Workshop 11: Computers, logic, memory 2

1 Getting Started

Make sure you have everything you need to complete this lab:

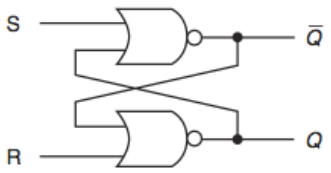
- Arduino for power supply
- breadboard
- black, red and blue jumper cable
- Multimeter with a red and black test lead cable
- NOR 74LS02N IC chip
- 4-bit Register
- four LEDs

2 Sequential circuits

A sequential circuit defines its output as a function of both its current inputs and its previous inputs. Therefore, **the output depends on past inputs**. To remember **previous inputs, sequential circuits must have some sort of storage element**. We typically refer to this storage element as a **flip-flop**.

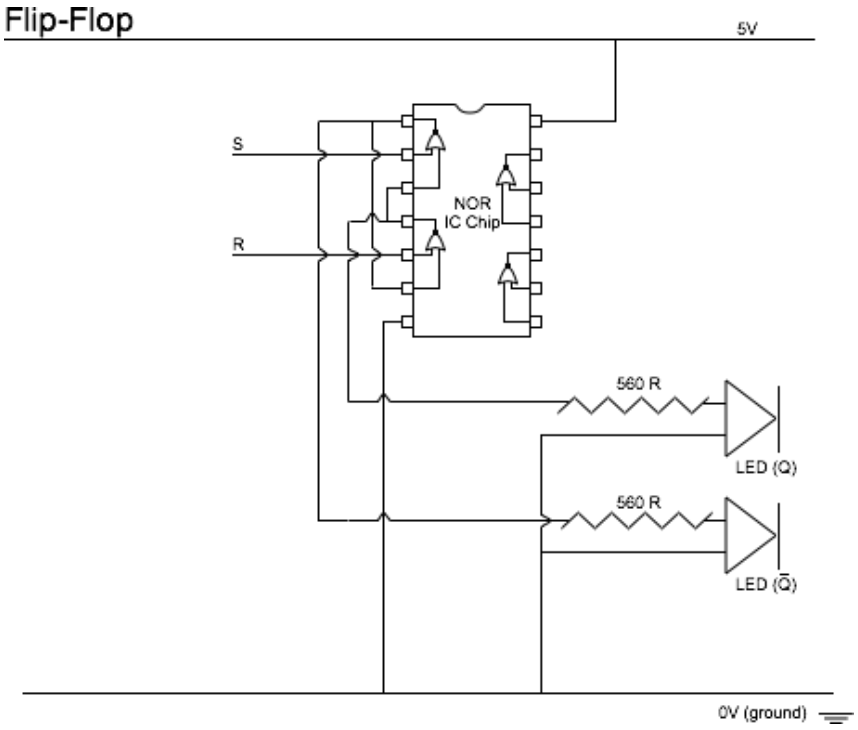
2.1 Flip-Flop

A level-triggered circuit is allowed to **change state whenever the clock signal** (it can be a simple switch button, pressed in specified intervals) is either **high or low**. In this exercise we are going to **build the most basic memory-unit called an SR flip-flop**. SR stands for set/reset. In the image below you will see how the flip-flop looks like, using NOR logic gates for the connection.



S	R	Q (t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	undefined

1. Build a flip-flop, using NOR IC chip. Use the circuit below to connect the elements. Because if it's a SR Flip-Flop, **set** refers to **light up the LED connected to Q**, **reset means turn off Q and light up Q'**.



2. Once you have finished the flip-flop, **fill out the table below**. What are the values for the next state, Q(t+1), record the changes from the experiment? What happens if both S and R are set to 1 at the same time? Why is that?

S	R	Present State Q(t)	Next State Q(t+1)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

3 Binary numbers

As you may already know the two most important rices in computer science are binary (base 2), and hexadecimal (base 16). The binary system uses only the digits 0 and 1. You will need some basic knowledge how to convert decimal numbers to binary and vice versa to solve the following exercises.

3.1 Convert decimal number to binary

You can use either of two methods for radix conversion: the subtraction method or the division remainder method. The image below show the division method:

$$
\begin{array}{r|l}
2 & 190 \quad = \quad 0 \\
2 & 95 \quad = \quad 1 \\
2 & 47 \quad = \quad 1 \\
2 & 23 \quad = \quad 1 \\
2 & 11 \quad = \quad 1 \\
2 & 5 \quad = \quad 1 \\
2 & 2 \quad = \quad 0 \\
2 & 1 \quad = \quad 1 \\
 & 0
\end{array}
$$

1. First we take the number that we wish to convert and divide it by the radix in which we want to express our result (in this case it's 2).
2. Record the quotient and the remainder.
3. Continue in this way until the quotient is zero.

To practise the binary conversion, fill out this table by converting the decimal numbers to binary:

Decimal	4-bit Binary
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

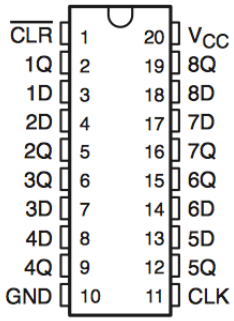
3.2 Convert binary numbers to decimal

Any integer quantity can be represented exactly using any base (or radix). The equation below shows how to convert a binary number to decimal:

$$
\begin{array}{rccccccccccc}
 & 1 \times 2^4 & + & 1 \times 2^3 & + & 0 \times 2^2 & + & 0 \times 2^1 & + & 1 \times 2^0 & & \\
= & 16 & + & 8 & + & 0 & + & 0 & + & 1 & = & 25
\end{array}
$$

4 4-bit register/binary counter

In this exercise we are going to use a 4-bit register IC chip, which includes 8 flip-flops (so it can represent binary numbers up to 8-bit), almost the same flip-flops that you have built previously.



In the connection diagram above, V_{cc} stands for the voltage supply, and **GND** is the ground. **CLR** means clearing the memory and **CLK** is for the clock input, that we talked about before. **D** ports are the **inputs** of the flip-flops and **Q is the output**.

- Use this IC chip to represent 4-bit binary numbers using LEDs. You should connect the LEDs to the output using a 560R resistor.
- Try to represent binary numbers like, **1010, 0001, 0101, 0110** or any other, using the LEDs light (1 means on, 0 means off).
- Once you have successfully represented the numbers, you should try to **count from 0000 to 1111** with the 4-bit register. For this you will need to use the **CLK** input.

HINT: To count in binary, it's a good idea to use a switch button connected to the **CLK** input. So pressing the button should increase the number by 1.