CS/EE 6710 Digital VLSI
CAD Assignment #5 (Group Assignment!)
Due Tuesday October 10th, 5:00pm

Overview: In this assignment you will, as a group, design and characterize a small (5-cell) standard cell library that has enough combinational cells to build combinational logic using Synopsys synthesis.

Groups: This is a group assignment. You should form groups as soon as possible. Groups can range from two to four members. Send email to teach-cs6710 with the names of the people in your group as soon as you know. If you're looking for a group, let me know and I'll try to match you up.

Procedure: Design five cells as described below following all the cell library template requirements in Section 5.7.1 of the Lab Manual. The distance between the center of the power and ground wires is fixed at 27 microns, for example, as defined in the template. The width of a cell is not fixed, but must be in increments of 2.4 microns. There are lots (!) of details in the Lab Manual.

For this lab you should make a new Cadence library called UofU_6710Lib to hold your cells, and only the cells. This will make things easier later to keep your library separate from your project design. Once I know who is in the groups are I can set up UNIX groups for you so that you can use the group rw features to give group access to one single library. You will, of course, have to choose which group member has the library in "their" directory, but you should all be able to point to it in your Library Path. If you set your UNIX permissions to RWX for your group, and make sure that all the files in that directory are owned by the group, you should be all right.

The cells you should add to your UofU_6710Lib library are:

- INVX1: standard 1X sized inverter
- NAND2X1: two-input NAND gate with standard size transistors (remember to take stacks into account when deciding on "standard" sizes)
- NOR2X1: two-input standard NOR gate
- TIEHI: A cell that provides a connection to vdd. There is only one output, and no inputs. The single output is a connection to vdd. This should be a resistive connection through a transistor so that the gate that it connects to is protected

from direct connection to the power supply. The schematic is given at the end of this document.

- TIELO: A cell that provides a connection to gnd. See the schematic at the end of this document.

Your cells should have the following views: cmos_sch, symbol, layout, behavioral (Verilog), extracted, and analog_extracted.

Once you have the cells designed and simulated (and passing DRC and LVS), you should characterize the cells as described in Chapter 7 in the Lab Manual. This should result in a liberty format file for your five-cell library. This is a very involved and somewhat picky procedure! I recommend using SignalStorm as described in Chapter 7, but this is a rather picky program in terms of getting the inputs just right. You can also simulate by hand or with scripts using Spectre directly if you have trouble using SignalStorm.

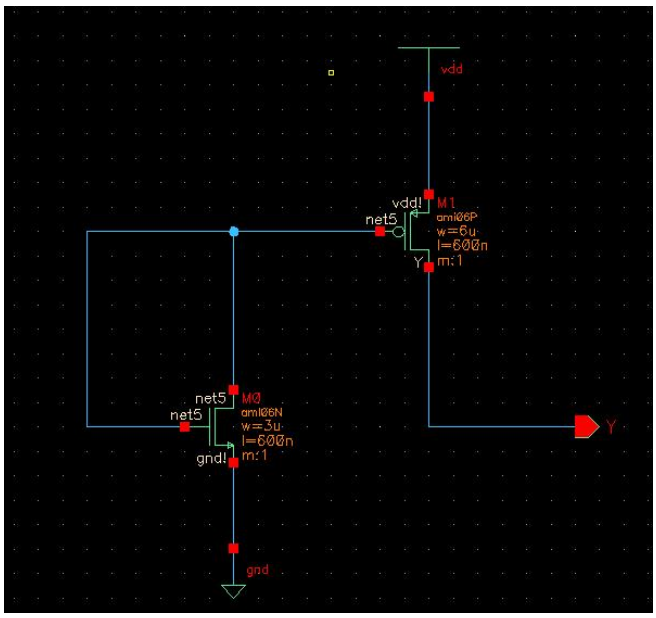
Once you have a .lib, you should compile it into a .db using Synopsys design_compiler. This is described at the end of Chapter 7.

With a .db file of your five-cell library, you should demonstrate that you can synthesize a simple combinational circuit using Synopsys. You can use the simple beh2str script as described in the first section of Chapter 8, or you can use the more general dc_shell script or design_vision versions if you like. The idea for this assignment is just to make sure that your cell library is specified correctly so that synthesis will work!

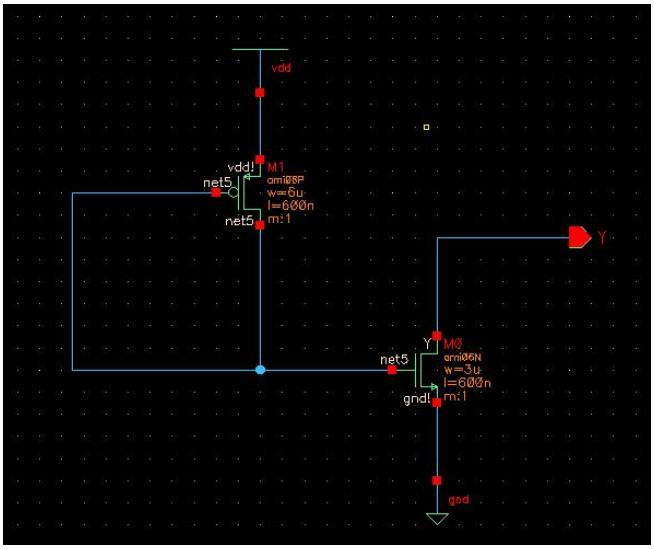
Turn in documentation of your cell library. For **each** cell turn in: cmos_sch, layout, and functional simulation (Verilog-XL) of the cmos_sch view (i.e. not the behavioral view). Also turn in your UofU_6710Lib.lib file with the entries for all five of the standard cells. Also turn in behavioral and structural examples demonstrating Synopsys synthesizing to your mini-library. The synthesis demo can be any small circuit you like, but should show that Synopsys is correctly synthesizing from your library. Note that since your current library has no flip flop, this will have to be a combinational circuit.

Turn in only one set of documentation per group!

TIEHI schematic – diode-connected N-type provides a low voltage to the gate of the P-type that provides the pull-up output.



TIELO schematic – diode-connected P-type provides high voltage to gate of the N-type that provides the pull-down output.



There is an interesting issue when trying to simulate these cells with Verilog-XL: the switch level simulator doesn't know how to simulate diode-connected transistors. So, in order to correctly simulate these cells you need to trick it. Remember the trick we did in the register schematic of giving a node an attribute of trireg? Well, here we need to do the same thing, but we want the trireg node to be initialized to vdd or gnd instead of just holding the previous value. This is because from Verilog-XL's point of view, there is no

previous value! So, we need to give it a value to start with. A trireg that is initialized to Vdd is called **tri1** and a trireg that is initialized to gnd is called **tri0**. So, use the same technique that you used to add the netType attribute to the wire in CAD3 and give the internal nodes of the Tiehi and Tielo cells the right tri1 or tri0 attributes so that they will have the correct values in Verilog-XL. Note that these attributes have no effect on Spectre, so your analog simulations will correctly model the diode-connected transistor and everything will work out.