Fundamentals of Digital Logic Design

ECE/CS 3700

Spring 2018, Homework # 3

Due Date: Tuesday, Feb 27, 5pm sharp, due in the HW locker. At 5pm, I will upload solutions to this HW, so that you will have ample time to prepare for the upcoming Mid-Term exam.

1) (15 points) **Propagate-generate-delete signals in adders**. In class, we studied the carry look-ahead adder as a means to speed-up carry propagation delay of the ripple carry adder. Look-ahead adders make use of the generate (G_i) and propagate (P_i) signals to *precompute* whether or not stage i would output a carry. Similar to using the propagate (P_i) and generate (G_i) signals, an adder can be designed by using a propagate (P_i), generate (G_i) and a delete (D_i) signal. Consider the Truth table of the full adder shown in Table I:

TABLE I

TRUTH TABLE OF A FULL ADDER

a	b	c_i	S	c_{i+1}	Carry Status
0	0	0	0	0	Delete
0	0	1	1	0	Delete
0	1	0	1	0	Propagate
0	1	1	0	1	Propagate
1	0	0	1	0	Propagate
1	0	1	0	1	Propagate
1	1	0	0	1	Generate
1	1	1	1	1	Generate

The first two minterms m_0, m_1 correspond to the condition where the carry-out signal gets suppressed (deleted) at c_{i+1}, independent of the value at c_i. It is easy to see that the *delete signal* $D_i = a' \cdot b'$. Prove that:

- $Sum = P_i \cdot \overline{C_i} + D_i \cdot C_i + G_i \cdot C_i$
- $C_{i+1} = G_i + \overline{D_i} \cdot C_i$.

2) (10 points) Given that A, B, C_i are the inputs to a full adder, S and C_o are sum and carry-out, respectively, prove that:

- $S = ABC_i + \overline{C_o}(A + B + C_i)$.

3) (25 points) **Subtractor design with two's complement numbers**. Suppose that you are given two **3-bit unsigned** numbers $A[2:0], B[2:0]$ that have to be added together ($C = A + B$). Suppose, further, that you are given a pre-designed *4-bit adder* that you have to use for this purpose. A 4-bit adder takes inputs $X[3:0], Y[3:0]$ and adds them. In order to do the addition correctly, we can take vectors A, B and concatenate a leading 0 to make them 4-bit vectors and then map the inputs; i.e. in Verilog terms: $X[3:0] = \{1'b0, A[2:0]\}$; and similarly $Y[3:0] = \{1'b0, B[2:0]\}$. This way, n-bit unsigned integers can be scaled to larger bits.

However, this technique may not work for 2's complement scheme. So, now you have to answer the following: You are given two 3-bit vectors $A[2:0], B[2:0]$ that are already given in **3-bit two's complement form**. You are asked to compute $A - B$ (subtraction). Suppose that you are already given a *4-bit ripple carry adder* as a pre-designed black-box as given in Fig. 1. This circuit takes as inputs $X[3:0], Y[3:0]$ and a carry-in C_i and computes the sum $S[3:0] = X[3:0] + Y[3:0]$. The ripple carry adder also produces a carry-out C_o. You have access to only the primary inputs and outputs of this block ($X[3:0], Y[3:0], C_i, S[3:0], C_o$). You cannot access any internal signals of this 4-bit ripple carry adder.

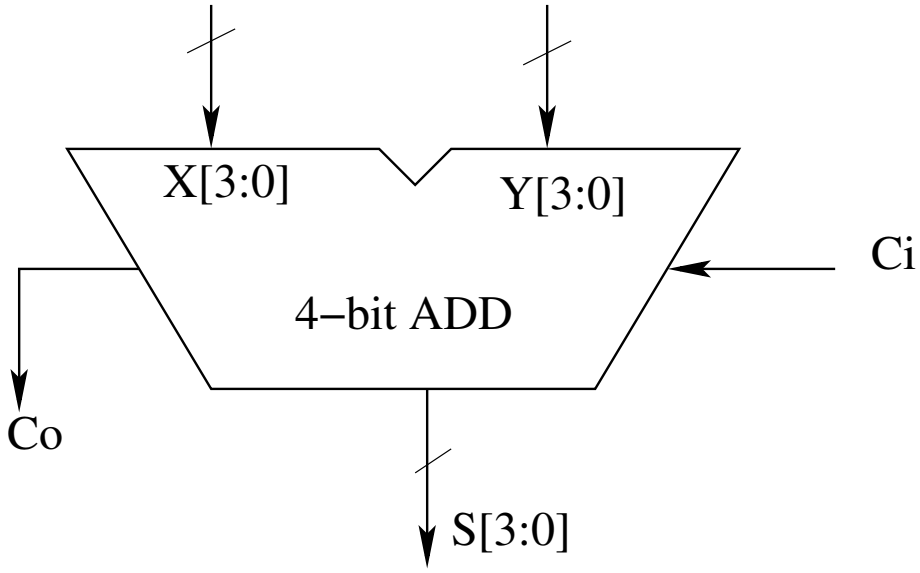

Fig. 1. A pre-designed 4-bit adder.

- You have to use this 4-bit adder to subtract the given 3-bit two's complement numbers A, B.
- Along with this 4-bit adder, you are allowed to use any AND/OR/XOR/NOT Boolean logic gates.
- Does your design require an overflow signal to be generated? If so, design the logic for the overflow signal. If you think an overflow is not needed, explain the reason.
- Show your design and schematic (or a Verilog code, if you wish), and please depict the signal connections properly (e.g. if A[3] is connected to X[3], show it clearly on the schematic). Also, demonstrate the correct functioning of your circuit using an example input stimulus.

4) (25 points) **Multiplier design**. You are asked to design an array multiplier that multiplies a 4-bit number $A = (a_3, a_2, a_1, a_0)$ by a 3-bit number $B = (b_2, b_1, b_0)$. Consider that pre-designed 4-bit adders are available to you. (Recall that a 4-bit adder adds two four-bit numbers). Design the multiplier using only **two 4-bit adders** and a minimum number of two-input AND/OR/NOT/XOR/XNOR gates. Show a block diagram or a schematic of your design depicting input and output bits clearly. (Solve this problem properly, and you've understood the concept of array multipliers!).

5) **(25 points)** This design example will also use a n-bit multiplexor; we will study its design and application

in class on Tuesday 2/20. You are asked to design a circuit that i) takes a 4-bit vector $X[3:0]$ already given in two's complement form; and ii) outputs a 4-bit signal $Y[3:0]$ where Y represents the absolute value of X, i.e. $Y = |X|$. Note that if A is an integer, $|A| = |-A| = A$. In other words, if $X = 3 = 4'b0011$, the circuit outputs $Y = 3 = 4'b0011$. If $X = -3 = 4'b1101$, $Y = 3 = 4'b0011$. Design this circuit and show the schematic. You may assume that a 4-bit adder of Fig. 1, along with 4-bit multiplexors and all AND/OR/XOR/NOT gates are available to you.

Have fun!