Week 4

This week we'll investigate how computers work!

Click this link for an interactive demonstration: `https://academo.org/demos/logic-gate-simulator/`

Question 1. The circuits of a computer are built out of some basic building blocks, called "logic gates".

Set up an AND gate, and connect it to two INPUTS (on the left) and an OUTPUT (on the right). What does this circuit do? Fill out this table:

In 1	In 2	Out
Off	Off	
Off	On	
On	Off	
On	On	

Question 2. Try some other gates.

OR

In 1	In 2	Out
Off	Off	
Off	On	
On	Off	
On	On	

NOT

In 1	Out
Off	
On	

NAND

In 1	In 2	Out
Off	Off	
Off	On	
On	Off	
On	On	

XOR

In 1	In 2	Out
Off	Off	
Off	On	
On	Off	
On	On	

Question 3. Connect some gates together to make circuits to match with the following tables:

In 1	In 2	Out
Off	Off	On
Off	On	Off
On	Off	Off
On	On	Off

In 1	In 2	Out
Off	Off	Off
Off	On	Off
On	Off	On
On	On	Off

In 1	In 2	Out
Off	Off	On
Off	On	Off
On	Off	On
On	On	Off

Question 4. Here are some more tables for circuits to build. Writing "off" and "on" all the time is tiring, so instead we'll write "0" and "1" (off = 0, on = 1).

In 1	In 2	In 3	Out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

In 1	In 2	In 3	Out
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

In 1	In 2	In 3	Out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Question 5. Can you build…

(a) a NOT gate, using only NAND gates?
(b) an AND gate, using only NAND gates?
(c) an OR gate, using only NAND gates?
(d) a NAND gate, using only AND gates?
(e) an OR gate, using only NOT gates?
(f) an OR gate, using only XOR gates?

Question 6. Design a *latch.* This circuit should initially have an output of Off. When the input is turned On, the output of the circuit should become On. But then when the input is turned Off, the output should stay On forever.

Can you modify your latch design so that a second input lets you turn the latch off again?

Let's build an addition circuit! Remember, we're using 0 and 1 to represent "off" and "on", so we can think of these as representing binary digits.

Question 7. We'll start by designing a circuit to add two one-digit binary numbers. The maximum number we could get by doing this is 10 (in binary, representing 2 in decimal), so we need two output digits. First, fill out this table for how the circuit should behave:

In number 1	In number 2	Out digit 1	Out digit 2
0	0		
0	1		
1	0		
1	1		

Now, design a circuit that matches the table.

Question 8. We can now add two one-digit numbers. Let's increase the difficulty by adding *three* one-digit binary numbers. First, fill out this table:

In 1	In 2	In 3	Out 1	Out 2
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

And now design a circuit.

Question 9. Now, try to design a circuit to add two two-digit numbers!

(Hint: You can re-use your designs for adding two or three single-digit numbers, if you need to.)

Can you extend your idea to be able to add a pair of three-digit numbers, or four-digit numbers?

We've figured out addition — what about multiplication?

Question 10. Design a circuit to multiply a six-digit number by a one-digit number.

Question 11. How would you design a circuit for multiplying a six-digit number by a two-digit number? (You know how to add numbers now, so if you like, you can pretend you have an "adding gate" that handles addition for you.)

Question 12. How would you build a circuit to divide numbers? Don't actually try to make this circuit, these problems are getting complicated! Instead, think about what kind of steps you would need to take, and how you might design circuits to do them. It might help to try doing some long division in binary by hand.

If you want to work on more challenges like these, check out `https://nandgame.com`*!*