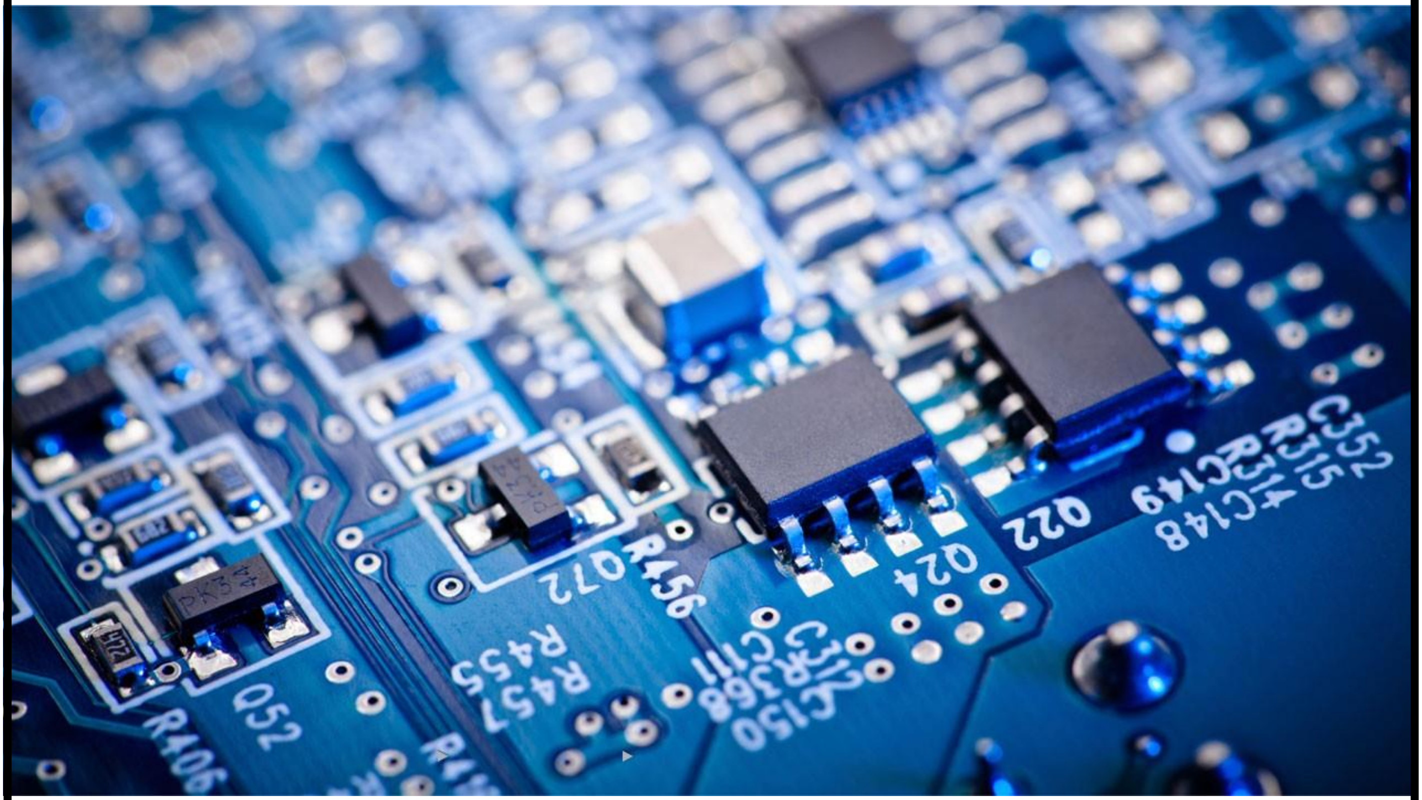
HIGHER ENGINEERING SCIENCE

Digital Electronic Engineering



Name ______________________

Class ______________________

Teacher ______________________

Ellon Academy
Technical Faculty

Learning Intentions

- I will learn about TTL and CMOS families of IC's, being able to distinguish between them
- I will be able to identify single logic gate symbols
- I will be able to complete Truth Tables for single logic gates and combinational logic circuits
- I will learn to analyse and simplify combinational logic circuits
- I will learn what a Boolean expression is and how to write one for a given logic circuit
- I will learn about NAND gates and I will be able to determine equivalent circuits made from them
- I will learn to form circuits to given specification

Success Criteria

I can develop digital electronic control systems by:

- Designing and constructing complex combinational logic circuits
- Describing logic functions using Boolean operators
- Simplifying logic circuits using NAND equivalents
- Testing and evaluating combinational logic circuits against a specification

YENKA is a free program you can download at home to build electronic circuits and help you with your studies

http://www.yenka.com/en/Free_student_home_licences/

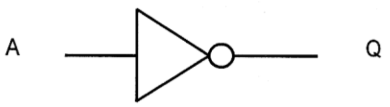
To access video clips that will help on this course go to **www.youtube.com/MacBeathsTech**

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.

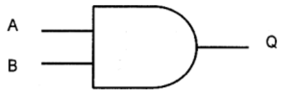
Logic gates are used in dealing with and processing a combination of different inputs. This switching logic can be applied to electrical switches and sensors, pneumatic valves or hydraulic systems. Switching logic uses logic gates to perform decisions. In National 5 you have already gained experience in the use of NOT, AND and OR logic gates.

NOT gate



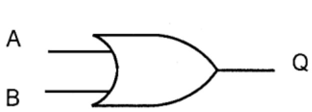
A	Q
0	1
1	0

AND gate



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

OR gate

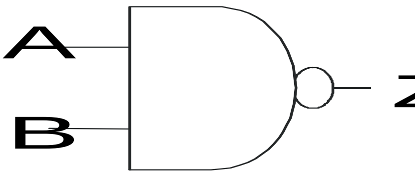


A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

http://goo.gl/yo2wOC

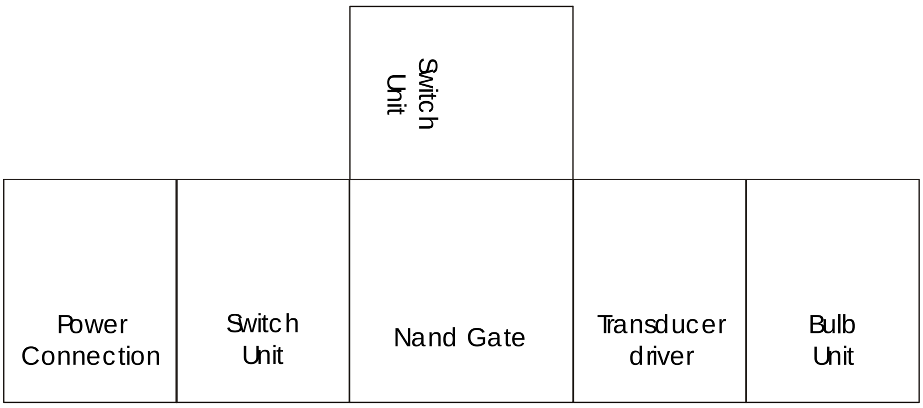
NAND logic gate

The NAND gate is effectively an *inverted* AND gate. In other words, the results obtained from the output are the opposite to those of the AND gate. This gate is often referred to as 'NOT AND'.
When drawing up the truth table for the NAND gate it can be difficult to 'picture' or imagine the results. The best way to do this is to pretend that it is an AND gate and then invert (reverse) the results, thus giving you the outputs for the NAND gate.



Symbol for NAND Gate

Task 1
Set up the E&L modular board electronic system and complete the NAND table using the system to confirm the outputs at Z.

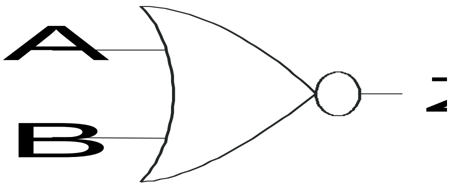


A	B	AND	Z
0	0		
0	1		
1	0		
1	1		

NAND Truth table

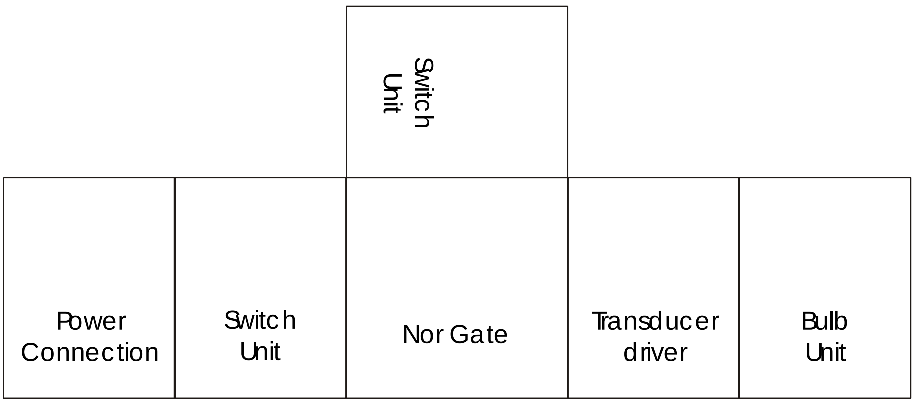
The NOR logic gate

The NOR gate is effectively an *inverted* OR gate. In other words, the results obtained from the output are the opposite to that of the OR gate. This gate is often referred to as 'NOT OR'.
As with the NAND gate, when drawing up the truth table for the NOR gate it can be difficult to 'picture' or imagine the results. The best way to do this is to pretend that it is an OR gate and then invert (reverse) the results, thus giving you the outputs for the NOR gate.



SYMBOL FOR NOR

Task 2
Using the E&L boards once more, build up this circuit to confirm how a NOR gate works, and complete the truth table.

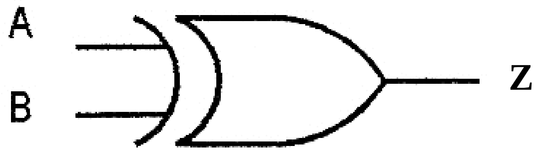


A	B	OR	Z
0	0		
0	1		
1	0		
1	1		

NOR Truth table

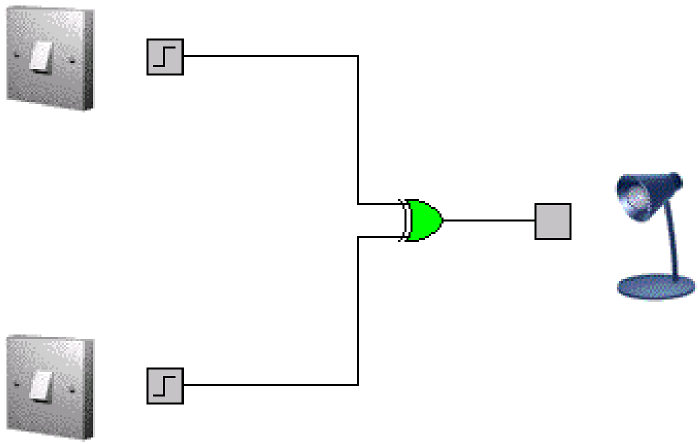
The XOR Logic Gate

This IC sends out logic when only 1 OR the other is sending logic 1. It will not work if both signals are sending out logic 1 like a normal OR gate does. This is why a XOR gate is sometimes known as an 'X-Treme OR' or an 'Exclusive OR'



Symbol for XOR Gate

Task 3
Using Crocodile Technology, build up this circuit to confirm how a XOR gate works, and complete the truth table.



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

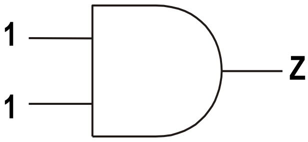
XOR Truth table

http://goo.gl/4uQfaP

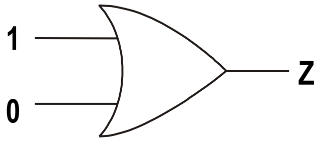
<u>Task 4</u>

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

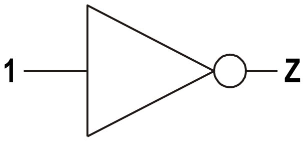
(a)



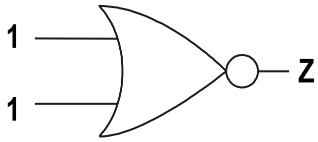
(b)



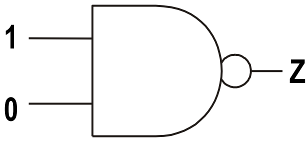
(c)



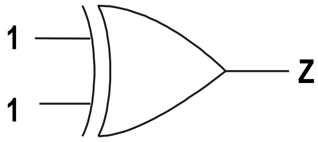
(d)



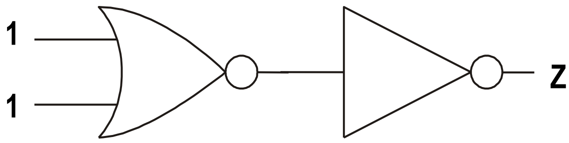
(e)



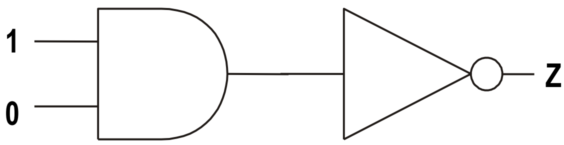
(f)



(g)



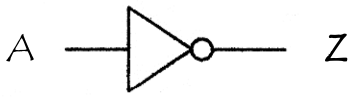
(h)



Boolean Expressions

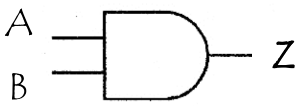
Each logic gate has a corresponding Boolean mathematical formula or expression. The use of these expressions saves us having to draw symbol diagrams over and over again.

NOT



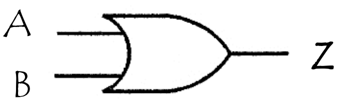
$Z = \overline{A}$

AND



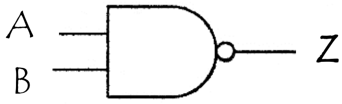
$Z = A.B$

OR



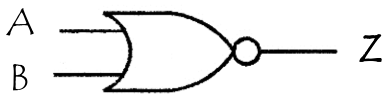
$Z = A+B$

NAND



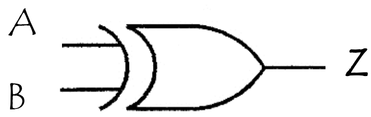
$Z = \overline{A.B}$

NOR



$Z = \overline{A+B}$

XOR

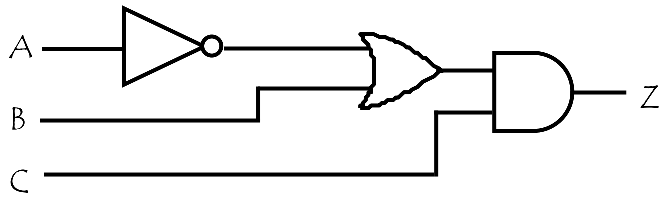


$Z = A \oplus B$

Combinational Boolean

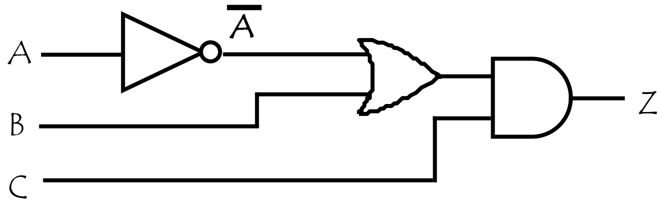
So far, we have only looked at simple logic systems. In reality, most logic systems use a combination of different types of logic gates. This is known as '*Combinational Logic*'. Boolean Expressions can be worked out from these to know the equation for the circuit..

Example



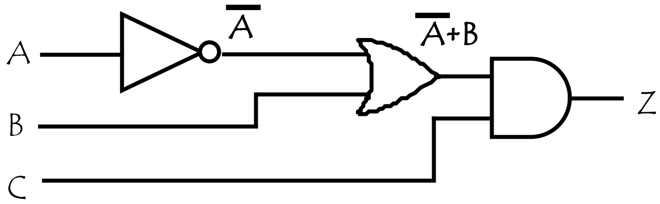
To work this out you have to take it one step at a time and work out the equation as it goes through each logic gate.

1.



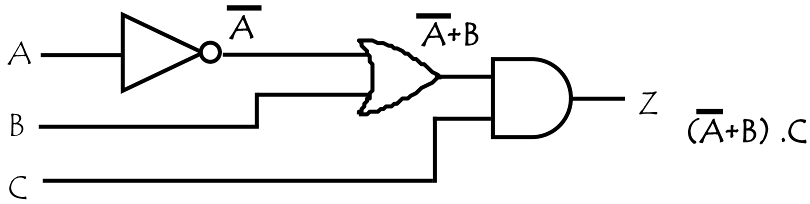
You can find out that the line for A soon turns into $\overline{A}$

2.



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

3.



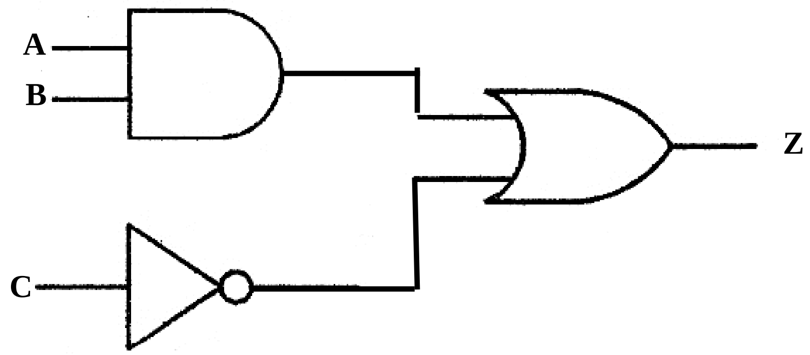
As we progress through the circuit we can now add C. **$Z = (A+B) \bullet C$**

Task 5

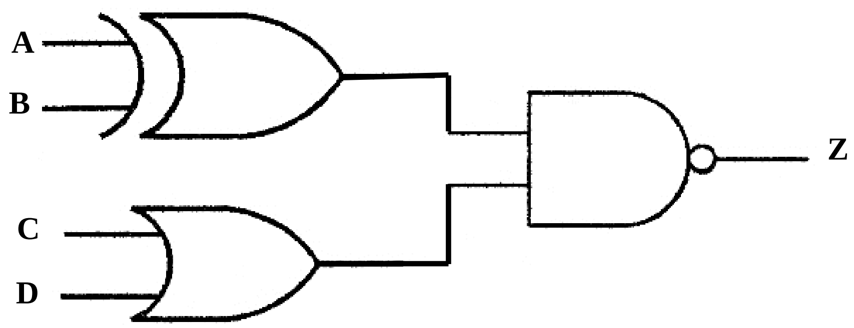
Work out the Boolean Expression for each of the following

a)

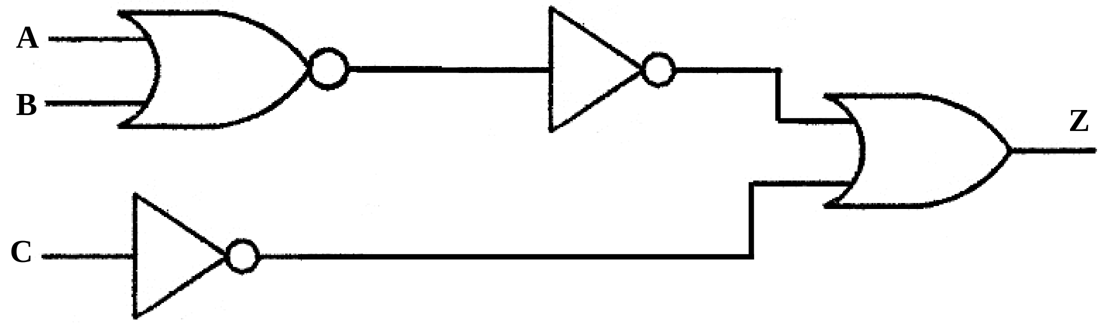
b)



c)



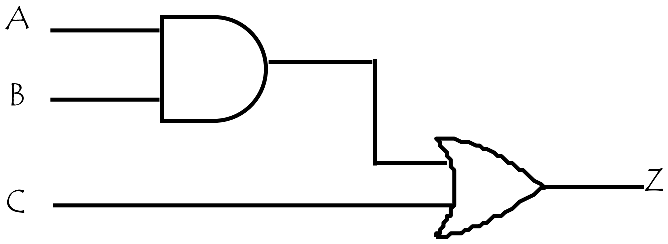
d)



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



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.

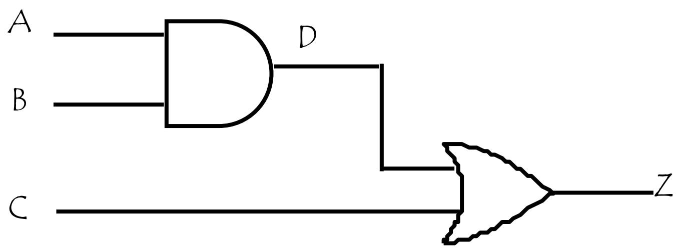
In this case we need to work out 2 possible options for everything the OR gate. This means there will be 8 possible options.

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.



We now have to take this in stages.

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

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

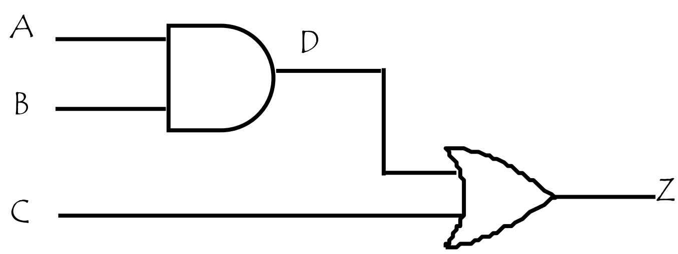
3.

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.



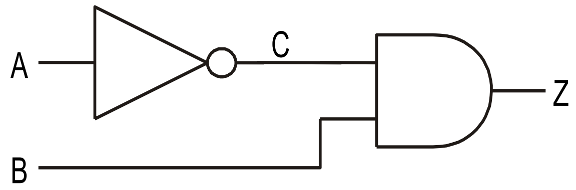
By looking at our diagram we can see that the inputs to Z is D & C. We now use these 2 rows to work out Z

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

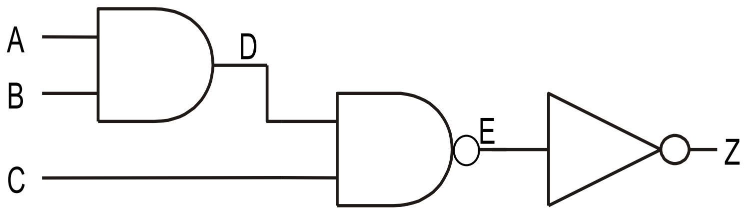
Task 6

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

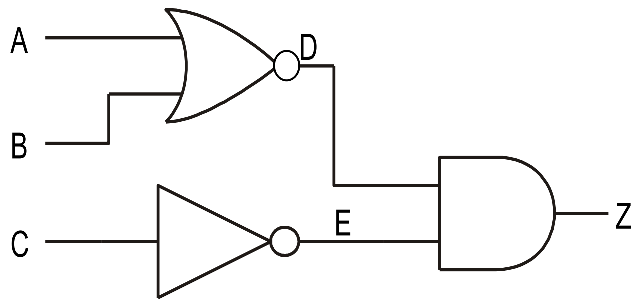
a)



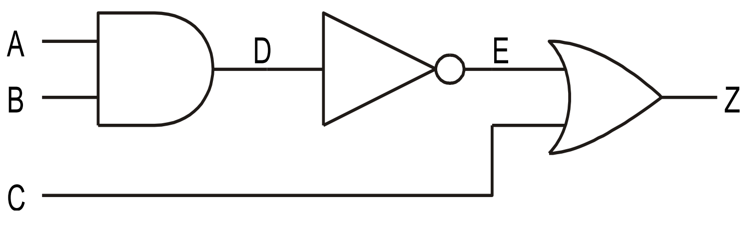
b)



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

→ $Z = A \bullet \overline{B}$

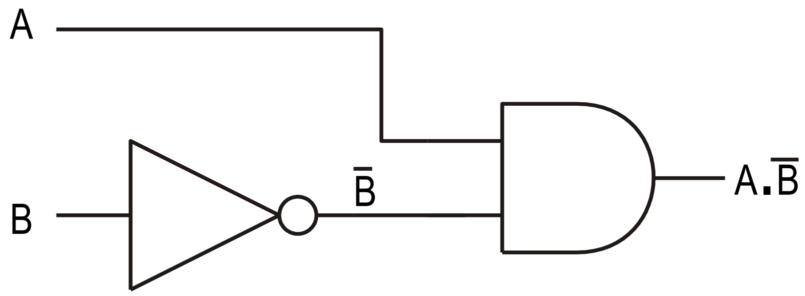
The output Z is at logic 1 in the third row down, and we can see that for this to happen A must be at logic 1 and B must be at logic 0. In other words

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.



<u>Example</u>

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

Row 0 1 1 1 → $Z = \overline{A}.B.C$

Row 1 1 0 1 → $Z = A.B.\overline{C}$

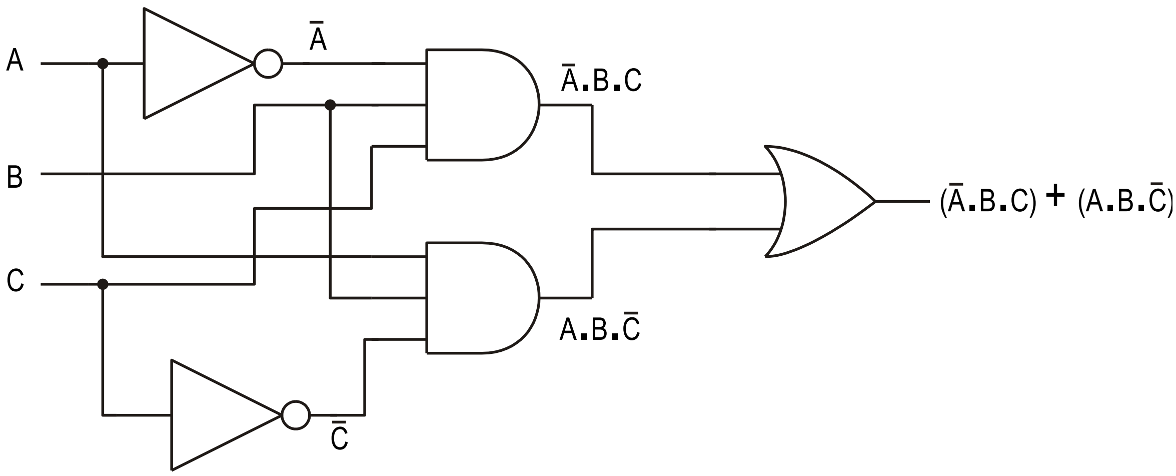
In other words, Z = 1 if

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

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

Task 7

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

http://goo.gl/1Lydl1

Task 8

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

Task 9

Design logic circuits to perform these Boolean equations

a) $Z = (A \oplus B) .C$

b) $Z = (A+B) . (C \oplus D)$

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

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

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

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

Task 10

A digital electronic system is needed to perform the following Boolean functions:

$Z = \overline{A \bullet B} + C$ $Y = \overline{A + B}$

a) Design a logic circuit to perform these requirements

b) Complete the truth table to show the expected outputs for each combination of inputs:

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

Task 10 (continued)

c) Construct the circuit you have designed on Croc clips or Yenka to ensure it works they way you want.
Attach a print out below

d) Build the circuit using a prototype board.
Attach a photo below

Task 10 (continued)

e) Describe how you tested the circuit, and record your results in the truth table.

inputs			expected outputs		results of tests	
A	B	C	Y	Z	Y	Z
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

f) Suggest at least one improvement that could be made to your circuit.

Task 11

A digital electronic system is needed to perform the following Boolean functions:

$$Z = A \oplus \overline{B + C} \qquad Y = A + \overline{C}$$

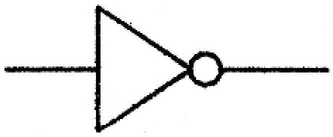
a) Design a logic circuit on Yenka or Croc Clips to perform these requirements. Attach printout of circuit below

b) Complete the truth table to show the expected outputs for each combination of inputs:

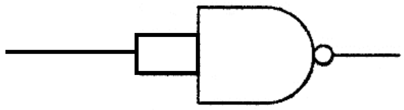
A	B	C					Y	Z
0	0	0						
0	0	1						
0	1	0						
0	1	1						
1	0	0						
1	0	1						
1	1	0						
1	1	1						

NAND Equivalent

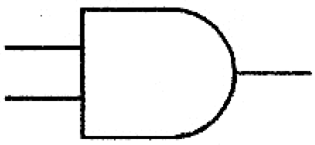
It is possible to make other logic gates (and circuits) by combining NAND gates. In practice, it is much easier and cheaper to make NAND gates than any others,



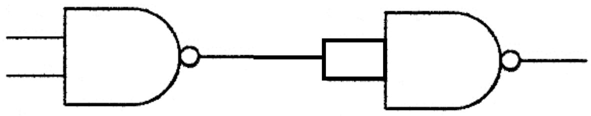
NOT GATE



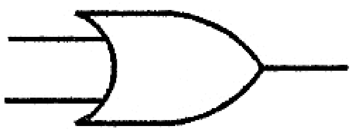
NOT GATE
NAND
EQUIVALENT



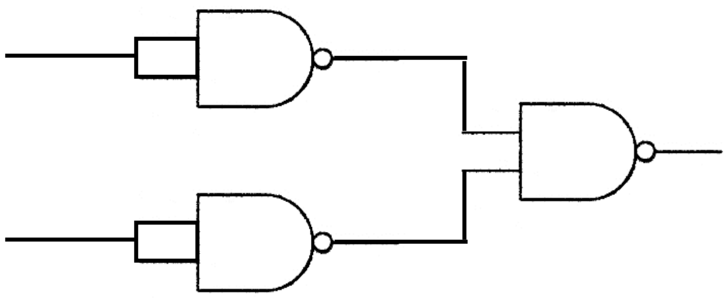
AND GATE



AND GATE
NAND
EQUIVALENT



OR GATE

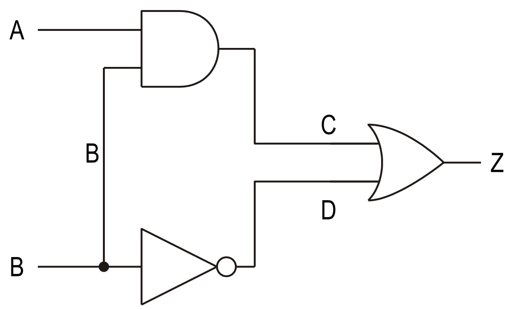


OR GATE
NAND
EQUIVALENT

Conversion to NAND Equivalent

Example

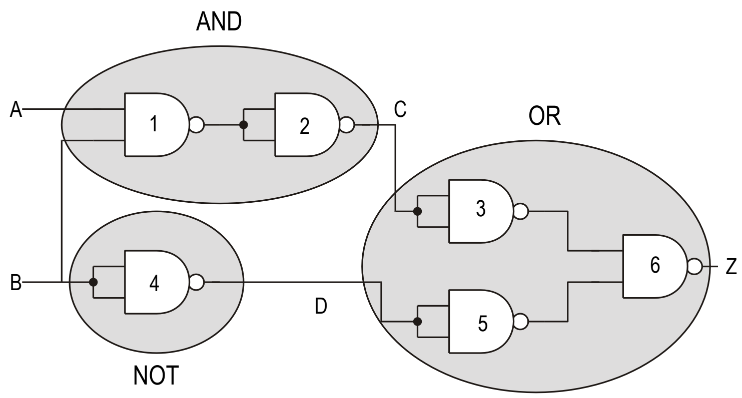
Consider the circuit shown



The system is made from a NOT gate an AND gate and an Or gate. The problem is to design a system with the same Truth Table, but made from NAND gates only.

Step 1

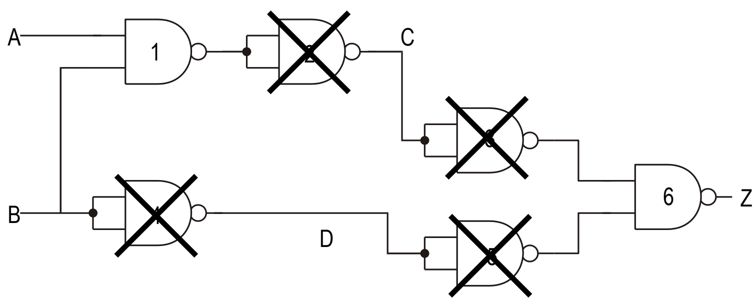
Redraw the circuit, replacing each gate with its NAND gate equivalent.



Step 2

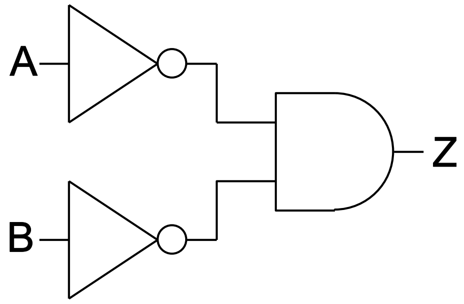
Examine the new arrangement and look for adjacent pairs of NOT equivalent gates. If you consider what happens when you feed a signal to a NOT gate then pass the signal on to another NOT gate you will find that the signal has been 'DOUBLE INVERTED' and cancels each other out.

In this circuit there are two such pairs. (2 & 3 and 4 & 5 are adjacent pairs). We must score these out.



Task 12

a) Write the Boolean expression that represents the circuit.



Z =

b) Complete the Truth Table for the circuit.

A	B			Z

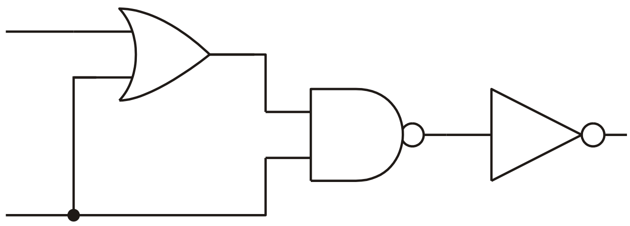
c) From the Truth Table name and draw the single gate that can replace the circuit.

d) From the circuit draw an equivalent logic circuit using only NAND Gates.

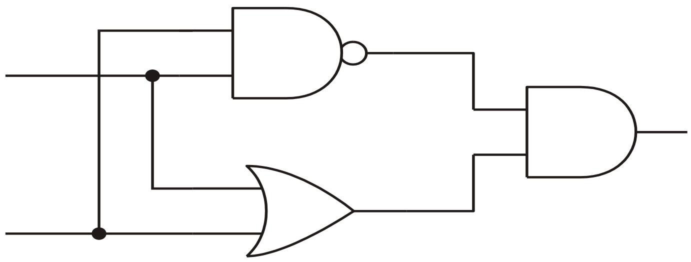
Task 13

Draw equivalent logic gates to the ones below using only NAND Gates, simplifying where possible.

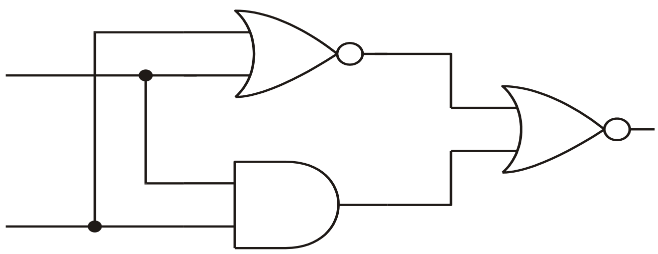
a)



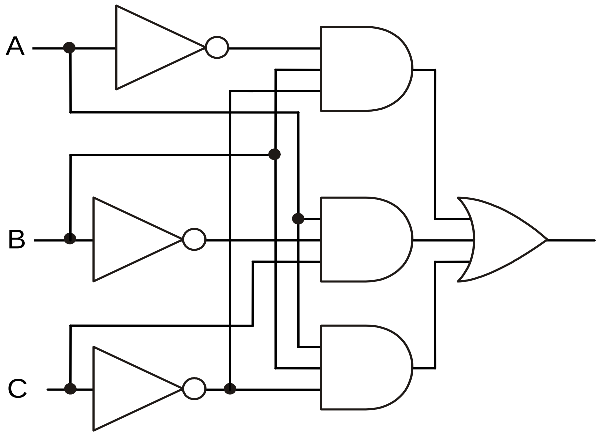
b)



c)



Task 14



a) Construct a truth table for the logic circuit shown and work out the Boolean Expression.

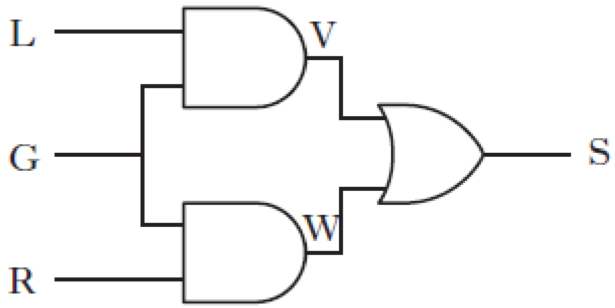
b) Redraw the circuit using NAND gates only, simplifying where necessary.

Task 15

The operation of an industrial stamping machine is controlled by a logic system with inputs and output as shown in the following table.

INPUTS	OUTPUT
L (= 1 when left-hand button is pressed)	S (= 1 to operate stamping machine)
R (= 1 when right-hand button is pressed)	
G (= 1 when guard is in position)	

The logic system shown ensures that the machine only operates when the guard is in place and either button is pressed.

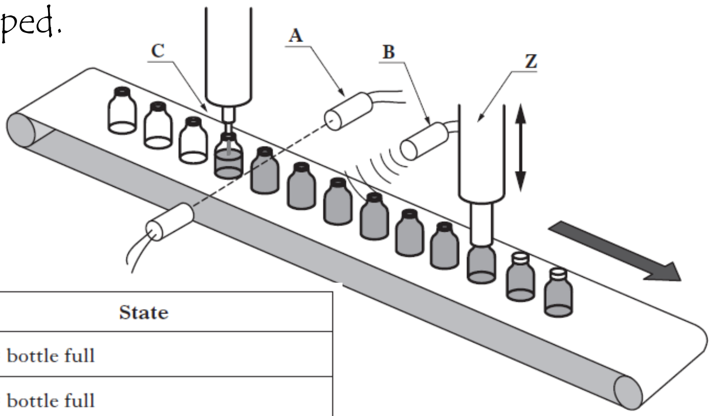


a) Write a Boolean expression for the output S in terms of the inputs L, G and R.

b) Using only two-input NAND gates, draw a logic system equivalent to the logic system shown. Delete any redundant gates.

Task 16

Below shows a production line for filling and capping bottles. A lid-fitting device (Z) operates when an ultrasound sensor (A) or a load cell (B) detects that the bottle is full, and a filling-nozzle sensor (C) detects that filling has stopped.



Transducer	State
Ultrasound sensor (A)	1 = bottle full
Load cell (B)	1 = bottle full
Filling-nozzle sensor (C)	0 = filling stopped
Lid-fitting device (Z)	1 = fit lid

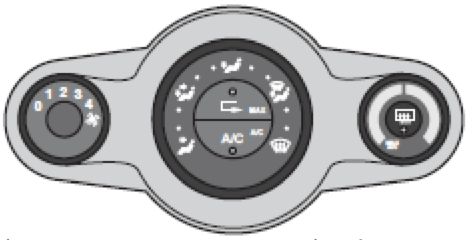
a) Draw the truth table for inputs A, B, C and output Z.

b) Write a Boolean expression for the output Z in terms of the three inputs A, B and C.

c) Draw the logic diagram for the output Z using only NAND gates. (2-input and 3-input NAND gates are available.)
Simplify where appropriate.

Task 17

The control panel for the climate-control system in a car is below.



A combinational-logic system controls the operation of a compressor (C). An air-conditioning select switch (A), a windscreen-demist switch (D), and a temperature sensor (T) provide input signals to the combinational-logic system.

A truth table for the system is shown below.

Air-conditioning Select (A)	Windscreen Demist (D)	Temperature Sensor (T)	Compressor (C)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

a) Write a Boolean equation for the compressor (C), in terms of A, D and T.

The compressor will operate only if the temperature sensed is above 5 °C.

b) State the logic value of the temperature sensor at 6 °C.

Task 17 (continued)

c) Draw a combinational-logic system to control the compressor using AND, OR and NOT gates.

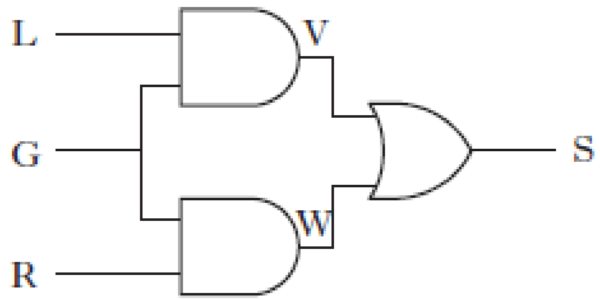
d) Draw an equivalent logic system using only NAND gates. Simplify where possible.

Task 18

The operation of an industrial stamping machine is controlled by a logic system with inputs and output as shown in the following table.

INPUTS	OUTPUT
L (= 1 when left-hand button is pressed)	S (= 1 to operate stamping machine)
R (= 1 when right-hand button is pressed)	
G (= 1 when guard is in position)	

The logic system shown below ensures that the machine only operates when the guard is in place and either button is pressed.

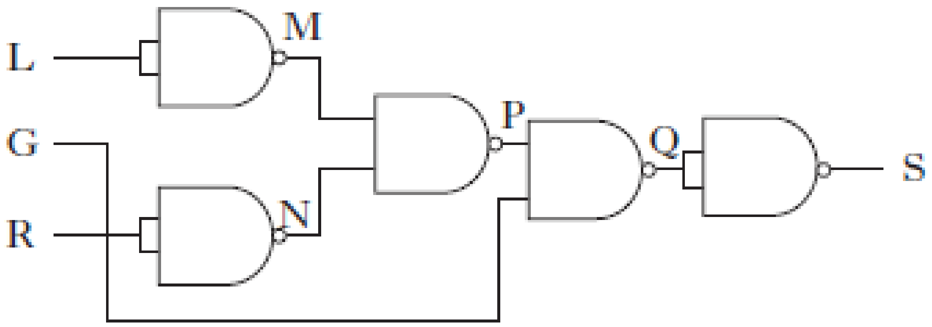


a) Write a Boolean expression for the output S in terms of the inputs L, G and R.

b) Using only two-input NAND gates, draw a logic system equivalent to the logic system shown in Figure Q1(*a*). Delete any redundant gates.

<u>Task 18 (continued)</u>

The logic system shown below was designed to provide the same logic function as the logic system before.



c) Draw and complete a truth table for the logic system above. Include the intermediate logic values M, N, P & Q.

d) Draw a logic system equivalent to this system using only two-input AND and OR gates.

Task 19

A logic system controls warning lights at the approach to the bridge shown

The input signals to the logic system are as follows:

A weight sensor (W) = 0 if a vehicle is too heavy for the bridge;
A height sensor (H) = 0 if a vehicle is too high for the bridge.

If a vehicle is either too heavy or too high, then the warning lights illuminate (L = 1).

a) Draw a truth table including the inputs W and H, and the output L.

b) State the name of the single logic gate which could provide the required logic function.

Task 19 (continued)

The system is modified by the addition of an override button (B):

B = 1 when the override button is pressed.

The output (L) from the redesigned logic system = 1 if the override button is pressed, or if a vehicle is either too heavy or too high.

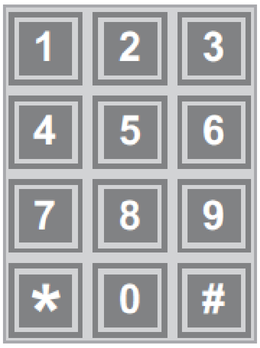
c) Write a Boolean expression for the redesigned logic system, in terms of W, H, B and L.

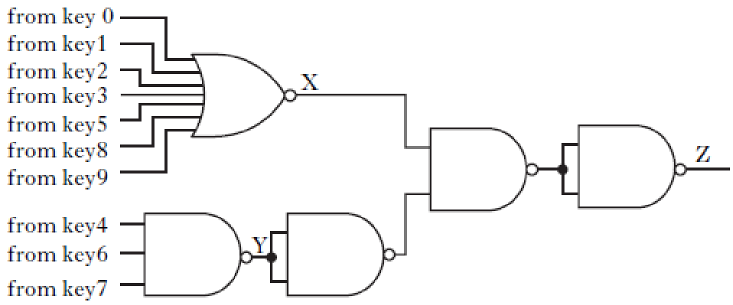
d) Draw a logic diagram for the redesigned logic system using NOT gates, AND gates and OR gates as appropriate.

e) Redraw the logic diagram in part (*d*) using only NAND gates. Simplify where appropriate.

Task 20

A security system has a digital keypad. A correct code must be entered on the keypad to open a door. The logic system shown below processes the signals from the keys 0–9. When a key is pressed a **high** signal is placed on an input line to the logic system, otherwise the signal on the input line is **low**.





a) **(i)** Draw a truth table with signals X and Y as inputs, and signal Z as the output.

(ii) From the truth table, write a Boolean expression for the output Z in terms of X and Y.

b) Explain what a user must do, in order to cause a high signal at Z.

A high signal at Z opens the door; a low signal at Z closes the door.

c) State **two** disadvantages of this security system.

Task 21

A combinational logic system controls the motor of a lazer cutter (M) according to the Boolean expression

$$M = Z.\overline{Y}.\overline{X} + Z.Y.\overline{X}$$

- The cutter motor is on when M = 1
- The stepper motor moves the cutter in the *x*-axis when X = 1
- The cylinder which moves the cutter in the *z*-axis outstrokes when Z = 1
- The cylinder which moves the cutter in the *y*-axis outstrokes when Y = 1

a) Complete the truth table for M in terms of Z, Y and X.

Z	Y	X	M
0	0	0	
0	0	1	

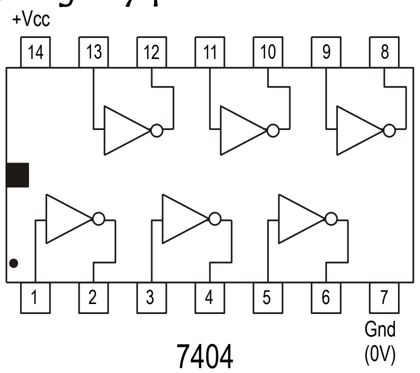
b) State which of the signals X, Y and Z does not affect the state of M.

<u>Building Circuits with IC Chips</u>

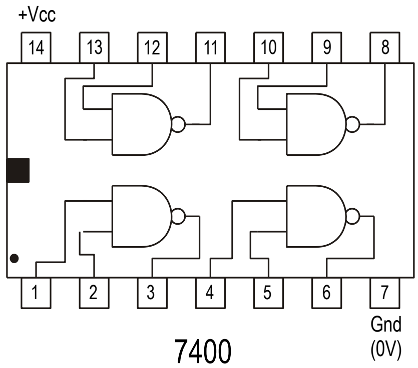
Although the chip contains complex circuitry, the internal wiring can be shown as simple logic circuits with the inputs and outputs of each logic gate shown. This is called a *pin-out diagram*. These are shown at the bottom. Your Data book also lists the most common IC's so you can use this during tests and exams.

Pin 14 is connected to the 5-volt stable supply (power supply) and pin 7 to 0 volts (ground line). This must ALWAYS be done to ensure the chip is powered.

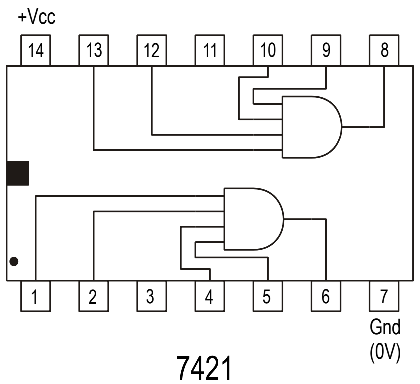
You do not need to know this for the exam, but it will help if you are doing any practical work.



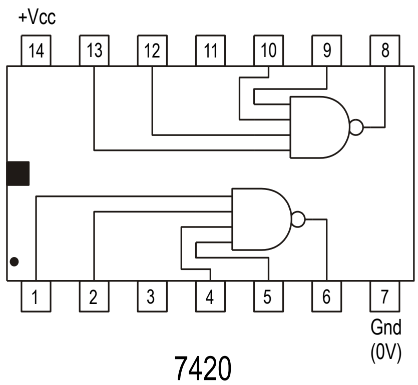
7404



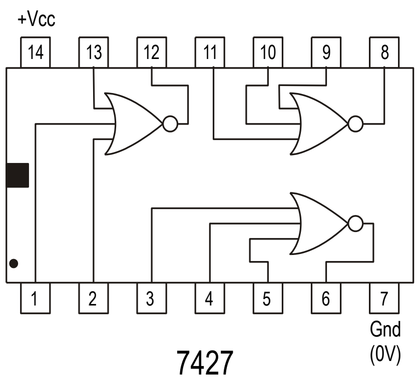
7400



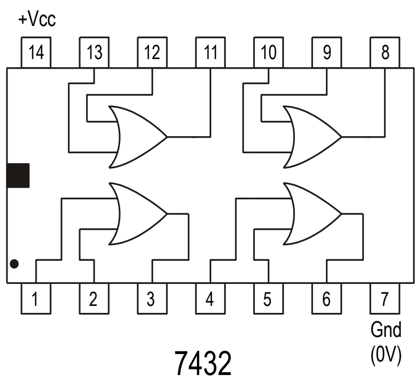
7421



7420



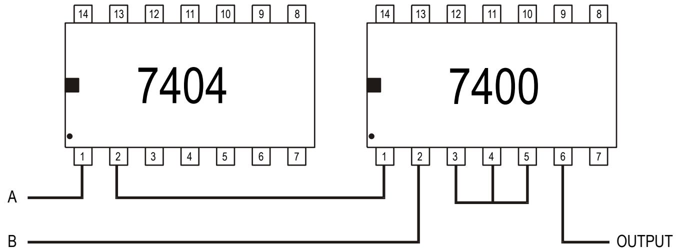
7427



7432

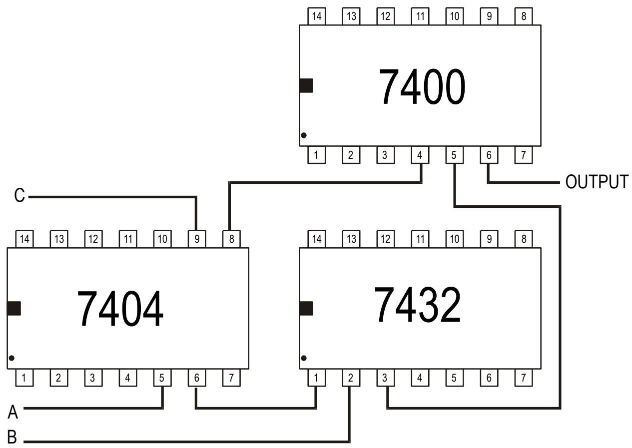
Task 22

a)(i) Draw the corresponding logic diagram for the circuit shown below.



ii) Write out the Boolean Expression of this circuit

b)(i)Draw the corresponding logic diagram for the circuit shown below.

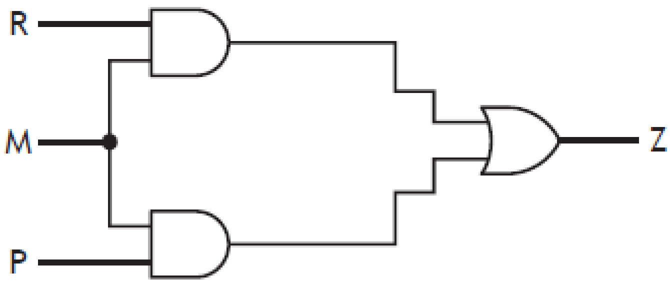


(ii) Write out the Boolean Expression of this circuit

<u>Task 23</u>

A garage shutter door can be operated via a remote control (R) or a push-button (P). Neither will operate if a main switch (M) is turned off.

The logic diagram below shows the control of the garage shutter door.



a) Write a Boolean expression for the logic diagram above.

Z =

(b) Using only two-input NAND gates, draw a logic system equivalent to the logic system shown above. Simplify if appropriate.