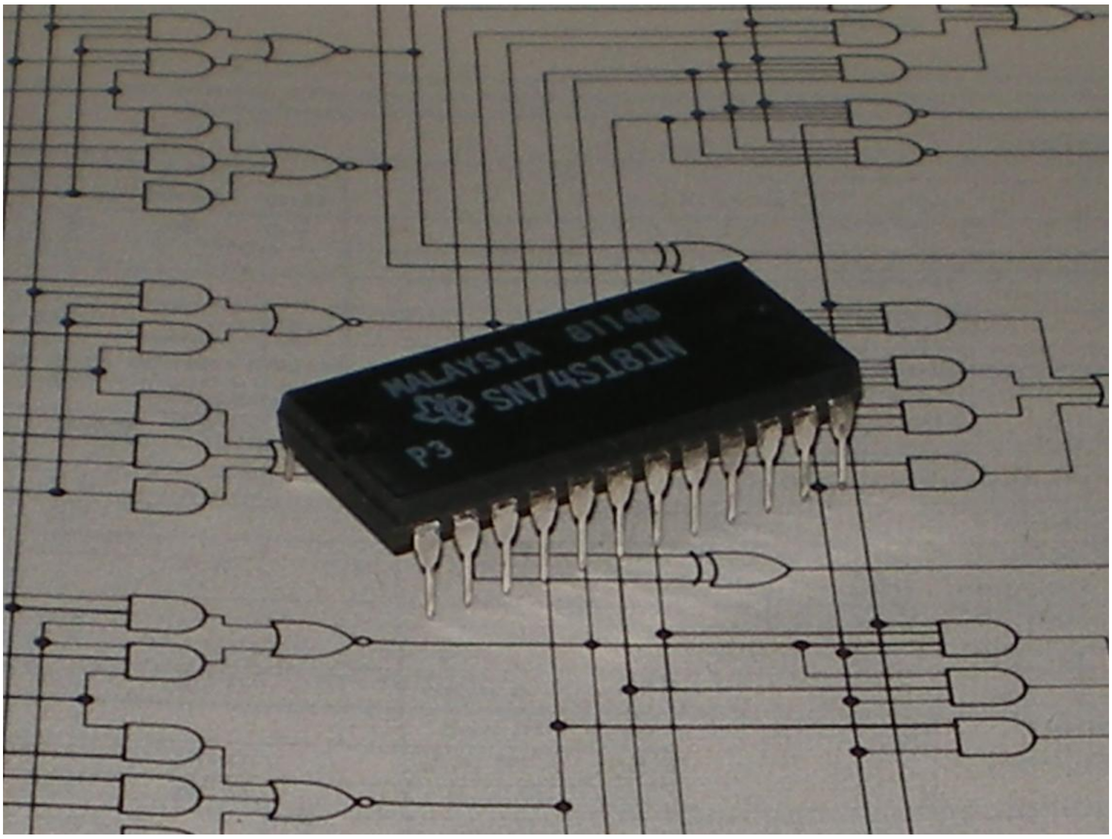
Engineering Science

Logic Control



Name ____________________

Class ____________________

Teacher ____________________

Learning Intentions

- To learn about "Logical" electronic systems
- To know what an IC chip is and how it is used
- To know about simple and complex logic gates
- To know and understand what a truth table is
- To know what a Boolean expression is and how to develop it

Success Criteria

- I can draw circuit diagrams of logic circuits
- I can develop analogue electronic control systems
- I can develop digital electronic control systems
- I am able to Simulate and construct digital control systems using logic components
- I understand the terms logic 1 and logic 0
- I can draw truth tables for the following gates: AND, OR, NOT
- Using simple formulae, I can construct Boolean equations

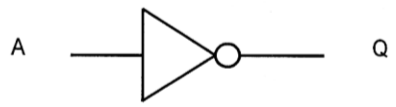
To access video clips that will help on this course go to **www.youtube.com/MacBeathsTech**

Switching Logic

Although it may not always seem like it, electronics and electronic systems are very logical in the way that they work. In the simplest form, if you want a light to come on, then you press a switch. Of course, it gets more complicated than that. Most technological systems involve making more complicated decisions: for example, sorting out bottles into different sizes, deciding whether a room has a burglar in it or not, or knowing when to turn a central heating system on or off.

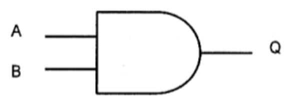
Logic gates are used in dealing with and processing a combination of different inputs. This switching logic can be applied to electrical switches and sensors, pneumatic valves or hydraulic systems. Switching logic uses logic gates to perform decisions. In S1 and S2 you have already have experience of NOT, AND and OR logic gates.

NOT gate



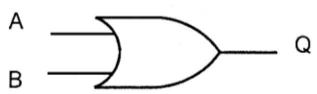
A	Q
0	1
1	0

AND gate



A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

OR gate

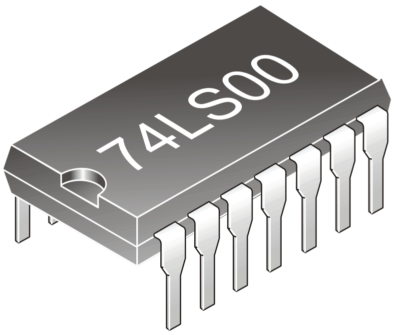


A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

http://www.bbc.co.uk/schools/gcsebitesize/design/electronics/logicrev2.shtml

Integrated circuits

Although logic gates have electronic symbols, they are not discrete components: they are contained in integrated circuits. Integrated circuits (ICs) are silicon-based components containing complex circuits. There are two main types TTL (transistor-transistor Logic or 7400IC series) or CMOS (Complimentary Metal Oxide Silicon or the 4000IC series).
You will be dealing with TTL IC Chips. TTL stands for transistor–transistor log.



We will be using TTL IC's and there reference numbers start with "74".
You can see the outline diagram of the 74 serious in your data booklet. This diagram will show you the pin connections as the pins. IC's are too small to have the numbers on in real life.
Key things to remember are that there will be a notch in one end of the IC and beside it a big dot. The dot is positioned next to Pin 1 and the pins are numbered anti-clockwise from pin1 to pin14.
Notice that the IC gets its power through Pin 7 and Pin 14 and it is a signal voltage going to or coming from the other pins.

Differences between TTL and CMOS ICs

TTL chips have been traditionally used in schools because CMOS chips were easily damaged by static energy. However there are other differences.

CMOS chips only use small currents (about 8uA) while TTL ones use a much larger current (about 3mA, which is almost 400 times as much) which is reflected in their power consumption especially when you may have hundreds of them in a mainframe computer.

TTL chips need more precise supply of voltage than their CMOS equivalents but they have much faster switching speeds, which is important in high-speed telecommunications, etc.

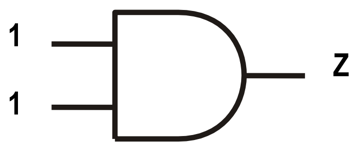
The main differences are summarised in the table below.

Property	TTL	CMOS
Series	7400	4000
Power supply	5 + or – 0.25v	3 to 15v
Unused inputs	Float high	indeterminate
Power consumption	High	Low
Switching speed	Fast	Slow
Effect of static	Not affected	Can be damaged

Task 1

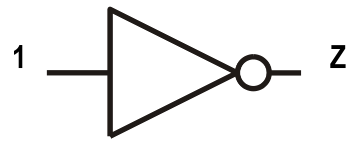
For each of the following examples, state whether the output Z is at logic 0 or logic 1.

a)



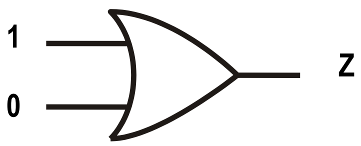
Z =

b)



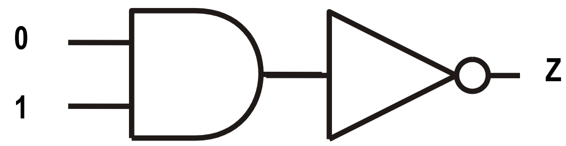
Z =

c)



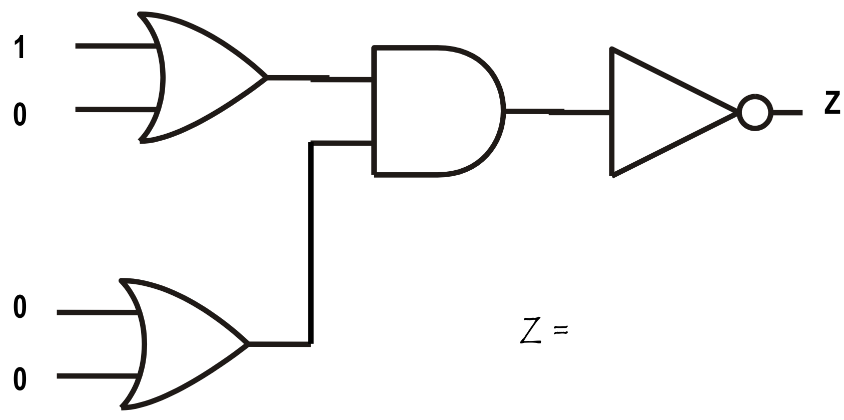
Z =

d)



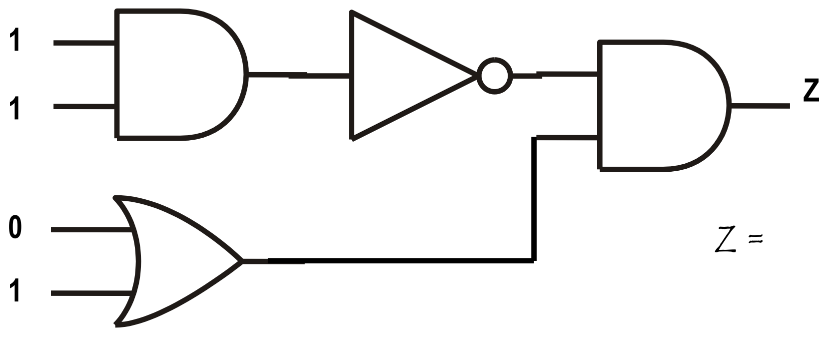
Z =

e)



Z =

d)



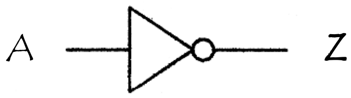
Z =

By using the E & L Boards, build the circuits using switches as inputs and a bulb as check if you are correct

Boolean Expressions

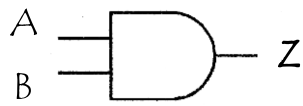
Each logic gate has a corresponding Boolean mathematical formula or expression. The use of these expressions saves us having to draw symbol diagrams over and over again.

NOT



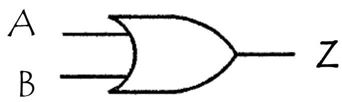
$$Z = \overline{A}$$

AND



$$Z = A.B$$

OR

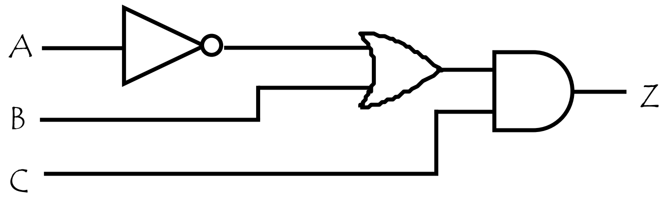


$$Z = A+B$$

Combinational Boolean

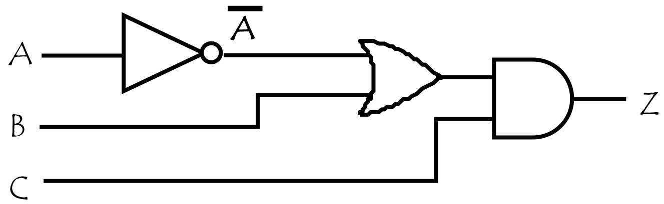
So far, we have only looked at simple logic systems. In reality, most logic systems use a combination of different types of logic gates. This is known as '*Combinational Logic*'. Boolean Expressions can be worked out from these to know the equation for the circuit..

Example



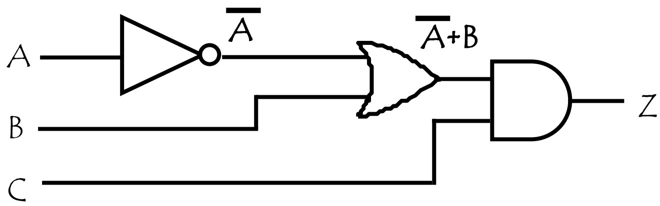
To work this out you have to take it one step at a time and work out the equation as it goes through each logic gate.

1.



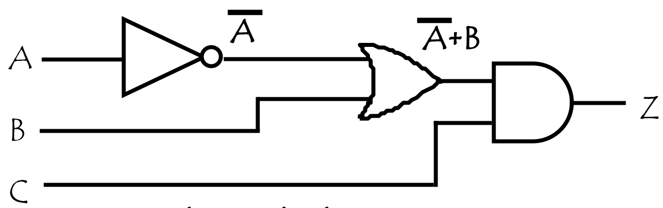
You can find out that the line for A soon turns into $\overline{A}$

2.



The lines for A and B now changes at it goes through the next logic gate.

3.



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

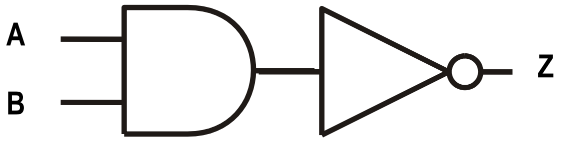
As we progress through the circuit we can now add C.

$Z = (\overline{A}+B) \cdot C$

Task 2

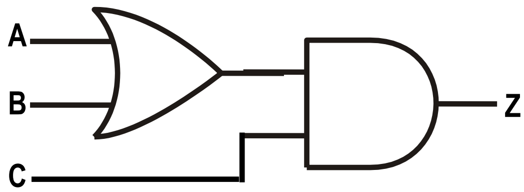
Work out the Boolean Expression for each of the following

a)



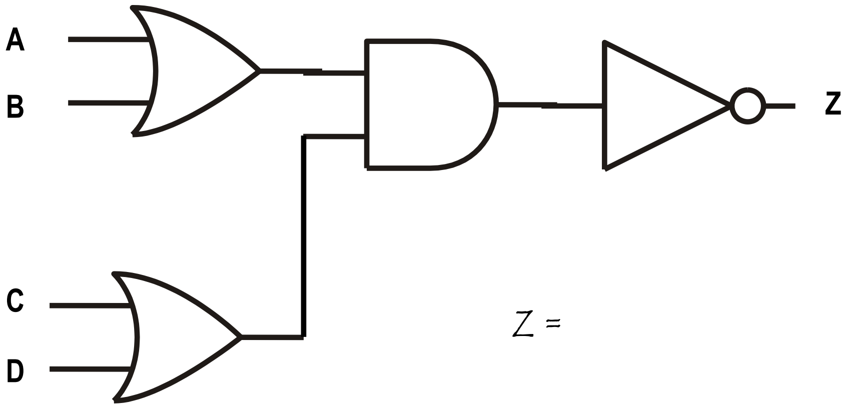
Z =

b)



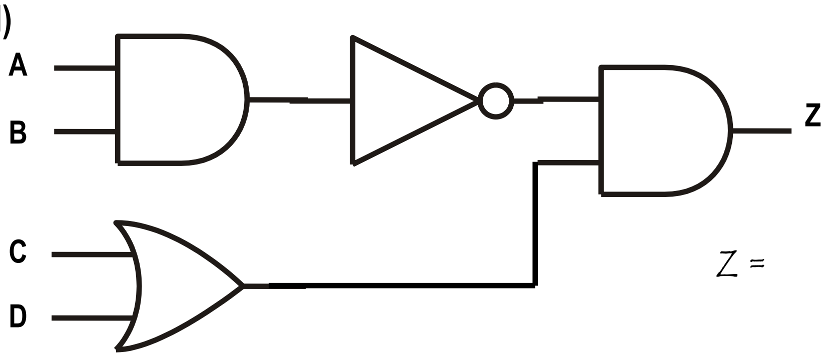
Z =

c)



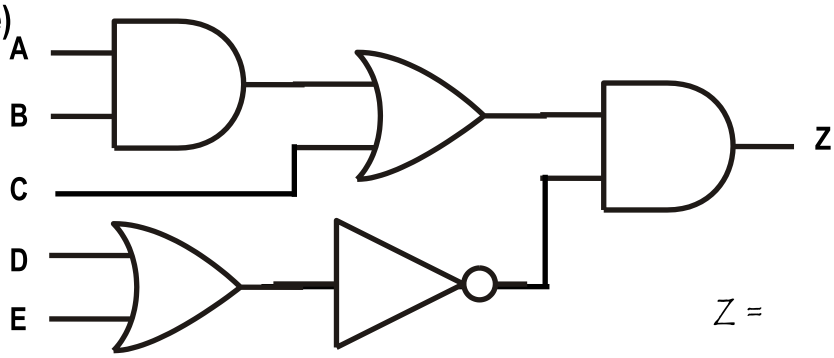
Z =

d)



Z =

e)

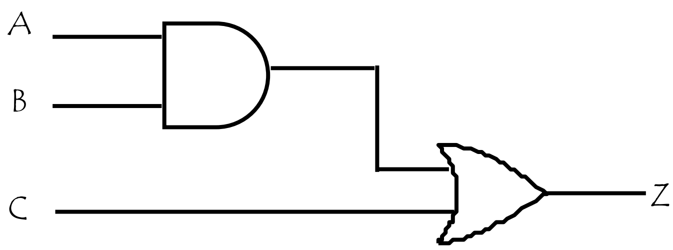


Z =

Combinational Logic tables

For each Combinational Logic diagram a logic table can also be worked out. This can look confusing but if we take our time it can be very simple.

Example



http://www.youtube.com/wah?v=6et_MMBJYRA

By creating a new row in our truth table for once the circuit has went through a logic gate we can take this in simple steps

1. We also need to have extra rows to ensure we are showing EVERY possibility.

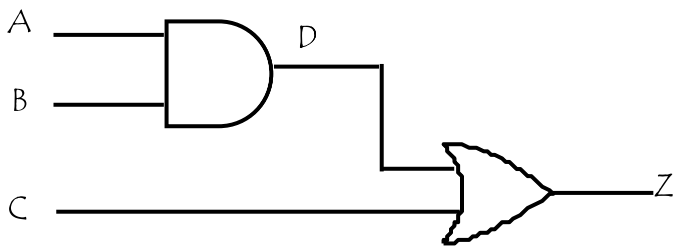
In this case we need to work out 2 possible options for everything the OR gate. This means there will be 8 possible options.

Another we of ensuring we have all the possible options is using the Say we have 3 inputs to a logic system, using powers of 2 we can calculate the number of possible combinations of input to the circuit as follows,

No. of combinations = 2^3 = 8

A	B	C	D	Z

2.



A	B	C	D	Z
0	0		0	[redacted]
0	0	[redacted]	0	[redacted]
0	1	[redacted]	0	[redacted]
0	1	[redacted]	0	[redacted]
1	0	[redacted]	0	[redacted]
1	0	[redacted]	0	[redacted]
1	1	[redacted]	1	[redacted]
1	1	[redacted]	1	[redacted]

We now have to take this in stages.

D is the output for A and B so this is what to work out first.

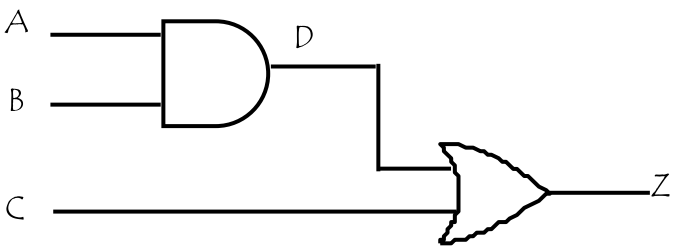
3.

We now have to put in the options for C ensuring we have covered ever single possibility.

Eg. There should be 2 0 – 0 combinations for A& B, so we have to put the 2 possible options in for C to ensure we have covered every single possible option

A	B	C	D	Z
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	
1	1	1	1	

4.



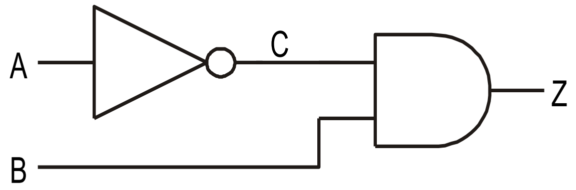
A	B	C	D	Z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

By looking at our diagram we can see that the inputs to Z is D & C. We now use these 2 rows to work out Z

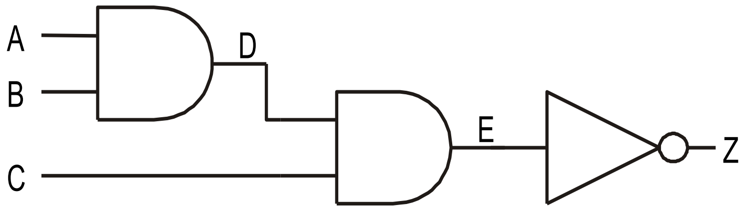
Task 3

In your jotters, redraw these combinational logic circuits, work out the Boolean expression then work out the logic tables.

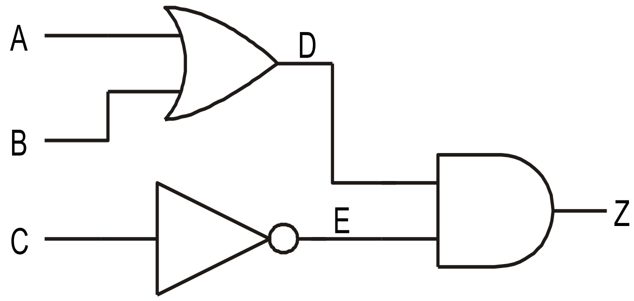
a)



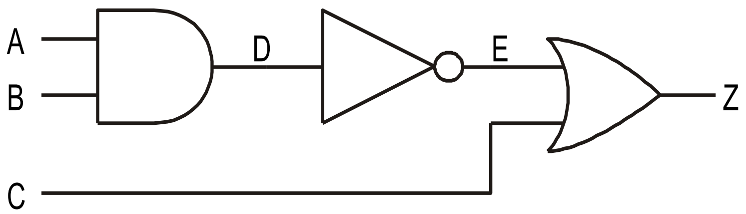
b)



c)



d)



Creating Logic Diagrams From Truth Tables

When designing systems, it is normal to design a logic diagram from a prepared truth table. This may seem difficult to start with, but if you concentrate on the **combinations** which give a **logic 1** condition in the **output column**, solutions can be found easily.

The truth table below shows two inputs, A and B, and one output, Z.

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

→ $Z = A \bullet \overline{B}$

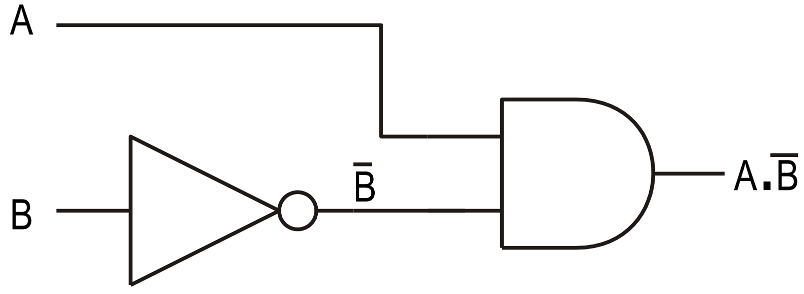
The output Z is at logic 1 in the third row down, and we can see that for this to happen A must be at logic 1 and B must be at logic 0. In other words

Z = A AND NOT B

This means that we need a two-input AND gate, with B being fed through a NOT gate. We can write the statement in shorthand Boolean as

$$Z = A \bullet \overline{B}$$

This means that the logic diagram is as shown below.



Example

In this problem we have three inputs, A, B and C, with one output, Z. From the truth table we can see that there are two occasions when the output goes to logic 1.

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

→ $Z = \overline{A}.B.C$

→ $Z = A.B.\overline{C}$

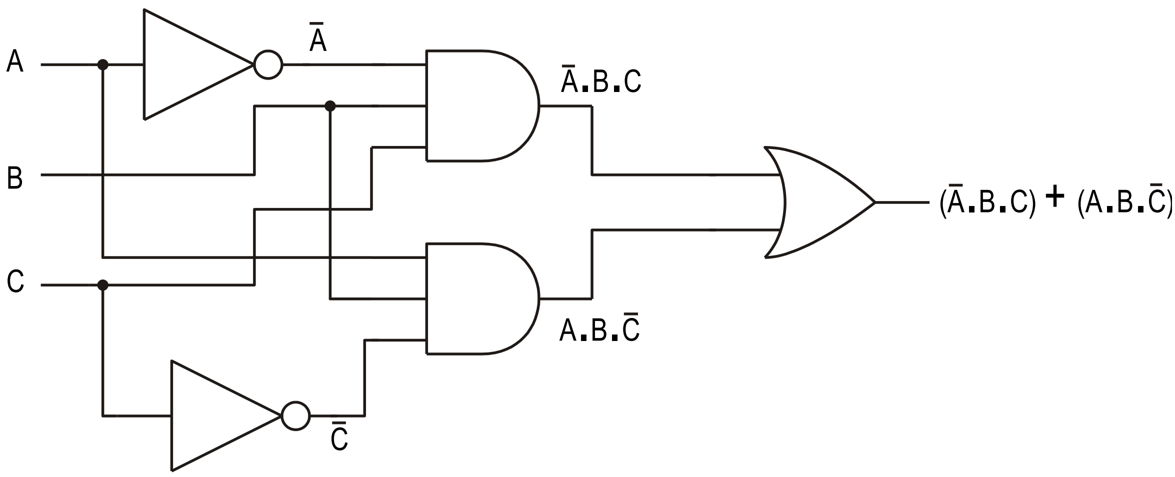
In other words, Z = 1 if

A is at logic 1 AND B is at logic 1 AND C is at logic 1

OR

A is at logic 1 AND B is at logic 1 AND C is at logic 0

This means we need a two-input OR gate being fed from two three-input AND gates as shown below.



The shorthand Boolean equation for this truth table is

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

Task4

Draw the logic diagrams and Boolean Expression for the following truth tables

a)

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

b)

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

c)

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

Task 5

Draw the logic diagrams and Boolean Expression for the following truth tables.

a)

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

b)

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