MC6840UM (AD-1)

Programmable Timer

Fundamentals and Applications

MC6840

MOTOROLA PERIPHERALS

Doing more so your processor can do more

MC6840UM (AD-1)

MOTOROLA INC.

MC6840 PROGRAMMABLE TIMER FUNDAMENTALS AND APPLICATIONS

Prepared by
NMOS Microcomputer Systems Applications
Austin, Texas

Motorola reserves the right to make changes to any products herein to improve reliability, function or design. Although the information in this document has been carefully reviewed for broad application, Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others.

Second Edition

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 2

CHAPTER 4

CHAPTER 5

LIST OF TABLES

CHAPTER 2

CHAPTER 3

CHAPTER 1
INTRODUCTION

1.0 INTRODUCTION TO THE MC6840 PROGRAMMABLE TIMER MODULE (PTM)

The purpose of using the MC6840 PTM is to allow the MPU and its associated RAM to be free of the timing function task. The three timers in the PTM will not bother the MPU until they have a result of their pre-programmed task. This is especially valuable when long Time Outs are involved.

The MC6840 is a totally compatible member of the M6800 family of peripheral parts. The eight write only and seven read only registers of the PTM appear as eight memory locations to the MPU. The following chapters contain complete information in interfacing the PTM to the M6800 MPU family, interfacing the PTM to a non-Motorola MPU, software initialization, applications, and ordering information.

If the user is not familiar with the M6800 instructions used throughout this manual, refer to the M6800 Microprocessor Instruction Set Summary (Programming Card). This card is available through your local Motorola Sales Office.

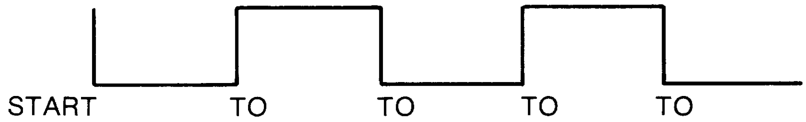
When refering to an MPU address or data in this manual, hexadecimal numbers are used exclusively and are shown as $XXXX. Earlier MC6840 data sheets referred to the Period Measurement Mode as the Frequency Comparison Mode, and the Pulse Width Measurement Mode as the Pulse Width Comparison Mode. The electrical function of these modes has not been altered. The mode names were changed in order to more accurately reflect the operation of the timers.

The PTM consists of three independent timers each capable of four basic modes of operation. Each mode, in turn, has several programmable variations that may be used to suit an individual application.

1.1 THE FOUR BASIC MODES OF OPERATION

1.1.1 Continuous Mode

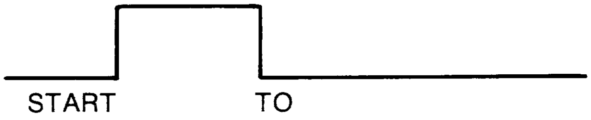
In this mode the timer counts down its initial programmed timer data word (16 bits) to zero. This is called a Time Out (TO) and may generate an interrupt if it is so programmed. The continuous output is:



The output is low for the first Time Out period, and changes state for each subsequent Time Out period. This produces a 50% duty cycle square twice the duration of the Time Out.

1.1.2 Single-Shot Mode

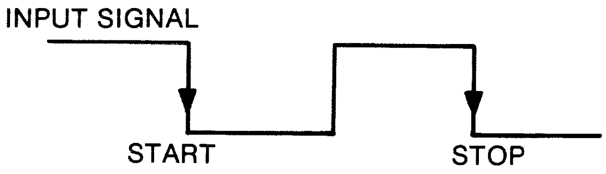
The Single-Shot Mode is similar to the continuous mode except that the output produces only one high pulse for the Time Out period.



In this mode a timer operates as a programmable, **retriggerable** one shot.

1.1.3 Period Measurement Mode

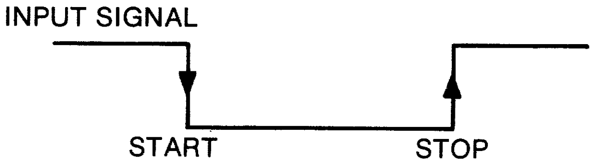
One operation performed in this mode is to measure the period (reciprocal of the frequency) of a given digital signal:



When the pulse goes low the first time, the timer starts counting down from its programmed number. When the signal goes low again, the timer stops counting down and generates an interrupt. The MPU can now determine the frequency easily (1/p = f).

1.1.4 Pulse Width Measurement Mode

The width of a given pulse may be determined by using this mode. The "down-time" of a digital signal can be measured in the same manner as in the Period Measurement Mode:



In this mode the timer starts decrementing when the input pulse goes low and stops decrementing and generates an interrupt when the input pulse returns high.

There are many variations of operation using the four previously described modes. The three timers are independent, and each may be:
1. Cascaded with another timer.
2. Operated in any of three basic modes.
3. Programmed to use the internal clock or an external clock.
The initialization chapter fully explains all the points available to the user.

CHAPTER 2
INTERFACING THE PTM TO THE MPU

2.0 INTERFACING THE PTM TO THE MC6800 OR MC6802

The Programmable Timer (MC6840), as with all Motorola 6800 family peripherals, interfaces easily to the Motorola MPU line. Figure 2-1 shows the PTM interfaced to either an MC6802 or MC6800. The base PTM address has been completely decoded for *$5000. The two hex inverters and a thirteen input NAND gate LS devices decode the address and minimize the loading on the address lines. Any valid base address desired from *$0010 through *$FFF8 may be completely decoded with a different configuration of the same hardware. If complete address decoding is not necessary, as in a minimum parts count system, two packs may be eliminated. This must be done with care to avoid addressing conflicts.

The three least significant bits of the 16-bit address lines are connected directly to the three register select (RSX) inputs. These three least significant bits, in conjunction with the state of the Read/Write line, select the seven Read Only and eight Write Only registers accessible to the MPU.

Figure 2-2 shows the PTM interfaced to the MEK6800D2 kit. It is identical to Figure 2-1 except for the two extra gates (available on the kit) which are used to disable the data bus extenders. Only addresses *$5000 thorugh *$5007 are disabled for kit expansion use. (See Motorola Application Note 771 — Expanding the MEK6800D2 kit).

Figure 2-3 shows an additional manual patching area which may be built up in the MEK6800D2 wire wrap area. This has proven to be a valuable design and learning tool for the PTM. Software may be quickly implemented and any timer output configurations may be quickly implemented in hardware and tested. Each timer output is buffered to drive an L.E.D. so that the output state may be observed.

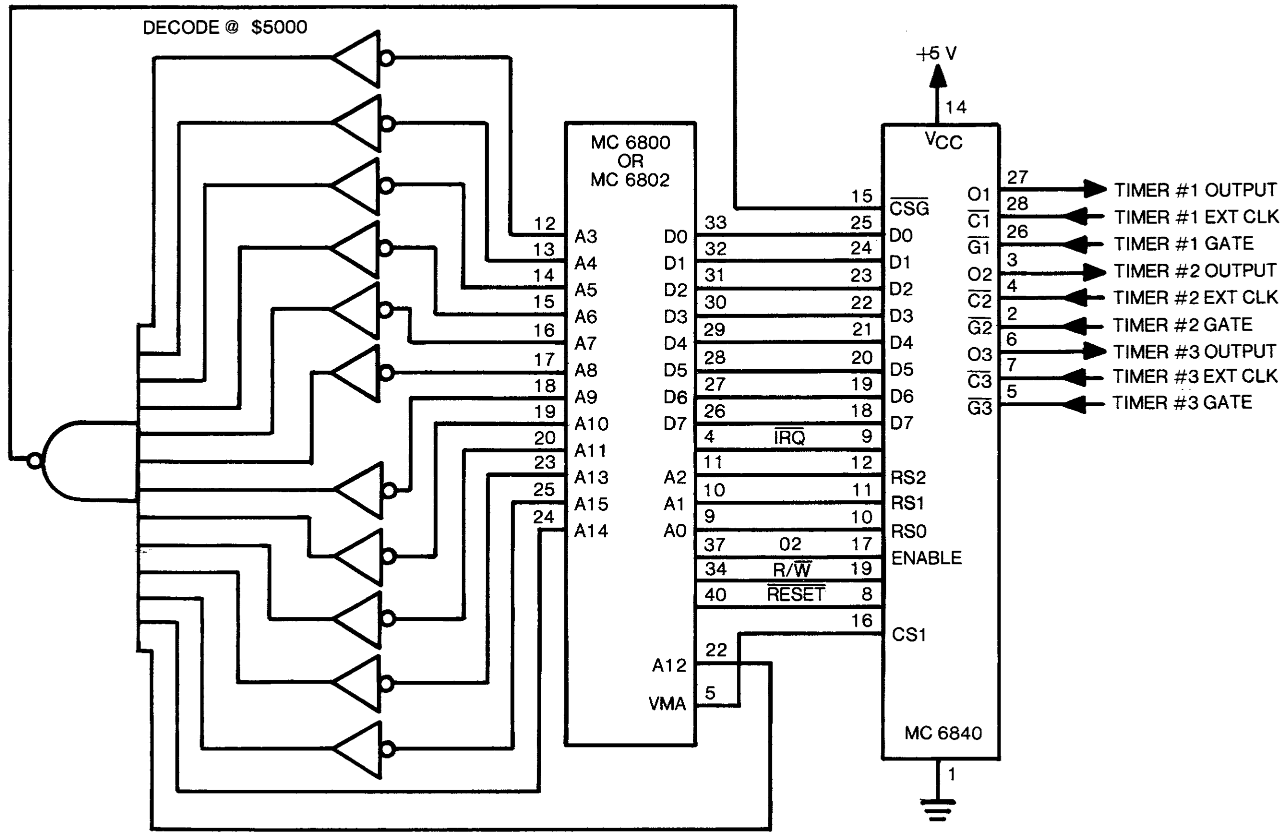


Figure 2-1 PTM Interfaced to the MPU

* hexadecimal number

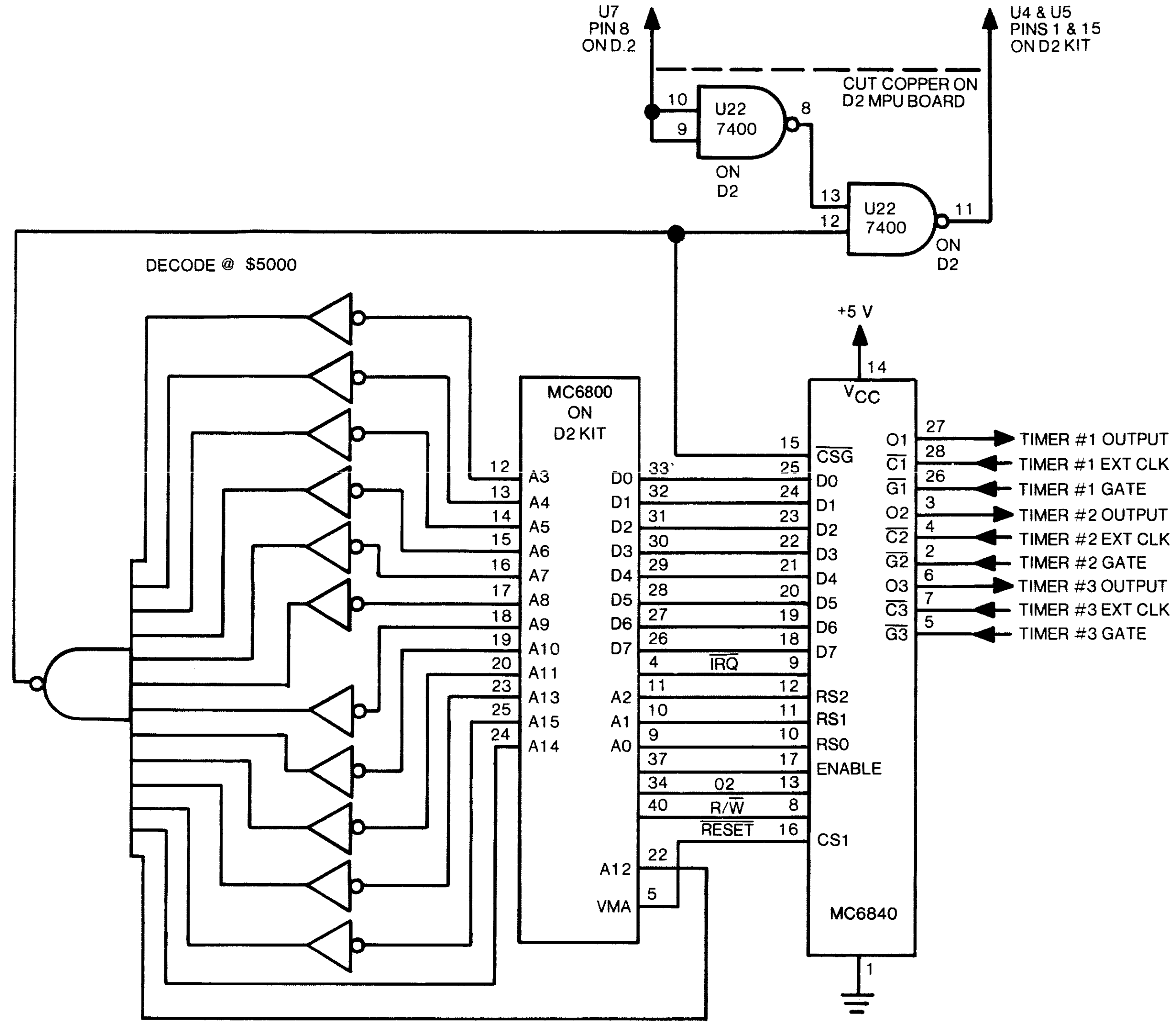


Figure 2-2 PTM Interfaced to MEK6800D2

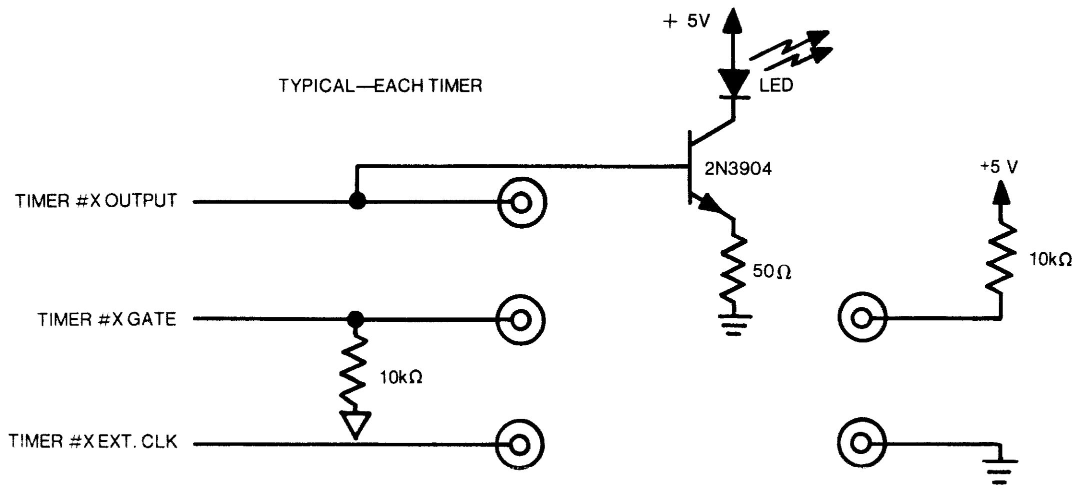


Figure 2-3 MEK6800D2 Patch Area Build-up

In high frequency output applications, the L.E.D. will appear partially on if the high state duty cycle is greater than 20%. The hardware patch area permits cascading timers, external clocking and gating, and interfacing the results to additional hardware.

The MC6840, when interfaced to the M6800 family of parts, provides the user with an intelligent peripheral that is not software intensive. Once the PTM has been conditioned by the MPU, further MPU time is not required in most applications until the PTM generates an interrupt. This can be especially valuable in industrial controls where the timer controls several interval dependent production functions. During the timers intervals, the MPU can be accomplishing other tasks.

2.1 INTERFACING THE MC6840 PTM TO A NON-MOTOROLA MPU

The user of a non-Motorola MPU who needs the superior capabilities of the MC68X40 PTM can be accommodated with minimal difficulty. This section deals with interfacing the Intel 8085 to the MC68A40 (1.5 MHz) PTM. MPU's similar to the Intel 8085 will interface to the Motorola PTM in a like manner, but care must be exercised in order to insure proper performance.

The user should have the MC6840 PTM Data Sheet, the Intel MCS-85 Users Manual and the Motorola M6800 MPU/Memory Data book on hand to ensure the proper use of both parts.

It should be noted that a 1.0 MHz PTM must not be used. Only the faster 1.5 MHz MC68A40 can be interfaced to the 8085. The method selected to provide for an E timing signal ($\overline{WR}$ + $\overline{RD}$ = E) will result in an E pulse with a 400 ns pulse width. The MC6840 PTM requires a pulse width of 480 ns minimum. Also, some setup times given by the 8085 exclude the use of the MC6840 PTM.

2.2 8085 INTERFACE

The hardware design is given in Figure 2-4 with the Motorola MC68A40 PTM shown connected to an 8085. Each signal input/output pin on the MC68A40 that must be connected to an 8085 pin is explained in succeeding paragraphs in the section titled, Derivation of the MC68A40 Signals.

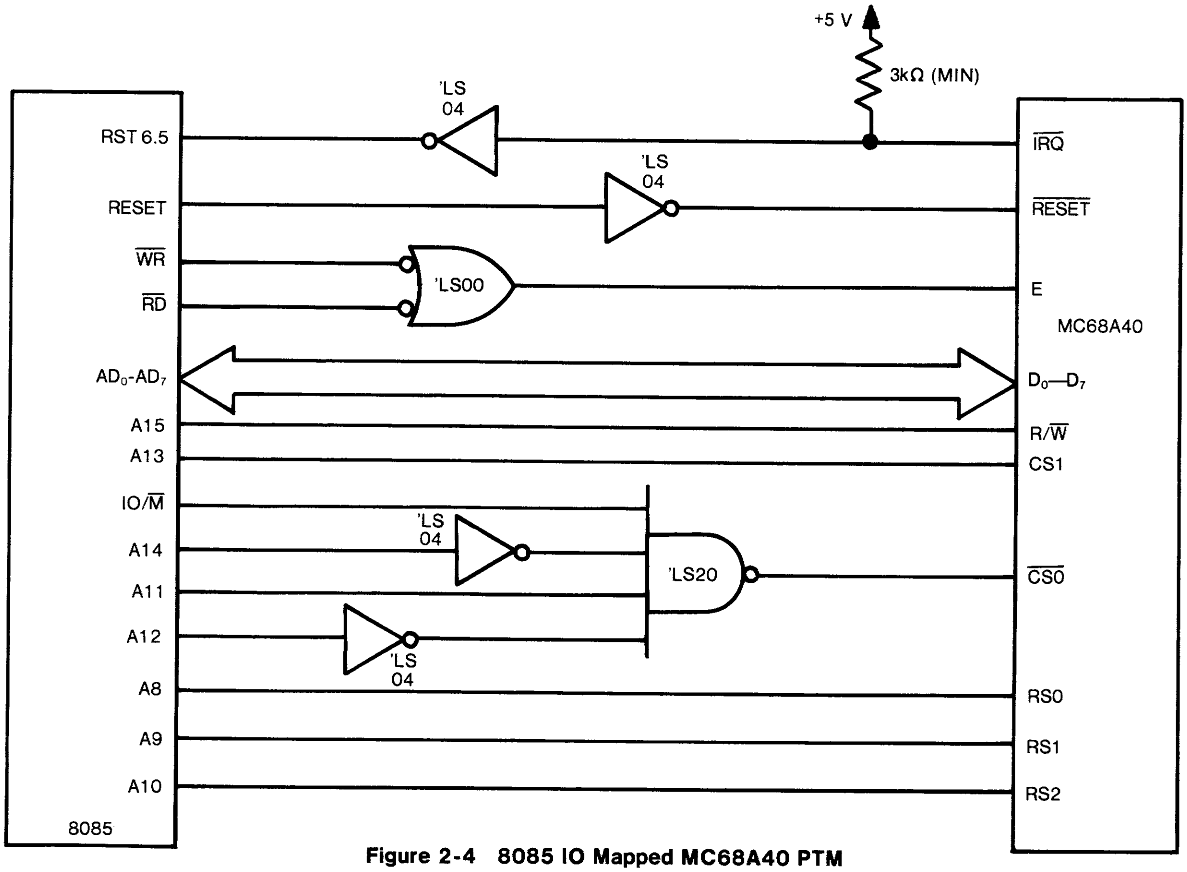


Figure 2-4 8085 IO Mapped MC68A40 PTM

The interface technique selected makes use of I/O mapped I/O. That is to say, the MC68A40 is addressed only by use of the INPUT or OUTPUT instructions in the 8085. Using the fact that the 8085 copies the I/O address (0-255) on to A8-A15 as well as A0-A7 during an INPUT or OUTPUT instruction, external latches are not required to store the I/O address. The address is valid on A8-A15 for a read or write cycle within an 8085 I/O instruction.

Using the $\overline{\text{WR}}$ and $\overline{\text{RD}}$ signals to produce the Enable (E) signal will result in an E pulse width of 400 ns. M6840 (1.0 MHz) parts need an E pulse width of 480 ns, minimum. MC68A40 (1.5 MHz) and MC68B40 (2.0 MHz) family parts have an E pulse width minimum down to 280 ns and 220 ns respectively.

Also note in Figure 2-4 that address bit 15 is used to provide the MC68A40 with its R/W signal. This was done to satisfy the timing requirements on the R/$\overline{\text{W}}$ signal to the MC68A40 PTM. The use of A15 as a R/$\overline{\text{W}}$ signal does limit the total number of I/O devices since there are now only 7 bits of address available for chip select decoding. This still leaves 128 device addresses that can be used for I/O, and for most requirements this number is more than sufficient. (NOTE: This means 128 device addresses and not 128 I/O lines. A device address can support many I/O lines). A15 was arbitrarily chosen to be used as R/$\overline{\text{W}}$. Any address bit within A15-A11 could have been used with the same results.

The use of $\overline{\text{RD}}$ logically ORed with $\overline{\text{WR}}$ to generate an Enable (E) pulse assumes that using the MC68A40 PTM, a maximum pulse width of 24 microseconds on the E signal must be observed. Wait and Halt states associated with 8085 instructions are much less than 25 microseconds during a read or write operation. There is also guaranteed to be several $\overline{\text{WR}}$ or $\overline{\text{RD}}$ signals in any instruction (each instruction must have one or more opcode fetches) so that the free running requirement on E is met. Since this E pulse does change frequency during operation the internal clock function will not have an accurate "real time" function.

2.3 MC68A40 SIGNAL DERIVATION

Register Selects (RS0, RS1, RS2) — These three input pins control the selection of the control registers, counters, latches, or status register within the MC68A40. In Figure 2-4; A8, A9 and A10 are shown driving these input pins.

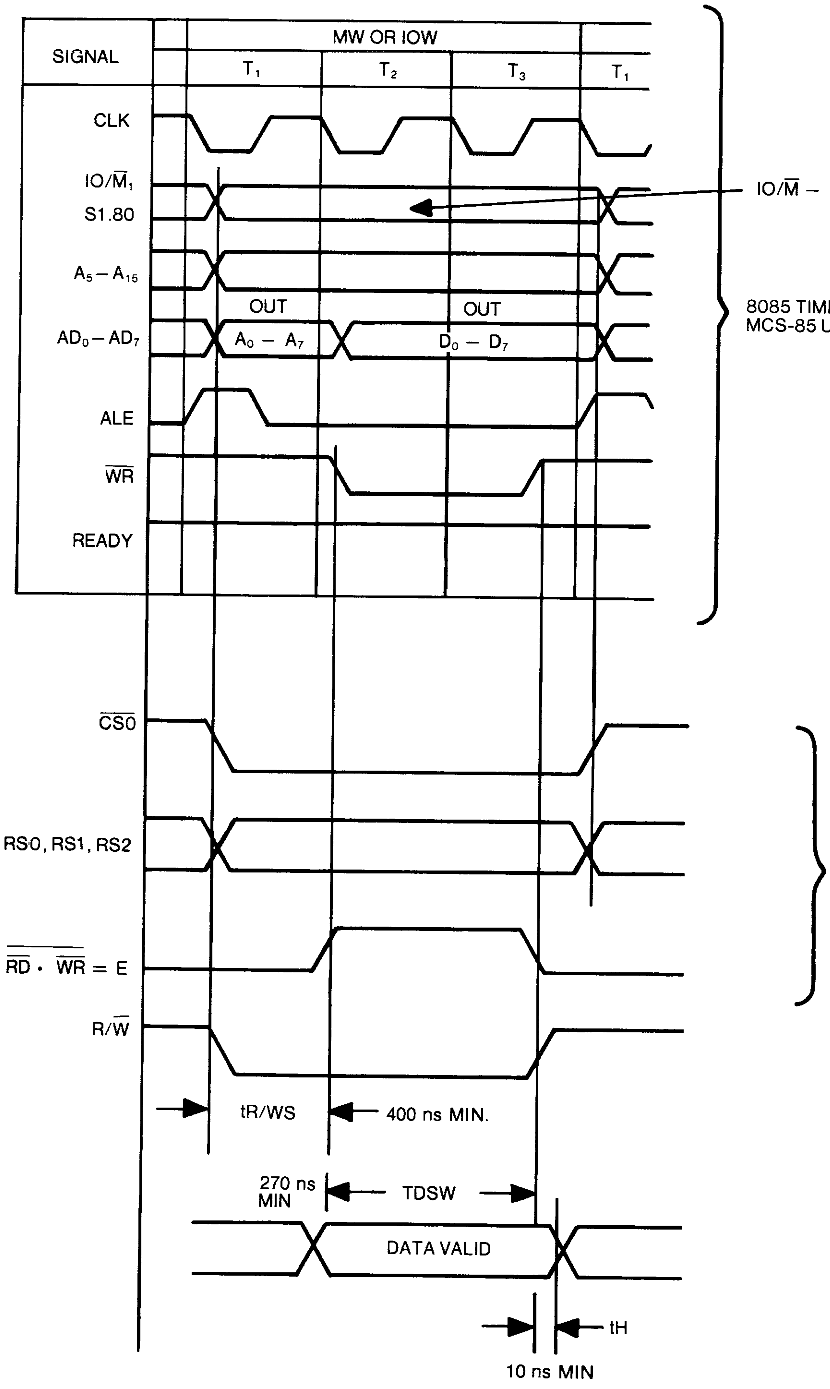
2.3.1 Chip Selects (CS1), (CS0) and R/$\overline{\text{W}}$

The $\overline{\text{CS0}}$ and CS1 inputs are decoded by the desired combination of 8085 address bits A11-A14. These four bits in conjunction with the register select bits, A8, A9 and A10, result in seven address bits to specify an I/O port address. The 8085 IO/M pin is also included in the address decode as the address bus will contain unwanted information during a memory read or write cycle.

As previously mentioned, address bit A15 is used as the R/$\overline{\text{W}}$ input to the MC68A40. As shown in the write timing diagram of Figure 2-5, R/$\overline{\text{W}}$ must be low for a minimum of 140 nanoseconds prior to the rising edge of the E pulse. Use of A15 will provide a low R/$\overline{\text{W}}$ line at least 270 nanoseconds before the rising edge of E. A consequence of using A15 as the R/$\overline{\text{W}}$ signal is that there must be separate addresses defined for read and write operations in software. The following example for MC68A40 PTM should be helpful.

Example 1: The 68A40 PTM eight write addresses and seven read addresses. RS0, RS1, RS2, A15 will select which addresses are to be accessed according to Table 2-1.

Now arbitrarily assigning values to A14-A11 of 0101 produces the complete address as shown in the column labeled as such in Table 2-1. In software, these addresses must be defined by using pseudo instructions (EQU):



IO/M̄ – O(MW) OR 1 (IOW), S1-0, S0-1

8085 TIMING FROM THE MCS-85 USERS MANUAL.

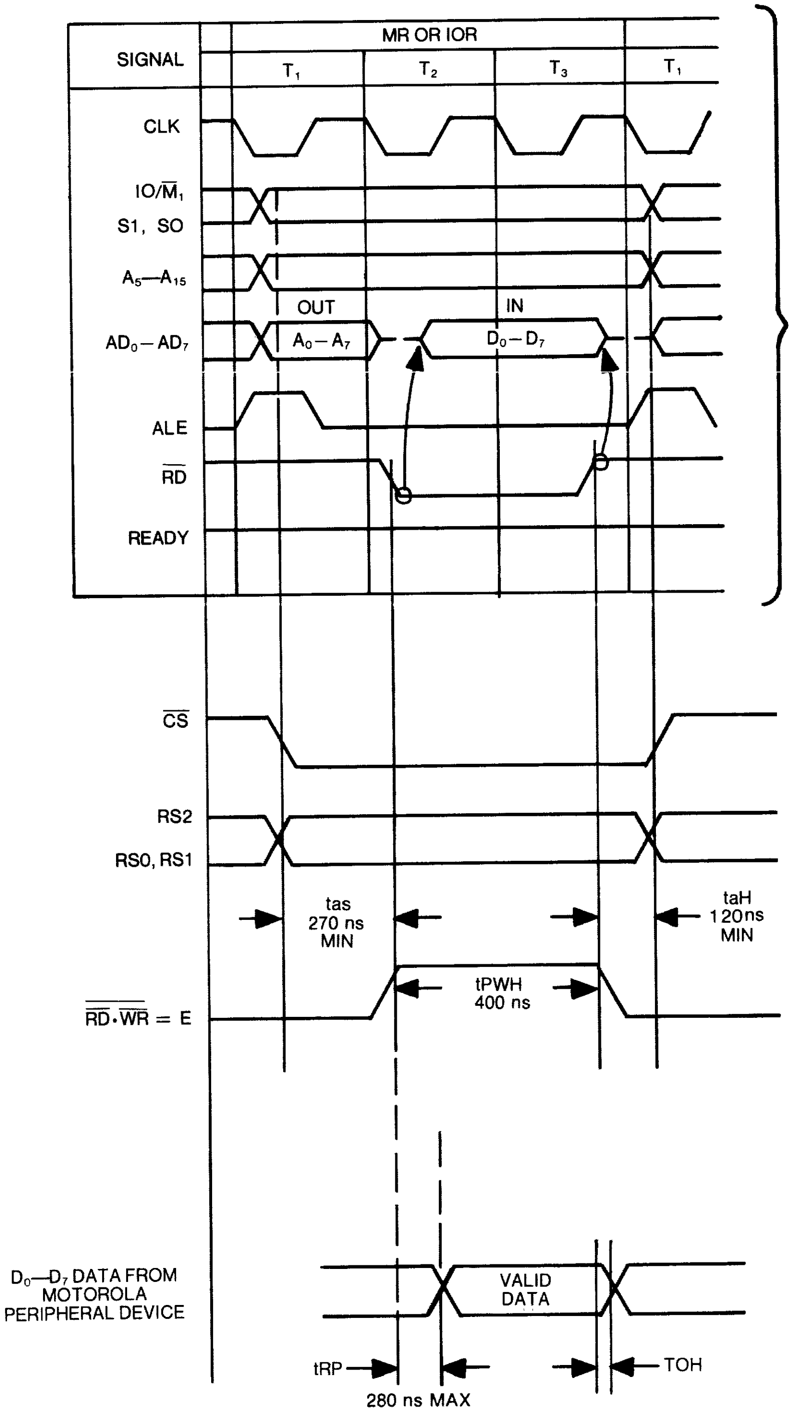
THIS TIMING IS IDENTICAL TO THAT SHOWN FOR READ I/O AS SHOWN IN FIGURE 2

tR/WS MUST BE 140 ns FOR THE PERIPHERAL DEVICE, THE 8085 PROVIDES 270 ns MINIMUM

tDSW MUST BE 80 ns FOR MOTOR PERIPHERAL, 8085 PROVIDES 400ns MINIMUM.

tH MUST BE 10 ns MINIMUM FOR MOTOROLA PERIPHERAL - 8085 PROVIDES 80 ns MINIMUM.

Figure 2-5 8085/MC68A40 I/O Write Timing



IO/$\overline{M}$-P (MR) OR 1 (OR), S1-1.50-0

8085 TIMING FROM THE
MCS-85 USER'S MANUAL

tas MUST BE 140 ns MINIMUM FOR THE PERIPHERAL DEVICE THE 8085 PROVIDES 120 ns MINIMUM.

tPWH MUST BE 280 ns MINIMUM. THE 8085 PROVIDES 400 ns MINIMUM.

taH MUST BE 10 ns MINIMUM. THE 8085 PROVIDES 120 ns MINIMUM.

tBD MUST BE 280 ns MAX. THE PERIPHERAL WILL PROVIDE 220 ns MAX.

TOH CAN BE ZERO, THE PERIPHERAL PROVIDES 10 ns MINIMUM.

Figure 2-5 8085/MC68A4 I/O Read Timing (continued)

					COMPLETE ADDRESS								HEXIDECIMAL
	RS2 = A10	RS1 = A9	RS0 = A8	R/W = A15	A15	A14	A13	A12	A11	A10	A9	A8	NOTATION
Unused	0	0	0	1	1	0	1	0	1	0	0	0	A8
Write Reg. 1 or Reg. 3	0	0	0	0	0	0	1	0	1	0	0	0	28
Read Status Register	0	0	1	1	1	0	1	0	1	0	0	1	A9
Write Register 2	0	0	1	0	0	0	1	0	1	0	0	1	29
Read Timer #1 Counter	0	1	0	1	1	0	1	0	1	0	1	0	AA
Write MSB Buffer Reg.	0	1	0	0	0	0	1	0	1	0	1	0	2A
Read LSB Buffer Reg.	0	1	1	1	1	0	1	0	1	0	1	1	AB
Write Timer #1 Latches	0	1	1	0	0	0	1	0	1	0	1	1	2B
Read Timer #2 Counter	1	0	0	1	1	0	1	0	1	1	0	0	AC
Write MSB Buffer Reg.	1	0	0	0	0	0	1	0	1	1	0	0	2C
Read LSB Buffer Reg.	1	0	1	1	1	0	1	0	1	1	0	1	AD
Write Timer #2 Latches	1	0	1	0	0	0	1	0	1	1	0	1	2D
Read Timer #3 Counter	1	1	0	1	1	0	1	0	1	1	1	0	AE
Write MSB Buffer Reg.	1	1	0	0	0	0	1	0	1	1	1	0	2E
Read LSB Buffer Reg.	1	1	1	1	1	0	1	0	1	1	1	1	AF
Write Timer #3 Latches	1	1	1	0	0	0	1	0	1	1	1	1	2F

Table 2-1 8085/MC68A40 Addressing

LABEL FIELD	OP CODE FIELD	OPERAND FIELD	COMMENT FIELD
RSTS	EQU	$A9	READ STATUS REG
RCN1	EQU	$AA	READ TIMER ONE
RLB1	EQU	$AB	READ LSB BUFFER
RCN2	EQU	$AC	READ TIMER TWO
RLB2	EQU	$AD	READ LSB BUFFER
RCN3	EQU	$AE	READ TIMER THREE
RLB3	EQU	$AF	READ LSB BUFFER
WR13	EQU	$28	WRITE REG. 1 OR 3
WRG2	EQU	$29	WRITE REG. 2
WMB1	EQU	$2A	WRITE MSB BUFFER
WLT1	EQU	$2B	WRITE TIMER 1 LATCHES
WMB2	EQU	$2C	WRITE MSB BUFFER
WLT2	EQU	$2D	WRITE TIMER 2 LATCHES
WMB3	EQU	$2E	WRITE MSB BUFFER
WLT3	EQU	$2F	WRITE TIMER 3 LATCHES

Table 2-2 Pseudo Instructions

The pseudo instructions (Table 2-2) are easy to use, but care should be exercised to avoid confusing the 8085 IN and OUT instructions when reading from or writing to the MC68A40 PTM. OUT is only used with a write. In order to avoid a mismatch, the first character in the label field of the pseudo instruction tells the user if it is a read or write instruction:

IN RXXX (READ INSTRUCTION)
OUT WXXX (WRITE INSTRUCTION)

Further, when writing to a timer latch or reading from a timer counter (this is a 16-bit wide word) two consecutive pseudo instructions must be used in the order shown:

TO WRITE TO A TIMER:
OUT WMBX (WRITE MSB BUFFER)
OUT WLTX (WRITE TIMER X LATCHES)

TO READ OUT A TIMER:
IN RCNX (READ TIMER X)
IN RLBX (READ LSB BUFFER X)

To read the MC68A40 status register:

IN RSTS (READ STATUS REGISTER)

To write a word to control register 2 (write control register 2), an output instruction must be used: WRG2 OUT.

In the IN instruction, A15 will be high (R/$\overline{\text{W}}$ = 1) for the read operation, while in the OUT instruction, A15 will be low (R/$\overline{\text{W}}$ = 0) for the write operation. In both cases, the MC68A40 chip selects input will become active for A14 - A11 = 0101.

This technique costs nothing in firmware as EQU statements are pseudo instructions and are used in the assembly process only.

2.3.2 Enable (E)

As pointed out earlier, the E pulse input to the MC68A40 is the logical OR of the 8085 outputs $\overline{\text{WR}}$ and $\overline{\text{RD}}$ (the logical OR here is done with a NAND gate performing a Low Level OR on $\overline{\text{WR}}$ and $\overline{\text{RD}}$). Referring to the timing diagrams of Figure 2-5, this will produce an E pulse with proper timing with relation to the address, data and R/$\overline{\text{W}}$ signals.

2.3.3 RESET (R)

This active low MC68A40 input is the inverse of the 8085 reset output.

2.3.4 $\overline{\text{IRQ}}$

This active low output from the MC68A40 PTM must be inverted to drive a RESTART INTERRUPT INPUT such as RST 6.5 on an 8085. The $\overline{\text{IRQ}}$ output is an open drain and should be pulled up to +5 volts.

The generation of an interrupt signal by the MC68A40 PTM should cause the 8085 to immediately branch to a service routine. At the end of the routine, a RET instruction should transfer control back to the original program. For this reason, the RESTART INTERRUPTS (RST 5.5, RST 6.5, RST 7.5, and TRAP) should be used and not the INTERRUPT REQUEST (INTR). The INTR input simply inhibits the Program Counter from incrementing and the INTERRUPT ACKNOWLEDGE signal ($\overline{\text{INTA}}$) is given. At this time, it is the responsibility of the peripheral device to insert a CALL or RESTART instruction to jump to the service routine. This routine must also be responsible for preserving the return address located in the Program Counter. The MC68A40 PTM is not able to do such lengthy and complicated procedures so it must use one of the four RESTART INTERRTUPTS. An interrupt signal on any of these inputs causes the immediate execution of a RESTART instruction in which the contents of the Program Counter are placed on the Stack and a branch is executed to a RESTART address. The addresses for each of the interrupts are:

TRAP — 24 (Hex)
RST 5.5 — 2C
RST 6.5 — 34
RST 7.5 — 3C

These locations should contain the JUMP instruction to the service routine. At the end of ths routine a RET instruction will branch back to the original program. RST 5.5, RST 6.5, RST 7.5 are maskable interrupts of increasing priority (RST 5.5 lowest). TRAP is non-maskable and has the highest priority of all. RST 5.5 and RST 6.5 are high level-sensitive and RST 7.5 is rising edge-sensitive. All are sampled on the last clock cycle of each instruction.

2.3.5 Read/$\overline{\text{Write}}$ Timing

The read and write timings are shown in Figure 2-5. Below each are the necessary inputs to a MC68A40 PTM generated from the 8085 outputs.

2.3.6 Summary

The interface between a Motorola MC68A40 PTM and the Intel 8085 microprocessors can be done with as little as two gates and two inverters.

When the technique of I/O mapped I/O is used, the IN/OUT instructions of the 8085 will result in the I/O address being copied onto the A15-A8 address bus of the 8085 for most of an I/O operation cycle. Therefore no external latches are required for address latching as would be the case if the I/O address were present on the multiplexed data/address bus AD0-AD7.

Because the MC68A40 PTM requires an E pulse width of 480 nanoseconds maximum and since the generation of E is done by logically ORing two 400 nanosecond pulses ($\overline{\text{RD}}$ and $\overline{\text{WR}}$), only 1.5 MHz and 2.0 MHz deviced (MC68A/B40) should be used.

Generation of the $\overline{\text{CS0}}$ and CS1 signals to MC68A40 PTM is done by decoding 8085 address bits A14-A11 while a R/$\overline{\text{W}}$ signal is obtained from A15 directly. This means that separate write and read addresses must be defined in software. The IO/$\overline{\text{M}}$ signal output from the 8085 is used in the A14-A11 chip decoder.

The multiplexed data/address bus AD0-AD7 of the 8085 is connected directly to the data port D0-D7 of the MC68A40 PTM.

For systems requiring several peripheral devices, the hardware desing shown here will need to be expanded to include appropriate drivers.

Since the "E" input pulse may be used as an internal clock for any of the timers, it must be remembered that this line will vary in frequency. Precise real-time period measurements cannot be made using the internal clock. Accurate measurements can be made using the external clock, but care must be used to insure conformity to the PWL and PWH timing limits in the data sheet section.

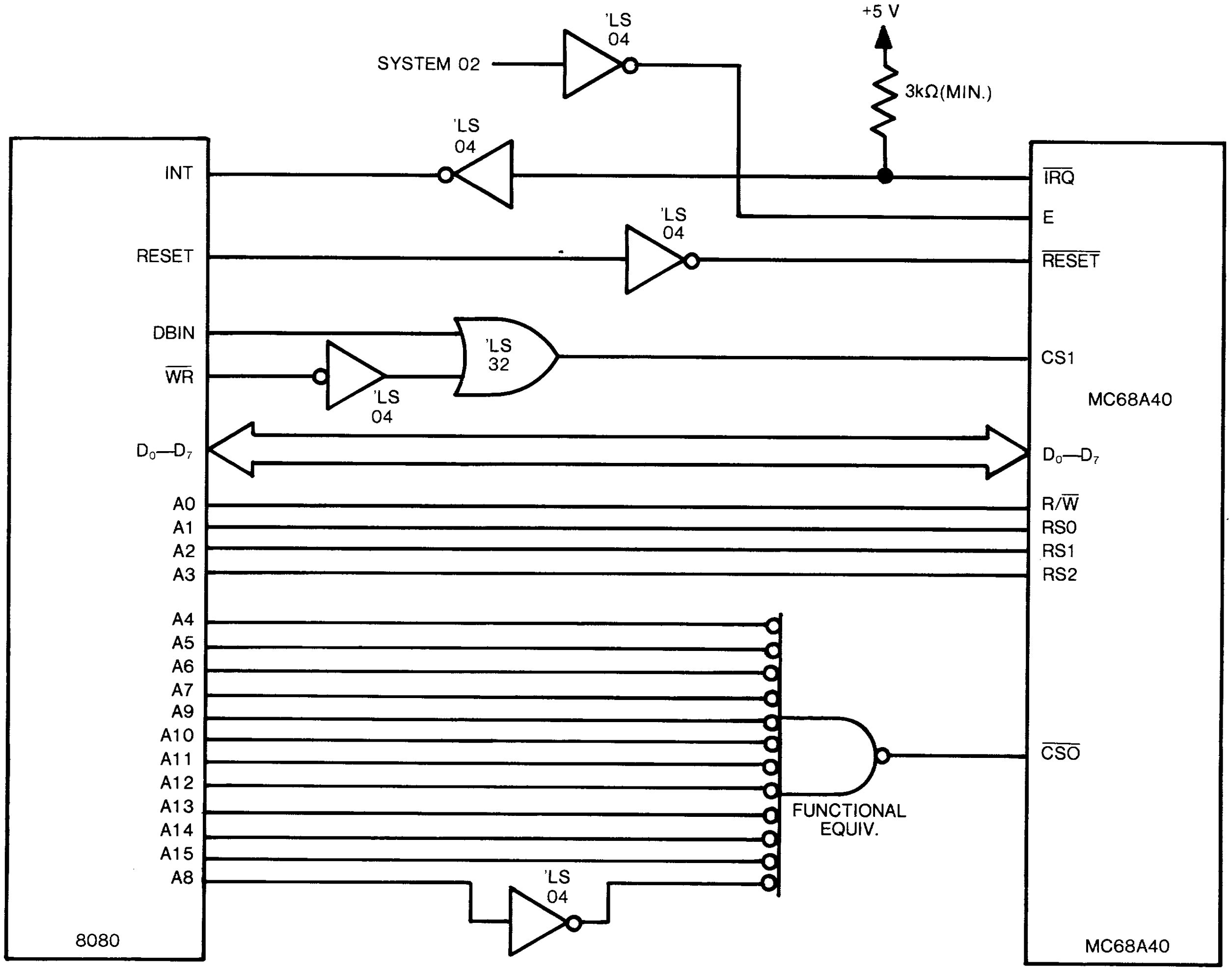
The MC68A40 may also be connected as a memory mapped location, as shown in Figure 2-6. This approach uses the 8212 latch to separate the address and data lines for the PTM. Address line Ao is used in this case for the Read/$\overline{\text{Write}}$ line. Ao was selected for use as the R/W line in this case as a matter of convenience. This permits all odd addresses to be "Read" addresses and all even addresses to be "Write" addresses. The timing requirements are essentially the same as shown in Figure 2-5.

2.4 8080 INTERFACE

The MC6840 may be interfaced to the 8080 in a similar manner to that shown in the preceding chapter. Figure 2-7 shows the basic approach. Care must be exercised to ensure the φ2 clock input frequency does not exceed the specified maximum:

MC6840 — 1.0 MHz
MC68A40 — 1.5 MHz
MC68B40 — 2.0 MHz

The incoming φ2 signal may be divided by a flip-flop, but the proper timing relationships must be maintained.



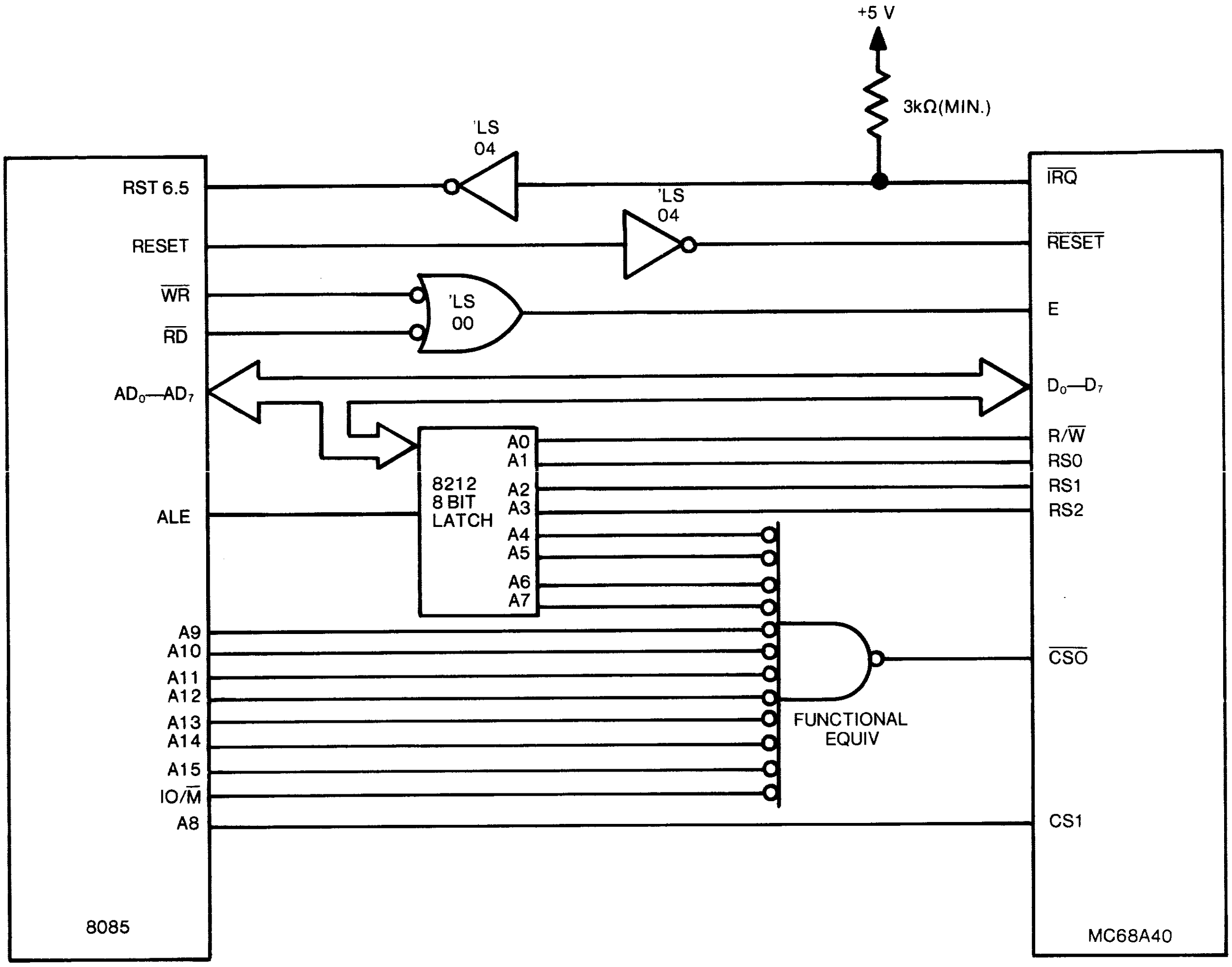
BASE ADDRESS—$0100

$ ADDR	RS2	RS1	RS0	$R/\overline{W} = 0$

0100	0	0	0	CR20 = 0 Write Control Register # 3
				CR20 = 1 Write Control Register #1

0102	0	0	1	Write Control Register #2
0104	0	0	1	Write MSB Buffer Register
0106	0	1	1	Write Timer #1 Latches
0108	1	0	0	Write MSB Buffer Register
010A	1	0	1	Write Timer #2 Latches
010C	1	1	0	Write MSB Buffer Register
101E	1	1	1	Write Timer #3 Latches
				$R/\overline{W} = 1$
0101	0	0	0	No Operation
0103	0	0	1	Read Status Register
0105	0	1	0	Read Timer #1 Counter
0107	0	1	1	Read LSB Buffer Register
0109	1	0	0	Read Timer #2 Counter
010B	1	0	1	Read LSB Buffer Register
010D	1	1	0	Read Timer #3 Counter
010F	1	1	1	Read LSB Buffer Register

Figure 2-6 8085 Memory Mapped MC68A40 PTM



BASE ADDRESS—$0100

$ ADDR	RS2	RS1	RS0	$R/\overline{W} = 0$
0100	0	0	0	CR20 = 0 Write Control Register # 3
				CR20 = 1 Write Control Register #1
0102	0	0	1	Write Control Register #2
0104	0	0	1	Write MSB Buffer Register
0106	0	1	1	Write Timer #1 Latches
0108	1	0	0	Write MSB Buffer Register
010A	1	0	1	Write Timer #2 Latches
010C	1	1	0	Write MSB Buffer Register
101E	1	1	1	Write Timer #3 Latches
				$R/\overline{W} = 1$
0101	0	0	0	No Operation
0103	0	0	1	Read Status Register
0105	0	1	0	Read Timer #1 Counter
0107	0	1	1	Read LSB Buffer Register
0109	1	0	0	Read Timer #2 Counter
010B	1	0	1	Read LSB Buffer Register
010D	1	1	0	Read Timer #3 Counter
010F	1	1	1	Read LSB Buffer Register

Figure 2-7 8080 Memory Mapped MC68A40 PTM

CHAPTER 3
SOFTWARE INITIALIZATION

3.0 INTRODUCTION

Programming the PTM requires the user to have an understanding of the registers. The versatility of this part requires that each timer have its own 8-bit control register in order to select the modes and timer characteristics needed by the user. These registers will be thoroughly examined in this chapter along with the Interrupt Status Register and the registers used to read and write data from and to the counters.

Table 3-1 shows how the registers, latches and counters are selected using the Register Select input pins. Further in this chapter, register selection is discussed as it will probably be connected in most applications (to the three lowest order address lines).

This chapter describes and defines
RESETS
COUNTER INITIALIZATION
COUNTER ENBLE
COUNTER LATCH INITIALIZATION
EXAMPLE COUNTER PROGRAMMING
BIT BY BIT CONTROL REGISTER PROGRAMMING FOR ALL MODES
INTERRUPTS AND THE INTERRUPT STATUS REGISTER

Register Select Inputs			**Operations**	
RS2	**RS1**	**RS0**	**R/W = 0**	**R/W = 1**

0	0	0	CR20 = 0 Write Control Register = 3
			CR20 = 1 Write Control Register = 1
				No Operation

0	0	1	Write Control Register #2	Read Status Register
0	1	0	Write MSB Buffer Register	Read Timer #1 Counter
0	1	1	Write Timer #1 Latches	Read LSB Buffer Register
1	0	0	Write MSB Buffer Register	Read Timer #2 Counter
1	0	1	Write Timer #2 Latches	Read LSB Buffer Register
1	1	0	Write MSB Buffer Register	Read Timer #3 Counter
1	1	1	Write Timer #3 Latches	Read LSB Buffer Register

Table 3-1 Register Selection

3.1 RESETS

3.1.1 Hardware

This pin on the PTM is normally tied to the system $\overline{\text{RESET}}$, and is asserted during power up and whenever the system "Reset" button is pushed. This Reset causes the following conditions:

- All operations (clocks) in progress in the PTM's timers are stopped.
- All interrupt flags in the interrupt Status Register are cleared.
- All Counter latch data is set to hexadecimal FFFF (65,536).
- All Control Register bits are cleared (0) except bit 0 of Control Register #1 which is set (1). This is the software Internal Reset Bit.

3.1.2 Software

The internal software Reset bit is set by a hardware Reset or being written high by the MPU. The software Reset being asserted causes the following conditions.

- All operations (clocks) in progress in the PTM's timers are stopped.
- All interrupt flags in the Interrupt Status Register are cleared.
- All Counter latches retain the data **last written** to them.
- All remaining Control Register bits are **not** changed.

3.2 COUNTER INITIALIZATION

Counter Initialization is caused by various conditions in the different modes. The conditions will be discussed for each mode. Counter initialization causes:

- The data stored in the counter latch will be written to the counter.
- The interrupt associated with the timer being initialized will be cleared.

3.3 COUNTER ENABLE

Counter Enable is the state where conditions allow the counter to operate. One condition is that all resets are always cleared. The other necessary conditions will be discussed in each mode. In order to operate a timer must be Initialized AND Enabled.

3.4 COUNTER LATCH INITIALIZATION

Counter Latch Initialization is the transferal of a 16-bit data word (two 8-bit words) from the MPU to the counter latches. Counter Latch Initialization is fully explained in the following example. The M6800 Instruction Set Summary card may be used to interpret the mnemonics:

3.5 EXAMPLE COUNTER PROGRAMMING

In Table 3-2 it is assumed that the register select lines have been connected to the three least significant address lines:

RS0 to A0
RS1 to A1
RS2 to A2

The MSB buffer register is a single, write only, 8-bit holding register with three separate addresses — $XXX2, $XXX4, $XXX6.

The LSB buffer register is a single, read only 8-bit holding register with three separate addresses — $XXX3, $XXX5, $XXX7.

ADDR	READ	WRITE
$XXX0	NOP	CONTROL REGISTER #1 OR #3
$XXX1	STATUS REGISTER	CONTROL REGISTER #2
$XXX2	TIMER #1 COUNTER	MSB BUFFER REGISTER
$XXX3	LSB BUFFER REGISTER	TIMER #1 LATCHES
$XXX4	TIMER #2 COUNTER	MSB BUFFER REGISTER
$XXX5	LSB BUFFER RGISTER	TIMER #2 LATCHES
$XXX6	TIMER #3 COUNTER	MSB BUFFER REGISTER
$XXX7	LSB BUFFER REGISTER	TIMER #3 LATCHES

Table 3-2 Register Selection As Memory Locations

3.5.1 Writing to a Timer Latch

Writing to a Timer Latch (typical for all three timers):

```
  LDX* $FDEC
```

(*$XXXX indicates an immediate addressing mode of a hexadecimal number; FDEC is used here as example data).

$ - HEXADECIMAL NUMBER

```
STX $XXX2
    (XH→M)          FD →  MSB BUFFER (@ $XXX2)
                                    ↓
    (XL→ M + 1)        EC ↘   MS BYTE OF LATCH
                              LS BYTE OF LATCH (@ $XXX3)
```

When writing to the LS byte of the timer latch (@ memory location $XXX3), the contents of the MSB buffer is internally written to the MS byte of that latch. (16-bit wide word)

3.5.2 Reading from a Timer Counter

```
LDX $XXX2 (location of timer 1)

   (M→XH) FD←MS BYTE OF COUNTER

                  LS BYTE OF COUNTER (@ $XXX2)
                              ↓
   (M + 1→XL) EC←LSB BUFFER (@ $XXX3)
```

When reading a timer counter (16-bits), the MS byte (8-bits) is read at $XXX2 (M) and the LS byte is internally written to the LSB buffer register which may be read at the next memory location — $XXX3 (M + 1).

The STS & STX instructions automatically perform the 16-bit write.

The LDS & LDX instructions automatically perform the 16-bit read.

The contents of the MSB Buffer and LSB Buffer are changed only by two methods:

- Being written over OR
- A hardware Reset occurs

3.6 CONTROL REGISTER PROGRAMMING:

3.6.1 Individual Bits

After initializing the counter latches, the next registers programmed are the control registers. Bits 1 through 7 perform the same function in all three control registers. Bit 0 performs a unique function in each control register. Control register three is located at address $0000 after a $\overline{\text{RESET}}$ or power-up and is usually programmed first. Bit 0 in control register #3 is a clock prescalar and is available only in control register #3. The prescalar divides either the internal or external clock.

CONTROL REGISTER #3

BITS	7	6	5	4	3	2	1	0
	X	X	X	X	X	X	X	↕

0	$\overline{\text{CLK}}$ ÷1	Prescaler
1	$\overline{\text{CLK}}$ ÷8	Prescaler

Bit 0 in control register #2 (@ $0001) determines which control register may be written at location $0000. This bit defaults to zero upon power up and a hardware reset.

CONTROL REGISTER #2

7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	↕

0	REG #3 MAY BE WRITTEN
1	REG #1 MAY BE WRITTEN

When bit zero in control register #2 is a "1", control register #1 may now be written at location $0000. This bit allows all three timers to operate or holds all three timers in a preset condition.

CONTROL REGISTER #1

7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	↕

0	ALL TIMERS OPERATE
1	ALL TIMERS PRESET

The remaining bits in the control registers perform the same operation in each register. The operation of each timer clock is completely independent on the other timers unless they are cascaded. Each timer has its own external clock pin or the internal enable clock may be used.

CONTROL REGISTER #X

7	6	5	4	3	2	1	0
X	X	X	X	X	X	↕	X

0	EXTERNAL CLOCK (CX INPUT)
1	INTERNAL CLOCK (ENABLE)

(X= 1, 2, or 3)

Bit 6 will mask or enable the individual interrupt flag to cause a composite interrupt. The individual interrupt flag is not maskable. (See Status Register section).

CONTROL REGISTER #X

7	6	5	4	3	2	1	0
X	↕	X	X	X	X	X	X

0	INTERRUPT FLAG MASKED ($\overline{\text{IRQ}}$)
1	INTERRUPT FLAG ENABLED ($\overline{\text{IRQ}}$)

Bit 7 simply enables the timer's output on the individual timer's output pin. If the output is masked it will always be electrically "low."

7	6	5	4	3	2	1	0	
↕	X	X	X	X	X	X	X	CONTROL REGISTER #X

0	TIMER OUTPUT MASKED
1	TIMER OUTPUT ENABLED

3.7 CONTINUOUS OPERATING MODE

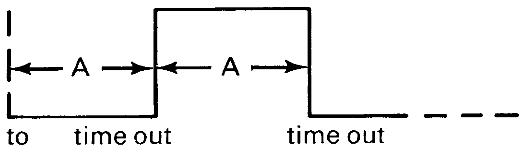
In the examination of the PTMs operating modes the following terms will be used:

- Reset — since a hardware Reset always causes a software Reset (which can also be programmed), and a software Reset cannot be cleared while a hardware Reset is present, the term "Reset" shall refer to the software Reset.
- Interrupt — when an individual timer's interrupt flag is high.
- Write to Counter Latches — Counter Latch Initialization.
- Gate — the individual input control pin that is associated with each timer. This pin's state or transition controls the various timer functions is described below.
- Time Out — occurance one count after the contents of a timer equals $0000.
- "Condition" clear — the named condition is not active.
- Counter Enable (as refers to **all** Continuous Mode conditions):
  Reset clear
  Gate pin is low

3.7.1 16-Bit Operation

16-bit operation is considered the "usual" count down operation of the timers. It is probably the most frequently used. The timer's clock counts down the 16-bit latch data word to zero and then generates an interrupt and an output transition as shown below:

16-bit operation



A = the total 16-bit count in the latch +1, times the period of the clock.

The control register programming for the two cases of Counter Initialization in 16-bit operation is shown below:

BITS	7	6	5	4	3	2	1	0	CONTROL REGISTER #X (X= 1, 2, or 3)
	X	X	0	1	0	0	X	X	

Interrupt:
  Occurs at each Time Out
Counter Initialization:
  Reset OR
  Gate pin goes low

7	6	5	4	3	2	1	0
X	X	0	0	0	0	X	X

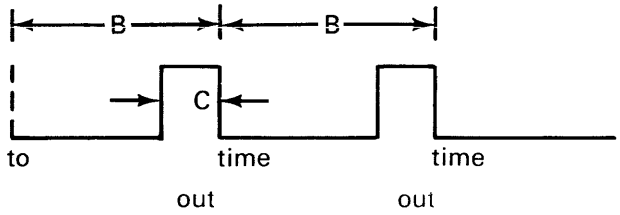
CONTROL REGISTER #X

Interrupt:
  Occurs at each Time Out
Counter Initialization:
  Reset OR
  Gate pin goes low OR
  Write to the Counter Latches

3.7.2 Dual 8-Bit Operation

This latch count down option treats the 16-bit word in the latch as two separate 8-bit wide words. The output transition occurs **before** the time out occurs. The dual 8-bit operation is described below:

Dual 8-bit operation



B = The count in the LSB latch +1, times the count in the MSB latch +1, times the period of the clock.

C = The count in the LSB latch period.

The control register programming for the two cases of Counter Initialization in the Dual 8-bit Mode are shown below:

7	6	5	4	3	2	1	0
X	X	0	1	0	1	X	X

CONTROL REGISTER #X

Interrupt:
  Occurs at each Time Out
Counter Initialization:
  Reset OR
  Gate pin goes low

7	6	5	4	3	2	1	0
X	X	0	0	0	1	X	X

CONTROL REGISTER #X

Interrupt:
  Occurs at each Time Out
Counter Initialization:
  Reset OR
  Gate pin goes low OR
  Write to the Counter latches

3.8 SINGLE SHOT MODE

Counter Enable (as refers to **all** Single Shot conditions)
  Reset Clear

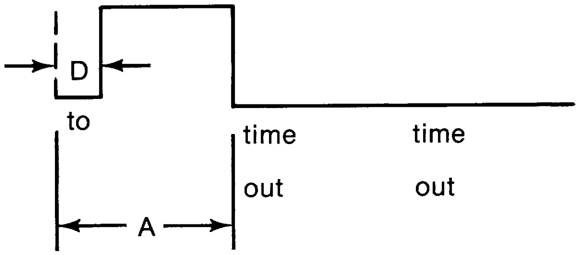
3.8.1 16-Bit Operation

The count-down operation is the same as described in 3.7.1. The only differences are:

- The output is hgh during the first time out as shown below.
- The gate pin need not be held low to enable the timer, but a low **transition** (Counter Initialization) will cause a "Single Shot."

The 16-bit operation of the Single Shot Mode is shown below:

16-bit operation



A = The total 16-bit count in the latch +1, times the period of the clock
D = one count times the clock period
The control register programming for the two cases of Counter Initialization in 16-bit operation are shown below. Each initialization causes a single shot (even during a single shot) if the counter is enabled;

7	6	5	4	3	2	1	0
1	X	1	1	0	0	X	X

CONTROL REGISTER #X

Interrupt:
  Occurs at each Time Out
Counter Initialization:
  Reset OR
  Gate pin goes low

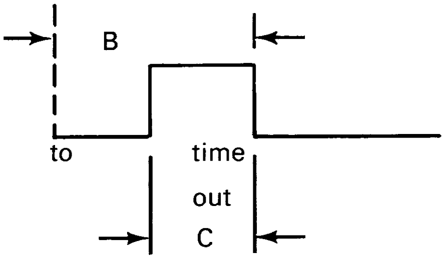
7	6	5	4	3	2	1	0
1	X	1	0	0	0	X	X

CONTROL REGISTER #X

Interrupt:
  Occurs at each Time Out
Counterlnitialization:
  Reset OR
  Gate pin goes low OR
  Write to the Counter Latches

3.8.2 Dual 8-Bit Operation

This count down option is the same as described in 3.7.1.



B = the counter in the LSB latch +1, times the count in the MSB latch +1, times the period of the clock.
C = the count in the LSB latch times the period of the clock.
The control register programming for the two cases of Counter Initialization in Dual 8-bit operation are shown below:

7	6	5	4	3	2	1	0
1	X	1	1	0	1	X	X

CONTROL REGISTER #X

Interrupt:
  Occurs at each Time Out
Counter Initialization:
  Reset OR
  Gate pin goes low

7	6	5	4	3	2	1	0
1	X	1	0	0	1	X	X

CONTROL REGISTER #X

Interrupt:
  Occurs at each Time Out
Counter Initialization:
  Reset OR
  Gate pin goes low OR
  Write to the Counter Latches

Bit 7, the output enable bit must be high in the Single Shot Mode. This enables the output for the "Single Shot." Internally, the count recycling is continuous as if in the Continuous Mode. Only one pulse is evident on the output pin for each Counter Initialization.

3.9 PERIOD MEASUREMENT MODE

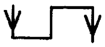
Counter Enable (refers to all Period Measurement Mode conditions).

The gate pin goes low AND
No write to the Counter Latches AND
Reset is cleared AND
There is no individual interrupt flag asserted

This mode compares the period of a digital signal with the period generated by the word in the counter times the period of the clock (Time Out). The control register may be programmed to generate an interrupt if the period of the incoming signal is elss than or greater than that of the Time Out. The digital signal to be measured is applied to the individual gate pin that is assigned to each timer. The two cases of control register programming are shown below:

7	6	5	4	3	2	1	0
0	X	0	0	1	X	X	X

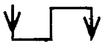
CONTROL REGISTER #X

Interrupt Flag Set:
  Interrupt will be generated if the  duration of the signal on the timer's gate pin is less than the Time Out
Counter Initialization:
  The gate pin goes low AND
  There is no individual interrupt flag asserted AND
  *(A Reset Counter Enable OR a Time Out has occurred) OR

*This prevents initialization on the trailing edge of a previous period measurement.

7	6	5	4	3	2	1	0
0	X	1	0	1	X	X	X

CONTROL REGISTER #X

Interrupt Flag Set:
Interrupt will be generated if the  duration of the signal on the timer's gate pin is greater the Time Out

Counter Initialization:
The gate pin goes low AND
There is no individual interrupt flag asserted OR
Reset

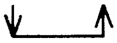
3.10 PULSE WIDTH MEASUREMENT MODE

Counter Enable (refers to **all** Pulse Width Measurement Mode conditions)
The gate pin goes low AND
No write to the Counter Latches AND
Reset is cleared AND
There is no individual interrupt flag asserted

The Pulse Width Comparison Mode operates in the same manner as the Period Measurement Mode in 3.9, except the "down time" of the pulse is compared to the Time Out. The pulse on the gate pin is defined as the period from the negative transition causing initialization to the first positive transition of the gate. An interrupt may be generated if the period of the "down time" pulse is greater than or less than the Time Out. The two cases of control register programming are shown below:

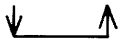
7	6	5	4	3	2	1	0
0	X	0	1	1	X	X	X

CONTROL REGISTER #X

Interrupt Flag Set:
Interrupt will be generated if the  duration of the signal on the timer's gate pin is less than the Time Out.

Counter Initialization:
The gate pin goes low AND
There is no individual interrupt flag asserted OR
Reset.

7	6	5	4	3	2	1	0
0	X	1	1	1	X	X	X

Interrupt Flag Set:
Interrupt will be generated if the  duration of the signal on the timer's gate pin is greater than the Time Out.

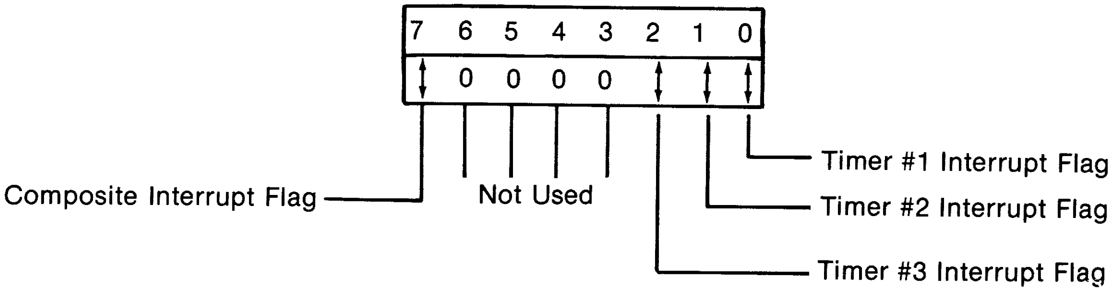
Counter Initialization:
The gate pin goes low AND
There is no individual interrupt flag asserted OR
Reset.

NOTE: The output is usually not used in the Period or Pulse Width Comparison Modes. It is defined. If the output of the timer is enabled, the output will operate in the following manner:

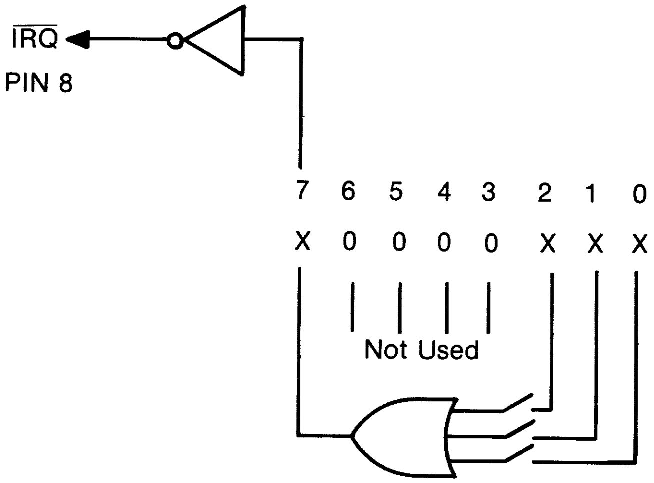
- **During** the first Time Out the output will be low.
- **If** the first Time Out is completed, the output will go high at the Time Out completion.
- **If** further Time Outs are allowed to be completed, the output will change state at the completion of each Time Out.

3.11 STATUS REGISTER OPERATION

The Status Register is a read only register that contains the interrupt flags, the three timers and the Composite Interrupt.



The relationship between the individual interrupt flags, the composite interrupt and the $\overline{\text{IRQ}}$ pin is shown below:



Bit 6 in each control register "closes" the switch in the above functional diagram when it is a logic "1". When "open" (bit 6 = 0) the functional OR gate input is low. The basic characteristics of the Status Register are:

- ● Individual timer interrupts cannot be masked.
- ● A composite interrupt is caused by a timer interrupt **and** that timer's interrupt flag enabled (CRX6 = 1).
- ● A composite interrupt causes IRQ to be asserted.

What clears interrupts:
 Hardware reset (RESET = 0)
 Software reset (CR10 = 1)

Read the status register (RS), then read the timer (RT) causing the interrupt. (An interrupt that occurs between RS and RT will **not** be cleared.)

Writing to the latches. (IF CR x 3 = 0 and CR x 4 = 0)

The gate (GX) goes low. (IF CR x 3 = 0)

CHAPTER 4
SAMPLE SOFTWARE INITIALIZATION ROUTINE

4.0 INTRODUCTION

The attached sample initialization routine demonstrates the ease of initializing the PTM. The purpose of this routine is to set up the timer in the continuous mode. By cascading the outputs (shown on Figure 4-1) to the external clock inputs, interrupts of from a few microseconds to over 282 years can be achieved on timer one.

4.1 PTM INITIALIZATION

The PTM is interfaced to the MPU as shown in Chapter 2, Figure 2-1. The external output and input pins of the PTM are connected as shown in Figure 4-1. The PTM may now be initialized. In order to achieve maximum software efficiency, the PTM should be initialized in the following sequence:

1. First write the latch data to the three counter latches —

```
LDX
STX
```

If the latches are not written, they default to $FFFF. The time out data may be loaded directly into the index register from two consecutive RAM memory locations. (e.g. This permits a routine to calculate a time out and leave it to be retrieved later for timer data.) The overall effect is a single instruction, 16 bit wide timer data word that is very software efficient. Since "writing to" or "reading from" a timer appears exactly like two successive memory locations to the MPU (M & M + 1), the index register and stack pointer operations handle the 16 bit timer data. The three timers occupy six consecutive locations and are totally independent.

2. Next, the control word is written to the three timer control registers. The three control registers differ only in their least significant bits. The three registers occupy two memory locations. Address "zero" (RS0 = 0, RS1 = 0, RS2 = 0) is in the location of both control register #1 and control register #3. The state of bit 0 of control register #2 selects which control register is selected at address "zero". (Bit 0 = 0 - control register #3 is selected; bit 0 = 1 - control register #1 is selected). Address "one" (RS0 = 1, RS1 = 0, RS2 = 0) contains only control register #2. Upon initialization (power up or hardware reset), bit 0 of control register #2 defaults to zero (control register #3 is selected address "zero").

The easiest method of loading the control words is the following order:

WRITE CONTROL REGISTER #3
WRITE CONTROL REGISTER #2
WRITE CONTROL REGISTER #1

The part powers up so Control Register #3 can be accessed directly at RS0 = RS1 = RS2 = 0. When writing control register #2, control register #1 is enabled by the user by writing bit 0 = 1. When writing register #1, bit 0 "turns on" all three timers.

The sample initialization routine, Figure 4-2, is a printout of an actual initialization of the PTM. Figure 4-3, MC6840 Control Register Programming, describes the function of the bits in each of the control registers.

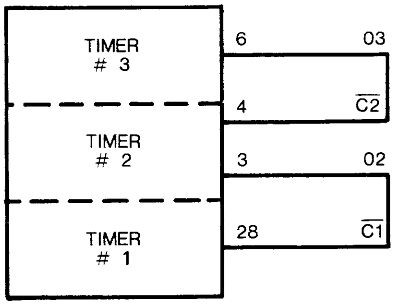
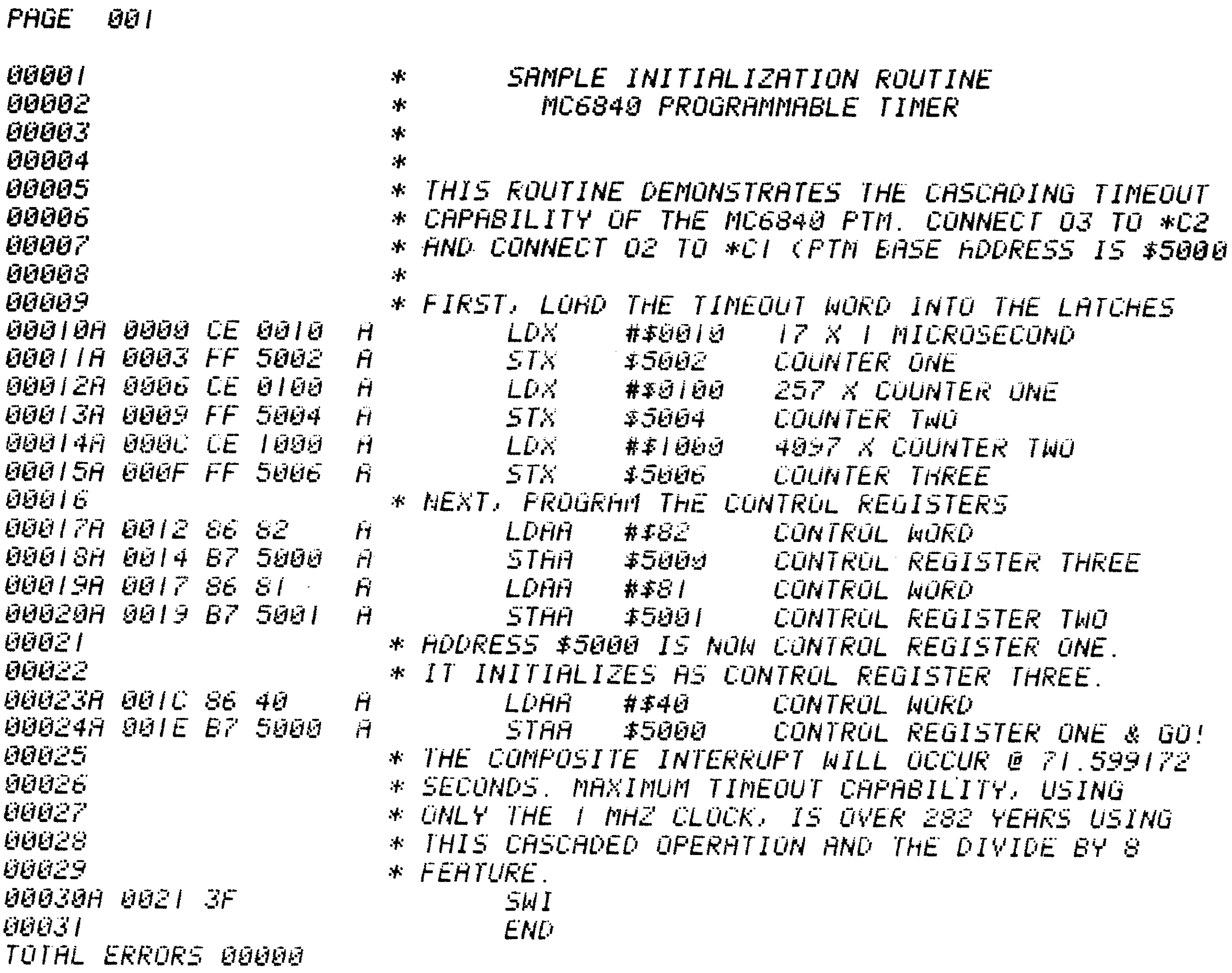


Figure 4-1 PTM Cascaded Operation



```
PAGE  001

00001                   *      SAMPLE INITIALIZATION ROUTINE
00002                   *        MC6840 PROGRAMMABLE TIMER
00003                   *
00004                   *
00005                   * THIS ROUTINE DEMONSTRATES THE CASCADING TIMEOUT
00006                   * CAPABILITY OF THE MC6840 PTM. CONNECT O3 TO *C2
00007                   * AND CONNECT O2 TO *C1 (PTM BASE ADDRESS IS $5000
00008                   *
00009                   * FIRST, LOAD THE TIMEOUT WORD INTO THE LATCHES
00010A 0000 CE 0010  A         LDX    #$0010    17 X 1 MICROSECOND
00011A 0003 FF 5002  A         STX    $5002     COUNTER ONE
00012A 0006 CE 0100  A         LDX    #$0100    257 X COUNTER ONE
00013A 0009 FF 5004  A         STX    $5004     COUNTER TWO
00014A 000C CE 1000  A         LDX    #$1000    4097 X COUNTER TWO
00015A 000F FF 5006  A         STX    $5006     COUNTER THREE
00016                   * NEXT, PROGRAM THE CONTROL REGISTERS
00017A 0012 86 82    A         LDAA   #$82      CONTROL WORD
00018A 0014 B7 5000  A         STAA   $5000     CONTROL REGISTER THREE
00019A 0017 86 81    A         LDAA   #$81      CONTROL WORD
00020A 0019 B7 5001  A         STAA   $5001     CONTROL REGISTER TWO
00021                   * ADDRESS $5000 IS NOW CONTROL REGISTER ONE.
00022                   * IT INITIALIZES AS CONTROL REGISTER THREE.
00023A 001C 86 40    A         LDAA   #$40      CONTROL WORD
00024A 001E B7 5000  A         STAA   $5000     CONTROL REGISTER ONE & GO!
00025                   * THE COMPOSITE INTERRUPT WILL OCCUR @ 71.599172
00026                   * SECONDS. MAXIMUM TIMEOUT CAPABILITY, USING
00027                   * ONLY THE 1 MHZ CLOCK, IS OVER 282 YEARS USING
00028                   * THIS CASCADED OPERATION AND THE DIVIDE BY 8
00029                   * FEATURE.
00030A 0021 3F                 SWI
00031                          END
TOTAL ERRORS 00000
```

Figure 4-2 Sample MC6840 Printout

7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	↕

	REGISTER 1	REGISTER 2	REGISTER 3
0	ALL TIMERS OPERATE	REG #3 MAY BE WRITTEN	T3 $\overline{\text{CLK}}$ ÷ 1
1	ALL TIMERS PRESET	REG #1 MAY BE WRITTEN	T3 $\overline{\text{CLK}}$ ÷ 8

7	6	5	4	3	2	1	0
X	X	X	X	X	X	↕	X

0	EXTERNAL CLOCK ($\overline{\text{CX}}$ INPUT)
1	INTERNAL CLOCK (ENABLE)

7	6	5	4	3	2	1	0
X	X	X	X	X	↕	X	X

0	NORMAL (16 BIT) COUNT MODE
1	DUAL 8 BIT COUNT MODE

7	6	5	4	3	2	1	0
X	X	0	0	0	X	X	X

CONTINUOUS OPERATING MODE: $\overline{\text{GATE}}$ ↓ OR WRITE TO LATCHES OR RESET CAUSES COUNTER INITIALIZATION

7	6	5	4	3	2	1	0
X	X	0	0	1	X	X	X

FREQUENCY COMPARISON MODE: INTERRUPT IF $\overline{\text{GATE}}$ ↓_┌‾↓ IS < COUNTER TIME OUT

7	6	5	4	3	2	1	0
X	X	0	1	0	X	X	X

CONTINUOUS OPERATING MODE: $\overline{\text{GATE}}$ ↓ OR RESET CAUSES COUNTER INITIALIZATION

7	6	5	4	3	2	1	0
X	X	0	1	1	X	X	X

PULSE WIDTH COMPARISON MODE: INTERRUPT IF $\overline{\text{GATE}}$ ↓_↑ IS < COUNTER TIME OUT

7	6	5	4	3	2	1	0
1	X	1	0	0	X	X	X

SINGLE SHOT MODE: $\overline{\text{GATE}}$ ↓ OR WRITE TO LATCHES OR RESET CAUSES COUNTER INITIALIZATION

7	6	5	4	3	2	1	0
X	X	1	0	1	X	X	X

FREQUENCY COMPARISON MODE: INTERRUPT IF $\overline{\text{GATE}}$ ↓_┌‾↓ IS > COUNTER TIME OUT

7	6	5	4	3	2	1	0
1	X	1	1	0	X	X	X

SINGLE SHOT MODE: $\overline{\text{GATE}}$ ↓ OR RESET CAUSES COUNTER INITIALIZATION

7	6	5	4	3	2	1	0
X	X	1	1	1	X	X	X

PULSE WIDTH COMPARISON MODE: INTERRUPT IF $\overline{\text{GATE}}$ ↓_↑ IS > COUNTER TIME OUT

7	6	5	4	3	2	1	0
X	↕	X	X	X	X	X	X

0	INTERRUPT FLAG MASKED ($\overline{\text{IRQ}}$)
1	INTERRUPT FLAG ENABLED ($\overline{\text{IRQ}}$)

7	6	5	4	3	2	1	0
↕	X	X	X	X	X	X	X

0	TIMER OUTPUT MASKED
1	TIMER OUTPUT ENABLE

NOTE: RESET IS HARDWARE OR SOFTWARE RESET
($\overline{\text{RESET}}$ = 0 OR CR10 = 1)

Figure 4-3 MC6840 Control Register Programming

CHAPTER 5

5.0 INSTRUCTION

This chapter describes four MC6840 Programmable Timer Module applications.

Programming Four Modes on the Programmable Timer
Thermometer
Real Time Timer
Ignition Analyzer

The PTM has many other applications in pulse generation, interval and period measurement, industrial timing control and programmable one-shot uses. The four applications in this chapter attempt to demonstrate the MC6840 PTM's versatility in different applications.

5.1 PROGRAMMING FOUR MODES ON THE PROGRAMMABLE TIMER

The purpose of this program is to bring a software tool into the hands of the user which demonstrates the four basic modes of the MC6840. The program flow chart is shown in Figure 5-1.

5.1.1 The 16 Bit Continuous Mode

The first mode in the program (Figure 5-2) is the normal 16 bit continuous mode. In this mode any timer will output a 50% duty cycle wave form at various frequencies (the period is twice the value in the latches). The counter decrements through the latches once during which time the output is low, it decrements again and the output is high. The frequence and duty cycle is determined by the number in the latches; i.e. the larger the number (FFFF) the lower the frequency; the smaller the number (0000) the higher the frequency. In this case (0000) a divide by 2 counter is established. In this same mode, with the dual 8 bit configuration selected, the duty cycle is whatever the LSB contains and the period is the number in both the latches plus 2. Start the counter decrementing by the gate signal going low ($\overline{G}\downarrow$) and stop the counter at the gate signal going high ($\overline{G}\uparrow$). The switches for $\overline{G}$ and $\overline{C}$ are connected as shown in the schematic in Figure 5-3.

5.1.2 The Single Shot Mode

The second mode is a single shot mode where the pulse can be either the width of the number in the MSB and LSB (normal 16 bit) or it can be the width of the number in the LSB with a delay of whatever is in the MSB (dual 8 bit mode). The pulse starts when the gate signal goes low.

5.1.3 The Period Measurement Mode

In the third mode, the period measurement mode requires $\overline{G}\downarrow$ before starting and another $\overline{G}\downarrow$ before a Time Out to Interrupt the MPU. When an IRQ is present the program jumps down to the pulse width comparison mode.

5.1.4 Pulse Width Measurement Mode

In the pulse width measurement mode, the fourth and last mode, the timers are activated by $\overline{G}\downarrow$, which decrements the latches until $\overline{G}\uparrow$. The $\overline{IRQ}$ line goes low signaling the MPU that a pulse occurred. Then the "duration" can be checked by reading the timer counter.

5.2 PTM THERMOMETER

The MC6840 can be used as a temperature monitoring device as shown in Figure 5-4. An MC1555 is used in the astable mode in order to generate a pulse train for the PTM to monitor. R_B is a thermistor.

In the astable mode the timer is connected so that it will retrigger itself and cause the capacitor voltage to oscillate between 1/3 V_{CC} and 2/3 V_{CC}.

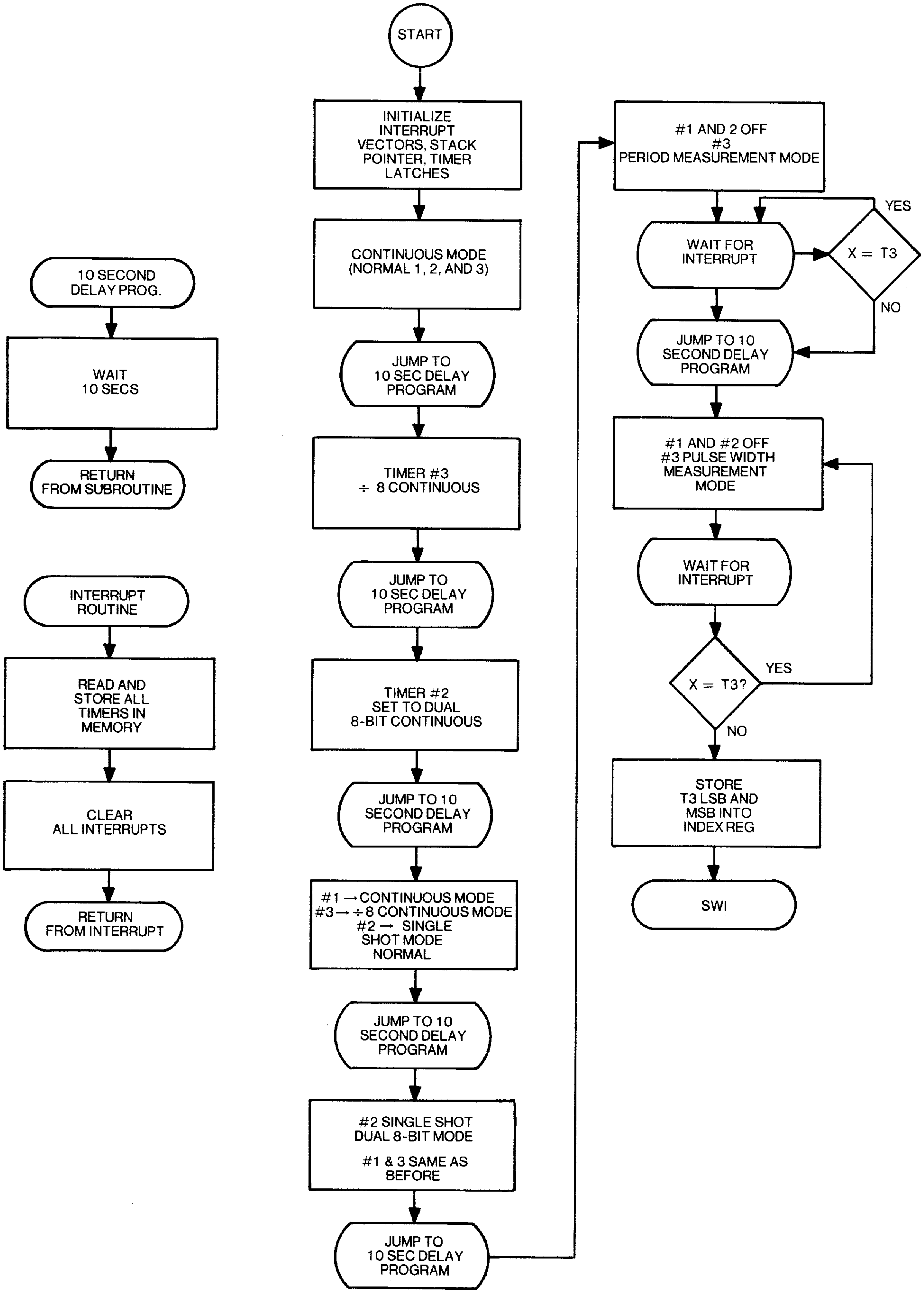
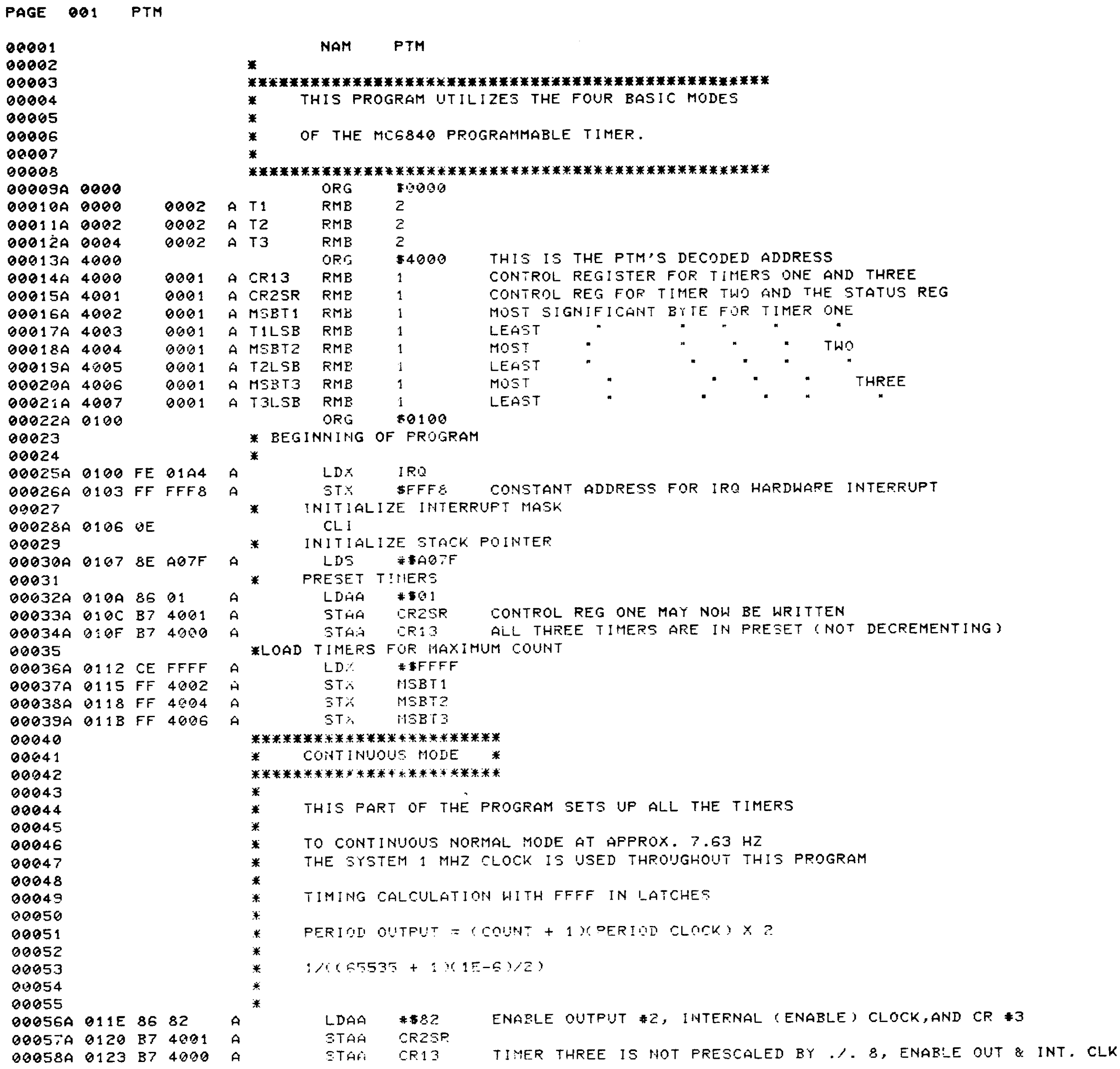

START
INITIALIZE INTERRUPT VECTORS, STACK POINTER, TIMER LATCHES
CONTINUOUS MODE (NORMAL 1, 2, AND 3)
JUMP TO 10 SEC DELAY PROGRAM
TIMER #3 ÷ 8 CONTINUOUS
JUMP TO 10 SEC DELAY PROGRAM
TIMER #2 SET TO DUAL 8-BIT CONTINUOUS
JUMP TO 10 SECOND DELAY PROGRAM
#1 → CONTINUOUS MODE
#3→ ÷ 8 CONTINUOUS MODE
#2 → SINGLE SHOT MODE NORMAL
JUMP TO 10 SECOND DELAY PROGRAM
#2 SINGLE SHOT DUAL 8-BIT MODE
#1 & 3 SAME AS BEFORE
JUMP TO 10 SEC DELAY PROGRAM
#1 AND 2 OFF #3 PERIOD MEASUREMENT MODE
WAIT FOR INTERRUPT
X = T3
YES
NO
JUMP TO 10 SECOND DELAY PROGRAM
#1 AND #2 OFF #3 PULSE WIDTH MEASUREMENT MODE
WAIT FOR INTERRUPT
X = T3?
YES
NO
STORE T3 LSB AND MSB INTO INDEX REG
SWI
10 SECOND DELAY PROG.
WAIT 10 SECS
RETURN FROM SUBROUTINE
INTERRUPT ROUTINE
READ AND STORE ALL TIMERS IN MEMORY
CLEAR ALL INTERRUPTS
RETURN FROM INTERRUPT

Figure 5-1 PTM Four Mode Program Flow Chart



```
PAGE  001   PTM

00001                           NAM     PTM
00002                      *
00003                      ****************************************************
00004                      *     THIS PROGRAM UTILIZES THE FOUR BASIC MODES
00005                      *
00006                      *     OF THE MC6840 PROGRAMMABLE TIMER.
00007                      *
00008                      ****************************************************
00009A 0000                     ORG     $0000
00010A 0000     0002  A T1      RMB     2
00011A 0002     0002  A T2      RMB     2
00012A 0004     0002  A T3      RMB     2
00013A 4000                     ORG     $4000     THIS IS THE PTM'S DECODED ADDRESS
00014A 4000     0001  A CR13    RMB     1         CONTROL REGISTER FOR TIMERS ONE AND THREE
00015A 4001     0001  A CR2SR   RMB     1         CONTROL REG FOR TIMER TWO AND THE STATUS REG
00016A 4002     0001  A MSBT1   RMB     1         MOST SIGNIFICANT BYTE FOR TIMER ONE
00017A 4003     0001  A T1LSB   RMB     1         LEAST   "          "   "    "    "
00018A 4004     0001  A MSBT2   RMB     1         MOST    "          "    "    "   TWO
00019A 4005     0001  A T2LSB   RMB     1         LEAST   "          "    "    "    "
00020A 4006     0001  A MSBT3   RMB     1         MOST     "          "    "    "    THREE
00021A 4007     0001  A T3LSB   RMB     1         LEAST    "          "    "    "     "
00022A 0100                     ORG     $0100
00023                      * BEGINNING OF PROGRAM
00024                      *
00025A 0100 FE 01A4   A           LDX     IRQ
00026A 0103 FF FFF8   A           STX     $FFF8     CONSTANT ADDRESS FOR IRQ HARDWARE INTERRUPT
09027                      *     INITIALIZE INTERRUPT MASK
00028A 0106 0E                    CLI
00029                      *     INITIALIZE STACK POINTER
00030A 0107 8E A07F   A           LDS     #$A07F
00031                      *     PRESET TIMERS
00032A 010A 86 01     A           LDAA    #$01
00033A 010C B7 4001   A           STAA    CR2SR     CONTROL REG ONE MAY NOW BE WRITTEN
00034A 010F B7 4000   A           STAA    CR13      ALL THREE TIMERS ARE IN PRESET (NOT DECREMENTING)
00035                      *LOAD TIMERS FOR MAXIMUM COUNT
00036A 0112 CE FFFF   A           LDX     #$FFFF
00037A 0115 FF 4002   A           STX     MSBT1
00038A 0118 FF 4004   A           STX     MSBT2
00039A 011B FF 4006   A           STX     MSBT3
00040                      ************************
00041                      *     CONTINUOUS MODE    *
00042                      ************************
00043                      *
00044                      *     THIS PART OF THE PROGRAM SETS UP ALL THE TIMERS
00045                      *
00046                      *     TO CONTINUOUS NORMAL MODE AT APPROX. 7.63 HZ
00047                      *     THE SYSTEM 1 MHZ CLOCK IS USED THROUGHOUT THIS PROGRAM
00048                      *
00049                      *     TIMING CALCULATION WITH FFFF IN LATCHES
00050                      *
00051                      *     PERIOD OUTPUT = (COUNT + 1)(PERIOD CLOCK) X 2
00052                      *
00053                      *     1/((65535 + 1)(1E-6)/2)
00054                      *
00055                      *
00056A 011E 86 82     A           LDAA    #$82      ENABLE OUTPUT #2, INTERNAL (ENABLE) CLOCK,AND CR #3
00057A 0120 B7 4001   A           STAA    CR2SR
00058A 0123 B7 4000   A           STAA    CR13      TIMER THREE IS NOT PRESCALED BY ./. 8, ENABLE OUT & INT. CLK
```

Figure 5-2   PTM Four Mode Program Assembly Listing

```
00059A 0126 C6 83    A        LDAB   #$83     ENABLE OUTPUT #2, INT CLK AND CR #1
00060A 0128 B7 4000  A        STAA   CR13     TIMERS ARE OPERATIONAL AND DECREMENTING THROUGH THE COUNTERS
00061A 012B BD 0193  A        JSR    DELAY
00062                     *     THIS PART OF THE PROGRAM PUTS TIMER #3 INTO
00063                     *
00064                     *     THE DIVIDE BY 8 PRESCALED MODE (CONTINUOUS)
00065                     *
00066                     *     CYCLING AT ABOUT 0.95 HZ.
00067                     *
00068                     *     TIMING CALCULATION :  1/((65535 + 1)(8E-6) X 2)
00069                     *
00070A 012E B7 4001  A        STAA   CR2SR    ENABLE WRITTING TO CR #3
00071A 0131 86 C3    A        LDAA   #$C3     THIS ALSO ENABLES THE IRQ LINE
00072A 0133 B7 4000  A        STAA   CR13
00073A 0136 BD 0193  A        JSR    DELAY
00074                     *
00075                     *     THIS PART OF THE PROGRAM SCALES TIMER #2
00076                     *
00077                     *     FOR A FREQUENCY OF 15.26 HZ AND A DUTY CYCLE
00078                     *
00079                     *     OF APPROX. 0.4% .
00080                     *
00081                     *     PERIOD = 65536 ENABLE PULSES X 1E-6 = 0.066
00082                     *
00083                     *     FREQUENCY = 15.26 HZ
00084                     *
00085                     *     DUTY CYCLE = 255 PULSES :  255/65536 = 0.0039
00086                     *
00087A 0139 C6 86    A        LDAB   #$86     USES THE DUAL 8 BIT MODE
00088A 013B F7 4001  A        STAB   CR2SR
00089A 013E BD 0193  A        JSR    DELAY
00090                     ***************************
00091                     *   SINGLE SHOT MODE      *
00092                     ***************************
00093                     *     THIS PART OF THE PROGRAM SETS TIMER #2 INTO
00094                     *
00095                     *     THE SINGLE SHOT WHERE GATING SWITCH #2
00096                     *     PRODUCES A PULSE 0.065 SECONDS WIDE  (1 MHZ SYSTEM CLK)
00097                     *
00098A 0141 86 E3    A        LDAA   #$E3
00099A 0143 B7 4001  A        STAA   CR2SR
00100A 0146 BD 0193  A        JSR    DELAY    ALLOWS YOU TO STAY IN THE MODE FOR APROX. 10 SECONDS
00101                     *************************************
00102                     *    DUAL 8 BIT SINGLE SHOT MODE    *
00103                     *************************************
00104                     *
00105                     *     NOW TIMER #2 IS PLACED INTO THE DUAL 8 BIT MODE
00106                     *
00107                     *     AND THE PULSE WIDTH IS CHANGED TO 256 MICROSECONDS
00108                     *
00109A 0149 86 E7    A        LDAA   #$E7
00110A 014B B7 4001  A        STAA   CR2SR
00111A 014E BD 0193  A        JSR    DELAY
00112                     *******************************************************************
00113                     *    FREQUENCY COMPARISON OR PERIOD MEASUREMENT MODE              *
00114                     *******************************************************************
00115                     *
00116                     *     IN THIS MODE, ALL OF THE TIMERS ARE DISABLED
```

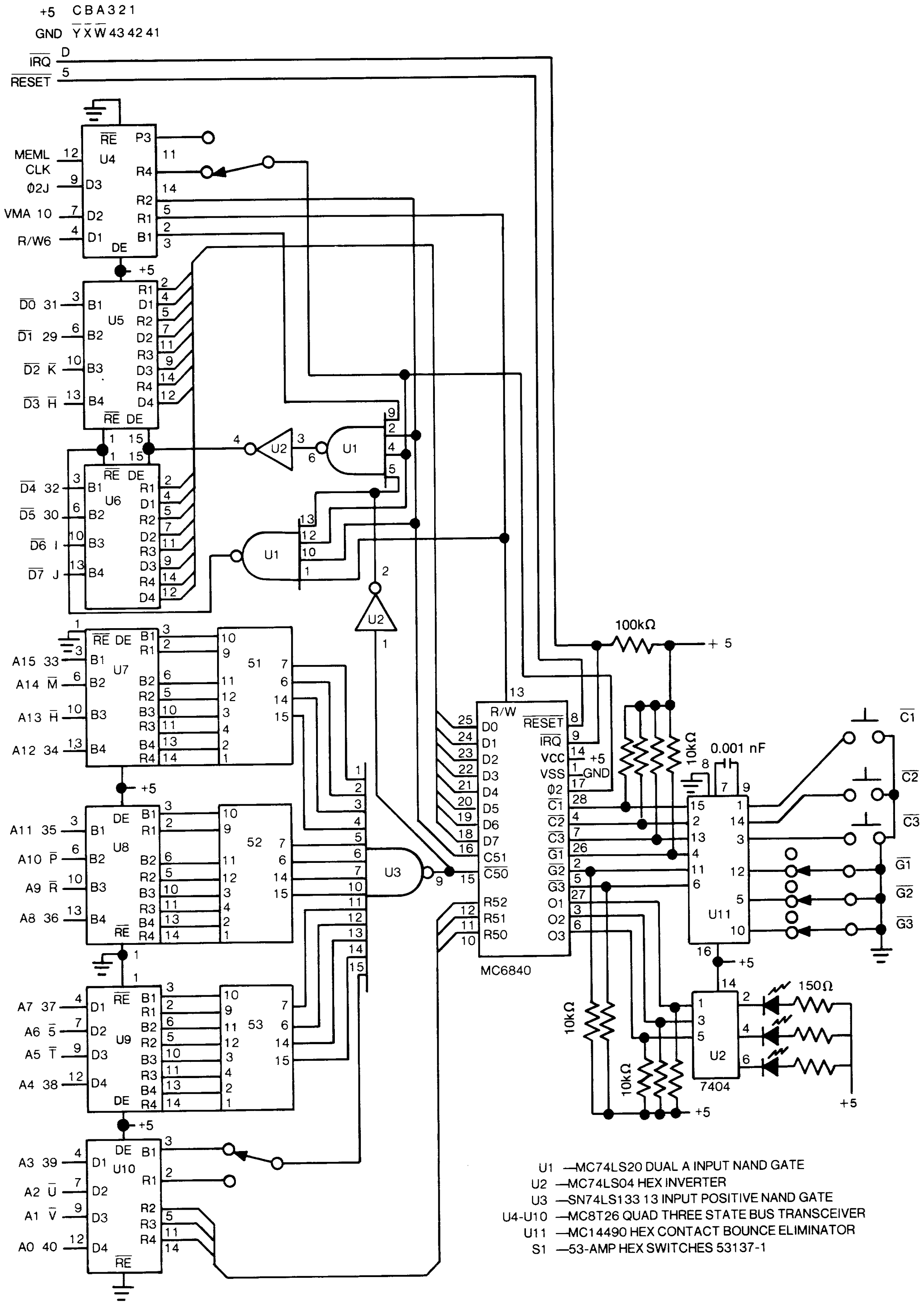
Figure 5-2 PTM Four Mode Program Assembly Listing (cont.)

```
PAGE  003   PTM

00117                     *
00118                     *     EXCEPT FOR #3.  WHEN DETECTING A NEGATIVE
00119                     *
00120                     *     TRANSITION OF THE GATE SWITCH, TIMER #3
00121                     *
00122                     *     CONTINUES TO COUNT DOWN UNTIL ANOTHER
00123                     *
00124                     *     POSITIVE AND NEGATIVE TRANSITION IS MADE.  IF THIS
00125                     *
00126                     *     IS DONE BEFORE A TIMEOUT HAS OCCURED, TIMER #3
00127                     *
00128                     *     PUTS OUT AN INTERRPUT (PULLS THE IRQ LINE LOW)
00129                     *
00130                     *     THE INTERUPPT SUBROUTINE IS JUMPED TO AND THE TIMERS
00131                     *
00132                     *     ARE READ, AND THE PROGRAM CONTINUES TO THE PULSE WIDTH
00133                     *
00134                     *     COMPARISON MODE AFTER A 10 SECOND DELAY.  IF THE TIME OUT
00135                     *
00136                     *     OCCURED FIRST, THE TIMER CONTENTS ARE STILL STORED,
00137                     *
00138                     *     BUT THE COMPARISON BRANCES BACK UP TO THE BEGINNING
00139                     *
00140                     *     TO GIVE THE SWITCH OPERATOR ANOTHER CHANCE
00141                     *
00142                     *     AT TOGGLING THE SWITCH FAST ENOUGH TO CHANGE MODES
00143                     *
00144A 0151 86 01     A         LDAA    #$01
00145A 0153 B7 4001   A         STAA    CR2SR     CR #1 MAY BE WRITTEN
00146A 0156 86 00     A         LDAA    #$00
00147A 0158 B7 4000   A         STAA    CR13      OPERATE COUNTERS ( #1 & #2 OUTPUTS ARE MASKED)
00148A 015B B7 4001   A         STAA    CR2SR     CR #3 MAY BE WRITTEN
00149A 015E 86 CF     A         LDAA    #$CF
00150A 0160 B7 4000   A         STAA    CR13      EXPLAINATION ABOVE
00151A 0163 3E            WAIT1 WAI
00152A 0164 CE 0000   A         LDX     #$0000
00153A 0167 9C 04     A         CPX     T3        FAST ENOUGH?
00154A 0169 27 F8 0163          BEQ     WAIT1     IF NOT TRY AGAIN
00155A 016B 86 01     A         LDAA    #$01      IF OPERATOR TOGGLES THE SWITCH QUICK ENOUGH THEN PRESET TIMERS
00156A 016D B7 4001   A         STAA    CR2SR
00157A 0170 B7 4000   A         STAA    CR13
00158A 0173 BD 0193   A         JSR     DELAY
00159                     ****************************************
00160                     *     PULSE WIDTH COMPARISON MODE        *
00161                     ****************************************
00162                     *
00163                     *     THIS PART IS THE SAME AS THE FREQ. OR PERIOD
00164                     *
00165                     *     MEASUREMENT MODE EXCEPT FOR A NEGATIVE TRANSITION
00166                     *
00167                     *     OF THE GATE SWITCH STARTS THE COUNTER AND AN UPWARD
00168                     *
00169                     *     TRANSISTION STOPS IT.
00170                     *
00171A 0176 86 00     A         LDAA    #$00
00172A 0178 B7 4001   A         STAA    CR2SR     CR #3 MAY BE WRITTEN
00173A 017B 86 DF     A         LDAA    #$DF
00174A 017D B7 4000   A         STAA    CR13
```

Figure 5-2. PTM Four Mode Program Assembly Listing (cont.)

```
PAGE  004    PTM

00175A 0180 86 01     A         LDAA   #$01
00176A 0182 B7 4001   A         STAA   CR2SR      CR #1 MAY BE WRITTEN
00177A 0185 7F 4000   A         CLR    CR13       TIMER ALLOWED TO OPERATE (#3)
00178A 0188 3E          WAIT2   WAI
00179A 0189 CE 0000   A         LDX    #$0000
00180A 018C 9C 04     A         CPX    T3
00181A 018E 27 F8 0188          BEQ    WAIT2
00182A 0190 DE 04     A         LDX    T3         IF OPERATOR IS QUICK THEY CAN STOP THE PROGRAM
00183A 0192 3F                  SWI
00184                   *
00185                   ****************************
00186                   *    SUBROUTINE DELAY      *
00187                   ****************************
00188                   *
00189                   *    THIS ROUTINE PRODUCES AN APPROX. 10 SECOND
00190                   *
00191                   *    DELAY BETWEEN MODE CHANGES
00192                   *
00193A 0193 C6 00     A DELAY   LDAB   #$00
00194A 0195 CE 0DDE   A LOOPB   LDX    #$0DDE     INNER COUNTER
00195A 0198 09          LOOPA   DEX
00196A 0199 8C 0000   A         CPX    #$0000
00197A 019C 26 FA 0198          BNE    LOOPA
00198A 019E 5A                  DECB
00199A 019F C1 00     A         CMPB   #$00
00200A 01A1 26 F2 0195          BNE    LOOPB
00201A 01A3 39                  RTS
00202                   ****************************************
00203                   *    CLEAR PTM INTERRUPT SUBROUTINE    *
00204                   ****************************************
00205                   *
00206                   *    THIS ROUTINE CLEARS THE PTM'S INTERRUPTS
00207                   *
00208                   *    BY READING THE STATUS REGISTER AND TIMER
00209                   *
00210                   *    LATCHES.  THE CONTENTS OF THERS LATCHES
00211                   *
00212                   *    ARE STORED IN MEMORY FOR LATER USE.
00213                   *
00214A 01A4 F6 4001   A IRQ     LDAB   CR2SR     READ STATUS REG
00215A 01A7 FE 4002   A         LDX    MSBT1     READ TIMER 1 COUNTER
00216A 01AA DF 00     A         STX    T1
00217A 01AC F6 4001   A         LDAB   CR2SR     READ STATUS REG SO AS NOT TO MISS ANY INTERRUPTS
00218A 01AF FE 4004   A         LDX    MSBT2
00219A 01B2 DF 02     A         STX    2
00220A 01B4 F6 4001   A         LDAB   CR2SR
00221A 01B7 FE 4006   A         LDX    MSBT3
00222A 01BA DF 04     A         STX    T3
00223A 01BC 3B                  RTI
00224                           END
TOTAL ERRORS 00000

   4000 CR13    00014*00034 00058 00060 00072 00147 00150 00157 00174 00177
   4001 CR2SR   00015*00033 00057 00070 00088 00099 00110 00145 00148 00156 00172 00176 00214 00217 00220
   0193 DELAY   00061 00073 00089 00100 00111 00155 00193*
   01A4 IRQ     00025 00214*

PAGE  005    PTM

   0198 LOOPA   00195*00197
   0195 LOOPB   00194*00200
   4002 MSBT1   00016*00037 00215
   4004 MSBT2   00018*00038 00218
   4006 MSBT3   00020*00039 00221
   0000 T1      00010*00216
   4003 T1LSB   00017*
   0002 T2      00011*
   4005 T2LSB   00019*
   0004 T3      00012*00153 00180 00182 00222
   4007 T3LSB   00021*
   0163 WAIT1   00151*00154
   0188 WAIT2   00178*00181
```

Figure 5-2. PTM Four Mode Program Assembly Listing (cont.)



U1 —MC74LS20 DUAL A INPUT NAND GATE
U2 —MC74LS04 HEX INVERTER
U3 —SN74LS133 13 INPUT POSITIVE NAND GATE
U4-U10 —MC8T26 QUAD THREE STATE BUS TRANSCEIVER
U11 —MC14490 HEX CONTACT BOUNCE ELIMINATOR
S1 —53-AMP HEX SWITCHES 53137-1

Figure 5-3 PTM MC6840 Wire Wrap Board Schematic

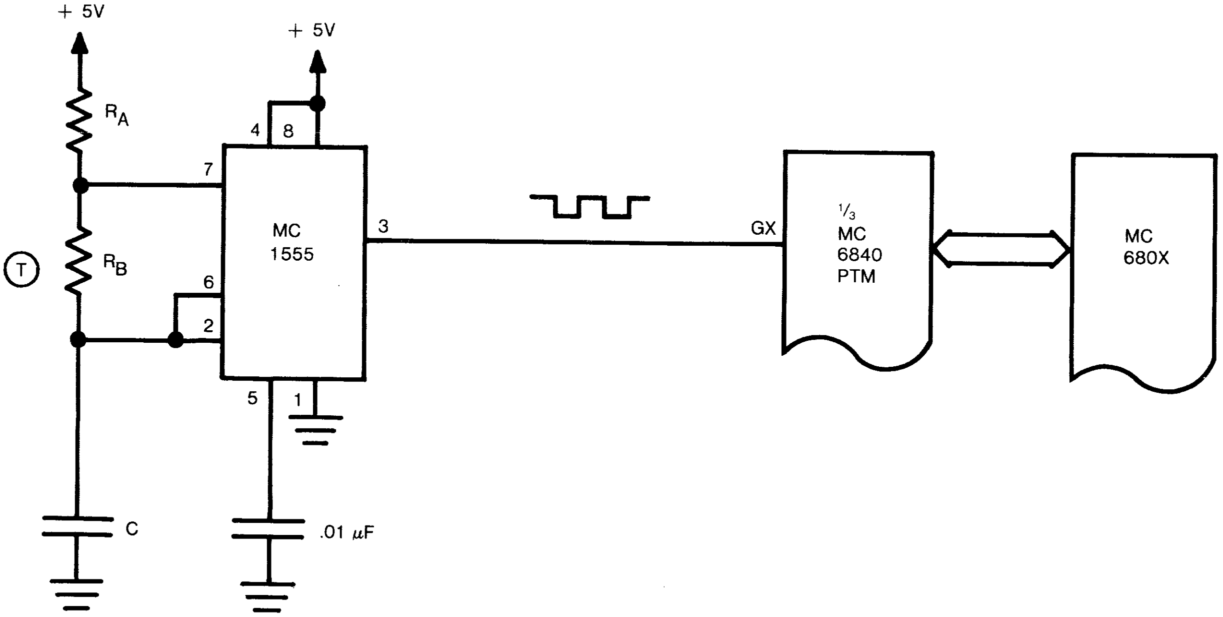


Figure 5-4 Temperature Monitoring Device

The external capacitor charges to 2/3 V_{CC} through RA and RB and discharges to 1/3 VCC through RB. By varying the ratio of these resistors the duty cycle can be varied. The charge and discharge times are independent of the supply voltage.

The charge time (output high) is given by: t1 = 0.695 (RA + RB)C
The discharge time (output low) by: t2 = 0.695 (RB)C
Thus the total period is given by: T = t1 + t2 = 0.695 (RA + 2RB)C
The frequency of oscillation is then: f = 1/T = 1.44/(RA + 2 RB)C
The duty cycle is given by: DC = RB/RA + 2RB
RA must be large enough to limit the discharge current (pin 7 current) within the maximum rating of the discharge transistor (200 mA).

The minimum value of RA is given by: RA = V_{CC}/17 A = V_{CC}/0.2

If R_A is many times greater than R_B (thermistor), the resistance change caused by a temperature change will have only a small effect on the frequency. The change in the pulse down time (output low) will vary in direct proportion to the resistance value of the thermistor.

A single timer in the PTM measures this pulse down time in the pulse width measurement mode. The maximum measurable down time in 65.536 milleseconds using the internal 1 MHz clock usually used in the MPU system (*$FFFF X φ2 period). Ten microseconds is the minimum pulse width recommended. This will insure ten counts in order to obtain a reasonably accurate reading.

When no data word is written to the timer it defaults to $FFFF. The pulse width measurement mode decrements this count each microsecond (the φ2 internal clock period) the pulse is low. When the pulse returns high, the PTM generates an interrupt and waits to be read by the MPU. The MPU reads the decremented count and calculates the temperature. The versatility of the MC6840 PTM allows the user to have one timer used as a thermometer and the other two timers performing other tasks. All three timers may also be used as completely independent thermometers.

5.3 MC6840 PTM CONTROLLED REAL-TIME TIMER

This section describes two applications using the MC6840: A time of day (TOD) clock and an event counter. These two applications have many uses; a software clock in a real time operating system

*hexadecimal number

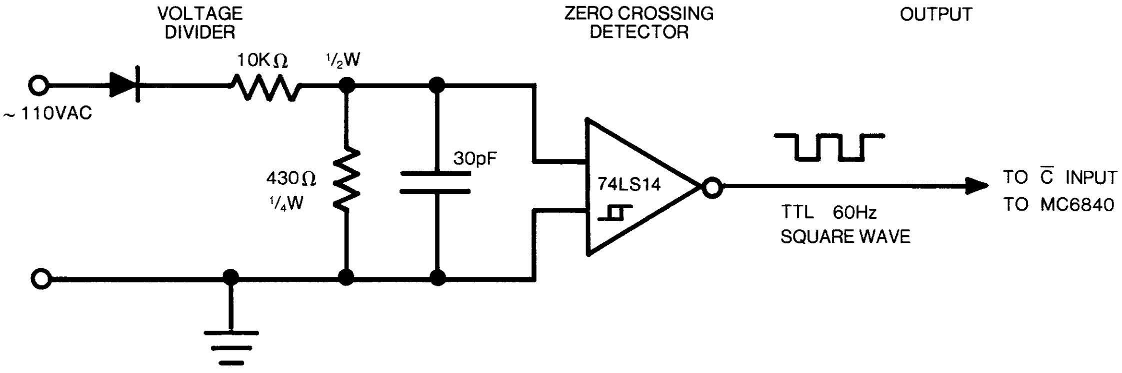


Figure 5-5 Hardware Schematic of TTL Conditioner

(RTOS), control applications which require periodic servicing (i.e. hourly, daily, etc.) or counting a given number of events to determine when to control a process.

Figure 5-5 shows a method of obtaining a 60 Hz square wave from the 60 Hz line. The diode is used to half-wave rectify the incoming sine wave. The voltage divider is used to get a 6-7V peak to peak wave form to be used as an input to the 74LS14 Schmitt trigger. The output of the 74LS14 is a TTL level square wave at 60 Hz. This type of circuit can be modified to condition any non-TTL signal into a TTL signal with some noise protection because of the Schmitt trigger hysterysis. This output is taken to the CX input on the MC6840. The timer is configured as follows: Continuous mode, normal 16-0 bit operation, external clock and interrupt enabled (timer output may be enabled or masked to the users' option). This initiation is done while the PTM is in the preset condition. During initialization the programmed options can be chosen: 12/24 hour clock options, alarm functions, control functions, and other user defined use. The 60 Hz line was chosen as the oscillator because of its availability in the United States and its good stability. If a user desires a crystal operated system with battery back-up, the crystal specification must include an accuracy of ±20 parts per million to insure a system accuracy for the clock and a crystal with battery as a secondary clock for use during a power outage, the accuracy must be 10± PPM to ensure a 1± sec/day accuracy for the back-up clock or a ±500 PPM crystal for a 1± min/day accuracy.

Figures 5-6 and 5-7 show the flow chart for the initialization routine and the interrupt service routine for the TOD clock. All major decision blocks and processes are given. The initialization routine is done only once, usually in the restart routine. A set routine could be implemented to set-up the counters to a pre-set time and start the clock. In the intiialization, two control words are given; one will enable the timer output, the other will disable it. The desired function is user selectable. The counter values may be stored as a binary count. With the use of the DAA instruction, it can be converted into packed BCD number representing hours, minutes and seconds. This last representation would make it easier to output to an ASCII output device — either a serial or a parallel port.

The second application is an event counter. In this configuration the timer is initialized the same as the TOD clock, except the clock input will not come from the source which will generate the clocking pulses. The hours, minutes, and seconds counter may not be needed or may be modified as BCD digits to obtain larger counts if necessary.

5.4 MC6840/MC6802 MPU-PTM CONTROLLED ENGINE ANALYZER

The cost of automobile services continues to rise year by year. People who didn't even know they had a crankcase a few years ago are now regularly changing their own oil. Many "dainty" women are under the hood doing their own maintenance. Auto makers offer service manuals and some even offer tool kits for their do-it-yourself customers. Car parts and crankcase oil are now available in supermarkets and fast service convenience stores. This was unheard of ten years ago.

A tune-up can be a major cost item at either the new car dealers' or your local service station. Many people now do their own tune-up. Brand name parts and tune up equipment are available at most

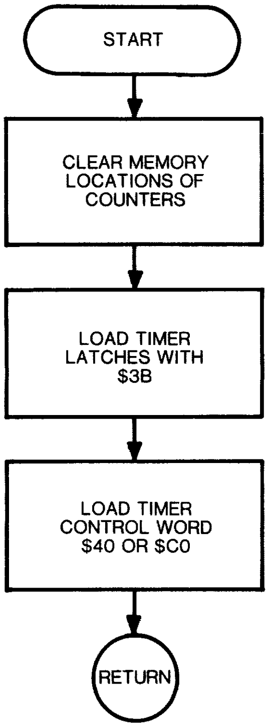

START
CLEAR MEMORY LOCATIONS OF COUNTERS
LOAD TIMER LATCHES WITH $3B
LOAD TIMER CONTROL WORD $40 OR $C0
RETURN

Figure 5-6 Flow Chart to Initialize Timer

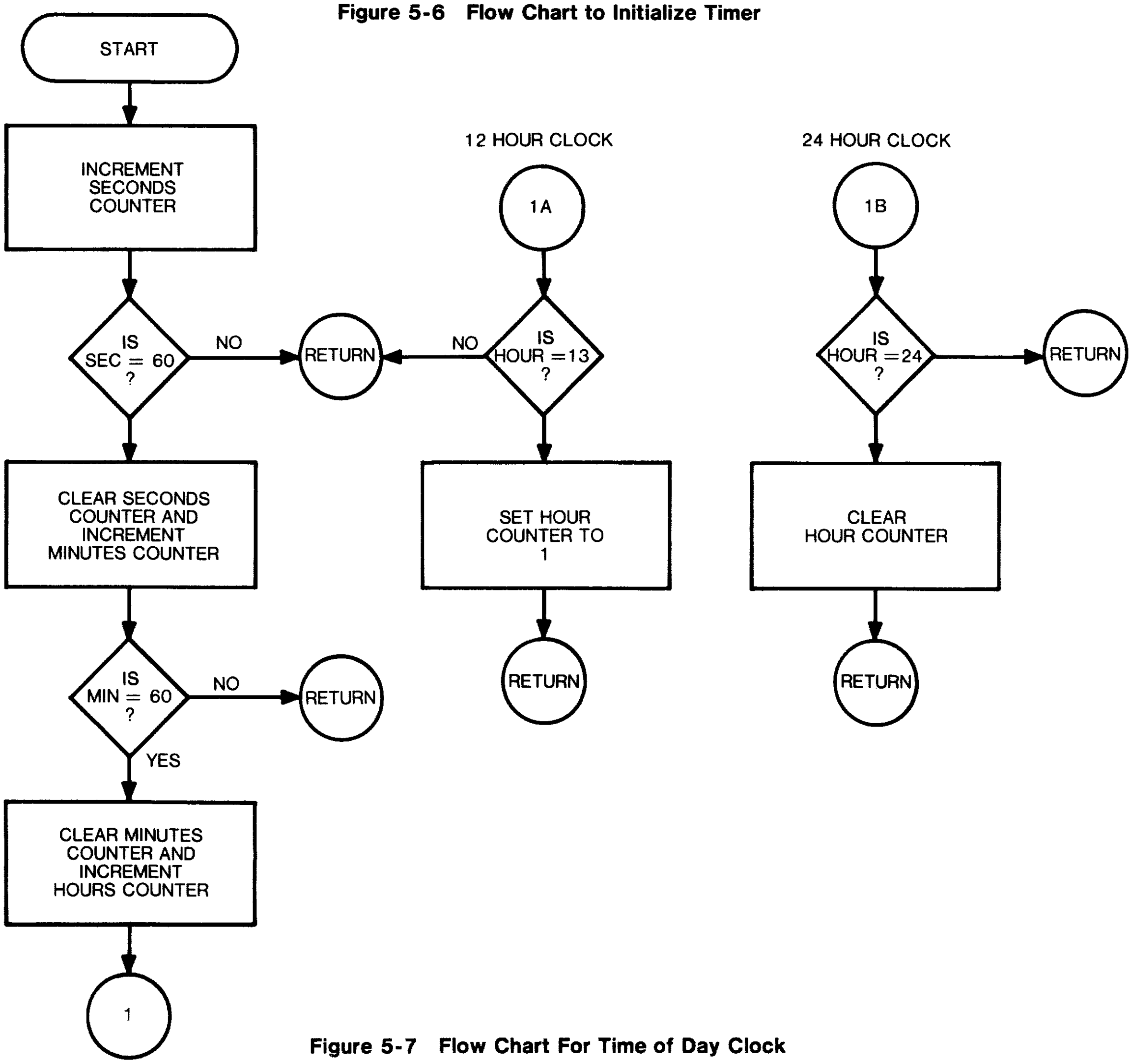


Figure 5-7 Flow Chart For Time of Day Clock

national chain department stores and through mail order sales. Using the analog meter — strobe gun — available today to the do-it-yourselfer — may be less accurate than desired and also presents a safety hazard. The strobe gun that is used to adjust the timing usually must be held a few inches from the fan.

This application deals with using the MC6840 Programmable Timer in a three function engine analyzer aimed at the consumer and service station market. This microprocessor controlled system will yield readings equal to professional new car dealer equipment and still be cost competitive with the better consumer equipment presently on the market.

5.5 ENGINE INPUT HARDWARE

Figures 5-8 through 5-11 show in basic block diagram the basic functioning of the engine analyzer. Figure 5-12 describes a four cylinder engine timing. These figures are shown in greater detail in Figures 5-13 through 5-16.

Three signals must be picked up from the engine and be conditioned to TTL levels:

1. The point closure.
2. Plug number one firing.
3. The point in time when the timing mark is at zero degrees top dead center.

Figure 5-15 shows three approaches to conditioning these three signals to TTL levels. There are several passive and active circuits that will perform the signal conditioning function. Any design technique that produces the TTL level signals shown on Figure 5-15 — at the correct point in time — is adequate. Care must be taken to insure the engines electrical systems are not loaded and the engine performance changed because of the analyzer connections. The output signals from the signal conditioners must contain no "spikes" or "glitches" that could cause false triggering. The point closure signal may be DC coupled into the signal conditioners, but care should be exercised to insure a very high input impedence is present. A low impedence could degrade the performance of the primary winding of the coil. Since thousands of volts are present on plug one, caution must be used here also. A DC coupling is not recommended on the plug. The "zero top dead center" point may be picked up magnetically or optically. A light source shining on a white painted timing mark can be picked up by a photo transistor and then conditioned to a TTL level.

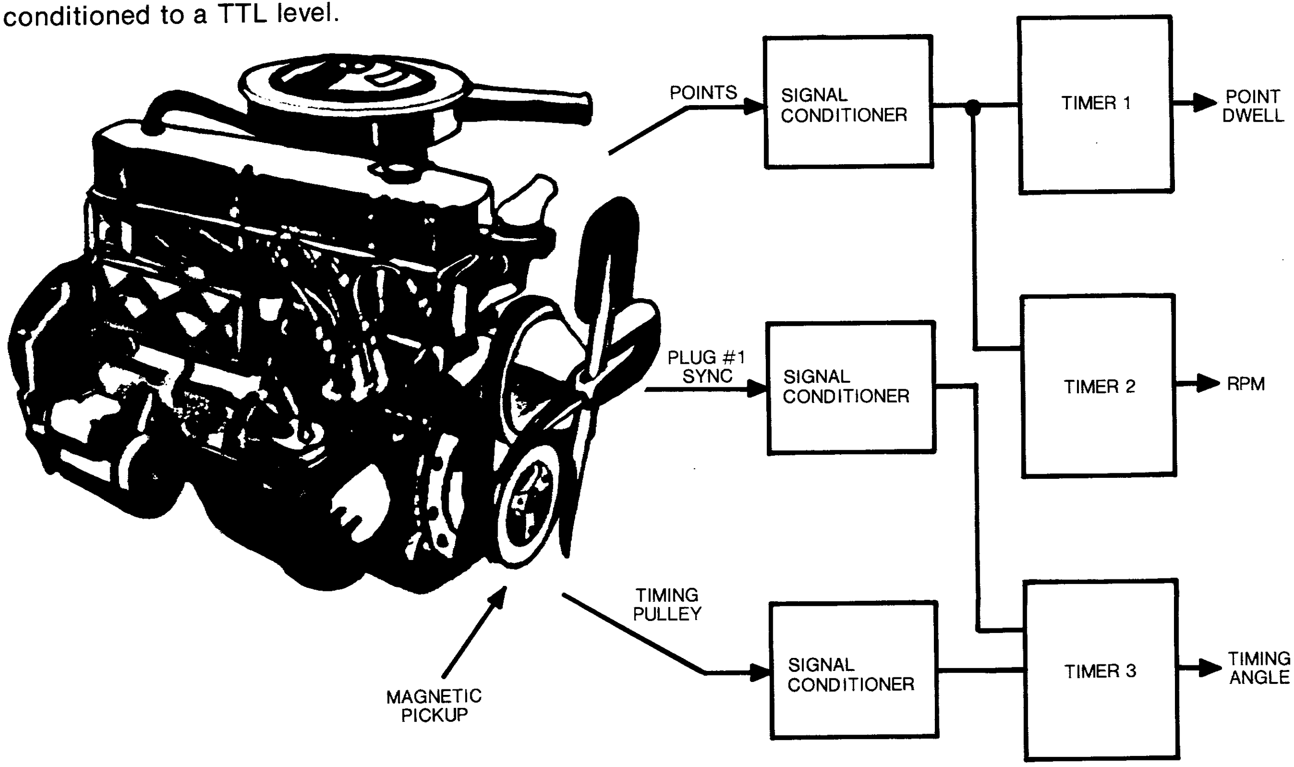


Figure 5-8 Digital Engine Analyzer

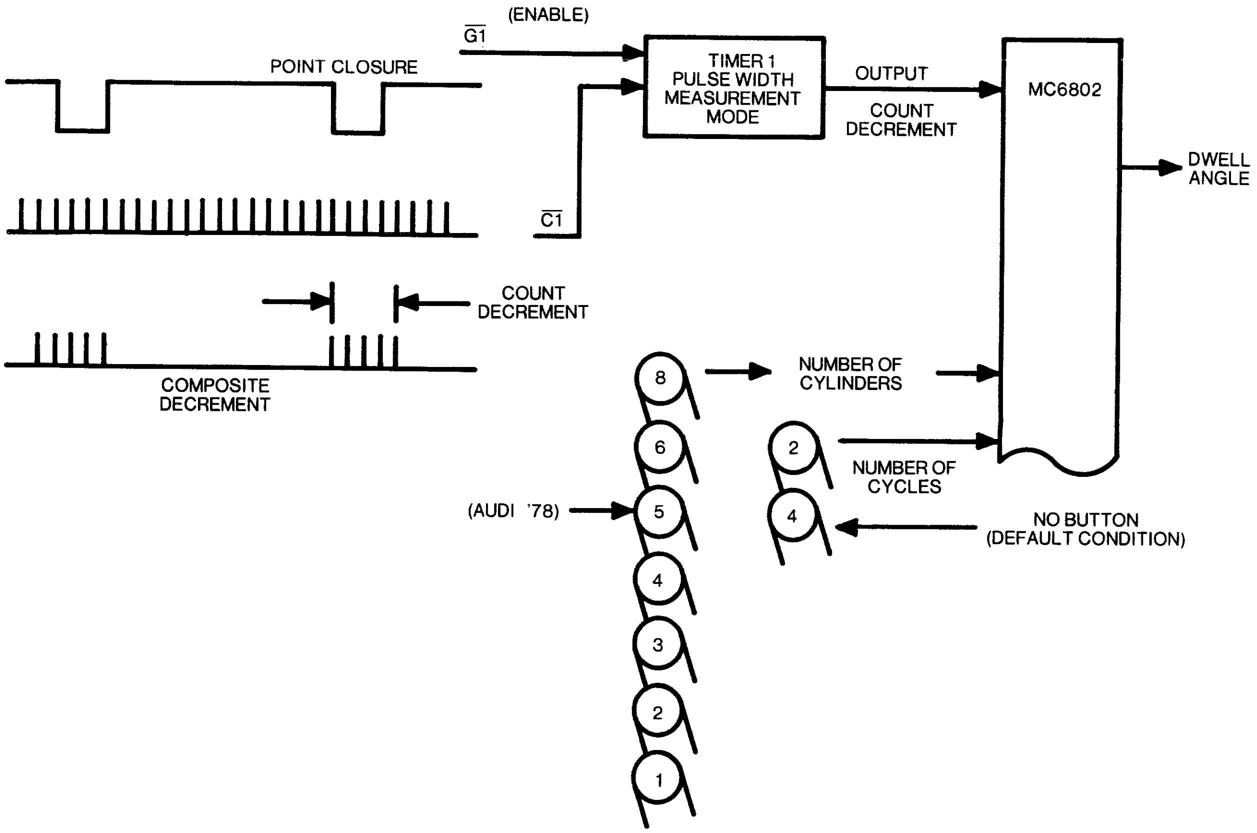


Figure 5-9 Dwell Angle

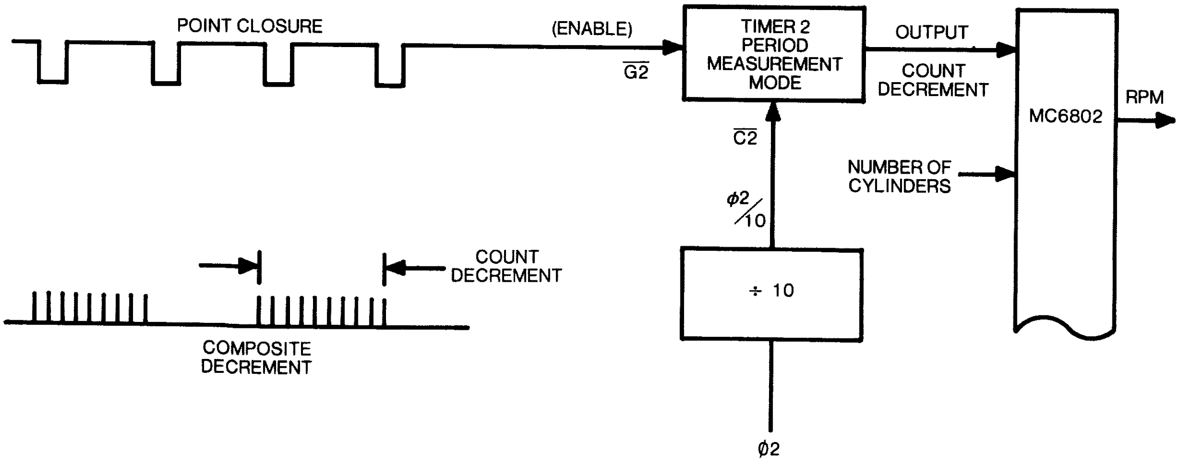


Figure 5-10 RPM

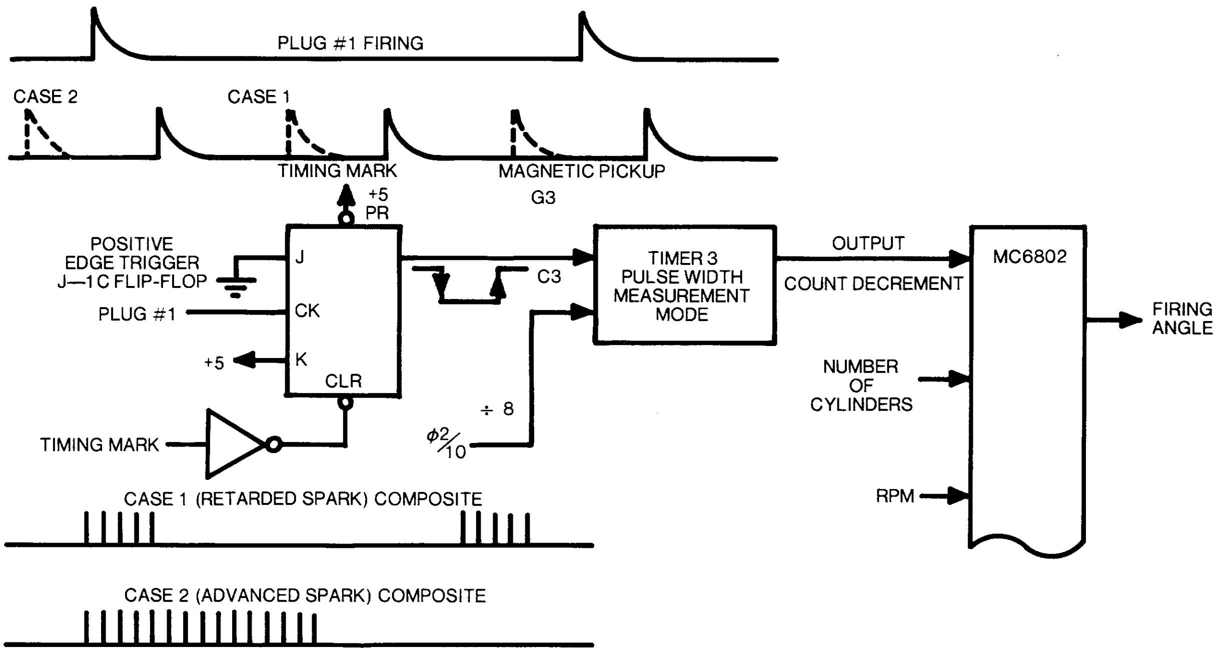


Figure 5-11 Timing Angle

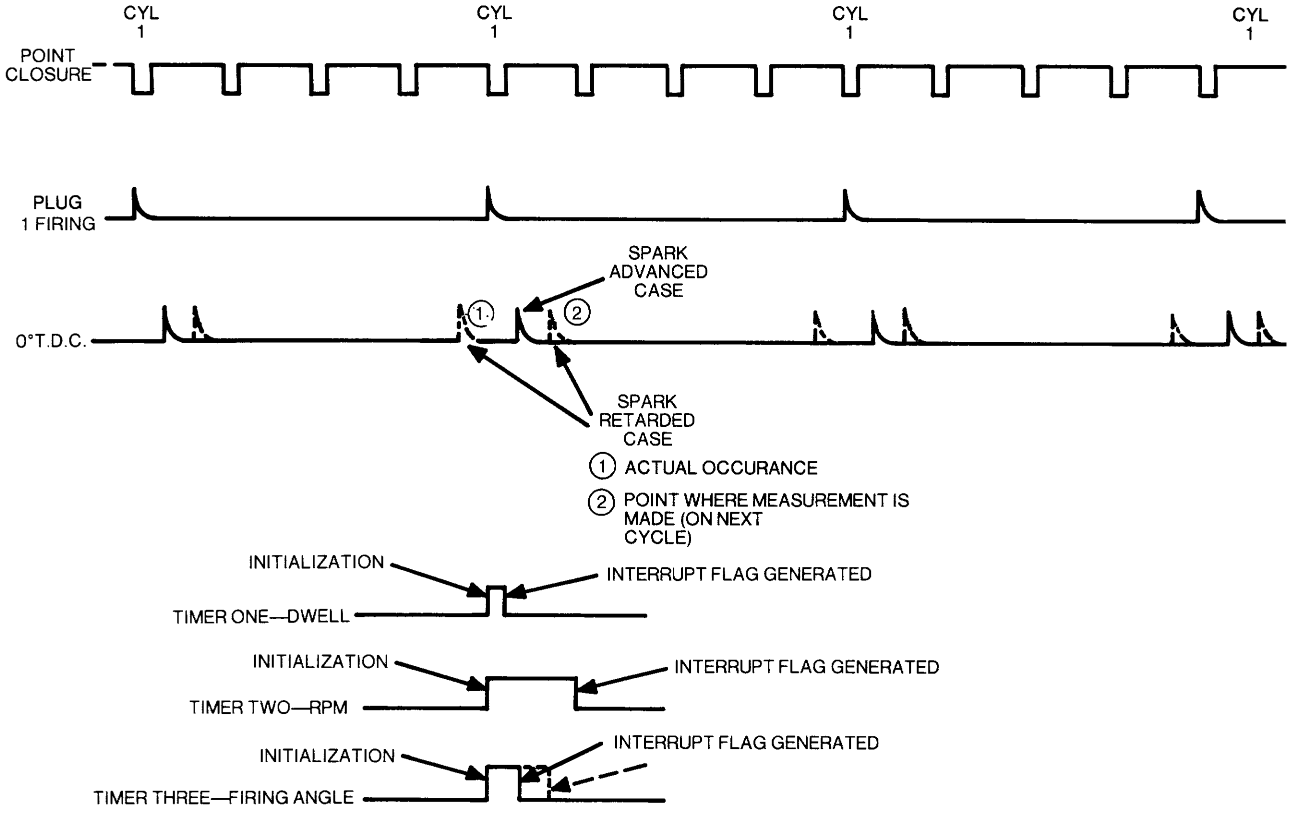


Figure 5-12 Four Cylinder Four Cycle Engine

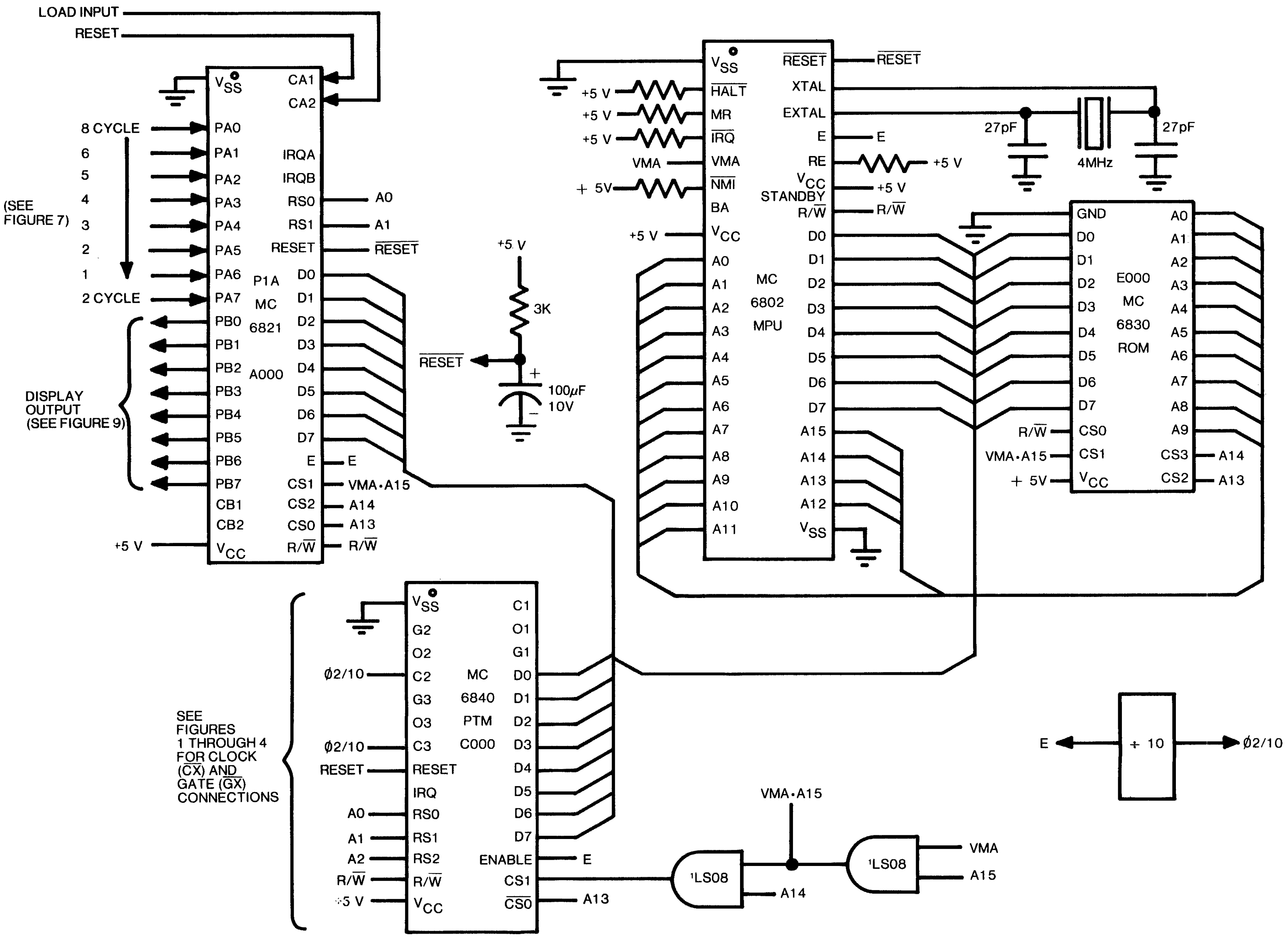


Figure 5-13 Engine Analyzer System Interconnection and Decoding

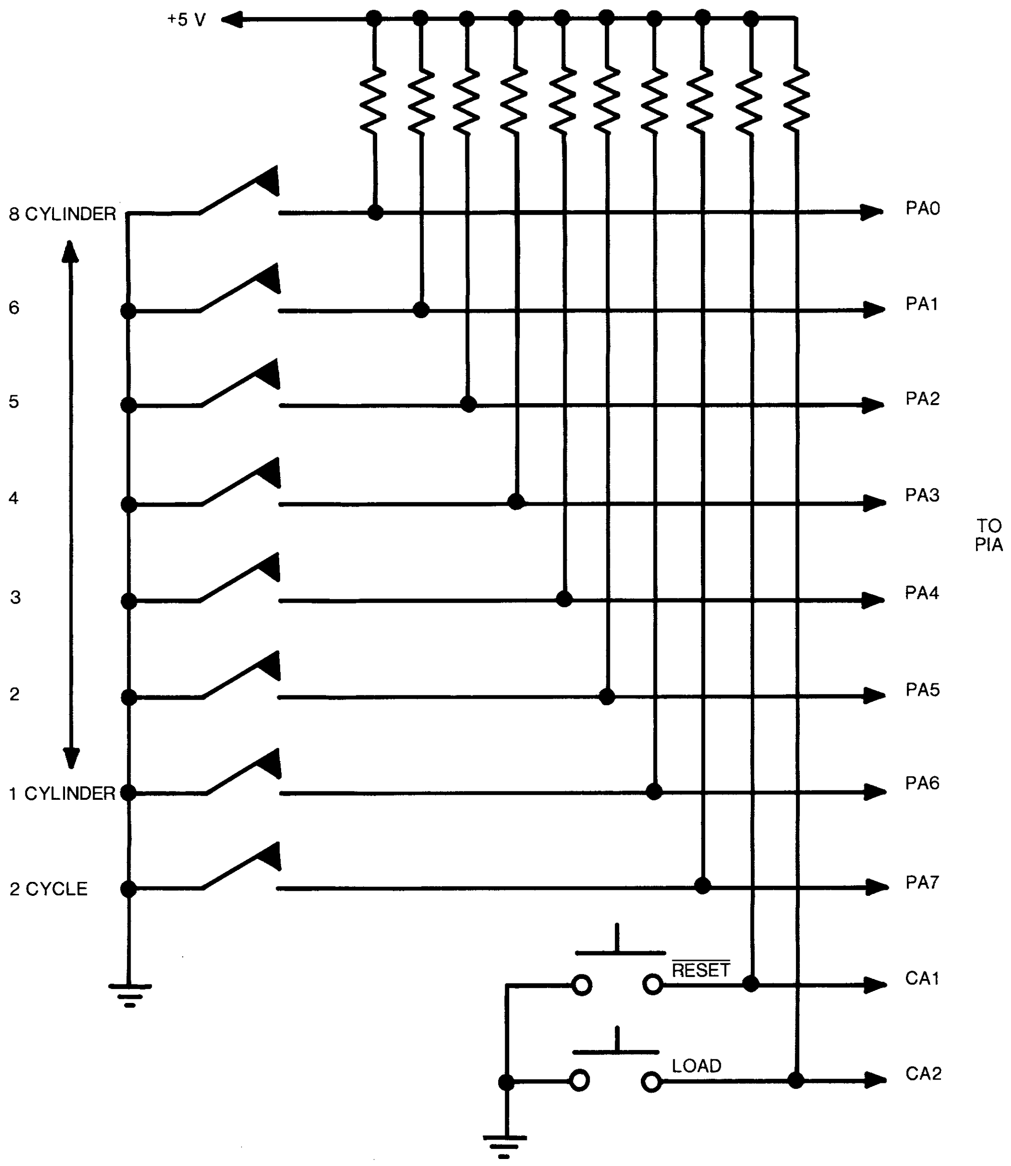


Figure 5-14 Input Controls

5.6 MANUAL INPUT HARDWARE

The user must manually input the umber of cylinders on the engine under test and whether it is a two or four cycle engine. Figure 5-14 shows the basic electrical approach to inputting the engine information. A rotary switch to select the number of cylinders is probably the least expensive approach. Push buttons with a "lockout" function (to prevent two buttons from being depressed) are easier to use, but more expensive. The analyzer will be in 4 cycle mode unless the 2 cycle switch is activated. The "load" button allows the transfer of the data throgh a PIA (MC6821) to the MPU. Pressing the reset button stops the analyzer operation and allows the input to be changed.

5.7 DISPLAY HARDWARE

Figure 5-16 shows the display system. Eight lines are brought out through the "B" side of the PIA to the display circuitry. Four of the lines are used to decode the Binary Coded Decimal input into buffered seven segment output drivers (active low). The remaining four lines are decoded and buffered to sequentially select each Light Emitting Diode display. A display routine in the MPU handles the multiplexing of the display.

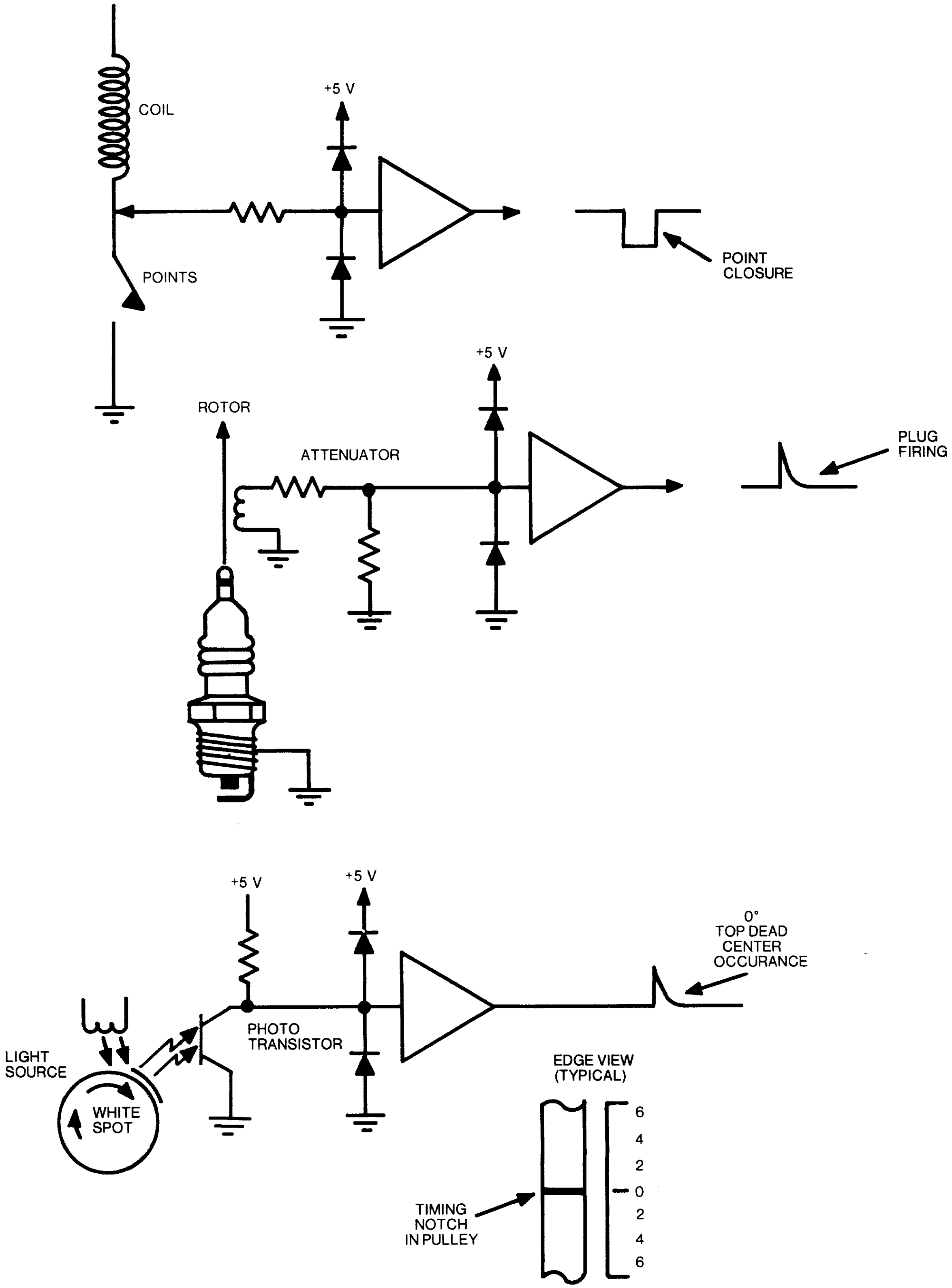


Figure 5-15 Signal Conditions

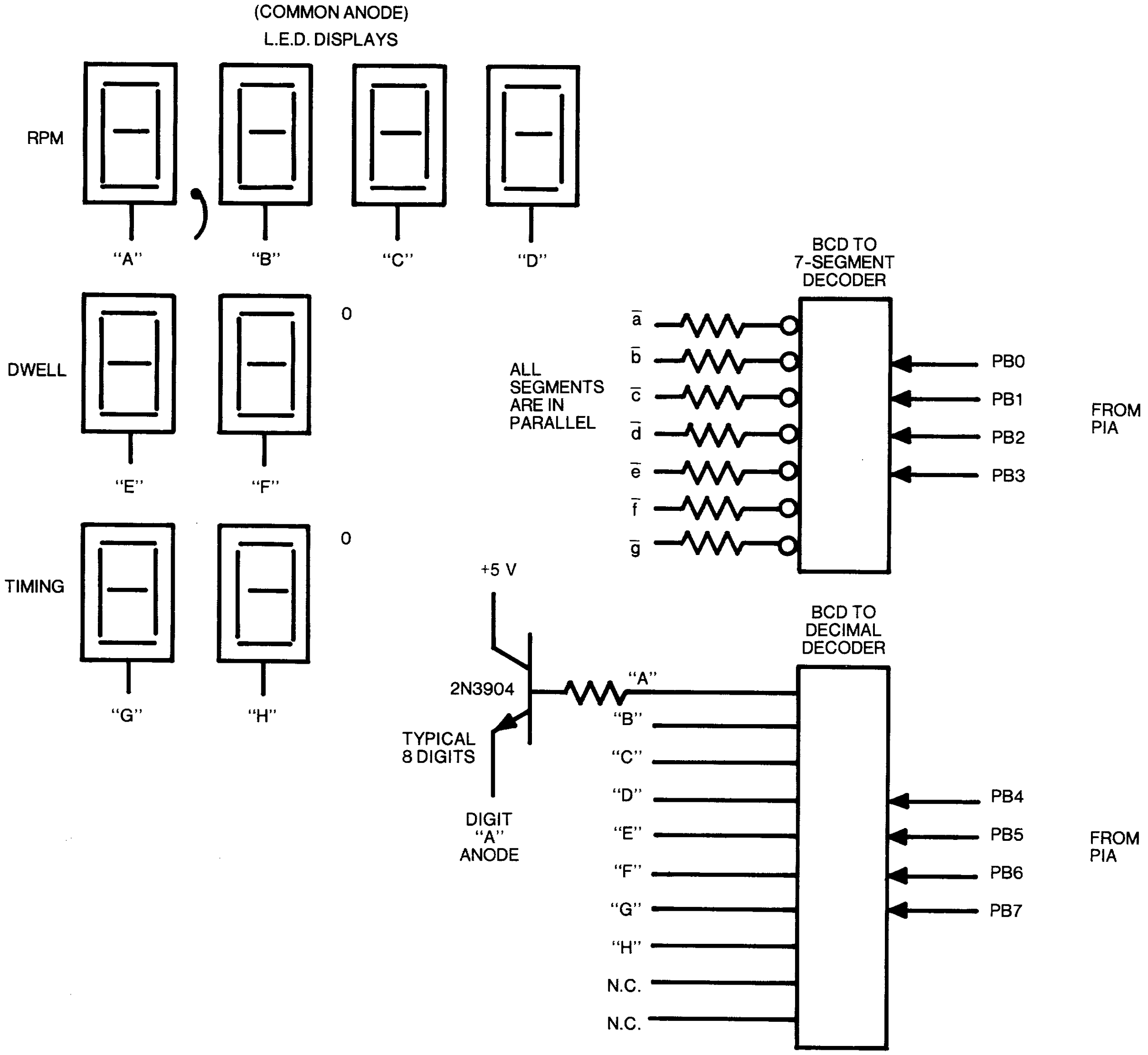


Figure 5-16 Display System

5.8 MPU INTERCONNECTION

The schematic shown in Figure 5-13 describes the interconnection of the MPU, Programmable Timer, ROM (Read Only Memory) and PIA (Peripheral Interface Adapter). One low ower schottky quad AND gate (74LS08) is used for address decoding. The MC6830 ROM is located at address $E000 and contains the program shown on the flow chart in Figure 13. The program will use the 128 bytes of RAM located in the MC6802 MPU as a scratchpad area where data is stored until needed for display or a further calculation. The MC6802 also contains a non-board oscillator and an MC6800 MPU. The MC6820 PIA is located at address $A000 and provides two interfacing ports to the MPU. One port allows the pushbutton information to be entered into the MPU's program. The other port outputs the display information to the L.E.D. circuitry. The PTM, which is located at address $C000 is initially conditioned by the MPU. It then receives the three TTL conditioned signals from the engine, computes pulse and period intervals, and signals the MPU when done. The MPU, which has been running the output dipslays during this time, then retrieves its new information and calculates new display values.

5.9 OPERATION

The flowchart in Figure 5-17 describes the basic software opration of the engine analyzer:

5.9.1 Power On Reset

When the power is switched on, the power up reset circuit initiatizes the program.

5.9.2 Input Engine Type Information

The MPU first conditions the PIA for proper operation, PA0-7 are programmed to be inputs and PB0-7 are programmed to be outputs. The program will then wait until the "LOAD INPUT" button is pushed. This sets a bit in the PIA control register. When the "LOAD INPUT" button has been pushed the information entered by the button is loaded and stored for future use by the program.

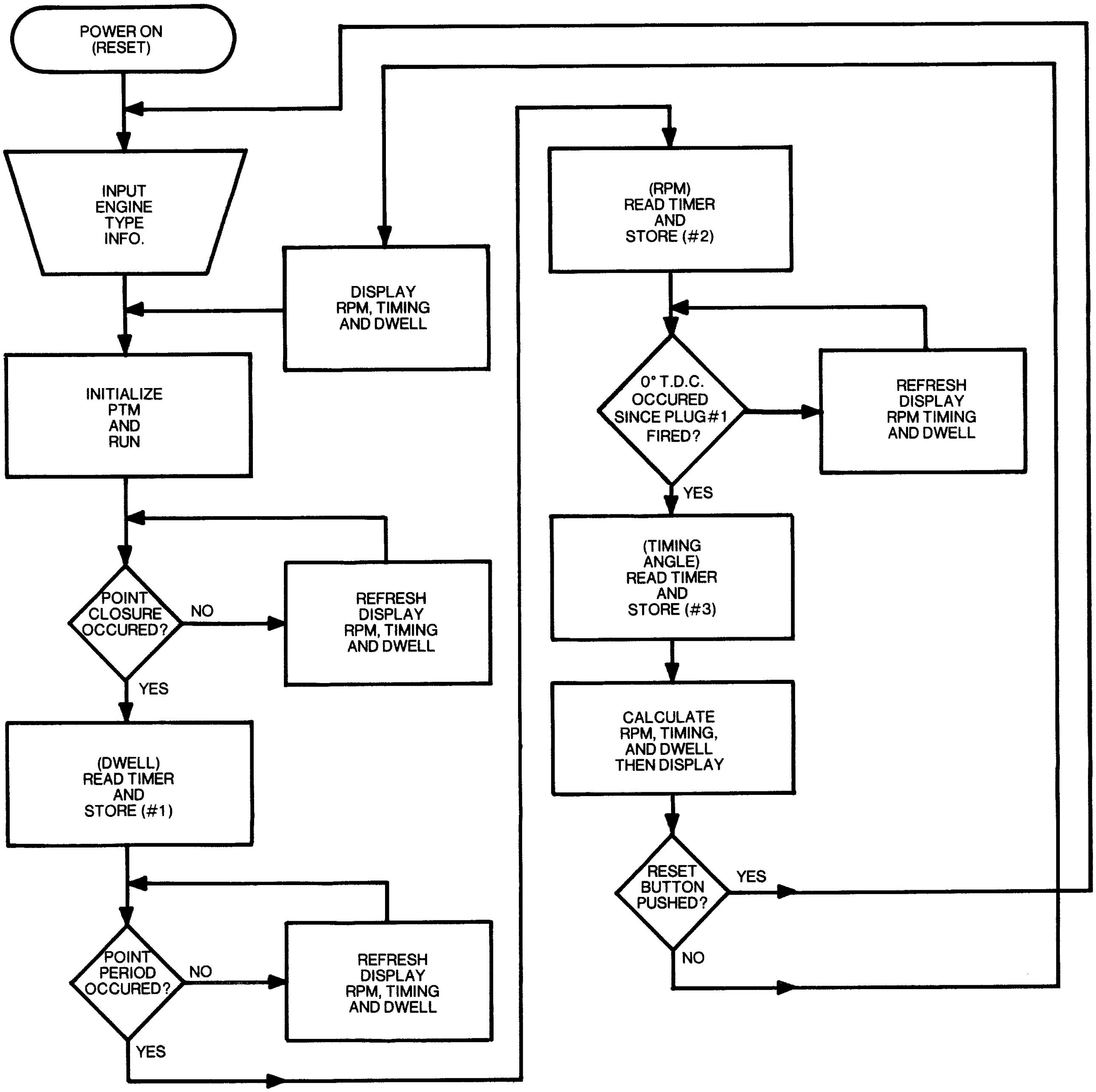

POWER ON (RESET)
INPUT ENGINE TYPE INFO.
DISPLAY RPM, TIMING AND DWELL
INITIALIZE PTM AND RUN
POINT CLOSURE OCCURED?
NO
REFRESH DISPLAY RPM, TIMING AND DWELL
YES
(DWELL) READ TIMER AND STORE (#1)
POINT PERIOD OCCURED?
NO
REFRESH DISPLAY RPM, TIMING AND DWELL
YES
(RPM) READ TIMER AND STORE (#2)
0° T.D.C. OCCURED SINCE PLUG#1 FIRED?
REFRESH DISPLAY RPM TIMING AND DWELL
YES
(TIMING ANGLE) READ TIMER AND STORE (#3)
CALCULATE RPM, TIMING, AND DWELL THEN DISPLAY
RESET BUTTON PUSHED?
YES
NO

Figure 5-17 Engine Analyzer Flow Chart

5.9.3 Initialize the PTM and Run

The PTM is now initialized and programmed by the MPU. The three timers are each measuring an engine interval as shown on Figures 4-8 through 4-11. Each timer can be loaded with a time out word which it will count down to zero — from $0001 (decimal 1) to $FFFF (decimal 65,536). If **no** time-out is loaded, the timer defaults (assumes) a word of $FFFF. This is very convenient to the software, and is how it functions in this application. All three measurements are parallel and completed on a single revolution of the engine. The timers are programmed in reverse order, as shown below.

Timer 3 — Timing Angle

Control Register 3 is loaded with the control word:
$38

This causes the following operation (see Figure 5-11):

1. The timer to start counting down when plug #1 fires.
2. To decrement one count every 10 microseconds.
3. To stop counting when 0° top dead center occurs.
4. Signal the MPU that the count is ready to be read.

Timer 2 — RPM

Control Register 2 is loaded with the control word:
$09

This causes the following operation (See Figure 5-10)

1. The timer to start counting down when the points close.
2. To decrement one count every 10 microseconds.
3. To stop counting when the points have closed a second time.
4. Signal the MPU that the count is ready to be read.

Timer 1 — Dwell

Control Register 1 is loaded with the control word:
$3A

This causes the following operation (see Figure 5-9)

1. The timer to start counting down when the points close.
2. To decrement one count every 1 microsecond.
3. To stop counting when the points have opened.
4. Signal the MPU that the count is ready to be read.

Timer 1 also contains the "operate" bit, so when the control word for timer 1 is written, all three timers start operating simultaneoulsy.

5.9.4 Point Closure

The MPU refreshes the display with the previously calculated engine parameters while waiting for the Timer #1 interrupt bit to signal that the points have opened.

5.9.5 Read Timer and Store (Dwell)

When the interrupt bit goes high, the MPU reads Timer #1 counter and stores the information in its internal RAM (Random Access Memory).

NOTE: $ = hexadecimal number

5.9.6 Point Period

The MPU operates in the same manner as in "Point Closure."

The #2 interrupt occurs on the second closure of the points.

5.9.7 Read Timer and Store (RPM)

The MPU stores the information from Timer #2's counter.

5.9.8 0° T.D.C. Occured Since Plug #1 Fired?

Again, the MPU refreshes the display while waiting for the interrupt from Timer #3.

5.9.9 Read Timer and Store (Timing Angle)

The MPU stores the information from Timer #3 counter.

5.9.10 Calculate

The MPU calculates the RPM, Timing and Dwell at this point. Figure 5-18 contains the basic operations to be performed by the MPU on these results. The MPU then formats and stores the information to be displayed. The MPU will be refreshing the display more than 70% of the time, due to the comparitive slowness of the engine, even at high speed. The operating range of the MPU—PTM Engine Analyzer is from 92RPM to 16000 RPM.

5.9.11 Reset Button Pushed?

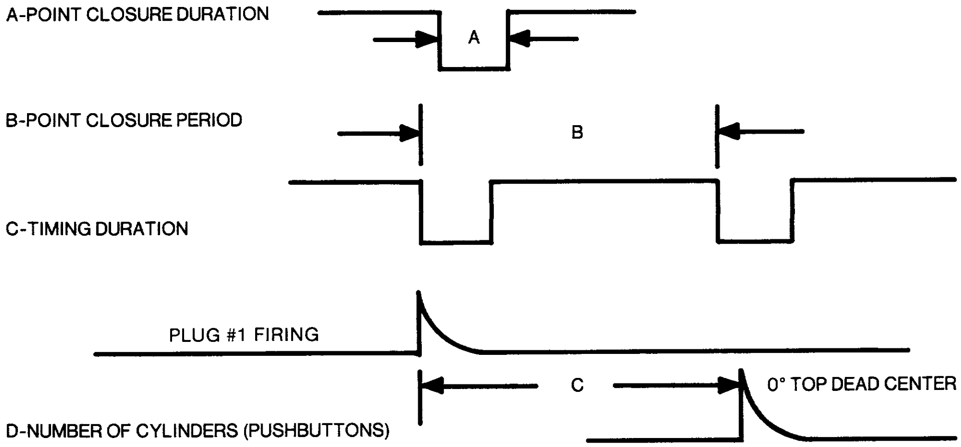
The MPU polls the interrupts of the PIA to determine if the "Reset" button has been pushed. If so, another "Load Input" must be performed. If not, the program refreshes the display, and initiatizes the PTM for the next engine RPM.

5.10 POWER

Power may be derived from the +12 V battery and a DC to DC converter to obtain +5 V. A three pin regulator capable of an output of 5 V @ 1A should be adequate, but the particular power requirements of the final circuit should be examined.

5.11 SUMMARY

The MC6840 MPU-PTM Controlled Engine Analyzer provides the serious do-it-yourselfer and service stations with professional results and few of the difficulties of analog equipment.



D-NUMBER OF CYLINDERS (PUSHBUTTONS)

E-CYCLES OF ENGINE (PUSHBUTTONS)

$\text{RPM} = (30*E)/(D*B)$

$\text{DWELL} (°) = (360*A)/(D*B)$

$\text{TIMING ANGLE} = (180*C*E)/(D*B)$ RETARDED SPARK

IF RESULT OF ABOVE $\geqq$ 100° THEN:

$\{[(180*C*E)/(D*B)] - 360\}$ ADVANCED SPARK(+)

CONVERT HEXADECIMAL RESULTS TO BCD AND FORMAT
FOR DISPLAYS

Figure 5-18 Software Calculations

MOTOROLA ***Semiconductor Products Inc.***

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

A12004-4 PRINTED IN USA 10-82 IMPERIAL LITHO C16700 5,000