CS/EE 5830/6830 -- CAD Assignment #1

Due Tuesday, January 25th, 5:00pm
Put assignments in the slot outside the SoC office

For this assignment you'll use *Digital VLSI Chip Design with Cadence and Synopsys CAD Tools*. The book is available at many on-line booksellers including Amazon. The required chapters from this book are also on the class web site. You'll need to look at Chapters 1-4 for this assignment. The chapters are tutorial in nature so you should be able to follow along as you're trying out the tools. I recommend that you don't print them out. They're somewhat long because of the step-by-step nature of the tutorials. Instead I'd open the chapters in your browser in one window, start Cadence in another window, and follow along on the screen. If you have a print copy of the book, that works too.

Note that the book, and the online chapters, are for a slightly different version of the Cadence tools. The book uses the v5.1.41 tools, and we are now using the v6.1.4 tools. Most things look and work very much the same, but you might find dialog boxes that are slightly different than in the book. It should be relaitvely easy to figure out what the difference is. I should have a v6.1.4 update sometime soon also.

First read Chapters 1 and 2 to get a picture of how the Cadence CAD tools are organized, and how you should set things up for using the tools. Then do the following:

1. Complete the first part of the Cadence Composer tutorial in Chapter 3 by making a new library and designing a Full Adder using the standard cells in the **UofU_Digital_v1_2** library. Note that this is a slightly different parts library than used in the CAD book tutorial! Pay attention to where your gates are coming from! You are welcome to use the same schematic as is used in the CAD book, or make a different gate-level circuit that also implements a Full Adder (there are lots of ways to make a full adder). Test the Full Adder using NC_Verilog. I recommend you name your new library **lab1** rather than what is used in the CAD manual.

2. Create a symbol for the Full Adder and use it in building a 4-bit ripple Adder. Simulate the 4-bit Adder using NC_Verilog and observe the timing diagrams. You'll need the information from Chapter 4 for simulation, but you should be able to get away with just the first few sections of that chapter (Section 4.1.1 or 4.1.2 in particular). I know that you may not be a Verilog expert already, but you should be able to understand enough by looking at the examples to get things simulating.

3. Complete the next part of the tutorial in Chapter 3 by designing a 2-input NAND gate using **nmos** and **pmos** transistors from the **UofU_Analog_Parts** library. Use **vdd** and **gnd** symbols for power and ground connections. Again, pay attention to where your transistors are coming from! Create a symbol for the NAND that looks something like a NAND symbol. That is, modify the rectangle provided by Cadence to something that looks more like a standard NAND symbol. Simulate your NAND using NC_Verilog and observe the timing diagrams.

4. Using *only* instances of the 2 input NAND gate that you have created, build a circuit that implements the following Boolean function (don't minimize or manipulate the function, and don't use any gates except for your own NAND gate):

$$F = \overline{A}\,\overline{B} + \overline{A}C + \overline{B}C$$

5. Simulate this circuit using NC_Verilog and observe the timing diagrams.

For all of these simulation tasks, make sure that your Verilog test fixture uses "if" and "$display" statements to check for the correct results in the simulation. You should be able to tell from running your test fixture whether the circuit is working correctly before you look at the timing waveforms (that is, if the circuit produces an incorrect output, an error message should be printed!). For these simulations, and for subsequent simulations in this class, you should either test things exhaustively (i.e. test for all possible input combinations), or describe on a separate sheet what tests you did run and justify why that is a good set of tests. It's unlikely that you'll be able to test larger circuits in future labs exhaustivly so you'll have to put some thought into what to test and why that is a good set of tests. For this assignment, exhaustive testing is the way to go though.

Also, make sure to use an **Asheet** frame from the **UofU_Sheets** library on every schematic! Spend the time to make your schematics neat and orderly! Straighten out the wires, space out the components appropriately, don't over crowd, and generally make things look nice. Neatness counts when grading schematics.

Note that this assignment is to be done individually! If team labs happen, they'll happen later in the semester.

Note also that the tool paths in Chapter 2 of the CAD manual are close, but not exactly the right paths to use. In particular:

- Use the **cad-ncsu** script from **/uusoc/facility/cad_common/local/bin/S11/cad-ncsu**
- The easiest way to do this is add **/uusoc/facility/cad_common/local/bin/S11** to your **path** so that you can then execute the **cad-ncsu** script without typing in the whole path
- Set your **LOCAL_CADSETUP** variable to **/uusoc/facility/cad_common/local/class/6830/S11**
- Make a new directory in which to start up tools. I recommend **IC_CAD** in your home directory
- Make a new directory for Cadence under that **IC_CAD** directory. I recommend calling it **cadence-s11** so that you can keep it isolated from other classes in the future that might use different versions of Cadence.
- Don't worry about copying the **.cdsinit** file as described in the book, the **cad-ncsu** script does this for you now.

Here is the recommended sequence of steps (read the material in Chapter 2 also!). First you will need to log in to a CADE machine (contact the opers in the CADE lab if you don't have a CADE account).

- cd
- mkdir IC_CAD
- cd IC_CAD
- mkdir cadence-s11

Now edit your .cshrc (or .tcshrc if you use that) to add the following things (if you use bash as your shell, the syntax will be slightly different and you will have to figure that out because I don't use bash…):

```
set path = ($path /uusoc/facility/cad_common/local/bin/S11)
setenv LOCAL_CADSETUP /uusoc/facility/cad_common/local/class/6830/S11
```

Now that these lines are in your .cshrc (or other setup file), the next time you login or start a new shell you will be able to connect to your IC_CAD/cadence-s11 directory and start up the Cadence tools with cad-ncsu.

- cd IC_CAD/cadence-s11
- cad-ncsu

Things to Turn In

Turn in hardcopies of:

1. Gate-level schematics of the Full Adder, 4-bit Adder, their Verilog test fixtures, simulation logs and timing diagrams
2. Transistor-level schematic of the 2 input NAND gate, Verilog test fixture, simulation log and timing diagram
3. Schematic of the Boolean function using your new NAND gate, the Verilog test fixture, simulation log and timing diagram

Information about the schematic capture tool, Verilog simulation, printing the simulation logs, etc. can all be found in the CAD book.

Make sure your name is easily visible on all the pages you turn in. Turn in assignments to the box outside the SoC (School of Computing) front office, or bring them to class.

In the future we'll likely use the CAD handin system, but for this assignment, print out things, organize them neatly , put your name clearly on the pages, staple, and hand in as hard copy.