The 8255A is a widely used, programmable, parallel I/O device. It can be programmed to transfer data under various conditions, from simple I/O to interrupt I/O. It is flexible, versatile, and economical (when multiple I/O ports are required), but somewhat complex. It is an important general-purpose I/O device that can be used with almost any microprocessor.

The 8255A has 24 I/O pins that can be grouped primarily in two 8-bit parallel ports: A and B, with the remaining eight bits as port C. The eight bits of port C can be used as individual bits or be grouped in two 4-bit ports: C_{UPPER} (C_U) and C_{LOWER} (C_L), as in Figure 15.1(a). The functions of these ports are defined by writing a control word in the control register.

Figure 15.1(b) shows all the functions of the 8255A, classified according to two modes: the Bit Set/Reset (BSR) mode and the I/O mode. The BSR mode is used to set or reset the bits in port C. The I/O mode is further divided into three modes: Mode 0, Mode 1, and Mode 2. In Mode 0, all ports function as simple I/O ports. Mode 1 is a handshake mode whereby ports A and/or B use bits from port C as handshake signals. In the handshake mode, two types of I/O data transfer can be implemented: status check and interrupt. In Mode 2, port A can be set up for bidirectional data transfer using handshake signals from port C, and port B can be set up either in Mode 0 or Mode 1.

15.1.1 Block Diagram of the 8255A

The block diagram in Figure 15.2(a) shows two 8-bit ports (A and B), two 4-bit ports (C_U and C_L), the data bus buffer, and control logic. Figure 15.2(b) shows a simplified but expanded version of the internal structure, including a control register. This block diagram includes all the elements of a programmable device; port C performs functions similar to that of the status register in addition to providing handshake signals.

CONTROL LOGIC

The control section has six lines. Their functions and connections are as follows:

- $\overline{\text{RD}}$ **(Read):** This control signal enables the Read operation. When the signal is low, the MPU reads data from a selected I/O port of the 8255A.
- $\overline{\text{WR}}$ **(Write):** This control signal enables the Write operation. When the signal goes low, the MPU writes into a selected I/O port or the control register.

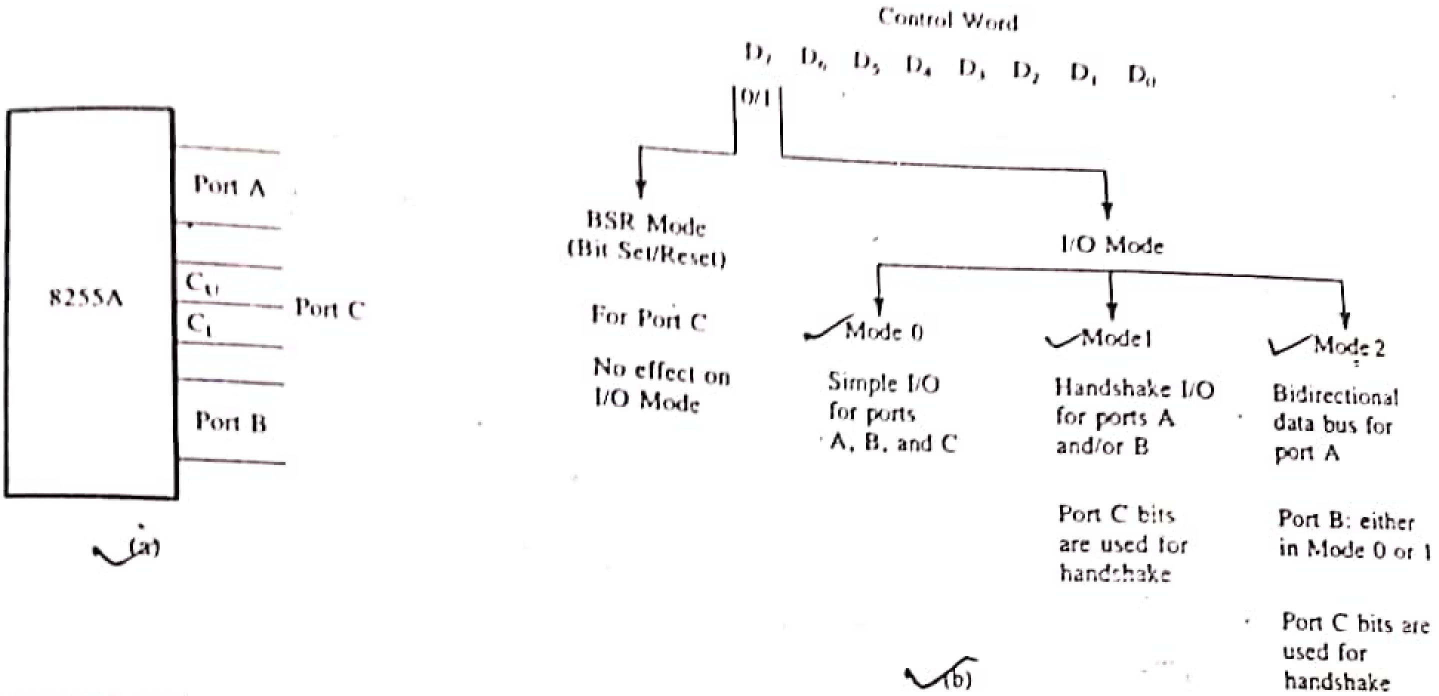


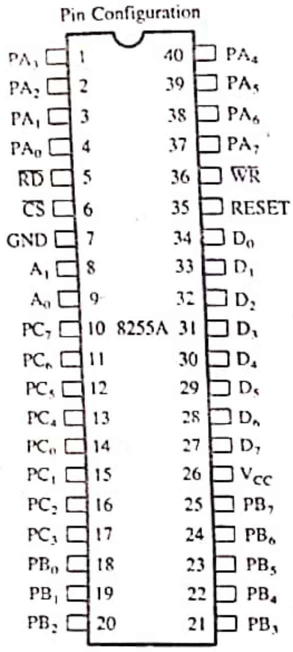
FIGURE 15.1
8255A I/O Ports (a) and Their Modes (b)

- ☐ **RESET (Reset):** This is an active high signal; it clears the control register and sets all ports in the input mode.
- ☐ **$\overline{CS}$, A_0, and A_1:** These are device select signals. $\overline{CS}$ is connected to a decoded address, and A_0 and A_1 are generally connected to MPU address lines A_0 and A_1, respectively.

The $\overline{CS}$ signal is the master Chip Select, and A_0 and A_1 specify one of the I/O ports or the control register as given below:

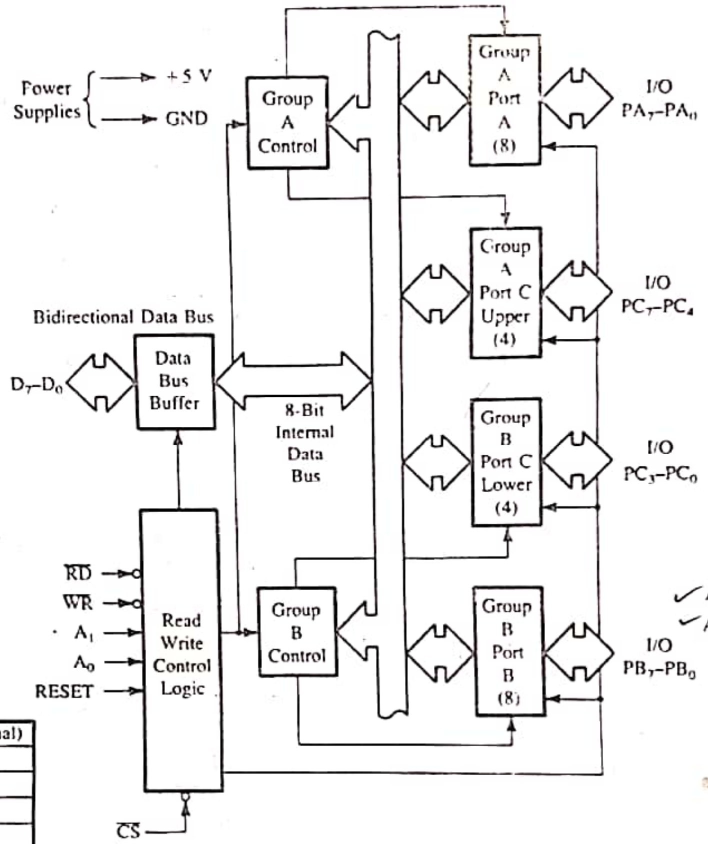
$\overline{CS}$	A_1	A_0	Selected
0	0	0	Port A
0	0	1	Port B
0	1	0	Port C
0	1	1	Control Register
1	X	X	8255A is not selected.

As an example, the port addresses in Figure 15.3(a) are determined by the $\overline{CS}$, A_0, and A_1 lines. The $\overline{CS}$ line goes low when $A_7 = 1$ and A_6 through A_2 are at logic 0. When these signals are combined with A_0 and A_1, the port addresses range from 80H to 83H, as shown in Figure 15.3(b).

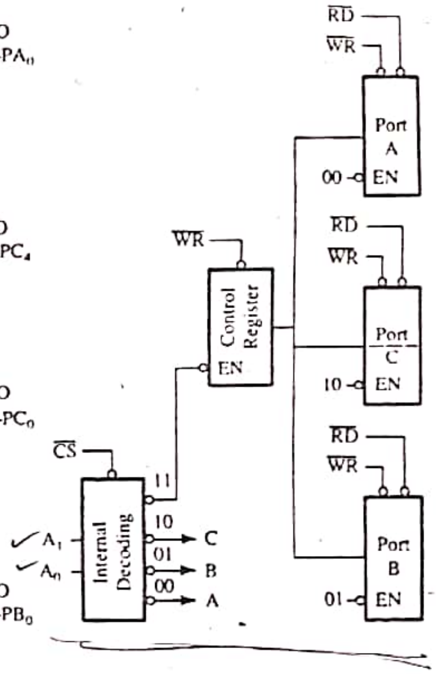


Pin Names

D_7–D_0	Data Bus (Bidirectional)
RESET	Reset Input
$\overline{CS}$	Chip Select
$\overline{RD}$	Read Input
$\overline{WR}$	Write Input
A_0, A_1	Port Address
PA_7–PA_0	Port A (Bit)
PB_7–PB_0	Port B (Bit)
PC_7–PC_0	Port C (Bit)
V_{CC}	+5 Volts
GND	0 Volts

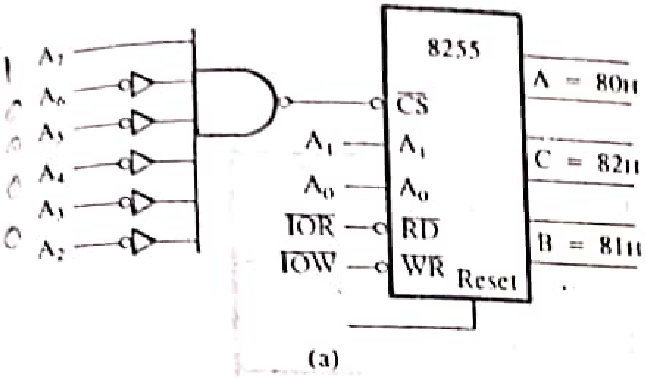


(a)



(b) block diagram of 8255A

FIGURE 15.2
8255A Block Diagram (a) and an Expanded Version of the Control Logic and I/O Ports (b)
SOURCE: A: Intel Corporation, *Peripheral Components* (Santa Clara, Calif.: Author, 1993), p. 3–100.



$\overline{CS}$		Hex Address	Port
A_7 A_6 A_5 A_4 A_3 A_2	A_1 A_0		
1 0 0 0 0 0	0 0	= 80H	A
	0 1	= 81H	B
	1 0	= 82H	C
	1 1	= 83H	Control Register

(b)

FIGURE 15.3
8255A Chip Select Logic (a) and I/O Port Addresses (b)

CONTROL WORD and Control Register.

Figure 15.2(b) shows a register called the **control register**. The contents of this register, called the **control word**, specify an I/O function for each port. This register can be accessed to write a control word when A_0 and A_1 are at logic 1, as mentioned previously. The register is not accessible for a Read operation.

Bit D_7 of the control register specifies either the I/O function or the Bit Set/Reset function, as classified in Figure 15.1(b). If bit $D_7 = 1$, bits D_6–D_0 determine I/O functions in various modes, as shown in Figure 15.4. If bit $D_7 = 0$, port C operates in the Bit Set/Reset (BSR) mode. The BSR control word does not affect the functions of ports A and B (the BSR mode will be described later).

To communicate with peripherals through the 8255A, three steps are necessary:

1. Determine the addresses of ports A, B, and C and of the control register according to the Chip Select logic and address lines A_0 and A_1.
2. Write a control word in the control register.
3. Write I/O instructions to communicate with peripherals through ports A, B, and C.

Examples of the various modes are given in the next section.

15.1.2 Mode 0: Simple Input or Output

In this mode, ports A and B are used as two simple 8-bit I/O ports and port C as two 4-bit ports. Each port (or half-port, in case of C) can be programmed to function as simply an input port or an output port. The input/output features in Mode 0 are as follows:

1. Outputs are latched.
2. Inputs are not latched.
3. Ports do not have handshake or interrupt capability.

Examp
15

1. Identify the port addresses in Figure 15.5.
2. Identify the Mode 0 control word to configure port A and port C_U as output ports and port B and port C_L as input ports.

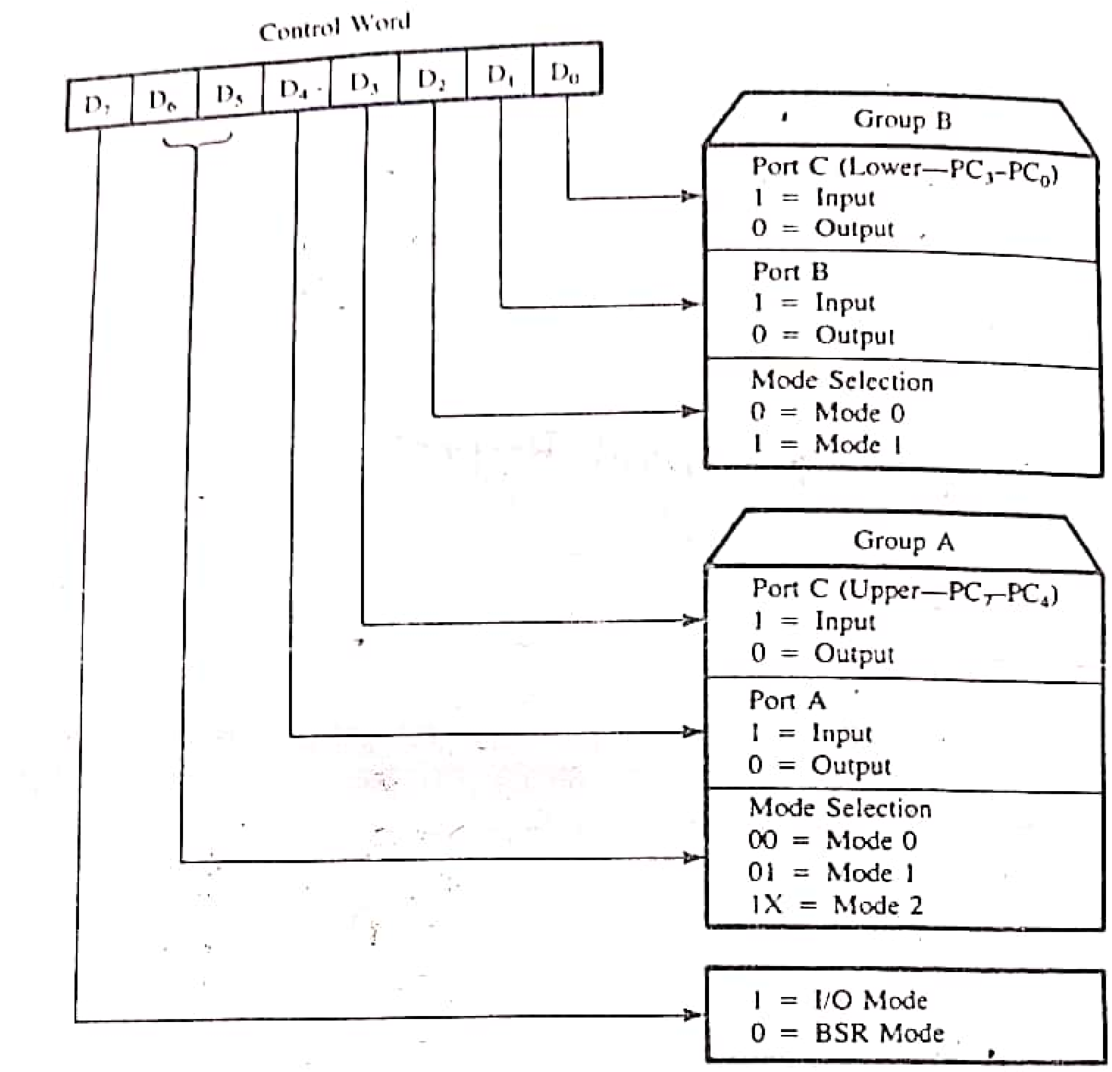


FIGURE 15.4
8255A Control Word Format for I/O Mode
SOURCE: Adapted from Intel Corporation, *Peripheral Components* (Santa Clara, Calif.: Author, 1993), p. 3-104.

3. Write a program to read the DIP switches and display the reading from port B at port A and from port C_L at port C_U.

Solution

1. **Port Addresses** This is a memory-mapped I/O; when the address line A_{15} is high, the Chip Select line is enabled. Assuming all don't care lines are at logic 0, the port addresses are as follows:

Port A	=	8000H ($A_1 = 0$, $A_0 = 0$)
Port B	=	8001H ($A_1 = 0$, $A_0 = 1$)
Port C	=	8002H ($A_1 = 1$, $A_0 = 0$)
Control Register	=	8003H ($A_1 = 1$, $A_0 = 1$)

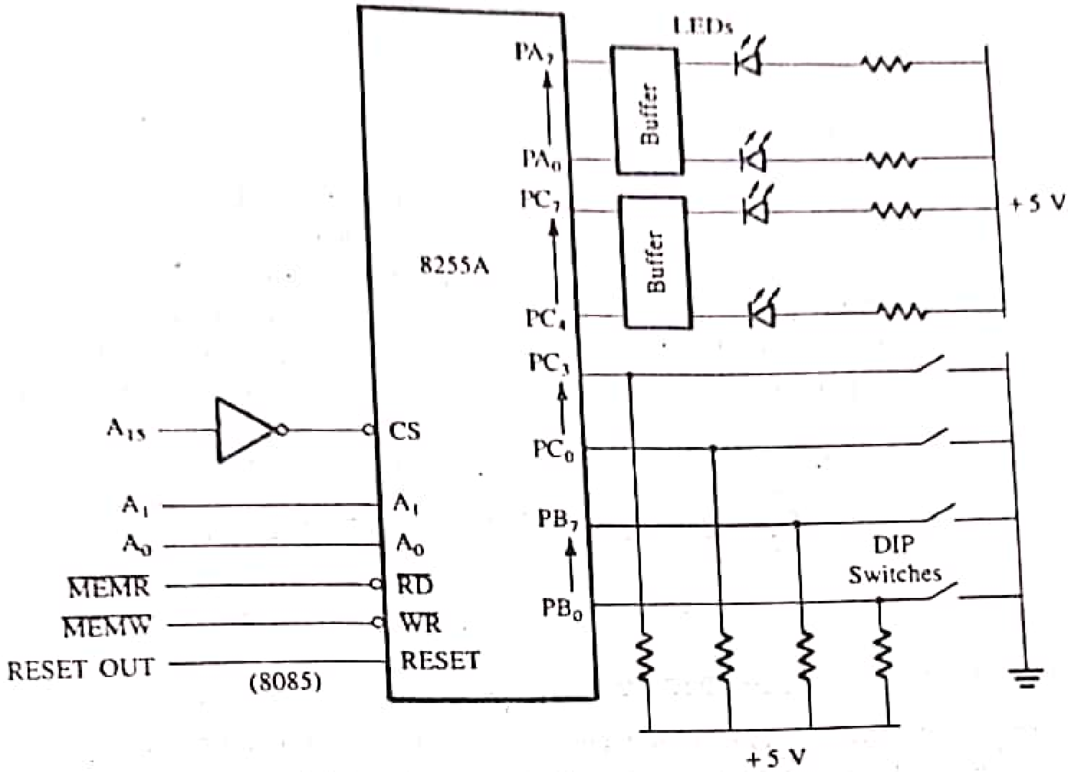


FIGURE 15.5
Interfacing 8255A I/O Ports in Mode 0

2. Control Word

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
1	0	0	0	0	0	1	1	= 83H

- D_7 = 1 → I/O Function
- D_6, D_5 = 0 0 → Port A in Mode 0
- D_4 = 0 → Port A = Output
- D_3 = 0 → Port C_U = Output
- D_2 = 0 → Port B in Mode 0
- D_1 = 1 → Port B = Input
- D_0 = 1 → Port C_L = Input

3. Program

```
MVI A,83H     ;Load accumulator with the control word
STA 8003H     ;Write word in the control register to initialize the ports
LDA 8001H     ;Read switches at port B
STA 8000H     ;Display the reading at port A
LDA 8002H     ;Read switches at port C
ANI 0FH       ;Mask the upper four bits of port C; these bits are not input data
RLC           ;Rotate and place data in the upper half of the accumulator
RLC
```

```
RLC
RLC
STA 8002H        ;Display data at port CU
HLT
```

Program Description The circuit is designed for memory-mapped I/O; therefore, the instructions are written as if all the 8255A ports are memory locations.

The ports are initialized by placing the control word 83H in the control register. The instructions STA and LDA are equivalent to the instructions OUT and IN, respectively.

In this example, the low four bits of port C are configured as input and the high four bits are configured as output; even though port C has one address for both halves C_U and C_L (8002H). Read and Write operations are differentiated by the control signals $\overline{MEMR}$ and $\overline{MEMW}$. When the MPU reads port C (e.g., LDA 8002H), it receives eight bits in the accumulator. However, the high-order bits (D_7–D_4) must be ignored because the input data bits are in PC_3–PC_0. To display these bits at the upper half of port C, bits (PC_3–PC_0) must be shifted to PC_7–PC_4.

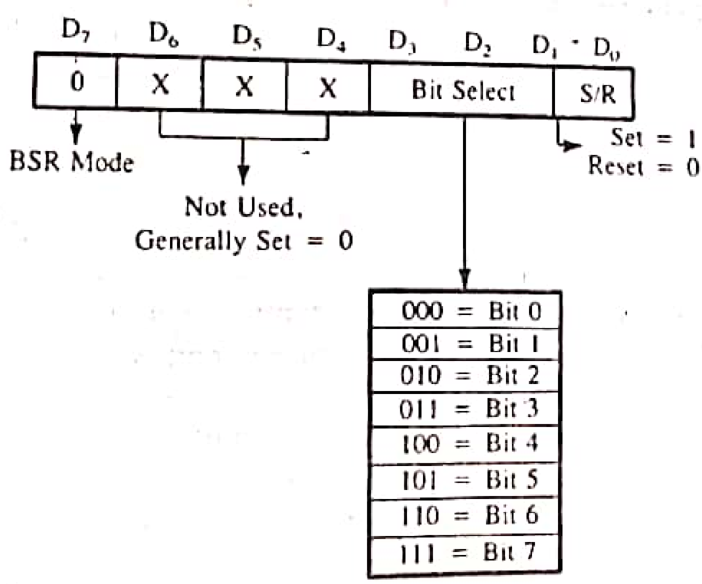
15.1.3 BSR (Bit Set/Reset) Mode

The BSR mode is concerned only with the eight bits of port C, which can be set or reset by writing an appropriate control word in the control register. A control word with bit D_7 = 0 is recognized as a BSR control word, and it does not alter any previously transmitted control word with bit D_7 = 1; thus the I/O operations of ports A and B are not affected by a BSR control word. In the BSR mode, individual bits of port C can be used for applications such as an on/off switch.

BSR CONTROL WORD

This control word, when written in the control register, sets or resets one bit at a time, as specified in Figure 15.6.

FIGURE 15.6
8255A Control Word Format in the BSR Mode



Write a BSR control word subroutine to set bits PC_7 and PC_3 and reset them after 10 ms. Use the schematic in Figure 15.3 and assume that a delay subroutine is available.

Example 15.2

Solution

BSR CONTROL WORDS

		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
To set bit PC_7	=	0	0	0	0	1	1	1	1	=	0FH
To reset bit PC_7	=	0	0	0	0	1	1	1	0	=	0EH
To set bit PC_3	=	0	0	0	0	0	1	1	1	=	07H
To reset bit PC_3	=	0	0	0	0	0	1	1	0	=	06H

PORT ADDRESS

Control register address = 83H; refer to Figure 15.3(b).

SUBROUTINE

```
BSR:    MVI A,0FH       ;Load byte in accumulator to set PC7
        OUT 83H         ;Set PC7 = 1
        MVI A,07H       ;Load byte in accumulator to set PC3
        OUT 83H         ;Set PC3 = 1
        CALL DELAY      ;This is a 10-ms delay
        MVI A,06H       ;Load accumulator with the byte to reset PC3
        OUT 83H         ;Reset PC7
        MVI A,0EH       ;Load accumulator with the byte to reset PC7
        OUT 83H         ;Rest PC7
        RET
```

From an analysis of the above routine, the following points can be noted:

1. To set/reset bits in port C, a control word is written in the control register and not in port C.
2. A BSR control word affects only one bit in port C.
3. The BSR control word does not affect the I/O mode.

15.1.4 Illustration: Interfacing I/O Devices for the MCTS Project Using an A/D Converter

In the MCTS project (described in Chapter 1), the system is expected to read the temperature in a room, display the temperature on a liquid crystal display (LCD) panel, turn on a fan if the temperature is above a set point, and turn on a heater if the temperature is below a set point. This illustration includes the interfacing of a temperature sensor using an A/D converter and I/O devices such as a fan and a heater; the LCD display is illustrated in Chapter 17.

PROBLEM STATEMENT

1. Interface a temperature sensor using an A/D converter and port A of the 8255.
2. Interface a fan and a heater using optocouplers and triacs to drive the I/O devices.
3. Write instructions to read the temperature. If the temperature is less than 10°C, turn on the heater, and if the temperature is higher than 35°C, turn on the fan.

The following components are specified for interfacing:

1. 8255—Port A in Mode 0 and Port C in BSR mode
2. 8-bit A/D Converter—National ADC0801 and Temperature Sensor—National LM135
3. Optoisolator—MOC 3011 and Triac 2N6071

CIRCUIT ANALYSIS

Figure 15.7 shows the interfacing circuit. In this illustration, we will use the symbolic addresses for ports A, C, and Control when we write instructions. The specific port addresses can be the same as in Figure 15.3 (80H to 83H). Figure 15.7 shows that the temperature sensor LM135 is connected as input to the A/D converter, and Port A is set up as an input port for the A/D converter. Port C is set up in BSR mode. Signal line, PC_0 is set up as an input—PC_0 is connected to the INTR. Four signal lines, PC_4–PC_7, are set up as outputs. PC_4 is used to start conversions, and PC_5 and PC_6 are connected to optoisolators that are used to drive the fan and the heater, respectively, using triacs.

TEMPERATURE SENSOR—LM 135

This is a three-terminal integrated circuit temperature sensor that can operate as a two-terminal zener device, and its third terminal can be used for calibration if necessary. Its output voltage is directly proportional to absolute temperature in Kelvin; it changes 10 mV/°K and operates over the temperature range from −55°C to +150°C. In Figure 15.7, the output of the sensor is connected to the input $V_{IN}(+)$ of the A/D converter ADC0801. At 0°C, which is 273°K, the output voltage of the sensor is 2.73 V, and at 25°C room temperature, the output voltage is 2.98 V. See the data sheet in Appendix D for additional information on the LM135. Temperature sensors are also available that have outputs proportional to temperatures in Fahrenheit and Centigrade.

A/D CONVERTER (NATIONAL SEMICONDUCTOR ADC0801)

The basic concepts underlying the operation of an A/D converter are discussed in Chapter 13. In Section 13.2.4, the operation of the ADC0801 and its interfacing are described.

This is an 8-bit A/D converter that can convert analog input voltage from 0 to 5 V. It has two inputs, $V_{IN}(+)$ and $V_{IN}(-)$, for differential analog input. If an analog input signal has a smaller range than 0 to 5 V, this converter provides a reference voltage (pin 9) $V_{REF}/2$ that can adjust lower analog voltages to full resolution. The dc voltage applied to pin 9 has a gain of 2. In Figure 15.7, $V_{REF}/2$ is connected to 1.28 V by adjusting the 10K pot; therefore, the full-scale range is calibrated for 0 to 2.56 V, and the output of the

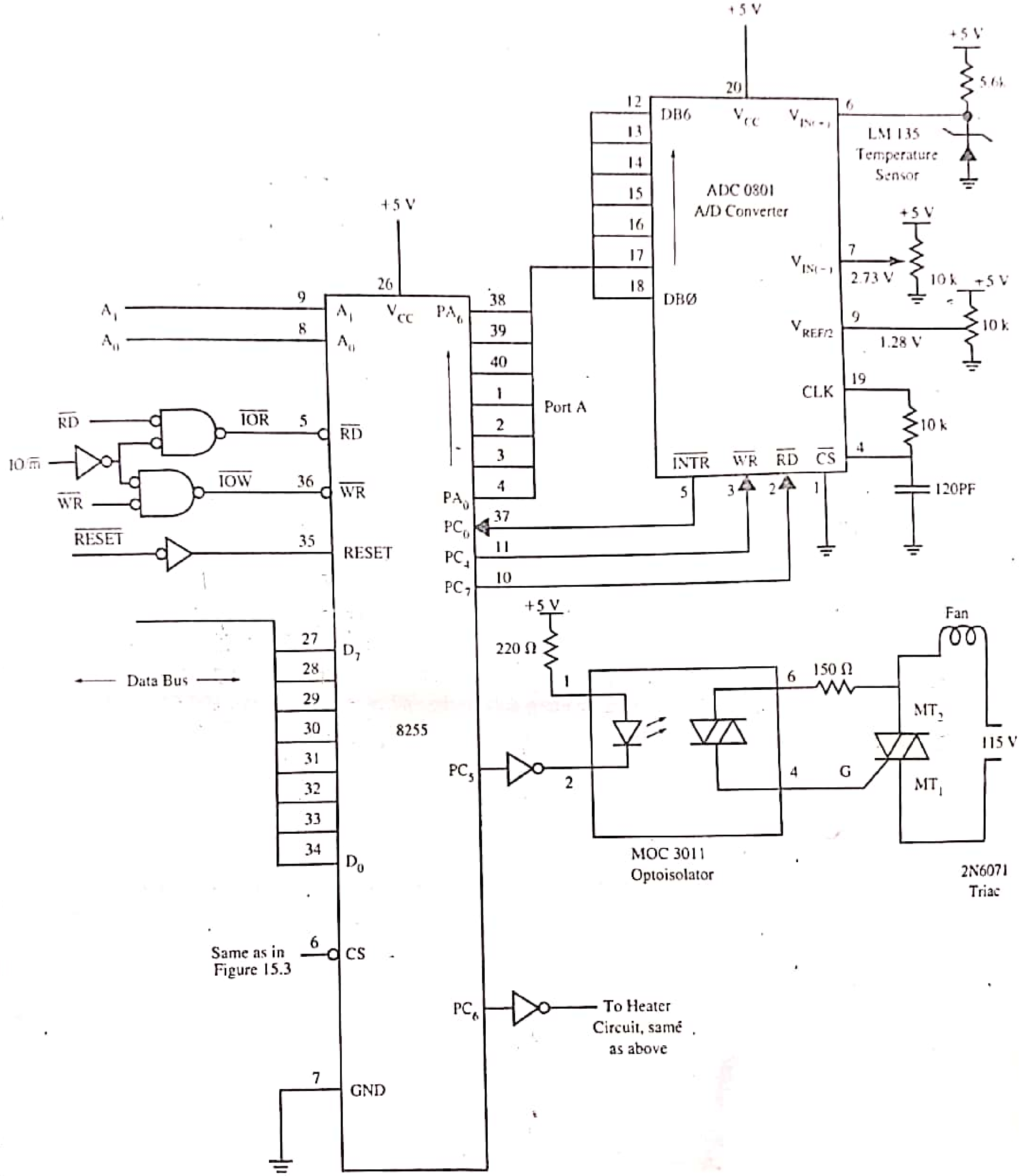


FIGURE 15.7
I/O Schematic for MCTS

converter will be FF_H when the input signal is 2.56 V. The pin $V_{IN}(-)$ is connected to 2.73 V, which is the output voltage of the temperature sensor at 0°C. This simplifies the software because each 10 mV change will provide 1° rise in temperature from 0°C.

OPTOISOLATOR AND TRIAC

Figure 15.7 shows that signal PC_5 is connected to an optoisolator (also called an optocoupler), MOC 3011, through an inverter, 7404. The primary purpose of the optoisolator is to isolate digital signals from high-voltage signals; therefore, the digital signal ground and the AC ground should never be connected together. The MC 3011 consists of a light emitting diode and a triac driver, and they are eclectically isolated. When sufficient current flows in the diode (approximately 10 mA in this case), it emits light that causes current flow in the driver circuit, and in turn, the driver turns on the triac.

In simplest terms the triac can be viewed as a three-terminal high-voltage switch. A high-voltage load can be connected in series with the terminals MT_1 and MT_2. When specified current flows in terminal G, it triggers the triac circuit, and when the current in terminal G is reduced to a specified limit, the triac is turned off.

In Figure 15.7, logic 1 at PC_5 provides logic 0 at the output of the inverter that turns the LED inside the optoisolator; the current flow in the LED is limited by the resistor 220 Ω. The inverter 7404 is used to sink the necessary current; Port C of the 8255 is not able to sink 10 mA of current. When the LED is turned on, it supplies enough current to the driver to trigger the triac, and in turn, the triac turns on the fan.

PROGRAM

The program consists of three segments: (1) initialization of the ports and the stack, (2) starting a conversion, reading data at the end of the conversion, and checking for set values, and (3) turning on and off the fan and the heater accordingly.

CONTROL WORDS

Mode 0: Control Word The configuration of the ports is specified as follows:

- Port A: As an input port.
- Port C_L: As an input port because bit PC_0 is used to read the end of conversion.
- Port C_U: As an output port because bit PC_4 is used to start conversion and PC_7 is used to assert $\overline{RD}$ signal.
- Port B: not used.

Therefore, the control words necessary to meet the requirements are as follows:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
1	0	0	1	0	0	0	1	= 91H
I/O function	Mode 0 for port A		Port A input	Port C_U output	Port B is not used		Port C_L input	

BSR Control Words

D_7	D_6 D_5 D_4	D_3 D_2 D_1	D_0
0	0 0 0	0 0 0	0/1 = 00 to reset and 01 to set for bit PC_0
BSR Mode	Don't Care	Bits	0 = Reset 1 = Set

These control words range from 00 to 0F for all eight bits.

Subroutine Instructions

```
GETTEMP:   ;This subroutine starts a conversion, checks the INTR input signal and
           ;continues to read Port A. When the conversion is complete, the INTR
           ;goes low. It reads the temperature, which is already adjusted to be read
           ;in degrees C. It checks high (35°C) and low temperature (10°C) limits
           ;and calls subroutines to turn on/off the fan or the heater accordingly.

GETTEMP:  MVI A, 10010001B   ;Control word to set up ports
          OUT CNTRL          ;Set up ports A and CL as input and CU as output
          MVI A, 00001111B   ;Word to set PC7 high
          OUT CNTRL          ;Disable RD
          MVI A, 00001000    ;Word to set PC4 (WR) low
          OUT CNTRL          ;Start conversion
          MVI A, 00001001    ;Word to set PC4 (WR) high
          OUT CNTRL          ;Set WR high
READ:     IN PORTC           ;Read port C to check PC0
          RAR                ;Place PC0 in Carry flag
          JC READ            ;If PC0 is high, go back and read again
          MVI A, 00001110    ;Word to set PC7 (RD) low
          OUT CNTRL          ;Assert RD signal
          IN PORTA           ;Read A/D converter
          MOV B, A           ;Save temperature reading
          MVI A, 00001111    ;Word to set PC7 (RD) high
          OUT CNTRL          ;Disable RD
          MOV A, B           ;Get saved temperature reading
          CPI 35D            ;Is it higher than 35°C
          CNC FANON          ;Turn on the fan
          CC FANOFF          ;Otherwise, turn off the fan
          CPI 10D            ;Is it less than 10°C
          CC HEATON          ;If yes, turn on the heater
          CNC HEATOFF        ;If no, turn off the heater
          RET                ;Go back to main program
FANON:    PUSH PSW           ;Save A and flags
```

```
           MVI A, 00001011     ;Word to set PC5 to turn on the fan
           OUT CNTRL           ;Turn on the fan
           POP PSW             ;Retrieve A and flags
           RET
FANOFF:    PUSH PSW            ;Save A and flags
           MVI A, 00001010     ;Word to reset PC5 to turn off the fan
           OUT CNTRL           ;Turn off the fan
           POP PSW             ;Retrieve A and flags
           RET
```

HEATON and HEATOFF are subroutines similar to FANON and FANOFF except that they turn on/off the bit PC_6.

PROGRAM DESCRIPTION

The subroutine initializes port A and the lower half of port C as inputs and the upper half of Port C as outputs. It disables $\overline{RD}$ and asserts $\overline{WR}$ signals. When $\overline{WR}$ goes low, the conversion begins and the INTR goes high. When the conversion is complete, the INTR goes low, indicating that the data byte is ready for reading (see Figure 13.13). The INTR signal can be read as input in a loop or it can be used to interrupt the processor.

The subroutine reads port C and places the status of PC_0 in the carry flag by rotating the bit right. If the CY flag is logic 1, the subroutine goes back to the label READ and reads Port C again. The subroutine stays in the loop until PC_0 is at logic zero. Once PC_0 goes low, the processor asserts $\overline{RD}$ low and reads the data byte. This data byte is already hardware calibrated to be read in degrees Centigrade. The subroutine compares the reading in A with the high set point (35°C). If the comparison does not generate a carry, it means the reading in A is higher than 35°C, and it calls a subroutine FANON to turn on the fan. If it generates a carry, it means the reading in A is lower than 35°C, and it turns off the fan by calling the subroutine FANOFF. However, the subroutine does not check whether a fan is already off or on; in some instances this may be a redundant action. Next it compares the reading with the low set point (10°C). If the reading is lower than 10°C, it calls the subroutine HEATON and if it is higher than 10°C, it calls the subroutine HEATOFF. The subroutines FANON, FANOFF, HEATON, and HEATOFF turn on/off bits PC_5 and PC_6 connected to the fan and the heater.

15.1.5 Mode 1: Input or Output with Handshake

In Mode 1, handshake signals are exchanged between the MPU and peripherals prior to data transfer. The features of this mode include the following:

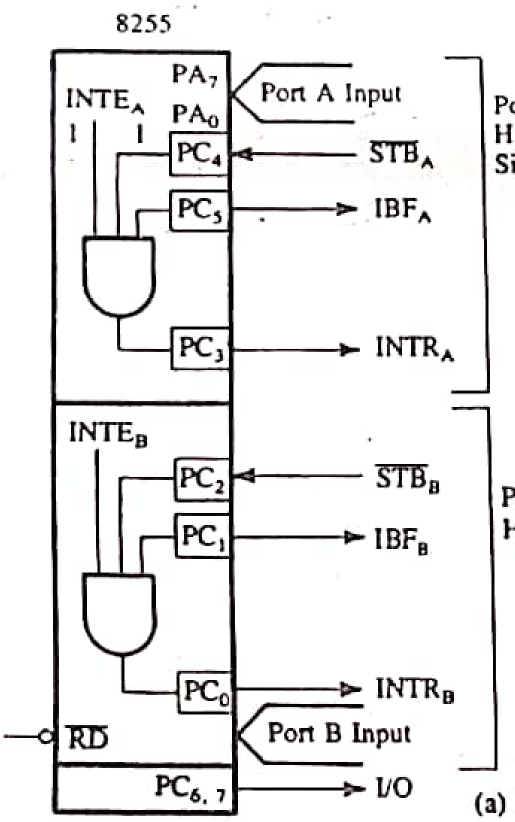
1. Two ports (A and B) function as 8-bit I/O ports. They can be configured either as input or output ports.
2. Each port uses three lines from port C as handshake signals. The remaining two lines of port C can be used for simple I/O functions.
3. Input and output data are latched.
4. Interrupt logic is supported.

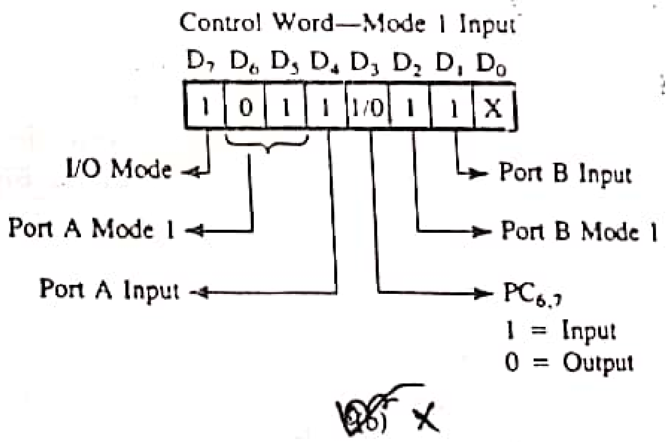
In the 8255A, the specific lines from port C used for handshake signals vary according to the I/O function of a port. Therefore, input and output functions in Mode 1 are discussed separately.

MODE 1: INPUT CONTROL SIGNALS

Figure 15.8(a) shows the associated control signals used for handshaking when ports A and B are configured as input ports. Port A uses the upper three signals: PC_3, PC_4, and PC_5. Port B uses the lower three signals: PC_2, PC_1, and PC_0. The functions of these signals are as follows:

- **$\overline{STB}$** (Strobe Input): This signal (active low) is generated by a peripheral device to indicate that it has transmitted a byte of data. The 8255A, in response to $\overline{STB}$, generates IBF and INTR, as shown in Figure 15.9.
- **IBF (Input Buffer Full):** This signal is an acknowledgment by the 8255A to indicate that the input latch has received the data byte. This is reset when the MPU reads the data (Figure 15.9).
- **INTR (Interrupt Request):** This is an output signal that may be used to interrupt the MPU. This signal is generated if $\overline{STB}$, IBF, and INTE (Internal flip-flop) are all at logic 1. This is reset by the falling edge of the $\overline{RD}$ signal (Figure 15.9).





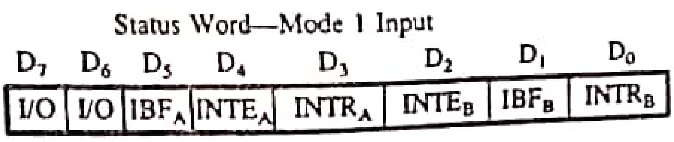


FIGURE 15.8
8255A Mode 1: Input Configuration
SOURCE: Adapted from Intel Corporation, *Peripheral Components* (Santa Clara, Calif.: Author, 1993), p. 3-110.

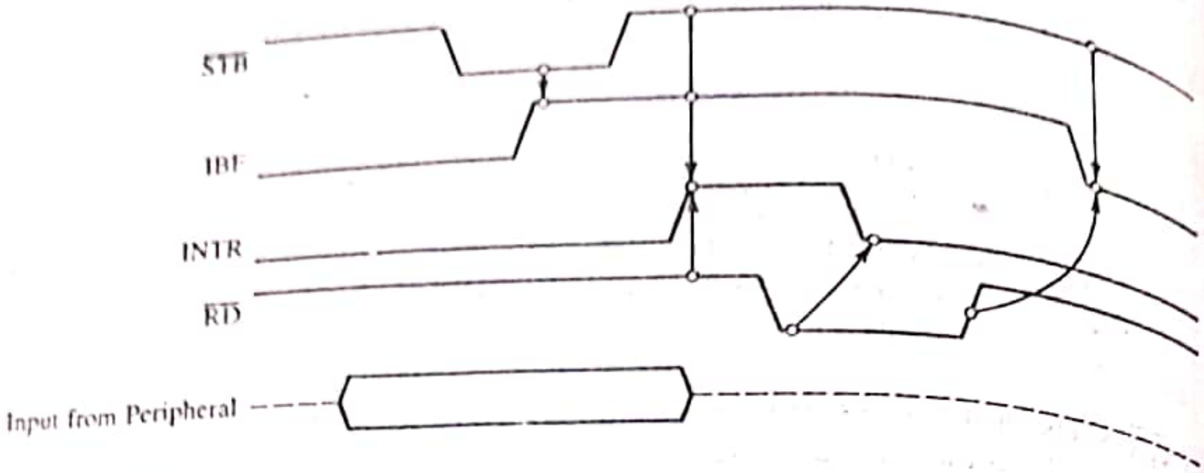


FIGURE 15.9
8255A Mode 1: Timing Waveforms for Strobed Input (with Handshake)
SOURCE: Adapted from Intel Corporation, *Peripheral Components* (Santa Clara, Calif.: Author, 1993), p. 3-110.

- **INTE (Interrupt Enable):** This is an internal flip-flop used to enable or disable the generation of the INTR signal. The two flip-flops $INTE_A$ and $INTE_B$ are set/reset using the BSR mode. The $INTE_A$ is enabled or disabled through PC_4, and $INTE_B$ is enabled or disabled through PC_2.

CONTROL AND STATUS WORDS

Figure 15.8(b) uses control words derived from Figure 15.4 to set up port A and port B as input ports in Mode 1. Similarly, Figure 15.8(c) also shows the status word, which will be placed in the accumulator if port C is read.

PROGRAMMING THE 8255A IN MODE 1

The 8255A can be programmed to function using either status check I/O or interrupt I/O. Figure 15.10(a) shows a flowchart for the status check I/O. In this flowchart, the MPU continues to check data status through the IBF line until it goes high. This is a simplified flowchart; however, it does not show how to handle data transfer if two ports are being used. The technique is similar to that of Mode 0 combined with the BSR mode. The disadvantage of the status check I/O with handshake is that the MPU is tied up in the loop.

The flowchart in Figure 15.10(b) shows the steps required for the interrupt I/O, assuming that vectored interrupts are available. The confusing step in the interrupt I/O is to set INTE either for port A or port B. Figure 15.8(a) shows that the $\overline{STB}$ signal is connected to pin PC_4 and the $INTE_A$ is also controlled by the pin PC_4. (In port B, pin PC_2 is used for the same purposes.) However, the $INTE_A$ is set or reset in the BSR mode and the BSR control word has no effect when ports A and B are set in Mode 1.

In case the INTR line is used to implement the interrupt, it may be necessary to read the status of $INTR_A$ and $INTR_B$ to identify the port requesting an interrupt service and to determine the priority through software, if necessary.

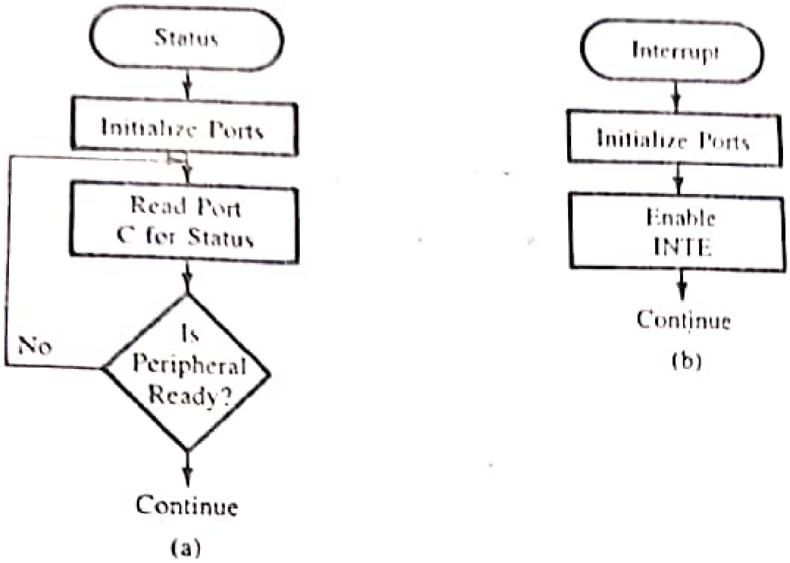


FIGURE 15.10
Flowcharts: Status Check I/O (a) and Interrupt I/O (b)

MODE 1: OUTPUT CONTROL SIGNALS

Figure 15.11 shows the control signals when ports A and B are configured as output ports. These signals are defined as follows:

- **$\overline{\text{OBF}}$ (Output Buffer Full):** This is an output signal that goes low when the MPU writes data into the output latch of the 8255A. This signal indicates to an output peripheral that new data are ready to be read (Figure 15.12). It goes high again after the 8255A receives an $\overline{\text{ACK}}$ from the peripheral.
- **$\overline{\text{ACK}}$** (Acknowledge): This is an input signal from a peripheral that must output a low when the peripheral receives the data from the 8255A ports (Figure 15.12).
- **INTR (Interrupt Request):** This is an output signal, and it is set by the rising edge of the $\overline{\text{ACK}}$ signal. This signal can be used to interrupt the MPU to request the next data byte for output. The INTR is set when $\overline{\text{OBF}}$, $\overline{\text{ACK}}$, and INTE are all one (Figure 15.12) and reset by the falling edge of $\overline{\text{WR}}$.
- **INTE (Interrupt Enable):** This is an internal flip-flop to a port and needs to be set to generate the INTR signal. The two flip-flops $INTE_A$ and $INTE_B$ are controlled by bits PC_6 and PC_2, respectively, through the BSR mode.
- **$PC_{4,5}$:** These two lines can be set up either as input or output.

CONTROL AND STATUS WORDS

Figure 15.11(b) shows the control word used to set up ports A and B as output ports in Mode 1. Similarly, Figure 15.11(c) also shows the status word, which will be placed in the accumulator if port C is read.

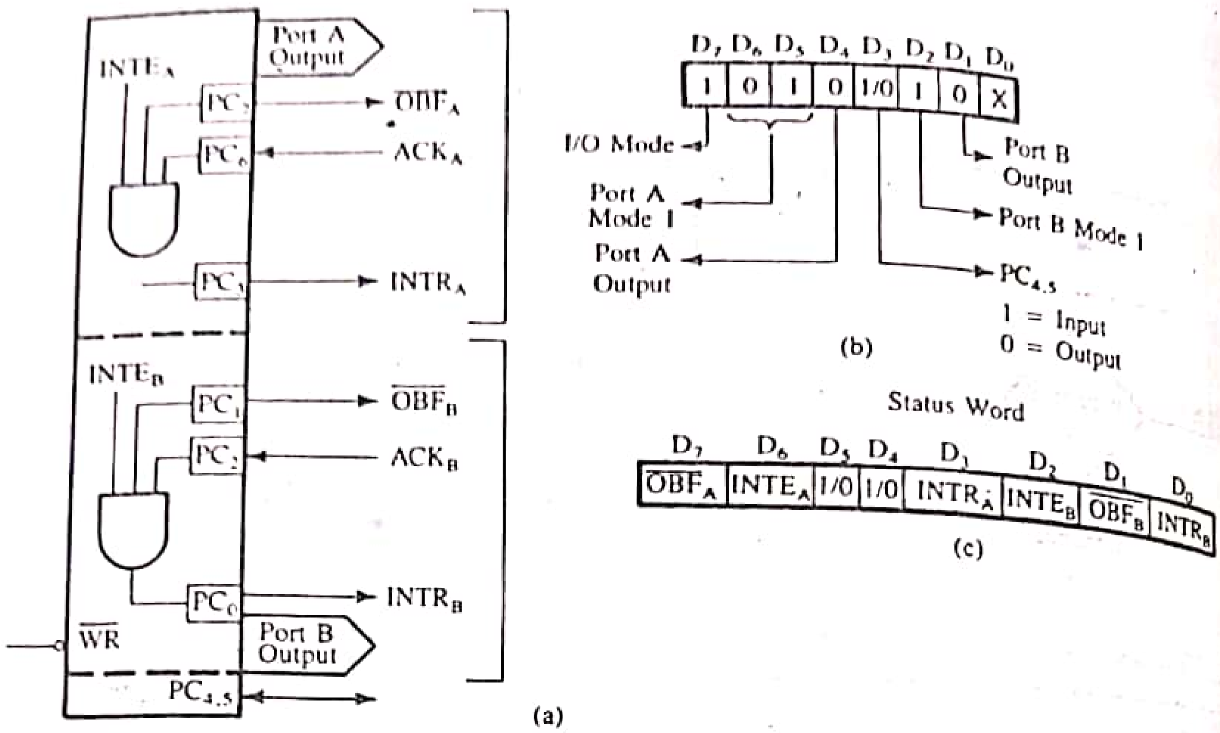


FIGURE 15.11
8255A Mode 1: Output Configuration
SOURCE: Adapted from Intel Corporation, *Peripheral Components* (Santa Clara, Calif.: Author, 1993), p. 3-111.

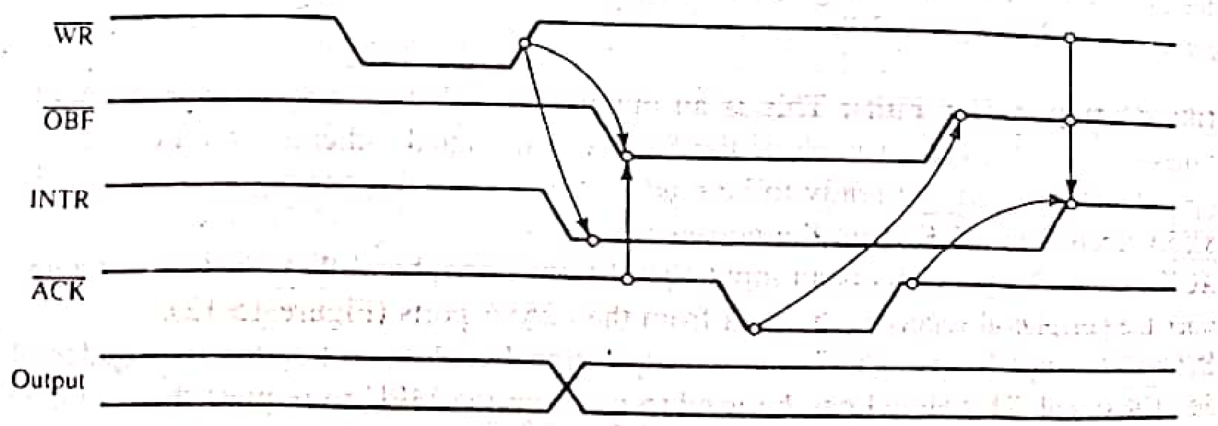


FIGURE 15.12
8255 Mode 1: Timing Waveforms for Strobed (with Handshake) Output
SOURCE: Adapted from Intel Corporation, *Peripheral Components* (Santa Clara, Calif.: Author, 1993), p. 3-111.

15.1.6 Illustration: An Application of the 8255A in the Handshake Mode (Mode 1)

PROBLEM STATEMENT

Figure 15.13 shows an interfacing circuit using the 8255A in Mode 1. Port A is designed as the input port for a keyboard with interrupt I/O, and port B is designed as the output port for a printer with status check I/O.

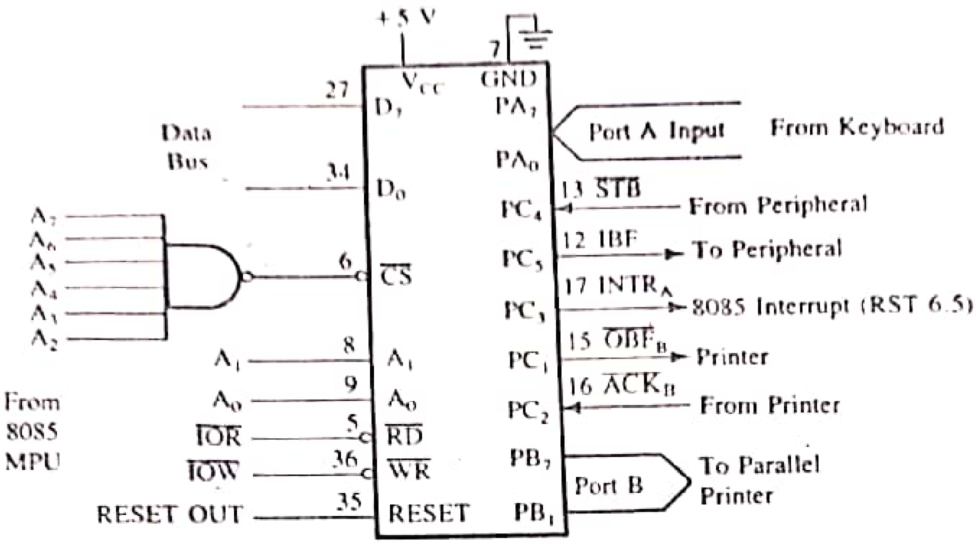


FIGURE 15.13
Interfacing the 8255A in Mode 1 (Strobed Input/Output)

1. Find port addresses by analyzing the decode logic.
2. Determine the control word to set up port A as input and port B as output in Mode 1.
3. Determine the BSR word to enable $INTE_A$ (port A).
4. Determine the masking byte to verify the $\overline{OBF}_B$ line in the status check I/O (port B).
5. Write initialization instructions and a printer subroutine to output characters that are stored in memory.

1. Port Addresses The 8255A is connected as peripheral I/O. When the address lines A_7–A_2 are all 1, the output of the NAND gate goes low and selects the 8255A. The individual ports are selected as follows:

Port A	= FCH ($A_1 = 0$, $A_0 = 0$)
Port B	= FDH ($A_1 = 0$, $A_0 = 1$)
Port C	= FEH ($A_1 = 1$, $A_0 = 0$)
Control Register	= FFH ($A_1 = 1$, $A_0 = 1$)

2. Control Word to Set Up Port A as Input and Port B as Output in Mode 1

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0 = B4H
1	0	1	1	0	1	0	0
I/O function	Port A in Mode 1		Port A as input	$PC_{6,7}$ as don't care	Port B in Mode 1	Port B as output	Bit D_0 is don't care

In the above control word, all bits are self-explanatory, and bits D_3 and D_0 are in a don't care logic state. To generate interrupt signal $INTR_A$, flip-flop $INTE_A$ should be set to 1, which can be accomplished by using the BSR Mode to set PC_4.

The output to the printer (port B) is status-controlled. Therefore, the status of line $\overline{OBF_B}$ can be checked by reading bit D_1 of port C_L.

3. BSR Word to Set $INTE_A$ To set the Interrupt Enable flip-flop of port A ($INTE_A$), bit PC_4 should be 1.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0 = 09H
0	0	0	0	1	0	0	1
BSR mode	Don't care			Bit PC_4			Bit set

4. Status Word to Check $\overline{OBF_B}$

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
X	X	X	X	X	X	$\overline{OBF_B}$	X

Masking byte: 02H

5. Initialization Program

```
        MVI A,B4H      ;Word to initialize port A as input,
                       ;  port B as output in Mode 1
        OUT FFH
        MVI A,09H      ;Set INTE_A (PC4)
        OUT FFH        ;Using BSR Mode
        EI             ;Enable interrupts
        CALL PRINT     ;Continue other tasks
```

Print Subroutine

```
PRINT:  LXI H,MEM      ;Point index to location of stored characters
        MVI B,COUNT    ;Number of characters to be printed
NEXT:   MOV A,M        ;Get character from memory
        MOV C,A        ;Save character
STATUS: IN FEH         ;Read port C for status of OBF
        ANI 02H        ;Mask all bits except D1
        JZ STATUS      ;If it is low, the printer is not ready; wait in the loop
        MOV A,C
        OUT FDH        ;Send a character to port B
        INX H          ;Point to the next character
        DCR B
        JNZ NEXT
        RET
```

PROGRAM DESCRIPTION

This I/O design using the 8255A in Mode 1 allows two operations: outputting to the printer and data entry through the keyboard. The printer interfacing is designed with the status check and the keyboard interfacing with the interrupt.

In the PRINT subroutine, the character is placed in the accumulator, and the status is read by the instruction IN FEH. Initially, port B is empty, bit PC_1 ($\overline{OBF_B}$) is high, and the instruction OUT FDH sends the first character to port B. The rising edge of the $\overline{WR}$ signal sets signal $\overline{OBF}$ low, indicating the presence of a data byte in port B, which is sent out to the printer (Figure 15.12). After receiving a character, the printer sends back an acknowledge signal ($\overline{ACK}$), which in turn sets $\overline{OBF_B}$ high, indicating that port B is ready for the next character, and the PRINT subroutine continues.

If a key is pressed during the PRINT, a data byte is transmitted to port A and the $\overline{STB_A}$ goes low, which sets IBF_A high. The initialization routine should set the $INTE_A$ flip-flop. When the $\overline{STB_A}$ goes high, all the conditions (i.e., $IBF_A = 1$, $INTE_A = 1$) to generate $INTR_A$ are met. This signal, which is connected to the RST 6.5, interrupts the MPU, and the program control is transferred to the service routine. This service routine would read the contents of port A, enable the interrupts, and return to the PRINT routine (the interrupt service routine is not shown here).

15.1.7 Mode 2: Bidirectional Data Transfer

This mode is used primarily in applications such as data transfer between two computers or floppy disk controller interface. In this mode, port A can be configured as the bidirectional port and port B either in Mode 0 or Mode 1. Port A uses five signals from port C as handshake signals for data transfer. The remaining three signals from port C can be used either as simple I/O or as handshake for port B. Figure 15.14 shows two configurations of Mode 2. This mode is illustrated in Section 15.3.

15.2 ILLUSTRATION: INTERFACING KEYBOARD AND SEVEN-SEGMENT DISPLAY

This illustration is concerned with interfacing a pushbutton keyboard and a seven-segment LED display using the 8255A. The emphasis in this illustration is not particularly on the features of 8255A but on how to integrate hardware and software. When a key is pressed, the binary reading of the key has almost no relationship to what it represents. Similarly, to display a number at a seven-segment LED, the binary value of the number needs to be converted into the seven-segment code, which is primarily decided by the hardware consideration. This illustration demonstrates how the microprocessor monitors the changes in hardware reading and converts into appropriate binary reading using its instruction set.

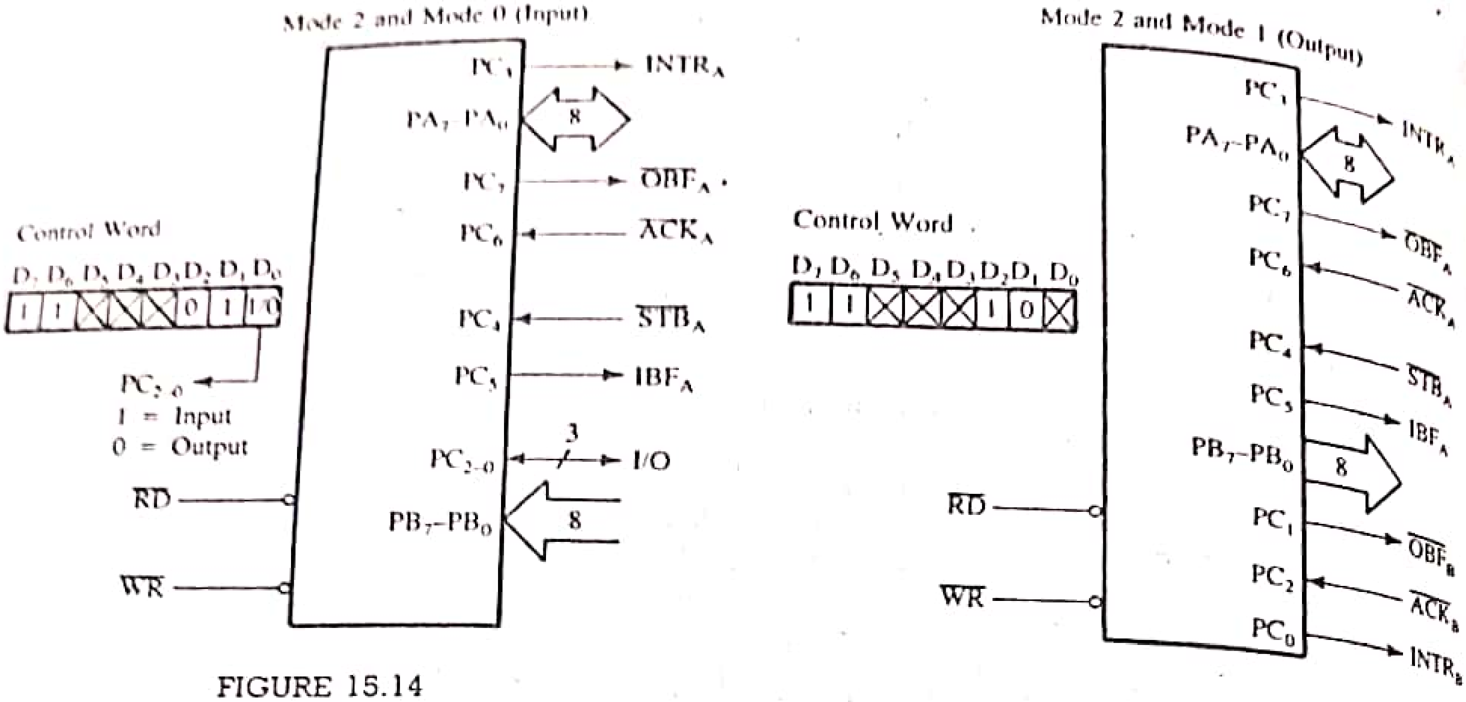


FIGURE 15.14
8255A Mode 2: Bidirectional Input/Output
SOURCE: Intel Corporation, *Peripheral Components* (Santa Clara, Calif.: Author, 1993), p. 3-113.

15.2.1 Problem Statement

A pushbutton keyboard is connected to port A and a seven-segment LED is connected to port B of the 8255A, as shown in Figure 15.15. Port A should be configured as an input port and port B as an output port; this is a simple I/O configuration in Mode 0 without the use of handshake signals or the interrupt.

Write a program to monitor the keyboard to sense a key pressed and display the number of the key at the seven-segment LED. For example, when the key K_7 is pressed, the digit 7 should be displayed at port B.

15.2.2 Problem Analysis

In this problem, the address decoding circuit is the same as in Figure 15.13; therefore, the port addresses range from FCH to FFH. The keyboard circuit shown in Figure 15.15 is similar to that in Figure 4.11 except that the DIP switches are replaced by pushbutton keys and the buffer is replaced by port A of the 8255A. When a pushbutton key is pressed, it bounces (makes and breaks contact) a few times before it makes a firm contact. To prevent multiple readings of the same key, it is necessary to debounce the key. The hardware solution to this problem is to use a key debounce circuit (cross-coupled NAND or NOR gates), and the software solution is to wait for 10 to 20 ms until the key is settled and then check the key again. The display circuit in Figure 15.15 uses a common-cathode seven-segment LED, connected to port B of the 8255A. To display a digit, it is necessary to turn to the appropriate segments of the LED. The appropriate binary code can be obtained by using the table look-up technique, described in Section 15.2.4. The programming of this problem can be divided into the following categories: