EEC 116 Fall 2011 Lab #1
Cadence Schematic Capture and Layout Tutorial

Dept. of Electrical and Computer Engineering
University of California, Davis

September 26, 2011

Reading: Rabaey Chapters 1, 2, A, 5, Section 6.2.1 [1].
Reference: Brunvand Chapters 1-3 [2].

1 OBJECTIVE

The objective of this lab is to set up Cadence, your design library, and start to familiarize yourself with a full custom IC design flow.

2 TOOL SETUP

To get started, you first need to set up the Cadence tools. Cadence creates a bunch of random text files which set all kinds of configuration information. It is best to keep these in a single directory for the purposes of the labs. Log in to one of the lab workstations and at the prompt type:

```
eec116-setup
```

This should create a directory which you should `cd` to before starting up Cadence:

```
cd ~/eec116
```

Type `ls -a` and take a look at some of the configuration files, just to familiarize yourself with their contents. Now you are ready to start Cadence. Type

```
virtuoso &
```

at the prompt, and you should see a window labeled "Virtuoso 6.1.4" open up. This window is part of Cadence's Design Framework interface (GUI). A "What's New" window and the **Library Manager** window should open up, too. The first window is the **Command Interpreter Window** or **CIW**. Scroll through it to look at the messages displayed as the tool loads. Ignore any warnings for now. Get in the habit of scanning these messages whenever you launch the tool, keeping an eye out for any that are out of the ordinary. This will be very helpful if you encounter any tool-related problems.

You will store your design information in a *library*. The Library Manager is the interface which allows you to manipulate your design information as well as other components from other libraries, which may be shared with other users. Although the library information is stored as Unix files, don't try to move or rename them using shell commands. This can lead to inconsistent behavior and will likely break your libraries.

Familiarize yourself with the Library Manager and the corresponding `cds.lib` file in your `eec116` directory. You should be able to see some basic utility libraries, two libraries specific to UC Davis (`UCD_Analog_Parts` and `UCD_Digital_Parts`) and a technology library from the NCSU Cadence Design Kit (CDK) called `NCSU_TechLib_tsmc02d`. This technology library provides the physical design data and verification rules for the MOSIS Scalable CMOS 0.18μm process using DEEP design rules. This will be the example process we use in EEC 116, EEC 118, and EEC 119AB. The File menu allows you to create new libraries and cells within a library, while the Edit menu lets you copy, rename, delete, and update the access permissions.

Create a new library by invoking File→New→Library... in the Library Manager. Name the library `EEC116_<xx>` where `<xx>` is your username. Leave the path alone and the new library will be created in your current working directory (∼`/eec116`). Hit OK and a new window called "Technology File for New Library" should appear. Choose "Attach to an existing technology library" and accept the default, `NCSU_TechLib_tsmc02d`.

2.1 Project Directory

Although it won't be used for this lab, you should have a directory with a greatly expanded capacity for storing design information, simulation results, etc. `cd` to `/project/<username>` and make sure you can create, edit, and save files to that directory using `cat` or a text editor.

2.2 Remote Access

You should be able to access Cadence tools and the `/project` directory tree by using `ssh` to log in to `snake.ece.ucdavis.edu`. You are free to work on any aspects of labs or projects remotely, but if you need questions answered in real-time then you should attend the lab sections or the instructor or TA office hours.

3 SCHEMATIC CAPTURE

Part 1 Inverter Schematic First, we will create a new *cell* which represents a CMOS inverter. Cells have multiple views, including schematic, symbol, layout, etc. The circuit schematic view for a cell will be called `schematic`. In the second part of the lab, you will create a physical design (or layout) view of the cell called `layout`.

In the Library Manager, choose File→New→Cell View... In your `EEC116_<xx>`, enter a cell name of `inv_1x` and a view name of `schematic`. The Type field should also say schematic. The `_1x` designation indicates that this is a minimum size inverter (i.e., the PMOS and NMOS transistors are as small as we are allowing in our design). There are

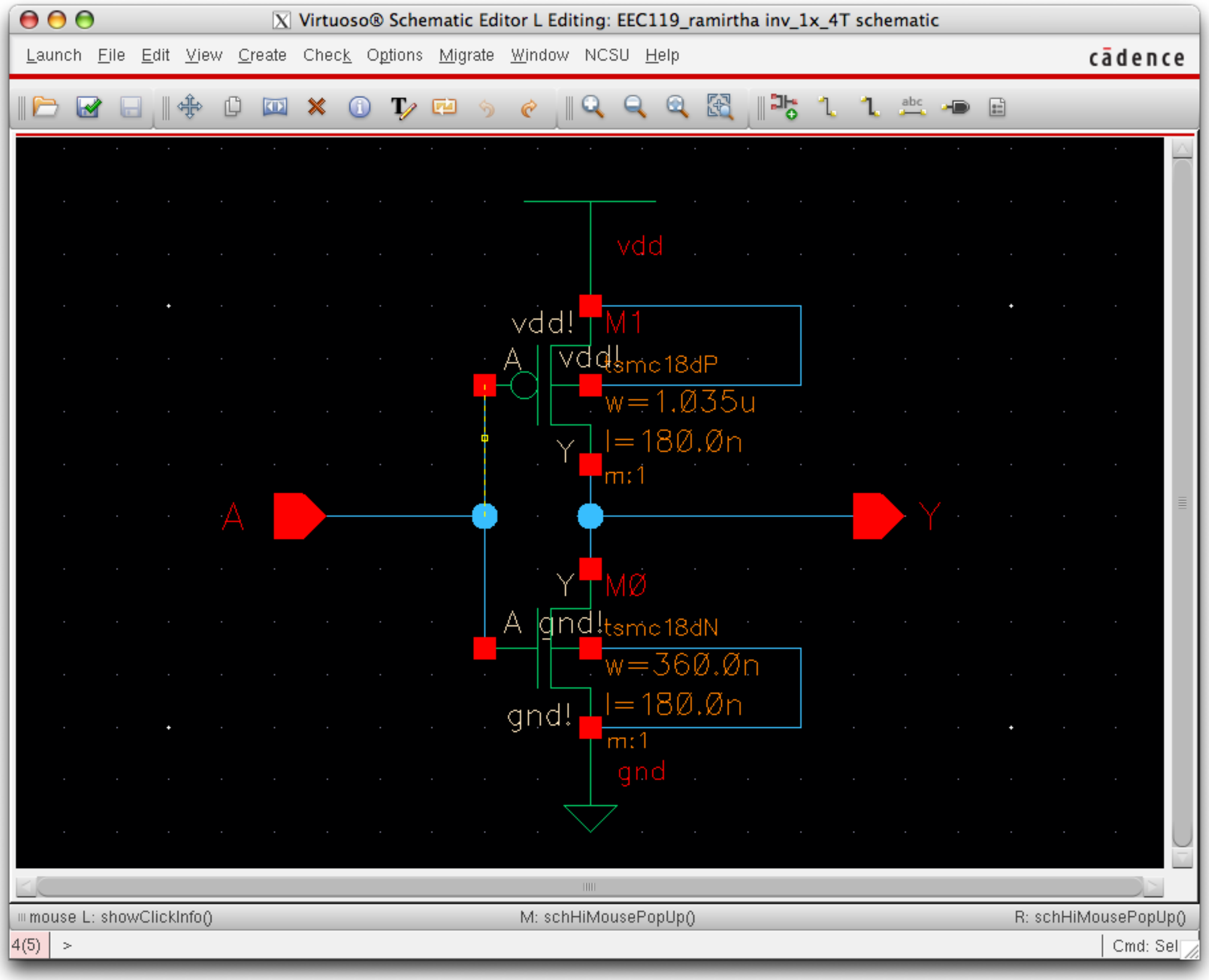

Figure 1: CMOS 1X inverter schematic.

times when you will want bigger inverters, for example, to drive long wires or other large capacitive loads.

Click OK. You may get a window asking you to confirm the association of the schematic view with a particular tool. An empty schematic window should open. Our goal is to create an inverter schematic identical to the one shown in Figure 1. We are working in a 0.18μm minimum feature size technology, meaning the smallest transistor gate length is 0.18μm. Scalable CMOS design rules require that all dimensions (for transistors, wires, resistors, capacitors, etc.) must be integer multiples of a fundamental unit called λ, typically equal to half the minimum feature size (0.09μm or 90nm in our case). However, the tool uses a grid of $\lambda/2$ (corresponding to 45nm). Take care when you are instantiating devices in a schematic or layout that you set dimensions as integer multiples of the λ unit to save yourself headaches later on.

Choose Create→Instance to open a **Component Browser** window. The menu lists keyboard shortcuts (for example, *i* for Instance) for various commands. You'll want to memorize the shortcuts you use most often. We will use the four terminal devices (`pmos4` and `nmos4`) from the `UCD_Analog_Parts` library. Choose `pmos4`. The Add Instance dialog window will appear. Look over the parameters to become familiar with them, and then set the Width property to 1.035u ("u" corresponds to microns). Click in the schematic window to place the device. Go back to the Component Browser and select and place the NMOS transistor (be sure to set its Width as well, "n" corresponds to nanometers). If you ever need to modify a devices properties, you can invoke the Edit→Properties→Objects... from the menu and then select the component to modify. Many times when you invoke a command from the menu or a hotkey, the command will remain in effect. Hit Esc to get out of that particular command mode. Many other useful commands are under the Edit menu tab (Undo, Redo, Move, Copy, Rotate, Delete). Familiarize yourself with these. Pay some attention to how the transistors are placed - neatness is very important. Move the elements around until it looks nice. You can look at the bottom line of the schematic editor window to keep track of which command mode you're in. I prefer placing series devices in a pullup or pulldown network in a line with a gap of at least one grid unit and two or more grid units between the NMOS and PMOS devices. Place the vdd and gnd (power and ground, respectively) components after the transistors are placed.

Next, choose Create→Pin to bring up the Add Pin dialog. Enter A, make sure the pin direction is set to "input", and place the pin. The tools are case sensitive so make sure the pin name is uppercase. Place an output pin named Y.

Now wire the elements together by invoking Create→Wire (narrow). Click on the various elements and terminals until all the connections are completed as shown in Figure 1.

Choose File→Check and Save to verify and save your schematic. It is a good to do this frequently to catch errors early and save your work in case of a crash. If you have any errors or warnings, fix them and run Check and Save again. A common mistake is wires that look like they might touch but don't actually connect. Delete the wire and redraw it.

Part 2 Inverter Symbol The next step is to create a symbol for the inverter. Rather than creating this from scratch, we are going to copy it from another library. Go to the Library Manager window and select the `symbol` view of the `inv` cell from the `UCD_Digital_Parts` library. Right-clicking on the view should bring up a dropdown menu. Select Copy... and fill in the dialog boxes with your library `EEC116_<xx>` in the To Library field and `inv_1x` in the To Cell field. Click OK. If any warnings appear, click OK again to dismiss the warning window. You can also invoke the Copy command by going to Edit→Copy... from the Library Manager menu bar after selecting the view.

Now, you can edit the new `symbol` view for your inverter cell by double-clicking on it in the Library Manager. Modify the symbol by selecting the "inv" label and editing its properties. Change the label to "inv_1x" to match the name of your cell. Then choose File→Check and Save to verify and save your symbol. Fix any warnings or errors and run Check and Save again if necessary.

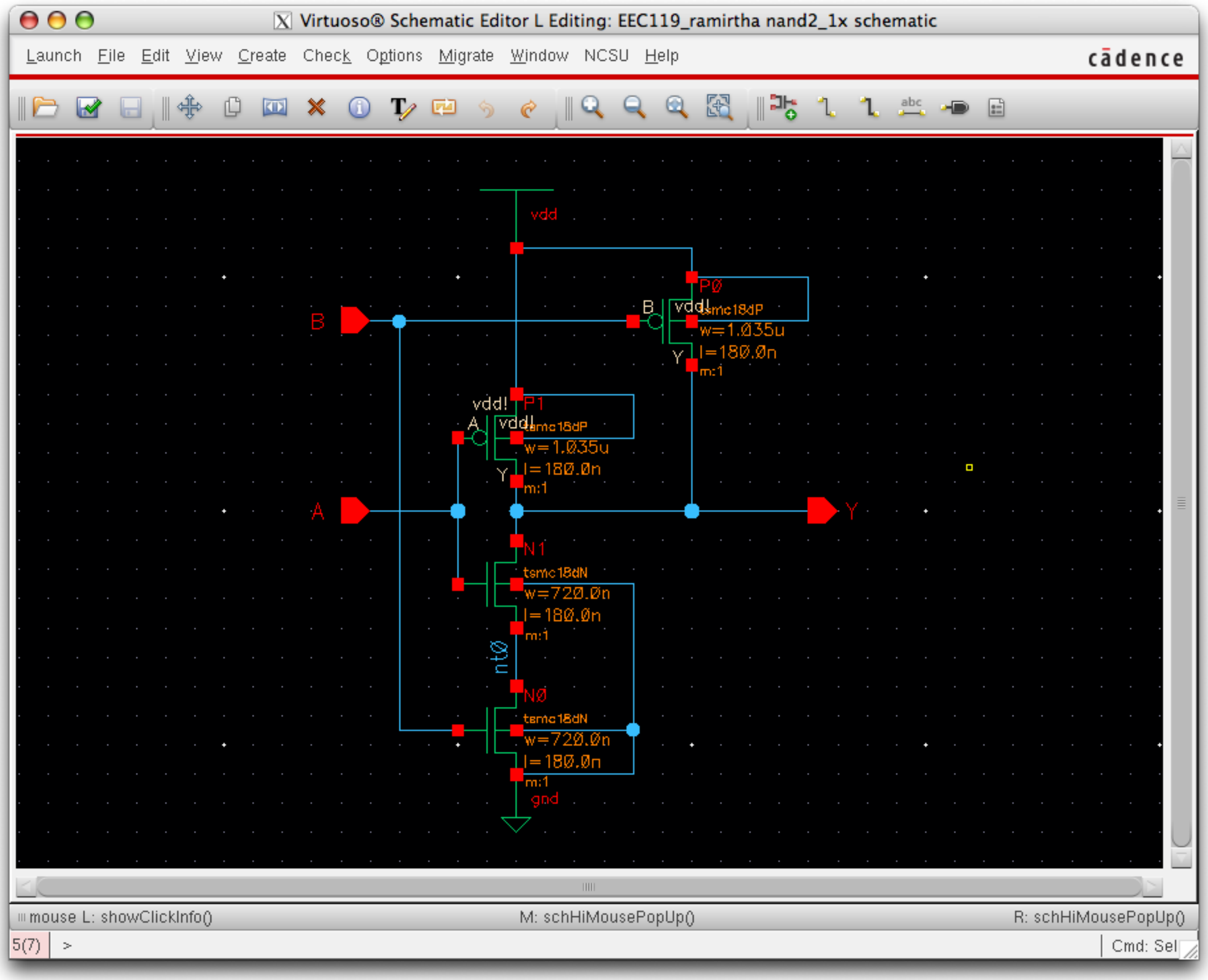

Figure 2: NAND2 gate schematic.

Part 3 NAND Gate Schematic Create a schematic for a two-input nand gate in a cell called `nand2_1x` in your 116 library. The final schematic should look similar to Figure 2. Note that the NMOS transistors have larger device sizes than for the inverter (i.e., the NAND gate is sized for approximately equal rise and fall times). Although pin order doesn't matter logically, it does matter physically and electrically, so you will get errors if you reverse the order. Place the input B pin as shown in the figure. In this class, we will use the convention that inputs which occur earlier in the alphabet appear closer to the output node in a series connection of devices, hence input A connects to the top NMOS.

It is a good idea to make sure every net (wire) in a design has a name. Otherwise, you'll have a tough time tracking down a problem later on one of the unnamed nets. Every net in your schematic is connected to a named pin or to power or ground except the net between the two series NMOS transistors. Choose Create→Wire name... Enter `nt0` and click on the wire to name it.

Check and Save your schematic, fixing any warnings and errors which may appear.

Part 4 NAND Gate Symbol Create a symbol view for your NAND gate like you did for the inverter in Part 2. Change the label to "nand2_1x" to match the name of your cell. Then choose File→Check and Save to verify and save your symbol. Fix any warnings or errors and run Check and Save again if necessary.

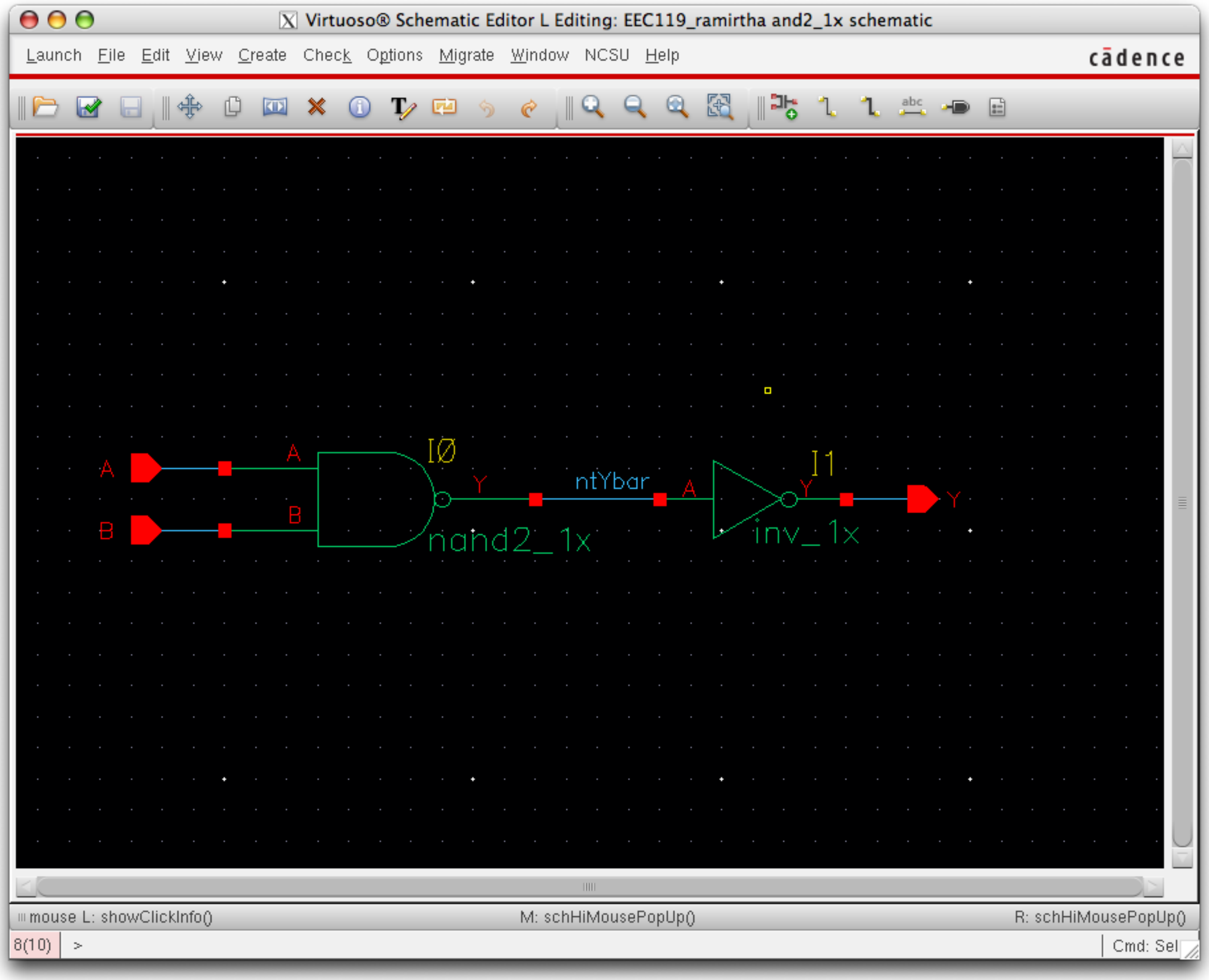
Virtuoso® Schematic Editor L Editing: EEC119_ramirtha and2_1x schematic
Launch File Edit View Create Check Options Migrate Window NCSU Help
cādence
A
B
A
B
IØ
Y
ntYbar
A
I1
Y
Y
nand2_1x
inv_1x
mouse L: showClickInfo()
M: schHiMousePopUp()
R: schHiMousePopUp()
8(10)
>
Cmd: Sel

Figure 3: AND2 gate schematic.

Part 5 Design Hierarchy: AND Gate Create a new schematic cell view in your library named `and2_1x`. A CMOS AND gate is a NAND gate followed by an inverter, and you will create a schematic for it by placing instances of the NAND gate and inverter you created above. This is an example of *hierarchical design* in which more complicated cells are composed of simpler cells. Add pins and wire labels. Your final schematic should look like Figure 3. Create a symbol view for your AND gate, modify any labels to be consistent, and Check and Save both the schematic and the symbol.

Part 6 NOR Gate Cell Design Create a new cell in your library called `nor2_1x`. Generate schematic and symbol views following the procedures you used for the previous sections of this lab. You must also choose the transistor sizes for the PMOS and NMOS devices. Make these consistent with the widths and lengths already used for the inverter and NAND gate.

4 LAYOUT AND VERIFICATION

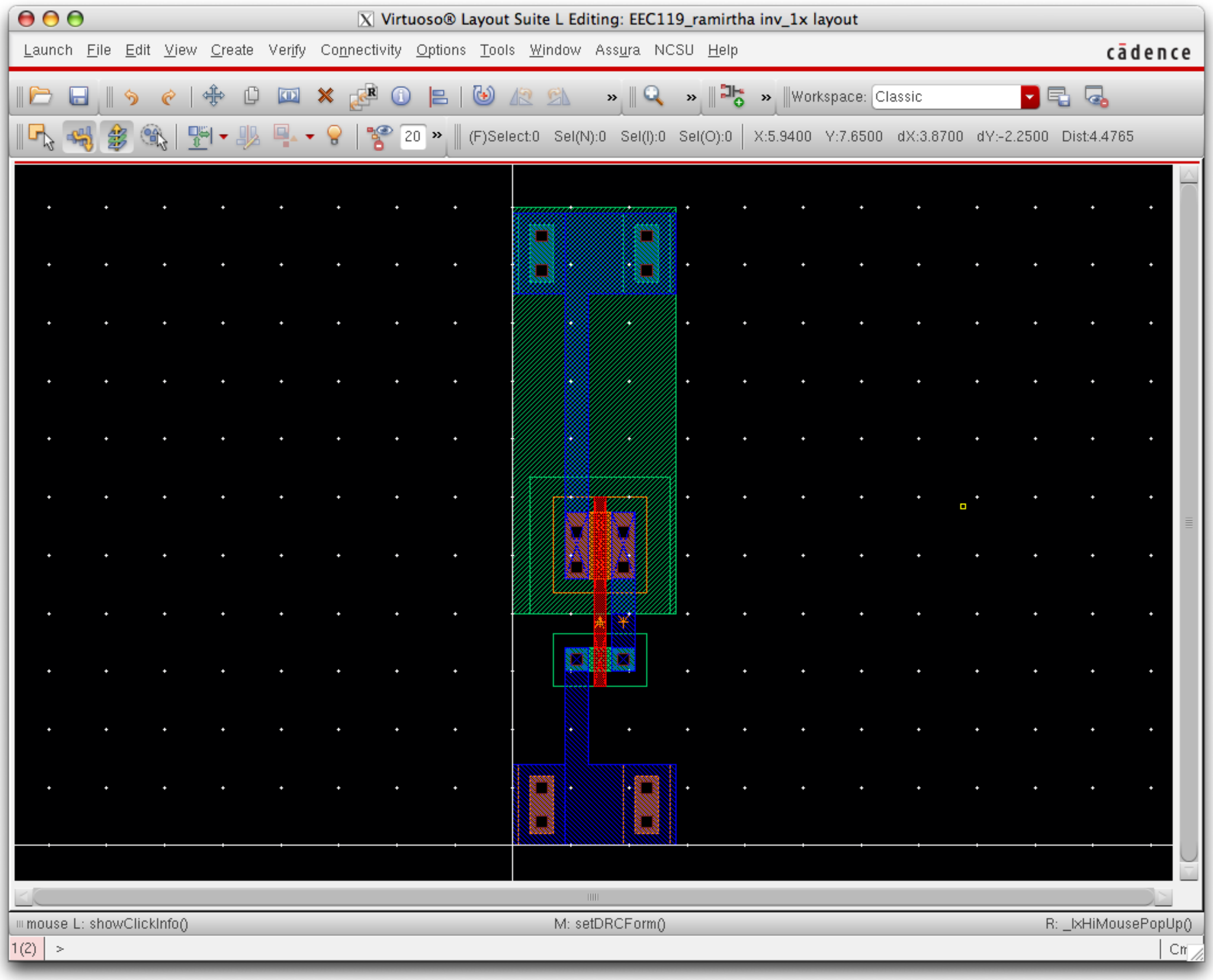
Virtuoso® Layout Suite L Editing: EEC119_ramirtha inv_1x layout
Launch File Edit View Create Verify Connectivity Options Tools Window Assura NCSU Help
cādence
Workspace: Classic
(F)Select:0 Sel(N):0 Sel(I):0 Sel(O):0 X:5.9400 Y:7.6500 dX:3.8700 dY:-2.2500 Dist:4.4765
mouse L: showClickInfo()
M: setDRCForm()
R: _lxHiMousePopUp()
1(2) >

Figure 4: CMOS 1X inverter layout.

Part 7 Inverter Layout Your first task is to create a physical design or layout for the minimum-sized inverter cell you created in the first part of the lab. Go to the Library Manager window, click on the `inv_1x` cell in your EEC116 library, and select File→New→Cell

View... Fill in `layout` in the View field and hit OK. A new empty layout editor window should appear along with the **Layer Selection Window** or **LSW**, which shows the various mask layers and gives you options for selecting them, making them visible or invisible, etc. You may also want to open the schematic for the cell to use as a reference for naming pins and creating the correct transistor dimensions.

In this part of the lab, you will create a layout identical to the one shown in Figure 4. This layout style follows a convention for *standard cells*, which require that all cells have consistent dimensions so they can easily snap together (in fact, a piece of software should be able to assemble them). Neatness and precision are necessary to achieve a good layout. Therefore, pay careful attention to all the dimensions of the design.

The drawing grid is there to help you meet all the dimension specifications and *design rules*. The major dots should be spaced at 10λ while the minor dots are spaced at 2λ ($\lambda = 0.09\mu$m for our process). The cursor snaps to a $\lambda/2$ grid, which may or may not irritate you depending on your design style. To change any of these settings, choose Options→Display... or Options→Editor... For example, you may find that you have difficulty selecting certain polygons because the cursor keeps snapping to an undesirable location. You can go to Options→Editor... and toggle Gravity On to turn this snapping behavior off. You can always press the Help button in the lower right hand corner of any menu to bring up the Cadence Documentation browser if you want to learn what various menu options do.

Start by drawing the metal lines for power and ground at the top and bottom of the cell. Select `metal1` in the LSW and invoke Create→Shape→Rectangle (hotkey "r"). Draw a strip 14λ wide with the lower left corner at the origin for the ground line. Then copy it using menu commands or the hotkey to a position above the ground line such that the center-to-center spacing between these power rails is 95λ. This spacing is much wider than necessary for an inverter, but may be required for more complicated cells. Next, draw a rectangle of `nwell` whose lower left corner is at 40λ above the origin and whose upper left corner is at 110λ. You can see the X,Y coordinates of the cursor in the toolbar of the layout editor menu. Also, Tools→Create Ruler (hotkey "k") is a very useful way to mark dimensions.

Now you are ready to create transistors. The design rules we are following are the Deep Submicron Scalable CMOS (SCMOS) from the MOSIS foundry service. You can view them at: `http://www.mosis.com/Technical/Designrules/scmos/scmos-main.html`. It is a good idea to get familiar with these rules as you complete your layout, since obeying the rules ahead of time will save you effort later. Cadence, unlike Magic, does not offer an interactive design rule check, but there are ways (see below) to emulate it. You can create the transistors by drawing rectangles directly, using the appropriate layers for gates (`poly`), diffusion (`nactive`/`pactive` for NMOS and PMOS respectively), selects (`nselect` and `pselect`) which must surround the diffusion/active regions, `cc` for contacts, and `metal1`. Remember that the `poly` rectangle over the diffusion rectangle must have one dimension equal to the transistor gate length (2λ) and the other dimension equal to the transistor width. Contacts are $2\lambda \times 2\lambda$ with a surrounding layer of 1λ. Alternatively, you can instantiate a layout *macro* representing an entire device by choosing Create→Instance... and filling in or browsing to the `NCSU_TechLib_tsmc02d` library and choosing the `nmos` or `pmos` cell. Place the instance in the layout where you want it. These macros are parametrized cells (or *pcells*) which means you can edit their properties to change the widths and lengths of the transistors. Sometimes you want to start with a pcell and then *flatten* it into its component rectangles so you can edit the layers individually. You can do this my selecting the instance and invoking Edit→Hierarchy→Flatten... Be sure to select "one level" for the Flatten Mode and toggle Flatten Pcells on. Play around with these various ways of creating transistors and other geometry as you complete the layout. Be sure to keep the spacing of the active regions to the `nwell` boundary at a minimum - this will define a set of horizontal lines (the bottom edges of the well and p-type diffusion and the top edge of the n-type diffusion) that make it easier to align cells by hand.

Lastly, you need to create pins and labels to help the tool verify your layout against the schematic. Invoke Create→Pin... and add input pin A on the `poly` layer and output pin Y on `metal1` by drawing rectangles at the appropriate places, overlapping other rectangles you may have drawn to make the physical connections. Also, add vdd! and gnd! pins on `metal1` (the exclamation points are important - they tell various tools that these are special global nets). Define these pins as inputOutput. Use Create→Label... to add the input and output labels as shown in Figure 4.

Part 8 Inverter Layout Verification Now you are ready to see if your layout meets the design rules and matches the circuit schematic. There are three steps in this verification process: Design Rule Checking (DRC), extraction, and Layout Versus Schematic (LVS) checking.

1. DRC: Invoke Verify→DRC... The DRC dialog window should open. Click OK and check the CIW for any design rule violations which are reported. Fix any errors and rerun DRC until none are reported. You can use Verify→Markers→Explain to figure out what violation a particular flashing marker represents. Note that by changing the Checking Limit to incremental or by area, you can check a subset of the layout. In this way you can sort of emulate interactive DRC (although it's not automatic every time you add or modify some layout geometry).

2. Extract: Extraction creates a new cell view (`extracted`) which results from mapping the layout geometry to circuit components. Sometimes this view is used solely for verification against a schematic. Other times extra components representing parasitics, such as the resistances and capacitances of wires, are also extracted from the layout for use in simulation. In this lab, we will only focus on verification. Choose Verify→Extract... In the Extractor dialog window, select Set Switches and Keep_labels_in_extracted_view. Click OK (twice) and watch and correct any errors. Go to the Library Manager and open the `extracted` view for the cell. Confirm that the labels from the layout are still visible.

3. LVS: Choose Verify→LVS... and the Artist LVS dialog window should open. LVS creates a run directory which it fills with all kinds of temporary files, some of which are quite useful for tracking down bugs. Modify the Run Directory field to point to `/project/<xx>/LVS` where `<xx>` is replaced with your username. This should write the files to the expanded storage you've been assigned for 116. Make sure the appropriate cell views are filled in the schematic and extracted entries under the Create Netlist buttons. Click Run and wait for the results. Correct any errors which appear. You can view various useful information by clicking the Output, Error Display, and Info

buttons. Make sure that the tool is also comparing transistor lengths and widths by choosing NCSU→Modify LVS Rules... from the layout editor toolbar and confirming that Compare FET Parameters is checked.

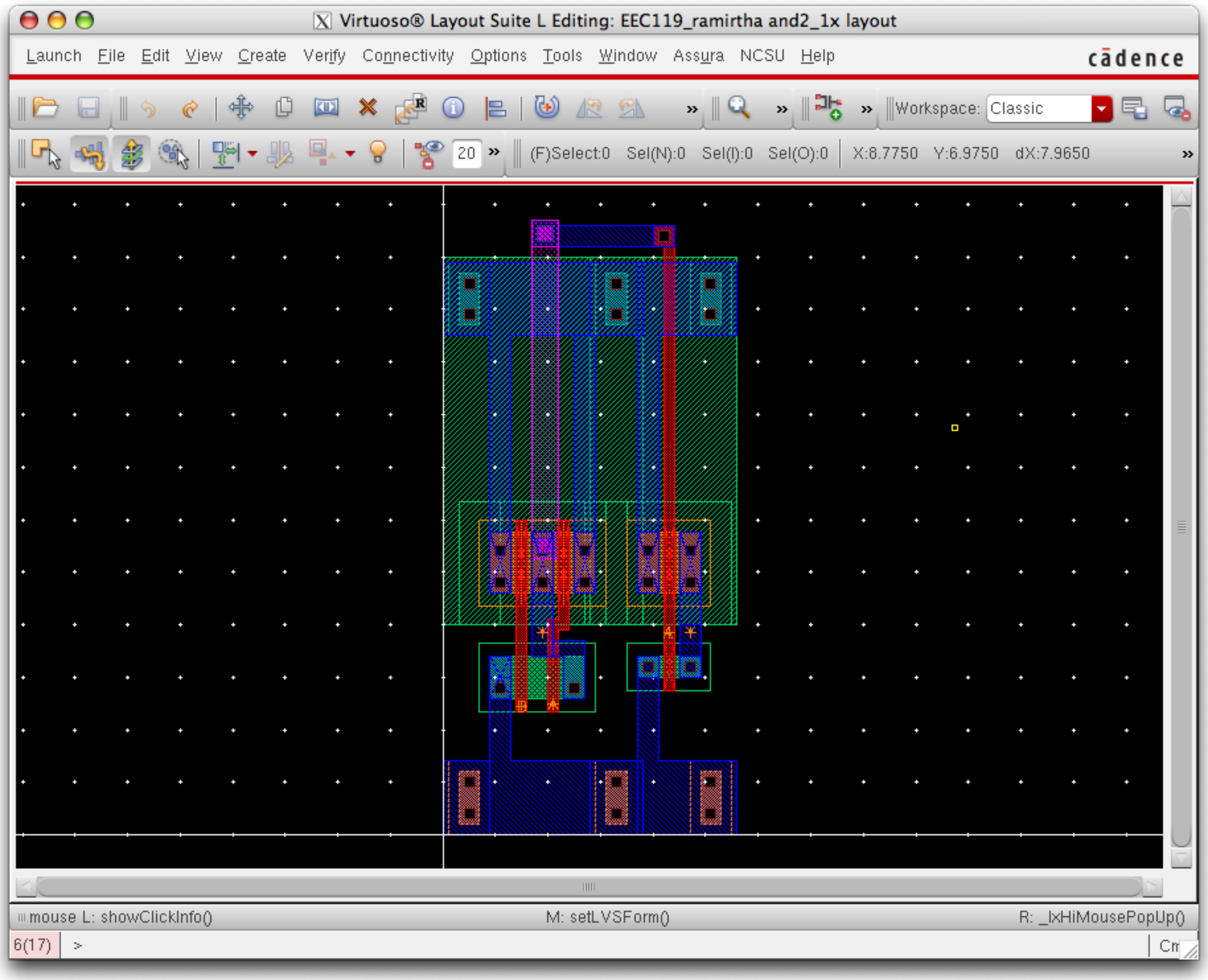
Virtuoso® Layout Suite L Editing: EEC119_ramirtha and2_1x layout
Launch File Edit View Create Verify Connectivity Options Tools Window Assura NCSU Help
cādence
Workspace: Classic
20
(F)Select:0 Sel(N):0 Sel(I):0 Sel(O):0 X:8.7750 Y:6.9750 dX:7.9650
mouse L: showClickInfo()
M: setLVSForm()
R: _lxHiMousePopUp()
6(17)
>

Figure 5: AND gate layout.

Part 9 Hierarchical Layout and Verification: AND Gate Create and verify a layout for the `and2_1x` cell you started in the first part of the lab. Begin by creating and verifying a layout for the `nand2_1x` cell and then wiring the NAND gate and the inverter as shown in Figure 5. The output of the NAND gate is routed to the input of the inverter in an external *routing track* for `metal1`. The vertical connection is made in `metal2`, therefore you need to create a *via* to connect the two metal layers. You can do this by hand or by instantiating a layout from the technology library. Note that there is some wasted space in the example which you can eliminate by taking advantage of certain symmetries in the the cells. See if you can improve the cell so it consumes less area. Add the input, output, and power supply pins and make sure your design is DRC and LVS clean (no errors).

Part 10 NOR and OR Gate Cell Physical Design Create a new cell in your library called `or2_1x`. Generate schematic, symbol, and layout views following the procedures you used for the previous sections of this lab. You must also choose the transistor sizes for the PMOS and NMOS devices. Make these consistent with the widths and lengths already used for the inverter and NAND gate. Verify your design and make sure there are no DRC errors and that the layout and schematic match. Note that you will need to add a layout view to the NOR2 cell you created above to make the OR gate.

Checkoff

Show your final completed cell views for all three cells to the TA or instructor for checkoff.

Acknowledgments

Parts of this lab were inspired by lab exercises developed by Prof. David Money Harris and others at Harvey Mudd College for the class E158: Introduction to CMOS VLSI Design.

References

[1] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc., 2003.

[2] E. Brunvand, *Digital VLSI Chip Design with Cadence and Synopsys CAD Tools*, 1st ed. San Francisco: Addison-Wesley, Inc., 2010.

EEC 116 Fall 2011 Lab #1 Summary

Name:

Grading:

Part	Checkoff	TA Initials	Date
1 CMOS Inverter Schematic			
2 CMOS Inverter Symbol			
3 NAND Gate Schematic			
4 NAND Gate Symbol			
5 AND Gate Schematic			
5 AND Gate Symbol			
6 NOR Gate Schematic			
6 NOR Gate Symbol			
10 OR Gate Schematic			
10 OR Gate Symbol			
7 CMOS Inverter Layout			
8 CMOS Inverter DRC			
8 CMOS Inverter LVS			
9 NAND Gate Layout			
9 NAND Gate DRC			
9 NAND Gate LVS			
9 AND Gate Layout			
9 AND Gate DRC			
9 AND Gate LVS			
10 NOR Gate Layout			
10 NOR Gate DRC			
10 NOR Gate LVS			
10 OR Gate Layout			
10 OR Gate DRC			
10 OR Gate LVS			