Example 2: *Write a program to generate a square wave of 1KHz frequency on OUT 1 pin of 8253/54. Assume CLK1 frequency is 1MHz and address for control register = 0BH, counter 1 = 09H and counter 2 = 0AH.*

Sol. : To get square wave mode 3 is selected count should be $\frac{1\,\text{MHz}}{1\,\text{KHz}} = 1000$

Control word :

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
SC_1	SC_2	RW_1	RW_0	M_2	M_1	M_0	BCD	
0	1	1	1	0	1	1	1	= 77H

Source program

```
MOV  AL,77H
OUT  0BH,AL        ; Loads control word (77H) in
                   ; the control register.
MOV  AL,00H        ; loads lower byte (00) of the count
OUT  09H,AL
MOV  AL,10         ; Loads higher byte (10) of the count
OUT  09H,AL
```

9.7 Interfacing of 8253/54 with 8086

9.7.1 With 8-Bit Address

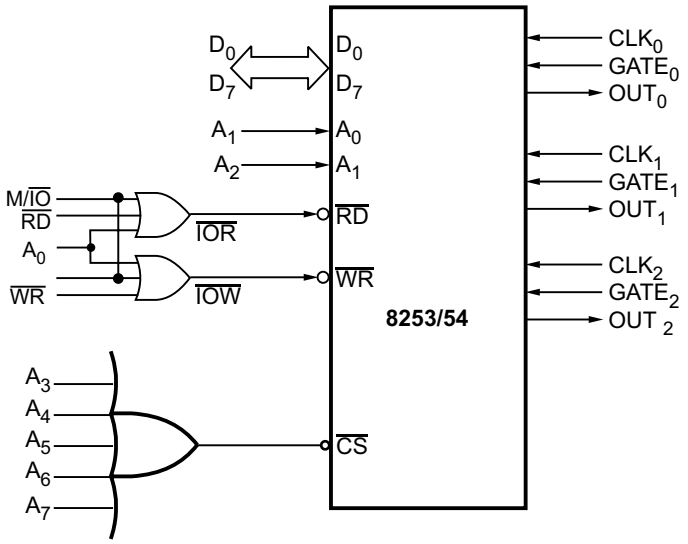


Fig. 9.12 Interfacing of 8253/54 with 8-bit address

We know that, 8253/54 has two address and eight data lines. Therefore, it is necessary to interface lower byte of demultiplexed data bus to 8253/54. The address lines A_1 and A_2 are connected to the the address lines of 8253/54. The 8253/54 IC decodes A_1 and A_2 lines internally to select one of its ports or control register. The remaining address lines (A_7-A_3) can be used to generate chip select signal. Fig. 9.12 (see Fig. on previous page) shows the interfacing of 8253/54 with 8086.

Address Map :

Ports / control Register	Address lines								Address
	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
Counter 0	0	0	0	0	0	0	0	0	00H
Counter 1	0	0	0	0	0	0	1	0	02H
Counter 2	0	0	0	0	0	1	0	0	04H
Control Register	0	0	0	0	0	1	1	0	06H

9.7.2 With 16-Bit Address

Fig. 9.13 shows the interfacing of 8253/54 with 8086 with 16-bit address. Here $\overline{RD}$ and $\overline{WR}$ signals are activated when M/$\overline{IO}$ signal is low, indicating I/O bus cycle. To get absolute address, all remaining address lines (A_3 - A_{15}) are used to decode the address for 8253/54.

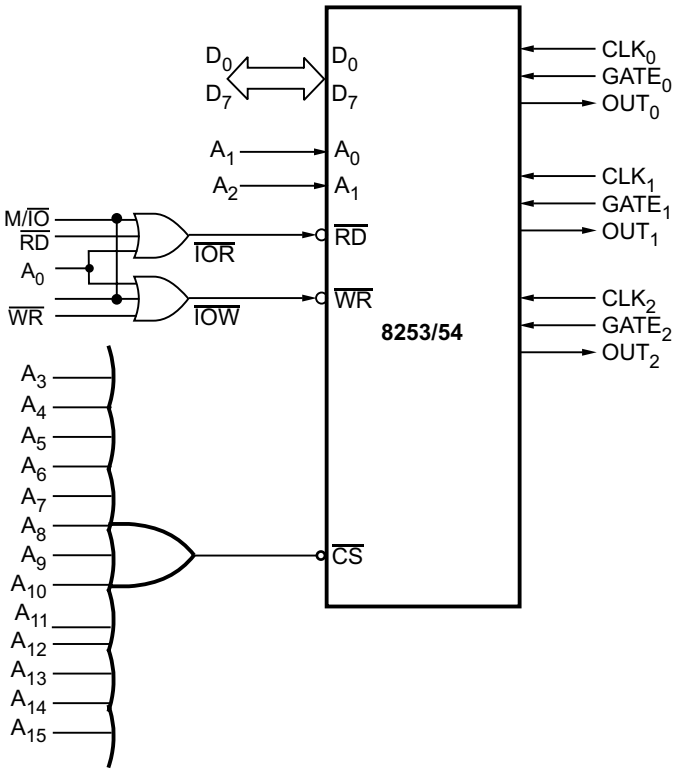


Fig. 9.13 Interfacing 8253/54 with 16-bit address

Note : When 16-bit address is used it is necessary to use indirect addressing mode to access 8253/8254.

Address Map :

Ports/Control Register	Address lines																Address
	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
Counter 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
Counter 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0002H
Counter 2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0004H
Control Register	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0006H

9.8 Interfacing of 8253/54 with 8088

9.8.1 With 8-Bit Address

We know that, 8253/54 has two address and eight data lines. Therefore, it is necessary to interface lower byte of demultiplexed data bus to 8253/54. The address lines A_0 and A_1 are connected to the address lines of 8253/54. The 8253/54 IC decodes A_0 and A_1 lines internally to select one of its ports or control register. The remaining address lines (A_7-A_2) can be used to generate chip select signal. Fig. 9.14 shows the interfacing of 8253/54 with 8086.

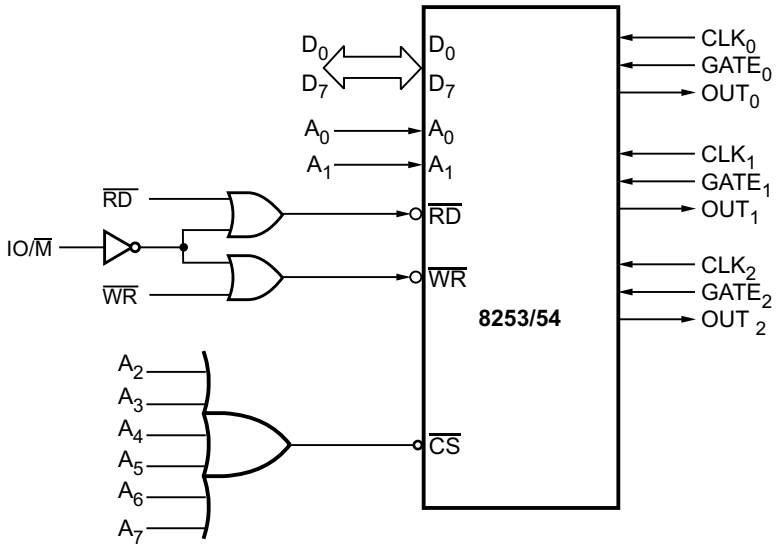


Fig. 9.14 Interfacing of 8253/54 with 8-bit address

Address map :

Ports / control register	Address lines								Address
	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
Counter 0	0	0	0	0	0	0	0	0	00H
Counter 1	0	0	0	0	0	0	0	1	01H
Counter 2	0	0	0	0	0	0	1	0	02H
Control Register	0	0	0	0	0	0	1	1	03H

9.8.2 With 16-Bit Address

Fig. 9.15 shows the interfacing of 8253/54 with 8086 with 16-bit address. Here $\overline{RD}$ and $\overline{WR}$ signals are activated when IO/$\overline{M}$ signal is high, indicating I/O bus cycle. To get absolute address, all remaining address lines (A_2 - A_{15}) are used to decode the address for 8253/54.

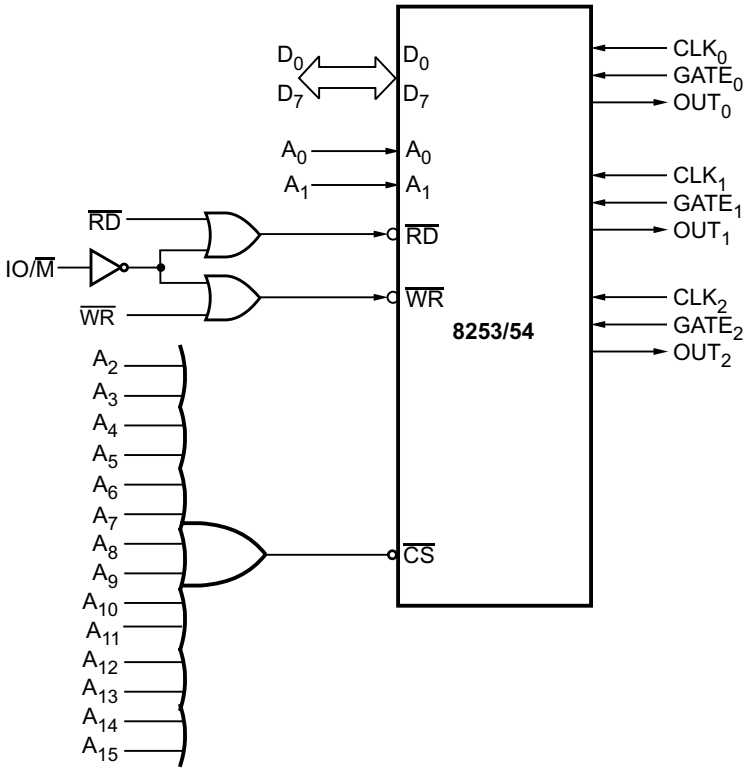


Fig. 9.15 Interfacing of 8253/54 with 8-bit address

Note : When 16-bit address is used it is necessary to use indirect addressing mode to access 8253/8254.

Address map :

Ports/Control Register	Address lines																Address
	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
Counter 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
Counter 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0001H
Counter 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0002H
Control Register	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0003H

9.9 Interfacing 8253/54 to 8086 in Memory Mapped I/O

In this type of I/O interfacing the 8086 uses 20 address lines to identify an I/O device, an I/O device is connected as if it is a memory register. The 8086 uses same control signals and instructions to access I/O as those of memory. Fig. 9.16 shows the interfacing of 8253/54 with 8086 in memory mapped I/O technique. Here $\overline{RD}$ and $\overline{WR}$ signals are activated when M/$\overline{IO}$ signal is high, indicating memory bus cycle. The A_1 - A_2 address lines are used by 8253/54 for internal decoding. To get absolute address, all remaining address lines (A_3 - A_{19}) are used to decode the address for 8255. Other signal connections are same as in I/O mapped I/O.

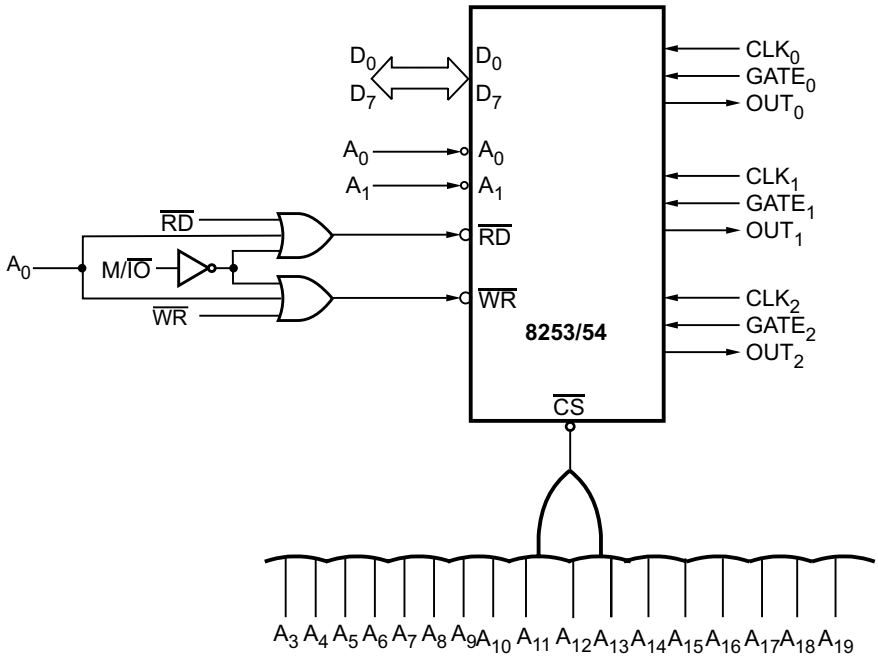


Fig. 9.16 Interfacing 8253/54 with 8086 in memory mapped I/O

I/O map :

Register	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Address
Counter 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
Counter 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0002H
Counter 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0004H
Control register	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0006H

9.10 Interfacing 8253/54 to 8088 in Memory Mapped I/O

Here, IO/$\overline{M}$ signal of 8088 is used to indicate memory bus cycle. When this signal low 8253/54 is activated, for read or write bus cycle. Address lines A_0 - A_1 are used for internal decoding of 8253/54 and remaining address lines (A_2 - A_{19}) are used to generate chip select signal for 8253/54 to get absolute address.

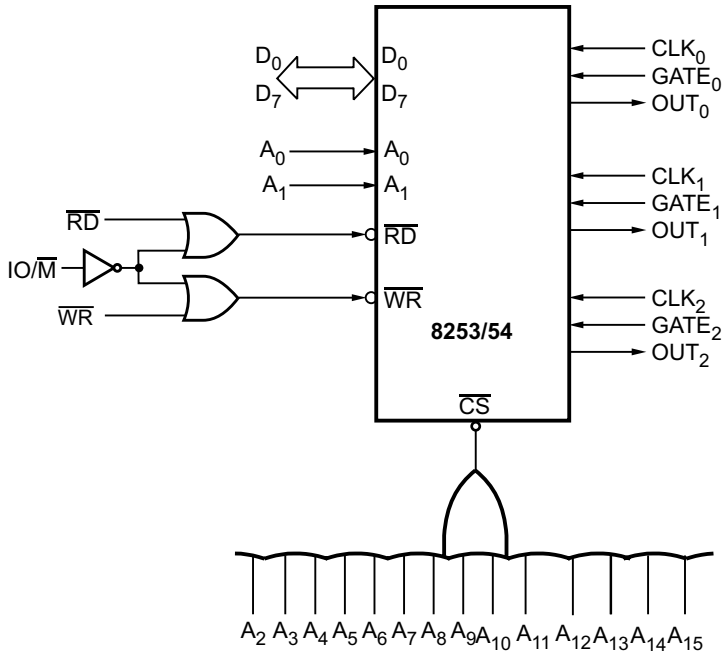


Fig. 9.17 Interfacing 8253/54 with 8088 in memory mapped I/O

I/O map :

Register	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Address
Counter 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
Counter 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0001H
Counter 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0002H
Control register	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		1	1	0003H

9.11 Applications

9.11.1 Pre-Settable Alarm System

Program Statement : Design a pre-settable alarm system using 8253/54 timer. Use thumbwheel switches to accept 4 digit value in seconds. Alarm should last for 5 seconds. Do not use interrupt.

Sol. : Fig. 9.18 shows the 8086 microprocessor based pre-settable alarm system. Thumbwheel switches are interfaced through 8255 ports. Timing parameters are derived from the 8253/54. 74LS138 decoder is used to generate chip select signals for 8253/54 and 8255. One more 74LS138 decoder is used to generate $\overline{IOR}$, $\overline{IOW}$, $\overline{MEMR}$, and $\overline{MEMW}$ signals.

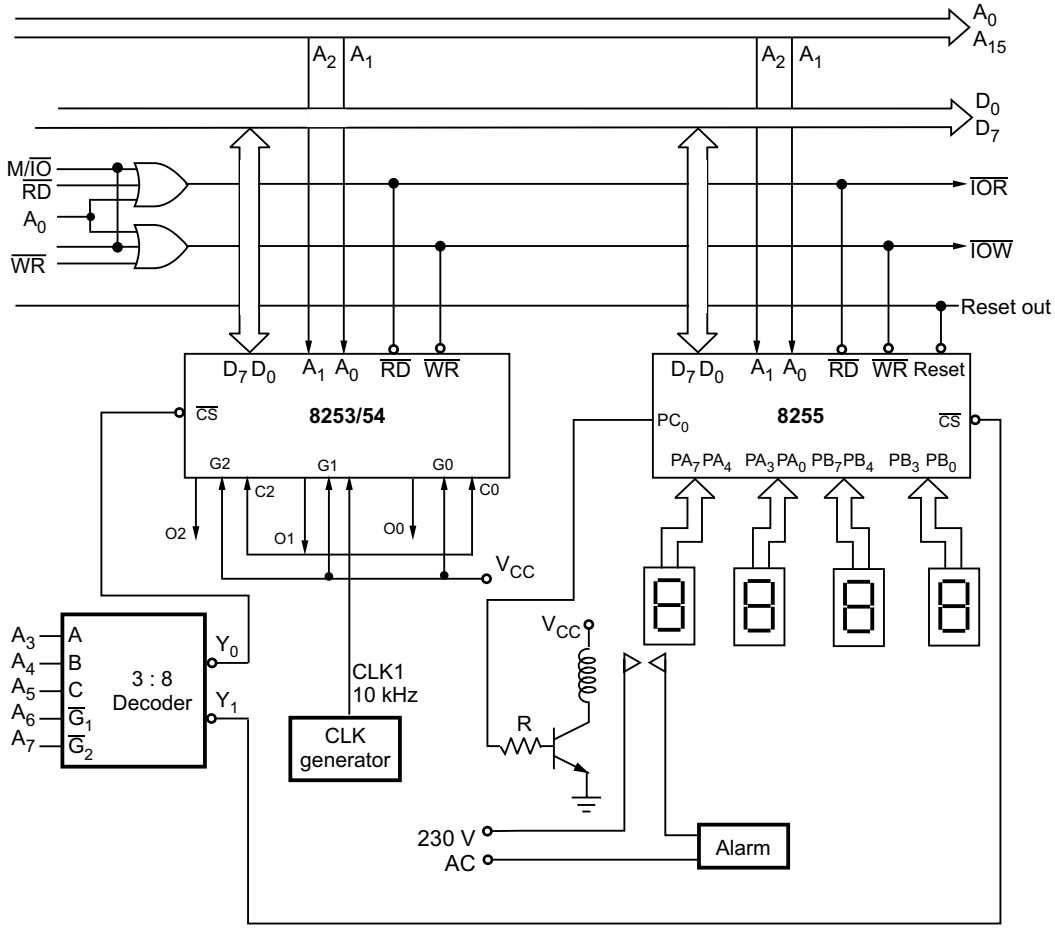


Fig. 9.18 Pre-settable alarm system

Counter 0 of 8253/54 is programmed in mode 0 to give the pre-settable time period and counter 2 is programmed in mode 0 to give delay of 5 seconds. As we know, clock input for 8253/54 is 10 kHZ count required to get 1 second time interval is

$$\text{Count} = \frac{\text{Required Period}}{\text{Clock Period}} = \frac{1\ \text{sec}}{100\ s} = 10000 = 1\ \ 10^4 = 2710\text{H}$$

This count value is loaded in the count register of the counter 1 and counter 1 is programmed in mode 2 to generate square wave with frequency 1 Hz. The output of counter 1 is fed to the clock input of counter 0 and counter 2.

To read four digit of count, we need four thumbwheels. One thumbwheel switch can be interfaced using four input lines. So to interface four thumbwheels we need 16 lines. The IC 8255 is used to interface these thumbwheel switches. Two thumbewheel switches are connected to port A and other two are connected to port B.

Address Map :

Ports / Control Register	Address lines								Address
	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
Counter 0	0	0	0	0	0	0	0	0	00H
Counter 1	0	0	0	0	0	0	1	0	02H
Counter 2	0	0	0	0	0	1	0	0	04H
Control register	0	0	0	0	0	1	1	0	06H
Port A	0	0	0	0	1	0	0	0	08H
Port B	0	0	0	0	1	0	1	0	0AH
Port C	0	0	0	0	1	1	0	0	0CH
Control register	0	0	0	0	1	1	1	0	0EH

Control Word to Read/Write value in count register of counter 0 of 8253/54

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
SC_1	SC_2	RW_1	RW_0	M_2	M_1	M_0	BCD	
0	0	1	1	0	0	0	1	= 31H

D_0 = 1 BCD Count
D_1, D_2, D_3 = 000 Mode : Square wave rate generator
D_4, D_5 = 11 Read/ Write lower byte first and then higher byte.
D_6, D_7 = 00 Counter 0

Control Word to Read/Write value in count register of counter 1 of 8253/54

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
SC_1	SC_2	RW_1	RW_0	M_2	M_1	M_0	BCD	
0	1	1	1	0	0	0	1	= 71H

D_0 = 1 BCD Count
D_1, D_2, D_3 = 000 Mode : Square wave rate generator
D_4, D_5 = 11 Read/Write lower byte first and then higher byte.
D_6, D_7 = 01 Counter 1

Control Word to latch value in count register of counter 0 of 8253/54

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
SC_1	**SC_2**	**RW_1**	**RW_0**	**M_2**	**M_1**	**M_0**	**BCD**	
0	0	0	0	0	0	0	1	= 01H

D_0 = 1 BCD Count
D_1, D_2, D_3 = 000 Mode : Square wave rate generator
D_4, D_5 = 00 Read/ Write lower byte first and then higher byte.
D_6, D_7 = 00 Counter 0

Control Word to latch value in count register of counter 1 of 8253/54

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
SC_1	**SC_2**	**RW_1**	**RW_0**	**M_2**	**M_1**	**M_0**	**BCD**	
0	1	0	0	0	0	0	1	= 41H

D_0 = 1 BCD Count
D_1, D_2, D_3 = 000 Mode : Square wave rate generator
D_4, D_5 = 00 Read/ Write lower byte first and then higher byte.
D_6, D_7 = 01 Counter 1

Control Word to initialise 8255 : Port A = I/P, Port B =I/P, Port C = O/P

BSR/IO	MODE A		PA	PC_H	MODE B	PB	PC_L	
1	0	0	1	0	0	1	0	= 92H

Program:

```
        MOV  AL,92H       ; Load control word in accumulator
        OUT  0EH,AL       ; Initialise 8255 by sending control
                          ; word at the
                          ; address of control register
START:  IN   AL,08H       ; Get the lower two digit of the count
        MOV  BL,AL        ; Store the lower  two digit of
                          ; the count
        IN   AL,0AH       ; Get the higher two digit of the count
        MOV  BH,AL        ; Store the higher two digit of
```

```
                        ; the count
        MOV  AL,31H
        OUT  06,AL      ; Loads control word (31H) in the control
                        ; register to
                        ; load 16-bit count in the count register
                        ; of counter 0
        MOV  AL,BL
        OUT  00,AL      ; Loads lower byte of the count.
        MOV  AL,BH
        OUT  00,AL      ; Loads higher byte of the count.
BACK:   MOV  AL,01H
        OUT  06,AL      ; Loads control word (01H) in the control
                        ; register to
                        ; latch 16-bit count in the count
                        ; register of counter 0
        IN   AL,00H     ; Get the lower two digits of the count
        CMP  AL,00H     ; Compare with zero
        JNZ  BACK       ; Repeat
        IN   AL,00H     ; Get the higher two digits of the count
        CMP  AL,00H     ; Compare with zero
        JNZ  BACK       ; Repeat
        MOV  AL,01H     ; Load bit pattern to run Alarm
        OUT  0CH,AL     ; Send it to Port C
        CALL DELAY      ; Wait for 5 Seconds
        MOV  AL,00H     ; Load bit pattern to stop Alarm
        OUT  0CH,AL     ; Send it to Port C
        JMP  START      ; Repeat
```

Delay Routine :

This delay routine gives a delay of 5 seconds. Counter1 of 8253/54 is used to give delay of 1 second. As output of counter1 is used as a clock for counter2, the count in the counter2 acts as a multiplying factor. Therefore, by loading 05H in the count register of counter2 we get a delay of 5 (5 1) seconds.

```
        MOV  AL,71H
        OUT  06,AL      ; Loads control word (71H)in the
                        ; control register.
        MOV  AL,10H
        OUT  02,AL      ; Loads lower byte of the count.
        MOV  AL,27H     ; Loads higher byte of the count.
```

```
        OUT   02H
        MOV   AL,B1H
        OUT   06H,AL
        MOV   AL,05H
        OUT   04,AL        ; Loads lower byte of the count.
        MOV   AL,00H       ; Loads higher byte of the count.
        OUT   04,AL
BACK:   MOV   AL,81H
        OUT   06,AL        ; Loads control word (81H) in the
                           ; control register
                           ; to latch 16-bit count in the count
                           ; register of
                           ; counter 2
        IN    AL,04H       ; Get the lower two digits of the count
        CMP   AL,00        ; Compare with 00H
        JNZ   BACK         ; If not zero, Repeat
        IN    AL,04H       ; Get the higher two digits of the count
        CMP   AL,00        ; Compare with 00H
        JNZ   BACK         ; If not zero, Repeat
        RET                ; Return to main program
```

9.11.2 DC Motor Speed and Direction Control

Let us see the application of 8254 timer as a DC motor speed controller. Fig. 9.19 shows the schematic diagram of the DC motor and its associated driver circuitry. As shown in the Fig .9.19 driver circuitry consists of IC 8254, a flip-flop, buffers and transistors. (See Fig on next page)

When flip-flop outputs are : Q = 0 and $\overline{Q}$ = 1, inverters 1 and 3 give output logic 0 and inverters 2 and 4 give output logic 1. As a result, transistors Q_1 and Q_4 turn ON and transistors Q_2 and Q_3 turn OFF. Due to this V_{CC}, supply is applied to the positive lead of the DC motor and ground is applied to the negative lead of the DC motor. This connection causes DC motor to rotate in the forward direction.

When flip-flop outputs are : Q = 1 and $\overline{Q}$ = 0, the conditions of transistors are reversed and DC motor rotate in the reverse direction. Thus the direction of motor can be controlled by changing state of the flip-flop.

If the duty cycle of the flip-flop output is kept 50%, then motor current is positive and negative for equal amount of time. If flip-flop output is varied with enough frequency then due to inertia of motor, motor will not rotate at all. But it will exhibit

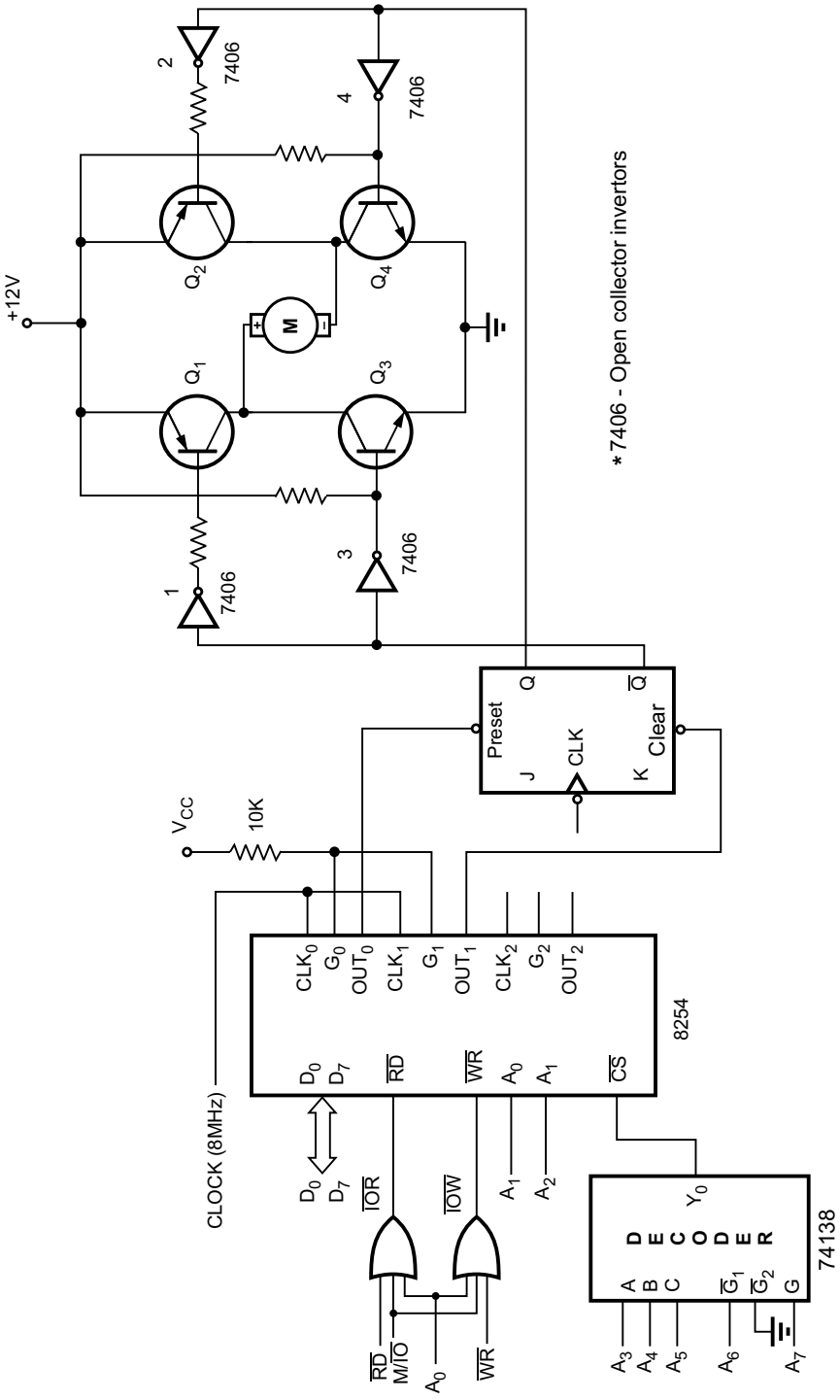
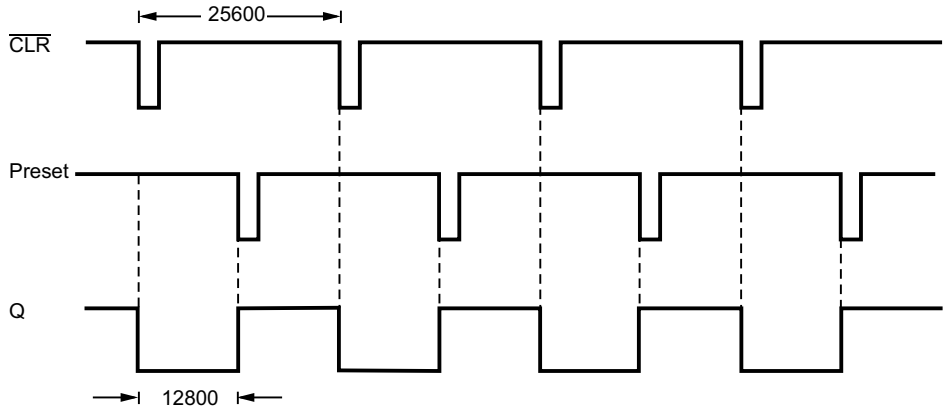
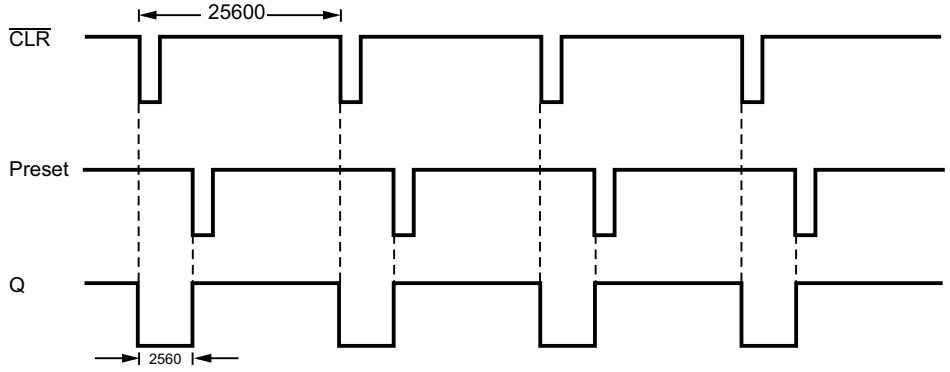


Fig. 9.19 DC motor speed and direction control using the 8254 timer

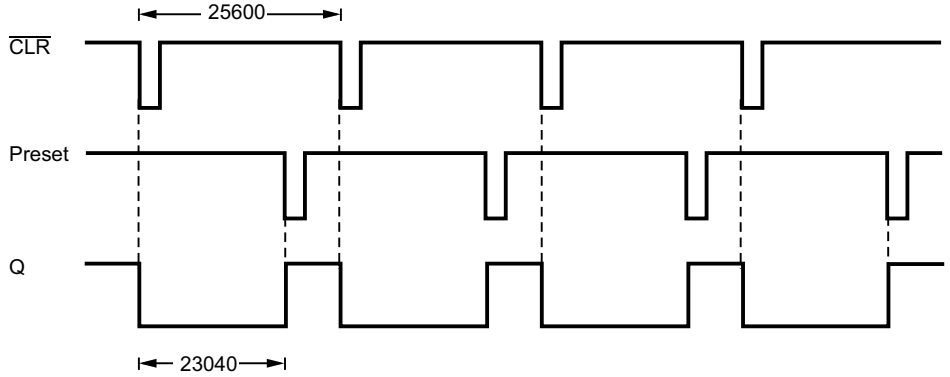
some holding torque because of the current flowing through it. Fig. 9.20 shows some timing diagrams and their effects on the speed and direction of the motor.



(a) No rotation



(b) High speed rotation in the reverse direction



(c) High speed rotation in the forward direction

Fig. 9.20

The variable duty cycle is achieved by programming two counters, counter 0 and counter 1 of 8254. The counter 0 and counter 1 are both programmed to divide the input clock by 25,600. The duty cycle now can be changed by changing the point at which counter 0 is started in relationship to counter 1. The clock input is divided by each counter by 25,600. This produces the basic operating frequency for the motor and as 25,600 is divisible by 256 a short program given below allows 256 different speeds.

```
;  Procedure to control speed and direction of DC motor
;  Addresses for 8254 are :
;  Counter 0 : 80H
;  Counter 1 : 82H
;  Counter 2 : 84H
;  Counter 3 : 86H
;  Control word for counter 0 : 00110100 B,
;                   counter 1 : 01110100 B
   COUNT EQU 25600
       PUSH AX          ; save register
       PUSH BX
       MOV BL,100
       MUL BL           ; Multiply by 100
       MOV BX,AX
       MOV AX,COUNT     ; Subtract result of
       SUB AX,BX        ; multiplication from COUNT
       MOV BX,AX
       MOV AL,01110100B ;  [ Load control word
       OUT 86H,AL       ;     for counter 1 ]
       MOV AX,COUNT     ;  [ Load counter 1
       OUT 82H,AL       ;     register with
       MOV AL,AH        ;     Count =
       OUT 82H,AL       ;     25600 ]
       MOV AL,11000010B ;  [ Send the command word
       OUT 82H,AL       ;     to latch the count using
                        ;     Read back command ]
AGAIN: IN AL,82H        ;  [ Read the
       MOV AH,AL        ;  Count
       IN AL,82H        ;  of
       XCHG AL,AH       ;  counter 1 ]
       CMP AX,BX        ;  Compare it with calculated
```

```
                        ;   count
JB  AGAIN               ;  If  not  equal  read  again
MOV  AL,00110100B       ;   [  Load  control  word  for
OUT  86H,AL             ;      Counter  1  ]
MOV  AX,COUNT           ;   [  Load  counter  0
OUT  80H,AL             ;      register  with
MOV  AL,AH              ;      count  =
OUT  80H,AL             ;      25600  ]
POP  BX                 ;   Restore  registers
POP  AX
RET
```

As shown in the program, value passed in AL register decides the direction and speed of the DC motor. If value in AL is 80H then motor does not rotate. However, when value in AL approaches 00H, the motor rotates with increase in speed in forward direction. On the other hand, when value in AL approaches FFH, the motor rotates with increase in speed in reverse direction.

The time interval between counter 1 and 0 is calculated in terms of count. This is accomplished by multiplying value in AL by 100 and then subtracting it from 25600. This is required because the counters are down-counters that count from the programmed count to 0, before restarting.

Initially, counter 1 is started with count 25600. It is read and compared with the calculated cound until it is equal. Once it reaches to the calculated count, counter 0 is started with count 25600. From this point forward, both counters continue generating clear and preset pulses until the procedure is called again to adjust the speed and direction of the DC motor.

Review Questions

1. *What is the necessity of the programmable interval timer?*
2. *List the features of any programmable interval timer.*
3. *List the differences between 8253 and 8254.*
4. *Draw and explain the functional block diagram of IC 8253/54.*
5. *Illustrate different modes of operations of 8253/54.*
6. *Give the control word format for 8253/54.*
7. *Using IC 8253, realise a square wave generator with 1 msec period if the input frequency to 8253 is 1MHz.*
8. *Design a monostable multivibrator to obtain a pulse width of 5 msec with 8253 using an external clock of 100 kHz.*

University Questions

1. *Explain the working of 8254 with block diagram.* **(VTU : Sept. 2000)**
2. *Write a short note on : D.C. motor speed control using microprocessor.* **(VTU : Sept. 2000)**
3. *The 8254 timer/counter is a 24 pin chip with the following pins. Show how it can be connected to 8088 (in I/O mapped I/O) to have the counter address located at 00H, 01H, 02H and the control word register addressed located at 03H.*

Data lines	*8*
$\overline{rd}$, $\overline{wr}$, $\overline{cs}$	*3*
3 sets of (CLK, gate, out pins)	*3 3 = 9*
A_1, A_0	*2*
V_{CC}, *GND*	*2*
Total	*24*

(VTU: August 2001)

❑❑❑