Basic Logic Gates vs. Universal Logic Gates

Teachers and textbooks often introduce logic gates with AND, OR, and NOT as the base gates and XOR, NAND, NOR, and XNOR as being built from these three gates [In fact, that is exactly what we did in the logic gate activity you completed in this unit]. We do this because AND, OR, and NOT mimic the three base operators in Boolean algebra used in programming, and many of the basic computer operations use these gates.

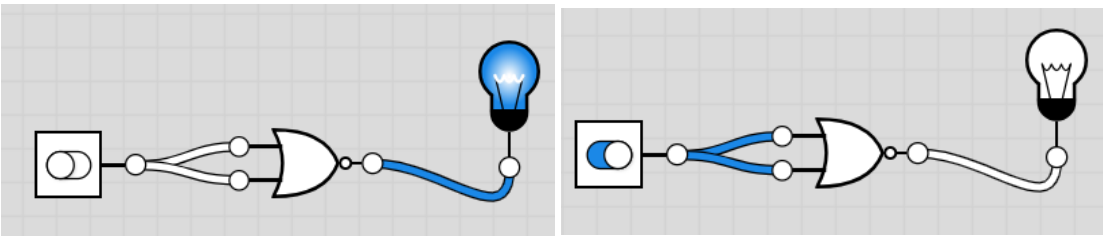
But the reality is that this isn't accurate at the hardware level. NAND and NOR are considered the "universal gates" and the other five can all be built from combinations of these. It turns out that these two gates are both inexpensive and easy to fabricate. Thus, in many system, the other logic gates are built from these universal gates.

This handout shows you how these other five gates can be made from only these two universal gates.

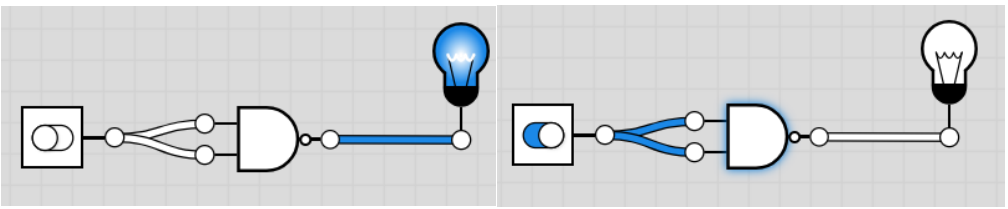
The NOT Gate

We start with this one because it is a major building block of the ones to come.

The NOT gate is the most unlike others because it only takes one input while all the others take two. So how do you create a gate that only takes/uses one input out of gates that take two inputs? The solution is to have that single value connected to both of the inputs of the NOR gate. If the input is 0 then the gate sees two 0s and outputs a 1. IF the input is 1 then the gate sees two 1s and outputs a 0.

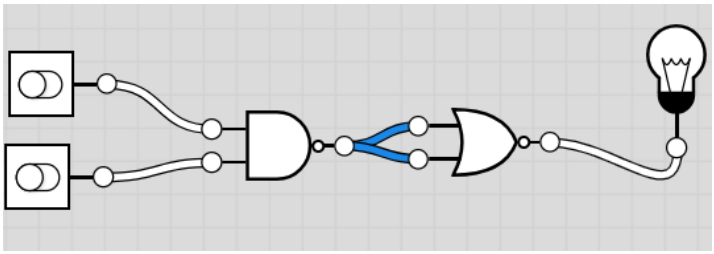


Notice that since NOR and NAND behave the same when both inputs are the same, we can will get the same effect by using a NAND gate instead. Either one can serve as a NOT.



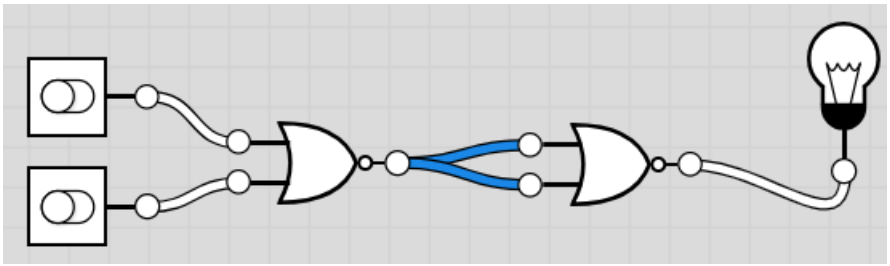
The AND Gate

Recall that the NAND gate was presented earlier as the AND gate followed by a NOT gate. But the flip is also true. An AND gate can be created from a NAND gate followed by a NOT. Since our NOT is just a specialized use of the NOR gate, we can make that work in this context too.



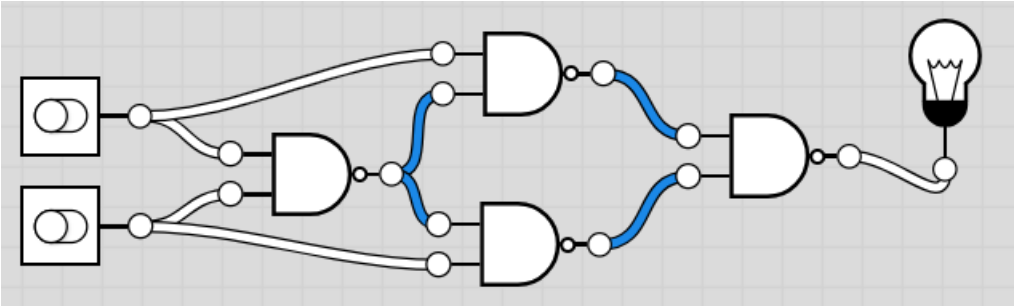
The OR Gate

Similarly, an OR gate can be created from two NOR gates.



The XOR Gate

The XOR gate is the hardest to understand. It is composed of four NAND gates.



The XNOR Gate

Since an XNOR is the “opposite” of the XOR so, like the others in this configuration, we just need to add a singular NAND or NOR gate after the XOR to negate it.