

EE140 Design problem 1 guidelines

1 Report guidelines

Your project report should include the following sections:

- Discussion of your topology choice: explain why you chose a certain topology and compare your topology with possible alternatives.
- Discussion of your FOM minimization strategy: explain how you chose all the W 's, L 's, V_{dsat} 's and currents; show your calculations.
- A schematic (drawn by hand) of your circuit, annotated with all the node voltages.
- Provide a table with W , L , V_{dsat} and I_{ds} for every transistor. For V_{dsat} and I_{ds} , include both the calculated and the simulated values.
- Provide a table with W , L and R for every resistor.
- Make a table with the simulation results from test benches and compare it with design specifications. Explain differences with your hand calculations.
- List FOM and the total area, the power consumption of your design.
- Comments and conclusion.

DO NOT INCLUDE SPICE NETLIST IN THE PROJECT REPORT.

2 Circuit submission guidelines

Send an email to ee140@cory.eecs.berkeley.edu with your name in the subject and your circuit in the body of the email. Make sure to send the email in plain text, not in html.

The body of the email should be the same as the file used to run the test bench:

```
.param VIC = < your midpoint of the common mode input range >  
.subckt amplifier inp inn out vdd vss  
< your circuit >  
.ends
```

The subcircuit should be named 'amplifier' and the order of the input nodes should be: positive input node, negative input node, output node, positive supply, negative supply. You can change the names of the nodes. Be certain, however that the 'inp' node is the POSITIVE input node and the 'inn' node is the NEGATIVE input node! If you swap these around, the testbenches (see below) will not work!

3 Information about testbenches

There are two testbenches available from the website: testbench1.sp and testbench2.sp .
To use them, do the following:

1. Make sure that your amplifier circuit is called `circuit.sp` . Both testbench netlists will include this subcircuit.
2. Make sure that `circuit.sp` , `ee140_models.sp` and your testbench files are all in the same directory.
3. type `'hspice -i testbench1.sp -o testbench1'` or `'hspice -i testbench2.sp -o testbench2'` and check the .lis files for your results

The testbenches perform 9 different tests. They are as follows:

1. Operating point and differential mode gain for $V_{IC}=\text{midpoint}$
 - a. Purpose: check $A_{DM} > 1000$
2. Operating point and differential mode gain for $V_{IC}=\text{midpoint}+0.25V$
 - a. Purpose: check $A_{DM} > 1000$ over common-mode variation
3. Operating point and differential mode gain for $V_{IC}=\text{midpoint}-0.25V$
 - a. Purpose: check $A_{DM} > 1000$ over common-mode variation
4. Operating point and common mode gain for $V_{IC}=\text{midpoint}$
 - a. Purpose: check $A_{CM} < 0.1$
5. Operating point and common mode gain for $V_{IC}=\text{midpoint}+0.25V$
 - a. Purpose: check $A_{CM} < 0.1$ over common-mode variation
6. Operating point and common mode gain for $V_{IC}=\text{midpoint}-0.25V$
 - a. Purpose: check $A_{CM} < 0.1$ over common-mode variation
7. AC analysis (differential mode gain) at 1Hz for $V_{OUT} = 0V$
8. AC analysis (differential mode gain) at 1Hz for $V_{OUT} = +0.3V$
 - a. Purpose: check $A_{DM} > 1000$ over output range
9. AC analysis (differential mode gain) at 1Hz for $V_{OUT} = -0.3V$
 - a. Purpose: check $A_{DM} > 1000$ over output range

testbench1.sp performs tests 1-6. testbench2.sp performs tests 7-9. testbench2.sp uses a simple feedback network to set the output voltage. An AC analysis at low-frequency is used instead of a .TF analysis, because the feedback network is active for DC analyses (including OP and TF), but not for AC analyses.

The outputs of these results can be found in the testbench<n>.lis file. You'll note that the .lis file will have lots of sets of outputs. This is because one complete set of outputs (operating point, transfer function analysis, etc) is generated for each test listed above.

If you look carefully at the testbench files, you'll see that it includes all the necessary input sources (signal, power) and output loads (R_L). Therefore, your subcircuit should NOT include these devices.

4 Automated testbench processing

Your TA's like to think of themselves as pretty nice people...the kind of people who will take the extra effort to make your life easier. (Perhaps some of you might think "less

hellish” is a more appropriate choice of words.) With that in mind, we’ve created some scripts for you to use to perform automated checking of your circuit performance against the specifications. To use it, do the following:

1. Grab `testbench_dp1.py` from the website. Put it in your design directory. Make sure the `testbench<n>.sp`, `model_ee140.sp`, and `circuit.sp` are in this same directory.
2. Once your circuit is fully functional (check against the testbench files mentioned above), type `'python testbench_dp1.py -r -e'` .
3. Observe the results on the screen.

Please note that this script is intended for final performance verification and is not intended for “design purposes”. That is, you shouldn’t plan on using this script to iterate to your final design. Trust us, it would take you way too much time to do and you wouldn’t be able to write insightful prose and analysis in your write-up.

As a final note, this script has been debugged, so if it doesn’t work for you, chances are that YOU are doing something wrong. Before coming to the TA’s to complain, try downloading the example `circuit.sp` from the website and running the script with that.

5 Important warnings and caveats

1. Do not have ANY voltage or current sources in your subcircuit definition!! (Resistors and transistors only.) The power supply voltages will be supplied by the testbench netlist, which will also instantiate your circuit netlist. Including a voltage or current source in your subcircuit will result in a significant grade deduction.
2. All W’s and L’s *must* be in increments of 0.01um. (Therefore `w=1.04um` is acceptable. `w=1.04835um` is not.) You should be using hand analysis, not subtle device tweaking, to meet your design specifications.