UNIVERSITY OF BRITISH COLUMBIA
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

EECE 479: Introduction to VLSI Systems
Fall 2007

Assignment 3: IRSIM

Due: October 12, 2007 before 11:59pm
Work on this assignment individually (one handin per person)

In this assignment, you will perform several tasks involving IRSIM simulation. As part of this assignment, you will learn to use IRSIM very well. In the project, you will use IRSIM extensively to help you debug, so the more familiar you get with the tool now, the better. You will only hand in Tasks 2 to 4, and will do so on-line using the "handin" program.

Task 1: Testing your circuit from Assignment 2:

In this task, you are to use IRSIM to test the functionality and speed of the circuit you designed in Assignment 2 (if you didn’t get a working layout, you can borrow a friend’s layout, although you should extract and simulate it yourself). You will not hand in anything from this task, but I encourage you to do it carefully to make sure you know how to use the tool.

To do this:

1. Set up your environment using the same command as in Assignment 2:
   source /CMC/cad/bin/magic7.csh

2. Read the IRSIM tutorial on the course web site at **http://www.ece.ubc.ca/~elec479/irsim_tut.pdf**. You may also find the last two parts of the Magic tutorial useful.

3. Start Magic, and open your Assignment 2 cell. In Magic, when your layout is loaded, type **:extract all** This creates a .ext file. If there are any errors or warnings at this point, deal with them.

4. Convert your .ext file to a .sim file by typing (in the shell, not in Magic):
   ext2sim <filename>

5. Then, run irsim on your file using
   irsim scmos50.prm <filename.sim>

   If you get an error that says the tool can not find the **scmos50.prm** file, you can get a copy from **http://www.ece.ubc.ca/~elec479/scmos50.prm**. Put the file in your working directory.

6. Use the commands illustrated in the tutorial to thoroughly test your circuit. Use the “path” command to find the delay of your circuit for several different possible input transitions.

7. Looking at your layout, attempt to figure out which input transitions would cause the worst-case delay. Using IRSIM, find the delay of this path. Is this actually the worst case delay? Does the delay depend on the history of input vectors you have applied? Why do you think this is? (don't hand in the answers to these questions, but think about them; they would make good midterm questions ☺).

From this task, you will hand nothing in.

(continued on next page…)

Task 2: Simulating an 8-bit adder

In the project, many of your simulations will require asserting and monitoring buses. While it would be possible to monitor each bit individually, this can be cumbersome and error-prone. IRSIM has facilities to set values on buses and monitor them. In this task, you will become familiar with these features.

1. Create a new layout: an 8-bit ripple adder. To do this, you will combine 8 copies of your cell from Assignment 2 (there are several ways to do this, see the Magic tutorial). The input bits should be called **A_0** to **A_7** and **B_0** to **B_7** (the underscore is important). The carry-in should be called **Cin** and the carry-out should be called **Cout**. The output sum should be called **S_0** to **S_7.** Note that you will be using a full-adder in your project, and you should be able to re-use what you create in this task. Call your file **add8.mag**

2. Extract the layout to a .ext file and covert it to a .sim file as described in Task 1. The sim file should be called **add8.sim**

3. The IRSIM tutorial describes how to define a bus using the vector command. Using this command, create vectors for the 8-bit inputs **A** and **B**, and the output **S.**

4. Simulate the adder to convince yourself it works properly.

From this task, you *may* wish to automate your testing using a .cmd file. A .cmd file is simply a list of IRSIM commands. To create a .cmd file, enter the commands in a file (for example task2.cmd) and then simulate using:

```
irsim scmos50.prm <filename.sim> -commandfile.cmd
```

(the dash before the command file name is important).

For this task, please hand your .sim file (you do not need to hand in your .cmd file… we will use our own to test your adder, so please make sure you follow the naming conventions above). You also do not need to hand in the .mag file.

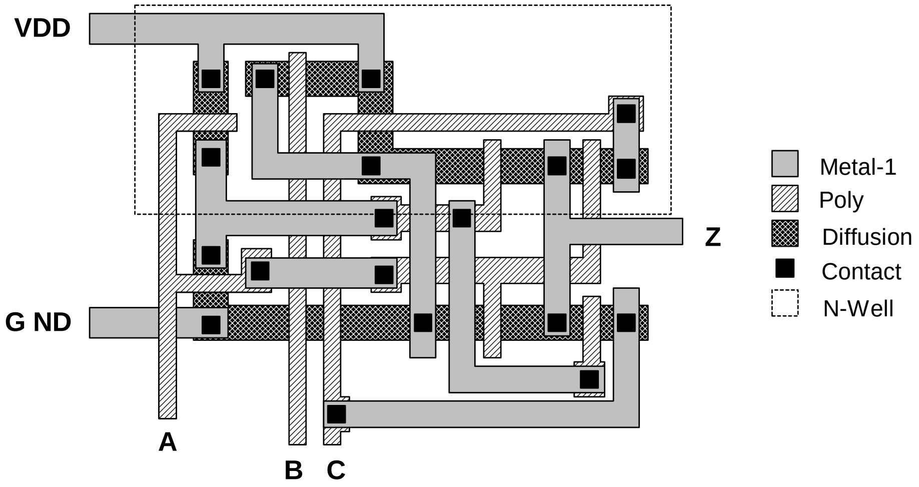
Task 3: Typing a .sim file by hand

In the previous task, you used IRSIM to test a previously designed layout. This is the normal flow you will use in your project; you will do your layouts using Magic, and then test them with IRSIM. However, during debugging, it is often invaluable to know how to modify a .sim file by hand. In this task, you will get comfortable with the .sim format.

1. Open the .sim file (using your favourite text editor) from Task 1. You should be able to identify the transistors (the lines starting with p's and n's). There may also be a number of R and C lines. These correspond to parasitic capacitances that have been extracted. Trace through the file by hand, and understand the correspondence between the layout and the .sim file (focus on the transistors, not the parasitic capacitances).

(continued on next page…)

2. Consider the following layout.



   Create a .sim file that describes this cell by hand. In other words, type in the .sim file from scratch using a text editor. Do not do a layout of the cell; that will take too long. Simulate the .sim file, and verify the functionality (you should be able to figure out the functionality by looking at the layout). Call your file **task3.sim**

3. Using IRSIM, determine whether the rise time of the output **Z** is the same as the fall time of the output **Z**. If not, figure out what you will have to do to make it so. You can change the transistor sizes just by changing the corresponding line in the .sim file.

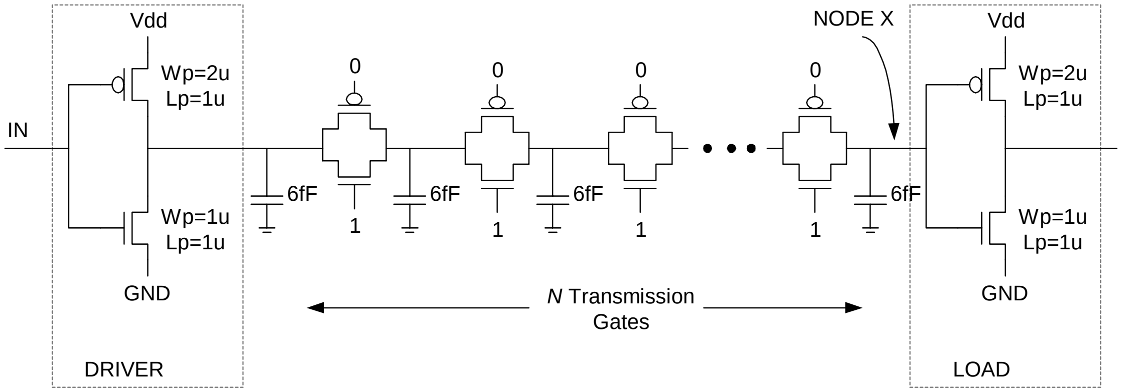
From this task, you will hand in the **task3.sim** file.

<u>Task 4: Transmission Gate Chain Delay</u>

Consider a connection that passes through a series of transmission gates before reaching the load, as shown on the next page. The connection passes through *N* transmission-gates. Assuming all transmission gates are minimum-sized (but that the P-transistor is twice the size of the N-transistor), and that the source and destination inverter are as shown, use IRSIM to find the delay of the driver for N=4. The delay is the time from IN passing Vdd/2 and Node X passing Vdd/2. Repeat this for N=8, N=16, and N=32 (construct a table showing the delay for each value of N). In all cases, assume the input to the driver is rising (hence, Node X is falling). What do you notice about the behaviour of the delay as N increases? (of course the delay increases, but does it increase sub-linearly, linearly, or faster than that). Why do you think this happens?

Note that you do **not** need to actually lay out these gates. Instead, construct a text file describing each circuit and use that as input to the simulator (as in previous tasks). You will have to insert the 6fF capacitor in the netlist yourself. To do this, you can specify a capacitor use the line "C <node> Gnd 6" (this specifies a 6fF capacitor between node <node> and Gnd). The 6fF capacitor models the source/drain parasitic capacitances of the transmission gates.

(continued on next page…)



For this task, put your answers in a text file called **task4.txt** (alternatively, **task4.pdf** if you want to include figures). You do not need to hand in any .sim files or IRSIM output, however, you may want to quote IRSIM results in your answer. I would expect your answer would contain a table of the delay for several values of N, and then a written explanation of your results.

What to hand in:

Each of Tasks 2 to 4 indicate a file that must be handed in:

Task 2: **add8.sim**
Task 3: **task3.sim**
Task 4: **task4.txt** (alternatively, **task4.pdf** if you want to include figures)

You will hand in these files electronically. To do this:

1. You already have a directory under your home directory called **eece479** from Assignment 2.
2. Make a subdirectory under your eece479 directory called **ass3** (note: not assgn3, assn3, etc.)
3. Put the three files listed above in your ass3 directory.
4. Create another file in your ass3 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 ass3 directory should contain ***only*** the three files listed above plus the info.txt file.
6. Enter the following to hand in the file:

```
handin eece479 ass3
```