UNIVERSITY OF BRITISH COLUMBIA

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

EECE 479: Introduction to VLSI Systems
Fall 2007

Assignment 2: Layout with Magic

Due: October 4th , 2007 before 11:59pm

Work on this assignment individually (one handin per person)

In this lab, you will become familiar with the layout capabilities of Magic. Warning: this assignment may take longer than you expect, so start early. Although it may seem like a lot of work, you will be able to *directly* use the result from this assignment as a building block for your project. You will only hand in Task 4.

Task 1: Set up your account:

You can run Magic on any of the undergrad Linux machines. You will need to access these machines remotely, either through one of the PCs in the department (such as in room MCLD 303), or else at home. Your first task is to make sure you can run Magic.

To do this assignment, you need an account on the ECE linux machines; you already have this (if not, or if you don't know your id and password, send an email to help@ece.ubc.ca and tell them you are taking this course). Although you will be running Magic on one of the department Linux servers, you will not actually be sitting in front of the Linux machine. Instead, you will access the Linux machine through one of the Windows computers within the department (such as MCLD 303), or from home. If you are running this in MCLD 303 (recommended), do the following. On the local machine in MCLD 303, log in, and run two programs:

1. First run X-Win32 (from the Start menu). This is an X-server program which will receive drawing commands from the remote server (the Linux machine) and use these commands to draw windows on your screen in MCLD 303.

2. Run the SSH tool (also from the Start menu in the computers in MCLD 303). The remote machine you want to connect to is called ssh-linux.ece.ubc.ca, and you can use your ECE id and password. If all goes well, SSH will start a terminal window. Commands you type in this window will be executed on the Linux machine.

In this terminal window, you can set up your environment by typing the following command:

```
source /CMC/cad/bin/magic7.csh
```

You then *might* have to set up your display variable, to tell the linux machine where to display your windows. The name of your local machine is on a yellow sticker on the computer itself. As an example, if your local machine is called "castlegar.ece.ubc.ca", then enter the following command to set up your display variable:

```
setenv DISPLAY castlegar.ece.ubc.ca:0.0
```

Then, you can run magic, by simply typing

```
magic
```

If this opens a new drawing window on your screen, then all is well, and you can proceed to Task 2.

If you are going to try to run this at home, see
http://help.ece.ubc.ca/HowToDisplayXApplicationsOnWindows.shtml
for details of how to set up your environment. At the bottom of the page, several alternative x-servers are identified; the Microimages server does not work with Magic, but the other one seems ok. Note that there is no way that the TAs can help you debug your home computer set-up. If you are unable to set-up your computer as described in the web page, then you should consider using the machines in MCLD 303.

Task 2: Magic tutorial

1. The Magic tutorial can be found on the course web site at http://www.ece.ubc.ca/~eece479/magic.pdf.

2. Work through tutorials 1 through 4. You can find the example .mag files in /CMC/tools/magic/tutorial. You may want to copy these files to your own directory so you can modify them (type

   cp -r /CMC/tools/magic/tutorial .

   from the linux command line. Note the space followed by the "." at the end of that command).

   Note: when running Magic, use the command: "magic tut1" instead of "magic tut1.mag". The latter won't find the input file, since it will be looking for tut1.mag.mag. In the past, many many students have spent a long time trying to debug their setup, when this was the simple cause of their problems.

Task 3: Magic Mystery Cell

Download the file http://www.ece.ubc.ca/~eece479/mystery.mag to your home directory. Examine this layout, and do the following:

a) Draw a transistor-level diagram of the cell
b) What is the logic function implemented by the cell?

Hint: type ":see no nwell" and ":see no pwell" to make the layout easier to see. Also, the use of the "s" and "S" commands will let you examine connectivity quickly.

Task 4: Layout using Magic

In this task, you are going to lay out a Full Adder cell. A Full-Adder cell will be required for your project, so if you do a good job here, you will be able to re-use your layout directly.

Recall from EECE 353 that a full adder has three inputs: A, B, Cin, and has two outputs, S and Cout. The equations for each output can be written as follows:

S = A xor B xor Cin
Cout = (A and B) or (Cin and (A or B))

Do the following:

a) Draw a transistor-level schematic of a CMOS circuit that implements the function. Hint: for the Sum output, you may want to use some of the techniques from Slide Set 3. You don't want degraded outputs, so use full transmission gates rather than pass transistors. For the Cout signal, I would recommend implementing the inverse of Cout, and then inverting the output. If you do this, then you would implement:

$$\overline{Cout} = \overline{(A \text{ and } B) \text{ or } (Cin \text{ and } (A \text{ or } B)}\ )$$

From this, the pull-down and pull-up network can be easily constructed. Since this constructs the inverse of Cout, you will want to follow this with an inverter. There are other ways to construct a full-adder (see Assignment 1, for example). Feel free to implement this any way you like, as long as the overall behaviour is that of a full-adder, and there are no degraded outputs.

b) Draw a sticks diagram that gives a rough estimate of how the cell could be laid out. The inputs should appear at the top of the cell, and the outputs should appear at the bottom of the cell (this will make the cell fit into your project easily). The VDD connection should consist of a rail that spans the entire cell width, across the top of the cell. The GND connection should consist of a rail that spans the entire cell width, across the bottom of the cell.

c) Using Magic, lay out the cell. Your cell must pass all design rules. Marks will be deducted for cells which are overly large. Hint: try hard to share source/diffusion areas between transistors. Label all inputs and outputs on your layout clearly.

d) The naming convention is as follows:

- The inputs should be named **A**, **B**, and **Cin**.
- The outputs should be named **S** and **Cout**.
- Despite what the tutorial says, the power rail should be called Vdd and the ground rail should be called GND. (no exclamation marks).

Any other naming convention will cause the automated layout checker to give your circuit a score of ***zero****.*

<u>What to hand in:</u>

Tasks 1-3 will not be marked. Your mark on the assignment will be based on your solution to Task 4c. For Task 4c, you will hand in your file electronically. We will do much of the marking electronically. For this to work, you must follow the following instructions exactly (if you don't, and we have to mark your assignment by hand, we will be REALLY REALLY TOUGH MARKERS).

1. Make a directory under your home directory called **eece479** (note: not elec479, e479, or anything else).
2. Make a subdirectory under your eece479 directory called **ass2** (note: not assgn2, assn2, etc.)
3. Put your .mag file in the ass2 directory, and call it **function.mag**
4. Create another file in your ass2 directory called **info.txt**. This file should contain *three* lines. The first line contains your name (last name first). The second line contains your student number. The third line contains your email address (this is the email address to which your mark will be sent, so be sure it is correct).
5. The ass2 directory should contain ***only*** the files **function.mag** and **info.txt**.
6. Enter the following to hand in the file:

handin eece479 ass2