

3

Communication Interface

Syllabus

Basic Peripherals & their interfacing with 8086/8088, Semiconductor Memory Interfacing-Dynamic RAM Interfacing-Interfacing I/O ports-PIO-8255, Modes of operation-interfacing Analog-Digital Data converter-stepper motor interfacing.

Contents

- 3.1 Basic Peripherals and their Interfacing with 8086/8088
- 3.2 Semiconductor Memory Interfacing
- 3.3 Dynamic RAM Interfacing
- 3.4 Programmable Peripheral Interface (PPI) - 8255
- 3.5 Digital to Analog Interface
- 3.6 Analog to Digital Interface
- 3.7 Stepper Motor Interfacing

Multiple Choice Questions

3.1 Basic Peripherals and their Interfacing with 8086/8088

- The most of the microprocessors support isolated I/O system. It partitions memory from I/O, via software, by having instructions that specifically access (address) memory and others that specifically access I/O.
- When these instructions are decoded by the microprocessor, an appropriate control signal is generated to activate either memory or I/O operation.
- In 8086, M/ \overline{IO} signal is used for this purpose. The 8086 outputs a logic '0' on the M/ \overline{IO} line for an I/O operation and a logic '1' for memory operation.
- In 8086, it is possible to connect 64 K I/O ports in the system since 8086 can send 16-bit address for I/O. The 8086 gives 8-bit or 16-bit I/O addresses depending on the addressing mode used. This means it can select one of the 256 or 65536 I/O ports.
- Input/output devices can be interfaced with microprocessor systems in two ways:
 - I/O mapped I/O
 - Memory mapped I/O

3.1.1 I/O Mapped I/O

- The 8086 has four special instructions IN, INS, OUT, and OUTS to transfer data through the input/output ports in I/O mapped I/O system. The IN instruction copies data from a port to the accumulator.
- If an 8-bit port is read, the data will go to AL and if an 16-bit port is read the data will goto AX.
- On the other hand, the OUT instruction copies a byte from AL or a word from AX to the specified port.
- The M/ \overline{IO} signal is always low when 8086 is executing these instructions. So M/ \overline{IO} signal is used to generate separate addresses for input/output.
- Only 256 (2^8) I/O addresses can be generated when direct addressing method is used. By using indirect address method this range can be extended up to 65536 (2^{16}) addresses.

3.1.2 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. Here RD and WR signals are activated when M/ \overline{IO} signal is high indicating memory bus cycle.

3.1.3 I/O Device Selection

- To select an appropriate I/O device it is necessary to do following things :
 - Decode the address bus to generate unique signal corresponding to the device address on the address bus.
 - When device address signal and control signal (\overline{IORC} or \overline{IOWC}) both are low, generate device select signal.
 - Use device select signal to activate the interfacing device (I/O port).
- Fig. 3.1.1 shows the absolute decoding circuit for the I/O device. The IC 74LS138, 3:8 decoder along with OR gate is used to generate device address signal. This signal is ORed with control signal (\overline{IORC} or \overline{IOWC}) to generate device select signal.

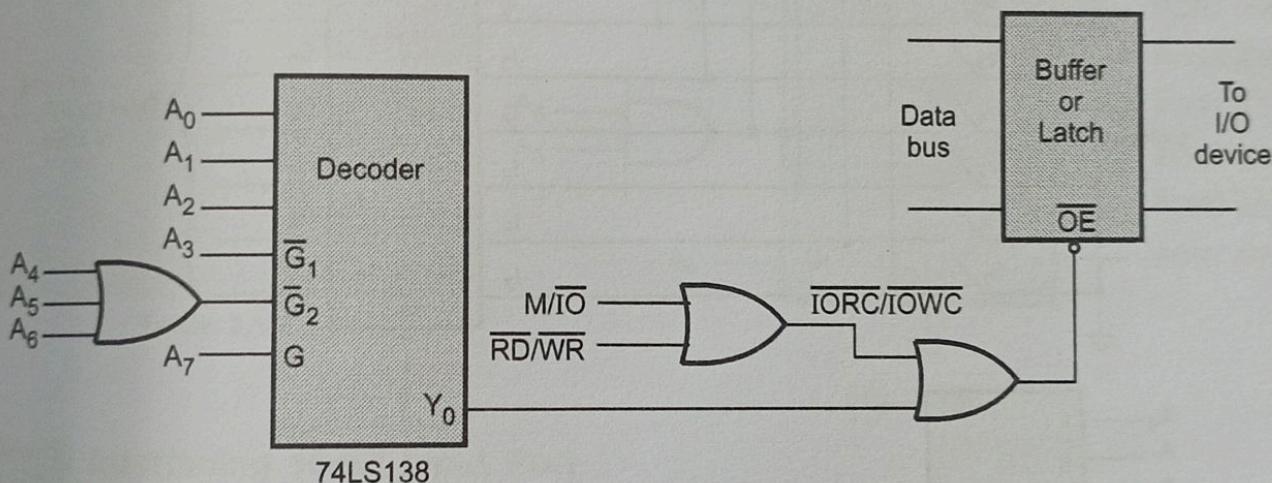


Fig 3.1.1 Absolute decoding circuit for the I/O device

- To generate device select signal (Y_0) low, the address on the address bus must be as given below :
 - $A_2 A_1 A_0 = 000$; Activates Y_0 output
 - $A_3 A_4 A_5 A_6 = 0000$; Makes \overline{G}_1 and \overline{G}_2 low to enable output of decoder
 - $A_7 = 1$; Makes G high to enable output of decoder

Note Decoder output is enabled only when control signals G_1 and G_2 are low and control signal G is high. Therefore, the address of this I/O device is 80H as shown in the Table 3.1.1.

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Address
1	0	0	0	0	0	0	0	80H

Table 3.1.1

3.1.4 Interfacing 8-bit Input Port

- The microprocessor 8086 accepts 8-bit or 16-bit data from the input port. Let us see interfacing of 8-bit input port. Fig. 3.1.2 shows the circuit diagram to interface input port (buffer) which is used to read the status of 8 switches. The address for this input device is 80H as device select signal goes low when address is 80H.

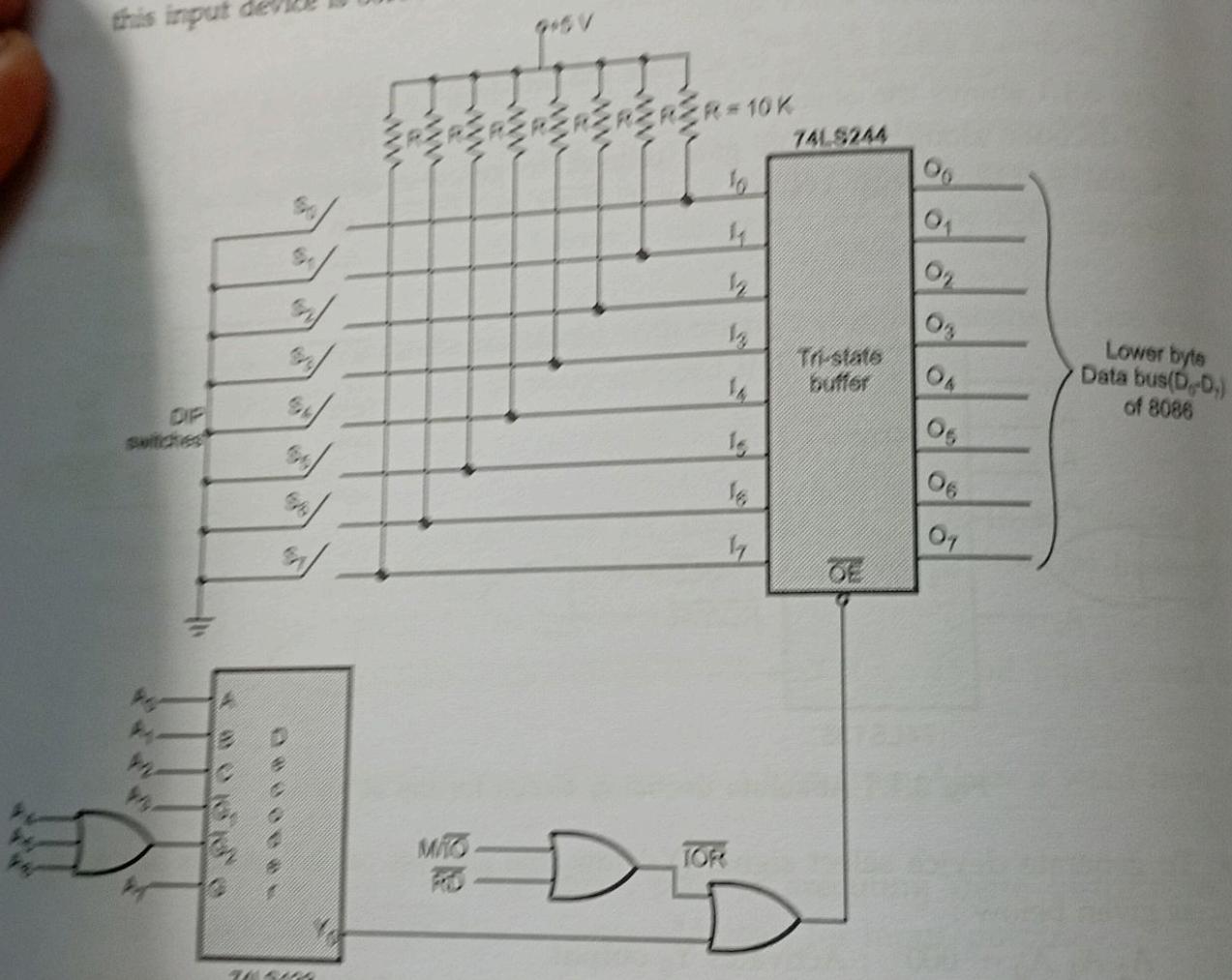


Fig. 3.1.2 Circuit diagram to interface 8-bit input port

- When the switch is in the released position, the status of line is high otherwise status is low. With this information microprocessor can check a particular key is pressed or not.
- The following program checks whether the switch 2 is pressed or not.

Program :

```

    III   AL.80H      ; Read status of all switches
    AND  AL.02H      ; Mask bit positions for other switches
    JZ   NEXT        ; If Z = 0, i.e. switch 2 is pressed then
                      ; program control is transferred to label NEXT.
    NEXT
  
```

Example 3.1.1 Explain 74138 decoder configurations to enable ports at address E8H to EFH.

Solution : Fig. 3.1.3 shows the configuration of 74138 decoder to enable ports at address E8H to EFH. The A₂, A₁, A₀ lines are decoded to generate eight port enable signals. To generate port address range from E8H to EFH 74138 decoder is enabled only when A₇, A₆ and A₅ lines are high and A₄ line is low. This is illustrated in Table. 3.1.2.

Address map

A	A	A	A	A	A	A	A	A	Addr ess
7	6	5	4	3	2	1	0		E8H
1	1	1	0	1	0	0	0		E9H
1	1	1	0	1	0	0	1		EAH
1	1	1	0	1	0	1	0		ABH
1	1	1	0	1	1	0	0		ECH
1	1	1	0	1	1	0	1		EDH
1	1	1	0	1	1	1	0		EEH
1	1	1	0	1	1	1	1		EFH

Table 3.1.2

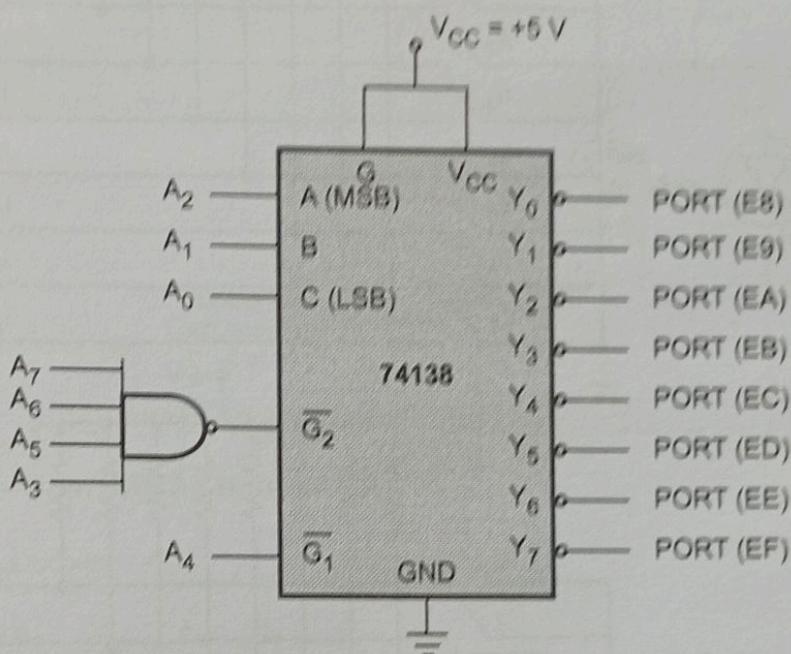


Fig. 3.1.3

3.1.5 Interfacing 16-bit Input Port

- Let us see interfacing of 16-bit input port. Fig. 3.1.4 shows the circuit diagram to interface 16-bit input port (buffer) which is used to read the status of 16 switches. The address for this input port is 80H as device select signal goes low when address is 80H. (See Fig. 3.1.4 on next page)
- The following program checks whether the switch 15 is pressed or not.

Program :

```

IN    AX,80H      ; Read status of all switches
AND AX,8000H      ; Mask bit positions for other switches
JZ    NEXT        ; if Z = 0, i.e. switch 15 is pressed then
                  ; program control is transferred to label NEXT.

```

3.1.6 Interfacing 8-bit Output Device

- The microprocessor 8086 sends 8-bit or 16-bit data to the output port according to instruction used for data transfer.

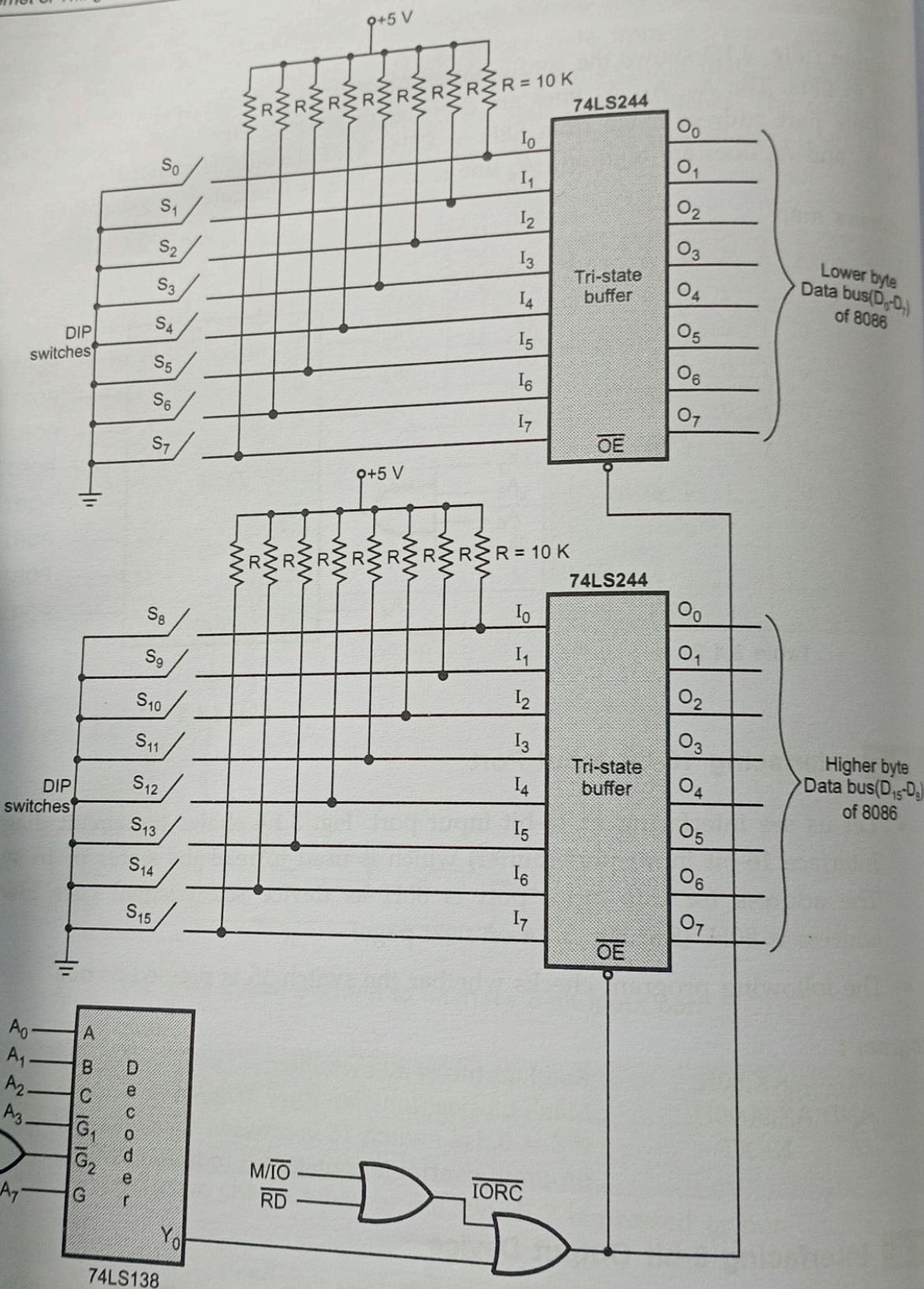


Fig. 3.1.4 Circuit diagram to interface 16-bit input port

- Fig. 3.1.5 shows the circuit diagram to interface 8-bit output port (latch) which is used to send the signal for glowing the LEDs. LED will glow when output pin status is low.
- The IC 74LS138 and OR gate are used to generate device select signal. The latch enable signal is active high. So NOR gate is used to generate latch enable signal, which goes high when Y_2 and \overline{IOWC} both are low.

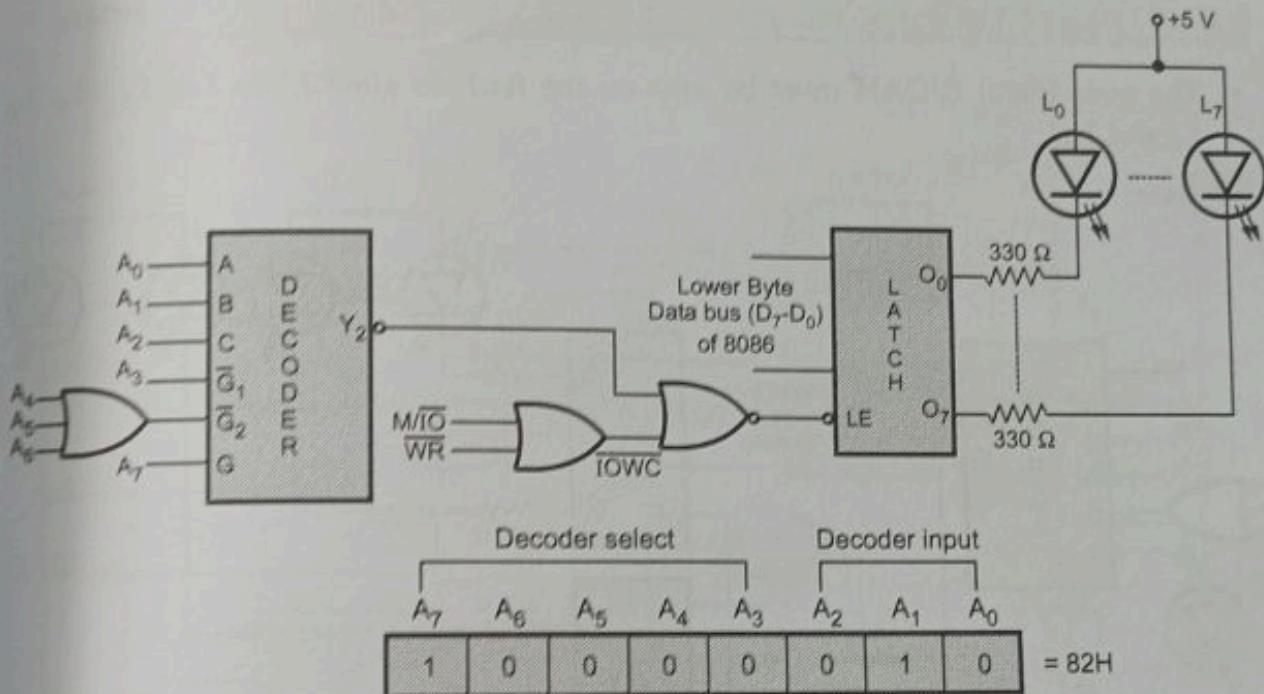


Fig. 3.1.5 Circuit diagram to interface 8-bit output port

- The following program glows the LEDs L_0 , L_2 and L_5 .

L_7	L_6	L_5	L_4	L_3	L_2	L_1	L_0	
1	1	0	1	1	0	1	0	$= DAH$

- The code (data) DAH must be sent on the latch to glow LEDs L_0 , L_2 and L_5 .

Program :

<pre>MOV AL,0DAH</pre>	$; \text{Loads the data in the accumulator.}$
<pre>OUT 82H,AL</pre>	$; \text{Sends the data on the latch.}$

3.1.7 Interfacing 16-bit Output Device

- Fig 3.1.6 shows the circuit diagram to interface 16-bit output port (latch) which is used to send the signal for glowing 16 LEDs. LED will glow when output pin status is low.

- The IC 74LS138 and OR gate are used to generate device select signal. The latch enable signal is active high. So NOR gate is used to generate latch enable signal, which goes high when Y_2 and \overline{IOWC} both are low.
- The following program glows the LEDs $L_{14}, L_{11}, L_8, L_5, L_2$ and L_0 .

L_{15}	L_{14}	L_{13}	L_{12}	L_{11}	L_{10}	L_9	L_8	L_7	L_6	L_5	L_4	L_3	L_2	L_1	L_0	= B6DAH
1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	

- The code (data) B6DAH must be sent on the latch to glow LEDs $L_{14}, L_{11}, L_8, L_5, L_2$ and L_0 .

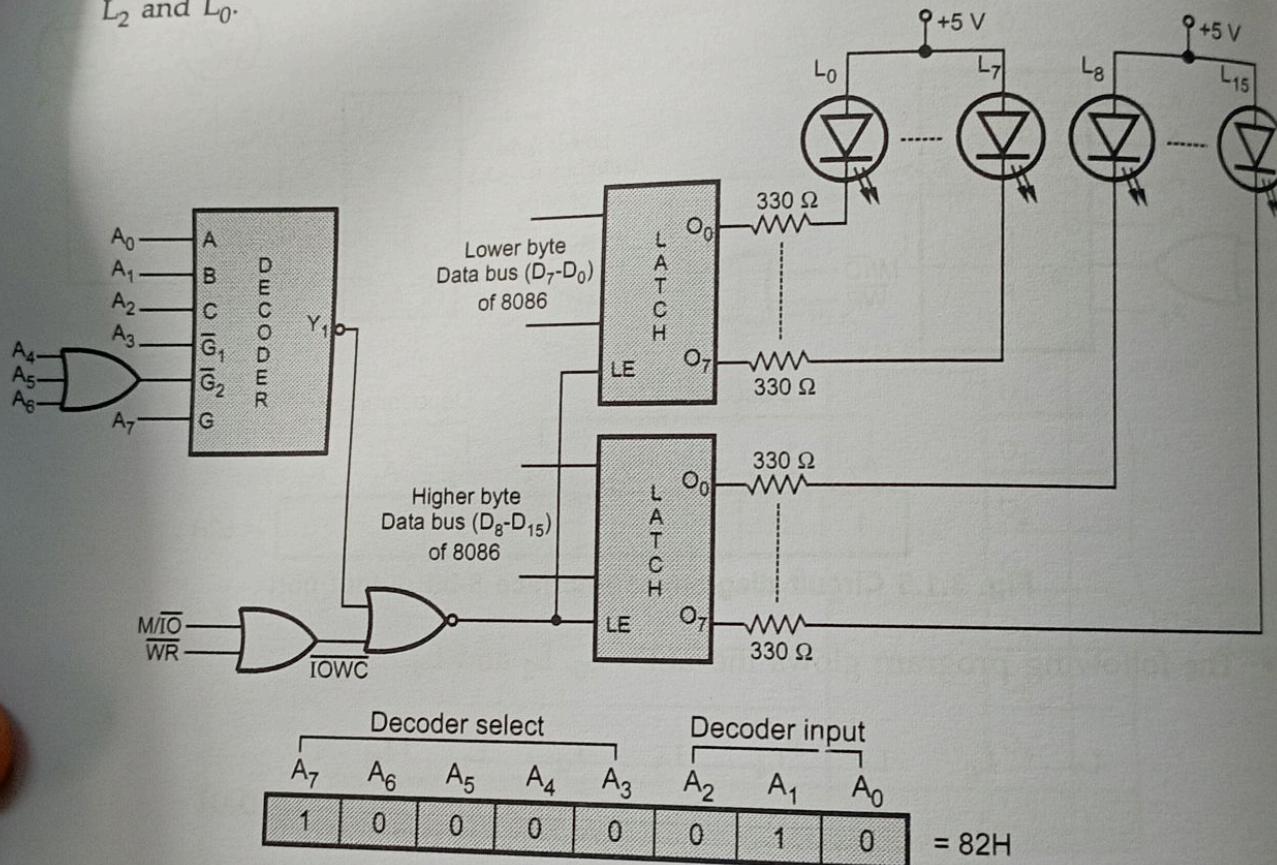


Fig. 3.1.6 Circuit diagram to interface 16-bit output port

Program :

```
MOV AX,0B6DAH ; Loads the data in the accumulator.
OUT 82H,AX ; Sends the data on the latch.
```

Note In the previous examples, we have 8-bit port addresses so only lower eight address lines are used for address decoding. For 16-bit port addresses we have to decode lower sixteen address lines to generate device select signal. The remaining part of the circuit is same.

- Fig. 3.1.7 shows the combined circuit for I/O interfacing. For this circuit the address of input port is 80H and address of output port is 82H. The following program displays the status of switches on the LEDs.

Program :

IN 80H	; Read status of all switches.
OUT 82H	; Send status on the output port.

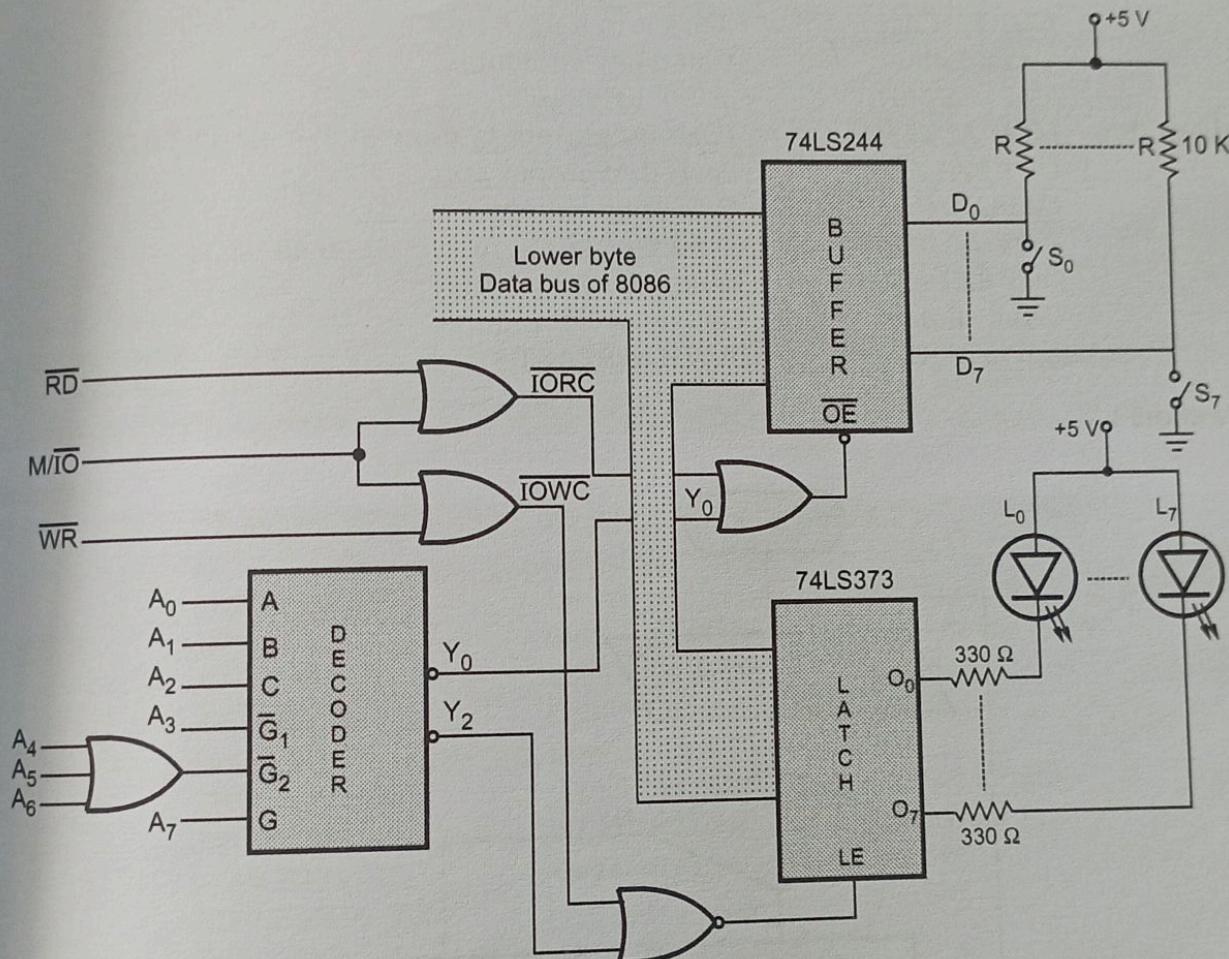


Fig. 3.1.7 I/O interfacing using I/O mapped I/O

Example 3.1.2 Refer Fig. 3.1.7 and write a program that will check the switch1 status and do accordingly.

i) $S_1 = 0$: Blink Lower nibble LEDs. ii) $S_1 = 1$: Blink Higher nibble LEDs.

Assume delay routine is available.

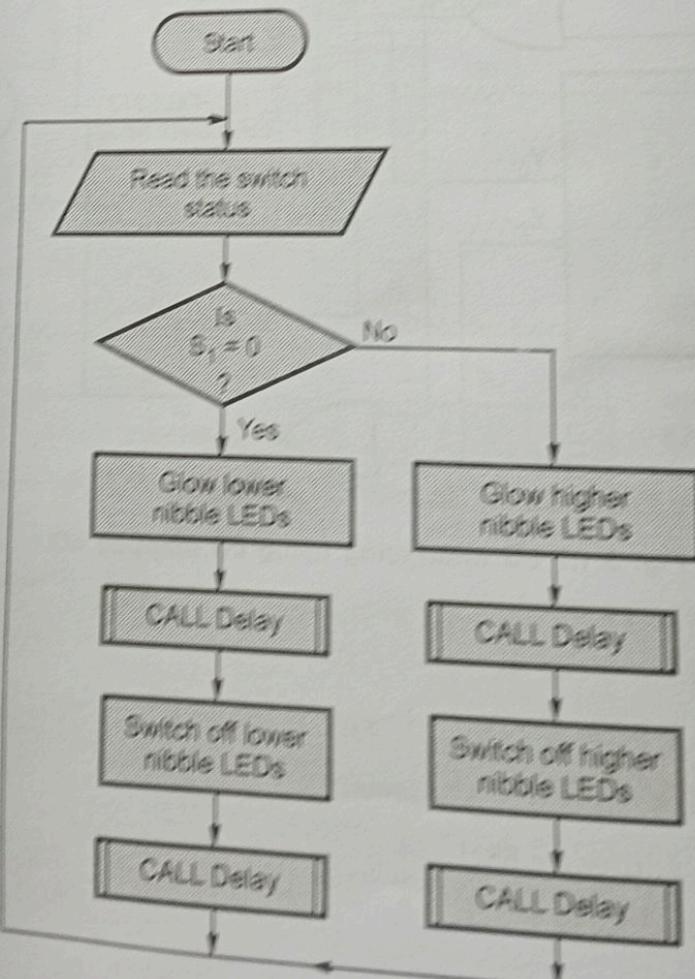
Solution : Input port address = 80H

Output port address = 82H

Source Program :

START :	IN AL,80H AND AL,01H JNZ HIGHER MOV AL,0FOH OUT 82H,AL CALL Delay MOV AL,0FFH OUT 82H,AL CALL Delay JMP START	; Read status of switches ; Masks Bit 1 to Bit 7 ; If sw1 is not pressed goto blink higher nibble ; Load bit pattern to glow lower nibble LEDs ; Send it to output port ; Call delay subroutine ; Load bit pattern to switch off all LEDs ; Send it to output port ; Call delay subroutine ; JUMP to START
HIGHER :	MOV AL,0FH OUT 82H,AL CALL Delay MOV AL,0FFH OUT 82H,AL CALL Delay JMP START	; Load bit pattern to glow higher nibble LEDs ; Send it to output port ; Call delay subroutine ; Load bit pattern to switch off all LEDs ; Send it to output port ; Call delay subroutine ; JUMP to START

Flowchart :



Example 3.1.3 Write an 8086 ALP to read a byte of data from port A and port B. Add the data and save the result in a memory location.

Solution :

```
.model small
.stack 100
.data
LOC      DB      0
.code
START :    MOV AX, @ data      ; [Initialize data segment]
            MOV DS, AX
            IN AL, PORTA      ; Get byte from port A
            MOV BL, AL          ; Save the data byte
            IN AL, PORTB      ; Get byte from port B
            ADD AL, BL          ; Add two bytes
            MOV LOC, AL          ; Save the result
            END START
            END
```

Example 3.1.4 Write an ALP using 8086 instructions to read a byte of data from port A and display its parity status as 00H or FFH for odd and even parity respectively, on port B.

Solution :

```
IN AL, PORTA      ; Read a byte from port A
MOV CX, 08         ; Initialize counter with value 8
MOV DL, 00H        ; Initialize counter to count number of 1s in the byte
BACK :   SHR AL, 1       ; Shift byte right so that LSB in the carry
         JNC SKIP        ; if CY = 0 skip next instruction
         INC DL           ; Otherwise increment count
SKIP :    LOOP BACK      ; Repeat until CX ≠ 0
         SHR DL, 1       ; Check whether count is even or odd
         JNC SKIP1        ; if CY = 0 (count even) goto skip 1
         MOV AL, 00H        ; Otherwise send 00 on port B
         OUT PORTB, AL
         JMP LAST
SKIP1 :   MOV AL, FFH        ; if CY ≠ 0 (count odd) send
         OUT PORTB, AL
LAST :    HLT             ; Stop
```

3.1.8 I/O Interfacing with 16-bit Port Address

- As mentioned earlier, for 16-bit port address we have to decode lower sixteen address lines to generate device select signal.
- Fig. 3.1.8 shows the interfacing of 8-bit input port (buffer) and 8-bit output port (latch) having 16-bit address. The address of input port is 0080H and address of output port is 0082H.

Example :

- Refer Fig. 3.1.8 and write a program to display the status of switches on the LEDs.

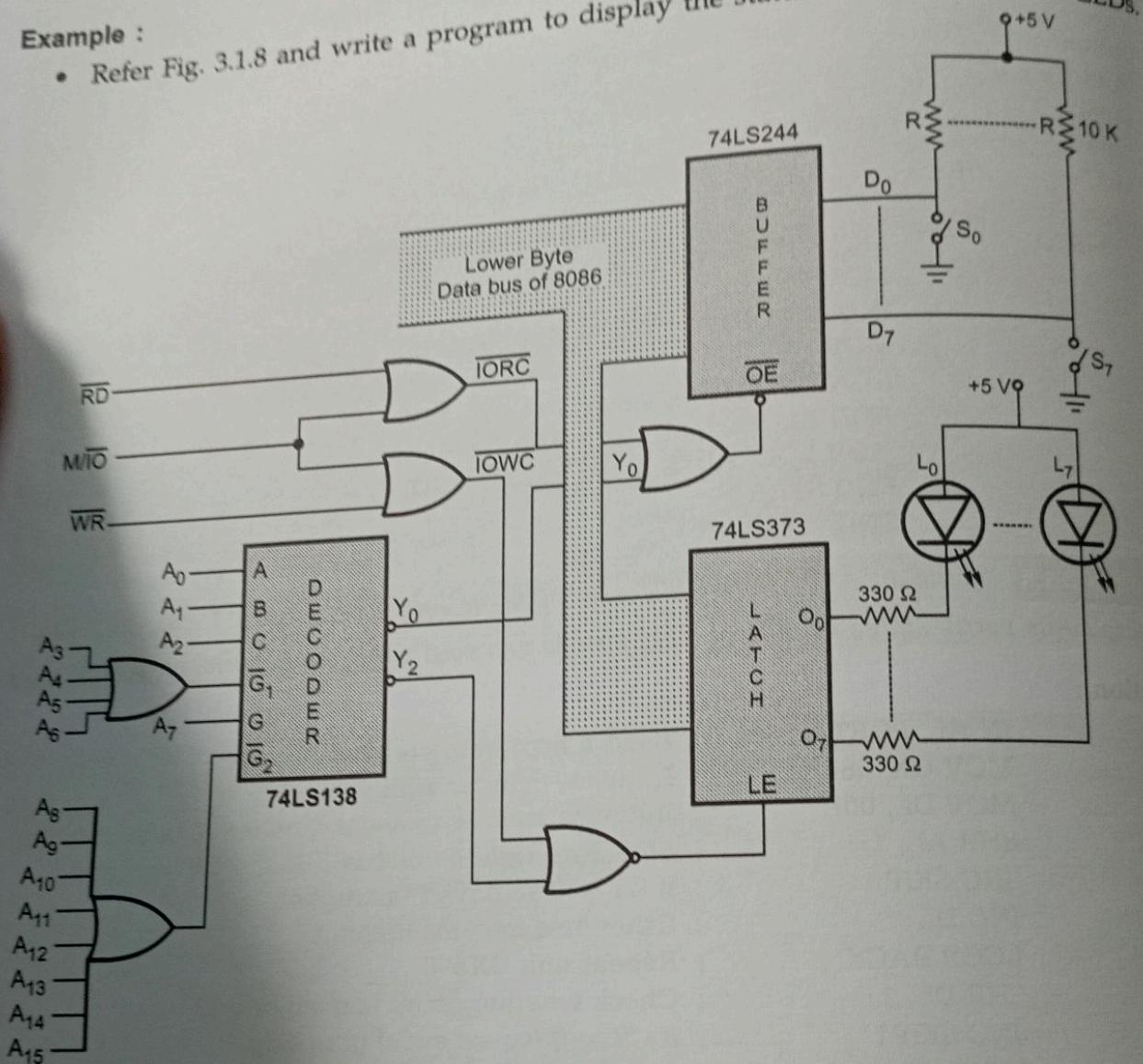


Fig. 3.1.8 Interfacing of 8-bit input port (buffer) and 8-bit output port (latch)

Program :

```

MOV DX,0080H      ; Load 16-bit address of the input port in DX
IN AL,DX          ; Read byte from input port in AL
MOV DX,0082H      ; Load 16-bit address of the output port in DX
OUT DX,AL         ; Send byte to output port from AL

```

3.1.9 I/O Interfacing with Memory Mapped I/O

- As mentioned earlier, in memory mapped 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.
- Fig. 3.1.9 shows the I/O interfacing with memory mapped I/O. As shown in the Fig. 3.1.9, the 8086 uses same control signals and instructions to access I/O as

those of memory. Here \overline{RD} and \overline{WR} signals are activated when M/\overline{IO} signal is high, indicating memory bus cycle.

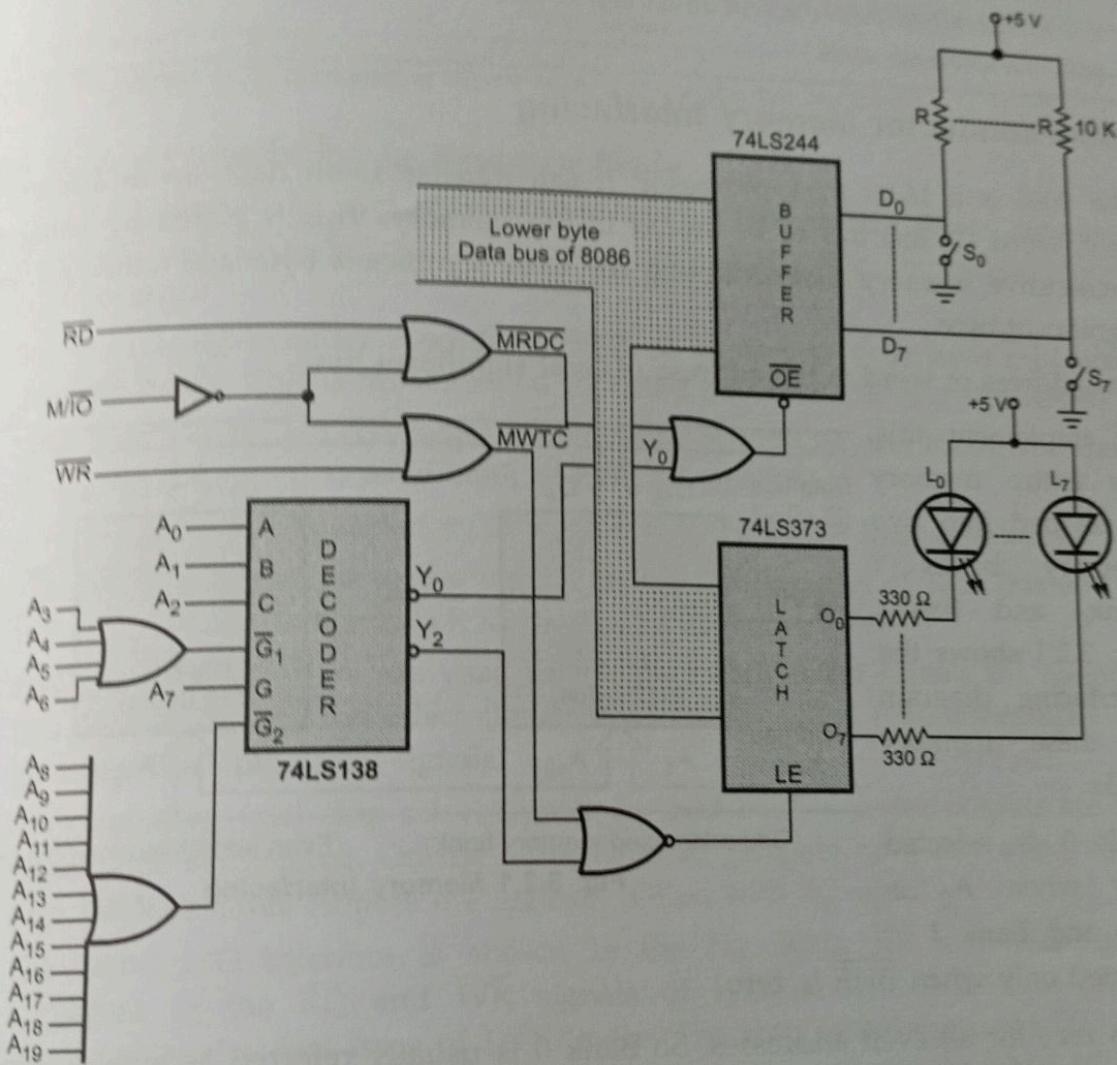


Fig. 3.1.9 I/O Interfacing with memory mapped I/O

Example :

Refer Fig. 3.1.9 and write a program to display the status of switches on the LEDs.

Program :

```
MOV AL,[0080H] ; Read byte from input port in AL
MOV [0082],AL ; Send byte to output port from AL
```

Review Questions

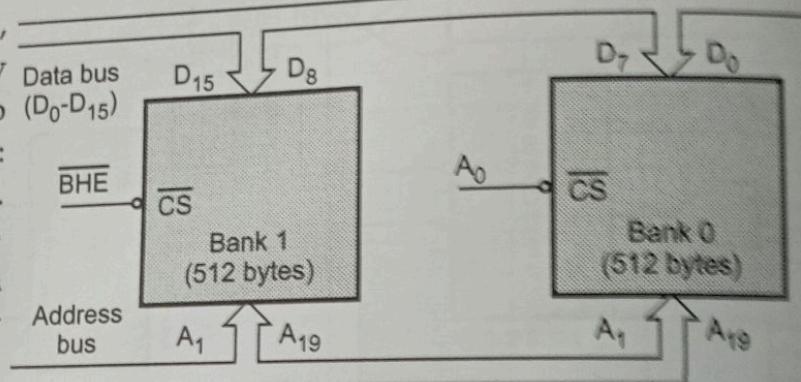
1. Explain the I/O data transfer using I/O ports with the help of neat diagram.
2. Explain the I/O interfacing techniques : a) I/O mapped I/O b) Memory mapped I/O.

3. Draw and explain the interfacing of
 a) An 8-bit input port b) An 16-bit input port
 c) An 8-bit output port d) An 16-bit output port
 with microprocessor 8086.

3.2 Semiconductor Memory Interfacing

- The 8086 is a 16-bit microprocessor, it can transfer 16-bit data. So in addition to byte, word (16-bit) has to be stored in the memory. This is stored by using two consecutive memory locations, one for least significant byte and other for most significant byte.
- The address of word is the address of least significant byte.

- To implement this, the entire memory is divided into two memory banks : Bank₀ and Bank₁. Fig. 3.2.1 shows the interfacing diagram to these memory banks.



(Odd addressed memory bank) (Even addressed memory bank)

Fig. 3.2.1 Memory interfacing

- Bank 0 is selected only when A₀ is zero and Bank 1 is selected only when BHE is zero.
- A₀ is zero for all even addresses. So Bank 0 is usually referred as even addressed memory bank.
- BHE is used to access higher order memory bank, referred to as odd addressed memory bank.
- Together BHE and A₀ tell the interface how the data appears on bus. Four possible combinations are shown in the table.

Sr.No.	Operation	BHE	A ₀	Data lines used
1.	Read/Write a byte at an even address	1	0	D ₇ - D ₀
2.	Read/Write a byte at an odd address	0	1	D ₁₅ - D ₈
3.	Read/Write a word at an even address	0	0	D ₁₅ - D ₀

4.	Read/Write a word at an odd address	0	1	$D_{15}-D_8$ in first operation byte from odd bank is transferred.
		1	0	D_7-D_0 in second operation byte from even bank is transferred.

Note : To access odd addressed word two bus cycles are required.

3.2.1 Memory Interfacing in Minimum Mode

- Fig. 3.2.2 shows the typical minimum mode 8086 system. Here, interfacing of memory and I/O devices are shown with the basic minimum mode 8086 configuration.
- For interfacing memory module to 8086, it is necessary to have odd and even memory banks. This is implemented by using two EPROMs and two RAMs.
- Data lines $D_{15}-D_8$ are connected to odd bank of EPROM and RAM, and data lines D_7-D_0 are connected to even bank of EPROM and RAM.
- Address lines are connected to EPROM and RAM as per their capacities.
- \overline{RD} signal is connected to the output enable (\overline{OE}) signals of EPROMs and RAMs. \overline{WR} signal is connected to \overline{WR} signal of RAMs.
- Two separate decoders are used to generate chip select signals for memory and I/O devices. These chip select signals are logically ORed with either \overline{BHE} or A_0 to generate final chip select signals.
- For generating final chip select signals for odd bank decoder outputs are logically ORed with \overline{BHE} signal. On the other hand to generate final chip select signals for even bank decoder outputs are logically ORed with A_0 signal.
- The 16-bit I/O interface is shown in the Fig. 3.2.2. \overline{RD} and \overline{WR} signals are connected to the \overline{RD} and \overline{WR} signals of I/O device. Data lines $D_{15}-D_0$ are connected to the data lines of I/O device. The chip select signal for I/O device is generated using separate decoder whose output is enabled only when M/\overline{IO} signal is low. (See Fig. 3.2.2 on next page.)

Example 3.2.1 Design an 8086 based system with the following specifications.

- i) 8086 in minimum mode. ii) 4 K ROM iii) 128 RAM

Draw the complete schematic of the design indicating address map.

Solution : The 8086 is a 16-bit microprocessor. It can access 16-bit data simultaneously. For interfacing memory module to 8086 CPU, it is necessary to have odd and even memory banks. This can be achieved by using two 2 kbyte ROMs and two 64 byte RAMs, one for odd bank and another for even bank.

- As 64 kbyte RAM needs 16 address lines, A_1 to A_{16} lines are used and 2 K ROM needs 11 address lines, A_1 to A_{11} . A_0 and \overline{BHE} are used to select even and odd memory banks respectively. Fig. 3.2.3 shows the interface between 8086 and two memory chips. (See Fig. 3.2.3 on page number 3 - 17)

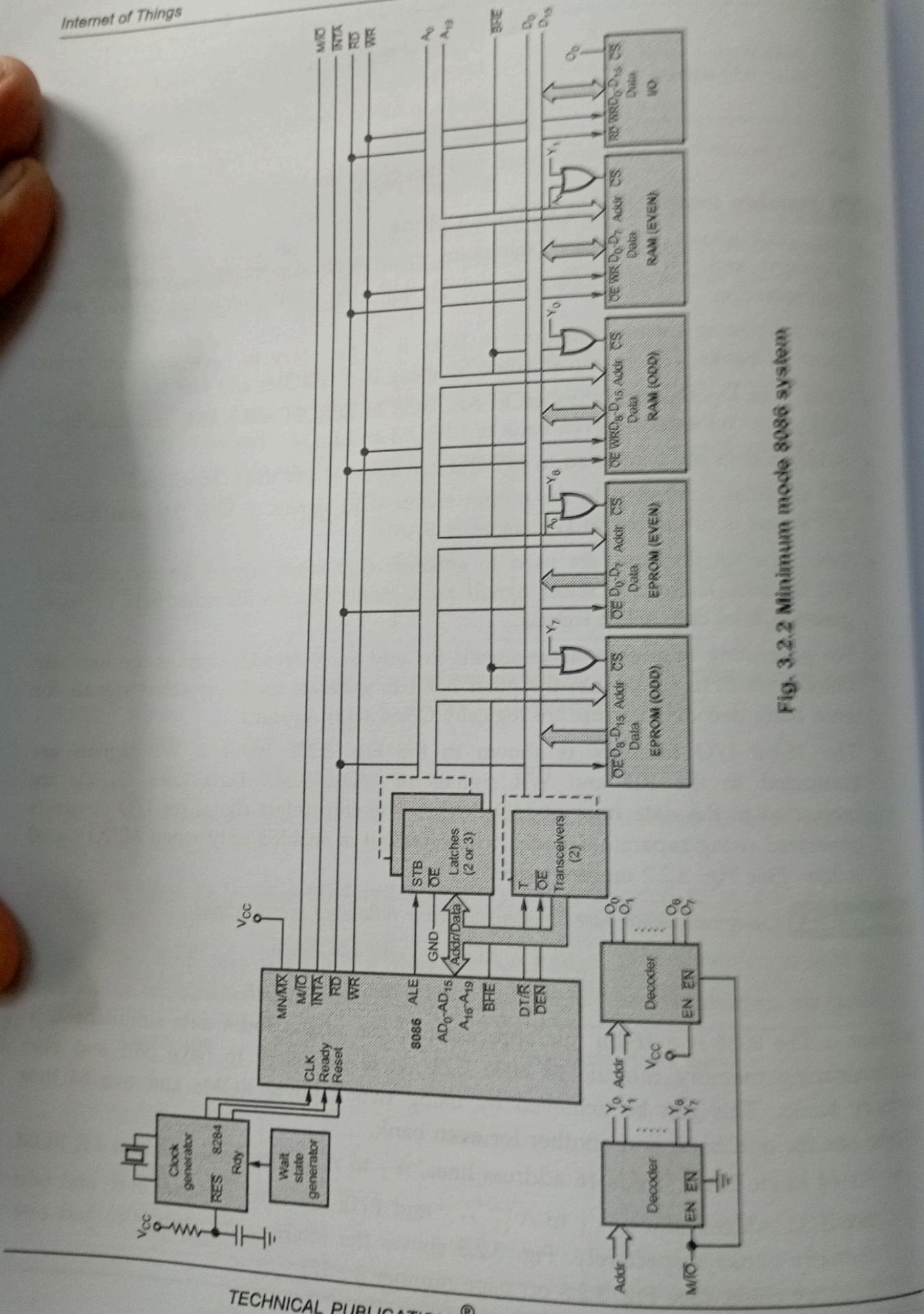


Fig. 3.2.2 Minimum mode 8086 system

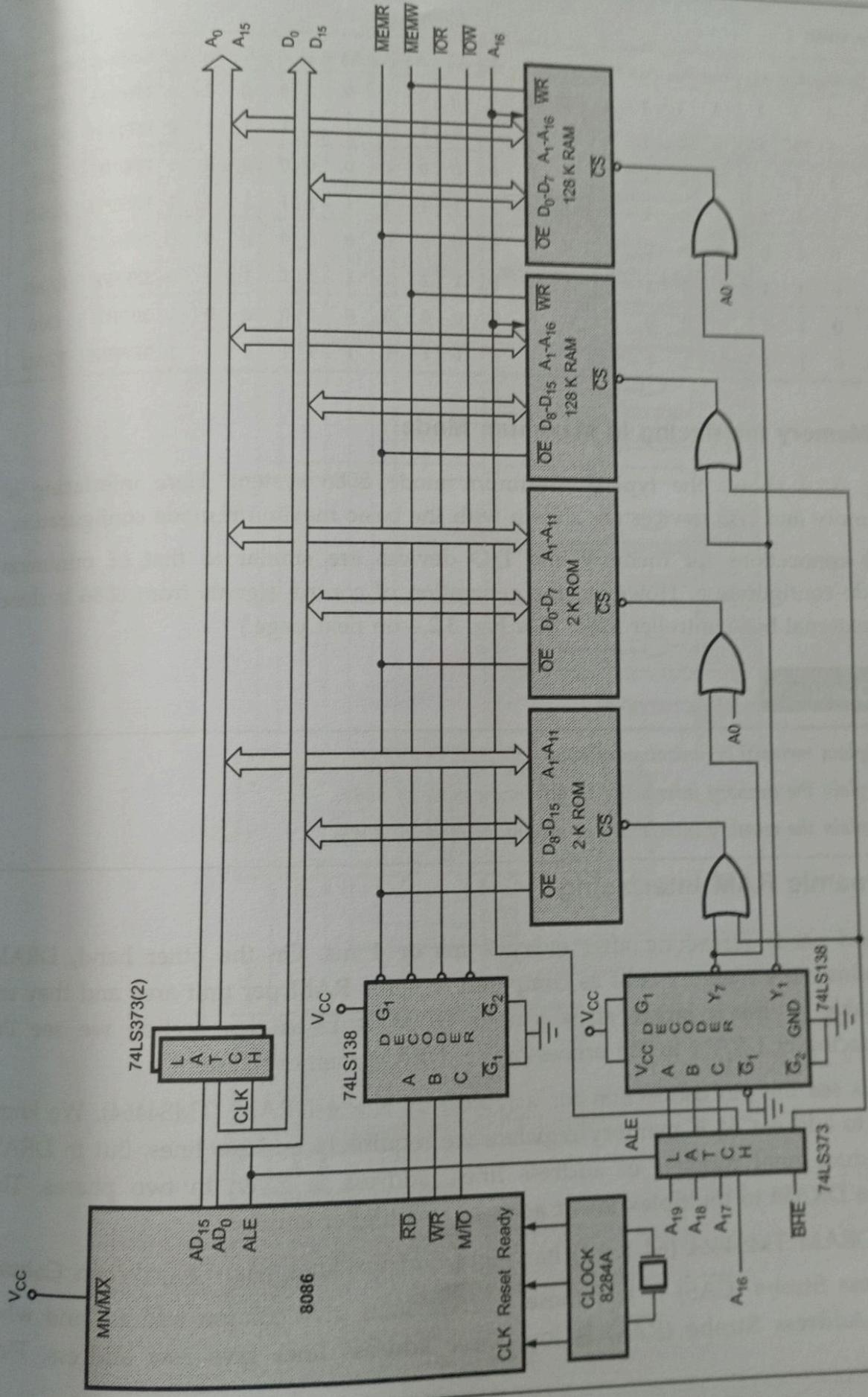


Fig. 3.2.3 Interfacing 128 K RAM and 2K ROM with 8086 in minimum mode

Memory map :

BHE	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	Memory		
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	FF000H	Even	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	FFFFEH	ROM1	
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	FF001H	Odd	
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFFH	ROM2
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20000H	Even	
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	3FFEIH	RAM1	
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	20001H	Odd	
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFFFH	RAM2	

3.2.2 Memory Interfacing in Maximum Mode

- Fig. 3.2.4 shows the typical maximum mode 8086 system. Here interfacing of memory and I/O devices are shown with the basic maximum mode configuration.
- The connections for memory and I/O devices are similar to that of minimum mode configuration. However, the generation of control signals from 8086 is done by external bus controller 8288. (See Fig. 3.2.4 on next page.)

Review Questions

- Explain memory organization of 8086.
- Explain the memory interfacing in minimum mode of 8086.
- Explain the memory interfacing in maximum mode of 8086.

3.3 Dynamic RAM Interfacing

- DRAM needs refreshing after every 2 ms or 4 ms. On the other hand, DRAM contains more memory cells as compared to static RAM per unit area and they are available in much larger sizes : upto $16 \text{ M} \times 1$. In this section we see the interfacing of DRAM to the processor, i.e. DRAM memory system.
- Let us see the pin connection for a typical $64 \text{ K} \times 4$ DRAM (TMS4464). We know that, to address 64 K memory registers we require 16 address lines. But in DRAM to reduce total number of address lines, address is given in two phases. This allows DRAM to multiplex lower address and higher address.
- The DRAM TMS4464 ($64 \text{ K} \times 4$) has eight address lines ($A_0 - A_7$). When Column Address Strobe ($\overline{\text{CAS}}$) is low these address lines give column address and when Row Address Strobe ($\overline{\text{RAS}}$) is low these address lines give row address. First,

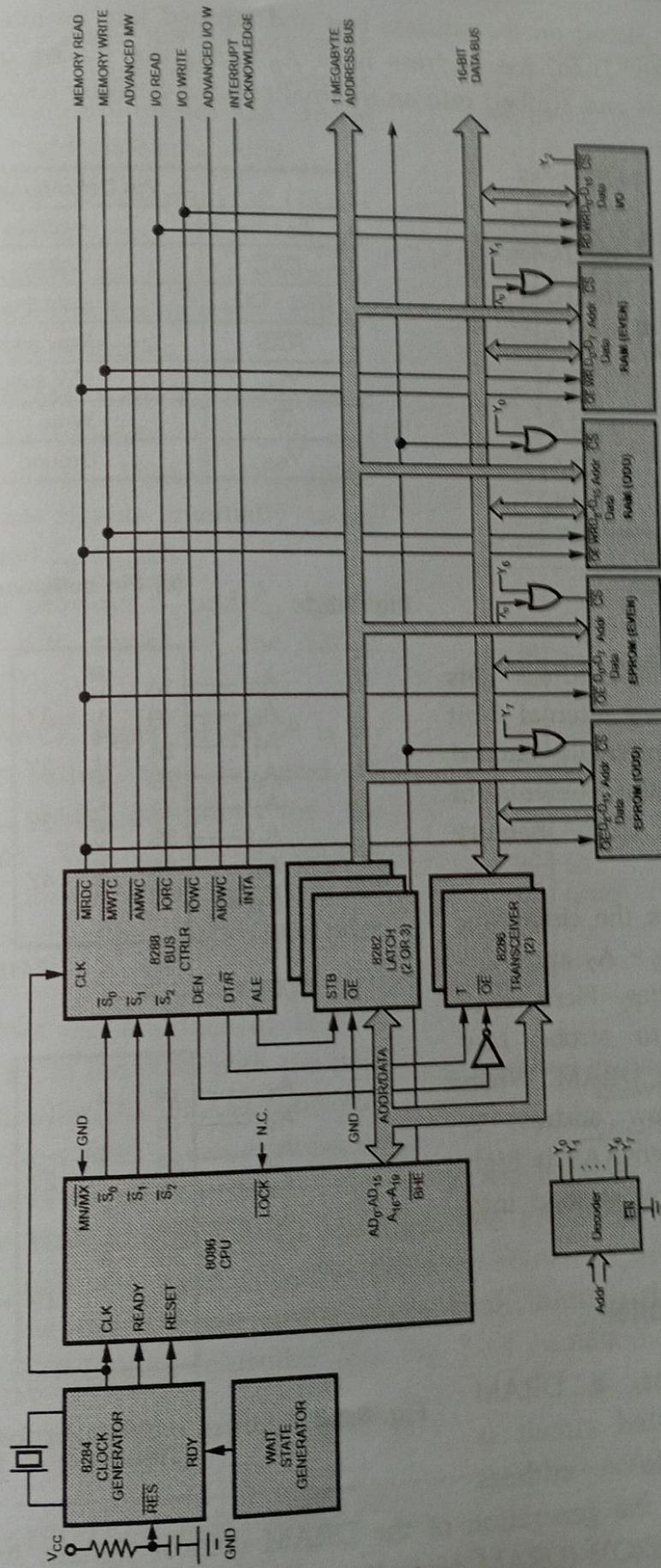
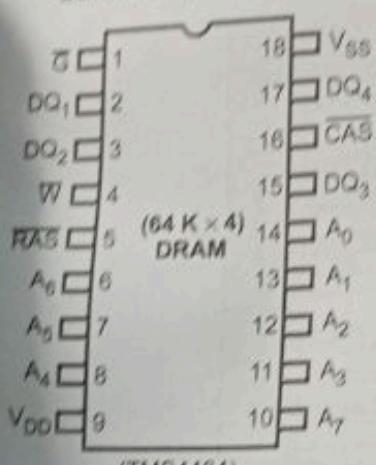


Fig. 3.2.4

$A_0 - A_7$ are placed on the address bus and strobed into an internal row latch by asserting \overline{RAS} . Then, the address lines $A_8 - A_{15}$ are placed on the same eight address inputs and strobed into an internal column latch by asserting \overline{CAS} .



(a) Pin diagram

Pin Definitions	
$A_0 - A_7$	Address lines
\overline{CAS}	Column address strobe
DQ ₁ - DQ ₄	Data lines
\overline{RAS}	Row address strobe
V _{DD}	+5V supply
W	Write enable
V _{SS}	Ground

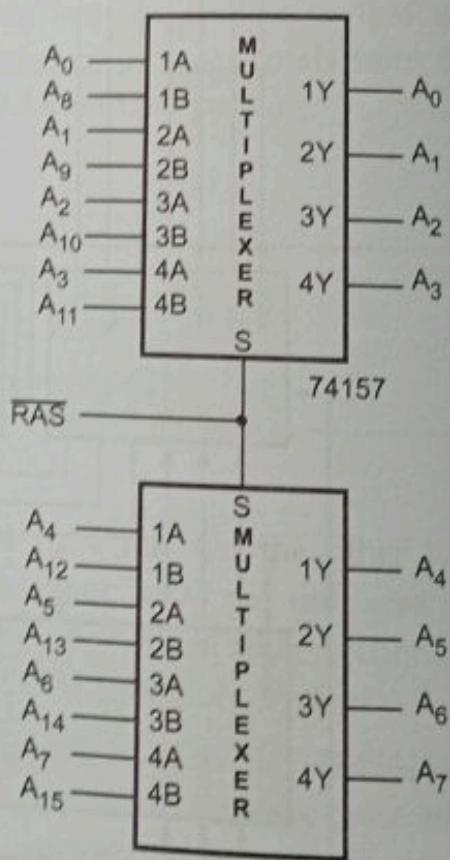
(b) Pin definitions

Fig. 3.3.1

- Therefore, 16-bit address gets latched into two internal 8-bit latches. The internally latched 16-bit access the contents of one of the 4-bit memory locations.
- Fig. 3.3.2 shows the circuit for multiplexing $A_0 - A_7$ and $A_8 - A_{15}$ address lines. Here, \overline{RAS} line is used to strobe row address into the DRAM. When \overline{RAS} is low row address is strobed and when \overline{RAS} is high column address is strobed into the DRAM.

3.3.1 DRAM Controller

- In most systems, a DRAM controller integrated circuit is used to do address multiplexing and the generation of the DRAM control signals. In this section we see the details of 82C08 (DRAM controller).

Fig. 3.3.2 Address multiplexer for 64×4 DRAM (TMS4464)

- The 82C08 can control upto 1 Mbyte of memory for 8086. It contains an address multiplexer that multiplexes an 18-bit address onto 9 address connections.
- Fig. 3.3.3 shows pin diagram of DRAM controller. AL_0 - AL_8 and AH_0 - AH_8 are the address inputs for DRAM controller and AO_0 - AO_8 are the address outputs from the DRAM controller.
- The 82C08, DRAM controller has internal circuitry which generates the \overline{CAS} and \overline{RAS} signals for the DRAM. The 82C08 generates these signals internally using the CLK, \bar{S}_1 and \bar{S}_0 signals.
- The processor provides \bar{S}_1 and \bar{S}_0 signals as \overline{RD} and \overline{WR} inputs to the 82C08 DRAM controller.
- The $\overline{AACK}/\overline{XACK}$ signal of 80C08 is an acknowledge output that is used to indicate the ready condition for the microprocessor. This signal is usually connected to the READY input of the microprocessor or the clock generator READY input.
- Fig. 3.3.4 shows the interface of 1Mbyte DRAM memory to the 8086 with the help of DRAM controller. As DRAM controller accepts 18 address lines it can address up to 256 K memory locations. Here, 1 Mbyte DRAM memory is interfaced using a series of four 256 K \times 8bit DRAM modules. The PAL 16L8 is used to generate bank select signals and other control signals, as shown in the Fig. 3.3.4. The A_{19} signal selects the upper bank or the lower bank through the BS (Bank Select) input to the 82C08 DRAM controller. (See Fig. 3.3.4 on next page)

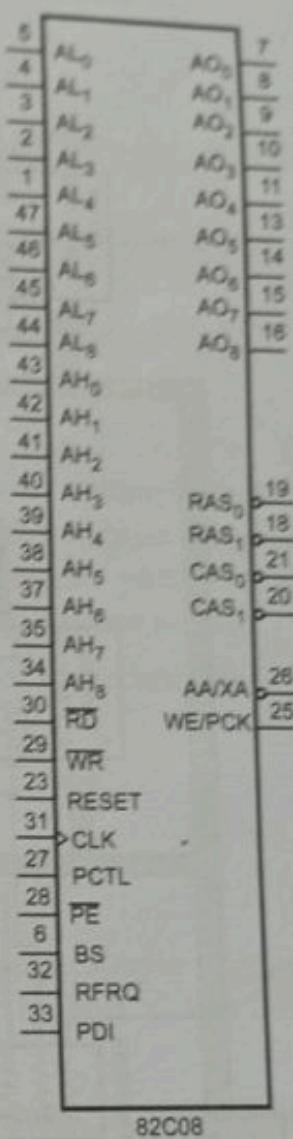


Fig. 3.3.3 Pin diagram of 82C08 DRAM controller

Review Questions

- Write a note of DRAM interfacing.
- What is the role of DRAM controller ?

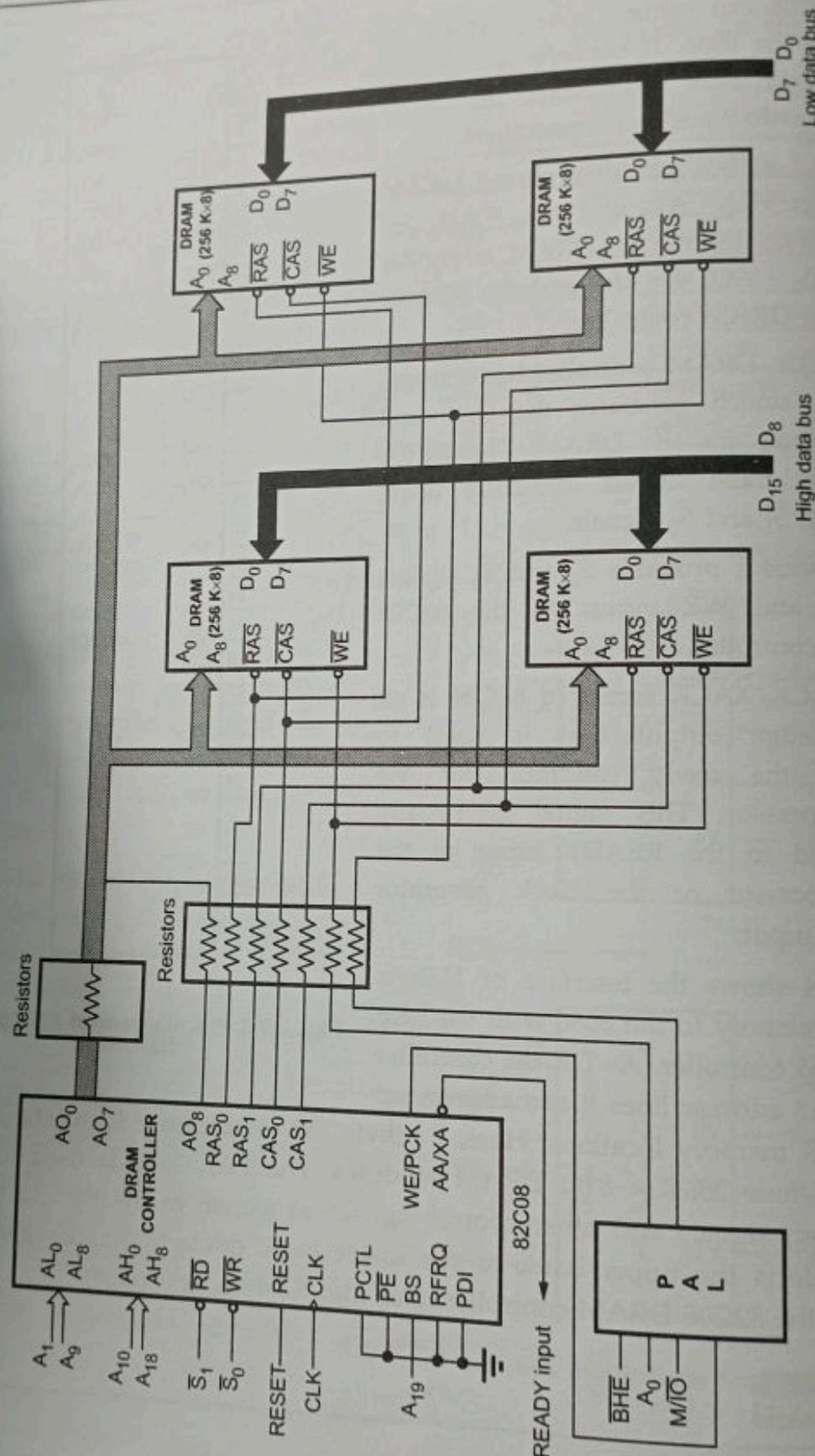


Fig. 3.3.4 The interface of 1 Mbyte DRAM memory to 8086

3.4 Programmable Peripheral Interface (PPI) - 8255

3.4.1 Features

1. The 8255A is a widely used, programmable, parallel I/O device or parallel communication interface.
2. It can be programmed to transfer data under various conditions, from simple I/O to interrupt I/O.
3. It is compatible with all Intel and most other microprocessors.
4. It is completely TTL compatible.
5. It has three 8-bit ports (24 I/O pins) : Port A, Port B and Port C, which are arranged in two groups of 12 pins. Each port has an unique address and data can be read from or written to a port. In addition to the address assigned to the three ports, another address is assigned to the control register into which control words are written for programming the 8255 to operate in various modes.
6. Its bit set/reset mode allows setting and resetting of individual bits of Port C.
7. The 8255 can operate in 3 I/O modes :

Mode 0 : Simple input/output

Mode 1 : Input/Output with handshake

Mode 2 : Bi-directional I/O data transfer

- a) In **Mode 0**, Port A and Port B can be configured as simple 8-bit input or output ports without handshaking. The two halves of Port C can be programmed separately as 4-bit input or output ports.
 - b) In **Mode 1**, two groups each of 12 pins are formed. Group A consists of Port A and the upper half of Port C while Group B consists of Port B and the lower half of Port C. Ports A and B can be programmed as 8-bit input or output ports with three lines of Port C in each group used for handshaking.
 - c) In **Mode 2**, only Port A can be used as a bidirectional port. The handshaking signals are provided on five lines of Port C ($PC_3 - PC_1$). Port B can be used in mode 0 or in mode 1.
8. All I/O pins of 8255 has 2.5 mA d.c. driving capacity (i.e. sourcing current of 2.5 mA).

3.4.2 Pin Diagram

- Fig. 3.4.1 shows the pin diagram of 8255.

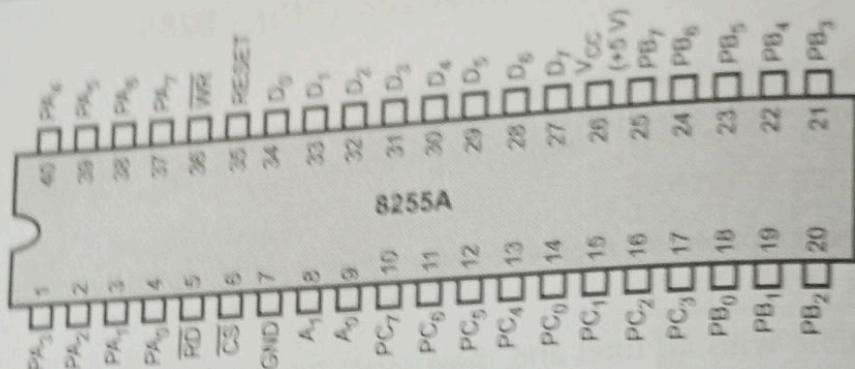


Fig. 3.4.1 Pin diagram of 8255A

Pin symbols	Function
D ₀ -D ₇ (Data bus)	These bi-directional, tri-state data bus lines are connected to the system data bus. They are used to transfer data and control word from microprocessor (8085) to 8255 or to receive data or status word from 8255 to the 8085.
PA ₀ -PA ₇ (Port A)	These 8-bit bi-directional I/O pins are used to send data to output device and to receive data from input device. It functions as an 8-bit data output latch/buffer, when used in output mode and an 8-bit data input buffer, when used in input mode.
PB ₀ -PB ₇ (Port B)	These 8-bit bi-directional I/O pins are used to send data to output device and to receive data from input device. It functions as an 8-bit data output latch/buffer when used in output mode and an 8-bit data input buffer, when used in input mode.
PC ₀ -PC ₇	These 8-bit bi-directional I/O pins are divided into two groups PC _L (PC ₃ -PC ₀) and PC _U (PC ₇ -PC ₄). These groups individually can transfer data in or out when programmed for simple I/O, and used as handshake signals when programmed for handshake or bi-directional modes.
RD (Read)	When this pin is low, the CPU can read the data in the ports or the status word, through the data buffer.
WR (Write)	When this input pin is low, the CPU can write data on the ports or in the control register through the data bus buffer.
CS (Chip Select)	This is an active low input which can be enabled for data transfer operation between the CPU and the 8255.
RESET	This is an active high input used to reset 8255. When RESET input is high, the control register is cleared and all the ports are set to the input mode. Usually RESET OUT signal from 8085 is used to reset 8255.
A ₀ and A ₁	These input signals along with RD and WR inputs control the selection of the control/status word registers or one of the three ports.

Address		Port
A ₁	A ₀	
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Control register

Table 3.4.1 Port selection

3.4.3 Block Diagram

- Fig. 3.4.2 shows the internal block diagram of 8255A. It consists of data bus buffer, control logic and Group A and Group B controls.

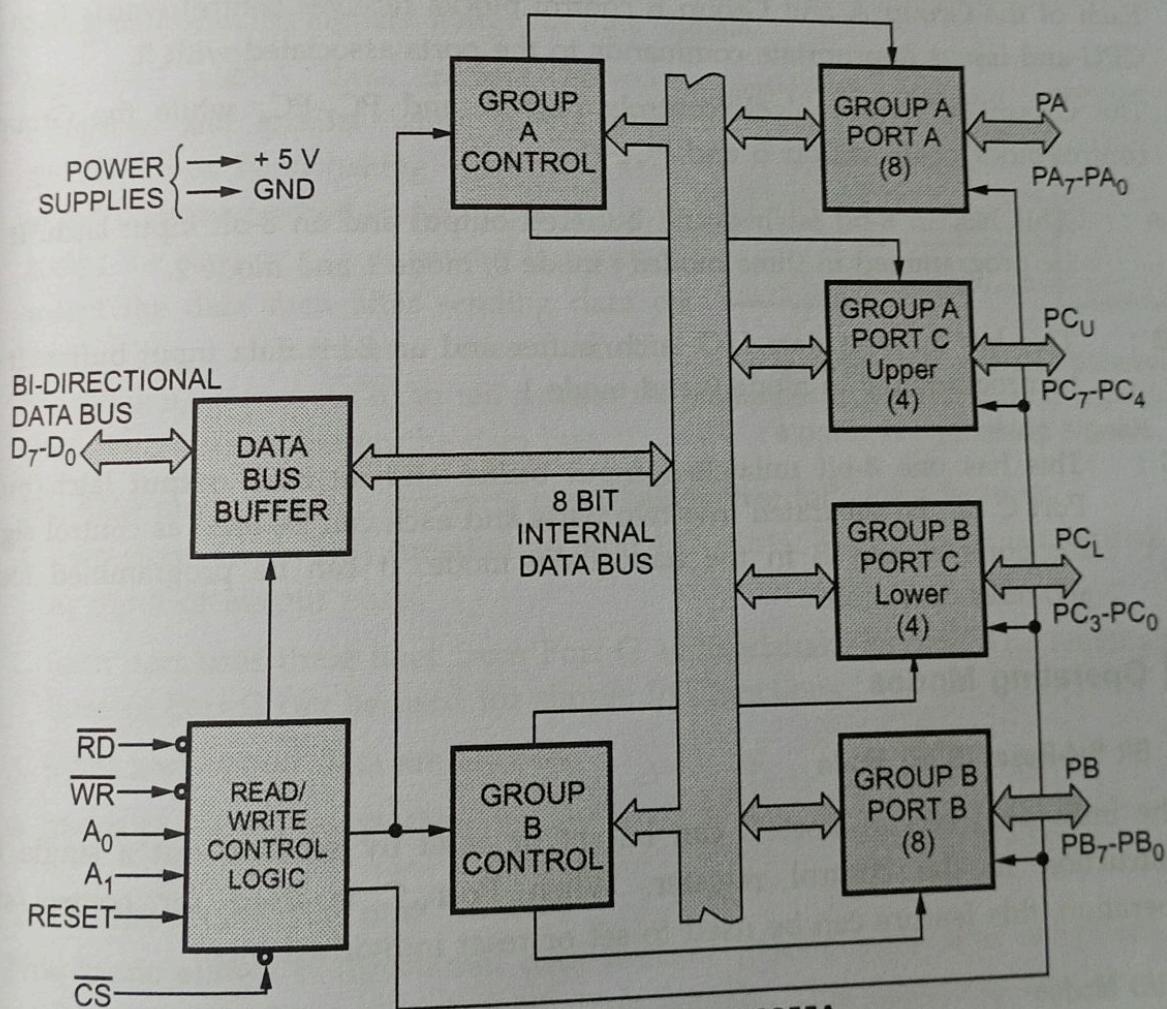


Fig. 3.4.2 Block diagram of 8255A

3.4.3.1 Data Bus Buffer

- This tri-state bi-directional buffer is used to interface the internal data bus of 8255 to the system data bus.

- Input or Output instructions executed by the CPU either Read data from, or Write data into the buffer.
- Output data from the CPU to the ports or control register, and input data to the CPU from the ports or status register are all passed through the buffer.

3.4.3.2 Control Logic

- The control logic block accepts control bus signals as well as inputs from the address bus and issues commands to the individual group control blocks (Group A control and Group B control). It issues appropriate enabling signals to access the required data/control words or status word.
- The input pins for the control logic section are described here.

3.4.3.3 Group A and Group B Controls

- Each of the Group A and Group B control blocks receives control words from the CPU and issues appropriate commands to the ports associated with it.
- The Group A control block controls Port A and PC₇-PC₄ while the Group B control block controls Port B and PC₃-PC₀.

Port A : This has an 8-bit latched and buffered output and an 8-bit input latch. It can be programmed in three modes : mode 0, mode 1 and mode 2.

Port B : This has an 8-bit data I/O latch/buffer and an 8-bit data input buffer. It can be programmed in mode 0 and mode 1.

Port C : This has one 8-bit unlatched input buffer and an 8-bit output latch/buffer. Port C can be separated into two parts and each can be used as control signals for ports A and B in the handshake mode. It can be programmed for bit set/reset operation.

3.4.4 Operating Modes

3.4.4.1 Bit Set-Reset (BSR) Mode

- The individual bits of Port C can be set or reset by sending out a single OUT instruction to the control register. When Port C is used for control/status operation, this feature can be used to set or reset individual bits.

3.4.4.2 I/O Modes

Mode 0 : Simple input/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 buffered, not latched.
 3. Ports do not have handshake or interrupt capability.

Mode 1 : Input/Output with handshake

- In this mode, input or output data transfer is controlled by handshaking signals.
- Handshaking signals are used to transfer data between devices whose data transfer speeds are not same.
- For example, computer can send data to the printer with large speed but printer can't accept data and print data with this rate. So computer has to send data with the speed with which printer can accept. This type of data transfer is achieved by using handshaking signals alongwith data signals.
- Fig. 3.4.3 shows data transfer between computer and printer using handshaking signals. These handshaking signals are used to tell computer whether printer is ready to accept the data or not. If printer is ready to accept the data then after sending data on data bus, computer uses another handshaking signal (STB) to tell printer that valid data is available on the data bus.

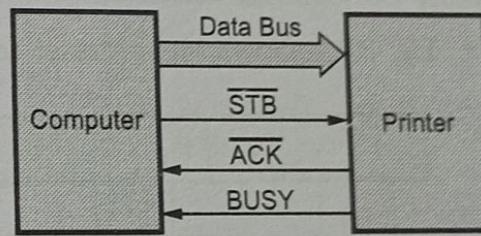


Fig. 3.4.3 Data transfer between computer and printer using handshaking signals

- The 8255 mode 1 which supports handshaking has following features.
 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.

Mode 2 : Bi-directional I/O data transfer

- This mode allows bi-directional data transfer (transmission and reception) over a single 8-bit data bus using handshaking signals. This feature is available only in Group A with Port A as the 8-bit bidirectional data bus ; and PC₃ - PC₇ are used for handshaking purpose.
- In this mode, both inputs and outputs are latched.

- Due to use of a single 8-bit data bus for bi-directional data transfer, the data sent out by the CPU through Port A appears on the bus connecting it to the peripheral, only when the peripheral requests it.
- The remaining lines of Port C i.e. PC₀-PC₂ can be used for simple I/O functions.
- The Port B can be programmed in mode 0 or in mode 1. When Port B is programmed in mode 1, PC₀-PC₂ lines of Port C are used as handshaking signals.

3.4.5 Control Word Formats

- A high on the RESET pin causes all 24 lines of the three 8-bit ports to be in the input mode. All flip-flops are cleared and the interrupts are reset. This condition is maintained even after the RESET goes low.
- The ports of the 8255 can then be programmed for any other mode by writing a single control word into the control register, when required.

3.4.5.1 For Bit Set/Reset Mode

- Fig. 3.4.4 shows bit set/reset control word format.

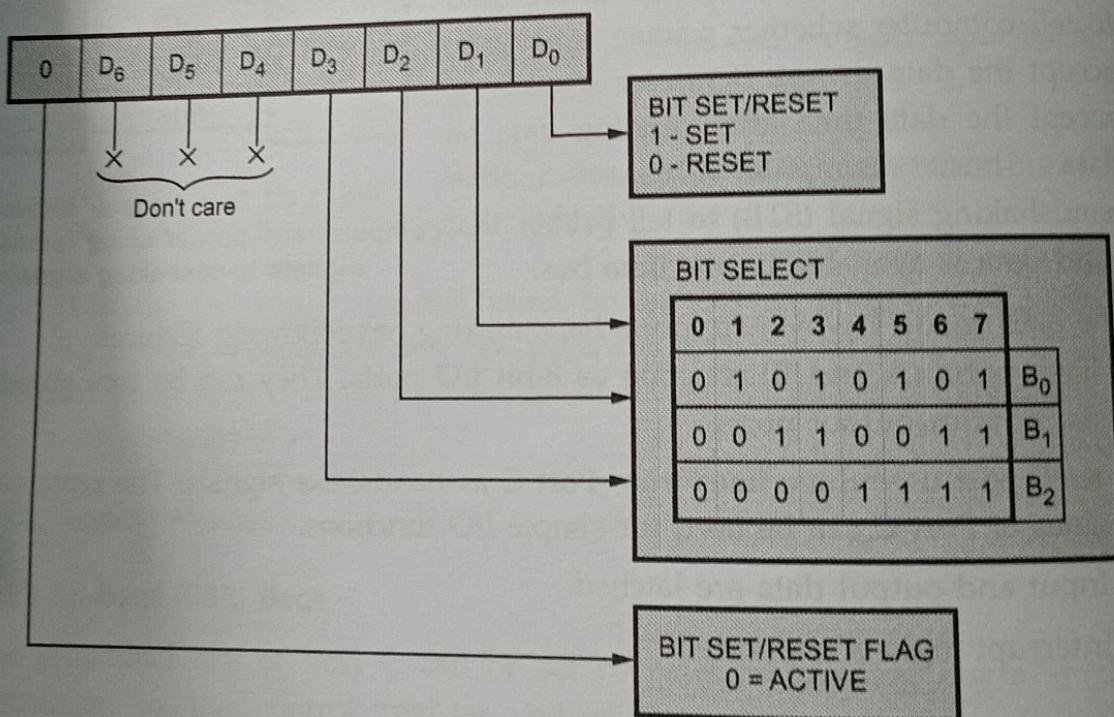


Fig. 3.4.4 Bit set/reset control word format

- The eight possible combinations of the states of bits D₃ - D₁ (B₂ B₁ B₀) in the Bit Set-Reset format (BSR) determine particular bit in PC₀ - PC₇ being set or reset as per the status of bit D₀.
- A BSR word is to be written for each bit that is to be set or reset.

- For example, if bit PC_3 is to be set and bit PC_4 is to be reset, the appropriate BSR words that will have to be loaded into the control register will be, $0 \times \times \times 0111$ and $0 \times \times \times 1000$, respectively, where \times is don't care.
- The BSR word can also be used for enabling or disabling interrupt signals generated by Port C when the 8255 is programmed for Mode 1 or 2 operation. This is done by setting or resetting the associated bits of the interrupts. This is described in detail in next section.

3.4.5.2 For I/O Mode

- The mode definition format for I/O mode is shown in Fig. 3.4.5. The control words for both, mode definition and bit set-reset are loaded into the same control register, with bit D_7 used for specifying whether the word loaded into the control register is a mode definition word or bit set-reset word. If D_7 is high, the word is

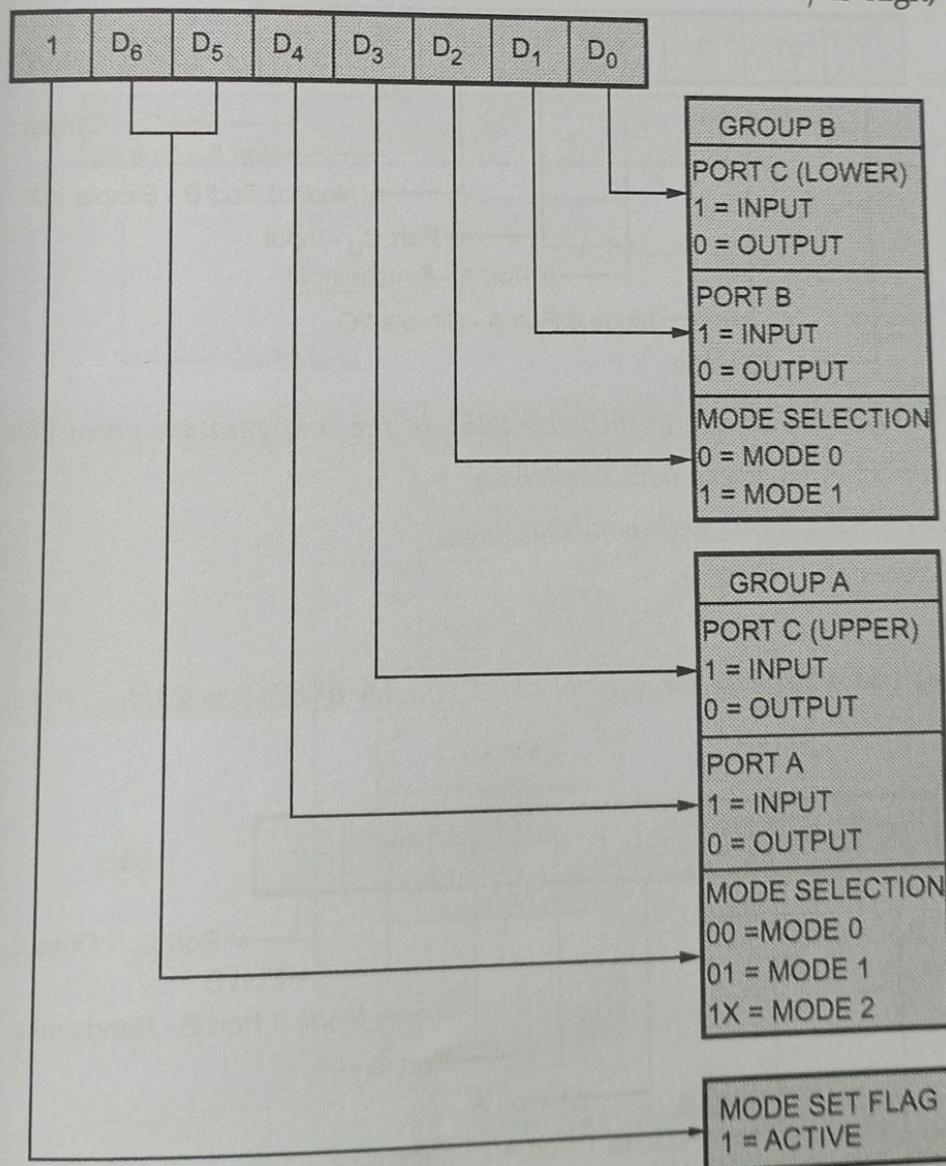


Fig. 3.4.5 8255 Mode definition format

taken as a mode definition word, and if it is low, it is taken as a bit set-reset word. The appropriate bits are set or reset depending on the type of operation desired, and loaded into the control register. (See Fig. 3.4.5 on previous page).

Example 3.4.1 Write a program to initialize 8255 in the configuration given below :

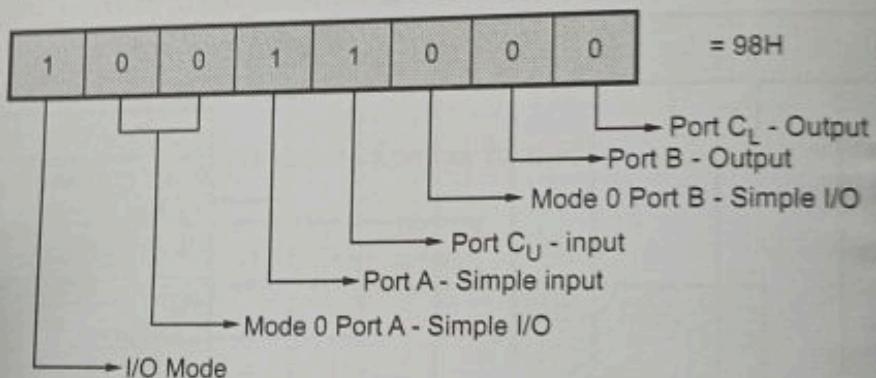
1. Port A : Simple input
2. Port B : Simple output
3. Port C_L : Output
4. Port C_U : Input

Assume address of the control word register of 8255 is 83H.

Solution :

Source program : `MOV AL,98H
OUT 83H,AL`

; Load control word
; Send control word

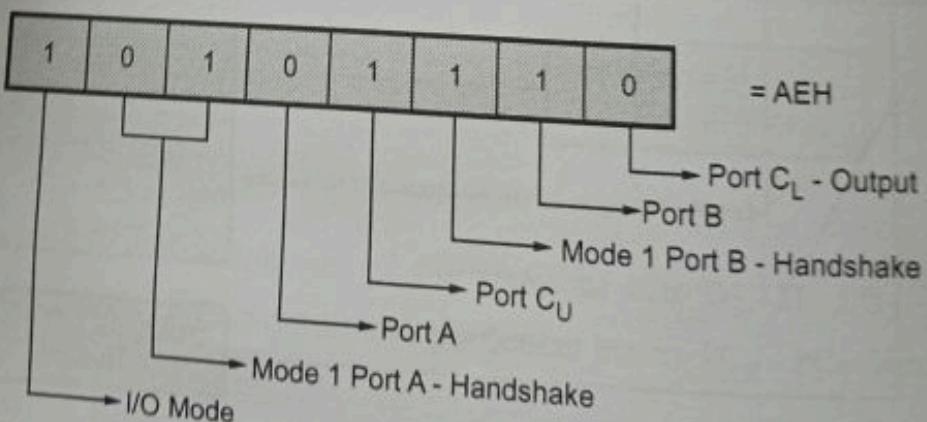


Example 3.4.2 Write a program to initialize 8255 in the configuration given below :

1. Port A : Output with handshake
2. Port B : Input with handshake
3. Port C_L : Output
4. Port C_U : Input

Assume address of the control word register of 8255 is 23H.

Solution :



Source program : MOV AL,0AEH
OUT 23H,AL

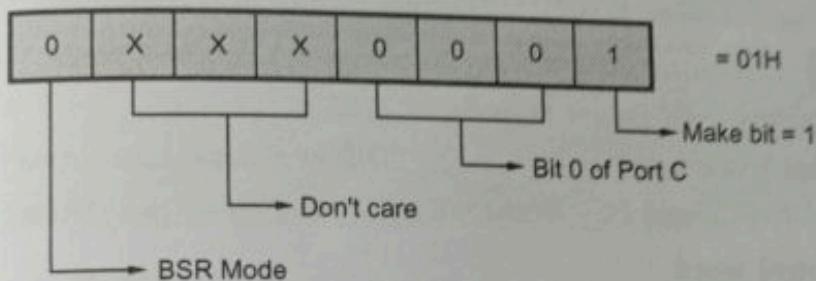
; Load control word
; Send control word

Program : Blink port C bit 0 of 8255.

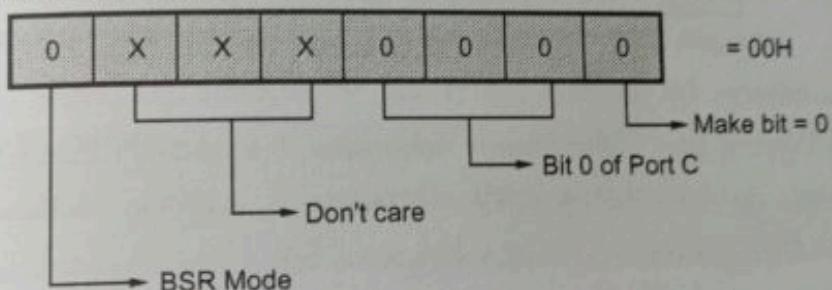
Program statement :

Write a program to blink Port C bit 0 of the 8255. Assume address of control word register of 8255 is 83H. Use bit set/reset mode.

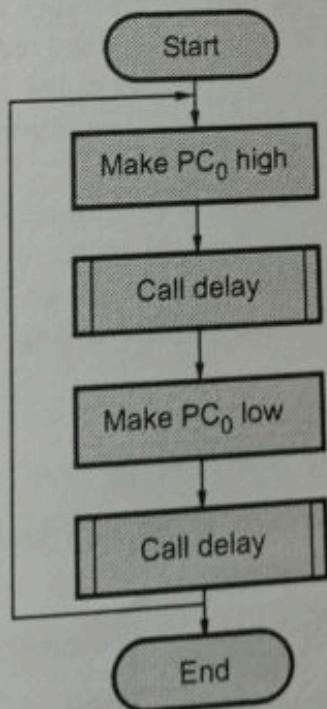
Solution : Control word to make bit 0 high.



Control word to make bit 0 low



Flowchart :



Source program :

BACK:

```

MOV AL,01H ; Load bit pattern to make PC0 high
OUT 83H,AL ; Send it to control word register
CALL DELAY ; Call Delay subroutine
MOV AL,00H ; Load bit pattern to make PC0 Low
OUT 83H,AL ; Send it to control word register
CALL Delay ; Call Delay subroutine
JMP BACK  ; Repeat

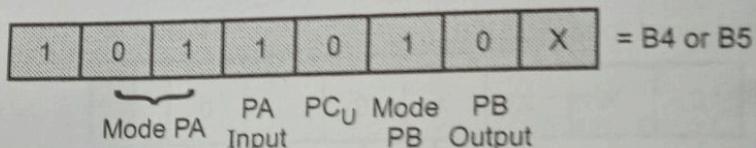
```

Example 3.4.3 Explain the control word format in I/O and BSR mode.

Prepare a control word needed to initialize 8255 as :

- i) Port A handshake input
- ii) Port B handshake output
- iii) Port C bits PC₆ and PC₇ as output.

Solution : Control word



Example 3.4.4 Write a BSR control word subroutine to set bits PC7 and PC3 and reset them after 10 msec. Assume that a delay subroutine is available. Address for control word register = 83 H.

Solution : Program

```

SR PROC NEAR
MOV AL, 07H ; Load bit pattern to make PC3 high
OUT 83H, AL ; Send it to control word register
MOV AL, 0FH ; Load bit pattern to make PC7 high
OUT 83H, AL ; Send it to control word register
CALL DELAY ; WAIT for 10 msec
MOV AL, 06H ; Load bit pattern to make PC3 low
OUT 83H, AL ; Send it to control word register
MOV AL, 0EH ; Load bit pattern to make PC7 low
OUT 83H, AL ; Send it to control word register
RET
ENDP

```

Example 3.4.5 Consider the following addresses for PA = 30H, PB = 31H, PC = 32H, CWR = 33H. Write a set of instructions to set PC0 and PC2 using BSR mode.

Solution :

```

MOV AL, 01H ; Set PC0
OUT 33H, AL
MOV AL, 05H ; Set PC2
OUT 33H, AL

```

Example 3.4.6 Show the mode set control word need to initialize an 8255A as follows :

- Port A - handshake input ;
- Port B - handshake output ;
- Port C - bits PC6 and PC7 as outputs.

Solution :

1	0	1	1	0	1	0	X	= B4H
I/O Mode	Mode 1 Port A	Port A Input	Port C Output	Mode 1 Port B	Port B Output	Port C		

3.4.6 8255 Programming and Operation

3.4.6.1 Programming in Mode 0

- The Ports A, B and C can be configured as simple input or output ports by writing the appropriate control word in the control word register.
- In the control word, D₇ is set to '1' (to define a mode set operation) and D₆, D₅ and D₂ are all set to '0' to configure all the ports in Mode 0 operation.
- The status of bits D₄, D₃, D₁ and D₀ then determine (refer to Fig. 3.4.7) whether the corresponding ports are to be configured as input or output.
- For example in mode 0, if Port A and Port B are to operate as output ports with Port C lower as input and Port C upper as output, the control word that will have to be loaded into the control register will be as follows.

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	= 81H
1	0	0	0	0	0	0	1	

- As mentioned earlier, this mode provides simple input and output operations for each of the three ports. No handshaking is required, data is simply written to or read from a specified port.

Input mode :

- Fig. 3.4.6 shows the timing diagram for mode 0 input mode.
- After initialization of 8255 in the input mode 0, CPU can read data through the input port by initiating read command with proper port address.
- Read command activates RD signal. Upon activation of RD signal CPU reads the data from the selected input port into the CPU register.

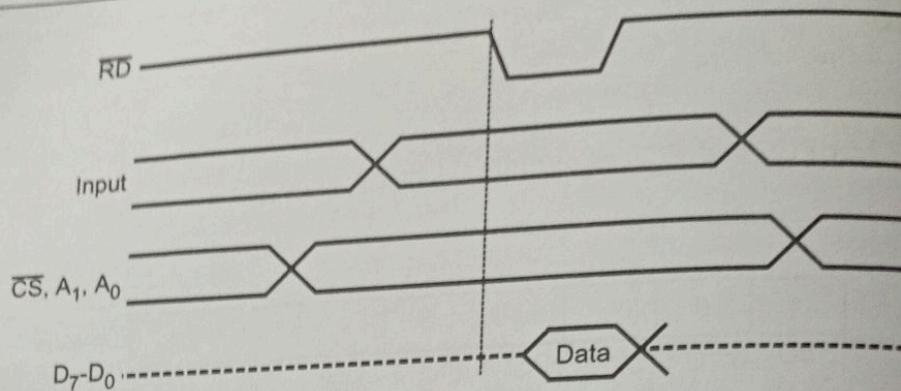


Fig. 3.4.6 Timing diagram for mode 0 input mode

Output mode : Fig. 3.4.7 shows the timing diagram for mode 0 output mode.

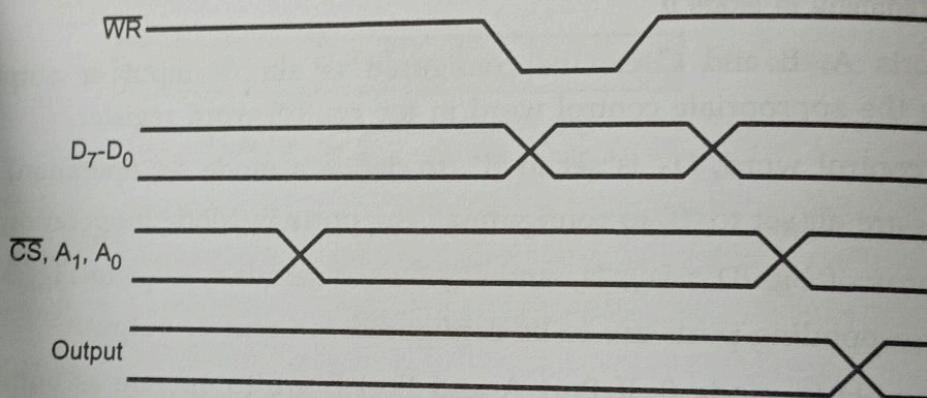


Fig. 3.4.7 Timing diagram for mode 0 output mode

After initialization of 8255 in the output mode 0, CPU can write data into the output port by initiating write command with proper port address. CPU sends data on the data bus and upon activation of \overline{WR} signal, data on the data bus gets latched on the selected output port.

Mode 0 configurations :

A			B		GROUP A			GROUP B		
D ₄	D ₃	D ₁	D ₀	PORT A	PORT C (Upper)	#	PORT B	PORT C (Lower)		
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT		
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT		
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT		
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT		

			OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT
0	1	0	1	INPUT	6	INPUT	INPUT
0	1	1	0	OUTPUT	INPUT	7	OUTPUT
0	1	1	1	OUTPUT	INPUT	8	INPUT
1	0	0	0	INPUT	OUTPUT	9	OUTPUT
1	0	0	1	INPUT	OUTPUT	10	INPUT
1	0	1	0	INPUT	OUTPUT	11	OUTPUT
1	0	1	1	INPUT	INPUT	12	INPUT
1	1	0	0	INPUT	INPUT	13	OUTPUT
1	1	0	1	INPUT	INPUT	14	INPUT
1	1	1	0	INPUT	INPUT	15	OUTPUT
1	1	1	1	INPUT	INPUT	INPUT	INPUT

3.4.6.2 Programming in Mode 1 (Input / Output with Handshake)

- Both Group A and Group B can operate in Mode 1, either together or individually, with each port containing an 8-bit latched Input or Output data port, and a 4-bit port which is used for control and status of the 8-bit port.
- When Port A is to be programmed as an input port, PC₃, PC₄ and PC₅ are used for control.
- PC₆ and PC₇ are not used and can be Input or Output, as programmed by bit D₃ of the control word.
- When Port A is programmed as an output port, PC₃, PC₆ and PC₇ are used for control and PC₄ and PC₅ can be Input or Output, as programmed by bit D₃, of the control word.
- When port B is to be programmed as an input or output port, PC₀, PC₁ and PC₂ are used for control.

Mode 1 input control signals :

1. STB (Strobe Input) :

- This is an active low input signal for 8255 and output signal for the input device.
- The input device activates this signal to indicate CPU that the data to be read is already sent on the port lines of 8255 port. Upon activation of this signal 8255 loads the data from the input port lines into the input buffer of that port.

2. IBF (Input Buffer Full) :

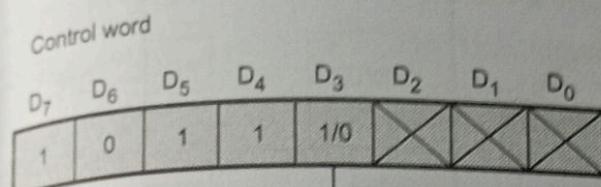
- This is an active high output signal for 8255 and an input signal for input device.
- This signal is generated by 8255 in response to \overline{STB} signal as an acknowledgement to input device.
- It also indicates to the input device that the input buffer is full and it is not ready to accept next byte from the input device. Therefore input device sends data on the port lines only when IBF signal is not active.
- The IBF signal is deactivated when CPU reads the data from input buffer of the respective port by activation of \overline{RD} signal.

3. INTR (Interrupt Request) :

- This is an active high output signal generated by 8255. A 'high' on this output can be used to interrupt the CPU when an input device is requesting service.
- The 8255 sets the INTR when \overline{STB} signal is 'one', IBF signal is 'one' and INTE is 'one', indicating CPU that the data from the input device is available in the input buffer.
- This signal is reset by the falling edge of the \overline{RD} signal i.e. immediately after reading the data from the input buffer.
- INTE (Interrupt Enable) flip-flop is used to enable or disable INTR (Interrupt request) signal.
- If INTE flip-flop is set, the interrupt request is generated depending on the status of \overline{STB} and IBF signals. If INTE flip-flop is reset, the interrupt request is not generated, allowing masking facility for the interrupt.

Mode 1 : Port A input operation

- Fig. 3.4.8 (a) shows Port A as an input port along with the control word and control signals (for handshaking with a peripheral). When the control word (as in Fig. 3.4.8 (a)) is loaded into the control register, Group A is configured in Mode 1 with Port A as an input port.
- Port A can accept parallel data from a peripheral (like a keyboard) and this data can be read by the CPU. The peripheral first loads data into Port by making the \overline{STB}_A input low. This latches the data placed by the peripheral on the common data bus into Port A. Port A acknowledges reception of data by making IBF_A high. IBF_A is set when the \overline{STB}_A input is made low, as shown in Fig. 3.4.8 (b).



PC₇, PC₆
1 = INPUT
0 = OUTPUT

RD →

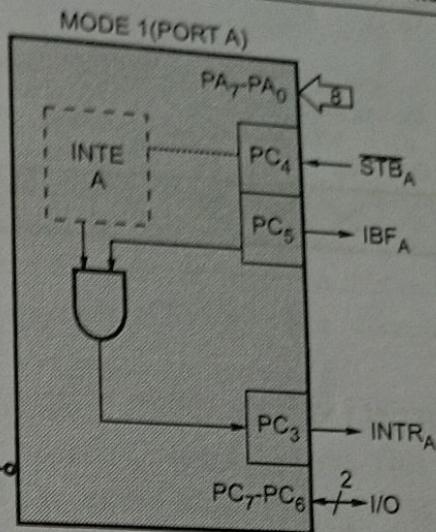


Fig. 3.4.8 (a) Port A in mode 1

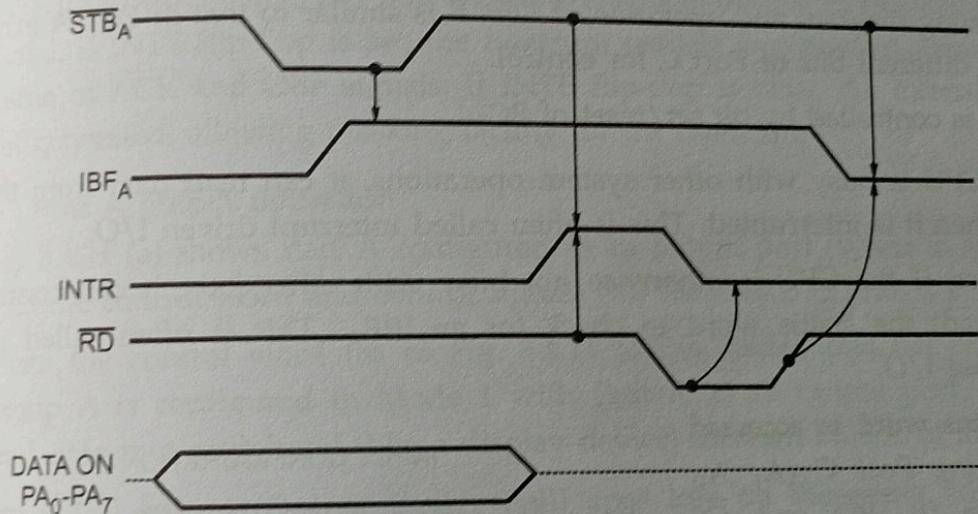


Fig. 3.4.8 (b) Timing diagram for port A in mode 1

- INT_A is an active high output signal which can be used to interrupt the CPU so that the CPU can suspend its current operation and read the data written into Port A by the peripheral. INT_A can be enabled or disabled by the INT_A flip-flop which is controlled by Bit Set-Reset operation of PC₄. INT_A is set (if enabled by setting the INT_A flip-flop) after the STB_A has gone high again and if IBF_A is high.
- On receipt of the interrupt, the CPU can be forced to read Port A. The falling edge of the RD input resets IBF_A and it goes low. This can be used to indicate to the peripheral that the input buffer is empty and that data can again be loaded into it.

Mode 1 : Port B input operation

- Fig. 3.4.9 shows Port B as an input port (when in Mode 1).

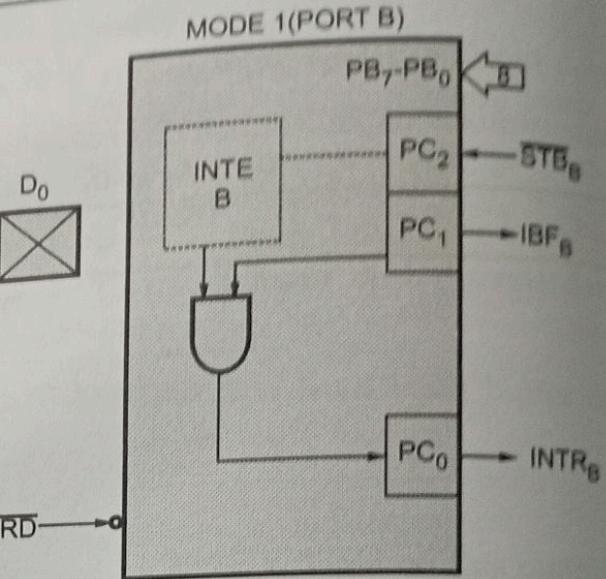
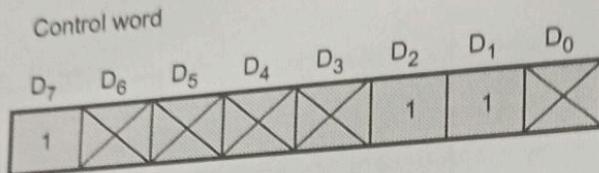


Fig. 3.4.9 Port B in mode 1

- The timing diagram and operation of Port B is similar to that of Port A except that it uses different bits of Port C for control.
- INTEB is controlled by bit set/reset of PC₂.
- If the CPU is busy with other system operations, it can read data from the input port when it is interrupted. This is often called interrupt driven I/O.
- However, if the CPU is otherwise not busy with other jobs, it can continuously poll (read) the status word to check for an IBFA. This is often called program controlled I/O.
- The status word is accessed by reading Port C (A₁ A₀ must be 10, RD and CS must be low). The status word format when Ports A and B are input ports in Mode 1, is shown in Fig. 3.4.10.

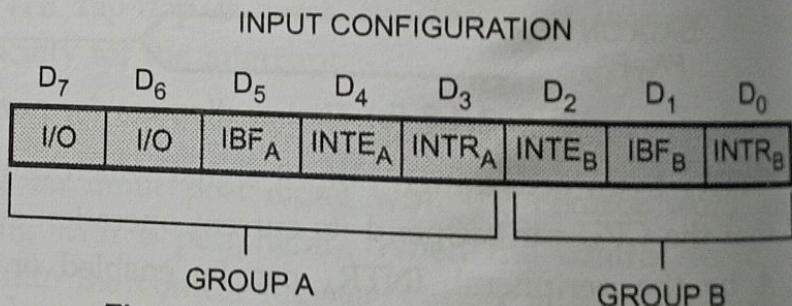


Fig. 3.4.10 Mode 1 status word (Input)

Mode 1 : Output control signals

1. OBF (Output Buffer Full) :

- This is an active low output signal for 8255 and input signal for the output device. The 8255 activates this signal to indicate output device that data is available on the output port. Upon activation of OBF signal, output device reads data from the output port and acknowledges it by ACK signal.
- The OBF signal is activated at the rising edge of the WR signal and de-activated at the falling edge of the ACK signal.

2. ACK (Acknowledge Input) :

- This is an active low input signal for 8255 and output signal for the output device.
- The output device generates this signal to indicate 8255 that the data from port A or Port B has been accepted.

3. INTR (Interrupt Request) :

- This is an active high output signal generated by 8255. A 'high' on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU.
- The 8255 sets the INTR when \overline{ACK} signal is 'one', \overline{OBF} is 'one' and INTE is 'one', indicating that the output device is ready to accept next data byte. This signal is reset by the falling edge of the \overline{WR} signal i.e. immediately after sending the data to the output port.
- INTE (Interrupt Enable) flip-flop is used to enable or disable INTR (Interrupt Request) signal. If INTE flip-flop is set, the interrupt request is generated depending on the status of \overline{ACK} and \overline{OBF} signals. If INTE flip-flop is reset, the interrupt request is not generated, allowing masking facility for the interrupt.

Mode 1 : Port A output operation

- Fig. 3.4.11 (a) shows Port A configured as an output port (When in Mode 1) along with the control word and control signals (for handshaking with a peripheral).
- When the control word (as in Fig. 3.4.11 (a)) is loaded into the control register, Group A is configured in Mode 1 with Port A as an output port. The CPU can send data to a peripheral (like a display device) through Port A of the 8255.
- The \overline{OBF}_A output (Output Buffer Full) goes low on the rising edge of the \overline{WR} signal (when the CPU writes data into the 8255).

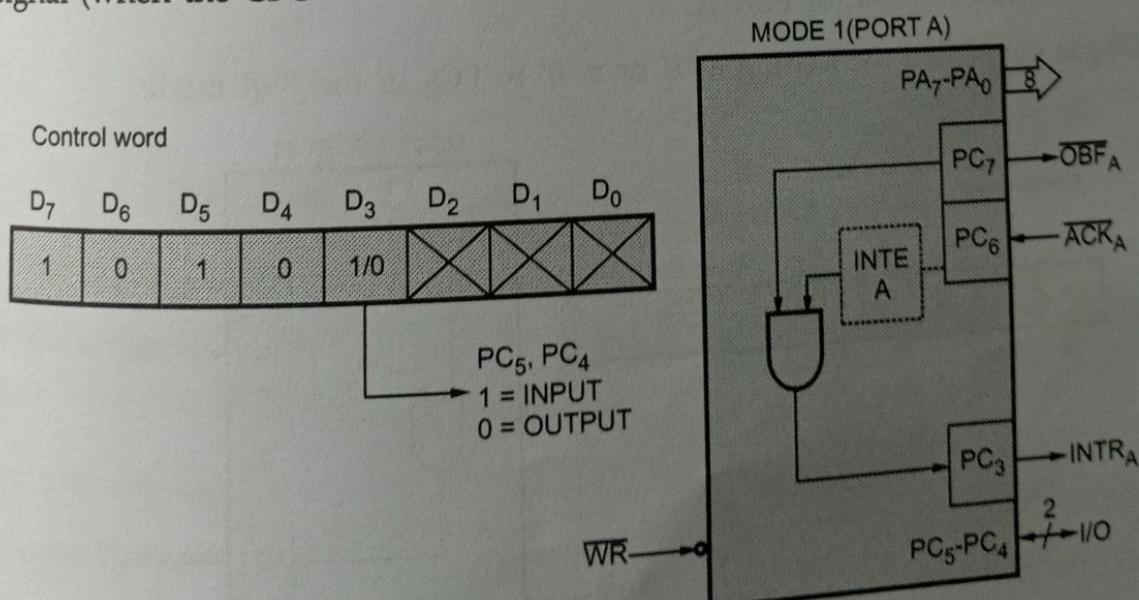


Fig. 3.4.11 (a) Port A in mode 1

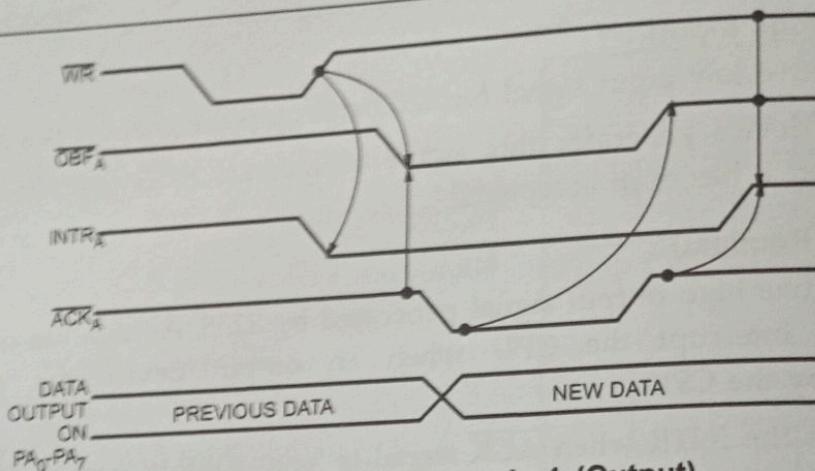


Fig. 3.4.11 (b) Port A in mode 1 (Output)

- The \overline{OBF}_A output from 8255 can be used as a strobe input to the peripheral to latch the contents of Port A.
- The peripheral responds to the receipt of data by making the \overline{ACK}_A input of the 8255 low, thus acknowledging that it has received the data sent by the CPU through Port A. The \overline{ACK}_A low sets the OBF_A signal, which can be polled by the CPU through OBF_A of the status word to load the next data when it is high again.
- $INTR_A$ is an active high output of the 8255 which is made high (if the associated $INTE_A$ flip-flop is set) when \overline{ACK}_A is made high again by the peripheral, and when OBF_A goes high again (see timing diagram in Fig. 3.4.12 (b)). It can be used to interrupt the CPU whenever the output buffer is empty. It is reset by the falling edge of WR when the CPU writes data onto Port A. It can be enabled or disabled by writing a '1' or a '0' respectively to PC_6 in the BSR mode.

Mode 1 : Port B output operation

- Fig. 3.4.12 shows Port B as an output port when in Mode 1. The operation of Port B is similar to that of Port A.
- $INTR_B$ is controlled by writing a '1' or a '0' to PC_2 in the BSR mode.

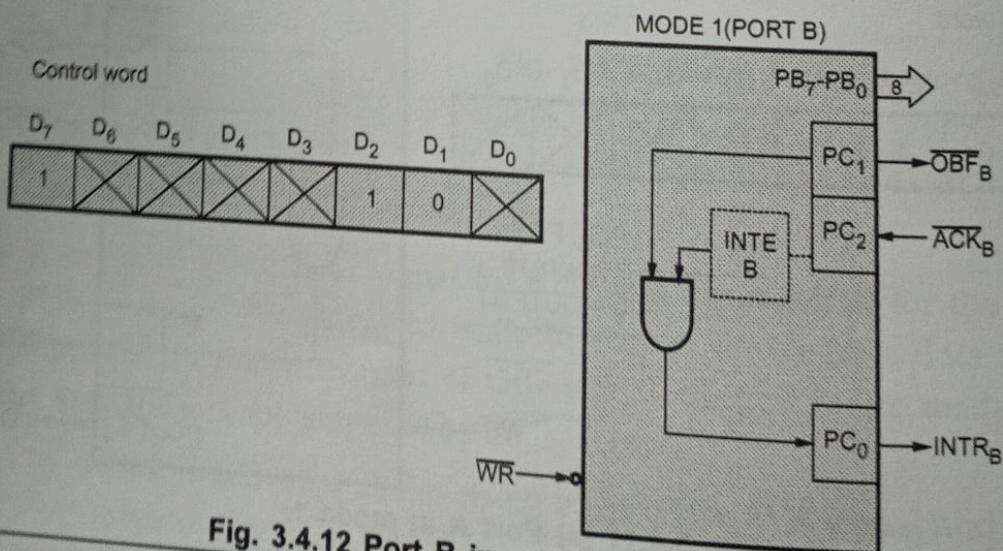


Fig. 3.4.12 Port B in mode 1 (Output)

- The status word is accessed by issuing a Read to Port C. The format of the status word when Ports A and B are Output ports in Mode 1 is shown in Fig. 3.4.13.

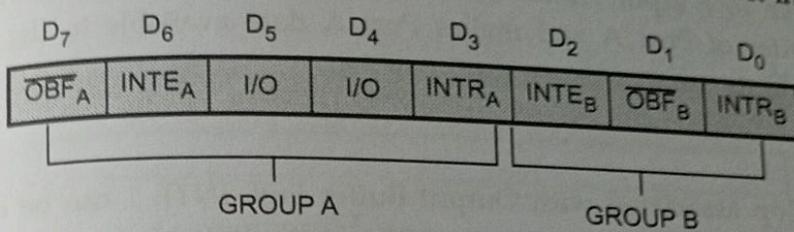


Fig. 3.4.13 Mode 1 status word (Output)

3.4.6.3 Programming in Mode 2 (Strobes Bi-directional Bus I/O)

- When the 8255 is operated in Mode 2 (by loading the appropriate control word), Port A can be used as a bi-directional 8-bit I/O bus using for handshaking.
- Port B can be programmed in Mode 0 or in Mode 1. When Port B is programmed in mode 1, PC₀ - PC₂ lines of Port C are used as handshaking signals.
- Fig. 3.4.14 shows the control word that should be loaded into the control port to configure 8255 in Mode 2.

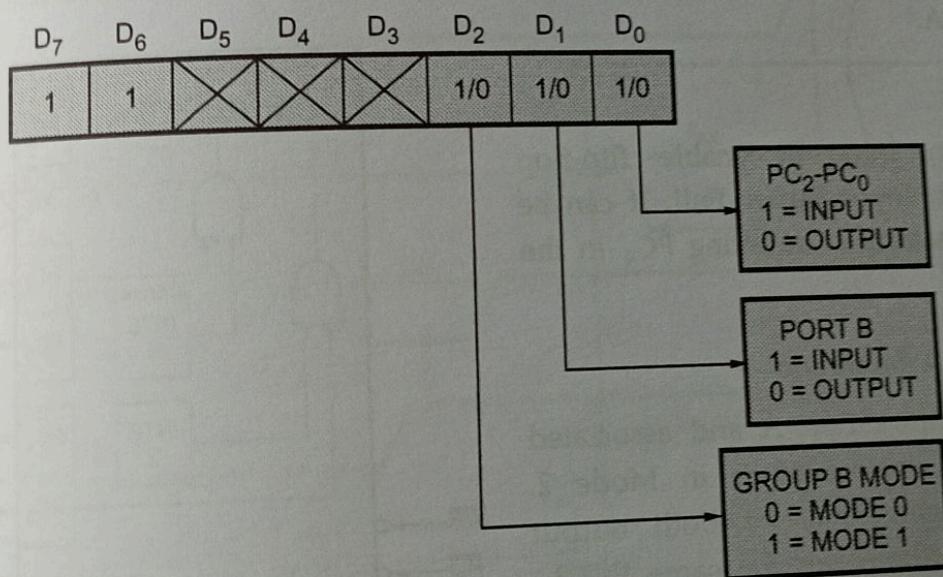


Fig. 3.4.14 Mode 2 control word

Mode 2 : Control signals

INTR (Interrupt Request) : A 'high' on this output can be used to interrupt the CPU for input or output operations.

Output control signals :

\overline{OBF}_A (Output Buffer Full)

This is an active low output which indicates that the CPU has written data into Port A.

ACK_A (Acknowledge)

This is an active low input signal (generated by the peripheral) which enables the tri-state output buffer of Port A and makes Port A data available to the peripheral. In Mode 2, Port A outputs are in tri-state until enabled.

INTE 1

This is the flip-flop associated with Output Buffer Full. INTE 1 can be used to enable or disable the interrupt by setting or resetting PC₆ in the BSR Mode.

Input control signals :**STB (Strobe Input)**

This is an active low input signal which enables Port A to latch the data available at its input.

IBF (Input Buffer Full Flip-Flop)

This is an active high output which indicates that data has been loaded into the input latch of Port A.

INTE 2

This is an Interrupt enable flip-flop associated with Input Buffer Full. It can be controlled by setting or resetting PC₄ in the BSR mode.

Mode 2 : Port A operation

Fig. 3.4.15 shows Port A and associated control signals when 8255 is in Mode 2. Interrupts are generated for both output and input operations on the same INTR_A (PC₃) line.

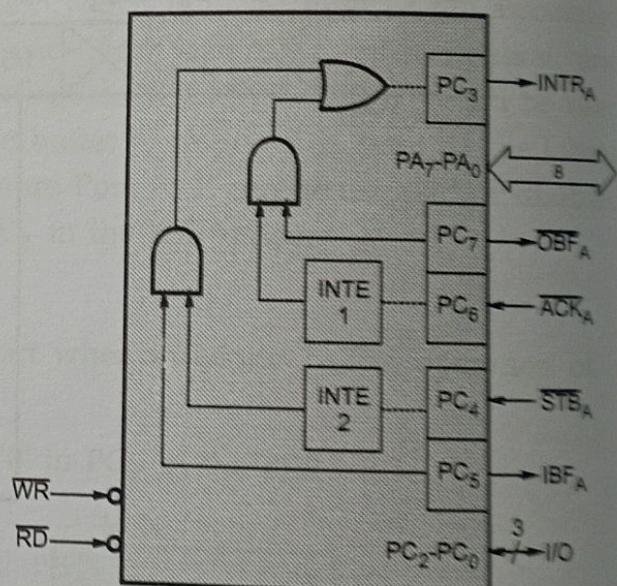


Fig. 3.4.15 Mode 2 operation

Status word in mode 2

- The status word for Mode 2 (accessed by reading Port C) is shown in Fig. 3.4.16.
- D₇-D₃ of the status word carry information about \overline{OBF}_A , INTE₁, IBF_A, INTE₂, INTR_A.
- The status of the bits D₂ - D₀ depend on the mode setting of Group B. If B is programmed in Mode 0, D₂ - D₀ are the same as PC₂ - PC₀ (simple I/O);

however if B is in Mode 1, $D_2 - D_0$ carry information about the control signals for Port B (as in Fig. 3.4.11 or Fig. 3.4.14), depending upon whether Port B is an input port or output port respectively.

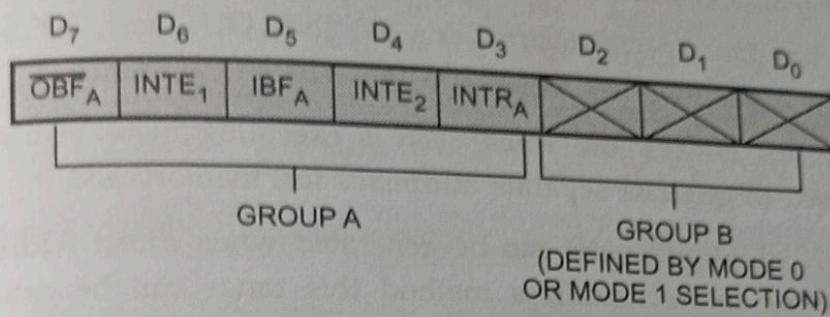


Fig. 3.4.16 Status word for mode 2

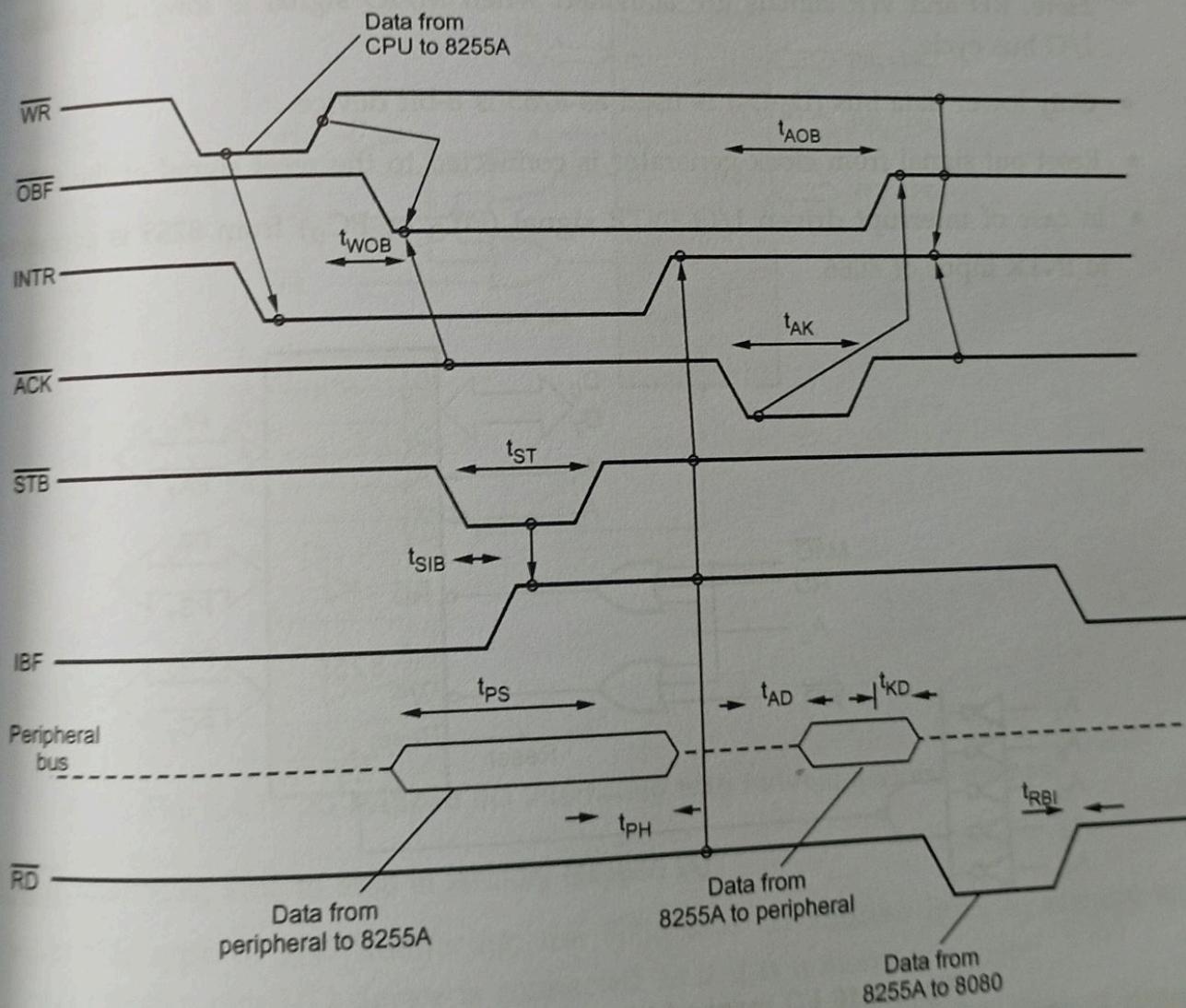


Fig. 3.4.17 Timing diagram for mode 2

3.4.7 Interfacing of 8255 with 8086

3.4.7.1 Interfacing 8255 to 8086 in I/O Mapped I/O Mode

- The 8086 has four special instructions IN, INS, OUT and OUTS to transfer data through the input/output ports in I/O mapped I/O system.
- M/ \overline{IO} signal is always low when 8086 is executing these instructions. So M/ \overline{IO} signal is used to generate separate addresses for, memory and input/output.
- Only 256 (2^8) I/O addresses can be generated when direct addressing method is used. By using indirect address method this range can be extended up to 65536 (2^{16}) addresses.
- Fig. 3.4.18 shows the interfacing of 8255 with 8086 in I/O mapped I/O technique. Here, \overline{RD} and \overline{WR} signals are activated when M/ \overline{IO} signal is low, indicating I/O bus cycle.
- Only lower data bus (D_0 - D_7) is used as 8255 is 8-bit device.
- Reset out signal from clock generator is connected to the reset signal of the 8255.
- In case of interrupt driven I/O INTR signal (PC₃ or PC₀) from 8255 is connected to INTR input of 8086.

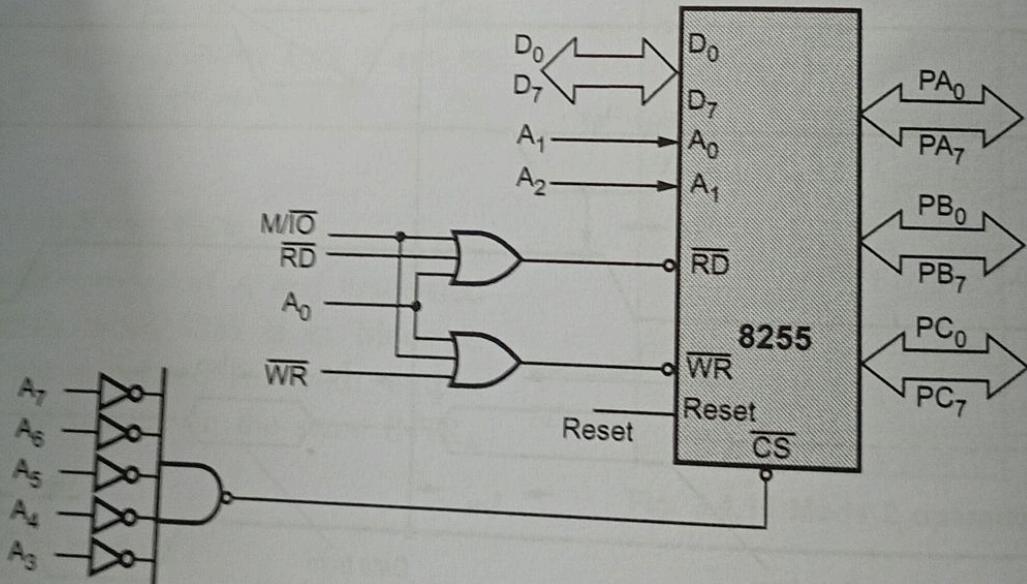


Fig. 3.4.18 I/O mapped I/O (Direct address method)

Port / Control register	Address lines								Address
	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	
Port A	0	0	0	0	0	0	0	0	00H
Port B	0	0	0	0	0	0	1	0	02H
Port C	0	0	0	0	0	1	0	0	04H
Control register	0	0	0	0	0	1	1	0	06H

Note It is assumed that the direct addressing is used.

Example 3.4.7 Interface an 8255 with 8086 so as to have Port-A address BCD1H, Port-B address BCD3H, Port-C address BCD5H and control word register address BCD7H.

Solution :

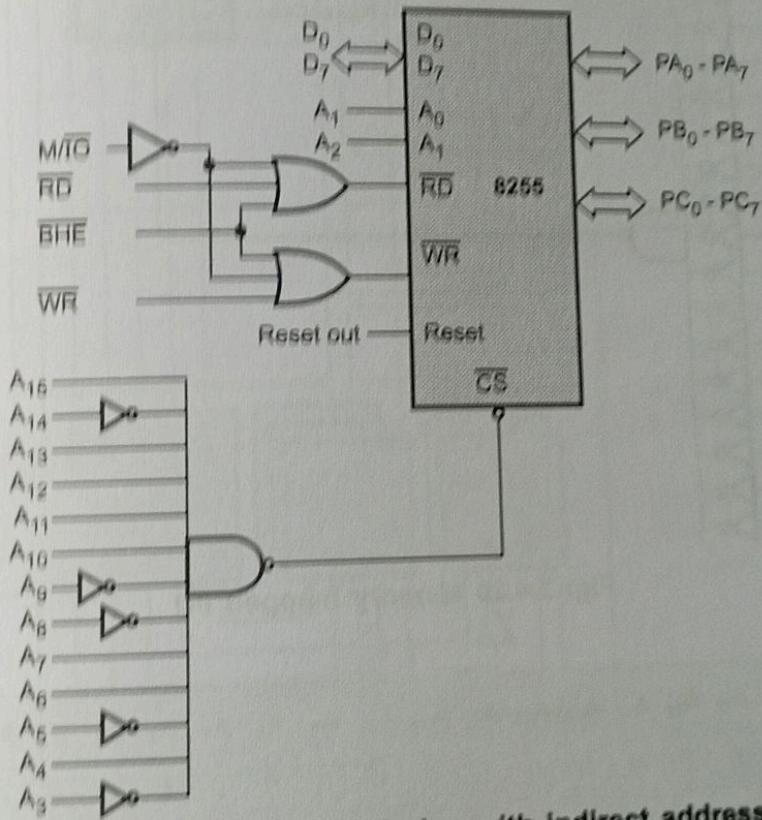


Fig. 3.4.19 I/O mapped I/O Interfacing with indirect address method

3.4.7.2 Interfacing 8255 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. 3.4.20 shows the interfacing of 8255 with 8086 in memory mapped I/O technique. Here RD and WR signals are activated when M/IO signal is high, indicating memory bus cycle.

- Address lines $A_0 - A_1$ are used by 8255 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.

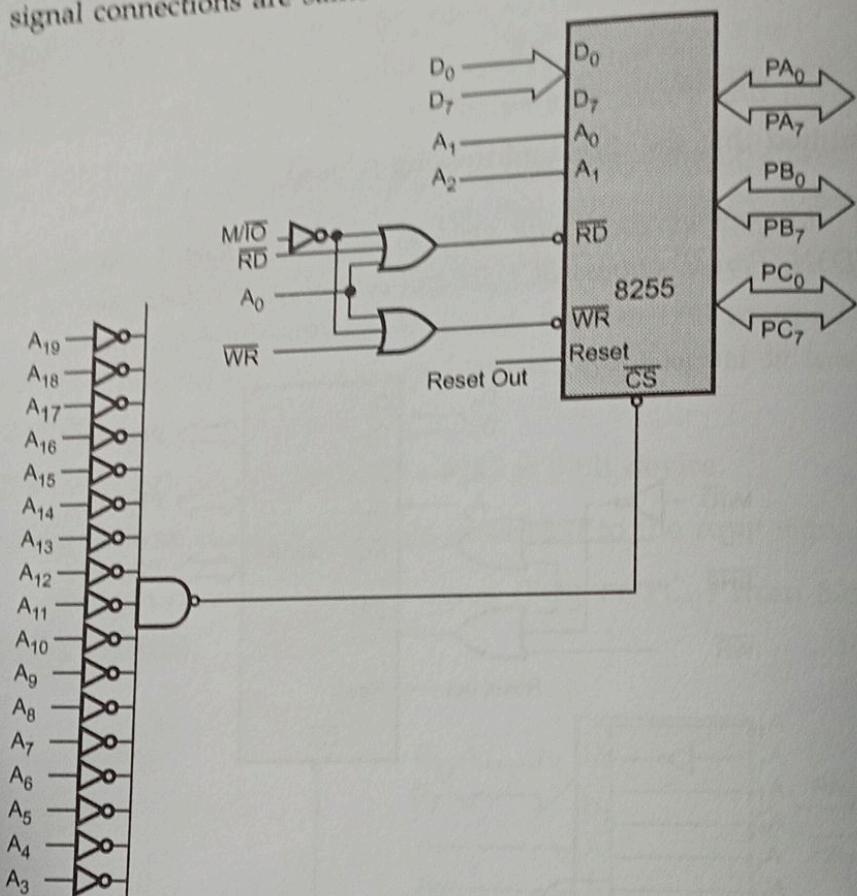


Fig. 3.4.20 Memory mapped I/O

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	
Port A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000H	
Port B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00001H	
Port C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	00002H	
Control register	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	00004H

Example 3.4.8 Interface 8255 PPI with 8086 microprocessor in maximum mode. Draw interfacing diagram and mention address map for 8255.

Solution : Address map for 8255

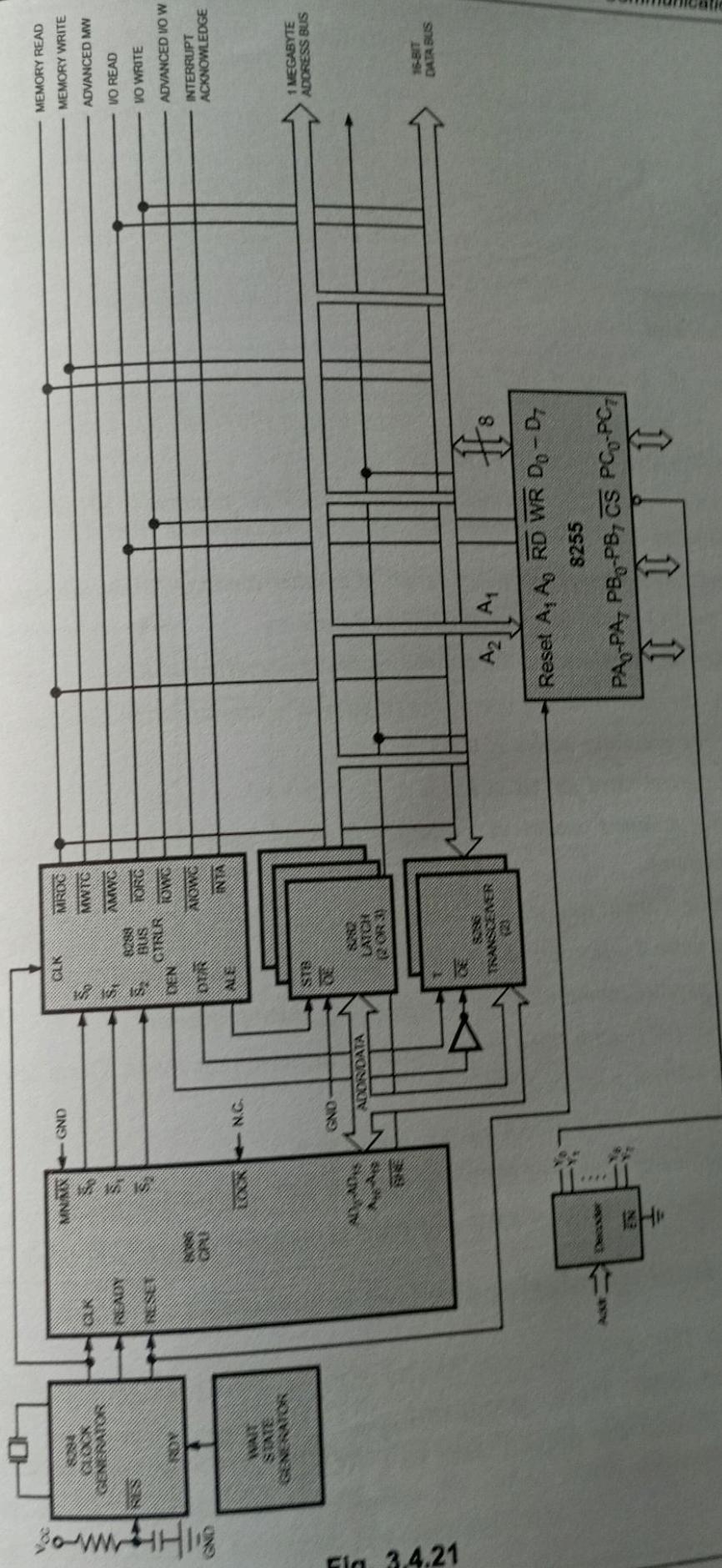


Fig. 3.4.21

Port/Register	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	Address
Port A	0	0	0	0	0	0	1	0	00H
Port B	0	0	0	0	0	1	0	0	02H
Port C	0	0	0	0	0	1	1	0	04H
Control register	0	0	0	0	0	1			06H

Review Questions

1. What are the features of 8255 ?
2. Explain in detail about the parallel communication interface.
3. Explain important signals of 8255 .
4. Discuss the organization and architecture of 8255 programmable peripheral interface IC with a functional block diagram.
5. Draw and explain the functional diagram of parallel communication interfacing chip.
6. Explain the I/O mode operation of 8255 in detail.
7. With a neat diagram discuss the various modes of operation of 8255 .
8. What are the basic modes of operations in parallel communication interfaces ?
9. What are the operating modes of 8255 ?
10. Draw the control word format of 8255 .
11. Discuss the different modes of operation of 8255 parallel communication interface with suitable diagrams.
12. Describe mode 1 input and mode 1 output operation of 8255 with timing diagrams.
13. Explain the mode 0 operation of 8255 programmable peripheral interface.
14. Explain the parallel communication interface with microprocessor.
15. Interface 8255 (PPI) with 8086 microprocessor in minimum mode. Draw interfacing diagram and mention address map for 8255 .
16. Draw and explain in brief 8255A system connection.
17. Draw the interfacing scheme of 8255 and 8086 in I/O mapped I/O mode.
18. Draw the interfacing scheme of 8255 and 8086 in memory mapped I/O mode.

3.5 Digital to Analog Interface

- Fig. 3.5.1 (See Fig. 3.5.1 on next page) shows the interfacing of DAC 0808 with microprocessor 8086. Here, programmable peripheral interface, 8255 is used as parallel port to send the digital data to DAC.

Port / Register	Address
Port A	0 0
Port B	0 2
Port C	0 4
Control Register	0 6

Program :

```

MOV AL,80 H      ; Initialization control word for 8255 to
OUT 06, AL      ; Configure all ports as output ports
MOV AL, data     ; Load 8-bit data to be sent at the input of 0808 DAC
OUT 00, AL      ; Send data on port A.

```

We now see how different waveforms can be generated using this circuit.

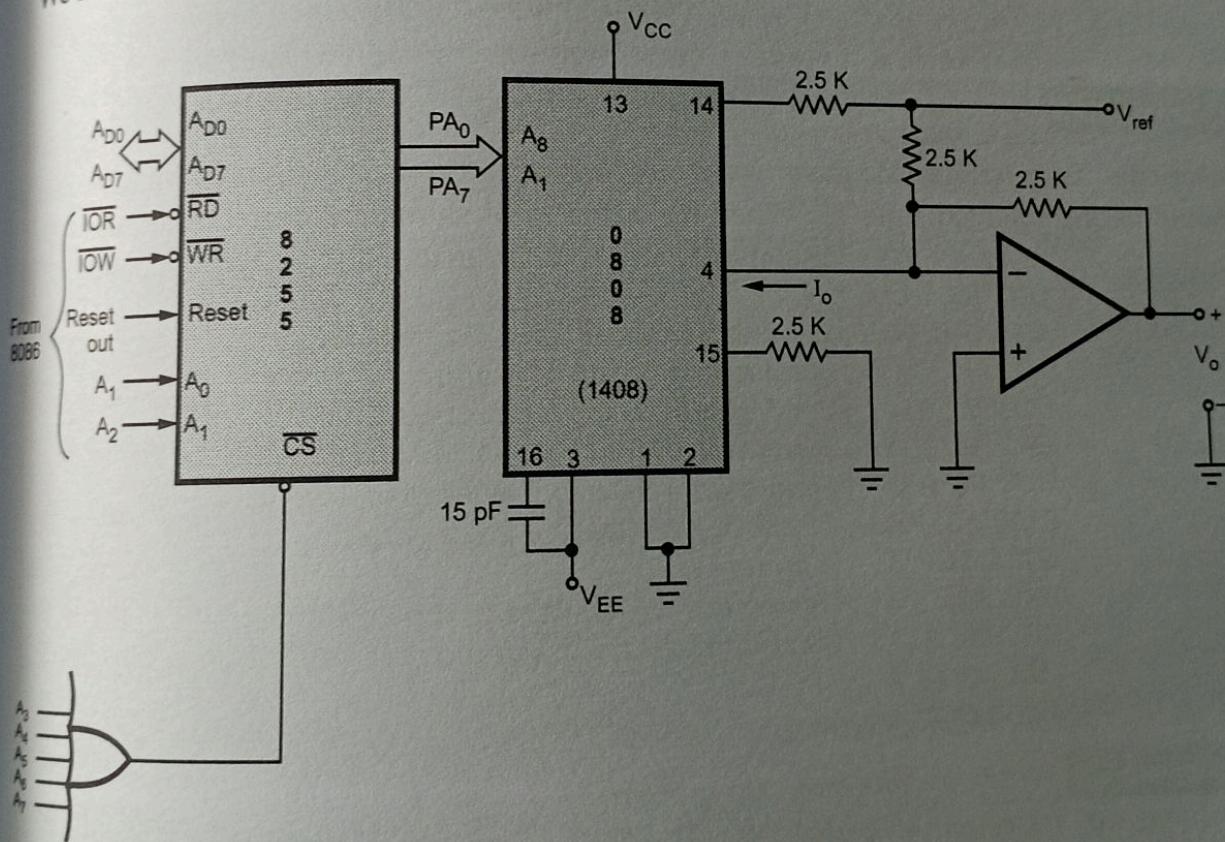


Fig. 3.5.1 Interfacing of 0808 with microprocessor

Example 3.5.1 Generate square wave using DAC 0808.

Solution : To generate square wave first we have to output FF and then 00 on Port A of 8255. The output of 8255 (Port A) connected to DAC 0808. According to frequency requirement delay is provided in between the two outputs.

Program	MOV AL, 80H OUT 06, AL MOV AL, 00	; Initialise 8255 to configure all ports as output ports ; Load A with digital data corresponds to -2.5 V output
LOOP :	OUT 00, AL CALL DELAY MOV AL, FFH	; Send digital data to the input of DAC 0808 ; Wait for specified time. ; Load A with digital data corresponds to +2.5 V output
	OUT 00, AL CALL DELAY	; Send digital data to the input of DAC 0808. ; Wait for specified time
	JMP LOOP	; Repeat

Delay Program

BACK :	MOV BL, 08 DEC B JNZ BACK	; Load delay count in register ; Decrement count ; Check if count = 0 ; otherwise repeat
	RET	

Example 3.5.2 Generate triangular wave using DAC 0808.

Solution : To generate triangular wave we have to output data from 00 initially, and it should be incremented upto FF. When it reaches FF it should be decremented upto 00.

MOV AL, 80H	; Initialise 8255 to configure all ports as output ports
OUT 06, AL	
MOV AL, 00	; Load accumulator with digital data corresponds to -2.5 V output
OUT 00, AL	; Send digital to the input of DAC 0808.
LOOP_1:	INC AL ; Increment digital data in the accumulator
OUT 00, AL	; Send digital data to the input of DAC 0808.
CMP AL, FFH	; Check digital data for Peak output
JNZ LOOP_1	; If no repeat,
LOOP_2:	DEC AL ; Decrement digital data in the accumulator
OUT 00, AL	; Send digital data to the input of DAC 0808.
JNZ LOOP_2	
JMP LOOP_1	

Example 3.5.3 Write a program to generate triangular waveform of period 10 ms. The CPU runs at 5 MHz clock frequency.

Solution : The Fig. 3.5.2 shows waveform to be generated. As shown in the Fig. 3.5.2, the time period for half cycle is 5 ms. In half cycle we execute loop 256 times. Thus execution time for loop will be 5 ms/256

$$= 19.53 \mu\text{s}$$

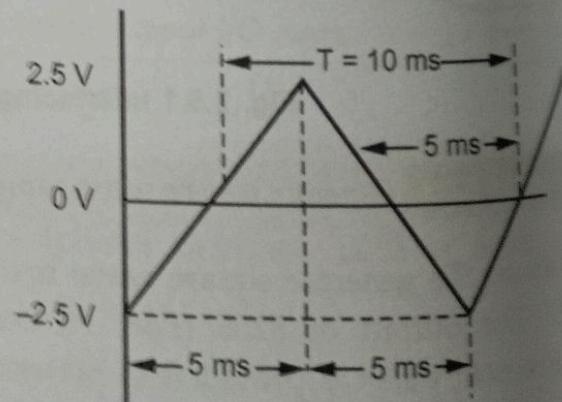


Fig. 3.5.2

The loop will be

LOOP_2 :	DEC AL	3	Clock cycles
	OUT 00, AL	10	Clock cycles
	JNZ LOOP_2	16	Clock cycles

$$\text{Time for 1 cycle} = \frac{1}{5 \times 10^6} = 0.2 \mu\text{s}$$

$$\text{Time for loop} = (3 + 10 + 16) \times 0.2 = 5.8 \mu\text{s}$$

Thus the delay time required in the inner loop will be

$$19.53 \mu\text{s} - 5.8 \mu\text{s} = 13.73 \mu\text{s} = 69 \text{ Clock cycles.}$$

BACK :	MOV CL, 04 →	4	=	4
	DEC CL →	3+3+3+3	=	12
	JNZ BACK →	16+16+16+4	=	<u>52</u> 69

The above three instructions in the loop will take 69 clock cycles, i.e. 13.73 μs to execute. Thus the final routine to generate desired waveform is as follows :

MOV AL, 80H	; Initialize 8255 to configure all
OUT 08, AL	; Ports as output ports
MOV AL, 00	; Load accumulator with digital data
OUT 00, AL	; Corresponds to -2.5 V output
LOOP_1 : INC AL	; Increment digital data in the accumulator
OUT 00, AL	; Send digital data to the input of DAC 0808
BACK1 : MOV CL, 04	; Delay to consume 69 T cycles
DEC CL	;
JNZ BACK1 ;	
CMP AL, FFH	; Check digital data for peak output
JNZ LOOP_1	; If no repeat
LOOP_2 : MOV CL, 04	; Delay to consume 69 T cycles
BACK2 : DEC CL	
JNZ BACK2	
DEC AL	; Decrement digital data in the accumulator
OUT 00, AL	; Send digital data to the input of DAC 0808
JNZ LOOP_2	
JMP LOOP_1	

Example 3.5.4 Interface a typical 12-bit DAC with 8255 and write program to generate triangular waveform of period 10 ms. The CPU runs at 5 MHz clock frequency.

Solution : In triangular waveform period for 1 ramp = $10 \text{ ms}/2 = 5 \text{ ms}$ and minimum time required to transfer data is about $40 \mu\text{s}$.

Therefore, steps in one ramp = $\frac{5 \text{ ms}}{8 \mu\text{s}} = 625$. So to reach from 0 - 4096 we have 625 steps

i.e. we have increment the digital count by $4096/625 = 6.55 = 7$

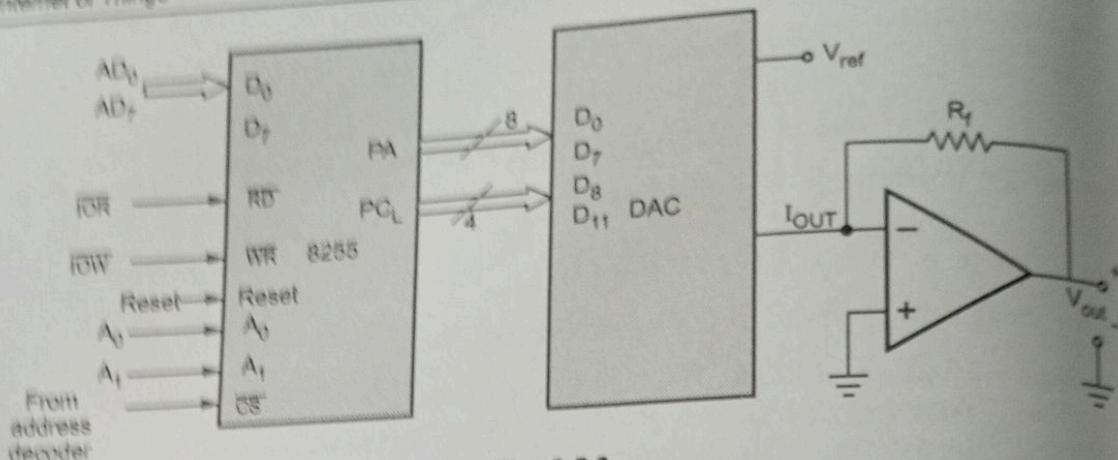


Fig. 3.5.3

With increment of 7 we get 585 steps and for each data transfer we get time

$$= \frac{5 \text{ ms}}{585} = 8.547 \mu\text{s}$$

The clock frequency of 8086 is given 5 MHz,

$$\text{Time for one clock cycle} = \frac{1}{5 \text{ MHz}} = 0.2 \mu\text{s}$$

$$\text{Therefore, clock cycles in one loop} = \frac{8.547 \mu\text{s}}{0.2 \mu\text{s}} \approx 43$$

Program :

START :	MOV CX, 0FFFH	
	MOV BX, 0000H	Clock cycles
BACK :	MOV AL, BL ...	2
	OUT PA, AL	10
	MOV AL, BH	2
	OUT PC, AL	10
	INC BX	2
	NOP	3
	INC DX	2
	DEC CX	2
	JNZ BACK	10

43

BACK1 :	MOV CX, 0FFFH
	DEC BX
	MOV AL, BL
	OUT PA, AL
	MOV AL, BH
	OUT PC, AL
	NOP

INC DX
DEC CX
JNZ BACK1
JMP START

Example 3.5.5 Generate sine wave using DAC 0808.

Solution : To generate sine wave we have to output digital equivalent values which will represent sine wave as shown in the Fig. 3.5.4. Digital data 00H represents - 2.5 V, 7FH represents 0 V and FFH represents + 2.5 V. The digital equivalent for sine wave can be calculated as follows.

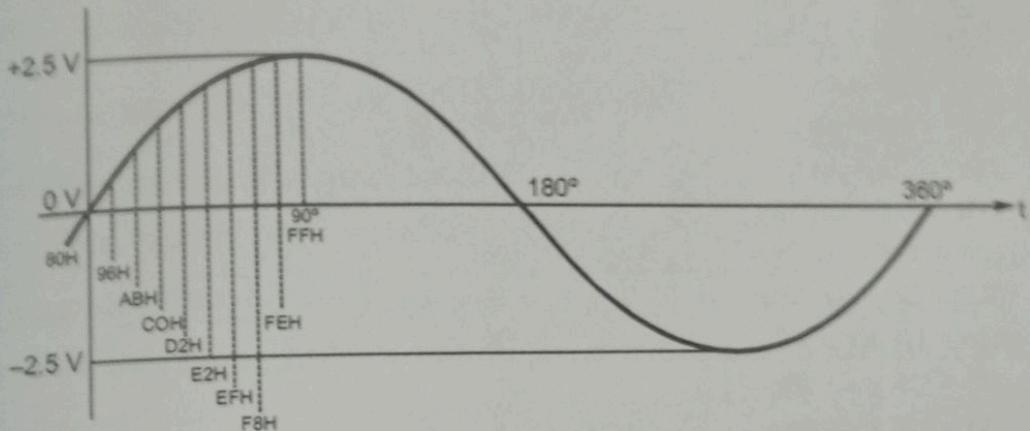


Fig. 3.5.4

We know that $\sin 0^\circ = 0$ and $\sin 90^\circ = 1$. The range $\sin 0^\circ$ to $\sin 90^\circ$ is distributed over digital range of 7FH to FFH i.e. (FFH - 7FH) 128 decimal steps. Therefore, taking 128 as a offset we can write,

Digital Equivalent Value (DEV) for $\sin x = (128 + 128 \times \sin x)$

where x is a angle in degrees and digital value is in decimal.

Lookup table shows the digital equivalent values for sine wave.

Degrees	Equation	Digital equivalent in decimal	Digital equivalent in hex
0°	(128 + 128 × sin 0)	128	80H
10°	(128 + 128 × sin 10)	150	96H
20°	(128 + 128 × sin 20)	171	ABH
30°	(128 + 128 × sin 30)	192	C0H
40°	(128 + 128 × sin 40)	156	D2H
50°	(128 + 128 × sin 50)	226	E2H
60°	(128 + 128 × sin 60)	239	EFH
70°	(128 + 128 × sin 70)	248	F8H
80°	(128 + 128 × sin 80)	254	FEH

		256 → 255	
90 °	(128 + 128 × sin 90)	254	FFH
100 °	(128 + 128 × sin 100)	248	FEH
110 °	(128 + 128 × sin 110)	239	F8H
120 °	(128 + 128 × sin 120)	226	EFH
130 °	(128 + 128 × sin 130)	156	E2H
140 °	(128 + 128 × sin 140)	192	D2H
150 °	(128 + 128 × sin 150)	171	C0H
160 °	(128 + 128 × sin 160)	150	ABH
170 °	(128 + 128 × sin 170)	128	96H
180 °	(128 + 128 × sin 180)	106	6AH
190 °	(128 + 128 × sin 190)	84	54H
200 °	(128 + 128 × sin 200)	64	40H
210 °	(128 + 128 × sin 210)	46	2EH
220 °	(128 + 128 × sin 220)	30	1EH
230 °	(128 + 128 × sin 230)	17	11H
240 °	(128 + 128 × sin 240)	08	08H
250 °	(128 + 128 × sin 250)	02	02H
260 °	(128 + 128 × sin 260)	00	00H
270 °	(128 + 128 × sin 270)	02	02H
280 °	(128 + 128 × sin 280)	08	08H
290 °	(128 + 128 × sin 290)	17	11H
300 °	(128 + 128 × sin 300)	30	1EH
310 °	(128 + 128 × sin 310)	46	2EH
320 °	(128 + 128 × sin 320)	64	40H
330 °	(128 + 128 × sin 330)	84	54H
340 °	(128 + 128 × sin 340)	106	6AH
350 °	(128 + 128 × sin 350)	128	80H
360 °	(128 + 128 × sin 360)		

Program :

```

START:    MOV AL,      80H      ; Initialize 8255 to configure all
          OUT 06H,    AL      ; Ports as output ports
          MOV CL,      25H      ; Initialize counter
          MOV BX,      OFFSET TABLE ; Initialize pointer to point to lookup table
BACK:     MOV AL, [BX]      ; Get the digital equivalent data
          OUT 00,      AL      ; from lookup table
          INC BX           ; Send digital data to DAC
          DEC CL           ; Increment lookup table pointer
          JNZ BACK         ; Decrement counter
          JMP START        ; If not zero, go to BACK
          TABLE DB        80, 96H, ABH ..... 54H, 6AH, 80H ; Repeat

```

Review Questions

1. Interface a typical 8-bit DAC with 8255.
2. Draw and explain the typical interfacing circuit for DAC 0808.
3. Draw and explain the typical interfacing circuit for DAC 0808 in the bipolar range.
4. Explain the procedure of interfacing D/A and A/D converter circuit.

3.6 Analog to Digital Interface

- The ADC 0808 and ADC 0809 are monolithic CMOS devices with an 8-channel multiplexer. These devices are also designed to operate from common microprocessor control buses, with tri-state output latches driving the data bus.

3.6.1 Features

- 8-bit successive approximation ADC.
- 8-channel multiplexer with address logic.
- Conversion time 100 μ s.
- It eliminates the need for external zero and full-scale adjustments.
- Easy to interface to all microprocessors.
- It operates on single 5 V power supply.
- Output meet TTL voltage level specifications.

3.6.2 Pin Diagram

- Fig. 3.6.1 shows pin diagram of 0808/0809 ADC.

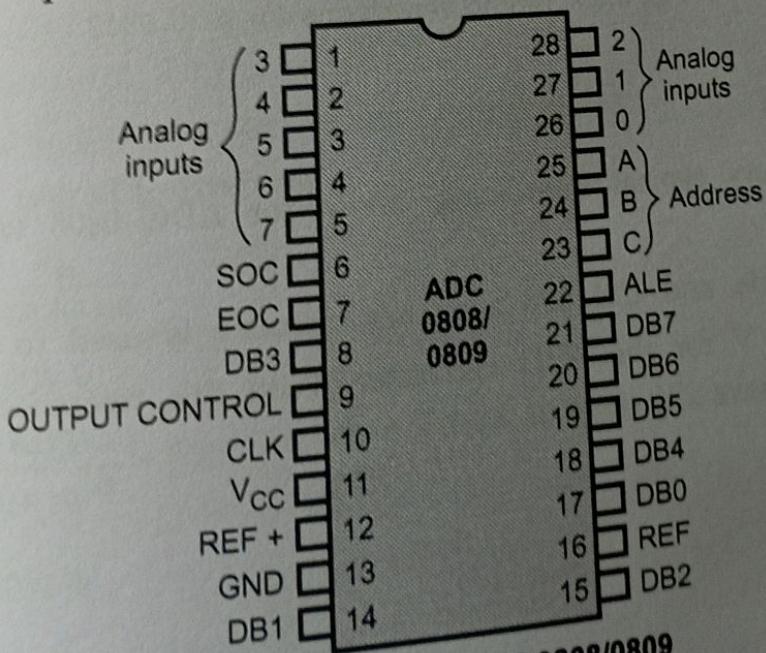


Fig. 3.6.1 Pin diagram of 0808/0809

3.6.3 Operation

- ADC 0808/0809 has eight input channels, so to select desired input channel, it is necessary to send 3-bit address on A, B and C inputs.
- The address of the desired channel is sent to the multiplexer address inputs through port pins. After at least 50 ns, this address must be latched. This can be achieved by sending ALE signal.
- After another 2.5 μ s, the Start Of Conversion (SOC) signal must be sent high and then low to start the conversion process.
- To indicate end of conversion ADC 0808/0809 activates EOC signal.
- The microprocessor system can read converted digital word through data bus by enabling the output enable signal after EOC is activated. This is illustrated in Fig. 3.6.2.

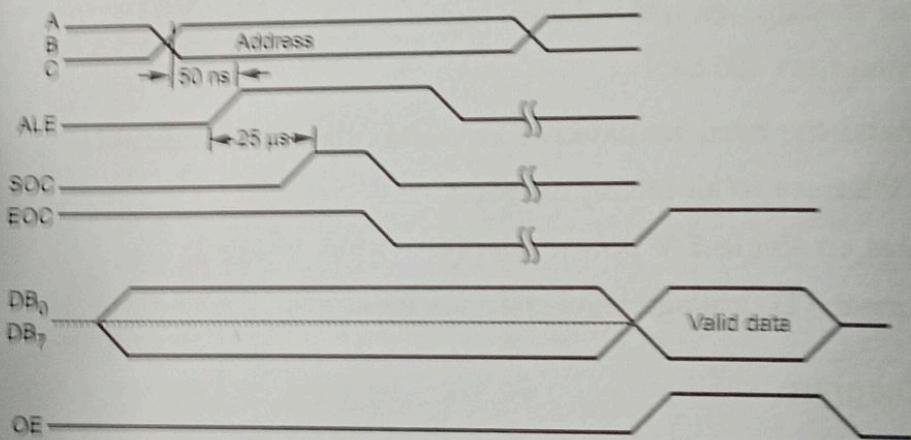


Fig. 3.6.2 Timing waveforms for ADC 0808

3.6.4 Interfacing

- Fig. 3.6.3 shows typical interfacing circuit for ADC 0808 with microprocessor system.
- The zener diode and LM 308 amplifier circuitry is used to produce a V_{CC} an $+V_{REF}$ of 5.12 V for the A/D converter. With this reference voltage the A/D converter will have 256 steps of 20 mV each.

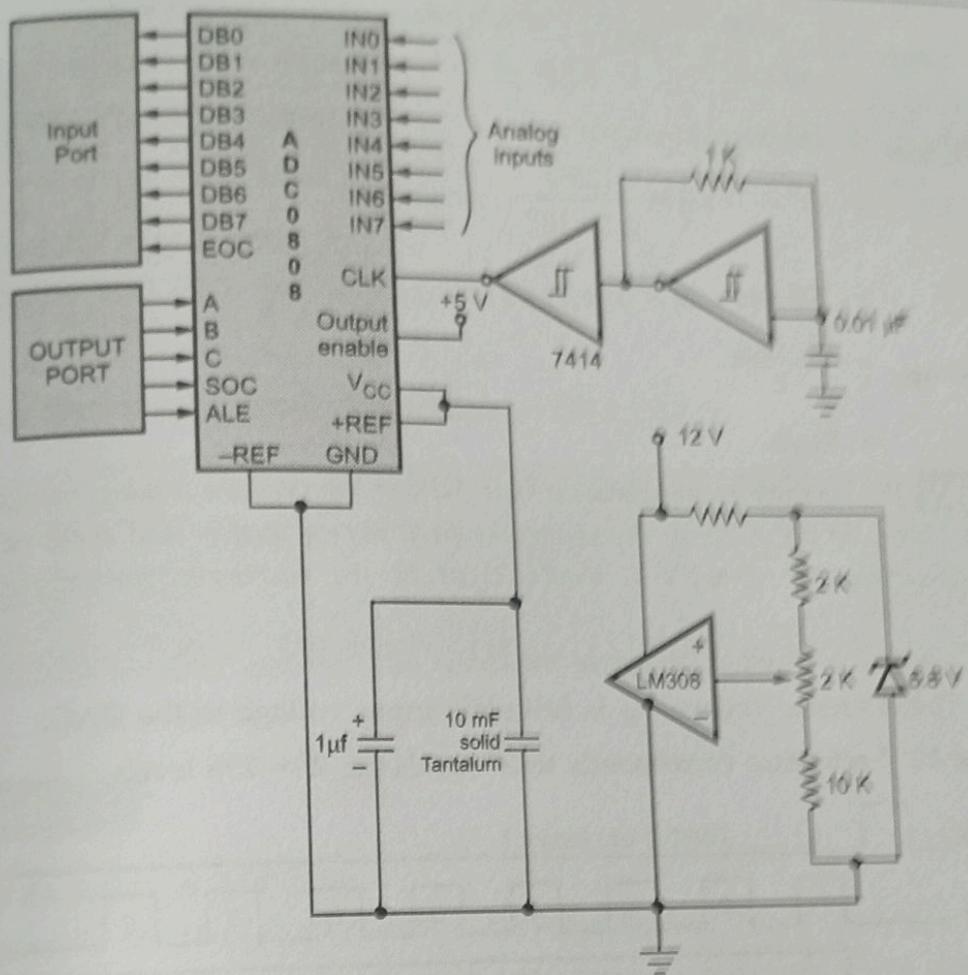


Fig. 3.6.3 Typical interface for 0808/0809

Example 3.6.1 An 8-bit digital ramp ADC with a 40 mV resolution uses a clock frequency of 2.5 MHz a comparator with $V_T = 1 \text{ mV}$. Determine the following values :

- The digital output for $V_{in} = 6.000 \text{ V}$.
- The digital output for $V_{in} = 6.035 \text{ V}$.
- The maximum and average conversion times for this ADC.

Solution : The digital output is given by,

$$\text{Count} = \frac{V_{in}}{\text{Resolution}}$$

i) $V_{in} = 6.000 \text{ V}$

$$\therefore \text{Count} = \frac{6.000}{40 \times 10^{-3}} = 150$$

ii) $V_{in} = 6.035 \text{ V}$

$$\therefore \text{Count} = \frac{6.035}{40 \times 10^{-3}} \equiv 150$$

$$\begin{aligned}\text{iii) } (T_C)_{\max} &= (\text{Maximum count}) \times T \\ &= (2^n - 1) \times \frac{1}{f} = \frac{(255)}{2.5 \times 10^6} \\ &= 102 \mu\text{s} \\ (T_C)_{\text{average}} &= \frac{(T_C)_{\max}}{2} \\ &= 51 \mu\text{s}\end{aligned}$$

Example 3.6.2 Use successive approximation type ADC to convert the analog voltage 7.28 V to 8-bit binary. If $V_R = 10$ V, determine the final binary answer and the error present. Show the timing waveforms that would occur in the successive approximation ADC process.

Note : V_R = Full scale input voltage to the DAC.

Solution : The reference voltage V_R is full scale input voltage to the DAC.

Hence the 10 V reference corresponds to 2^n levels i.e. $2^8 = 256$ levels.

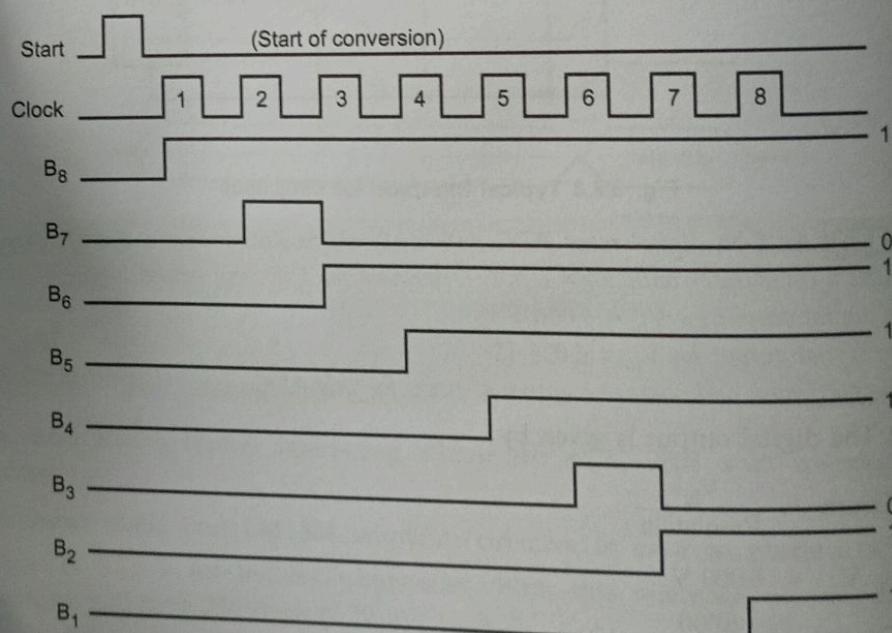


Fig. 3.6.4

Hence one level corresponds to,

$$= \frac{10}{256} = 0.03906 \text{ V}$$

This is nothing but the resolution of the ADC.

The timing waveforms are shown in the Fig. 3.6.4

Hence the final binary number is,

$$= (10111011)_2$$

Hence the total output count is,

$$\begin{aligned} &= (1 \times 128) + (0 \times 64) + (1 \times 32) + (1 \times 16) + (1 \times 8) + (0 \times 4) + (1 \times 2) + (1 \times 1) \\ &= 187 \end{aligned}$$

$$\text{Actual output} = \text{Count} \times \text{Resolution} = 187 \times 0.03906$$

$$= 7.3046 \text{ V}$$

$$= 7.28 \text{ V}$$

$$\text{Exact value} = 7.28 \text{ V}$$

$$\text{Error} = 7.3046 - 7.28$$

$$= 0.024 \text{ V}$$

Review Questions

1. List the features of ADC 0808.
2. Explain the interfacing of ADC 0808 to the microprocessor.
3. Show a typical 8-bit ADC interface with 8086. Explain functionality of each signal used.
4. Explain the procedure of interfacing D/A and A/D converter circuit.

3.7 Stepper Motor Interfacing

- * A stepper motor is a digital motor. It can be driven by digital signal.
- * Fig. 3.7.1 shows the typical 2 phase motor interfaced using 8255. Motor shown in the circuit has two phases, with center-tap winding. The center taps of these windings are connected to the 12 V supply. Due to this, motor can be excited by grounding four terminals of the two windings.
- * Motor can be rotated in steps by giving proper excitation sequence to these windings. The PA0 through PA3 pins of 8255 are used to generate excitation signals in the proper sequence.

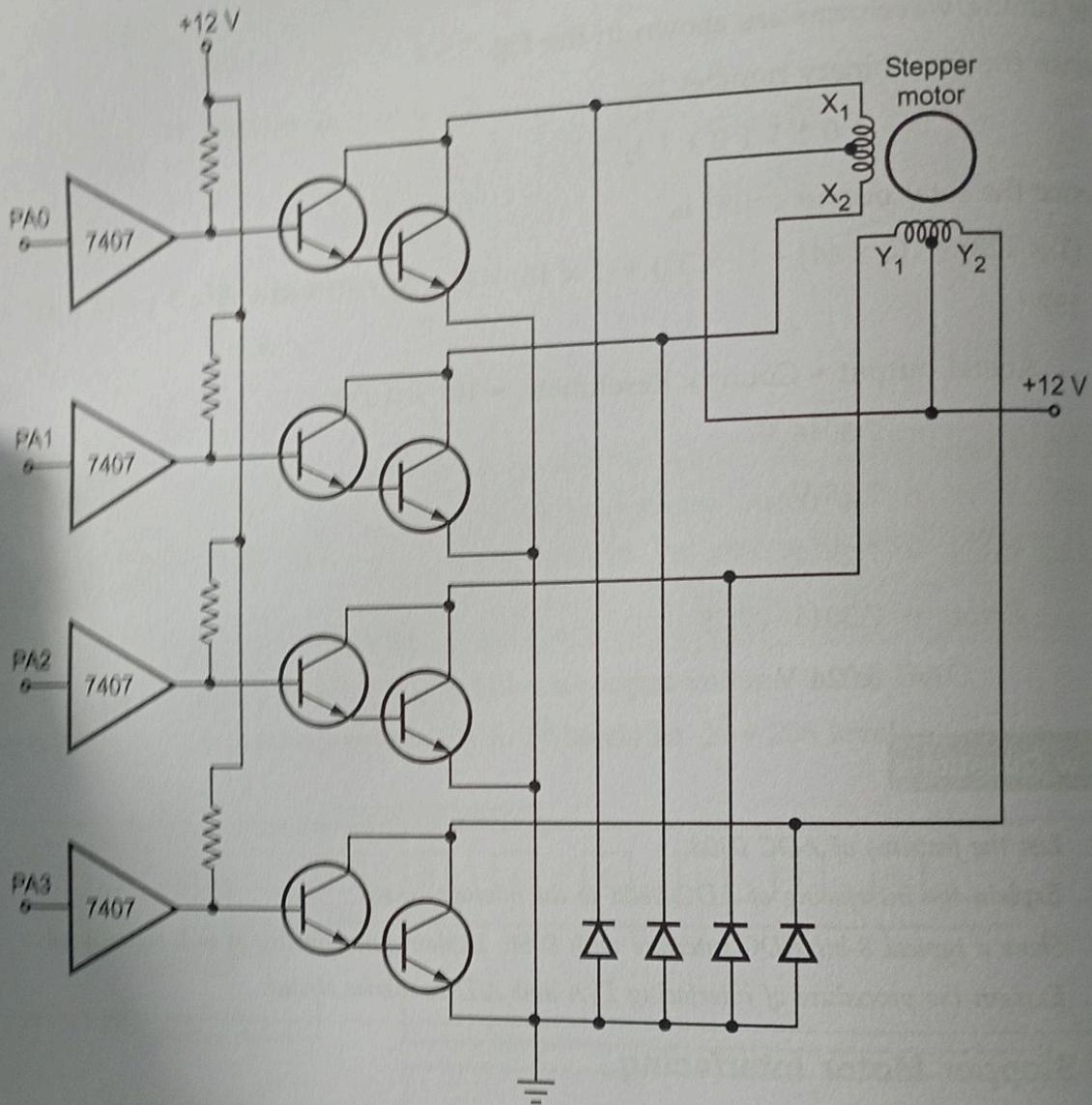


Fig. 3.7.1 Stepper motor interface

- Table 3.7.1 shows typical excitation sequence. The given excitation sequence rotates the motor in clockwise direction. To rotate motor in anticlockwise direction we have to excite motor in a reverse sequence.
- The excitation sequence for stepper motor may change due to change in winding connections. However, it is not desirable to excite both the ends of the same winding simultaneously. This cancels the flux and motor winding may damage. To avoid this, digital locking system must be designed.

Step	X_1	X_2	Y_1	Y_2
1	0	1	0	1
2	1	0	0	1
3	1	0	1	0
4	0	1	1	0
1	0	1	0	1

Table 3.7.1 Full step excitation sequence

- Fig. 3.7.2 shows a simple digital locking system. Only one output is activated (made low) when properly excited ; otherwise output is disabled (made high).
- The excitation sequence given in Table 3.7.1 is called **full step sequence**. In which excitation ends of the phase are changed in one step and motor is rotated by 1.8° . The excitation sequence given in Table 3.7.2 takes two steps to change the excitation ends of the phase. Such a sequence is called **half step sequence** and in each step the motor is rotated by 0.9° .

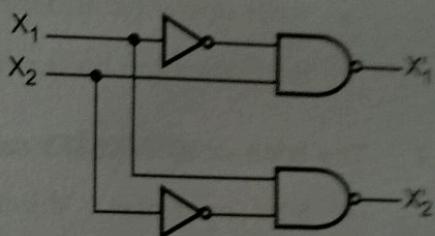


Fig. 3.7.2 Digital locking system

Step	X_1	X_2	Y_1	Y_2
1	0	1	0	1
2	0	0	0	1
3	1	0	0	1
4	1	0	0	0
5	1	0	1	0
6	0	0	1	0
7	0	1	1	0
8	0	1	0	0
1	0	1	0	1

Table 3.7.2 Half step excitation sequence

Review Questions

- Draw and explain the schematic for interfacing a stepper motor using 8255.
- What is full step and half step sequence ?

Unit II Multiple Choice Questions

- Q.1 The operation, ICWR (active low) performs _____.
- a) write operation on input data
 - b) write operation on output data
 - c) read operation on input data
 - d) read operation on output data
- Q.2 The latch or IC 74LS373 can be used as _____.
- a) 8-bit input port
 - b) 16-bit input port
 - c) 8-bit output port
 - d) 16-bit output port
- Q.3 The chip 74LS245 is _____.
- a) 8-bit bidirectional buffer
 - b) 16-bit bidirectional buffer
 - c) 8-bit unidirectional buffer
 - d) 16-bit unidirectional buffer
- Q.4 _____ IC addresses can be generated when direct addressing method is used.
- a) 8
 - b) 16
 - c) 256
 - d) 65536
- Q.5 _____ IC addresses can be generated when indirect addressing method is used.
- a) 8
 - b) 16
 - c) 256
 - d) 65536
- Q.6 In _____ 8086 uses 20 address lines to identify an I/O device.
- a) I/O direct addressing
 - b) I/O indirect addressing
 - c) memory mapped I/O
 - d) none of the above

- Q.7 (WR) signal is ____ when M / (IO) signal is ____ , indicating memory write bus cycle.
a low, low b high, low
c low, high d high, high
- Q.8 In _____ IN, INS, OUT and OUTS instructions are used to transfer data.
a I/O direct addressing b I/O indirect addressing
c memory mapped I/O d both a and b
- Q.9 _____ requires more hardware for decoding.
a I/O direct addressing b I/O indirect addressing
c Memory mapped I/O d Both a and b
- Q.10 The _____ bus controller device decodes the signals to produce the control bus signal.
a internal b data
c external d address
- Q.11 When $(BHE) = \underline{\quad}$ odd bank (Bank 1) is selected and when $A_0 = \underline{\quad}$ even bank (Bank 0) is selected.
a 0, 0 b 0, 1
c 1, 0 d 1, 1
- Q.12 A 20-bit address bus can locate _____.
a 1,048,576 locations b 2,097,152 locations
c 4,194,304 locations d 8,388,608 locations
- Q.13 To access odd addressed word ____ bus cycle(s) is/are required.
a 1 b 2
c 3 d 4
- Q.14 To access even addressed word ____ bus cycle(s) is/are required.
a 1 b 2
c 3 d 4

Q.15 The locations from _____ are used for Interrupt Vector Table (IVT).

- a 00000H to 001FFH
- b 00000H to 003FFH
- c 00000H to 005FFH
- d 00000H to 007FFH

Q.16 _____ needs refreshing after every 2 to 4 ms.

- a ROM
- b RAM
- c DRAM
- d EPROM

Q.17 Port C of 8255 can function independently as _____.

- a input port
- b output port
- c either input or output ports
- d both input and output ports

Q.18 All the functions of the ports of 8255 are achieved by programming the bits of an internal register called _____.

- a data bus control
- b read logic control
- c control word register
- d status register

Q.19 The data bus buffer is controlled by _____.

- a control word register
- b read/write control logic
- c data bus
- d status register

Q.20 The port that is used for the generation of handshake lines in mode 1 or mode 2 is _____.

- a port A
- b port B
- c port C lower
- d port C upper

Q.21 If $A_1=0, A_0=1$ then the input read cycle is performed from _____.

- a port A to data bus
- b port B to data bus
- c port C to data bus
- d CWR to data bus

Q.22 In the I/O mode, the 8255 ports work as _____.

- a reset pins
- b set pins
- c programmable I/O ports
- d only output ports

Q.23 In BSR mode, only port C can be used to _____.

- a set individual ports
- b reset individual ports
- c either set or reset individual port pins
- d programmable I/O ports

Q.24 In which mode do all the ports of the 8255 PPI work as input-output units for data transfer ?

- a BSR mode
- b Mode 0 of I/O mode
- c Mode 1 of I/O mode
- d Mode 2 of I/O mode

Q.25 What is the size of internal bus of the 8255 PPI ?

- a 16 bits
- b 12 bits
- c 8 bits
- d None of the above

Q.26 Which port of the 8255 PPI is capable of performing the handshaking function with the interfaced devices ?

- a Port A
- b Port B
- c Port C
- d All of the above

Q.27 In which of the following modes of the 8255 PPI, only port C is taken into consideration ?

- a BSR mode
- b Mode 0 of I/O mode
- c Mode 1 of I/O mode
- d Mode 2 of I/O mode

Q.28 In mode 2 of I/O mode, which of the following ports are capable of transferring the data in both the directions ?

- a Port A
- b Port B
- c Port C
- d All of the above

Q.29 The resolution of 8 bit DAC/ADC is ____.

- a 64
- b 128
- c 256
- d 512

- Q.30 _____ is a device used to obtain an accurate position control of rotating shafts.
- a DC motor b AC motor
 c Stepper motor d All of the above

Answer Keys for Multiple Choice Questions :

Q.1	b	Q.2	c	Q.3	a	Q.4	e
Q.5	d	Q.6	c	Q.7	c	Q.8	d
Q.9	c	Q.10	c	Q.11	a	Q.12	s
Q.13	b	Q.14	a	Q.15	b	Q.16	e
Q.17	c	Q.18	c	Q.19	b	Q.20	b
Q.21	b	Q.22	c	Q.23	c	Q.24	b
Q.25	c	Q.26	c	Q.27	a	Q.28	s
Q.29	c	Q.30	c				