University of Nottingham

Electromechanical devices MM2EMD

Lecture 3 - Digital design and race times

Dr. Roderick MacKenzie
roderick.mackenzie@nottingham.ac.uk
Summer 2015

@rcimackenzie

Outline of the lecture

- **Recap of last lecture**

- Using JK flip flops to run motors

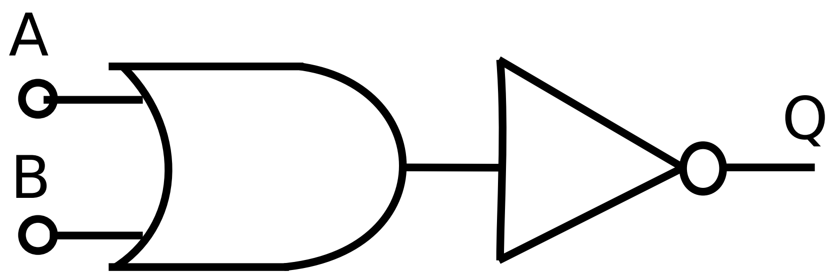
- Digital design
    - What is digital design?
    - How to do it.
    - Where you might meat digital design
    - FPGAs

- Race times

Recap: Understanding more complex digital circuits

- If we have a circuit comprising of multiple gates we can analyze it by first writing out the truth table containing all possible input combinations.

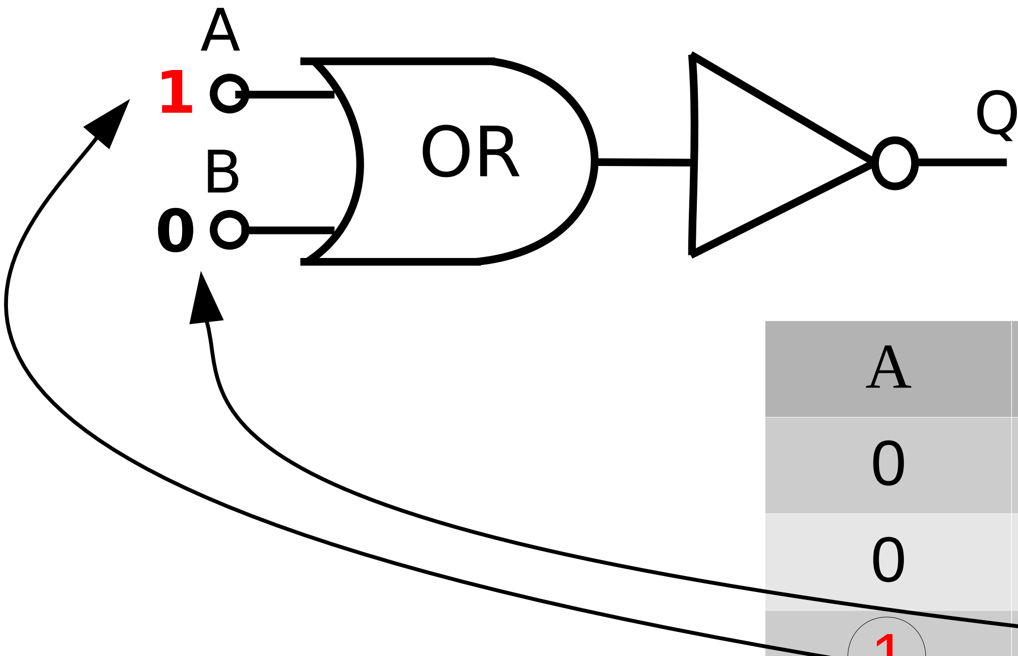
- With the output column left blank.



A	B	Q
0	0	
0	1	
1	0	
1	1	

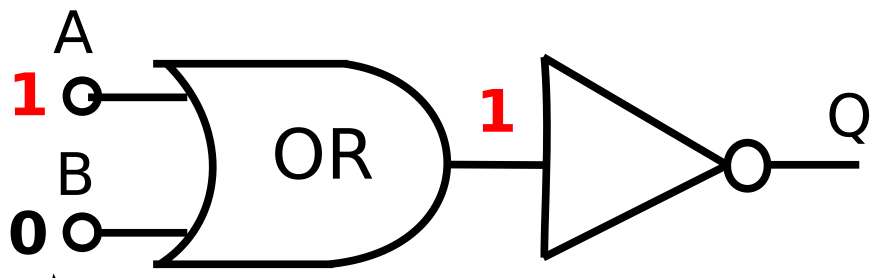
Recap: Understanding more complex digital circuits

•Then one line at a time, we propagate the values through the circuit.



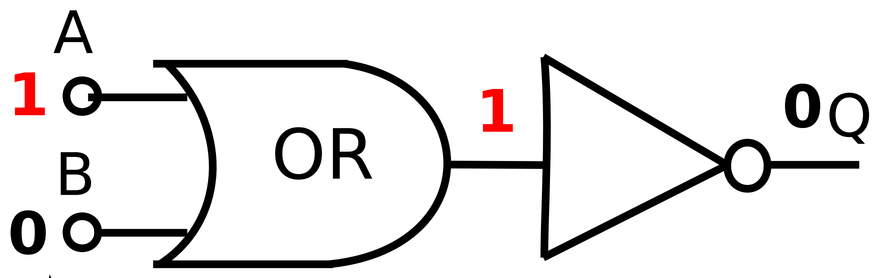
A	B	Q
0	0	
0	1	
1	0	
1	1	

Recap: Understanding more complex digital circuits



A	B	Q
0	0	
0	1	
1	0	
1	1	

Recap: Understanding more complex digital circuits

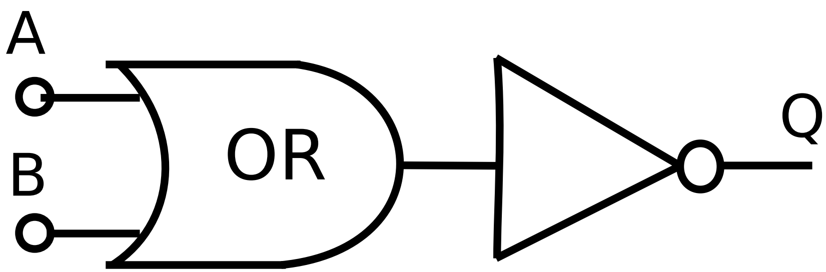


A	B	Q
0	0	
0	1	
1	0	0
1	1	

Then write the table in the output column.

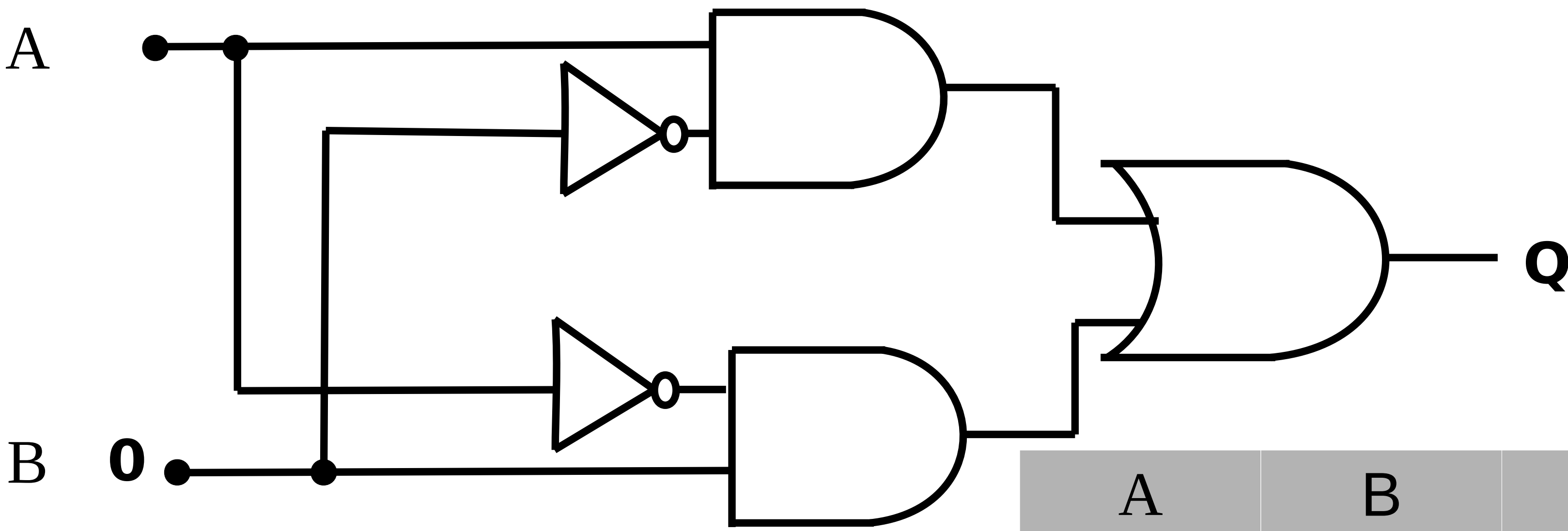
Recap: Understanding more complex digital circuits

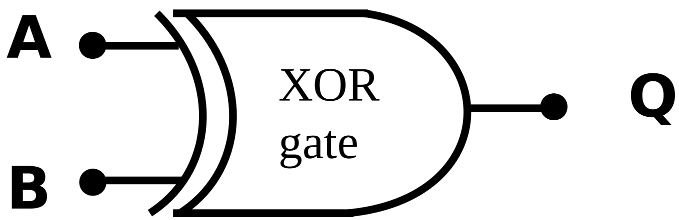
If we do this with all inputs we can figure out what the circuit does



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

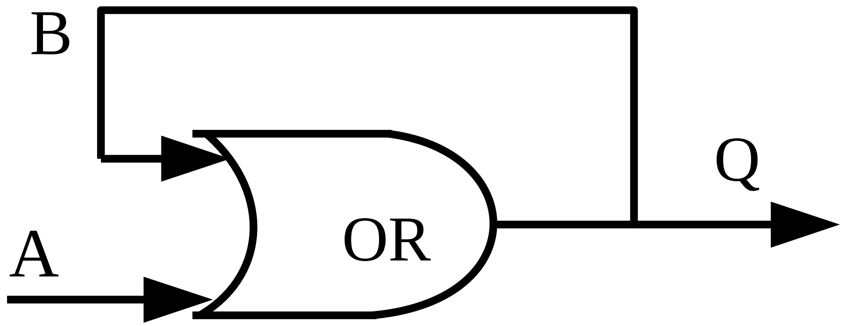
Recap: We did this with the more complex XOR gate circuit:





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

Recap: Basic latches

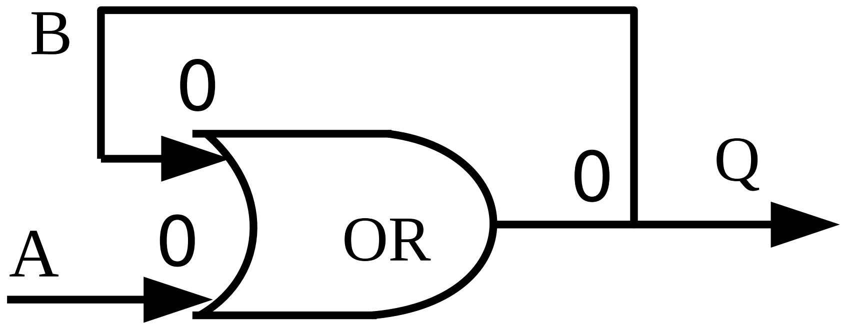


- I introduced the principle of a latch.

- Latches are used to remember past electronic events.

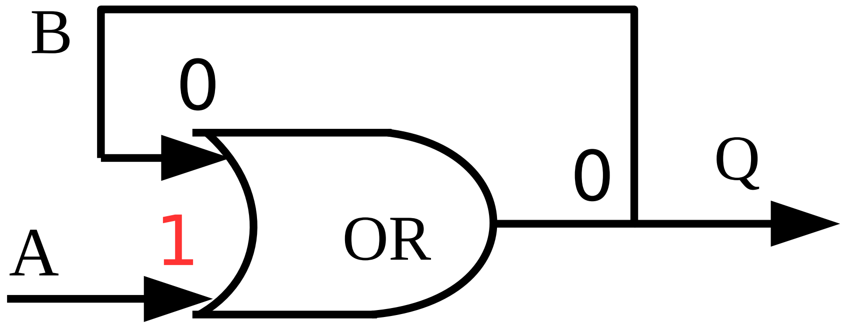
- A latch always has some type of feed back element, here is an example....

Recap: Basic latches



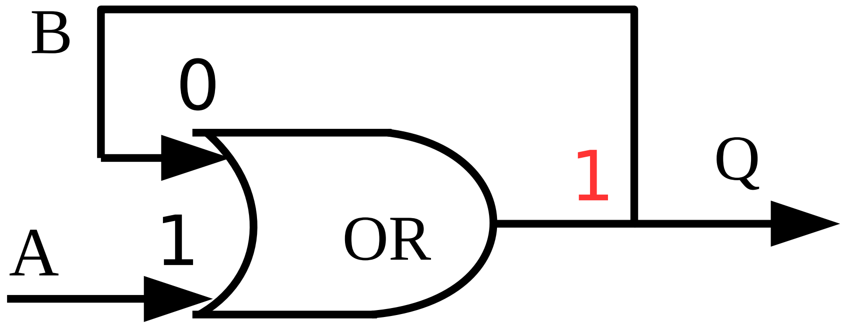
- If initially A, B and Q are zero

Recap: Basic latches



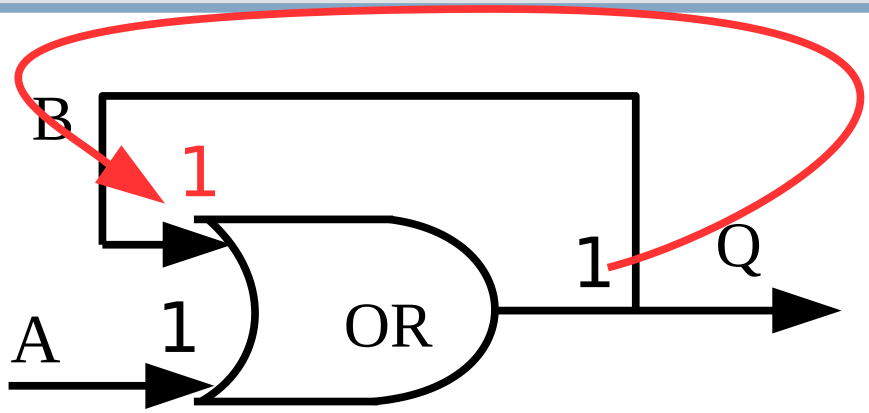
- If initially A, B and Q are zero
- Then you make input A=1

Recap: Basic latches



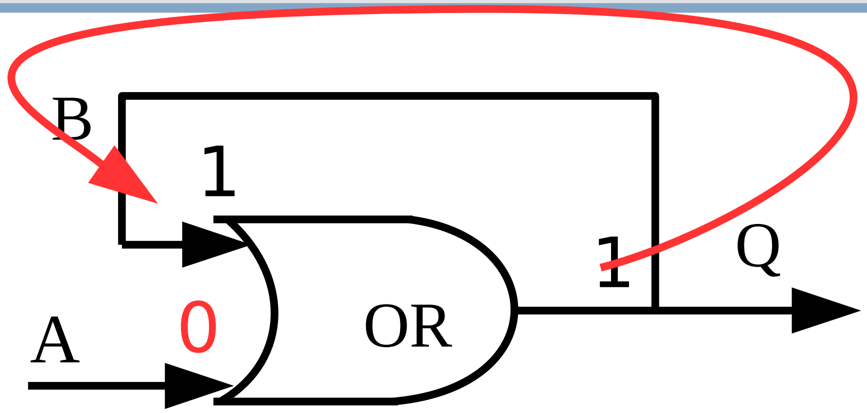
•This will make the output=1

Recap: Basic latches



- It will also make the input B, 1

Recap: Basic latches



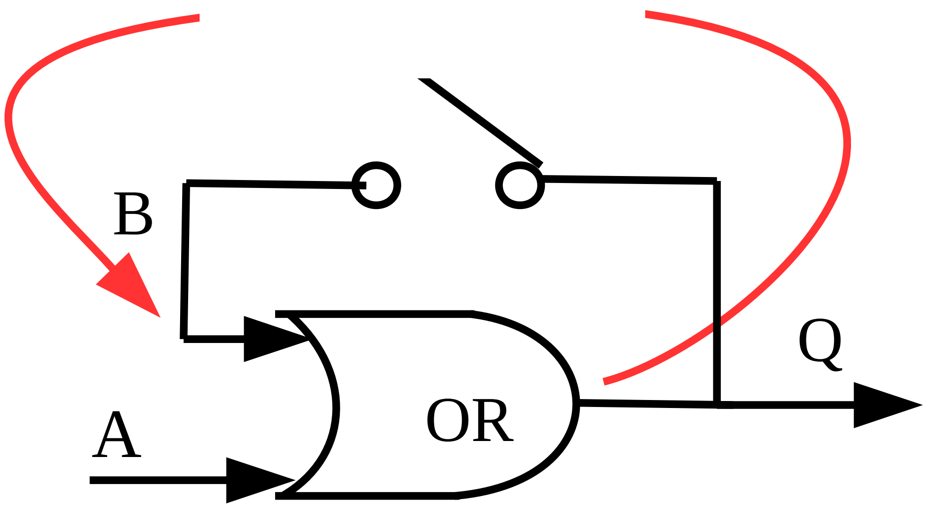
•Now if we again make input A 0

•No matter what we do the A the output will always remain 1 because of the feedback.

•The OR gate latch has remembered it was turned on due the the feedback – **it has latched**.

Recap: Latches

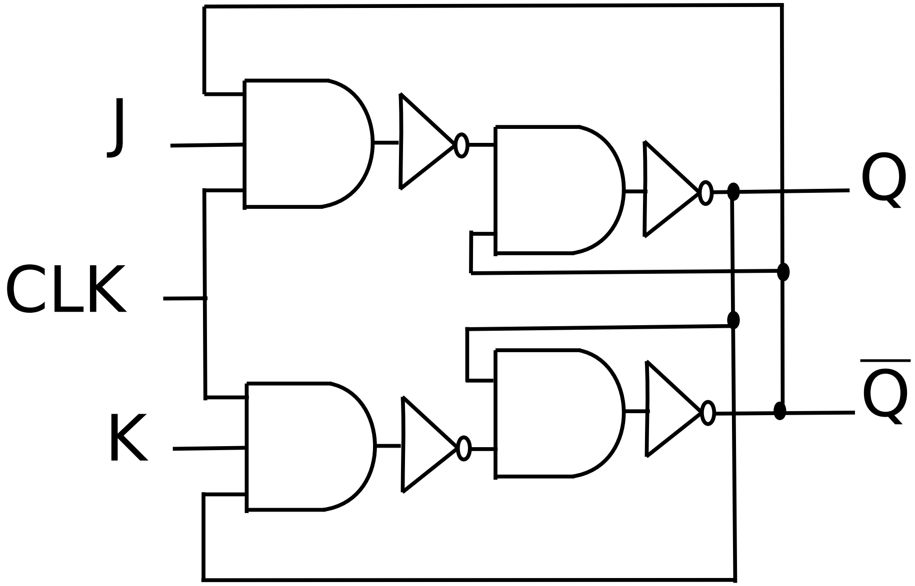
•However the problem with a latch is that the only way to turn it off is by breaking the feedback loop.



•We found that JK flip-flops are a more flexible latch which don't have this problem.

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

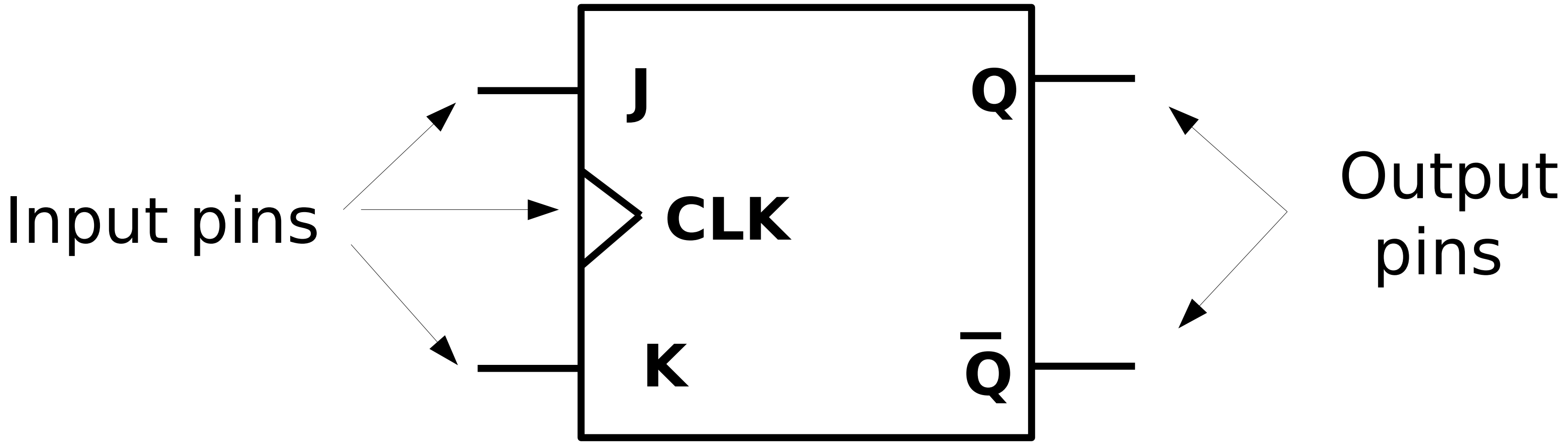
•This is a real memory circuit that you would find in a computer. It can store a **1** or a **0**.



Dejdżer / Digga - Flickr

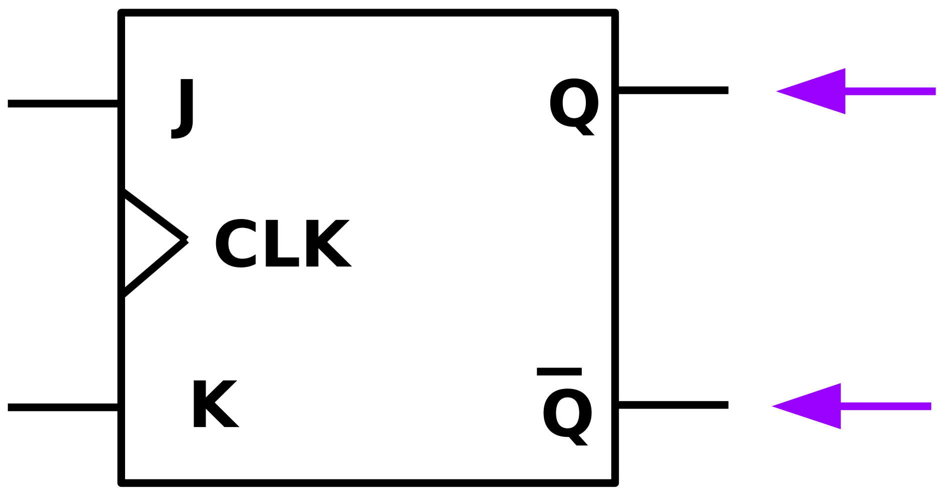
•It looks a bit complicated.

Recap: The JK flipflop

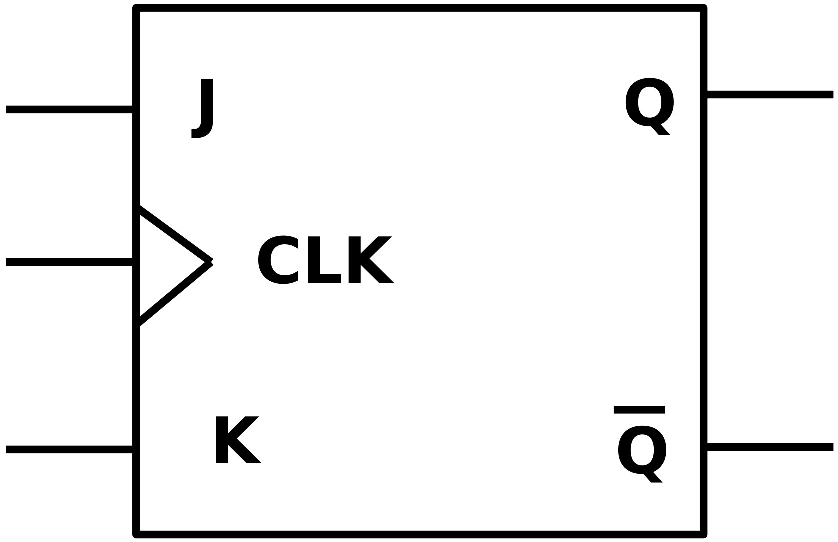


Recap: The JK flipflop

The value the flip flop is currently remembering appears on pin Q and the inverse appears on pin $\overline{Q}$.



Recap: The JK flip flop



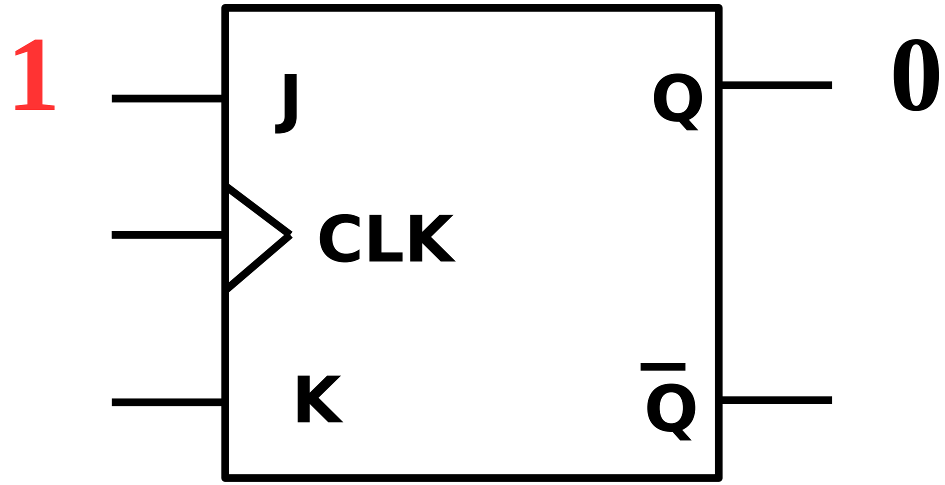
•This is the most basic electronic memory element used to store 1's or 0's.

•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...

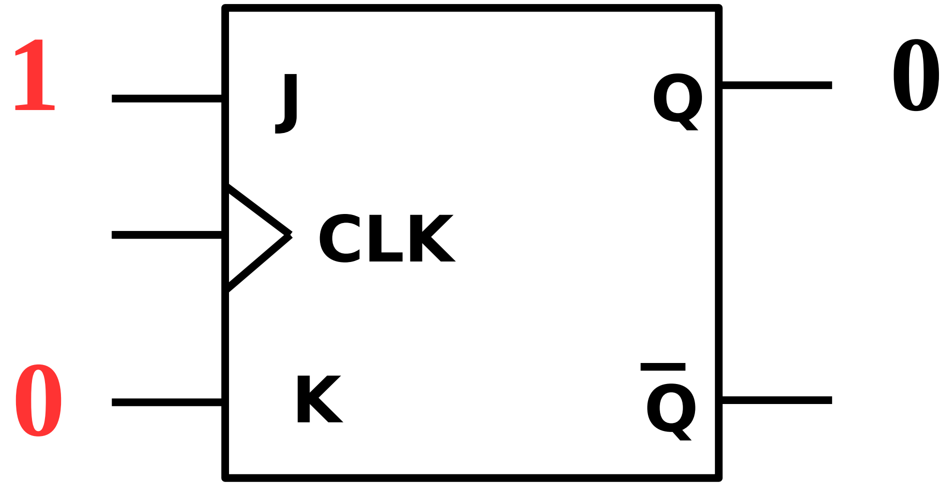
Recap: Storing a 1

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



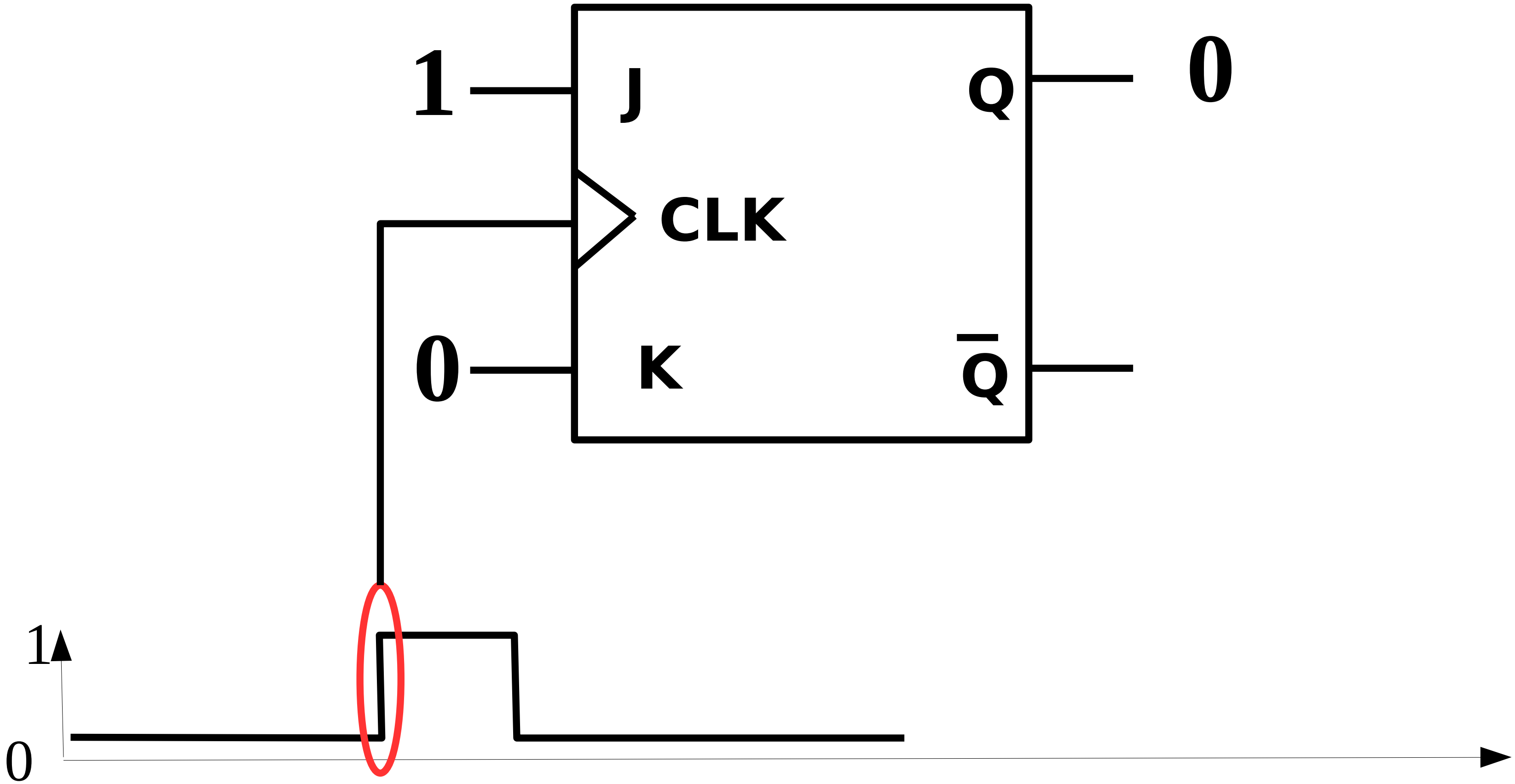
Recap: Storing a 1

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

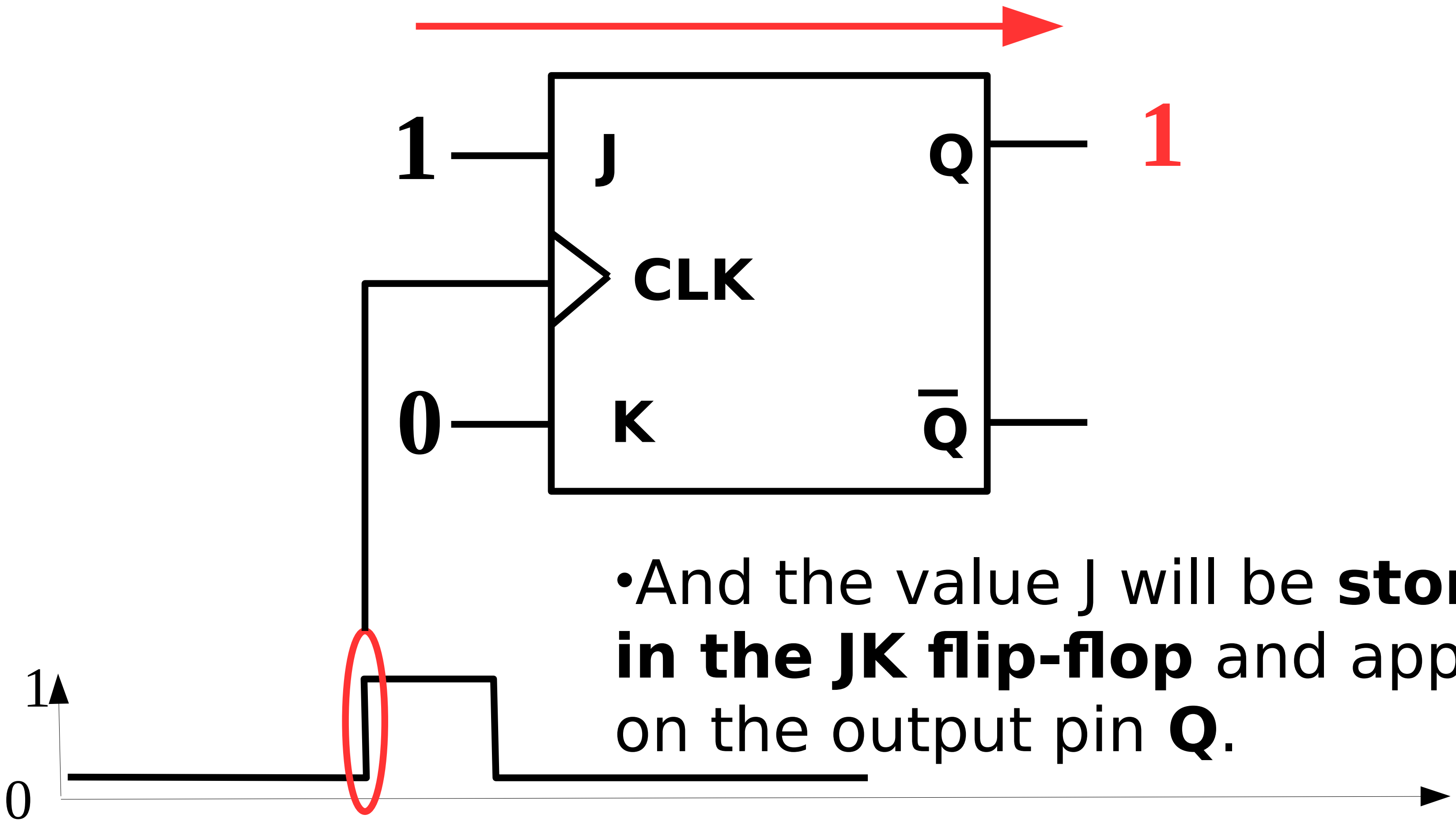


Recap: Storing a 1

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



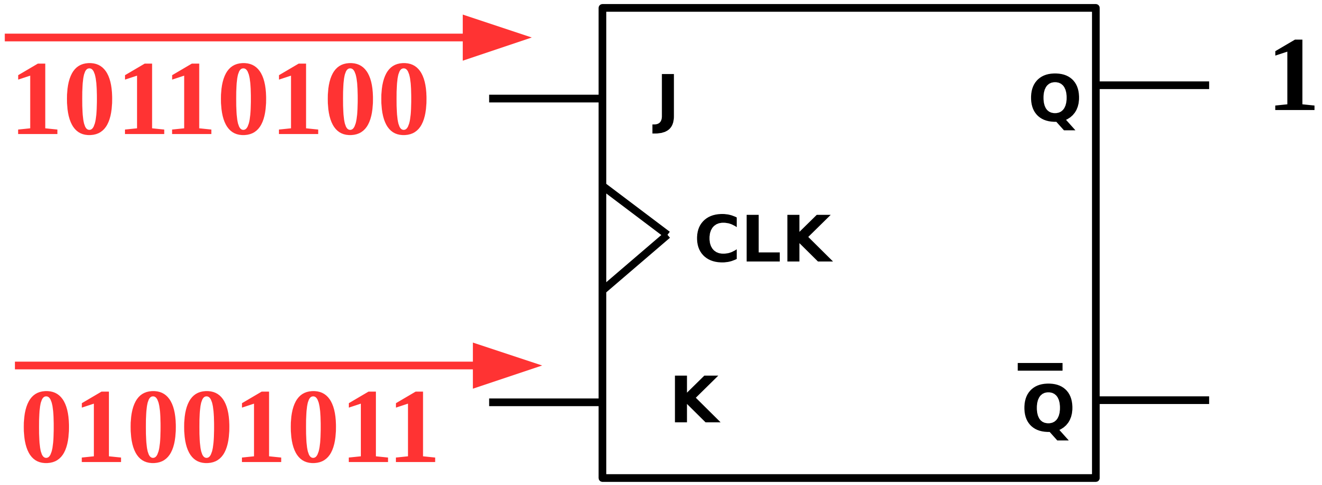
Recap: Storing a 1



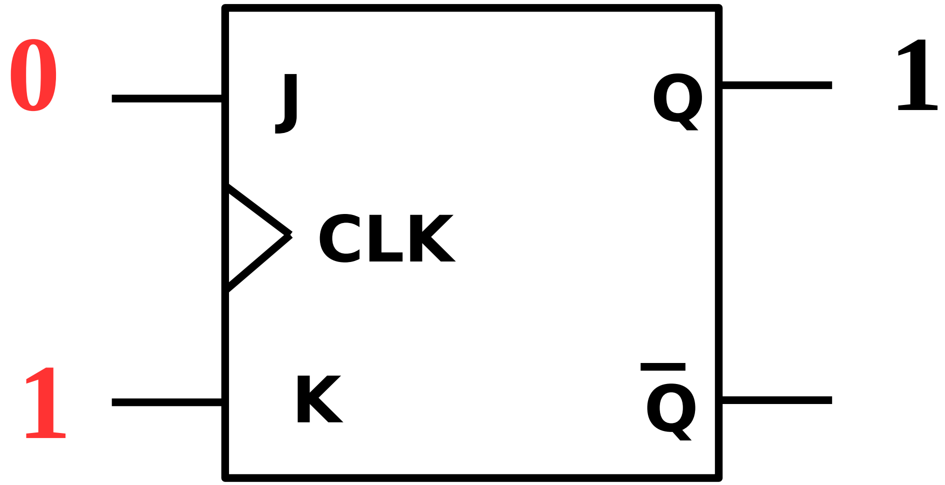
•And the value J will be **stored in the JK flip-flop** and appear on the output pin **Q**.

Recap: We need a clock pulse to remember things.

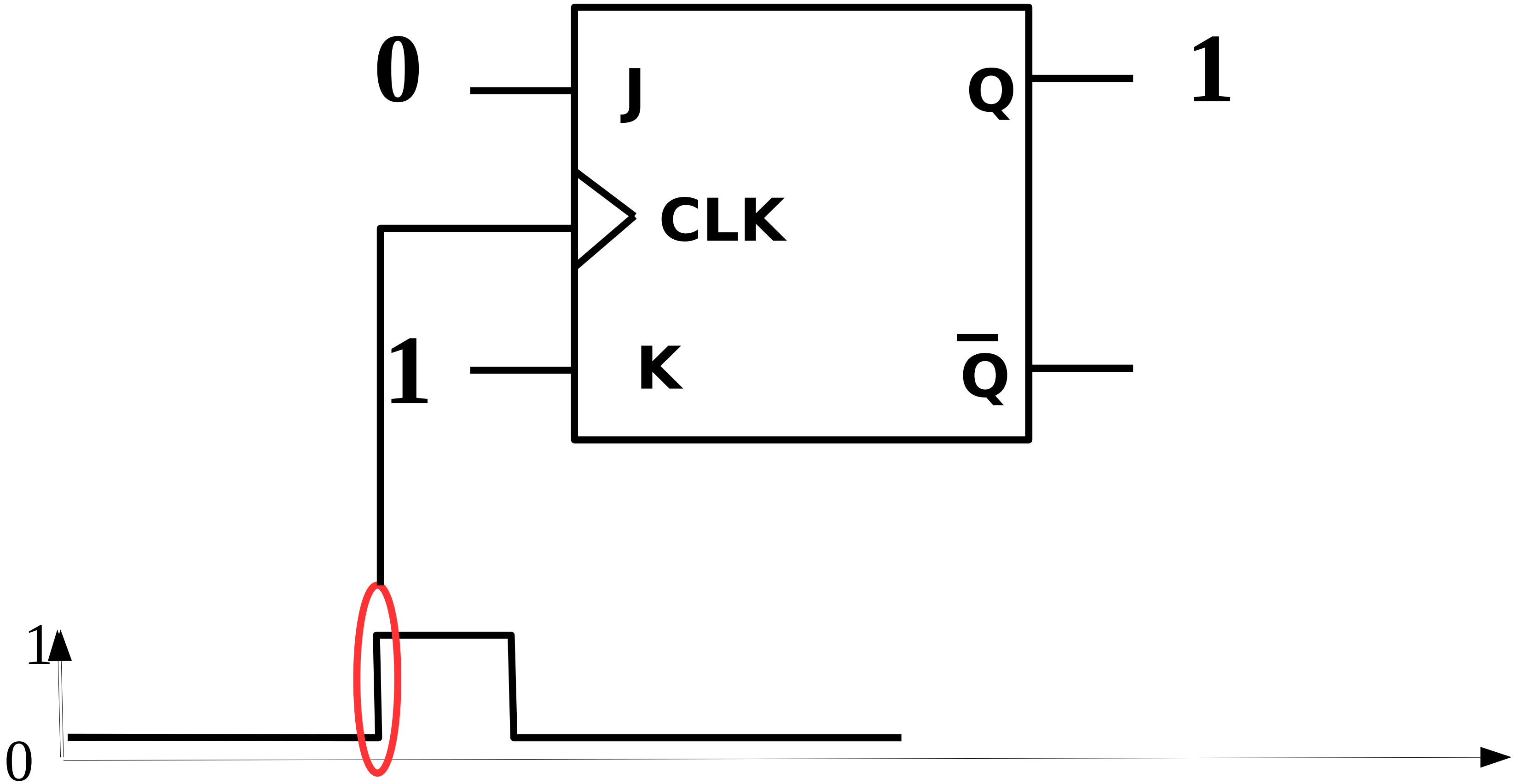
•Then no matter what you do the **J** or K pins the **output will remain unchanged**



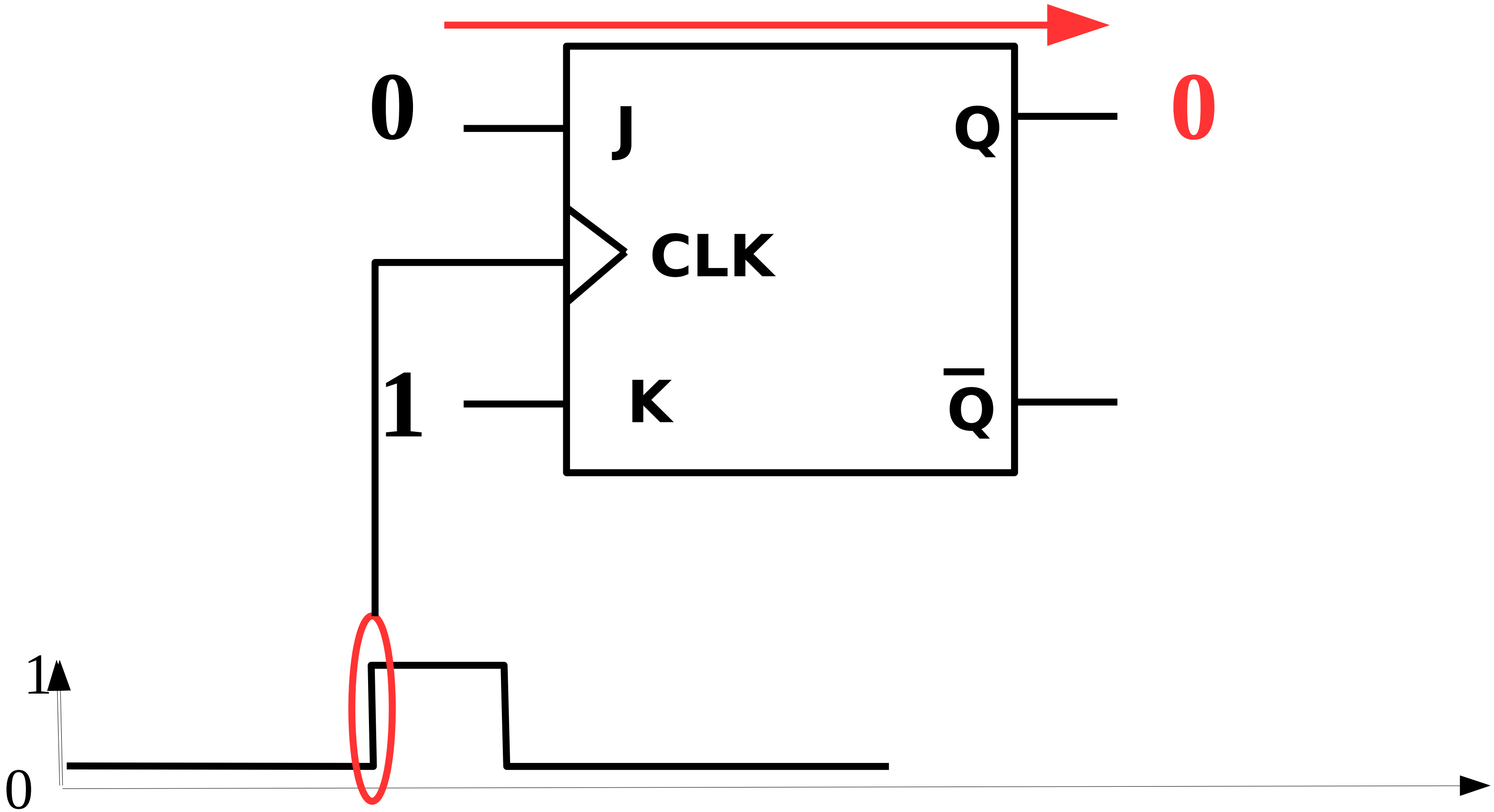
Recap: Storing a zero



Recap: Storing a zero

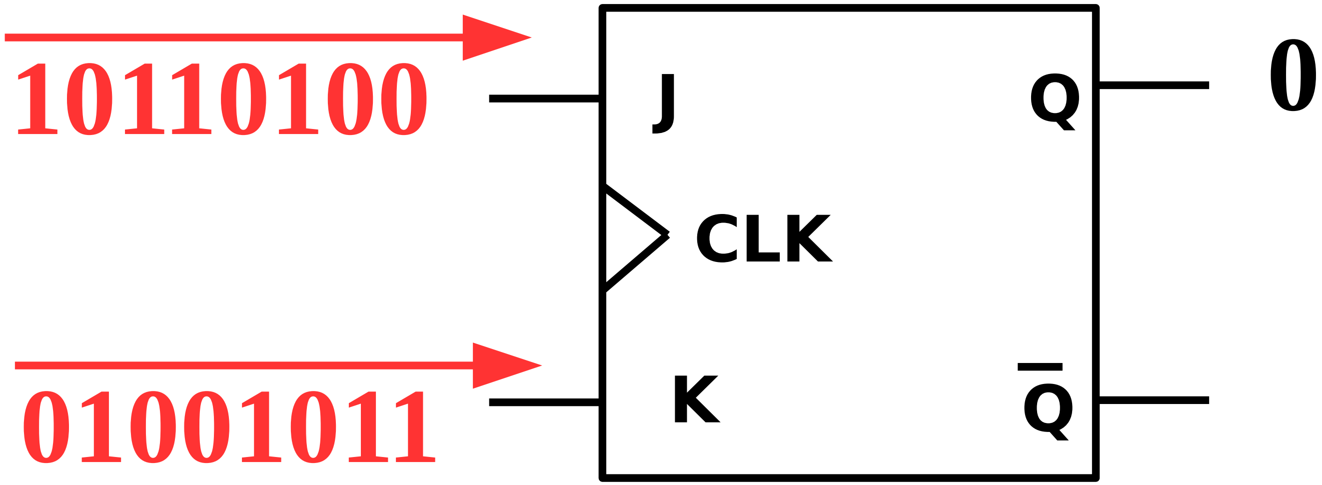


Recap: Storing a zero

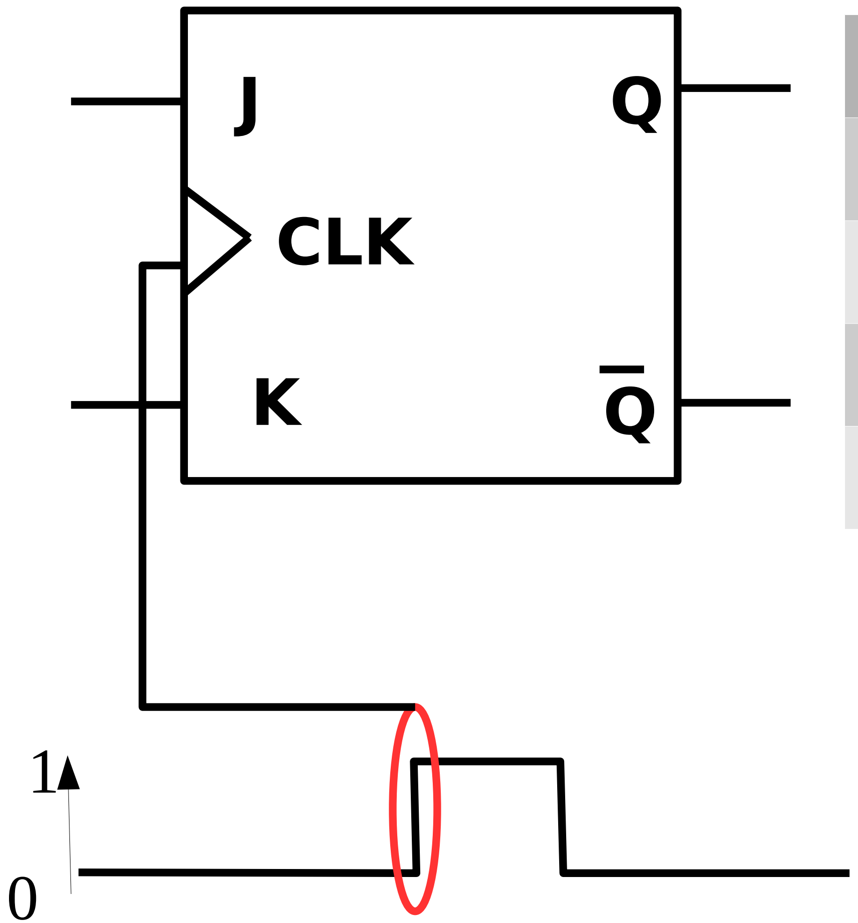


Recap: Nothing happens without a clock pulse.

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



Recap: The JK flipflop truth table



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

Summary of today's lecture

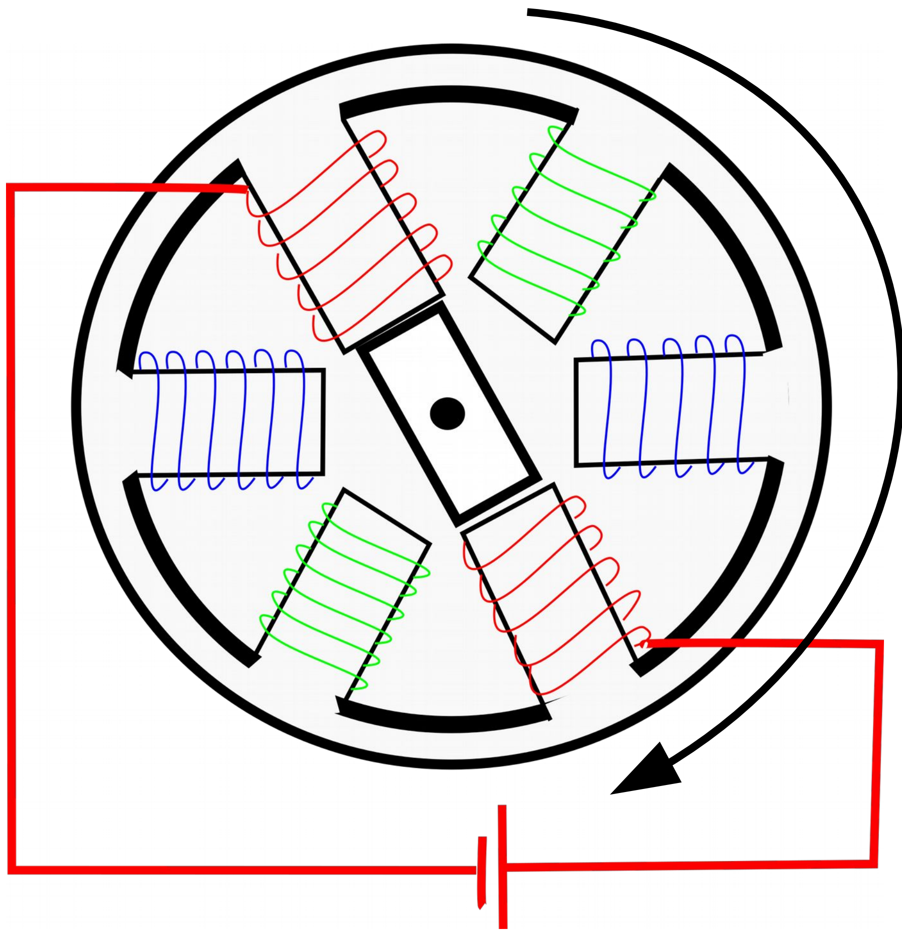
- Recap of last lecture

- **Using JK flip flops to run motors**

- Digital design
    - What is digital design?
    - How to do it.
    - Where you might meat digital design
    - FPGAs

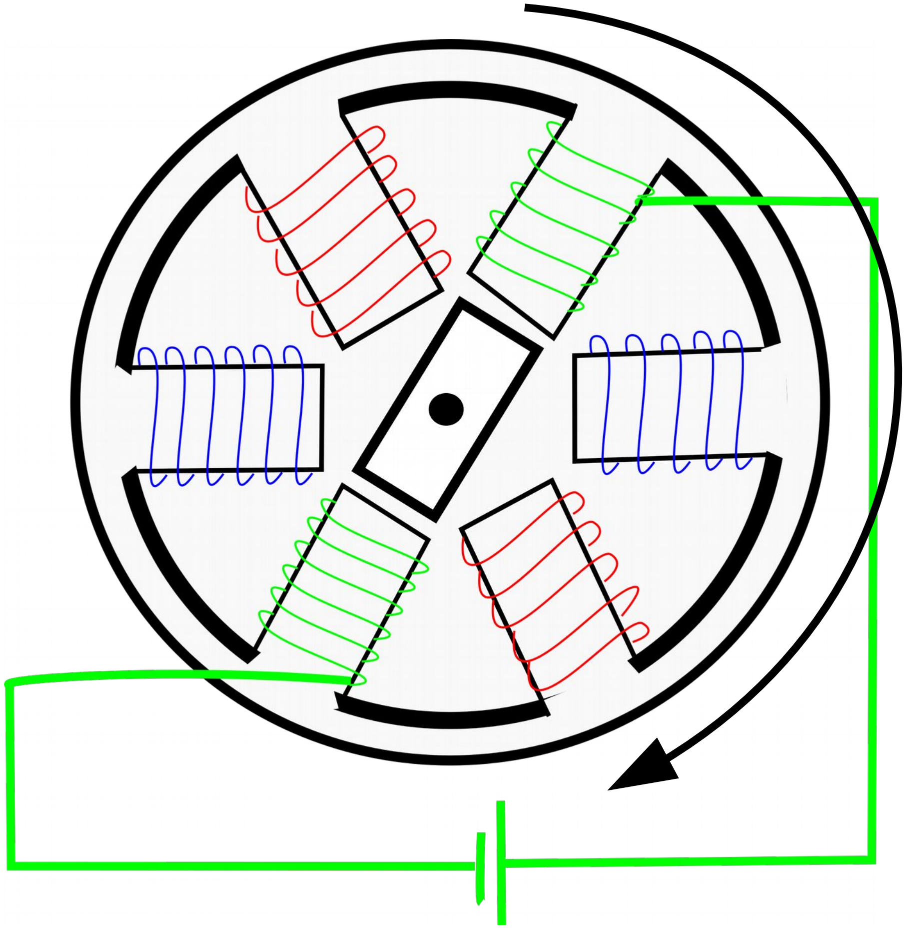
- Race times

Switch reluctance (SR) motor

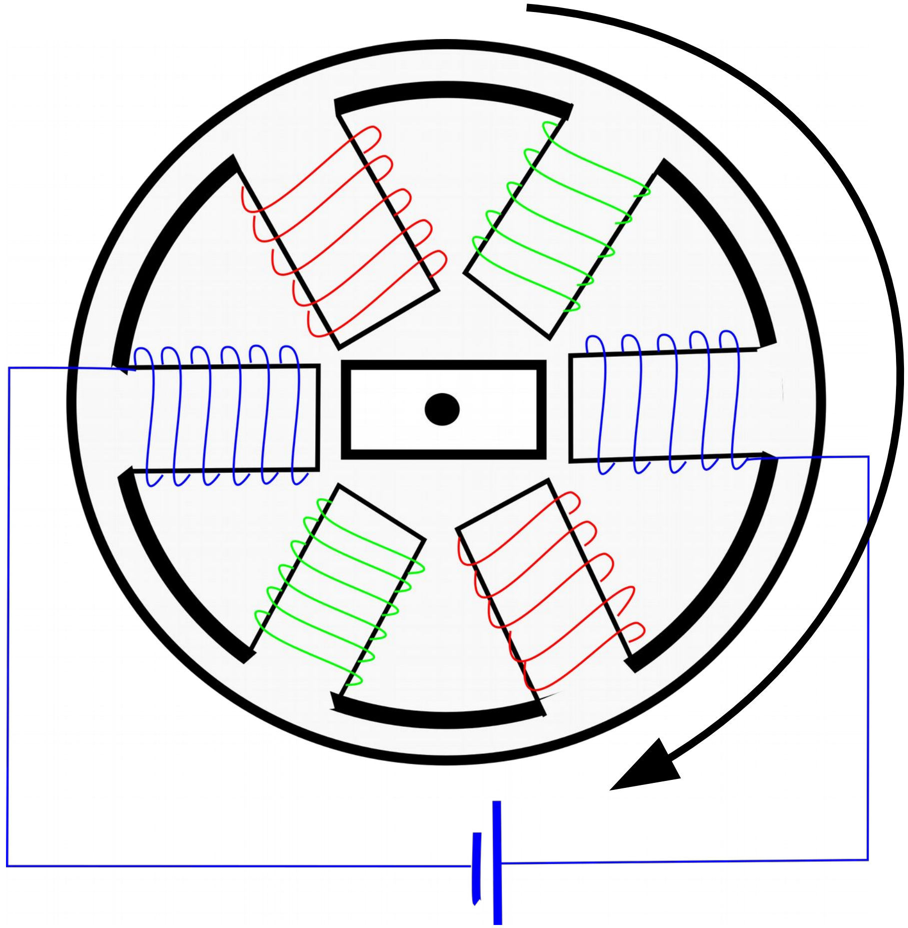


- An SR motor is a motor which has an iron bar at it's center.
- You can move the bar by turning on and off the field in the different electromagnets.
- It's like the stepper motor Arthur has been teaching you about but a bit more simple

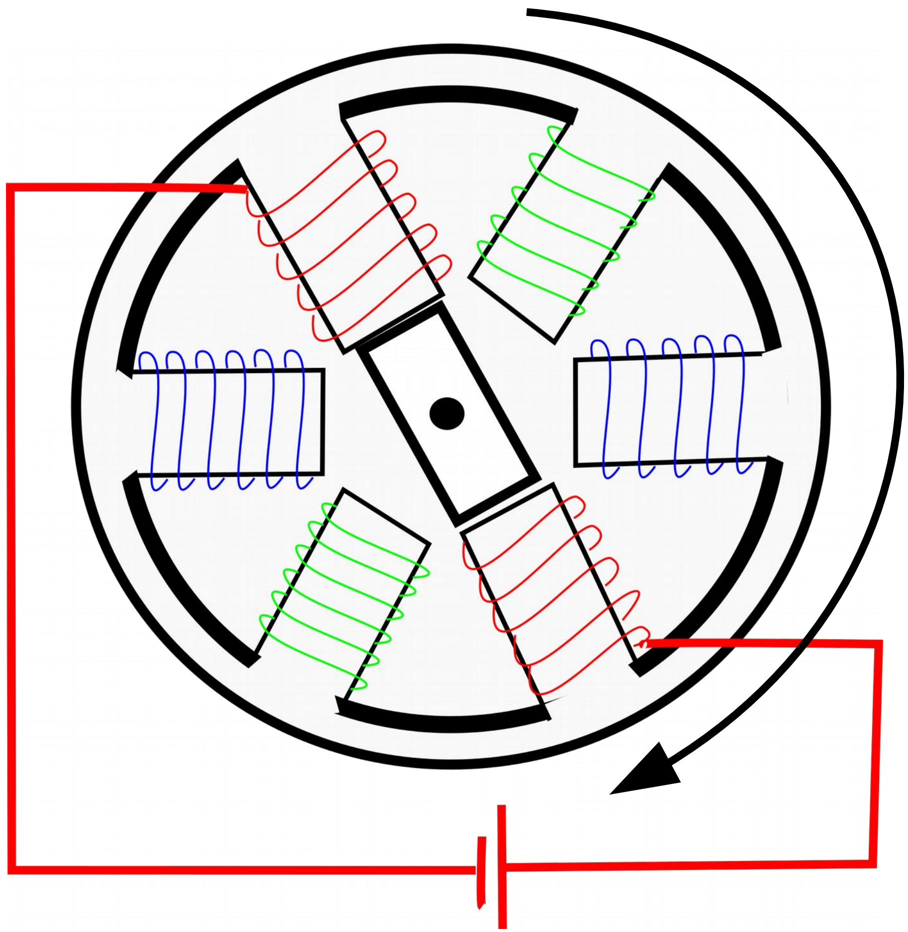
Green electromagnet on



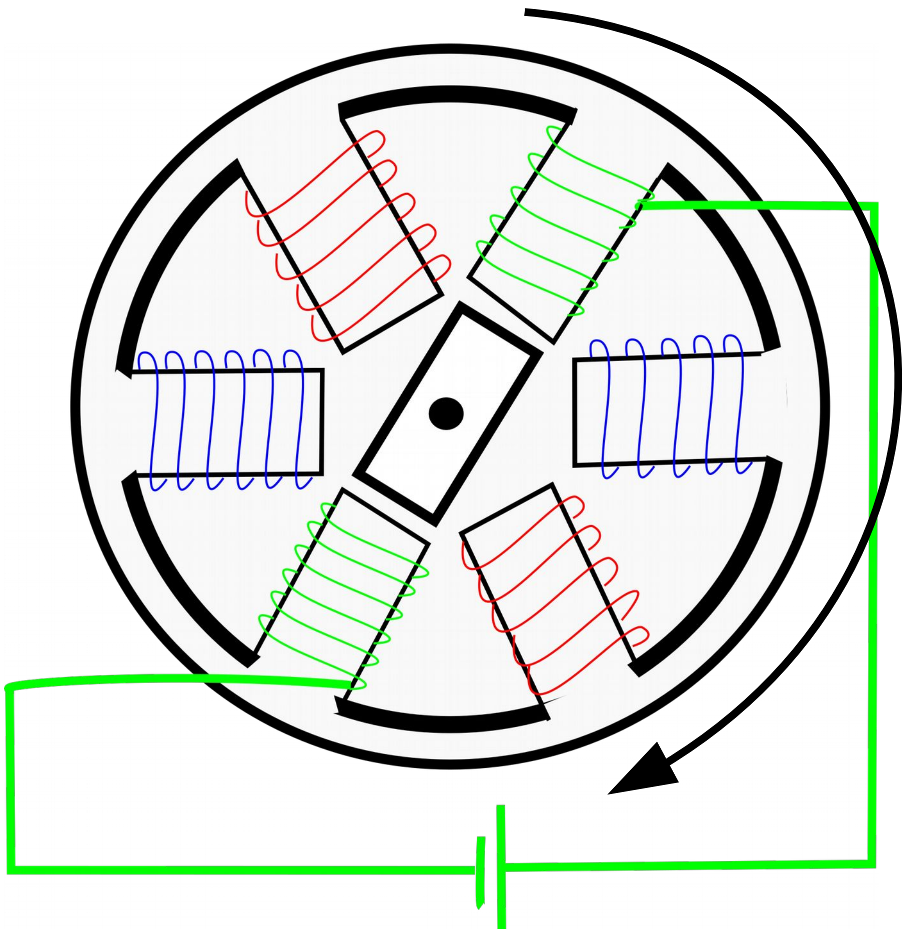
Blue electromagnet on



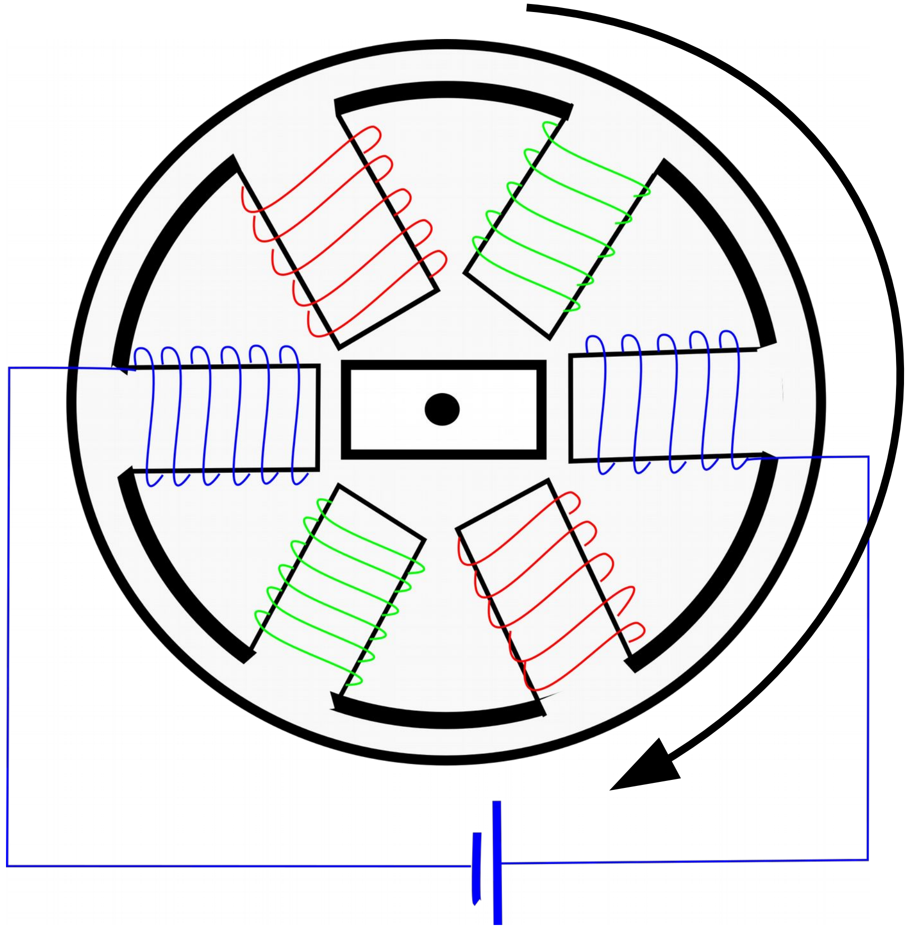
Red electromagnet on



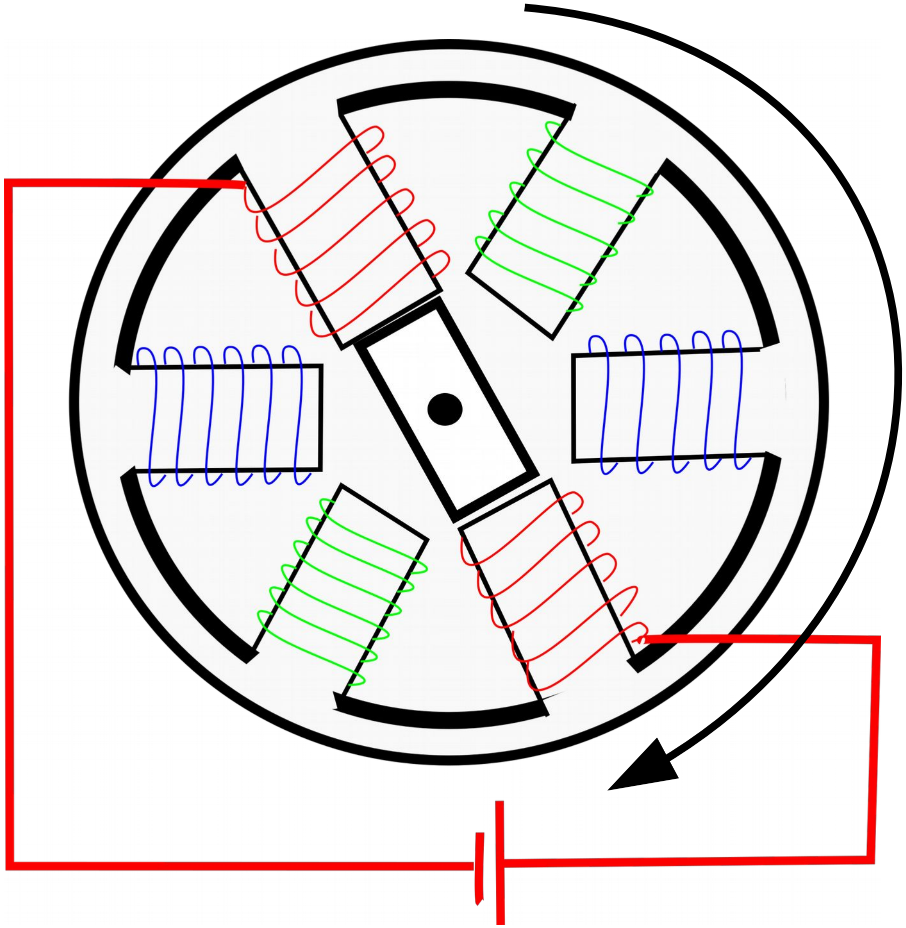
Green electromagnet on



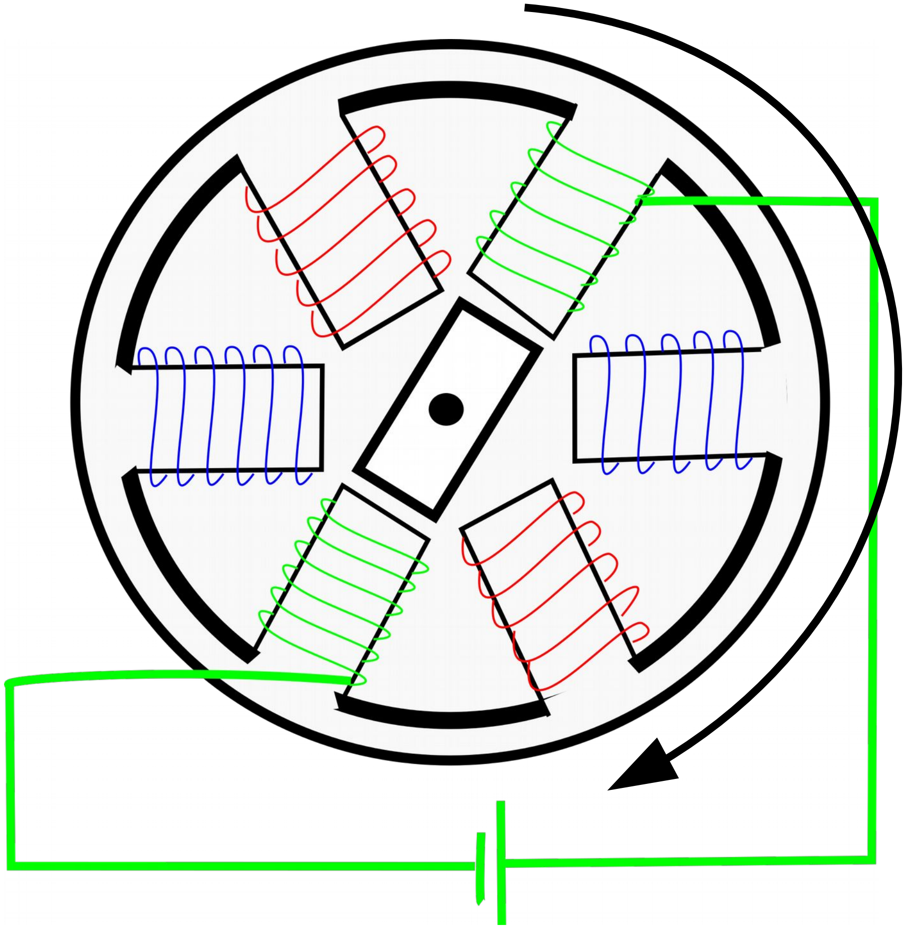
Blue electromagnet on



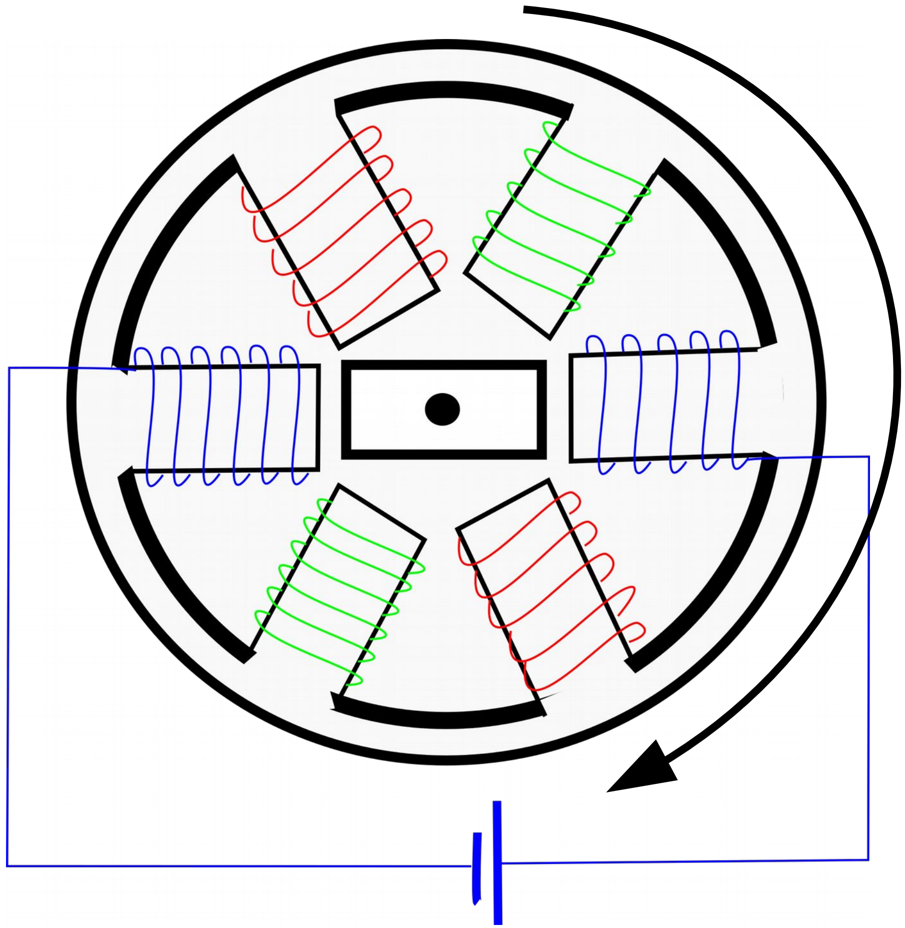
Red electromagnet on



Green electromagnet on



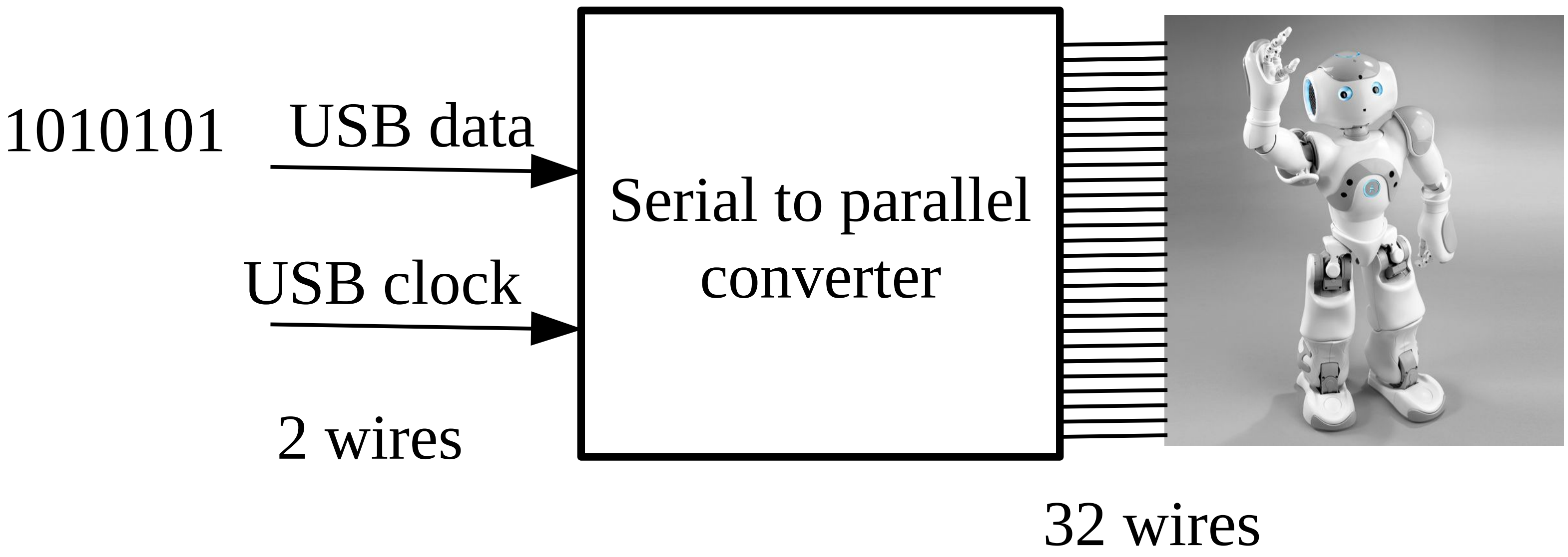
Blue electromagnet on



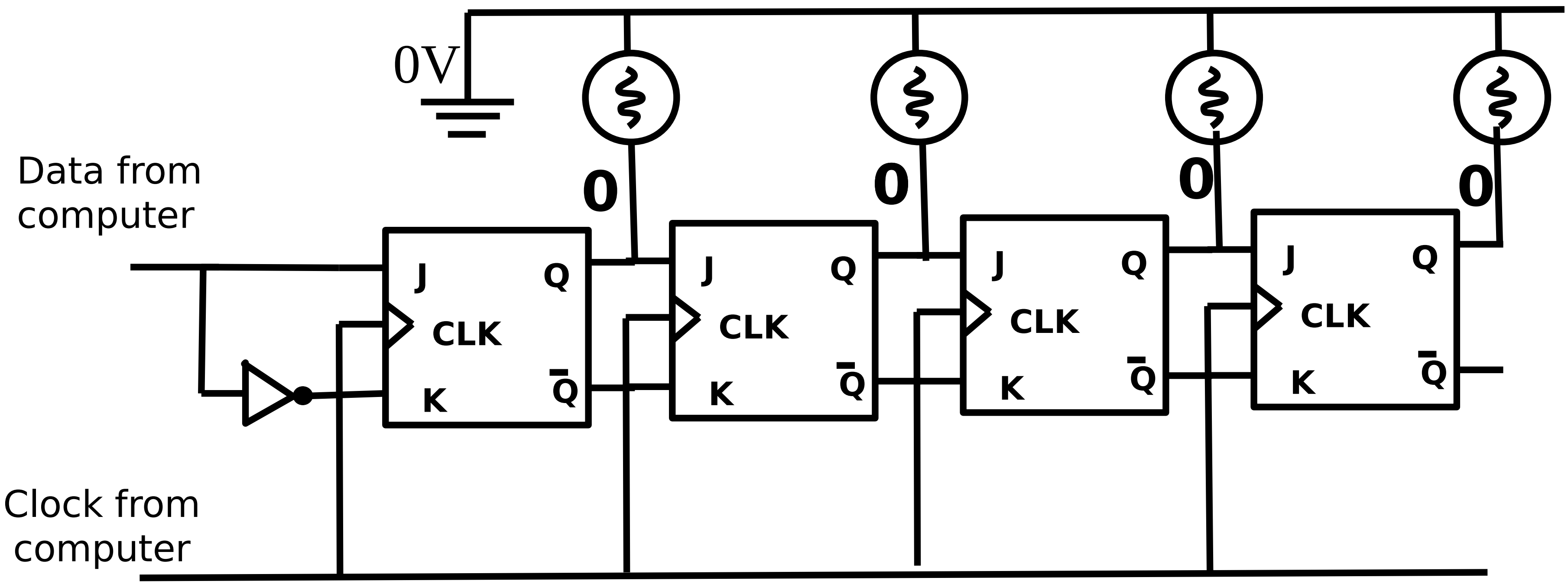
- By switching on and off the magnetic field you can move the motor.

- Good, but what has this got to do with digital electronics?

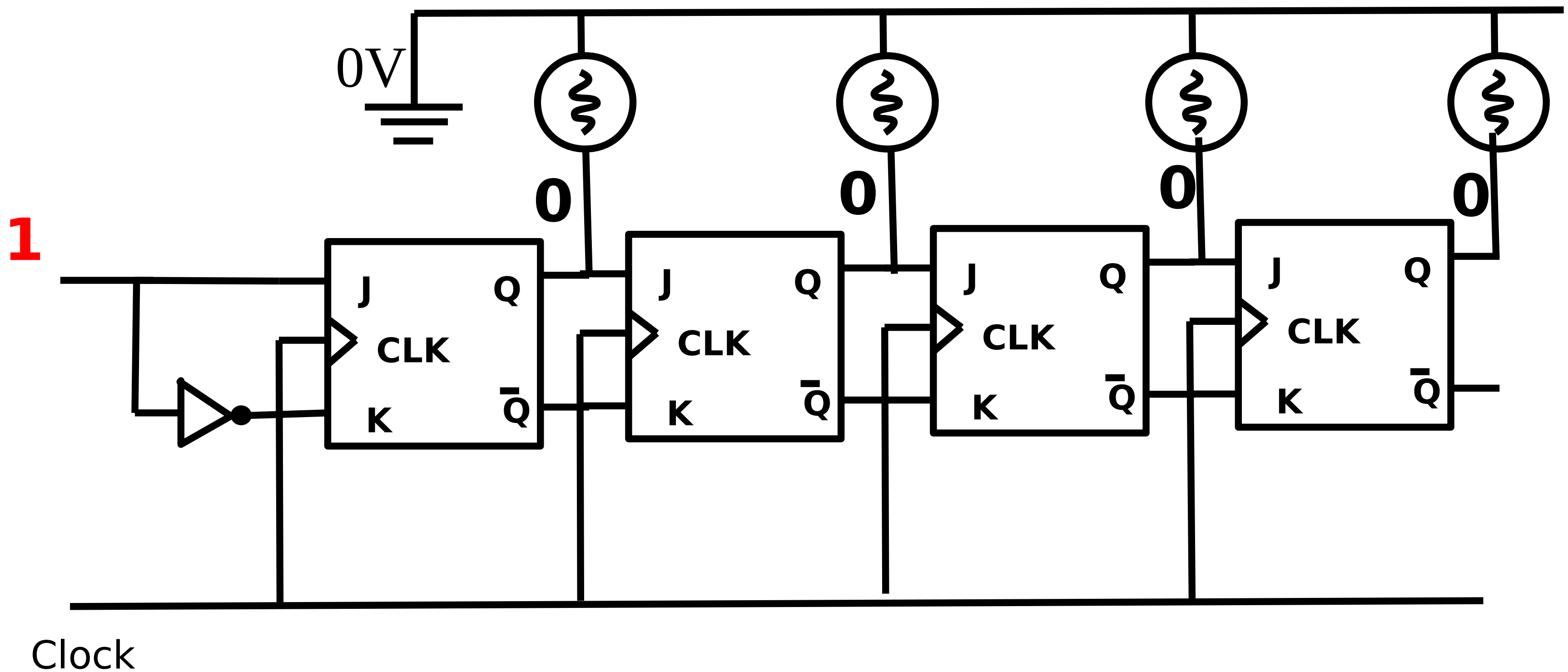
Serial to parallel converter



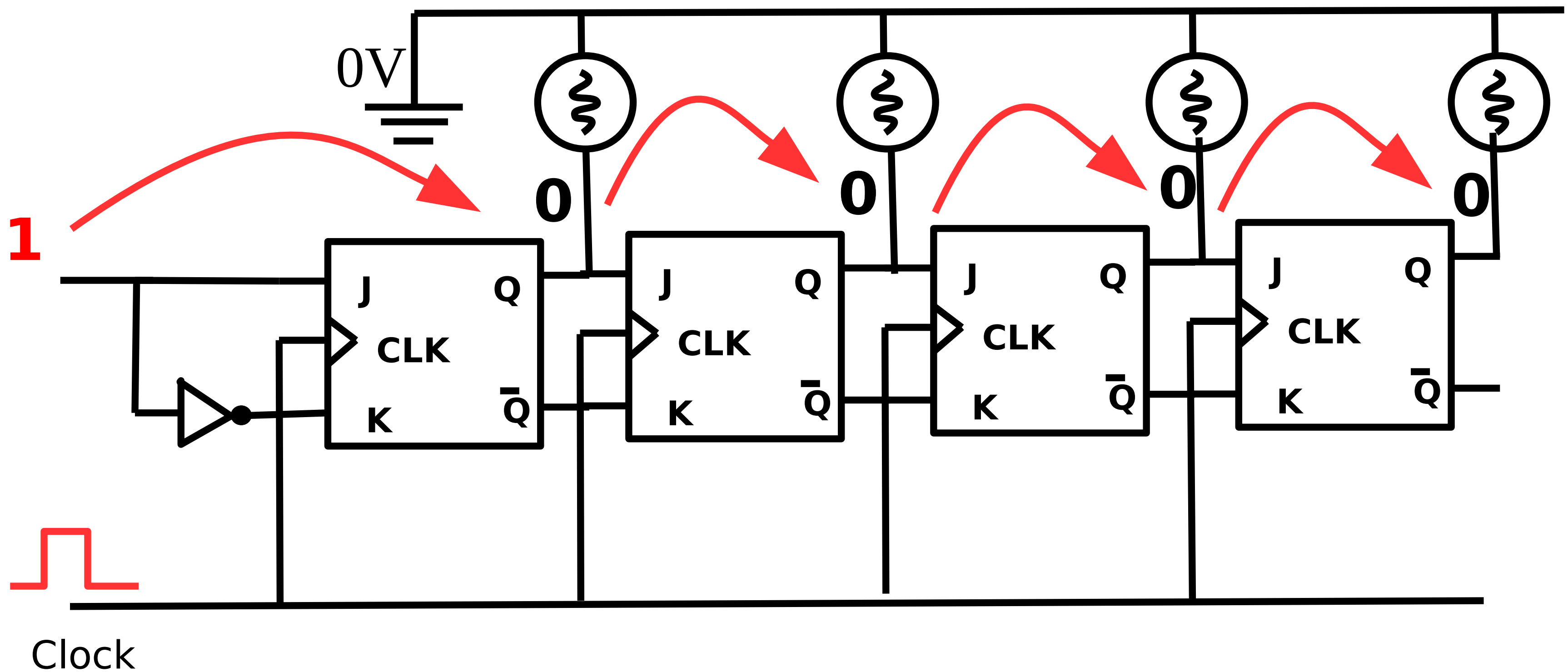
Serial to parallel converter



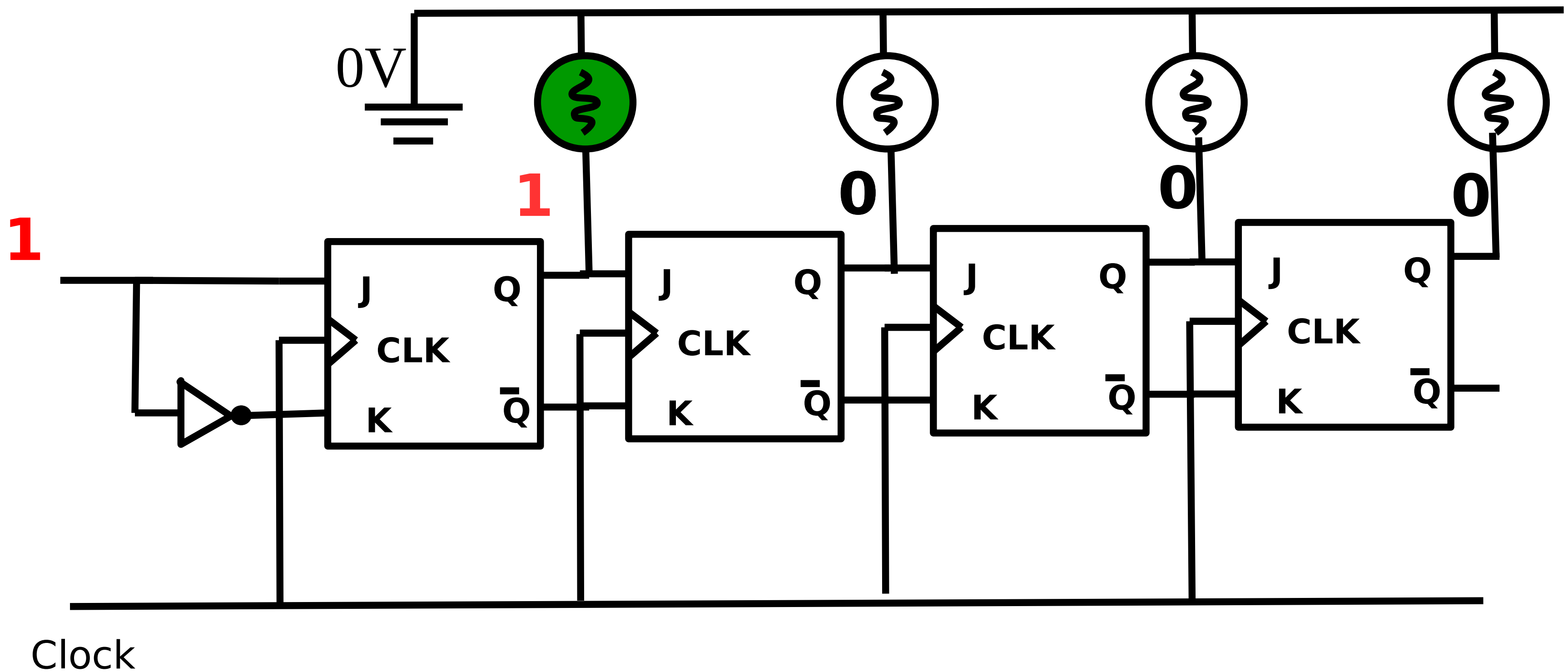
Serial to parallel converter



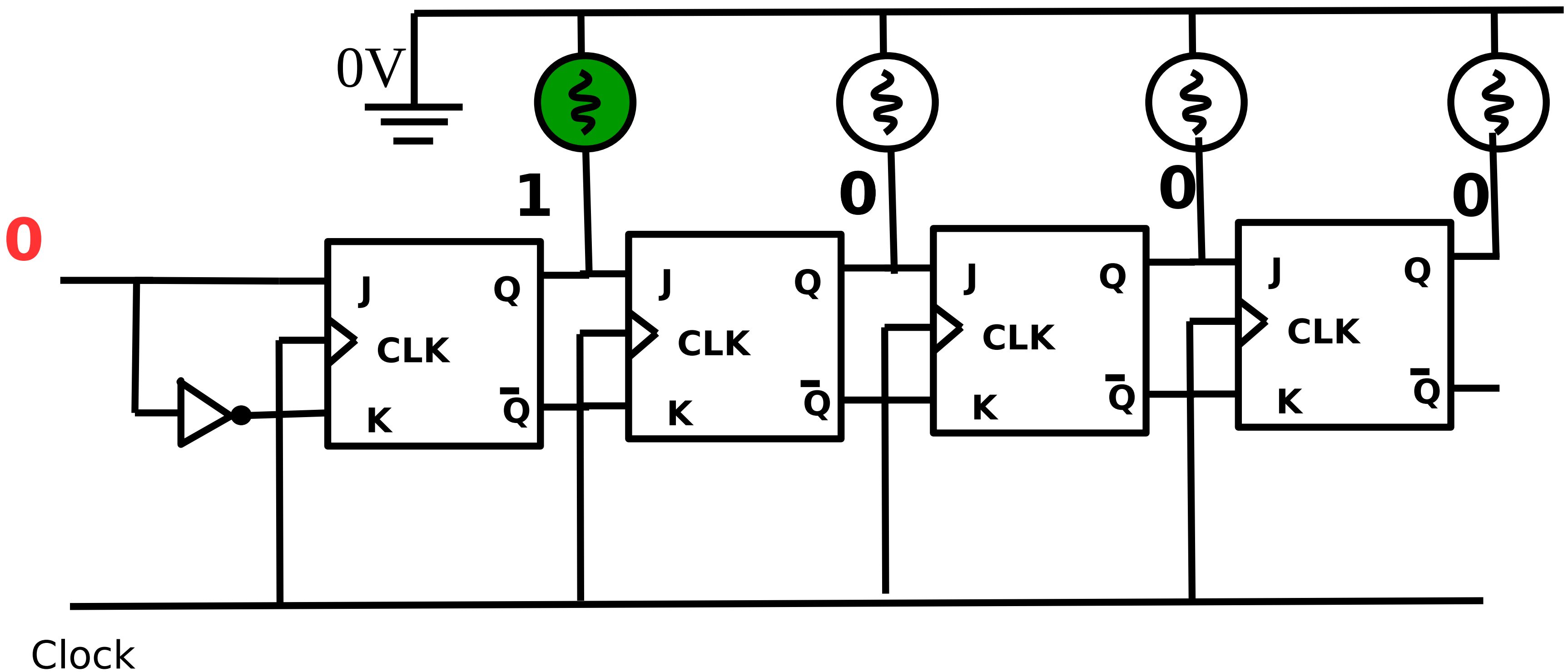
Serial to parallel converter



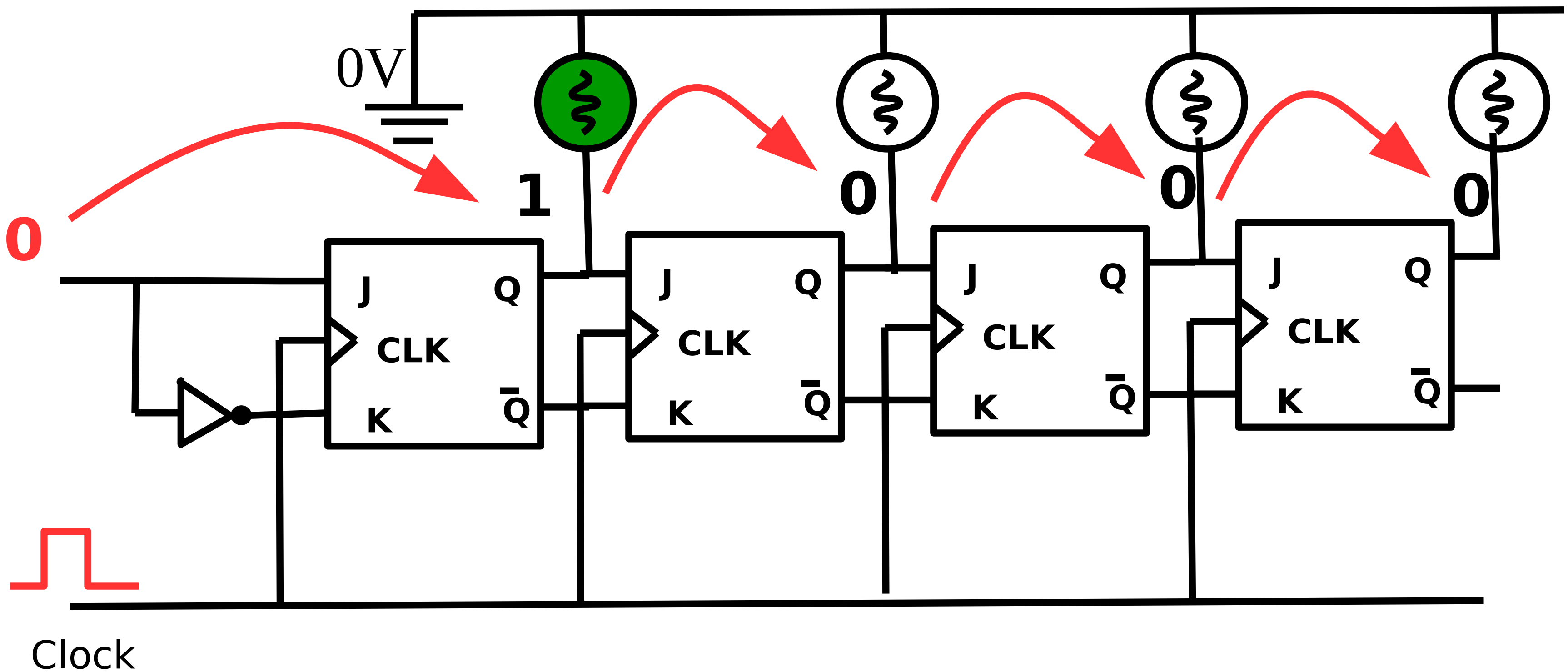
Serial to parallel converter



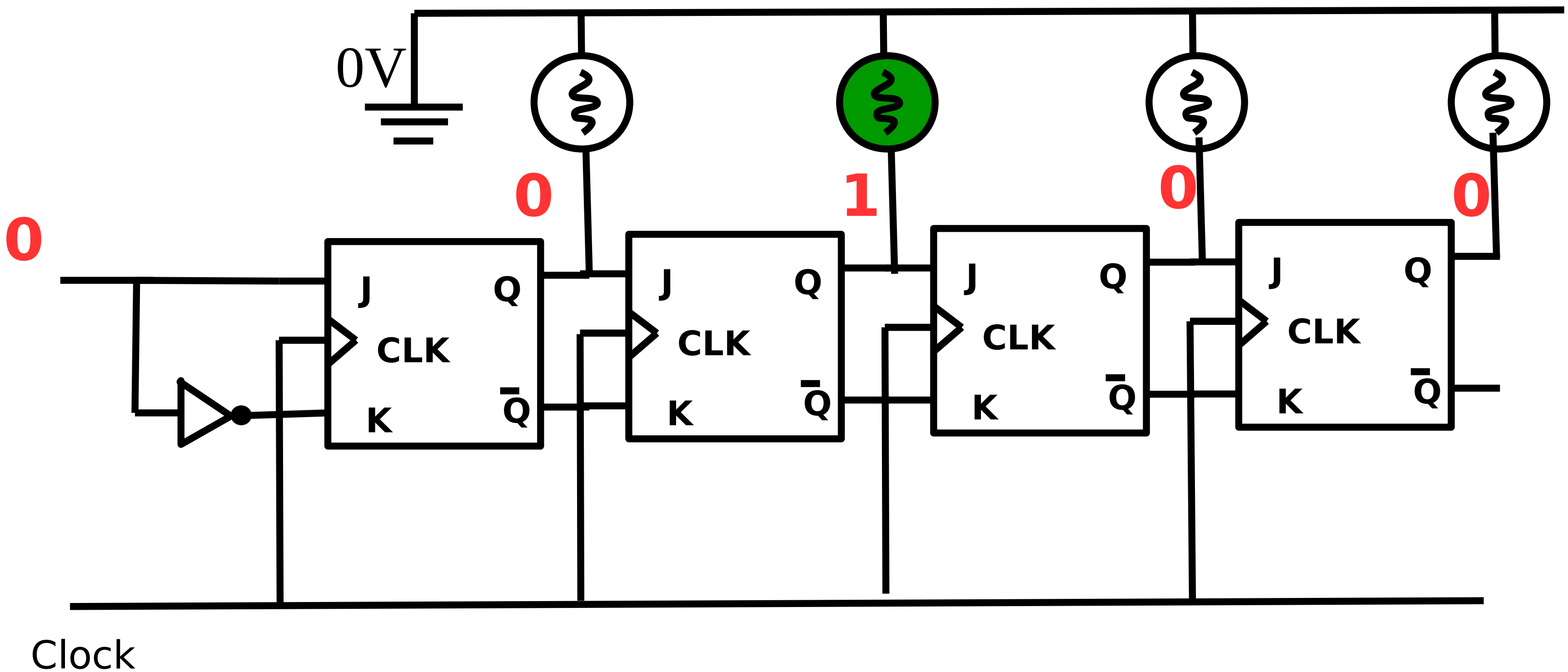
Serial to parallel converter



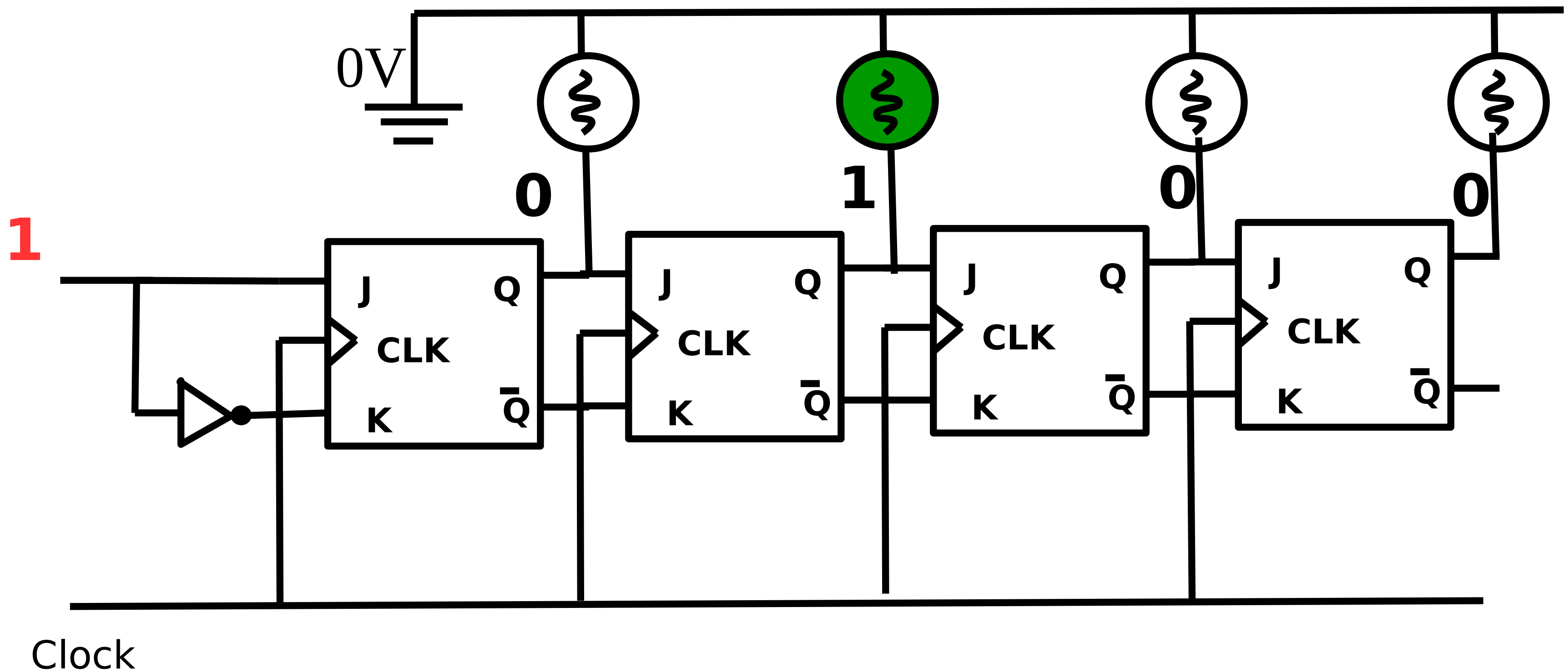
Serial to parallel converter



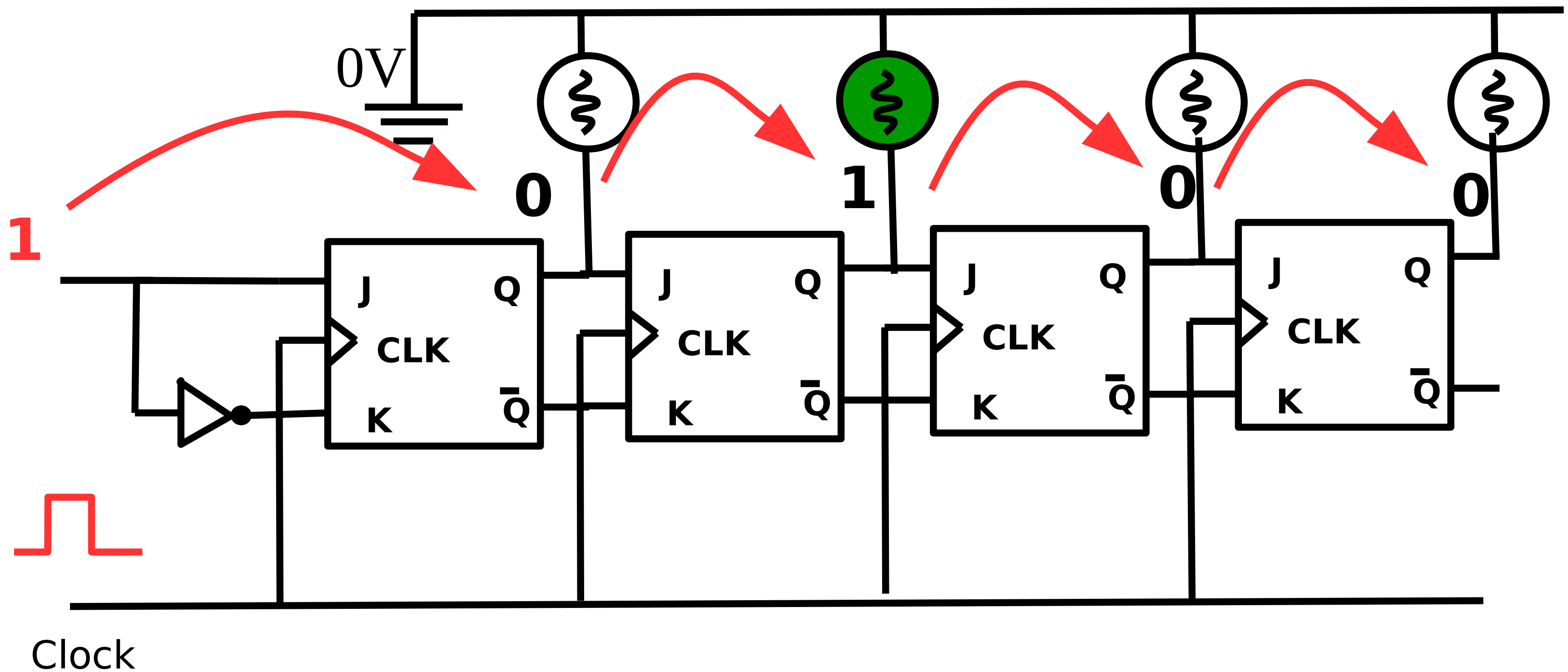
Serial to parallel converter



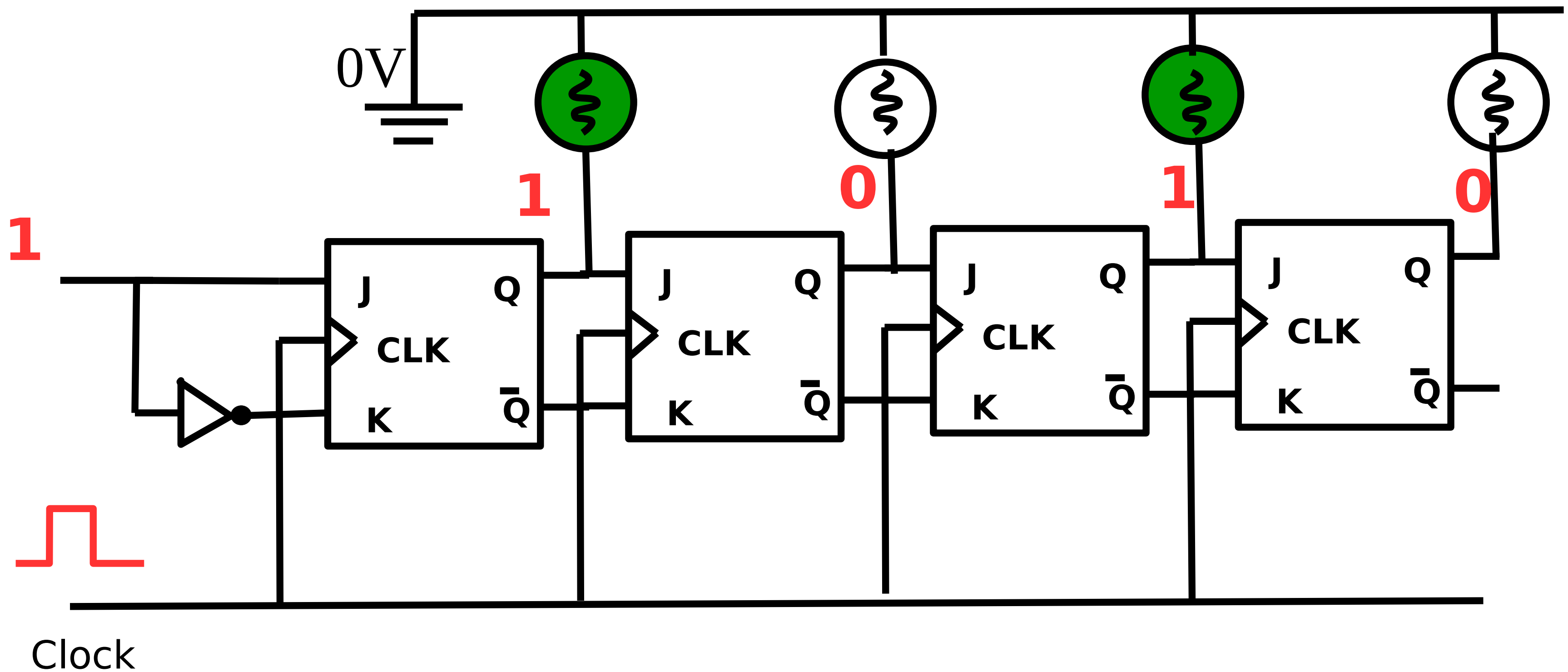
Serial to parallel converter



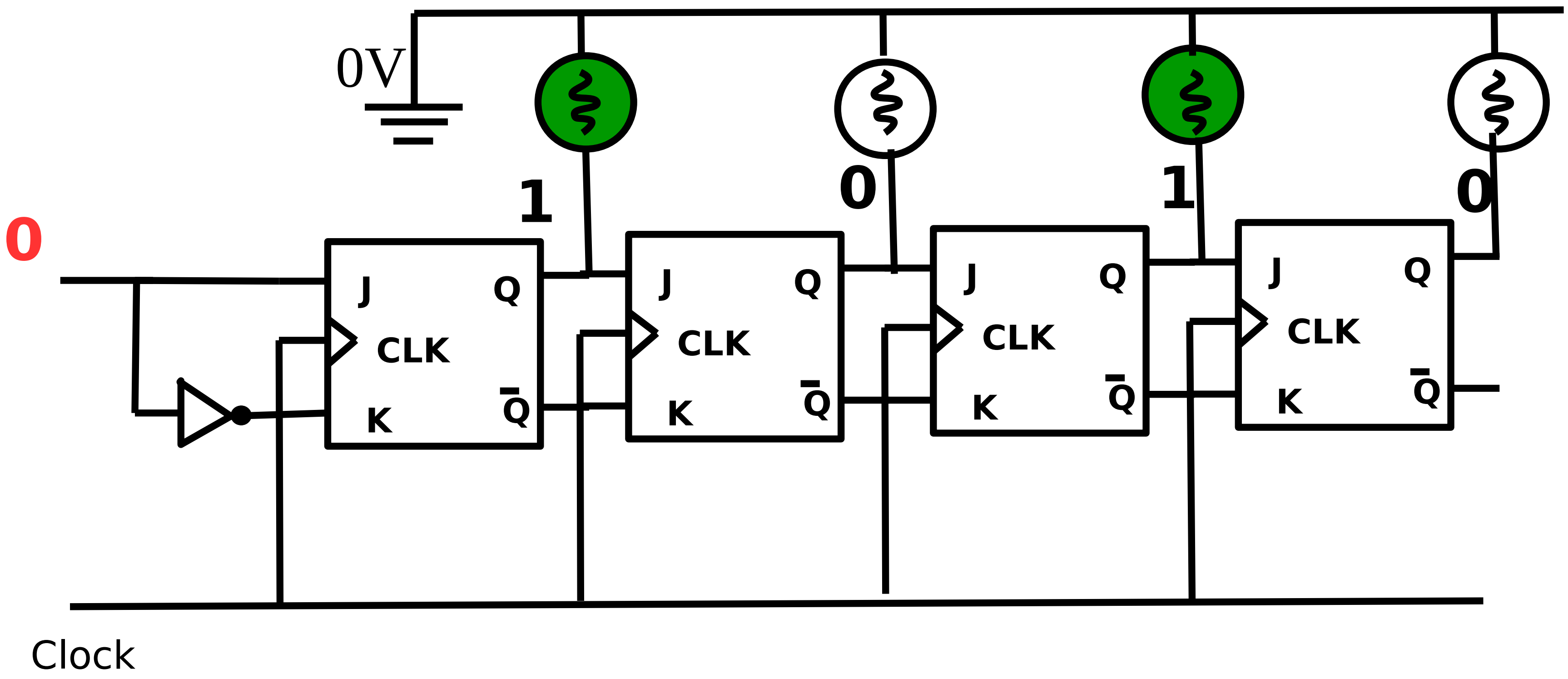
Serial to parallel converter



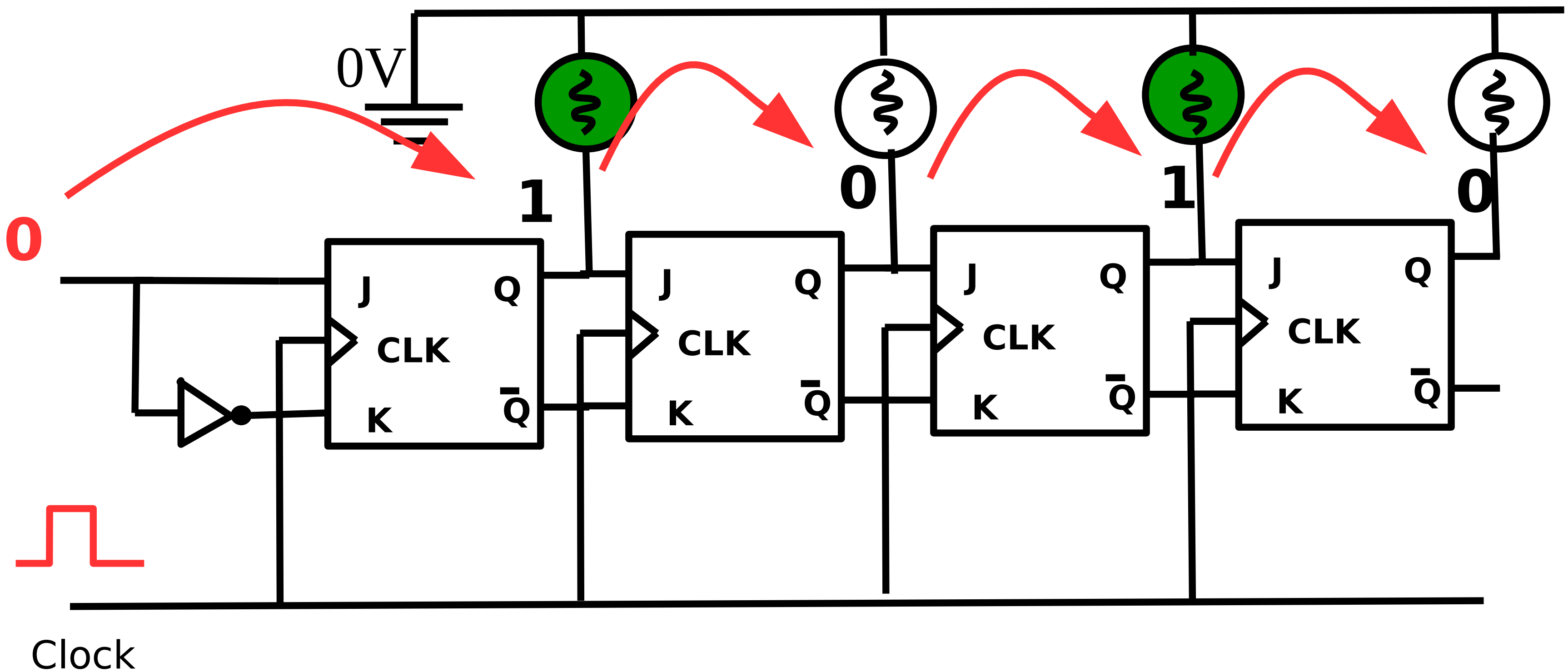
Serial to parallel converter



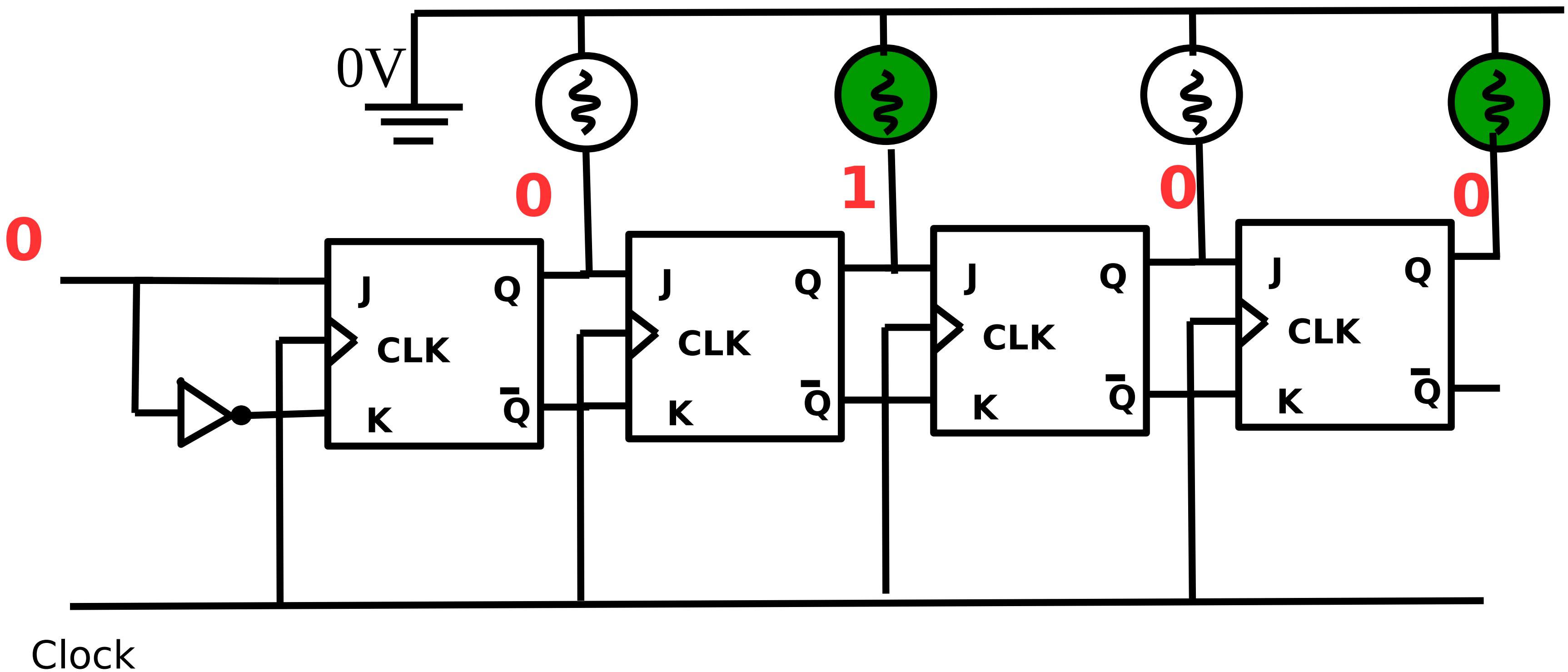
Serial to parallel converter



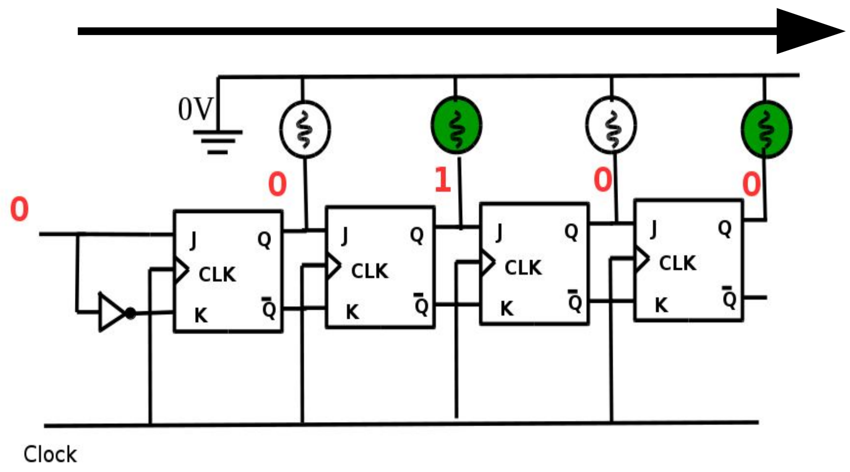
Serial to parallel converter

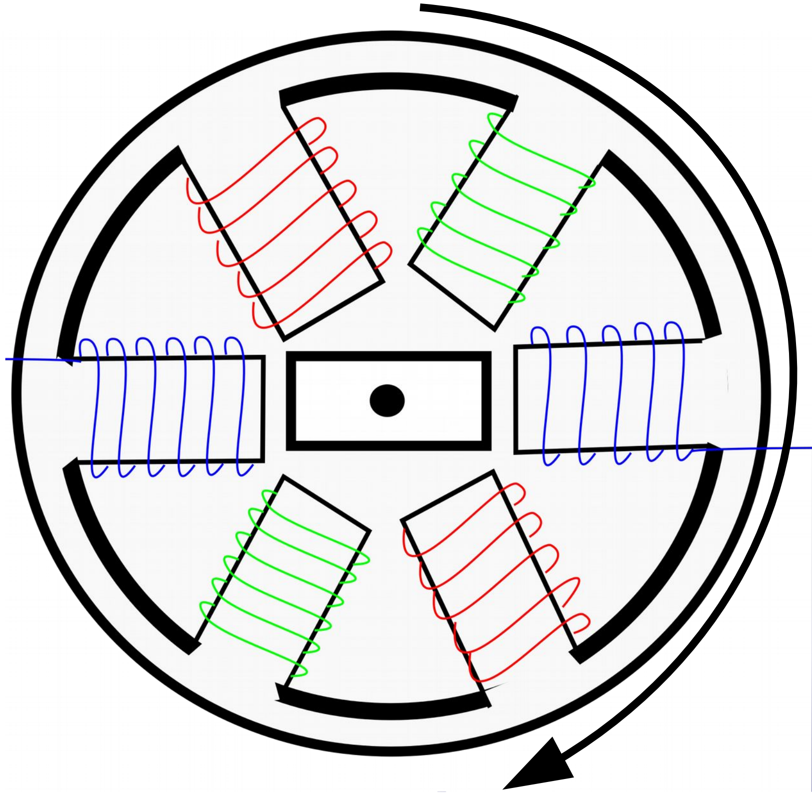


Serial to parallel converter



Are there any similarities between the motor and the serial to parallel converter?

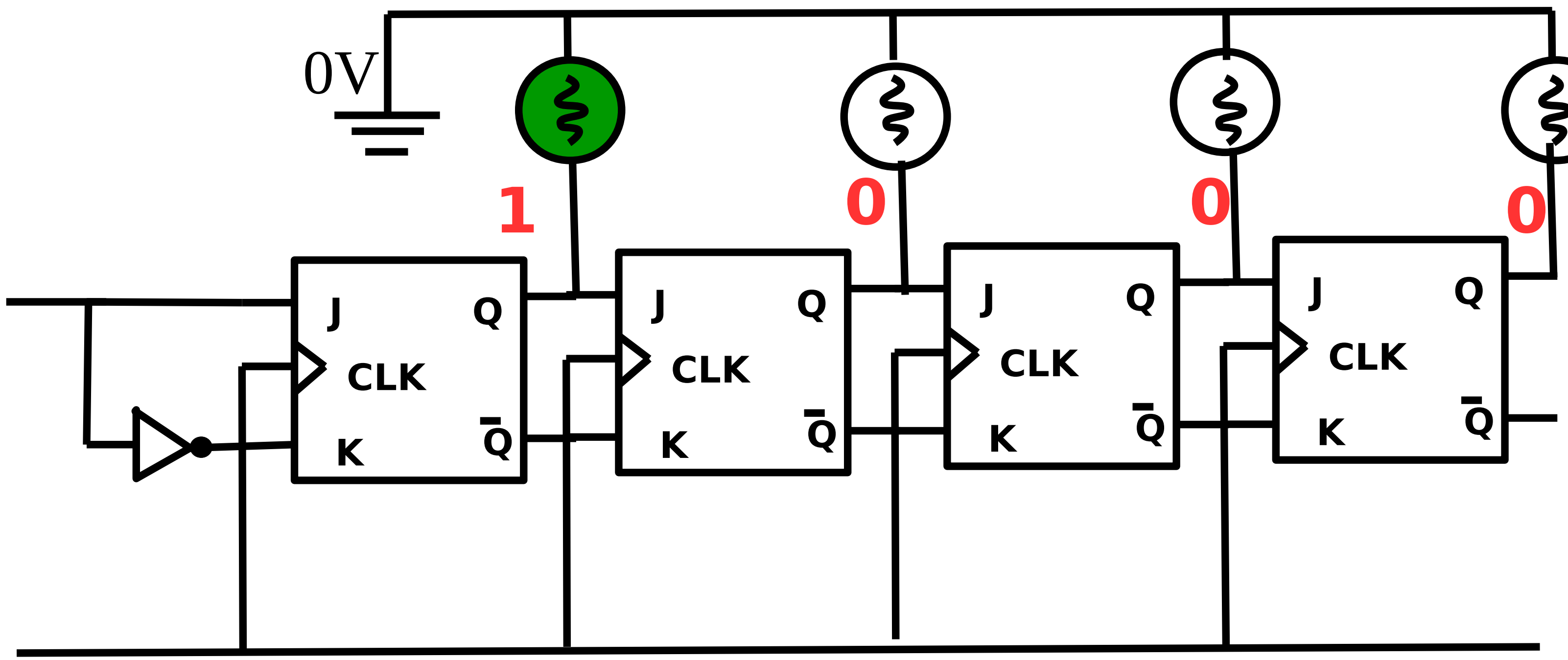




We can actually convert the serial to parallel converter circuit to a motor driver circuit.

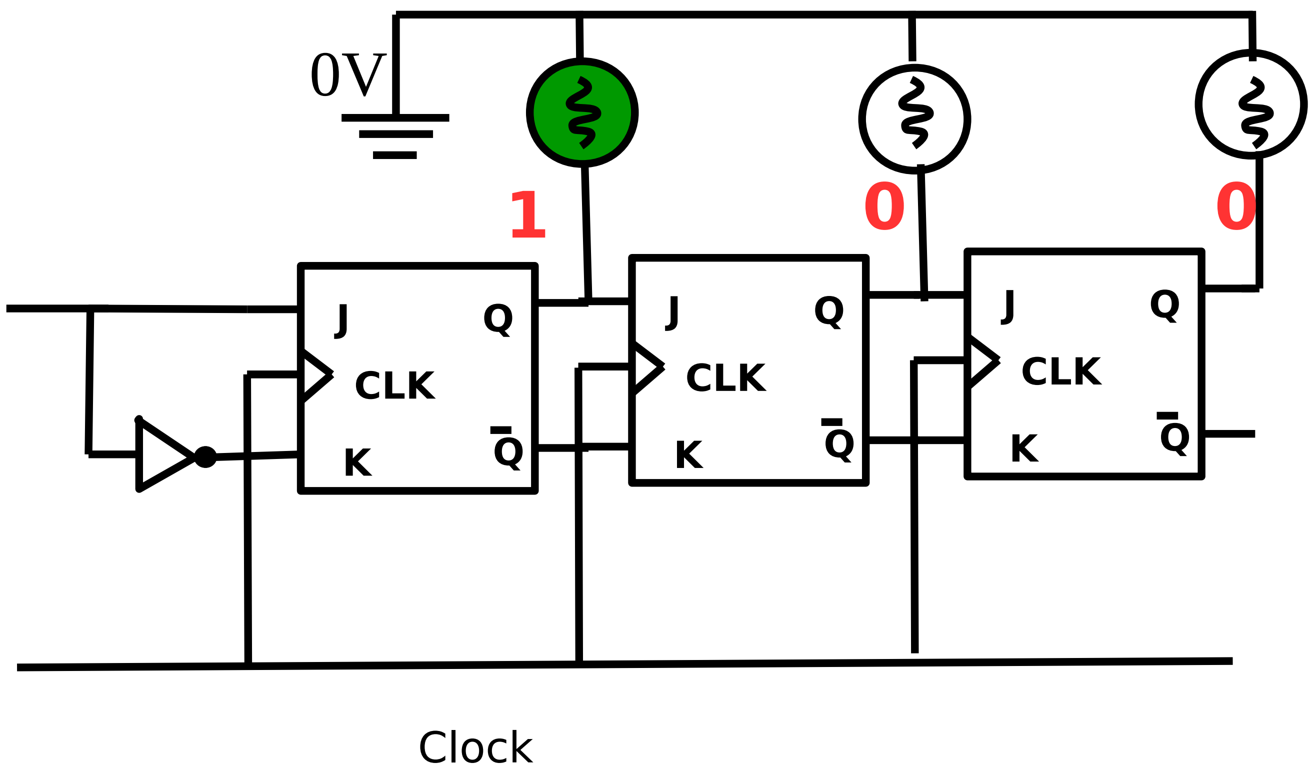
Let's modify it

- Firstly let's only switch on one JK flipflop



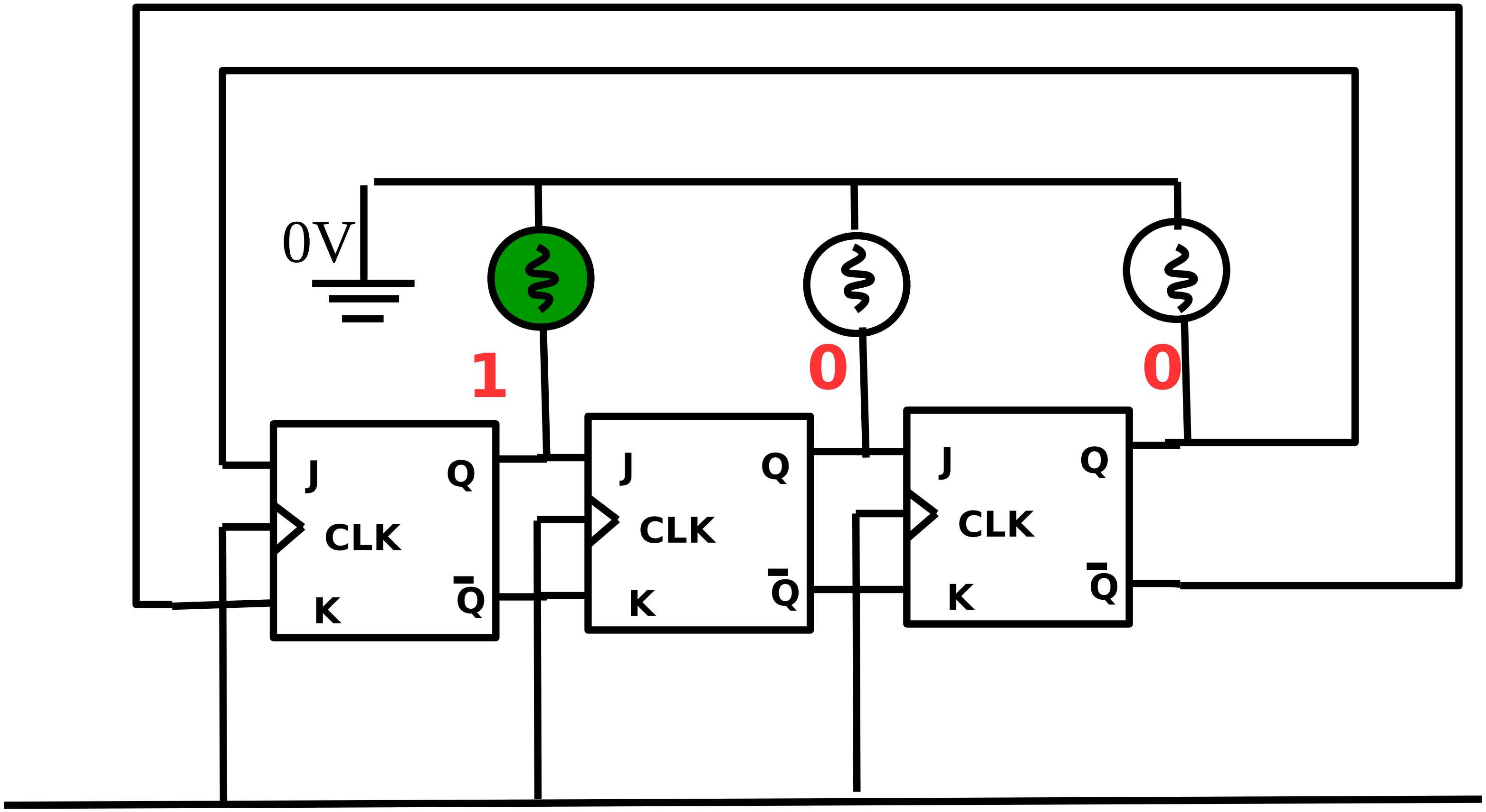
- And get rid of one flipflop

Let's get rid of one flipflop



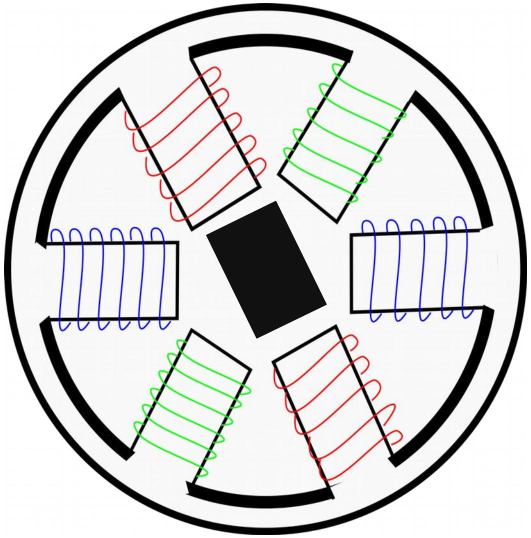
- Let's connect the input to the output.....

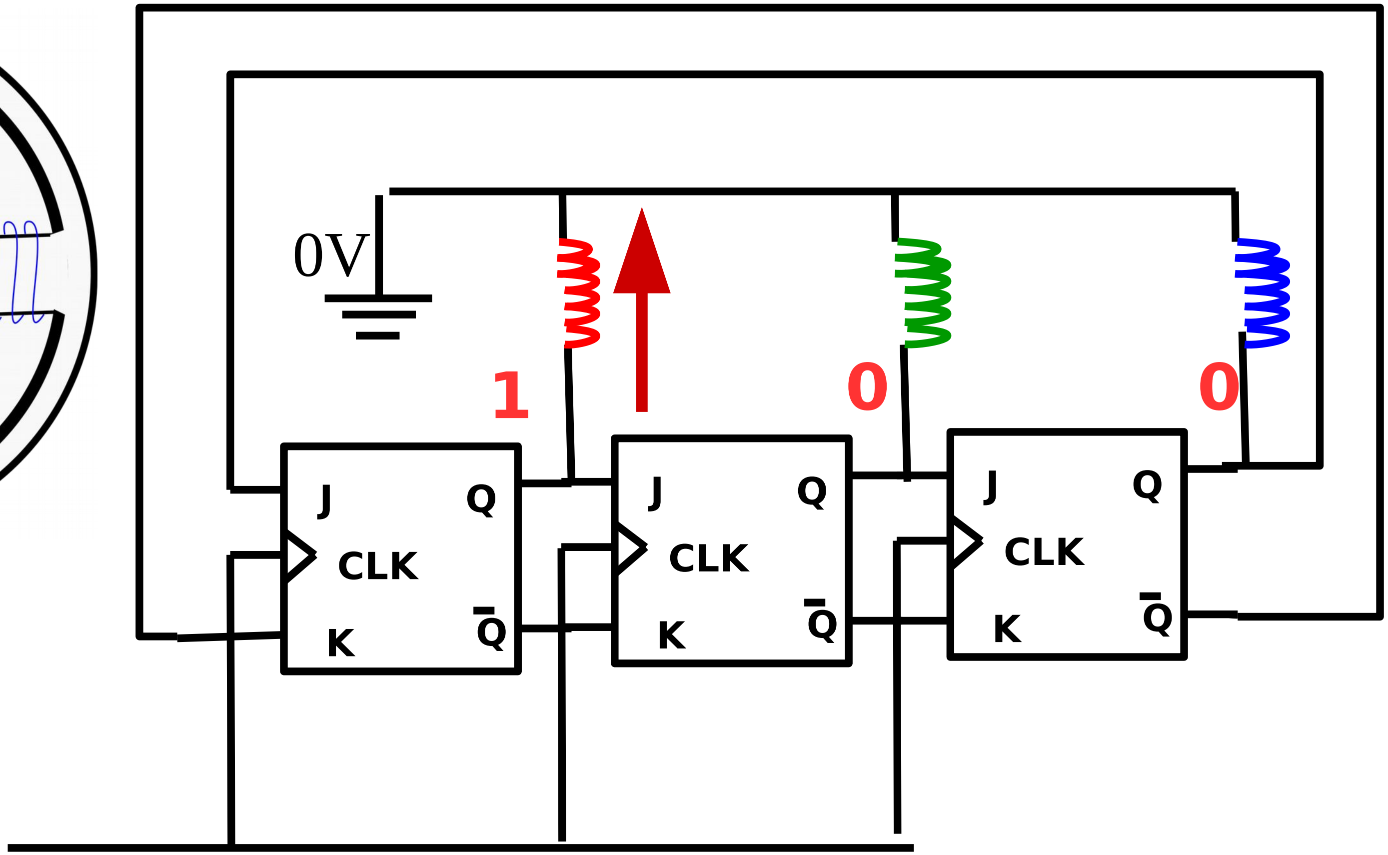
Now let's join the input to the output



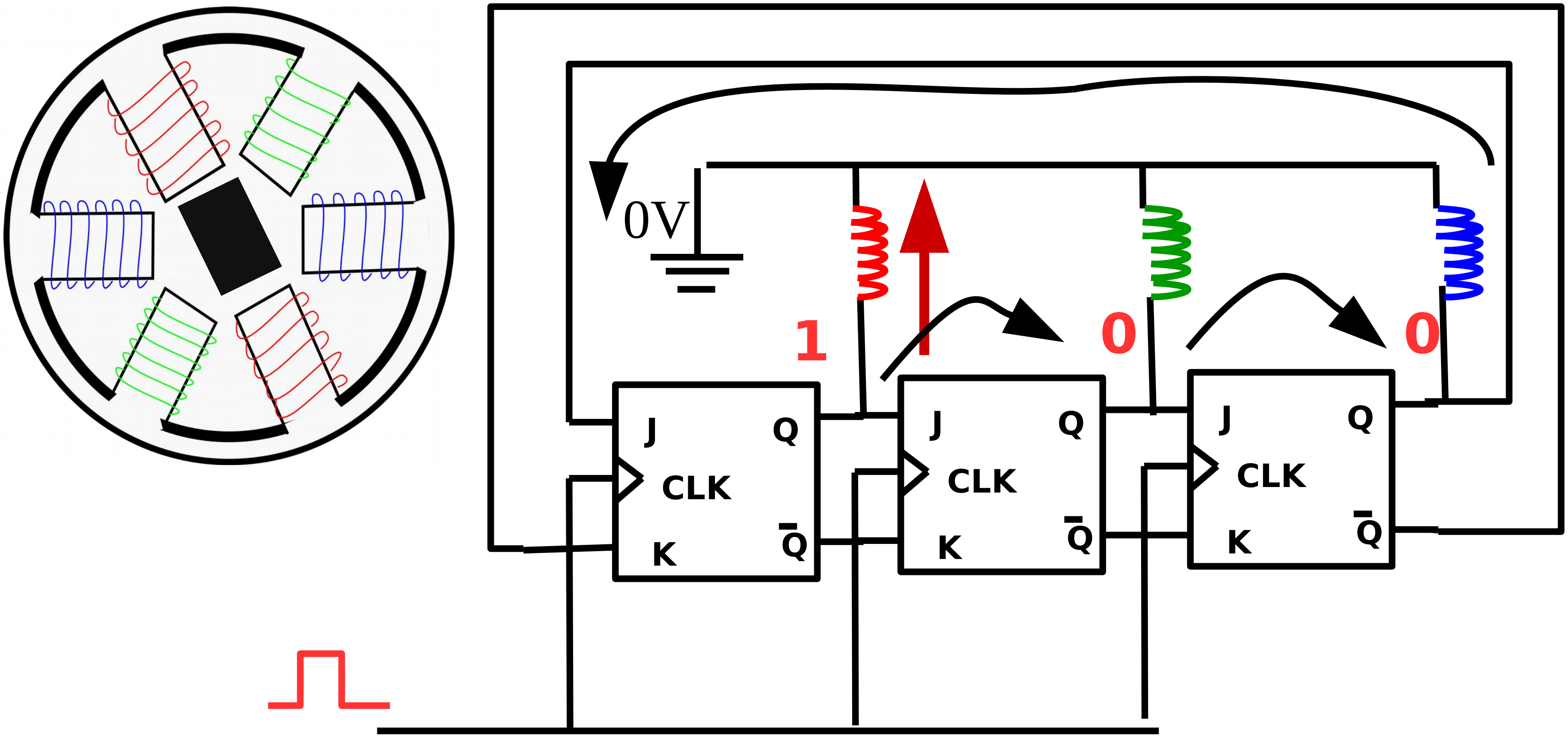
- And replace the bulbs with the coils of the motor....

Replace the bulbs with the coils of the motor.

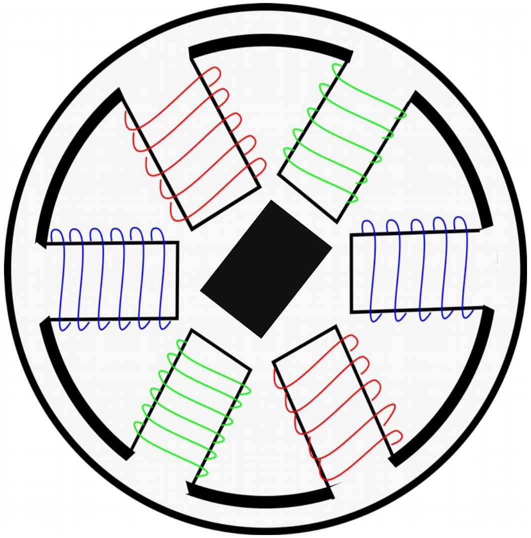


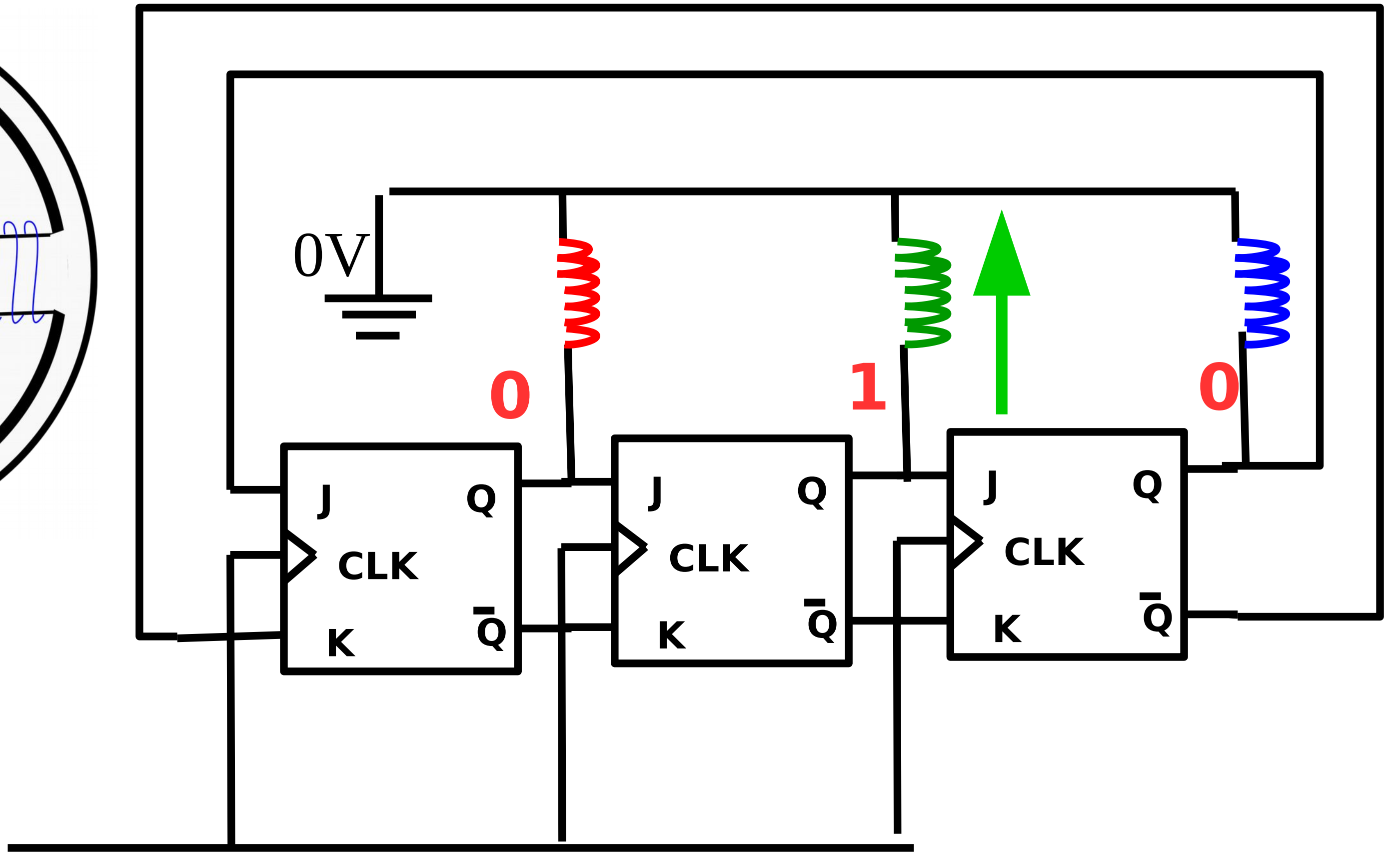


And apply a clock pulse...

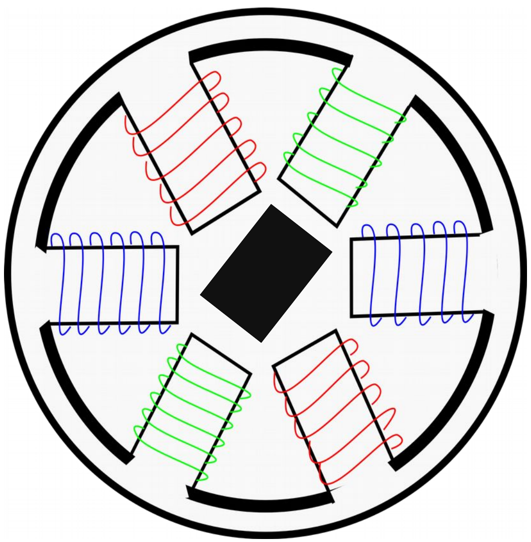


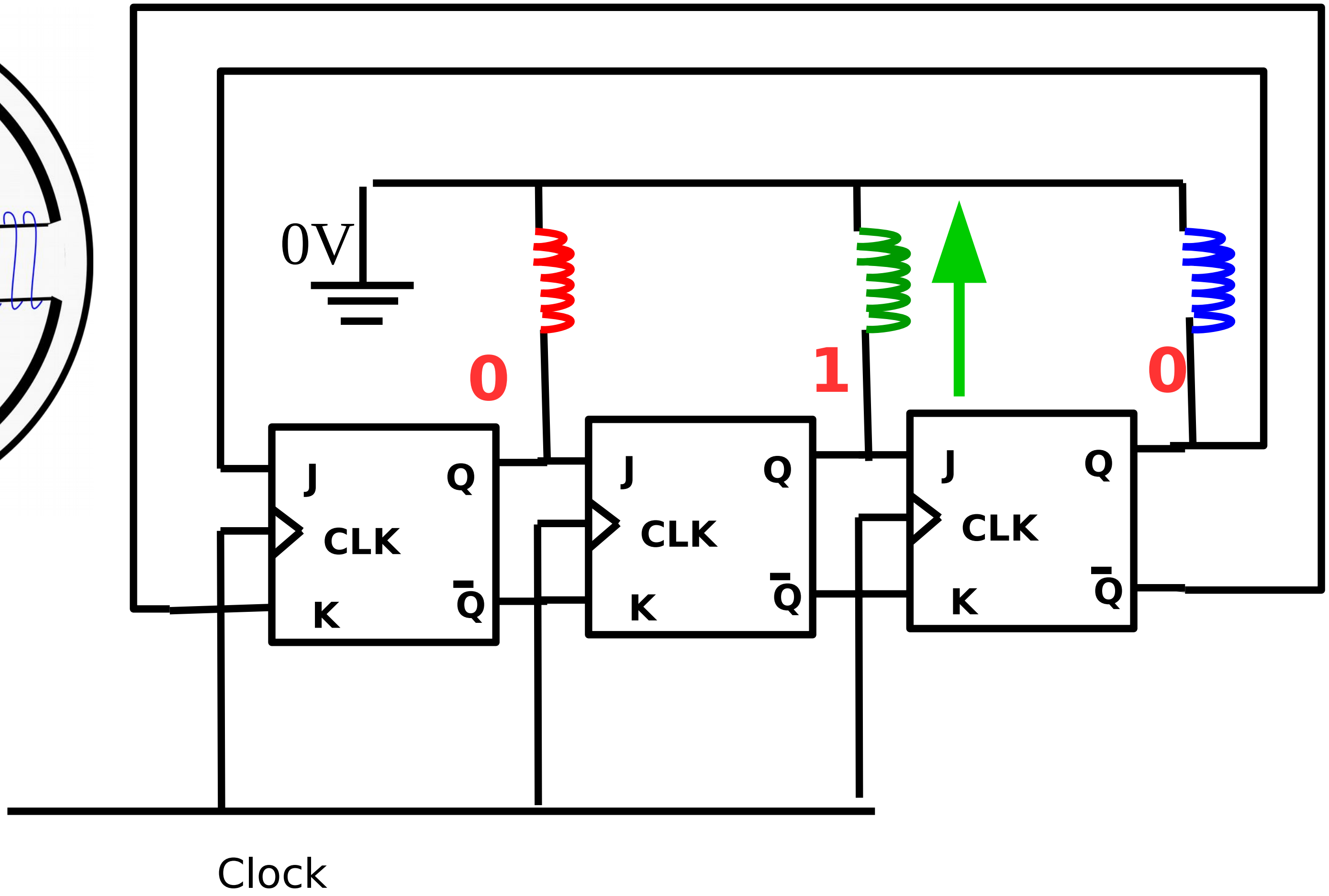
Now let's join the input to the output





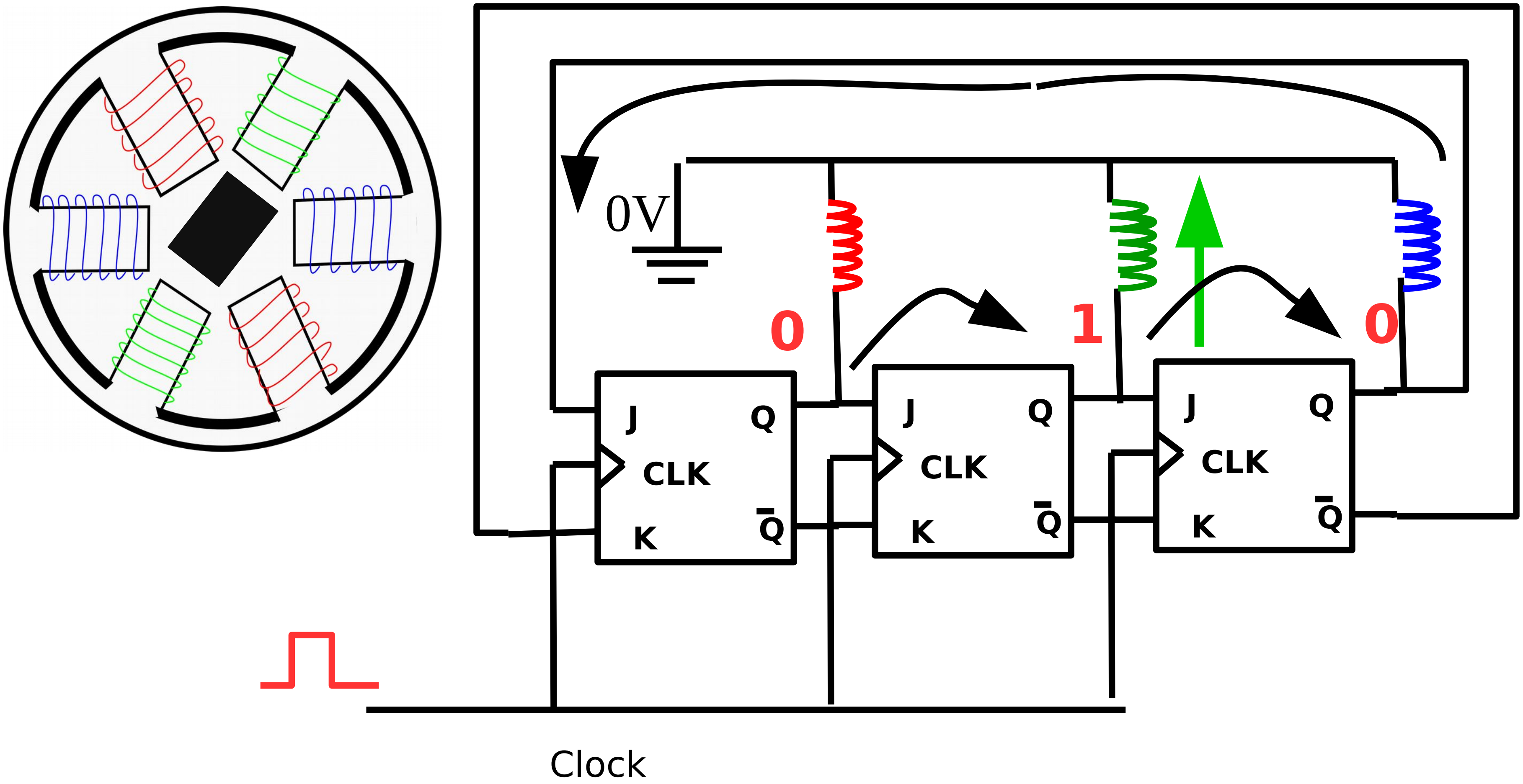
Now let's join the input to the output





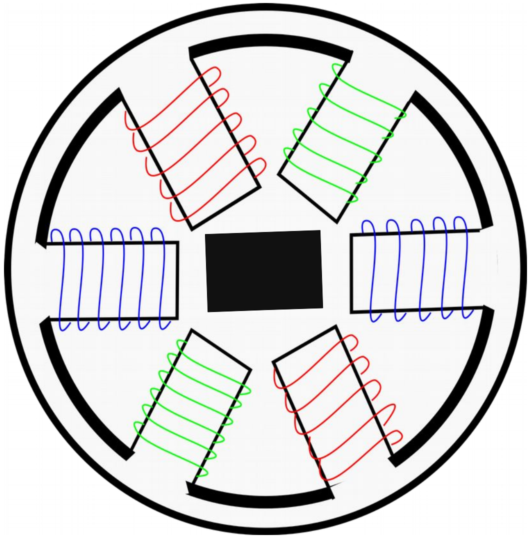
- Now apply another clock pulse.

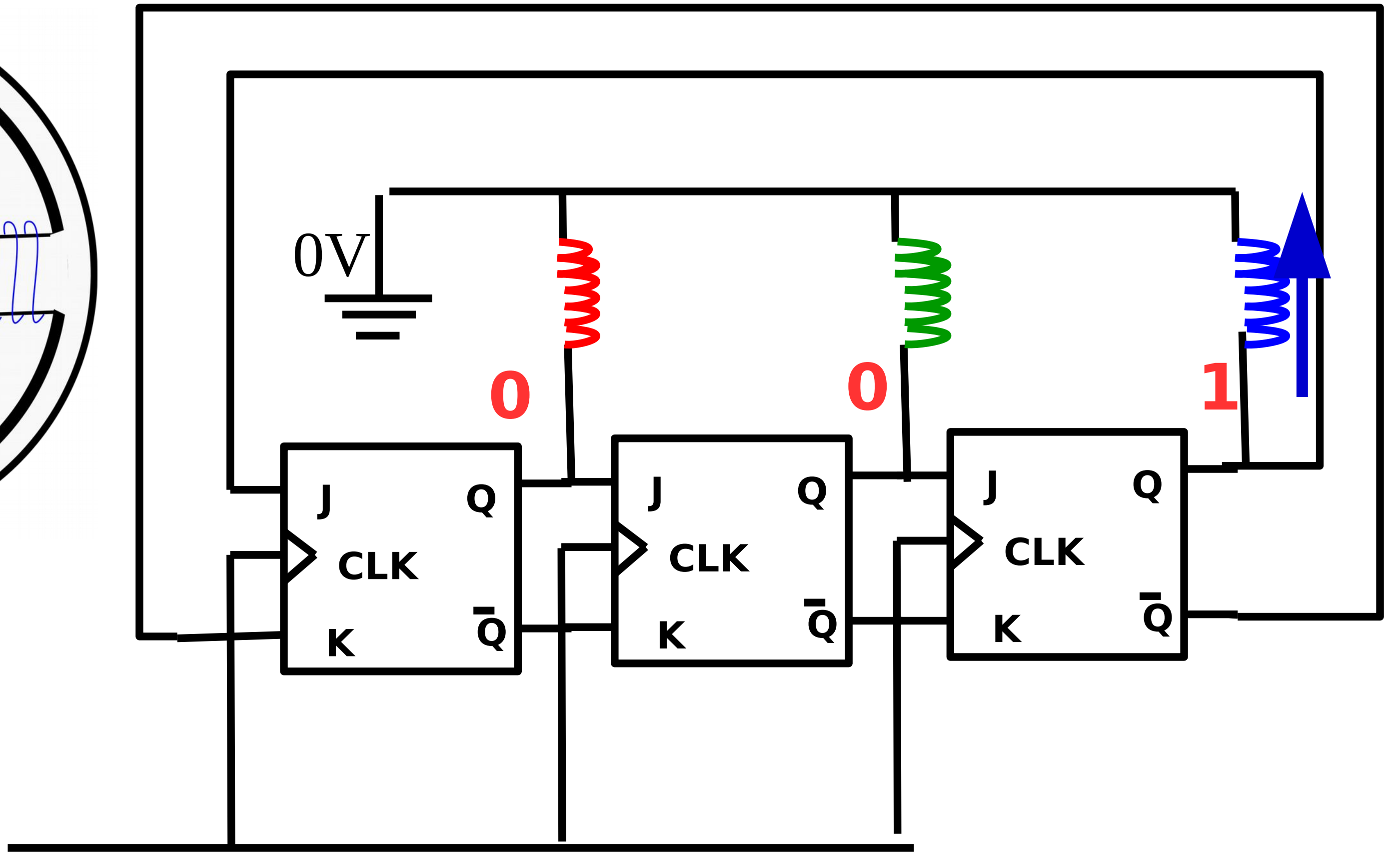
Now let's join the input to the output



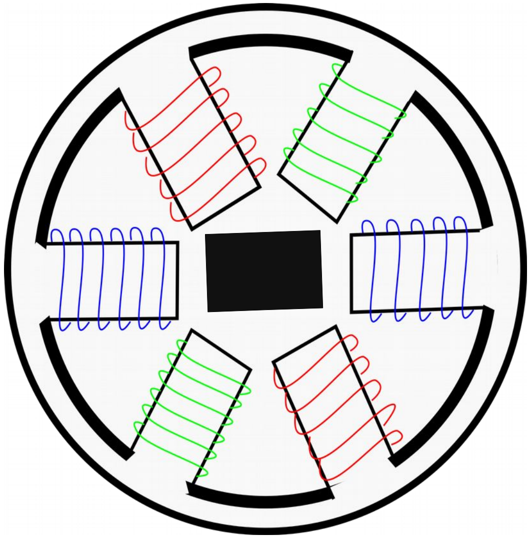
- Now apply another clock pulse.

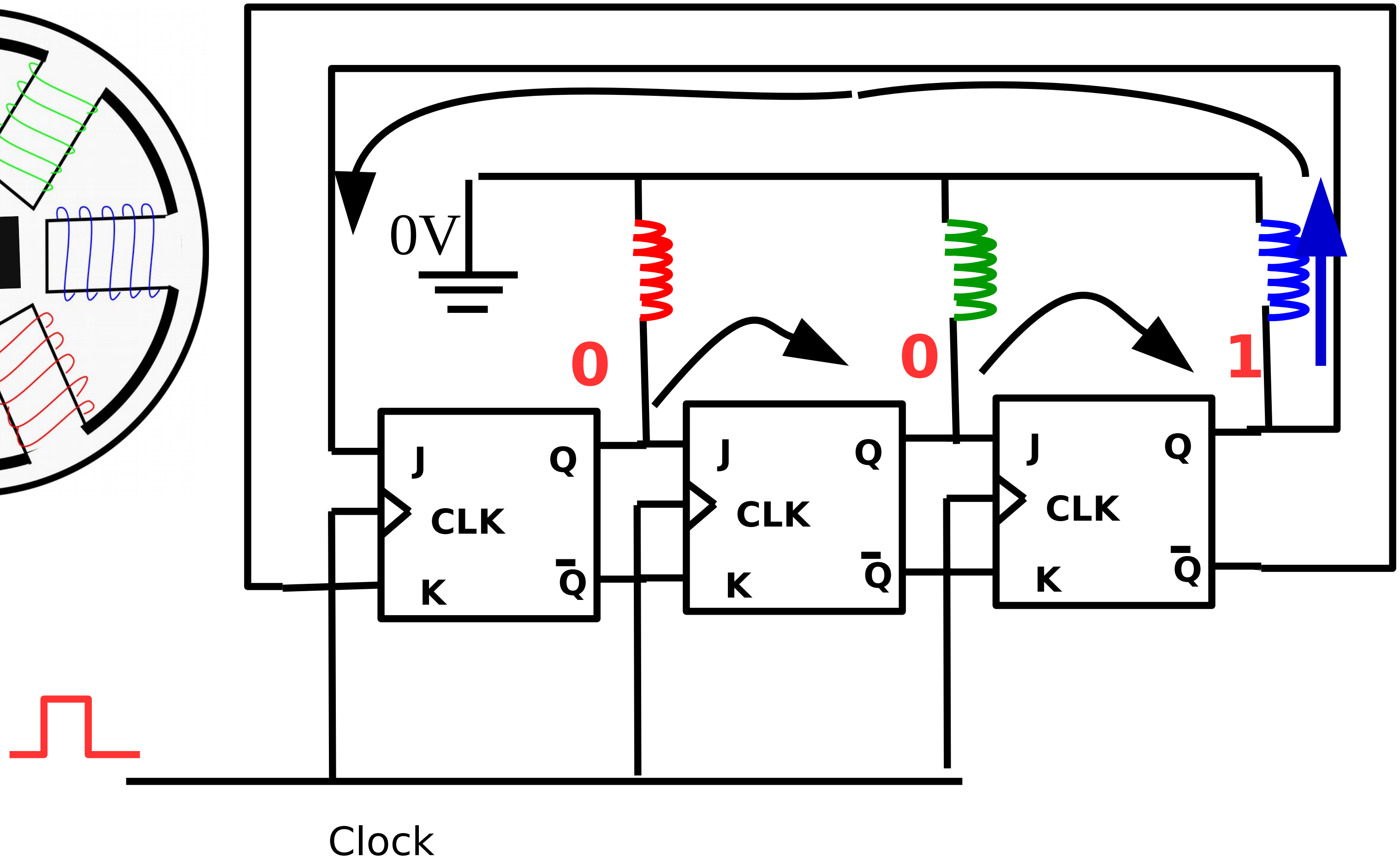
Now let's join the input to the output





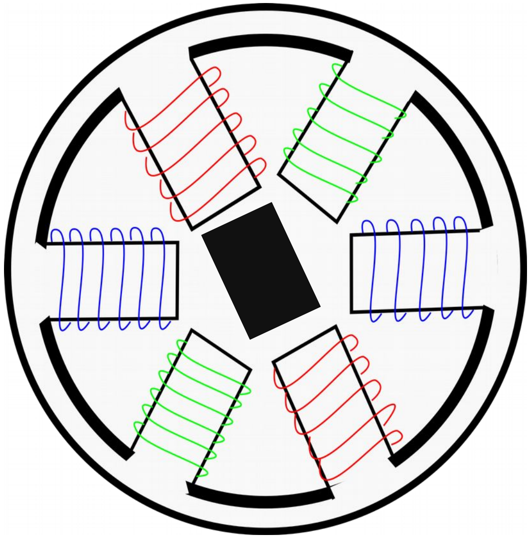
Now let's join the input to the output

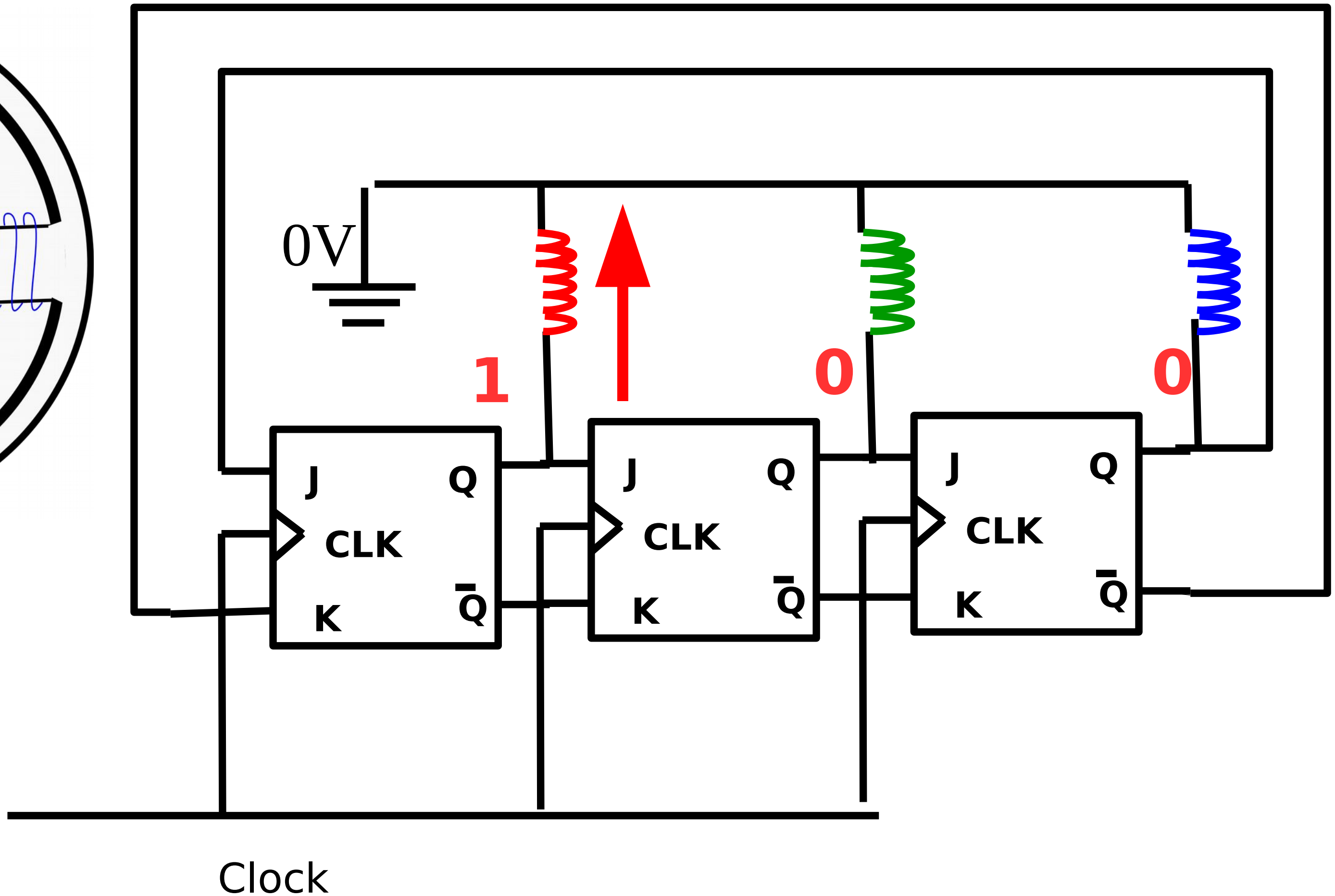




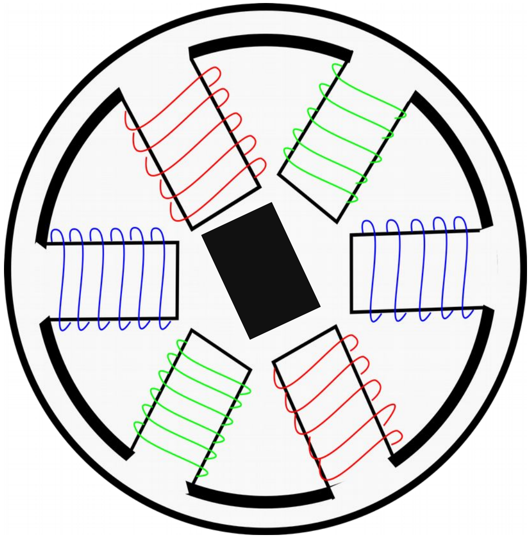
- Now apply another clock pulse.

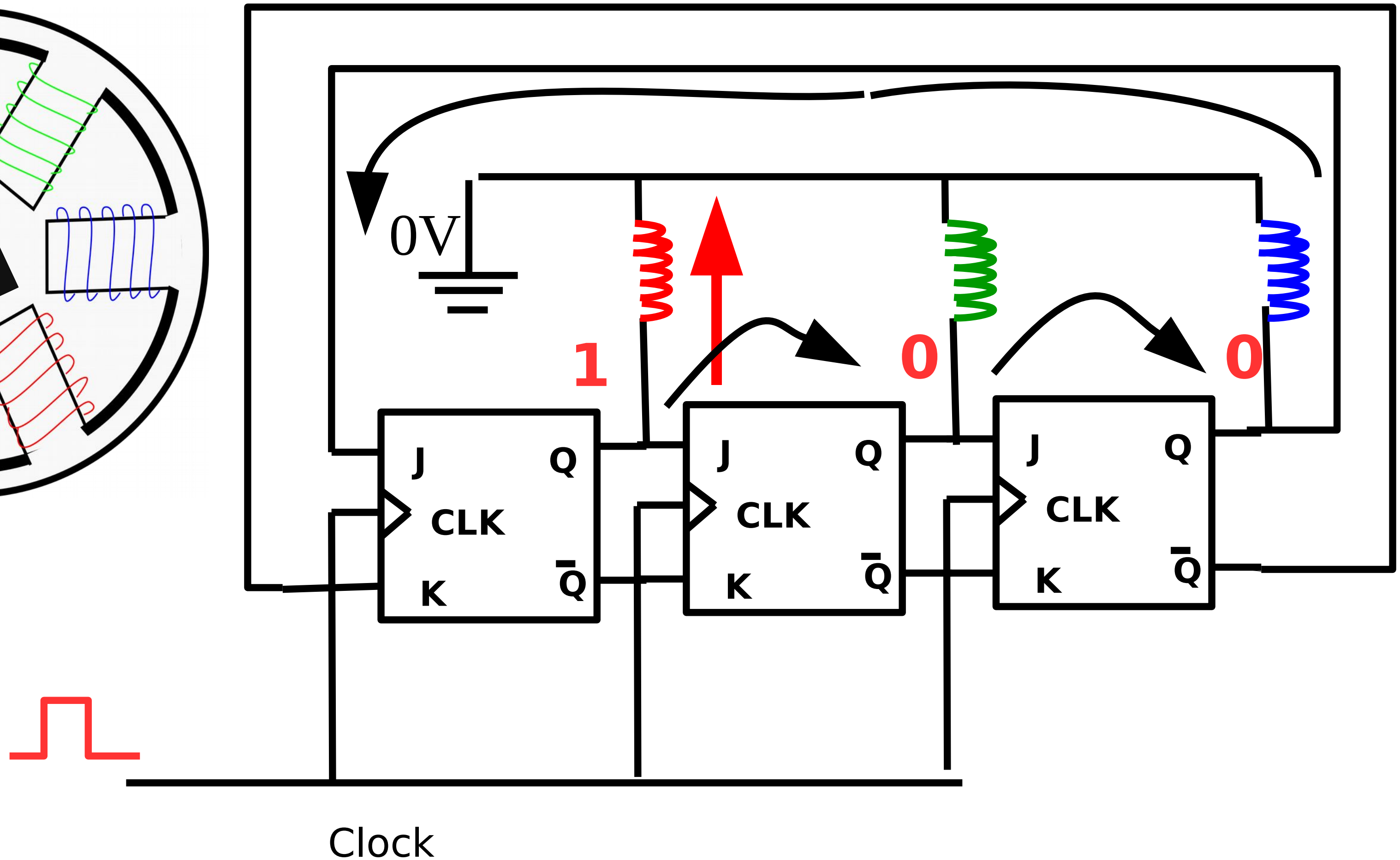
Now let's join the input to the output



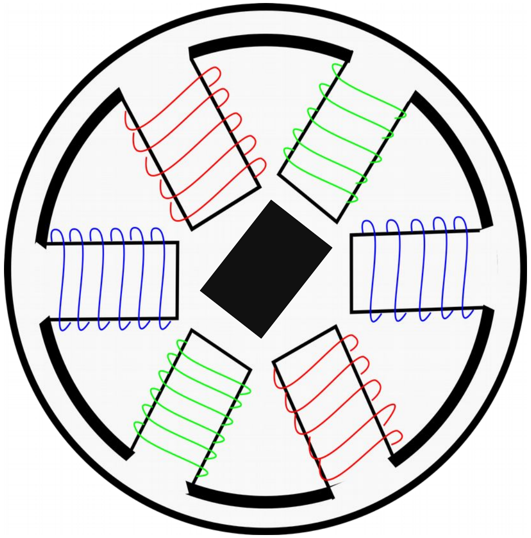


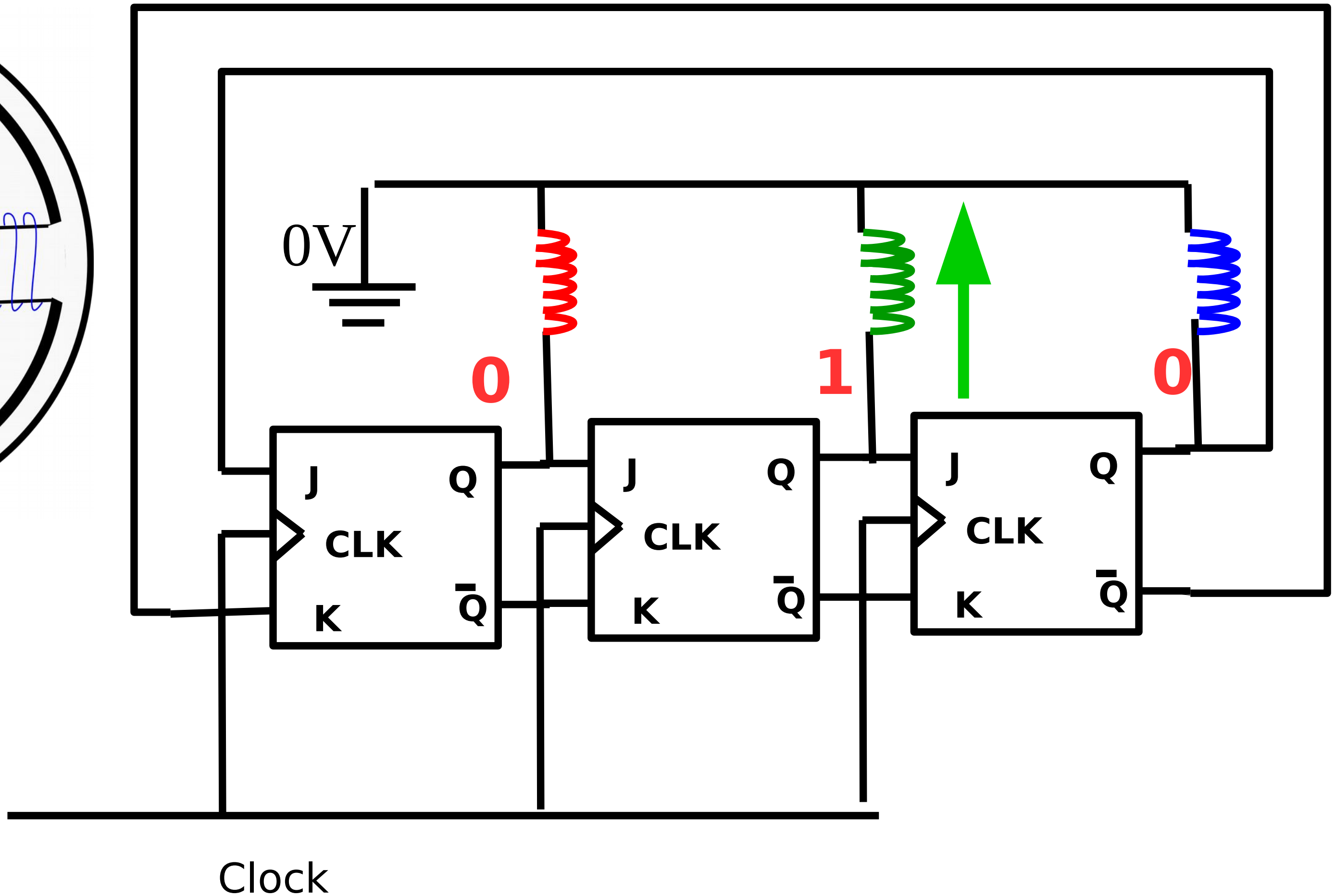
Now let's join the input to the output





Now let's join the input to the output





- Now apply another clock pulse.

Outline of the lecture

- Recap of last lecture

- Using JK flip flops to run motors

- **Digital design**
  - What is digital design?
  - How to do it.
  - Where you might meat digital design
  - FPGAs

- Race times

What is digital design?

- In the first couple of lectures we learnt about:
  - AND gates
  - OR gates
  - NOT gates
  - XOR gates

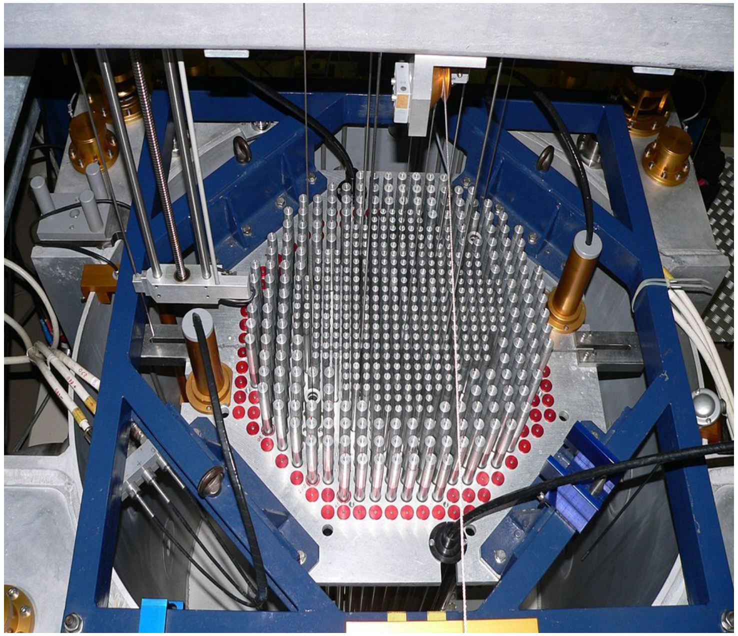
- We also made some simple circuits involving AND gates and OR gates.

- However there are often times when you need to design much more complex circuits as an engineer.

Examples of slightly complex digital circuits

•Imagine you were designing a circuit to monitor a digital thermometer embedded in a nuclear reactor.

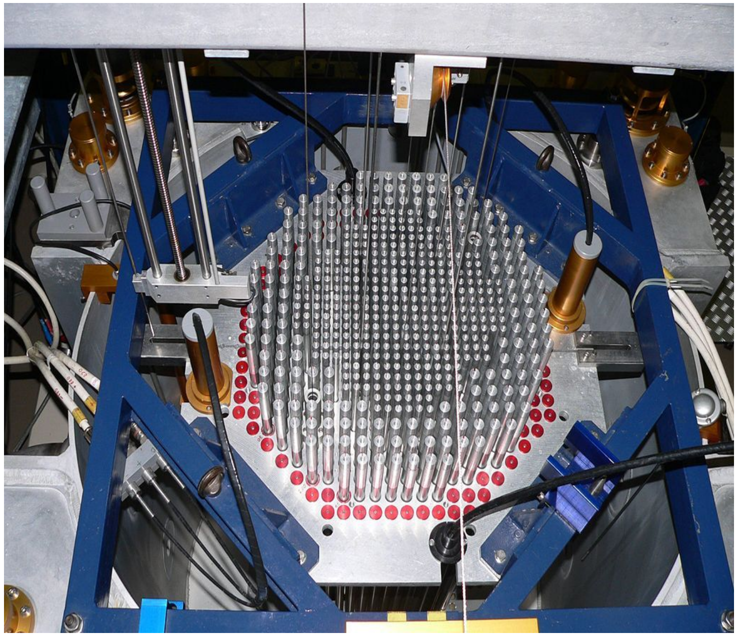
Atomic Energy of Canada Limited



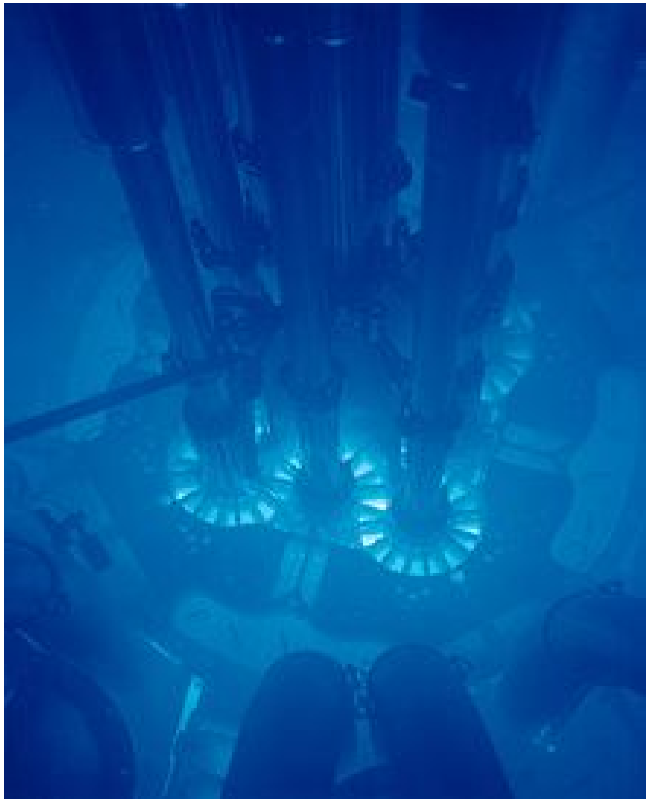
Rama

Examples of slightly complex digital circuits

•You wanted to automatically shut off the reactor when the cooling fluid (water) got higher than 50 degrees.



Rama



Argonne National Laboratory

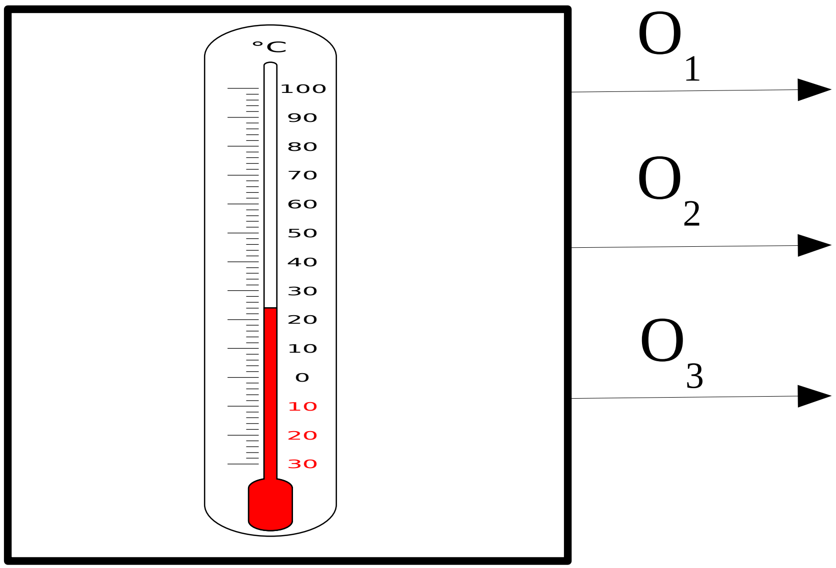
Examples of slightly complex digital circuits

- They did not have one of these circuits in Chernobyl…

- Quite a useful type of circuit to learn about(!)

Examples of slightly complex digital circuits

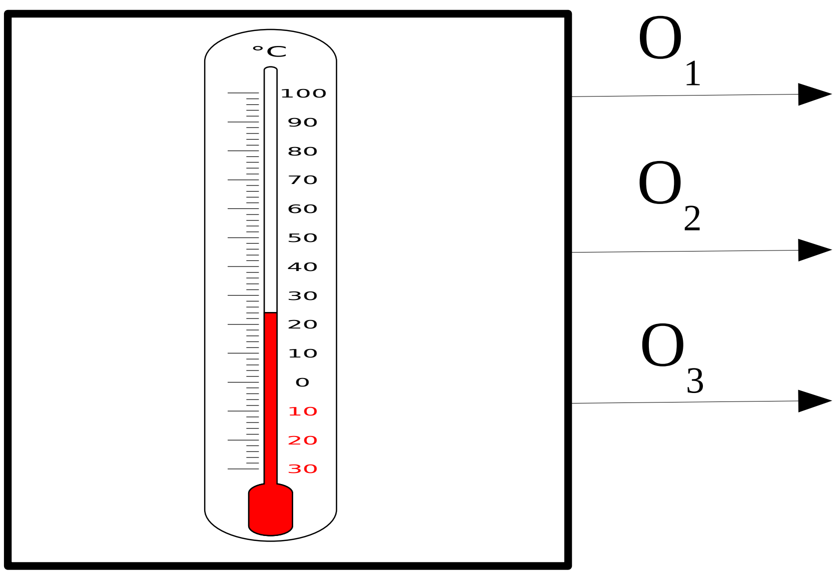
•Your digital thermometer gives it's output using a three digit binary number.



O_1	O_2	O_3
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Examples of slightly complex digital circuits

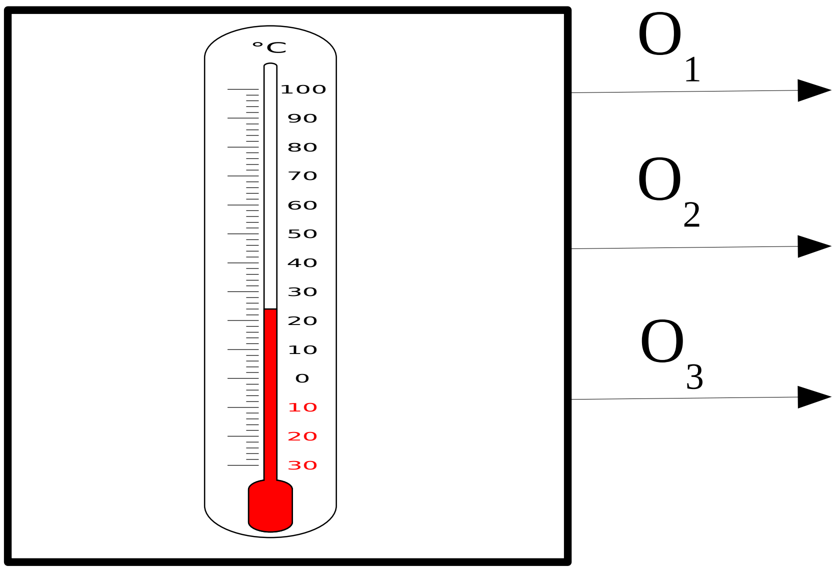
- In normal numbers that is.....



O_1	O_2	O_3	Number
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Examples of slightly complex digital circuits

•Each number corresponds to a 10 degree step in temperature...

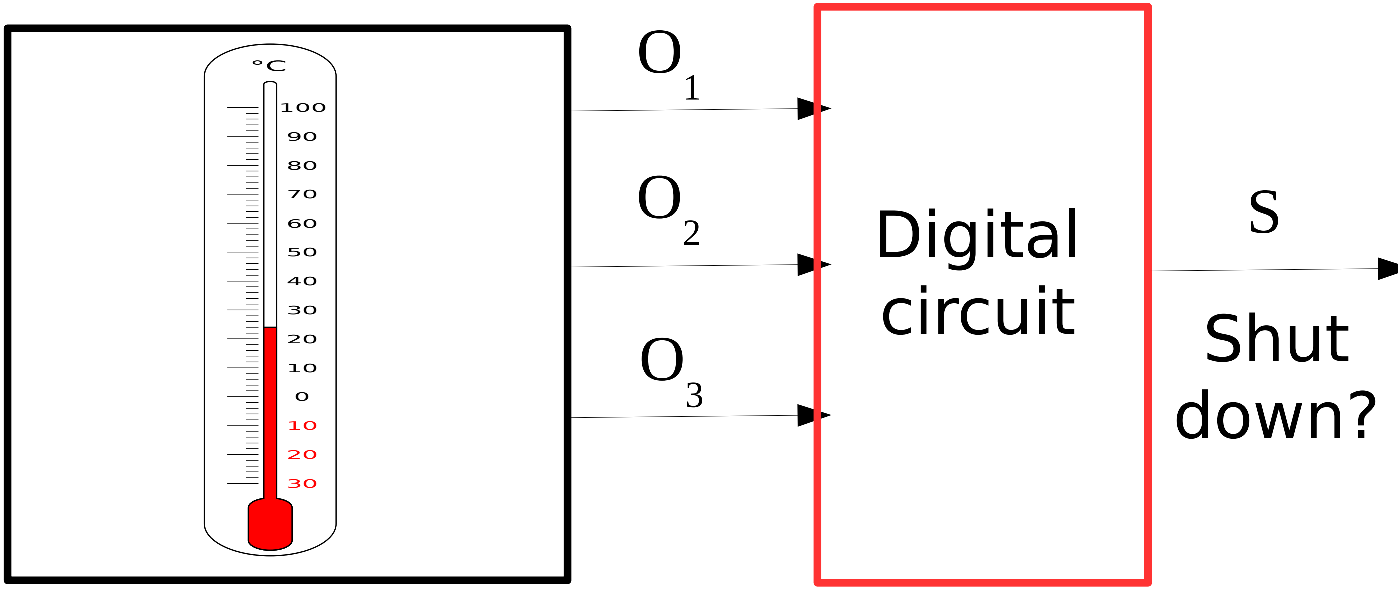


O_1	O_2	O_3	Number	Temperature
0	0	0	0	0
0	0	1	1	10
0	1	0	2	20
0	1	1	3	30
1	0	0	4	40
1	0	1	5	50
1	1	0	6	60
1	1	1	7	70

•We want to design a circuit that shuts off the reactor if the temperature is above 50 degrees.

Examples of slightly complex digital circuits

•We want to design a circuit that shuts off the reactor if the temperature is above 50 degrees.



If **S=1** the reactor will shutdown and if **S=0** the reactor will continue working.

Draw a truth table for the circuit we want to design

O_1	O_2	O_3	Number	Temperature	S
0	0	0	0	0	0
0	0	1	1	10	0
0	1	0	2	20	0
0	1	1	3	30	0
1	0	0	4	40	0
1	0	1	5	50	0
1	1	0	6	60	1
1	1	1	7	70	1

- Remember
  - S=0 means stay on
  - S=1 shutdown

Truth table

O_1	O_2	O_3	Number	Temperature	S
0	0	0	0	0	0
0	0	1	1	10	0
0	1	0	2	20	0
0	1	1	3	30	0
1	0	0	4	40	0
1	0	1	5	50	0
1	1	0	6	60	1
1	1	1	7	70	1

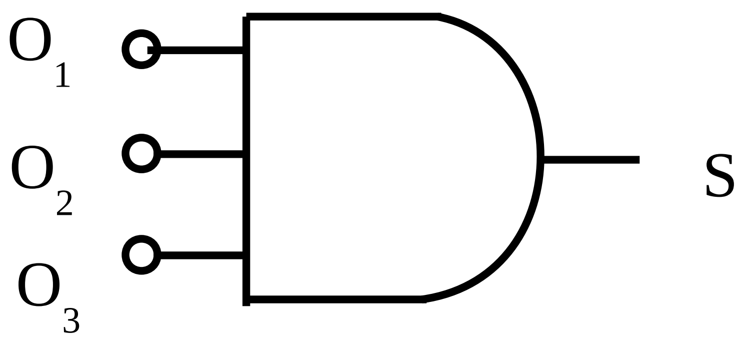
- We need to construct two logic circuits that give a 1 when the inputs are 110 and 111.

- Let's look at the 111 case first because it's easy

Let's draw a logic table

- We want a circuit that will produce a 1 when all inputs are 1

- Let's just use an AND agate with three inputs.



O_1	O_2	O_3	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Now let's make a circuit for the 60 degree case

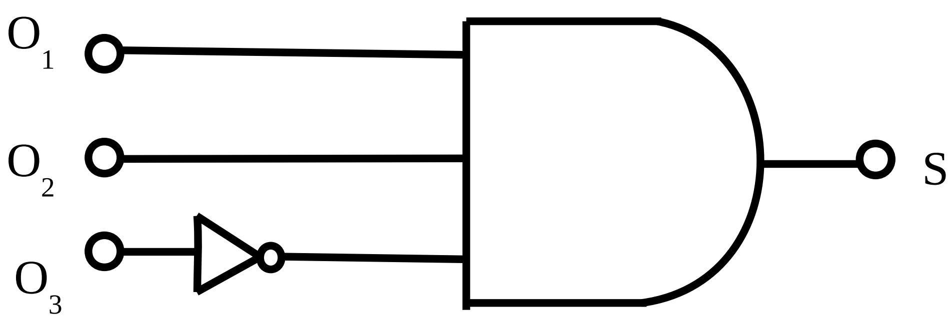
O_1	O_2	O_3	Number	Temperature	S
0	0	0	0	0	0
0	0	1	1	10	0
0	1	0	2	20	0
0	1	1	3	30	0
1	0	0	4	40	0
1	0	1	5	50	0
1	1	0	6	60	1
1	1	1	7	70	1

- We want a circuit that will produce a 1 when all inputs are 1 except O_3.

Let's draw a logic table

- We want a circuit that will produce a 1 when all inputs are 1 except O_3.

  - Let's just make an and gate with input O_3 inverted.



O_1	O_2	O_3	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

One more example

- Presumably it would also be bad if our coolant froze so let's make a circuit to shut down the reactor if the temperature is freezing

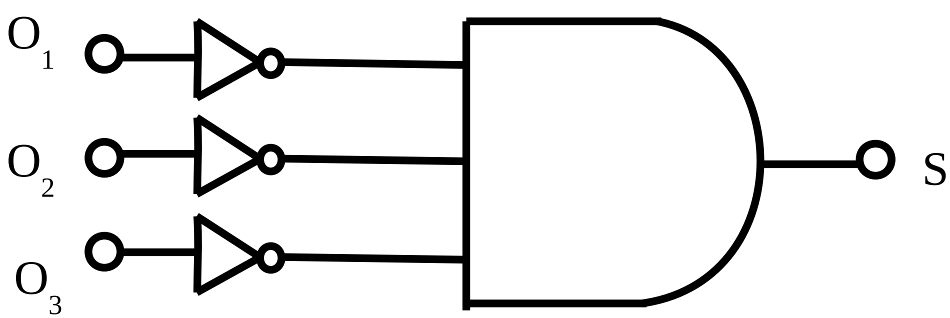
O_1	O_2	O_3	Number	Temperature	S
0	**0**	**0**	**0**	**0**	**1**
0	0	1	1	10	0
0	1	0	2	20	0
0	1	1	3	30	0
1	0	0	4	40	0
1	0	1	5	50	0
1	1	0	6	60	1
1	1	1	7	70	1

Ian Mackenzie

Let's draw a logic table

•So now we want our circuit to produce a 1 when O1, O2 and O3 are all set to 0.

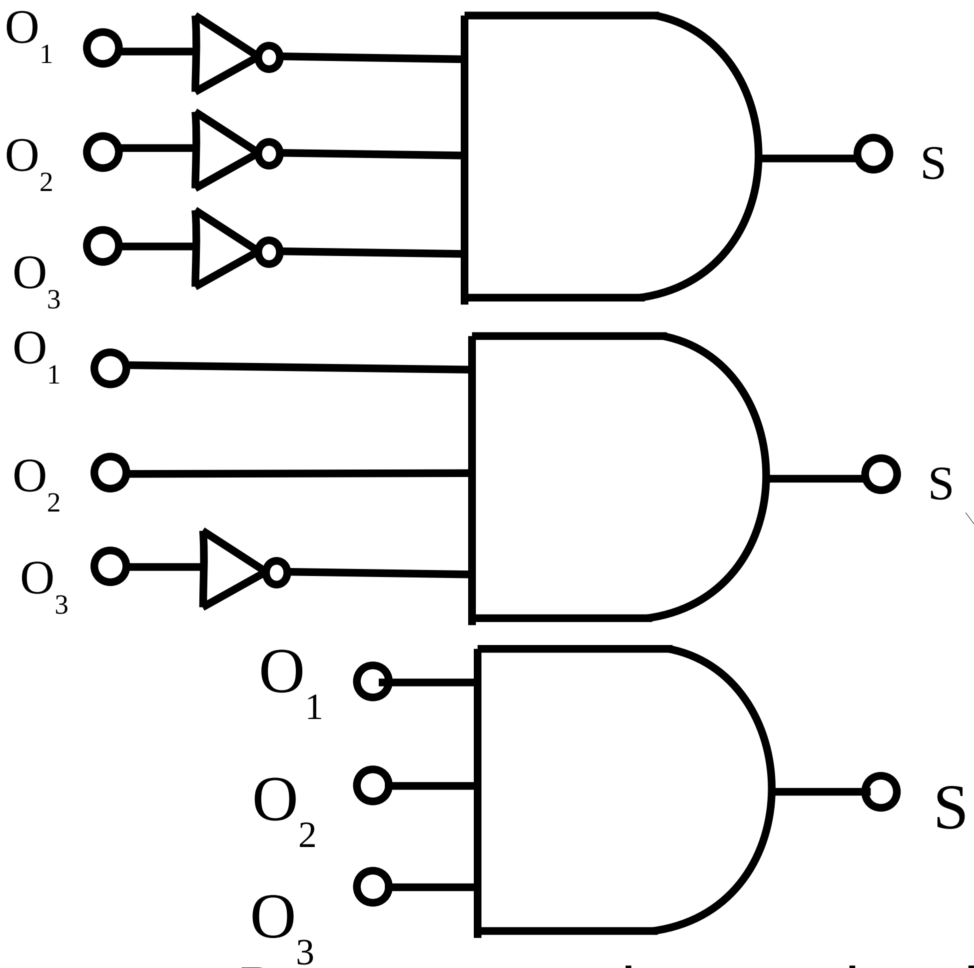
So let's just invert all the inputs. So the AND gate only turns on when all inputs are 0.



O_1	O_2	O_3	S
0	**0**	**0**	**1**
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

The final circuit (sort of)

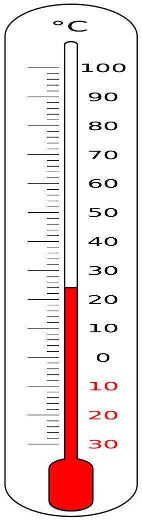
- We now have three circuits which all give one when our criteria to shut down the reactor are met.

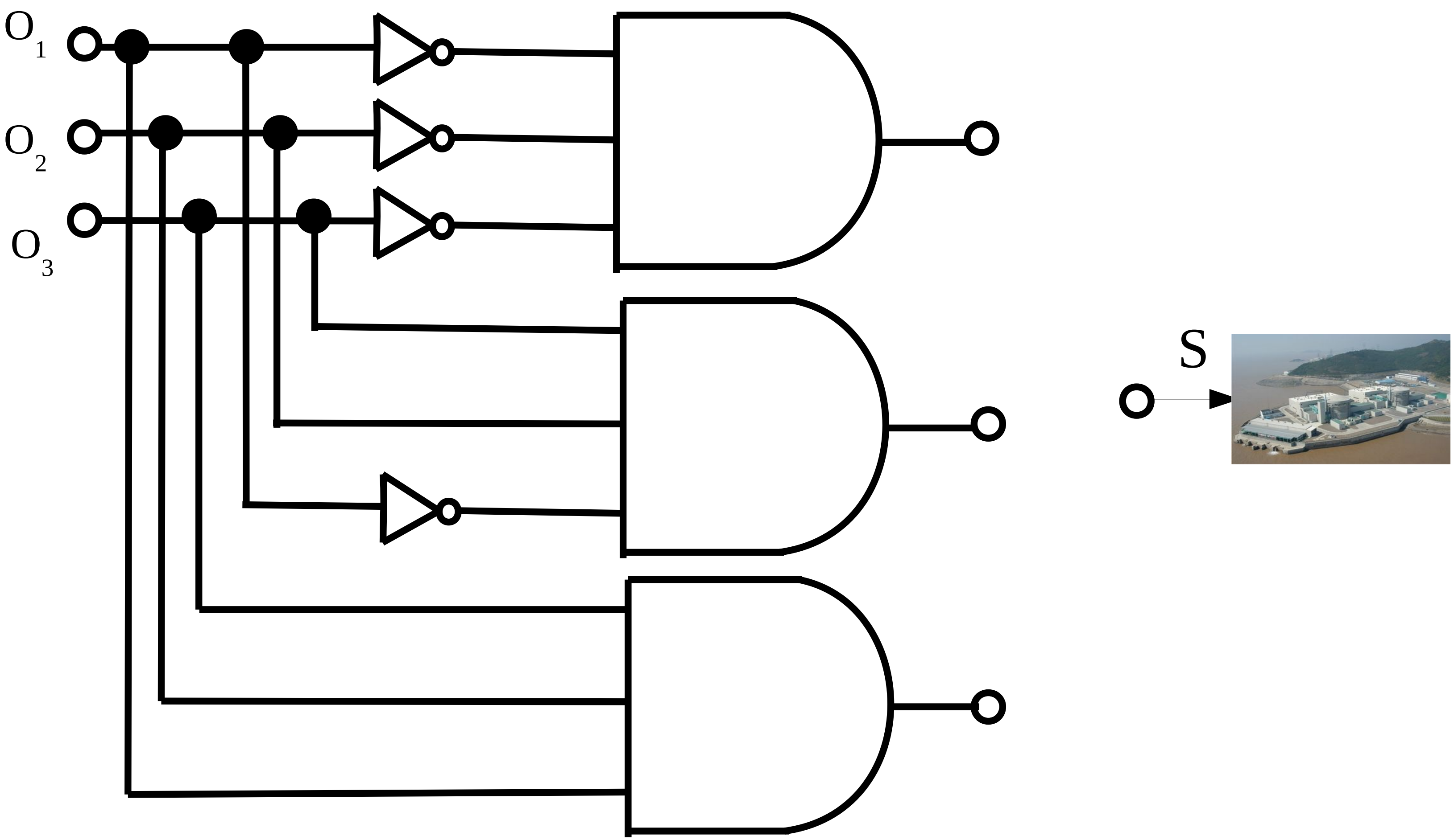


O_1	O_2	O_3	S
0	**0**	**0**	**1**
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	**1**	**0**	**1**
1	**1**	**1**	**1**

- But we need to make this into one circuit......

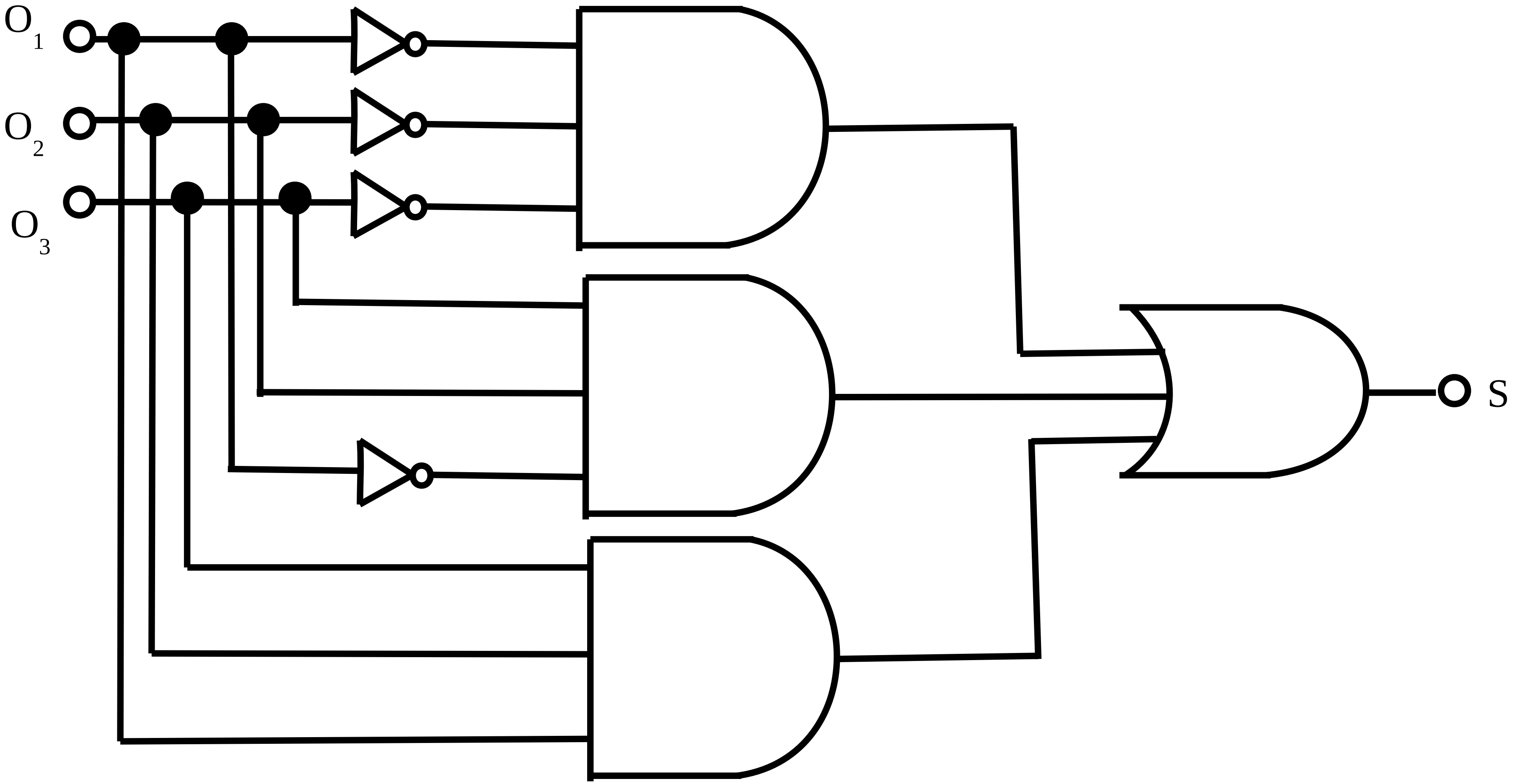
Join together the inputs





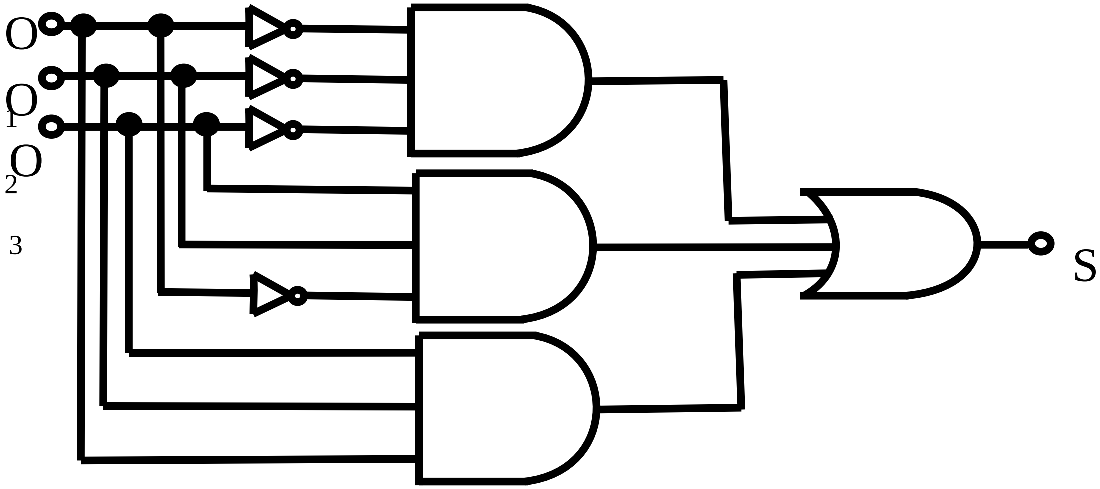
- Looking good, now let's join the outputs together with an OR gate.

Join together the inputs



- Finished!!!

Join together the inputs



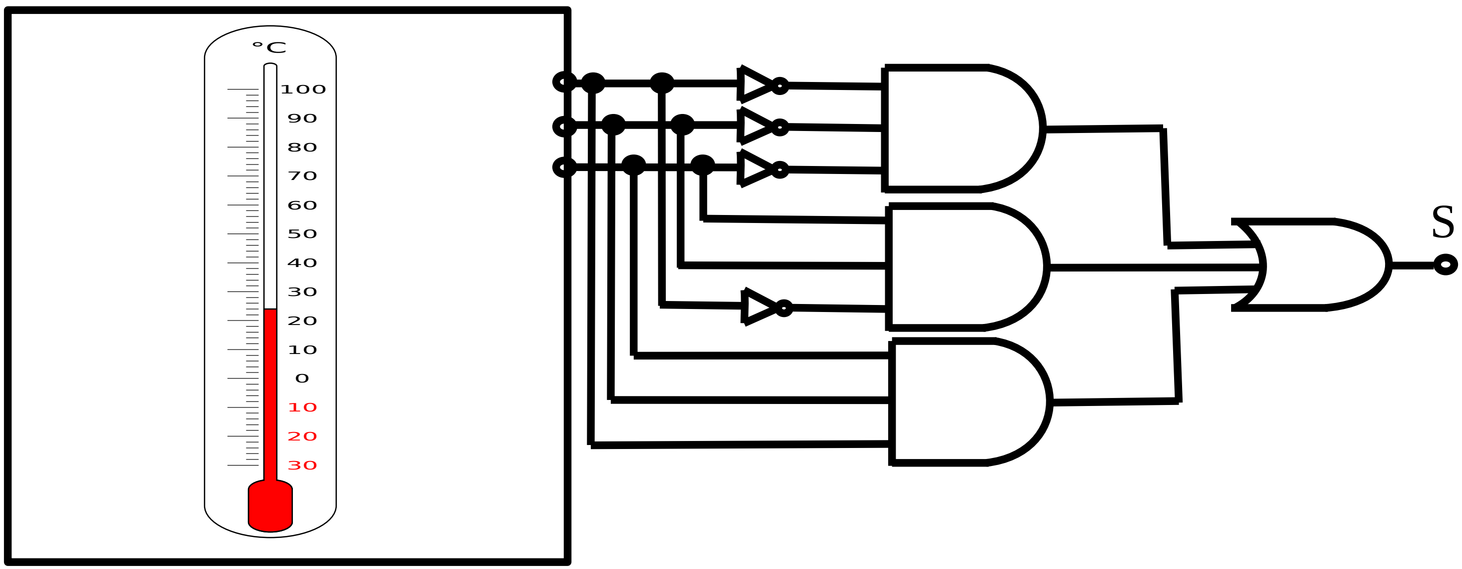
•This circuit will produce this truth table.

O_1	O_2	O_3	S
0	**0**	**0**	**1**
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	**1**	**0**	**1**
1	**1**	**1**	**1**

•Finished!!!

Examples of slightly complex digital circuits

•And we have correctly interfaced our thermometer to the nuclear power plant shut down circuit.



•And our nuclear power plant can no longer blow up!

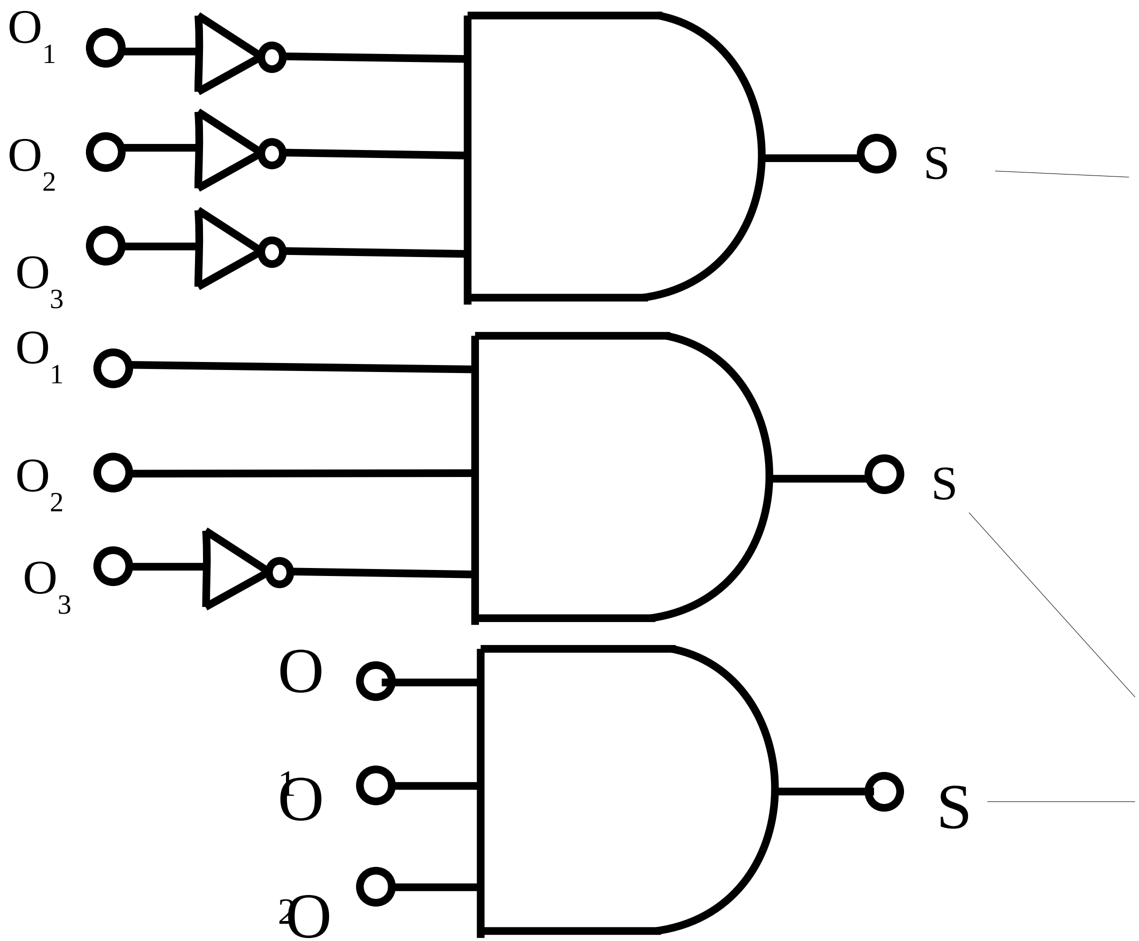
Recap – how to design digital circuits 1

- Write out the truth table

O_1	O_2	O_3	S
0	**0**	**0**	**1**
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	**1**	**0**	**1**
1	**1**	**1**	**1**

Recap – how to design digital circuits 2

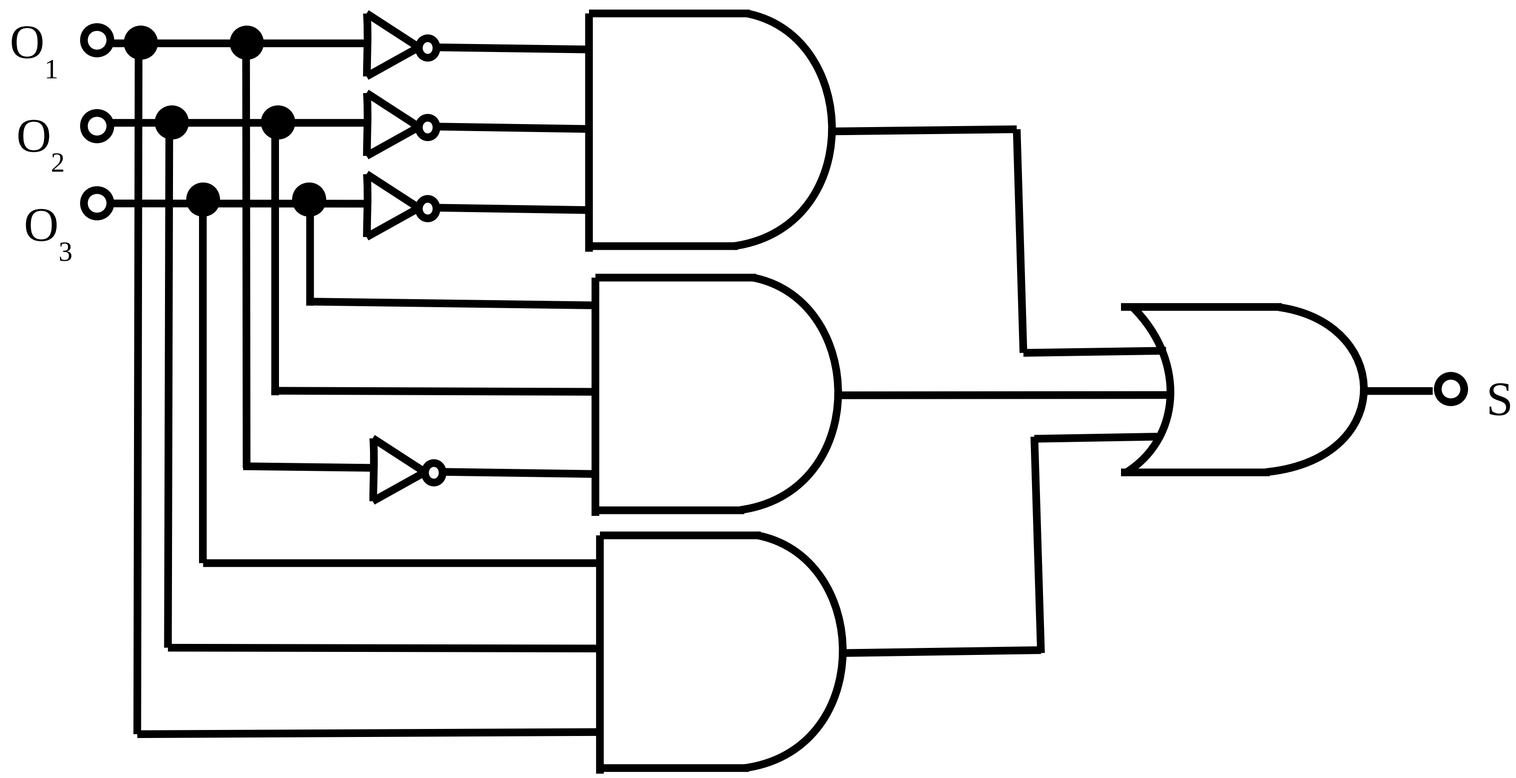
- Use AND and NOT gates that produce circuits that only give a 1 when a set of desired inputs is met



O_1	O_2	O_3	S
0	**0**	**0**	**1**
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	**1**	**0**	**1**
1	**1**	**1**	**1**

Recap – how to design digital circuits 3

- Stitch the inputs together and join the outputs with an OR gate.



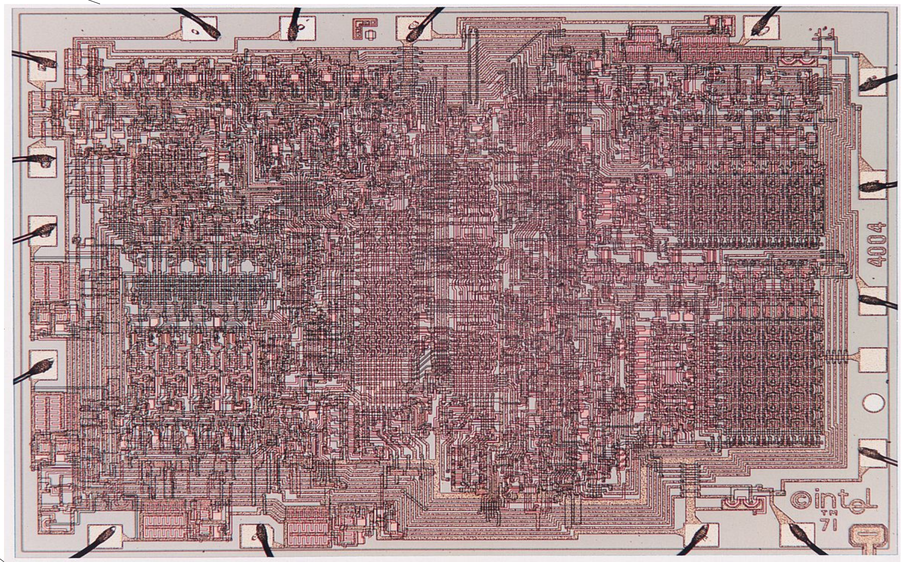
- And that is it!

Summary digital design

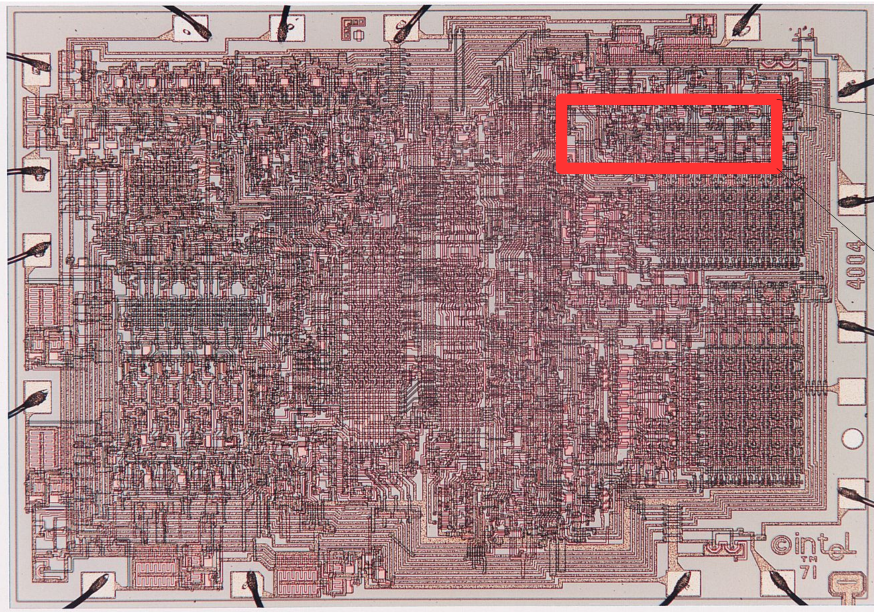
•You can now design any digital circuit that is thrown at you if you follow those simple steps.

• You can now even design the circuits that go into computer chips using this method

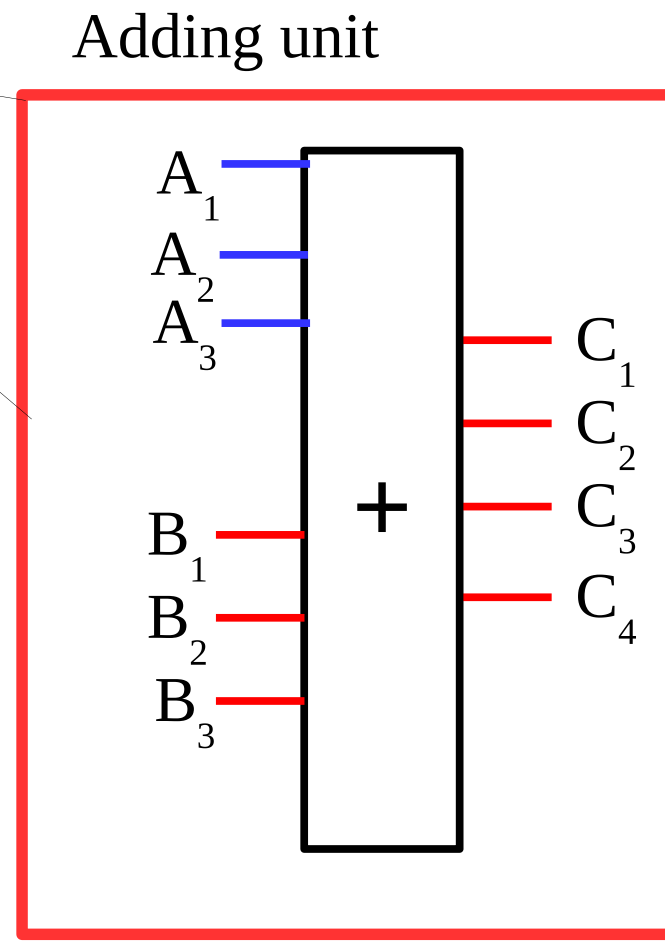
Processor



Adding unit



The electronics inside a processor



- Simply draw out the truth tables and you would get the answer.

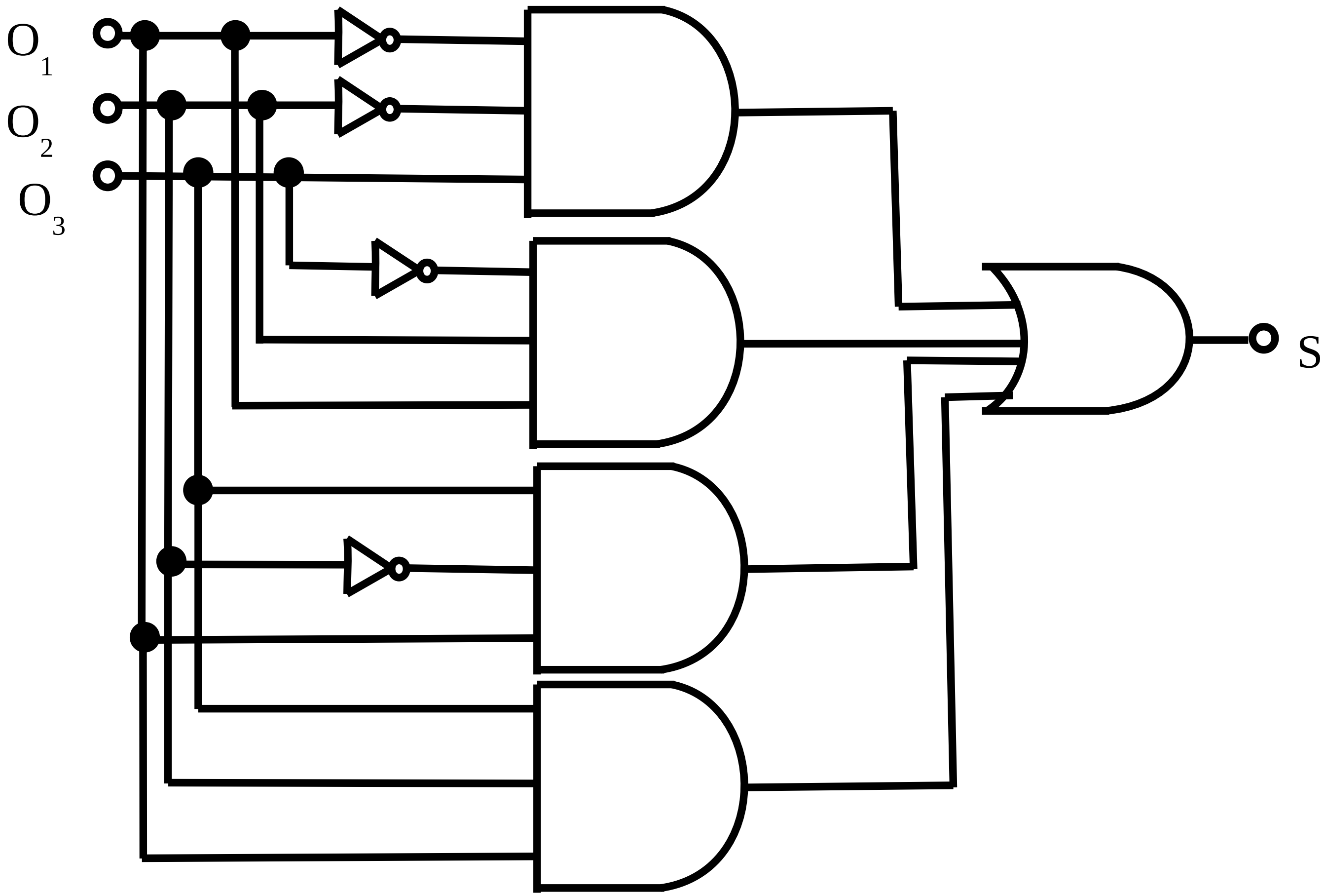
Now' its your turn

•However that's a bit complex to do in the lecture. So you will now design a circuit to detect if a number is odd:

•Hint you need four AND gates, one OR gate, and a collection of NOT gates.

O_1	O_2	O_3	S
0	**0**	**0**	**0**
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	**1**	**0**	**0**
1	**1**	**1**	**1**

- The answer



O_1	O_2	O_3	S
0	**0**	**0**	**0**
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	**1**	**0**	**0**
1	**1**	**1**	**1**

However

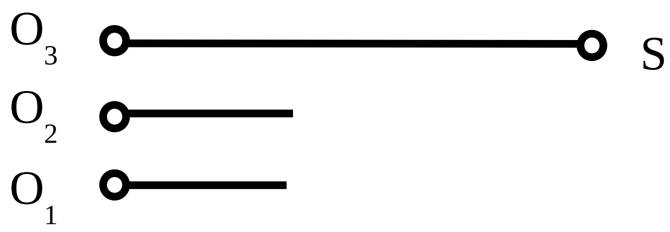
- Do you notice something about this truth table??

Value	O_1	O_2	O_3	S
0	**0**	**0**	**0**	**0**
1	0	0	**1**	**1**
2	0	1	0	0
3	0	1	**1**	**1**
4	1	0	0	0
5	1	0	**1**	**1**
6	**1**	**1**	**0**	**0**
7	**1**	**1**	**1**	**1**

- When ever O_3 is one we have an odd number

A much simpler circuit

•We could actually replace this whole circuit with a wire!! (in this case):

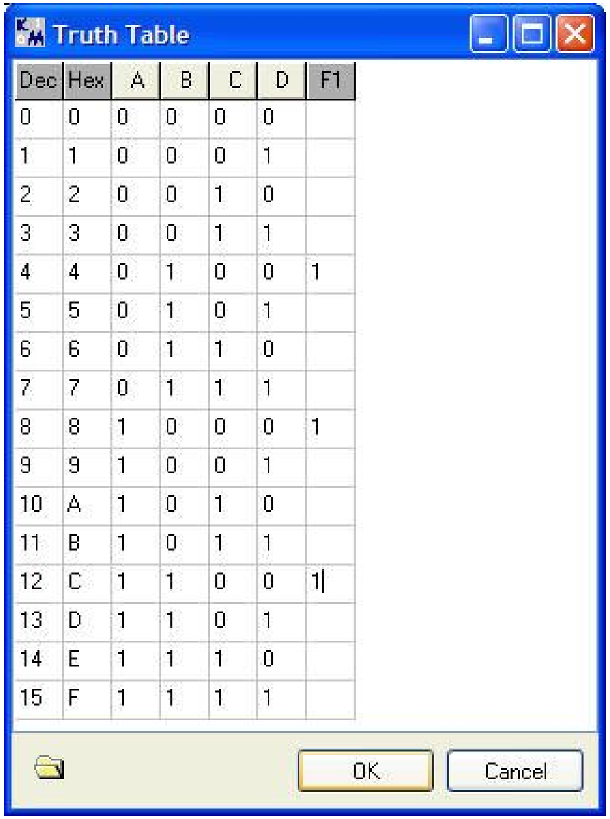


•The method I have just taught you will work 100% of the time without any problems.

•However, it will not try to simplify the circuit at all and you may end up using far more gates than you really need.

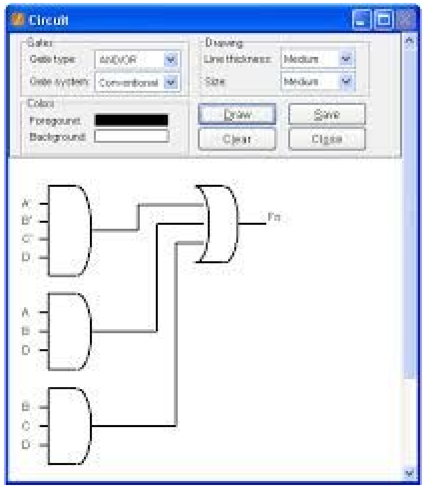
A much simpler circuit

•There are programs which will take a truth table and convert in into a minimized circuit which uses the least number of gates that are possible.


Truth Table

Dec	Hex	A	B	C	D	F1
0	0	0	0	0	0	
1	1	0	0	0	1	
2	2	0	0	1	0	
3	3	0	0	1	1	
4	4	0	1	0	0	1
5	5	0	1	0	1	
6	6	0	1	1	0	
7	7	0	1	1	1	
8	8	1	0	0	0	1
9	9	1	0	0	1	
10	A	1	0	1	0	
11	B	1	0	1	1	
12	C	1	1	0	0	1
13	D	1	1	0	1	
14	E	1	1	1	0	
15	F	1	1	1	1	

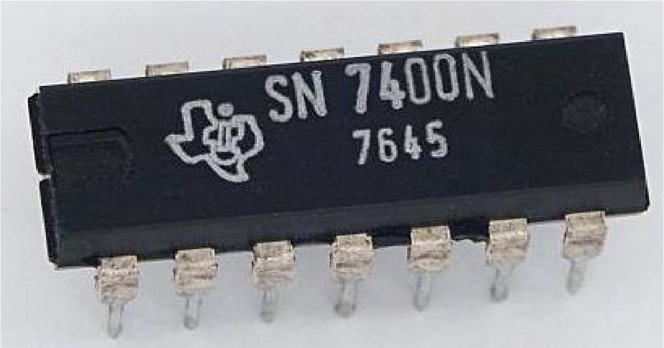
OK Cancel

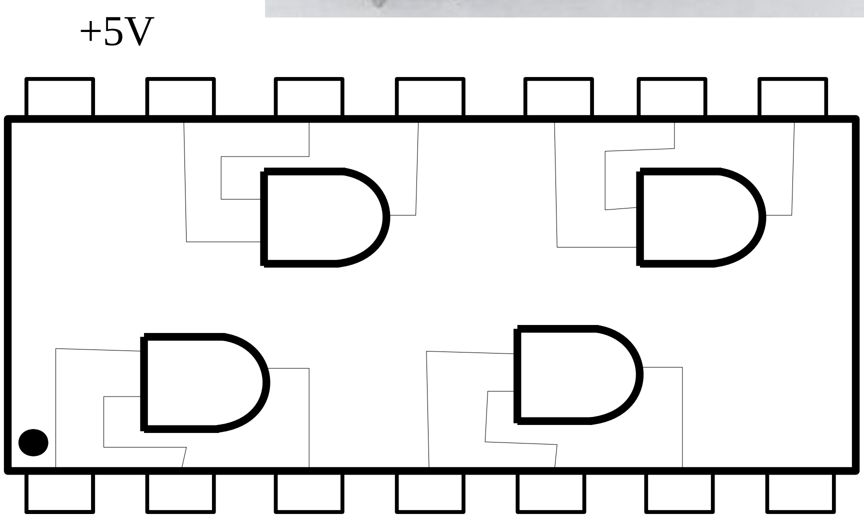


•If you start doing this seriously in industry I suggest you take a look at some of this software..

Field programmable gate array

•Final tip, I have introduced you to logic chips that look like this, they have contain 8 gates.





Field programmable gate array

•Although these chips are still used (especially in glue logic).
•It is very common today to use something called an (Field programmable gate array) FPGA.

Cost <10 pounds

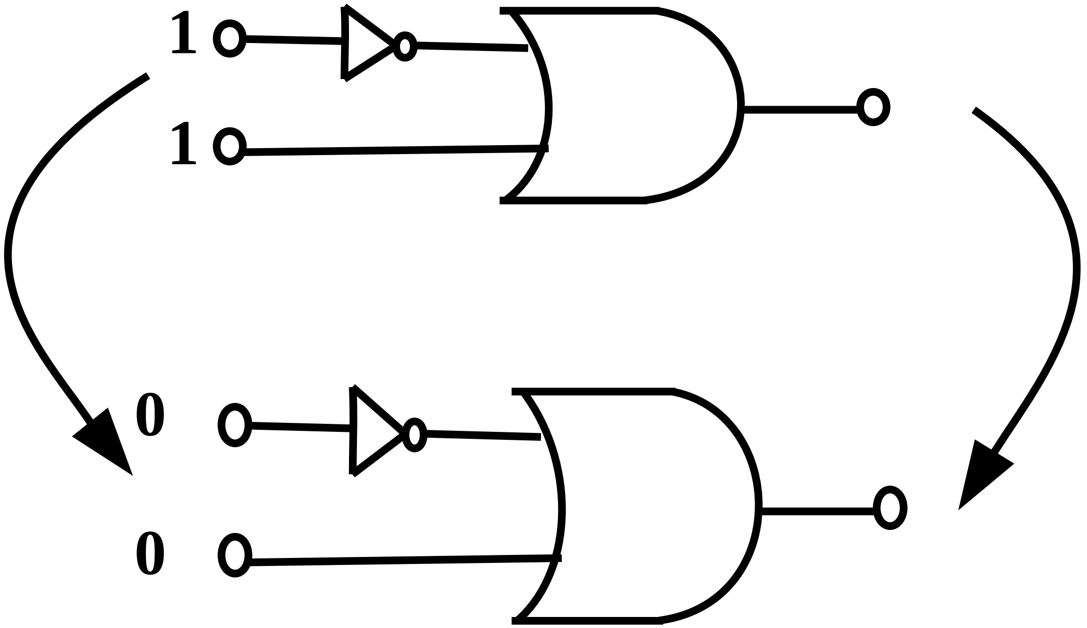
•This single chip can contain **half a million gates** and the connections between each gate are programmable.

•This means you can actually 'rewire' your circuit with a computer while your machine is running.

•You will meet theses in industry.

Last tip on digital electrons **race times**

•Think about this circuit, if you changed the inputs from 11 to 00 the outputs would not change.



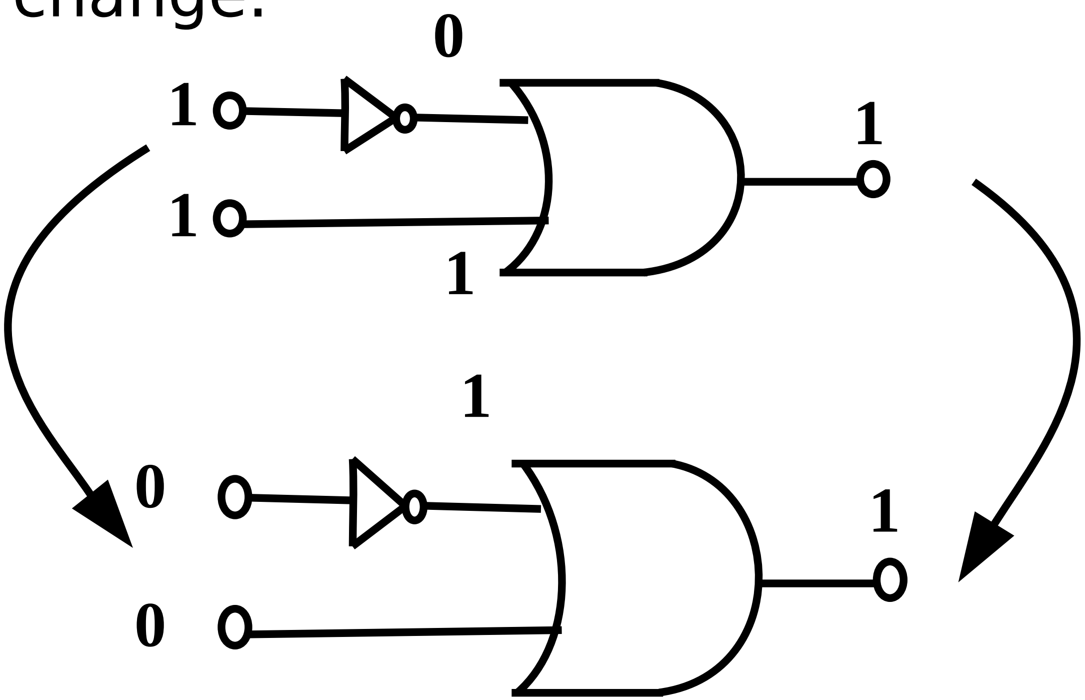

Slooby from Chicago

•Let's draw this out in detail

Last tip on digital electrons **race times**

Slooby from Chicago

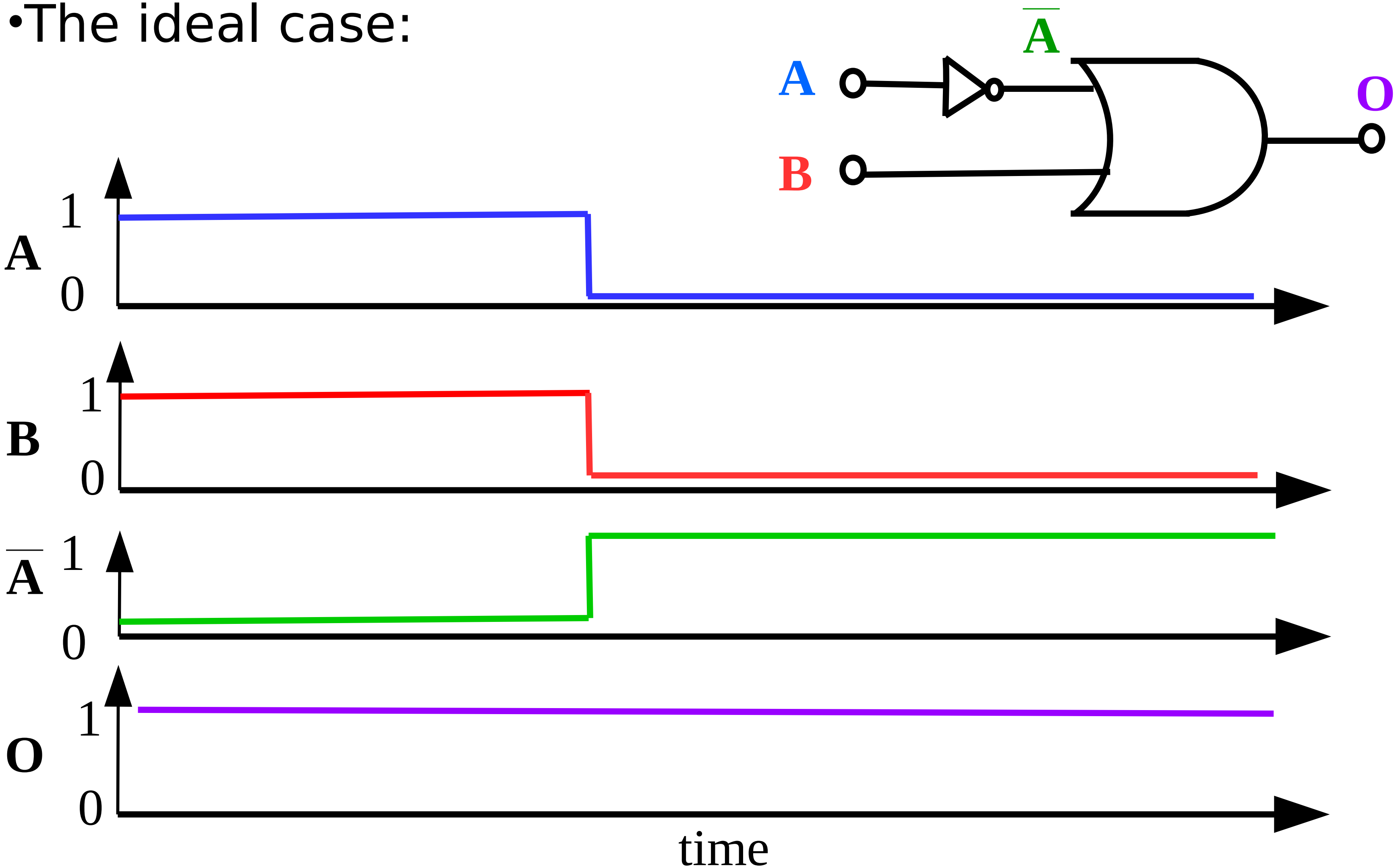
•Think about this circuit, if you changed the inputs from 11 to 00 the outputs would not change.



•Let's draw this out in detail

Last tip on digital electrons **race times**

- The ideal case:

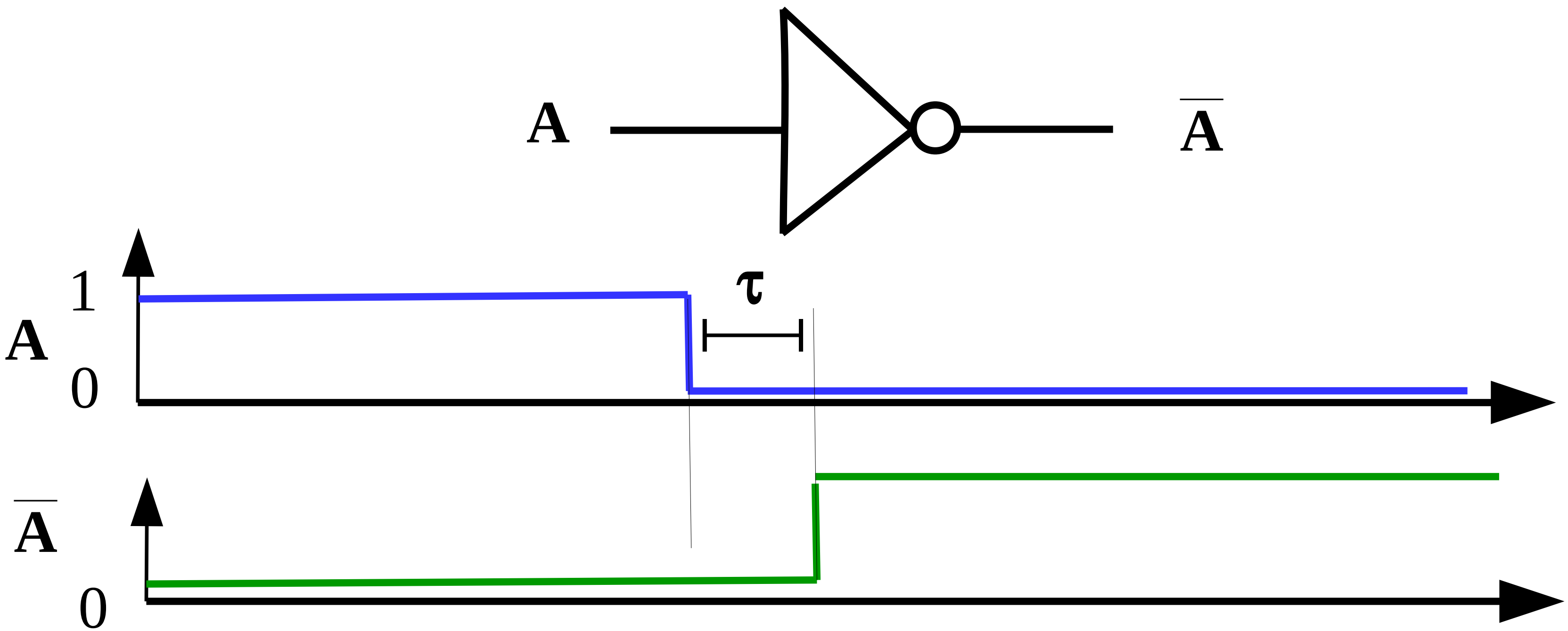


Slooby from Chicago

However, you should know that....

•All gates have a 'turn on' and 'turn off' time.

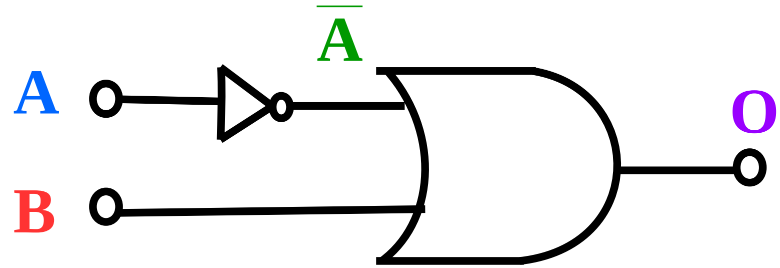
•This is in effect a time it takes the gate to react to an input - τ.

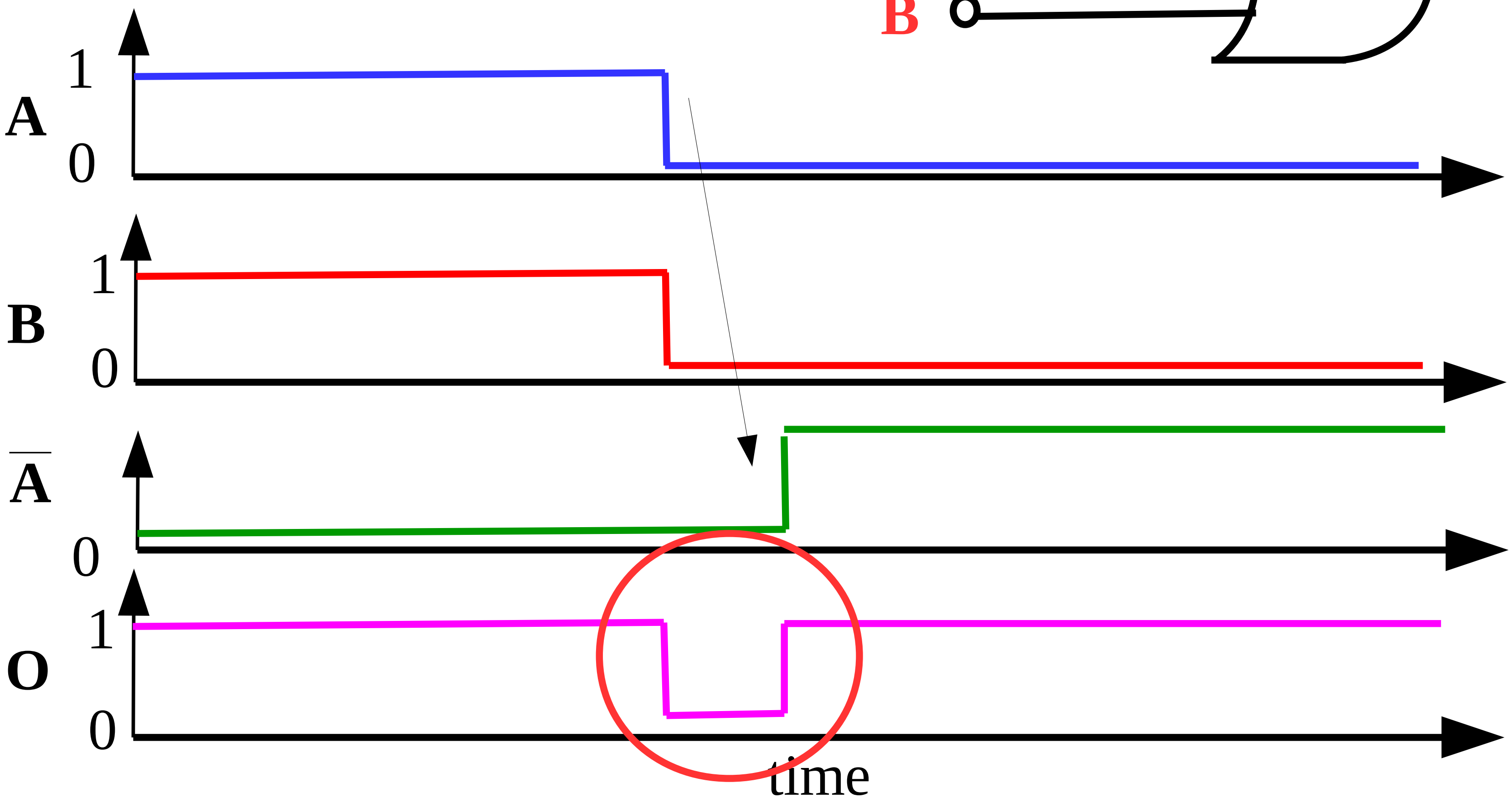


Slooby from Chicago

Last tip on digital electrons **race times**

- However every gate takes a finite time to respond (micro seconds).





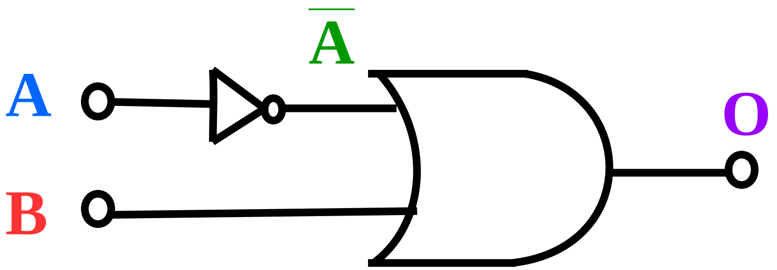
Last tip on digital electrons **race times**

•Gates take microseconds to respond.

•This means your output can be wrong for up to a microsecond.

•The more gates you have the longer theses effects will last for.

•Often glitches don't matter, but if you are designing sensitive (fast) circuits, they can be really really important.



Summary of today's lecture

- Recap of last lecture

- Using JK flip flops to run motors

- Digital design
    - What is digital design?
    - How to do it.
    - Where you might meat digital design
    - FPGAs

- Race times