Draw a circuit diagram

Circuit diagrams are used to design and document logic circuits. Diagrams use standard symbols to represent logic gates; you can see an example of each logic gate here.

You need to be able to draw a circuit diagram by hand, but there are plenty of online tools that you can use to practise your skills. These often have the advantage that you can also run them as a simulation, thus testing the logic of your design. You may be asked to produce a circuit diagram from a Boolean expression or at an advanced level, you may be given a problem statement or a truth table to work from. If you are **not** given the expression, you will need to work it out before you can draw the diagram.

You can read about how to create a Boolean expression from a truth table here and at an advanced level, how to create a Boolean expression from a problem statement here.

GCSE Operator precedence – logic gates

You have already learned that in a Boolean expression operators have a given order of precedence. This is:

1. Brackets
2. NOT
3. AND
4. OR

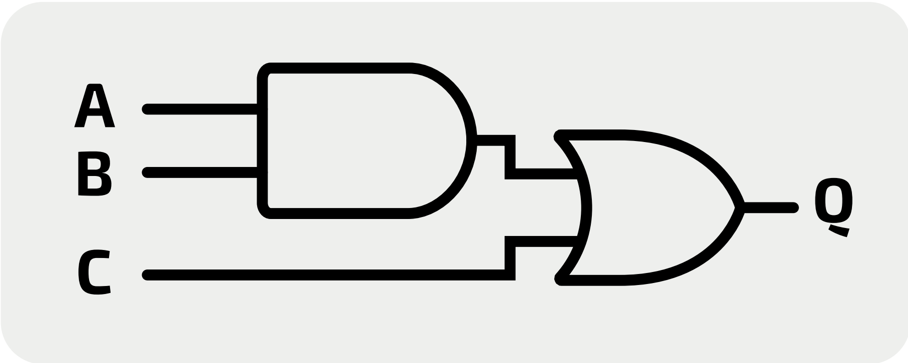
When working with circuits at an advanced level, you will have gates that represent additional logic operations. This is because gates can combine more than one basic logic operation. For example, a NAND gate combines the logic of AND and NOT. Thus, you must consider the precedence of these additional operators as shown below. Where more than one operator is listed on the same line (e.g. AND and NAND), they have equal precedence and can be executed in any order.

1. Brackets
2. NOT
3. AND, NAND
4. OR, NOR, XOR

GCSE **Draw a circuit diagram from a simple expression**

Video explainer demonstrating how to draw a circuit diagram from a simple expression

Embedded YouTube video: https://www.youtube.com/watch?v=WK2F71V-mMw.
Below, you can see the circuit diagram produced in the explainer video.

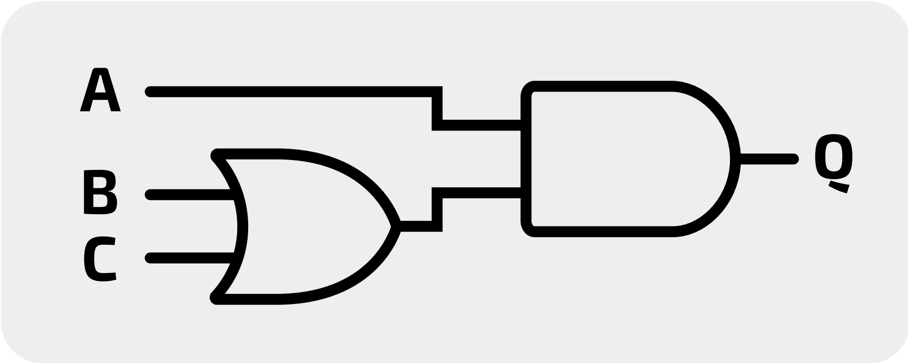


Here is the associated truth table for that circuit diagram.

A	B	C	$A \wedge B$	$(A \wedge B) \vee C$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

You can see an alternative circuit diagram below, based on the idea that order of precedence doesn't matter when you are creating a circuit diagram.

Look at the image below. This is how the circuit diagram would look if you had chosen to deal with the OR operator from the expression first.



Here is the associated truth table for that particular circuit diagram.

A	B	C	$B \vee C$	$A \wedge (B \vee C)$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

One thing to note here is that the two expressions are **not** equivalent. You can observe this by looking at the truth table for each expression and you will notice that the outputs are not the same.

Create a circuit diagram to represent the Boolean expression $Q = (A \wedge B) \vee (C \wedge D)$.

Click a button to show the answer

What is your level of confidence that your own answer is correct?

Low Medium High

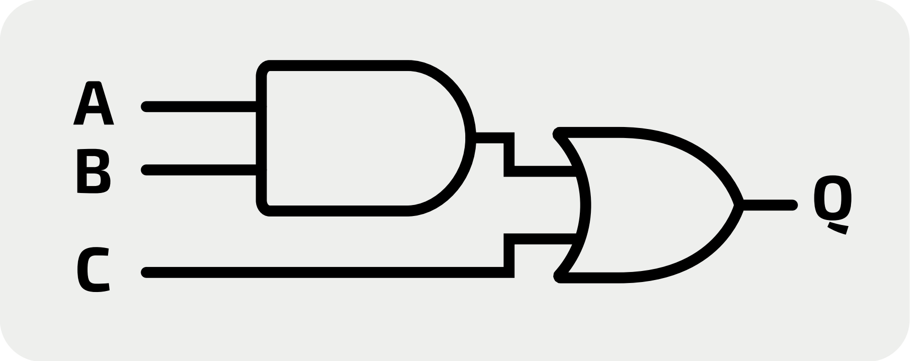
A Level

Draw a circuit diagram from a simple expression

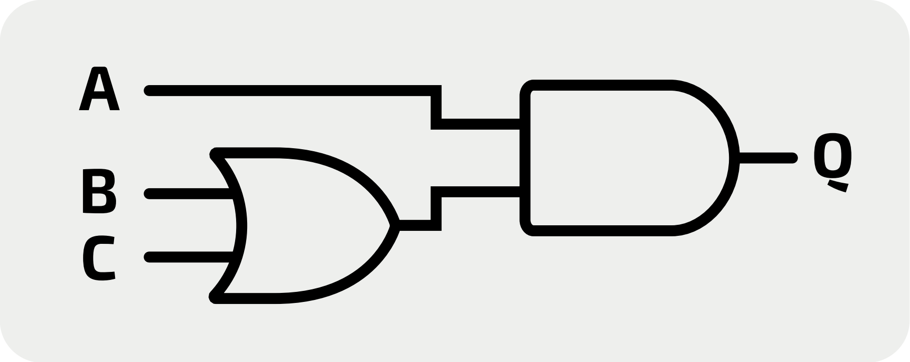
Think about how you would draw a logic gate diagram for the Boolean expression:

$Q = A \wedge B \vee C$

You might draw the diagram in one of the two following ways:



Option A



Option B

The correct diagram is option A. This is because the order of operator precedence specifies that the AND operator takes precedence over the OR operator, so this operation must be carried out first.

It is very easy to make a mistake with precedence when you are given an expression to work with. It is very useful to add brackets to remind you of the order in which the expression must be evaluated. In this case, you could write:

$Q = (A \wedge B) \vee C.$

One thing to note here is that the two options are **not** equivalent. You can prove this by producing a truth table for each option and you will observe that the output is not the same.

Produce the truth table for option A.

Click a button to show the answer

What is your level of confidence that your own answer is correct?

Low Medium High

Produce the truth table for option B.

Click a button to show the answer

What is your level of confidence that your own answer is correct?

Low Medium High

A Level

Draw a circuit diagram from a complex expression

Now work through a more complicated example. Here is the expression:

$Q = (A \lor B) \land (\neg C \land A)$

Remember the order of operations:

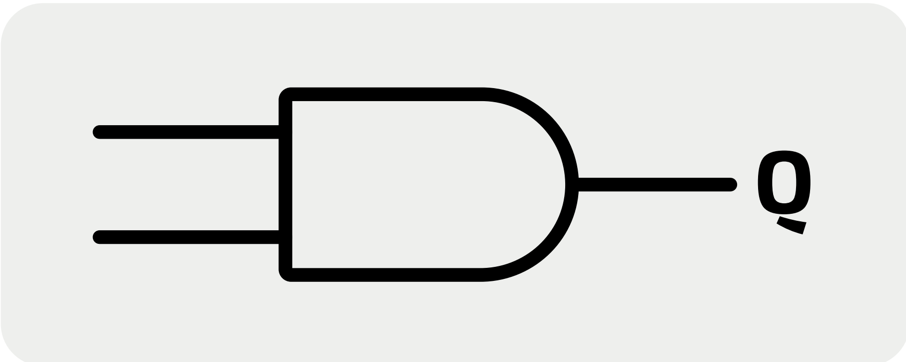
1. Brackets
2. NOT
3. AND, NAND
4. OR, NOR, XOR

It is conventional for a circuit diagram to have the inputs on the left and the output(s) on the right. So start by writing the letters A, B, and C on the left of a piece of paper and the letter Q on the right.

Before you start adding gates to the diagram, it is useful to identify how many of each type of gate you will need. The expression you are working with has two **AND** operators, one **OR** operator, and one **NOT** operator. Your final diagram will need to include these four gates.

Step 1: The final AND gate

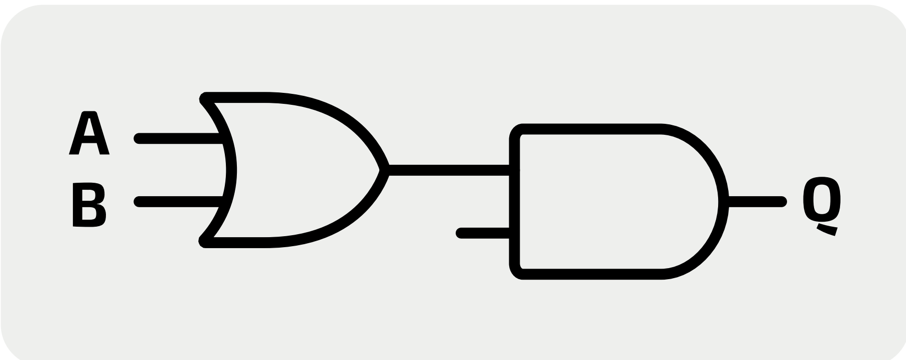
In the given expression, the **last** operator that will be evaluated is between the two bracketed statements: it is an **AND** gate. Start your diagram by adding an AND gate immediately before the output Q and connect the output from the gate to Q.



Step 1: AND gate

Step 2: The OR gate

Now look at either one of the statements that feeds into the **AND** gate. On the left-hand side you can find $(A \lor B)$. This needs an **OR** gate, so add it to the diagram to the left of the **AND** gate that you added previously and attach the output from this **OR** gate to one of the inputs of the **AND** gate. It doesn't matter which of the input connectors you pick.

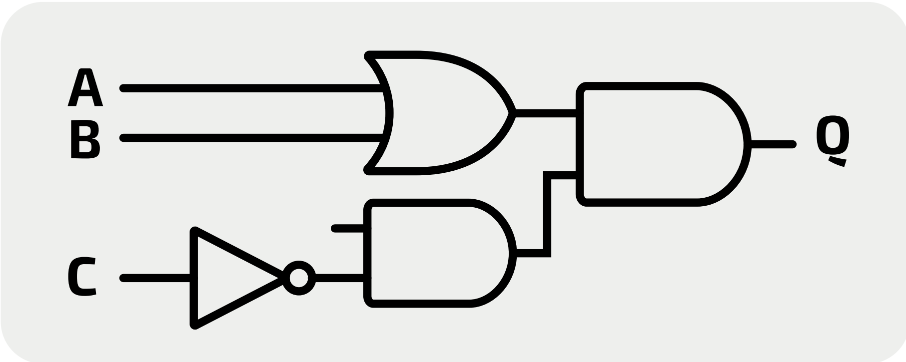


Step 2: OR gate

Step 3: The NOT gate and second AND gate

Now take the statement on the right-hand side. This is $(\neg C \wedge A)$. **NOT** has a higher priority than **AND**, so you know that $\neg C$ must happen first.

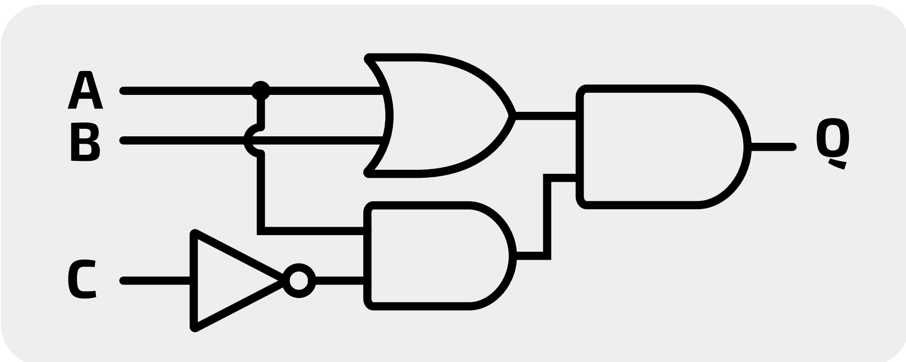
Draw a **NOT** gate with its input from C and its output leading into one of the inputs of the second **AND** gate, which you must position between the **NOT** gate and the final **AND** gate.



Step 3: NOT gate and AND gate

Step 4: The final connection

Now you must connect the other input of the second AND gate to A. If several gates receive the same input, like A in this example, you normally produce a branch in the line using a dot as a connector.



Step 4: The finished circuit

In a busy circuit, you may get overlapping lines. Note the curve in the line from the A branch that overlaps the line from B to the **OR** gate; this curve indicates that the lines are **not** connected.

A Level

Draw a circuit diagram from a truth table

If you are asked to produce a circuit diagram from a truth table, you will need to form the Boolean expression first. Consider the following truth table:

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

Here, you can see that Q is 1 when:

A is 1 **AND** (B is 1 **OR** C is 1)

Therefore, the expression for the truth table is:

$Q = A \wedge (B \vee C)$

With this information, you can draw the circuit diagram (shown in **Figure 3** below) using the techniques described previously.

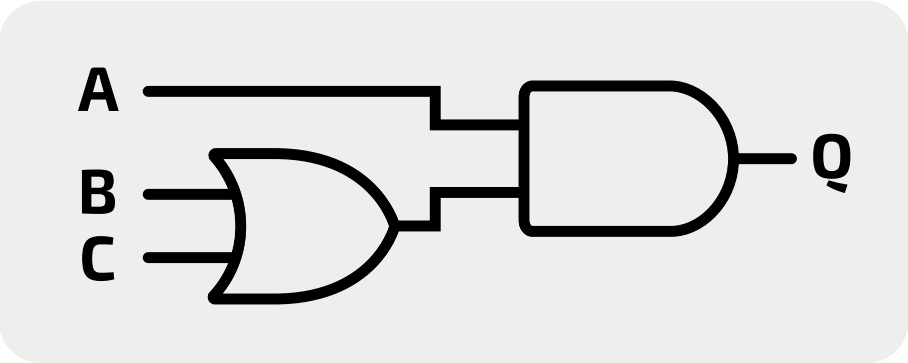


Figure 3: The circuit diagram for $Q = A \wedge (B \vee C)$