EE 140/240A Linear Integrated Circuits
Spring 2020 Lab 5

1 Introduction

For this lab, you may consult the professor, the TAs, your friends, the textbook, the internet, and any other living or inanimate objects, with the exception of your peers' lab reports. You may obtain data in pairs, but must **submit your own written report**. Be concise. Hand calculations should be to 1 or at most 2 digits of precision. **Don't use a calculator—I won't let you use one on the exam and it's good to get in practice.**

2 Lab: Folded Cascode Design

The goal is to design and simulate a PMOS input folded cascode for your programmable gain amplifier (PGA). The figures below show the topology circuitry of a PMOS input folded cascode (Is this topology reasonable in our technology? If not, how would you modify it? *Hint: Do you have the headroom for a stacked current mirror in the cascode?*). You also need to design the bias circuit to bias your amplifier. The constant-g_m circuit from Lab 4 is good candidate to start with. In any case, you might want to start with longer channel lengths (e.g. $L = 1\mu m$) to keep the devices in quadratic mode where they are easier to analyze. There are many ways to bias this circuit, this is only one of many possibilities.

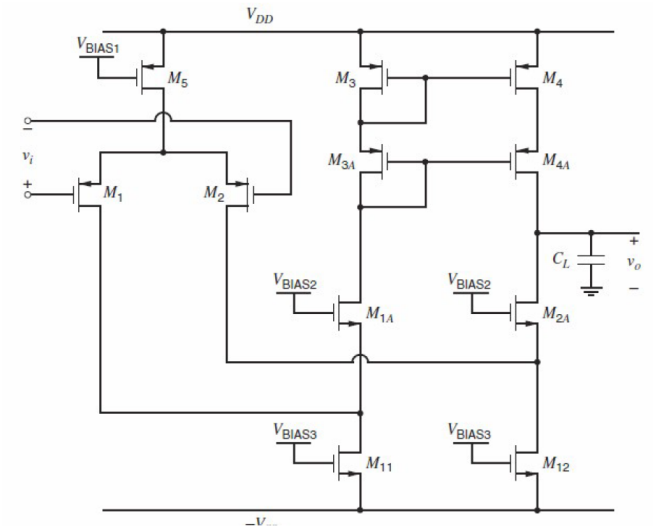

Figure 1: PMOS Input Folded Cascode

Specification	Requirement
V_{DD}	1.2V
Settling Accuracy ($f = \frac{1}{8}$)	$< 0.4\%$
Settling Time ($f = \frac{1}{8}$)	$< 5\mu s$
C_L	400fF
Input Common Mode Range	Includes ground
Temperature	$-40°C < T < +85°C$

At a high level, you'll be:

(1) Using hand calculation to design your amplifier based on the spec you're given.

(2) Verifying that your bias points are correct with a DC simulation. If simulated values are very far off from what you calculated by hand, **STOP**. Try to find the source of the discrepancy—if your DC bias is wrong, your amplifier will never behave the way you expect.

(3) Checking the AC parameters of your amplifier. Again, if simulated values are vastly different from your hand calculations, you should look for the source of the discrepancy.

(4) Verifying the stability of your amplifier in feedback.

2.1 Hand Calculation

(1) Calculate the requirements for open loop gain and unity gain bandwidth.

(2) Choose device lengths ($L = 1\mu m$ is a good place to start) and overdrive voltages.

(3) Calculate the transconductance required for the bandwidth and the current given your choice of overdrive voltage.

(4) Calculate widths for all transistors. Start with the current mirror for the input diff pair and work your way toward the output. The table below should give you a reasonable place to start with hand calculations using the quadratic model.

	$\lvert V_{th}\rvert$	$\lambda @ L = 1\mu m$	μC_{ox}
ne	0.65V	0.14 1/V	$140\mu A/V^2$
nel	0.35V	0.16 1/V	$150\mu A/V^2$
pe	0.65V	0.09 1/V	$30\mu A/V^2$
pel	0.30V	0.15 1/V	$60\mu A/V^2$

2.2 DC Simulation

(1) Embrace good hierarchy practices. Like your inverter for Lab 2, you'll want to have your amplifier in a cell of its own. We've provided a starter cell in library where you can populate the schematic. Feel free to modify the symbol as you like.

   a. Make sure you can see the `ee140_gsi` library in Cadence. If it isn't there, in your Library Manager click Edit → Edit Library Path. From there, Edit → Add Library → add library `ee140_gsi` in `/home/ff/ee140/sp20/cadence/ee140_gsi`. Afterward, press OK or File → Save to save the result in your cds.lib

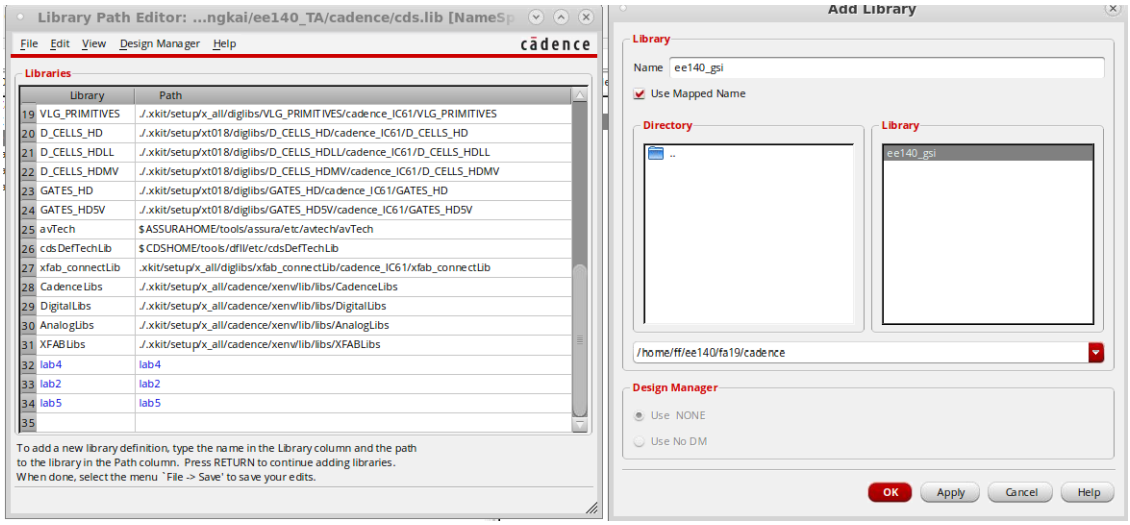

Figure 2: Library Manager

b. Copy `pmos_cascode` with right click + Copy in the Library Manager.

c. In the pop-up menu make sure to provide your library name under “To”. By default it would be `ee140_gsi`, which you need to replace with the name of your own library.

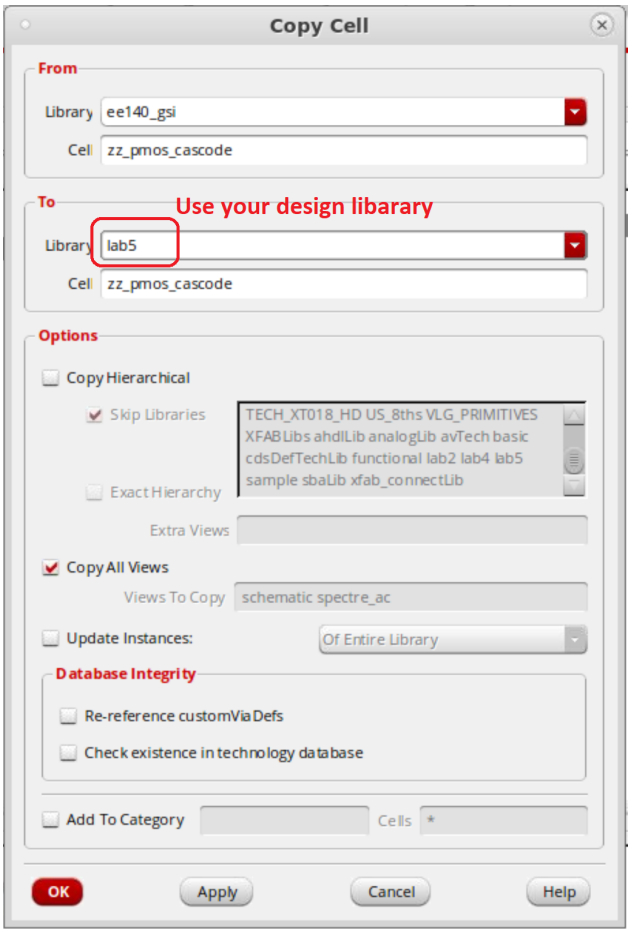

Figure 3: Copy Cell

d. If you get a window showing the Copy Problems, click on Overwrite All and then click on OK. This would finish copying the cell to your library.

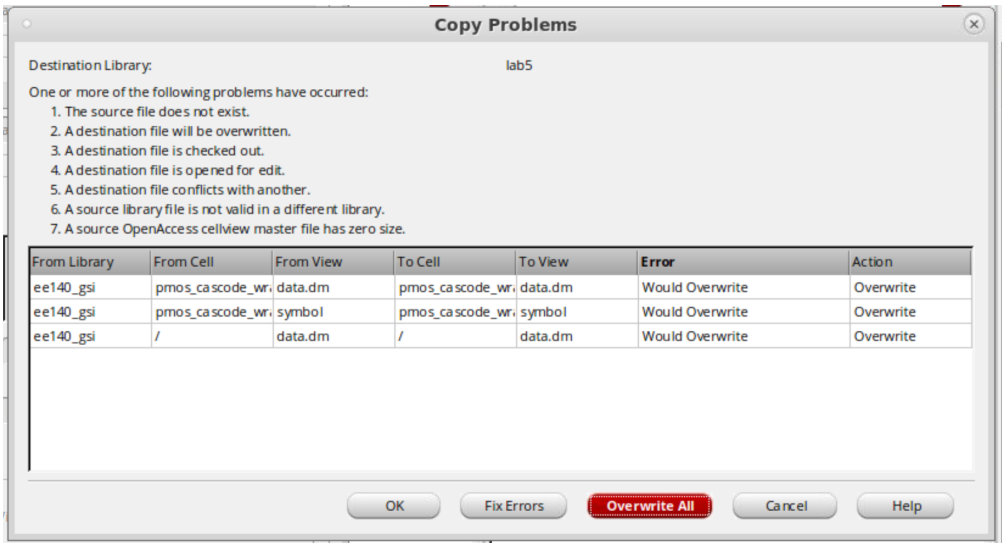

Figure 4: Copy Problems

(2) Update the schematic of the `pmos_cascode` you copied with your own design. Check and save.

(3) Copy `zz_pmos_cascode_openLoop` into your own library. We'll use this to characterize your amplifier in an open loop configuration.

(4) The amp in your copied test bench currently points to the cell in `ee140_gsi`. Replace the amplifier in `zz_pmos_cascode_openLoop` with the cell from your own library.

(5) We've provided a starter measurement setup for you in `zz_pmos_cascode/openLoop/spectre_state1`. Make sure the "dc" checkbox is ticked in the ADE window.

(6) Run the simulation.

(7) To have Cadence show you the DC node voltages on your schematic: Results → Annotate → DC Node Voltages

(8) To have Cadence show you the DC operating points in your schematic: Results → Annotate → DC Operating Points

(9) Ensure that your bias network's DC node voltages check out first, then check the main body of your amplifier next.

(10) If there is a large discrepancy between your DC simulation and hand calculation, **STOP**! Go through systematically to try and find the source of the discrepancy. If your DC bias is wrong, nothing else will work.

2.3 AC Simulation

(1) In the same testbench as your DC simulation (`zz_pmos_cascode_openLoop`), make sure the "ac" checkbox is ticked.

(2) Run the simulation.

(3) You should now see plots of your AC magnitude and phase, and we've also included bandwidth and unity gain frequency calculations (these are specialized functions in Cadence).

(4) Once again, if there's a large discrepancy in your AC simulation and hand calculation, try and find where the discrepancy is coming from. Chances are it's a result of biasing problems or a miscalculation of g_m, r_o, etc.

2.4 Stability Simulation

Stability analysis is a powerful tool that has more capabilities than standard AC analysis because it doesn't mask loading effects of parasitic elements like AC analysis can when calculating phase margin. It calculates the frequency response of the loop gain

$$\text{Loop Gain} = T = A_0 f$$

(1) Copy `zz_pmos_cascode_closedLoop` from the `ee140_gsi` library to your own.

(2) Make sure the amp in your copied testbench points to the cell in *your* library and not `ee140_gsi`. Change it if necessary.

(3) The variable "BETA" is the same thing as f (some textbooks use β instead of f).

(4) Make sure the "stb" checkbox is ticked.

(5) Run the simulation.

(6) Click Results → Direct Plot → Main Form. In the window that pops up, choose "stb" if it isn't already selected under "Analysis".

(7) To find phase margin, select "Phase Margin". This is how the TAs added the output `stb_PM` to your test bench.

(8) Once you have an estimate of phase margin, put your device in unity gain feedback and see if the response to a step input looks like you expect.

3 Deliverables

(1) Show how you calculated gain and bandwidth requirements.

(2) Schematic of opamp and bias generation with sizes and currents annotated. **Do not use Cadence schematics for submission**. You should either

- Draw neatly by hand
- Use some nice schematic drawing software (circuitikz in LaTeX, Adobe Illustrator, Digi-Key Scheme-It, etc.)

(3) Bode plots of amplifier frequency response (mag and phase) under different temperatures with DC gain and bandwidth annotated. **Do not use default Cadence plots for submission**. You should either

- Export and plot with Python, MATLAB, Excel, etc. with readable axis labels
- Modify your Cadence plot settings:
  - Graph → Properties → Set background to white
  - Graph → Properties → Graph Options → Font → Fixed [Sony] → Size 18
  - Right click on traces → Width → Extra Thick

(4) The following table:

Temperature	DC Gain	Unity Gain Bandwidth	Phase Margin @ $f = 1$
$-40°$C			
27°C			
85°C			