

Logic and computers

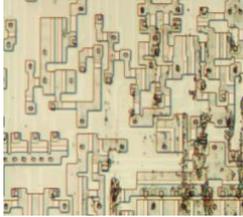
What this work will do:

The aim is to cover

1. Introduction
2. A starter activity - What is logic in computing ?
3. What are the building blocks of logic?
4. Simple Boolean Algebra
 - a. How do we turn a true or false statement into a logic diagram,
 - b. Truth tables – representing possible inputs and results
5. Turning the algebra expression into a logic circuit
6. Building a demonstration circuit using logic designer software
7. Bringing it together - building the major computer circuits
8. Summary Question – try a real A level Q
9. Follow up – questions about the course

1. Introduction

Computers are built around **electronic circuits** which are made up from wires and billions of transistors. These are engineered to form 'logic gates' like this



Source: <https://www.quora.com/How-are-electricity-and-computers-related>

Electricity flows through these logic gates and represents signals we need to act on

In computers we need to

- A. **Store** values being worked on.
- B. Do **CALCULATIONS**
- C. **Compare** one binary value with another.
- D. **Select** which signal to send from one of several inputs.
- E. **Output** a value e.g. for an 8 SEGMENT LED

Computers work on a **Binary or base 2** system, where values are on or off (AKA true or false, yes or no, 0 or 1).

It makes sense to create the circuits we need based on a similar ON/OFF, TRUE or FALSE, 0 or 1 system.

The system used is that based on **BOOLEAN LOGIC**.

This uses inputs and outputs of ON/OFF only.

In the computer's case the circuits take in signals (voltages) representing 0 or 1 and produce outputs signals (voltages) representing 0 or 1.

The results therefore depend on what you put in and what you designed the circuit to do logically to give you the output you want.

For example

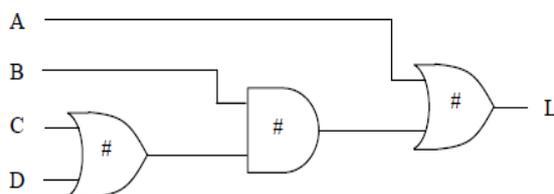
An interior light in a two-door car is controlled by two switches that the driver can turn on or off and two sensors, one per door. The switches are named A and B.

The door sensors are named C and D. The interior light is named L.

If a door is open the output of its sensor is on (C or D = TRUE).

If a door is closed the output of its sensor is off (C or D = FALSE).

- If both switches A and B are off then the light L is always off. (A and B = FALSE)
- If switch A is on the light L is always on. (A = TRUE)
- If switch B is on and switch A is off then:
 - the light L turns on if one or more of the car doors is opened (B = TRUE and C OR D = true)
 - the light L turns off if both of the doors are closed. (C OR D OR BOTH = FALSE)



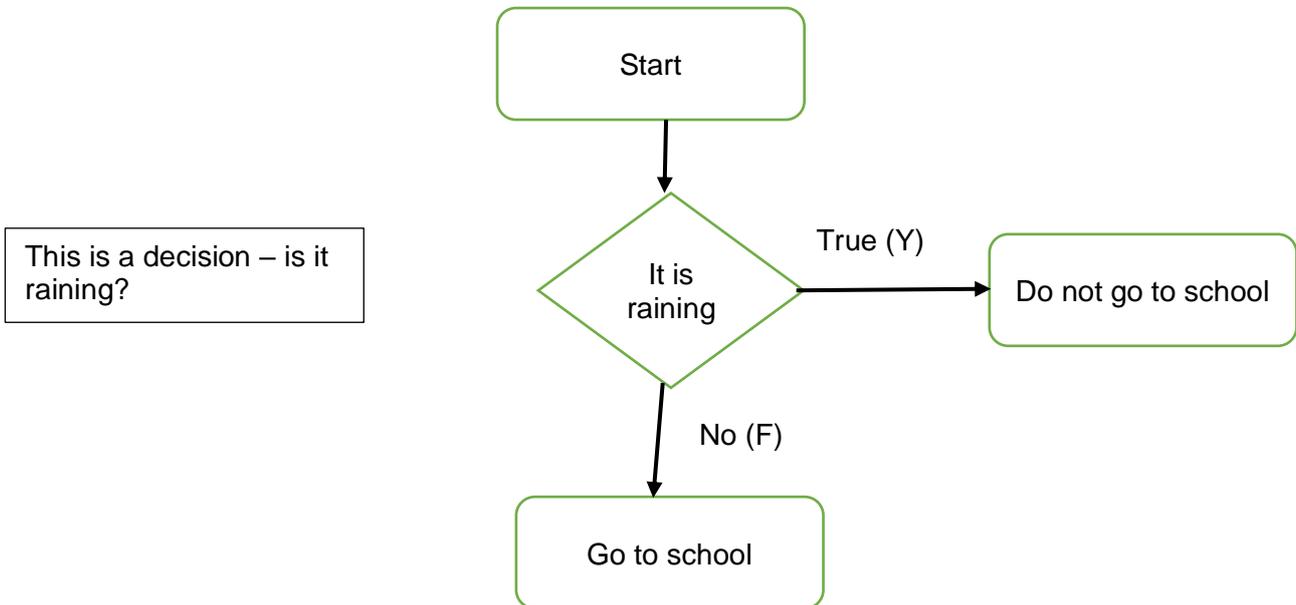
Becomes:

(there are at least three other ways of doing this!)

A starter activity - What is logic in computing ?

Logic is based on input values that can be TRUE or FALSE. These values are used to give a result (which is also true or false).

Example Flowchart: If it is raining then I will not go to school.



We can use a table to get the answer.

It is raining	Result
T	Do not go to school
F	Go to school

This is a single question – you get 2 possible results

We can add in more questions

Example: If it is raining AND I have to walk the dog then I will not go to school.



A **truth table** shows **all possible input combinations and the answer** for each of combination of questions.

e.g. 1 question = 2 results,
2 questions = 4 results and so on

Fill in the new TRUTH table.

It is raining	I have to walk the dog	Result
Y		
Y		
N		
N		

This is 2 questions – you get 4 results because there are 2 possible results for combination of input

2. What are the building blocks of logic?

The basic operations in logic are called logical operators (logical really!!). These are:

- a. AND
- b. OR
- c. NOT
- d. XOR

An operator takes at **least one** and **usually 2 or more** input values and produces an output

If you are not sure, look at <http://www.bbc.co.uk/education/guides/zc4bb9q/revision/1>

TASK 1 Quick definitions – fill in the blanks on the grid below

Common operator name	Posh proper name	Number of inputs (1, 2)	Explanation	How can it be written as a logical operation? (Optional)
AND	Conjunction	2	Both inputs must be true for the output to be true Input 1 = it is raining = T, Input 2 = I have to walk the dog = T, Output = will not go to school =T	A.B A AND B A ^ B
OR	Dysjunction			
XOR	Exclusive Dysjunction			
NOT	Negation			

There are other logical operators to enjoy(!) e.g.

NAND,

NOR and

NXOR.

These are combinations of the ones in the above table

4. Simple Boolean algebra

Using letters rather than sentences enables us to write a more **algebraic** form of the problem (i.e. in a simpler form). The following statement:

If it is raining and I have to walk the dog then I will not go to school

can be divided into three parts - two expressions (questions if you like) and a result based on whether the questions are answered as True or False

Question	Algebra form	Possible result
is it raining? (the first expression)	A	True or False
do I have to walk the dog (the second expression)	B	True or False
I will not go to school (result)	R	True or False depending on the value of the inputs



By replacing the questions with letters we simplify the problem to an algebraic form . this is a **truth table**

e.g.

A	B	R
F	F	F
F	T	F
T	F	F
T	T	T

This gives us an expression of $R = A \text{ AND } B$ (this can also be written as $R = A.B$ or $R = A \wedge B$)

TASK:

Fill in the truth table below for the following problem

If it is sunny (A) or I am happy (B), then I will go out for the day (R)

What are the possible values for A and B? What is the possible result in each case for R?

A (it is sunny)	B (I am happy)?	R (Go out for the day)
F	F	F

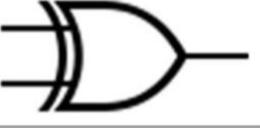
What's the Boolean algebra expression for this?

R=

.....

5. Turning the algebra expression into a logic diagram

To represent these circuits, we use LOGIC GATE diagrams which have a number of input lines and an output line. Each **logical operator** has a **logic gate** shape for it

GATE	SYMBOL
AND	
OR	
NOT	
XOR	

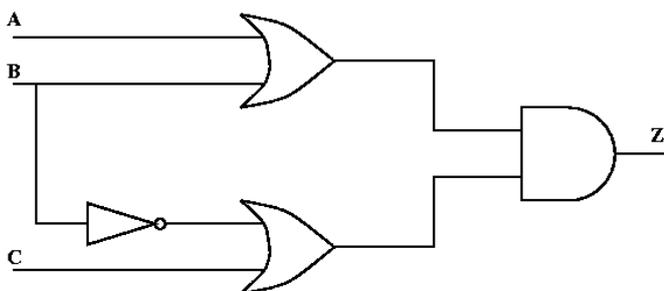
Converting between LOGICAL ALGEBRA EXPRESSIONS & LOGIC CIRCUITS

A logical algebra expression is converted to a logic circuit by substituting in an appropriate logic gate for each part of the logical algebra expression

Example 1: the logical algebra expression $A \cdot B$ becomes a single AND gate.



Example 2: the logical algebra expression $Z = (A \text{ OR } B) \text{ AND } (\text{not}(B) \text{ OR } C)$ becomes the following circuit diagram



6. Building a circuit using logic simulator software

We can create the circuit in a simulator and test it to see if it gives the correct results.

The link allows you to create and test logic circuits the <http://www.neuroproductions.be/logic-lab/>

The software is drag and drop so drag out the components, drop them on the screen then link together from the contact points (when these go green then drag to the next point to link the components together)

Task: Use the **inputs**, **logic ports** and **output** sections to build solutions to the following problems.

- **INPUT A** = **LYNN likes chocolate ice cream (can be true or false)**
- **INPUT B** = **LYNN likes vanilla ice cream (can be true or false)**
- **RESULT R** = **is true if Lynn likes chocolate ice cream(A= T) AND likes vanilla ice cream(B=T)**

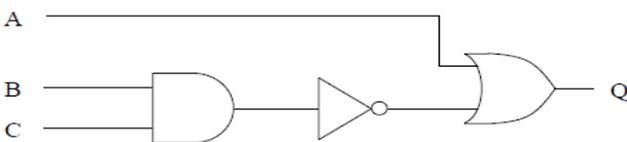


This shows the circuit built from the expression A and B. Click on the switches to see it work

Task: Try building the circuit for the extended ice cream problem

- **INPUT A** = **LYNN likes chocolate ice cream (can be true or false)**
- **INPUT B** = **LYNN likes vanilla ice cream (can be true or false)**
- **INPUT C** = **LYNN does not like ginger ice cream**
- **RESULT R** = **is true if Lynn likes chocolate ice cream(A= T) OR likes vanilla ice cream(B=T) AND dislikes ginger ice cream (C =T)**

Task: Given the diagram below, try building and testing the circuit for it



TASK: here are a range of logical algebra expressions. Build and test the circuit for each one

1. $A \text{ XOR } B \text{ XOR } C$
2. $(A \text{ XOR } C) \text{ OR } B$
3. $A \text{ OR } (\text{not}(B))$
4. $A \text{ AND not}(C \text{ OR } B)$
5. $\text{Not}(A \text{ AND } B \text{ AND } C)$

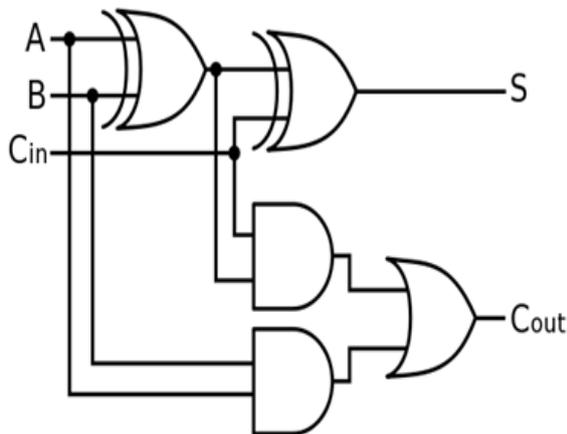
7. Bringing it together - building the major computer circuits

TASK: Using the <http://www.neuroproductions.be/logic-lab/> link, build and test the following circuit diagrams

You will need to create a truth table for each circuit to prove that your answer is correct

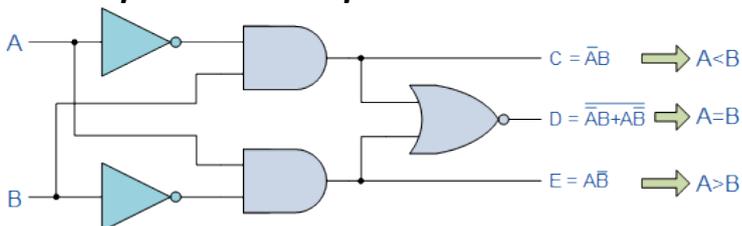
1. Addition - Full adder

TRUTH TABLE(S)



A	B	C _{in}	S	C _{out}
1	1	1		
1	1	0		
1	0	1		
1	0	0		
0	1	1		
0	1	0		
0	0	1		
0	0	0		

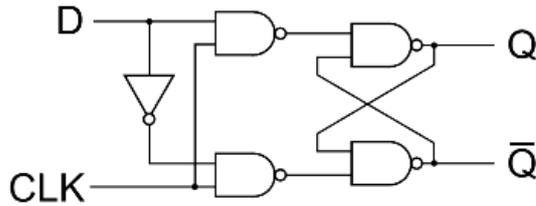
2. Comparison - Comparator circuit



TRUTH TABLE

A	Not(A)	B	Not(B)	C = B and not(A)	E = A and not(B)	D = Not(C or D)
0		0				
0		1				
1		0				
1		1				

3. Storage - D type flip flop (e.g. used in registers or RAM)

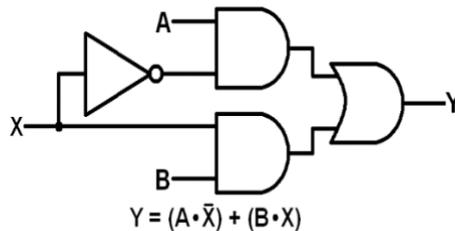


TRUTH TABLE

D	Clk	Q	Not Q
0	0	0	
0	1	0	
1	0	0	
1	1	1	

e.g. flip flop example answer at : <http://www.neuroproductions.be/logic-lab/index.php?id=91268>

4. Multiplexor A multiplexer performs the function of selecting the input on any one of 'n' input lines and feeding this input to one output line. The diagram below is a 2 to 1 multiplexer



TRUTH TABLE

A	B	X	Not(X)	A and not(X)	B and not(X)	Y

TASK: Extension

Can you join the adder and flipflop together so that the results of the adder (the sum(S) and the carry(C)) get stored (S goes into one flip flop, C goes into another!)

Congrats – you have just built a simple computer!!

8. Summary Question - a real A level Question:

Q17.

A burglar alarm system is to be implemented that has the following sensors:

- a door sensor **D** that outputs TRUE when the door is open and FALSE when the door is shut
- a pressure mat sensor **M** that outputs TRUE while a weight is detected on it and FALSE when no weight is detected on it.

The alarm also has a key **K** that turns the alarm on and off. **K** outputs a TRUE signal when the alarm is switched on and FALSE when the alarm is off.

The alarm output **A** sounds a bell. It should be TRUE if:

- the alarm is on AND
- either of the sensors **D** or **M** are set to the value TRUE.

- (a) In the space below, draw a logic circuit that will behave as described above for the inputs **D**, **M** and **K** and the output **A**.



(2)

- (b) Write a Boolean expression to represent the logic of this alarm system.

A=.....

Build it and prove it!!!

So what else can we use logic for?

Have a look at these

<http://pigofminecraft.blogspot.com/2011/07/xor-gates-how-and-why.html>

<https://learn.sparkfun.com/tutorials/logicblocks-experiment-guide/7-2-to-1-multiplexer>