UNIVERSAL EPROM PROGRAMMER MARK II

Make programming even exotic EPROMs easy with our upgraded Universal EPROM programmer. Mike Bedford (hardware) and Gordon Bennett (software) produce a software-driven version of this invaluable device in both upgrade and self-contained form.

As far as the world of microelectronics is concerned, a lot of water can pass under the bridge in 18 months. In August 1983, a design for a universal EPROM programmer was published in ETI and, as its name suggests, this piece of equipment allowed virtually all the common single supply EPROMs to be programmed. At this time the largest device available was the 27128, only preliminary data being available for the 27256. In the intervening period the 27256, 27512 and 27513 have become available and since they use a different programming voltage to the previous devices may not be programmed by the original programmer. The 2764A and 27128A have also made an appearance, these being lower programming voltage versions of the 2764 and 27128 respectively. This being the case, it seemed appropriate to introduce a MkII version of the EPROM programmer to support these new devices and at the same time make some other improvements. We have produced an upgrade board to allow existing users of the MkI board to enhance it and also a single MkII board for those without the earlier board.

The MkII Universal EPROM Programmer is capable of programming a comprehensive range of single supply EPROMs varying in size from the 2758 to the 27512 and 27513 and including the 27-series, 25-series and the Motorola 68-series as well as a number of EEPROMs. In addition it allows the 2764 and larger devices to be programmed by the intelligent programming method hence reducing programming times drastically. All supply voltages have been made switchable under program controls so there is no need for a switch on the programming console. Two LEDs have been provided to indicate the current status of the programmer — in particular whether or not it is safe to remove the device. A modification to speed up EPROM reading has been made and, as a final enhancement, it is easier to set up since the adjustment of the programming voltages has been made much finer and the potentiometers are now more accessible when the board is rack-mounted.

In both versions, the programmer is fully programmable and everything is controlled by software. It is designed around the Tanbus specification which means that it should be an easy task to interface it to any 6502 or 6809 based system and users of a Tangerine computer will be able to plug the programmer directly into the system rack.

New Devices

Before describing the new programmer it is helpful to outline the advances in the realm of EPROMs which have made this upgrade necessary. Table 1 shows the pin-outs of all the devices which are supported. A similar illustration was included with the original article showing how standard pin-outs made designing a *universal* programmer relatively easy. The new devices conform to the same standard and also have JEDEC pin-outs.

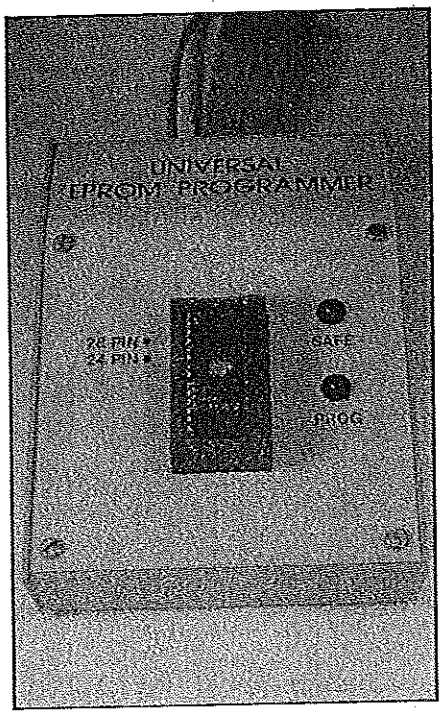
This family of devices must now include the largest single page EPROMs which will be used with 8-bit processors, as the capacity of a 27512 is 64K bytes — in other words, it occupies the entire memory map of an 8-bit system. The 27513 has the same capacity as the 27512 but the memory is organised differently having 4 pages of 16K bytes each and therefore representing the first of a new class of devices — paged EPROMs. To specify the page to be accessed a write operation is performed. The least significant 2 bits of the following data word

then specify the page number.

The major fact about the 27256 and 27512/2513 which makes them incompatible with the original EPROM programmer is that the programming voltage is 12.5V. This follows the trend of decreasing programming voltages as the capacity increases and the silicon die size decreases, the devices up to and including the 2732 using 25V and the 2764 and 27128 using 21V. In addition, versions of the 2764 and 27128 which also use the new 12.5V Vpp have been released. These are known as the M2764A and 27128A.

Intelligent programming is possible on all devices from the 2764 upwards. In this case, 1ms programming pulses are applied to the EPROM until it verifies, at which point a further pulse is applied. This contrasts with the standard programming method in which a 50ms programming pulse is always used. As the larger EPROMs are introduced, intelligent programming becomes increasingly desirable. It can reduce programming times from almost one hour to about eight minutes for the 27512.

Intelligent programming requires the supply voltage to be raised from the normal 5V to 6V during the programming cycle, a facility not available on the Mkl board. A different programming time reduction method has been introduced on the latest version of the Texas 25 series devices and on some manufacturers' recent 2732s and 2764s. These devices use a fixed length programming pulse of 10mS rather than the standard 50mS pulse.



The console of the programmer.

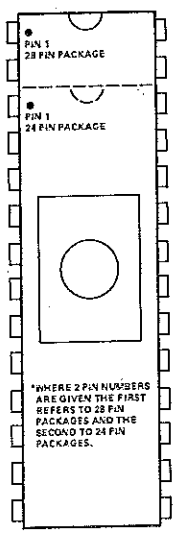
Another facility introduced on some of the newer devices is referred to as 'intelligent identifier' or 'auto select mode'. After applying +12V to A9, where this facility is available, one of two bytes may be read out depending on the logic level of A0. These two bytes contain codes identifying both the device type and the manufacturer. It was decided not to implement this mode for two reasons. Firstly, the facility was designed for industrial production programming where the process is often carried out by those with a minimal knowledge of electronics. By contrast, the home user will probably be clear about what device type is being used. Secondly, not all devices include the facility, and it is reasonable to assume that applying +12V to A9 of EPROMs without intelligent identifier will be detrimental. In an environment in which all devices from the 2758 upwards are to be programmed, the provision of the feature will increase the likelihood of destroying EPROMs.

The price of EPROMs has been influenced by the fact that quartz windows could only be fitted in ceramic packages. Recent advances now allow a seal to be made between quartz and plastic and, as a result, some manufacturers are releasing EPROMs in plastic packages at a significant cost reduction. Over the past few years, the price of EPROMs has already reduced to the point where they are comparable to the price of ROMs. Since a large proportion of the remaining cost is due to the quartz window, manufacturers have also started producing EPROMs without the quartz window at an even lower price.

The lack of quartz window means that these devices cannot be exposed to ultra voilet and erased. They are referred to as production EPROMs or OTP EPROMs (One Time Programmable). Since these EPROMs are electrically identical to standard EPROMs, they are programmed in exactly the same way.

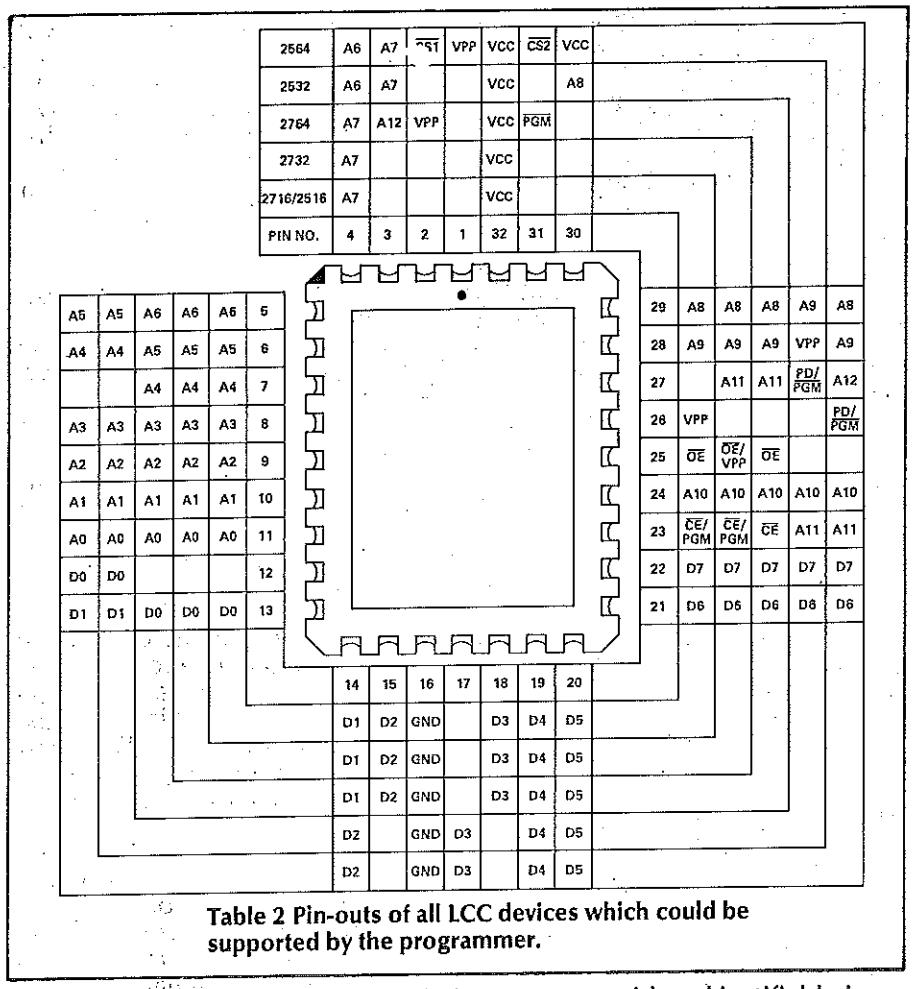
Some EPROMs are now available not only in the standard 24 or 28 pin DIL (dual-in-line) packages but also in the newer, smaller, LCC (leadless-chip-carrier) packages which have pins spaced at

27513	27512	27256	27128 27128A	2564	2764 2764A	68764	2532	2732 2732A	68732	2716 2516	2758	*PIN NO.
N/C	A15	VPP	VPP	VPP	VPP							1
A12	A12	A12	A12	$\overline{CS1}$	A12							2
A7	A7	A7	A7	A7	A7	A7	A7	A7	A7	A7	A7	3/1
A6	A6	A6	A6	A6	A6	A6	A6	A6	A6	A6	A6	4/2
A5	A5	A5	A5	A5	A5	A5	A5	A5	A5	A5	A5	5/3
A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	A4	6/4
A3	A3	A3	A3	A3	A3	A3	A3	A3	A3	A3	A3	7/5
A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	8/6
A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	9/7
A0	A0	A0	A0	A0	A0	A0	A0	A0	A0	A0	A0	10/8
D0	D0	D0	D0	D0	D0	D0	D0	D0	D0	D0	D0	11/9
D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	12/10
D2	D2	D2	D2	D2	D2	D2	D2	D2	D2	D2	O2	13/11
GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	14/12



*PIN NO.	2758	2716 2516	68732	2732 2732A	2532	68764	2764 2764A	2564	27128 27128A	27256	27512	27513
28							VCC	VCC	VCC	VCC	VCC	VCC
27							$\overline{PGM}$	$\overline{CS2}$	$\overline{PGM}$	A14	A14	$\overline{WE}$
26/24	VCC	VCC	VCC	VCC	VCC	VCC	N/C	N/C	A13	A13	A13	A13
25/23	A8	A8	A8	A8	A8	A8	A8	A8	A8	A8	A8	A8
24/22	A9	A9	A9	A9	A9	A9	A9	A9	A9	A9	A9	A9
23/21	VPP	VPP	AR	A11	VPP	A12	A11	A12	A11	A11	A11	A11
22/20	$\overline{OE}$	$\overline{OE}$	$\overline{E}$/ VPP	$\overline{OE}$/ VPP	PD/ $\overline{PGM}$	$\overline{E}$/ VPP	$\overline{OE}$	PD/ $\overline{PGM}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$/ VPP	$\overline{OE}$/ VPP
21/19	AR	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10	A10
20/18	$\overline{CE}$/ $\overline{PGM}$	$\overline{CE}$/ $\overline{PGM}$	A11	$\overline{CE}$/ $\overline{PGM}$	A11	A11	$\overline{CE}$	A11	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$/ $\overline{PGM}$	$\overline{CE}$
19/17	D7	D7	D7	D7	D7	D7	D7	D7	D7	D7	D7	D7
18/16	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6
17/15	D5	D5	D5	D5	D5	D5	D5	D5	D5	D5	D5	D5
16/14	D4	D4	D4	D4	D4	D4	D4	D4	D4	D4	D4	D4
15/13	D3	D3	D3	D3	D3	D3	D3	D3	D3	D3	D3	D3

Table 1 Pin-outs of all devices supported by the programmer.



2564	A6	A7	$\overline{CS1}$	VPP	VCC	$\overline{CS2}$	VCC
2532	A6	A7			VCC		A8
2764	A7	A12	VPP		VCC	$\overline{PGM}$	
2732	A7				VCC		
2716/2516	A7				VCC		
PIN NO.	4	3	2	1	32	31	30

A5	A5	A6	A6	A6	5
A4	A4	A5	A5	A5	6
		A4	A4	A4	7
A3	A3	A3	A3	A3	8
A2	A2	A2	A2	A2	9
A1	A1	A1	A1	A1	10
A0	A0	A0	A0	A0	11
D0	D0				12
D1	D1	D0	D0	D0	13

29	A8	A8	A8	A9	A8
28	A9	A9	A9	VPP	A9
27		A11	A11	$\overline{PD}$/PGM	A12
26	VPP				$\overline{PD}$/PGM
25	$\overline{OE}$	$\overline{OE}$/VPP	$\overline{OE}$		
24	A10	A10	A10	A10	A10
23	$\overline{CE}$/PGM	$\overline{CE}$/PGM	$\overline{CE}$	A11	A11
22	D7	D7	D7	D7	D7
21	D6	D6	D6	D6	D6

14	15	16	17	18	19	20
D1	D2	GND		D3	D4	D5
D1	D2	GND		D3	D4	D5
D1	D2	GND		D3	D4	D5
D2		GND	D3		D4	D5
D2		GND	D3		D4	D5

Table 2 Pin-outs of all LCC devices which could be supported by the programmer.

0.05" on 4 sides of a rectangle and so allow a much greater PCB packing density. Internally these devices are identical to conventionally packaged EPROMs and as a result the programming requirements are the same. To handle them, the EPROM programmer only needs to be provided with a different socket on the console. Table 2 shows the pin-out of those devices currently available in this package. It should be noted that the 25 series devices differ from each other and the 27 series devices in this configuration more than in the standard DIL package. Accordingly, it would be advisable to consider programming only the 27 series EPROMs in LCC configuration or to provide a number of different sockets.

Similar Devices

The term EPROM is usually taken to mean UV erasable PROM, but there is a closely related family of devices — electrically erasable programmable read-only memories, known as EEPROMs or E²PROMs. At the time of designing the original programmer, the extra complexity involved in supporting EEPROMs was not considered justifiable in view of their high cost. The price of EEPROMs has not dropped drastically and they are, therefore, still quite rare among home computer users. But numerous enhancements to these devices have been made which simplify the programming and accordingly they may be supported by the MkII programmer.

The fact which complicated the programming of the original Intel 2816 (2K x 8 EEPROM) was the fact that it used a 21V programming voltage which had to be shaped by an RC circuit to give an exponential rise. the next development still used 21V for programming but the waveform shaping requirement was relaxed, the only restriction then being on the fall time of the Vpp pulse. The latest EEPROMs don't even require a high programming voltage, internal circuitry generating this from the +5V supply. In addition, there are now some devices which support these very latest programming techniques but are compatible with earlier devices, accepting either 21V or TTL programming levels.

EEPROMs have also developed in the method of programming. On the first devices, a byte could only be programmed if it were first erased, either by writing an FF(HEX) to that byte or by using the complete chip erase facility. On the more recent devices, bytes may be directly re-programmed without the need for erasing first. Programming times and the number of programming cycles have also seen improvements. The first 2816 required 10mS programming pulses whereas some of the newer versions will programme in 2mS per byte. The technology used in EEPROMs, HMOS-E FLOTEX cell design, has an inherent limitation on the number of programming cycles. The original EEPROMs had a lifetime of 10,000 cycles but 1 million cycles is now not uncommonly quoted.

Unfortunately, there isn't the same degree of standardisation among EEPROMs as with UV EPROMs. Although a 2816 is always a 2K x 8 EEPROM, different manufacturers' devices with this number may represent a number of different points within the progression outlined above. In addition 2816A, 2817, 2817A and 5213 are variations on the same theme by various manufacturers. Because of these complications, we won't give a list of EEPROM type numbers which are supported by the MkII Universal EPROM programmer. It will, in fact, handle all those 2K x 8 devices which feature TTL level programming. Some 8K x 8 EEPROMs are also becomming available — for example, the 2864 and 52B33. Where these are programmed by TTL levels, they may also be supported by the MkII programmer.

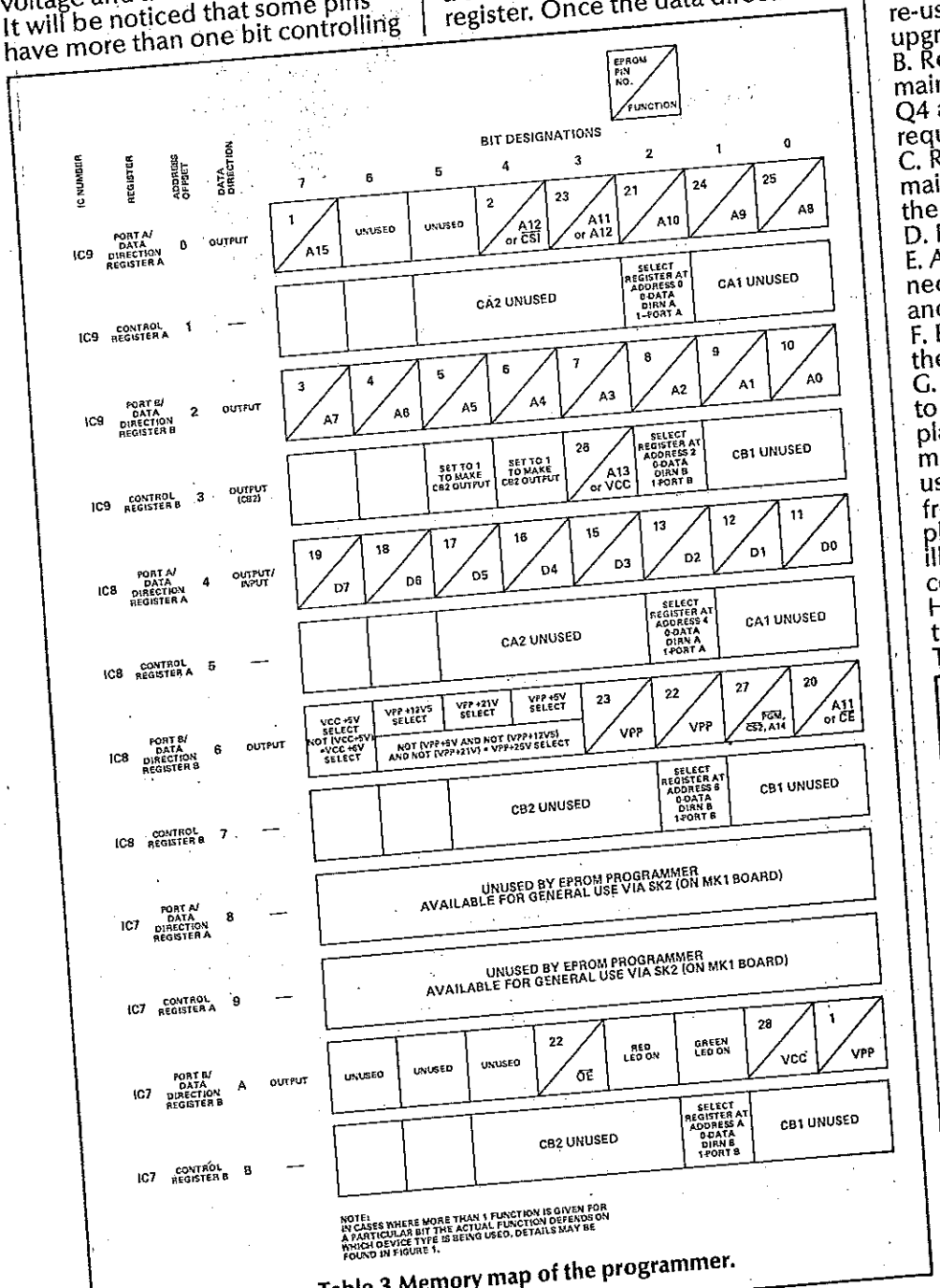
Mark II Board — Hardware

This section refers to either the MkII EPROM programmer or the MkI with the addition of the upgrade board, the hardware of these two configurations being identical with one exception. The MkI board has 4 x 6821 PIAs, of which 2 are used by the programmer for control functions leaving 2 free for general use. The MkII board utilises 3 PIAs for controlling the programmer, the 4th having been omitted in order to fit the extra circuitry onto the PCB. The upgrade board makes use of the 3rd PIA on the MkI board but does not, however, affect the 4th one which means that this configuration gives a spare PIA, the true MkII board not having this facility. (See 'How It Works' for part

numbering.)

Table 3 is a memory map of the MkII Universal EPROM Programmer in which the function of each bit is outlined. Some bits control certain functions such as Vcc and Vpp voltage levels, the majority, however, control the signal levels on various pins of SK3, the EPROM socket. For all bits connected to SK3, except those marked Vcc or Vpp, writing a 1 will set the pin to a logic high, whereas writing a 0 will set it to a logic low. For the Vcc and Vpp bits, a 1 sets the pin to the currently selected Vcc or Vpp voltage and a 0 sets the pin to 0V. It will be noticed that some pins have more than one bit controlling them. This happens where a particular pin can take either a logic level or a Vpp voltage. In such cases, although this wouldn't normally be required, it would not be harmful to set both bits high at the same time since the two corresponding outputs are isolated by use of diodes.

Finally, in the 6821 PIA, the data direction registers are double addressed with the corresponding I/O port register. Bit 2 in the appropriate control register determines which of these two registers actually will be addressed, a 1 selecting the I/O port register and a 0 selecting the data direction register. Once the data direction register is selected, setting a 1 to a bit in this register selects the corresponding bit in the I/O port to be an output whereas a 0 selects the I/O port bit to be an input.



NOTE: IN CASES WHERE MORE THAN 1 FUNCTION IS GIVEN FOR A PARTICULAR BIT THE ACTUAL FUNCTION DEPENDS ON WHICH DEVICE TYPE IS BEING USED. DETAILS MAY BE FOUND IN FIGURE 1.

Table 3 Memory map of the programmer.

The Upgrade Board — Construction

Construction of the upgrade board for the MkI programmer is straightforward and no special comments need to be made. Interfacing to the main board and setting up do require explanation. The procedure is as follows:

A. Remove the regulator IC10 from the main board. This may be re-used as IC12 or IC13 on the upgrade board.
B. Remove R4, R5 and C1 from the main board. RV1, RV2, RV3, Q3, Q4 and R6 may also be removed if required.
C. Remove SW1 if fitted to the main PCB or if not fitted remove the two wire links in its place.
D. Remove D1 on the main board.
E. Add D3 to the main board, connecting the cathode to SK3 pin 1 and the anode to IC9 pin 9.
F. Replace C5 (10n) by 100n on the main board.
G. Physically fix the upgrade board to the main board by use of three plastic bolts. If the fixing holes marked on the upgrade board are used they will allign with 'track-free' areas of the main board. The photographs with this article illustrate the means of interconnection.
H. make the connections between the two boards as shown in Table 4.

An early prototype of the upgrade

J. If using a programming console remove the switch, connecting the two wires which this interrruped directly to the appropriate ZIF socket pins and add 1 green and 1 red LED which are wired to connector A on the upgrade board via a 4-way cable as follows:

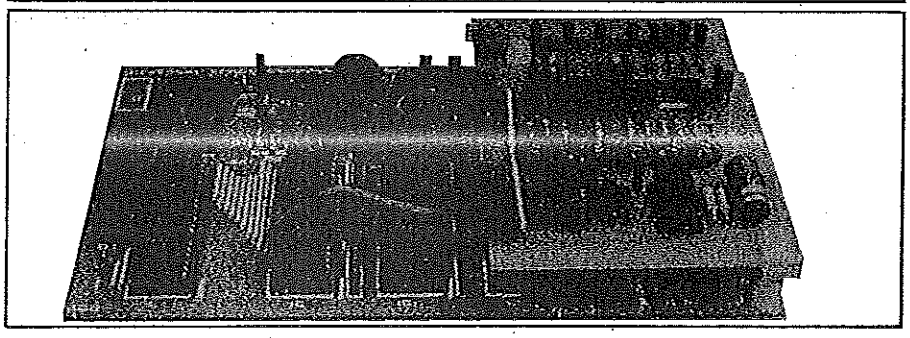
A/1 Green LED anode
A/2 Green LED cathode
A/3 Red LED anode
A/4 Red LED cathode

K. Installation is now complete and Vcc and Vpp voltages need to be set up as follows after first temporarily removing the wires to connector C:

1. Apply +5V to C7 only and adjust RV4 to give +5V on B4.
2. Apply +5V to C6 only and adjust RV5 to give +12.5V on B4.
3. Apply +5V to C5 only and adjust RV6 to give +21V on B4.
4. Remove +5V from C5 and adjust RV7 to give +25V on B4.
3. Apply +5V to C4 only and adjust RV8 to give +5V on B2.
4. Remove +5V from C4 and adjust RV9 to give +26V on B2.

UPGRADE BOARD CONNECTOR	MAIN BOARD DESTINATION
C/ 1	IC7 PIN 14
C/ 2	IC7 PIN 13
C/ 3	IC7 PIN 12
C/ 4	IC8 PIN 17
C/ 5	IC8 PIN 15
C/ 6	IC8 PIN 16
C/ 7	IC8 PIN 14
C/ 8	IC7 PIN 11
C/ 9	IC7 PIN 10
C/10	IC8 PIN 12
B/ 1	IC11 PIN 1
B/ 2	NO CONNECTION
B/ 3	SK3 PIN 28
B/ 4	O/P (CENTRE) OF IC10
B/ 5	SK3 PIN 1
B/ 6	OV (IC11 PIN 11)
B/ 7	SK3 PIN 22
B/ 8	+5V (IC11 PIN 13)

Table 4 Connections between the upgrade board and the original programmer.



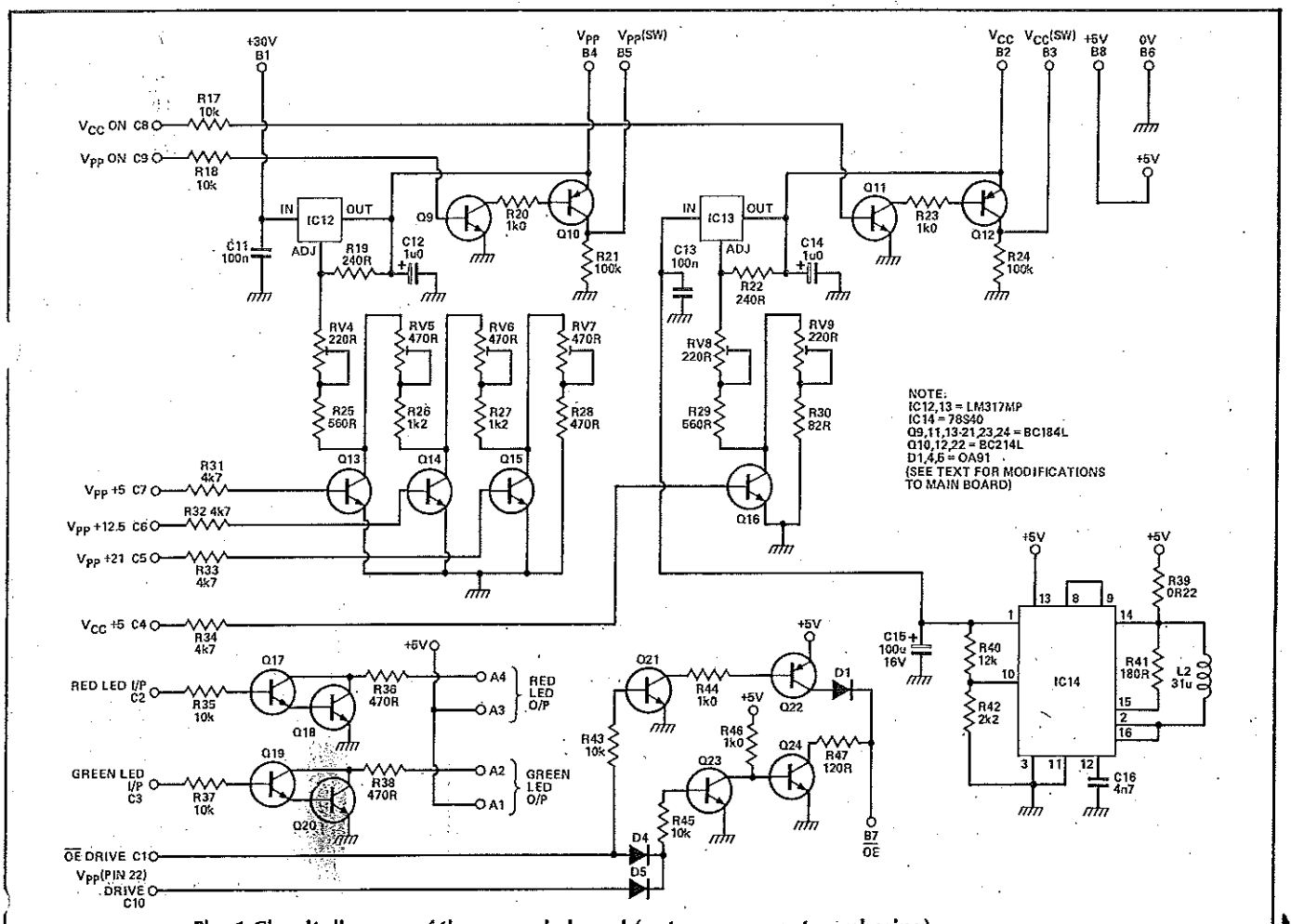


Fig. 1 Circuit diagram of the upgrade board (note component numbering).

PROJECT: EPROM Programmer MkII

HOW IT WORKS

Readers should note that the component numbers on Fig. 1 — the upgrade board circuit diagram — do not start at 1. Instead they follow on from the component numbers on the main MkI board. The following description assumes a knowledge of the workings of the MkI board to which the upgrade board is connected and a description of which may be found in ETI August 1983.

The upgrade board supplies Vpp (selectable to +5V, +12.5V, +21V or +25V), Vcc (selectable to +5V or +6V), a replacement driver for $\overline{OE}$ (the active low output enable line) and drivers for two LEDs. The old part of the Vpp circuitry which generates an unregulated +30V by use of a 78S40 has been retained. However, the regulator consisting of a LM317MP and a resistor chain, in which portions of the chain could be switched out by transistors, has been replaced. The new regulator is similar to the one on the MkI board but differs in two respects. Firstly each variable resistor in the chain has a fixed resistor in series with it, hence giving a more accurate means of setting up the voltages. Secondly an extra resistor portion and transistor have been added to allow the +12.5V programming voltage to be selected. This regulation circuit comprises IC12, Q13, Q14, Q15 and the associated passive components. An unswitched Vpp is passed to various Vpp switches on the main board. Transistors Q9 and Q10 provide a switched Vpp which replaces the supply to EPROM pin 1, previously switched manually. IC14 and its associated components form a second step-up circuit providing a +8V supply which is regulated to either +5V or +6V for Vcc. This regulator circuit is built around IC13 and is a similar configuration to the Vpp regulator. Transistors Q11 and Q12 provide a switched Vcc supply which replaces the original, manually switched supply to EPROM pin 28.

It should be noted that the Vcc supply to pin 24 on the EPROM need not come from this circuitry as no 24pin devices feature intelligent programming, so +5V will always be used.

On the original board a 10nF capacitor, C5, was connected between $\overline{OE}$/Vpp on pin 22 of the EPROM socket and 0V. This was a compromise between the 100nF suppression capacitor actually specified in the 2732 data sheet and a value which wouldn't slow down logic edges too much. On the new circuit the recommended 100nF capactitor is used but logic signals are not significantly slowed down as a result of the Q21/Q22 combination which provides a high current $\overline{OE}$ signal capable of charging the capacitor rapidly and Q24 which provides a logic low signal bypassing the suppression capacitor. Capacitor C5 should be changed on the original board.

Transistor Q23 provides a NOR function, turning Q24 on when neither of the signals driving EPROM pin 22 are present.

Transistors Q17, Q18, Q19 and Q20 simply form two darlington drivers with built-in current limiting resistors to drive two LEDs indicating programmer status. In addition to the extra circuitry on the upgrade board an extra diode, D3, is added to the main board. This is to provide the extra address line A15 to pin 1 of the EPROM socket, the diode being required to isolate it from the Vpp supply which can also be present on this pin. This diode — an OA91 — should be fitted between SK3 pin 1 (cathode) and IC9 pin 9 (anode).

PARTS LIST

RESISTORS (all ¼W, 5% unless stated)

R17, 18,35,37,43, 45	10k
R19,22	240R
R20,23,44,46	1k0
R21,24	100k
R25,29	560R
R26,27	1k2
R28,36,38	470R
R30	82R
R31,32,33,34	4k7
R39	OR22 W/W
R40	12k
R42	2k2
R47	120R 1/2W
RV4,8,9	220R vertical min preset
RV5,6,7	470R vertical min preset
CAPACITORS	
C5 (replace on main board)	100n ceramic
C11,13	100n ceramic
C12,14	1u tantalum
C15	100u 16V axial electrolytic
C16	4n7 polyester
SEMICONDUCTORS	
IC12,13	LM317 MP
IC14	78S40
Q9,11,13,14,15,16, Q17,18,19,20,21, 23,24	BC184L
Q10,12,22	BC214L
D1,D4,D5	OA91
D3 (fit on main board)	OA91
MISCELLANEOUS	
L2	31uH, 13 turns 22 SWG on RM6 pot core (AL=250)
Connectors A,B,C	0.1" pitch right angled molex connectors. 4,9 and 8 ways respectively.

PCB, three plastic bolts and nuts for attaching to main PCB.

BUY LINES

All components are standard. The biggest problem may be in finding a 28-pin ZIF socket. These are supplied by Watford Electronics and Technomatic. Electrovalue and Maplin will supply a OR22 wirewound resistor and Electrovalue will also supply the RM6 pot cores. The Molex connectors are standard inter-PCB connectors and the Euro connector for the MkII board likewise. The version of the LM317 you should look for is one in a TO 202 or TO 220 case - a 317 M or 317 T will do if you can't find a 317 MP. All semiconductors should be available from any supplier with a good stock - Technomatic, Rapid and TK advertise the LM317 T, Watford advertises all the other ICs.

Over the next two months, we'll be dealing with an entirely self-contained version of the MkII and with the software to drive the upgrade.

ETI

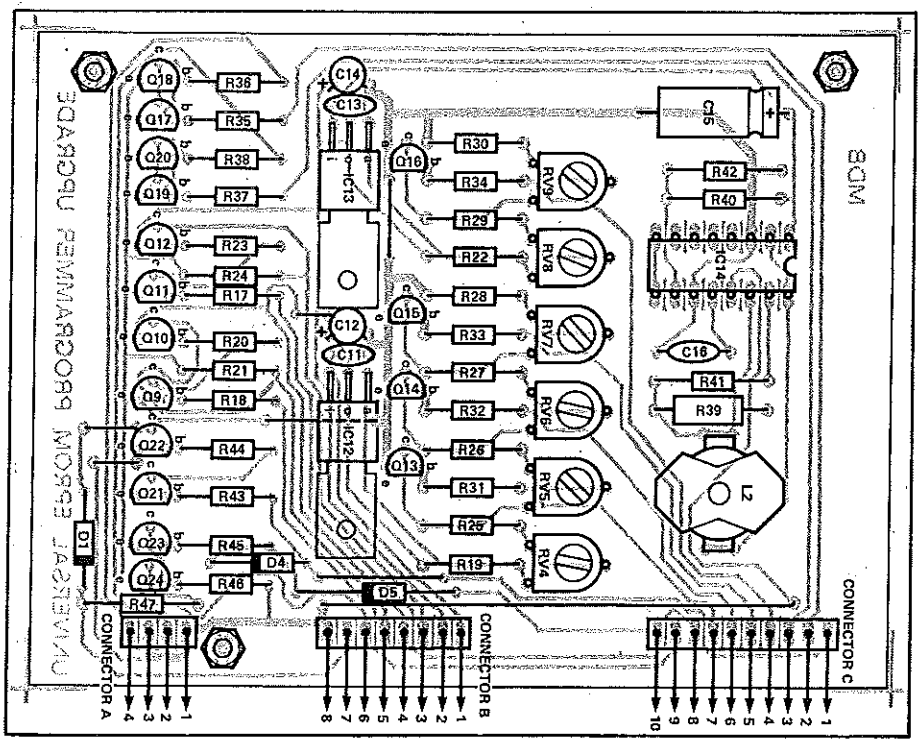


Fig. 2 Overlay diagram of the upgrade board.

UNIVERSAL EPROM PROGRAMMER MKII

Following on from last month's article which covered the theory and described an upgrade modification for existing programmers, Mike Bedford and Gordon Bennett describe an improved EPROM programmer for those building from scratch.

Unlike the MkI board, the MkII board has been made double sided to cope with the greater component density. In order to keep down the costs, plated through holes have not been used which means that the first task to be carried out in building this project is to insert pins into all the holes marked as such on the component overlay diagram, soldering them on both sides of the board. After having carried out this through pinning, the construction is quite straightforward. One point worth noting is that component leads are sometimes relied upon to make a connection from one side of the board to the other. This means that if a component lead passes through a hole with pads on both sides of the board, the lead should be soldered to them both.

The MKII board will be used in conjunction with a programming console housing a 28 pin ZIF (zero insertion force) socket and 2 LEDs (see photograph). The 2 LEDs on the console connect to the main board via a 3 or 4 core cable connected to SK4, the anodes being connected to A1 and A3, the cathode of the green LED to A2 and the cathode of the red one to A4. The ZIF socket is connected via a length of ribbon cable and a 28-pin DIL header to SK3 on the main board on a pin to pin basis. It should be noted that the DIL socket SK3 is the "wrong way round" with respect to all the DIL ICs on the board and accordingly care should be taken in plugging in the ribbon cable to the console. A 0.1uF capacitor should be connected between pin 28 and pin 14 on the ZIF socket.

Construction having been completed, it now remains to configure the board to reside at the required address and to set up the various Vcc and Vpp voltages. The addressing is determined by the links, LK1, which are wired into a

OOPS!

Since the appearance of last month's article, a problem has come to light regarding the programming of 27512 EPROMs.

The problem occurs when using the fast programming algorithm with the 27512 and results from the necessary sequence of operation adopted in the software. The $\overline{OE}$ line is held high until dropped to access the EPROM for reading and the $\overline{CE}$ line goes low as soon as the programming voltage is removed from the EPROM.

But on the 27512 the $\overline{OE}$ line is also the Vpp select line and so, although this line is set low by the software at the correct time, the combined line is still held high by the $\overline{OE}$ bit until it is time to read the EPROM. This is because the hardware combines these two lines in an OR gate. The effect is to hold the 27512 in programming mode for an extra 300 micro seconds at a time when, although the address and data busses should not be varying, the programmer itself is changing from program to verifying mode. It is quite possible that this would cause no ill effects, but it is undesirable and should be corrected.

A software solution would require a separate procedure for the 27512 in an already crowded EPROM, but a far simpler hardware modification is possible. It consists of the removal of two diodes and the substitution of a wire link for one of them. The diodes in question perform an OR function at the input of the active pulldown circuit which operates on pin 22 of the EPROM. They were put there to prevent high dissipation in the 120R resistor by removing the possibility of the software turning on both transistors simultaneously. No problems have been found using the existing software package without these diodes, and their absence has no effect upon the operation of the programmer with other EPROMs.

The modification is:

1) locate and remove the diode in the line from pin 14 of PIO 3 (IC7), the $\overline{OE}$ line;

2) locate and remove the diode in the line from pin 12 of PIO 2 (IC8), the Vpp select line;

3) replace this latter diode with a wire link.

This will prevent the $\overline{OE}$ line from influencing the pulldown of the Vpp/$\overline{OE}$ line.

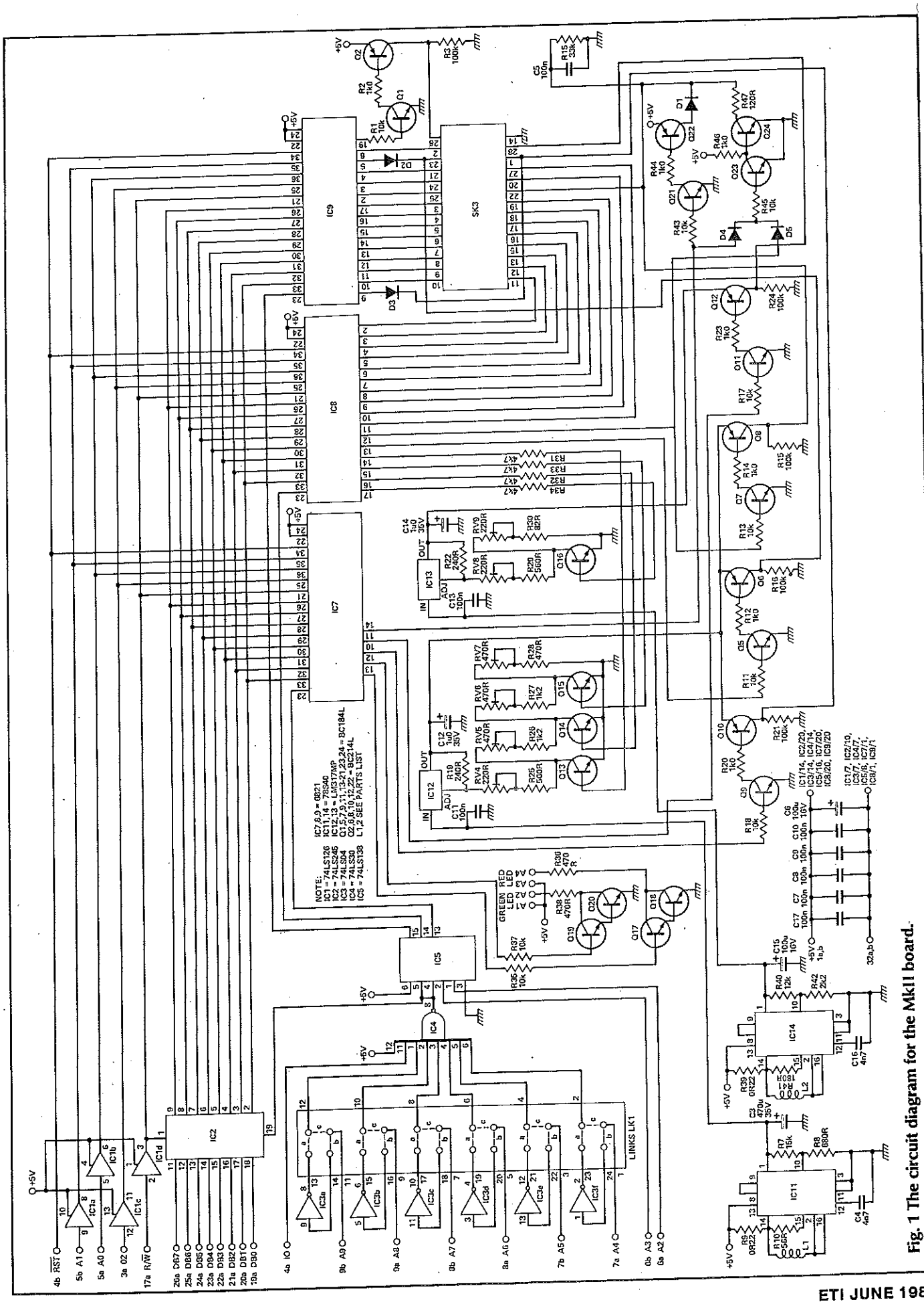


Fig. 1 The circuit diagram for the Mk11 board.

HOW IT WORKS

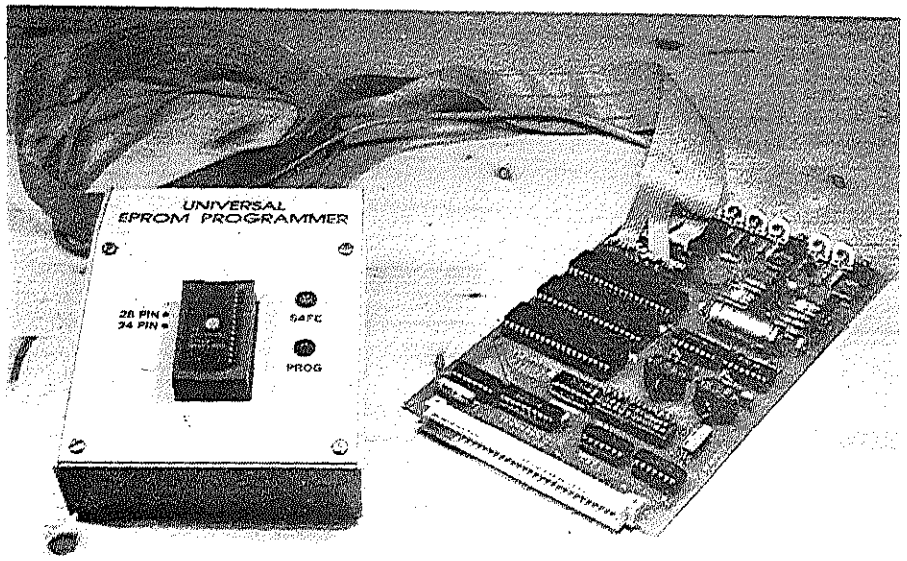
The components in Fig. 1, the circuit diagram of the Mk II board, have been numbered in such a way that they correspond to the component numbers on the Mk I and upgrade boards. Since a few components are removed from the Mk I board when the upgrade board is fitted there will be some gaps in the component numbers on the Mk II programmer. Once this is realised, this arrangement should cause less confusion than if components with the same function were to have different numbers in the two configurations.

The heart of the circuit is three 6821 PIAs which control all the programmer functions. These are interfaced in a standard way to the Tanbus signals on the edge connector. IC1 and IC2 buffer various signals to ensure that only 1 TTL load is applied to a bussed signal and the combination of IC3, IC4, IC5 and the links control the addressing and allow the board to be located within any 16 byte block in the I/O area. The Vpp supply is generated at +30V by the circuitry associated with IC11 which is a step-up circuit and is then regulated to the required level (+25V, +21V, +12.5V or +5V) by a programmable LM317MP regulator, IC12. Since the voltage output of an LM317MP is determined by the value of the resistor between the adjust pin and 0V, the Vpp level is controlled by switching the transistors Q13, Q14 and Q15 from PIA IC8 so cutting out portions of the resistor chain.

A similar approach is used to generate Vcc, IC14 generating +8V and IC13 regulating to +5V or +6V as controlled from IC8. Where a pin on the EPROM (which is connected to SK3) requires a TTL signal level it is connected directly to an output of one of the PIAs. Where a Vpp or Vcc level is required, however, a NPN/PNP transistor pair is used to carry out the switching under the control of a PIA output. In all such cases the transistor pair must be connected to a PIA 'B' port, these having totem-pole outputs which can supply sufficient current to switch a transistor.

The signal level on some pins may be either TTL or Vpp, depending on the EPROM type. In such cases, both signals are connected to the appropriate pin but the two are isolated from each other by use of a diode on the TTL signal line. When a TTL level is isolated by a diode, this is driven by a PIA 'A' port since these outputs have resistive pull-ups and will give a level that is high enough to be a true TTL high even after allowing for the voltage drop across a germanium diode.

The data sheets for the 2732 call for a 100uF capacitor (C5) connected between pin 22 and 0V while programming. This will suppress spikes on the Vpp supply which could be detrimental to the EPROM. Unfortunately the provision of such a suppression capacitor will have the result of slowing down logic edges when a TTL level is applied to pin 22. For this reason, the time constant is kept to a minimum by using Q21 and Q22 as a high current OE drive and Q24 to provide a logic low bypassing C5. Transistor Q23 turns on Q24 when neither the OE nor the Vpp signal driving EPROM pin 22 is present. To complete the circuit description, Q17, Q18, Q19 and Q20 form two darlington pairs which are used to drive a pair of LEDs on the programming console.



Voltage required	ZIF pin to monitor	Register address offset	Value to write to register (HEX)	Potentiometer to adjust
–	-	0B	00	—
–	-	0A	FF	—
–	-	0B	04	—
–	-	0A	03	—
–	-	07	00	—
–	-	06	FF	—
–	-	07	04	—
+5V	1	06	10	RV4
+12.5V	1	06	40	RV5
+21V	1	06	20	RV6
+25V	1	06	00	RV7
+5V	28	06	80	RV8
+6V	28	06	00	RV9

Table 1 The set-up arrangements for monitoring and adjusting program voltages.

PARTS LIST

Programming Console

1 x Instrument case with sloping top
1 x 28-pin DIL Zero Insertion Force socket
1 x Length of 28-way ribbon cable
1 x 28-pin DIL header
1 x 100 μF ceramic capacitor
1 x Red LED
1 x Green LED
1 x Length of 4-way cable

DIL header and plugged into the appropriate DIL socket. The board occupies a 16-byte block within the 1K Tanbus I/O space, the start address relative to the start of this I/O area being 16 times the binary number represented by the block of links. The examples of link selection in Fig. 2 should make it quite clear how to set up any required addresses. The MkII board has been designed with the voltage setting potentiometers placed along the edge of the board so that they may be easily adjusted once the board has been positioned in a card frame. The voltages may now be monitored on the programming console and adjusted, using the potentiometers, by writing values to the programmer registers using the system monitor (or a BASIC program). Table 1 shows the requisite programming voltages, associated pins, registers, data and potentiometers.

MKII UNIVERSAL EPROM PROGRAMMER : HARDWARE SPECIFICATION

Devices supported	: 2758, 2716, 2516, 2732, 2732A, 2532, 68732, 2764, 2764A, 2564, 68764, 27128, 27128A, 27256, 27512, 27513,2816, 2864
Device selection	: Software controlled
Programming methods	: Intelligent or fixed pulse
Vpp voltages	: +25V, +21V, +12.5V
Vcc voltages	: +6V, +5V
Indicators	: 2 LEDs on console
PCB format	: 8" x 4½" with indirect connector
Interface	: Tanbus (6502, 6800, 6809 adaptable)
Power requirements	: +5V @ 900mA
Memory space occupied	: 12 bytes selectable to any 16 byte boundary within the I/O area
System requirements	:RAM — 1K for 2758 to 32K for 27256, 27512* and 27513* plus small amount for support firmware. (*:these EPROMs programmed in 2 segments) EPROM — 2K utilities package

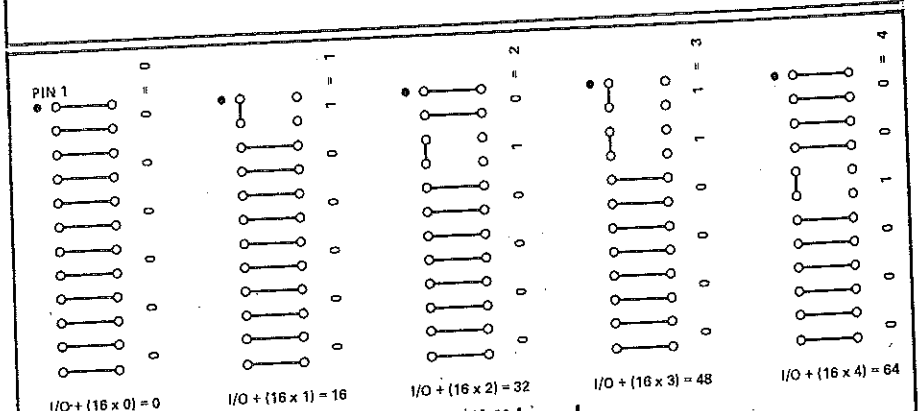
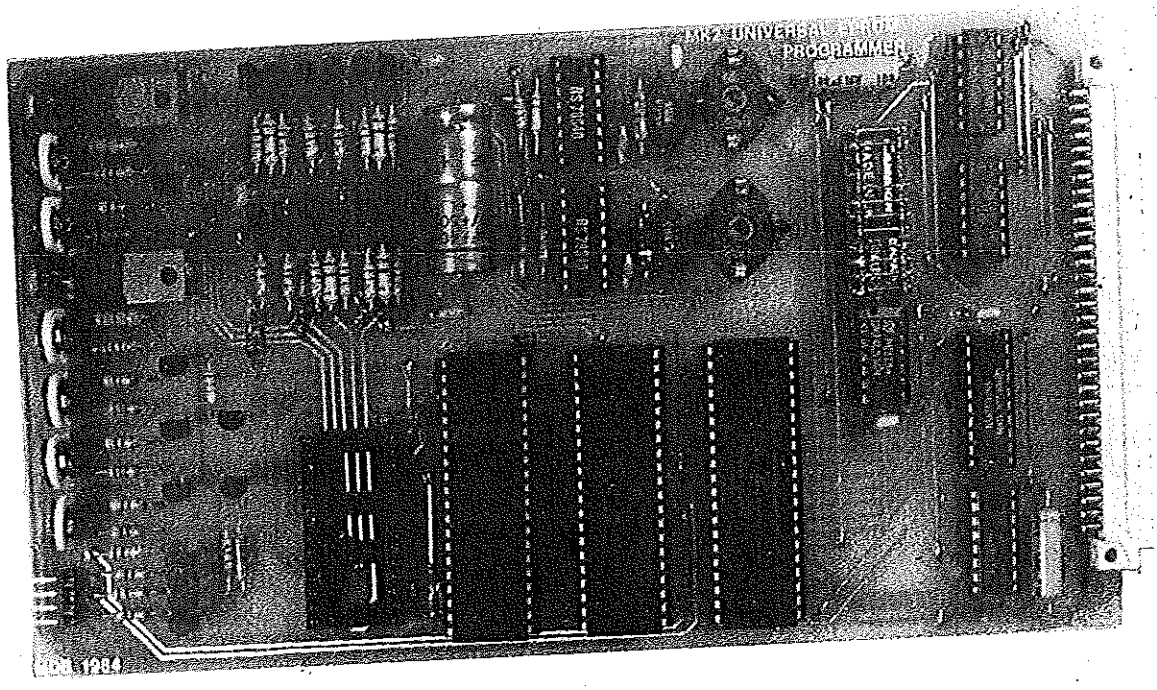


Fig. 2 LK1 address link selection for the MkII board.



PARTS LIST

RESISTORS (All ¼W, 5% unless stated)

R1,11,13,17,18,35, 37,43,45	10k
R2,12,14,20,23,44, 46	1k0
R3,15,16,21,24	100k
R7	15k 2%
R8	680R 2%
R9,39	OR22 W/W
R10	56R, 1W
R41	180R, 1W
R41	180R, 1W
R19,22	240R
R25,29	560R
R26,27	1k2
R28,36,38	470R
R30	82R
R31,32,33,34	4k7
R40	12k
R42	2k2
R47	120R ½W
RV4,8,9	220R vertical miniature preset
RV5,6,7	470R vertical miniature preset

CAPACITORS

C5,7,8,9,10,11,13, 17	100n Ceramic
C3	470μ 35V axial electrolytic
C4,C16	4n7 ceramic
C6,C15	100μ 16V axial electrolytic
C12,C14	1u0 35V tantalum

SEMICONDUCTORS

IC1	74LS126
IC2	74LS245
IC3	74LS04
IC5	74LS138
IC7,8,9	6821 (or 6520 etc)
IC11,14	78S40
IC12,13	LM317MP
Q1,5,7,9,11,13,14, 15,16,17,18,19, 20,21,23,24	BC184L
Q2,6,8,10,12,22	BC214L
D1,2,3,4,5,	OA91

MISCELLANEOUS

L1	34 turns 24 SWG wire on RM6 pot core (AL=250)
L2	13 turns 22 SWG wire on RM6 pot core (AL=250)
SK3	28 pin DIL socket
Connector A	4 way 0.1" pitch right angled molex connector.
Links	Links wired on 24-way 0.3" width DIL header plugged into DIL socket (use 16-way +8-way)

PCB; 1 x 32-way A+B DIN Euro connector, male angled pins.

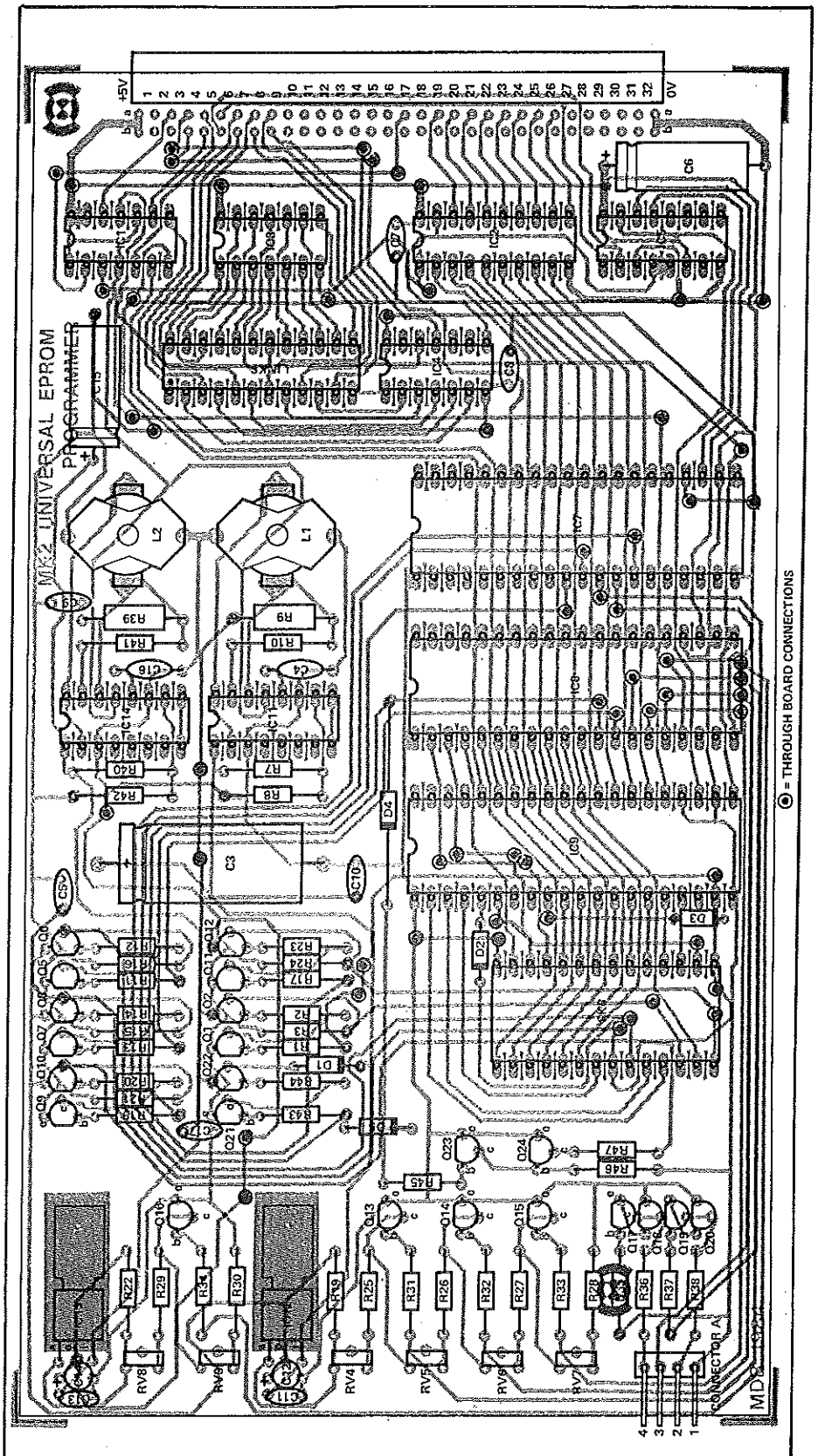


Fig. 3 Overlay diagram of the complete MkII board.

NOTE: Component numbering conforms to original project. R4, R5, R6, C1, C2, IC6, IC10, Q3 and Q4 have not been accidentally omitted. The numbers refer to components which have been removed from the original board in the course of producing the MarkII board.

To be continued.

ETI

UNIVERSAL EPROM PROGRAMMER MKII

In which Gordon Bennett jumps to all the right subroutines so that you can blow your EPROMs and blow your minds with the software for Mike Bedford's better programmer.

When the original articles for the Universal Eprom programmer were published, I was in the process of looking for a new Eprom blower, as the one I was using was horrendously slow. It was a much modified serial driven device originally published in the December 1978 'Computing Today', when this was still a supplement given away in ETI.

However, it was several months before I embarked on the construction of the Universal Eprom Programmer board. It then became apparent that the control software was somewhat unwieldy in its form of both a machine code and BASIC program and that something easier to load and use was required. This prompted me to write a suitable control Eprom for use in the spare slot at E800h in the Microtan memory map.

This Eprom eventually found its way into the hands of Mike Bedford and led to a phone call in which he asked if I would be interested in writing the software for a new enhanced version of the programmer that was under development. The new programmer was to be capable of supporting the interactive programming algorithms which allow the larger devices to be programmed in much reduced times, a 27128 in about 2 minutes and a 27512 in 7 minutes.

The resulting program is described in this article. An idea of the Eproms supported can be gained from Table 1, which also gives a list of those that have actually been programmed using the new hardware and software.

One reason for creating the software package in an Eprom is that it saves considerable time not having to load both a BASIC and machine code program from tape. Those with discs will not find this so much of a problem, of course. Another reason for using an Eprom based package is the efficient memory utilisation. There never seems to be enough memory available even when you have a lot. People with small memories (Sorry! people with small computer memories) are in an even worse position. It is annoying to have a large chunk of your computer memory taken up by the

EPROM TYPE	SOFTWARE SUPPORTED	ALGORITHM FAST/SLOW	WHETHER PROGRAMED
2758	YES	S	NO
2716	YES	S	YES
2516	YES	S	NO
2732	YES	S	YES
2732A	YES	S	
2532	NO		YES
68732	YES	F/S	NO
2764	YES	F/S	NO
2764A	YES	F/S	
2564	NO		YES
68764	YES	F/S	NO
27128	YES	F/S	YES
27128A	YES	F/S	NO
27256	YES	F/S	
27512	NO		
27513*	NO		
2816**	NO		
2864			

* The 27513 is selectable in four 16K banks, each of which is programmed as if it were a 16K Eprom in its own right. Although the programmer software will handle 16K Eproms there is no ability built into it to allow to allow the bank selection mechanism to operate.

** The 2816 Eprom requires only a short pulse to initiate the internal programming cycle followed by a delay of 10ms to allow completion. The software does not support this as it stands but would need minimal changes to allow the use of this device.

Table 1 EPROMs supported by the programmer.

control programs. It often means making two passes when programming.

Ease of use is of prime importance in a package of this type and to this end it has been made as simple as possible to move back and forth between Eprom programmer software and Tanbug monitor. The programmer software has both a cold and warm start vector, the cold start is at E800h and the warm start is at E803h.

The software should always be entered the first time at E800h as this performs the initialisation of the PIOs. If it becomes necessary to leave the programmer, when developing software with an assembler or using the Tanbug monitor facilities, then re-entry is via E803h.

The provision of memory modify and list commands was not necessary. The ease of movement in and out of the program makes it simple to use the Tanbug and X-bug monitor commands for modification, listing and disassembly. There is one command, however, that is useful for displaying the contents of memory, both on the screen and on a printer: the Dump command. It was developed as an aid to give hexadecimal printer dumps of areas of memory, during Eprom development.

The software actually implements three different Fast programming algorithms depending on the type of Eprom being programmed. For the 2764 and 27128 the flowchart is similar to the one featured in the original article (ETI, August 1983). The 27256 and 27512 are slightly different and this is reflected in the flowcharts of the algorithms for these two Eproms (Figs. 1 and 2). The method used by the 27512 should be quicker than that for the 27256, and approximate times for those Eproms programmed so far are shown in Table 2.

Points Of Note

The present package (EP3V75) will support the new hardware for both fast and slow methods of programming. It will *not* support the original hardware as I/O bits are assigned to the PIO ports in a different way. Whilst the package will work with Tanbug V2.3 and V3.1 it will not work with V1.0, because of the way in which the system routine calls are vectored through the jump table at the beginning of Tanbug.

In a 2K package such as this it is not possible to include all desirable features — particularly totally comprehensive error checking. Providing a sensible approach, the error checking should be adequate. Entering only the command letter when parameters are required will cause the command action to operate on the first byte at the present Base. Entering parameters in the wrong order will be trapped and a '?' displayed.

Testing a 27512 Eprom must be done in two 32K parts, and will produce two 'EPROM ERASED OK' messages, approximately 20 seconds apart. This is not an error — honest!

EPROM TYPE	SIZE	PROGRAM SLOW	FAST	TEST	VERIFY
MEASURED TIMES					
2716	2K	1.75m	—	1.25s	1.25s
2732	4K	3.5m	—	2.5s	2.5s
2532	4K	3.5m	—	2.5s	2.5s
2764	8K	7.5m	1m	5s	5s
27128	16K	15.5m	2m	10s	10s
27256	32K	30m	4m	20s	20s
ESTIMATED TIMES					
2758	1K	0.8m	—	0.6s	0.6s
27512	64K	60m	7m	41s	41s

Table 2 Approximate programming speeds.

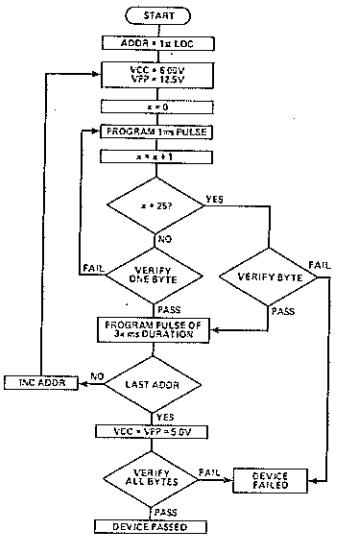


Fig. 1 Flowchart for 27256 programming routine.

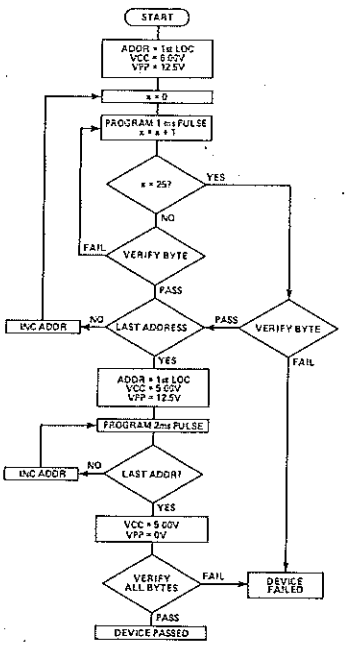


Fig. 2 Flowchart for 27512 programming routine.

Hardware Configuration

The program assumes that the Eprom Programmer board is based at address BC20h. The original Microtan screen is assumed for obtaining parameters, and in the clear screen routine. Locations in zero page from 45h to 5Fh, are used. The Eprom programming software re-initialises the locations it needs when entry is made at E800h. If other user programs are in memory at the same time occupying any of these locations the contents will be overwritten.

Menu And Commands

The program us started from the Tanbug monitor by typing GE800<CR>. You are then asked to enter the type of Eprom that you wish to program and the base address at which your object code resides in memory. You should then see a display of current Eprom type, current base address and the menu of available commands.

This screen display is shown below. It can be obtained at any time, when not actually executing a command, by pressing the 'H' key.

```
EPROM=XXXXX  BASE=$HHHH

(H)elp
(Q)uit
(T)est
(R)ead
(V)erify
(F)ast prog.
(S)low prog.
(D)ump
(N)ew type
(B)ase
```

The command input format uses the capital initial letter of the command, as shown on the help menu. Some commands — 'Help', 'Quit' and 'Test' — execute immediately. Others, like 'New type' and 'Base', prompt for a further input. The rest require a parameter range. The normal format is:

X,ssss,ffff
(X is the start address in hex and ffff is the finish address in hex).
The delineator can be any non alphanumeric.

Unlike the old version of the software the range parameters do not require leading zeros. All of the following are valid commands:

R,0,7FF or R,0000,7FF reads 0 to 7FF.
S,801,802 slow programs 801 and 802.
F,FFF,FFF fast programs location FFF.
D/0/2FF and D—100,1FF both dump to the screen.

All commands that act on the Eprom socket also turn off the green LED and, in case of a programming command, turn on the red LED.

If the range of the parameters entered is too big for the Eprom type selected, the message 'EPROM SIZE EXCEEDED' will be displayed.

120 Command Description

An explanation of the com mands may be useful since there are many new features.

HELP:- gives a display of the current Eprom type, base address and the commands available, exactly as it you had just entered the program from Tanbug.

QUIT:- takes you back into the Tanbug monitor and resets the stack, after making the PIO outputs safe.

TEST:- examines the Eprom to see if all locations contain FFh. If they do, you should get a message 'EPROM ERASED OK'. Otherwise you will get a display of the addresses and contents. If there are more than fourteen locations not containing FFh, the program waits for you to press 'CR' before displaying the next lines. To abandon the display press SPACE BAR and you will be returned to the menu.

READ:- reads into memory the contents of an Eprom currently in the programming socket. Requires a parameter range.

VERIFY:- verifies that the Eprom holds the same code as the object code in memory at the current base address. If not, the code in both the Eprom and the memory will be displayed. Like the errors reported in the Test mode, this will be shown fourteen lines at a time, 'CR' will show the next screen full and the SPACE BAR will return you to the menu. Successful verification will produce the message, 'EPROM VERIFIED OK'. Requires a parameter range.

FAST PROG.:- invokes the fast programming mode for Eproms of 8k and larger. With a smaller Eprom currently in the programming socket, it will automatically default to the slow mode to avoid damage. After successful programming there is an automatic verify of the whole range programmed, indicated by the message 'EPROM PROGRAMMED, VERIFYING', which gives way to the 'EPROM VERIFIED OK' report on completion. Lights red LED. Requires a parameter range.

SLOW PROG.:- the mode for programming Eproms smaller than 8K. With an Eprom of 8K or larger, selection of this mode allows programming in the standard way. This allows a certain degree of flexibility, since you can program Eproms of uncertain size with a tried and tested method. The messages used are the same as for the Fast mode. Lights red LED. Requires a parameter range.

DUMP:- gives a hex dump to the screen and printer in the following format: the absolute address in memory followed by the relative address from the start of the dump, then sixteen hex bytes of data and finally a two byte check sum computed from the previous sixteen data bytes. To get printer output, enter the command and parameters then press CTRL-P before the carriage return. Don't forget to turn off printer control afterwards, with another CTRL-P, or the program will appear to 'hang' for 10 seconds, whilst the print output routine times out. Requires a parameter range.

NEW TYPE:- the command to change the type of Eprom you are working with.

BASE:- the means of changing the start address in the memory to that of any new object code.

Way Out

A common feature of all the commands that require parameters, is that the command sequence can be aborted at any time before pressing carriage return by use of the SPACE BAR.

Without using the 'QUIT' command, the program can be left by pressing either the 'ESC' key or the 'RESET' button. Neither is recommended, since they both interrupt commands at indeterminate points. The 'ESC' key is particularly bad as it will leave the programming socket with power and signals present. If 'RESET' is used it will be necessary to restart the programmer software via the cold start vector. The reset is also issued to the PIO's, setting the ports to a safe state. They will then need to be re-initialised before they can be used again. It should be obvious when the socket is unsafe, because the green LED will not be lit.

The best method of exit is the 'QUIT' command which can only be issued when the programmer is in a safe state and which allows faster re-entry via the warm start vector at E803h.

If you should chance to use 'ESC' or 'RESET', re-entry to the program will re-initialise the PIO ports and set the socket to a safe state.

Waveform Diagrams

The outputs of the programmer hardware during fast programming change rapidly compared to those resulting from the 50ms pulses of the slow programming mode.

Figures 3, 4 and 5 show the programming voltage VPP, the

supply voltage VCC and the actual program pulse, NPGM (Not PGM), as they appear on a oscilloscope for a number of different Eproms in fast mode.

Figure 3 shows the waveforms for the 2764 and 27128. The 2764A and 27128A are the same, except that VPP is only 12.5 volts.

Figure 4 is the diagram for the 27256 and shows the effect of having chip enable (CE) on the same pin as the programming pulse.

Figure 5 shows the waveforms for the 27512, in which output enable is on the same pin as the programming voltage, and as with the 27256, CE and PGM share a pin.

Future Developments

The author's system includes a TUG Eprom Storage Card (ECS) and he is currently developing a 4K software package with enhanced error checking and additional routines for use with this. Features planned include support for other devices, new utilities, such as memory fill and relocation, and access to programs on the ESC and disc.

Currently being developed is a hardware interface to connect the programmer to a BBC Microcomputer and a sideways rom to allow its use. The hardware has actually been finished and tested. The sideways rom is in mid development. *(Keep watching ETI — Ed.).*

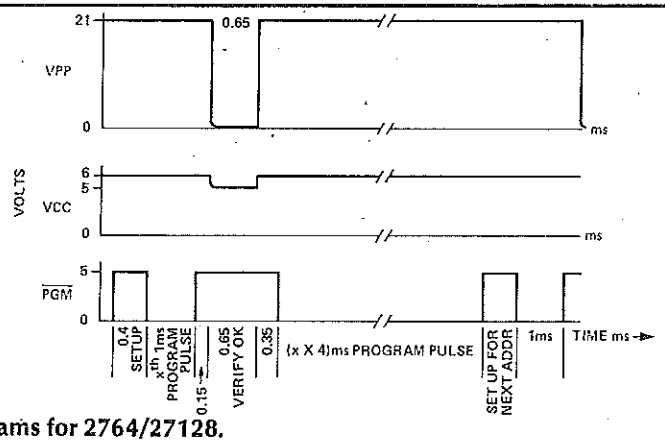


Fig. 3 Timing diagrams for 2764/27128.

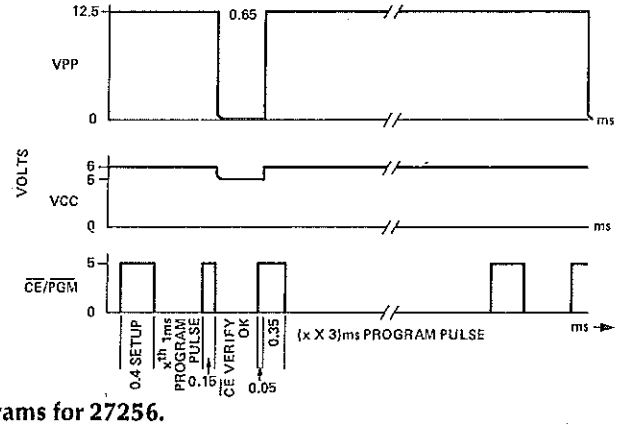


Fig. 4 Timing diagrams for 27256.

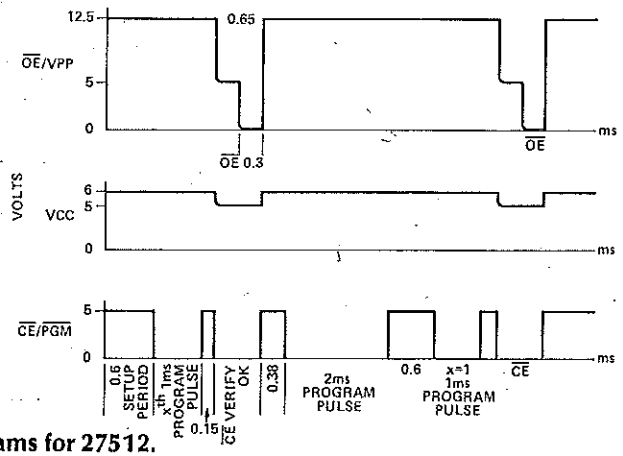


Fig. 5 Timing diagrams for 27512.

HOW IT WORKS

On entry through the cold start the PIOs are initialised to the correct inputs and outputs and then zeroed to a safe state. The header message is displayed and the type of Eprom to be operated on is requested along with the base address in memory where the object code resides. Then the help menu is displayed and the software waits for an input.

The program runs a background loop (GETCMD) waiting for characters typed at the keyboard. When an input is received the character is checked against a list of valid commands. On finding one, a jump is made to the corresponding subroutine, otherwise the program simply returns to the background loop.

Entry through the warm start vector does not initialise the PIOs or ask for Eprom type and address, as these are assumed. Instead, the menu is displayed and control passes directly into the background loop. This is intended for a quick return to the program after using the Tanbug monitor facilities. If a program has been run, such as a two pass assembler, which might have corrupted zero page locations used by the Eprom programmer software, it would be wise to return through the cold start vector.

Immediately after the two vectors (at E800h and E803h) in the Eprom there are the tables used for setting up the PIO ports in the configurations required for the functions and Eproms supported. To add other Eprom types to the software these tables would have to be extended and further entries made in the type and length parameters stored from E861h and E8C9h. This is not easy without a full disassembled listing and the use of a two pass assembler. There is no further space available in the current Eprom, so something would have to be removed, which should be no great problem as all subroutines are modular.

The screen clear routine (CLRSCN) is only called twice, from the header at E9CFh and by the help routine at EA23h. It will not work with 80 column boards. It is the last subroutine in the Eprom at EFECh.

BUYLINES

For those not wishing to type in the code from the hex dump, a ready programmed Eprom complete with assembler listing is available for £10.00, from G. J. Bennett, 35 Fowler Road, Cove, Farnborough, Hants. Alternatively, the source code (for use with the TUG two pass assembler/editor) and object code on tape are available for £5.00 from the same address. Please allow 28 days for delivery.

The hex dump and full disassembled listing will appear in ETI next month.

UNIVERSAL EPROM PROGRAMMER MKII

This hex dump spells a fully operational EPROM Programmer — and you can't byte that. Gordon Bennett performed the magic.

Last month, lack of space obliged us to hold over the software listings for the EPROM Programmer. This month we remedy that omission with a complete hex dump of the Universal EPROM Programmer software. Unfortunately, space reasons forbid the publication of the complete disassembled listings — which runs to some 26 pages of print-out.

The hex dump provides everything you need in order to enter the code to run the programmer. Notes on the various locations involved can be found in last month's introductory article and below. If anyone should require the disassembled listing, for the purposes of modification or out of simple interest, we can provide a photocopy at a price of £3. Please send a cheque, made out to ASP Ltd., to ETI, Dept. UEP, 1 Golden Square, London W1R 3AB, and enclose a large stamped addressed envelope. The author will provide a ready programmed EPROM plus listing for £10.00 or a tape including source and object code (for use with the TUG two pass/assembler editor) for £5.00. Please write to 187 Beaulieu Gardens, Blackwater, Camberley, Surrey GU17 0LF. Note that this address is different from the one published last month, and is the correct address. Please allow 28 days for delivery on any of these items.

The following hex dump was performed using the DUMP command in the EPROM Programmer software. The first address on a line is the absolute address in memory, the second is the offset address from the current Base selected. Then there are 16 bytes of data and the final number is the checksum computed from that line of data. With reference to last month's notes, the screen clear routine is located at EFEC h to EFFF h, inclusive. It may be removed for the addition of extra facilities in the program. It is called twice, by the command JSR CLRSCRN (the hex bytes 20 EC EF), at locations E9CF h and EA23 h (the header and the help routines). Locations E861 h through E8C8 h contain the type numbers of the EPROMs supported by the software. Length parameters are contained in the locations E8C9 h through E8D5 h. Messages used in the programme are contained in locations E8F6 h through E9C h. The table used to set up PIO ports for the EPROMs supported are contained in locations E806 h through E860 h. These tables could be altered to support different EPROMs, but — once again — close study of the disassembled listing is recommended.

The hex dump from E800 h to EFFF h.

```
E800 0000  4C D8 E9 4C E7 E9 3C 3C 3C 3C 3C 3C 34 34 34 34  0661
E810 0010  34 34 34 98 98 98 90 90 98 92 92 90 92 92 90 90  0814
E820 0020  18 18 18 08 08 18 10 10 10 10 10 10 00 88 88 88  0268
E830 0030  85 A5 88 22 C2 00 22 C2 C1 C5 01 01 01 01 01 10  0515
E840 0040  02 02 10 02 02 01 01 06 06 06 06 06 0A 06 06 0A  0058
E850 0050  06 06 06 06 01 01 01 01 01 00 01 01 00 01 01 01  0022
E860 0060  01 32 37 35 38 20 20 20 20 32 37 31 36 20 20 20  0287
E870 0070  20 32 35 31 36 20 20 20 20 32 37 33 32 20 20 20  029C
E880 0080  20 32 37 33 32 41 20 20 20 32 35 33 32 20 20 20  02BB
E890 0090  20 32 37 36 34 20 20 20 20 32 37 36 34 41 20 20  02C7
E8A0 00A0  20 32 35 36 34 20 20 20 20 32 37 31 32 38 20 20  02B5
E8B0 00B0  20 32 37 31 32 38 41 20 20 32 37 32 35 36 20 20  02EB
```

```
E8C0 00C0  20 32 37 35 31 32 20 20 20 03 07 07 0F 0F 0F 1F  01DE
E8D0 00D0  1F 1F 3F 3F 7F FF 48 51 54 52 56 46 53 4E 44 42  053C
E8E0 00E0  23 EA 87 EB 8D EB 28 ED DF EB 88 ED 84 ED 26 EB  0ACD
E8F0 00F0  38 EC 68 EB EA E9 45 50 52 4F 4D 20 54 59 50 45  072F
E900 0100  00 42 41 53 45 20 41 44 44 52 45 53 53 20 28 48  03D1
E910 0110  45 58 29 00 45 50 52 4F 4D 20 50 52 4F 47 52 41  0434
E920 0120  4D 4D 45 44 2C 20 56 45 52 49 46 59 49 4E 47 2E  0450
E930 0130  00 45 50 52 4F 4D 20 43 48 45 43 4B 20 46 49 4E  03FE
E940 0140  49 53 48 45 44 00 45 50 52 4F 4D 20 56 45 52 49  0446
E950 0150  46 49 45 44 20 4F 4B 00 45 50 52 4F 4D 20 45 52  040C
E960 0160  41 53 45 44 20 4F 4B 00 45 50 52 4F 4D 20 53 49  0416
E970 0170  5A 45 20 45 58 43 45 45 44 45 44 00 20 20 20 20  0376
E980 0180  20 20 20 00 45 52 52 4F 52 20 40 20 24 00 20 4D  02FB
E990 0190  45 4D 2E 3D 00 20 45 50 52 4F 4D 3D 00 20 42 41  0380
E9A0 01A0  53 45 20 3D 20 24 00 0D 20 20 20 2D 20 45 50 52  02DA
E9B0 01B0  4F 4D 20 55 54 49 4C 49 54 49 45 53 20 56 33 2E  044F
E9C0 01C0  37 35 2D 0D 0D 0D 45 6E 74 65 72 3A 2D 0D 00 20  0352
E9D0 01D0  EC EF A0 B1 20 6B EF 60 20 C2 EA 20 DF EA 20 CF  09AA
E9E0 01E0  E9 20 26 EB 20 68 EB 20 23 EA EA 20 DF EA 20 14  07C1
E9F0 01F0  EB 20 1D F8 A5 01 48 20 0E F8 68 A2 00 DD D6 E8  07D9
EA00 0200  F0 08 E8 E0 0A D0 F6 4C EB E9 8A 0A AA BD E0 E8  0A73
EA10 0210  85 5C E8 BD E0 E8 85 5D AD F5 E8 48 AD F4 E8 48  0AD3
EA20 0220  6C 5C 00 20 EC EF 20 0C F8 A0 9F 20 6B EF A9 00  0749
EA30 0230  85 53 A5 52 0A 0A 0A AA BD 61 E8 20 0E F8 E8 E6  0791
EA40 0240  53 A5 53 C9 08 90 F1 A0 A7 20 6B EF A5 4B 20 1A  0788
EA50 0250  F8 A5 4A 20 1A F8 A2 00 BD 67 EA C9 00 F0 07 20  07A9
EA60 0260  0E F8 E8 4C 58 EA 60 0D 0D 28 48 29 65 6C 70 0D  05DD
EA70 0270  28 51 29 75 69 74 0D 28 54 29 65 73 74 0D 28 52  0479
EA80 0280  29 65 61 64 0D 28 56 29 65 72 69 66 79 0D 28 46  04A1
EA90 0290  29 61 73 74 20 50 72 6F 67 2E 0D 28 53 29 6C 6F  04E3
EAA0 02A0  77 20 50 72 6F 67 2E 0D 28 44 29 75 6D 70 0D 28  0486
EAB0 02B0  4E 29 65 77 20 74 79 70 65 0D 28 42 29 61 73 65  050E
EAC0 02C0  0D 00 A9 30 8D 21 BC 8D 23 BC 8D 27 BC 8D 2B BC  06A0
EAD0 02D0  A9 FF 8D 20 BC 8D 22 BC 8D 26 BC 8D 2A BC 60 A9  0867
EAE0 02E0  04 8D 27 BC 8D 2B BC 8D 2A BC A9 90 8D 26 BC A9  07AC
EAF0 02F0  34 8D 21 BC 8D 23 BC A9 00 8D 25 BC 8D 20 BC 8D  0717
EB00 0300  22 BC A9 FF 8D 24 BC A9 04 8D 25 BC A9 00 8D 24  0768
EB10 0310  BC 85 48 60 20 0C F8 A9 3E 20 0E F8 60 20 0C F8  069E
EB20 0320  A9 24 20 0E F8 60 20 0C F8 A0 00 20 6B EF 20 14  05C5
EB30 0330  EB 20 C4 EF 20 E6 EF A2 00 86 52 A0 01 B1 0A C9  0852
EB40 0340  20 F0 24 DD 61 E8 D0 05 C8 E8 4C 3D EB E6 52 A5  0930
EB50 0350  52 0A 0A 0A C9 68 B0 04 AA 4C 3B EB 20 0C F8 A9  063E
EB60 0360  3F 20 0E F8 4C 2E EB 60 20 0C F8 A0 0B 20 6B EF  0673
EB70 0370  20 1D EB 20 C4 EF A0 00 20 17 F8 A5 13 85 4A A5  06F6
EB80 0380  14 85 4B 20 0C F8 60 20 0C F8 4C 20 F8 A9 00 85  061E
EB90 0390  53 85 4C 85 4D A9 FF 85 4E A6 52 BD C9 E8 E0 0C  08C3
EBA0 03A0  D0 0B A9 7F 20 AD EB A9 80 85 4D A9 FF 85 4F 20  0852
EBB0 03B0  CC EC A9 01 85 51 20 35 ED 20 44 EE D0 06 A0 62  07A4
EBC0 03C0  20 6B EF 60 A0 8E 20 6B EF 20 7A EF A0 9F 20 6B  07D5
EBD0 03D0  EF 20 98 EF 20 A3 EF D0 03 20 AA EF 4C B6 EB 20  08E1
EBE0 03E0  99 EC 20 CC EC 20 59 EF A9 80 85 51 A9 00 85 47  0839
EBF0 03F0  85 53 20 35 ED 20 44 EE D0 13 20 DF EA A5 47 F0  0814
EC00 0400  06 A0 3B 20 6B EF 60 A0 50 20 6B EF 60 20 1F EC  06B0
EC10 0410  A9 01 85 47 20 A3 EF D0 03 20 AA EF 4C F2 EB A0  087D
EC20 0420  8E 20 6B EF 20 7A EF A0 98 20 6B EF 20 90 EF A0  0882
EC30 0430  9F 20 6B EF 20 98 EF 60 20 99 EC 20 CC EC 20 0C  07C9
EC40 0440  F8 A9 00 85 53 85 5E 85 5F 20 32 EE F0 25 20 85  073A
EC50 0450  EF 20 E6 EF 20 7A EF 20 E6 EF 20 E6 EF A2 00 A1  099A
```

```
EC60 0460  45 48 20 8D EC 68 20 1A F8 20 86 EC F0 06 20 32  069A
EC70 0470  EE D0 E7 60 A0 8B 20 6B EF A5 5F 20 1A F8 A5 5E  08E3
EC80 0480  20 1A F8 4C 3E EC E6 53 A5 53 C9 10 60 18 65 5E  06ED
EC90 0490  85 5E A5 5F 69 00 85 5F 60 20 C4 EF A0 02 20 17  0640
ECA0 04A0  F8 A5 13 85 4C A5 14 85 4D 20 17 F8 A5 13 85 4E  06C6
ECB0 04B0  A5 14 85 4F C5 4D 90 09 D0 06 A5 4E C5 4C 90 01  06A3
ECC0 04C0  60 20 0C F8 A9 3F 20 0E F8 4C C1 EF 38 A5 4C E9  07A0
ECD0 04D0  01 85 55 A5 4D E9 00 85 56 18 A5 4E 69 01 85 57  05E2
ECE0 04E0  A5 4F 69 00 85 58 18 A5 4A 65 55 85 45 A5 4B 65  061A
ECF0 04F0  56 85 46 20 0C F8 60 A6 52 BD 06 E8 8D 23 BC A9  075D
ED00 0500  00 8D 20 BC A9 04 8D 25 BC BD 13 E8 8D 26 BC A9  0754
ED10 0510  12 8D 2A BC 60 AD 2A BC 49 10 8D 2A BC 20 4F EE  06A1
ED20 0520  AD 24 BC 85 50 A2 00 60 20 99 EC 20 CC EC 20 59  075A
ED30 0530  EF A9 00 85 51 A2 00 20 55 EE 20 F7 EC 20 4F EE  07D3
ED40 0540  20 32 EE F0 1F 20 D1 ED 20 15 ED A4 51 F0 10 30  0774
ED50 0550  07 C9 FF F0 E5 4C 64 ED C1 45 F0 DE 4C 64 ED 81  0A33
ED60 0560  45 4C 3A ED 60 A6 52 BD 06 E8 8D 23 BC A9 00 8D  075D
ED70 0570  20 BC A9 04 8D 25 BC BD 20 E8 8D 2A BC BD 2D E8  0801
ED80 0580  8D 26 BC 60 A9 01 85 48 20 99 EC 20 CC EC 20 59  073C
ED90 0590  EF A5 52 C9 06 90 07 A5 48 D0 03 4C 7E EE A2 FF  0865
EDA0 05A0  20 55 EE 20 65 ED 20 21 EF 20 4F EE 20 32 EE F0  0792
EDB0 05B0  1D 20 D1 ED A2 00 A1 45 8D 24 BC 20 63 EE A0 1D  071E
EDC0 05C0  A2 FF CA D0 FD 88 D0 FA 20 63 EE 4C A3 ED 4C F5  0B18
EDD0 05D0  EE A5 55 8D 22 BC BD 54 E8 F0 08 A5 56 8D 20 BC  08A8
EDE0 05E0  4C 07 EE AD 20 BC 05 59 8D 20 BC A5 56 29 10 F0  06B5
EDF0 05F0  08 AD 20 BC 09 08 8D 20 BC A5 56 29 08 F0 08 AD  05DC
EE00 0600  26 BC 09 01 8D 26 BC A5 56 29 20 F0 08 AD 23 BC  0623
EE10 0610  09 08 8D 23 BC A5 56 29 40 F0 08 AD 26 BC 09 02  0573
EE20 0620  8D 26 BC A5 56 29 80 F0 08 AD 20 BC 09 80 8D 20  06CA
EE30 0630  BC 60 E6 45 D0 02 E6 46 E6 55 D0 02 E6 56 A5 56  0889
EE40 0640  29 E7 85 59 A5 55 C5 57 D0 04 A5 56 C5 58 60 A0  07F0
EE50 0650  18 88 D0 FD 60 A9 00 8D 25 BC 8E 24 BC A9 04 8D  078C
EE60 0660  25 BC 60 A6 52 BC 47 E8 B9 20 BC 5D 3A E8 99 20  07F1
EE70 0670  BC 60 A2 FF 20 55 EE 20 65 ED 20 19 EF 60 A2 00  07BC
EE80 0680  86 5A E8 86 5B 20 32 EE 20 44 EE F0 68 20 72 EE  0813
EE90 0690  20 4F EE 20 D1 ED A2 00 A1 45 8D 24 BC 20 08 EF  0747
EEA0 06A0  A6 5A E8 86 5A A6 52 BD D9 EF C5 5A F0 05 20 42  08BB
EEB0 06B0  EF D0 DA 20 72 EE 20 4F EE 20 D1 ED A5 5A 0A 0A  0867
EEC0 06C0  A8 A6 52 E0 0C D0 05 A0 02 4C D8 EE BD D9 EF C9  0963
EED0 06D0  0F F0 05 98 38 E5 5A A8 84 5B 20 08 EF A6 52 BD  0766
EEE0 06E0  D9 EF C5 5A D0 98 20 42 EF F0 93 20 39 EF 20 DF  096A
EEF0 06F0  EA 20 1F EC 60 20 39 EF AD 2A BC 29 12 8D 2A BC  06FE
EF00 0700  A0 1E 20 6B EF 4C E2 EB 20 63 EE A4 5B A2 90 CA  08BD
EF10 0710  D0 FD 88 D0 F8 20 63 EE 60 AD 26 BC 29 7F 8D 26  08D8
EF20 0720  BC A5 52 C9 0C F0 09 AD 2A BC 09 1B 8D 2A BC 60  070B
EF30 0730  AD 2A BC 09 1A 8D 2A BC 60 AD 26 BC 09 80 8D 26  0654
EF40 0740  BC 60 A2 00 20 55 EE 20 F7 EC 20 39 EF 20 4F EE  07C9
EF50 0750  20 D1 ED 20 15 ED C1 45 60 A6 52 BD C9 E8 C5 4F  08E0
EF60 0760  B0 08 A0 72 20 6B EF 4C C1 EF 60 B9 F6 E8 C9 00  0900
EF70 0770  F0 07 20 0E F8 C8 4C 6B EF 60 A5 56 20 1A F8 A5  07BD
EF80 0780  55 20 1A F8 60 A5 46 20 1A F8 A5 45 20 1A F8 60  0680
EF90 0790  A0 00 B1 45 20 1A F8 60 A0 00 A5 50 20 1A F8 20  060F
EFA0 07A0  0C F8 60 E6 53 A5 53 C9 0E 60 20 1D F8 A5 01 C9  0770
EFB0 07B0  20 F0 0E C9 0D D0 F3 A9 00 85 53 20 0C F8 60 68  0724
EFC0 07C0  68 68 68 60 A2 00 20 1D F8 E8 A5 01 C9 20 F0 EF  07C5
EFD0 07D0  90 06 20 0E F8 4C C6 EF 60 00 00 00 00 00 00 0F  042C
EFE0 07E0  19 0F 0F 19 19 19 A9 20 20 0E F8 60 A0 00 A9 20  043A
EFF0 07F0  99 00 02 99 00 03 C8 D0 F7 84 0A A9 02 85 0B 60  05EF
```

ETI