INTERFACING I/O PORTS :-

* I/o ports are the communication channel of the microprocessor with the external devices.
* Input port - read data from keyboard, mouse etc.
* Output port - send data to output devices (CRT display)

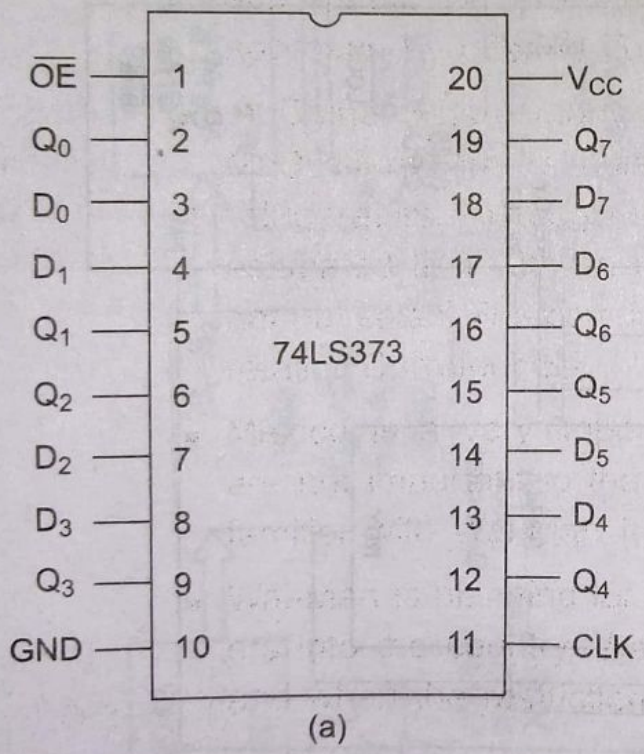
Steps in Interfacing an I/o device :-

→ Connect the data bus of the microprocessor to the data bus of the I/o port.
→ Derive a device address pulse by decoding the required address of the device and use it as chip select.
→ Use a suitable control signal i.e. -
connect $\overline{IOKD}$ to $\overline{RD}$ input of the device or
connect $\overline{IOWR}$ to $\overline{WR}$ input of the device.

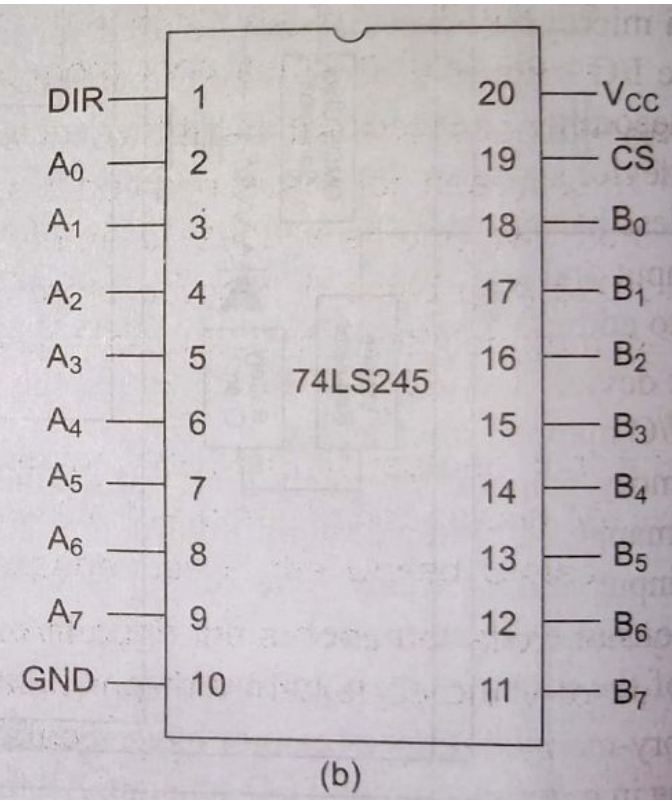
Methods of Interfacing I/o devices :-

(i) I/o mapped
(ii) memory - mapped.

I/o mapped I/o	Memory mapped I/o
* Memory and I/o devices have separate address space.	* Both have same address space.
* All address can be used by the memory	* memory is less as memory space is shared with I/o devices.
* Separate control instruction for read/write of I/o devices and memory	* same instructions can control both I/o and memory
* Efficient due to separate buses	* Less efficient.
* Complex hardware as separate logic is used to control both.	* Simpler logic is used as I/o is also treated as memory only.



(a)



(b)

Fig. 5.11 *(a) Latch (O/P port) (b) Buffer (I/P port)*

Problem 5.6

Interface an input port 74LS245 to read the status of switches SW_1 to SW_8. The switches, when shorted, input a '1' else input a '0' to the microprocessor system. Store the status in register BL. The address of the port is 0740H.

Solution The hardware interface circuit is shown in Fig. 5.12. The address, control and data lines are assumed to be readily available at the microprocessor system.

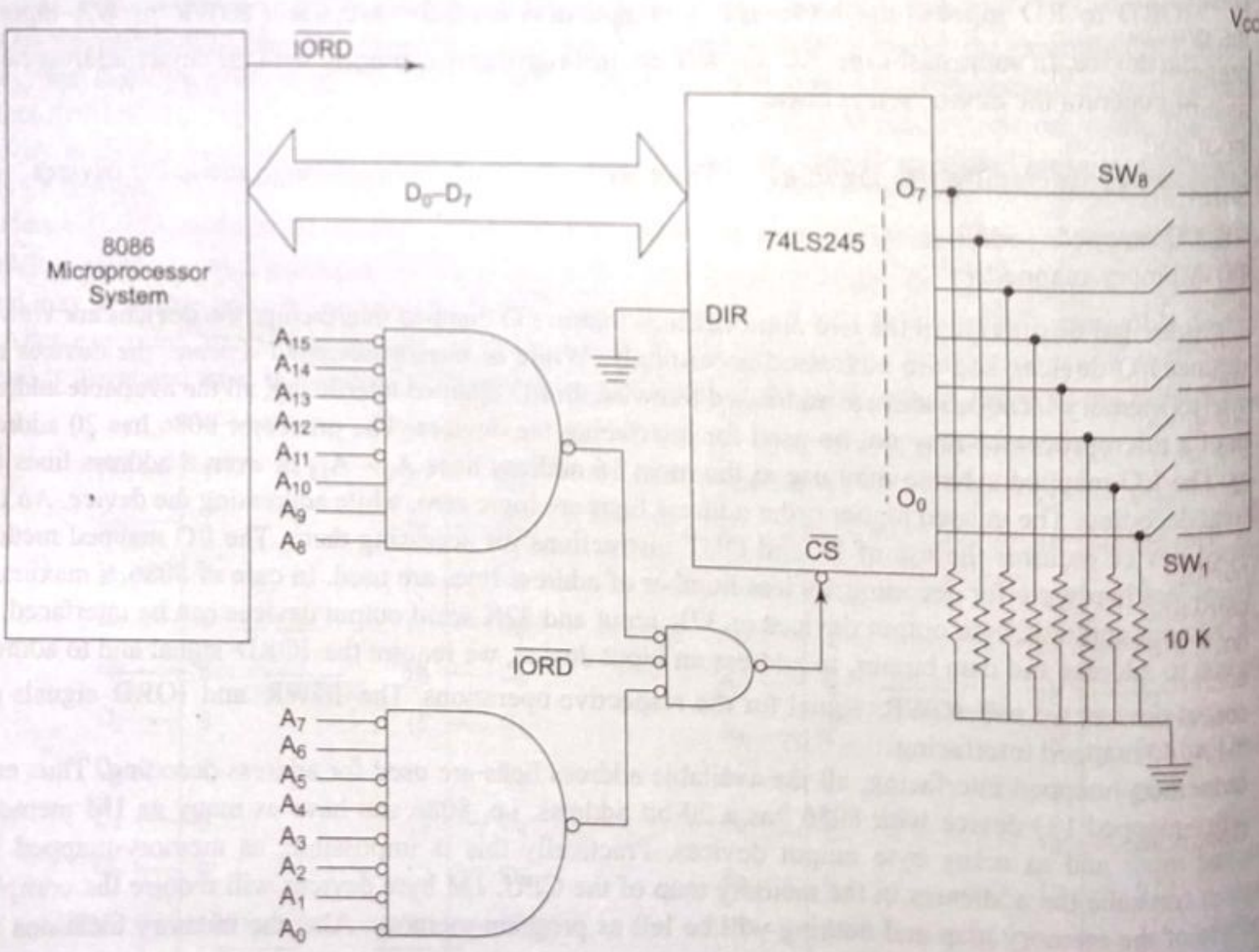


Fig. 5.12 *Interfacing Input Port 74LS245*

```
THE ALP IS GIVEN AS FOLLOWS:
MOV BL, 00H        ; Clear BL for status
MOV DX, 0740H      ; 16-bit port addrenss in DX
IN AL, DX          ; Read port 0740H for switch positions
MOV BL, AL         ; Store status of switches from AL into BL
HLT                ; Stop
```

Program 5.1 *ALP for Problem 5.6.*

Here LSB bit of BL corresponds to the status of SW_1 and likewise the MSB of BL corresponds to the status of SW_8. The '1' indicates 'on' or shorted switch and the '0' indicates an 'off' or opened switch. The pull-up registers in Fig. 5.12 are necessary because the open switches should input a '0' to the system but the TTL port 74LS245 will read the free input as '1'. (Free TTL inputs are always read as logic '1'.)

Problem 5.7

Design an interface of an input port 74LS245 to read the status of switches SW_1 to SW_8 (as in the previous problem), and an output port 74LS373 with 8086. Display the number of a key that is pressed, i.e. from 1 to 8 on a 7-seg display with help of the output port. Write an ALP for this task, assume that only one key is pressed at a time. Draw the schematic of the required hardware. The input port address is 0008H and the output port address is 000AH.

Solution In the previous problem, one might have noted that a lot of hardware is required to decode the port address absolutely. Thus instead of decoding the address completely, only a part of it may be decoded. For example, instead of using 16 address lines A_{15}–A_0, one may use only A_3–A_0. In this problem, the address 0008H may then be converted to xxx8H, where x denotes a don't care condition. Thus the port may have more than one address, for example 2358H, 1728H etc. Only the least significant nibble of the address needs to be 8H. The disadvantage of the scheme is that there are a number of addresses of the same port. Hence, the system must have only one port that has the lowest nibble address 8H, otherwise, the system may malfunction. Thus for smaller systems containing a few I/O ports, this scheme is suitable and advantageous as it requires less hardware.

The status of the switches is first read into the register AL. For displaying the shorted switch number in the 7-seg display, the bit corresponding to the switch is checked by rotating AL through carry and then checking the carry flag. If the carry flag is '1', after one left rotation, it means SW_1 is on. If the carry flag sets after two rotations, SW_2 is on and so on. Register CL is incremented after each rotation so that it contains the pressed switch number. The 8-bit contents of register CL are converted to 7-segment codes by a BCD to 7-seg decoder. The complete hardware (Fig. 5.13) and the ALP is given as shown. Note that both the ports are interfaced at even addresses, i.e. with lower order data bus $D_0 - D_7$.

Common cathode displays are used along with corresponding BCD to 7-seg decoder.

```
MOV BL, 00    ; Clear BL for switch status
MOV CL, 00    ; Clear CL for switch number
XOR AX, AX    ; Clear accumulators and flag
IN AL,08H     ; Read switch status
```

```
      INC CL          ; Increment CL for 1st switch
YY:   RCR AL          ; Rotate switch status
      JC XX           ; If carry, halt,
      INC CL          ; else increment CL for next switch
      JMP YY          ; number till carry is 1
XX:   MOV AL,CL       ; Take switch number into AL
      OUT 0AH,AL      ; Out BCD switch number for display
      HLT             ; Stop
```

Program 5.2 *ALP for Problem 5.7*

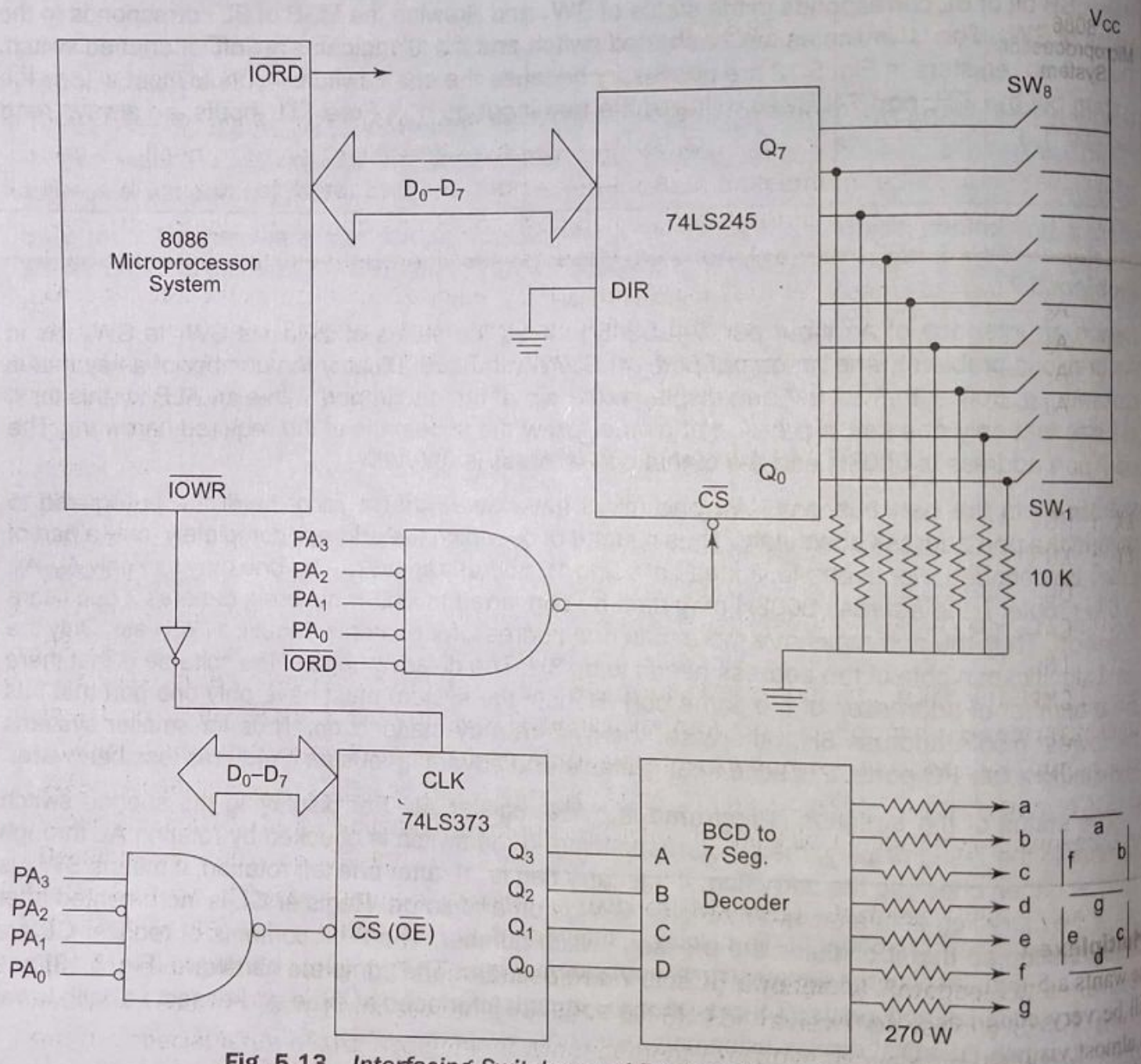


Fig. 5.13 *Interfacing Switches and Displays for Problem 5.7*

Problem 5.8

Using 74LS373 output ports and 7-segment displays, design a seconds counter that counts from 0 to 9. Draw the suitable hardware schematic and write an ALP for this problem. Assume that a delay of 1sec is available as a subroutine. Select the port address suitably.

Solution The counter hardware is shown in Fig. 5.14. Common cathode displays are used along with a suitable BCD to 7-segment decoder. The ALP calls the subroutine 'DELAY' that generates a delay program of 1sec. After counting from 0 to 9, it again starts from 0. The output port is interfaced at address 0008H.

The ALP for generating the seconds count is given below.

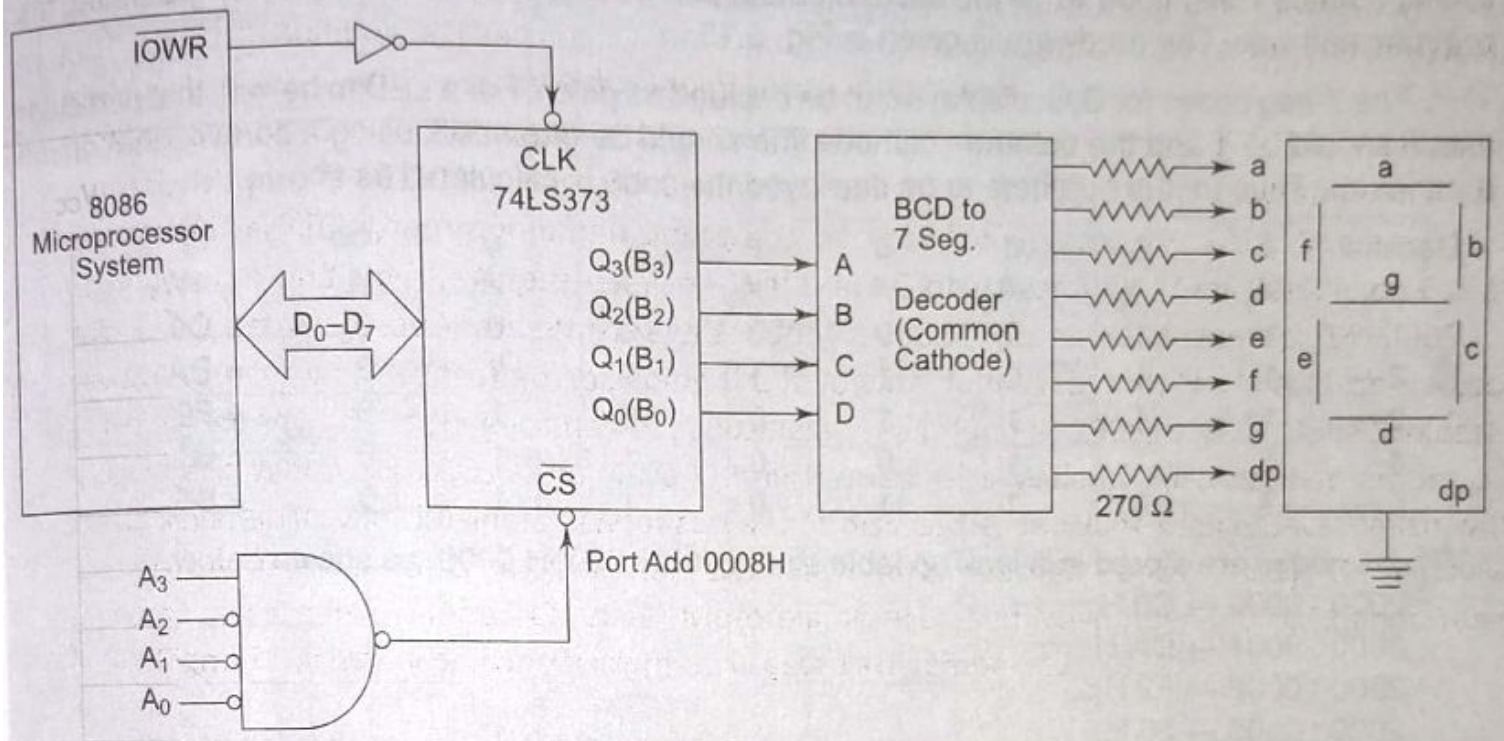


Fig. 5.14 *Interfacing Circuit Diagram for Problem 5.8*

```
XX : MOV AL,00H          ; Start from 00
YY : XOR AL,AL           ; Clear AL and flags
     OUT 08H,AL          ; Display 0H
     CALL DELAY          ; Wait for one second
     INC AL              ; Increment count for next
     CMP AL,0A H         ; display. Is it above 9H?
     JZ XX               ; If yes, start from 00,
     JMP YY              ; else continue.
```

Program 5.3 *ALP for Problem 5.8*

Multiplexed 7-seg Displays To display a single digit, at least one output port is required. Suppose one wants a 5-digit display for some practical application, Five such ports will be required, i.e. the hardware will be very complex and costly. To minimize the complexity and cost of hardware one may implement an almost visually identical display using only two ports. Suppose there are four 7-seg displays, numbered 1, 2, 3 and 4. One of the two ports, say port1 selects one of these displays, say display 1 at a time, while port 2 sends the data to be displayed (i.e. a,b,c,d,e,f,g and dp) to the first display for a fixed short duration. Port 1 next, selects the display 2 and port 2 sends the appropriate display data to it. Each of the display unit remains active for a short duration and the process continues in a loop. This is repeated at a high frequency so that the complete display containing more than one7-seg display appears to be stationary due to the persistence of vision. Instead of BCD to 7-seg decoder circuit, look up table technique may be used for converting BCD numbers to equivalent 7-seg codes.

Problem 5.9

Draw a schematic hardware circuit for interfacing five, 7-seg displays (common cathode) with 8086 using output ports. Display numbers 1 to 5 on them continuously. The 7-seg codes are stored in a look-up table serially at the address 2000:0000 H onwards starting from code for 1.

Solution Let us select the two port addresses 0004H and 0008H for the output ports. The first port 0004H outputs 7-seg code while the second output port 0008H selects the display by grounding the common cathode. The hardware is given in Fig. 5.15.

The 7-seg codes for C.C. displays can be decided as given. For a LED to be 'on', that particular anode should be 1 and the common cathode line should be grounded, using a port line that drives a transistor. Thus for the numbers to be displayed the code is calculated as shown.

Decimal	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*dp*	
no.	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
1—	1	1	0	0	0	0	0	0	= C0
2—	1	1	0	1	1	0	1	0	= DA
3—	1	1	1	1	0	0	1	0	= F2
4—	0	1	1	0	0	1	1	0	= 66
5—	1	0	1	1	0	1	1	0	= B6

These codes are stored in a look up table starting from 2000H:0000, as shown below.

2000 : 0000 → C0 H
2000 : 0001 → DA H
2000 : 0002 → F2 H
2000 : 0003 → 66 H
2000 : 0004 → B6 H

Only one display should be selected at a time, i.e. only the corresponding bit of port 2 should be high for selecting a common cathode display. All the other bits should be low to keep the other displays disabled. Thus to enable the least significant display, the LSB of the 8-bit selected port should remain '1'. Hence AL should have 01 or E1H in it to select the least significant display. The codes for the selection of displays and 7-segment codes directly depend upon the hardwared connections between them.

```
        MOV AX, 2000H          ; Initialize pointer to
        MOV DS, AX             ; Code table DS:BX
        MOV BX, 0000H
NEXT :  MOV AL, 00H            ; Get 1st number from the table.
        MOV DH,AL
        MOV CL, 05H            ; Count for display
        MOV DL, E1H            ; Selection code for 1st display
AGAIN : XLAT                   ;
        OUT 04H, AL            ; Out the code for the first
                               ; number to port 04H.
        MOV AL,DL              ; Get to be enabled display code.
        OUT 08H,AL             ; Select 1st display.
        ROL DL                 ; decide code for selecting next
        INC DH                 ; display for next number
        MOV AL,DH              ; get next num. to be displayed.
        LOOP AGAIN             ; Repeat five times
        JMP NEXT               ; Continue the procedure
```

Program 5.4 *ALP for Problem 5.9*

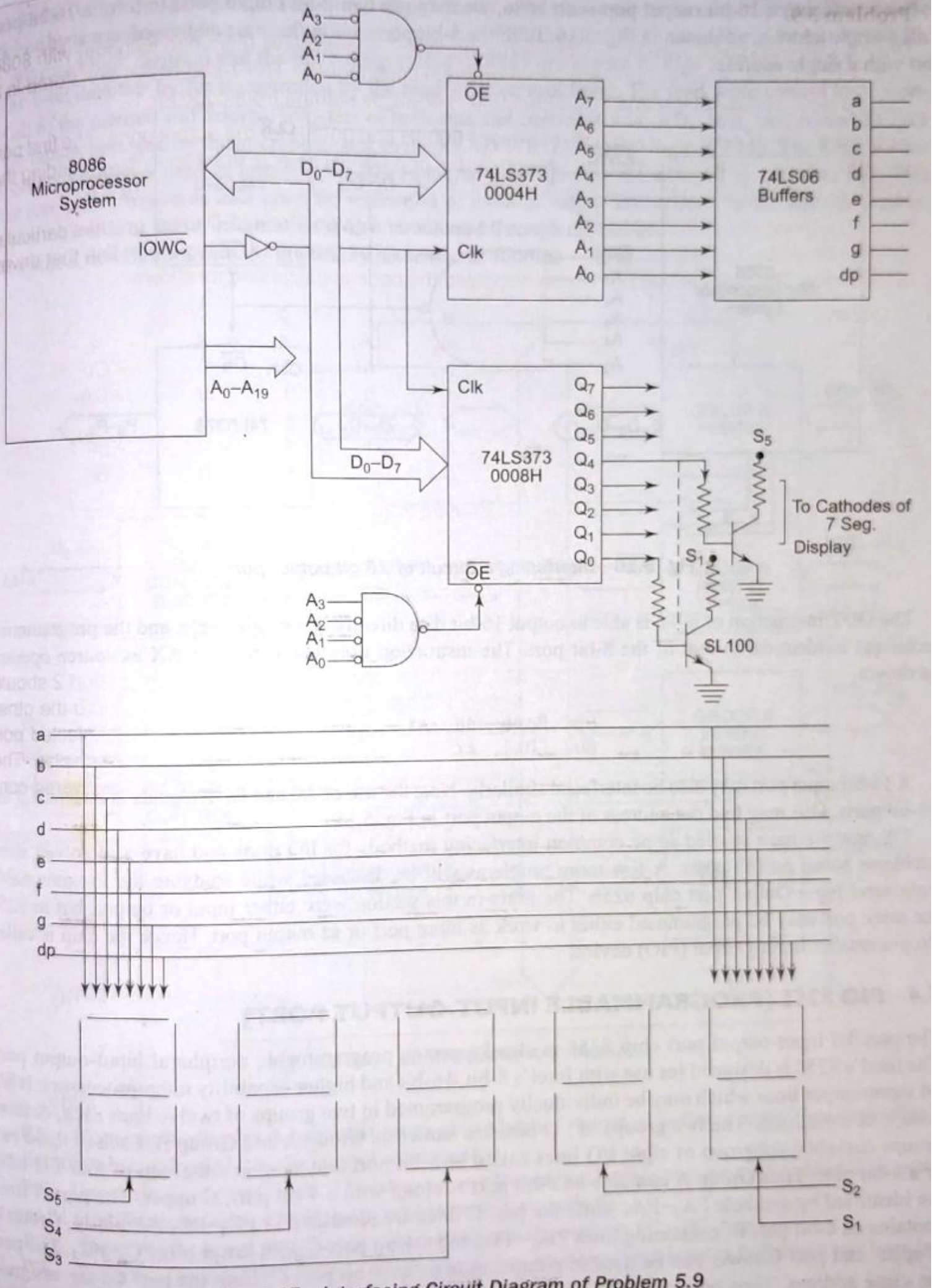


Fig. 5.15 *Interfacing Circuit Diagram of Problem 5.9*

For interfacing a 16-bit output port with 8086, we may use two 8-bit output ports to form a 16-bit port, with a single address, as shown in Fig. 5.16. Both the 8-bit ports are in this case addressed as a single 16-bit port with a single address.

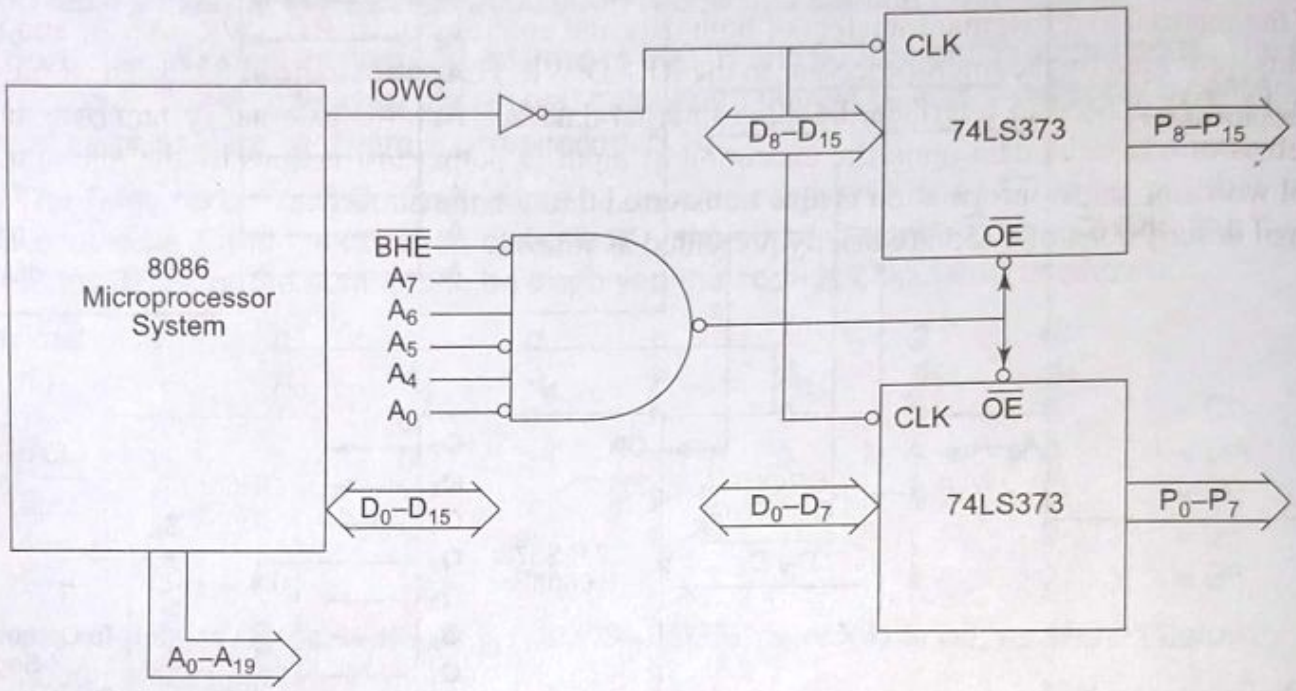


Fig. 5.16 *Interfacing a Circuit of 16 bit output port*

PIO 8255 : (Programmable I/o port)

* Also known as programmable peripheral input-output port.
* can be used with 8-bit or 16-bit or higher capacity microprocessors.
* has 24 input/output lines.
* 24 lines divided into 2 groups:

  Group A → 12 lines (Port A → 8 bits
  Port Cupper → 4 bits)

  Group B → 12 lines (Port B → 8 bits
  Port Clower → 4bits)

* Hence, there can be 3 8-bit ports i.e. Port A, Port B and Port C (Cupper and Clower combined) or there can be 2 8-bits ports (A & B) and 2 4-bit ports (Cupper and Clower).
* All the above ports can function independently and can be programmed using an internal register of P255 called Control word Register (CWR)

⇒ PIN diagram of 8255 and internal architecture attached..

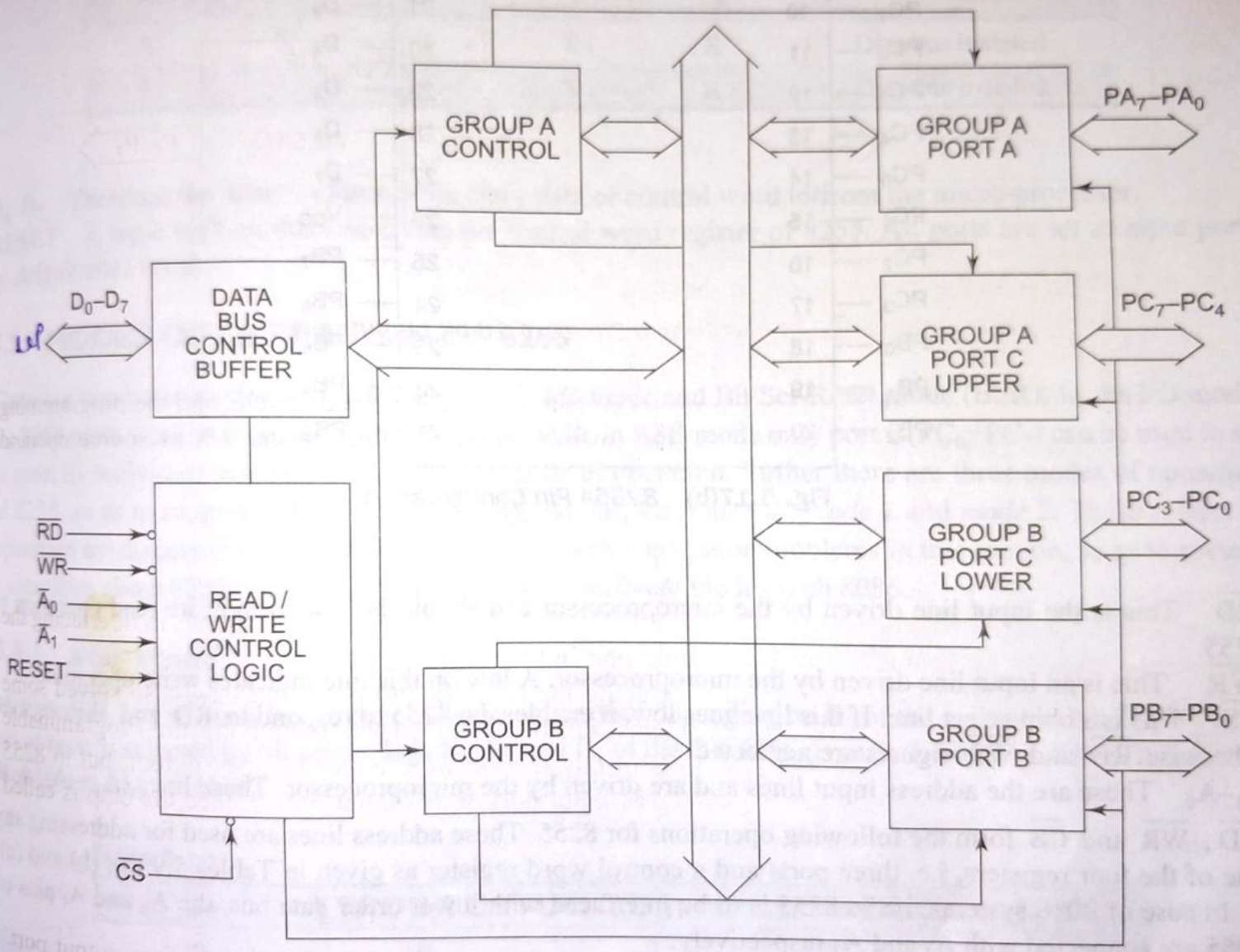


Fig. 5.17(a) *8255 Internal Architecture*

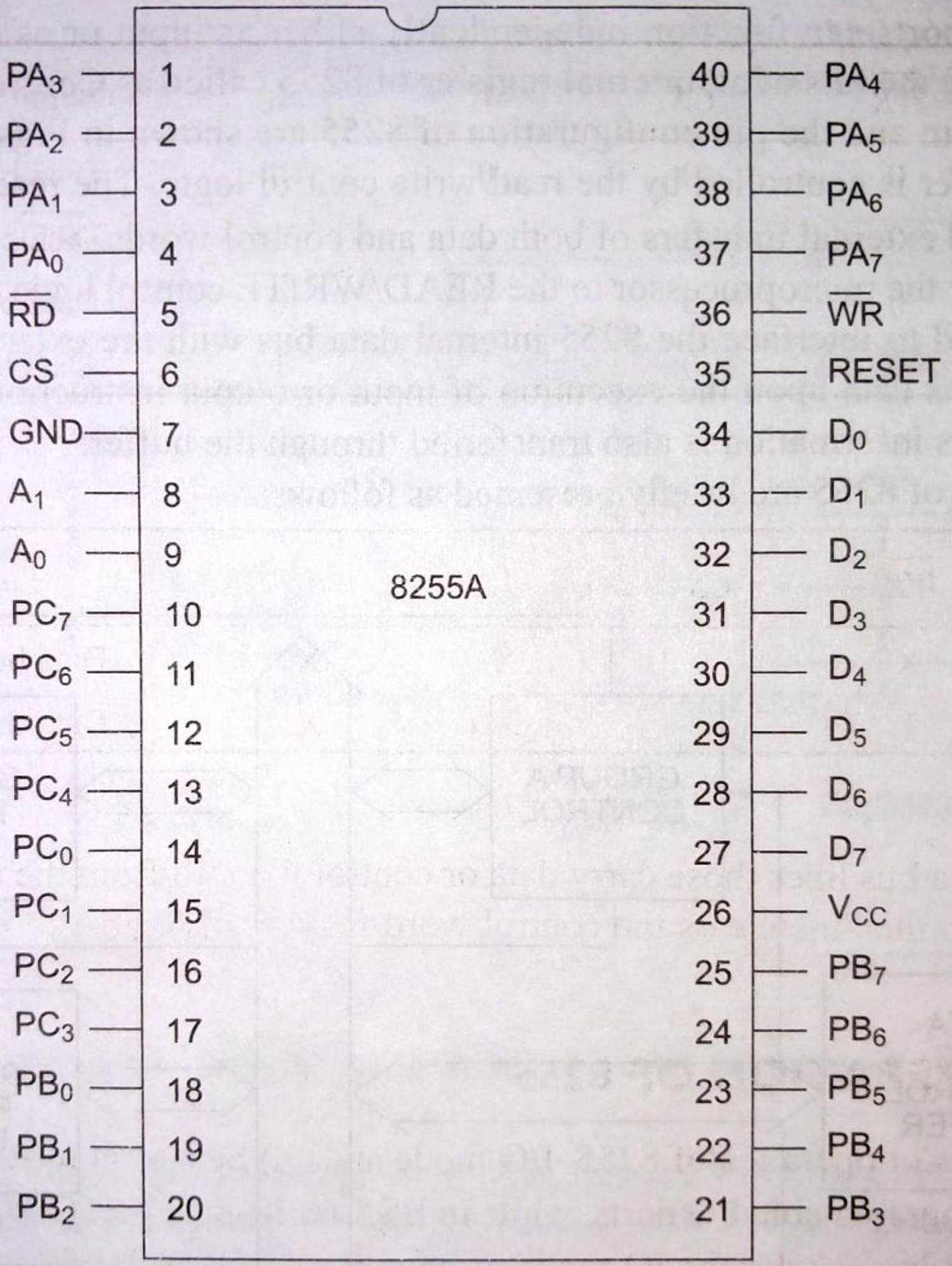


Fig. 5.17(b) *8255A Pin Configuration*

$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	A_1	A_0	Input (Read) cycle
0	1	0	0	0	Port A to data bus
0	1	0	0	1	" B " "
0	1	0	1	0	" C " "
0	1	0	1	1	CWR to data bus.

$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	$\overline{A_1}$	A_0	Output (write) cycle
1	0	0	0	0	Data bus to port A
1	0	0	0	1	" " " " B
1	0	0	1	0	" " " " C
1	0	0	1	1	" " " CWR

If $\overline{CS} = 1$, $\overline{RD} = \overline{WR} = X$ (Don't care) } then data bus tristated.
$A_1 = A_0 = X$

Also, if $\overline{CS} = 0$, $\overline{RD} = \overline{WR} = 1$ } then data bus tristated.
$A_1 = A_0 = X$

MODES of OPERATION of 8255 :-

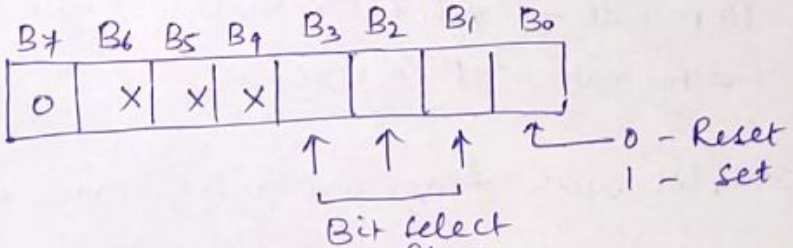
* Basically, there are 2 modes of operation of 8255:

(a) BSR mode (Bit set - Reset mode)

(b) I/O mode → mode 0, mode 1, mode 2

(1) BSR mode :-

→ Any of the 8-bit of Port C can be set or reset depending on B_0 of the control word.



Control Word Register format in BSR Mode

* The bit to be set or reset is selected by bit select flags B_3, B_2 and B_1 of CWR.

B_3	B_2	B_1	Selected bits of Port C
0	0	0	→ Bit - 0 (LSB)
0	0	1	→ 1
0	1	0	→ 2
0	1	1	→ 3
1	0	0	→ 4
1	0	1	→ 5
1	1	0	→ 6
1	1	1	→ 7 (MSB).

I/O MODES:-

(a) MODE 0 (Basic I/O mode)

→ Simple input and output capability using each of the three ports.

→ Two 8-bit ports and 2 - 4bit ports are available. The 2 4-bit ports can be combined to form a third 8-bit port.

→ o/p ports are latched.
i/p ports are not latched.

→ 4 ports available ⇒ 16 I/O configurations available.

* All the modes are selected by programming a register internal to 8255 known as control word Register which has 2 formats.

1st Format — Valid I/O mode - 0, 1 or 2

2nd format - BSR mode.

⇒ Examples with diagrams to be provided.

MODE 0 (Basic I/O mode) This mode is also known as *basic input/output mode*. This mode provides simple input and output capability using each of the three ports. Data can be simply read from and written to the input and output ports respectively, after appropriate initialisation.

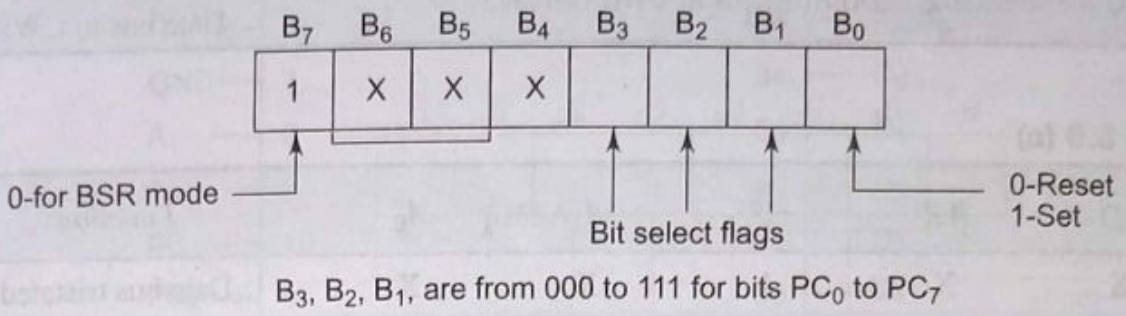


Fig. 5.18(a) *BSR Mode Control Word Register Format*

The salient features of this mode are as listed below:

(i) Two 8-bit ports (port A and port B) and two 4-bit ports (port C upper and lower) are available. The two 4-bit ports can be combinedly used as a third 8-bit port.
(ii) Any port can be used as an input or output port.
(iii) Output ports are latched. Input ports are not latched.
(iv) A maximum of four ports are available so that overall 16 I/O configurations are possible.

All these modes can be selected by programming a register internal to 8255, known as Control Word Register (CWR) which has two formats. The first format is valid for I/O modes of operation, i.e. modes 0, mode 1 and mode 2 while the second format is valid for bit set/reset (BSR) mode of operation. This format is shown in Fig. 5.18(b).

Now let us consider some interfacing problems so as to elaborate the hardware interfacing and I/O programming ideas using 8255 in mode 0.

Problem 5.10

Interface an 8255 with 8086 to work as an I/O port. Initialize port A as output port, port B as input port and port C as output port. Port A address should be 0740H. Write a program to sense switch positions SW_0–SW_7 connected at port B. The sensed pattern is to be displayed on port A, to which 8 LEDs are connected, while the port C lower displays number of on switches out of the total eight switches.

Solution The control word is decided upon as follows:

B7	B_6	B_5	B_4	B_3	B_2	B_1	B_0	Control word
1	0	0	0	0	0	1	0	= 82H
I/O mode	Port A in mode 0		Port A,o/p	Port C,o/p	Port B,mode 0	Port B,i/p	Port C,o/p	