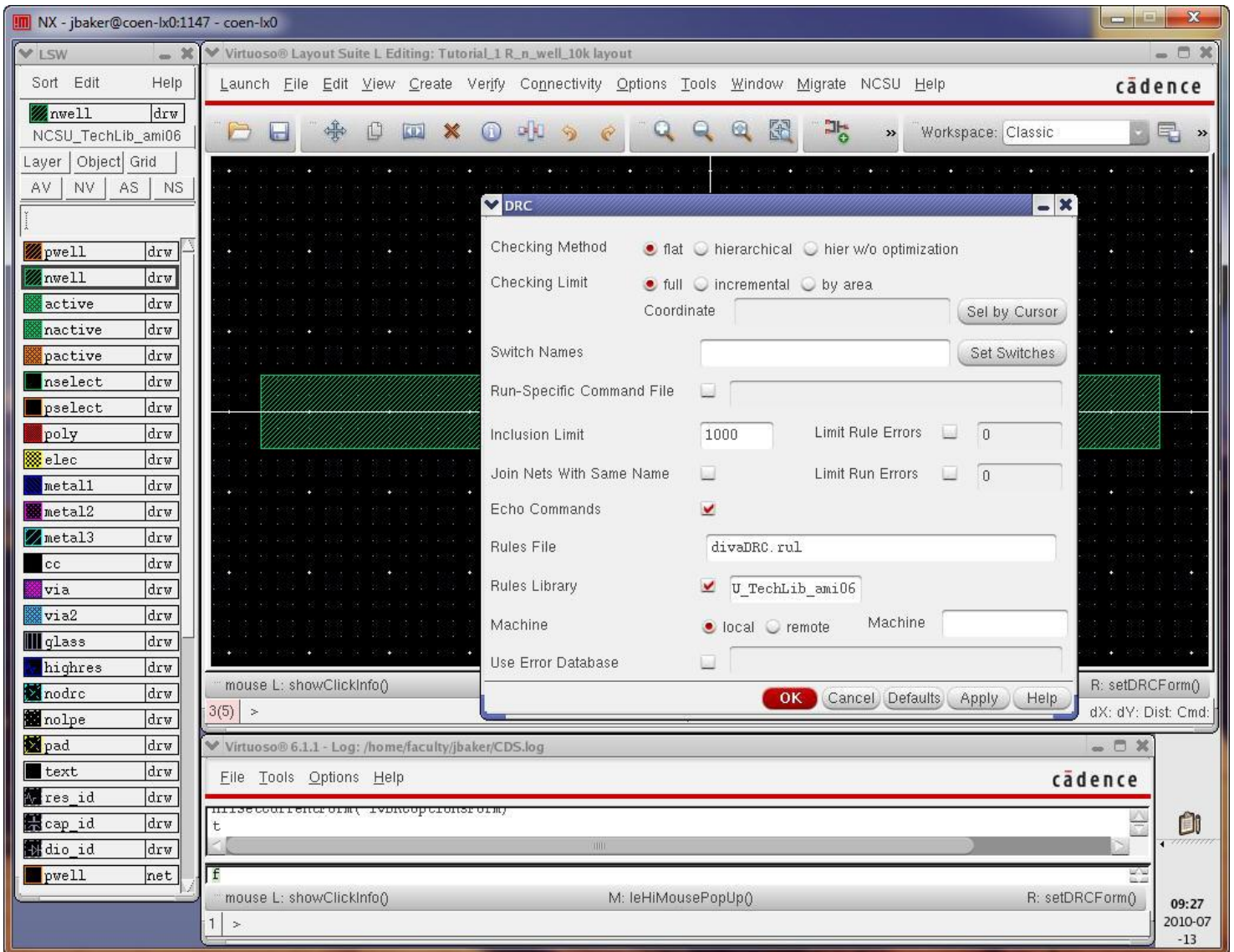
As calculated above we want a resistor that is 56 um long and 4.5 um wide.



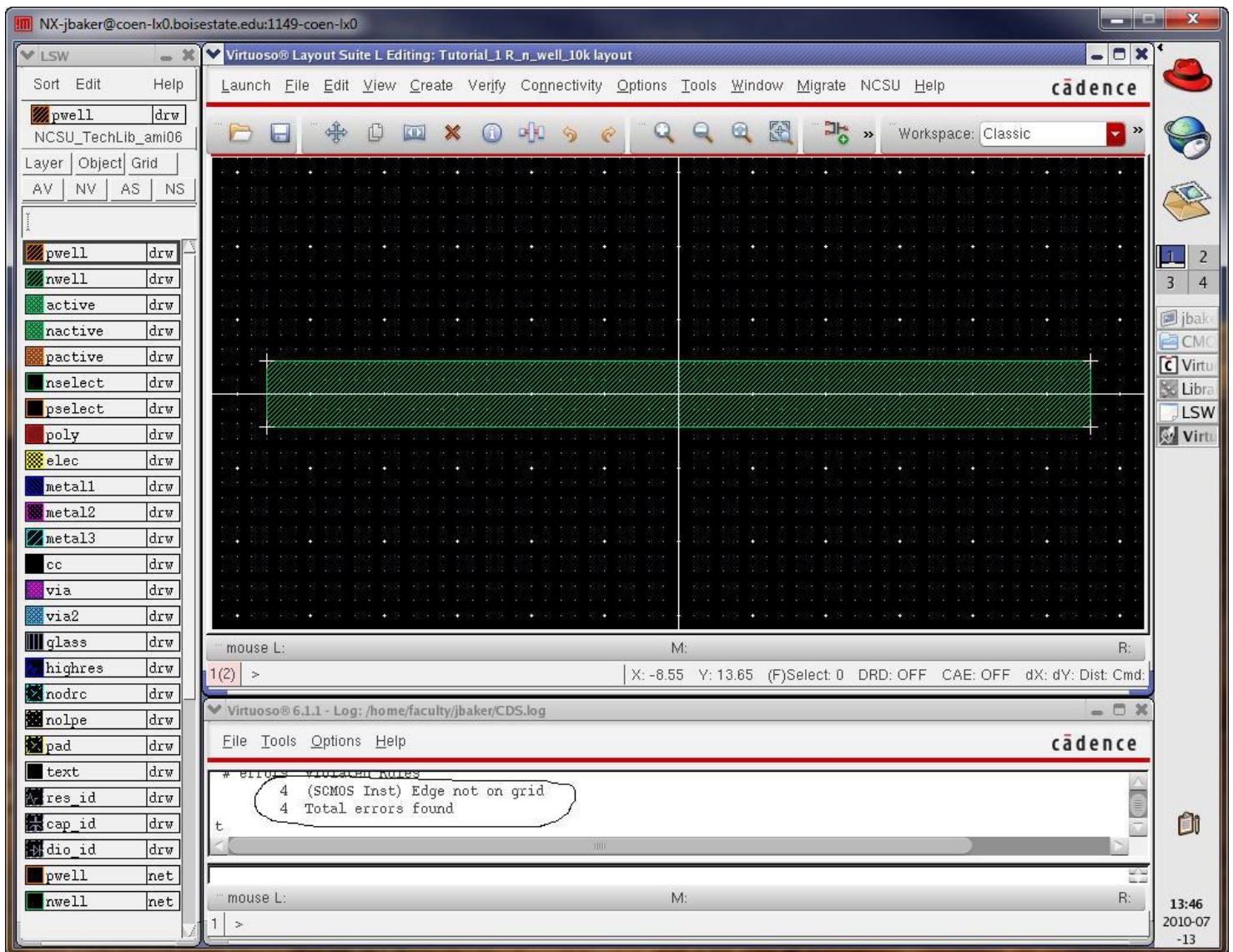
After clicking on OK and fitting the layout we get the following.



Let's design rule check (DRC) this layout before continuing.  
 Using the menu Verify -> DRC the following window pops up.  
 The Rules Library field seen below is NCSU\_TechLib\_ami06



Pressing OK starts the DRC.



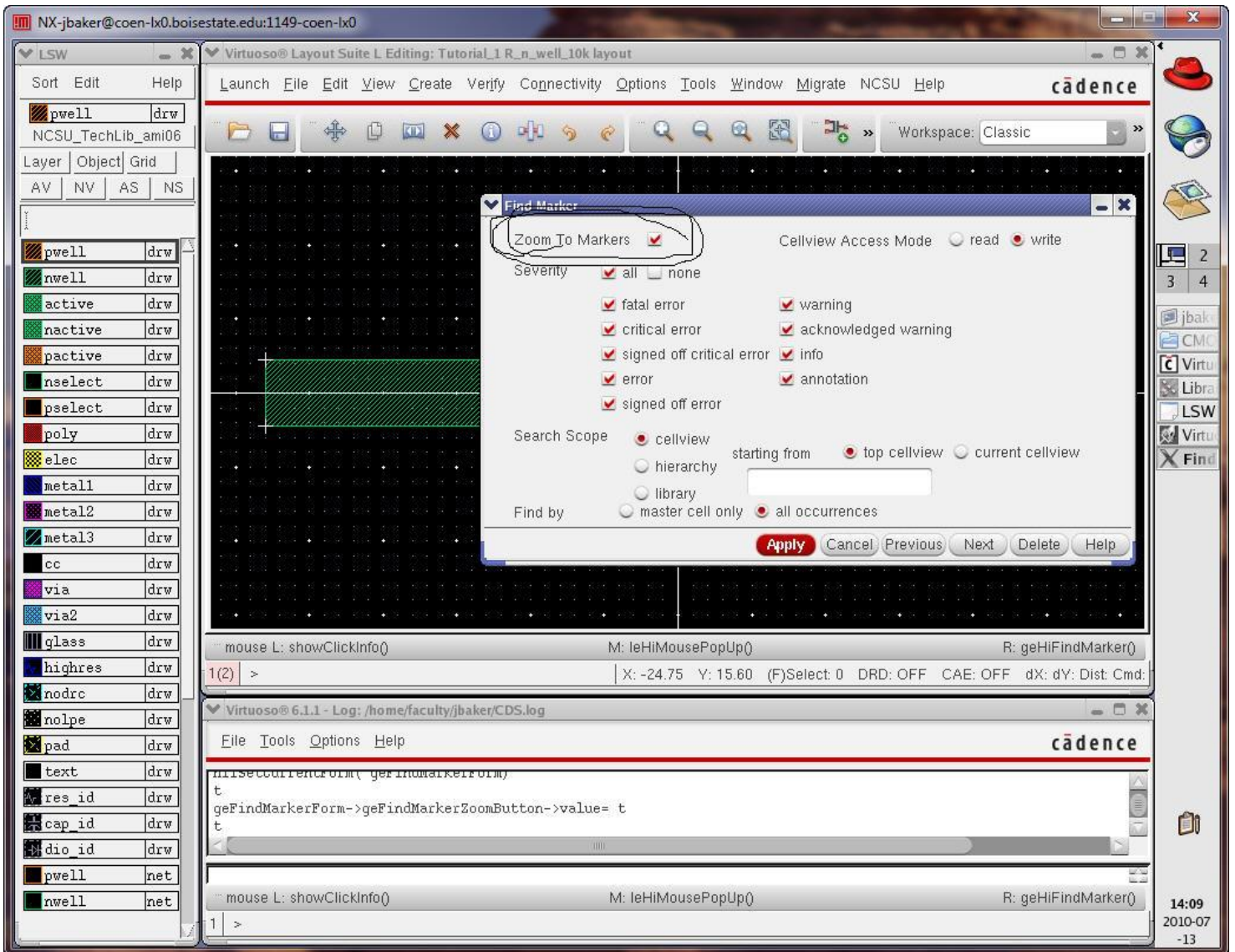
The CIW shows there are 4 errors (the edges are not on grid).

The questions are how do we find out the grid settings and how to view the markers showing the location of the errors (above seeing the markers is easy

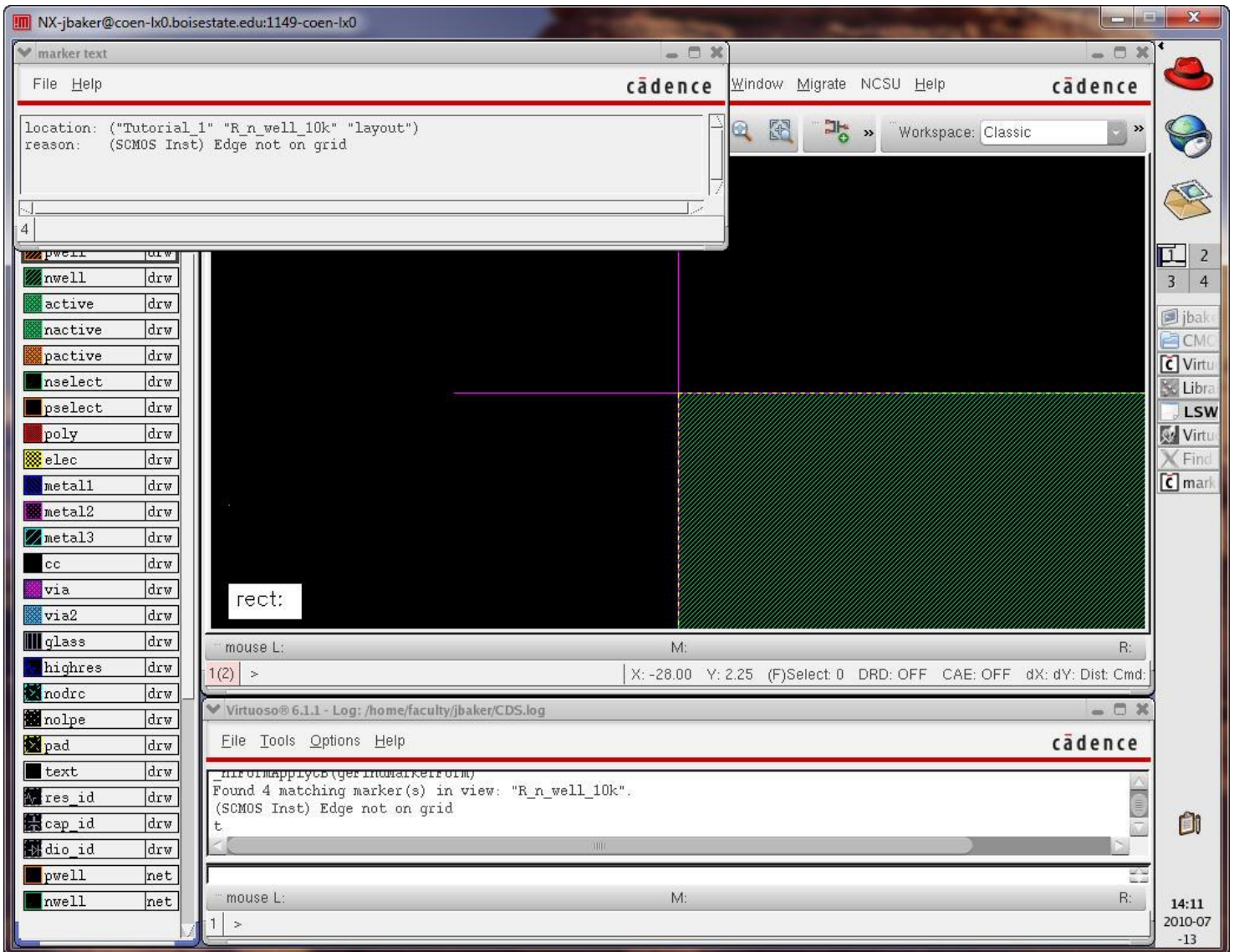
since the layout is simple....the crosses in the corners are the markers).

Using the menu Verify -> Markers -> Find Marker (notice you can delete the markers in this menu path too).

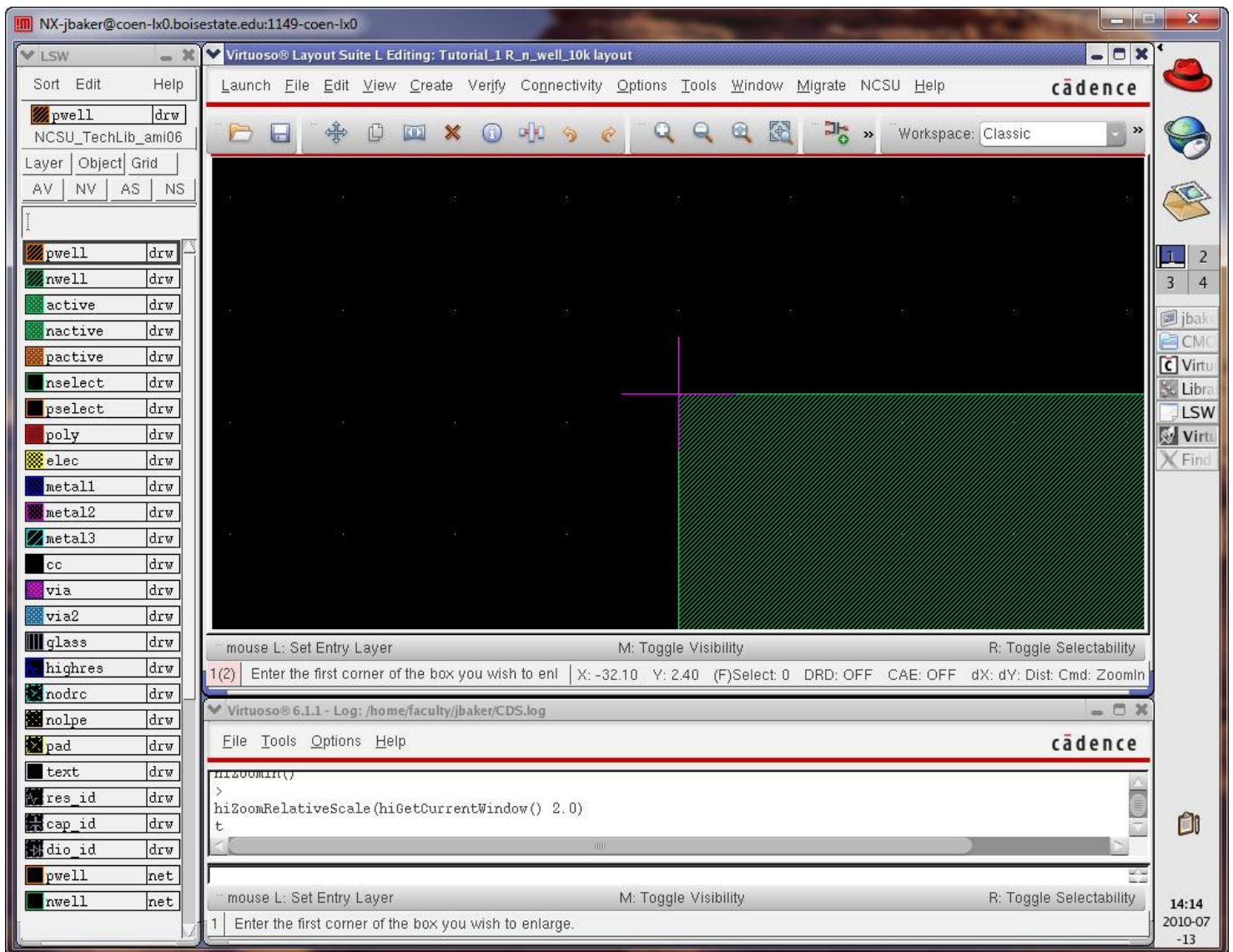
Select Zoom To Markers



Hitting Apply we get



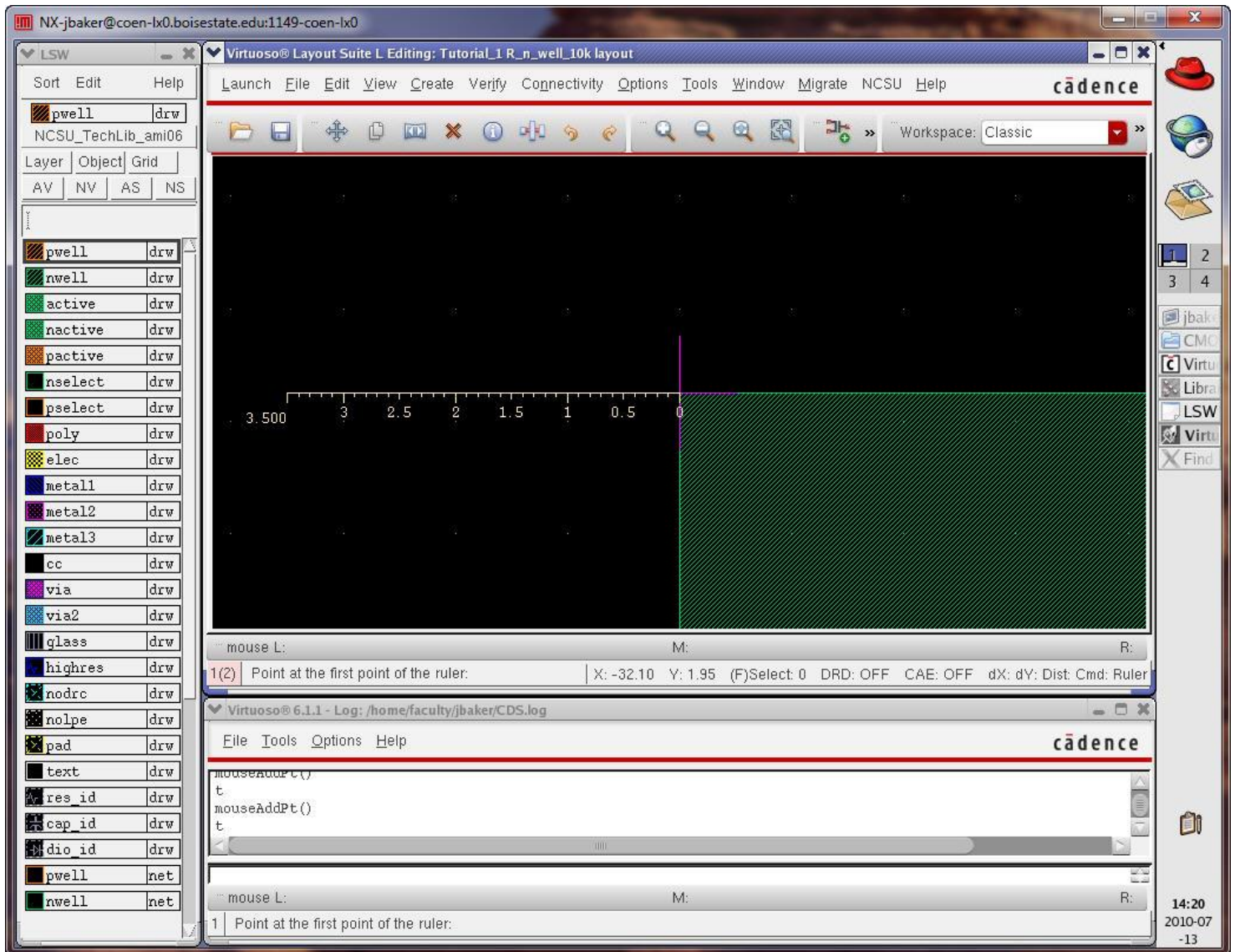
Close the marker text window, select the layout window, and zoom out a couple of times (press Z) until you can see the grid.



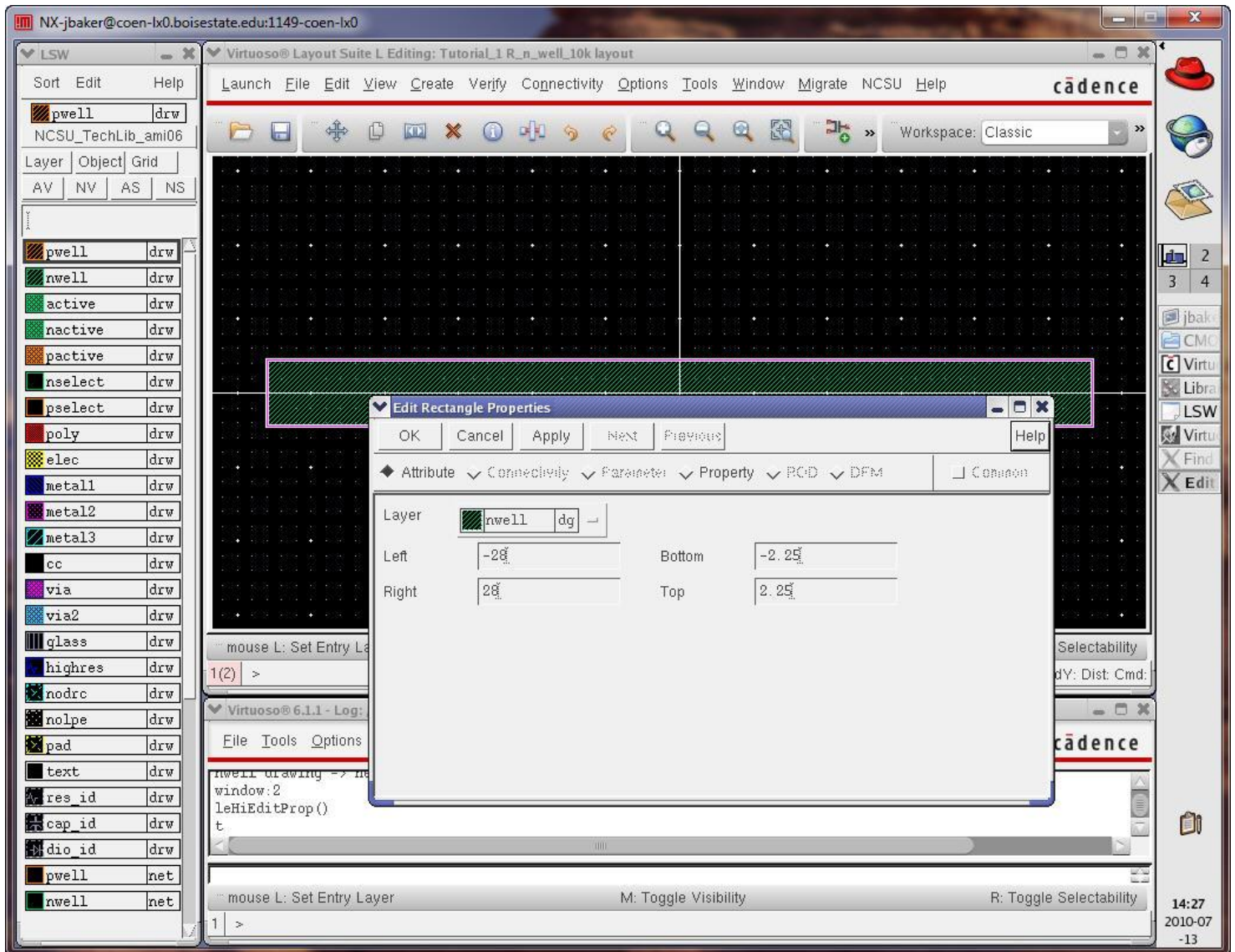
While it's a little challenging to see in the figure above the corner isn't snapped to the y-axis grid.

Use Tools -> Create Ruler (or the Bindkey k) to measure the distance between grid points (or just press e, Options -> Display to see X and Y Snap Spacing is set to 0.15 microns)





The distance between grid points is 1 micron and, as mentioned, the X and Y snapping is 0.15 microns. Clear the ruler by using the menu items Tools -> Clear All Rulers (or just press K). Press f to fit the layout in the drawing area. Use Verify -> Markers -> Delete All followed by OK to delete the markers. Press Esc a few times to ensure no commands are active. Next select the layout and press q to get the following



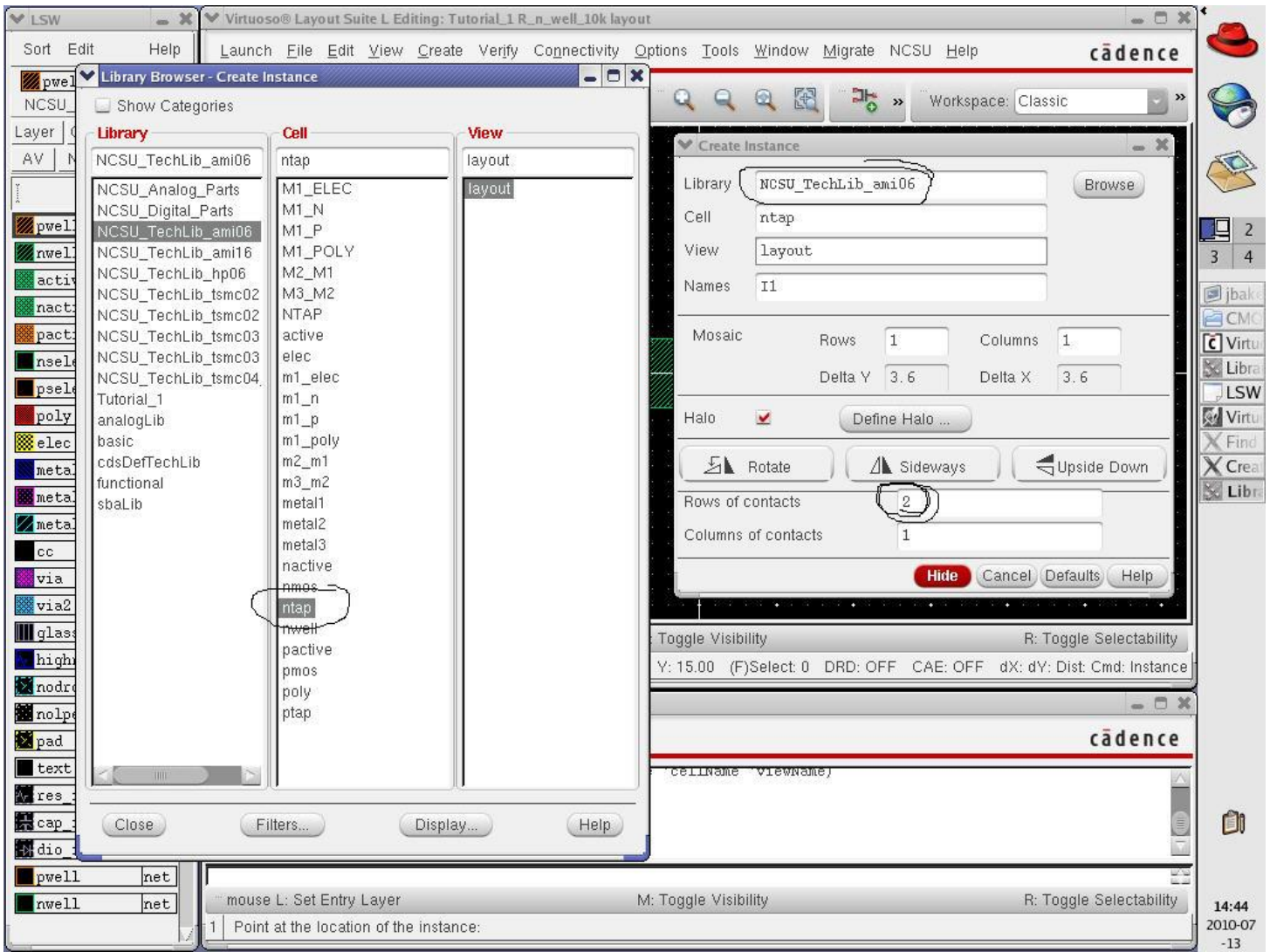
So the length of the resistor is  $56$  and  $56/.15 = 373.3333$ . To make this a whole number let's increase the length to  $56.1$  (so we enter **28.05** in the **Left/Right** above).

For the width we used  $4.5$  and  $4.5/.15 = 30$  so we are okay there.

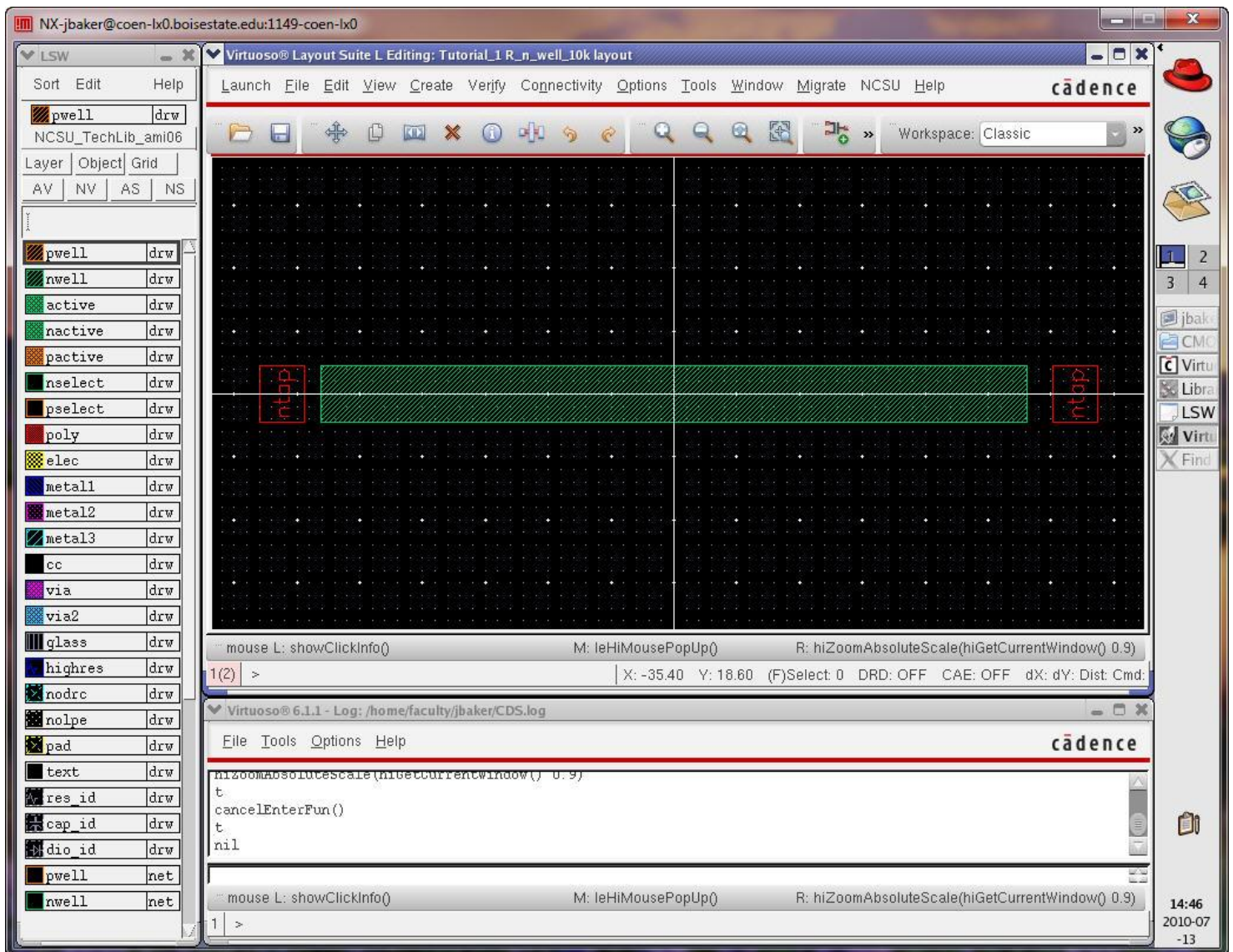
Running the DRC shows no errors are found.

Next let's add the connections to the ends of the resistors.

Press **i** and navigate/select the ntap (metal1 connection to n-well) as seen below.



Adding these connections to the ends of the n-well resistor we get the following.



You may have noticed that when placing the nodes they had an affinity to the n-well rectangle. This is called gravity, which can be useful. However, here it's not useful so let's turn it off. Go to Options -> Editor (or press **shift-E**) and deselect the "Gravity On" check box.

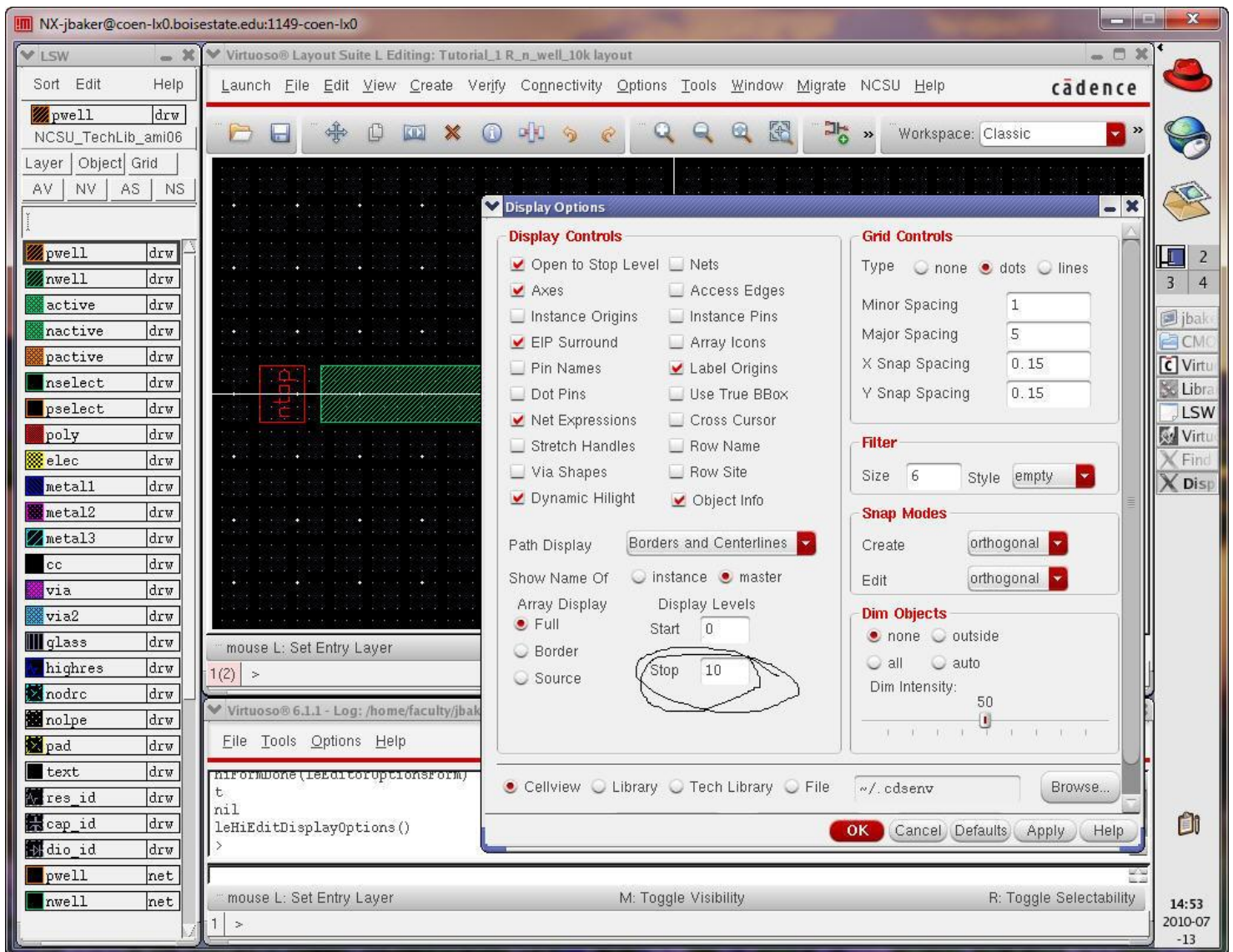
The screenshot shows the Cadence Virtuoso Layout Suite L Editing interface. The main window displays a layout editor with a dark background and a grid. A dialog box titled "Layout Editor Options" is open, showing various settings. The "Gravity Controls" section is highlighted with a red circle, and the "Gravity On" checkbox is checked. The dialog also shows "Editor Controls" (Repeat Commands, Display Reference Point, Auto Set Reference Point, Recursion Check, Maintain Connections, Abut Server, QCell Auto Abutment) and "Snapping" (Instance to Row, Snap To Track) options. The "Tap" section is also visible, with options like Wire Auto Tap, Shape Auto Tap, Tap Layer, Tap Attributes, and Select from Overlaps. The "File" field shows the path ~/./cdserv. The "OK" button is highlighted in red.

Below the dialog box, the command line shows the following command:

```
leHiEditEditorOptions()
>
leEditorOptionsForm->gravityOn->value= nil
nil
```

The bottom right corner of the window shows the date and time: 14:51 2010-07 -13.

Next notice that the ntap cells are drawn as outlines. Go to Options -> Display (or just press e) and set the depth of display to 10.

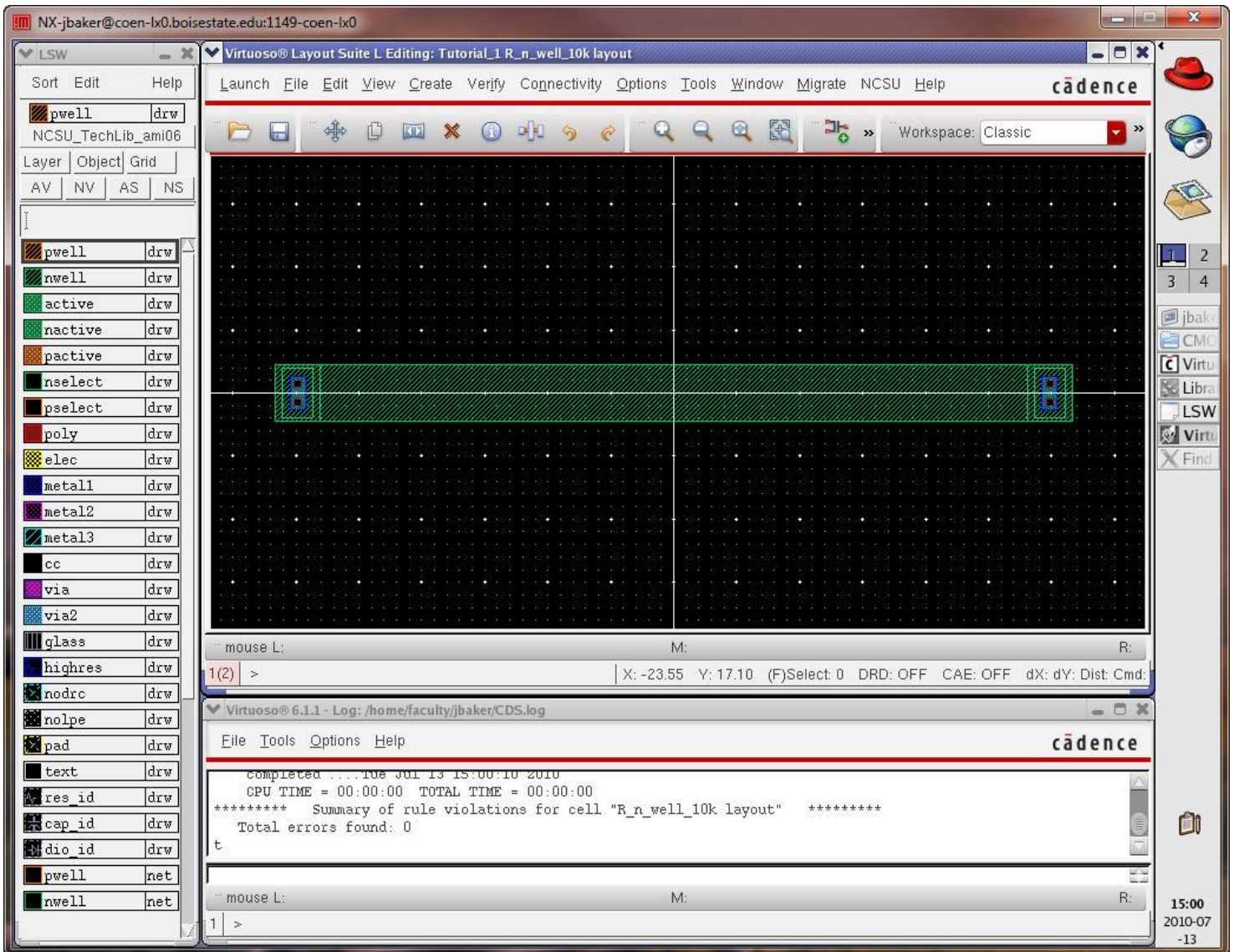


Next select the ntap cells then press m to move them (or use the menu Edit -> Move)

Line the cells up as seen below.

Pressing z then click, move the mouse, then click again to set the window (you can't click and drag to zoom in).

DRC the layout to ensure no errors.

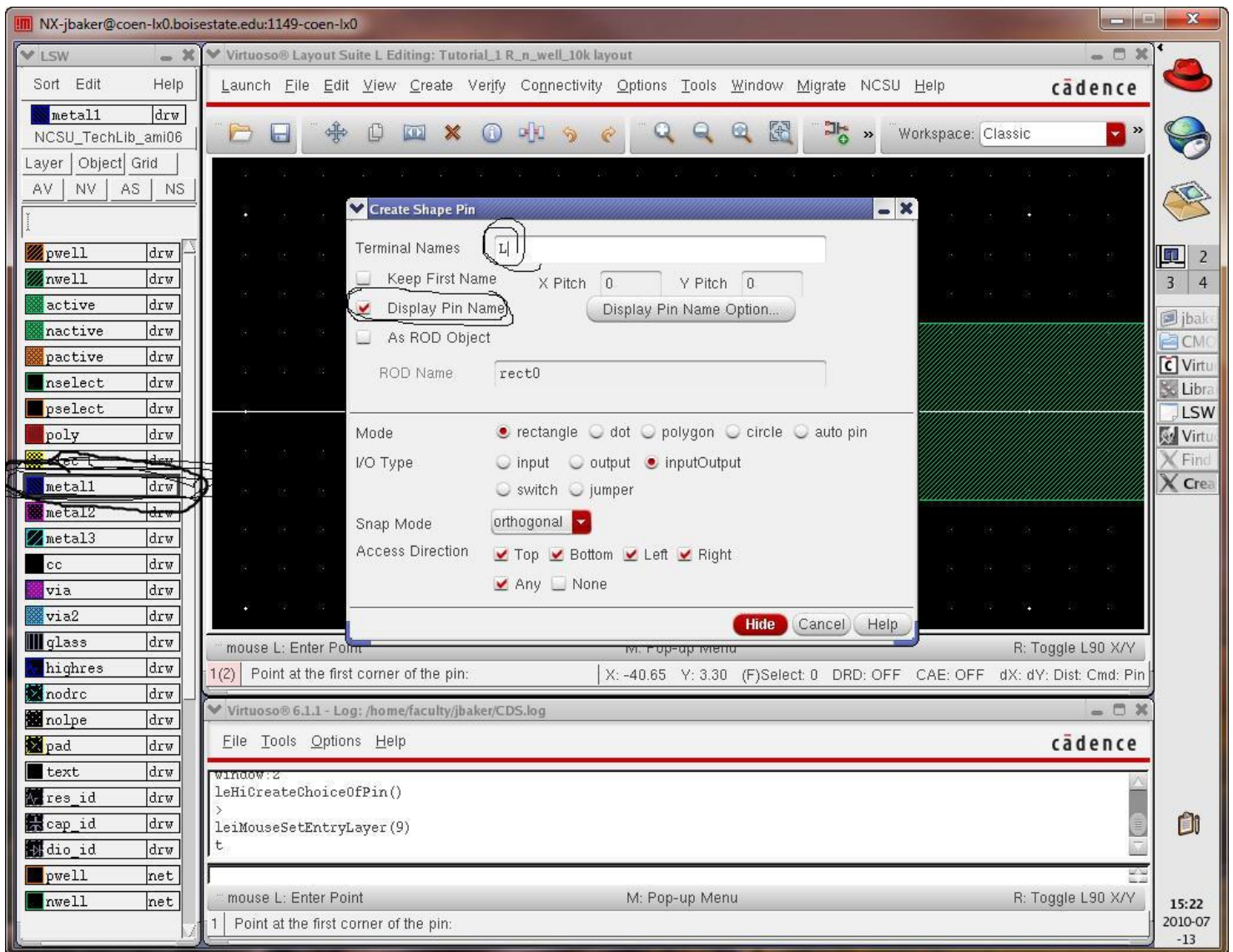


Next let's add pins to the layout.

Zoom in on the left side of the layout.

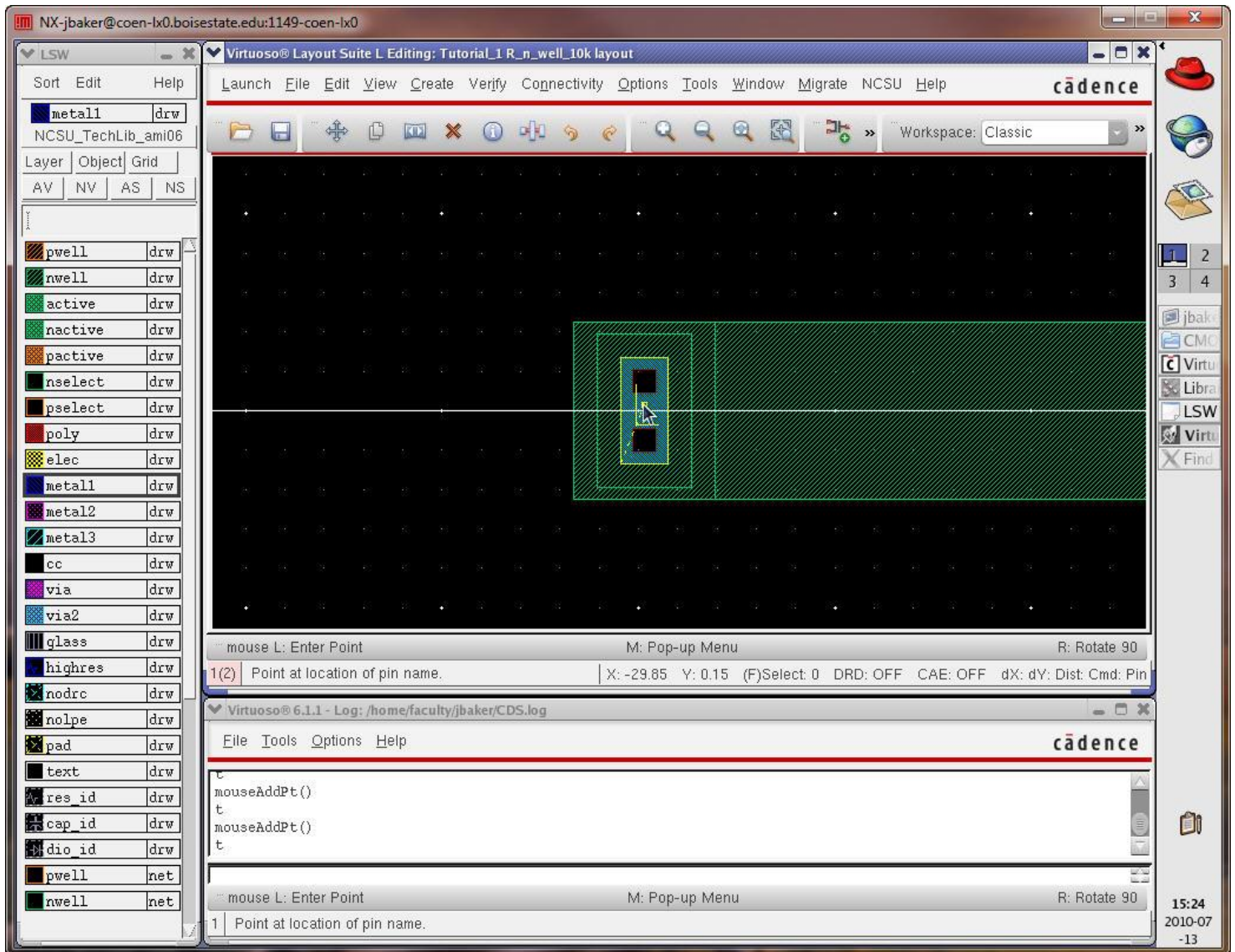
Then use the menu commands *Create -> Pin*

Select "Display Pin Names", a name of L (left), and the metal1 layer as seen below.



Select Hide and then draw a rectangle around the metal1 on the ntap placing the Pin Name on the center of the metal1 rectangle.





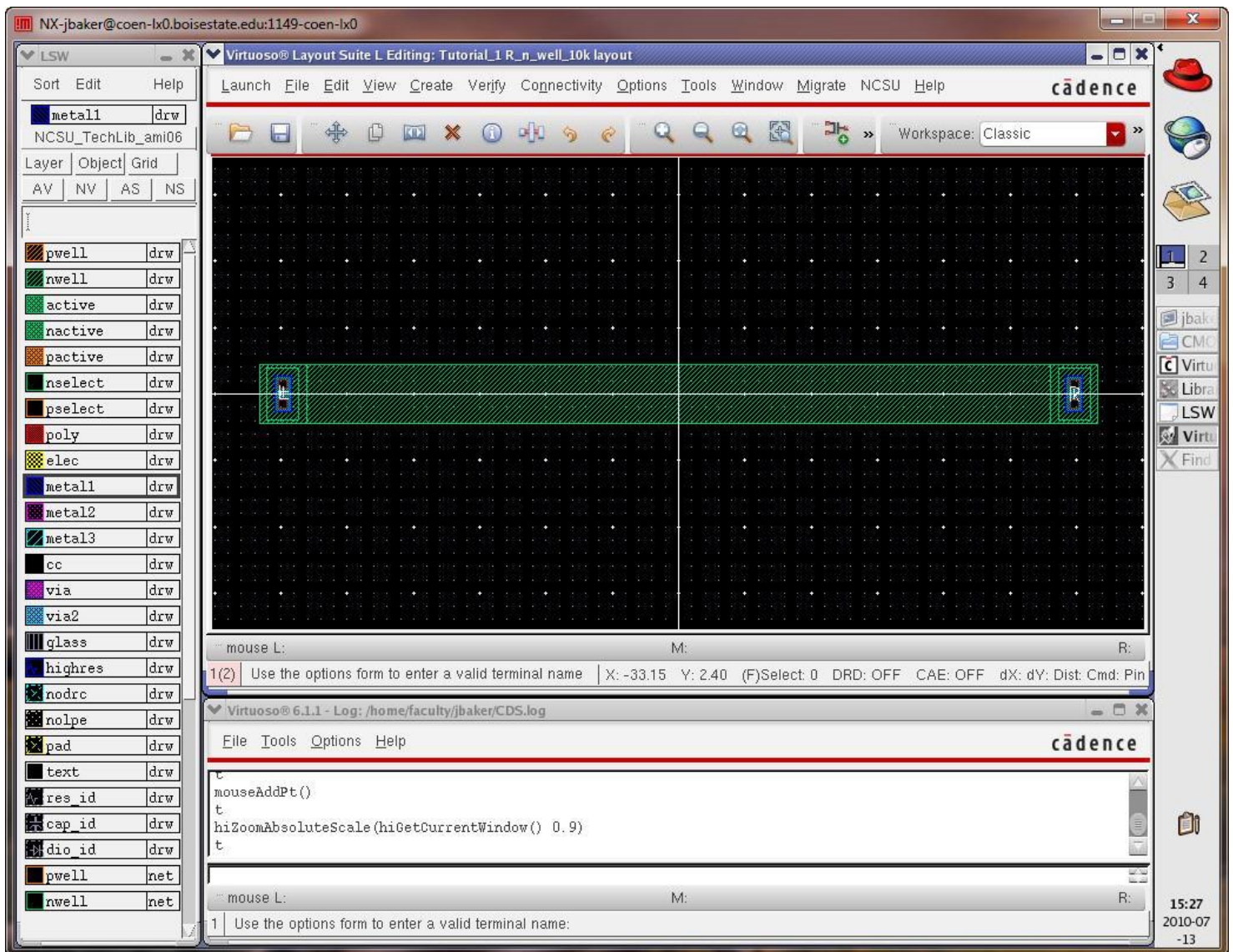
Repeat, but use a Pin Name of R (right), for the other side.

Note that to change the size of the text used for the pin's name you select the associated text and edit (q) it.

The pin's name, however, can't be changed by selecting the text which can be counterintuitive.

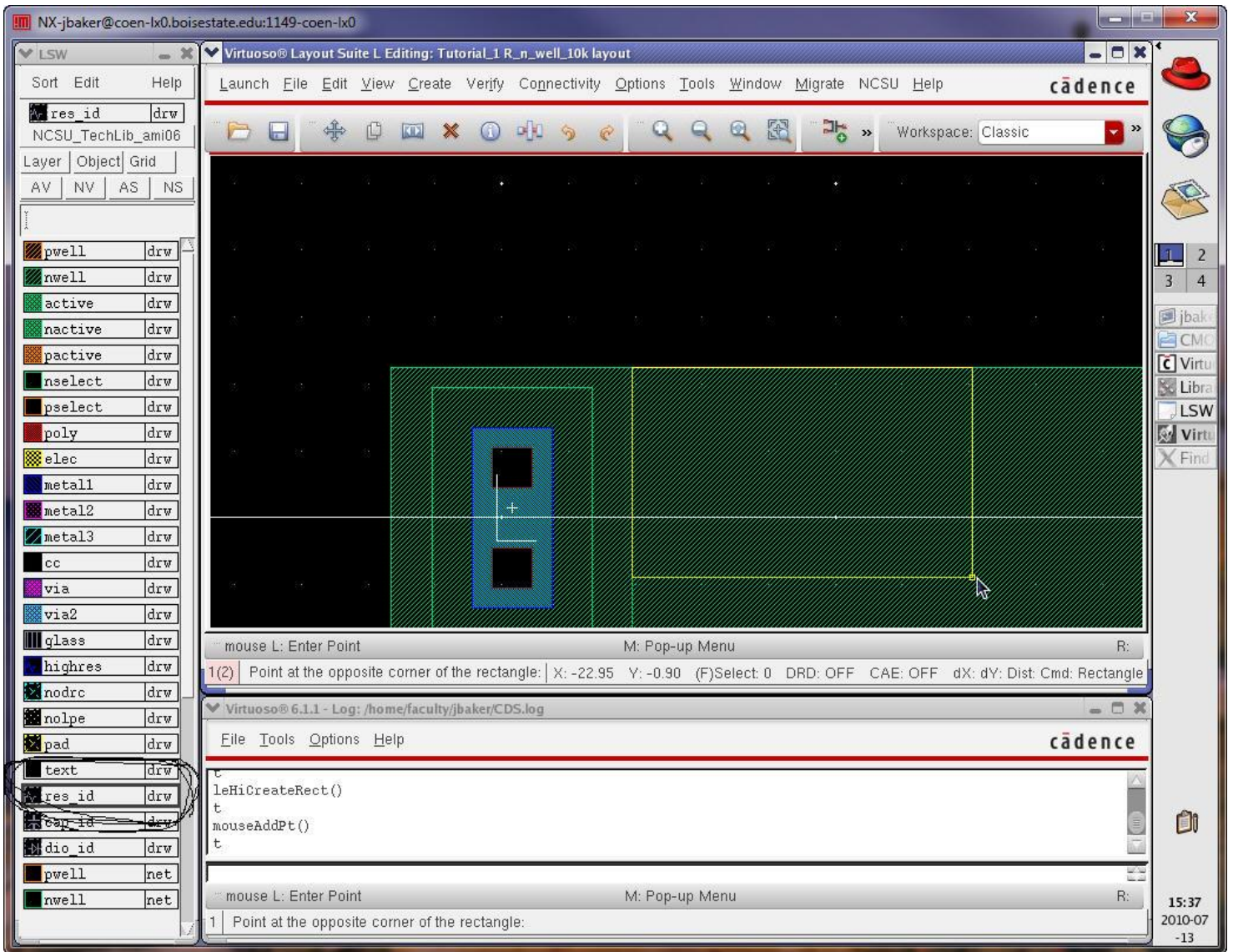
Rather, to change the pin's name, select the shape that the pin is drawn with, here a rectangle on metal1, and then q it.

\*\*\*Not understanding this can waste time as you are drawing layouts and trying to change pin names.\*\*\*

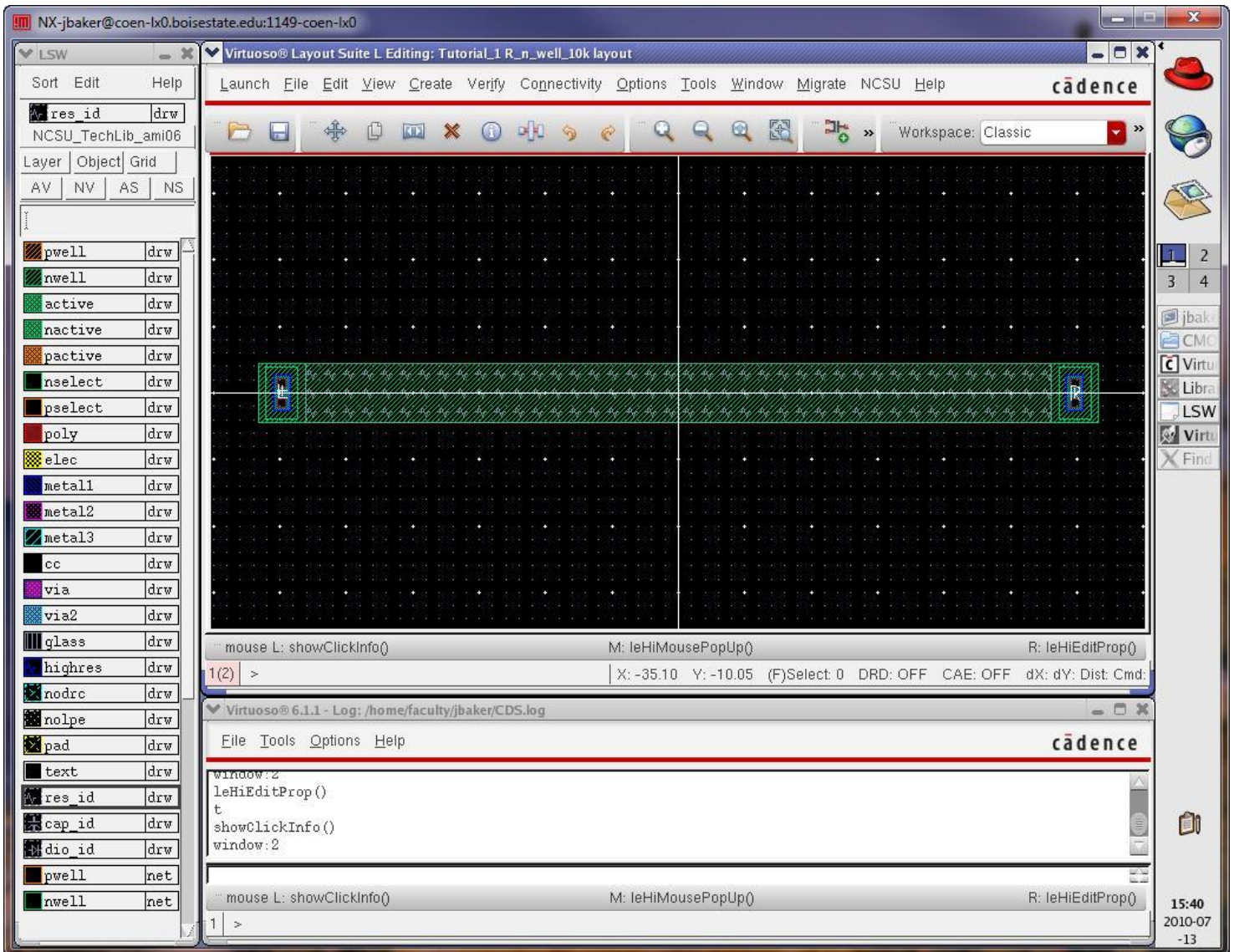


If we were to DRC this layout we would get an error "Label/Pin "R" is on a net already named "L".  
 The way the layout is now the n-well is treated as a conductor (a wire) not a resistor.  
 In this situation Cadence thinks that we are trying to label the same wire with two names.  
 To indicate that we are using the n-well as a resistor, and thus get rid of the error, we use the layer `res_id`.

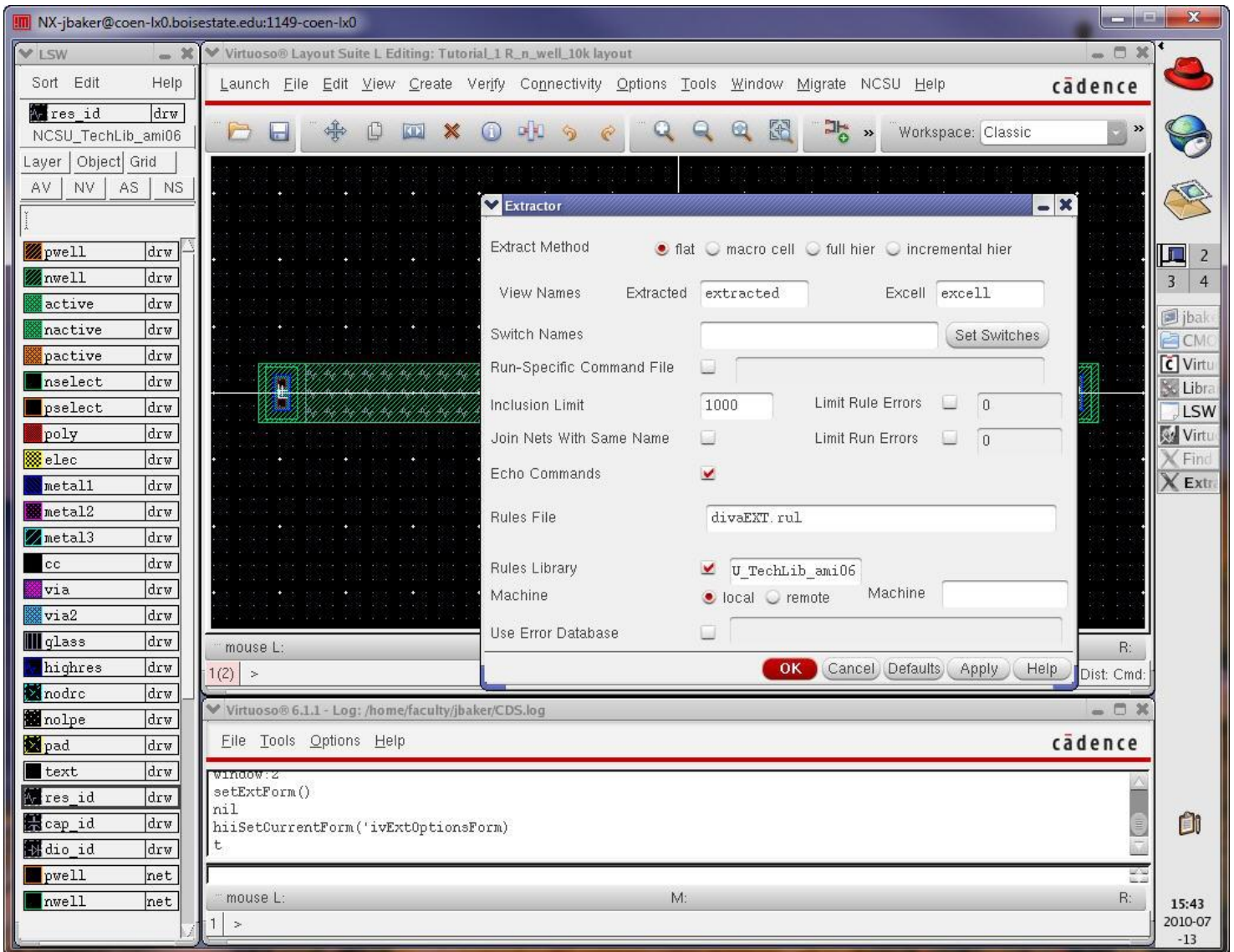
Next select the layer `res_id` (to identify resistors).  
 Select `r` to draw a rectangle.  
 Zoom in and start drawing a rectangle.



When finished the layout should look like



Next let's Extract the layout to determine the resistance's value (and to see if the setups match the 800 ohms n-well sheet resistance we got from MOSIS).  
Go to Verify -> Extract



After hitting OK the window closes and an extracted view is created in the R\_n\_well\_10k cell group.

The screenshot shows the Cadence Virtuoso Library Manager interface. The main window is titled "Library Manager: WorkArea: /home/faculty/jbaker/CMOSedu". It features three panes: "Library", "Cell", and "View".

- Library Pane:** Lists various libraries including Tutorial\_1, NCSU\_Analog\_Parts, NCSU\_Digital\_Parts, and NCSU\_TechLib\_ami06.
- Cell Pane:** Shows the selected cell "R\_n\_well\_10k" with its components: R\_div, R\_n\_well\_10k, and sim\_R\_div.
- View Pane:** Displays the "extracted" view of the resistor, with the value "10.21k" circled in red.

Below the Library Manager is a terminal window titled "Virtuoso@6.1.1 - Log: /home/faculty/jbaker/CDS.log". It shows the following log output:

```

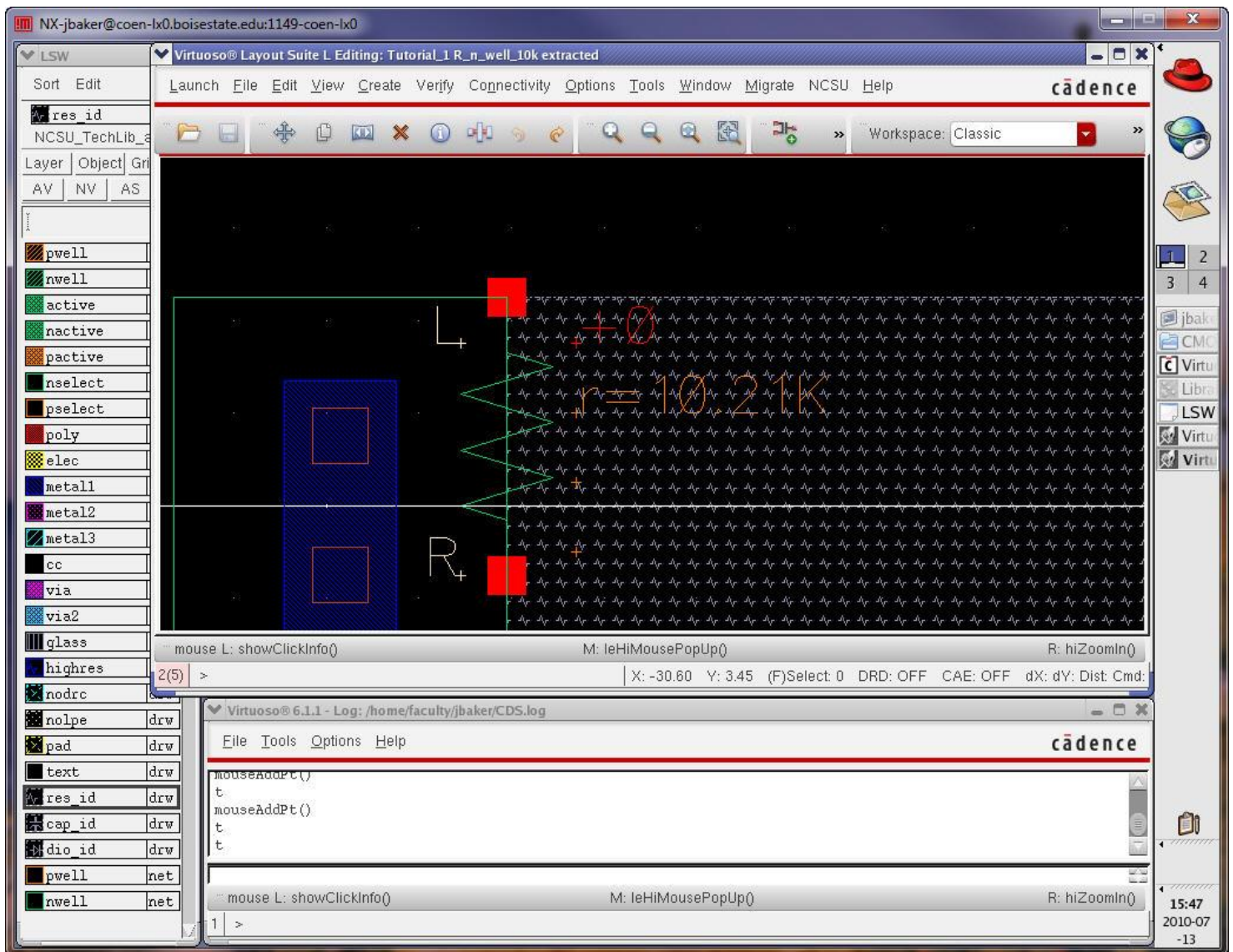
informCancel(getInfoMarkerFrom)
t
nil
(virtuoso) Syncing library list with the Library Manager.
mouse L: M: R:
1 >

```

The system clock in the bottom right corner indicates the time is 15:45 on 2010-07-13.

Open this view.

Zoom in to see the resistor's value is 10.21k.



Close the extracted view.

Save and close the layout view of the resistor.

We are now ready to draw the layout of the R\_div cell.

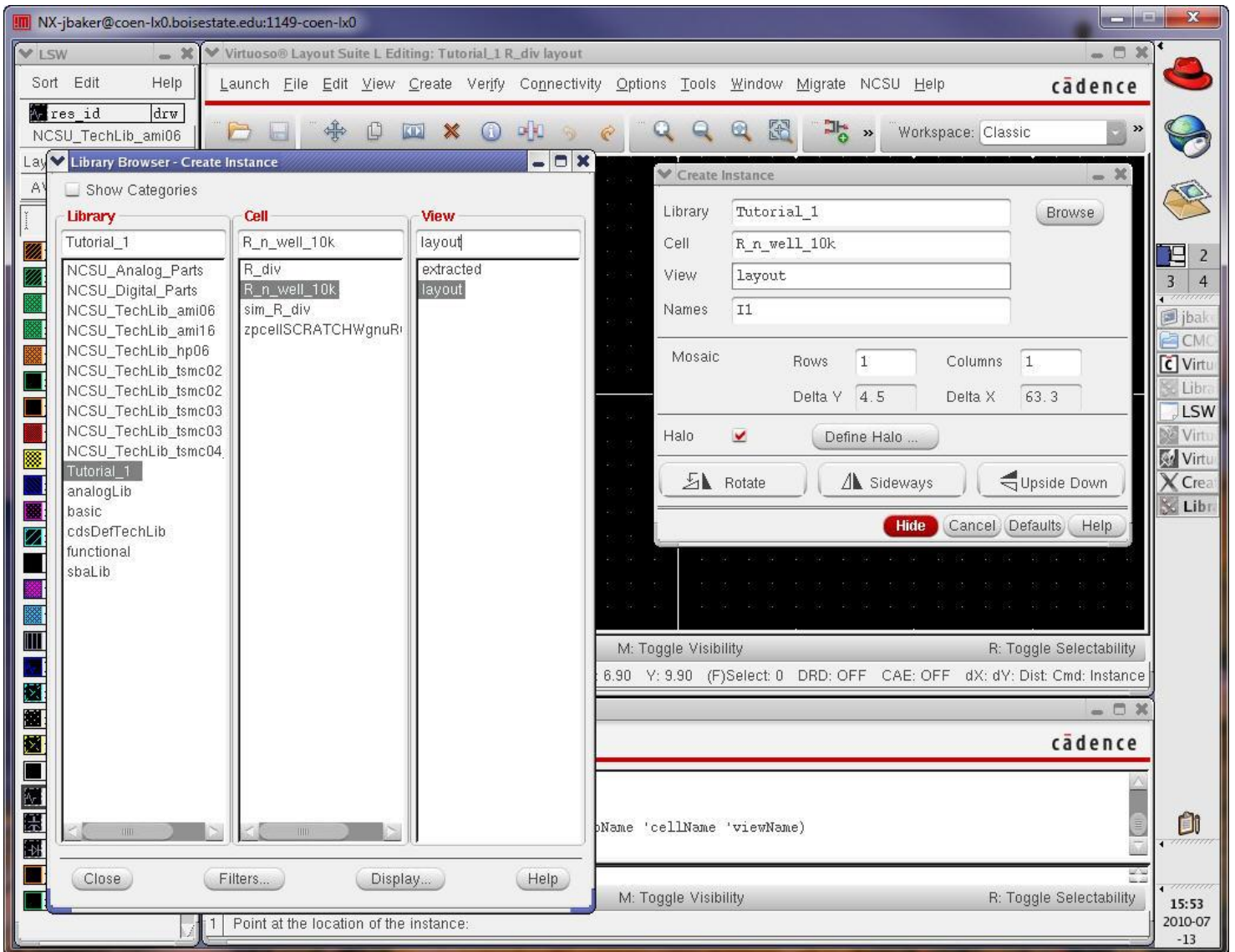
Open the schematic view of the R\_div cell (so we remember what is in it, like Pin Names).

Open the layout view for the R\_div cell (nothing is in this cell).

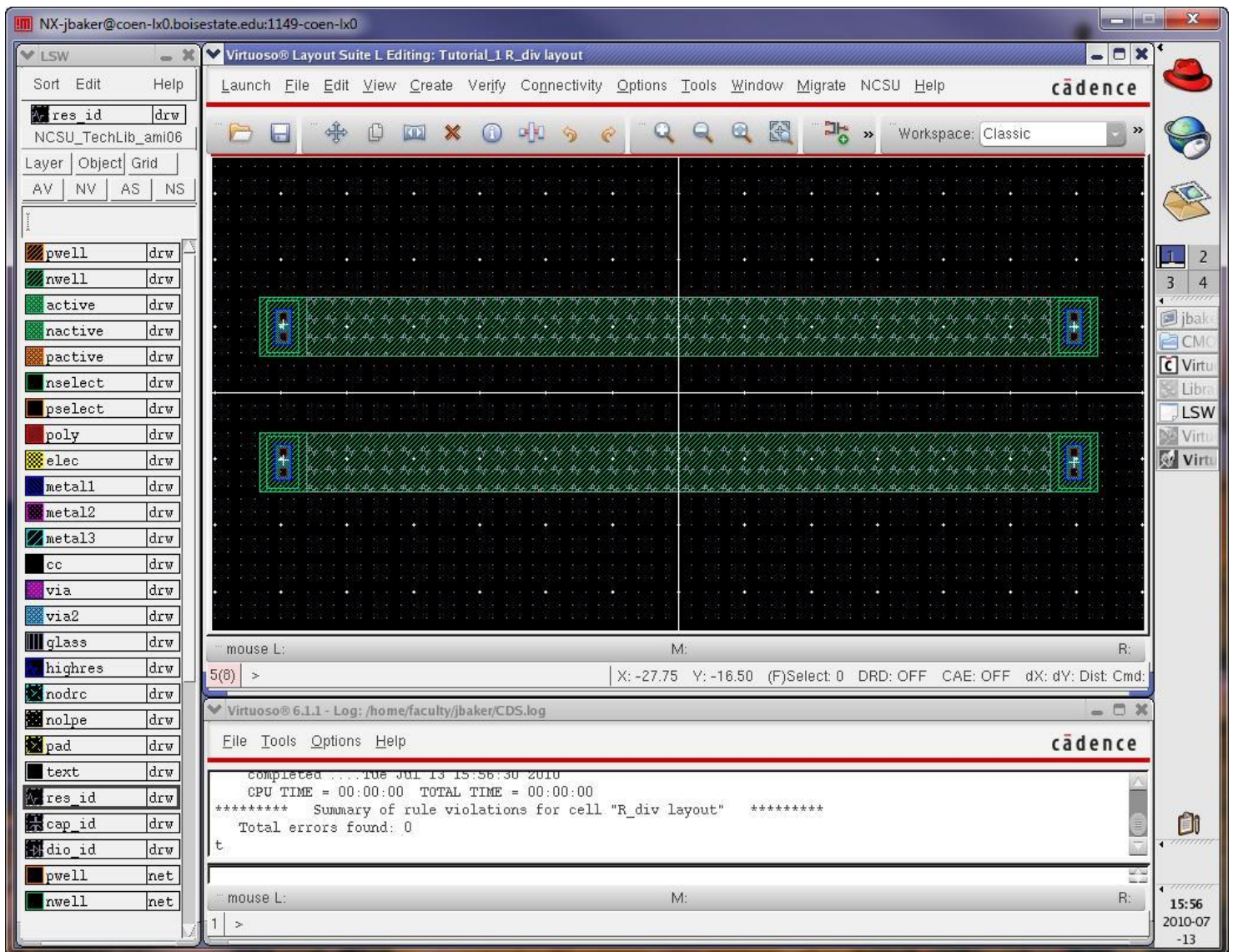
Instantiate two of the 10k n-well resistor layouts

Remember pressing Esc leaves the instantiate command.

DRC your layout to ensure the resistors are far enough apart.







Next select the metal1 layer in the LSW and add rectangles to connect the resistors together and to connect to the Pins of the resistors.

\*\*\*When drawing layouts always use the "drw" layer and not the "net" layer. Selecting, for example, the metal 1 net layer instead of metal 1 drw

layer is very easy to do and will waste your time! If a layout looks right to you, and something isn't working, check that you didn't use a net layer. \*\*\*

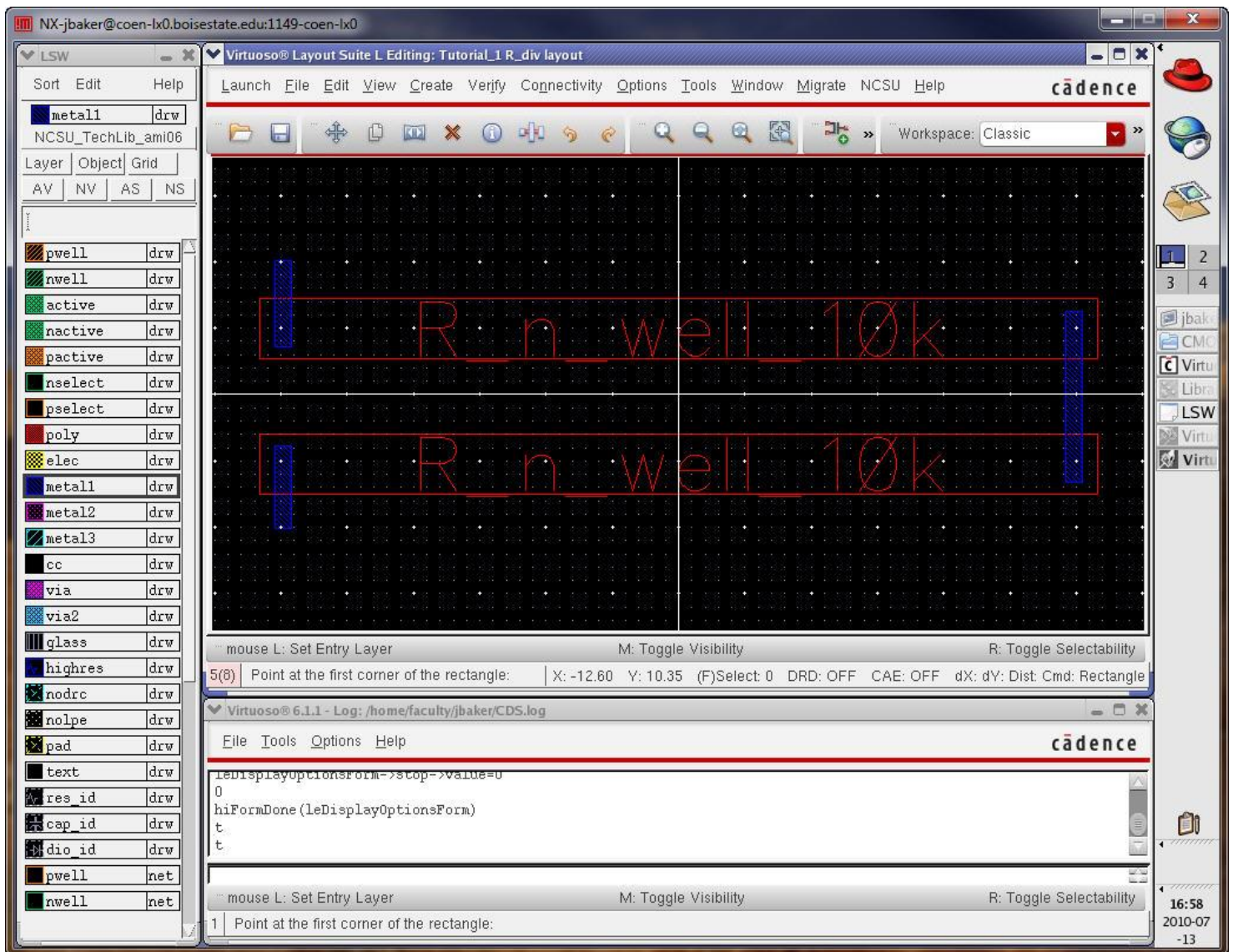
The rectangles don't have to overlap the Pins, just touch (abut) the metal1 Pins on the n-well resistors.

(I like to overlap the Pins with metal1.) One example is seen below.

DRC the design to ensure no errors.

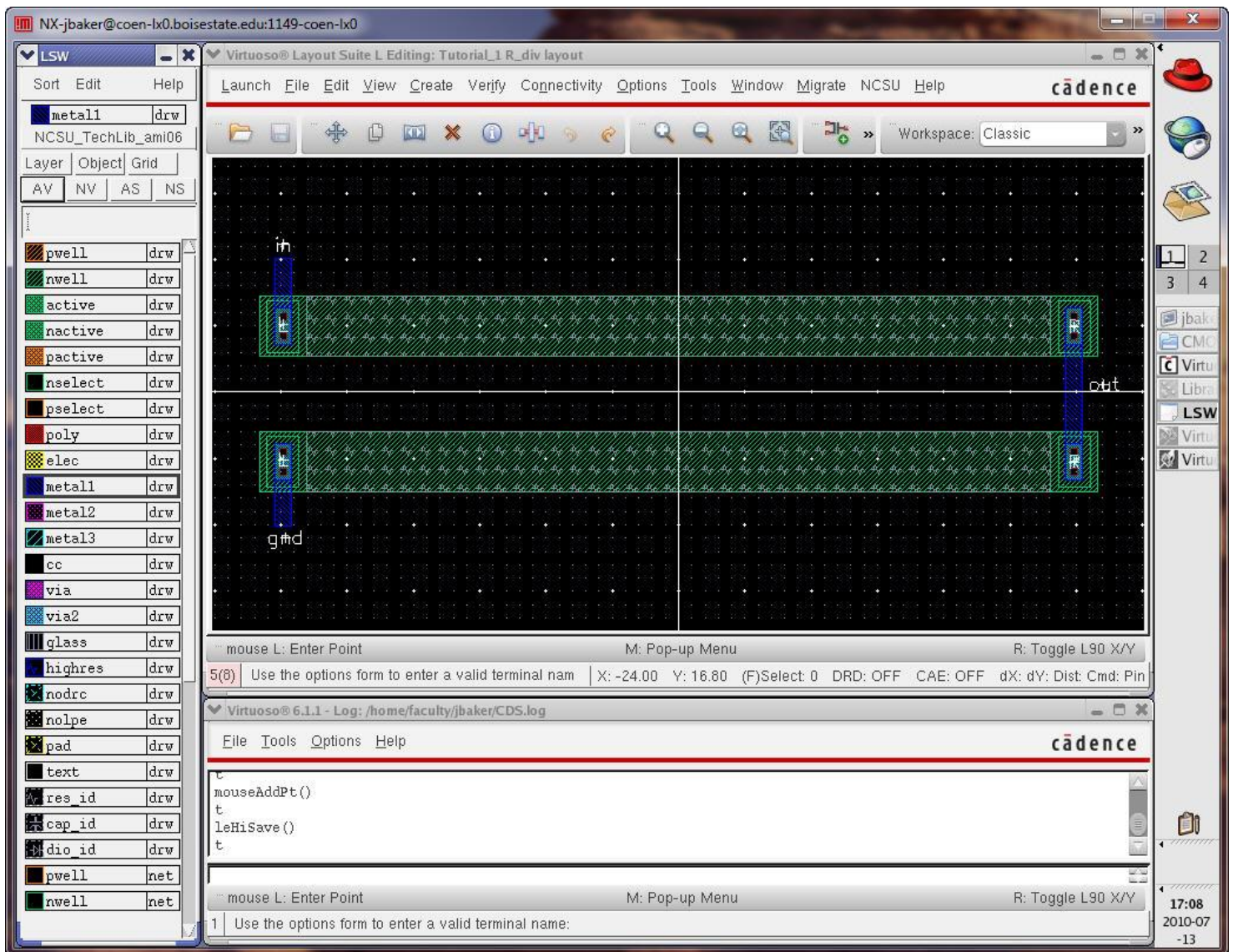
The screenshot displays the Cadence Virtuoso Layout Suite L Editing interface. The main workspace shows a PCB layout with two horizontal traces. The traces are filled with a green diagonal pattern and have blue outlines. The interface includes a menu bar (Launch, File, Edit, View, Create, Verify, Connectivity, Options, Tools, Window, Migrate, NCSU, Help), a toolbar, a layer/object grid, and a layer list on the left. The layer list shows various layers such as pwell, nwell, active, nactive, pactive, nselect, pselect, poly, elec, metall, metal2, metal3, cc, via, via2, glass, highres, nodrc, nolpe, pad, text, res\_id, cap\_id, dio\_id, pwell, and nwell. The command line at the bottom shows the command '5(8) Point at the first corner of the rectangle: X: -4.05 Y: 15.90 (F)Select: 0 DRD: OFF CAE: OFF dx: dY: Dist: Cmd: Rectangle'. A log window at the bottom shows a summary of rule violations for cell 'R\_div layout' with 0 errors found.

Pressing **e** and set the Stop Depth to zero results in showing the outlines of the cells.



Press **e** again and set the Stop Depth back to 10. Also ensure, when the Display Options Window is open, that Pin Names is still set to display.

Next add Pins on the metal1 layer named in, out, and gnd. Set the rectangle size of the Pin to the same size as the metal1 seen above.



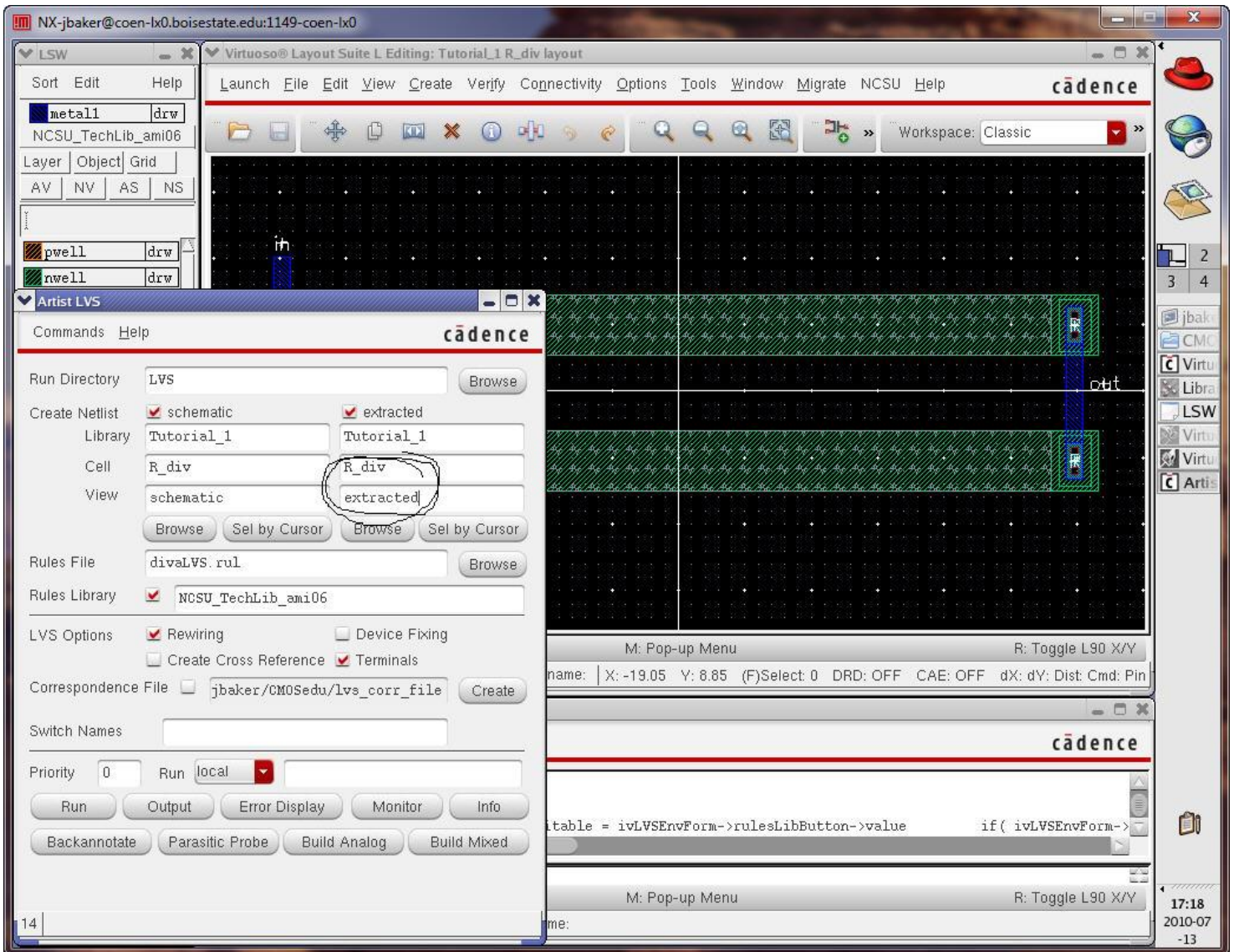
DRC your layout.

Extract your layout (we need to do this before we do a layout versus schematic, LVS)

Finally, select Verify -> LVS and set the extracted view's cell name (here R\_div) and that its view is extracted as seen below.

Hit Run and OK to "Save Cellviews" (if asked)

When the LVS is done it will, hopefully, tell you it has succeeded (hit OK).



While the LVS succeeded to run this does not tell us if the layout and schematic match!  
After pressing Output above we get the following.

The screenshot shows the Cadence LVS tool interface. The main window displays the output of the LVS command, which includes a netlist summary and terminal correspondence points. The output is as follows:

```

@(#)SCDS: LVS.exe version 6.1.1 10/23/2007 02:12 (cds126047) $
Command line: /usr/local/Cadence/IC610/tools.lnx86/dfII/bin/32bit/LVS.exe -dir /home/faculty/jbaker/CMOSedu/LVS -l -s -t /ho
Like matching is enabled.
Net swapping is enabled.
Using terminal names as correspondence points.
Compiling Diva LVS rules...

Net-list summary for /home/faculty/jbaker/CMOSedu/LVS/layout/netlist
count
  3      nets
  3      terminals
  2      res

Net-list summary for /home/faculty/jbaker/CMOSedu/LVS/schematic/netlist
count
  3      nets
  3      terminals
  2      res

Terminal correspondence points
N2      N3      in
N1      N2      out

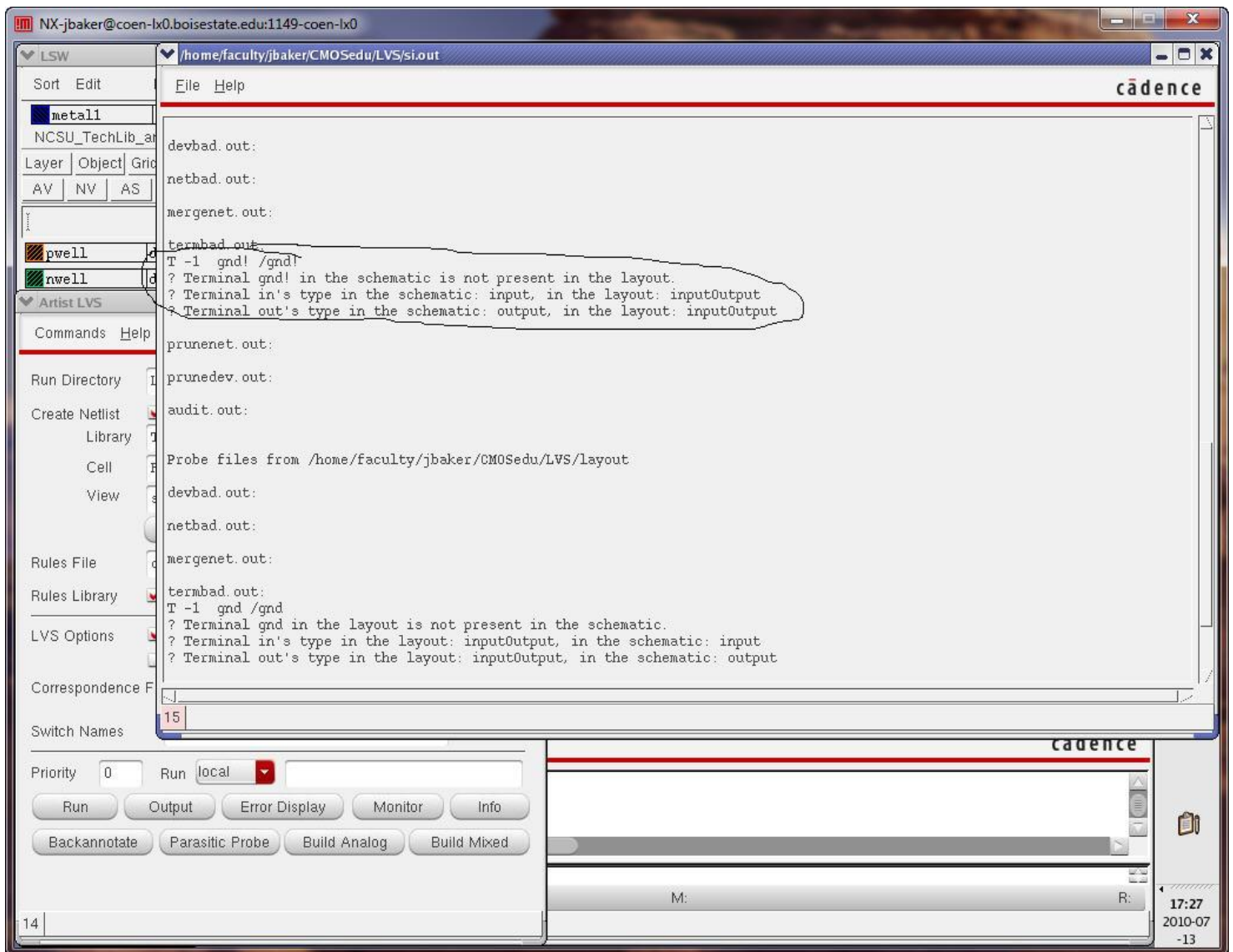
Devices in the netlist but not in the rules:
res
Devices in the rules but not in the netlist:
cap nfet pfet rmos pmos rmos4 pmos4

The net-lists failed to match.

                                layout schematic
  
```

The interface also shows a control panel at the bottom with buttons for Run, Output, Error Display, Monitor, Info, Backannotate, Parasitic Probe, Build Analog, and Build Mixed. The status bar at the bottom right indicates the time 17:22 on 2010-07-13.

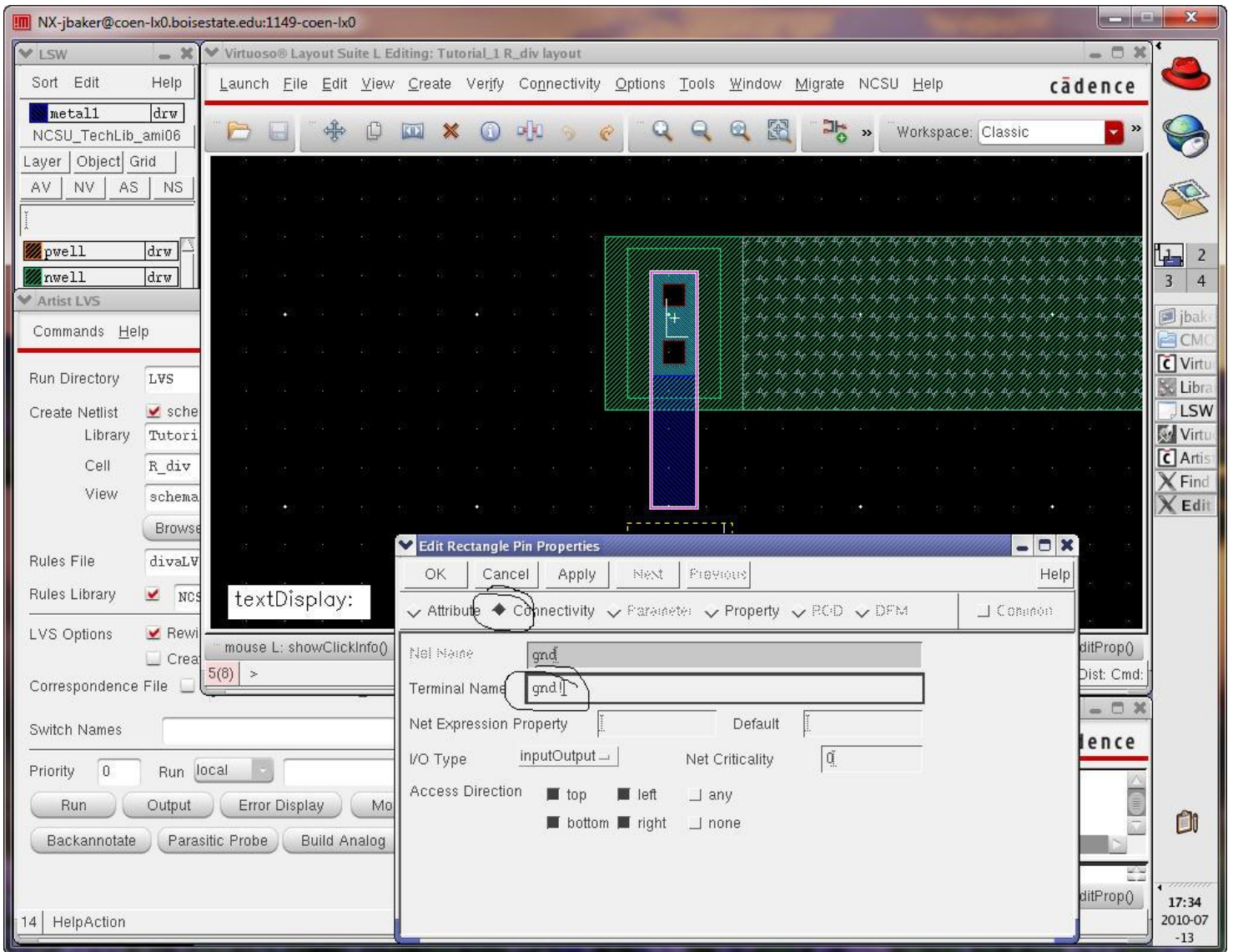
Scrolling down in the si.out file shows the problems.



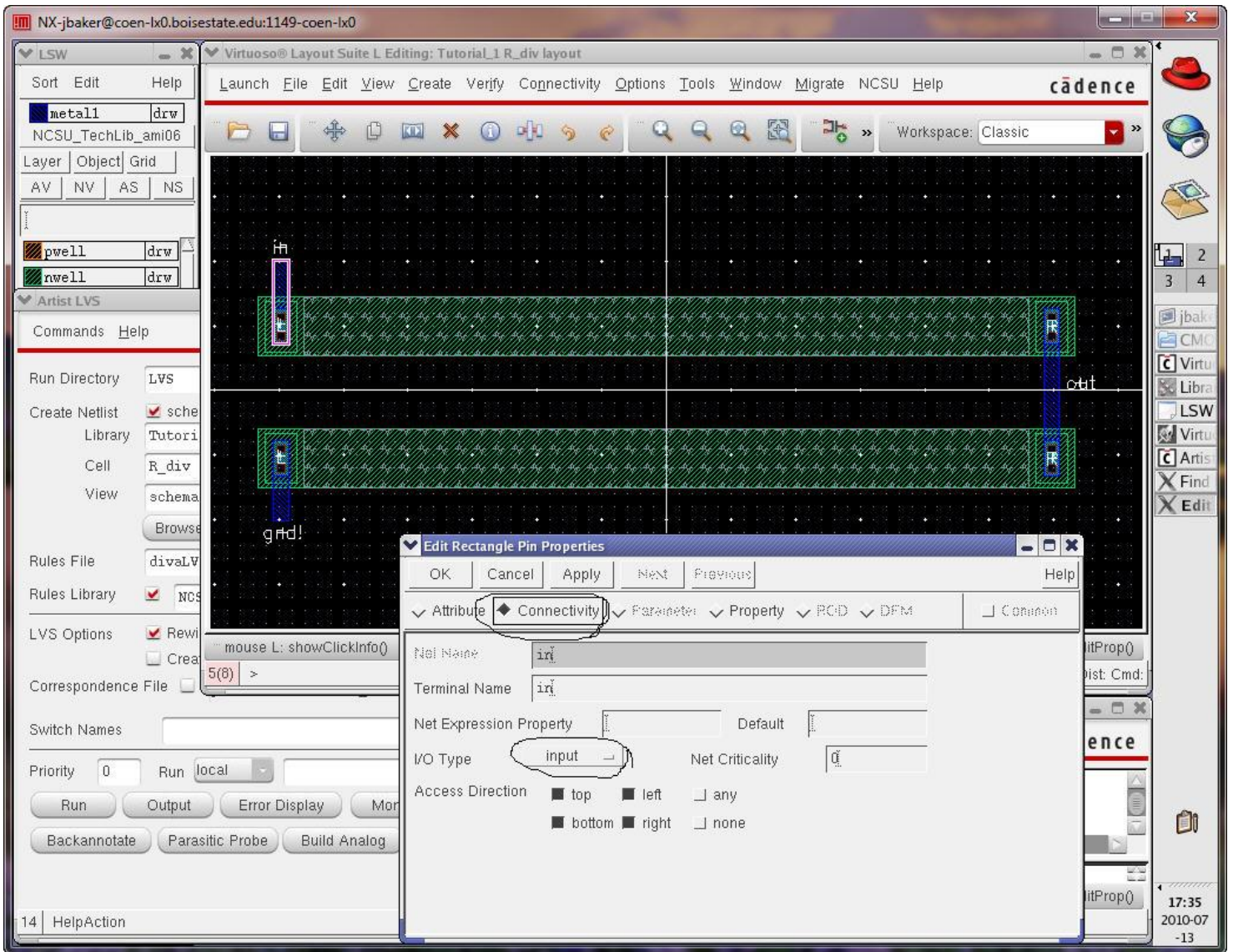
We should have labeled gnd in the layout gnd! (the exclamation point indicates a global value), and our in pin should have characteristics of input (not input/output)

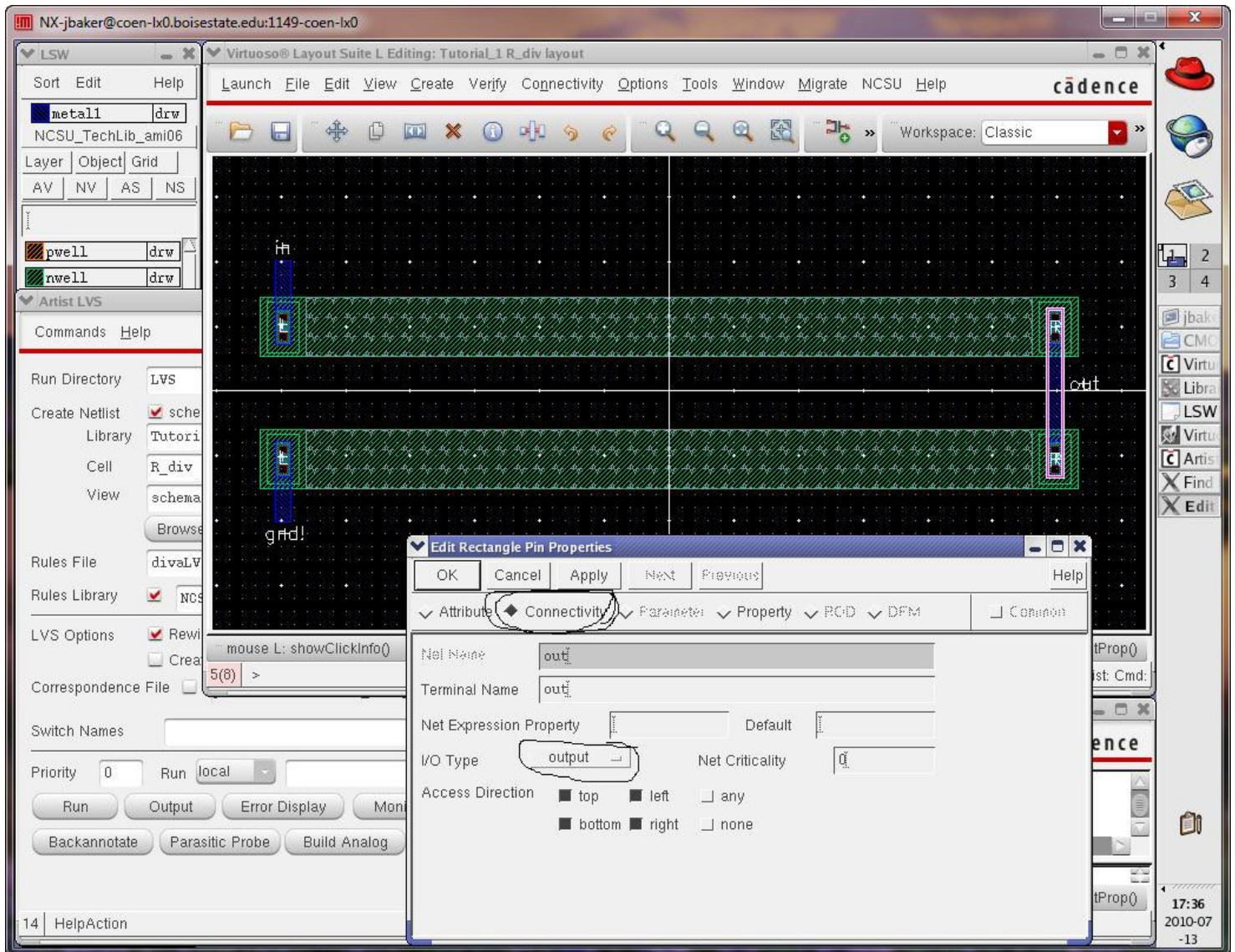
and the out pin should be an output (not input/output).

Note that we could also select the Error Display button above and view the errors in the extracted view (often much easier than viewing text).









After saving the layout, extracting the layout again, and then running the LVS again we get the following.

The screenshot shows the Cadence LVS tool interface. The main window displays the following text:

```

Devices in the netlist but not in the rules:
res
Devices in the rules but not in the netlist:
cap nfet pfet nmos pmos pmos4 pmos4
The net-lists match.

```

Below this, there are three tables comparing layout and schematic instances, nets, and terminals:

	layout	schematic
instances		
un-matched	0	0
rewired	0	0
size errors	0	0
pruned	0	0
active	2	2
total	2	2

	layout	schematic
nets		
un-matched	0	0
merged	0	0
pruned	0	0
active	3	3
total	3	3

	layout	schematic
terminals		
un-matched	0	0
matched but different type	0	0
total	3	3

Probe files from /home/faculty/jbaker/CMOSedu/LVS/schematic  
devbad.out:

17

Priority 0 Run local

Run Output Error Display Monitor Info

Backannotate Parasitic Probe Build Analog Build Mixed

14 HelpAction

17:37 2010-07 -13

This ends our first tutorial.

In this tutorial we've covered the fundamental operation of Cadence.

Mastering the topics in this tutorial is important before moving on to the other Tutorials.

For your reference the Tutorial\_1 directory is available in [Tutorial\\_1.zip](#).

Again, to use this design directory put the Tutorial\_1 folder (unzipped from above) in the CMOSedu directory and then add, to the cds.lib

```
DEFINE Tutorial_1 $HOME/CMOSedu/Tutorial_1
```

[Return](#)

