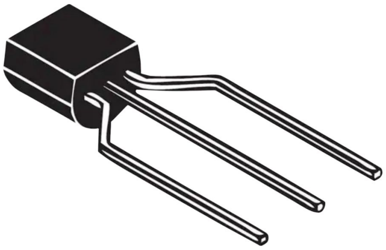
PHYSICAL LIMITATION AND SYNCHRONIZATION

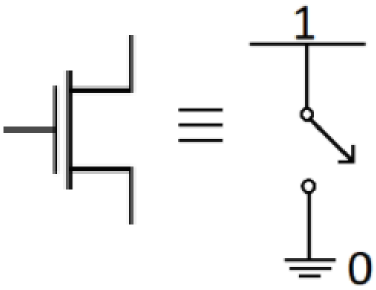
Understanding the electrical circuit behind logic gates and purpose of synchronization.



So far we know how logic gates work and how we can use them in computer science, but we still have to learn how they are built in order to fully understand the physical limits that come with electronics. Here we remove another layer of abstraction to see what electronic components are used to represent logic gates.

TRANSISTORS

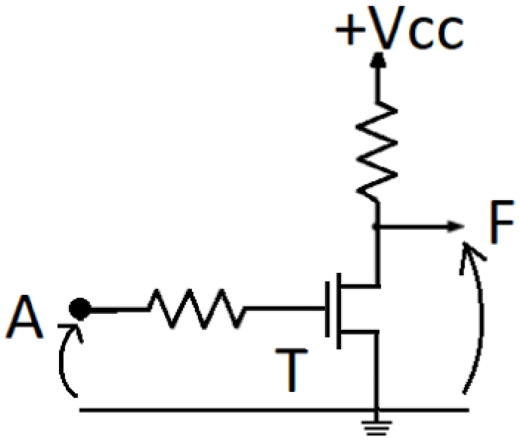
Logic gates are composed of **transistors**, which depending on the voltage applied to them, can act as breaking points or wire (or in general, as a switch). A transistor schematic looks as follows (image below), with 3 wires connected to it, each with a different function. The corresponding letters mean the following: G- Gate, D-Drain and S-Source. In other classes, you will learn that we distinguish 2 different types of transistors, but the idea behind them remains the same. The transistor works similarly to an electric lock, with G (gate) being the signal that releases it. When there is voltage applied to G, the “lock” is open and the current may flow from D to S and the transistor acts as a wire. In case of no signal applied to Gate, the “lock” remains closed and there is no current flow to the collector, which means that the transistor acts as an open circuit.



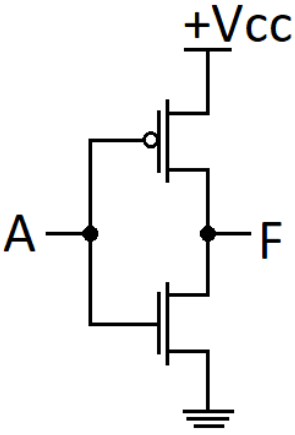
In future courses you may learn much more about actual operating of different transistors and that this point of view is a simplification, but at this moment it is more than enough for you to understand how it is used inside the logic gates.

INVERTER / NOT GATE

The NOT gate is the most basic both in operation and construction, which makes it a perfect example to start with. Below you can find a very simplified example of a NOT Gate schematic:

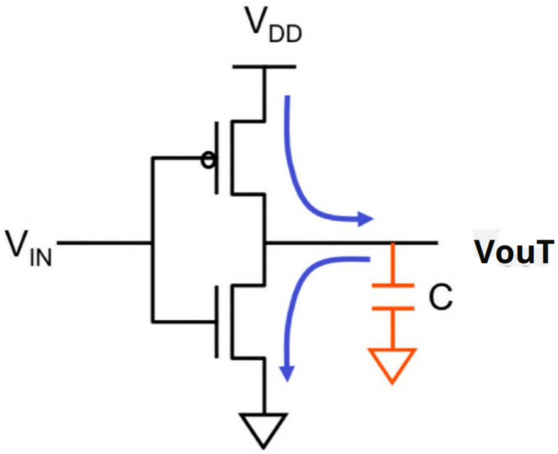


From the electric point of view, the logical 0 is equal to approximately 0 V, and logical 1 is equal to whatever voltage is originally applied to the output. In this case, logical 1 is equal to Vcc. If we track down the signal, we can see that with A = 1 (meaning that there is voltage applied to the Base), the Transistor will act similarly to a short wire, causing direct connection between voltage source and ground, which means that there will be no voltage drop on output (F =0). On the other hand, with no voltage on the A, the transistor will not allow the current to flow, leading to difference in potentials, though F being equal to 1. There are also other schematics with different levels of precision/different construction, that when tracked down, lead to the very same outcome, for example:



PHYSICAL LIMITATION OF A TRANSISTOR

So far we focused on an ideal model of a transistor, which works without delay or voltage loss. In the real world, the delay between state changes of a logic gate may vary between a couple nanoseconds down to a few picoseconds. The change of the state may be represented as a capacitor on the output, where the charge is kept for a brief moment after state change of transistor:



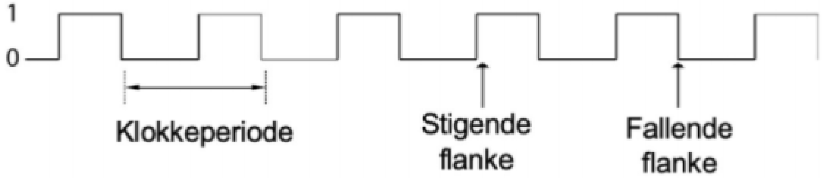
It does not seem like a lot, but it may cause problems in very large circuits. Imagine a circuit with millions of logic gates (as there are in CPU) working together, with each one of them getting the signal with a nanosecond delay time, and each logic gate transferring signals as fast as it can. In a couple minutes, there may be a delay of a couple seconds between the gates at the “beginning” of the circuit and the “end”. And now imagine, that most of the gates get their input signals from other trees of gates with different delay time at the end. It will lead to complete disaster and definitely a crash on your last DS-1 boss. That is why all computers use synchronization.

SYNCHRONIZATION

Imagine a big factory with a giant clock in the middle, where every employee is responsible to make one small adjustment to the product they are making. Every minute, the clock makes a loud sound that every worker can hear, and precisely on that note, they pass a component that they worked on to the following person responsible for making the next adjustment. Thanks to that clock, everyone works in a perfect symphony, without the stress of not finishing the adjustment on time or getting it too early. This is an example of synchronized work, which is used in electronic circuits a lot.

CLOCK SIGNAL

Clock Signal is a signal that generates jumps in constant time intervals. This signal is composed of two transfer states, usually described as rising edge and falling edge, which are used in the process of synchronization. Exemplary waveform of a clock is as follows:



In synchronized operations, every component passes the results on one of the states and waits on the other. Thanks to the “breaks” there is no delay between components and we can be sure that every signal is sent on the proper time.

FREQUENCY

Frequency is a parameter used for describing clock changes speed. It is described as the inverse of time interval length, where time interval is the time necessary for one complete clock cycle to finish (one jump and one drop) . For example, If the time interval between clock “ticks'' is 1 ms then the frequency is equal

$$f = 1/1ms = 1/0.001s = 1kHz$$

The greater the frequency of the clock signal, the faster the computer may work. Nevertheless, we cannot increase the clock frequency to our will, as the breaks eventually may come as too short for the components to properly operate. Maximum frequency that a computer may generate is limited by the physical capabilities of the machine and those electrical components.

EXAMPLES OF SYNCHRONIZATION

An Example of usage of Synchronization is the Multiple Bit Full Adder. Thanks to the clock applied, all logic gates transfer the proper outcome in proper order and time.