THE MC68701 MICROCOMPUTER UNIT

E.1 INTRODUCTION

The MC68701 Microcomputer Unit (MCU) is a monolithic computer that is nearly identical to the MC6801 MCU. The primary difference is that the read-only memory (ROM) in the MC68701 contains an unltraviolet (UV) Erasable Programmable Read-Only Memory (EPROM) instead of the masked ROM available in the MC6801.

The ability to program the EPROM allows the MC68701 to be used in a variety of applications. The MC68701 is particularly effective in:

- applications which are low in volume and do not warrant mask programming, and
- prototype equipment in which it is desired to debug the resident firmware before committing it to an MC6801 masked ROM.

Note that in the former case, compatibility with the MC6801 is inherently achieved. Should the product volume increase to economical levels, a mask programmed MC6801 could be substituted for the MC68701 with minimal changes to the system.

The MC68701 includes a clock oscillator, microprocessor unit (MPU), 2048 bytes of EPROM, 128 bytes of RAM, serial communications interface (SCI), programmable timer, and input/output pins. The resources, except for the EPROM, are identical to those of the MC6801. Because the MC68701 is so similar to the MC6801, this appendix will focus only on the differences between the two parts.

E.2 DIFFERENCES BETWEEN MC6801 AND MC68701 MCUs

The MC68701 contains 2048 bytes of Erasable Programmable Read-Only Memory (EPROM) which replaces the 2048 bytes of Read-Only Memory (ROM) contained in the MC6801. The differences between the two parts involve MC68701 features which support EPROM programming. The essential differences are: (1) the functional and electrical characteristics of the $\overline{\text{RESET}}$ pin, (2) the Mode 0 memory map, (3) mask options involving the ROM, and (4) the RAM Control Register ($14). Signal timing and other detailed information are presented in the MC68701 Data Sheet. Symbolic values used in this discussion — such as V_{PP} and t_{PP} — are defined quantitatively in the MC68701 Data Sheet.

E.2.1 MC68701 $\overline{\text{RESET}}$/V_{PP} Pin

The $\overline{\text{RESET}}$/V_{PP} pin for the MC68701 performs three functions: (1) it resets the microcomputer when the pin voltage falls below V_{IL}, (2) it is used as a control signal to capture the operating mode of the MCU, and (3) it provides an input for an EPROM programming voltage (V_{PP}) at a maximum current of I_{PP}.

THE MC68701 MICROCOMPUTER UNIT

E.1 INTRODUCTION

The MC68701 Microcomputer Unit (MCU) is a monolithic computer that is nearly identical to the MC6801 MCU. The primary difference is that the read-only memory (ROM) in the MC68701 contains an unltraviolet (UV) Erasable Programmable Read-Only Memory (EPROM) instead of the masked ROM available in the MC6801.

The ability to program the EPROM allows the MC68701 to be used in a variety of applications. The MC68701 is particularly effective in:

- applications which are low in volume and do not warrant mask programming, and
- prototype equipment in which it is desired to debug the resident firmware before committing it to an MC6801 masked ROM.

Note that in the former case, compatibility with the MC6801 is inherently achieved. Should the product volume increase to economical levels, a mask programmed MC6801 could be substituted for the MC68701 with minimal changes to the system.

The MC68701 includes a clock oscillator, microprocessor unit (MPU), 2048 bytes of EPROM, 128 bytes of RAM, serial communications interface (SCI), programmable timer, and input/output pins. The resources, except for the EPROM, are identical to those of the MC6801. Because the MC68701 is so similar to the MC6801, this appendix will focus only on the differences between the two parts.

E.2 DIFFERENCES BETWEEN MC6801 AND MC68701 MCUs

The MC68701 contains 2048 bytes of Erasable Programmable Read-Only Memory (EPROM) which replaces the 2048 bytes of Read-Only Memory (ROM) contained in the MC6801. The differences between the two parts involve MC68701 features which support EPROM programming. The essential differences are: (1) the functional and electrical characteristics of the $\overline{\text{RESET}}$ pin, (2) the Mode 0 memory map, (3) mask options involving the ROM, and (4) the RAM Control Register ($14). Signal timing and other detailed information are presented in the MC68701 Data Sheet. Symbolic values used in this discussion — such as V_{PP} and t_{PP} — are defined quantitatively in the MC68701 Data Sheet.

E.2.1 MC68701 $\overline{\text{RESET}}$/V_{PP} Pin

The $\overline{\text{RESET}}$/V_{PP} pin for the MC68701 performs three functions: (1) it resets the microcomputer when the pin voltage falls below V_{IL}, (2) it is used as a control signal to capture the operating mode of the MCU, and (3) it provides an input for an EPROM programming voltage (V_{PP}) at a maximum current of I_{PP}.

It is possible that an external Reset circuit for the MC6801 may not function with the MC68701 due to the input current requirements. An external circuit designed for the MC68701, however, can be designed to be compatible with the MC6801. For low volume applications, one should consider implementing the reset circuit for the MC68701 and thus achieving dual compatibility.

Several Reset circuits can be designed for the MC68701 depending upon the objectives of the designer. For example, a circuit could be designed for (a) both EPROM programming and normal operation, (b) normal operation only, or (c) EPROM programming only. Three circuits designed for (a) through (c) are shown in Figures E-1 through E-3, respectively. Each of the circuits has its advantages and disadvantages.

A general purpose Reset circuit for the MC68701 is shown in Figure E-1. The circuit provides the capability of switch selecting the operating mode and enabling or inhibiting programming power. Therefore, this circuit can be used to both program the EPROM and execute instructions from it depending upon the position of S1. In designing the $\overline{\text{RESET}}$ circuit, the reader should note that the current specifications (I_{in}) for the MC68701 and MC6801 are significantly different with respect to each other and with respect to whether or not the MC68701 is programming the EPROM.

In Figure E-1, compatibility between both MCUs with respect to I_{in} is achieved by using a value of V_{CC} for V. A voltage of V_{IH} must be achieved at the $\overline{\text{RESET}}/V_{PP}$ input at operating current, I_{in}. Assuming a voltage drop of 0.7 volt across the diode, D2, the voltage drop across the pullup resistor, R2, must equal $(V - V_{IH} - 0.7)$ volts. With a normal load current of I_{in}, the value (ohms) of the resistance, R2, can be calculated from

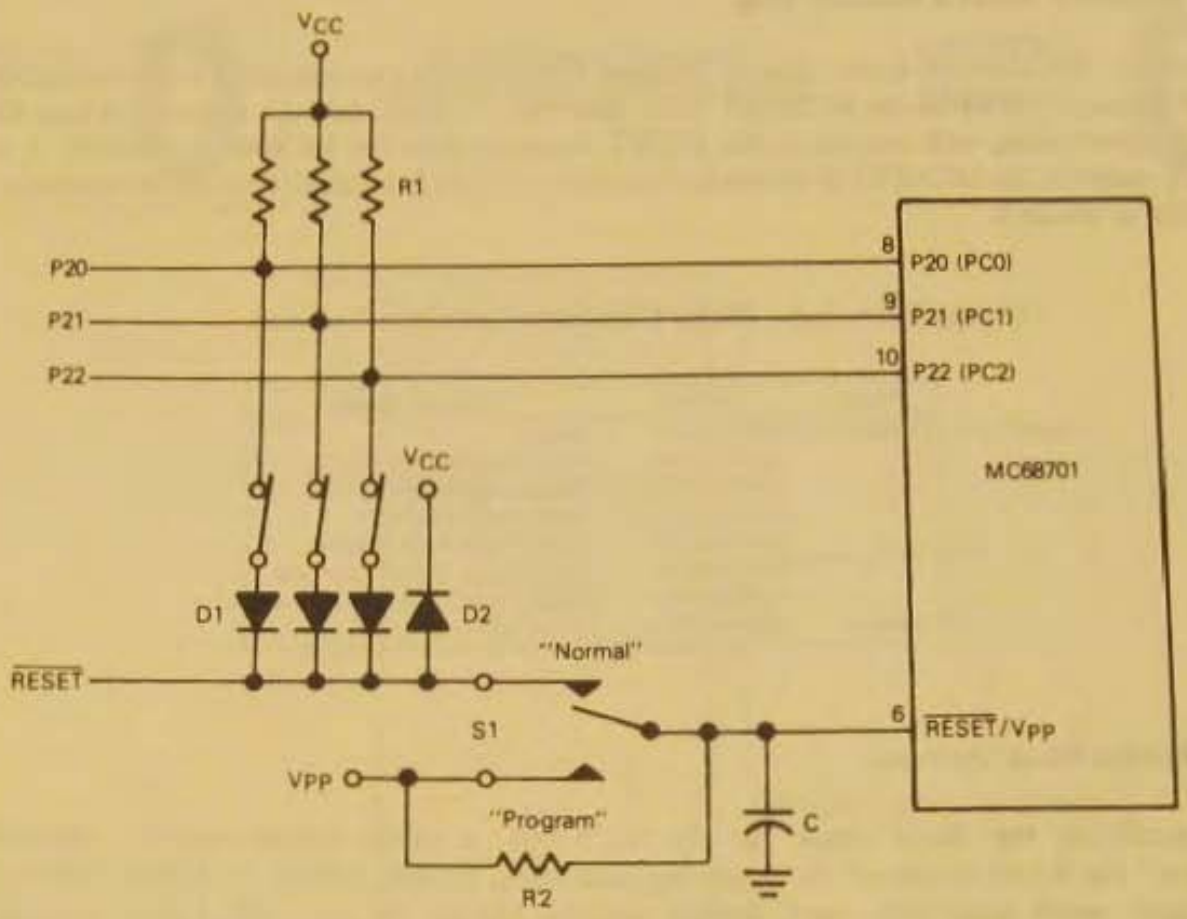
$$R2 = (V - V_{IH} - 0.7)/(I_{in})$$

The principal advantage of this circuit is that it requires only a single power supply (V_{CC}) except when programming the EPROM. Its principal disadvantage is the high power consumption in both the programming and normal modes due to amount of the current which must be sunk by the $\overline{\text{RESET}}$ circuit.

When programming the EPROM is not a consideration, the circuit of Figure E-2 can be used. It uses a single power supply and does not require the switch and diode of Figure E-1. However, it also results in high power consumption due to the amount of the current which must be sunk by the $\overline{\text{RESET}}$ circuit. Note this is the same $\overline{\text{RESET}}$ circuit recommended for the MC6801.

The advantages of the circuit in Figure E-3 are that it consumes less power and supports both normal operation and EPROM programming. However, its disadvantage is that V_{PP} is also required in normal operation. When compared with the circuit of Figure E-1, one finds that the lower power consumption is obtained by using a higher pullup voltage (V_{PP} instead of V) in series with a higher resistance for R2. A clamping diode keeps the level at the $\overline{\text{RESET}}$ value in the "Normal" switch position and maintains compatibility with the MC6801 $\overline{\text{RESET}}$ leakage current. The value of the resistor, R2, is obtained by using I_{in} (V_{in} High) for the $\overline{\text{RESET}}$ input and 2 milliamperes for the clamping diode. This results in a $(V_{PP} - V_{IH})$ voltage drop across the resistor with a current of $(I_{in} + 0.002)$ amperes or

$$R2 = (V_{PP} - V_{IH})/(I_{in} + 0.002) \text{ ohms}$$



Notes:

1. Mode 0 as shown.
2. R1 = 10k ohms (typical).
3. The $\overline{\text{RESET}}$ time constant is equal to RC where R is the equivalent parallel resistance of R2 and the number of resistors (R1) placed in the circuit by closed mode control switches.
4. D2 = 1N914, 1N4001 (typical).
5. $R2 = V/I = (V_{PP} - V_{IH})/(I_{in} + 0.002)$ ohms.
6. Switch S1 allows selection of normal ($\overline{\text{RESET}}$) or programming (V_{PP}) as the input to the $\overline{\text{RESET}}/V_{PP}$ pin. During switching, the input level is held at a value determined by R2 and V_{PP}.
7. See Data Sheet for typical diode for D1.
8. The diode, D2, clamps the maximum $\overline{\text{RESET}}$ input voltage to $(V_{PP} - V_{CC} - 0.7)$ volts for compatibility with the MC6801.

Figure E-3. MC68701 $\overline{\text{RESET}}$ Circuit for EPROM Programming

E.2.2 MC68701 Mode 0 Memory Map

In Mode 0, the interrupt vector area is changed from $FFF0 through $FFFF (in the MC6801) to $BFF0 through $BFFF in the MC68701. Note that this is a static address assignment that does not depend upon timing with respect to the $\overline{\text{RESET}}$ signal as does the MC6801 in Mode 0. A Mode 0 memory map for the MC68701 is shown in Figure E-4. Table E-1 lists the MC68701 interrupt vector locations in Mode 0.

Table E-1. Mode 0 External Interrupt Vectors

Priority	Location	Interrupt Vector
Highest	$BFFE:BFFF	RESET
	$BFFC:BFFD	Non-Maskable Interrupt
	$BFFA:BFFB	Software Interrupt (SWI)
	$BFF8:BFF9	IRQ1/Input Strobe B
	$BFF6:BFF7	IRQ2/Timer Input Capture
	$BFF4:BFF5	IRQ2/Timer Output Compare
	$BFF2:BFF3	IRQ2/Timer Counter Overflow
Lowest	$BFF0:BFF1	IRQ2/SCI Interrupt

E.2.3 MC6801 Mask Options

When specifying the ROM mask for the MC67801, a mask option may be selected which "relocates" the ROM to one of the following addresses: $C800, $D800, or $E800. Useful memory maps which result from this mask option include Modes 1R and 6R. Initial versions of the MC68701, however, do not support either of these modes. Therefore, the current Data Sheet should be referenced to determine availability.

NOTE

If attempting to emulate the MC6801 1R and 6R operating modes with an MC68701, a current MC68701 Data Sheet should be referenced to determine if these modes are supported.

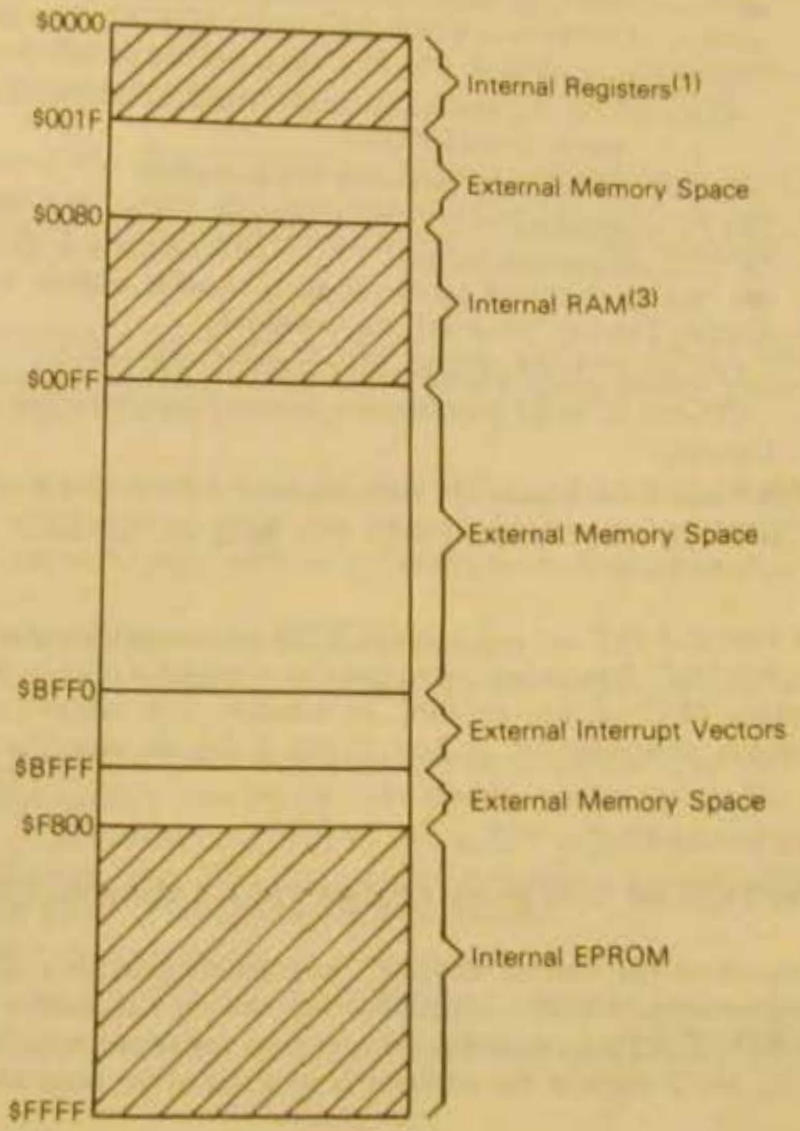
E.2.4 MC68701 RAM/EPROM Control Register ($14)

The RAM/EPROM Control Register provides a function similar to the RAM Control Register in the MC6801. The register contains four bits: STBY PWR, RAME, PPC, and PLC. The STBY PWR and RAME bits are described in the discussion for the RAM Control Register (Section 3.1.5.1.3). In the MC68701, two additional control bits (PPC and PLC) are included in the RAM/EPROM Control Register to facilitate programming the EPROM. The PLC and PPC bits are readable in all modes but can be changed only in Mode 0. The PLC bit can be written without restriction in Mode 0 but operation of the PPC bit is controlled by the value of PLC. A description of this register follows.

MC68701 Mode 0

Multiplexed Test and EPROM Programming Mode

MC68701 Mode 0



Notes:

(1) Excludes the following addresses which can be used externally: $04, $05, $06, $07, and $0F.

(2) There must be no overlapping of internal and external memory spaces to avoid driving the data bus with more than one device.

(3) Assumes RAME (RAM Enable bit) is set.

Figure E-4. MC68701 Memory Map for Mode 0

MC68701 RAM/EPROM Control Register

7	6	5	4	3	2	1	0	
STBY PWR	RAME	X	X	X	X	PPC	PLC	$14

Bit 0 PLC The Programming Latch Control bit controls (a) a latch which captures the EPROM address to be programmed and (b) whether the PPC bit can be cleared. The latch is triggered by an MPU write to a location in the EPROM. This bit is set by Reset and can be cleared only in Mode 0. The PLC bit is defined as follows:

PLC = 0 EPROM address latch enabled; EPROM address is latched during MPU writes to the EPROM.

PLC = 1 EPROM address latch is transparent.

Bit 1 PPC The Programming Power Control bit gates power from the $\overline{\text{RESET}}$/VPP pin to the EPROM programming circuit. PPC is set by Reset and whenever the PLC bit is set. It can be cleared only if (a) operating in Mode 0, and (b) if PLC has been previously cleared. The PPC bit is defined as follows:

PPC = 0 EPROM programming power (VPP) enabled.

PPC = 1 EPROM programming power (VPP) is not applied.

Bits 2-5 Unused.

Bit 6 RAME. RAM Enable bit. Refer to the RAM Control Register (Section 3.1.5.1.3).

Bit 7 STBY PWR. Standby Power bit. Refer to the RAM Control Register (Section 3.1.5.1.3).

Note that if PPC and PLC are set, they cannot be simultaneously cleared as the result of a single MPU write. The PLC bit must be cleared prior to attempting to clear PPC. If both PPC and PLC are clear, setting PLC will also set PPC. In addition, it is assumed that VPP is applied to the $\overline{\text{RESET}}$/VPP pin whenever PPC is clear. If this is not the case, the results to the EPROM are undefined.

E.3 DESCRIPTION OF INTERNAL EPROM PROGRAMMING CIRCUITRY

A block diagram of the internal EPROM programming circuitry is shown in Figure E-5. The EPROM programming circuitry consists of (a) address and data latches (b) RAM/EPROM Control Register, (c) $\overline{\text{RESET}}$/VPP programming power, and (d) associated control logic. The output data buffer used for MPU reads of the EPROM is also shown for completeness.

Data associated with an MPU write to an EPROM address is always captured in an 8-bit data latch. The 11-bit EPROM address latch is transparent providing the PLC bit is set. When PLC is clear, however, it latches the address during MPU writes to the EPROM. When VPP is subsequently applied to the EPROM by clearing the PPC bit, the "1's" in the data latch are programmed into the EPROM location specified by the address latch.

Programming power, Vpp, is used to program the EPROM and control some EPROM functions. Whenever Vpp is applied to the $\overline{\text{RESET}}$/Vpp pin, it is always provided to the EPROM control circuits. This could adversely affect the result of an EPROM MPU read.

NOTE

While Vpp is applied to the RESET/Vpp pin, the result of an EPROM MPU read is undefined regardless of the operating mode or value of PPC.

E.4 PROGRAMMING THE MC68701 EPROM

Ultraviolet erasure will clear all bits of the EPROM to the "0" state. Note that this erased state differs from that of some other widely used EPROMs (such as the MCM68708) where the erased state is a "1". The MC68701 EPROM is programmed by erasing it to "0's" and entering "1's" into the desired bit locations.

When the MC68701 is released from Reset in Mode 0, a vector is fetched from location $BFFE:$BFFF. This provides a method for an external program to obtain control of the microcomputer with access to every location in the EPROM.

To program the EPROM, it is necessary to operate the MC68701 in Mode 0 under the control of a program* resident in external memory which can facilitate loading and programming of the EPROM. After the pattern has been loaded into external memory, the EPROM can be programmed as follows:

a. Apply programming power (Vpp) to the $\overline{\text{RESET}}$/Vpp pin.
b. Clear the PLC control bit and set the PPC bit by writing $FE to the RAM/EPROM Control Register.
c. Write data to the next EPROM location to be programmed. Triggered by an MPU write to the EPROM, internal latches capture both the EPROM address and the data byte.
d. Clear the PPC bit for programming time, tpp, by writing $FC to the RAM/EPROM Control Register and waiting for time, tpp. This step gates the programming power (Vpp) from the $\overline{\text{RESET}}$/Vpp pin to the EPROM which programs the location.
e. Repeat steps b through d for each byte to be programmed.
f. Set PPC and PLC by writing $FF to the RAM/EPROM Register.
g. Remove the programming power (Vpp) from the $\overline{\text{RESET}}$/Vpp pin. The EPROM can now be read and verified.

Because the erased state of an EPROM byte is $00, it is not necessary to program a location which is to contain $00. Finally, it should be noted that the result of inadvertently programming a location more than once is the logical OR of the data patterns.

*A monitor called PRObug™ is available in a masked ROM which can be used to load a pattern and then program it into the EPROM. The monitor can be used with the MEX6801EVM Evaluation Module (see Appendix K). See PRObug manual for details. Motorola Microsystems, 3102 N. 56th St., Phoenix, Arizona 85018.

A routine which can be used to program the MC68701 EPROM is shown in Figure E-6. This non-reentrant routine requires four double byte variables named IMBEG, IMEND, PNTR, and WAIT to be initialized prior to entry to the routine. These variables indicate (a) the first and last memory locations which bound the data to be programmed into the EPROM, (b) the first EPROM location to be programmed, and (c) a quantity which can be used to generate the programming time delay. The last variable, WAIT, takes into account the MCU input crystal (or TTL-compatible clock) frequency to insure the programming time, t_{PP}, is met. WAIT is defined as the number of MPU E-cycles that will occur in the real-time EPROM programming interval, t_{PP}. For example, if $t_{PP} = 50$ milliseconds and the MC68701 is being driven with a 4.00 MHz TTL-compatible clock:

$$\begin{aligned}\text{WAIT (MPU E-cycles)} &= t_{PP}(\text{MCU INPUT FREQ})/4\times 10^6\\ &= (50000)(4\times 10^6)/4\times 10^6\\ &= 50000\end{aligned}$$

E.5 ERASING THE MC68701 EPROM

The MC68701 EPROM can be erased by exposing it to high-intensity ultraviolet light of a particular wavelength. The erasure time is a function of the intensity of the light and the exposure time. The MC68701 Data Sheet should be referenced for the details involved in selection of vendor equipment.

NOTE

The MC68701 transparent lid should always be covered with an opaque material after erasing. This shields both the EPROM and dynamic light-sensitive nodes from unintentional exposure to ultraviolet light.

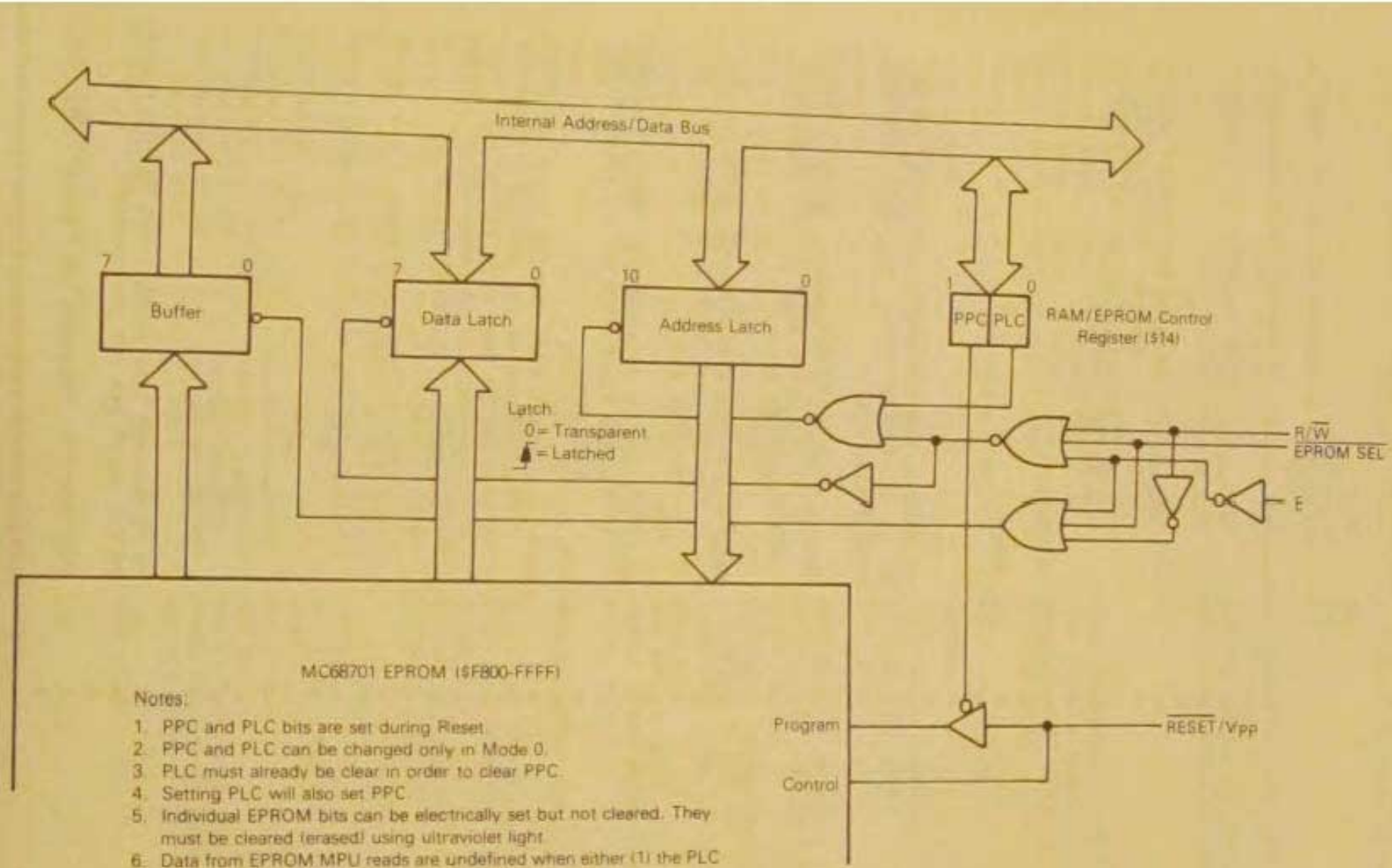


Figure E-5. Block Diagram of MC68701 EPROM Programming Circuit

```
PAGE  001  EPROM   .SA:1  EPROM  *** ROUTINE TO PROGRAM THE MC68701 EPROM ***

00001                       NAM    EPROM
00002                       OPT    Z01,LLEN=80
00003                       TTL    *** ROUTINE TO PROGRAM THE MC68701 EPROM **
00004
00005                 ************************************************************
00006                 *
00007                 *  E P R O M -- A  NON-REENTRANT  ROUTINE  TO  PROGRAM
00008                 *                 THE MC68701 EPROM.
00009                 *
00010                 *                 THE ROUTINE PROGRAMS THE MC68701 EPROM
00011                 *                 STARTING  AT  ADDRESS  "PNTR"  FROM A
00012                 *                 BLOCK OF MEMORY  STARTING  AT  "IMBEG"
00013                 *                 AND ENDING AT "IMEND".
00014                 *
00015                 *  CALLING CONVENTION:
00016                 *
00017                 *     JSR  EPROM
00018                 *
00019                 *  NOTES:
00020                 *
00021                 *     1.  THE ROUTINE EXPECTS FOUR DOUBLE BYTE  VALUES
00022                 *         TO  BE INITIALIZED PRIOR  TO  BEING  CALLED.
00023                 *         THESE VALUES ARE:
00024                 *
00025                 *         IMBEG = A DOUBLE BYTE ADDRESS  WHICH  POINTS
00026                 *                 TO THE FIRST BYTE TO  BE  PROGRAMMED
00027                 *                 INTO THE EPROM.
00028                 *
00029                 *         IMEND = A DOUBLE BYTE ADDRESS  WHICH  POINTS
00030                 *                 TO THE LAST BYTE TO BE PROGRAMED IN-
00031                 *                 INTO THE EPROM.
00032                 *
00033                 *         PNTR  = A DOUBLE BYTE ADDRESS  WHICH  POINTS
00034                 *                 TO THE FIRST BYTE IN THE EPROM TO BE
00035                 *                 PROGRAMMED.
00036                 *
00037                 *         WAIT  = A DOUBLE BYTE COUNTER VALUE WHICH IS
00038                 *                 A FUNCTION OF THE MCU INPUT FREQUEN-
00039                 *                 CY AND IS USED WITH THE OUTPUT  COM-
00040                 *                 PARE FUNCTION TO  GENERATE A 50 MSEC
00041                 *                 TIMEOUT.  IT IS EQUIVALENT TO
00042                 *
00043                 *                 50000 * (MCU INPUT FREQ) / 4 * 10**6
00044                 *
00045                 *                 VALUES FOR TYPICAL INPUT FREQS  ARE:
00046                 *
00047                 *                   WAIT                  MCU INPUT FREQ
00048                 *                 -------------           --------------
00049                 *                 30615 ($7797)             2.45 MHZ
00050                 *                 50000 ($C350)             4.00 MHZ
00051                 *                 61375 ($EF8F)             4.91 MHZ
00052                 *
00053                 *     2.  IT IS ASSUMED THAT POWER (VPP) IS  AVAILABLE
00054                 *         TO THE RESET PIN FOR PROGRAMMING.
00055                 *
00056                 *     3.  THIS ROUTINE  PERFORMS  NO  ERROR  CHECKING.
00057                 *
00058                 ************************************************************
```

Figure E-6. Programming the MC68701 EPROM: EPROM

```
PAGE  002  EPROM   .SA:1  EPROM  *** ROUTINE TO PROGRAM THE MC68701 EPROM ***

00060
00061                    * E Q U A T E S
00062
00063           0008  A TCSR   EQU    $08      TIMER CONTROL/STAT REGISTER
00064           0009  A TIMER  EQU    $09      COUNTER REGISTER
00065           000B  A OUTCMP EQU    $0B      OUTPUT COMPARE REGISTER
00066           0014  A EPMCNT EQU    $14      RAM/EPROM CONTROL REGISTER
00067
00068                    * L O C A L    V A R I A B L E S
00069
00070A 0080                      ORG    $80
00071A 0080     0002  A IMBEG   RMB    2        START OF MEMORY BLOCK
00072A 0082     0002  A IMEND   RMB    2        LAST BYTE OF MEMORY BLOCK
00073A 0084     0002  A PNTR    RMB    2        FIRST BYTE OF EPROM TO BE PGM'D
00074A 0086     0002  A WAIT    RMB    2        COUNTER VALUE
00075
00076                    * E P R O M    S T A R T S    H E R E
00077
00078A 3000                      ORG    $3000
00079A 3000 DE 84     A EPROM   LDX    PNTR     SAVE CALLING ARGUMENT
00080A 3002 3C                   PSHX            RESTORE WHEN DONE
00081A 3003 DE 80     A          LDX    IMBEG    USE STACK
00082
00083A 3005 3C          EPR002   PSHX            SAVE POINTER ON STACK
00084A 3006 86 FE     A          LDAA   #$FE     REMOVE VPP, SET LATCH
00085A 3008 97 14     A          STAA   EPMCNT   PPC=1, PLC=0
00086A 300A A6 00     A          LDAA   X        MOVE DATA MEMORY-TO-LATCH
00087A 300C DE 84     A          LDX    PNTR     GET WHERE TO PUT IT
00088A 300E A7 00     A          STAA   X        STASH AND LATCH
00089A 3010 08                   INX             NEXT ADDR
00090A 3011 DF 84     A          STX    PNTR     ALL SET FOR NEXT
00091A 3013 86 FC     A          LDAA   #$FC     ENABLE EPROM POWER (VPP)
00092A 3015 97 14     A          STAA   EPMCNT   PPC=0, PLC=0
00093
00094                    * NOW WAIT FOR 50 MSEC TIMEOUT USING OUTPUT COMPARE.
00095
00096A 3017 DC 86     A          LDD    WAIT     GET CYCLE COUNTER
00097A 3019 D3 09     A          ADDD   TIMER    BUMP CURRENT VALUE
00098A 301B 7F 0008   A          CLR    TCSR     CLEAR OCF
00099A 301E DD 0B     A          STD    OUTCMP   SET OUTPUT COMPARE
00100A 3020 86 40     A          LDAA   #$40     NOW WAIT FOR OCF
00101
00102A 3022 95 08     A EPR004  BITA   TCSR
00103A 3024 27 FC 3022          BEQ    EPR004   NOT YET
00104A 3026 38                  PULX            SETUP FOR NEXT ONE
00105A 3027 08                  INX             NEXT
00106A 3028 9C 82     A         CPX    IMEND    MAYBE DONE
00107A 302A 23 D9 3005          BLS    EPR002   NOT YET
00108A 302C 86 FF     A         LDAA   #$FF     REMOVE VPP, INHIBIT LATCH
00109A 302E 97 14     A         STAA   EPMCNT   EPROM CAN NOW BE READ
00110A 3030 38                  PULX            RESTORE PNTR
00111A 3031 DF 84     A         STX    PNTR
00112A 3033 39                  RTS             THAT'S ALL
00113                           END
TOTAL ERRORS 00000--00000
```