ELEKTOR ELECTRONICS

THE INTERNATIONAL ELECTRONICS MAGAZINE

November 1994
£2·45

RS

ispLSI TECHNOLOGY

SOLID-STATE DISK

Debugging the 8031 series

Capacitance meter

Optical window comparator

Single-wire communication

Phantom power supply

9 770268 451043 11

AUDIO & HI-FI • COMPUTERS & MICROPROCESSORS • DESIGN IDEAS • RADIO, TELEVISION & COMMUNICATIONS • SCIENCE & TECHNOLOGY • TEST & MEASUREMENT

Existing subscribers to *Elektor Electronics* can order the Digital Multimeter, which provides a battery test facility, a diode test facility and 14 meter ranges, from our Dorchester office (use the order form on page 36) enclosing payment of £8·50.

FREE

WITH YOUR FIRST ANNUAL SUBSCRIPTION

940064X-I

PHANTOM POWER SUPPLY FOR GUITARS

Design by P. Goossens

The phantom power supply enables the power required by a guitar's integral preamplifier to be conveyed via the signal cable.

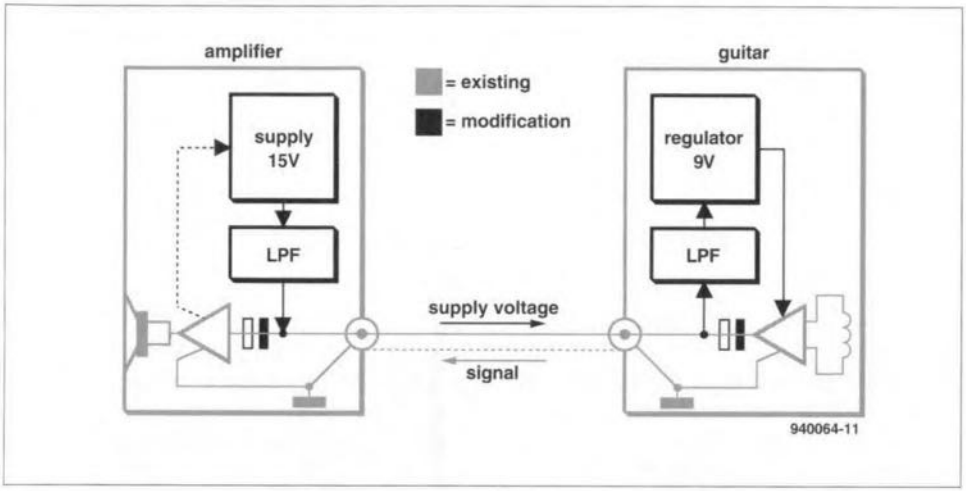


Fig. 1. Block schematic of the phantom power supply for guitars.

Many guitars are provided nowadays with an integral preamplifier. This preamplifier transforms the output impedance of the guitar element down so that this is no longer affected by the loading on it. This means that the connecting cable and the properties of the power amplifier have no influence on the sound of the instrument. Moreover, the preamplifier may include a tone control, which is of benefit to the sound reproduction.

Commercial guitar power amplifiers are sometimes provided with a power supply for the preamplifier in the guitar. This is normally done by a four-core screened cable instead of the usual two-core cable. This can not really be called a phantom power supply, however.

The basis of a true phantom supply is that the audio signals on the signal cable can be superimposed on to a direct volt-

age without any difficulty. This is shown schematically in **Fig. 1**. The 15 V supply can usually be taken directly from the power amplifier. The capacitor at the output of the power amplifier prevents the 15 V line entering the power amplifier, but it presents negligible impedance to the audio signals. The low-pass filter ensures that the audio signals are not loaded by the output of the power supply.

Similarly, a capacitor at the output of the preamplifier prevents the power supply affecting the signal output of the preamplifier, while another low-pass filter obviates any audio signals 'leaking away' via the power line.

The voltage drop across each of the low-pass filters is 2 V. The remaining 11 V is sufficient to allow a 9 V regulator to be used to ensure a stable supply to the preamplifier.

Circuit description

The upper part of the circuit in **Fig. 2** is (to be) built into the power amplifier; the lower part into the guitar. Connector K_1 is the input socket of the power amplifier and K_2 the output socket of the preamplifier. These sockets are interconnected by a single-core screened cable.

The input to the 15 V regulator section is derived from the power amplifier supply. If this is 18–25 V, R_1, T_1 and D_1 can be omitted. If it is higher than 25 V, these components serve as a series regulator that lowers the input to about 25 V. The value of the resistor is

$$R_1 = (U_{in} - 22) \; / \; I_z \; [\Omega],$$

where I_z is the current through D_1. The value of R_1 may be rounded off to the nearest E12 value.

The direct voltage of about 25 V is held stable at 15 V by regulator IC_1.

The low-pass filter is formed by gyrator T_2. The values of R_2 and C_6 permit only very slow variations of the current through T_2. This means that it presents a high impedance to the audio signal on K_1, but does not affect the direct voltage.

The lower part of the circuit is virtually a mirror of its upper part. Capacitor C_4 prevents any direct voltage to the preamplifier output. Transistor T_3 forms the gyrator that together with R_1 and C_1 functions as a low-pass filter. The direct voltage (about 13 V) at K_2 is held constant at 9 V by regulator IC_2. The current drain of the preamplifier must not exceed about 5 mA.

Construction

The construction of the supply will depend to a large extent on the available space in the guitar. Usually there will be enough room in the power amplifier for the regulator, gyrator and series regulator (if used). The existing connection between K_1 and the power amplifier input must, of course, be broken to insert C_5. If there is already a capacitor in series with the power amplifier input, this should be shorted by a wire bridge.

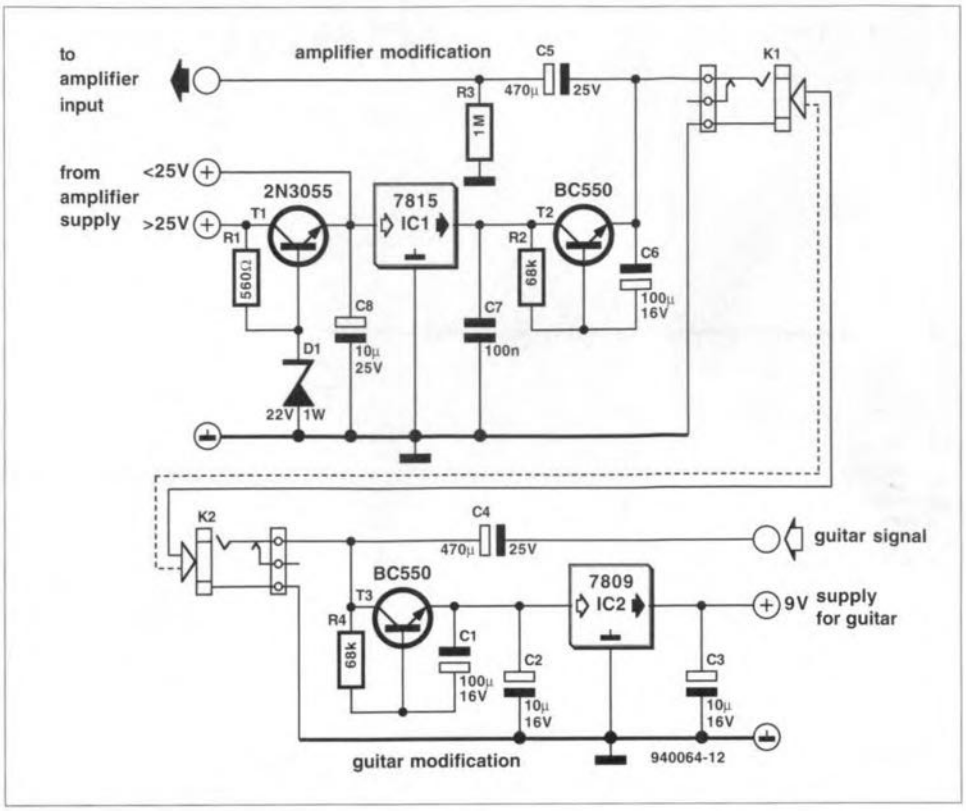


Fig. 2. Circuit diagram of the phantom power supply for guitars.

Smilarly, the existing connection between the preamplifier output and K_2 in the guitar must be broken to insert C_4.

It is best to fit the components for each of the two sections of the circuit on a small piece of prototyping, or similar, board. Resistor R_3 and electrolytic capacitors C_4 and C_5, however, should be soldered directly to the relevant jack socket. The prototype sections are shown in **Fig. 3**. But, as stated, the shape of that for building into the guitar depends entirely on the space available in the guitar.

Finally, the direct voltage from the power amplifier is best taken directly from the (electrolytic) buffer capacitors in the supply of that amplifier. For the earth connections, use only that at the jack sockets to obviate any earth loops.

[940064]

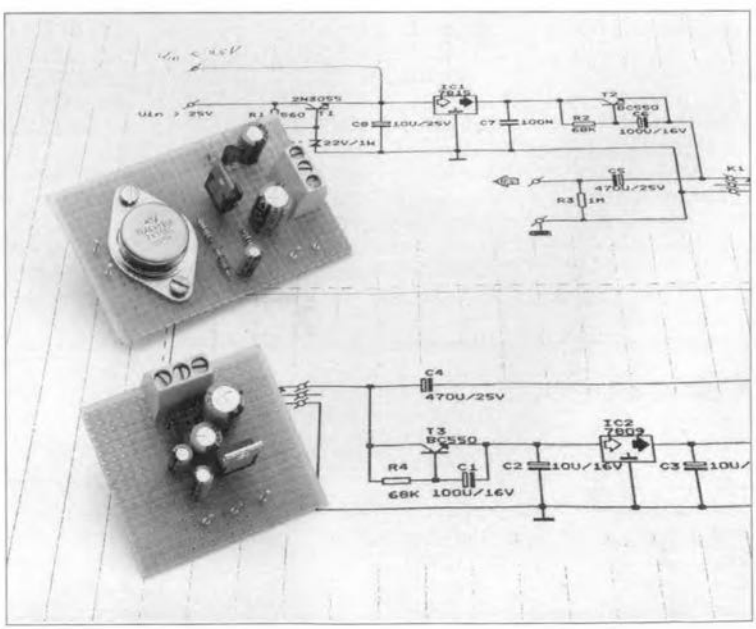


Fig. 3. Prototypes built on protopying boards.

LOW-COST ispLSI PROGRAMMER

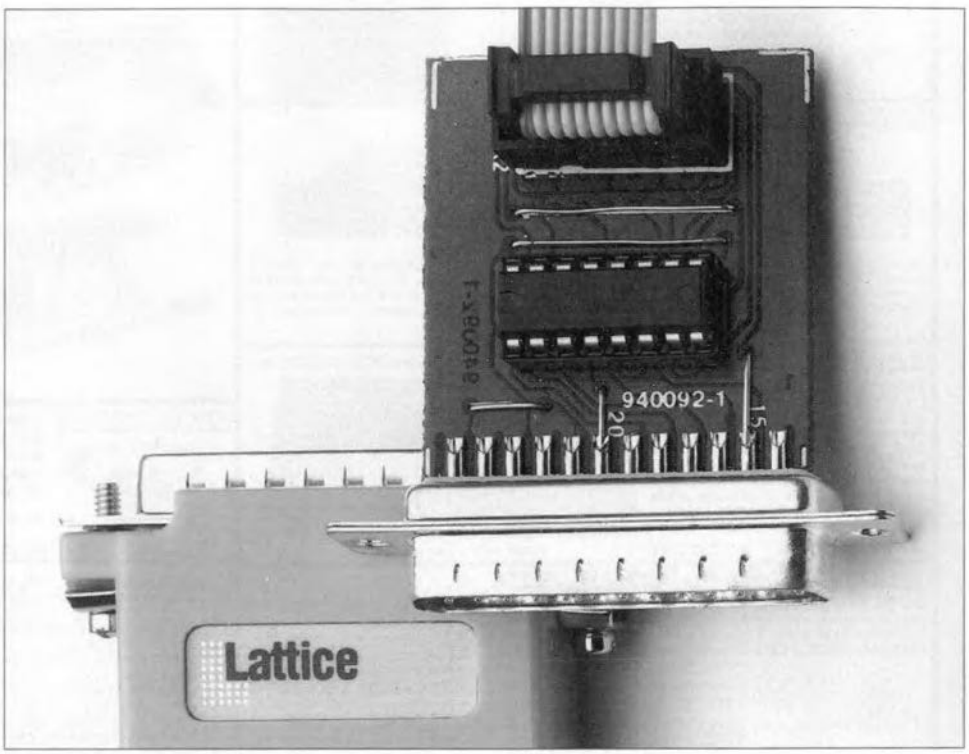
Programming your own logic components without breaking the bank is now possible thanks to ispLSI devices from Lattice. In addition to a general introduction to the structure and operation of ispLSI devices, this article presents the simplest possible ispLSI programmer which can be connected to a PC.

By our editorial staff.

Source: Lattice Semiconductor Corporation.

THERE are currently four members in the ispLSI family from Lattice. The abbreviation isp stands for in-system-programmable, which means that these components can be programmed without being removed from their application circuit. A second series is available with only a 'p' prefix. These devices are intended for mass production, and can not be programmed in-circuit. The letters LSI stand for large-scale integration, a term which was first used with the arrival of complex TTL and CMOS functions on a chip.

The main difference between the four ispLSI components is the number of GLBs (generic logic blocks). The smallest component has 16 of these blocks, the largest, 48. The circuits consist of logic, registers, I/O cells, different clock circuits, a 'global routeing pool' (GRP) and an 'output routeing pool' (ORP). An ispLSI device selection guide is shown in **Table 1**. The table clearly shows the differences between the four devices. The structure of the ispLSI1032 device is shown in **Fig. 1**.



Generic logic block (GLB)

A GLB has 18 inputs, four outputs, four D-type registers, and all logic necessary to implement about 90% of all four-bit logic functions. Internally the GLB may be subdivided into an AND array, a product term sharing array (PTSA), output logic macro cells (OLMCs) and control functions. The simplified structure of the logic blocks is shown in **Fig. 2**.

AND array and PTSA

The AND array (**Fig. 2** connects the different input signals to the 20 product terms which can be used in the GLB. The product terms originate from signals at the 18 GLB inputs and the programmed fuses in the AND array. Either 4, 5 or 6 of these product terms can be ADDed ('summed') with the aid of OR gates. Next, the PTSA ensures that the outputs of the OR gates and/or the product terms are combined with the desired GLB outputs. It is even possible to have a product term control one output directly. The GLB output signals are subsequently available for further processing via the global routeing pool (GRP) and the output routeing pool (ORP). Should speed be a decisive factor, it is possible to take PTSA outputs directly to IC outputs. This so-called bypass function eliminates the propagation delay introduced by the PTSA and the associated XOR gate.

Function	ispLSI1016	ispLSI1024	ispLSI1032	ispLSI1048
GLB	16	24	32	48
I/O	32+4	48+6	64+8	96+10
Registers	96	144	192	288
GRPs	1	1	1	1
ORPs	2	3	4	6
Clocks	5	5	5	5

Table 1. An ispLSI device selection guide.

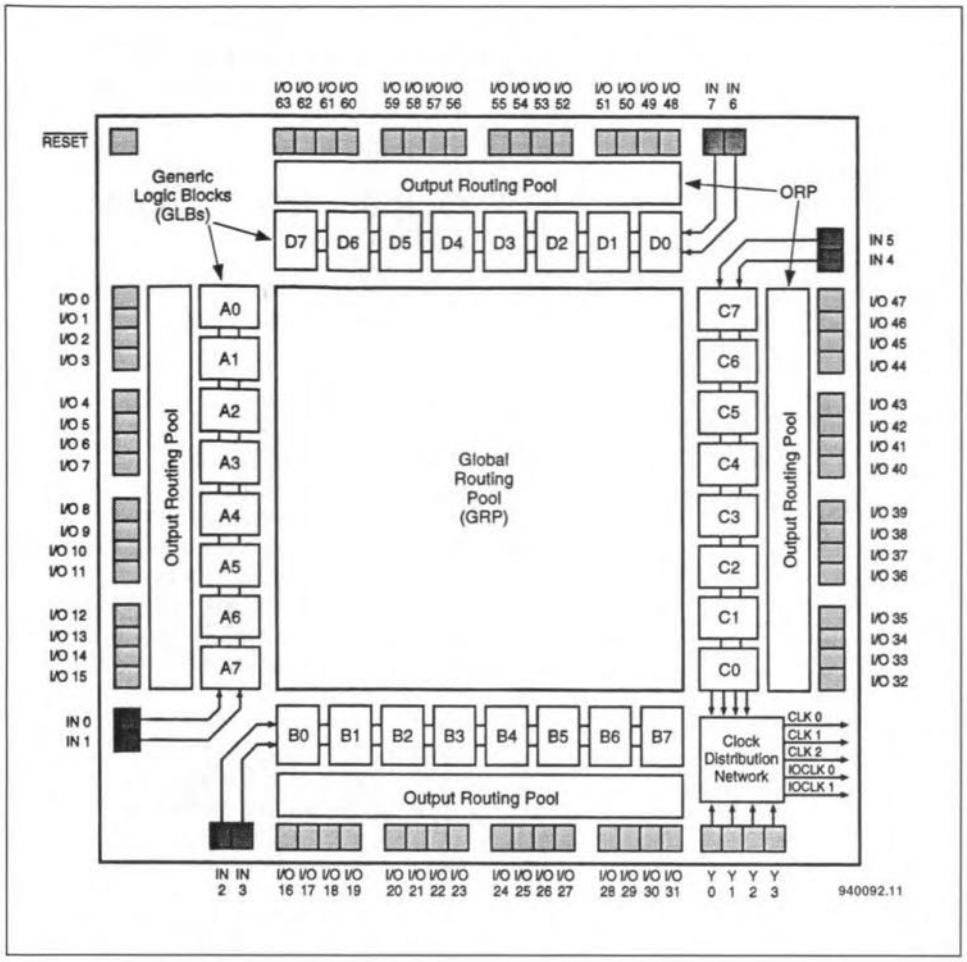


Fig. 1. Functional diagram of the ispLSI1032. The higher the type number, the greater the number of logic functions available in the IC. The ispLSI1016, for instance, has only half the functionality shown here.

OLMCs

The outputs of the GLBs are in fact 'output logic macro cells' which contain four D-type bistables with XOR gates at their inputs. These gates may be used to create logic functions. Alternatively, they may be used to turn the D-type bistable into a J-K or T-type. Thanks to the power and flexibility of the PTSA, practically any combination of features is possible. That is demonstrated by the example configuration shown in **Fig. 3**. The XOR gate is used on output O3, while the bypass function is used for four terms on output O2. Output O1 uses only one of the five inputs offered by the XOR gate. Finally, O0 combines the four remaining product terms with the seven product terms at the input of the OR gate. The end result is a logic function consisting of 11 product terms.

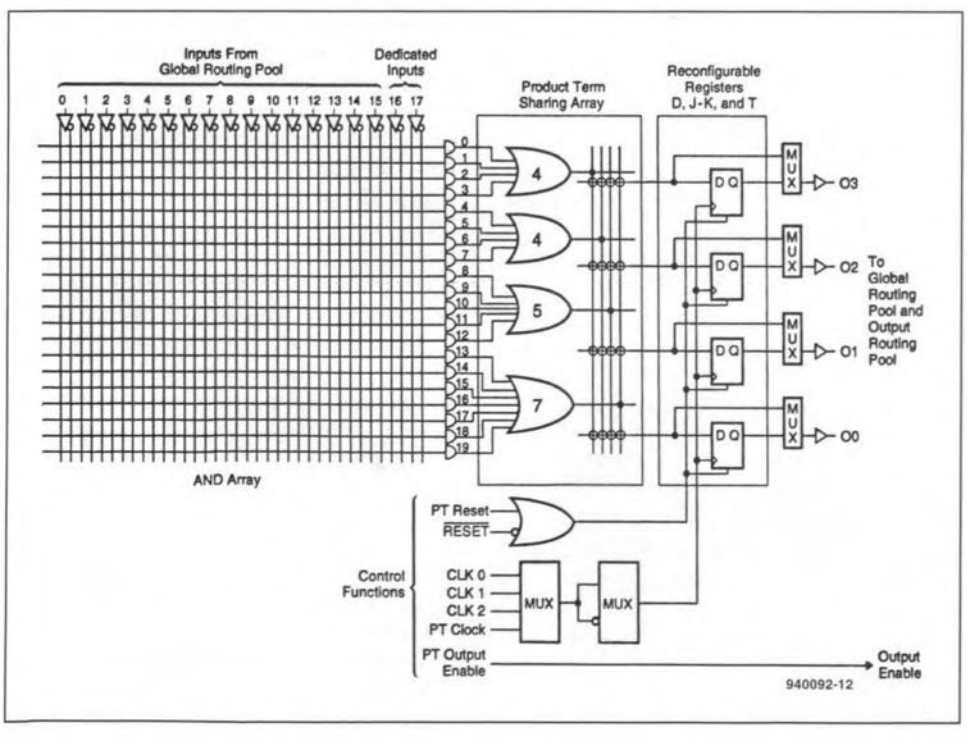


Fig. 2. The structure of the GLB guarantees a high degree of functionality.

Control functions

Signals necessary to control the GLBs are supplied by the 'control' section. The synchronous clock signal for the registers is supplied by one of three sources provided by the 'clock distribution network'. An asynchronous clock signal may be applied via PT12 in the GLB. The reset signal for the GLB comes from the 'global reset' input, or inputs PT12 or PT13 of this unit. The output enable signal for the I/O cells which form an integral part of the GLB may be applied to input PT19. If it is necessary for this signal to contain a logic function, the relevant term is not available for other functions within the GLB.

I/O cells

All ispLSI devices have a number of I/O connections which are user-configurable. These connections are linked to I/O cells which may be set up as bistables or latches, as illustrated in **Fig. 4**. The polarity of the output enable (OE) signal may be inverted with the aid of the internal multiplexer. The signal used to create the OE signal can be furnished by the ORP in two ways. For fast, time-critical applications, the ORP signal is available 'via bypass'. The diagram in Fig. 4 also shows that each I/O cell offers a choice between a bistable (register) or a latch function at its input. One of these options is selected with the aid of the R/L (register/latch) signal. After the intermediate storage function, the input signal goes directly to the GRP. For combinational logic inputs, the storage function can be disabled via a multiplexer.

A further option concerns triggering, which may occur on the positive-going or negative-going edge of the two clock signals, IOCLK 0 and IOCLK 1. The global reset signal is used to reset each I/O cell at power-on. As soon as the external reset input is made logic low, all I/O cells are switched to their 'low' state. A pull-up resistor is auto-

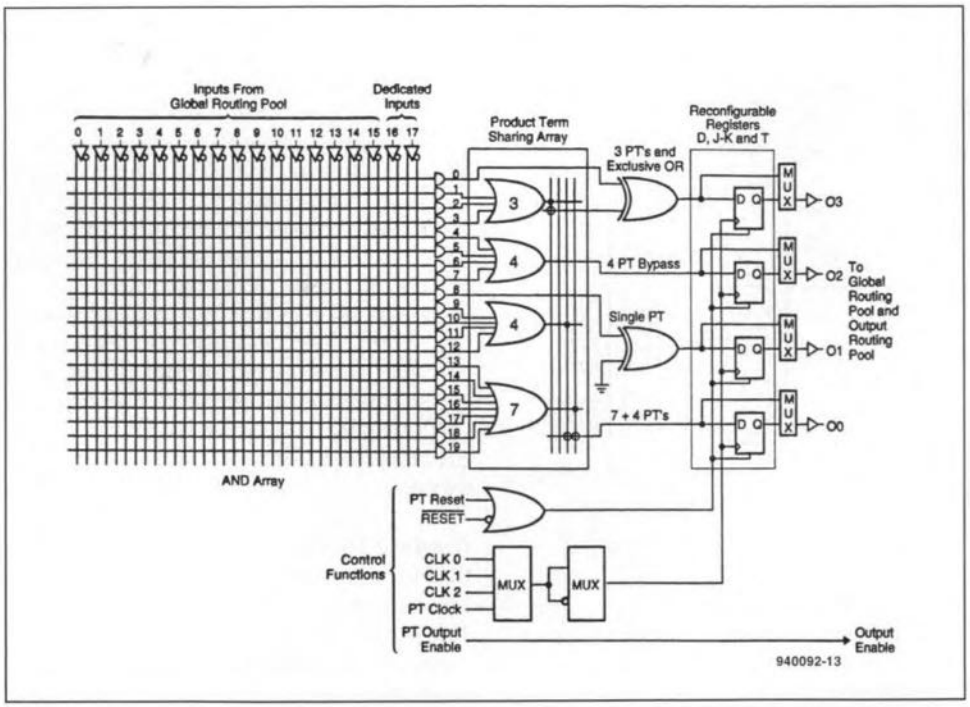


Fig. 3. Overview of different configurations selected in the GLB.

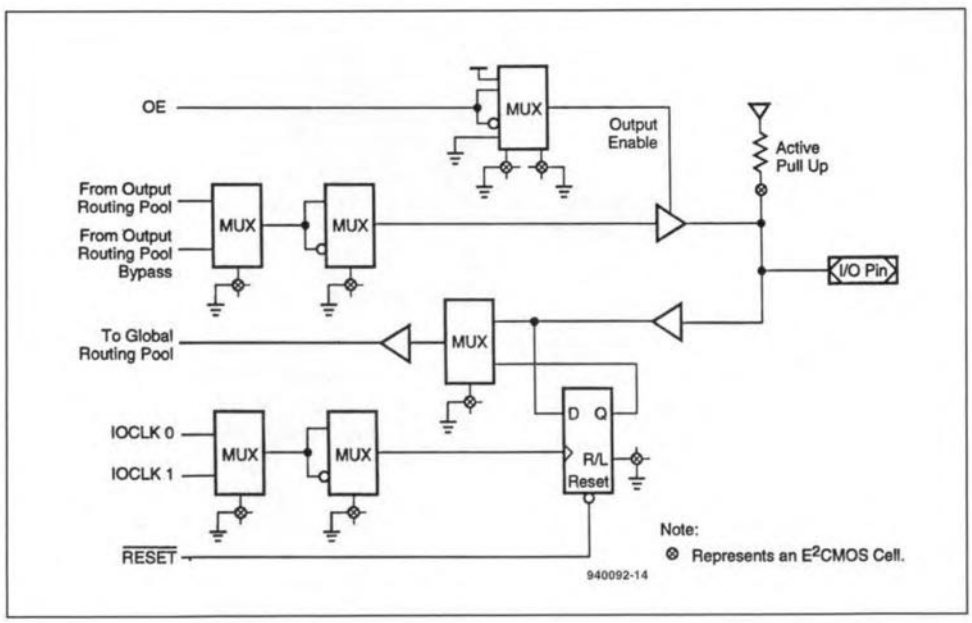


Fig. 4. Structure of an I/O cell. The functions of both the input and the output are fully programmable.

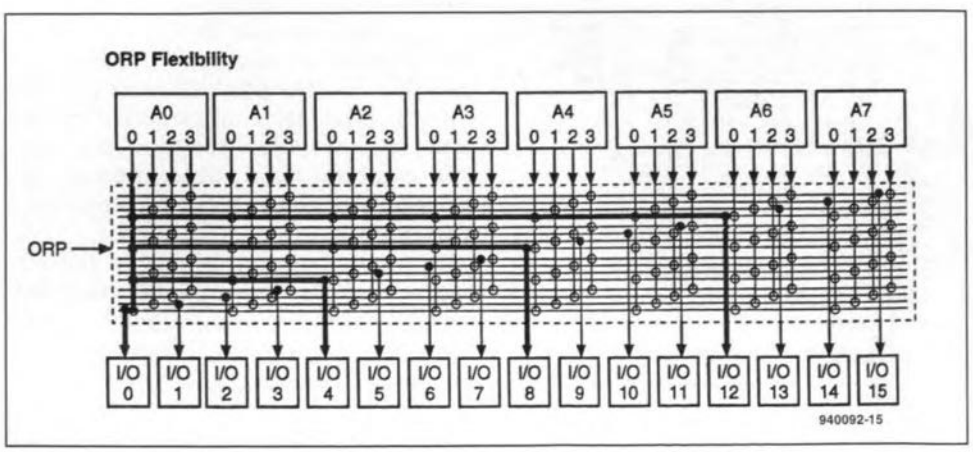


Fig. 5. Illustrating the flexibility of the ORP (output routeing pool).

matically connected if an I/O pin remains unused. That is done to prevent floating inputs, which are prone to cause noise and unnecessary current consumption.

Output routeing pool (ORP)

Functions of the ORP include routeing the GLB signals to suitable I/O pins, assigning pin functions (in a flexible manner), and bypassing functions to achieve high operating speeds.

An ORP is shared by eight GLBs and 16 I/O pins. That makes the ORP a major component in the so-called megablock, which consists of an ORP, 16 I/O cells, eight GLBs, two dedicated inputs, and one OE connection. The ispLSI1016 contains two of these megablocks; the ispLSI1024, three; the ispLSI1032, four; and the ispLSI1048, six.

The example in **Fig. 5** shows how output signal GLB A0 is fed to I/O lines 0, 4, 8 and 12. The other signals from the GLBs are routed in a similar manner. The ORP bypass is useful when designing short, fast signal paths. The use of bypasses must be limited, however, to prevent blocking the rest of the ORP.

Do your own programming

The ispLSI devices mentioned in this article are simple to program, that is, configure, without expensive programming tools. In fact, a programmer consists of a simple TTL buffer/line driver and a handful of passive parts. The circuit diagram of the ispLSI programmer is shown in **Fig. 6**. The unit is connected to the parallel printer (Centronics) port available on any PC. The cable conveys eight signals from the PC to a dedicated connector which must be available on the board containing the ispLSI device.

The artwork of the printed circuit board designed for the low-cost ispLSI programmer is shown in **Fig. 7**. Unfortunately this board is not available ready-made through our Readers Services. With components so few and so common, construction should not present problems. Simply mount all parts on to the board, solder carefully, connect the header to the cable, check for short-circuits and bad solder spots, and hook up the programmer to the PC.

The programming routines which belong to the present programmer may be found in the handbooks published by Lattice. These programs are written in C, and are simple to turn into executable code if have a suitable C compiler. The timing diagram in **Fig. 8** should give you an idea of the kind of

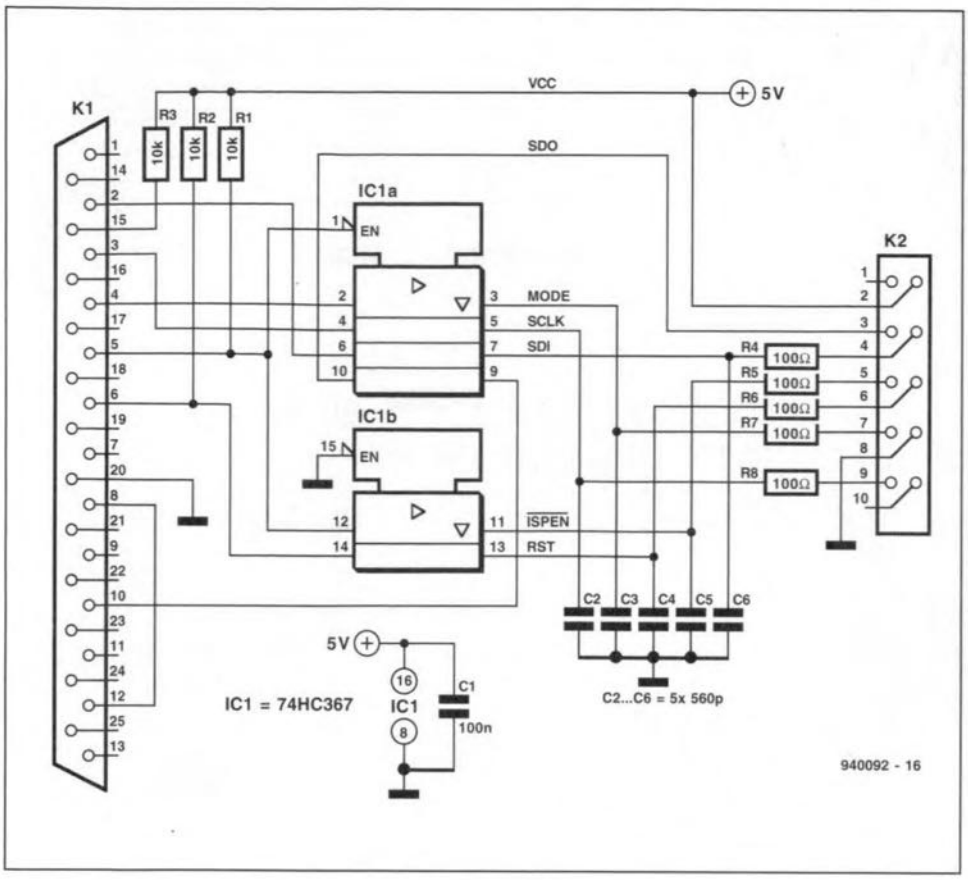


Fig. 6. Circuit diagram of the low-cost ispLSI programmer, which is connected to the PC's Centronics port.

signals generated by the programming software. On a final note, those of you who have purchased the ispLSI Starter Kit from Lattice (to be discussed in next month's issue) need not build the circuit in Fig. 6, because that is contained in the kit.

(940092)

COMPONENTS LIST

Resistors:
R_1,R_2 = 10kΩ (SMT)
R_3 = 10kΩ (SMT)
R_4-R_8 = 100Ω (SMT)

Capacitors:
C_1 = 1nF (SMT)
C_2-C_6 = 560pF (SMT)

Semiconductors:
IC_1 = 74HC367

Miscellaneous:
K_1 = 25-way D connector (female).
K_2 = 10-way flatcable connector for PCB mounting.
10-wire flatcable, max length 150 cm.
8-way SIL-socket.

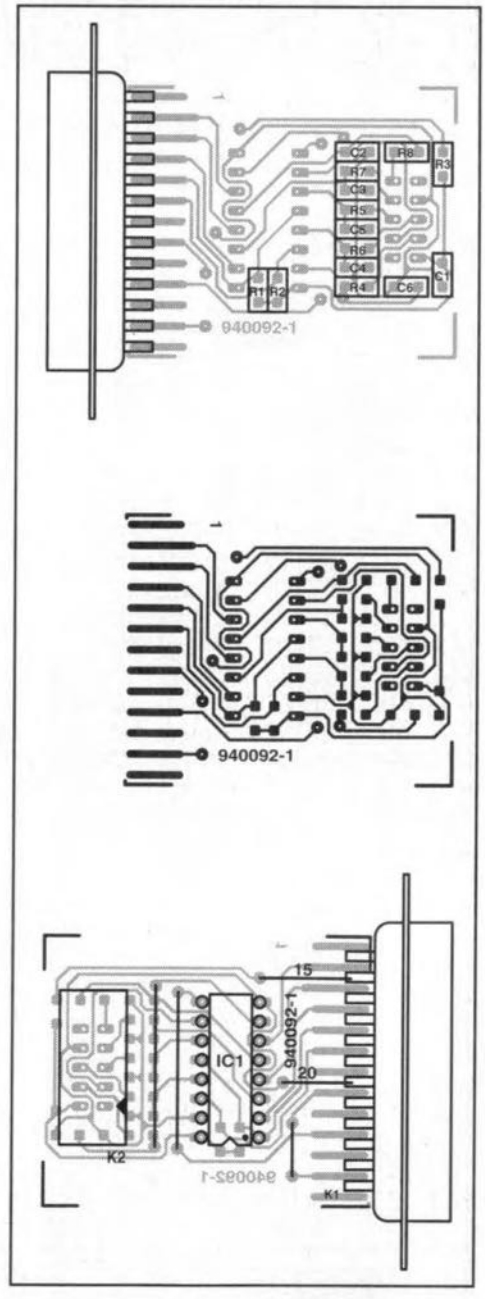


Fig. 7. PCB design for the low-cost ispLSI programmer (PCB not available ready-made through the Readers Services).

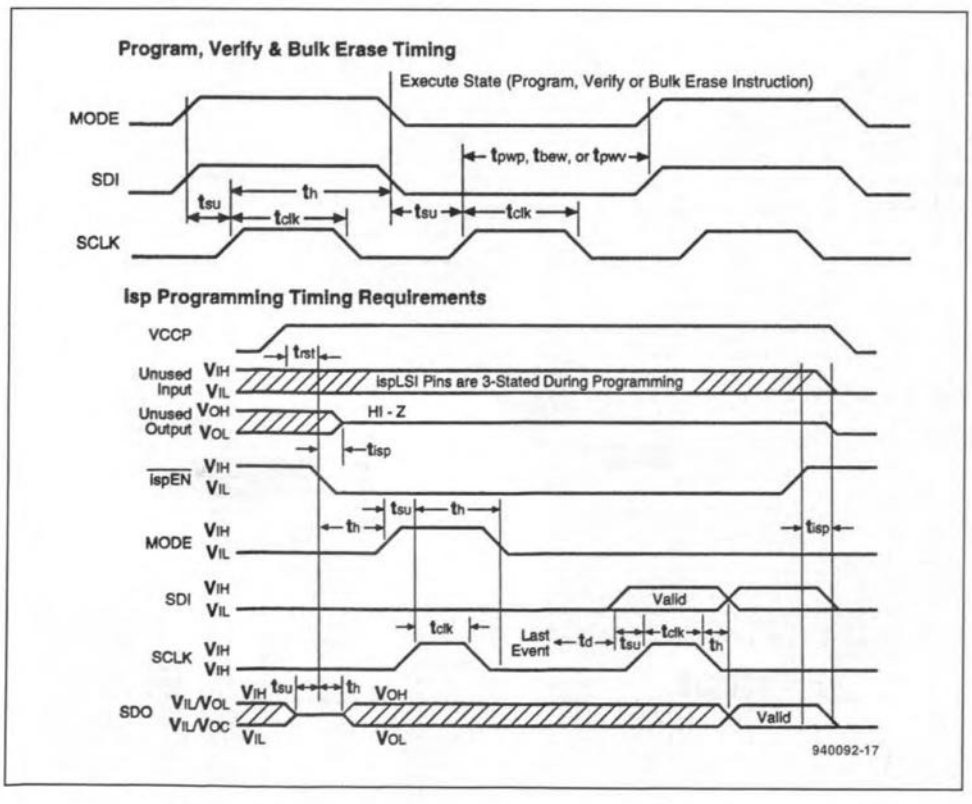
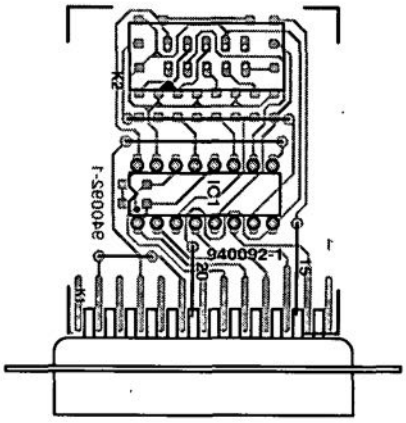
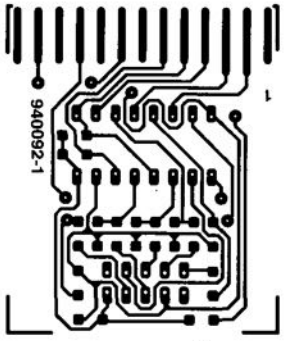
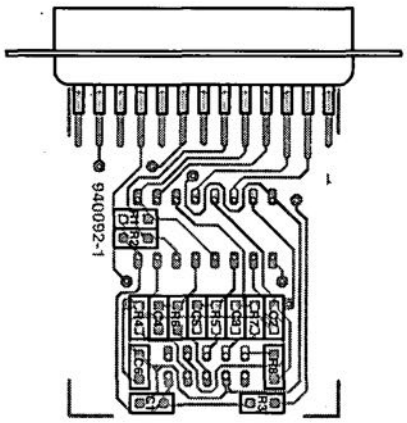


Fig. 8. Timing diagram of signals needed to program an ispLSI device.


R2
IC1
940092-1
15
1


940092-1
1


940092-1
1

SIMPLE CAPACITANCE METER

Design by H. Bonekamp

A circuit is described that is a good compromise between complexity and accuracy. The results can be displayed on a standard digital voltmeter or multimeter.

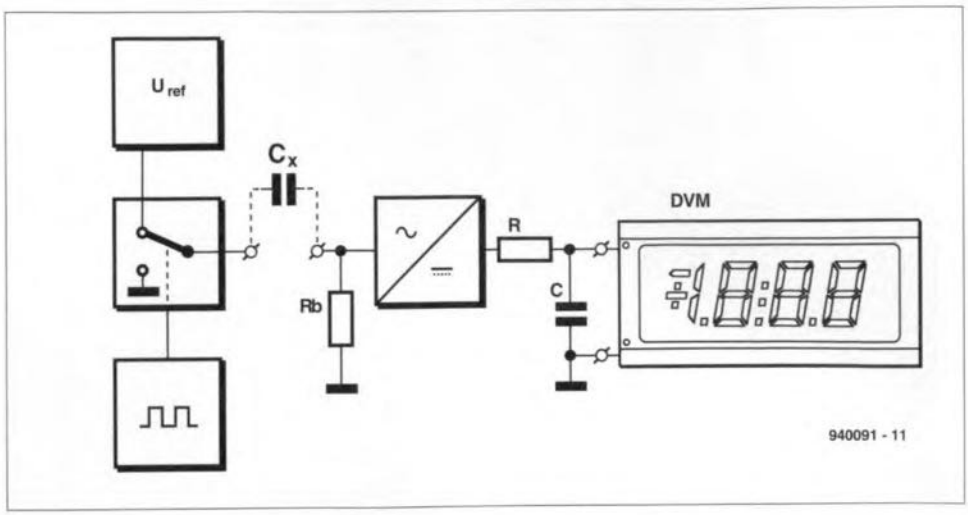


Fig. 1. Block diagram of the simple capacitance meter.

Measuring the value of a capacitor seems a fairly simple matter: charge it for a certain time by a given constant-value direct voltage and measure the potential developed across it, from which the capacitance can be easily calculated. Or direct an alternating current through the capacitor and, by synchronous detection, compare the potential developed across the capacitor with a reference voltage. The practical execution of these methods is, however, not as straightforward as it appears.

In the present circuit, the capacitor to be measured, C_x, is made part of a (non-ideal) *RC* differentiating network with a step response. The design is based on the fact that the average output voltage of such a network is directly proportional to the capacitance of the capacitor, provided that the voltage is averaged for a sufficiently long time.

Block schematic

Since it is imperative for good results that the test signal is accurate and reliable, the design does not use a conventional square-wave oscillator, but a rather more elaborate one. It consists of three elements: a reference-voltage source, U_{ref}, an electronic switch and a

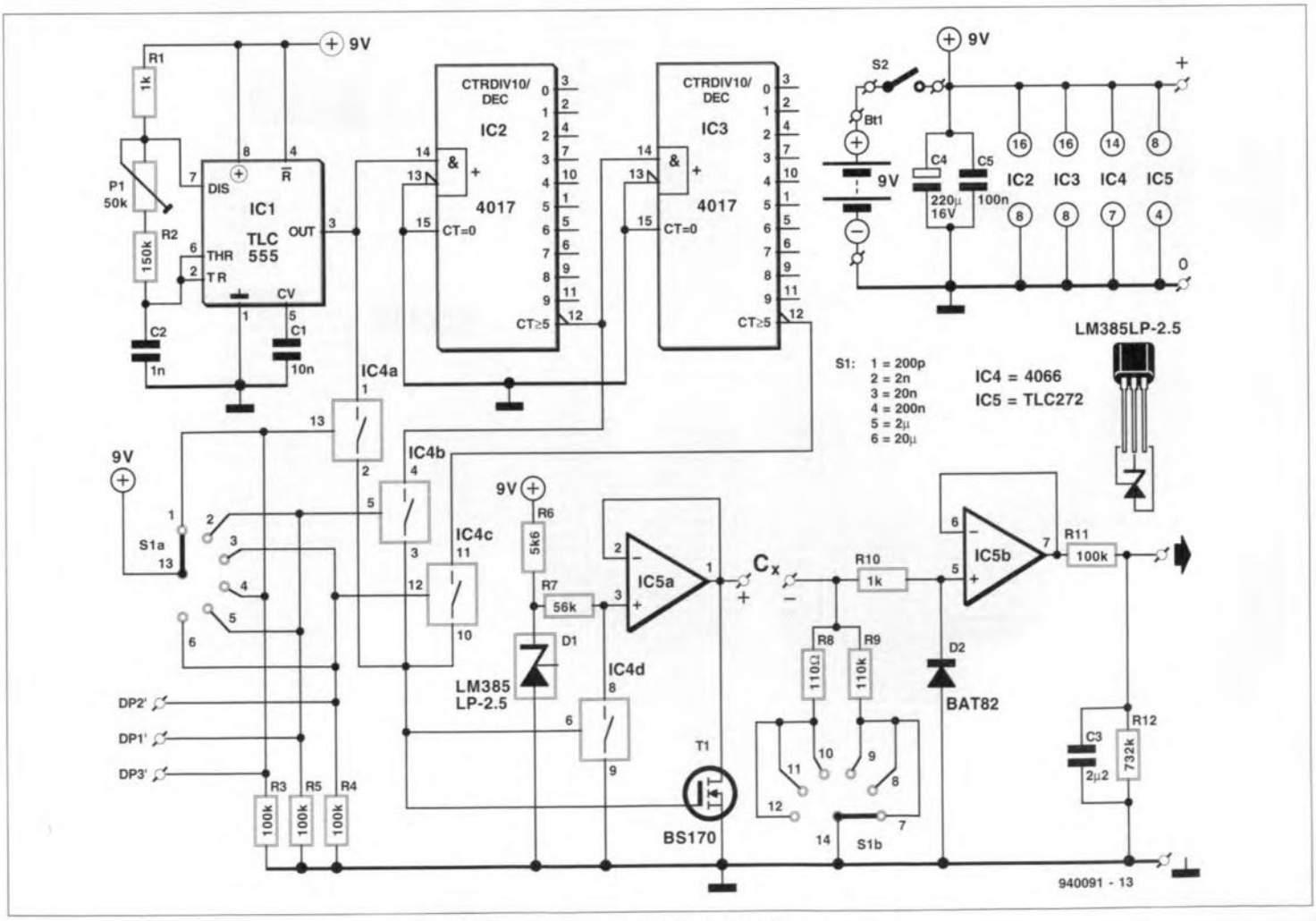


Fig. 2. Circuit diagram of the capacitance meter.

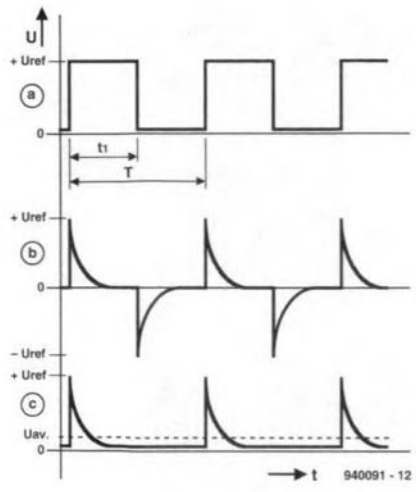


Fig. 3. Signals at various points in the circuit—see text.

square-wave oscillator—see **Fig. 1**. The oscillator operates the switch and this in turn ensures that one terminal of C_x is connected alternately to U_{ref} and to earth. The voltage applied to C_x is shown in **Fig. 3a**.

Differentiating network R_b-C_x converts the voltage applied to C_x to a train of positive and negative pulses as shown in **Fig. 3b**. Since both the test signal and R_b are fixed, any variations in these pulses must be caused by C_x. This means that to get the required information the pulses need to be half-wave rectified (~/=) and averaged by an integrating network, *RC*.

The direct voltage thus obtained, U_{av}—see **Fig. 3c**—is directly proportional to the value of C_x and may be applied to a digital voltmeter (module) or multimeter (not described in this article). The scale of the meter may be calibrated in farad with the aid of a standard test capacitor.

Circuit description

The circuit diagram of the meter is shown in **Fig. 2**. The measuring range is split into six by S_1: 200 pF, 2 nF, 20 nF, 200 nF, 2 µF, and 200 µF. The switch changes the frequency of the oscillator as well as the value of 'R_b', which consists of R_8 and R_9. The values of these resistors have a ratio of 1:1000.

The square-wave oscillator is formed by IC_1, IC_2 and IC_3. Switch section S_{1a} operates electronic switches IC_{4a}, IC_{4b} and IC_{4c}, each of which selects one frequency: 4.13 kHz, 413 Hz and 41.3 Hz respectively. Decadic dividers IC_2 and IC_3 ensure that the test signal (at their pin 12) remains a square wave.

The electronic switches negate the inevitable crosstalk between S_{1a} and S_{1b} and thus prevent any unnecessary measuring errors. A further benefit of these switches is that signals may be tapped directly from S_{1a} to control the decimal point of the digital voltmeter module: DP_1, DP_2, DP_3.

The reference potential of 2.5 V is

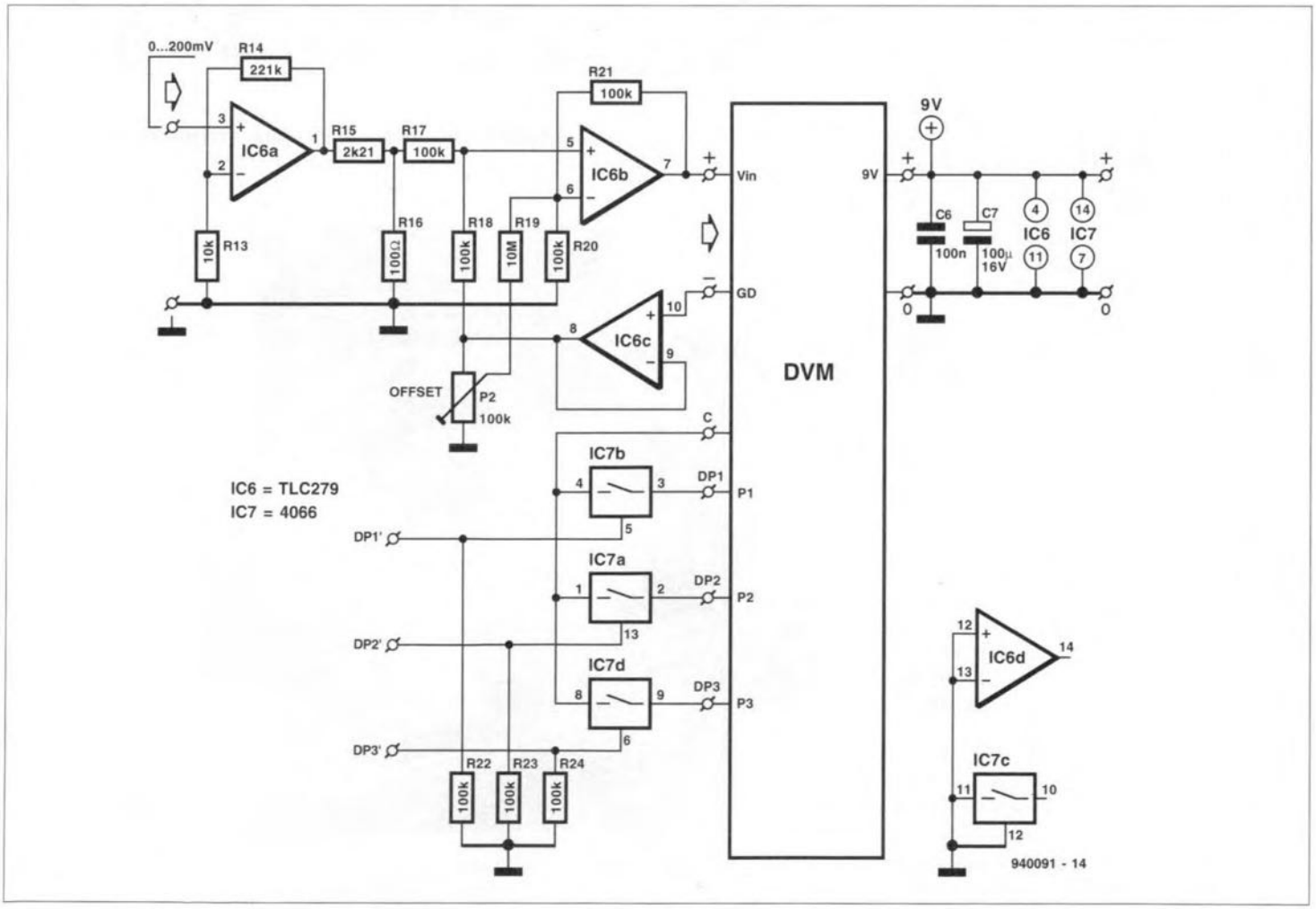


Fig. 4. Circuit diagram of the interface; not every constructor may need this—see text.

established by voltage reference diode D_1. Its low level (relative to the battery voltage, U_b) ensures that any variations in U_b do not affect the measurement. The voltage applied to C_x is switched between U_{ref} and earth by electronic switch IC_{4d}.

Since the test voltage must not be loaded by C_x, it is applied to the capacitor via buffer IC_{5a}. However, even the operational amplifier can not cope with the discharge current of C_x when S_{1b} is in positions 4, 5 or 6. Therefore, T_1 is connected in parallel with the output of IC_{5a}. This FET begins to conduct as soon as IC_{4d} is connected to earth and thus takes a large part of the discharge current.

Because the signal differentiated by C_x and R_8 or R_9 must not be loaded if measuring errors are to be avoided, it is applied to the output via IC_{5b}. This buffer might also serve as half-wave rectifier, were it not for the fact that it can not handle negative voltages greater than –0.3 V. Therefore, the half-wave rectification is effected by Schottky diode D_2.

The output voltage is averaged and brought to the required level of 200 mV by integrating/dividing network R11-R_{12}-C_3. Since, to ensure good linearity of the output, the capacitor is a polypropylene type (high impedance), the digital voltmeter module must have an input impedance of ≥10 MΩ.

If a digital voltmeter module is used (as distinct from a complete digital voltmeter or multimeter), an interface as shown in **Fig. 4** is required between it and the capacitance meter circuit. This makes it possible to supply both the meter and the module from the same battery and also, as stated earlier, to control the decimal point of the module by the meter circuit.

The interface is needed because inexpensive DVM modules using the Type 7106 IC require an electrically isolated supply. This can be bypassed by providing the input signal to the module with an offset voltage equal to the potential at the earthy terminal of the module. In fact, therefore, the circuit in **Fig. 4** adds that potential to the input voltage.

In **Fig. 4**, IC_{6b} is the summing circuit. The signals to be added together are applied across R_{17} and R_{18}. They need to be of low impedance to prevent summing errors. Also, IC_{6b} provides a gain of 3 dB, since the summing method used results in an attenuation of –3 dB.

Buffer IC_{6c} prevents loading of the earthy potential of the DVM module and ensures that the summed signal is of low impedance.

Circuit IC_{6a} fulfils a similar function at the input side and also provides an amplification of ×23.1. Subsequently, the signal is attenuated by R_{15}-R_{16} by

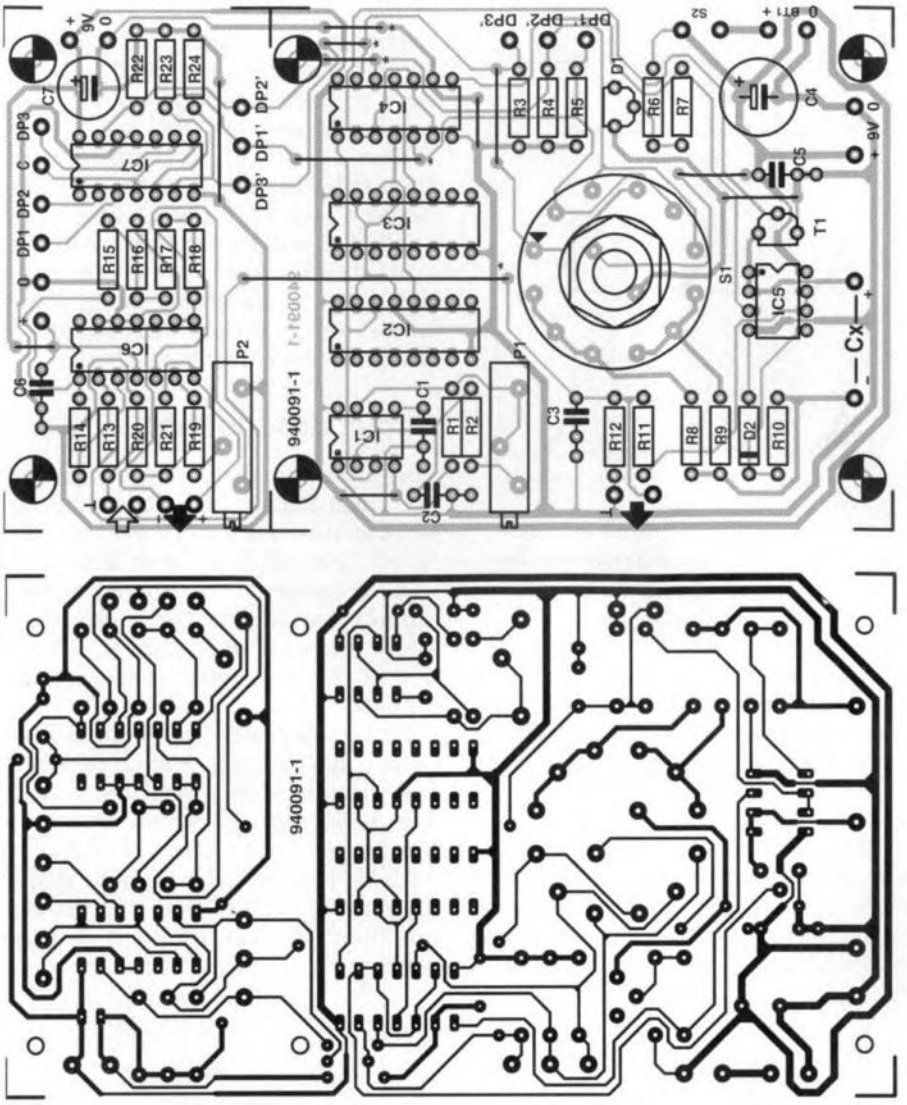


Fig. 5. Printed-circuit board for the capacitance meter and interface.

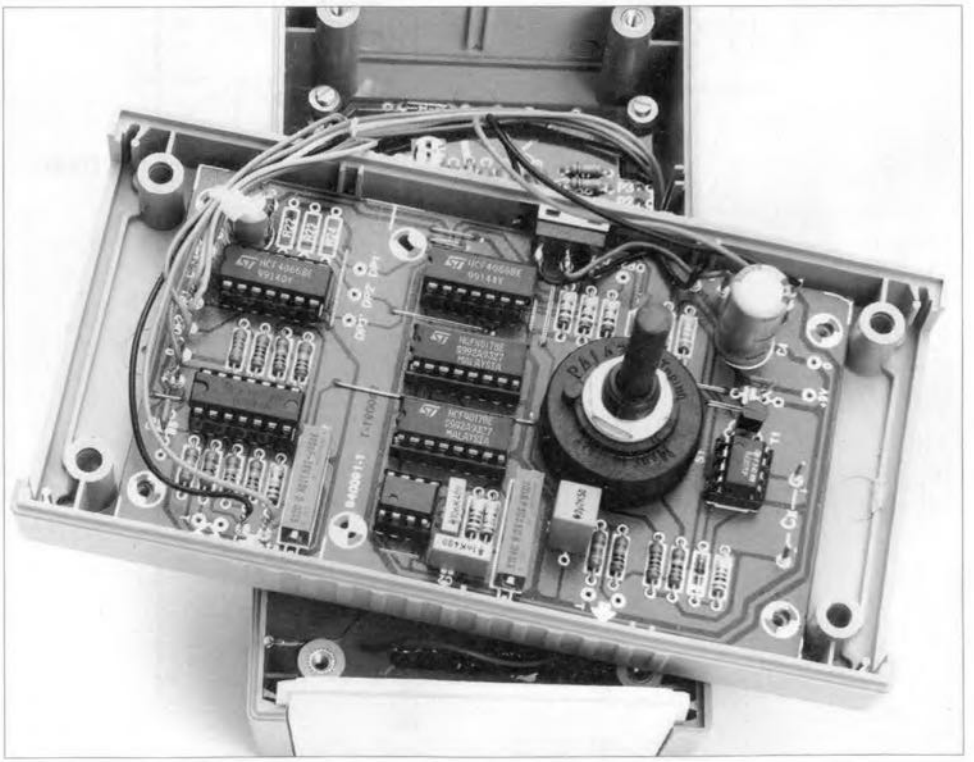


Fig. 6. Completed prototype board of the capacitance meter.

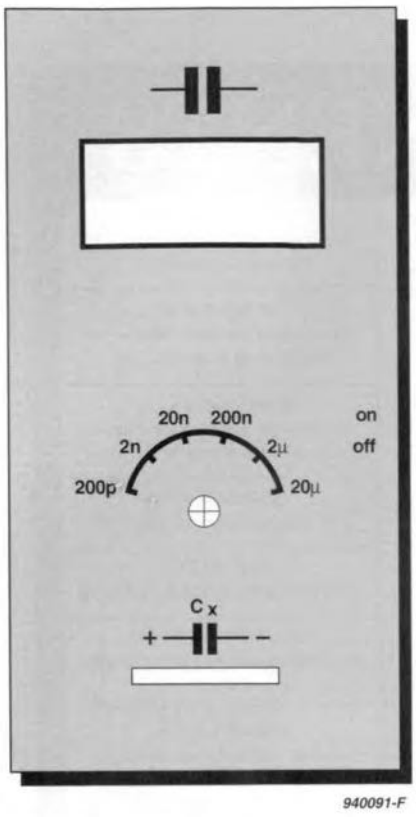


Fig. 7. Suggested front panel for the capacitance meter.

an identical factor. This arrangement means that the absolute error is reduced by the same factor. Note that a TLC279 has a much lower offset than the specified alternative TLC274.

Any residual errors, including that caused by the tolerances of R_{17}–R_{20}, can be compensated by P_2.

The decimal points are controlled by S_{1a} in the meter circuit via electronic switches IC_{7a}. IC_{7b} and IC_{7c}. In a number of cases (to be determined by trial and error), resistors R_{22}–R_{24} may not be needed.

Construction

The printed-circuit boards for the meter circuit and the interface are combined as shown in **Fig. 5**. If a module is not used (and the interface is thus not needed), the interface section may be cut off the board. However, if the specified enclosure is used, it is better not to cut the section off, because of the fixing holes it provides.

Completion of the board is straightforward and should not present any difficulties. Note, however, that six-position switch S_1 is mounted directly on to the board. The completed prototype board is shown in **Fig. 6**.

In the specified case, fit the DVM module (if used) and the battery in the lid; the battery between the module and S_1. A suggested front panel layout is given in **Fig. 7**.

Alignment

Switch on the meter with open-circuit C_x terminals. Adjust P_2 until the display (or voltmeter used) shows 0.00.

Connect a 10 nF reference capacitor to the C_x terminals. If a reference model is not available, use a 1% polystyrene type. Set S_1 to position 4 and adjust P_1 until the display (or meter used) reads 10.00.

Set S_1 to position 3 and verify that the display (or meter) reads 100.00. If it does not, readjust P_1 slightly: this may need to be done a couple of times in both ranges.

Finally

The relative measurement error is <1.5% in all ranges. The supply voltage may vary between 6.5 V and 10.0 V. Since the current drain does not exceed 12 mA, a 9 V (PP3) battery will enable quite a few measurements to be carried out.

Parts list

METER CIRCUIT

Resistors:
R_1, R_{10} = 1 kΩ
R_2 = 150 kΩ
R_3–R_5 = 100 kΩ
R_6 = 5.6 kΩ
R_7 = 56 kΩ
R_8 = 100 Ω, 1%
R_9 = 110 kΩ, 1%
R_{11} = 100 kΩ, 1%
R_{12} = 732 kΩ, 1%
P_1 = 50 kΩ 10-turn preset (horizontal)

Capacitors:
C_1 = 10 nF
C_2 = 1 nF, polypropylene
C_3 = 2.2 µF, polypropylene
C_4 = 220 µF, 16 V, radial
C_5 = 100 nF

Semiconductors:
D_1 = LM385LP-2.5
D_2 = BAT82
T_1 = BS170

Integrated circuits:
IC_1 = TLC555
IC_2, IC_3 = 4017
IC_4 = 4066
IC_5 = TLC272

Miscellaneous:
S_1 = 2-pole, 6-position rotary switch
S_2 = SPST switch
BT_1 = 9 V battery with connecting clip
Enclosure: 150×80×45 mm
(6×3$^1/_8$×1$^3/_4$ in); for instance G416 from Velleman (via Maplin)
SIL connector (for C_x terminals)
PCB Order No. 940091-1

INTERFACE

Resistors:
R_{13} = 10 kΩ, 1%
R_{14} = 221 kΩ, 1%
R_{15} = 2.21 kΩ, 1%
R_{16} = 100 Ω, 1%
R_{17}, R_{18}, R_{20}, R_{21} = 100 kΩ, 1%
R_{22}–R_{24} = 100 kΩ (see text)
R_{19} = 10 MΩ
P_2 = 100 kΩ 10-turn preset (horizontal)

Capacitors:
C_6 = 100 nF
C_7 = 100 µF, 16 V, radial

Integrated circuits:
IC_6 = TLC279 (or TLC274 – see text)
IC_7 = 4066

Miscellaneous:
DVM = digital voltmeter display module

[940091]

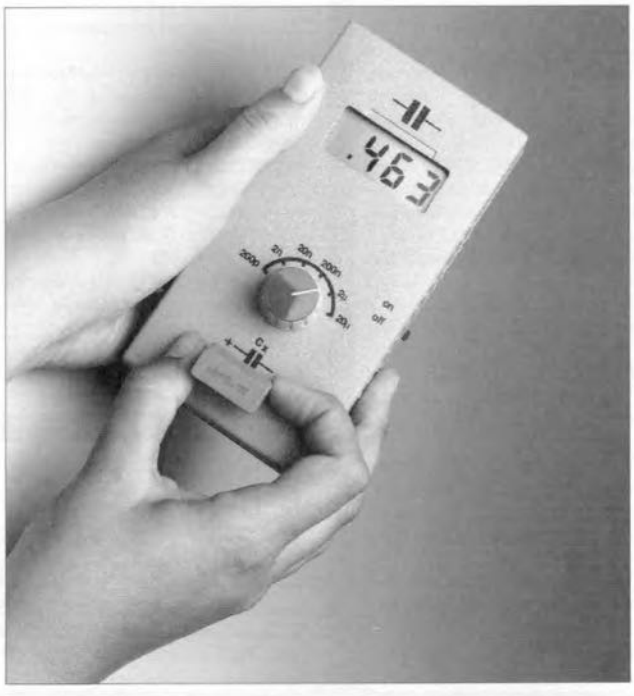
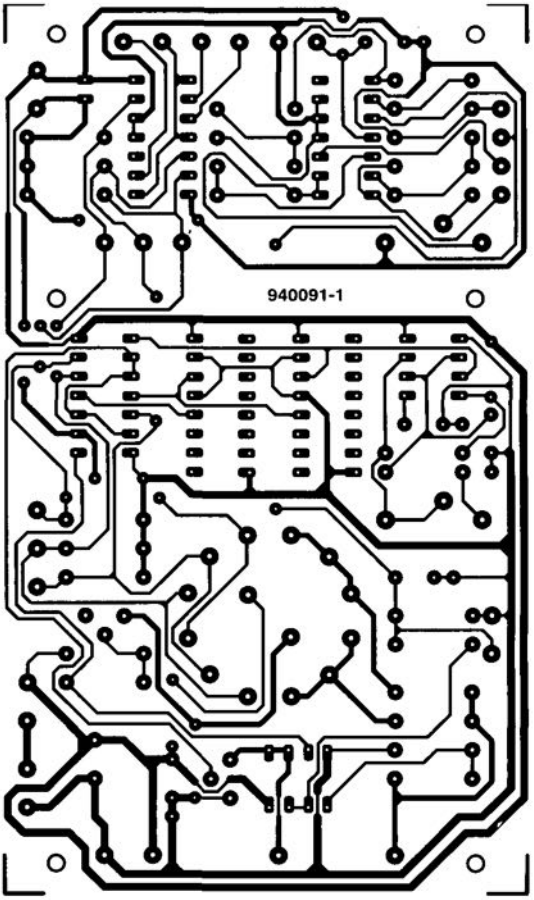
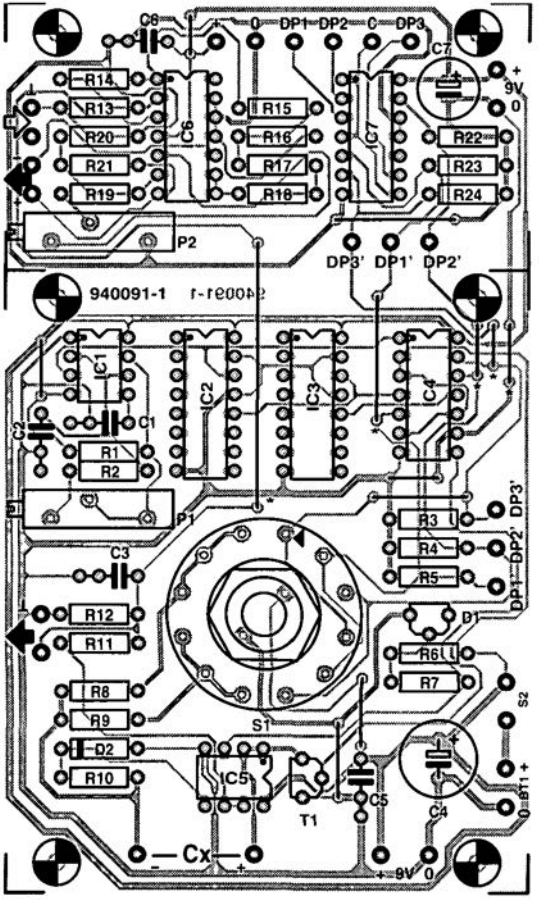


Fig. 8. For accurate results, the capacitor on test should not have long terminating wires.


940091-1


940091-1
C6
C7
DP1
DP2
DP3
9V
R14
R13
R20
R21
R19
R15
R16
R17
R18
R22
R23
R24
P2
DP3'
DP1'
DP2'
IC1
IC2
IC3
IC4
C1
C2
R1
R2
P1
C3
R3
R4
R5
R12
R11
R8
R9
D2
R10
R6
R7
D1
S1
S2
C5
T1
C4
Cx
0

IN-CAR AUDIO AMPLIFIER PART 2

Design by T. Giesberts

In this instalment, the power inverter will be discussed. Note that this requires the car battery to be in good condition, since at full drive in both stereo channels it draws a current of some 50 A.

At an AF output of 2×200 W, the power supply needs an input of about 600 W. The present inverter can provide 650 W continuously and, because of its large (2×40,000 µF) reservoir capacitors, has a substantial peak reserve capacity. Its efficiency is a creditable 86%. In view of the large current, the board for it is provided with broad tracks, while most cables used have a cross-sectional area of 25 mm^2. These measures ensure that the inverter continues working satisfactorily and to specification in the most demanding circumstances.

Design considerations

Although most commercial in-car AF amplifiers are powered by unregulated supplies, the supply from the present inverter is regulated. A regulated power supply is a more reliable reference for protection circuits. These circuits, as well as the feedback loops, make use of optoisolaters to eliminate the risk of earth loops.

A block schematic of the inverter is shown in **Fig. 6**. Much of the circuitry is standard: an oscillator controls a switch which, together with a capacitor and an inductor, creates an alternating voltage from the 12 V battery. However, although in many commercial inverters for in-car amplifiers the alternating voltage is simply rectified and smoothed, in the present circuit it is stepped up to a value which, after rectification, gives a direct voltage of ±48 V. After this has been buffered, a part of it is fed back via an optoisolator to the oscillator, which uses it to vary the width of its output pulses accordingly. If the output voltage is low, the width is increased; if it is high, the width is reduced. Such a design is called a controlled push-pull inverter.

If a fault occurs, the protection circuits, depending on the nature of the fault, either deenergize the output relays of the amplifier or disable the oscillator.

The dashed line indicates the separation between the control/protection section and the power section. This line will also be encountered in the circuit diagram (**Fig. 7**).

Circuit description

The oscillator is formed by IC_1 (**Fig. 7**). Apart from an oscillator, this 16-pin IC provides presettable 'dead time', soft start, 5.1 V reference voltage, error amplifier, shutdown and output drivers. The 'innards' of the IC are shown in **Fig. 8**.

The sync input of the IC is not used in the present application.

The 'dead time' can be preset via pin 7 within fairly wide limits.

Only a timing capacitor is needed for the soft start operation.

The signal at the shutdown input, pin 10, has an effect on the soft start as well as on the output circuits. Short pulses at this input cause immediate shutdown via a PWM latch, while longer ones result in a renewed soft start. The same functions are fulfilled by the UV (under voltage) lockout when the input level is below normal: the outputs then remain disabled and the soft start capacitor is discharged. To prevent jitter, the lockout function has an hysteresis of about 500 mV.

If the PWM pulse is interrupted for whatever reason, the PWM latch holds the outputs in the disabled state during the remaining part of the period. The latch is reset at each and every clock pulse.

The output section consists of four NOR gates that are followed by push-pull output amplifiers which can deliver well over 200 mA.

The 12 V supply lines to IC_1 are well decoupled by R_6-R_8-C_5-C_9. The output circuit of the device has its own supply connection (pin 13), which is decoupled separately by R_6-C_9.

The oscillator frequency is determined by R_5 and C_3 and is here 110 kHz. The 'dead time' is determined

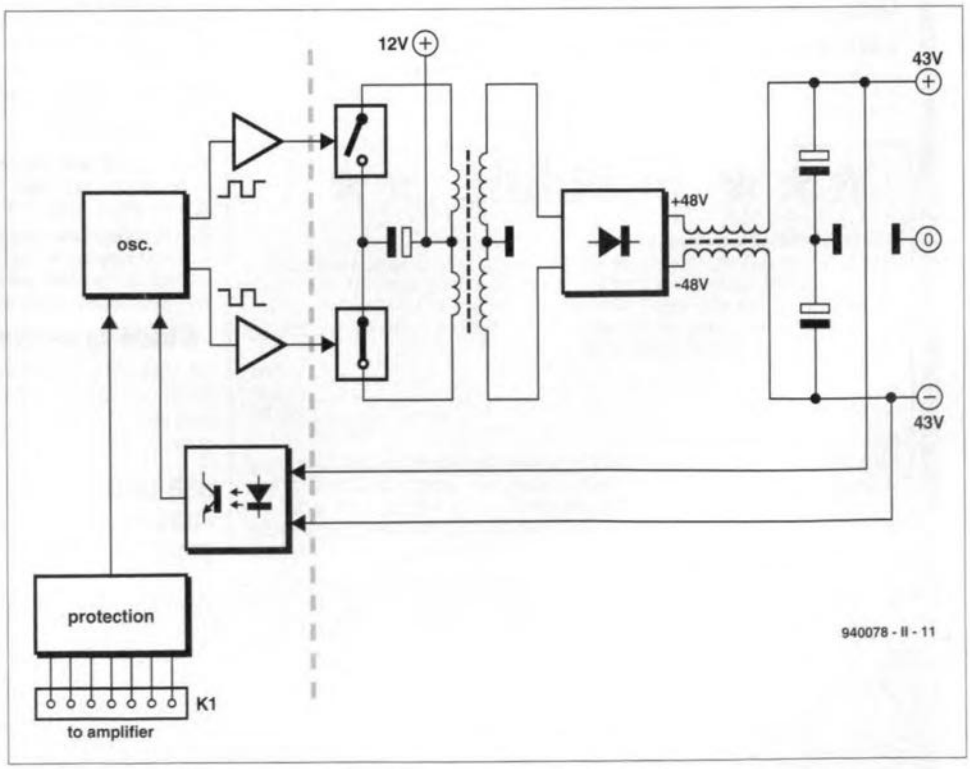


Fig. 6. Block schematic of the inverter for the in-car audio amplifier.

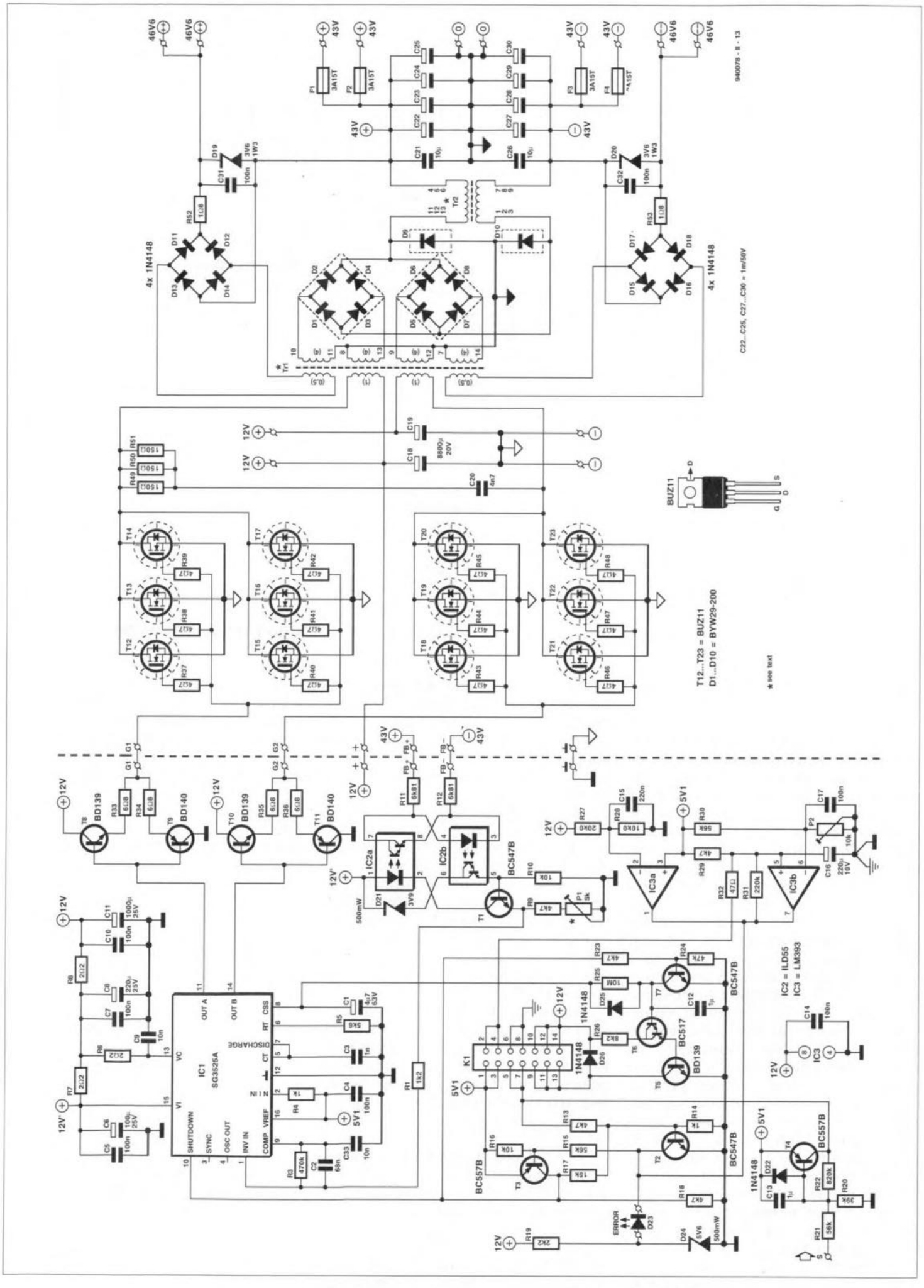


Fig. 7. Circuit diagram of the inverter for the in-car audio amplifier.

by the value of R_2; in the present circuit it has been kept to a minimum by the omission of R_2.

The duration of the soft start is determined by the value of C_1, which is also responsible for the magnitude of the start-up current of the inverter. (Strictly speaking, it is the rate at which C_1 is charged that determines the start-up current). This requires careful design, because after transformation and smoothing a capacitance of 80,000 µF must be charged to 43 V. The specified value of C_1 gives a start-up current of about 20 A.

A current of up to 20 mA may be drawn from the internal reference voltage source of IC_1, which is the reason that several voltage dividers, and also the error amplifier, are supplied by this source. Capacitors C_2 and C_{33} improve the stability; C_{33} also limits the open-loop bandwidth of the error amplifier.

If, in a symmetrical step-down inverter as used here, the feedback signal for the control circuits is fed back via an optoisolator, it can be referred to the earth of the control circuits. In practice, this is bothersome, however, because optoisolators have a very large temperature coefficient. Moreover, their current transfer ratio has an appreciable spread. All this is obviated by local feedback in the control circuits. Therefore, IC_2 is a dual type in which IC_{2a} feeds back the output current of IC_{2b}, which is amplified by T_1, to the input of IC_{2b}. Nevertheless, it may be necessary in some cases to change the value of R_9 and P_1 to some degree.

The protection circuits in the power amplifiers control the shutdown input of IC_1. When this input is actuated, the corresponding output driver is disabled instantly and C_1 is discharged by an internal current source. If the fault persists, the supply is thus switched permanently to standby; when the fault is remedied, a fresh soft start takes place.

Although the output circuits of IC_1 can provide a current of up to 200 mA, direct driving of the MOSFETs that function as power switches proved unsatisfactory. The large gate capacitance in conjunction with the limited output current delayed the switching process to an unacceptable extent. For this reason, additional buffers, T_8–T_{11}, have been provided in series with pins 11 and 14. This raises the level of the current that can be switched to an optimum value of 1.5 A. Although this level can be even higher, the consequent additional dissipation in the various components lowers the efficiency.

The alternating voltage is generated by a push-pull inverter, comprising Tr_1, C_{18} and C_{19}, and two switches consisting of MOSFETs T_{12}–T_{23}.

The transformer has a turns ratio of 4. The voltage across the primary is 24 V_{pp}. The secondary voltage is rectified by two parallel-connected bridges, D_1–D_8. The use of two bridges limits the current in each secondary (lower copper losses), halves the current peaks in the diodes and lowers the threshold voltages to some extent. All this improves the efficiency. To ensure good current division, the two bridges are thermally coupled by being fitted to the opposite sides of a heat sink.

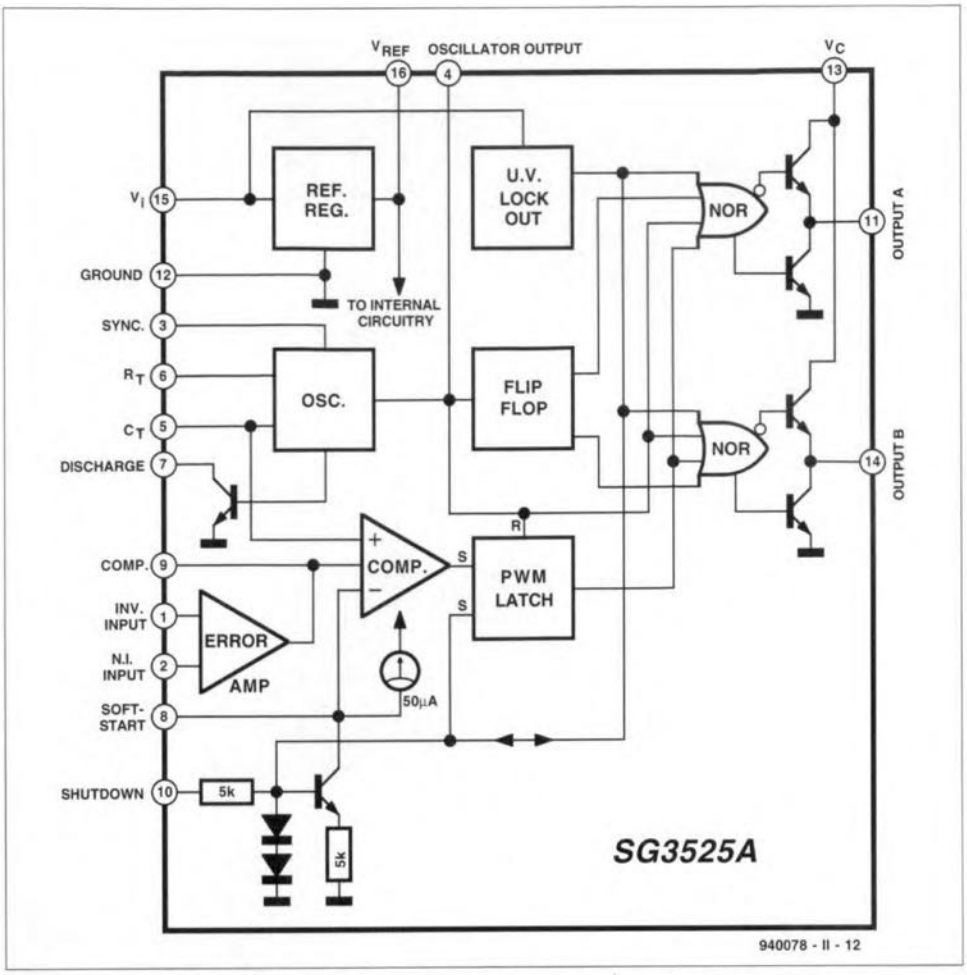


Fig. 8. Schematic diagram of SGS Thomson's SG3525A.

The symmetrical step-down converter consists at the positive side of the first secondary of Tr_2 in conjunction with D_9, and at the negative side of the second secondary and D_{10}. These diodes are mounted on the same heat sink as D_1–D_8.

Smoothing is effected in the first instance by C_{21} and C_{26}, which are polypropylene types because of their excellent HF properties. The 'real smoothing' is, however, effected by the four 10,000 µF capacitors, C_{22}–C_{25} and C_{27}–C_{30}, in parallel with C_{21} and C_{26} respectively.

All 43 V lines have their own delayed action fuse, rated at 3.15 A (F_1–F_4). This rating limits the power to the output amplifiers fairly accurately to 200 W into 4 Ω.

As stated in Part 1, the voltage amplifying stages in the amplifier operate from ±46.6 V. This is to ensure that when these supply lines vary in sync with the current amplifier, the emitter followers can not be overdriven. Because of this requirement, Tr_1 has separate secondaries that provide auxiliary voltages of 6 V. These are rectified by D_{11}–D_{14} and D_{15}–D_{18}, kept constant at 3.6 V by zener diodes D_{19} and D_{20} and then simply added to the relevant ±43 V lines. The auxiliary voltage lines are decoupled for HF by C_{31} and C_{32}.

Owing to the switching frequency of 110 kHz, the inevitable dissipation, life expectancy and the extremes of temperature occurring in cars, standard electrolytic capacitors can not be used for C_{18} and C_{19}, since these would become very hot within a very short time and would present a serious risk of exploding. Fortunately, there are special high-grade electrolytic capacitors on the market for this purpose.

The protection circuits are five-fold. Three of them are contained in the power amplifier, and the other two in the present inverter circuit.

The amplifiers are protected against too high currents, too high temperatures of the heat sinks and direct voltage. The sensors for the first two in both channmels are linked to the protection circuit via four parallel connected optoisolators (pins 1, 3, 5 and 7 of K_1). [940078-II]

SINGLE-WIRE COMMUNICATION

Although wireless communication is nothing special these days, it is not a panacea. Below the ground, for instance, radio waves will not get you very far. In such cases, nothing beats copper wire to set up a reliable communication link. The communication system described here is remarkable because it uses only one such wire. Low-cost, and ideal for a camping site intercom, or a cave explorers' talk line.

Design by K. Walraven

MOST frequencies used for wireless communication are so high that only a very small portion of the electromagnetic energy penetrates the soil, not even mentioning solid rock. Although it would just be possible to use very low frequencies in the ELF (extremely long frequency) band for underground communication (similar to submarine calling systems), the relevant antennas and transmitters are of a totally impractical size.

Speleologists (cave explorers) usually employ common-or-garden field telephones linked by a two-wire cable. Bearing in mind that such a cable can have a length of several kilometres, the cost of all this copper can be reduced significantly if a single-wire cable were used. The other 'wire' (return signal) is then formed by the soil (or 'earth', as the word is used in connection with electrical systems). Particularly below the ground, in relatively humid conditions, such an earth connection is a fine alternative to a return wire.

Although excellent for speleologists, the present single-wire communication system is also suitable as low-cost intercom above the ground (where most of you will definitely feel more comfortable).

Design considerations

In fact we are dealing with a normal intercom circuit, with a telephone mouthpiece acting as a combined microphone/loudspeaker. The connection between the (hand-held) extensions is made via one wire and a ground return path. Depending on the dryness of the top soil, the return path can have a resistance of several megaohms (dry soil) down to less than 1 kΩ (wet soil). The input resistance of the circuit and the ground return resistance form a voltage divider which attenuates the desired signal. That is why the circuit is designed to have an input impedance as high as 1 MΩ.

Any number of send/receive units ('telephones') may be connected to the line, since every circuit functions as a receiver by default. The transmitting unit is then heard on all receivers.

Circuit description

The circuit (**Fig. 1**), simple by any standard, is based on the familiar 741 opamp which most of you will be able to dig up from the junkbox. The telephone has two switches: S_2 to switch the unit on and off, and S_1 for the push-to-talk function. The circuit is normally switched to 'receive' mode. Signals transmitted by another unit, and, of course, noise, are received via capacitor C_1. Next, a low-pass filter, R_1-C_2, ensures that frequencies above 5 kHz are attenuated (the roll-of frequency is slightly dependent on the line impedance). The signal is subsequently applied to a high-pass filter consisting of C_3-R_2-R_3, in which the resistors set the receiver impedance to about 1 MΩ. This filter serves to attenuate hum on the signal.

The 'cleaned' signal is applied to the non-inverting input of IC_1. With switch S_1 set to the position shown in the circuit diagram, the opamp functions as a buffer, feeding its output signal to the telephone mouthpiece element via coupling capacitor C_4.

Two things happen when S_1 is pressed. The microphone is connected to the inverting input of the opamp via C_4. The gain is then determined by resistor R_4 and the impedance of the microphone. The mouthpiece used in the prototype has an impedance of about 100 Ω, fixing the opamp gain at about 1,000. Although the microphone impedance should really be of the order of 350 Ω, the 100-Ω type also gave good results. The amplified signal is put on to the communication line via switch contact S_{1b} and coupling capacitor C_1. The transmit level is about 3 V_{pp}. At the same time, switch contact S_{1a} takes junction R_1-D_2-C_2-C_3 to the circuit ground, preventing the non-inverting input of IC_1 from picking up the output signal (which would cause positive feedback). Also, LED D_1 lights, indicating that the unit is in 'transmit' mode, and at the same time assuring the user of sufficient battery energy. If the battery voltage drops below the level determined by the value of zener diode D_2 and the LED voltage (approx. 7 V), the LED will no longer light. In view of the tolerance on the actual zener voltage, it may be necessary to use a different zener diode than indicated. In any case, be sure to use a 400-mW type, since this guarantees a sharper 'knee' voltage than 1-watt types.

That is just about everything there is to say about the operation of the circuit. The final points concern the current consumption. In 'receive' mode,

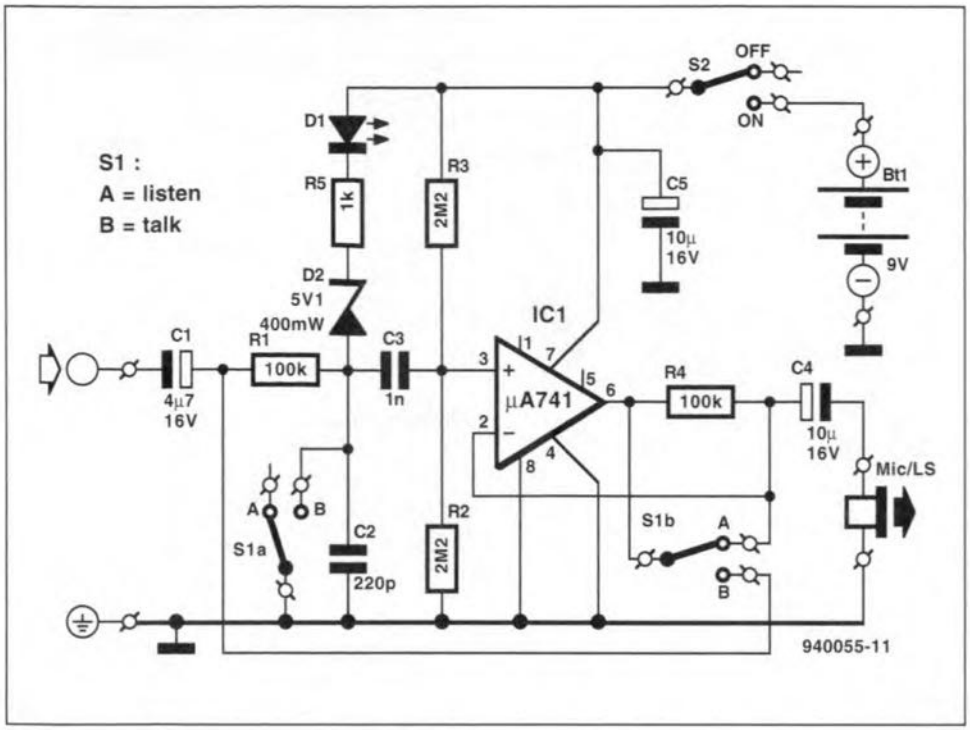


Fig. 1. Circuit diagram of the single-wire communication system. Please welcome an old faithful: the 741 opamp.

the circuit draws about 3.5 mA, which rises to about 5 mA in 'transmit' mode. In both cases, a modest current drain, which guarantees a long battery life.

The 741 may be replaced by a more modern version such as the TLC271, which reduces the current consumption even further, and also results in a slightly higher transmit level (approx. 3.5 V_{pp}). Since pin 8 of IC_1 is tied to the circuit ground, the bias current of the TLC271 is automatically set to 'high' (pin 8 is not connected on the 741).

Rugged construction

Rugged construction is a must particularly if the circuit is to be used for cave expeditions and other activities underground. The printed circuit board designed for the circuit is small (**Fig. 2**), enabling the telephone to be built into a compact, easy to handle, enclosure.

The enclosure must be a metal type (preferably aluminium or die-cast) because the earth return connection is made via the user's hand.

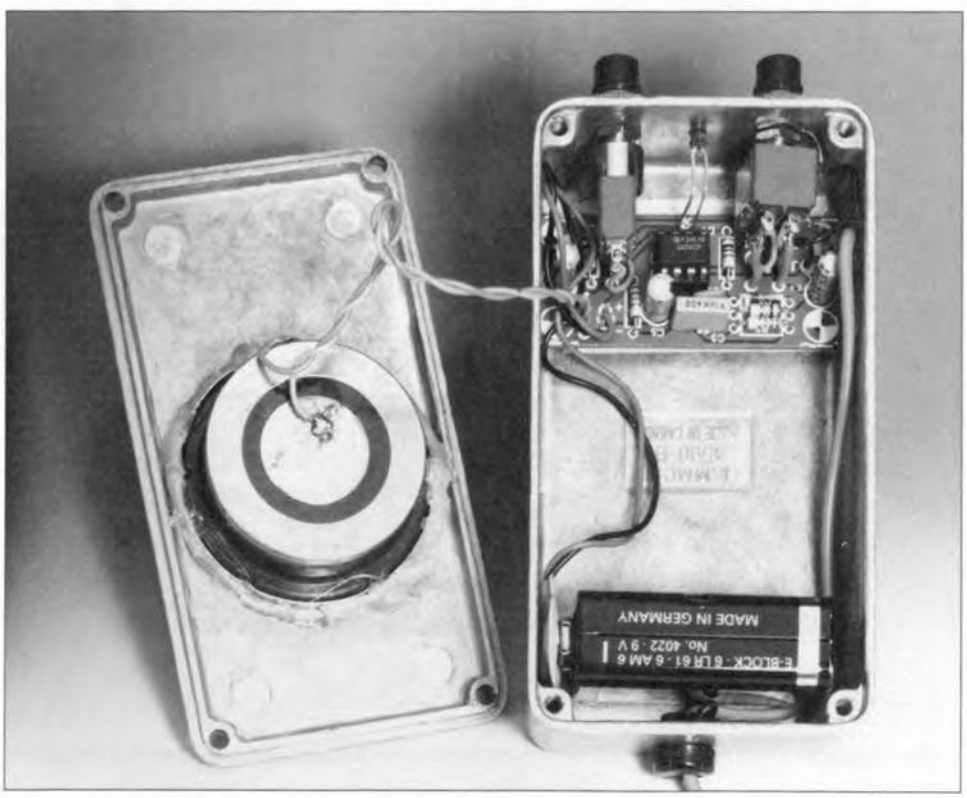
With only a handful of parts to be fitted, populating the printed circuit board is all plain sailing. Less experienced constructors should, however, pay attention to the polarity of the electrolytic capacitors, the LED and the zener diode. The single IC must also be fitted the right way around (have a good look at the symbol printed on the component overlay). Cut the wires at the solder side as short as possible.

Because of the relatively small size of the board, a separate wiring diagram is given **Fig. 3**, showing clearly how the switches, the mouthpiece element and the battery are connected to the board. Although most switches have their mother contact ('pole') in the centre, the ones used to build the prototype (see parts list) happen to have different connections. Therefore always check out the location of the pole and the contacts before connecting a switch. The PCB is designed for switches which have the 'rest' contact nearest to the edge, then come the pole and the 'make' contact. Whatever type of switches you use, make sure they are wired as shown in Fig. 3.

The following method is recommended to build the PCB into the case. Start by securing the switches on to the case, and then connect the board to the switches via a few pieces of solid wire. In this way, the board is held in position by the wires. Stick a few pieces of insulating tape at the inside of the case, below the circuit board, so that short circuits with the solder connections can not occur.

An alternative construction method is to secure the board to the case bottom with sticky rubber feet as used for enclosures. That allows flexible wire to be used for the connections to the controls.

The input ground wire is clamped under the washer of switch S1 to establish the contact with the case (without this connection, the circuit does not work). The telephone mouthpiece (acquired from an electronic surplus outlet) is secured rigidly in the cover of the case, and may be protected with water-resistant foil. Depending on the type of element, one of its terminals may already be connected to the metal body, in which case it is also connected to the cover (check with an ohmmeter). If that is the case, this connection of the element **must** be



wired separately to the output ground terminal on the board.

The unit is connected to the communication wire via a kind of probe, which is a piece of flexible wire about 1 m long. The end of this wire is fitted with a needle or a crocodile clip. To prevent the wire insulation from cracking where it enters the metal case of the telephone, and so causing a short-circuit, a rubber grommet must be used. The switches are types with a protective rubber cover, which makes them reasonably water-resistant. The rest of the unit is made waterproof by using silicon compound. Apply a little compound around the LED, and also fill the strain relief for the 'antenna' wire. Make a packing to seal the joint between the case and the cover. Beforehand, apply a little lubricant, or butter, to the edge of the case, so that the silicone compound will attach to the cover only, and does not crack when the telephone is opened to replace the battery. If the compound is still soft, it is best not to leave the cover on for a while with the screws not fully tightened. In that way, the packing remains a little thicker.

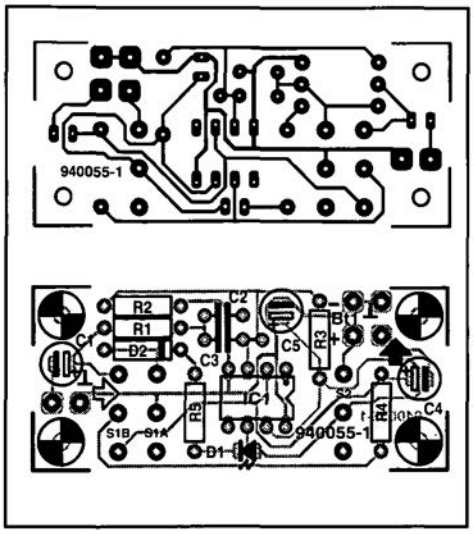


Fig. 2. The printed circuit board designed for the 'earth telephone' is small, and contains only a handful of parts.

Some practical notes

Some information is given below on the practical use of the single-wire communication system. Use insulated wire for the long cable via which the extensions communicate. Almost any number of extensions may be connected to this cable. The actual connection is made by removing the insulation locally and attaching the croc clip. An alternative method which leaves the cable almost intact is to use a needle or a pin instead of a croc clip. To connect up to the system at a certain location, pierce the cable with the pin or needle to make contact with the copper wire. In this way, the cable is damaged less.

Like mobile radios, all telephones are normally in 'receive' mode. It is not unusual for the receiver to produce noise, hum, whistling notes and even radio stations. Although difficult to avoid in such a simple communication system, any noise produced by the receiver is also useful because it is a sign that the unit is working! After pressing the 'listen/talk' key, you can start talking into the microphone, and put your message on to the cable for all other units to hear. The quality of the received signal will depend, among others, on the soil humidity, and the resistance between the unit and the ground. A user wearing rubber boots will usually transmit and receive relatively weak signals. In practice, it is recommended to touch the ground, or a wall, while transmitting. That increases the signal level on the cable considerably (both while transmitting and receiving). Obviously, the aim is to make the resistance between the metal case and 'earth' as low as possible.

The battery will last pretty long. An alkaline battery enables a telephone to be used for a couple of days at a stretch, which will be sufficient for most, if not all, applications.

(950055)

COMPONENTS LIST

Resistors:
R_1,R_4 = 100kΩ
R_2,R_3 = 2MΩ2
R_5 = 1kΩ

Capacitors:
C_1 = 4µF7 16V radial
C_2 = 220pF
C_3 = 1nF
C_4,C_5 = 10µF 16V radial

Semiconductors:
D_1 = LED, high efficiency (3 mA)
D_2 = 5.1 V/400 mW zener diode
IC_1 = 741 or TLC 271

Miscellaneous:
S_1 = presskey w. 2 change-over contacts (e.g. APEM 18545).
S_2 = switch w. make contact (e.g. APEM 8636)
2 rubber caps for switches (APEM U-1401).
LSP/MIC_1 = telephone mouthpiece, impedance: 350 Ω.
Metal case, outside dim. 112 x 61 x 32 mm (e.g. Hammond 1590B or Velleman G106).
Printed circuit board, order code 940055-1 (see page 70).

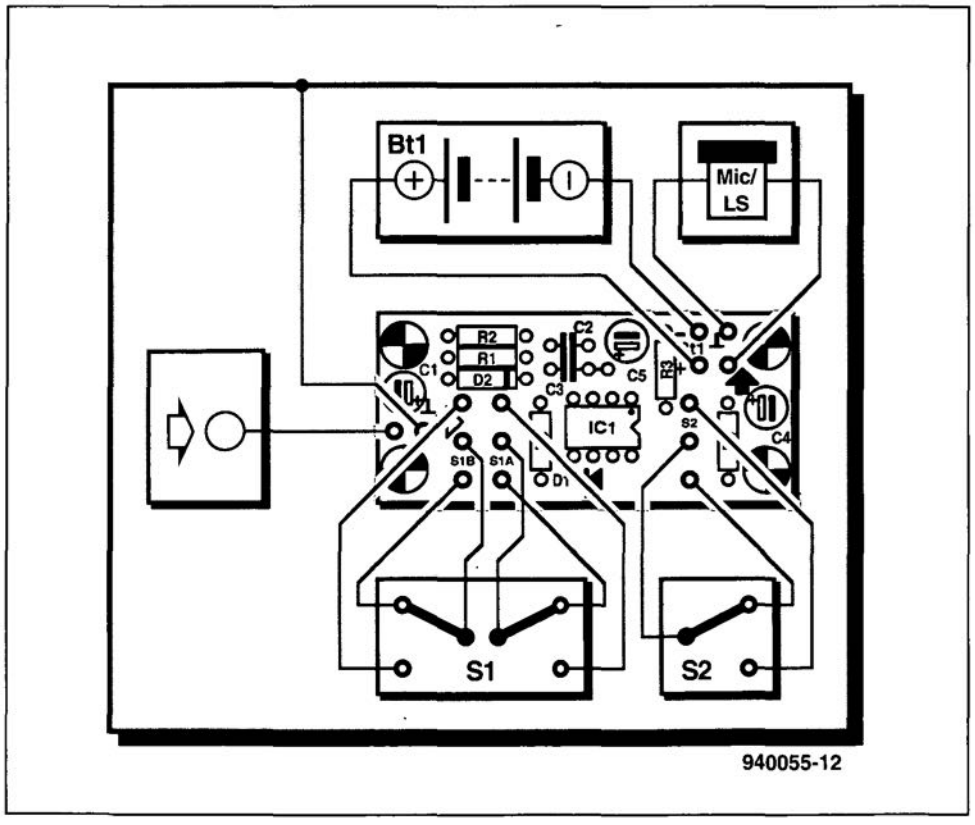


Fig. 3. Connection diagram. The wiring shown is valid for the switch types stated in the parts list.

SOLID-STATE DISK

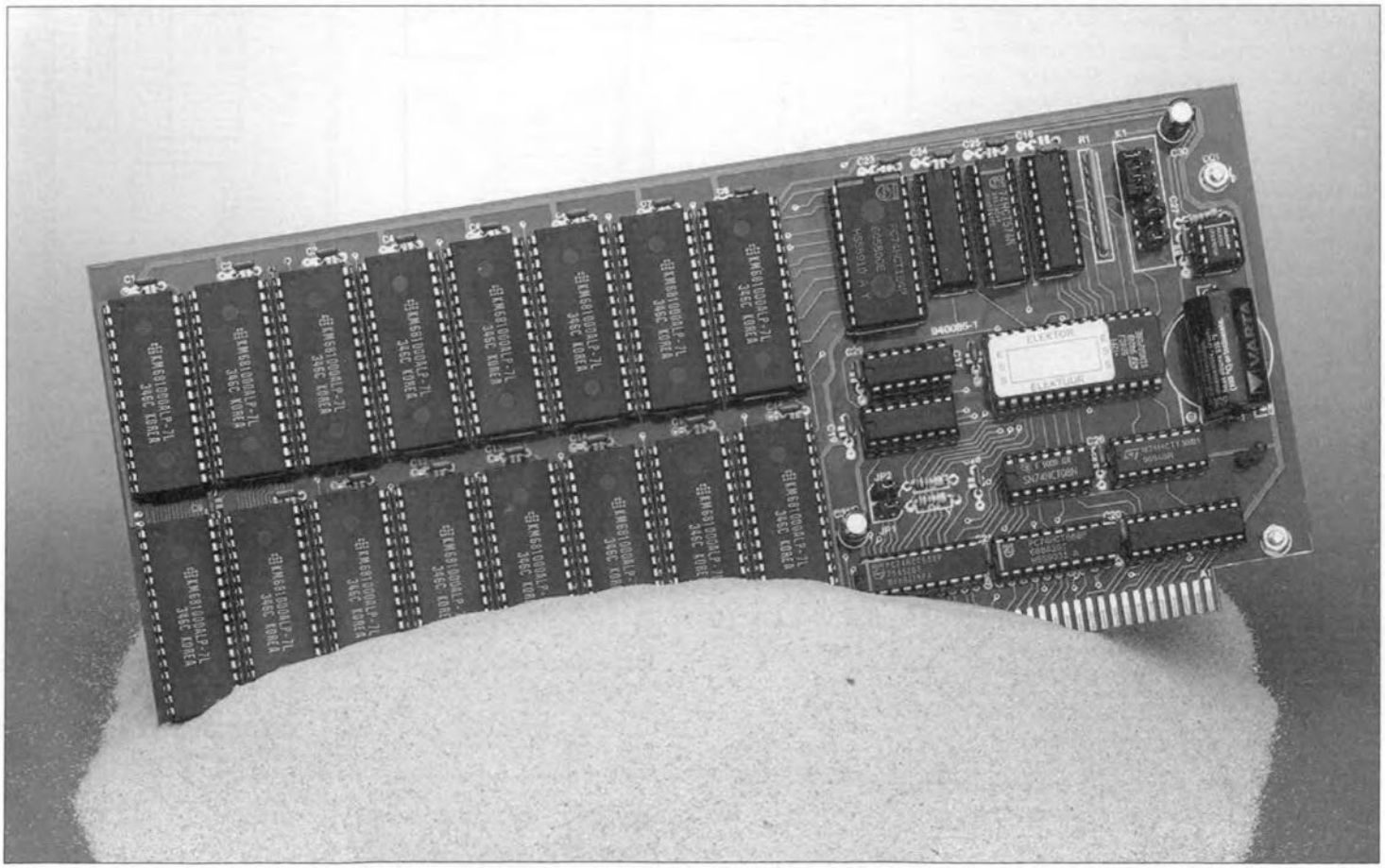
Practically all of today's PCs have a hard disk and one or more floppy disk drives. These 'media' allow the computer to load the files it needs to run a certain program. Unfortunately, there are circumstances in which the use of these magnetic/mechanical media is problematic. In cases where shock, vibration, large temperature variations or an otherwise electrically hostile environment force you to turn away from the 'usual' disk drives, the solid-state disk discussed in this article is a very good alternative.

Design by B. Yahya

ALTHOUGH much larger media are pressing at the gates, the floppy disk is still the most popular portable information carrier. A DOS formatted 3.5-inch floppy disk has a capacity of 1.44 MByte, which is, in principle, enough for the MS-DOS operating system and one or two small application programs. The main disadvantages of the floppy disk are the relatively low data transfer speed, and the risk of data corruption. Also, a diskette is easily lost or stolen. Like the drive unit of a hard disk, the floppy disk drive is a mechanical system with read/write heads and motors. Unfortunately, mechanical parts are subject to wear and tear and eventual breakdown. Furthermore, additional safety precautions have to be taken where a floppy disk drive is used in environments with a high explosion risk. The danger is caused mainly by the tiny sparks which occur in the drive motor.

MAIN SPECIFICATIONS

Application:	in any MS-DOS PC
Capacity:	max. 2 MByte
Disk emulation:	1.44 or 2.88 MByte
Memory type:	static
Data buffer:	Lithium battery
I/O addresses:	300H, 308H, 310H or 318H
BIOS:	in EPROM
Software:	not required
Hardware:	8-bit insertion card



The solid-state version of the floppy disk drive is a plug-in board containing static RAMs and memory backup battery. Arguably, it is totally free from the above disadvantages. It is quiet, fast and easily protected against corruption of data. Also, being secured inside the PC, the solid-state disk is not so easily lost or stolen.

The solid-state disk may be configured such that it can be used to 'boot' the computer, i.e., it acts as a disk from which the PC loads its start-up software. With the correct software installed, the PC starts automatically from the solid-state disk after every reset. A write protect switch on the board makes unauthorized modification of the data contained in RAM far from easy. That also adds to the reliability of the PC, enabling it to log in straight away on a network. Another application where a solid-state disk has the edge over the more traditional magneto-mechanical media is remote logging, for instance, in an automated weather or air quality monitoring system.

Circuit description

The structure of the solid-state disk is relatively simple. Static memory devices are used to create a RAM bank of 2 MByte. A section of 1.44 MByte is allocated to the RAM disk function, the rest is freely available. In principle, it is even possible to use the full 2 MByte simply by configuring the solid-state disk as a diskette with a capacity of 2.88 MByte. This works as long as you, the user, ensure that no more than 2 MByte is written to the disk. Although an error report is generated if the 2-MByte limit is exceeded, many application programs will unfortunately prove unable to handle this properly.

The circuit also contains an EPROM in which all relevant BIOS (basic input/output system) information is stored. These BIOS routines serve to simulate a mechanical diskette station. The solid-state disk is therefore called just like an 'ordinary' disk drive by interrupt routine 13H. That interrupt has three basic functions: reading a sector (02); writing a sector (03); and formatting a sector (05). Those BIOS routines that have no relevance for the solid-state disk contain only a return instruction reporting 'OK' (C=0; AH=0). This enables the computer to actually

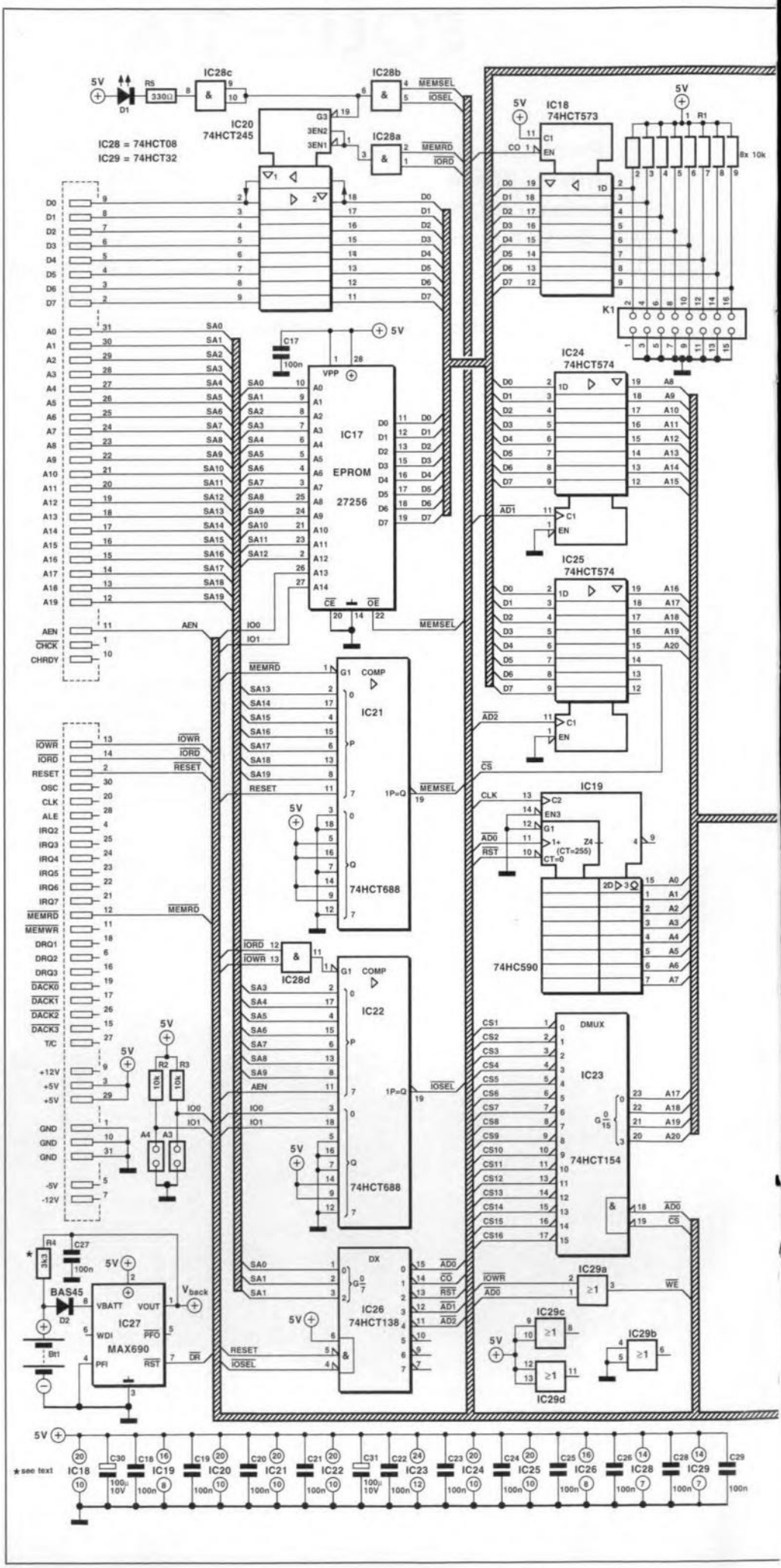
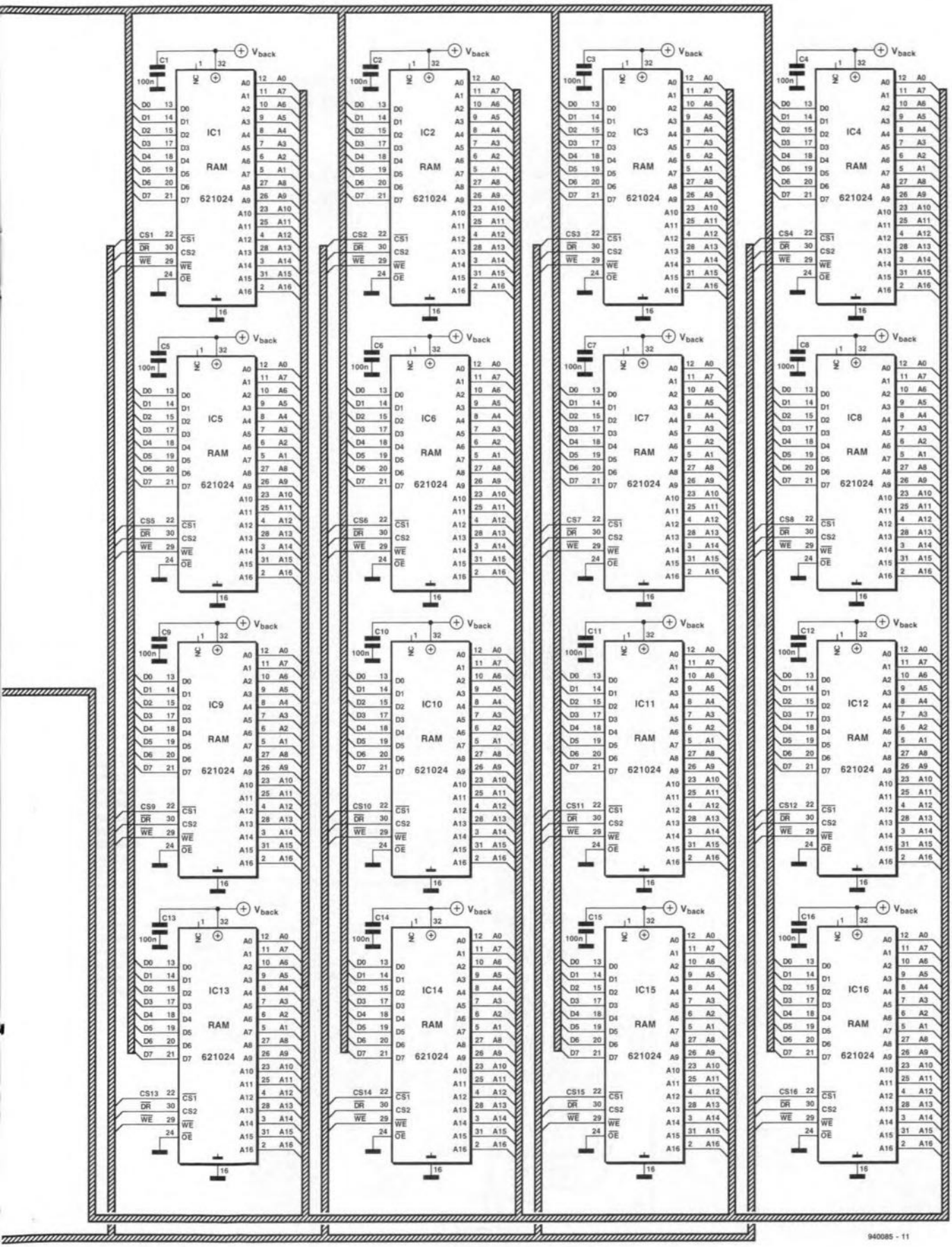


Fig. 1. Circuit diagram of the solid-state disk card. Note the huge memory bank consisting of static RAMs.


C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
C14
C15
C16
100n
V back
IC1
IC2
IC3
IC4
IC5
IC6
IC7
IC8
IC9
IC10
IC11
IC12
IC13
IC14
IC15
IC16
RAM
621024
CS1
CS2
CS3
CS4
CS5
CS6
CS7
CS8
CS9
CS10
CS11
CS12
CS13
CS14
CS15
CS16
DR
WE
OE
940085 - 11

'boot' from the solid-state disk.

The circuit diagram of the solid-state disk is given in **Fig. 1**. The striking element in the drawing is, of course, the RAM bank which consists of 16 ICs with a capacity of 128-KByte each.

With the exception of the BIOS ROM, the entire circuit is addressed in the I/O range. The base address of the card may be set to 300H, 308H, 310H or 318H with the aid of jumpers. In addition to the space occupied by the BIOS ROM, the card uses five memory locations, namely (base) through (base+4). The BIOS ROM is addressed in the memory block between D8000H through D9FFFH.

Nine ICs are used to couple to card to the PC's extension bus. Three of these, IC_{21}, IC_{22} and IC_{26}, perform the address selection. IC_{21} and IC_{22} are comparators type 74HCT688. The databus is buffered by IC_{20}, a 74HCT245. The BIOS ROM address is selected by IC_{21}, while IC_{22} and IC_{26} fix the I/O address assigned to the card.

Some tricks are in order to be able to read the memory bank whilst emulating a mechanical (floppy) disk drive. A floppy disk is usually partitioned in sectors of 256 bytes each (each of the 80 tracks on a 3.5-inch diskette has either 18 or 36 sectors). These 256 bytes are read in succession by issuing an equal number of read pulses. This system is mimicked on the solid-state disk. The 256 read pulses are used to enable IC_{19}, a synchronous 8-bit counter, to drive the lower-order eight address lines. The higher-order address lines are supplied by IC_{24} and IC_{25}.

Circuit IC_{18} has a key function in the circuit, because it reads the eight configuration bits set by the user with the aid of jumpers. Note, however, that the software set-up uses only four of these eight bits.

To address a memory location on the solid-state disk, the desired address is copied into two latches, IC_{24} and IC_{25}. Addresses (base+3) and (base+4) are available for that purpose.

Base:	read/write to/from RAM memory; increment sector counter
Base+1:	read configuration bits
Base+2:	reset sector counter
Base+3:	set A8-A15
Base+4:	set A16-A20 and CS

Table 1. Solid-state disk card addresses.

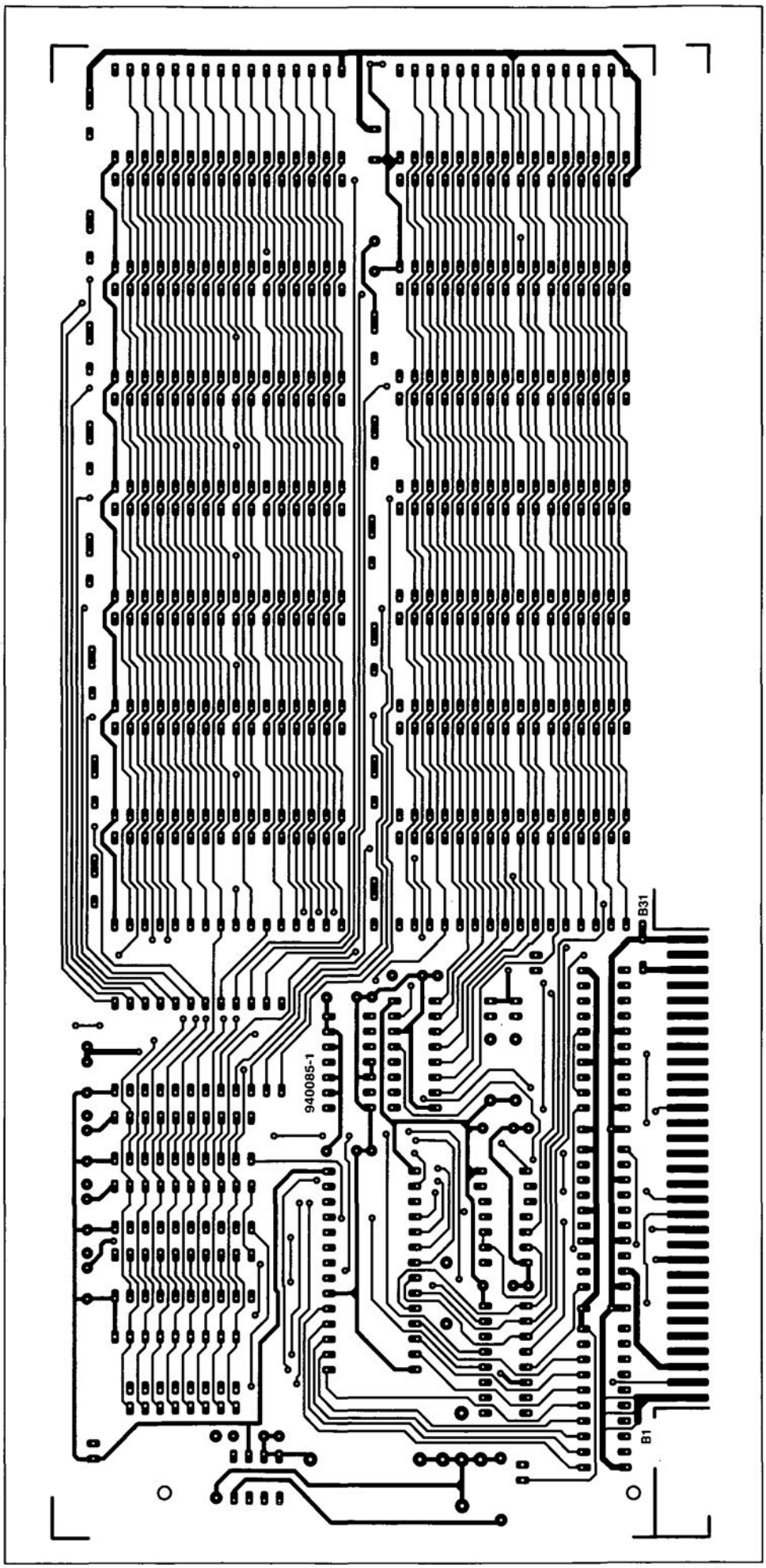
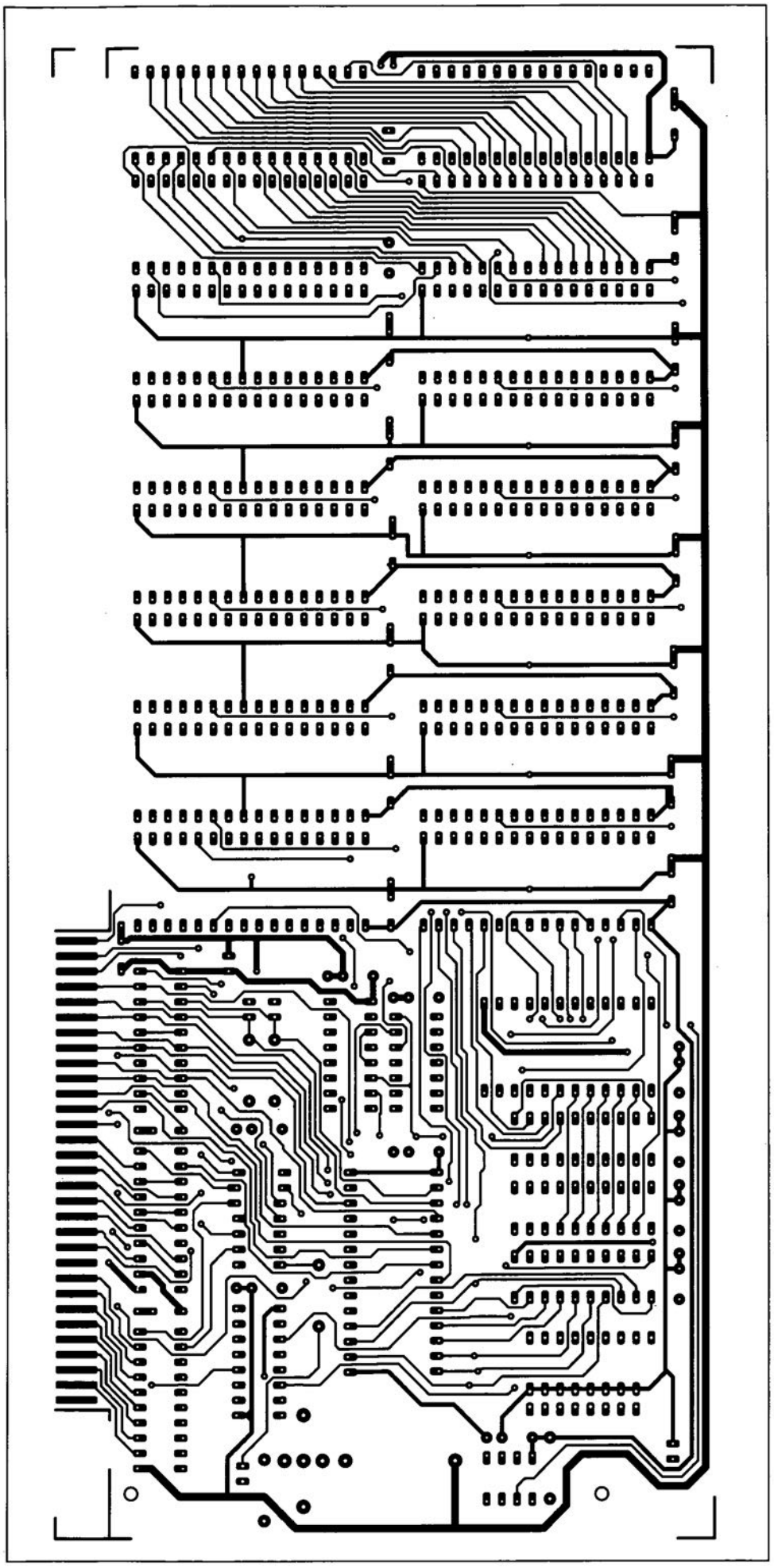


Fig. 2a. Track layouts of the double-sided, through-plated printed circuit board. This board is



Writing to address (base+2) causes 8-bit counter IC_{19} to be reset. As soon as the address is copied into the latches via the PC's I/O connection, the information from the selected sector can be requested by sending out a series of read pulses.

The individual memory components are selected via IC_{23}, a type 74HCT154. This IC generates a selection signal on the basis of the highest address lines of the memory chips.

An important function is reserved for IC_{27}, a type MAX690. This 'watchdog' IC continuously monitors the supply voltage, and ensures that the write protection is actuated via the $\overline{DR}$ signal when the supply voltage drops below a certain minimum level. The write protection then prevents the contents of the memory being changed. Next, battery Bt_1 takes over the supply of the memory via diode D_2. The specified battery guarantees a safe storage period of three years. The watchdog IC also ensures that all memory components are enabled again when the supply voltage is restored. Resistor R_4 has a function if a NiCd (Nickel-Cadmium) battery is used for Bt_1. It is omitted if a Lithium battery is used.

Finally, LED D_1 has the same function as the light on the front panel of any diskette station. The LED lights whenever the solid-state disk is selected, indicating that the computer is busy with a read or write operation on the RAM disk. If you use a high-efficiency LED in this position, you are sure not to miss even the shortest read/write pulses.

Construction

Although the circuit looks quite sizeable, construction is by no means a long-winded affair. Anyone capable of soldering accurately should be able to produce a working copy of the card. The track layouts and component mounting plan of the printed circuit board are shown in **Fig. 2**. Since the board is double-sided and through-plated, it is not recommended to make it yourself. Moreover, to prevent serious problems, the integrated connector should fit very accurately into the PC's extension bus slot. The ready-made board supplied through our Readers Services comes with gold-plated contact fingers for the best possible connection to the extension slot contacts.

In view of the cost of the memory components on the board it is recommended to use IC sockets for the RAMs and the BIOS EPROM. Start the construction by fitting these sockets. Next, do the passive parts, taking good care to observe the polarity of the electrolytic capacitors. The ICs are fitted

available ready-made, see page 70.

COMPONENTS LIST

Resistors:
R_1 = 8-way 10kΩ SIL resistor-array
R_2,R_3 = 10kΩ
R_4 = 3kΩ3
R_5 = 330Ω

Capacitors:
C_1-C_{29} = 100nF
C_{30},C_{31} = 100µF/10 V

Semiconductors:
D_1 = high-efficiency-LED, dia 3 mm
D_2 = BAS45
IC_1-IC_{16} = 621024 or KM681000 (1 Mbit static RAM)
IC_{17} = 27256 (order code 946641-1)
IC_{18} = 74HCT573
IC_{19} = 74HC590
IC_{20} = 74HCT245
IC_{21},IC_{22} = 74HCT688
IC_{23} = 74HCT154
IC_{24},IC_{25} = 74HCT574
IC_{26} = 74HCT138
IC_{27} = MAX690
IC_{28} = 74HCT08
IC_{29} = 74HCT32

Miscellaneous:
K_1 = double-row 16-pin header. Jumpers as required.
Bt_1 = ER 1/2 AA (Varta).
1 cover bracket type KHPC L22833 (Eurodis Texim).
Printed circuit board plus BIOS EPROM: order code 940085 (see page 70).

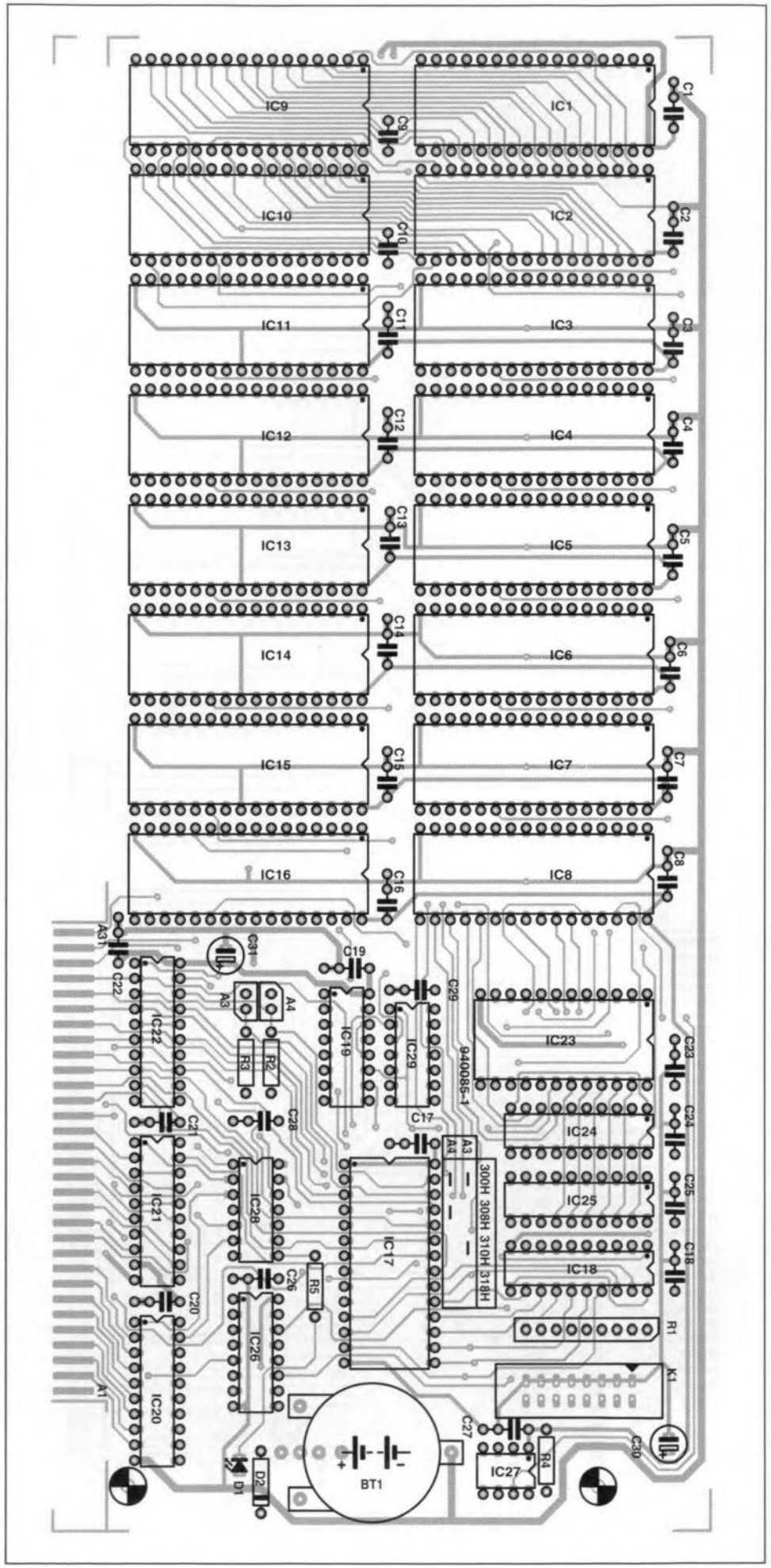


Fig. 2b. Component mounting plan.

last. Here, too, mind the polarity, because errors are easier made than found, paid for and corrected (in that order). Next, check for errors in your soldering work. If you are convinced that everything is in order, the mounting bracket may be fitted on to the PCB.

Setting up

The functions of the jumper options offered by K_1 are listed in **Table 2**. The default base address of the card (no jumpers fitted) is 318H. That address range is reserved for experimental circuits, and is usually free. If 318H is already occupied by another extension card in your system, relocate the solid-state disk to an alternative base address using the A3 and A4 jumpers. Alternatives are indicated on the component overlay printed on the circuit

COMPONENTS LIST

Resistors:
R_1 = 8-way 10kΩ SIL resistor-array
R_2,R_3 = 10kΩ
R_4 = 3kΩ3
R_5 = 330Ω

Capacitors:
C_1-C_{29} = 100nF
C_{30},C_{31} = 100µF/10 V

Semiconductors:
D_1 = high-efficiency-LED, dia 3 mm
D_2 = BAS45
IC_1-IC_{16} = 621024 or KM681000 (1 Mbit static RAM)
IC_{17} = 27256 (order code 946641-1)
IC_{18} = 74HCT573
IC_{19} = 74HC590
IC_{20} = 74HCT245
IC_{21},IC_{22} = 74HCT688
IC_{23} = 74HCT154
IC_{24},IC_{25} = 74HCT574
IC_{26} = 74HCT138
IC_{27} = MAX690
IC_{28} = 74HCT08
IC_{29} = 74HCT32

Miscellaneous:
K_1 = double-row 16-pin header.
Jumpers as required.
Bt_1 = ER 1/2 AA (Varta).
1 cover bracket type KHPC L22833 (Eurodis Texim).
Printed circuit board plus BIOS EPROM: order code 940085 (see page 70).

last. Here, too, mind the polarity, because errors are easier made than found, paid for and corrected (in that order). Next, check for errors in your soldering work. If you are convinced that everything is in order, the mounting bracket may be fitted on to the PCB.

Setting up

The functions of the jumper options offered by K_1 are listed in **Table 2**. The default base address of the card (no jumpers fitted) is 318H. That address range is reserved for experimental circuits, and is usually free. If 318H is already occupied by another extension card in your system, relocate the solid-state disk to an alternative base address using the A3 and A4 jumpers. Alternatives are indicated on the component overlay printed on the circuit

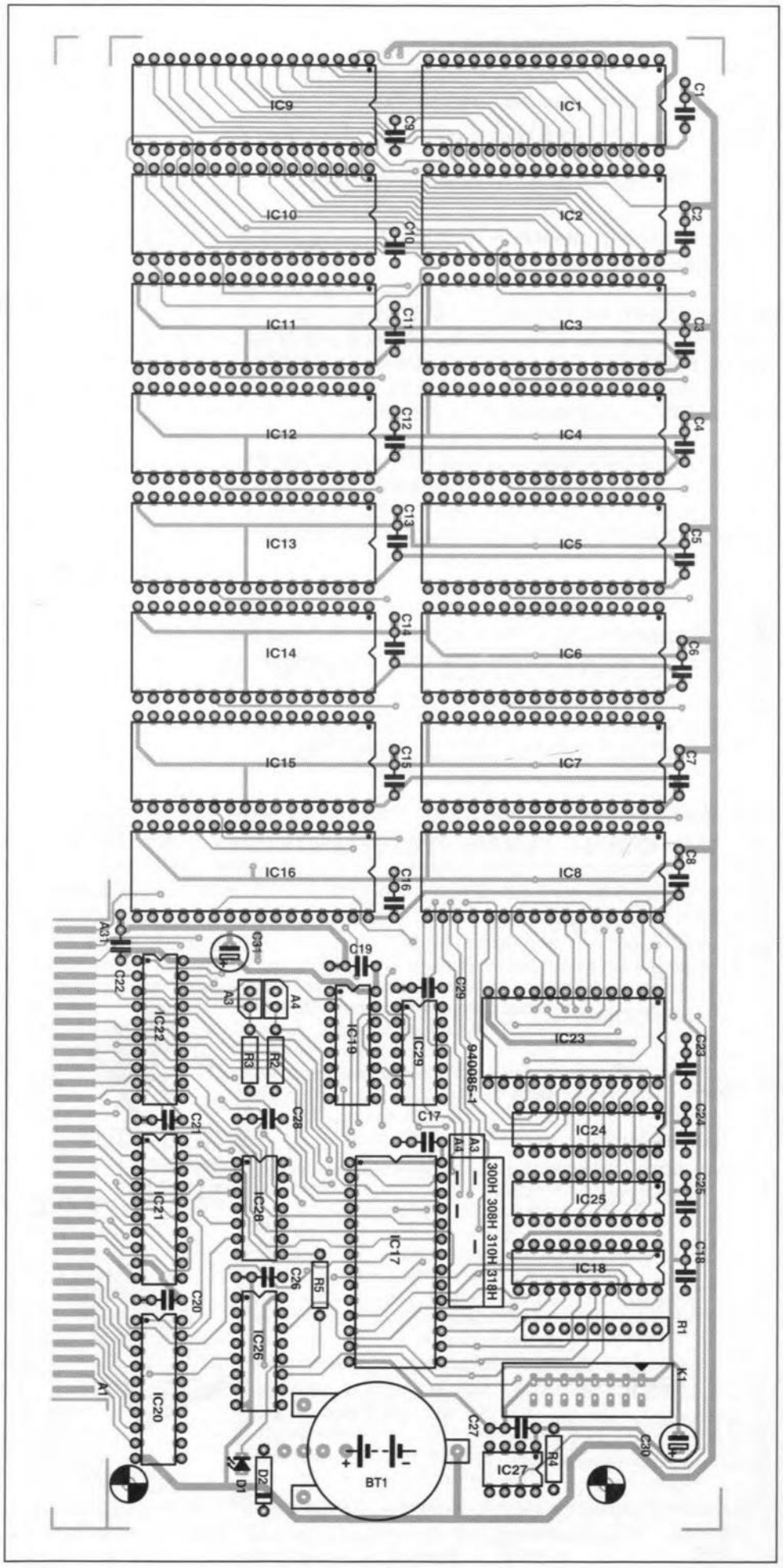


Fig. 2b. Component mounting plan.

FIGURING IT OUT

PART 20 – USING A COMPUTER

By Owen Bishop

This series is intended to help you with the quantitative aspects of electronic design: predicting currents, voltage, waveforms, and other aspects of the behaviour of circuits. Our aim is to provide more than just a collection of rule-of-thumb formulas. We will explain the underlying electronic theory and, whenever appropriate, render some insights into the mathematics involved.

The circuits that we have analysed in this series have all been simple ones, usually consisting of no more than four components. The reason for this is to illustrate the principles and mathematical techniques as clearly as posible. Analysing a circuit of the complexity of, for example, a radio receiver is just a matter of breaking its circuit down into self-contained sections and steadily working out the characteristics of each part. No new principles or techniques are required, but there is no denying that an analysis of even a fairly basic circuit can soon fill several sheets of paper. As the calculations get longer, the risk of arithmetical errors increases – not to mention the boredom that may set in at a very early stage!

During the 20 months that this series has been running, the pages of this magazine have shown a marked increase in the number of advertisements for simulation software. No longer is it necessary to cover sheet upon sheet with close-packed numbers. No longer is it necessary to wield a soldering iron, or switch on an oscilloscope, to investigate a circuit that defies analysis. A circuit can be assembled on the screen of your computer and then subjected to a wide range of tests to determine its characteristics. One such simulator program is **SpiceAge**© for **Windows**™. In this final part of the series we look at what this software can do. First of all, to illustrate the principles involved, we shall try it out on some of the circuits that we have analysed in previous months. Then we shall put it through its paces on something more demanding. We shall also look at an alternative approach, using the mathematical package **Mathematica**®, and some of its associated electronic software.

```
V volts-source    –output: gnd    +output:pos      v=0.000000    Ex=none    Of=12.000000
I amps-source     –output: gnd    +output:output   v=0.000000    Ex=none    Of=5.000000
R  R1             p1:pos          p2:output        v=3.000000
R  R2             p1:output       p2:gnd           v=2.000000
```

Fig. 159

[27.00C]

Component name	amps	Component name	amps	Component name	amps
V volts-source	399.9992m	I amps-source	5.000000	R R1	399.9992m
R R2	5.3999992				

Fig. 160

[27.00C]

Component name	amps	Component name	amps	Component name	amps
V U2	433.4301m	V U3	–306.211m	V U5	649.4064
R R1	38.46074m	R R2A	267.7502m	R R2B	433.4301
R R3	–165.680m	R R4	–649.406m	R R5	–127.219m
R R7	343.1954m				

Fig. 161

Spice

SPICE is a set of rules or a code for specifying electrical or electronic circuits. SPICE has been in existence for a number of years and is used by several simulation programs. **SpiceAge**© for **Windows**™ (or **SAW**, as we shall call it) is based on an updated variant of SPICE, though it will also accept specifications written in SPICE.

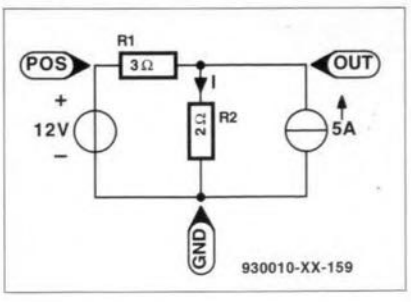


Fig. 162

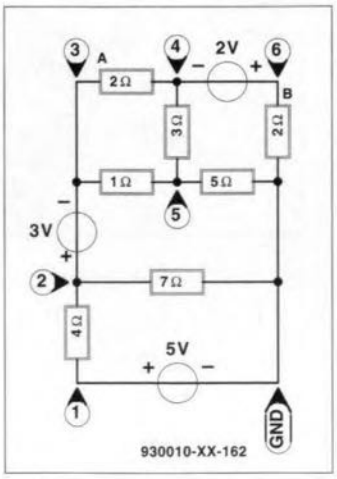


Fig. 163

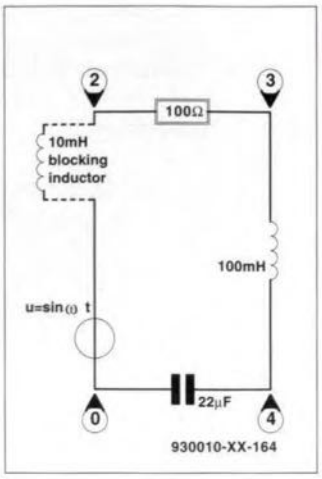


Fig. 164

V source	–output: 0	+output:1	v=1.000000	Ex=sine	Fr=500.0000
R res	p1:2	p2:3	v=100.0000		
L ind	p1:3	p2:4	v=100.0000m		
C cap	p1:4	p2:0	v=22.0000u		
L block	p1:1	p2:2	v=10.0000M	Blocking inductor to prevent low impedance of source swamping results	

Fig. 165

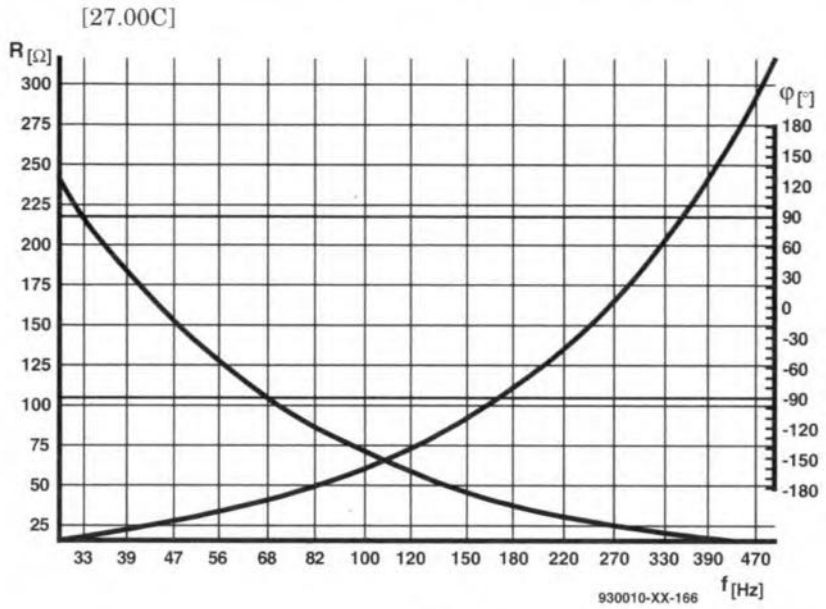


Fig. 166

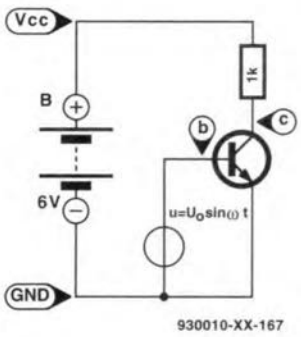


Fig. 167

The procedure is straightforward. The circuit is specified by a **netlist**, which consists of a list of the components, their parameters, and the circuit nodes to which they are connected. Then we are able to run a wide range of tests on it and to display the results as tables or graphs. As an elementary example, take the circuit of **Fig. 162**, which is derived from **Fig. 21** in Part 3. We have given names to the nodes of the circuit, though SAW would be equally content with numbers.

Components are specified in any order. In the netlist (**Fig. 159**), R_2 is on the bottom line of the list. The letter 'R' specifies the component type – resistor. Next comes its designation, 'R_2', to distinguish it from other resistors. The code 'p1:output' tells SAW that pin 1 of the resistor is connected to the node named 'output'. Code 'p2:gnd' shows that pin 2 is connected to the node named 'ground". With a non-polarised device such as a resistor, it does not matter which pin is called p1, and which is p2. The final entry on this line is 'v=2.000000'. This is the **value**, which in this case is 2 Ω. There is no need to specify the unit, but we use the suffix k for kilohms, M for megohms, and so on. No other parameters are imperative, but we could include tolerance, temperature and temperature coefficient, if these were to be taken into account in the analysis. R_1 is specified in the line above as a 3 Ω resistor connected between 'pos' and 'gnd'.

The voltage source is to deliver a constant 12 V. The line begins with V to specify a voltage source. Its negative output terminal, –out, is connected to 'gnd' and its positive terminal, +out, goes to 'pos'. 'Ex' stands for 'excitation'. This specifies the waveform, whether it is a sine wave, a step, a ramp or some other recognisable function, or random noise. Here we give it no excitation, but simply make its offset 'Of' equal to 12 V. Similarly, the current generator, specified by 'I', has no excitation, but has an offset of 5 A.

B Vcc	negative:gnd	positive:vcc	v=6.000000				
R load	p1:vcc	p2:c	v=1.000000k	%=10.00			
V source	–output:gnd	+output:b	v=10.0000m	Ex=sine		Fr=1.00000k	Of=610.000m
>TR1	npn.lib	base:b	collector:c	emitter:gnd			

Fig. 168

Writing out a netlist may be unfamiliar at first but the conventions are easy to learn. An alternative method for inputting a circuit specification is to create a schematic diagram of the circuit on the screen, using a schematic capture program that can automatically produce a compatible netlist. SAW has a sister program, **GESECA**, which produces SpiceAge© netlists from schematics.

Analysis

Once the netlist has been entered, it is saved to disc and then subjected to a variety of analyses. We wanted to check on the calculation done in Part 3, which was to find the current flowing through R_2. It is possible to 'attach' up to four probes to the circuit. Often these are used to measure the voltages between various pairs of nodes. They an also be used for measuring the current through any specified component. We set a probe to measure the current through R_2, which automatically causes the program to calculate currents through all components. The resulting table of the quiescent state of the circuit is shown in **Fig. 160**. It shows that the current through R_2 is 5.3999992 A. Our result in Part 3 was 5.4 A, using fewer significant figures. Note the much smaller current (approximately 400 mA) through R_1, also listed as flowing through the voltage source with which it is in series. The analysis is done in a fraction of a second, and no mistakes are made; already we are beginning to appreciate the benefits of the computer. Above the table, '27.00C' indicates that the analysis is performed with an ambient temperature of 27 °C. If tempcos had been specified for the resistors, we could investigate the circuit performance at any other temperature.

Dodging determinants

Figure 163 shows a resistor network with four meshes, requiring four simultaneous equations and a fourth-order determinant to solve it, as explained in Part 4. The netlist has specifications similar to those of **Fig. 159**, except that it lists three voltage sources and seven resistors. The quies-

cent analyis **Fig. 161** shows the currents through all components. The circuit was one of the 'Test Yourself' questions of Part 4. The task was to find the current through the 3 Ω resistor. For convenience, we have named the resistors by the value of their resistances. As can be seen in the printout of the analysis, **Fig. 161**, the current through R_3 is −165.680 mA. The negative sign shows that it flows from pin 2 (node 5) to pin 1 (node 4). This agrees with the result found in Part 4.

If we run an analysis of power dissipation, we obtain a table similar to **Fig. 161**, but quoting power instead of current. The analysis shows that the highest dissipation is 1.69 W, in the 4 Ω resistor.

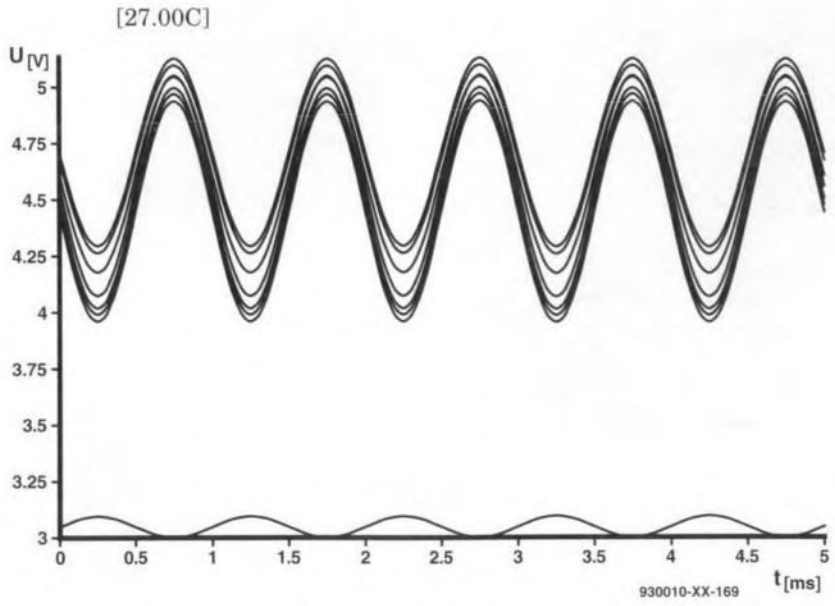


Fig. 169

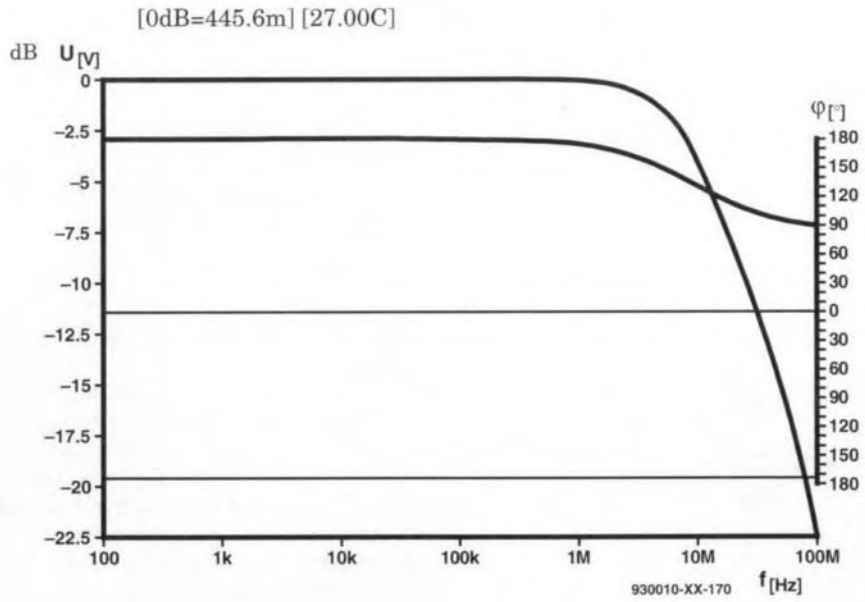


Fig. 170

Reactance

The software really comes into its own when the circuit contains reactive components and is excited by an alternating signal. In **Fig. 164** we see a basic *RLC* circuit, which first appeared as **Fig. 45** in Part 6. We were asked to find the total impedance of the circuit and the phase angle. **Figure 165** shows the netlist; note that it includes a massive 10 megahenries blocking inductor (not present in the original circuit, see later) in series with the voltage source. The voltage source has sine-wave excitation, amplitude 1 V. For calculating inductance, the software replaces the source by a small resistance equivalent to its internal resistance and applies an alternating signal of small amplitude to the nodes on either side of each component, in order to calculate the impedance and phase angle.

For this circuit, we set up a 'spot impedance' test, which finds the impedances at a selected frequency, in this case, 500 kHz. We sited the probes to measure the impedances between the pairs of nodes to which the resistor, capacitor and the inductor were connected. The readout gave the following results: nodes 2–3 (resistor): 100 Ω, Φ = 0°; nodes 3–4 (inductor): 314.159 Ω, Φ = 90°; nodes 4–0 (capacitor): 14.4686 Ω, Φ = −90°; nodes 0–2 (total of *RLC*): 329.691 Ω, Φ = 72.34°. These results are in accord with the results calculated in Part 6. The reason for adding the blocking inductor to the circuit is that the voltage source is virtually a short-circuit during the analysis. Without the large inductor, the inductance between 0 and 2 is almost zero; in effect, the software looks into the low-impedance source. The blocking inductor makes it look the other way: into the *RLC* in series.

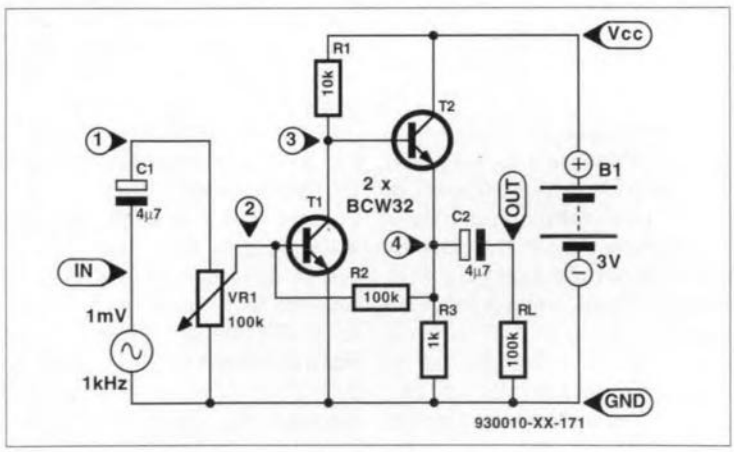


Fig. 171

Figure 166 shows another analysis performed on this circuit. Here, the frequency is swept over the range 30 Hz to 500 Hz and the graph shows the response of the circuit at all frequencies in this range. We selected this range to illustrate how the impedance of the capacitor decreases (the curve falling from the left) and that of the inductor increases (the curve rising to the right), as frequency increases. The graphs of phase angle are horizontal lines. The scale on the right shows phase to be a steady −90° for the capacitor and +90° for the inductor. This analysis took just under 10 s with a 486 running at 25 MHz;

it could be even faster if a maths coprocessor was installed.

Transistor amplifiers

A very basic amplifier circuit is shown in **Fig. 167**, derived from the circuit of **Fig. 101a** of Part 11. **Figure 168** is its net-list. For the moment, we will ignore the entries '%=10' and 0f=610.000m', which were not there when the first simulations were run. A transient analysis with the probes on b-gnd and c-gnd (corresponding to the input and output of this circuit) showed the input as a sine wave of amplitude 10 mV, and frequency 1 kHz. The output was an almost square wave, falling rapidly to 78 mV as the input rose above approximately 0.7 V and rising to 6 V when the input fell below 0.7 V. The transistor was being driven alternately between non-conduction and saturation, acting as a switch. Entering '0f=0.610V' on the netlist gives the source an offset of 0.61 V which biases the transistor into its ampifying range. **Figure 169** shows the result of the transient analysis, with the input (lower curve) plotted on a ×10 scale. The output is sinusoidal, with a phase angle of 180°; in other words, this is an inverting amplifier. Here we have in fact run ten analyses simultaneously with ten slightly different values for the load resistor. This is achieved by specifying the tolerance of the resistor by the statement '%=10'. We then instructed SAW to select ten random values for the resistor within the range 1000 Ω ±100 Ω and produce a transient for each value. This result shows how the performance of the amplifier would vary subject to tolerance variations in the value of the load resistor. It shows that although the amplification varies, the output remains sinusoidal. We are also able to give the resistor a temperature coefficient and investigate how the amplifier behaves when swept over a specified range of temperatures. In **Fig. 170**, the frequency response of the amplifier is plotted on a decibel scale (we could alternatively call for a linear scale) for the range of input frequencies from 100 Hz to 100 MHz. The two horizontal lines are the input amplitude and phase. The steeply sloping line shows how output amplitude falls off above about 3 MHz. At the same point, the phase angle begins to fall from 180° to 90°.

The transistor model used in this simulation is a 'general purpose' n-p-n transistor. For greater precision, we can make use of a library of more complex netlists based on the characteristics of production types. SAW includes a library of ZETEX transistors and it is also possible for the user to write a netlist of transistors of other types, based on data sheets provided by the manu-

B batt	negative:gnd	positive:vcc	v=3.000000		
CC1	p1:in	p2:1	v=4.70000u	%=20.00	
CC2	p1:4	p2:out	v=4.70000u	%=20.00	
RR1	p1:vcc	p2:3	v=10.0000k	%=2.00	
RR2	p1:4	p2:2	v=100.000k	%=2.00	
RR3	p1:4	p2:gnd	v=1.00000k	%=2.00	
RR4A	p1:1	p2:2	v=10.0000k		
RR4B	p1:2	p2:gnd	v=90.0000k		
RRload	p1:out	p2:gnd	v=100.000k	%=2.00	
Vsource	–output:gnd	+output:in	v=1.00000m	Ex=sine	Fr=1.00000k
>TR1	BCW32	collector:3	base:2	emitter:gnd	
>TR2	BCW32	collector:vcc	base:3	emitter:4	

Fig. 172

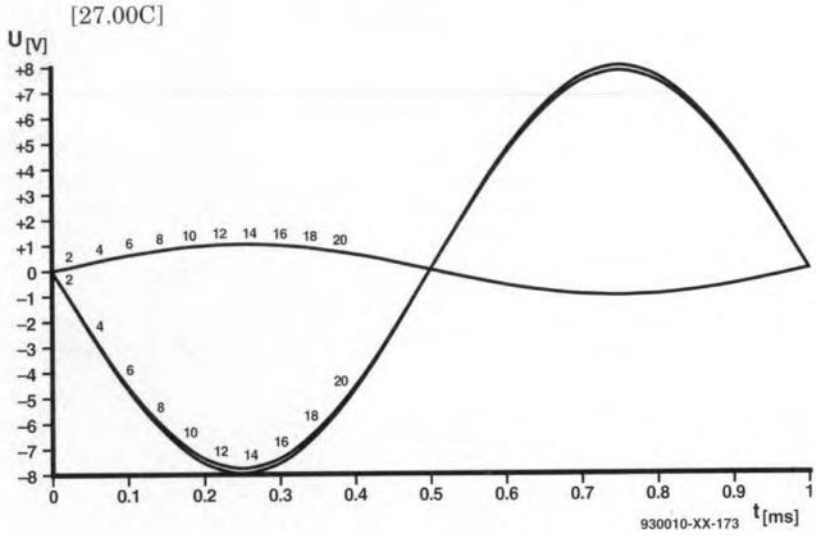


Fig. 173

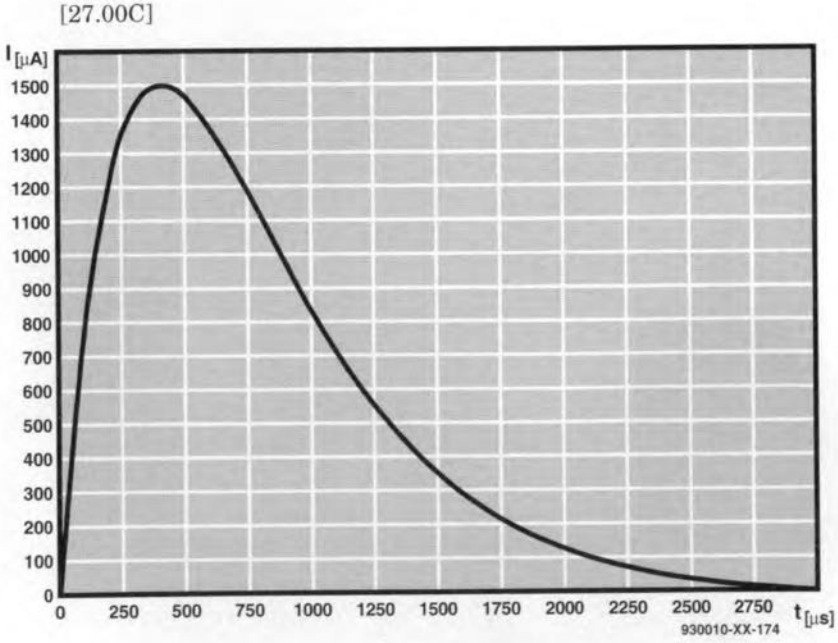


Fig. 174

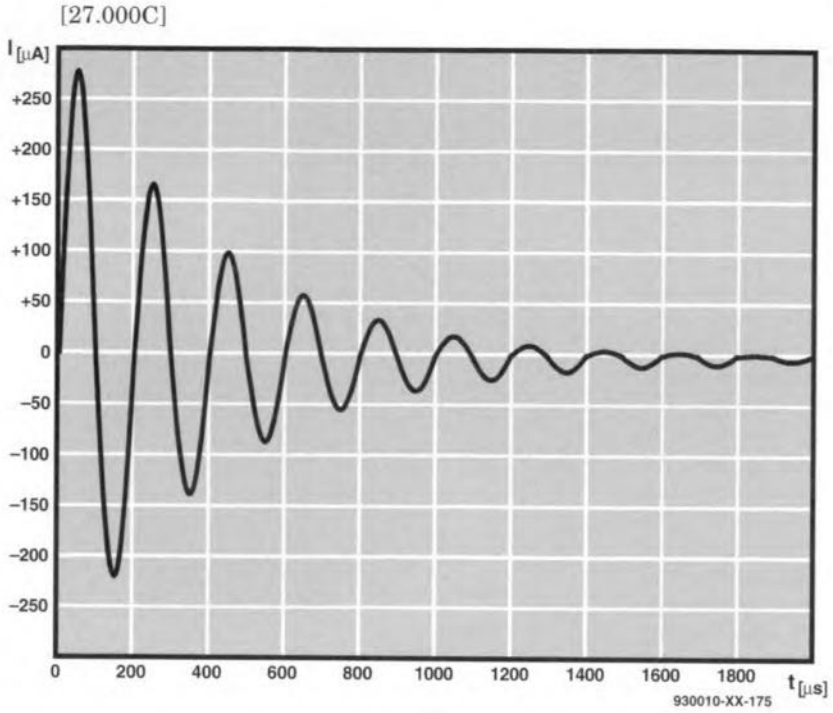


Fig. 175

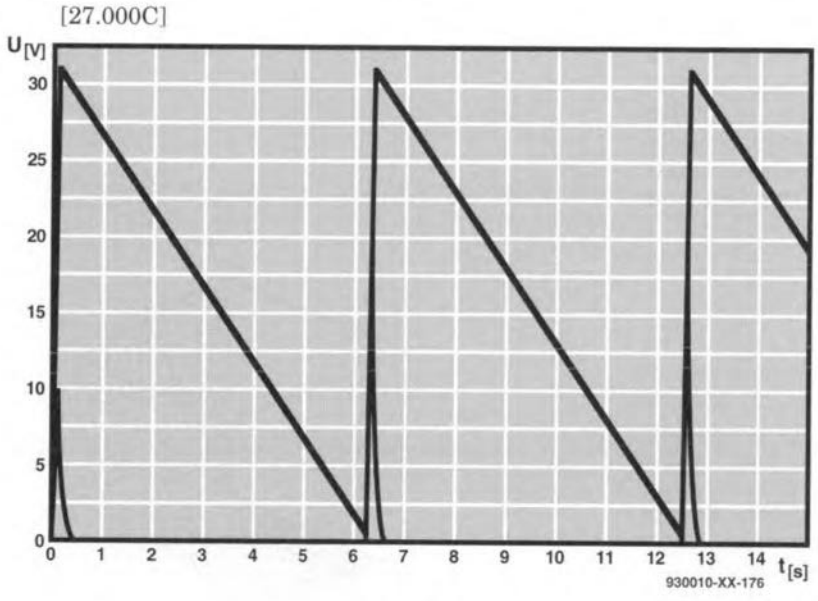


Fig. 176

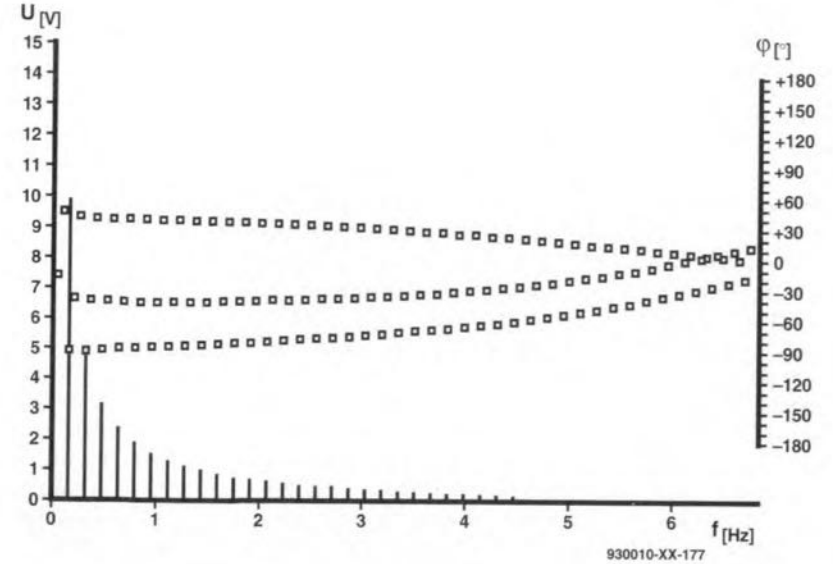


Fig. 177

facturers. Using these more complex models gives maximum precision to the simulation, but remember that the more complex the netlist, the longer it takes to perform an analysis.

Figure 171 shows an amplifier taken from *SMT Projects*[1], page 93. This is described in the book as a low-voltage a.f. amplifier with a voltage gain of 30. Keying in the netlist (**Fig. 172**) from the schematic in the book took no more than five minutes. Note the 20% tolerance of the electrolytic capacitors and the 2% tolerance of the resistors. The 100 kΩ potentiometer, VR_1, appears in the netlist as two resistors in series, R_{4A} and R_{4B}, with values set to 10 kΩ and 90 kΩ. We can edit this pair of values when necessary to simulate adjustment of VR_1. The source amplitude is 10 mV, which is within the range quoted for this amplifier. The interesting point is that the transistors are models of the BCW32, which is a surface-mount type. One of the problems of designing in SMT is that it is not possible to use a conventional breadboard for bench testing: the components are just too small and fiddly. The only course is to use conventional components and trust that the design will work when realised in SMDs. With simulators such as SAW, the design takes place on the computer and the miniscule size of the components is of no consequence.

We decided to check up on the performance of this circuit. The transient analysis—see **Fig. 173**—shows its output response to one cycle of the source and, as expected, this is 180° out of phase. The thickening of the line of the output is the result of sweeping the components through their tolerance ranges. We stipulated 20 runs, with the values being selected at random. The graph shows that variations due to tolerance make virtually no difference to the output. The small numerals beside the curves indicate which of the 20 runs corresponds to each curve, but the curves are so close together (fortunately) that it is not possible to say which is which. Instead of random selection, values can be equally

spaced over a given range. For both methods of selection, the scale may be linear, logarithmic or antilogarithmic. In a second analysis, we edited the netlist to increase the resistor tolerances to 10%; the amplifier continued to perform perfectly, so it would be reasonable to specify only 10% tolerance for the resistors.

The gain of the amplifier as read from **Fig. 173** is approximately ×8, which is rather low when compared with a gain of ×30 quoted in the book. However, this is partly because of using only medium-gain transistors. The netlist specifies BCW32 transistors which have an h_{fe} of 400. The BCF33 transistors in the original design have a gain of 800. But, by making R_{4A} equal to 1 kΩ and R_{4B} equal to 99 kΩ, we were able to increase the amplifier gain to ×50. By setting the value of the battery to various levels, we found that gain depends on V_{cc}. With two NiCd cells giving 2.6 V, the gain falls to ×4. With a single alkaline cell at 1.5 V, the gain is only ×1.6, so it qualifies as a low-voltage amplifier, but with low gain. With V_{cc} reduced to 1.3 V (a single NiCd cell), response begins to depart from the sine curve and there is no gain.

The original description quoted the power consumption as 0.6 mA, which is an average value determined by measurements taken on the prototype. We set up a probe to monitor the current through the battery and ran a frequency analysis. This showed current rising with frequency over the audio range; it was 0.6 mA at 7 kHz, and 1.4 mA at 10 kHz. We also measured output over a range of frequencies and found that output was level up to 50 kHz, above which it began to fall. The –3 dB frequency was 607 kHz. These 'measurements', each obtained in a few seconds, show how useful this kind of software is to the designer and experimenter.

Transients

In previous examples we have used the transient routine to analyse a circuit driven by an alternating source. The routine is also valuable for finding out what happens in a circuit when the input voltage changes rapidly from one steady value to another. An example of this type of calculation appeared in Part 13, **Fig. 110**, which was an *RLC* circuit similar to **Fig. 164**. We entered the netlist (omitting the blocking inductor) with R = 100 Ω, C = 1.6 µF and L = 100 mH. The source was 1 V, switched to zero at t = 0. **Figure 174** shows the current through the resistor. As in the worked example in Part 13, **Fig. 113**, there is a surge of current which dies away in about 2.75 ms, with what is close to critical damping. Changing C to 10 nF and repeating the analysis, we find that the circuit is underdamped. The current alternates with decreasing amplitude, eventually becoming zero (**Fig. 175**).

Fourier

There are many other facilities in SAW for which we do not have space to describe here. Its ability to analyse logic circuits and mixed logic/analogue circuits will be a topic for our new series 'The Digital Solution', which begins next month. But we can not finish this account without looking at Fourier analyses. Again we take an example that we have previously dealt with. In Part 15, **Fig. 127**, we found the Fourier series for a triangular wave of period 2π seconds, amplitude 5π volts, displaced 5π volts, so that its minimum value is 0 V. We then analysed what happens when it is passed through a high-pass filter. For the SAW analysis, we netlisted an *RC* high-pass filter with C = 800 µF and R = 100 Ω, which has a –3 dB point at 2 Hz. The input to the circuit was the triangular wave. A transient analysis of this shows its triangular shape; the duty factor is set to 0, so that it begins with a rapid rise, followed by a steady fall (**Fig. 176**). On the same graph we have plotted the output, after filtering. This has the same form as that calculated in **Fig. 130**, Part 15, being a sharp spike at the beginning of each cycle. The first spike is not as high as later spikes since we are simulating a real circuit (not using a pure mathematical function as in Part 13) and it begins in an unexcited state. Having run the transient analyses, we can perform a Fourier analysis on each. The analysis of the triangular wave is plotted as a frequency spectrum in **Fig. 177**. We can read off the amplitudes of the fundamental and harmonics: 10 V, 5 V, 3.33 V, 2.5 V ... This is the same as **Fig. 138** of Part 16, but here we have values for many more of the higher harmonics. Note that there are also some intermediate harmonics of relatively low amplitude, resulting again from the fact that we are analysing a simulated circuit, not a mathematical equation. The graph also displays information on the phase angle of each harmonic, indicated by the small

```
In[27]:=
NSolve [{6i[1]–3i[2]–i[3]==0, –3i[1]+10i[2]–5i[3]==2, –i[1]–5i[2]+13i[3]–7i[4]==–3, –7i[3]+11i[4]==5}, {i[1], i[2], i[3], i[4]}]
Out[27]=
{{i[4]–>0.649408, i[1] –>0.267751, i[2] –>0.433432, i[3] –>0.306213}}
```

Fig. 178

```
In[43]:=
DSolve [{y'[t]==–y[t] / rc, y[0]==0.00423}, y[t], t]
Out[43]=
–{–y[t] –>0.00423/E^(t/rc)}

In[44]:=
Plot [0.00423/E^(0.0967*t), {t, 0.10}]
Out[44]=
```

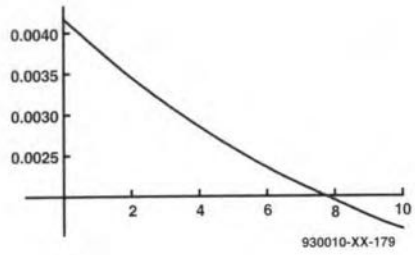


Fig. 179

```
In[3]:=
Plot3D[u/r], {r, 0.01, 10}, {u, 0, 10}, ViewPoint –>{15, 15, 10},
AxesLabel –>{"u", "r", "i"}, BoxRatios –>{1, 1, 1}]
Out[3]=
```

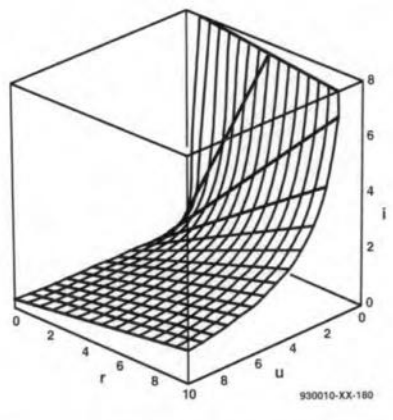


Fig. 180

squares.

Alternative approach

With SpiceAge© and comparable simulation software we **build** a circuit by netlisting it (instead of physically assembling it from real components). Then we use the routines of the simulator to **test** it (instead of probing it with a multimeter, an oscilloscope or a frequency analyser). The approach is essentially a **practical** one even though it is carried out on a computer. The alternative approach is theoretical, in which we work with laws and equations.

To highlight the differences in the two approaches, let us return to the network in **Fig. 163**. Following the techniques described in Part 4, we would reduce this to a set of simultaneous equations, one for each mesh of the network, based on Kirchoff's Voltage Law. Writing out the equations is easily done; solving them is a lengthy matter. The recognised technique is to evaluate the fourth-order determinant based on the coefficients of the equations. With **Mathematica** we need only to key in the equations and instruct the software to solve them. **Figure 178** shows the screen for this computation, which was the 27th one in the current session. The command 'NSolve' means 'solve the following equation and give numerical results'. The first set of curly brackets shows the four equations; note that 'i[1]' is the standard way to key in a suffixed symbol such as i_1, and also that the double equals sign '==' is **Mathematica**'s equivalent of '='. The second curly bracket lists the variables, i_1 to i_4 for which values are required. The output appears on the screen within a few seconds. This gives $i_1 = 0.267751$ and $i_2 = 0.433432$. We were asked to find the current, i, through the 3 Ω resistor:

$i = i_1 - i_2 = 0.267751 - 0.433432$
$= -0.165681$ A.

This is the same result as obtained from simulation.

Mathematica is essentially a mathematical package and it provides a wealth of mathematical routines for anyone who has maths to do. Routines of special interest to the electrical engineer include its ability to differentiate and integrate. More than that, it can solve differential equations. Our first modelling example (Part 12) produced a differential equation (Eq. 83) for a charging capacitor:

$dq/dt = -q/RC$.

Figure 179 shows this entered as a command, using the variable y in place of q, and quoting the border condition $y(0) = 0.00423$. This value is the value of A that was used in Eq. 85. The program then calculates and displays the solution of the equation in symbolic form; interpreting the conventions for printing out:

$$y = 0.00423/e^{t/RC}.$$

This is the equivalent of Eq. 86, for

$$0.00423/e^{t/RC} \equiv 0.00423e^{-t/RC}.$$

This function is plotted out by the computer for $t = 0$ to $t = 10$, and corresponds to the thick-lined curve of **Fig. 104**.

Another differential equation (Eq. 94) used in this series for modelling an *RLC* circuit:

$$d^2i/dt^2 + (R/L)di/dt + 91/LC)i = 0.$$

This equation is purely symbolic and it is one of the strengths of **Mathematica** that it can work exclusively in symbols. Its solution of the equation is:

$$i = C_1e^{-[[CR-\sqrt{(-4CL+C^2R^2)}]t]/(2CL)} + C_2e^{-[[CR+\sqrt{(-4CL+C^2R^2)}]t]/(2CL)}.$$

C_1 and C_2 correspond to the constants A and B, determinable by knowing the starting conditions. Here, the equation is in a form that can be applied to any combination of values for R, L, and C. If we insert the values used in the example in Part 13, we obtain the particular solution given in Eq. 100. We could, of course, have put these values into the original calculation, along with starting conditions, when we would have obtained the numerical solution directly. For those who are familiar with the Laplace Transform (Parts 18 and 19) the program can perform transforms in both directions.

Graphics

The graphic output of **Mathematica** includes 3D plots, with colour shading to enhance the effect. **Figure 180** is a plot of the equation $i = u/r$, which was illustrated by **Fig. 143** in Part 17. The value of u ranges from 0 to 10, the value of r runs from 0.001 to 10, so as to avoid a 'division by zero' error. When generated, the default viewpoint looks over the high part of the surface, so we have specified another viewpoint which corresponds to that used in **Fig. 143**.

Electronic applications

A purely mathematical approach is excellent for those who are well up on their theory, but many users would prefer to be provided with routines adapted to electrical and electronic applications. These are available as a library of files (known as Notebooks) published as an **Electrical Engineering Pack**. The notebooks comprise text to explain the applications and ready-made routines for use by engineers. These make direct use of the computational routines of **Mathematica**, acting as an interface between the practical engineer and the mathematics. To help expedite the calculations, the pack includes a number of routines not already present in **Mathematica**, which are specially applicable to electronics. These include converting between a linear magnitude scale and the decibel scale, and converting from polar coordinates to complex (rectangular) coordinates. Another useful routine selects the nearest standard resistor value to a specified value. These routines are simple enough in themselves, but their real benefit is that they can be written into other more complicated routines.

Nodal is a circuit analysis package written to interface with **Mathematica**. It accepts circuits specified by netlists and analyses circuit behaviour by means of symbolic equations, numerical results and graphics. Like **Mathematica**, it favours the analytical approach to circuit design, but concentrates the engineer's attention on those aspects of **Mathematica** relevant to electronics. Unfortunately, this article was almost complete when we received our copy of **Nodal** so a more detailed examination of this promising software will have to be deferred until another occasion.

The choice

In this final part of 'Figuring it Out', we have demonstrated two complementary approaches to computer-assisted circuit design. We believe that the choice between a practical approach and a more analytical one is very much a matter of personal preference. Some will like one, some will like the other. We like both.

[930010-XX]

Acknowledgments

We wish to thank Those Engineers Ltd for their valuable advice and assistance with **SpiceAge**© for **Windows**™; Wolfram Research Europe Ltd for their help with **Mathematica**® and their **Electrical Engineering Pack**; and Goth, Goth and Chandleri for assistance with **Nodal**.

Reference

SMT Projects, Owen Bishop, published 1993 by Elektor Electronics (Publishing).

Answers to Test Yourself Part 19)

1. See Fig. 159.
   $U(s)=220[1/s-1/(25.22+s)$.
   $u(t)=220(1-e^{-25.22t})$. Initial voltage = 0 V. Final voltage = 220 V.

2. $I(s)=20/(s^2+5000+10^5)$. The denominator has real roots, 20.08 and 4980, and we can obtain partial fractions..
   $I(s)=4.032[1/(s+20.08)-1/(s+4980)]$.
   $i(t)=4.032(e^{-20.80t}-e^{-4980t})$.
   The current rises steeply to a peak of 3.93 A at about 1 ms, then dies away smoothly, and much more slowly, without oscillating, falling to 7.7 nA in 1 s. It is overdamped.

PIC® PROGRAMMING COURSE

PART 4 (FINAL) : INSTRUCTION SET

In this fourth and final part of the course we tie up some loose ends and tackle the remaining instructions from the instruction set. In particular, conditional and non-conditional branch instructions are discussed, which enable you to control the program flow.

By our editorial staff.

Source: Microchip Technology Inc.

Let us start by returning to the example program, Fig. 2 in last month's instalment. Usually, a computer program is executed from the lowest to the highest memory location. Special instructions, for example, skipping a piece of the program, are available to control the flow of a program. The simplest of these is **NOP**, as used in line 106. This instruction increases the program counter by one, and takes exactly one instruction cycle. For the rest, it does nothing. The formal description of NOP, like that of all other instructions mentioned in this instalment, is shown in a box on the next page.

All other instructions which control the program flow are branch ('jump') instructions. Non-conditional instructions are, for instance, **GOTO** (line 108) and **CALL** (line 135), which branch **always**. By contrast, conditional branch instructions do not jump until a certain condition is satisfied.

Non-conditional jumps

In a **GOTO** instruction, the lowest bits are used to indicate the target address. In the case of the PIC16C56, the level of bit 9 is determined via the status register. Similarly, bit 9 and 10 are so determined with the PIC16C57. These bits must be configured in accordance with the desired address. After a hardware reset, these bits are all set to 0, so that a simple GOTO instruction may be used after the RESET address (16C54 and 55: 1FFH; 16C56: 3FFH; 16C57: 7FFH) to jump to the start of the program.

In 'classic' programming techniques, subroutines have an important function: they allow an instruction sequence which is needed many times to be included at only one location in the program. Any time the subroutine is needed, it is simply called. Modern programming techniques use subroutines to construct a program on the basis of functional blocks.

Subroutines always have a starting address, and are closed with a **RET** instruction, which marks the end of the sub-program. When a subroutine is called using a CALL instruction, it is important to remember that address bit 8 of the jump address is always at 0. The CALL instruction only allows bits 0 through 7 to be entered. As with the GOTO instruction, address bit 9 of the PIC16C56, and address bits 9 and 10 of the PIC16C57, are determined by bits PA0 and PA1 from the status register. Consequently, the start addresses of subroutines must always be in the lower part of the program memory. This rather annoying restriction can be obviated by placing a GOTO instruction in the lower memory block. This GOTO should point to subroutines in the higher memory block. When the CALL instruction is executed, the current contents of the program counter (PC) are pushed on to the stack. Since there are only two memory locations available on the stack, only two CALLs can be nested. Because the complete contents of the program counter is stored into one memory location on the stack, CALL instructions may be included at any location in the program.

Each subroutine must be terminated by an **RETLW** instruction. RETLW pops the stack contents and copies it to the program counter. The program then continues at the first instruction after the CALL.

RETLW also places a constant in the W register. This constant may be used to convey information from the subroutine to the main program.

Conditional jumps

Conditional jumps in PIC16C5x assembly code have a fixed jump address. If the previously defined condition is satisfied, the instruction after the conditional jump is skipped. As illustrated by lines 130 and 131, the 'skipped' memory location may contain a GOTO instruction if it is desired to jump to a certain routine.

The PIC's instruction set has only four conditional jump (or branch) instructions: **BTFSC f,b** and **BTFSS f,b**, which test bit b

```
        ; Compare Register 'us_Register' with 046H

Comp_Value equ 046H

COMPARE  movlw Comp_Value      ; Load W with 046H
         xorwf us_Register, W ; XOR W with us_Register
         btfsc 3,2            ; W = 0?
           goto EQUAL         ; Yes: us_Register=046H
NOT_EQUAL                     ; No: us_Register<>046H
         movlw Comp_Value     ;
         subwf us_Register, W ; us_Register - W
         btfsc 3,0            ; W > 0 (Carry = 1?)
           goto GREATER       ; Yes: us_Register>046H
         goto LESS            ; No:  us_Register<046H
LESS     .....                ; Routine for us_Register
                              ; <046H
GREATER  .....                ; Routine for us_Register
                              ; >046H
EQUAL    .....                ; Routine for us_Register
                              ; =046H
```

Fig. 1. Example of a routine written to compare a variable with a constant.

in register f, and the pair **DECFSZ f,d** and **INCFSZ f,d**, which decrease and increase, respectively, the contents of register f by one. The instruction which follows any of these conditional jump instructions is skipped if the result of the relevant operation is 0. If d=1, the new value is stored back again in register f. If d=0, it is stored in the W register. In the latter case, the original value is left in register f. See the page opposite for the formal descriptions of these instructions.

Comparing with a constant
It will often be required to check a variable against a constant. The example program listed in **Fig. 1** shows how that can be achieved in a simple manner. The approach is almost identical if you wish to compare the contents of two registers; all you have to do is replace the **MOVLW Comp_Value** instruction by a **MOVF** instruction.

Logic instructions

The 16C5x processor offers seven logic instructions. One of these is **XORWF**, which is also used in the example program. There is not much to say about these instructions, all operations being carried out bit-by-bit. With instructions that make use of a register from the register file, the result of an operation (for instance, an addition) may be stored either in the W register, or back into the register file. With all logical instructions, the Z flag is actuated if the result of the operation is 0.

Rotate instructions are usually classified with logic operations. The PIC processor has two rotate instructions: clockwise, **RRF** (rotate right), and anti-clockwise, **RLF** (rotate left). The 'carry' bit is also involved in the rotate operation. Rotate right means that the bits are shifted down one position, the MSB taking the value of the Carry bit, and the LSB moving into the Carry bit. The other operation, RLF, does the same, but the other way around. Just as with all other byte-oriented instructions, the result of RRF and RLF may be stored either in the W register or in a memory location.

NOP	No OPeration		
Syntax	nop	Status bits	none
Description	No operation, program counter incremented by 1.		
Example	*nop*		

GOTO	Unconditional Branch		
Syntax	goto <addr>	Status bits	none
Description	Direct branch to an address of which the 9 lower bits (0 to 8) come from <addr>; bit 9 of the PIC 16C56 and bits 9 and 10 of the PIC 16C57 are loaded from the PA <2:0> bits in the status register.		
Example	*goto Main_Start*		

CALL	Subroutine Call		
Syntax	call <addr>	Status bits	none
Description	Value (Program Counter + 1) placed onto stack. Branch to address <addr> in the lower half of the page (bit 8 = 0 !) defined by the status register (A9 = PA0, A10 = PA1 for PIC16C54 and 16C57 only).		
Example	*call Wait_ms*		

RETLW	Return literal to W		
Syntax	retlw k	Status bits	none
Description	The W register is loaded with the 8-bit literal k. The program counter is loaded from the top of stack register 1. The content of stack register 2 is moved to stack register 1		
Example	*retlw 000H*		

BTFSC	Bit Test, skip if clear		
Syntax	btfsc f, b	Status bits	none
Description	Skip next instruction if bit b in register f is 0.		
Example	*btfsc 3,0 ;jump if CARRY (C) = 0*		

IORLW	Inclusive OR literal with W		
Syntax	iorlw k	Status bits	Z
Description	The contents of the W register are OR'ed with the 8-bit literal k. The result is placed in the W register.		
Example	*iorlw 046H*		

XORLW	Exclusive OR literal and W		
Syntax	xorlw k	Status bits	Z
Description	The contents of the W register are XOR'ed with the 8-bit literal k. The result is placed in the W register..		
Example	*xorlw 046H*		

COMF	Complement f		
Syntax	comf f, d	Status bits	Z
Description	The contents of register f are complemented. d = 0: result stored in W register d = 1: result stored in register f		
Example	*movlw 01010111B ;* *movwf us_Register ; us_Register = 01010111B* *comf us_Register, 1 ; us_Register = 10101000B*		

RRF	Rotate Right f through Carry		
Syntax	rrf f, d	Status bits	C
Description	The contents of register f are rotated one bit to the right through the Carry Flag. d = 0: result stored in W register d = 1: result stored in register f		
Example	*movlw 10011010B ; load 10011010 into register RReg* *movwf RReg ; RReg = 10011010* *bsf 3,0 ; C = 1* *rrf RReg, 1 ; => RReg = 11001101; C = 0*		

PIC16C5x PROGRAMMING

This short course is aimed at providing an introduction into programming and hardware aspects of the PIC16C5x family of microcontrollers manufactured by Microchip Technology Inc.

An **assembler** is offered on disk in support of the course. This assembler is distributed with the permission of Microchip Technology Inc., and supports the PIC16C5x and PIC16Cxx series of controllers. It offers a full featured macro and conditional assembly capacity. It can also generate various object code formats including several hex formats to support Microchip's proprietary development tools as well as third party tools. Also supported are hex (default), decimal and octal source and listing formats. An assembler users manual is available from Microchip Technology distributors for detailed support. The disk also contains a software **simulator**.

The PIC programming course disk may be obtained through the *Elektor Electronics* Readers Services under order number **946196-1**. For price and ordering information, see page 70 of this issue.

The files produced by the assembler can be downloaded to the **PIC programmer** described in *Elektor Electronics* March 1994.

Bit instructions

PIC processors have a total of four bit-oriented instructions. Two of these, **BTFSC** and **BTFSS**, are bit-oriented

jump instructions. These instructions execute a jump depending on a bit condition. Both were discussed above in the section about conditional jump instructions. The two other instructions may be used to set or reset a bit in one of the memory locations of the register file. These instructions have two operands: the first indicates the address in the register file, and the second, the bit number. Bit 0 always refers to the least-significant bit (LSB).

Line 133 of the example program contains the **BCF** instruction. It is used to switch on the LED connected to the RAO line by pulling the port output logic low. The LED is turned off again in line 137 by making RA0 high using the **BSF** instruction.

The BCF and BSF instructions have a special feature: they are processed by internally accessing the indicated location at a 'width' of eight bits. This means that the current state of a port (with the exception of the bit to be changed) is read and stored directly into a buffer memory. If an I/O port is configured as an input for the bit instruction which is applied to one of the other pins, and it is configured as an output after the execution of the instruction, the level at the relevant output is made equal to that at the input as it was while the instruction was being executed. It is, therefore, advisable to make sure that the buffer is at a fixed level before switching the associated port from input to output.

Hardware component settings

Back to the example program. The main program should always start with an initialisation of the hardware components. This information is contained in lines 126 and 127, which set up the I/O ports and the option register. Two instructions are available to do so. The **TRIS** instruction allows you to define a port line as an input or an output, while **OPTION** serves to fill the OPTION register. The hardware settings may be changed as the program evolves. To ensure that the program does not crash in an electrically noisy environment, Microchip advises programmers to repeat the hardware settings at several locations in the program. In fact, an internal mechanism in the processor even repeats the definition of the I/O pins with each action on the I/O port. Such drastic measures are necessary because of the irrevocable resetting action of the watchdog timer, which forces a reset when the program crashes. Such a reset may also cause other bits than the ones specified to be reset, i.e., corrupted.

RLF	**Rotate left f through Carry**		
Syntax	rlf f, d	**Status bits**	C
Description	The contents of register f are rotated one bit to the left through the Carry Flag. d = 0: result stored in W register d = 1: result stored in register f		
Example	*movlw 10011010B ; load 10011010 into register RReg* *movwf RReg ; RReg = 10011010* *bsf 3,0 ; C = 1* *rlf RReg, 1 ; => RReg = 00110101; C = 0*		

BSF	**Bit Set f**		
Syntax	bsf f, b	**Status bits**	none
Description	Bit b in register f is set to 1		
Example	*bsf Port_A, 000H ; set bit 0 of Port_A*		

BTFSS	**Bit test, skip if Set**		
Syntax	btfss f, b	**Status bits**	none
Description	Skip next instruction if bit b in register f is 1.		
Example	*btfss 3,2 ;jump if zero (Z = 1)*		

INCFSZ	**Increment f, skip if 0**		
Syntax	incfsz f, d	**Status bits**	none
Description	The contents of register f are incremented. d = 0: result stored in W register d = 1: result stored in register f. Skip next instruction if the result is 0.		
Example	*incfsz us_Register, F*		

DECFSZ	**Decrement f, skip if 0**		
Syntax	decfsz f, d	**Status bits**	none
Description	The contents of register f are decremented. d = 0: result stored in W register d = 1: result stored back in register f Skip next instruction if the result is 0.		
Example	*decfsz us_Register, F*		

ANDWF	**AND W with f**		
Syntax	andwf f, d	**Status bits**	Z
Description	AND the W register with register f. d = 0: result stored in W register d = 1: result stored in register f.		
Example	*andwf us_Register, W*		

IORWF	**Inclusive OR W with f**		
Syntax	iorwf f, d	**Status bits**	Z
Description	Inclusive OR the W register with register f. d = 0: result stored in W register d = 1: result stored in register f		
Example	*iorwf us_Register, W*		

XORWF	**Exclusive OR W with f**		
Syntax	xorwf f, d	**Status bits**	Z
Description	Exclusive OR the contents of the W register with register f. d = 0: result stored in W register d = 1: result stored in register f		
Example	*xorwf us_Register, W*		

ANDLW	**AND literal and W**		
Syntax	andlw k	**Status bits**	Z
Description	The contents of the W register are ANDed with the 8-bit literal k. The result is placed in the W register.		
Example	*andl 046H*		

The OPTION instruction

The **OPTION** instruction puts the contents of the W register into the OPTION register. The functions of the individual bits in that register were discussed as part of the processor's hardware description.

Other instructions

Apart from the those discussed with reference to the example program, there are three more instructions.

The **SWAPF** is hard to place in a certain category. It swaps the lower and the higher nibble (bits 0 through 3, and 4 through 7, respectively). This instruction is often used for BCD arithmetic.

The watchdog timer of the PIC processor is actuated by programming. The timer may be reset using the **CLRWDT** instruction. If the prescaler is assigned to the watchdog, it is reset also. The watchdog timer may also be reset using the **SLEEP** instruction. This instruction causes the PIC processor to switch to power-down mode. The processor 'wakes up' again when the watchdog time has elapsed, or when a hardware reset occurs.

End of the program

The end of a PIC program invariably contains the defini-

BCF	Bit Clear f		
Syntax	bcf f, b	Status bits	none
Description	Reset bit b in register f to 0.		
Example	*bcf Port_A, 000H ; reset bit 0 of Port_A*		

TRIS	Load TRIS register (tristate port)		
Syntax	tris f	Status bits	none
Description	TRIS register f is loaded with the contents of the W register. Valid values for f are: 05H for Port A, 06H for Port b, and 07H for Port C. A set bit (1) defines a port pin as an input; a reset bit (0) defines a port pin as an output.		
Example	*movlw 00EH ; set RA0 to output* *tris Port_A ;*		

OPTION	Load Option Register		
Syntax	option	Status bits	none
Description	The contents of the W register are loaded in the OPTION register.		
Example	*movlw 00100110B ; Transition on RTTC pin* *; Increment on low to high* *; transition on RTTC pin* *; Prescaler to RTTC* *; Prescaler value = 1 : 128* *option ; Load contents of W register* *; into OPTION register*		

SWAPF	Swap f		
Syntax	swapf f, d	Status bits	none
Description	The upper and lower nibbles of of register f are exchanged. d = 0: result stored into W register d = 1: result stored back in register f		
Example	*movlw 10011010B ;* *movwf RReg ; RReg = 10011010B* *swapf RReg, 1 ; RReg = 10101001B*		

CLRWDT	Clear Watchdog Timer		
Syntax	clrwdt	Status bits	TO, PD
Description	Reset the watchdog timer. Also reset the prescaler of the WDT. Set PD and TO.		
Example	*clrwdt*		

SLEEP	Put into sleep mode		
Syntax	sleep	Status bits	TO, PD
Description	The power-down status bit (PD) is cleared. Time-out status bit (TO) is set. Watchdog Timer and its prescaler are cleared.		
Example	*sleep*		

tion of the RESET vector. Since a program will not normally occupy the entire memory of the processor, a trick must be used to make the vector point to the end of the memory range. That can be achieved by using the **ORG** instruction, as illustrated in line 144. Because the different types of PIC controller have their own memory size, the end address is set up with the aid of an IF-ENDIF structure at the start of the program.

The description of the end of the program also marks the end of this short course in PIC programming. If you have developed an interesting application based on a PIC, let us know.

(940062-4)

PIC is a registered trademark of Microchip Technology, Inc.

DEBUGGING THE 8031 SERIES

Few things are more frustrating than a circuit which refuses to come to life even after burning the midnight oil trying to find out what is wrong. Fortunately, any fault can be located fairly quickly if you stick to a structured way of 'debugging'. In this article, a ten-step guide is given for successful faultfinding on an MCS-51 (Intel 8031 series) microcontroller system.

By our editorial and design staff.

BEFORE you start scouring for faults and measuring signals on an 8031-based microcontroller system which looks dead as a doornail, take a moment to reflect on the following assumptions which are necessary to be able to use this article. Firstly, it is assumed that the circuit is complete, that is, all ICs are fitted, including the programmed EPROM. Also, measurements are made at the top side of the board, directly on the IC terminals. Never push a probe or test pin into a spring-loaded receptacle as used in low-cost IC sockets. That will deform the contacts a little, causing a bad contact with the IC terminal, and still more problems, after the test.

01 Always start by checking the supply voltage. Connect the common terminal of your voltmeter to the supply ground, and then put the + probe on the positive supply terminal (usually pin 40) of the processor. Do the same for all ICs in the circuit. In case of doubt, refer to the circuit diagram to locate the $+V_{cc}$ or $+V_{dd}$ terminal of a particular IC.

If the supply voltage is missing, check for bad IC sockets, bad solder joints and/or broken PCB tracks.

02 Next come the ground connections of all ICs. In the case of the processor, this is usually pin 20. One test probe of the DVM is connected to the positive output of the supply. The other is used to probe around on all IC ground connections, checking for the presence of the supply voltage (the meter will indicate 5 V or –5 V). All measurements should produce virtually the same meter indication. The voltages may only differ a few millivolts from the one measured directly at the output of the power supply. The supply voltage itself must lie in the range between 4.75 V and 5.25 V. Do the above test on all ICs in the circuit.

If the ground connection is broken, check for a bad IC socket, bad solder joints and/or broken PCB tracks.

03 If all supply terminals are okay, it is time to concentrate on a number of functions. The first of these is the processor oscillator. Check its operation by connecting your oscilloscope to pin 18 of the processor, via a 10:1 probe. Check if the measured fre-

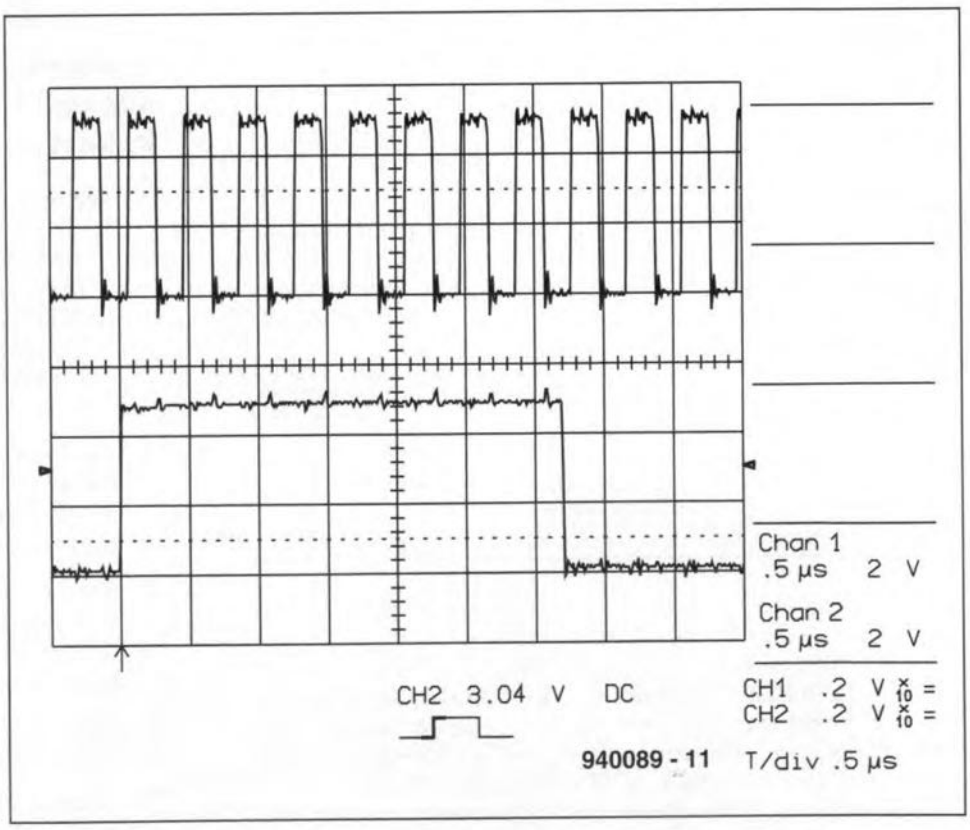


Fig. 1. Upper trace: PSEN signal used to enable the EPROM. Lower trace: LSB of port P1 while the test program runs. An 80451 processor is used here. This is a CMOS device which is capable of producing a port output voltage swing of almost 0 to 5 V. With NMOS processors, the logic high level will be between 3 V and 3.5 V. The 8031 series is TTL compatible, which means that with CMOS processors, too, a voltage level below 0.8 V is recognized as a logic zero, and any value greater than 2 V, as a logic one. Levels between these values may never occur in a system. The amount of noise shown here is normal, considering that a 150-MHz digital oscilloscope was used for this screen snapshot. If a 20-MHz scope is used, the signal will appear to be much cleaner. In practice, a processor running at a clock of 15 MHz can be expected to produce harmonics and spurious products up to 200 MHz.

quency is roughly equal to the one printed on the crystal. The level of the oscillator signal should be a couple of volts (peak-to-peak), and the signal should have the shape of a sine-wave with some distortion and irregularities. If the oscillator does not work, pins 18 and 19 will be at a steady logic level, that is, at +5 V or 0 V. Once the oscillator has stopped (for whatever reason), it will not be able to start again on its own. For the oscillator to start properly, the processor must be reset first. It is, therefore, recommended to press the reset key a few times while looking at the oscilloscope screen to detect oscillator action.

☞ **If the oscillator fails to work, check the PCB tracks to pins 18 and 19, and the relevant solder joints. Check that both capacitors are properly connected, also to ground. Check the frequency printed on the crystal, which is usually in kHz, although MHz indications may be used if the frequency is a multiple of 1 MHz. Temporarily exchange the crystal by a type which is known to have worked properly in another circuit.**

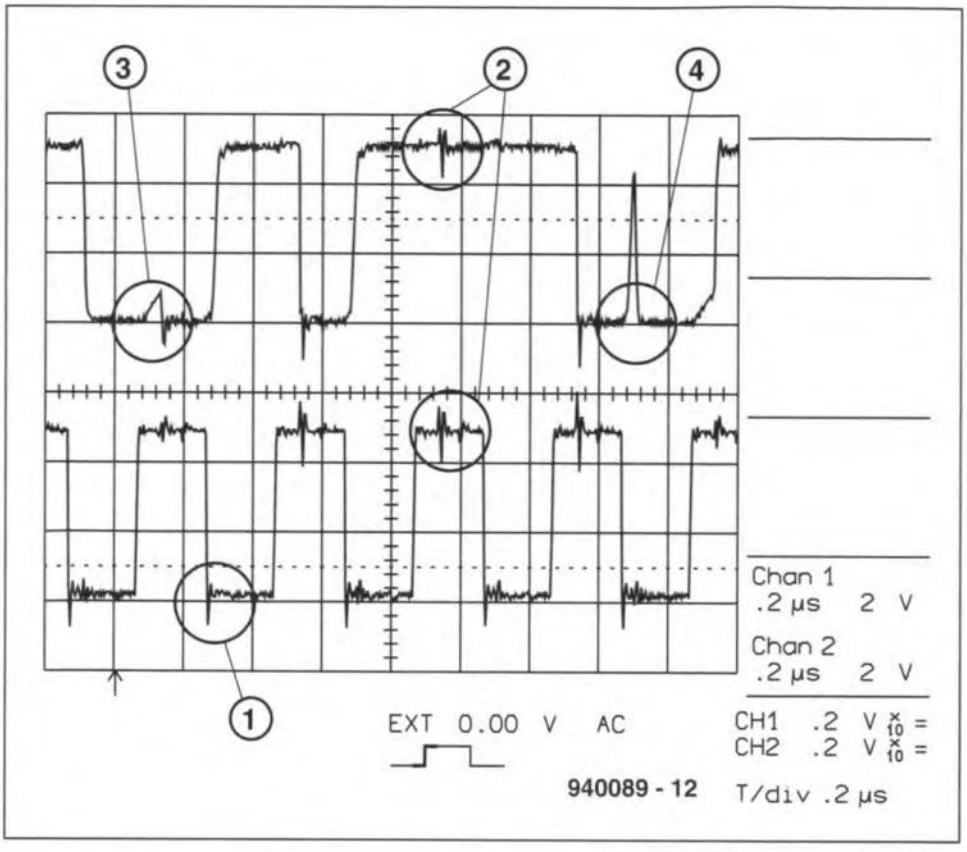


Fig. 2. Multiplexed processor data bus (upper trace) and the PSEN signal (lower trace). Because the lines are not terminated on the board, the PSEN pulses go negative at times (1). The overshoot is, in principle, the same on the positive voltage change. It is, however, less pronounced because positive-going edged are a little slower than negative-going edges. Fortunately, today's logic circuits are up to these voltage peaks, and just keep ticking. Here, too, a 20-MHz scope will fail to reveal such peaks. The positive supply voltage has a lot of pulse ringing on it (2). This effect is caused by the switching of the address latch (not shown). Although they appear to be faulty at first glance, signals (3) and (4) are normal. At (3), the CE line of the EPROM is no longer active, and the databus is, therefore, in high-impedance mode. The pull-up resistors cause the voltage level to rise slowly. A little later, the processor puts the address information on the bus, in this case, a nought. At (4) you can see the inertia of the EPROM. With a CE pulse, data appears instantly which belongs to a random internal state, in this case, a logic one. It takes a while for the applied address to be decoded internally, whereupon the right data, in this case, a logic zero, appears at the output. The spurious voltage peak does not cause problems because the processor copies the data on the positive-going edge of the PSEN signal. By that time, the data has been steady for 200 ns already.

04 In some cases it is less desirable to measure directly at the oscillator pins, mainly because of the extra capacitance introduced by the oscilloscope probe. That is why is it is also useful to check for 'digital activity' on the processor's ALE and PSEN terminals. Note, however, that the ALE and PSEN signals are only present if the processor is active — they will not be found as long as the reset pin is actuated. Use the scope to check if the ALE pin supplies a square wave with a frequency of about 1/6th of the oscillator frequency, and a duty factor of about 0.3. The PSEN pin supplies a signal with the same frequency as ALE, but a duty factor of about 0.5. If available, use a frequency meter to check the frequency of the ALE pulses. This is in fact a reliable test on the oscillator, which is not loaded in that way. Unfortunately, if your program uses instructions to access data in external RAM locations, some ALE and PSEN pulses will be missing. In these cases, there is no alternative but to measure directly on the oscillator pins.

☞ **If the oscillator signals are faulty, keep resetting and starting the processor. It could be that the reset terminal remains 'stuck' at the high level. Remove all ICs which are connected to the ALE or PSEN line. If these signals remain faulty, do a short-circuit check on the relevant PCB tracks and IC sockets.**

05 If an external EPROM or ROM is used in the circuit, it is necessary to check that the level at the processor's EA (external access) terminal is logic low. EA is usually found on pin 31. Use the oscilloscope to inspect all signals on the combined data/address lines (port P0). It is not normally necessary, or indeed useful, to keep the scope triggered — simply watch the low and high voltage levels. The low levels must be lower than 0.4 V. The high levels must be higher than 3 V. Some variation may be seen in the high levels. Voltage levels in between 'low' and 'high' are a clear sign that something is amiss.

☞ **Lines which seem to remain at a fixed logic level are suspect. In such cases, scrutinize the PCB tracks and IC sockets for short-circuits. If you are debugging a circuit designed by yourself, go back to the circuit diagram to check for fundamental errors.**

Intermediate voltage levels point to short-circuits between PCB tracks. The 'other' track is simple to find because it has the same, faulty, signals. To be able find it, use the same trigger source for both channels of your dual-channel scope. Alternatively, use 'chopper' mode (instead of 'alternate').

06 Check out all address lines as under point five. Be sure to include the lines 'behind' the latch. The higher address lines are

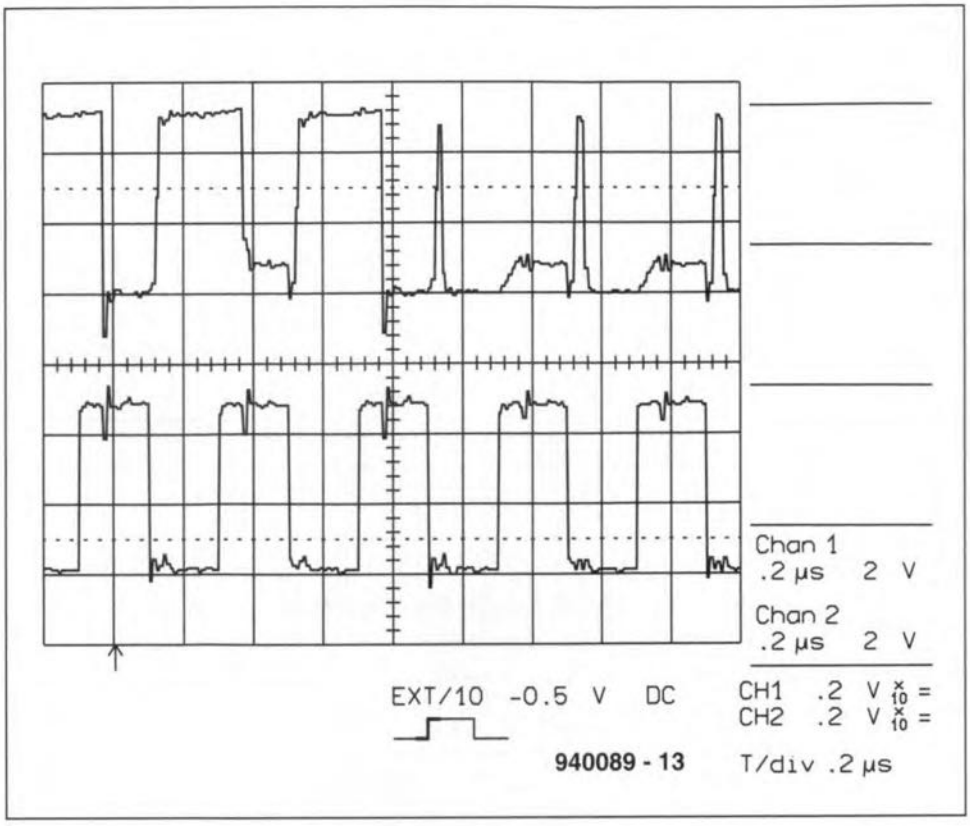


Fig. 3. Short-circuited databus lines (upper trace) and the PSEN signal (lower trace). Apart from the virtually perfect high and low levels, an intermediate level of about 1 V appears occasionally. This is caused by one databus line trying to pull the line logic high, and the other, logic low. In MCS-51 devices, the pull-down resistor is 'stronger' than the pull-up resistor, causing a contention voltage of about 1 V instead of the expected 2.5 V.

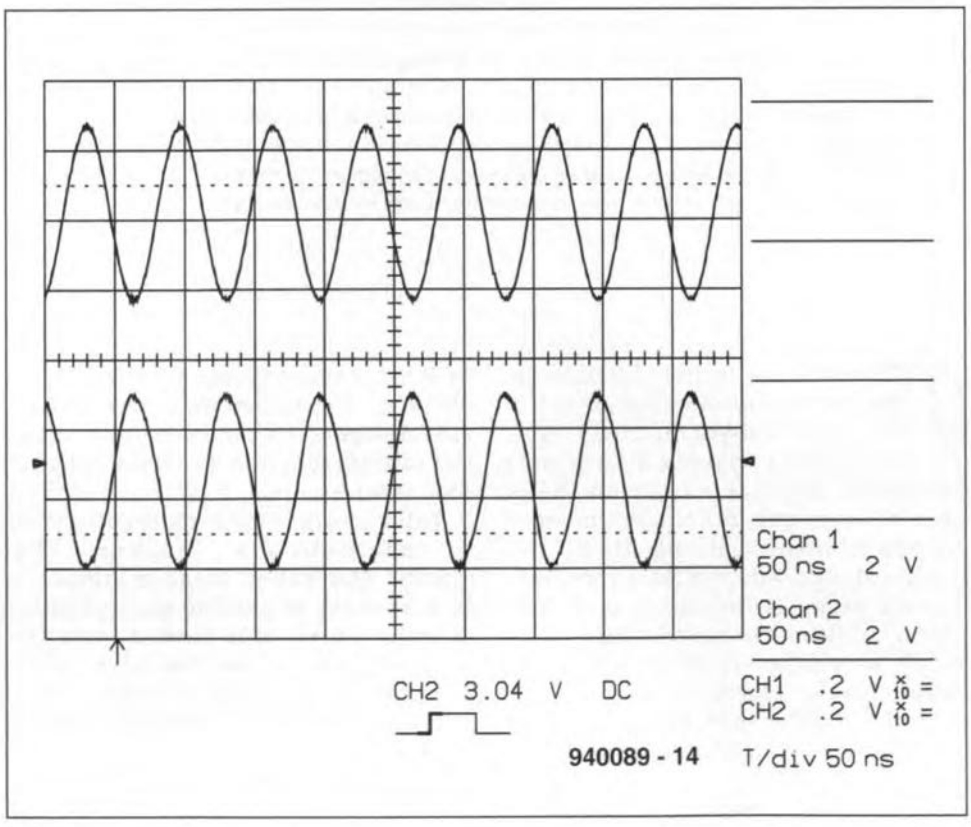


Fig. 4. Oscillator voltages. There is little to say about them. A 20-MHz scope will normally indicate a smaller peak-to-peak level. Oscillator levels of a few volts are sufficient to make the processor work correctly.

often stable because only a limited part of the memory is used in most small systems. The correct operation of these lines can be verified with the aid of the 'NOP' method described further on.

Intermediate voltage levels, or address lines which appear to remain at a steady level, are always suspect. Check the relevant PCB tracks and IC socket contacts for short-circuits. If you are debugging a circuit designed by yourself, go back to the circuit diagram to check for fundamental errors.

07 Use the scope to inspect the signals and logic levels at all other processor pins. Ask yourself if the voltage level measured makes sense. Do take into account that the 8031 has internal 30-kΩ pull-up resistors on all port lines. Consequently, a port line is capable of sourcing only 150 µA. The low-value pull-down resistors, however, enable a current of up to 1.5 mA to be switched to ground ('sink current'). These two resistors on each port line cause a voltage level which depends strongly on the load.

Over to software

When tests five, six, and seven fail to isolate the problem and get the circuit to work, there is no alternative but to use a special test program. Such a test program is easy to write yourself, and should consist of the simplest possible loop in which, for instance, a continuously changing bit pattern is written to an I/O port. By looking at the data on the relevant I/O port you are, hopefully, able to see if the processor gets stuck somewhere in the loop. An abstract but none the less suitable example:

```
increment accumulator by 1
write accumulator contents to port
jump to start
```

If the program does not work, check if all jumpers are correctly fitted for the EPROM type used. You should also check the settings of the EPROM emulator, if such a tool is used.

Nothing works!

Normally, the fault should have been located by now. However, in those few cases where the malfunction seems to persist in spite of your best attempts to find it, do not give up. Your persistence will be rewarded, as evidenced by many happy letters received by our Technical Queries service over the past few years. So, read on.

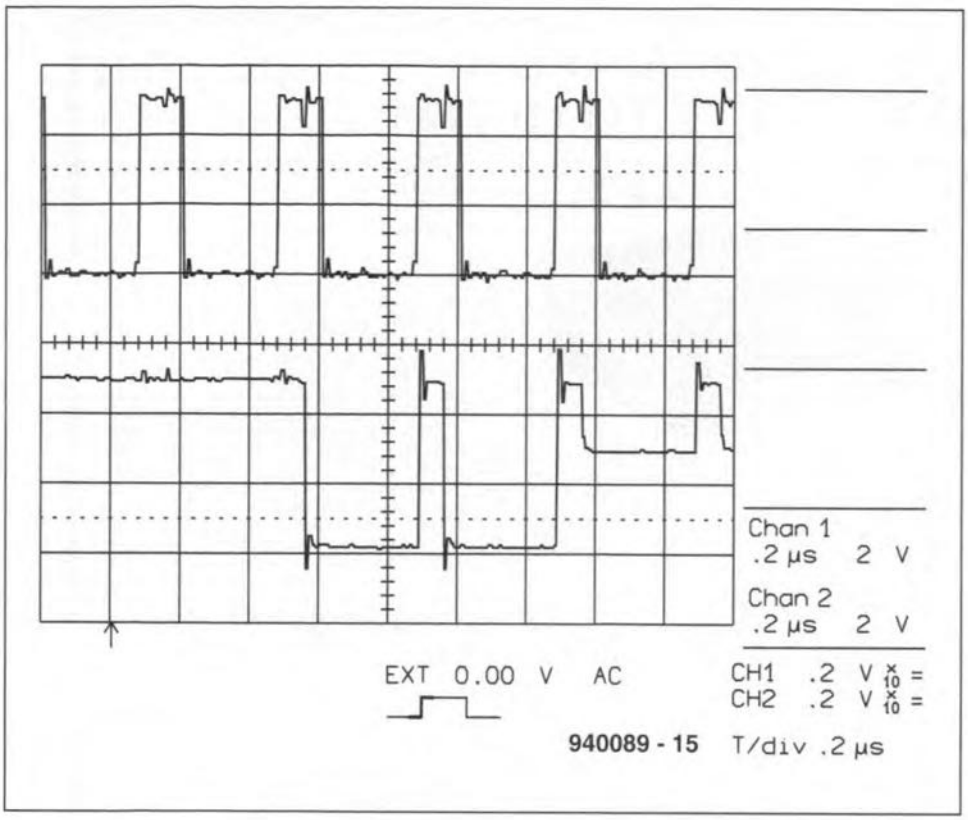


Fig. 5. A short-circuit on the address bus (lower trace) behind the latch (a HCT573). The upper trace shows the ALE signal. Apart from the virtually perfect high and low levels, a third level of about 3 V appears occasionally. It is caused by one address line trying to pull the line logic high, and the other, trying to pull it low. Apparently the outputs of the latch are not equally loaded, causing an intermediate level of about 3 V instead of the expected 2.5 V. Although this level is within the normal specifications, it is still a clear fault indication! Also note that an extra change occurs on the address bus while ALE is high. This can be explained as follows: if the EPROM produced a '1' during the PSEN pulse, that level will remain on the bus even after the PSEN pulse is ended. The effect is caused by the pull-up resistors. As soon as the ALE line is made high, this '1' is copied to the address bus. At about half-way the ALE pulse, the processor copies the actual address information on to the bus. The latch conveys this information also, and furthermore retains it until after the ALE pulse has gone low again.

08 Remove the EPROM from the circuit, and replace it with one that contains nothing but NOP instructions (all zeroes). Run this 'program'. Starting the measurement on address line A0, the scope should indicate that each successive address line supplies half the frequency of the previous one. Measure the signals on the eight low-order address lines behind the latch. The high-order address lines are found on I/O port P2.

☞ **The signals must be fully cyclic (repetitive). Look at the way the data is copied by the latch on the ALE pulse. The variable address must be found on the data/address bus from about half-way the ALE pulse. It is followed by the stable NOP from the EPROM, which occurs during the PSEN pulse. Check the levels on the OE (output enable) and CE (chip enable) pins of the EPROM.**

09 If the test with the 'NOP' EPROM works, but the application program still fails to work properly (although you are sure that it is all right), it is time to start watching (external) interrupts. Unexpected or unidentified interrupts can make havoc of the normal operation of the processor, causing very complex faults. Also check if the EPROM is fed with the right number of address lines (jumper settings).

10 Unless you have access to a logic analyser, it is nearly impossible to trace the program flow with ordinary means. It is, however, possible to test the start of a program by connecting a square-wave generator to the reset input. That obviously requires the power-up reset capacitor to be removed temporarily. The circuit will start to work almost immediately (a few microseconds) after the reset input goes low. A stable oscilloscope picture can be obtained by triggering on the negative-going edge of the pulse. The first address to appear on the address bus must be 0000. The associated data (from the EPROM) can be verified with the aid of the scope (this is not easy, though). In this way you are able to trace the first few instructions. This method is sure to reveal serious hardware errors. If desired you can write short programs which do something that can be traced with the scope connected to an I/O port line.

The following little program outputs increasing values between 00 and FF to port P1:

```
                    1     ; test for 80451
                    2
                    3     ; equates
                    4     P1 equ 90H
                    5
0000                6     start:
0000    04          7     inc A
0001    F590        8     mov P1,A
0003    80FB        9     sjmp start
                   10
                   11
                   12
```

That nearly exhausts the techniques which may be used to locate faults in 8031-based systems. In practice, you should have a working circuit at this stage. If not, an elementary error exists in the design of the circuit. These cases are unfortunately beyond traditional faultfinding techniques as described here.

(940089)

Note: the oscilloscope used to produce the screendumps printed in this article is a LeCroy 9410 dual-channel 150-MHz type. Screendumps produced via HPGL file output option and subsequent EPS conversion.

APPLICATION NOTE

The content of this note is based on information received from manufacturers in the electrical and electronics industries, or their representatives, and does not imply practical experience by *Elektor Electronics* or its consultants.

OPTICAL WINDOW DISCRIMINATOR

By F. Hüber

When, in an electronic circuit, light is converted into electrical signals, use is normally made of photo diodes, photo transistors or light-dependent resistors (LDRs). These components translate the irradiance into analogue electrical quantities. When these converters are used for measurements on optical signals, the limits of current technology are reached in many cases. Now, new opto-electronic components with digital output, such as the Type TSL220 from Texas Instruments offer many new possibilities in this field. This device is an integrated light-to-frequency converter, the frequency of whose digital output depends on the quantity of light falling on to it. The IC, which is housed in a transparent DIL8 enclosure, operates from 5 V and draws a current of only 7–8 mA.

The TSL220 is used in an experimental setup in which the distance between it and a low-voltage incandescent bulb has to be measured accuractely. The distance is <100 mm and has to be determined to within 3 mm. The variation in illuminance over such short distances is very small indeed and can certainly not be detected by the human eye.

The TSL220 comprises a large photodiode and a complete current-to-frequency converter. The combination of these allows light to be converted directly into a digital signal of variable frequency.

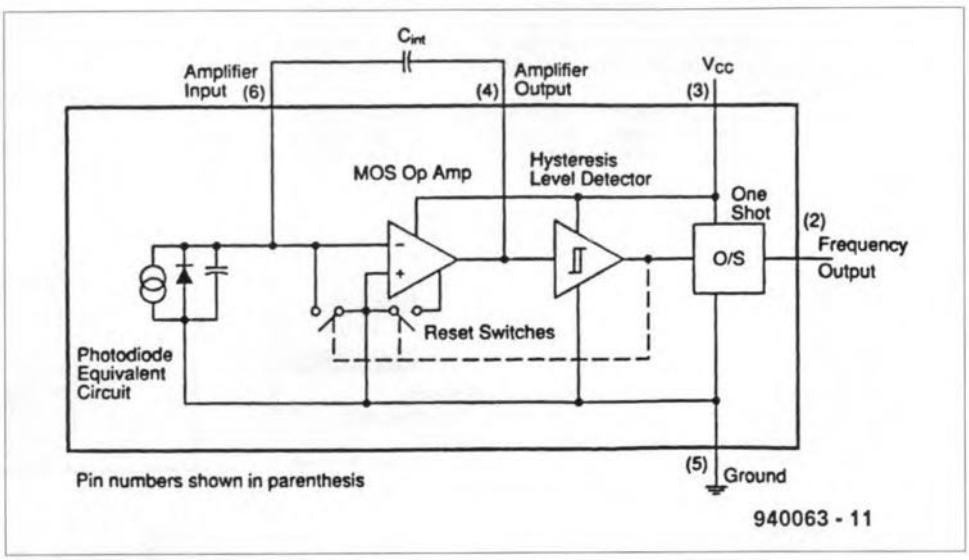


Fig. 1. Block schematic of Texas Instruments' TSL220.

The converter—see **Fig. 1**— consists of an integrator (which needs an external capacitor), a reset circuit, a Schmitt trigger level detector and a monostable multivibrator.

The integrator converts the current from the photodiode into a sawtooth-shaped signal. The slope of the sawtooth edges depends on the irradiance. When the sawtooth reaches a preset value, the level detector causes the capacitor to be discharged rapidly. At the same instant, the integrator returns to the start state and

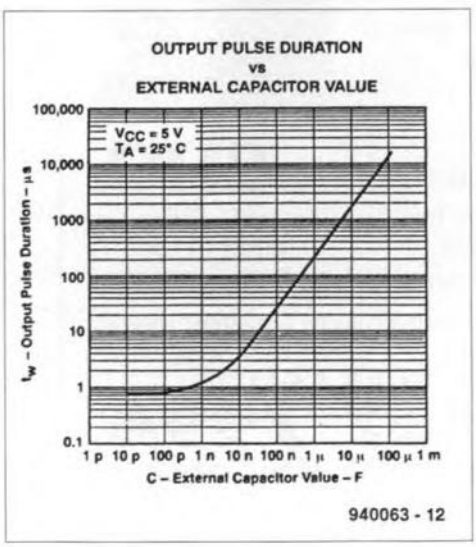


Fig. 2.

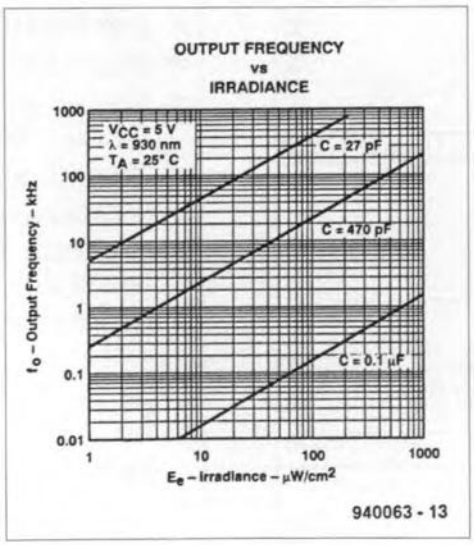


Fig. 3.

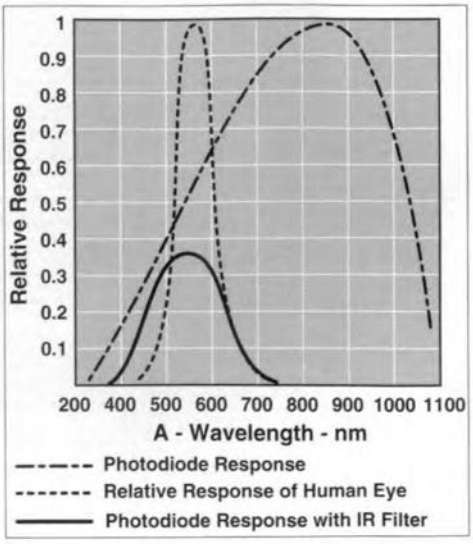


Fig. 4.

the monostable generates a short pulse. The width of that pulse depends on the value of the external capacitor. The relationship between the pulse and the capacitor is shown in **Fig. 2**. The frequency of the output signal is also determined by an external capacitor—see **Fig. 3**.

The output level of the TSL220 is compatible with all TTL and CMOS inputs. When a TTL input is driven, it is recommended to use a 3.3 kΩ resistor between pin 2 and earth. Maximum sensitivity of the sensor is at 850 nm—see **Fig. 4**. In many applications, it is a requirement that the sensor behaves like the human eye, the sensitivity of which is also shown in **Fig. 4**. When an infra-red (IR) filter is placed in front of the sensor, the sensitivity characteristic of the combination is similar to that of the human eye. A suitable filter is the Type CM500 from Hoya.

The sensor has a brightness range of 1:300 000, which makes registration of tiny variations possible. Because of the digital output, such tiny variations can be applied to another part of the circuit, even if this is at some distance. A corresponding analogue variation would be totally lost in the system noise. Moreover, an analogue system would be saturated when the irradiance is great. Nevertheless, even with the TSL220, the use of a grey filter in sunny summer daylight is recommended: it guarantees linear behaviour of the sensor in those bright circumstances.

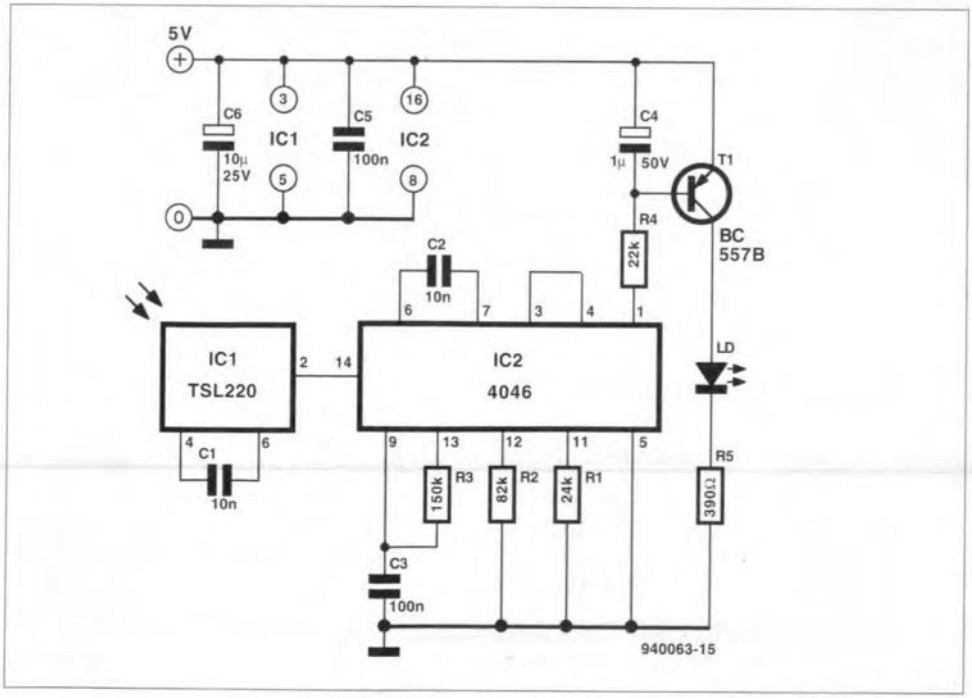


Fig. 5. Circuit for determining the distance between sensor and a source of light.

A practical circuit

A possible circuit to carry out the measurement mentioned in the second paragraph is shown in **Fig. 5**. The light of the incandescent bulb falls on to IC_1. With the specified value of 100 nF for C_1, the output frequency is 5000 Hz. With the desired mechanical variations, this varies from 4500 Hz to 5500 Hz, which the phase-locked loop (PLL) based on IC_2 has no difficulty in processing. The central frequency and the capture range of the PLL are determined by resistors R_1 and R_2. Resistor R_1 and capacitor C_2 determine the fundamental frequency at which the PLL oscillates in the absence of an input signal. Resistor R_2 determines the capture range.

The LED and transistor T_1 form a simple lock indicator. As long as the PLL is not locked, an irregular rectangular signal exists on pin 1. This rectangular signal charges C_4, whereupon T_1 begins to conduct, so that the LED lights. When the PLL is locked, the rectangular wave ceases and the level at pin 1 goes high, wherepon the LED goes out. The light source is then within the stated limits.

Note that the circuit works only if the lamp is powered by direct voltage. Alternating voltage can not be used, because the output signal of the TSL220 would then be frequency modulated and the PLL is not suitable for locking to such a signal.

[940063]

ELECTRONIC KNOW-HOW

Printed-circuit board production

By our editorial staff

WHEN the English edition of *Elektor* (as it was then called) was initiated nearly 20 years ago, it was probably the first European electronics hobby magazine to offer ready-made printed circuit boards to its readers. PCB were then designed by hand, and produced in a fairly primitive way. Today's production process consists of very advanced, high-tech, and almost completely automated, operations. The changes in the industrial PCB production process have occurred mainly as a result of large-scale automation, and, of course, by the strict regulations set up in respect of the environment.

This article describes the production process of a double-sided printed circuit board, from artwork to ready-made product. All PCBs supplied through the *Elektor Electronics* Readers Services are produced by Scaldian Electronics Works (SEW N.V.) in Hamme, Belgium, on the basis of artwork supplied by the *Elektor Electronics* design department.

To begin with, SEW is supplied with a PCB production order and the relevant Gerber files. These files are the end result of the PCB artwork design process, and the start of the actual, industrial, production. They contain all relevant information on the board size, the copper tracks, the solder mask, the component overlay and the drilling. Each side of the board has its own copper track file. The solder mask covers the entire board with the exception of the solder spots, and has a protective function. The component overlay contains all component symbols and other texts, and is printed on the component side. The drill file contains information on the position of the holes, and the drill size used to make each of these. A report file, finally, tells the PCB producer how many holes are drilled with a certain drill, and also indicates the drill diameter.

Inspection of the Gerber files supplied by the customer.

From file to board

The complete production process, from receipt of the production order and the Gerber files to the shipping of the boards to our central stores, normally takes about four weeks. Even for an 'ordinary' double-sided board, there are no fewer than 15 production steps. Multilayer boards have far more, and can be produced with up to 32 layers within a thickness of 3.2 mm.

The first production step involves making special films, which are transparent brown. These are not the same as the ordinary transparent films which may have been supplied by the customer, or produced on the basis of Gerber files. These films are archived. Since the brown films wear out during the production process, it is sometimes necessary to make new copies.

The standard material for *Elektor Electronics* PCBs is epoxy with a 17-µm copper layer. Other materials are available for special applications, while is it also possible to step up the copper thickness to 35 µm, 70 µm or 105 µm.

The first real production operation is drilling all holes which are through-plated or smaller than 3 mm. To speed up this operation, a number of boards are stacked and drilled in one go. Depending on the board material, two, four or six boards are stacked. The drilling machine works on three of these packages at the same time. The number of PCBs which is in production at the same time can be quite high because the entire process is based on large boards with a standard size, which accommodate a fair number of PCBs. For obvious reasons, this standard board size is used right up to the end of the production process. In the last instance, the PCBs are fraised out of the large board.

Automated drilling machine. A number of boards are stacked and drilled in one go.

A laminate is used to apply a photosensitive layer on to the printed circuit board.

Board materials

CEM1:	composite paper
CEM2:	composite paper
Fr2:	pertinax
Fr4:	epoxy
Fr5;	HF epoxy
Polyimide	
Teflon®	

Copper thickness

5, 17.5, 35, 70 or 105 µm

Metal layers

electrolytically applied:	
copper:	20 µm
tin/lead:	5 to 20 µm
nickel:	6 µm
gold:	1 µm
chemically applied:	
nickel:	4 µm
gold:	0.2 µm

After the drilling phase, the circuit board is laminated at both sides with a photosensitive film. The board sides are photographically exposed, and subsequently developed. The film remains in place in those places where the copper is to be etched away.

A chemical process is used to apply a thin copper layer (three to four micron) on the bare parts of the board. Next, this layer is thickened to about 20 µm by means of electrolysis. This results in holes being through-plated with a 20-µm thick copper layer, and the future copper track pattern 'growing' to a thickness of more than 35 µm. Once the copper has 'grown', the board is tinned. Tin deposits on all uncovered copper areas. During the etching phase, the tin layer is used to protect the copper layer (which must remain intact) against the aggressive etching solution. The tin layer is far more stable than a protective layer on the basis of film, and ensures a constant quality and thickness of the copper layer. Next, the film residue is removed from the tinned board. The future track pattern is then visible as a tin pattern on a copper background.

The board is etched using an ammonia water solution. The copper dissolved in the etching solution is recovered with the aid of a recycling process. Obviously, since many steps in the production are based on chemical substances and heavy metals, great attention must be paid to environmentally protective measures.

SEW makes use of piped water which is purified in an on-site processor (an investment worth over £300,000). Ordinary 'tap' water is not clean enough. Water is recovered at several stages in the production process, and fed through the cleaner so that it meets the (high) quality standards again. The waste water which is inevitably produced by the purifier is also cleaned to the level of ordinary tap water. The residue of the purifying process is recycled again, recovering the heavy metals.

Two options exist after the etching phase. In both cases, however, a visual inspection is carried out on the results of the etching operation. Etching errors then become apparent and can be corrected in a number of cases.

The first option available after etching is only suitable for PCBS intended for manual soldering. The tarnished tin layer on the board may be reflown with the aid of a heat source to produce a smooth, shining, tinned track pattern. The PCB is then given a 'perfect' look by applying the solder mask. Unfortunately, there is a great drawback to PCBs produced in this manner: they are not suitable for automated soldering techniques. The reason is fairly simple: if such a board is placed over a hot solder bath, the tin underneath the solder mask will melt, causing cracks in the solder mask. That spoils the look of the board, and reduces the protective operation of the solder mask. Since the boards supplied through the *Elektor Electronics* Readers Services are intended for manual soldering only, the reflow technique may be applied without prob-

Silk screen printing is used to put the solder mask on the PCB.

This machine applies the solder mask to the printed circuit board.

lems to give the board a professional 'finishing touch'. Only boards which require local gold-plating to be applied later (such as our PC insertion cards) have to go through a special production process.

The other 'route' is identical to the one taken by boards intended for automated soldering. On these boards, the tin layer is completely removed after the etching operation. This results in a clean copper surface. Depending on the production volume, either a photosensitive layer or a silk screen is applied to make the solder mask. Low-volume series generally use photolacquer. With larger series, it may be more economical to apply silk screen printing using a two-component lacquer. After the photographic exposure phase, the board has a tough protective layer, the solder mask, with clearances around the solder spots.

The PCB is then ready for selective tinning, which is achieved by immersing it in a bath of liquid tin. While the board is hoisted out of the bath, hot air is used to blow the tin out of the holes. The bath deposits a thin tin layer on all bare copper surfaces on the board. The result is a perfectly tinned PCB.

Gold and silver plating

It is sometimes necessary to cover parts of a circuit board with a silver or gold layer. Unfortunately, selective silver-plating or gold-plating is not possible on PCBs with a reflow-finish. This means that the tin layer must be removed after etching, if a PCB is to be gold-plated or silver-plated.

If the PCB has areas which are to be selectively gold or silver plated, these are covered with tape during the selective tinning phase. The tape is subsequently removed, and new tape is applied on the tinned areas. Next, a 0.4-µm to 0.5-µm thick nickel layer is applied on the bare areas. This is done to improve the adhesion of the gold layer, which has a thickness of 0.2 µm. The advantage of a gold layer is that it is free from corrosion, and generally better up to wear and tear than a tin or copper layer.

The final stage comprises applying the component overlay on the solder mask at the component side of the board. This is done in a silk screen printing operation.

During the second drill operation, all holes are drilled which do not have to be through-plated, and are larger than 3 mm. Also, slots are fraised in the board.

That completes the production process. Before they leave the SEW works, all PCBs are electrically and visually tested. The electrical testing is fully automatic, and uses a pin matrix which tests all contacts on the board. During this test procedure, the behaviour of the board is compared to one that has been approved. Next, the visual inspection is done. If the board looks all right, it is ready for packaging.

(940043)

Electrical end check.