technoteach

technocamps

UNDEB EWROPEAIDD
EUROPEAN UNION

Llywodraeth Cymru
Welsh Government

Cronfa Gymdeithasol Ewrop
European Social Fund

Prifysgol Abertawe
Swansea University

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

PRIFYSGOL BANGOR UNIVERSITY

Cardiff Metropolitan University
Prifysgol Metropolitan Caerdydd

i.t.wales

PRIFYSGOL ABERYSTWYTH UNIVERSITY

Prifysgol Glyndŵr Wrecsam
Wrexham Glyndŵr University

University of South Wales
Prifysgol De Cymru

Boolean Algebra

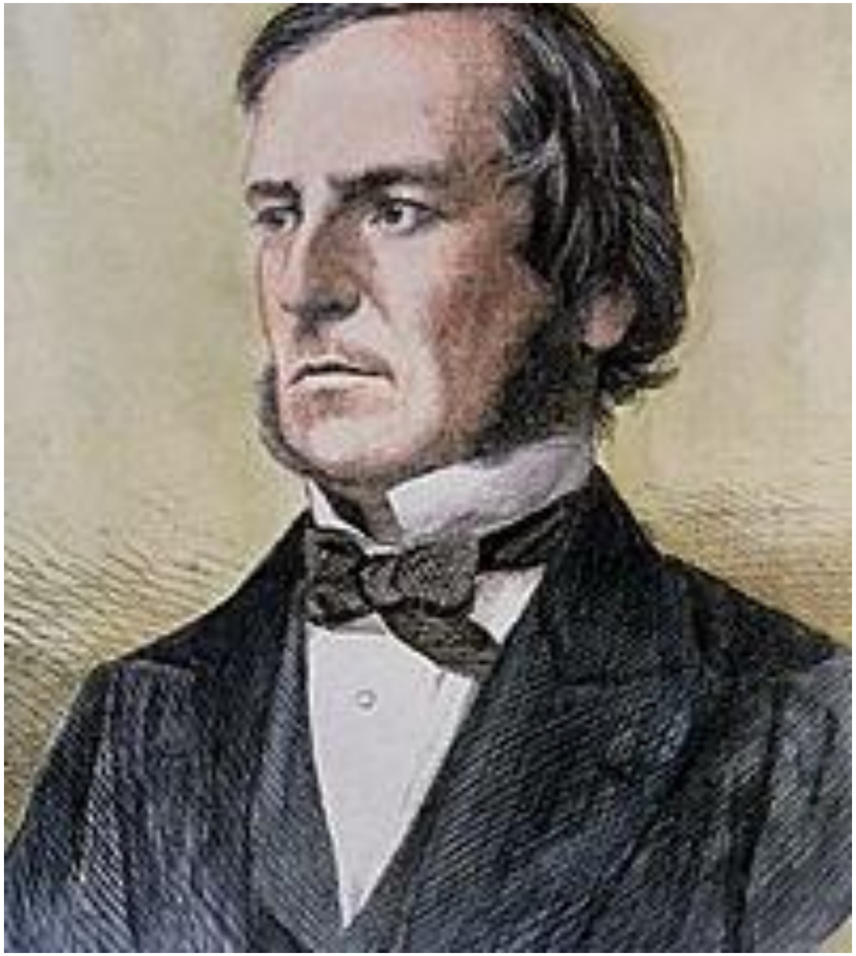
Who is George Boole?

George Boole (1815-1864)

George Boole was primarily a mathematician and worked as the first Queen's College Professor of Mathematics in Cork, Ireland.

He worked in the areas of differential equations and algebraic logic.

He is best known for his book "The Laws of Thought" which includes Boolean Algebra and is therefore seen as the forerunner of the information age.



Boolean Algebra

Boolean Algebra is just like normal algebra where we substitute a statement with letters. The difference is that the statements can only be true or false.

For example, we can represent the following logical statements in Boolean Algebra:

A – Today is Monday.

B – It is snowing.

These are either true or false.

Boolean Algebra

We are familiar with and use Boolean Algebra every day, maybe without realising.

For example:

If A is true, then NOT(A) must be false.

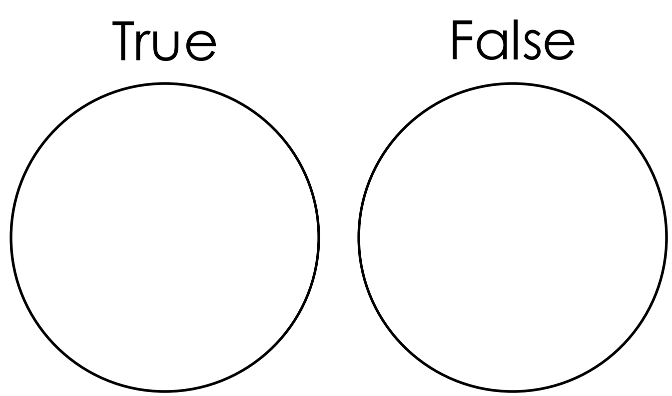
It sounds strange, but it makes sense if the statement is "Today is Monday.":

If "Today is Monday" is true, "Today is not Monday" must be false.

To Summarise:

Boolean Algebra is:

- Starting with the idea that some statement A, is either true or false, there is nothing between the two possibilities. (The law of the excluded middle.)
- By combining statements using operators like OR, AND, and NOT you can form new statements that are also either true or false.



Logical Operators

We use different symbols to represent the relations between statements.

Informal Name	Formal Name	Symbol
AND	Conjuction	$.$
OR	Disjunction	$+$
NOT	Negation	$\bar{A}$
XOR (Exclusive Or)	Exclusive Disjunction	$\oplus$

Combining Operators

We can combine Logical Operators:

Today is Monday AND It is Snowing.

A AND B

A . B

With this statement, both **A** and **B** must be true for **A . B** to be true.

That is, if it's Tuesday and it's snowing, **A . B** cannot be true.

AND (.)

I'm wearing a hat and I'm wearing a scarf, and I'm wearing gloves.

A AND B AND C

A . B . C

If I'm not wearing any of those items then the statement is not true. So every statement must be true with an AND operator to make the entire combination of statements true.

OR (+)

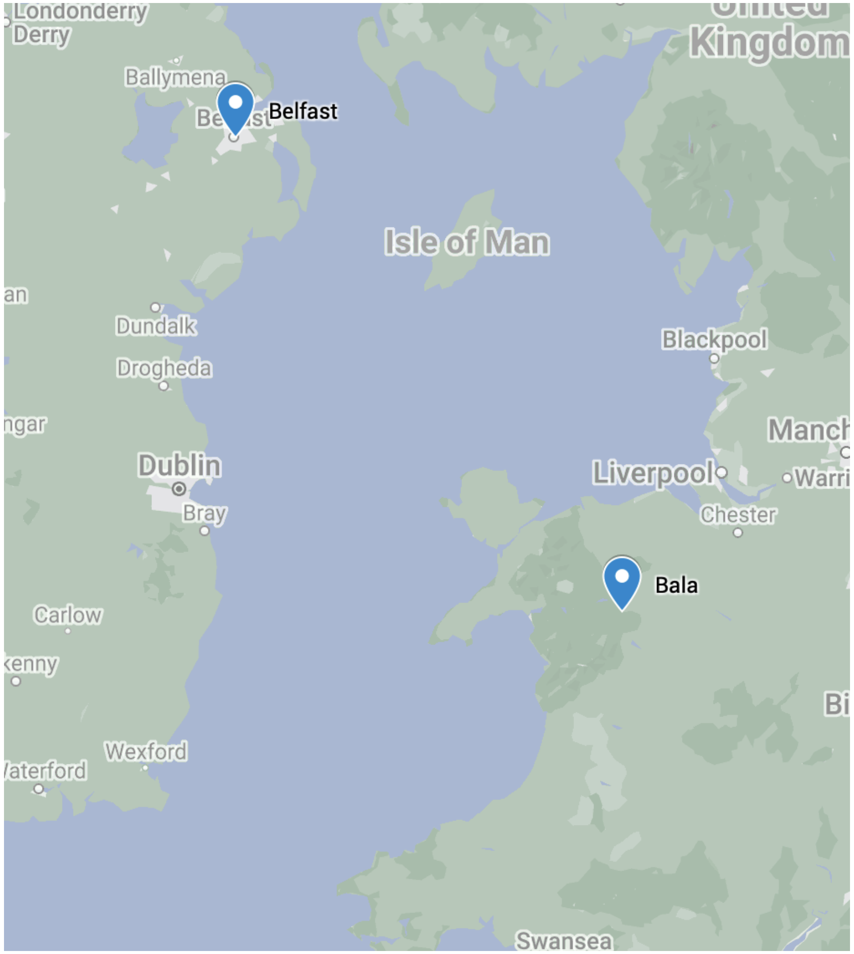
Consider the following statement:

I'm going to Bala or I'm going to Belfast.

A OR B

A + B

Do both A and B need to be true for this statement to be true?



NOT ($\bar{x}$)

NOT is a little more obvious than the others but in natural English the location of it is different.

It is NOT snowing.

NOT (It is snowing)

NOT A

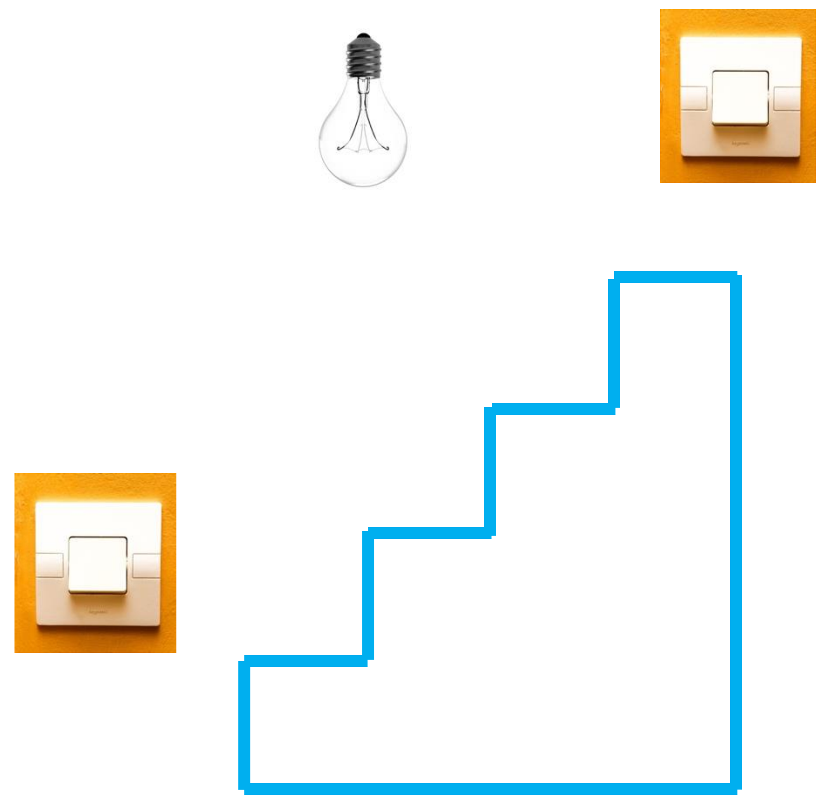
$$\bar{A}$$

‘Exclusive OR’ / XOR (⊕)

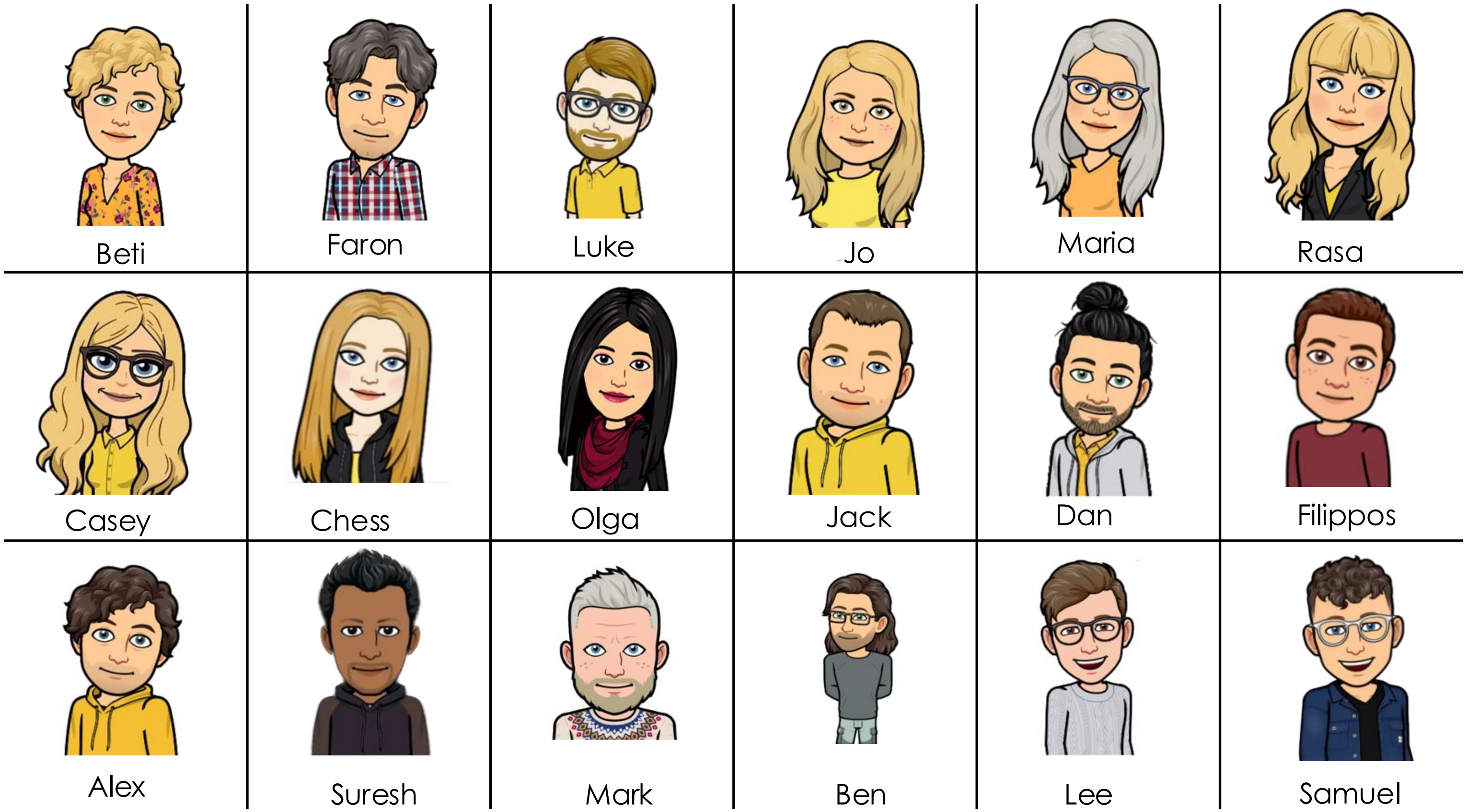
This is a bit trickier.

It’s similar to OR, but it is also **false if both are true.**

Imagine lights on stairs in a house. If there are two switches, the light is only on if one of the switches is on. Turning two on turns off the light again.



Bool' Who?


Beti
Faron
Luke
Jo
Maria
Rasa
Casey
Chess
Olga
Jack
Dan
Filippos
Alex
Suresh
Mark
Ben
Lee
Samuel

Truth Tables

A truth table is a way of checking whether a combination of statements is true or not, depending on whether the individual statements are true or false.

Truth Table Example:

A	B	Combination
False	False	**False**
True	False	**False**
False	True	**False**
True	True	**True**

What sort of combination is being shown in the truth table?

Truth Tables

I could use the numbers 0 and 1 instead of false and true to cut down on how much writing is needed to represent the data.

A	B	(A . B)
0	0	**0**
1	0	**0**
0	1	**0**
1	1	**1**

We can use this idea to make a truth table for every combination of statements.

Truth Table - AND

Draw a Truth Table to show the AND operator:

Truth Table - AND

Draw a Truth Table to show the AND operator:

A	B	(A . B)
0	0	**0**
1	0	**0**
0	1	**0**
1	1	**1**

Truth Table - OR

Draw a Truth Table to show the OR operator:

Truth Table - OR

Draw a Truth Table to show the OR operator:

A	B	(A + B)
0	0	**0**
1	0	**1**
0	1	**1**
1	1	**1**

Truth Table - NOT

Draw a Truth Table to show the NOT operator:

Truth Table - NOT

Draw a Truth Table to show the NOT operator:

A	$\overline{A}$
0	**1**
1	**0**

Truth Table – XOR

Draw a Truth Table to show the XOR operator:

Truth Table – XOR

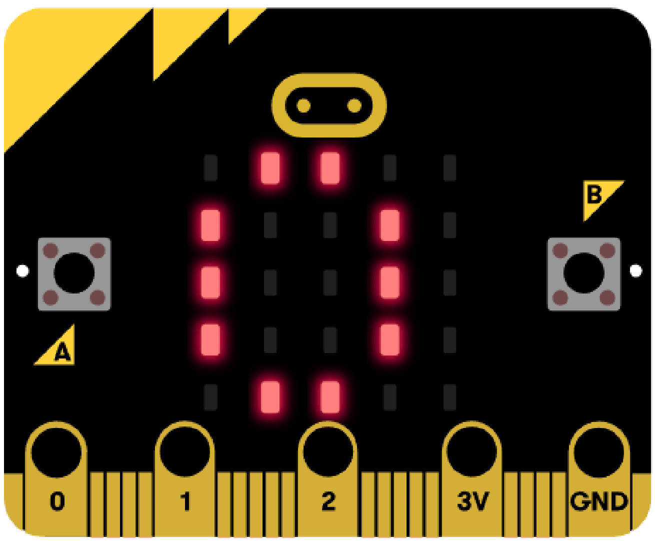
Draw a Truth Table to show the XOR operator:

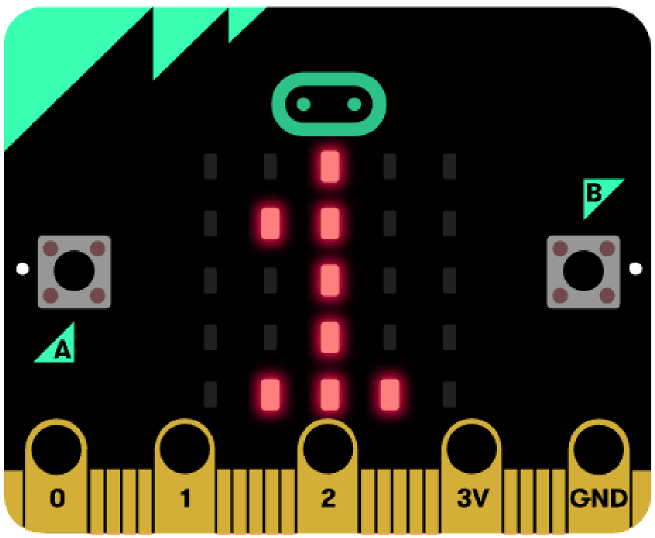
A	B	$(A \oplus B)$
0	0	**0**
1	0	**1**
0	1	**1**
1	1	**0**

Link for online simulation of program:
https://makecode.microbit.org/_1jrRJKd8j47U

Using micro:Bits to draw Truth Tables

After transferring the LogicGates program to the micro:Bit, shake the micro:Bit and it will show a random number.





Now press the buttons, A and B individually, or both together and see if it gives output or not. On the worksheet, fill in the relevant truth table based on the number you saw at the beginning.

Task: Chain of Operators

1 OR 0 = **answerA**

answerA AND 0 = **answerB**

answerB XOR 0 **= answerC**

NOT **answerC = answerD**

NOT (**answerD** XOR 1) = **answerE**

What is answerE?

Task: Chain of Operators

1 OR 0 = **answerA**	**= 1**
answerA AND 0 = **answerB**	**= 0**
answerB XOR 0 **= answerC**	**= 0**
NOT **answerC = answerD**	**= 1**
NOT (**answerD** XOR 1) = **answerE**	**= 1**

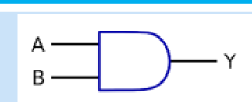
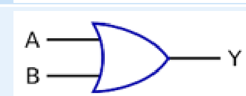
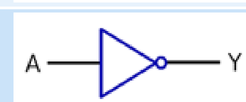
What is answerE?

Why is this Useful?

Logic Gates

The same logic (that is a binary system), is used when thinking about a series of switches or transistors.

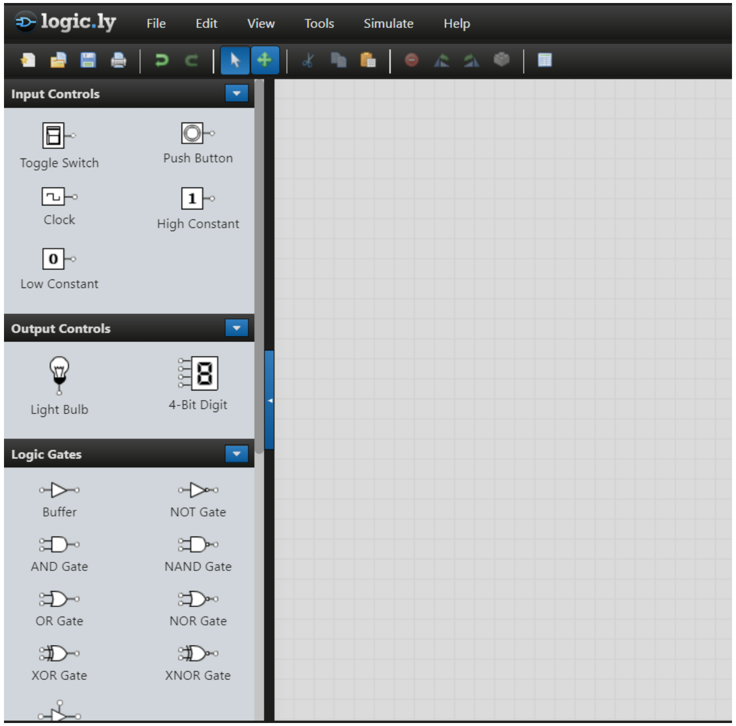
This means we can use it when thinking of electrical circuits:

Name	Symbol	Circuit Symbol
AND	$\cdot$	
OR	$+$	
NOT	$\bar{A}$	
XOR	$\oplus$	

Logic.ly

We can simulate electrical circuits using logic.ly

https://logic.ly/demo

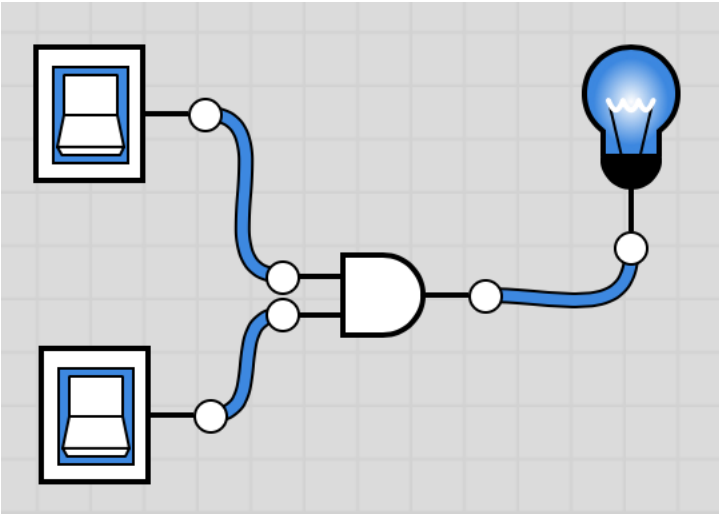


Task Logic.ly

For every one of the 4 operators, try to build a circuit that simulates it.

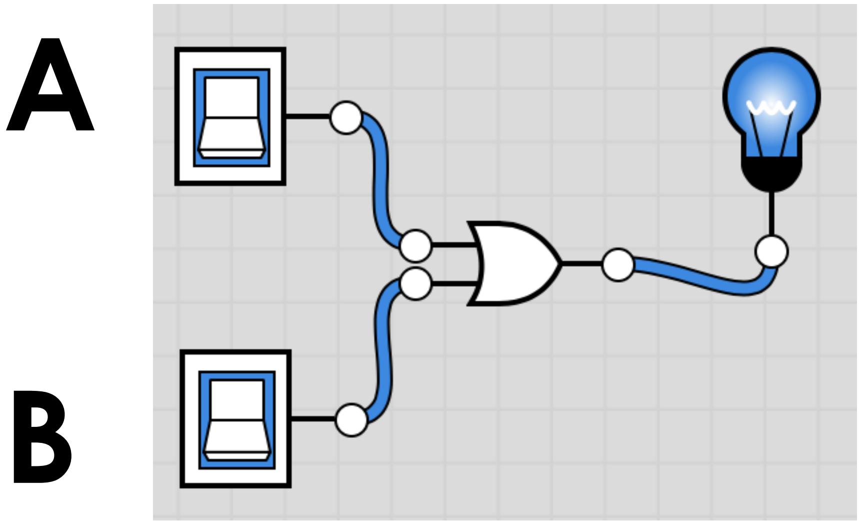
Truth Table 1 – A AND B

A



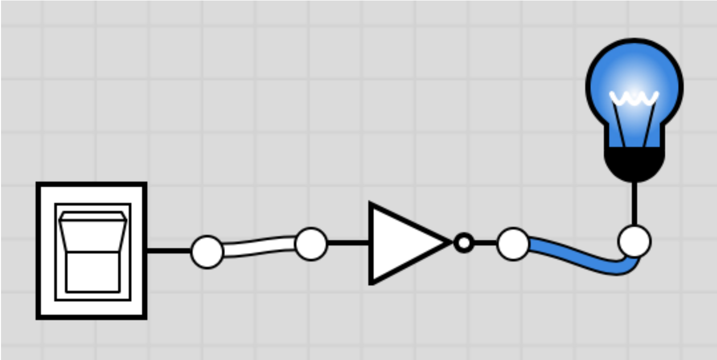
B

Truth Table 2 – A OR B



Truth Table 3 – NOT A

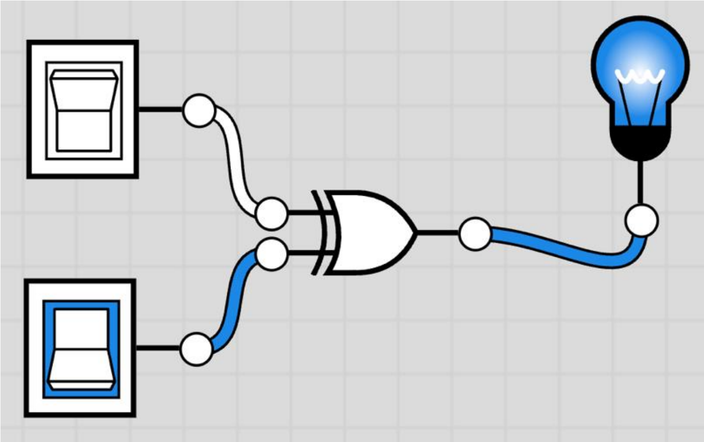
A



Truth Table 4 – A XOR B

A

B



BIDMAS/BODMAS

In mathematics, operations such as division, multiplication, addition and subtraction must be done in a particular order. We remember that by using BODMAS. Boolean algebra operations also have a specific routine.

Normal Algebra
Brackets
Indexes (Orders/Powers)
Division
Multiplication
Addition
Subtraction

Boolean Algebra
Brackets
NOT
XOR
AND
OR

AND Before OR

Because of the priority ranking, an AND operation is always done before an OR.

For Example:

$$5 \times 10 + 2 =$$

$$A \cdot B + C =$$

AND Before OR

Because of the priority ranking, an AND operation is always done before an OR.

For Example:

5 x 10 + 2 = (5 x 10) + 2 ✓

5 x 10 + 2 = 5 x (10 + 2) ✗

A . B + C = (A . B) + C ✓

A . B + C = A . (B + C) ✗

technoteach technocamps

Boolean Laws

Inverse Law $A + \overline{A} = 1$	***Complement Law*** $A \cdot \overline{A} = 0$	***Double Complement*** $\overline{\overline{A}} = A$
Annulment Law A + 1 = 1 A. 0 = 0	***Identity Law*** A + 0 = A A . 1 = A	***Idempotent Law*** A + A = A A . A = A
Commutative Law A + B = B + A A . B = B . A	***Associative Law*** A + (B + C) = (A + B) + C A . (B . C) = (A . B) . C	***Absorptive Law*** A + A . B = A A . (A + B) = A
Distributive Law A . (B + C) = A.B + A.C	**DeMorgan's Law** $\overline{A + B} = \overline{A} \cdot \overline{B}$ $\overline{A \cdot B} = \overline{A} + \overline{B}$	

Understanding The Laws

By using specific statements for A, B a C, try reading the laws to see if they make sense.

e.g. A = It is Raining

B = I'm wearing a hat

C = I have a cat

Commutative Laws: A + B = B + A , A . B = B . A

It's raining OR I'm wearing a hat = I'm wearing a hat OR it's Raining.

It's raining AND I'm wearing a hat = I'm wearing a hat AC it's Raining.

Understanding The Laws

Annulment Laws:

$$A + 1 = 1$$

$$A \cdot 0 = 0$$

It's raining OR TRUE = TRUE

It's raining AND FALSE = FALSE

Distributive Law

The distributive law tells us how to expand brackets.

The easy way to do this is to think of it as your normal algebra equivalents with "." as multiplication and "+" as addition.

For example:

$$2 \,.\, (3 + 4) = (2 \,.\, 3) + (2 \,.\, 4)$$

$$A \,.\, (B + C) = (A \,.\, B) + (A \,.\, C)$$

Try expanding the following statement: **(A + B) . C**

Distributive Law

By using the same law we can also factor statements into brackets.

For Example:

$$D . E + D . F = D . (E + F)$$

Try factoring the following statement: **A . B . C + A . D . E**

Absorptive Law

The Absorptive Law tells us that that:

$$A + A . B = A$$

$$A . (A + B) = A$$

This tracks when written as statements:

"I like Marvel" **OR** "I like Marvel **AND** Harry Potter"

"I like Marvel" **AND** "I like Marvel **OR** Harry Potter"

…as long as you like Marvel, these statement are true!

Absorptive Law

There is one additional expression in the Absorptive Law:

$$\mathbf{A + (\overline{A} . B) = A + B}$$

This is trickier to understand, however:

"I'm under 30" **OR** "I'm **NOT** under 30 **AND** I'm old"

is equivalent to:

"I'm under 30" **OR** "I'm old"

technoteach technocamps

Simplifying Expressions

1. A + A	2. B + B
3. A.B + A.B	4. D.F + D.F + G
5. D.E + $A.\overline{B}$ + D.E	6. A.A
7. H.H	8. (A + B).(A + B)
9. X.Y.X	10. B.1
11 C + 0	12. A.B.A.C.1
13 D + 0	14. A.B + 0
15. E.1	16. E.0
17. (A + A.B).0	18. A.B.1
19. $\overline{\overline{D}}$	20. $\overline{\overline{B+B}}$
21. $\overline{C+\overline{\overline{D}}}$	22. $A + \overline{B.\overline{B}}$

Simplifying a more Complex Expression

$$A.B.\bar{C} + A.B.C + A.\bar{B}$$

GCSE Questions

3. *(a)* State the logical operator that has been used to produce the output in the following truth table. [1]

Input		Output
A	**B**	**C**
0	0	0
1	0	1
0	1	1
1	1	0

....

(b) State the logical operator that has been used to produce the output in the following truth table. [1]

Input		Output
A	**B**	**C**
0	0	0
1	0	0
0	1	0
1	1	1

....

GCSE Questions

(c) **Tick** (✓) the correct boxes below to show the Boolean expression that represents the function described by each truth table.

(i) [1]

Input		Output
A	**B**	**C**
0	0	0
1	0	1
0	1	0
1	1	0

$C = A.B$ — 1 ☐

$C = \overline{A + B}$ — 2 ☐

$C = A.\overline{B}$ — 3 ☐

$C = A + B$ — 4 ☐

GCSE Questions

(ii)

Input		Output
A	B	C
0	0	1
1	0	0
0	1	0
1	1	0

[1]

$C = A.B$ 1 ☐

$C = \overline{A + B}$ 2 ☐

$C = A + B$ 3 ☐

$C = \overline{A.B}$ 4 ☐

GCSE Questions (2017 SAMs)

10. (a) (i) Complete the following truth table. [4]

A	B	$\overline{B}$	$A.B$	$A.\overline{B}$	$B + (A.\overline{B})$
1	1				
1	0				
0	1				
0	0				

(ii) Use this truth table to simplify the expression. [1]

$$B + (A.\overline{B})$$

....

....

....

GCSE Questions (2017 SAMs)

(b) (i) Using the following identities:

$$\boldsymbol{P.1 = P}$$

$$\boldsymbol{P.Q + P.R = P.(Q + R)}$$

$$\boldsymbol{P + \overline{P} = 1}$$

simplify the Boolean expression: [3]

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

....

....

....

....

GCSE Questions (2017 SAMs)

(ii) Draw a truth table for the expression: [4]

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

A Level Questions

Clearly showing each step, simplify the following Boolean expression: [5]

$$A.(A + C) + C.(A + B)$$

....

....

....

....

....

....

A Level Questions

(a) Complete the truth table below. [4]

A	B	A OR B	A AND B	A XOR B	A OR (NOT B)
0	0				
0	1				
1	0				
1	1				

How does a computer do this?

https://www.nandgame.com/