

CS/ECE 5710/6710 Digital VLSI Design
CAD Assignment #7
Due Friday October 30th, 11:59pm

Overview: In this assignment you will, as a group, augment your library with a few simple, but important, cells. You will then use your library to synthesize a slightly larger example, and learn how to use the place and route tool (SOC Encounter) on that example.

Two aspects are fundamental to a good library: proper functionality, as discussed in the previous lab, and providing cells with the proper drive strength. In this lab you will extend your base library to include multiple drive strengths for many cells.

To make the best circuit, the CAD tools must choose between cells of the same functionality but different drive strengths. Simply using the functionality of the cell is insufficient, as there can be multiple designs of the same functionality (think of different transistor level XOR designs, and their various delays and properties). Thus, all cells with the same functionality and similar designs are placed into classes, called a footprint. The tools recognize that to improve the performance and power efficiency of a design, they can freely change the drive strength by picking a larger or smaller cell of the same functionality footprint. As we extend our cell library with different drive strengths, we need to include footprint information regarding the cell classifications.

Procedure:

1. Augment your library with the following cells. This means creating layout, cmos_sch, symbol, behavioral, extracted, analog_extracted, and abstract views, along with .lib, .lef, and .v information. Add these cells to your Lib6710_xx library. Remember that all cells with the same function should have the same footprint. So, the INVXx cells should all have footprint “inv” and all non-inverting buffer cells should have the same footprint (i.e. buf), etc.
 - a. **INVX4** – an inverter with 4x drive
 - b. **INVX8** – an inverter with 8x drive
 - c. **BUF4X** – a non-inverting buffer with 4x drive. This is just a series of two inverters with the first being a 1x and the second being a 4x
 - d. **BUF8X** – a non-inverting buffer with 8x drive. The first inverter should be a 2x and the second an 8x inverter.
 - e. **NANDX2** – a NAND gate with 2x drive
2. You should look the other cells in your library, and consider adding versions of those cells with different output drive strengths.

3. It's not critical, but it will make your final layouts look better if you add a few extra widths of filler cells. If your original filler is named FILL1, consider adding a FILL4 and FILL8 cell that are 4 and 8 "standard cell widths" wide. A standard cell width is 2.4μ (the width of your FILL1 cell). This will let the place and route tool fill in open areas with fewer filler cells.
4. Assemble a new Lib6710_xx library (.lib, .lef, and .v) with those new cells.
5. Use your new Lib6710_xx library to synthesize the **controller** state machine from the mips example in your textbook. The **controller.v** behavioral file is in **/uusoc/facility/cad_common/local/class/6710/F09/examples**. Use a target speed of a 5ns clock period for your synthesis. Look at the report file to see if you were able to hit that target speed according to design compiler. Use **syn-dc** and the **syn-script.tcl** (suitably modified for your own library and constraints) from **/uusoc/facility/cad_common/local/class/6710/F09/synopsys** and not beh2str, or use **design_vision** gui (**syn-dv**) and directly drive the synthesis process. Make sure to have **.synopsys_dc.setup** (copied from .../6710/F09/synopsys) in the directory from which you run **syn-dc** or **syn-dv**. As output from synthesis to the next step you'll need a structural Verilog file, and a **.sdc** delay constraints file from synthesis.
6. Place and route the structural Verilog file (**controller_struct.v**) using SOC Encounter (**cad-soc**). See Chapter 11 in the CAD book for details. Use your buffer cells (footprint buf) as your clock-tree synthesis cells. You should end up with a correctly placed and routed circuit with no geometry or connectivity errors, and with (at least) the following files as a result
 - a. **controller_soc.v** – the structural file from SOC Encounter which includes the generated clock tree and the optimizations.
 - b. **controller.def** – the design exchange format (DEF) file that describes the placed and routed circuit
 - c. a report file for the final post-route optimization which shows whether the placed and routed circuit met the 5ns timing.
 - d. A **routed.enc** file which saves the SOC state of the final placed and routed circuit.
7. Now read the **controller.def** file back into icfb (See CAD Chapters 9 and 11 for details).
 - a. Make a new library named **control** (make sure to attach the right technology file)
 - b. Import the **controller.def** file to this library as a layout view. Replace the abstract cell views with layout cell views and run DRC and Extract

- c. Import the **controller_soc.v** file (the structural Verilog file of the final placed and routed circuit from SOC) in as a schematic and symbol view
- d. Compare the schematic and extracted views with LVS and verify that they are the same.

What to turn in:

1. Tar and handin electronically the following directories:
 - a. **Lib6710_xx** – your library directory
 - b. A copy of your **Lib6710_xx.lib**, **.lef**, **.db**, and **.v** files. These can be put in your Lib6710_xx library directory if you like.
 - c. **control** – the Cadence library you used to read in the controller layout and schematic
 - d. from **soc/control** (the directory you used when you ran SOC Encounter)- The structural Verilog file from synopsys, the structural Verilog file from SOC, the controller.def file from SOC and the SOC saved file called routed.enc that saves the final state of your placed and routed design.
2. Print and hand in physically, or produce PDF and include in your handin bundle:
 - a. Your LVS log showing that the layout and the schematic (both produced by SOC Encounter) match
 - b. The timing report from Synopsys design compiler
 - c. The post-Route timing report from SOC Encounter