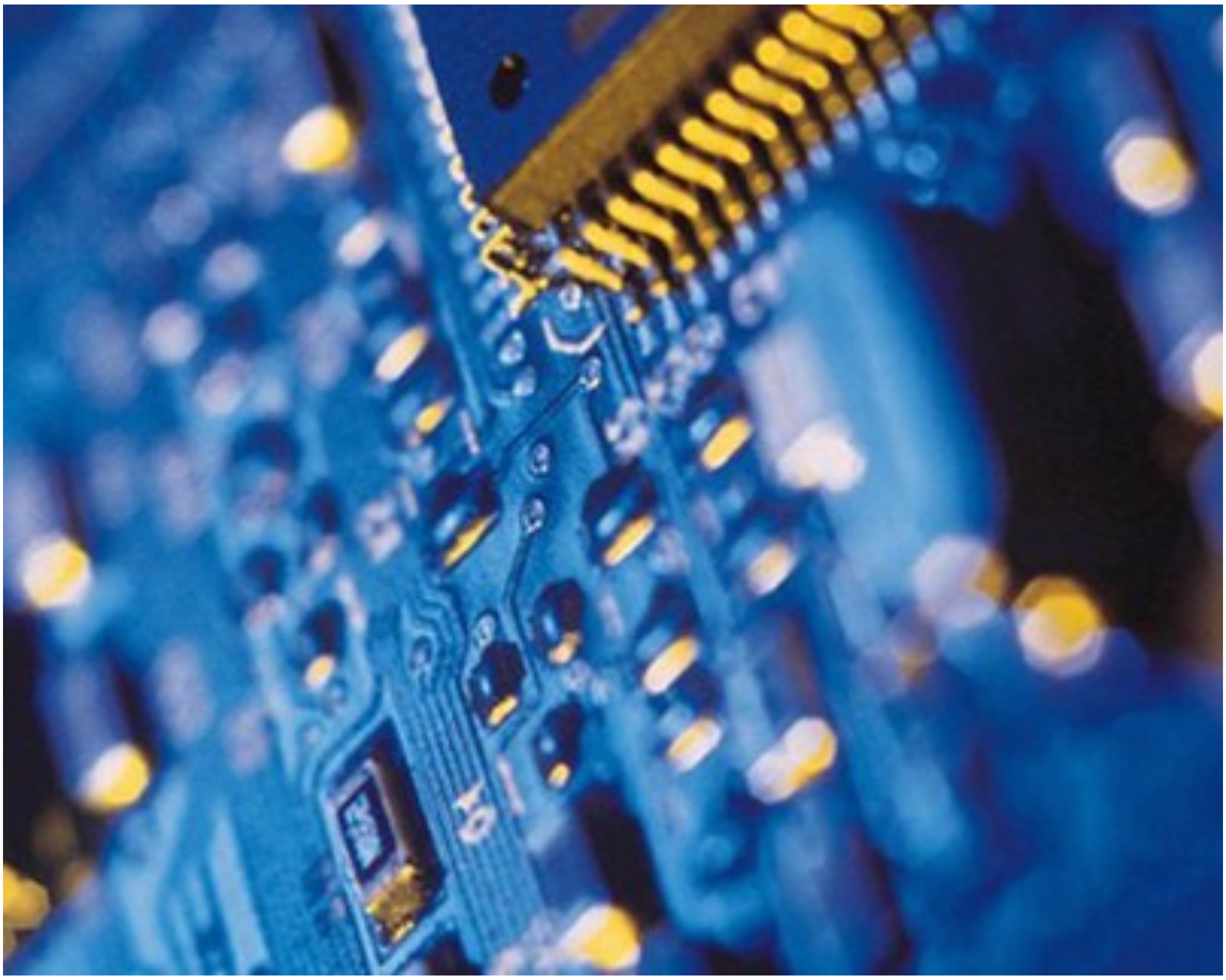
Higher Engineering Science

Electronics and Control

Book 1 of 3 - Digital Electronics



Name: Class:

Introduction

This course follows on from your studies in National 5 Engineering Science. If you have not previously studied Engineering Science it is essential that you study the online materials at www.technologyinthemearns.net . Specifically, this unit of work continues on from learning in the following units:

- Systems
- Electronics

By the end of this topic you will be expected to meet the following success criteria:

2 - Develop digital electronic control systems by:

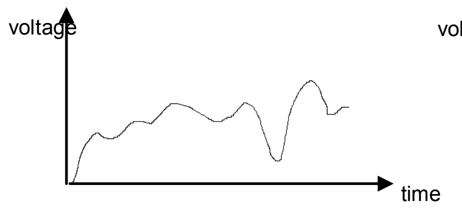
- **2.1 - Designing and constructing complex combinational logic circuits**
- **2.2 - Describing logic functions using Boolean operators**
- **2.3 - Simplifying logic circuits using NAND equivalents**
- **2.4 - Testing and evaluating combinational logic circuits against a specification**

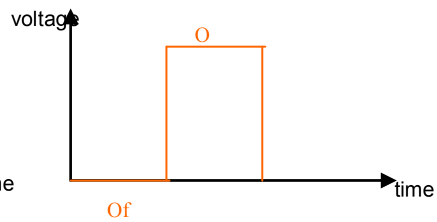
Before You Start...

Digital electronic circuits use logic gates to make decisions. Input signals, which often have to be converted from analogue to digital, are fed into logic circuits which then instruct an output device or devices to switch on or off.

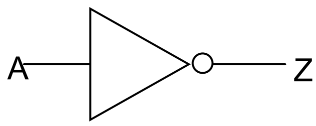
It is assumed from your studies at National 5 you will already have an understanding of:

- The differences between analogue and digital signals





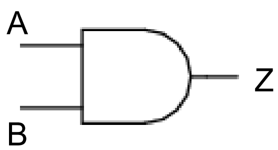
- NOT gate



Z =

Input A	Output Z
0	
1	

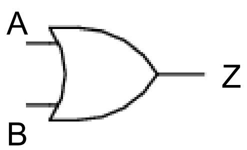
- AND gate



Z =

Input A	Input B	Output Z
0	0	
0	1	
1	0	
1	1	

- OR gate



Z =

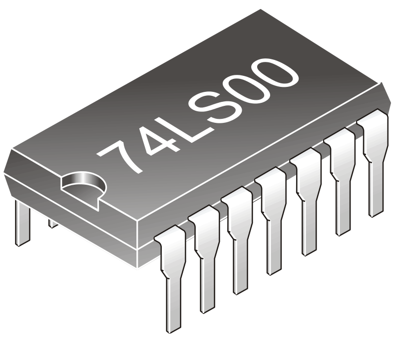
Input A	Input B	Output Z
0	0	
0	1	
1	0	
1	1	

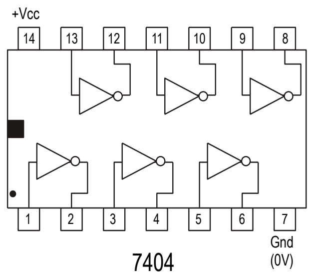
Useful Information about ICs

An IC is an Integrated Circuit—essentially a chip that has multiple components built into it using semiconductor technology.

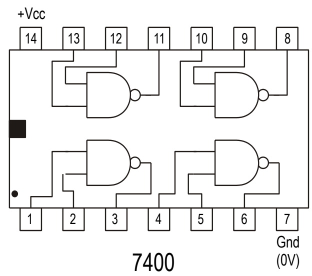
Logic gates are not available as discreet components, rather multiple logic gates are built into a single IC. This can reduce the cost of manufacturing circuits as one IC can be used to provide multiple logic gates. There are 2 main types of ICs, each which have their advantages and disadvantages. Although we will be using TTL chips it is useful to know the comparisons between TTL and the other main family of ICs, CMOS:

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

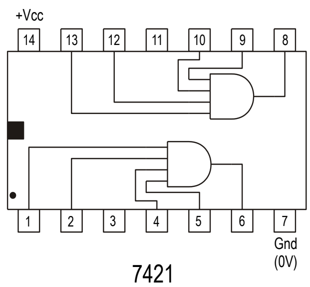




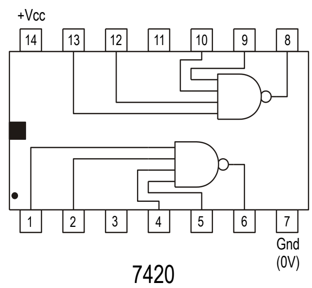
7404



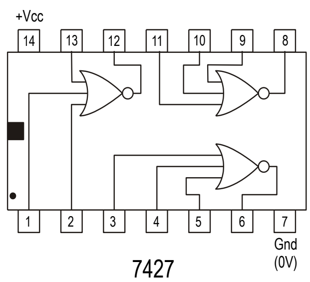
7400



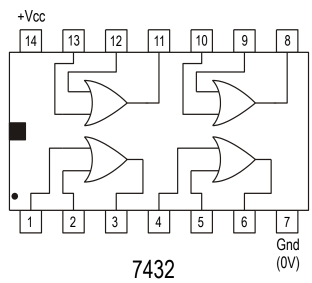
7421



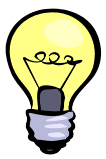
7420



7427



7432

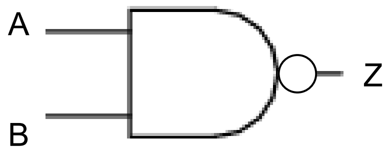


The pin out diagrams on the right show how TTL chips are constructed to provide multiple logic gates on a single chip.

The NAND 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 sometimes 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. A useful way to produce the truth table is to first write out the truth table for an AND gate and then invert (reverse) the results, thus giving you the outputs for the NAND gate.



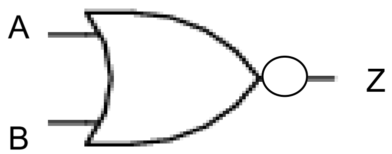
Z =

Input A	Input B	AND	Output Z
0	0		
0	1		
1	0		
1	1		

The NOR Gate

Similarly, the NOR gate is effectively an *inverted* OR gate. In other words, the results obtained from the output are the opposite to those of the OR gate. This gate is sometimes referred to as 'NOT OR'.

When drawing up the truth table for the NOR gate it can be difficult to 'picture' or imagine the results. A useful way to produce the truth table is to first write out the truth table for an OR gate and then invert (reverse) the results, thus giving you the outputs for the NOR gate.



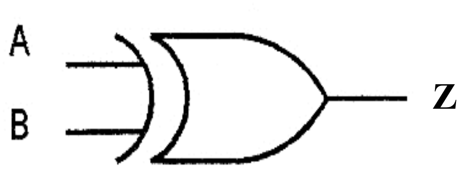
Z =

Input A	Input B	OR	Output Z
0	0		
0	1		
1	0		
1	1		

The XOR Gate

If you look closely at the truth table for an OR gate you will see that in the bottom row, when both inputs are on, the output is also on. In this way, when both inputs are on the OR gate behaves in the same way as an AND gate.

The XOR gate behaves differently—it will only switch on when one input OR the other input is on, but not when both are on. For this reason the full name for this gate is the ‘EXCLUSIVE OR’ gate, often shortened to XOR or EOR.



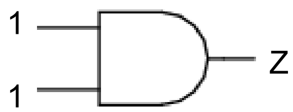
Z =

Input A	Input B	Output Z
0	0	
0	1	
1	0	
1	1	

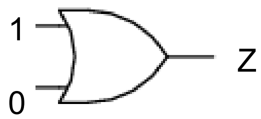
Assignment 1 - Basic Logic Gates

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

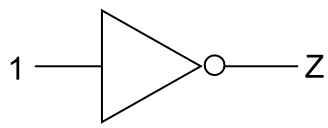
(a)



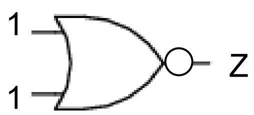
(b)



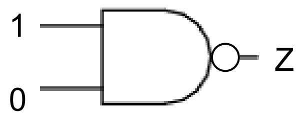
(c)



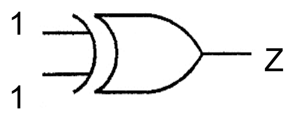
(d)



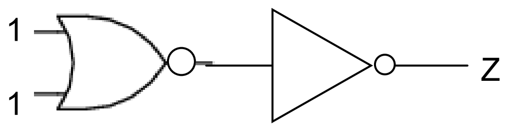
(e)



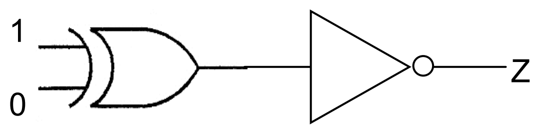
(f)



(g)

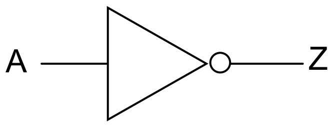
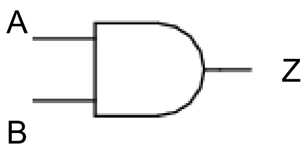
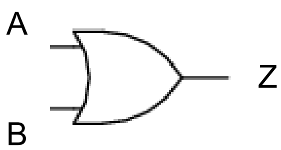
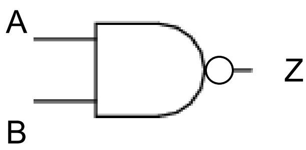
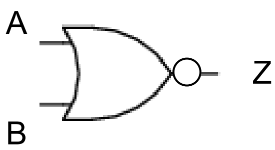
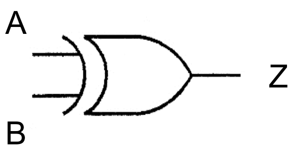


(h)



Boolean Algebra and Boolean Expressions

Boolean algebra, developed by British mathematician George Boole, is used in digital electronics to mathematically describe how a circuit will behave—each logic gate has it's own Boolean expression. By applying this knowledge to construct the Boolean expression for a complex circuit it is often possible to then simplify the expression in exactly the same way as you would with normal algebra, and thus simplify the overall circuit design.

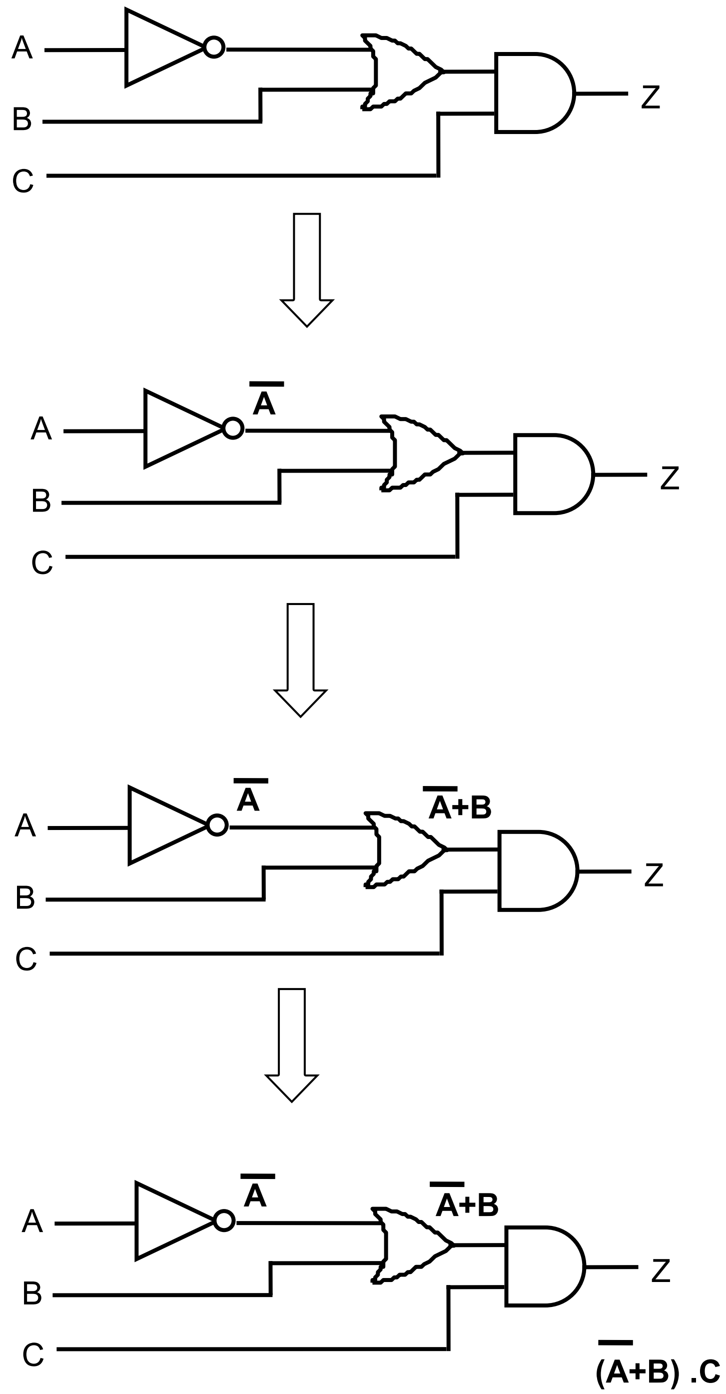
Symbol	**Gate**	**Boolean Expression**
	NOT	$Z = \overline{A}$
	AND	$Z = A.B$
	OR	$Z = A+B$
	NAND	$Z = \overline{A.B}$
	NOR	$Z = \overline{A+B}$
	XOR	$Z = A \oplus B$

Creating a Combinational Boolean Expression

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 form an equation that describes how the circuit will behave:

To create a Boolean expression for this circuit, first create a Boolean expression for the output from each logic gate.

These can then be combined through the final logic gate to give the final Boolean expression for the circuit.



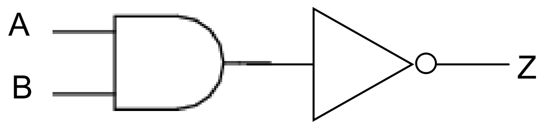
The final Boolean expression for the circuit is: $Z = (\overline{A}+B) \cdot C$

Z = (NOT A OR B) AND C

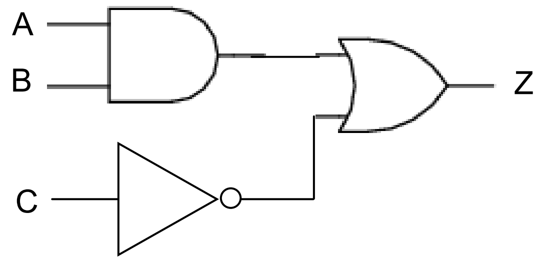
Assignment 2 - Combinational Boolean

Copy each of these circuits into your jotter, and work out the Boolean expression for each.

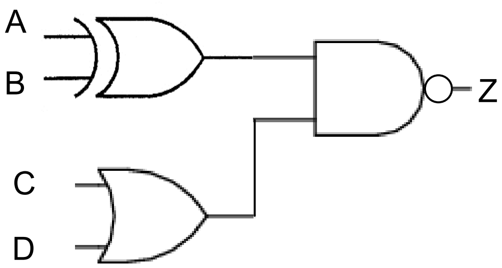
(a)



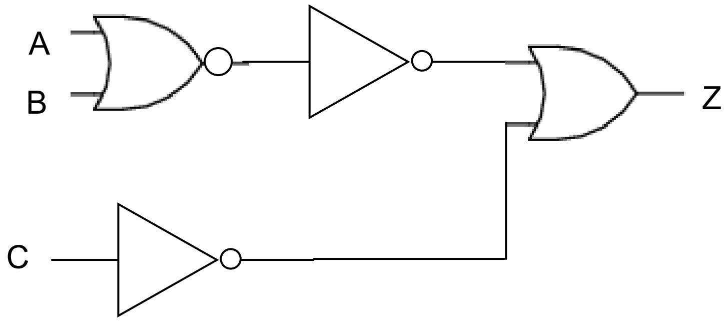
(b)



(c)



(d)



Creating a Logic Diagram From a Truth Table

With a knowledge of Boolean Algebra it is possible to design a circuit from a truth table. In Designing a complex system it is common to first identify all input signals, then decide which combinations of input signals will cause the output to turn on. Filling this information into a truth table then allows an electronic design engineer to develop a Boolean expression, and finally design the circuit.

Step 1: Identify the conditions when the output is on

Step 2: Complete a Boolean expression for each condition

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

Step 3: Combine these to complete the Boolean expression for the circuit

Step 4: Create the logic diagram

A —

B — — Z

C —

Assignment 3 - Truth Tables, Boolean, and Logic Circuits

In your jotter, write out the Boolean expression and create the logic diagram for each of 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

(d)

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

(e)

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

(f)

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

Assignment 4 - More Logic and Boolean

In your jotter, design logic circuits to perform these Boolean expressions. You can check your answers using the Yenka circuit simulation software.

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

(g) A digital electronic system is needed to perform the following functions:

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

(i) Design a logic circuit to meet these requirements
(ii) Simulate your design and complete a truth table describing how it was tested

Inputs			Expected results		Test results	
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				

(h) A digital electronic system is needed to perform the following functions:

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

(i) Design a logic circuit to meet these requirements
(ii) Simulate your design and complete a truth table describing how it was tested

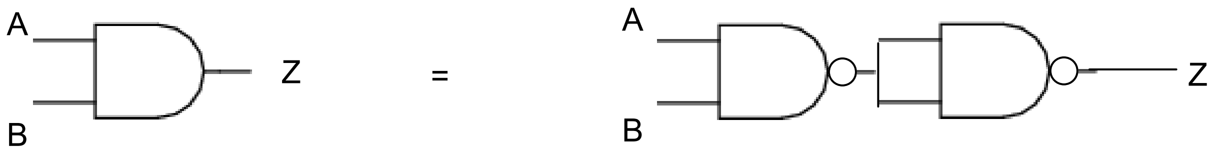
NAND Equivalent Circuits

It is possible to construct each logic gate—NOT, AND, OR, NAND, NOR, XOR entirely out of NAND gates. Whilst this may appear to make logic circuits look much more complicated this is of huge benefit to electronics manufacturers. Why?

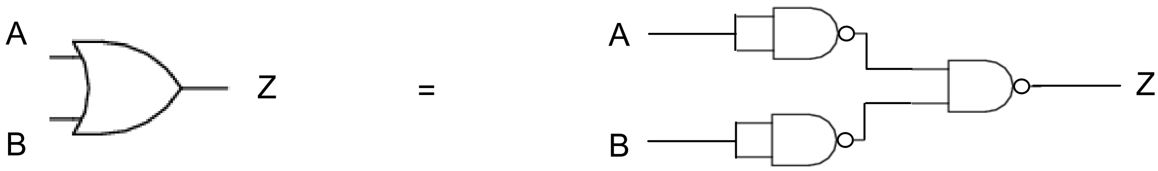
- Only 1 type of IC needs to be purchased to produce any given circuit
- Less wastage, as there are far less unused gates on each IC
- Buying NAND gate ICs in bulk greatly reduces the unit cost

NOT

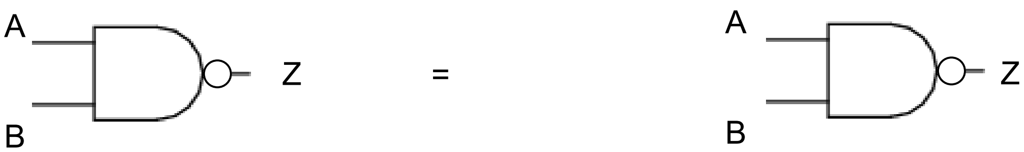
AND



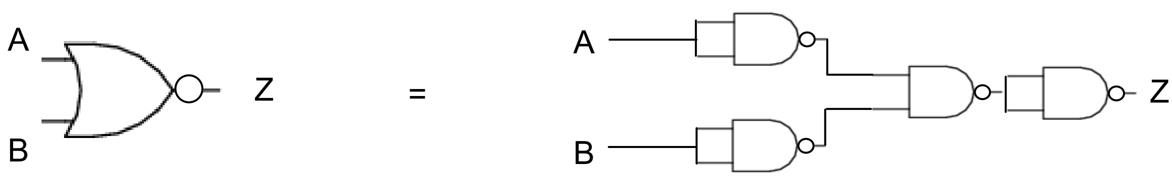
OR



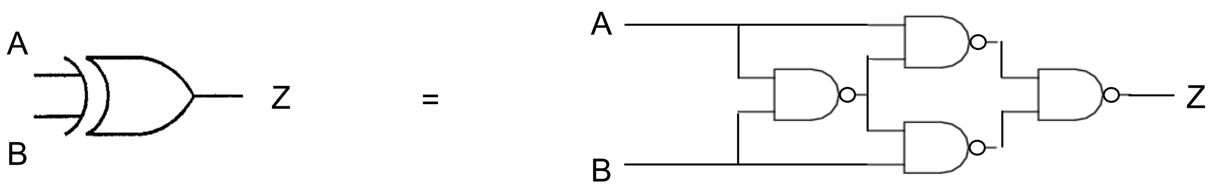
NAND



NOR

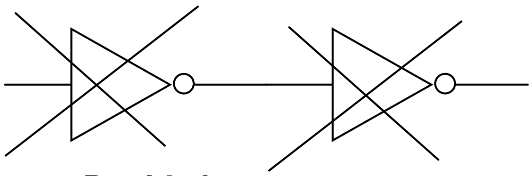


XOR

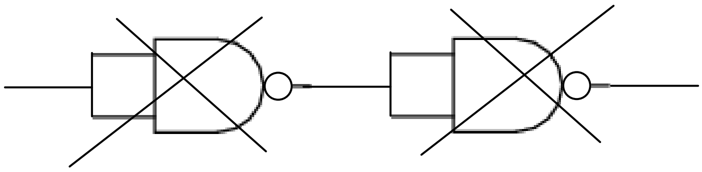


Converting to NAND Equivalents and Simplifying

Another benefit of converting circuits to their NAND equivalents is it often means that the circuit itself can then be simplified. The simple rule to apply here is where you identify 2 NOT gates in succession, these gates can be cancelled out:

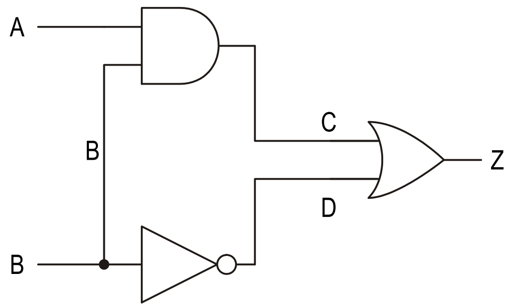


Double inverters can be cancelled out

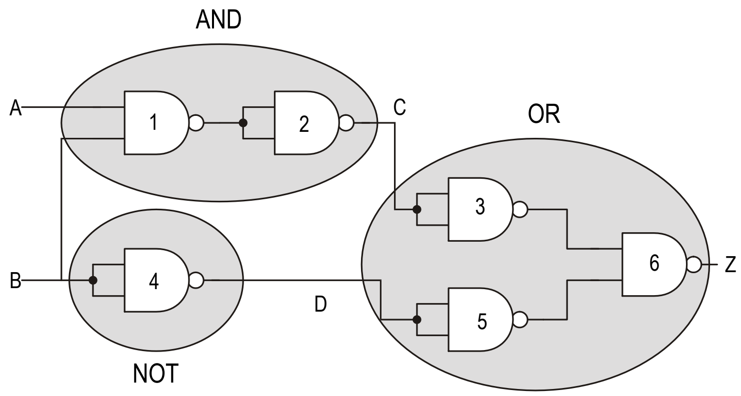


NAND equivalent double inverters can be cancelled out

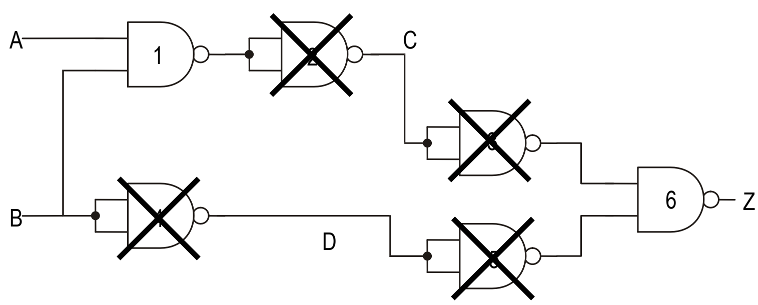
Consider the circuit shown to the right. It is constructed from NOT, AND and OR gates.



The circuit can be redrawn using NAND equivalents

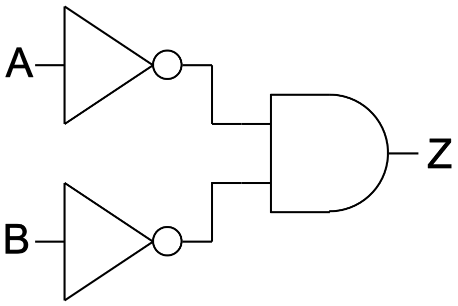


Where 2 NOT gates follow in succession these are crossed out. The circuit can then be redrawn to show it in it's simplified version. It will behave in exactly the same manner.



Assignment 5 - NAND Equivalents

1. (a) Write the Boolean expression that represents this circuit.



A	B			Z
0	0			
0	1			
1	0			
1	1			

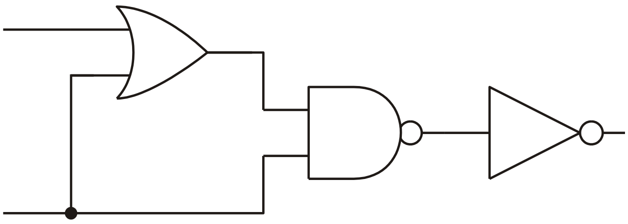
(b) Complete the truth table for the circuit

(c) From the truth table, identify the single gate that can replace the circuit

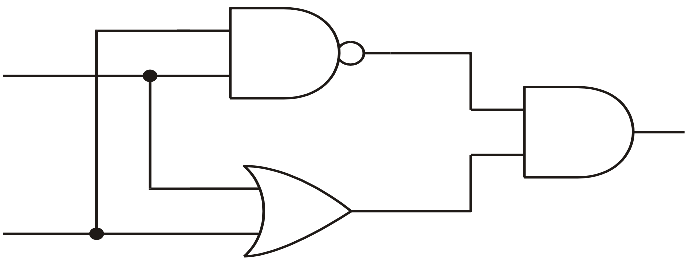
(d) Draw the equivalent NAND gate circuit

2. Draw equivalent NAND gate circuits for each of the following. You should simplify the circuits wherever possible.

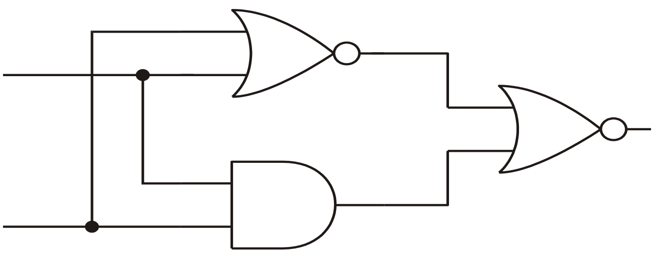
(a)



(b)

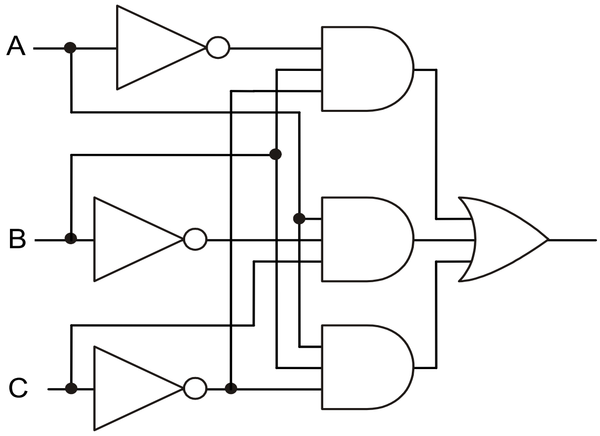


(c)



Assignment 6 - Complex Tasks

1. (a) Construct a truth table for the logic circuit shown, and from this work out the Boolean expression

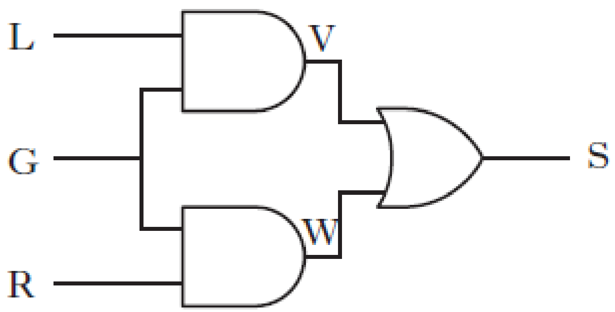


(b) Redraw the circuit using NAND gates, simplifying where possible

2. The operation of an industrial stamping machine is controlled by a logic system with inputs and outputs 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)	

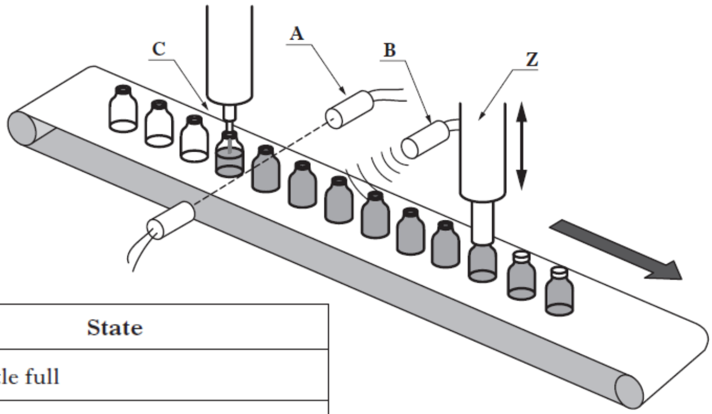
The logic system shown below ensures that the machine only operates when a safety guard is in place and either button is pressed.



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

(b) Using only 2-input NAND gates, draw a logic diagram to represent the circuit. Delete and redundant gates.

3. 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 the Boolean expression for the output Z in terms of inputs A, B and C

(c) Draw the logic diagram for the output Z using only NAND gates. Simplify where possible

4. A combinational logic system is used in the climate control system in a car. The air-conditioning select switch (A), windscreen demist switch (D) and temperature sensor (T) provide input signals which control the operation of the compressor (C)

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

(b) Draw a combinational logic system to control the compressor using NOT, AND and OR gates

(c) Redraw the circuit using only NAND gates. Simplify where possible

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

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) Which single logic gate could be used to provide the required function?

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, AND and OR gates

(e) Redraw the logic diagram using only NAND gates. Simplify where possible

6. A combinational logic system controls the motor on a laser cutter (M) according to the Boolean expression below:

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

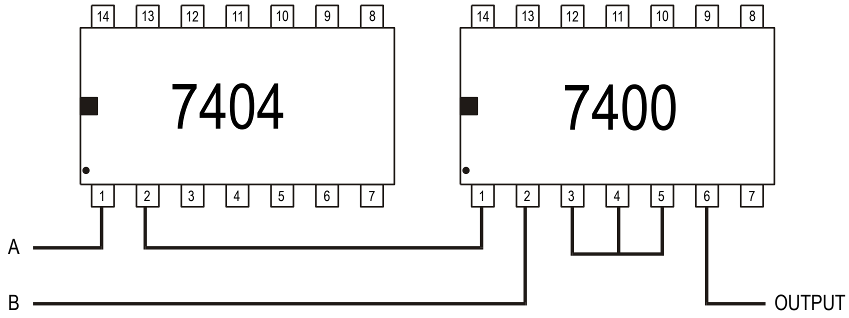
- The cutter motor is on when M = 1
- The stepper motor moves the cutter in the x-axis when X = 1
- The cylinder which moves the cutter in the z-axis outstrokes when Z = 1
- The cylinder which moves the cutter in the y-axis outstrokes when Y = 1

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

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

Assignment 7– Practical Circuits

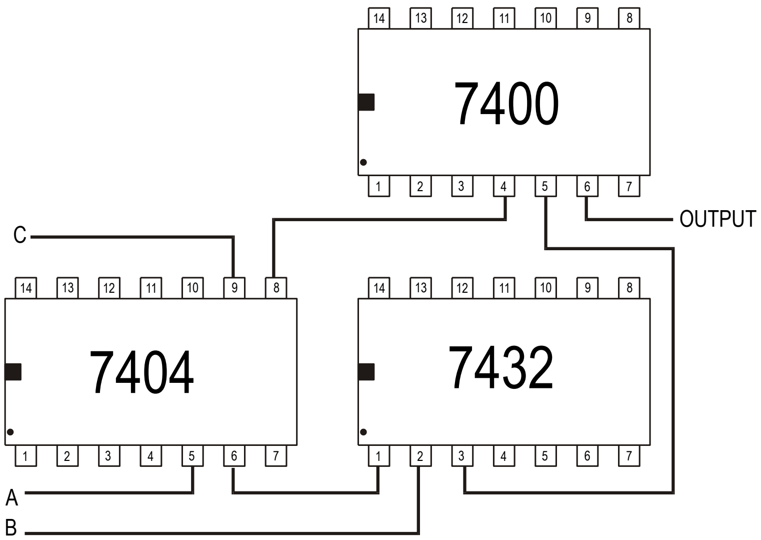
1. (a) Draw the corresponding logic diagram for the circuit shown below:



(b) Write out the Boolean expression for this circuit

(c) Build and test the circuit on breadboard

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



(b) Write out the Boolean expression for this circuit

(c) Build and test the circuit on breadboard

3. A proposal for the control of the automatic doors (Z), safety light (S) and a warning light (W) on the Edinburgh tram must meet the following criteria:

- The doors must open when the two sensors (A and B) both go high to indicate that the vehicle has reached the correct position
- An emergency override switch © must also open the door, when it goes high
- The doors must not open when the vehicle is still moving (M = 1 when the vehicle is moving)

The doors open when Z = 1

(a) Complete a Boolean expression for the door to open (Z =)

In addition to the above:

- The safety light must be on when the doors are open
- The warning light must be on at all other times

(b) Simulate the full electronic system using Yenka.

(c) Convert your circuit design to it's NAND equivalent and simplify where possible.

(d) Test your circuit and record the results in the truth table shown below

A	B	C	M	Expected result			Test result		
				Z	S	W	Z	S	W
0	0	0	0	0	0	1			
0	0	0	1	0	0	1			
0	0	1	0	1	1	0			
0	0	1	1	0	0	1			
0	1	0	0	0	0	1			
0	1	0	1	0	0	1			
0	1	1	0	1	1	0			
0	1	1	1	0	0	1			
1	0	0	0	0	0	1			
1	0	0	1	0	0	1			
1	0	1	0	1	1	0			
1	0	1	1	0	0	1			
1	1	0	0	1	1	0			
1	1	0	1	0	0	1			
1	1	1	0	1	1	0			
1	1	1	1	0	0	1			