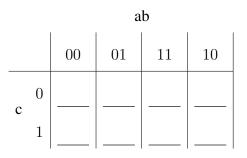# Computer Mathematics

Week 8 Examples

1. The *majority* function has three inputs. The output is true if two or more of the inputs are true.

(1) Draw the truth table for the *majority* function.

a	b	c	majority(a, b, c)
0	0	0	____
0	0	1	____
0	1	0	____
0	1	1	____
1	0	0	____
1	0	1	____
1	1	0	____
1	1	1	____

(2) Use a Karnaugh map to simplify the truth table of the *majority* function.

ab

		00	01	11	10
c	0	____	____	____	____
	1	____	____	____	____

(3) Draw the logic circuit that implements the *majority* function.

(4) What use can you think of for the *majority* function? What use can you think of for the *minority* function (whose output is true when the majority of input bits are zero)?

2. Combining AND with NOT creates a NAND gate. NAND gates can be used to implement any logic function. (This is useful because some semiconductor technologies are most efficient when only one kind of gate is used.)

a	b	$\overline{ab}$
0	0	1
0	1	1
1	0	1
1	1	0

NAND gate

Using <u>only</u> NAND gates, draw logic circuits that implement the following basic Boolean operations:

NOT   AND   OR

3. Using <u>only</u> NAND gates, draw a logic circuit that implements XOR (exclusive OR).

4. Using <u>only</u> NAND gates, draw the logic circuit of a *half* adder: { C, S } = A + B

5. Prove, using the simplest argument you can think of, that all of the examples above could also be implemented using <u>only</u> NOR gates.