

# Laboratory Exercise 10 – DIGITAL and ANALOGUE ELECTRONICS

In this exercise you will use electronic ‘chips’ to build some useful circuits. The first part illustrates the use of digital integrated circuits (ICs) with a counter and display, and the second extends this to making a rudimentary timer. The third part is a study of analogue operational amplifiers (‘op-amp’) ICs, while the last part unites the analogue and digital worlds by using an op-amp to build a simple digital-to-analogue converter.

## Part A: Counter, decoder, and LED display

### Introduction

In this section you will build a circuit for counting and displaying signals, using digital integrated circuits (ICs, or ‘chips’). ‘Digital’ means that the signals have only two discrete levels, which may be called true or false, high or low, on or off. For consistency, we use binary digits 0 and 1 to indicate low and high, and represent them by voltages 0 V (‘close to zero’) and +5 V (‘about five volts’). These levels are separated by a clear margin, so they don’t have to be generated very precisely. They are called TTL signals (from Transistor–Transistor Logic). We will use so-called CMOS chips (Complementary Metal-Oxide-Silicon), which are made up of many transistors used as switches. These chips are common and inexpensive. Some are very simple, but others are extremely complex. Their operation is summarised in the *CMOS Data Book*, of which we have copies in the laboratory.

The family of chips we will mainly use is the 4000 series, with numbers of the form 4XXX. They always need to be connected both to a power source, typically labelled  $V_{DD}$  and between +3 to +15 V (we always use +5 V), and to earth (0 V), labelled  $V_{SS}$ .

CMOS devices are easily damaged by electrostatic charge, due to their very high input resistance. When not plugged into a circuit board they are normally stored on aluminium foil, conducting plastic or conducting foam. Do not rub them on man-made fabrics, and it is often a good idea to wear an earthing strap to avoid static discharge when handling them.

So-called ‘**pinout**’ diagrams show where power, earth, and all logic signals must be connected. Note that the pin numbering, as viewed from *above*, always has pin 1 at bottom left and goes around the chip *anti-clockwise*. You can tell which end has pin 1 from a small dot or notch on the chip package. (Packages like this with two rows of pins are called dual in-line, or DIL.) The pinout diagram should indicate which pins are for input signals and which for outputs. If an **input** to a chip is *not used*, you *must* nearly always connect it to an appropriate level, either low or high, to be sure that the logic will work as planned — beware of any ‘floating’ connections.

The equipment for this exercise consists of a test box with a **breadboard** on top and **power supplies** inside. It also has **switches** to give voltages corresponding to **logic 1 when up** and **logic 0 when down**, and **light-emitting diodes** which **light up when a logic 1 (high) is applied**.

The top row of holes on the breadboard are all connected together, and so is the bottom row (see figure 4 in exercise 2). Normally, you should connect the top row of holes to the +5 V supply connector on the rear of the unit, and the bottom row of contacts to the earth connector. This allows fairly short, neat connections to any part of the breadboard.

Although very simple logic can be used to construct complex circuits, it is better to use ready-made chips that integrate entire tasks. This makes for easier design and construction of practical, reliable equipment. In this part we will count and display electronic pulses using counter, decoder, and display chips. We will start at the display end and work backwards.

## Binary-coded decimal (BCD) representation

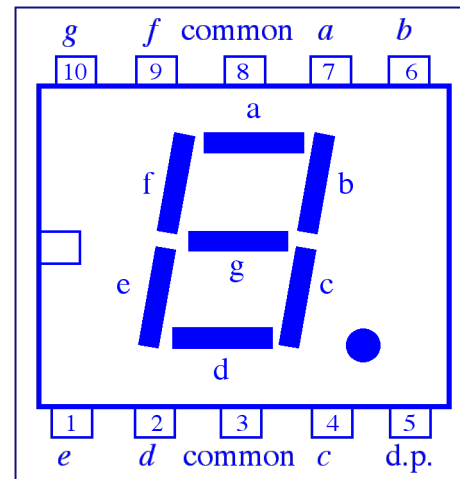
Digital logic is closely tied to the use of binary, or base 2, numbers. This is fine inside computers, but humans prefer decimal notation. One common way to do this is by using binary-coded decimal (BCD), which uses four binary bits for each decimal digit. In principle, four bits can count from 0 to 15 (i.e. 16 combinations) but in BCD the four bits, representing 1, 2, 4, and 8, only count from 0 to 9 — the combinations adding up to more than 9 are simply not used. (*If you are unfamiliar with binary notation or do not understand BCD then ask a demonstrator.*)

## 7-segment LED display

Light-emitting diodes (LEDs) behave like ordinary  $p-n$  diodes. The direction of current flow is indicated by the diode symbol. The forward current through the diode must be limited by a series resistor. The current allowed varies with the size of the diode, but a typical value is  $\sim 10\text{--}15$  mA. Thus for a +5 V supply a typical resistor would be  $\sim 330\text{--}500\ \Omega$ . (For rough calculations like this you can neglect the small resistance of the conducting diode.) LEDs are commonly available in red, yellow, green and infrared (used for remote controls and fibre optics). Blue LEDs are rarer and dearer.

Seven-segment displays for displaying numbers are very common. Each digit is made up of an array of seven LEDs in the form of strips (see figure 1), with each strip denoted by a letter between  $a$  and  $g$ . This arrangement also allows a few letters to be displayed, but this has problems since it confuses characters such as  $b$  and  $6$ , or  $D$  and  $0$  and  $O$ .)

Each of the LEDs must be supplied from a voltage source via a series resistor. The number of connections needed is minimised by having a common cathode connection to earth for all the diodes. Another simplification is to use a ‘resistor pack’ containing eight resistors in a DIL package similar to an IC. The resistors are connected between corresponding pins on opposite sides of the package.



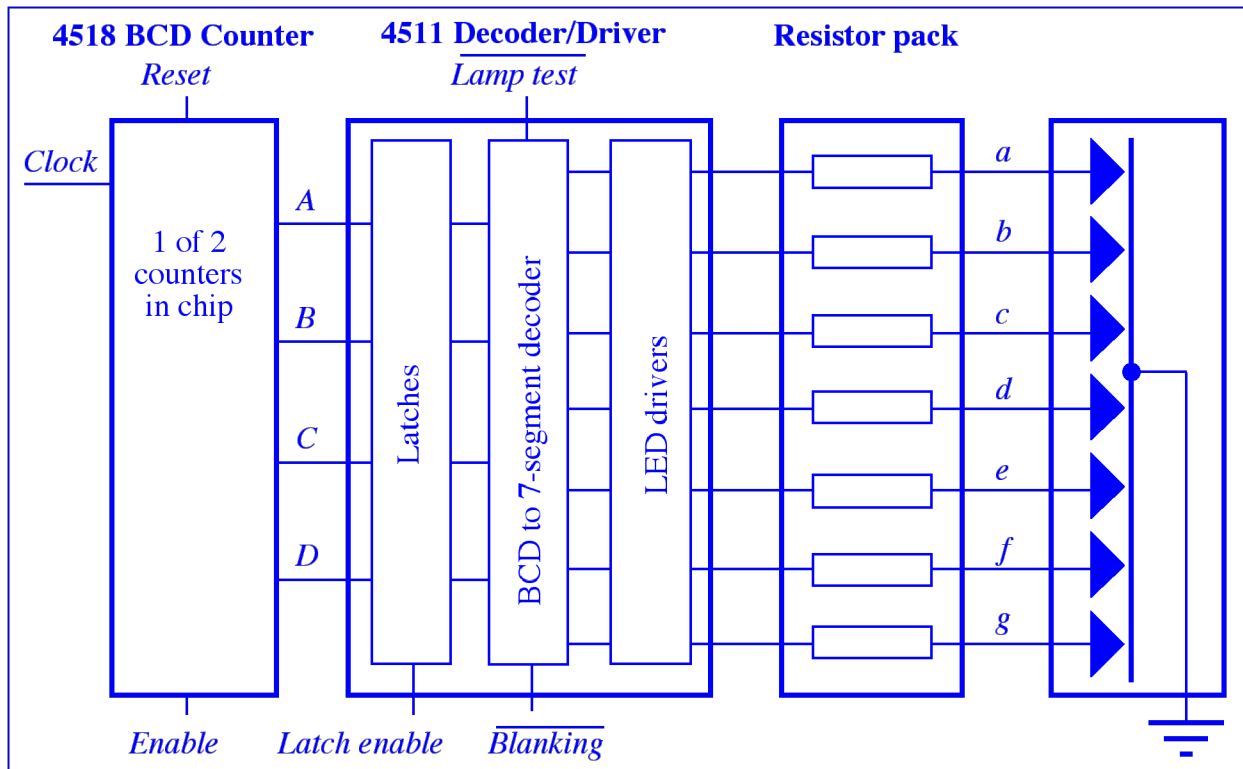
**Figure 1** Segments and pinout of LED display

➤ Start to **wire up** the circuit shown in figure 2. Plug chips into the breadboard with their *pins on either side of the central gap*. Put the **display** at the *right-hand end* of the breadboard, and the **resistor pack** ( $150\ \Omega$ ) next to it. Leave room at the left-hand end for the decoder/driver and counter chips, which each have 16 pins. Connect the *common cathode* to earth — two pins are provided but only *one* of them has to be connected. Ignore the decimal point (*d.p.*).

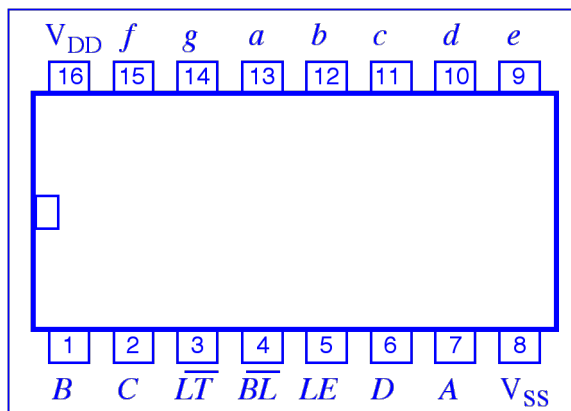
## 4511 Decoder/driver

This is a fairly complex chip; its pinout is given in figure 3. First, it **latches** (i.e. stores) binary-coded decimal (BCD) input data, so that the display remains lit even while the inputs (labelled  $A, B, C, D$ ) are changing. Next, the BCD (1, 2, 4, 8) data are **decoded** into separate signals for each of the seven segments of the display. Finally, the signals are **amplified** by drivers. These can sustain  $\leq 25$  mA each, which is ok since the LED display segments must be limited to 20 mA. This would seem to imply resistors of at least  $250\ \Omega$ , but in fact there is some resistance in the 4511 output driver, so that resistors of  $\geq 150\ \Omega$  are acceptable. (A full specification for the 4511 is available in the *CMOS Data Book*.)

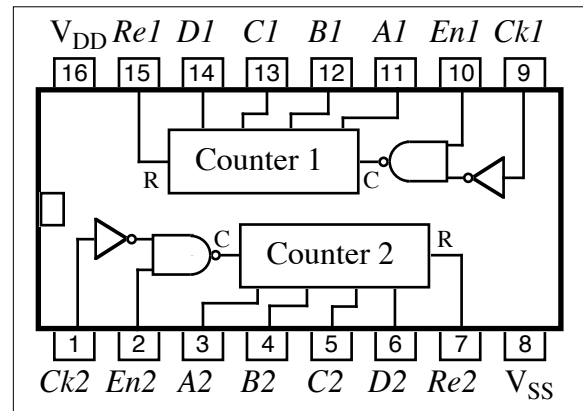
➤ **Connect the 4511** to the display via the resistor pack. **Take care** that the output connections are not shorted to earth or each other. Connect  $V_{DD}$ ,  $\overline{BL}$  and  $\overline{LT}$  to +5 V, and  $V_{SS}$  and  $LE$  to earth.



**Figure 2** Overall layout of counter, decoder, and display circuit



**Figure 3** Pinout of 4511 decoder/driver chip



**Figure 4** Pinout of 4518 dual decimal counter chip

➤ **Examine the behaviour of the 4511** using the test-box switches to supply the inputs  $A$ ,  $B$ ,  $C$ ,  $D$ , with the least significant binary digit ( $A$ ) on the right. **Check the truth table (table 1)** given below. ( $X$  means ‘don’t care’, in other words the signal can be either 0 or 1 without having any effect.) Since BCD goes only from 0 to 9, find out what happens if ‘invalid’ codes 10, 11, 12, 13, 14, 15 are applied.

➤ Be sure also to **check the blanking** ( $\overline{BL}$ ), **lamp test** ( $\overline{LT}$ ), and **latch enable** ( $LE$ ) features of the chip to understand what they do. Note that  $\overline{BL}$  and  $\overline{LT}$  are ‘active low’ (i.e. they cause things to happen when they are at 0 V, not +5 V), while  $LE$  is ‘active high’.







### 4518 BCD counter

We will count so-called ‘clock’ pulses using half of a 4518 dual decimal counter. (We ignore the other counter for now.) This works by counting in binary, but some extra logic causes it to reset when it has counted to 10 so that we get BCD. Its pinout is given in figure 4, and full details are given in the *CMOS Data Book*. (In the book,  $A$  is called  $Q1$ ,  $B$  is  $Q2$ , etc.) The counter increments either on positive-going edges of the *Clock* ( $Ck$ ) input or negative-going edges of the *Enable* ( $En$ ) input, and is reset by the *Reset* ( $Re$ ) input. This is summarised in **table 2**.

**Table 1** Truth table for 4511 Decoder/Driver

Inputs							Outputs							
<i>LE</i>	$\overline{BL}$	$\overline{LT}$	<i>D</i>	<i>C</i>	<i>B</i>	<i>A</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	Display
X	X	0	X	X	X	X	1	1	1	1	1	1	1	8
X	0	1	X	X	X	X	0	0	0	0	0	0	0	
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	1	1	0	1	1	0	1	2
0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
0	1	1	1	0	0	1	1	1	1	0	0	1	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	
0	1	1	1	0	1	1	0	0	0	0	0	0	0	
0	1	1	1	1	0	0	0	0	0	0	0	0	0	
0	1	1	1	1	0	1	0	0	0	0	0	0	0	
0	1	1	1	1	1	0	0	0	0	0	0	0	0	
0	1	1	1	1	1	1	0	0	0	0	0	0	0	
1	1	1	X	X	X	X								

**Table 2** 4518 BCD counter actions

Clock	Enable	Reset	Action
	1	0	Increment counter
0		0	Increment counter
	X	0	No change
X		0	No change
	0	0	No change
1		0	No change
X	X	1	$A = B = C = D = 0$

➤ Connect earth, then connect the *A1*, *B1*, *C1*, *D1* outputs of the 4518 to the corresponding inputs of the 4511, and then connect +5 V. Connect *En1* to one of the test-box switches, set to 1, and *Re1* to another switch, set to 0.

➤ **Test** the counter by using a breadboard switch, connected to *Ck1*, as a source of input pulses. Check the operation of *Enable* and *Reset*. Then **speed it up** by changing the input to use an oscillator at a rate of less than 10 Hz. (If your oscillator will not go to a low enough frequency, you can use the other half of the dual counter, Counter 2, to divide the pulse rate by 10. Connect the output *D2* of Counter 2 to *En1* of Counter 1, and earth the *Ck1* input. The output *D2* goes from 1 to 0 whenever 10 pulses are counted, and this increments the second counter.)

**DO NOT DISMANTLE YOUR CIRCUIT – IT IS NEEDED FOR PART B**

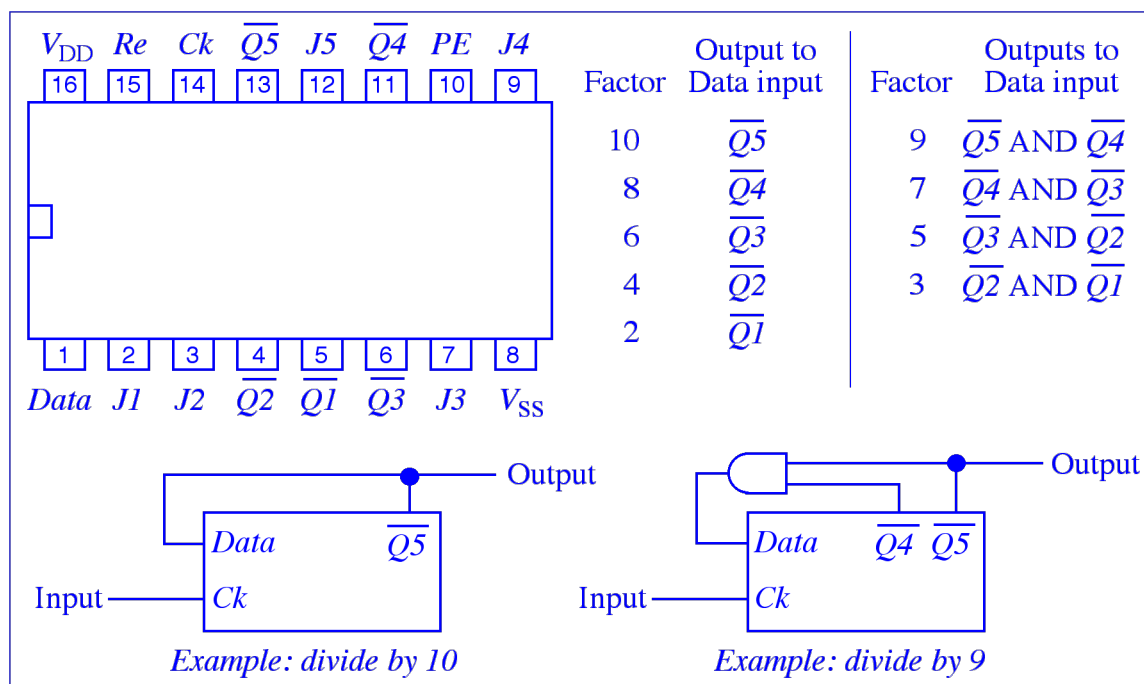
## Part B: A seconds timer

### Introduction

In this section you will study the 4018 divide-by- $N$  counter chip, which has facilities for dividing the number of input pulses by a programmable factor ranging between two and ten. You will then use it, together with a circuit to rectify a 50 Hz sine-wave voltage derived from mains electricity, to turn your counter into one that counts seconds.

### 4018 Divide-by- $N$ counter

This is a 5-stage counter with facilities for dividing the input by any factor between two and ten. It is made more complex by having a ‘preset’ facility, to begin counting at any 5-bit (i.e.  $\leq 32$ ) binary number as well as zero. A pinout diagram and information on its division options are in figure 5, while a diagram showing how the logic works is given in the *CMOS Data Book*.



**Figure 5** Pinout and division options for 4018 divide-by- $N$  chip

The various divide options are selected by connecting different  $\overline{Q}$  outputs back to the Data input. For example,  $\overline{Q5}$  selects division by 10. In each case, the divided-down output appears on the  $\overline{Q}$  pin that has been fed back to the input. For example, to see the input pulse train divided by 10 you use the  $\overline{Q5}$  output. Note that division by even factors simply requires a connection, while for odd factors two outputs must go through an additional AND gate, e.g. 4081. (For odd-factor output, look at the higher-numbered  $\overline{Q}$  pin.) Examples are shown in figure 5.

For normal operation the *Preset Enable (PE)* is held low. To start counting at a non-zero value, *Preset Enable* is set high and the desired binary number is put onto the five *Jam (J)* inputs. We will *not* test this facility.

➤ Connect up the 4018 to the 4518 counter and LED display from part A. On the 4018, *Preset Enable* should be earthed, and the *Jam* inputs can be left unconnected. The TTL output of the signal generator should be connected to the Clock input. Reset (*Re*) can be connected to a switch, normally low, to allow the chip to be cleared. The output being used should be connected to the 4518 counter. However, because the outputs are inverted we'd like the 4518 to

increment on negative-going pulses. To do this, connect the 4018's output to the Enable input of the 4518, and connect the Clock input of the 4518 to earth (see table 2).

➤ Test the chip dividing by an **even** factor. To do this, display the pulses going in and coming out on a scope, and also count the outputs using the counter and LED display. Note that the outputs are inverted. When you have this working, divide by an **odd** factor as well.

## Rectification of AC and conversion to TTL pulses

A half-wave rectifier using the 9 V AC supply is shown at the left of figure 6. *Be sure to use the AC outputs of the supply and not the earth output.* This rectifier only lets through the positive half of the sine-wave mains cycle, since the diode chops off the negative half. The symbol shown in the middle is a Zener diode, a device that does not allow the voltage across its terminals to exceed a given value. We use a Zener diode here to limit the voltage to a little under +5 V, 4.7 V to be precise. This should give us square-ish 50 Hz pulses that can drive TTL logic, since the 'low' should be close enough to 0 V. The circuit must also include a resistor of  $\sim 100 \Omega$  to limit the current through the Zener diode when it starts to conduct. We finally add a simple TTL circuit to clean up the pulses; almost anything would do but a very simple choice is a 7404 inverter since it only needs one input to produce an output. Use pin 1 as input, pin 2 as output, connect +5 V to pin 14, and earth pin 7. The TTL input and output need load resistors of  $\sim 1 \text{ k}\Omega$ .

➤ **Build this circuit** and **examine** the waveform produced on the scope. Sketch both the original and the cleaned-up pulses. These 50 Hz pulses will form the basis of the seconds timer in the next section.

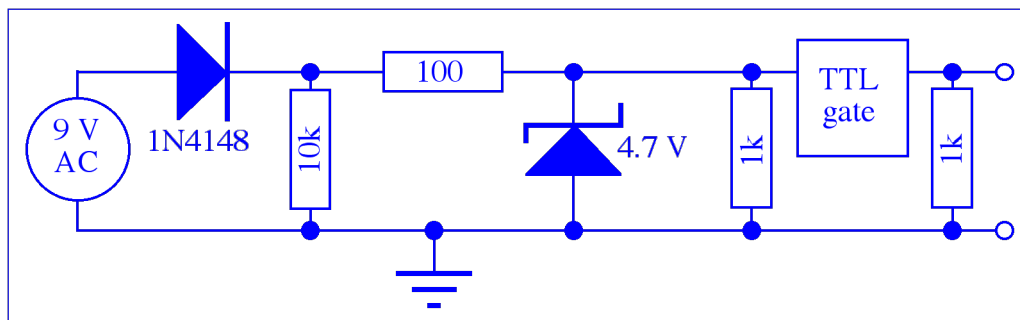


Figure 6 AC to TTL converter, including 'clean-up' circuit

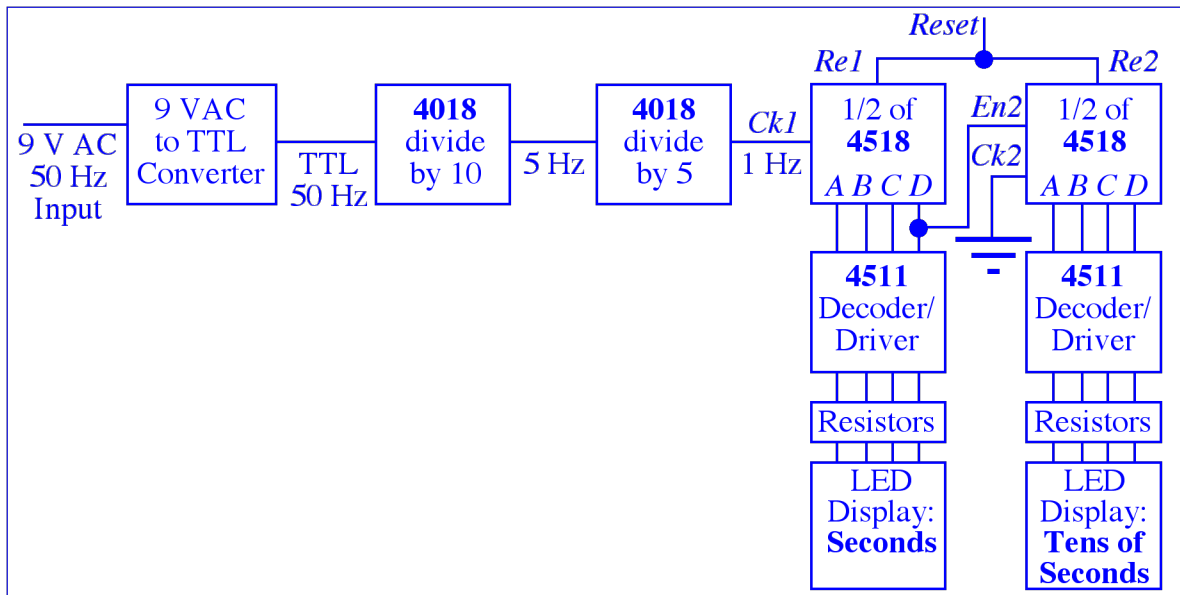
## Seconds timer

The source of the timing information is 50 Hz AC, converted to TTL as above. The pulse rate must then be reduced by a factor of 50, using two 4018s to divide first by 10 and then by 5 to give 1 Hz. This can then be fed to half of the 4518 dual decade counter, which will drive the seconds display. A block diagram of the entire system is shown in figure 7. For a ten-second timer you do not need any of the circuits along the right-hand side of the diagram; these are used for the *optional* 100-second timer extension that follows.

➤ **Build** and **test** the ten-second timer circuit.

If you are feeling ambitious and have some time, this can be extended into a two-digit 100-second timer. This requires the use of two LED displays and 4511 decoder/drivers, as in figure 7. Production of the 1 Hz signal is just as before. It is fed to half of the 4518 dual decade counter, which drives the seconds digit. In order to get the tens-of-seconds digit you can also use this half of the 4518 to divide the rate by 10, as described at the end of part A. Use the other half of the 4518 to drive the tens-of-seconds display. Use a common reset signal for both digits.

➤ **Optional (for extra credit): Build and test** this circuit.



**Figure 7** Block diagram of seconds timer. The chips along the right-hand side are only needed for the optional 100-second timer — omit them for a ten-second timer.

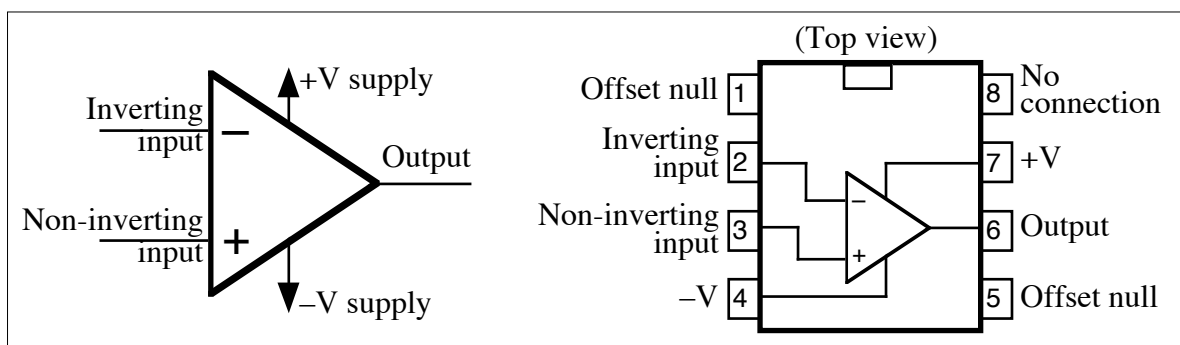
## Part C: Operational amplifiers

### Introduction

The ideal operational amplifier, or op-amp, is a high-gain differential-input amplifier denoted by the triangular symbol shown in figure 8. It has two inputs, called inverting (–) and non-inverting (+), and requires both positive and negative voltage supplies but no explicit earth. The op-amp amplifies the voltage *difference* between the two inputs, so that the output voltage is

$$v_{\text{out}} = A_o(v_+ - v_-)$$

where  $A_o$  is called the open-loop, or open-circuit, gain. For a good op-amp  $A_o$  is typically of the order of a million at low frequencies, so that even a voltage difference of a few microvolts will give an appreciable output. (We use lower-case letters like  $v$  and  $i$  for voltages and currents that can change with time, in contrast to steady DC ones which would be called  $V$  and  $I$ .)



**Figure 8** Operational amplifier, and pinout of 741 and 081

In reality it is impractical to use the amplifier in this way, since most of the time the output will simply saturate to nearly the level of the positive or negative power supply because the input signal is too large. It is almost always necessary to use *negative feedback*, which means that the output is connected back to the inverting input. This reduces the gain to a reasonable level.

In addition, an ideal op-amp has a very high input resistance (ideally infinite, typically at least of the order of megohms) and a low output resistance (ideally zero, typically of the order of a hundred ohms). Typical op-amp limitations are that the gain decreases at high frequencies, and the response to very rapidly-changing inputs is not instantaneous.

When using an op-amp with feedback, we can get quite far in understanding what happens simply by applying two ‘golden rules’:

- The output attempts to do whatever is necessary to make the voltage difference between the inputs zero.
- The inputs draw essentially no current.

In this part we will use two common and inexpensive op-amps, the 741 and the 081. Their pinout is shown in figure 8. Pins 1 and 5, *offset null*, are used to fine-tune the output to zero when the two inputs are equal. We can ignore them.

## Inverting amplifier

This simple configuration is shown in figure 9. It uses an input resistor  $R_i$  and a feedback resistor  $R_f$ . **Build the circuit** step-by-step as follows.

➤ First, place a 741 across the gap on the breadboard, and connect pin 7 to the +12 V and pin 4 to the –12 V power terminals on the back of the test box. Connect the bottom horizontal row of breadboard connectors to the earth terminal of the power supply.

➤ Switch power on and use a multimeter to **check** the supply voltages, connecting one meter lead to earth. Also **measure** the open-circuit output voltage.

➤ Switch power off and connect the two resistors. For  $R_f$  use 100 k $\Omega$  connected between pins 6 and 2, and for  $R_i$  use 10 k $\Omega$  connected between pin 2 and a free breadboard connection.

➤ Connect one of the oscillator outputs to  $R_i$  and the other to earth. Connect channel 1 of the scope to the oscillator output, and connect channel 2 to the op-amp output (pin 6). This will allow a direct comparison of the op-amp’s input and output signals. At least one scope **earth** lead should be connected to the earth of the breadboard. (Note that even though the common earth is not shown in many op-amp diagrams, it will be present since the power supply is connected to it internally.) Using the scope, **set up the oscillator** for sine waves of about 1 kHz with an amplitude of roughly 100 mV.

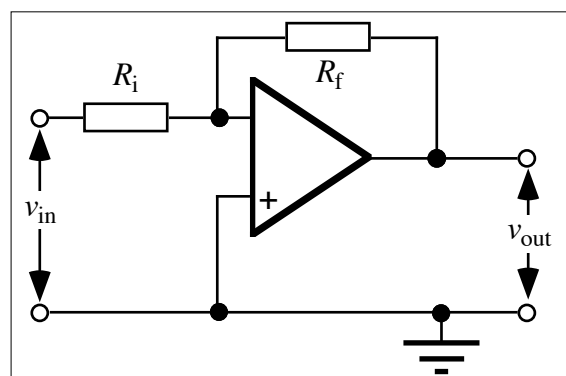
➤ Switch power on and **measure the voltage gain** obtained with your amplifier by comparing the amplitudes of the input and output signals on the scope. **Compare** the voltage gain with the predicted value, which is:

$$\text{Voltage gain } A \equiv \frac{v_{\text{out}}}{v_{\text{in}}} = -\frac{R_f}{R_i}$$

This result, which depends *only on the resistor values and not at all on the op-amp chip*, follows from the ‘golden rules’. The first rule says that the inverting input is at  $\sim 0$  V (it’s called a ‘virtual earth’), so the voltage across  $R_f$  is  $v_{\text{out}}$  and the voltage across  $R_i$  is  $v_{\text{in}}$ . The second rule says no current flows into the op-amp, so the currents in the two resistors must be equal. By Ohm’s Law we have:

$$I = v_{\text{out}}/R_f = -v_{\text{in}}/R_i$$

which can be rearranged to get the equation above. The minus sign indicates the inversion of the signal, which you should see on the scope. This equation is valid over a wide range of resistor values and voltage levels. The *resistor* values must lie below the input resistance ( $\sim 2$  M $\Omega$  for the 741 and  $\sim 10^{12}$   $\Omega$  for the 081) and above the output resistance ( $\sim 75$   $\Omega$ ). The *voltages* will be above the op-amp’s ‘offset’ ( $\sim$ mV) but cannot exceed the voltage supply levels.



**Figure 9** Inverting amplifier



The other restriction on the validity of the simple gain equation is frequency. At DC and low frequencies there is no problem, but at high frequencies the gain decreases.

➤ **Measure the frequency response** by increasing the sine-wave frequency of the oscillator, and measuring and plotting the gain at a number of points. The 741 should not show any decrease below about 10 kHz, so few points are needed below this value; concentrate your measurements where things start to change. **Plot** the results — a *log* scale is needed for the horizontal axis because of the wide frequency range, i.e. an axis labelled 1, 10, 100, 1000 etc.

➤ **Replace** the 741 with an 081, and **repeat** the measurements and gain vs. frequency **plot**. Notice the improved high-frequency response. If the oscillator will not go to sufficiently high frequencies to see the fall-off, try increasing  $R_f$  or decreasing  $R_i$ , since (as discussed below) the fall-off occurs at lower frequencies when the gain is higher.

There is always a trade-off between high-frequency response and gain. This is expressed by the **gain–bandwidth product**. The **bandwidth** is defined as the high-frequency cut-off, i.e. the frequency where the voltage gain has fallen to  $1/\sqrt{2}$  of its maximum (usually mid-frequency) value. ( $1/\sqrt{2}$  in voltage is equivalent to half the output power). For a 741, the gain–bandwidth product is  $\sim 10^6$ , so for example if the gain is 100 then we'd expect a bandwidth of  $\sim 10^4$  Hz.

The **decibel (dB)** is often used in connection with amplifier gain because it is a logarithmic unit, and so can deal with widely-ranging values. Decibels are defined in terms of *power* ratios; the relative power gain in decibels is

$$\text{dB} = 10 \log_{10} \frac{P_{\text{out}}}{P_{\text{in}}}$$

In other words, a power ratio of 10 is +10 dB. An increase in power by a factor of 2 is  $10 \log_{10}(2) = 3.01$  dB, which is normally called 3 dB, while a decrease by a factor 2 is –3 dB. If we wish to use voltage rather than power, then

$$P = \frac{V^2}{R} \quad \text{so dB} = 10 \log_{10} \frac{P_{\text{out}}}{P_{\text{in}}} = 10 \log_{10} \frac{V_{\text{out}}^2/R}{V_{\text{in}}^2/R} = 10 \log_{10} \frac{V_{\text{out}}^2}{V_{\text{in}}^2} = 20 \log_{10} \frac{V_{\text{out}}}{V_{\text{in}}}$$

Thus a *voltage* ratio of 2 means  $20 \log_{10}(2) = 6.02$  dB, approximately +6 dB, while a ratio of 0.5 means –6 dB. A ratio of 10 is +20 dB, while 0.1 is –20 dB.

Hence, a gain decrease of –3 dB represents a decrease in power of a factor 2, and a decrease in voltage of a factor  $1/\sqrt{2}$ .

If we plot an amplifier's gain in decibels vs. logarithm of the frequency, we have a log–log plot on which the drop-off in gain as a function of frequency should be roughly a straight line. The usual units for the slope of this line are dB per decade, i.e. the drop in gain (in decibels) for each factor of 10 increase in frequency.

➤ **Re-plot** the frequency responses of the 741 and 081 in this way so that you can **estimate** the **slopes** at which they fall off. Estimate the **bandwidth** of each chip, and hence find their **gain–bandwidth products**.

## Non-inverting amplifier

This circuit is shown in figure 10. The input signal now goes to the non-inverting (+) input, with the feedback still connected to the inverting (–) input.

➤ **Wire up this circuit**, again using 100 k $\Omega$  for  $R_f$  and 10 k $\Omega$  for  $R_i$ .

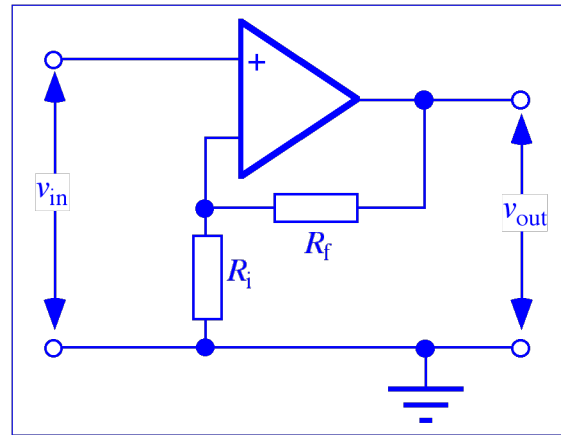
➤ **Measure** the **gain**, and **compare** with the predicted voltage gain, which is:

$$A = \frac{v_{out}}{v_{in}} = 1 + \frac{R_f}{R_i}$$

This can again be deduced from the ‘golden rules’. Here,  $v_+ = v_- = v_{in}$ , and  $v_-$  comes from what is effectively a voltage divider:

$$v_- = v_{out} \frac{R_i}{R_i + R_f}$$

Simple substitution gives the gain of the amplifier. As with the inverting amplifier, we see that if the op-amp behaves ideally then the gain depends *only* on the *external resistor values* and not on the properties of the op-amp chip itself. The positive value means that there is no inversion, and the absolute value of the gain is higher than that of the inverting amplifier by 1.



**Figure 10** Non-inverting amplifier

## Part D: A simple digital-to-analogue converter

### Introduction

The world around us is full of analogue signals. The availability of sophisticated but cheap digital logic and microprocessors means that it is also essential to have converters, both to digitise analogue signals for processing, and to turn digital results back into analogue. In this part you will first use an op-amp to sum input signals, and then study an electronically controlled switch. Finally, these will be put together to build a simple 4-bit digital-to-analogue converter (DAC) in order to understand some of the basic principles.

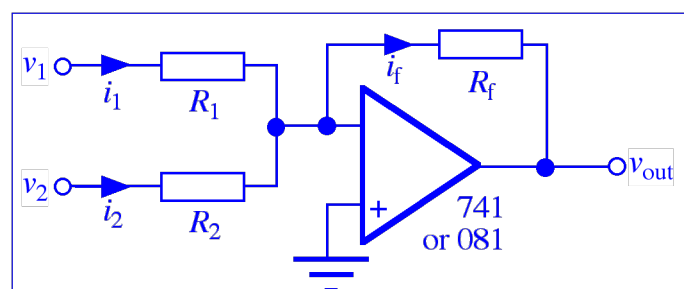
The DAC constructed in this exercise works by summing currents, each of whose magnitude is proportional to the binary bit that it represents. The binary number is formed by using an electronically-controlled switch for each current, so that each binary bit can be turned on or off. To show this circuit in action, the binary number is taken from a counter so that it varies in a simple and easily understood manner.

However, this DAC has very limited performance. For real applications demanding higher precision and speed it is far better to buy ready-made integrated converter chips than to try to design your own circuits. Very high-performance conversion is not easy, and prices go up rapidly as the number of bits and/or the speed increase. For example, the 16-bit DACs used to play CDs were initially state-of-the-art but have now become both more sophisticated *and* very cheap.

### Summing amplifier

This circuit is a variation on the inverting amplifier of part B, and is shown in figure 11. Since no current flows into the op-amp input (golden rule), the currents in the two input resistors  $R_1$  and  $R_2$  are summed in the feedback loop resistor  $R_f$ :  $i_1 + i_2 = i_f$  or in terms of voltage:

$$\frac{v_1}{R_1} + \frac{v_2}{R_2} = -\frac{v_{out}}{R_f}$$



If  $R_1 = R_2 = R_f$ , then  $v_{out} = -(v_1 + v_2)$ . **Figure 11** Summing amplifier

This means that, apart from the sign, the output depends on the *sum* of the inputs.

➤ **Build the summing circuit**, using  $\pm 12$  V supplies for the op-amp. The choice of resistance is not critical — 10 k $\Omega$  is suitable for all three resistors.

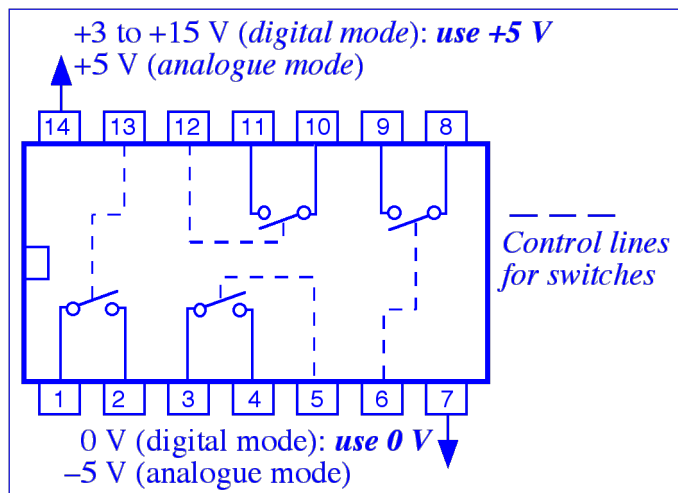
➤ **Test** the circuit by putting a sine wave signal into one of the inputs, and +5 V DC into the other input. Observe the results on the scope, noting the DC level as well as the sine wave.

**DO NOT DISMANTLE YOUR CIRCUIT — IT IS NEEDED LATER**

## 4066 CMOS switch

An ideal switch has no resistance when closed and an infinite resistance when open. Mechanical switches come close, while electronic ones are further from ideal. However, the electronic versions allow much higher switching speeds. We will use a 14-pin CMOS chip which has four single-pole switches and four control inputs. The switches have a resistance of about 90  $\Omega$  when closed, and can switch 5 V signals at speeds up to 5 MHz. The pinout is given in figure 12. Note that there are two modes, *analogue* and *digital*. We will use **digital** mode. The rules for use are:

- A switch is off when its control voltage equals the pin 7 voltage (i.e. 0 V).
- A switch is on when its control voltage equals the pin 14 voltage (i.e. +5 V).
- Signals passing through the switches must never go below the pin 7 voltage, nor above the pin 14 voltage.
- The switches are bi-directional, and there is no difference between input and output terminals. In other words, the switches behave just like a 90  $\Omega$  resistor.



**Figure 12** Pinout of 4066 switch

➤ **Check** how the 4066 works by using a multimeter to measure the resistance of one of the switches (e.g. pins 1 and 2) when it is open and closed. **Control** the switch by using one of the breadboard's switches to set the control voltage (e.g. on pin 13). **Record** the resistance with control settings of 0 V and +5 V.

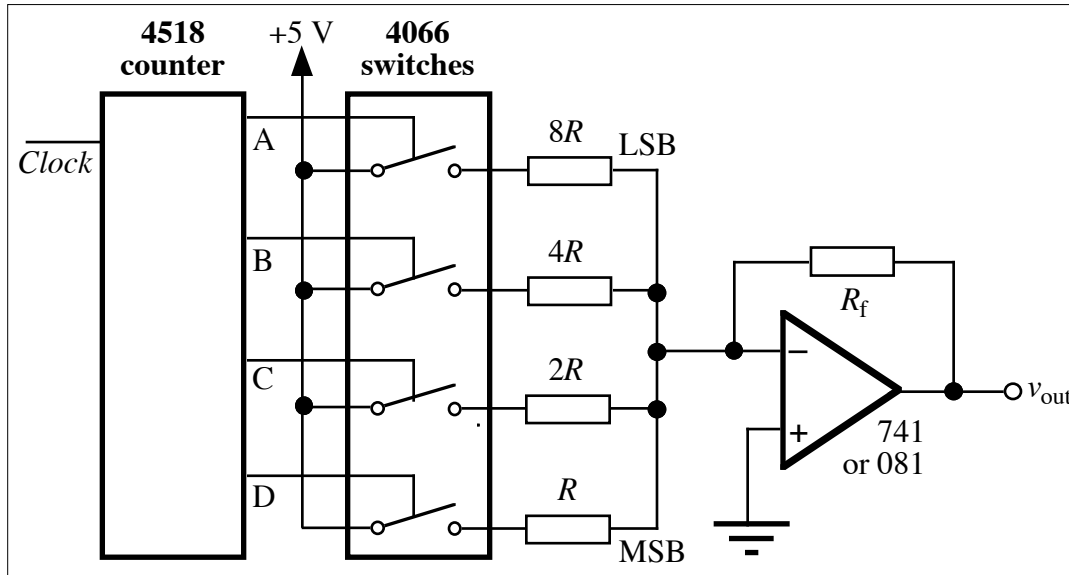
➤ **Use** the 4066 by putting two of its switches in series with  $R_1$  and  $R_2$  to control the inputs of the op-amp summing amplifier previously built. **Connect** +5 V to one side of one of the switches (e.g. pin 1), and  $R_1$  to the other side of the same switch (e.g. pin 2). **Connect** the TTL output of the signal generator to one side of another switch (e.g. pin 3), and  $R_2$  to the other side of the same switch (e.g. pin 4). **Control** the two switches by using two of the breadboard switches (connected to e.g. pins 13 and 5, respectively). Use the scope to see what happens: make **drawings** of the waveforms with one switch on, the other switch on, and both switches on. (Pay attention to DC levels as well as waveform.) Is this what you would expect?

## Digital-to-analogue converter

The aim now is to build a 4-bit DAC, using the circuit shown in figure 13. An op-amp is used to sum four inputs, each representing one of the binary bits. The bits will be generated by +5 V signals, with the resulting currents weighted by using resistors in the ratio  $8R : 4R : 2R : R$ . Each of the bits will be turned off or on by one of the CMOS switches of a 4066. This means that the output voltage of the op-amp will be:

$$v_{\text{out}} = -v_R \left( \frac{1 \text{ or } 0}{1} + \frac{1 \text{ or } 0}{2} + \frac{1 \text{ or } 0}{4} + \frac{1 \text{ or } 0}{8} \right)$$

Note that the output is inverted. This could be corrected by adding an inverting amplifier with unity gain after the summing amplifier. The most-significant bit, or MSB (i.e. the left-hand one in binary) comes from the smallest resistor, and the least-significant bit (i.e. the right-hand one in binary) from the largest resistor.



**Figure 13** Digital-to-analogue converter

In order to test the DAC, the circuit includes a 4518 counter, to count TTL pulses from the TTL output of the signal generator. This gives binary outputs that ramp up from zero to maximum value and then reset.

Suitable resistor values are 3.3 k $\Omega$ , 6.8 k $\Omega$ , 15 k $\Omega$  and 27 k $\Omega$ , with a feedback resistor of 3.3 k $\Omega$ . Note that the switches also add 90  $\Omega$  each, and that due to the imprecise resistor values ( $\pm 5\%$ ) the binary ratios will only be approximate. It is possible to do much better by using higher-precision resistors.

➤ **Build this circuit.** Note that the op-amp uses  $\pm 12$  V supplies, while all the other chips use +5 V.

➤ **Examine** and **sketch** the waveforms at the input and output of the circuit, using the scope, with the TTL input to the counter going at a rate such that everything can be seen clearly. If all is well, you should see a pattern like a staircase. Do you understand the number of steps?

➤ Finally, **measure** the output voltage corresponding to each of the 10 possible binary (BCD) numbers. This is easier if you replace the TTL input to the counter by one of the breadboard switches. This allows you to clock up the counter one count at a time. Values can be read off the scope, with care, or you can use a DMM. Make a **graph** of the results, with counter values on the x-axis and the measured voltage on the y-axis. Is the plot **linear**?

Practical DACs need more than four bits, and using this ‘binary ladder’ method would demand resistors with a huge range of values. (An 8-bit DAC would require the largest resistor value to be 256 times the smallest one.) Therefore, a slightly more complicated method called the ‘R–2R ladder’ is used; as the name implies, only two resistor values are needed.