CSEE W3827

Fundamentals of Computer Systems

Homework Assignment 1

Prof. Stephen A. Edwards

Columbia University

Due Tuesday, June 9th, 2015 at 5:30 PM

Print this out and turn it in. Enter answers on the computer or manually on the printout.

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

	00010011	10011010
binary		
one’s complement		
two’s complement		

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

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

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		

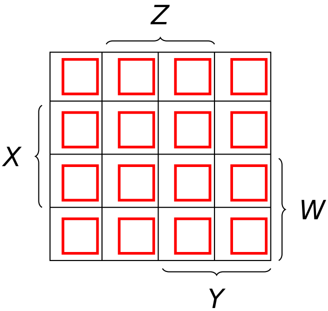
3. (20 pts.) Consider the function *F* whose truth table is shown below

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

(a) Write the function *F* in sum-of-minterms form. Two are given.

!W!XY!Z + !W!XYZ

(b) Fill in this Karnaugh map for *F*



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

In Logisim,

(d) Implement the circuit corresponding to your minimal sum-of-products representation. 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 4-to-10 decoder using AND gates and inverters only. Arrange and name the inputs and outputs as shown below. Treat W as the most significant bit. Only one of the outputs should ever be true.

$$\begin{array}{rcl} W \to & \qquad & \to A0 \\ X \to & & \to A1 \\ Y \to & & \vdots \\ Z \to & & \to A9 \end{array}$$

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

5. (15 pts.) In Logisim, implement $F = X\,Y\,\overline{Z} + YZ + \overline{X}\,Y$ 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.

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

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

   Arrange your inputs and outputs as shown below.

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

   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 three-bit binary counter with an enable input. Give it four inputs, X, Y, Z, and E, and three outputs A, B, and C.

   When E is 0, A, B, and C should be X, Y, and Z respectively.

   When E is 1, A, B, and C should be X, Y, and Z plus one, with A and X the MSBs.

   Your counter should wrap around, i.e., 7 + 1 = 0.

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