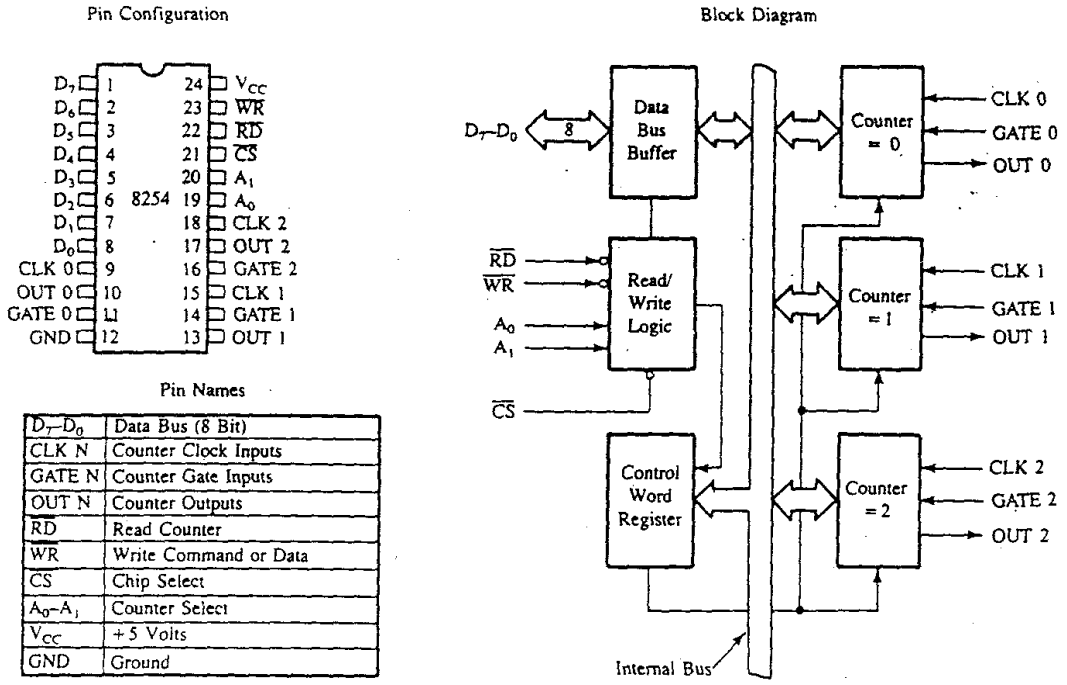
# THE 8254 PROGRAMMABLE INTERVAL TIMER (PIT)

The 8254 programmable interval timer/counter is used to generate accurate time delays and can be used for applications such as a real-time clock, an event counter, a digital one-shot, a square-wave generator, and a complex waveform generator.
The 8254 includes three identical 16-bit counters that can operate independently in any one of the six modes (to be described later). It is packaged in a 24-pin DIP and requires a single +5 V power supply. To operate a counter, a 16-bit count is loaded in its register and, on command, begins to decrement the count until it reaches 0. At the end of the count it generates a pulse that can be used to interrupt the MPU. The counter can count either in binary or BCD. In addition, a count can be read by the MPU while the counter is decrementing.

## 7.1 Block Diagram of the 8254

Figure 7.1 is the block diagram of the 8254; it includes three counters (0, 1, and 2), a data bus buffer. Read/Write control logic, and a control register. Each counter has two input signals—Clock (CLK) and GATE—and one output signal—OUT.




Pin Names

| | |
|---|---|
| $D_7$–$D_0$ | Data Bus (8 Bit) |
| CLK N | Counter Clock Inputs |
| GATE N | Counter Gate Inputs |
| OUT N | Counter Outputs |
| $\overline{RD}$ | Read Counter |
| $\overline{WR}$ | Write Command or Data |
| $\overline{CS}$ | Chip Select |
| $A_0$–$A_1$ | Counter Select |
| $V_{CC}$ | +5 Volts |
| GND | Ground |

*FIGURE 7.1:*

*8254 Block Diagram*

DATA BUS BUFFER
This tri-state, 8-bit, bidirectional buffer is connected to the data bus of the MPU.

CONTROL LOGIC
The control section has five signals: RD (Read), WR (Write), CS (Chip Select) and address lines A0,A1.

The control word register and counters are selected according to the signals on lines A0 and A1, as shown below:

| $A_1$ | $A_0$ | Selection |
|---|---|---|
| 0 | 0 | Counter 0 |
| 0 | 1 | Counter 1 |
| 1 | 0 | Counter 2 |
| 1 | 1 | Control Register |

CONTROL WORD REGISTER

This register is accessed when lines A0 and A1 are at logic 1. It is used to write a command word which specifies the counter to be used, its mode, and either a Read or a Write operation. The control word format is shown in Figure 7.2.

| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|---|---|
| SC1 | SC0 | RW1 | RW0 | M2 | M1 | M0 | BCD |

SC—Select Counter:

| SC1 | SC0 | |
|---|---|---|
| 0 | 0 | Select Counter 0 |
| 0 | 1 | Select Counter 1 |
| 1 | 0 | Select Counter 2 |
| 1 | 1 | Read-Back Command (See Read Operations) |

M—MODE:

| M2 | M1 | M0 | |
|---|---|---|---|
| 0 | 0 | 0 | Mode 0 |
| 0 | 0 | 1 | Mode 1 |
| X | 1 | 0 | Mode 2 |
| X | 1 | 1 | Mode 3 |
| 1 | 0 | 0 | Mode 4 |
| 1 | 0 | 1 | Mode 5 |

RW—Read/Write:

| RW1 | RW0 | |
|---|---|---|
| 0 | 0 | Counter Latch Command (see Read Operations) |
| 0 | 1 | Read/Write least significant byte only. |
| 1 | 0 | Read/Write most significant byte only. |
| 1 | 1 | Read/Write least significant byte first, then most significant byte. |

BCD:

| | |
|---|---|
| 0 | Binary Counter 16-bits |
| 1 | Binary Coded Decimal (BCD) Counter (4 Decades) |

Note: Don't Care Bits (X) Should Be 0 to Ensure Compatability with Future Intel Products.

*FIGURE 7.2:*

*8254 Control Word Format*

MODES

The 8254 can operate in six different modes, and the gate of a counter is used either to disable or enable counting, as shown in Figure 7.3. However, to maintain clarity, only one mode (Mode 0) is illustrated first, and details of the remaining modes are discussed in a coming section.

In Mode 0, after the count is written and if the gate is high, the count is decremented every clock cycle. When the count reaches zero; the output goes high and remains high until a new count or mode word is loaded.

## 7.2 Programming the 8254

The 8254 can be programmed to provide various types of output through Write operations, or to check a count while counting through Read operations. The details of these operations are given below.

| Modes \ Signal Status | Low or Going Low | Rising | High |
|---|---|---|---|
| 0 | Disables counting | — | Enables counting |
| 1 | — | (1) Initiates counting<br>(2) Resets output after next clock | — |
| 2 | (1) Disables counting<br>(2) Sets output immediately high | (1) Reloads counter<br>(2) Initiates counting | Enables counting |
| 3 | (1) Disables counting<br>(2) Sets output immediately high | Initiates counting | Enables counting |
| 4 | Disables counting | — | Enables counting |
| 5 | — | Initiates counting | — |

*FIGURE 7.3:*

*Gate Settings of a Counter*

WRITE OPERATIONS

To initialize a counter, the following steps are necessary.

1. Write a control word into the control register.
2. Load the low-order byte of a count in the counter register.
3. Load the high-order byte of a count in the counter register.

With a clock and an appropriate gate signal to one of the counters, the above steps should start the counter and provide appropriate output according to the control word.

READ OPERATIONS

In some applications, especially in event counters, it is necessary to read the value of the count in progress. This can be done by either of two methods. One method involves reading a count after inhibiting (stopping) the counter to be read. The, second method involves reading a count while the count is in progress (known as reading on the fly).

In the first method, counting is stopped (or inhibited) by controlling the gate input or the clock input of the selected counter, and two IO read operations are performed by the MPU. The first IO operation reads the low-order byte, and the second IO operation reads the high-order byte.

In the second method, an appropriate control word is written into the control register to latch a count in the output latch, and two IO Read operations are performed by the MPU. These Read/Write operations are illustrated below.

## 7.3 Illustration: The 8254 as a Counter

PROBLEM STATEMENT

1. Identify the port addresses of the control register and counter 2 in Figure 7.4.
2. Write a subroutine to initialize counter 2 in Mode 0 with a count of 50,000. The subroutine should also include reading counts on the fly; when the count reaches zero, it should return to the main program.
3. Write a main program to display seconds by calling the subroutine as many times as necessary.

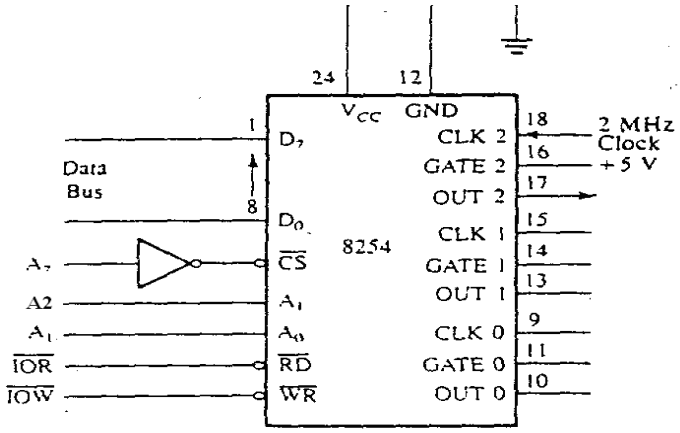



*FIGURE 7.4:*

*Schematic: Interfacing the 8254*



1. Port Addresses

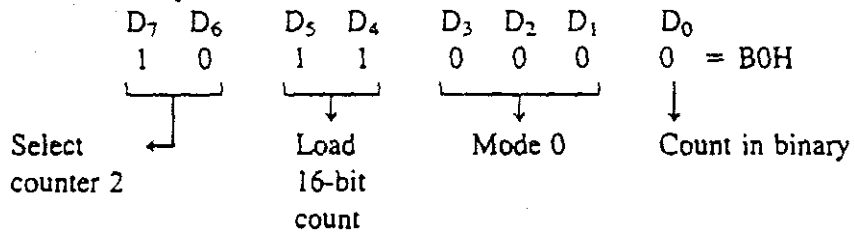
The Chip Select is enabled when A7 = 1, and the control register is selected when A1 and A2 = 1. Similarly, counter 2 is selected when A2 = 1 and A1 = 0. Assuming that the unused address lines A5 to A3 are at logic 0, the port addresses will be as follows:

| | |
|---|---|
| counter2 | 84H |
| control register | 86H |

2. Subroutine Counter

To initialize the 8254 for counter 2 in Mode 0, the following control word is necessary:

*Control Word* (Load Operation)




Control Word for Latching: Bits $D_5$ and $D_4$ should be 0 = 80H

Count = 20,000 = 4E20H;

Subroutine

```
CNTR2     EQU   84H
CNREG     EQU   86H
CNWRD1    EQU   0B0H
CNWRD2    EQU   80H
LOBYTE    EQU   20H
HIBYTE    EQU   4EH

COUNTER:  PUSH  AX
          MOV   AL, CNWRD1   ;Control word to initialize counter 2
          OUT   CNREG, AL    ;Write in the control registe
          MOV   AL, LOBYTE   ;Low-order byte of the count 50000
          OUT   CNTR2, AL    ;Load counter 2 with the low-order byte
          MOV   AL,HIBYTE    ;High-order byte of the count 50000
```

```
        MOV     CNTR2, AL     ;Load counter 2 with the high-order byte
READ:   MOV     AL, CNWRD2    ;Control word to latch a count
        OUT     CNREG, AL     ;Write in the control register
        IN      AL, CNTR2     ;Read low-order byte
        MOV     AH, AL        ;Store low-order byte
        IN      AL, CNTR2     ;Read high-order byte
        OR      AX, AX        ;OR low- and High-order bytes to set Z flag
        JNZ     READ          ;If counter ≠ 0, go back to read next count
        POP     AX
        RET
```

*Subroutine Description*

The subroutine has two segments. In the first segment, counter 2 is initialized by writing a control word in the control register and a 16-bit count specified as LOBYTE and HIBYTE in the counter register. The hexadecimal value equivalent to
50000 must be calculated.

In the second segment (beginning at READ), a control word is written into the control register to sample a count, and the 16-bit count is read by performing two input operations. The reading of the counter is repeated until the counter reaches 0: the Zero flag is checked by ORing the read count.

2. Main Program

The subroutine COUNTER provides 25 ms (50000 X 0.5 micro .s Clock) delay; if this routine is called 40 times, the total delay will be one second.

```
        XOR     AL, AL      ; clear register AL to save # seconds
SECONDS MOV     CX, 40      ; set up register CX with 40
WAIT:   CALL    COUNTER     ; wait for 25ms
        LOOP    WAIT
        INC     AL          ; increment second counter
        AAA                 ; decimal adjust the count
        OUT     PORT1, AL   ; display the count on PORT1
        JMP     SECONDS     ; go back and start counting the next sec.
```

Program Description

The main program loads register CX with the count of 40 and sets up the WAIT loop. The loop calls the COUNI'ER subroutine 40 times to generate a one-second delay. At the end of the loop, it increments the seconds in register AL, decimal-adjusts the byte, and displays seconds. The sequence is repeated until register AL reaches 99 BCD. After the 99th second, register AL is cleared and the clock sequence is repeated.

This program is just to demonstrate the Read and Write operations of the 8254; this clock design does not take into account the errors caused by the delay in executing the program instructions. A better way of designing a real-time clock is to interrupt the MPU at the end of a count (see Problem 15 at the end of this chapter).

## 7.4 Modes

As mentioned earlier, the 8254 can operate in six different modes: we already illustrated Mode 0. Now we will describe briefly various modes of the 8254 including Mode 0.

MODE 0: INTERRUPT ON TERMINAL COUNT

In this mode, initially the OUT is low. Once a count is loaded in the register, the counter is decremented every cycle, and when the count reaches zero, the OUT goes high. This can be used as an interrupt. The OIJ'T remains high until a new count or a command word is loaded. Figure 15.27 also shows that the counting (m = 5) is temporarily stopped when the Gate is disabled (G = 0). and continued again when the Gate is at logic 1.

MODE 1: HARDWARE-RETRIGGERABLE ONE-SHOT

In this mode, the OUT is initially high. When the Gate is triggered, the OUT goes low, and at the end of the count, the OUT goes high again, thus generating a one-shot pulse (Figure7.5, Mode 1).

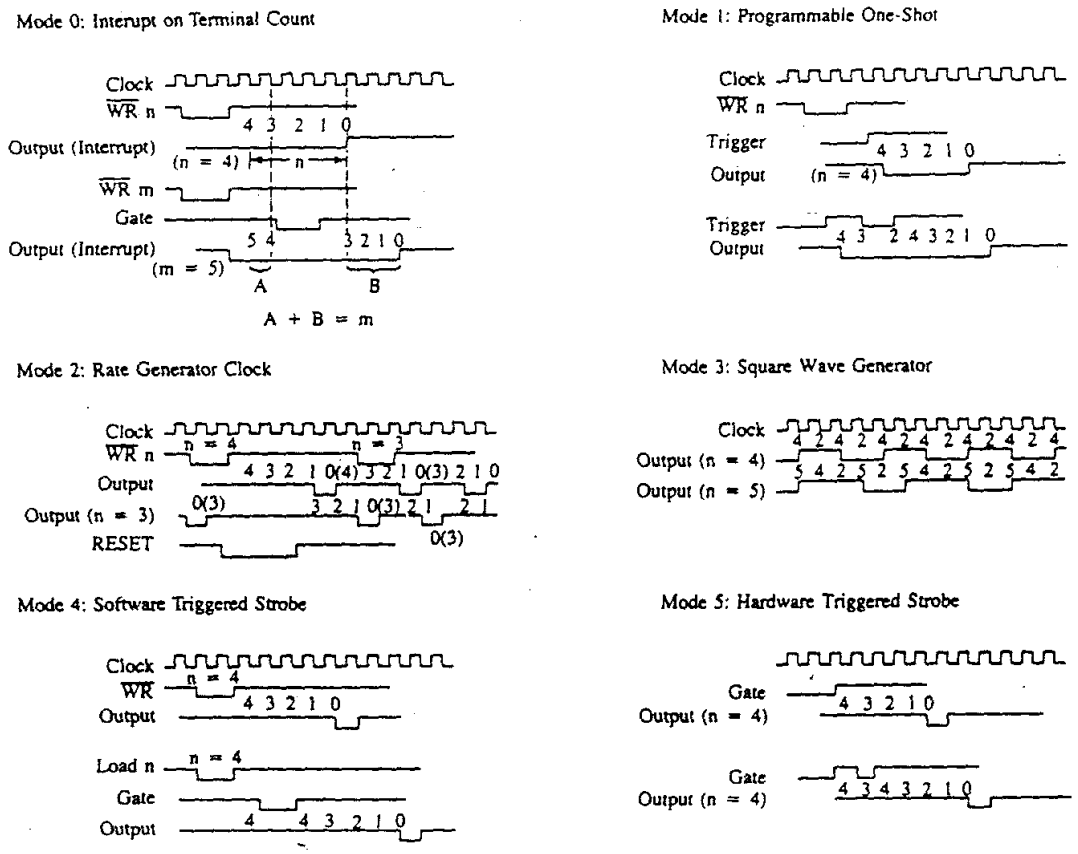



*FIGURE 7.5:*

*Six Modes of the 8254*

MODE 2: RATE GENERATOR

This mode is used to generate a pulse equal to the clock period at a given interval. When a count is loaded, the OUT stays high until the count reaches 1, and then the OUT goes low for one clock period. The count is reloaded automatically, and the pulse is generated continuously. The count = 1 is illegal in this mode.

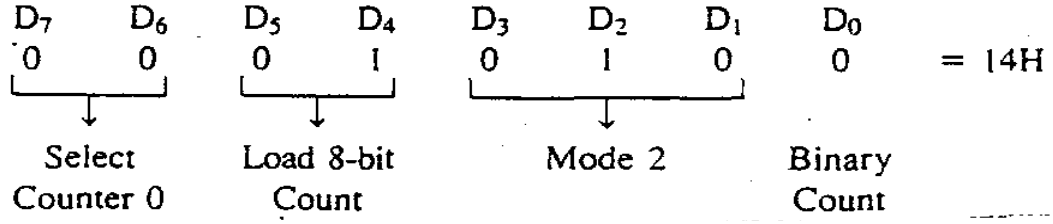
## Example

Write instructions to generate a pulse every 50 µs from Counter 0 (Refer to Figure7.4).

Solution

To generate a pulse every 50 µs, from Counter 0, it should be initialized in Mode 2 (Refer to Figure 7.5 for modes), and the Gate 0 should be high.

**Control Word**




**Count:** In Mode 2, the count is decremented every clock period, and at the last count, the counter generates a pulse equivalent to the clock period of the timer. Here the clock frequency of the 8254 is 2 MHz (0.5 μs clock period), and the pulse should be generated every 50 μs. Therefore, the count is calculated as follows:

$$\text{Count} = \frac{50 \times 10^{-6}}{0.5 \times 10^{-6}} = 100 = 64\text{H}$$

In this example, the frequency of the pulse is 20 kHz (1/50 μs). This count can also be calculated by dividing the clock frequency by the frequency of the pulse (2 MHz/ 20 kHz = 100).

**Instructions:**

```
MOV   AL, 14H   ;Control word Mode 2 and Counter 0
OUT   86H, AL   ;Write in 8254 control register
MOV   AL, 64H   ;Low-order byte of the count
OUT   80H,AL    ;Load Counter 0 with low-order byte
HLT
```

MODE 3: SQUARE-WAVE GENERATOR

In this mode, when a count is loaded, the OUT is high. The count is decremented by two at every clock cycle, and when it reaches zero, the OUT goes low, and the count is reloaded again-. This is repeated continuously; thus, a continuous square wave with period equal to the period of the count is generated. In other words, the frequency of the square wave is equal to the frequency f the clock divided by the count. If the count (N) is odd, the pulse stays high for (N + 1)12 clock cycles and stays low for (N — 1)/2 clock cycles.

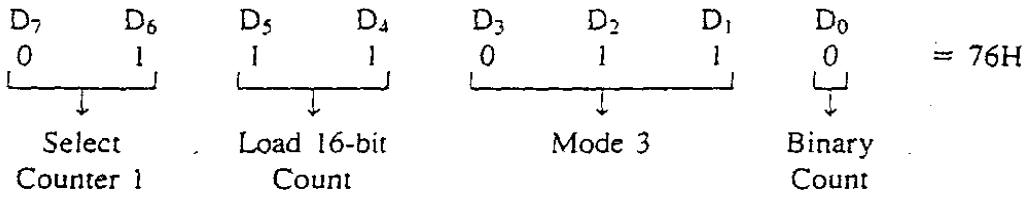
## Example

Write instructions to generate a 1 kHz square wave from Counter 1 (Refer to Figure7.4). Assume the gate of Counter 1 is tied to +5 V through a 10 k resistor. Explain the significance of connecting the gate to +5 V.

*Solution*

To generate a square wave from Counter 1, it should be initialized in Mode 3 (Refer to Figure 15.27 for modes.)

**Control Word**



**Count:** In Mode 3, the count is decremented by two for every clock period. If the count is N, N/2 clock pulses provide the upper half of the square wave. The count is loaded again and N/2 clock pulses provide the lower half.

In this example, the clock frequency of the 8254 is 2 Mhz (0.5 μs clock period), and the square wave frequency is 1 kHz (1 ms clock period). Therefore, we need a count for 1 ms delay.

$$\text{Count} = \frac{1 \times 10^{-3}}{0.5 \times 10^{-6}} = 2000 = 07D0H$$

This count can also be calculated by dividing the clock frequency by the square wave frequency (2 MHz/1 kHz = 2000).

**Instructions:**

```
MOV   AL, 76H    ; control word Mode 3 and counter 1
OUT   86H, AL    ;Write in 8254 control register
MOV   AL, 0D0H   ;Low-order byte of the count
OUT   81H, AL    ; Load counter 1 with low order byte
MOV   AL, 0D0H   ; High order byte of the count
OUT   81H, AL    ; Load counter 1 with high order byte
HLT
```

To run Counter 1, the gate of that counter must be tied high; otherwise the counter action is inhibited.

MODE 4: SOFTWARE-TRIGGERED STROBE

In this mode, the OUT is initially high; it goes low for one clock period at the end of the count. The count must be reloaded for subsequent outputs.

MODE 5: HARD WARE-TRIGGERED STROBE

This mode is similar to Mode 4, except that it is triggered by the rising pulse at the gate. Initially, the OUT is low, and when the Gate pulse is triggered from low to high, the count begins. At the end of the count, the OUT goes low for one clock period.

READ-BACK COMMAND

The Read-Back Command in the 8254 allows the user to read the count and the status of the counter; this command is not available in the 8253. The format of the command is shown in Figure 7.6(a).

The command is written in the control register, and the count of the specified counter(s) can be latched if COUNT (bit D5) is 0. A counter or a combination of counters is specified by keeping the respective CNT bits (D1, D2, and D3) high. For example, the control word 1 1 0 1 0 1 1 0 (D6H) written in the control register will latch the counts of Counter 0 and Counter 1, and these counts can be obtained by reading respective counter port addresses. The latched counts are held until they are read or the counters are reprogrammed. The Read-Back Command eliminates the need of writing separate counter-latch commands for different counters.

The status of the counter(s) can be read if STATUS bit (D4) of the Read-Back Command is low. Figure 7.6(b) shows the format of the status byte.

(a) Read-Back Command Format

| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | $\overline{\text{COUNT}}$ | $\overline{\text{STATUS}}$ | CNT 2 | CNT 1 | CNT 0 | 0 |

$D_5$: 0 = Latch Count of Selected Counter(s)
$D_4$: 0 = Latch Status of Selected Counter(s)
$D_3$: 1 = Select Counter 2
$D_2$: 1 = Select Counter 1
$D_1$: 1 = Select Counter 0
$D_0$: Reserved for Future Expansion; Must Be 0

$A_0, A_1$ = 11
$\overline{CS}$ = 0
$\overline{RD}$ = 1
$\overline{WR}$ = 0

(b) Status Byte

| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|---|---|
| OUTPUT | NULL COUNT | RW1 | RW0 | M2 | M1 | M0 | BCD |

$D_7$: 1 = Out Pin is 1
: 0 = Out Pin is 0
$D_6$: 1 = Null Count
: 0 = Count Available for Reading
$D_5$-$D_0$: Counter Programmed Mode

*FIGURE 7.6:*
*Read-8ackComrnand Format (a) and Status Byte (b);*

## Example

Write a subroutine to generate an interrupt every 1 sec. Refer to Figure 7.2 for counter addresses.

*Solution*

The clock frequency shown in Figure 7.2 is 2 MHz; thus, a count is decremented every 0.5 µs. To obtain a delay of one second, the count should be (1 sec/0.5 X $10^{-6}$) 2 Meg. This count is too large for one 16-bit counter. We can divide this count, as an example; 50,000 for Counter 1 and 40 for second Counter 2 (50 k X 40 = 2 Meg.). If we-set up Counter 1 in Mode 2 with 50,000 as a count, it will generate a pulse every 25 millisec. The output of Counter 1 can be used as a clock input to Counter 2. That means the count in Counter 2 will be decremented every 25 millisec. If Counter 2 is also set up in Mode 2 with the count = 40, the output pulse from Counter 2 is generated every second that can be used to interrupt the processor.

**Control Word**

| | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | |
|---|---|---|---|---|---|---|---|---|---|
| Counter 1: | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | = 74H |
| Counter 2: | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | = 94H |
| | Select Counter | | Load Count | | Mode 2 | | | Binary Count | |

## Instructions:

```
;The following subroutine is an initialization for 8254 timer.
;It uses two counters to generate an interrupt every one second.
CNT1LO      EQU     50H             ;Low-order byte of count 50,000
CNT1HI      EQU     0C3H            ;High-order byte of count 50,000
COUNT2      EQU     40              ;Count for Counter 2
CNTREG      EQU     86H             ; Control register address
CNTWRD1     EQU     74H             ;Control word: Mode 2, Counter 1
CNTWRD2     EQU     94H             ;Control word: Mode 2, Counter 2
CNTR1       EQU     82H             ; Counter 1 address
CNTR2       EQU     84H             ; Counter 2 address

SECOND:     MOV     AL, CNTWRD1
            OUT     CNTREG, AL
            MOV     AL, CNTWRD2
            OUT     CNTREG, AL
            MOV     AL, CNT1LO
            OUT     CNTR1, AL
            MOV     AL, CNT1HI
            OUT     CNTR1, AL
            MOV     AL, COUNT2
            OUT     CNTR2, AL
            RET
```

This subroutine sends two control words in a sequence to initialize Counter 1 and Counter 2. The 8254 differentiates these words according to the specified counter in the control words. We could have initialized these counters in a different sequence; for example, the control word for Counter 1 followed by its count. Initially, the equates CNT1LO and CNT1HI are defined in Hex numbers and the equate COUNT2 is specified in decimal. This is based on the assumption that an assembler will convert the decimal count in Hex equivalent. However, this procedure is not feasible for the 16-bit count of 50,000; therefore, the equivalent Hex count (C350H) is loaded into two registers as 50H (low) and C3H (high).

Once this subroutine is called, the output of Counter 2 will generate a pulse every 1 second because the counts are reloaded automatically at the end of each count. This pulse can be used to interrupt the processor, and the processor can update the clock every second. This is a hardware dependent timing and more accurate than the technique used in the previous section