University of
Nottingham

Electromechanical devices MM2EMD

Lecture 2- Figuring out what electronic circuits do and making electronics remember past events.

Dr. Roderick MacKenzie
roderick.mackenzie@nottingham.ac.uk

Summer 2015

@rcimackenzie

Outline of the lecture

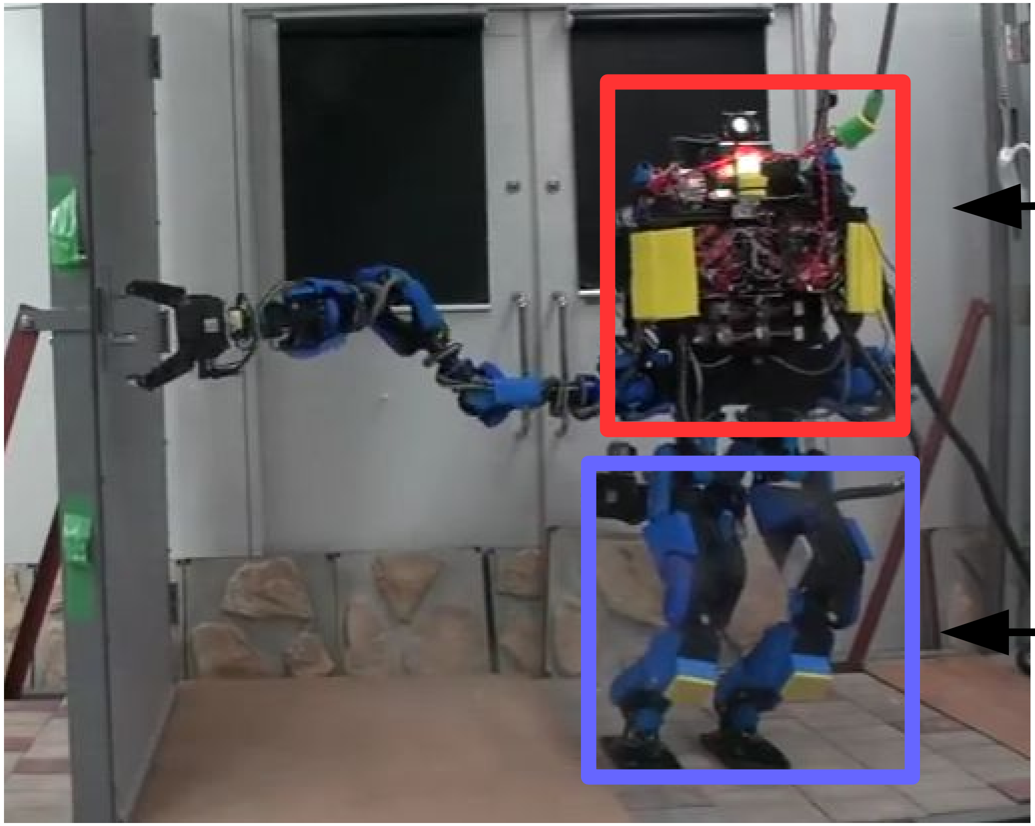
- **Recap of last lecture**
- Recap of logic gates - Mini quiz
- Figuring out what electronic circuits do
- Making electronics remember things
    - Flip flops
    - Serial to parallel converters

Alureiter from wikipedia

- We learnt that electr**ical** engineering is to do with **large voltages** and **currents**.
- Such as those used to drive this motor on a ship. ~500V, ~100A

- We then learnt that electr**onic** engineering is about designing smart **low voltage** decision making circuits to make your device **smart**.

Recap: Electronic engineering produces the brains of you machine.



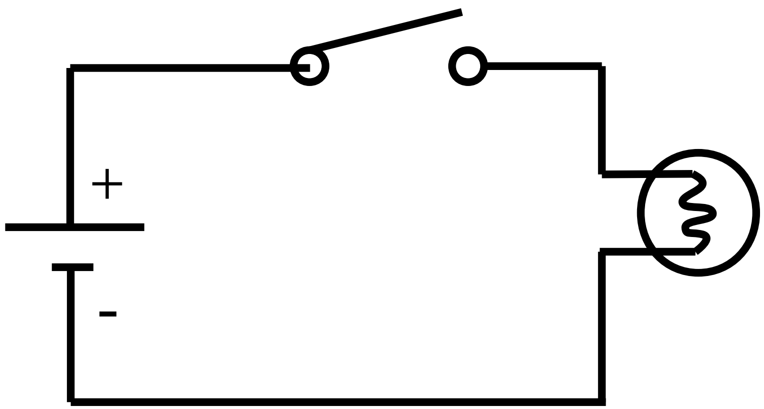
Smart low voltage electronic Circuits (low voltage)

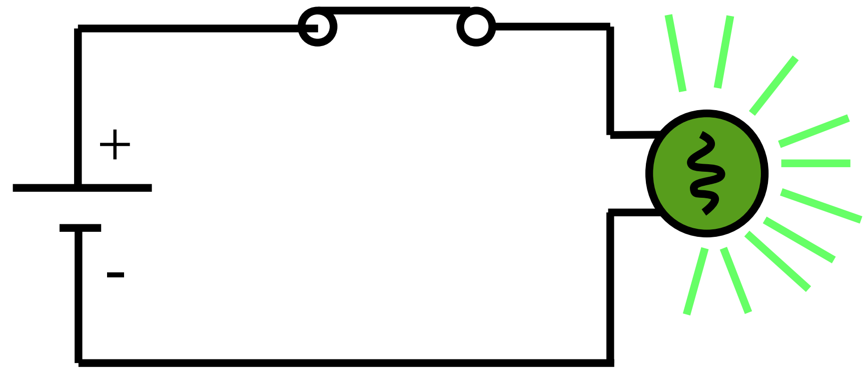
driving

Simple high voltage/current electrical Circuits

Recap: Representing numbers in electronics

•We learnt that we can represent a 1 by switching the circuit on and a 0 by switching a circuit off....





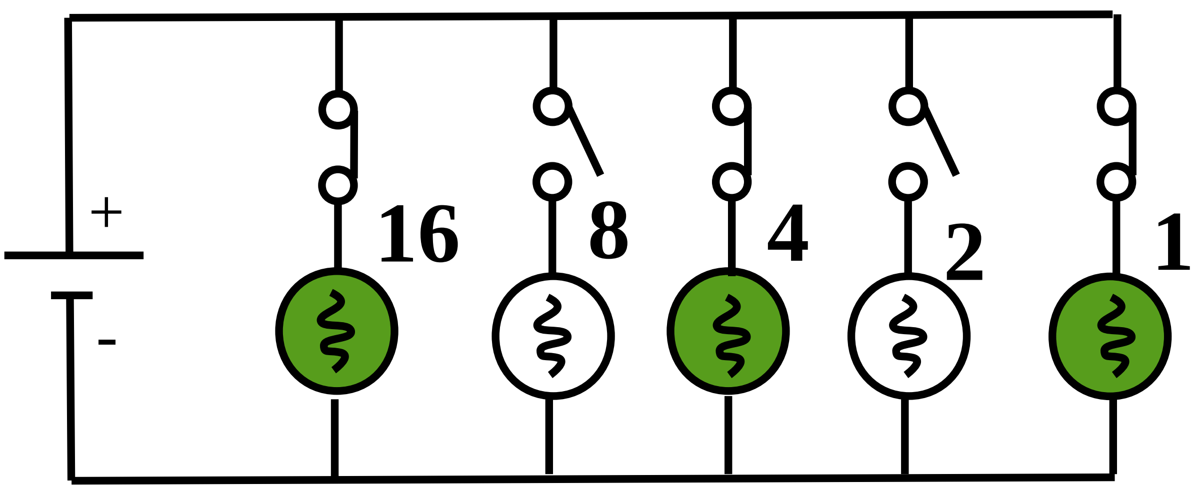
Off = 0 **On = 1**

•This is why on/off switches have 1s and 0s on them.

g4ll4is

Recap: Numbers in electronics

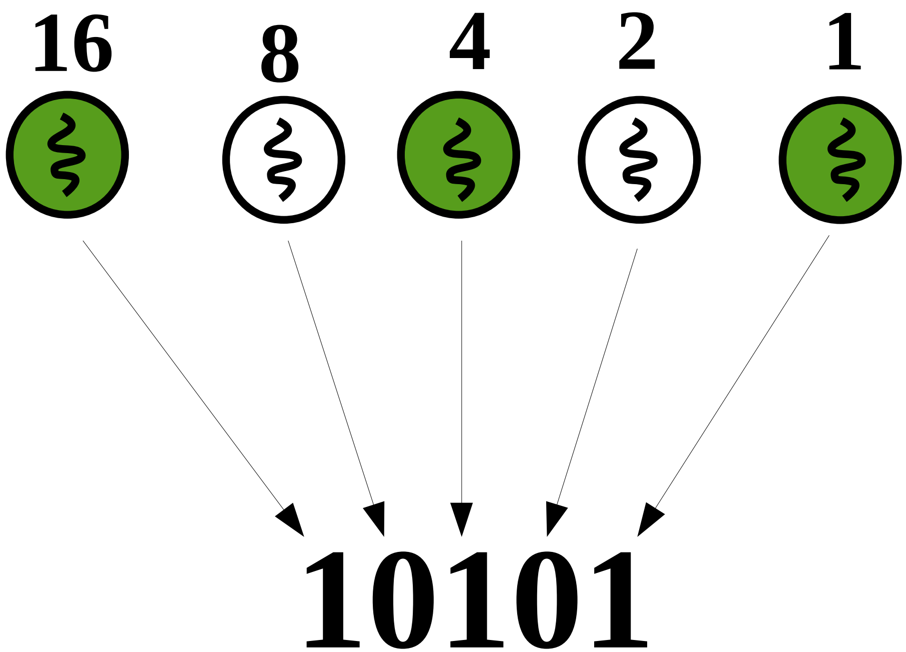
- We then learnt that by using multiple on/off signals we can represent any number.



16+4+1=21

- Representing data in this way is called binary notation.

Recap: Binary numbers



- When writing binary numbers on paper we can represent these on/off signals using 1's and 0's.

- These are called binary numbers.

Recap: We then learnt to count in binary..

Binary number

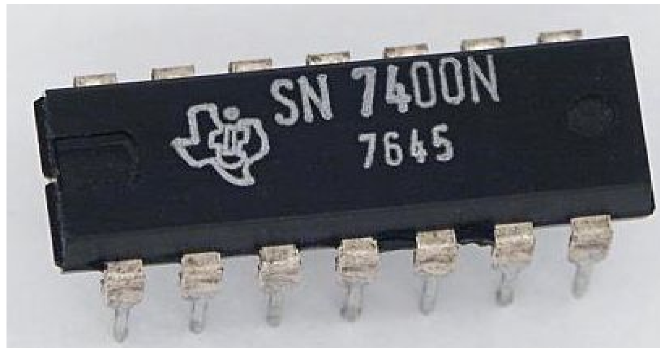
Number	16	8	4	2	1	
0	0	0	0	0	0	=0+0+0+0+0
1	0	0	0	0	1	=0+0+0+0+1
2	0	0	0	1	0	=0+0+0+2+0
3	0	0	0	1	1	=0+0+0+2+1
4	0	0	1	0	0	=0+0+4+0+0
5	0	0	1	0	1	=0+0+4+0+1
						=0+0+4+2+0

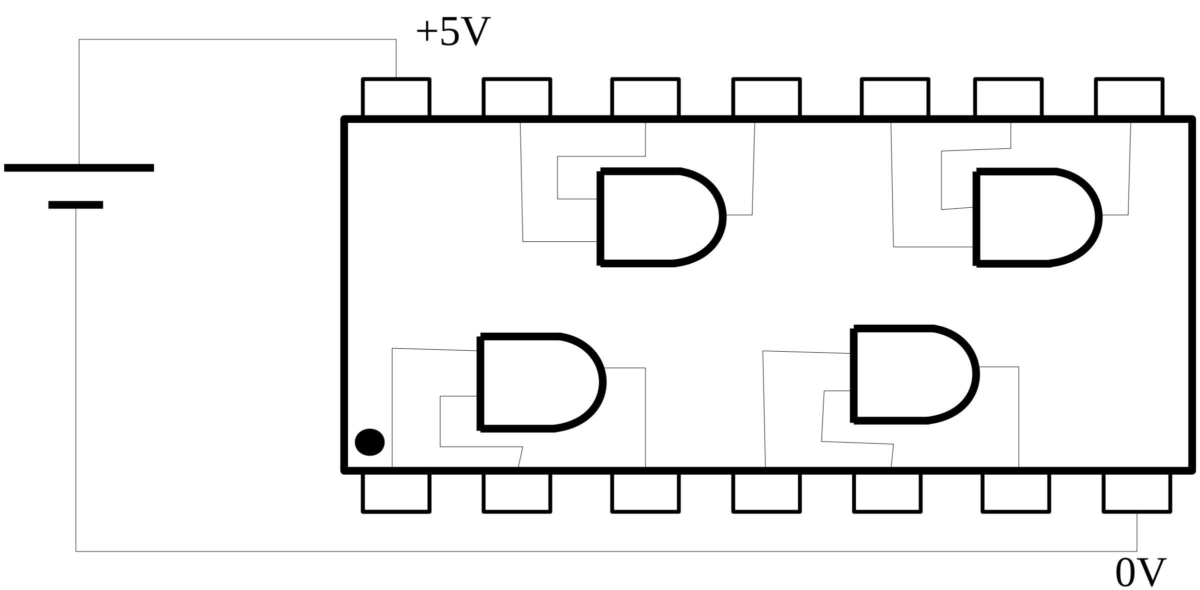
You all got the hang of this and ate all my muffins!

Outline of the lecture

- Recap of last lecture
- **Recap of logic gates - Mini quiz**
- Figuring out what electronic circuits do
- Making electronics remember things
    - Flip flops
    - Serial to parallel converters

Recap: Logic gates look like this

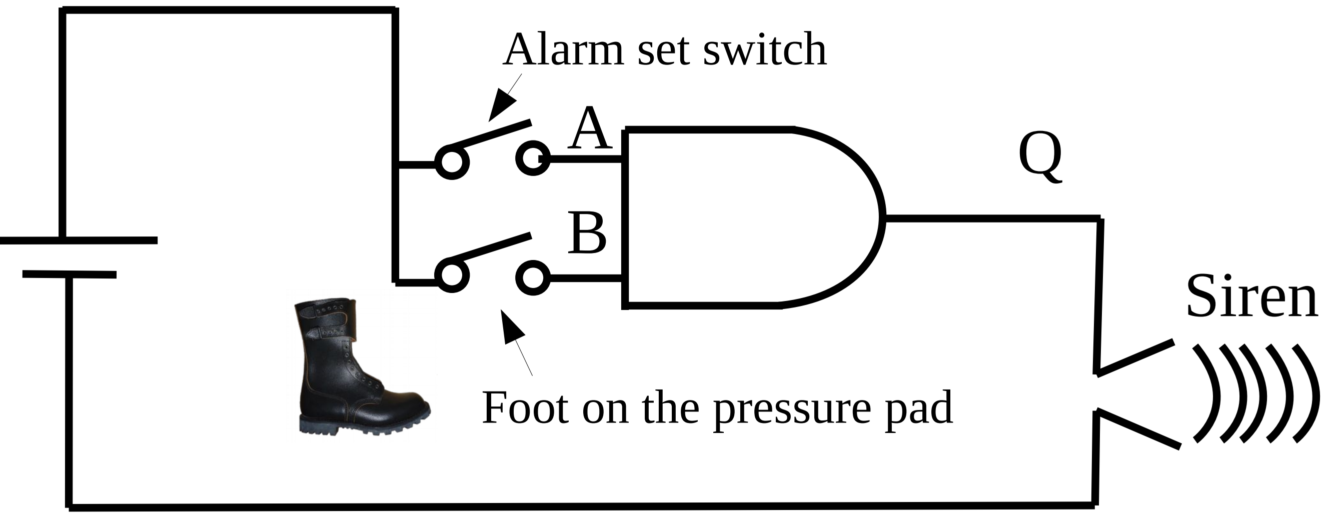




Recap: The AND gate

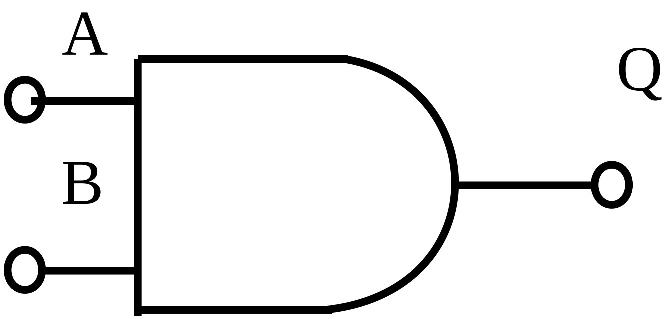
An AND gate will only activate the output when all inputs are 1.

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1



Recap: Mini quiz – AND gate.

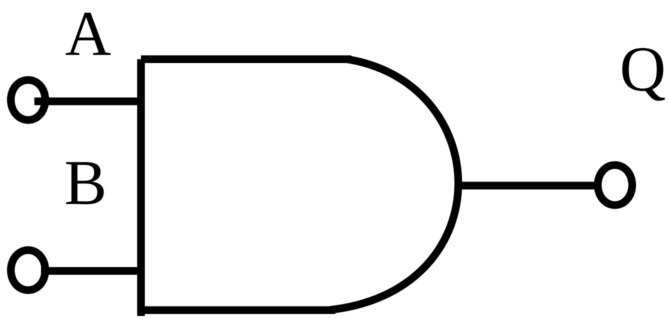
Without looking at the lecture notes fill out the truth table for the AND gate.



A	B	Q
0	0	
0	1	
1	0	
1	1	

Recap: Mini quiz – AND gate.

Without looking at the lecture notes fill out the truth table for the AND gate.

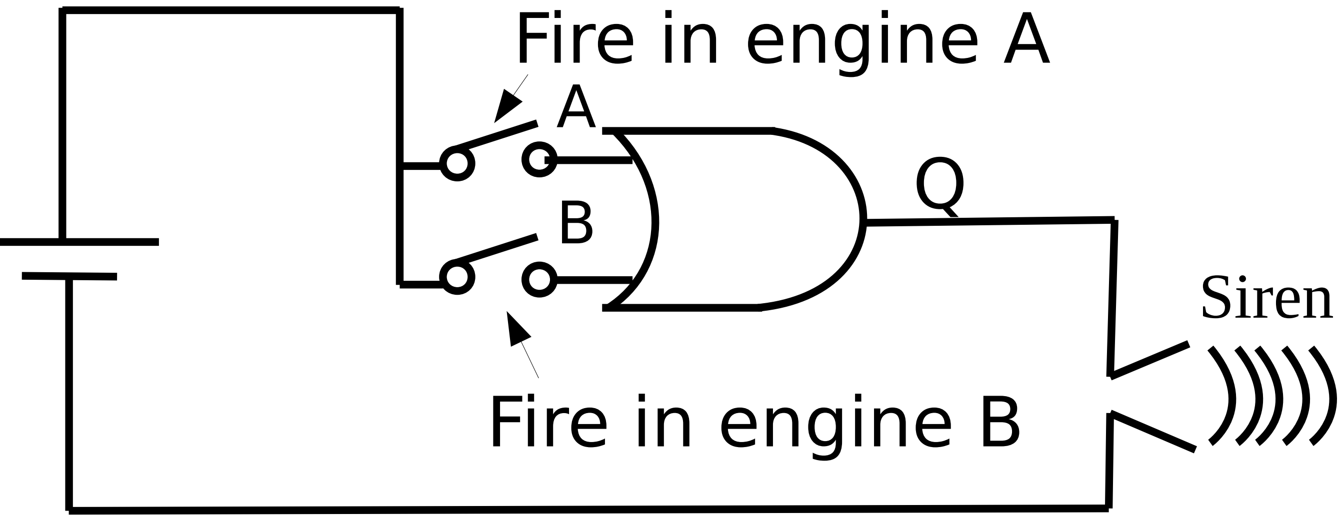


A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

Recap: The OR gate

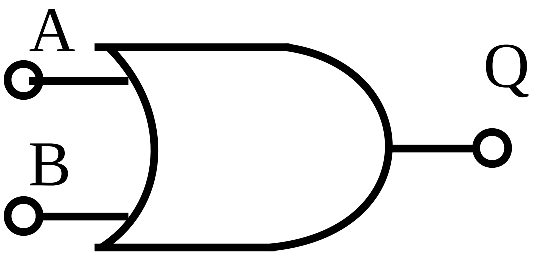
The OR gate will activate the output when only one input is 1.

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1



Recap: Mini quiz – OR gate.

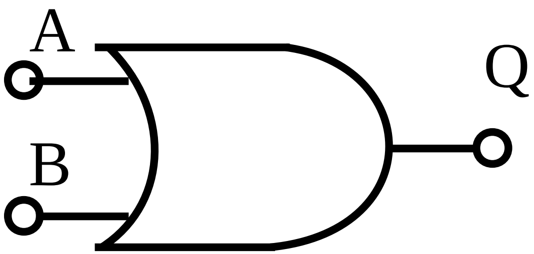
Without looking at the lecture notes fill out the truth table for the OR gate.



A	B	Q
0	0	
0	1	
1	0	
1	1	

Recap: Mini quiz – OR gate.

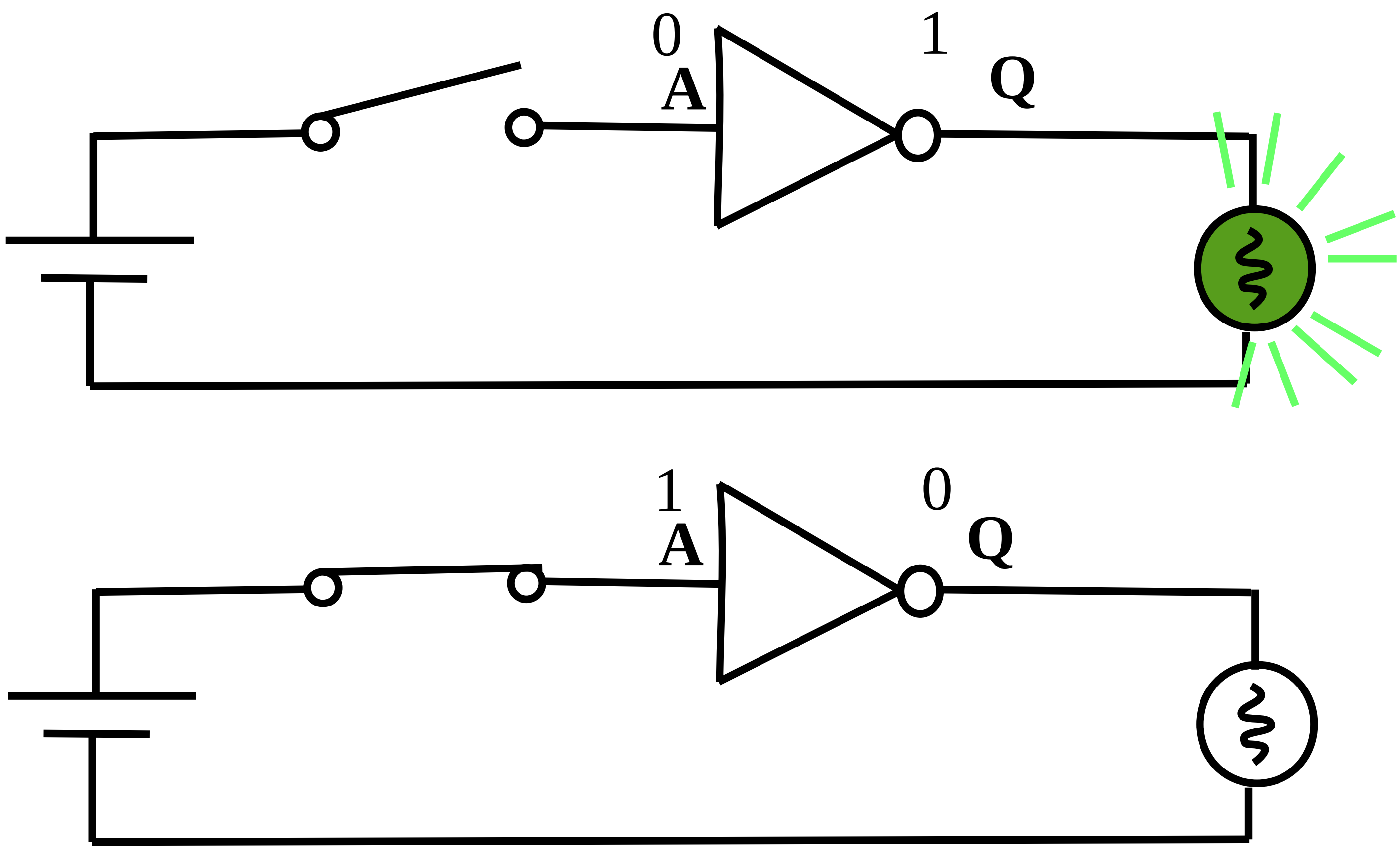
Without looking at the lecture notes fill out the truth table for the OR gate.



A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

Recap: The NOT gate

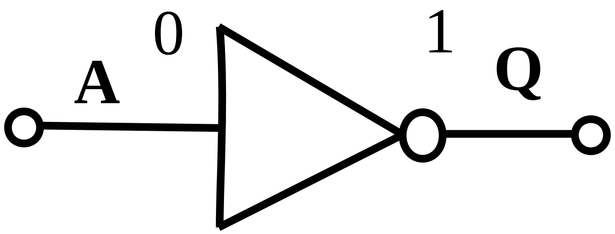
The NOT gate will only activate the output when then input is 0.



A	Q
1	0
0	1

Recap: Mini quiz – NOT gate.

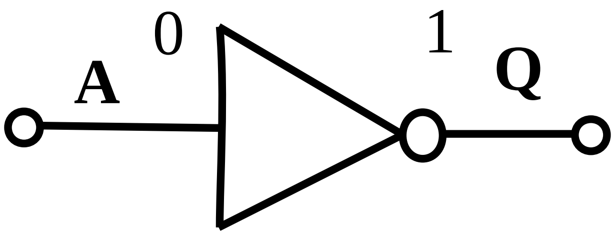
Without looking at the lecture notes fill out the truth table for the NOT gate.



A	Q
1	
0	

Recap: Mini quiz – NOT gate.

Without looking at the lecture notes fill out the truth table for the NOT gate.



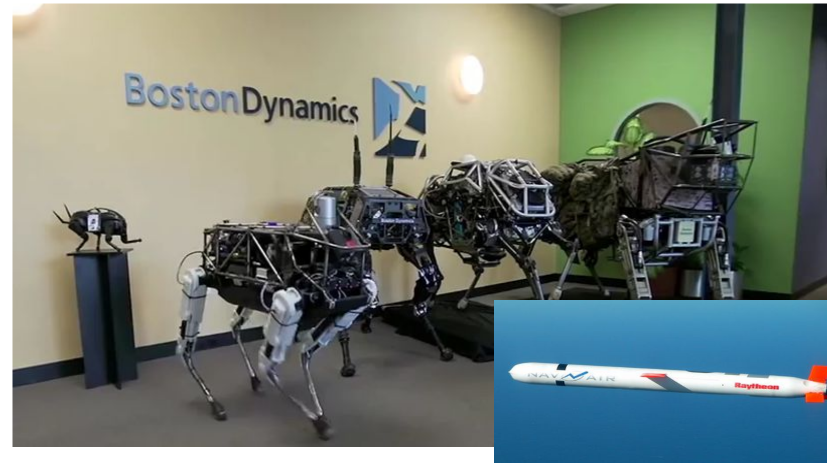
A	Q
1	0
0	1

Outline of the lecture

- Recap of last lecture
- Recap of gates - Mini quiz
- **Figuring out what electronic circuits do**
- Making electronics remember things
    - Flip flops
    - Serial to parallel converters

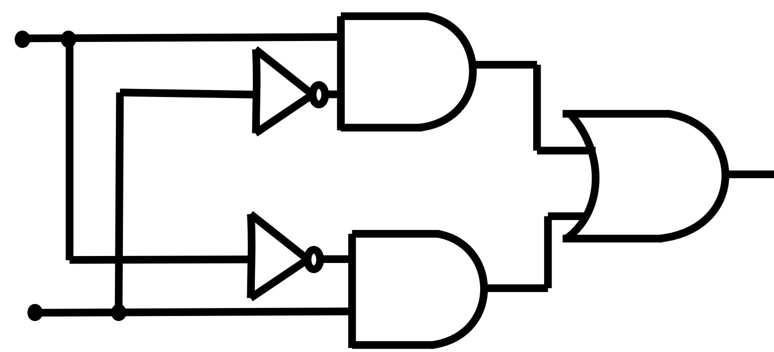
Combining logic gates

- We have used single gates to make simple circuits such as **bugler alarms** and **fire detection systems**.



Video

- To make more interesting circuits which can make **robots walk** or **guide missiles to targets** we need to **combine lots of gates together** so we can do more complex tasks.



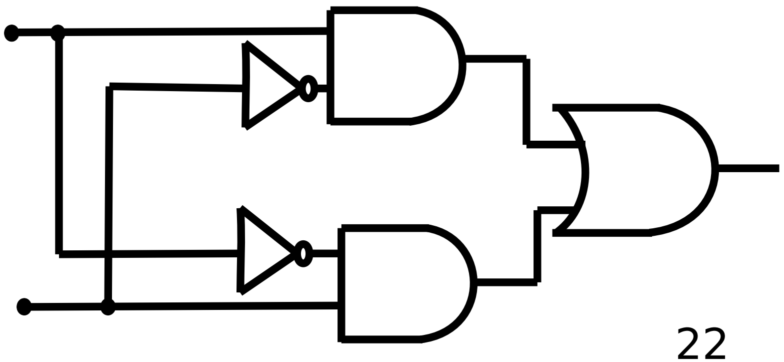
- Lots of gates joined together

Designing complex circuits is a little tricky so....

- **Designing** large digital circuits with lots of gates can be a little tricky.

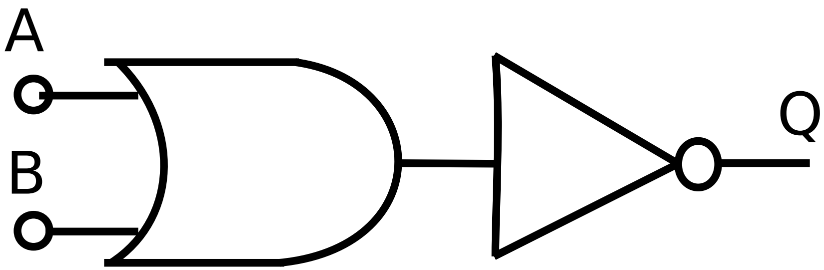
- So let's first start off with **an easier** task first

- I'm going show you how to **analyze any digital** circuit with **lots of gates** and find out what it does.



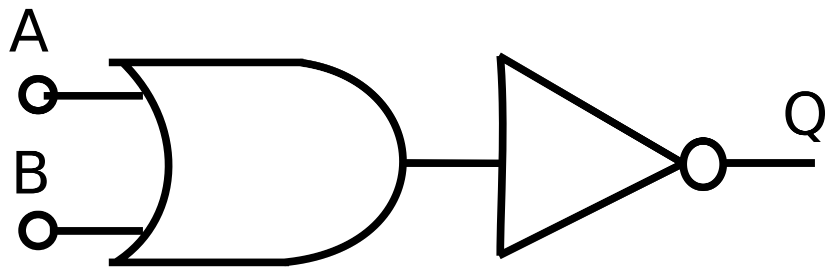
Step 1: Get a picture of the circuit you are interested in

- Then label the inputs and outputs:



Step 2: Draw an empty truth table of the circuit

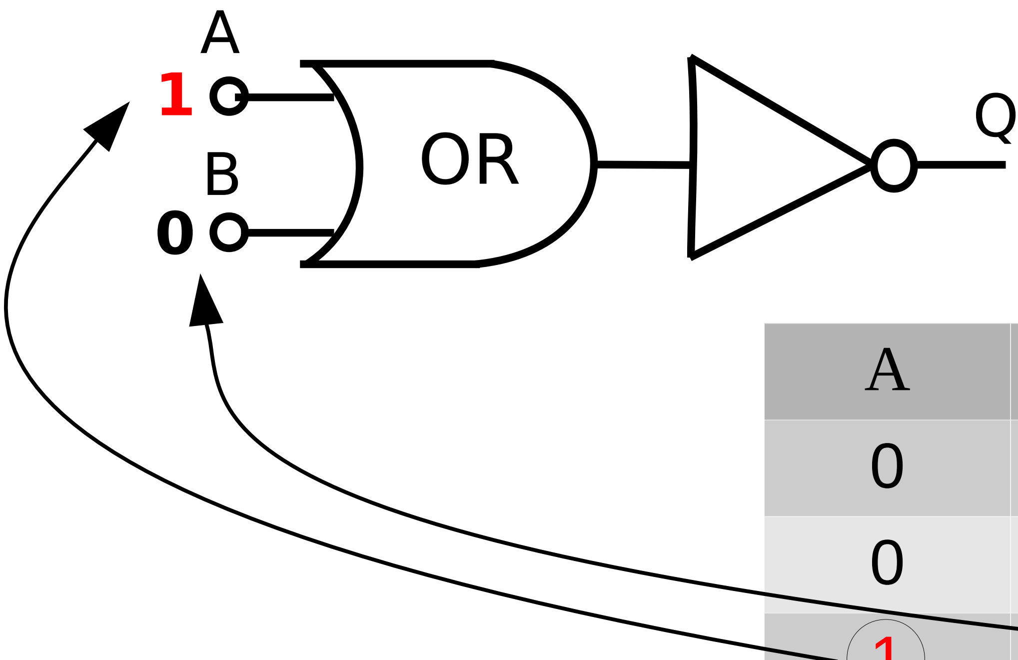
•Draw an empty truth table of the circuit leaving the output column empty.



•Note that columns A and B contain every possible combination of inputs.

A	B	Q
0	0	
0	1	
1	0	
1	1	

Step 2: One line at a time put A and B on the inputs

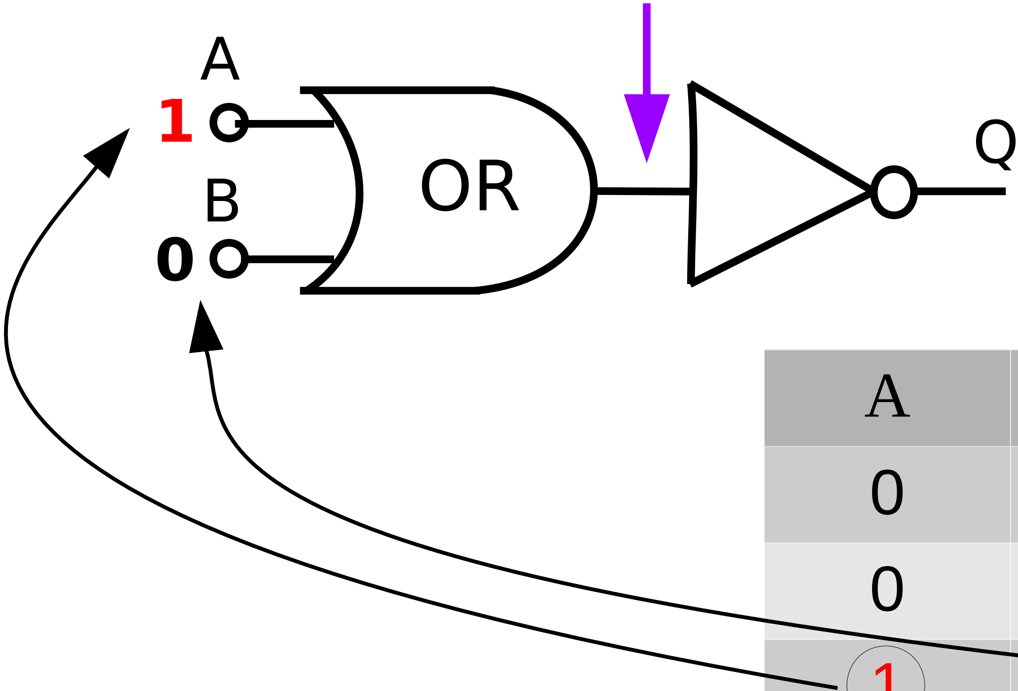


I'm starting off with the third line for this example because it is more interesting.

A	B	Q
0	0	
0	1	
1	0	
1	1	

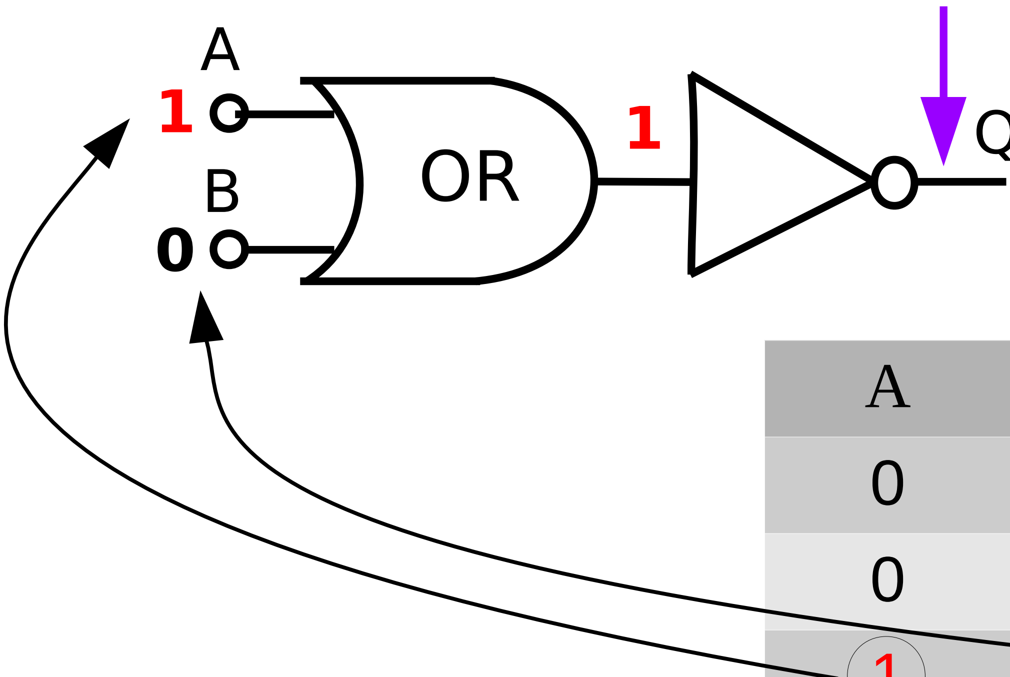
Then propagate the values through the circuit....

What will the output value of the OR gate be ? (at the purple arrow)



A	B	Q
0	0	
0	1	
1	0	
1	1	

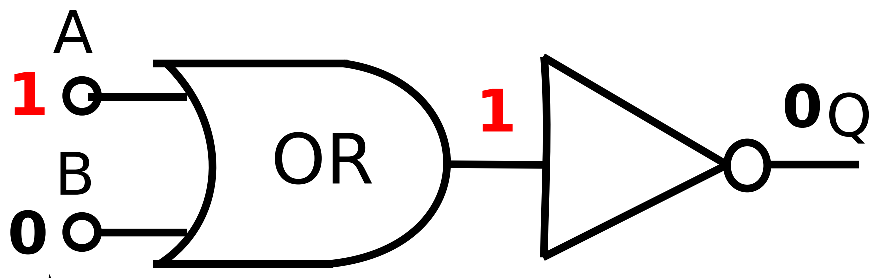
What will the output value of the NOT gate be (at the purple arrow)?



A	B	Q
0	0	
0	1	
1	0	
1	1	

Then propagate the values through the circuit....

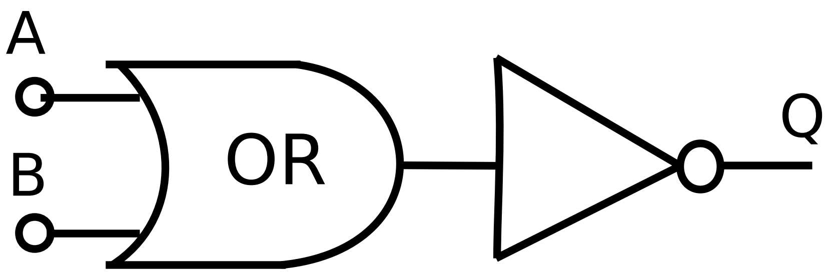
Step 4: Write the value at the output in the truth table.



A	B	Q
0	0	
0	1	
1	0	0
1	1	

Step 4: Write the value at the output in the truth table.

If you do this for 00, 01, 11 you will be able to fill in the entire truth table.



A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

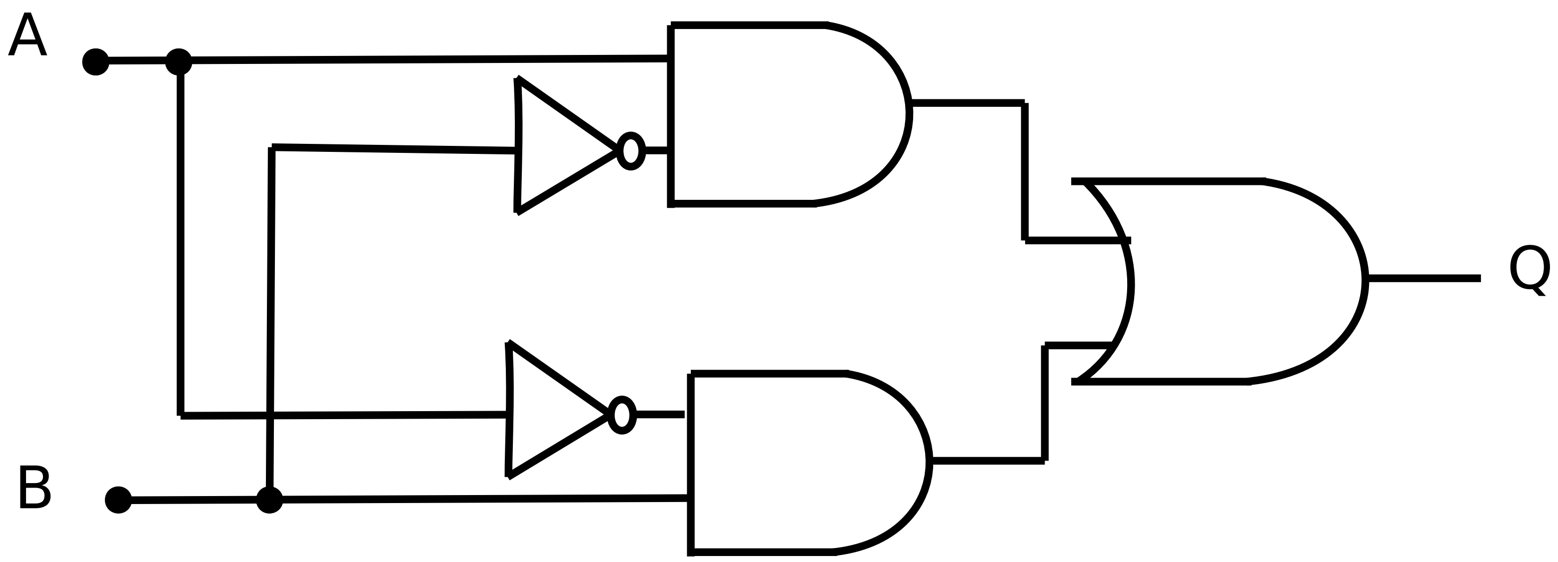
And describe exactly what the circuit does.

Step 4: Write the value at the output in the truth table.

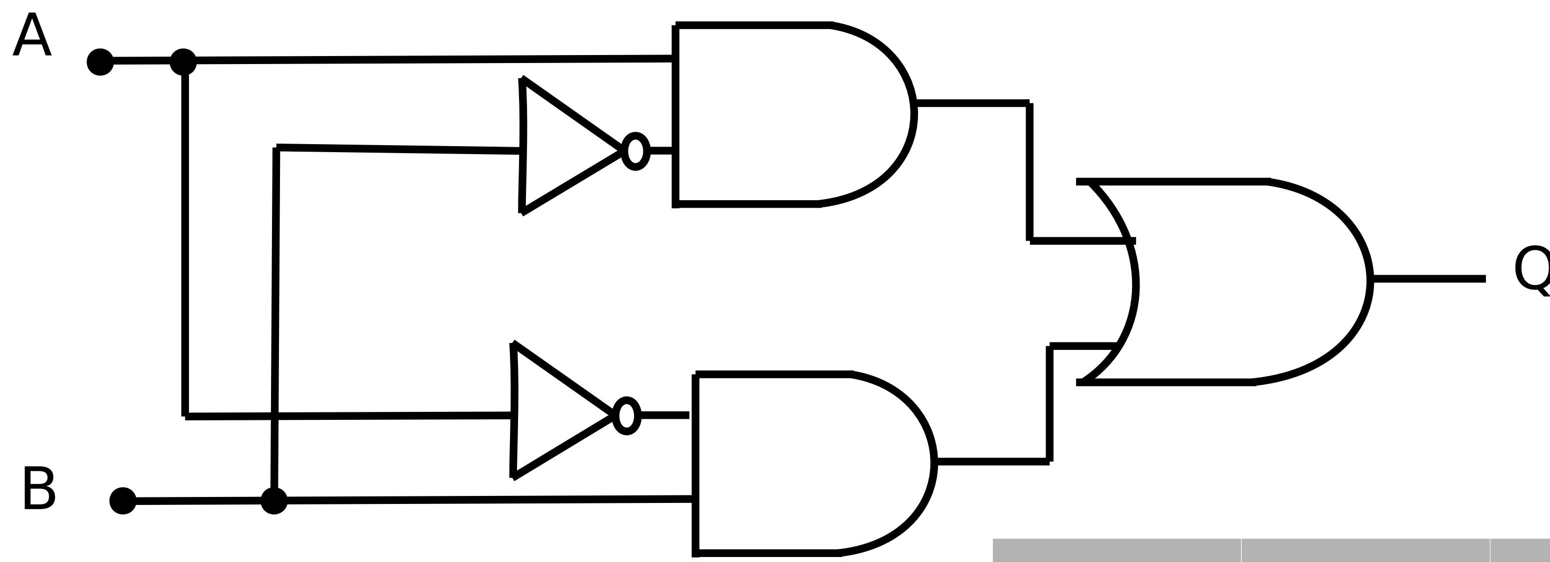
This approach will work for all digital circuits you will come across no matter how complex.

Let's look at something a little more complex......

Combining logic gates a more complex example:

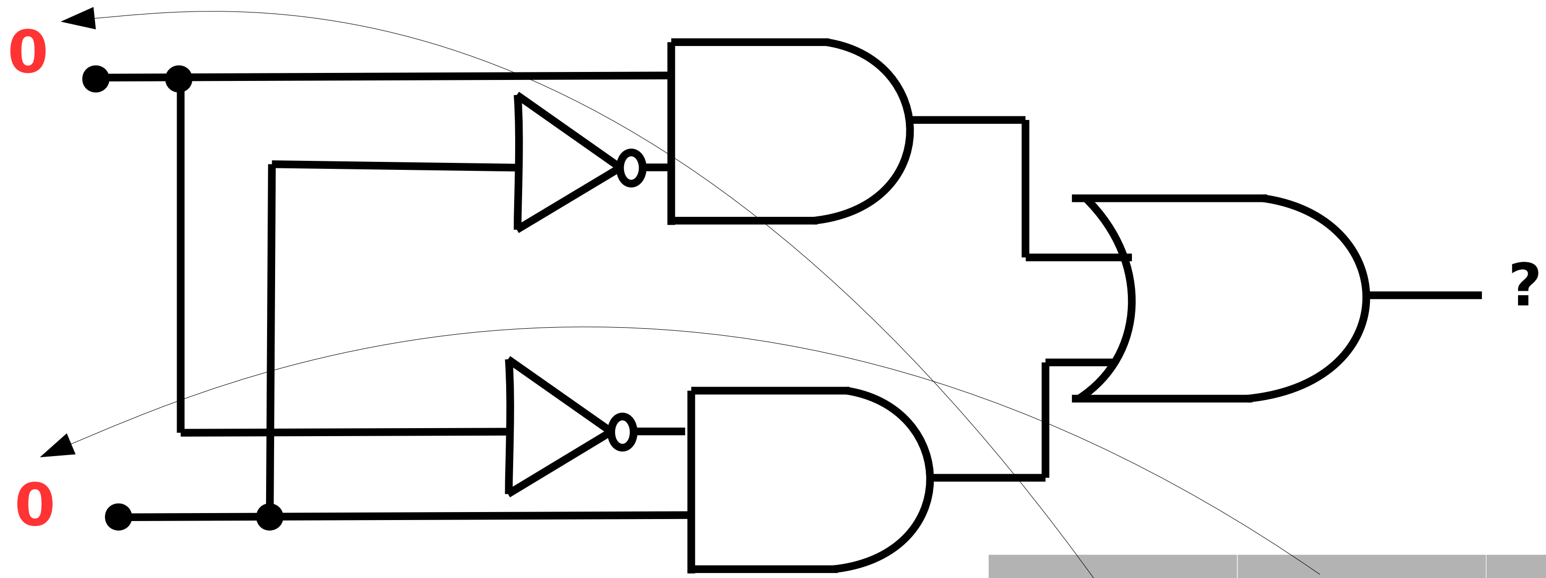


Combining logic gates: Write out the truth table



A	B	Q
0	0	?
0	1	?
1	0	?
1	1	?

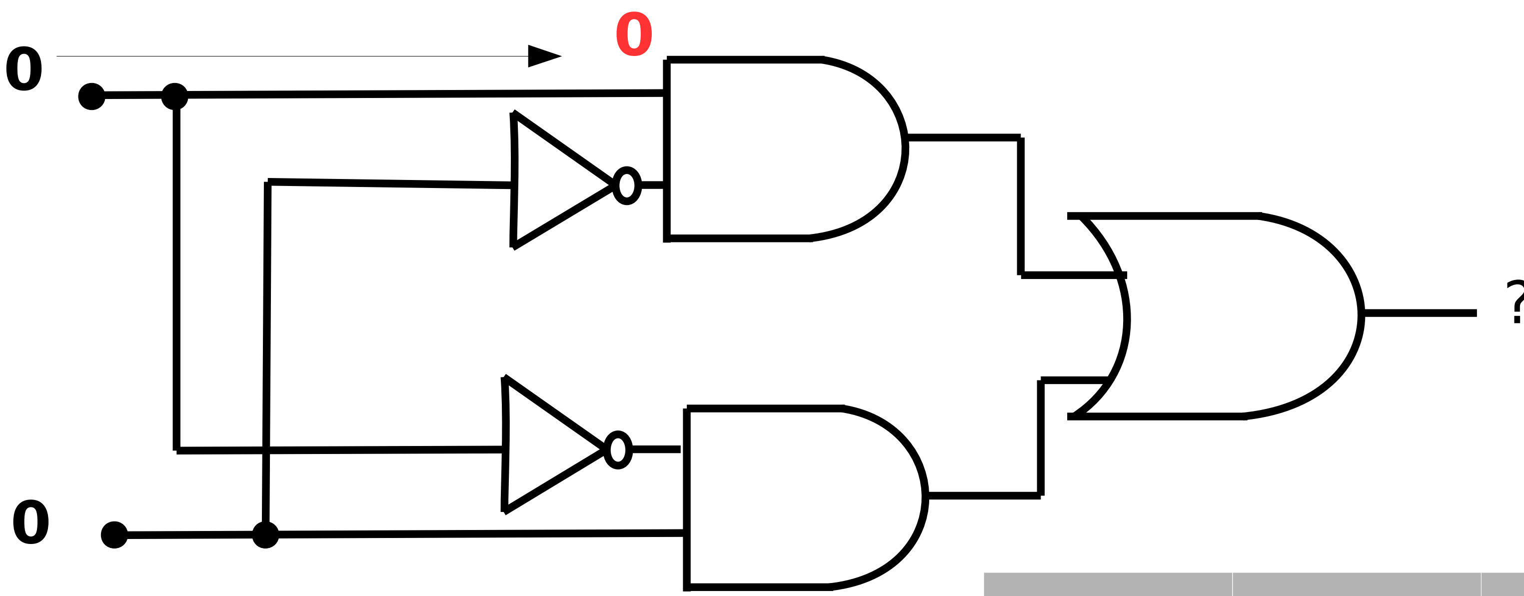
Combining logic gates a more complex example:



Start off by writing the values of the inputs on the diagram.

A	B	Q
0	**0**	?
0	1	?
1	0	?
1	1	?

Combining logic gates a more complex example:

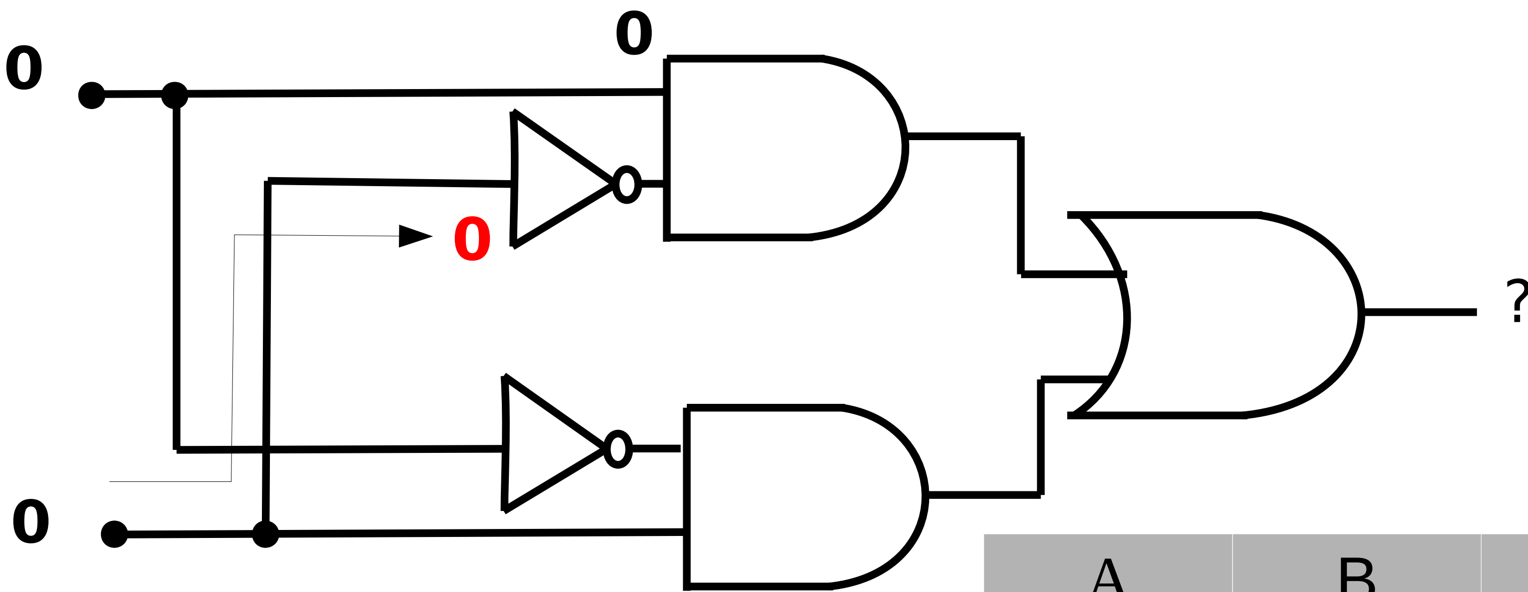


A	B	Q
0	0	?
0	1	?
1	0	?
1	1	?

Then start propagating the numbers through the circuit

Combining logic gates a more complex example:

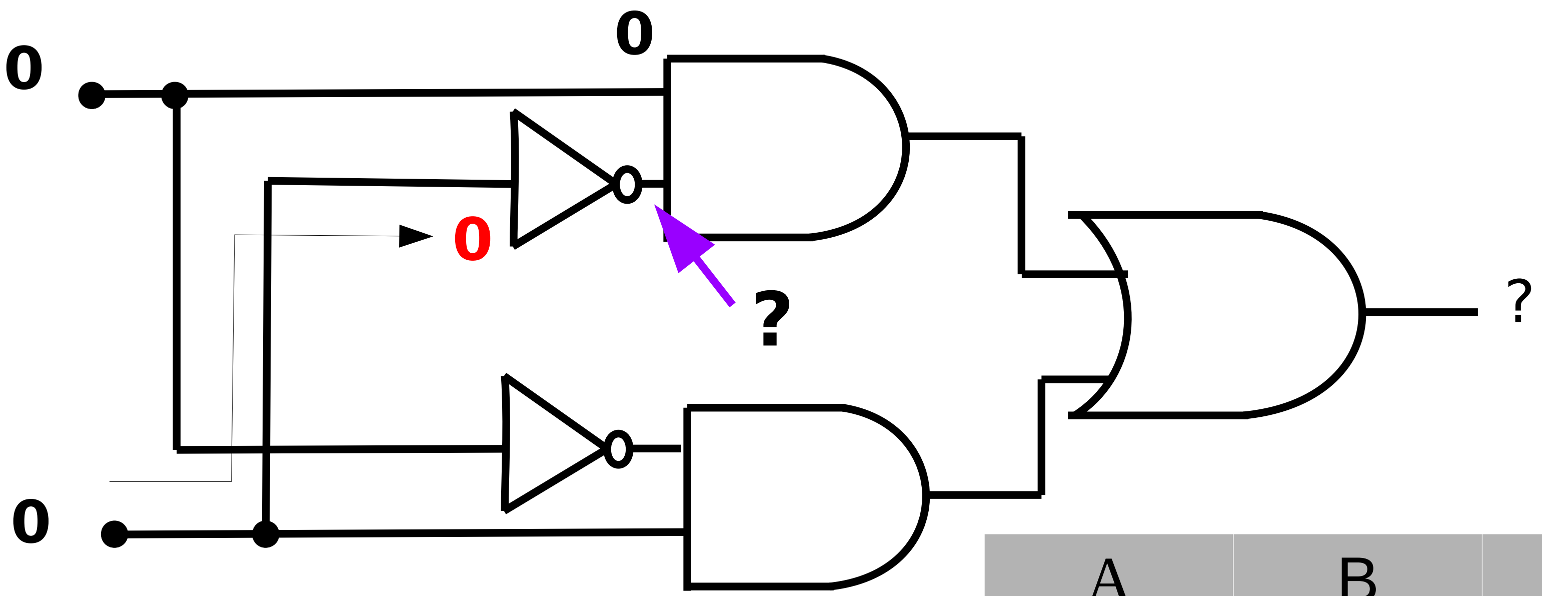
The look at the top NOT gate...



A	B	Q
0	0	?
0	1	?
1	0	?
1	1	?

Combining logic gates a more complex example:

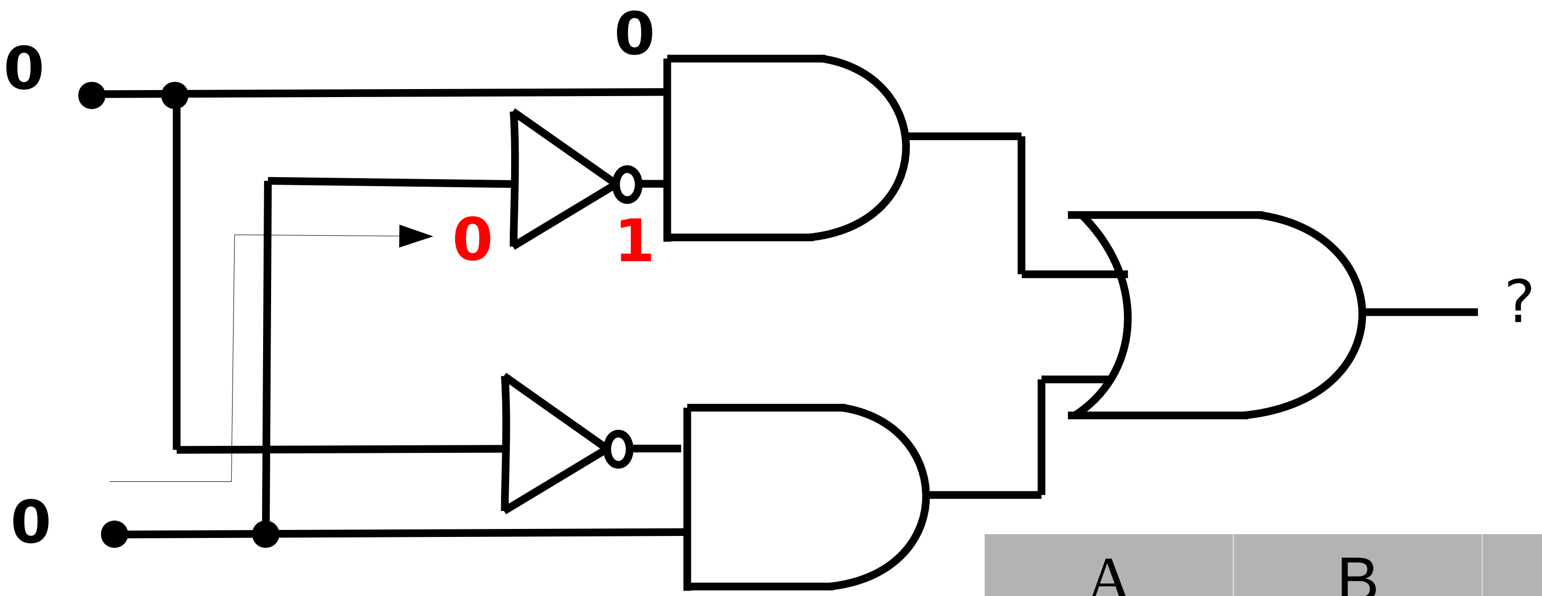
The look at the top NOT gate...



A	B	Q
0	0	?
0	1	?
1	0	?
1	1	?

Combining logic gates a more complex example:

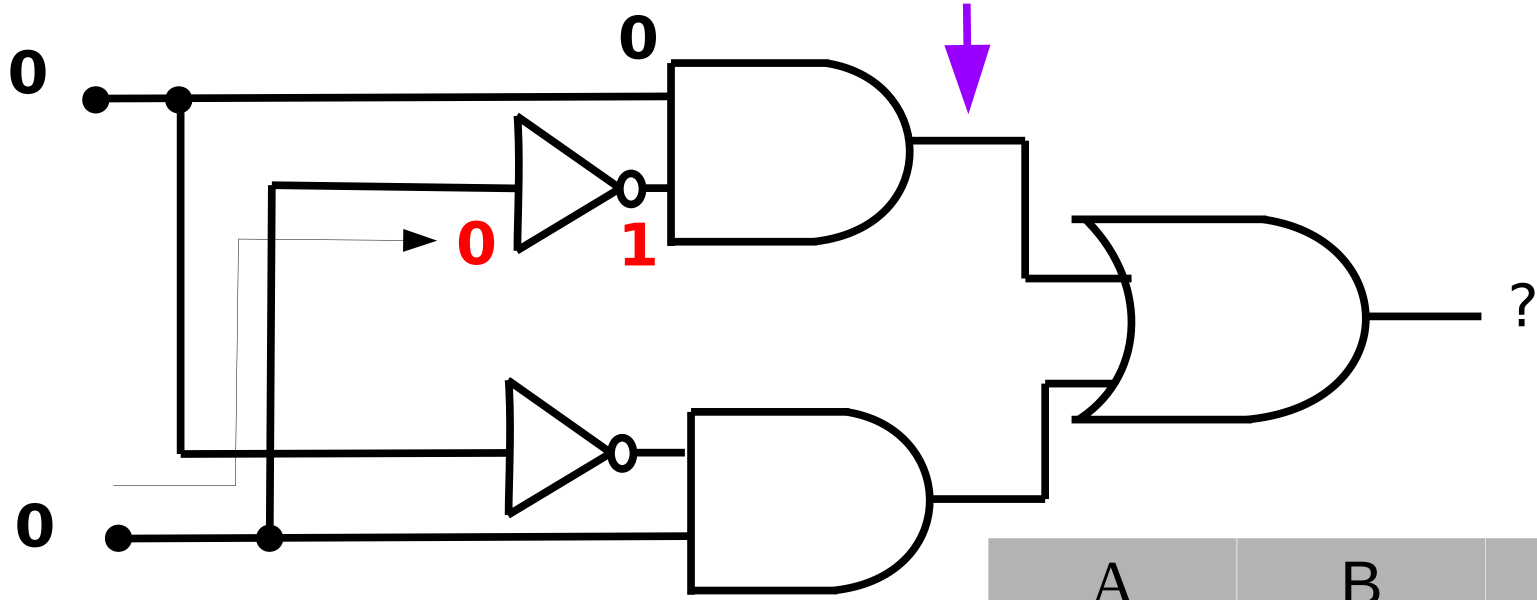
The look at the top NOT gate...



A	B	Q
0	0	?
0	1	?
1	0	?
1	1	?

Combining logic gates a more complex example:

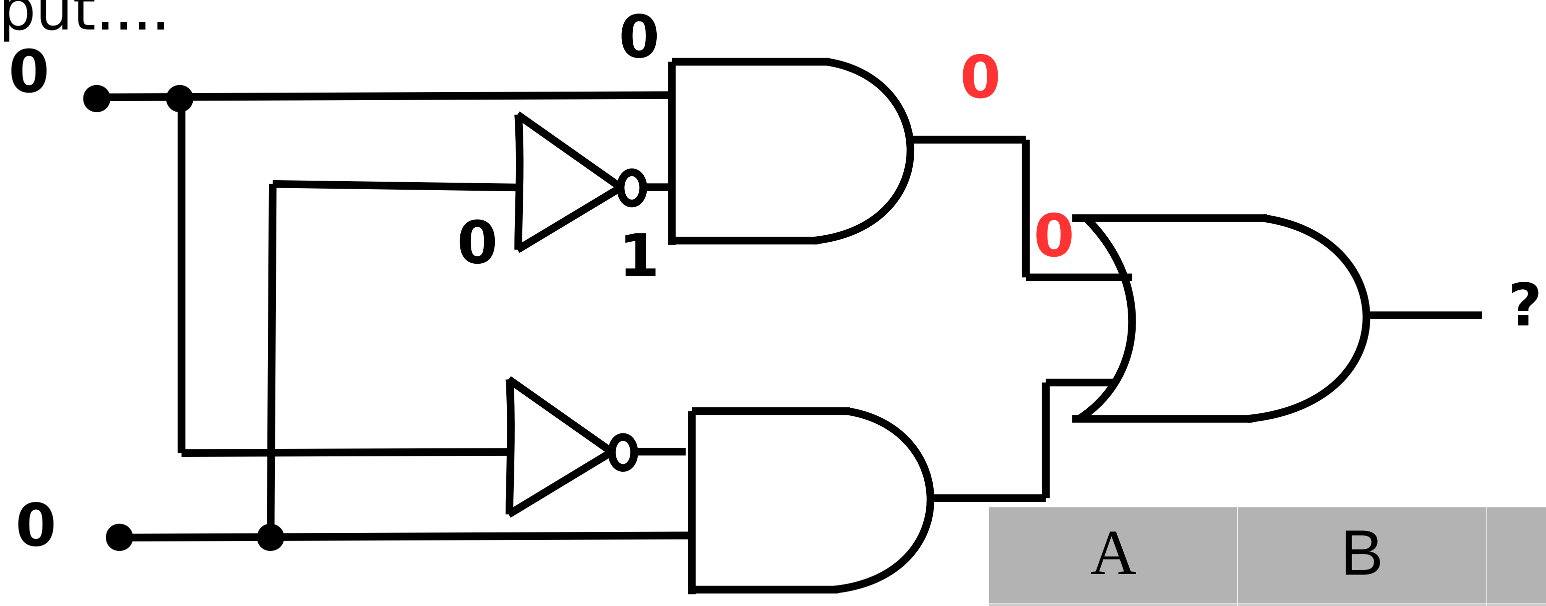
What is the value at the output of the AND gate going to be?



A	B	Q
0	0	?
0	1	?
1	0	?
1	1	?

Combining logic gates a more complex example:

So we know one input to to AND gate, so we know it's output....

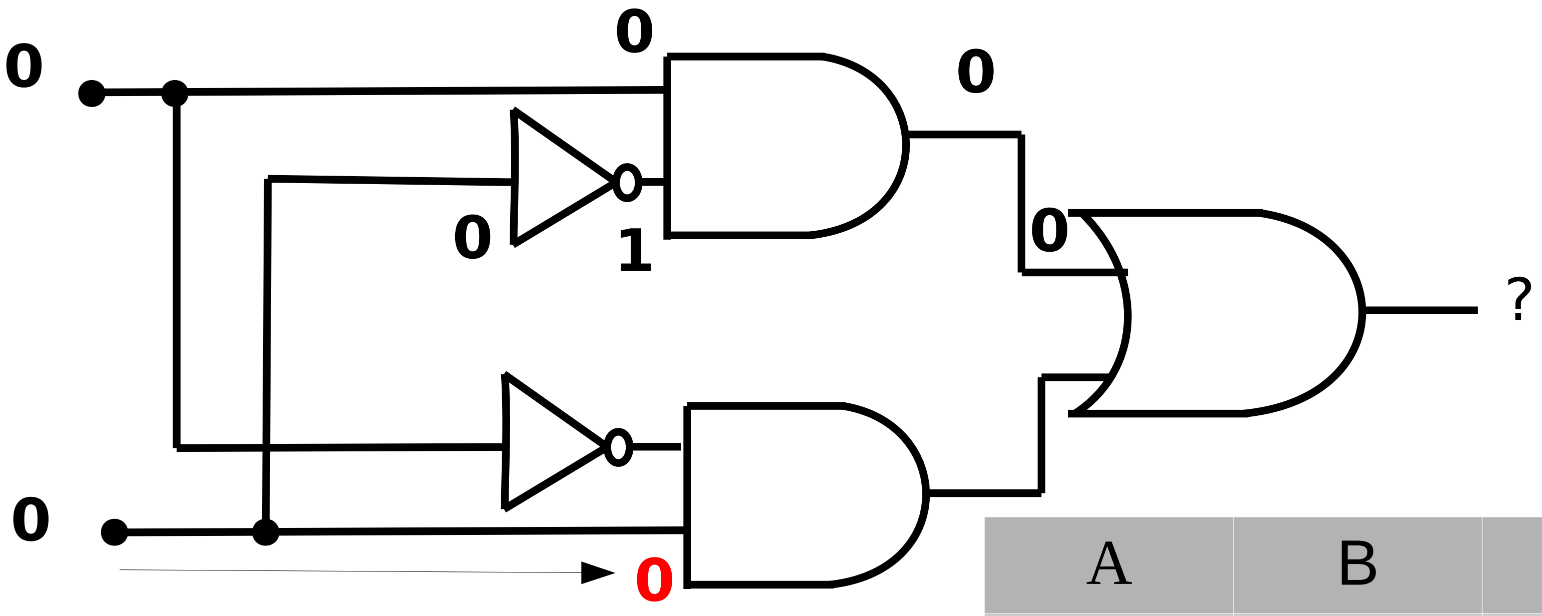


We therefore know one input to the final OR gate.

A	B	Q
0	0	?
0	1	?
1	0	?
1	1	?

Combining logic gates a more complex example:

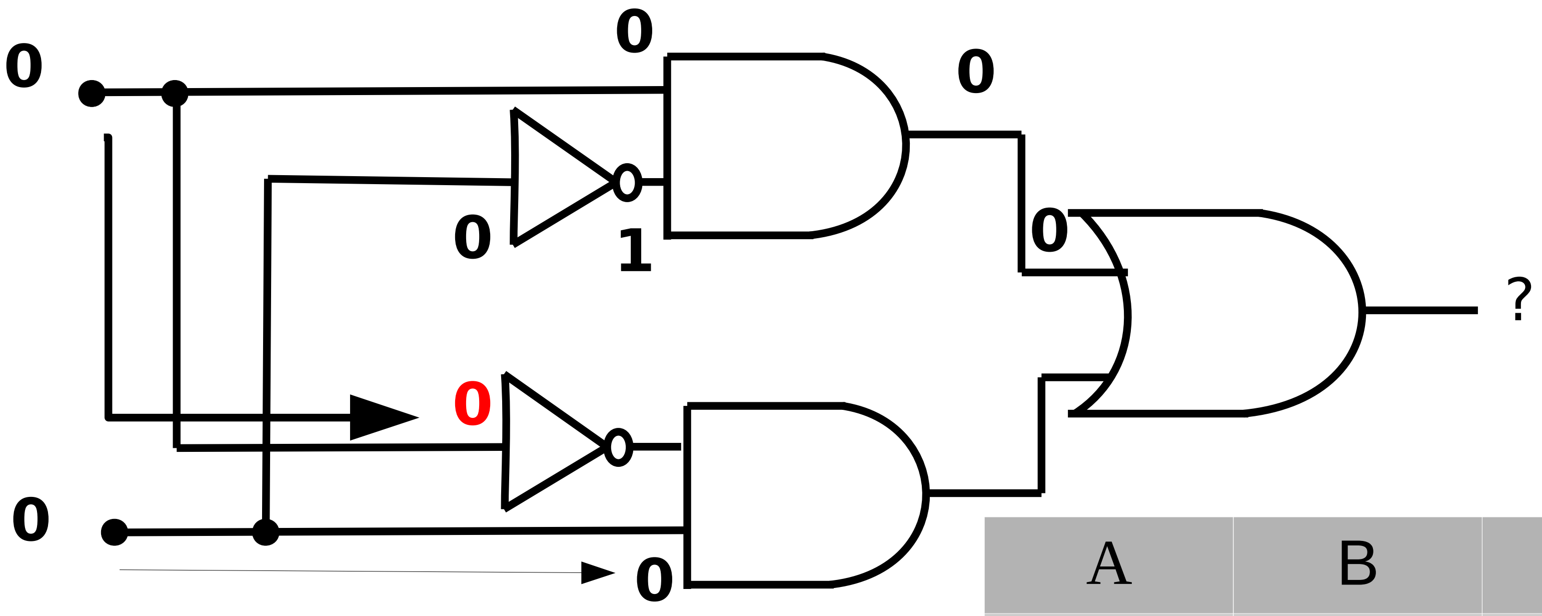
Look at the lower AND gate now....



A	B	Q
0	0	?
0	1	?
1	0	?
1	1	?

Combining logic gates a more complex example:

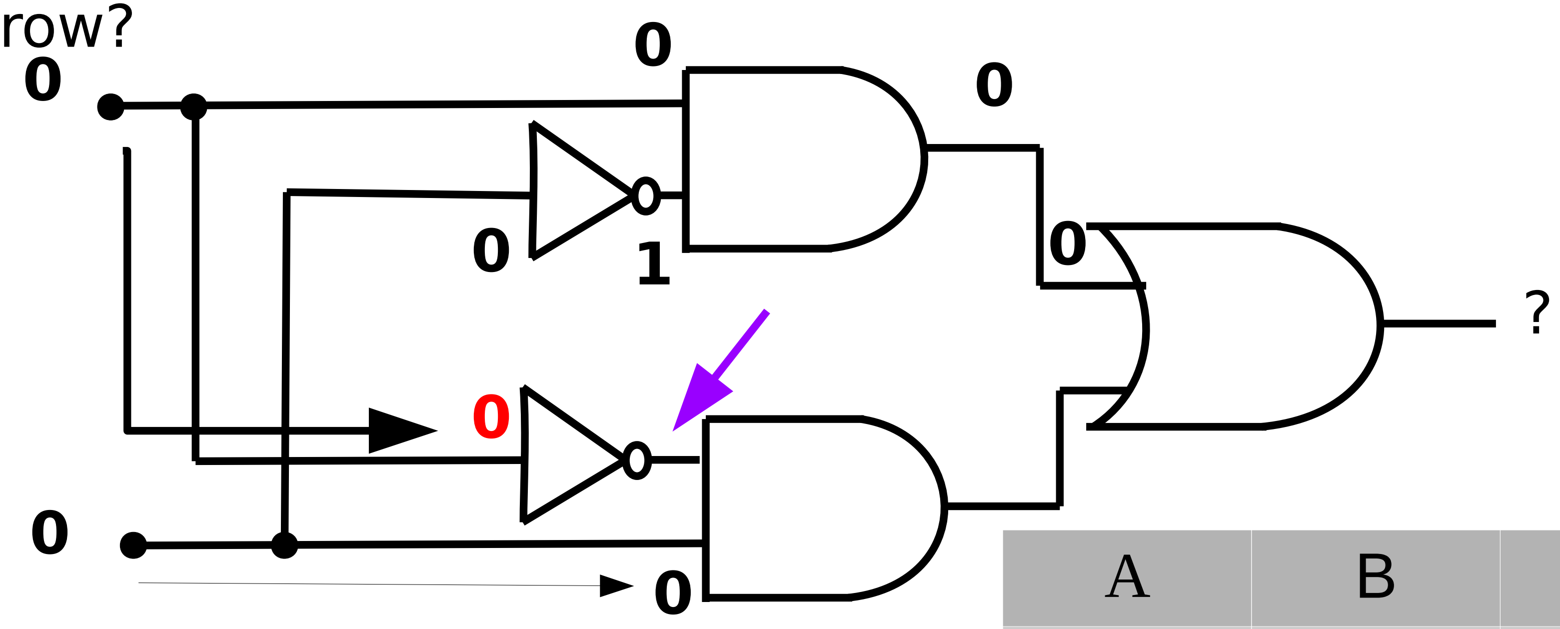
Look at the lower AND gate now....



A	B	Q
0	0	?
0	1	?
1	0	?
1	1	?

Combining logic gates a more complex example:

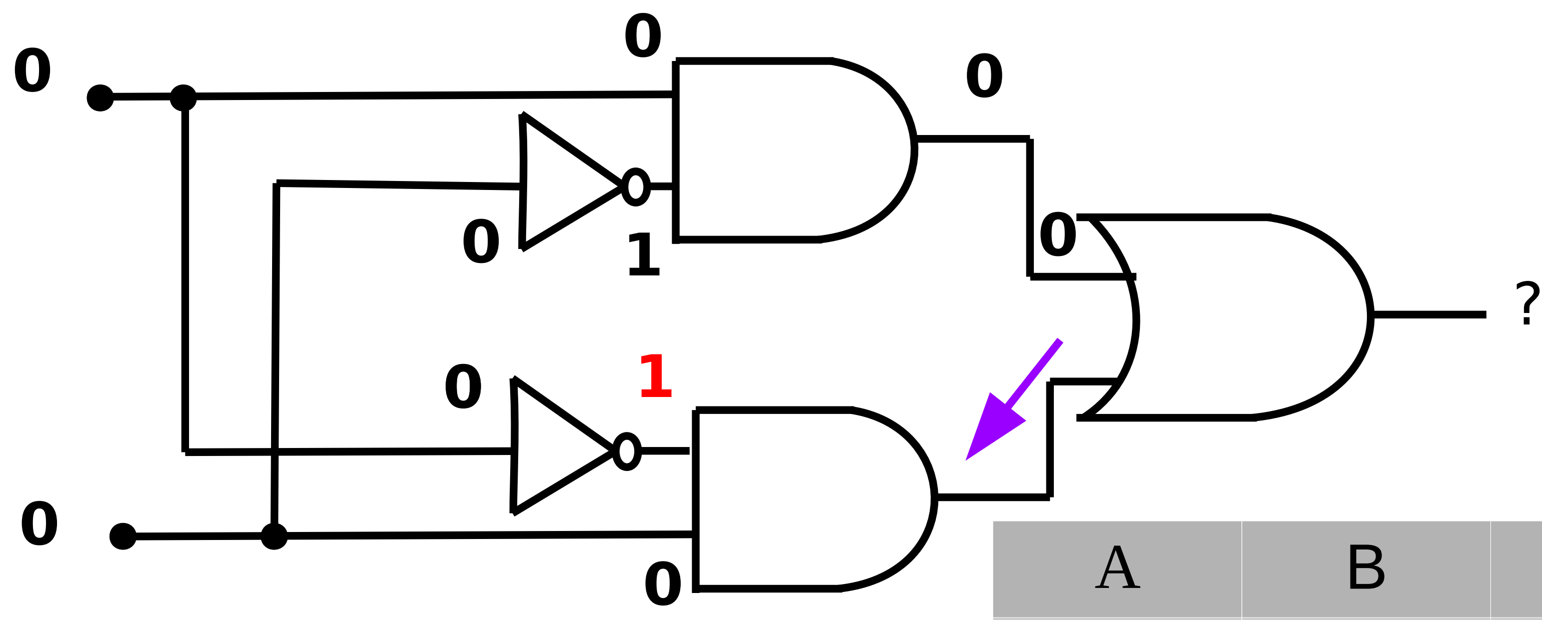
What is the output of the not gate going to be at the purple arrow?



A	B	Q
0	0	?
0	1	?
1	0	?
1	1	?

Combining logic gates a more complex example:

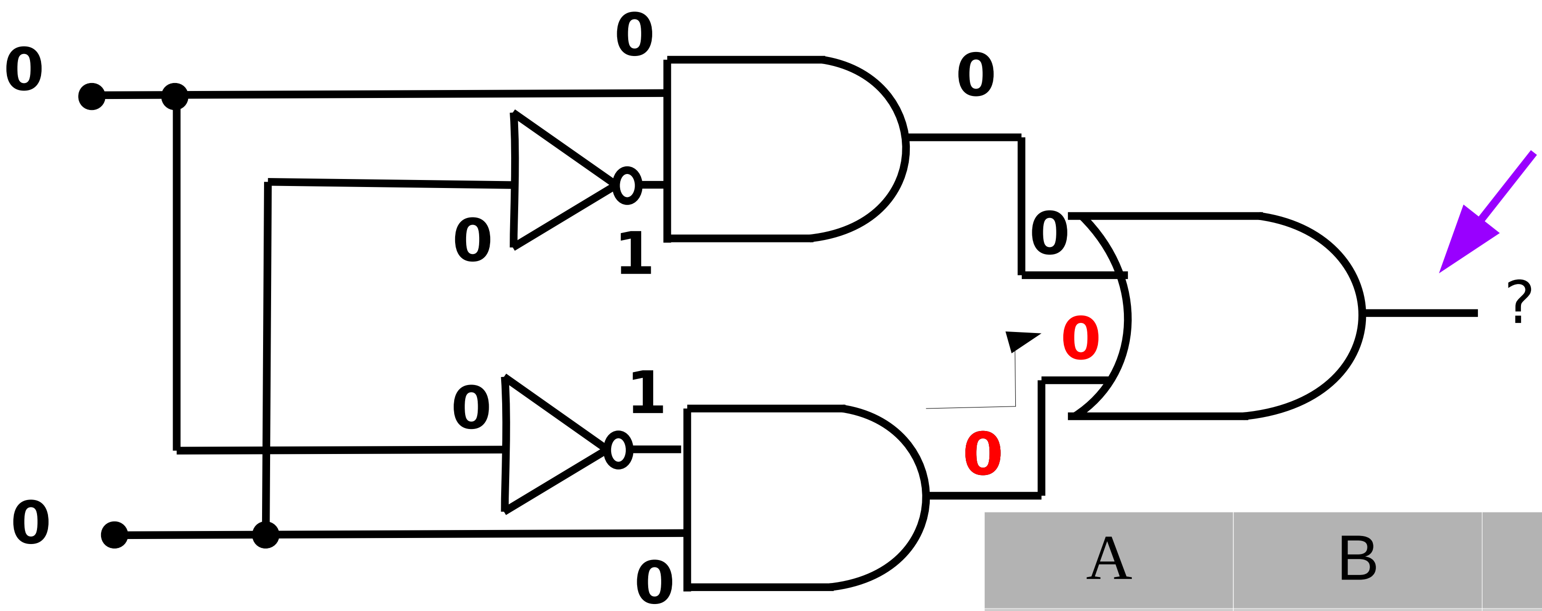
Now what will the output of the second AND gate be?



A	B	Q
0	0	?
0	1	?
1	0	?
1	1	?

Combining logic gates a more complex example:

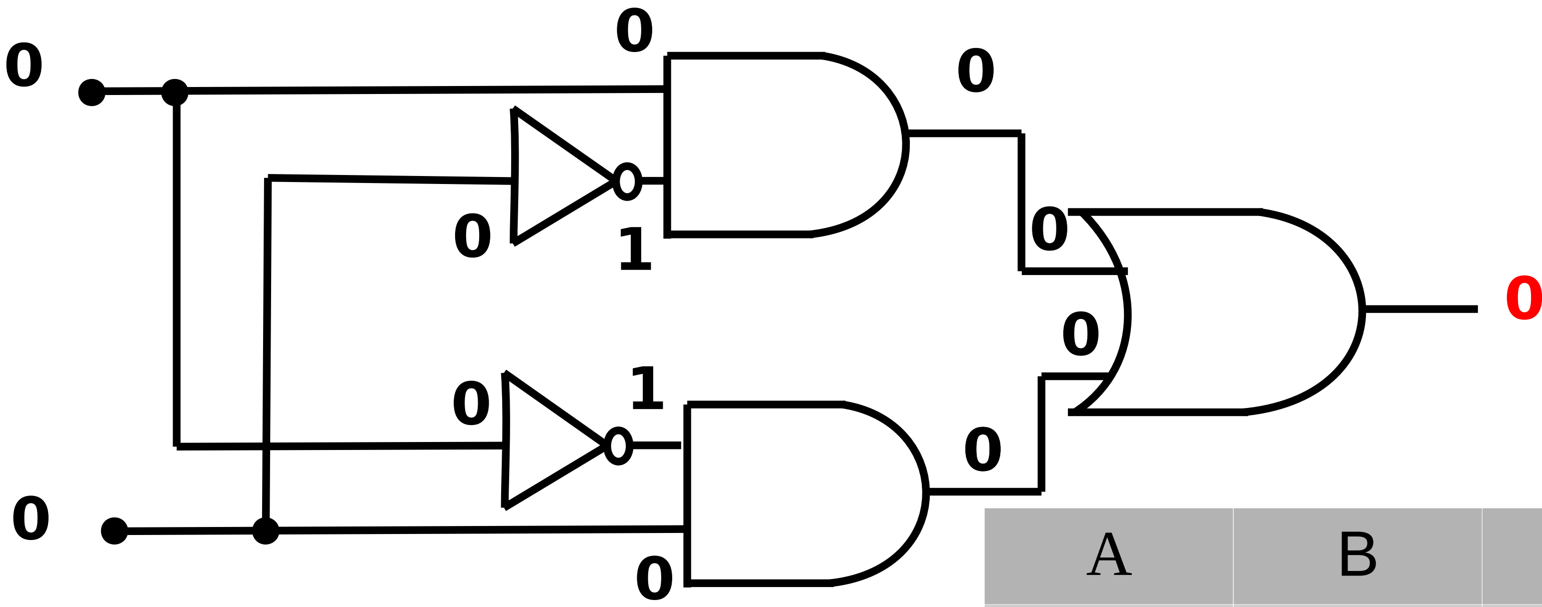
What will the output to the final OR gate be?



A	B	Q
0	0	?
0	1	?
1	0	?
1	1	?

Combining logic gates a more complex example:

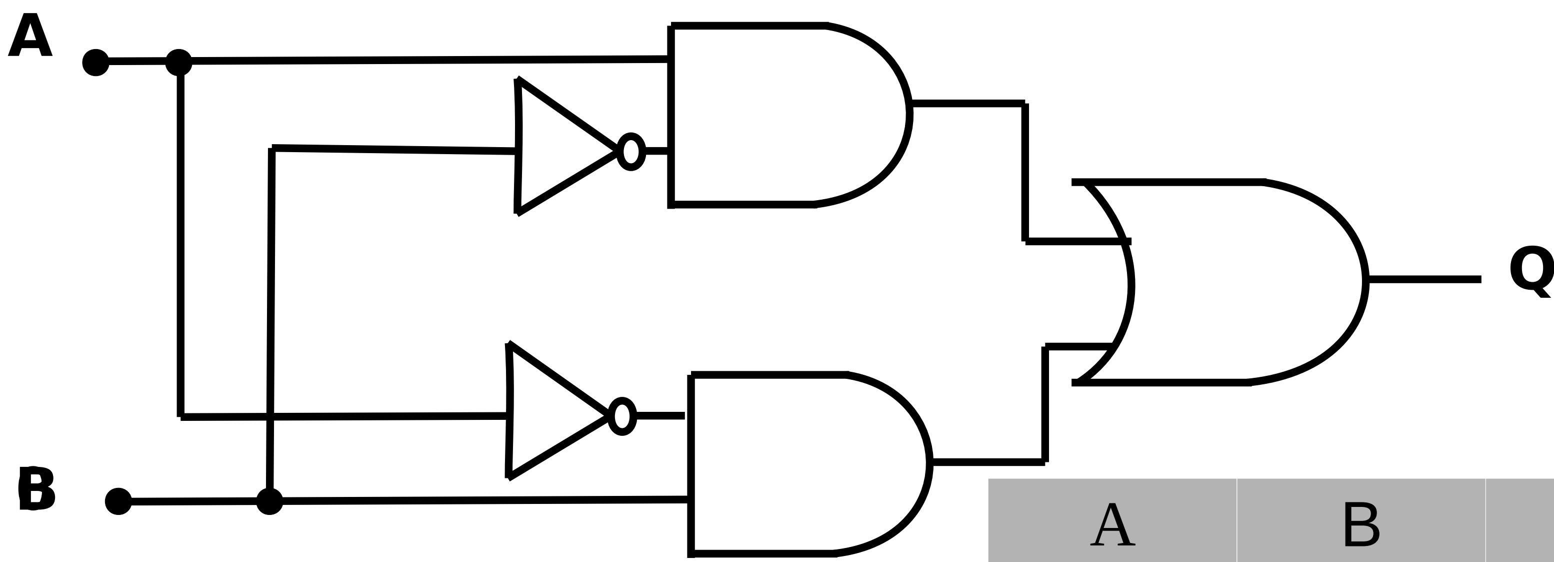
Now write the final value in the truth table...



A	B	Q
0	0	0
0	1	?
1	0	?
1	1	?

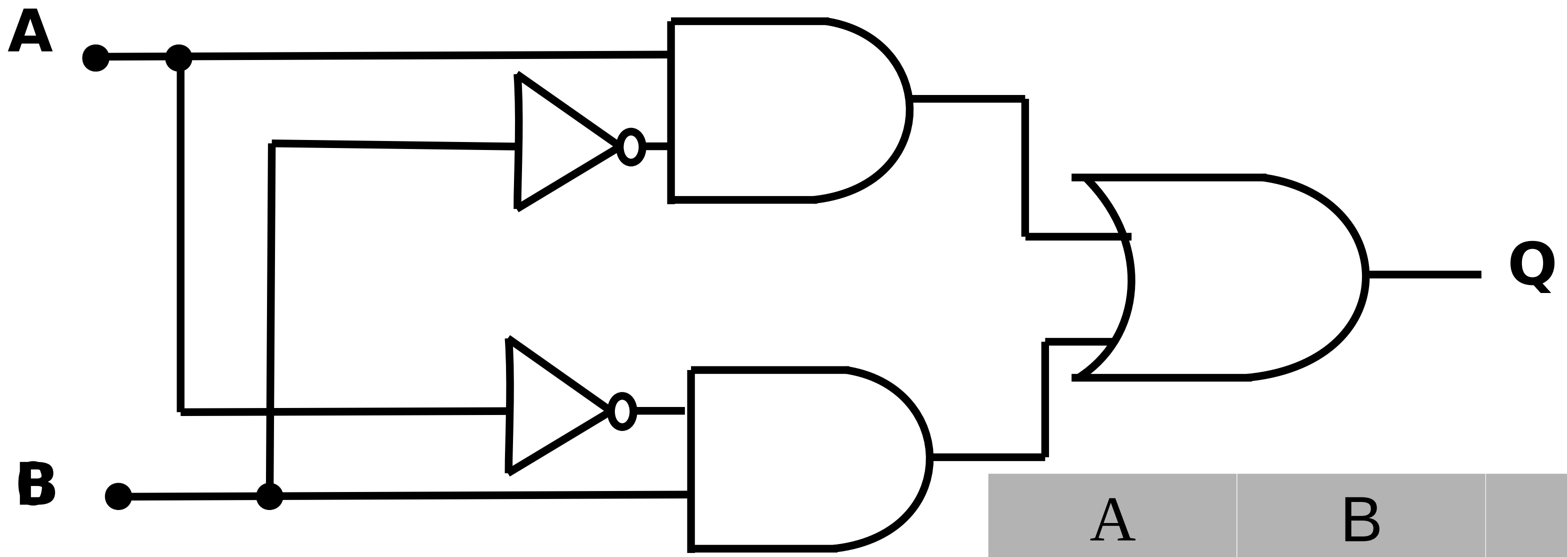
Your go!

Fill in the net three rows of the logic table:



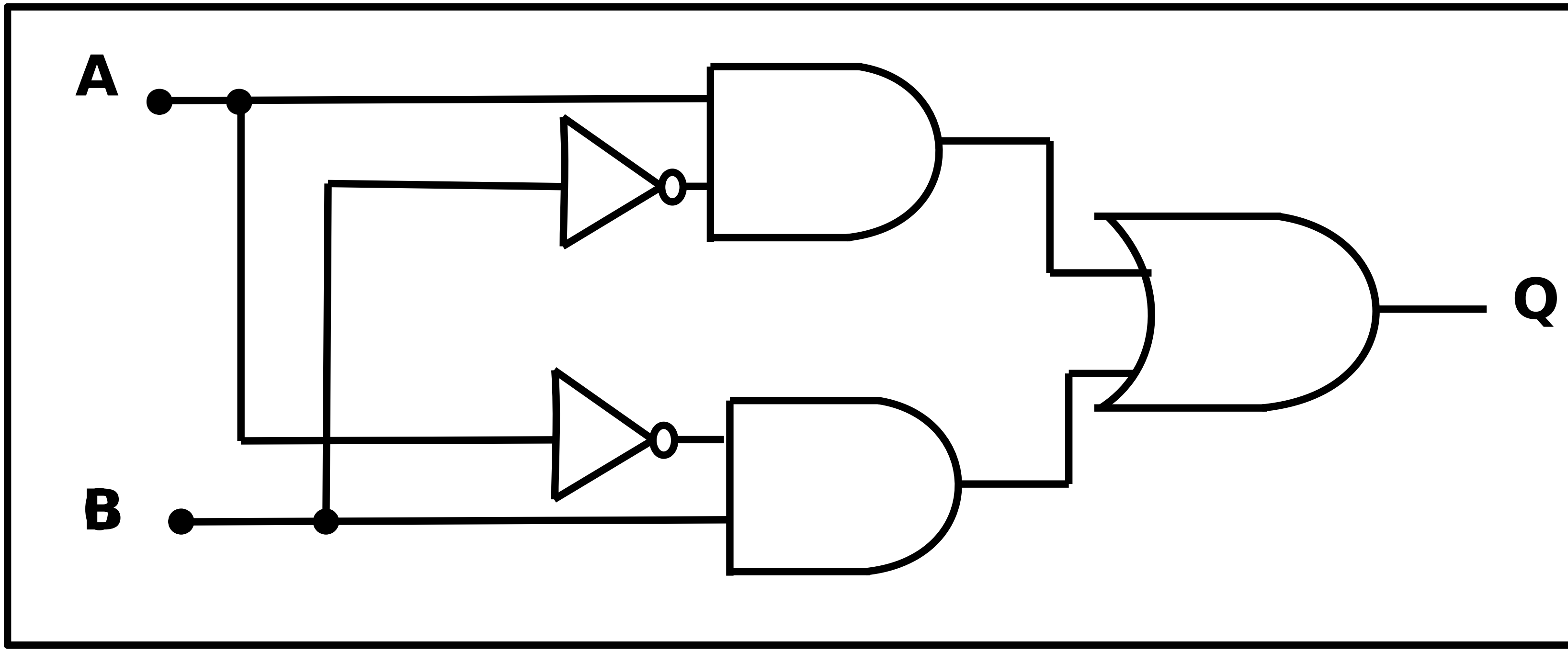
A	B	Q
0	0	0
0	1	?
1	0	?
1	1	?

The answer!

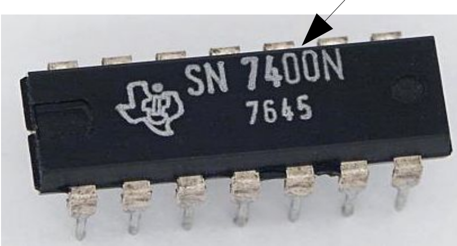


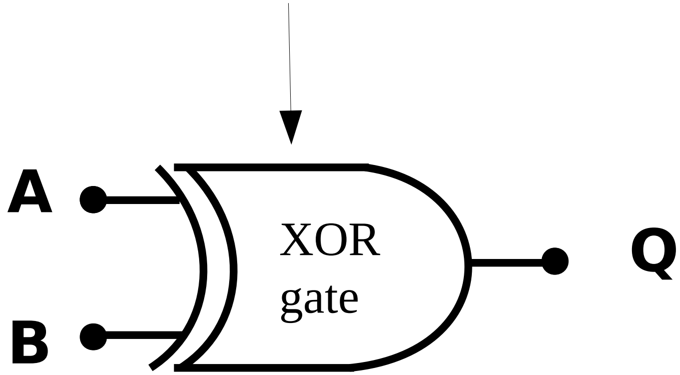
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

This is actually another type of special gate called an XOR gate



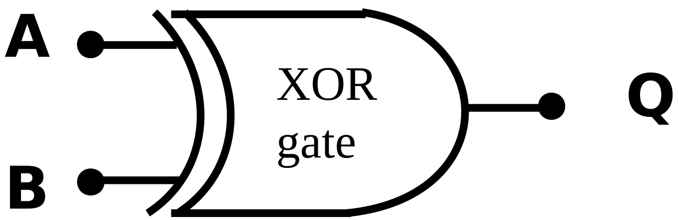
74266





XOR gates

I'm not going to teach you more about XOR gates, but you will come across them if you:



- Start doing mathematical operations in electronics:
  - Adding numbers
  - Subtracting numbers
  - Multiplying numbers

Outline of the lecture

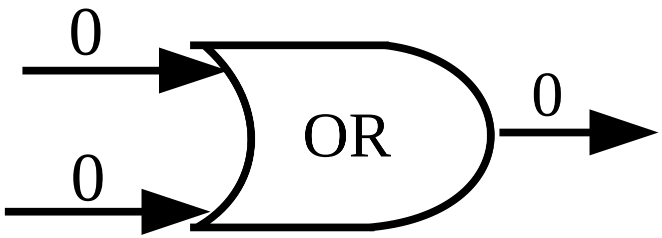
- Recap of last lecture
- Recap of gates - Mini quiz
- Figuring out what electronic circuits do
- **Making electronics remember things**
  - **Flip flops**
  - Serial to parallel converters

Outline of the lecture

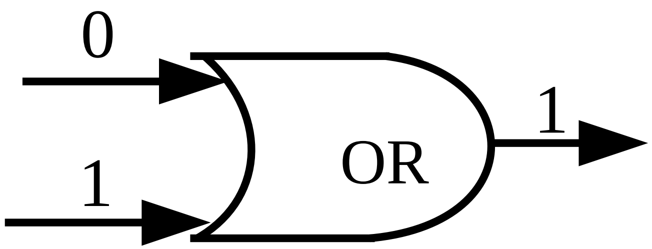
- So far we have learnt how electronics can be used to:
    - Count – binary
    - Make simple decisions – logic gates
    - Detect position – shaft encoders
- But to make our circuits really smart they need to be able to **remember** things.
- This is the next step...

Patrick J. Lynch,

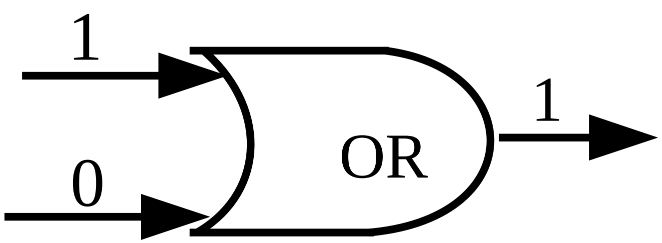
Making electronics remember things: First recap the OR gate

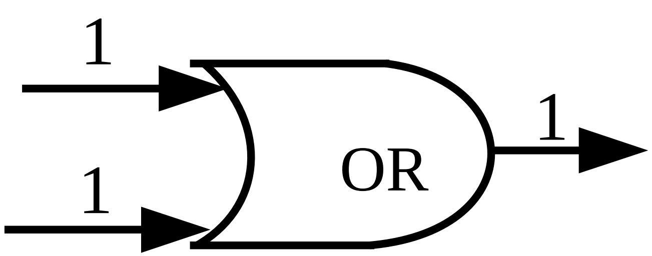


•If **any** of the **inputs** are **1** then the output will be **1**.



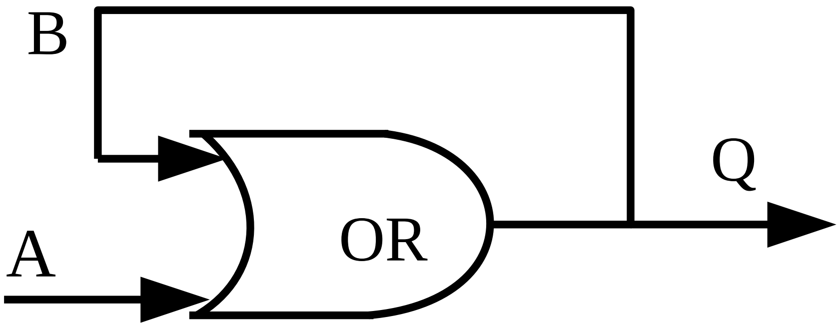
•You only get **zero** when **both** inputs are **zero**.





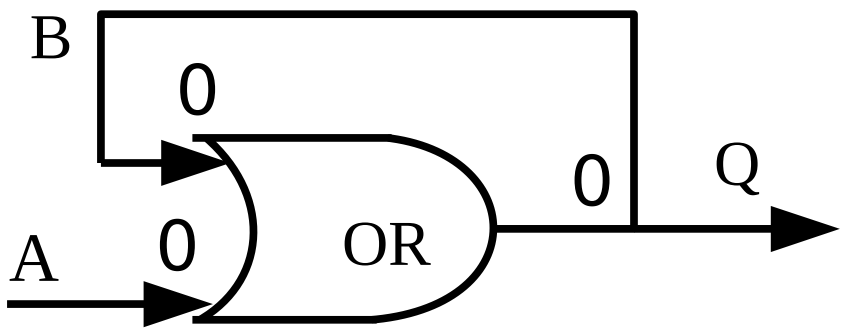
•Now have a think about this circuit...

Let's do a thought experiment, think about this circuit.



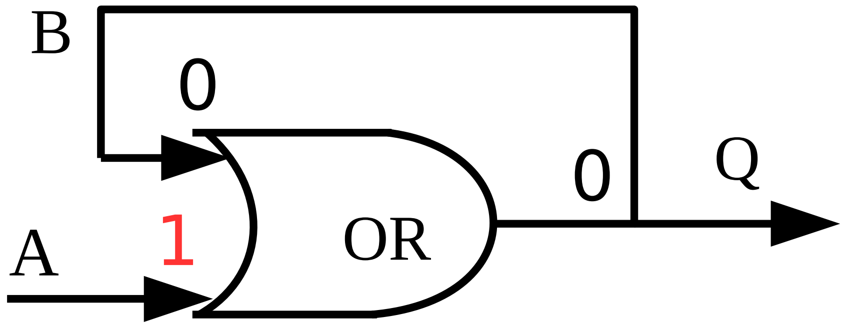
- It's got a feedback loop

Let's do a thought experiment, think about this circuit.



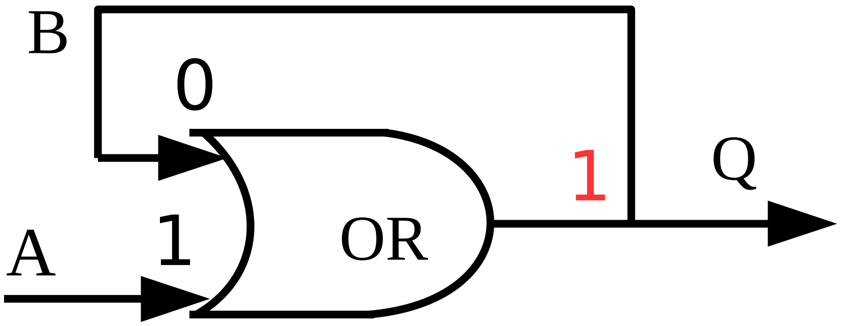
- If initially A, B and Q are zero

Let's do a thought experiment, think about this circuit.



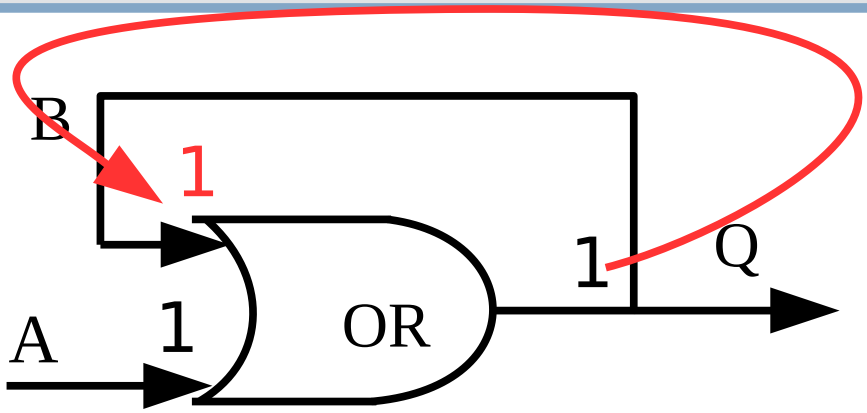
- If initially A, B and Q are zero
- Then you make input A=1

Let's do a thought experiment, think about this circuit.



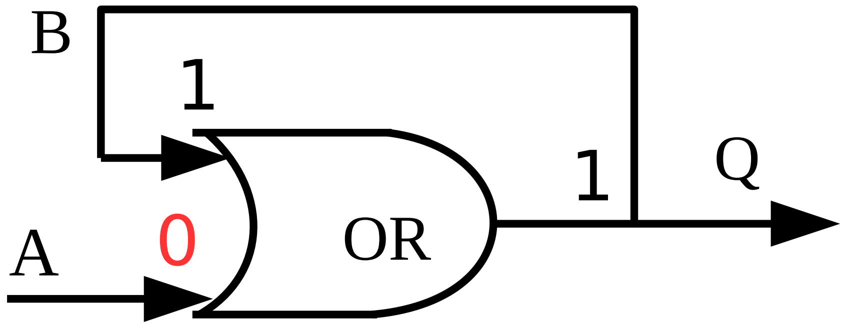
- This will make the output=1

Let's do a thought experiment, think about this circuit.



- It will also make the input B, 1

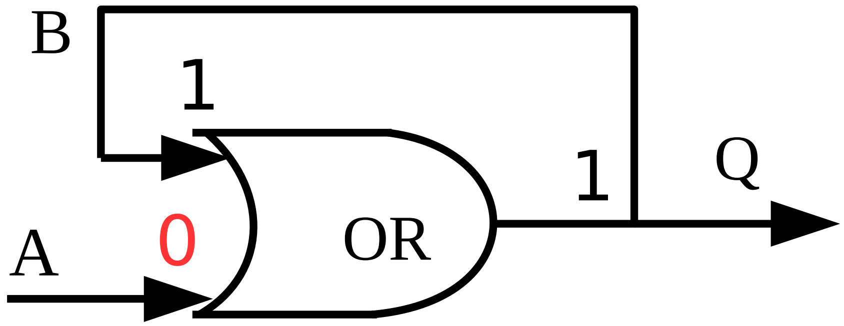
Let's do a thought experiment, think about this circuit.



•Now if we again make input A 0

•What happen to the output?

Let's do a thought experiment, think about this circuit.



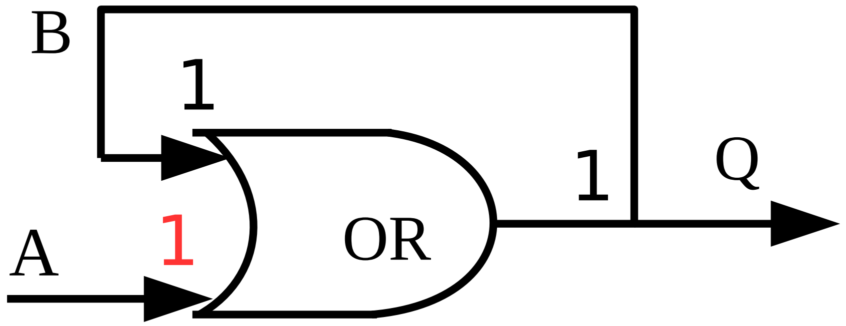
- The output will now stay 1 forever!
    - No matter what we do.

- We have made our or gate remember that it was turned on!

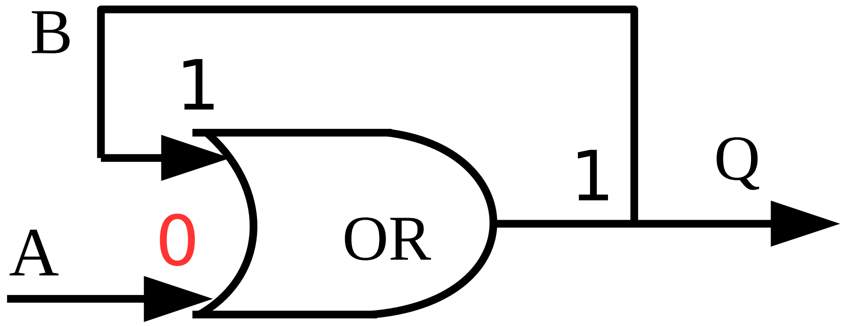
- We have computer memory!

Patrick J. Lynch,

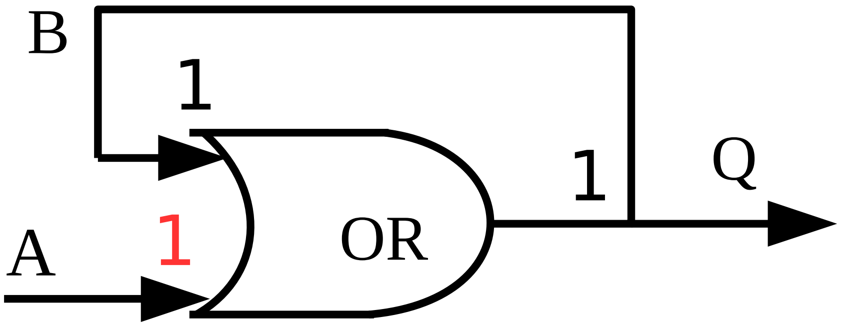
Let's do a thought experiment, think about this circuit.



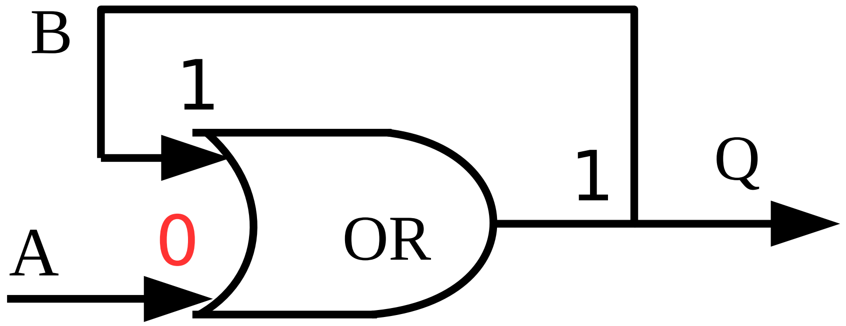
Let's do a thought experiment, think about this circuit.



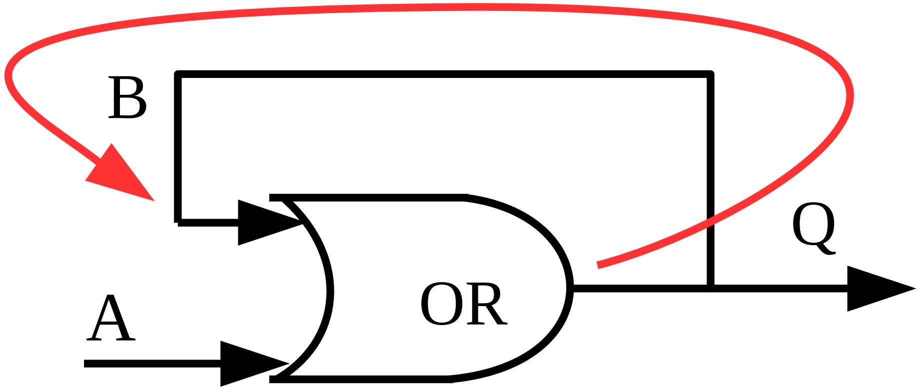
Let's do a thought experiment, think about this circuit.



Let's do a thought experiment, think about this circuit.

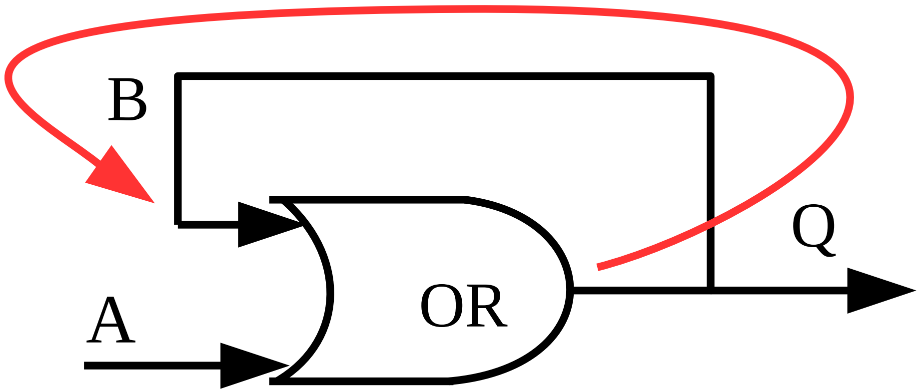


This is called a latch circuit

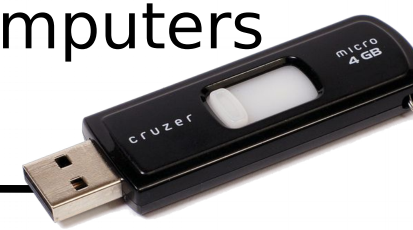


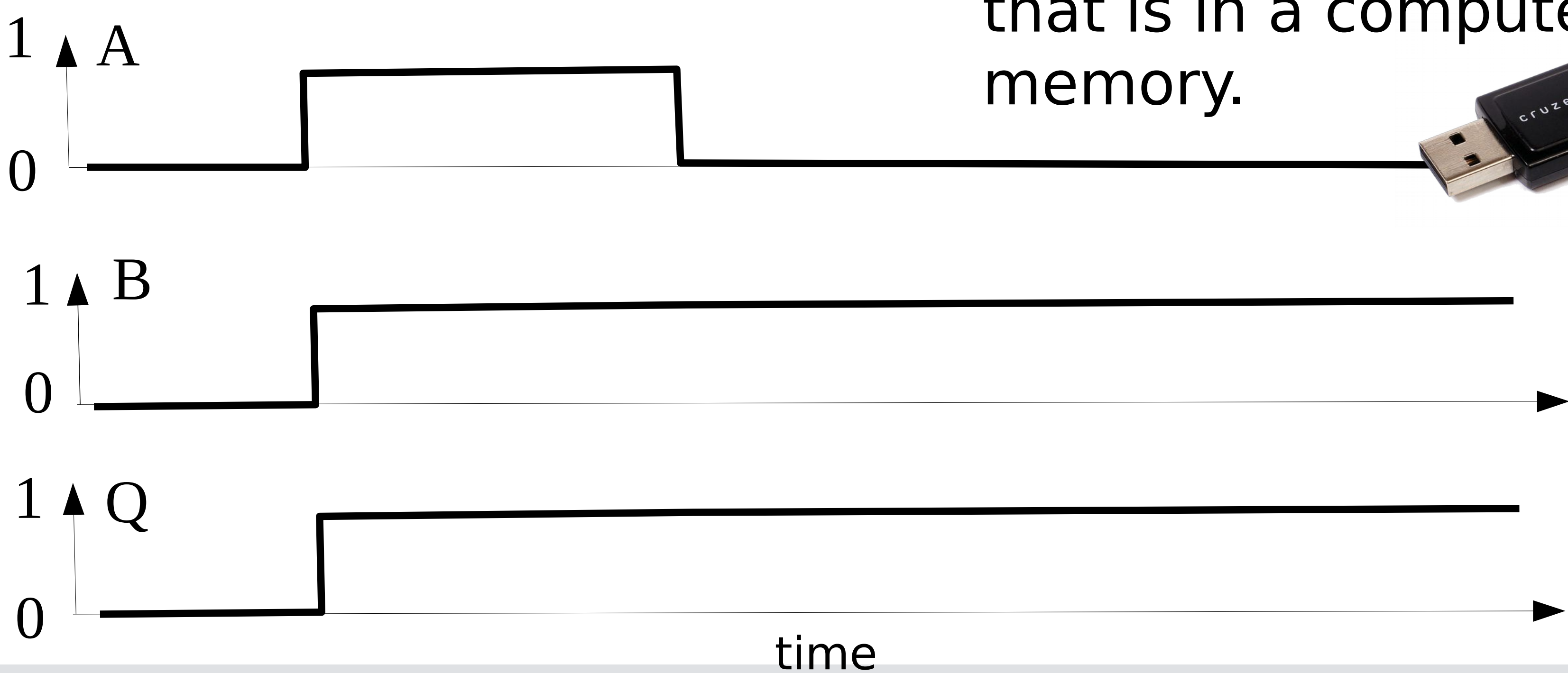
- Because it latches on, just like a door latch.

A timing diagram



- Let's look at the levels of each gate as a function of time.
- This is the type of circuit that is in a computers memory.

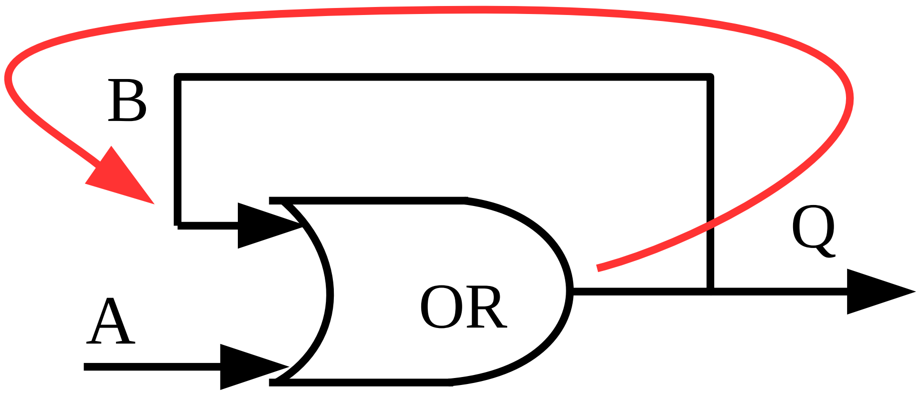




There is a problem....

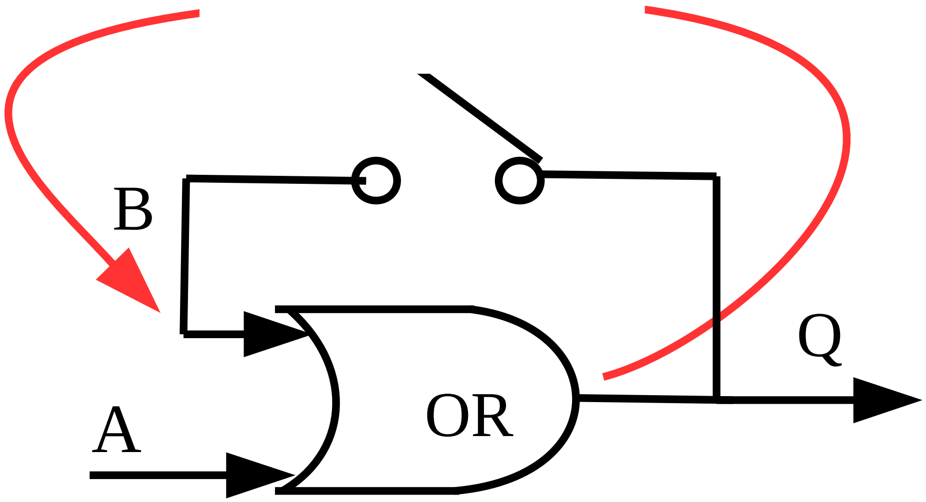
•Once we turn our memory element **on** we can never turn it **off**.

•Meaning once we have stored a **1** we can **never store a 0.**



Let's look at the in and outputs

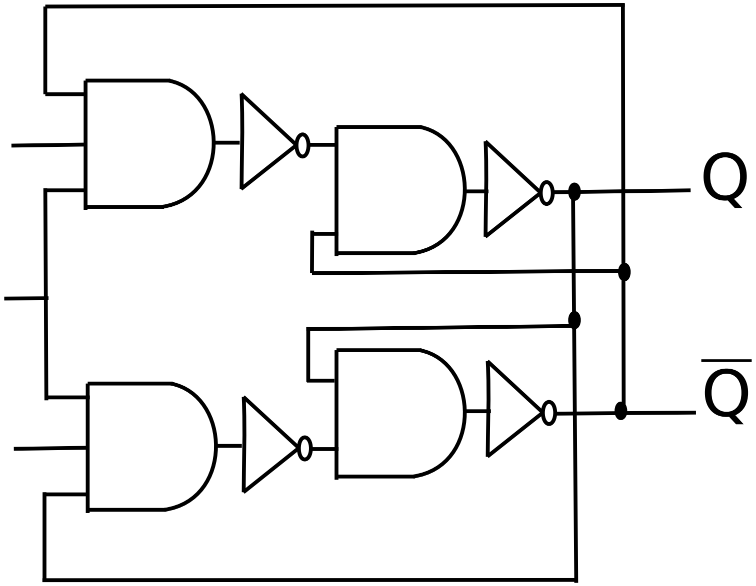
•We could get around this by putting some type of **switch** into the loop to stop the feedback.



•What we actually do is a little more complex.....

We use a circuit called a JK flip-flop – it looks like this:

•This is a real memory circuit that you would find in a computer. It can store a **1** or a **0**.

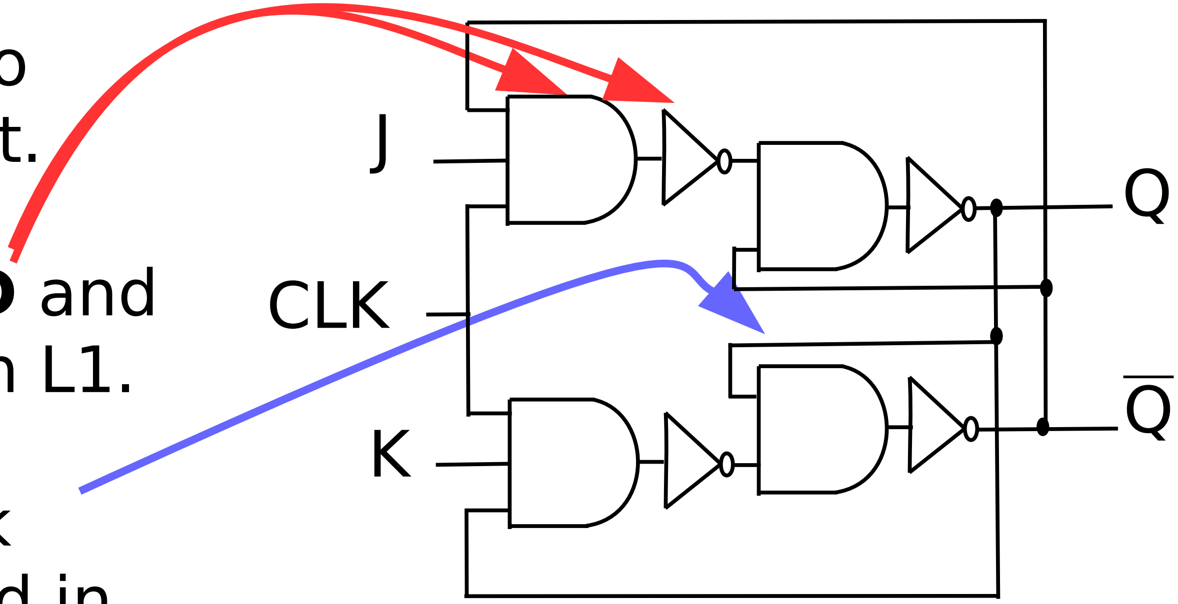



Dejdżer / Digga - Flickr

•It looks a bit complicated.

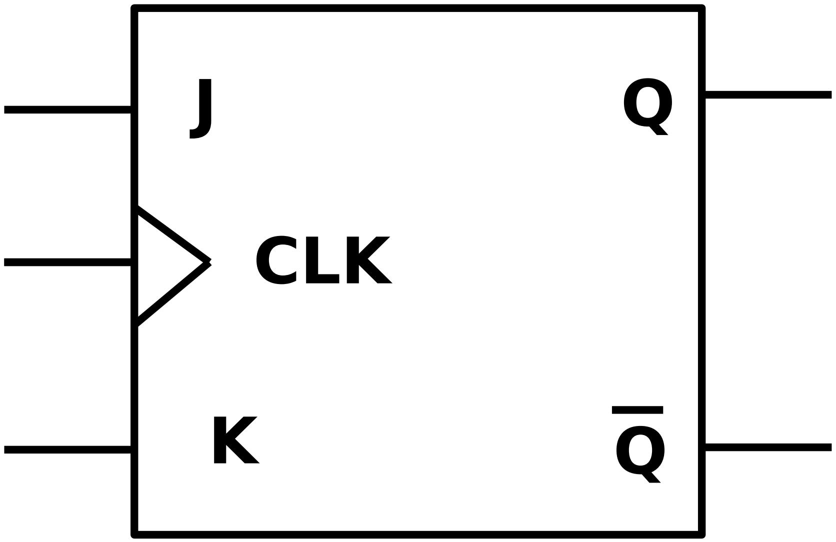
We use a circuit that looks like this:

- I don't expect you to remember the circuit.
- But notice, the **AND** and NOT gates we met in L1.
- Notice the feedback wires just like we had in our latching OR gate.



- From now on I am going to hide all the complicated detail by putting them in a box.

The JK flip flop

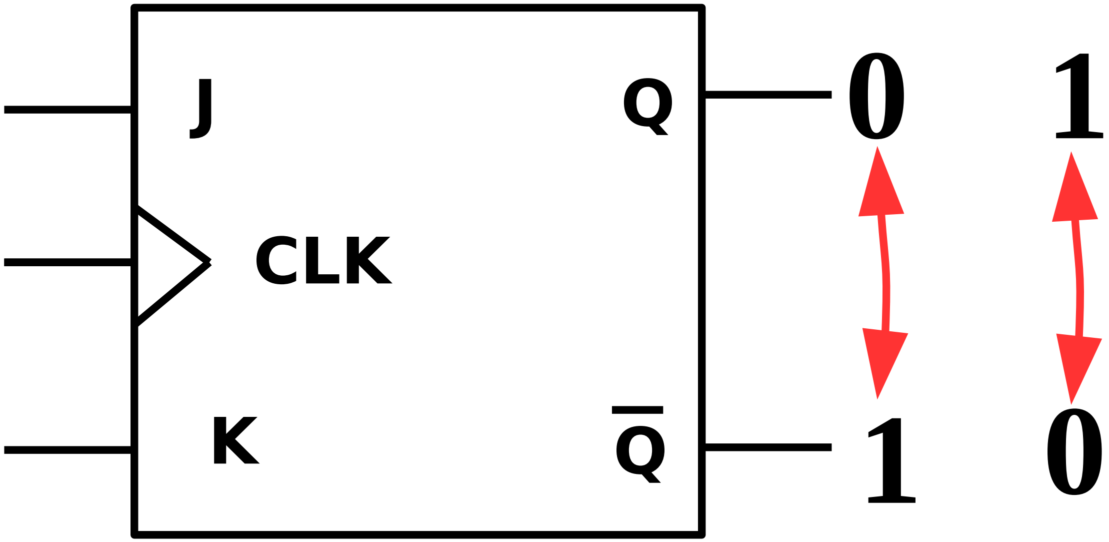


•This is the most basic electronic memory element used to store 1's or 0's.

•It is therefore important to remember what it does and how you would use it.

•You will meet this all the time as soon as you start working with electronic circuits. Used in **memory**, **counters**, **processors** etc...

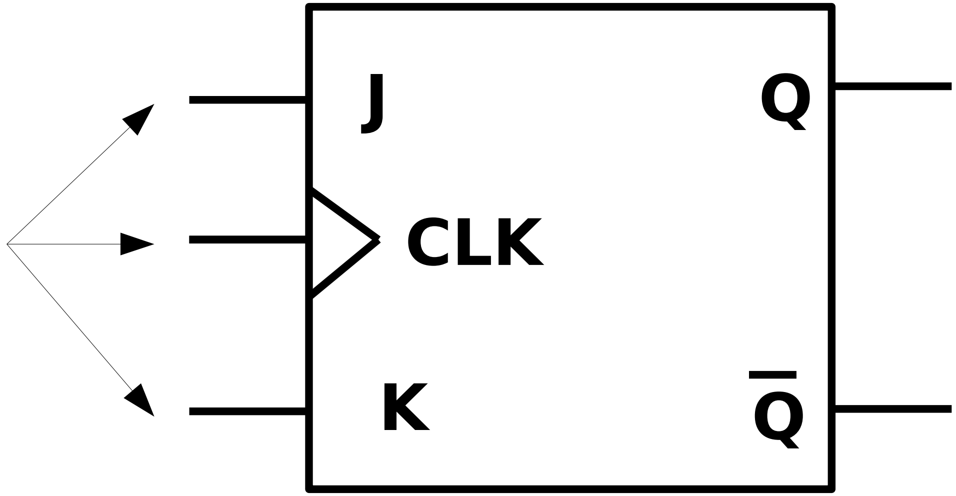
•It's called a flip flop because the outputs can flip and flop between two states 1 and 0.



•And this is how it works....

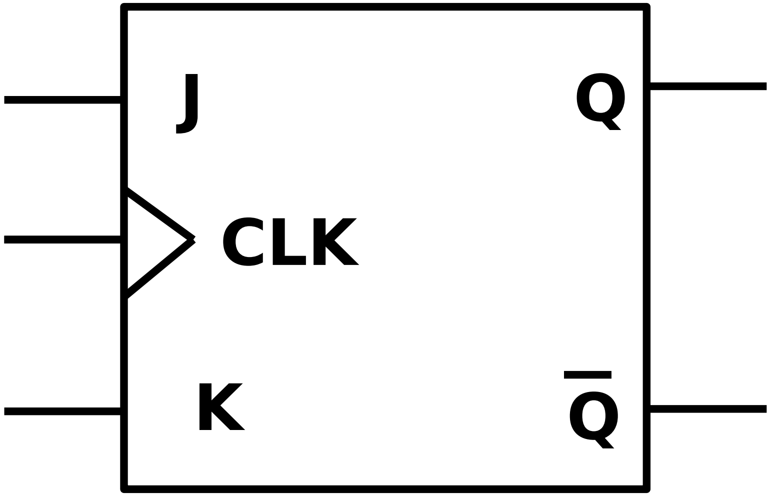
How a JK flip flop works

J and **K** are the inputs



I will explain what the difference between the pins in a moment.

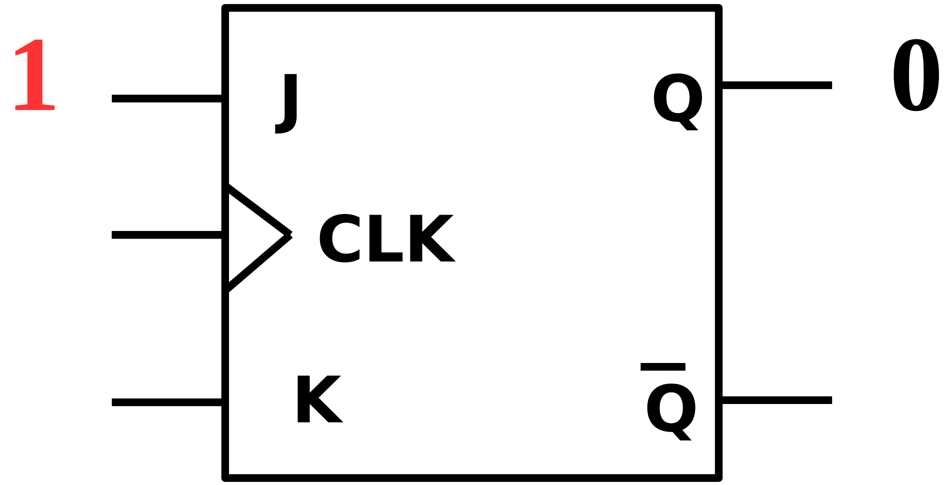
How a JK flip flop works



Q and $\overline{\mathbf{Q}}$ are the outputs

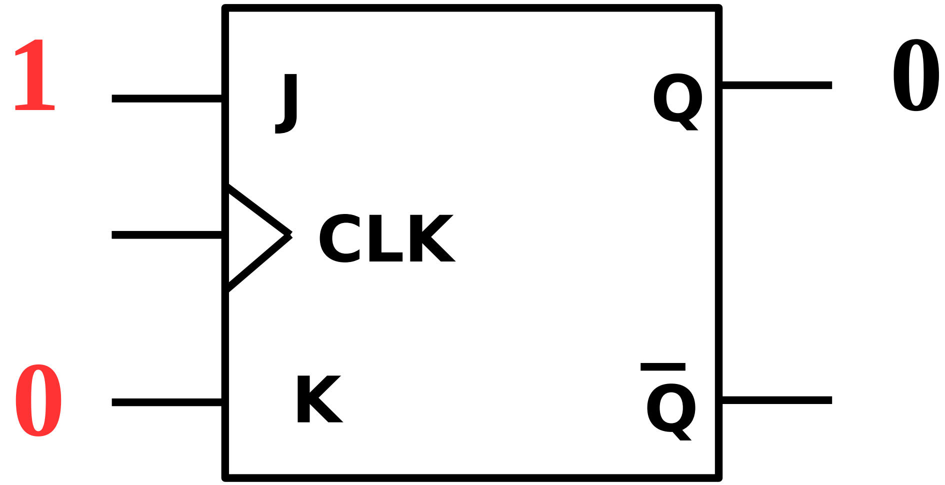
Storing a 1

- The basic idea is that you put what you want to remember on the J input – in this case a 1.



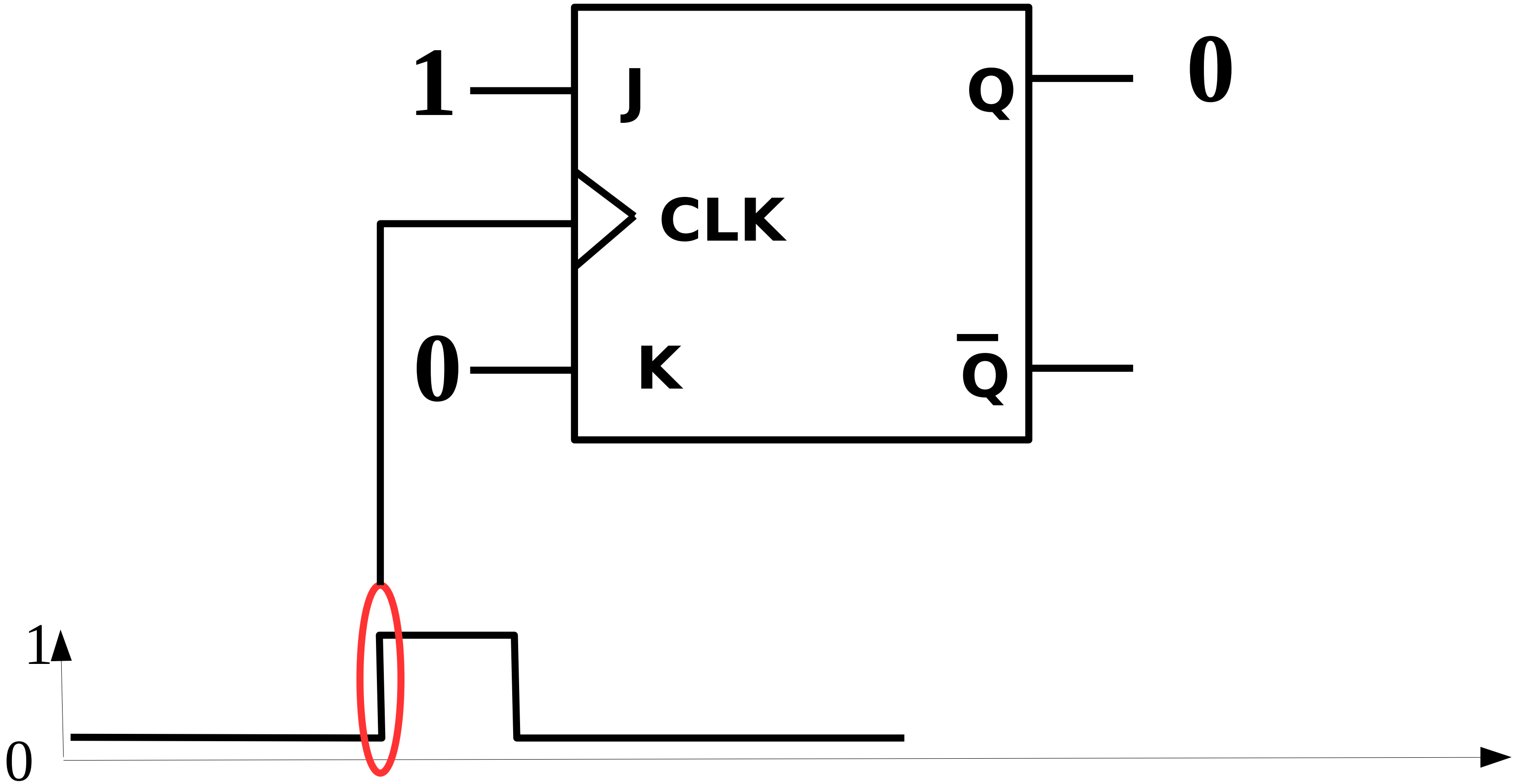
Storing a 1

- At the same time you put the inverse of what you want to store on the K input:

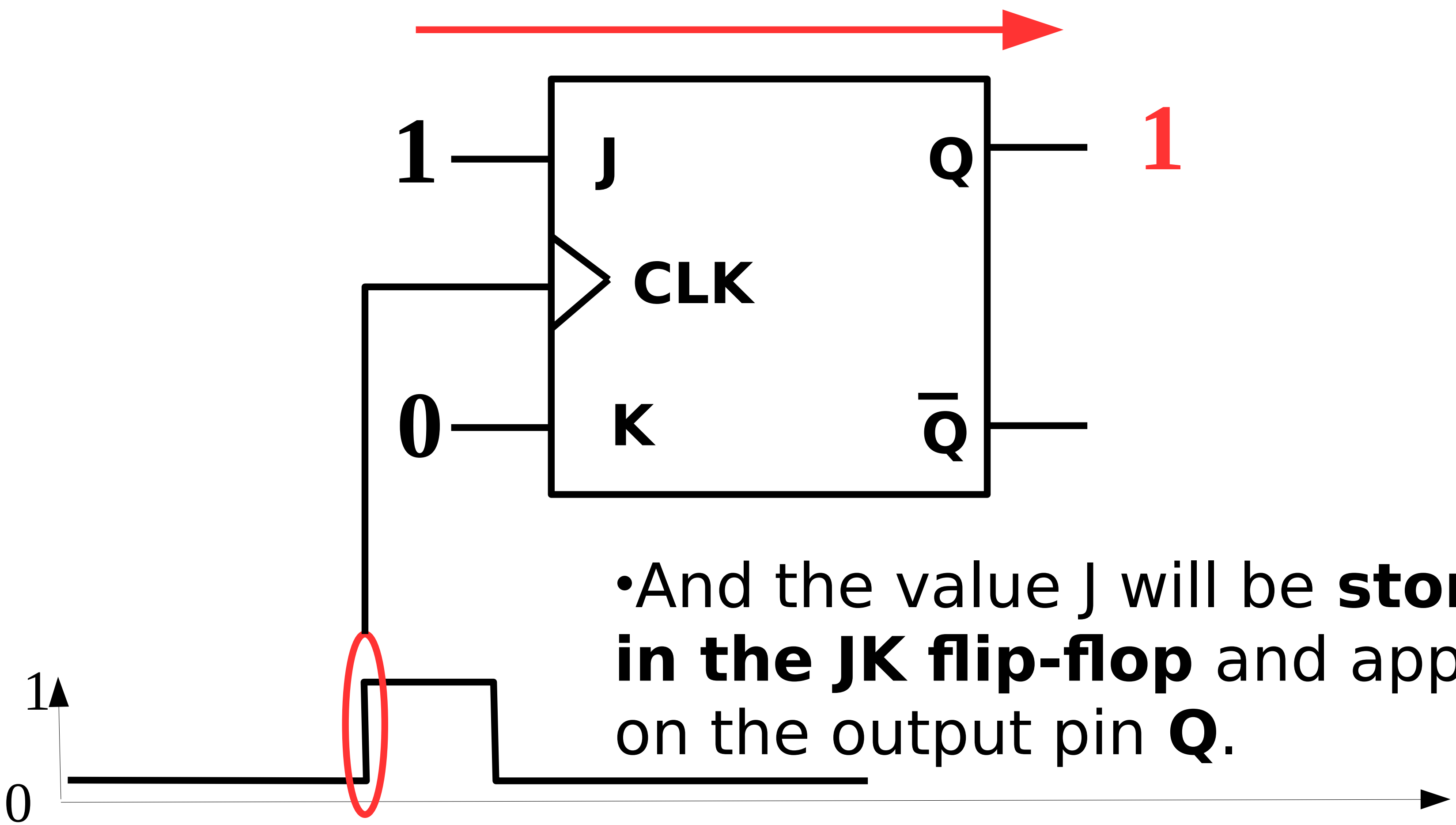


Storing a 1

- You then change the CLK (clock) pin from a 0 to a 1.



Storing a 1



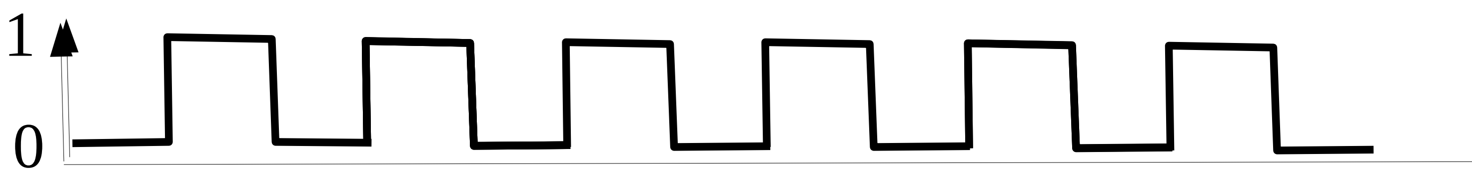
•And the value J will be **stored in the JK flip-flop** and appear on the output pin **Q**.

Clocks in digital electronics

•A clock (CLK) in digital electronics is just a series of on off pulses which are used to synchronize a circuit.

JuergenG,

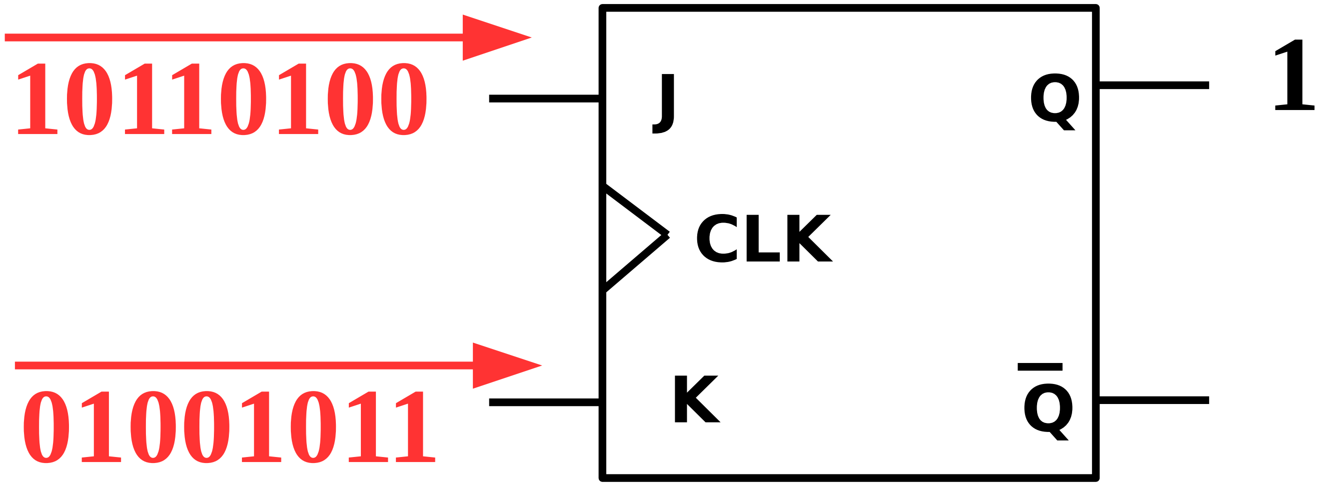
•The faster the clock pulse the faster the circuit will change it's states.



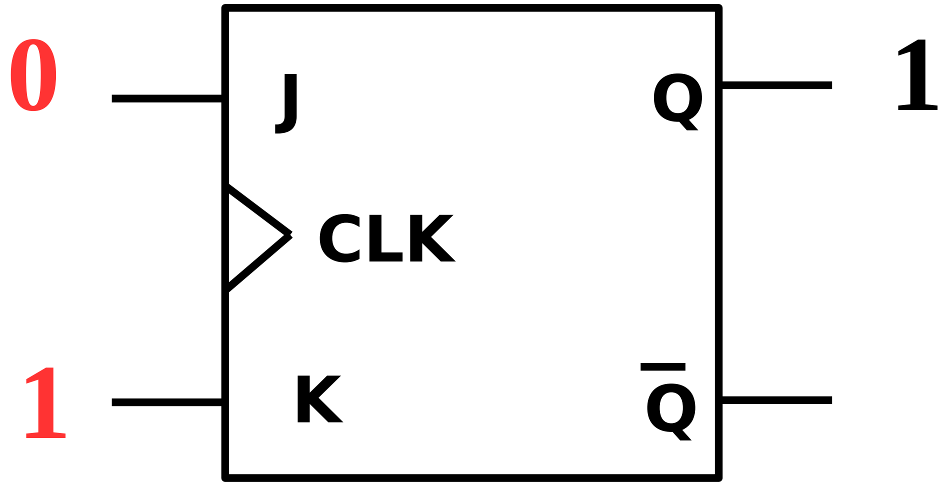
•This is why a faster clock speed will give you a faster computer.

We need a clock pulse to remember things.

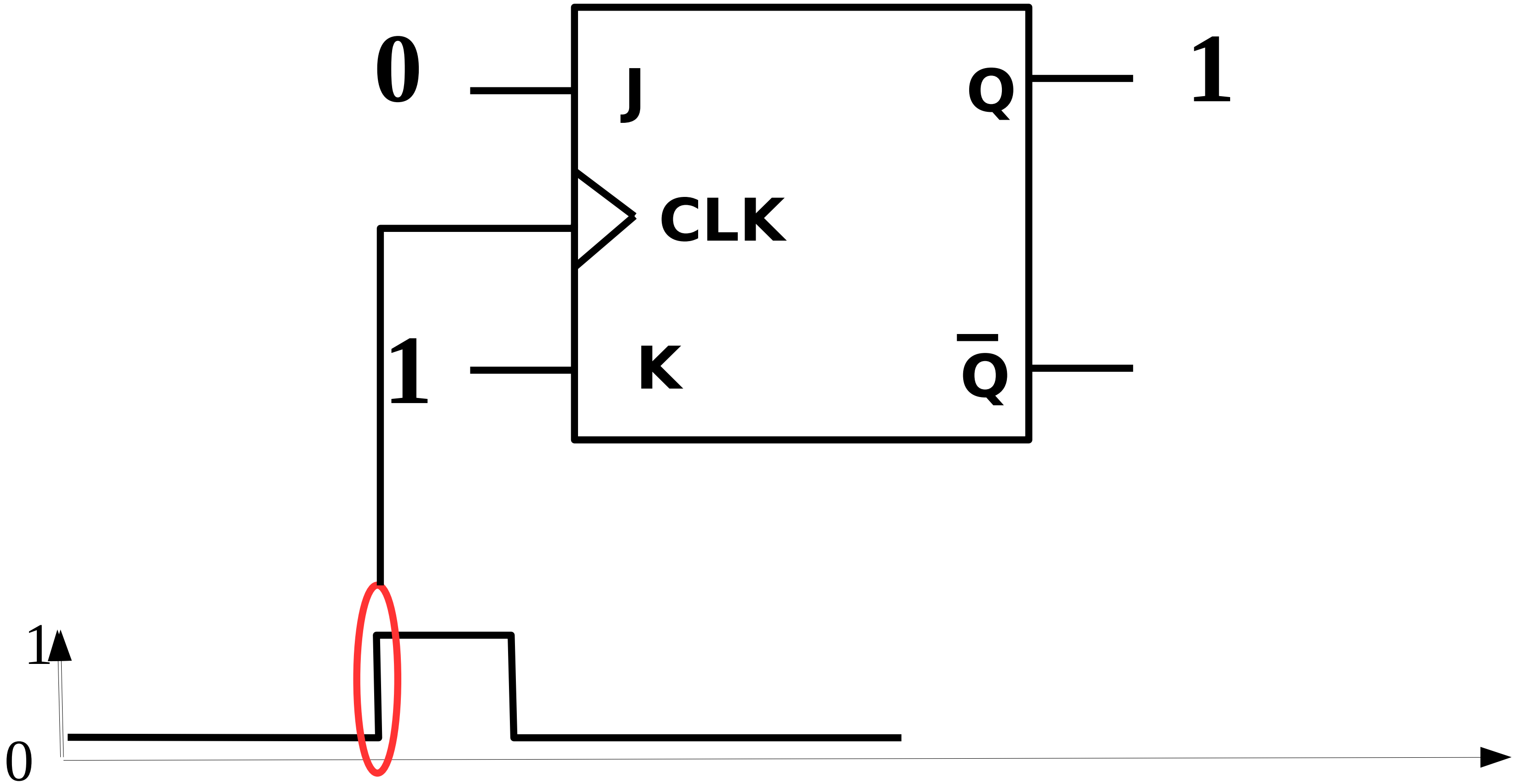
•Then no matter what you do the **J** or K pins the **output will remain unchanged**



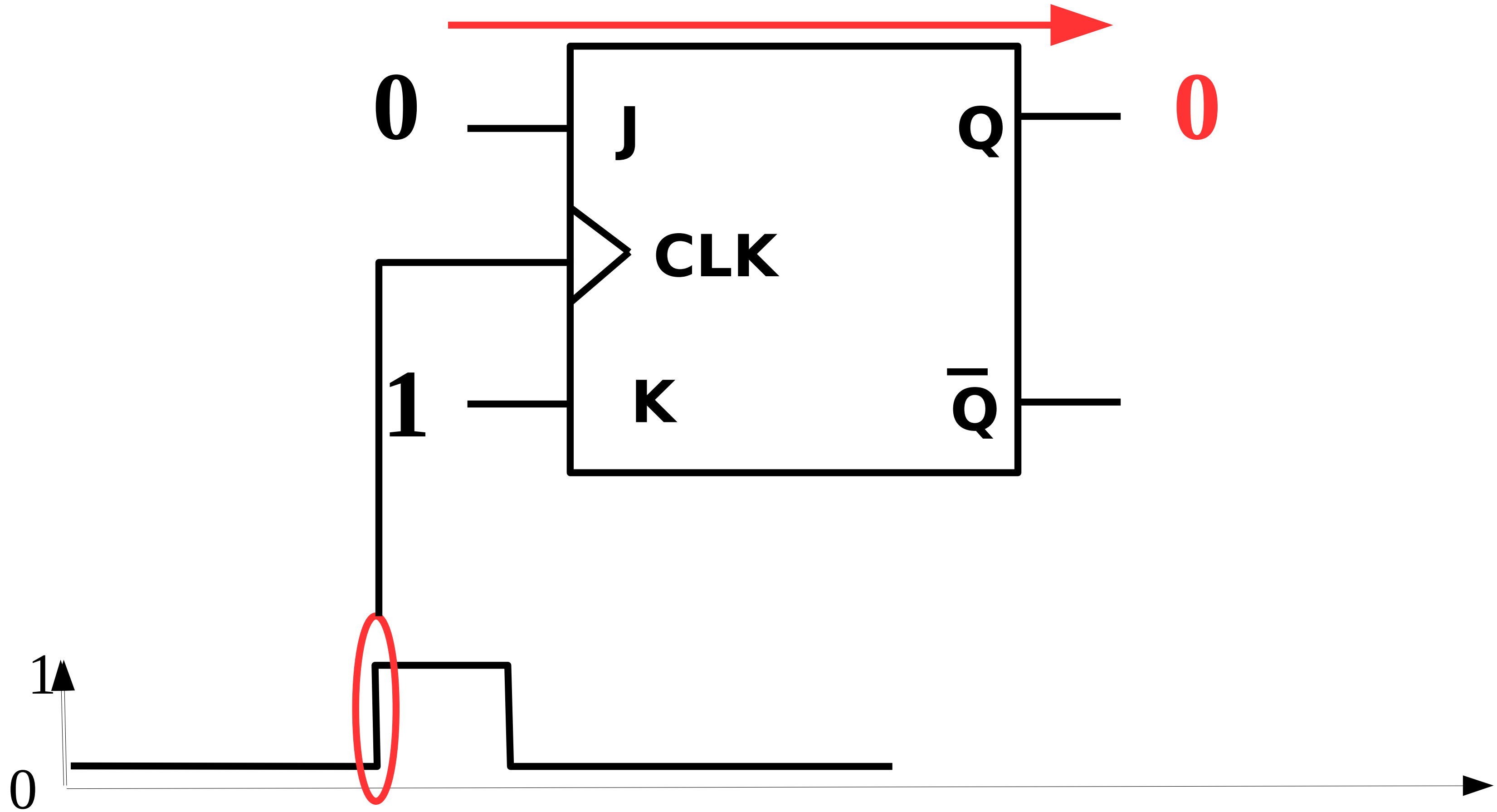
Storing a zero



Storing a zero

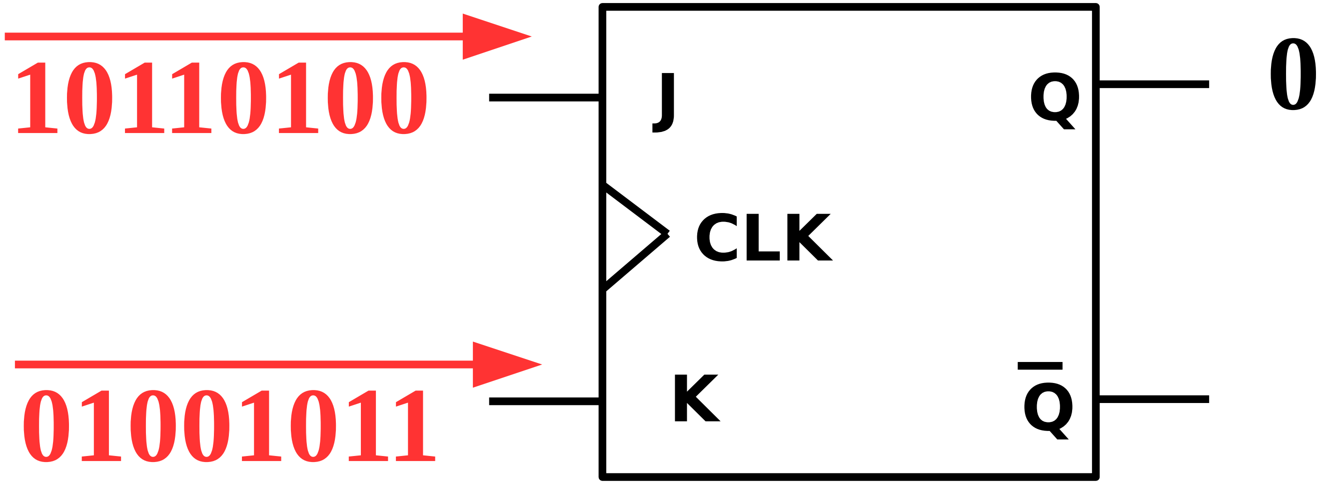


Storing a zero



How does it work.

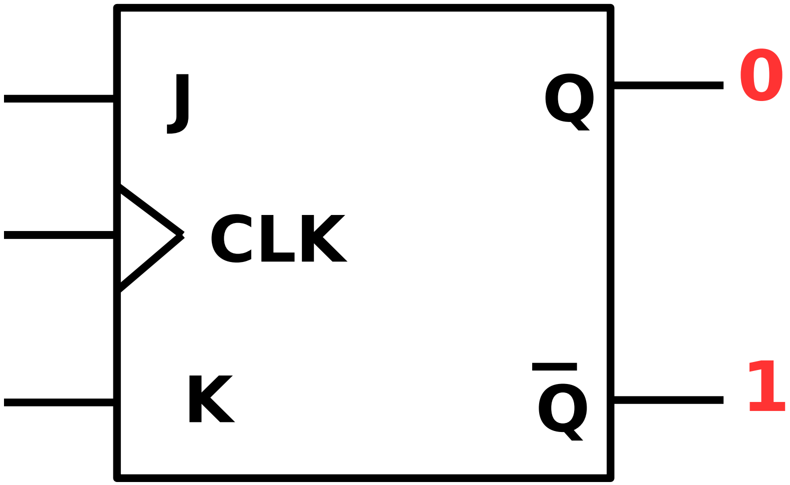
•Then no matter what you do the **J** or K pins the **output will remain unchanged**

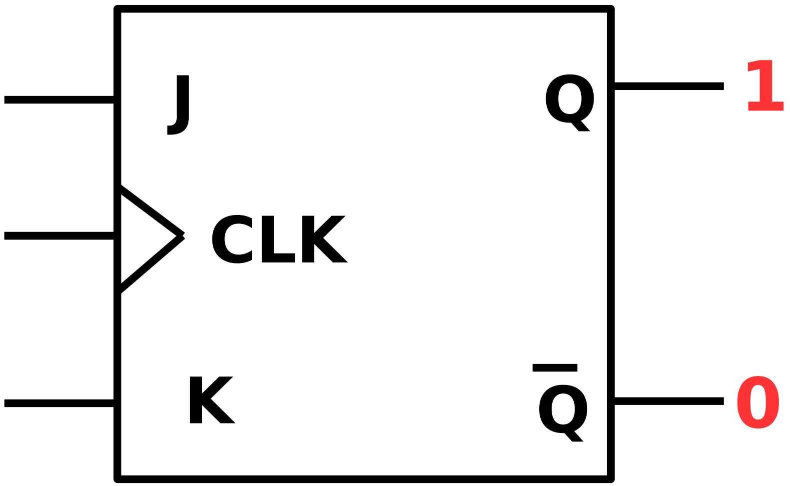


What is this $\overline{Q}$ pin??

- In Electronics a $\overline{\textbf{BAR}}$ above a letter means take the inverse: i.e. $\overline{\mathbf{1}}$ **actually means 0** and $\overline{\mathbf{0}}$ **means 1**.

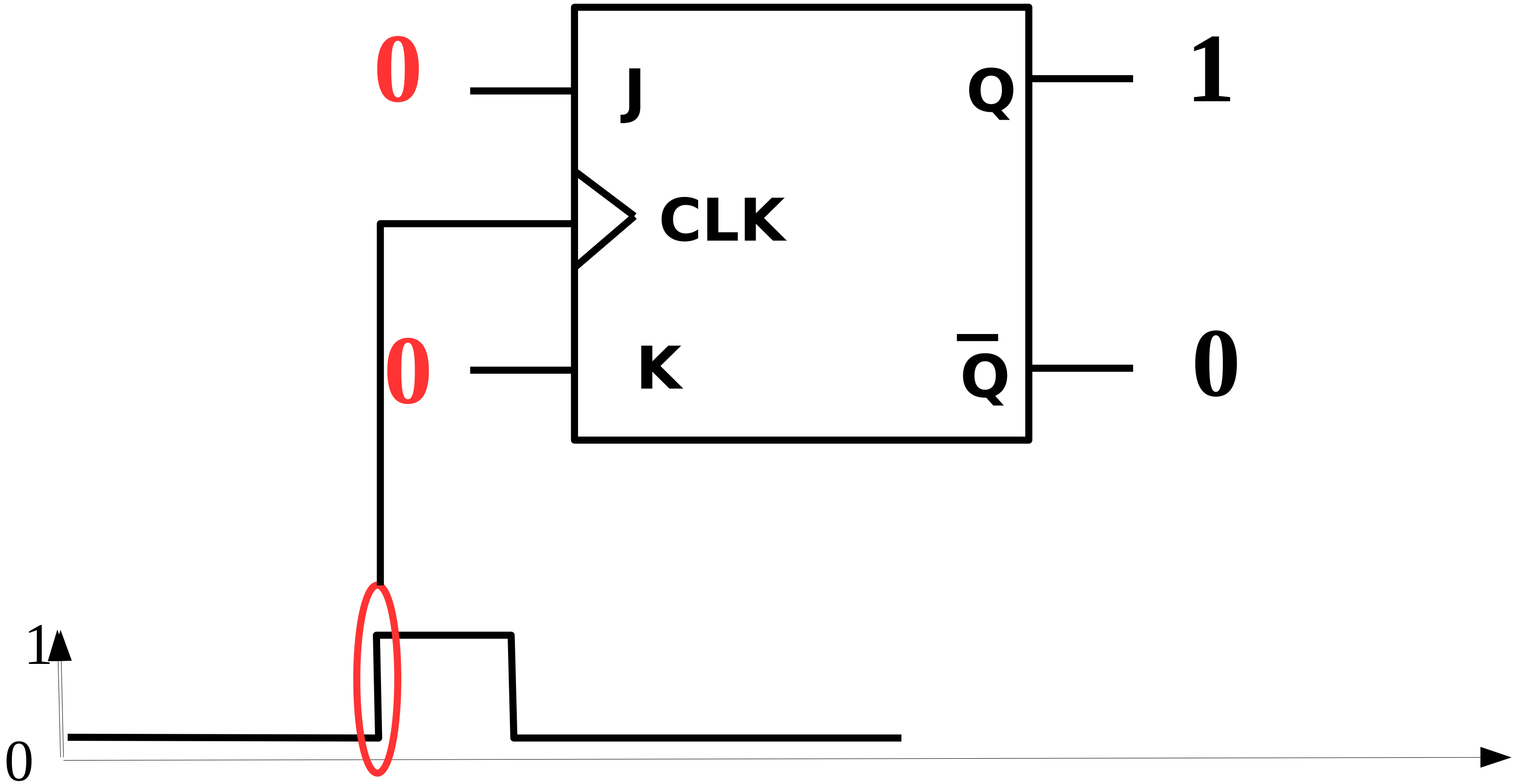
- This means that $\overline{\mathbf{Q}}$ output is always the exact opposite of what **Q** is i.e.:





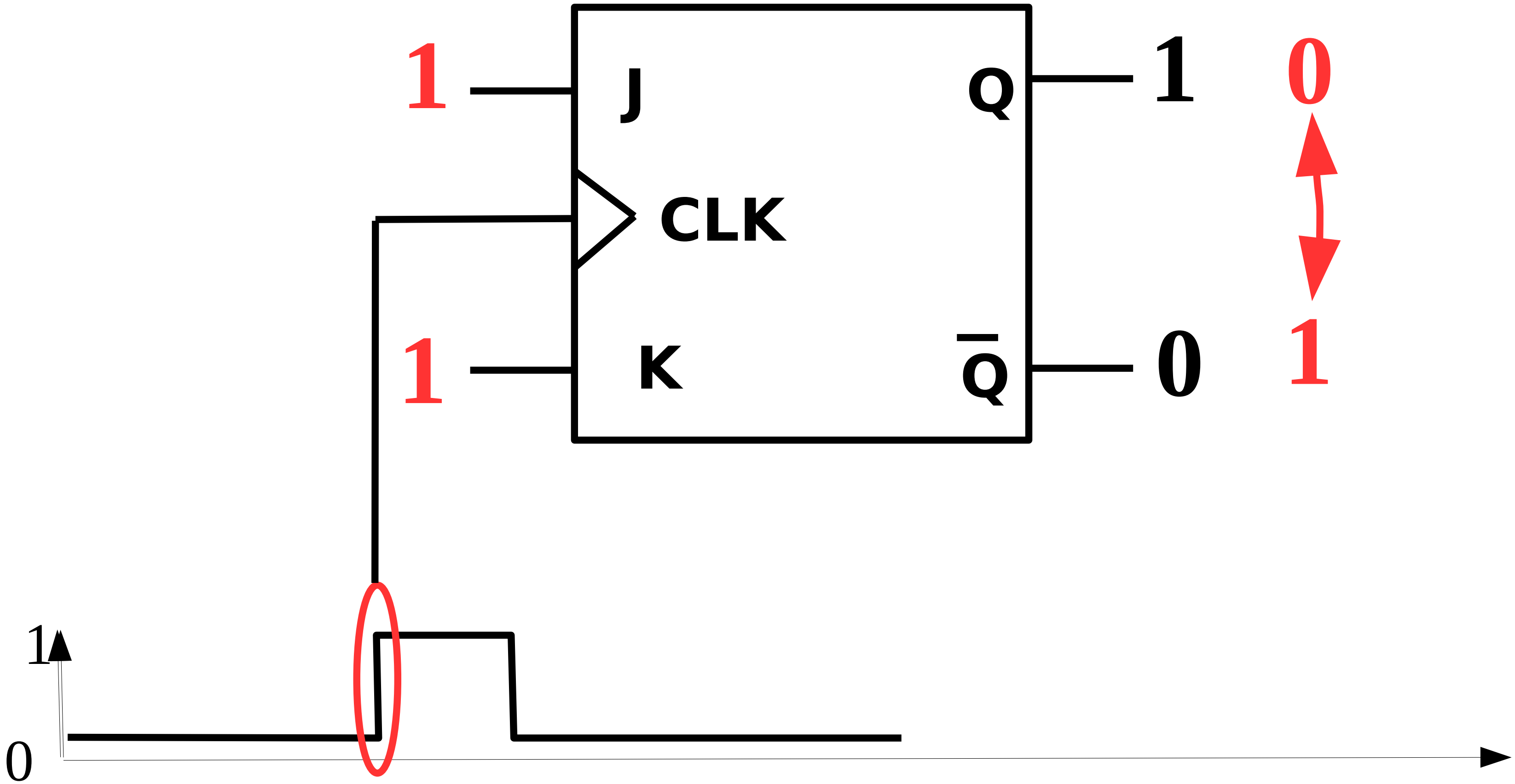
Last two things about the JK flip flop - first thing:

- If you set J=0 and K=0, then clock it – nothing happens.



Last two things about the JK flip flop – second thing thing:

- If you set J=1 and K=1, then clock it – the values of Q and $\overline{Q}$ are flipped.

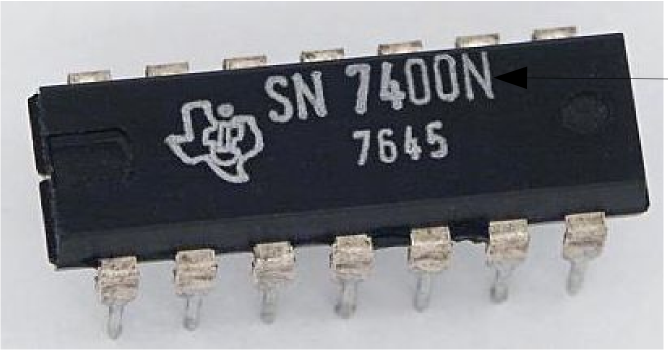


Here is the JK flipflop truth table

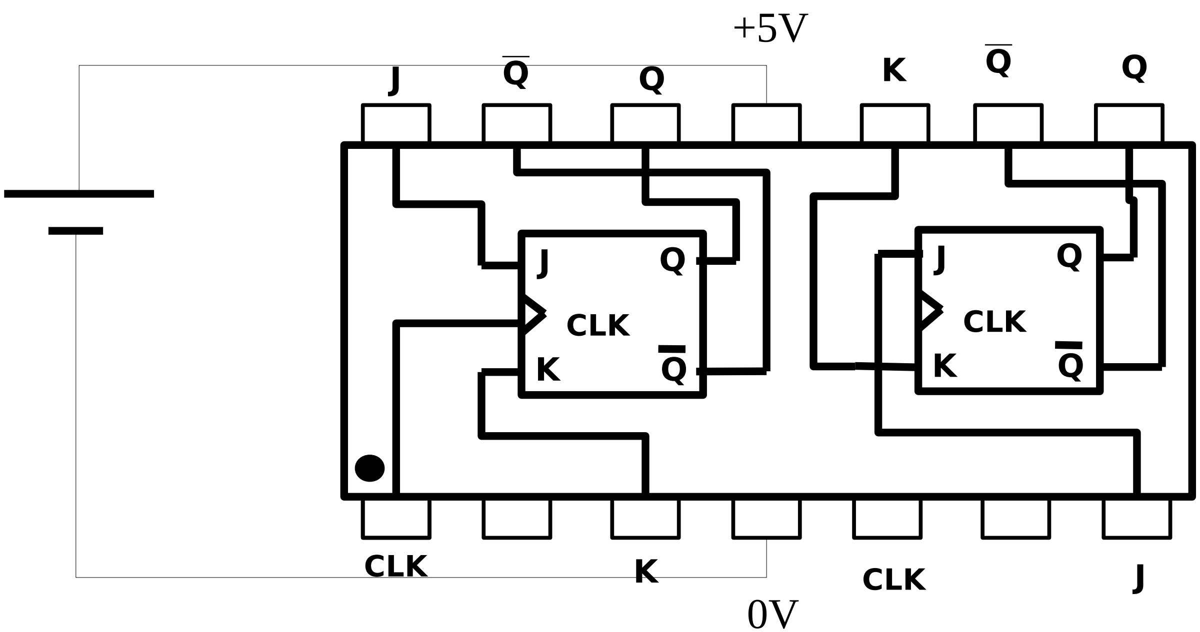
- In summary the truth table is as follows:

J	K	Q	Comment
0	0	Q	No change
1	0	1	Set Q to 1
0	1	0	Set Q to 0
1	1	flip	Flip

And this is what they look like

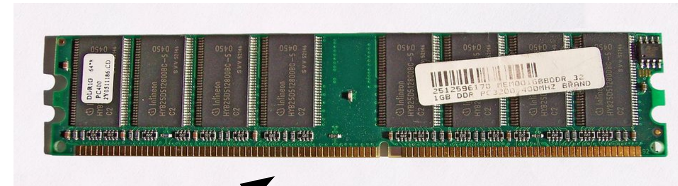


7473

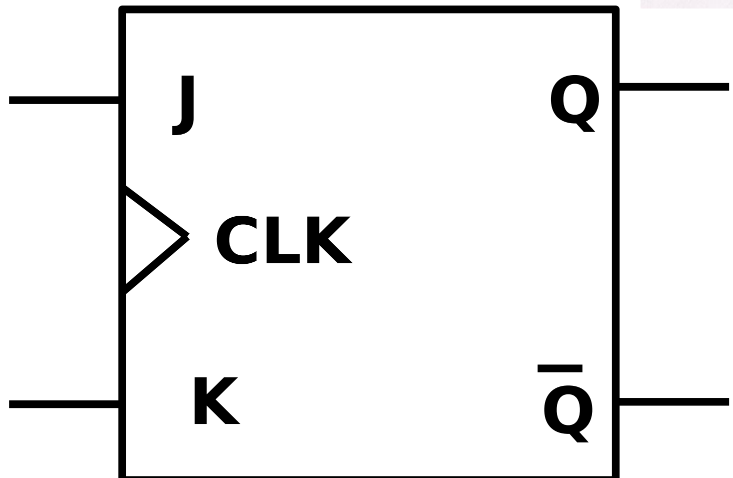


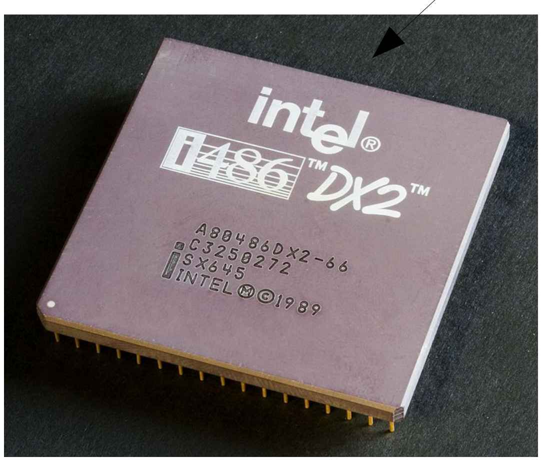
These flip-flops are used ever where in electronics to store 1's and 0's

Intel Free Press



László Szalai





Andrew Dunn

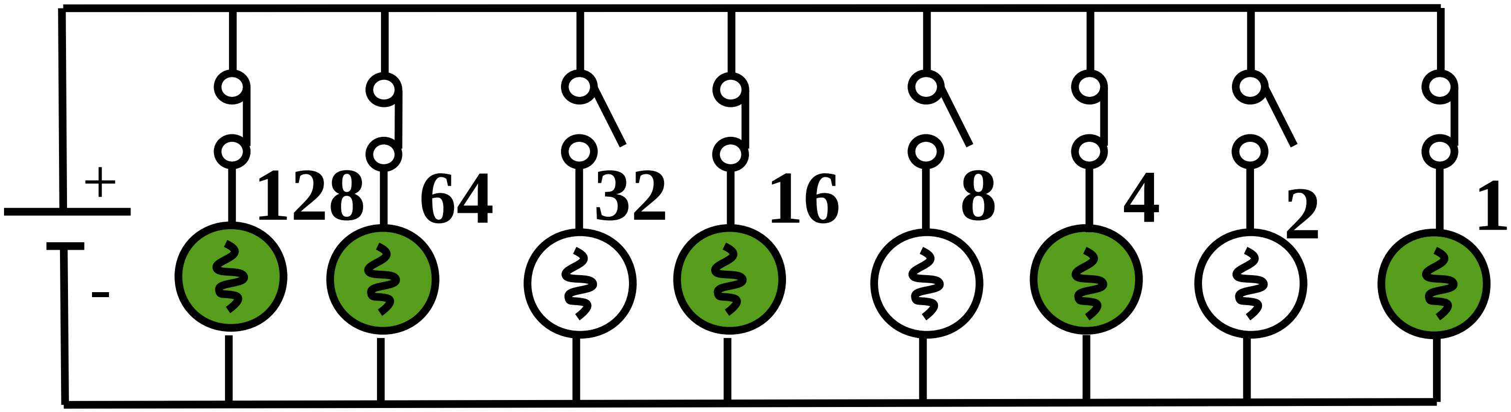
- But to demonstrate it's usefulness I'm going to show you something more simple.

Outline of the lecture

- Recap of last lecture
- Recap of gates - Mini quiz
- Figuring out what electronic circuits do
- **Making electronics remember things**
    - Flip flops
    - **Serial to parallel converters**

Serial to parallel converter

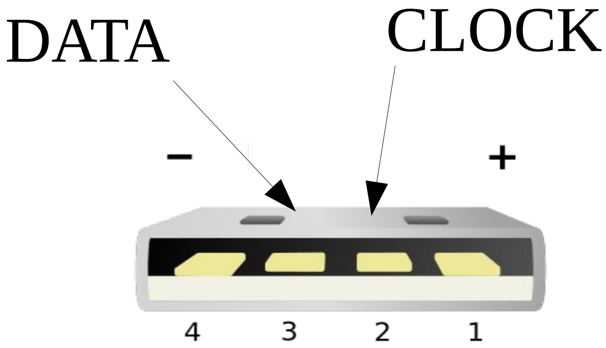
•Do you remember from lecture 1 that electronic circuits store all numbers/information as a series of on and off signals?



•This robot is no exception he transmits all information internally using a set of 32 copper wires.

Serial to parallel converter

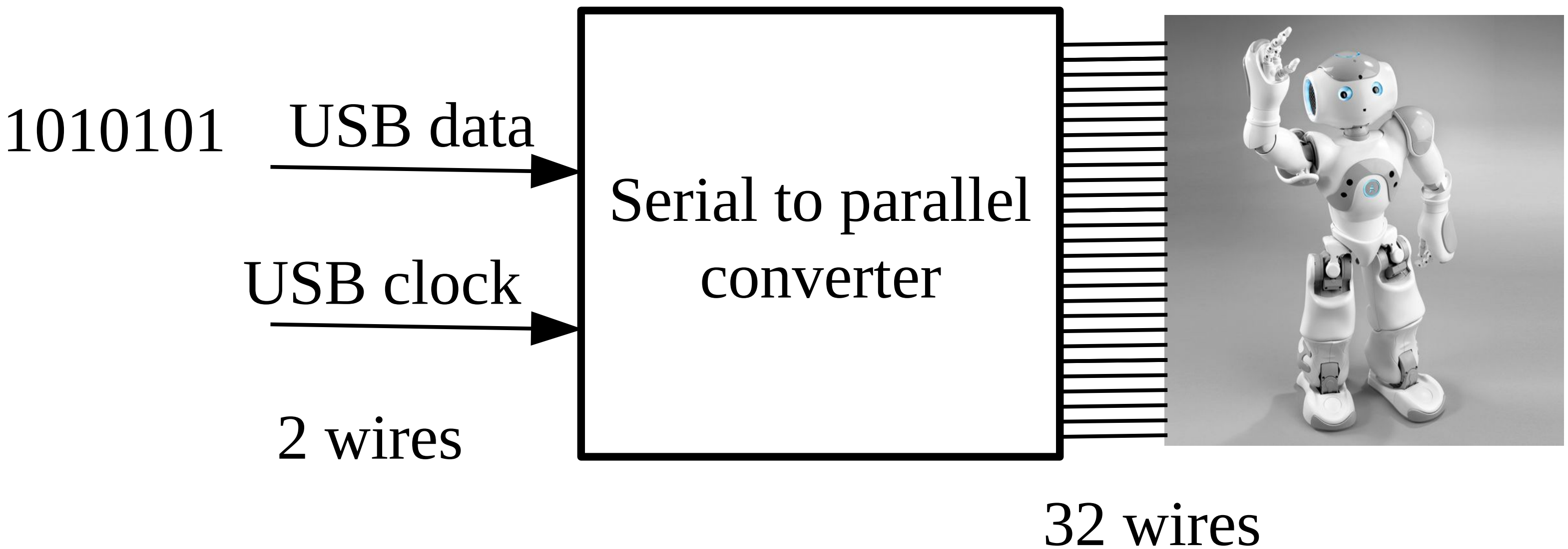
•To program him we need to use a USB cable:



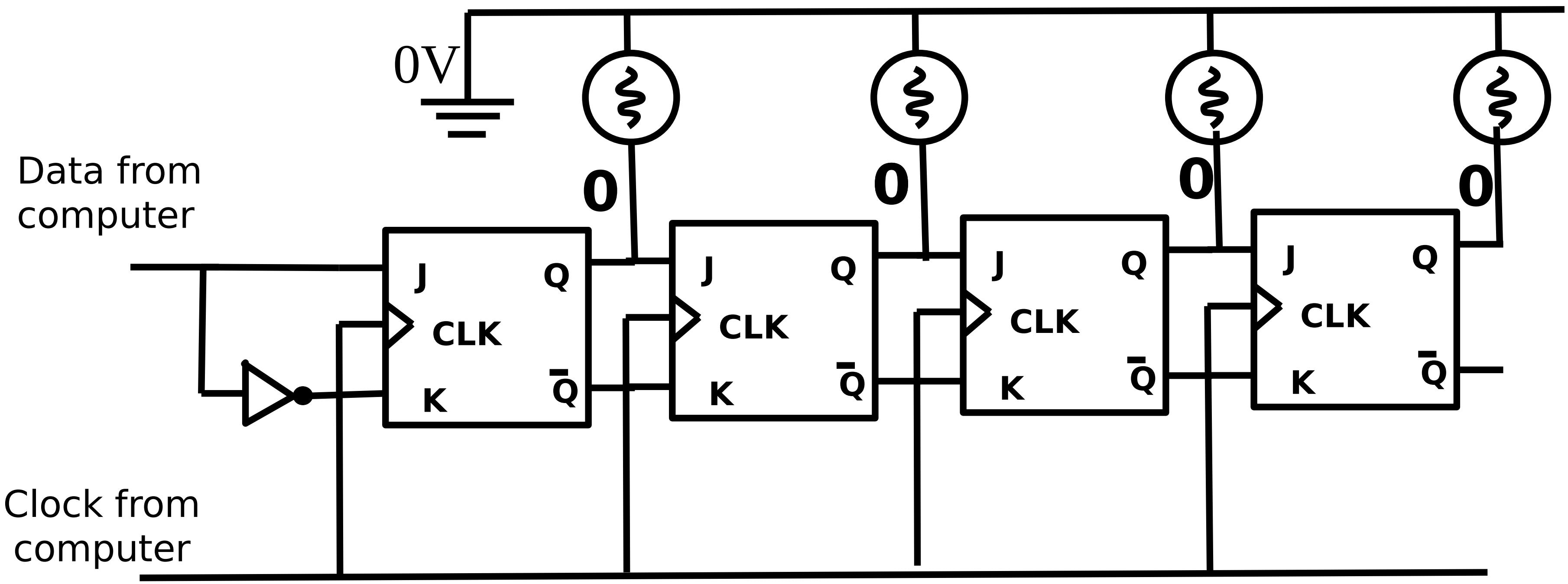
•Unfortunately to save money a USB cable has one wire that can carry data and one clock signal.

•So the robot needs convert the data coming along the 1 USB cable to the 32 signals he uses internally.

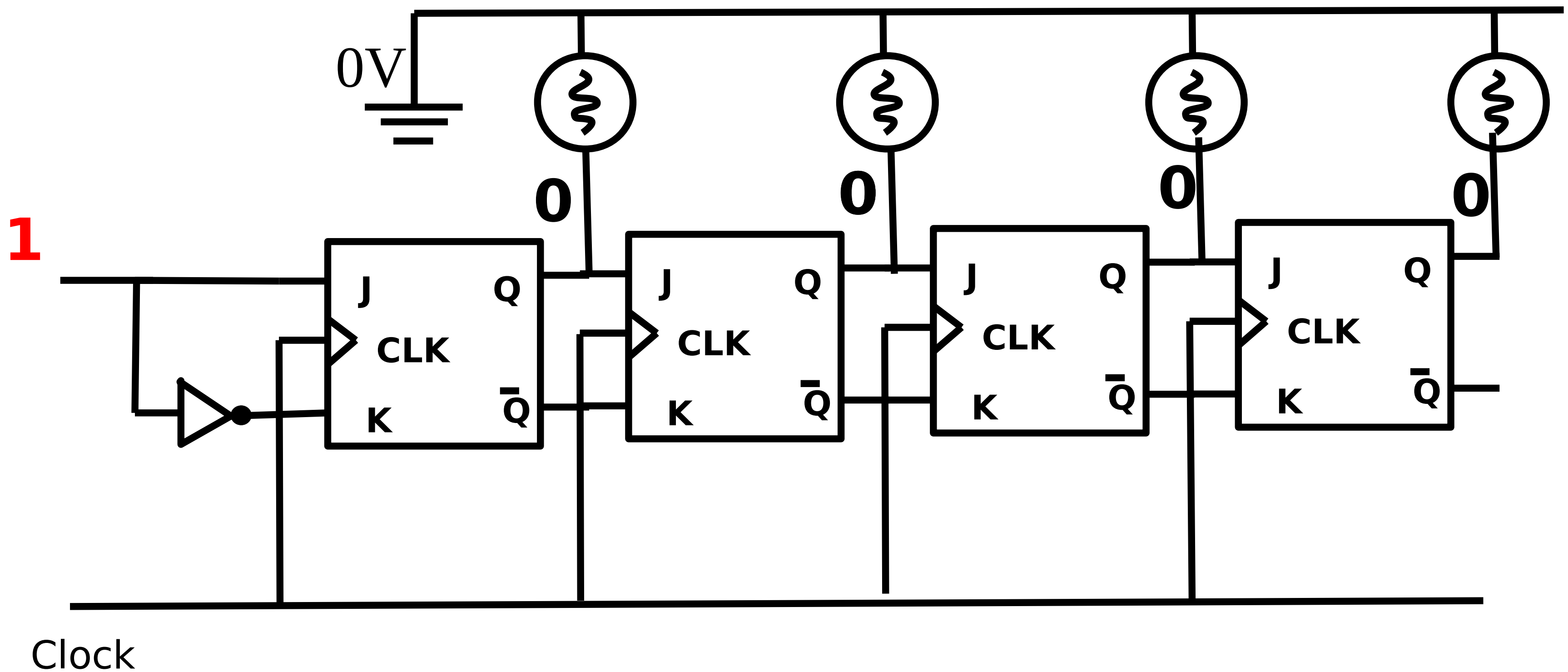
Serial to parallel converter



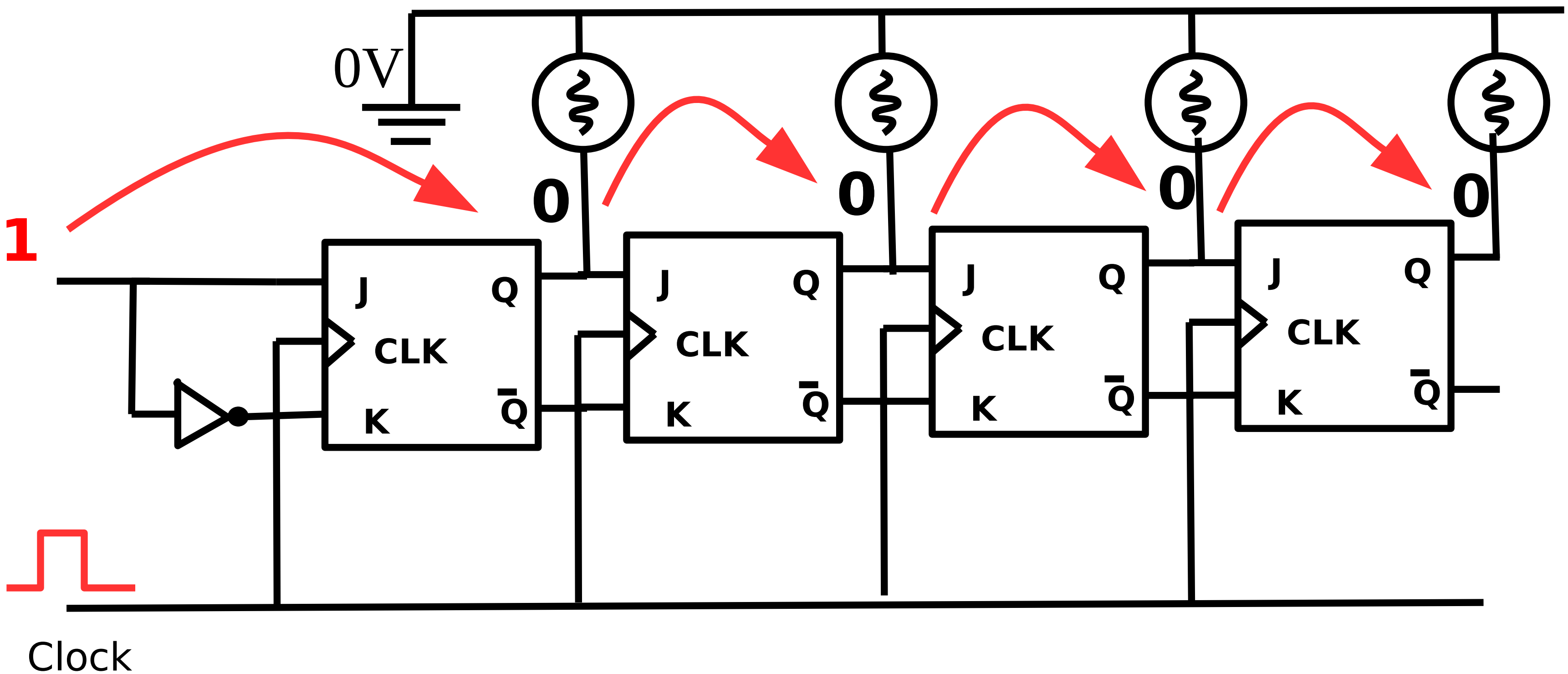
Serial to parallel converter



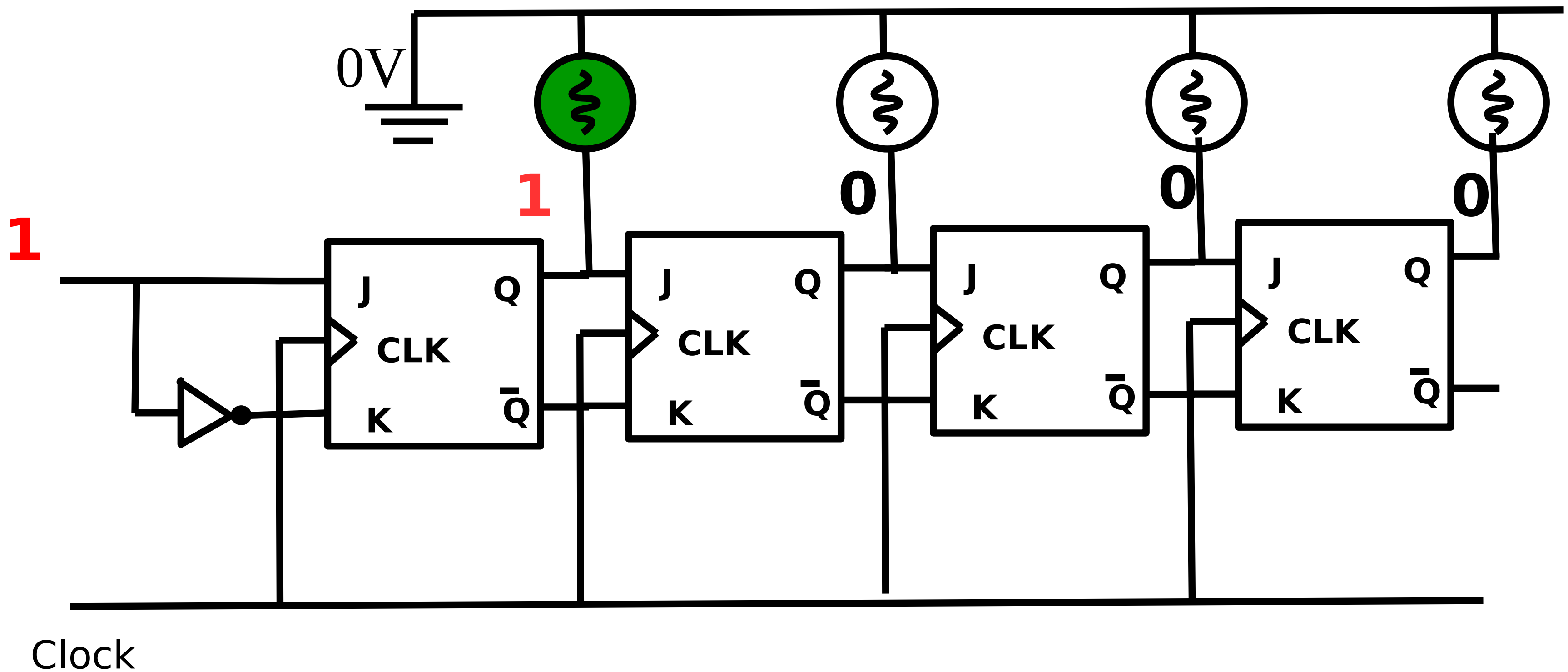
Serial to parallel converter



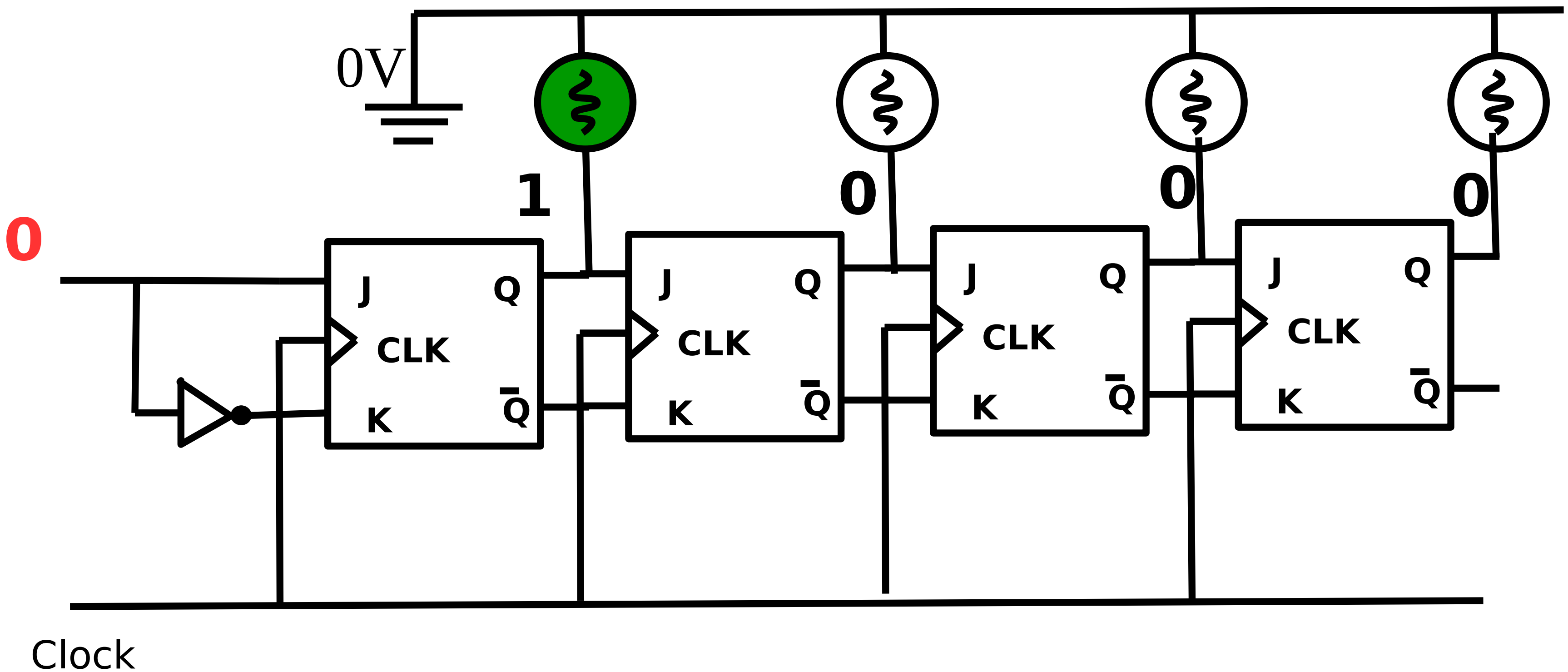
Serial to parallel converter



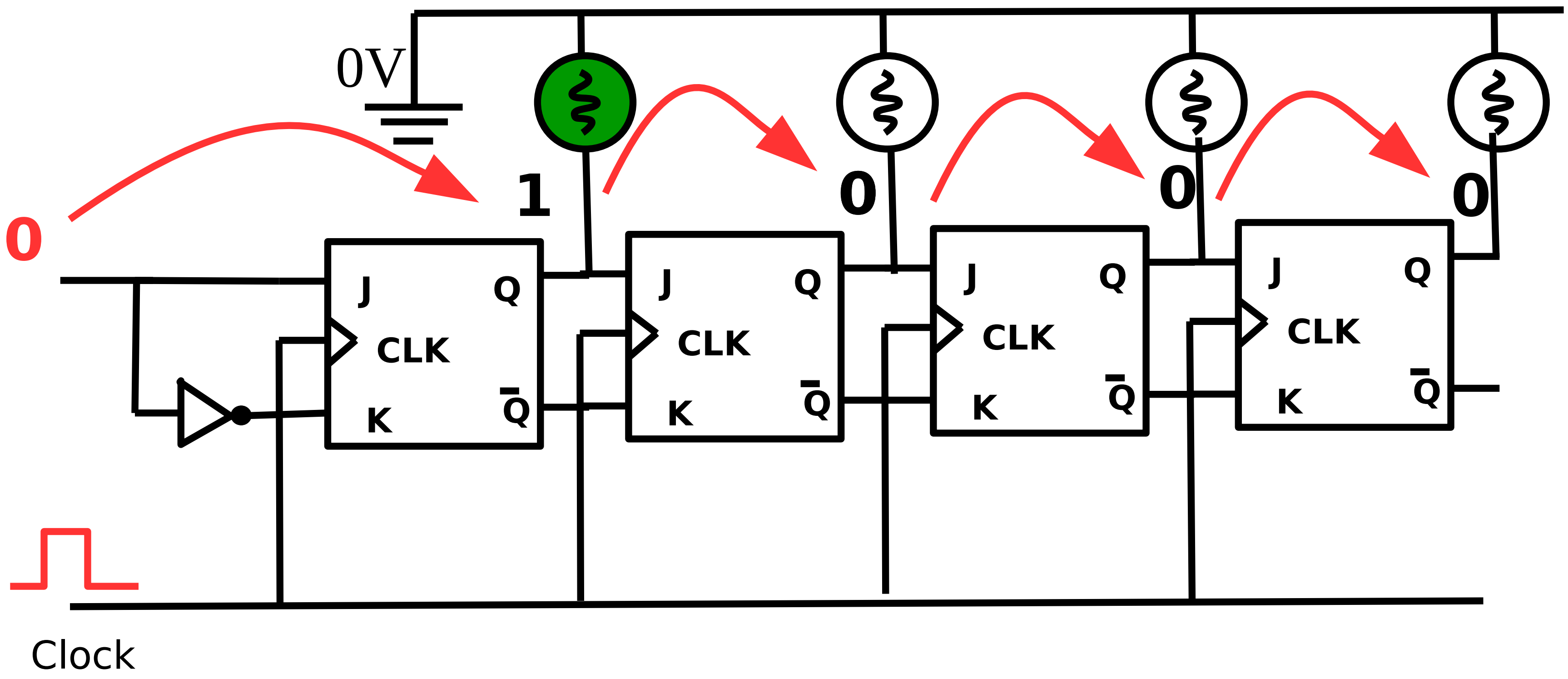
Serial to parallel converter



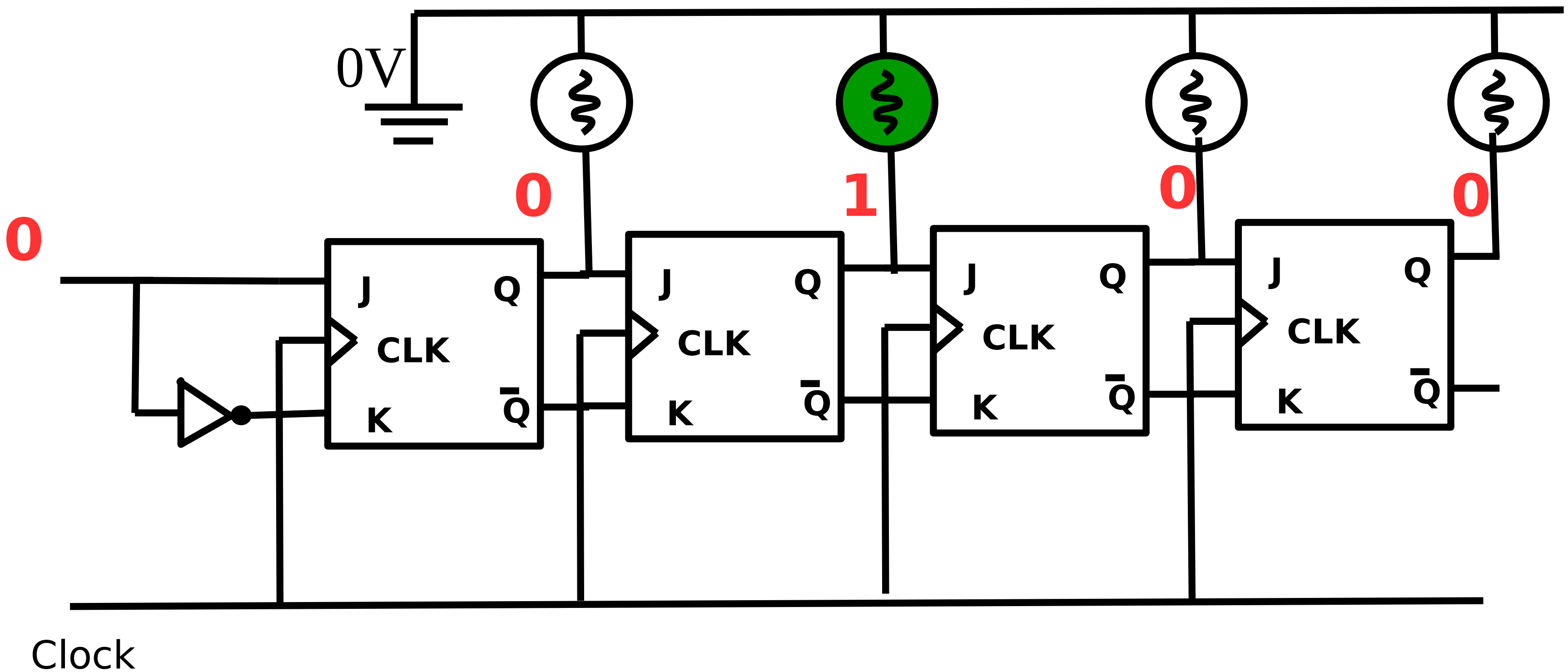
Serial to parallel converter



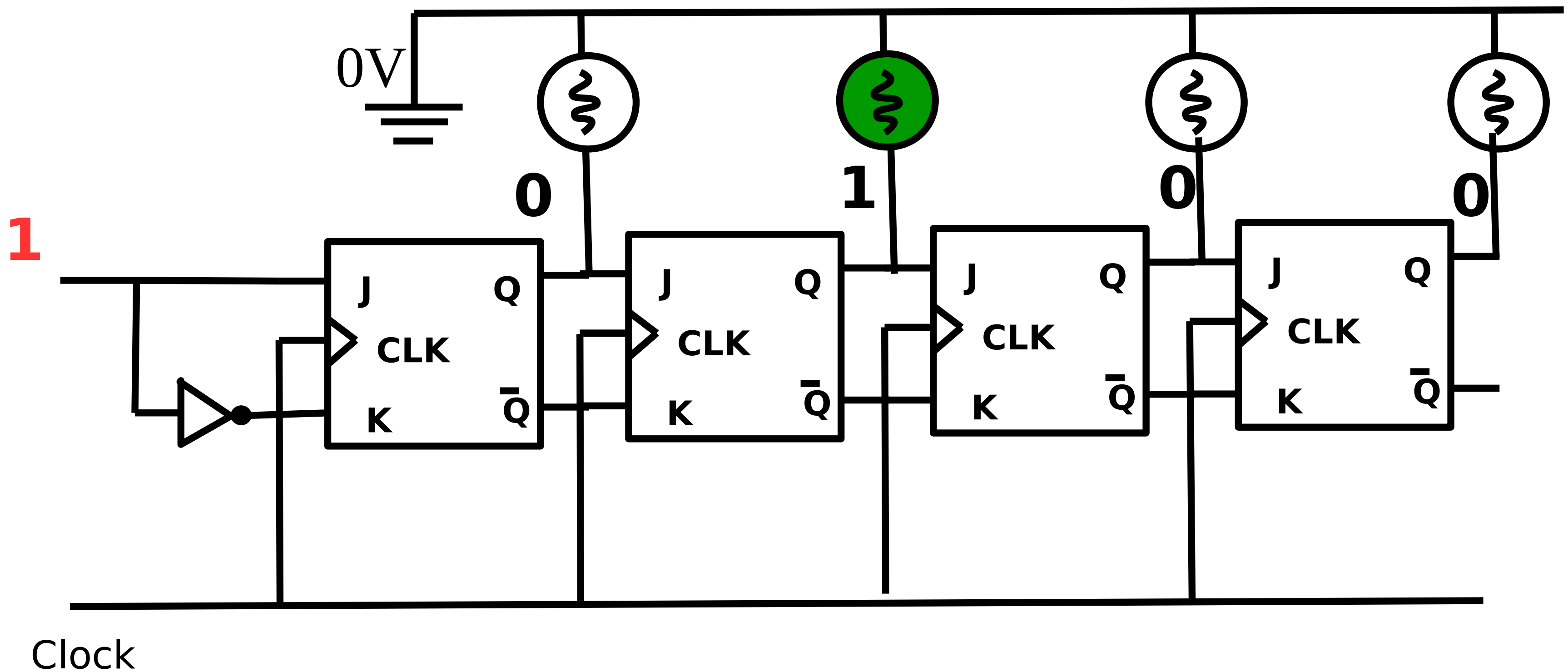
Serial to parallel converter



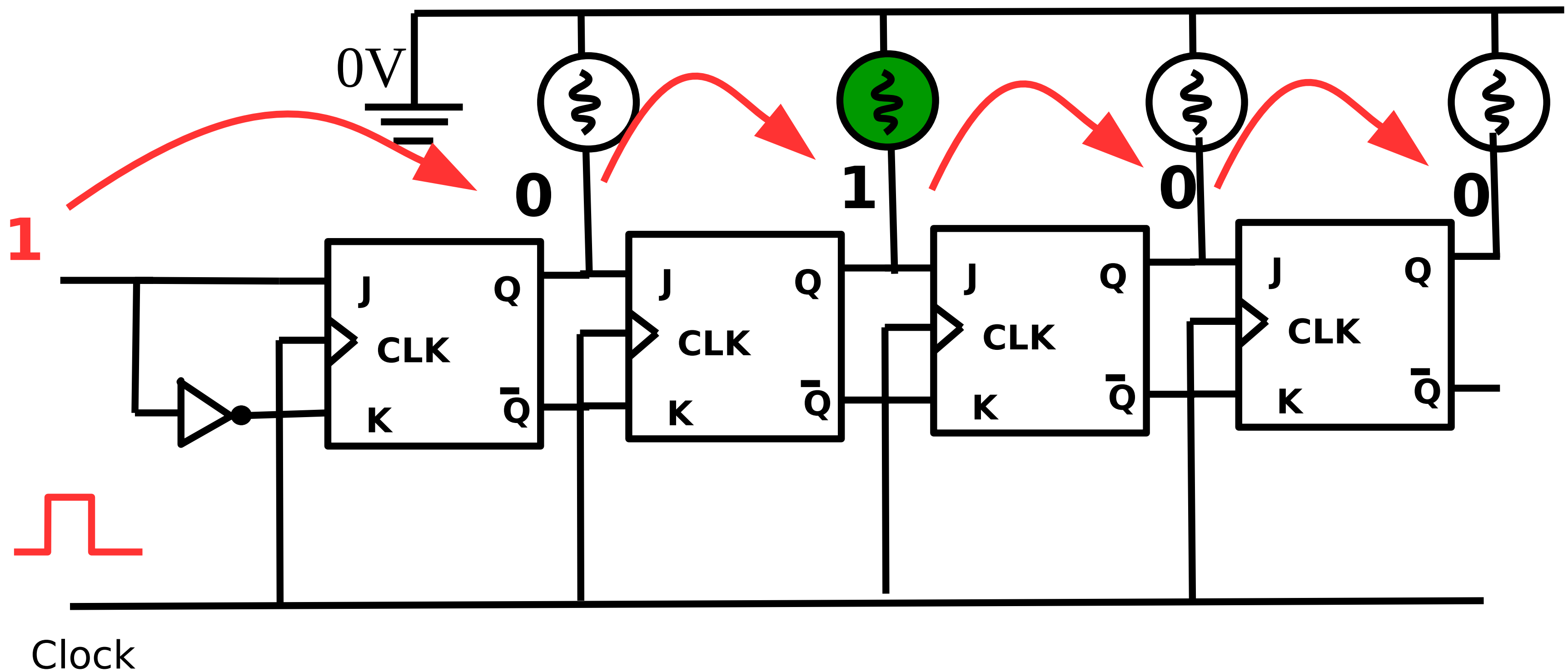
Serial to parallel converter



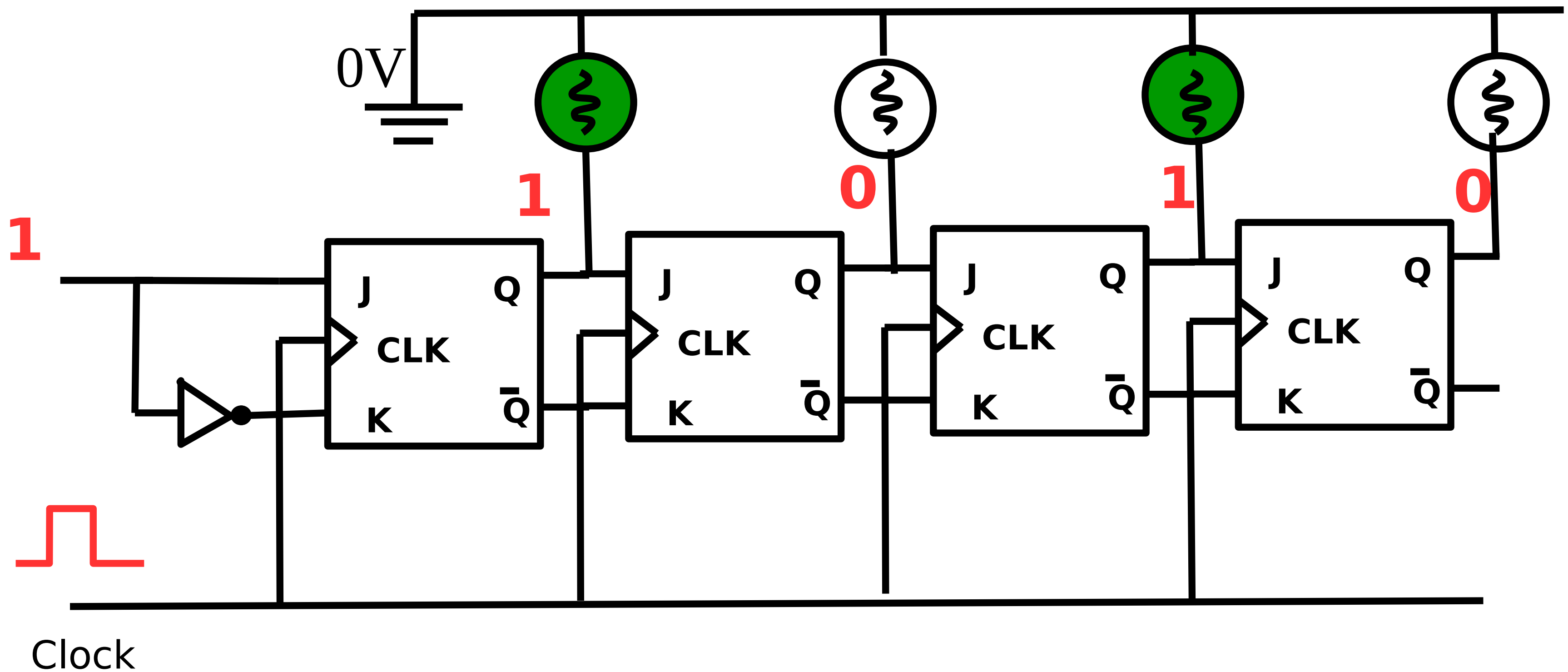
Serial to parallel converter



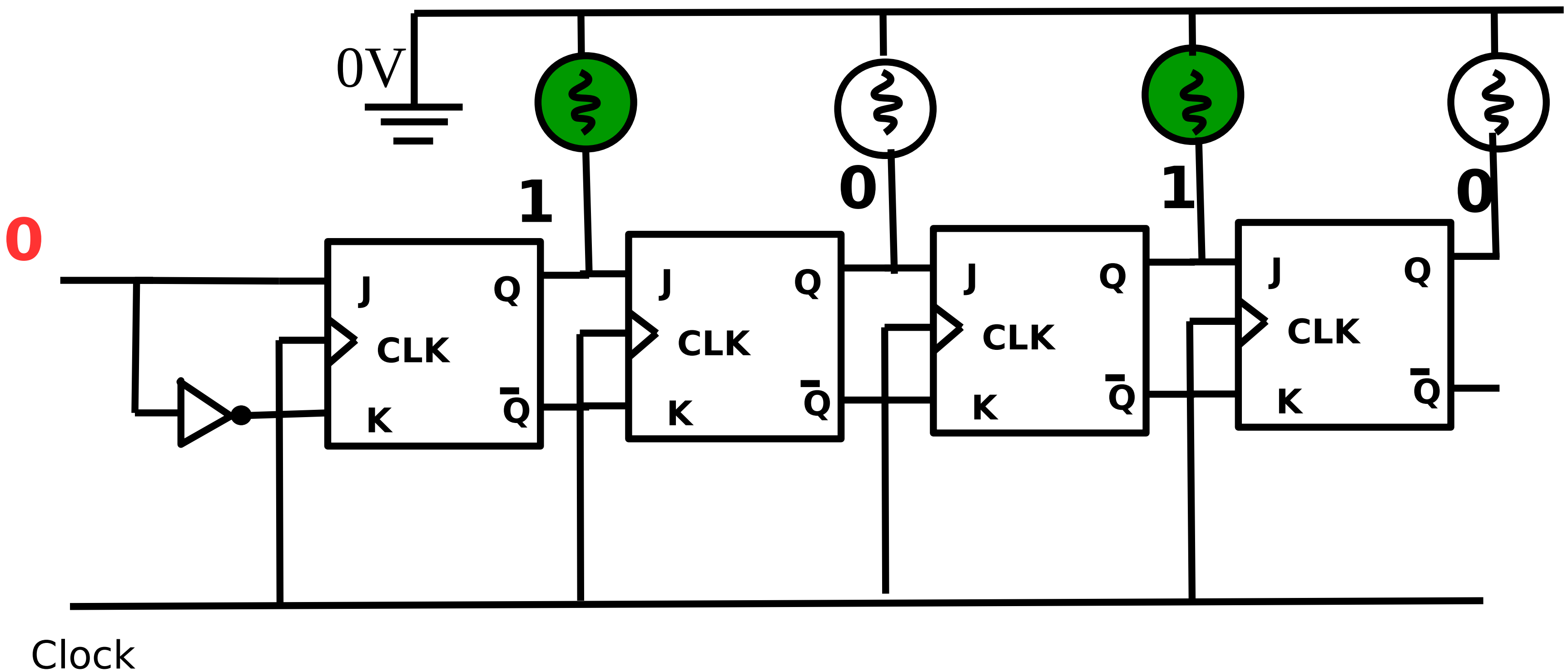
Serial to parallel converter



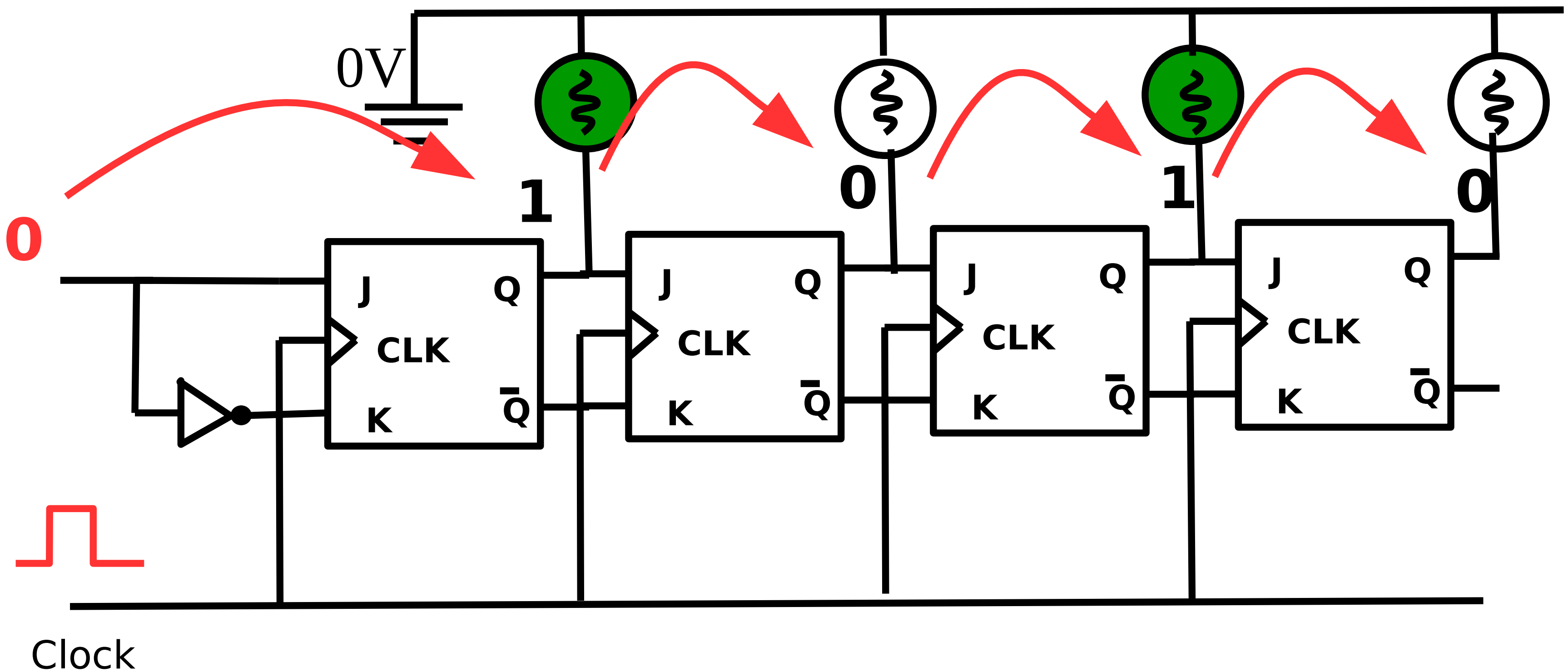
Serial to parallel converter



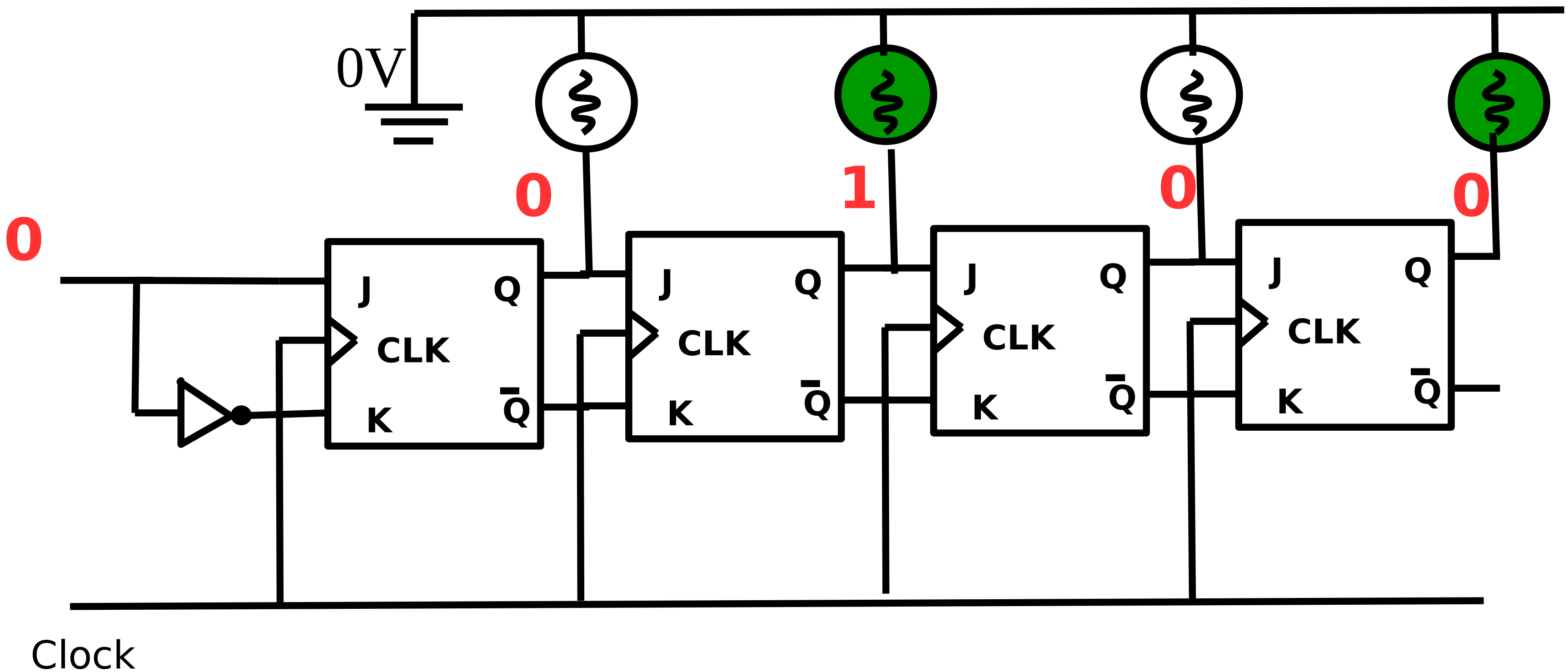
Serial to parallel converter



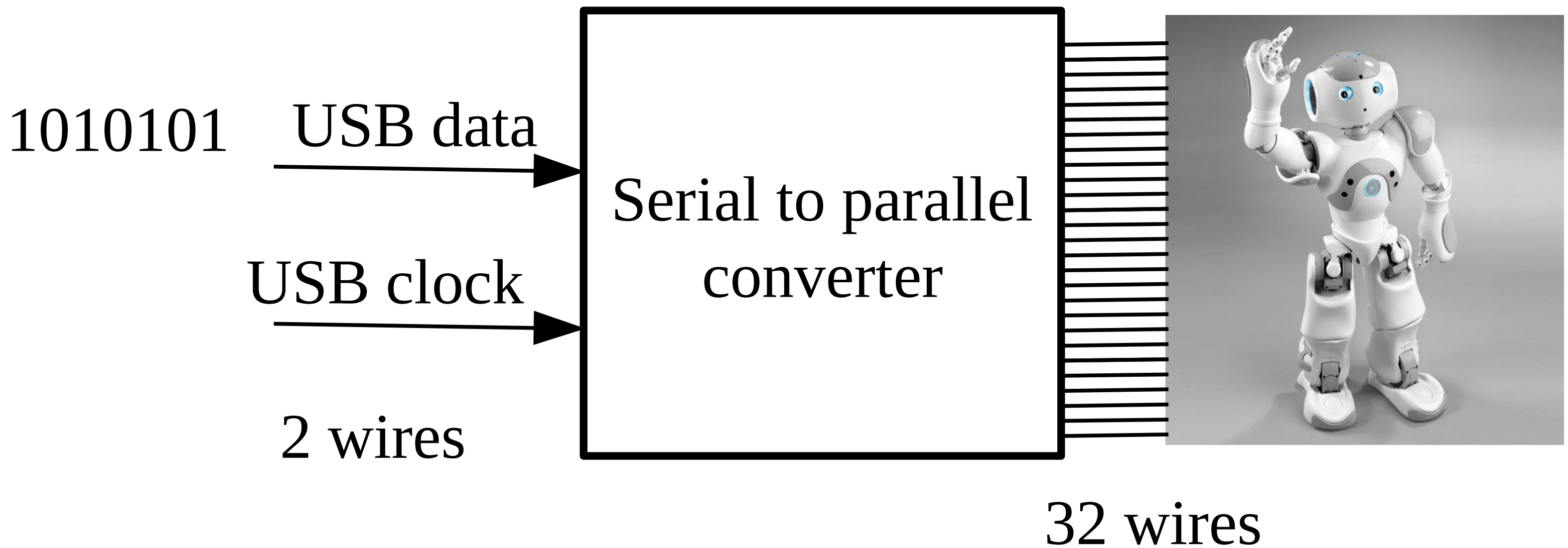
Serial to parallel converter



Serial to parallel converter



Serial to parallel converter



Summary

- Recap of last lecture
- Recap of gates - Mini quiz
- Figuring out what electronic circuits do
- Making electronics remember things
    - Flip flops
    - Serial to parallel converters