CS/ ECE 252 Introduction to Computer Engineering

Homework 4 – Due at Lecture on Wednesday, March 4th

Instructions: You should do this homework in a group of **TWO** or **THREE** students from the SAME 252 section. You should hand in **ONE** copy of the homework. Front page of the answer sheet should contain

- **Name** and **UW ID** of the students in that group
- **Section number** (Lec 001 or Lec 002)
- Multiple pages should be **stapled**.

Warning: Most home works will use questions from your textbook, Patt and Patel's *Introduction to Computing Systems*, which we abbreviate *(ItCS)*

First contact for questions is TA Maheswaran Venkatachalam (kvmakes@cs.wisc.edu)

Problem 1

A – 2-bit input {A[1] A[0]}

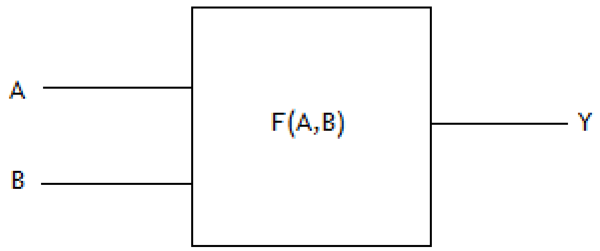
B – 2-bit input {B[1] B[0]}

Y – 3-bit output {Y[2] Y[1] Y[0]}

Y = F(A, B) = NOT(A) + B (+ → binary addition. Don't confuse with OR)

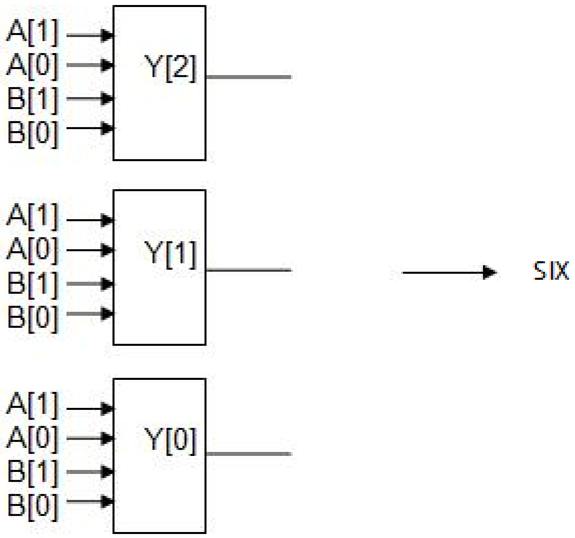
a. Draw the truth table for the function F(A,B).

A[1]	A[0]	B[1]	B[0]	Y[2]	Y[1]	Y[0]
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			



b. Implement Y[0] using AND/ OR/ NOT gates.

c. Given the blocks for implementation of Y[2], Y[1], Y[0], implement the logic that will indicate if F(A,B) is equal to 6.



Problem 2

You are assigned the task of designing a circuit with some sequential logic elements and logic gates. The circuit should turn on a light when the input remains 1 (High) for 3 consecutive clock cycles. Once the light is turned on, it should be in ON position irrespective of the changes in the input. You have to design the circuit using one-hot-encoding for state machines. One hot encoding uses one flip-flop for each state (one active flip-flop per state).

The table below shows how states are represented by bits.

States	S_0 S_1 S_2
A	1 0 0
B	0 1 0
C	0 0 1

a. Draw the state diagram as in figure 3.31 on page 78 ItCS. How many flip-flops do we need?

b. Show the truth table for the circuit.

c. Show the block diagram of the circuit and the combinational logic used as in figures 3.32a and 3.32b on page 79 of ItCS.

d. How many fewer flip-flops would it have taken using binary state encoding?

Problem 3

With a 2:1 MUX, we can implement any two variable function F(A,B) by appropriately connecting 0,1,A, NOT(A), B,NOT(B) to the various inputs.

a) Design a 2 input OR gate using only a 2:1 MUX

b) Design a 2 input AND gate using only a 2:1 MUX
(Note: No additional logic elements are to be used)

Problem 4

If you see the figure 4.3 in page 102 of ItCS, there is one register called PC (Program Counter). It is a 16 bit register and has an input which loads the program counter.

Your job is to design the program counter using master-slave flip-flops (You can use AND/ OR/ NOT/ MUX). When the enable pin is active, new value should be loaded into the register. When the enable pin is low, the register must retain its value. A single master-slave flip-flop can store 1 bit. Hence, you can go for a hierarchical design – show a 4 bit register using 4 master-slave flip-flops. Combine 4 of this type to get the 16 bit register (PC).

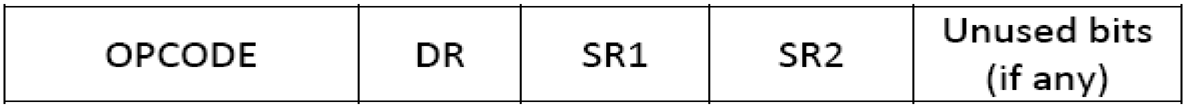
Constraint: Input to the clock terminal of the flip-flop should only be clock signal (i.e. you are not allowed to combine the clock signal with any other signal and input that at the clock terminal of the flip-flop)

Problem 5

State the phases of instruction cycle and briefly describe what operations occur in each phase of the instruction cycle.

Problem 6

A 32 bit instruction has the following format. DR, SR1 and SR2 are registers.



OPCODE	DR	SR1	SR2	Unused bits (if any)

If there are 255 opcodes and 64 registers,

a) What is the minimum number of bits required to represent the OPCODE?
b) What can be the maximum number of bits that can represent the OPCODE?
c) What can be the maximum number of the unused bits?
d) What could be the potential advantage of using more than the minimum required number of bits for representing the OPCODE?

Problem 7

a) If the machine cycle is 4 nanoseconds (i.e. $4*10^{-9}$ seconds), how many machine cycles occur each second?
b) If the computer requires on average 5 cycles to process each instruction, and the computer processes instructions one at a time from beginning to end, how many instructions can the computer process in 1 second?