ECE/CS 5710/6710 – Digital VLSI Design
Lab Assignment #1

**Due Tuesday Sept. 9th via Canvas**

# 1 Introduction

For this assignment you'll use Dr. Brunvand's book *Digital VLSI Chip Design with Cadence and Synopsys CAD Tools* to walk you through the steps of completing the assignment. You'll need to use Chapters 1 through 4 for this assignment. The chapters are tutorial in nature so you should be able to follow along as you're learning the tools and completing the assignment.

# 2 Assignment

Following are the tasks which you will need to complete for this Lab.

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 `NCSU_Digital_Parts` library. Note that this is a different parts library than used in the CAD book tutorial! Pay attention to where your gates are coming from! You are welcome to copy the schematic in the CAD manual, or make a different gate-level circuit that also implements a Full Adder. Please name your new library **lab1** rather than what is used in the CAD manual.

2. Test the Full Adder using Verilog-XL or NC_Verilog. Chapter 4 contains information you will need to read in order to simulate your design. Deliverables for this lab include the Verilog testbench for your full adder, the output from the simulation, and the timing waveform that results from the simulation.

3. Create a symbol for the Full Adder and use it in building a 3-bit Adder. You will need to turn in the schematic for your 3-bit adder.

4. Simulate the 3-bit Adder using VerilogXL or NC_Verilog from a testbench that you create, and validate the results. You will turn in the Verilog testbench, the output from the simulation, and the waveform that results.

5. Complete the next part of the tutorial in Chapter 3 by designing a 2-input NAND gate using nmos and pmos transistors from the `NCSU_Analog_Parts` library. Use `vdd`

and `gnd` symbols for power and ground connections. Pay attention to the library where your gates 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 VerilogXL or NC_Verilog and validate the waveforms. Turn in the schematic for the NAND, the Verilog testbench, simulation output, and simulation waveforms.

6. Build a circuit that implements the following boolean function. You are *only* allowed to use instances of the 2–input NAND gate that you designed in the previous step. Don't minimize or manipulate the function. Turn in the schematic for this circuit.

$$F = \overline{a}\overline{b} + \overline{a}c + \overline{b}c \tag{1}$$

7. Simulate this circuit using Verilog XL or NC_Verilog and validate the timing diagrams. Turn in the Verilog test bench, the simulation output, and timing waveform from the simulation.

# 3 Instructions

## 3.1 Setting up your CAD tools

You will need to configure and use the CAD tools correctly in order to complete this lab. Not that the tool paths in Chapter 2 of the book are not the right paths to use. In particular:

1. Make sure your search path contains the following path:
   /uusoc/facility/cad_common/local/bin/F13

2. Use **cad-ncsu** to run Cadence.

3. Set your shell variable **LOCAL_CADSETUP** to:
   /uusoc/facility/cad_common/local/class/6710/F13

4. Make a new directory structure for running the Cadence tools. We strongly suggest that you create a directory named **IC_CAD** in your home directory on the CADE machines. Underneath that create a directory for this class or for the current version you are using, called **5710**.

5. Link the **.cdsinit** file from the design kit to your ∼/IC_CAD/5710 directory in /uusoc/facility/cad_common/NCSU/CDK6-F13/ to correctly define the tool interfaces.

Following is a quick set of steps to perform the setup from your home directory (this is a one-time deal):

```
cd
mkdir IC_CAD
cd IC_CAD
mkdir 5710
cd 5710
ln -s /uusoc/facility/cad_common/NCSU/CDK6-F13/.cdsinit .
```

You will then need to edit your .cshrc or .tcshrc files depending on the shell you use to define the paths and environment variables. (Syntax is similar but a little different if you use bash):

```
set path = ( $path /uusoc/facility/cad_common/local/bin/F13 )
setenv LOCAL_CADSETUP /uusoc/facility/cad_common/local/class/6710/F13
```

Whenever you run the tools and work on the class you will need to connect to the work directory (`cd IC_CAD/5710`), and then start cadence with `cad-ncsu`.

### 3.2 Lab Assignment Aids

For all of the simulation tasks in this lab assignment, 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 exhaustively so you'll have to put some thought into what to test and why that is a good set of tests. However, exhaustive testing will suffice for this assignment.

Make sure to use an `Ax` 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. We'll form teams later.

## 4 Deliverables

Note the following:

1. All the deliverables will need to be placed into a `tar` file and e-mailed as an attachment to `teach-5710@list.eng.utah.edu` Feel free to compress the tar file with `gzip`.

2. CAD tools create LOTS of files! Only include the files that are part of the deliverable, and nothing else.

3. You will need to use the `xv` program to take screen shots of the some of the work that you create. Save the screen shots as .jpg or other compressed format to save disk space.

You can run tar using the following command:

```
tar cvf lab1.tar dir1/file1.jpg dir1/file2.jpg
```

This will create a file lab1.tar with file1.jpg and file2.jpg from directory dir1. If you copy all the required files to one directory, then you can just copy the directory:

```
tar cvf lab1.tar dir1
```

This will copy the whole `dir1` directory into the tar file `lab1.tar`

Following are the deliverables for this lab:

1. A screen shot of the gate-level schematics of the Full Adder and 3-bit Adder.

2. Files containing the Verilog test fixtures, simulation logs, and timing diagrams of the two adders.

3. A screen shot of the transistor level schematic of the 2-input NAND gate.

4. Files containing the Verilog test fixture, simulation log, and timing diagram of the NAND gate.

5. A screen shot of the schematic of the Boolean function in Section 2 using your new NAND gate.

6. Files containing the Verilog test fixture, simulation log, and timing diagram of the boolean function.