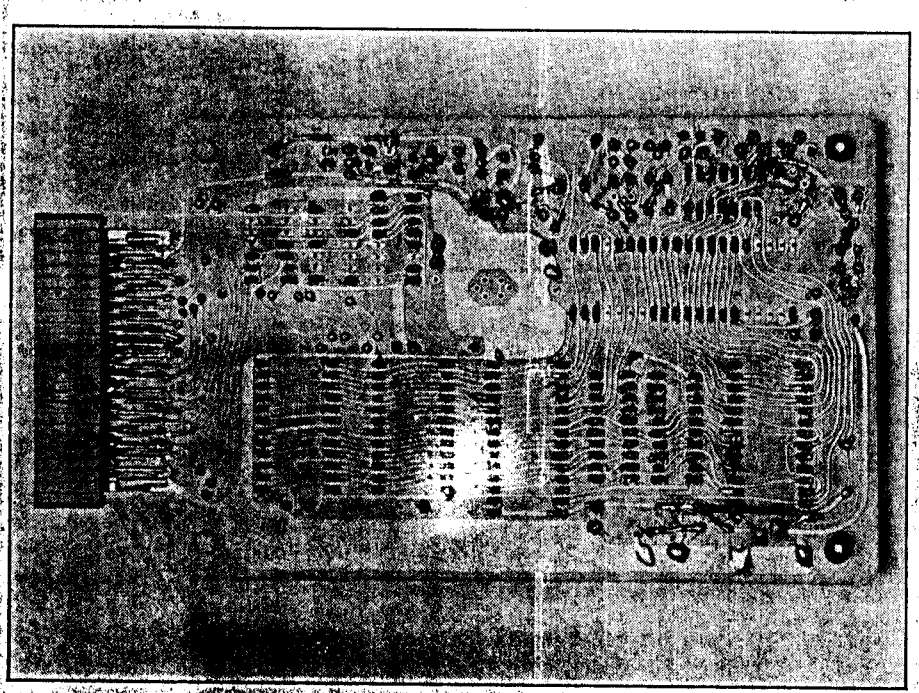
ETI-1611: VZ300 EPROM programmer

Part 1

Customise your computer with this EPROM programmer. This month the hardware, next month, the software.

Herman Nacinovich



FOR ANYONE SERIOUSLY involved with microprocessors or computers, this EPROM programmer will prove to be an invaluable tool. It has lots of features, some of which may only be found in commercial programmers costing much more. Yet it uses relatively few parts, including cheap, readily available IC's and discrete components. Everything is on a single board which plugs directly, or via a ribbon cable plus socket, into the memory expansion slot of a VZ300 computer. Power for the programmer is derived from the internal power supply for the VZ300, thereby saving the cost of having a separate power supply. Also, there is no need for a housing and this represents a further saving in cost.

I designed this EPROM programmer for use with a VZ300 computer for the simple reason that I happen to have a VZ300. Apart from that, however, the choice of a VZ300 for this application has the advantage that it is available at a very attractive price, yet it is more than adequate for the job. In fact, the total cost of this EPROM programmer plus a VZ300 may be less than the cost of a commercial programmer with similar features but without the computer. Thus, if you need an EPROM programmer but don't have a VZ300, it might be worth considering whether the low cost of this computer would justify its purchase for this application. After all, a second computer can always come in handy, can't it?

Among the features built into the EPROM programmer is versatility. This is because most of its operation is under software control. This includes selection of programming voltages appropriate to EPROMs from different manufacturers, modes of data transfer and editing capabilities. There are no switches as these are made unnecessary by virtue of the software programmability.

A ZIF (zero insertion force) socket is provided on the board for a 28-pin EPROM to be programmed. There is provision on the board for an optional, second ZIF socket for a second EPROM which has already been programmed. This allows direct copying from one EPROM to another. In addition, there is provision for an optional 4K of RAM which can be used to extend the internal RAM capacity of the VZ300. This can be useful for editing or for temporarily storing large chunks of machine code before burning them into an EPROM. Also, with 4K of RAM, the board can be used to extend the memory capacity of the VZ300 when it is not used to program EPROMs.

With suitable software, this EPROM programmer can be programmed to do such useful things as verify whether an EPROM has been fully erased before programming, copy from one EPROM to another (as mentioned), transfer data from EPROM to RAM and vice versa, manually enter data temporarily into RAM and editing before transferring to EPROM. One of the good features of this EPROM programmer is that the software can be modified to extend its capabilities without any change to the board.

The programmer is designed primarily for programming 28-pin EPROMs of the 2764, 27128 and 27256 types (and their CMOS equivalents). There are, of course, other types around, but to try to cater for

all available types would require a horrendously complex switching arrangement and an overall cost which could not be justified. Besides, many of the earlier types (such as the 2708) would seem to be obsolete, hard to get and, on top of that, ridiculously expensive. On the other hand, the 2764, 27128 and 27256 EPROM types would seem to be the most popular and useful currently available. Furthermore, they are substantially pin compatible with each other, which simplifies the design of a programmer considerably. With these points in mind, it seems reasonable to limit the design of a programmer for use with these three EPROM types as a compromise between versatility and circuit complexity.

EPROM Characteristics

For those not fully familiar with EPROM characteristics, a general description of these devices may be useful.

All EPROMs of the types with which we are concerned have a set of Address pins, a set of DATA pins and a set of CONTROL pins. The number of address pins reflects the bit capacity of an EPROM. Thus, the 2764 (64K bits) has 13 address pins, the 27128 (128K bits) has 14 address pins and the 27256 (256K bits) has 15 address pins. All EPROMs of this series have eight data pins. That is, data bits are programmed into, and read out of, these devices as 8-bit groups, or bytes.

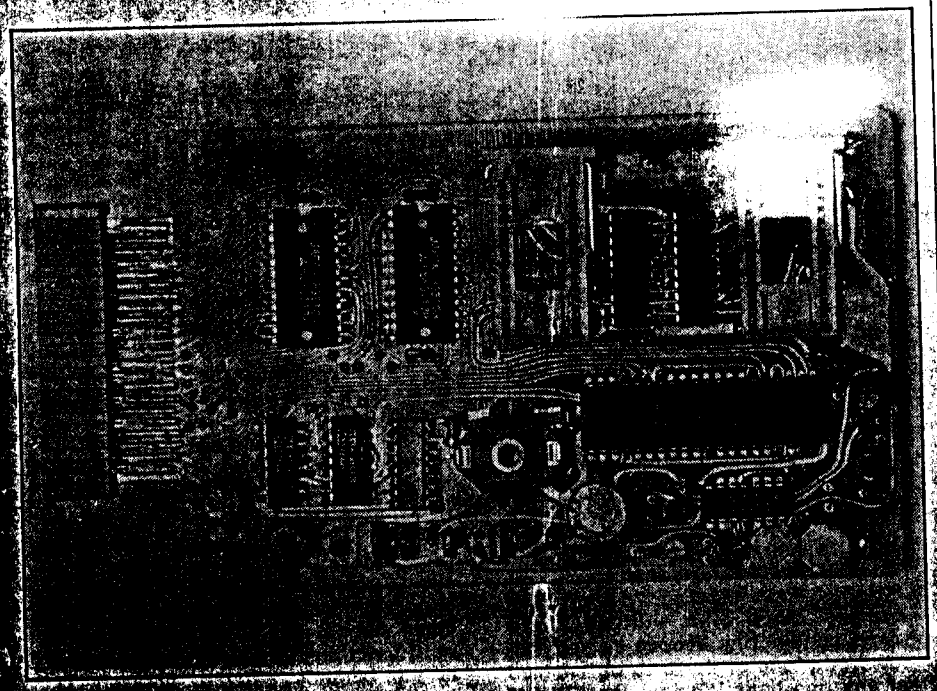
The control pin functions are labelled CE (chip enable), OE (output enable) and PGM (program). The bars over these letters mean that these functions are activated by a logic LOW signal and, conversely, de-activated by a logic HIGH signal, at the respective pins. In the 27256, the CE and PGM functions are combined and accessed at a single pin, while in the other two EPROM types these are associated with separate pins. Incidently, all address and control signals are specified to be at TTL levels.

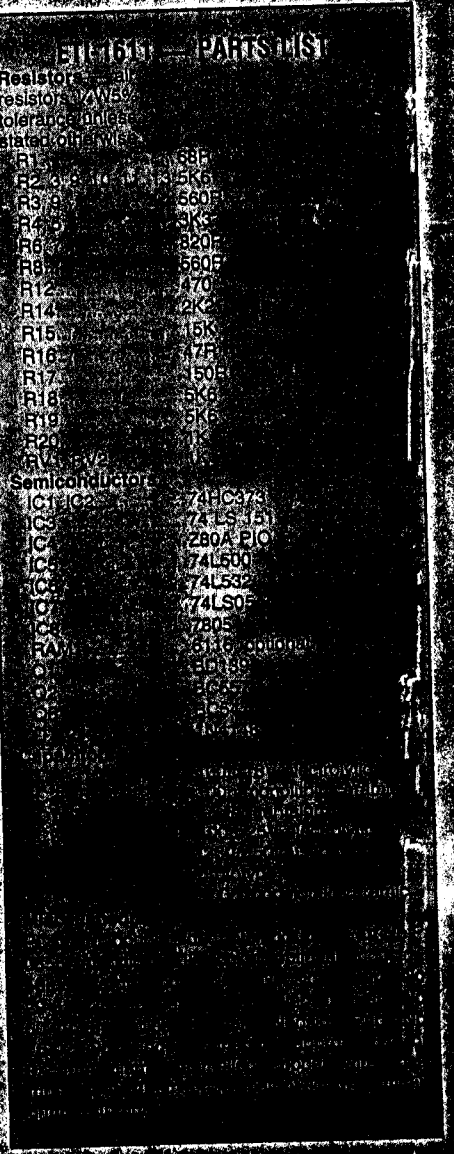
In addition to ADDRESS and DATA pins, these EPROMs have a GROUND (0V supply), Vcc (+5V supply) and Vpp (programming voltage supply). Vpp is specified to be +5V for READ operations and either +12.5V or +21V (typically), depending upon the manufacturer, for PROGRAMMING operations.

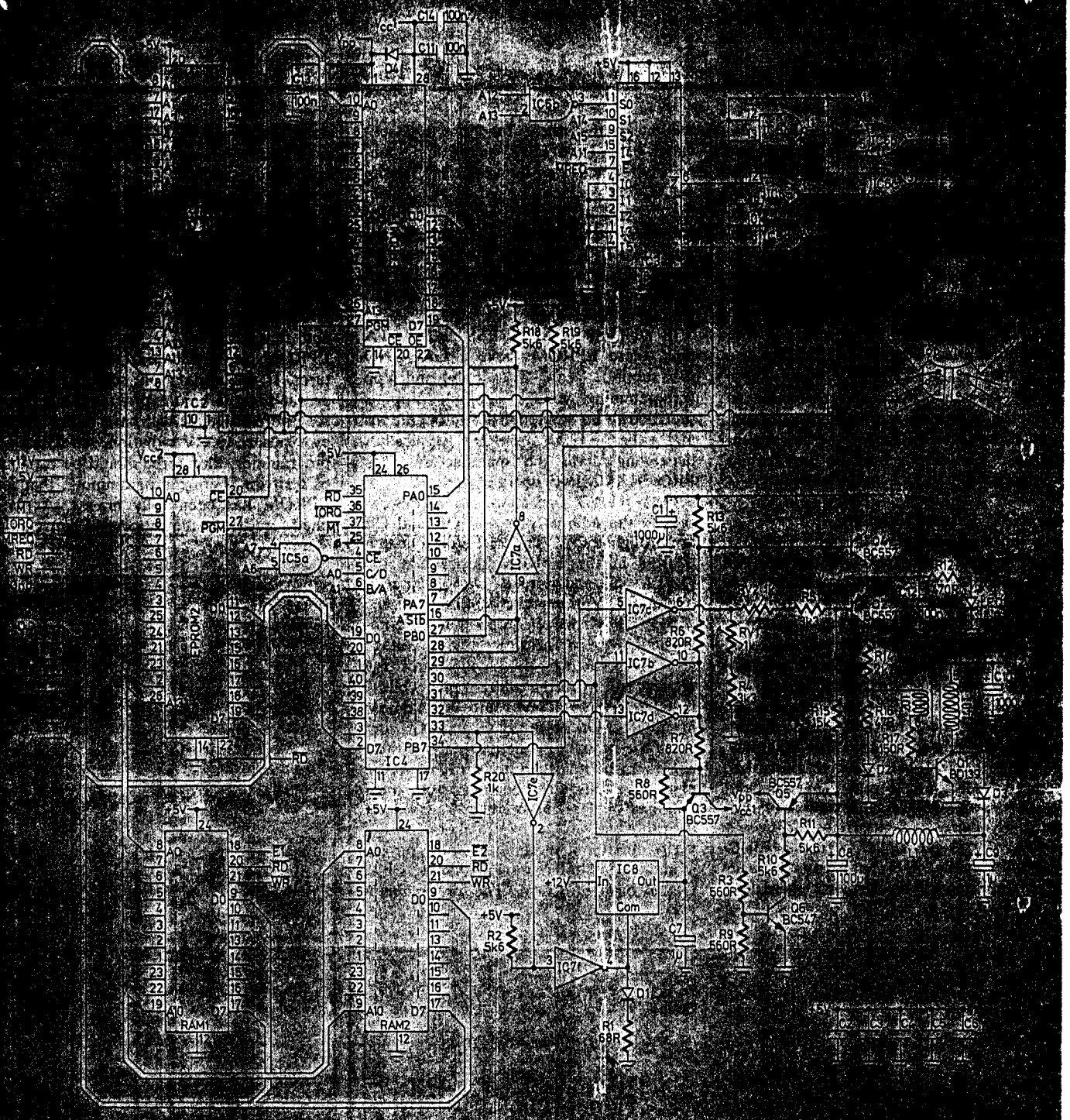
In a READ operation, an address is sent to the address pins and OE and CE are brought LOW. The byte stored at that address in the EPROM appears at the DATA pins and is read. During all read operations, Vpp must be kept at +5V.

A PROGRAM operation is more complicated. Vpp is raised to a high voltage level as specified by the manufacturer. An address is sent to the address pins while a byte to be programmed into that address location is sent to the data pins. CE and PGM are brought momentarily LOW. The usual practice is then to verify that the eight data bits have been correctly programmed before proceeding to program data into the next address location. In the verify operation, the address and Vpp are maintained in their previous states, while OE is brought LOW. The programmed data bits appear at the data pins and are read. If the bits are verified as being correctly programmed then programming proceeds to the next address.

During programming, only 0's can be programmed into selected bit locations. It is not possible to reverse the process by electrically changing a 0 bit to a 1 bit. Thus, initially, all bits in an unprogrammed EPROM must be at a logic 1, and that is generally the case with all EPROMs as they come from the manufacturer. If, for any reason, some of the bits are at logic 0 before programming, then the entire EPROM will have to be erased by exposure to UV radiation. An EPROM programmer, therefore, should be capable of verifying, before programming, that an EPROM has been fully erased. As implied, erasure is the process of converting







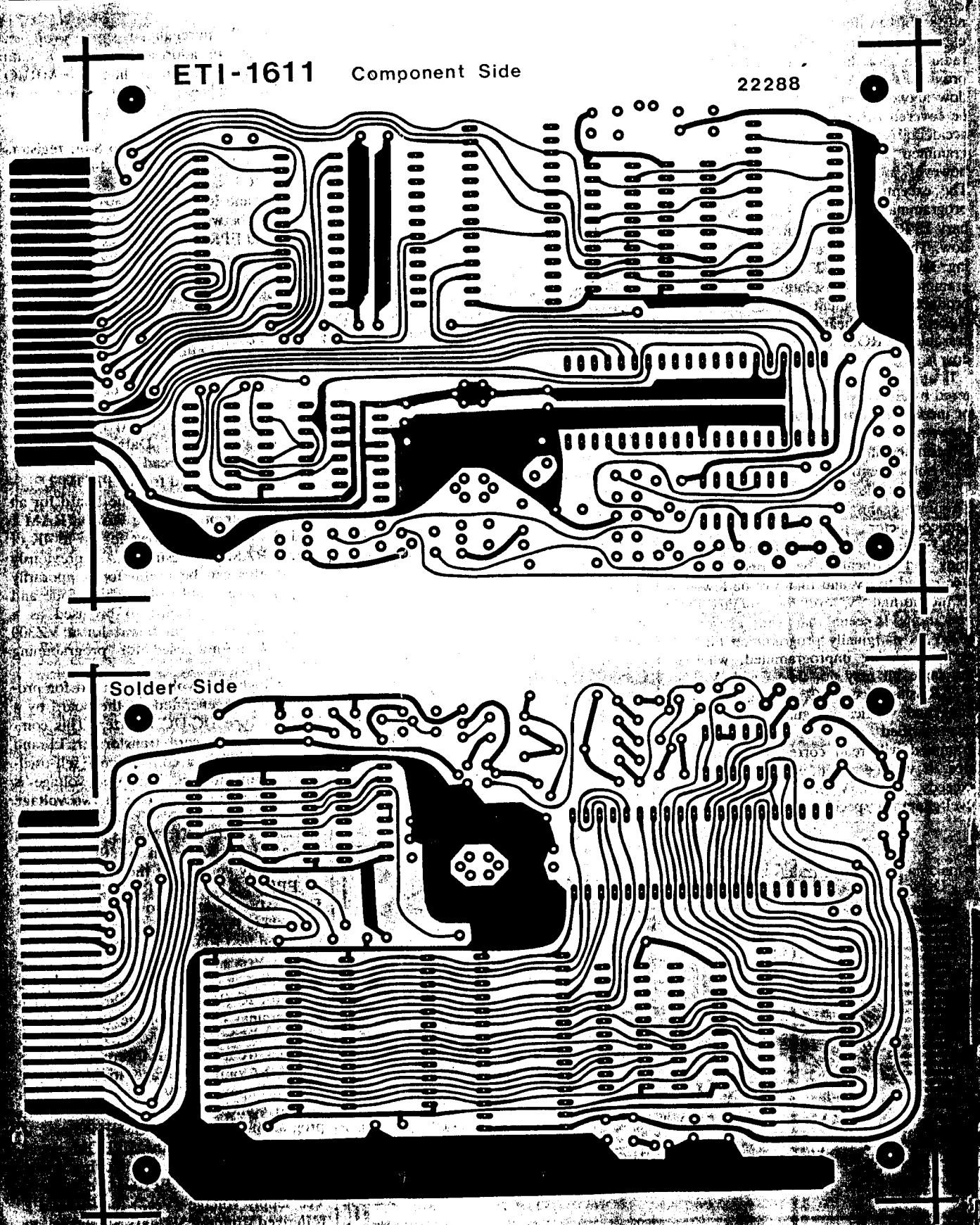
ETI-1611 — HOW IT WORKS

As it happens, the VZ300 has unused memory address space in the range B800H to FFFFH, which is available for external memory expansion, etc. Address decoder IC3 generates enable signals for the address latch, on-board RAM and EPROM 2 whenever the VZ300 executes a memory read or write instruction for an address in this range. When IC1 and IC2 have been enabled, the address is latched in their outputs and sent to the address inputs of EPROM 2 and RAM 1 and RAM 2.

IC4 provides the interface between the VZ300's microprocessor and EPROM 1 and the associated control circuitry. In use, PORT A is programmed by instructions from the VZ300 for bidirectional data transfer between the VZS300 and EPROM 1. PORT B is programmed as an output port, also by instructions from the VZ300, and generates all the necessary control signals for EPROM read and program operations in response to an appropriately coded instruction from the VZ300. During an EPROM read operation, data is read by an IN instruction addressed to PORT A. During an EPROM program instruction, data is sent to PORT A by an OUT instruction addressed to that port.

RAM 1 and RAM 2 share a common address range with EPROM 2. To avoid conflict, the decode circuitry allows only one of these to be enabled at any one time. Whether the EPROM or one of the RAMs is selected depends on a control bit sent to port B.

The total address space available for external memory in the case of the VZ300 is only a little over 16K. To program a 27256, which has 32K bytes capacity, it is necessary to generate the most significant address bit by some means other than via the VZ300's address bus. The problem is solved by using one of the port B lines for this purpose. As it happens, the PGM CONTROL Pin on the 2764 and 27128 becomes the most significant address pin on the 27256, so the same port B line is used to control both functions. The only complication is that slightly different software is needed for the 27256.


ETI-1611 Component Side
22288
Solder Side

all the bits in the EPROM to a logic 1 by exposure of the EPROM chip to UV radiation. For this purpose, EPROMs are provided with a transparent quartz window above the chip. This window should be covered by an opaque label to prevent accidental erasure in the case of a programmed EPROM. Not all EPROMs, however, are erasable (despite the name). The exception is known as a 'one-time-programmable EPROM', which is an ordinary EPROM but without the quartz window. This device is fully erased when leaving the factory and can only be programmed once. It is intended for use in production equipment and has the advantage of being cheaper to make than an erasable EPROM because a quartz window is not required.

It appears that most problems encountered by EPROM users arise due to faulty or incomplete programming. A marginally programmed bit, for example, may verify OK immediately after programming but may subsequently revert to the opposite logic level while the EPROM is in service. To guard against this possibility, National Semiconductor recommend, for their CMOS range of EPROMs, that programming and verification be carried out with Vcc raised to 6V and that Vcc be lowered to the normal 5V level for ordinary read operations. It seems that, with Vcc raised to 6V, a marginally programmed bit will verify as being unprogrammed, whereas the same bit may not do so with Vcc at 5V. Raising Vcc to 6V during programming and verification guarantees that all bits verified as being correctly programmed will read correctly during service. It will be noted, however, that 6V exceeds the 5.5V maximum operating level generally specified for EPROMs and manufacturers' specification should always be consulted if in doubt. In any case, the present EPROM programmer can be programmed to apply either 5V or 6V to Vcc during programming according to the user's selection.

An important consideration, also, when programming EPROMs, is the width of the PGM pulse which is applied during programming. Older EPROM types such as the 2708 were specified to be programmed with a single 50mS pulse per address location. With many later types, typified by the 27064 to 270256 series, a maximum pulse width as short as 10mS may be specified. Some manufacturers recommend an interactive programming algorithm to minimise the overall programming time. In an example of such an algorithm, a programming pulse of 0.5mS is applied and the programmed byte is verified. If it verifies as correctly programmed then programming proceeds to the next address. If not, then another 0.5mS pulse is applied with the current address and the process repeated until the byte verifies OK. If, after 20 pulses, a given address still does not verify OK then the EPROM is rejected as unprogrammable. With the present programmer it is a simple matter to adapt the software to any programming algorithm that may be recommended by an EPROM manufacturer.

Circuit Description

When plugged into the memory expansion slot of a VZ300 computer, this EPROM programmer has direct access to the address, data and control lines of the VZ300's internal Z80 microprocessor. Additionally, the memory expansion bus provides a 5V regulated supply voltage and a 12V unregulated supply voltage. There are 16 address lines and 8 data lines. The main control lines are MREQ (memory request), IORQ (input/output request), RD (read), WR (write) and O (clock).

The circuit comprises two 8-bit registers (IC1 and IC2) wired as a 14-bit address latch. IC3 and IC5b form an address decoder and IC4 provides a programmable interface between the VZ300's microprocessor and EPROM 1 which is the EPROM to be programmed. A 28-ZIF socket is provided on the board to enable the EPROM to be easily inserted and removed. Although more expensive than an ordinary IC socket, this type saves a lot of frustration and effort and is well worth the cost. There is space on the board for an optional, second, ZIF socket for EPROM 2. This is provided in case there is a need to copy from one EPROM to another as quickly as possible. Data can be programmed into, or read from EPROM 1 but can only be read from EPROM 2.

There is also space on the board for an optional pair of 2K static RAMs (RAM 1 and RAM 2). This allows for up to 4K of extra RAM if desired. As previously noted, this can be useful for temporarily storing large chunks of machine code and also allows the board to be used as a handy 4K expansion board for a VZ300 when it is not used for programming EPROMs.

The high Vpp voltage required for programming is generated on the board by a fly-back type DC-DC inverter. This comprises a ferrite core transformer T1 and transistor Q1 in a conventional self-oscillating configuration. The Vpp voltage is regulated by Q2, with one of two voltage levels (21V and 12.5V) selected under software control. Transistors Q3, Q4 and Q5 are used to switch off the Vcc and Vpp supply voltages at the respective pins of EPROM 1 and EPROM 2 before an EPROM is inserted into or removed from its socket. Power ON to the EPROMs is indicated by LED 1 lighting up.

The Vcc supply voltage (Vcc1) for EPROM 1 is obtained from a 5V voltage regulator IC (IC8) on the board. Although the nominal output voltage of this IC is 5V, a resistor R1 and diode D1 connected in series from the COM terminal of IC8 boost the output voltage to around 6V (plus or minus 0.25V). This higher than nominal voltage for Vcc is available when programming an EPROM (subject to recommendations of the EPROM manufacturer) and is reduced under software control to 5V in the EPROM read mode. Vcc supply voltage (Vcc2) for EPROM 2 is derived from the VZ300's 5V supply.