The UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

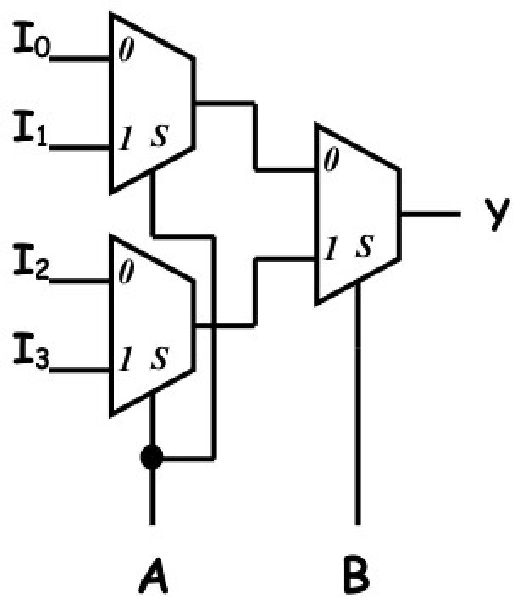
Comp 411 Computer Organization
Spring 2010

Problem Set #5
Issued Monday, 15 March 2010; Due Monday 29 March 2010

Homework Information: Some of the problems are probably too time consuming to be done the night before the due date, so plan accordingly. Late homework will not be accepted. Feel free to get help from others, but the work you hand in should be your own.

Problem 1. Mux Madness (15 points)

Suppose you wanted to implement a Boolean function *Y* of two inputs *A* and *B* using multiplexors, as shown in the figure.



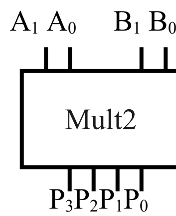
Give binary values for I_0, I_1, I_2, and I_3 which implement the following functions on the two inputs *A* and *B:*

a) Y = XOR(A,B)
b) Y = (A == B)
c) Y = (A <= B)

Problem 2. "Go Forth and Multiply" (55 points)

a) [15 points] Design logic to perform the multiplication of two 2-bit unsigned integers, producing a 4-bit result. (*Hint:* First write a truth table for each output bit.) Draw a gate-level circuit diagram, i.e., show an implementation using Boolean gates (AND, OR, NOT, XOR, etc.), *not* using transistors. How many Boolean gates did you use?

b) [10 points] Assume that the 2-bit multiplier that you designed in part a) is represented as the following function block:



(i) Use this function block and single-bit full-adder blocks to build a 4-bit multiplier (a multiplier that takes two 4-bit inputs, and generates an 8-bit result). *Note:* Do not build the 4-bit multiplier from scratch using basic Boolean gates; you must build it using 2-bit multiplier blocks and full-adder blocks!

(ii) Make a symbol for your 4-bit multiplier showing all the inputs and outputs and call it Mult4, similar to the symbol for Mult2 above.

(iii) How many Boolean gates did your design need in all?

c) [15 points] Design a 3-input 2-bit multiplier using Mult2 and Mult4 blocks. That is, this unit should take three inputs: A, B and C. Each of these inputs is a 2-bit number (i.e., A_1A_0, B_1B_0 and C_1C_0.).

d) [15 points] Suppose you wanted to compute the cube (third power) of a 2-bit number. You could do this in two different ways:

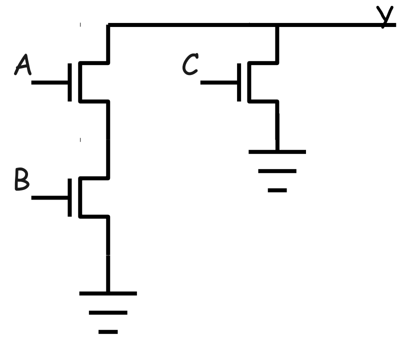
(i) You could use your design from part c) above, and feed it the same 2-bit number into all of its inputs. How many Boolean gates would this need in all?

(ii) Alternatively, you could build a cube circuit from scratch. Write a truth table in which in the input column contains the 2-bit input, and the output column is the 6-bit result. For each of the 6 output bits, give the Boolean equation (circuit not needed). How many Boolean gates are needed for this implementation?

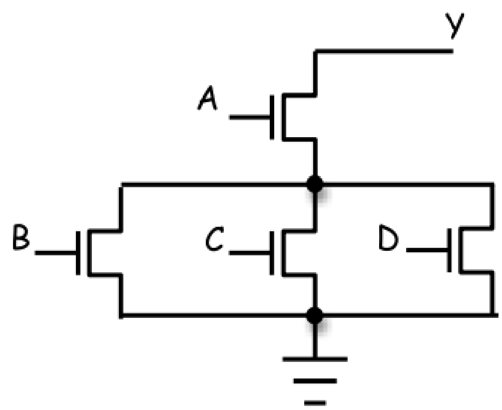
Problem 3. "Fishing for Complements" (30 points)

Show the complementary set of p-channel or n-channel transistors that complete the following CMOS circuits, and give a truth table for the resulting circuit:

a)



b)



c)