

Homework 2

Due: 30 Sep 2009 1:55pm

All homeworks are due at 1:55pm in the **CS 31 bin** on the second floor. No late homeworks are accepted.

Please include your *login name* on each piece of paper you hand in, and please staple your pages together before handing in.

Some of the problems on this homework must be handed in electronically. Please do each problem in a separate file. Run `cs031_handin hw02` in the directory where your files are stored to hand in. You will be prompted for the name of each file.

Problem 2.1

Draw circuits corresponding to the following logical expressions.

Please label your inputs and output, and **please keep your inputs in alphabetical order starting from the top**.

Hand in a separate diglog file for each part of the problem (i.e. do part (a) in one file, and part (b) in another).

Remember that AND takes priority over OR.

- a. $(A + B) \bullet (A \bullet C')$
- b. $(D' + A) \bullet ((B + C')')$

Problem 2.2

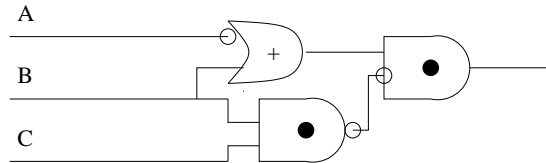
Label each of the following logical equalities as true or false.

- a. $A' + B \bullet C' = (A' + B') \bullet (A' + C')$
- b. $(C + A) \bullet (A + B) + B \bullet C + A' = B \bullet C$
- c. $A \bullet B + C = ((A' + B') \bullet C)'$
- d. $(A + (B \bullet C)) \bullet (D \bullet (E + F)) = (A + B) \bullet (A + C) \bullet (D + E) \bullet (D + F)$
- e. $(A + B') \bullet C = ((A' \bullet B) + C)'$

Problem 2.3

Write a logical expression for the following circuit.

Use * to indicate • in your write-up.



Problem 2.4

The following Karnaugh map has a few possible minimal sum-of-products expressions. Give two such expressions (at least two exist).

Note: This Karnaugh map is in a slightly different format from the ones done in class. The variables are true where they fall inside the associated span (e.g. z is true in the middle two columns; w is true for the bottom half of the table).

$F(w, x, y, z)$

		x	
		z	
		0	1
		5	4
		1	0
		3	7
		0	1
		2	6
		0	1
		10	14
		1	0
		8	12
		9	13
		1	1
		11	15
		0	0
		1	1
		1	0
		0	1
		2	3
		0	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
		0	1
		2	3
		0	0
		1	1
		10	11
		1	0
		8	9
		1	1
		1	0
</			

Problem 2.5

In this problem, you will design a combinatorial circuit. You will start from an informal specification and end with a hardware realization.

Recall the definition of Binary Coded Decimal (BCD) numbers from problem 1.7 (last week's homework).

You have as inputs four wires W , X , Y , and Z , which, together, comprise a single BCD digit (e.g. if $W=0$ $X=1$ $Y=0$ $Z=1$, you have the BCD digit 5).

From these four inputs, your circuit must produce two outputs.

Output **A** will be high if the BCD digit is a composite number (i.e. a number that can be expressed as a product of two or more prime numbers). Note that 1 is not a prime number - the first prime numbers are 2, 3, and 5.

Output **B** will be high if the BCD digit is a perfect square. Note that 0 and 1 are both perfect squares, but neither is composite.

You may assume that only valid BCD numbers will be represented (e.g. you will never have 1110 as an input).

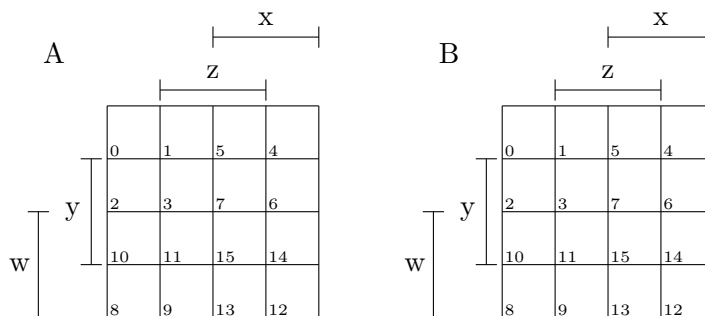
Finally, the circuit you create cannot include more than three OR gates (2-input) and four AND gates (2-input or 3-input). You may use as many inverters (NOT gates) as your heart desires, but may want to limit yourself to however many you feel are necessary to complete your circuit.

(Continued on next page)

- a. Fill in the truth tables below. Indicate any “don’t care’s” with x ’s.

#	W	X	Y	Z	A	B	#	W	X	Y	Z	A	B
0	0	0	0	0			8	1	0	0	0		
1	0	0	0	1			9	1	0	0	1		
2	0	0	1	0			10	1	0	1	0		
3	0	0	1	1			11	1	0	1	1		
4	0	1	0	0			12	1	1	0	0		
5	0	1	0	1			13	1	1	0	1		
6	0	1	1	0			14	1	1	1	0		
7	0	1	1	1			15	1	1	1	1		

- b. Fill in the Karnaugh maps for the circuit’s outputs. Note that x ’s should be grouped in boxes if this results in larger boxes. (If you redraw the Karnaugh maps, make sure to keep the same format as below¹)



- c. Based on your Karnaugh maps, write a minimal sum of products expression for the circuit.
- d. Draw a circuit corresponding to your expression, keeping in mind the limitations on the number of gates allowed.
- Label your four inputs and two outputs.
- Hand in a separate diglog file for this.

¹or else!

Problem 2.6

Build a two-to-one multiplexer using only NAND gates, input buttons and output lights.

Your circuit should have 2 data inputs (A_0 and A_1), one control wire (labeled **control**), and one output (labeled **output**).

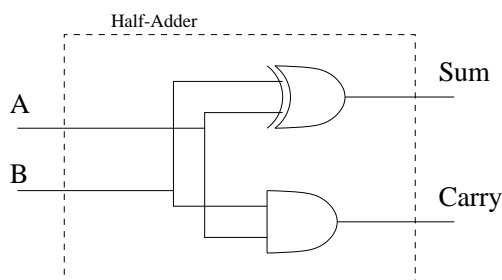
Hand in a separate diglog file for this problem.

Problem 2.7

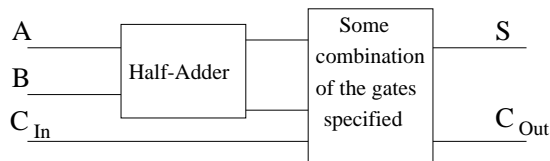
Ron is stuck in the Engin lab attempting to build a full adder but he's on the verge of crying because he doesn't have all the supplies he thinks he needs (he is not as familiar with muggle inventions as his father is). He was given a half-adder, plus one 2-input AND gate, one 2-input OR gate, one 2-input XOR gate, and one 2-input NAND gate. (**Note: no inverters are allowed**)

Help him build a full adder using some combination of the gates above (not necessarily all of them).

First, build the half-adder pictured below and verify that it works. (The gates used here don't count against the quotas above.)



Add the third input and build a working full-adder.



Label the three inputs (A, B, and C_{in}), and label the output (output).

Hand in a separate diglog file for this problem.