

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

Tutorial 6 - Placing circuit layouts in a padframe for fabrication

In this [tutorial](#) we'll place the R_div, NMOS_IV, PMOS_IV, inverter, nand2, and ring oscillator that we laid out in the earlier tutorials in a padframe for fabrication through [MOSIS](#).

Note that we won't include a buffer to drive the large, around 30 pF, off-chip load capacitance from the scope probes, packaging, and test board needed to test the chips.

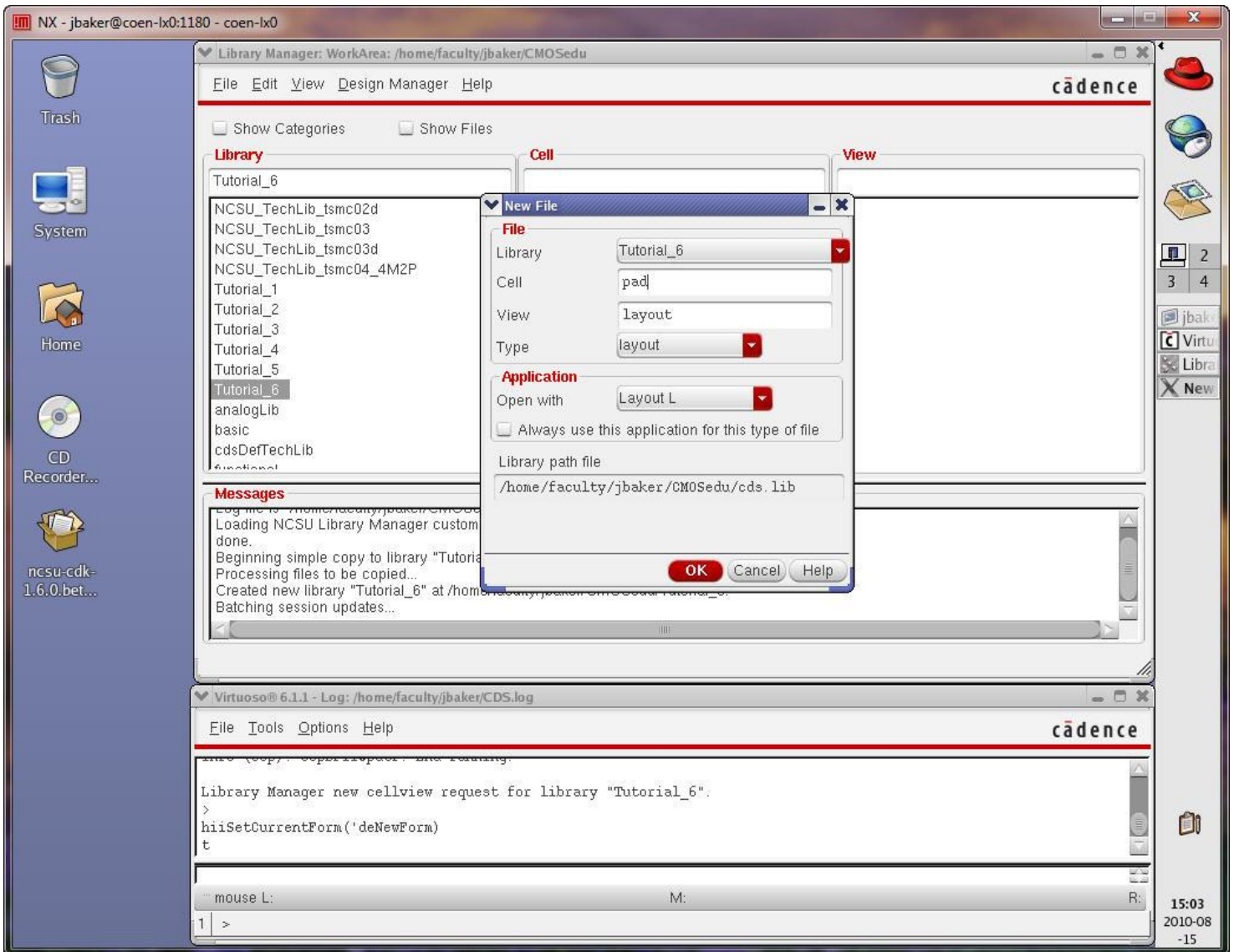
What this means is that if the chip we lay out here is fabricated it's likely that attempting to measure the oscillation frequency, or the gate delay, with a scope probe will result in drastically different results than the simulations. It's likely that measuring the ring oscillator frequency will actually result in killing the oscillations (see the [CMOS](#) book for details on the use of output buffers to drive off-chip loads).

Okay, let's begin by copying the library, Tutorial_5, into a new library called Tutorial_6. Ensure, when you copy, that "update instances" is selected so that the new library doesn't reference cells in the other libraries. As always, put the new library in \$HOME/CMOSedu

We are using On Semiconductor's [C5 process](#) for fabrication through [MOSIS](#). Further, this process uses the MOSIS scalable CMOS (SCMOS) [submicron design rules](#) with lambda of 300 nm (technology code SCN3ME_SUBM at 0.3).

A "tiny chip" fabricated in this process via the MOSIS Educational Program (MEP) measures, at most, 1.5 mm by 1.5 mm. Information about the MOSIS Educational Program (MEP) is found [here](#).

Create a new layout cellview called pad.



If we have 12 pad cells on a side (10 plus two corners) then each cell must be $1.5\text{mm}/12$ or 125 μm .

In order to ensure this pad falls on the 0.15 μm grid let's reduce the pad cell size to 120 μm .

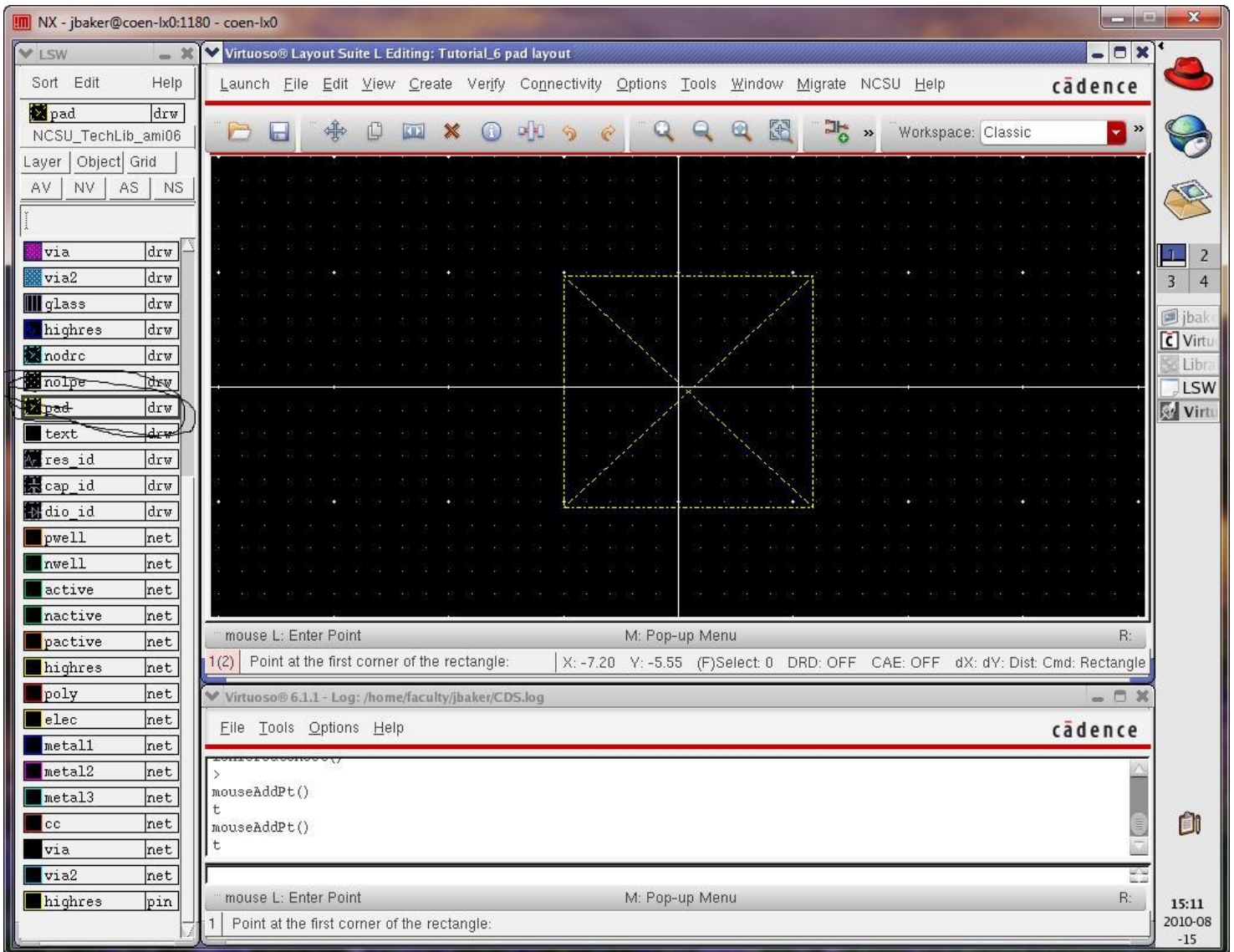
This also ensures that the final chip size is smaller than the 1.5 mm square size we are allowed.

Select the pad layer and add an arbitrary size rectangle as seen below.

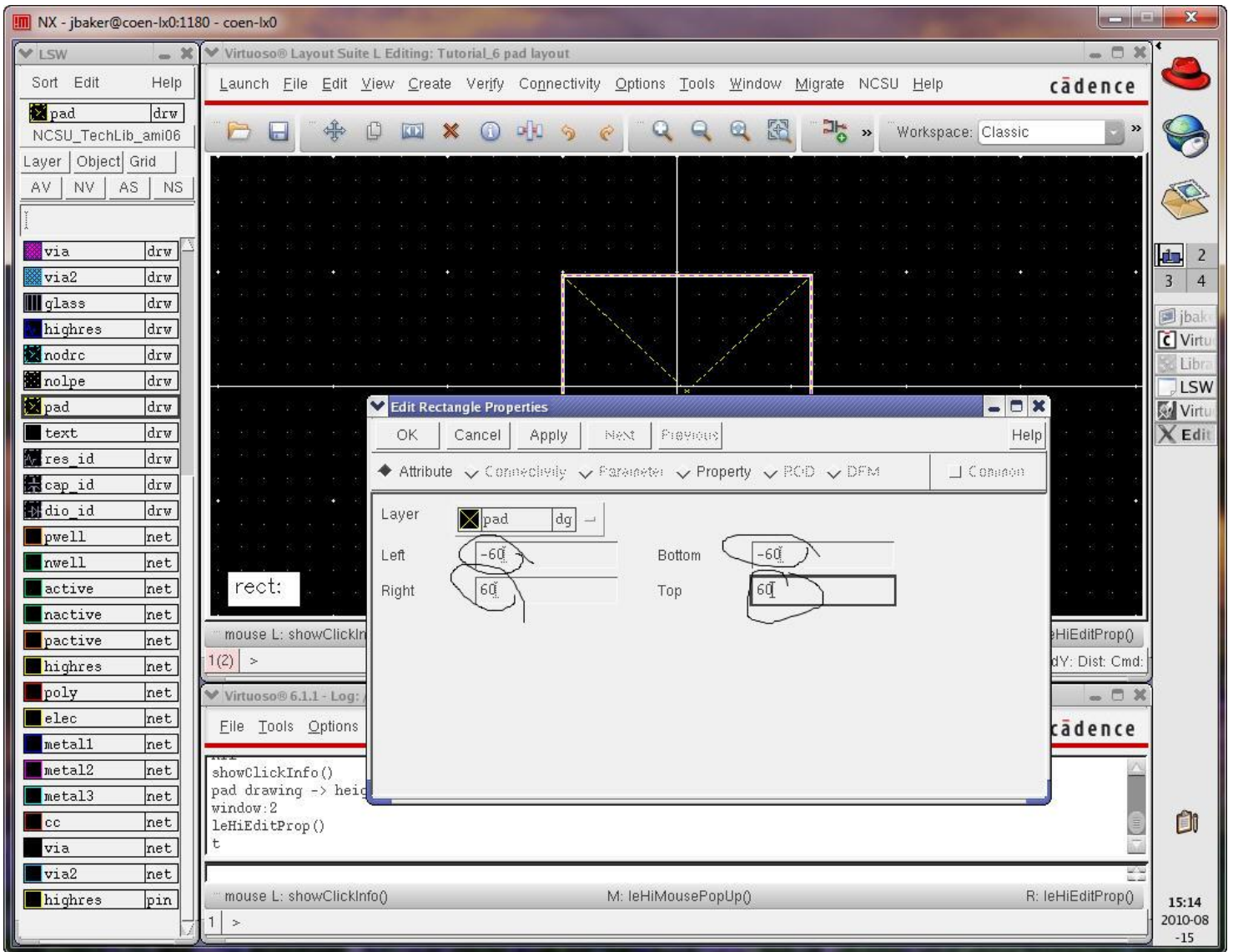
The pad layer has no fabrication significance (so you really don't need it).

The pad layer was used by MOSIS to indicate the location of the pads (now the location of the glass layer, that is cuts in the passivation, is used).

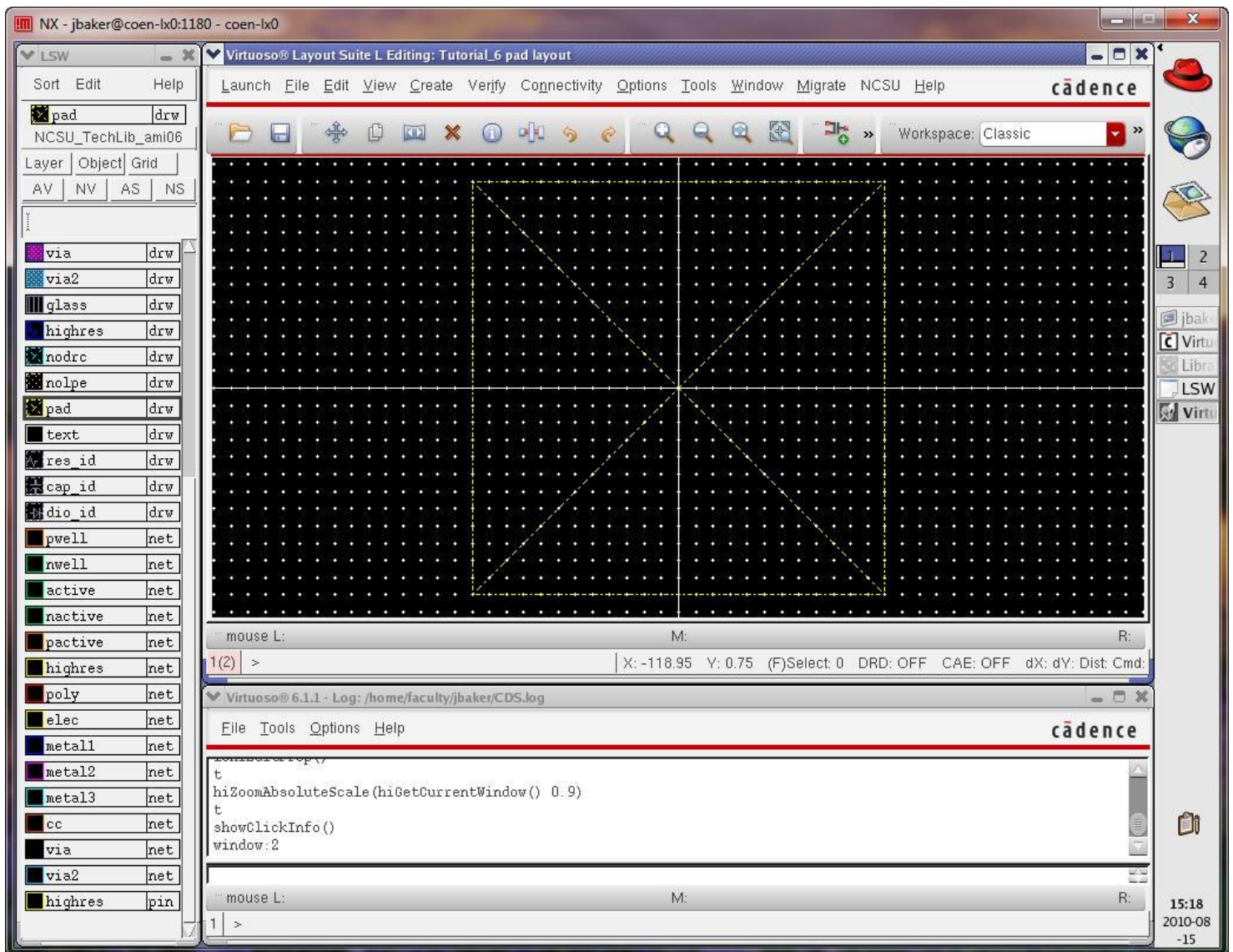
We'll use the pad layer as an outline for the cell. We could also use the text layer for a cell outline.



Next edit the rectangle's properties so that it is 120 um square centered upon the origin.



After fitting the layout we get the following.



The pad size (metal3) is 75 μm square with overglass (called glass, the opening in the top layer passivation) 6 μm smaller or a rectangle of 63 μm .

Note that both of these sizes, 75 and 63, are divisible by 0.15 (to avoid DRC errors by drawing layout off grid).

Add two rectangles on these layers, centered around 0,0 to get the layout of a pad.

DRC the layout to ensure no errors. Save the cell.

The screenshot displays the Cadence Virtuoso Layout Suite L Editing interface. The main workspace shows a grid with a central rectangular area defined by a dashed yellow border and a hatched green interior. The layer list on the left includes various layers such as glass, pwell, nwell, active, nactive, pactive, nselect, pselect, poly, elec, metal1, metal2, metal3, cc, via, via2, glass, highres, nodrc, nolpe, pad, text, res_id, cap_id, dio_id, pwell, and nwell. The terminal window at the bottom shows the following log output:

```

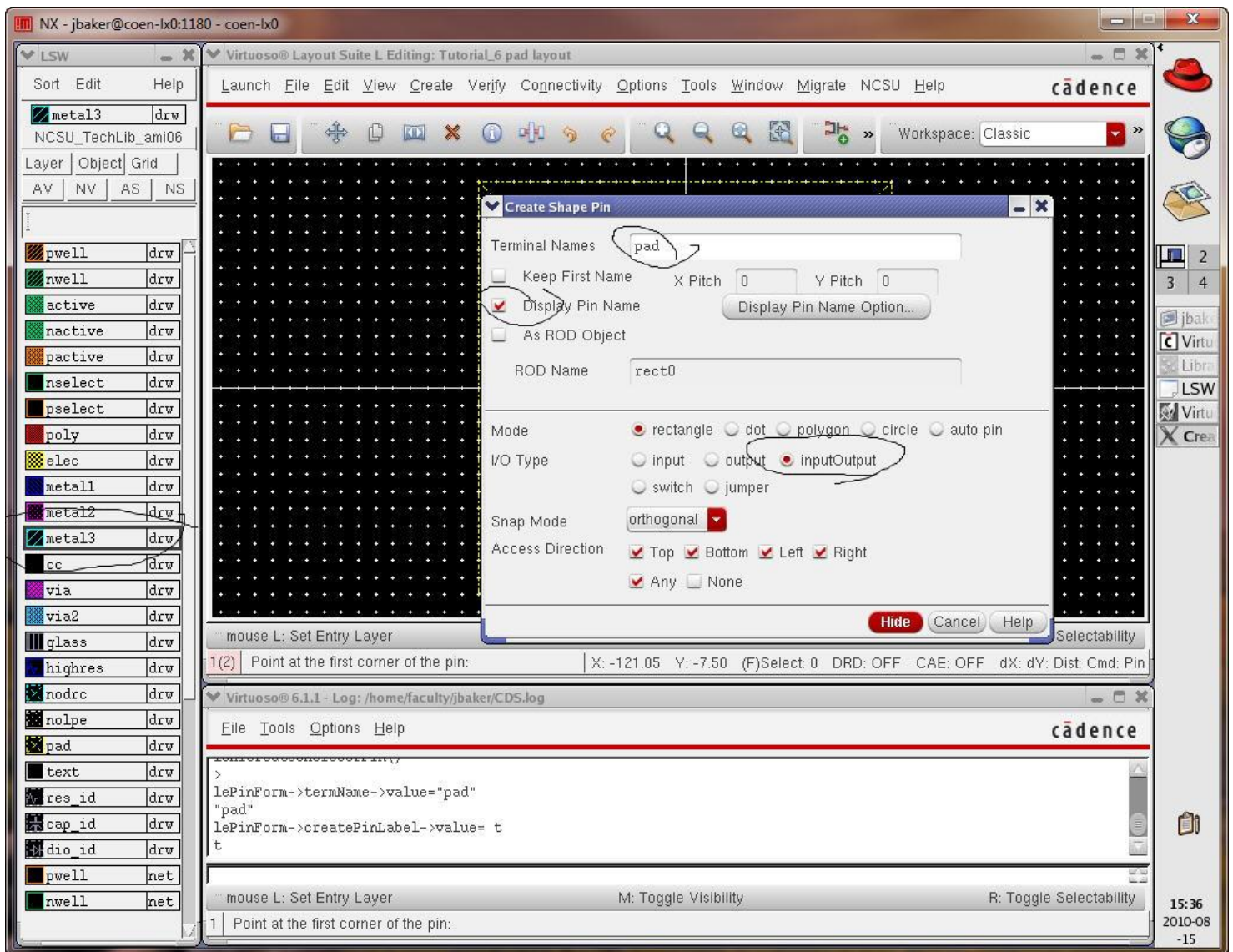
Virtuoso@6.1.1 - Log: /home/faculty/jbaker/CDS.log
File Tools Options Help
completed Sun Aug 15 15:29:39 2010
CPU TIME = 00:00:00 TOTAL TIME = 00:00:01
***** Summary of rule violations for cell "pad layout" *****
Total errors found: 0

```

Next instantiate a pin called "pad" with a direction inputOutput.

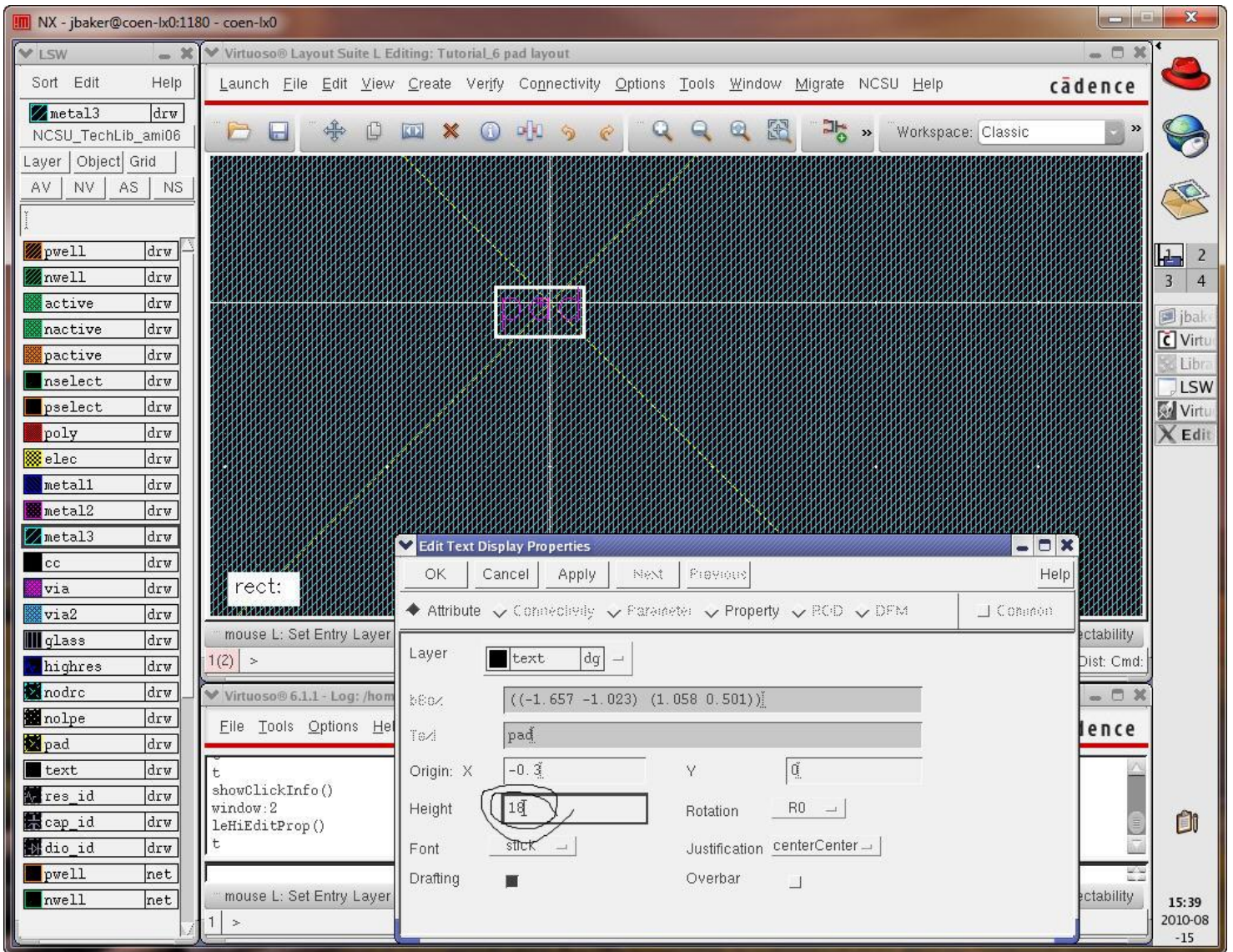
Use the metal3 layer and place the pin over the entire metal3 rectangle (so the size is 75 um square).

Also ensure Display Pin Name is selected.

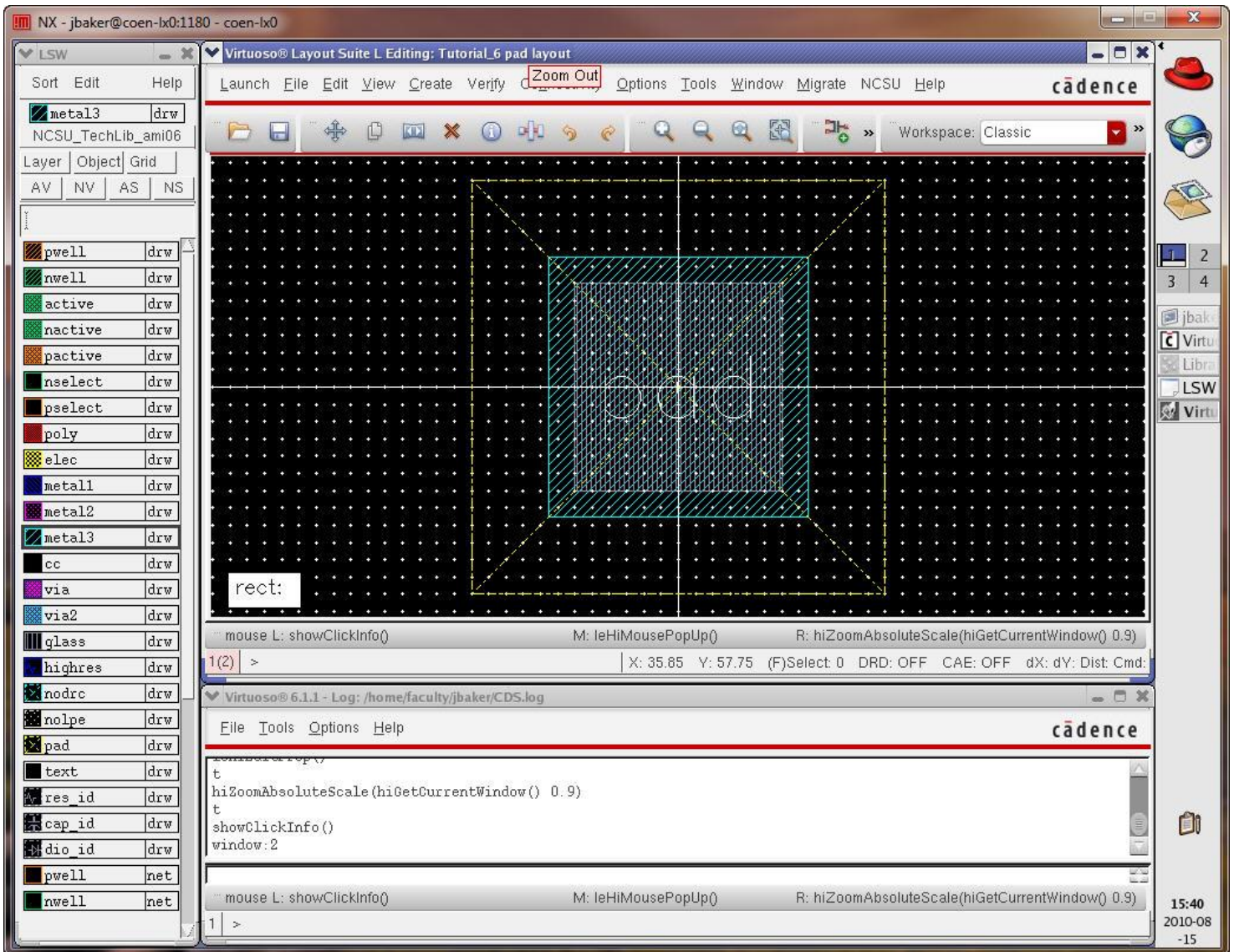


Zoom in and select the pin name and set its size to 18 um.

Note that if you want to change the pin's name (grayed out below) you have to select the rectangle on metal3 and *not* the pin name (text).

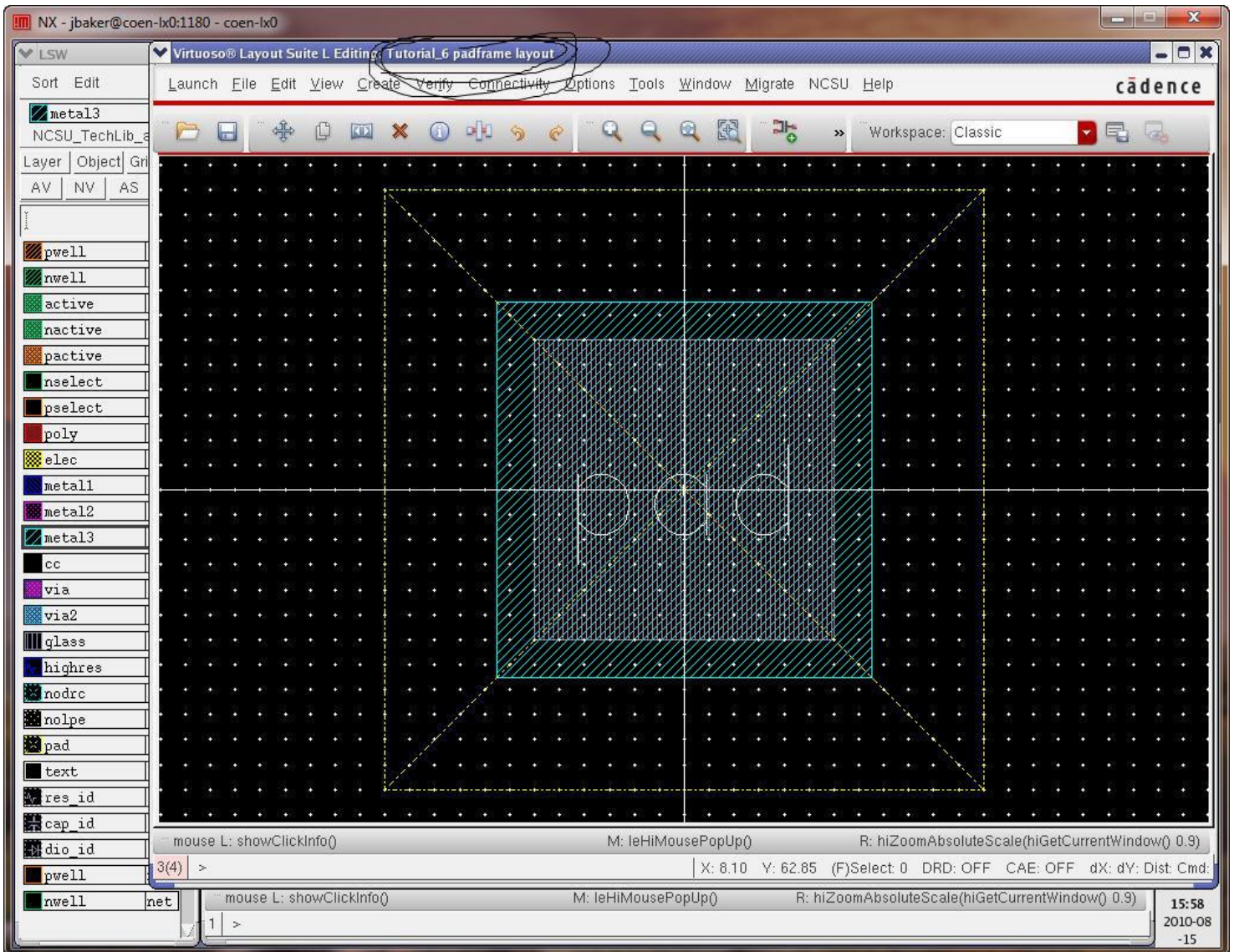


After fitting the layout we get the following

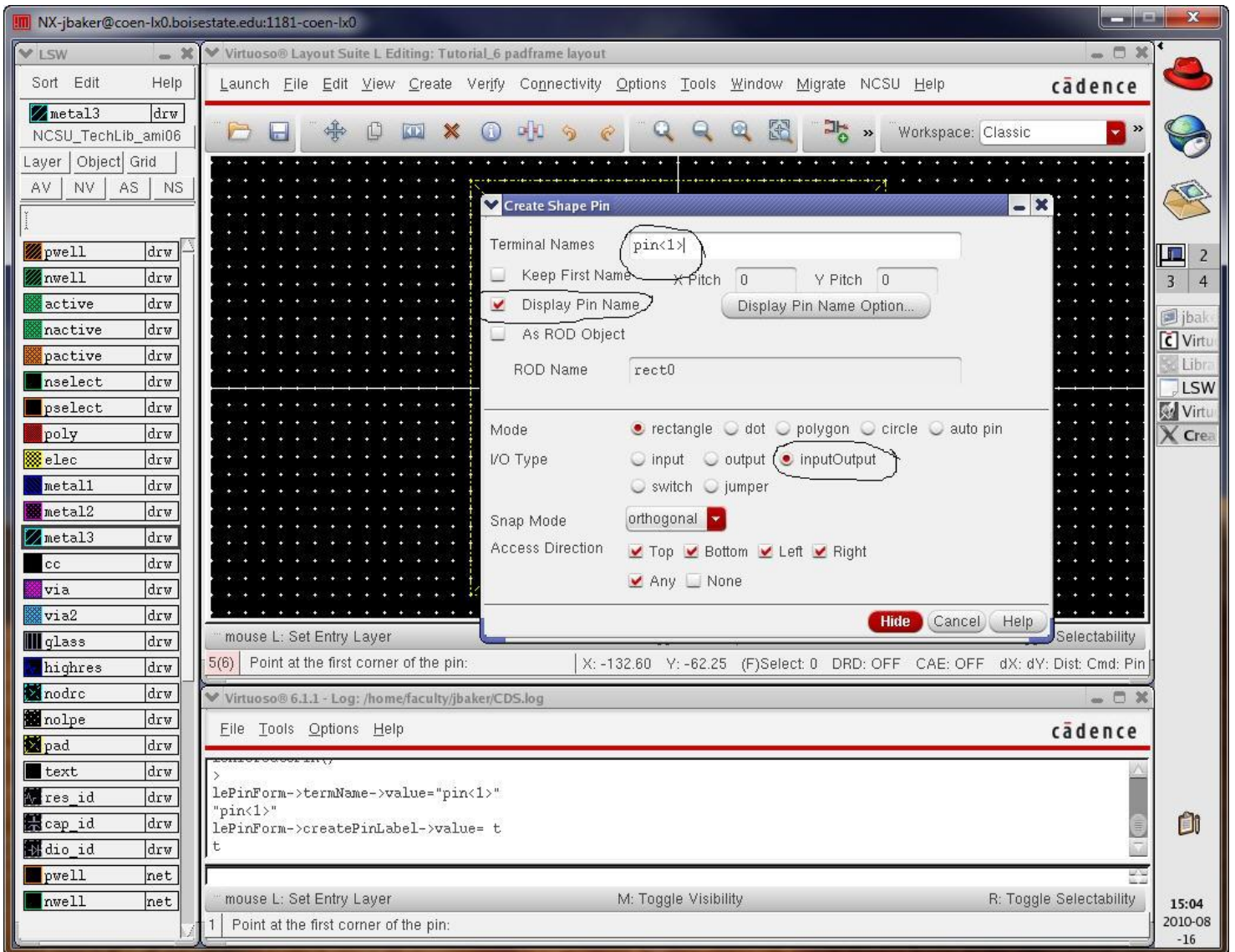


Save and DRC the layout.

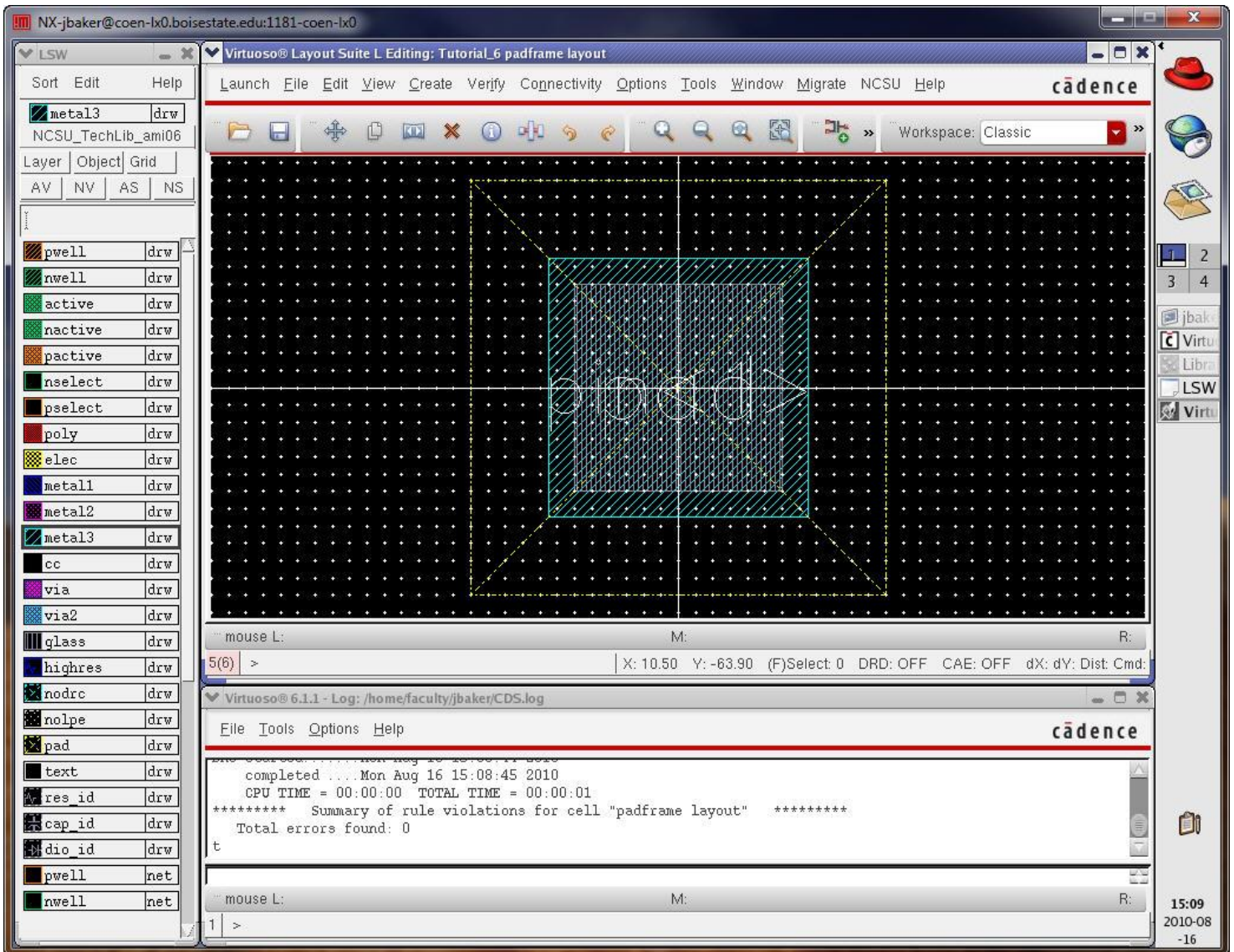
Next create a layout cellview called padframe and instantiate the pad cell into this new cell as seen below.



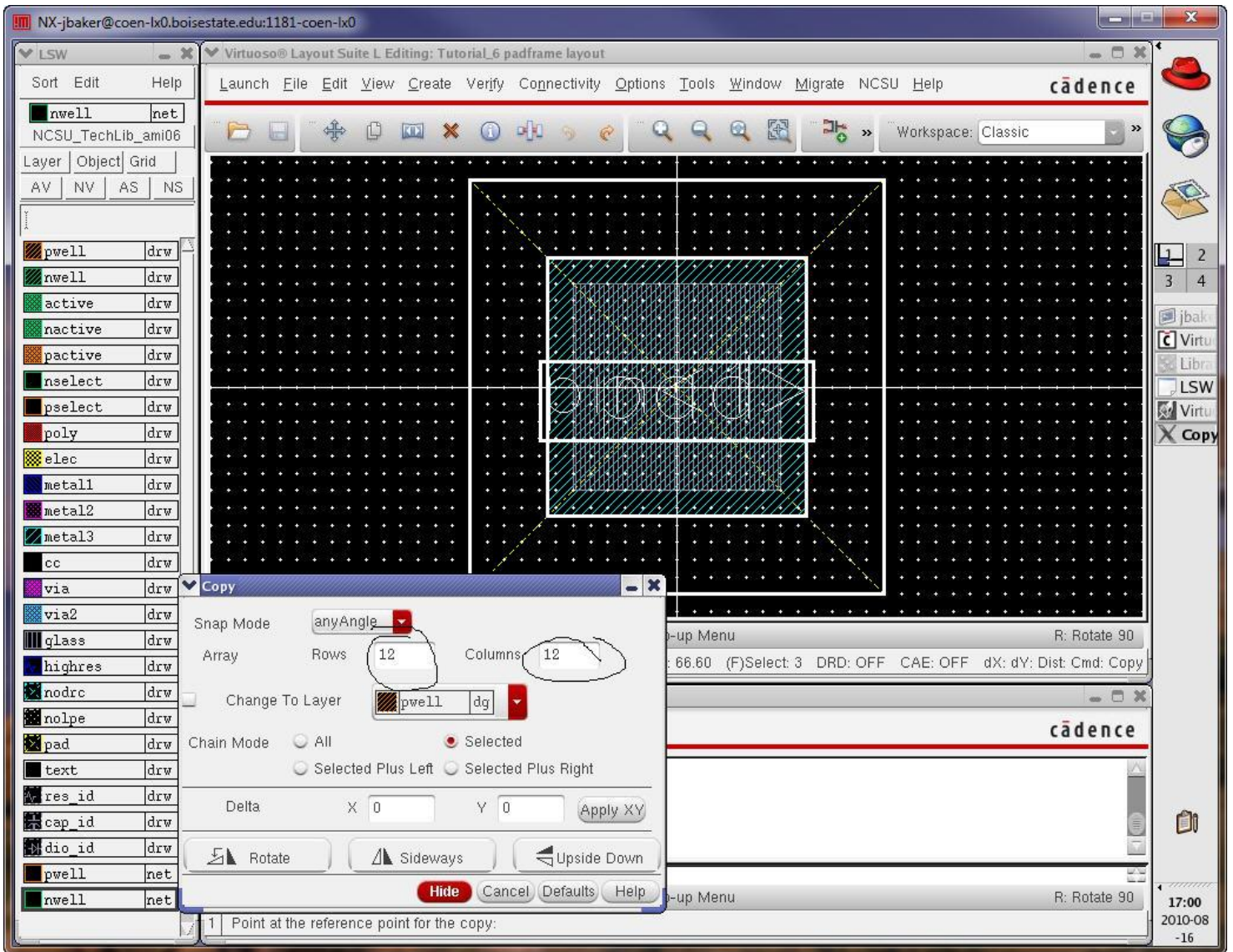
Add a pin on the metal3 layer called pin<1> with a direction of inputOutput.
Ensure that Display Pin Name is selected.



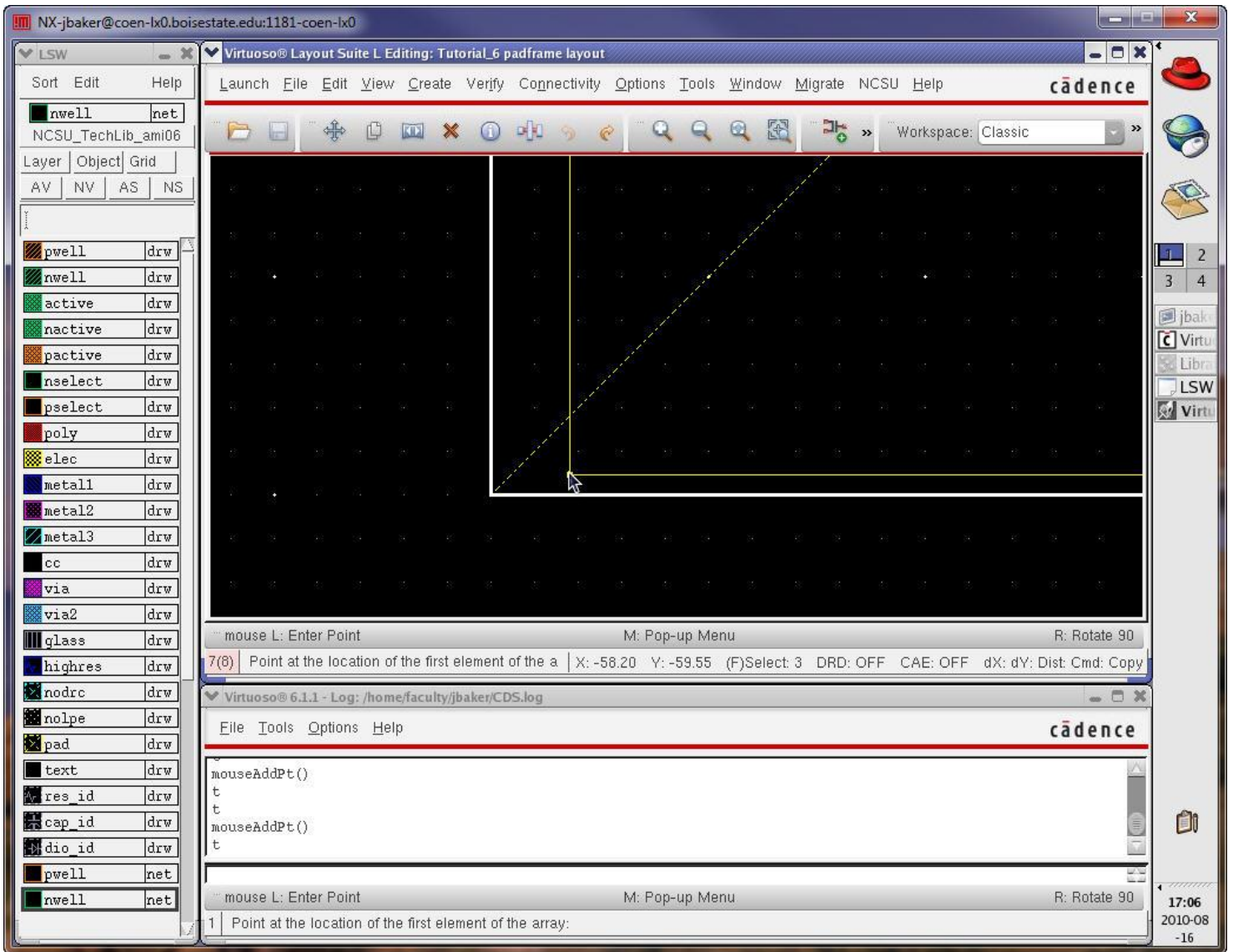
After setting the pin<1> text size to 15 um we get



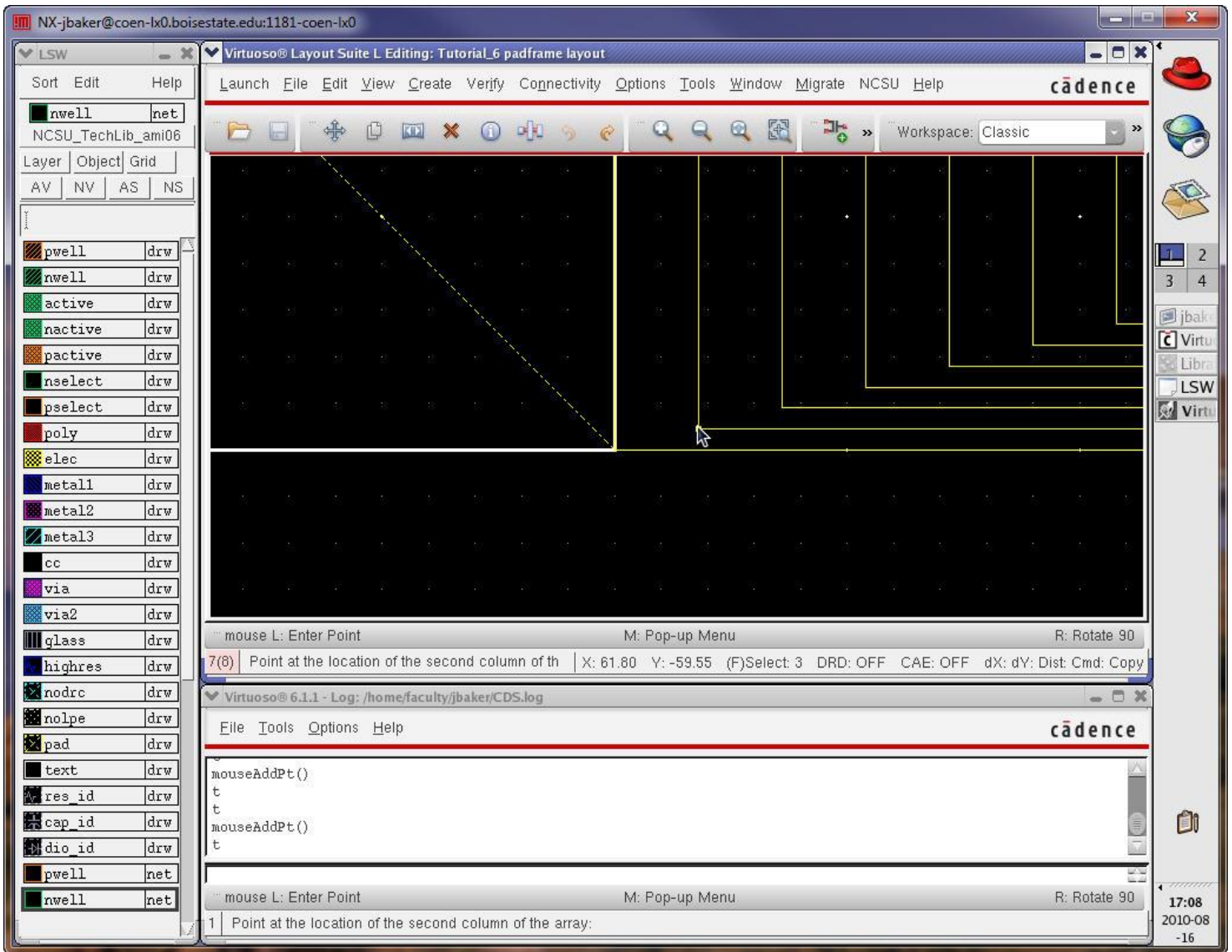
Next copy, c, the pad cell and pin then use F3 and set the number of rows and columns to 12.



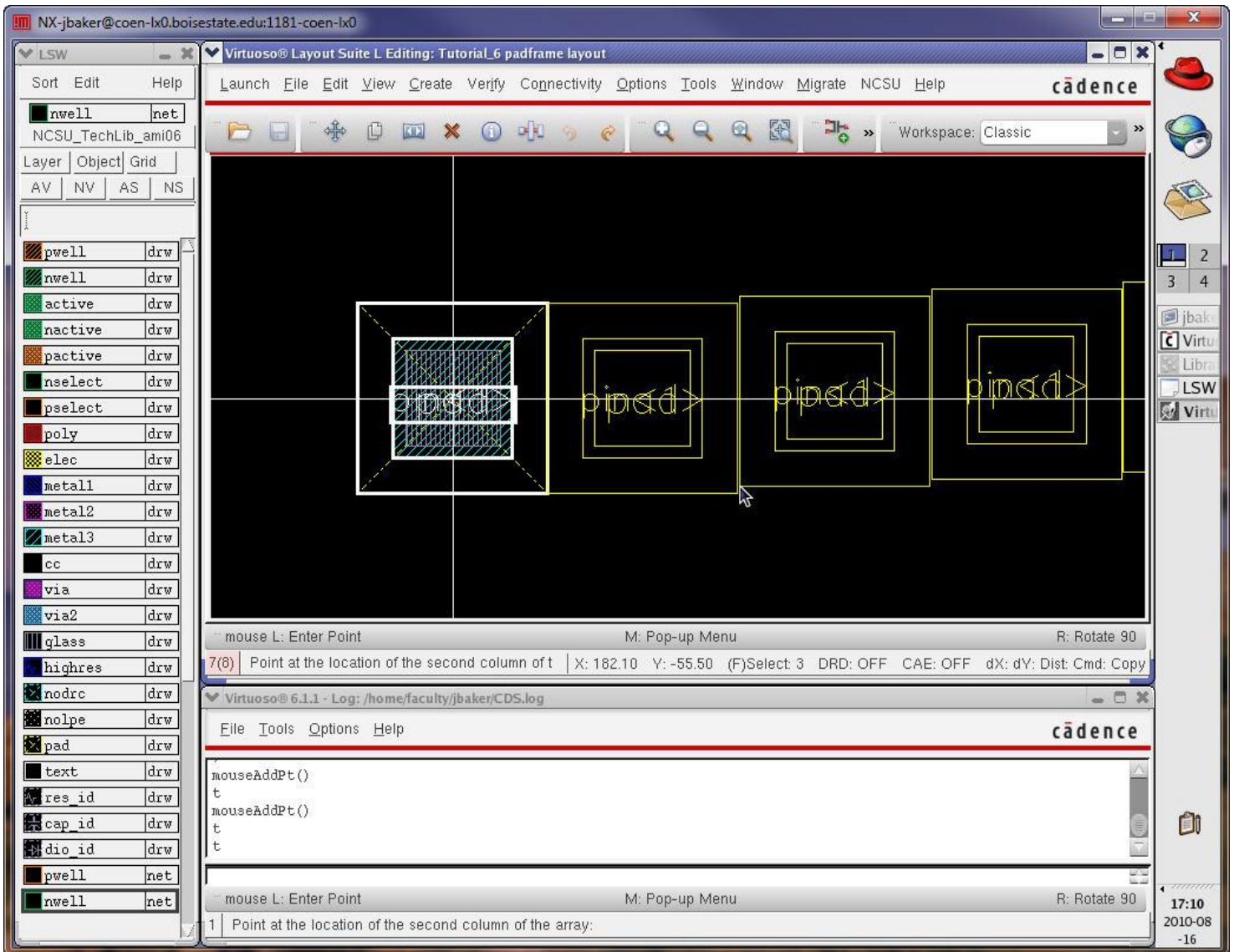
Select the lower left corner of the layout.



Next click on the lower right corner of the layout.



Following this click in the lower right corner of the first copied cell as seen below (zoom in to ensure that the cells are aligned properly).



Finally, click in the top, right, of the first copied cell to get the layout seen below.
It's okay to make mistakes and try again (hit **u** to undo and start over).

The screenshot displays the Cadence Virtuoso Layout Suite L Editing interface. The main window shows a grid of pads/pins in a padframe layout. The interface includes a menu bar (Launch, File, Edit, View, Create, Verify, Connectivity, Options, Tools, Window, Migrate, NCSU, Help) and a toolbar. The workspace is set to 'Classic'. The left panel shows a list of layers and objects, including 'nwell', 'pwell', '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 bottom status bar shows '7(8) Select the figure to be copied: | X: -438.00 Y: 1439.55 (F)Select: 0 DRD: OFF CAE: OFF dX: dY: Dist: Cmd: Copy'. The bottom right corner shows the system clock '17:14 2010-08 -16'.

Delete the middle, corner, and extra pads/pins to get the basic padframe seen below.
DRC and save the layout of the padframe.

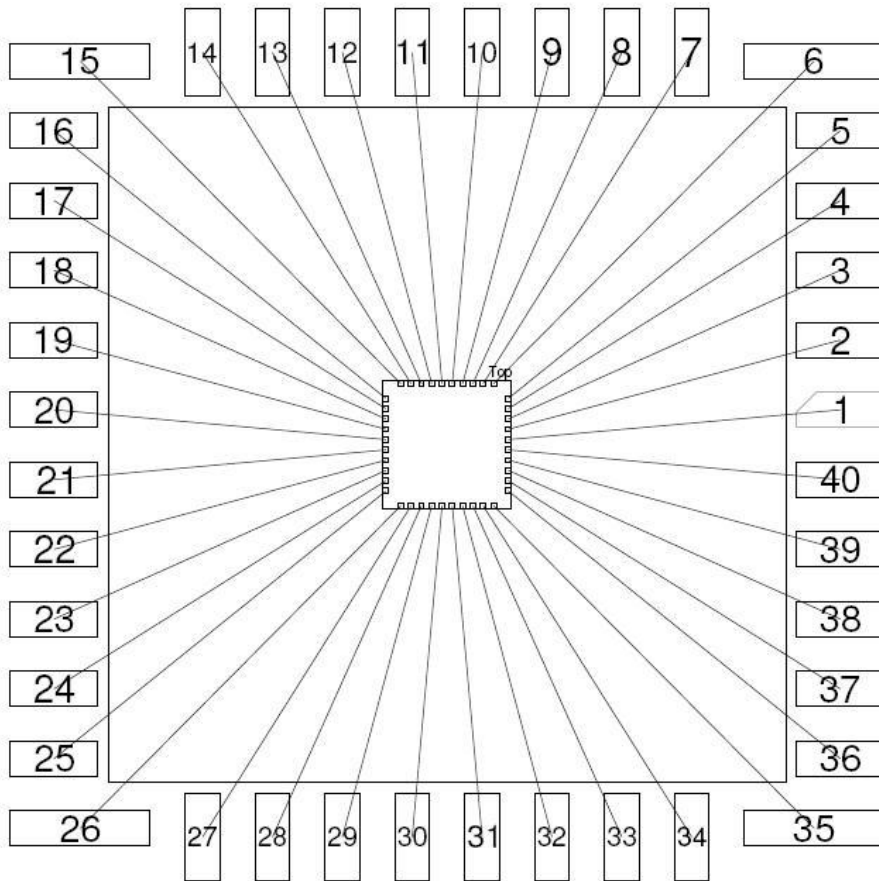
The screenshot shows the Cadence Virtuoso Layout Suite L Editing interface. The main window displays a padframe layout with a grid of pins. The left sidebar shows a layer list with various materials like nwell, pwell, active, nactive, pactive, nselect, pselect, poly, elec, metall, metal2, metal3, cc, via, via2, glass, highres, nodrc, nolpe, pad, text, res_id, cap_id, dio_id, pwell, and nwell. The bottom status bar shows coordinates and tool options. A terminal window at the bottom displays a log message: "completed ... Mon Aug 16 17:18:56 2010 CPU TIME = 00:00:00 TOTAL TIME = 00:00:00 ***** Summary of rule violations for cell 'padframe layout' ***** Total errors found: 0".

The padframe is likely bonded to the package with the configuration seen below (the bonding diagram).

All of the pins in the above layout are labeled pin1>.

Change the labels so that they match the bonding diagram seen below.

For example, pin 1 of the package corresponds to the 5th bonding pad from the top on the right side of the die.



The screenshot displays the Cadence Virtuoso Layout Suite L Editing interface. The main workspace shows a grid of components arranged in a rectangular pattern, representing a padframe layout. The components are highlighted with a dashed yellow border. The interface includes a menu bar (Launch, File, Edit, View, Create, Verify, Connectivity, Options, Tools, Window, Migrate, NCSU, Help) and a toolbar. The workspace is titled "Virtuoso@ Layout Suite L Editing: Tutorial_6 padframe layout".

On the left side, there is a layer list with the following items:

Layer	Object	Grid
nwell	net	
NCSU_TechLib_ami06		
AV	NV	AS
NS		

Below the layer list, there is a list of layers with their corresponding objects:

pwell	drw
nwell	drw
active	drw
nactive	drw
pactive	drw
nselect	drw
pselect	drw
poly	drw
elec	drw
metall	drw
metal2	drw
metal3	drw
cc	drw
via	drw
via2	drw
glass	drw
highres	drw
nodrc	drw
nolpe	drw
pad	drw
text	drw
res_id	drw
cap_id	drw
dio_id	drw
pwell	net
nwell	net

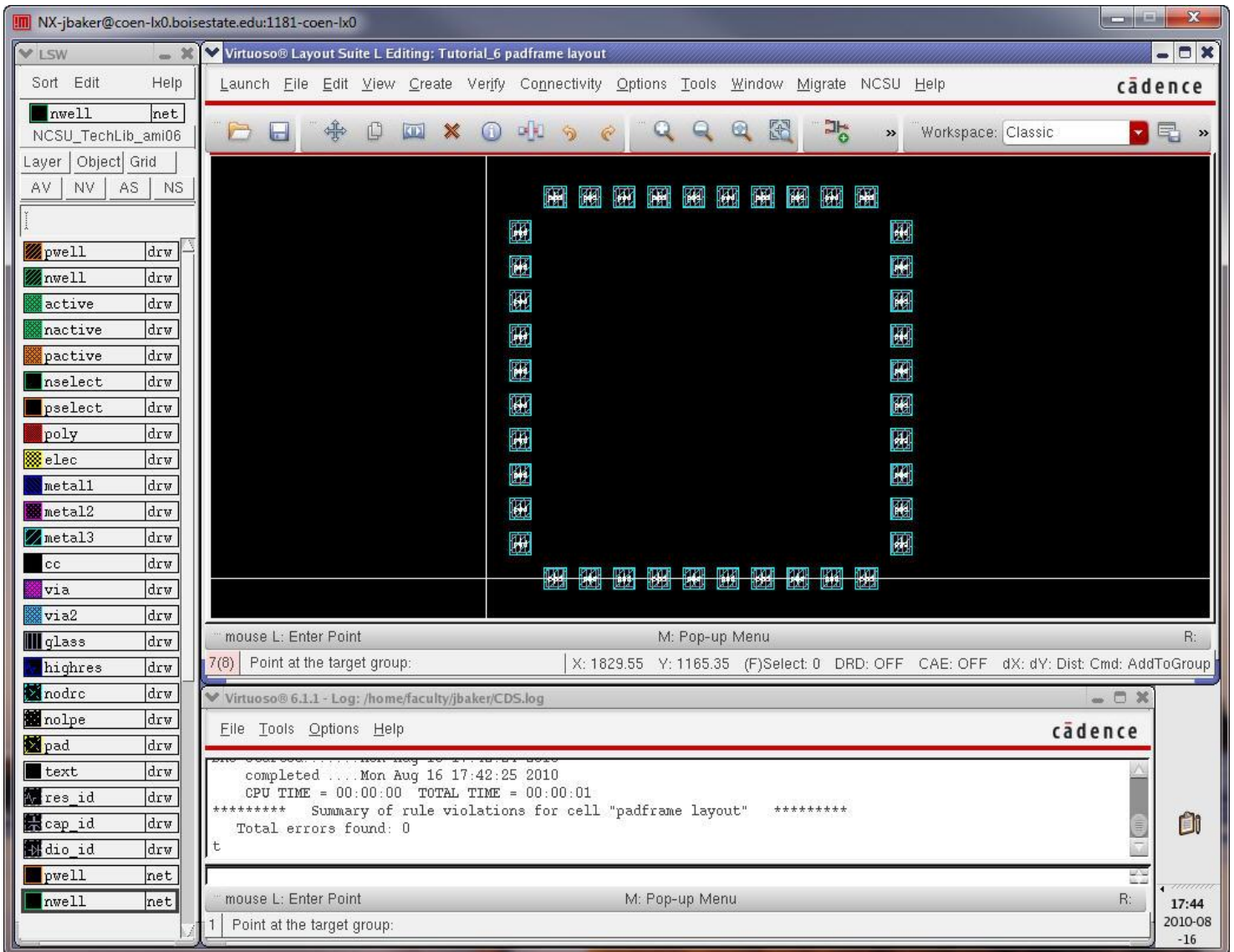
The terminal window at the bottom shows the following output:

```

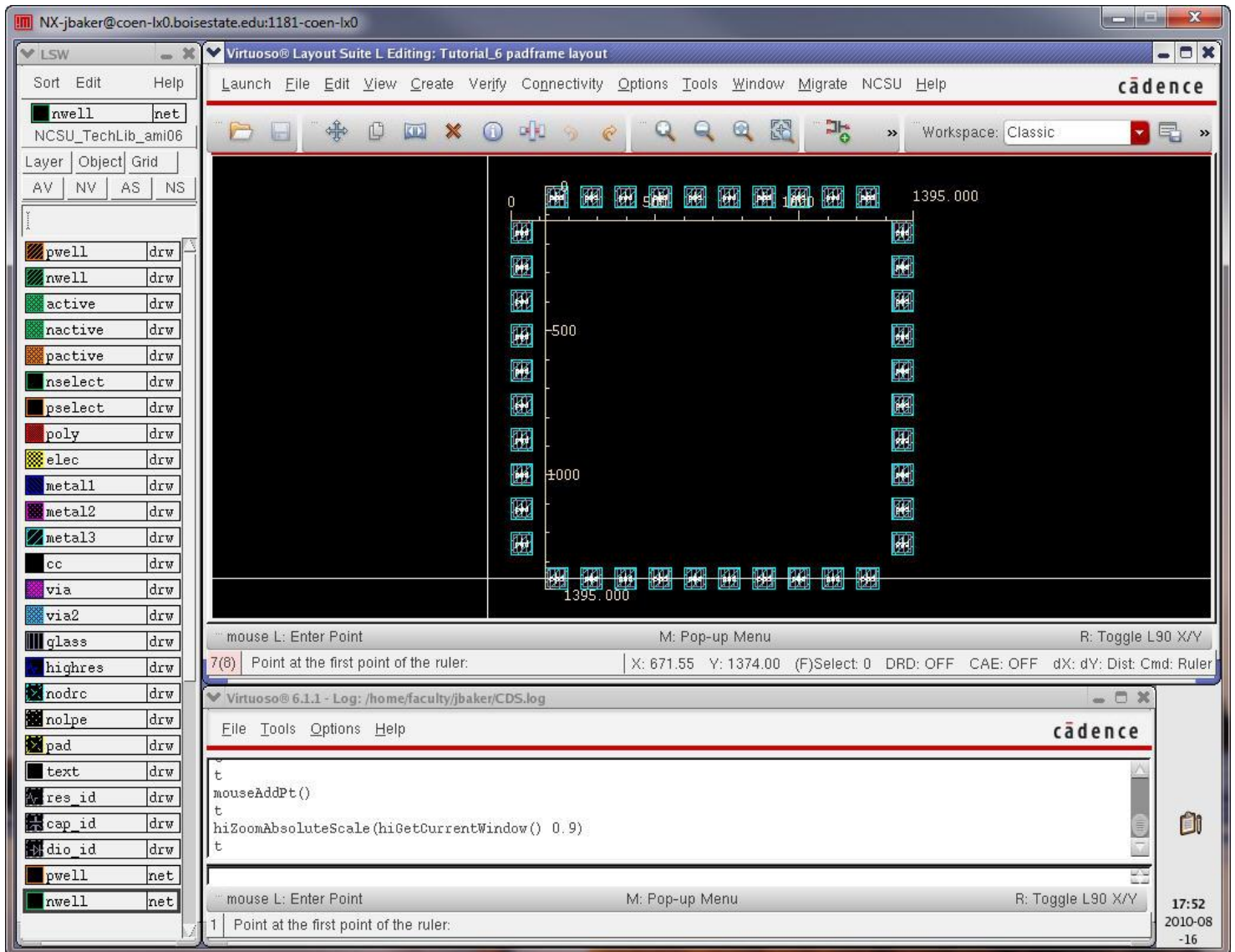
Virtuoso@ 6.1.1 - Log: /home/faculty/jbaker/CDS.log
File Tools Options Help
completed Mon Aug 16 17:30:00 2010
CPU TIME = 00:00:00 TOTAL TIME = 00:00:01
***** Summary of rule violations for cell "padframe layout" *****
Total errors found: 0
t

```

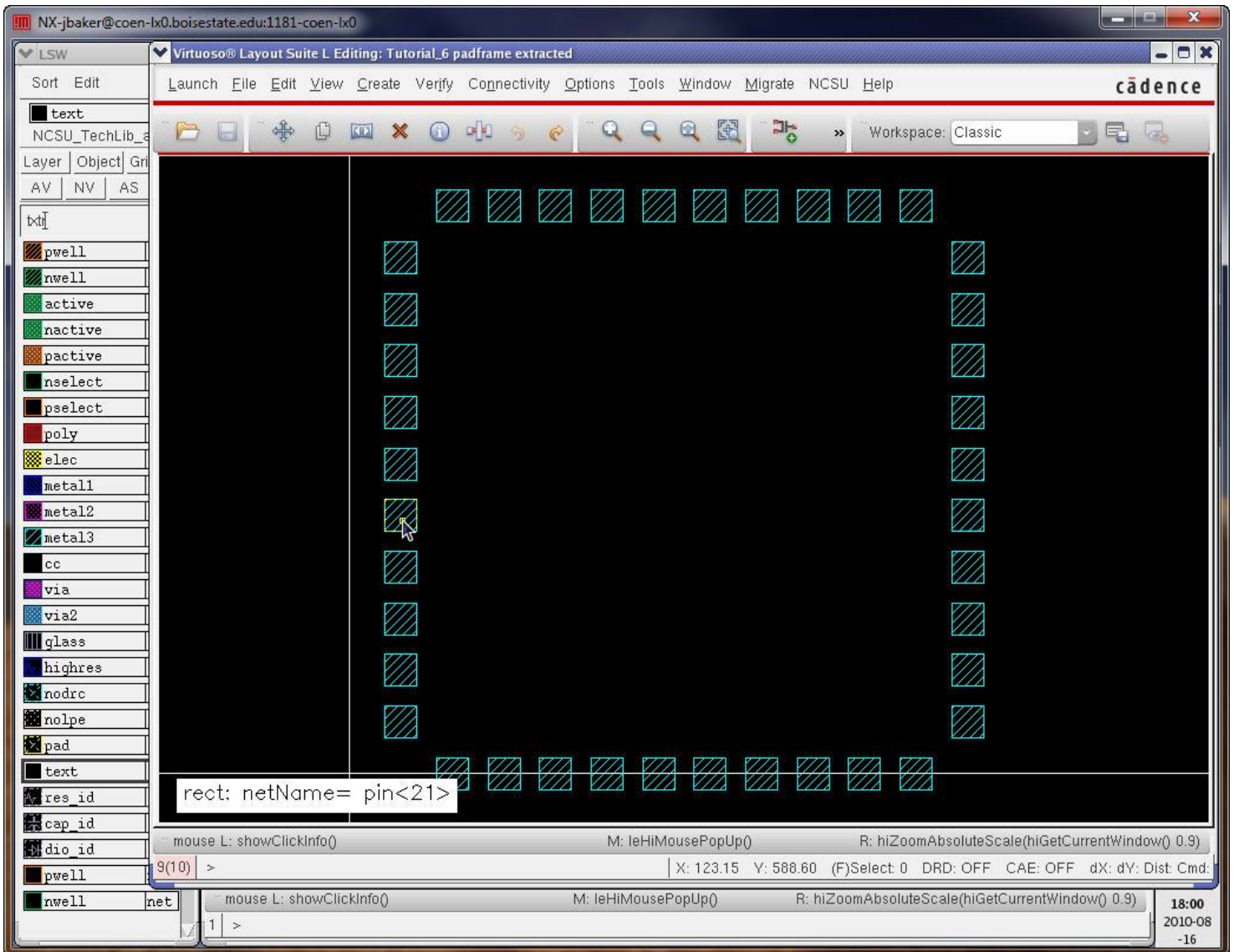
Remove the pad layer from the pad cell.



Let's measure the size of the padframe (our chip) since we'll need this information when we submit to MOSIS. Using the ruler we get the following sizes (1.395 mm square).



Before leaving the layout view of the padframe let's generate the extracted view.



The next thing we need to do is generate schematics for the pad and padframe layouts.

Generate a schematic cell view for the pad cell and add a pin, called pad, with a direction of inputOutput.

The screenshot shows the Cadence Virtuoso Schematic Editor interface. The main window is titled "Virtuoso@ Schematic Editor L Editing: Tutorial_6 pad schematic". A "Add Pin" dialog box is open in the center, with the following settings:

- Pin Names: pad
- Direction: inputOutput
- Usage: schematic
- Attach Net Expression: No
- Property Name: (empty)
- Default Net Name: (empty)
- Font Height: 0.0625
- Font Style: stick

Buttons at the bottom of the dialog include Rotate, Sideways, Upside Down, Show Sensitivity >>, Hide, Cancel, Defaults, and Help. The background shows a schematic editor window with a layer list on the left and a terminal window at the bottom. The terminal window shows the following text:

```

File Tools Options Help
>
hiiSetCurrentForm('schCreatePinForm)
t
schCreatePinForm->direction->value="inputOutput"
"inputOutput"

```

Check and Save the schematic.

Use Check -> find marker to ignore the warning that the pin is floating.

Check and Save the schematic again.

Note that an LVS can't be performed between the pad schematic and layout since there are no devices (resistors, capacitors, transistors) in the circuit.

The screenshot displays the Cadence Virtuoso Schematic Editor interface. The main workspace shows a schematic with the text "pad" and a red octagon. The left panel shows a list of layers, including "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 top menu bar includes "Launch", "File", "Edit", "View", "Create", "Check", "Options", "Migrate", "Window", "NCSU", and "Help". The bottom terminal window shows the following output:

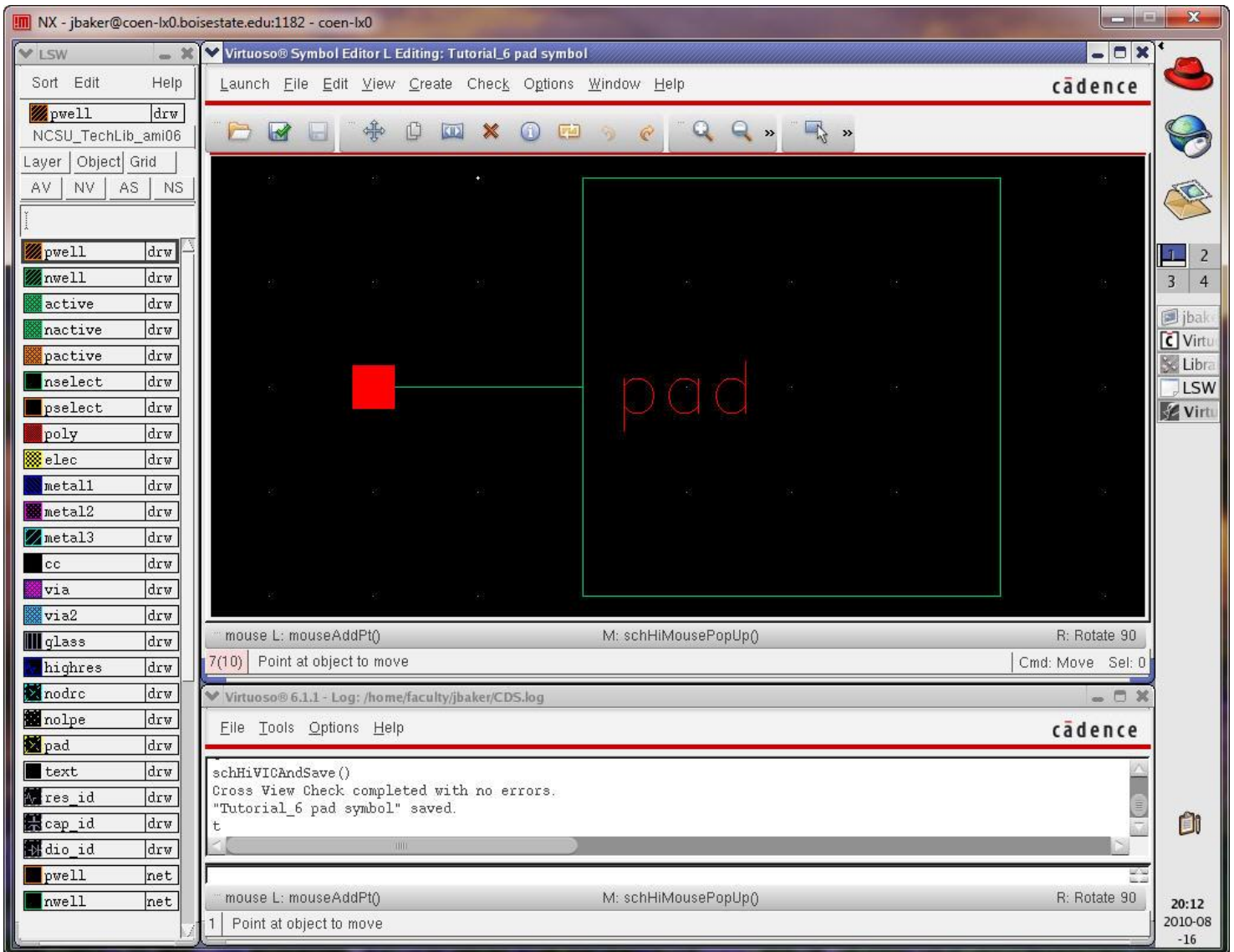
```
Virtuoso@6.1.1 - Log: /home/faculty/jbaker/CDS.log  
File Tools Options Help  
schHiCheckAndSave()  
Extracting "pad schematic"  
Schematic check completed with no errors.  
"Tutorial_6 pad schematic" saved.  
t
```

Next use Create -> Cellview -> From Cellview to create a symbol for the pad.

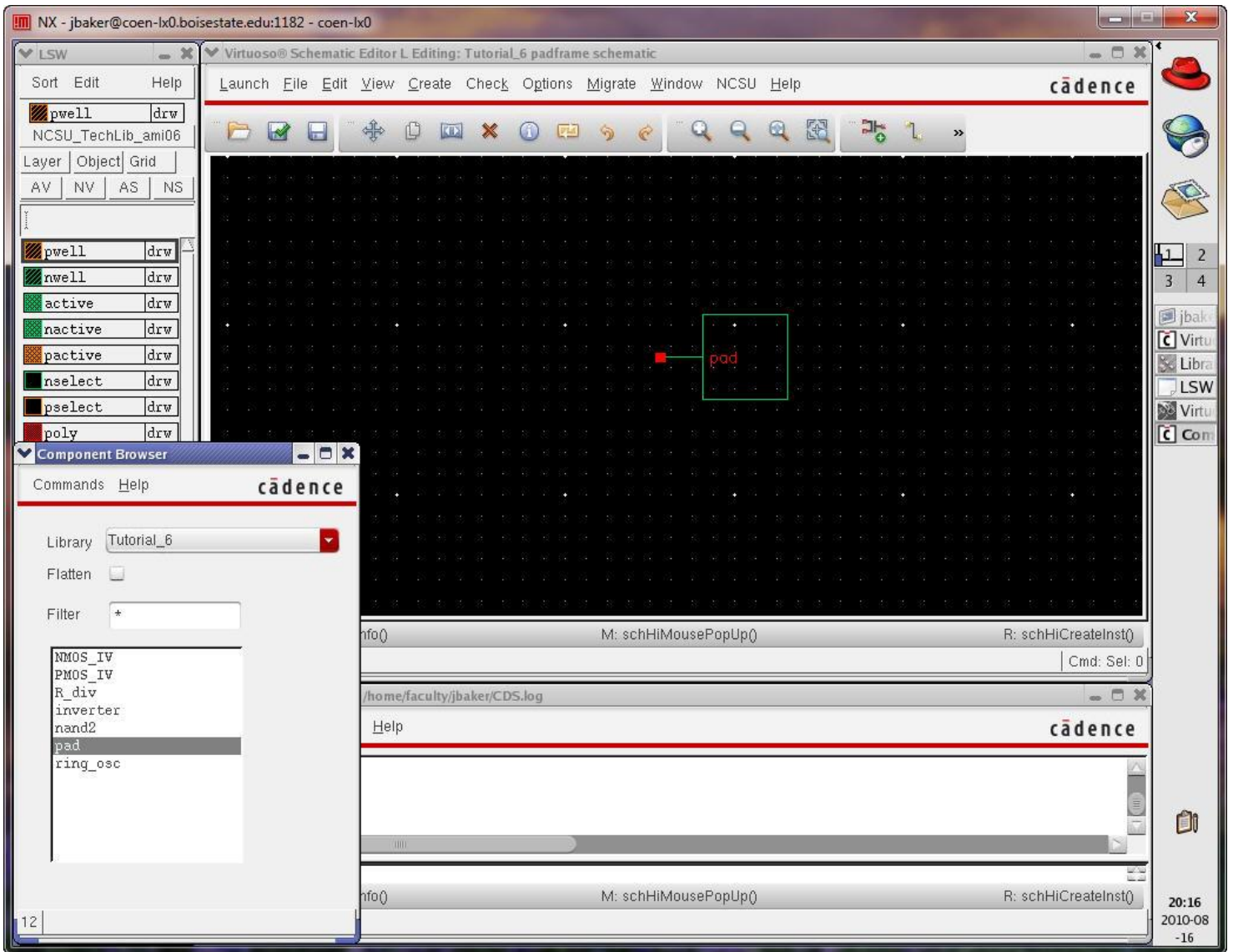
The screenshot displays the Cadence Virtuoso Symbol Editor interface. The main workspace shows a schematic diagram for a cellview named 'Tutorial_6 pad symbol'. The diagram consists of a central rectangular box containing three stacked 'cdsParam' objects, labeled 'cdsParam(1)', 'cdsParam(2)', and 'cdsParam(3)'. To the right of this box is a 'cdsName()' object. A red square is positioned at the top center of the box. The interface includes a menu bar with 'Launch', 'File', 'Edit', 'View', 'Create', 'Check', 'Options', 'Window', and 'Help'. A toolbar with various icons is located below the menu bar. On the left side, there is a 'LSW' (Library Symbol Window) panel showing a list of objects and their types (e.g., 'pwell' drw, 'NCSU_TechLib_ami06', 'Layer', 'Object', 'Grid', 'AV', 'NV', 'AS', 'NS'). Below the main workspace, there is a command line showing '7(10) >' and a log window titled 'Virtuoso@6.1.1 - Log: /home/faculty/jbaker/CDS.log' with the following text: 'Getting schematic propert bag', 't', 'schZoomFit(1.0 0.9)', 't'. The bottom right corner of the window shows the time '18:49', date '2010-08', and page number '-16'.

Delete most items and rotate to get the following.

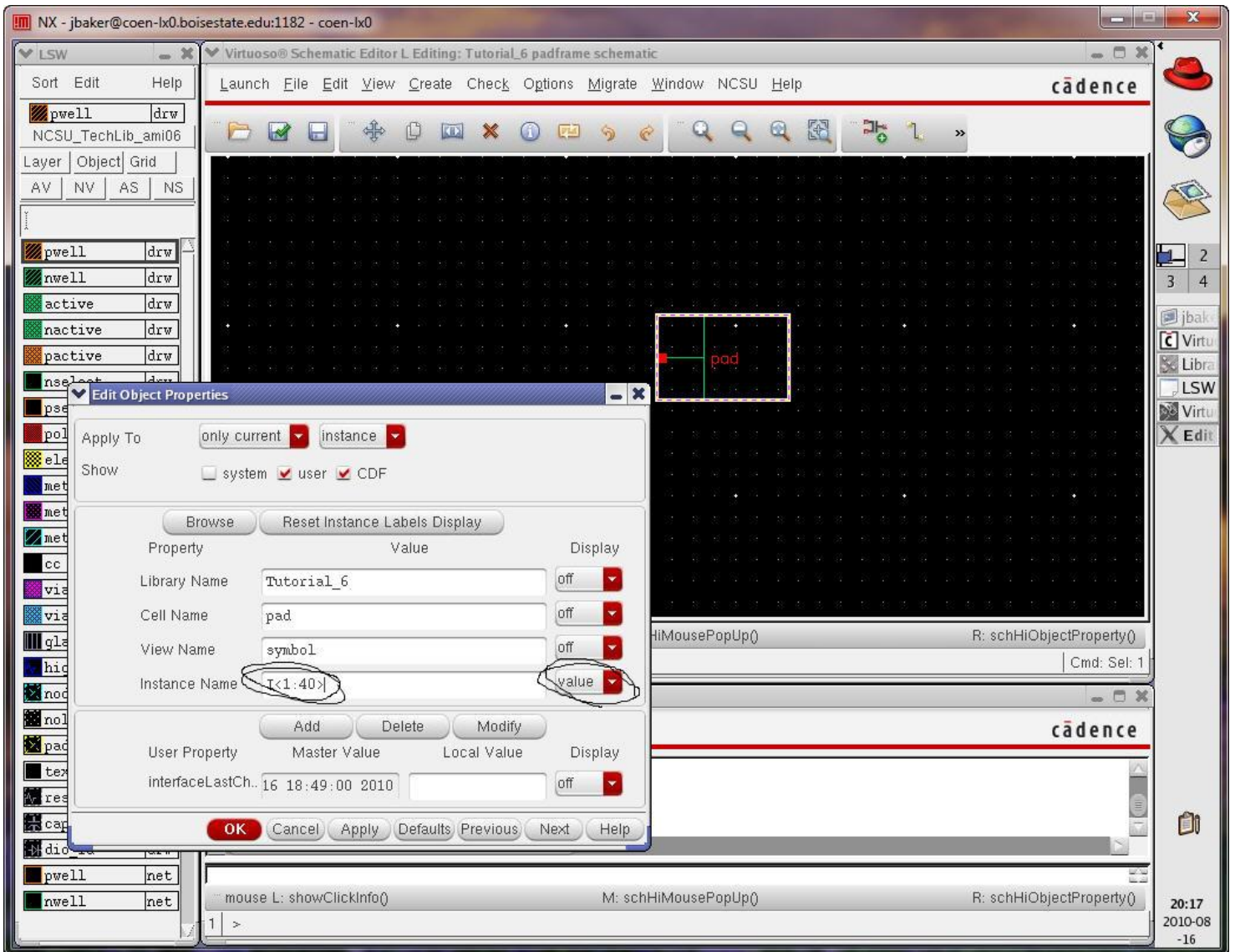
Save the pad symbol cellview



Create a schematic cellview for the padframe and instantiate the pad cell.



Array the instance name and ensure the Display is set to value as seen below.



The screenshot shows the Cadence Virtuoso Schematic Editor interface. The main window displays a schematic diagram with a red square on the left and a green rectangle labeled "pad" on the right. The text "I<1:40>" is visible in the top left of the schematic area. The interface includes a menu bar, a toolbar, and a layer list on the left. A console window at the bottom shows the command "schZoomFit(1.0 0.9)".

Add a pin with a name of pin<1:40> and a direction of inputOutput to match the layout of the padframe.

The screenshot displays the Cadence Virtuoso Schematic Editor interface. The main window is titled "Virtuoso@ Schematic Editor L Editing: Tutorial_6 padframe schematic". A dialog box titled "Add Pin" is open, showing the following settings:

- Pin Names: pin<1:40>
- Direction: inputOutput
- Bus Expansion: off
- Usage: schematic
- Placement: single
- Attach Net Expression: No
- Property Name: (empty)
- Default Net Name: (empty)
- Font Height: 0.0625
- Font Style: stick

Buttons at the bottom of the dialog include Rotate, Sideways, Upside Down, Show Sensitivity >>, Hide, Cancel, Defaults, and Help. The background schematic shows a red square and a yellow arrow pointing to the dialog. The interface includes a menu bar (Launch, File, Edit, View, Create, Check, Options, Migrate, Window, NCSU, Help), a toolbar, and a layer/object grid panel on the left.

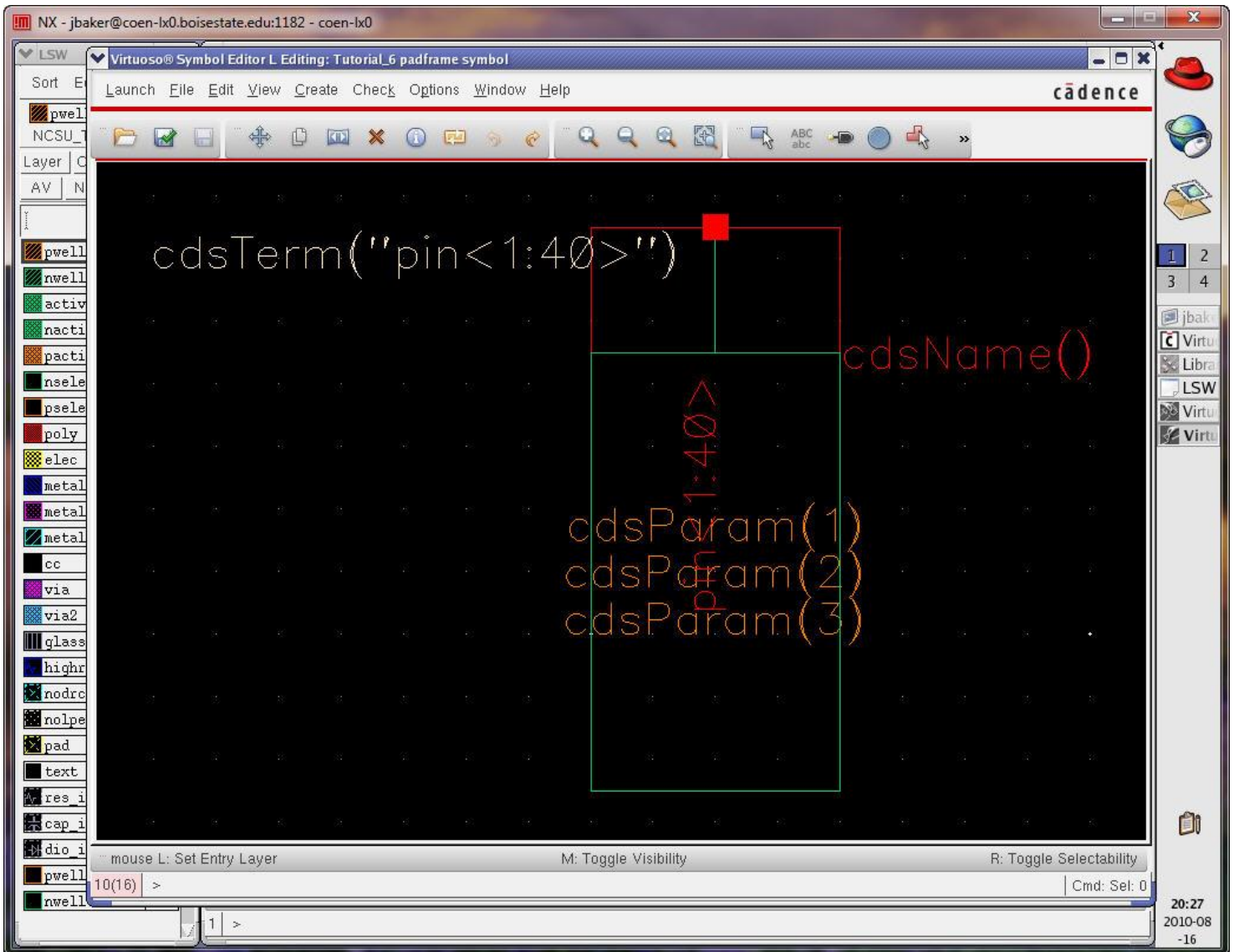
Add a wide wire with a label pin<1:40>

Check and Save the schematic view of the padframe.

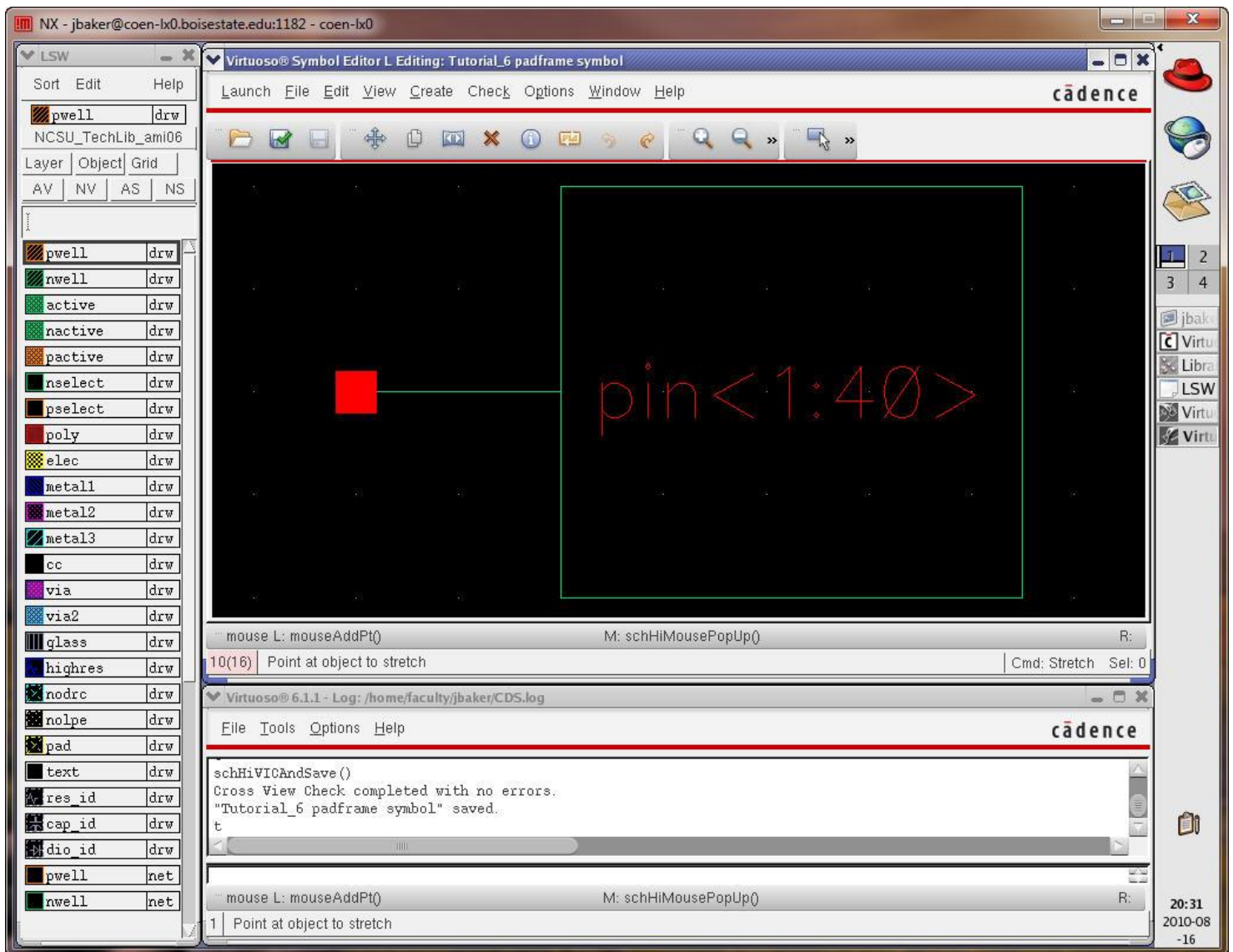
Note that an LVS can't be performed between the padframe schematic and layout since there are no devices in the circuit.

The screenshot displays the Cadence Virtuoso Schematic Editor interface. The main window shows a schematic diagram with a red diamond symbol labeled "pin<1:40>", a blue wire labeled "pin<1:40>", and a green box labeled "pad". The text "I<1:40>" is also visible. The left sidebar shows a list of layers and objects, including "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 bottom window shows a log message: "Extracting 'padframe schematic' Schematic check completed with no errors. 'Tutorial_6 padframe schematic' saved." The system tray on the right shows the date and time: "20:24 2010-08 -16".

Next let's create a symbol for the padframe using Create -> Cellview -> From Cellview



Again, delete most items, stretch, and rotate to get the following.
Check and Save the symbol view.

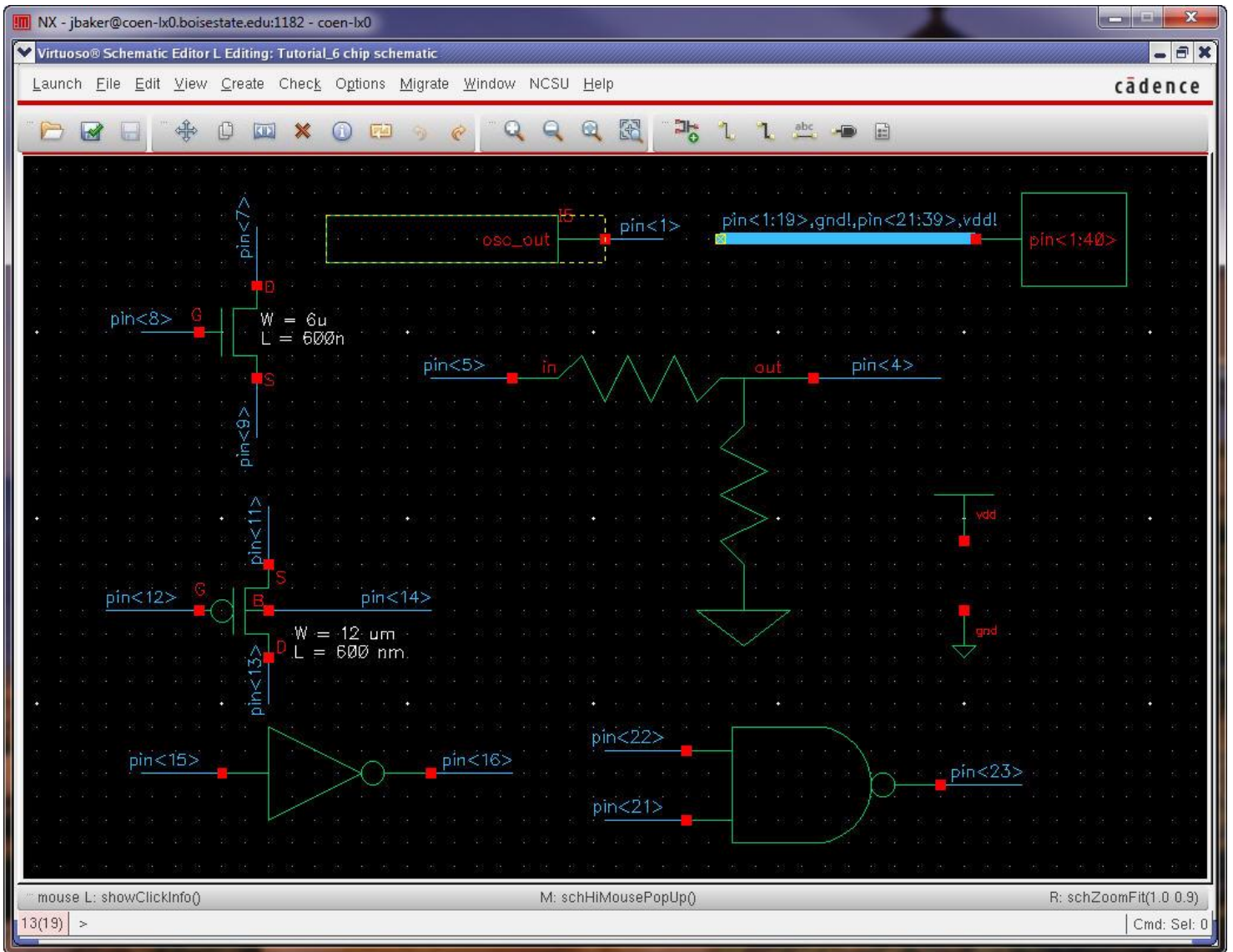


Okay, we are ready to create our chip layout.

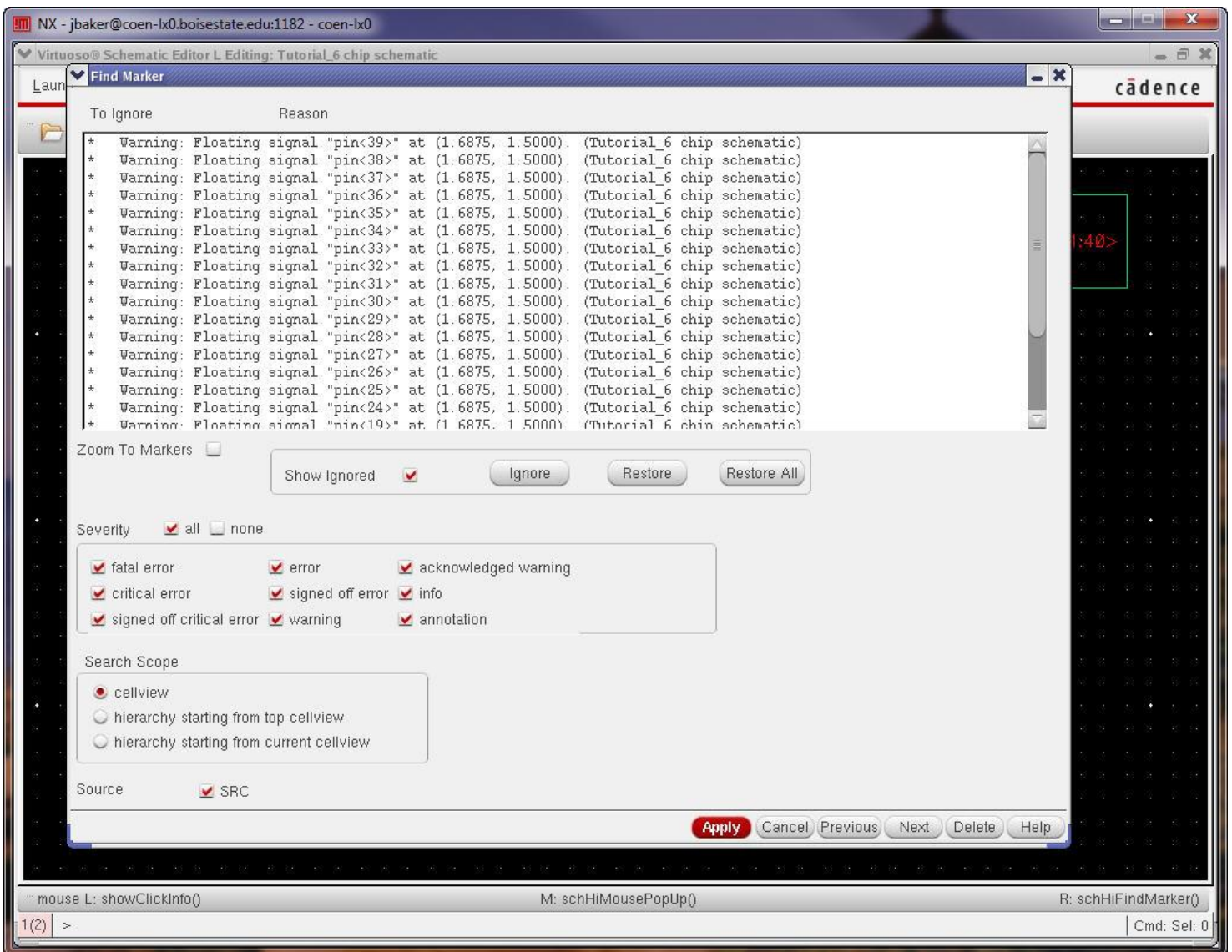
Create a schematic view of a cell called "chip" and add the cells we've created in the tutorial.

Wire the cells up as seen (an arbitrary connection).

Note how we've used pin<20> for ground and pin<40> for power.



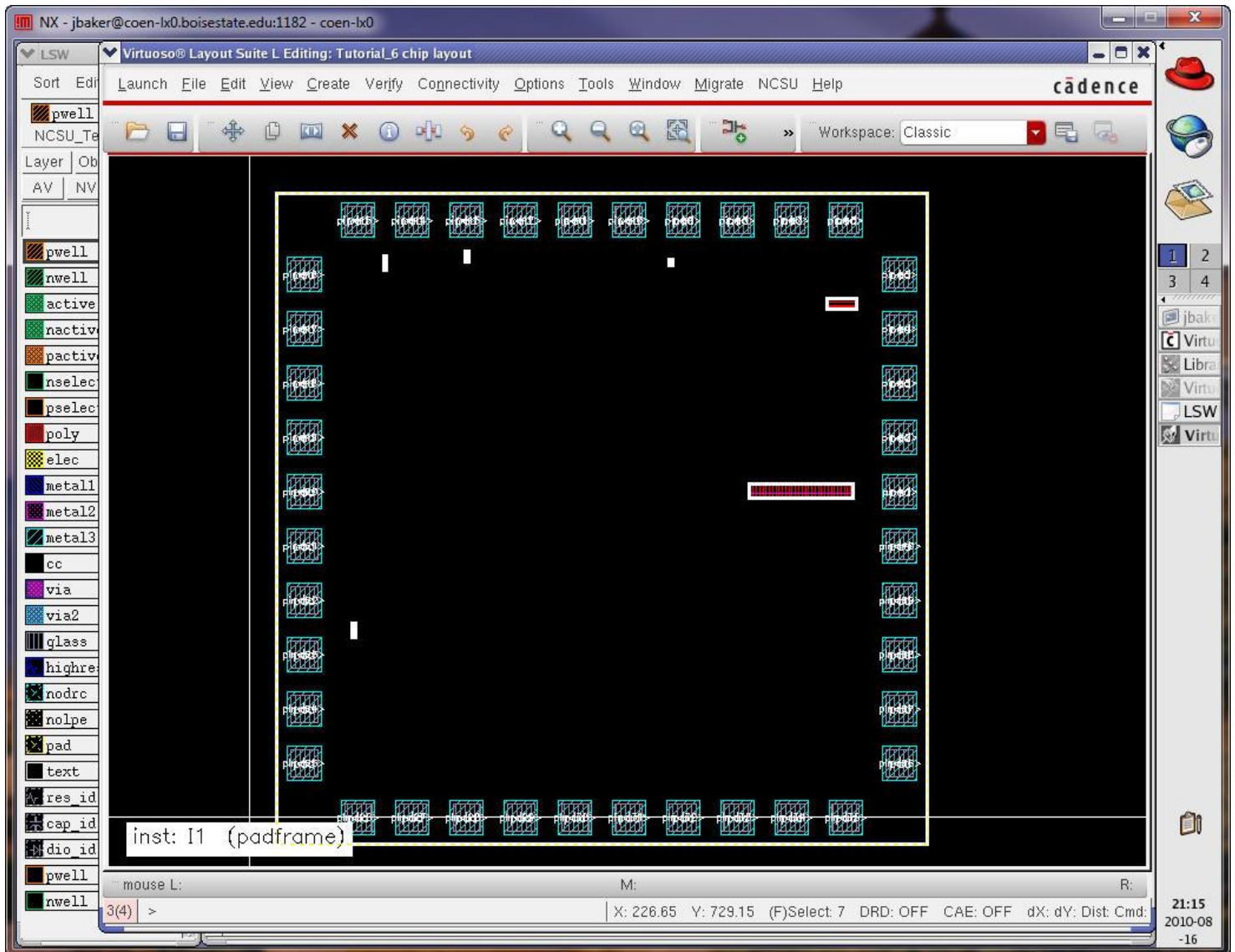
Check and Save the chip schematic.



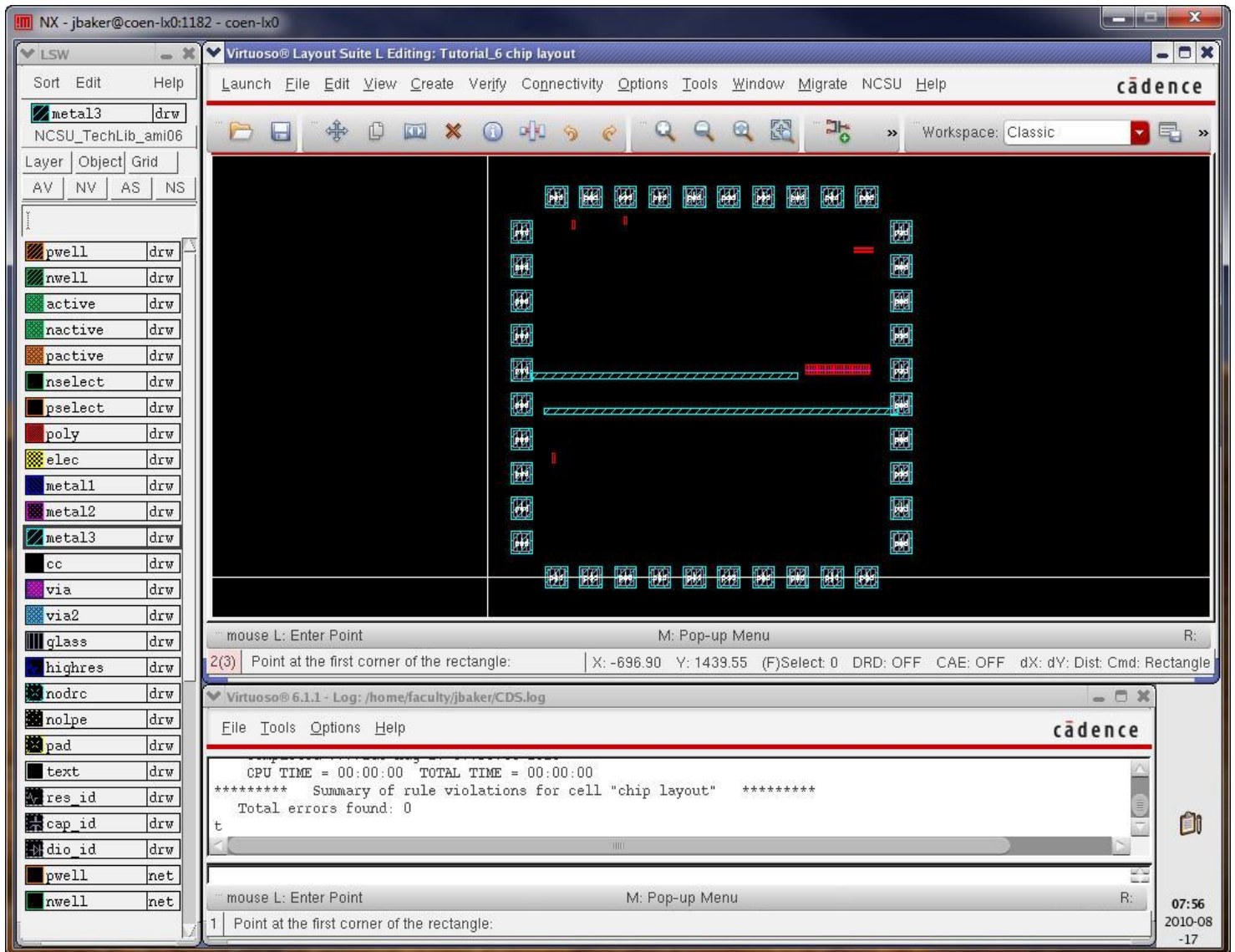
Notice that all of the unused pins on the padframe are floating.
Ignore these warnings and then Check and Save again.

We are ready to connect the cells up to the padframe in the layout.
Create of layout view of the "chip" cell.

Place the R_div, NMOS_IV, PMOS_IV, inverter, nand2, and ring oscillator cells near the pads that they will connect to, below.
We can adjust the cells' positions as needed in a moment.



We won't try to make the layouts pretty but rather we'll focus on wiring the circuits to the padframe quickly. Let's begin by drawing rectangles on metal3 for connections to vdd! and gnd!



Next let's add rectangles on the metal layers to move towards connecting the cells up to the padframe.

Below are examples zoomed in around the ring oscillator and resistive divider.

Again, we are not trying to make a "pretty" layout (or a good one for that matter) but rather just connect the circuitry up.

It's useful to DRC the layout as you go to fix the errors that will occur.

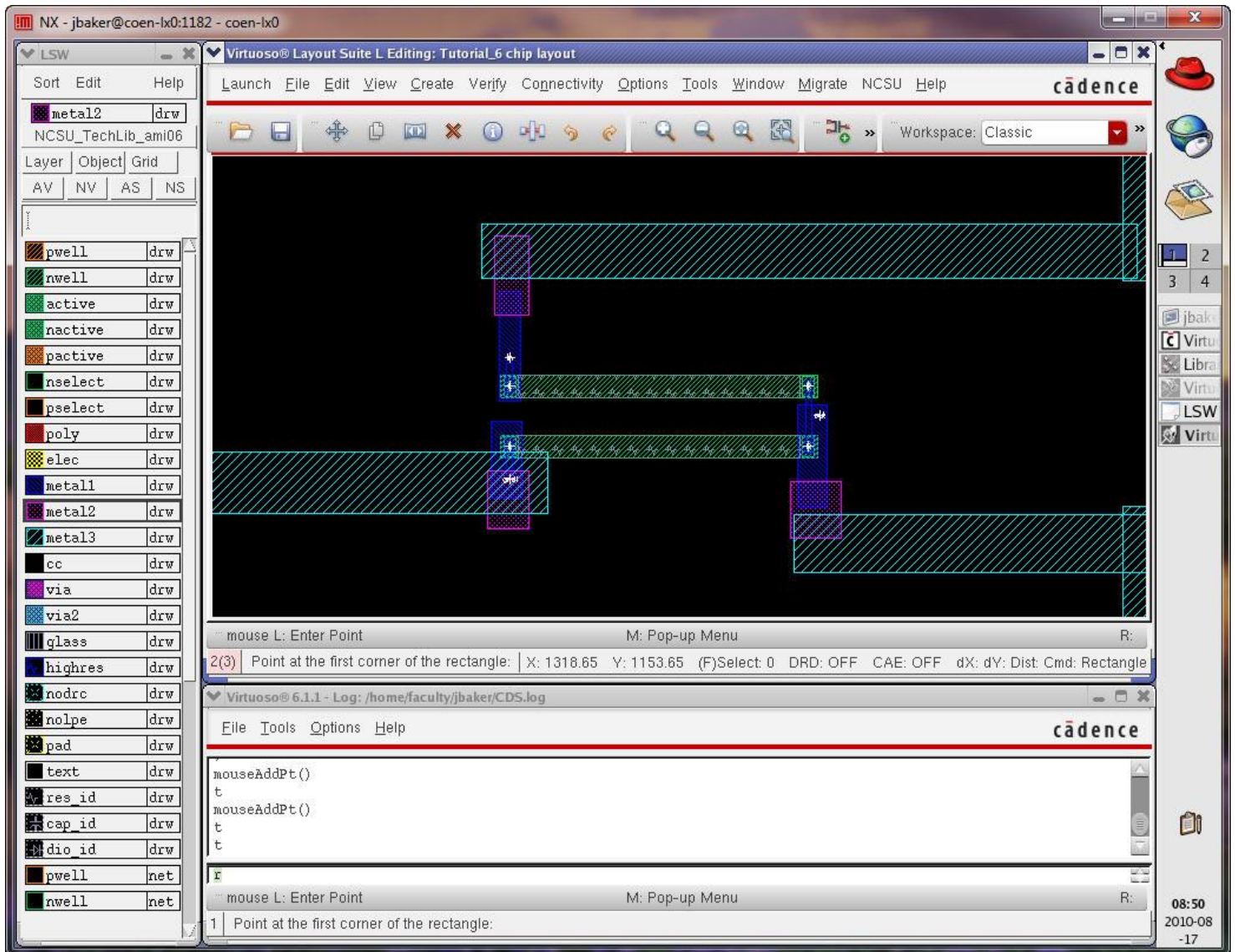
The screenshot displays the Cadence Virtuoso Layout Suite L Editing interface for a project named "Tutorial_6 chip layout". The main workspace shows a complex layout with various layers and patterns, including labels like "vdd!", "gnd!", and "A0csfout". A toolbar at the top provides navigation and editing tools. On the left, a 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".

At the bottom, a log window titled "Virtuoso@6.1.1 - Log: /home/faculty/jbaker/CDS.log" displays the following text:

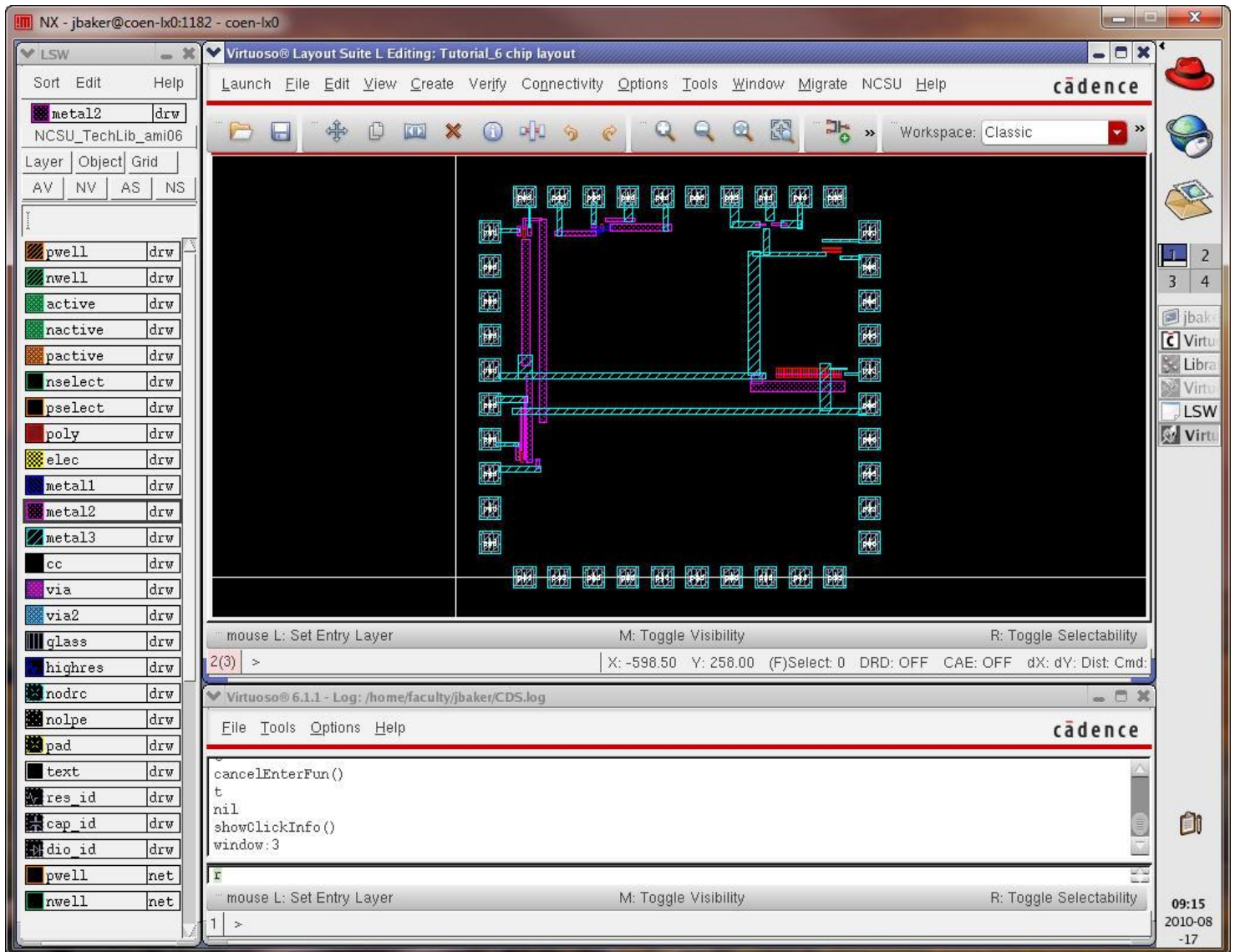
```
File Tools Options Help
completed Tue Aug 17 08:50:14 2010
CPU TIME = 00:00:00 TOTAL TIME = 00:00:00
***** Summary of rule violations for cell "chip layout" *****
Total errors found: 0
t
```

Navigation and control elements include mouse L: Set Entry Layer, M: Toggle Visibility, and R: Toggle Selectability. A status bar at the bottom indicates coordinates (X: 1347.90, Y: 729.60) and tool settings (F)Select: 0, DRD: OFF, CAE: OFF, dX: dY: Dist: Cmd: Rectangle.

On the right side of the interface, a taskbar shows the current user as "jbaker" and lists open windows including "Virtuoso@6.1.1 - Log: /home/faculty/jbaker/CDS.log", "LSW", and "Virtuoso@6.1.1 - Log: /home/faculty/jbaker/CDS.log". The system clock shows 08:50 on 2010-08-17.

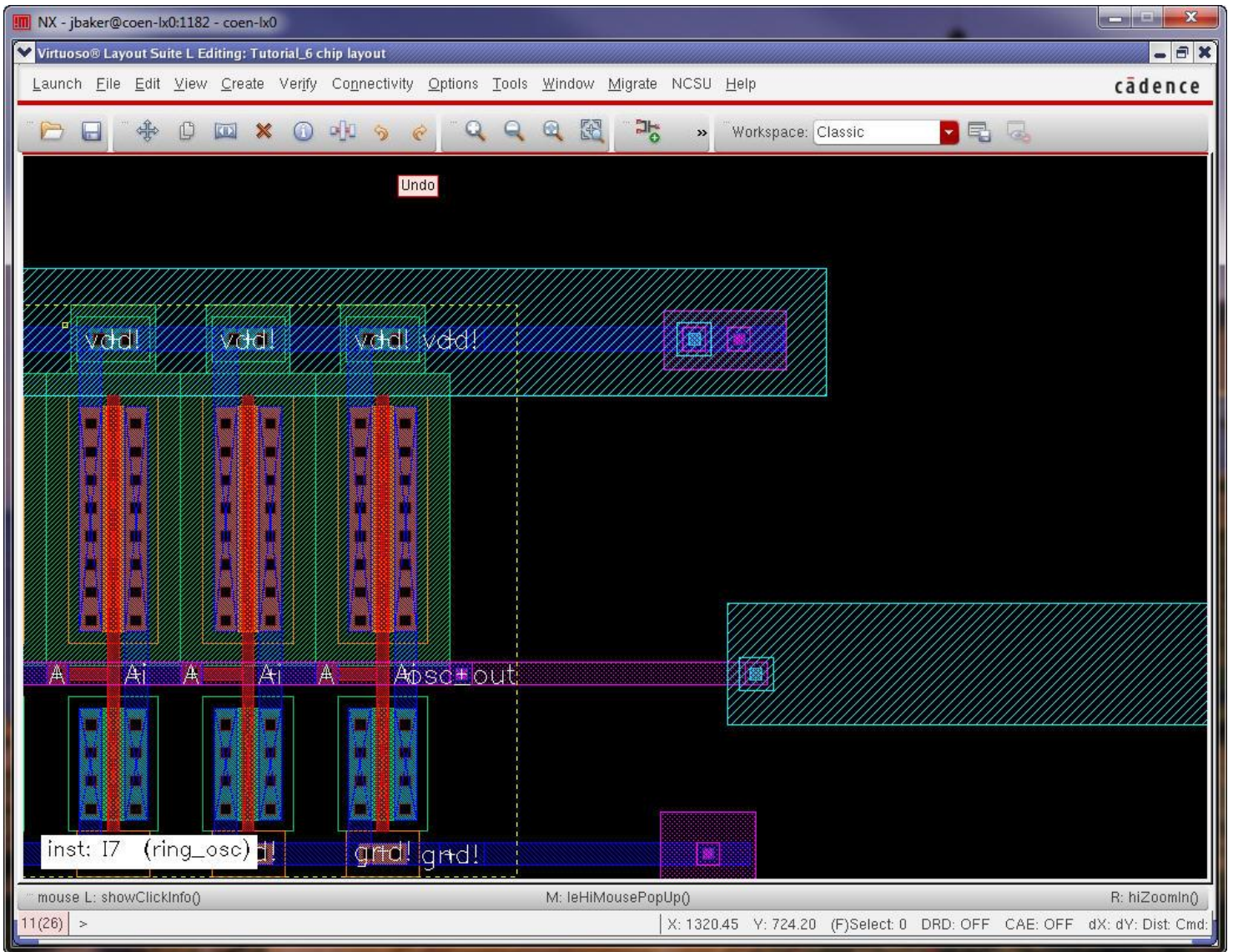


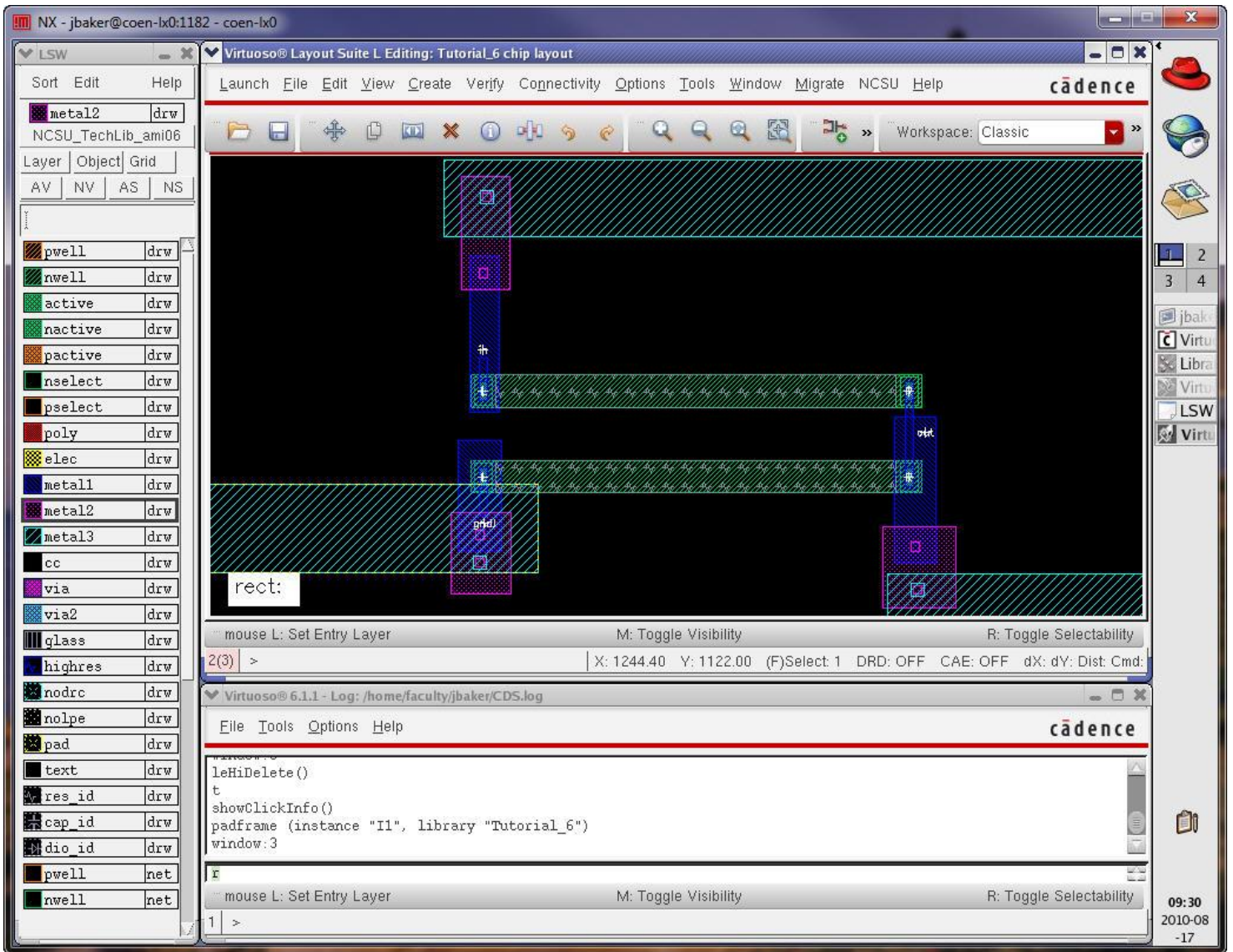
After adding all rectangles our chip layout may look like the following.



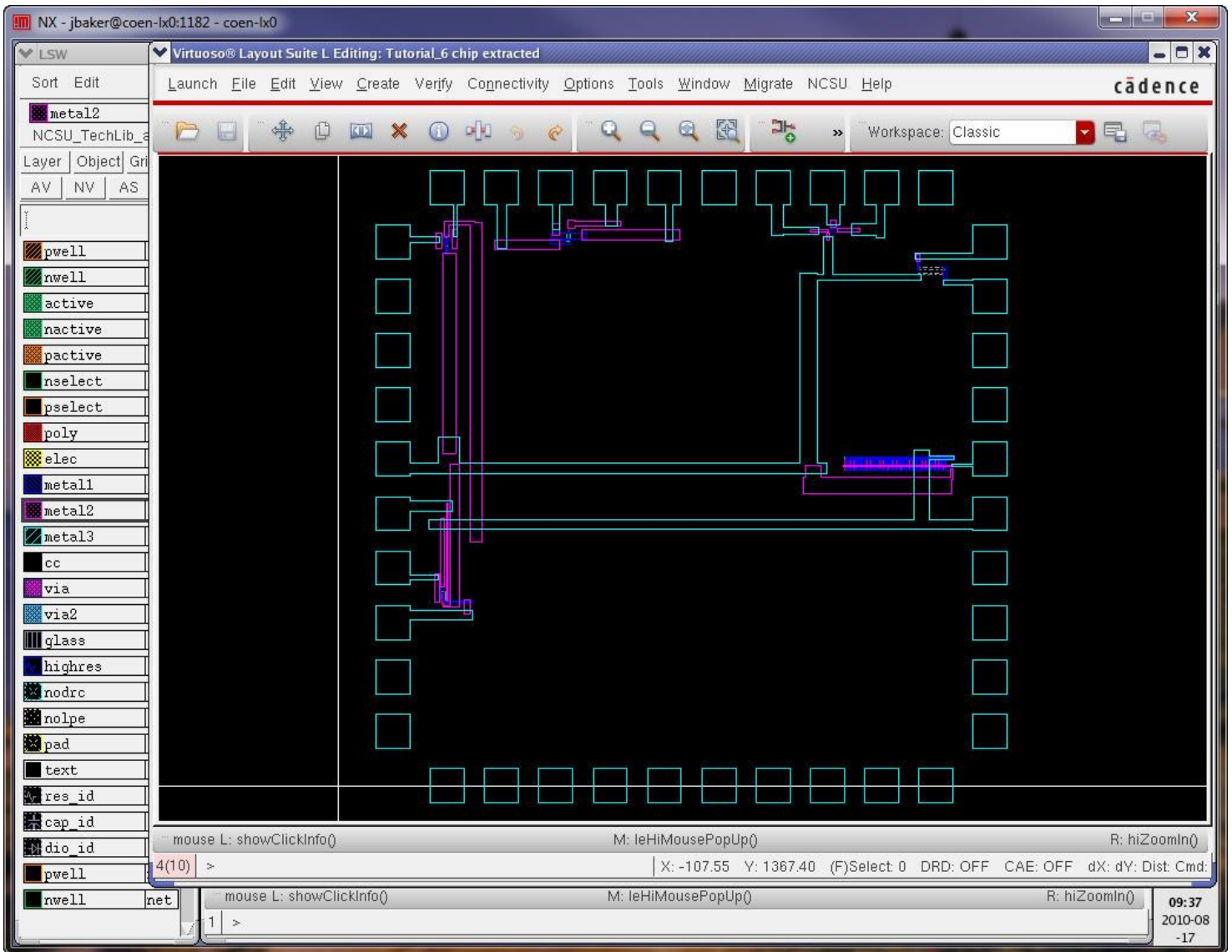
Next let's add vias.

Below are examples (again, these are not examples of good layouts but rather simply to illustrate the operation of the tools).

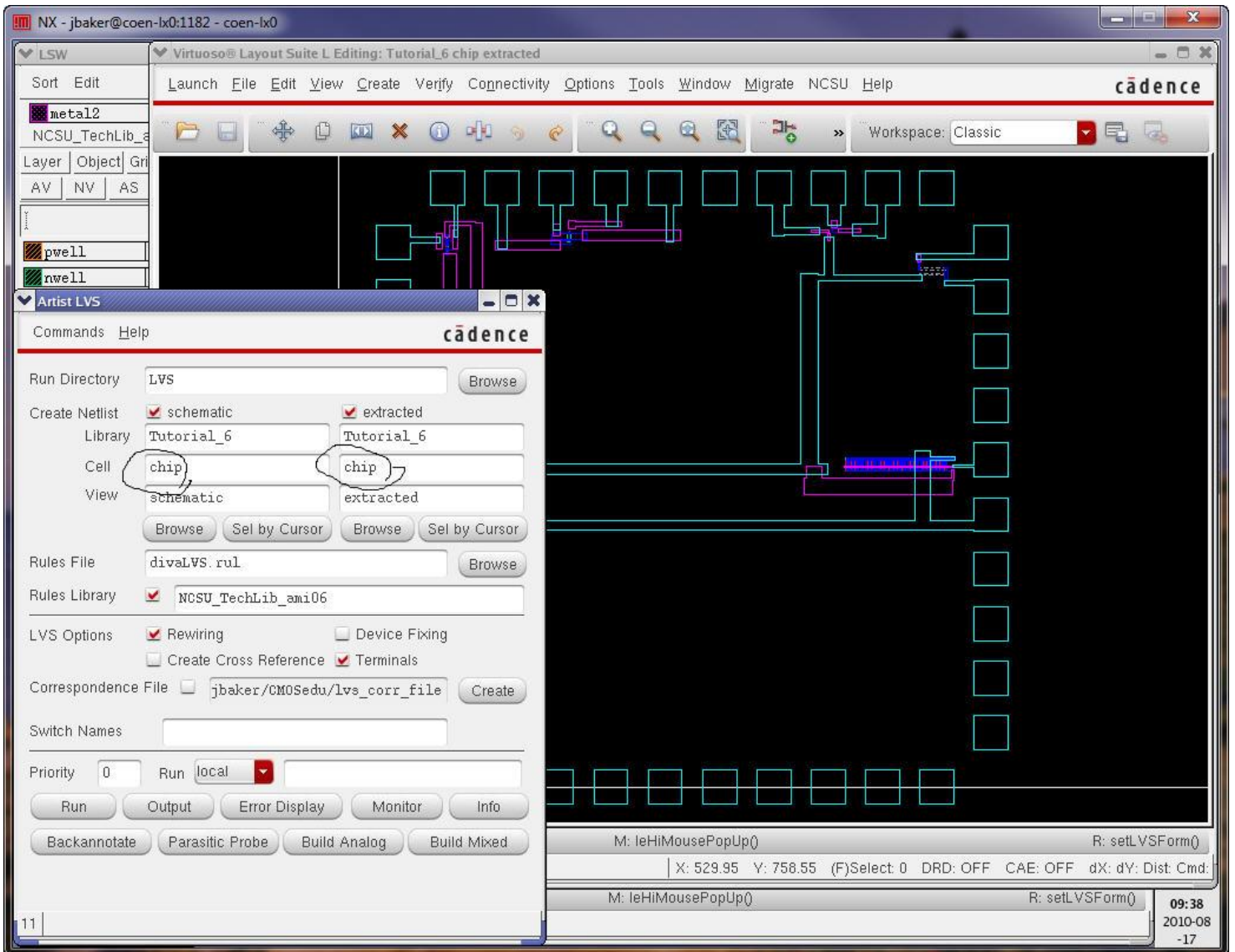




After adding the vias let's extract the layout.



And then perform an LVS.



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

```

2          res
35         pmos
35         rmos

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

4 net-list ambiguities were resolved by random selection.

The net-lists match.

```

	layout	schematic
instances		
un-matched	0	0
rewired	0	0
size errors	0	0
pruned	0	0
active	72	72
total	72	72
nets		
un-matched	0	0
merged	0	0
pruned	0	0
active	48	48
total	48	48
terminals		
un-matched	0	0
matched but different type	0	0

The status bar at the bottom right shows the date and time: 09:59 2010-08-17.

This concludes Tutorial 6.

For your reference the Tutorial_6 directory is available in [Tutorial_6.zip](#).

[Return](#)