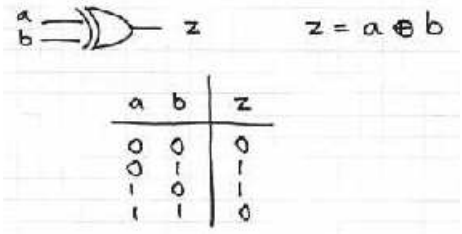
Digital Hardware
Problems 1

Mike Spivey, Hilary Term 2004

1 An exclusive OR (XOR) gate has the following truth table:

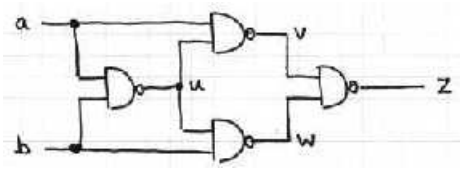


(i) Show that $\oplus$ is associative and commutative. Does it have an identity element?

(ii) Show how to build an XOR gate from a 2-input OR gate, two 2-input AND gates and two inverters.

(iii) Can you still build an XOR gate if one of the two AND gates is replaced by an OR gate?

(iv) Show that the following circuit of four NAND gates also computes $z = a \oplus b$.



2 A full adder has three inputs a, b and cin and two outputs s and $cout$, and satisfies the specification $2 \times cout + s = a + b + cin$ (where + denotes integer addition).

(i) Draw up a truth table for the full adder like this:

a	b	cin	s	$cout$
0	0	0	0	0
...	...		...	

(ii) Show how to implement s using two XOR gates.

(iii) Use a Karnaugh map to find a good implementation of *cout* using NAND gates.

3 An AND–OR–INVERT gate has three inputs a, b and c, and computes the output $z = \neg((a \wedge b) \vee c)$. Show how to implement this gate using six transistors.

4 A circuit has four inputs a, b, c, d and two outputs x, y. If some or all of the inputs are 1, then xy in binary gives the index of the first input that is a 1: thus $xy = 00$ if $a = 1$; $xy = 01$ if $a = 0$ and $b = 1$; etc. If $a = b = c = d = 0$ then the output is undefined. Use Karnaugh maps to find minimal sum-of-products expressions for x and y, exploiting don't-cares as appropriate.

5 Design a circuit that inputs two 2-bit binary numbers and outputs their product, also as a binary number. Simplify the output expressions as much as possible, using XOR gates if they help.

6 De Morgan's laws are

$$\neg(a \wedge b) = \neg a \vee \neg b$$
$$\neg(a \vee b) = \neg a \wedge \neg b$$

(i) Use truth tables to check the validity of these laws.

(ii) Draw a Karnaugh map for the expression

$$z = (a \wedge \neg b) \vee (a \wedge \neg c \wedge \neg d) \vee (\neg b \wedge c) \vee (\neg a \wedge c \wedge d).$$

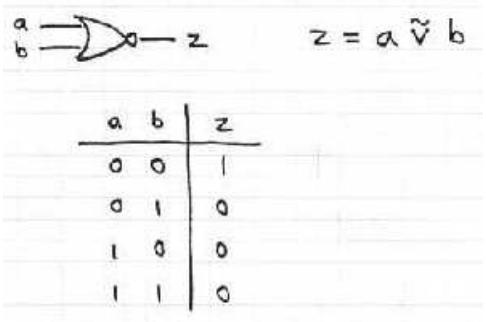
(iii) Interchange 0's and 1's to draw a Karnaugh map for $\neg z$.

(iv) Identify minimal product terms and express $\neg z$ as a sum of products.

(v) Use de Morgan's laws to express z as a product of sums.

(vi) Explain how to use Karnaugh maps to express functions in product-of-sums form directly.

(vii) A NOR gate has the following truth table



Multi-input NOR gates are defined similarly, as for multi-input NAND gates. Show how to implement z using only NOR gates.

7 Design a circuit that outputs (as a 2-bit binary number) the total number of 1 bits among its n inputs reduced modulo 3. (Thus if the circuit had 8 inputs, all 1, then the output would be $10_2 = 2$.) Try to ensure that the combinational delay of your circuit grows only slowly as n gets large.