

## 1 Report guidelines (total 4 pages)

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. Be sure to discuss your compensation/stabilization strategy and settling time minimization strategy.
- Show your hand calculations for dominant and first non-dominant poles, slew rate, and DC gain, and loaded loop gain. You can use small transistor signal parameters (gm, caps, etc.) obtained from SPICE.
- Discussion of how your circuit behavior is affected by process variation and which specifications are affected by which process corners. (i.e. is your circuit speed worse in the fast corner or slow corner? Is your circuit DC gain worse in the fast corner or slow corner?) Discuss how your design methodology takes this process variation into account.
- A schematic (drawn by hand) of your circuit, annotated with all the node voltages.
- Provide a table with W, L, and the simulated  $V_{dsat}$  and  $I_{ds}$  for every transistor (typical corner). Also include any compensation cap values..
- Include simulated closed-loop time-domain waveforms of the output voltage transient for the following cases: typical corner with positive input step, slow corner with negative input step.
- Include a table of the following performance specs for each process corner: final settling error, settling time (to 0.02% accuracy), current consumption, DC common-mode gain.

Note:

Current consumption is measured from the positive supply.

**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:

**.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

POSTIVE 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 4 testbenches available from the website. To use them, do the following:

1. Make sure that your amplifier circuit is called `circuit.sp`.
2. Make sure that `circuit.sp`, `ee140_models.lib`, `parameters.sp` and your testbench files are all in the same directory.
3. type `'hspice -i testbench1.sp -o testbench1'` and check the `.lis` files for your results. (same for other testbenches)

#### Brief description of each testbench:

1. **Testbench1.sp:** Closed-loop time-domain response. Sweeps over process and input voltage step polarity. Measures current consumption, final settling error, and settling time. The transient is currently setup to run for 100ns, so if your circuit takes more than 100nS to settle, you may need to modify this spice file.
2. **Testbench2.sp:** Closed loop frequency response. Sweeps over process and measures closed loop DC gain and 3db freq.
3. **Testbench3.sp:** Open loop differential-mode frequency response. Sweeps over process and measures DC gain, unity gain freq, phase margin, and gain margin.
4. **Testbench4.sp:** Open loop common-mode frequency response. Sweeps over process and measures DC common mode gain.

### 4 Automated testbench processing

As usual, we've got some automated test scripts for you. To use:

1. Grab `testbench.py` from the website. Put it in your design 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.

**IMPORTANT CAVEAT:** Although we made every effort to debug these testbenches, we're not perfect. Please note that these testbenches may be somewhat buggy and may give erroneous information. (We don't think that'll happen, but we can't guarantee anything.) If you see something fishy in your simulation results, **USE YOUR OWN JUDGEMENT** to determine if the problem is from our testbenches or something in your circuit. If you find an error in our testbenches, please let us know. Anyone pointing out an actual bug in our testbenches will receive some measure of extra credit on the project. (However, in order to limit the number of "false alarms", you will be docked a few points if you send us too many incorrect "bug reports". Again, use your judgement....)