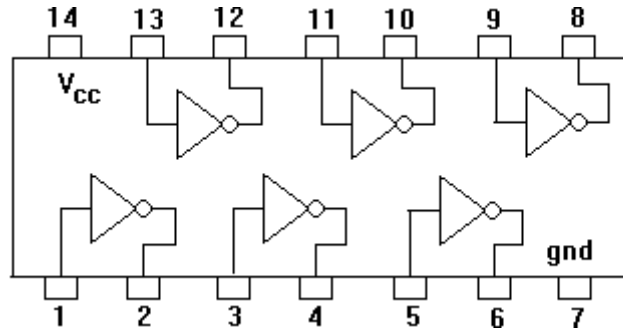
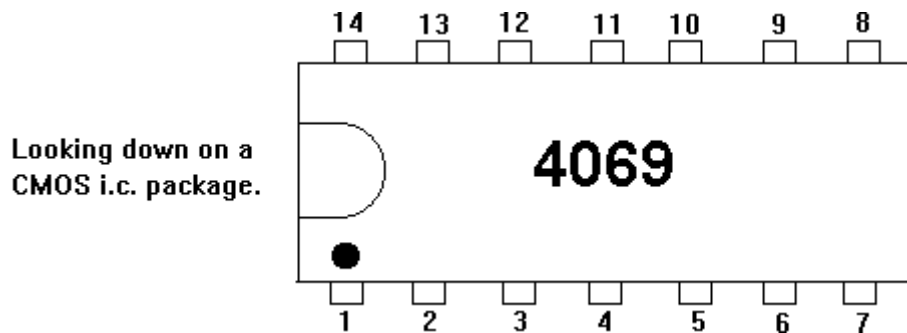


### Investigating the NOT gate characteristic

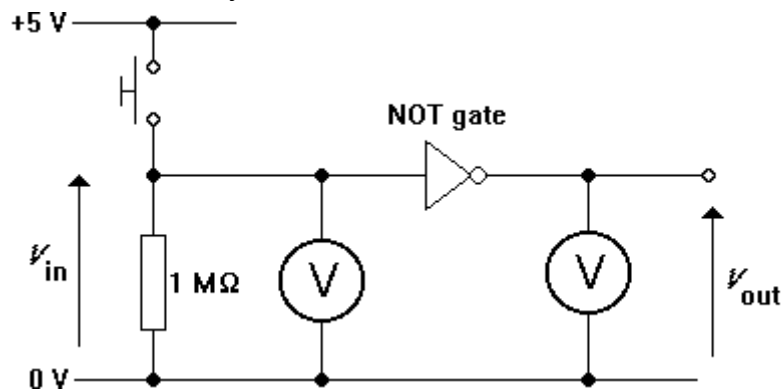
You are going to use a voltmeter to explore the behaviour of a NOT gate in a 4069 integrated circuit (i.c.). Here is the pinout of the i.c., showing you the layout of the six separate gates within the package. Pin 14 should be connected to +5 V and pin 7 to 0 V to provide power for the gates.



The diagram below shows you how to identify pin 1 of the package.



1. Assemble the circuit below on breadboard, without the voltmeters. Use the Discrete Components sheet to identify the 1 M $\Omega$  resistor.

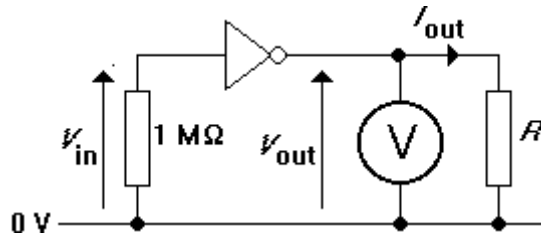


2. Connect one terminal of a voltmeter to 0 V. Connect the other terminal to the output of the NOT gate to measure  $V_{out}$ . If all is well it should go from +5 V to 0 V when the switch is pressed.
3. Now use the voltmeter to measure the values of  $V_{in}$  when the switch is pressed and released.

4. Use your results to complete this table for the NOT gate.

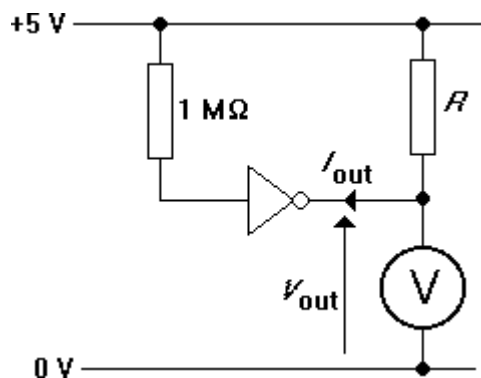
input	output
low	?
high	?

5. Now pull the input low with a 1 MΩ resistor (below). Check that  $V_{in}$  is at 0 V. Insert the values of  $R$  shown in the table, measuring the value of  $V_{out}$  each time.
- 6.



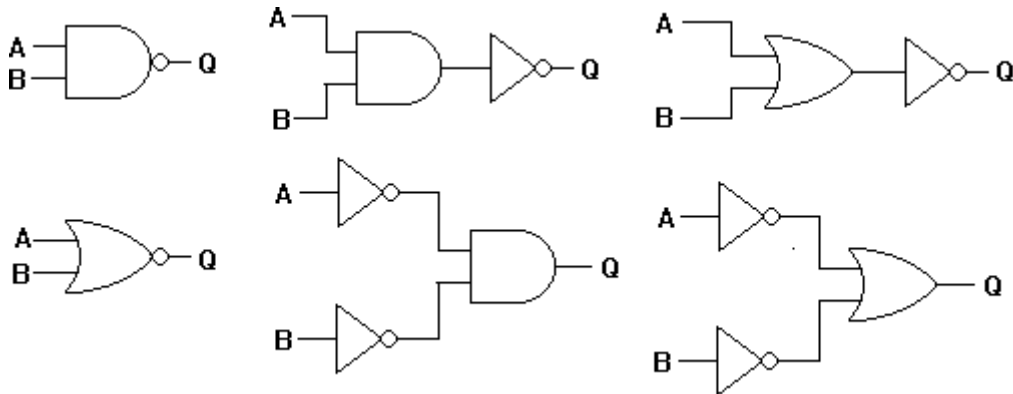
$R/\Omega$	$V_{out}/V$	$I_{out}/mA$
100		
470		
1000		
4700		
10 000		

6. Use the values of  $V_{out}$  and  $R$  to calculate  $I_{out}$ , the current provided by the output of the NOT gate. As the resistance increases, the current should decrease.
7. Use the circuit below to find out how  $V_{out}$  depends on  $I_{out}$  when the gate input is high. Use the range of resistors given in the table above. Hint: when you calculate the current in  $R$ , remember that the voltage drop across  $R$  is  $5.0 - V_{out}$ .

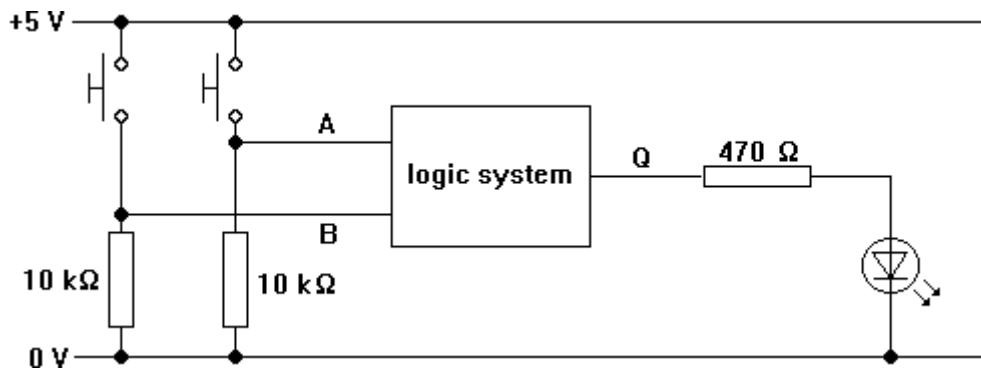


### Truth tables of some logic systems

You are going to find out the truth tables of the logic systems shown here.



The circuit below shows the arrangement of switches, resistors and LED needed to get signals into and out of the system.



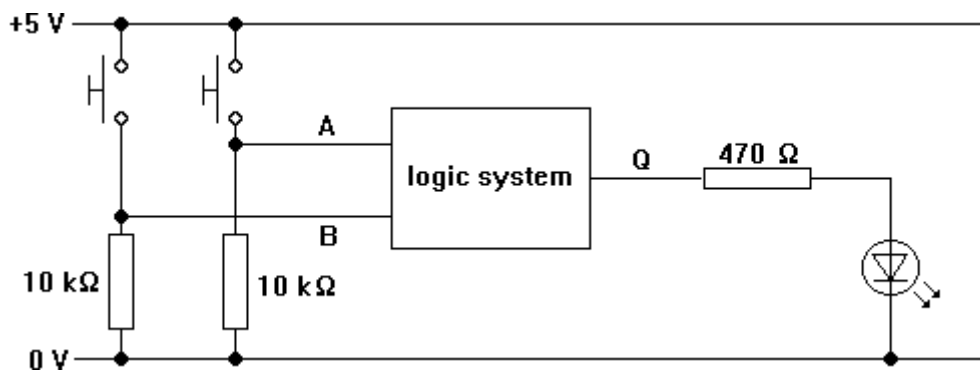
1. Assemble the switches and their 10 kΩ pull-down resistors on the left-hand end of your breadboard. Use a voltmeter to check that A and B go high when the relevant switch is pressed.
2. Add the LED and its 470 Ω current-limiting resistor at the right-hand end. Check that the LED glows when Q is plugged into the top supply rail.
3. Place a 4011 NAND gate i.c. in the centre of your breadboard. Connect one of its gates to A, B and Q. Record its truth table by pressing the switches in all four combinations.
4. Use gates from 4071, 4081, 4001 and 4069 i.c.s to assemble the other logic systems. Record their truth tables.

### Designing an Exclusive-NOR gate

1. Design a logic system made from AND, OR and NOT gates which has this truth table.

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

2. Assemble the circuit on breadboard. Make sure that you have connected each i.c. to the +5 V and 0 V power supply rails.
3. Use the arrangement of switches and LED shown below to test the circuit once you have assembled it on breadboard.

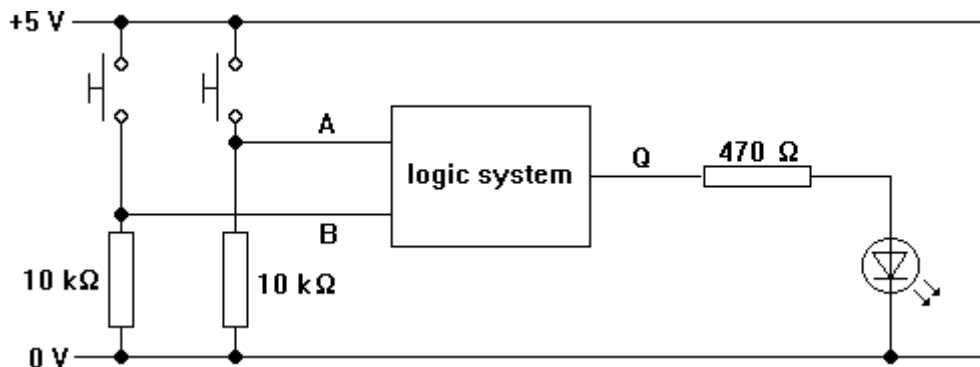


**NAND-gate designs**

1. Design a NAND-gate circuit which has this truth table. It acts as an Exclusive-OR gate.

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

2. Assemble the circuit and test it with switches and an LED as shown below.



3. Design and test a NAND-gate system with two inputs and three outputs. The number of outputs which are high must correspond to the binary number fed into the inputs. Here is its truth table.

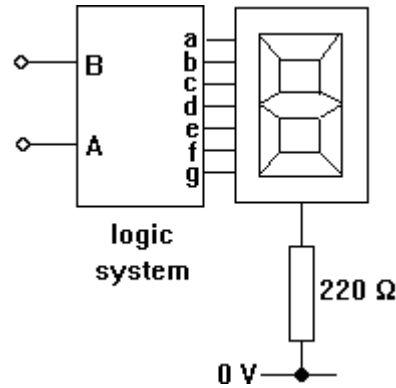
B	A	P	Q	R
0	0	0	0	0
0	1	1	0	0
1	0	1	1	0
1	1	1	1	1

4. Design and test a NAND-gate system which acts as a binary-to-one-of-four converter with this truth table.

B	A	W	X	Y	Z
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

### Binary-to-seven-segment converter

The logic system shown in the diagram below displays a number on the seven-segment LED corresponding to the binary value of the word BA.



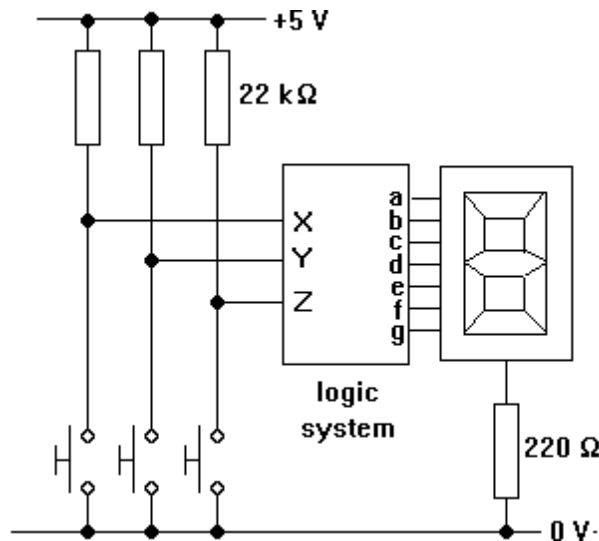
The number displayed on the seven-segment LED must be as shown in the table below.

B	A	number
0	0	zero
0	1	one
1	0	two
1	1	three

1. Write out a truth table for the logic system, showing how the outputs (a,b,....., f, g) are related to the inputs (B, A).
2. Write down seven separate Boolean algebra expressions, one for each output in terms of the two inputs.
3. Design a NAND gate circuit for the logic system. Try to minimise the number of NAND gates being used. Any output which controls an LED will not produce a clean logic signal at the input of another gate.
4. Assemble the system and test it. Use pull-down resistors and switches to feed signals into the inputs.

### Switch counter

Three switches control the state of three inputs X, Y and Z in the circuit below.



The logic system obeys this truth table.

number of pressed switches	number displayed
0	0
1	1
2	2
3	3

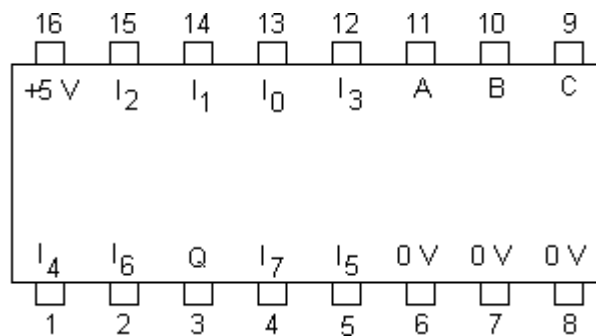
1. Write out a truth table to show how the state of the outputs (a to g) depend on the state of the inputs (B, A).
2. Write down Boolean algebra expressions for each output in terms of the inputs.
3. Design NAND-gate circuits to generate the seven output signals (a to g). Bear in mind that any gate output which is used to make an LED glow will not provide a reliable logic 1 when connected to the input of another gate.
4. Assemble your design and test it.

### Programmable logic gates

You are going to use a 4051 multiplexer i.c. to make a logic system which obeys this expression.

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

1. Reduce the expression to standard form.
2. Write out a truth table for the system.
3. By connecting the eight inputs ( $I_0$  to  $I_7$ ) of the multiplexer to the supply rails, force the multiplexer output  $Q$  to obey the truth table.

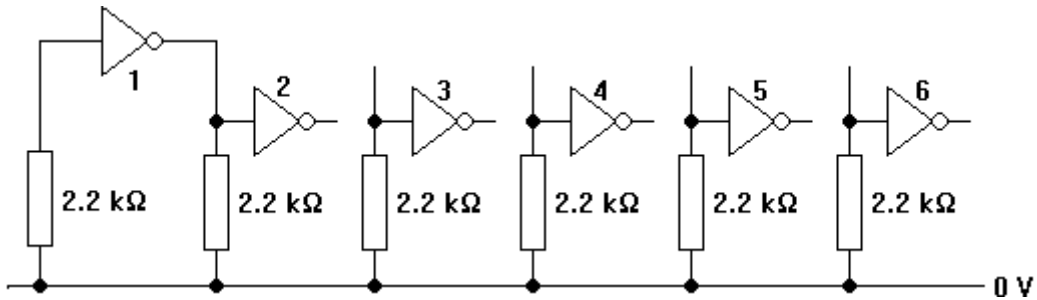


4. With the help of three switches, pulldown resistors and an LED (with a series resistor), test the system.
5. Reprogram the multiplexer so that it feeds out a 1 only when any two or more of its inputs are held high.

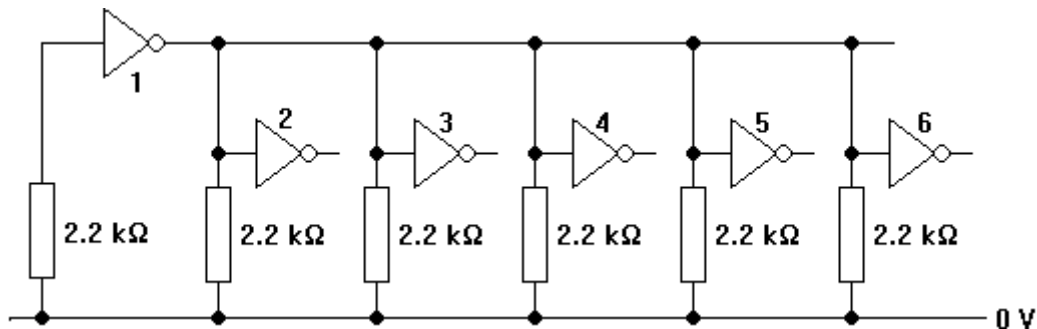


## Measuring fanout

1. Place a 4069 NOT gate i.c. on your breadboard. Number the gates from 1 to 6.
2. Use six 2.2 k $\Omega$  pull-down resistors to hold the inputs of the gates low. Connect the output of gate 1 to the input of gate 2.



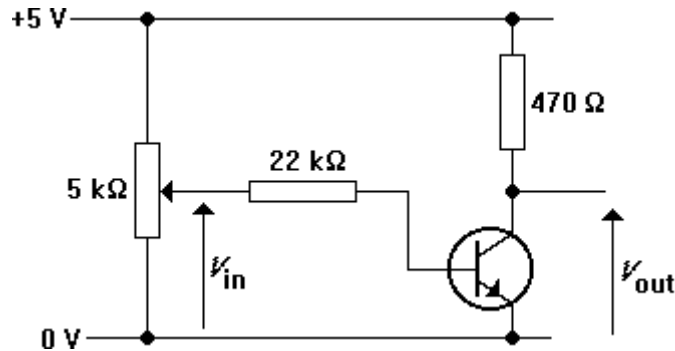
3. Use a double-beam CRO to look at the outputs of gates 1 and 2.
4. Record the voltages on the CRO screen as the inputs to gates 3, 4, 5 and 6 are connected to the output of gate 1, **one after the other**, until all four are connected as shown below.



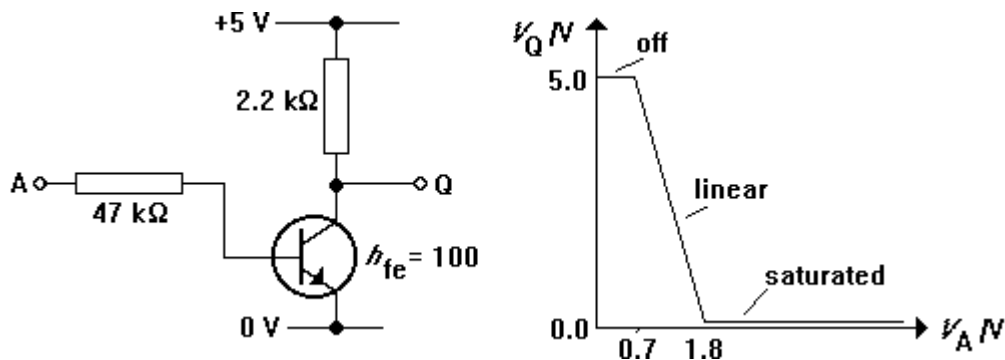
5. On the same set of axes, plot graphs to show how the input and output voltages of gate 2 depend on the number of inputs connected to the output of gate 1.
6. Use the graphs to determine the fanout of CMOS gates whose inputs are pulled low by 2.2 k $\Omega$  resistors.
7. Investigate the effect of replacing the pull-down resistors with 4.7 k $\Omega$  and 1.0 k $\Omega$  resistors in turn.

### Exploring a bipolar transistor

This circuit will allow you to determine the  $h_{FE}$  of a BC107 transistor.



1. Assemble the circuit.
2. Use a double-beam CRO to look at  $V_{in}$  and  $V_{out}$ . Use 1 V/div, with 0 V at the bottom of the screen. If all is well,  $V_{out}$  should fall from 5 V to 0 V as  $V_{in}$  rises from 0 V to 5 V.
3. Measure the value of  $V_{out}$  for values of  $V_{in}$  covering the range 0.0 V to 5.0 V at intervals of 0.25 V.
4. Plot the data on a graph similar to that below. Join up the dots with three straight lines (off, linear and saturated).

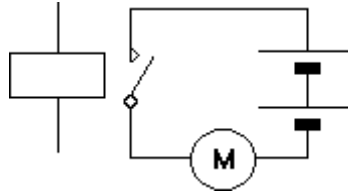


5. Repeat steps 3 and 4, but looking at the base voltage instead of the collector voltage.
6. Use your graph to read off the value of  $V_{in}$  which makes  $V_{out}$  2.0 V. By considering the voltage drop across the base and collector resistors, calculate values for  $I_b$  and  $I_c$ . Hence determine the value of  $h_{FE}$  for your transistor.
7. Replace the 2.2 kΩ resistor with 1 kΩ and 4.7 kΩ in turn. Note what effect this has on the  $V_{out}-V_{in}$  graph.

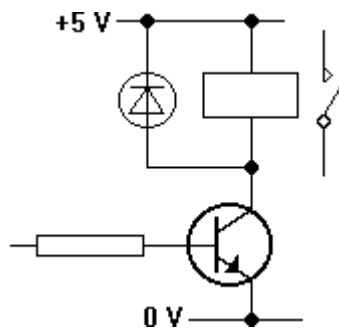
## Driving a motor when it gets wet

You are going to design and assemble a circuit which makes a small d.c. motor spin when a pair of wires are dipped in water.

1. Connect the d.c. motor to a 3 V battery via a relay. Check that the motor operates when there is 5 V across the coil.



2. Use a BC107 to sink current from the coil. Use a 4.7 k $\Omega$  base resistor and don't forget the reverse-biased diode in parallel with the coil.



3. Use a 4069 NOT gate to control the transistor. Check that the motor only operates when the gate input is low.
4. Use a 1 M $\Omega$  resistor to pull the gate input high. Use a pair of wires to connect the input to 0 V via some water in a beaker. Verify that the motor only operates when the wires are wet.

