

**UNIVERSITY OF CALIFORNIA**  
**College of Engineering**  
**Department of Electrical Engineering and Computer Sciences**  
*Last modified on September 17, 2001 by Joshua Garrett ([joshuag@eecs.berkeley.edu](mailto:joshuag@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 SUE, 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 (SUE)?*

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 SUE, 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 SUE allows you to select a net (a wire with some label/name) in your schematic and see the corresponding net in your MAX layout. Another feature in MAX 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. It is also possible to use the schematics from SUE to document certain characteristics of your circuit such as critical path. This can be very useful come crunch time.

## 2. Tasks

- a. Copy the files necessary for this lab (inverter\_template.max and MyNAND.sue) into your account by executing the following at your UNIX prompt:

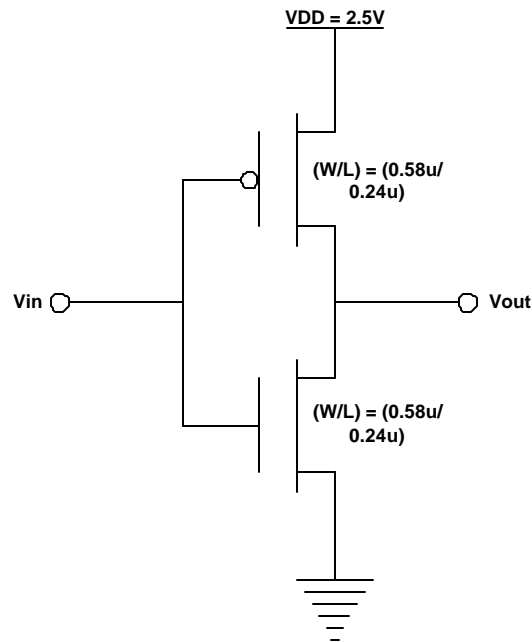
```
> cp ~ee141/LAB3/* .
```

- 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\\_f00/Notes/chapter2.pdf](http://bwrc.eecs.berkeley.edu/Classes/ICDesign/EE141_f00/Notes/chapter2.pdf)

Keep in mind that all layers have a minimum length, and some layers have enclosure and distance minimums. Take advantage of the real-time DRC checking by looking for the small white dots that represent DRC errors.

- c. Layout a CMOS inverter using MAX, and using the provided template file **inverter\_template.max** as a starting point. 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.*



Edit and complete the inverter using MAX. Label the input VIN (using type “input”) and the output VOUT (using type “output”). Also add VDD and GND using a “global” net.

Add material only within the prboundary (the shaded region). Be sure to add the nwell (the pwell being implicit in this process), and several nwell (nwc) and pwell (pwc) contacts. The well contacts can be created easily from the via gcells. Complete the layout and eliminate any DRC errors. Turn on the grid (**g**) and print out your results.

\*Note: The contacts are allowed outside the boundary, but all others are not.

- d. Next, layout an inverter twice the size of the previous one (NMOS and PMOS  $W/L=1.16/0.24$ ), again using the provided template file, and again adding material only within the prboundary. 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). Turn on the grid and print out your results.
- e. Go through the rest of the MAX tutorial and become familiarized with SUE's interface as well as cross-probing. The file MyNAND.sue is available in the **~ee141/LAB3** directory.

- e. Generate an inverter schematic using SUE. 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.
- f. Use the cross-probing features shown in the MAX 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 SUE.
- Printouts of both of your inverter layouts from MAX – please label all nodes, annotate transistor sizes, and **display the grid!** (no messing with the prboundary)
- Provide hand calculations of  $t_{pLH}$ ,  $t_{pHL}$ , and  $t_p$   
Use the following parameters:  $V_{tn} = 0.43V$ ;  $V_{tp} = -0.43V$ ;  
 $V_{DSATn} = 0.63V$ ;  $V_{DSATp} = -1.0V$ ;  
 $k_p = 115 \mu A/V^2$ ;  $k_n = -30 \mu A/V^2$