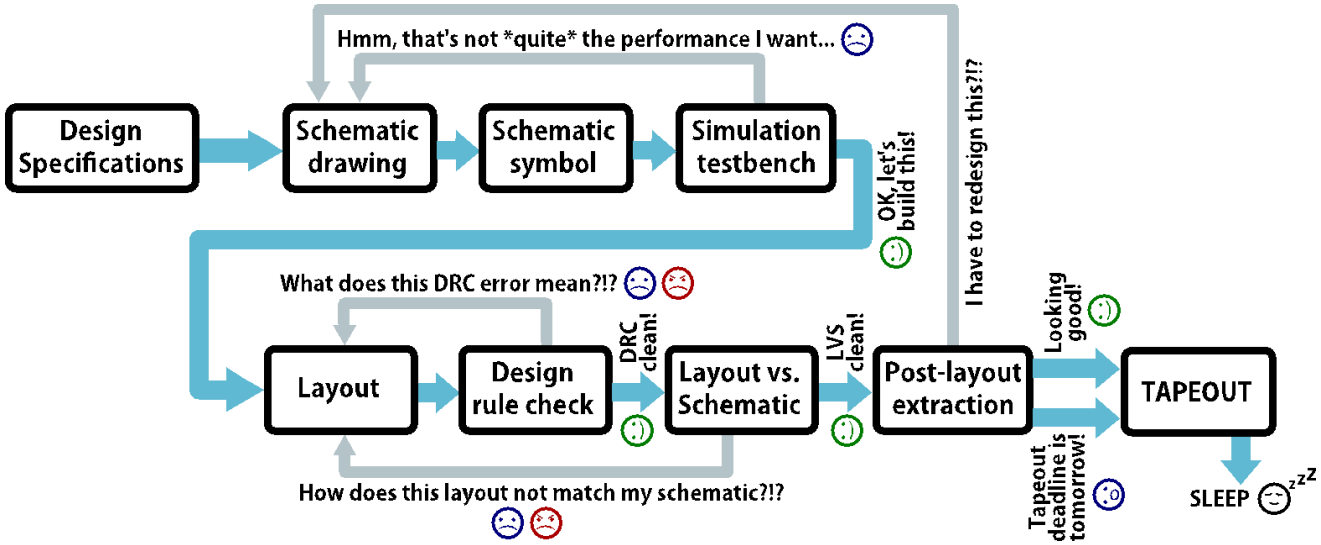
Lab 1: Introduction to Cadence

Sashank Krishnamurty, Matthew Dharmawan, Micah Roschelle
Spring 2024

CONTENTS:

OBJECTIVE

The goal of this lab is to introduce you to Cadence Virtuoso, an industry-standard IC design tool that we will be using throughout the semester. To do this, the lab walks through the process an analog design engineer might use for chip design: **creating a design schematic, wrapping this in a symbol, and setting up a simulation test bench**. Realistically, the full design process also involves layout, developing the physical layout of the design, verifying the layout is correct, and simulating the design again including parasitic elements from the layout… then there is a fair amount of iteration (as shown below); however, that won't be necessary in this lab, which means it won't leave you too sad or frustrated.



Typical IC design process.

DELIVERABLE

This lab has a checkoff to demonstrate understanding and a required formal written report. Think of a checkoff as a way to keep you on track. When you have finished the lab up to the checkoff, show the GSI the deliverables on the last page. **Plots should be understandable: traces should be visible and axes should have legible labels.** We will provide instructions for generating a "legible" plot in the lab, so as long as you follow the instructions you'll be fine. (TODO: Add this legible plot section)

1. Remote Server Login

You may want to access the servers from home in order to use Cadence outside of the scheduled lab times for future homeworks and the project. We recommend using X2Go instead of SSH, as Cadence can be very graphics intensive and may run slowly on SSH. For this lab, feel free to work either on the lab machines in front of you, or from your own laptops. You can also run X2Go from the Windows machines by the lab benches.

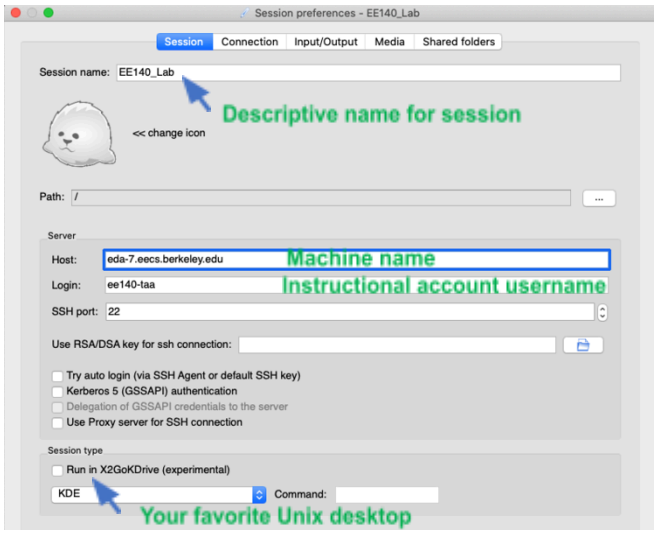
To setup X2Go on your laptops:

1. (For Mac users) Download XQuartz here: https://www.xquartz.org/

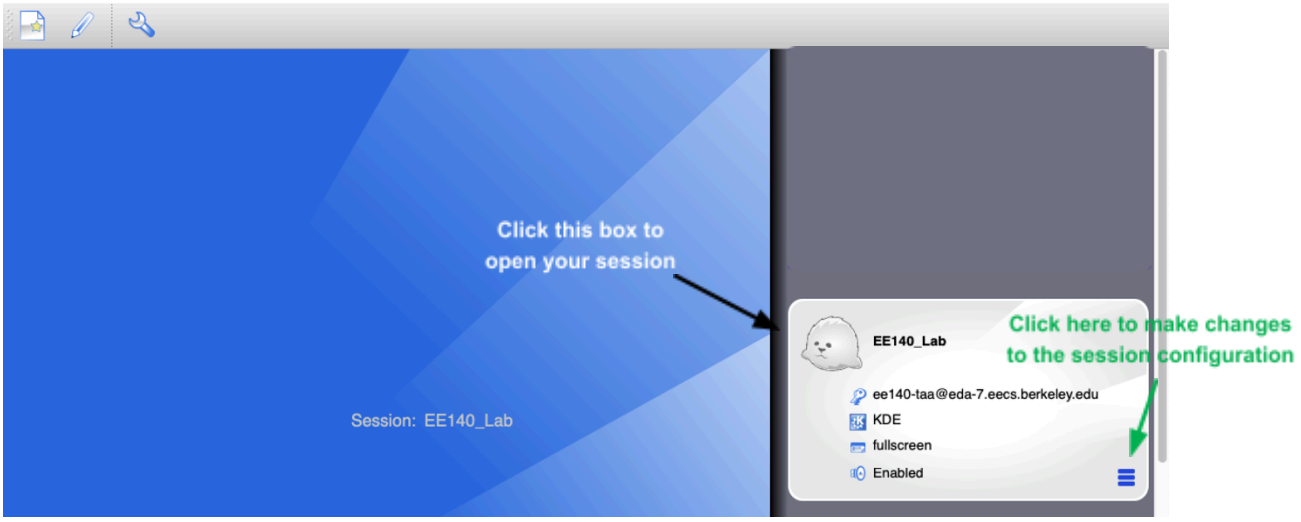
2. Download the appropriate (for your operating system) version of the X2Go client from the following website: http://wiki.x2go.org/doku.php, and then open X2Go.

3. Create a new connection by clicking on the “New session” icon at the top left: 

4. Fill out the session preferences as below, replacing the “Login:” field with your username. To choose a machine, type [machine name].eecs.berkeley.edu in the “Host:” field. The example below connects to the eda-7 machine. In this class, you can remotely log in to eda-1 through eda-11. Choose GNOME (if GNOME does not work, try MATE) from the “Session type” drop-down menu at the bottom. Once you have finished, click OK.



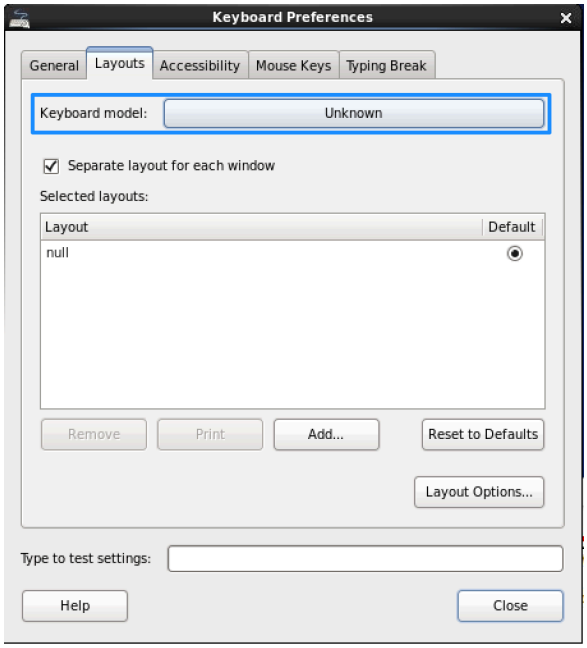
5. To connect to the session you just configured, click the gray box with a balloon-like seal creature on the right. When the authentication window opens, provide your instructional account login information. If you want to make changes to your setup, click the down arrow in the bottom right corner of the box.



6. After you provide your account information, a new window with your remote connection should open. You may see pop-up windows warning you about SSH problems, but these can be ignored.
7. When you are finished, **save your work** – sessions sometimes have problems opening again – and close the remote desktop window or click the pause button ( ) in the X2Go client window. You can then close X2Go and connect to your session later (or leave X2Go open if you plan to re-connect soon).
8. To connect to your session again, simply click the same gray box with the session you created next time you open X2Go.

Remote desktop software such as X2Go is convenient because it allows you to suspend a session and then pick up your work later. Save your work frequently, though – connectivity issues can result in dropped sessions from time to time.

Note: If you use X2Go to switch between Windows and Mac machines, you may notice keyboard mapping problems. This seems to be a problem on GNOME, but not KDE desktops. To change the keyboard mapping on your desktop in GNOME, go to System Preferences Keyboard, and select the “Layouts” tab. From here, you can select the appropriate keyboard model in the menu (“Macintosh Old” seems to work for Mac devices):



2. Cadence Setup and Launch

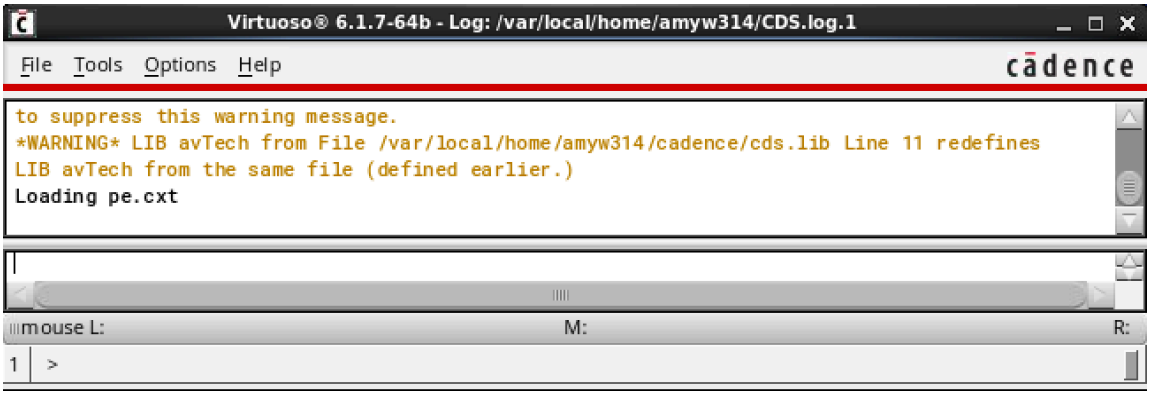
Cadence is the one-stop-shop for nearly all of your analog circuits needs. To set up Cadence on your instructional account, open a command terminal (in GNOME, you can do this by right-clicking anywhere on your blank desktop and selecting “Open in Terminal”) and navigate to your home directory (you can do this by typing `cd ~`). Then type the following commands:

```
cp -r /home/ff/ee140/fa18/cadence .
cd cadence
bash
source /share/b/bin/cds6
virtuoso &
```

If you are unfamiliar with Unix: The above commands will create a local copy of the ‘cadence’ folder from the ee140 instructional directory, and then source the setup script it contains to configure your operating environment to run Cadence. Once this is done, you can open Cadence with the ‘virtuoso’ command (the `&` just ensures that Cadence will keep running even if you use the terminal to run other commands).

NOTE: Next time you need to start Cadence, navigate into this directory and re-type the last two commands (“`source /share/b/bin/cds6`” and “`virtuoso &`”) – no need to recreate the Cadence directory*! If you do, you will lose your existing libraries.*

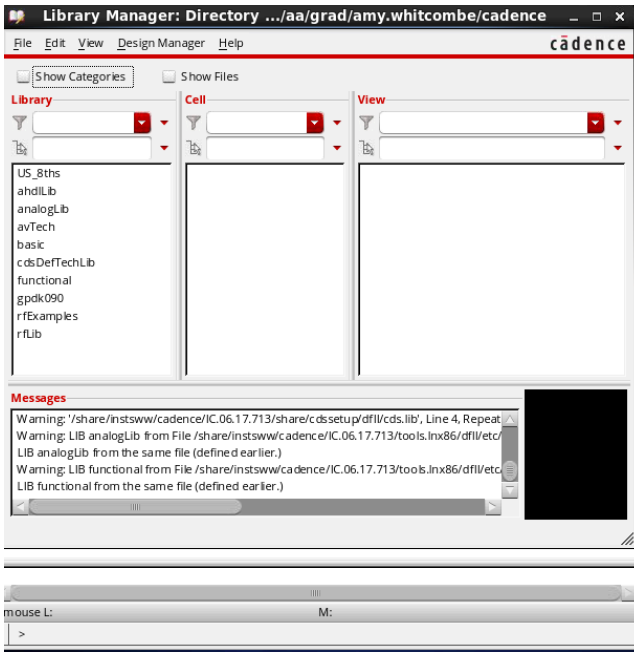
Note: You might experience the case that the virtual desktop gets suspended. You have to re-login but this shouldn’t stop you seeing the contents below.
You should see the Command Interpreter Window (CIW) appear:



To see Cadence documentation, you can click the “Help” drop-down menu in the CIW. There are also many online tutorials and forums that discuss using Cadence if you would like to learn about additional features that we won’t be able to fully cover in this class.

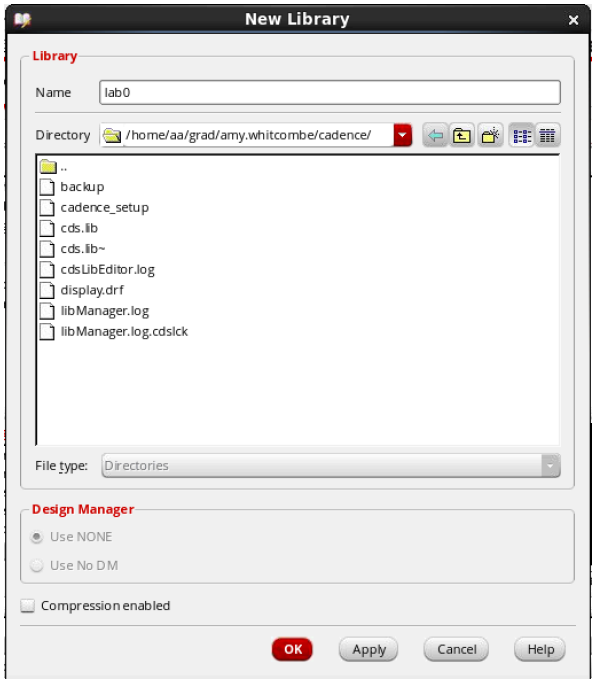
3. Creating a Schematic View

Next, open the library manager by going to Tools □ Library Manager:



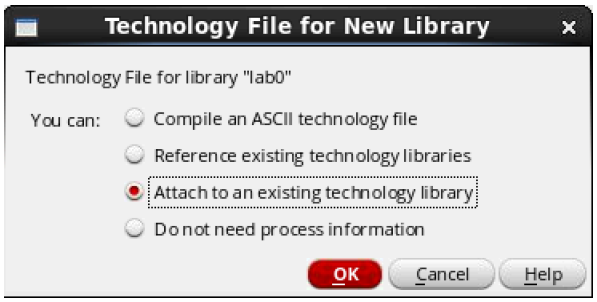
In Cadence, there is a relatively straightforward hierarchical organizational structure. Libraries (the leftmost column in the library manager) are collections of Cells, and Cells are collections of Views. For example, you might have an “ee140” library which has several cells such as “op-amp” or “current mirror”, with each cell having “schematic,” “symbol” and “layout” views.

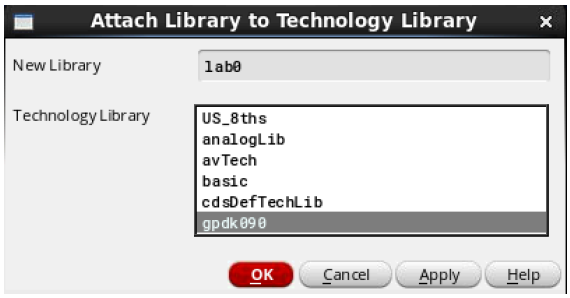
The first thing that you will need is a new library. To do this, go to the library manager and click File □ New □ Library. A new window will pop up. Type “lab1” in the name field and press OK.



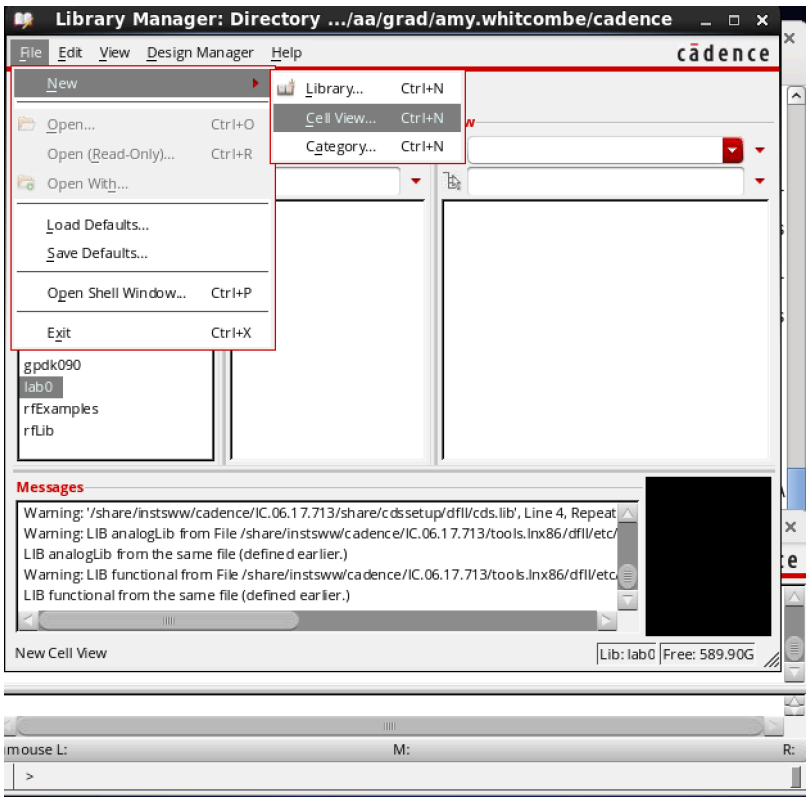
At this point, Cadence will prompt you for something called a Technology File. The technology file is a collection of information and libraries that define the layers and devices available for a given process technology.

To import the technology file for this class, click the ‘Attach to an existing technology library’ button, press OK, and use the menu to select ‘gpdk045’. This stands for “general-purpose design kit, 45 nm.” “General purpose” refers to the type of process (typical choices are general purpose GP, low power LP, high performance HP, and radio frequency RF), and the “45 nm” refers to the minimum feature size available.

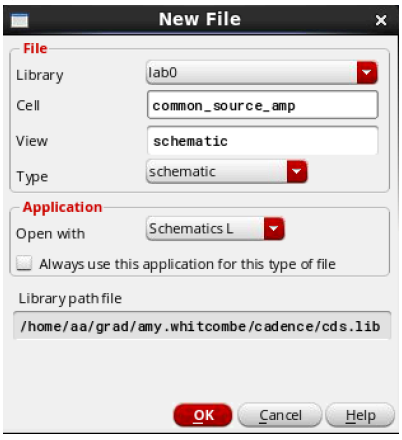




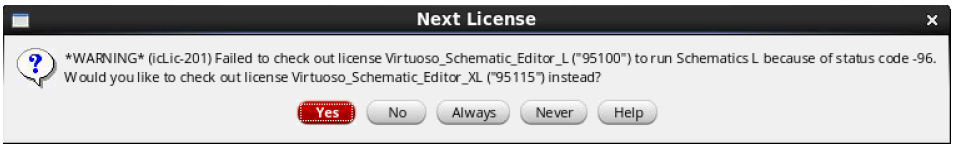
Now that you have a library, you can create your first schematic. Select your new ‘lab1’ library in the library manager, and click File New Cell View.



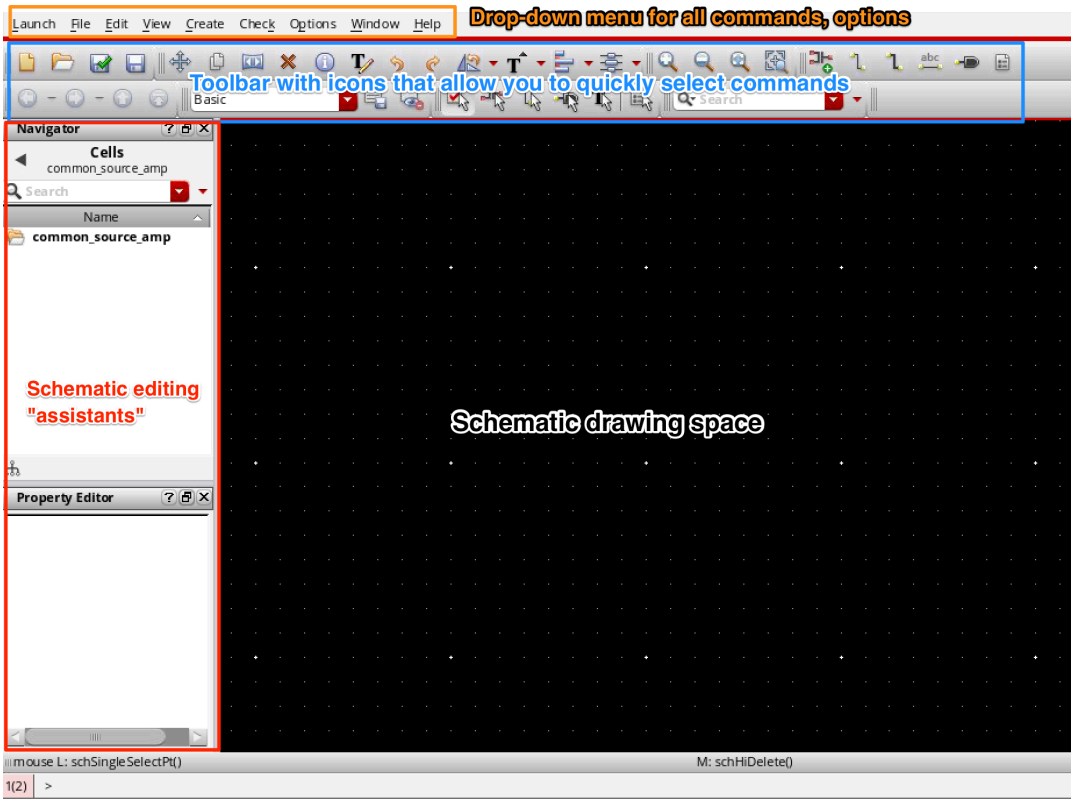
When prompted, name the new cell “common_source_amp,” enter “schematic” in the View field, and select “schematic” from the Type drop-down menu, as shown below.



The schematic editor should open up. Now we can build a schematic representation of our circuit at the transistor level. You may see a warning about licenses; feel free to click “Yes” or “Always” if you see a dialog box like this:

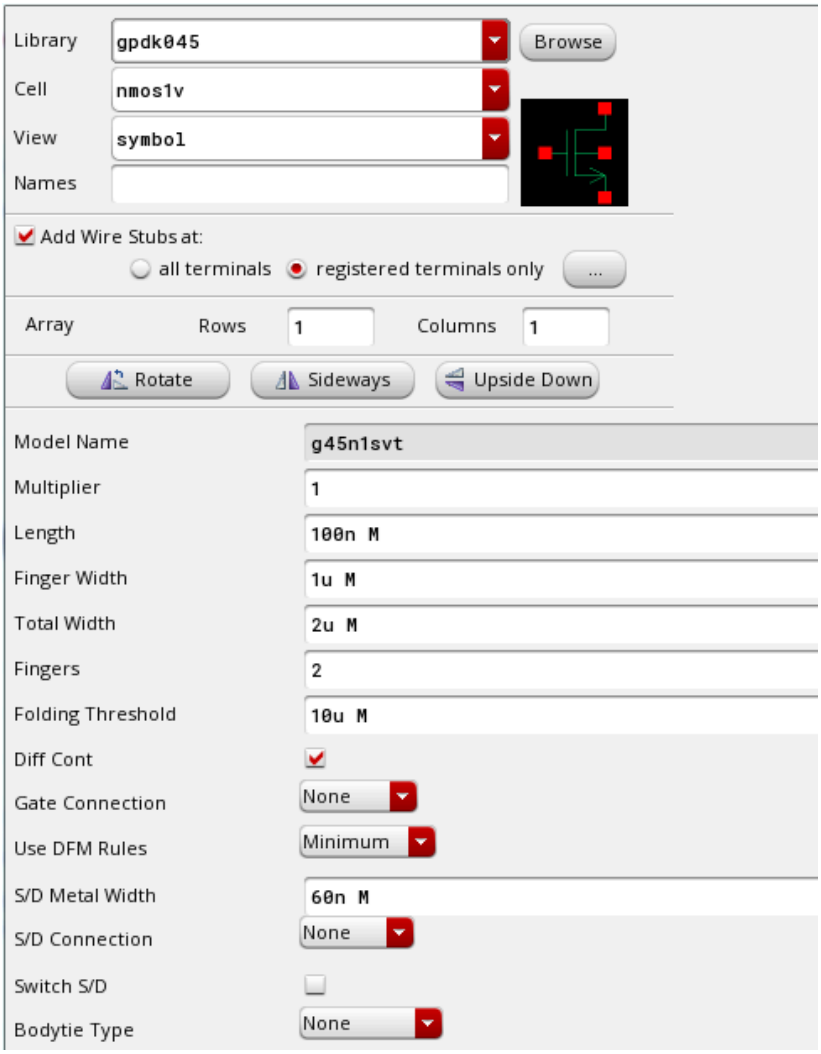


The schematic editing window is shown below. Almost all commands and settings can be configured with the top set of drop-down menus, but there is also an icon toolbar below that allows you to quickly select some common commands. Many of these commands can also be executed via hotkeys; we will provide a supplemental handout that summarizes many of these. At the left, you can add “assistants”' to help you manage the schematic.



To instantiate circuit elements in the schematic, press the ‘i’ key to bring up a menu with fields “Library,” “Cell,” “View,” and “Names.” To place an NFET into your schematic, use the “Browse” button and navigate to gpdk045 □ nmos1v □ symbol. You can also type this manually (“Library” is gpdk045, “Cell” is nmos1v and “View” is symbol). Now you will see more design parameter fields in the menu, as you are actually adding a parameterized cell (“pcell”) to the schematic. The transistor gate width, length, and number of fingers (essentially the number of parallel devices) are all parameters, and your choices will be reflected in the schematic.

Set the NFET to have length = 100n, 2 fingers, and total width = 2u (suffixes like ‘n’ and ‘u’ will automatically be interpreted as multipliers like 1e-9 and 1e-6). Generally, the number of fingers is chosen to minimize parasitics, constrain the layout area, or improve matching. Press the “Hide” button or hit the Enter key in the menu, and then click on the schematic to place the transistor. Afterwards, you can press the Escape key, since we will not be placing multiple devices.



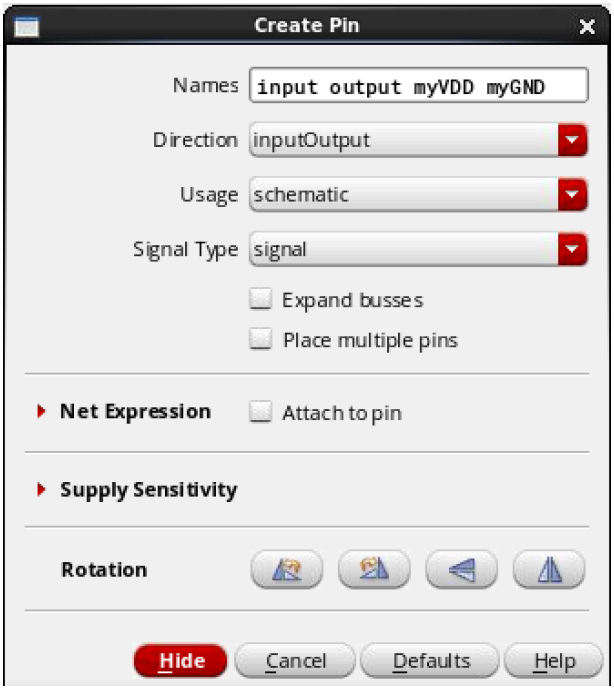
If you ever want to change an instance's properties later (e.g., to resize the transistor), click on the instance to select it and then press 'q'.

Next, instantiate a polysilicon resistor (resnsppoly in the gpdk045 library) with 2 series segments 500 nm wide and 3 µm long using the same procedure as the NFET. It should have a total resistance of 7.8 kΩ. Note that there are many possible widths and lengths that could yield the same resistance, since resistance is dictated by the number of square segments in the resistor and the resistivity of the material being used. In practice, sizing is often decided based on factors such as area/form factor, matching, and electromigration (maximum current density).

Now that all of our components are in the schematic, you will need to connect them together to build a common source amplifier. Press the 'w' key and click between terminals (red squares on each instance) to add a wire. Note that the bulk terminals of the NFET and the poly resistor must be connected to ground, so they should be tied to the source of the NFET. You can also click to select an instance, and then press the spacebar to automatically add labeled wire stubs, which can be renamed by double-clicking the text. To undo or redo an action, press 'u' or 'Shift + u'. To delete a trace, you can use the 'Delete' key (or click to select the object you would like to delete, and then right click Delete).

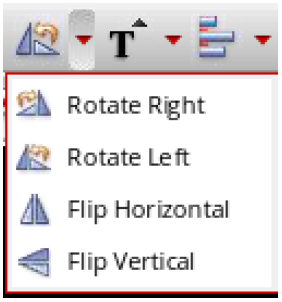
To navigate the schematic, use the arrow keys on the keyboard to pan around. You can zoom in/out by using your scroll wheel, clicking the magnifying glass buttons on the toolbar, or with the hotkeys '[' and ']'. To zoom into a specific area, drag a box around a region of interest with a right mouse click. Alternatively, you can press 'z' and click two points to create this box.

Lastly, you need pins to mark the input and output of your amplifier and its supplies. The pins allow your design to be used in higher level schematics. There are 4 ports we will want to expose out of the amplifier: the NFET gate (input), the NFET drain / bottom of the resistor (output), the NFET source (ground), and the top of the resistor (V_{DD}). Press 'p' to open the Create Pin window, and type the names of your pins. Here I used "input", "output", "myVDD", and "myGND". Make sure that you select "InputOutput" in the direction dropdown menu, click "Hide," place your pins in the schematic, and add them to the schematic. You can add each of the pins individually, or type each name separated by a space as follows:



If you do this, a new pin will be created every time you click on the schematic. In order to modify the schematic, here are some useful hot keys to know (see the supplemental handout for more):

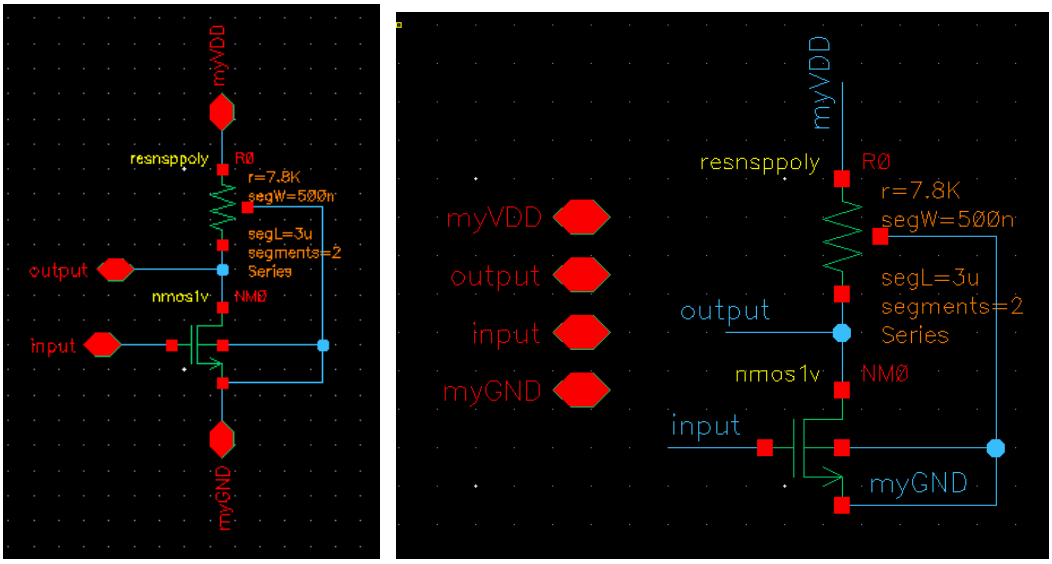
- 'm' / Edit □ Stretch: Moving a component while maintaining wire connections
- 'Shift + m' / Edit □ Move: Move a component without maintaining wire connections
    - Use 'Shift + r' to mirror a component while you are moving it, and 'r' to rotate a component while you are moving it
- 'r' / Edit □ Rotate: Rotate a component 90 degrees
    - Many of the commands for mirroring and rotating are also available via icons at the top of the schematic editing window, if you click the red downwards triangle next to the 'rotate left' icon:



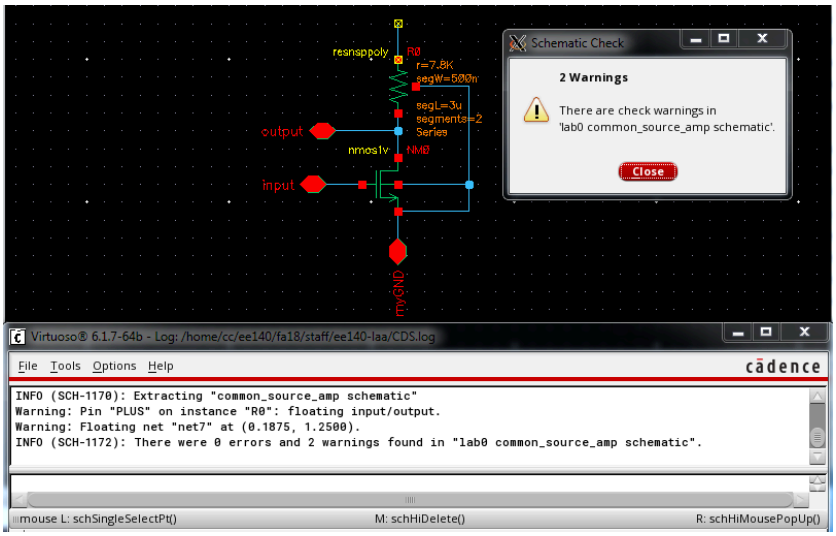
- 'l' / Create □ Wire Name: Add labels to wires. Sometimes it is useful to label nodes instead of directly connecting them with a wire. To connect two nodes, just label them with the same name. Note that if you type a long list of names separated with spaces, a new label from your list will be placed each time you click on a wire.[1]

[1] We won't be dealing with buses much in this class, but if you type a label like 'DigitalSignal<4:0>', this will be treated as a bus containing 5 separate wires. You can also check the 'Expand bus names' option in the label menu to quickly place 5 separate labels: 'DigitalSignal<4>' through 'DigitalSignal<0>'.

Feel free to try out some of these commands to get a feel for schematic editing in Cadence—familiarizing yourself with shortcuts in the tool can be really helpful. Once you have finished, if you followed the instructions carefully, your schematic should look like the one of following:



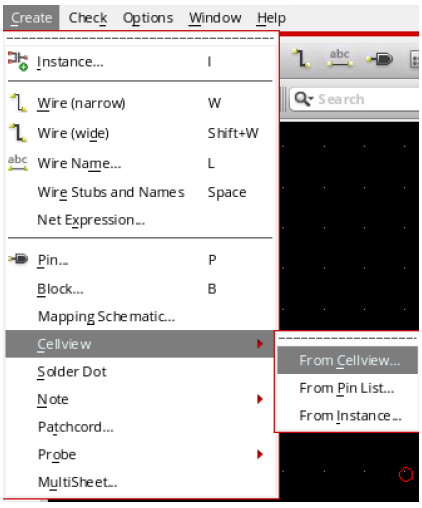
When it does, check and save the schematic (click  or press 'Shift + x'). You shouldn't see any warnings; if you do, the part of the schematic associated with the warning should flash yellow, and you can find more details about the issue in the CIW. Here's an example of what happens when the output pin is disconnected:



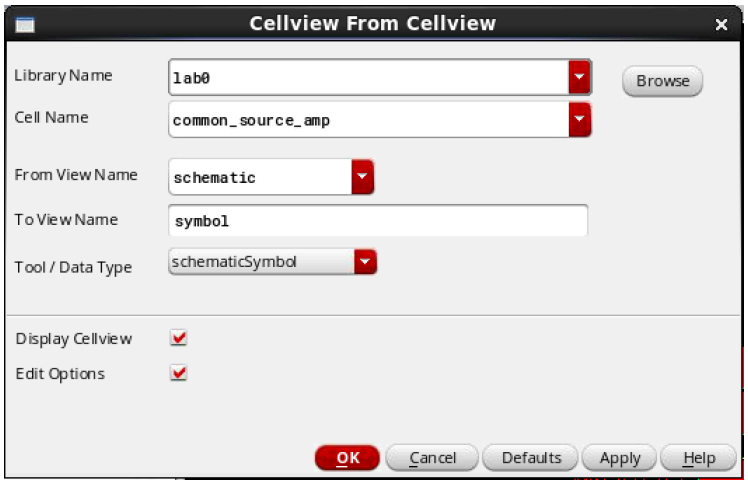
Once your schematic is free of warnings, congratulations—your first schematic is done! Keep the schematic editor open, because we will use it for the next step.

4. Creating a Symbol View

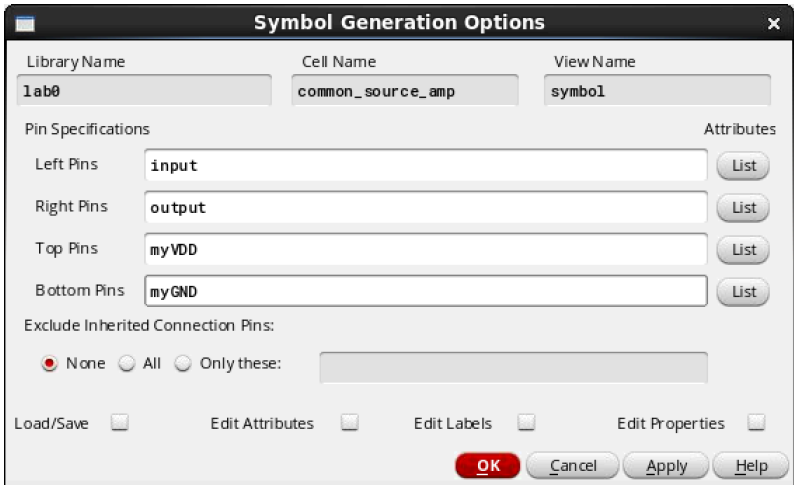
Now that your amplifier schematic is finished, we're going to wrap it in a symbol, so that we can use it in other schematic cells such as simulation test benches. To create a symbol, click on Create □ Cellview □ From Cellview in the schematic editor window:



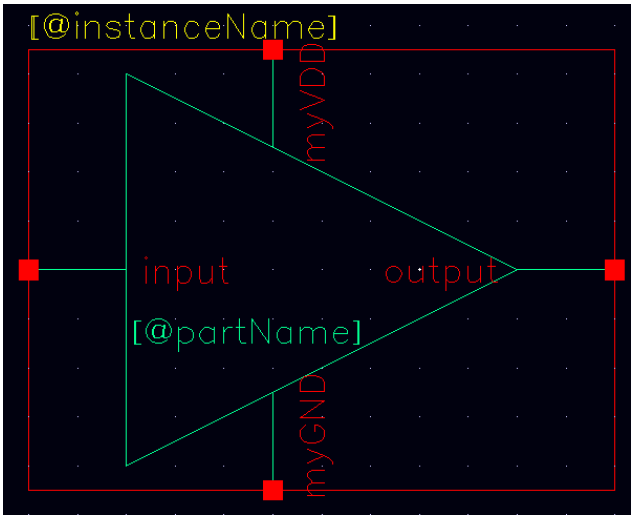
Populate the pop-up window as follows and press OK.



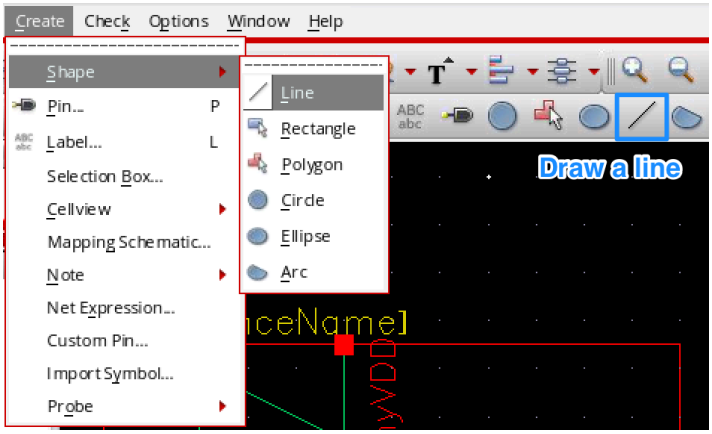
In the next window, you can modify where the default pin names appear if you would like (these can also be set later when drawing the symbol). By convention, inputs should be left hand side of the symbol, outputs on the right, VDD at the top and VSS or GND at the bottom:



Click ‘OK’ to generate the symbol. Edit it to look like an amplifier, as shown below. It is good practice to make symbols understandable, so that schematics using them will be readable.



To draw the triangle, you can delete the green rectangle and then draw your own shape with the Create □ Shape drop down menu or by clicking the appropriate icons in the editing toolbar:



Many of the hotkeys used for schematic editing are also valid for editing symbols (e.g. ‘m’ for moving objects and ‘r’ for rotating), with a few exceptions – for instance, ‘w’ will not draw a line. When you place the symbol in a schematic, it also appears a bit differently. The red box that you see in the symbol editor will not be displayed -- it represents the selection boundary of the symbol. The [@instanceName] term will be replaced by a specific instance number (e.g., ‘I0’), and [@partName] will be replaced by the name of the cell (‘common_source_amp’ in this example).

When you are finished, check and save your design. Take a screenshot for checkoff and the report! Try to make your best triangle amplifier shape. When you are done, close the symbol editor when it is free of warnings (you can debug any warnings through the CIW).

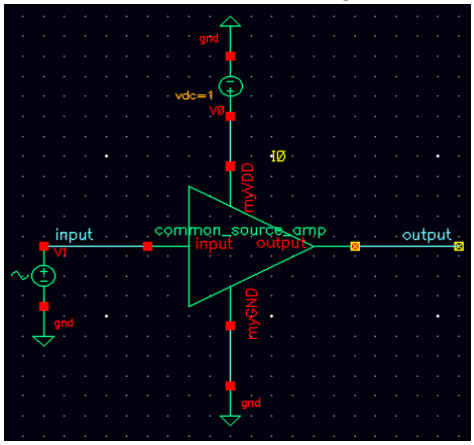
5. Circuit Simulation with ADE

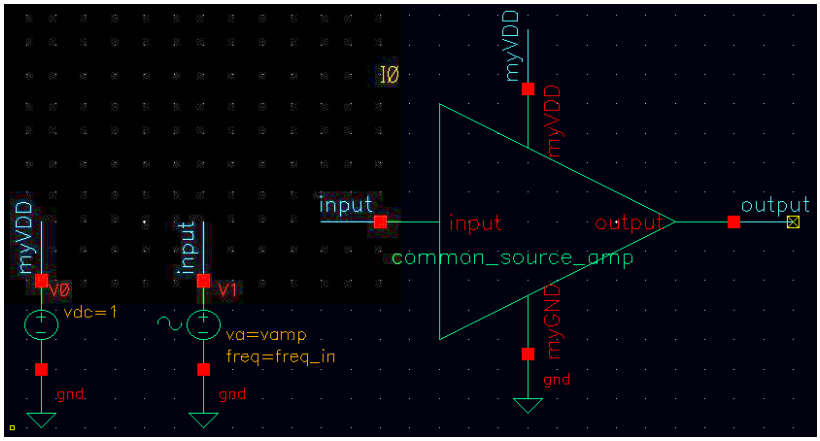
To simulate your amplifier, you will need to build a test bench. To create this, go back to the library manager and make a new schematic called “common_source_amp_tb” in the lab1 library. Then, add the following instances (press ‘i’, just as you did to place instances before):

- Your amplifier (“symbol” view of the “common_source_amp” cellview in the “lab1” library)
- A “gnd” instance from the “analogLib” library (connect this to myGND on the amplifier)
- A “vdc” instance from the “analogLib” library – this will be our supply.
    - Set its DC voltage to 1.1 (units are volts).
    - Connect this between myVDD on the amplifier and another gnd symbol.
- A “vsin” instance from the “analogLib” library – this will be the input to our amplifier.
    - Set its DC voltage to “vdc,” Amplitude to “vamp,” and Frequency to “freq_in” as shown below. Assigning these parameters to variables allows you to change them easily and sweep them in the simulator.

    - Connect this source between the amplifier’s input and another gnd symbol.

You should label the input and output nets so they can be easily identified when viewing the simulation results. Finally, check and save your schematic. You may see 2 warnings about the floating output of the amplifier, which are safe to ignore. **Take a screenshot of your testbench for checkoff and the report.** Your test bench schematic should look like one of the following:



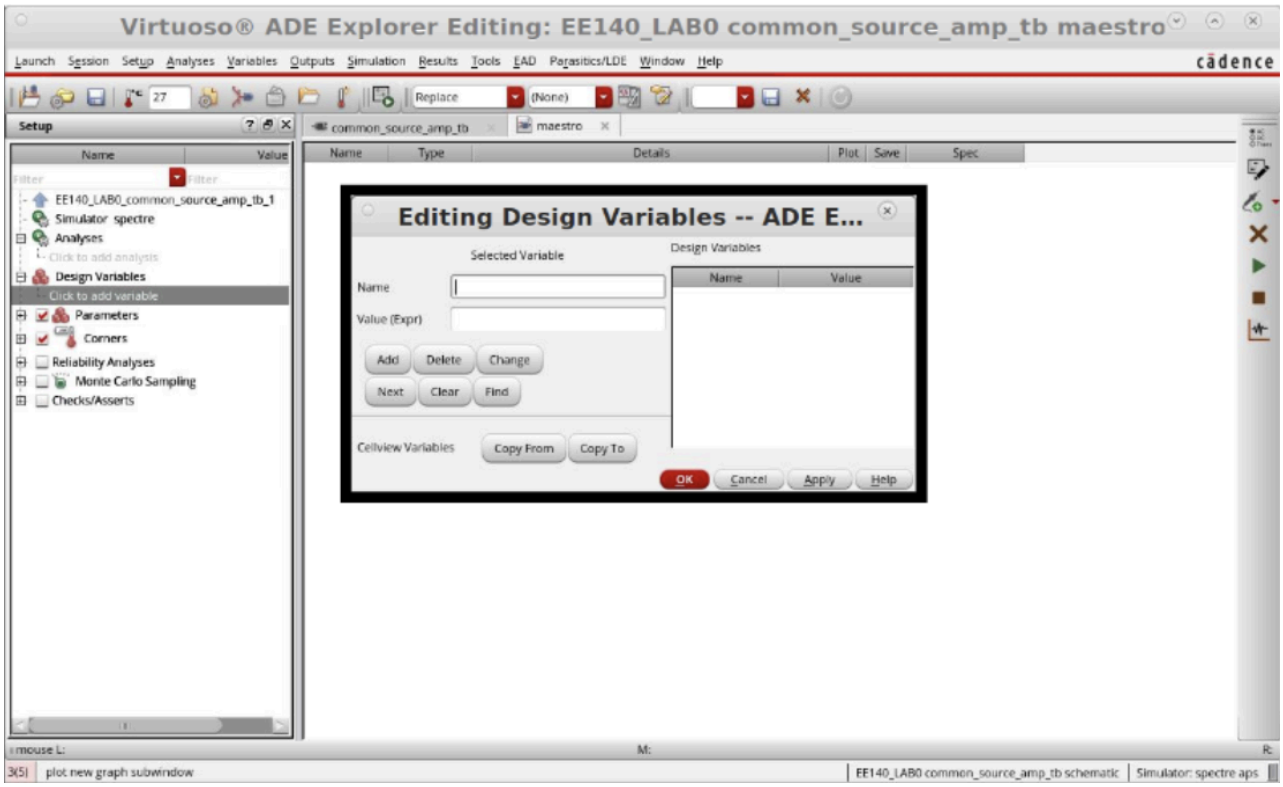


Now you can set up the simulation for your amplifier. Click Launch ADE Explorer to open up the Virtuoso Analog Design Environment (ADE). In the window, select “Create New View” as shown below.

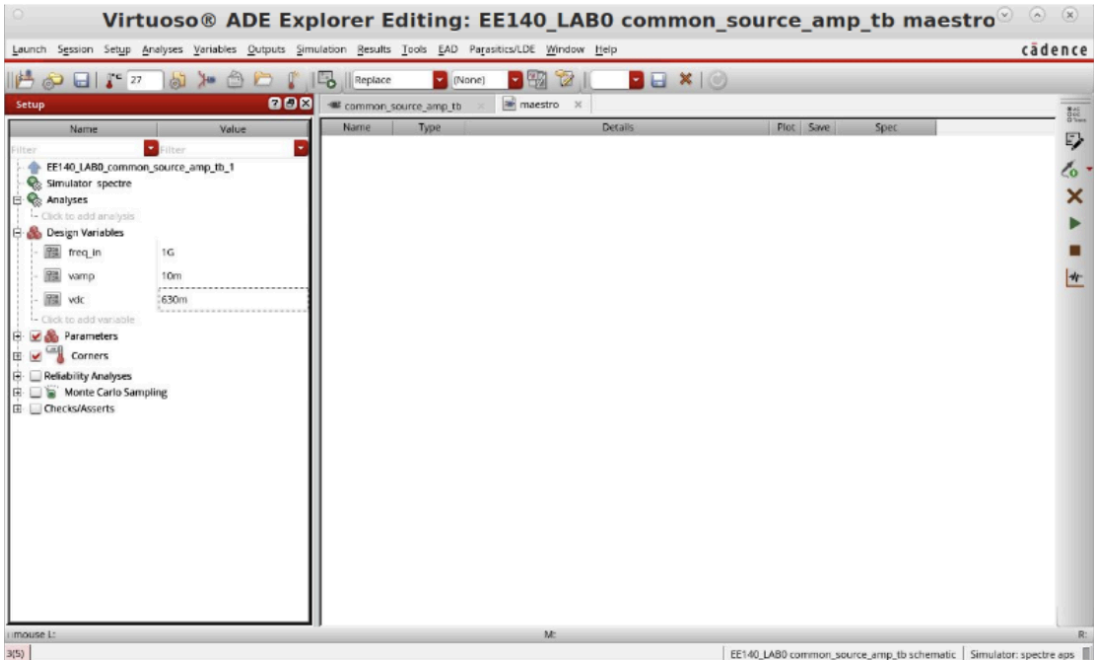
There are 3 main steps in running a simulation in ADE, corresponding to the 3 panes in the ADE Design Environment window:

1. Setting up the design variables/parameters in the leftmost pane
2. Setting up the analyses (simulation types) in the leftmost pane
3. Setting up the simulation outputs in the right pane

Let’s first set up our input parameters. Under “Design Variables” click on “Click to Add Variable” text and select “Copy From” at the bottom of the menu.

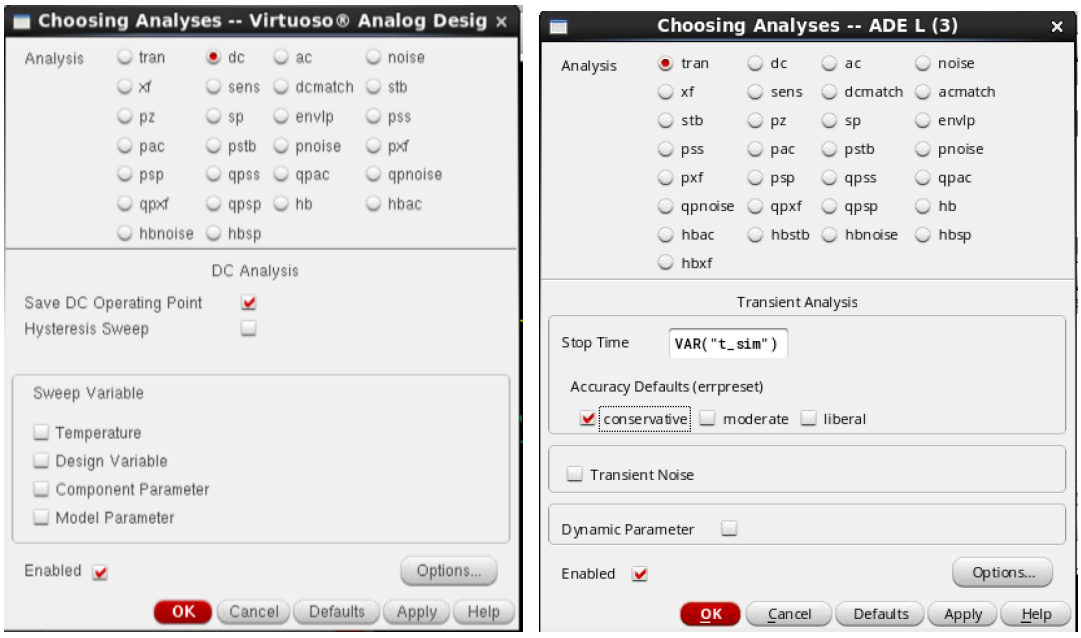


For now, set “freq_in” to 1G, “vamp” to 10m, and “vdc” to 630m, as shown below:



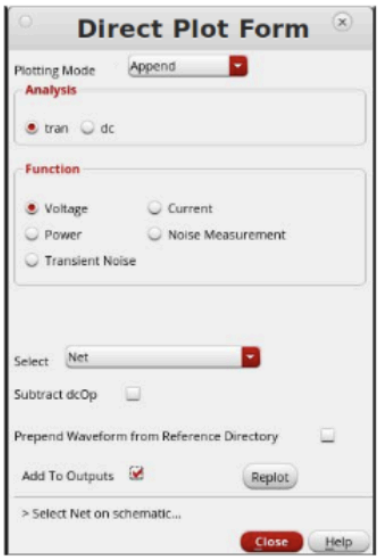
We’ll next set up our first simulation, a DC simulation to measure the operating point. To do this, click on the “Click to add Analysis” text under the Analysis drop down on the left. Select “dc,” check “Save DC Operating Point” (as shown below), and hit OK.

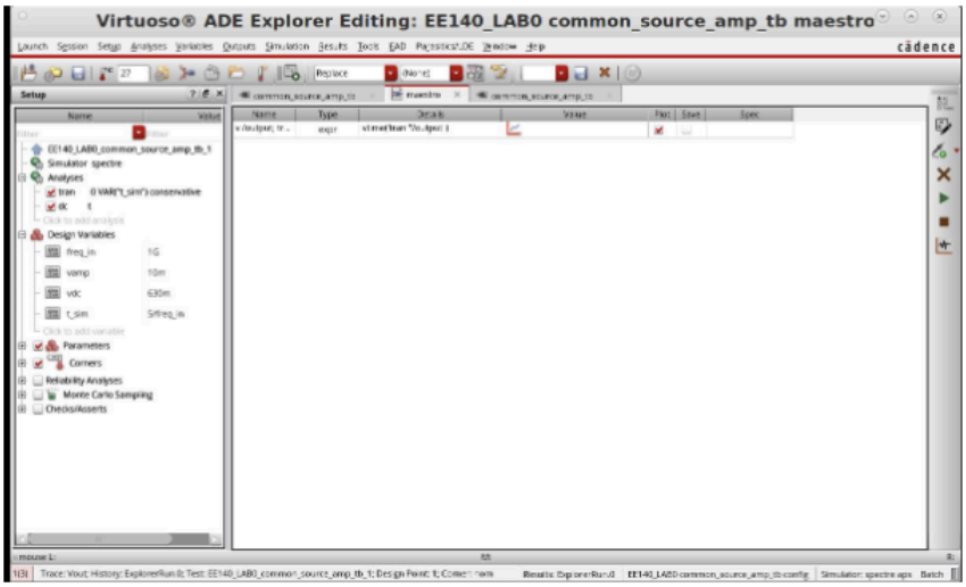
Next, set up a transient simulation. Open the “Choosing Analyses” window again, and select “tran” (for transient). Set the stop time to “VAR(“t_sim”)” and accuracy to “conservative” (as shown below), and hit OK. The variable “t_sim” should have appeared in the Design Variables pane. Give it a value of “5/freq_in.” This will run the transient simulation for 5 input waveform periods, and will scale with your choice of freq_in. DC analysis (left) and transient analysis (right) configuration.



DC analysis (left) and transient analysis (right) configuration.

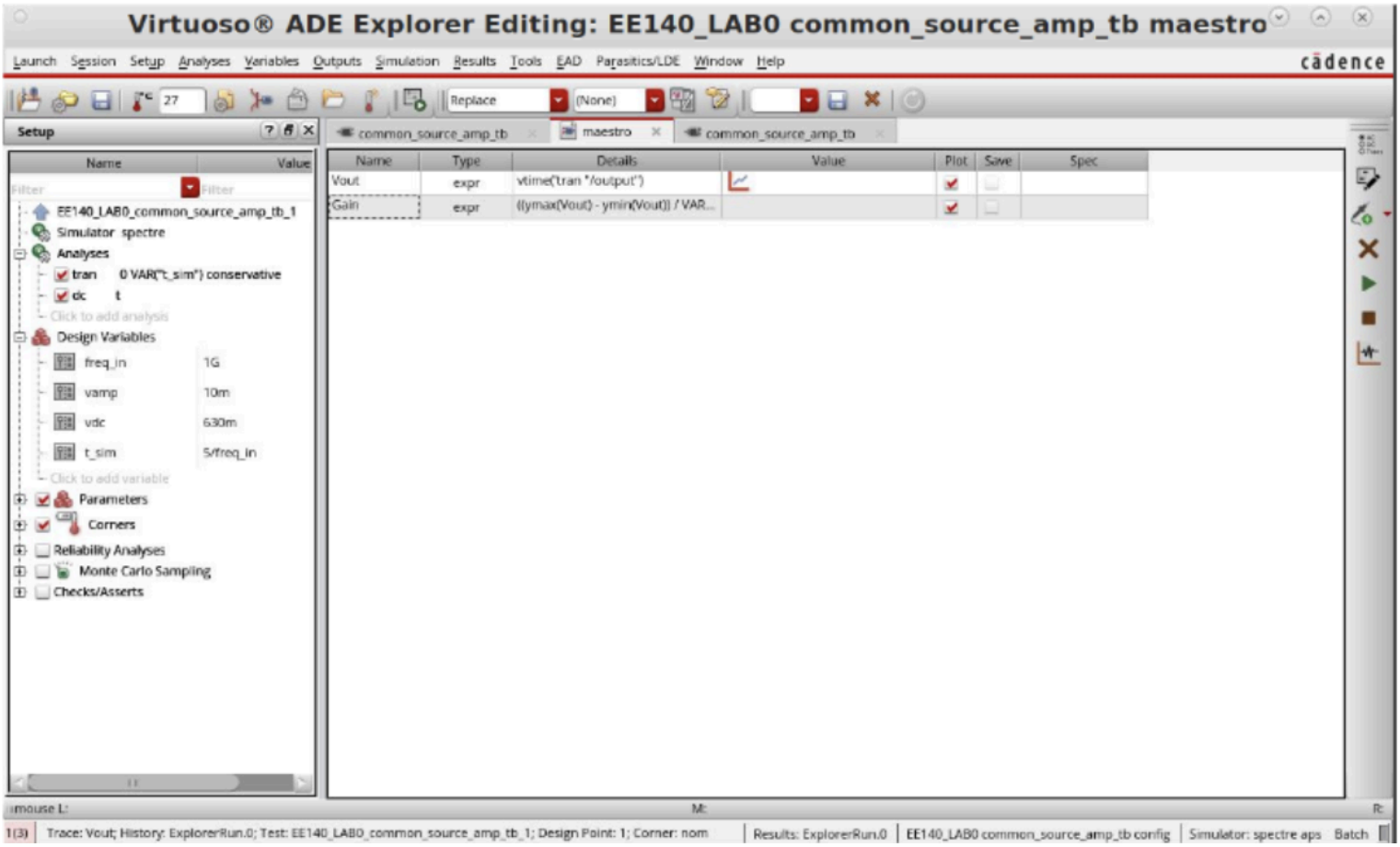
At this point, we can run the simulation. Press the  icon on the right side of the ADE Explorer window. Once the simulation has run, we can add signals to the plot. Go to Results -> Direct Plot -> Main Form to launch the form shown below. The tool should automatically pull up the schematic view, but if it does not, select the schematic window. Setup the form as shown below, and then select the “output” signal on your schematic to plot the output of the simulation. After selecting the output, if you go back to the ADE Explorer window, you should see an output added as shown on the right screenshot below.



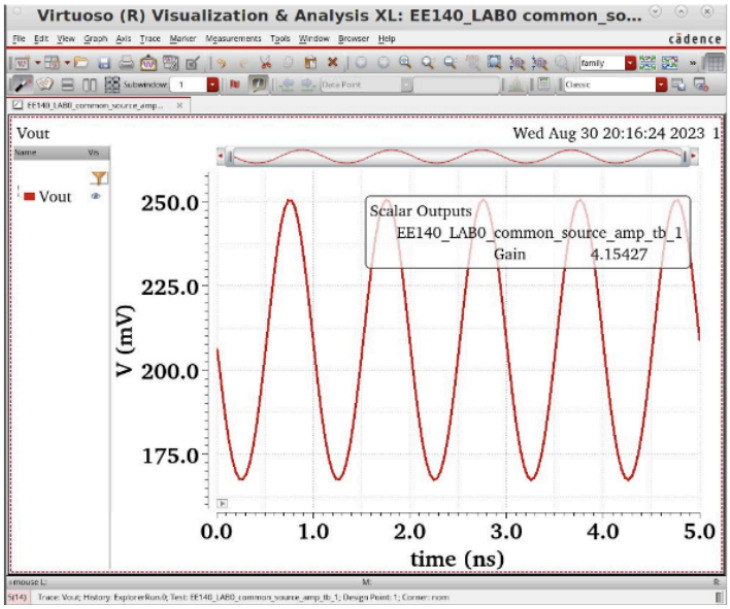


Let’s change the name of this output to make it easier to refer to this output. Double click on the name cell in the and change the name to “Vout”. One of the useful things about ADE Explorer is the ability to perform math on the outputs of simulations. Let’s measure the gain of the output. To do so, on the right select the  icon to add an output. Name this new output “Gain” and double click on the “Details” cell on this new output we are creating. Select the icon shown on the right of the “Details” cell to launch the output editing window as shown in the screenshot below.

This window is the ADE Explorer output editor. You can find the functions ADE explorer has and if you start typing a function into the above window you will see a pop up that describes the function. Add the following function “(ymax(Vout)-ymin(Vout))/(2 * VAR(“vamp”))” and click the green check at the top right of the window. Your final ADE Explorer window should look like this:



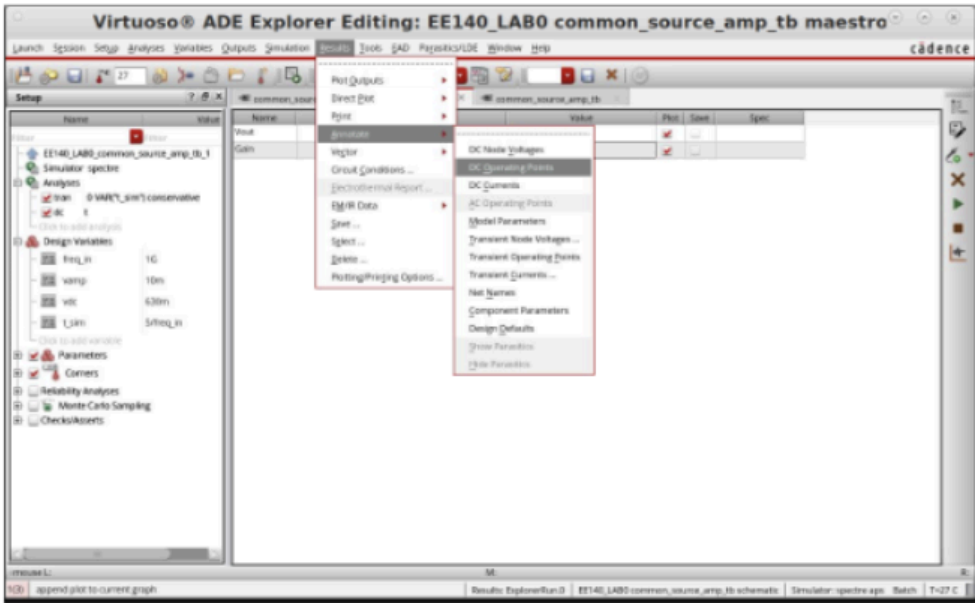
Now, to see our new outputs, we DO NOT have to re-run the simulation. Instead, we can hit the plot  icon on the right to plot the outputs of the simulation. You can double click on the title of the plot to give it a more useful title. Your final plot should look something like what is shown below (the gain in the image is not what you would get in your simulation):



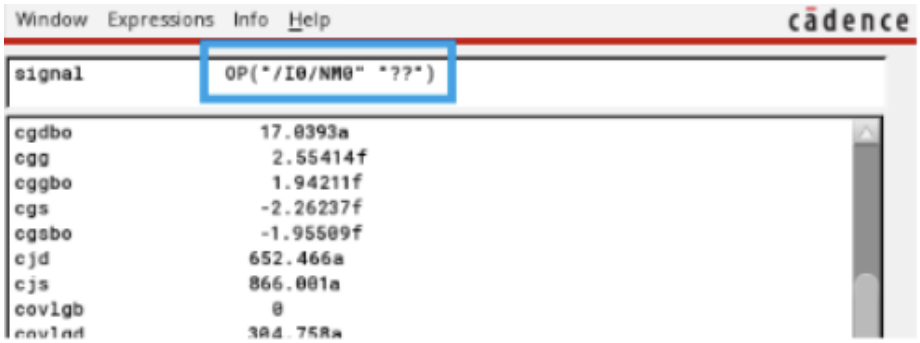
Change the trace to your favorite (legible) color, take a screenshot of the resulting plot and save it for the checkoff and the report. You can also export the data by right-clicking on the trace and selecting Send To -> Export..., and then use software like MATLAB or Python to create a cleaner plot (or Send To -> Table ->New Window and then copy and paste the data directly from the table).

Think about the transistor and how the input voltage affects the output voltage. At these operating points we've chosen, the transistor is in saturation. Why do transistors in saturation in this common source configuration yield a gain? This will be asked in the checkoff, and you will write an explanation of this in your lab report.

The Results menu of the ADE window provides more ways to see results. Click Results -> Annotate -> DC Node Voltages; the schematic should open. To descend into your amplifier's schematic, press 'e' and then click on the amplifier symbol. You should see the DC bias point of each node in the resistor and NFET. **Take a screenshot of your common_source_amp schematic with the operating point annotations, and save it for the checkoff and the report.**



We can also use ADE to compare the simulated gain with the gain we would expect using small-signal device parameters. Go back to the ADE window and select Results -> Print -> DC Operating Point. In the testbench, press "e" and select your amplifier symbol to descend into your amplifier. Then, click on the NMOS transistor in your amplifier to see the full list of parameters in this MOSFET model. Note the OP() expression at the top of the window:



The first argument is the device name ("/I0/NM0" above), and the second argument can be any of the parameters you see ("cdd," "cbb", etc.). This expression can be used directly in the calculator – for instance, "OP("/I0/NM0" "gm")" and "OP("/I0/NM0" "rout")" will give the NFET's small-signal transconductance and output resistance, respectively. **Derive a new expression that calculates the theoretical gain of the amplifier using NFET's small-signal parameters and the fixed 7.8 kΩ poly resistance similar to how you created the Gain expression. Attach your derivation and the theoretical gain value to your lab report.**

In addition, compare your simulated and theoretical gain and explain any discrepancies (if any). Please report these values in a table.

Finally, we will explore and show what happens when our “small-signal” is not in fact small. We will change vamp to a parameter sweep. Choose an upper limit such that the output waveform gets distorted (output is not sinusoidal). You can add a parameter sweep through the Design Variables panel. Double click your vamp value and select the three dots that appear. Click that, and then add a From/To sweep, choosing 10m as the “from”, a higher value of your choice as “to”. You’ll have to specify a “linear” step, then input a step size that is reasonable (something of about 5-10 steps total). Simulate your transient waveform again, and observe the distortion that occurs.

Attach an image of your parameter sweep plots, clearly showing an example of distortion of the sinusoidal output signal. In addition, explain why this is the case, and why we cannot classify the input signal as “small-signal.” Attach a plot showing the effects of the gain as your input amplitude increases.

Finally, save your testbench so you can bring up the results any time.

6. Deliverables and Grading

To get checked off, you must show the GSI (Matthew) the following:

- Circuit symbol
- Testbench
- Circuit schematic with the DC operating points
- Transient waveform of a small-signal input (vamp = 10m).
- Explain why a transistor in saturation gives gain at all.

In total, checkoff is 5pts and will be graded on a completed/not completed basis.

For the lab report, please include the following on a single pdf, submitted to Gradescope:

- (1pt) An image of your circuit schematic with the DC operating points.
- (1pt) An image of your circuit symbol.
- (1pt) An image of your testbench.
- (2pts) A derivation of the theoretical gain in terms of gm and rout, and an explanation of why biasing a transistor in saturation and putting it in a common source configuration gives gain at all.
- (1pt) A table comparing the theoretical gain and the simulated gain, and an explanation of any discrepancies between the two (if any).
- (1pt) Image of transient waveform of the small-signal input (vamp = 10m).
- (3pts) Image of transient waveform where the output waveform is clipped (distorted) and an explanation of why this is the case. Make a comparison to vamp = 10m in terms of the overall shape of the transient response. Please also report the vamp value(s) that was (were) chosen to show this distortion. Attach also the gain(s) associated with this (these) vamp value(s).
- (0pts, Optional) Feedback for the lab. What went well, what went bad, what was confusing, and what was helpful. This part is optional, but we would greatly appreciate it!

Lab 1 is due Feb 6th at 11:59pm. This includes being checked off and a submission of a lab report on Gradescope.

There are many opportunities for you to get checked off:

- During today's lab sections (Jan 30)
- During Matthew's Monday OH (5-6pm, Cory 400)
- During next week's lab section (Feb 6). We will be starting Lab 1 this week, so it would be preferable if you get checked off beforehand so you won't fall behind. The lab is due this day at midnight.

You will be graded on the following scale:

- You will get a 0 if you don't show up to the lab.
- +5 points for the checkoff
- +10 points for the lab report. You'll get full credit if you have all the items.