Australia 95c New Zealand 95c South Africa 90c Malaysia $2.75

PRACTICAL ELECTRONICS

DECEMBER 1979

55p

Seiko WATCH OFFER!

...Saves you £45...Full Details Inside

Ultrasonic BURGLAR ALARM

SPECIAL Supplement... SOLID STATE CAR INSTRUMENTS

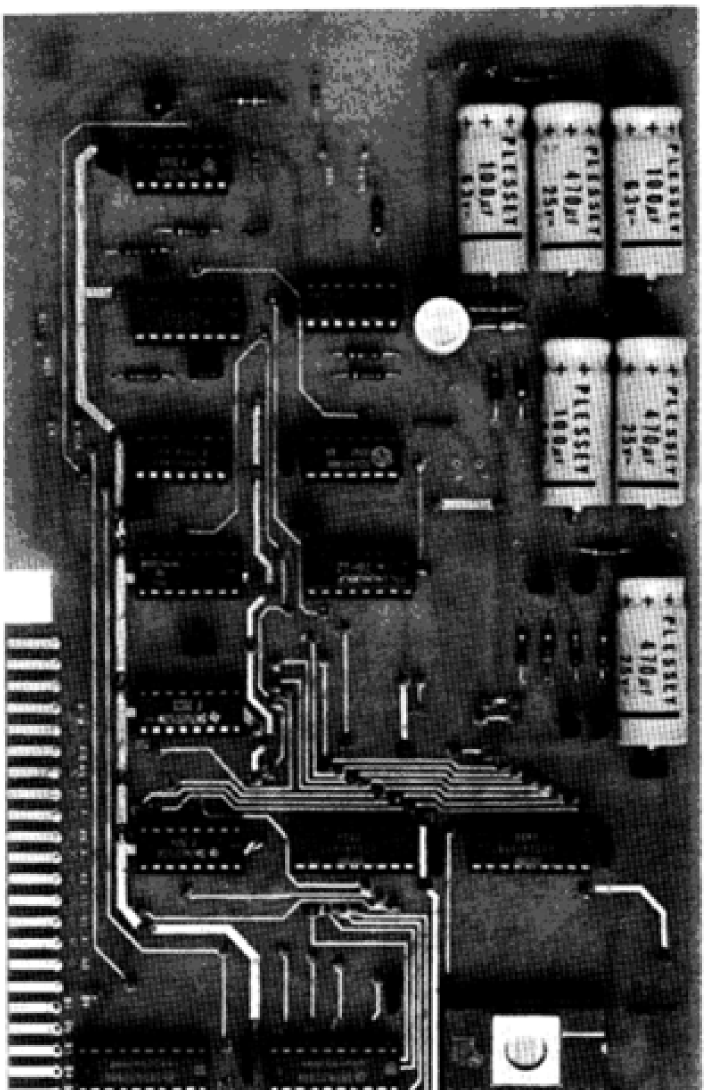
EPROM PROGRAMMER
PART 1

A.A. BERK B.Sc. Ph.D.

THE EPROM programmer is a memory-mapped peripheral for the 2708 family of Erasable Programmable Read Only Memories (EPROM's) and programs at about the maximum possible speed (around 2 minutes). The p.c.b. contains its own power supply giving +12V, −5V, +27V, and only requires +5V and a 9–0–9 volt transformer. On board, there is 1K of RAM for storing information to be transferred to the EPROM. The EPROM may be read before or after programming for verification, and used *in-situ* as a normal 1K block of Read Only Memory. The programmer thus effectively expands the host system by 2K of memory (1K of EPROM and 1K of RAM), with the added facility of being able to reprogram the EPROM section. Connection of the machine is simple and almost identical to the *PE* VDU connections.

Eleven Address Bus Lines (A0–A10), eight Data Bus Lines (D0–D7), a Read/Write line and some address decoding logic is all that is required. Interface to the COMPUKIT UK 101 and many other machines requires just a couple of d.i.l. plugs. The COMPUKIT would allow the necessary routines to be written in BASIC. A minimal MPU system containing the machine is shown in Fig. 1.

Use of the programmer is restricted to writing the desired target program or data into the 1K of on-board RAM, running and checking it, and then switching on the programmer which then takes around two minutes to copy the RAM contents onto EPROM. The machine automatically hands EPROM and RAM back to the MPU system.



EPROM THEORY

Each individual cell within the 2708 contains a "floating gate", which is able to collect a charge from a "pumping" pulse-train produced by a programmer. When sufficient charge has been accumulated within this completely insulated storage element, the cell, initially showing a "one", will return a "zero" along its data line when read.

The 2708 family members have a quartz window in the upper surfaces of their packages. Through this, a strong ultra-violet light may penetrate to knock the electrons in the storage elements out of their shallow energy levels and allow them to leak away via the substrate.

This clears the cell back to a "one". The organisation of the 2708 causes 8 cells (one byte) to be read and programmed at a time, hence requiring normal Address and Data lines.

To program the EPROM, the programmer produces a succession of addresses along A0–A9, and, for each, presents the data to be programmed along D0–D7. The EPROM thus starts, after erasure, with FF (8 "ones") in each location.

The requirements of the 2708 family demand that *all* locations be presented with data a large number of times during the programming sequence. The number of such complete addressing cycles is determined by the speed of operation.

The Block and Connections diagram shows a typical Read Only Memory device with the addition of a Programming pin. This pin is pulsed up to +27 volts during programming.

The $\overline{CS}$/WE pin is a normal chip select except when programming, when it is held at +12 volts. The timing Diagram (Fig. 3), shows the signals necessary to program the 2708. It should be emphasised that the programming cycle consists of cycling through all 1024 memory locations N times and presenting the data to be stored over and over again. The sequence is as follows. $\overline{CS}$/WE is switched to +12V, the first address and data are presented and >10µS later, pin 18 is pulsed up to +27V. These conditions are held for 0·1 to 1ms (0·5ms is chosen for the machine presented here) and then pin 18 drops to zero and the next address and data loaded. The cycle from address 0 to address 1023 thus takes a little more than half a second.

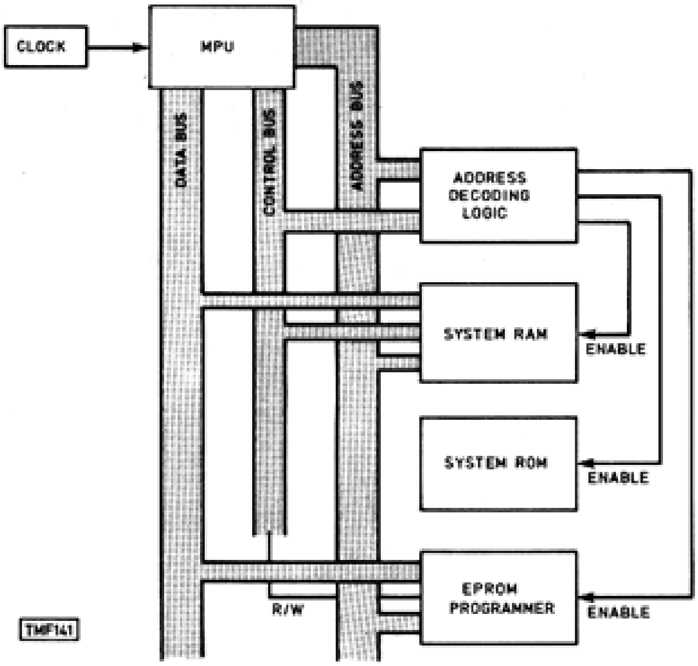
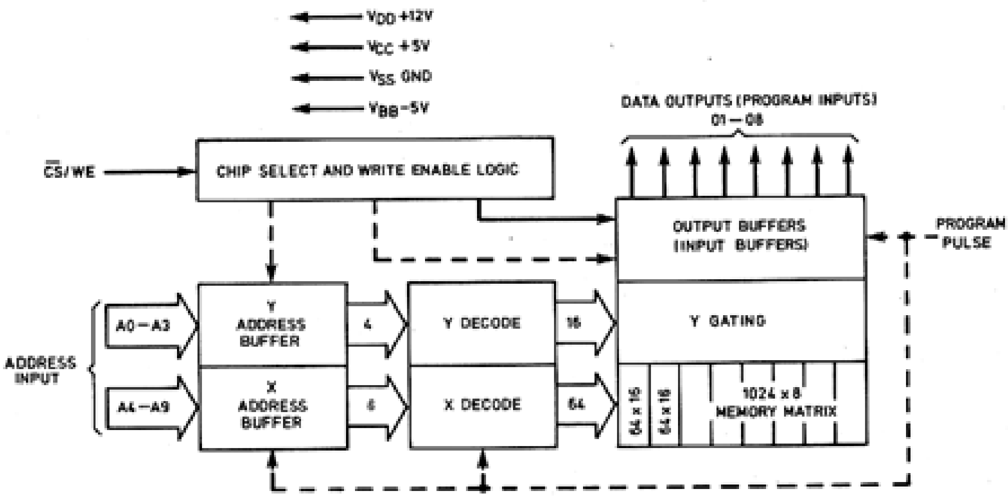


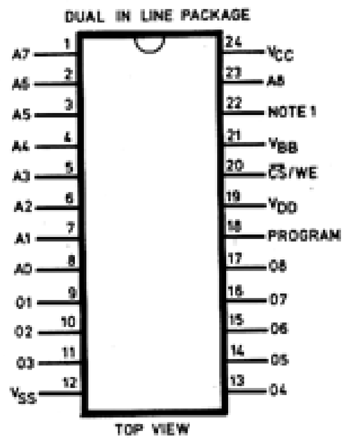
Fig. 1. Minimal system containing EPROM Programmer

This entire sequence is then repeated a number of times. The total number, N, of such loops is between 100 and 1000 (possibly more) depending upon the width (tp) of the program pulse (+27 volts). The formula for determining N is: $N \times tp > 100ms$. Thus, for 0·5ms, N is a minimum of 200. The total programming time for this number is slightly over 100 seconds.

The EPROM starts with its locations containing FF, thus if the programmer presents just FFs to the EPROM, it will retain its "erased" state. Even though each memory location must be accessed during programming (many times in fact), those presented with FF will remain untouched. Thus, a block of the EPROM may be programmed selectively without affecting the rest of the device. In fact, this is true of an EPROM already containing any other block of information. Contents of that block, whatever they may be, remain unaffected by the programming procedure as long as FF is presented to each of its locations during the cycles.

Indeed, a single bit in any given location can be set to zero selectively by presenting 1's everywhere else during programming.





PIN CONNECTION DURING READ OR PROGRAM

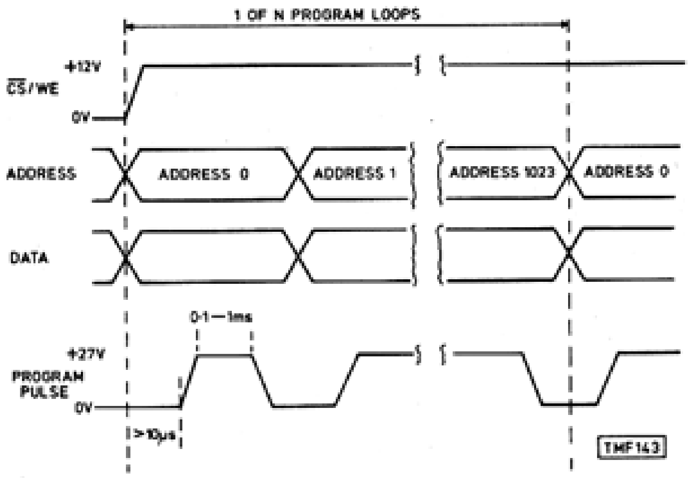
MODE	PIN NUMBER						
	9—11, 13—17	12	18	19	20	21	24
READ	D_{OUT}	V_{SS}	V_{SS}	V_{DD}	0V	V_{BB}	V_{CC}
PROGRAM	D_{IN}	V_{SS}	PULSED 27V	V_{DD}	V_{DD}	V_{BB}	V_{CC}

PIN DESCRIPTION
A0—A9 ADDRESS INPUTS
01—08 DATA OUTPUTS
$\overline{CS}$/WE CHIP SELECT/WRITE ENABLE INPUT

TMF142

Fig. 2. The 2708 EPROM

Fig. 3. Program timing diagram



It should be noted that the machine described here is *not* suited to the *single supply* 2716, nor types of EPROM other than 2704, 2708 and the multi-supply 2716. The reason for this is that the cycle-timing diagram requirements are quite unique to this family.

Fig. 4 shows the hardware set-up of the programmer. The heart of the device is in the timing control block. It is here that the cycles of sequentially presented addresses are produced, as well as the correct repetition rate and number of complete loops necessary.

The counter-produced addresses are switched over to the RAM and EPROM by the Address Bus switch after a Program Request signal is received. A Ready line, and l.e.d. on the p.c.b., signals when the programming cycle is in progress.

The RAM is held in the READ state, and the Counter addresses RAM and EPROM sequentially. The RAM (which holds the desired program) places the data on the Data Bus and hence onto the input pins of the EPROM. The timing control then waits a few microseconds and switches the 27 volt level on to the Program pin of the EPROM for 0·5ms.

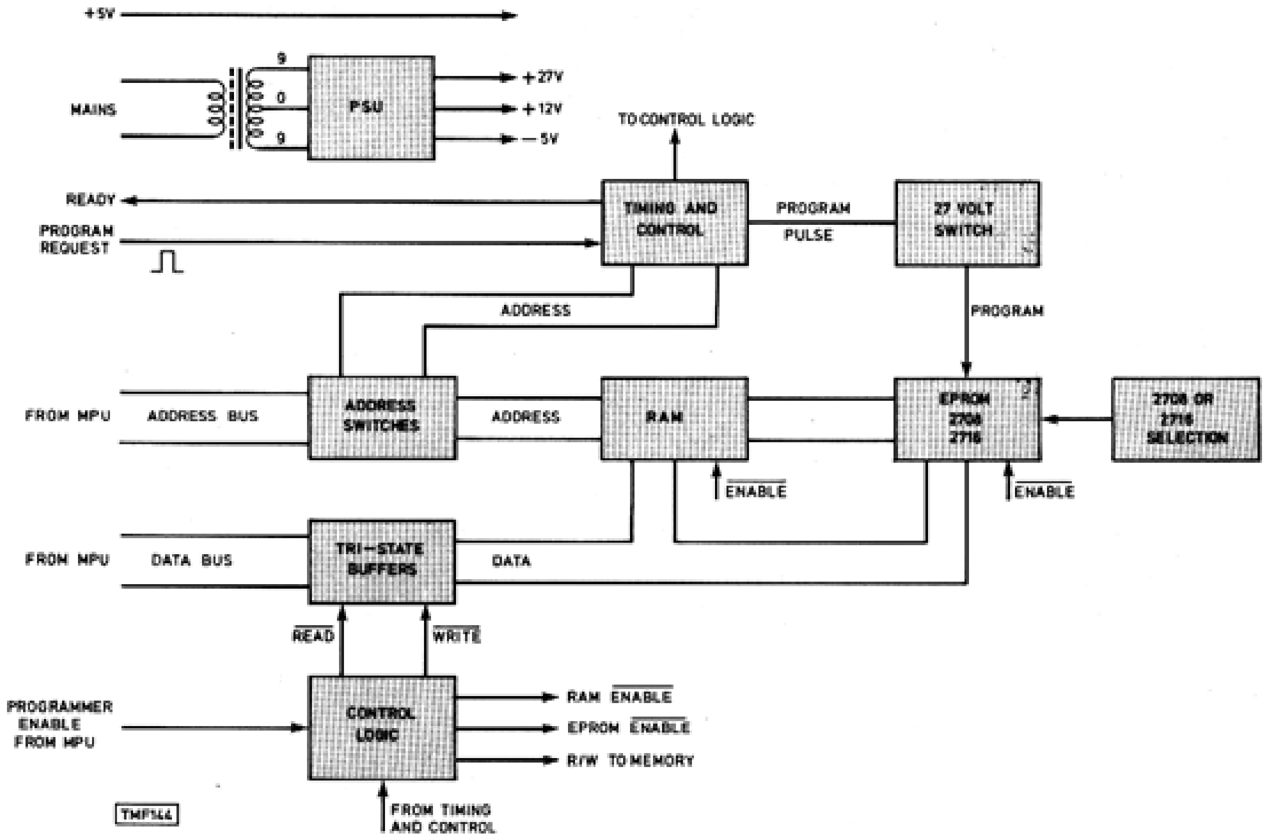


Fig. 4. Block diagram of EPROM Programmer

COMPONENTS . . .

Resistors

R1	1k ½W
R2	1k
R3	47
R4	150
R5	100
R6, R9	2k2 (2 off)
R7	15k
R8	8k2
R10, R12	10k (2 off)
R11	33k

All resistors ¼W 5% unless othewwise stated.

Capacitors

C1, C2, C4	470µF/25V elect (3 off)
C3, C5, C6	100µF/63V elect (3 off)
C7	1n ceramic
C8	100n ceramic
C9	10n ceramic
C10	33µ elect
C11 upwards	100n Supply decoupling

Transistors and Diodes

D1–D4	1N4004 (4 off)
D5, D6	1N914 (2 off)
D7	27V/1W Zener
D8	Any l.e.d.
TR1, TR3	BC549 (2 off)
TR2	BC559

Integrated Circuits

IC1	7474
IC2	4024
IC3	4040
IC4	2708 or 2716 EPROM
IC5, IC6	2114 (2 off)
IC7, IC14	74LS244 (2 off)
IC8, IC13	7400 (2 off)
IC9	74123
IC10–IC12	74LS157 (3 off)
IC13	79L05
IC14	78L12

Sockets

14 pin	4 off
16 pin	5 off
18 pin	2 off
20 pin	2 off
24 pin	1 off

Miscellaneous

T1 Mains transformer. Sec. 9–0–9V at 1A
S1 Double pole single throw
Pins for through-board connection
Double sided 0·156in. pitch 50-way edge socket
Printed circuit board

Constructor's Note

Kit of parts, including double-sided, drilled and tinned p.c.b. (not plated-through), is available from: **Modus Systems Ltd., Dept. EP, 29a East Cheap, Letchworth, Herts. SG6 3DA. £37·30 VAT + p.p. inc. P.c.b. £8·45 inc.**

The kit excludes T1, S1, IC4 and edge socket, but sockets are included for all i.c.s.

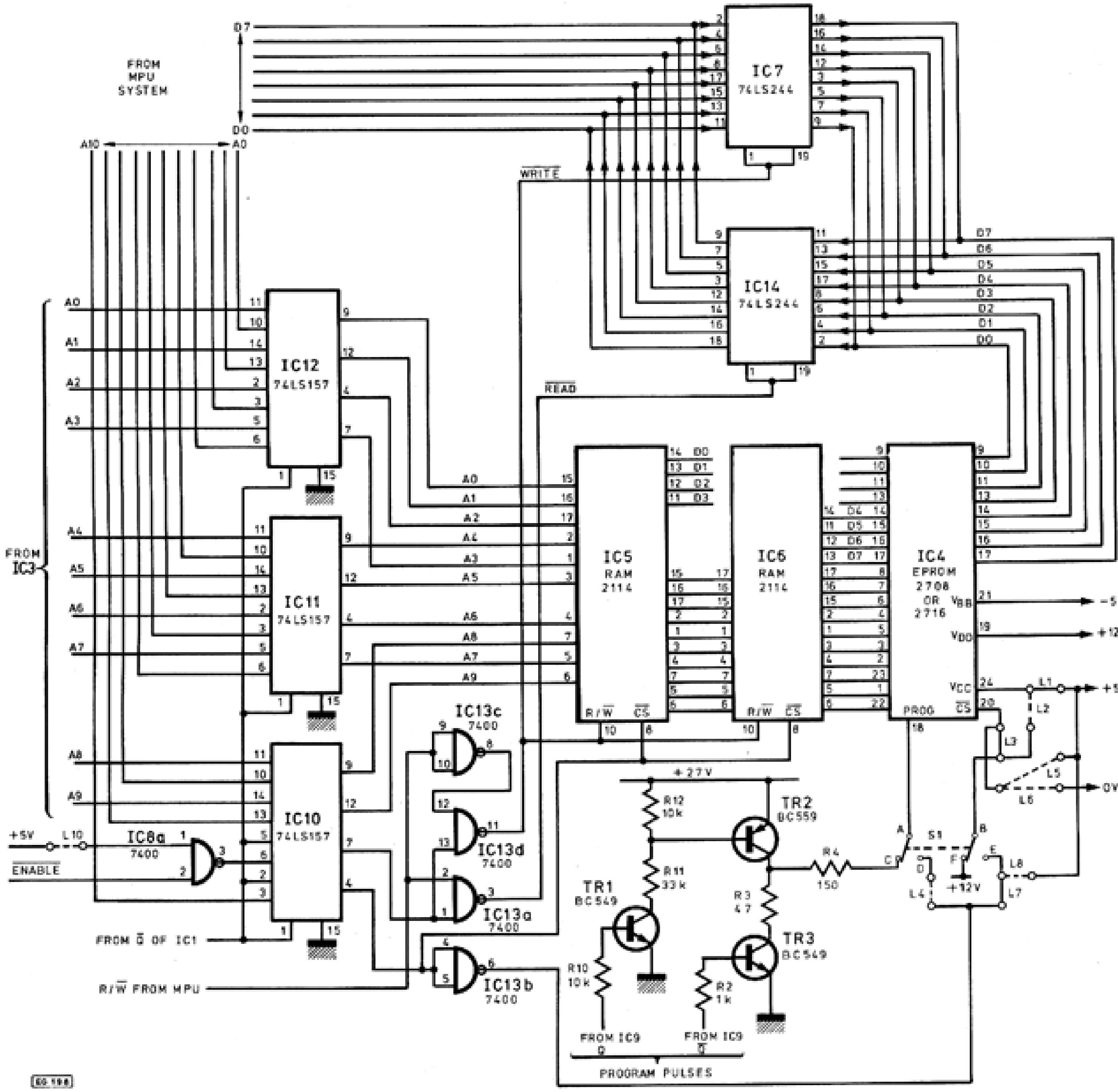


Fig. 5. Memory and switching of the Programmer. S1 is in the "Program" position. Links are shown connected for 2708

The cycles continue until complete and the EPROM and RAM are then handed back to the MPU system for normal use.

It should be clear then, that the role of the MPU system hosting the process is to provide for the RAM, programs or data to be "burned" into EPROM. To this effect, the RAM should be filled with FFs beforehand, so that any unused RAM space does not affect the EPROM.

The RAM and EPROM form a normal part of the MPU's address map, such that programs and data may be stored in either or both for running and use *in-situ*. The only time the memory is not available to the MPU is *during* programming. When the tri-state buffers are switched off, the data Bus of the MPU system is disconnected from the machine.

Last month, in the final part of the COMPUKIT UK 101 article, "plug-in" methods of expanding the machine utilising the upper part of the 2114 memory were described. This is the most convenient method of attaching a programmer and is described next month.

POWER SUPPLY

The Power Supply Unit (Fig. 6) gives all necessary power levels except for the +5V which is usually available from the MPU system. An external transformer with a secondary of 9–0–9 volts at 1 Amp is necessary for the PSU and may be purchased from Modus Systems Ltd., who can also supply the p.c.b. and full kit.

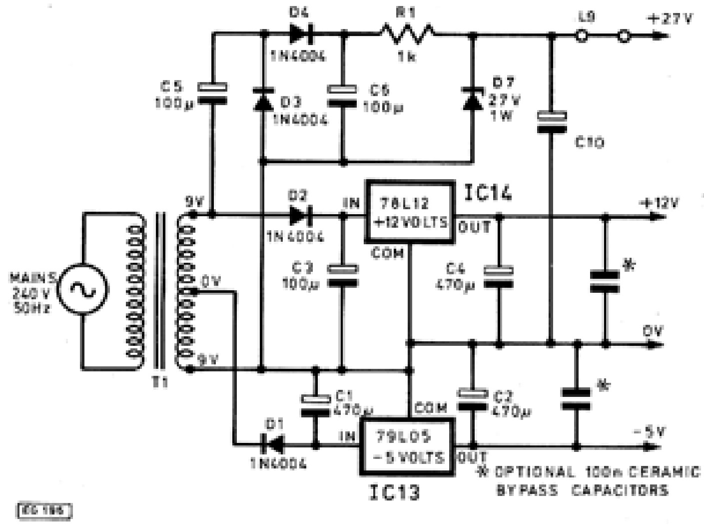


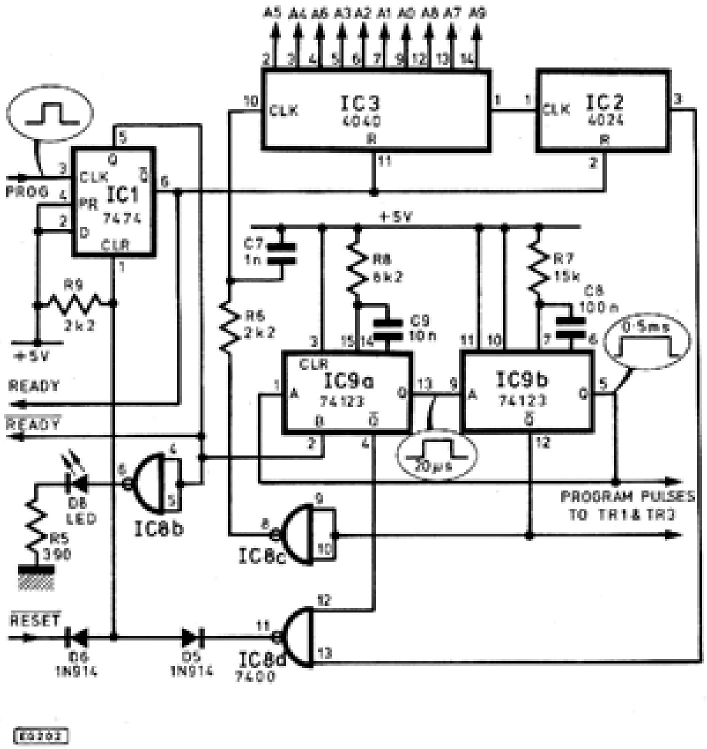
Fig. 6. Power supply

The full 18 volts is halfwave rectified to supply a regulator for the 12 volt level, and a 9 volt winding is used for the −5 volt regulator. Plenty of power supply bypassing is included on the p.c.b. and little rippling was detected on the prototype.

The 27 volt supply is formed by voltage doubling the 18 volts via D3 and C5. The unregulated voltage developed across C6 is held at 27 volts by a Zener diode.

A note of caution is applicable here. For safety, the link L9 should only be connected for programming, since the voltage is quite sufficient to destroy all the i.c.s on the board if accidentally bridged to nearby tracks. In addition, power-up could place the programmer in the program mode, and this would destroy the contents of an EPROM if in place. A reset line is provided to stop the programming cycle at any time, and a power-up reset could be automatically applied by a capacitor and resistor in series if required. +5V, −5V and +12V are all necessary for normal operation of the EPROM and none should be applied to the EPROM for any length of time without the others.

Fig. 7. Program timing and control of the programmer



The 27 volt pulse, in particular, must *never* be applied unless the others are present exactly as required in the timing diagram, and certainly should never be applied for more than 1ms continuously.

PROGRAM TIMING AND CONTROL

The circuit diagram of Fig. 7 shows the program timing and address counter elements. The user (or his MPU system) switches the Program Input (pin 3 of IC1) to a "one". This latches IC1 on, until its Clear (pin 1) is set to Zero.

Q and $\overline{Q}$ of IC1 are set to "1" and "0" respectively, removing the reset level from IC2 and 3 and starting IC9a timing. 20µs later, Q of IC9a falls and starts IC9b timing via its "A" input, which is sensitive to falling edges. Program pulses are sent out via Q and $\overline{Q}$ to Q1 and Q3 which are part of the 27 volt switch. The 27 volt program pulse then holds for 0·5ms.

Throughout the process so far, the outputs of IC3 have remained in their reset state of all zeros. This address is presented to the RAM and EPROM to Program Address 0 in the EPROM. When the 0·5ms ends, a falling edge triggers off IC9a again via its A input and a short time later, via IC8c, R6 and C7, the CLOCK input of IC3 receives its first high-to-low transition. This causes an advance. The delay allows the EPROM to settle fully out of the "program" state. This new address is then allowed to settle before IC9a's Q output falls and programming of the next address occurs.

IC3 is a 12-bit counter and pin 1 is its twelfth bit. Thus, the addresses are cycled through four times before pin 1 suffers a high-to-low transition which advances IC2 by one count. This chip is a 7-bit counter and pin 3 is its seventh bit. Thus the addresses are cycled through $4 \times 2^6 = 256$ times before pin 3 goes high which sets IC8d pin 13 to a "1". As soon as IC9a has finished timing, pin 12 of IC8d is set to a "1", clearing IC1, and the system returns to its quiescent state. The theoretical time, assuming accurate pulse lengths, is thus $(520\mu s) \times 2^{12} \times 2^6 = 136$ seconds.

A certain amount of leeway in timing has been included to offset variations in programmer components and EPROM manufacture.

USE OF THE MACHINE

The software to run the programmer should be in the machine already. Any MPU system must have routines for accepting information from the user, usually via a keyboard, and placing it in any memory location desired. By this means, the RAM block may be filled with the necessary data by hand. The only software necessary will be a small routine to fill the RAM with FFs if the whole EPROM is not programmed. Verification of EPROM contents is, again, performed via the computer's normal system of reading out data from any selected memory location, by simply reading through the appropriate EPROM addresses, as with any block of memory in your machine.

A short flow-chart to fill the RAM with FFs is presented in Fig. 8. This can be written for any simple machine-code based system and requires either one testable 16-bit register for scratchpad, or two 8-bit locations to store an address.

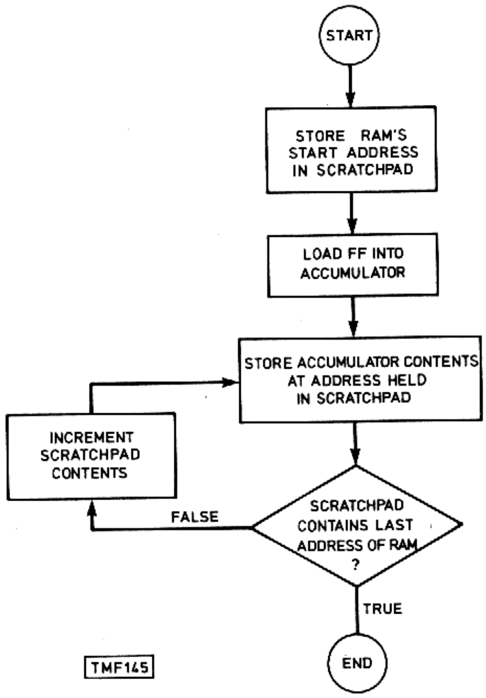
The flow-chart will code into instructions more easily on some machines than others. For instance, 6800 and Z80 both contain 16-bit registers and may be used with ease to discover when the last address in RAM has been set. SC/MP and 6502 do not have this facility and checking sixteen bits of information is a more involved job. However, any machine is capable of performing the routine, and provided there is a bus-structure available, interfacing in hardware and software will be quite straightforward.

As a final note on the above, it is possible to perform the setting of RAM to FF by hardware. All the facilities for stepping through memory exist on the programmer, and it is just a question of keeping the EPROM disabled, the RAM in the write-state, the Bus buffers tri-state and all Data lines to a logic "one". This will fill the RAM with 1s as IC3 cycles through once.

Next month, the final part covers the rest of the hardware, p.c.b. layout, construction and hardware interfacing to your system.

The EPROM Programmer may be run from the UK101

Fig. 8. Flow diagram to fill the RAM with FFs



EPROM PROGRAMMER

PART 2

A.A. BERK B.Sc. Ph.D.

THE EPROM programmer is a straightforward device to add to any Bus-oriented MPU system, and this month an interface is described for the Compukit in particular, and other machines in general. A particularly simple machine to interface would be the KIM, for instance, since its "K" outputs provide 1K blocks of address decoding automatically. The only other connections are Data Bus, Address Bus and a single Read/Write line—again brought out to convenient external connections on the KIM.

Also, the rest of the hardware of the programmer is described along with Construction, Setting Up and Troubleshooting.

MEMORY AND SWITCHING

As explained last month, IC3 provides the cycling addresses for RAM and EPROM while programming. During this time, the RAM places data to be "burned" on to the programmer's internal Data Bus. It is essential that none of this is allowed on to the MPU's Data Bus to cause a conflict. Tri-State buffers (IC7 & 14) perform this isolation when their $\overline{\text{ENABLE}}$ lines are held at a logic 1 by IC13 as explained below.

In addition, the MPU Address Bus must be disconnected from the EPROM and RAM while IC3 is cycling, and IC3's counter addresses must be allowed through. This function is performed by the selectors IC10, 11, and 12 when their pin 1's are held at a logic 0 by IC1. A zero from IC1 is also allowed through to pins 1, 4, 5, and 13 of IC13 during programming. This holds pins 11, 3 and 6 at logic 1. Pin 11 holds IC7 in its Tri-State mode and the RAM in its "READ" state to allow data out of the RAM. Pin 3 of IC13 holds IC14 Tri-State and pin 6 is unused in the program mode. The R/W line from the MPU is also ineffective during programming as IC13a and IC13d are held with their outputs "high" by the above. Finally, the Chip Select pins of the RAM are held "low" by IC1's $\overline{\text{Q}}$ output via IC10.

While the machine is programming, IC9b produces 5 volt programming pulses to be converted to 27 volts for the EPROM. TR1, TR2 and TR3 perform this function. While Q and $\overline{\text{Q}}$ of IC9b are at 1 and 0 respectively, TR1 is *on* forcing TR2 *on* and TR3 is *off*. This connects the program pin of IC4 (pin 18) to the 27 volt supply via R4, and TR2's "on" resistance. When the opposite condition holds, TR3 turns *on* and TR1 and TR2 *off* forcing pin 18 of IC4 to Ground through R4, R3 and TR3's *on* resistance.

Using p.n.p. and n.p.n. transistors (TR2 and TR3) in this mode ensures that almost all of the current supplied by the 27 volt supply is used by IC4 and very little sunk to Ground. At no time (theoretically) is the +27 volt supply connected to Ground via the output transistors. The only consistent path to Earth is via TR1, which, with R12 and R11 in series, will draw something of the order of half a milliamp from the 27 volt supply. This is, of course, apart from the normal leakage of TR2 and TR3 when *off*, and during the fast change over from *off* to *on*, which is at relatively low repetition rate.

EPROM REQUIREMENTS

The 2708 specifications demand that, during programming, $\overline{\text{CS}}$ (pin 20) be held at +12 volts. Thus, switch S1 is connected to supply this as well as the 27 volt program pulse to IC4. The difference between 2708 and 2716 are catered for by links L1-L8. The 2716 must receive +12 volts on pin 24 during programming instead of +5 volts, therefore L1 and L3 are replaced by L2. L7 is replaced by L8 so that when S1 is switched out of program mode, +5 volts is again applied to pin 24. Pin 20 is A10 and not $\overline{\text{CS}}$ on the 2716, thus links L5 and L6 are used to apply a logic 1 or 0 to this pin, to select one or other 1K half of the 2716. Pin 18 doubles as the $\overline{\text{CS}}$ and Program pin, and hence, out of the program mode, linking L4 supplies the $\overline{\text{ENABLE}}$ signal from IC13b.

Thus, during Program mode, the machine is disconnected from all external systems. Addresses are generated internally by the counters and fed to the EPROM and RAM. The latter, being in Read mode, places data to be programmed on the internal Data Bus which is collected by the EPROM and "burned in" over a number of progam pulses.

When IC1 is reset, the machine re-enters its normal Read/Write mode. $\overline{\text{Q}}$ of IC15 is at a "1" and hence pins 1 of IC10, 11 and 12 select the external Address Bus which is then allowed to communicate with the RAM and EPROM.

The $\overline{\text{ENABLE}}$ line (pin 2 of IC8) must be low to select the EPROM programmer, and the host microcomputer generates this signal as a normal part of its address decoding. The nature of the decoding decides which locations in memory the EPROM programmer occupies. When it is selected, the $\overline{\text{ENABLE}}$ must go "low" which, through IC8a and IC10, sets a logic 1 on pins 1 and 13 of IC13. This allows the R/W from the MPU system to gate a zero on $\overline{\text{WRITE}}$ or $\overline{\text{READ}}$ (to IC7 or IC14) depending upon whether it is performing a write or read operation respectively. During $\overline{\text{READ}}$, R/W is "high" which, via IC13, places a zero on $\overline{\text{WRITE}}$. This controls data direction through IC7 and IC14. If the RAM is to be selected, either the MPU system or the user must place a zero on the external A10 line—this becomes a zero on $\overline{\text{CS}}$ to the RAM.

To select the EPROM, a logic one on A10 becomes zero (via IC13b) on the $\overline{\text{CS}}$ pin of IC4 as long as S1 is in the non-program position. The MPU system may thus use the RAM and EPROM as normal blocks of memory. Programs may be written to, and run, in either.

LED INDICATOR

While programming, the l.e.d. on IC8 is off, giving a positive indication of the state of the machine. If the l.e.d. is off after power up or at any time, a zero may be applied to the $\overline{\text{RESET}}$ line via IC1 and the programming cycle stops. A READY (and $\overline{\text{READY}}$) signal is generated by the EPROM programmer and may be used by the MPU system to determine when the programmer is "BUSY". The positive-going "Program Initiate Pulse", via IC1, may also be generated by the MPU system, or a push switch operated by the user.

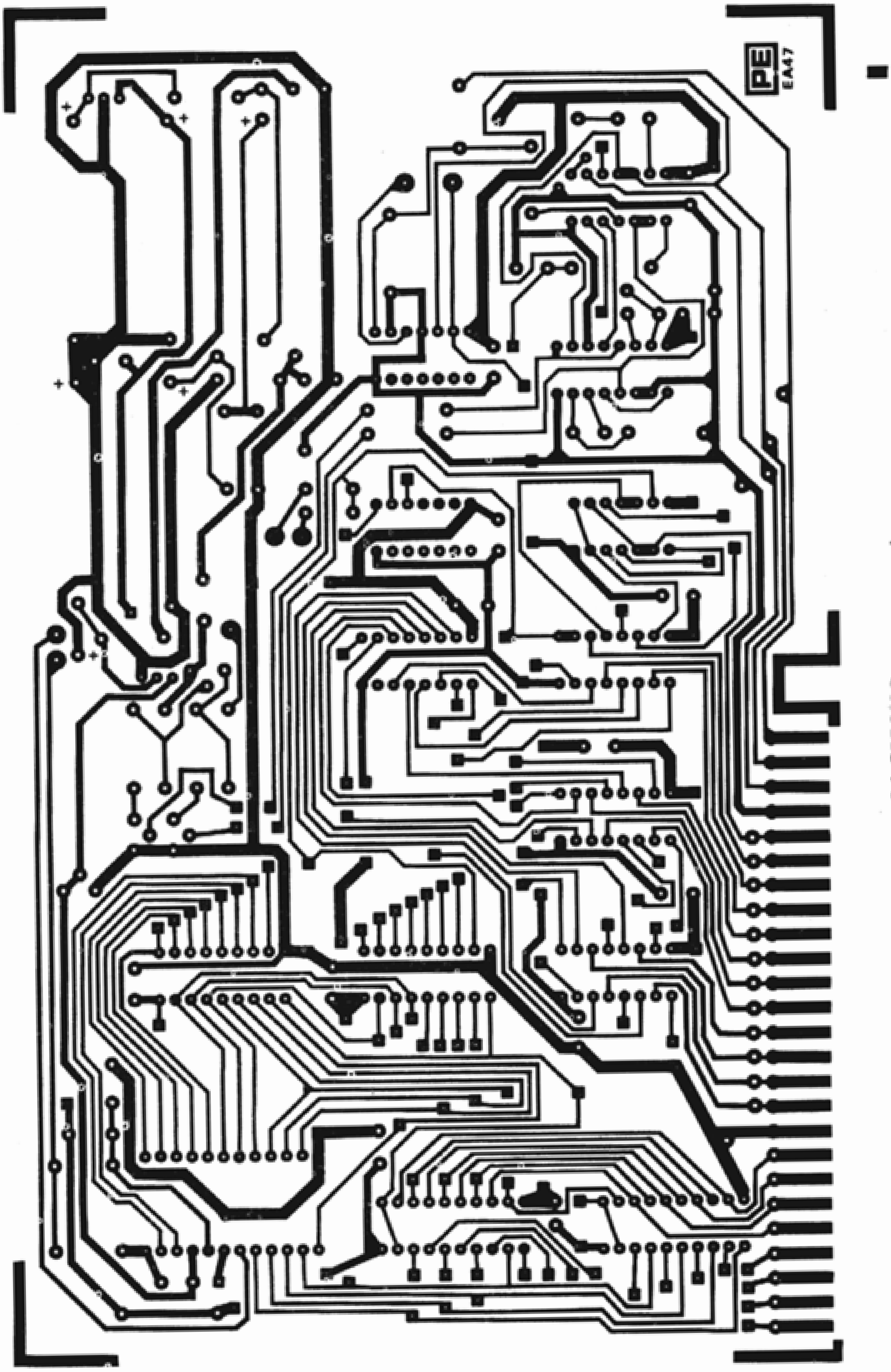


Fig. 2.1. EPROM Programmer p.c.b.
(Copyright of Modus Systems Ltd.)

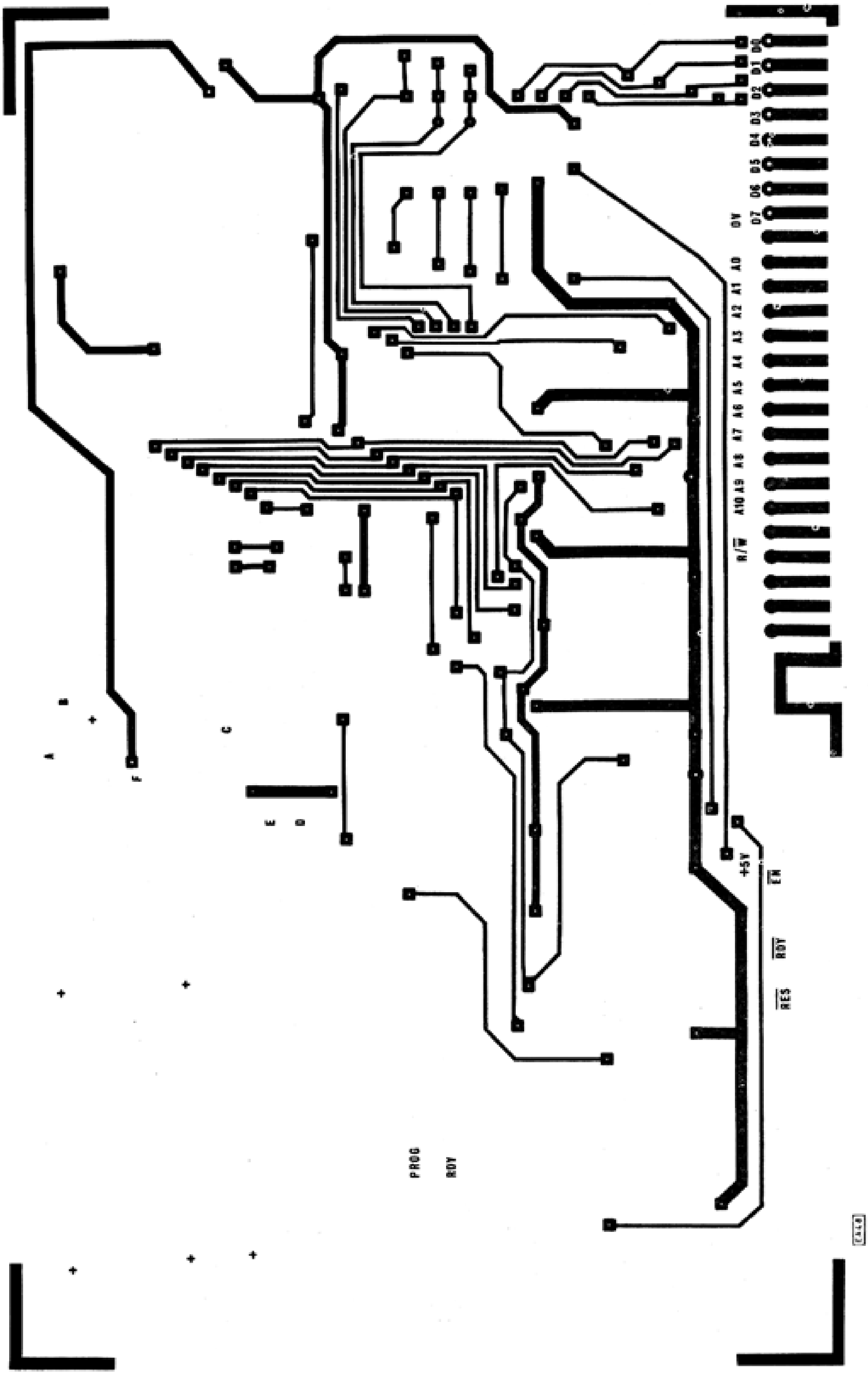


Fig. 2.2 EPROM Programmer p.c.b. (component side)
(Copyright of Modus Systems Ltd.)

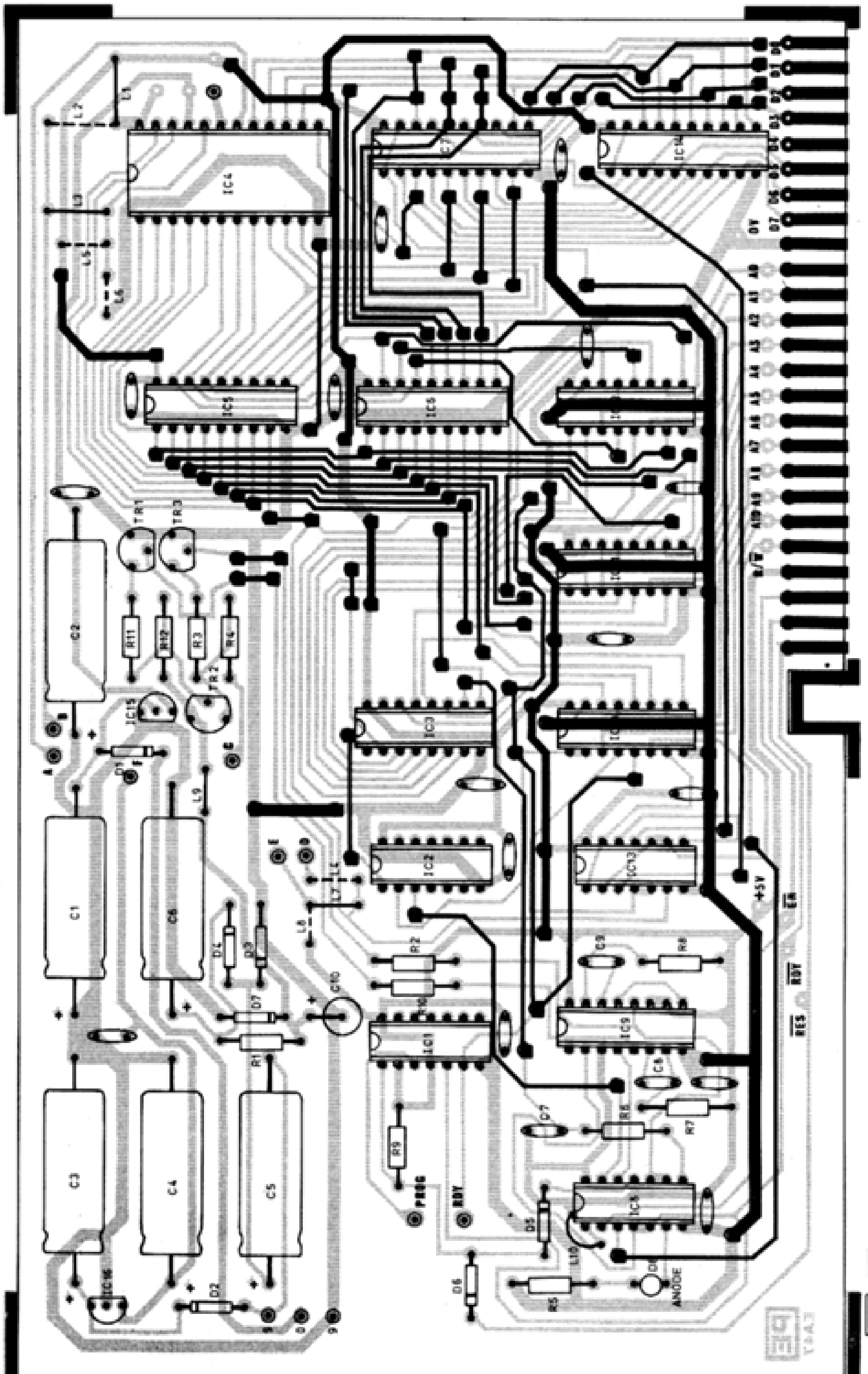


Fig. 2.3. Component layout. Note: R5 is 390Ω, and S1 is a d.p.d.t. Last month's components list is incorrect on these two points

CONSTRUCTION

The machine has been designed for ease of construction. All connections are made to the bottom of the board, except for through-pins, and the odd discrete component whose lead may act as a through-pin. All i.c.s point the same way and all external connections are labelled on the p.c.b.'s upper surface. See Figs. 2.1, 2.2 and 2.3.

External wire links appear on the p.c.b. only insofar as they provide flexibility of options as described in the text, and they may well be replaced by switches for convenience.

A good plan is to through-pin the board first. In general pins will not push right through the board and they should not be forced to do so. Push enough of the pin through to connect the two sides of the p.c.b., solder both sides and clip off the excess pin length. This produces a very neat aspect to the board. The exception is for the externally connected pins which may be left unclipped to facilitate solder or crocodile clip connection.

The i.c. sockets should be inserted and soldered next, followed by the discrete components. Make sure that regulators, diodes, transistors and capacitors are connected up correctly as shown in the component overlay diagram, or permanent damage will result. The correct positioning of transistors and i.c. regulators is indicated on the p.c.b.'s upper surface. Finally, insert the i.c.'s and turn to the set-up and testing section.

INTERFACE

Interfacing is a question of physically connecting wires for Data, Address and Read/Write lines, and arranging the address decoding.

A general interface with full MPU control of all lines is quite possible but is a little inappropriate for the majority of users, hence Fig. 2.4 suggests a simple interface where control is mostly directed by switches.

A two-bit I/O port on the MPU system could easily be utilised to provide 1's and 0's for RES (Reset) and PROG (program initiate). A further line could accept the READY signal from the EPROM programmer. However, most systems will require an EPROM programmer reasonably rarely and special software would be required for a fully MPU-controlled machine.

Thus, the EPROM programmer reset line is connected to a Ground switch and pull-up resistor, and the Program-Initiate pin to a +5V switch and pull-down resistor for hand use. A power-up reset is also suggested with a capacitor across the Reset switch. A 1K Address-Decode line is assumed here, thus only A0-A9 are fed to the programmer and A10, again, is set by hand to select RAM or EPROM. If a 2K or larger Address Decode line is available this may be connected to ENABLE and the programmer's A10 connected to the MPU's A10 line: RAM and EPROM are then addressable together.

If two separate 1K decode lines are available (as with KIM and COMPUKIT). then Fig. 2.5 suggests how they may be connected to achieve the same result. Here, a zero on D1 or D2 forces ENABLE to zero via nand gates a and b. If D1 is low, then D2 must be high and gate c receives two "1's, forcing A10 low which selects the RAM. If D2 is low and D1 high, then c receives a logic 1 and a 0. This forces A10 high, selecting the EPROM. A later section deals with interfacing directly to the COMPUKIT.

SETTING UP AND TESTING

Assuming the EPROM programmer has been connected to the MPU system via Data Bus, Address Bus, ENABLE and R/W lines, the correct links should be inserted for the EPROM type chosen. Table 1 describes all the link options

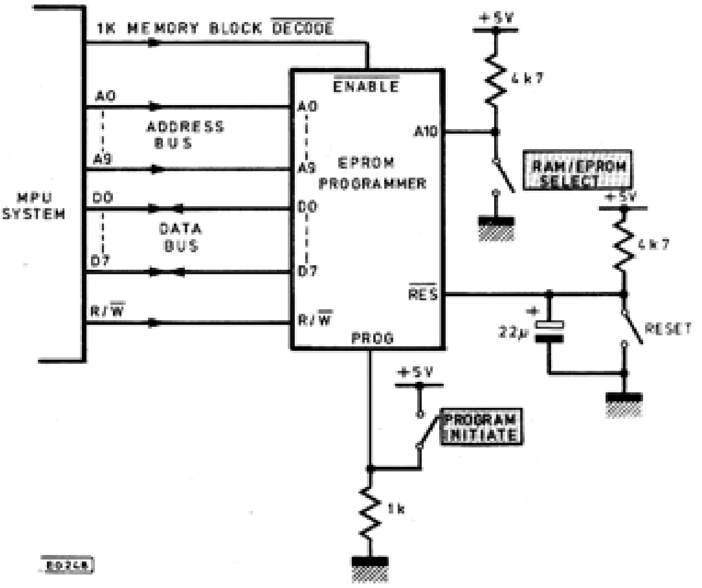


Fig. 2.4 Simple interface

on the board. Of course, any or all of the links could be replaced by appropriate switches if the links are to be changed frequently.

L5 and L6 must not both be connected simultaneously, even momentarily, as the power supply will be shorted. By using L5 and L6, each half of the 2716 is programmed separately, since the RAM holds just 1K of data and the 2716 has 2K. Connecting L1, L3, L7, excludes L2, L4, L8 and vice-versa. L9 is connected only for programming and a switch in this position (as for L5 and L6 if 2716's are to be used) would be most advantageous.

L10 should always be in place unless pin 1 of IC8 is required as a further ENABLE input. This can be used to simplify Address Decoding in some systems. In this case, L10 is left unconnected and pin 1 of IC8a connected to a top-side pad of the edge connector for external connections via the edge socket. L10 is assumed connected in the following.

Certain checks are worth making when first using the machine. Make sure, before applying mains, that the secondary windings of the mains transformer are connected correctly—centre tap to the centre pin.

AT SWITCH-ON

When first switching on, leave the EPROM out and L9 unlinked. Check the −5V and +12V supplies at the EPROM socket. Check +27V at L9.

The 12 volt regulator may become very hot during use. This can be replaced by a 1 Amp device at a very small cost if the overheating is excessive. Similarily, R1 dissipates some heat, and this can be replaced by two ½W 2K resistors in parallel or a 1W 1K resistor if the problem is excessive.

TABLE 1

LINKS	OPERATION
L1, L3, L7	2708
L2, L4, L8	2716
L5	Selects lower half of 2716
L6	Selects upper half of 2716
L9	Supplies 27 volts to Program Pulse switches
L10	Places a "1" on pin one of IC8. This is *normally connected*—but see text.

Neither of these modifications proved necessary on the prototype, even after many hours of use.

If the l.e.d. is off after switch-on, short $\overline{\text{RESET}}$ to zero momentarily. If the light remains off, check the +5V supply and then refer to the Troubleshooting section. Assuming everything is okay, check that the RAM is working, by Reading and Writing to it using the MPU system's monitor. The RAM must be selected by connecting A10 to zero if the MPU system does not perform this action automatically (see the section on interfacing). A small memory test program may be written to check that information can be written to, and read from, each location in the RAM.

If the RAM operation is satisfactory, check that S1 is in the non-program position and plug the EPROM into its socket. The EPROM should then be selected by ensuring that A10 is at "one". A test may be performed without sacrificing the contents of the EPROM in the following manner. The RAM should be filled with random data and either the EPROM absent or S1 in the non-program position. Set the programming cycle going by applying a positive pulse to the PROGRAM line. The Address lines to EPROM should be oscillating with A0 (pin 8) changing the most rapidly, A1 (pin 7) at half that speed and so on up to A9 (pin 22). At the same time, D0-D7 should all be oscillating and R4 should be applying 0·5 ms pulses at 27 volts to pin C of S1 with L9 linked. These are all necessary conditions for the correct working of the system.

PROGRAMMING

Ideally, the sequence of operation is as follows. 2708 or 2716 links should be selected as in Table 1, with L5 or L6 in place if a 2716 is being used. A zero is placed on A10 and the RAM contents changed to the selected data for programming into the EPROM. If necessary, the contents of RAM may be checked—perhaps by running a program in the RAM block. Parts of the EPROM which are to remain unchanged must be presented with FF during programming and the corresponding blocks in RAM should contain FF, as explained in Part 1 last month.

Check that S1 is in the program position, link L9 and apply a logic 1 to the program line on IC1. The l.e.d. should switch off for about two minutes. During this time, the EPROM is being programmed. When the programming cycle is over, switch S1 back, take A10 to a "one" and check that the contents of the EPROM have been altered correctly. This may be done in several ways. A very convenient method of such verification is to write a program which compares the EPROM contents byte for byte with the contents of the RAM, or some "mirror" of the RAM contents in the MPU's RAM. An Error message could signal any disparities which may arise. Alternatively, if a program has been stored, it may be run *in-situ* in the EPROM.

L9 should be removed for safety, as soon as programming has finished.

TROUBLESHOOTING

The test given in the above section is a good one for checking that counters, address switches and the 27 volt switch are all working, as well as the RAM itself.

If reading from RAM and EPROM or writing to RAM is not occuring, check connection between MPU and pins of IC10, 11, 12, 7 and 14 with a continuity tester and check that, while out of program mode, address information is passing through to Address pins of RAM and EPROM. A similar check may be made for data on IC7 and 14. An oscilliscope is helpful in troubleshooting, and will normally narrow down the fault very quickly to a poorly-soldered joint, missing component or a solder bridge.

IC13's connections, particularly, should be checked very carefully throughout the system with a continuity tester, as this component forms a major link in the control of the machine.

If the programming cycle is not even starting, check that IC1 can be turned on and off by PROG and $\overline{\text{RESET}}$. Check the connections to IC9, where all the clocking information is generated, and check for appropriate length pulses on pins 13 and 5. Do not be alarmed if the 20µs turns out to be 30µs or the 0·5 ms is ±10 or 15%.

If the cycling is occuring but never stops, check that IC8c is connected correctly and read through last month's explanation of the working of the Program Timing and Control sections.

The usual theme for troubleshooting is to start at the clock, make that work and follow it right through the system. Again, a 'scope and a continuity tester are very important.

COMPUKIT INTERFACE

The EPROM programmer has been designed, in part, to be plug-compatable with the Compukit, and this section describes the actual connection necessary.

As in Fig. 2.4, connections for the Data Bus, Address Bus (A0-A9), R/W Line and $\overline{\text{ENABLE}}$ line must be provided for the interface. The first three groups are quite straightforward. The $\overline{\text{ENABLE}}$ line, however, varies from system to system.

The EPROM programmer, illustrated in Fig. 2.4 acts like a 1K block of RAM or EPROM, depending upon the state of A10. That block of memory must be inserted into the memory map of the MPU system and only enabled for the appropriate 1K of Memory Addresses. The KIM has a set of outputs (called K outputs) used specifically for this purpose of "slotting in" external blocks of memory.

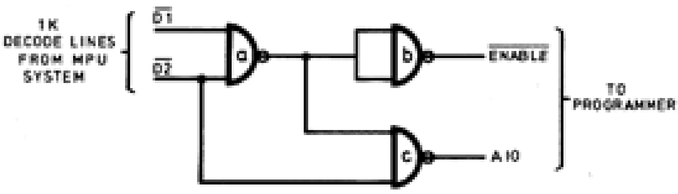


Fig. 2.5 Enable decode for Compukit and Kim

If a machine does not provide those external address-decoding lines, it is usually possible to replace an internal 1K block of Memory with the external block by using the internal block's dedicated Address Decoding line. This techinique is used on the Compukit. IC38 and IC52 on the Compukit are removed from their sockets and the EPROM programmer plugged into their place. This places the memory of the programmer in the address block: 1C00 → 1FFF. Of course, any of the 1K blocks from 0400 onwards may be used, but this will restrict the BASIC workspace. 0000 → 03FF may not be used as this provides scratchpad for the COMPUKIT's monitor program. Fig. 2.5 shows the Connection diagram for the above interface.

The Programmer's +5V power may be supplied by a Compukit with improved regulator heat-sinking. Two d.i.l. plugs and ribbon cable are most convenient for this link up and the cable may be soldered to an edge-connector socket which is plugged into the programmer's edge-connector surface. This forms a very neat job and allows instant unplugging of the devices when not in use.

The Address Decoding RS7 is supplied by IC38's socket in Fig. 2.6, as is the R/W line and Address Bus. Only half of the Data Bus is available at IC38's socket, hence the connection to IC52's socket. Use of the machine is quite simple. A10 is set to a low level to select the RAM and the program switch is off. $\overline{\text{RESET}}$ is brought low if the l.e.d. is off and the machine is ready for use. (Check that L10 is connected).

If some of the EPROM is to remain unprogrammed, a short BASIC program is written to fill addresses 1C00–1FFF with FF before the RAM is filled with a block of information to be stored in the EPROM. A program is given below for this process. The only point to remember is that all addresses and data are in *decimal* in BASIC.

```
10 FOR I = 7168 TO 8191
20 POKE I, 255
30 NEXT
```

This can be checked by replacing line 20 by:

```
20 IF PEEK(I) < >255 THEN PRINT I
```

Any locations not containing FF (255 in decimal) will be found and their addresses printed out on the screen. The erased EPROM may be checked by taking A10 "high" (which selects EPROM) and running this program.

The RAM may now be filled with binary information either by resetting the Compukit and using "M" to enter the machine code monitor, or by using the Compukit's powerful extended machine code monitor.

To program the EPROM, L9 is connected, the program switch (S1) thrown, and PROG brought high momentarily. When the l.e.d. comes on again, switch S1 back, remove S9, bring A10 to a high level and read the EPROM through to check the contents.

If Fig. 2.5 has been used to replace the upper 2K of memory with the programmer, the EPROM and RAM may be read and compared directly by the Compukit to verify EPROM contents. The connections for this would require IC37 and IC51 to be removed and RS6 (pin 8 of either socket) would be connected to $\overline{\text{D2}}$ (Fig. 3). $\overline{\text{D1}}$ would be connected to RS7 (pin 8, of IC38). The RAM would then reside at addresses 1C00→1FFF and the EPROM at addresses 1800→1BFF (DECIMAL 6144→7167).

The following program would check EPROM against Ram:

```
10 FOR I = 0 TO 1023
20 IF PEEK (6144 +I) < > PEEK (7168 +I)
   THEN PRINT (6144 +I)
30 NEXT
```

This will print out those addresses in EPROM which disagree with the RAM contents.

TABLE 2

Special EPROM programmer signals	Function
A10	HIGH level selects EPROM LOW selects RAM
$\overline{\text{RESET}}$	LOW pulse stops programming cycle
PROG	HIGH pulse initiates programming cycle
$\overline{\text{ENABLE}}$	LOW level selects RAM or EPROM depending upon A10
$\overline{\text{READY}}$ (output)	LOW level output from here implies programming cycle has stopped—l.e.d. in ON condition
READY (output)	Inverse of $\overline{\text{READY}}$ line

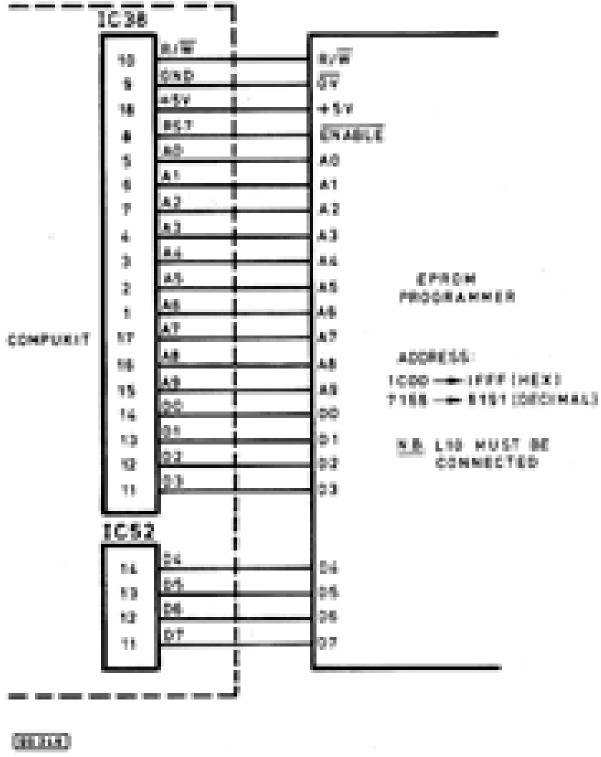


Fig. 2.6 Interface for Compukit

If this occurs, the RAM should be checked through for correct data and the EPROM reprogrammed. If a few unrelated locations remain unprogrammed, then either the EPROM was not fully erased or the chip is faulty. If many locations are incorrect, the programmer's operation must be suspected.

Adding the programmer to the Compukit, as in Fig. 2.6, does not in any way restrict the Compukit's memory, as 1K of RAM is still available in the programmer for normal use. This 1K of RAM, however, has the added advantage that its contents may be stored permanently.

ERASING

Erasure of the EPROM consists of shining a strong Ultra Violet light through the transparent quartz window in the package's upper surface. Short wave UV is required (around 2500 Angstrom units) and exposure time varies from 8 minutes to one hour or more, depending mainly upon UV intensity. A medicinal "sunray" light has been found to perform erasure quite effectively, and some experimentation for erasure time is essential.

Erasers may also be purchased and several are advertised in this magazine. In addition, PE has published an EPROM eraser (June 1978) as part of the CHAMP articles and this would be an excellent inexpensive alternative.

CONCLUSION

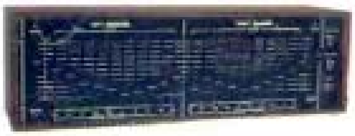
For many years, the problem of non-volatile memory storage has been solved by media such as paper-tape, disc and cassettes. These media are still important for mass storage. However, the coming of Micros has led to the need for permanent *alterable* storage of small capacity. This need is satisfied by the 2708 family and the EPROM programmer described here in an inexpensive, but hightly flexible manner.

In addition, anyone attempting the development of an MPU system from scratch, may use another system plus the programmer to produce the all-important and previously elusive System Monitor required by any machine when first powered up. ★

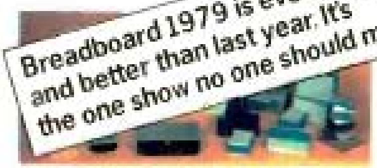
MAPLIN

A 63-key ASCII keyboard with 625-line TV interface, 4-page memory and microprocessor interface. Details in our catalogue.

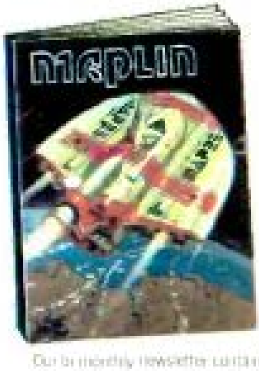
Our catalogue even includes some popular car accessories at marvellous prices.



A 10-channel stereo graphic equaliser with a quality specification at an unbeatable price when you build it yourself. Full specification in our catalogue.

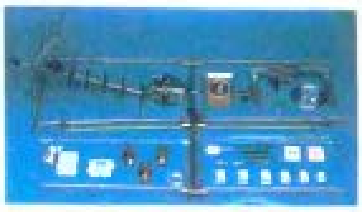


These are just some of the metal cases we stock. There are dozens of plastic ones to choose from as well.
See pages 52 to 57 of our catalogue.



A massive new catalogue from Maplin that's even bigger and better than before. If you ever buy electronic components, this is the one catalogue you must not be without. Over 280 pages – some in full colour – it's a comprehensive guide to electronic components with hundreds of photographs and illustrations and page after page of invaluable data.

Our bi-monthly newsletter contains guaranteed prices, special offers and all the latest news from Maplin.



Mobile amateur radio, TV and FM aerials plus lots of accessories are described in our catalogue.

A digitally controlled stereo synthesiser the 5600S with more facilities than almost anything up to £3,000. Build it yourself for less than £750. Full specification in our catalogue.

A superb range of microphones and accessories at really low prices.
Take a look in our catalogue – send the coupon now!

An attractive mains alarm clock with radio switching function and battery back up! Complete kit with case only £18.38 (incl. VAT & p & p) MA1023 module only £8.42 (incl. VAT).

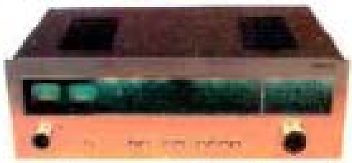
Post this coupon now for your copy of our 1979-80 catalogue price 70p.

Please send me a copy of your 280 page catalogue. I enclose 70p (plus 37p p&p). If I am not completely satisfied I may return the catalogue to you and have my money refunded. If you live outside the U.K. send £1.35 or ten International Reply Coupons.
I enclose £1.07.

NAME ____________________

ADDRESS ____________________

PE 180



A hi-fi stereo tuner with medium and long wave, FM stereo and UHF TV sound! Full construction details in our catalogue.

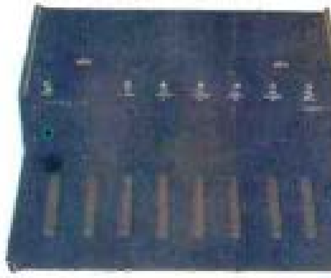
Breadboard 1979 is even bigger and better than last year. It's the one show no one should miss!

BREAD BOARD 1979

4th – 8th December,
Royal Horticultural Halls,
Elverton St., London.
(Nr. St. James' Park Underground)

Maplin will be there in force with all our projects on permanent demonstration. There will be plenty to see and many of our interesting lines will be on sale.

A superb technical bookshop in your home! All you need is our catalogue. Post the coupon now!



Add-on bass pedal unit for organs. Has excellent bass guitar stop for guitarists accompaniment. Specification in our catalogue.

MAPLIN
ELECTRONIC SUPPLIES LTD

All mail to:–
P.O. Box 3, Rayleigh, Essex SS6 8LR.
Telephone: Southend (0702) 554155.
Shop: 284 London Road, Westcliff-on-Sea, Essex.
(Closed on Monday).
Telephone: Southend (0702) 554000.