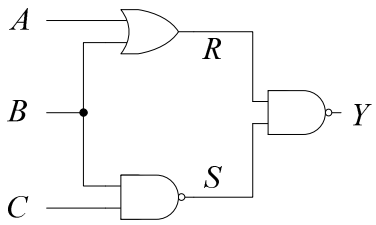
Question 1. Boolean algebra. **[6 marks]**



(a) [2 marks] Consider the circuit above, write expressions for *R* and for *S*.

R=A+B
S=(B.C)'

Some of you interpreted the small dot near the B input as an inverter. This is incorrect, that is simply a junction to indicate a connection between two wires. Inversion operations are always open circles and are never floating on their own; they are always "attached" to a gate.

(b) [1 mark] What is *Y* as a function of *R* and *S*?

Y=(R.S)'

(c) [1 mark] Substitute your results from part (a) into that from part (b) to obtain an expression for *Y* as a function of *A,B* and *C*.

Y=[(A+B).(B.C)']'
Some of you wrote [A+B(R.S)']' or something like it. It is important to include the parentheses around the (A+B) term. Otherwise you are calculating B.(RS)' and then OR'ing the result with A.

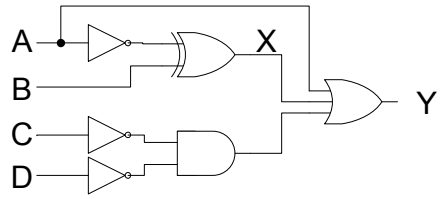
(d) [2 marks] Use DeMorgan's rule to show that $Y = \overline{A}\overline{B} + BC$.

Y=[(A+B)(B.C)']'
=(A+B)' + (B.C)'' By DeMorgan's rule
=A'B' + BC The first term by DeMorgan's rule.
The second term is because A''=A

Several of you tackled by starting with an expression (sometimes wrong), writing some algebra (sometimes wrong) and then simply stating that this was equal to the required result. Do not do this. Your markers do follow the logic of your entire answer and trying to trick them does not impress.

Question 2. Constructing a circuit. **[4 marks]**

Draw a circuit that produces a single output $Y = A + X + \overline{C}.\overline{D}$, where $X = \overline{A} \oplus B$.

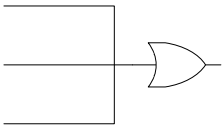


Many of you replaced the discrete inverters with inversion circles on the inputs of the gates, which is fine too.

In some cases I had trouble distinguishing what sort of gate you were attempting to portray. Please draw your gates clearly so we know what you mean. DO NOT attempt the rather hopeful trick of drawing something half way between an AND and an OR gate…

Note that C'.D' is not the same as (CD)'. Many of you incorrectly used a NAND gate to implement the C'D' term, which is incorrect. The following two gates are not equivalent!

Some of you also incorrectly drew the final OR gate as something like the following:



This is not correct; each input to a logic gate must be distinct.

Question 3. Number representation. **[4 marks]**

(a) [1 mark] Express 23_{10} as the sum of powers of two.

$23 = 16+4+2+1 = 2^4+2^2+2^1+2^0$
Either expression was fine.

(b) [1 mark] What is the binary representation of 23_{10}?

$23_{10} = 10111_2$ by reading the power of two in the above expression.

(c) [2 marks] Write the eight-bit 2's complement representation of -23_{10}. Ensure that you show your working and/or explain your conversion technique.

23_{10} is 0001 0111 in an eight bit representation. To form the two's complement, we can follow two methods:
1. Invert all bits and then add one.

0001 0111 -> 1110 1000 (This is the ones complement)
+0000 0001
1110 1001

2. Use the quick method of writing down the number up to and including the first 1 and then inverting all bits, which yields the answer in one step.

Question 4. Adders **[4 marks]**

(a) [2 marks] Draw truth tables for the carry-out and sum outputs of a half-adder having inputs *A* and *B*. Remember that there is no carry-in to a half-adder.

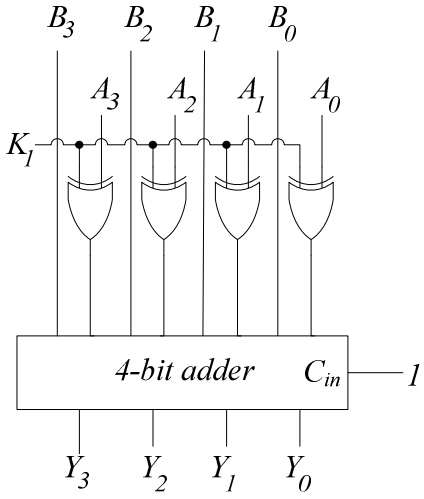
A	B	Carry Out	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Drawing a pair of Karnaugh maps was fine too.
Note that some of you used rather unorthodox orderings of the A and B elements in your table. You will not lose marks for this, but it makes it rather harder to see what is going on, both for you and for your markers.

(b) [2 marks] Use your truth tables to write Boolean expressions for the carry-out and sum outputs.

Cout = A.B
Sum = A XOR B

Question 5. ALU. **[2 marks]**



(a) [1 mark] What is the function of the XOR gates in the partial ALU circuit shown above?

When K1 is low (0) the A vector passes through to the adder unchanged. When K1 is high (1) A is inverted as it passes through the XOR gates. Thus the XOR gates act as a controllable inverter.

(b) [1 mark] What operation does the entire circuit perform when $K_1 = 1$.

When K1 is 1 the XOR gates invert the A signals as described above. As we have inverted the bits of a A and added 1 via the carry in signal we have formed the 2's complement representation of –A.

The circuit then adds this to B, so overall we get Y=B-A.