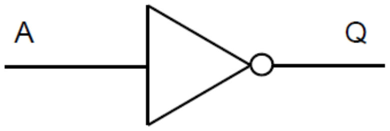
Logic Gates

The term logic gate actually gives a clue as to the function of these devices in an electronic circuit. ‘Logic’ implies some sort of rational thought process taking place and a ‘gate’ in everyday language allows something through when it is opened. A Logic Gate in an electronic sense makes a ‘logical’ decision based upon a set of rules, and if the appropriate conditions are met then the gate is opened and an output signal is produced.

The basic building blocks of a computer are called logical gates or just gates. Gates are basic circuits that have at least one (and usually more) input and exactly one output. Input and output values are the logical values true and false. In computer architecture it is common to use 0 for false and 1 for true. The value of the output depends only on the current value of the inputs.

The NOT gate (or inverter)

This is the simplest form of logic gate and has only 1 input and 1 output. So how can it make a decision if it only has 1 input ? Simply the purpose of this gate is to invert the input signal so if a Logic 0 is at the input, the output will be at Logic 1 and vice versa. The symbol for a NOT gate is as follows.



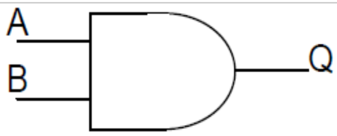
The output of a logic gate can also be summarised in the form of a table, called a ‘Truth Table’. The truth table for a NOT gate is the simplest of all Truth Tables and is shown below.

Input	Output
A	Q
0	1
1	0

The Boolean expression $Q=\overline{A}$

The AND Gate

The AND gate implements the AND function. With the gate shown to the left, both inputs must have logic 1 signals applied to them in order for the output to be a logic 1. With either input at logic 0, the output will be held to logic 0.

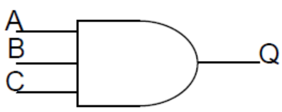


The truth table for the 2 input AND gate is shown below.

Inputs		Output
B	A	Q
0	0	0
0	1	0
1	0	0
1	1	1

The Boolean expression **Q= A.B**

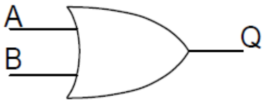
Now we will consider a 3 input AND gate. The symbol is:



Inputs			Output
C	B	A	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

The OR gate

The *OR gate* gets its name from the fact that it behaves after the fashion of the logical inclusive "or." The output is "true" if either or both of the inputs are "true." If both inputs are "false," then the output is "false." In other words, for the output to be 1, at least input one OR two must be 1.

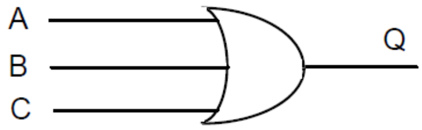


The truth table for the 2 input OR gate is shown below.

Inputs		Output
B	A	Q
0	0	0
0	1	1
1	0	1
1	1	1

The Boolean expression **Q= A+B**

Now we will consider a 3 input OR gate. The symbol is:



Inputs			Output
C	B	A	Q
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Combined gates

The NAND gate

The *nand*-gate is an *and*-gate with an inverter on the output. So instead of drawing several gates like this

The nand-gate, like the and-gate can take an arbitrary number of inputs. The truth table for the nand-gate is like the one for the and-gate, except that all output values have been inverted.

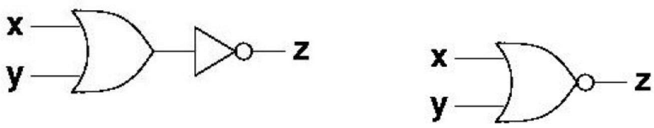
The nand-gate, like the and-gate can take an arbitrary number of inputs. The truth table for the nand-gate is like the one for the and-gate, except that all output values have been inverted:

Inputs		Output
B	A	Q
0	0	1
0	1	1
1	0	1
1	1	0

The Boolean expression $Q = \overline{A.B}$

The nor-gate

The nor-gate is an or-gate with an inverter on the output. So instead of drawing several gates like this. We draw a single or-gate with a little ring on the output like this:



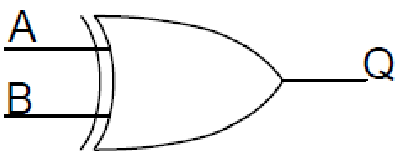
The truth table

Inputs		Output
B	A	Q
0	0	1
0	1	0
1	0	0
1	1	0

The Boolean expression $Q=\overline{A+B}$

The XOR gate

The exclusive-or-gate is similar to an or-gate. It can have an arbitrary number of inputs, and its output value is 1 if and only if exactly one input is 1 (and thus the others 0). Otherwise, the output is 0. We draw an exclusive-or-gate like this:



The truth table for the 2 input XOR gate is shown below

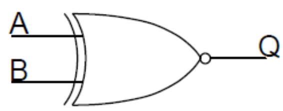
Inputs		Output
B	A	Q
0	0	0
0	1	1
1	0	1
1	1	0

The Boolean expression **$Q= A\oplus B$**

$$Q= A.\overline{B} + \overline{A}.B$$

The XNOR gate

The XNOR gate has 2 inputs and is the inverted form of the EXOR gate. The symbol for a 2 input XNOR gate is as follows



The truth table for the 2 input XNOR gate is shown below

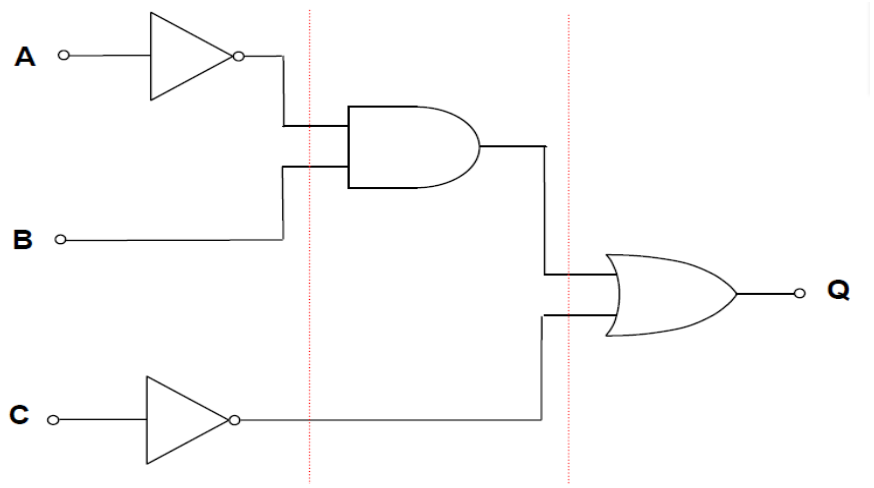
Inputs		Output
B	A	Q
0	0	1
0	1	0
1	0	0
1	1	1

The Boolean expression $Q=\overline{A\oplus B}$

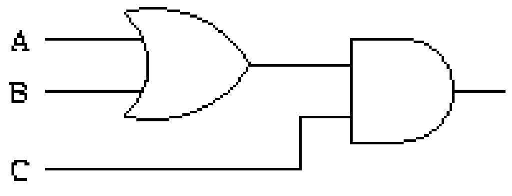
$$Q=\overline{A}\cdot\overline{B}\oplus A\cdot B$$

Example

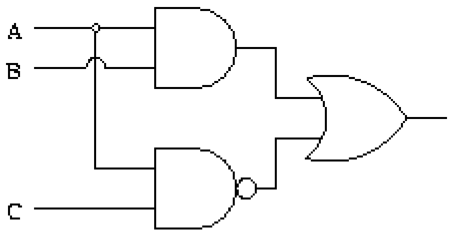
Draw the Boolean expression $Q=\overline{A}\cdot B+\overline{C}$



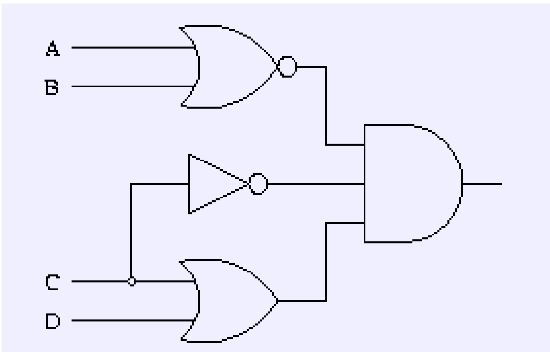
Example: Draw a logic circuit for (A + B)C



Example: Draw a logic circuit for AB + AC.



Example: Draw a logic circuit for (A + B)(C + D)C



Example: Find the truth table for ***(A+BC)***

A	*B*	*C*	*BC*	*A+BC*
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Example: Find a truth table for *PT(P+Z)*.

P	*T*	*Z*	*T*	*PT*	*P+Z*	*PT(P+Z)*
0	0	0	1	0	0	0
0	0	1	1	0	1	0
0	1	0	0	0	0	0
0	1	1	0	0	1	0
1	0	0	1	1	1	1
1	0	1	1	1	1	1
1	1	0	0	0	1	0
1	1	1	0	0	1	0