CSEE W3827: Fundamentals of Computer Systems

Homework Assignment 1. Stephen Edwards. Columbia University

Due Sunday, May 16 at 11:59 PM EDT via Courseworks

Put your answers in the dashed boxes provided and upload a PDF file to Courseworks. E.g., by editing it in Inkscape (https://inkscape.org) or printing and scanning. This assignment also requires you to use Digital, a logic simulator available at https://github.com/hneemann/Digital. Download hw1.zip from the course website.

Name: Uni:

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

	00110100	10111000
binary		
one's complement		
two's complement		

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

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

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		

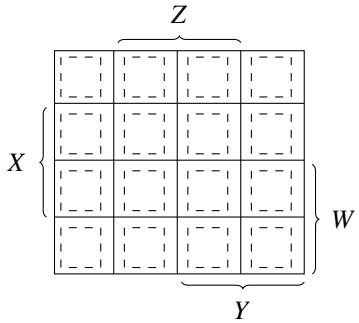
3. (20 pts.) Consider the function F whose truth table is shown below.

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

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

(a) Write F in sum-of-minterms form; use ! for negation.

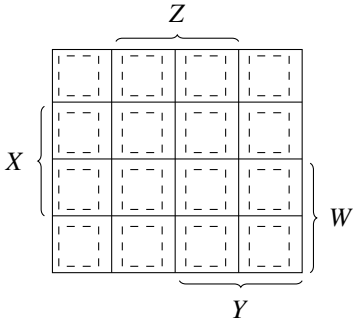
(b) Fill in this Karnaugh map for F and circle a minimum sum-of-product.



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

(d) In the Digital simulator, starting from the hw1-3d.dig template, implement the circuit corresponding to your minimal sum-of-products representation. Draw the circuit yourself using AND, OR, and NOT gates only. Verify your circuit using the tests provided. Run them with Simulation→Run Tests. Upload your modified hw1-3d.dig file to Courseworks.

(e) Again, fill in this Karnaugh map for F, but this time circle a minimum product-of-sums.



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

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

4. (20 pts.) Starting from hw1-4.dig, create a circuit for a 3-to-8 decoder using NAND gates only. Make sure your circuit passes the included tests and upload your hw1-4.dig file to Courseworks.

5. (15 pts.) Starting from hw1-5.dig, implement $F = X\overline{Y}\,\overline{Z} + Y(X+Z)$ using just constants (Components→Wires→Constant value) and

   (a) a 3-to-8 decoder and an OR gate (you may have to change the number of its inputs);

   (b) an 8 input mux; and

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

6. (15 pts.) Starting from hw1-6.dig, implement an eight-bit mux using four two-input muxes and one four-input mux. $A0$ through $A7$ are the eight data inputs, X, Y, and Z are the three selector inputs (X is most significant, selecting between, e.g., $A0$ and $A4$). Use a splitter (Components→Wires→Splitter/Merger) to assemble the a two-bit bus for the four-input multiplexer's select input. Try ''1,1'' for input splitting and ''2'' for output splitting.

7. (20 pts.) Implement the combinational portion of a four-bit binary counter that wraps around after 9, i.e., for inputs 0, 1, ..., 8, and 9, it should return 1, 2, ..., 9, and 0, respectively. Give it four inputs, W, X, Y, and Z, and four outputs A, B, C, and D. E.g., when $WXYZ = 0010$, $ABCD = 0011$.

   Implement your circuit in Digital from hw1-7.dig using only AND, NAND, OR, NOR, XOR, XNOR, and inverters, using as few as you can, and verify your circuit. Hint: use the XOR gates provided.