CSEE W3827

Fundamentals of Computer Systems

Homework Assignment 1

Prof. Stephen A. Edwards

Columbia University

Due Friday, June 2nd, 2017 at 1:00 PM

Print this out and turn it in. You may enter answers on the computer or write them in by hand.

This homework requires you to use Logisim, which you can download from `http://www.cburch.com/logisim/`

Name:

Uni:

1. (5 pts.) What are the values, in decimal, of the following bytes if they are interpreted as 8-bit numbers in

	00110101	10111010
binary		
one's complement		
two's complement		

2. (5 pts.) Complete the truth table for the following Boolean functions:

$$a = X\overline{Y} + XYZ + \overline{X}\,\overline{Y}$$
$$b = (X + \overline{Y})(X + Z)(\overline{X} + Z)$$

X	Y	Z	a	b
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

3. (20 pts.) Consider the function F whose truth table is shown below.

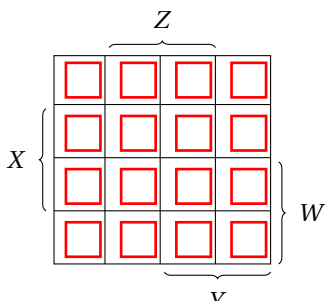
W	X	Y	Z	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	X
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Note: "X" indicates a don't-care output

(a) Write F in sum-of-minterms form.

!W!X!YZ + !WX!YZ + ?

(b) Fill in this Karnaugh map for F



(c) Use your Karnaugh map to write a minimal sum-of-products for F

In Logisim,

(d) Implement the circuit corresponding to your minimal sum-of-products representation. Draw the circuit yourself; do not use "Build Circuit." Verify your circuit using Logisim's Combinational Analysis feature (Project→Analyze Circuit).

Print your solution and attach it.

(e) Use your Karnaugh map to write a minimal product-of-sums representation for F.

(f) Implement the circuit corresponding to your minimal product-of-sums. Again, verify your circuit.

Print your solution and attach it.

4. (20 pts.) Create a circuit for a 3-to-8 decoder using AND gates and inverters only. Arrange and name the inputs and outputs as shown below. Treat the X input as the most significant bit. Only one output should be true at any time.

$$\begin{array}{ll} & \rightarrow A0 \\ X \rightarrow & \rightarrow A1 \\ Y \rightarrow & \\ Z \rightarrow & \vdots \\ & \rightarrow A7 \end{array}$$

Implement your circuit in Logisim, verify it, and print and attach it.

5. (15 pts.) In Logisim, implement $F = X\overline{Y}Z + \overline{Y}\,\overline{Z} + XY\overline{Z} + X\overline{Y}Z$ using just constants and

   (a) a 3-to-8 decoder (under "Plexers→Decoder." Set "include enable" to "No" and note the input wires are a bundle at the bottom) and an OR gate;

   (b) an 8 input mux; and

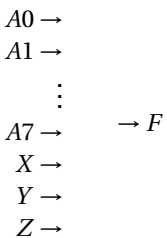
   (c) a 4 input mux whose select inputs are X and Y, and an inverter.

   Implement each of these circuits in Logisim, verify them, and print and attach them.

$$\begin{array}{l} X \rightarrow \\ Y \rightarrow \quad \rightarrow F \\ Z \rightarrow \end{array}$$

6. (15 pts.) Implement an eight-input mux using two-input muxes only (constants are OK).

   Arrange your inputs and outputs as shown below.



   Here, $A0$ through $A7$ are the eight inputs, and X, Y, and Z are the three selects. X is the most significant bit, selecting between, e.g., $A0$ and $A4$.

   Implement your circuit in Logisim, verify it, and print and attach it.

7. (20 pts.) Implement the combinational portion of a four-bit binary counter that wraps around after 9, i.e., for inputs 0, 1, . . . , and 9, it should return 1, 2, . . . , 9, and 0, respectively. Give it four inputs, W, X, Y, and Z, and three outputs A, B, C, and D. E.g., when $WXYZ = 0010$, $ABCD = 0011$.

   Implement your circuit in Logisim using only AND, NAND, OR, NOR, XOR, XNOR, and inverters, using as few as you can, verify your circuit, and print and attach it.