GCSE Electronics: Component 1

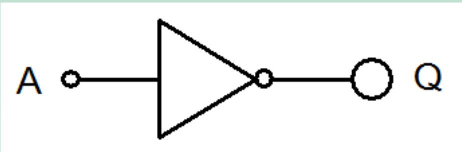
Unit 6: Combinational logic systems (page 1)

Logic systems work on digital signals. A digital signal can only have **two** values, usually zero and the maximum of the power supply. **Changes between these two occur instantaneously.**

Logic gates

There are five basic logic gates:

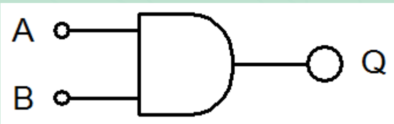
1. NOT gate



Input	Output
A	**Q**
0	1
1	0

Signal out of the gate is the opposite of the signal in, i.e. it **inverts** the input signal.

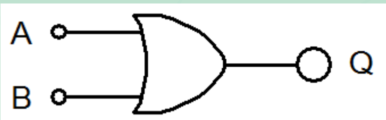
2. AND gate



Input		Output
B	**A**	**Q**
0	0	0
0	1	0
1	0	0
1	1	1

The output Q is only at a logic 1 when input A AND input B are at a logic 1.

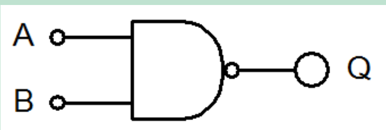
3. OR gate



Input		Output
B	**A**	**Q**
0	0	0
0	1	1
1	0	1
1	1	1

The output Q is only at a logic 1 when input A OR input B are at a logic 1.

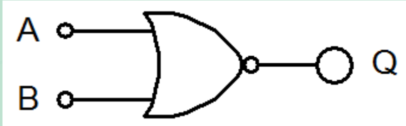
4. NAND gate



Input		Output
B	**A**	**Q**
0	0	1
0	1	1
1	0	1
1	1	0

The output Q is the exact opposite of the AND gate

5. NOR gate

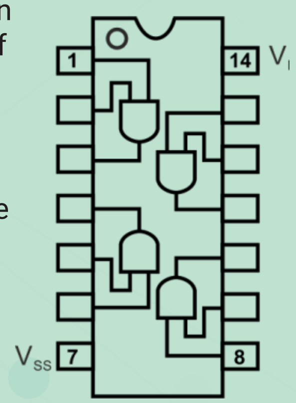


Input		Output
B	**A**	**Q**
0	0	1
0	1	0
1	0	0
1	1	0

The output Q is the exact opposite to an OR gate.

Practical logic gates

Logic gates are available within an **integrated circuit (IC) -** a set of electronic circuits built on the same wafer of semiconductor material. These logic ICs are usually supplied in plastic dual in line (DIL) packages containing several logic gates of the same type.



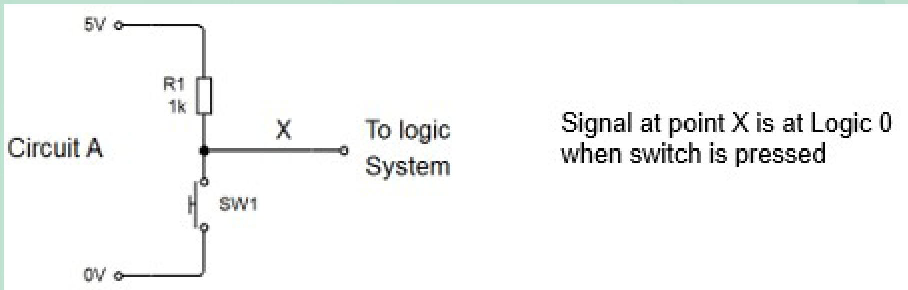
There are two common types of package available, known as the TTL or 7400 series and the CMOS or 4000 series. The data sheet for a logic gate package includes the pin out diagram showing how the pins connect to the logic gates inside it.

Using the right pin out is important, as incorrect connections can damage the whole package.

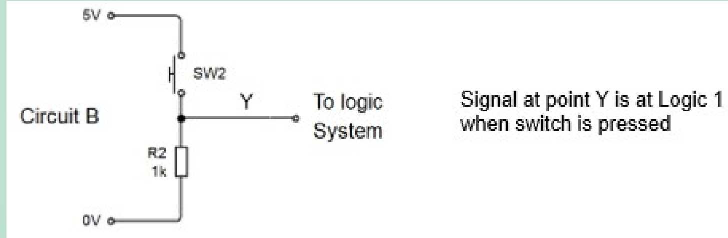
Pull up/pull down resistors

We have to provide the logic gate with a suitable input subsystem to provide the correct logic levels. The input to a logic gate can come from a number of different sources but for the basics we will use mechanical switches.

Switches have to be used along with a series resistor as part of a voltage divider circuit. The orientation of the resistor and switch in the voltage divider circuit change the logic level produced when the switch is pressed. Two input sub-system circuits using a push to make switch are shown below.



The resistor in circuit A is called a **pull-up** resistor.



The resistor in circuit B is called a **pull-down** resistor.

Converting a truth table into a logic diagram

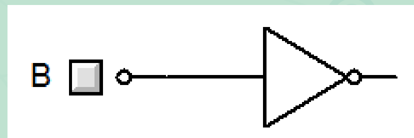
Constructing a logic circuit diagram from a truth table is best done by looking at an example.

Convert the following truth table to a logic circuit:

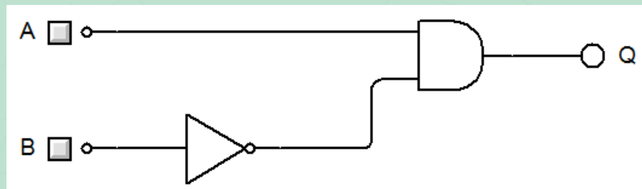
Input		Output
B	**A**	**Q**
0	0	0
0	1	1
1	0	0
1	1	0

First identify all the combinations of the inputs that produce a logic 1 at the output. In this case, it only occurs once, when input A is on and input B is not on.

The NOT gate is used to invert the B input, as shown below:



The output of this NOT gate is then connected to the AND gate, with input **A** to provide the full solution:



Quick rule

In any 2-input logic system, for every row of the truth table where the output is logic 1, it can be written in terms of the following input conditions: **A, NOT A, B** or **NOT B**, depending whether there is a 0 or a 1 in that input cell. The two inputs are linked with an AND.

GCSE Electronics: Component 1

Unit 6: Combinational logic systems (page 2)

wjec cbac eduqas

Boolean notation

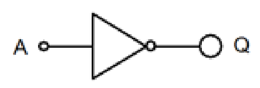
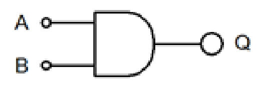
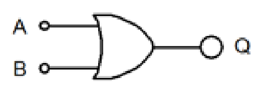
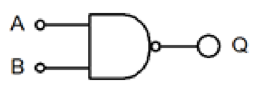
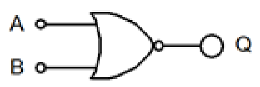
This a shorthand way of writing down the function of logic gates, using a special type of algebra called **Boolean algebra**. We will start by looking at the five basic gates we have introduced previously.

There are three basic things to remember

1. A dot or period "." between two input labels is read as "AND".

2. A plus "+" between two input labels is read as "OR".

3. A bar "‾" over the top of an input label is read as "NOT".

Gate	Symbol	Boolean Notation	
NOT		$Q = \bar{A}$	(read as Q = NOT A)
AND		$Q = A.B$	(read as Q = A AND B)
OR		$Q = A+B$	(read as Q = A OR B)
NAND		$Q = \overline{A.B}$	(read as Q = A NAND B)
NOR		$Q = \overline{A+B}$	(read as Q = A NOR B)

In addition to the five Boolean notations shown above, each line of a truth table for which the output is a "1" can also be written in Boolean notation:

Input		Output
B	A	Q
0	0	1
0	1	0
1	0	0
1	1	1

Q= ($\bar{A}$ AND $\bar{B}$) OR (A AND B)

Q= $(\bar{A}.\bar{B})+(A.B)$

Two special identities

1. The Boolean expression $\overline{(A.B)}$ is the same as $\bar{A}+\bar{B}$. We can check this by looking at the truth table below:

B	A	$\bar{B}$	$\bar{A}$	$\overline{A.B}$	$\bar{A}+\bar{B}$
0	0	1	1	1	1
0	1	1	0	1	1
1	0	0	1	1	1
1	1	0	0	0	0

2. The Boolean expression $\overline{A+B}$ is the same as $\bar{A}.\bar{B}$. We can check this by looking at the truth table below:

B	A	$\bar{B}$	$\bar{A}$	$\overline{A+B}$	$\bar{A}.\bar{B}$
0	0	1	1	1	1
0	1	1	0	0	0
1	0	0	1	0	0
1	1	0	0	0	0

NAND gate implementation

Logic systems designs often require a number of different types of logic gate (e.g. NOT, AND and OR) in order to fulfil the function required.

In some of the designs, we may need three different types of logic gate in the final design. There could be as many as six identical logic gates in an IC package, of which we may only use one. This is wasteful in terms of both unused devices and in the space needed on circuit boards.

The inverted gates, NAND and NOR, are special because the function of all other gates can be made from various combinations of NAND or NOR gates.

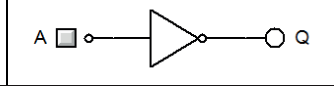
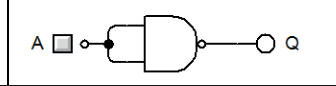
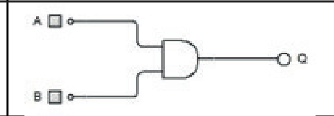
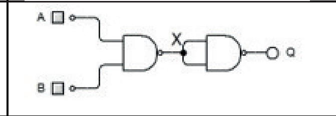
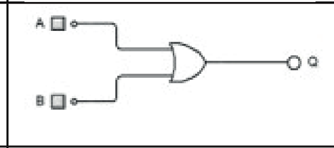
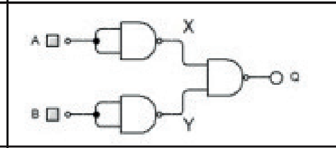
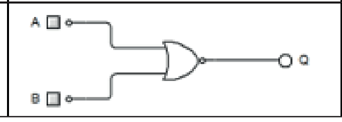
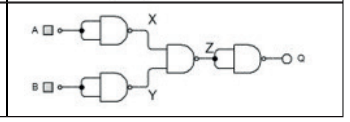
By using just one type of logic gate we may be able to reduce the number of types of logic gate required to make any particular design.

This has a number of advantages:

i. There will be less confusion about which type of gate goes where in the circuit as they are all the same.

ii. There will be no need to keep stocks of all the different types of logic gate, therefore saving money.

iii. Larger quantities of a single type of gate can be purchased, which makes cost lower.

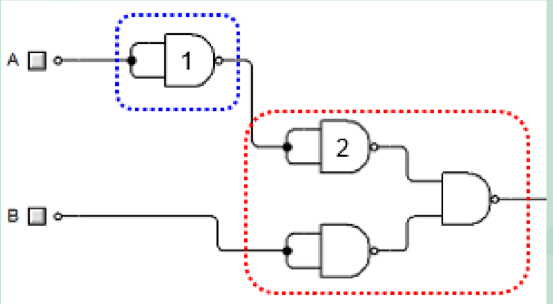
Gate	Normal Configuration	NAND gate equivalent
NOT		
AND		
OR		
NOR		

Converting a logic circuit to NAND gates only

1. Redraw the NAND equivalent circuits of the gates shown above, where possible retain the position of these gates so that you can identify the connections afterwards.

2. Connect the equivalent circuits together.

3. Examine the completed circuit for possible **double inversions.**



4. Logic gates 1 and 2 above are redundant since the logic signal emerging from gate 2 is the same as A. The circuit becomes: