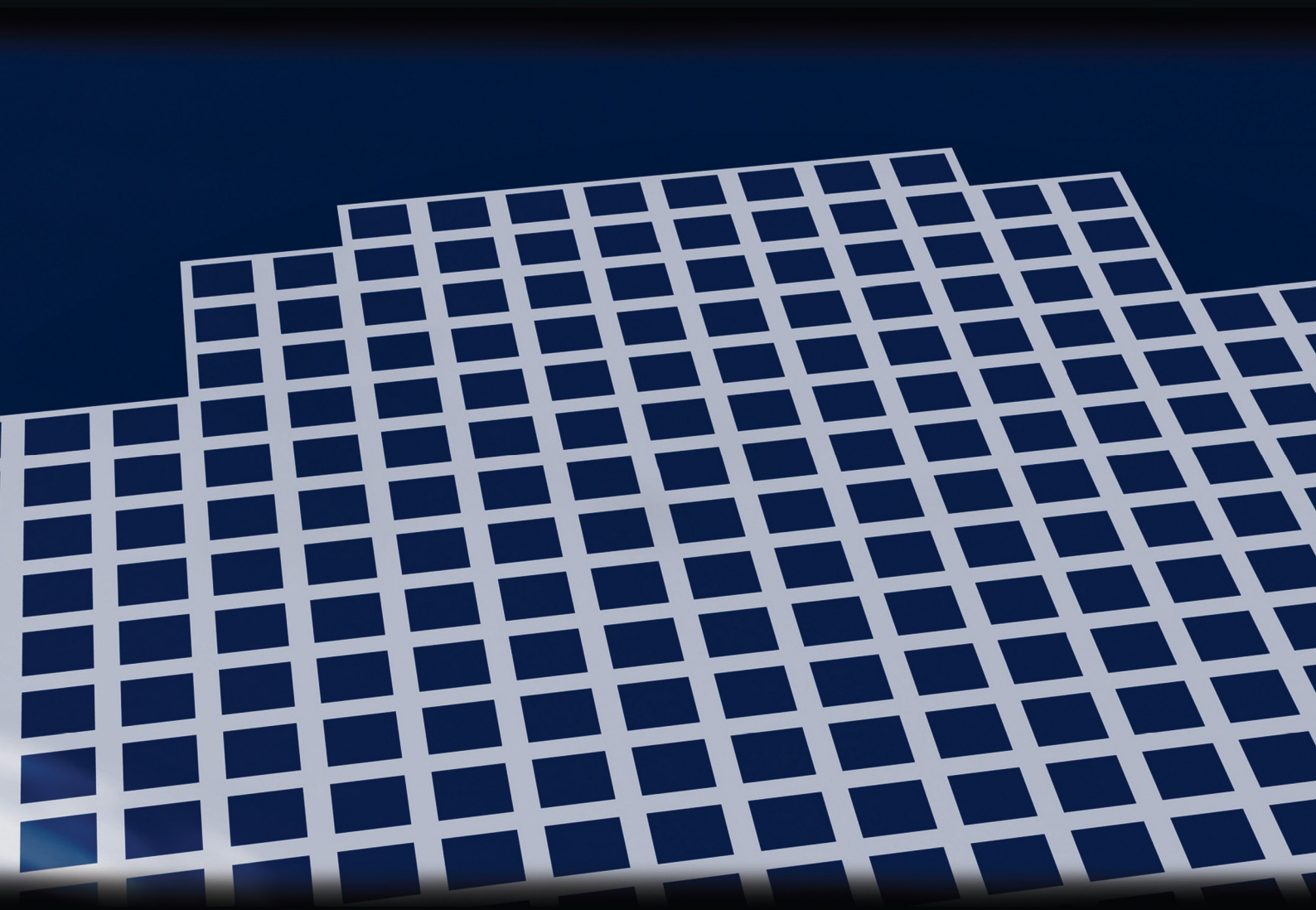
locktronics

Simplifying Electricity

Combinational logic



LK2094

MATRIX

www.matrixtsl.com

Contents

Combinational logic

Worksheet 1
Analogue vs digital

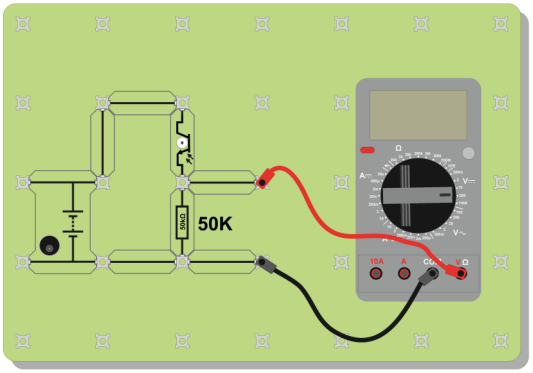
Combinational logic

We hear more and more about the ‘**d**’ and the ’**e**’ words - **d**igital television, **d**ab, **d**vd, digital cameras ... **e**-cards, **e**-commerce, **e**-books, **e**-skills, **e**-learning ...

Why is that? Whatever happened to the ‘**a**’ word - **a**nalogue?

Over to you:

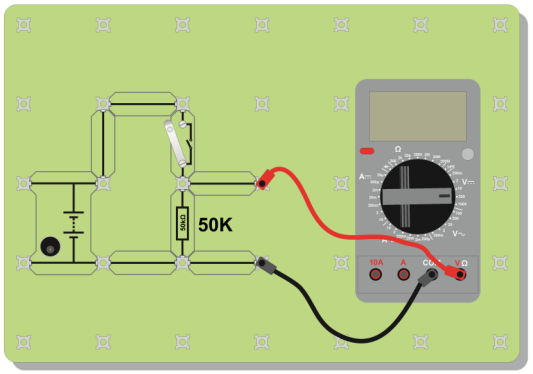
The first circuit uses an analogue sensing device, a phototransistor, and a series 50kΩ resistor, to make a light-sensing unit. The output voltage is measured using a DMM (digital multimeter.)



DC ⎓

- Set up this circuit shown opposite.
- Set the DC power supply to 6V. Set the DMM to read voltages up to 20V DC. (The symbol for DC is shown underneath the picture.)
- Switch it on.
- Vary the amount of light reaching the sensor by slowly lowering your hand over it. What do you notice about the output voltage?

The second circuit is a digital sensing unit, using a switch unit.

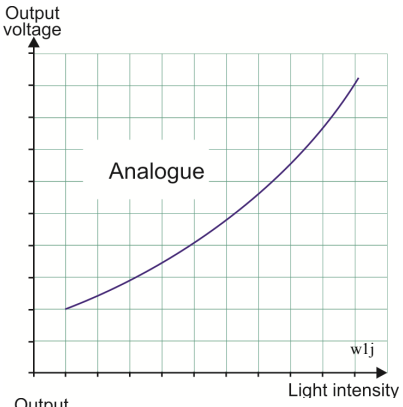


- Set up the circuit - just replace the phototransistor with a switch.
- Measure the output voltage when the switch is open (off) and again when it is closed (on.)
- Invert the switch unit. (Swap over the switch and the resistor.)
- Measure the output voltages again, with the switch open and then closed .
- Compare the behaviour of the analogue and digital circuits.

Worksheet 1
Analogue vs digital

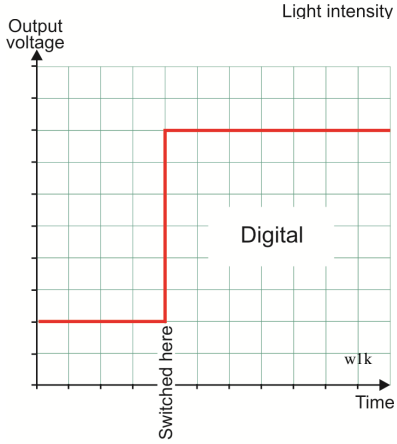
Combinational logic

So what?



An analogue sensor gives an analogy - a copy of the behaviour it is sensing. With the light-sensing unit, as the light level goes down, the output voltage goes down - the voltage mimics the light level. We can change the light level by very small amounts, so we can change the output voltage by very small amounts.

A digital sensor, on the other hand, is a two-state affair. A switch is either on or off - just two possible states. The output voltage, as a result, has one of only two possible values.



These ideas are shown in the graphs opposite. The top one shows an analogue signal, changing continuously as the light intensity changes. The lower graph must be plotted in a different way. The state of the switch does not change smoothly from off to on. It can't be slightly on, and then a bit more on, and so on. It is **on** or **off**. The horizontal axis shows the time at which the change from on to off occurs. The output voltage always has one of only two values.

The vocabulary of digital electronics talks about these two voltages as 'logic 0' and 'logic 1'. Somewhere in a particular design, these will be defined, usually as a range of possible voltages. For example, logic 0 may be defined as any value between 0V and 1.0V, while logic 1 is any value between 10.0V and 12.0V. Giving a range of values recognises that signals can change a little as they move through an electronic system.

A major advantage of digital signals is that we, and electronic systems themselves, can make a pretty good guess at what the signal should be if, for some reason, it arrives with a voltage of say 8.7V. We'd guess that it was really logic 1. This ability to recreate the original signal is called **regeneration**, and is one of the major benefits of digital signals. Analogue signals do not allow us to do this.

For your records:

- An analogue quantity is one that copies the behaviour of another.
- An analogue signal can have any voltage value, usually between the voltages of the power supply rails.
- A digital quantity has only two possible states. A switch, for example can be off or on.
- A digital signal has only two possible voltage values, usually known as logic 0 and logic 1.
- This allows a digital signal to be regenerated - returned to its original value, when it has been affected by noise or distortion .
- Analogue signals cannot be regenerated in this way.

Worksheet 2
The NOT function

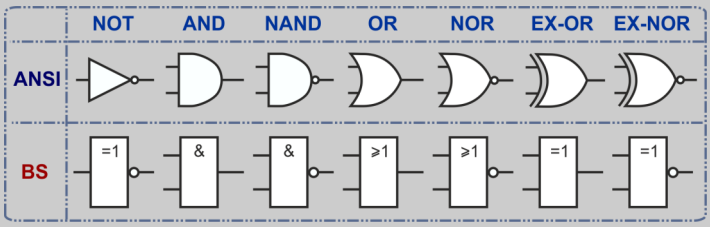
Combinational logic

A logic *function* is one way of manipulating digital signals. A logic *gate* is a device that will carry out a particular logic function. There are not many logic functions. This worksheet looks at the simplest, the NOT function. It could be used to trigger a warning when a vehicle door is NOT closed.

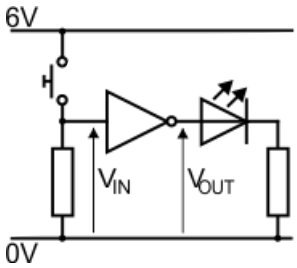
Logic gates can be built in a number of ways, and this has lead to a number of logic 'families' each with its own set of capabilities and limitations.

One such family is called **CMOS**. The photograph shows a CMOS NOT gate, identified by '4049'. It is known as a 'hex inverting buffer,' meaning that there are six NOT gates on the chip, (hex = 6, inverting = NOT), which buffer the signal (deliver a current of a few milliamps.)

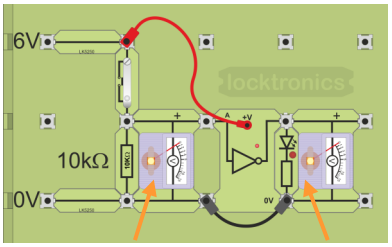
Like all electronic devices, logic gates are represented by circuit symbols. However, there several versions. The common ones are 'ANSI' (American National Standards Institute), and 'BS' (British Standard) sometimes called 'SB' (System Block) symbols.



Over to you:



- Set up the circuit shown, with a 6V power supply.
- Notice the LED connected between the output of the NOT gate and 0V, in addition to the LED built into the NOT gate carrier itself.
- With the switch turned off, measure the voltage:
  - V_{IN}, at the input of the NOT gate;
  - V_{OUT}, at the output of the NOT gate.
- Record both and note whether the LED is on or off.



Input voltmeter

Output voltmeter

Switch unit	V_{IN}	V_{OUT}	State of LED
Open (off)			
Closed (on)			

- Now close the switch. Record the new measurements.
- Invert the switch unit, by swapping over the switch and 10kΩ resistor.
- Repeat the measurements and record them in a second table.

Switch unit - inverted	V_{IN}	V_{OUT}	State of LED
Open (off)			
Closed (on)			

Worksheet 2
The NOT function

Combinational logic

So what?

First, a word about logic levels! For CMOS logic gates:

- logic 1 is any voltage greater than 70% of the supply voltage, (i.e. >4.2V in this case.)
- logic 0 is any voltage less than 30% of supply voltage, (i.e. <1.8V in this case.)

Ideally, voltmeters have infinite resistance, and do not affect the voltage being measured. The ones used here are far from ideal, and may pull the output below this logic 1 level. As stated previously, an advantage of digital signals is that we can mentally compensate for this.

- Use this information to convert your output voltage readings into logic levels.
- Then complete the table with these logic levels.

NOT gate	
Input	**Output**
(Logic) 0	
(Logic) 1	

This **truth-table** describes the behaviour of the NOT gate.
It produces the same effect, whether the switch unit is inverted or not - it turns a logic 0 input into a logic 1 output, and vice-versa.

However, the switch unit behaviour does change when inverted. Initially, it generates logic 0 when the switch is open, and logic 1 when closed. Inverted, the behaviour inverts so that it generates logic 1 with the switch open, and logic 0 with it closed .

A challenge -
Why do we need a resistor in the switch unit? Why not just have the switch?

- See what happens when you remove the resistor in both arrangements.

With the switch between +6V supply and input, things seem to behave in the same way. However, with it connected between 0V and input, nothing changes when it is closed. The NOT gate output always sits at logic 1, regardless of the state of the switch!

CMOS circuitry is wonderful, but its inputs must not ‘float’ (be left unconnected,) as then the output is unpredictable. It can even oscillate between logic 0 and logic 1 so rapidly that the IC overheats. Always use a resistor, either between positive supply and input, (to ‘pull’ the input up to logic 1), or between 0V and input (to ‘pull’ it down to logic 0). The ‘Locktronics’ NOT gate carrier is wired up so that the input sits at logic 0, when nothing is connected to it.

For your records:

- Copy the table with the seven logic gate circuit symbols, (for both ANSI and BS systems).
- Copy the NOT gate truth table.
- For CMOS logic gates, logic 1 is any voltage greater than 70% of the supply voltage, and logic 0 anything less than 30% of supply voltage.
- CMOS inputs must not be allowed to ‘float’. Always use either a ’pull-up’ or a ‘pull-down’ resistor. The resistance is unimportant. Anything from 1kΩ to 1MΩ will work.
- Complete the sentence:

  When the NOT gate input is at logic 0, the output is at logic ..., and vice-versa.

Worksheet 3
The AND function

Combinational logic

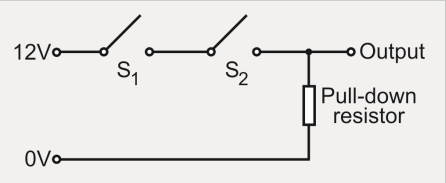
Often, electrical devices in a car, indicators for example, operate only if the ignition switch **AND** the switch for the device are both turned on.

Equally, the headlight washers may activate only when the windscreen washers are operated **AND** the headlights are on.

Here's a second logic function, the AND function!

It can be implemented using just switches, as shown, but that can make the wiring very complicated.

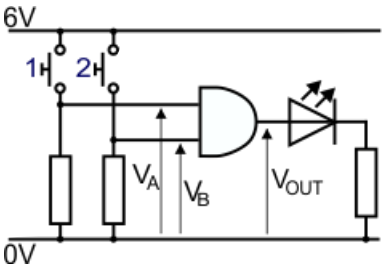
This worksheet looks at the behaviour of the AND gate.



Over to you:

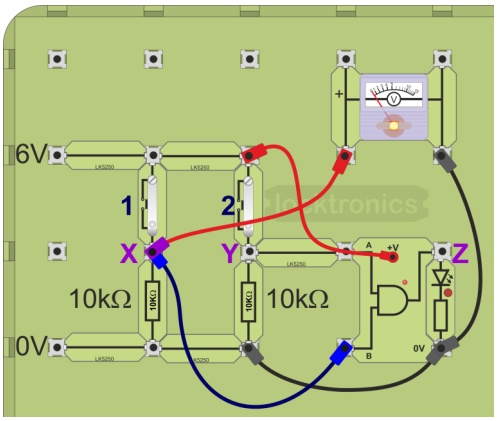
- Set up the circuit shown, with the DC power supply set to 6V.
- Notice that only one voltmeter is used.

  It can be used to make the following voltage measurements:
  - voltage V_A, at input **A** (by plugging into socket **X**, as shown);
  - voltage V_B, at input **B** (by plugging into socket **Y**);
  - voltage V_{OUT}, at the output (by plugging into socket **Z**).



- There are four sets of measurements to make.

  For the first, leave both switches off and:
  - measure V_A;
  - measure V_B;
  - measure V_{OUT}.
- Record them in the first row of the table.
- Now close switch 1, leaving switch 2 open.
- Repeat the measurements, and record them in the second line of the table.
- Continue in this way to complete the table for the other combinations of switch positions.



Inputs		AND gate		
Switch 1	**Switch 2**	V_A	V_B	V_{OUT}
Open (off)	Open (off)			
Open (off)	Closed (on)			
Closed (on)	Open (off)			
Closed (on)	Closed (on)			

Worksheet 3
The AND function

Combinational logic

So what?

As before, you used a 6V power supply, so that logic 1 is around 4.2V or so, and logic 0 less than 1.8V.
Convert your measurements to logic signals, and complete the truth-table for the AND gate.

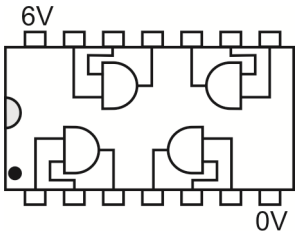
Input A	Input B	Output
0	0	
0	1	
1	0	
1	1	

The logic AND function is a straightforward one to understand. The output of the system will be logic 1 **only** when input A **AND** input B (**AND** input C etc. if there are more inputs,) are all logic 1.

In other words, the truth-table for the AND function is that shown below:

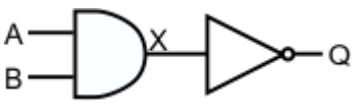
AND gate		
Input A	**Input B**	**Output**
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

One way to implement the AND function is to use an AND gate, (though there are others, as we will see.)
A CMOS 2-input AND gate chip is numbered 4081.
The pinout for this chip is shown below. Notice that there are four AND gates on the chip.



For your records:

- Copy the truth table for the AND gate.
- Complete the sentence: The output of an AND gate is at logic 1 only when ....
- Copy the diagram and complete the truth table for the arrangement shown.



Input A	Input B	X (AND gate output)	Q (Final output)
0	**0**		
0	**1**		
1	**0**		
1	**1**		

Worksheet 4
The OR function

Combinational logic

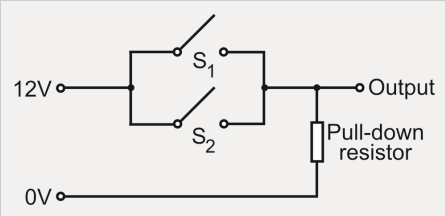
A typical car theft-alarm system has:

- door sensors to detect when a door is opened;
- a vibration sensor to detect someone breaking a window;
- a tilt sensor to warn when the car is being towed away.

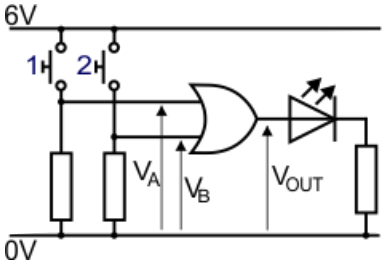
The electronic control system will switch on the alarm if the door sensor **OR** the vibration sensor **OR** the tilt sensor is triggered.

This is an application of the OR logic function.
It can be visualised using switches, as shown opposite.

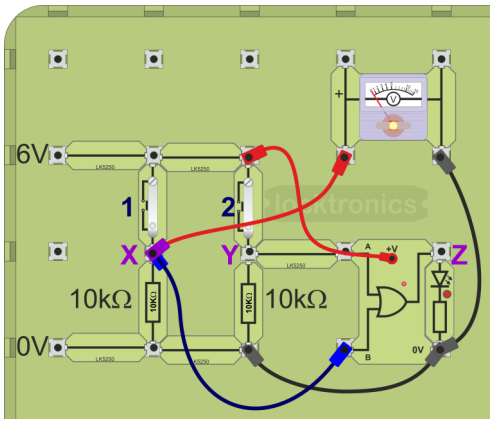
This worksheet investigates the behaviour of an OR gate.



Over to you:



- Set up the circuit shown, with the DC power supply set to 6V.
- As before, the voltmeter is used to make the following voltage measurements:
  - voltage V_A, at input **A** (by plugging into socket **X**, as shown);
  - voltage V_B, at input **B** (by plugging into socket **Y**);
  - voltage V_{OUT}, at the output (by plugging into socket **Z**).
- Again, there are four sets of measurements to make. First, leave both switches off and:
  - measure V_A;
  - measure V_B;
  - measure V_{OUT}.
- Record them in the first row of the table.
- Now close switch 1, leaving switch 2 open.
- Repeat the measurements, and record them in the second line of the table.



- Continue in this way to complete the table for the other combinations of switch positions.

Inputs		OR gate		
Switch 1	**Switch 2**	V_A	V_B	V_{OUT}
Open (off)	Open (off)			
Open (off)	Closed (on)			
Closed (on)	Open (off)			
Closed (on)	Closed (on)			

Worksheet 4
The OR function

Combinational logic

So what?

As before, you used a 6V power supply, so that logic 1 is around 4.2V or more, and logic 0 is less than 1.8V.
Convert your measurements to logic signals, and complete the truth-table for the OR gate.

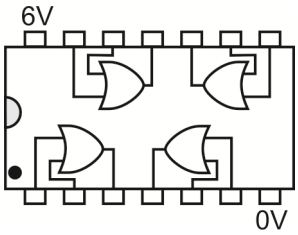
Input A	Input B	Output
0	0	
0	1	
1	0	
1	1	

The logic OR function is also straightforward to understand.
The output of the system will be logic 1 **only** when input A **OR** input B (**OR** input C etc. if there are more inputs,) is logic 1.

In other words, the truth-table for the OR function is that shown below:

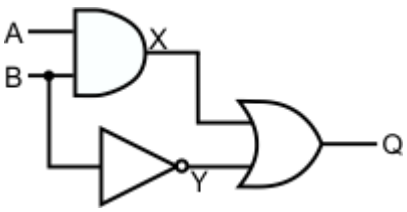
OR gate		
Input A	**Input B**	**Output**
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

One way to implement it is to use a CMOS 2-input OR gate chip, numbered 4081.
The pinout for this chip is shown below. Notice that there are four OR gates on the chip.



For your records:

- Copy the truth table for the OR gate.
- Complete the sentence: The output of an OR gate is at logic 1 when ....
- Copy the diagram and complete the truth table for the arrangement shown.



Input A	Input B	X (AND gate output)	Y (NOT gate output)	Q (Final output)
0	**0**			
0	**1**			
1	**0**			
1	**1**			

Worksheet 5
NAND, NOR and EXOR

Combinational logic

In the 19th century, Boolean algebra was developed. Even though it predated electronics by a century or so, it now allows us to design and analyse systems of logic gates, such as that shown, which allows remote control of some functions in a car.

The table shows the shorthand symbols used for the logic functions.

	NOT	AND	NAND	OR	NOR	EX-OR	EX-NOR
Symbol	$\overline{}$	$\cdot$	$\overline{\cdot}$	$+$	$\overline{+}$	$\oplus$	$\overline{\oplus}$
Example	$\overline{X}$	$A \cdot B$	$\overline{A \cdot B}$	$A+B$	$\overline{A+B}$	$A \oplus B$	$\overline{A \oplus B}$

There are some unusual consequences for those used only to 'normal' maths.

For example: **A + 1 = 1**

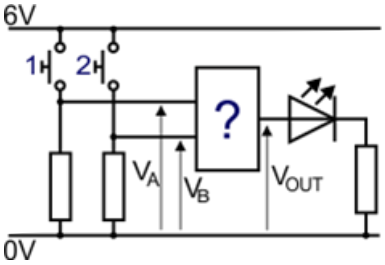
(A 2-input OR-gate with one input held at logic 1 always has an output of logic 1 no matter what signal ('A') is connected to the other input.)

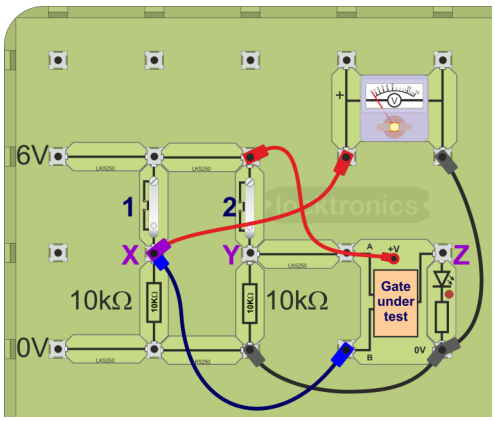
A . A = A

(When a 2-input AND gate has the same signal ('A') connected to both inputs, the output copies that signal. (If 'A' = logic1, the output will be logic 1; if 'A' = logic 0, the output will be logic 0.)

Over to you:

- The circuit used to investigate the AND and OR gates can be used for the NAND, NOR and EXOR gates too.
- The method is the same as that in the previous worksheet.
- For each gate, the procedure is measure:
  - V_A;
  - V_B;
  - V_{OUT}.
- Do so four times:
  - with both switches off;
  - with switch 1 off and switch 2 on;
  - with switch 1 on and switch 2 off;
  - with both switches on;
- Record the results in the table.





Inputs		..... gate		
Switch 1	Switch 2	V_A	V_B	V_{OUT}
Open (off)	Open (off)			
Open (off)	Closed (on)			
Closed (on)	Open (off)			
Closed (on)	Closed (on)			

Worksheet 5
NAND, NOR and EXOR

Combinational logic

So what?

As before, using a 6V power supply, logic 1 is bigger than 4.2V and logic 0 is less than 1.8V.

- Convert your measurements to logic signals, and complete the following truth-table

NAND gate		
Input A	**Input B**	**Output**
0	0	
0	1	
1	0	
1	1	

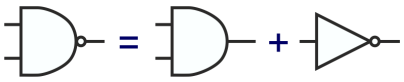
NOR gate		
Input A	**Input B**	**Output**
0	0	
0	1	
1	0	
1	1	

EXOR gate		
Input A	**Input B**	**Output**
0	0	
0	1	
1	0	
1	1	

- Compare the output columns for the AND gate (worksheet 3) and for the NAND gate. You should find that they are opposites. In other words,

  NAND = NOT AND

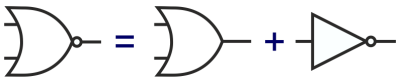
  or, using symbols,



- Compare the output columns for the OR gate (worksheet 4) and for the NOR gate. Again, they are opposites. Putting this in words,

  NOR = NOT OR

  or, using symbols,



  (We explore this idea of building a logic gate using a combination of other gates in the next worksheet.)

- The OR gate, studied in worksheet 4, is sometimes called the ***Inclusive OR*** as it includes the case where both inputs are logic 1 when outputting logic 1.
- The ***Exclusive OR*** does not set the output to logic 1 when both inputs are logic 1. It is also called a ***Non-equivalence*** gate, as it outputs logic 1 only when the inputs are different.
- (The ***Exclusive NOR*** gate is the opposite, and is called an ***Equivalence*** gate.)

For your records:

- Copy the truth tables for the NAND, NOR and Exclusive OR gates.
- Copy and complete the sentences:
  - (a) The output of the …….. gate is always the opposite of its input.
  - (b) The output of the …….. gate is at logic 1 only when both inputs are at logic 1.
  - (c) The output of the …….. gate is at logic 0 only when both inputs are at logic 1.
  - (d) The output of the …….. gate is at logic 0 only when both inputs are at logic 0.
  - (e) The output of the …….. gate is at logic 1 only when both inputs are at logic 1.
  - (f) The output of the …….. gate is at logic 0 only when the inputs are different.

Worksheet 6

Three inputs (or more)

Combinational logic

The logic gates studied so far have had only two inputs (or one in the case of the NOT gate.)

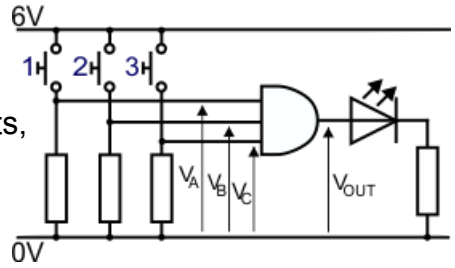
Sometimes, additional inputs are needed.

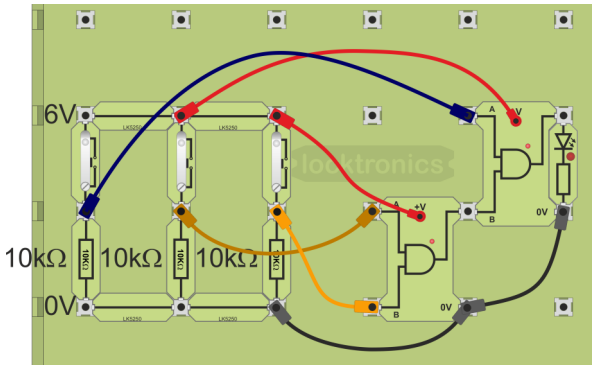
The table gives a list of some CMOS multi-input logic gates. The first word in the description, 'triple', 'dual' etc., indicates how many of the gates are found on the chip.

Logic gate	CMOS number
Triple 3-input AND	4023
Triple 3-input OR	4074
Triple 3-input NOR	4025
Dual 4-input AND	4082
Dual 4-input OR	4072
Dual 4-input NAND	4012
8-input NOR	4078

Over to you:

- The circuit used to investigate the 3-input AND gate is shown opposite.
- The method is the same as that used in previous worksheets, except that there are more switch combinations.
- Measure:
  - voltages V_A, V_B and V_C at the inputs ;
  - the output voltage V_{OUT}.
- Do so eight times, guided by the table below.
- Record the results in the table.





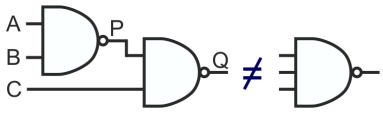
Switch 1	Switch 2	Switch 3	V_A	V_B	V_C	V_{OUT}
Open (off)	Open (off)	Open (off)				
Open (off)	Open (off)	Closed (on)				
Open (off)	Closed (on)	Open (off)				
Open (off)	Closed (on)	Closed (on)				
Closed (on)	Open (off)	Open (off)				
Closed (on)	Open (off)	Closed (on)				
Closed (on)	Closed (on)	Open (off)				
Closed (on)	Closed (on)	Closed (on)				

Worksheet 6
Three inputs (or more)

Combinational logic

So what?

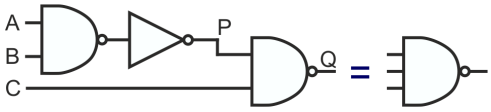
- The connection between a 2-input gate and a 3-input (or more) gate is usually obvious.
  A 2-input AND gate outputs a logic 1 signal only when input A AND input B are at logic 1.
  A 3-input AND gate outputs logic 1 only when input A AND input B AND input C are logic 1.
- Sometimes, it is not quite so straight forward to set up the 3-input gate from 2-input gates.
  For example, a 3-input NAND gate is **not** produced by the arrangement shown below:



To see that this is the case, work through the following truth table:

Input A	Input B	Input C	P	Output Q
0	0	0	1	1
0	0	1	1	0
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	1
1	1	1	0	1

Remember, for a NAND gate, when any input is at logic 0, the output is at logic 1.
A correct solution is shown below:



A challenge!
Build this arrangement and test it to confirm that it behaves like a three-input NAND gate.

For your records:

- Draw a diagram to show how to make a 3-input AND gate from two 2-input AND gates.
- Copy the diagram for the 3-input NAND gate. Create a truth table similar to that above to prove that the solution for the 3-input NAND gate works.
- Show how each of the following 3-input logic gates can be made, using combinations of 2-input gates. In each case, include a truth table to justify your design.
  - 3-input OR gate;
  - 3-input NOR gate.

  For each, build and test it to confirm that it behaves as it should.
- **A challenge!** Do the same for a 3-input EXOR gate.
  (Use the rule - any odd number of inputs at logic 1 makes the output logic 1).

Worksheet 7

Making gates from other gates

Combinational logic

All those different kinds of logic gates - how expensive!
If only we could focus on just one!

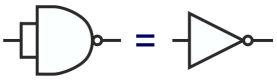
The way forward is to use one type of logic gate to build the others. Even though the result looks more complex, it may be cheaper, because:

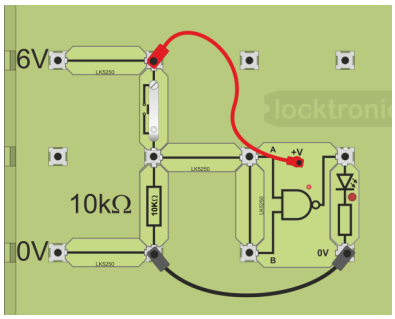
- buying in volume reduces unit costs;
- each chip may contain multiple gates - the 'complex' solution may make better use of all the gates on the chip.

Over to you:

Circuit A:

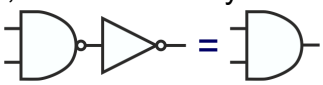
- Set up the circuit shown opposite.
  Notice the link used to connect together the two inputs of the NAND gate!
- Test it to confirm that it behaves like the NOT gate studied in worksheet 2.
  In other words, test the following identity:

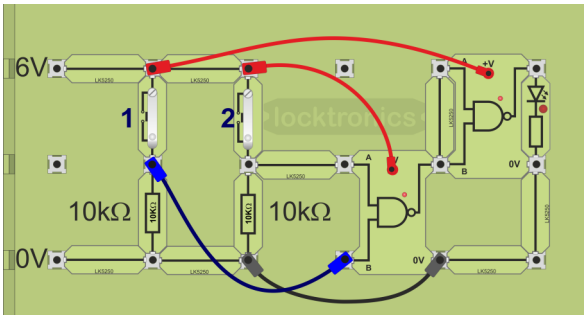




Circuit B:

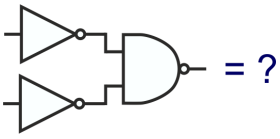
- Set up the circuit shown opposite, consisting of a NAND gate, followed by a second NAND gate set up as a NOT gate.
- Test it to confirm that it behaves like the AND gate studied in worksheet 3.
  In other words, test the identity:

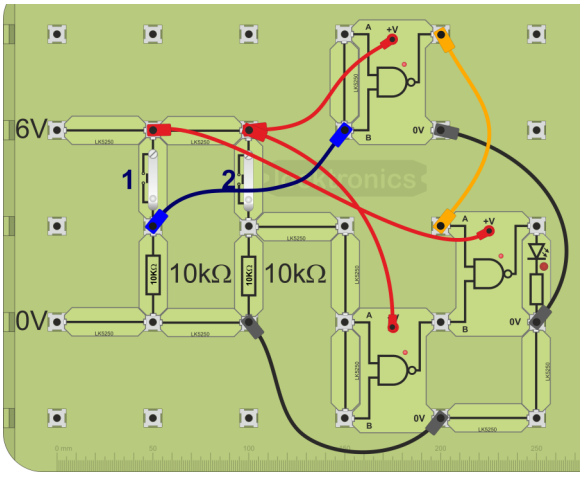




Circuit C:

- Set up the circuit shown opposite - two NAND 'NOT' gates, followed by a third NAND gate.
- Test it. What single logic gate could replace it? (Although it looks complex, it has only three NAND gates. A CMOS 4011 chip contains four, so this is still a single chip solution!)
- Complete the identity:





Worksheet 7

Making gates from other gates

Combinational logic

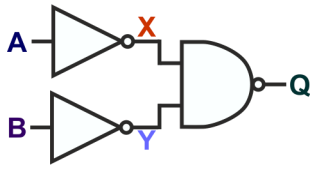
So what?

- Look closely at circuit A. The two inputs are connected together and so they must sit at the same voltage. That will be either logic 0 or logic 1. That reduces the truth table for the NAND gate to just two possible variations - both inputs at logic 0, or both at logic 1.

	NAND gate		
	Input A	**Input B**	**Output**
	0	0	**1**
Not allowed	0	1	**0**
Not allowed	1	0	**0**
	1	1	**0**

As the truth table shows, for these two possibilities, the output is the opposite to the inputs, i.e. the arrangement behaves as a NOT gate.

- In circuit B, the second NAND gate has its inputs connected together and acts as a NOT gate. This turns the output of the first NAND gate into that of an AND gate.
- Let's look closely at what is happening in circuit C, using a truth table.

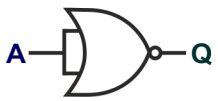


A	B	X (= $\overline{A}$)	Y (=$\overline{B}$)	Q
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1

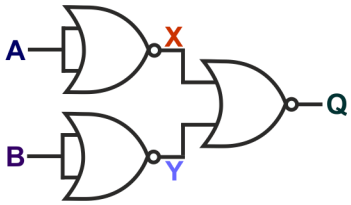
Remember that the final NAND gate outputs a logic 1 signal whenever either or both inputs are logic 0. This treatment shows that the system behaves as an OR gate.

For your records:

- Copy the diagrams that show how to create NOT, AND and OR gates from NAND gates.
- Design a system of NAND gates which behaves like a NOR gate:
  - Use only NAND gates.
  - Draw your system using correct logic gate symbols.
  - Justify your design by producing the truth table for the system.
- Alternatively, the various types of logic gate can be replaced with systems of NOR gates. For the two systems shown below, complete a truth table and hence identify the single logic gate which has the same effect.



(a)



(b)

Confirm your answers by building and testing each system.

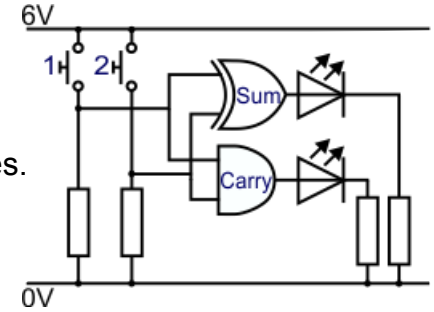
Worksheet 8
The half-adder

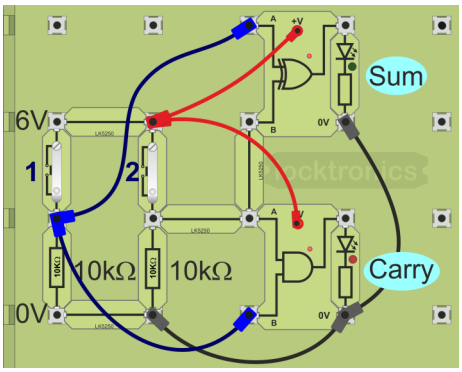
Combinational logic

The computer is everywhere!
At its heart is the CPU (central processing unit.) In turn, this relies on the ALU (arithmetic and logic unit,) which does exactly what the name suggests - carries out arithmetic and logic operations. It does so using arrays of logic gates embedded into its circuitry.
One of the basic subsystems within it is the half-adder. The circuit looks complicated, and yet it carries out a very simple task - adding together two single bit binary numbers.
That is the focus of this worksheet.

Over to you:

- Build the half-adder circuit shown opposite. (There are other ways to do it - this one highlights the connections needed.)
- Test it in the usual way, looking at all four possible input states.
- Rather than measure voltages, use the state of the LEDs to determine logic levels at the Sum and Carry outputs. (LED on ⇒ output is logic 1, etc.)
- Use your results to complete the truth table.





Switch 1	Switch 2	Sum	Carry
Open (off)	Open (off)		
Open (off)	Closed (on)		
Closed (on)	Open (off)		
Closed (on)	Closed (on)		

The binary number system uses only two digits - 0 and 1.
The table gives binary equivalents for some decimal numbers.
Although binary seems long and tedious, computers complete millions of calculations each second using this system.

Binary number			Decimal value
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	0
1	0	0	4

The final table shows the results of adding together two single digit binary numbers, **A** and **B.**

It shows, for example, that :

$1_2 + 0_2 = 1_2$; (Carry = 0_2, Sum = 1_2)

$1_2 + 1_2 = 10_2$ (= 2_{10} in decimal.)

Binary numbers		A + B	
A	B	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

(The subscript '2' is used to show that it is a binary number, and not decimal.
Decimal numbers use subscript '10'.)

Worksheet 8
The half-adder

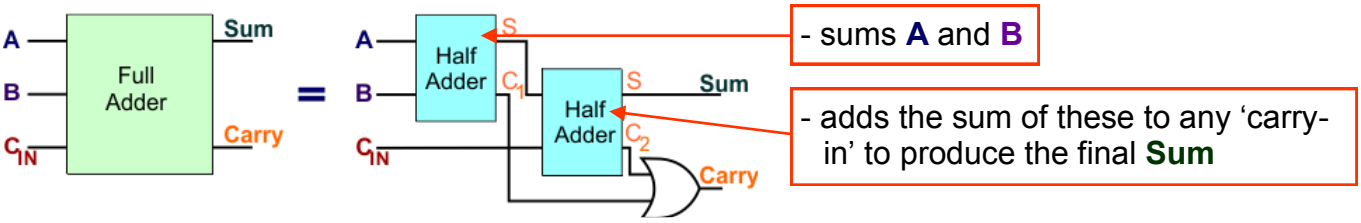
Combinational logic

So what?

The half-adder is not the full story!

The table on the previous page showed that a ‘carry’ could be generated - when both numbers are logic 1. When adding together multi-bit numbers, these ‘carry’s have to be incorporated.

Step-forward the **full-adder**! This adds together **three** single digit binary numbers - the **A** and **B** numbers, (as did the half-adder) and also a ‘carry-in’ number, C_{IN}. It can be realised using two half-adders and an OR gate:



- Either half-adder may produce a ‘carry’, hence the OR gate to combine them.
- Both cannot produce ‘carry’s at the same time - the first produces a ‘carry’ only when **A** and **B** are **both** logic 1, and then the ‘sum’ is logic 0. ($1_2 + 1_2 = 10_2$.) Regardless of the value of C_{IN}, the second half-adder cannot then also produce a ‘carry’.

<u>Example of addition :</u> $\mathbf{1101}_2 + \mathbf{1001}_2$.

The table illustrates the process:

	'8'	'4'	'2'	'1'	
A	1	1	0	1	
B	1	0	0	1	
carry-in	0	0	1		
Carry 1	0	1	1	0	Sum

- ‘1’s column - straightforward! No ‘carry-in’ from the column to the right, as there isn’t a column to the right! Only two digits to add, so a half-adder suffices.
- The ‘2’s, ‘4’s and ‘8’s columns may have a ‘carry-in’ from the columns to the right, giving three numbers to add together. These need full-adders.

To check the result, we convert the two numbers to decimals and then add them.

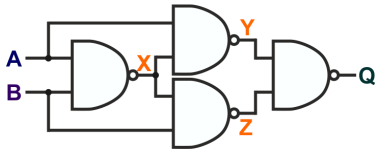
$$\mathbf{1101}_2 = 8 + 4 + 0 + 1 = \mathbf{13}_{10} \qquad \mathbf{1001}_2 = 8 + 1 + 0 + 0 = \mathbf{9}_{10}$$

so the final answer is $\mathbf{13} + \mathbf{9} = \mathbf{22}_{10}$ (= 16 + 0 + 4 + 2 + 0 and so $\mathbf{10110}_2$, as in the table.)

For your records:

- Draw diagrams to show how:
  - a half-adder can be made from an EXOR gate and an AND gate;
  - a full adder can be made from two half-adders and an OR gate.

  Combine with another group of students and build a full-adder!
- Confirm that the arrangement shown is an EX-OR gate by:
  - completing the truth table,
  - building and testing the system.



A	B	X	Y	Z	Q
0	0				
0	1				
1	0				
1	1				

Worksheet 9
Encoders and decoders

Combinational logic

Encoders and decoders are common applications of combinational logic, converting raw data into a form more suitable for a particular situation. They also allow data compression, speeding transmission and reducing the cost of hardware.

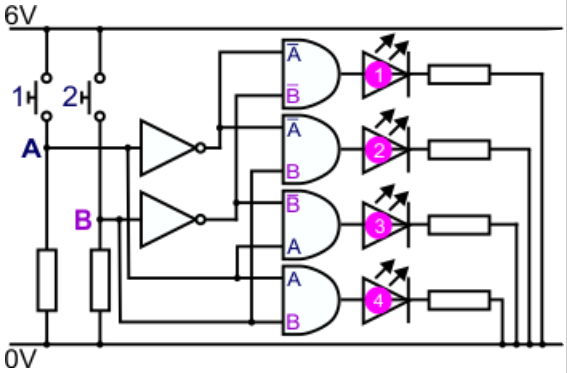
Encoders:

- convert audio/video data into standard formats, e.g mpeg;
- identify a user using a PIN or password to enhance security;
- allow some signal sources priority over others, e.g. interrupts;
- allow multiple signal sources simultaneous use of one transmission channel .

Decoders carry out the reverse process to recover the original data.

Over to you:

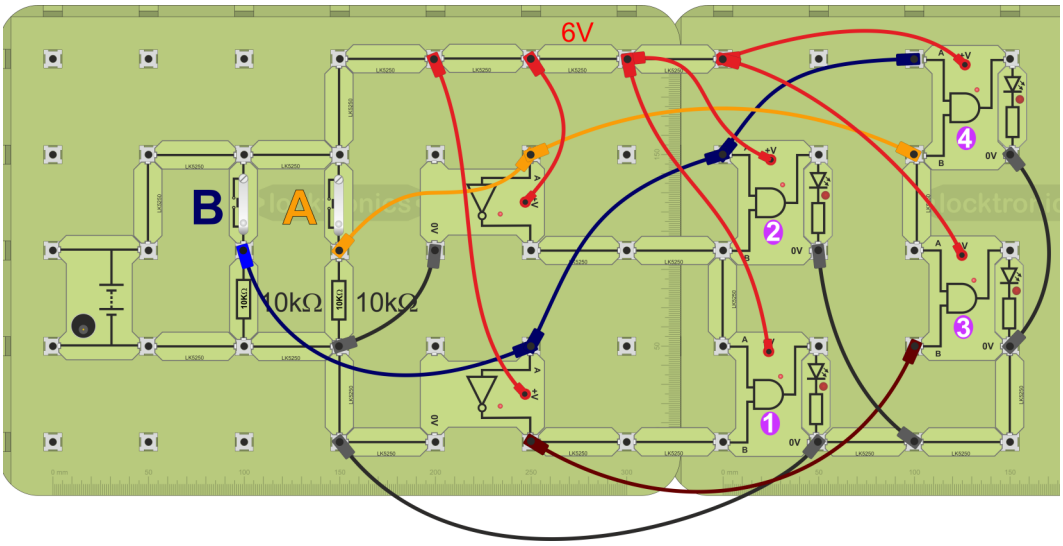
This exercise investigates a 2:4 decoder, where a two-bit number at the input selects one of four outputs.



- Build the decoder.
  The layout shown below needs two baseboards as it uses the plug-top power supply and carrier.
- Test it in the usual way, with all four input states.
- Record your results in the table.

Input state		LED lit
A	B	
0	0	
0	1	
1	0	
1	1	

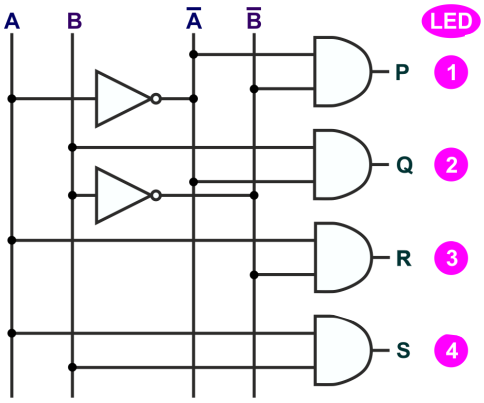
Each input should cause only one LED to light - and a different one each time.) In other words, the number **AB** is decoded to select one subsystem, (LED in this case.)



Worksheet 9
Encoders and decoders

Combinational logic

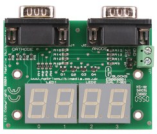
So what?



- One way to view the system you just investigated is shown opposite.
  - **P** = $\bar{A}.\bar{B}$ so LED **1** lights when **A** = 0 and **B** = 0;
  - **Q** = $\bar{A}.B$ so LED **2** lights when **A** = 0 and **B** = 1;
  - **R** = $A.\bar{B}$ so LED **3** lights when **A** = 1 and **B** = 0;
  - **S** = **A**.**B** so LED **4** lights when **A** = 1 and **B** = 1.
- Each LED is lit by only one input combination.

Examples of decoder applications:

- Within a computer system:
  - Address decoding:
    The system just described is an example of address decoding. Each LED has an 'address'. The input **AB** is used to select a particular LED.
    More generally, a number of input lines, called an address bus, ('bus' = bundle of associated wires,) is used to activate one of a number of devices, such as memory locations or chips. The outputs of the decoder control 'enable' or 'chip select' pins on the device.
    With '**n**' input lines, an address decoder can control $\mathbf{2^n}$ devices. For example, the system studied earlier used two wires, carrying signals **A** and **B**, and could control 4 (= 2^2) LEDs.
  - Instruction decoding:
    Computer programs consist of series of instructions, linked to data. Eventually, the instructions are written in 'machine-code', binary numbers. The computer translates this code into a series of actions, using an instruction decoder. Here, the incoming binary number selects the area of memory containing the required actions.
- Elsewhere:
  - Data is often transmitted in an encoded format, for security or to reduce the effects of distortion and noise. At the receiver, a decoder is used to return the data to its original format.
  - A 7-segment decoder/driver is used to convert a 4-bit binary number into the signals needed to light the correct LEDs in a 7-segment display.
    The photograph shows four of these on a 'Matrix E-Blocks' board.



For your records:

- Copy the circuit diagram showing how a 2:4 decoder can be made from four 2-input AND gates and two NOT gates.
- Design, build and test a 3:8 decoder, using three NOT gates and eight 3-input AND gates.
- Describe how a multiplexer chip could be used to carry out the same task.
  Your answer should include a diagram and an explanation of how your system works.
  (Use the internet, or a suitable text book to research your answer.)

Instructor Guide

Combinational logic

About this course

Introduction

The course is essentially a practical one. Locktronics equipment makes it simple and quick to construct and investigate electrical circuits. The end result can look exactly like the circuit diagram, thanks to the symbols printed on each component carrier.

Aim

The course introduces students to individual logic gates, and how they can be constructed from other logic gates - a form of combinational logic. Further combinational logic systems and their applications are then investigated.

Prior Knowledge

It is recommended that students have followed the ‘Electricity Matters 1’ and ‘Electricity Matters 2’ courses, or have equivalent knowledge that enables them to construct circuits from circuit diagrams and take measurements on them using voltmeters and ammeters .

Learning Objectives

On successful completion of this course the student will be able to:

- recall that an analogue quantity is one that copies the behaviour of another;
- recall that an analogue signal can have any voltage value, usually between the voltages of the power supply rails;
- recall that a digital quantity has only two possible states, known as ‘off’ and ‘on’ or ‘logic 0’ and ‘logic 1’;
- recall that digital signals can be regenerated, to remove the effects of noise and distortion, whereas analogue signals cannot;
- use a LED and series resistor to test the output state of a logic system;
- set up a switch unit to output a logic 1 signal when the switch is pressed, and logic 0 when not pressed;
- set up a switch unit to output the inverse behaviour;
- test and hence identify a logic function using two switch units and a LED unit;
- identify a logic gate from its symbol, using either ANSI or BS symbols;
- complete the truth tables that describe NOT, AND, NAND, OR, NOR and EXOR logic functions;
- recognise, and describe the behaviour of NOT, AND, NAND, OR, NOR and EXOR logic gates;
- complete a truth table for a combination of up to three logic gates;
- connect NAND gates to perform the following logic functions: NOT, AND, OR, NOR and EXOR;
- give an advantage for replacing logic gates with their NAND gate equivalent;
- create 3-input AND, OR, NAND, NOR and EXOR gates using combinations of 2-input logic gates;
- describe the function of a half-adder subsystem;
- draw the circuit diagram for a half-adder built from an EXOR gate and an AND gate;
- distinguish between the functions of a half-adder and a full-adder;
- draw the circuit diagram to show how a full-adder can be built from two half-adders and an OR gate;
- add together two 4-bit binary numbers;
- describe four applications of encoders and decoders;
- draw the circuit diagram and construct the circuit for a 2:4 decoder;
- describe how a multiplexer can be used as a 3:8 decoder.

Instructor Guide

Combinational logic

What the student will need:

To complete the combinational logic course, the student will need the following equipment:

- 2 LK8900 Locktronics Baseboard (1)
- 21 LK5250 connecting links (12)
- 1 LK5603 4mm to 4mm lead red (1)
- 5 LK5604 4mm to 4mm lead black (5)
- 2 LK5607 4mm to 4mm lead yellow (2)
- 3 LK5609 4mm to 4mm lead blue(3)
- 3 LK5203 10kΩ resistor carriers (2)
- 1 LK6231 50KΩ resistor (0)
- 3 LK6207 push-to-make switch carriers (2)
- 1 LK7290 phototransistor (0)
- 2 LK6635 red LED carrier (1)
- 2 LK6636 green LED carrier (2)
- 2 LK6862L NOT gate carriers (2)
- 4 LK6860L AND gate carriers (4)
- 1 LK6861L OR gate carrier (1)
- 3 LK6863L NAND gate carriers (3)
- 3 LK6864L NOR gate carrier (3)
- 1 LK6865L EXOR carrier (1)
- 2 LK3982 0 - 15V voltmeter carrier (1)
- 1 HP2666 Power supply (0)
- 1 LK8275 Power supply carrier (0)

Numbers in brackets are the quantities in the LK6904 Combinational Logic add-on kit.

Using this course:

The series of experiments in this course should be integrated with teaching to introduce the theory behind it, and reinforce it with written examples, assignments and calculations.

The worksheets should be printed / photocopied / laminated, preferably in colour, for the students' use. They should then make their own notes, and copy the results tables and sections marked 'For your records' for themselves. They are unlikely to need their own permanent copy of the worksheets.

Each worksheet has:

- an introduction to the topic under investigation;
- step-by-step instructions for the investigation that follows;
- a section headed 'So What', to collate and summarise results, offer extension work and encourage development of ideas, through collaboration with partners and with the instructor.
- a section headed 'For your records', to be copied and completed in students' exercise books.

This format encourages self-study, with students working at a rate that suits their ability. It is for the instructor to monitor that students' understanding keeps pace with their progress through the worksheets. One way to do this is to 'sign off' each worksheet, as a student completes it, and in doing so have a brief chat with the student to assess grasp of the ideas involved in the exercises it contains.

Time: It should take students between 6 and 8 hours to complete the worksheets.
A similar length of time will be needed to support the learning that takes place as a result.

Instructor Guide

Scheme of Work

Combinational logic

Worksheet	Notes for the Instructor	Timing
1	The first aim is to distinguish between analogue and digital signals. This worksheet sets up two sensing sub-systems, one analogue and the other digital. Students take voltage measurements on the analogue voltmeter. Some may not have done so for some time, and may need a reminder of how to do so. No record of measurements is taken. They allow the student to contrast the performances of the two systems - to realise that the analogue signal can take any value between 0V and 6V, the power supply voltages, whereas the digital signal has one of two voltage values. The 'So What' section points that, because of the nature of electrical signal transmission, digital signals use bands of voltages rather than specific values. In a TTL system, (Transistor-Transistor-Logic, one of the logic gate families,) for example, any voltage from 0V to 0.8V is guaranteed to be taken as logic 0, and anything from 3.5V to 5V (the maximum voltage for a TTL system,) as logic 1. Signals may have voltages between these bands (unfortunately,) in which case the outcome is ambiguous. The system will regard these as either logic 0 or logic 1, but exactly which is uncertain and may vary from system to system, and even from day to day. The section also introduces the idea of regeneration, that a digital signal can be returned to its original state, removing the effects of added noise signals, and of distortion, (where the components of the system do not reproduce the signal accurately.) This is not possible with an analogue signal. (The other advantage of digital processing is that it allows error detection, and correction, whereas analogue signals do not.) The logical nature of logic circuits is demonstrated as the last activity of this worksheet. Students find that if you turn the switching unit upside down, you turn the signal 'upside down'. Initially, pressing the switch generated a logic 1 signal, and not pressing it a logic 0. Once the switch unit is inverted, pressing the switch generates a logic 0, and not pressing it a logic 1!	**20 - 30 mins**
2	This worksheet introduces the first, and simplest, logic gate, the NOT gate. Before that, the introduction makes the necessary distinction between logic gates and logic functions. The more important is the logic function. There are a number of ways to implement a logic function. In digital electronics, you can use a dedicated, discrete logic gate, you could use a series of NAND gates, or NOR gates, or use a programmable system. On a wider front, optical logic gates produce the same logic functions, but using laser light, to speed up the switching process. The technology may differ, but all produce the same outcomes in terms of logic functions. The introduction also contains an important table of logic symbols, both in the ANSI format, and the BS (sometimes called SB) format. The students may encounter further formats, such as the IEC (International Electrotechnical Commission) system. The investigation involves setting up a switch unit and using it to generate a digital input signal for a NOT gate. The students use this to construct a voltage truth-table for the NOT gate. They then invert the switch unit, but observe that this has no effect on the NOT function itself. Throughout this course, their measurements are recorded in tables given on the worksheets. These should be copied into their own notebooks and completed there with their measurements. Continued on the next page...	**30 - 40 mins**

Instructor Guide

Scheme of Work

Combinational logic

Worksheet	Notes for the Instructor	Timing
2	Continued from previous page… The circuit layout in the worksheet shows the use of two voltmeter carriers. However, one voltmeter carrier can be 'hopped' from the 'input voltmeter' position to the 'output voltmeter' position, provided it is replaced by connecting links, one to connect the switch unit to the input of the NOT gate, and the other to connect the 0V power supply terminal to the lead connected to the 0V terminal of the NOT gate carrier. The 'So What' section details the voltage bands used by CMOS gates (like that used on the Locktronics NOT gate carrier,) and the students use this information to turn their voltage measurements into logic levels, and re-build the truth-table. The students then investigate what happens when the resistor is removed from the switch unit. In general, this is an unwise move for CMOS gates. The inputs operate on minute currents and so can be affected by stray electromagnetic fields, such as radio, television and mobile phone transmissions. As a result, the inputs can switch rapidly between logic 0 and 1. As they do so, they draw enough current to cause local overheating, which can damage the IC. The rule, then, is that CMOS inputs should not be allowed to 'float', but instead must be either 'pulled down' to the 0V rail, or 'pulled-up' to the positive power rail, by a resistor. The Locktronics NOT carrier uses a large value resistor, housed within the carrier, to connect unused inputs to 0V.	**30 - 45 mins**
3	This worksheet investigates the behaviour of an AND gate. It introduces two situations where the AND function might be encountered in a car. The introduction points out a simple way to view the AND function as two switches in series. It is worth the instructor spending time to drive home this picture. The diagram includes a pull-down resistor, to ensure that output sits at logic 0 when either switch is open. Again, the significance of this needs to be emphasised. The students set up two switch units and use them to input four combinations of logic signals. Measuring input and output voltages, they complete a table of results, again copied into their notebooks. They then turn these into logic levels and generate the AND gate truth-table. They are encouraged to view the AND function as one which generates a logic 1 output **only** when both inputs are at logic 1. The 'So What' section includes the pinout for a CMOS 4081 IC. Part of the importance is to highlight again the need to avoid floating inputs. Zealous students can take this message too far, and connect unused **outputs** to the nearest power rail. This is unfortunate, because the logic applied by the gate may try to drive the output to logic 1, while the student has connected it to the 0V rail, or vice-versa. The message then is that the outputs take care of themselves. It is only unused inputs that require our attention. In the 'For your records' section, they are asked to create the truth table for a combination of a NOT gate and an AND gate, their first foray into the world of combinational logic.	**20 - 30 mins**

Instructor Guide

Scheme of Work

Combinational logic

Worksheet	Notes for the Instructor	Timing
4	A similar approach is now used to investigate the OR gate. The introduction points out a way to view the OR function as two switches in parallel. Again, the diagram includes a pull-down resistor, to ensure that output sits at logic 0 when both switches are open. The significance of both points needs to be emphasised by the instructor. It discusses a typical application, the car security system, though the details of the sensors used will decide what logic function is needed. The procedure is the same as for the previous investigation. The students set up two switch units and generate four different sets of logic signals in turn. They complete a table of results, showing the four sets of input voltages with corresponding output voltages and then turn these into logic levels to create the OR gate truth-table, expressed in words as a logic function which generates a logic 1 output when either input is at logic 1. In the 'For your records' section, they are asked to create the truth table for a combination of a NOT gate, an AND gate and an OR gate. Though not mentioned as such, the result illustrates 'redundant term' simplification in Boolean algebra, as the final output is actually 'A + B' - the 'NOT A' term is redundant.	**20 - 30 mins**
5	The introduction shows students the shorthand form of Boolean operators. It goes on to show that there is ample scope for confusion with the operators used in 'normal' maths. The instructor may need to spend time talking through the two examples given, if the students are to grasp this issue. The procedure used in the previous worksheets is followed to produce truth tables for the NAND, NOR and EXOR logic functions. In the 'So what' section, the results are used to show that the NAND function can be considered as the inverse of the AND (e.g. an AND gate followed by a NOT gate,) while the NOR function is the inverse of OR (e.g. an OR gate followed by a NOT gate). The EXOR (exclusive-OR) gate is not related, despite its name. The OR function studied in worksheet 4 is sometimes known as the Inclusive-OR, as it includes the case where both inputs are at logic 1, to generate a logic 1 output signal. On the other hand, the EXOR excludes this case. In some ways, a more profitable way to view the performance of the EXOR is as a non-equivalence function - it outputs logic 1 to show that the inputs are in different logic states, (one at logic 1, the other at logic 0.) (In similar fashion, the EXNOR gate can be viewed as an equivalence gate, outputting logic 1 when the inputs are the same). The 'For your records' includes a cloze exercise where students identify the logic gate described in each statement. This is a conclusion to the first part of the course, identifying the nature of different logic functions.	**45 - 60 mins**
6	In moving to logic gates with more than two inputs, the logic functions themselves do not change. It is possible to find chips that contain these gates - the '4073' chip houses three 3-input AND gates, the '4002' contains two 4-input NOR gates. As chips contain four 2-input gates, it can be useful to use spare 2-input gates to create multi-input gates. In some cases, this is more difficult than in others. The 'So what' section explores the case of a 3-input NAND gate. The 'For your records' section goes on to cover 3-input OR and NOR gates. The 3-input EXOR is tricky as different interpretations exist as to what this is. For that reason, the students are told to use the description "Any odd number of inputs at logic 1 makes the output logic 1." to create it.	**20 - 30 mins**

Instructor Guide

Scheme of Work

Combinational logic

Worksheet	Notes for the Instructor	Timing
7	Modern electronic devices are relatively cheap even though they involve highly advanced technologies. The reason for the low cost is mass production. The same principle applies when buying logic gate chips - buying them in small numbers is expensive. As a result, it makes sense to try to stick to one type, and use that to create other logic functions when needed. This worksheet explores how to do that. It is shown that combinations of NAND gates can create NOT, AND and OR functions. In the 'For your records' section, the students go on to show how to create the NOR function from NANDs. It goes on to look at how NOR gates can be combined. These are the only gates that offer this flexibility, the reason being that both create the NOT function when their inputs are connected together. Other gates then follow. The 'So what' section uses a truth table to justify two NAND gate creations.	**30 - 40 mins**
8	This worksheet introduces the topic of binary arithmetic and the electronics used to carry it out. The first issue is that computers work so fast that they make short work of what, to us, appears tremendously cumbersome. Having investigated the half-adder, students are shown the significance of the results through a couple of binary additions. (Students need to know that the '+' here means 'plus' and not 'OR'.) The second example is the more significant, in that it includes a 'carry-out' from the calculation, which must be taken into account in evaluating the final value of the addition. The 'So what' section takes this further with the full-adder, a subsystem which can add together two 'raw' binary digits AND a 'carry-in' from a previous calculation. The instructor probably needs to support these ideas with further explanation and examples if the students are to appreciate fully the process. The example given is a starting point, but much more support work may be needed. The '4008' chip is a full-adder that adds together two 4-bit numbers (and a 'carry-in, if needed,) to produce a 4-bit sum, and a 'carry-out' digit. The 'For your records' section includes an EXOR gate from four NAND gates.	**30 - 40 mins**
9	The topic of encoding/decoding is hugely important in data transmission and reception. Logic signals are very rarely transmitted in their 'raw' form. Depending on the type of data and the transmission medium involved, they are encoded into a different form, to enhance security, reduce the effect of noise, make the transmission suitable for the medium etc. At the receiver, the reverse process, decoding, takes place to recover the data in its original form. This worksheet shows how a combinational logic system can be used as a decoder. Here, the data transmitted contains the 'address' of a particular subsystem. The decoder sends a signal to that subsystem to activate it. The system can be built on one baseboard, but it involves a number of 4mm leads, and does not include any switch units. Instead the students would plug the 'A' and 'B' leads directly into the power supply connections to obtain the logic 1 and 0 signals. The subsystems to be addressed are LEDs in this case, and the student identifies which is activated by each input state. The 'So what' section uses Boolean notation to identify the four outputs of the decoder. Instructors may need to elaborate on the detail given. Examples of the use of decoders are given as well. The 'For your records' section asks the students to research the use of a multiplexer chip, such as the '4051', to act as a 3:8 decoder. Their findings could be delivered as a presentation to the class, or as a wall display.	**30 - 40 mins**

Apparatus for this course

This curriculum is intended for use with the apparatus provided in the Locktronics Combinational logic add-on kit—LK6904
Please note that this does not include all of the items needed for this worksheet. In particular...

- It is assumed that a suitable 6V DC power supply and connecting leads are already available. The supply should be capable of supplying 6 Volts at up to 500 mA.
- Additional Locktronics carriers may be required. It is assumed that the apparatus from our LK9071 'Electricity Matters' solution will be available.
- Some worksheet experiments will require groups of students to combine apparatus from more than one kit in order to build some of the more complex circuits.

Version Control

Version	Released	Notes
LK2094-80-01	2015 June 02	First public release