Problem Set #2

Date Assigned: Wednesday, February 18, 2016
Date Due: Friday, February 26, 2016
Points: 50

1. (15 pts) Consider the Boolean function: $F(XYZ) = \bar{X}YZ + XZ$

   a. Derive a simplified algebraic expression for $\bar{F}$. Express your simplified expression in sum-of-products form. However you do your derivation, I want to see one step at a time.

   b. Show that $F \cdot \bar{F} = 0$, one step at a time.

   c. Show that $F + \bar{F} = 1$, one step at a time.

2. (20 pts) Using Logisim v2.7.1 (http://www.cburch.com/logisim/index.html), you are to create a logic circuit that converts Binary Code into Gray Code. Gray code is an alternative representation of integers. The following table shows the decimals 0-15, their binary code representation, and their Gray code representation.

Decimal	Binary Code (input)	Gray Code (output)
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

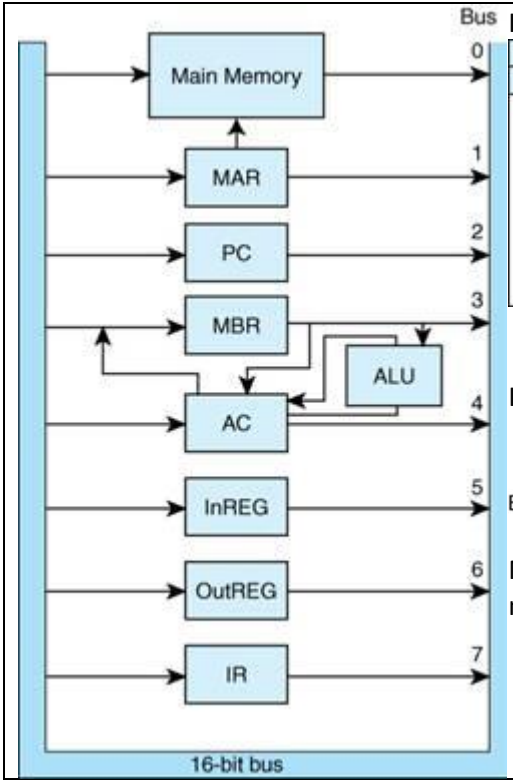
   a. In your own words, explain the motivation behind designing Gray Code? What was the problem with natural binary codes that Gray Codes solve?

   b. Write expressions for each of the four outputs in simplified POS form.

   c. Simplify the expressions that you derived in part b.

d. Build the combinational logic circuit for the binary to gray code convertor in Logisim. Your circuit must contain:
    a. The gates, wiring, and output pins.
    b. A single clock.
    c. A single counter that counts from 0 to 15.
    d. A single splitter that takes the output from your counter as input and feeds its output into your circuit that converts the four-bit inputs into the correct outputs for the Gray code. So, when your counter contains the decimal 8, your output will be 1100.
    e. Save your circuit as gray.circ

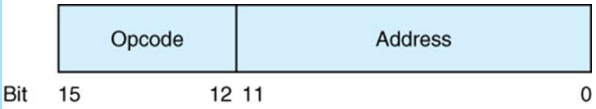
3. (15 pts) Consider the following non-IAS architecture:



Instruction Set

Instruction Number		Instruction	Meaning
Bin	Hex		
0001	1	Load *X*	Load the contents of address *X* into AC.
0010	2	Store *X*	Store the contents of AC at address *X*.
0011	3	Add *X*	Add the contents of address *X* to AC and store the result in
0100	4	Subt *X*	Subtract the contents of address *X* from AC and store the result in AC.
0101	5	Input	Input a value from the keyboard into AC.
0110	6	Output	Output the value in AC to the display.
0111	7	Halt	Terminate the program.
1000	8	Skipcond	Skip the next instruction on condition.
1001	9	Jump *X*	Load the value of *X* into PC.

Instruction Format



Note: The Main Memory is byte-addressable and the width of each memory location is 1 byte.

a. What is the maximum directly addressable memory capacity (in bytes)? Explain.

b. What is the minimum number of bits needed for each of the following registers? Explain your answer.
(1) MAR (2) MBR

c. Consider the following program:
Load 104
Add 105
Store 106
Halt

What is the machine language (in HEX) of this assembly language program?

d. If the program starts at location 0 in Main Memory, exactly how many main memory accesses are needed to complete the program execution? Explain your answer.

How to turn in your solution:

- Please make sure your problem sets are typed, answered in order, and stapled together. Name your word document 02punet.docx.

- A hard copy of your Problem Set Solution is due on the instructor's desk by 9:15am on the day the assignment is due.

- Create a folder named (02punetid) and place: a) the word document with all of your solutions typed up and b) the file gray.circ into the folder. Then drop the 02punetid folder into the CS 430 drop folder on Grace.