MOTOROLA
Semiconductor Products Inc.

AN906A/D
Application Note

SELF-PROGRAMMING THE MC68701 AND THE MC68701U4

Prepared By:
Patrick Svatek
Microprocessor Applications Engineering Department
Motorola Inc.
Austin, Texas

INTRODUCTION

The MC68701 and MC68701U4 are EPROM versions of the M6801 microcomputer (MCU) Family. The MC68701 on-chip resources include a 2K-byte EPROM, a three-function timer, a serial communication interface (SCI), up to 29 parallel lines, 128 bytes of RAM, and an oscillator. These resources give it extensive power and flexibility for ease of design. The MC68701U4 enhances the capabilities of the MC68701. Improved resources include a 4K-byte EPROM, two input-capture functions, three output-compare functions, a counter alternate address, and 192 bytes of RAM.

The MC68701/U4 MCUs can also program themselves. The MC68701/U4 CPU controls all movement of data into the on-chip EPROM during programming and requires only a few external devices to do the task. This application note explains how the MC68701/U4 MCUs program themselves and describes a fully-tested self programmer (including software and 1:1 artwork). The self-programmer includes a check to determine which of the two devices is being programmed.

ON-CHIP EPROM

A dual-purpose pin, $\overline{\text{RESET}}/V_{PP}$, is used to reset the MCU and to power the on-chip EPROM. This pin is normally at 5.0 volts during non-programming operations and must be raised to V_{PP} (21 V) during programming of the EPROM.

The MCU EPROM is controlled by two bits (PLC and PPC) in the RAM/EPROM control register (see Figure 1).

Bit 0 of the register is called the programming latch control (PLC) and is used to control an address latch used during programming of the EPROM. When PLC is set, the latch is transparent. When PLC is clear, the address latch is enabled and latches each EPROM address asserted by the CPU. The PLC should be set during normal nonprogramming MCU operation and should be cleared only to program the EPROM. This bit is set during reset and can be cleared only in mode 0.

Bit 1 of the RAM/EPROM control register is called programming power control (PPC) and is used to gate programming power (V_{PP}) to the EPROM during programming. When PPC is set, V_{PP} is not applied to the EPROM. During normal nonprogramming operation, PPC should be set. The PPC bit should be cleared only to program the EPROM. This bit is set during reset and whenever the PLC bit is set. Bit 1 can be cleared only in mode 0 with the PLC bit clear.

The MC68701/U4 MCUs are programmed in mode 0. In this mode, all the interrupt and reset vectors are located at \$BFFO — \$BFFF. The on-chip EPROM for the MC68701 and MC68701U4 are located at \$F800 — \$FFFF and \$F000 — \$FFFF, respectively. The reset vectors direct the CPU to a bootstrap program that will fetch data sequentially from external memory or a peripheral controller and program each byte into the MCU EPROM. Once V_{PP} is applied to the $\overline{\text{RESET}}/V_{PP}$ pin, each data byte is programmed as follows:

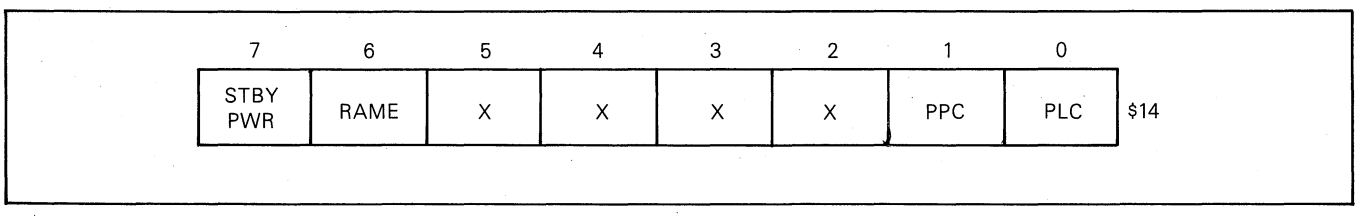


FIGURE 1 — RAM/EPROM Control Register

1. Apply programming power (V_{PP} = 21 V) to the $\overline{RESET}/V_{PP}$ pin.
2. Clear the PLC control bit and set the PCC bit by writing $FE to the RAM/EPROM control register.
3. Write data to the next EPROM location to be programmed. When triggered by a MPU write to the EPROM, internal latches capture both the EPROM address and the data byte.
4. Clear the PPC bit for programming time (t_{PP}) by writing $FC to the RAM/EPROM control register. This step gates V_{PP} from the $\overline{RESET}/V_{PP}$ pin to the EPROM.
5. Repeat Steps 1-4 for each byte to be programmed.
6. Set the PLC and PPC bits by writing $FF to the RAM/EPROM control register.
7. Remove the programming power (V_{PP}) from the $\overline{RESET}/V_{PP}$ pin. The EPROM can now be read and verified.

A MC68701/U4 SELF-PROGRAMMER

The MC68701/U4 self-programmer (see Figure 2) is designed for simplicity, low cost, and ease of use. The hardware and associated software provide for: (1) determination of which device type is being programmed, (2) verification that the inserted MCU is initially fully erased, (3) the programming of the MCU, and (4) verification of the programmed code.

After applying power, the user just toggles one switch and then monitors three LEDs which indicate MCU EPROM status. The self-programmer will enter either 2K or 4K bytes of the external 8K U4 EPROM into the MCU EPROM depending on which device is being programmed.

A copy of the 1:1 artwork necessary to fabricate a printed circuit board (PCB) for the self-programmer can be found at the end of this application note. In addition, a list of parts necessary to complete the PCB is furnished.

USING THE SELF-PROGRAMMER

To use the self-programmer, one does not need knowledge of the MC68701/U4 operation. However, a little knowledge of electronics is needed to program a device. Five steps are required as follows:

1. Insert the U4 EPROM containing the code to be programmed.
2. Insert the desired MCU (MC68701) or MC68701U4) into its socket.
3. Apply power using switch S1.
4. Set switch S2 to the program position.
5. Monitor the LEDs.

Shortly after switch S2 is set to the program position, LED #1 (ERASE) should light indicating that the MCU EPROM is fully erased. At this point, the self-programmer has determined which of the two devices will be programmed. Within a few seconds, LED #1 will turn off and MCU EPROM programming will begin.

Approximately 105 (MC68701) or 210 (MC68701U4) seconds later, either (1) LED #2 (PASS) should light indicating that the MC68701/U4 is programmed and its contents have been verified or (2) LED #3 (FAIL) will light indicating that the MCU EPROM has failed verification after programming. At this time, switch S2 should be toggled to the RESET position and the power removed (S1). Another MCU may now be programmed.

If LED #1 (ERASE) and LED #3 (FAIL) both light, then the MCU is not fully erased. The self-programmer will make no further attempt to check for full erasure of the MCU.

The LEDs are color-coded to provide readily recognized pass and fail indications. LED #1 (ERASE) is amber, LED #2 (PASS) is green, and LED #3 (FAIL) is red. Zero insertion force sockets should be used for the MCU and the program U4 EPROM to simplify the use of the self-programmer.

CIRCUIT DESCRIPTION

The self-programmer consists of two MCM68766 EPROMs, a SN74LS373 transparent latch, a SN74LS138 1-of-8 decoder, a MCU socket, and associated parts as shown in Figure 2.

A 4-MHz crystal is used to obtain a 1-MHz clock operation. If another clock frequency is used, a change in the bootstrap software (MINPRGU4) will be required to ensure at least 50 milliseconds of programming time for each byte entered into the MCU EPROM. Byte programming time is governed by WAIT in MINPRGU4 and is indirectly related to the MCU clock frequency. An increase in the MCU clock frequency requires a proportional increase in the value of WAIT. A decrease in clock frequency should, likewise, be reflected in the value of WAIT.

The MCU can be optionally driven by an external TTL clock at pin 3 (with pin 2 grounded). If this option is used, the capacitors shown connected to pins 2 and 3 are not required.

Pins 8, 9, and 10 are connected to ground to place the MCU in mode 0 (programming mode) on the rising edge of $\overline{RESET}$. The $\overline{IRQ}$ and $\overline{NMI}$ pins are connected as logic high to eliminate external interrupts.

The $\overline{RESET}/V_{PP}$ pin is driven by a circuit that provides three voltage levels to this pin. Before applying power with switch S1, the user should place switch S2 in the $\overline{RESET}$ position. This action forces the $\overline{RESET}/V_{PP}$ pin low. The second voltage level, established to toggling switch S2 to the PROG position, brings the MCU out of a $\overline{RESET}$ condition. The mode of operation (mode 0) is established during the rising edge of $\overline{RESET}$. The MCU fetches the RESTART vector now located at $BFFE — $BFFF and executes the bootstrap program.

During programming, 21 volts is applied to the $\overline{RESET}/V_{PP}$ pin by the transistor pair, Q1 and Q2. Initially, transistor Q1 is on and transistor Q2 is off. Port pin P14 (pin 17) is set low forcing Q1 to turn off. With Q1 off, a Zener voltage of 22 volts is established at the base of Q2 forcing Q2 to conduct and reference the Q2 emitter and the $\overline{RESET}/V_{PP}$ pin to approximately 21.3 volts.

A SN74LS373 latch is used to demultiplex port 3 which is used both as a lower address port (A0-A7) and as a data port. An address strobe from the MCU is connected to LE of the SN74LS373 to latch the lower addresses at the proper time during each bus cycle. Once the addresses are latched, the port is used to data transfer.

A SN74LS373 1-of-8 decoder is used to address decoding of two external 8K EPROMs. The external EPROM containing the user program is decoded at $6000 — $7FFF while the bootstrap program is decoded at $A000 — $BFFF. The SN74LS138 decoder is gated with the MCU E clock to ensure that the EPROM drivers are in a high impedance during E clock low cycle time thus eliminating contention on the lower multiplexed address/data bus.

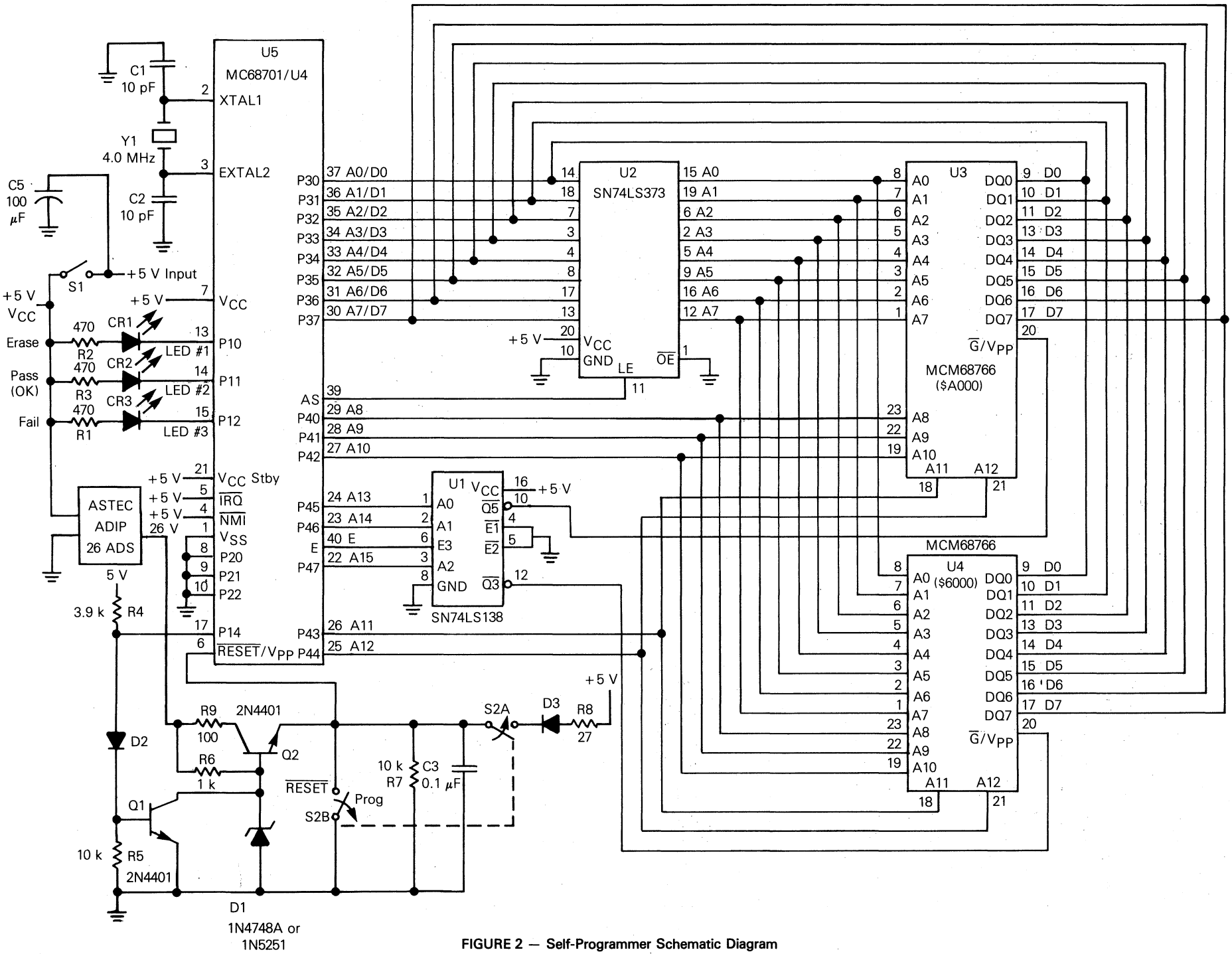


FIGURE 2 — Self-Programmer Schematic Diagram

MEMORY MAP

The self-programmer memory map consists of five address spaces and is shown in Figure 3. Four of the address spaces are fixed by the MCU during programming and cannot be relocated. These spaces consist of a MCU internal register area ($0000 — $001F) and MCU external interrupt vectors ($BFFO — $BFFF). The other two areas are device dependent and are listed below:

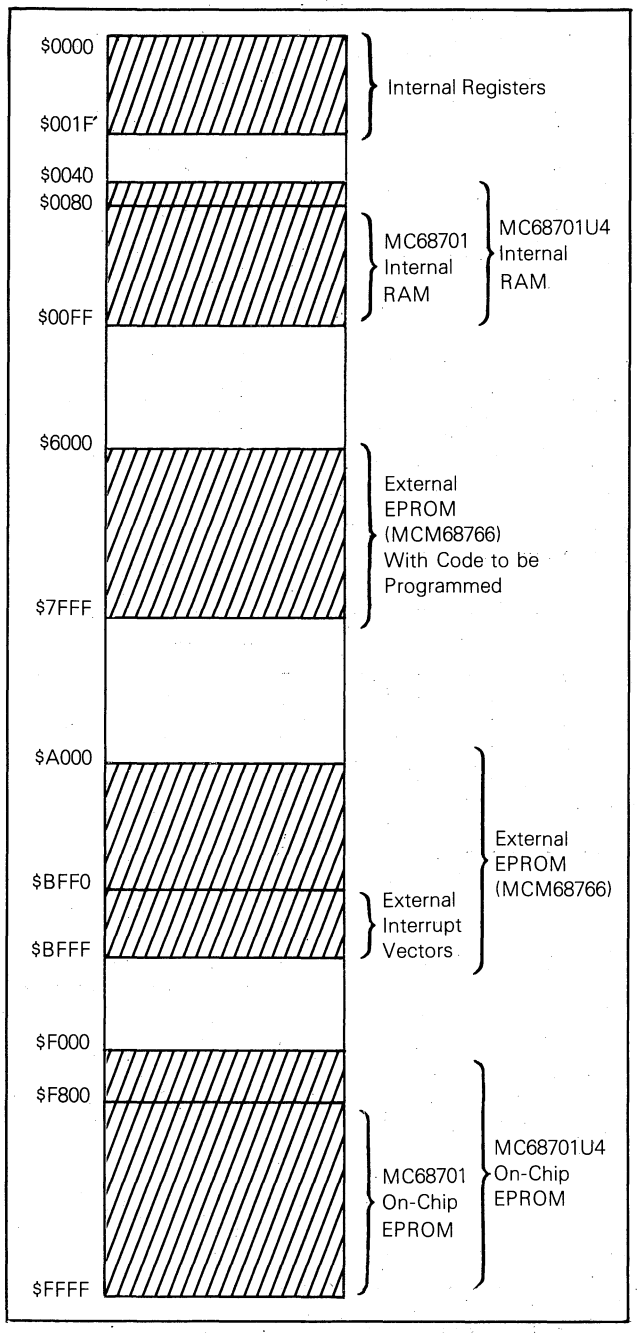


FIGURE 3 — Self-Programmer Memory Map

Function	MC68701	MC68701U4
MCU Internal RAM	$0080 — $00FF	$0040 — $00FF
MCU Internal EPROM	$F800 — $FFFF	$F000 — $FFFF

The fifth address space is used for an MCM68766 8K EPROM which contains the code to be entered into the MCU on-chip EPROM. This MCM68766 EPROM has been arbitrarily located at $6000 — $7FFF and can be relocated for a custom programmer design. Since the MCM68766 is a 8K EPROM, the user will have to locate this program in the upper 2K bytes ($7800 — $7FFF) or upper 4K ($7000 — $7FFF) for programming a MC68701 or a MC68701U4, respectively.

The user should map MINPRGU4 at address $1800 — $1FFF within U3 EPROM. The MCU program should reside at $1800 — $1FFF (MC68701) and $1000 — $1FFF (MC68701U4) within U4 EPROM for correct correspondence with the memory maps.

PROGRAM DESCRIPTION

The self-programmer uses a bootstrap program, MINPRGU4, to control programming of the MCU EPROM. The program performs the following functions:

1. Initializes the MCU.
2. Determines whether a MC68701 or MC68701U4 MCU is being programmed.
3. Checks that the EPROM is fully erased.
4. Programs the EPROM.
5. Verifies the program.

The MINPRGU4 bootstrap program also controls the state of the three LEDs that indicate the programming status of the MCU. A detailed flowchart of MINPRGU4 is shown in Figure 4. A complete listing is presented at the back of this application note.

PROGRAM MODIFICATIONS AND CONSIDERATIONS

Additions or modifications to MINPRGU4 can be made by inserting routines between the basic blocks shown on the flowchart in Figure 4. For convenience, the start and stop addresses of each block are located directly to the left of each block (see Figure 4).

Parameters IMBEG, IMEND, PNTR, and WAIT (stored in RAM locations $80 — $87) determine the size of the data block to be programmed into the MCU, the first MCU EPROM location to be programmed, and the time period that V_{PP} will be applied to the EPROM. These parameters can be changed to allow programming of selected EPROM locations and to allow changes in the MCU operating frequency. These parameters, once selected, should remain constant during programming.

One modification to MINPRGU4 can be verification of the MCU EPROM if the EPROM is not fully erased. This is an alternative to lighting LEDs #1 and #3 and waiting. This modification allows verification of MCUs that have been previously programmed and used.

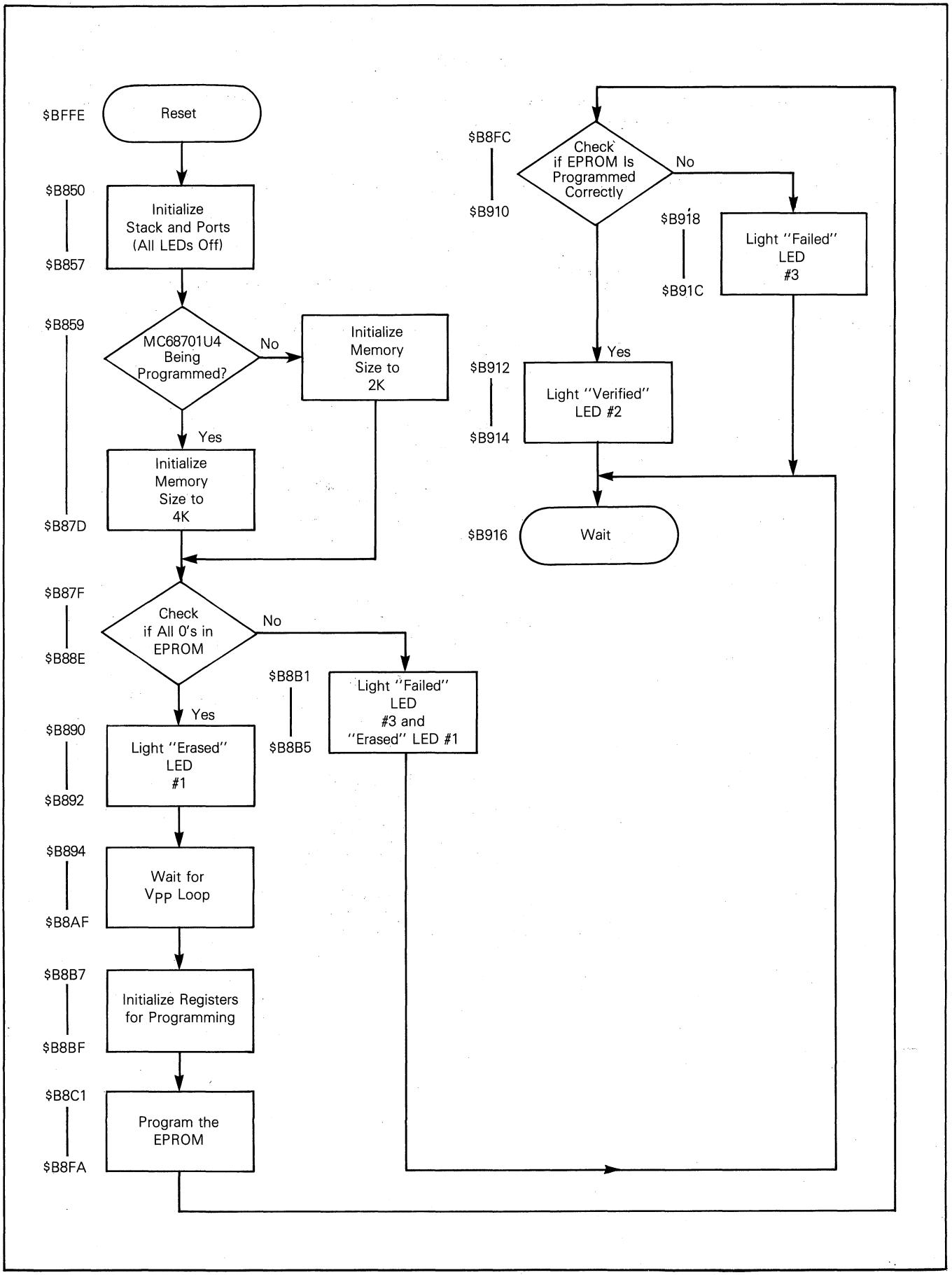


FIGURE 4 — Flow Chart for MINPRGU4

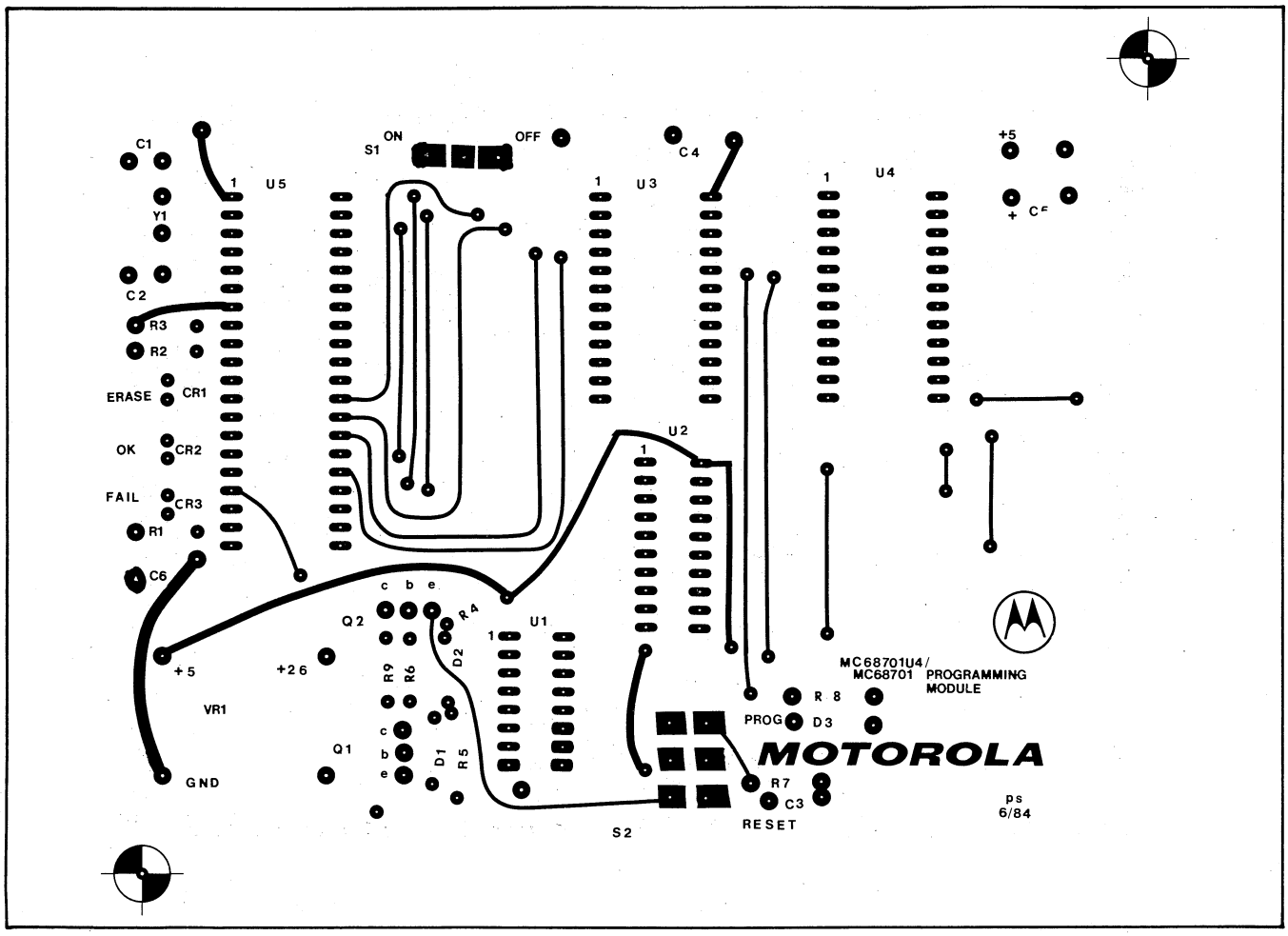
APPENDIX A

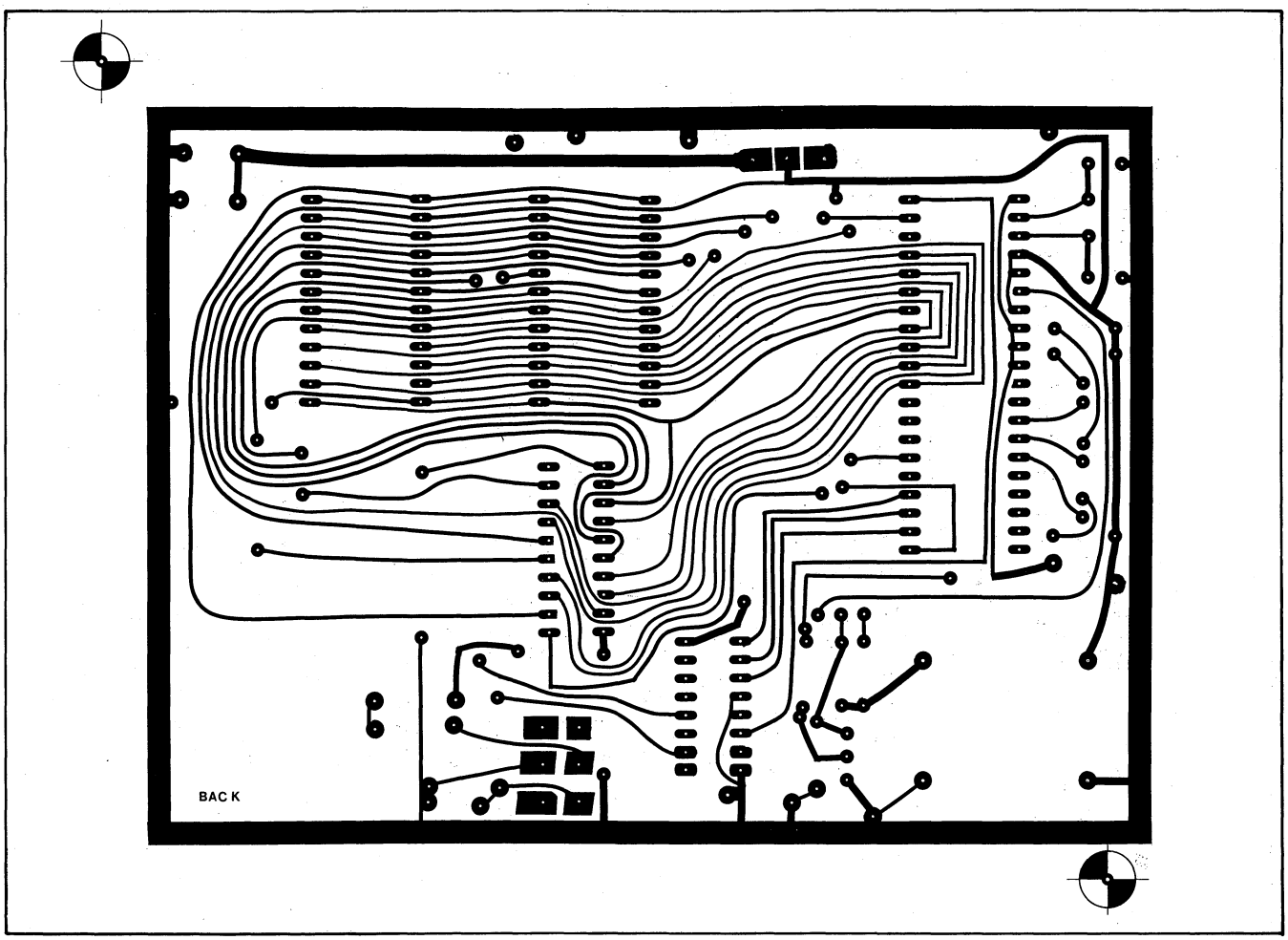
This appendix provides a copy of the 1:1 artwork necessary to fabricate a printed circuit board (PCB) for the self-programmer. In addition, a parts list if furnished to allow the user to complete the PCB.

NOTE

Permission is hereby granted by Motorola, Inc., Microprocessor Products Division, in Austin, Texas for use of this artwork.

Qty.	Reference Design	Value/Description
		Resistors (1/4 Watt)
3	R1-R3	470 ohms
1	R4	3.9 kilohms
2	R5,R7	10 kilohms
1	R6	1.0 kilohms
1	R8	27 ohms (1/2 watt)
1	R9	100 ohms
		Diodes/Transistors
2	Q-Q2	2N4401 transistor (NPN)
1	D1	1N4748A or 1N5251 Zener (22 V ±5%)
2	D2, D3	Silicon (1N3064, 1N4148, etc.)
3	CR1-CR3	LED
		Switches
1	S1	SPDT American ST1-1 or C & K 7101
1	S2	DPDT C & K 7201
		Capacitors
2	C1, C2	10 pF
1	C3	0.1 μF
2	C4, C5	100 μF, 35 V
		Motorola ICs
1	U1	SN74LS138 Decoder
1	U2	SN74LS373 Latch
2	U3, U4	MC68766 8K × 8 EPROM
1	U5	MC68701 or MC68701U4 MCU
		Miscellaneous
1	Y1	4.0 MHz Crystal (NYMPH)
1	—	ASTEC ADIP 26ADS (26 V)


C1
Y1
C2
R3
R2
ERASE
CR1
OK
CR2
FAIL
CR3
R1
C6
+5
GND
VR1
+26
U5
S1
ON
OFF
U3
C4
U4
+5
C5
U2
U1
Q2
Q1
c b e
R4
D2
R9
R6
D1
R5
S2
RESET
PROG
R 8
D3
R7
C3
MC68701U4/
MC68701 PROGRAMMING
MODULE
MOTOROLA
ps
6/84


BACK

```
PAGE  001  MINPRGU4.SA:1

00001                    *
00002                    *
00003                           OPT    Z01,LLE=96
00004                    *
00005                    *
00006                    *      THIS PROGRAM WILL CHECK, PROGRAM AND VERIFY
00007                    *      THE MC68701 OR THE MC68701U4 EPROM.  IT ALSO
00008                    *      DETERMINES WHETHER A MC68701 OR A MC68701U4 IS
00009                    *      BEING PROGRAMMED.
00010                    *
00011                    *
00012                    *           E Q U A T E S
00013           0000  A P1DDR  EQU    $00      PORT 1 DATA DIR. REGISTER
00014           0002  A P1DR   EQU    $02      PORT 1 DATA REGISTER
00015           0008  A TCSR   EQU    $08      TIMER CONTROL/STAT REGISTER
00016           0009  A TIMER  EQU    $09      COUNTER REGISTER
00017           000B  A OUTCMP EQU    $0B      OUTPUT COMPARE REGISTER
00018           0014  A EPMCNT EQU    $14      RAM/EROM CONTROL REGISTER
00019           0018  A TCR2   EQU    $18      TIMER/CONTROL REG. 2
00020                    *
00021                    *          L O C A L   V A R I A B L E S
00022                    *
00023A 0080                     ORG    $80
00024A 0080     0002  A IMBEG  RMB    2        START OF MEMORY BLOCK
00025A 0082     0002  A IMEND  RMB    2        LAST BYTE OF MEMORY BLOCK
00026A 0084     0002  A PNTR   RMB    2        FIRST BYTE OF EPROM TO BE PGM'D
00027A 0086     0002  A WAIT   RMB    2        COUNTER VALUE
00028                    *
00029A B850                     ORG    $B850
00030A B850 8E 00FF   A START  LDS    #$FF     INITIALIZE STACK
00031A B853 86 17     A        LDAA   #$17     INIT. PORT 1
00032A B855 97 00     A        STAA   P1DDR    DDR
00033A B857 97 02     A        STAA   P1DR     DATA REGISTER (ALL LED'S OFF)
00034                    *                                (NO Vpp APPLIED)
00035                    *
00036                    *             DETERMINE WHETHER A MC68701 OR A MC68701U4
00037                    *             IS BEING PROGRAMMED.
00038                    *
00039A B859 96 18     A        LDAA   TCR2     TCR2  = $03 ON RESET
00040A B85B 81 03     A        CMPA   #%00000011 IF 701U4, THIS VALUE
00041A B85D 27 16 B875         BEQ    P4K      GO TO '701U4 MEMORY SETUP
00042A B85F 86 FE     A        LDAA   #$FE     SECOND CHECK
00043A B861 97 18     A        STAA   TCR2     WRITE A ZERO TO TCR2-0 (CLOCK)
00044A B863 96 18     A        LDAA   TCR2     NOW READ IT BACK
00045A B865 84 01     A        ANDA   #$01     MASK CLOCK BIT
00046A B867 27 0C B875         BEQ    P4K      MC68701U4 IF "Z" = 1
00047                    *
00048                    *        INITIALIZE EPROM MEMORY SIZE TO MC68701(2K)
00049                    *
00050A B869 CC 7800   A        LDD    #$7800   START OF EPROM
00051A B86C DD 80     A        STD    IMBEG
00052A B86E CC F800   A        LDD    #$F800   START OF '701 EPROM
00053A B871 DD 84     A        STD    PNTR
00054A B873 20 0A B87F         BRA    BLKROM
00055                    *
00056                    *        INITIALIZE EPROM MEMORY SIZE TO MC68701U4(4K)
00057                    *
00058A B875 CC 7000   A P4K    LDD    #$7000   START OF EPROM
```

```
PAGE  002  MINPRGU4.SA:1

00059A B878 DD 80    A         STD    IMBEG
00060A B87A CC F000  A         LDD    #$F000   START OF '701U4 EPROM
00061A B87D DD 84    A         STD    PNTR
00062                  *
00063                  *         B L A N K   C H E C K
00064                  *
00065A B87F DE 84    A BLKROM  LDX    PNTR     CHECK IF EPROM ERASED
00066A B881 C6 00    A         LDAB   #$00     GET READY FOR CMPR.
00067A B883 A6 00    A ERASE   LDAA   0,X      LOAD EPROM CONTENTS
00068A B885 11                 CBA             COMPARE TO ZERO
00069A B886 26 29 B8B1         BNE    ERROR1   BRANCH IF NOT ZERO
00070A B888 8C FFFF  A         CPX    #$FFFF   CHECK IF DONE
00071A B88B 27 03 B890         BEQ    NEXT     IF SO BRANCH
00072A B88D 08                 INX             GO AGAIN
00073A B88E 20 F3 B883         BRA    ERASE
00074                  * *
00075A B890 86 16    A NEXT    LDAA   #$16     TURN ON ERASED LED
00076A B892 97 02    A         STAA   P1DR
00077                  * *
00078                  *         D E L A Y   L O O P   (3.5 SEC)
00079                  *
00080A B894 DF 86    A         STX    WAIT
00081A B896 CE 0046  A         LDX    #$0046   GET READY FOR 70 TIMES THRU LOOP
00082A B899 09           STALL1 DEX
00083A B89A CC C350  A         LDD    #$C350   INIT. 50MS LOOP
00084A B89D D3 09    A         ADDD   TIMER    BUMP CURRENT VALUE
00085A B89F 7F 0008  A         CLR    TCSR     CLEAR OCF
00086A B8A2 DD 0B    A         STD    OUTCMP   SET OUTPUT COMPARE
00087A B8A4 86 40    A         LDAA   #$40     NOW WAIT FOR OCF
00088A B8A6 95 08    A STALL2  BITA   TCSR
00089A B8A8 27 FC B8A6         BEQ    STALL2   NOT YET
00090A B8AA 8C 0000  A         CPX    #$0000   70 TIMES YET?
00091A B8AD 26 EA B899         BNE    STALL1   NOPE
00092A B8AF 20 06 B8B7         BRA    PGINT
00093                  * *
00094A B8B1 86 02    A ERROR1  LDAA   #$02     LIGHT ERROR AND ERASE LED
00095A B8B3 97 02    A         STAA   P1DR
00096A B8B5 20 5F B916         BRA    SELF
00097                  * *
00098A B8B7 CE 7FFF  A PGINT   LDX    #$7FFF   INIT. IMEND
00099A B8BA DF 82    A         STX    IMEND
00100A B8BC CE C350  A         LDX    #$C350   INIT. WAIT  (4.0 MHZ)
00101A B8BF DF 86    A         STX    WAIT
00102                  *
00103                  *         P R O G A M M I N G   L O O P
00104                  *
00105A B8C1 86 07    A EPROM   LDAA   #$07     TURN OFF LEDS AND APPLY Vpp
00106A B8C3 97 02    A         STAA   P1DR
00107A B8C5 DE 84    A         LDX    PNTR     SAVE CALLING ARGUMENT
00108A B8C7 3C                 PSHX            RESTORE WHEN DONE
00109A B8C8 DE 80    A         LDX    IMBEG    USE STACK
00110A B8CA 3C           EPR002 PSHX            SAVE POINTER ON STACK
00111A B8CB 86 FE    A         LDAA   #$FE     REMOVE VPP, SET LATCH
00112A B8CD 97 14    A         STAA   EPMCNT   PPC=1,PLC=0
00113A B8CF A6 00    A         LDAA   0,X      MOVE DATA MEMORY-TO-LATCH
00114A B8D1 DE 84    A         LDX    PNTR     GET WHERE TO PUT IT
00115A B8D3 A7 00    A         STAA   0,X      STASH AND LATCH
00116A B8D5 08                 INX             NEXT ADDR.
```

```
PAGE  003  MINPRGU4.SA:1

00117A B8D6 DF 84    A          STX    PNTR      ALL SET FOR NEXT
00118A B8D8 86 FC    A          LDAA   #$FC      ENABLE EPROM POWER (VPP)
00119A B8DA 97 14    A          STAA   EPMCNT    PPC=0,PLC=0
00120                 *
00121                 *       NOW WAIT 50 MSEC TIMEOUT USING COMPARE
00122                 *
00123A B8DC DC 86    A          LDD    WAIT      GET CYCLE COUNTER
00124A B8DE D3 09    A          ADDD   TIMER     BUMP CURRENT VALUE
00125A B8E0 7F 0008  A          CLR    TCSR      CLEAR OCF
00126A B8E3 DD 0B    A          STD    OUTCMP    SET OUTPUT COMPARE
00127A B8E5 86 40    A          LDAA   #$40      NOW WAIT FOR OCF
00128A B8E7 95 08    A EPR004   BITA   TCSR
00129A B8E9 27 FC B8E7          BEQ    EPR004    NOT YET
00130A B8EB 38                  PULX             SET UP FOR NEXT ONE
00131A B8EC 08                  INX              NEXT
00132A B8ED 9C 82    A          CPX    IMEND     MAYBE DONE
00133A B8EF 23 D9 B8CA          BLS    EPR002    NOT YET
00134A B8F1 86 17    A          LDAA   #$17      REMOVE Vpp AT PIN
00135A B8F3 97 02    A          STAA   P1DR
00136A B8F5 86 FF    A          LDAA   #$FF      REMOVE VPP, INHIBIT LATCH
00137A B8F7 97 14    A          STAA   EPMCNT    EPROM CAN NOW BE READ
00138A B8F9 38                  PULX             RESTORE PNTR
00139A B8FA DF 84    A          STX    PNTR
00140                 *
00141                 *          V E R I F Y   N E W   C O D E
00142                 *
00143A B8FC DE 80    A          LDX    IMBEG     SET UP POINTER
00144A B8FE 3C          VERF2   PSHX             SAVE POINTER ON STACK
00145A B8FF A6 00    A          LDAA   0,X       GET DESIRED DATA
00146A B901 DE 84    A          LDX    PNTR      GET EPROM ADDR.
00147A B903 E6 00    A          LDAB   0,X       GET DATA TO BE CHECKED
00148A B905 11                  CBA              CHECK IF SAME
00149A B906 26 10 B918          BNE    ERROR2    BRANCH IF ERROR(LIGHT LED)
00150A B908 08                  INX              NEXT ADDR
00151A B909 DF 84    A          STX    PNTR      ALL SET FOR NEXT
00152A B90B 38                  PULX             SETUP FOR NEXT ONE
00153A B90C 08                  INX              NEXT
00154A B90D 8C 8000  A          CPX    #$8000    MAYBE DONE
00155A B910 26 EC B8FE          BNE    VERF2     NOT YET
00156                 * *
00157A B912 86 15    A          LDAA   #$15
00158A B914 97 02    A          STAA   P1DR      LIGHT VERIFY LED
00159                 * *
00160A B916 20 FE B916 SELF     BRA    SELF      WAIT FOREVER
00161                 * *
00162A B918 86 13    A ERROR2   LDAA   #$13      LIGHT ERROR  LED
00163A B91A 97 02    A          STAA   P1DR
00164A B91C 20 F8 B916          BRA    SELF
00165                 * *
00166                 *       R E S T A R T   A N D   I N T R . V E C .
00167                 *
00168A BFF0                     ORG    $BFF0
00169A BFF0    B916  A          FDB    SELF
00170A BFF2    B916  A          FDB    SELF
00171A BFF4    B916  A          FDB    SELF
00172A BFF6    B916  A          FDB    SELF
00173A BFF8    B916  A          FDB    SELF
00174A BFFA    B916  A          FDB    SELF
```

```
PAGE  004  MINPRGU4.SA:1

00175A BFFC    B916  A        FDB    SELF
00176A BFFE    B850  A        FDB    START
00177                         END
TOTAL ERRORS 00000--00000


   B87F BLKROM 00054 00065*
   0014 EPMCNT 00018*00112 00119 00137
   B8C1 EPROM  00105*
   B8CA EPR002 00110*00133
   B8E7 EPR004 00128*00129
   B883 ERASE  00067*00073
   B8B1 ERROR1 00069 00094*
   B918 ERROR2 00149 00162*
   0080 IMBEG  00024*00051 00059 00109 00143
   0082 IMEND  00025*00099 00132
   B890 NEXT   00071 00075*
   000B OUTCMP 00017*00086 00126
   0000 P1DDR  00013*00032
   0002 P1DR   00014*00033 00076 00095 00106 00135 00158 00163
   B875 P4K    00041 00046 00058*
   B8B7 PGINT  00092 00098*
   0084 PNTR   00026*00053 00061 00065 00107 00114 00117 00139 00146 00151
   B916 SELF   00096 00160*00160 00164 00169 00170 00171 00172 00173 00174 00175
   B899 STALL1 00082*00091
   B8A6 STALL2 00088*00089
   B850 START  00030*00176
   0018 TCR2   00019*00039 00043 00044
   0008 TCSR   00015*00085 00088 00125 00128
   0009 TIMER  00016*00084 00124
   B8FE VERF2  00144*00155
   0086 WAIT   00027*00080 00101 00123
```


MOTOROLA *Semiconductor Products Inc.*

3501 ED BLUESTEIN BLVD., AUSTIN, TEXAS 78721 • A SUBSIDIARY OF MOTOROLA INC.

A17777-3 PRINTED IN USA 8-87 IMPERIAL LITHO C50672 18,000 AN-906A/D