# Logic Gates

Logic gates are an important part of computing. Computers use logic gates with binary data to model data. The main logic gates are as follows:

* NOT gates: NOT gates invert values between true or false; NOT true is false and NOT false is true.
* AND gates: AND gates are true only when both inputs are true; false AND false is false, false AND true is false, true AND false is false, and true AND true is true.
* OR gates: OR gates are true when one or both of their inputs are true: false OR false is false, false OR true is true, true OR false is true, and true OR true is false.

There are, however, ways to expand on these basic gates:

* XOR gates (exclusive OR gates): XOR gates are true when one but not both of their inputs are true; false XOR false is false, false XOR true is true, true XOR false is false, and true XOR true is true.
* NAND gates (NOT AND gates): NAND gates are true when at least one of their inputs is false; false NAND false is true, false NAND true is true, true NAND false is true, and true NAND true is false.
* NOR gates (NOT OR gates): NOR gates are true when both their inputs are false; false NOR false is true, false NOR true is false, true NOR false is false, and true NOR true is false.

One of the reasons why logic gates are important is that they can be used with conditional statements to control how a program acts. Consider the following pseudocode:

X = false

PastX = false

Y = false

PastY = false

Repeat forever:

X = input(is x true or false now?)

Y = input(is y true or false now?)

If x is different from PastX:

X has changed.

If y is different from PastY:

Y has changed.

If x is different from PastX AND y is different from PastY:

Both x and y have changed.

PastX = x

PastY = y

In the example above, we want to know if x and y have both changed. We use x and PastX to keep track of changes in x, and we use y and PastY to keep track of changes in y. But how do we know if both of them have changed? The easiest way to is to use an AND gate- if x has changed AND y has changed, then x and y have both changed.

These gates are essential to how computers work. Consider the following example:

P = false

Q = false

Repeat forever:

P = input(is p true?)

Q = P OR Q

In this example, p and q both start out as being false. However, once p becomes true, q becomes true as well. If p becomes false later, however, q is still true. This is an example of how logic gates can be used to remember a value- q remembers whether or not p has ever been true. If we expand the example to add the option to reset Q, we end up with the ability to remember to store a value. This is part of how computers remember information by using logic gates and binary.