Laboratory 4: Layout, DRC, and LVS
Fall 2024

Table of Contents

Background Information

In the previous lab, you created a "layout" for a p-channel transistor. The process involved laying out the geometric features on several "layers" that comprised the transistor. After doing the layout, no additional actions were performed. In reality, there are many checks that need to be performed on a design before it can be fabricated; so many, in fact, that it's not worth listing here and certainly not worth checking by hand. For this reason, many checks are implemented with automated tools which go through and look for possible mistakes or rule violations in your design. In this experiment, two of these tools will be investigated.

Checkpoints

The checkpoints for this lab are as follows:

1. Completed Inverter Layout and DRC (from Part 1)
2. Completed Inverter Layout, DRC, and LVS (from Part 2)
3. NAND/NOR Testbench Results

These checkpoints must be shown to a lab TA before you submit your report. These checkpoints should be included in your lab report.

Part 1: Creating a Layout

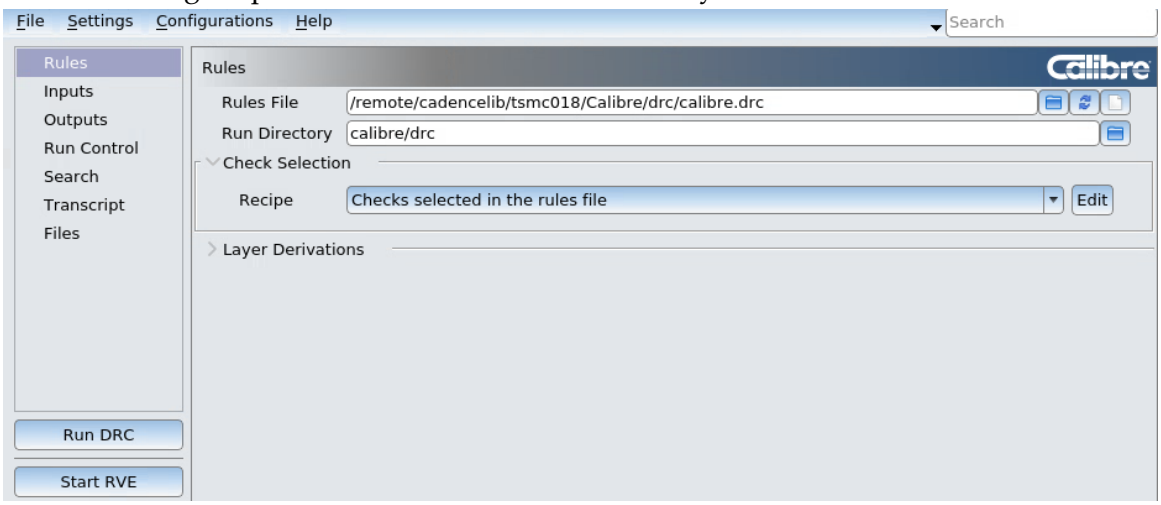
Open the layout view of the inverter that you created in Part 3 of Lab 2. For this first layout, we **will not concern ourselves with size**, but in later layouts making the transistors the proper size will be important.

Last week we created a p-channel transistor. This week we will use that transistor and add an n-channel transistor to the design to create an inverter. Before continuing however, we need to go through some of the checks. The first such check is DRC (Design Rule Check). This tool checks your layout to make sure the different trace sizes, shapes and positioning of your layout is compatible with the manufacturing process. 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 end. It is sometimes hard to see where Cadence finds an error, so run the DRC check frequently so you know where you need to

make changes. It must be emphasized, however, that the DRC tool checks only for design rule errors and not circuit errors.

Running DRC

Instructions for DRC are included in the caliber guide on the class website. We advise saving a runset after setting it up. The Rules File and Run Directory are shown below.

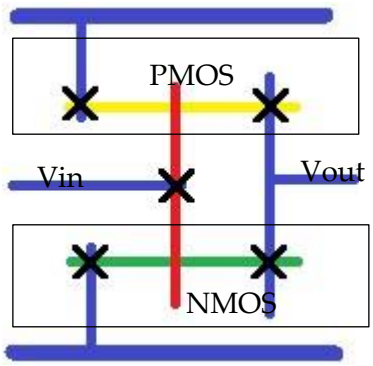


Make sure to run DRC frequently when working on the layout. There are many design rules that are checked by this tool, and fixing them is important.

Completing the inverter: From Stick Diagram to Physical Layer

One way to look at the physical layout of systems at a rather high level is with a stick diagram representation. A diagram of an inverter is on the right with:

- 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)

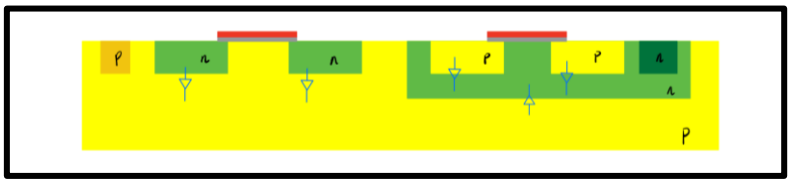


Finish the inverter by adding an n-channel transistor and the necessary connections to make your design look like the stick diagram. You already have the p-channel device so you will need to add the n-channel transistor as well as a Metal 1 line on top for Vdd and one on the bottom for Vss. You will also need to actually connect the drains and sources of the n-channel and p-channel devices with vias and Metal 1, but we will do that in the next few steps.

Bulk Connections

In lecture and design, we generally look at n-channel and p-channel devices as if they are three-terminal components, being made of only a gate, drain, and source. While these are typically the only terminals that we care about, this is not consistent with what MOSFETs actually look like, physically. In reality, MOSFETs have four terminals, with the fourth being the "bulk." In Virtuoso, the bulk connection for the nmos or pmos is the terminal located between the drain and source.

The bulk connection has several purposes in the operation of a MOSFET, including having an effect on the device's threshold voltage, but we'll ignore those purposes for this lab. What is important to realize right now is that, when you manufacture a MOSFET, you create a number of PN Junctions throughout the body of the device. Recall that a PN Junction forms a diode, as shown in the image below:



While these parasitic diodes can be useful in some cases (in fact, many cases), they can also be harmful. Consider if one of these diodes somehow becomes forward biased. Because the resistance that they see in series with them is minimal, if a single diode becomes forward biased, it can begin conducing large amounts of current – ultimately, enough current to destroy the MOS device that it is inside of. For this reason, it is critical that these parasitic diodes are kept reverse biased. This is one of the purposes of the MOSFET's bulk connection. In the image above, the extra p-doped region to the left of the n-channel device and the n-doped region to the right of the p-channel device form the bulk connections. In an n-channel device, the bulk connects to the p-substrate, and so it is desirable for this bulk to be connected to the lowest voltage present in a circuit (that is, V_{SS}). Because the n-channel bulk connects to the p-substrate, and there is only one p-substrate in an IC, it is only technically necessary to have one NMOS bulk connection in the circuit layout. In a p-channel device, the bulk connects to the n-well, and so it is often desirable for this bulk to be connected to the highest voltage present in the circuit (V_{DD}). Because the p-channel bulk connects to an n-well surrounding the p-channel device, it is necessary for every p-channel device in the layout to have its own bulk connection if there the n-wells for each p-channel device are electrically isolated from each other. It is possible to put several p-channel devices in a single n-well and in this case a single bulk connection for several p-channel devices is possible.

The **most common thing for new students to forget** when designing the layout is to create bulk connections. As stated in the previous paragraph, **each p-active region requires a bulk connection**. For the p-channel device, a via called an "M1_NWELL" is used to form the bulk connection. The n-active regions also need a bulk, so for these we will use the "M1_SUB" via. Unlike the M1_NWELL, most designs we will be making will only need one M1_SUB connection for the entire design.

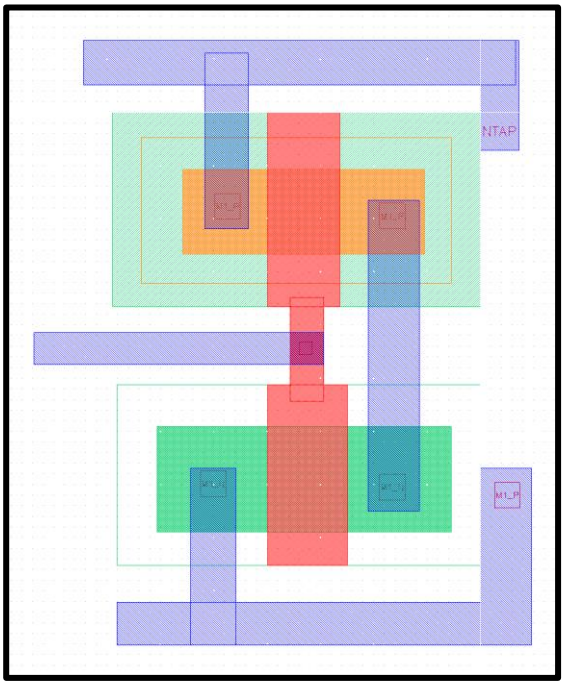
For reasons that will be discussed later, however, it is common to have more than one connection to an n-well or a p-bulk. This is necessary to prevent an undesired phenomenon that can cause the circuit to fail during normal operation that is termed "latchup". At this point, we will not worry about latchup issues.

After setting the NMOS bulk, we now need to address the Drain and Source connections to both the NMOS and PMOS. These connections need to be made to the P-active and N-active regions. Use an M1_DIFF for P active and N active, and then use Metal 1 rectangles to route the connection where you need it.

Now the only connection left is for the gates of the MOS devices, which are already made of POLY. For routing purposes, you can use "M1_POLY" to go from the Poly layer to Metal 1.

Create -> Via

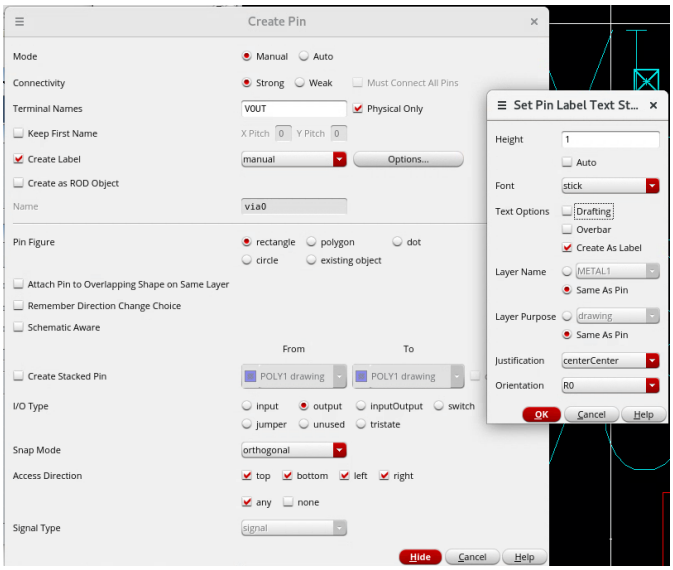
Your layout should now look something like this (with different Colors):



Pins

The final step to complete your layout will be to make inputs and outputs for your circuit. This is done so that when you use your component's layout in other designs, you will be able to run a check to make sure everything is connected to the right ports.

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 **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.

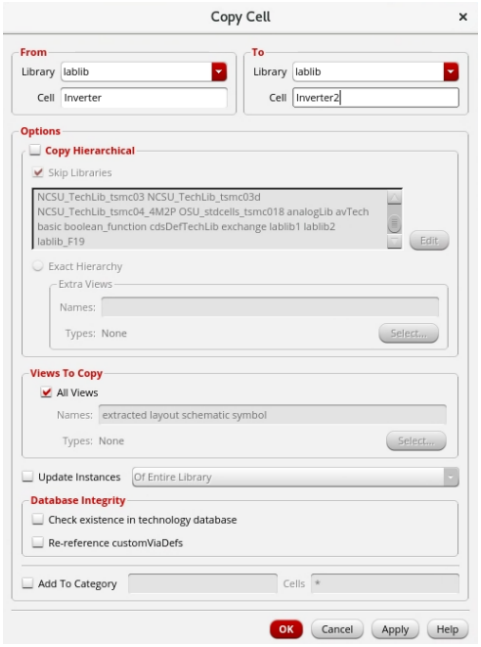


Create the pins for your Input, Output, and V_{SS} and V_{DD}. Once complete, run a DRC on your circuit.

Complete the layout of your inverter and showing a DRC with zero errors is your first checkpoint for this lab. (excluding the expected DRC errors related to density)

Part 2: Using P cells

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. This is much faster since they are pre-made and you can resize them to the ratio you need.

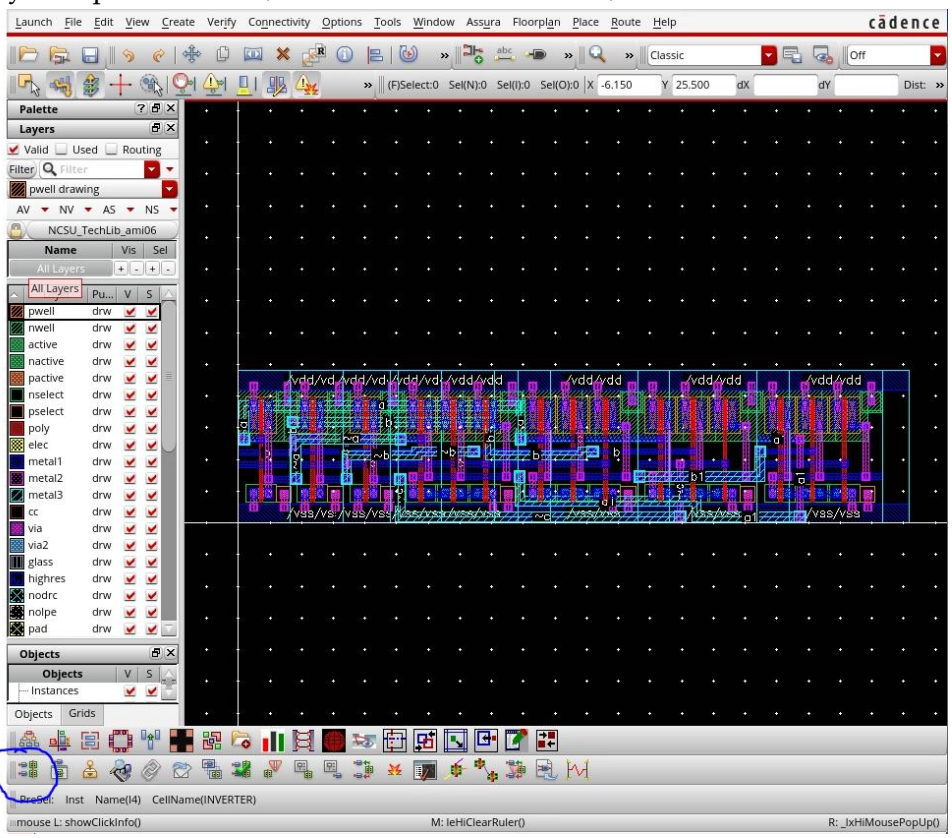


Start by going to the library manager and select the Inverter schematic. Right click on it and go to Copy, under "To" name the Cell "Inverter2". Create a layout for this new Inverter2 schematic **(File → New → Cell View → Layout)**. It will prompt you to overwrite the old one.

Generating From Source

Sometimes, when doing layout, it is more convenient to generate components rather than instantiating them. Luckily, Cadence is powerful software, and allows us to do this. However, use of these tools may not be intuitive or user friendly. So, this guide is meant to explain how to generate components in a layout and reduce the amount of work we do in cell placement and routing.

To generate our layout, first we will need a fully functional schematic; you'll probably want to have something that you know works, has been verified by a testbench, and only contains components that have layouts or standard cells. Now, launch Layout XL within Cadence like we would for any other layout. From here, navigate to the bottom left of the window, and press the "Generate All From Source" button, as seen below. This feature is also available under the "Connectivity" dropdown menu, under the "Generate" tab, as the "All From Source" button.

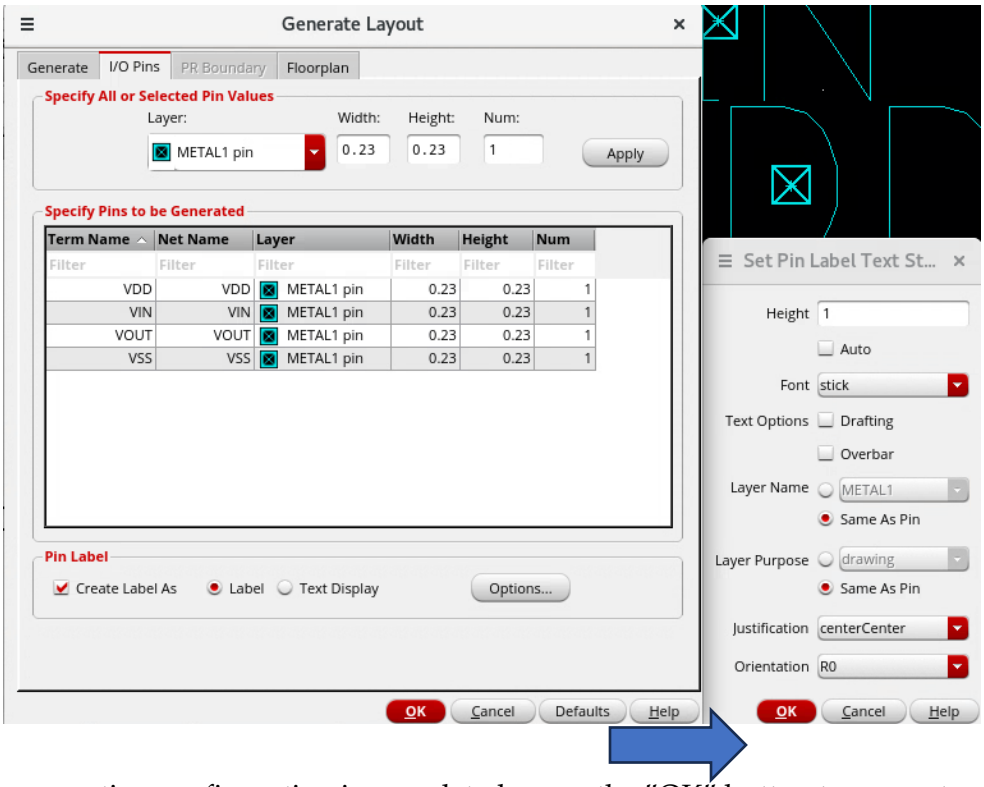


Pressing the above button should open a menu. If you already have objects placed in your layout, it will ask if you want to proceed. If you don't care about what you've already placed, go ahead and hit "yes". If not, make a backup copy before you proceed.

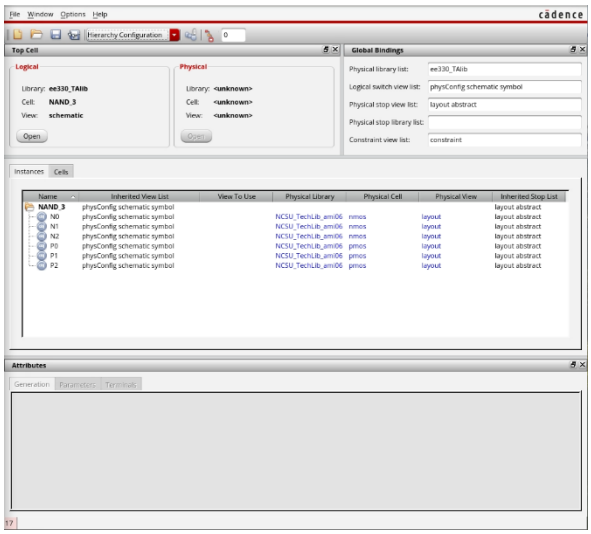
There are plenty of tools available to us in the generation menu. There are two tabs that we will want to use every time we generate a layout: the I/O Pins tab and the PR Boundary tab. The I/O pins tab will allow you to specify the size, quantity, and layer that each pin will be placed on.

The default configuration is usually good enough, and pins can be edited after they are generated. However, if you already have a plan in mind, this is where you can change pins before they are made. The PR Boundary pin will allow you to specify a boundary region for your layout; this is useful if you have a certain space constraint you must meet with your layout. You can specify shapes as well as a general shape and area utilization percentage. We will not be using a PR boundary for this layout. Make sure the Instances and I/O Pins are selected from the generate options.

Pins should generally be using a metal layer. For this invertor layout we are going to use Metal 1. The Pin label is how the the program identifies each pin later in verifcaiton. Ensure Create label as is checked and label is selected

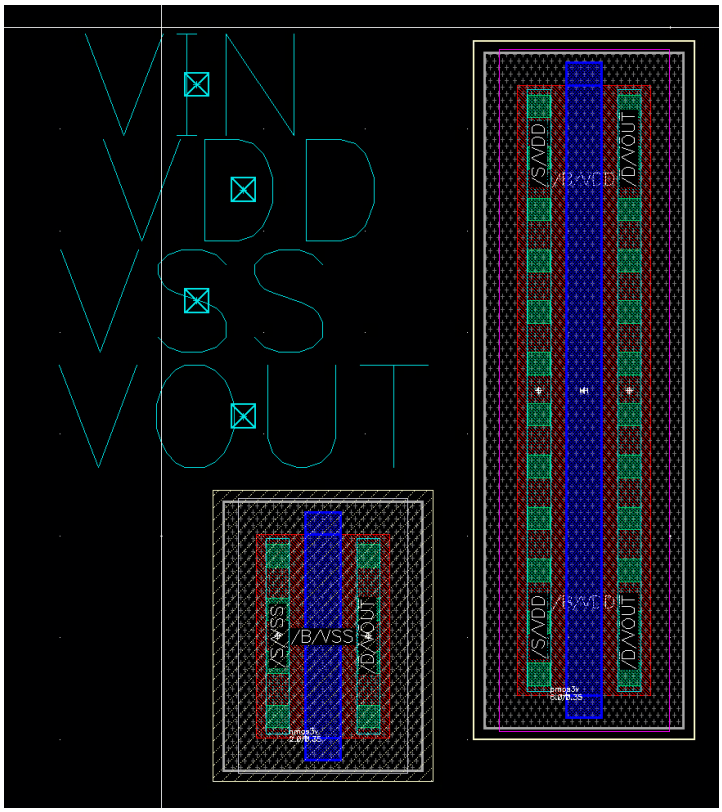


Once your generation configuration is completed, press the ”OK” button to generate your layout. This will place pins and standard cells in your layout as well as creating a physConfig cell view. If you only see your pins in your layout and not cells, the physConfig cell view is what you will want to change. This cell view contains all the information that the Generate from Source tool uses to place cell layout information within your layout. For example, if you wanted to generate a PMOS transistor, you would select the library that contains the process you want to use for Physical Library, and then specify PMOS under Physical Cell. This also works if you’ve already created a layout for a cell; select your lab library under Physical Library, and pick the cell you’re interested in under Physical Cell. Once you’ve successfully selected everything, save the file and rerun the Generate from Source tool. You should see the cells you are trying to if they did not generate previously.



Assuming you've done everything correctly, you should see every cell and pin you need for your layout. In addition, every terminal within your layout should now be labeled as to what net it corresponds to within your layout. The Generate from Source tool will also pull in the netlist from the corresponding schematic, giving the connections that will need to be made within the layout. This tool will prove valuable when it comes time to run the LVS tool; if you already know what connections need to be made, it is less likely that you will miss one and fail LVS check. Now that you have all of your cells and pins generated, you can place and route to your heart's content.

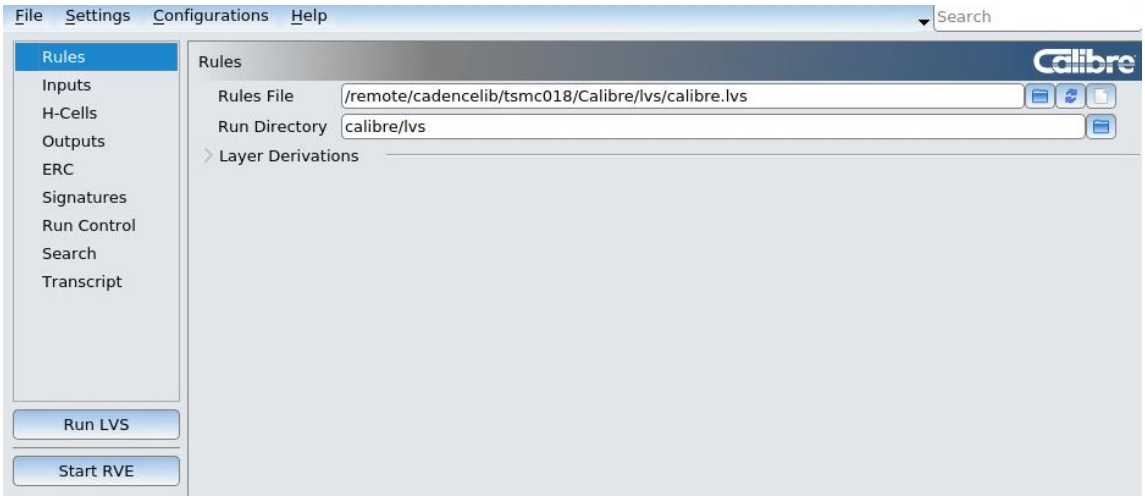
The generated MOSFETs will appear as a red box that says PMOS/NMOS, press **Shift+F** to view the individual cells placed**.**



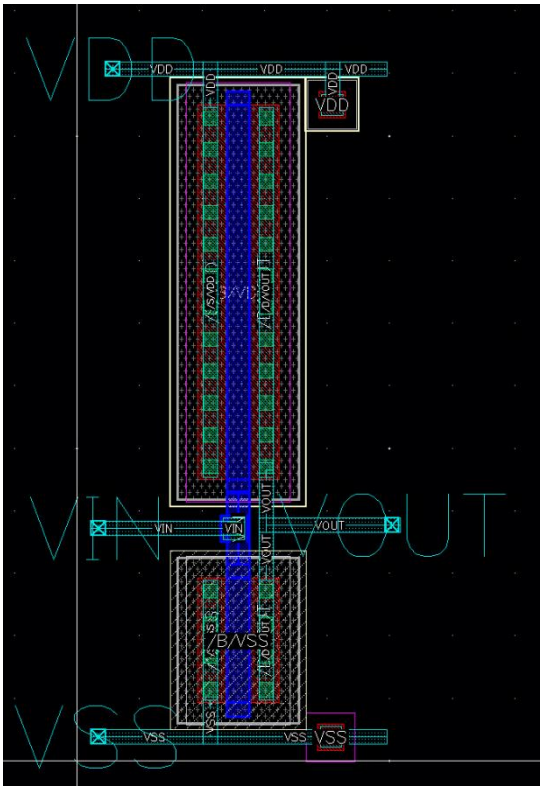
Ensure that all pins have a label, large text, with the name of the pin. These are vital to passing LVS later in the lab. If these are not present, either generate the layout again or place the pins individually using create -> pin.

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**. Remember to add the Bulks and Pins as well. When finished **run a DRC, extract, and run LVS**. Your correct DRC and LVS outputs are the third checkpoint for this lab.

Instructions for LVS are included in the caliber guide on the class website. We advise saving a runset after setting it up. The Rules File and Run Directory are shown below. LVS should be run after completing the layout.



Complete another layout of the invertor, fixing any DRC and LVS errors that bring



Part 3: NAND or NOR

In the next lab, a logic circuit implementing an arbitrary Boolean function will be designed jointly by two students. 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 that your gate works as expected.

Show the Schematic with a working testbench as the final checkpoint of this lab.

Looking Forward

Next week, we will be creating a layout for your three-input NAND / NOR gate, exchanging gates, and creating the schematic, test bench, and layout for your Boolean function. It will be 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.

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
Path	p
Via	o
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