Engineering Science

Electronics & Control

Digital Electronics

Introduction

Electronic circuits can be use to control a huge variety of systems but in each case there are INPUTS, PROCESSES and OUTPUTS.

In this next section we are going to focus on some of the components that form the PROCESSES.

The ones we will look at are know as ***digital logic gates***.

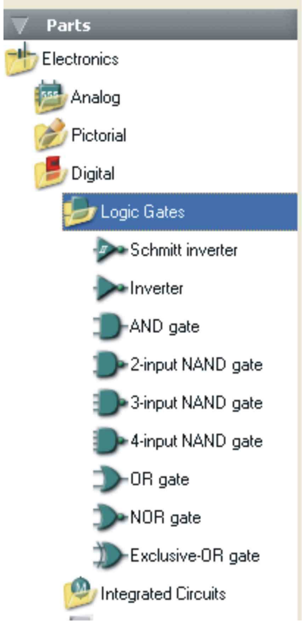
Digital means a device can either be on or off. We often refer to this as '1' or '0' or 'High' and 'Low'

1 = On = High

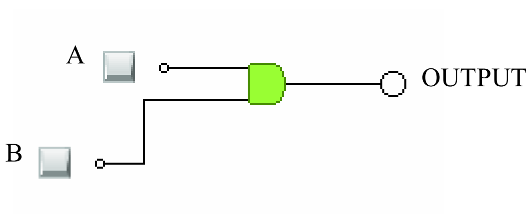
0 = Off = Low

Crocodile Technology can be used to simulate digital circuits.

Use the mouse to select the ***Logic Gates*** option in the folders within Crocodile Technology as shown below.



Build the following circuit using an ***AND*** gate and fill in the truth table below.



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

When is the Output ='1' or 'ON' ? Write the answer below.

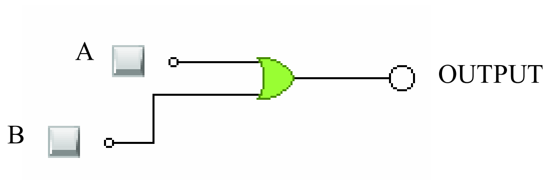
____

Outside the classroom, can you think where this could be used ?

____

____

Build the following circuit using an '***OR***' gate and fill in the truth table below.

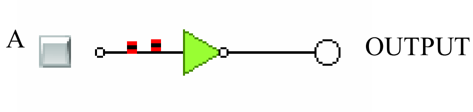


Input A	Input B	Output
0	0	
0	1	
1	0	
1	1	

When is the Output ='1' or 'ON' ? Write the answer below.

Outside the classroom, can you think where this could be used ?

Build the following circuit using an '***INVERTER***' gate (also known as a '***NOT***' gate) and fill in the truth table below.



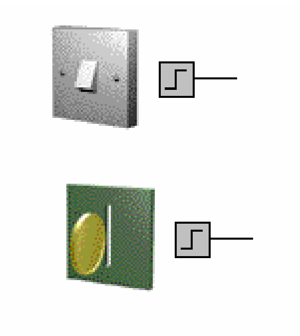
Input A	Output
0	
1	

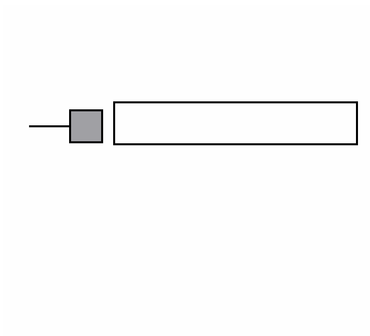
When is the Output ='1' or 'ON' ? Write the answer below.

Outside the classroom, can you think where this could be used ?

Problem 1

A video game at an amusement park can be played only when
(a) the game is switched on AND
(b) the money has been inserted in the game.
Build a digital circuit to satisfy the above specification.

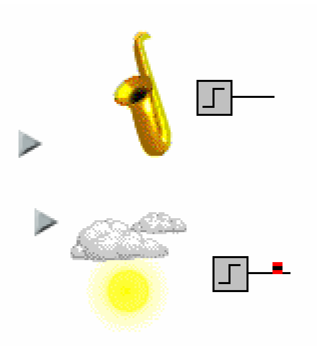


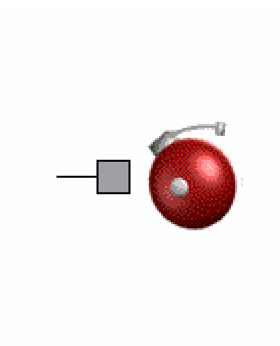


Teacher's Signature

Problem 2

A house owner in a Neighbourhood Watch Area installed an alarm system
(a) to detect noisy neighbours OR
(b) to detect if any lights had been left on OR
(c) to detect (a) and (b).
Build a digital circuit to satisfy the above specification.





Teacher's Signature

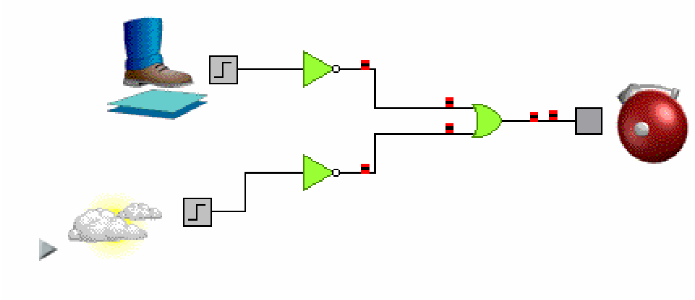
Problem 3

A central heating system will go on when the temperature outside is NOT hot .
Build a digital circuit to satisfy the above specification.

Teacher's Signature

4. From the circuit below fill in the Truth table to satisfy the following conditions :
(a) when there is **no** pressure applied to a pressure sensor an alarm should sound
(b) when a light sensor senses it is dark an alarm should sound
(c) when (a) OR (b) happen an alarm should sound

Check your answer by building the circuit in Crocodile Technology.



Pressure Sensor	Light Sensor	Alarm
0	0	
0	1	
1	0	
1	1	

Teacher's Signature

5. Build the circuit in Crocodile Technology to satisfy the following conditions :
(a) when the temperature in the room is cold a heater will go 'ON'
(b) every time a switch is pushed 'ON' a heater will go 'ON'
(c) Both (a) and (b) should be satisfied for the system to work properly

Draw the completed circuit and fill in the Truth Table below.

Temperature	Switch	Heater
0	0	
0	1	
1	0	
1	1	

Teacher's Signature

6. Build the circuit in Crocodile Technology to satisfy the following Truth Table.

Temperature Sensor	Light Sensor	Moisture Sensor	Alarm
0	0	0	1
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	1

Draw the completed circuit below.

Teacher's Signature

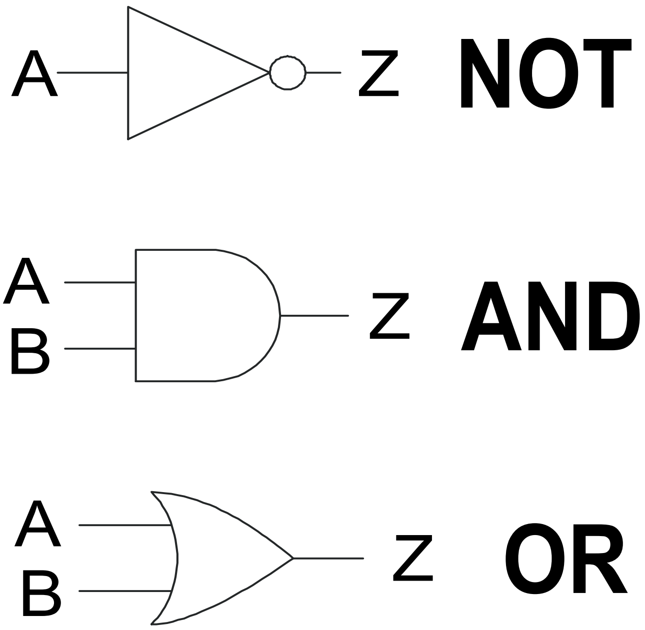
SWITCHING LOGIC

Making decisions

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

Logic gates are very useful 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 previous work you have already seen NOT, AND and OR logic gates.

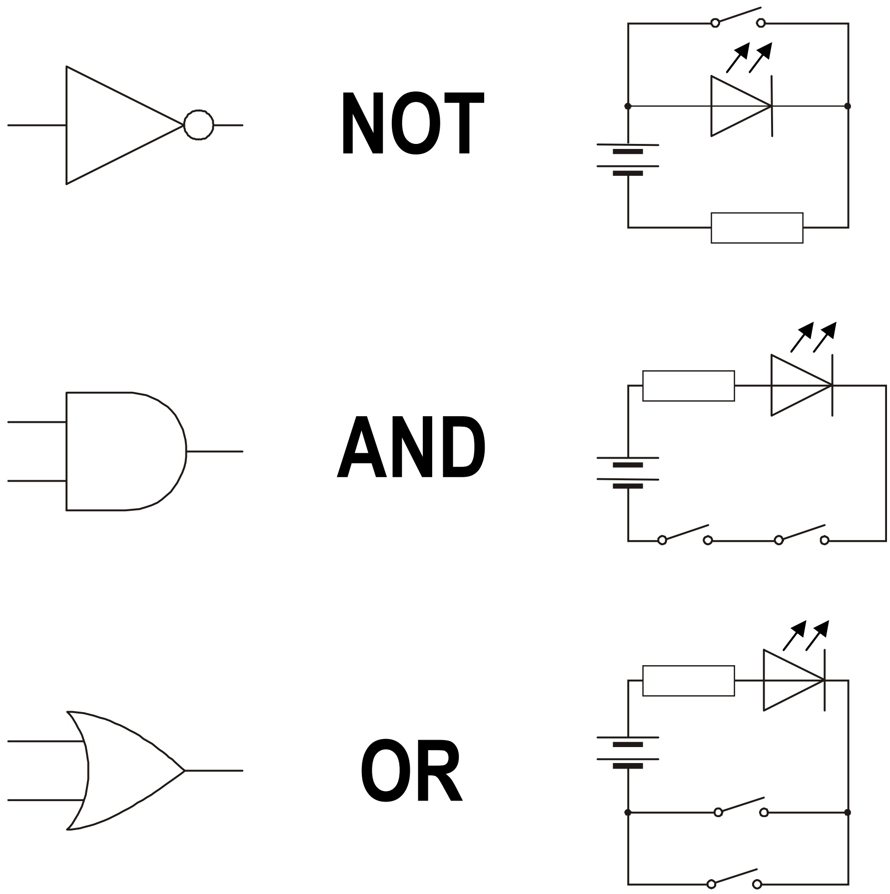


It is worth remembering that logic gates are part of digital systems and, as such, respond to either logic 1 or logic 0 signals only.

Logic in simple component circuits

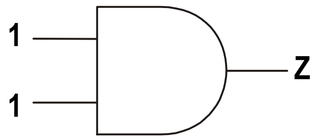
As we will see later in the course, digital logic can also be seen in simple component circuits as well as in pneumatic, hydraulic and other systems.

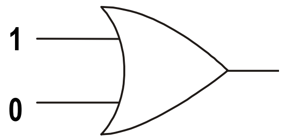
The circuits below show the three main types of logic.

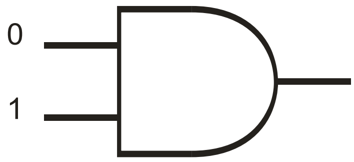


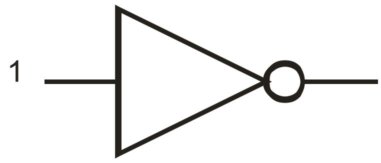
Logic gate exercises

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

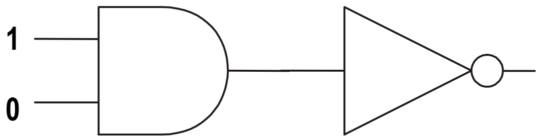


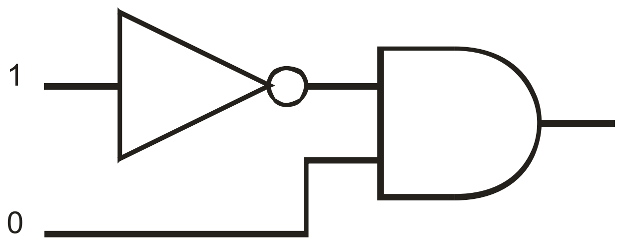






)

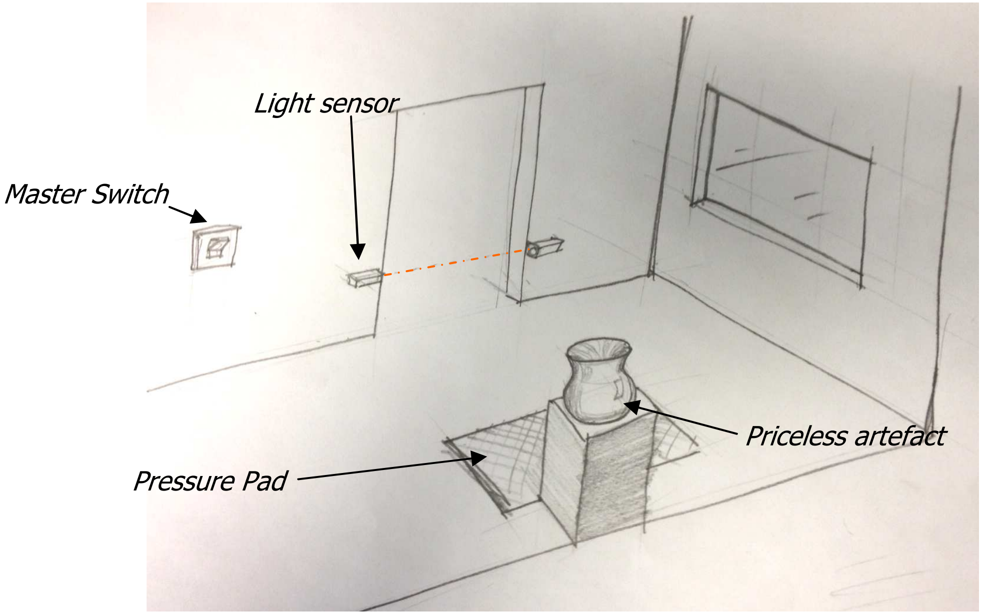




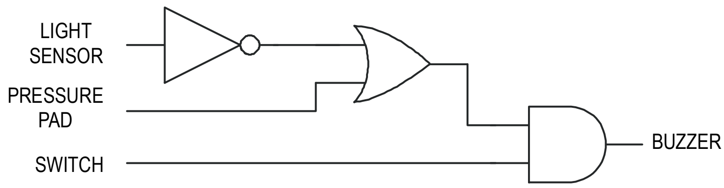
Combinational logic

So far in this unit of work we have only looked at simple logic systems on their own. In reality, most logic systems use a combination of different types of logic gates in one system. This type of logic control is known as *combinational logic*.

For example, the burglar alarm system for a museum artefact shown below will need a number of different logic gates in it.



We can draw a logic diagram of this system, as shown below. There is more than one logic gate in this system and so it is known as a *combinational logic diagram*.



Questions

This system could be used as a burglar alarm.
1. What is the purpose of the AND gate?
2. Why is the inverter (NOT gate) included?

Truth tables for combinational logic systems

Drawing up a truth table for a system with more than one logic gate is not too difficult. As long as you know how each of the basic gates work, you can treat each gate on its own and then work your way through the system.

Before going ahead to look at the outputs in the truth tables, it is worth reminding ourselves of the number of combinations of inputs possible per number of actual inputs.

One input

If there is only *one* input (A), then there are only *two* combinations (logic 0 or logic 1). So the incomplete truth table would be drawn up as below (ignoring the results in the output column, Z).

A	Z
0	
1	

Two inputs

If there are *two* inputs (A and B) they can be arranged in *four* different combinations:

- A and B both off
- A off and B on
- A on and B off
- A and B both on.

You cannot create any other combinations. The truth table would therefore be drawn up as below (ignoring the results in the output column, Z).

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

You should notice that the input columns are arranged in binary number order.

Three inputs

If there are three inputs (A, B and C) they can be arranged in eight different combinations. The truth table for a 3-input system is shown below.

A	B	C	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	

Summary

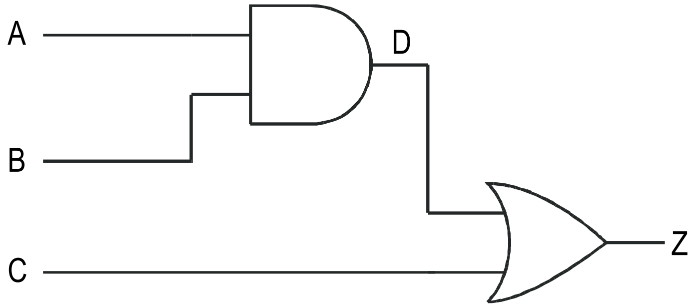
The pattern in the truth tables above is clear. Starting with one input giving two combinations, you simply *double the number of combinations* each time an input is added.

- 1 input: 2 combinations
- 2 inputs: 4 combinations
- 3 inputs: 8 combinations
- 4 inputs: 16 combinations

and so on.

You will never be asked to work with a system that has more than three inputs.

Worked example



The example below shows a logic diagram that has two logic gates. There are three inputs, so this gives eight combinations in the truth table.

Stage 1

Draw up the results for point D.(This is the output from the AND gate, being fed by inputs A and B only.)

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	

Stage 2

Draw up the results for point Z. (This is the output from the OR gate, being fed by output D and input C only.)

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

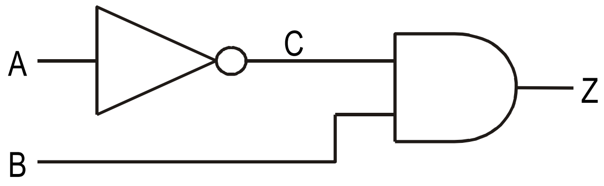
By following this technique, logic system problems can be solved easily.

You could use a circuit simulation program to check your results.

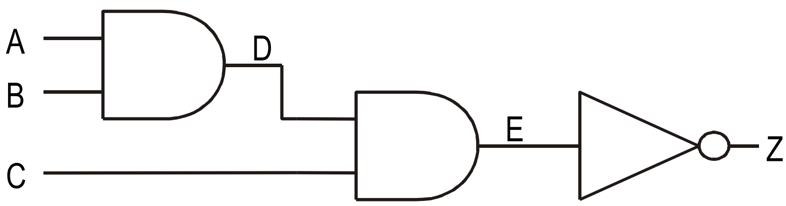
Exercises

Draw up a ***truth table*** for each of the following logic systems.

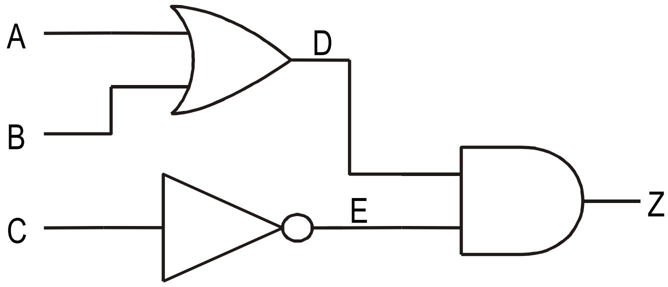
1 :



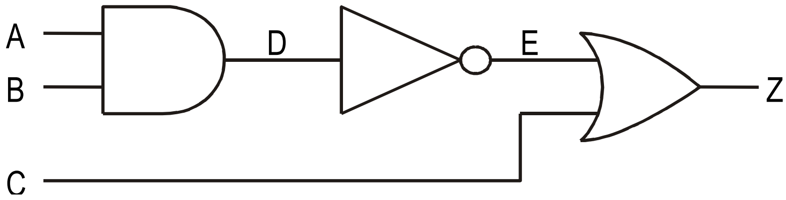
2 :



3 :



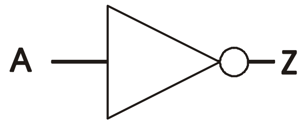
4 :



Boolean Algebra

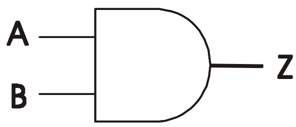
Circuits can get quite complex and sometimes using maths can help us describe what's going on in a very neat way. It can also be used to simplify circuits to reduce the number of logic gates we need.

$$Z = \overline{A}$$



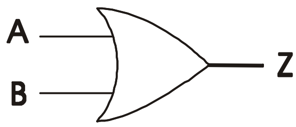
NOT

$$Z = A.B$$



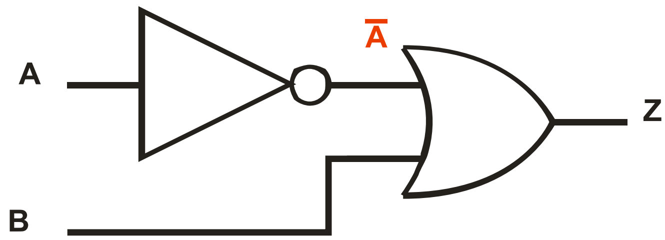
AND

$$Z = A+B$$



OR

Example :



The output Z will come on when **A is on OR B is off**. The Boolean expression that says this is,

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

Now try to write Boolean expressions for the circuits on the previous two pages.

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

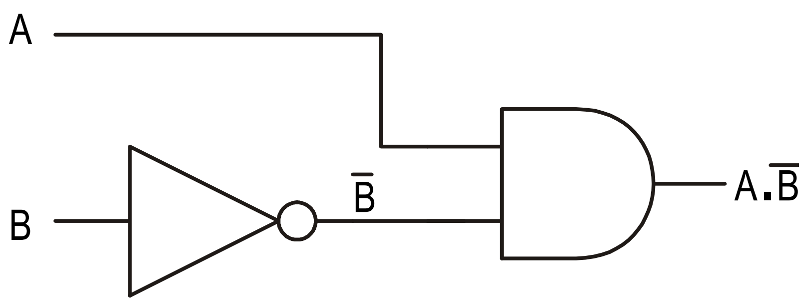
The output Z is at logic 1 in the third row down, and we can see that for this to happen A must be at logic 1 and B must be at logic 0. In other words

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

$$\mathbf{Z = A.\overline{B}}$$

This means that the logic diagram is as shown below.

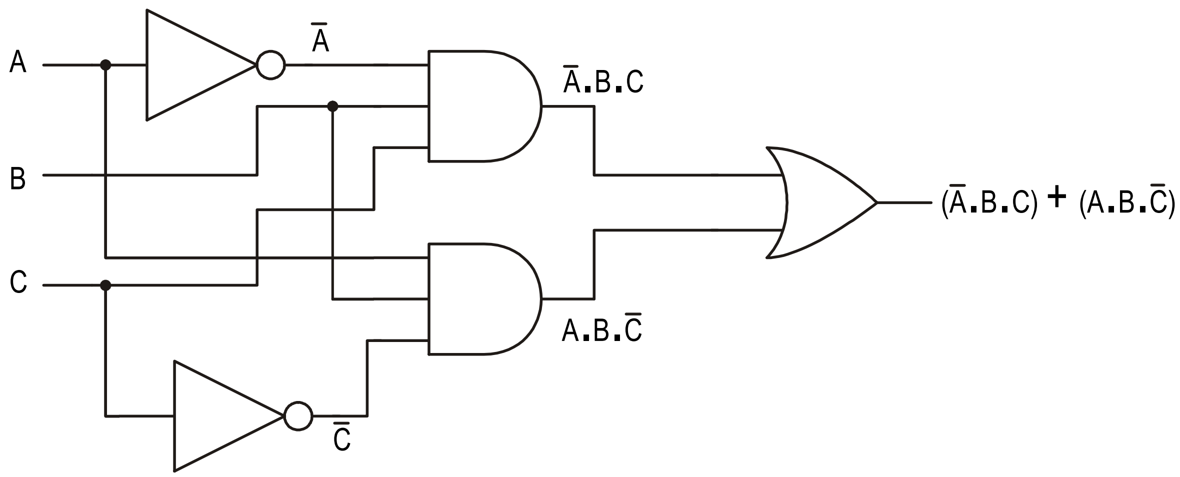


Worked example

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

In other words, Z = 1 if (A is at logic 1 AND B is at logic 1 AND C is at logic 1) OR if (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

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

Exercise

Complete a Boolean expression and draw the logic diagrams for each of the following truth tables.

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

(a)

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

(b)

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

(c)

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

(d)

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

(e)

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

(f)

Creating logic systems from written specifications

Perhaps the most common application of switching logic is creating a logic system to meet a given specification. Normally, by reading the specification carefully, the system designer can almost 'see' the required logic system.

Worked example

A burglar alarm system is to sound if a master switch is on and either a light beam is broken or a pressure pad is stood on.

Draw a logic diagram and a truth table for this system.

Read the specification carefully. You should notice that it has *three* inputs. These are:

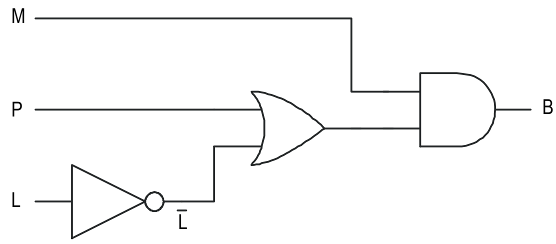
- a master switch (M)
- a light sensor (L), and
- a pressure pad (P).

It has *one* output, an alarm bell (B).

The bell should go to logic 1 if the master switch is at 1 and either the light beam goes to logic 0 or the pressure pad goes to logic 1. This can be written in Boolean as:

$$\mathbf{B = M \, . \, (L + P)}$$

Note: The alarm has to be triggered when the light beam is broken and so a NOT gate is needed.



In other words, you need a two-input AND gate that is fed directly from M and also from a two-input OR gate that is fed from L (through an inverter) and P. The logic diagram is shown below.

The truth table for this system is shown below. Again, all you have to do is read the specification carefully and then read across each row, one at a time, and decide whether the bell should be ringing or not. There are some short cuts. For example, in the first four rows the master switch is off; therefore the bell must be at logic 0 – even if there is a burglar in the house.

M	L	P	B
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Exercises

1. A house doorbell is to ring if a push button at the front door, a push button at the back door or both buttons are operated. Draw a logic diagram and write a Boolean equation.

2. A lift motor is to start only when, by closing, the door has actuated a switch and a passenger has pressed a button. Prepare a truth table, a logic diagram and a Boolean equation for this system.

3. The driver of a dustcart is to be able to operate the loading claw by pressing a button, but only when the senior loader at the rear of the cart has pressed a button to give the 'all clear'. Draw a logic diagram and write a Boolean equation for this system.

4. An automatic central heating system is to heat the radiators (R) if the mains switch (M) is on, the timing control switch (T) is closed and the override button (O) is not selected. Draw a logic diagram, truth table and Boolean statement for this system.

5. A drill is to operate if an isolator is closed, a guard is in place (closing a microswitch), either 'HI' or 'LOW' speed is selected and a foot pedal is operated. Draw a suitable logic diagram for this system. Draw up a truth table.

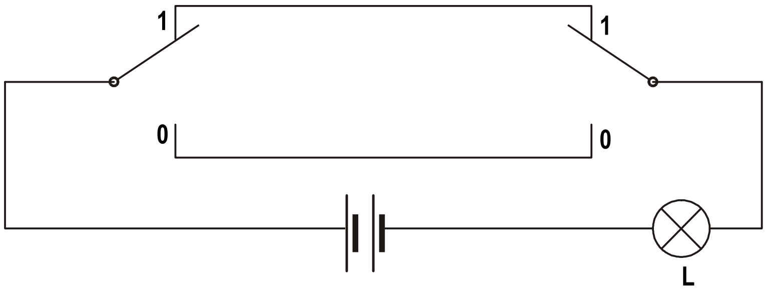
6. A large hall has three temperature sensors. A logic system is to operate the radiator when any two of the temperature sensors fall below a preset level. Draw up a truth table for this system and draw a logic diagram.

7. A burglar alarm will operate if the mains switch is on and either an electronic beam is broken, a pressure pad is stood on or a window is opened. Draw a logic diagram for this system.

8. At the start of a boxing match, a bell is to ring, provided:
   - boxers A and B are present
   - the referee and the time-keeper are present.

If either or both boxers fail to appear, the match is to start with the next pair of boxers, C and D. Draw a logic diagram for this system.

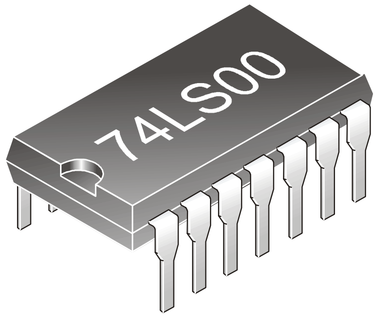
9. A switching system for corridor lighting is shown below.



   (a) Draw a truth table for this system.
   (b) Write a Boolean equation for this system.
   (c) Draw a logic diagram of an electronic system that could be used to achieve the same control of the light.

Logic gate integrated circuits (ICs)

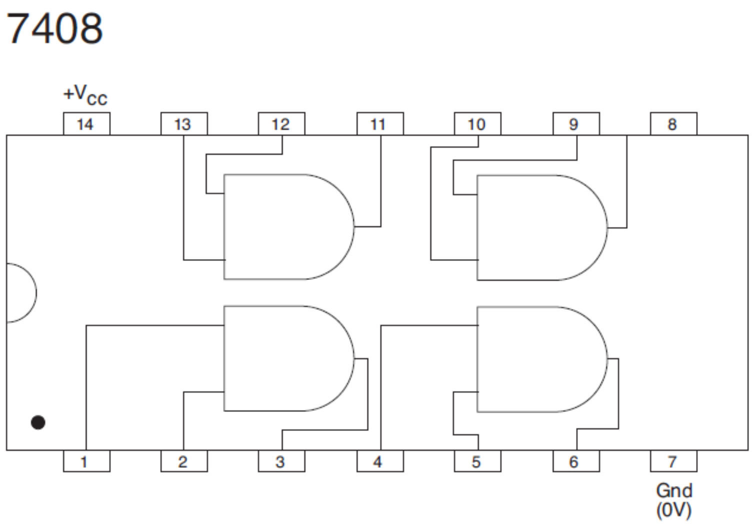
Integrated circuits consist of plastic cases filled with electronic circuitry. There are many resistors, transistors and other components packed into the chips. There are literally thousands of ICs on the market, all designed to do different jobs – logic gates, amplifiers, timers, etc.



The chips we are using are known as TTL chips and require a stable 5V supply to work properly. (Great difficulties will be met if any other voltage is used.) Any unconnected pins automatically go to logic 1. In other words, if a wire connected to a pin is connected to the 0-volt rail (logic 0), it will go to logic 0. If the wire is disconnected from the 0-volt rail it will go to logic 1. However, it is good practice to connect pins to 'high' or 'low' as needed.

All TTL chips have a four-digit code number, which always starts with 74. For example, a 7408 is a quad two-input AND chip.

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*.



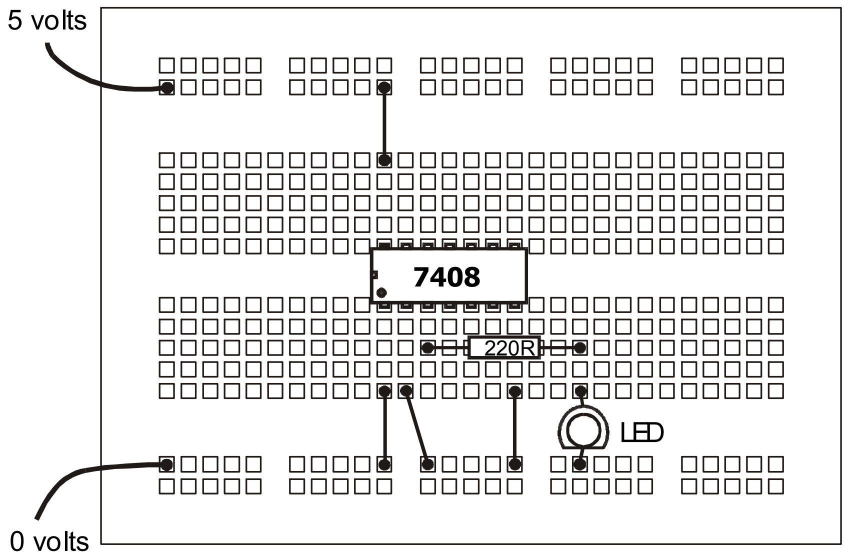
Pin 14 is connected to the 5-volt stable supply and pin 7 to 0 volts.

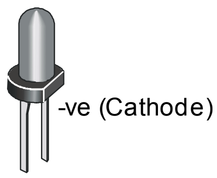
Logic ICs in prototype board circuits

Transferring and converting circuit diagrams to prototype layout diagrams can be confusing at first, but once you grasp the technique you should find it quite straightforward and enjoyable.

Here are some points worth noting before you start building logic circuits.

1. Although in theory we should wire up and use real mechanical switches to provide logic 0 and logic 1 inputs to the chip, we can simply use wires to do this. When logic 0 is required at a pin, the connecting wire is simply connected to the 0-volt rail. Similarly, when logic 1 is required, the connecting wire is pushed into the 5-volt rail.





3. To show the output condition of any logic system, an LED will be used. This will glow when the output is at logic 1 and be off at logic 0. Remember that LEDs are polarity conscious; that is, they must be connected with the negative terminal towards the 0-volt rail. The 'flat' or short leg on an LED is the cathode or negative side.

3. LEDs must be protected from excess current. A protective resistor must be placed in series with the LED to do this. As the current and voltage are already relatively low, a 220 R resistor will suffice.

4. Remember that when connecting one component to another you must use parallel vertical columns on the breadboard. If you connect two components into the same column the prototype board will short circuit them.

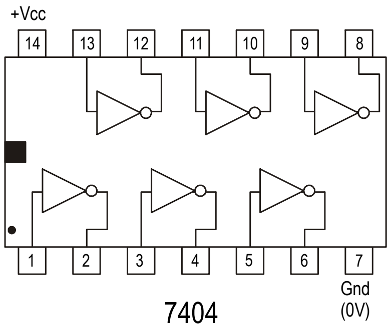
Pin-out diagrams

ICs are impossible to use without the manufacturer's data sheets to show what facilities are available on the chip and how the pins are to be connected. These data sheets contain pin-out diagrams. A pin-out diagram is a graphical layout of the chip and its contents.

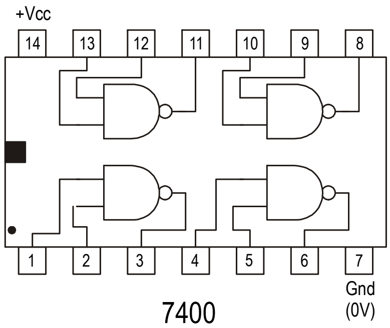
Note: all chips have either a notch or a small dot (or both) above pin number 1 so that the user can identify all the pins without them being numbered. The dot is always at pin 1.

Pin-out diagrams for common TTL logic ICs

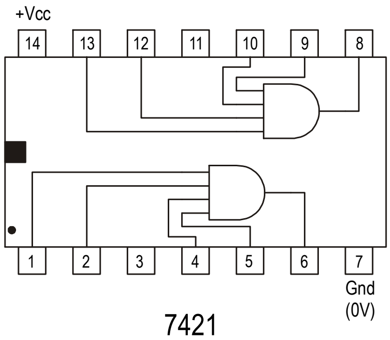
The description of each pin-out diagram gives details of the chip. For example, a 'dual four-input OR' means the chip has two (dual) OR gates on it, each having four inputs. A 'quad two-input AND' means the chip has four AND gates, each gate having two inputs.



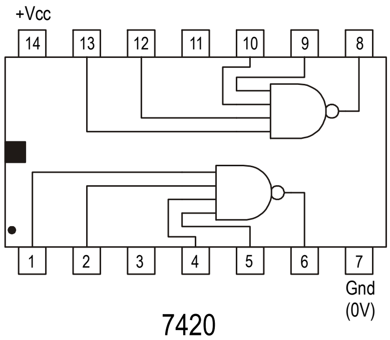
7404



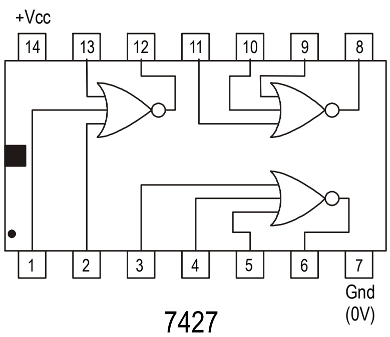
7400



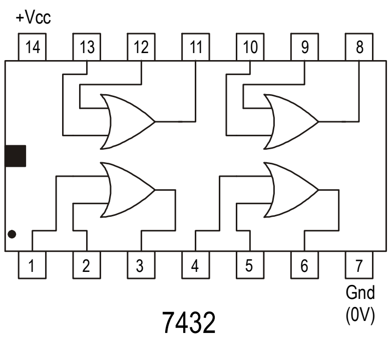
7421



7420



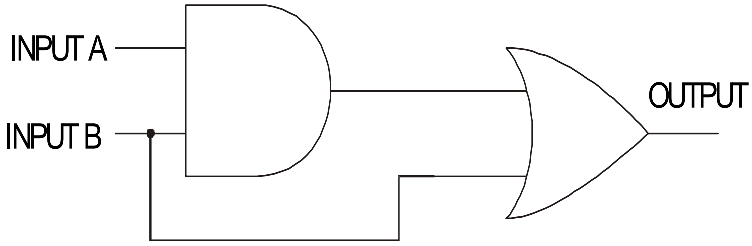
7427



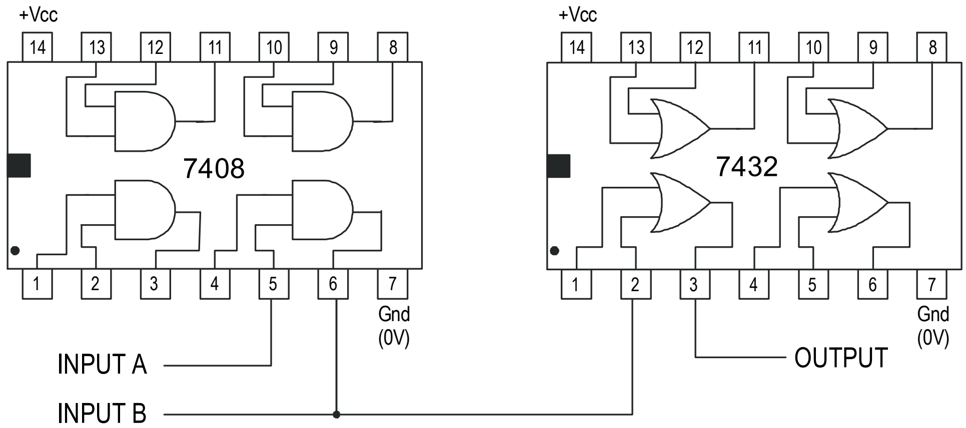
7432

Pin-out and wiring diagrams - example

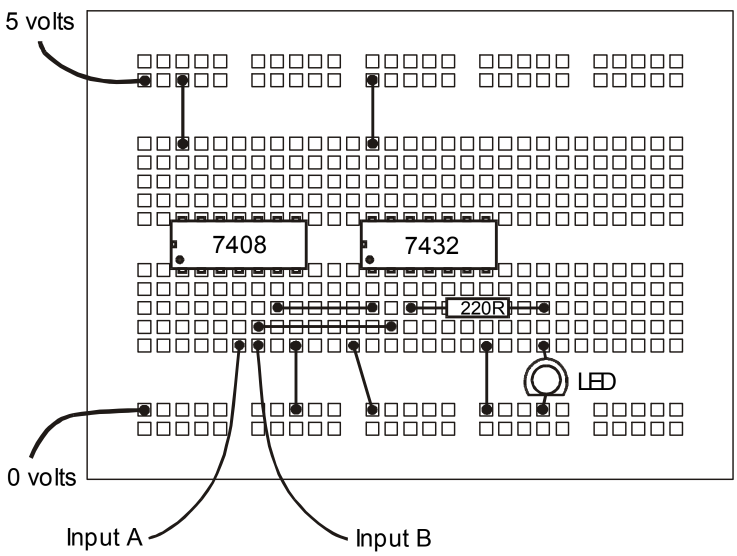
The following logic circuit could be constructed using ICs.



Since the gates within an IC are identical, any one of them can be used. An example of possible connections is shown in the IC circuit diagram below.



Prototype circuit layout/wiring diagram



The two ICs are mounted on a prototype circuit board as shown below. Connections between pins are made by 0.6 mm solid-core wire.

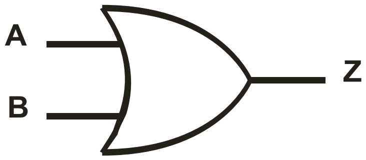
The circuit would now be tested against the truth table to check its operation. Inputs A and B can be made by connecting to the 5-volt or 0-volt rails.

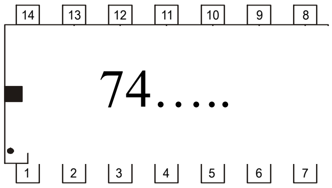
Note: in order for a logic circuit to work, it must be powered up; that is, the correct power supply must be connected to pin 14 and pin 7 must be connected to 0 volts.

Exercises

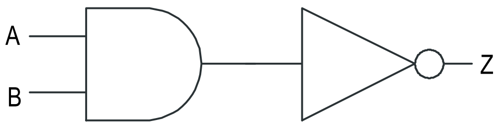
Select the required ICs and draw an IC circuit diagram for each logic system below.

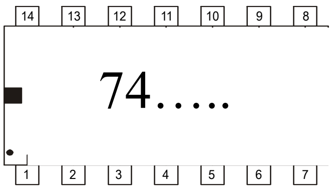
1 :

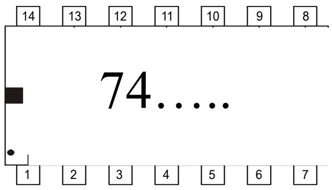




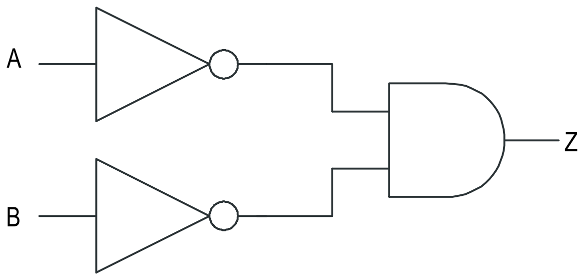
2 :

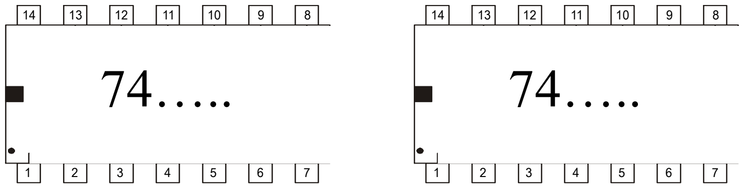




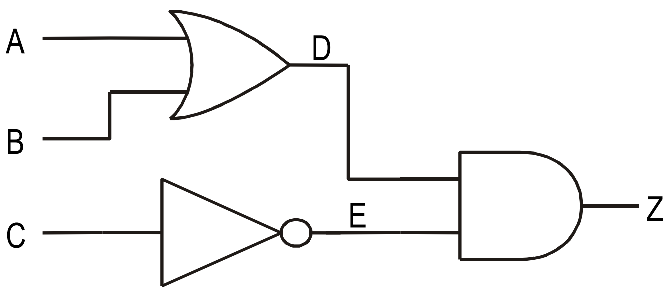


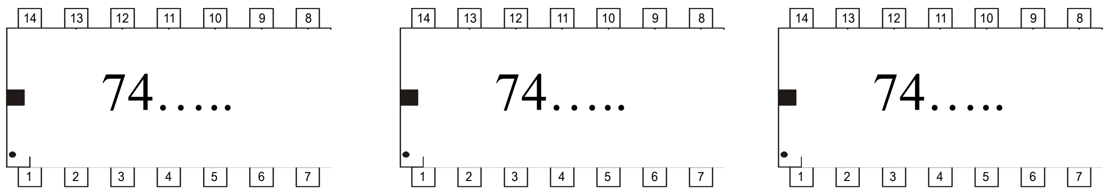
3 :


A
B
Z


14 13 12 11 10 9 8
74.....
1 2 3 4 5 6 7
14 13 12 11 10 9 8
74.....
1 2 3 4 5 6 7

4 :


A
B
C
D
E
Z


14 13 12 11 10 9 8
74.....
1 2 3 4 5 6 7
14 13 12 11 10 9 8
74.....
1 2 3 4 5 6 7
14 13 12 11 10 9 8
74.....
1 2 3 4 5 6 7

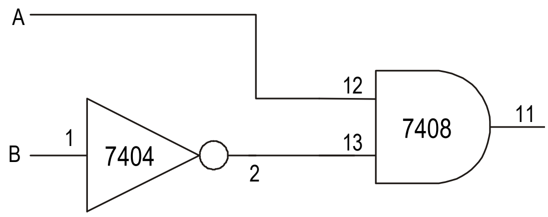
Extension work

Draw up a truth table for each example and use circuit simulation software to check your results.

Practical tasks

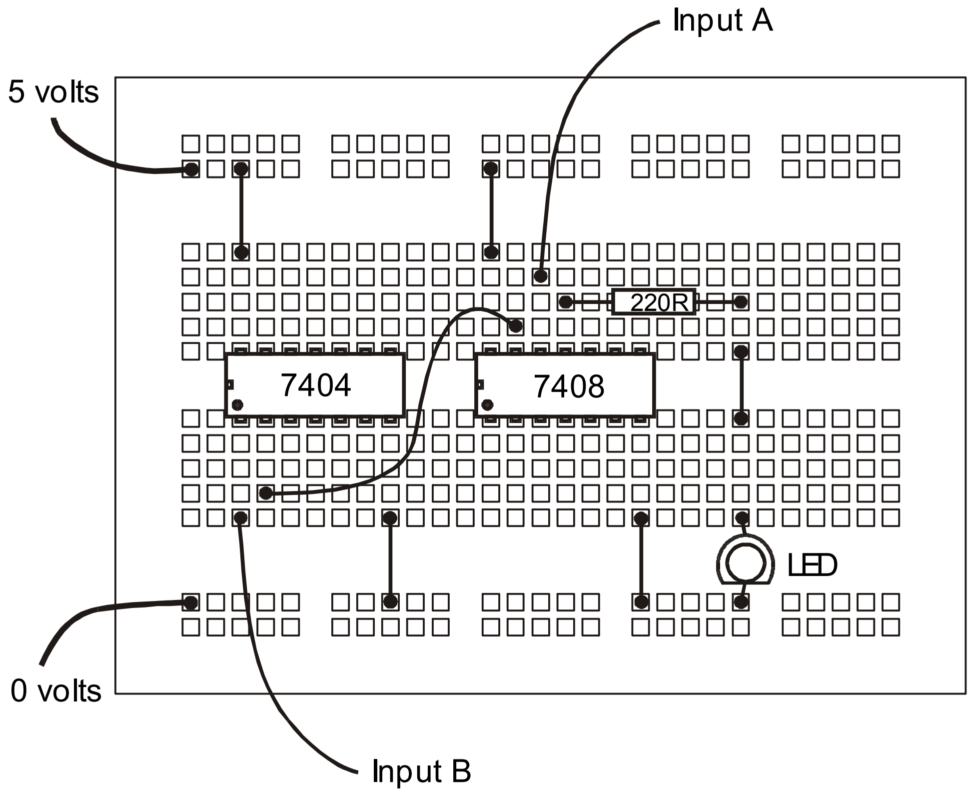
Worked example

Here is the design for a logic system that is to be tested on a prototype circuit.



- Identify the required pin-out diagrams and number each input and output being used on the chips. In this case the chips are a 7404 and a 7408.

- Insert the chips on to the prototype board and make the connections to the +V and 0-volt rails.
- Make the other connections and insert the LED and resistor.



- Create input wires as required.

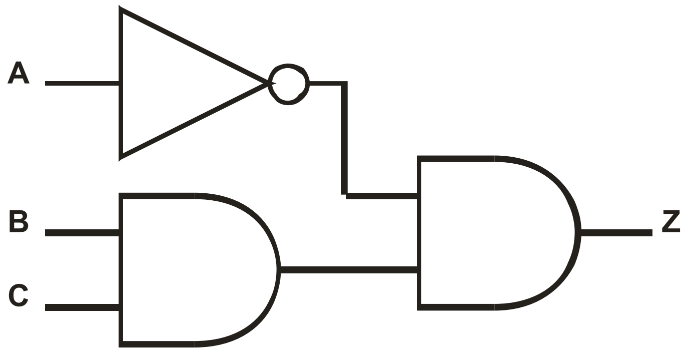
Task 1

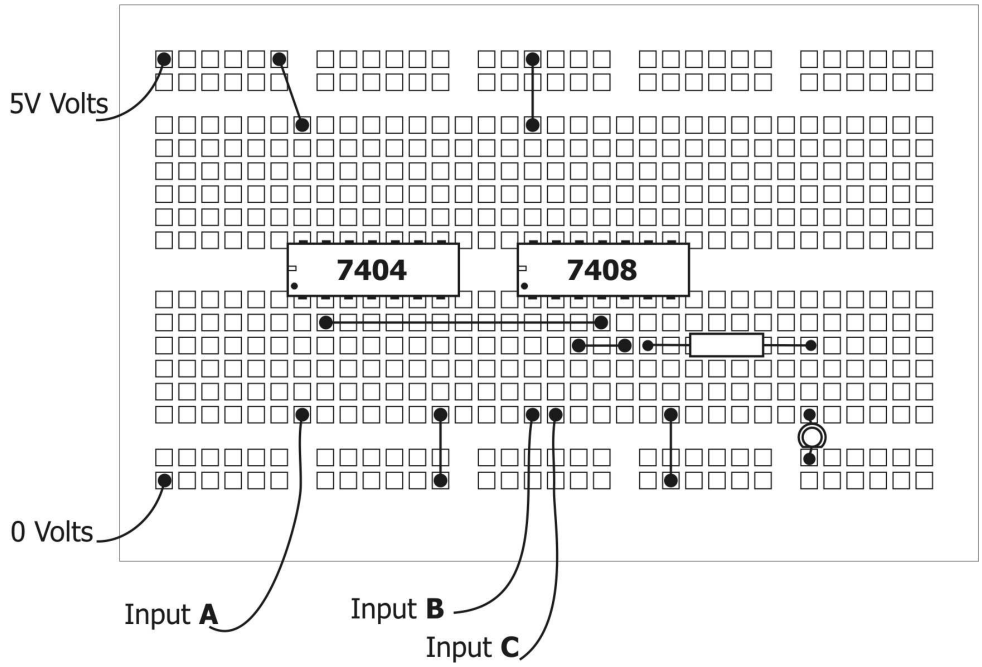
- Build the circuit shown above.
- Work your way through each row of the truth table and draw up the results in output column Z.

A	B		Z

Task 2

Build the following logic system and establish a truth table showing all possible combinations of the inputs.





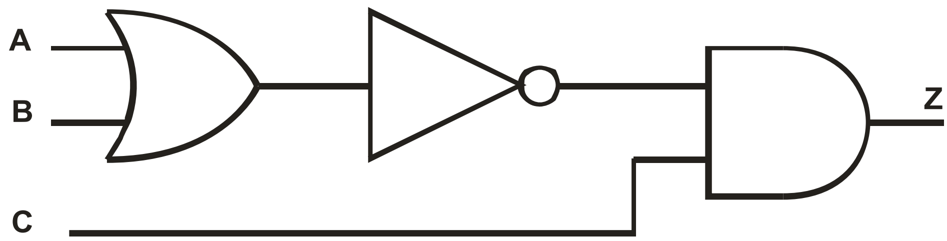
A	B	C					Z
0	0	0					
0	0	1					
0							

Task 3

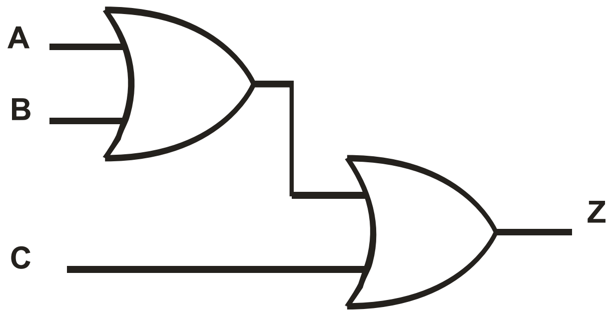
Three different logic systems are shown below. Develop a truth table for each one, then build the system on a prototype circuit board. Confirm your results predicted in the truth table.

Note: do not just test the logic-1 conditions; make sure that you test the outputs given when logic 0 is applied.

1 :



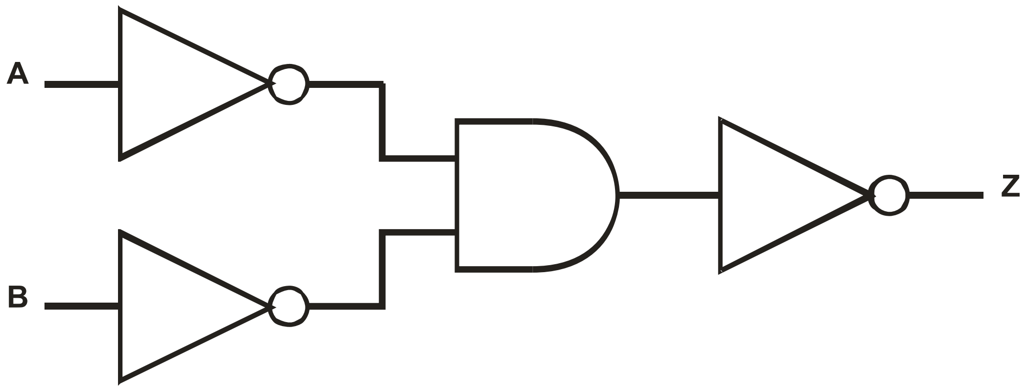
A	B	C			Z	Simulation



2 :

A	B	C		Z	Simulation

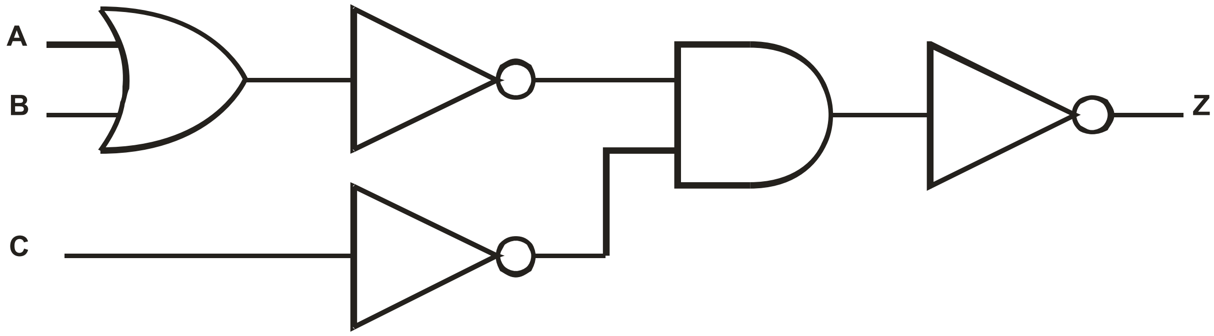
3 :



A	B					Z	Simulation

Task 4

Set up the following logic system on a prototype circuit board and draw up a truth table. Work your way through the truth table conditions and observe the output.



A	B	C					Z	Simulation

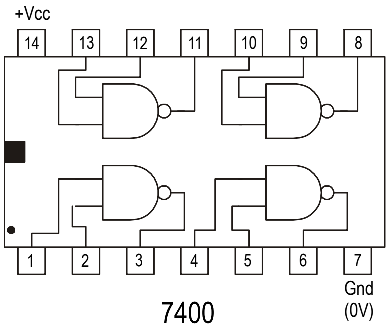
Test equipment

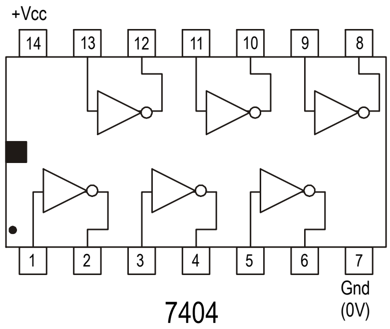
When trying to establish logic levels within a complex system or to monitor a logic output without using an LED, we use a digital logic probe.

The logic probe is powered from the same supply as the logic circuit being tested and the needle point is pushed against the various pins on the IC to test their logic level. Normally the logic probe gives out a high-pitched sound and a red LED lights if the pin being tested is at logic 1. If the point tested is at logic 0, a low-pitched sound is emitted and a green LED lights.

<u>Alternative chip names</u>

ICs are often referred to with full names that clearly describe their function.





QUAD—2 Input NAND ***Hex Inverter***

Look in your databook and complete full names for the following ICs.

7402 ____

7408 ____

7432 ____