CSCE 434/834, Fall 2010

Homework 2 (Due Sept 15)

Note: All problems are equally weighted.

1. A four-part problem:
   (a) Represent the Boolean function $F(x,y,z) = x(y+z)+yz$ as an *ordered binary decision diagram* (OBDD) with the variable ordering (x, y, z). Your goal should be produce the smallest possible representation in terms of the number of distinct nodes in the diagram. (I briefly discussed binary decision diagrams in the class. You can learn more about it from online sources, including Wikipedia).

   (b) Derive a mux-based implementation for the function *F* in Problem 1 that follows directly from the OBDD representation of the function. For inputs, you may use the constant functions 0 and 1, in addition to the *uncomplemented* variables *x, y,* and *z.*

   (c) Translate the mux-based implementation to a CMOS implementation that uses only *transmission gates* (TGs) – it is allowed to use inverters for control inputs of the transmission gate.

   (d) Assume, again, that a TG implementation is desired but the logic levels in the implementation must be restored after each stage of the transmission gate by *inverting buffers* (i.e. standard CMOS inverters with certain drive capability). This means that on any path from inputs any successive TGs must be separated by an inverter. Redo, the TG implementation under this constraint.

2. You are to implement the function F in Problem 1 in a *hierarchical* design that makes use of the library of six combinational-logic cells whose layouts are shown on the inside back cover of your text book. Specifically, these six cells implement the following functions: Inverter, Nand2, Nor2, Nand3, AOI21, and Mux2; further, they are laid out according to the guidelines stated on p. 16 of Lab 1 (these are also the guidelines followed in Lab 1 for the Nand2) so that they can placed horizontally by simple abutment of cells.

Your goal should be to use a minimum number of library cells in your implementation. Assume the cells are placed horizontally. Show the implementation at the top level, with cells shown symbolically, and connected together using vertical metal2 wires and horizontal metal3 wires. After you have finished your implementation, estimate the overall area (width x height) of the layout in the lambda units.

3. Suppose the *complement* of function F in Problem 1 was implemented as a single CMOS gate that has the fewest possible number of transistors (essentially, you did this in the last problem of the previous homework). For this problem, you are asked to translate the CMOS schematic to a corresponding *stick-diagram* of the layout. Your goal should be to have a neat and compact layout that, preferably, follows the layout guidelines for standard cells stated in Lab 1. Then, following the method described in Section 1.5.5 of the textbook, estimate the area of your layout. Compare the area with that you get for Problem 2 and comment on the difference.

4. The stick diagrams for the Inverter and Nand3 are shown on the inside back cover of the textbook. During design, one could start with these diagrams and "flesh" them out to the layouts shown there. Conversely, it is also possible to *reverse engineer* a layout to a stick diagram and from there to a CMOS schematic.

(a) Consider the 6T SRAM layout given on the inside back cover. You are given that this layout corresponds to the CMOS schematic shown in Fig. 12.3. Go through the reverse engineering process to verify this fact, i.e. you should be able to relate each transistor in the layout with the corresponding one in the schematic. Further, you should also verify that all the connections are correct according to the schematic.

(b) (**Bonus**) After you have practiced the process of circuit extraction from the layout for the SRAM example, now do the same for the more complex layout of the full adder shown on the inside back cover of your textbook.