450

OSI

OHIO SCIENTIFIC INSTRUMENTS

11679 HAYDEN STREET, HIRAM, OHIO 44234

THEORY OF OPERATION:

Introduction:

The OSI Model 450 is a 400 series PROM board. It used AMI S6834 PROMS (512 x 8) to provide up to 8K bytes per board. An important feature is the on-board PROM programmer which, together with the driver routines in this manual, allows the user to program his own PROMs.

The board is comprised of four subsystems: addressing and data buffering, PROM operation (read only), PROM programmer, and PIA port.

Addressing and Data Buffering:

The PROMs and PROM programmer are two separate systems occupying different locations in memory space, hence each has its own address decoder. (See Diagram 1). A programmer enable signal $\overline{BEP}$ is derived by NANDing A_8-A_{15}. These address lines are present in both inverted and non-inverted forms in the address jumper area in the lower right corner of the board. Normally this should be jumpered for FØXX to be compatible with OSI programming software.

A memory enable signal $\overline{BEM}$ derives from A_{12}-A_{15}. The exact jumper implementation depends on whether the board will contain 4K or 8K of PROM. A 4K board requires decoding all four address lines, which is done by jumpering to the inputs of a 4-input NAND (7420, 1C-K). Jumpering for an even-numbered address (e.g. 6XXX, A_{12}=0) will result in the eight low order PROM locations on the board being used, while an odd-numbered address (e.g., 7XXX, A_{12}-1) will cause the eight high-order locations to be used. For 8K implementation A_{12} is not jumpered to the NAND, rather the unused input is pulled up via R11 (See parts placement diagram). Thus, $\overline{BEM}$ will become active for two adjacent 4K memory blocks (e.g., 6XXX and 7XXX) since A_{12} is not decoded here. In this case, A_9A_{12} are fully decoded to access any one of the 16 PROMS on-board.

$\overline{BEM}$ and $\overline{BEP}$ must be jumpered so that they do not duplicate an address already in use in the system. In particular, they cannot both be set for the same address. If $\overline{BEP}$ is set for FØXX, then $\overline{BEM}$ cannot be set for FXXX.

Decoding of A_9-A_{12} is done by a 4-to-16 line decoder (74154, 1C-I). It outputs a low-true condition on one of sixteen lines selected by A_9-A_{12}. Each line goes to the chip select input (low true) of a different PROM. Inputs G1 and G2 are chip enables (low to enable) which are internally ANDed. If either one goes high all sixteen outputs go high (i.e., A_9-A_{12} are ignored). Consequently the PROMs only enable when the board enables ($\overline{BEM}$ low) and R/W is high (CPU is reading) and Ø2•VMA is high. Notice that Ø2•VMA clocks any active output line, in turn clocking the selected PROM.

Low order addresses $A_Ø$A-$_8$ are fed to all PROMs via latches (7475, 1C-F, G, H). Normally, when the board is not in program mode, memory enable (ME) is high and the latches are inoperative. The latch outputs follow the inputs, hence they appear transparent to the PROMs. Latching occurs only in the programming mode, to be discussed below.

A master board enable (BE), derived by NANDing $\overline{BEP}$ and $\overline{BEM}$, activates when even either section is addressed. It is used to provide certain control signals. Each time the board is accessed a "wait" signal is generated on B1, stretching Ø2. This is necessary for the relatively slow S6834's. Also, BE gates system R/W (B40) onto the board via a NAND gate (7420, 1C-KO. R/W determines data direction on the 8T26 data buffers (1C-D, E), and also signals data direction to the CPU via DD (B4). Notice that if the board is not enabled (BE low), the 8T26s are always in the "listen" mode, that is, receiving data from the bus.

PROM Operation (read only)

The switch in the upper left corner places the board in normal ("N" position) or program ("P" position) mode. Normal mode places the 8K of PROM on line and disables the programmer. (See Diagram 4). ME (from 1C-J, pin 6) is forced high so $A_{\emptyset}$-A_8 go directly to the PROMs, i.e., they are not latched. This is the usual board configuration.

For a system Ø2 of 750 KHz or above, wait diode D4 must be installed. Also the Model 400 CPU must be equipped with the two-speed clock option detailed in APP. Note #4A.

PROM Programmer

The Model 450 PROM Programmer and software allow the user to program and verify an S6834. After erasure, all bits of the PROM are low. Programming is accomplished by setting up an address and data word, pulling the R/W line low, and applying a series of -50V pulses to the Vprog. input. This voltage must be supplied by an off-board power supply (required current is 50ma) which is user-supplied.

For programming, the timing relationships are as follows. The address, data, and $\overline{CS}$ must be set up at least 10 μsec before the first programming pulse and R/W must go low at least 10 μsec before that. The -50V must then be pulsed a specified number of times of specified duration. The amount of programming energy supplies by the -50V pulses must be

$$N \times t_{pw} \geq 60 \text{ msec.}$$

to insure program retention, where t_{pw}=pulse width and N=number of pulses. The maximum pulse width is 5ms, so at least 12 pulses must be applied. Consequently, R/W must hold for a minimum of 66.03 mses, and addresses, data, and $\overline{CS}$ must hold for at least 66.02 msec, including

a 10 μsec hold time after the end of the last pulse. Some form of latching is required.

Address lines $A_{\emptyset}$-A_8 latch whenever ME line is brought low (Diagrams 2 and 4). With the board in program mode, (switch to the right), thus occurs when the CPU attempts to write into one of the 16 PROMs. $\overline{BEM}$ activates allowing Ø2•VMA goes high, and will hold that address when $\overline{BEM}$ goes high (board deselects) on the next instruction. The address ALØ-AL8 is applied to the PROM to be programmed, and will not change through subsequent programming steps until similarly accessed.

There is no data bus conflict between the 16 PROMs and the programmer because R/W low inhibits the 74154. Thus no other PROM is enabled. Since $\overline{BEM}$ does not enable the 16 PROMs in program mode, it is not possible to duplicate directly a PROM in one of these locations. The contents of the PROM must first be transferred into RAM.

Data is applied to the PROM to be programmed via an MC6820 PIA (Diagram 3), peripheral section "A",. Data stored in the output register will remain "latched" until removed by software. The "B" section of the PIA controls $\overline{CS}$ and R/W (PB0 & PB1, respectively).

Pulsing of the -50V supply is done by loading a logical "1" into PB2, then loading a logical "Ø". The one-shot (74123, IC-A0 triggers on the falling edge applying a high-level pulse to the base of Q1 through R3 (100ohm). The width of the pulse is determined by R6 (18K) and C1 (1uf), with these values 5 msec.) This turns on Q2, applying a -50V pulse to the Vprog input.

Timing requirements between R/W, address, data and $\overline{CS}$ lines must be met by software timing. Since the times involved (10 usec) are on the order of 10 clock cycles, this can be done with timing loops. This

method may also be used to determine when a Vprog. pulse has completed (5 msec delay), however, the $\overline{Q}$ output of the one-shot can be jumpered to PB4. A polling routine which reads the location and delays execution of the program until the line returns high can then be used.

A programmer driver routine is listed at the end of the manual. When operating, it will display the prompter ")", and expects an input of one of the following four commands (XXX is in hexadecimal form):

(1) PXXX - Programs 512_{10} contiguous bytes of RAM into PROM, starting at address XXXX.

(2) VXXXX - Verify. Does a byte-by-byte comparison of PROM with RAM starting at location XXXX for 512 bytes. Normally this should be the same address as used in the program command. The routine outputs a "@" for each unsuccessful comparison.

(3) RXXXX - Reads the contents of the PROM and displays on teletype or CRT. XXXX is a dummy address and can be any value.

(4) MXXXX - Move. Used to duplicate a PROM only. The routine will move the contents of a PROM (in the programming socket) into RAM starting at address XXXX.

The programming procedure section contains specific information on using the routines.

The driver routine as listed is for use with the OSI 65A Monitor and serial terminal. For use with OSI 65V, the user must modify the program. Load OSI's CRT simulator at ØE53. Then, change the jump address at location 1060 (assembler line 570) from FE0B to ØE53 "INCH". (location 1063, assembler line 580), inputs a character from a keyboard and echoes it. To simulate this under 65V load the following program starting at location 11DA:

11DA 20 ED FE 20 53 ØE 60

Then, change the JMP address at 1063 from FE00 to 11DA.

PIA PORT

The 450 board contains an OSI standard PIA port (Diagram 3). The output lines run to two Molex connectors at the top left of the board. +5V power and GND are also brought out at several places on the connectors.

R9-C2 form an automatic power-on reset circuit. In lieu of this, B21 may be dedicated as a system reset. An off-board power-on reset circuit must then be implemented, and jumper J2 installed.

If interrupt operation is desired, install J3 to system $\overline{IRQ}$. The user must supply software servicing routines.

Parts List

PROM only

1 Model 450 Board
2 8T26
1 7400
2 7404
1 7420
1 7430
3 7475
1 74154
2 1K 1/4 watt (R10, R11)
2 IN914 (D3, D4)
28 .luf bypass caps
1 25uf 24V electrolytic (C3)
1 to 16 S6834 PROMs

Programmer

1 74123
1 MC6820 PIA
1 100ohm (R3)
1 220 ohm (R5)
1 1K (R7)
4 4.7K (R1, R2, R8, R9)
1 18K (R6)
1 22K (R4)
1 1.0 uf mylar (C1)
1 25uf 25V electrolytic (C2)
2 IN4001 diode (D1, D2)
2 PNP transistors 2N398B or equiv. (Q1, Q2) CGE 71
1 DPDT slide switch
1 24-pin socket (ZIF preffered)
1 4 pin Molex (KK-156) Power connections

Optional

PROM Sockets as Desired

4 Molex KK-156 Connectors (B1-B48)
2 Molex KK-156 Connectors (F1-F24) (PIA PORT only)

CONSTRUCTION:

Step 1: Carefully inspect for foil shorts and breaks. Backlight the board and, using a magnifying glass, inspect the back of the board. Pay particular attention to the foil runs between pins on the PROMs. If there are any shorts they will probably occur here. Cut shorts with a sharp razor or Exacto knife.

Step 2: Install as many PROM sockets as desired. Install a socket at the programming location [marked (Prog) on overlay]. Install all TTL parts, 8T26s, and PIA. Notice that the 7475s are installed with pin 1 toward the bottom of the board.

Step 3: Install resistors R1-R11. Refer to parts list for values, and overlay for locations. Similarly install capacitors C1-C3. (Note: R9-C2 provide automatic power-on reset of the PIA. B21 may be dedicated as a system reset line, in which case, omit R9-C2 and install J2. External reset must then be provided to B21.) Install the switch.

Step 4: Install jumper J4. J1 and J3 are optional.

Step 5: Install D1-D4, observing the correct polarity. Install Q1, Q2.

Step 6: Install all bypass capacitors (marked "C" on the overlay).

Step 7: (Address Jumpering) The board must normally be jumpered to enable the PROMs on AXXX, the programmer on FØXX. Jumper as shown on the overlay (lower right corner).

Step 8: Install four Molex connectors along the right edge of the board (B1-B48). Install the power connector (4 pin Molex) at P1 (top left corner of the board). If using the PIA port, install two male Molex connectors at F1-F12.

Step 9: Verify that the memory space of the Model 450 is unused by any other board in the system. Install the board on backplane or Model 498 Card Extender to backplane. Verify programmer operation by programming a PROM according to the procedure outlines at the end of this section. CAUTION: NEVER install or remove a PROM from the programming socket with the switch in "Program" position. Doing so may result in destruction of the PROM. The PROM may be removed with the switch in "normal" position.

Step 10: Install the PROMs programmed in Step 9. The location marked "S6834 (Typ. for 16)" on the overlay is the lowest-order PROM. Next fill the bottom position in this column, then the top position next column, etc. CAUTION: PROMs should never be removed from these sockets with the system power on!

Programming Procedure:

(1) With system power turned off, connect Model 498 Card Extender to backplane. Connect Model 450 PROM Board to Extender.

(2) Turn system power on.

(3) Load in programmer driver routines. If using a video-based system, modify as described in theory of operation. On either system, load locations 0129, 012A, 012B, 012C with 00. Load 012D with 28.

(4) To duplicate a PROM, disconnect -50V supply and place mode switch to the right (program). Execute the driver routine (starting address is 1000 hex). The routine will output the prompter ")". Enter MXXXX to move the contents of PROM into RAM starting at address XXXX. Be careful not to over-write the driver routine! Return switch to normal position, and remove the PROM.

(5) Make sure the Mode switch is in the normal position. Connect -50V supply via the 4-pin Molex connector at the top left corner of the board.

(6) With mode switch in normal position, insert a blank PROM into the programming socket. Set mode switch to "Program" position. Enter PXXX, where XXXX is the starting address of the block of RAM to be programmed into PROM. For Ø2 of 1 MHz, programming requires two minutes. (Note: the driver routines can be used with clock speeds up to 2 MHz.) When programming is complete, the driver will output the prompter. Verify correct programming by entering VXXXX, where XXXX is the same starting address as

used to program the PROM. If all is well, the prompter will return. Otherwise a "@" will output, once for each faulty byte. If this occurs, return to the beginning of Step 6 and reprogram. (This is recommended even if all bytes check out. Programming twice helps ensure data retention.) Return mode switch to normal position, and remove the PROM.

(7) If another PROM is to be programmed, return to Step 4 (to duplicate) or Step 5.

All S6834 PROMs are shipped from the factory with all bits high, and so must be erased prior to programming. A GE germicidal lamp is a suitable source. It can be used in any fluorescent fixture. Place the quartz window of the PROM 4 inches from the lamp for 15 minutes. CAUTION: The UV light produced by this lamp can damage eyesight! NEVER look directly into the lamp!

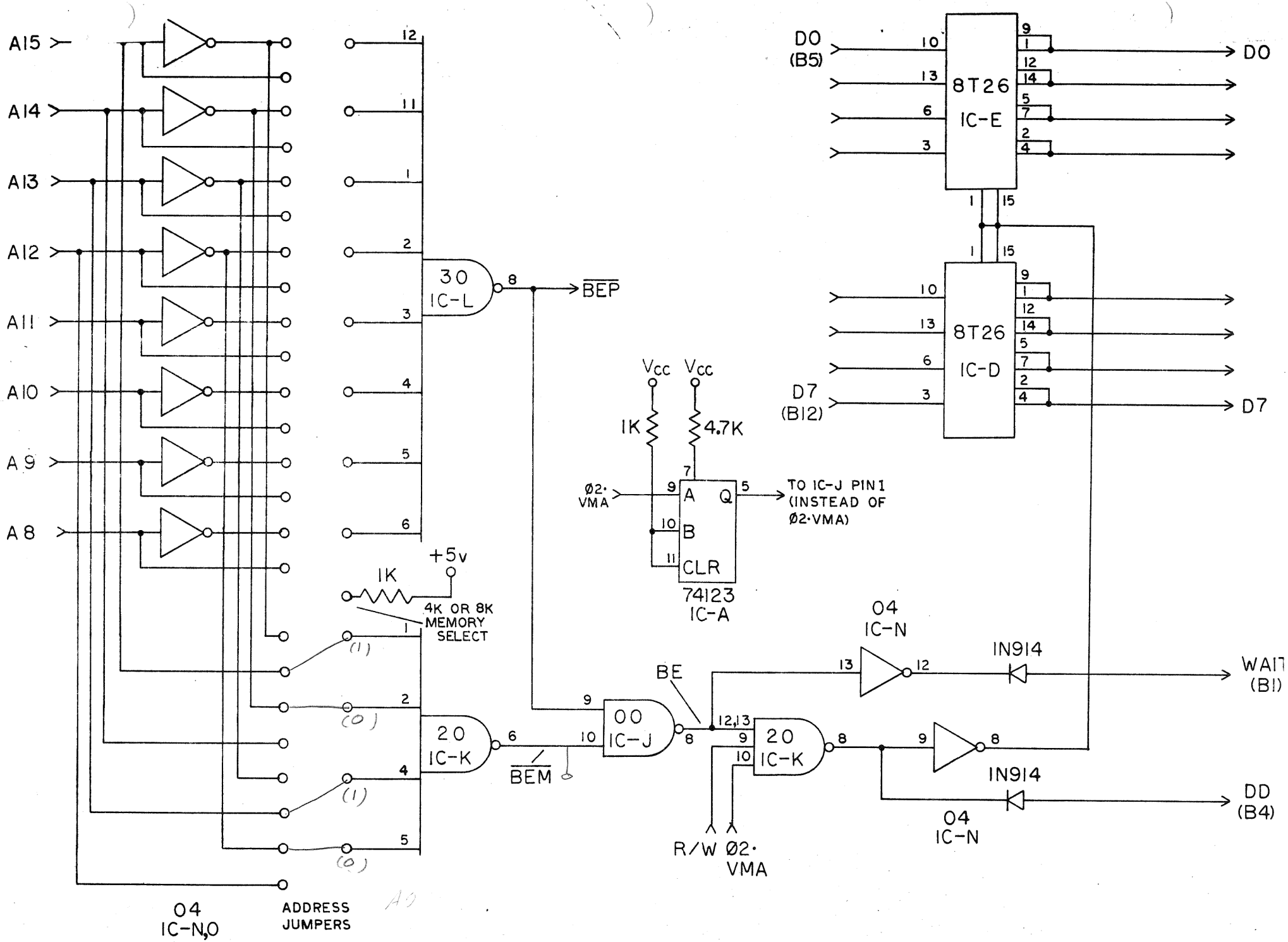


DIAGRAM 1- ADDRESS DECODING

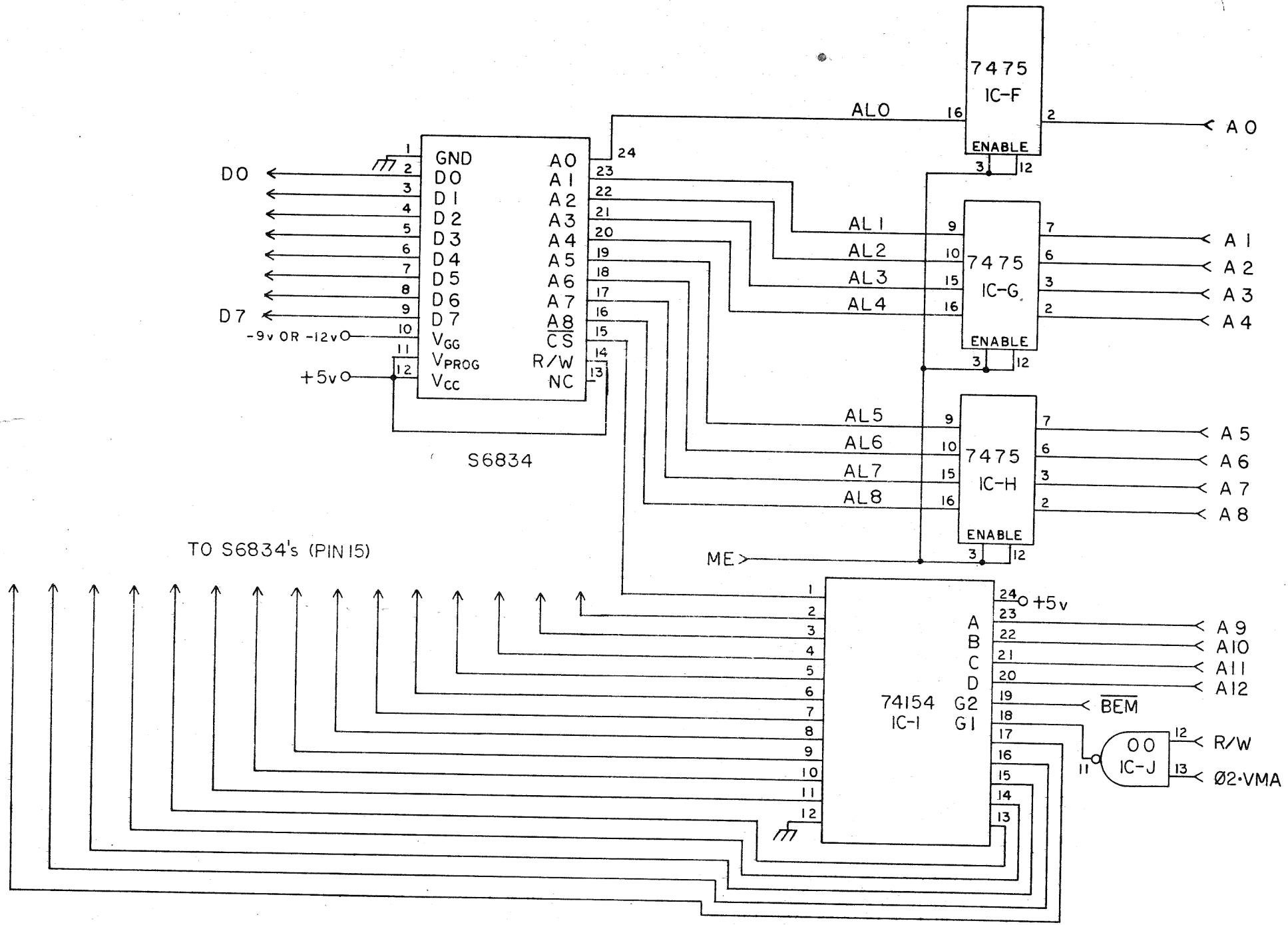


DIAGRAM 2- PROM IMPLEMENTATION

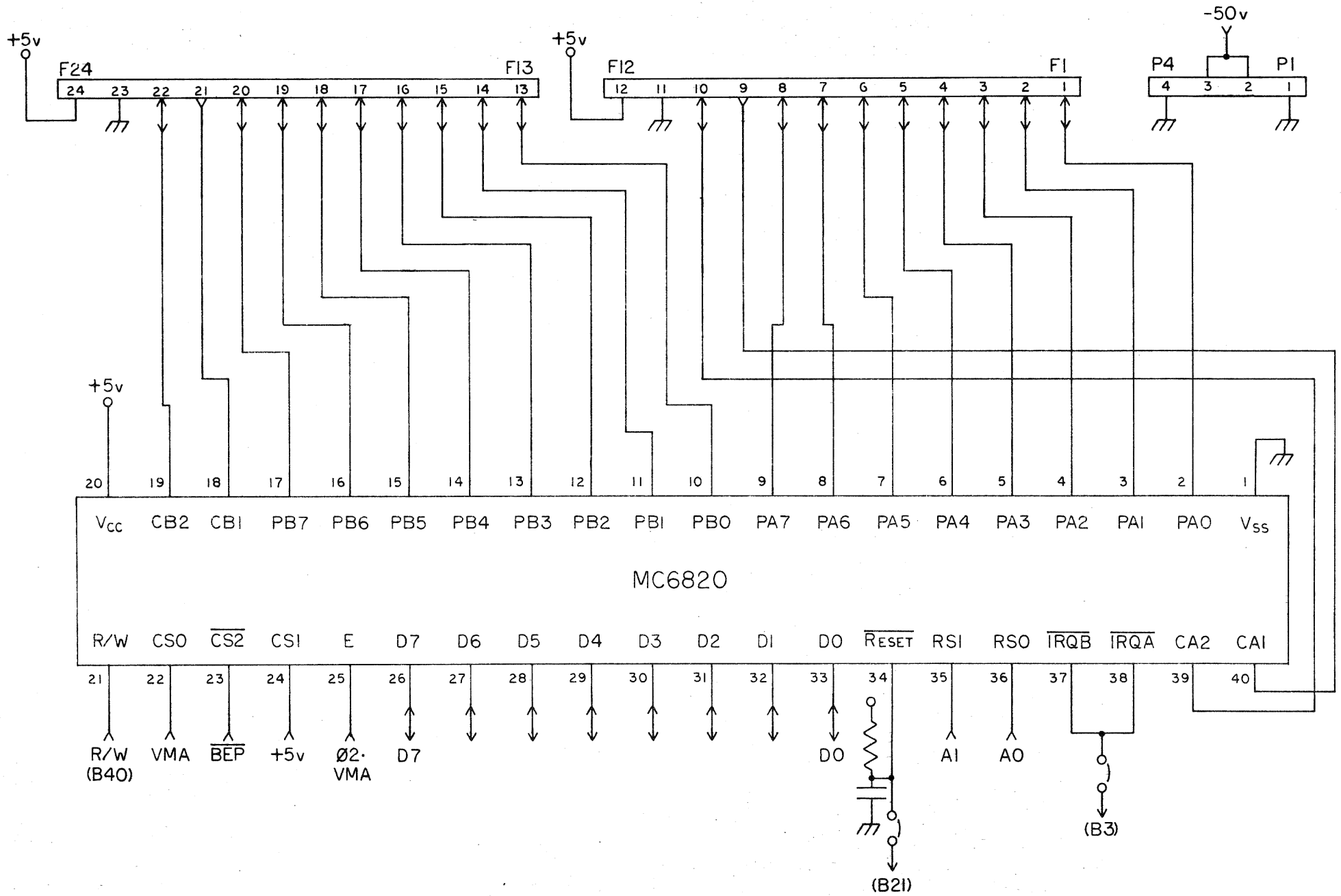


DIAGRAM 3- PIA

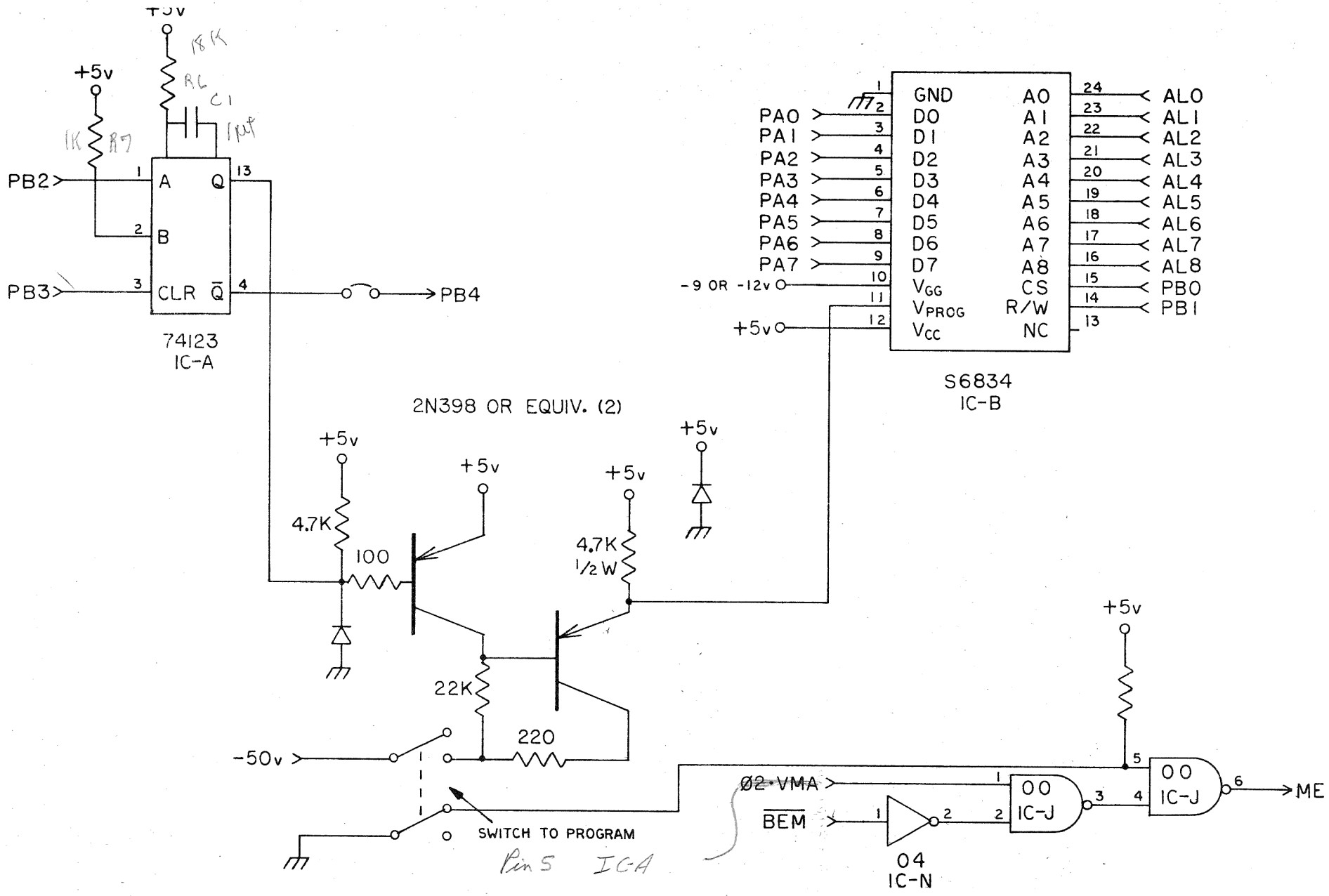
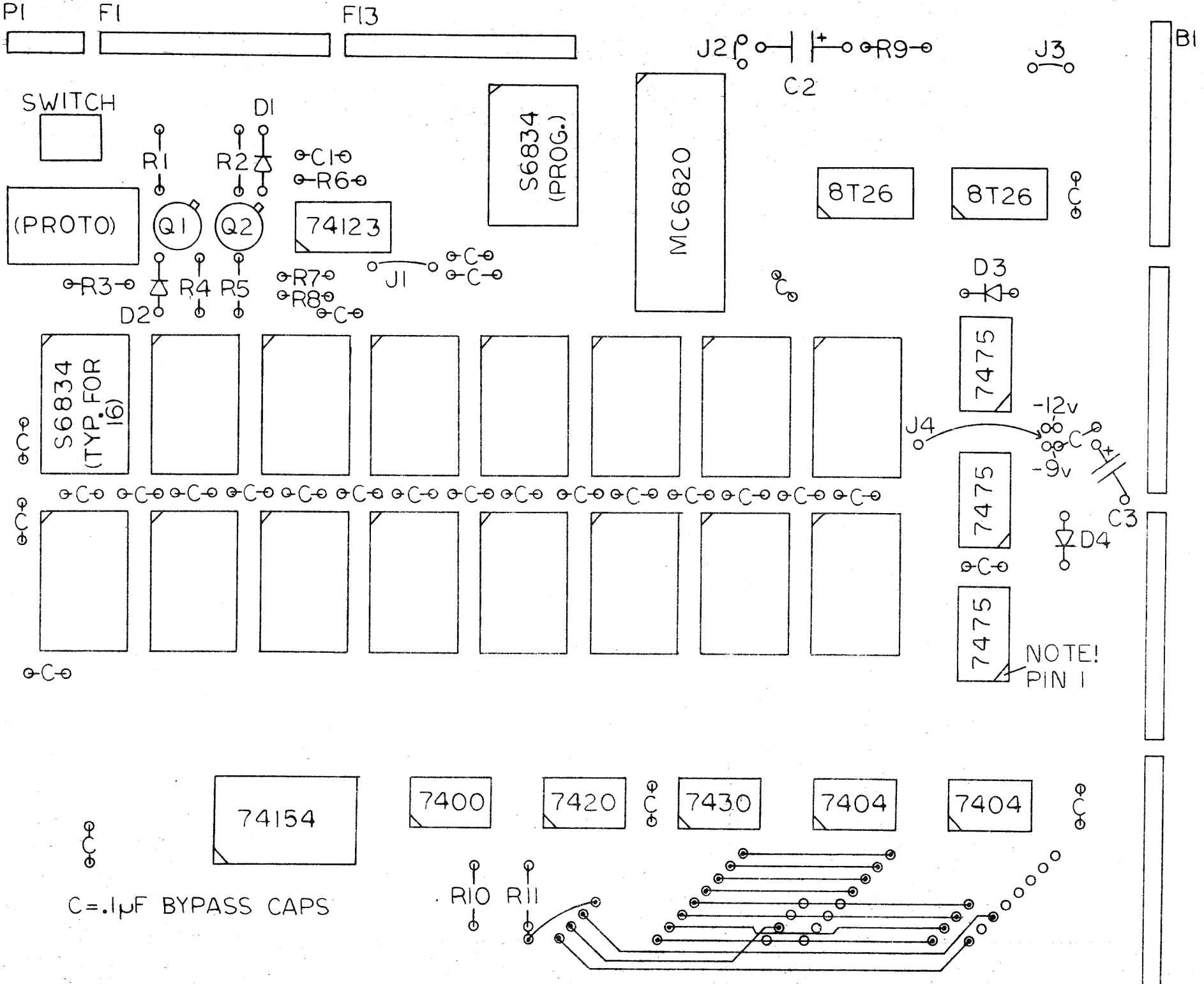


DIAGRAM 4- PROM PROGRAMMER


P1
F1
F13
J2
C2
R9
J3
B1
SWITCH
D1
R1
R2
C1
R6
S6834
(PROG.)
MC6820
8T26
8T26
(PROTO)
Q1
Q2
74123
R3
R4
R5
R7
R8
J1
D2
D3
7475
S6834
(TYP. FOR
16)
J4
-12v
-9v
C3
D4
7475
7475
NOTE!
PIN 1
74154
7400
7420
7430
7404
7404
R10
R11
C=.1μF BYPASS CAPS

```
ASSEM

  10 0000           ; OSI 450 PROM PROGRAMMER
  20 0000           ; DRIVER ROUTINES.  (CT>2MHZ)
  30 0000           WHRL =$0
  40 0000           WHRH=$1
  50 0000           SAML=$2
  60 0000           SAMH=$3
  70 0000           CNT=$4
  80 0000           MODE=$5  0=>READ, ELSE COMPARE
  90 0000           FLAG=$6  0=>READ, ELSE MOVE
 100 0000           FLIPF=$7
 110 0000           WHERE=$10
 120 1000           *=$1000
 130 1000           ;
 140 1000 A228      PCNTR LDX #$28  CONTROL LOOP
 150 1002 9A              TXS
 160 1003 201D11          JSR CRLF
 170 1006 A929            LDA #'
 180 1008 206010          JSR OUTCH
 190 100B A910            LDA #WHERE
 200 100D 8500            STA WHRL
 210 100F A000            LDY #0
 220 1011 8401            STY WHRH
 230 1013 AD7F11          LDA STRT
 240 1016 8502            STA SAML
 250 1018 AD8011          LDA STRT+1
 260 101B 8503            STA SAMH
 270 101D AE8111          LDX LEN
 280 1020 B102      MOVL  LDA (SAML),Y
 290 1022 9100            STA (WHRL),Y
 300 1024 C8              INY
 310 1025 CA              DEX
 320 1026 D0F8            BNE MOVL
 330 1028 206310          JSR INCH
 340 102B 48              PHA
 350 102C 206610          JSR BUILD
 360 102F 68              PLA
 370 1030 C956            CMP #'V
 380 1032 D006            BNE NX1
 390 1034 209610          JSR CMPR
 400 1037 4C0010          JMP PCNTR
 410 103A C950      NX1   CMP #'P
 420 103C D006            BNE NX2
 430 103E 203411          JSR PRGM
 440 1041 4C0010          JMP PCNTR
 450 1044 C94D      NX2   CMP #'M
 460 1046 D006            BNE NX3
 470 1048 20A710          JSR MOVE
 480 104B 4C0010          JMP PCNTR
 490 104E C952      NX3   CMP #'R
 500 1050 D006            BNE ILLEG
 510 1052 209A10          JSR READ
 520 1055 4C0010          JMP PCNTR
 530 1058 A93F      ILLEG LDA #'?
 540 105A 206010          JSR OUTCH
 550 105D 4C0010          JMP PCNTR
 560 1060           ;
 570 1060 4C0BFE    OUTCH JMP $FE0B
 580 1063 4C00FE    INCH  JMP $FE00
 590 1066
```

```
 600 1066 A201     BUILD LDX #1  READ ADDRESS
 610 1068 206C10         JSR BYTE
 620 106B CA             DEX
 630 106C 207D10   BYTE  JSR IND
 640 106F 0A             ASL A
 650 1070 0A             ASL A
 660 1071 0A             ASL A
 670 1072 0A             ASL A
 680 1073 9500           STA WHRL,X
 690 1075 207D10         JSR IND
 700 1078 1500           ORA WHRL,X
 710 107A 9500           STA WHRL,X
 720 107C 60             RTS
 730 107D          ;
 740 107D 206310   IND    JSR INCH
 750 1080 C930           CMP #'0
 760 1082 30D4           BMI ILLEG
 770 1084 C93A           CMP #':
 780 1086 300B           BMI OK1
 790 1088 C941           CMP #'A
 800 108A 30CC           BMI ILLEG
 810 108C C947           CMP #'G
 820 108E 10C8           BPL ILLEG
 830 1090 18             CLC
 840 1091 E906           SBC #6
 850 1093 290F     OK1   AND #$F
 860 1095 60             RTS
 870 1096          ;
 880 1096 A9FF     CMPR  LDA #$FF  COMPARE DATA
 890 1098 3013           BMI INTO
 900 109A A200     READ  LDX #0  PRINT DATA
 910 109C A500           LDA WHRL
 920 109E 8502           STA SAML
 930 10A0 A501           LDA WHRH
 940 10A2 8503           STA SAMH
 950 10A4 8A             TXA
 960 10A5 F004           BEQ INT
 970 10A7 A2FF     MOVE  LDX #$FF  MOVE DATA
 980 10A9 A900           LDA #0
 990 10AB 8606     INT   STX FLAG
1000 10AD 8505     INTO  STA MODE
1010 10AF A000           LDY #0  INIT. PIA
1020 10B1 8C01F0         STY $F001
1030 10B4 8C00F0         STY $F000
1040 10B7 A204           LDX #4
1050 10B9 8E01F0         STX $F001
1060 10BC 8C03F0         STY $F003
1070 10BF 88             DEY
1080 10C0 8C02F0         STY $F002
1090 10C3 8E03F0         STX $F003
1100 10C6 C8             INY
1110 10C7 A505           LDA MODE
1120 10C9 D004           BNE IN0
1130 10CB A506           LDA FLAG
1140 10CD F006           BEQ OTHR
1150 10CF 8402     IN0   STY SAML
1160 10D1 A9A0           LDA #$A0
1170 10D3 8503           STA SAMH
1180 10D5 A908     OTHR  LDA #8
1190 10D7 8504           STA CNT
1200 10D9 201D11         JSR CRLF
1210 10DC 204400   ADR   JSR OBTAN-BLAST+WHERE
1220 10DF E406           CPX FLAG
1230 10E1 B005           BCS ROP
1240 10E3 9100           STA (WHRL),Y
```

```
1250 10E5 4C0B11          JMP NEXT
1260 10E8 48        NOP   PHA
1270 10E9 A90F            LDA #$F
1280 10EB 8D02F0          STA $F002
1290 10EE 68              PLA
1300 10EF 48              PHA
1310 10F0 4A              LSR A
1320 10F1 4A              LSR A
1330 10F2 4A              LSR A
1340 10F3 4A              LSR A
1350 10F4 202711          JSR DIGIT
1360 10F7 68              PLA
1370 10F8 202711          JSR DIGIT
1380 10FB A920            LDA #' '
1390 10FD 206010          JSR OUTCH
1400 1100 C604            DEC CNT
1410 1102 D00C            BNE NEXTR
1420 1104 201D11          JSR CRLF
1430 1107 A908            LDA #8
1440 1109 8504            STA CNT
1450 110B A90F      NEXT  LDA #$F
1460 110D 8D02F0          STA $F002
1470 1110 C8        NEXTR INY
1480 1111 D0C9            BNE ADR
1490 1113 E601            INC WHRH
1500 1115 E603            INC SAMH
1510 1117 E8              INX
1520 1118 E006            CPX #6
1530 111A D0C0            BNE ADR
1540 111C 60              RTS
1550 111D           ;
1560 111D A90D      CRLF  LDA #$D
1570 111F 206010          JSR OUTCH
1580 1122 A90A            LDA #$A
1590 1124 4C6010          JMP OUTCH
1600 1127           ;
1610 1127 290F      DIGIT AND #$F
1620 1129 0930            ORA #$30
1630 112B C93A            CMP #$3A
1640 112D 9002            BCC NC1
1650 112F 6906            ADC #6
1660 1131 4C6010    NC1   JMP OUTCH
1670 1134           ;
1680 1134 A000      PRGM  LDY #0  PROGRAM PROM
1690 1136 8C01F0          STY $F001  INIT. PIA
1700 1139 8C03F0          STY $F003
1710 113C 88              DEY
1720 113D 8C00F0          STY $F000
1730 1140 8C02F0          STY $F002
1740 1143 A204            LDX #4
1750 1145 8E01F0          STX $F001
1760 1148 8E03F0          STX $F003
1770 114B C8              INY
1780 114C 8402            STY SAML
1790 114E 8407            STY FLIPF  ZERO FLIP FLOP
1800 1150 A9A0            LDA #$A0
1810 1152 8503            STA SAMH
1820 1154 A914            LDA #$14
1830 1156 8504            STA CNT
1840 1158 A90D      LOOP  LDA #$D
1850 115A 8D02F0          STA $F002
1860 115D 20B111          JSR TEN
```

```
1870 1160 201000        JSR WHERE
1880 1163 A914          LDA #$14
1890 1165 8504          STA CNT
1900 1167 A507          LDA FLIPF  DOUBLE PROGRAM 1ST BYTE
1910 1169 D006          BNE NTFST
1920 116B A901          LDA #1
1930 116D 8507          STA FLIPF
1940 116F D0E7          BNE LOOP
1950 1171 C8      NTFST INY
1960 1172 D0E4          BNE LOOP
1970 1174 E601          INC WHRH
1980 1176 E603          INC SAMH
1990 1178 A9A2          LDA #$A2
2000 117A C503          CMP SAMH
2010 117C D0DA          BNE LOOP
2020 117E 60            RTS
2030 117F               ;
2040 117F 8211    STRT  .WORD BLAST
2050 1181 54      LEN   .BYTE ENDR-BLAST+1
2060 1182               ;
2070 1182 B102    BLAST LDA (SAML),Y  THESE CAN'T RESIDE
2080 1184 A90C          LDA #$C       BETWEEN $A000-$B000
2090 1186 8D02F0        STA $F002
2100 1189 B100          LDA (WHRL),Y
2110 118B 8D00F0        STA $F000
2120 118E 203F00        JSR TEN-BLAST+WHERE
2130 1191 A908    PULSE LDA #8
2140 1193 8D02F0        STA $F002
2150 1196 A90C          LDA #$C
2160 1198 8D02F0        STA $F002
2170 119B 38            SEC
2180 119C A910          LDA #$10
2190 119E A280    TIM1  LDX #$80
2200 11A0 CA      TIM2  DEX
2210 11A1 D0FD          BNE TIM2
2220 11A3 E901          SBC #1
2230 11A5 B0F7          BCS TIM1
2240 11A7 C604          DEC CNT
2250 11A9 D0E6          BNE PULSE
2260 11AB A90F          LDA #$F
2270 11AD 8D02F0        STA $F002
2280 11B0 60            RTS
2290 11B1               ;
2300 11B1 EA      TEN   NOP
2310 11B2 EA            NOP
2320 11B3 EA            NOP
2330 11B4 EA            NOP
2340 11B5 60            RTS
2350 11B6               ;
2360 11B6 B102    OBTAN LDA (SAML),Y
2370 11B8 A90E          LDA #$E
2380 11BA 8D02F0        STA $F002
2390 11BD A505          LDA MODE
2400 11BF F011          BEQ RDR
2410 11C1 AD00F0        LDA $F000
2420 11C4 D100          CMP (WHRL),Y
2430 11C6 F005          BEQ NEXTE
2440 11C8 A940          LDA #'@
```

```
2450 11CA 206010        JSR OUTCH
2460 11CD 68       NEXTE PLA
2470 11CE 68             PLA
2480 11CF 4C0B11         JMP NEXT
2490 11D2 AD00F0   RDR    LDA $F000
2500 11D5 60       ENDR   RTS
2510 11D6                 .END
```

A2

```
;181000A2289A201D11A92920601CA9108500A0008401AD7F1185020763
;181018AD8011B503AE8111B1029100C8CAD0F8206310482066106809BD
;181030C956D006209610 4C0010C950D006203411 4C0010C94DD006080B
;18104820A7104C0010C952D006209A104C0010A93F2060104C0010068E
;1810604C0BFE4C00FEA201206C10CA207D100A0A0A0A950020 7D100747
;181078150095006020631 0C93030D4C93A300BC94130CCC94710C80966
;18109018E906290F60A9FF3013A200A5008502A5018503 8AF004A2095E
;1810A8FFA900860685 05A0008C01F08C00F0A2048E01F08C03F0880B53
;1810C08C02F08E03F0C8A505D004A506F0068402A9A08503A908850B5B
;1810D804201D1120 4400E406B00591004C0B1148A90F8D02F0684807 7D
;1810F04A4A4A4A202711682027 11A92020601 0C604D00C201D11A9074E
;18110808850 4A90F8D02F0C8D0C9E601E603E8E006D0C060A90D200BBE
;18112060 10A90A4C6010290F0930C93A900269064C6010A0008C010686
;181138F08C03F0888C00F08C02F0A2048E01F08E03F0C88402840 70BD1
;181150A9A08503A9148504A90D8D02F020B111201000A9148504A508C2
;1811680 7D006A901850 7D0E7C8D0E4E601E603A9A2C503D0DA60820D46
;1811801154B102A90C8D02F0B1008D00F0203F00A9088D02F0A90C0967
;1811988D02F038A910A280CAD0FDE901B0F7C604D0E6A90F8D02F00E32
;1811B060EAEAEAEA60B102A90E8D02F0A505F011AD00F0D100F0050D38
:0E11C8A94020601068684C0B11AD00F0600595
```