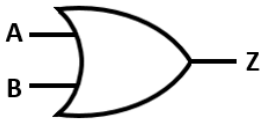
Computational Logic

AND

Both inputs must by positive to get a positive result

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

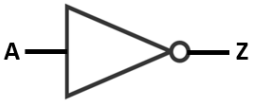
OR



At least one input must by positive to get a positive result

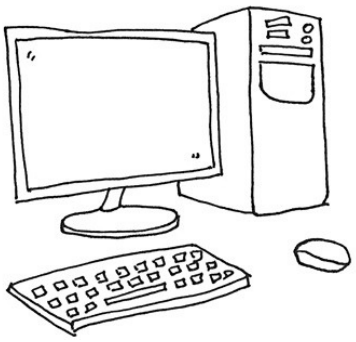
A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

NOT



The output is the opposite of the input

A	Z
0	1
1	0

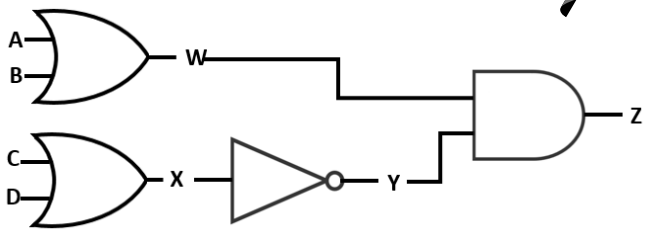


TRUTH TABLES

Create a column for each input needed and use binary counting to populate it with data (i.e. if there are 4 inputs you need 4 input columns). In this example we will use binary counting with 4 bits to fill the table, 0000, 0001, 0010, 0011 etc. Create a column for each output (i.e. if there are 4 logic gates used, there will be 4 outputs) and label each input and output column with a different letter of the alphabet.

Work out each output column using the correct input columns, in this example W (an OR gate) would only be using inputs A and B. Continue until all columns have been populated.

COMBINING LOGIC GATES



Shorthand

When writing logic problems, shorthand is often used to represent AND, OR and NOT.

A AND B = A ^ B

A OR B = A ˅ B

NOT A = ¬ A

Inputs				Outputs			
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	0
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	0
0	0	1	1	0	1	0	0
0	1	0	0	1	0	1	1
0	1	0	1	1	1	0	0
0	1	1	0	1	1	0	0
0	1	1	1	1	1	0	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	0
1	1	0	0	1	0	1	1
1	1	0	1	1	1	0	0
1	1	1	0	1	1	0	0
1	1	1	1	1	1	0	0

Computing-related Mathematics

Addition **+** (3 + 4 = 7)
Subtraction **-** (12 - 3 = 9)
Multiplication ***** (3 * 4 = 13)
Division / (100 / 4 = 25)
Exponentiation (to the power of) **^** (10 ^ 3 = 1000)
Whole number division **DIV** (13 DIV 5 = 2)
Remainder after whole number division **MOD** (13 MOD 5 = 3)