THE MEMORY LINGERS ON ~PART 2

Last September PCW published a design by John Stephenson for a simple SC/MP based PROM programmer. He has now developed an enhancement which enables the copying of selected sections of existing PROMs, thus, for example, allowing you to bypass burnt-in bugs while keying in the correct code.

A SELECTIVE PROM COPIER

Copier hardware

Two additional sockets are required, connected pin-for-pin to their opposite numbers SK1 and SK2 on the Programmer, with the exception of Pins 18 and 20 (PGM and $\overline{CS}$). Since these two sockets are only used as ROM, Pin 18 of each is grounded. Both Pin 20s are taken to one side of S3, and the other side of this is connected to Pin 20 on SK2. The lead which ran from SK2 Pin 20 to IC1 Pin 3 is disconnected from SK2 and is run to the wiper of S3.

S3 now forms a Read Master/Read Copy switch, necessary since both Master and Copy occupy the same memory position, 0400 — 07FF.

Programmer

To provide faster copying for 2708s the Monostable period may be reduced to 10ms (nominal) by means of S4, selecting a 2MFD capacitor. This must not be used when programming 2716s, and, with 2708s, preferably not for blocks of data less than 16 bytes long, as data corruption may then occur.

Software

This program can be used for both programming and copying. When loaded with the start and end addresses of the data to be copied from the Master EPROM, (at 0F20, 21, 22, 23) and the start address to which the data is to begin in the copy EPROM, the program sequences through the addresses, reading data from the Master and programming it into the copy. Auto-indexed-addressing is used to allow easy sequencing, and P1 is used to point to the Master source address, with P2 as the copy destination address. The program can be used to insert blocks of data into a copy, corrections for example, by copying up to the bug, programming the fresh data, which has been loaded into RAM, from 0F70 to 0FEF, and then continuing with the copying from the end of the bug.

Operation

Load up with Master and Copy, and apply power (don't be tempted to insert chips with power on!).

Load start address of Master block
to 0F20, 21
Load end address of Master block
to 0F22, 23
Load start address of copy to 0F24, 25
(Blocks may be any size up to 1K)
Load 0F37 with the number of cycle repeats required, 80H for a 2708 (128 repeats) and 01H for a 2716.
Remember that S2 must be set to select the upper or lower 1K of a 2716 to be programmed.

Set S1 to "Program", S3 to "Read Master", and S4 as appropriate to the block length or EPROM type, and run the program from "START", 0F29.

Notes and comments

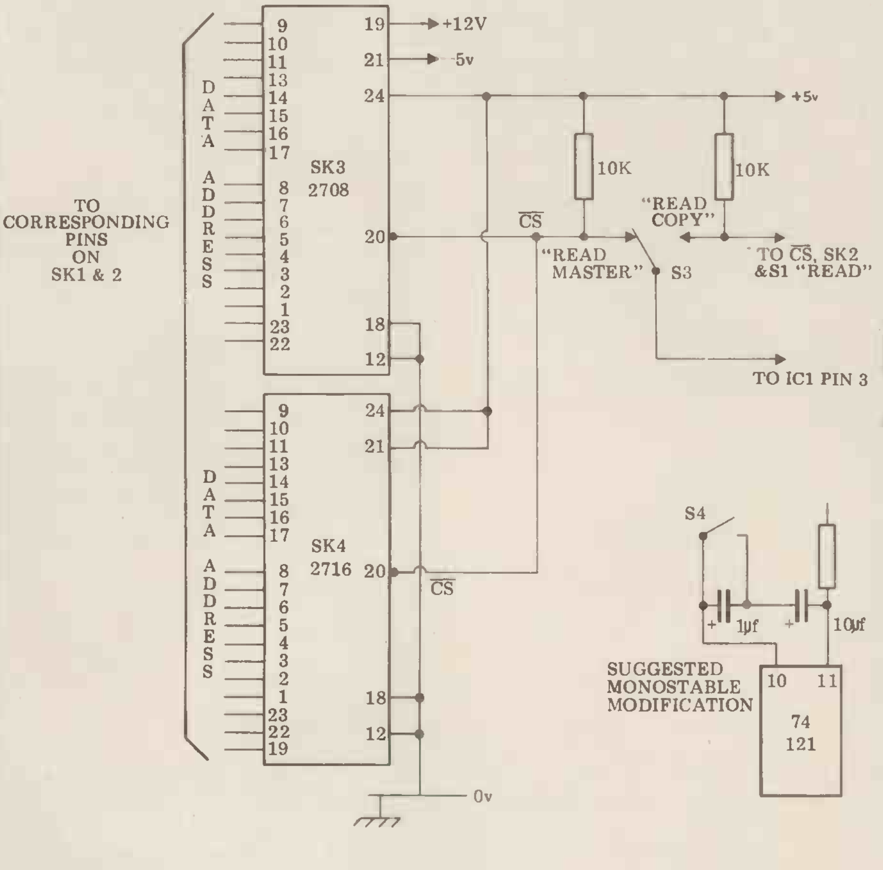
The prototype copies ¼K in approximately 7½ minutes. This is not up to the theoretical limit of 30 seconds, but who cares? People who *do* care can reduce the monostable time still further but only by adding complication and at the risk of destroying the PROM!

The multiple switches could be replaced by a multi-pole switch if required, as S1 and S3 can be ganged.

Further, S3 could be implemented with a handful of gates operated by "flag 0" on the SC/MP. It would then be very simple to write a verification program to compare copy and master for errors caused by duff PROMs, etc.

If you can't afford Zero-insertion-force sockets (like me!) shop around for 24 pin gold-plated ones and ease out the contacts with a dressmaking pin. This avoids lots of bent I.C. legs!

Program listing, p102


TO CORRESPONDING PINS ON SK1 & 2
DATA
ADDRESS
SK3 2708
SK4 2716
9
10
11
13
14
15
16
17
8
7
6
5
4
3
2
1
23
22
19
24
21
20
18
12
+12V
-5v
+5v
10K
10K
$\overline{CS}$
"READ COPY"
"READ MASTER"
S3
TO $\overline{CS}$, SK2 &S1 "READ"
TO IC1 PIN 3
0v
S4
+
1µf
+
10µf
SUGGESTED MONOSTABLE MODIFICATION
10
11
74 121