Homework 1:
Introduction to CS/EE 181 and `magic`

Homework is due in class on the date indicated above. Late homeworks will be penalized by weighting errors on those problems that were completed past the due date by the number of days past the due date plus an extra 5% per day. Please indicate which problems, if any, took extra time.

By the end of this assignment, you should

- Feel comfortable interacting with the UNIX operating system.
- Know how to design simple structures with `magic`.
- Be able to verify the operation of simple combinational CMOS circuits with the `irsim` switch-level simulator.

1 UNIX tutorials (optional)

0%

For this assignment, you will need to use the CS cluster machines in Annenberg 104.[1] If you have not already done so, get an account on these systems by filling out an account request form online at:

`http://acctreq.cms.caltech.edu/cgi-bin/request.cgi`

If you are unfamiliar with the use of the UNIX operating system, go ahead and study the UNIX tutorials on the CS181 web page:

`http://www.async.caltech.edu/~cs181/`

2 `magic` tutorials

0%

You do not have to turn anything in for this part of the assignment, but you will appreciate having learned the mechanics of `magic` for the next part. Once you log in on one of the CS cluster machines, you will need to perform some set up to get the tools working. Open a terminal and enter: [2]

```
% /cs/courses/cs181/setup.sh
```

then close the terminal. All future terminals you open will now be properly set up to use the tools. Now, start up `magic` by opening a terminal and typing:

```
% magic tut1
```

[1] Using the 104 machines is highly recommended. However, the tools for this assignment are freely available at `http://opencircuitdesign.com`. A CS account is still required to submit the assignment.

[2] In the examples, the `%`-sign is used to indicate the UNIX command-interpreter prompt, which may be different for your shell. You should not type it.

at a UNIX prompt. You should get a `magic` window with some colored polygons in it. You're all set! (If not, contact a TA!)

Next you need to find a `magic` manual somewhere. There may be hardcopies in the lab [3], and there is an online copy on the CS181 webpage, but these are both a little out of date. The most recent version of the documentation is at

`http://opencircuitdesign.com/magic/magic_docs.html`

Once you have a manual, work through the tutorials. Start with tutorial 1, *"Getting Started."* When you think you have mastered the material in tutorial 1, continue with tutorials 2 *"Basic Painting and Selection,"* 4 *"Cell Hierarchies,"* 6 *"Design-Rule Checking,"* and 8 *"Circuit Extraction,"* in that order. Ignore the other tutorials for now—they cover topics that are inessential to CS181. Tutorials 1, 2, 4, 6, 8 cover all you need to know this term, but if you are interested in raising your productivity (e.g., if you have a lot of layout to do for your term project), then take a look at the remaining tutorials. Note however that even the most recent manual is not really up to date with magic itself, so not all of the additional tutorials will actually work.

3 Your first layout

100%

Be certain to read the entire assignment before starting. If anything remains unclear, go ahead and ask a TA for help.

If a problem set is entirely optional, chances are most people will not bother to do it. Since we would not want that to happen, you are going to be required to do some simple layout to hand in. The process is rather involved, but bear with us—it is not really difficult. Above all, do not be alarmed if there is something you do not understand, or if you feel like you have no idea what is going on. Almost everyone feels like that when he is first confronted with doing layout. Ask a TA for help!

i. Create a subdirectory in your home directory called "`cs181`" (use the UNIX `mkdir` command to do this). Create directories called `cs181/181a`, `cs181/181a/lab` and finally `cs181/181a/lab/lab1`. This will be your working directory for the rest of this lab. (If for some reason you feel you need to create the top-level `cs181` directory somewhere other than in your home directory, this can be accomodated, but please let the TAs know where you want to put it. We would appreciate, however, that you at least maintain the suggested internal structure below `cs181`.)

ii. By hand on a piece of paper, or otherwise, draw the transistor diagram for a three-input NAND gate. For this assignment only, consider the MOSFET as a four-terminal device; i.e., include the substrate connection in your transistor diagram. Remember to put your name on the transistor diagram.

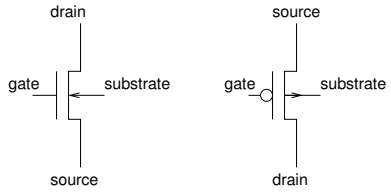

Figure 1: Suggested transistor symbols for nFET and pFET, respectively, for this assignment.

iii. Implement a three-input NAND gate in restoring CMOS logic (no pass gates). A three-input NAND gate computes the boolean function

$$nand3 = \neg(a \wedge b \wedge c)$$

[3] It is part of a book titled *"1986 VLSI Tools: Still More Works by the Original Artists."*

and is a simple generalization of a two-input NAND gate. Label the inputs `a`, `b`, and `c`. Also label the output `nand` and the power supply rails `Vdd` and `GND`. [4] Remember to plug both wells (i.e., tie them to the rails using the proper well contacts, `psubstratepcontact/ppc` for the p-well bloated around active n-diffusion and `nsubstratencontact/nnc` for the n-well bloated around p-diffusion)!

Save the design by clicking on `File->Save layout` in the layout window. Please ensure that you use the file name `nand3.mag`.

iv. Check that your design operates correctly with the `irsim` switch-level simulator. Here is how you do this:

Extract the electrical parameters of your design with the `magic` extractor. Use the magic command "`:extract`".

Next, convert this to a file format that `irsim` understands with the `ext2sim` tool. This can be run from the command line (`% ext2sim nand3`), or from within `magic` ("`:ext2sim`").

Finally, run irsim. This also can be done from the command line (`% irsim nand3`), or from within `magic` ("`:irsim`").

Now you are ready to see if your design behaves as intended!

```
%w nand
%s
nand=X
time = 10.000ns
%vector in a b c
%w in
%s
in=XXX nand=X
time = 20.000ns
%l a
%s
in=0XX nand=1
time = 30.000ns
%h a
%s
in=1XX nand=X
time = 40.000ns
%setvector in 111
%s
in=111 nand=0
time = 50.000ns
```

After you have verified that your design behaves as a three-input NAND gate should, you are done with the assignment. Please leave the `nand3.sim` and `nand3.mag` files in the directory so that the TAs can find them. Turn in a sheet indicating where the TAs may find the files you generated. Make sure one of the files is a transcript of your `irsim` session (you can generate this with `File->Save...->All` in the irsim window). Do not forget to put your name on your homework—it will not be considered turned in until you have turned in the filenames on paper.

Keep these guidelines in mind as you design the circuit:

1. Try to keep the design as small as possible without violating any design rules.
2. Try to follow the general layout guidelines in the *CS181 Guidelines for Reasonable Layout.*

[4] The tutorials recommend using the labels `Vdd!` and `GND!`. Please ignore this.

3. For this assignment, draw all your transistors with a width of four λ and a length (i.e., polysilicon width) of two λ. Remember λ is just the unit size (a.k.a one square is 1λ x 1λ).

4. There should not be any extract warnings when you type the `:extract` command in `magic`; if there are, it means you have not yet finished the layout: you still have to finish wiring some nets.

In Case of Difficulty

1. If you get lots of warnings from `irsim`, make sure your nodes are wired correctly and your power supplies are connected and properly named (same case).

You may find it instructive to read Peter Hofstee's three-chapter introduction to CMOS, which you can reach from the CS181 homepage `http://www.async.caltech.edu/~cs181`.