isaac computer science

Logic gates

A logic gate is a fundamental component of a digital circuit. Each gate has one or more inputs and produces a single output that depends on the input(s). Multiple logic gates can be combined to make complex circuits to carry out processing.

Each logic gate has a symbol, that is used in circuit diagrams, and a unique truth table that shows how the output changes with different inputs.

You can think of the inputs to logic gates as 1's and 0's. These values are determined by the voltage that flows around the system where a high voltage is evaluated as 1 (True) and a low voltage is evaluated as 0 (False).

GCSE **Algebraic symbols**

There are two standards for algebraic symbols: those used in maths and those used in electronics. If you create an account for Isaac Computer Science, you will be asked to select the notation you would prefer to see.

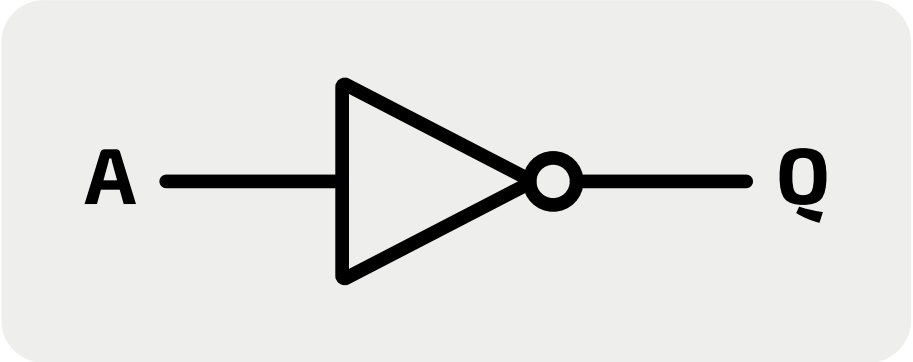
GCSE

NOT gate

The **NOT** gate takes a single input value and flips it to its binary opposite.

- If the input is 1 (True), then the output will be 0 (False).
- If the input is 0 (False), then the output will be 1 (True).

This is the symbol for a **NOT** gate:



NOT gate

The truth table of a NOT gate looks like this:

Input A	Output Q
0	1
1	0

The Boolean expression for this gate can be written as $Q = \neg A$, where ¬ represents NOT.

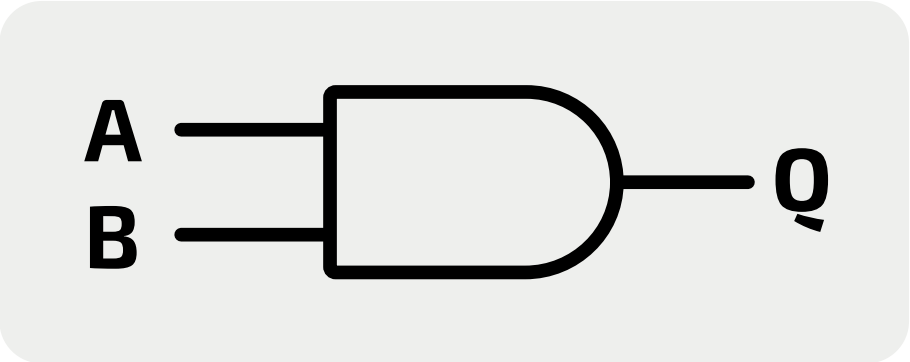
GCSE

AND gate

The **AND** gate takes two input values and produces an output of 1 (True) if **both** inputs are 1 (True).

This is exactly what you would expect — after all, if you asked for fish **and** chips and did not receive both, you would be disappointed!

This is the symbol for an AND gate:



AND gate

Here is the truth table for the AND gate:

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

The Boolean expression for this gate can be written as $Q = A \wedge B$, where $\wedge$ represents AND.

GCSE

OR gate

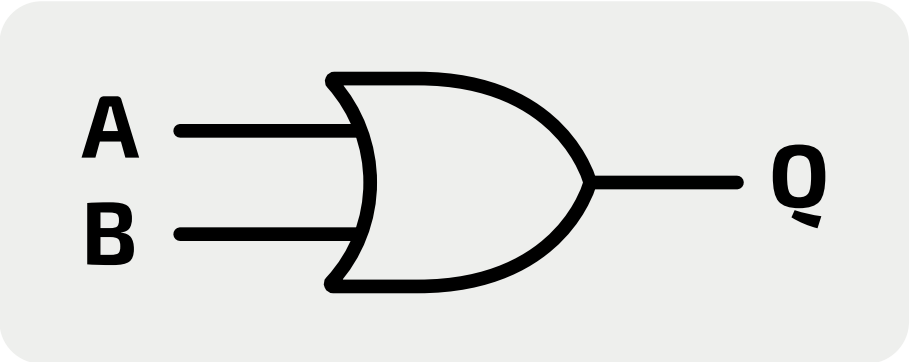
The **OR** gate takes two input values and produces an output of 1 (True) if **either** one of the inputs are 1 (True).

In English, the word 'or' has two possible meanings. For example:

- You might get a treat if you help out at home **or** do well at school. So if you help out at home and do well at school, you should still get a treat.
- As the treat, you might be offered a trip to the cinema **or** to the zoo. But you won't get to go to both the zoo and the cinema as the treat — the options are exclusive.

The OR gate implements the logic of the first example. There is another gate — XOR — that implements the second set of logic.

This is the symbol for an **OR** gate:



OR gate

Here is the truth table for the **OR** gate:

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

The Boolean expression for this gate can be written as $Q = A \vee B$, where $\vee$ represents OR.

GCSE

XOR gate

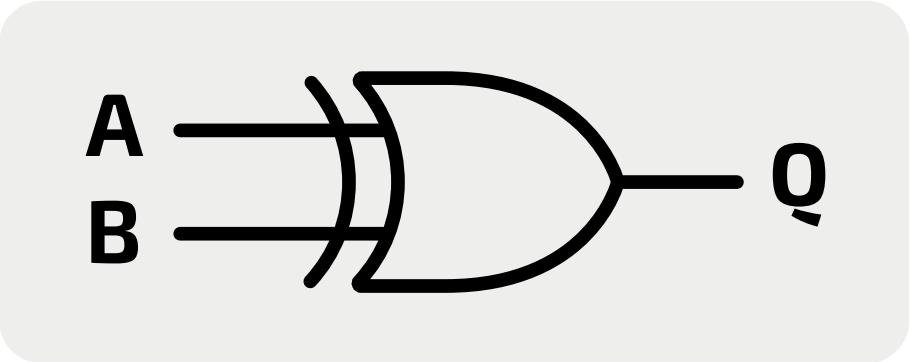
The **XOR** gate takes two input values and produces an output of 1 (True) if **either but not both** inputs are 1 (True). This is the 'exclusive OR' and is also sometimes called **EOR**.

In English, the word 'or' has two possible meanings. For example:

- You might get a treat if you help out at home **or** do well at school. So if you help out at home and do well at school, you should still get a treat.
- As the treat, you might be offered a trip to the cinema **or** to the zoo. But you won't get to go to both the zoo and the cinema as the treat — the options are exclusive.

The XOR gate implements the logic of the second example. There is another gate — OR — that implements the first set of logic.

Here is the XOR gate symbol:



XOR gate

Here is the truth table for the XOR gate:

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

The Boolean expression for this gate can be written as $Q = A \veebar B$, where $\veebar$ represents XOR.

GCSE

Multi-input logic gates

Logic gates can be designed to take more than one input. In this instance, a single gate will implement the logic of a set of the same gates in series. For example, a three-input AND gate will implement the logic of two AND gates in series, as shown in **Figure 1** below.

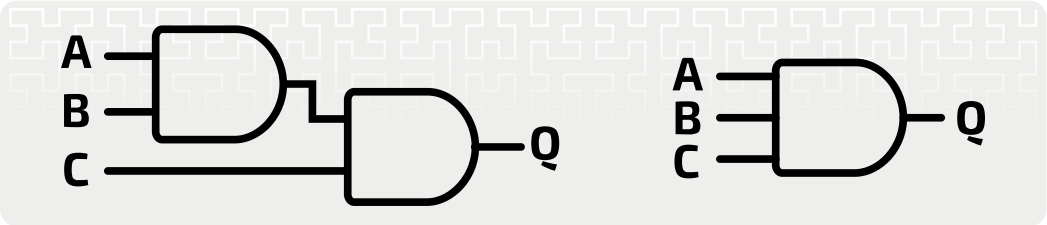


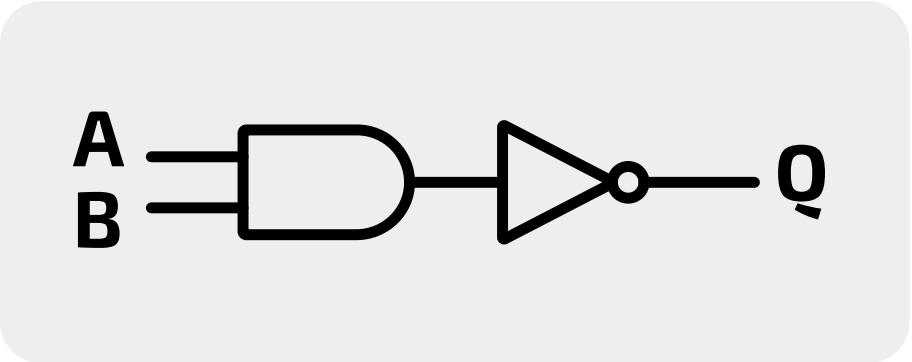
Figure 1: The equivalent logic of a three-input AND gate

Logic gate	Two inputs	Three inputs	Four inputs
AND gate	$A \wedge B$	$A \wedge B \wedge C$	$A \wedge B \wedge C \wedge D$
OR gate	$A \vee B$	$A \vee B \vee C$	$A \vee B \vee C \vee D$
NAND gate	$\neg(A \wedge B)$	$\neg(A \wedge B \wedge C)$	$\neg(A \wedge B \wedge C \wedge D)$
NOR gate	$\neg(A \vee B)$	$\neg(A \vee B \vee C)$	$\neg(A \vee B \vee C \vee D)$
XOR gate	$A \veebar B$	$A \veebar B \veebar C$	$A \veebar B \veebar C \veebar D$

A Level

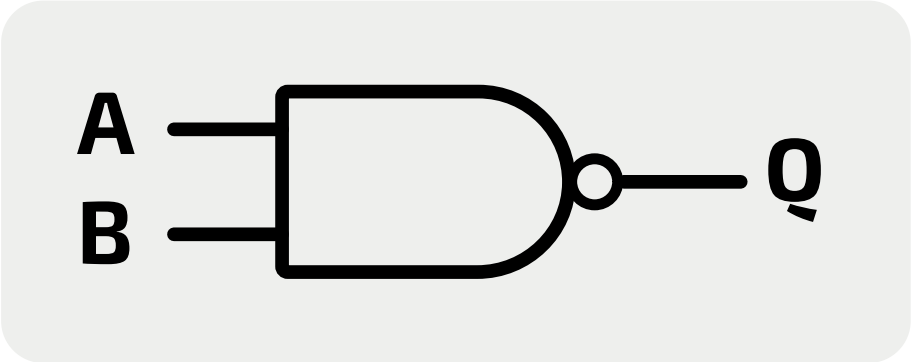
NAND gate

The **NAND** gate combines the logic of an AND gate and a NOT gate; it produces the inverse of the output of an AND gate.



The **logic** of a NAND gate

The symbol for a NAND gate is a combination of the symbols for the two gates:



NAND gate

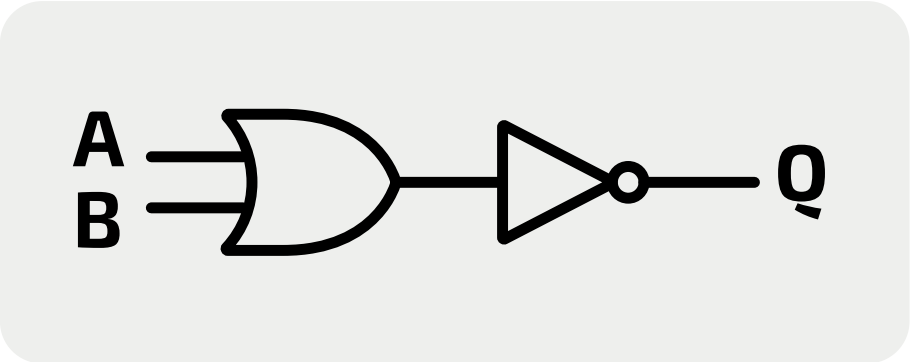
Here is the truth table for the NAND gate:

INPUT A	INPUT B	OUTPUT Q
0	0	1
0	1	1
1	0	1
1	1	0

A Level

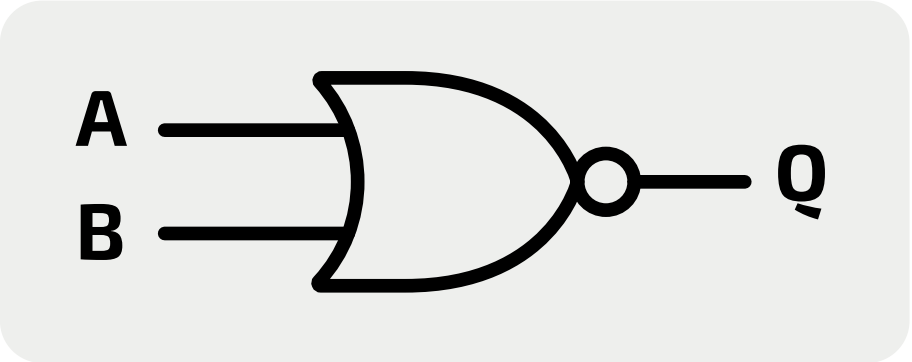
NOR gate

The **NOR** gate combines the logic of an OR gate and a NOT gate; it produces the inverse of the output of an OR gate.



The **logic** of a NOR gate

The symbol for a NOR gate is a combination of the symbols for the two gates:



NOR gate

Here is the truth table for the NOR gate:

INPUT A	INPUT B	OUTPUT Q
0	0	1
0	1	0
1	0	0
1	1	0