# Workshop 10: Computers, logic, memory 1

## 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
- two BC337 transistors
- NAND 74LS00 IC chip
- three 560R resistors
- LED

## 2 Logical operators

Logical operators are mainly used to **control program flow**. Usually, you will find them as part of an *if*, a *while*, or some other *control statement*. The concept of logical operators is simple. They allow a program to **make a decision based on multiple conditions**. Each operand is considered a condition that can be evaluated to a true or false value. Logical operators can be **represented by truth-tables**.

### 2.1 NOT

The negation NEGx of a formula x is **true when x is false** and false otherwise. Fill out the corresponding truth-table representation for the NOT logical operators:

| x | NEGx |
|---|---|
| 0 | |
| 1 | |

### 2.2 AND

The conjunction x AND y is **true** if and only if **both x and y are true**. Fill out the corresponding truth-table:

| x | y | x AND y |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

### 2.3 OR

The disjunction x OR y is **true if and only if at least one of x, y is true** (i.e., either or both). Fill out the corresponding truth table representation:

| x | y | x OR y |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

### 2.4 NAND

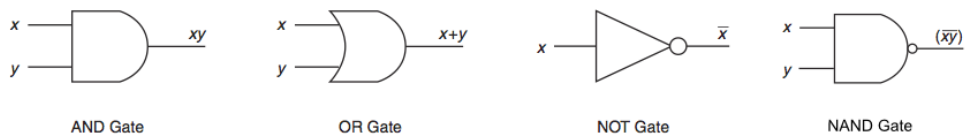
The NAND operator gives **false if and only if x and y are true**, and gives true otherwise, **equivalent to NOT AND**. Fill out the truth-table below:

| x | y | x NAND y |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

Of course there are more logical operators, such as XOR or NOR, but these four will be sufficient to understand the elementary logic.

## 3 Logic gates with LEDs

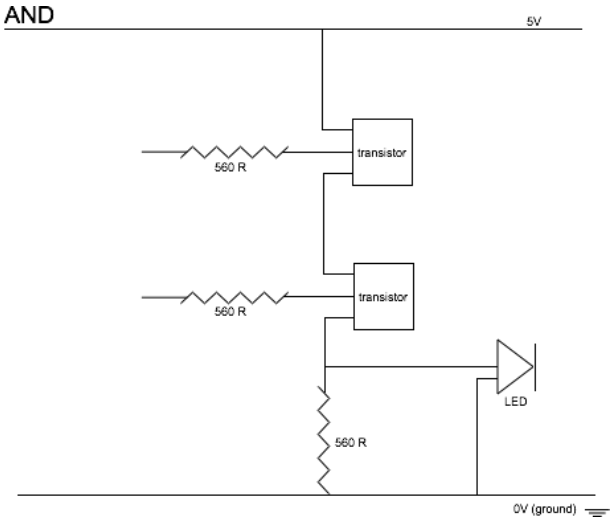
The logical operators AND, OR, and NOT that we have discussed have been represented thus far in an abstract sense using truth tables and Boolean expressions. The actual physical components, or **digital circuits, such as those that perform arithmetic operations or make choices in a computer, are constructed from** a number of primitive elements called **gates**. The image below shows the graphical representation of the gate that corresponds to each operator:



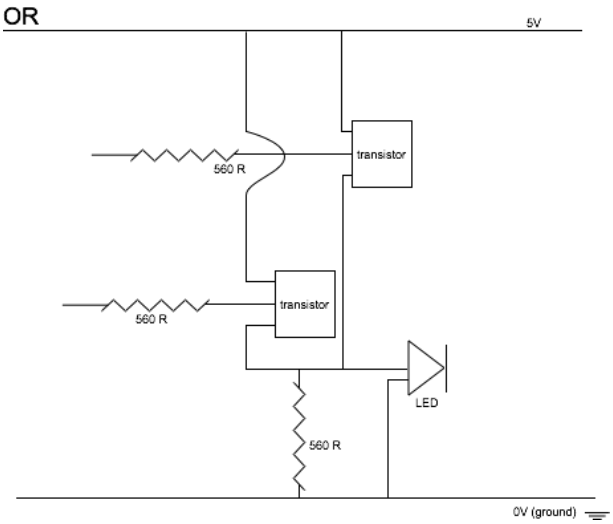

Once you understand the concept of logic gates, let's **build the AND, OR and NAND logical gates**, using resistors and transistors. For this you will need to build the logical gates using the circuits on the images below. **IMPORTANT! Not just build the gates, but also think about why it works the way**

**it does, and also calculate the output value of for all interpretation using truth-tables.** All the resistors, and transistors should be FULLY connected through the voltage supply and the ground.

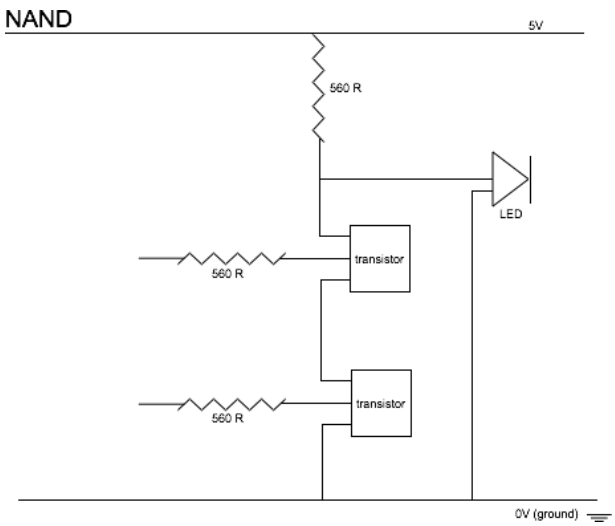
1. Build the **AND** logic gate by using **three 560R resistors and two BC337 transistors**. Follow the AND circuit below in order to build the logic gate. When you connecting the LED, do not forget that the **slightly longer lead is the positive**.




2. Build the **OR** logic gate by using the same amount of transistors and resistors.



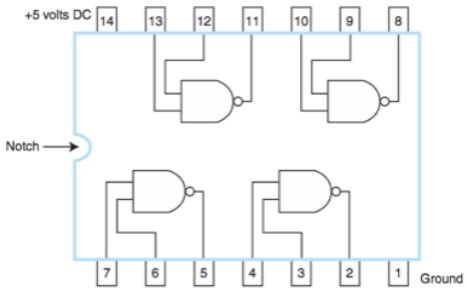

3. Build the **NAND** logic gate! What kind of modification do you need to perform from the **AND** logic gate? Why is the LED turns off, when both input values are 1?

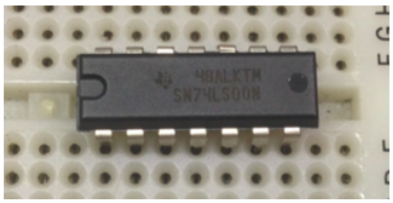



**HINT:** The logical value 1 refers to HIGH power, so when the input is connected to the voltage supply, its value is 1. The logical value 0 refers to LOW power, so when the input is connected to the ground or not connected at all. **Ask for help if you have any trouble!**

## 4 Logic gates with NAND IC chip

Building logic gates manually is a good way to learn how they work, but fortunately we have a pre-build IC (Integrated Circuit) NAND chip. This chip contains **4 NAND logic gates with two input values and an output value, it also has a 5V DC input and a ground input**. The NAND gate is commonly referred to as a universal gate, because **any electronic circuit can be constructed using only NAND gates**.



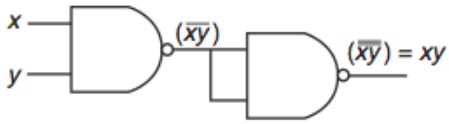



There are two reasons to investigate using only NAND gates:

1. NAND gates are **cheaper to build** than the other gates.
2. complex integrated circuits are often much **easier to build using the same building block** (i.e., several NAND gates) rather than a collection of the basic building blocks

Your task is to **build the AND and OR logic gates**, using NAND IC. For this you need to understand the truth tables of all the three logic operators.

1. First, build the **AND** logic gate, the image below should help how to connect the NAND circuits together:




AND Gate

2. Now try to build the **OR** logic gate by yourself! How the circuit should look like? **HINT:** Use the truth-table to manipulate the values and create a circuit just like the image above. It is a good idea to think about how many NAND logic gate is needed to get an OR logic gate! **Ask for help if you have any trouble!**