AN ARGUS SPECIALIST PUBLICATION Vol.4 No.4

ELECTRONICS DIGEST

£2.25

MICROCOMPUTER DESIGN

Electronic circuits for the beginner to the expert

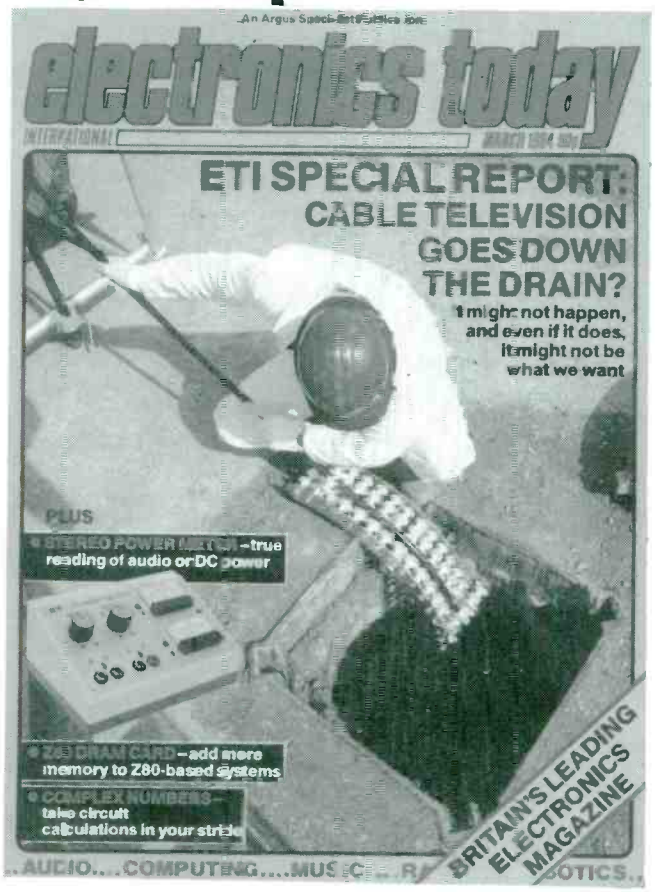
Circuitry in computers—understand your own system

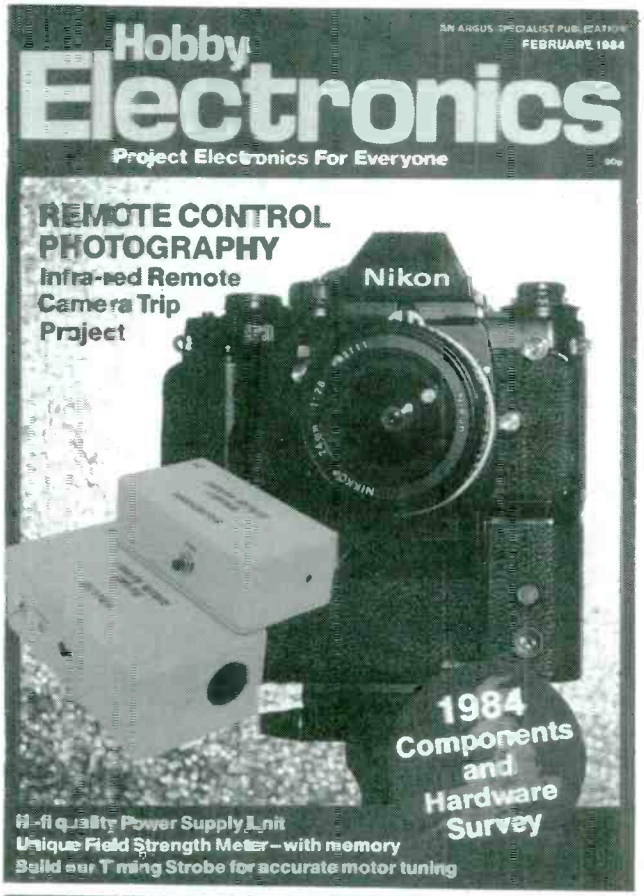
Teletext explained

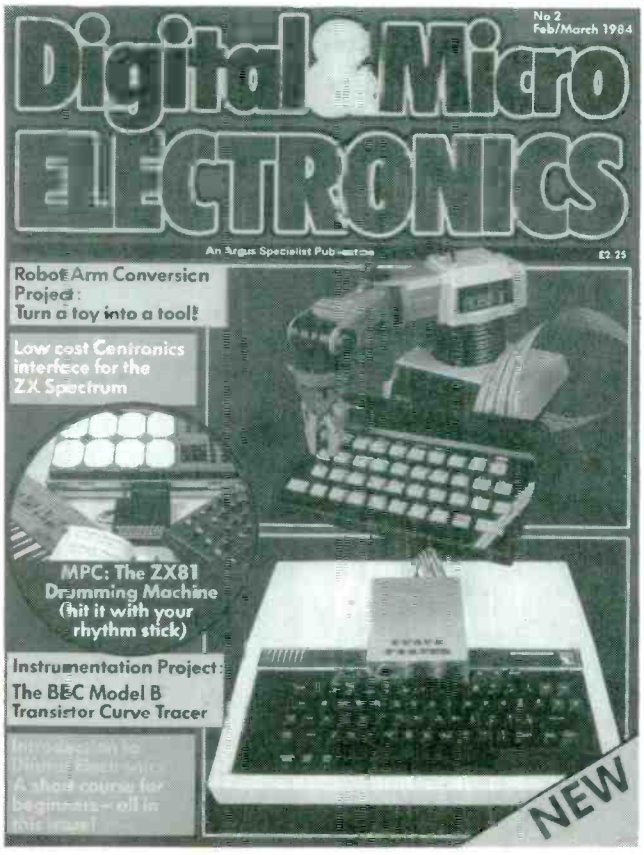
Microphones, music and videos uncovered

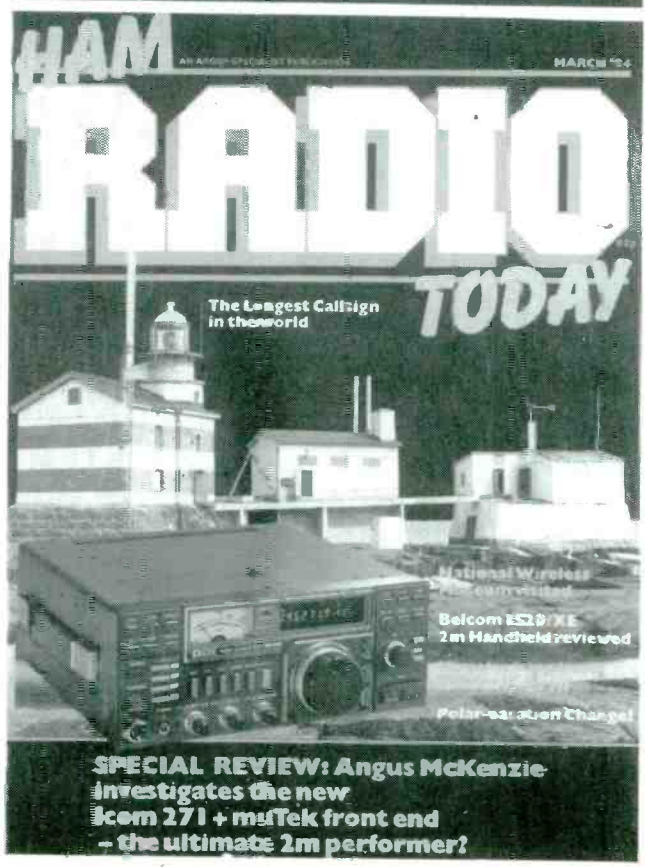
From the publishers of electronics today

Argus Specialist Publications . . . . .









. . . . . for the latest in electronics, computing and radio

ELECTRONICS DIGEST

Volume 4 No. 4

Editor:
Dave Bradshaw
Special Publications Editor:
Wendy J. Palmer
Managing Editor:
Ron Harris
Advertisement Manger:
Paul Stanyer
Chief Executive:
T. J. Connell

ORIGINATED BY: Tabmag, Northampton.
PRINTED BY: Garden City Press, Letchworth.

Subscription rates upon application to Electronics Digest, Subscriptions Dept., PO Box 35, Wolsey House, Wolsey Road, Hemel Hempstead, Herts HP2 4SS.

PUBLISHED BY: Argus Specialist Publications Ltd, 1 Golden Square, London W1R 3AB.

DISTRIBUTED BY: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS (British Isles).

INTRODUCTION

In order to understand computers, you must know some electronics; so the first part of this volume, Configurations, is a summary of some of the most frequently met circuits in electronics. This section assumes that you have already met transistors — if this is not the case, then we suggest turning first to Electronic Digest Volume 4 No 1 — Introduction to Circuit Design.

The Interlude deals with a few topics that don't fit into either of the other two sections, yet that will still be of use to you.

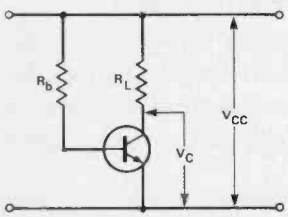
The final section is the description of computer systems itself. While we cannot hope to teach you all there is to know about the workings of computers in such a slim volume, we hope that what is presented here will equip you to understand the more advanced books and articles that you will come across.

CONTENTS

CONFIGURATIONS 1

Not our answer to James Burke but a series aimed at the designer. Ian Sinclair will be looking at some of the basic workaday circuits that often get eclipsed by the more glamorous ICs, showing you how and why they work and how and why to use them. We kick off with common-emitter transistor bias.

Configurations is a series which aims to provide you with fundamental circuit design data for a number of the most commonly used circuit blocks. A very large amount of circuit work concerns these standard arrangements, so that you will be able to build up a complete designer's handbook of circuits and their design data. We're starting with the most fundamental of all — biasing and calculating gain and bandwidth of the single-stage common-emitter amplifier, using a silicon transistor with a resistive load.



(a) TO FIND A VALUE FOR R_b, GIVEN A DESIRED VALUE OF V_C:

$$R_b = \frac{R_L h\ R_L.h_{fe}.(V_{CC} - 0.6)}{V_{CC} - V_C}$$

R_L AND R_b IN KILOHMS

(b) TO FIND WHAT VALUE OF V_C WILL BE CAUSED BY A GIVEN BIAS RESISTOR:

$$V_C = \frac{V_{CC}.R_b - R_L.h_{fe}.(V_{CC} - 0.6)}{R_b}$$

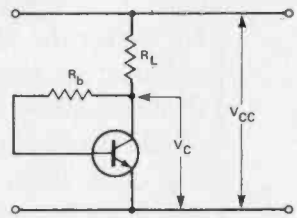
R_L AND R_b IN KILOHMS

Fig. 1 Simple single-resistor bias circuit. The value of resistance depends critically on the value of h_{fe} for the transistor.

The simplest bias circuit, of course, is that of Fig. 1, using a single bias resistor connected between the base and the supply positive. We're not going to spend much time on this one, because it's not a very good bias method from any point of view. The reason is that the resistor value has to be spot on for this method of work, and you have to know the current gain (h_{fe}) value for that particular transistor (not just the average for the type) to a fair degree of accuracy. If you need to use that method and have a box of 1% tolerance resistors handy, then the design data is illustrated in Fog. 1. One of the few things that can be said for the circuit is that a high input resistance is attainable, but more on that subject later.

A Favourable Bias

A much more practical bias method is illustrated in Fig. 2. This makes use of DC shunt feedback between the collector and the base of the transistor, and is less likely to be upset by the changes that occur in the characteristics of the transistor as it heats up. You still have to know the h_{fe} value for the transistor, but the collector voltage won't be so far out if you just use an average value for the type and it happens that the one you're using is at one end of the range of values. The formula, like the previous one, assumes that the base-to-emitter voltage when the transistor is conducting will be 0V6 and since this is the



(a) TO FIND A VALUE FOR R_b, GIVEN A DESIRED VALUE OF V_C:

$$R_b = \frac{R_L.h_{fe}.(V_C - 0.6)}{V_{CC} - V_C}$$

R_L AND R_b IN KILOHMS

EXAMPLE: IF R_L = 2k2, h_{fe} = 100, V_{CC} = 9V, V_C = 3V, THEN

$$R_b = \frac{2.2 \times 100 \times 2.4}{6} = 88k$$

(b) TO FIND A VALUE FOR V_C, GIVEN R_b:

$$V_C = \frac{R_b.V_{CC} + 0.6R_L.h_{fe}}{R_b + R_L.h_{fe}}$$

Fig. 2 Shunt-feedback bias. The value of collector voltage is less dependent on the h_{fe} value. Note the units, with all resistances in kilohms.

quantity that changes most as a silicon transistor heats up, it's worth while taking a look at how this bias method is affected.

Figure 3 shows two calculations of collector voltage, both assuming a supply voltage (V_{cc}) of + 9 V, load resistor of 2k2, h_{fe} value of 100, and bias resistor R_b of 88k. However, one uses 0V6 as the V_{be} figure and the other uses

ASSUME IN BOTH CASES THAT R_b = 88k, h_{fe} = 100, V_{CC} = 9V, R_L = 2k2.

WHEN V_{be} = 0V6, $V_C = \frac{88 \times 9 + 0.6 \times 2.2 \times 100}{88 + 2.2 \times 100} = 3V$

WHEN V_{be} = 0V5, $V_C = \frac{88 \times 9 + 0.5 \times 2.2 \times 100}{88 + 2.2 \times 100} = 2V93$

DIFFERENCE IN V_C = 70 mV

Fig. 3 Effect of temperature. The V_{be} (assumed 0V6 for a silicon transistor) decreases as the temperature rises. The calculations show that the collector voltage value is hardly affected.

0V5. The difference in the collector voltage is negligible, which points to this method of bias as being a very useful one when you are worried about the effect of temperature changes on the performance of the transistor.

The circuit uses feedback, of course, and unless something is done to remove the feedback of AC, the gain of the stage and its input resistance will be reduced. The

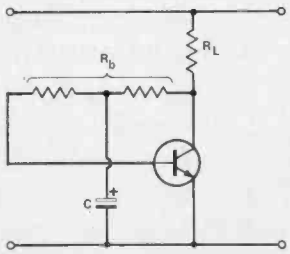
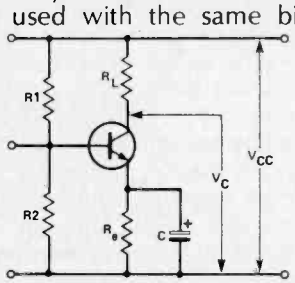


Fig. 4 Removing AC feedback by splitting the bias resistor into two sections and decoupling.

reduction in gain isn't serious for most circuits, but the input resistance problem can be more serious — it's detailed later in this article. Both can be tackled if AC is removed from the feedback path by splitting the bias resistor into two parts and decoupling it, as shown in Fig. 4.

An Arrangement With Potential

The most-extensively used of all bias methods is our old friend in Fig. 5, which uses a potential divider to provide a constant voltage (we hope) at the base of the transistor, and DC feedback (series feedback this time) through the emitter resistor to stabilise the bias. The notable thing about the formula is that h_{fe} doesn't appear anywhere in it, so that the bias should not be affected by the value of h_{fe}. This means that the circuit is very tolerant of wide ranges of h_{fe} values, providing the base current of the transistor is not so large that it disturbs the base bias voltage set by the potential divider. As a rule of thumb, if the current flowing through R1 and R2 (equal to $V_{cc}/(R1 + R2)$) is something like 100 times the base current of the transistor, then the circuit will work exactly as per design, and any transistor whose base current is within limits can be used with the same bias components. If the base



(a) TO FIND R_e FOR A DESIRED V_C:

$$R_e = \frac{R_L \cdot \left(\frac{V_{CC} \cdot R2}{R1 + R2}\right) - 0.6}{V_{CC} - V_C}$$

(b) TO FIND V_C FOR A GIVEN R_e:

$$V_C = V_{CC} - \frac{R_L \cdot \left(\frac{V_{CC} \cdot R2}{R1 + R2}\right) - 0.6}{R_e}$$

Fig. 5 The potential-divider bias method. This is particularly useful for mass-produced circuits, because bias does not depend on h_{fe} values.

current of the transistor is far from negligible then complication arise, and it's easier just to use lower values of R1 and R2 — but for the effect on input resistance, see later. For voltage amplifier stages where the collector current is only about 1 mA, values like 6k8 and 1k5 on a 9 V supply will suit the circuit very nicely.

One disadvantage of the circuit is that there's an emitter DC voltage so that the available voltage swing at the collector is correspondingly reduced. The other point, which is important where space is limited, is that decoupling of the emitter resistor is essential. Without decoupling the gain is low; it's given by R_L/R_e and will be around two to six times for the kind of values you are likely to end up with in a practical circuit. The decoupling capacitor operates at low voltage, so that a 3 V or 6 V type is normally adequate, but its value has to be large to avoid a noticeable loss of gain at low frequencies. It certainly isn't enough to have the reactance of the capacitor equal to the resistance value of R_e at the lowest frequency for which the amplifier is intended to be used, because if you make this assumption for each coupling and decoupling time constant, you'll end up with practically no gain at that frequency. Aim for a capacitor reactance of about one fifth of the emitter resistance value at the lowest frequency you intend to use and the results will be more acceptable. With C in microfarads and R_e in kilohms, this means a value given by the equation $C = 5000 / 2\pi f.R_e$ and for a 390 ohm emitter resistor, this indicates a capacitor value of around 50uF for a 40 Hz breakpoint. Even at 3 V working, this is going to be a component that will be larger than the resistors or the transistor.

The input resistance of an amplifying stage is, as the name suggests, the ratio of input voltage to input current

IF h_{ie} = INPUT RESISTANCE IN KILOHMS, I_C = STEADY COLLECTOR CURRENT IN MILLIAMPS, AND h_{fe} = VALUE OF CURRENT GAIN, THEN

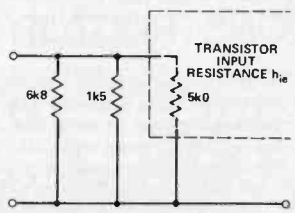
$$h_{ie} = \frac{h_{fe}}{40.I_C}$$

EXAMPLE: IF h_{fe} = 400 AND I_C = 1mA, THEN $h_{ie} = \frac{400}{40 \times 1} = 10k$

Fig. 6 Transistor input resistance, h_{ie}. Note this is for the transistor only, and assumes zero-signal conditions.

for an AC signal at a frequency in the middle of the bandwidth. The input resistance of a transistor is not constant, but if we take the value which it has at the setting of the bias current, with no signal, then this is a reasonable average to take for small signal inputs — small meaning millivolts. The value is calculated as shown in Fig. 6, and it depends on the h_{fe} value and the bias current. In general, using transistors with high h_{fe} values operated at low collector currents will give the highest input resistance values for the transistor, but you can usually assume values in the region of 1k0 to 10k.

These are just input resistance figures for the transistor itself, however, and the total input resistance will be affected by the bias components. When we use the potential divider bias circuit, for example, both R1 and R2 (in Fig. 5) are connected between the base and a line which is at 0 V (AC). How so, you ask? Well, as far as AC voltage is concerned, the supply positive line is as much of an earth as the genuine earth line, since they are connected to each other by a whopping great electrolytic in the power supply circuit. Hence all of these bias resistors are in parallel across the base-to-earth path, considerably lowering the input resistance (Fig. 7). If you think that the shunt feedback circuit of Fig. 2 is better then think again, because the collector end of the bias resistor is connected to a voltage which is in antiphase to (and of much greater amplitude than) the base voltage, so it behaves as if its value were R_b/G connected to earth. G is the voltage gain of the stage, so that if R_b = 88k and G = 50, then the bias resistor is effectively 1k76 in parallel with the input resistance of the transistor itself.



$1/R_{IN}$ = 1/5 + 1/1.5 + 1/6.8 (ALL RESISTANCES IN KILOHMS), SO R_{IN} = 0.986k = 986R WHERE R_{IN} IS TOTAL INPUT RESISTANCE

Fig. 7 The effect of bias components and h_{ie} on total input resistance.

Gainful Employment

The output resistance of a single transistor amplifier consists of the output resistance of the transistor itself, usually around 30k, in parallel with the load resistor. Since load resistor values are usually of the order of 1k0 to 10k, this in practice means that we can use the load resistor value as the value for output resistance when we are dealing with resistor-capacitor coupled stages.

The crunch comes when we want to find what the gain of an amplifying stage will be. For a silicon transistor which has enough h_{fe} to class it as being in the land of the living, the maximum voltage gain is given by 40 × V_{RL}, where V_{RL} is the voltage across the load resistor when no signal is applied — the bias voltage in other words across R_L. For example, if you are using a transistor with 4V5 across the load resistor, then the maximum gain is 40 × 4.5, which is 180 times, and that's the value which can be measured if you use a low impedance signal generator, a

very small signal amplitude, and a high-impedance oscilloscope to measure the output.

Practical circuits, however, use higher-resistance devices as signal sources and low resistance devices as signal loads, so that the gain when we take into account the potential-dividing effect of all these loss-makers is a lot less. For example, if we imagine our transistor stage to be fed with a signal from another stage with an output resistance of 2k2 and feeding into a stage with input resistance of 1k0 (and with these same values itself) then its gain (Fig. 8) will be a miserable 17.5 times. It's not the gain of the transistor which has fallen, notice; it's the attenuation caused by the potential dividers which is dissipating the signal. The moral is that input and output resistances are important when you are aiming for maximum gain, and that everything you can do to raise

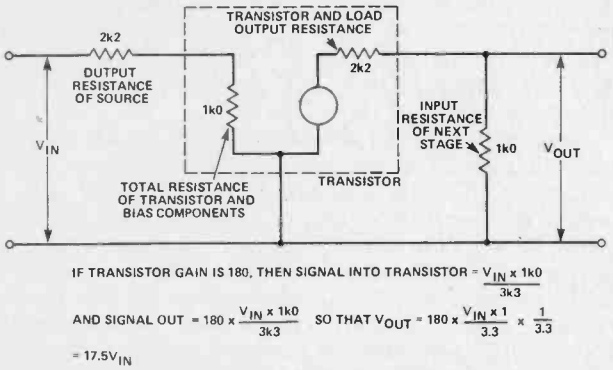


Fig. 8 The effect of input and output resistances in forming potential divider circuits with the internal resistances of source and load.

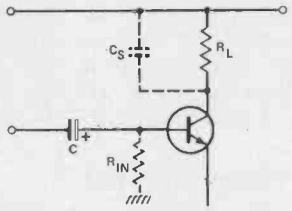


Fig. 9 The time constants which affect bandwidth for a single stage. Not the different units for the two equations.

input resistance and reduce output resistance can be useful.

The bandwidth of an amplifier stage is defined as the range of frequencies over which the gain does not drop 3 dB below its mid-band value. For a simple amplifier stage, the limits of bandwidth are set by the time constants of the coupling and emitter-decoupling capacitors at the low frequencies, and by the effects of stray capacitance at the high frequencies — Fig. 9 shows the details. Most modern transistors have good high-frequency gain and since stray capacitance can be made small with modern circuit layouts, frequencies into the many megahertz range can be expected. This is much more than is necessary (or desirable) in many cases, and it's a wise precaution to trim the bandwidth for the purpose you need. This can be done by introducing a time constant into the feedback network of a simple filter.

CONFIGURATIONS 2

Our topic this time is twins and triplets. Don't worry, you haven't picked up a copy of 'Mother and Baby' by mistake — it's just Ian Sinclair explaining transistor amplifier feedback loops.

One of the major problems about the straightforward transistor amplifier is the calculation of gain. As we saw last time, the gain of the transistor itself is easily calculated, but if we are to allow for the effect of input and output resistance, we have to know a lot about the particular transistor we are using. The task becomes even more difficult when the amplifier operates at high frequencies, so that we have to deal with input and output impedances rather than with pure resistances.

Negative feedback is the main method that we use to get around this problem. To start with, imagine an amplifier whose voltage gain is G — this is sometimes called the 'open loop gain', the gain with no feedback loop connected (Fig. 1). The amplifier has an input connection and a feedback connection, and 1/n of the output signal is fed back to the input. Since the output signal is V_{OUT}, the amount fed back is V_{OUT}/n, and so the actual signal available to amplify is $V_{IN} - V_{OUT}/n$, the input voltage minus the feedback signal. This is the signal which appears at the inputs of the amplifier to be amplified, so that with gain G, the output from this must be $V_{OUT} = G(V_{IN} - V_{OUT}/n)$.

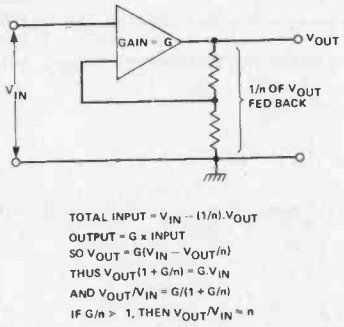


Fig. 1 Principle of negative feedback.

An Open And Closed Case

When we collect terms in this equation and rearrange it to get V_{OUT}/V_{IN} (which is the gain with the feedback connected), we get this figure as G/(1 + G/n). This is the gain-with-feedback or closed-loop gain, and this is the equation that should be used in calculations if G is small or n is large. If the ratio G/n is large compared with 1, however, we can ignore the 1 and rewrite the equation as G/(G/n), which is simply n.

This important result shows that if the open-loop gain is large enough the gain when feedback is used is independent of anything except the potential divider that produces the 1/n of the signal which is fed back. This is the result that is often quoted in textbooks, but with a slightly different proof. One important point is that feedback is useful only if there is gain — if an amplifier has zero gain under certain circumstances, no amount of feedback can produce gain.

Looking On The Negative Side

Given that negative feedback can be useful, let's look at some circuits that make use of it. Since gain is important, there's not too much point in spending time looking at single-transistor stages, though both of the circuits in Fig. 2 are important and are widely used. The equations are approximations which apply to silicon transistors; no bias components have been shown.

Negative feedback really comes into its own when two or more transistor stages are used, and if the feedback path is a DC one then both gain and bias will be controlled by the feedback. This is seldom completely convenient, so that the basic circuits have to be modified to allow for separating bias and signal feedback paths. One of the most-widely-used of these circuits is the feedback pair circuit of Fig. 3. In this circuit, the transistors are direct-coupled, and the negative feedback path is from the emitter of Q2 to the base of Q1. Because of the bypass capacitor C2, there is no signal voltage across R6, and the signal voltage that is fed back is the voltage across R5 only. This value, along with the value of feedback resistor R7, can be chosen to produce the amount of gain that is

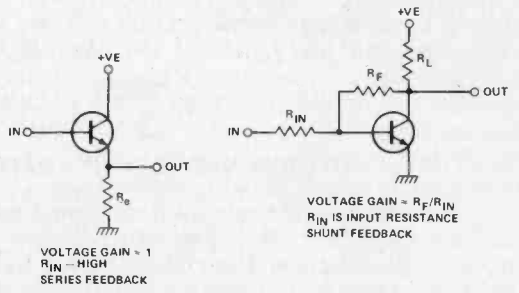


Fig. 2 Feedback over a single stage gives rise to these well-known circuits.

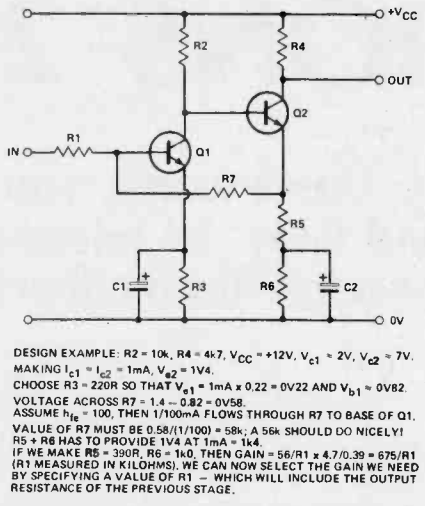


Fig. 3 A two-transistor negative feedback circuit using emitter-to-base feedback.

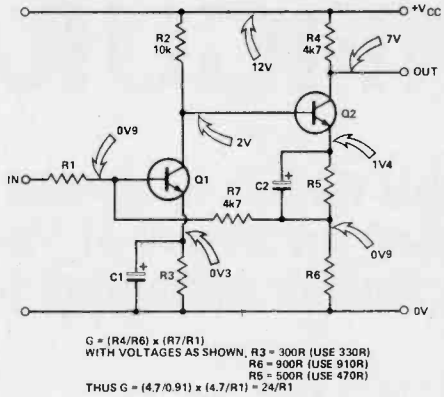


Fig. 4 A modified feedback circuit which is easier to design — sample voltages are shown.

desired; providing of course, that the value is small compared to the open-loop gain which will be the product of the open-loop gain figure for each transistor multiplied by the loss for the potential divider effects of input and output resistances.

Getting Good Value(s)

Design of the circuit should start with the load and bias components. The values of the two load resistors R2 and R4 are fixed first; these should be fixed at values which will permit the transistor to operate at suitable currents. Values in the range 1k0 to 12k are usually suitable, though R2 can be in the range 15k to 47k if Q1 is operated at low current to reduce noise signals. If you prefer it, write down the desired collector voltages (Q2 must have a higher collector voltage than Q1) and choose current values, then use Ohm's law to calculate the load resistors.

Once the load resistor values, the currents and the voltage levels have been fixed, the bias components can be calculated. The voltage at the emitter of Q2 will be 0V6 less than the base voltage of Q2, and the voltage at the base of Q1 will be 0V6 higher than the emitter voltage of Q1. Subtracting these gives the voltage across R7, and we then have to pick a value which will allow the correct amount of base current to flow. Once this has been done, we can split the total resistance of R5 and R6 (which has to be calculated to produce the correct bias voltage) so as to allow for enough gain for Q2(R4/R6). We can then choose the total figure of gain by selecting R1, which is the sum of a resistor and the output resistance of the circuit that is providing the signal.

Though this circuit works well, it is never easy to get the bias spot on by calculation unless you know what base current Q2 requires, and the total gain is rather dependent on the source resistance value. The bias problem can be eased by a modification of the circuit shown in Fig. 4, where R7 has a fairly low value, around 4k7, so that the base voltage at Q1 is practically equal to the voltage across R6. This allows the h_{fe} value of the transistors to be neglected, but the penalty to be paid is that the gain is lower unless R1 is also small.

Further Feedback

Figure 5 is another direct-coupled two-transistor circuit which uses negative feedback, but this time from the collector of Q2 to the emitter of Q1. As shown the circuit is impractical (because there are no bias components) and a very useful compromise is to combine this circuit for AC

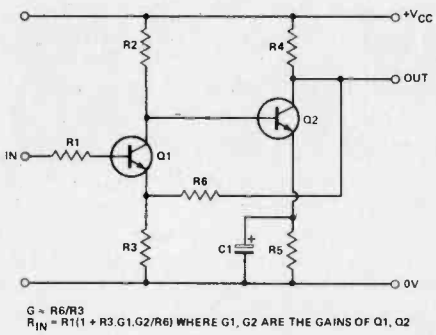


Fig. 5 An alternative two-transistor circuit, using collector-to-emitter feedback.

feedback with the previous circuit for DC feedback (Fig. 6). The gain of this example is approximately (R7 + R3)/R3, and the bias is set by the DC network R3, R6, R5. The design procedure is to decide on operating currents and voltage levels as before, calculate load resistor values, and then assuming that V_{C2} will be 0V6 less than V_{C1}, find a value for R5. Make R6 a size that will suit its use as an input load (around 4k7) and calculate a value for R3 which will make the emitter voltage of Q1 0V6 less than the emitter voltage of Q2. You can then pick a value for R7 which will give you the gain you want, but remember that R1 (the source resistance of the signal supply stage) will form a potential divider with R6, so that the overall gain will be lower than the calculated gain from the stage.

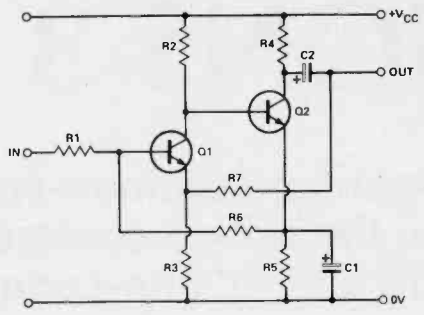


Fig. 6 Combining feedback systems, using one for DC and the other for AC.

After Twins, Triplets!

On the basis that negative feedback works best when there is plenty of gain, three-transistor amplifiers should perform better than twins. There's a catch, though, concerned with time constants. Each time constant in a signal path will cause phase shift at one of the extremes of frequency, high or low. Coupling capacitors will cause phase shift at low frequencies, as will emitter decoupling capacitors, and stray capacitances will cause phase shifts at high frequencies — usually beyond the audio limits. This phase shift in each time constant will cause a total phase shift over the whole amplifier circuit that can be quite large — and when it approaches 180° your negative feedback is in danger of turning into the opposite type — positive feedback. This converts your amplifier into an oscillator if the gain in the amplifying part is greater than the losses in the time constants at the frequency where the 180° phase shift occurs. Calculations on this are far from simple unless you use a computer (and if you're waiting for the Sinclair QL, then it's quicker to work it out on an abacus anyway). Two useful points are (a) to use direct coupling as far as possible, and (b) if you must use time constants, don't use a set of identical time constants.

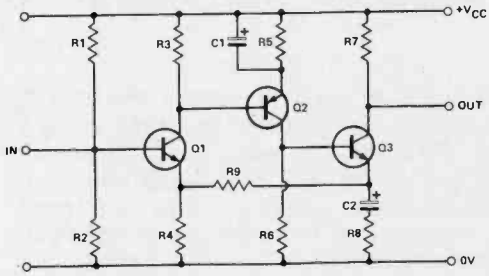
A Compromising Situation

A circuit which can be a useful compromise is shown in Fig. 7. It uses direct coupling, made possible by having a PNP transistor as the intermediate stage, and the negative feedback is used to stabilise the bias as well as the gain. Looking at bias first, the base voltage of Q1 is set by R1 and R2, and the feedback of bias is achieved by passing all the DC current from Q2 through the emitter resistor of Q1. This would cause the AC gain to be very low, but by using R8, which is coupled to the emitter of Q3 through C2, the amount of AC signal at the emitter of Q3 can be reduced, allowing the amount of AC signal fed back to be controlled. The gain of the stage is given by (R7/R8) × (R9/R4).

Design starts as usual with biasing. Assuming, as is reasonable, that the first transistor will be run at a lower current than the others to reduce noise, we can choose current levels of 100 uA for Q1 and 1 mA each for the other two. We can then choose voltage levels, making the voltage of the collector of Q1 reasonably high, the collector voltage of Q2 reasonably low, and the collector voltage of Q3 about midway between V_{CC} and the collector voltage of Q2. These figures and our selected currents allow us to calculate values for load resistors and for R5.

With this done, we can start to look at R9 and R4. We have to allocate the total voltage at the emitter of Q3 between these resistors, and a ratio of something like 5:1 is reasonable here — the voltage across R9 being about five times as much as the voltage across R4. I've used 6 in the example. Since R9 carries the allocated current of Q3, its value can be calculated right away, and the value of R4, which carries the current of Q1 as well as that of Q3 can be calculated.

The only remaining factor is R8, whose value then decides the total gain. Calculate the gain contribution of R9/R4, and divide this into the total gain required. This latter figure will then be the ratio R7/R8, and since you have already decided on a value for R7, a value of R8 can be calculated, followed by R1 and R2. C1 and C2 should have values which allow the gain to be maintained at the lowest frequencies at which the amplifier will be used, and the time constants R5-C1 and R8-C2 should be quite different.



EXAMPLE: V_{CC} = 9V, V_{c1} = 7V, V_{c2} = 3V, V_{c3} = 7V.
MAKE I_{c1} = 0.1mA, I_{c2} = I_{c3} = 1mA.
THEN R3 = 5k0, R5 = 1k4, R6 = 3k0, R7 = 5k0 (USE NEAREST PREFERRED VALUES)
R9 CARRIES 1mA, SO R9 + R4 HAS A 2V4 DROP.
ALLOW 0V4 AT THE EMITTER OF Q1 AND 2V ACROSS R9, SO THAT R9 = 2k0.
0V4 ACROSS R4 FOR 1.1mA MEANS R4 = 363R.
V_{b1} = 1V SO THAT R1 = 8k0, R2 = 1k0.
R9/R4 = 6 SO FOR A GAIN OF 100, R7/R8 = 100/6 = 16.
THUS R8 = 5k0/16 = 300R.

Fig. 7 A three-transistor circuit, making use of a PNP transistor to change DC voltage levels and contribute gain.

Which Do You Prefer?

Finally, an important point. The resistor values which are obtained by calculation are seldom preferred values, so obviously you have to substitute the nearest preferred values. This is not a major worry, because feedback amplifiers are self-correcting, but there will be one resistor in each circuit whose value sets the gain and which will have to be adjusted — R8 is such a value in Fig 7. The other point which is often raised is — why bother about running Q1 at low current to reduce noise? Since this is a feedback amplifier, the feedback will do a pretty good job of noise reduction anyhow. The answer is that you use negative feedback to stabilise gain and operating conditions. It doesn't make a lousy amplifier into a good one; all it can do is to make a sound design easy to analyse and reliable to mass-produce, so you aim to start with a good amplifier and add negative feedback to it.

CONFIGURATIONS 3

So far we've looked at the common-or-garden common-emitter configuration, but this time we examine the not-so-common common-collector and common-base circuits. It's un-commonly good!

The usual configuration for a transistor amplifier is the common-emitter one that we have used in each circuit so far. A number of useful and interesting circuits, however, are based on the use of common-collector and common-base configurations, and even more interesting variations can be assembled using two or more transistors in these circuits.

The classic single-transistor common-collector (or emitter follower) circuit is shown in Fig. 1. As shown, this uses two resistors to set the DC base voltage, but these and the input coupling capacitor C1 can usually be dispensed with by using direct coupling (Fig. 2) to the collector of the previous stage when the common collector stage follows another amplifying stage. The design of the circuit is simple; just decide what current, I, that you want to flow in the transistor, and then the value of the emitter resistor R3 (in Fig. 1) is $(V_b - 0.6)/I$. The voltage gain is less than 1 (in other words, there is a loss of signal amplitude in the circuit) but the input impedance (resistance, if you're using low frequencies only) is very high and the output impedance is very low, which is ideal for a lot of purposes. It's particularly useful, for example, when placed between two common-emitter amplifier stages, because the emitter follower acts as a high resistance load for one amplifier stage and as a very low-resistance supply for the second stage. In this way, it's possible to get more gain from two transistors by adding a stage which causes a loss!

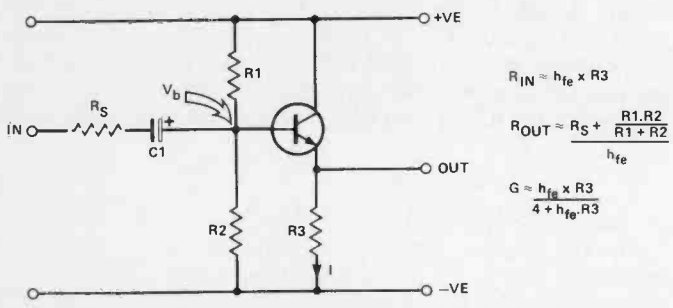


Fig. 1 The straightforward emitter-follower (common-collector) circuit and bias network.

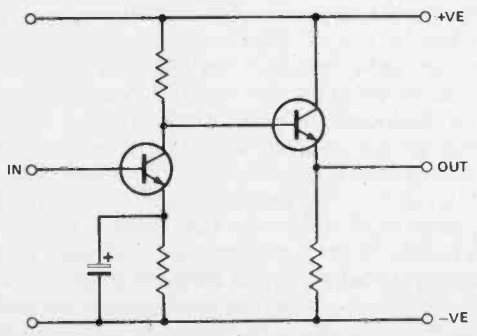


Fig. 2 Using an emitter-follower with direct-coupling.

Low Gain Is No Loss

Though the voltage gain of the emitter-follower is less than 1, and can be a lot less than 1 if the emitter resistor has a low value, the stage has a current gain which is about the same as the current gain measured for the transistor operating in common-emitter mode. This makes the emitter-follower a useful driver stage where extra current is needed, but some care is required when the circuit is used with pulses or high-frequency signals. There are inevitably stray capacitances across the emitter resistor, and these will be charged by the current flowing through the transistor when the base voltage goes positive. When the base voltage drops, however, the emitter voltage cannot change at a rate faster than that permitted by the time constant of the emitter load resistor and the stray capacitance, so that the trailing edge of a pulse has a slow fall time. This principle can be deliberately used in a demodulator for AM signals, by connecting a capacitor across the emitter resistor so that the time constant is long compared with the carrier wavetime but short compared with the wavetime of the modulation.

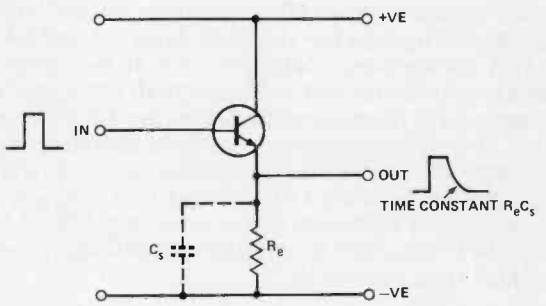


Fig. 3 The effect of stray capacitances on a positive pulse. The emitter-follower cuts off at the negative edge, leaving the stray capacitance to discharge through R_e.

Doubling Up

If pulses and high-frequency signals have to be used, a double emitter-follower is a better bet. The basic circuit is illustrated in Fig. 4a — it consists of an NPN emitter follower in series with a PNP one. The bias network uses two resistors R1 and R2, both of high value, and the diodes D1 and D2. The values of R1 and R2 have to be equal and large (100k to 2M2) to set the current through the transistors to a suitable value, usually 1 mA to 10 mA. Two parallel coupling capacitors are shown, but an alternative arrangement is a series coupling capacitor arrangement as shown in Fig. 4b. We'll look at this particular configuration in more detail in a later part when we consider power output stages, because it's the basis of most output circuits in transistor amplifiers.

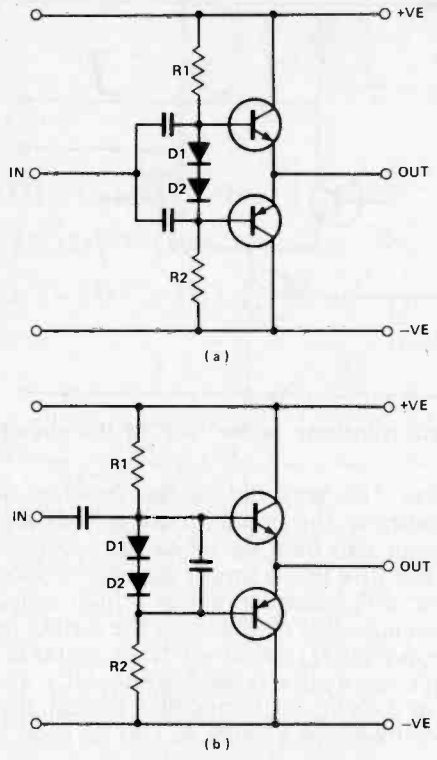


Fig. 4 a and b Two different arrangements of a double emitter-follower, using PNP and NPN transistors.

Meantime, take a look at the arrangement of Fig. 5, which is a type of double-emitter-follower. This circuit is often referred to as a Darlington Pair, a name I prefer to reserve for the version of the circuit which uses a load resistor in the collector circuit of both transistors. Bias is arranged in the same way as for a single emitter-follower, but the DC voltage drop between the input and the output is around 1V2, since there are two base-emitter junctions in series. The input resistance is very high, about $R_e \times h_{fe1} \times h_{fe2}$, in parallel with the bias components, and the output resistance is very low, about $R_s/(h_{fe1} \times h_{fe2})$. The current gain is $h_{fe1} \times h_{fe2}$, which is also very large. For example, if we use transistors with h_{fe} = 100 for both stages, then the compound current gain is 10,000, the input resistance with R_e = 10k is 1M0, and the output resistance when the source resistance is 10k will be 1 ohm! For a lot of applications, Q2 can be a power transistor with a low value of h_{fe}, and the circuit can be used to simulate the action of a power transistor with a high h_{fe} value. The Darlington circuit of Fig. 6 might be expected to have a very high voltage gain, but does not — there is only one amplifying transistor, and the signal feedback from the collector of Q1 back to its base will reduce the overall gain quite noticeably if the circuit is driven (as it usually is) from a high value of source resistance.

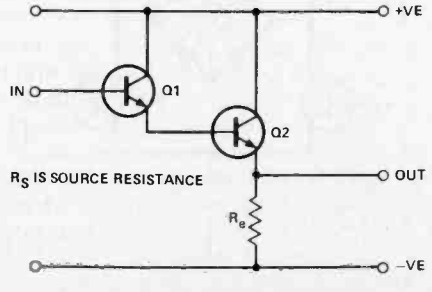


Fig. 5 A compound emitter-follower.

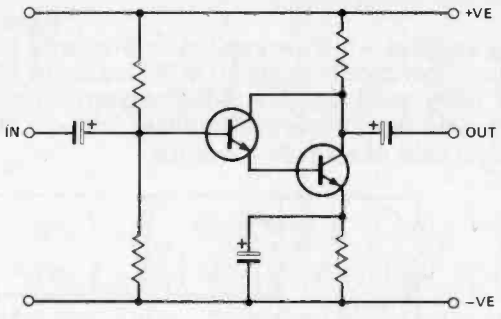


Fig. 6 A Darlington amplifier circuit.

Transistor Load? That's Cascode

The cascode circuit was one that was familiar to TV service engineers back in the days of valves (remember them?), but its transistor equivalent has never been so widely known, which is rather a pity. A cascode circuit consists of a common-emitter amplifier stage directly coupled to a common-base amplifier stage (Fig. 7). The input resistance is fairly low but the output resistance is very high, which makes the circuit particularly suitable for use with tuned-circuit loads or any other type of load which has a very high impedance. The gain is of about the same value as for a single transistor, but the impedance-transforming action (the reverse of the action of the emitter follower) can be very useful.

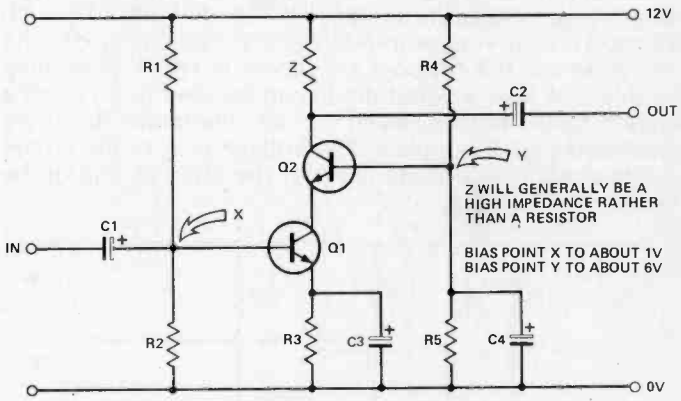


Fig. 7 The cascode circuit, which uses a common-emitter stage driving a common-base stage.

Long-Tailed Pairs Live On!

Of all the two-transistor amplifier circuits, though, the most commonly used is the long-tailed pair, simply because it features so much in linear ICs. The long-tailed pair, a circuit adapted from the days of valves, is a differential amplifier; basically this is an amplifier with two inputs, whose output voltage is proportional to the difference between the signal voltages at the two inputs.

The basic circuit is shown in Fig. 8. The emitters of the transistors are connected, and the bases are biased to about the same DC voltage. The bias is correct when both transistors contribute the same amount of current to the common emitter resistor, and this may need some adjustment unless the transistors are well matched. For ideal action, the value of R4 has to be high, so that large values of R5 and R6 are also needed. A true differential signal at the bases will cause one base voltage to increase as the other decreases, so causing one collector voltage to decrease as the other increases. This creates a large difference signal between the collectors, and ideally the total current through R4 does not change. For a common-mode signal,

meaning a signal which is applied in the same phase to both bases, the collector signals will also be in phase so that the differential signal is, ideally, zero. The voltage gain, operated as a differential amplifier, is about the same as the ideal gain of a single transistor.

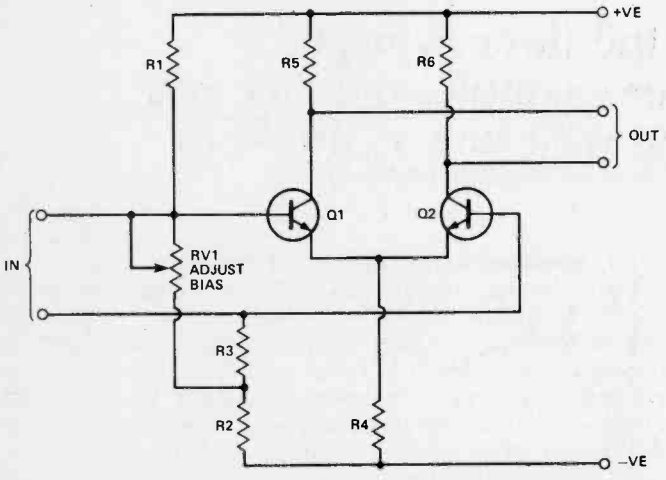
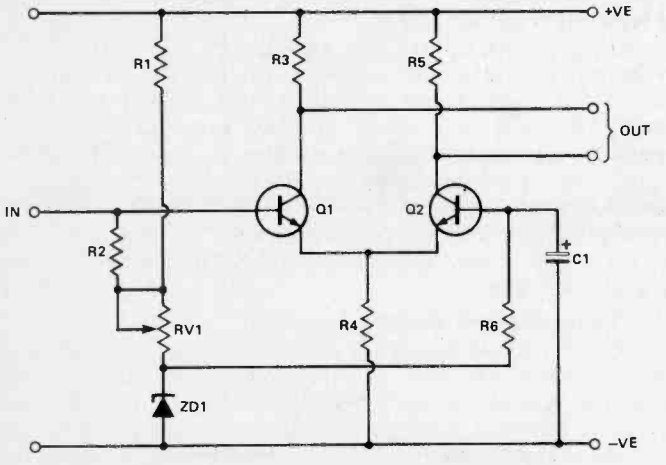


Fig. 8 The long-tailed pair (or balanced amplifier) circuit. This version is the balanced in, balanced out type.

The long-tailed pair is often used as a way of converting a single-ended input into a differential input for such purposes as driving push-pull circuits. The changes to the circuit to suit this purpose are shown in Fig. 9, illustrating in this case how a zener diode can be used to establish a 'half-supply' voltage level, as an alternative to using balanced power supplies. The voltage gain of the circuit when used in this mode is about the same as that of the differential mode.



BIAS: THE BASE VOLTAGES ARE SET BY THE VALUE OF THE ZENER DIODE. R4 SHOULD BE HIGH (22k OR MORE), AND THIS VALUE WILL SET THE TRANSISTOR CURRENTS I_{c1} AND I_{c2} SO THAT $R4(I_{c1} + I_{c2}) = V_b - 0.6$. CHOOSE THE VALUES OF R3 AND R5 TO GIVE COLLECTOR VOLTAGE LEVELS MIDWAY BETWEEN THE SUPPLY VOLTAGE AND THE EMITTER VOLTAGE.

Fig. 9 A long-tailed pair used to convert unbalanced signals into balanced.

Transistor Tails

The restriction that affects this type of circuit is the value of the common emitter resistor. Since the value should be high to achieve true differential amplifier action, the currents in both transistors are forced to be small, and this is not always desirable unless the amplifier is used in an early stage. The way out of the problem is to use another transistor as the emitter load, as indicated in Fig. 10. The currents can then be set to suit whatever values are needed, because this is simply a matter of biasing. The 'tail' transistor will, however, act as a high resistance for AC signals, because this resistance is the output resistance of the common-emitter transistor. This is nearer to the type of differential circuitry that is used in linear ICs, and further refinements of the circuit are possible, though the effort is not really worthwhile if a linear IC can be used instead.

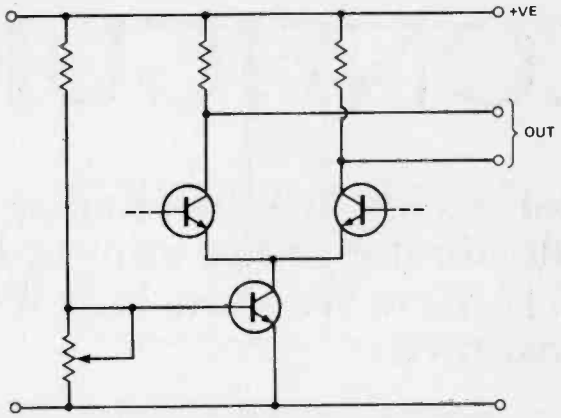


Fig. 10 Using a transistor as the 'tail' of the circuit.

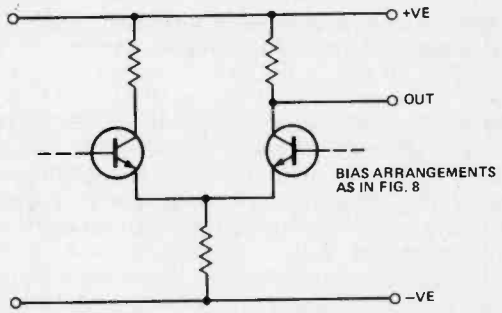


Fig. 11 Obtaining an unbalanced output at one collector.

The long-tail pair circuit turns up in all sorts of applications, and one of them is the conversion of differential signals to single-ended output as illustrated in Fig. 11. The gain in this type of circuit is only half as much as can be expected in the differential mode. Another frequently used circuit is the metering circuit of Fig. 12 in which the differential signals at the output of the long-tailed pair are applied to a bridge rectifier and used to drive a meter. No part of the meter circuit is earthed, and zero setting is carried out by adjusting the biasing of the differential amplifier. This is a very useful basis for an AC milliammeter circuit.

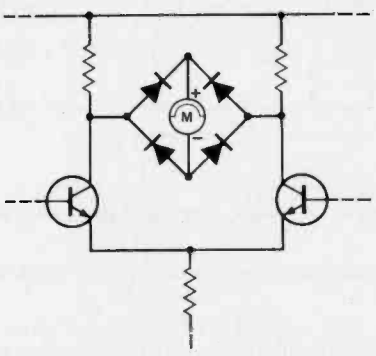


Fig. 12 Driving a bridge rectifier circuit.

CONFIGURATIONS 4

Most readers will be familiar with the three types of multivibrator in the form of ICs (for example, the 555 and 4013). Here we show how it's done with just a couple of transistors.

The multivibrator is a circuit of remarkable antiquity — it is attributed to Abrahams and Bloch at around 1918. Over the years, as various valve and subsequently transistor versions were devised, a variety of names were attached to them. However, the names astable, monostable and bistable are particularly useful as descriptions of the various multivibrator circuits.

The astable is a circuit which oscillates continually (no stable state — hence the name), producing steep-sided waveforms whose frequency can be adjusted by changing RC time constants. The monostable has only one stable state, and an input pulse will disturb this state for a time that depends on the time constant of the circuit, following which it returns to the stable waiting state. This circuit is widely used as a pulse generator, because ideally the duration of the pulse (the pulse-width) is independent of the repetition rate of the pulse, which is determined by the rate at which the monostable is triggered. The bistable circuit (two stable states) is the basis of digital circuitry, but is seldom used in discrete form nowadays because of the low cost of digital ICs. A source of confusion over names, incidentally, has been the use of 'flip-flop' by digital circuit designers, whereas the name was traditionally used to mean a monostable.

Astable Antics

Two varieties of astable exist, the parallel and the serial, of which the parallel is much the better known. The basic circuit is shown in Fig. 1, but unless your needs are very simple, the waveform from this circuit is not really good enough without reshaping. A much better circuit is shown in Fig. 2; this uses a diode to isolate the collector from which the output is taken. When Q2 cuts off, its collector voltage can rise sharply, leaving D1 reverse biased. In the simple circuit, a sharp rise of voltage at the collector of Q2 is made impossible by the capacitor C2 which has to be charged through the collector load resistor (R4 in Fig. 1). By using the diode, the collector voltage can rise sharply, and the charging of C2 is done at a slower rate by the additional resistor R5. This ensures a much sharper shape of waveform at the output.

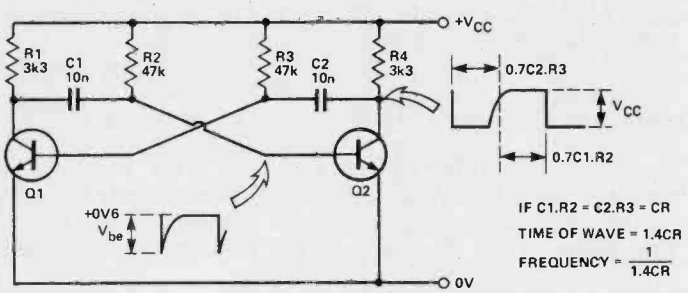


Fig. 1 The simple multivibrator astable, with period formula.

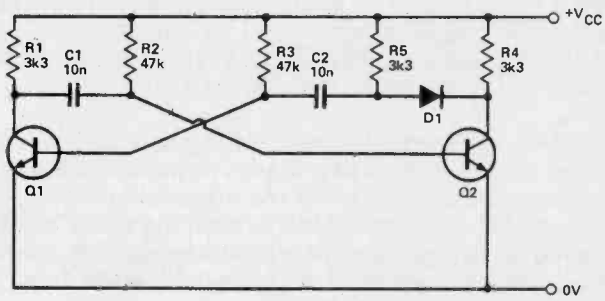


Fig. 2 Using an isolating diode to sharpen the leading edge.

Another problem of the simple circuit is that, contrary to theory, its frequency changes as the supply is changed. This is because silicon planar transistors will conduct readily in the reverse direction when the base voltage is negative with respect to the emitter (for an NPN transistor). This is a form of zener breakdown, but it can be prevented by connecting silicon diodes with a higher reverse breakdown voltage in series with the base leads, as shown in Fig. 3, so greatly improving the frequency stability of the astable.

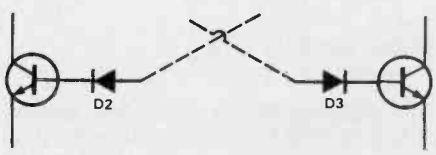


Fig. 3 Using base-isolating diodes to prevent base-emitter breakdown.

Formula For Success

The astable frequency is given by the formula shown in Fig. 1 — it depends on the two sets of time constants. These should not be greatly different — don't be tempted

to try to produce pulses with very large or very small values of mark-to-space ratio by using an astable with very different values of the two time constants. The waveform will probably be disappointing, and the circuit may not start oscillating reliably.

The frequency stability of an astable is generally poor compared with that of the LC type of oscillator, and this feature makes the astable particularly useful inasmuch as it can be easily synchronised to external pulses. Unless the 'natural' frequency of the astable is reasonably close to the incoming frequency, however, synchronisation cannot be relied on, and slowly-changing waves such as sinewaves are not useful for synchronisation because the triggering point is liable to vary (or jitter) from one wave to the next. The astable can be synchronised by pulses at one base, and this can be done by pulsing a cut-off base into conduction or by pulsing a conducting base to cut-off.

Whichever method is used, the trigger pulse should be isolated from the astable by diodes to prevent the astable interfering with the action of the trigger circuit (Fig. 4). If the 'pulse-off' method is used, a catching diode must be included to prevent the transistor base-emitter junction from being reverse-biased which would cause zener action on each negative pulse.

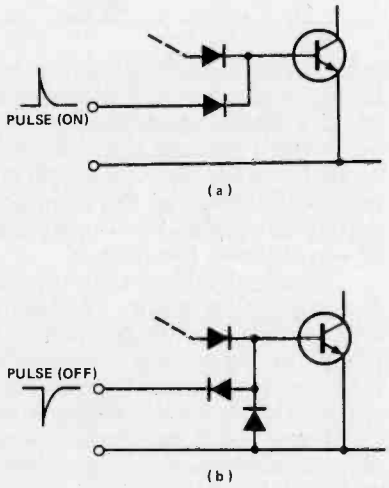


Fig. 4 Diodes are used in the synchronising circuits, too!

Breakfast Serial?

The other type of astable, much less well known, is the serial astable, which is a particularly good way of producing very short pulses which can pack a lot of energy. A circuit is shown in Fig. 5, and since the circuit is not so familiar as that of the parallel astable, a run through its action might be of interest. Note that only one time constant, R1.C1, is used, and that the transistors are complementary; one PNP, one NPN.

We can start by supposing R2 = R3 and V_{cc} = 10 V, with C1 discharged so that the emitter of Q1 is at a low voltage. Q1 will be cut off, and will stay that way until its base voltage is more negative than its emitter voltage, or to put it another way, until its emitter voltage rises to more than its base voltage. With the base voltage fixed at +5 V by R2 and R3, the capacitor will have to charge to around 5V6 before much will happen.

Meantime, because Q1 cut off, no current is reaching the base of Q2, and this transistor also is cut off. The circuit remains with neither transistor conducting until the charging capacitor reaches the voltage at which Q1 turns on. This also turns on Q2, because the collector current of Q1 goes to the base of Q2. This in turn drastically lowers the base voltage of Q1, and the emitter voltage will follow it, rapidly discharging C1. Once the emitter voltage of Q1 drops, however, the circuit recovers, and we're back where we started.

Unlike the parallel circuit, in which one transistor conducts while the other is cut off, the serial multivibrator spends most of its life with both transistors cut off, and only brief intervals with both turned on. The cut-off time is the time needed to charge C1 through R1 to about 0V6 above the voltage supplied by R2 and R3 — the formula for the time is shown in Fig. 5. The time for which both transistors are on is less easy to estimate because it depends on the effective resistance of the transistors at saturation; it is normally very short compared to the charging time.

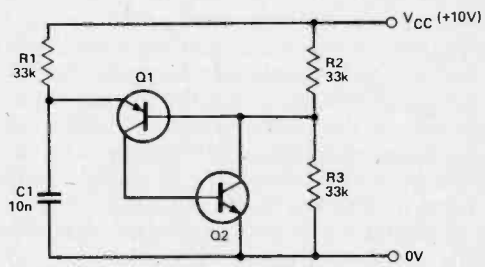


Fig. 5 The serial astable circuit, using complementary transistors.

Monostables

The classic monostable uses a parallel circuit with one DC coupling, as shown in Fig. 6. With no trigger-pulse input, Q2 is held on because of current flowing through R2. The collector voltage of Q2 is very low, and this ensures that Q1 is held off. This is the stable state of the circuit, and it wil remain in this condition, with C1 charged, until a positive-going trigger pulse arrives, with enough amplitude to make Q1 conduct. This pulse causes current to flow in Q1, so that the collector voltage drops, and the drop in voltage at the base of Q2 cuts Q2 off. This condition lasts until C1 charges through R2 sufficiently to turn the base of Q2 on again, when the circuit switches back. The diode in the base lead of Q2 prevents base-emitter breakdown from affecting the timing.

As in all parallel circuits, there is always one transistor conducting and the other cut off. The serial version of the monostable (Fig. 7) uses complementary transistors, and will pass no current in its waiting state. When a trigger pulse arrives, Q2 switches on, and the voltage at its collector drops. This switches on Q1, via the capacitor C1, causing the base circuit of Q2 to be heavily forward biased. C1 now charges through R2 until the voltage at the base of Q1 rises to its cut-off value of around V_e—0V6. Both transistors then cut off.

The pulse width of the output depends on the time constant of C1.R2, and will vary as the supply voltage varies. Diodes can be used to prevent base-emitter breakdown in the usual way.

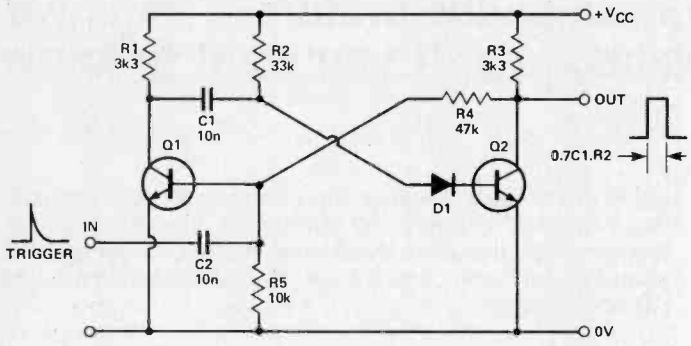


Fig. 6 The parallel monostable circuit.

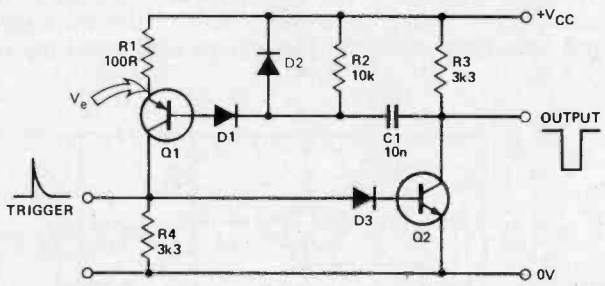


Fig. 7 — A serial monostable — a useful circuit which is seldom seen.

Bistables, Two By Two

Figure 8 shows the classic bistable circuit using two NPN transistors. The circuit is stable with either Q1 on and Q2 off, or in the alternate condition of Q1 off and Q2 on. Switching is done by using the A or B inputs. With Q1 on, a negative pulse at A will switch the circuit over, and a negative pulse at B will switch it back. This action corresponds to that of the simple set-reset latch.

Counting action may be obtained if steering diodes are added to the basic circuit, as illustrated in Fig. 9. Suppose Q1 is conducting: its base voltage will be around 0V6, and its collector voltage about 0V1, so that D1 is almost conducting. D2 is cut off because its anode voltage will be about zero and its cathode voltage almost at supply voltage. C1 will carry virtually zero charge, while C2 will be charged to around supply voltage, V_{cc}. When a negative trigger pulse of amplitude around 1 V is applied D1 conducts, allowing Q1 to be cut off by the trigger pulse; however, D2 is held off by the charge on C2. The bistable then changes over so that Q2 is fully conducting and Q1 cut off. In this condition, it is D2 which is biased almost on and D1 completely off. When the next trigger pulse arrives, then, Q2 will be cut off by it, and the circuit will switch back to its original state. Hence, if the circuit is triggered regularly, either output will be a square wave with half the frequency of the trigger pulses.

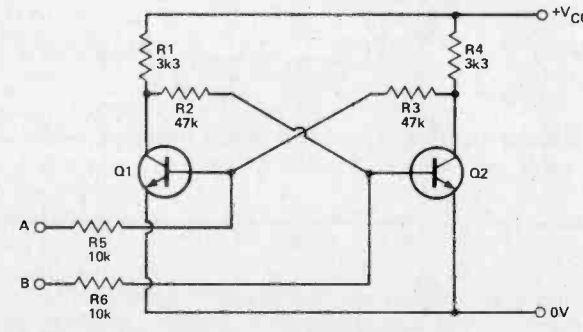


Fig. 8 The simple R-S type of bistable.

The waveforms at the collectors of a bistable can be much closer to square than those from simple astables, so that one way of creating well shaped square waves is to drive a bistable from an astable (Fig. 10). In this circuit, the output from the bistable has the same frequency as the astable. Nowadays, the ready availability of bistables in integrated form discourages us from using the discrete variety — we'll look at IC types later in this series.

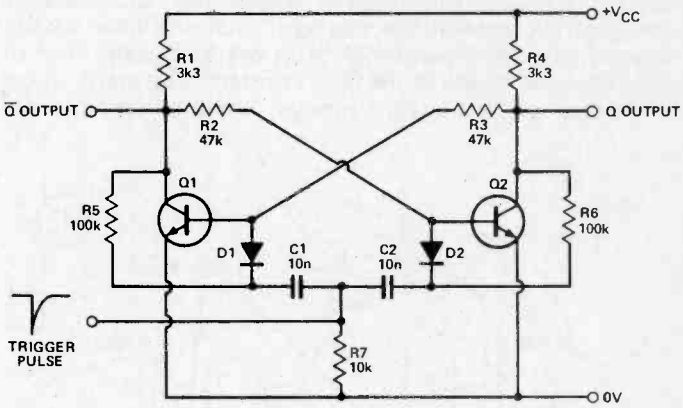


Fig. 9 A bistable with steering diodes to give the scale-of-two action.

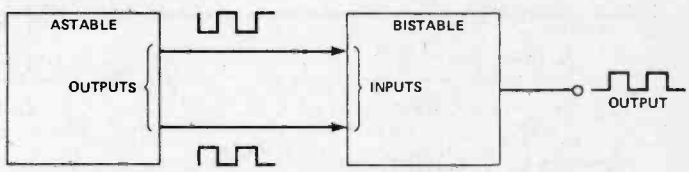


Fig. 10 Following an astable with a bistable to sharpen the waveform.

CONFIGURATIONS 5

Transistors as amplifiers, transistors as multivibrators — now we consider transistors as sawtooth generators. If you want to know the timebase, ask Ian Sinclair.

The timebase is a circuit which generates a sawtooth waveform, one whose voltage changes linearly with time: a graph of voltage plotted against time will be as shown in Fig. 1 (though it may be either positive-going or negative-going). The best-known application is in oscillosope timebases, but the circuit can also find use in digital-analogue converters and in timing circuits.

The most simple timing circuit is, of course, a capacitor charging through a resistor (Fig. 2). The time constant CR determines the total charging time which, though theoretically infinite, is in practice about four or five times the length of the time constant. The graph shape of voltage plotted against time is, however, exponential rather than linear because the charging current drops as the capacitor charges. All timebases of the capacitor-charging type therefore need some method of keeping the charging current constant as the voltage across the capacitor rises.

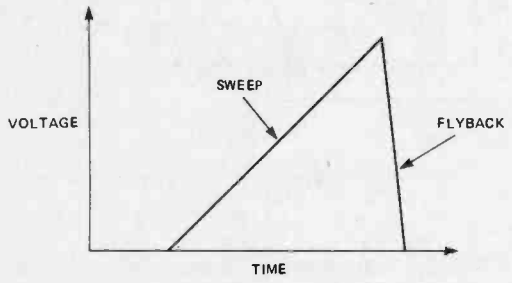


Fig. 1 The waveform of a perfect timebase — this should be a straight line.

Transistor Control

In the days of valves, many elaborate circuits were devised to overcome the problem of constant current control, but it took the development of the transistor to come up with a really simple system with good perform-

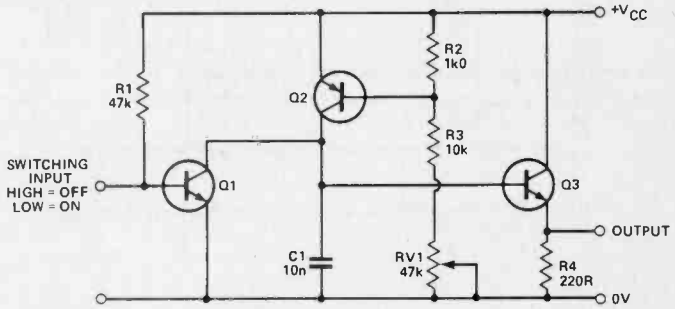


Fig. 3 Using a transistor in place of a resistor for capacitor charging. Since the current through the transistor remains constant, the sweep waveform is straight rather than exponential.

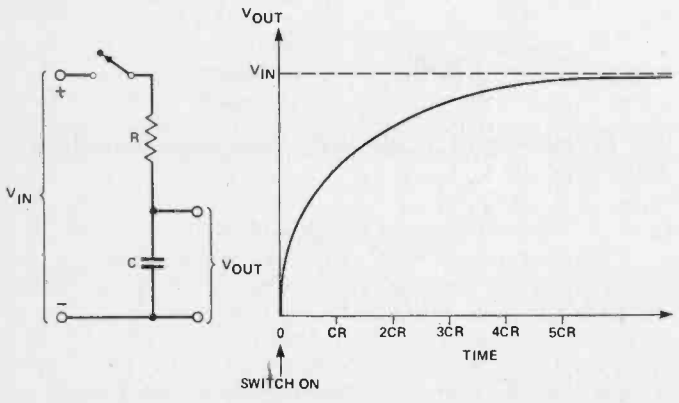


Fig. 2 Capacitor charging. When a capacitor is charged through a resistor the waveform is an exponential rather than a straight line.

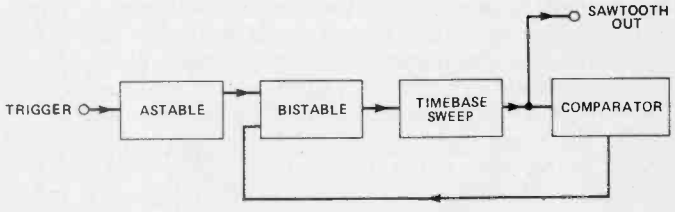


Fig. 4 Block diagram of an oscilloscope timebase.

ance. A transistor whose base-emitter junction passes a constant current will also pass a (larger) constant current between its collector and its emitter, and this current can

be maintained up to the level where the collector voltage is less than half a volt different from the emitter voltage.

Figure 3 shows a simple timebase circuit using this principle. Q1 is a switching transistor which is normally conducting, keeping the voltage across the capacitor low. Q2 is a PNP transistor whose base current is set by the resistor chain R2, R3, RV1, and which can be varied by altering the value of RV1. Since the base current is constant, the collector current will also be constant. Q3 is simply an emitter-follower to avoid non-linear effects which would be caused by a resistive load connected across the charging capacitor (since a resistance takes more current as the voltage across it is increased). For best results, Q3 should be a transistor with a high h_{fe} value, and a double emitter-follower is often preferable to ensure the highest possible input resistance.

The action is as follows. When Q1 is cut off by a negative pulse at its base, capacitor C1 can be charged by current flowing through Q2. This current will not change until the collector voltage of Q2 has reached a value close to the positive supply voltage, so that the wave form is linear up to this region. If Q1 remains cut off, the wave-form will then flatten off, but if Q1 is switched on again before this point is reached, then a good sawtooth shape is preserved.

Timing The Timebase

The action depends to a large extent on switching the transistor Q1 at the correct times, and all timebases consist basically of two sections — a square wave generator which handles the switching and a sawtooth generator which provides the desired waveform. An oscilloscope timebase would use a level-detecting circuit at the output to ensure that the switching transistor Q1 was switched off before the voltage level at the output reached the non-linear region — a block diagram with waveforms is shown in Fig. 4. In this arrangement, the repetition rate of the timebase is determined by an astable which provides a trigger pulse. The trigger pulse sets the bistable, which in turn cuts off the switching transistor of the timebase generator and so starts the charging of the capacitor. When the charging has reached some preset voltage level, the level detector (comparator) circuit switches the bistable back, so discharging the capacitor ready for another sweep. For many oscilloscope purposes, the astable is set to run freely at a low speed, and is synchronised to whatever waveform is to be displayed — this is the auto timebase system found on most modern oscilloscopes. The sweep speed is then determined by the time constant of the charging capacitor.

The use of a transistor as a constant current device for a timebase is good enough for many purposes, but two other methods of creating linear sweep waveforms from the basic capacitor charging circuit have been well established for many decades in oscilloscope circuitry. One of these is the bootstrap circuit. Bootstrapping is positive feedback applied over a circuit in which the gain is less than unity, so that it does not cause oscillation.

By His Bootstraps

The principle of the bootstrap is shown in Fig. 5. A capacitor is charged through two series resistors, and a unity-gain amplifier is connected so that the voltage across the capacitor can be applied, in phase but with its DC level shifted, to the point where the resistors join. When the capacitor starts to charge, the increase of voltage across the capacitor causes a matching increase of voltage across R2, so that the voltage across R2 has not changed in this time. Since the voltage across R2 is constant, the current through R2 is also constant, which is the condition for a linear sweep.

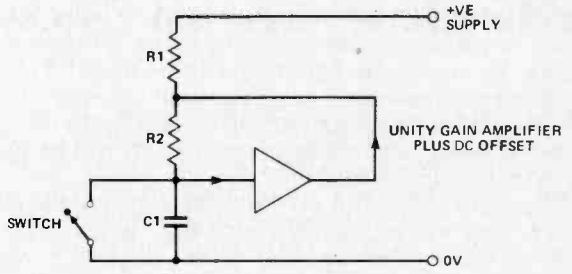


Fig. 5 The principle of the bootstrap timebase circuit.

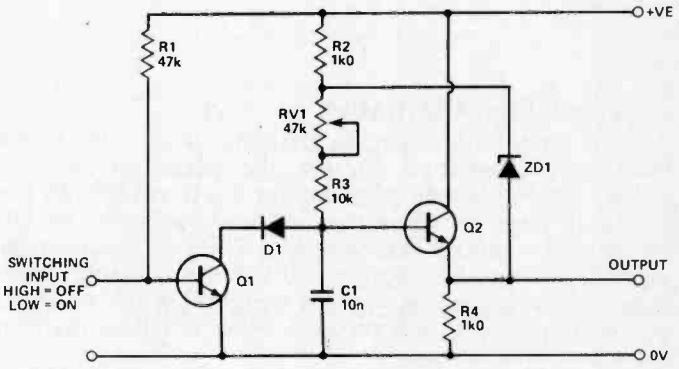


Fig. 6 A practical form of the bootstrap circuit, using an emitter-follower as the unity-gain amplifier.

The bootstrap depends on being able to keep the voltage at the junction of the resistors at a constant amount greater than the voltage across the capacitor. The whole idea seemed so absurd when it was first proposed that the (US) inventor remarked that it seemed "rather like lifting yourself by your own bootstraps". As so often happens, the name stuck.

A practical form of the timebase is shown in Fig. 6. Q1 is, as before, the switching transistor which starts and stops the sweep. The charging resistor chain consists of R2, R3 and RV1, of which R3 is a limiting resistor whose value is set so that excessive current does not flow through Q1 when the variable is set at its minimum value. D1 is used to prevent C1 from discharging below about 0V7, so ensuring that Q2 will not switch off, causing non-linearity. If Q2 is allowed to switch off, then the timebase output will have a decided 'kink' at the voltage at which Q2 switches on.

Q2 is an emitter-follower, whose emitter is connected through a zener diode ZD1 to the junction of R2 and R3. The zener diode, along with the base-emitter voltage drop of Q2 determines the voltage across R3 and RV1, so that the charging rate can be calculated. For example, suppose the voltage is 6 V, the values of RV1 and R3 add to 56k and C1 is 22nF. The charging current I is 6/56 mA, which is 0.107 mA, and the rate of change of voltage across C1 is I/C1. Using units of milliamps and nanofarads, the rate of rise of voltage will be in volts/microsecond, and the example gives 0.00486, equivalent to 4.86 volts per millisecond. If you know the sensitivity figure for the cathode ray tube for which the timebase is to be used (in terms of centimetres of deflection per volt), then you can calculate what amount of amplification will be needed to obtain full screen coverage, and what time constants will be needed for the various scan speeds.

There are limitations on the voltage gain of the emitter follower and the frequency range over which the zener diode remains effective, but with suitable choice of components, good timebase circuits can be designed around this core configuration. Commercial circuits of this type often look remarkably complicated, but once the bootstrap section is separated from the other parts of the complete timebase (the triggering and the comparator sections), the essential simplicity of the circuit can be seen.

The Miller Alternative

The other basic capacitor charging circuit is the Miller integrator. These two circuits, the bootstrap and the Miller, were curiously polarised for many years, with the bootstrap used on US equipment and the Miller on UK equipment almost exclusively. This is no longer completely true, but though you will see bootstrap timebases appearing on equipment designed in this country, you will even now seldom see a Miller timebase used on the other side of the pond.

The Miller timebase is named after (yes, got it!) Miller, who discovered the result of negative feedback across the anode-grid capacitance of triode valves. The name became attached to the timebase (which was not designed by Miller) because the Miller timebase makes deliberate use of such feedback to achieve linearity. The basic circuit is shown in Fig. 7, and the most startling thing about it is its simplicity, because the switching transistor is also the current regulator! If we imagine the transistor starting cut-off, then a square wave applied to the input will raise the base voltage until the transistor starts to conduct. When conduction starts, however, the collector voltage will drop, and the negative feedback through C1 will prevent the base voltage from rising to the level of the input voltage. Once this has happened, the base voltage can rise only as fast as the capacitor C1 can be discharged, and the discharge is at a steady rate because of the negative feedback.

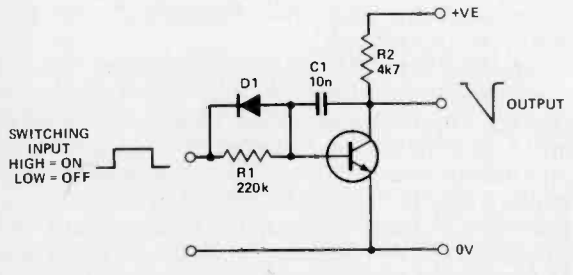


Fig. 7 The basic Miller timebase circuit.

The time constant for the Miller integrator is given by the value of R1 and C1 rather than R2 and C1 as you might expect, and the conventional use of the circuit as shown here produces a timebase waveform which is negative-going, with a small 'step', as shown in Fig. 8, just at the point where the transistor switches on.

The circuit will operate in the opposite direction, when the 'free' end of R1 is at ground potential. In this case, the voltage at the transistor's collector rises just quickly enough to keep sufficient current flowing into its base (and also R1) to keep it on. In both cases, the simplest way to achieve the fly-back is to connect a diode, D1, in parallel with R1. For operation in the opposite direction from that first described, the direction of the diode must be reversed.

More elaborate versions of the Miller use two stages of amplification with the output in phase, and a low-impedance stage driving the capacitor. Very good results can be obtained, and with a wide-band op-amp used in place of a transistor, excellent timebase linearity is possible.

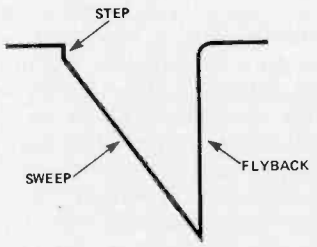


Fig. 8 The waveform from the simple Miller circuit.

Before we leave the subject, timebases can also make use of the growth of current through an inductor. The effect that is used here is the inductive equivalent of capacitor charging, and it is useful because if the inductor is also a deflection coil for a cathode-ray tube, then the timebase and deflection system can be combined. Linearity is much less easy to achieve, however, and one method is the use of a saturable reactor in series with the inductor which carries out the timebase action. The inductance of a saturable reactor will vary with the amount of voltage across it in order to keep the current constant. Using this and other components, it is possible to balance out the worst of the non-linearity of the charging process. For truly linear timebases, however, the capacitor charging circuits which we have described in this article are considerably superior to inductive timebases. No-one watching TV seems to care too much if the characters are very slightly fatter on the right hand side of the screen than on the left, but we need to know the truth from our oscilloscopes!

CONFIGURATIONS 6

And so to op-amps, the most venerable members of the linear IC family. Ian Sinclair traces their ancestors, descendents and lifestyle.

Before the reason for the name becomes shrouded in the mists of history, perhaps it's just as well to look at the origins. Operational amplifiers were designed for analogue computers, which are machines used for solving mathematical equations. They do so, not by using binary arithmetic as digital computers do, but by connecting up a network of components which represents either a mathematical relation or an equation. In the case of a mathematical relation (eg, $y=x^2$) the circuit will have an input, x, and an output, y, that will vary according to the relationset up and according to the value of x. Equations can be either ordinary (eg, $x^2+4x+3=0$) or differential (eg, $d^2y/dx^2+x=0$); the circuit will be connected in a loop, and in the case of the ordinary equation it will give an output that represents the solution (or one of the solutions) to the equation. The solution to a differential equation is itself a mathematical relation (in the case of the example given above, $y=A\sin x+B\cos x$), so the circuit will have an input and an output (the coefficients of the equation, A and B, will be determined by the initial values of the circuit voltages, but that takes us a bit beyond the present scope of this article).

An essential part of representing a mathematical operation in electrical terms is an amplifier with very high gain whose frequency response can be modified by using negative feedback. Typical operations that can be simulated by amplifiers of this sort include the mathematically important ones of differentiation and integration (Fig. 1), and the amplifiers which were designed for these purposes very reasonably became known as operational amplifiers.

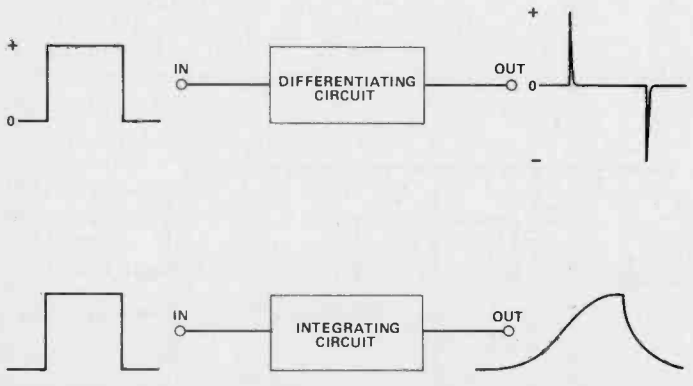


Fig. 1 The operations of differentiation and integration performed on a square wave.

The Perfect Specimen

The specification for a perfect operational amplifier was that it should have infinitely high gain, infinitely high input resistance, zero output resistance, and as much bandwidth as was needed — it was particularly important to have the gain maintained right down to DC. Analogue computers are still produced, though they don't have the importance they once had, and the operational amplifiers which were once made using valves, and then transistors, are now made as ICs. The requirements are still pretty much the same, because our definition of an operational amplifier nowadays is as a high gain DC-coupled amplifier whose behaviour can easily be controlled by using negative feedback. Since the behaviour (gain, bandwidth, shape of gain-bandwidth curve) is so easily modified by the use of negative feedback, the operational amplifier is the nearest thing we have to an all-purpose amplifier, and that's why operational amplifiers were among the first linear ICs that were produced.

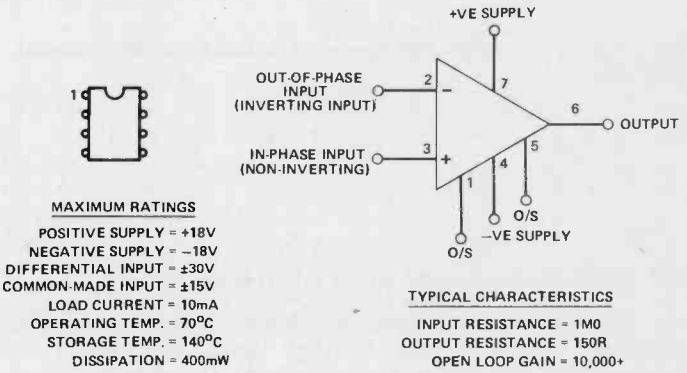


Fig. 2 Part of the specification for the 741 op-amp.

To start with, consider the typical specification of one of the best-known op-amps, the 741. This is illustrated in Fig. 2, to show how close we can get to the ideal specification. One point of importance is the bandwidth. If you use a 741 at its full gain, you must expect the bandwidth to be very severely limited — less than 100 Hz at maximum gain. Some care has to be taken if 741s are used in audio circuits, because in some feedback circuits that include filtering the chip may be working at a very high gain at the ends of the bandwidth, even though its midband gain is low.

Offset Problems

Getting down to configurations, the main point about op-amp circuits is how to bias them. Very few applications call for the 741 to be operated as a differential DC amplifier at full gain, but for these applications a balanced power supply is needed. Additionally, some form of input

offset balancing will be needed. This is necessary because there are bound to be some very small mis-matches between the resistors and transistors that make up the two input circuits (see later). The gain of the op-amp is so high that any imbalance will be amplified up, so that with both inputs tied to zero, the output of the op-amp will not be zero by quite a margin.

Manufacturers usually specify typical and maximum *input offset voltage* and *input offset current*. These are the *differences* between the input voltages and the input currents (with both inputs very close to zero volts) needed to obtain an output voltage of zero. With the 741 and many other op-amps there are offset trim connections that allow you to trim out the voltage offset. A circuit for the 741 is shown in Fig. 3. However, the input currents will still be slightly different, and there may be the odd circuit

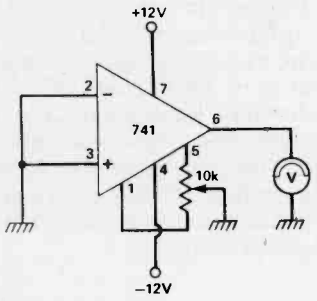


Fig. 3 Using the offset adjustment to balance out the internal currents.

for which this will need to be taken into account.

The offset adjustment will have to be repeated at intervals, because the settings *drift*. The effect of temperature and time conspire to make the output voltage change (drift) away from zero, so that an op-amp at full gain is a rather unstable device which needs frequent checking. Fortunately, we seldom need to make use of the full gain of the op-amp, and most of the circuit configurations make use of feedback bias circuits.

Figure 4 shows one of the most common bias methods. The circuit uses a balanced power supply, and bias is obtained by connecting a resistor between the output and the out-of-phase or inverting input (marked as –). The in-phase or non-inverting input (+) is connected to earth, so that the output voltage will be almost zero, just enough to apply the correct offset voltage (which is usually less than a millivolt) to the inverting input. The gain of this circuit depends on the resistance of the signal source. If we represent this as a resistor in series with the input, R1, then the gain is simply –R2/R1 (the – sign indicates that the signal is inverted).

This circuit is DC-coupled throughout, but if we do not need DC gain, then a single-ended supply version can be constructed, as indicated in Fig. 5. Capacitor coupling must then be used to avoid shorting out the bias voltage, choosing capacitors with low leakage, and the supply voltage must be adequate — the quoted minimum voltage across the chip is 3 V.

When this configuration is used, the inverting input voltage remains practically constant when a signal is applied. When a balanced power supply is used, in fact, the inverting input is virtually at earth voltage, and this 'virtual earth' effect means that signals applied to the input terminal (one end of R1) are flowing through R1 to a point

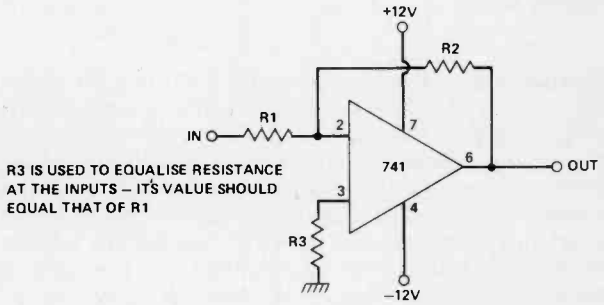


Fig. 4 The feedback bias system in a circuit which uses the out-of-phase, or inverting input for signals.

which is as good as earthed as far as signals are concerned. This makes the input resistance of the circuit equal to the value of R1, and it limits the application of the circuit to some extent, because if the input resistance is to be reasonably high, then the feedback resistor R2 will have to be of an unreasonably high value to achieve a modest gain. If the feedback resistor has too high a value (in the megohm region), then the bias currents at the input of the chip, typically 200 nA, will cause voltage drops which we can't ignore without making our calculations go considerably astray. The input resistance of the op-amp itself is large, but the use of negative feedback to the same

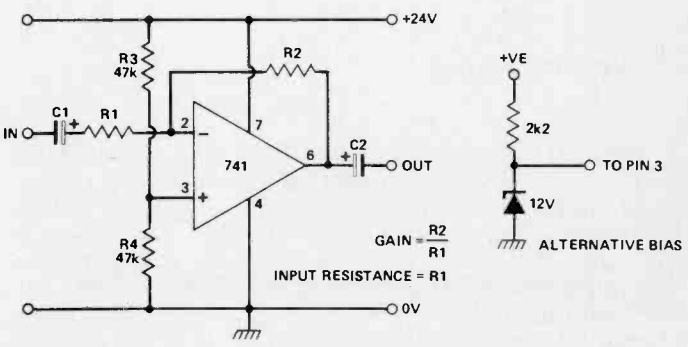


Fig. 5 A single-ended power supply version of the Fig. 4 circuit.

input as the signal makes the input resistance low because of the 'virtual-earth' effect.

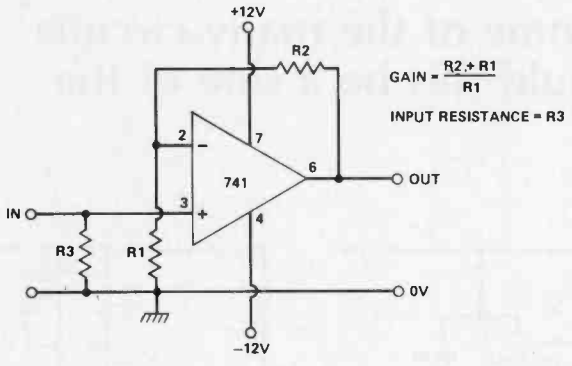


Fig. 6 Using signal input to the in-phase, or non-inverting input of the 741.

Improved Impedances

Another configuration of the op-amp is illustrated in Fig. 6. This time the input is taken to the non-inverting input, and the inverting input is used only for the feedback. In this balanced version of the circuit, the input resistance can be higher, because the resistance R3 does not control the gain of the amplifier, and the source resistance is of no interest unless it is unusually high. The gain is given by (R2 + R1)/R1.

It's quite straightforward to combine the biasing arrangements of Fig. 5 with the non-inverting circuit of Fig. 6. However, a word of caution: all those resistors and all those capacitors combine to form low-pass filters, and at frequencies around their cut-offs, these will all produce considerable phase-shifts, and this may lead to what you've designed as an amplifer actually turning out to be an oscillator!

Slewing About

The 741 type of operational amplifier has a lot a merits, but it is a design which is now showing its age. Much more recent designs have, in particular, wider bandwidths, and are impressively better in one respect — slew rate. The slew rate of an operational amplifier is the maximum rate-of-change of output voltage expressed in volts per microsecond, and it affects large signals (which change by a greater voltage) more than small signals. The point is that if the maximum rate of change of voltage is 1 V/us, then a 10 V change would need 10 us, and a 10 V signal is limited to one tenth of the bandwidth of a 1 V signal. The effect in practical terms is that the useful bandwidth of the amplifier for sine waves depends on the amplitude of the waves, and the shape of output for a square wave input also depends on the amplitude of the wave.

Slew rate limiting is caused by stray capacitances within the chip. When voltages change, these stray capacitances have to be charged or discharged, and the amount of current which flows in the input stages is very small, not enough to allow these capacitances to be charged or discharged quickly. All amplifiers suffer from this to some extent, but slew rate is much less of a problem for discrete component circuits whose circuits are not DC-coupled and which can therefore use large currents and small values of load resistors. The typical slew rate of the 741 is 0.5 V/uS, and this is rather poor in comparison with more modern designs such as the Motorola MC1741S, which has a slew rate of 15 V/uS.

The other feature of the 741 which causes problems is that the peak amplitude of signal output must not be allowed to approach the supply voltage limits, because the internal biasing is no longer effective if this is done. This restriction can be quite irksome if the op-amp is to be used with low voltage single-ended supplies, and an alternative for such applications is the current difference amplifier (CDA), of which the best known example is the National Semiconductor LM3900N. This chip is an operational *current* amplifier whose internal circuitry, though remarkably similar to that of the 741, allows operation at output voltage levels very close to either ofthe supply voltages.

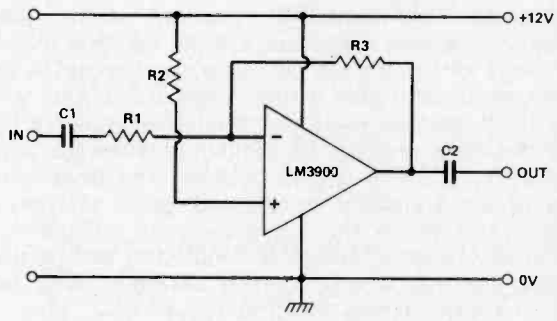


Fig. 7 A typical LM3900 circuit.

The design principles for CDAs are very different from those used in the 741. The output voltage depends on the difference between the *currents* at the two inputs, and the circuits that use these chips are distinguished by large resistor values. In the circuit of Fig. 7, for example, if we aim for an output voltage which is half of the supply voltage, then, remembering that the current through R3 must be the same as the current through R2, the value of R3 must be half of the value of R2. Since the input currents are very low, these resistor values have to be high, and values of several megohms are common. The voltage gain in the circuit shown is R3/R1, as for the 741 type of amplifier, but the voltage swing at the output can reach very close to the supply voltage limits. Current difference amplifiers are used mainly in circuits which operate at the lower ranges of frequency because of the effects of stray capacitances on the very large value bias resistors.

CONFIGURATIONS 7

This time we turn out attention to some of the many circuits that can be used to make waves: could this be a sine of the times?

When you first start to take an interest in circuit configurations, one of the first things that strikes you is the huge variety of sine wave oscillators, many of them known by names that go right back into the mists of time. When you look at these circuits more closely, however, what strikes you is not how different they are but how similar — and that's our starting point for this month.

An oscillator consists of an amplifier with a positive feedback loop and some circuit which has a time constant or is resonant to some frequency. Using this definition, we can include multivibrators among our oscillators, and rightly so, but since we dealt with multivibrators in Configurations Part 4, we'll confine ourselves to sine wave oscillators in this part.

The Shrinking Sine

At times, the amplifier portion seems almost superfluous, because a resonant circuit, which is the most familiar way of forcing an oscillator to operate at some fixed frequency and give a sine wave, is a circuit which will, by itself, oscillate quite happily! The circuit of Fig. 1 will, for example, produce an oscillation when the base of the transistor is briefly pulsed positive. The peak emitter voltage of the transistor during this pulse charges the capacitor, and when the transistor cuts off again, the capacitor discharges through the inductor, setting up an oscillating current which in turn causes a sine wave voltage to appear across the circuit.

This wave decays, however, as Fig. 2 shows, because the coil has resistance and any resistance in a circuit will dissipate energy (as heat) when a current flows through it. That's why an amplifier is needed — just to replace the energy that is lost in the resistance of the coil! Sine wave oscillators that make use of LC resonant circuits do not need a high-gain amplifier, and too much gain will, in fact, distort the waveform, changing it from a pure sine into something more like a square wave. Any oscillator that makes use of the LC resonant circuit must therefore include some means of controlling the gain of the amplifier, because a really well-shaped sine wave will be obtained only when the amplifier gain is just enough to sustain oscillation, and no more. There's another reason; a circuit which includes positive feedback is never very stable, so that the oscillator must include some method of limiting — preventing the amplitude of the oscillation from growing until the tips of the wave start to square off. That's one thing which can be done most easily when the gain of the amplifier is low, because the gain will drop as the transistor approaches the cut-off or the bottomed conditions, and if the gain is low to start with, this drop should be enough to limit the amplitude of oscillation with only a small amount of distortion of the waveform.

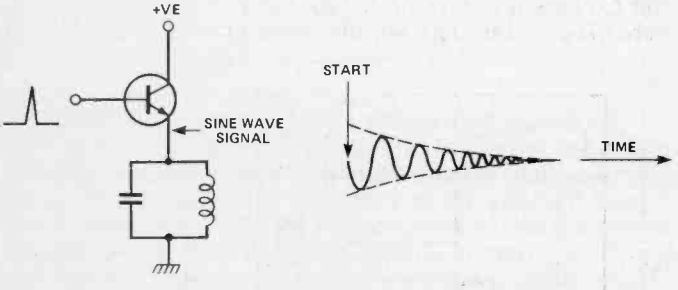


Fig. 1 (Left)A ringing circuit, using a resonant circuit in the emitter of a transistor which is normally cut off, but which can be pulsed briefly into conduction.

Fig. 2 (Right) The form of the 'ringing' wave — this is a sine wave which decays to zero amplitude. If the circuit resistance is very low, the decay may take a 'long' time, meaning that many cycles of wave will be executed before the amplitude becomes zero.

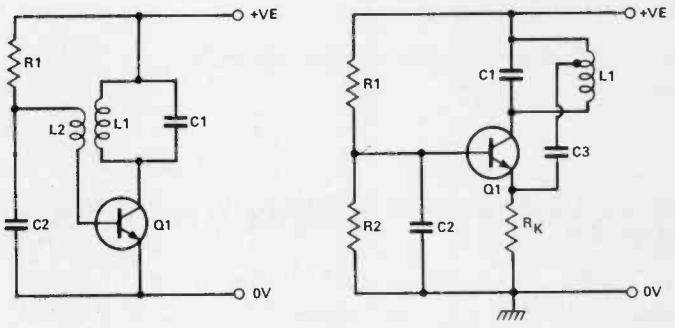


Fig. 3 (Left) A simple two-winding oscillator. This is easy to construct, but not so easy to adjust for a pure sine wave.

Fig. 4 (Right) The Hartley oscillator in one of its many forms. The positive feedback loop is from the collector circuit to the emitter.

Winding You Up

With these words on general principles out of the way, then, we can take a look at some oscillator configurations. Let's start with the simplest one — the two-winding transformer type as shown in Fig. 3. The three building-blocks of amplifier, feedback loop and resonant circuit are obvious, but it's by no means the easiest type to obtain a pure sine wave from. The reason is that the bias of the transistor has to be set by the value of R1, and the gain must be set by the design of L2 — a few turns spaced some distance from L1. There's no quick and easy way of

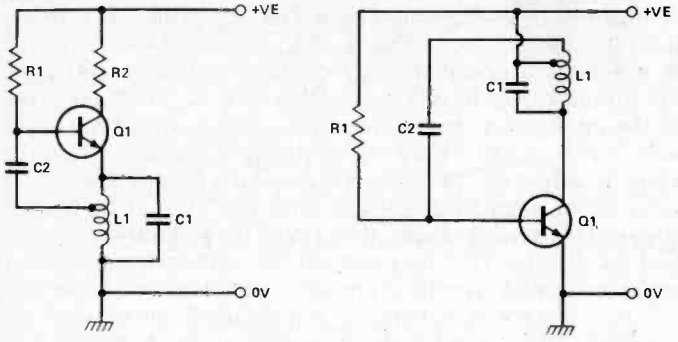


Fig. 5 (Left) Another form of the Hartley oscillator, with feedback from the emitter to the base.

Fig. 6 (Right) A third form of the Hartley circuit, using the coil tapping to invert the signal.

calculating the number of turns and spacing of L2, so that design invariably ends up with cut-and-try methods. The usual technique is to start with too many turns, get the circuit oscillating (which may mean reversing the connections to L2 if you got them the wrong way round), then removing turns for as long as the circuit will continue to oscillate, and finally, restoring a turn or a half-turn. This has to be done to ensure that the circuit will start each time it is switched on.

Much better waveshapes can be obtained with less effort by using the traditional Hartley and Colpitts oscillators that are so beloved of radio hams. One version of the Hartley oscillator is shown in Fig. 4 — this uses feedback from a tapping on the coil to the emitter terminal. This oscillator can give well-shaped sine waves,

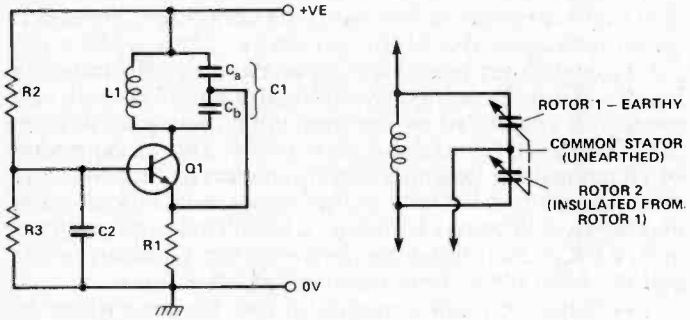


Fig. 7 (Left) The basic Colpitts oscillator.

Fig. 8 (Right) A variable tuning Colpitts can make use of a variable two-gang capacitor of specialised construction.

and seldom needs adjustment. The rule of thumb is to use a coil tapped at about 10% to 20% of its total turns from the 'cold' end (the end connected to the supply), a resistance of around 330R for R_k, and design the bias components R1 and R2 so that the oscillation is a sine wave and is self-starting. The decoupling of the base is essential if the oscillator is to be used as shown; alternatively the base can be driven from a low-impedance source which will cause the output from the resonant circuit to be amplitude modulated.

Figure 5 shows another version of the circuit, in which the tapped coil is in the emitter circuit, feeding back to the base this time, and so leaving the collector free to deliver the waveform. The output wave at the collector is not a pure sine wave, however, so that this output is useful mainly when the output is to be squared to generate harmonics, or if a resonant circuit is to be included in the collector circuit. A wave of better sine shape can be taken from the emitter. Figure 6 shows another variant of the Hartley circuit which uses the tapped coil in the collector circuit — in this example, the tapping is connected to the supply voltage so that the remainder of the coil phase-inverts the signal to feed the base.

The Colpitts oscillator uses a very similar circuit to the Hartley, but with a single coil winding, untapped. The tapping is arranged by using two capacitors connected across the coil, as shown in Fig. 7, which shows the tuned-collector version of the circuit. The combination of the capacitors in series is the tuning capacitance for the inductor, and the ratio of the values should be arranged so that C_a is around $5C_b$ (or more) to give the required signal

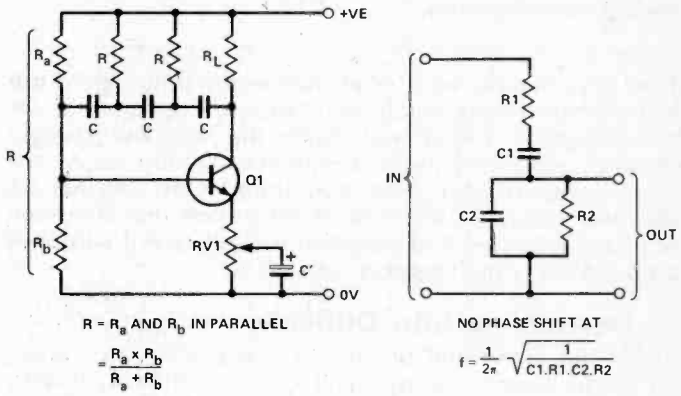


Fig. 9 (Left) The phase-shift oscillator in its simplest form.

Fig. 10 (Right) The Wien bridge circuit, originally devised for measurement purposes.

division ratio. The Colpitts configuration has the advantage that only a two-terminal coil is needed, but the capacitor dividing chain is a nuisance, particularly if frequency has to be varied by using variable capacitors — one solution is the type of twin ganged capacitor shown in Fig. 8.

Since it's possible to make RF oscillators from every conceivable arrangement of amplifier, resonant circuit and positive feedback loop, there are dozens of RF sine wave oscillator circuits, some of them (like the Hartley and the Colpitts designs) stretching back to the 20s and 30s. What keeps the most popular ones alive is that they give

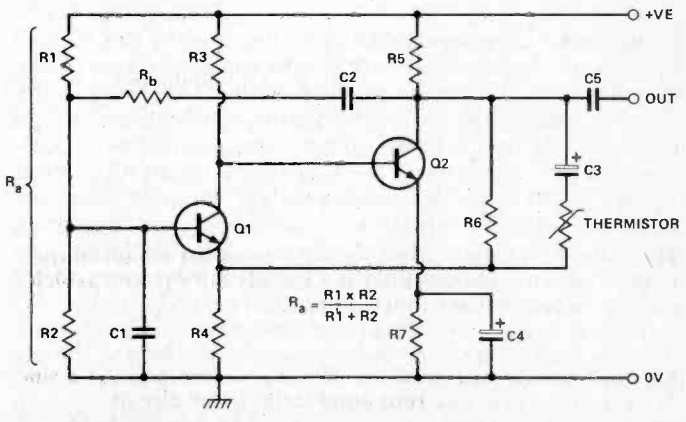


Fig. 11 A Wien bridge circuit circuit, using a thermistor to stabilise amplitude.

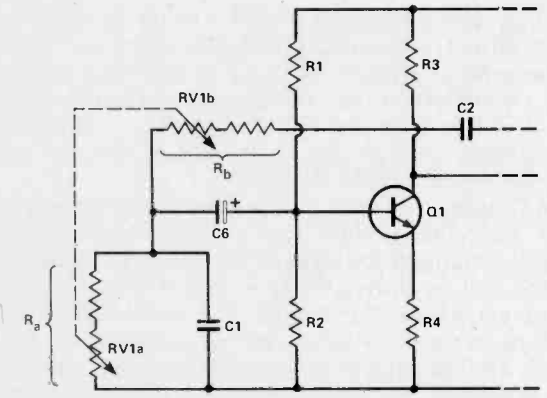


Fig. 12 Frequency variation on a Wien oscillator using ganged potentiometers.

good sine waves, with small frequency drift. There are many other designs which have simply dropped out of use because they could not fulfil the ever-increasingly stringent requirements for frequency stability, so please don't write in with what you think is an original RF oscillator circuit — it's a thousand to one that someone will have patented it in nineteen oatcake and it will have dropped out of use for good reason!

Descending Into Difficulty

At the lower end of the frequency scale, sine wave oscillators which use resonant circuits start to run into component problems. The inductors need iron cores, causing non-linearity, and have very low Q figures (ratio of reactance to resistance), which also permits poorly-shaped

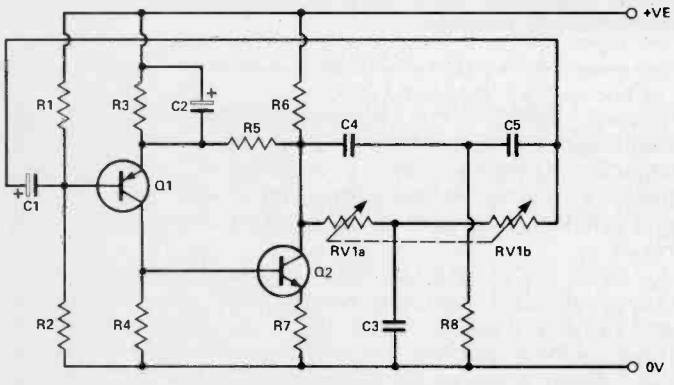


Fig. 13 The twin-T network used in an oscillator circuit.

waveforms. Capacitors tend to be more leaky because of their large capacitance values, and the sheer size of the resonance circuit can be very unwieldy. This leads to the use of RC oscillators for low frequency applications, and to the design of much more complicated oscillator circuits. The problem is that there are no truly resonant RC circuits that are in any way comparable with LC circuits. There are many RC circuits which have minimum or maximum attenuation or phase shift at a 'resonant' frequency, but their selectivity, in terms of the change of frequency is very poor compared to the LC circuit. For this reason, RC oscillators will never give a pure sine wave unless the gain of the amplifier stage is very closely controlled — hence the circuit complications.

The 'classic' RC oscillator circuit is the phase-shift oscillator of Fig. 9. The three RC time constants should be approximately equal, and each should cause a 60° phase shift at the desired oscillating frequency. The set of three then causes a 180° phase shift overall, which is the requirement for oscillation if the gain is sufficient overall. A fair amount of gain will be needed in this circuit because of the attenuation of the three RC circuits, and transistors with low h_{fe} values may not oscillate in this circuit at all. Gain is adjusted by using RV1, which varies the small amount of negative feedback, and this should be set for the best sine wave shape, which will be when the circuit is just oscillating. This may not be the setting for obtaining reliable starting when the circuit is switched on, however.

For purposes where a high-quality sine wave is needed, and particularly if variation of frequency is wanted, more elaborate circuits are used, of which the Wien Bridge and the Twin-T are typical. The Wien bridge circuit itself is shown in Fig. 10 — its peculiar property is that is has zero phase shift at its 'resonant' frequency as shown in the formula. This configuration can be used in a positive feedback loop to ensure that the feedback is in the correct phase only at the correct frequency, but close control of the gain will be needed if the waveform is to remain of good sine shape.

Getting A Bead On It

The conventional way of achieving this on modern RC oscillators is by the use of a subminiature thermistor with negative temperature coefficient. A bead thermistor is used, which will be heated by a signal current of a milliamp or less, and this component is included in a negative feedback loop which controls the gain of the complete amplifier section. When the signal current flowing through the thermistor is high, the thermistor resistance decreases and the increased feedback causes the gain of the amplifier to be reduced. The opposite process occurs if the signal current is too small. The thermistor therefore takes over the task of controlling the gain of the amplifier, allowing us to concentrate our efforts on designing the rest of the circuit.

A typical Wien bridge oscillator circuit is illustrated in Fig. 11. A direct-coupled two-transistor circuit is used, and the gain is controlled by the loop which feeds signal from the collector of Q2 to the emitter of Q1. This loop consists of a fixed resistor R6 and the combination of capacitor and thermistor in parallel with it; the values must be chosen so that the overall gain is fairly low. The Wien bridge network of $R_a/C1/R_b/C2$ is connected between the collector of Q2 and the base of Q1 (thje positive feedback loop).

Variable frequency operation can be carried out by varying either the resistors of the Wien network or the capacitors, but it is not enough to alter just one component value. Most amateur circuits (see Fig. 12) use small value fixed resistors, plus a ganged potentiometer in series, to make up R_a and R_b: R1 and R2 have high values, so their effect is negligible. Commercial Wien bridge circuits tend to use a FET for Q1, and to carry out frequency variation by using a ganged capacitor for C1 and C2. This gives a larger sweep of frequency, so that fewer ranges are needed, but requires very large resistor values, since the variable capacitors are only 500pF in value. Resistors of many megohms are needed if low frequencies are to be generated, hence the need for the FET at the front end. Commercial Wien bridge circuits can cope with a frequency range of 10 Hz to over 1 MHz in three or four switched ranges.

Finally, Fig. 13 shows a typical circuit using a twin-T network but omitting the complications of the thermistor amplitude control. The twin-T has, for some reason, never been so widely used in this country as in the USA — it seems to be the old Bootstrap v Miller timebase attitude all over again!

CONFIGURATIONS 8

The editor expressly forbids his neighbours from reading this Configurations, because it's all about audio power output stages. Ian Sinclair shows the way to deafness . . . oh for the quiet life.

A lot of people who feel quite happy with the design of voltage amplifier stages are never quite so confident with power output stages. The reasons are not difficult to find, because few textbooks go into much detail about transistor power output stages, and one or two offer rather misleading advice.

The essential problem of power output is to get power delivered into a load, and a theorem which is often quoted in this respect is the maximum power theorem — Fig. 1. This states that if the power source has fixed values of internal resistance and supply voltage, then the maximum transfer of power will occur when the load has the same value of resistance. The maximum power in the load will then be 50% of the total power, with the other 50% being dissipated across the internal resistance. The use of this theorem governed the design of valve output stages for decades.

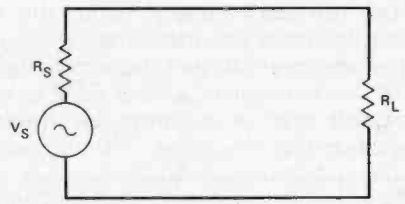


Fig. 1 Maximum-power theorem. This states that the maximum *possible* power is transferred to the load when R_L = R_S. This does not imply that the load gets the maximum *share* of power.

Things have changed, however, and we can easily obtain transistors with very low internal resistance values and use them in circuits whose internal resistance can be as little as a fraction of an ohm. The maximum power theorem isn't useful here, because we don't actually want maximum power, only as much as we can handle, and preferably most of it across the loudspeaker terminals. In any case, we don't want to have to use loudspeakers with impedances of only a fraction of an ohm. When we use a loudspeaker whose impedance is several times that of the amplifier, we can be sure that most of the power dissipated is across the loudspeaker rather than across the output transistors, and that's just what we want, whether it's the maximum possible power or not.

The very low output resistance of transistor power amplifiers also explains why it is that the power output of a transistor amplifier increases when you use a loudspeaker of lower impedance — your average pre-war textbook can't explain that one! If we were to attempt to use a load that matched the resistance of the amplifier output stage, it's pretty certain we would burn out the transistors.

Some Like It Hot

Speaking of which brings us to the second point about power output stages. With low internal resistance, there is no problem about delivering power, but the performance of a power amplifier in this respect is limited by the rate at which the transistors in the output stage can dissipate the heat which is inevitably caused when current flows through them. As long as the rate of heat generation (which is volts × amps) equals the rate of heat dissipation, the temperature will remain steady, but raising the dissipation also means raising the temperature, and this is the limiting factor for transistors, because if the collector junction, which is where the heat is generated, gets too hot, it will melt, and that's the end of the transistor. Many output transistors which could dissipate 150 W if the heat could be removed efficiently will dissipate only a miserable 20 W under realistic working conditions.

The design of a transistor power amplifier, therefore, starts with a consideration of heatsinks. The traditional method is to use a quantity called thermal resistance, which is defined as the temperature rise per degree (centigrade) of dissipated power. When a heatsink has a thermal resistance of 4°C/W, then it will be 4°C hotter than the air around it when it is dissipating 1 W, 40°C hotter than the air when it is dissipating 10 W, and so on — the temperature rise equals thermal resistance times power dissipated. For a transistor bolted on to a heatsink, there are several thermal resistances in series (Fig. 2) — the thermal resistance of the collector junction to the mounting surface of the transistor, the thermal resistance of the mounting surface to the heatsink (which will be increased if you use a mica washer for insulation), and the thermal resistance of the heatsink itself to the air. Like electrical resistances, these can be added (that's why we use them!), and when the result is multiplied by the intended power dissipation, the result will tell you how much hotter than the air your transistor junction will be. Remember that the air actually around the area of the heatsink may not be all that cool — a conservative figure to use is 40°C — and then add on the rise in temperature that you have calculated. If the result is well short of the maximum

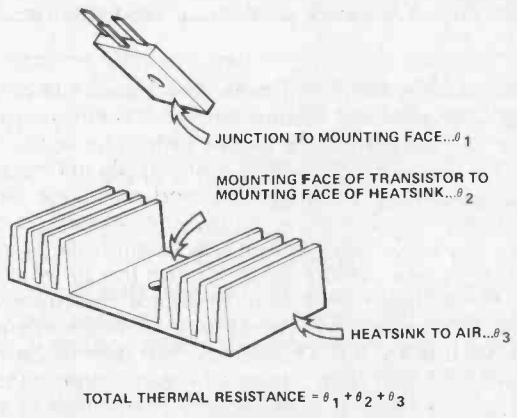


Fig. 2 Thermal resistances. For a single transistor on a heatsink, the thermal resistances are in series, and can be added. For other configurations, these quantities can be treated exactly like electrical resistances.

allowable figure for the transistor, well and good; if not, then you need to improve your heatsinking, or use a different transistor, or both.

Making A Transformation

With that out of the way, we can now look at some configurations. Most of us automatically think of the PNP-NPN direct coupled pair when we think of output stages, but there are still a lot of single-ended transformer-coupled stages around, similar to the design of Fig. 3. A Class A stage like this is designed by finding the maximum dissipa-

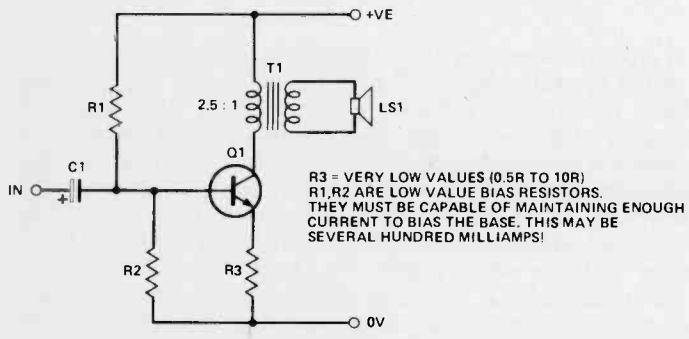


Fig. 3 A single-ended Class A stage, using transformer coupling.

tion you can get away with, and then fixing the supply voltage and calculating the signal current. If we take it that the average DC level at the collector of the transistor is equal to the supply voltage (Fig. 4), then at peak power output, the signal voltage (instantaneous voltage, that is)

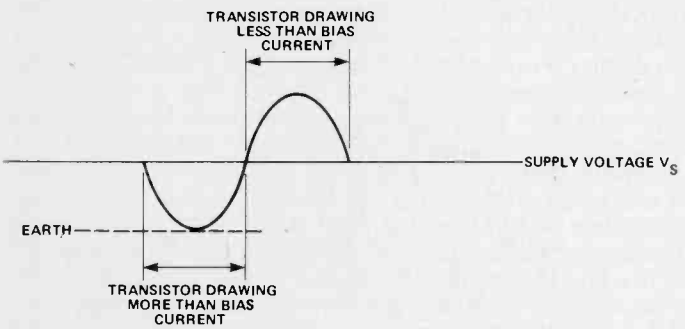


Fig. 4 The (ideal) output waveform at the collector of the circuit of Fig. 3. The inductance of the transformer is responsible for the portion of the wave which is above supply voltage.

will drop to zero and rise to twice supply voltage. This makes the peak voltage of the output signal equal to the supply voltage, and the RMS power is equal to

$$\frac{\text{peak volts x peak current}}{8}$$

so that peak current, I_p, equals

$$\frac{8 \times \text{power}}{\text{supply power}}$$

— you will have to check for yourself that the transistor can cope with this peak current. The next step is to calculate the transformer ratio. The peak voltage V_p across the loudspeaker will be

$$\sqrt{\frac{\text{power}}{8}}$$

and the transformer ratio will have to be

$$\frac{\text{supply voltage}}{V_p}$$

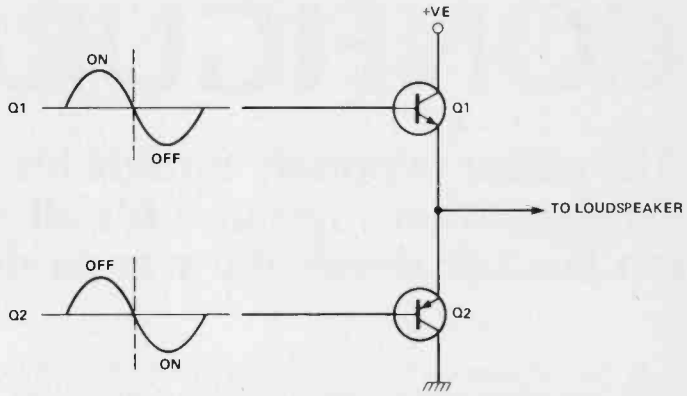


Fig. 5 The basis of the single-ended push-pull Class B stage, most succinctly known as the totem-pole output.

— this is usually a step-down ratio. The transformer should have enough primary inductance to ensure that it will handle low frequencies reasonably well, but you don't want to design the transformer in detail unless you are a card-carrying masochist.

The single-ended stage is not exactly brilliant from the point of view of distortion, and the voltage gain is usually very low, so that negative feedback from the speaker terminals to the input of the power stage is rather ineffective. The negative feedback can be taken to an earlier stage, but the drawback here is that the phase shifts may be excessive, particularly since a transformer is present, and these can make your negative feedback become positive at one end of the frequency range, causing distinctly nasty sounds to come from the speaker. The main merit of a single-ended transformer-coupled stage of this type is that it can deliver a fair amount of power from a low supply voltage, something that is not easy for the traditional direct-coupled design.

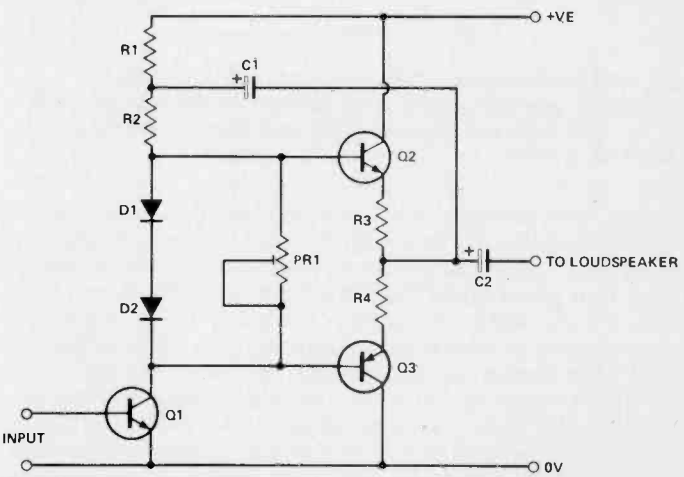


Fig. 6 Using two diodes in series to stabilise the bias of the output pair.

The Traditional Transistors

With that brief introduction — meet the traditional direct-coupled output stage as used in practically all of the hi-fi, medium-fi and no-fi amplifiers in the world. The design consists of a pair of complementary emitter-followers in a Class B single-ended push-pull circuit (Fig. 5), with both transistors on one heatsink, capacitor coupling to the loudspeaker, and lots of negative feedback. It's a design on which an incredible number of variations can be achieved, however, and also one whose performance can be greatly enhanced by careful choice of components,

and well-planned construction. The driver stage for the output pair may use diodes to adjust the DC voltage difference that is needed between the bases of the output pair (Fig. 6) or an almost-saturated transistors (Fig. 7) or with a common-emitter pair used in place of emitter-followers, and driven by an op-amp (Fig. 8).

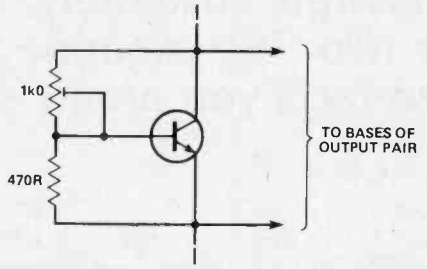


Fig. 7 An alternative method of stabilisation using a transistor.

Like all Class B stages, linearity, especially near the crossover point (Fig. 9) where one transistor starts to conduct as the other one stops, is always a problem. The gain in this region is very low, so that negative feedback is not the cure-all that many designers seem to expect it to be. Since the basic output stage is a couple of emitter-followers, it has a voltage gain that is less than unity, and the driver stage usually has a low gain also, so that the feedback loop has to be taken over a number of stages (Fig. 10).

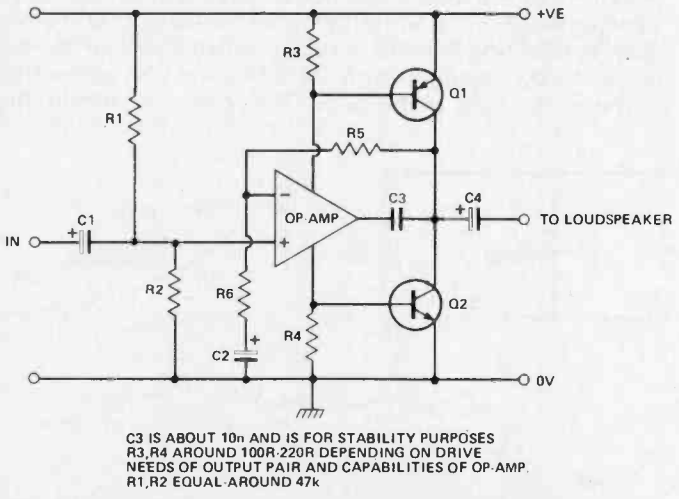


Fig. 8 A circuit which uses an op-amp to drive an output pair. The transistors are NOT in the normal totem-pole configuration, because each is being used as a common-emitter amplifier rather than as an emitter-follower. If Darlington power output transistors are used, this can be a very economical high-power stage.

The problem of crossover distortion has driven several designers to use Class A stages of very similar configuration. When both of the transistors of the output pair are driven with a signal, the efficiency can be as high as 30% (as compared with 78% for Class B), and the availability of high-power transistors with low thermal resistances has encouraged the use of Class A — a typical circuit is shown in Fig. 11. The distortion figure, measured before applying feedback, is still fairly high (10% or more at full power), but feedback greatly improves this. More important, the distortion level tends to be least at low power outputs, unlike the Class B circuit in which the distortion is greatest at low level — when it is also most noticeable — due to the crossover problem.

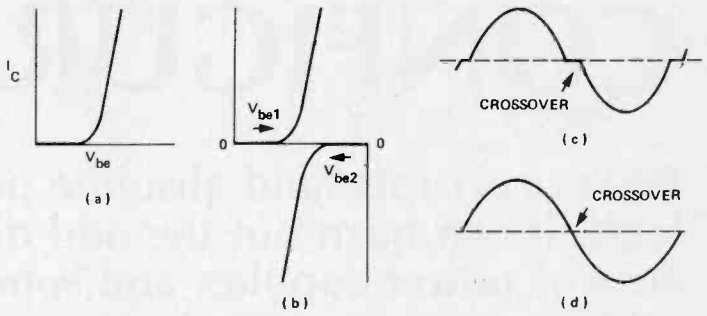


Fig. 9 Crossover distortion. (a) an ideal mutual characteristic for a power transistor. (b) How the characteristics of two identical transistors would combine if each were just cut-off with no signal present. (c) The distortion of wave-shape caused at crossover with insufficient bias. (d) Crossover distortion can be reduced by increasing bias on each transistor, but unless the transistors have unusually straight characteristics, a lot of bias will be needed.

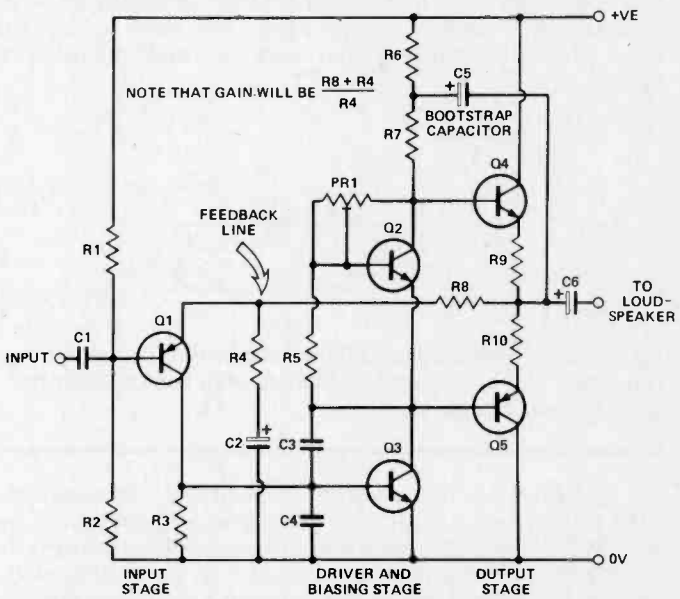


Fig. 10 Using feedback over several stages in a Class B output circuit.

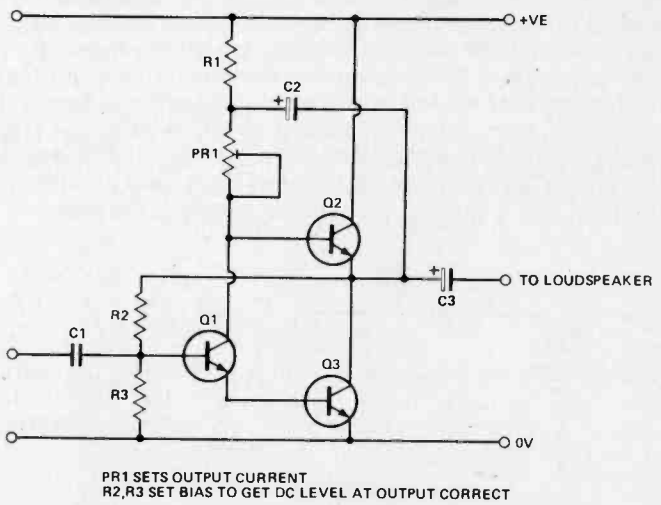


Fig. 11 A Class A output stage — note that both output transistors are of the same type. For low distortion, they should be carefully matched.

CONFIGURATIONS 9

Power corrupts, and absolute power corrupts absolutely. At least, it can burn out the odd diode or two. We examine the area of power supplies and some of the facts you aren't often told.

Power packs, you might think, are among the simpler of electronic circuits to design, and yet there is probably more cut-and-try used in the power supply section of a circuit than in all the rest of the circuitry that you construct. The reason seems to be a lack of coherent explanations of the action of the reservoir capacitor — only too often you are simply told that it "provides an earth route for AC ripple", and no more. We have to start this time, then, by putting that sort of misconception to rights.

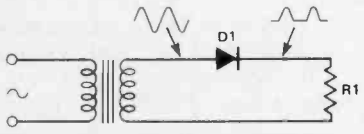


Fig. 1 Simple half-wave rectifier circuit with no reservoir capacitor. The waveform is unidirectional, but certainly not what we would call DC.

Consider for the sake of simplicity, a half-wave rectifier circuit and a load (Fig. 1). The waveform across the load will consist of about half of the input waveform, the positive half in this example because of the way we have chosen to connect the diode — reverse the diode and you will select the negative half of the wave. This type of output is called a *unidirectional* wave — the peaks are in one direction (positive) only, with no negative peaks — but it isn't exactly anyone's idea of DC. A DC voltmeter connected to the load of this circuit reads what DC voltmeters always read, the average voltage, which is around E_o/π; approximately 0.32 E_o, assuming that the diode is 'perfect' in the sense of having no forward voltage drop across it. We can allow for the forward drop, which can't be neglected if the output voltage is low, by subtracting its value from E_o, the peak AC input. This is only an approximation, but it is good enough for practical purposes.

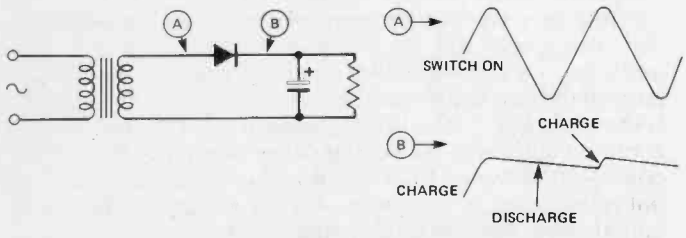


Fig. 2 A half-wave circuit with a reservoir capacitor added. The capacitor charges to the peak voltage of the input wave, and the charged capacitor supplies the load while the diode is reverse-biased.

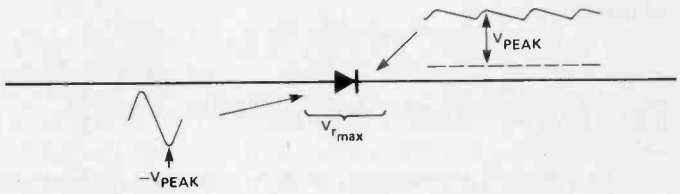


Fig. 3 This shows why the peak reverse voltage on the diode is doubled when a reservoir capacitor is used.

Bring On The Reserves

Now when a reservoir capacitor is connected to the circuit (Fig. 2), things change considerably. To start with, imagine that the load resistance is very high, so that only a small amount of current is being taken. Instead of the rectifier conducting for the whole positive cycle of the AC wave, it now conducts only for a tiny fraction of the time of the wave, right at the peak. The reason is that the first

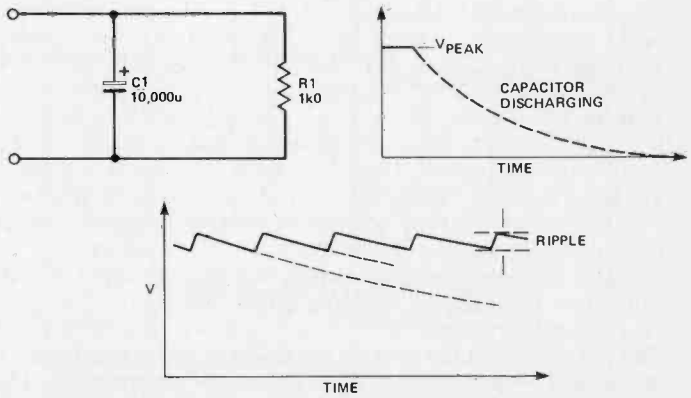


Fig. 4 The waveform of ripple, caused by the time constant of the reservoir capacitor and load resistance.

half-cycle, when the supply is switched on, will charge the reservoir capacitor to the peak positive value of the AC wave, less the forward diode drop, and when the AC input at the anode of the diode drops below this value, the diode will cut off. From this moment until the next positive peak of the wave comes along, all the current that is supplied to the load is supplied from the reservoir capacitor, which is why it's called a reservoir! Far from just being a bypass for AC, the reservoir is the main store and supplier of DC to the load.

All the current that dribbles out from the capacitor results in the voltage across the capacitor dropping as its charge is drained, so that the diode has to supply this

charge again next time it conducts. You don't get something for nothing — the diode passes large currents for short time intervals instead of conducting steadily over a half-cycle as it did when no reservoir was used. The overall result is that the diode has to be able to pass peak currents that are many times greater than the average current, it spends most of its time cut off, the maximum reverse voltage across the diode is twice the AC peak voltage (see Fig. 3), and there is a 'ripple' on the output wave which is caused by the drop in voltage as the reservoir capacitory discharges (Fig. 4). The waveform of this ripple is a sawtooth, rich in harmonics, not simply a piece of left-over sine wave as some explanations would hint at, so that it is a potent source of hum interference in the rest of the circuit.

The approximate amplitude (peak to peak) of the ripple is given by It/C, where I is the average current drawn by the load, C is the size of reservoir capacitor, and t is the time between positive wavepeaks. Using units of milliamps for I, microfarads for C and milliseconds for t, we get units of volts for the amplitude of ripple. For example, if you draw 100 mA from a 1000uF capacitor with a half-wave rectifier for which t is about 20 mS, then the

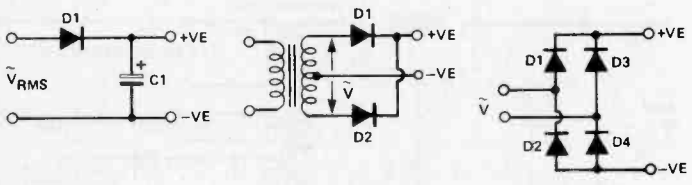


Fig. 5 A summary of the conditions for common power supply configurations.

ripple amplitude is (100 × 20)/1000, or 2 V, which isn't exactly negligible. Using a full-wave rectifier, which recharges the capacitor at 10 mS intervals, you get a 1 V ripple. This formula *isn't* foolproof — it applies only when you have the situation in hand, and will give silly answers if the reservoir capacitor is much too small or if the amplitude of the AC input is very small, but it's a good guide to realistic values for power supplies generally.

The voltage output of the circuit with no load current is equal to the AC peak voltage, but as the load current increases, the ripple also increases and the average DC output drops until it can become almost as low as the value you would get with no reservoir, 0.32 E_o for half-wave, and twice as much as for full-wave (bridge or split-secondary type of circuit). Figure 5 summarises the operating conditions for different rectifier configurations. Ripple, and the drop of output voltage when output load

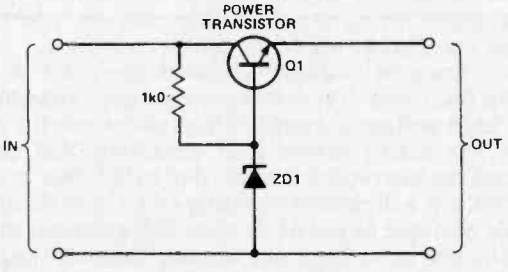


Fig. 6 An elementary stabiliser — the power transistor in this example would be a medium-power type with a high value of h_{fe}.

current is taken, can be minimised by increasing the size of the reservoir capacitor. Obviously, it is also an advantage to have a short time between recharging the reservoir, so that high-frequency supplies need less in the way of reservoir capacitance — one of the many reasons for the popularity of switch-mode power supplies these days.

A Stable Situation

Another defensive measure is stabilisation. Stabilisation does not mean that some circuit is used which will miraculously bump up the voltage output from the reservoir capacitor, it simply means making the best of what you have. Suppose you have a nominal 8 V supply, and that at the full planned output current of 150 mA it can have a 2 V peak-to-peak ripple. This value implies that the voltage will drop momentarily as low as 6 V twice on each AC cycle, assuming that full-wave rectification is used, so that if we use only 5 V of this supply, these changes caused by ripple will not affect the 5 V output at all. This is the action of a stabiliser — it's a circuit which is a voltage-dropper, but arranged so that the drop is variable, keeping the output voltage constant while the input voltage varies.

A stabiliser has to operate so as to fulfil two requirements. First it must keep its output voltage constant as the input voltage varies, and second, it must keep the output voltage constant as the load current varies. The two may sound identical at first glance, but they are not — the first calls for the output to be constant while the voltage across the stabiliser is varying, the second calls for the combination of the stabiliser and the rest of the power pack to have almost zero internal resistance.

Figure 6 shows a very basic form of stabiliser. The voltage at the output is set by the value of the zener diode, and because of the voltage across the base-emitter of a transistor, the output voltage will be around 0V6 less than the zener diode voltage. This should ensure that the voltage of the output is stabilised against changes at the input, but the stabilisation against changes in the current

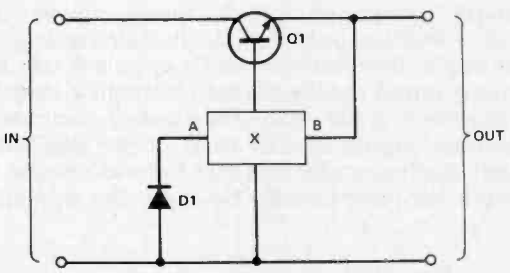


Fig. 7 A block diagram of the comparator type of power supply stabiliser. This type is rarely built nowadays because of the ready availability of IC equivalents.

taken by the load is not so good, because the V_{be} must increase to some extent as the load current increases. Nevertheless the stabilisation is better than it would be in the absence of the circuit (something wrong if it were not!), and can be improved by amplifying the signal to the base of the regulator transistor — a variation on the circuit is shown in Fig. 7. The output voltage is compared with the zener voltage, and the output of the comparator is used to control the base of the regulator transistor. Very low output resistances of the order of a few milliohms can be obtained using circuits of this type.

I've drawn the circuit as a block diagram because it isn't very often nowadays that we have to build stabilisers with separate components. The reason, of course, is the ready availability of IC regulators, particularly the 78

series. These take advantage of being ICs (so that circuit complications are not a problem for production, only for design) to incorporate features such as current foldback, meaning that the current will be regulated if there is any risk of over-dissipation. This ought to prevent overload and give these regulators a very long life — I say ought, because in my experience these regulators quite frequently fail, and I suspect that the fold-back arrangements are not always completely effective.

The 78 series covers most of the 'popular' supply voltages, but if we should want an odd value then a modification to the circuitry, as shown in Fig. 8, can do the

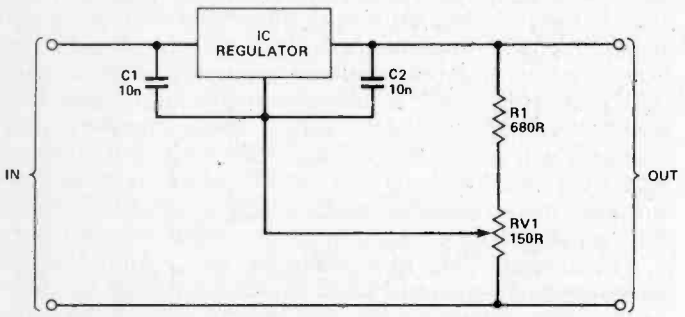


Fig. 8 Varying the output voltage of an IC stabiliser. A variable resistor is illustrated, but a fixed value resistor could be used once the correct value has been established.

needful, at the expense of a slight loss in stabilisation. Similarly, if we want a lot more current from the output than the normal 78 series can supply, then we can use the IC to control an external transistor, as shown in Fig. 9. Circuits like these can cope with about 99 per cent of our needs.

Switching The Subject

Having mentioned switch mode power supplies, however, I feel I should explain further because, unless you follow the development of TV circuitry, you may not have come across details of them (though a switch mode supply was used in the venerable Apple 2 computer, and a switch mode supply is now used in the BBC computer after early users complained that the old version burned the varnish off their tables). Basically the principle is to

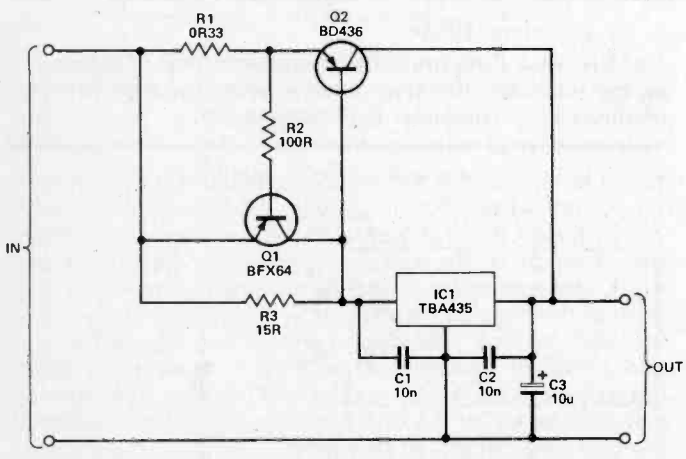


Fig. 9 Increasing the current-handling capability of an IC stabiliser. The stabiliser handles the rated current, and any amount beyond this value is handled by the auxiliary transistor circuit, preserving voltage stability.

dispense with a mains transformer, and rectify the mains voltage so as to produce a high voltage DC. By dispensing with the resistance of a mains transformer, and by using a reservoir capacitor of surprisingly modest capacitance (but rated for 500 V!), this supply voltage can be quite stable. It is then applied to a switching circuit which charges a capacitor several thousand times per second and discharges it just as frequently into the primary of a transformer which, because it operates with high-frequency signals, can be small and well-insulated. The outputs of this transformer are rectified, and need only small reservoir capacitances because of the high frequency that is used. There is no need for a stabiliser of the old-fashioned wasteful type either, because the output voltage can be sampled by a comparator, and the output of the comparator used to alter the switching times. The idea is that if the output voltage drops, the switch can spend more time passing current into the primary of the transformer; if the output voltage is too high, the switching circuits cut off earlier. There is no waste involved — what is not used is held in the reservoir capacitor ready for the next switching operation.

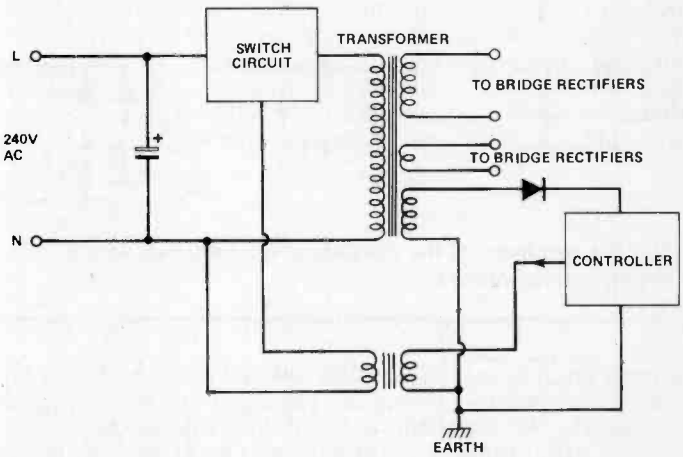


Fig. 10 An outline of a switched mode power supply. No values are shown, because the transformer is a critical component and the other circuitry can be obtained in IC form.

The main advantage is that the supply runs astonishingly cool, with no huge heatsinks needed for the regulator. The advantages for TVs and computers are obvious — I remember one computer which left scorch marks and which could have served as a sandwich toaster. Another advantage is that no AC voltage adjuster is needed — whatever the mains voltage happens to be will be compensated for by the switching process, and there are ICs which will take care of the whole operation.

One point of caution concerns servicing. If you are working on a switch mode power supply, remember that it uses high voltages, and that part of the circuit is always live to the mains when it is operating. On many TV receivers, in accordance with the belief that a designer worth his salt will make the inside of a TV as dangerous as possible in order to kill off amateur mechanics, the whole chassis is live or at least not isolated from the mains. The growing trend to make TVs in monitor form so that they can be connected directly to video recorders instead of by the ridiculous method of re-modulating the signal may at last bring us electrically into line with the rest of the world in this respect.

CONFIGURATIONS 10

And so to solid state switches. In this Configurations Ian Sinclair looks at the basic techniques involving the thyristor and its close relatives.

As a component, the thyristor is so closely related to the diode that thyristor circuits just had to follow the treatment of power supplies last month. Technically, the thyristor is a four-layer diode, but as far as we are concerned, it's a silicon diode that is switched into conduction by a signal at a third electrode, the **gate**, as shown in Fig. 1. In many respects, however, the action is very much that of a normal silicon diode; for example, it will not conduct in the reverse direction (cathode positive), and it has about 0V6 forward drop across the anode-cathode terminal when it conducts. The distinguishing feature is that the start of forward conduction only occurs when a **trigger** pulse arrives at the gate and **fires** the thyristor. Whatever you subsequently do to the gate, the thyristor will continue to conduct until the forward current falls below a value known as the **holding current,** at which point the thyristor will turn off. However, while the thyristor is on, it is as fully conducting as a silicon diode would be.

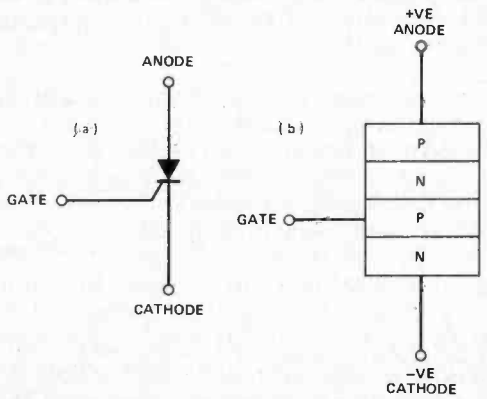


Fig. 1 The thyristor: (a) circuit symbol, (b) arrangement of semiconductor layers.

Triggers Fingered

One point that is not always sufficiently understood is that the triggering requirements can vary enormously from one type of thyristor to another. A lot of small thyristors will trigger for a gate current of only a fraction of a microamp, so that interference signals will trigger the thyristor if the gate terminal is not 'earthed' to the cathode by a low-value resistor. A lot of false triggering of burglar alarms seems to be due to thyristor circuits in which the gate has too high a resistance to the cathode, making the gate circuit a very efficient aerial for any radiated energy! Even when quite low resistance values are used, thyristors can trigger in lightning storms or because of static discharges, so that some careful design of the gate circuit and extensive testing is needed if you are in the alarm business. The combination of low resistance and a suppressor ferrite bead placed at the gate terminal helps a lot! Large thyristors need rather more in the way of gate current, but even these can be triggered by a fraction of a milliamp.

Thyristors are most at home in circuits which use DC or unsmoothed (but rectified) AC. The use of rectified AC is particularly popular (Fig. 2) because the thyristor will

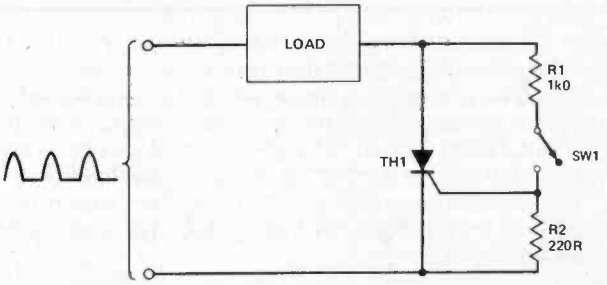


Fig. 2 Elementary switching circuit for use with rectified AC. When the switch is on, current will flow through the load.

switch off each time the supply voltage reaches zero, and all that we need to concentrate our attention on is the triggering which switches it on again. Where a thyristor is used in a DC circuit, there is the extra complication of reducing the voltage across the thyristor to zero in order to switch it off (Fig. 3).

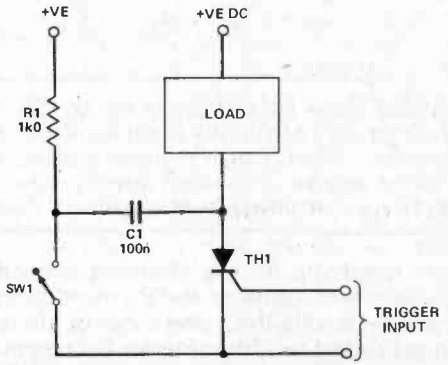


Fig. 3 Turning off a thyristor which is operated from DC. Pressing the switch will discharge the capacitor, pulsing the anode of the thyristor and so stopping the current. This is enough to prevent conduction until the gate is pulsed again.

A Passing Phase

Down to configurations. The most useful basic triggering circuit is the phase-controlled thyristor fed with rectified AC as illustrated in Fig. 4. The load can be placed in the leads to the bridge rectifier, in which case the thyristor will control the average power dissipated in the load,

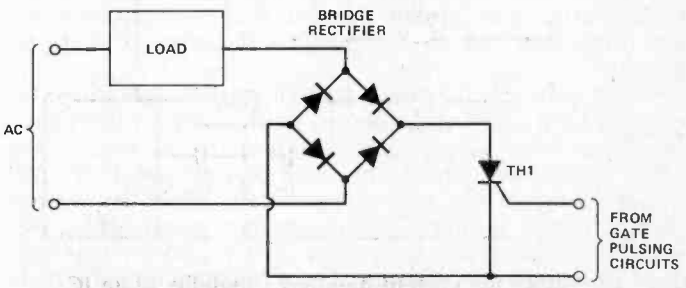


Fig. 4 Basic circuit for thyristor control of an AC circuit, using a bridge rectifier to supply the thyristor. The load, however, operates from AC.

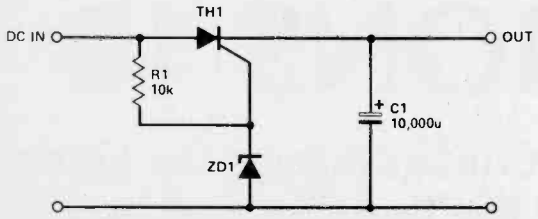


Fig. 5 A thyristor regulator. This makes a very useful pre-stabiliser circuit, or can be used as a stabiliser in its own right where very precise stabilisation is not needed.

despite the fact that the load is working on AC and the thyristor is controlling a rectified supply. An interesting option is to place a reservoir capacitor on the cathode side of the thyristor, giving a low-cost and low-dissipation form of voltage regulation (Fig. 5). The gate control can be obtained from a charging capacitor, as demonstrated in Fig. 6, or from a zener diode as in Fig. 5 — remember that there is no triggering until the gate voltage is about 0V6 above the cathode voltage.

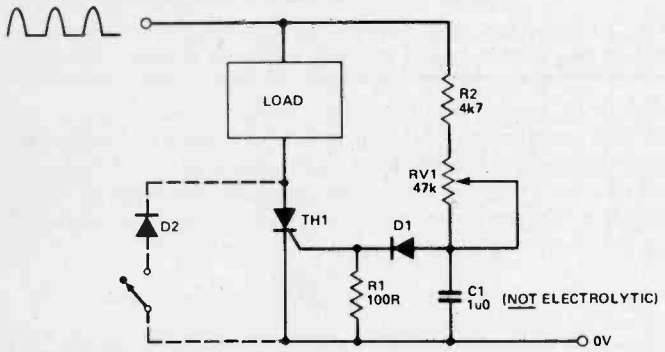


Fig. 6 A typical phase control circuit for AC. The thyristor will conduct on only half of the input wave, so that a 'power-doubler' circuit, which switches a diode across the thyristor in the reverse conduction direction may be needed for a larger range of power control (shown dotted).

Simple triggering from a charging capacitor is never entirely satisfactory, because the thyristor cannot be relied upon to fire at exactly the same stage of charging in each cycle. To get round this, the simpler circuits make use of a trigger diode or **diac** which ensures more reliable triggering. The trigger diode has the curious characteristic that it will remain non-conducting while the voltage across it in either direction builds up, suddenly conduct at some voltage level which is determined by its construction, and remain fully conducting until the voltage across it has dropped almost to zero (Fig. 7). A diac wired between a charging capacitor and the gate of the thyristor, with a load of a few hundred ohms connected between the gate and the cathode to avoid unwanted triggering will serve nicely to make the triggering much more reliable. What you then have to be sure of is that you have enough voltage around to operate the diac — depending on type, you may need up to 15 V across it before it starts to conduct.

The very simple phase-control system operates well enough for a lot of applications, particularly for light dimming, but more care is needed where electric motors are being controlled, mainly because of the back-EMF that motors of the AC/DC type will generate. When any motor of this type is spinning, it will act as a generator of DC (even if the supply to the motor is AC), and the thyristor must be capable of withstanding a reverse voltage which consists of the peak reverse AC plus this additional voltage generated by the motor.

The methods that are used for thyristor control of the

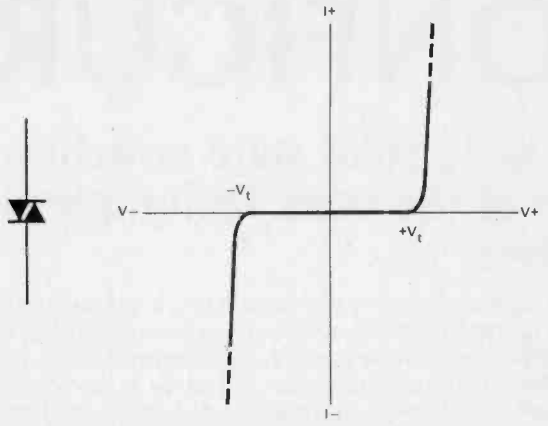


Fig 7. The diac, and its typical characteristic.

larger motors, larger than your domestic power drill/food mixer motor, are a lot more specialised. For these circuits, charging capacitors are simply not precise enough as a method of triggering the thyristor at the correct point in the waveform: more elaborate trigger circuits, synchronised to the mains frequency, have to be used. These pulse-generating circuits can be coupled to the thyristor circuitry by using small pulse transformers, so that the timing circuits need not be connected to the circuits that the thyristor controls. This is particularly important when thyristors are used in high-voltage three-phase circuits, because the thyristors may be operating at voltages well above or below earth, yet the control box needs to be earthed.

Radio interference is a continual problem for any thyristor circuit which makes use of phase control. Because the thyristor is being switched on when there is a substantial voltage across it, there are large current pulses which can be devastating for radio or TV receivers in the neighbourhood and which can also trigger other thyristors. It's essential, therefore, to design really effective pulse-transient suppression into the gate and anode circuits, and to ensure in the practical construction that the suppressors are placed as close as possible to the terminals of each thyristor. In general, small series inductors and

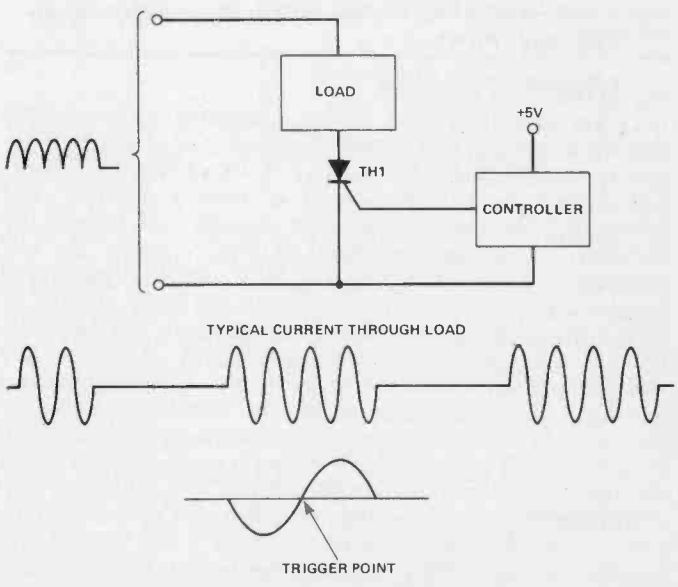


Fig. 8 Principles of zero-voltage switching circuits. The controller (usually an IC) will switch the thyristor on at the point when the AC wave passes through zero. This ensures minimal RF interference, unlike the phase-control method.

parallel capacitors will do all that is needed, but they have to be capable of taking high peak currents, and must be wired close enough to prevent any wiring from acting as a radiating aerial.

The Zero Option

The other way of controlling thyristors in energy-control circuits is seen much less in the small-scale circuits that we tend to be more familiar with. This alternative is zero-voltage switching, and it involves switching the thyristors on at the instant when the voltage between anode and cathode is zero. This has the advantage of generating no more interference than a silicon diode would, which is very much less than is generated by the phase-control circuit: but it can be used only with loads like water-heaters which have very long time constants. If you switch your electric drill motor on for 100 mS in each second, the speed will be rather erratic to say the least, but a water or room heater switched in this way does not cause noticeable fluctuations of temperature because the temperature does not shoot up rapidly when the heater is on, nor shoot down when the heater is off. Figure 8 shows an outline of a typical zero-voltage control circuit — there is an IC which can be used to govern the whole operation.

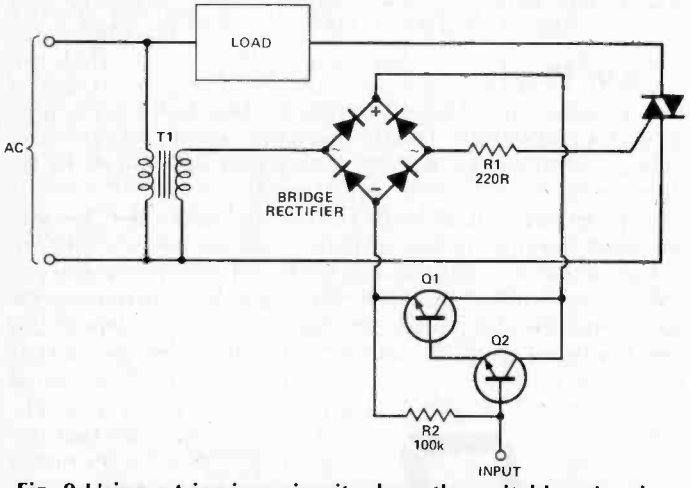


Fig. 9 Using a triac in a circuit where the switching signals are very small. Note that the whole circuit is live to mains.

For My Next Triac . . .

The triac is a two-way equivalent of the thyristor, with the main circuit terminals labelled MT1 and MT2 rather than anode and cathode, since current can flow in either direction through the triac. Like the thyristor, the triac remains non-conducting until it has been triggered by a pulse at its gate terminal; the pulse can be of either polarity, but the minimum amplitude for firing is not the same for the two possible polarities. Again like the thyristor, the triac ceases to conduct when the current through it becomes too low to sustain conduction. Triacs are extensively used to switch raw AC because a triac circuit represents a considerable saving on components as compared to a small thyristor circuit, even if the equivalent triac is more expensive than two thyristors. Figure 9 shows a typical triac circuit for AC use that can operate using a very small triggering input, such as from a microphone or photocell. The transformer supplies a low voltage for the gate circuit, and the rectifier bridge is arranged so that an unsmoothed full-wave rectified voltage is fed to the transistor amplifier circuit. When the transistor conducts, the current flowing in the bridge rectifier will also flow through the gate of the triac, triggering the triac on each half-cycle. The trigger current is AC because the gate is wired in the AC side of the transformer. Note that the whole circuit is connected to mains — if an isolated low-voltage circuit is needed, then the gate must be triggered by a circuit using a pulse transformer rather than directly as in this example, and the part-circuit shown in Fig. 10 is needed.

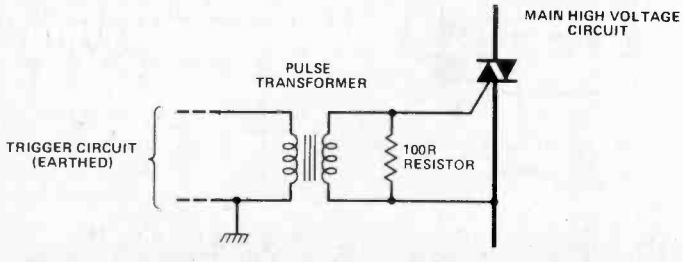


Fig. 10 Isolating the mains part of the circuit from the control part by using a pulse transformer.

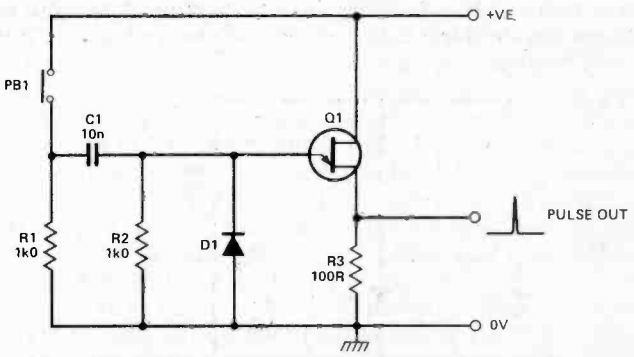


Fig. 11 The unijunction connected to provide a short pulse when a switch is pressed.

Triggering thyristors or triacs via a pulse transformer needs a fairly sharp spike waveform, and one of the devices that has traditionally been used to provide this type of waveform is the unijunction. As the name suggests, this uses one junction on an N-type silicon base whose doping normally ensures that the conductivity is low (resistance high). The junction is placed so as to provide an emitter terminal, and when the emitter voltage is raised to the conducting level, the injection of holes into the bar will make it highly conductive. This is the triggered state, which can be maintained only if a current continues to flow through the emitter. Unijunction circuits are arranged so as to prevent this continuous current, so ensuring a clean sharp pulse.

A unijunction 'one-shot' pulse generator is illustrated in Fig. 11. With the switch open, the emitter of the unijunction is earthed, and the device is non-conducting. Closing the switch contacts changes the voltage on one side of the capacitor from earth to the positive supply voltage, and the voltage on the other side will increase similarly, so triggering the unijunction. The conducting unijunction generates a positive-going spike at the earthy end of its circuit, and also charges the capacitor so that the end of the capacitor connected to the emitter is at about earth voltage. This process is very brief, and when the switch opens again, the emitter of the unijunction is protected from negative pulses by a diode.

The triggering voltage for a unijunction is a fixed fraction of the total voltage applied across the main terminals — the fraction is known as the 'intrinsic stand-off ratio', and is usually around 0.6, implying that the device will trigger when the emitter voltage is about 60 per cent of the supply voltage. Because this ratio is fixed, changes in the supply voltage do not make much difference to the frequency of the output.

CONFIGURATIONS 11

Ian Sinclair has seen the light! Now he wants to illuminate the rest of us. In case you hadn't already guessed, the topic is opto-electronics.

Opto-electronics is a word that hadn't been thought of a few years ago, but which is now used to describe a set of devices that are important enough to merit a part of this series all to themselves. An opto-electronic device is one which makes use of light as part of its electronic function, so this label includes all varieties of devices that convert light signals into electrical signals or the other way round.

The simplest opto-electronic devices of the electricity-to-light type are the familiar LEDs. Familiar they may be, but even experienced engineers are not always aware of their eccentricities. Like any other diode the LED has an anode and a cathode, and passes current in the forward bias direction; this is when the light is emitted. What is not nearly so well known is that the peak reverse voltage of these diodes is very low; if you get an LED the wrong way round in a circuit, it's usually curtains for the LED when the voltage is switched on. A typical value of peak reverse voltage is 3 V, so practically any circuit that will operate the LED when it is connected the right way round (Fig. 1) will blow it up if it happens to be the wrong way round.

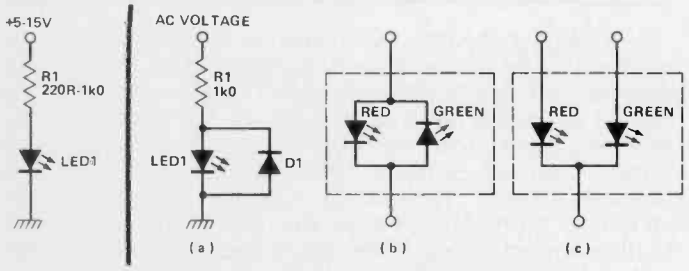


Fig. 1 (Right) The basic LED operating circuit. A current-limiting resistor must always be used unless the output resistance of the driving circuit is high.

Fig. 2 (Left) LED operation. (a) For use with AC, a silicon diode must be connected across the LED terminals as shown. (b) The two-colour LED uses two LED junctions connected in opposite directions. (c) The tri-colour LED uses separate LEDs with a common cathode connection.

In addition, the forward voltage across the LED is very much higher than the 0V6 that we merrily assume for a silicon diode. For gallium arsenide, the material used for many types of LEDs, the forward voltage is more like 2V1 to 2V4, so that LEDs are of little use in very low-voltage circuits — they won't, for example, work from a 1V5 cell.

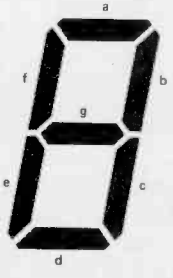
Current Affairs

One of the major snags about LEDs is that they consume a surprising amount of current. Manufacturers quote 'adequate' light output for red LEDs with 5 to 25 mA, and for the green/yellow varieties with 10 to 40 mA. This wouldn't be missed in a circuit operating at 5 V, 2.5 amps, but it can be quite a drain on battery equipment, often considerably more than all the CMOS ICs in a circuit intended for battery operation.

LEDs can be used with AC supplies providing there is a diode connected in reverse across each LED (to prevent excessive reverse voltage) as well as the usual current limiting resistor (Fig. 2a). Bi-colour LEDs consist of a package of two LEDs in one casing, connected in inverse parallel so that current in one direction will give a light of one colour, while the other colour is achieved by reversing the current (Fig. 2b). In this circuit, one LED protects the other against reverse voltage. Tri-colour indicators (Fig. 2c) use two diodes with a common cathode connection and separate anode leads, so that three colours can be indicated, one in each lead, plus yellow when both LED sections are activated. Personally, for indicating when mains voltage is on, I much prefer the old-fashioned neon.

On Display

When it comes to digit displays, LED types have quite a lot of competition. The traditional seven-segment display (Fig. 3) comes as a common anode or a common-cathode type (Fig. 4), and each type needs a separate limiting resistor in each driver lead. The normal method of use is to connect the display to a decoder chip such as the 7448 or 7447, which in turn takes the digital information in as BCD signals — four bits per digit. The snag again is the current consumption, 10-20 mA per segment, which means that displaying a figure '8' uses 7 x 20 mA — 140 mA just to



No.	SEGMENTS LIT
0	a b c d e f ·
1	· b c · · · ·
2	a b · d e · g
3	a b c d · · g
4	· b c · e f g
5	a · c d · f g
6	a · c d e f g
7	a b c · · · ·
8	a b c d e f g
9	a b c · · f g

Fig. 3 Layout of the seven-segment display, with segment guide. An eighth segment, the decimal point, is often added.

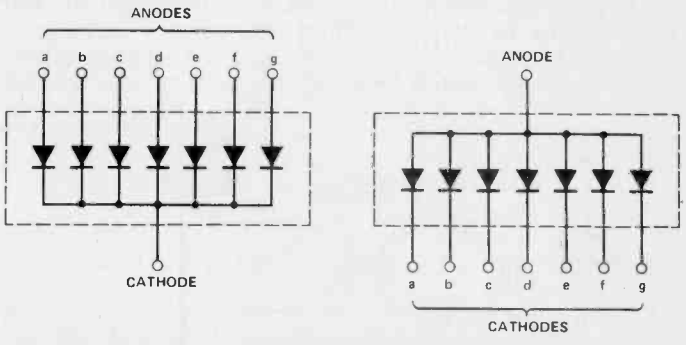


Fig. 4 Internal connections for common anode and common cathode displays. Whichever type is used, there must be a separate limiting resistor for each segment.

display one digit! While mains-powered equipment isn't too upset by this size of current, the LED seven-segment display did not last very long in battery-powered devices, even when multiplexing was used. Multiplexing means that only one digit at a time is activated, the digits being switched on in sequence fast enough to present the appearance of all the digits being illuminated at once.

Oddly enough, the forward voltage for the segments of an LED seven-segment display tends to be lower than for diodes, around 1V3 to 1V7. At temperatures above about 25°C, the maximum current has to be reduced by 0.3 mA per degree to avoid over-dissipation of the junction in each segment.

One competitive display that seems to be much less well-known is the filament seven-segment display. This can use as little as 5 mA per segment, and looks surprisingly bright — it can be driven by a decoder directly with no limiting resistors, and for many purposes is superior to LED displays. The usual reason for preferring solid-state displays is long life, but the quoted life of more than 100,000 hours for the filament type of display is pretty competitive, and some LED displays are notorious for short life — one frequent candidate for replacement in my experience is the display used in the old KIM microprocessor units.

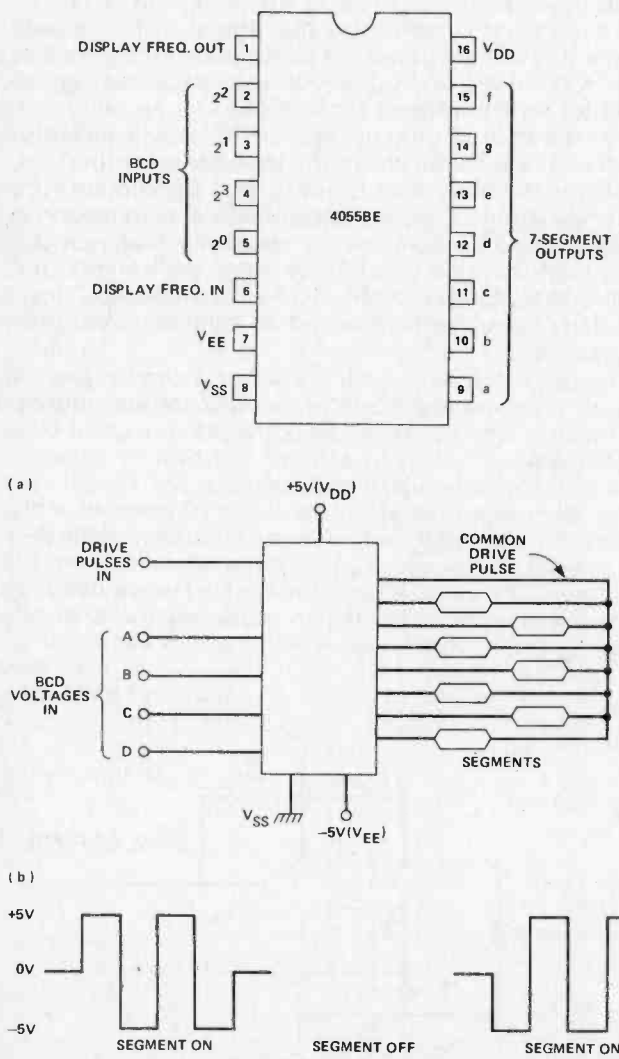


Fig. 5 Driving LCD displays. The common lead of the LCD display must not be earthed; it has to be returned to the driver IC. The waveform (b) applied is AC with no trace of DC.

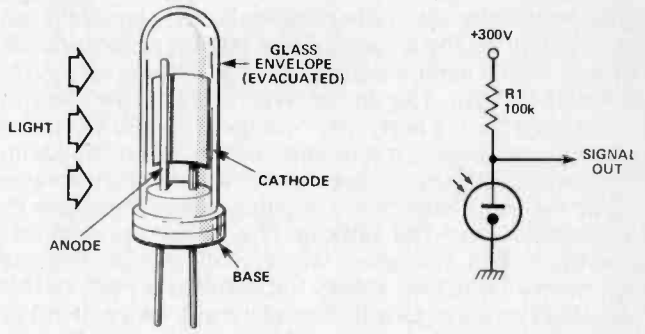


Fig. 6 The vacuum photocell, and a typical circuit arrangement.

Oldies But Goodies

The two older types of displays which are also worth considering are the electron-beam type and the gas-discharge type. The electron beam display uses a miniature cathode wire to emit electrons, which will then be attracted to any positive anode. The anodes are coated with phosphors (similar to the phosphors used in cathode ray tubes), and any anode which is positive to the cathode by a sufficient voltage will glow. A 24-40 V supply is needed, which usually means the use of an inverter when low-voltage batteries are used, as in calculators. The display is easy to read, and uses less current than the LED type — I still prefer a calculator using this type of display to one using the more-common LCD display.

The gas-discharge display is an older type which uses the principle of the neon light — ionisation of a low-pressure gas in an electric field. Like all gas-discharge, this needs a high operating voltage, around 150-250 V, but the operating current is very low: only 0.7 mA per segment in a typical application. The display is very bright, and is worth considering for mains-operated equipment whose display has to be viewed under difficult illumination conditions, such as alternate brightness and darkness. A driver IC is available nowadays — in times past (dare we say the Dark Ages?), the major handicap of using this type of display was the lack of suitable driver transistors.

Liquid Light

Last among the displays, of course, there is the LCD. A good LCD can give a dense black indication against a light grey background, is clearly visible in bright light, and reasonably visible even in low illumination conditions. There's a lot of variation between displays, however, even from the same manufacturer, and some are poor, with low contrast and very slow response to changing digits. Prices also vary considerably — one catalogue I have lists the price of a calculator-size display as being twice as much as I would have to pay for a complete calculator using a similar display!

Operating conditions for these displays are very different from those of other types of displays, because they have to be operated from high-frequency AC supplies. For this reason, displays either come with all the necessary circuitry for generating their driver pulses built in, or they can be used with a standard chip intended for this purpose. It's particularly important not to apply DC to the segments of an LCD display, because this can kill the display very rapidly.

On The Receiving End

Moving to the other end of the opto-electronics business, we find the photocells. Vacuum photocells and

photomultipliers are rather specialised, and we'll only touch briefly on these types. They rely on photocathodes, surfaces which emit electrons into a vacuum when they are struck by light. The anode which collects the electrons (Fig. 6) must be at a fairly high voltage (100-500 V), and the currents are small: microamps rather than milliamps. Photomultipliers obtain greater sensitivity and increased output by using secondary multiplication, meaning that the electrons from the cathode (Fig. 7) are accelerated to surfaces, called dynodes, which will release electrons each time an electron strikes the surface. If each of these multipliers releases two to five electrons for each striking electron, spectacular gain can be achieved which, unlike amplification of signals by conventional methods, is practically noise-free.

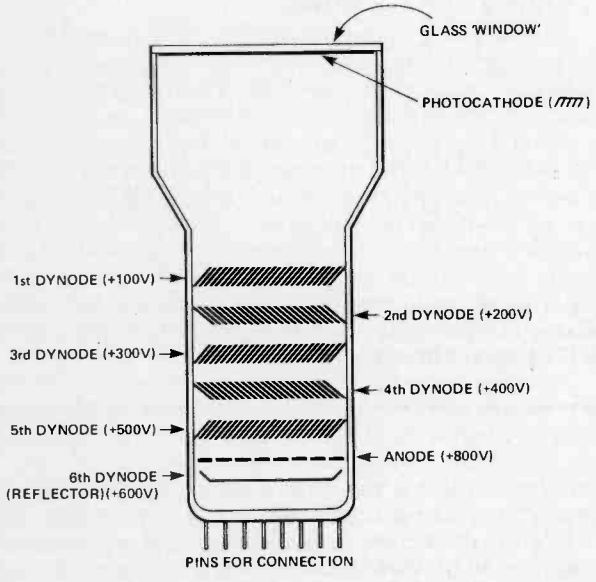


Fig. 7 Cross-section of a photomultiplier, used for detection of very low light levels.

The more familiar solid-state light-to-electrical-signal devices that we use are the solid-state photodetectors, of which the most commonly used is the cadmium sulphide cell. The ORP12 is the standard device of this type, often called an LDR (light dependent resistor). The cell consists of a strip of cadmium sulphide whose resistance decreases as light falls on it. The resistance in the dark is high, up to 10M, and the resistance can fall as low as 100R in bright sunlight. A less well-known aspect of these cells is that they can withstand a fairly high voltage, around 100 V; subject to their dissipation limit of 200 mW, meaning that you might need a limiting resistor connected in series. The cadmium sulphide cell is a slow-acting device, needing about 350 mS for the resistance to fall on exposure to light, and around 75 mS for the resistance to rise again when the light is shut off. The response to different colours is generally similar to that of the human eye, but the cadmium sulphide is much more sensitive to red and infrared, which is why its use in cameras is now less common than it was some 10 years ago.

Fun With Photodiodes

Other light detectors need some degree of amplification. Photodiodes are diodes of fairly conventional construction, with a transparent window over the junction,

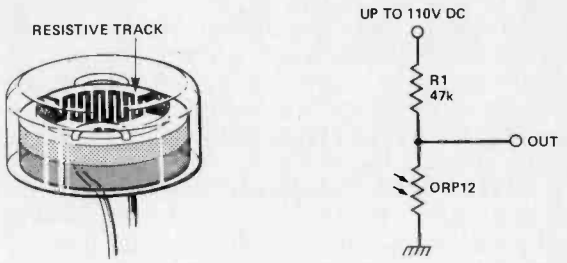


Fig. 8 The photoresistive cell or light-dependent resistor (LDR).

which are used reverse-biased. For such a diode, the reverse leakage current increases as the intensity of light on the junction is increased. This current is small, ranging from around 1 nA in darkness to almost 1 mA in very bright light, so that amplification is usually necessary, as in the circuit of Fig. 9. The response time is about 250 nS, so that the op-amps shown in Fig. 9 would have to be replaced by a transistor circuit, using high-speed switching transistors, if you wanted to use the photodiode for high-speed signals. Combined photodiode/op-amp packages can be bought for medium-speed applications.

The old-style phototransistor, which was a transistor formed with a window above the base-collector junction, is a thing of the past: what is now called a phototransistor is a combination of silicon photodiode and transistor in one package. This combines a sensitivity that is much greater than that of a photodiode alone with a good fast response time, giving typically a 2 MHz bandwidth. This is particularly useful for receiver use in light-beam transmission systems.

The optoisolators, which consist of a combination of LED and phototransistor are embedded in clear plastic, which allows light transmission but which is a good electrical insulator. It's easy to achieve isolation to at least 4 kV, with reasonable signal transmission. For an ordinary isolator, the output signal will be about 20 per cent of the amplitude of the input, but when a Darlington phototransistor is used, the output can be three times or more the amplitude of the input. It's just the device I was looking for 25 years ago when I wanted to modulate the grid of a cathode-ray tube which was working at −4 kV!

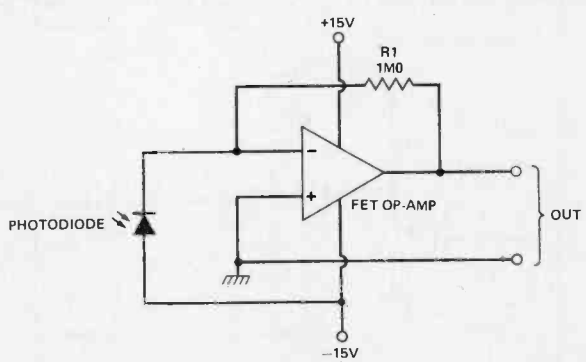


Fig. 9 Using a photodiode in conjunction with a FET op-amp. The FET type is needed because of the very high impedance of the photodiode circuit.

CONFIGURATIONS 12

Now Ian Sinclair brings the series to a close, shutting the AND, OR, and NOT gates behind him.

For anyone who has worked with linear circuits for a long time, the first contact with digital circuits always comes as a shock because the action of digital circuits is unfamiliar, and the way in which the circuits respond to signals is equally unfamiliar. In this final part of Configurations, therefore, we shall concentrate on the most basic of digital circuit, the gate, and how the two most common of 'families' of digital circuits, TTL and CMOS, carry out gate action. For once, also, we're going to assume rather less in the way of background knowledge than we've taken for granted in previous parts, because all the problems in adapting to digital circuitry are at the start — once you have had some experience, this sort of message is not needed!

Let's be clear from the start what we mean by digital circuits and gates. A digital circuit is, strictly speaking, one which works with signals that consist of several separate voltage levels, so that a voltage which is to be counted as a signal must be at or near one of these levels. The digital circuits that we make most use of are binary digital circuits, meaning that the signals into them and from them consist of only two voltage levels which we refer to as a matter of convenience as 0 and 1. What the actual voltages happen to be is unimportant — the important feature is that there should be just these two levels. Most logic circuits operate with what we call positive logic, in which 0 means zero volts and 1 means a positive voltage; a few older circuits can still be found which use negative logic, in which 1 is a negative voltage.

The advantages of using just two voltage levels are considerable. We don't have to worry about bias, for example, in the design of circuits, provided that we arrange for each active device in a circuit to be turned on at one voltage level and off at the other. This encourages the use of ICs, because bias is difficult to arrange reliably inside ICs. We don't need much in the way of voltage amplification, because with only two voltage levels to consider, the output signals can be of about the same voltage levels as the input signals. The only voltage amplification we need to consider is as much as is needed to restore the 1 level to normal when it has been reduced by, say, the 0V6 drop across a conducting diode (Fig. 1). The third major factor is that tolerances in component values have much less effect on signals than they have in linear circuits. A logic 1 voltage which is nominally 5 V can drop as low as 3V6 and still be useable as a logic 1 voltage. The logic 0 voltage can rise as high as 0V8 and still be useable as a logic 0 voltage.

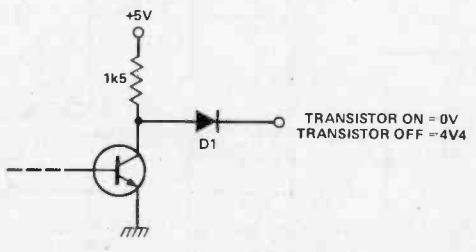


Fig. 1. Voltage levels. The presence of a diode, or a transistor junction, in the path of an output can change the output level by 0V6 or so. The tolerance of voltage must be enough to make allowances for this.

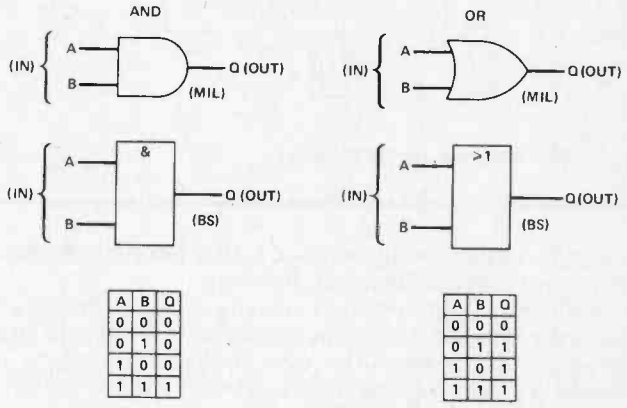


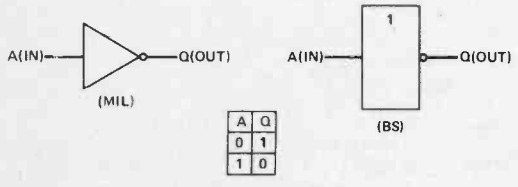
Fig. 2. The two main gate types, with International (MIL) symbols, and the BS symbols that are used for TEC and C & G courses. The truth tables describe the gate actions.

Since the normal concern of linear circuits, amplification with low distortion, is simply not necessary for digital circuits, the actions that digital circuits perform are necessarily quite different. One of the fundamental actions of a digital circuit is gating, and it is gating that we shall look at in the rest of this article.

Digital Gates

A digital gate is a circuit which has inputs that are digital signals and an output (or more than one output) which is also a digital signal. Since the output is a digital signal, it must have a voltage level at any instant which is at logic 0 or at logic 1, and what the level actually is depends entirely on the combination of inputs that happens to be present at that instant. It is for this reason that the gate circuit is often referred to as a **combinational** circuit. The two most important types of gate circuits are referred to as AND and OR gates respectively, and we can describe their actions by a table that shows what the output will be for every possible combination of inputs. Such a table is called a 'truth table', and the truth tables for AND and OR gates with two inputs are illustrated in Fig. 2. These tables show that for the two-input AND gate, the output will be at logic level 1 only when both inputs are at level 1: for the OR gate, the output will be at level 1 when either or both inputs are at level 1.

Truth tables become less useful when a gate has a large number of inputs, because the number of lines needed for a truth table is 2^n, where n is the number of inputs to the gate. The same rules apply, however, irrespective of the number of inputs, so that the action of the AND and the OR gates can be described in ways that are more compact

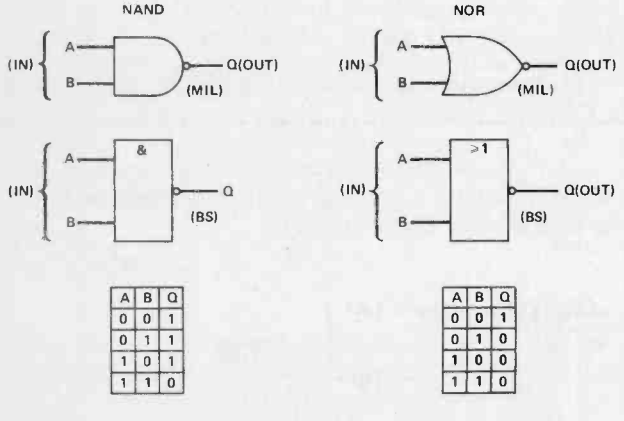


A	Q
0	1
1	0

Fig. 3. The inverter or NOT gate.

than truth tables, using what is called Boolean Algebra. We haven't space to deal with this here.

Another circuit which is usually classed among the gates is the inverter, sometimes called a NOT gate. Its truth table (Fig. 3) is simple — the logic voltage output is the inverse of its logic voltage input. Circuits which combine the action of the NOT gate with the action of AND are called NAND gates; circuits which combine NOT action with OR action are called NOR gates, and the truth tables for these types are shown in Fig. 4. One further gate which is less important as a basic circuit, but which is needed in arithmetic circuits, is the exclusive-OR gate, or EXOR-gate, whose action is illustrated in Fig. 5. The name comes from the fact that the action is like that of the OR gate but excluding the case where both inputs are 1.

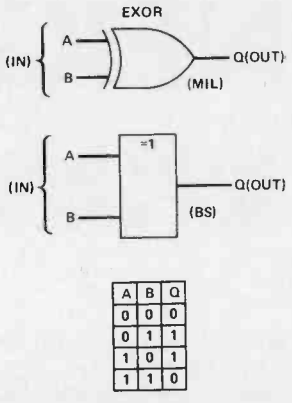


A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

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

Fig. 4. NAND and NOR gates, formed by combining AND/OR gates with inverters.

Logic circuits which make use of gates are connected so that the output of one gate can pass signals to the input of the next gate in the circuit — we say that one output can **drive** one or more inputs. This usually means that the output has to be able to supply (**source**) or absorb (**sink**) current, and the number of inputs that can be driven by one output is called the **fanout** of that gate. The size of the fanout depends on the design of the input and the output stages of the gates. A fanout of 10 is generally considered



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

Fig. 5. The exclusive-OR (EXOR) gate and its truth table.

to be satisfactory, meaning that 10 gate inputs can reliably be driven from one gate output.

TTL Gates

The old-style 'standard' TTL gate uses bipolar transistors, but using a common-base circuit rather than the more familiar common-emitter. The inputs (Fig. 6) are to the emitters of transistors whose bases are connected through a current-limiting resistor to the supply positive voltage of 5 V. A common feature of the IC construction is the creation of several emitters on to one base, so that several inputs are fed in by the same transistor. An input stage like this will draw no current when the input voltage is logic 1, because such an input biases the transistor off. An input which is at logic 0, however, has the effect of earthing the input terminal, and current will flow through the base-emitter junction of the transistor to earth. Unlike our linear circuits, this input current comes *out* from the input! Standard TTL is constructed so that this current is about 1.6 mA, so the resistance between the input terminal and earth must be low enough to ensure that when this amount of current flows, the input voltage at the terminal must not rise above the maximum voltage level permitted for logic 0, usually around 0V8.

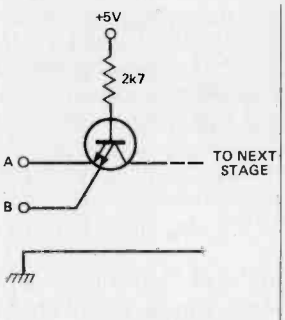


Fig. 6 (Left) TTL input. The base of the transistor is connected to + 5 V through a resistor, and the inputs are to emitters — more than one emitter (and as many as 13) can be formed on to one base.

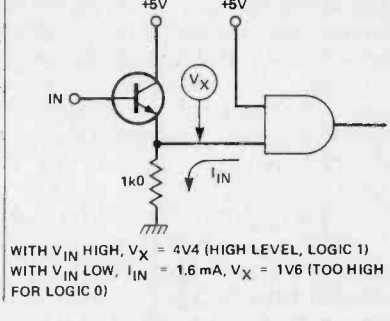


Fig. 7 (Right) Driving a TTL stage from an NPN emitter-follower. The logic 0 voltage is likely to be too high because of the current from the input of the gate.

The requirement to have current flowing *out* from the input at logic 0 means that not all driving circuits are useable. In particular, the NPN emitter-follower, which is so often the automatic choice for many purposes, is unsuitable because (Fig. 7) when the input is at logic 0, the current from the gate will flow through the emitter resistor. A PNP emitter-follower, arranged as shown in Fig. 8, can allow a satisfactory logic 0 voltage, but only if the voltage at the base of the emitter follower can be taken low enough — preferably to a negative voltage, because of the inevitable 0V6 difference between base and emitter voltage levels. The most satisfactory simple driving stage is the straightforward common-emitter amplifier circuit as shown in Fig. 9.

No driving problems should exist if the input of a gate is driven by the output of another gate of the same family. Figure 10 shows the conventional circuit arrangement for a standard TTL gate output, which uses two transistors and a diode in series. A logic 1 output corresponds to having

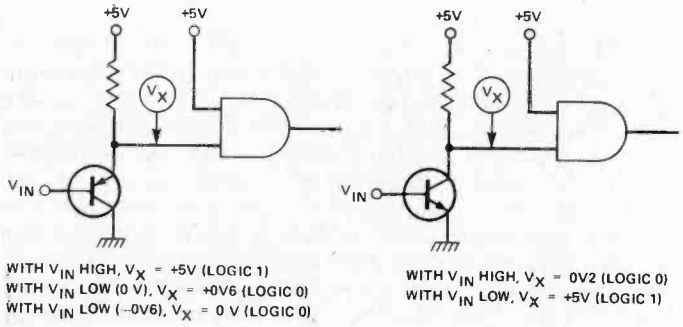


Fig. 8. (Left) Using a PNP emitter follower as a driving stage — a better approach.

Fig. 9. (Right) Driving a gate from a common-emitter stage — the most satisfactory single-transistor drive stage.

the top transistor of the pair conducting and the bottom transistor shut off, and because the base voltage of this top transistor cannot be more than the supply voltage of +5 V, the emitter voltage must be no more than 4V4-4V5, which makes the output voltage (because of the diode) only around 3V8-4V0. Don't be surprised, then, if you find that the logic 1 output from a gate is lower than the supply voltage. The logic 0 voltage from this circuit will be the voltage across the bottom transistor when it is fully conducting, which can be as low as 0V2, depending on the load.

The layout of the output stage is such that only one of the output pair of transistors will be conducting at any time during normal operation. If two gate outputs are connected together, however, it would be possible to have one output at logic 1 (top transistor conducting) and the other at logic 0 (bottom transistor conducting), so that at a low resistance path for current was created (Fig. 11). This would have the effect of burning out one transistor in each gate, so that for the few applications in which gate outputs have to be connected together, special gate ICs described as open-collector types are used. These have no 'top' transistor in the output stages, and are designed to work with an externally connected resistor load (Fig. 12).

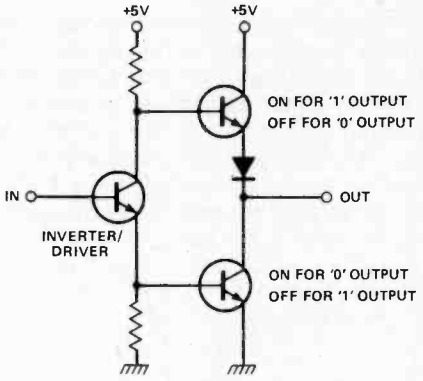


Fig. 10. The usual TTL output stage. One of the pair of output transistors will conduct to connect the output to either 0 or 1 levels.

Standard TTL, though still circulating in very large numbers, has been replaced in production by the low-power Schottky TTL chips, distinguished by the letters LS in the type numbers. These LS chips make use of a component, the Schottky diode, which is not particularly well known, so that some description is called for. The Schottky diode uses a combination of metal (usually aluminium) and semiconductor in its junction to obtain a very low for-

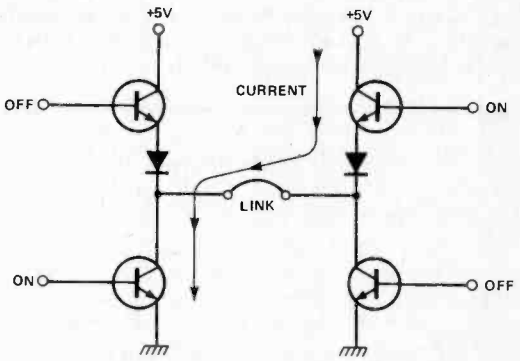


Fig. 11. Why gate outputs should not be connected together.

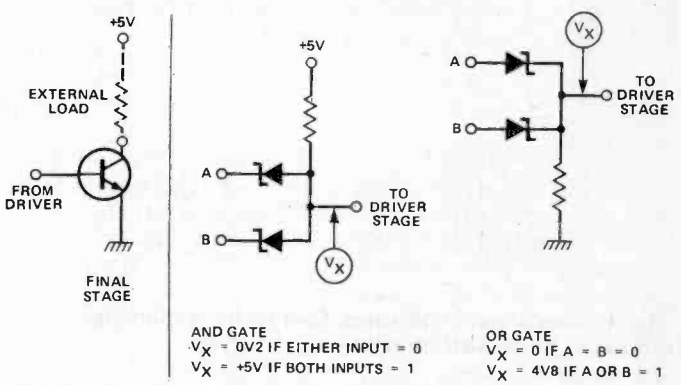


Fig. 12. (Left) The output stage of an 'open-collector' stage. These stages need an external load resistor.

Fig. 13. (Right) Using Schottky diodes as logic elements.

ward voltage, between 0V1 and 0V2 as compared to the 0V6 for a silicon diode. This makes these diodes ideal for use in logic circuits, as illustrated in Fig. 13, and also makes it possible to construct transistor stages which do

not saturate. Saturation occurs in a conventional transistor stage when the base current of a transistor, which has a collector load, is so high that the collector voltage bottoms. The effects of saturation are to achieve a very low collector voltage, around 0V2, but also to flood the base junction with charge carriers (electrons or holes depending on whether the transistor is PNP or NPN). When the base voltage is suddenly removed from such a saturated transistor, this charge takes some time to clear, so that the transistor remains conducting — it will not switch rapidly from the conducting state to the non-conducting state. The time is usually less than a microsecond, but it limits the speed at which a gate circuit can operate reliably.

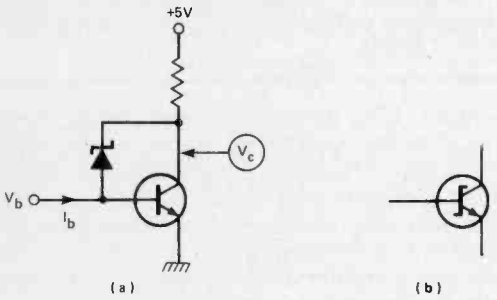


Fig. 14. (a) Using a Schottky diode to prevent transistor saturation. (b) The symbol for a transistor into which a Schottky diode has been incorporated.

A Schottky diode placed between the collector and the base of a transistor (Fig. 14) will prevent such saturation. When the collector voltage reaches a level which is about 0V2 lower than the base voltage, the Schottky diode wil conduct, connecting the base and the collector circuits and so bypassing the base. By avoiding saturation in this way, the transistor can be made to switch very much more rapidly at the minor expense of having a collector voltage which does not reach quite so low as that of a standard TTL stage. Figure 15 shows the internal circuitry of a typical LS type of gate circuit in which the presence of Schottky diodes is indicated by the modification to the shape of the base symbol in the transistors.

CMOS

Finally among the commonly-used logic gate circuits we have the CMOS types. These depend on the use of MOSFETs rather than bipolar transistors, and the inputs are invariably to the gates of the MOSFETs as compared to the emitters of the transistors in TTL stages. For this reason, no measurable current flows either into or out from the input of a CMOS gate when we use low-frequency signals, and the fanout under these conditions can be very high. The size of fanout is limited by the ability of the outputs of CMOS gates to supply currents of more than a milliamp or so, because the capacitance of each CMOS input is fairly high, and rapid switching demands that each capacitance be charged and discharged rapidly, calling for current which the output of a CMOS gate may not be able to supply. The operating currents and the dissipation of a CMOS gate will therefore increase as the operating frequency is increased, and it is this factor which limits the fanout and the speed of these gates. A typical CMOS gate circuit is shown in Fig. 16.

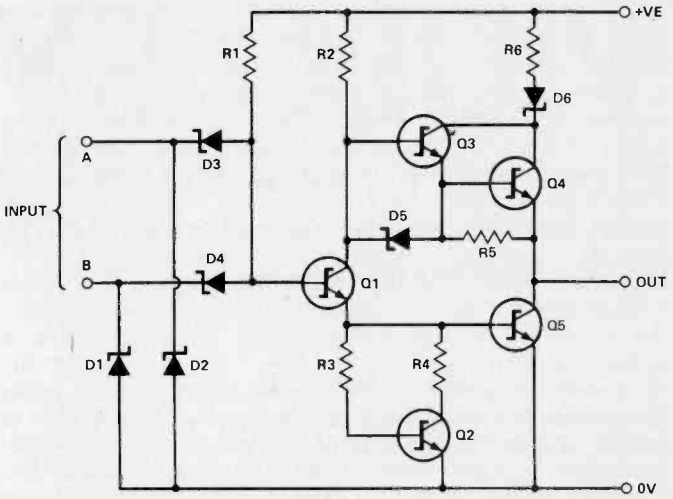


Fig. 15. Circuit for a NAND gate using Schottky diodes.

CMOS gates have a convincing list of advantages for many purposes. The supply voltage can be in the range + 3 V to + 15 V rather than the fixed + 5 V of the TTL circuits. The currents that are required by CMOS gates are very much smaller, so that CMOS is almost an automatic choice when battery operation is required.

For most practical purposes, your choice of logic circuits will be between LS TTL types and CMOS types, with the CMOS types chosen from the 4000 family (RCA), or from the less-well known 74C series (National Semiconductor) in which the type numbers correspond to those of the 74 series of TTL chips. For all purposes which require

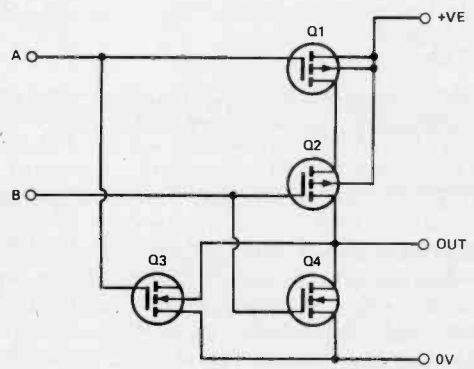


Fig. 16. Circuit of a typical CMOS gate — this one is an AND gate.

low consumption, lower operating speeds and small power outputs, CMOS is the more likely choice, but LS TTL chips are essential for many computing operations in which a high clock-rate is used — you may even find that you need the still-faster (and more power-consuming) 74H types.

We have now reached the end of Configurations, having covered a large number of devices and circuits. I hope that you have found the descriptions and the hints useful, and that some of them will have opened up new frontiers in circuit design for you, because that was the aim of the series. All I can do now is to wish that your circuits will always do what you want them to!

VIDEO SYSTEMS

It's probably the most advanced piece of engineering you'll ever have in your home — but it isn't that difficult to understand. Stan Curtis makes the hardware look easy.

In a matter of just a few years home video has become big business; second only to home computers as a means of taking away our hard earned money in exchange for boxes of wonder electronics. The pace of development continues with four video recorder formats now in use, three video disc formats imminent in the shops, and new camera/recorder/television technologies just around the corner. This rate of product change (Grundig released then replaced three recorders in 15 months!) coupled with a puzzling reluctance on the part of the manufacturers to release anything resembling technical information has left the electronics enthusiast a little in the dark. Many have just thrown in the towel and work on the basis of "an input socket and an output socket and what's in between is none of my business". Others have made innocent enough enquiries of the so-called technical departments of some of the importing companies. The standard responses vary from a shovel-load of pseudo-scientific mumbo-jumbo to downright suspicion of the "why do you want to know; you're not going to tamper with one of our machines, are you?" kind.

So the time has obviously come for us to present a basic primer on the state of today's video technology together with some background on basic video principles.

The Basic Principles

Before you can start to understand how video equipment works it is useful to learn a few of the principles and a few of the key words. For example, just how do we get a picture on the television screen? Each complete picture is termed a frame and lasts for 1/25th of a second; in other words synchronised to the 50 Hz mains supply with 25 frames per second. Each frame consists of 625 horizontal lines (in the UK) which are written across the screen during the frame time. Unfortunately the picture rate of 25 per second causes a flickering effect which is most annoying, so a way had to be found to increase the effective picture rate to 50 per second without increasing the video bandwidth. The answer was interlaced scanning, where the picture is scanned at 50 frames per second rate (to avoid flickering) but on each scan only half the lines are traced out, leaving a gap between each pair for the missing lines. Each scan is called a field and during the second field all the missing lines are scanned. The picture is made up of odd lines, even lines, odd lines, etc so that in every second exactly the same amount of data is transferred (hence the same signal bandwidth) but without the flicker.

That Syncing Feeling

In order that the picture be accurately reconstituted on the screen it is necessary that there be some sort of synchonisation between the signal source and the receiver. The synchronisation is achieved by the use of pulses. There is a sync pulse at the start of each line and a series of sync pulses at the start of each field. These field sync pulses are repeated at half line spacings so that the line sync is not lost and a series of equalising pulses (of opposite mark/space ratio) are also added to maintain the average signal level.

Sync pulses and picture signals are kept separate by keeping the former below and the latter above the black level. Thus it is quite simple to separate out the sync pulses at a later time. The combined signal of both picture information and the sync pulses is usually referred to as composite video.

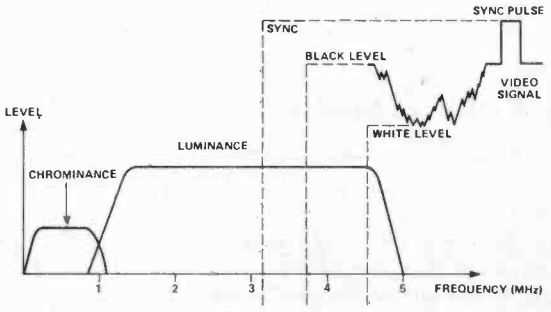


Fig. 1 The bandwidth of signals during video FM recording.

A very wide bandwidth is required to handle this signal which extends down to DC. The DC component must be accurately maintained because any change in its value will affect the average brightness of the signal. The high frequency bandwidth can be calculated by considering the picture resolution. Each of the 625 lines has a duration of 64 microseconds of which 13 microseconds is used for a black margin at either side of the picture. Thus for horizontal resolution of 575 picture elements there will be a need for a bandwidth of about 5.6 MHz. Similarly we can see that if a domestic video cassette recorder has a bandwidth of 3 MHz the horizontal resolution will drop to below 300 picture elements.

The video picture signal varies in DC level at any instant, the voltage determining the grey tone of the picture. The highest DC level represents white while the lowest DC level is black, the greyness varying linearly between these limits. The brightness signal is termed the luminance signal to distinguish it from the colour or chrominance signal.

Hue And Y

Once colour is considered the video theory becomes steadily more complex. Colour has two characteristics; hue which describes its colour (red, yellow, etc) and saturation which describes the percentage depth of the colour. Thus a 10% red will be a faint pink while 100% will be a deep strong red.

The colour camera converts the colours of the subject into three outputs, red, green and blue, from which any of the original hues can be reconstituted. They can also be mixed in the ratio 30%-59%-11% to produce the luminance signal (Y):

$$Y = 0.3R + 0.59G + 0.11B$$

The percentages are chosen to follow the sensitivity of the eye. The chrominance signals are then derived by subtracting Y from

each to give the three difference signals R—Y, G—Y, and B—Y.

Although it is possible to send each of these signals separately it is obviously more convenient to combine them as a single colour signal. The first operational system to do this was the NTSC developed in the early 1950s in the USA. Later came the SECAM system in France and the PAL system developed by Telefunken in Germany. The three systems are incompatible with each other as many people have learned to their cost when they have imported NTSC equipment from the USA.

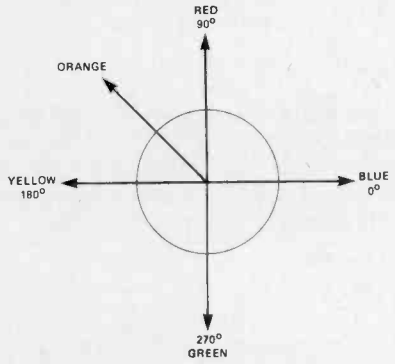


Fig. 2 How colours are determined by the phase angle of the sub-carrier signal.

The colour signal is encoded using suppressed carrier quadrature modulation. This means that the R—Y signal is modulated on the 4.43 MHz subcarrier whilst the B—Y signal is modulated on the same subcarrier 90° out of phase. When the two subcarriers are combined the result is a single signal whose phase angle varies in relation to the two components (see Fig. 2).

Thus the hue of the colour is defined by the phase angle and the saturation by its amplitude. To do this we have only used the B—Y and R—Y components since the G—Y component can always be derived from the other two.

Suppressing The Truth

Now we come to the suppressed sub-carrier bit. The chosen frequency of 4.43 MHz sits right inside the 5 MHz luminance bandwidth and its presence would therefore cause a visible pattern on the screen. The solution is to suppress the carrier frequency leaving just the sidebands.

Again some sort of synchronising signal is needed to enable the colour signal to be reconstituted accurately. So for colour a 10 cycle burst of 4.43 MHz carrier is inserted ahead of the video picture signal. This gives an accurate reference frequency to enable the suppressed sub-carrier to be reformed by a local oscillator in the TV which is 'kicked' into sync by this colour burst. The phase of this burst also acts as a reference in decoding the difference signals.

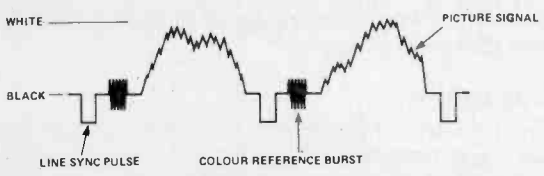


Fig. 3 The composite video signal with line sync pulses.

The foregoing applies to both the NTSC and the PAL systems but in the latter the phase of the R—Y signal is reversed on alternate scan-lines and so the reference colour burst changes phase through 90° on alternate scans. This allows phase errors to be averaged over adjacent lines, avoiding the colour shift which has earned NTSC the nickname 'Never The Same Colour'.

Video Cassette Recorders

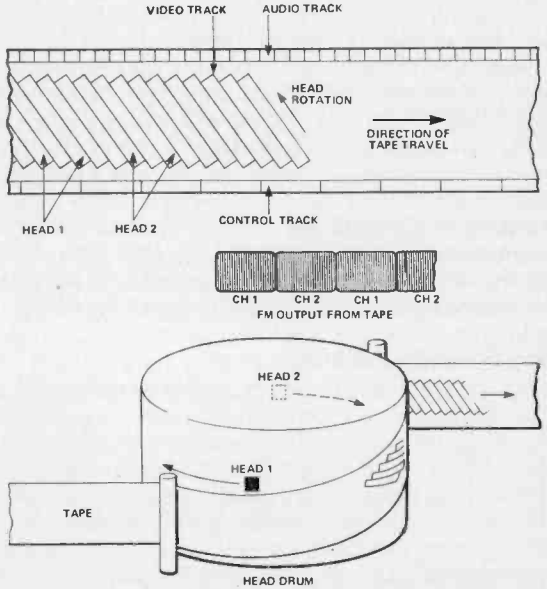
The first video cassette recorder appeared in the early 70s with Sony's ¾" tape U-matic being introduced in 1970 and the Philips NR-1500 system a year or so later. Both systems used the helical scan technique (see the box) and although both aroused some interest in the domestic market the great majority were sold to educational and industrial users. In time the U-matic recorders became the standard format for industrial users while Philips went on to the 1700 series and, to all intents and purposes, had the domestic market entirely to themselves.

However, in late 1975 Sony introduced the first examples of their Betamax home VCRs, whose technology was broadly based upon the U-matics although scaled down to use half-inch tape. Not long afterwards JVC (despite making U-matics under license from Sony) jumped in with their competing system, VHS. Initially this system offered longer recording times than Betamax (two hours) and for a few years a war was waged in the main market (the USA) with each format trying to offer longer play times. Indeed half-speed VCRs went on sale in the USA, and although they offer frugal use of tape the picture quality is truly awful. The broadcast video bandwidth is about 5 MHz and the average VHS recorder can manage about 2.8 MHz. Halve the recording speed and the video bandwidth drops to 1.4 MHz while the video noise level rises. The result is a fuzzy, grainy picture which is almost unviewable. Once VHS reached a playback time of six hours (half-speed) the competition became pointless — after all how many six hour movies do you want to watch?

HELICAL SCANNING

A conventional audio tape recorder uses linear scanning — the tape moves across the recording heads horizontally with the audio signal being recorded along the length of the tape. This system works well at the tape widths and speeds used because of the limited audio bandwidth (only 20 kHz or less). However, a video signal has a much greater bandwidth, as described in the main text; to record the TV pictures requires about 200 times as much information per second, yet the video tape is only four times wider than audio tape and travels about the same speed. How can the machine pack all the extra information on?

The trick is to make the recording head move as well; a speed of approximately 1500 RPM is used. Instead of passing the tape horizontally across the rotating head drum, the tape guides position it at an angle as shown in the diagram. Two tiny recording heads are positioned half-way up the drum and on opposite sides, so that one is always in contact with the tape. The rotation of the drum means that the heads sweep across the tape at about 5 metres per second; 200 times faster than an audio recorder. As the first head passes across the tape it writes a diagonal stripe of information; the slow movement of the tape across the rapidly spinning drum ensures that the second head will write its stripe adjacent to the first, and so on. This technique is called helical scanning and is used by all video recorders of all formats at present.



EEC VCC

Meanwhile back in Europe Philips, working at what seemed to be a leisurely pace, conceived their 2000 system which gradually became known as the VCC (Video Compact Cassette). The system was launched in partnership with Grundig and at once a major blunder was revealed. Somewhere along the line both companies arrived at a different understanding of the same drawing and positioned their audio heads at different points. The result was instant incompatibility between the two compatible models. There were red faces and dark mutterings all round, after which it would appear that Grundig dug their heels in and Philips did some quick mods! In its final form the VCC format offers a turn-over cassette offering four hours recording-time per side — so on the basis of playing time they have really socked it to the Japanese. The picture quality is very good and this is due in part to the clever use of a technique called Dynamic Track Following (DTF) which is explained in the second boxed section.

A more recent format intended for portable recorders is the Funai which also appears under the Technicolour and Grundig brand names. These recorders use an audio-sized cassette filled with ¼" metal tape. Although the quality is very good the high writing speed has limited the playing time to 30 minutes. The tape transport mechanism is very small and light with the result that the weight of a typical ¼" video recorder is now not much above 3 kg; hence the Japanese are now designing combined camera/recorders.

DYNAMIC TRACK FOLLOWING

The Philips and Grundig VCC video cassette recorders use an ingenious control system called Dynamic Track Following (DTF). Unlike the VHS and Beta recorders the VCC machines do not have a linear control track recorded on the tape; instead they have an arrangement based around the use of two video heads whose height can be adjusted by the means of a piezo-ceramic element.

During the recording process one head is held in a fixed position and the other is capable of being moved by a special error correcting signal. When the vertical blanking period occurs (and hence no visible picture) Head One is switched to playback and it sweeps the track just recorded by Head Two. One of the recorded signals is of 233 kHz and the detected signal causes Head Two to be moved until this signal reaches its maximum amplitude. When this is achieved the two heads are in their correct relative positions.

During playback the control is maintained by detecting pilot signals recorded along with the video signal. If the playback head reads only one signal then it is tracking correctly. If, however, it is mistracking it will sense two frequencies and an interference (or beat) frequency will occur. Thus if Head One is too high the error signal will be 47 kHz, and too low, 15 kHz. For Head Two too high the error signal will be 15 kHz and too low 47 kHz.

With this system the video heads will always be positioned correctly even with a still frame playback — in consequence a feature which VCC recorders excel at producing.

There is, though, a possibility that as the tape speed drops both heads will be lowered until they run out of their range. This is corrected by the Automatic Tracking Control (ATC) which, when it senses both heads mistracking, feeds a signal to the tape servo system to increase the linear tape speed.

Transports Of Delight

Electronically all these video cassette recorders are basically the same, their main differences being in the design of the tape transport; each format has adopted its own tape path and arguments continue about which is the best arrangement. For example, on Betamax recorders the tape remains wound around the video head drum at all times and the picture can still be viewed when the tape is being wound or rewound. The normal VHS deck has to unthread the tape for fast wind, rewind, and stop operations and this causes a tedious operating delay if you want to wind, check the picture, wind etc while looking for a particular portion. The latest generation of VHS machines can keep the tape against the head drum for cuing back and forth so the differences between these two formats are gradually becoming fewer.

The linear tape speeds are lowest on the Betamax (1.87 cm/sec), 2.34 cm/sec for VHS, and 2.44 cm/sec for the Philips VCC. Similarly there are differences in the writing speed, Betamax being 6.6 m/sec, VCC 5 m/sec, and VHS 4.85 m/sec. The linear speed is important to the fidelity of the soundtrack because the audio signal is recorded conventionally along a narrow track at one edge of the tape. As all three of these VCR formats have a linear speed of about half that of an ordinary audio cassette deck, the audio quality is for the most part pretty indifferent. A typical video recorder can have its audio performance compared to a low-cost cassette deck and still come out badly. Some video recorders now fit Dolby B, which is worth a 10 dB improvement on the signal-to-noise ratio, and Toshiba have a similar noise reduction system.

How Do VCRs Work?

The drawings show the block diagram arrangement of the video record and playback circuits. First of all it must be remembered that the recording process can only handle the wide bandwidth of the video luminance signal (DC to over 3 MHz) by using frequency modulation. The carrier signal frequency will vary with the amplitude of the video signal. Thus the peak white level may shift the carrier to 4 MHz, a black level to 3.3 MHz, and the sync pulses down to 3 MHz. This change in carrier frequency is referred to as 'deviation' and the total modulation is called 'modulation index' (M). Then

$$M = \frac{\text{deviation}}{\text{centre frequency}}$$

and for video recording will typically be 0.5. The FM will be passed through a low-pass filter to remove all components above the maximum deviation to give a band response as shown in Fig. 1. The process of frequency modulation is achieved by letting the luminance signal control the frequency of an oscillator whose output drives the recording head.

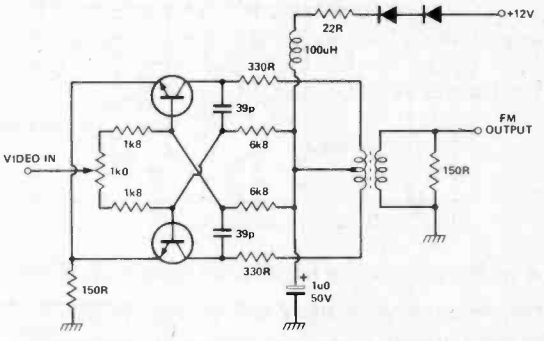


Fig. 4 A typical FM modulator used in a home VCR.

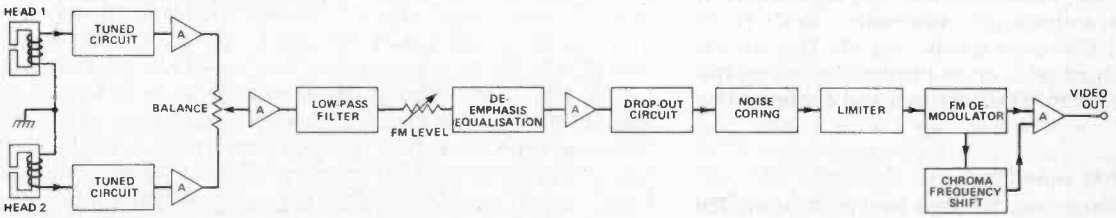


Fig. 5 Block diagram of a VCR playback system.

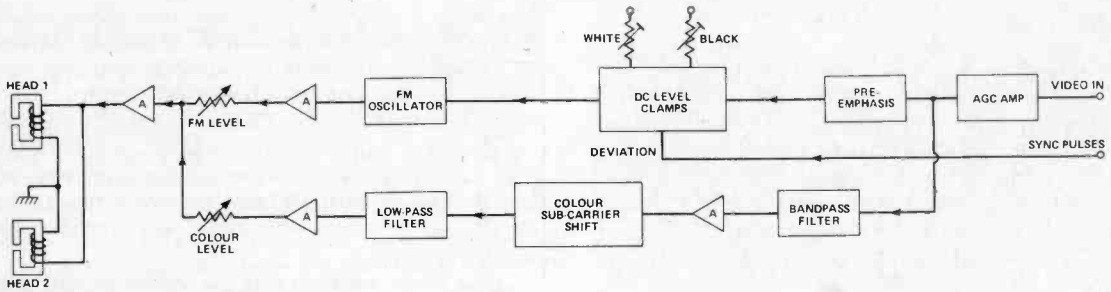


Fig. 6 Block diagram of a VCR recording system.

The chroma or colour signal doesn't modulate the same oscillator, even though on television signals it's modulated on a high-frequency sub-carrier (4.43 MHz). Instead the chroma signal modulates a low-frequency subcarrier of 750 kHz.

Two Heads Are Better Than One

If we now look at the block diagram we can see that on replay the output from the two video heads (alternately as they sweep across the tape) is first fed to a tuned circuit which resonates at about 5 MHz to peak up the frequency response which is falling off rapidly above 3 MHz. The output from the two heads is then balanced and passed through a low-pass filter to remove out-of-band noise, etc. De-emphasis follows to equalise the HF boost applied during recording. There then follow drop-out compensation and noise-reduction circuits which we will look at separately as they are of some interest. Then, after limiting to remove any AM components the signal can be demodulated and the chroma component frequency shifted to restore it. The chroma and luminance signals can then be mixed in a video amplifier to give a composite video output.

The recording process is almost the reverse with a slight variation. The chroma signal is frequency shifted but the luminance signal must have its maximum and minimum amplitudes defined by DC clamps before modulation in order to establish the maximum deviation. The two FM carriers are mixed at the output and fed to both of the video heads.

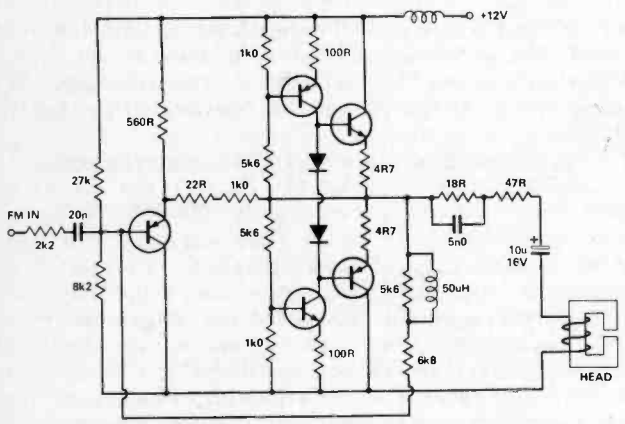


Fig. 7 A typical record drive amplifier as used in a JVC VCR.

Noise Coring

Video noise is an unavoidable result of the recording process, although it can be reduced with the best designs of video recorder and high-performance video tape. The effect of the noise is to make the picture grainy and hence lose the sharpness of lines and edges. To improve the subjective appearance of a picture, VCR manufacturers use a video-noise reduction circuit technique which is called noise coring. This works in the same way as a replay-only noise gate in audio. First, the high-frequency video signal is separated from the low freqencies. The HF signal is put through a clipper or limiter which removes the noise energy located on its average axis. The HF and LF signals are then recombined.

As with so much video circuitry there is virtually no value in drawing a circuit diagram of such a system, because it would consist of just three integrated circuits and a few resistors. For example, JVC use the 9V107 Filter/Amplifier, the SN7667 Limiter, and the VC2011 Mixing Amplifier/Buffer. The latest models use even fewer ICs!

If excessive coring is applied the picture will appear sharp but will also seem very unnatural, because much of the fine picture detail will be lost along with the noise.

Drop-out Compensator

The term drop-out is almost self-explanatory. When a segment of the tape has shed its oxide or has an embedded impurity then the recording will be interrupted and the signal will drop out. On the television screen these gaps are visible as random white lines that appear fleetingly on the screen. Get enough of them and they'll certainly ruin your viewing, so again the manufacturers have sought ways to minimise their effect. One technique is to substitute a picture line for the missing one but without an expensive memory this seems, at first glance, more than a little difficult.

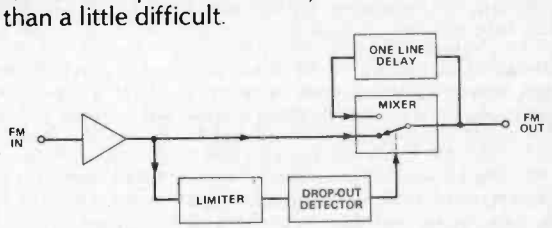


Fig. 8 Drop-out compensation.

However, as the drawing shows the usual circuit is quite simple. The FM signal is played back through a limiter (to maintain constant amplitude) and then fed to a drop-out detector which senses gaps in the signal. If a gap is found a DC control pulse is fed to the switch/mixer to disconnect the direct signal path and to connect instead the output of a 'one line' delay line. Thus the previous signal is substituted for the missing one. In this way the worst drop-outs remain unseen although a long term drop-out cannot be accommodated.

The Video Disc

Quite staggering sums have been spent by the electronics industry in developing and launching three competing video disc systems. It is seen as the Great White Hope for making billions of dollars of profits in the coming decade, although many industry observers feel there will be strong consumer resistance to a playback-only system. My favourite quote was from an RCA spokesman; "What's £200 million to a company like RCA"! The RCA system is called Selectavision and is made in the USA. From Philips (Holland)/Magnavox (USA)/ Pioneer (Japan) there is the LaserVision system and from JVC (Japan) there is VHD; so called because they haven't thought up a punchy trade name yet. It's no surprise that all three systems are totally incompatible and use completely different approaches.

Just Lasing Around

The Philips system is called LaserVision and is an optical system reading the signal encoded as pits on a reflective disc by means of a laser beam. This system was first launched under the Magnavox label (a subsidary of Philips) in America in late 1978. The system works well and the players have sold steadily but there have been continual problems with disc quality with (according to some observers) a 90% reject rate sometimes occurring. The current models use an expensive gas (neon) laser, although the design originally conceived the use of solid state lasers which will become available at a far lower cost — eventually, that is, so Philips are keeping an unusually low profile in their marketing, at least until they can make a worthwhile profit on the players. The difficulty is in manufacturing a solid state laser which has a wavelength short enough to focus on the very narrow signal track on the disc. Because the video signal is recorded in an analogue (not digital) form, it is not easy to correct the errors and ghosting which occur if parts of two adjacent tracks are simultaneously illuminated by the laser beam.

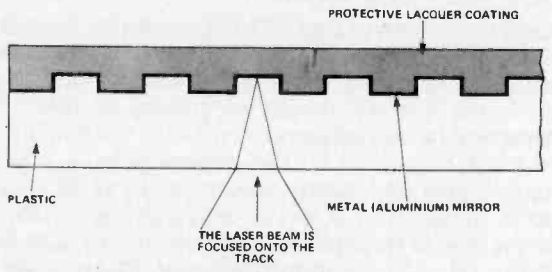


Fig. 9 A cross-section through the LaserVision disc.

Two types of disc can be viewed on the LaserVision player. These are CLV (Long Play) which gives a playback time of one hour per side; and CAV (Active Play), giving about 36 minutes per side. The long playing CLV disc keeps each video field the same length but increases the rotational speed as the 'groove lines' get nearer to the centre of the record. Thus the speed at the outer edge is about 500 RPM but by the centre of the disc thia has risen to 1500 RPM. The CAV disc is played with a constant motor speed but the length of the fields decreases closer to the centre as a consequence of the reducing diameter. This is the type of disc which makes LaserVision more interesting. The laser can move across the disc in 24 seconds, passing over some 54,000 separate television frames. Thus with the correct control mechanism the frames can be read one at a time (rather like a massive card index library), watched in fast or slow motion in either direction or be held in a 'still frame' mode — for months if necessary because in the absence of physical contact there is no wear on the disc.

The CLV discs will normally be used for recordings of entertainment because these 'special effects' are not possible. The rotational speed of the disc varies so that there is never a fixed number of picture frames in each rotation of the disc.

The Versatile VHD

The late-runner in the video disc contest is VHD (Video High Density or Video Home Disc) system developed by JVC in Japan and backed in the UK by the Thorn-EMI group. This system is very similar to both LaserVision and Selectavision but uses a 10" diameter disc instead of one of 12". The disc (which plays for one hour each side) is pressed from conductive plastic, with the signal pressed in as raised and lowered patterns but no groove as such. The signal is read by a capacitive pickup which follows the spiral track by using a sort of parallel tracking servo-controlled arm. The VHD disc suffers some wear in use because the tracking stylus slides over its surface and so the repetitive play of, say, a still frame could, as with Selectavision, shorten the disc's life.

As it stands the VHD system is not ideal for the playback of still-frame pictures. Each rotation of the disc holds in its signal track two full television frames ie four interlaced fields. Thus if the same track is scanned repeatedly there will be some visible 'judder' of the picture as the two different frames alternate. Two solutions exist. The first is to feed the frame into a digital frame store where it is converted from analogue to digital form and loaded into a memory. The 'frozen' picture can then by continually readout from the memory, converted back to an analogue video signal and fed to the television to give a perfect still frame.

This is the approach the television companies take and it's an ideal approach except for one thing — the video frame stores cost £17,000 and upwards, hardly suitable for fitting inside a £350 video disc player. So JVC have adopted the somewhat more pragmatic approach of recording each TV frame twice, so that for every rotation of the disc only one picture is seen. Now such a doubling-up will mean that the programme will be viewed in slow-motion. Solution? Easy — just double the rotation speed and accept that these discs (Type II) will only run for 30 minutes a side. If we put a three hour blockbuster movie on to these discs we will need three, viewing all six sides, so the early years of video disc may resemble the days when a complete opera could only be heard by playing a stack of 78 RPM records! Already there is talk in Japan of an 'autochanger' video disc player — the juke-box of the future.

The RCA Selectavision Video Disc System

This system uses a flat circular disc of 12" diameter which has the television sound and picture signals recorded on a spiral groove rather like an audio disc. However, there are 10,000 grooves per inch (compared to 250 on a audio disc) and the information is recorded as frequency-modulated vertical undulations of the V-shaped groove. The plastic disc contains a fine carbon dust to make it conductive and is covered with a film of oil to lubricate the playback stylus and so increase the

VIDEO DISC MANUFACTURE

Of all the video disc types, probably the hardest to manufacture is the optical disc used in the LaserVision system.

The original video programme is recorded onto a professional video tape recorder using 1" or 2" tape. It is played back on to the disc 'cutting lathe' where a high-power laser beam tracks a modulated light beam over an ultra-smooth glass disc coated with a photo-resist material. The exposed disc is chemically developed, etched, and washed to leave a visible spiral of pits etched into the glass surface. The glass is then coated with a fine coating of silver to form a conductive layer, which also allows the disc to be played as a quality check without damage (there is no physical contact with the disc's treated surface).

The disc is then plated with nickel, followed by a layer of aluminium. The glass master is then removed (and damaged beyond further use) to leave a negative which is referred to as the 'father'. More nickel is electro-deposited onto the record side of the father to produce a positive 'mother', from which a number of nickel negative stampers can be grown again by electroplating.

The video discs proper start off as a blank sheet of 1.3 mm acrylic (Perspex) which are thoroughly washed and then coated with a 30 micron thick layer of photosensitive lacquer. The disc is then gently pressed against one of the negative stampers to give the spiral track of indentations; they can then be exposed to the ultraviolet light which hardens the photo-resist. The discs are then loaded into large vacuum chambers where they are immersed in an aluminium vapour for about 30 minutes. This vapour causes a very fine reflective coating to be deposited on the disc; a coating which is then protected by a layer of clear lacquer.

So far one disc side has been produced, so it is glued to another side and the final two-sided disc is balanced electronically to ensure stable rotation in the player.

The entire process is semi-automatic up to the last important stage — final inspection. At present the discs are checked by actually playing them with an operator watching the programme on a television screen. The inspector checks four discs at a time (four screens) in what must be one of the most boring jobs of all time. However it has not yet proved possible to automate this process or to rely upon fast playback during inspection.

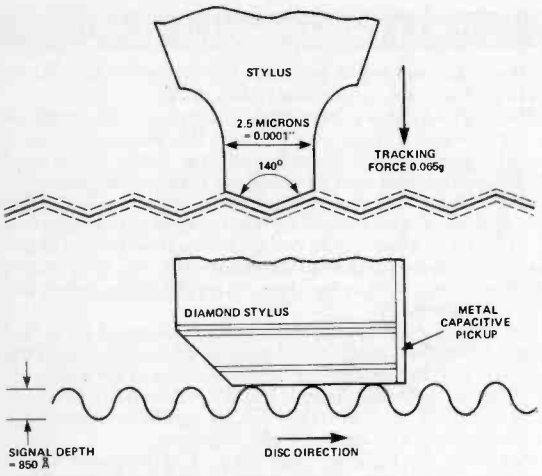


Fig. 10 RCA Selectavision groove geometry.

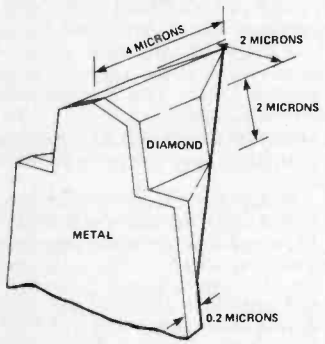


Fig. 11 The RCA stylus.

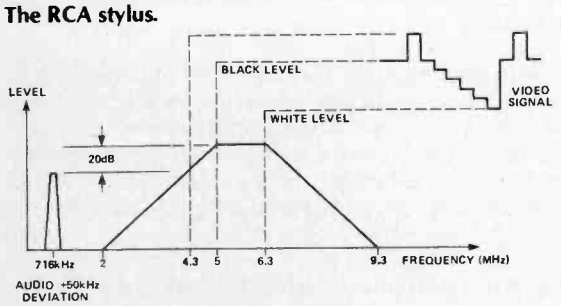


Fig. 12 The RCA video disc signals.

life of the disc. The stylus runs in the groove but actually senses the signal through the use of capacitive coupling between the stylus and the disc — each being one plate of a capacitor. The stylus is long enough to ride over the signal peaks pressed into the groove, so the disc surfaces rises and falls under the stylus electrode giving a capacitance variation of about 1×10^{-4} pF peak-to-peak. This almost insignificant change in capacitance is what constitutes the output signal.

The capacitance is made part of a 910 MHz resonant circuit which is fed with a signal from a 915 MHz oscillator; a frequency which falls at the half-amplitude point on the resonant curve. As the disc-stylus capacitance varies, so does the resonant frequency and the amplitude of the 915 MHz oscillator output signal. Thus over a period of time the 915 MHz signal is amplitude-modulated by the disc signal which can be simply recovered using a diode detector. The output signal is, in fact, two frequency-modulated carriers, these being 716 kHz (audio) and 5 MHz video. These carriers are fed through limiter amplifiers to take care of the 20 dB or so of level variations that occur, and the constant amplitude signal is fed to phase-locked loops for demodulation. The remaining circuitry contains quite complex arrangements for reconstituting the composite video signal and others that detect and compensate for playback errors.

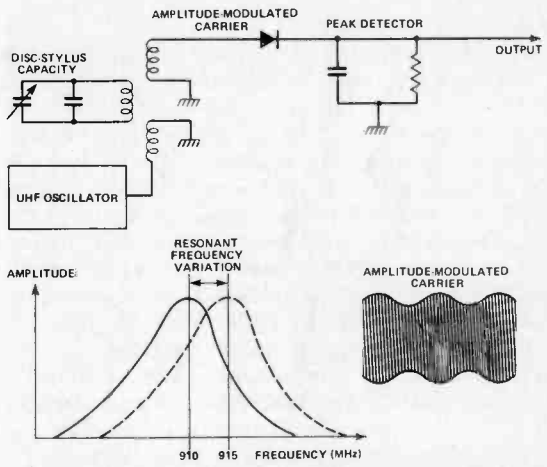


Fig. 13 The Selectavision playback demodulation system.

Are You Being Servoed?

Early prototypes of the CED disc system (as it was known) used an ultra-lightweight tone-arm which supported the capacitive pickup in a conventional record player fashion. However the current design of pickup is quite heavy, incorporating the high frequency resonators and amplifiers; and so it cannot be guided by the extremely small side forces generated by the microscopic groove walls. For this reason the pickup is mounted in a servo-controlled arm which tracks across the disc in response to the stylus motion (see Fig. 14). Obviously no servo or gear-train could follow every small movement of the stylus so the pickup is allowed some 2 mils of free motion.

As the drawing shows, a conducting 'flylead' is positioned in-line with the stylus and its position is detected by two sensors, one to each side. The two sensors are varactors whose capacitance is modulated by a 260 kHz oscillator. Each is of opposite polarity, so with the flylead absolutely central the capacitively coupled 260 kHz variations cancel out. An offset in position will cause a 260 kHz component to be detected by the flylead and result in an error signal being sent to the arm motor which will reposition the arm. The use of these opposing sensors largely cancels out most of the temperature variations and provides a very stable electrical centre.

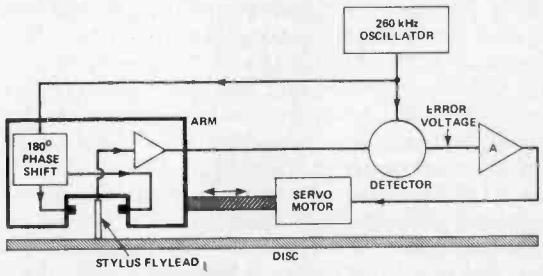


Fig. 14 The stylus position servo system.

Looking Ahead

Finally, the future. Matsushita and Pioneer (with some work by RCA) have produced optical video disc systems which can record programmes (or data) and subsequently replay them. During recording the laser works at full power and burns into the surface of a blank disc; when switched to low power it reads it back LaserVision-fashion. The Matsushita system puts 15,000 still pictures on to an 8" disc and although that represents only five minutes or so of a video programme, it will only be a short step to a complete optical record/playback disc. Meanwhile Sharp and Matsushita (Panasonic) are working on a magnetic disc recorder which also uses a laser, but in this case it alters the magnetic characteristics of the disc coating.

COMPUTER-CONTROLLED LIVE MUSIC

Computers have been making inroads into the music business for some time now, but Peter Finbarr-Smith shows how even small groups can take the whole concept much further.

Music has always been popularly regarded as a purely aesthetic activity, with the musician condescending to operate the instrument simply because there had to be some way to make his music audible to the lesser creatures who provided his meals! Along with this idea goes the thoughts of some 'purists' that there is no place at all for electronics in 'serious' music. On the practical side, however, it's hard to beat a craftily applied wah-wah pedal as an emotional shifter, just as Segovia (or John Williams) can make strong matrons weep (or so I'm told), by wiggling a finger on a guitar string!

Of course, the answer lies in technique — a method of producing a specific effect in a listener by means of a trick, whether mechanical or electronic. Nevertheless, the 'old school' make a hasty distinction and throw their hands up in horror when a musician 'wiggles his finger' electronically! But whether the old school like it or not, they cannot halt evolution, in music or in anything else, and with the microprocessor very firmly with us, evolution will accelerate.

Whoever it was who said that the microcomputer would affect all of society was guilty of understatement. With some of the 'big name' groups turning to total computer-control of live shows it would seem that, in the near future, if you haven't got a

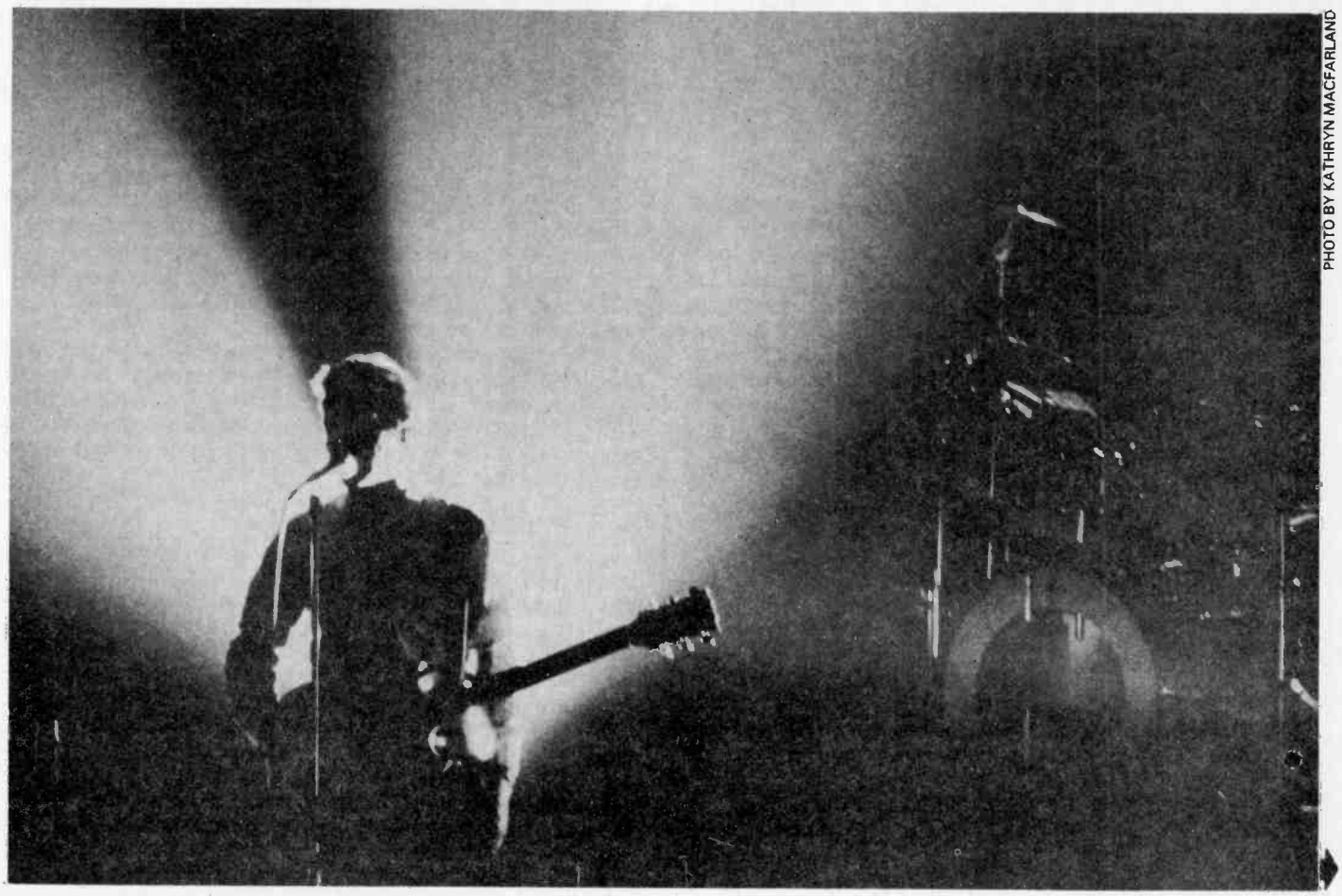


PHOTO BY KATHRYN MACFARLAND

microprocessor inside your guitar then you might as well not bother with strings either! In fact, while it may not quite come to that in the immediate future, computer-control (of the equipment, not the music) is neither particularly difficult nor, these days, incredibly costly. Anyone owning one of the many brands of 'personal' computers can start to think seriously of some of the possibilities of equipment control, plus, perhaps, computer generated sound effects.

This article is intended to show the scope of computer-control when applied to amplifiers and auxiliary equipment such as mixers, echo-units and (for the really advanced!) synthesisers and to show how it is done. At least, some thinking about new possibilities in music should be sparked-off.

Programmed Pots

To begin with, a computer can control volume, treble and bass, plus any other controls, on any or all amplifiers, plus parameters on other equipment used in a live performance or in a recording studio, with an accuracy which is unrepeatable by manual means. The computer can be small and easily carried, with a minimum of setting-up. In fact, once it has been programmed, setting-up is very much faster than before; probably the only adjustment necessary will be for the public address level, which is likely to be affected by the acoustics of the venue.

Although rehearsals, often conducted at a leisurely pace, can allow careful adjustment of equipment for special effects during a number on stage, there is usually very little time to make careful adjustments — especially when there are full mixers and synthesisers to take care of. The actual results are often a compromise, leaving the musicians disappointed, and the roadies (yet again) accused of sabotaging a carefully rehearsed effect. A computer, however, can re-set the whole lot quicker than you can say 'you're sacked', and can give results identical to those obtained at rehearsals. Apart from the obvious benefits of this arrangement, it is unbeatable for live recording as well saving a lot of time 'on the night'. Lights can be controlled, not only for on-off, but also for the exact level of brightness required for a difficult colour mix. Of course, once it's in the computer's memory, it never changes — although it can be changed very easily when necessary.

Data Chains

Here's how it's done. For convenience, we shall consider a set-up consisting only of amplifiers. The explanation also applies to the use of other equipment. The computer to be used for the system is set to the mode which would normally pass information to a cassette for storage, or to a printer. But now these items are replaced by the amplifiers. The computer doesn't know the difference (not so smart, eh?), and so is quite willing to send data (information) this way, even if it seems to go nowhere at all! Figure 1 shows how the system is connected, both for a small to medium set-up (a) and for a large system, which might also include such items as recording equipment and lights (b).

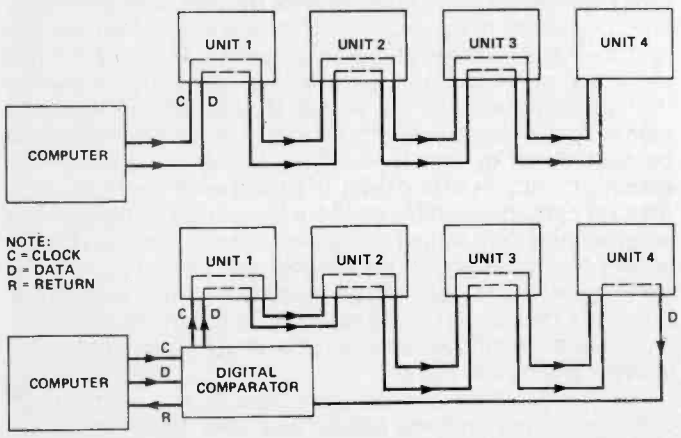


Fig. 1 How the computer control lines are wired for a small system (a), and a larger set-up (b).

A cable is connected at one end to the computer's tape or printer socket, its other end going to a special socket on the back of the first amplifier in the 'chain'. Inside the amplifier the data goes to a decoder/command circuit. The data is also 'beefed-up' a bit and sent on its way out of a second socket via a similar cable to the next amplifier in the chain. In this way the data is sent to each of the amplifiers, and ends up in the last unit in the chain. Other units may be added at any time by simply joining them onto the chain.

For larger systems, another gadget is added into the chain. This is called a 'digital comparator' and both the computer output and the last amplifier in the chain connect to it. This device now issues the data to the units, and there are no prizes for guessing what it does! It takes a sample of the data leaving the computer and compares it with the data coming back from the last unit. If there is any difference between the two (usually caused by a broken cable or a disconnected plug) then a red light illuminates on the comparator's front panel. Computers using the printer socket for data output will halt the program until the fault has been fixed. The comparator usually gives its warning during set-up, with the lamp lit until the whole system has been connected, as faults of this nature rarely happen during a performance.

It will be seen from Fig. 1 that there is a second cable: the clock line. The cable comprises two wires, data and clock, inside a screen (rather like a thin guitar lead) and the screen keeps interference out, as well as preventing the fast computer signals from radiating electrical noise. The clock line is used to synchronise the data signals, and will be described later in the article.

Inside Story

Figure 2 shows a block diagram of a computer-controlled amplifier. It will be noted, after a quick glance, that the main amplifier is not actually connected in any way to the special electronics. The keen do-it-yourselfer will immediately realise that if he were to obtain the decoder/command board, then a slave amplifier would complete the unit! Quite correct. It should not be long before such boards are available, and normal amplifiers could then also be modified, possibly with a switch marked 'CC' (Computer Control) and 'N' (Normal), fitted to the front panel of the amplifier. However, a commercial product would either present the decoder/command board in a 'black box' for connection to a slave amplifier, or would consist of a unit combining both the board and an amplifier.

Getting back to what happens inside, it will be seen in Fig. 2 that the clock and data lines enter into and exit from the data receiver. In fact only the data enters the receiver, the clock being used to merely operate the input gate, ensuring that each bit of the data word enters the receiver at the correct time. A bit is either a '1' or a '0', and (usually) eight of these form a data word. More about that later!

As the amplifiers are in a chain, with each passing data to the next, then there has to be some means by which each unit 'knows' which commands are intended for it. The decoder takes care of this by looking at the first word in a group of words sent from the computer. If the correct address (the unit's personal name) is represented by the word, then that unit will accept the commands which follow immediately. As the data flows around the amplifier chain at the speed of light, it can be said that all units receive the data at the same time. The actual delay during the 'passing on' process occupies only millionths of a second.

Having accepted the first data word as the correct address, the receiver then passes each of the following words to the registers associated with each control on the amplifier. The registers are small memories capable of holding a single word (whereas the main computer memory can hold thousands of

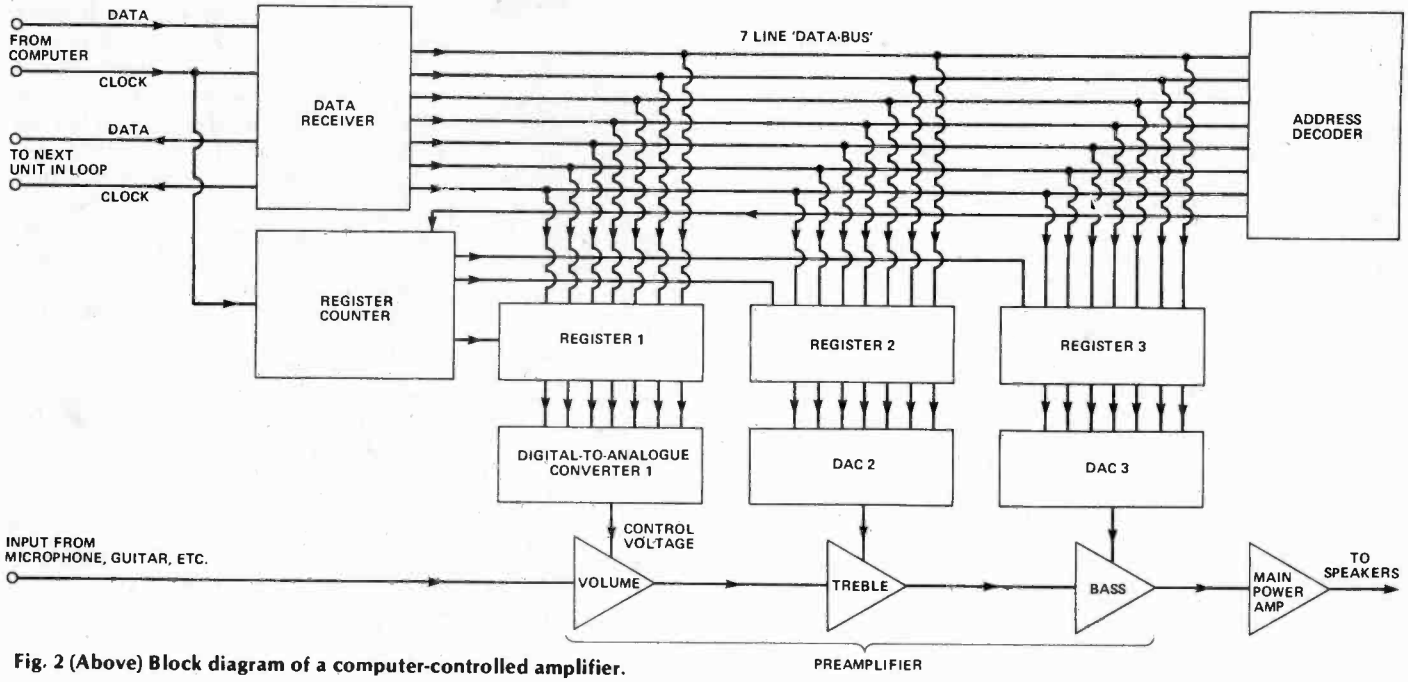


Fig. 2 (Above) Block diagram of a computer-controlled amplifier.

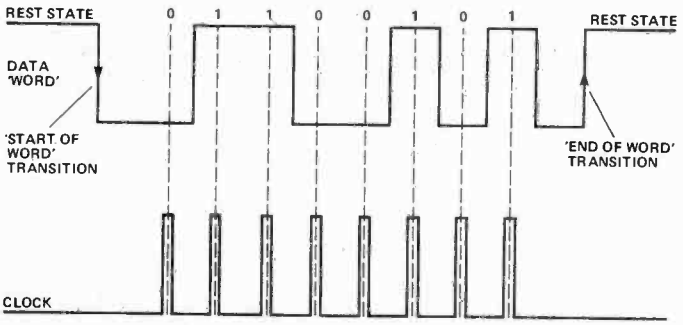


Fig. 3 Typical data and clock signals when an eight bit word is being transmitted by the system. The data is clocked in to the registers by the pulses in the clock signal, shown here as a separate waveform. Some systems may combine the clock and data into one signal.

words). Each word stored in the registers — consisting of eight bits, either 1s or 0s — is then converted into a voltage. The voltage is then applied to the appropriate part of the preamplifier, Volume, Treble or Bass, and these 'controls' are set until the computer sends the next batch of data. This usually happens at the end of a 'number', unless the controls need to be altered during the number for special effects. Some of these effects can be really startling, as when the computer has control of both tone and volume, an instrument can be made to virtually 'talk'! In other situations, the controls on a particular amplifier may not need alteration for extended periods. One obvious occasion when alteration of volume is needed is when a guitar changes from rhythm to lead, for example, when it may be necessary to go into the melody fast, without wasting time twiddling knobs!

Hi, Lo, Hi, Hi, Lo . . . . .

Having looked briefly at what goes on inside the amplifier, we shall now take a look at the clock and data signals. These are shown in Fig. 3. When the signal is high, it is said to be a 1, when it is low it is said to be a 0. The data line is normally high when in the rest state, to minimise the risk of interference producing false data. The level then changes to low before the word begins, then follows a series of 1s and 0s (usually eight of them) before the data line returns high, signifiying the end of the word. The highs and lows of the word tend to flow together if there are more than one of either type together, so the clock is used to 'chop' them into something recognisable by the decoder. In this way, the data is said to be 'clocked in'. In Fig. 3 the clock is separate from the data for this purpose, but in some systems the clock is 'buried' in the data, to be separated in the unit being controlled. Other systems use the change from the rest state into the active state as a signal to the unit to start its own clock which then clocks the data into the receiver. The choice of method seems to depend on the manufacturer's individual preference.

So now that we all know how it works, let's take a look at the programming, without which the system is useless.

Programming

This job logically falls to a roadie. There is much said about roadies, but with the musicians all busy, who else is there? Joking aside, this isn't such a bad thing, as many of the 'gear humpers' are of exactly the right temperament to do an efficient job on the programming. It goes like this: the equipment having been set up for practice, the various controls need to be set. As the controls are now electronic (no, they're not pots driven by motors!), they have to be set up from the computer keyboard. First the address of the unit in question is given by the programmer, and then the various controls are set, one by one. During this process, the computer is programming itself in answer to the questions and, if the set-up is satisfactory, then this represents the repertoire-entry for that number. Alterations can be carried out by simply recalling a unit's address, and re-entering the set-up information. In a number of cases the levels used for practice are different from those used on a gig, so the programming may actually be done in silence, based on live experience. Normally a 'frame' program will be used, which has been prepared in advance, and which asks the appropriate questions, thus considerably speeding up the whole process. A typical example of questions (and answers) seen on a computer monitor is shown in Fig. 4.

In this example, the system is capable of controlling 26 units (amplifiers, mixers, lights, etc), and these are given alphabetic addresses for convenience. In fact, the number of unique combinations in an eight-bit data word is 256, so there is

no reason why a system should not have numbered units — up to 256, if necessary! Ideas for multi-group outdoor concerts arise here! The numbers following the questions relate to the settings of the controls. Each control is capable of 256 positions (corresponding to a knob with 256 marked positions, if it were not electronic). It is obvious that the precision and degree of control setting is far superior to the best efforts of us mere mortals!

From a programming point of view, a mixer is considered to be a number of amplifiers, one for each channel, and something like an echo unit has its own questions on the screen, relating to individual controls such as reverb levels for example. The same applies to the special controls on a mixer (foldback, etc). The reader may now be wondering how the computer knows what controls are available on what equipment. It knows because the frame program has been specially written to suit the group's equipment. Software (programs) has to be prepared for any serious use of a computer, but given the ground rules, the software is usually produced by a member of the group, unless a 'package' has been bought from a company specialising in machine-control (as this branch of computing is called). This is, of course, expensive and while a dozen or so of the top name groups might invest to this extent, there will be hundreds who are prepared to try it themselves, with varying degrees of success.

Minding The Store

At rehearsal, as the gear is set up, the computer is busily programming itself. Therefore, when the group are satisfied with the performance of the equipment for that number, it is held permanently within the computer's memory, along with its title. As different numbers are practised, so a repertoire is built up. In fact this process, coupled with false starts and other problems, may take quite a long time to complete; but as with all such things, when the computer is switched off it promptly forgets everything — losing the lot! So there has to be some way of storing the information for later use.

On the smallest systems this can be done on cassette. For computers with only a small memory, a full repertoire may have to be stored in more than one section, each section being loaded during a gig — possibly in the interval, for the second section. Alternatively, the particular show may have to be assembled in advance, with just the required numbers being stored in the computer before the show starts. Each new set-up for all or just some of the equipment is sent out by pressing two keys on the computer keyboard, either between numbers, or during them, for special effects.

For larger systems, especially those being used for live recordings, a floppy disc unit is necessary, rather than a cassette recorder, as the great storage capacity and high speed of a disc unit means that a large repertoire may be stored on one disc and any part of it called up almost instantaneously.

Before the group begin to play the next number, the operator simply calls the list of titles on to the screen, and a question at the bottom of the list asks: WHICH NUMBER DO YOU WANT? In answer to this, the choice is keyed-in, and the system is immediately programmed for that item. The number chosen may require a change to some (or all) of the gear during the actual performance. The operator simply presses two keys, and the re-programming is done instantly while the group are still playing. Too simple? Well, he's got to press the keys at the right time, naturally!

Getting Started

At the present time there is very little commercial equipment available for the computerisation of sound equipment. There is a reasonable amount of studio gear around, but it is only applicable to live recordings, and does not help the group. There is, however, some equipment in the pipeline, and we are told that this will take three distinct forms:

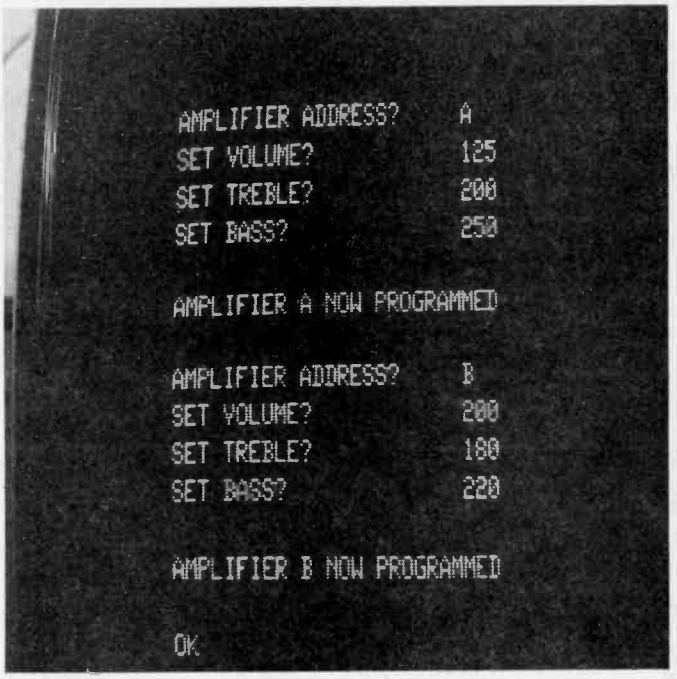


Fig. 4 Typical screen readout during programming.

- The 'bare board' type of computer-controllable preamplifier, which can drive slave amplifiers directly or can be fitted inside a normal amplifier.
- The above preamp inside a case, complete with power supplies and ready for use with slave amplifiers.
- The complete computerised amplifier in several output wattages: an almost featureless black box with no controls at all, except a power-on lamp and an instrument input socket in the front, with the two computer sockets on the back.

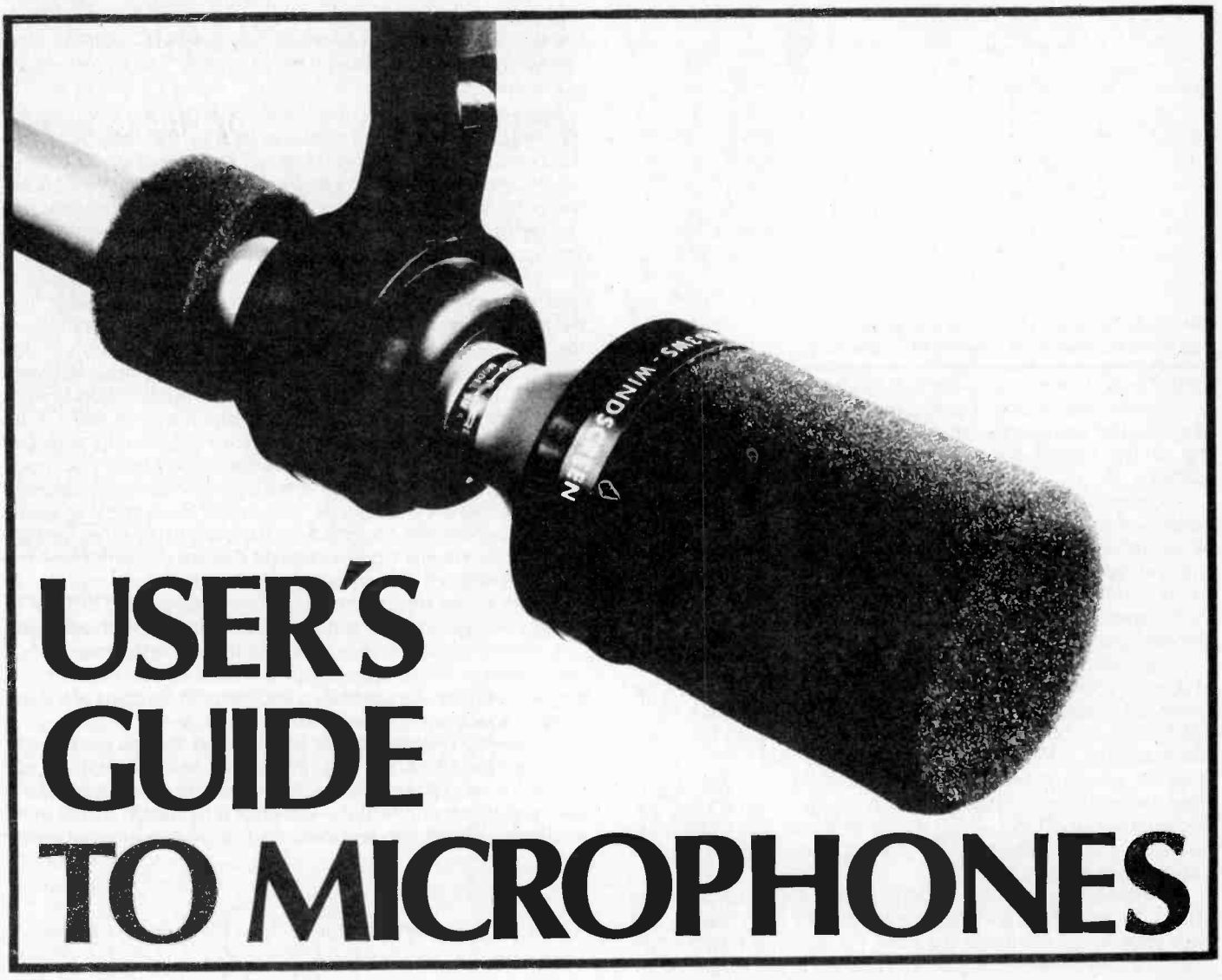
Other expected equipment may take longer to appear, such as echo units, effects generators, lighting control units, mixers, and a synthesiser with no controls. Needless to say, the latter item will not be cheap!

So do we sit back and wait? Well, if the preamplifier boards are going to be available reasonably soon, then we still need a computer, and some time to learn to program it, so perhaps that should be the first step. Computers such as PET, Apple, TRS-80 and so on, are reasonably priced and are potentially suitable, but even if there is a printer socket (there will certainly be a cassette socket) available, we still need to add a clock line. This is achieved by finding the point inside the computer where the system clock enters the printer or cassette circuitry, and using gates and a counter (three integrated circuits needed), the correct number of clock pulses can be sent along with the printer data (from a spare pin on the printer socket).

If the chosen computer has only a cassette socket, then the modification is still not too difficult. Having found the clock, we must now find the data where it enters the cassette circuit. This is brought out in the same way as the clock, or, ideally, to a new socket fitted for the purpose, to the computer. And we're ready for business! The foregoing is not as difficult as it may sound, but it certainly takes guts to stand by and watch a roadie disembowel a new microcomputer! For those who can't face that, take heart. There is at least one computer specially designed for the job already available, ready to plug in and run.

Conclusion

In the musical field as much as in any other, the microcomputer will make a powerful contribution toward evolution. Its applications are limited only by the imagination of the user. As the machines become lower in price (because everyone in Britain except you has already got three of the things), we should be seeing simple guitar tuning aids and so on, being run by computers almost powerful enough to take man to the moon, and with sufficient memory to catalogue the British Museum.



USER'S GUIDE TO MICROPHONES

So you think you know about microphones? After reading this article by Vivian Capel you will !

Along with the loudspeaker, the microphone is the most often used transducer in the world. In broadcasting, recording, noise measurement and communications; they come in all sorts and sizes. Millions are in use at any given moment. Yet unlike the loudspeaker, they are little understood except by the professional user. As with any other component, appreciation of the working principles and operation helps considerably in the choice of a suitable unit for a particular purpose and its subsequent utilisation. This article will explore these factors with a view to making the most suitable choice.

Transducer Types

There is more than one way of converting sound into electrical energy, and most of the practical ones have been used in various microphone designs. Some of these are as follows:

Carbon: Carbon granules are packed between two plates, often also made of carbon, one of which is fixed and the other moveable. The moving one is linked to a diaphragm so that sound pressure waves acting on it alternately compress and release the granules. This produces a variation of resistance across the plates through the granules. A current is passed through from an external DC source, which varies according to the resistance fluctuations.

An output voltage can be obtained by a resistance/capacity coupling circuit or by means of a transformer. The transformer primary, DC source and microphone are connected in series, so that an AC output signal appears across the secondary.

The advantage of the carbon microphone is high output (it can drive an earpiece direct with no amplification) and it is also extremely robust. Disadvantages are poor frequency response due to the inertia of the moving system, non-linearity (and hence distortion) and noise produced by the granules rubbing over each other. Uses are therefore confined at present to telephone and other communications.

Crystal: Various natural and man-made substances such as rochelle salt (sodium potassium tartrate), quartz, tourmaline, barium titanate and lead zirconate titanate, exhibit the piezoelectric effect — they generate a voltage when subject to stress or strain. The basic crystal microphone consists of a thin

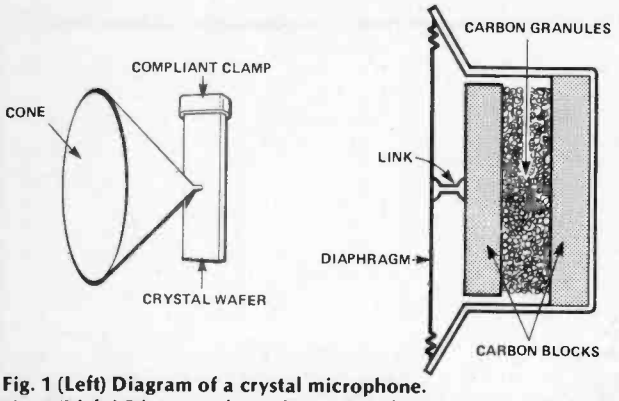


Fig. 1 (Left) Diagram of a crystal microphone.
Fig. 2 (Right) Diagram of a carbon microphone.

wafer of crystal secured at one side but free at the other. A cone is mounted with its apex bearing against the wafer near its free side and the sound pressure variations cause sympathetic flexing of the crystal and the production of a corresponding voltage.

Variations of this are the bimorph, which has two wafers cemented together to give a push-pull effect, whereby one is stretched while the other compressed; this gives twice the output and partial cancellation of non-linearities. There is also the multimorph, or sound-cell, which has several crystal elements and dispenses with the cone; the sound activating the crystals directly. Output is lower, but cone resonance is eliminated.

The advantage of the crystal unit is that of a high output voltage without an external DC source; it is also relatively inexpensive. Disadvantages are its fragility; the wafers crack easily with physical shock and they are affected by humidity. Man-made crystals are rather better than the natural ones in these respects. Frequency response is uneven and poor at the top end due to inertia effects. The main disadvantage is the very high impedance (around 1M0) which results in severe loss of high frequencies over comparatively short screened cables due to capacitance effects.

When used (with short leads) to drive valve input circuits in cheap PA amplifiers and tape recorders, the high output and high impedance eliminate the need for an input transformer. They are also used to measure vibration in some industrial applications.

Moving-Iron: A steel disc is supported over the pole-pieces of a permanent magnet on which are wound a pair of coils. It vibrates as a result of the sound pressure wave and so the spacing between the disc and the pole-pieces varies. The magnetic field also varies therefore and the flux lines cut the windings, inducing an EMF. Variations include balanced and rocking armature types.

Advantages are robustness and that units can be made very small if required. They can also be used for machinery vibration tests which would destroy other types of transducer. Disadvantages are poor frequency response, pronounced resonance of the disc and distortion caused by unequal flux variation. They are rarely used other than for vibration or sub-miniature applications.

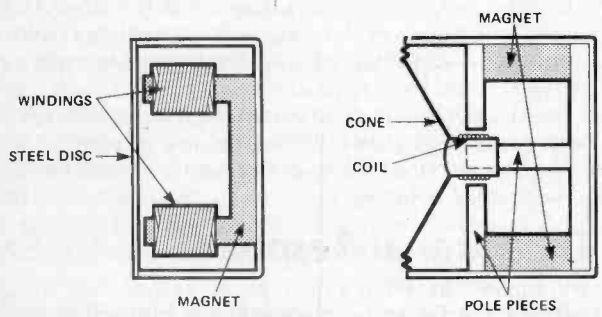


Fig. 3 (Left) Diagram of a moving-iron, or magnetic, microphone.

Fig. 4 (Right) Diagram of a moving-coil microphone.

Moving-coil: One of the most common types of transducer, the moving-coil microphone works in exactly the same way as a loudspeaker, only in reverse. The coil at the apex of the cone or diaphragm is immersed in a magnetic field between the concentric poles of a permanent magnet. Movement of the cone and coil induces current into the windings which is proportional to the sound pressure acting on the cone.

The advantages are high quality signals with extended frequency response and the ability to withstand hard usage. It is not easily overloaded by very loud sounds. The disadvantage is the comparatively high mass of the cone/coil assembly which puts a resonance peak in the frequency response between 2-5 kHz. This is a particular drawback for public address work where any resonance in the microphone response will initiate early acoustic feedback. With some microphones the peak can be 5-6 dB or even more, but the better models have it damped to around 2-3 dB. Double-transducer models are available, although these are expensive. They work like a speaker with a woofer and tweeter, one handling the low and the other the high frequencies. An electronic crossover ensures that only those frequencies within the designed range of each unit are supplied to the output. The resonant peaks of each are pushed outside of the pass range, so a flat response free from peaks is achieved.

With some applications the moving-coil resonance peak is considered an advantage, and gives what is sometimes termed a 'presence effect'. As the peak coincides with that part of the frequency spectrum at which most ambient noise occurs, it gives such ambience emphasis. So, background sounds are brought into on-the-spot recordings, interviews and the like to add realism. It can also brighten some instruments such as the piano and render certain vocalists with poor articulation a mite more intelligible. It will also emphasize a lisp, and give strings a wiry tone.

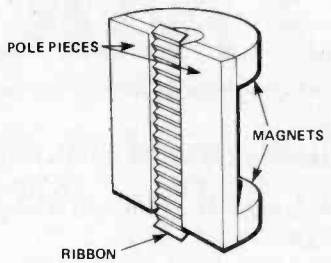


Fig. 5 Diagram of a ribbon microphone.

Ribbon: A corrugated ribbon is suspended edgeways between the pole pieces of a magnet so that as the air molecules impinge upon it, it moves back and forth and generates a current. In effect it is a single-turn moving coil without a cone. The impedance is very low, around 0.1 ohms and so a built-in matching transformer is required. The area affected by the sound wave is small and so the output is also low. Some increase in efficiency is obtained in some models by mounting a small parabolic wave-guide in front of the ribbon like a minature horn.

The advantage is that the ribbon has a very low mass, resulting in an extended high-frequency response and excellent response to transients. The resonance is high, near or beyond the normal range, so that the response is smooth over the normal audio spectrum. This makes it well suited for public address use. It has a warm, natural tone.

The low output is not a major disadvantage as modern designs give little less than many moving-coils and most input circuits possess sufficient sensitivity. Ribbons are rather fragile,

though, and can be damaged by shock or by blowing into them (never 'test' any microphone by this means). They tend to be expensive but not more so than the better moving-coils. Bass is over-emphasized when used close to the lips and the explosive consonants p and b can have a shattering effect. Pop shields of foam rubber or other material are frequently used to minimise this.

Note that not all ribbon microphones are free from resonances. Some have a peak designed-in to satisfy the demand from some quarters for a presence effect!

Capacitor: The value of a capacitor depends in part on the distance between the plates. If one is fixed but the other can be moved by air pressure variations, the distance (and hence the capacitance) will be varied. Capacitor microphones consist of a plastic metal-coated diaphragm stretched over a circular shallow cavity in a metal casing. The back of the cavity serves as the fixed plate, while the diaphragm is the moving one. Aluminium or gold is used for the metal coating, and the diaphragm is sometimes embossed to give a required degree of elasticity.

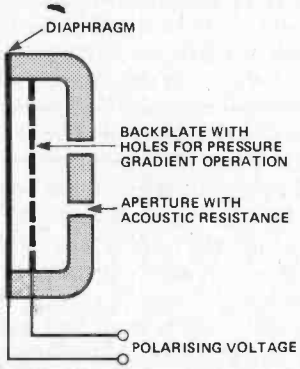


Fig. 6 Diagram of a capacitor microphone.

As the capacitance varies, current from an external source flows in and out of the device, and a signal voltage is developed over a series resistor. The applied voltage tensions the diaphragm, imparting rigidity, and so enabling it to move accurately in response to the sound pressure variations.

The diaphragm has a low mass which gives the advantage of a flat and extended frequency response, free from resonance peaks over its range, and an excellent response to transients.

A disadvantage is the need of a high polarising voltage, usually about 50 V. The capacitance of the transducer is small, around 20-30pF, and so are the variations. Thus the high voltage is required to produce usable output fluctuations. This means that the device has a very high impedance which would cause excessive HF loss over even a few inches of screened cable. To overcome this the preamp is included in the microphone case, receiving its power from the polarising supply.

This supply must be incorporated in the equipment input circuits; a means for conducting the current must also be provided through the cable. This can take the form of 'phantom' powering, in which one pole of the supply is conveyed along the signal wires through a real or artifical centre tap to the input transformer, and the other pole along the braiding. Another

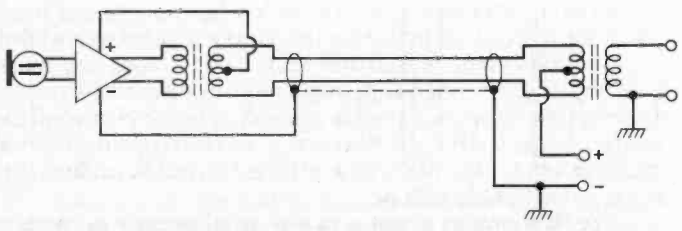


Fig. 7 Phantom powering of a capacitor mike. The supply is sent via the centre tap on the transformers and the screening.

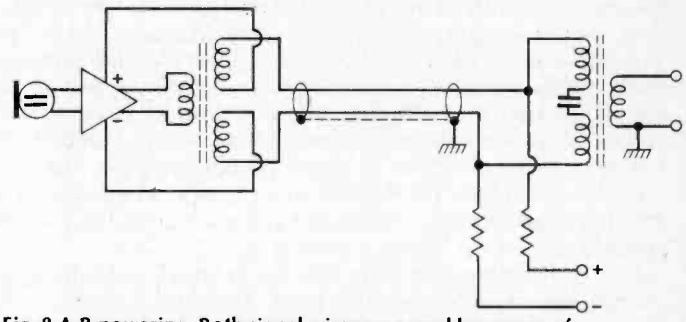


Fig. 8 A-B powering. Both signal wires ares used by means of capacitively-coupled split transformer windings. This avoids using the screen to conduct current.

method is A-B powering, where the two supply poles are carried by the two signal wires — these are DC-isolated from each other by capacitors between split windings on the internal output and equipment input transformers.

A further disadvantage is that the microphones and their ancillary equipment are expensive. This is why they are usually found only in broadcast and recording studios, or other professional applications.

There is another method of utilising the advantages of the capacitor principle without requiring polarising voltages; this is by using it to tune a small FM radio frequency oscillator and demodulator. The capacitance variations produce frequency modulations which are demodulated to produce an AF output signal. This system is used with Sennheiser capacitor microphones.

Electret: A comparative newcomer to the scene, the electret is a capacitor microphone but with the polarising charge permanently implanted in the diaphragm. The diaphragm is placed between the plates of an air-spaced capacitor which is charged from a high-voltage source. The diaphragm is heated and then allowed to cool, whereupon the charge is fixed and permanent.

It needs to be thicker than the normal capacitor microphone, which increases its mass and curtails its transient and frequency response. An internal preamp is still required, but this can be a simple two-transistor design operating from a single internal 1V5 cell.

The advantages are a smooth extended response which, although not to the standard of powered capacitor models, is very good and better than most moving-coils. The diaphragm resonance is high (typically 8-10 kHz on the better models), and unlike the mid-range resonance of the moving-coil, this can be tamed with a little treble cut on the tone control. Electrets are light and can be made quite small, making them ideal for the tie-clip units which are favoured by many speakers and entertainers. In this case the battery cell can be a pen-torch HP7 housed in the microphone plug, or a small mercury cell in the microphone itself. The current drain is very small (a fraction of a milliamp) and a long battery life can be expected.

A major advantage is cost, as electrets are very inexpensive. Unfortunately this had encouraged many makers to produce poor specimens which can give disappointing results. As with capacitor mikes, they are affected by high humidity and can be overloaded by very loud sound sources. Given a good make, the electret has much to commend it for general use and has been found particularly useful for public address work, where the absence of a mid-range resonance enables feedback to be more easily controlled.

Directional Response

The manner in which a microphone responds to a sound field differs according to the operating principles and, of course, is important for choosing the correct type for a particular task. task.

Omnidirectional: When the back of the diaphragm of a microphone is sealed, there is an isolated body of air trapped in

the enclosure, just as in a sealed speaker cabinet. This exerts a fixed pressure against which the diaphragm works. The sound pressure wave is alternately greater and less than the pressure exercised on the back of the diaphragm by the trapped air, so the diaphragm moves backwards and forward. Now sound pressure (like barometric pressure) exerts force in all directions, not only in the direction of sound propagation. So, just as a barometer will indicate pressure of air irrespective of how it is mounted, the pressure microphone will react to sound whether pointed toward the sound source or not.

It is therefore said to be omnidirectional, responding to sounds coming from all directions. This characteristic only holds good up to the freqeuency where the wavelength is greater than the diameter of the diaphragm. Above this (at

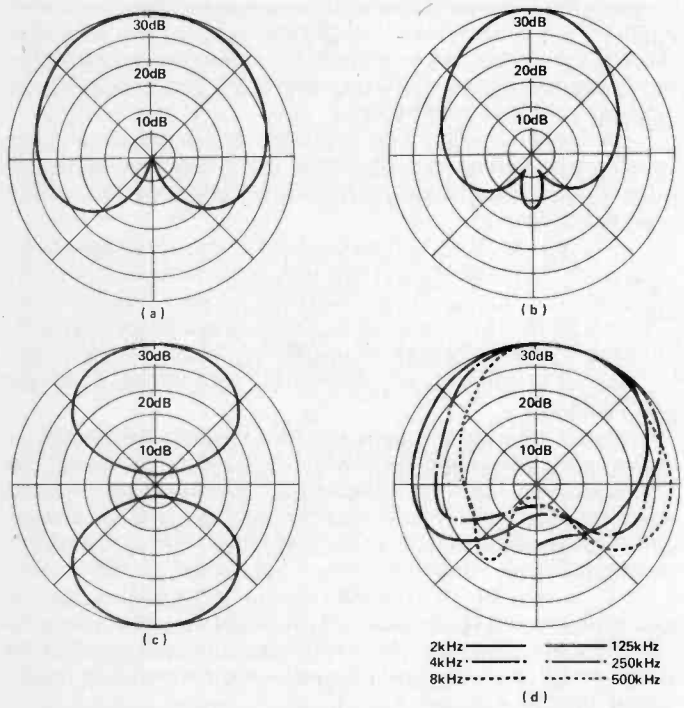


Fig. 9 Polar diagrams of nominal directional response. (a) cardioid; (b) hypercarioid; (c) figure eight; (d) polar response varies with frequency; plots are given for six different frequencies.

shorter wavelengths) the microphone begins to exhibit directional properties. When pointing at 90° from the sound source there are interference effects across the diaphragm, with different parts being affected by different portions of the sound wave at the same time. It can therefore only respond to a limited extent, proportional to the resultant pressure over the whole area.

When the microphone is pointing in the opposite direction to the sound source, the diaphragm is in an acoustic shadow for these short wavelengths, as all objects cast a sound shadow behind them (due to diffraction) when their size is greater than the wavelength of the sound. Hence the microphone response is very low at such frequencies in that position. To give some idea of the frequencies involved, at 1" the frequency is 13.5 kHz, at 1.5", 9 kHz. Most modern microphones have diaphragms smaller than 1" so the effect only occurs at the upper limits of the audio spectrum.

Cardioid: If an aperture is made in the chamber behind the diaphragm so that it is no longer sealed but is open to atmospheric pressure, the mode of operation is changed. Sound pressure is exerted on both sides of the diaphragm which, if equal, would result in zero movement and output.

However, access to the rear through vents at the sides of the microphone is restricted so that the pressure is not equal. Furthermore, the direct action of air molecules moving back and forth along the axis of propagation imparts energy to the front of the diaphragm when the source is situated at the front. Therefore part of the resulting motion is due to the velocity of the air molecules, so the microphone is often termed a velocity type. Pressure differences also are involved so it can be known as a pressure gradient microphone.

When the sound source is to the side of the microphone, similar pressure differences exist between the front and back of the diaphragm but there is no direct impingement of air particles on the front of the microphone. In fact, there can be limited velocity effect on the back of the diaphragm due to diffraction around the edges of the vent. Hence the output is much lower, with sounds coming from the rear of the microphone being almost completely cancelled and producing very little output.

The unit is therefore directional, favouring sounds arriving from the front and progressively rejecting others as the angle from the front increases. When plotted on a circular graph (in which the concentric circles represent the output level) the response resembles a heart-shape — hence the term cardioid.

Hypercardioid: If the acoustic resistance offered by the vents is modified, so that pressure exerted on the back of the diaphragm is increased, the response to sounds arriving from the side is reduced even further, and the directivity increases. The polar response is therefore narrower and becomes a hypercardioid. One drawback is that the increased rear sensitivity often more than cancels the effect of sounds coming from the rear on the front of the diaphragm. This results in a negative lobe at the rear.

Figure Eight: Ribbon microphones are equally sensitive to sounds coming from the front or rear unless the back of the instrument is deliberately restricted by pads. As they are velocity devices, the response to sounds coming from the sides is zero. The polar diagram with its front and rear lobes therefore appears like a figure eight. Some ribbon models have been designed to give a cardioid or hypercardioid polar response by using sound guides at the front and suppression at the rear.

Gun Microphone: These use an interference tube in conjunction with a pressure gradient transducer to give a highly directional forward response. The tube is open at the end and has a series of slots or holes running along one side.

When pointed at the source, pressure waves enter the end of the tube and the side holes to arrive simultaneously at the transducer diaphragm. When the source is to one side, the sound follows various paths of different lengths — the shortest

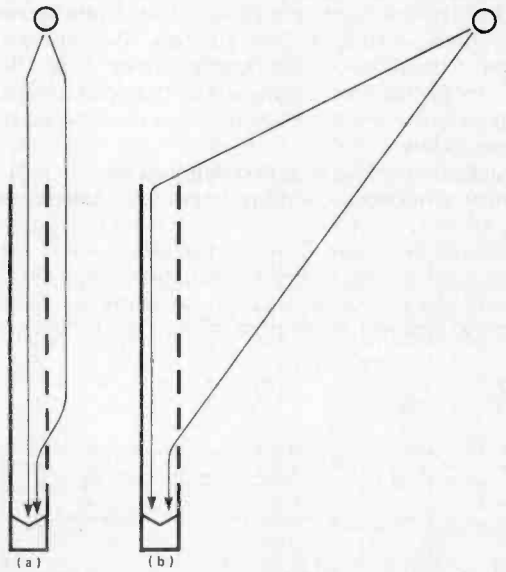
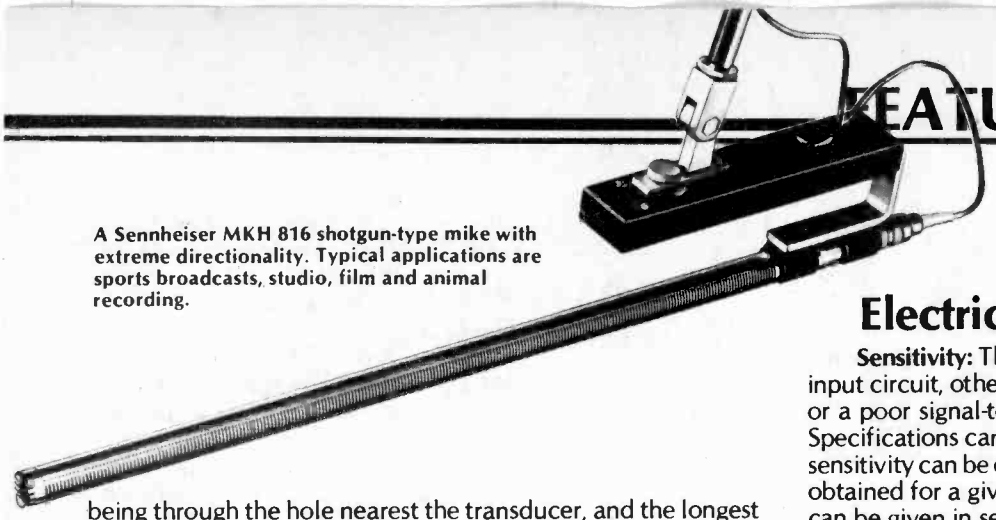


Fig. 10 (a) Sound enters gun microphone from the end and side to arrive at the diaphragm simultaneously when the source is the front. (b) With a source at the side, sound paths differ in length. When the difference is half a wavelength cancellation occurs. This happens over a range of frequencies due to the different spacing of the holes.



A Sennheiser MKH 816 shotgun-type mike with extreme directionality. Typical applications are sports broadcasts, studio, film and animal recording.

being through the hole nearest the transducer, and the longest being via the end of the tube. If the difference between these is half a wavelength, they will cancel and there will be zero pressure exerted on the transducer.

Obviously this happens at only one frequency, but as there is a series of holes there will be cancellation effects for all frequencies within the range of the tube. Cancellation also occurs when the path difference is 1½, 2½, 3½ (and so on) times the wavelength. Of course there are multiple paths, not just the two that provide cancellation, but being spaced between the critical pair these tend to balance out and produce secondary cancellation.

The tube is only effective at frequencies above the half-wavelength value of its total length — below this it reverts to ordinary hypercardioid operation or whatever the transducer mode is without the tube. Some gun microphones have omni transducers, which seems odd since there is no directivity at all below the tube frequency range. Omnis can be identified by having no vents behind the diaphragm. The half-wavelength frequency of a 2 ft tube is around 280 Hz. Directivity increases linearly with frequency.

Frequency Response

Frequency response is often quoted between two limits, such as 30 Hz-18 kHz. This is useless as it does not specify the levels at which the given limits are measured; they could be at −3 dB, −20 dB or any level in between. Furthermore, they give no indication of the flatness or otherwise of the response. An improvement is to give level limits such as 30 Hz-18 kHz ± 3 dB. This means that the limit frequencies (and all in between) do not vary by more than ± 3 dB.

Even this is not entirely satisfactory as it does not reveal whether deviations are sharp peaks or broad plateaus, nor their frequencies. The only certain method is the response curve, but even here there are pitfalls. Take a careful look at the vertical calibration; if the divisions are denoting large steps, the curve will look a lot flatter than it really is. The divisions should be in 1 or 2 dB steps, but 4 or 5 dB increments are not unknown and so can be misleading.

The actual response required depends on the application, but for most, smoothness is more important than extent.

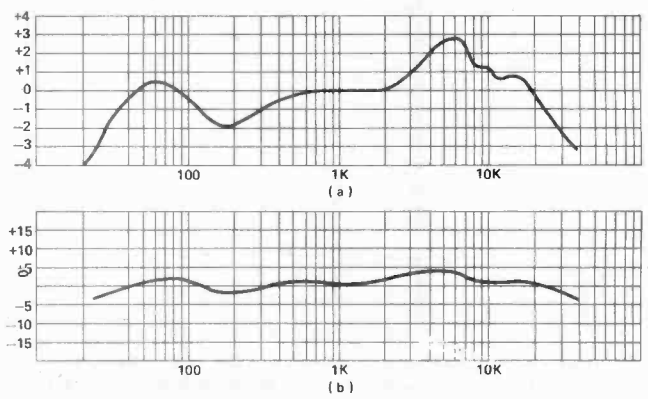


Fig. 11 Which microphone would you prefer, (a) or (b)? A close look at the vertical calibration shows that the frequency response is the same for both. Always check calibration when comparing response curves.

Electrical Characteristics.

Sensitivity: The output must be sufficient to fully drive the input circuit, otherwise a power amp may not attain full power or a poor signal-to-noise ratio may result in other equipment. Specifications can be confusing due to the many ways in which sensitivity can be expressed — it stipulates the electrical output obtained for a given sound pressure, and both of these factors can be given in several different units.

Impedance: There are three commonly used impedance ranges; low (30-50 ohms), medium (200-1000 ohms, and sometimes also called low) and high (47 kilohms). High impedance screened cables result in HF losses over all but short lengths, while low impedance is vulnerable to resistive losses over long cable lengths and noise due to connecting plug contacts. It is the medium impedance, 200 ohms in particular, that is commonly used by professionals, and it is taking over from the previous standard of 600 ohms. Impedances in this range match well with most transistor input stages and can be fed without the use of a transformer.

Noise: When placed in a vacuum with no air actuation of the transducer, a microphone will still produce an output. This is due to thermal agitation of the wire in the case of moving-coil or ribbon units; for capacitor microphones it is caused by similar agitation of the diaphragm metalising, plus irregularities in the supply and noise in the preamp. Owing to the extra sources, the latter generate more noise than the former.

There are several ways of specifying microphone noise — the simplest is the actual output in microvolts. This may be weighted to take account of the nuisance effect that some frequencies have over others. The DIN 45-405 weighting curve, which peaks at between 3-4 kHz and rolls off on either side, is commonly used.

To be meaningful, the noise should be related to the sensitivity, as a given noise level will obviously be more obtrusive with a low signal output than a high. One way of doing this is to convert the noise voltage into equivalent sound pressure — the sound pressure that would produce the same output voltage in that microphone. This may be expressed as a decibel ratio, using the hearing threshold (0.0002 ubar) as the reference level. Alternatively it may be expressed in phons, which is numerically the same as the dB value.

Hum: Moving-coil or ribbon transducers are susceptible to hum when working in strong hum fields. Some have hum-cancelling coils built-in to minimise it. Occasionally the manufacturers a figure which indicates hum sensitivity, the unit being the microvolt per microTesla. The standard field density is 5 uTesla.

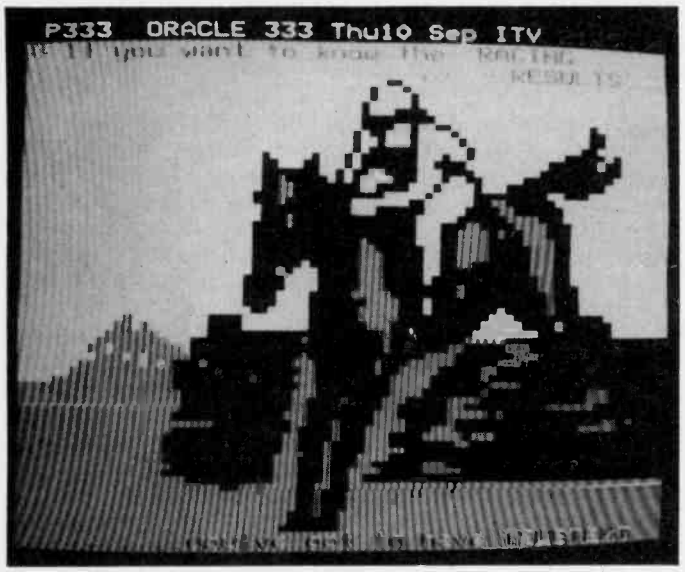
Overload and Distortion: Capacitor microphones are the most easily overloaded by loud sounds so a maximum sound pressure is often quoted, 200-300 ubars being a common value. Lower output models can handle larger volumes, some going to 500 and one even to 1,000 ubars. One model designed for instrumentation applications has an extremely low output that will accept up to 5,000 ubars, a level that would destroy human hearing.

Dynamic microphones (moving-coil and ribbons) are not usually given a maximum sound pressure rating, but they often have a level specified at which distortion will rise to a particular value. This is normally 0.5% THD (total harmonic distortion), but some of the cheaper ones use a 1.0% THD reference. The description usually reads SPL for 0.5% THD = 300 ubars (for example).

There is a tremendous range of microphones now available, making the best choice none too easy. However, if the acoustic and electrical requirements (together with the other features described) are isolated, the selection and subsequent use will be reasonably straightforward.

TELETEXT EXPLAINED

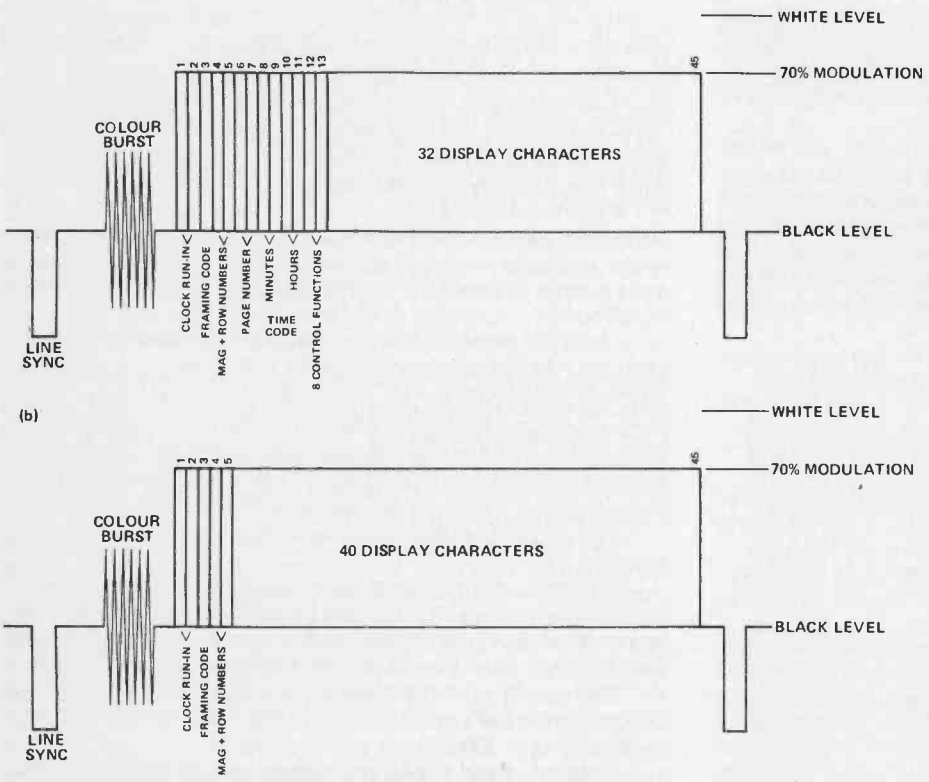
Take a trip inside your TV with Vivian Capel and find out how the Teletext information makes its way onto your screen.



Over the recent months there has been much publicity for the Teletext services of the BBC and ITA, called Ceefax and Oracle respectively, although the first experimental transmissions began as long ago as March 1973. At first, the two services differed in technical details, but commendably, these were mutually resolved and in the following year, April 1974, the standards were unified and a service was launched in September of the the same year.

As most readers will already be aware, the service consists of magazines of up to 100 pages, each of which can be selected by the viewer by means of a keypad. Once selected, the page can be held on the screen indefinitely, yet can be updated if additional information is added in the transmission. Though primarily of written material, pages can also contain rudimentary graphics and illustrations. Lettering can be of different colours and sizes.

The most remarkable feature of Teletext is that all this can be transmitted along with the normal television service without interfering with it in any way and without using extra bandwidth. Thus it is compatible with existing receivers, most of which can receive Teletext with the addition of a suitable decoder.



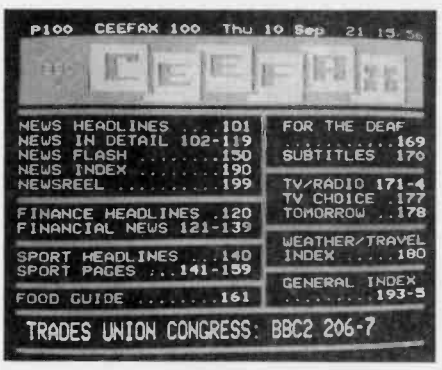


Fig. 1 (a) Composition of a TV scanning line showing the make-up of the 45 Teletext bytes for a header row. The first 13 bytes are control bytes as indicated. (b) A normal Teletext row. The first five bytes are the same as for the header, and are followed by 40 display characters.

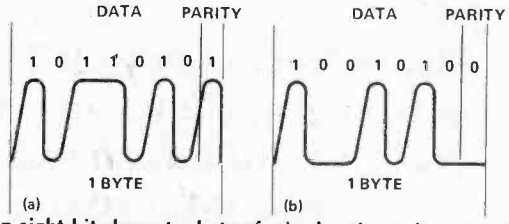


Fig. 2 (a) An eight-bit character byte of raised cosine pulses. Seven bits carry the data code, the eighth being a parity protection bit. If the number of bits at logic 1 is even, the parity bit is also 1 to give a total odd number. In (b), the number of parity bits is already an odd number, so the parity bit is O.

So how is it done? Well, the 625 lines that are transmitted to make up each complete television frame are not all used to convey picture information. The frame is divided into two fields which are interlaced so that the scanning lines of each fall between those of the next. This greatly reduces flicker. However, there is a gap or margin at the top and bottom of each field which totals 25 lines; it is made up of 22½ lines at the top and 3 at the bottom of the first field, and 22 at the top and 2½ at the bottom of the second. If this seems odd, remember that the bottom of the first field runs on to the top of the second and the bottom of the second to the first of the next frame, thus there are always 25 lines between fields.

The purpose of these blank lines is to ensure that the flyback lines from bottom to top of the frame are blanked out and also that the time bases have settled down nicely after the start of a new scan before any actual picture information appears. They do, however, offer the opportunity of carrying extra information which is not directly intended for reproduction on the viewing screen. For example, lines 19 and 20, and 332 and 333 on alternate fields carry certain transmitter test signals.

It is on the previous lines (that is 17 and 18) and 330 and 331, that the Teletext signals are carried. If the height of a normal TV picture is reduced so that black bands appear at the top and bottom, the Teletext signals can be clearly seen as a row of dots and dashes which are continuously changing form.

Teletext Pages

The pages are not transmitted as a normal TV picture by means of a video signal with each part of the displayed page being successively built up by the scanning lines. This would take not two, but all of the lines which are transmitted, and furthermore, it would not be possible to freeze a page, anymore than a frame of a normal TV programme can be frozen.

Instead, each character is transmitted as a separate and complete signal consisting of an 8-bit binary word or byte. The 1 or high bits correspond to about 70% of peak-white modulation of the carrier, while the 0 or low bits are represented by the black level. Bit pulses are not square in form and do not need to be; actually an extended video bandwidth would be necessary to handle square waves of the bit frequency which is 6.9375 MHz. The waveform is described as a raised cosine.

Each page row consists of 45 bytes, and as 5 bytes are required for synchronizing and address purposes in a normal row, this leaves 40 for the display characters. There is a maximum of 24 rows to a page so this means that up to 960 characters can be displayed on each page.

Following the identification coding for page one, each of the 24 rows are transmitted consecutively, whereupon the code for page two follows succeeded by its 24 rows, and so on until all the pages in the magazine have been transmitted. Then page one is transmitted again and all are repeated as if on an endless belt.

When the user keys a particular page the decoder waits until the appropriate identification code comes round again, and then loads all the following information into a memory bank until the code for the next page is received, when it ceases storing the signal. From the memory, the data signal is fed to the alphanumeric generators which produce a video signal to reproduce the characters on the receiver's display tube. The generators will continue to scan the memory circuits until the unit is switched off or another page is selected. Thus the page is displayed continuously, irrespective of what is being transmitted subsequently. But there is a provision whereby information on the same page can be changed if following versions of the page are altered with fresh information.

Access Time

So we have a bird's eye view of the basic principles. We will now take a closer look at the details. First, the access time. This is the time that elapses from when the page is keyed to when it actually appears on the screen. It is obviously a matter of chance - like waiting for a bus - it depends on whether the one you want is due or has just passed. So what is the maximum time you may have to wait?

One complete page row is transmitted during each TV line, so as two lines per field carry Teletext signals, a full 24-row page is transmitted during twelve fields, which at a field frequency of 50 Hz is 0.24 seconds. If 100 pages are contained in the magazine, the complete magazine cycle will take 24 seconds. So if you have just missed the required page, that is the maximum time it will take for it to come around again.

However, while each row takes the same time to transmit — whether full or not — the blank spaces each requiring a character byte, this is not the case with the pages. Any that have less than 24 rows are transmitted more quickly. Also, not all of the 100 pages are used in each magazine at present. Therefore the complete cycle will be rather less than 24 seconds and, of course, you will not just miss the chosen one every time. So average access time should be under 12 seconds.

Extra pages or magazines mean longer access times as long as the transmitted data is restricted to two television lines. It is possible to use more lines (up to 16 in each field) and this would extend the capacity to eight times its present maximum without adding to the access time. It is even possible to send different versions of the same page under differing time codes to give further extension. As there are now three, and shortly four, channels available, the potential of information available via teletext is quite considerable.

Coding

Each normal row consists, as we have seen, of 45 bytes. The first two of these is a series of alternating 1 and 0 bits there being 16 in all. Thus a pulse train at the bit frequency of 6.9375 MHz is produced, and this serves to synchronize the decoder's data clock which detects and recovers the individual bits. Readers who are familiar with colour TV principles will recognise this as being similar to the colour-burst occurring after each line sync pulse that keeps the colour oscillator in step.

One form of clock circuit is a high-gain tuned circuit resonating at the bit frequency and which is set into oscillation by the incoming stream of pulses. The phase of the clock output wave form is delayed so that the peaks occur about the middle of each bit.

The third byte is known as the Start or Framing Code. As each byte in the sequence follows its predecessor without a break, the decoder must have some means of determining where to start. Again referring to television as an example, the scanning lines must all start at the same point, and to ensure this, line sync pulses are inserted. Without them, the screen is filled with a meaningless jumble of lines which can be produced if the line-hold control is misadjusted.

The Framing Code performs the same function, defining the start of the next byte, so keeping the detector circuits in step until the end of the row when, after the next clock run-in bytes, a further framing code byte is received. This byte takes the form 11100100, which is so chosen that it cannot be confused with any character byte even if one bit has been wrongly received.

The fourth and fifth bytes carry codes to identify the magazine number which is also the hundreds digit of the page number, (pages 200-299 are in magazine No. 2; pages 300-399, magazine No. 3 and so on), and also the row number. The latter enables rows to be missed to give spacing without having to waste transmission time by including a row of spaces. Thus if row 10 is signalled after row 6 there will be three spaces without transmitting three empty rows.

Header Row

The above description is for normal rows, five sync and control bytes followed by 40 character bytes, but the first or header row has additional information. This incidentally is termed row 0, the following ones being numbered 1-23.

After the five bytes which are the same as for normal rows, come bytes six and seven which carry the code for the page number in units and tens respectively. Next follow four bytes which convey a timing code; minutes units, minutes tens, hours units, and hours tens. The time thus indicated is the nominal transmission time for that page and may not necessarily be the actual clock time. It really is a form of reference which enables different editions of the same page to be sent at different times. At say one a minute, there could be over 1,000 editions of a particular page in a day, each identified by its time code. Some decoders have the facility of storing a particular page for future display, so pages that had been superseded by later editions could still be recalled if they had been so stored. Further possibilities whereby this code can be used to provide further facilities are in the course of development.

This makes 11 control bytes so far for the header row and thus brings us to the last two, 12 and 13. There are 16 bits available here, and eight are used to control the same number of functions. When the appropriate bit is at 1, the particular function is activated. The remaining eight are protection bits which we shall deal with in a moment.

The eight functions controlled by bytes 12 and 13 are as follows.

Erase Page. Should the information contained on any page be significantly different from that contained on a previously transmitted page bearing the same number, the former page will be erased to avoid confusion. This will be done with the bit set at 1.

Newsflash. When the newsflash page is keyed, the viewer watches an ordinary programme which is mixed with the blank Teletext page. If a newsflash is transmitted, it appears boxed and superimposed over the normal picture. When the appropriate bit is set to 1 in the transmission the newsflash appears. The viewer can wipe the flash using a control on this keypad, but if he stays tuned to the page, the next flash will appear in due course.

Subtitles. This is another page to be superimposed on a normal TV picture. Certain programmes are subtitled, and when this page is selected, the presence of a 1 bit in this position brings up the subtitles in a box.

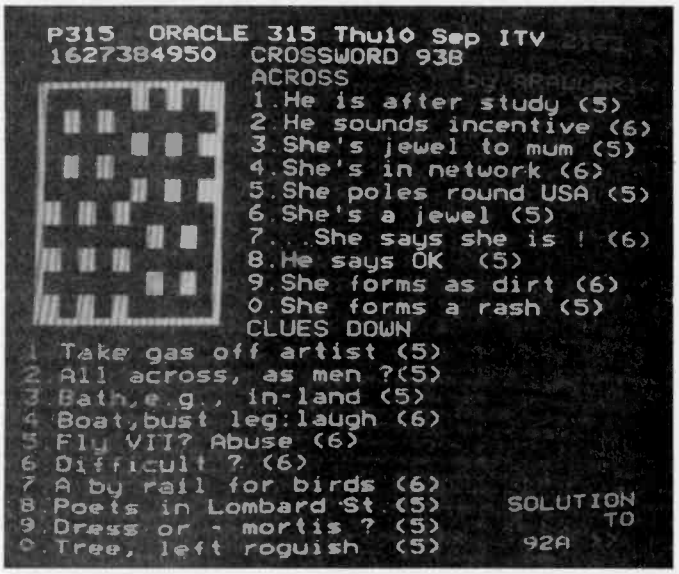
Header Suppression. The rest of the header row after the 13 control bytes consists of 32 characters which always follow the same format for each page and are the only ones in the row that are visible. They contain the page number displayed (as distinct from the page code which identifies the page for the decoder), the originating service (Ceefax or Oracle), the date and day and finally the clock time. For some pages such as a newsflash, it may be desirable to suppress the header, and a 1 bit here will do just that.

Update Instruction. Where part of a page has been updated from the previous transmission, it may not be transmitted in its entirety, only the new portion. This control bit instructs the memory to replace the old rows with the new.

Interrupted Sequence. Some pages are transmitted more often than their natural sequence, such as the Index and others deemed in greatest demand, to reduce their access time. Also some pages such as the subtitles are granted priority, as a change of subtitle may be required more often than the normal page cycle. This bit ensures priority handling of the data, and also suppresses discontinuous page numbers.

Inhibit Display. If for any reason a page has become unintelligible, this control bit will inhibit its display.

Rolling Pages. If material is too much to be contained in a single page, rather than take up another numbered page, a second, third or even more sub-pages can be displayed in sequence with each page being held for a specified time, say a minute. In such cases the headers of sub pages are not needed, and the control bit inititates the next sub-page. An example of this type of material is the football results. The user can hold any sub-page if desired.



Protection Codes

It is obvious that a signal depending on a string of pulses and spaces is vulnerable to error. A momentary break in the signal where there should be a pulse could be interpreted as a 0 instead of a 1 by the decoding circuits. Also an interference spike could trick the circuit into reading a pulse or 1 instead of an 0.

In the case of the characters, this could result in a completely wrong character being produced resulting in confusion or even a misleading message. To avoid this, each character byte includes what is known as a parity bit. The code to produce the range of characters required needs only seven binary code bits out of the eight in each byte.

The eighth becomes a 1 if there are an even number of 1 bits already in the byte, but becomes a 0 if the number of 1 bits are odd. In this way the 1 bits are always odd in number. The decoding circuits count the number of 1's in each byte and if they are an even number it is obvious that one has been lost or there is one too many, so the entire byte is rejected. Thus a blank space occurs which is better than a wrong character. This is termed odd parity.

While it offers a good degree of protection against error, it is not foolproof; two errors could occur in a byte to give the required odd number of 1 bits. In the case of the characters, this could be confusing if a wrong character was thereby displayed, but it would not be disastrous. Such an error in any of the address codes could cause complete mayhem! Hence, a greater degree of protection is required for these than for the characters.

A method is used that was devised by R.W. Hamming of the Bell Telephone System of America in 1950. In the eight-bit byte, only four bits are used to carry the signal, the other four are parity check bits. This is why, as described earlier, the eight functions determined in the header row require two bytes, numbers 12 and 13 totalling 16 bits. Eight of those are Hamming protection bits.

These protection bits are interleaved with the data bits so that bits 2,4,6, and 8 carry the data, while bits 1,3,5, and 7 are the parity ones. This provides a check on every single data bit which

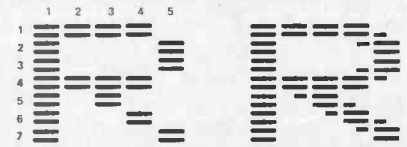


Fig. 3 The 7 x 5 matrix on which characters are built, with and without character rounding.

if incorrect, cannot only be detected but corrected. Thus several errors in the same byte can be put right. So a high degree of error protection is achieved although at the expense of bit capacity.

The Characters

Normal characters occupy seven field scanning lines in the reproduced picture; however, some lower case letters that have tails such as q,y,p,g, and j, need an extra two lines below normal. One line is allowed between rows, so this makes ten lines in all. As the picture is made up of two interlaced fields, the total number of lines is twenty. As the character cannot be produced simultaneously but must be built up by the line scan, the character generator must consult the memory many times before the character is complete.

Horizontally, the characters are made up of five segments or dots. So with the seven lines height, this forms a matrix of 35. A sixth dot is allowed for spacing between the characters. The time taken to scan one line of one character is 1 us which divided by the total of six dots, gives a dot frequency of 6 MHz, the normal maximum video bandwidth.

Unlike lettering appearing in a normal TV picture, such as captions, the form and appearance is not fixed at the programme source, only the type of character. Form is determined by the character generators in the decoder and so can vary, some decoders have a double-height. Another improvement that is becoming more common is the rounding of edges which otherwise present a stepped appearance. In the case of the double height, the instruction to do this is built into the character code, but a decoder without this facility just ignores it and reproduces at normal height.

A total of 94 characters can be displayed including upper and lower case letters, numerals, the fractions ¼, ½, and ¾, punctuation signs, and various others that are commonly found on typewriter keyboards; plus spacing in background or display colour.

Graphics

In addition to alphanumeric signs, basic illustrations can be produced, being built up from simple graphic building blocks. For these, a matrix three units high by two wide is used, the graphic block being the same dimension as a character.

These are 64 possible was in which the six units in the graphic block can be illuminated, from all on to all off, so these make up the 64 graphic symbols. In addition, the units illuminated can either be contiguous, that is appearing as one unbroken illuminated segment, or separated from the adjoining ones. So for example, if all six in the block are on in the contiguous mode, it will appear as a continous rectangle of light, but if they are in the separated condition, it will appear as six small blocks in the larger one which will look like a dark grating. The required mode is conveyed by part of the code.

Finally, we will look again at the character coding, which you remember consists of a single byte of seven data bits and one parity bit. The first three bits carry the colour information. They give seven combinations and these correspond to seven different colours; red, green, yellow, blue, magenta, cyan, and white. The fourth bit,in combination with the first three, signals steady display or flashing, start or end of a box and normal or double height for alphanumeric characters. When graphics are being displayed, the same four-bit combination gives concealed display, contiguous graphics or separated graphics, black background, new background, hold or release graphics. There are eight possible combinations when the fourth bit is set to 1, but not all are used.

The remaining three bits select either alphanumerics or graphics and the specific character. So seven bits are utilised, the eighth being, as we have seen, the protection parity bit.

Add-on decoders designed for use with an existing set contain a tuner, IF and demodulator circuits so that the aerial is plugged straight into a socket on the unit. A PAL encoder and RF modulator enable the output to be applied to the aerial socket of the receiver which is tuned to an unused TV channel. The decoder tuner is then used for normal TV viewing.

This extra circuitry increases the cost and, in theory should add a degree of noise to the normal TV picture as the signal passes through two tuners and a modulator instead of the receiver's single tuner. In older sets employing tuners that tended to be noisy, there could actually be an improvement as the Teletext unit may have some gain through its tuner and modulator and so act as a preamp.

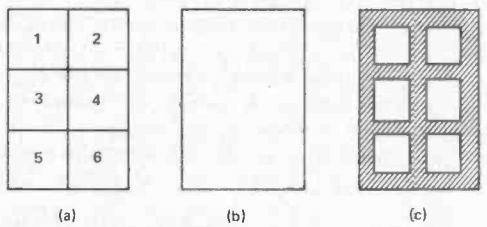


Fig. 4 (Above) (a) Graphics are built up on a 3 x 2 matrix numbered as shown – any position may be illuminated. (b) When the units are coded for contiguous display, they join with those adjacent. Here all six are on to give a completely illuminated rectangle. (c) When coded for separate display, they are isolated from their neighbours as shown here.

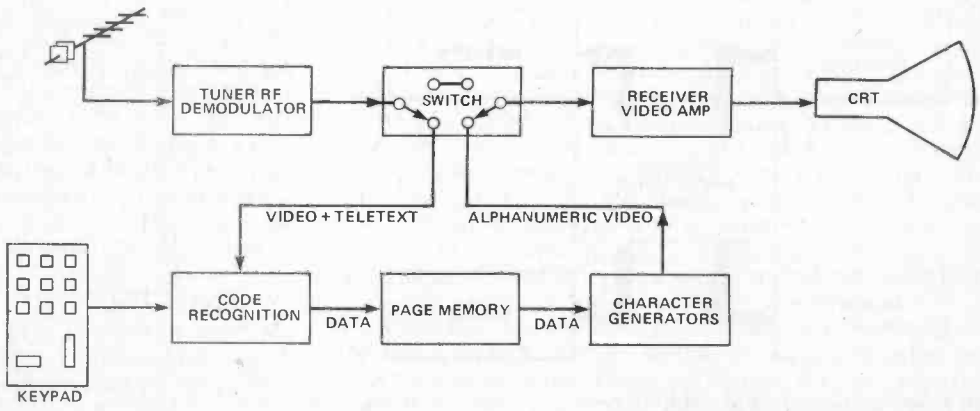


Fig. 5 Simplified block diagram of a Teletext receiver.

MICROS 1

In this series, Owen Bishop takes the lid off computers and the ICs that go into them. This is the definitive treatise on hardware.

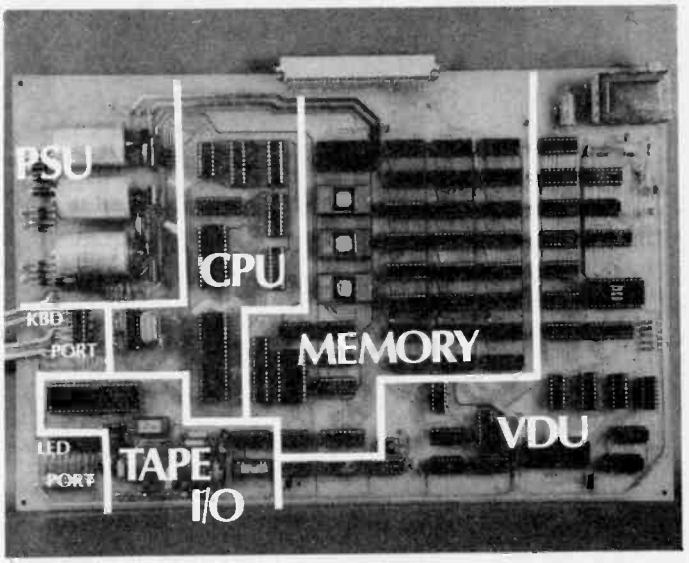
This series is aimed at those readers who already know something about electronics, but who would like to know how electronics is being used today in perhaps its most important application of all — the computer. The series will be concerned with only one of the two types of electronic computer, the digital computer. The other type, the analogue computer, has several important applications but in the main its work has been taken over by digital computers.

We still owe something to the analogue compuer, for our trusty work-horse, the op-amp, was originally designed as its building block. Whereas the analogue computer operates with precisely determined voltages which are allowed to vary continuously over their range and are analogues of continuous physical quanities, the digital computer operates with only two discrete voltage levels. The analogue computer depends on the high precision of its op-amps, and need an op-amp for every step in its computations.

As we shall see, the electronic requirements for the digital computer are much simpler, allowing designers to concentrate on obtaining high speeds of action. The units of the circuit are simple logic gates, thousands of which can be manufactured on a single slice of silicon, already connected to form the complex logic circuits of the computer. This allows the digital computer to have great computing power combined with flexibility of function. It also allows the computer to be mass-produced cheaply so that, today, anyone with a few tens of pounds to spend can buy one.

The Heart Of The Matter

Figure 1 shows the heart of the computer to be its central processing unit (CPU). It is connected to a number of other devices — the peripheral devices. Input devices usually include a keyboard, so that the operator can send information to the CPU. Information may consist of instructions and data. Input devices might include sensors (eg circuits to measure temperature) so that the CPU can obtain its data directly without need for intervention by a human operator. One essential part of this would be an analogue-to-digital converter sub-circuit, to convert the analogue quantity (in this case temperature), to its digital coded equivalent.



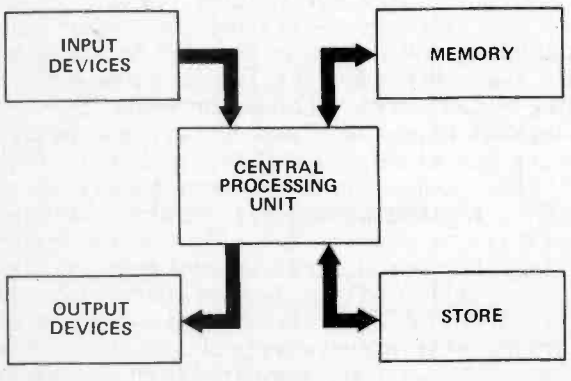


Fig. 1 Block diagram of a computer.

Output devices allow the CPU to communicate the results of its computations to the world outside. There is usually a monitor screen on which messages and the results of calculations are displayed. There may also be a printer or a chart plotter. Alternatively there may be direct control of a robot arm or similar device.

The memory is one place where information is stored. The instructions tell the CPU what to do (its program), and it is provided with data to work on. The computer is able to use part of the memory for storing other data which arises from its computations. Information can be transferred between CPU and memory very rapidly and in either direction. Memory is where the currently-used information is held. The store is for information that is not required urgently. The store may consist of a tape deck or disk drive, by means of which information is stored in magnetic form. Blocks of information can be transferred between CPU and store in either direction, but only relatively slowly. The amount of information which can be held in store is much greater than the amount held in memory.

The CPU

This has the job of receiving instructions and data, either from input, memory or store, processing the data according to the instruction, and then sending the results

of its computation to an output device, memory, store, or possibly to more than one of these. In a main-frame computer, the CPU occupies several circuit boards, but in the personal computer the whole CPU is replaced by a single integrated circuit, the microprocessor. This article and the remainder of the series will concentrate on the personal computer, or microcomputer, using a microprocessor as its CPU.

We have been able to use very large scale integration (VLSI) to put all the logical parts of the CPU on to one slice of silicon. The CPU must include an oscillator, or clock, by means of which all its actions and the actions of other periperal devices are synchronised. It is not possible to reduce the physical size of the components required for this, in particular the quartz crystal, so a least part of the clock circuit is external to the microprocessor. The clock circuit and microprocessor (MPU) together constitute the CPU of the microcomputer.

We Want Information

Before we look at what goes on inside the MPU we must consider the concept of information in more detail. The unit of information is the bit. The term 'bit' is a shortened version of 'binary digit'. A bit can have one of two values, '0' or '1' but not any other value. This binary concept is widespread in thought, in logic and also in electronics. Table 1 shows pairs of opposite and mutually exclusive states. A binary digit is '0' or '1'; it cannot be anything else. A statement is true of false; truth is by definition the *whole* truth, for half-truth is meaningless. A switch is either on or it is off; it cannot be partly on. If the circuits are made so that only two voltages (low and high) produce definite results and so that intermediate voltages give indeterminate results, then voltages are either high or low. Transistors are either fully off, or fully on (saturated). Given these binary states, the state of any one pair in Table 1 can be used to represent the state of any other pair. For example, we can stipulate that the digit '0' is represented in a computer circuit by a low voltage, and the digit '1' is represented by a high voltage; falsity by '0' or a low voltage, truth by '1'. Here we have a system which allows

TABLE 1

0	1
No	Yes
False	True
Absent	Present
Switch off	Switch on
Transistor off	Transistor on
Open circuit	Closed circuit
Low voltage	High voltage

numerical values and logical statements to be represented in terms of electrical signals. This is the basis of the digital computer.

Grab A Byte

In this system, the bit is the minimum quantity of information to be dealt with. Normally a computer deals with far more information than this. Bits are usually handled in groups. Some of the earlier MPUs handled bits in groups of four, but the majority of micros handle them in groups of eight. A group of eight digits is called a byte. In the computer, a byte is represented by a set of eight lines (eg tracks on the PCB), each at high (= 1) or low (= 0) voltage. Or it might be represented by a set of eight flips-flops or bistables, each one either set (= 1) or reset (= 0). According to the interpretation placed on it, the byte could represent:

- A binary value, ranging from 0000 0000 (= 0 decimal) to 1111 1111 (= 225 decimal)
- The truth of falsity of eight different logical statements.
- A coded instruction to the computer.

There is more to be said on this subject later, but for the moment we will rest with the fact that the computer has to handle binary information represented in electronic form.

On The Level

For most MPUs the low and high voltages are standardised at 0 V and + 5 V respectively. These are the same levels as are used in the 7400 TTL series of ICs. These values are nominal; a Z80 MPU, for example, interprets any voltage between — 0V3 and 0V8 as 'low'. Any voltage between 2 V and 5V is interpreted as 'high'. Voltages between 0V8 and 2 V produce indeterminate results and must not be allowed to occur. The lack of insistence on precise voltage levels allows computer circuits to remain relatively simple in electronic terms, yet be highly reliable in action.

Those Important Little Places

If the CPU is the heart of the computer, the heart of the CPU is its **arithmetic logic unit.** The ALU is where data is manipulated according to the instructions stored in memory; we shall describe some of its operations later. The ALU is able to operate on all eight bits of a byte in a single operation. We say that the **word length** is eight bits, or one byte. Some MPUs, such as the Texas 9980A, have a 16-bit word, but the general principles of its operations are the same as described below.

As an example of a well-known MPU we shall first consider the 6502 (Fig. 2). This successful but relatively simple MPU is used in the Apple, the PET, the BBC Microcomputer, and several other popular microcomputers. The ALU operates in close conjunction with the **Accumulator.** This is a set of eight flip-flops which temporarily hold a byte which is to be operaed on by the ALU, or is the result of an operation performed by the ALU. The two registers known as X and Y many also be used to store one byte of data each. Data can be transferred between these registers and the Accumulator in either direction. These registers are therefore useful for storing values obtained in one stage of a calculation, ready for use at a later stage. They are also used as index registers, in which the values held in X or Y are the base addresses of selected blocks of memory. This makes it simpler to access blocks of memory; when storing a table of data, for example.

Since data has to be transferred from one register to another, or from a register to the ALU, it speeds the opera-

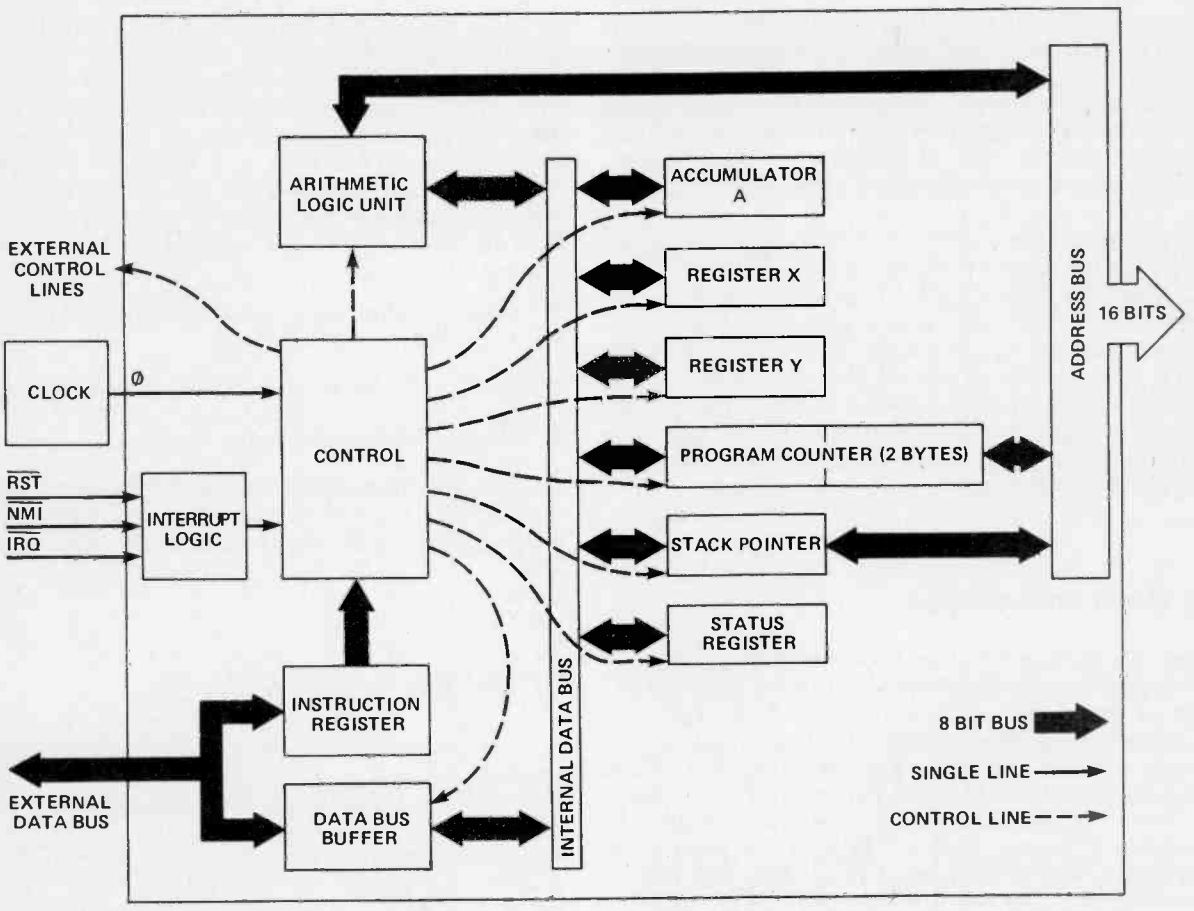


Fig. 2 The internal structure of the popular 6502 microprosessor.

tion of the MPU if a whole byte is transferred in one operation, rather than bit-by-bit. This requires a set of eight lines connecting all the registers and the ALU. This is called the **data bus.** To distinguish it from a similar set of lines which connect the MPU with the peripheral devices, it is more precisely known as the **internal data bus.**

It's Under Control

The **control bus** consists of several lines along which signals are sent to coordinate the actions of the various parts of the MPU. For example, if the data held in a register X is to be sent to the ALU, a signal must be sent along a control line to register X, making it place the data on the data bus. Register X makes the lines go 'high' or 'low' according to the pattern of 0s and 1s held in its eight flip-flops. At the same time a signal must be sent along another control line to the ALU, making it accept the data now present on the data bus. The control lines emanate from a special part of the MPU called the **Control.**

Despite its impressive name, the Control is no more than a slave. It knows how to carry out the tasks it is allotted, but does not remember what it has just done, and does not know what task it must perform next. The list of tasks (the program) is stored in memory at a sequence of locations. The control simply fetches these instructions

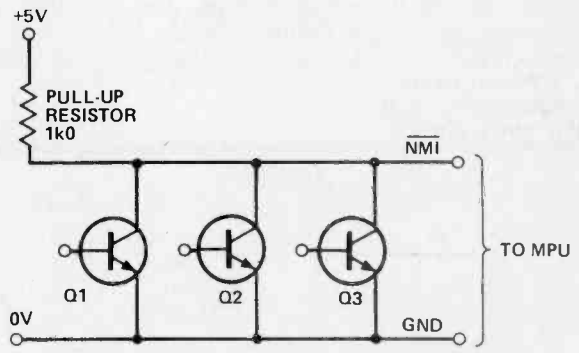


Fig. 3 Switching on any of the transistors generates an NMI.

from memory, a byte at a time, and acts on each immediately it is received. For this purpose it needs the **Program Counter,** a register in which it records how far it has reached in the program — a sort of 'bookmark'. Since a single byte cannot store numbers greater than 255 (decimal), and since most programs have far more bytes than this, the Program Counter is a double-byte register. Its 16 bits allow any number up to 1111 1111 1111 1111 (binary) to be stored, equivalent 65535 (decimal).

During its calculations, the MPU often has to store data in the **Stack,** a special section of memory set aside for

this purpose. As data is added to or removed from the Stack, the postion in memory of the first item in the Stack (the Top of Stack) changes. The **Stack Pointer** register records the current position of Top of Stack, so that the MPU knows where to go to retrieve the stacked data.

Status Symbols

The **status register** should be considered as eight individual bits, arranged together for convenience as a byte. Each bit is **set** (made equal to 1), or **reset** (made equal to 0) individually as the result of a particular operation. For example, bit 7 is set whenever the result of an operation results in a negative value. Bit 1 is set when the result of an operation is zero. These bits, which indicate whether a particular event has occurred or not, are often known as **flags.** Bit 0 holds the 'carry' digit from additions or substractions in the accumulator.

The remaining sections of the MPU are concerned with communicating with the peripheral circuity. There is the **data bus buffer** which detects voltage levels on the external data bus and copies these on to the internal data bus. Or it can operate in the reverse direction. If the data bus is carrying an instruction, this is accepted by the **instruction register.** From there is goes to the control which decodes it and then acts upon it. The **address bus** receives outputs from certain registers putting voltage levels on the 16 address lines, a subject which will be dealt with later.

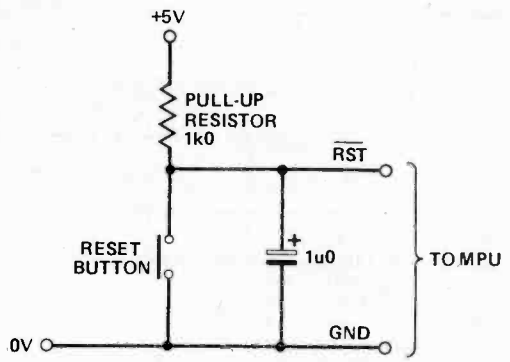


Fig. 4 A suitable circuit for generating a power-on reset pulse. A manual reset button is also provided.

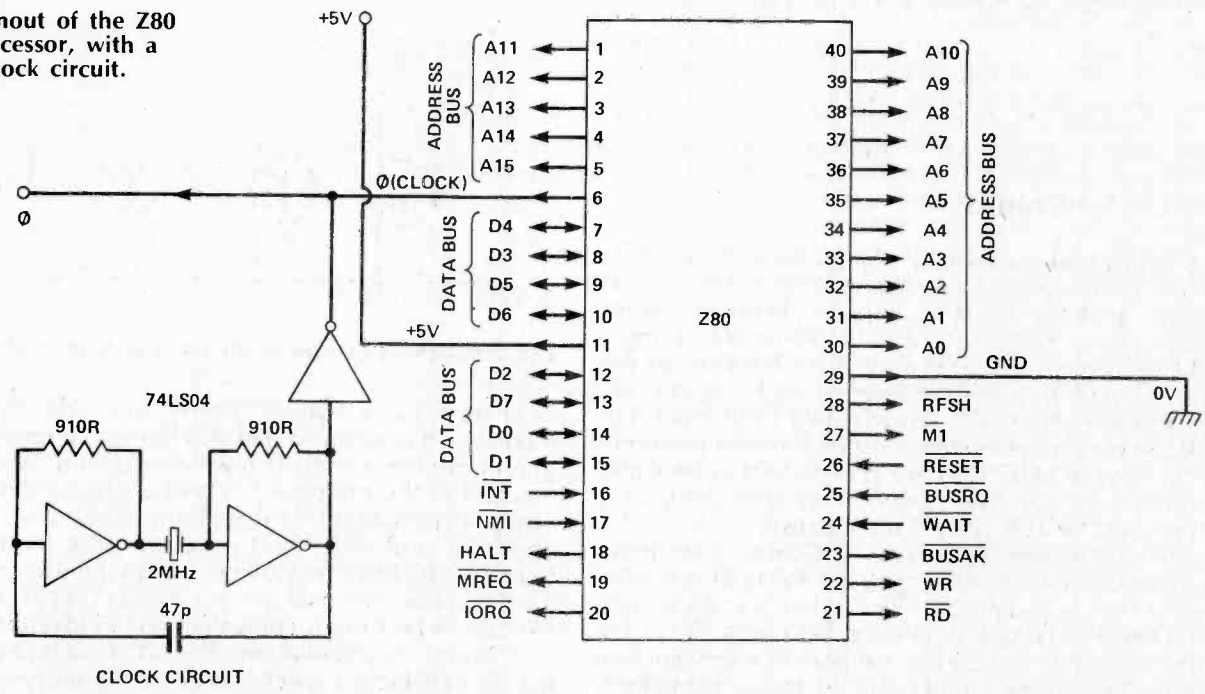
Dealing With Interruptions

The **interrupt logic** receives signals along any of three lines. All three lines are normally held high by pull-up resistors. The lines are thus described as 'active-low'. In other words, it requires a low level on the line to make the MPU respond. Most control lines in the computer are active-low. This makes it simple for any number of devices to bring the line to its low state. If the line is connected to open-collector transistors for example (Fig. 3) this is equivalent to a wired-OR configuration. Then if any one of these transistors is turned on, the voltage level on the line is made low. If a line is active-low, this fact is indicated by a line above its abbreviation (eg $\overline{RST}$ for active-low 'reset').

The **reset** line is used to initialize the MPU, either when the computer is first switched on or if it gets into a 'latched-up' condition, in which normal methods of controlling it do not work. There is generally a pull-up resistor holding the voltage high, with a 'Reset' press-button hidden in a fairly inaccessible place at the rear of the computer. Pressing this button temporarily grounds the reset line.

When the computer is first switched on, resetting is usually done automatically, by having a large-value capacitor to hold the line low for a short period while the rest of the system reaches its full voltage levels (Fig 4).

Fig. 5 Pinout of the Z80 microprocessor, with a suitble clock circuit.



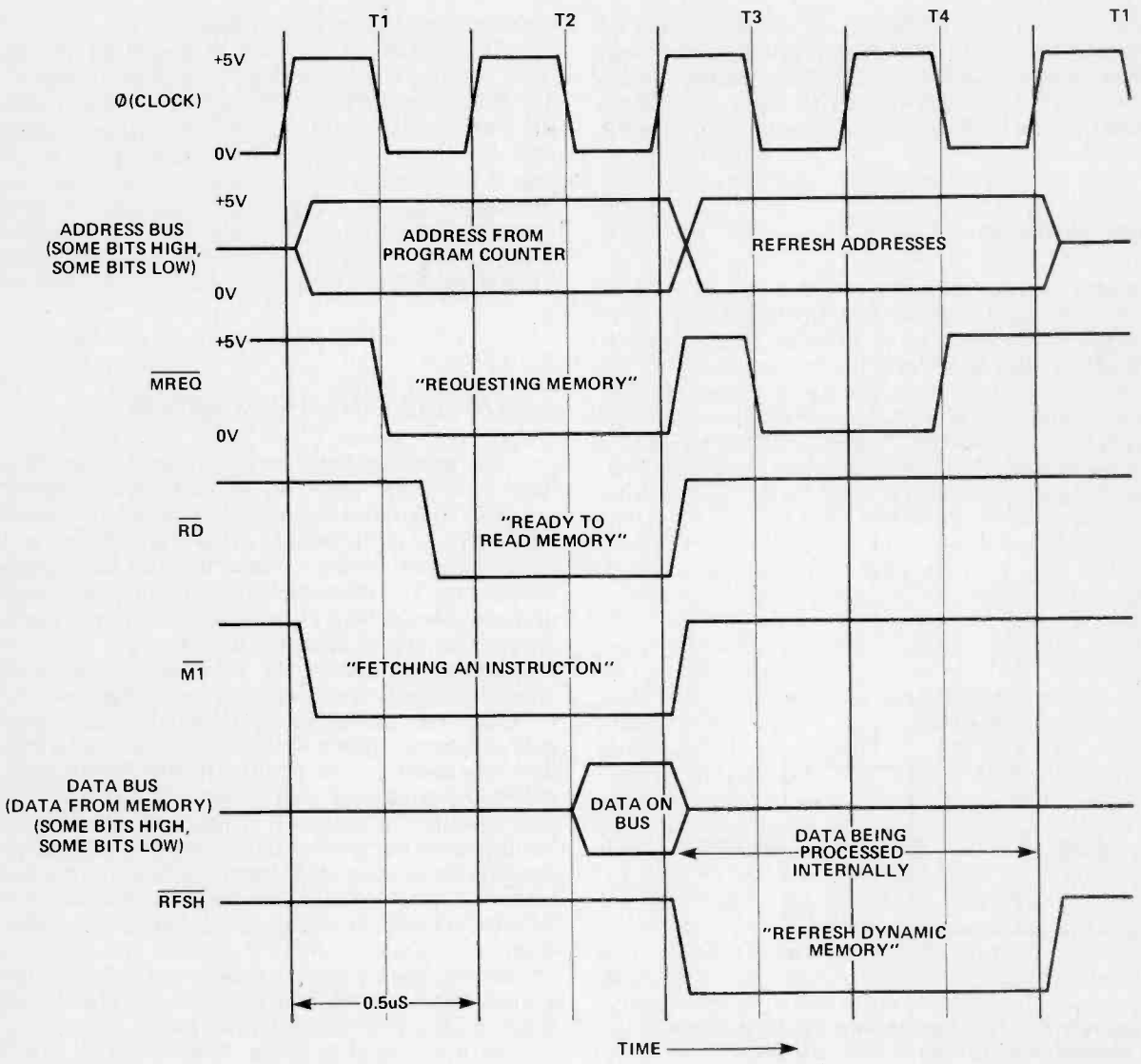


Fig. 6 The clock and control signals for the Z80.

There is no reset button in the Sinclair ZX-81. To reset, you simply turn off the power, wait a moment or two and then reapply power. Resetting the MPU resets the program counter to zero, so that it returns to the beginning of the program stored in memory and starts again.

On receiving a low signal on one of the interrupt lines ($\overline{NMI}$ or $\overline{IRQ}$) the MPU finishes whatever operation it is engaged in, then stores away (on the stack) any data relating to that operation. This takes only a few microseconds, after which the program counter is set to the address in memory of a special interrupt service program. It performs whatever this program requires, then returns to its original program, recovers the data from the stack and continues with the original program as if nothing had happened. Interrupts are used by peripheral devices to gain the attention of the MPU when it is urgently required. The non-maskable interrupt ($\overline{NMI}$) takes priority. It cannot be ignored by the MPU, and, while the MPU is performing the $\overline{NMI}$ task, it cannot be interrupted again. The Interrupt Request ($\overline{IRQ}$) has second priority. The MPU can be pre-programmed to ignore an $\overline{IRQ}$ altogether. In the 6502, this is done by setting digit 2 of the Status Register to '1'. An $\overline{IRQ}$ task can be interrupted by an $\overline{NMI}$. After completing the $\overline{NMI}$ task, the MPU continues with the interrupted $\overline{IRQ}$ task. When this is completed (assuming there is no further $\overline{NMI}$) it returns to its original program.

Z80 Anatomy

Most MPUs have the same kind of organization, or architecture, as the 6502. The Z80 MPU, which is the processor for a wide range of computers including the TRS-80 Models I and II, the Research Machines 380Z, and the Sinclair ZX81, has a rather more elaborate set of registers. The main set comprises the accumulator (A), the flag register (F, corresponding to the status register of the 6502), and registers B, C, D, E, H, and L, which are general-purpose registers. There is also an alternate set of registers, A', F', B', C', D', E', H', and L'. The MPU normally beings operations by using the main set, but can be switched over to use the alternate set instead, leaving the main set unaffected. It can be switched back on the main set

again.

In addition there are two index registers (IX, IY corresponding to X and Y in the 6502), a stack pointed, and a program counter. In the Z80, IX, IY, SP and PC are double-byte registers (16 bits). Finally there's the interrupt vector register (I) in which instructions for a complex series of vectored interrupts can be stored, and the memoory refresh register (R) which is used in connection with refreshing the dynamic memory of the system. This topic will be dealt with in a later article.

Making The Connection

The typical MPU is contained in a 40-pin DIL package as shown in Fig. 5, which uses the Z80 as an example. It requires a regulated 5 V DC supply which is applied between pins 29 (system ground) and 11 (+ 5 V). The clock circuit supplies pulses at 2 MHz in the case of the original Z80 MPU. The newer Z80A can operate with a clock rate up to 4 MHz. The c lock signal may also be taken to peripherals; for example, the circuits which control the monitor.

The eight data lines, D0 to D7, come direct from the data buffer (Fig. 2). These may act as inputs or outputs, though not in both capacities at the same time. The data bus is taken to the peripherals, to allow for transier of data between these and the MPU. In order that the peripherals will know which one (and *only* one) of them is to receive or transmit data, each peripheral is also connected to the address bus. This is a set of 16 lines, A0 to A15. Address lines are outputs from the MPU. By putting various combinations of highs and lows on these lines the MPU can indicate which peripheral it is addressing. The peripheral may be a printer or a relay on a control board. It may be a single location in memory. Since there are 16 lines, there are 65536 possible combinations of highs and lows, this being the maximum number of locations which can be directly addressed. This figure is usually written in its shorter from 64K, where one 'K' is not 1000, but 1024 ($= 2^{10}$).

Peripheral Procedures

The remaining pins of the IC are connected to control lines which connect the MPU to certain of the peripherals. We will consider the input control lines first. The functions of $\overline{RST}$, $\overline{NMI}$ and $\overline{INT}$ ($=\overline{IRQ}$) have already been dealt with. A low level on $\overline{WAIT}$ causes the MPU to halt its operations. It may have asked a peripheral to send data to it but the peripheral is not ready to put the data on the bus. Instead the peripheral sends the $\overline{WAIT}$ signal, and the MPU suspends action until the peripheral has had time to put the required data on the bus and let the $\overline{WAIT}$ Line go high again. The bus request signal ($\overline{BUSRQ}$) is used by certain peripherals to force the MPU to hand over control of the address bus, the data bus and certain control lines. This is used during an operation known as **Direct Memory Access** (DMA) in which blocks of data are transferred between memory and other peripheral devices without the intervention of the MPU. This is not usually implemented on.the smaller microcomputers.

There are eight outputs in the control bus of which we shall mention only three now, dealing with the rest later as part of specific examples. The Machine Cycle One output ($\overline{M1}$, pin 27) indicates when the MPU is fetching an instruction from memory. Two outputs of special importance are read ($\overline{RD}$) and write ($\overline{WR}$). When the MPU is to receive data from a peripheral it puts the address of the peripheral on the address bus and makes the $\overline{RD}$ line low. This indicates to the peripheral, which is also wired to the $\overline{RD}$ line, that it is to transmit data and not to receive it. When the MPU wants to transmit data to a peripheral, it puts the address on the address bus and make the $\overline{WR}$ line low.

Clocking On

With so many signals being passed in several directions, and with the data bus being required for transmissions into or out of the MPU, it is essential that all these activities take place to a clearly defined schedule. Although micros and their peripherals act at fantastic speeds, these are only fast according to our human scale of appreciation. To an MPU a memory which responds in a microsecond is not particularly speedy. The MPU even has to wait a while to give it time to put the data on the bus, and for the voltages to settle to their intended levels. To keep all sections of the system operating in an orderly way, and to allow the circuits a finite (even if infinitesimal) time to react, the clock is of major importance.

As an example of the way the various parts of the system interact, let us consider what heppens when the MPU goes to memory to find the instruction which it is to execute next. Figure 6 shows the voltage levels on the lines concerned. The top curve shows the regular pulsing of the system clock at, say, 2 MHz. At this frequency, each of the periods T_1 to T_4 is 0.5 microseconds (uS). The MPU beings by making $\overline{M1}$ low, indicating that it is about to fetch an instruction from memory. At the same time it puts on the address bus the address of the memory location in which this instruction is stored. It has obtained this address from its program counter, which has just been incremented following the execution of the previous instruction. The addressed location does not know at this stage whether it is to be read or written to.

On the next low-going edge of clock, the Memory Request line ($\overline{MREQ}$) indicates that this is an operation involving memory (as opposed to a printer, or monitor peripheral, for example). Immediately after this, the $\overline{RD}$ line is taken low by the MPU, indicating that this is a read operation. The $\overline{MREQ}$ signal is used to enable (or 'turn on') the memory IC so that it is ready to puts its data on the bus. Since many such ICs are permanently wired to the bus and since only one can be allowed to put data on to any line at any one time, memories have tri-state outputs. These can be high, low or 'high impedance'. The high impedance state means that the output is virtually isolated from the bus and not able to communicate with it. Outputs are in this state until a $\overline{RD}$ signl is received by the IC. The $\overline{RD}$ signal can be fed to the memory IC so as to make its outputs change to low impedance and take the lines of the bus to high or low states.

As soon as the data has appeared on the bus the CPU reads it into its instruction·register. It has until the next rising edge of the clock to do this. Then $\overline{MI}$, $\overline{MREQ}$ and $\overline{RD}$ are made high, indicating that the operation has been completed. The total time for the whole operation is 1 uS. During the next 1 uS the CPU passes the data along its internal bus to its control, where the data is decoded as an instruction and then acted upon. While this is happening there is no need to take in further data and, since the instruction is still being coded, the time for acting upon it has not yet arrived. In the Z80, this period is used for refreshng dynamic memoris, as will be explained in a later article.

MICROS 2

Bemused by those funny-looking groups of hexadecimal numbers that microprocessors use for a language? From voltages to binary, binary to hex, and hex to English — Owen Bishop does the translating.

In describing the many and complex circuits which go to make up the CPU and its peripheral devices we often refer to **information** being transferred from one to the other. Last month, it was explained that such information consists of either **instructions** or **data**. In order to understand how the several parts of a computer interact to form an operating system we need to go into more detail about these instructions and the data. In this month's instalment we pause from considering electronic aspects in order to discuss the nature of the information and the form in which it is transferred.

Information Technology?

The MPU and other integrated circuits composing the micro respond to only one kind of information. This is an electrical signal, a voltage level present on one of its input lines. To the input circuit of the IC, this level is either 0V (or so close to 0V that it counts as 0V), or +5V (or high enough to count as +5V). An instruction given to a Z80 MPU is received as a set of such voltages on each of the eight lines of the data bus. For example, the Z80 might receive a set of signals as set out in Table 1. When (and only when) these signals are present, internal logic within the Z80 is set and causes every flip-flop of the accumulator register to change state. Every 'set' flip-flop is reset, every 'reset' flip-flop is set.

The description above is in purely electronic terms, which is reasonable enough, for a Z80 MPU is a purely electronic device. When we use it in a computer and communicate with it through the agency of a keyboard, using a high-level language such as BASIC, we tend to forget that it is only a rather complicated electronic circuit. In order to make it do anything at all we have to communicate with it by sending it information (ie what we want it to do) in terms of electrical signals.

Coding For Clarity

Having looked at this from the MPU's point of view, let us look at it from ours. We are somewhat cleverer than CPUs and can take advantage of this to make things simpler for ourselves. For example, it is a cumbersome procedure to state: "To make each 'set' flip-flop become reset, and each 'reset' flip-flop become set, place +5V on line D0, +5V on line D1, . . ., and 0V on line D8." We need a way of symbolising the actions we want performed on the flip-flops, and we need a way of symbolising the signal levels to be applied to the data bus. In short, we need **codes** which will relate our requirements to the electronic activities of the computer. Mathematics provides us with ways of coding operations performed on the accumulator or other registers. The information sent on the data bus is coded in another way, known as **machine code**. These codes make it much easier for us to follow the workings of the computer, to tell it what to do and to discover what it has done. They make it easier for us to discuss computer activities among ourselves. But keep in mind that these codes are used for *our* convenience to symbolise events which are essentially electronic.

TABLE 1

DATA LINE	VOLTAGE LEVEL
D0	+5 V
D1	+5 V
D2	+5 V
D3	+5 V
D4	0 V
D5	+5 V
D6	0 V
D7	0 V

Bits and Bytes

As mentioned last month, the actions of the electronic circuits of a computer are binary in nature. Voltage levels are 0V or +5V; intermediate voltages are not recognised. Flip-flops are either set or reset; there are no other stable states. Consequently, the simplest way of symbolising voltage levels is to represent them in binary form.

Conventionally, 0V is represented by the numeral '0', while +5V is represented by '1'. Using this convention, we only have to write out a row of eight such binary digits to represent the eight voltage levels on the data bus. In doing this we use another convention, that the digits refer to lines D0 to D7, in numerical order, written from right to left. We usually split the eight binary digits (or **bits**, a shortened form of the words '**BI**nary digi**T**') into two groups of four bits each. A group of eight bits is referred to as a **byte**. We can write out the voltage levels of Table 1 as a single byte:

0010 1111

A set of bits representing a set of voltage levels which are interpreted by the MPU as an instruction is called an **op code.**

The action of flip-flops is binary too — we can represent the state of the flip-flops of the accumulator or other registers as a set of bits. The accumulator of the commonly-used MPUs has eight flip-flops, so its contents may be represented by a byte. The Z80 and 6502 are of this type. If the contents of the accumulator are, say, '0101 1100', the op code 0010 1111 instructs the Z80 to change every '0' into a '1' and every '1' into a '0', so that the contents become '1010 0011'.

Such an operation has a mathematical name; we call it **complementing**. If we symbolise the accumulator contents as A, and the inverse of its contents as $\overline{A}$, the whole operation can be represented by the equation:

$$A \leftarrow \overline{A}$$

(A is replaced by $\overline{A}$).
We can now state what happens in much simpler (to us!) terms:

Op code 0010 1111 causes operation $A \leftarrow \overline{A}$

To the Z80, it is still a set of voltages causing a change of flip-flops.

Helpful Hexadecimal

Putting op codes into binary form is certainly simpler than referring to voltages in terms of their actual values, but it still has features which are inconvenient. For instance, writing out strings of 1s and 0s is tedious, and readily subject to error. It is not easy to notice the difference between 1001 1101 and 1001 1001, and the situation becomes worse when we are dealing with 16-bit codes. Although many of the early systems used switches to feed in a series of bits to the data lines, we prefer not to have to key in each bit separately. There is a simpler way of doing this.

TABLE 2

BINARY	HEXADECIMAL	DECIMAL
0	0	0
1	1	1
10	2	2
11	3	3
100	4	4
101	5	5
110	6	6
111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

The hexadecimal code is a short-hand way of representing the binary code. More than this, it is a number scale in its own right. Whereas the binary scale is based on powers of two, and has only two kinds of figure (0 and 1), the hexadecimal scale is based on powers of 16 and has 16 kinds of figure. We already have 10 kinds of figure (0 to 9) available in our common decimal scale and, rather than invent six new symbols, we have adopted the first six letters of the alphabet for use as figures. Table 2 shows how this is done.

By using hexadecimal coding we are able to save time and confusion in writing out lists of instructions for the computer (programs). We are also able to make use of a 16-key keyboard for entering these instructions quickly. Against these advantages there is the disadvantage that we begin to lose sight of the binary nature of the operation of the computer. The individual 0s and 1s no longer appear in our reckoning and we are one stage further removed from the working procedures of the MPU.

One source of confusion in using several number scales is that we must be careful to state which scale we are using. For example, the digits '11' may represent 'three', if they are in the binary scale, 'eleven' if they are in the decimal scale, or 'seventeen' if they are in the hexadecimal scale. In this article we will write all binary numbers in blocks of four bits, and begin them with at least one zero. All hexadecimal numbers will by followed by the letter H. Thus '0011' represents 'three', '11' represents 'eleven', while '11H' represents 'seventeen'.

Giving Instructions

A byte can represent any number from zero (0000) up to 255 (0000 1111 1111). It is therefore possible to specify up to 256 different instructions for the MPU, using just one byte. Such a set of op codes is called the **instruction set**. Examples of the instruction set of the Z80 are listed in Table 3. The Z80 can respond to more op codes than listed there; in fact, its instruction set contains over 500 different instructions. This is more than 256, so some of these are coded by using two bytes. Such a large instruction set makes it a very powerful MPU.

On the other hand, the 6502 has a relatively restricted set, consisting of only 148 instructions, yet remains a popular MPU in spite of this. There is no doubt that the limited set makes it easier for a programmer to get to know how to use the capabilities of the 6502 to the full. Needless to say, the op codes used by different types of MPU do not mean the same thing. When a Z80 receives the op code 25H it subtracts 1 from the contents of its H register; in other words, it decrements H. If you present this same op code to a 6502 it performs the logical AND operation on all bits in the accumulator.

Taking Several Bytes

The previous example raises the next point to be considered. An operation such as 'complement' or 'decrement' involves only the register concerned. When we ask the 6502 to AND its accumulator, the MPU must be given something to AND it with. The same applies for instructions such as 'add'. If something is to be added to the accumulator, we must tell the MPU what to add.

TABLE 3

Z80 OP CODE	INTERPRETATION
00	Do nothing
04	Increment register B
05	Decrement register B
25	Decrement register H
2F	Complement accumulator
5A	Load register E with contents of register D
5E	Load register E with the contents of a memory register, the address of which is stored in registers H and L.
76	Halt
FB	Enable interrupts

The instructions which have been mentioned earlier have consisted of a single op code. Most instructions require the op code to be followed by one or more bytes to supply the data upon which the MPU is to operate.

As an example, consider the op code 25H, which tells the 6502 to AND its accumulator with something. When the 6502 has received this instruction, the next byte fed to its data bus inputs must be a byte which specifies what to AND the accumulator with. The 25H op code is what is known as a **zero-page** op code in that the single byte which follows is to be taken as an address located in page zero of memory. Since the address bus has 16 bits, it requires a double-byte to specify any given address. Zero-page addresses in a 6502 system are those which begin with 00H, for example, the address 00 32H. With zero-page addressing we do not need to send the MPU the first byte (00H) for it already knows from the op code that a zero-page address is forthcoming. We need only send the second byte (32H). Thus the full instruction is '25 32', consisting of op code (25H) followed by a single **operand** (32H). On receiving the second byte the 6502 fetches the

contents of address 00 32H from RAM, and ANDs it with the contents of the accumulator, leaving the result of this logical operation in the accumulator.

Using The (Post?) Code

Zero-page addressing is a feature of the 6502 which allows a certain range of addresses to be passed to the MPU in an economical way. As a second example, suppose that the value to be ANDed with accumulator was stored not in page zero (00 32H), but at an address higher in memory, for example, 2D 32H. This address needs *two* bytes to specify it, so after the MPU has received the ANDing op code, it must wait for the next two bytes before proceeding to carry out the instruction. The op code for this procedure is now 2DH (0010 1101) instead of 25H (0010 0101) as before. Looking at the binary code (which is what the MPU is looking at), we see that data line 3 is at +5V now instead of at 0V, as before. The effect of this is to make the MPU wait for two bytes and put them together to make up the full 16-bit address. Then it addresses that unit of memory, fetches the value stored there and ANDs it with the value present in the accumulator.

This second example is a triple-byte instruction, and would be written out as '2D 2D 32'. The first byte (2DH) is interpreted by the MPU as an op code. The next, though it has the same hexadecimal value, is not taken as an op code but as the first byte of a double-byte address. The MPU is able to distinguish between instructions (op codes) and data (eg addresses) by their context. It is rather like our ability to distinguish between the meanings of the word 'lead' in these two sentences:

1) He mended the lead pipe from the kitchen sink.
2) He played the lead guitar in the local pop group.

We know which way to interpret the word 'lead' by its context; by its relation to the other words which occur with it in the same sentence. Similarly, the op code 2DH and the partial address 2DH give rise to precisely the same set of voltages on the data bus, but they are interpreted differently by the MPU according to what has gone before. Information on the data bus can therefore be an instruction (op code) *or* data (eg an address).

For Immediate Attention

An address is one kind of data, but this is not the only kind. We have mentioned that the MPU fetches a value from a location in memory in order to AND it with the accumulator. This value is transferred from that location to the MPU as a byte on the data bus. Data can include values of this kind, to be used in arithmetical or logical operations, or it might be a value which is part of an instruction. For example, the op code 29H is yet another instruction to perform the AND operation, but this one ANDs the accumulator with the next value to appear on the data bus. This is often called the **intermediate mode**. Thus the instruction '29 16' tells the MPU to AND the accumulator with the value 16H. The MPU knows that 16H is a value, not part of an address, because of its context: it follows the op code 29H.

The op code D0H causes the value which follows it to be added to the value held in the program counter register. The effect of this is to make the MPU jump from one part of its program to another. This op code is only obeyed if a given condition holds true. If the result of the most recent operation has left 00H in the accumulator, the jump is effected. If not, the instruction is ignored.

Talking To The Chips . . .

Ultimately, the only way of sending information to the MPU is by means of voltages on the data lines. The only way that the MPU can send information to other parts of the micro, and to the world outside (including us) is the same. In the simpler micro systems, including most of those specialised systems used in control applications (eg automatic washing machines and other robots), all communication is at this level.

If the system has a keyboard, it is often a 16-key hexadecimal one with perhaps a few additional function keys. Readers may remember the Sinclair MK-14, the popular but primitive forerunner of the ZX80 and ZX81. This had a 16-key keypad and a reset button. Visual output was by means of an eight-digit seven-segment LED display, which displayed the figures 0 to 9 and the letters A to F. Unless you were using the tape-recorder input, one had to load a program by laboriously keying in machine code. To write one's own programs it was essential to master the machine code of is MPU, the National SC/MP 8060.

. . . In Machine Code . . .

Nowadays such systems are mainly used for industrial control applications (eg Acorn System 1) and for development of programs to run on other such systems (eg Softy). Most people prefer to be able to instruct the MPU by using a higher level language, such as BASIC, and to receive its output in graphical form on a monitor screen.

At the input and output to the MPU only machine code can be used. Since it is directly understandable by the MPU, a machine code program runs faster than one in a higher level language; it also takes up less storage space in memory. The monitor program stored in ROM (see next month's article) is in machine code. It is common for complex and lengthy software to be in code, simply to make it possible to cram it into RAM. An example is the SCRIPSIT word-processing program which is being used in preparing this article.

Many games programs such as Adventure games and the several versions of Chess are in code. Great complexity can be packed into a reasonable amount of memory and they do not take long to respond to the player's commands. Many utility programs (eg renumbering programs) are written in code because, although they are short and simple, they must be made as compact as possible in order

As an example of the MPU as a general purpose logic chip, we'll consider one operation. The logical operation called AND can be summarised like this:

INPUTS		OUTPUT
A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

Output Z is 1 if (and only if) both input A AND input B are 1, otherwise Z is 0. Many readers will be familiar with the TTL or CMOS logic gates which perform this operation for two (or more) inputs. The 74LS08 has four two-input AND gates; the CD4081 is its CMOS equivalent. When the Z80 CPU receives the instruction 29H, its logic circuits in the arithmetic logic unit (ALU, see last month) are configured as eight two-input AND gates. Each gate deals with one bit. The eight bits are ANDed simultaneously: if the accumulator holds

0010 1101

And the next byte on the data bus is

1011 0100

then the result of ANDing the corresponding bits is

0010 0100

This result then replaces the value previously held in the accumulator.

to leave plenty of vacant space in memory for the programs on which they are to operate.

. . . With An Interpreter . . .

One factor in the rise in popularity of the microcomputer is the ease with which people unskilled in (or even totally ignorant of!) machine code are nevertheless able to program the MPU and make it do their bidding. This involves the use of a high-level language, the most popular of which is BASIC. When a BASIC program line is typed in, it is stored in memory. To save space, the words are usually coded so that each word or variable requires only one byte; the handbook usually lists the system of coding used. This gives yet another possible interpretation of a byte on the data line. For example, in the ZX81, the code FBH may be interpreted as:

- An opcode — enable interrupts
- A value — equivalent to 251 in decimal
- A code for the BASIC command 'CLS' (clear screen)

The code is interpreted according to context, which includes the mode in which the computer is operating at that time.

Once a BASIC program has been loaded and the RUN command given, the MPU reads the program from memory one byte at a time. It cannot operate directly on the bytes as it reads them. If the byte is FBH, for example, there is no way in which the MPU can directly clear the screen. At this stage the MPU is under the control of a special program called a BASIC **interpreter.** The interpreter contains the complete set of machine code instructions for the operation of clearing the screen. When FBH has been read, the MPU goes back to the interpreter program to find out what to do and how to do it.

Although the MPU is working just as fast as ever, it has to go back to the interpreter program at every step in order to find out what to do. This is why a program takes so much longer to run when it is written using a BASIC interpreter. The interpreting is done line by line and, moreover, has to be repeated every time the program is run.

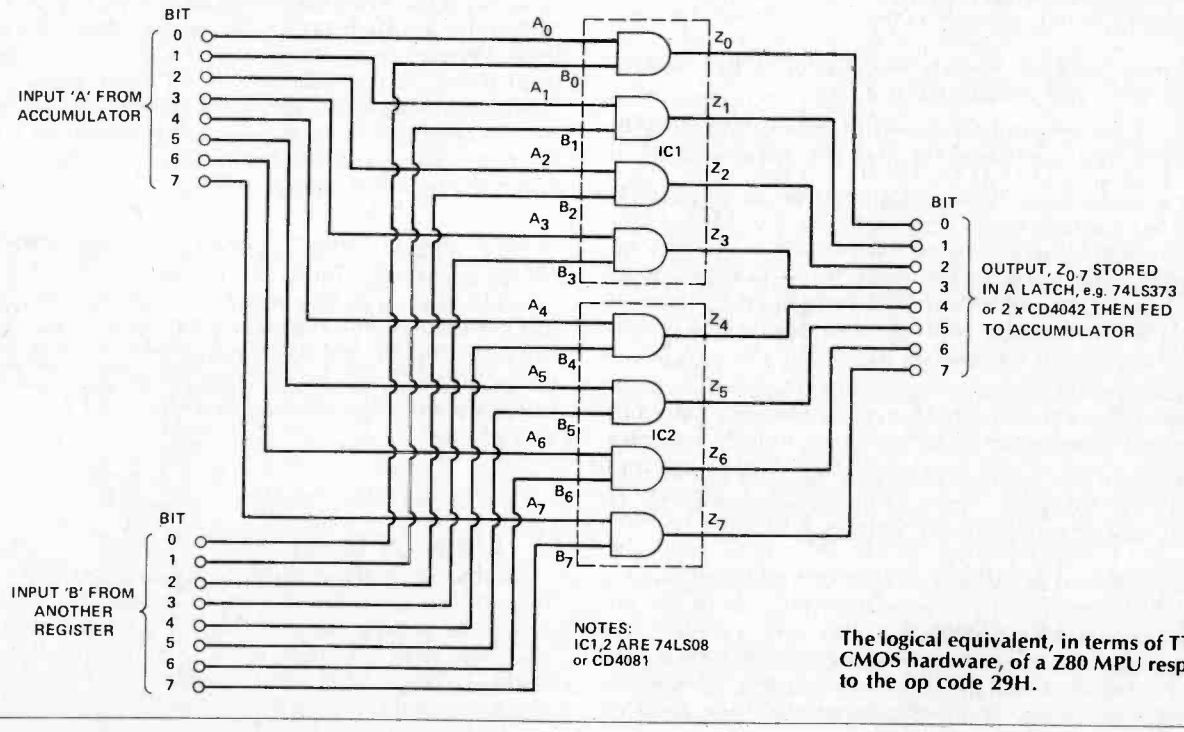
. . . Or By More Basic Means

A faster method is to use a BASIC **compiler** program. Once the program has been written, the compiler goes through it and converts it into machine code. This is done once and for all, after which only the machine code program is used. The difficulty with this approach is that the program must be recompiled if even a small change is to be made. For this reason most people prefer to put up with the slowness of an interpreter.

Many microcomputers have the BASIC interpreter already stored in ROM, so that it is available immediately the micro is switched on. Other micros have no resident language program. If you want to use a high level language, you have to load the interpreter or compiler program into RAM before you are able to enter your own programs in the high level language.

Those who prefer, or are forced for various reasons, to program in machine code can make use of yet another type of program called an **assembler.** Instructions to the MPU are typed out in a symbolic form, consisting mainly of abbreviations of the operations which are to be performed. These abbreviations are generally known as **mnemonics,** which means 'to help the memory'. For example, 'decrement register H' is written as 'DEC H'. When the assembler is run, the op code 25H is assembled into the machine code program. The assembler can convert decimal numbers to hexadecimal, removing a constant source of headaches for the programmer, and can calculate the hexadecimal values required in jumping from one part of memory to another. Provided it is well written, an assembler can be of great assistance, yet it is not too far removed from machine code. The programmer is dealing with particular registers within the MPU and specifying each step of operation of the MPU. If it is *not* well written (and some assemblers are not), then one might just as well learn the commonly-used op codes, have a table of op codes handy, and work out the hexadecimal values on a scrap of paper. Once you have gained a little insight into its peculiarities there is really nothing quite as satisfying as talking direct to your MPU in its own language!

LOGICAL OPERATIONS IN THE MPU



The logical equivalent, in terms of TTL or CMOS hardware, of a Z80 MPU responding to the op code 29H.

MICROS 3

A computer never forgets — provided the information is stored in ROM. We explain the different types and the different applications.

When the structure of a computer was described in Part 1, 'memory' was considered simply as a section of the circuit responsible for storing information (instructions or data). Part 2 showed how this information is in the form of a sequence of binary code groups, each usually containing eight bits (one byte).

The first point to consider is why a computer needs memory. Although the MPU has internal programming which makes it perform a given operation (such as adding together the contents of two registers) when it receives a machine code instruction on the data bus, there is nothing *inside* it to tell it what operation it is to perform next. The operations it performs automatically are very simple ones. Adding together two single-byte numbers is an example already given. Other examples are incrementing a number by 1, decrementing a number by one, transferring the contents of one register to another register, logically ANDing the contents of two registers and so on.

Simple Sums, Complex Sequences

It takes a fairly long sequence of these fundamental operations to perform even the simplest of calculations. For example, to add two numbers which are already stored in memory and to display the result on the monitor screen, the MPU has to:

- Read one number, already stored in memory location X, and store it in its accumulator
- Read the other number, stored in memory location Y, and add it to the number already in the accumulator
- Refer to a table of data already stored in memory to convert this number to its equivalent in the ASCII code. The ASCII code (of which more will be said in a later issue) is a special code used in the majority of computers for representing letters, numbers and punctuation marks in binary form. It takes the MPU several operations to find the ASCII code, and then it has to
- Find out in which screen position the answer is to be displayed and work out which byte of memory the code group must be stored in to achieve this. Finally it must
- Transfer the ASCII group along the data bus to that byte of the video memory, so causing the answer to be displayed on the screen in the correct position.

The procedure above may sound complicated but it is a gross oversimplification of what the MPU has to do. For instance, we have assumed that the two numbers are stored as single bytes, but a single byte can take values only from 0 to 255. Most micros store integers as double-bytes (allowing values from −32768 to +32767). They require four bytes for floating-point numbers (seven-figure precision) or eight bytes for 16-figure precision. To add two such numbers together, the MPU must work with the pairs of bytes in turn (one from each number), adding in any carry-over digit from one addition to that of the next higher pair of bytes. It is clear that even the simplest of mathematical operations requires a long sequence of operations by the MPU. Since the MPU can accept and act on only one instruction at a time, the sequence of instructions is set out in a **program,** which is held in **memory.**

ROM And RAM

At this stage we must distinguish between the two kinds of memory that may be used for instructing the MPU. In physical terms, both kinds of memory consists of arrays of integrated circuits, as will be described later. When you type a program into the computer, or when you load a program from a cassette tape or a floppy disk, it is stored away in a part of **Random Access Memory** (RAM for short). We say it is 'written into' RAM. The information is stored in memory cells (bit-storing sub-circuits), hundreds of thousands of which go to make up the circuitry of each RAM IC. With RAM you can, if you wish, supply the MPU with a different program every time the computer is used. When a program is run the MPU 'reads' the program from RAM. When you switch off the computer, or type NEW, the program is lost. RAM will be the subject of next month's article but, to sum up its main features, we can say its contents may readily be changed, and it loses its contents when the power supply is cut off.

By contrast, the contents of **Read Only Memory** (ROM, for short) cannot be changed or, if changeable, can be altered only as the result of a special procedure, and are not lost when the power supply is cut off. The fact that the contents are not readily changeable is reflected in the name 'Read Only'. In other words, this kind of memory is intended only (or primarily) to be read from, not to be written into.

Kinds Of ROM

Obviously, there must be a way of writing instructions into ROM, otherwise it would contain no instructions and would be totally useless. There are several different types of ROM with different ways of writing instructions into them. To begin with we will look at the type known as the **mask-programmed** ROM.

Each memory cell (each bit) is programmed at the manufacturing stage. Typically, the cell consists of a transistor with its gate either connected to ground or open-circuit. A connection to ground means that the output from the cell is 1; open-circuit gives an output 0. Before the ROM is made, the program which is to be stored in it is very carefully tested to ensure that it is free from error. The masks used for making and linking the components on the slice of silicon are drawn out accordingly. Cells which are to store a 1 have the gate of the transistor grounded. When the ROM is in use and the MPU addresses a particular cell by applying the appropriate combination of voltages to the address terminals of the ROM IC, the cell's output is gated to one of the data lines. This output may then be read by the MPU. Thus the program is permanently built in to the structure of the IC and cannot be altered after manufacture.

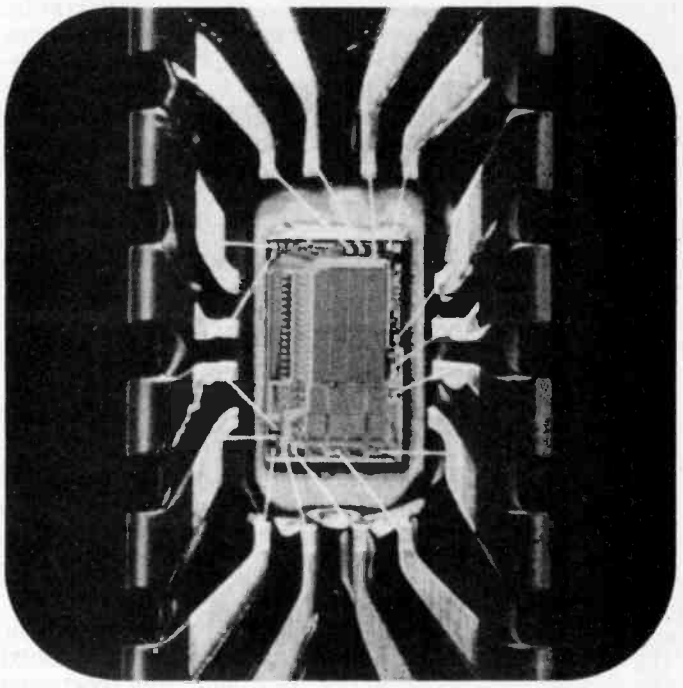
High Volume Equals Low Cost

As might be expected, this procedure is an expensive one, due to the high cost of preparing the special masks. Only if hundreds or thousands of ROMs are to be manufactured with exactly the same program, does the cost fall to a reasonable level. This is the type of ROM generally used for holding the monitor program of a computer and perhaps a resident language, as explained later. Mask-programmed ROMs are made with capacities between 1 kilobyte and 8 kilobytes.

The advantage of the mask-programmed ROM is that once the program has been finalised it is possible to manufacture identically programmed ROMs in large quantities very cheaply. For prototyping and for applications in which it is known that only a few ROMs with a given program are likely to be required, we need a ROM that does not depend on mass-production for cheapness. For this purpose we use a different type of IC, known as a programmable ROM, or PROM for short. There are several kinds of PROM, as will now be described.

Fusible-link PROMs

In a typical version of this kind of PROM the gate of the transistor of each cell is joined to ground by a fusible link. This is a connection which can be destroyed by passing a high current through it. To begin with, all transistor gates are grounded so all the cells in the PROM are set with output 1. When a particular cell is addressed by putting the appropriate combination of inputs on the address lines of the IC and a high voltage is applied to a special terminal of the IC, the link is 'blown'. From then on, that cell gives a 0 output when addressed. Fusible-link PROMs are usually programmed by special electronic PROM programmers which may operate under the control of a computer. The controlling computer holds the program which is to be written into the PROM and coordinates the processes of applying addresses and 'blowing' the links. Once the PROM has been programmed it can not be reprogrammed, for it is possible only to convert 1 to 0, but not 0 to 1. If only a part of it has been programmed, the remainder still being all 1s, the remainder can be programmed on a later occasion. An example of a fusible-link PROM is the Intel 3624, which stores 512 bytes ($\frac{1}{2}$K).



A photomicrograph of a typical mask-programmed ROM chip.

The Ubiquitous EPROM

An erasable PROM can be erased and reprogrammed. The most frequently used EPROM is that which is erased by exposing the chip to ultraviolet light. You can recognise this kind of IC by the window of quartz, through which the chip itself can just about be distinguished as a greyish object about 4 or 5 mm square. Quartz must be used rather than glass, since glass is not transparent to ultraviolet

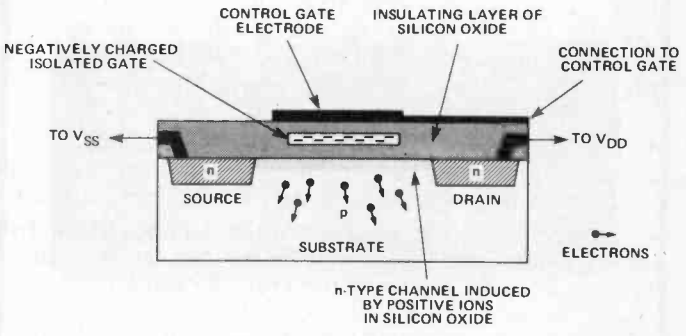


Fig. 1 Diagram of a memory cell in a UV-erasable EPROM. The gate is an N-channel depletion-type MOSFET; here the gate is shown with its negative charge depleting the N-type channel by repelling the electrons in it.

radiation. The UV-erasable EPROM stores information by making use of the extremely high electrical insulation properties of silicon oxide. Insulating layers can be readily formed on the surface of the substrate, simply by oxidising it. In each memory cell, the gate of the transistor has a control gate situated close to it, but separated by an insulating layer (Fig. 1). The gate itself is left unconnected and floats with whatever charge it may acquire.

After manufacture, the gate has no charge, so each cell gives a '0' output. In programming, a high voltage (about 25 V) is applied to the control gate (Fig. 2). Some electrons in the control gate gain sufficient energy to cross the insulating layer and charge the gate of the transistor. Once the gate has been charged, the charge remains for decades, and the cell gives a '1' output. The only way of rapidly removing the charge is to expose the gate to a highly energetic radiation, such a short-wave ultraviolet radiation. This is making use of the photoelectric effect.

A Wavelength That Wipes

The amount of energy carried by a photon of radiation depends on its wavelength; the shorter the wavelength the greater the energy. In the UV range, only short-wave UV photons carry enough energy each to dislodge an electron from the floating gate. Photons of visible light and UV of longer wavelength each carry too little energy so they have no effect whatever. Radiation of shorter wavelength, such as X-rays and gamma rays (which are emitted by certain radioactive elements and produced by a nuclear explosion, for example) are also able to erase EPROMs though these radiations are not in normal use for erasing!

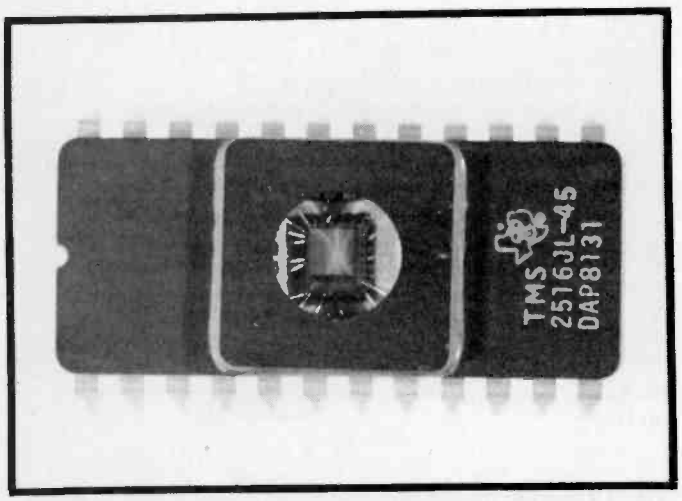
After an exposure to short-wave UV lasting a half an hour or more, the gates of all cells on the IC become discharged. The EPROM is then ready to be programmed again. In this way it has a great advantage over the fusible-link PROM and is widely used in microprocessor systems. Although the EPROM has to be removed from the computer and placed in a special EPROM programmer device (which may be under computer control) the programming, erasing and reprogramming of EPROMs is a straightforward matter. Naturally, it is preferable for the program to be correct to begin with and to have no bugs but, if an error *is* discovered in the program, it can be replaced by a corrected version in an hour or so.

An EPROM is a PROM which is read from most of the time but which can be written into occasionally. The operation of writing generally takes rather longer than the reading operation. For these reasons some people refer to EPROMs as read-mostly memories (RMMs).

EEROMs And EAROMs

When an EPROM is erased the exceedingly small size of the memory cells makes it impossible to pick out any one cell or group of cells for treatment. It is necessary to erase the whole EPROM and program it all again. There are other kinds of PROM which can be erased electrically, known as EEROMs and EAROMs. The EEROMs, or electrically erasable ROMS are similar to EPROMs in that the whole array of cells must be erased, but erasing is achieved by passing a current through the device. EAROMs, or Electrically Alterable ROMs allow us to discharge the gate by a signal applied to the control gate. In this way we can alter any one or more memory cells without affecting the state of the others.

An example of an EAROM is the General Instruments ER3400. Most of these devices employ NMOS transistors and, while they are relatively expensive at present, their cost is beginning to fall, and they will soon be very competitive with the UV-erasable ROMs. It takes so little time to erase and reprogram these devices that it is feasible for their programming to be carried out while they are still plugged in to the computer circuit. The usual power supply at +5 V is required for reading and supplies at other voltages such as −12 V and −30 V are required for programming. We now have the possibility of the computer with appropriate power supplies being able to alter its ROM during the course of running a program; the distinction between ROM and RAM becomes more clouded, though there remains the fact that ROM is permanent (if we want it to be) while RAM is not.



A typical EPROM. The silicon ROM chip can be seen under the quartz window.

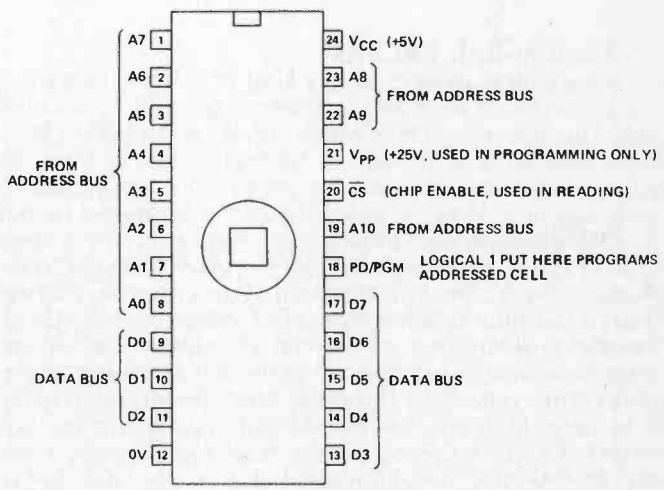


Fig. 2 The pin connections of the 2716, a commonly-used 2K EPROM. During programming the data bus is used to input the required bytes.

Using ROM

Before going on to discuss how ROM is used, we will discuss *why* ROM is needed in a computer. Why not just use RAM which is so flexible and can be readily altered at will? To answer this, let us follow the sequence of events when a computer is first switched on. First of all the MPU is reset. This is usually done automatically by a capacitor connected to the reset input, so that the voltage there is held low for a fraction of a second after all other inputs have reached 5 V (Fig. 3). In addition, most computers have a reset button to allow manual resetting — this is particularly useful if the computer latches up in some otherwise interminable cycle of operations, as it may well do if there is a bug in the program.

When the reset input is made low, either at power-up or by manual resetting, one of the most important results is that the program counter of the MPU is set to a fixed value. In the Z80 and 8505 the program counter is cleared to address 0000. The first thing that the MPU will then do is to try to read an instruction from memory at address 0000. We must make sure that there is an instruction there for it to execute, otherwise it will never be able to do anything. It will be no good typing in instructions at the keyboard or trying to load them from tape or disk. Until the MPU has been told to scan the keyboard for input or to register the signals coming in at the tape or disk sockets you will be unable to communicate with it. What it needs *as a minimum* is a short program to allow it to acquire instructions through the keyboards, or from tape or disk, and store these instructions in RAM (they cannot go in ROM, of course, because ROM cannot be altered). ROM is essential for holding the initialising program which tells the MPU how to get information from the keyboard, tape or disk.

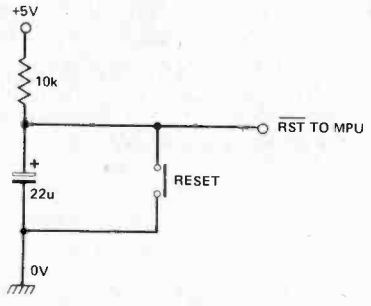


Fig. 3 A simple resetting circuit.

Putting The Boot In

This kind of short program which enables the MPU to get started on its more important tasks is called a **bootstrap program.** It helps the MPU pull itself up by its bootstraps! Since such a program must already be in the computer from the moment power is switched on, the obvious course is to place this program there permanently in ROM. Most computers have additional ROM programs to instruct the MPU how to do other kinds of routine jobs, such as send output to the display. Quite often a message such as 'APPLE II' or 'MEMORY SIZE?' is placed on the screen when the computer is switched on. The program to do this is held in ROM. The complete program may occupy a few kilobytes of memory. Such a program is generally called a **monitor program.** This is another use of the word 'monitor', the name usually given to a purpose-built computer video display (as opposed to a domestic TV set being pressed into service as a computer display).

The monitor program is usually written in machine code (see last month's article), for this is the most compact way of instructing the MPU and allowing it to operate at its maximum speed. Most computer users prefer to communicate with the MPU by using a high-level language, such as BASIC. MPUs do not understand BASIC, so a program is needed to interpret programs written in BASIC and convert them to machine code. Then the MPU can understand what it must do. The interpreting program (or interpreter) can be loaded from tape to disc into RAM but, since such a program is likely to be required every time the computer is used, it is more convenient to hold it in ROM. Thus the ROM of a microcomputer may have, say, 12K of ROM which holds all the routines (in machine code) for converting BASIC commands into the corresponding op codes.

When buying a microcomputer it is essential to find out whether you need to buy BASIC on a disc and load it every time you want to use it, or whether the BASIC is resident in ROM. Usually the memory space quoted for a computer is the amount of RAM it has. A computer which is listed as having 48K will allow you to use almost the whole of that for your programs if its BASIC is in ROM. On the other hand, if the computer has 56K but no resident language, the language may use up 12K of that space when loaded in RAM, leaving you with only 44K for your own programs.

A Change Of Character

Another use for ROM is to hold tables which are to be frequently used by the computer. A good example is the 'character generator' ROM. Before it can put a character on to the screen the MPU must find out exactly what pattern of dots are required to produce the letter, numeral or symbol that is to be displayed. These patterns are held in the character generator ROM. The MPU reads the appropriate pattern from the ROM and sends it to the video area of RAM, causing the character to be displayed.

It is feasible to manufacture several different character generator ROMs, each programming a different selection of characters. There can be different type-faces, or the selection of letters and symbols can be chosen according to the country in which the computer is to be used. For example, Video Inc. manufacture a series of such EPROMS for Apple II including the French, German, Spanish and Katakana (Japanese) alphabets, and one holding mathematical and Greek symbols.

Another type of plug-in ROM which is widely used is that which holds a complete games program, educational program or utility program. Instead of loading the program from tape or disk, the user simply plugs in a module containing a pre-programmed ROM. The Atari and Tandy TRS-80 Color Computer are examples of machines with this facility, as are many of the more specialised TV games machines.

and is located in ROM 2. All three ROMs receive the signals on lines A11 to A0. How can we ensure that only ROM 1 responds to the address 10C7, while only ROM 2 responds to 20C7?

TABLE 1

ROM NO.	ADDRESS RANGE		
	DECIMAL	HEXADECIMAL	BINARY
0	0 to 4095	0000 to 0FFF	0000 0000 0000 0000 to 0000 1111 1111 1111
1	4096 to 8191	1000 to 1FFF	0001 0000 0000 0000 to 0001 1111 1111 1111
2	8192 to 12287	2000 to 2FFF	0010 0000 0000 0000 to 0010 1111 1111 1111

The outputs of the roms to the data bus are **tri-state** outputs. That is to say they normally present a very high impedance; they are virtually disconnected from the bus and are incapable of either sending or receiving signals. Each ROM has one (more in some types) special input known as **chip select** ($\overline{CS}$). The bar over the $\overline{CS}$ indicates that this is an active-low input. When the $\overline{CS}$ input is made low, data outputs go to a low-impedance state and whatever data is present on the set of cells currently being addressed is put on to the data bus. Fig. 4 shows how we control which ROM is to be active at any given time.

Go Low To Go

The function of the circuit is to enable one ROM at a time by making its $\overline{CS}$ input go low. The ROM to be selected is determined by the signals present on the upper four address lines and on a control line which indicates when the MPU is ready to read data. In the Z80 this is the $\overline{MREQ}$ line, which is active-low. The 74LS138 is a 3-to-8-line decoder which is typical of the ICs used for decoding address lines in computers.

Addressing Memory

We have often referred to the MPU addressing a given byte in memory at a particular address, without giving any indication of how this is done. Let us look into this in a little more detail. As an example, consider a ROM (it might be a regular mask-programmed ROM or some type of PROM) which stores 4 kilobytes. Each bit of these 4096 bytes is represented by a memory cell. This very-large-scale integrated circuit (VLSI) therefore contains 32768 memory cells, each consisting of a transistor which is set on or off depending upon whether it corresponds to a 0 or a 1. It also contains the logic circuits required to ensure that when any one of the 4096 possible combinations of voltage levels (the 4K addresses) is put on the lower 12 lines of the address buses (lines A11 to A0), then the eight bits of information stored by the corresponding eight transistors will be gated on to the eight lines of the data bus.

The extreme complexity of such a circuit is difficult to imagine, yet it is commonplace on the computer circuit board. To accommodate a monitor program and a resident language we may need three such ICs, giving a total ROM memory of 12K. Suppose that this is to run from the very bottom of the computer's memory (from address 0000 onwards). The addresses corresponding to the three ICs will be as shown in Table 1. From the binary address it is clear that the lower 12 address lines are responsible for differentiating between all the addresses held in a single ROM. The state of the upper four address lines (A15 to A12) tell us which of the three ROMs is to be addressed at any particular time. Thus the address 4295 (10C7 in hex) appears on the bus as:

0001 0000 1100 0111

and refers to ROM 1. The address 8391 (20C7 in hex) appears on the bus as:

0010 0000 1100 0111

Lines A12, A13 and A14 go to the A, B, and C inputs of this IC. Their states are treated as a three-bit binary input. Outputs Y0 to Y7 are normally high but, provided that the IC is enabled, one of the outputs is low at any one time, depending on the binary input to A,B,C. Thus when A is low, B is high and C is low, this corresponds to 010 (equivalent to decimal 2), and output Y2 goes low. This makes the $\overline{CS}$ input of ROM 2 low, so ROM 2 is enabled while the other ROMs remain disabled in the high-impedance state. The location in ROM 2 which is to be addressed is then determined by the state of the other (lines A11 to A0) of the address bus.

The decoding of inputs A, B and C as described above takes place only if the decoder itself is enabled. It has three enable inputs, G1, G2A and G2B. To enable the chip, G1 must be high and either G2A *or* G2B must be low. You will notice the convention on the drawing that small circles are drawn at G2A and G2B to indicate that they are active low. The outputs also have these circles, for the same reason.

There are several possible ways of using the enable inputs to make sure that the chip is enabled only when address line A15 is low and the $\overline{MREQ}$ line is low too. The solution shown here is to NOR A15 and $\overline{MREQ}$ together and feed the result to G1. This makes G1 high only if both A15 and $\overline{MREQ}$ are low. Inputs G2A and G2B are not used and are grounded.

Let us sum up the procedure of reading from ROM. In order to address any particular cell of ROM, the MPU puts its address on the bus. In the example given, the address will be received along lines A0 to A11 *in all three ROMs.* The states of lines A12 to A14 are to be decoded by the 74LS138 and line A15 must be low to allow the decoder to be enabled. Then the MPU takes its $\overline{MREQ}$ output low to indicate that it wishes to read data. This makes the NOR gate output go low, enabling the decoder. One of its outputs then goes low, enabling *one* of the ROMs. The ROM so addressed puts the data on the bus and this is read by the MPU.

The procedure outlined above must be carried out according to a strict time schedule. For instance, if the decoder acts too slowly, the MPU may be trying to read data before it is there. Some mention of this problem was made in Part 1, and we shall discuss it again in connection with RAM, next month.

ROM At The Top?

Readers who have taken the trouble to look at the circuit diagram of a computer and compare it with Fig. 4 may find that the ROM of their computer does not appear to be decoded to the low addresses in memory. This is the case if the computer concerned is based on the 6502 MPU.

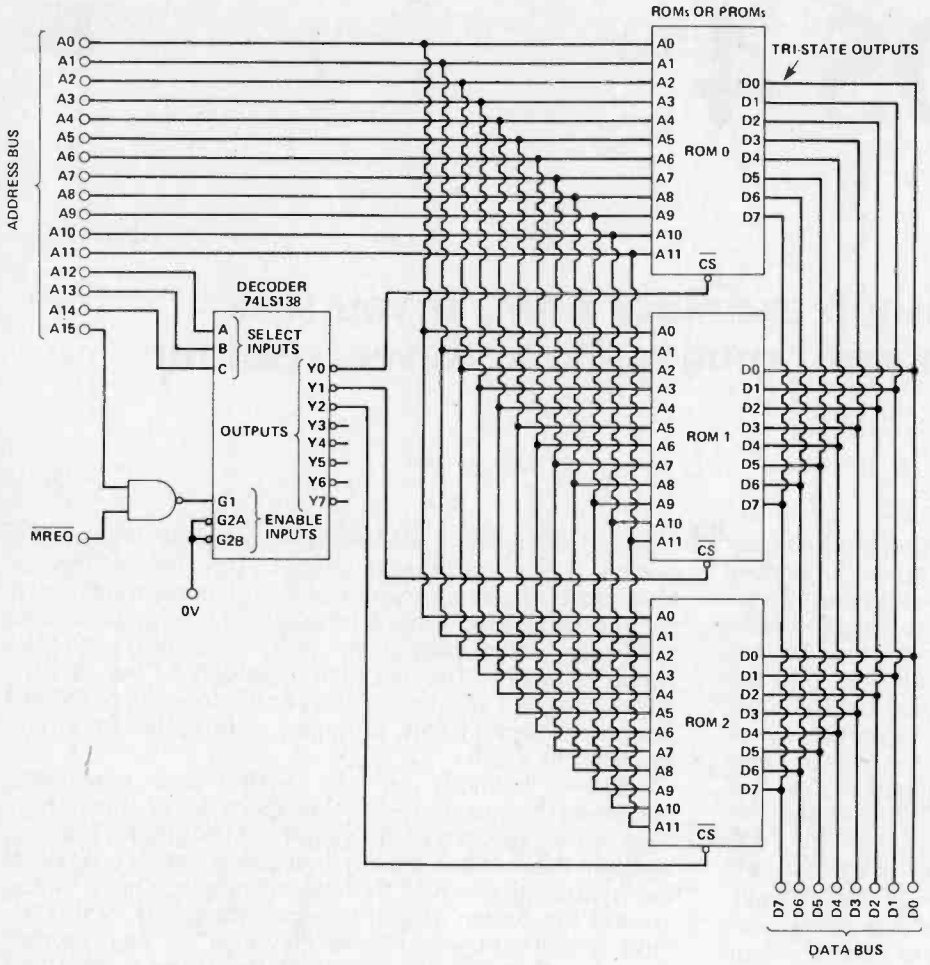


Fig. 4 Addressing three 4K ROMs.

Unlike the Z80 and several other MPUs, the 6502 does not begin reading its program at 0000 after being reset. The 6502 has the special feature of **zero-page addressing;** this means that addresses in the range 0000 to 00FF can be addressed by using the lower byte only (00 to FF). This greatly simplifies and shortens programs, making this area of memory a very useful place in which to store frequently referred-to variables and tables. To take advantage of this facility, this part of memory must be allocated to RAM. Consequently, it is better if ROM is located at the top end (higher addresses) of memory instead. When the 6502 is reset, it first reads the two bytes which are stored at FFFC and FFFD, addresses almost at the top of memory. It is essential to place ROM so that it covers these addresses.

Memory cells FFFC and FFFD in ROM contain the two bytes which are the address of the beginning of the monitor program. In the Apple II with the Autostart ROM, for example, these bytes are 62 and FA, respectively. The 6502, having read these two bytes, sets its program counter to the address they indicate (FA62 in this example) and then goes to that address to begin reading and executing the monitor program.

Integrating ICs

Nowadays there is a move toward reducing the number of ICs required in computer systems. This is particularly important for special-purpose computers that are to be used in control applications, such as those in washing machines or video-recorders. The program required may be relatively small (perhaps only 2K or 3K) so there is no reason why the ROM should not be accommodated on the same slice of silicon as the MPU. If we have ROM, why not have RAM as well and any other useful devices such as input/output ports and timers? A good example of this approach is the Zilog Z8 'computer-on-a-chip'; a similar device is the Mostek MK3870. The Z8 not only has an MPU but also 2K of ROM, 128 bytes of RAM (enough to use as a 'scratch-pad' to hold temporary data), four eight-bit I/O ports, two counter/timers and an asynchronous serial interface. The ROM in the Z8 has to be mask-programmed, so this IC is not one that the hobbyist is likely to be using. The professional can obtain a version of the Z8 or the MK3870 with an EPROM mounted on it in piggy-back style. This can be programmed during the course of development and after all has been settled, the final program can be mask-programmed into the internal ROM of the production version.

A special version of the Z8, known as the Z8671, has the ROM pre-programmed with a Tiny BASIC interpreter as well as its monitor program. This can be used as the basis of a simple computer system. It has 144 bytes of RAM for use as a scratch-pad, but it can address up to 124K blocks of external of RAM or ROM for the storage of programs. This version with its general-purpose BASIC in ROM has a wider appeal than custom-programmed versions so it can be manufactured in quantity. Prices are falling and already several development boards are on sale which feature this IC. The phrase 'chips with everything' can now be taken to mean 'chips with everything on them'!

MICROS 4

In this article Owen Bishop examines RAM. If you seek enlightenment about horned ruminants, however, read no further.

Strictly speaking, random access memory (or RAM, for short) includes every part of a computer's memory which can be read to obtain information and can be written into to store information. In other words, everything that is not ROM (read-only memory — see last month's article) is RAM. This RAM includes not only the arrays of ICs in which information is stored by solid-state circuitry, but also any magnetic cassette tape-recorder or disc drive which may be connected to the micro. Tape-recorders and disc drives will be considered in Part 7 of this series, because, both in form and in function, they are entirely different from the solid-state devices on the computer board. Most people nowadays take the term RAM to cover only the ICs and not the magnetic storage devices.

The name 'random access memory' is a curious one and something of a misnomer. 'Random access' means that the computer can go instantly to any memory cell (a bit) or any group of eight memory cells (a byte) and read from it or write into it. The computer can skip from one location to another according to the program. The situation is analogous to the **random access file,** used in data base systems, and usually stored on disc or tape. The computer can find any location within the file almost instantly and read from it or write to it, without affecting the adjacent locations. This contrasts with the **serial access file,** in which every location in the file must be read from or written into in order, from the beginning of the file to the end.

While the use of the term 'random access' (as opposed to 'serial access') is fairly clear in connection with tiles, even so, it is unlikely that the computer would be accessing items in the file purely on a chance or *random* basis. it usually has a very precise notion of which location it should access on any one occasion. The term 'random access' is even more unsuitable in connection with memory. The computer can, and frequently does, skip about from one part of *ROM* to another, particularly if there is a BASIC interpreter in ROM and it has to go to a different section of ROM to process each command. So ROM is accessed in the same fashion as RAM, and the term RAM makes an inapplicable distinction. A better pair of terms would be ROM (read-only memory) and RAWM (read and write memory), but it seems that we are saddled with RAM and must continue to use it despite its illogicality. A strange anomaly in the world of logical machines!

Where Do You Use RAM . . .

The essence of RAM is that it is *alterable.* You can store information in it, alter parts of the information if required, or replace it altogether with an entirely new lot of information. Some of the main uses of RAM in a micro are:

Scratch-pad This is a (usually) small area of memory reserved for holding information about the state of the system, or where the computer 'jots down' the intermediate results of a series of calculations, ready to be picked up again at some later stage. The scratch-pad can hold such information as the address where the table of variables begins. This is called a **pointer** to the variable table. There wil also be a pointer to the location of the first line of the stored BASIC program, and to other important locations in RAM.

Some locations in the scratch-pad may hold parameters connected with the operation of the system, such as the positions of the margins of the graphics display areas on the monitor screen, the current screen position of the cursor (that small flashing rectangle which moves around the screen as you type), or the name of the key most recently pressed. There may also be 'flags', which are bytes that indicate certain states of the system. For example, INVFLG at address 0032 in the Apple holds the value −1 if the screen is to display normal text, 0 for flashing text and +1 for inverse text.

In 6502-based micros, such as the BBC Microcomputer and Apple, the scratch-pad is usually located at the bottom of memory (the early addresses 0000 to 00FF). This allows the monitor to take advantage of the faster and simpler zero-page addressing featured by the 6502, as mentioned last month. In other micros the scratch-pad may be at the bottom *or* top, but is usually not in the middle, where it could so easily be overwritten by loaded programs.

Tables of variables This may include arrays and strings, for use in the program.

The program itself This may be in BASIC or some other high-level language, or in machine code. Often small machine-code programs (such as editing or renumbering programs) can be tucked away at one end of RAM, where they will not be disturbed by the main program.

The video RAM This is an area of RAM set aside for holding information about what is to be displayed on the screen. The video RAM is usually near the lower end of memory, perhaps just above the scratch-pad. More about this in Part 5.

Buffer RAM These are sections of memory reserved for holding data temporarily before it is transferred somewhere else. For example, when you type in a line of program, your keystrokes are stored in a **line buffer.** When you press 'Return' or 'Enter' the line you have just typed is tranferred from the buffer to the next vacant locations in the area where the program is being stored. Buffers are useful when data is to be transferred rapidly between the micro and a peripheral device such as a printer or disk drive. A block of incoming information, such as the

contents of a file on disc, are held in a file buffer in RAM. It is then available for use by the main program, and can be replaced by information from other files in due course.

. . . And How Much Do You Need?

The amount of RAM a system requires depends on how many of the functions listed above are to be implemented. A microprocessor control system (such as that fitted in an automatic washing-machine) will have its program in ROM (or PROM). The washing machine has no video, and needs no variable table or RAM buffers, but it may need a scratch-pad on which to keep account of the settings of the controls or the stage it has reached in processing the washing. It needs a very small RAM, simply as a scratch-pad. An IC such as the Mostek MK3805 (Fig. 1) provides just 24 bytes of RAM. The chip also includes a real-time clock-calendar.

At the other extreme many personal computers come with between 16K and 48K of RAM, and can be expanded up to 64K or even more. The great advantage of this is that

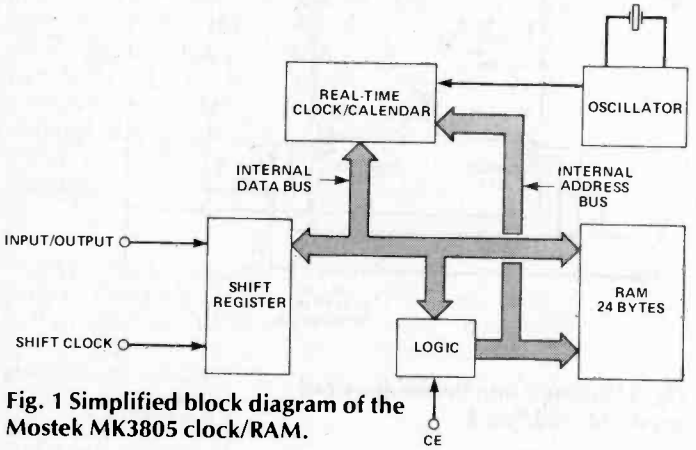


Fig. 1 Simplified block diagram of the Mostek MK3805 clock/RAM.

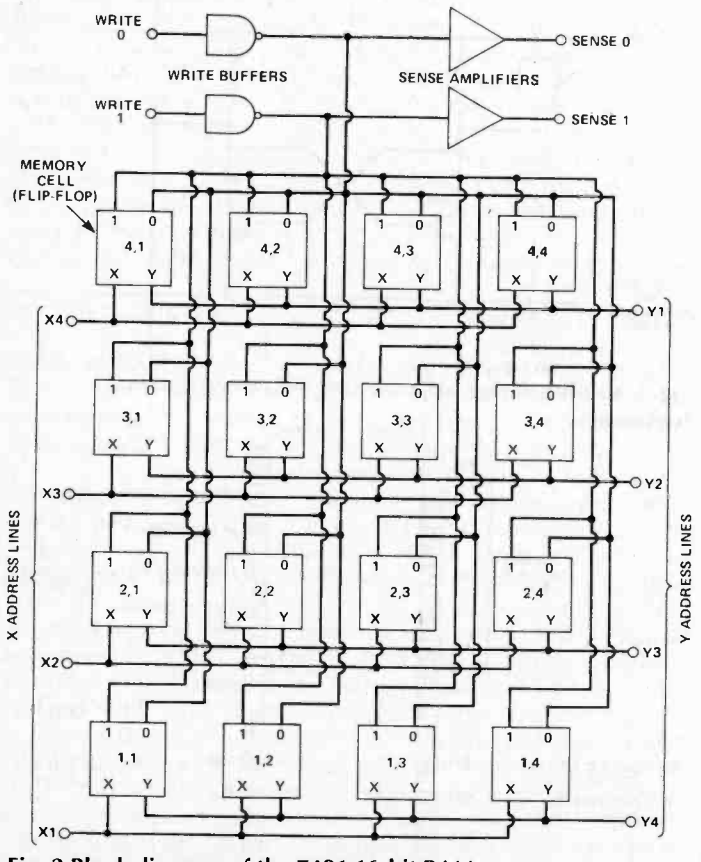


Fig. 2 Block diagram of the 7481 16-bit RAM.

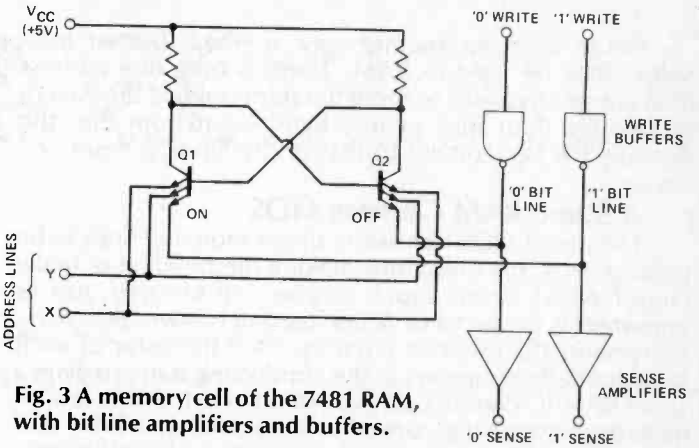


Fig. 3 A memory cell of the 7481 RAM, with bit line amplifiers and buffers.

lengthy programs can be loaded, making it possible for the computer to run anything from a sophisticated accounts program to a complex and perplexing adventure game. It is the steady decline in the cost of RAM ICs which has led to the increasing power and hence increasing popularity of the personal microcomputer. Nowadays the typical micro has enough RAM to do things which formerly only an expensive mini could do. In 1974, the only RAM ICs available on the hobby market were the TTL 7481 and 7489. The 7489 cost £4.95, and contained 64 bits (16 four-bit words), a rate of 13 bits per pound. In spite of inflation we can today buy a 64 *kilobyte* IC for only £13.40, a rate of about 4000 bits per pound.

Bipolar Bistables = Bits

The 7481 and 7489 are early examples of RAM based on bipolar transistors. The 7481 (Fig. 2) has 16 bits, each individually addressable. Each bit is represented by a flip-flop (Fig. 3). Readers will recognise the familiar cross-connected configuration of the transistors, but the triple emitters are a distinctive feature.

Each flip-flop can be in one of two states, one of which represents a stored '0' and the other a stored '1'. In Fig. 3 the flip-flop represents '1' when Q1 is on and '0' when Q1 is off. The address lines X and Y are normally low, and the current through the 'on' transistor (Q1 in this example) flows to the address lines. To address any particular flip-flop the corresponding lines for row and column are made high. The result of this is that for the addressed flip flop, both X and Y lines become high. For the other flip-flops at least one of X and Y is low.

Let us follow the way the stored data is read from the flip-flop. Figure 4 shows what happens. When both address lines go high, the current (if any) through the transistor can no longer flow to the address lines. Instead it is diverted to the bit line, in this example the '1' bit line. The current is detected by the sense amplifier, the output of which falls from its normal state ('1') to '0'. No current is flowing through Q2, so the '0' bit line and its amplifier are unaffected. It needs only a logic gate and buffer to output the signal from the sense amplifiers to the data bus of the computer. Reading has no effect on the contents of the flip-flop. Q1 stays on due to the current flowing to the '1' bit line. Q2 is off and stays off.

Now let us look at the write operation (Fig. 5). Inputs to the write buffers are normally low. To write a '0' at the flip-flop we apply a 'high' voltage to the '0' write buffer and make the appropriate address lines high. The buffers are inverters, so their output is not normally '0'. Now the output on the '0' bit line changes to '0'. Q2 is able to conduct because of the low level on the '0' bit line. Consequently the flip-flop changes state, the stored '1' being replaced by a '0'. Had we tried to write a '1' to this flip-flop, the flip-flop would have remained in its previous state.

Figure 6 shows another way in which bipolar transistors may be used in RAM. There is only one address line, common to four or more flip-flops holding the data of one nibble (four bits) or one byte. Apart from this, the operation is very similar to that of the flip-flop described above.

A Static RAM Gathers MOS

The use of MOS transistors allows more flip-flops to be packed on to the chip, thus making the building of really large memory arrays much simpler and cheaper. Just as important is the fact that MOS has much lower power requirements than bipolar circuitry. One transistor of each bipolar flip-flop is always in the conducting state, so even a small RAM IC continuously draws a current that is tens of milliamps. Supplying current to a large bipolar RAM and dissipating the heat generated are major problems. Nevertheless bipolar RAM has the advantage of very high speed of access (of the order of 20 nanoseconds). It is favoured in input and output buffers of main-frame computers where large amounts of data have to be transferred at high speed between the computer and peripherals such as hard-disc drives.

By contrast, MOS circuits require hardly any current while in the quiescent state. The 5101 CMOS RAM (Fig. 7) draws only 10 uA while quiescent. Even when it is being read from or written into at maximum rate, the current requirement never exceeeds 25 mA. The price for low current consumption is paid in longer access times — of the order of 450 nanoseconds — though some MOS RAMs are faster. However, longer access time is no disadvantage for the typical micro.

Figure 8 shows a typical MOS memory cell. It has the same general structure and connections as its bipolar counterpart, except that it employs two transistors (Q3, Q4) to act as drain resistors. These are easier to fabricate on the chip than ordinary resistors would be.

One of the distinguishing features of solid-state RAM is that it loses all stored information when the power is switched off. Provided that the power supply to the micro has provision for covering brief interruptions of the mains supply, this is not a problem. In portable computers which are to be used in the field, and in pocket calculators, it may be desirable that stored information should be retained while the power is off. To retain the information in RAM, some kind of battery back-up is needed for the RAM section of the computer circuit, though power to the display and peripherals can be completely shut off. Here again, MOS circuitry has its advantage of microamp power consumption in the quiescent state (ie when not being used). A small battery retains information in memory for weeks or months. The 5101 has the additional feature of requiring only a 2 V supply to retain memory, though it normally operates on 5 V.

A Wee DRAM? It's Refreshing

The devices described above all belong to the class known as static RAM. Once a flip-flop has been set to a given state, it remains in that state until the power is removed. It is **static.**

Modern micros also employ an entirely different type of RAM called **dynamic** RAM. The characteristic of this is that a memory cell does not hold its information indefinitely. After a while (a few microseconds) the stored information fades away. If information is to be retained, it must be renewed or 'refreshed' periodically.

Figure 9 shows the circuit of a typical memory cell. Eight such cells are connected to a single address line, which is normally held low (0 V). The eight cells hold the eight bits which make up a single byte of information. Decoding logic within the IC ensures that the address line goes high (+5 V) when the address of the byte, of which

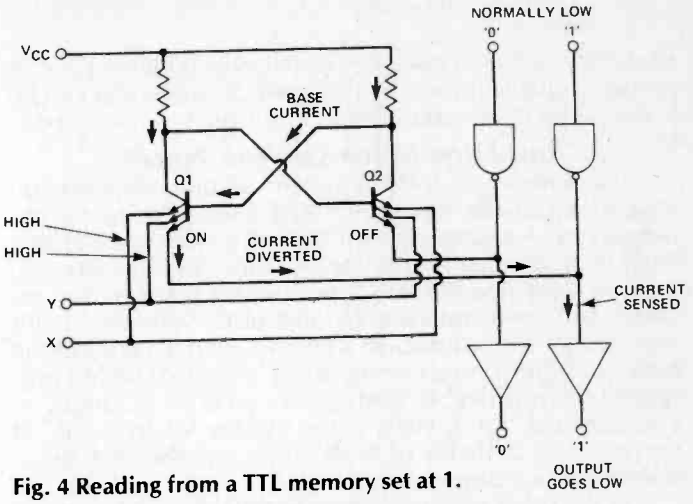


Fig. 4 Reading from a TTL memory set at 1.

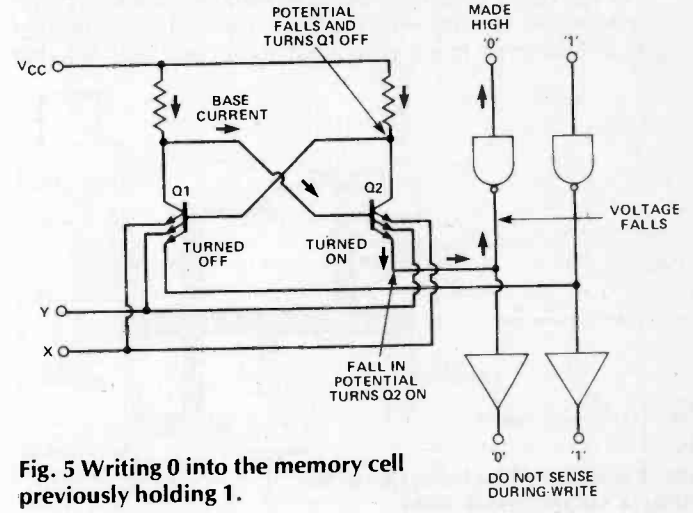


Fig. 5 Writing 0 into the memory cell previously holding 1.

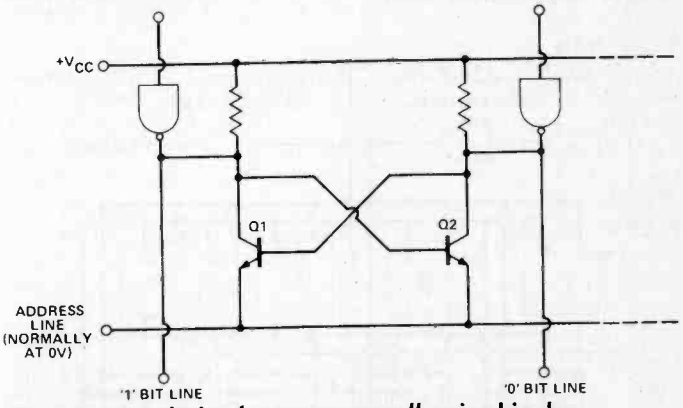


Fig. 6 Another design for a memory cell, using bipolar transistors.

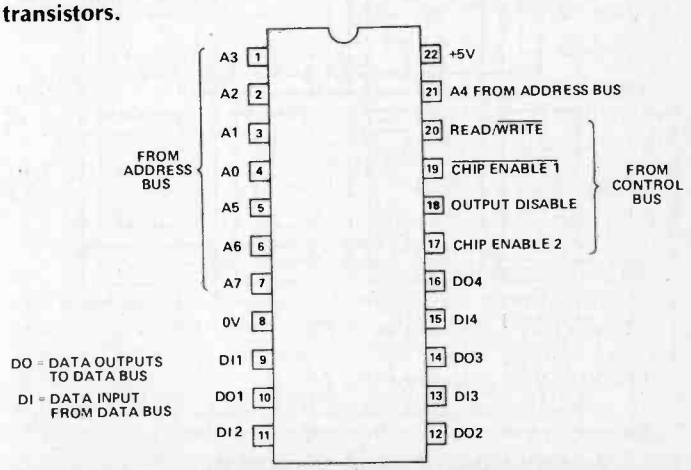


Fig. 7 Pin outline of the 5101 CMOS static RAM.

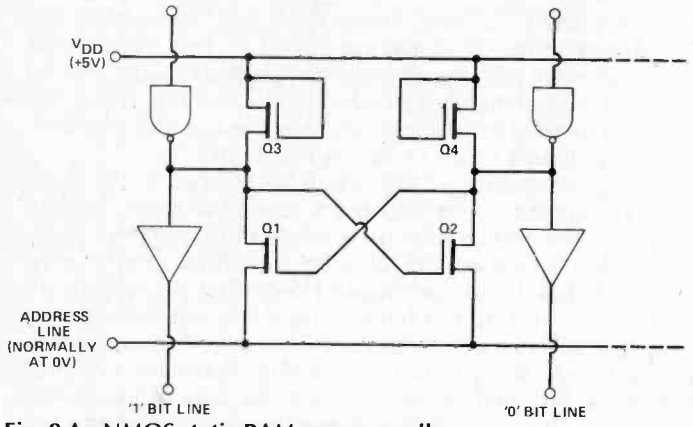


Fig. 8 An NMOS static RAM memory cell.

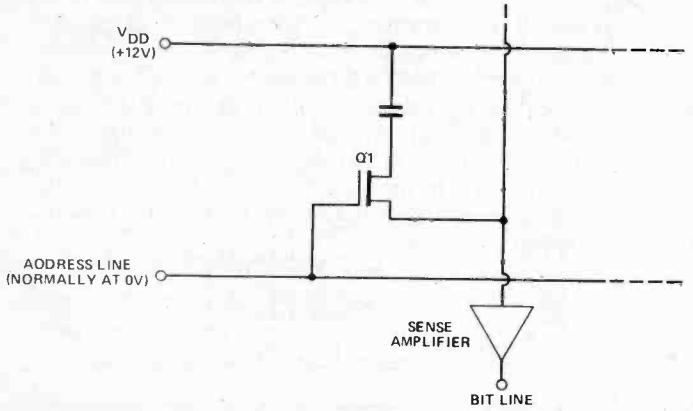


Fig. 9 A MOS dynamic RAM memory cell.

this cell is a part, is present on the address bus of the computer. Each of the eight cells is connected to a different bit line, one corresponding to each line of the data bus. Note that there is only a single bit line, not a '0' bit line and a '1' bit line.

When data is to be written into the cell, the address line goes high, turning on the transistor. If the bit line connected to that transistor is at 0 V, a potential difference of 12 V develops across the capacitor. If the bit line is at 5 V, the potential difference is only 7 V. The address line then goes low again and the potential across the capacitor remains. The effect of this operation is that the information is now stored on the capacitor. The information can be altered by making the address line high, with a different level present on the bit line.

The information can be read by making the address line high, once more connecting the capacitor to the bit line. Charge present on the capacitor is shared with a sense amplifier connected to the bit line. The amplifier outputs a '1' or '0' to the corresponding line of the data bus.

Left to itself, the capacitor would gradually lose its charge through leakage. It also loses some of its charge every time it is read. Figure 10 shows how the charge is refreshed. The 'switches' are in fact transistors in the control circuits of the IC. When both switches are set to position B they feed back the output of the sense amplifier to the bit line. This is positive feedback so the amplified output instantly restores the charge to its correct value.

The need to refresh RAM every few milliseconds imposes an additional task on the MPU, but the advantages of dynamic RAM (see later) are such that this is acceptable for a system with *large* amounts of RAM. Some microprocessors, such as the Z80, provide a special $\overline{\text{RFSH}}$ output which goes low during the second half of each of code fetch cycle. During the first half of this cycle the MPU reads an instruction from ROM or RAM. During the second half of the cycle it is busily processing the instruction internally before acting upon it. This is then a suitable time for the RAM to be refreshed. The $\overline{\text{RFSH}}$ signal, in conjunction with the $\overline{\text{MREQ}}$ (memory request, which also goes low while $\overline{\text{RFSH}}$ is low) can be used in various ways to instruct the RAM ICs to refresh themselves. To see how this is done, let us look at a commonly used dynamic RAM (or DRAM), the 4116.

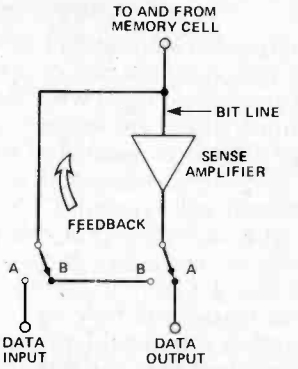


Fig. 10 Dynamic RAM control switching, shown set for data output, Feedback applies when both switches are set to B.

Dynamic RAM Gives Denser Data

This IC well illustrates the great advantage of dynamic RAM. The cells have so few components (compare Figs. 8 and 9) that they can be densely packed on the chip, giving us enormous numbers of cells in a single IC at relatively low cost. The 4116 (Fig. 11) is only a 16-pin device yet it can hold 16 kilobits of information. These are organised as 16K individually addressable bits. In practice we would take eight such ICs and operate them in parallel to obtain 16 kilo*bytes* of memory, one IC corresponding to each bit in the byte (Fig. 12).

A 16K RAM can cover addresses from 0000 to 3FFF (in hexadecimal); in binary this is from 00 0000 0000 0000 to 11 1111 1111 1111.

This means that 14 address lines are required to specify an address. A quick check of Fig. 11 reveals that the 4116 has only seven address input pins! Of course, if the IC had to have 14 address pins, it would need 23 pins altogether, making it physically much larger. We are up against one of the limiting factors with integration. No matter how much circuitry we can cram on to a chip measuring only a few millimetres across, connections to the world outside *must* be relatively large and relatively widely spaced. The case and the pins take up far more board space that the actual chip. Having larger ICs means that we can accommodate correspondingly fewer of them on the computer board, so throwing away some of the advantage gained by high-denisty packing on the chip. The use of seven address pins instead of 14 keeps IC size down yet requires only a little additional logic in the addressing system.

The addressing system is controlled by three signals (Fig. 13); $\overline{\text{RAS}}$ (row address strobe), $\overline{\text{CAS}}$ (column address strobe), and $\overline{\text{MUX}}$ (multiplex). These are obtained from the $\overline{\text{RFSH}}$ (if available), $\overline{\text{MREQ}}$, $\overline{\text{RD}}$ or $\overline{\text{WR}}$ outputs of the MPU in various ways by a simple logic circuit. In a 16K RAM there is only one row and one column, and $\overline{\text{RAS}}$ is identical to $\overline{\text{MREQ}}$. It goes low whenever a read or write operation is in progress. When $\overline{\text{RAS}}$ goes low the multiplexer (controlled by $\overline{\text{MUX}}$) already has the lines A0 to A6 connected to the RAM ICs. The low half of the required address is thus loaded into each IC. Remember that we are trying to load the same address into each of the eight ICs so as to access the eight bits corresponding to the same byte.

Next $\overline{\text{MUX}}$ goes high. This switches the multiplexer IC so that the RAM ICs are now connected to lines A7 to A13. An instant later $\overline{\text{CAS}}$ goes low and the upper seven bits of the address are loaded into each IC. The RAM ICs now hold the complete address and, after a short delay, the appropriate bit can be read or written in the usual way.

In a larger RAM we may have two or more sets

(columns) of eight ICs, each with its own $\overline{CAS}$ line. The appropriate $\overline{CAS}$ line is selected by decoding the two upper address lines (A14 and A15) and combining them with the $\overline{CAS}$ signal from the MPU. Columns which are not being addressed will receive and store the lower seven bits of any address as a result of the $\overline{RAS}$ signal which all ICs receive. Only the addressed column will receive a $\overline{CAS}$ signal and respond to a read or write operation. For other configurations of memory, it may be necessary to have several $\overline{RAS}$ and $\overline{CAS}$ lines to bring different memory blocks into operation by row and by column.

The $\overline{RAS}$ input to the 4116 has an additional function, that of refreshing RAM. When $\overline{RAS}$ goes low, the internal switches are thrown so as to refresh every cell in the IC. Thus during every read or write operation to RAM *all* ICs are refreshed while the low half of the address is being loaded. The $\overline{RAS}$ signal operates for *all* read and write operations, whether these are to RAM or ROM. Thus, even while the MPU is reading a program from its monitor or resident language in ROM, is it still causing its RAM to be refreshed regularly.

One-Chip RAM

The majority of current micros have a 16-bit address bus and are therefore able to address up to 64K. This must include ROM too, so it would be uneconomical and somewhat complicated to use a 64K RAM IC with part of it overlapping ROM. However, with the 64K chip coming into full production (a forecast of 140 million 64K RAM ICs in 1983) for use in minis and mainframes, we may expect to find them in frequent use in micros before long.

With all the address decoding on the chip, the design of the computer board is correspondingly simplified. It has been reported in New Scientist (8 July 1982) that the British firm Inmos has just produced its first 64K DRAM which shows a number of interesting features. One feature is that it automatically refreshes itself, so eliminating the need for special refresh circuitry on the computer board. Also, it operates twice as fast as the 4116.

Another feature is that the RAM carries eight spare rows of memory cells and eight spare columns. Making very complicated circuits on a single chip has the advantage that the connections between different units (eg between RAM and multiplexing and decoding circuits) are all on the chip and do not have to be taken out through terminal pins. This means that the IC need have fewer pins in proportion to the amount of circuitry it contains. Against this is the fact that as we increase the area of silicon on which the chip is made and as we increase the number of components put there, the chance of blemishes and faults rises steeply. It is common to manufacture dozens of chips on a single slice of silicon and, after testing them individually, to reject a high percentage. Obviously a high rejection rate puts up the final cost of the product. With eight spare rows and columns of memory cells, the spare ones can be connected in place of faulty ones after the RAM has been tested in manufacture. This means that the rejection rate falls and eventually the cost of the product can be reduced.

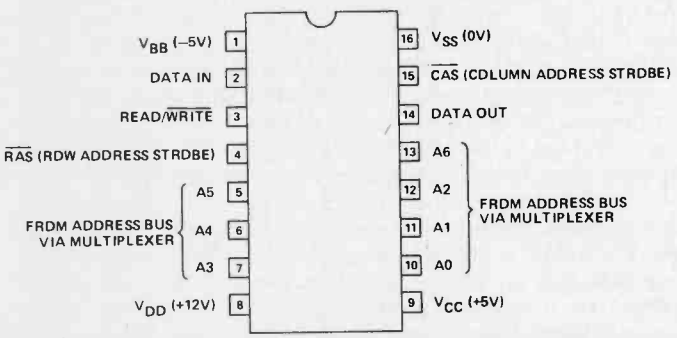


Fig. 11 (Above) Pin connections of the 4116 dynamic RAM.
Fig. 12 (Below) Block diagram of 16K of dynamic RAM.

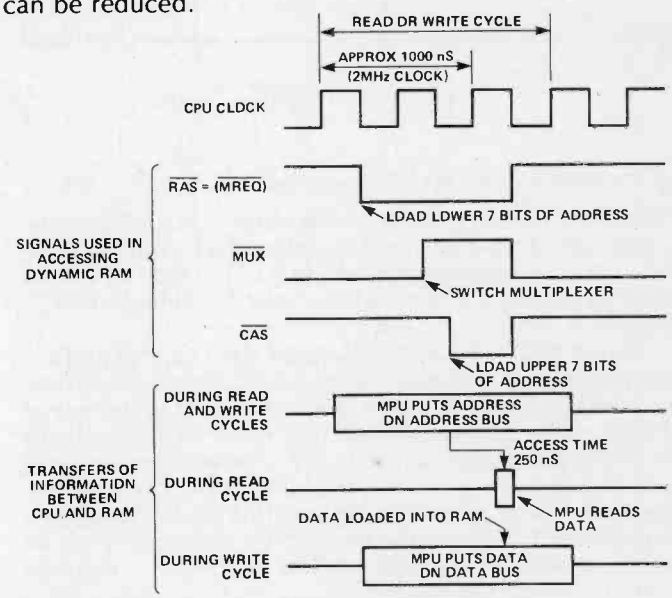
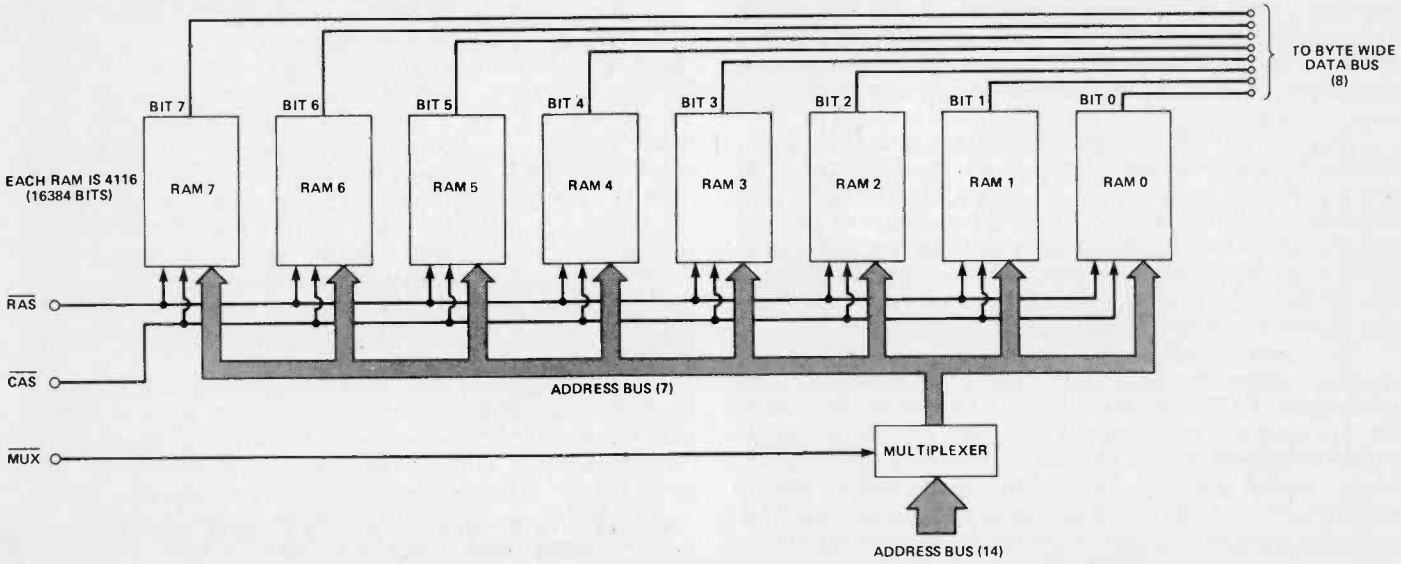


Fig. 13 (Above) Reading or writing to dynamic RAM.



MICROS 5

So far we've covered the brains of a computer, but it's still deaf and dumb, electronically. This time Owen Bishop takes on the role of ear, nose and throat specialist.

The CPU, its ROM and its RAM, the subjects of previous parts of this series, are a tightly-knit section of all computer systems. In most micros, they are mounted together on a single computer board. This month, we are concerned with the way in which this section of the computer circuit communicates with the rest of the circuit and with devices outside the computer proper. This aspect of computer design is known as **Input/Output,** or I/O for short.

In The Right Key

Leaving aside special-purpose computers such as those used in control applications, the most important source of input to the computer is its keyboard. This is where our finger-tips send information (instructions on what to do, and data to do it with) to the computer. As I write this sentence, my fingers are pressing keys on a computer keyboard. Each key is marked with a letter of the alphabet, a numeral or other symbol. There are also a space bar and two shift keys. How does the computer know which keys I have pressed? If I press the fifth key from the left of the second row down, I want it to put 'r' on the screen. If I also press a shift key, I want 'R'. How does it know which key means which letter?

If a keyboard is to provide input to the CPU, it must somehow place information on the data bus. The keyboard of the computer which I use for word-processing does this in a simple way. The method is one which is commonly used in micros at the lower end of the price range. Figure 1 shows the main features of the circuit. The first point to note is that there is a bank of eight **buffers** between the keyboard circuit and the data bus. It would be no good if data were put directly on to the bus every time I happened to touch a key. That might be just the moment when the MPU is reading from RAM. My pressing key 'r' just then could have disastrous results! It is essential that there is *something* between the keyboard and the data bus. This is the function of the buffers.

The buffers are under the control of the MPU. Each buffer has a data input, a data output and an enable input. The keyboard uses eight such buffers and they are all enabled together. When the enable input is held high (+5 V) the buffers are in the high-impedance state: in effect, the outputs are disconnected from the data bus. The buffers are held in this state when the MPU is busy reading RAM, or, for any other reason, does not want to know what is happening at the keyboard. When the enable input is made low (0 V) the outputs of the buffers take the states opposite to their data inputs (they are inverting buffers). The data present at the inputs appears inverted on the data bus lines.

Addressing The Problem

Enabling is under the control of a logical circuit, an address decoder. In Part 3 we described how an address is

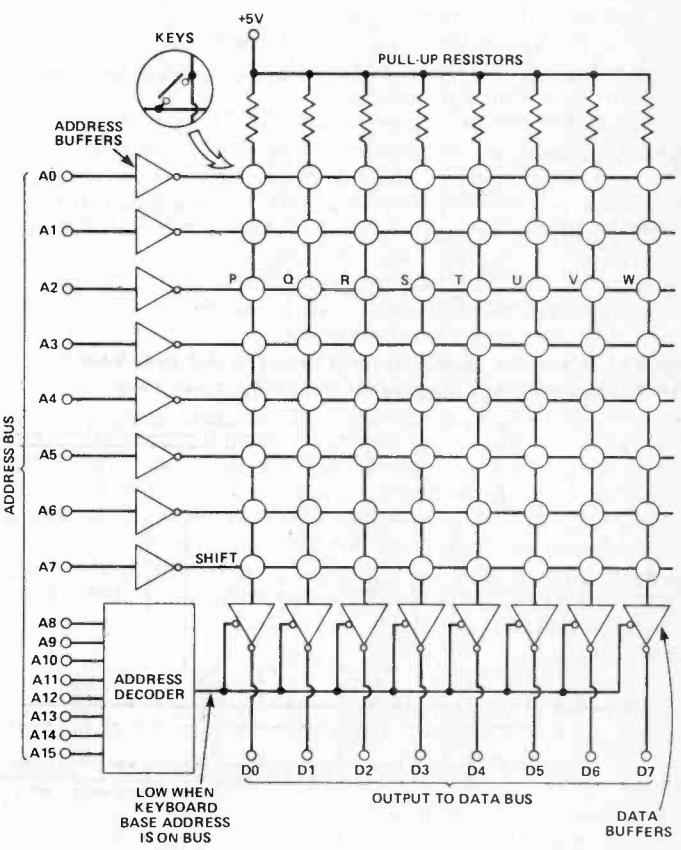


Fig. 1 A typical keyboard circuit. To simplify this, only one row of keys has been drawn.

decoded in order that a particular memory cell in ROM or RAM can be read from or written to. The same technique is used here. Although the keyboard is not memory in the sense that it stores information, it is addressed in the same way as memory. Most addresses are allocated to RAM or ROM, but a few are allocated to the keyboard.

In my computer, the keyboard is addressed at 3800 to 38FF, though only a few of these addresses are actually used. The address-decoding logic gives a low output (to enable the buffers) whenever '0011 1000' appears on the upper eight address lines (A15 to A8). The lower eight address lines (A7 to A0) go to the keyboard matrix. As it enters the matrix, each line goes to a buffer. These are inverting buffers with open-collector outputs.

You will see from Fig. 1 that the matrix consists of eight address buffer output lines crossed by eight data buffer input lines. The keys are simple press-to-make push-buttons, joining an address output to a data input. The buffer input lines are normally held high because of the resistors connecting them to the +5 V supply line. When a key is pressed, an address buffer output becomes connected to a data buffer input. The fact that the address buffers have open-collector outputs means that if a buffer has a low output, it pulls the level down to 0 V. Otherwise the level remains at +5 V.

The Soft Solution

The rest of the input procedure depends on software: the monitor program in ROM contains a routine for reading the keyboard. The MPU addresses the keyboard by putting '0011 1000' (=38 in hex) on the high address lines (A15 to A8) and putting '1' on *only one* of the remaining address lines. For example, to address the first row of keys, the full address is '0011 1000 0000 0001' (=3801). For the next row we have '0011 1000 0000 0010' (=3802), then '0011 1000 0000 0100' (=3804) and so on through 3808, 3810, 3820 and 3840 to 3880 (all hex numbers, remember). The MPU puts these eight addresses in rotation on the address bus. When *any* one of these addresses is on the bus, the address decoder circuit enables all the data buffers. If no key is being pressed at that moment, all data outputs are low. But if one of the keys is being pressed at the same time as its address buffer output is low, a 'high' appears on one of the data lines. Thus if I press key 'r' when the MPU is addressing 3804, line A2 is high, so its buffer output is low. Since key 'r' connects this output to the buffer for data line D2, '0000 0100' (=04 in hex) appears on the data bus. The MPU now has to go to a monitor routine to interpret this data. Using this routine, it finds out that if the data is '04' when the address is 3804, then key 'r' has been pressed. An instant later, it will be addressing 3880 and, if the data becomes '0000 0001' (=01) it can then tell that the shift key also has been pressed, and that the upper-case 'R' is intended.

The MPU continually scans the keyboard in this way when waiting for input, decoding the data according to which address is in force at that instant. This approach to input relies heavily on software, and it takes several operations to detect and decode each key-stroke. Response is relatively slow. The routine required is further complicated by the need to deal with two keys being pressed simultaneously or in very rapid succession. It is necessary to check that a pressed key has been released before attempting to decode the next key that is pressed. This feature is known as two-key rollover. Fortunately, microprocessors work so quickly that even an experienced touch-typist is not able to outpace the keyboard decoding routines.

Encoding Made Easy

Although the circuit described above is simple and cheap to build, the MPU is required to do a lot of work. If this work could be done elsewhere, it would leave the MPU with more time to spend on other and perhaps less routine jobs. The alternative approach to keyboard decoding is to employ a special decoder IC (Fig. 2). Again, the keys are connected at the intersections of a matrix, but now both sets of lines come from the encoder IC. The IC has its own clock circuit and scans the matrix rapidly to find which X line and which Y line have been connected by a pressed key. Having detected a key-press, the output latches of the IC are set to produce a seven-bit code corresponding to the pressed key, taking into account whether or not the shift key or possibly the 'control' key has been pressed at the same time.

You can think of the keyboard encoder as having some of the features of a ROM. When a set of eight memory cells in ROM is addressed for reading by the MPU, its output latches deliver to the data bus the byte stored in that cell. Similarly, the memory cells of the keyboard encoder each contain one code byte. The X and Y lines from the keyboard correspond to address lines. When a particular address is set up by pressing a particular key or combination of keys, the corresponding memory cells place their stored byte in the output registers of the IC. The data stored in the registers remains there until the MPU addresses the encoder. Then its register puts the stored code on the data bus and the MPU reads the code. Note that the MPU only has to perform *one* addressing operation: the keyboard address in the Apple II, for example, is C000. This operation is much quicker than the laborious scanning operation described earlier. The only other thing the MPU has to do is to address the encoder reset (address C001) to reset the latches, ready for them to be set by the next key-press. Note that the encoder *holds* the code until the MPU requests it. In the previously described system, if the MPU is expecting input from the keyboard, it must continually scan the keyboard in case it should miss a key-press.

Ask Me In ASCII

Whereas the code generated by the circuit of Fig. 1 depends on how the circuit is wired, the code generated in Fig. 2 depends on the codes programmed into the memory of the IC during manufacture. In order to promote good communication between keyboards, MPUs and other I/O devices, a standard code has been drawn up for use in computer systems. This is the American Standard Code for Information Interchange, known as the ASCII code (Table 1). Most keyboard encoders produce ASCII code and most computers understand it!

A quick glance at Table 1 reveals that the seven-bit codes cover more than the printable alphabetical and numerical characters and symbols. The first two columns contain what are usually termed **control codes.** These are

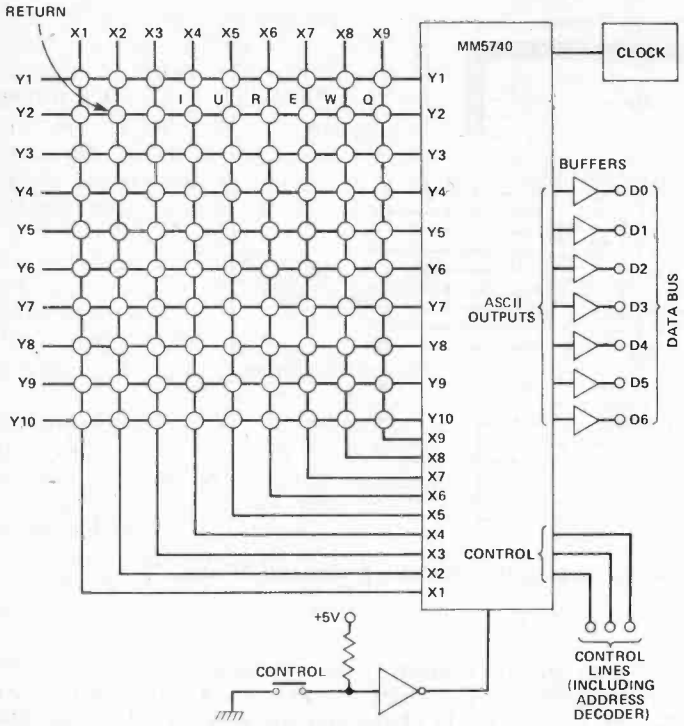


Fig. 2 A keyboard circuit using an ASCII encoder (simplified circuit; only a few keys drawn).

TABLE 1 : THE ASCII CODE

High nibble	0	1	2	3	4	5	6	7
Low nibble								
0	NUL	DLE		0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	EXT	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	↑	n	~
F	SI	US	/	?	O	_	o	DEL

The code is obtained by combining the high nibble (top margin) with the low nibble (left margin) to make a byte. For example the code for upper case W is '57'. The code '20' represents a space.

instructions for the control of peripheral devices, especially printers. They are generated when the CONTROL key is pressed at the same time as one of the alphabetical keys. The code BS, for example, is generated by pressing CONTROL and H, and means 'backspace'. Since this is a frequently used command, many keyboards have a special 'backspace' key (←) which generates this command with a single keystroke. CR means 'carriage return'. When you press the RETURN (or ENTER) key, the keyboard sends a CR code (000 1101) to the computer. This can be used, for example, to tell the computer that the program line which has just been typed in is complete and ready to be stored in program RAM. If the MPU sends such a signal to a printer, it instructs the print-head to return to the left-hand edge of the page. The DC1 to DC4 codes are Device Control codes, available for miscellaneous functions differing from one machine to another. On the TRS-80, code DC4 instructs the line printer to print at 16.7 cpi, whereas on the Apple II it is a toggle instruction to the Silentype printer to echo its printout to the monitor screen.

A further refinement found on some systems is a FIFO, or **first-in-first-out** device. It is wired between the encoder IC and the data line buffers. As each key is pressed, the encoder sends the corresponding ASCII code to the FIFO, which stores it. Typically, it can store up to 16 ASCII codes. The codes are sent out to the buffers in the same order as they are fed in. When the MPU is ready to read a code, a strobe signal to the FIFO results in the next available code being sent to the buffers. In this way, we have **asynchronous transfer** of data between keyboard and CPU. 'Asynchronous' means that the MPU and keyboard do not have to keep in step. If the MPU is temporarily busy and not able to accept input from the keyboard, the data queues up in the FIFO until the MPU is ready to accept it.

Plugging In Peripherals

Now that micros are becoming more commonplace, people are beginning to recognise that they are capable of far more than just playing arcade games or taking charge of the book-keeping. There is an increasing interest in being able to connect external devices to the micro — anything from a simple games control to a robot arm. The more recently made micros, even those in the lower price range, now incorporate ICs which allow a variety of peripherals to be attached. These I/O channels are often referred to as 'ports'.

There are two main types of port IC. The **parallel I/O** device (or PIO) allows data to be transferred between the computer and the peripheral several bits at a time. Commonly there are eight lines, allowing transfer of one byte at a time. The **serial I/O** device (SIO) transfers data a bit at a time, but groups bits into eights (usually) so that a byte is transmitted as a series of eight bits. We will deal with SIOs in a later issue.

Parallel Lines

Although it is only recently that PIOs have become standard on many low-cost micros, they have always been an almost essential feature of the simple computers intended principally for control applications. A well-known example of a PIO is the INS8154 (Fig. 3). Our old favourite, the Sinclair MK-14, had a socket to take an 8154, though the MPU used in this system (the 8060 or SC/MP) has a few direct I/O terminals of its own. Its three 'Flag' outputs can be programmed to have high or low outputs, giving a three-bit data output. The MPU also has ▶

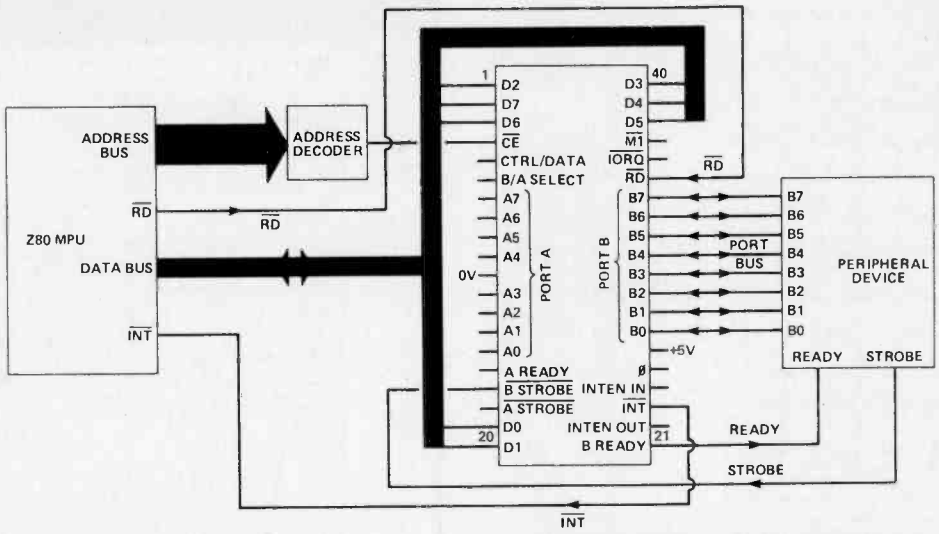


Fig. 4 The Z80 PIO, showing its main connections when linking the Z80 to a peripheral device.

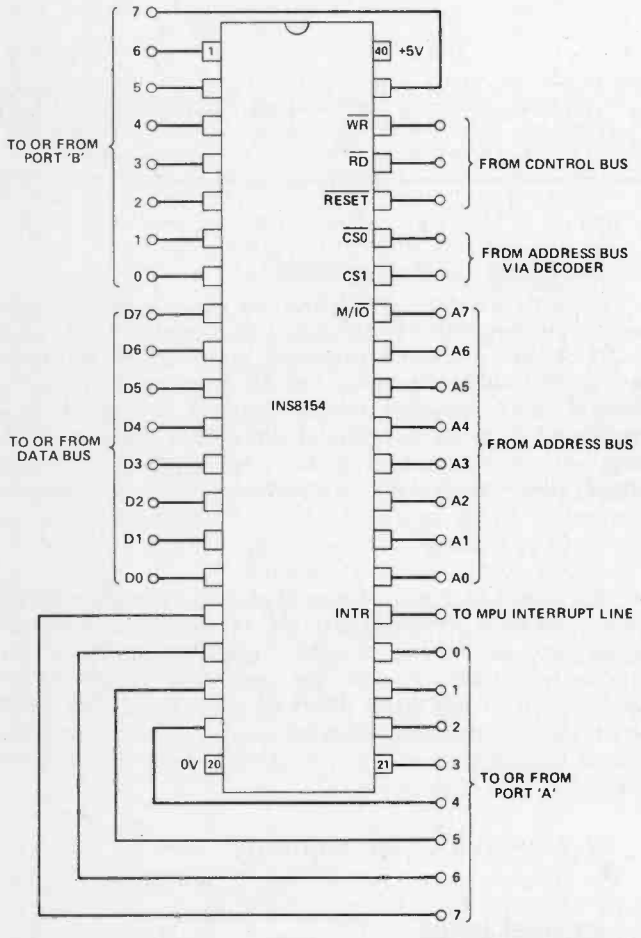


Fig. 3 Pin connections for the INS8154 I/O device.

two 'Serial' inputs which allow two sets of input data to be fed directly to the MPU. This feature of built-in I/O is quite enough for simple control applications and may dispense with the need for a separate I/O IC.

The Acorn System 1 is a well-established control computer. It has sockets for two 8154s, the second of which is used for I/O between the CPU and the cassette recorder. As with the keyboard, an I/O device has to be 'located' in a certain part of memory: we say that it is 'memory-mapped'. When addressing the 8154, the top eight address bits (A15 to A8) are used for establishing the base address of the IC in the way we have already described. The IC has two chip-select inputs, one of which (CS1) is active-high, and the other ($\overline{CS0}$) is active-low. Either or both inputs can be used to enable the chip, making it easier to work out an economical address-decoding circuit.

The M/IO input is unusual, for as well as being an I/O device, the 8154 carries 128 bytes of RAM. This memory/IO combination is handy for control systems, for which 128 bytes may be all the RAM that is needed. The M/$\overline{IO}$ input is usually controlled by line A7. The remaining lines (A6 to A0) are decoded inside the 8154. To operate the 8154 as RAM, the M/$\overline{IO}$ input is made high. If the base address is A000 (as in the Acorn System 1), RAM extends from A080 to A0FF (bit A7 always high for memory operations). To use the IC for I/O the M/$\overline{IO}$ input is made low (bit 7 always low for I/O). This section of the IC thus comes in the range A000 to A0FF. Actually, only a few of these addresses are used. Some of the addresses are used to initiate certain modes of operation; others are used when sending or receiving data. The method of programming the IC is too complex to go into here, but we can outline what it is possible to do.

Data is passed between the CPU and the IC by way of the eight-bit data bus. Data is passed between the IC and the outside world (TTL levels only) by the 16 I/O lines. These are organised as two eight-bit ports, A and B. Each port can be controlled and addressed separately. Reading and writing to the device is totally under the control of the MPU. The registers in each port can be instructed by the MPU to act as outputs, or as inputs. It is also possible to control each line of a port individually, so that some of them are inputs and others are outputs.

When data is being output, it is transferred to the IC and appears on those lines which have been selected as outputs. The data stays there, even though the original

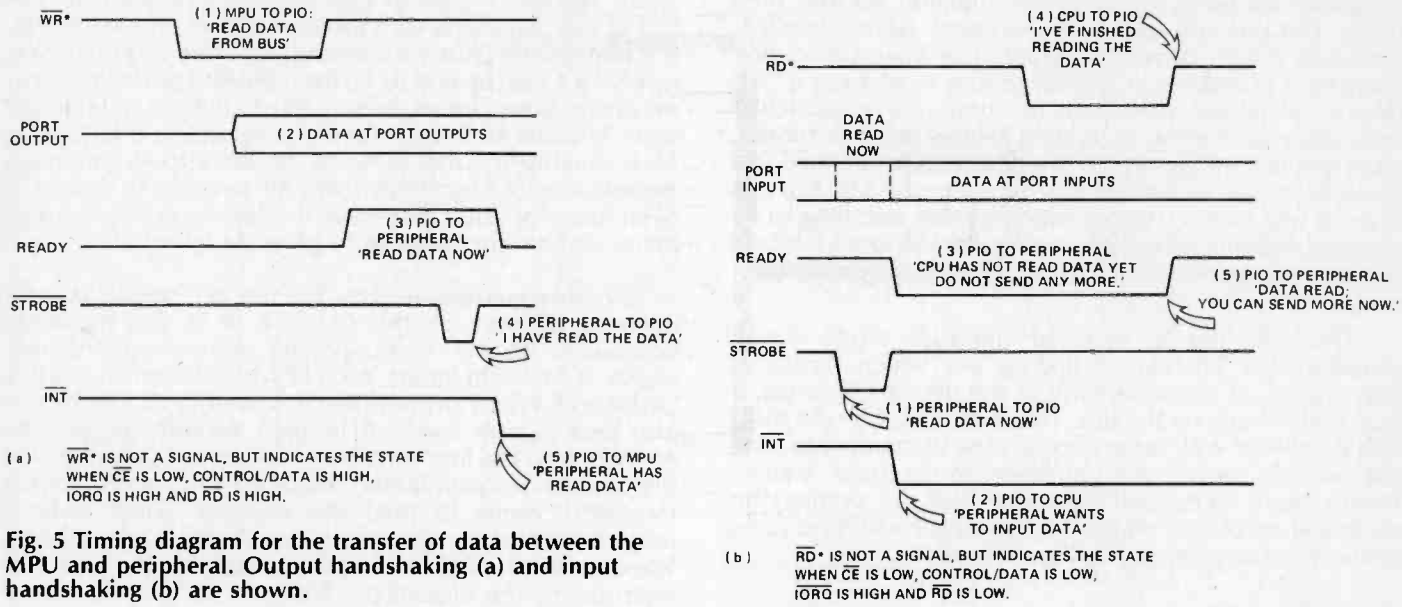


Fig. 5 Timing diagram for the transfer of data between the MPU and peripheral. Output handshaking (a) and input handshaking (b) are shown.

signals may have been removed from the data bus and the MPU is busy doing something else. The data can remain until the external device is ready for it, allowing the asynchronous transfer of data, as mentioned earlier. When the CPU reads from input lines, the data it receives is that which is being transmitted from the peripheral at that instant.

The Hardware Handshake

Obviously there can be problems in transmitting data through an I/O. How does the MPU know that the peripheral has received the data which has been sent to it? It is no use for the MPU to send a new set of data until it is sure that the peripheral has actually received the previous set. Conversely, how does the MPU know that there is a set of data waiting at the input port? How does the peripheral know when this data has been read by the MPU? Again, it is no good the peripheral inputting data to a port if the CPU has switched that port to the output function.

In some systems the sequence of operations and their timing may be such that complete transfer of data is assured. In other systems it is necessary to provide for signals to be sent between the MPU and a peripheral to control the flow of data. This is known as 'handshaking'.

The Z80-PIO (Fig. 4) has special control inputs and outputs and the necessary logic circuits to provide for handshaking. Like the 8154, it has two eight-bit ports, each of which can be individually programmed to act as an input port or an output port. Port A can also be programmed as a bidirectional port, allowing direct communication between the peripheral and the data bus. Alternatively, the individual lines of the port can be set for input or output, as described for the 8154. Figure 5a shows how data is sent from the MPU to a peripheral. As soon as data has been written to the IC and has appeared at an output port, the READY output goes high: this is a signal to the peripheral. When the peripheral receives this signal it knows that it must read the data. As soon as it has read the data, the peripheral puts a low pulse on the $\overline{\text{STROBE}}$ line. This causes the IC to generate a low pulse on the $\overline{\text{INT}}$ line. This goes to the MPU, telling it that the data has been read. The MPU may now send a further byte of data to the peripheral.

When inputting data (Fig. 5b), the peripheral begins by making $\overline{\text{STROBE}}$ low. The $\overline{\text{INT}}$ pulse generated by the I/O device interrupts the MPU to tell it that there is data to be read. At the same time, the READY output goes low, indicating to the peripheral that the data is being held, waiting for the MPU to read it, and that no more data should be sent in the meantime. As soon as the computer has read the data, the end of the $\overline{\text{RD}}$ pulse resets READY, so that the peripheral knows that reading is complete and more data can be sent. Thus the sender and receiver each know which stage the other has reached. Data is transferred between them in either direction without loss.

The 8154 has a similar handshaking procedure but this is limited to port A. The $\overline{\text{INTR}}$ line has the same function as the $\overline{\text{INT}}$ line, but Fig. 3 shows that there are no special control lines to correspond with READY and $\overline{\text{STROBE}}$. Instead, two of the lines of port B are taken over for this purpose when port A is to be used in the handshake mode. The remaining six lines of port B can be used independently, in the usual way.

Dealing With Interruptions

We have seen how the interrupt is an essential part of handshaking by PIO devices. The interrupt may also be used when other peripherals want to communicate with the MPU, either through an I/O device or directly to the data bus. Often, there are several peripherals connected to a system yet all give the same interrupt signal. How is the MPU to know which one of these peripherals it is dealing with?

One method is 'device polling'. Each device has a latch circuit which gives a high output when the device is trying to input data to the MPU. The latches are enabled by an address decoder, and each is separately addressed. When interrupted, the MPU goes to its interrupt routine

program, disabling the interrupt function for the time being: this prevents it being interrupted again while it is attending to the current interrupt. The interrupt routine instructs it to read each register in turn to find out which device is interrupting and to jump to a particular subroutine according to which device has interrupted. Note that this program polls the devices one at a time *in a pre-determined order.* We can program the MPU to test first the registers of devices which cannot wait long to be serviced, leaving other less urgent devices until later. In this way the software establishes a system of **priorities.**

The Z80 has a vectored interrupt mode which simplifies the process of finding out which device is interrupting: at the same time as the device interrupts, it puts certain data on the bus. This data is read by the MPU and combined with other data already in memory to form the address where the appropriate interrupt routine begins. Each peripheral identifies itself by putting this particular set of data on the bus, causing the MPU to jump to the corresponding servicing subroutine.

Who's Shouting The Loudest?

Most I/O devices have two ports, some have three, and many computers have more than one I/O device. If the MPU has two or more peripherals and all are trying to communicate with it at the same time, the situation is like a political meeting with everyone trying to shout at once! There must be a system of priorities so that, when one of the more important peripherals is communicating, the less important ones are ignored. We have seen that software provides priority, but only after the interrupt has occurred. Hardware priority ensures that a high-priority peripheral will always get preference whenever it interrupts. The most commonly used method is known as daisy-chaining.

Daisy-chaining works like this. All the PIOs or other peripherals are connected to the $\overline{INT}$ line by open-collector outputs. The line is normally held high by a pull-up resistor connected to +5 V, but when any one or more interrupt outputs goes low, the voltage on the line is pulled down and the MPU goes into its interrupt routine. In order to be able to generate an interrupt output, a peripheral must be receiving a high voltage level at its interrupt enable input (IEI). Normally, the interrupt enable output (IEO) of the peripheral has the same level as its interrupt input. The IEI on a peripheral receives its input from the IEO of the peripheral with the next higher priority. In Fig. 6, if none of the PIOs are interrupting, every one of them is receiving a high level at its IEI from the PIO next above it in the chain. Every one of them is able to initiate an interrupt when it wants to do so. When a peripheral is interrupting or is waiting for the MPU to respond to its interrupt request, its IEO becomes low. All peripherals below it (with lower priority) then have the low level fed down to them, and are then unable to generate interrupts.

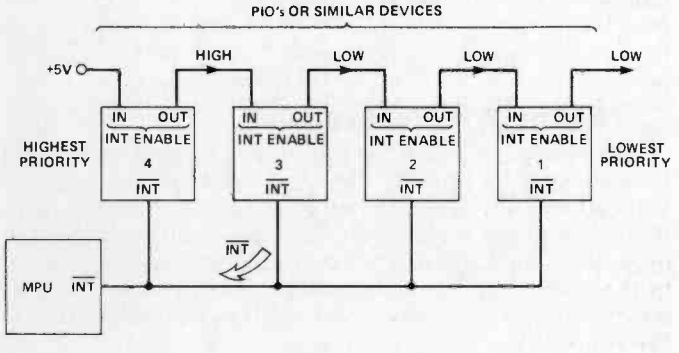


Fig. 6 Daisy-chain priority control: all PIOs are connected to the $\overline{INT}$ line. PIO no. 3 is interrupting and passing a low signal to nos. 2 and 1 to prevent them interrupting.

Another method involves the use of a special priority encoder IC such as the CD4532. It is the hardware equivalent of the device-polling software mentioned above. It has eight inputs, each of which is connected to a peripheral. When any peripheral is causing an interrupt, it also puts a high level on its own encoder input. The encoder also has four outputs which can be connected to the data bus through buffers which are enabled whenever the MPU wants to read the encoder. Their outputs indicate in binary code which peripheral is interrupting. For example, if peripheral no. 6 (connected to input 6) is interrupting, the outputs put binary code 6 (0110) on the data bus. By reading the bus, the MPU can find out which device is interrupting. If more than one peripheral is interrupting at the same time, the binary code for that with the higher priority (highest number) appears at the output.

Sending A Cable

We have been so preoccupied with logic that we have largely ignored one of the main problems of the input and output of data — the wiring between the computer and the peripheral. If this is to be long, special line-driving buffers must be employed though, if the computer and equipment are in the same room, this is rarely necessary. Computers work so fast that electrical signals can travel only a few centimetres during one cycle of operation. If wires are long, it may be impossible for the computer and its peripherals to remain perfectly in step with one another. This is one of the reasons for employing I/O ports with asynchronous interchange of data, as described above.

A more practical problem is the sheer number of conductors required. An eight-bit connection (the minimum commonly use) requires eight lines, plus a ground line and probably several control lines as well. There is a wide variety of multi-way connectors available for joining cables to computers and peripherals. Most are designed for use with ribbon cable.

Electromagnetic interference between adjacent conductors is a serious problem, especially with long runs of cable, and can lead to errors in the data being transferred. The data signals themselves are not so likely to interfere with each other, since they are all put on to the lines at the same instant, and there is a short period before they are read (again, all at the same time) during which switch-on and switch-off disturbances can settle. However, if the cable carries control signals, which are generally *not* turned on and off at the same time as data signals, these may interfere with the data carried in adjacent conductors. One solution is to ground alternate conductors, and use only those between them. A better solution is to use twisted pairs: one wire of a pair is used for the signal and the other wire is grounded. Special ribbon cable is made with twisted pairs with untwisted regions spaced along it, where it may be cut and linked to connectors using insulation-displacement.

MICROS 6

The previous article explained how the computer handles input and output of data but how many of us read and write in binary? This time Owen Bishop explains the operation of more user-friendly interfaces.

Quite a number of electronic circuits produce their output in analogue form, and it is beyond the ability of a computer to read such data unless it is first converted to digital form. Examples of devices with analogue outputs are electronic thermometers, pressure transducers, audio amplifiers (for speech recognition etc) and indeed any device which produces an output voltage varying over a predefined range. Even a simple carbon pot can be included in this list. The Games Controller of the Apple II, for example, uses a 150k potentiometer (there is also a push-button connected to a memory-mapped latch, but this is a digital input). The position, or setting, of the potentiometer is the analogue quantity to be measured. The computer has a quadruple timer IC and, when the Games Controller is plugged in to the computer board, the pot becomes part of the RC circuit of the timer. When the MPU is to read the setting of the Controller, it first triggers the timer (the trigger input is memory-mapped) then measures the length of pulse produced. It does this by reading the memory-mapped output over and over again, counting how many times it reads 'high' until it eventually reads a 'low'. The number of 'high' reads is approximately proportional to the angular setting of the control knob. The analogue-to-digital conversion is crude and far from linear, but certainly good enough for its intended application.

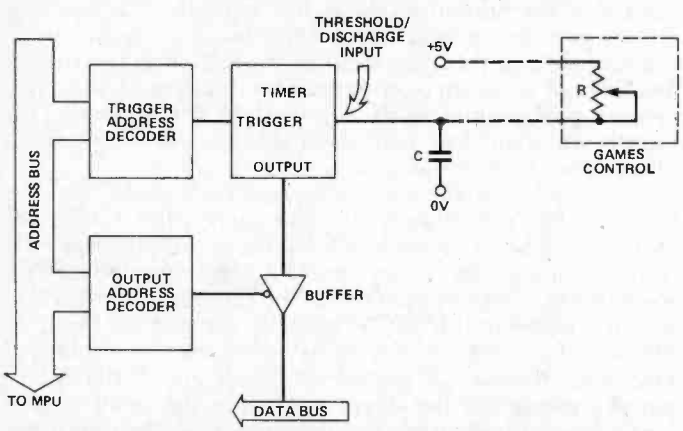


Fig. 1 Analogue-to-digital conversion for a games controller.

A-to-D Conversion . . .

Many computers have on-board A-to-D ICs such as the National Semiconductor ADC0801 (Fig. 2). This converts any input voltage in the range 0 V to 5 V to digital output in the range 0 to 255 (00 to FF in hex). The heart of the IC is a chain of resistors in series, with 5 V across the ends of the chain.

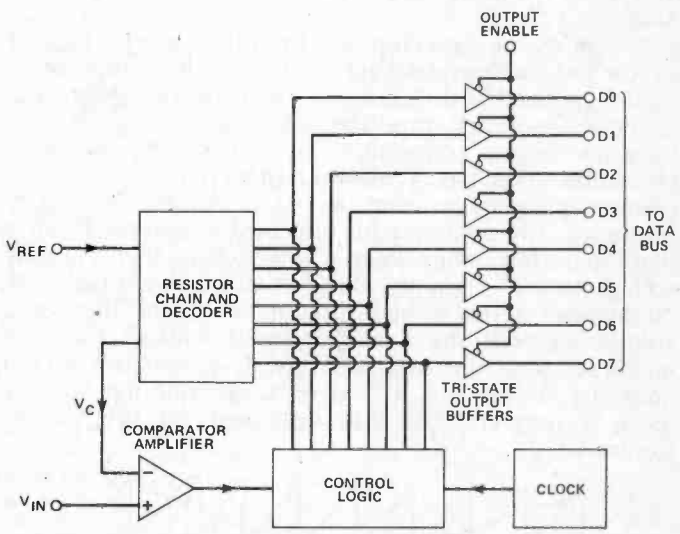


Fig. 2 Simplified block diagram of an A-to-D converter such as the ADC0801. V_{REF} is an on-chip or external reference voltage. V_C is the output voltage from the resistor chain, which is compared with V_{IN}, the analogue voltage which is to be converted.

Internal logic controls CMOS analogue switches which switch resistors into or out of the chain, so producing a voltage (V_c) which can range from 0 V to 5 V in 256 steps. At each stage a comparator matches the output of the chain against the analogue voltage (V_{IN}). The largest resistor is switched in and out first, to determine if the input is less than or greater than 2V5. Then the next largest resistor is switched in and out to narrow the possible range to within 1V25. At each stage the chain output and analogue input are matched more and more closely. After eight attempts the closest match will have been found. The logic signals which have produced the match are then used to set the eight output buffers to one of the 256 possible combinations which can be read as a byte (0 to 255) by the computer. This IC converts the voltage with true linearity, with an accuracy of half a step in 256 steps and takes only a few hundred microseconds to do so. If we want greater accuracy, there are similar A-to-D ICs with a 12-bit output. Note that it is the converter which does the work: the MPU only has to read the result. We do not need to write software to instruct the MPU to measure pulse lengths as with the Games Controller. This saves time and simplifies programming.

Most A-to-D ICs have ways of altering the span of the input range, so that voltages from, say, 0 V to 2 V produce the full-scale output range, 0 to 255. In addition you can

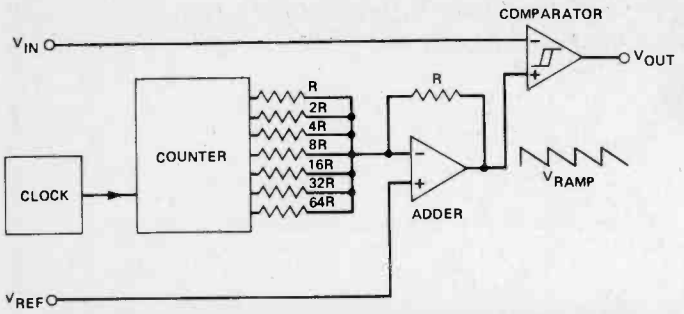


Fig. 3 Simplified block diagram of a simple A-to-D converter, the 507C. V_{REF} is an on-chip or external reference voltage. V_{IN} is the analogue voltage to the converted; V_{OUT} is the square wave output.

adjust the offset so that, for example, you obtain a reading of 0 when the input is 10 V and reading of 255 when the output is 12 V.

As might be expected, a A-to-D IC is a sophisticated circuit and is correspondingly expensive. If an application requires several analogue inputs and high conversion time is not of paramount importance, the inexpensive 507C IC provides linear conversion in 1 mS with seven-bit resolution. This has a resistor ladder (Fig. 3), and the counter supplies current to each resistor in binary sequence. The op-amp adds the currents and the result is that the output ramps down a voltage from 0.75 x supply voltage to 0.25 x supply. If the enable input is high, the comparator gives a high output whenever the ramp voltage exceeds the analogue input voltage: thus the length of time the output is low is a measure of the analogue voltage (Fig. 4). The MPU can find this time by using a program like that described for the Games Controller.

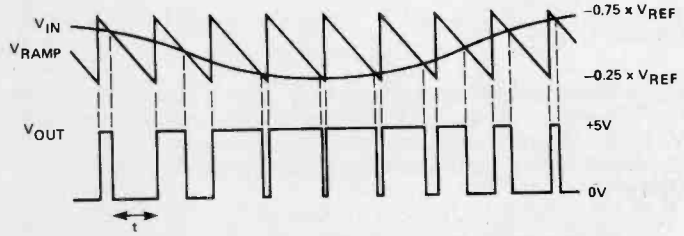


Fig. 4 The input and output voltages of the 507C A-to-D converter. The computer measures t, which is proportional to V_{IN}.

. . . And The Converse Conversion

If a peripheral needs an analogue signal to control it (eg controlling the speed of a motor), we need circuits which can convert the digital output of the computer into its analogue equivalent.

The ZN425 D-to-A converter makes use of an R-2R ladder (Fig. 5). The switches are under the logical control of the eight-bit input. As the count increases, the output voltage increases in proportion (see panel). To drive a low impedance device the output must be buffered by an operational amplifier.

This IC may also be used for A-to-D conversion, using its binary counter. The counter is clocked by an external pulse generator, and as it counts the pulses, switches controlling the R-2R ladder are closed and opened in a binary sequence. The output is a staircase ramp of 256 steps. An external amplifier compares the output from the ladder with the analogue voltage which is to be converted. When the ramp output equals the analogue voltage, the output from the amplifier inhibits the clock. At this point the logic output which controls the switches can be read as an eight-bit equivalent of the analogue input.

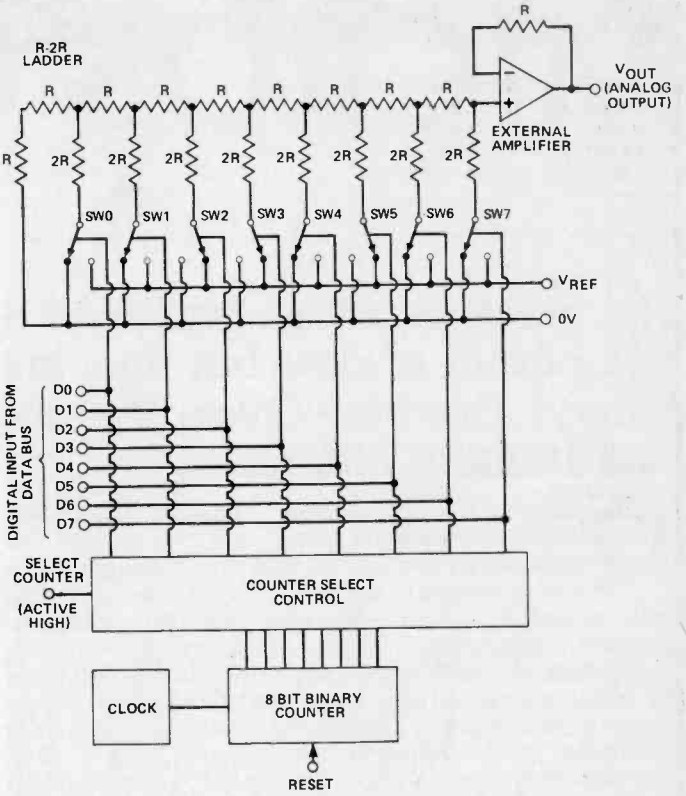


Fig. 5 Block diagram of the ZN425E D-to-A converter. V_{REF} is an on-chip or external reference voltage. The amplifier is not on the chip but is an external op-amp (eg 741).

Screens And Printers

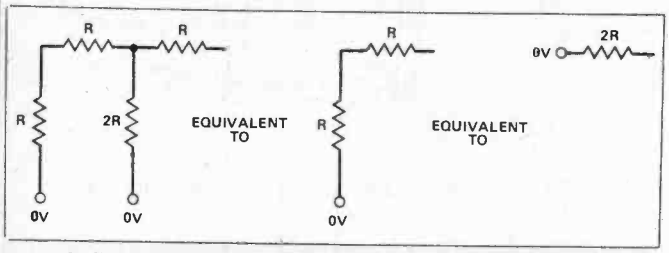
The way most owners receive information from their micros is through the screen. This may be a domestic TV set or a monitor unit specially designed for the purpose. Those whose main interest lies in arcade games, adventures, and the like usually need no more than the screen, but anyone with an interest in programming soon finds that the screen alone is not enough. There is the tedium of copying long listings of favourite programs from the screen, and the frustration of being able to see only a few lines of program at any one time. Sooner or later, the serious programmer adds a printer to the system. This month we shall deal with both these ways of receiving information from the micro.

The fact that the technology of high-speed (for the time) printing was already available in the form of teletype machines, lead to the early mainframe computers having a printer but usually no screen. We are reminded of this early use of teletypes by some of the curious control codes which abound in the ASCII character set (see last month's article). The screen of the early mainframe computers, if any, was often a CRT in which the X and Y deflection plates were under the direct control of the computer. It was a kind of high-grade oscilloscope, with the computer using the electron beam to 'draw' an image on the screen. This was suitable for displaying charts and graphs, but not much use for text.

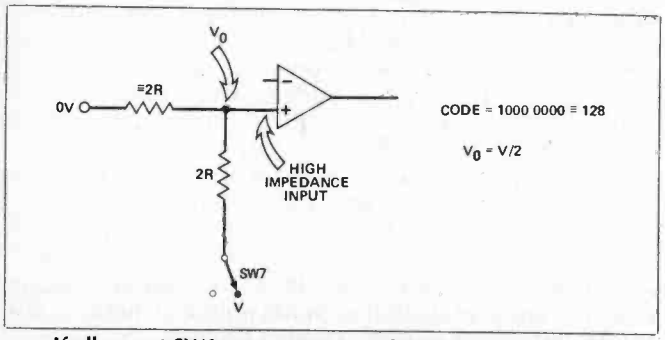
With a modern computer, the electron beam scans a rectangular area on the end of the CRT, in the same way as a TV set. The field (or **raster**) consists of a large number of horizontal lines (the number varies according to the system) placed close together (Fig. 6). Each line is scanned in turn, from the top to the bottom of the screen. Most TV

HOW THE R-2R LADDER WORKS

Assume all switches are set to 0 V. The three left-hand resistors are equal to 2R and 2R in parallel, and so can be replaced by a single R resistor. Thus the four resistors shown are equivalent to 2R switched to 0 V. We can carry this reasoning all along the ladder, until we reach a switch that is not set to 0 V.



If all except SW7 are set to 0 V, and SW7 is set to V (the reference voltage), we can consider all resistors to the left as equivalent to a single 2R resistor, switched to 0 V. We have a potential-divider, and $V_o = V/2$. This corresponds to the expected output, since 128/256 = 1/2.



If all except SW6 are set to 0 V and SW6 is set to V, we can consider all resistors to the left to be replaced by a single 2R resistor, switched to 0 V.

$$I_o = V/3R; \quad I_1 = (V - V_1)/2R; \quad I_2 = V_1/2R$$

By Kirchoff's Law (sum of currents entering a point must equal sum of currents leaving a point):

$$I_o = I_1 - I_2, \quad \text{so}$$

$V_1/3R = (V - V_1)/2R - V_1/2R$, giving

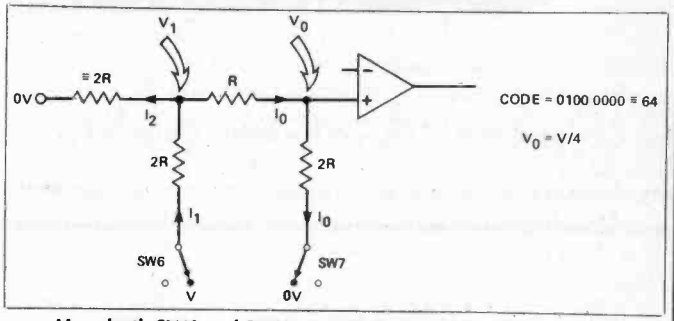
$$2V_1 = 3V - 3V_1 - 3V_1$$

$$V_1 = 3V/8$$

Now we can calculate V_o:

$$V_o = 2V_1/3 = V/4$$

This corresponds to what is expected since 64/256 = 1/4.



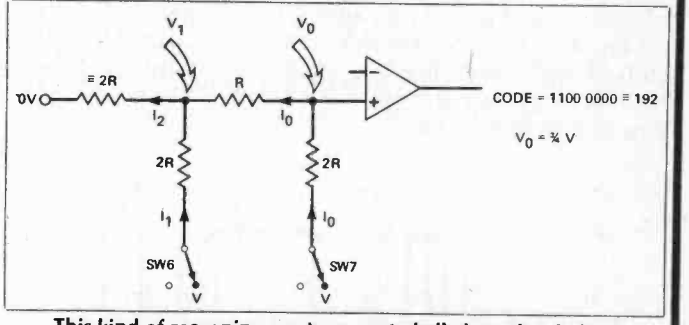
Here both SW6 and SW7 are switched to V.

$I_o = (V - V_1)/3R$ but I_1 and I_2 are as above.

$$I_2 = I_o + I_1, \quad \text{so}$$

$$V_1/2R = (V - V_1)/2R + (V - V_1)/3R,$$

which simplifies to $V_1 = 5V/8$.
Now $V_o = V_1 + (V - V_1)/3 = 5V/8 + V/8 = 3V/4$.
This is what we expect, since 192/256 = 3/4.



This kind of reasoning can be repeated all along the chain, getting more and more complicated, but with the same kind of result. If all the switches are set to V, the maximum V_o is obtained, ie 255/256V.

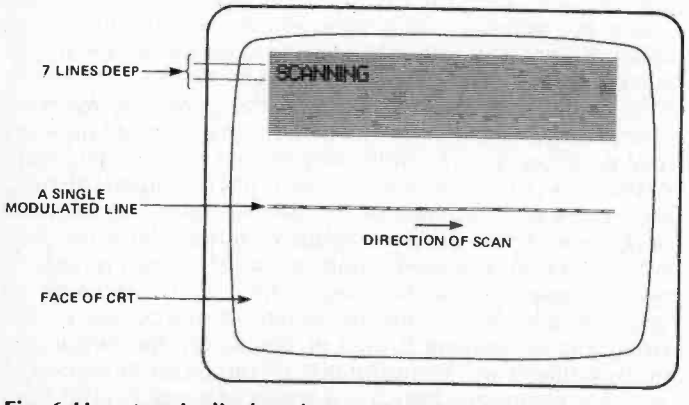


Fig. 6 How text is displayed on a TV screen.

systems have an interlaced raster in which the beam first scans alternate lines down the screen and then returns to scan the ones between the first set. As the beam scans, its intensity (and hence the brightness of the glow produced on the screen) is modulated. If the beam is strongly modulated, so that it produces either bright light or none, we can use it to produce a textual display of good contrast. Fig. 7 shows how a line of text can be built up from dots of light in seven successive scans. These are followed by a number of lines in which the beam is blanked (off) to provide spacing between rows of text. This process is repeated all the way down the screen.

As Fig. 7 shows, it is possible to build up well-defined alphabetic characters from a 7×5 matrix of dots. Numeric characters, punctuation marks and various other symbols can also built up in this way. We will now look in more detail at what the computer has to do in order to produce such a display. Visual displays are a field in which microcomputer designers have felt themselves free to use their inventiveness. Consequently, there are almost as many ways of producing the display as there are makes of microcomputer. Our discussion will therefore deal only with the main principles which are common to most micros.

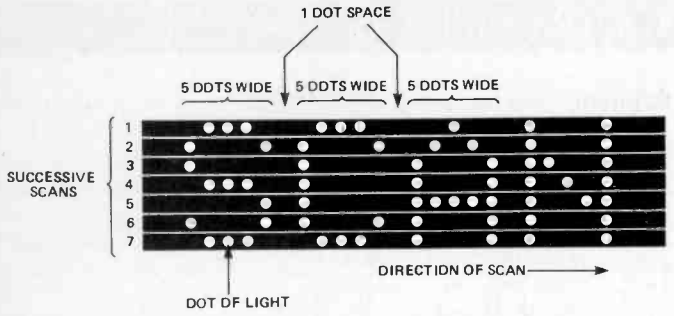


Fig. 7 Enlarged view of the first few letters shown in Fig. 6.

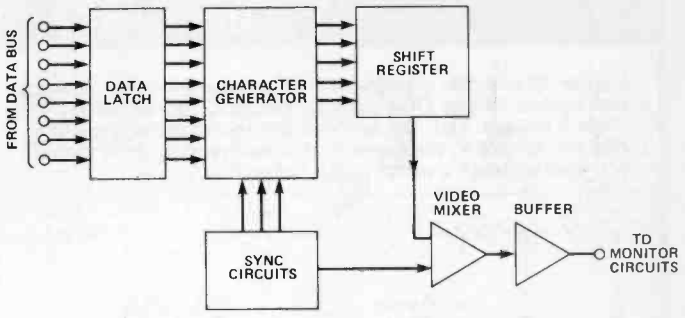


Fig. 9 Block diagram of the circuit for displaying text.

The Writing's On The Screen . . .

Figure 8 shows the signal which is fed to the grid of the CRT to modulate the beam to produce scan 5 of Fig. 7. The length of a single dot-producing pulse is of the order of 15 uS. This waveform is produced by a circuit like that shown as a block diagram in Fig. 9. Most micros have what is termed a **memory-mapped display**. A certain block of memory addresses is set aside for holding the text. Normally it requires one byte of data for each character. For example, if the screen has 16 lines, with 64 characters per line, the memory area must consist of 1024 bytes, or 1 kilobyte. Whenever text is to be displayed, the CPU stores the ASCII codes corresponding to each character in the appropriate memory cell.

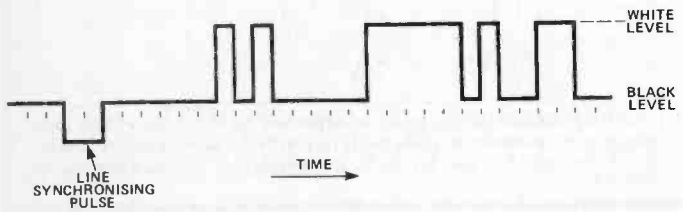


Fig. 8 The signal required to produce the fifth scan of Fig. 7.

The **video RAM**, as this section of memory is usually called, is read in sequence by the video circuitry. Although video RAM can be addressed by the MPU like any other part of RAM, in some micros it has its own control signals and its own data bus to connect it with the video circuit. For most of the time it operates independently of the microprocessor. The ASCII code for each character in turn is transferred to the data latch. It is held there while the next code is being fetched. The output from the latch goes to a **character generator** IC. This is a special kind of ROM (see Designing Micro Systems, ETI October 1982) which converts an ASCII code into the corresponding pattern of dots. It has inputs from the latch to tell it which character is to be generated, and from the synchronising circuit to tell it which one of the seven lines of dots is to be generated. It has five outputs that indicate which dots are to be displayed as white and which will be black.

The output from the character generator is converted from parallel to serial form by the shift register. This produces a train of pulses, some high, some low, like those shown in Fig. 8. These are fed to a video mixer circuit where line synchronising pulses and frame synchronising pulses are added. The combined video signal is then passed through a buffer circuit to the monitor.

Such a signal is not suitable for sending to a domestic TV set. The TV is expecting a UHF carrier signal from an aerial, modulated by the video signal. So, if a TV is being used, the signal from the video mixer must first go to a modulator. This produces a UHF carrier signal (usually on Channel 36) with the video signal imposed on to it. When the TV receives this signal it demodulates it, recovering the original video signal that it then uses to produce the display. The additional processes of modulation and demodulation inevitably lead to distortion; consequently, the resolution obtainable on a TV is inferior to that obtained on a proper monitor screen. A TV is acceptable when there are 40 or fewer characters per line, but when there are 60 characters or more the use of a monitor is much to be preferred.

Graphics Galore

The variety in methods of producing textual displays is exceeded by the variety of techniques used for producing graphics. A few micros (eg ZX81 and PET) use the character generator to produce geometrical shapes and other designs and symbols. These can be combined on the screen to produce designs of almost infinite complexity. This technique exploits one of the useful features of character generators: they can be programmed to produce any or all of the possible patterns on a 5×7 (or larger) matrix. For example, we can have them programmed for different styles of letter or for special letters for different languages. There 2^{35} permutations of dots, far more than can be accommodated within a single IC, so the snag of this method is that the user is limited to the range of symbols selected by the manufacturer. If you are writing programs for playing Bridge or Blackjack, the hearts and clubs symbols will be useful but, if your interests lie in cash account programs, they are a waste of space on the chip.

Many computers use graphics blocks (Fig. 10) as a means of constructing displays. A block may consist of six sub-blocks (or **pixels**, which is the name used for picture elements). Designing displays by this method involves interpreting your picture six blocks at a time and programming the computer with the corresponding code. The codes are stored in video RAM, as with text. Separate

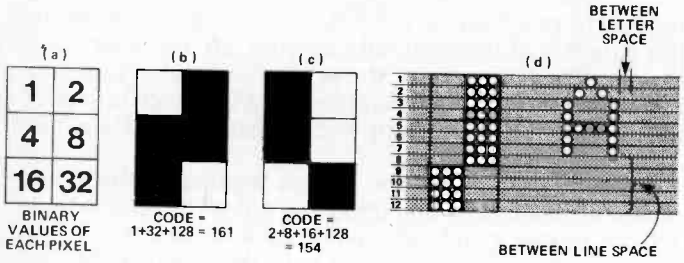


Fig. 10 Graphics blocks, as used in the TRS-80 Model I. (a) Each pixel has the binary value shown. (b) and (c) The sum of values of 'on' blocks plus 128 gives the code. (d) How a block is displayed by scanning (letter A for comparison of sizes). The blocks are displayed on a 6 × 12 matrix, leaving no space between adjacent blocks.

circuits are used in place of the character generator to convert the code to the corresponding set of video signals and feed them to the shift register. If the designs required are regular (such as decorative borders), programming is simple but it becomes very time-consuming if you want to draw complicated pictures.

A third approach to graphics is to deal with each pixel separately, and allocate one bit in video RAM to each pixel. If the value of the bit is '0', the corresponding pixel is 'off' (black screen). If it is '1', the pixel is 'on' (white screen). A medium-resolution graphics display, for example MODE 5 on the BBC Microcomputer, has 256 lines, each with 160 pixels. This gives a total of 40960 pixels. Allotting one bit per pixel, the video RAM must provide 5 kilobytes. If the display is to be in colour, an additional 5K bytes are required to indicate colour information for four colours, or an additional 15K for 16 colours. With the high-resolution display on the BBC Microcomputer (MODE 0) there are 640 pixels per line, requiring 20K, but allowing for only two colours. It can be seen that high resolution graphics, and particularly high-resolution *colour* graphics, require a very extensive video RAM. The cost of RAM has fallen in recent years, making it feasible to provide micros with good high-resolution colour graphics at relatively low cost. But, unless special 'paging' address circuitry is introduced, a micro with a 16-bit address bus is limited to 64K of memory, into which ROM, program RAM and the video RAM must be fitted. Consequently, an increase in the size of the video RAM means a decrease in the address space left for program RAM. If video RAM is physically a section of RAM itself, instead of being a separate entity as in some micros (see above), this section of RAM can be used for video when a program is to have plenty of graphics in colour, but can be turned over to program or data storage when graphics are not required. This is the system generally adopted in micros with high resolution graphics.

A Colourful PAL

In the PAL system of colour television, used in most European countries, colour transformation is transmitted by modulating the luminance (brightness signal) with a very high frequency chrominance (colour) signal. The way in which the chrominance signal is derived and subsequently decoded in the TV set is too complex to go into here. The final result is that three separate signals are derived to control the red, green and blue guns of the colour tube.

The output from a computer to an RGB colour monitor consists of four signals, on separate lines. The 'sync' signal provides the pulses needed for synchronising scanning with the reading of video RAM. The other three signals (R, G and B) control the three electron guns of the colour tube. Whenever there is a pulse on R, a red dot is produced on the screen. Whenever there is a pulse on G, we obtain a green dot. In either case only one kind of phosphor (red or green) is made to glow. If there are pulses on R and G at the same time, both electron guns are activated. A red dot and a green dot are produced in the same region of the screen. From the normal viewing distance it appears that there is a yellow dot on the screen. All colours are produced by mixing red, green and blue in various combinations and proportions.

The availability of separate signals for the red, green and blue guns means that excellent colour rendering with full saturation may be readily obtained on an RBG monitor. For those who wish to use a domestic colour TV, micros with colour graphics usually have a TV output. In the video mixer circuit the RGB signals are combined with the luminance signal before modulation and the composite signal is sent to the TV set. As with monochrome TVs, losses of signal quality occurring during demodulation and decoding mean that resolution and colour rendering is not as good as with a monitor.

High resolution colour graphics can give an intricate picture but, with so many pixels to be individually dealt with, one might think that programming would be too laborious for the average user. In fact, high resolution graphics may be easier to handle than the graphics blocks or generated characters described earlier. Since there is only one shape (a dot) instead of dozens or hundreds, we avoid the need to specify which shape is to be displayed. Since each pixel can be specified solely by its X and Y coordinates of the screen, the basis of pixel graphics is mathematical and it lends itself readily to mathematical treatment. It is easy to write routines for drawing lines, circles, or triangles, and for filling in areas with solid colour. The high-level language may include commands such as DRAW, PLOT, and CIRCLE, which perform these functions automatically, leaving the user to supply only the parameters. Graphs, bar charts, clock faces and all kinds of designs which are composed of reasonably simple geometrical shapes can be programmed in a few lines.

Getting Into Print

Controlling a printer is very different from controlling a monitor or TV. When controlling a monitor, the computer is responsible for all the timing and signal generation. The monitor merely transfers this signal to the screen as a raster of lines, varying in brightness along their length. Once the data has been transmitted, there are no further problems for the computer, for the monitor is able to work fast. The signals it receives are almost immediately translated into a pattern on the screen.

A printer takes a much larger share of the work on itself. The computer simply tells the printer which letter is to be printed next. Then the printer works out how and where to print the letter, or when to feed the paper on to print the next line. It can even organise itself to save time by printing alternate lines from right to left! In order to do this the printer needs an elaborate logic circuit. This may often include a microprocessor specially devoted to managing its activities. If the printer is of the dot-matrix type, it also needs a character generator to tell it which combinations of printing needles to fire at the ribbon (Fig. 11).

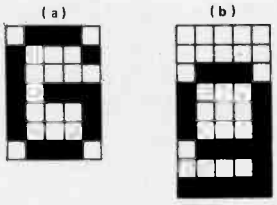


Fig. 11 Dot-matrix printers use a matrix of printing needles. If a needle is fired it hits the ribbon and makes a dot on the paper. In (a) a capital G is produced on a 5 x 7 matrix. In (b) a lower-case g is produced with a tail beneath it using a 5 x 9 matrix. In general, printers with few rows are not able to offer descenders like this.

The main disadvantage of a printer compared with a screen is that it deals with data much more slowly. There is a physical limit on how rapidly we can accelerate and then decelerate the appreciable mass of the print head (be it a matrix of needles or a daisywheel) and the rollers or sprockets which feed the paper to it. By contrast, the beam

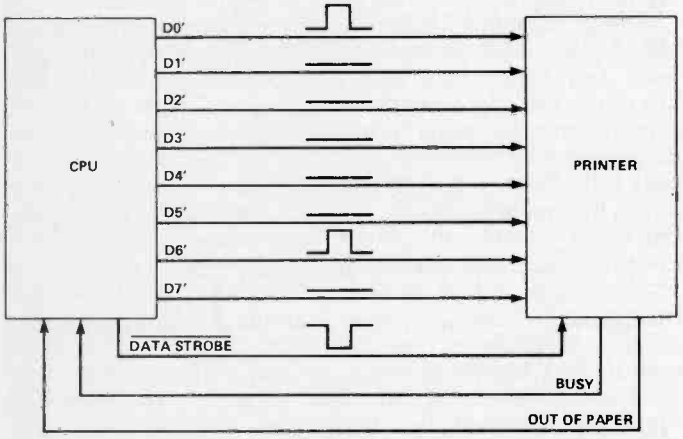


Fig. 12 Parallel data transfer between computer and printer.

of electrons in a CRT is virtually massless and can be directed and modulated almost instantly.

There are two main ways in which a computer and printer may be connected. The parallel transfer of data is illustrated in Fig. 12. An example of this system is the Centronics interface, originally devised by manufacturers of Centronics printers but now adopted by many other manufacturers. The first point to note is that there is two-way communication, in contrast with the one-way communication between computer and monitor. This is a consequence of the relatively slow speed of a printer. A computer can instruct a printer far faster than the printer can print. Rather than have the computer waste its valuable time waiting for the printer to operate letter by letter, we let the computer send a long string of commands to the printer in rapid succession. Since there are eight data lines, the computer can send a byte at a time. This is normally the ASCII code for the letter required. The printer can also interpret ASCII control characters for operations such as line feed (LF) and carriage return (CR). Whenever the computer is outputting data it makes the $\overline{\text{DATA STROBE}}$ line low. this has the same function as the $\overline{\text{WR}}$ control line used internally, and is derived almost directly from it. Similarly, the data lines are separated from the data bus of the computer only by latches, which hold the data long enough for the printer to be able to receive it. In some micros a general-purpose I/O device is used for this purpose (see Designing Micro Systems, ETI December 1982).

The I/O device or the buffers leading to the printer data lines need only one decoding circuit to enable them. Thus a printer needs to have only one address in RAM allocated to it. In comparison with the video screen, the printer makes minimal demands!

Printer Buffer

When data is received by the printer it is stored in a small RAM, called the **holder buffer**. This holds the codes for about 80 characters (maybe more), which is enough to print one line of text. It stores codes as they come in, then reads out codes previously stored and prints the characters they represent. When the computer sends a long string of codes the buffer is likely to become full. Also the printer has occasionally to stop printing to move on the paper to the next line. Again, codes will accumulate in the buffer. At this stage the printer puts a signal on the BUSY line. The effect of this is to interrupt the computer and make it stop sending any more data. When the printer has printed all that it has stored and its buffer is empty, the BUSY signal is taken off the line and the computer is free to send the next batch of data. On some interfaces there is also an acknowledge line ($\overline{\text{ACK}}$), a handshaking line by which the printer informs the computer that it has done whatever it was told to do and is awaiting fresh instructions. There may be an OUT OF PAPER line for

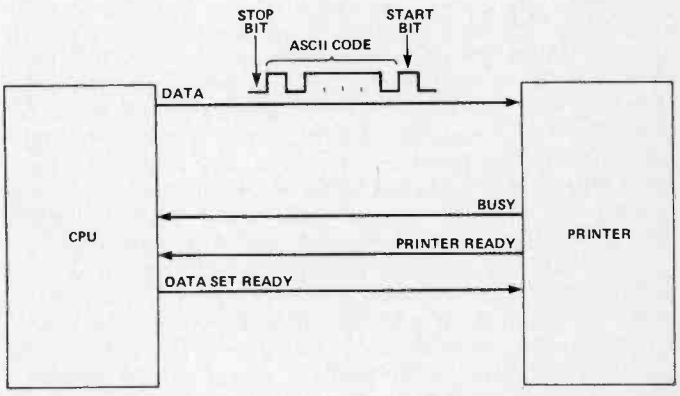


Fig. 13 Serial transfer of data between computer and printer.

signalling this fact to the computer. The level on this line is usually controlled by a micro-switch connected to a lever which is in contact with the paper. An OUT OF PAPER signal causes the computer to send no more data until the normal level is restored.

Are You Being Serialed?

The alternative way of sending data to a printer is to transmit a series of pulses along a single line. This has

obvious advantages in that only a single data line is required instead of eight. The most frequently used system of serial data transfer is known as the RS232C standard. The standard specifies voltage levels and rates of data transfer and the system to be used for coding the data. The standard also covers the types of connector to be used so that any pair of devices employing RS232C may be coupled together and expected to communicate reliably.

In Fig. 12 the pulses drawn above the parallel data lines indicate that the computer is sending 0100 0001 (or 65 decimal, the ASCII code for 'A'). In Fig. 13 the same ASCII code is being sent serially along one data line. Sending eight bits one after another is obviously slower than sending them in parallel, a byte at a time, as in Fig. 12, but since printers are relatively slow this is not a great disadvantage.

There are various ICs available for converting parallel data to serial data. A simple parallel-in-serial-out (PISO) shift register such as the 74LS166 will do the job, but ICs specially designed for computers do it better. A universal asynchronous receiver transmitter (UART) is an example of such an IC. This provides two-way communication, being able to receive parallel data from the CPU and transmit it as serial data, and to receive serial data from a peripheral and pass it to the CPU as parallel data. The latter function is not required for use with a printer, but would be used, for example, when two computers are required to communicate with each other. Not only does the UART convert from parallel to serial (or the other way about) but it takes the parallel data, makes it into a train of eight pulses, and adds a 'start' pulse and a 'stop' pulse to the beginning and end of the train.

Correcting The Errors

Since it requires only one data line, serial data transmission is suited for long distance. Parallel data transmission is rarely used under such circumstances. The longer the line, the greater the chance of stray electromagnetic interference finding its way on to the line and into the data receiving circuit. This is why the train of pulses often includes an extra pulse known as the **parity bit**. The idea of this is to allow the receiving device to check that no spurious pulse has been added as a result of interference during transmission. The parity bit is calculated by the UART before the data is transmitted, and is added to the end of the train of data pulses, then followed by the stop bit or bits. The value to be given to the parity bit is found by counting how many 1s are present in the data. If the number is even, the parity bit is made 1, so the total number of 1s becomes odd. If the number of 1s in the data is already odd, the parity bit is made 0, so retaining the odd number of 1s. At the receiving end the UART simply has to count the number of 1s in the train. If it is odd, all is well and it then sends on the train (minus the parity bit) to be decoded. If the number of 1s is even, a transmission error has occurred and the device or its operator can be alerted. This system is not absolutely error-proof for two errors could occur which would be self-cancelling. However, if the average rate of error is, say 1 in 100,000 bits on any occasion, the chance of two errors occurring on that occasion is 1 in 10,000,000,000 bits, which can be fairly safely disregarded.

The system described above is known as **odd parity**. It is also possible to work with **even parity**, in which the parity bit makes up the 1s to an even number. Most UARTs can be programmed to deal with either type of parity.

It's Your Timing That's Crucial

Figure 10 shows the train of pulses required to transmit the ASCII code for 'A serially. It includes an even parity bit. The voltage level specified for signalling 0 is +3 V or more, while the level for 1 is any voltage lower than −3 V. The interval between successive *groups* of pulses can be as long as necessary. The receiver waits until a start bit arrives and then decodes the nine or so bits which follow. There is no interval between successive pulses. The sequence of five 0s, for example, is received as one long high pulse. It follows that the transmitter and receiver must both have a method of timing the duration of pulses. Both circuits have oscillators or clocks built in to them to fix the rate at which they work. When two devices are coupled both clocks must operate at the same frequency. To assist standardisation, a number of frequencies have been selected for use with RS232C interfaces.

The rate of transmission of data is expressed in **baud**. This unit, named after a French engineer, J. M. E. Baudot, is equal to the number of bits transmitted per second. Standard rates are 110, 150, 300, 600, 1200, 2400, 4800, 9600 and 19200 baud, though the higher ones are not included in the RS232C standard. To simplify circuit design there are baud rate generator ICs. These are driven by a high frequency crystal oscillator circuit; the high

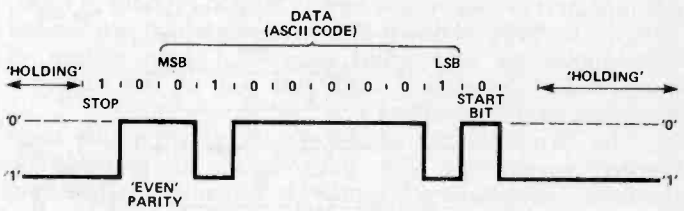


Fig. 14 The waveform of a serial signal (see text). There may be one or two 'stop' bits. The holding period between successive signals may be any length, which is why the system is called asynchronous.

frequency is divided by internal counter circuits to produce a range of output frequencies at standard baud rates. A UART may be connected to one or other of these outputs, depending on which baud rate is to be used. An interface usually has the facility for switching the UART to any one of the generator outputs, so that the rates on transmitter and receiver may be matched.

Since this is an asynchronous system, matching of timing does not have to be of high precision. Timing at the receiver begins when a start bit is received. The clock at the receiver has to remain in phase with the transmitting clock only for the duration of 10 to 11 pulses. The receiving clock probably runs slightly slower or faster than the transmitting clock, but this does not matter. It can get only a fraction of a pulse out of phase in such a short time, and this is not enough to cause errors in decoding. When the *next* train of pulses arrives, timing begins all over again from the arrival of the start pulse. Any discrepancies of timing which might have accumulated between trains are eliminated.

MICROS 7

To conclude the series, Owen Bishop takes a brief look at the two main ways in which information may be stored and retrieved by the microcomputer.

While the micro is in use, a lot of information may be held in ROM and in RAM. When the power is switched off, all the information in RAM is lost. This might consist of programs, tables of data, and information of various other kinds. If we want to retain this information for use on a future occasion, it must be stored in a form in which it can easily be put back into RAM when required. For certain applications RAM is not large enough to hold all the information we have to deal with. A business might be running a data-base program and require access to names and addresses of thousands of customers. These cannot be held simultaneously in RAM, so they must be loaded in and dealt with batch by batch. A complicated program may be too long to be held in RAM, but can be broken down into sections which are loaded individually for use when required. Some system of transferring information into and out of the micro is therefore almost essential.

The two methods of storage most commonly used involve transferring the information to a magnetic medium. Almost all micros provide a means of transferring information between RAM and a cassette tape. The other method makes use of a plastic disc coated on one or both sides with magnetic material. This is often referred to as a **floppy disc**, to contrast it with the **hard disc** which is often used with minicomputers but (at present) rarely with micros. Discs are made in two standard sizes, 8″ and 5¼″ in diameter. The smaller of these is the kind most often used with micros and is more correctly described as a **diskette.** Diskettes of even smaller diameter are now being produced.

Tape Measures

Information is stored on tape in the form of a square wave. Successive regions of the tape are magnetised in one direction or the other. The prime requirement is for a tape with low noise and freedom from blemishes. With an audio tape the occasional region with faulty coating makes little difference to the sound, but when the tape is being used for recording binary digits, such a 'drop-out' may convert a 0 to a 1 or the other way about, rendering the recording nonsensical from that point onward. Although it is possible to use ordinary tape for recording computer programs, most people prefer to use specially tested

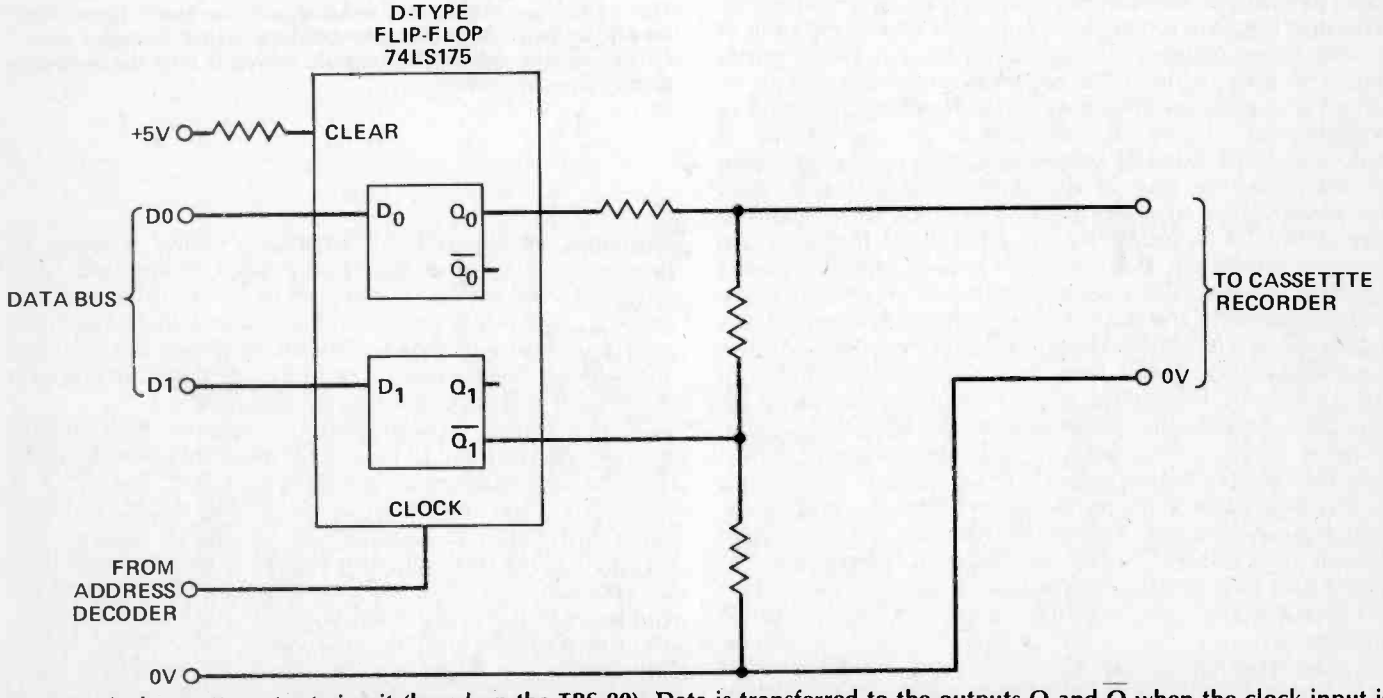


Fig. 1 Typical cassette output circuit (based on the TRS-80). Data is transferred to the outputs Q and $\overline{Q}$ when the clock input is made low. It is then latched until the next write operation. The resistors are chosen to give suitable output levels with different combinations of outputs from the flip-flops.

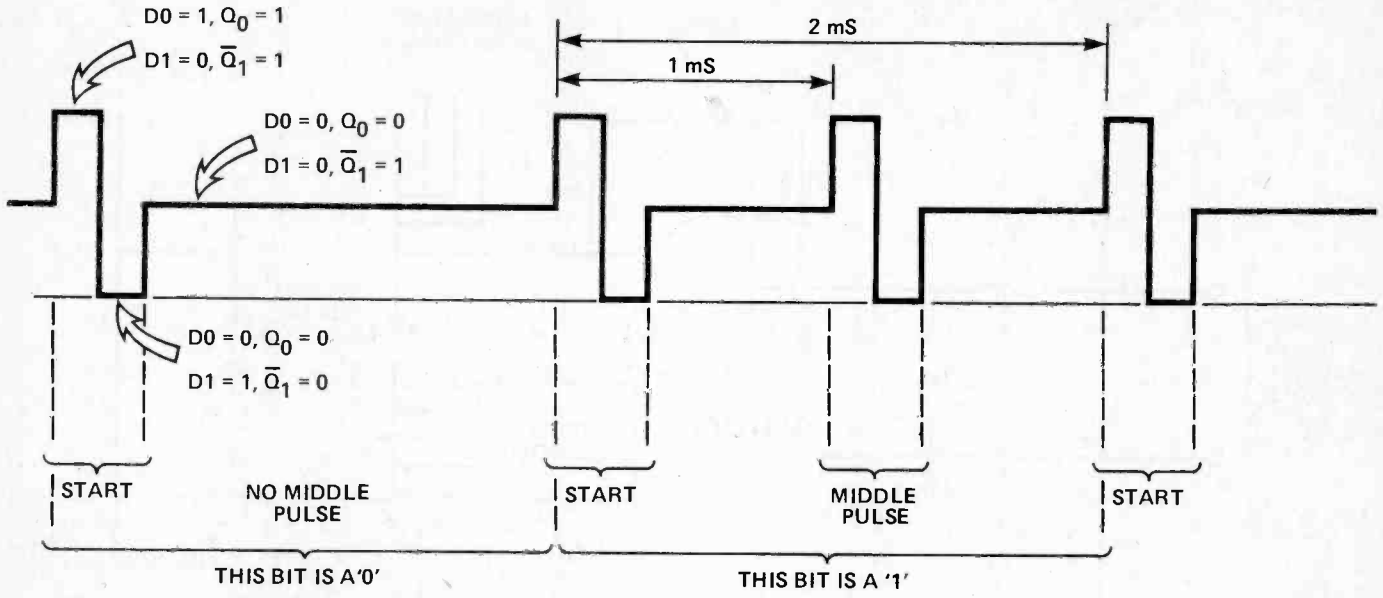


Fig. 2 The coding of the bits in the TRS-80.

'digital' tapes which are guaranteed free from such defects. They are usually supplied in shorter lengths than audio tapes. For example, C10 and C15 are two commonly available lengths. A program of 16K (or slightly longer) fits into a single side of such a tape when recorded at the standard rate of 300 baud. If your program is longer than this, you will probably prefer not to wait as long as 10 minutes or a quarter of an hour to load it and will be thinking of investing in a disc drive.

Figure 1 shows the circuit of a typical cassette output circuit. It occupies a single address in the memory stage of the computer. Data is fed to it as a series of 0s and 1s and a corresponding voltage is fed to the recorder. In the example shown, a positive voltage represents 1, 0V represents 0. In the absence of a signal the output voltage is one half of the '1' level. However, there is little standardisation of microcomputer outputs to cassette recorders, and there are many variations on this theme.

All micros record and load the data serially, that is to say, one bit at a time. Before a byte of information is recorded it is broken down into its eight bits and each bit is sent separately to the cassette output circuit. Methods of coding the information vary widely from one micro to another. This is why it is almost impossible to load one micro with a program saved on a machine of different make.

Bits And Blips

Not only does each micro have its own system of formatting the tape — the way it begins and ends each transmission of data — but the 1s and 0s may be represented in several different ways. The TRS-80, for example sends data at the rate of one bit every 2 mS (500 baud). It indicates the beginning of each bit by a 'start pulse' or 'blip' as in Fig. 2. This is a short swing to high, then to low and finally a return to the 'no-signal' level. If the bit is a 0, no further signal is sent. After 2 mS the next 'blip' indicates the start of the next bit. If the bit is a 1, a blip is sent exactly 1 mS after the start pulse.

When the tape is played back the signal is taken in through a circuit as in Fig. 3, where the 'blips' are detected. Though they do not have exactly the same form as the original signal, the timing is the same and this is all that matters. The micro is programmed to wait until a signal is detected and to sample the input exactly 1 mS later. If a signal is detected at this stage too, the bit is taken to be a 1. If no pulse is detected, the bit is taken as a 0. It then awaits the arrival of the next signal to indicate the beginning of the next bit. At each stage it stores which kind of bit (0 or 1) it has received. When it has received eight bits these are assembled into a byte and stored in RAM. If a flaw in the tape causes a bit to be missed, or an extra bit to be recorded, this upsets the decoding of all bytes for the remainder of the recording. An incorrectly-read bit may alter only the byte it is part of. This too can affect the interpretation of the whole of the remainder of the recording, especially if the recording is a machine-code program.

Another system of recording data depends on **frequency shift keying** (FSK, for short). This method is also used in transferring data from one micro to another by wire. Two standard frequencies are used, one of them having perhaps twice the frequency of the other. When we say 'standard' we mean standard for that model of micro, but for tape recording different makes of micro almost invariably work to different standards. A 0 is represented by a short tone burst of one frequency and a 1 by a burst of the other frequency. On playback (or on receiving the transmission over a line) the computer can easily measure pulse length and so find out which frequency is being sent at each instant. This information is then converted into 0s and 1s and assembled into bytes to be stored in RAM.

Tape Versus Disc

Tape recording computer programs and data files has some considerable advantages which must be matched against considerable disadvantages. The main advantage is cheapness. Within the computer we need relatively simple output and input circuits. The tape recorder itself can be a simple and cheap mass-produced model. Many intending microcomputer owners already possess a tape recorder, so the only expense is the lead to connect it to the micro. A sophisticated hi-fi recorder often gives trouble owing to its noise reduction circuits which turn up the volume during periods of no signal, so feeding the computer with amplified tape noise and confounding its signal detection program. It has often been said the the cheaper recorder, the better it is for use with a micro. However, certain micros give problems

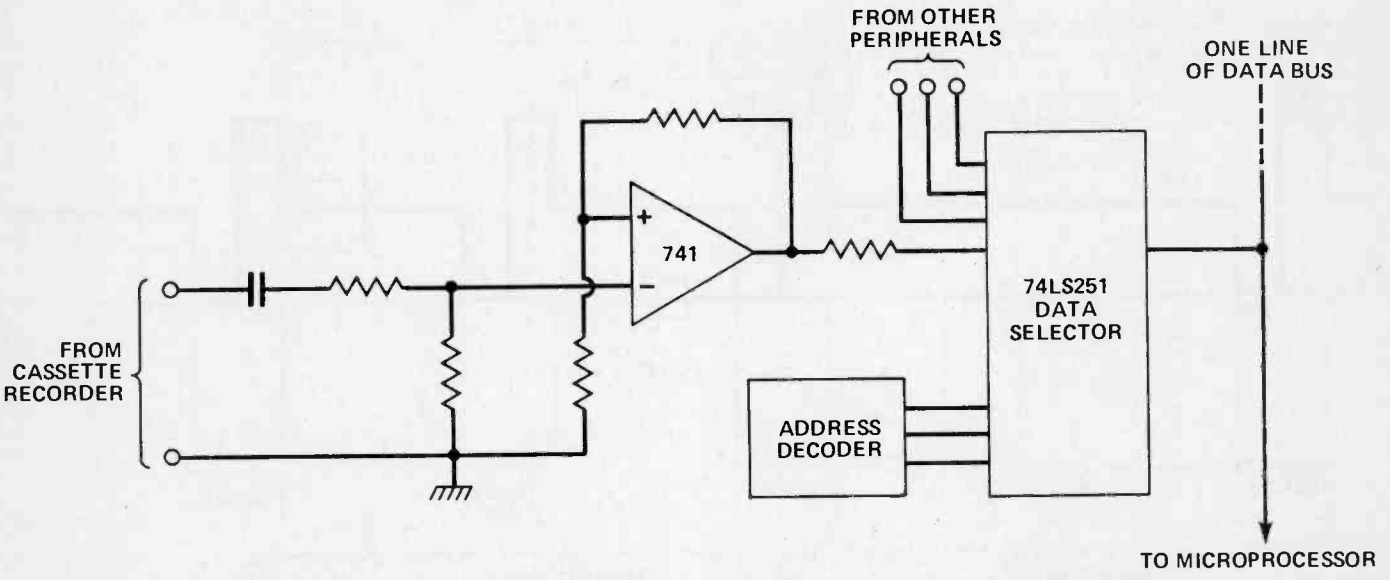


Fig. 3 A typical cassette input circuit has an op-amp wired with positive feedback so that it saturates and its output swings fully in either direction. The data selector is used to send the input from the recorder or from other peripherals such as games controllers.

when loading from tape, and make it necessary to set the playback volume fairly carefully to avoid either too small a signal or a large signal which saturates the input circuits.

Manual Controls

Provided that the user requires only to save and load relatively short programs, the cheapness and availability of cassette recorders outweighs their disadvantages. If a large amount of data is to be handled and if time is costly, the balance of advantage swings firmly toward the disc. One of the disadvantages of the cassette recorder is that its 'record', 'play' and tape winding controls cannot be operated automatically by the computer. They must be operated by the user, with the inevitable consequence of making a mistake. At the least, this wastes time and, at the worst, may cause a valuable program or set of data to be erased.

The only control which micros have over the recorder is to start and stop the motor at the beginning and end of each recording. There is a small relay in the computer which is connected in place of this 'on/off' switch often found on the stem of a microphone. This relay is controlled by the MPU, having its own address or addresses in memory. Usually there is a flip-flop which is toggled by writing to one address or the other. Figure 4 is a typical motor-switching circuit. Such a circuit is very simple and therefore found in all except the very cheapest micros.

A cassette tape passes over the recording/playback head at the standard speed of 1 15/16 inches per second (50 mm/S). At this relatively low speed the rate of recording is limited to a few hundreds of bits per second. Consequently it takes several minutes to record a program which is more than a few kilobytes long. This results in excessively long delays when running data-bases and similar programs.

Other problems with tape are connected with the fact that programs or data files are recorded one after the other along the length of the tape. Recordings can be played back only in the order on which they were recorded. If you want to return to a recording which is earlier on the track, it is necessary to operate the recorder manually, rewinding it to a position in advance of its new starting point. This takes time and is a tedious operation even with the aid of the footage meter. If the item of data you need is part of a long recording, you have to play the recording through from beginning to end to retrieve the single item you need.

Discs have none of these disadvantages and are altogether more reliable than cassette tapes. On the other hand, a disc drive is considerably more expensive than an ordinary cassette recorder. But if we abandon the idea of cheap data storage, we can take advantage of the best available technology and design a device which is ideal for its purpose.

Spin A Disc

The recording medium, disc or diskette, consists of a disc of mylar film coated with magnetic oxide; it may be coated on one side only or on both sides. The magnetic head is very close to the disc when reading and writing data and the disc rotates at high speed, so it is essential to exclude particles of dust. Even the dust from cigarette smoke can cause malfunction (another good reason for giving up — Ed). The disc is therefore sealed in a plastic sleeve, lined with a textured material which lubricates the surface of the disc and removes debris. The case has a slot (Fig. 5) to allow the magnetic head access to the disc. It also has a small hole through which the **sector holes** are visible. These are holes punched in the disc and spaced regularly around it. There is a fixed number, depending on the system on which the disc is being used. Commonly there are 16 such holes, giving a 16-sector disc. The effect of these is that the tracks on the disc, which are concentric, are divided into 16 sectors. The disc drive has a light source located on one side of the disc to shine through the sector holes as the disc spins around. A phototransistor on the other side of the disc detects when a hole passes. This aids the drive in sensing the position of the disc.

Discs which have holes to mark the sectors are known as **hard-sectored** discs. An alternative system has a single hole for detecting each rotation of the disc but relies on software for dividing the track into sectors. Such a disc is known as a **soft-sectored** disc.

Another phototransistor in the drive is used to detect whether the disc is 'write-protected'. There is a notch in

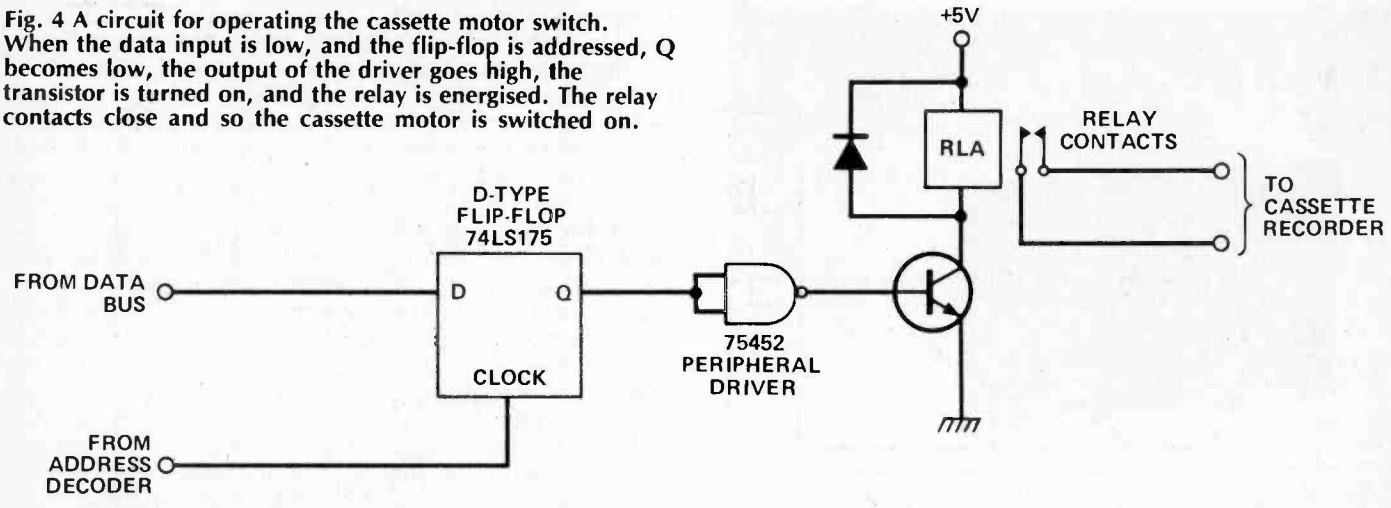


Fig. 4 A circuit for operating the cassette motor switch. When the data input is low, and the flip-flop is addressed, Q becomes low, the output of the driver goes high, the transistor is turned on, and the relay is energised. The relay contacts close and so the cassette motor is switched on.

the edge of the case of the disc: light shines through this notch from below and falls on the phototransistor. The user may fix a sticky tag over this notch to prevent light from passing. In this event, the phototransistor is not activated and the writing action of the drive is inhibited. This serves to prevent the accidental or intentional overwriting or altering of data or programs. This is simply a safety measure: the tag is peeled off should further writing be required.

The Faster Format

In a typical disc drive the disc is rotated at a constant speed of several hundred revolutions per minute. For example, the Siemens FDD100-5 drive rotates at 300 RPM. At the middle track, this gives the magnetic material a speed of about 1400 mm/S relative to the head, compared with 50 mm/S in the cassette recorder. As a result of this and the physically small size of the read/write head, data can be recorded and read at 125 kilobits per second. Reading and writing of data is therefore extremely fast. There is a delay of one second while the disc comes up to full speed: the head takes an average of 300 mS to find the required track and a further 15 mS to settle into position. After that, data is transferred at the rate mentioned above. Should the head need to change from one track to another, as it will if much data has to be transferred, it takes only 25 mS to move from one track to another.

It is evident from the description above that access to data is very much quicker and more direct than is the case with tape. Instead of having to run from one end of a tape

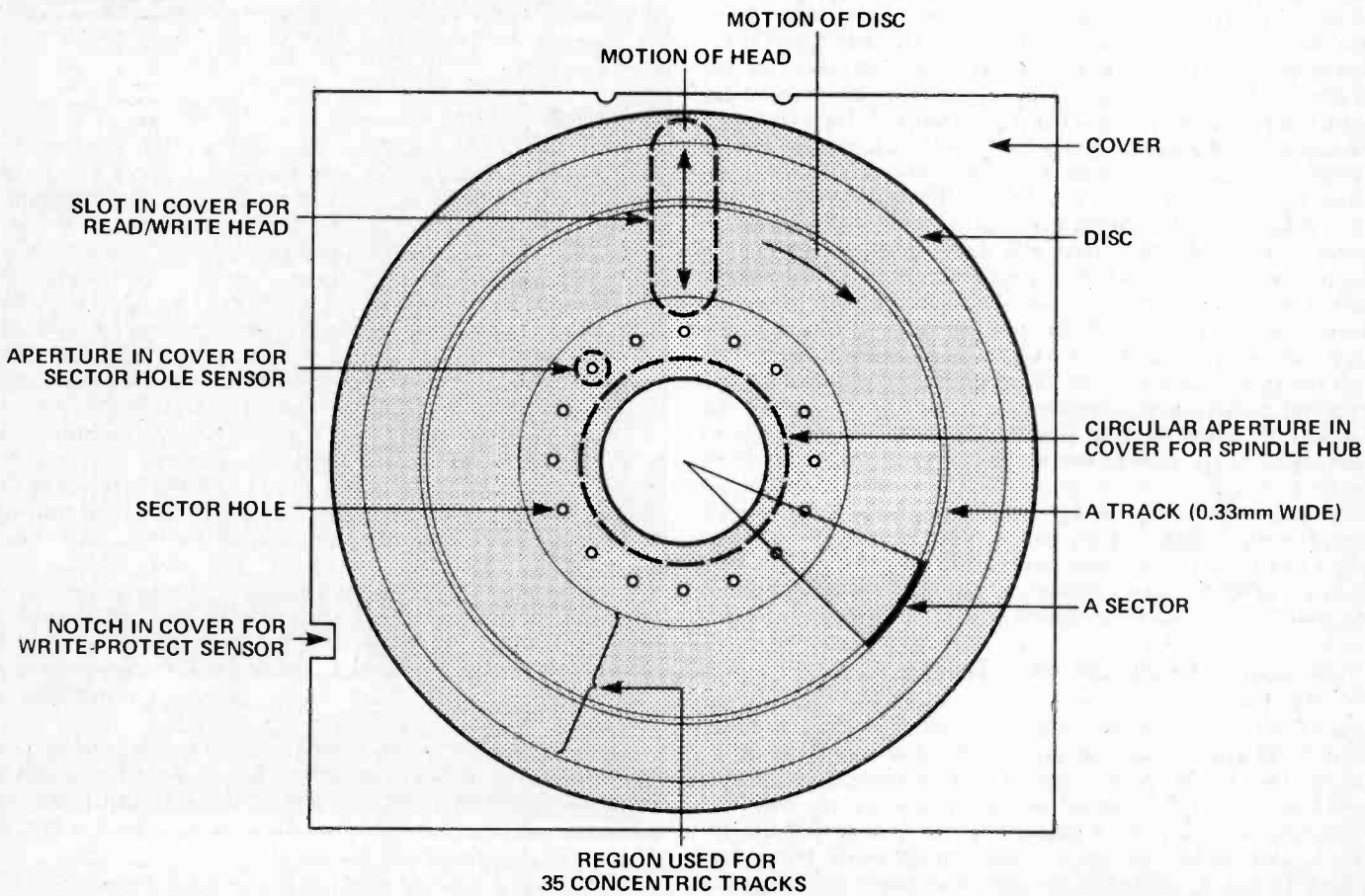
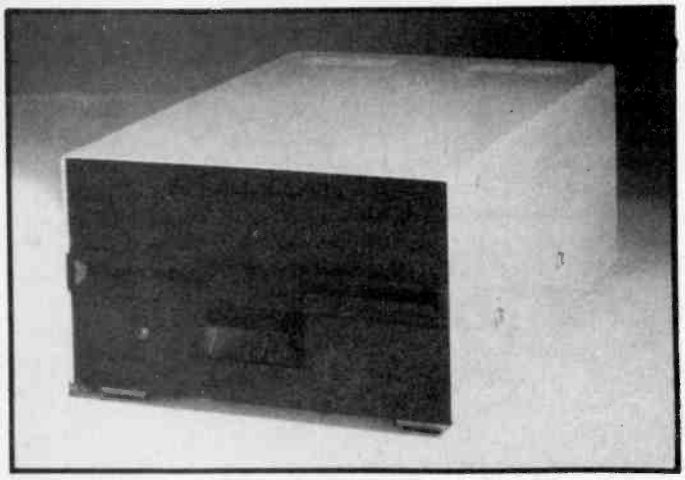


Fig. 5 'See-through' diagram of a hard sectored floppy diskette in its cover. Normally, of course, you cannot see the disc itself.



A typical single disc drive unit. The disc (still in its cover) is inserted in the slot which runs almost the full width of the case. The LED is lit to indicate when the drive is in operation.

to the other, the head can go directly to the track, then to the sector within the track, to find what is required. Changing from one track to another is effected by a stepper motor connected to a worm gear which moves the head radially. The signals from the sector hole sensor tell the drive when the required sector is in position to be read from or written to.

Naturally, the operation of the drive cannot be manual. The drive contains a complicated array of electronics (see photo) to control the disc, the stepper motor, the raising and lowering of the head, as well as those circuits responsible for handling the signals which are to be put on to the disc or have been taken from it. Synchronising these operations requires an impressive amount of logic circuitry: some of the more advanced disc drives even incorporate a microprocessor to take charge of the operation. This has several advantages, especially with a soft-sectored disc. The rate at which the medium passes the head depends on the radial distance from the centre: consequently data is more compressed on the inner tracks and widely spaced on the outer tracks. For reliability, it is the speed on the innermost track which limits the maximum rate of data transfer. Using a microprocessor, it is possible to perform rapid calculations which allow the drive to vary the rate of data transfer according to the radial position of the head. Data is stored at *about* the same density on all tracks but the outer tracks can have more sectors, so the overall storage capacity of the disc is markedly increased.

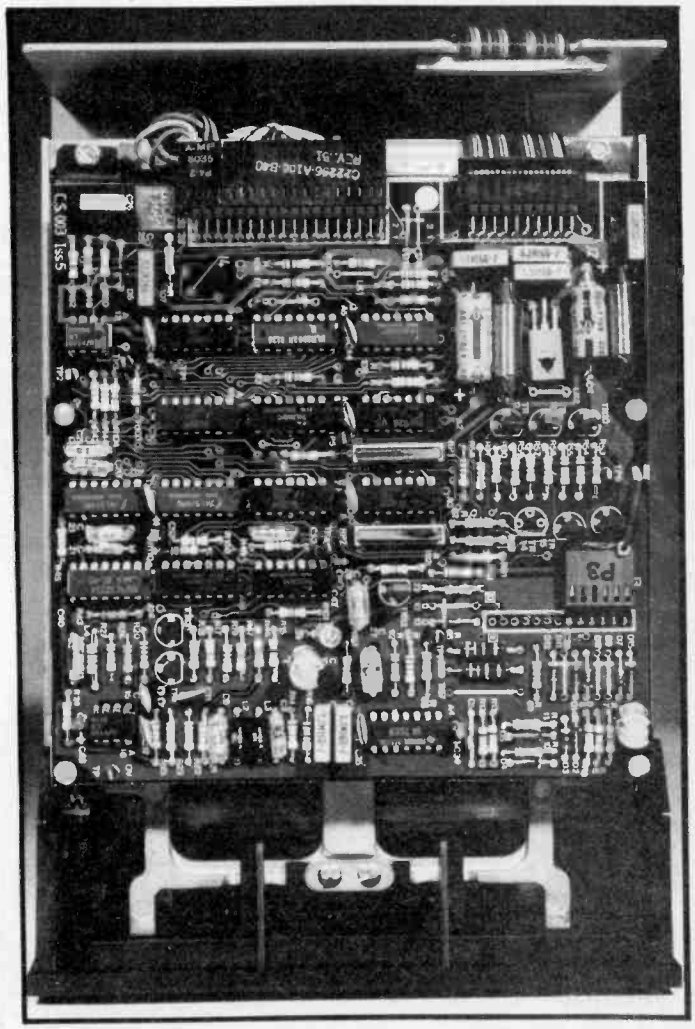
On the whole, the standard method of storage is adequate. A $5\frac{1}{4}$" mini-diskette may have 31 tracks each of 16 sectors, and each sector stores 256 bytes. This gives a total storage of 124 kilobytes on a single-sided disc. Double-sided discs store twice this amount, and the capacity can be further increased by using 'double density' discs in which there are almost twice as many tracks, placed closer together.

Keeping Track Of The Tracks

The disc referred to above which has 31 tracks available for storage of data or programs also has four additional tracks reserved for use by the disc operating system. In order to make efficient use of the disc space, in which items of data may be continually being written, replaced, and deleted, it is essential for a large amount of 'book-keeping' to be done. The system must know on which sector of which track each item has been placed. Items longer than 256 bytes occupy more than one sector, so the system must know how to direct the head from sector to sector to pick up all the data in the correct order.

The reserved tracks contain an index or directory of the contents of the disc so that the whereabouts of every item of data is known. The directory also helps the head to find vacant sectors when a new item of data is to be placed on the disc. The reserved tracks also hold a special program, the disc operating program, which is loaded into RAM when the micro is first powered up. This provides the instructions for accessing the directory tracks and obtaining whatever information is required, and for placing new information on the disc. This program (provided it is well written, which some are not) together with the hardware of the disc drive itself, completely automates the transfer of information between micro and magnetic storage medium. The operator is almost unaware of what is happening except for the comforting clunks and whirrs emanating from the drive. With a well-made drive and by observing a few simple precautions in gentle handling of the discs themselves, the reliability is far higher than with tapes, making this a relatively expensive but infinitely preferable method of data storage.

In concluding this series, we would like to thank Cumana Ltd., 35 Walnut Tree Close, Guildford, Surrey GU1 4UN, for helpful information and for permission to reproduce the photograph of their circuit-board for the Siemens drive, as adapted for use with the Apple II computer.



Some of the hardware you get when you invest in a disc drive. You need another card full of hardware to interface the drive to the computer.

electronics today international
BOOK SERVICE

How to order: indicate the books required by ticking the boxes and send this page, together with your payment, to: ETI Book Service, Argus Specialist Publications Ltd, 145 Charing Cross Road, London WC2 0EE. Make cheques payable to ETI Book Service. Payment in sterling only please. All prices include P & P. Prices may be subject to change without notice.

BEGINNERS GUIDE

☐ **Beginner's Guide to Basic Programming** Stephenson	£5.35
☐ **Beginner's Guide to Digital Electronics**	£5.35
☐ **Beginner's Guide to Electronics**	£5.35
☐ **Beginner's Guide to Integrated Circuits**	£5.35
☐ **Beginner's Guide to Computers**	£5.35
☐ **Beginner's Guide to Microprocessors**	£5.35

COOKBOOKS

☐ **Master IC Cookbook** Hallmark	£10.15
☐ **Microprocessor Cookbook** M. Hordeski	£7.70
☐ **IC Op Amp Cookbook** Jung	£14.25
☐ **PLL Synthesiser Cookbook** H. Kinley	£7.70
☐ **Active Filter Cookbook** Lancaster	£13.40
☐ **TV Typewriter Cookbook** Lancaster	£11.15
☐ **CMOS Cookbook** Lancaster	£11.85
☐ **TTL Cookbook** Lancaster	£10.95
☐ **Micro Cookbook Vol. 1** Lancaster	£15.30
☐ **BASIC Cookbook** K. Tracton	£6.00
☐ **MC6809 Cookbook** C. Warren	£7.25

ELECTRONICS

☐ **Principles of Transistor Circuits** Amos	£8.50
☐ **Design of Active Filters with experiments** Berlin	£11.30
☐ **49 Easy to Build Electronic Projects** Brown	£6.00
☐ **Electronic Devices & Circuit Theory** Boylestad	£13.20
☐ **How to build Electronic Kits** Capel	£3.55
☐ **How to Design and build electronic instrumentation** Carr	£9.35
☐ **Introduction to Microcomputers** Daglecs	£7.20
☐ **Electronic Components and Systems** Dennis	£15.00
☐ **Principles of Electronic Instrumentation** De Sa	£11.40
☐ **Giant Handbook of Computer Software**	£12.95
☐ **Giant Handbook of Electronic Circuits**	£17.35
☐ **Giant Handbook of Electronic Projects**	£11.75
☐ **Electronic Logic Circuits** Gibson	£5.55
☐ **Analysis and Design of Analogue Integrated Circuits** Gray	£30.25
☐ **Basic Electronics** Grob	£11.30
☐ **Lasers — The Light Fantastic** Hallmark	£7.70
☐ **Introduction to Digital Electronics & Logic** Joynson	£5.25
☐ **Electronic Testing and Fault Diagnosis** Loveday	£7.85
☐ **Electronic Fault Diagnosis** Loveday	£6.25
☐ **Essential Electronics A-Z Guide** Loveday	£7.50
☐ **Microelectronics Digital & Analogue circuits and systems** Millman	£12.70
☐ **103 Projects for Electronics Experimenters** Minis	£8.30
☐ **VLSI System Design** Muroga	£34.10
☐ **Power FETs and their application** Oxner	£9.40
☐ **Practical Solid State Circuit Design** Olesky	£25.00
☐ **Master Handbook of IC Circuits** Powers	£12.85
☐ **Electronic Drafting and Design** Raskhodoff	£22.15
☐ **VOM — VTVM Handbook** Risse	£8.50
☐ **Video and Digital Electronic Displays** Sherr	£28.85
☐ **Understanding Electronic Components** Sinclair	£7.50
☐ **Electronic Fault Diagnosis** Sinclair	£4.50
☐ **Physics of Semiconductor Devices** Sze	£17.35
☐ **Digital Circuits and Microprocessors** Taub	£32.00
☐ **Active Filter Handbook**	£7.60
☐ **Designing with TTL Integrated Circuits** Texas	£15.20
☐ **Transistor Circuit Design** Texas	£15.20
☐ **Digital Systems: Principles and Applications** Tocci	£12.95
☐ **Master Handbook of Telephones** Traister	£10.00
☐ **How to build Metal/Treasure Locators** Traister	£6.00
☐ **99 Fun to Make Electronic Projects** Tymony	£8.50
☐ **33 Electronic Music Projects you can build** Winston	£6.95

COMPUTERS & MICROCOMPUTERS

☐ **BASIC Computer Games** Ahl	£6.35
☐ **From BASIC to PASCAL** Anderson	£9.95
☐ **Mastering Machine Code on your ZX81** T. Baker	£7.25
☐ **UNIX — The Book** Banaham	£8.75
☐ **Z80 Microcomputer Handbook** Barden	£10.95
☐ **Microcomputer Maths** Barden	£11.90
☐ **Digital Computer Fundamentals** Barter	£9.90
☐ **Visicalc Book. APPLE Edition** Bell	£15.55
☐ **Visicalc Book. ATARI Edition** Bell	£15.55
☐ **Introduction to Microprocessors** Brunner	£23.00
☐ **Programming your APPLE II Computer** Bryan	£9.25
☐ **Microprocessor Interfacing** Carr	£7.70
☐ **Microcomputer Interfacing Handbook A/D & D/A** Carr	£9.50
☐ **Musical Applications of Microprocessors** Chamberlain	£28.85
☐ **30 Computer Programs for the Home Owner in BASIC** D. Chance	£9.25
☐ **Microcomputers** Dirkson	£9.30
☐ **APPLE Personal Computer for Beginners** Dunn	£9.50
☐ **Microcomputers/Microcomputers — An Intro** Gioone	£11.80
☐ **Troubleshooting Microprocessors and Digital Logic** Goodman	£9.25
☐ **Getting Acquainted with your VIC 20** Hartnell	£8.50
☐ **Getting Acquainted with your ZX81** Hartnell	£5.95
☐ **Let your BBC Micro Teach you to program** Hartnell	£7.90
☐ **Programming your ZX Spectrum** Hartnell	£8.50
☐ **The ZX Spectrum Explored** Hartnell	£6.95
☐ **How to Design, Build and Program your own working Computer System** Haviland	£9.30
☐ **BASIC Principles and Practice of Microprocessors** Heffer	£7.15
☐ **Hints and Tips for the ZX81** Hewson	£5.25
☐ **What to do when you get your hand on a Microcomputer** Holtzman	£9.95
☐ **34 More Tested Ready to Run Game Programs in BASIC** Horn	£7.70
☐ **Microcomputer Builders' Bible** Johnson	£12.40
☐ **Digital Circuits and Microcomputers** Johnson	£14.55
☐ **PASCAL for Students** Kemp	£7.20
☐ **The C — Programming Language** Kernighan	£18.20
☐ **COBOL** Jackson	£9.25
☐ **The ZX81 Companion** Maunder	£9.50
☐ **Guide to Good Programming Practice** Meek	£6.40
☐ **Principles of Interactive Computer Graphics** Newman	£13.95
☐ **Theory and Practice of Microprocessors** Nicholas	£11.35
☐ **Exploring the World of the Personal Computer** Nilles	£12.95
☐ **Microprocessor Circuits Vol. 1. Fundamentals and Microcontrollers** Noll	£9.80
☐ **Beginner's Guide to Microprocessors** Parr	£5.35
☐ **Microcomputer Based Design** Peatman	£11.30
☐ **Digital Hardware Design** Peatman	£9.80
☐ **BBC Micro Reavealed** Ruston	£9.45
☐ **Handbook of Advanced Robotics** Safford	£14.45
☐ **1001 Things to do with your own personal computer** Sawusch	£8.50
☐ **Easy Programming for the ZX Spectrum** Stewart	£7.15
☐ **Microprocessor Applications Handbook** Stout	£34.40
☐ **Handbook of Microprocessor Design and Applications** Stout	£37.60
☐ **Programming the PET/CBM** West	£17.80
☐ **An Introduction to Microcomputer Technology** Williamson	£8.20
☐ **Computer Peripherals that you can build** Wolfe	£12.40
☐ **Microprocessors and Microcomputers for Engineering Students and Technicians** Wooland	£7.10

REFERENCE BOOKS

☐ **Electronic Engineers' Handbook** Fink	£56.45
☐ **Electronic Designers' Handbook** Giacoletto	£59.55
☐ **Illustrated Dictionary of Microcomputer Technology** Hordeski	£8.45
☐ **Handbook for Electronic Engineering Technicians** Kauffman	£27.50
☐ **Handbook of Electronic Calculators** Kauffman	£35.00
☐ **Modern Electronic Circuit Reference Manual** Marcus	£44.00
☐ **International Transistor Selector** Towers	£10.70
☐ **International Microprocessor Selector** Towers	£16.00
☐ **International Digital IC Selector** Towers	£10.95
☐ **International Op Amp Linear IC Selector** Towers	£8.50
☐ **Illustrated Dictionary of Electronics** Turner	£12.95

VIDEO

☐ **Servicing Home Video Cassette Recorders** Hobbs	£12.95
☐ **Complete Handbook of Videocassette Recorders** Kybett	£9.25
☐ **Theory and Servicing of Videocassette Recorders** McGinty	£12.95
☐ **Beginner's Guide to Video** Matthewson	£5.35
☐ **Video Recording: Theory and Practice** Robinson	£14.40
☐ **Video Handbook** Van Wezel	£21.90
☐ **Video Techniques** White	£12.95

Please send me the books indicated. I enclose cheque/postal order for £.....Prices include postage and packing
I wish to pay by Access/Barclaycard. Please debit my account.

5	2	2	4													

4	9	2	9													

Signed.....

Name .....

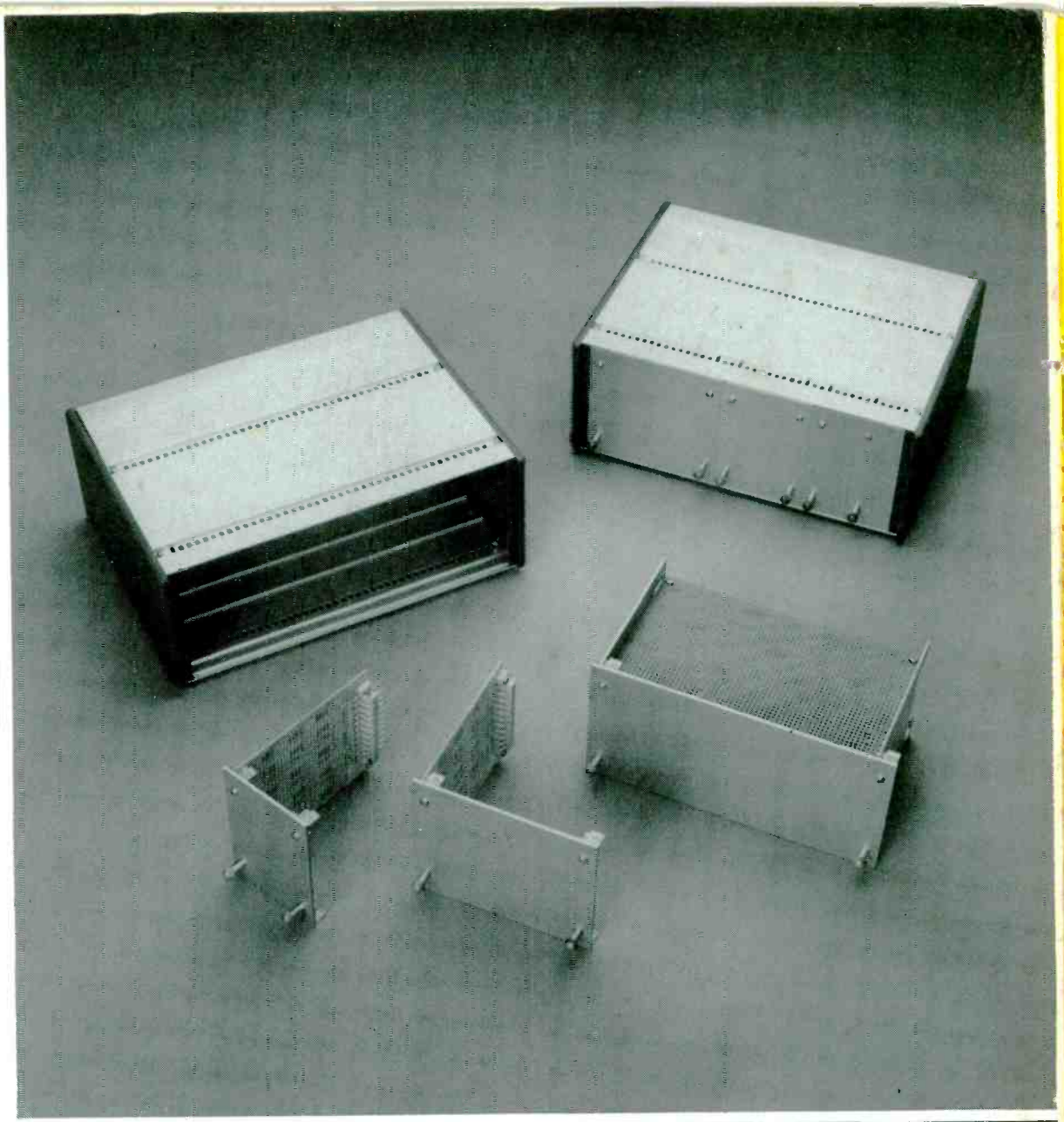
Address .....

.....

.....

KMT

THE NEW SMALL MODULE RACK



The unique new modular enclosure.
Suitable for alarm systems, counters, interfaces, amplifiers, model control units and many other projects.
Easy to assemble - just 10 screws.
Easy on the pocket - house your projects economically and professionally.
For full size eurocards (100 x 160 mm) mounted horizontally in the 35TE front panel kit.
Half size eurocards (100 x 80 mm) mounted vertically included as part of all front panel kits (except 35TE), with connector included.

·Low Cost·

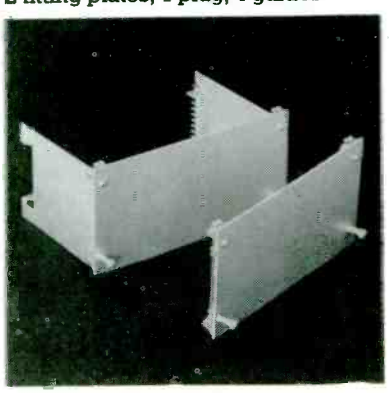
42 TE front panel set includes:-
Front panel, 2 card holders, 2 handles
35 TE front panel set includes:-
Front panel, 2 card holders, 2 handles, 2 fitting plates, 1 plug, 4 guides

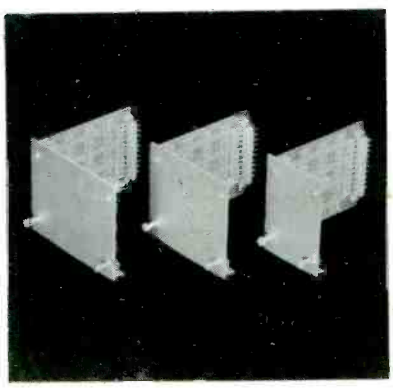
·Easy Assembly·

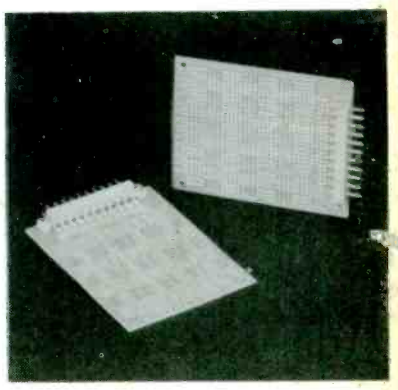
The 21, 18, and 12 TE front panel sets each include:-
Front panel, 2 card holders, 2 handles, 1 card, 1 plug, 2 guides

·Versatile Design·

The plug-in card set includes:-
Card (phenol resin), 1 plug, 2 guides
Guides in sets of 10







BICC-VERO ELECTRONICS LIMITED

Retail Department,
Industrial Estate,
Chandlers Ford,
Hants SO5 3ZR
Tel (04215) 62829

BICC vero

Send 50p for a copy of the catalogue with details of the full range available.

Ordering information

Description	Order code
KMT small module rack	127–31427F
42TE front panel set	127–32381K
35TE front panel set	127–32382G
21TE front panel set	127–32380B
18TE front panel set	127–32379A
12TE front panel set	127–32378D

Ordering information

Description	Order code
7TE front panel set	127–32377G
6TE front panel set	127–32376K
Plug in card set	127–32526J
Guides (10)	127–32166B
Plug	017–1216B
Socket	017–1221K