

UNIVERSITY OF CALIFORNIA
College of Engineering
Department of Electrical Engineering and Computer Sciences
Last modified on September 3, 2004 by Stanley Wang and Henry Jen
[*\(sbtwang@eecs.berkeley.edu\)*](mailto:sbtwang@eecs.berkeley.edu)

Borivoje Nikolic

Lab #3: CMOS Inverters and NAND Gates

EECS 141

1. Objective

There are two primary objectives in this lab. The first objective is to become familiarized with the Cadence Virtuoso Schematic Composer, which is the schematic editor being used in this course. The second is to gain experience in laying out some basic building blocks.

Before the lab begins however, it is important that the following question be addressed:

I know how to do layout now. Why do I have to learn another tool (Schematic Composer)?

Before answering this question, one must first understand that a schematic is a symbolic representation of your circuit. It creates a level of abstraction by grouping basic circuit elements (i.e. transistors, capacitors, etc.) into symbols. By doing this, the schematic simplifies the circuit representation into a network of easily recognizable elements by abstracting away from the layout implementation.

As the course progresses, larger and more complex designs than your simple inverter or 2-input NAND gate will be necessary. When the time comes for layout, and you find yourself looking at several hundred transistors in a jumble of polygons and colors from which no sense can be made (maybe some), it is nice to turn back and utilize the schematic as a guide towards design and verification.

Focusing back on Virtuoso Schematic Composer, it is a tool that allows you to create schematics. It has some important and useful features that could prove beneficial later on in the course during debugging such as *cross-probing* and *layout-vs-schematic checking (LVS)*. Cross-probing in Virtuoso allows you to select a net (a wire with some label/name) in your schematic and see the corresponding net in your layout. Another feature in the Schematic Composer is hierarchy and abstraction. An example of this would be if you wanted to represent an array of 10 full adder blocks. Instead of having a large layout of polygons, you could create a symbol for a single adder and tile 10 of those together. LVS is a function using algorithms to traverse your layout and schematic and ensure a one-to-one correspondence. Should LVS find mismatches, it will provide output files leading you to your problem.

2. Tasks

- a.
 - Create a working directory for this lab (eg. `~/ee141/lab3/`).
 - Create a library "inverter" linked to the TSMC 0.24um technology (see lab2).
 - Create a layout cellview "inv1".
 - Now go to the directory `~/ee141/lab3/inverter/inv1/layout/` and type `cp ~/ee141/lab3/layout/* .` (note the dot (.) after the star (*) symbol). Reply "y" to all "overwrite" prompts.
 - Go back to Cadence and reopen the layout view of inv1. You should see the template for your inverter.

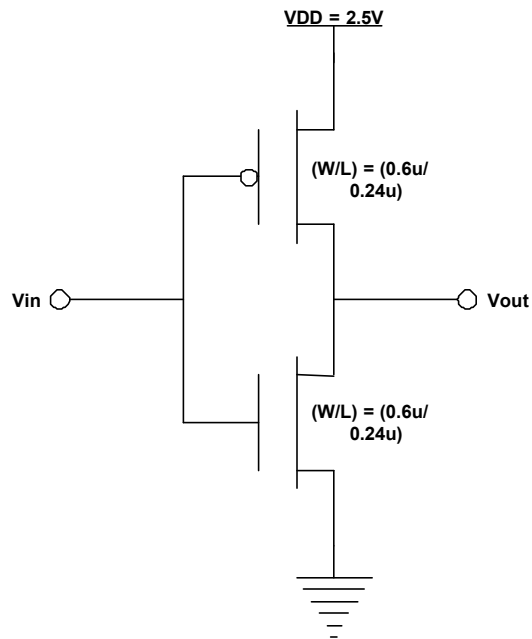
Now you are ready to proceed with the rest of the lab.

- b. Read the section about Layout Design Rules, which can be found in chapter 2 of the textbook, or at the following link:

(http://bwrc.eecs.berkeley.edu/Classes/ICDesign/EE141_f01/Notes/chapter2.pdf)

Keep in mind that all layers have a minimum length, and some layers have enclosure and distance minimums.

- c. Layout a CMOS inverter using Virtuoso Layout Editor. The circuit diagram of the inverter is shown in the schematic figure below. Utilize the W/L indicated in the diagram. *Please state in your report whether or not you think these sizes are reasonable.*



Complete the layout inverter using Virtuoso Layout Editor. Use Create->Pin (Shape pin) to label the nodes. Label the input V_{IN} (using type “input”) and the output V_{OUT} (using type “output”). Also add V_{DD} and GND using an “inputOutput” pin type.

Be sure to add the nwell (the pwell being implicit in this process), and several nwell (ntap) and pwell (ptap) contacts. The well contacts can be instantiated from the NCSU_TechLib_tsmc03d library. Complete the layout and eliminate any DRC errors. Save your results.

- d. Next, layout an inverter twice the size of the previous one (NMOS and PMOS $W/L=1.2/0.24$). All transistor gates should be oriented vertically. Remember to add the nwell, and nwell and pwell contacts. Eliminate any DRC errors. (Hint: if you can't get the transistors to fit, think about the structure of the NAND layout you did in lab2). Save your results.
- e. Go through the Schematic tutorial and become familiarized with Virtuoso's interface as well as cross-probing.
- f. Generate an inverter schematic using Virtuoso Schematic Editor. Create this schematic using NMOS and PMOS transistors, as well as VDD, GND, and the Input and Output nets, which you will label VIN and VOUT respectively.
- g. Use the cross-probing features shown in the tutorial to verify that VIN and VOUT in your first layout correspond with those in your schematic.

3. Report

For your report, please hand in the following:

- Printout of your inverter schematic from Virtuoso Schematic Composer.
- Printouts of both of your inverter layouts from Virtuoso Layout Editor – please label all nodes, annotate transistor sizes.
- Provide hand calculations of t_{pLH} , t_{pHL} , and t_p
 Use the following parameters: $V_{tn} = 0.43V$; $V_{tp} = -0.43V$;
 $V_{DSATn} = 0.63V$; $V_{DSATp} = -1.0V$;
 $k_p = 115 \mu A/V^2$; $k_n = -30 \mu A/V^2$