EE 330 Laboratory 3
Layout, DRC, and LVS

Spring 2018

Contents

Objective:

This experiment focuses on the concept of layout of integrated circuits. Manual and partially automated methods of layout will be discussed. Design rules and the use of CAD tools for verifying that all design rules are satisfied, termed design rule checkers (DRC), will be introduced. Finally, CAD tools for verifying that the actual equivalent circuit corresponding to a layout agrees with the original circuit schematic, known as layout versus schematic (LVS) tools, will be introduced.

In order to allow tens or hundreds of engineers to work on the same design a large number of design rules have been made to make sure all layouts created will work properly. The **full design rules** we use can be obtained from the MOSIS WEB site https://www.mosis.com/files/scmos/scmos.pdf but a subset of the design rules that **will be adequate** for this experiment can be found on the class WEB site at http://class.ece.iastate.edu/ee330/miscHandouts/MOSIS%20Rules%20Pictorial%202009.pdf/. These rules can be referenced when an error occurs during a Design Rule Check (DRC) check if needed.

Note: Lambda for the NCSU_TechLib_tmsc02 technology is is 0.1µ.

Part 1 creating a layout

Open the layout view of the inverter, created in Part 4 of Lab 2. For the first layout we **will not concern ourselves with size**, but in later layouts making the transistors the proper size will be important.

Last week we created a PMOS transistor. This week we will use that transistor and add an NMOS transistor to the design to create our inverter. Before continuing however, we want to run DRC. Normally we want to run DRC early and often in the process so that we do not have to make a lot of fixes at the same time.

1.1 Run DRC

To run a DRC, go to **Verify → DRC → {OK** or **Apply}.** If there is a design rule violation, the DRC tool will identify what it is and where it is at. For example, if the smallest width of a poly strip is 0.2µm, and one was drawn at 0.1µm, then the Design Rule Check will print out an error and create a yellow symbol in the layout. This symbol will not go away until you fix the error **and run another DRC**. If the DRC does not find any errors, the circuit should not fail when fabricated because of spacing-related concerns.

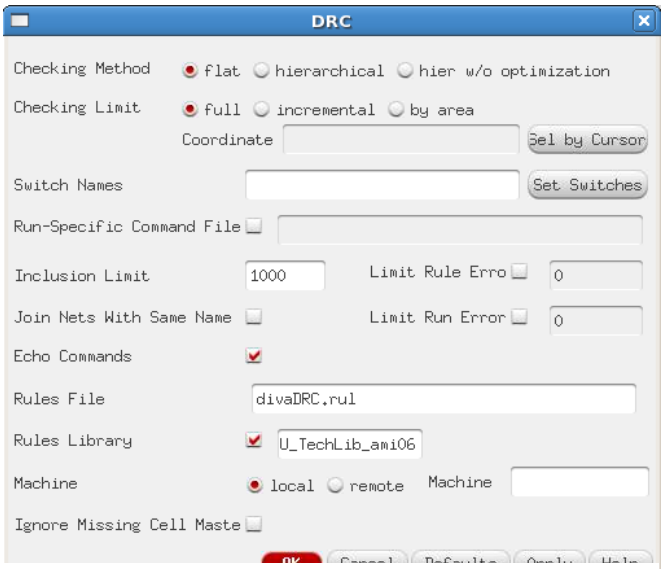


Figure 1

The DRC check does not guarantee that the circuit corresponding to the layout is the same as the desired circuit nor does it guarantee that the actual circuit schematic meets performance requirements. The file that contains the design rules in the ISU installation of the Cadence toolset is divaDRC.rul

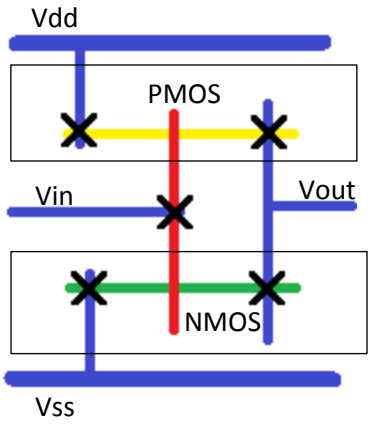
If your layout has no errors, you should see the following message in the CIW:

```
Total errors found: 0
```

If you'd like to debug your errors, you can go to **Verify** → **Markers** → **Explain**, and click on any of the white error markers on your layout. Over time you will learn what white markers mean: for example, if you see that a rectangle has a white cross inside it; it means it is not properly shaped. If you see white markings in the area between two rectangles, it means that they are too close.

1.2 Stick Diagram to Physical Layer

To determine how to create physical layouts of systems one representation is a stick diagram. A diagram of an inverter is on the right.



- Blue Lines – Metal 1
- Yellow Lines – P-diffusion, which is an N-well, P-select, and P-active
- Green Lines – N-diffusion, made of N-select and N-active
- Red Lines – Polysilicon
- Black X – Connection (Via)

Finish the inverter by adding an NMOS transistor and connections to make your design similar to the stick diagram.

1.3 Bulk Connections

The **most common thing to forget** when designing the layout is to create bulk connections. Each P active region requires a via called a "N Tap", so designs with many different P active regions require multiple N taps. The Vss line needs an "M1_P" as its bulk connection. Unlike the N tap, most designs we will be making will only need one of these bulk connections for the entire design.

These bulk connections need to be connected to the P active and N active regions. Use a Metal 1 rectangle to connect them and M1_P and M1_N vias to connect that metal to the active regions.

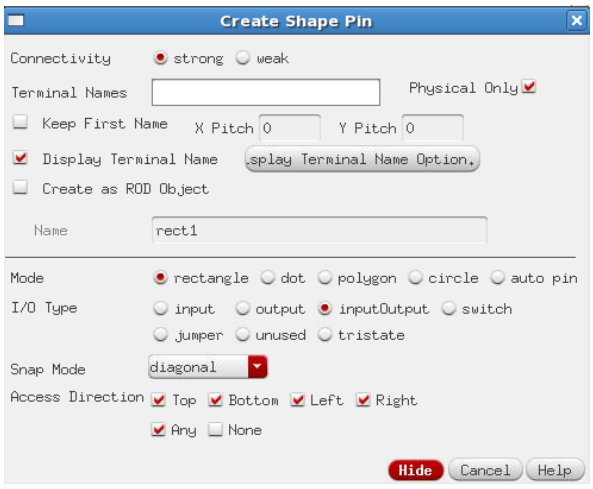


Figure 2

1.4 Pins

The final step will be to make inputs and outputs for your circuit. To create a pin, go to **Create** → **Pin**. Make sure the pins are consistent with the schematic: names *and* I/O types. **Pin names** are **case sensitive** and **cannot be changed** without deleting and remaking them. Any other variables of a pin, such as its layer or I/O type can be changed in its properties later. Check the **Display Terminal Name (now called Create Label)** and **Physical Only** boxes. When you know where the pin goes, make sure to have the same layer selected in the Layers toolbox.

Create the pins for your Input, Output, and Vss and Vdd connections. Vss and Vdd will be I/O type inputOutput. These will often be made of Metal 1, but the Input can be made from Poly to connect it directly to the gate rather than adding a "M1_Poly" via.

Notable DRC Error with Pins

Label/Pin “{Pin Name}” is causing two nets to have the same name. This error is fairly common when running DRC once Pins have been placed, but is not really an error most of the time. The problem is that the computer does not recognize that the active region of the transistors is being split into two different nets with the use of the gate. This error can usually be ignored, but may need to be looked into if there is a problem with the Layout Vs Schematic (LVS) later.

Part 2 Extracted View

Once you believe the design is complete it is time to extract the layout. This means that the computer will generate a schematic from the layout. To create this schematic, go to **Verify → Extract → OK**. The extraction rules are in the *divaEXT.rul* file.

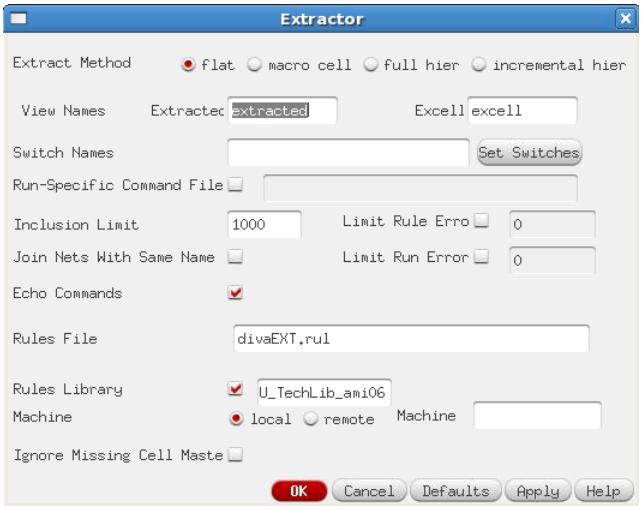


Figure 3

With the layout now extracted open the extracted view. The different outlined rectangles show where nets are, while a solid color shows where a pin is. The color tells you what they are made of, the blue is Metal 1, the red is Poly, etc. Later we will use this to see what the computer interprets as resistors, capacitors, diodes, etc. Move the pmos4 box and the nmos4 box so they are not over the design and press **Shift+F**.

Shift+F is used to show more detail in these boxes, they should now look like Transistors with a width and length, as in Figure 4. By clicking on any box it will change the outline to be white and show what parts are connected. It also shows what point on each component that connection correlates to. As can be seen on in the figure the metal on the right is connected to the Drain of both the NMOS and PMOS, as we want for an inverter.

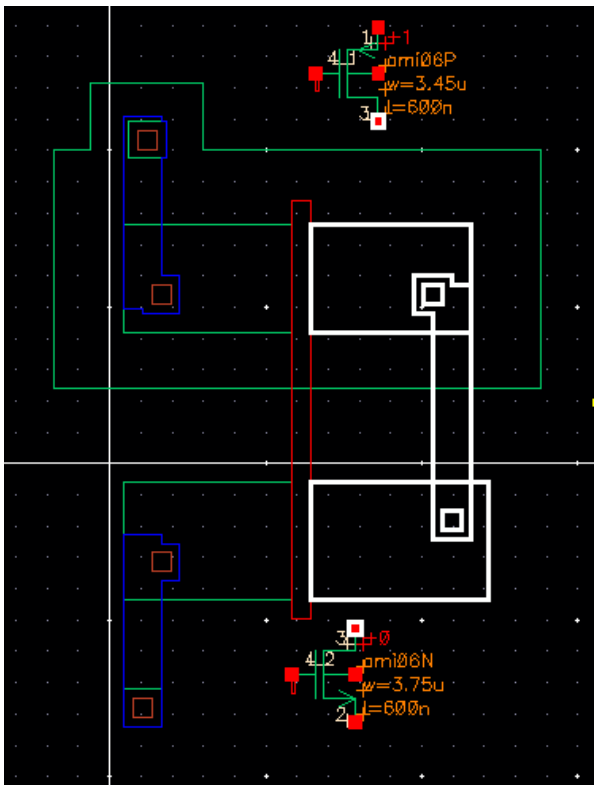


Figure 4: Extracted View with transistor information

2.1 Layout vs. Schematic (LVS) comparison:

When you are satisfied the design is correct it is time to test it. Run the LVS tool on your layout to verify that the circuit you have laid out agrees with your original schematic. Diva is the name of the LVS tool we will be working with in the Cadence toolset. To run LVS in the Cadence environment, go to **Verify → LVS**. Make sure you are comparing the right schematic and extracted views by hitting **Browse**. Hit **Run** when you are ready to run the LVS.

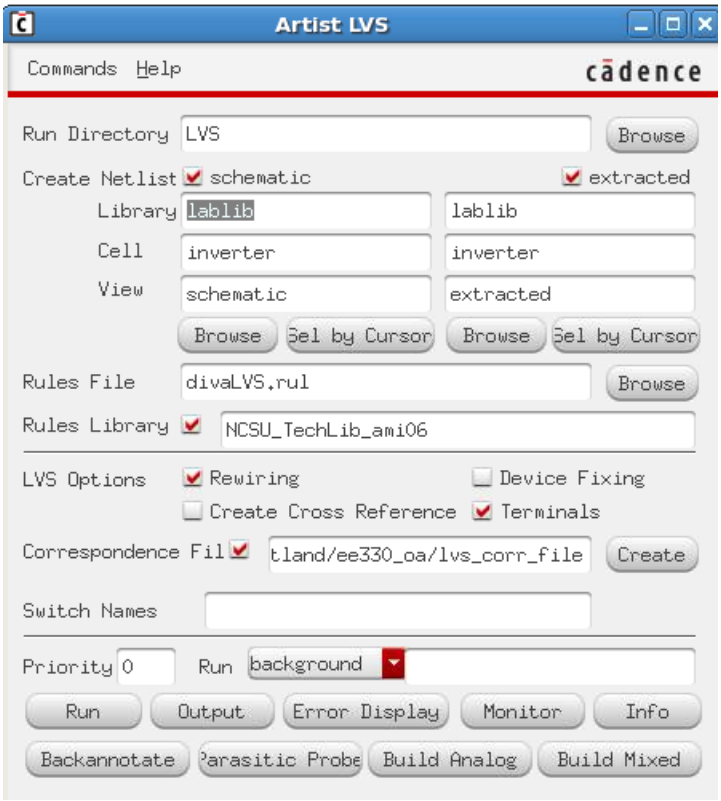


Figure 5

If you have everything checked and saved before you run the LVS, you will get the message:

```
Job 'directory' that was
started at 'time' has succeeded.
```

This only means that the LVS ran properly, not that your design passed the test. If you don't save your schematic before running the LVS you will get a fail message, so make sure you always save your work. In order to know whether your design passes the LVS, go back to the LVS window and hit **Output**. Go over the file "si.out" that opens and study its different sections. This is the only text you will have for debugging. If your layout is acceptable and if you scroll down a bit you should see:

```
The net-lists match.
```

If they don't match, you need to alter your layout and re-do all the extraction and LVS steps again until they match.

How to locate a net on the extracted view:

Sometimes it is very hard to find a net in your extracted view. Probing the design makes finding nets a much easier job. Probing can be initiated by the command **Verify/ Probe**. In the Probing form, click on **Add Net**. Then go to the CIW, and type "X" (X is the name of the net that you wish to locate) at the command prompt, including the double quotes, and press Enter. Back to the extracted cell-view window, you should see a mask layer being highlighted, which has the given net name. You can also left-click on a net and look at the CIW to know its name. The latter method is faster and more suitable for smaller designs, while the first one is better for larger designs.

Part 3 Using P cells

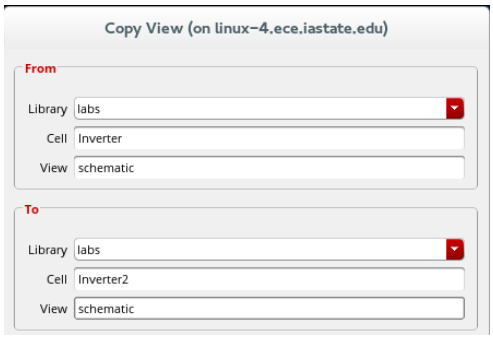


Figure 6

Now that you are used to using the Layout tools, we are going to remake the inverter more efficiently. Instead of creating the PMOS and NMOS cells by hand, we are going to use Instances of standard cells that automatically create minimum sized P active and N active regions for use.

Start by going to the library manager and select the Inverter schematic. Right click on it and go to Copy, under "To" name the Cell "Inverter2" as in Figure 6. Create a layout for this new Inverter2 schematic **(File → New → Cell View → Layout)**.

In the layout create a new **Instance (shortcut "I")** we want to automatically generate a PMOS layout, so select the NCSU_TechLib_tsmc02 library and the pmos Cell, the view should default to layout (Figure 7). This will appear as a red box that says PMOS, place this in the layout and press **Shift+F.**

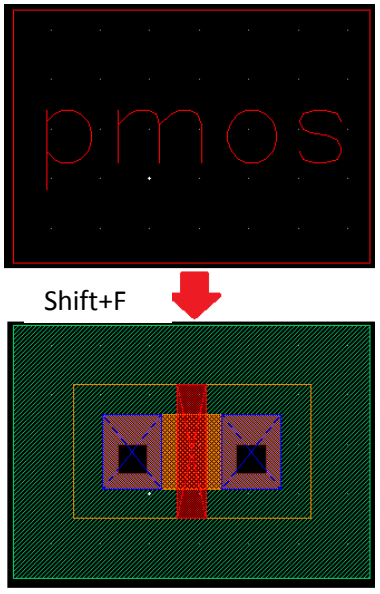


Figure8: Use Control + F to reverse this.

You will now see something similar to the original PMOS you made, but likely much smaller. The instance is automatically minimally sized, but we can change many things by clicking on it and going to Properties **(shortcut "Q")**.

There are many important things that can be changed here, including the Width, Length, **Fingers**, and **Multiplicity**. For an Inverter we want these at their default values, but change the Multiplier to 3 to see the design change to include 3 gates and 4 areas designed to make connections with Metal 1. **This will be used to create NAND and NOR gates later**, for now return to Multiplicity 1 to continue making the inverter.

Next create another instance this time of NMOS and connect the two gates with a Path **(Shortcut "P")** of Poly, connect one side of each transistor with Metal 1, and add extensions out to be used to connect to Pins and Bulk Connections. There are many ways to design this to take up as little area as possible, but remember to always make it easy to connect to the inputs and outputs with different parts.

We advise you always put **Vdd on the top**, **Vss on the Bottom**, all of the **Inputs on the Left**, and all the **Outputs on the Right**. When finished **extract and run LVS**.

Part 4 NAND xor NOR

In the next experiment, a logic circuit will be designed jointly by two students. For this design, every student will choose a Boolean function for use in the next experiment from the prelab for next week. The Boolean function will be realized with NAND and NOR logic gates. One student will be responsible for creating a three input NAND gate and the other for creating a three input NOR gate.

Find a partner and decide who will be responsible for each gate, then **create the schematic and test bench** for the gate you are responsible for. **Run the test bench** and verify your gate works as expected.

Looking Forward

Next week we will be creating a layout for your gate, exchanging gates, and creating the schematic, test bench, and layout for your Boolean function. It has been a long lab, so if you have time you may want to try to finish the layout for the NAND or NOR gate this week. Also, there is a **Pre-Lab** for next week, including creating a stick diagram for your gate, which will make creating the layout easier if you do it first. To create the gate layout you will want to use pcells with **fingers** and **multiplicity,** discussed above.

Useful keyboard shortcuts in schematic view:

Action	Key
Add Instance	i
Add Pin	P
Wire	w
Undo	u
Redo	shift +u
Properties	q
Rotate	r
Copy	c
Check and Save	F8
Zoom to Fit	f
Move	m
Wire Name	L

Useful keyboard shortcuts in layout view:

Action	Key
Create rectangle	r
More detail in layout	shift + f
Less detail in layout	ctrl + f
Stretch rectangle	s
Zoom to Fit	f
create ruler	k
clear all rulers	shift + k
Undo	u
Redo	shift +u
Copy	c
Properties	q