THE MEMORY LINGERS ON

SC/MP UVEPROM BURNER

John A. Stephenson shows the construction and software required for 'permanently' programming the 2708 and 2716 Proms.

As the owner of an early Mk14 SC/MP system, the author became aware of its severe limitations and decided to proceed to something more ambitious, using a Z80 and a home-designed system and monitor. To this end the Mk14 has been put to good use as a RAM tester (since one can get at about 1K of Mk14 memory addressing space very easily), and, lately, as a programmer for 2708 and 2758 or 2716 UVE PROMs.

The Mk14 could itself be improved by a better monitor. And the ability to easily and cheaply program such devices is very useful, since the fusible-link Proms used by the Mk14 are not exactly convenient or easy to obtain cheaply.

The circuit described here, together with software, will allow the owner of a Mk14 (or any SC/MP system) to use the existing monitor to load and check the data to be programmed. A short program then loads this data, in manageable 'chunks' into the Prom. The total memory requirement is 48 bytes for the program and 128 for the data store, although owners with more memory to spare can up this to 256 bytes and load twice as much at once.

There is no reason why the system should not run from any processor arrangement which has a 'Hold' or 'Wait' input, but the circuit shown is for a Mk14, and the necessary modifications are included to allow the required further address decoding.

Circuit operation

To avoid complicated switching arrangements two sockets are provided for the two types of Prom. The key part of the circuit is based around the 74121 monostable. This provides a 50 millisecond O/P pulse when triggered and serves two purposes. When programming 2716's it provides the TTL programming pulse required. When using 2708's it serves to slow down the programming cycle to avoid overheating the Prom.

Programming 2716's or 2758's is easy; set up an address plus data, apply +25v to pin 21, and a 50ms pulse to pin 18 and you have programmed that address. This circuit avoids using latches by having the processor supply address and data, and then, at the leading edge of the next write data strobe, it starts a 50ms monostable. The Q output of this is fed back to the processor NHOLD pin, and the 'write' cycle is thus extended by 50ms. The Q O/P is used to provide the programming pulse. The 74121 has an input, pin 5, which inhibits triggering when low. This is fed from the 'Flag 1' O/P of the Mk14, and the software sets flag 1 high only when a programming run is in progress, so preventing spurious data from entering the Prom during normal monitor operations.

IC 1 and 2 provide further address decoding which, with the modification shown to IC17 on the Mk14 P.C.B., restricts the monitor to the lowest ½K of memory, leaving 0400-07FF for the Prom.

The 2708 programmer is a little more involved. The two transistors form a 1ms monostable which is triggered by the leading edge of the 74121 O/P pulse, to provide a high-current drive +24v pulse to pin 18 of the 2708 socket. The manufacturers of 2708's do not approve of programming the chip any way other than by cycling through all 1024 locations at least 100 times. Anything else will cook the chip, I was told. This is not true if one allows the circuit to space out the programming pulses. The 50ms monostable does this admirably, and only slows the process down a little when one considers that the longest job is typing in the 1K of data!

The 2708 requires +25 and +12 volts as well as -5 volts, and the +12 volts is derived here from the +25v line to simplify matters. Do NOT leave out the diodes on the power lines, or you may blow up the Prom.

2708's require the CS pin, pin 20, to be at +12v during programming, but TTL levels are required for read cycles and normal Rom use. S1 is used as the simplest and cheapest way of providing this function and is set to the appropriate position prior to programming or checking the Prom.

Construction

The author's prototype was constructed on 3"x4" of Veroboard, wired up with single-core insulated wire on the copper side, tracks being cut where appropriate. The switches were small slide-switches, soldered to P.C. pins inserted through the board, and the interconnections were made direct on to the I.C. P.C. pads, using ribbon cable. The NHOLD connection was made to pin 6, which is linked to the "continue" I/P of the SC/MP but this does not affect operation. Only one track on the Mk14 needs to be cut; that to the monitor ROM's CE I/P's.

Software

The program shown is written for the SC/MP, and will run on the basic Mk14 (with only 256 bytes RAM). Modifications will be required should the reader wish to use the system with other processors.

To start, the program is loaded, by hand or cassette, beginning at OF20 (the whole program is relocatable). The START address of the RAM data block to be copied is placed in OF22, OF23, and the START address of the Prom position that's to be copied to is stored at OF24, OF25. The number of bytes to be copied (MINIMUM 128) is loaded (in HEX) at OF27. For 128 bytes this is 80. Finally, the number of 'loops' to be executed is placed at OF2B; if a 2716 or 2758 is to be programmed this should be 1, for a 2708 it should be at least 100, and the author recommends 128 loops (80 HEX).

Operation

To load and program a 2708, set the RAM START ADDRESS to OF50. The lowest PROM START ADDRESS is 0400, and extends for 1K to 07FF, in 128 BYTE BLOCKS (keep a careful written record of which blocks you have programmed!). The Prom occupies 0400-07FF. Select the block of data that you wish to program and, using the monitor, load the data to RAM starting at OF50. Check carefully the contents of the RAM that you have just loaded, as mistakes cannot be rectified afterwards! Switch the READ/PROGRAM switch, S1 to "PROGRAM".

Load the start address (OF26) and run the program. You now have time for a quick cup of coffee before the display returns. Change S1 to "READ", and, using the monitor, the Prom contents may be examined. You can then

load the next 128 bytes into the RAM positions, and change the start address for the Prom, e.g. to 0480. Set S1 to "PROGRAM" and run again from OF26, continuing the sequence to fill the Prom, in steps of 128 bytes at a time. To program 2716's (or 2758's) set the "Loop Count", CF27, to "01", and load data as described for 2708's. Run the program, and it will perform one complete load of 128 Bytes, and return to the monitor. Prom contents may be examined by 'reading' 0400-07FF, (S1 is not used).

For programming a 2716 (which is a 2K device) S2 must be set to choose the 'Upper' or 'Lower' half of the chip since pin 19 is the M.S.B. of the Address. For 2758's it is set to '0' (Ground, O.v dc).

Now, here is something interesting; would you believe that many 2758's are really 2716's in disguise? It seems that there are two 1K chips in a 2716 from some manufacturers and that the best of the two chips is selected, internally, at time of manufacture. The other 1K is still accessible though, as if the chip was a genuine 2716. I cannot guarantee this to be true of all the 2758's around but it certainly is for some — check yours and see!

John A Stephenson is a development engineer working in the field of background music systems. John's next project will be to develop a selective PROM copier. Why selective? — Well, it will enable you to copy parts of existing PROMs — ideal for bypassing burnt-in bugs — while keying in the correct code. Details coming soon.

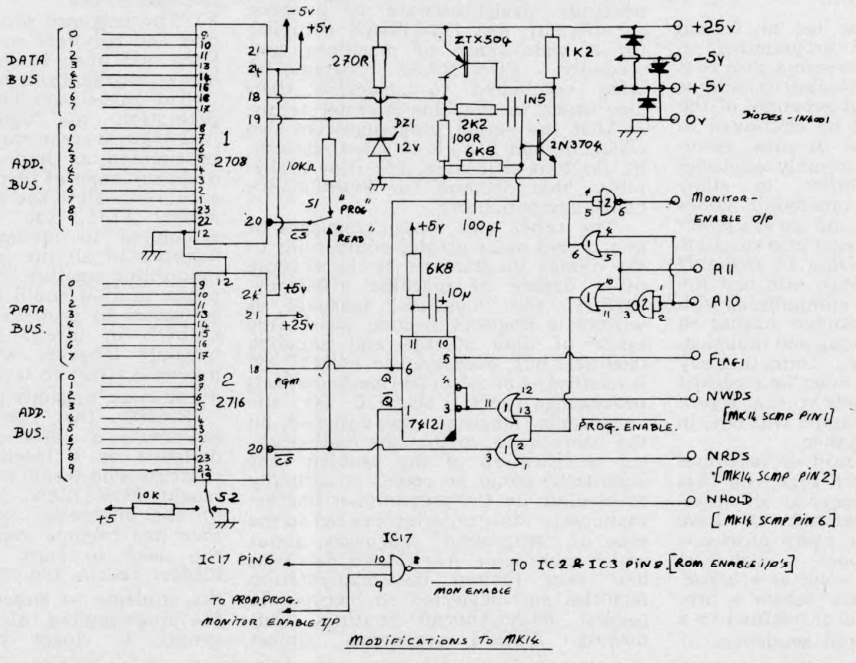
Program listing
SC/MP~MK14 PROM programmer

```
USES P1 TO POINT TO RAM STORE AND P2 TO PROM.
RELOCATABLE.
DATA TO TRANSFER STORED OF50-OFCF

          0F20  00       A COUNT
          0F21  00       B COUNT
          0F22  0F50     RAM START ADD.(STORAGE)
          0F24  0400     PROM START ADD.

PROGRAM RUNS FROM 0F26

          0F26  C4       LDI            LOAD NO.OF
          0F27  80       LOOP TOTAL     LOOPS
"LOOP"    0F28  C8 F8    ST(B COUNT)    STORE AT B.
          0F2A  C4       LDI            LOAD NO. OF
          0F2B  80       BYTE TOTAL     BYTES
          0F2C  C8 F3    ST (A COUNT)   STORE AT A
          0F2E  C4 02    LDI 02         LOAD 2
          0F30  07       CAS            SET FLAG 1
          0F31  C0 F0    LD RAM Hi      LOAD
          0F33  35       XPAH 1         POINTER
          0F34  C0 EE    LD RAM Lo      1 WITH
          0F36  31       XPAL 1         RAM POSITION
          0F37  C0 EC    LD PROM Hi     LOAD
          0F39  36       XPAH2          POINTER
          0F3A  C0 EA    LD PROM Lo     2 WITH
          0F3C  32       XPAL2          PROM POSITION
"WRITE"   0F3D  C5 01    LD @ (RAM)     LOAD DATA
          0F3F  CE 01    ST @ (PROM)    WRITE TO PROM
          0F41  B8 DE    DLD (A COUNT)  DECREMENT & GO TO
          0F43  9C F8    JNZ "WRITE"    "WRITE" ON ZERO
                                        BYTE COUNT
          0F45  B8 DB    DLD (B COUNT)  DECREMENT & GO TO
          0F47  9C E1    JNZ "LOOP"     "LOOP" ON ZERO LOOP
                                        COUNT
          0F49  C4 00    LDI 0          LOAD 0 &
          0F4B  07       CAS            CLEAR FLAG 1
          0F4C  3F       XPPC3          RETURN TO MONITOR
          0F50           START OF DATA STORE
          0FCF           END OF DATA STORE (128 BYTES)
```



MODIFICATIONS TO MK14