Interfacing I/O Ports:

I/O ports or input/output ports are devices through which the microprocessor communicates with other devices or external data sources/ destinations.

Input activity is the activity that enables the microprocessor to read data from external devices like keyboards, joy sticks, mouse etc.

Output activity transfers data from the microprocessor to external devices like CRT display, 7-segment displays, printers etc.

Note that an input device can only be read and an output device can only be written.

→ $\overline{IORD}$ operation is related with reading data from an input device and not an output device.

$\overline{IOWR}$ operation is related to writing data to an output device and not an input device.

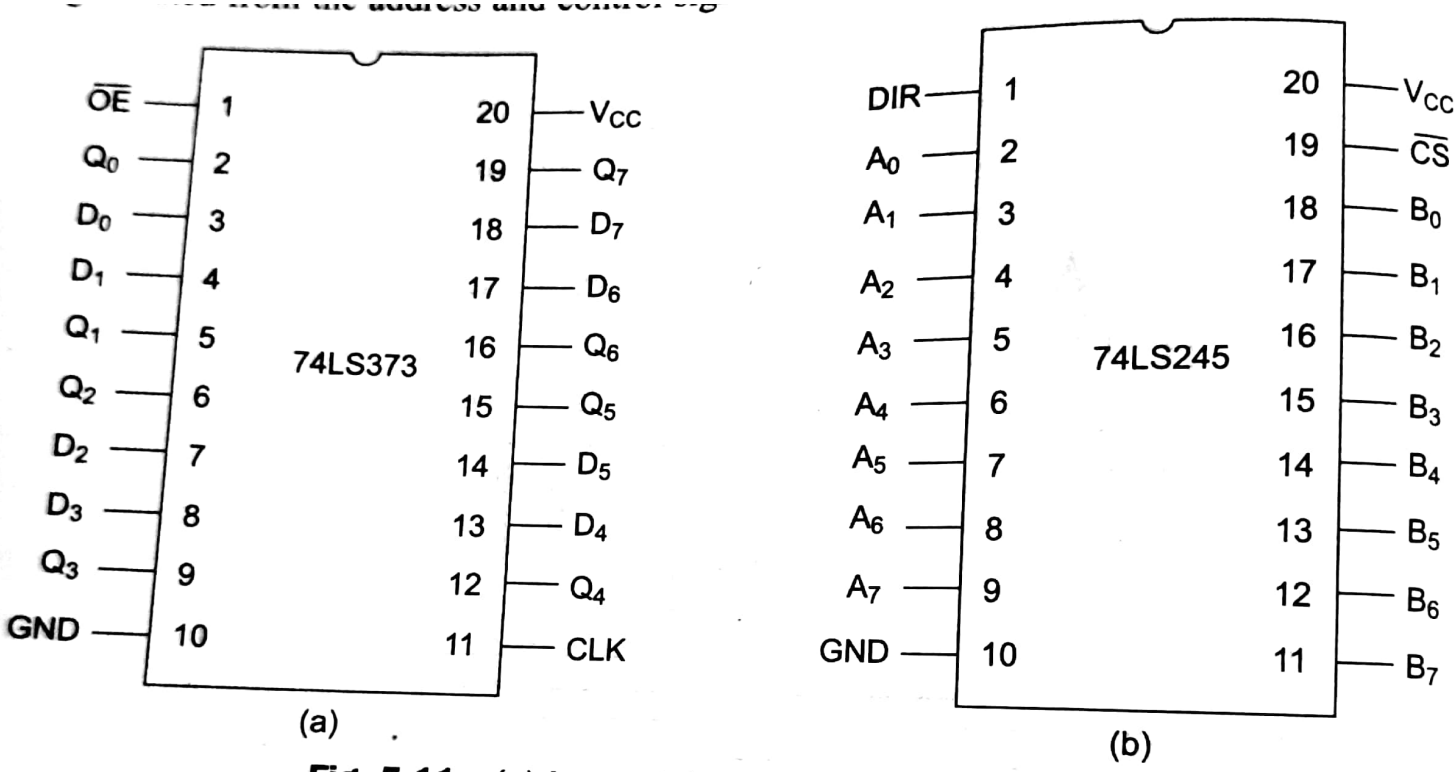

Fig. 5.11 *(a) Latch (O/P port) (b) Buffer (I/P port)*

→ The chip 74LS373, contains 8 buffered latches and can be used as an 8 bit output port.

→ The chip 74LS245 contains 8 buffers and may be used as an 8 bit input port.

74LS245 is a bidirectional buffer, but while using it as an input device, only one direction is useful. This direction of data transfer is selected by using its DIR pin.

DIR = 1 then direction is from A (I/Ps) to B (O/P's)
otherwise direction is from B (I/Ps) to A (O/P's).

Steps in interfacing an I/O device:

(1) Connect the data bus of the microprocessor system with the data bus of the I/O port.

(2) Derive a device address pulse by decoding the required address of the device and use it as the chip select of the device.

(3) Use a suitable control signal, ie. $\overline{IORD}$ or $\overline{IOWR}$ to carry out device operations.

ie. Connect $\overline{IORD}$ to $\overline{RD}$ input of the device if it is an input device

or connect $\overline{IOWR}$ to $\overline{WR}$ input of the device, it is an output device.

In some cases, $\overline{RD}$ or $\overline{WR}$ control signals are combined with device address pulse to generate the device select pulse.

Methods of Interfacing I/O devices

There are two methods
(i) I/O mapped (ii) memory mapped

The main difference between the two approaches is that in I/O mapped interfacing the devices are viewed as distinct I/O devices and are addressed accordingly.
While in memory-mapped scheme, the devices are viewed as memory locations and are addressed accordingly.

I/O mapped Addressing:

→ The 8086 processor has 20 address lines. The I/O mapped addressing scheme may use at the most 16 address lines A_0-A_{15} or even 8 address lines for address decoding.

→ The unused higher order address lines are zero, while addressing the device.

→ An I/O mapped devices requires the use of IN and OUT instructions for accessing them.

→ I/O mapped method requires less hardware for decoding as less address lines are used.

→ A maximum of 64 k input and 64k byte output devices or 32 k input and 32 k word output devices can be interfaced.

→ In addition to address and data lines, to address an input device, we require $\overline{IORD}$ signal, and $\overline{IOWR}$ to address an output device. The $\overline{IORD}$ and $\overline{IOWR}$ signals are used for I/O mapped interfacing.

→ I/O mapped devices are slow in operation.

Memory mapped Interfacing:

→ All the available (20) address lines are used for address decoding.
→ Complex hardware for decoding is required.
→ Each memory mapped I/O device has a 20 bit address i.e 8086 can have as many as 1M memory mapped input and as many byte output devices.

Practically, this is impossible, as memory mapped devices consume the address as the memory map of the CPU and nothing will be left as program memory.

→ Also the memory locations and memory mapped devices cannot have the same addresses.

→ The $\overline{MRDC}$ and $\overline{MRTC}$ signals are used for interfacing in memory mapped I/O scheme.

→ All applicable data transfer instructions (MOV, LEA etc) can be used to communicate with memory mapped I/O devices.

→ This scheme is faster in operation.

→ In 8086 based systems, the memory mapped scheme is not used. Hence all the peripherals are essentially I/O mapped devices.

→ 8086 has a 16 bit data bus, hence interfacing of 8 bit devices with 8086 need special consideration.

→ Usually 8 bit I/O devices are interfaced with lower order data bus of 8086 i.e. $D_7 - D_0$.

The 16 bit devices are interfaced directly with 16 bit data bus using A_0 and $\overline{BHE}$ pins of 8086.

→ We will see some example problems explaining the actual method of interfacing I/O devices with 8086.

The interfacing hardware always needs supporting application program to carry out the desired operation.

Ex:-

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.

Sol

→ The address, control and data lines of microprocessor are readily available for interfacing.

→ As I/O mapped interfacing uses 16 address lines. Address lines $A_0 - A_{15}$ can be used for decoding of the chip select of the 74LS245 IC buffer.

→ DIR should be made '0' as it has to accept some input data.

are assumed to be readily available at

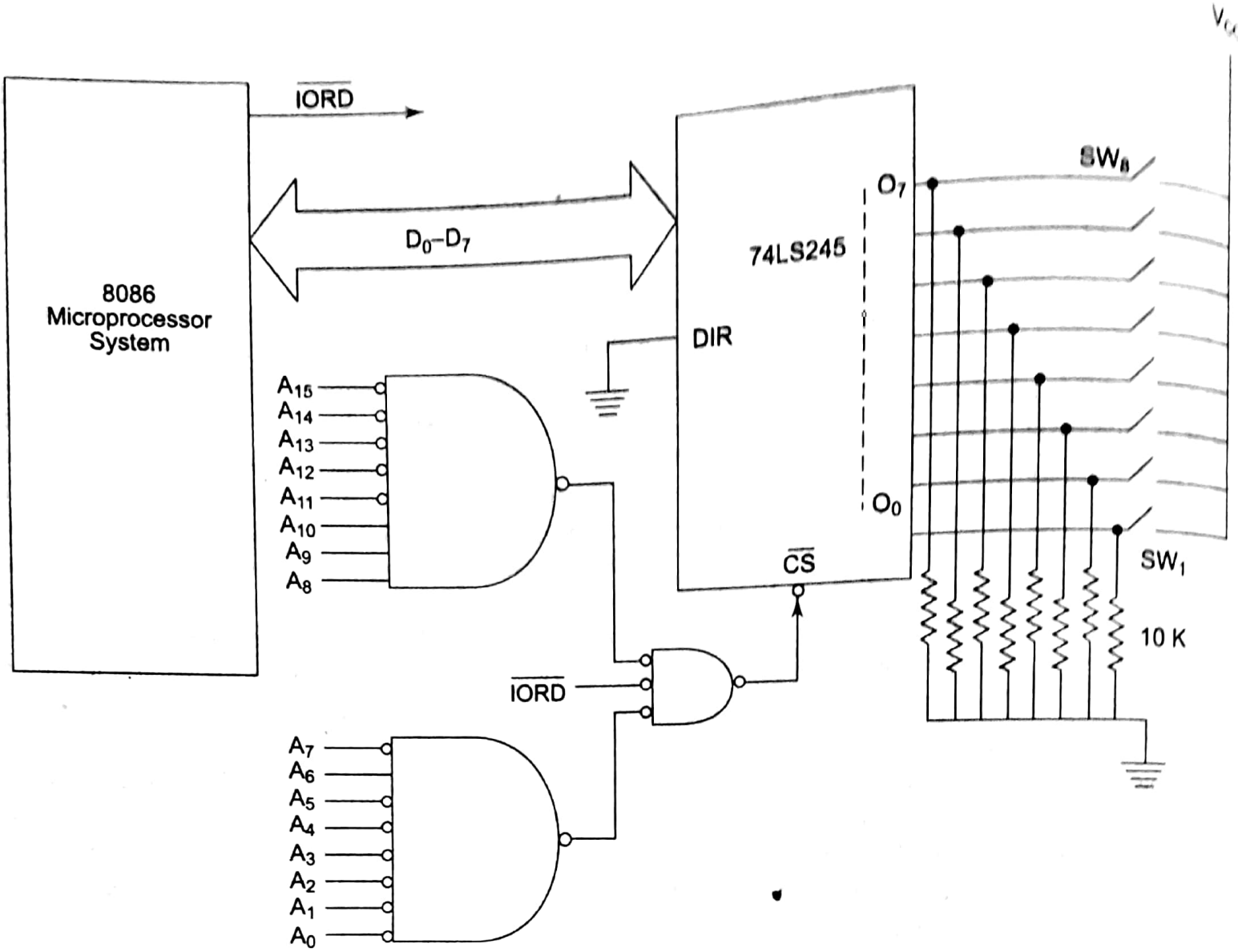

Fig. 5.12 *Interfacing Input Port 74LS245*

The ALP for the above interfacing circuit can be :

```
MOV   BL, 00H   ; clear BL for status
MOV   DX, 0740H ; 16 bit port address in DX
IN    AL, DX    ; read port 0740H for switch positions
MOV   BL, AL    ; store status of switches from AL to BL
HLT
```

Here LSB bit of BL corresponds to status of SW_1 and MSB of BL corresponds to status of SW_8.

The pull up resistors are necessary because the open switches should input a '0' to the system, but TTL port 74LS245 will read the free input as '1'.

Ex:-

Design an interface of an input port 74LS245 to read the status of switches SW_1 to SW_8 as in 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 segment display with the help of the output port. Also 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 output port address is 000AH.

Sol

In the previous problem, we have need all the 16 address lines for enabling the chip select pin of the I/O buffer IC.

In this case if we observe the address given to input port and output port all the higher order address line are 0 except the lower 4 lines of the address line $A_0 - A_3$.

∴ Here we can use only the least significant nibble of the address for enabling the chip select pin of the I/P port buffer IC.

The port may have more than one address for example 2358H, 1728H etc.

∴ we can ~~add~~ convert the address 0008H as xxx8H where x denotes a don't care condition.

The disadvantage of this method is that there can be multiple addresses of the same port.

Hence to use this scheme for the chip select enable, the system must have only one port that has the lowest nibble as 8H, otherwise the system may malfunction.

Thus, for small systems, containing fewer I/O ports, this scheme is suitable and advantageous as it requires less hardware.

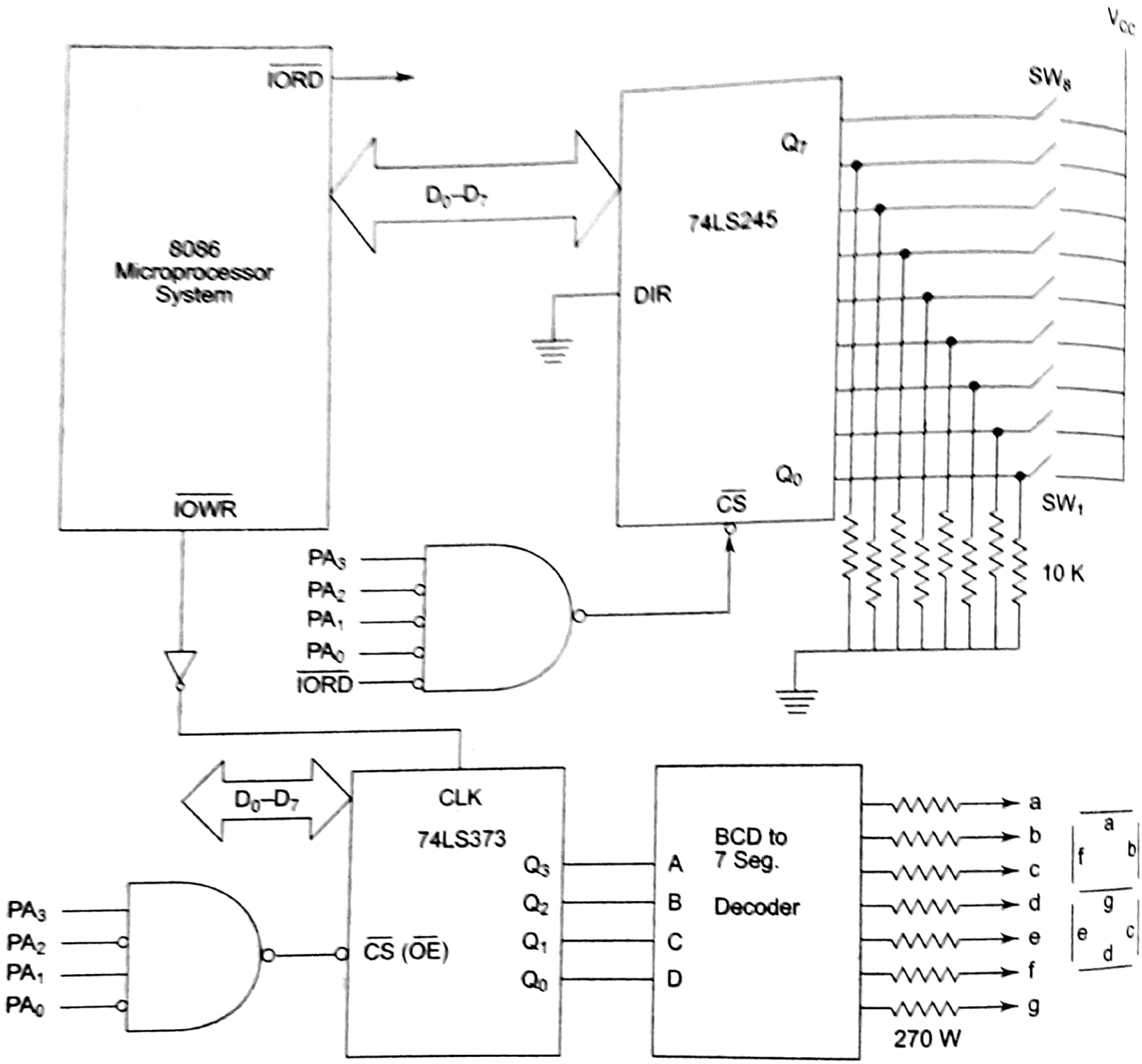

Fig. 5.13 *Interfacing Switches and Displays for Problem 5.7*

The ALP for the above interfacing circuit can be :

```
        MOV  BL, 00  ; clear BL for switch status
        MOV  CL, 00  ; clear CL for switch number
        XOR  AX, AX  ; clear Acc. and flags.
        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      ; increment CL for next switch
        JMP  YY      ; till carry is 1
XX :    MOV  AL, CL  ; Take switch no. in AL
        MOV  0AH, AL ; Out BCD switch no. for output display
        HLT
```

Q:

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 1 sec is available as a subroutine. Select the port address suitably.

Sol

Let us assume that the output port address is 0008H.

Let we use only 4 LSB address lines ($A_3 - A_0$) to enable the $\overline{CS}$ pin of the 74LS373 o/p latch IC.

The circuit accepts the input of the count sequence from the microprocessors and sends to the output buffer and from there to BCD to 7 seg. decoder IC, and from there to enable the 7 segment display device.

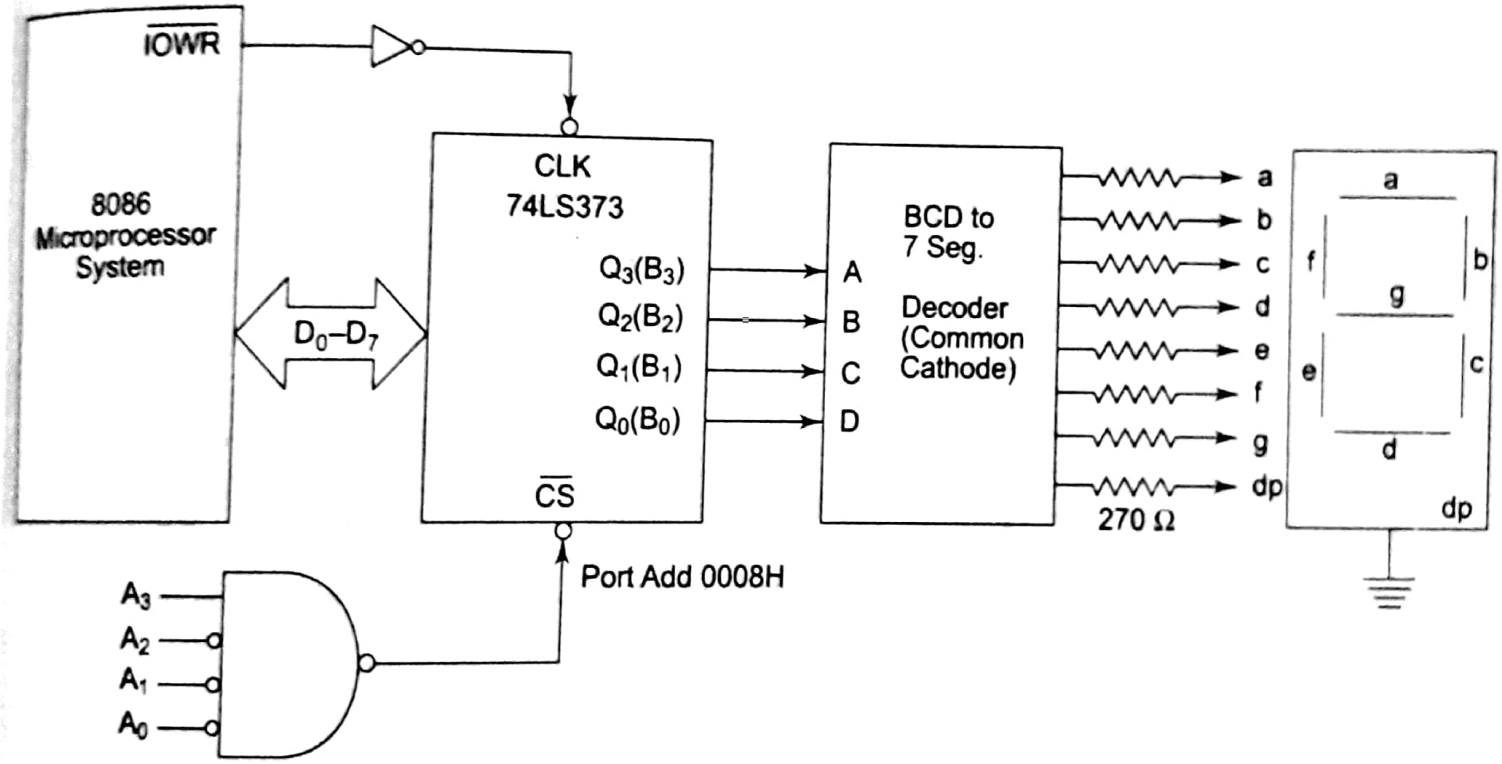

Fig. 5.14 *Interfacing Circuit Diagram for Problem 5.8*

```
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*

Interfacing 16 bit input and output ports to 8086

→ 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.

→ Both the 8 bit ports in this case are addressed as a single 16 bit port with a single address.

⇒ The OUT instruction of 8086 is able to output 16 bit data directly in a single cycle, and the programming technique is identical to that of 8 bit port.

```
OUT   Port_Add , AX

OUT   [DX] , AX
```

→ A 16 bit input port may also be interfaced similarly.

→ A_0 and $\overline{BHE}$ signals are used in interfacing 16 bit

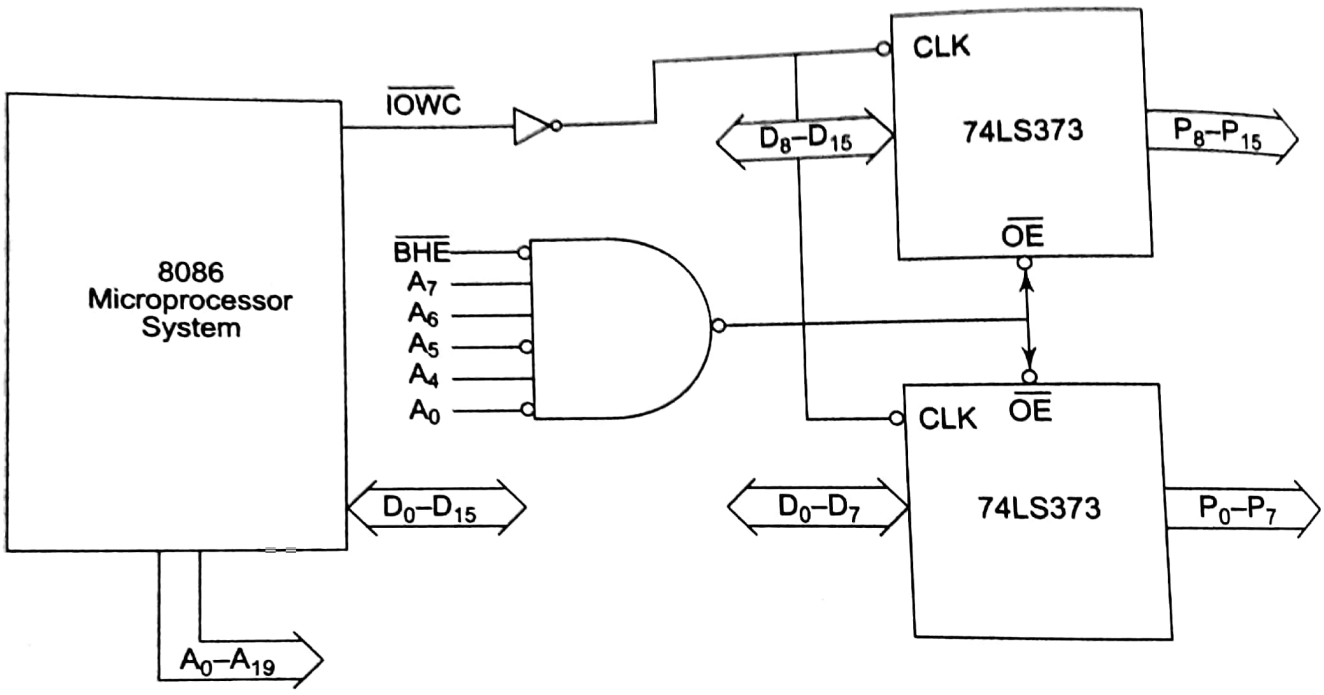

Fig. 5.16 *Interfacing a Circuit of 16 bit output port*