MODULE 3

Memory and Peripherals Interfacing

1.1 Memory Interfacing –RAM and ROM Decoding Techniques –Partial and Absolute.

1.2 8255-PPI-Block diagram, CWR, Operations modes, interfacing with 8086.

1.3 8257-DMAC Block diagram, DMA operations and transfer modes.

1.4 Programmable Interrupt Controller 8259-Block Diagram, Interfacing the 8259 in single and cascaded mode.

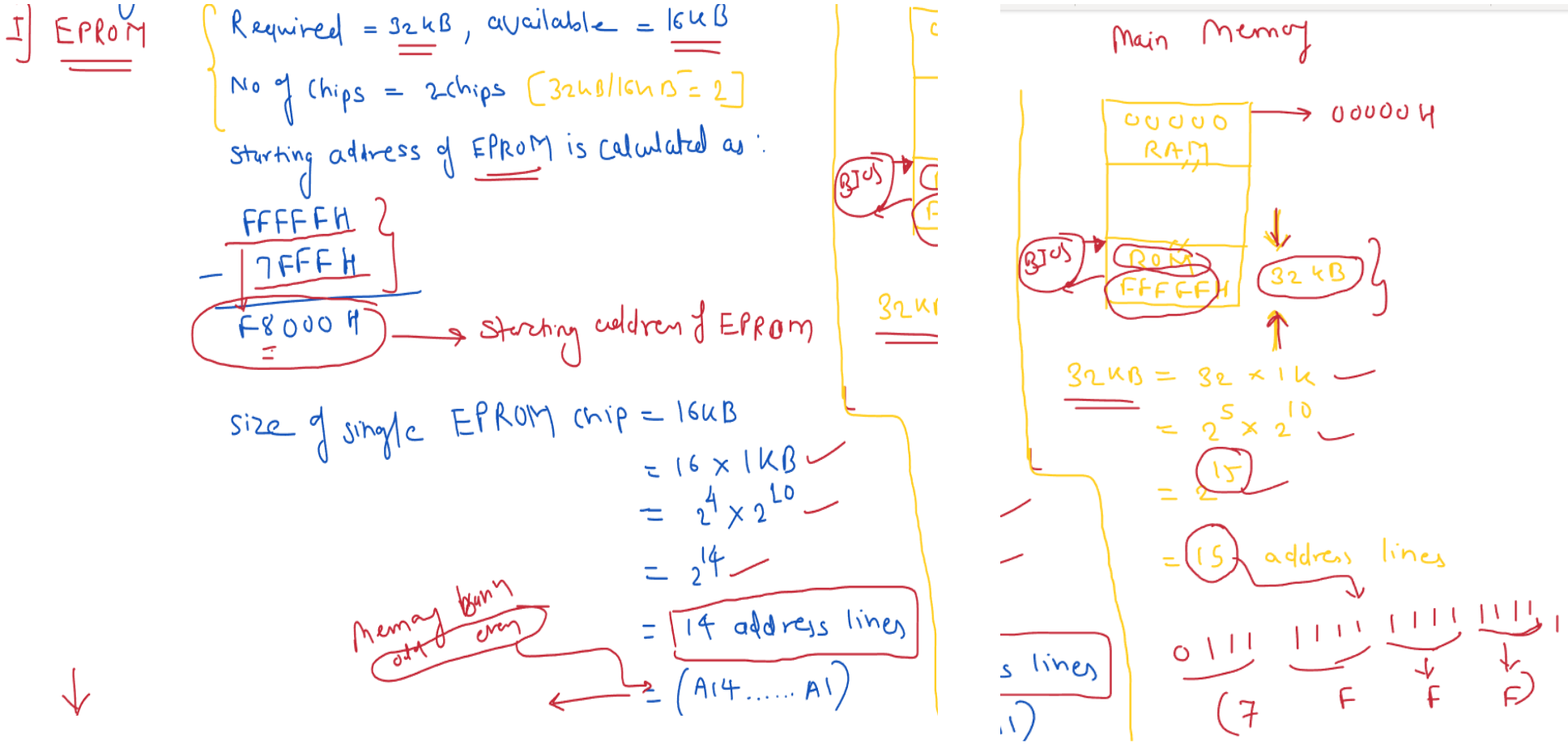
MEMORY INTERFACING

Memory Design Problem:

1. Design an 8086 based Maximum Mode system working at 6 MHz having the following:

- 32KB EPROM using 16KB chips, 128KB RAM using 32KB chips .

Calculation



calculation

E] RAM :- Required = 128kB , Available = 32kB

No of chips = 4 chips * (128kB / 32kB) = 4

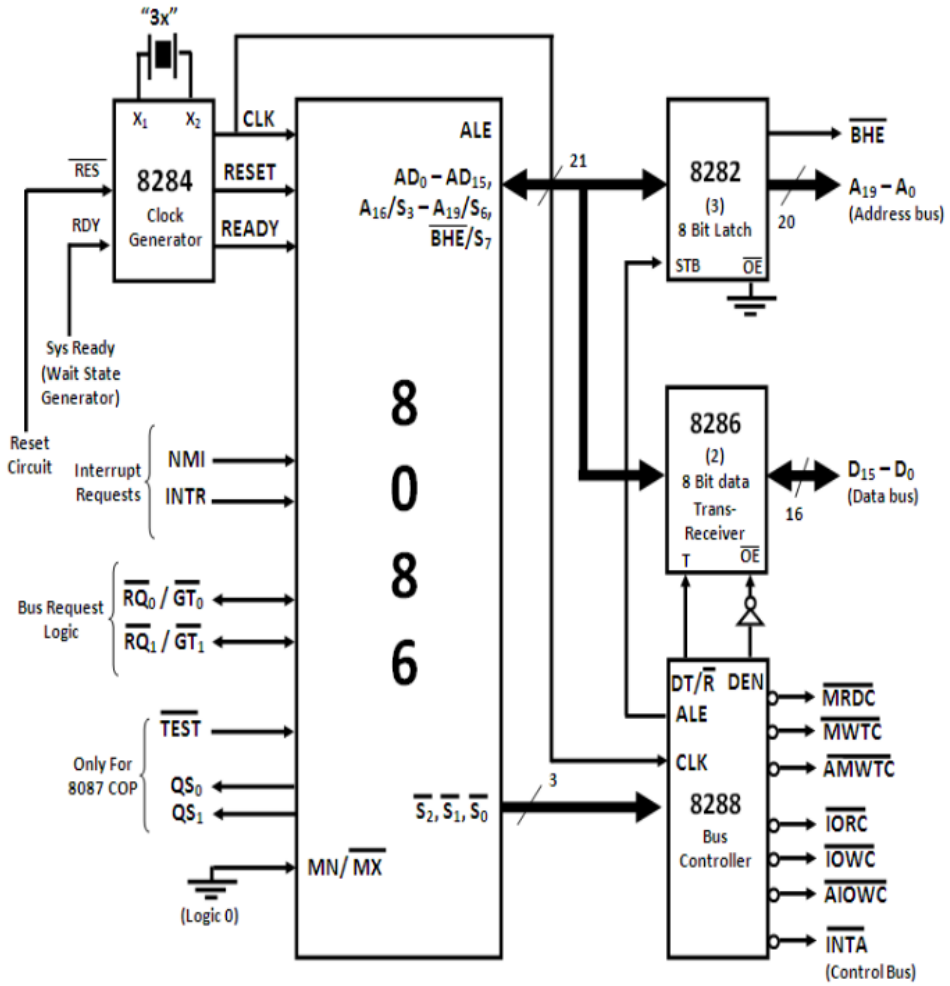
starting address of RAM is 00000H

Size of single RAM chip = 32kB

$= 32 \times 1kB = 2^5 \times 2^{10} = 2^{15}$ = 15 address lines

$= (A15 \ldots A1)$

MEMORY INTERFACING

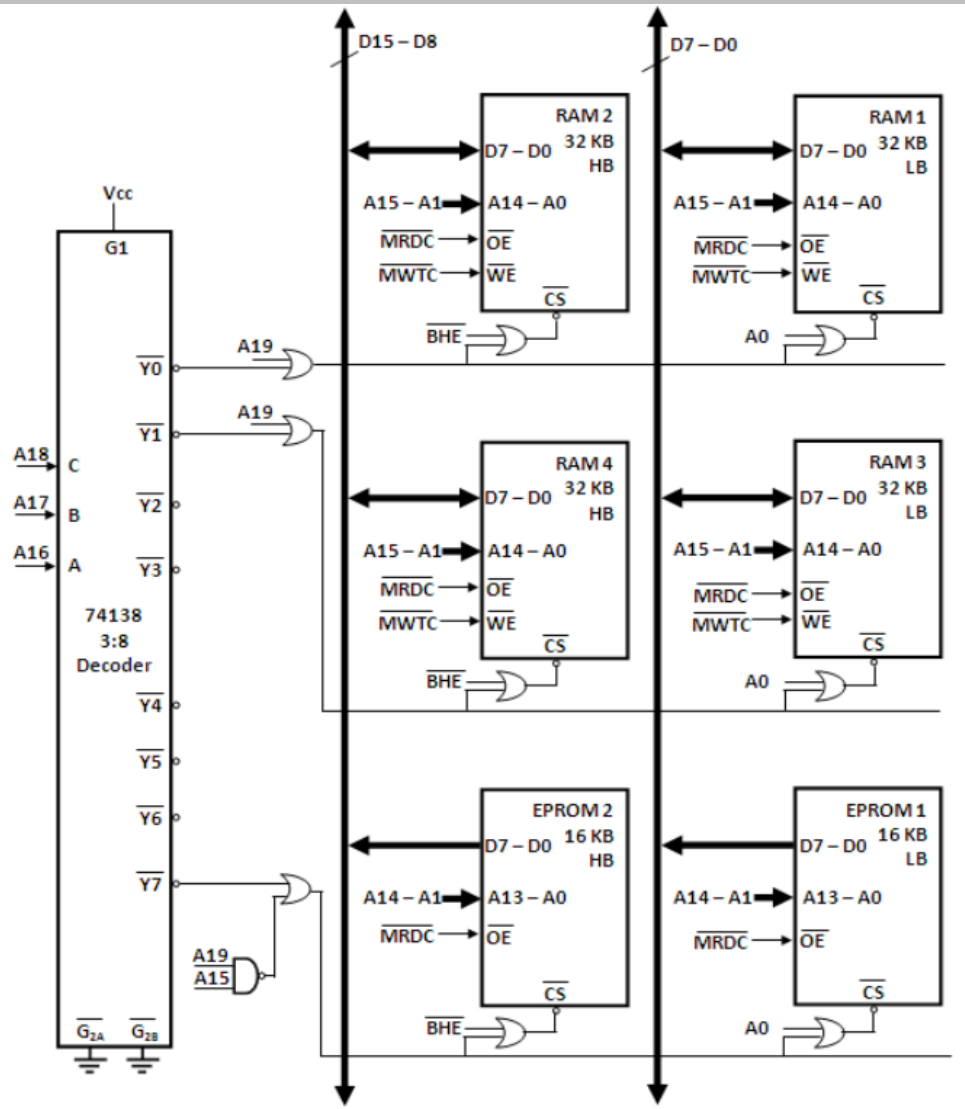


MEMORY INTERFACING

MEMORY MAP

Memory Chip	Address Bus																				Memory Address
	A19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	A0	
RAM 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000H
(LB)	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0FFFEH
RAM 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	00001H
(HB)	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0FFFFH
RAM 3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10000H
(LB)	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1FFFEH
RAM 4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	10001H
(HB)	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFFFH
EPORM 1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F8000H
(LB)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	FFFFEH
EPORM 2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	F8001H
(HB)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFFH

MEMORY INTERFACING



MEMORY INTERFACING

2. Design 8086 based minimum mode system working at 6MHZ having the following :

- 128KB EPROM using 32KB chips, 128 KB RAM using 64KB chips

8257 DMA CONTROLLER

Steps for performing a DMA trasfer :

1. µp initializes the DMAC
   - This is done by giving the starting address and the number of bytes to be transferred.
2. I/O device requests the DMAC
   - I/O device makes the DREQ signal = 1.
3. DMAC requests the uP for control of the system bus
   - DMAC makes HOLD = 1
4. Microprocessor releases control of the system bus
   - µp finishes the current machine cycle and releases control of the system bus.
   - µp informs the DMAC that the bus is released by making HLDA = 1. Now up enters HOLD state.
5. DMAC becomes the bus master.
   - On getting HLDA from uP, DMAC becomes the bus master.
   - It informs the I/O device that DMA transfer is about to begin by activating the DACK signal
6. DMA Transfer begins
   - DMAC transfers data between memory and I/O, one byte in one cycle.
   - After every cycle, the Address Register is incremented and the Count Register is decremented
   - This continues till the Count reaches zero (Terminal Count). Now the DMA transfer is completed.
7. 7. DMA Transfer ends
   - DMAC releases control of the system bus. & It makes HOLD = 0.
   - This makes up come out of Hold state and once again become the bus master. µP takes control of the system bus and continues its operation.

8257 DMA CONTROLLER

Advantage of DMA:

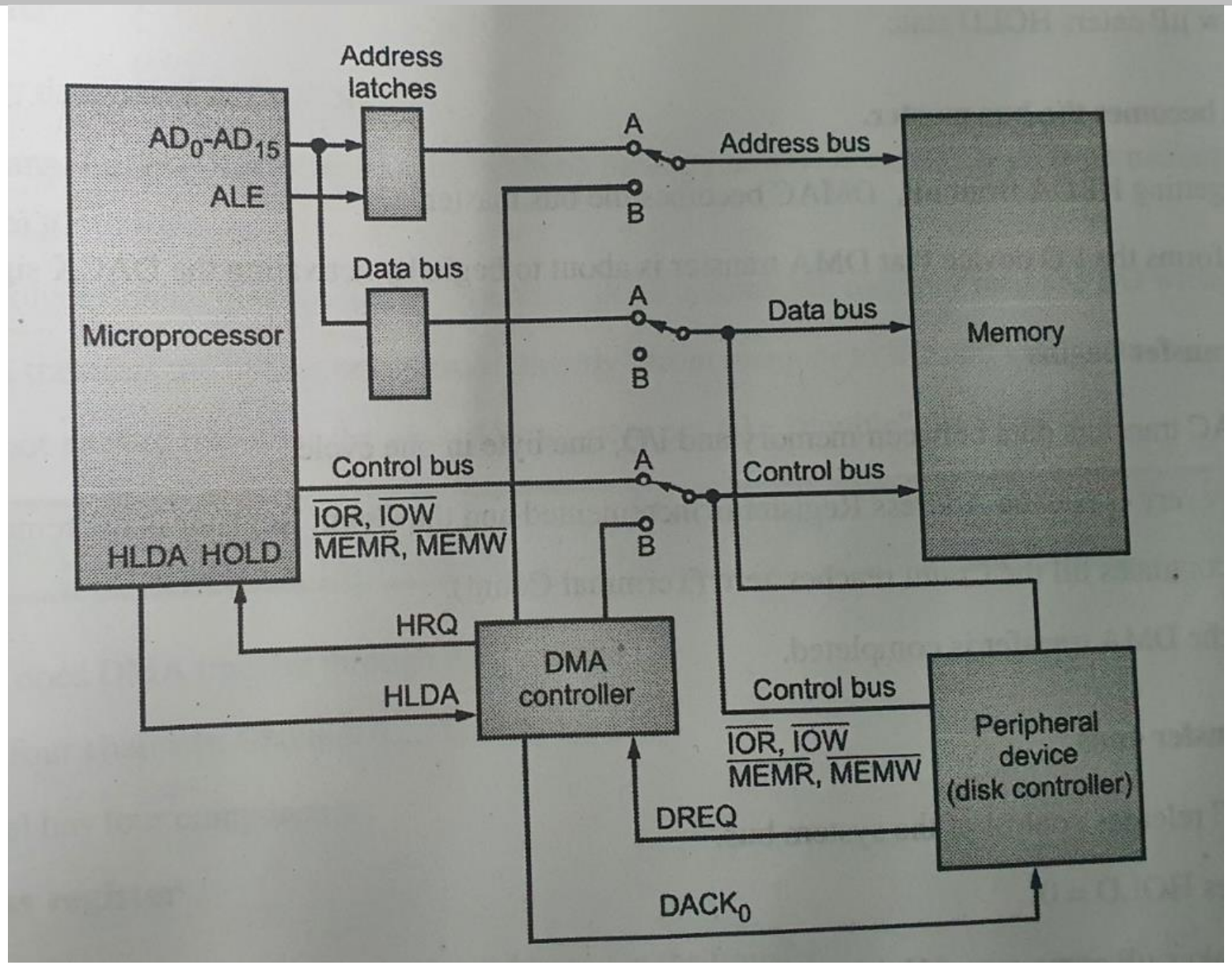
1. Hardware based : massive speed
2. Direct transfer : without involvement of processor

DMA Channels :

- DMAC does DMA transfer through its channels.
- DMAC has four channels : channel 0…….channel 3
- Each channel has four components:
  - Address register
  - Count register
  - DERQ
  - DACK

8257 DMA CONTROLLER

Concept of DMA controller :



8257 DMA CONTROLLER

Pin diagram :

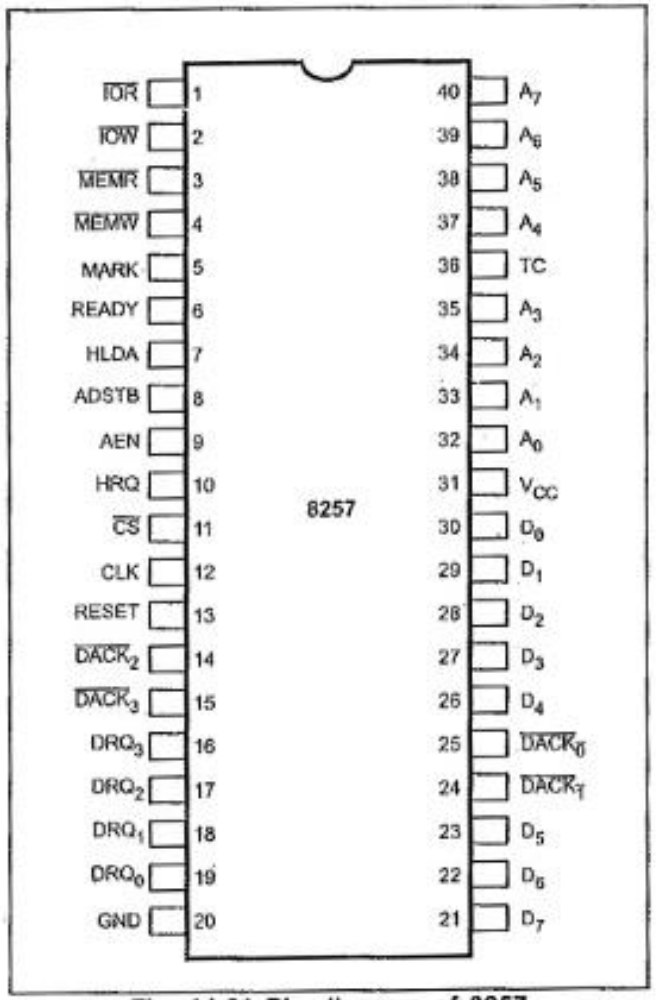
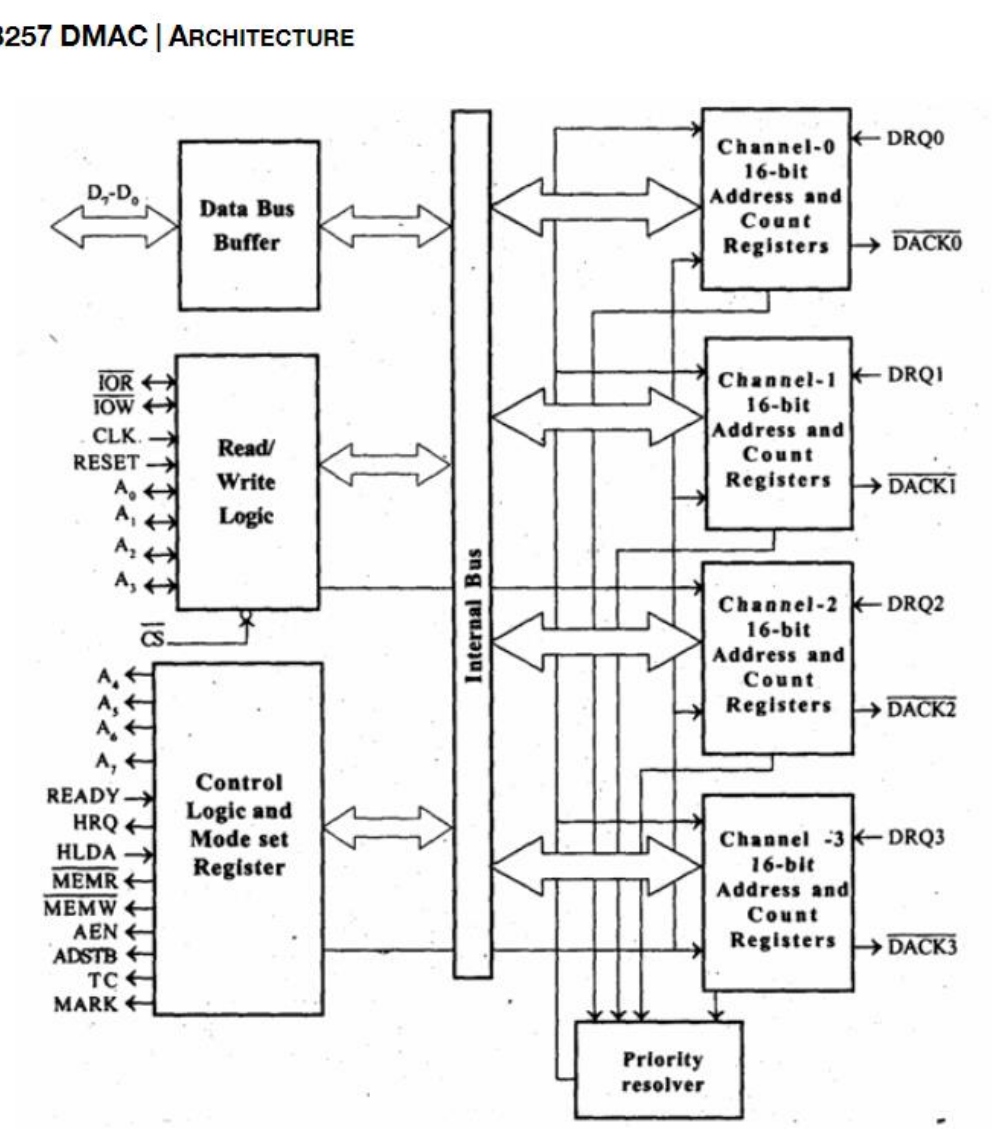


Fig. 14.61 Pin diagram of 8257

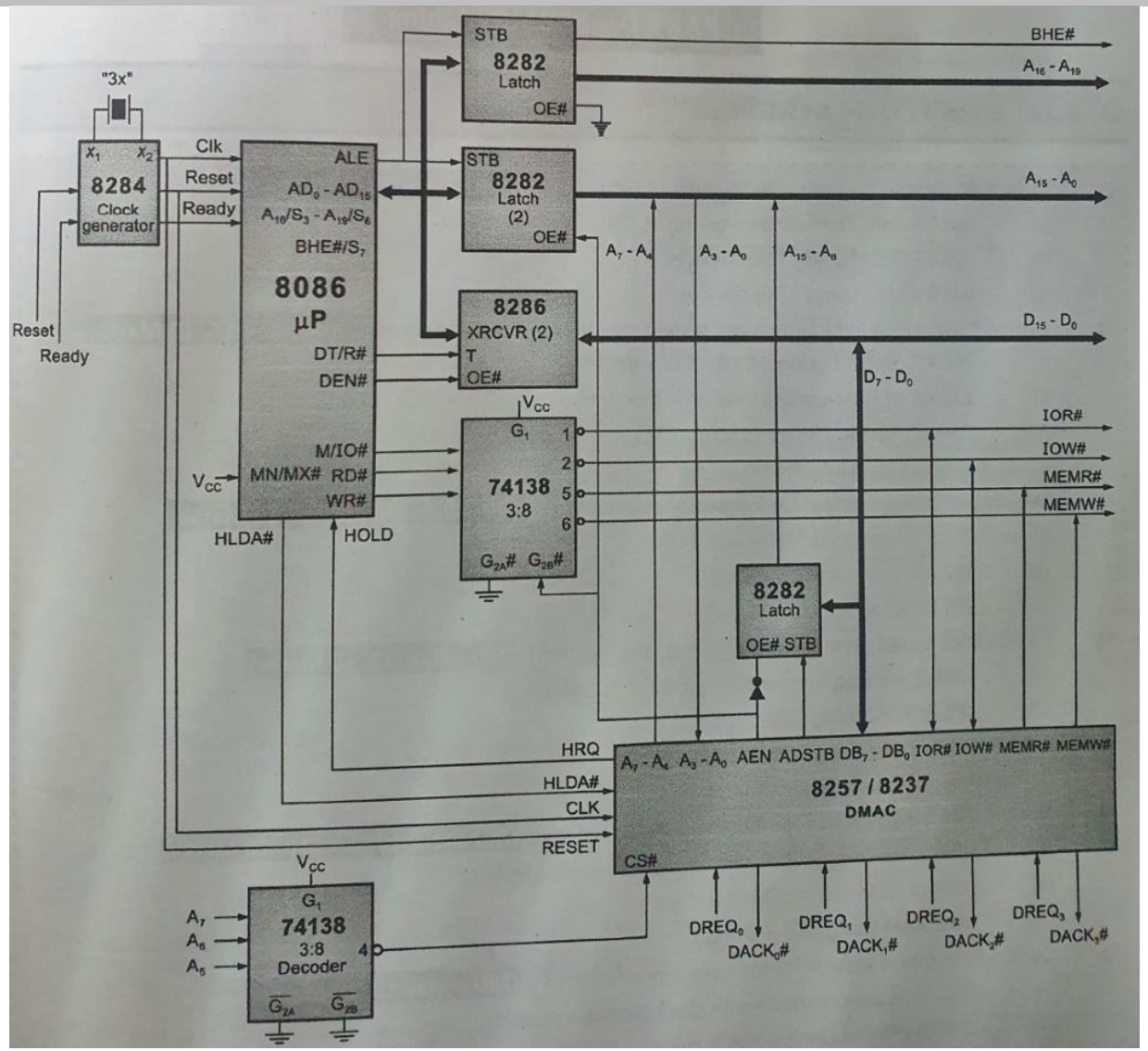
8257 DMA CONTROLLER

8257 DMAC ARCHITECTURE :

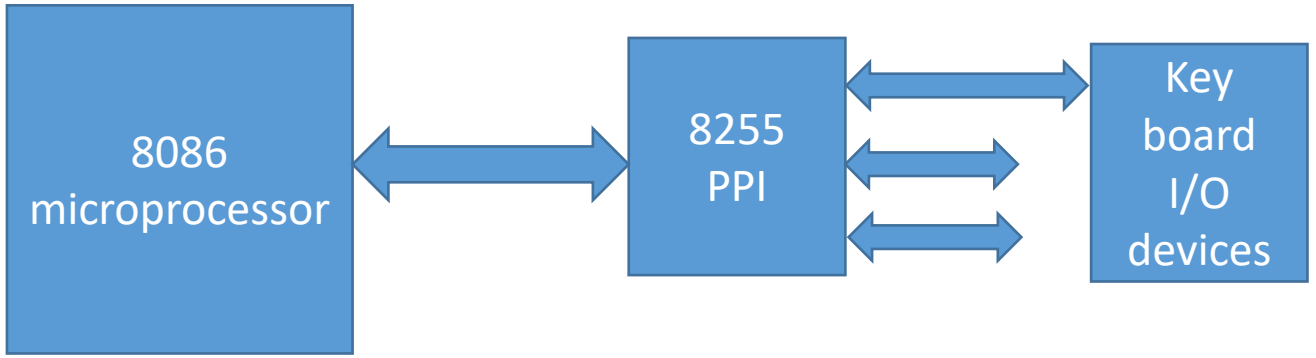


8257 DMA CONTROLLER

Interfacing with 8086 :



8255 PROGRAMMABLE PERIPHERAL INTERFACE



- I/O devices cant connect directly to µp (as transfer is very risky)
- I/O device connected through 8255 ports

8255 PROGRAMMABLE PERIPHERAL INTERFACE

Introduction :

1. It is used to interface (connect) up with I/O devices such as a printer, keyboard etc.
2. 8255 has three 8 bit bidirectional I/O ports called Port A, Port B and Port C
3. They can be used as input ports or as output ports.
4. These ports can operate in three different modes of data transfer.
5. Mode 0 is called Simple I/O. Here the three ports perform basic 8 bit data transfers.
6. Mode 1 is called Handshake I/O. Here port A and Port B transfer data using an advanced technique called handshaking. This prevents any loss of data and hence makes the transfer more reliable. To carry out handshaking, lines of Port C are used up by Port A and Port B.
7. Mode 2 is bidirectional handshaking in which as the name suggests, the port can perform input as well as output handshaking.
8. The modes are selected by the programmer by giving commands to 8255. Depending upon the system requirement, the appropriate command is formed and is first stored in a register like AL. Then using an instruction like OUT, the command is sent to 8255 through the data bus.
9. Additionally, 8255 has an excellent feature called BSR command. It stands for Bit Set Reset. Using BSR command, the programmer can individually set or reset (send 1 or 0) to any line of Port C without affecting other lines. This basically transforms Port C from being one 8 bit port to becoming eight 1 bit ports that can be individually controlled by the programmer. This feature is very useful in complex interfaces like ADC. LCD controller etc

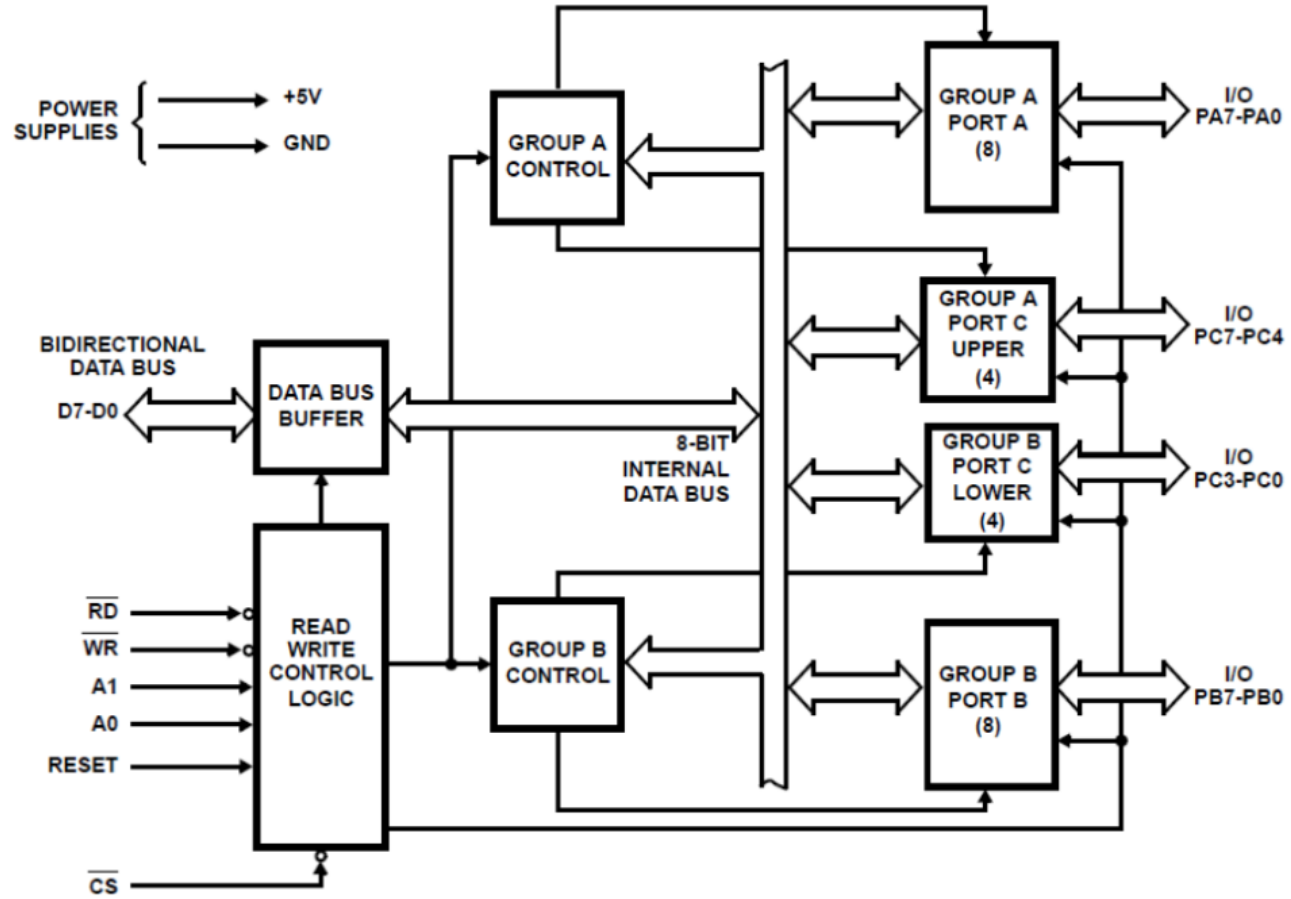
8255 PROGRAMMABLE PERIPHERAL INTERFACE

Features :

1. 8255 is a programmable peripheral interface / general-purpose I/O device.
2. It is used to connect microprocessor with I/O devices such printer, keyboard
3. It has 3 8-bit bi-directional I/O . ports: Port A, Port B, and Port C.
4. It provides 3 modes of data transfer: Simple I/O, Handshake I/O 4 and Bi-directional Handshake.
5. It also provides a Bit Set Reset Modes to alter individual bits of Port C.

8255 PROGRAMMABLE PERIPHERAL INTERFACE

ARCHITECTURE OF 8255



8255 PROGRAMMABLE PERIPHERAL INTERFACE

Main components of 8255 architecture:

- Data bus buffer
- Read/Write control logic
- Group A control
- Group B control
- Port A, Port B Port C

8255 PROGRAMMABLE PERIPHERAL INTERFACE

- Data bus buffer

➢ This is a 8-bit bi-directional buffer used to interface the internal data bus of 8255 with the external (system) data bus.

➢ The CPU transfers data to and from the 8255 through the data bus via this buffer.

➢ The commands given to 8255 (I/O command and BSR command) are also given through the same datu bus.

8255 PROGRAMMABLE PERIPHERAL INTERFACE

- Read/Write control logic :

➢ It accepts address and control signals from the µP.

➢ The Control signals determine whether it is a read or a write operation.

➢ During a read, data will be transferred from 8255 to the µp.

➢ During a write, data will be transferred from µp to 8255.

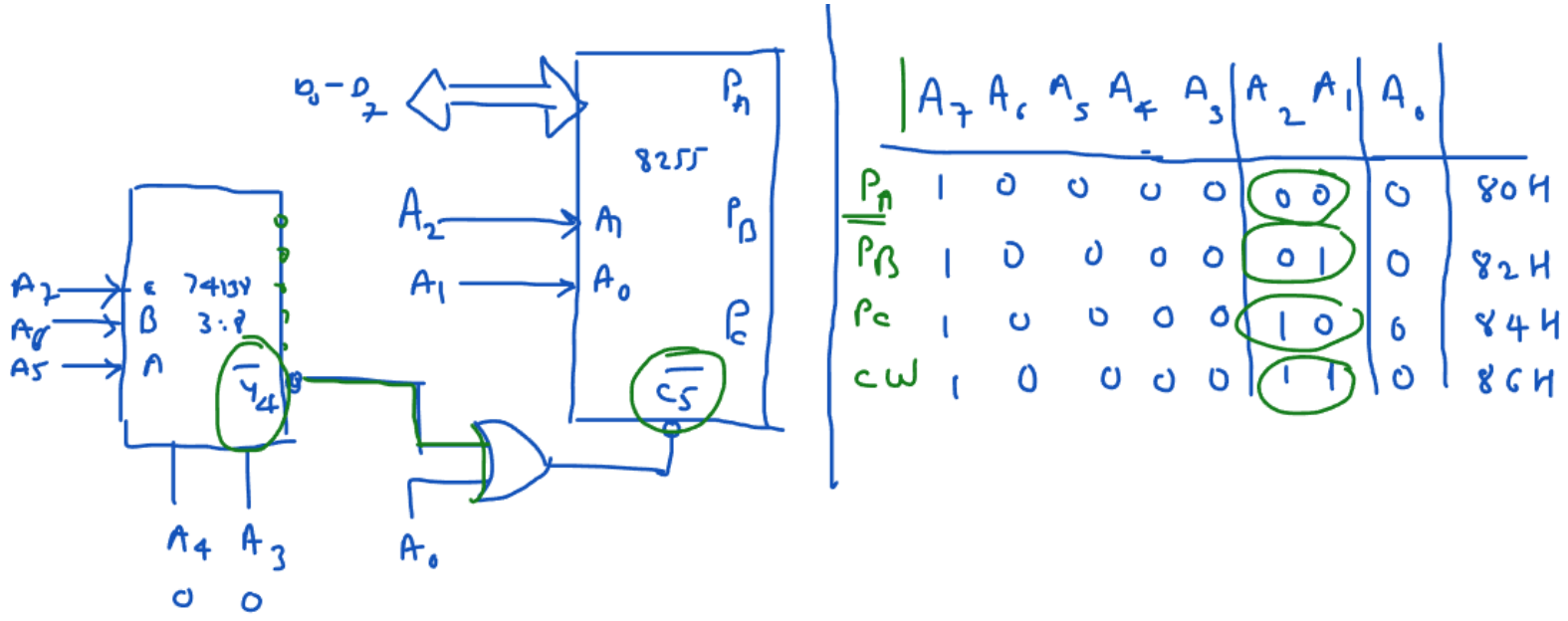
➢ There is a chip selection signal that selects 8255 on the basis of its address.

➢ The reset signal is to reset 8255 and stop all current transfers.

➢ Finally there are two address lines A1 and A0 to which we connect the address lines A2 and A1 of the µP 8086. These lines are uses to make internal selection within 8255.

8255 PROGRAMMABLE PERIPHERAL INTERFACE

For 8255 A_1 A_0	For 8086 A_2 A_1	Selection	Sample address
0 0	0 0	Port A	80 H (i.e. 1000 0**00**0)
0 1	0 1	Port B	82 H (i.e. 1000 0**01**0)
1 0	1 0	Port C	84 H (i.e. 1000 0**10**0)
1 1	1 1	Control Word	86 H (i.e. 1000 0**11**0)



8255 PROGRAMMABLE PERIPHERAL INTERFACE

Group A control :

- This Control block controls Port A and Port CUpper i.e. PC7-PC4.
- It accepts Control signals from the Control Word and forwards them to the respective Ports.

Group B Control :

- This Control block controls Port B and Port CLower i.e. PC3-PCO..
- It accepts Control signals from the Control Word and forwards them to the respective Ports.

8255 PROGRAMMABLE PERIPHERAL INTERFACE

Port A, Port B, Port C :

These are 8-bit Bi-directional PortsThey can be programmed to wok in the various modes as follows:

Port	Mode 0	Mode 1	Mode 2
Port A	Yes	Yes	Yes
Port B	Yes	Yes	**No** *(Mode 0 or Mode 1)*
Port C	Yes	**No** *(Handshake signals)*	**No** *(Handshake signals)*

ONLY Port C can also be programmed to work in Bit Set reset Mode to manipulate its individual bits.

8255 PROGRAMMABLE PERIPHERAL INTERFACE

Group A control :

- This Control block controls Port A and Port CUpper i.e. PC7-PC4.
- It accepts Control signals from the Control Word and forwards them to the respective Ports.

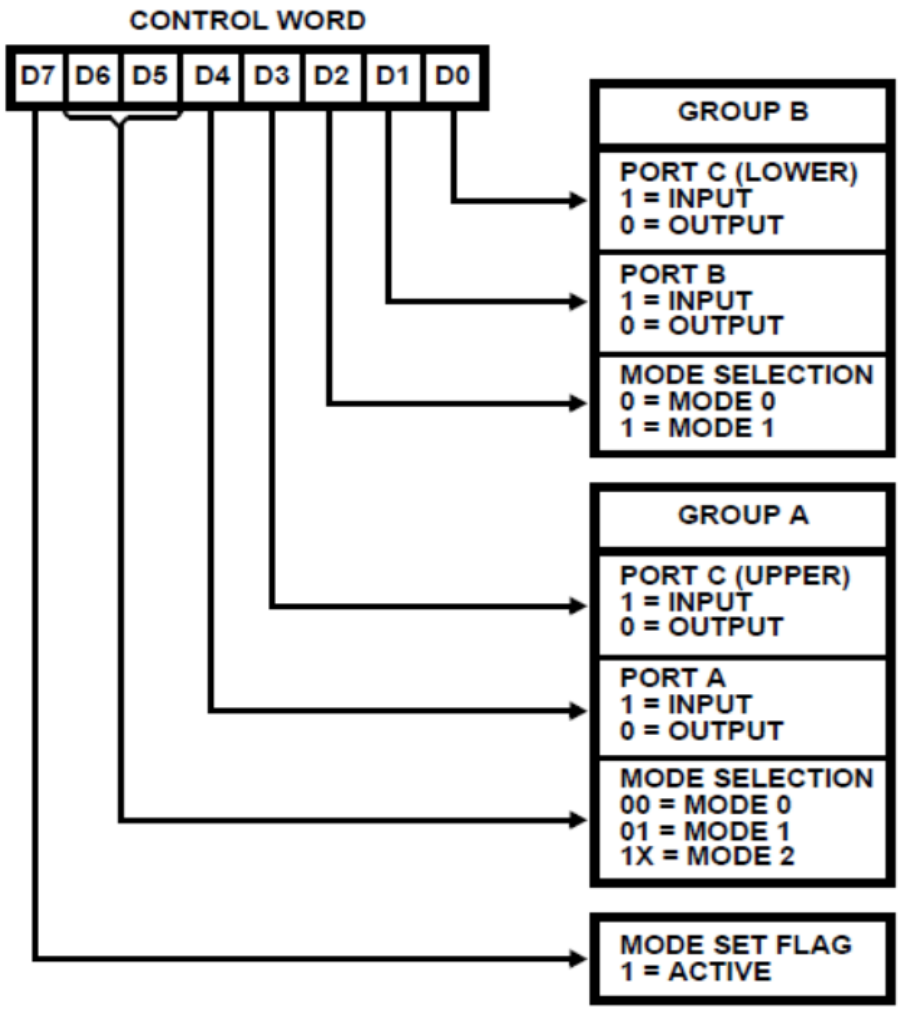
Group B Control :

- This Control block controls Port B and Port CLower i.e. PC3-PCO..
- It accepts Control signals from the Control Word and forwards them to the respective Ports.

8255 PROGRAMMABLE PERIPHERAL INTERFACE

I/O mode control word:

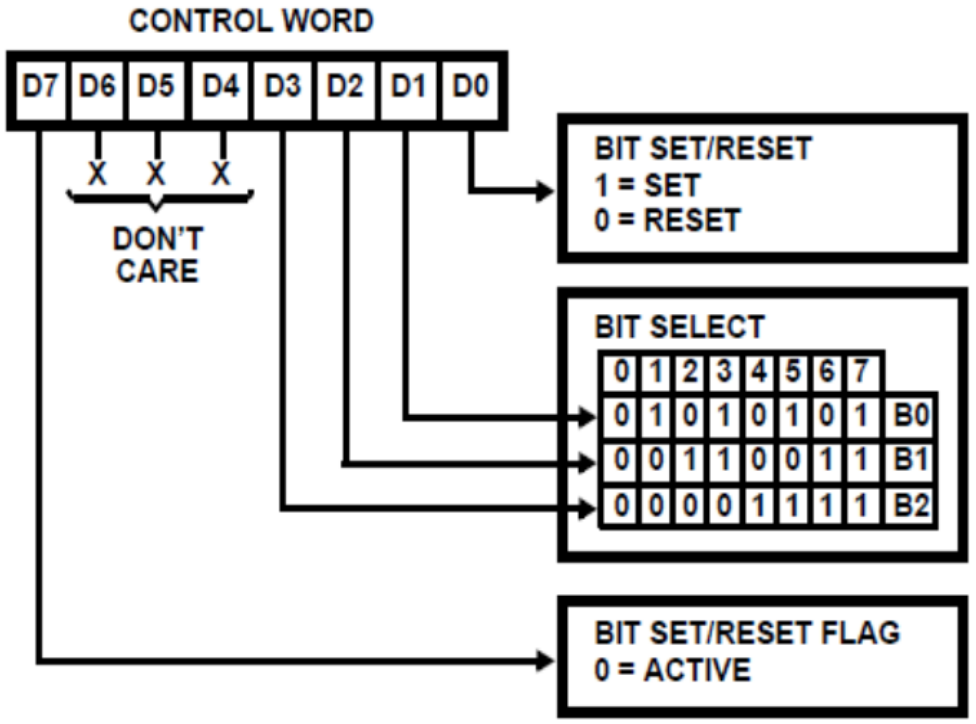
- To do 8-bit data transfer using the Ports A, B or C, 8255 needs to be in the IO mode.
- The bit pattern for the control word in the IO mode is as follows:



8255 PROGRAMMABLE PERIPHERAL INTERFACE

BSR mode control word:

- The BSR Mode is used ONLY for Port C.
- In In this Mode the individual bits of Port C can be set or reset.
- This is very useful as it provides 8 individually controllable lines which can be used while interfacing with devices like A to D Converter or a 7-segment display etc..
- The individual bit is selected and Set/reset through the control word.
- Since the D7 bit of the Control Word is 0, the BSR operation will not affect the modes



8255 PROGRAMMABLE PERIPHERAL INTERFACE

Data Transfer Modes:

1. Mode 0 (Simple Bi-directional I/O):

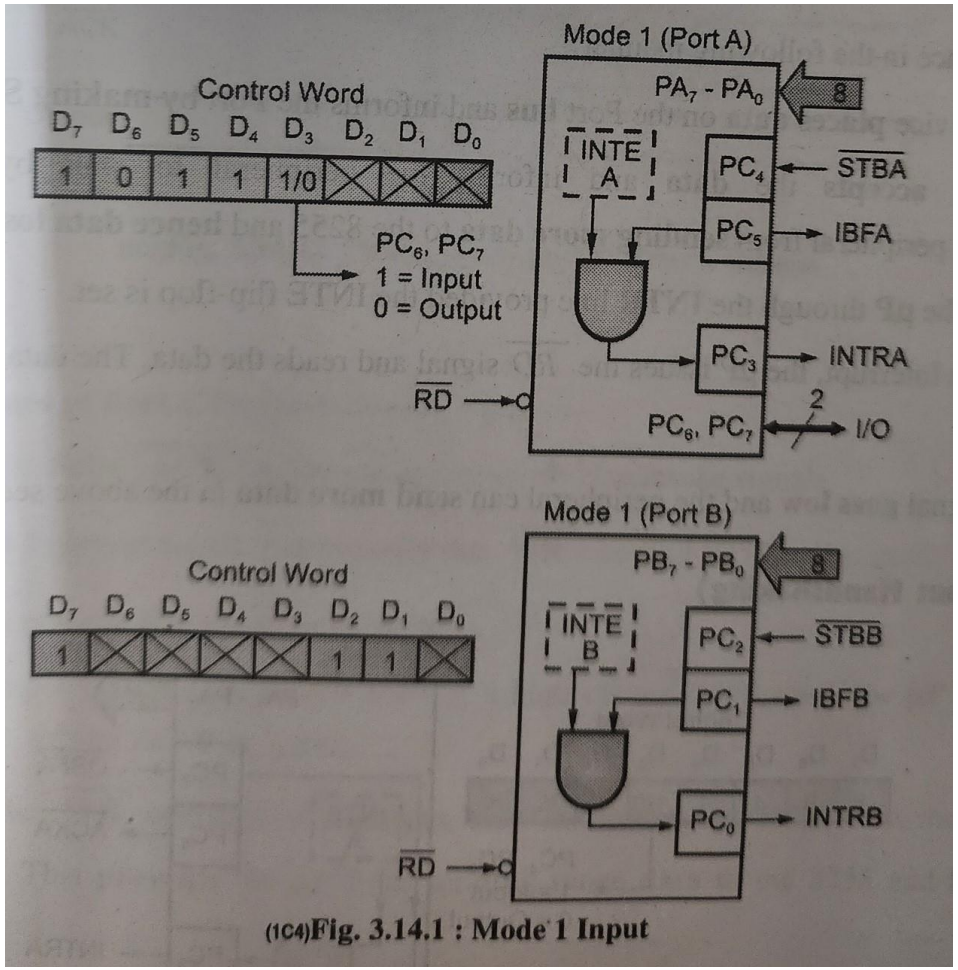
- Port A and Port B used as 2 Simple 8-bit I/O Ports,
- Port C is used as 2 simple 4-bit I/O Ports.
- Each port can be programmed as input or output individually.
- Ports do not have handshake or interrupting capability.
- Hence, slower devices cannot be interfaced.

2. Mode 1 : (Handshake I/O)

- Mode 1 (Handshake 1/0)In Mode 1, handshake signals are exchanged between the devices before the data transfer takes place.
- Port A and Port B used as 2 8-bit I/O Ports that can programmed in Input OR in output mode.
- Each Port uses 3 lines from Port C for handshake. The remaining lines of Port C can be used for simple IO.
- Interrupt driven data transfer and Status driven data transfer possible.
- Hence, slower devices can be interfaced
- The handshake signals are different for input and output modes:

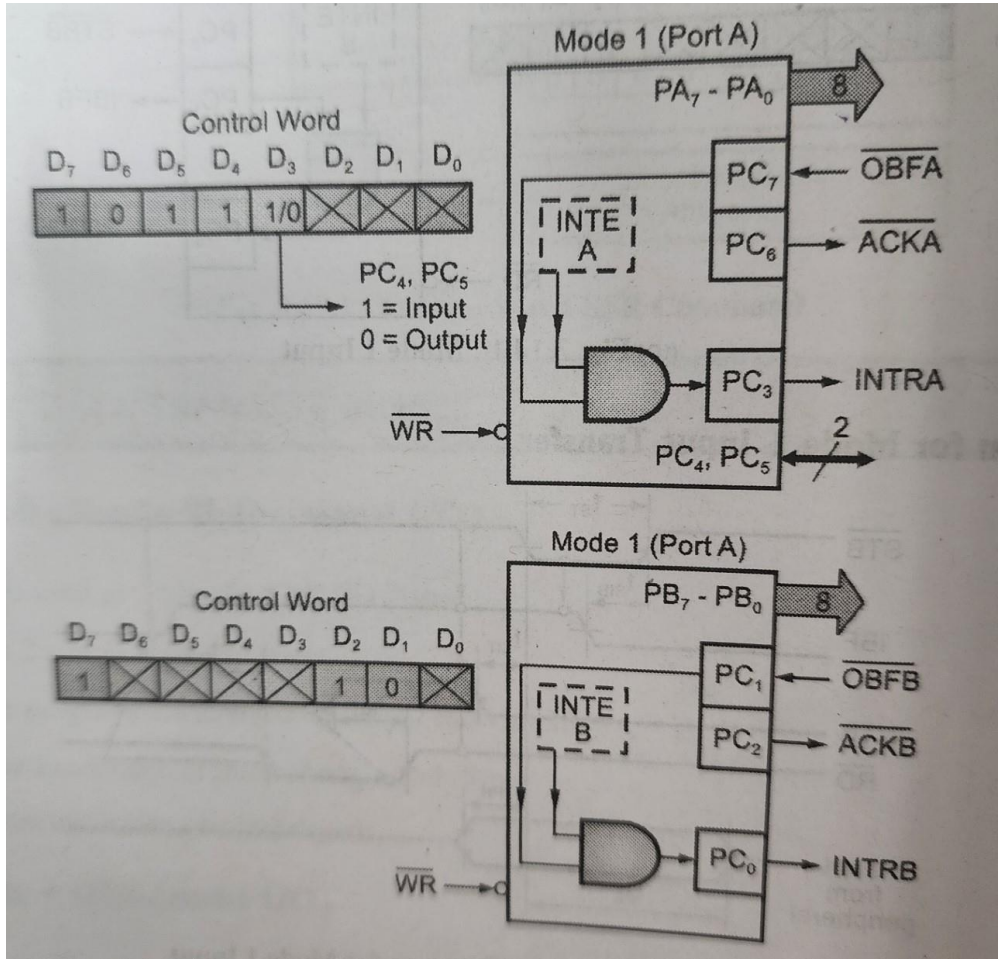
8255 PROGRAMMABLE PERIPHERAL INTERFACE

Mode 1 Input handshaking



(1C4)Fig. 3.14.1 : Mode 1 Input

Mode 1 output handshaking



8255 PROGRAMMABLE PERIPHERAL INTERFACE

For Input:

STB and IBF handshaking signals, INTR→ Interrupt signal

For Output:

OBF and ACK handshaking signals, INTR→ Interrupt signal.

Thus the 5 signals used from Port C are:

STB, IBF, INTR, OBF and ACK. :

8255 PROGRAMMABLE PERIPHERAL INTERFACE

- Working
  - Each port uses 3 lines of Port C for the following signals:
  - STB (Strobe), IBF (Input Buffer Full) Handhsake signals
  - INTR (interrupt) → Interrupt signal
  - Additionally the RD signal of 8255 is also used

Handshaking takes place in the following manner.

(1) The peripheral device places data on the Port bus and informs the Port by making STB low.

(2) The input Port accepts the data and informs the peripheral to wait by making IBF high. This prevents the peripheral from sending more data to the 8255 and hence data loss is prevented.

(3) 8255 interrupts the uP through the INTR line provided the INTE flip-flop is set.

(4) In response to the Interrupt, the uP issues the RD signal and reads the data. The data byte is thus transferred to the up

(5) Now, the IBF signal goes low and the peripheral can send more data in the above sequence.

8255 PROGRAMMABLE PERIPHERAL INTERFACE

- Working
  - Each port uses 3 lines of Port C for the following signals:
  - OBF (Output Buffer Full). ACK (Acknowledgement)→ Handhsake signals
  - INTR (interrupt) Interrupt signal. Additionally the WR signal of 8255 is also used.

Handshaking takes place in the following manner:

(1) When the output port is empty (indicated by a high on the INTR line), the uP writes data on the output port by giving the WR signal

(2) As soon as the WR operation is complete, the 8255 makes the INTR low, indicating that the up should wait. This prevents the uP from sending more data to the 8255 and hence data loss is prevented

(3) 8255 also makes the OBF low to indicate to the output peripheral that data is available on the databus:

(4) The peripheral accepts the data and sends an acknowledgement by making the ACK low. The data byte is thus transferred to the peripheral.

(5) Now, the OBF and ACK lines go high.

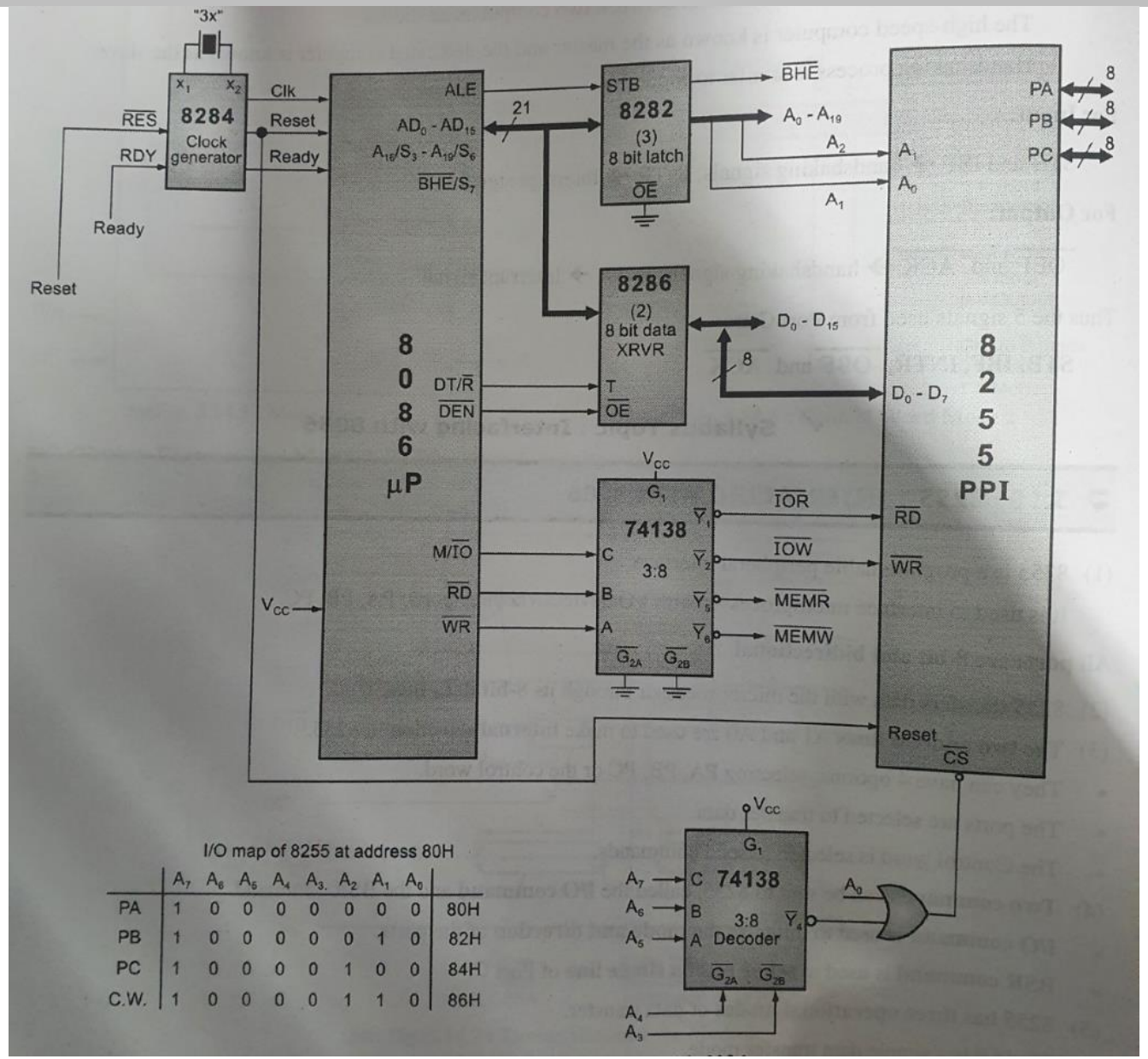
6) The INTR line becomes high to inform the uP that another byte can be sent. i.e. the output port is empty.

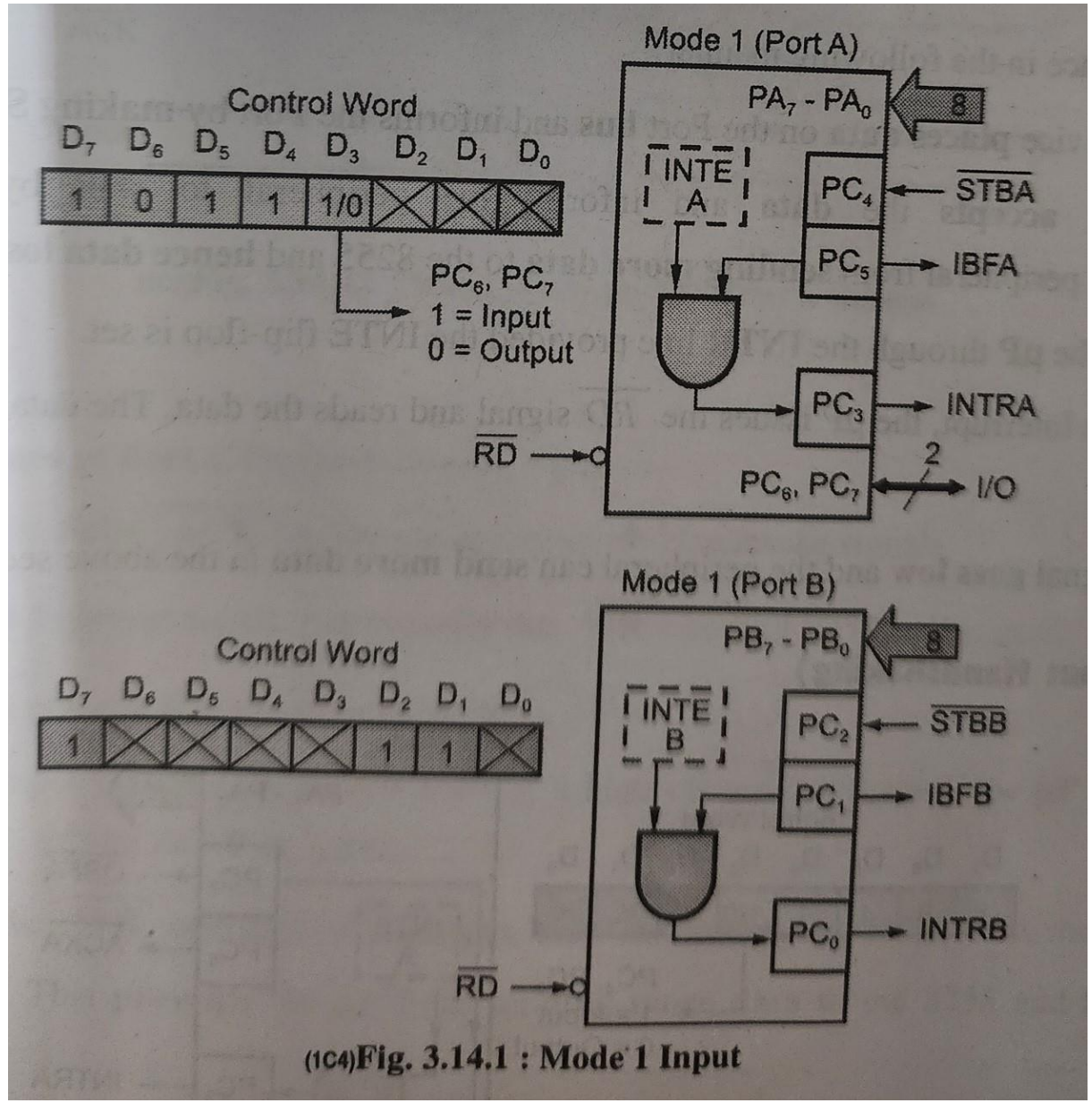
(6) This process is repeated for further bytes.

8255 PROGRAMMABLE PERIPHERAL INTERFACE

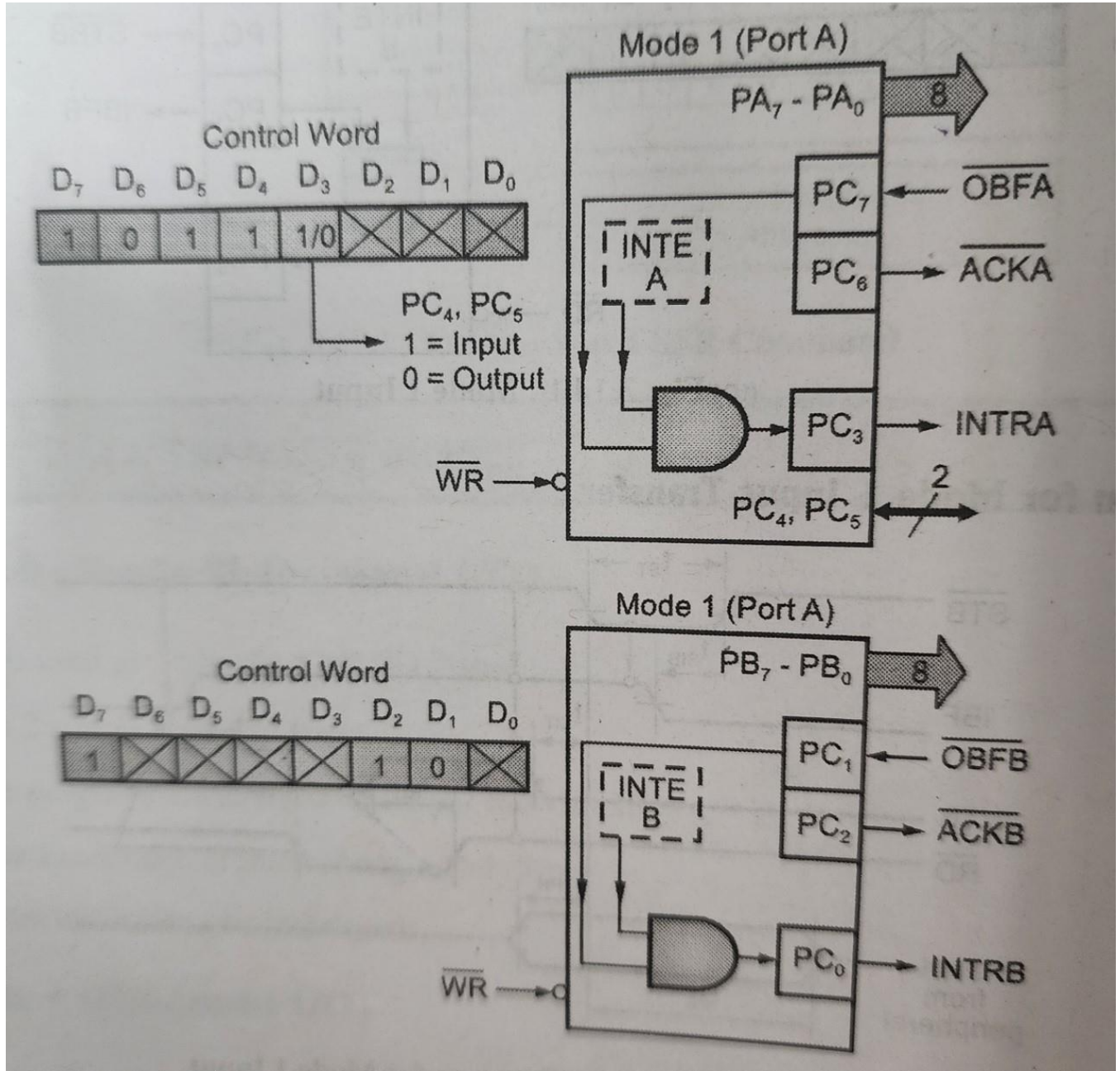
3. Mode 2 : (Bi- directional Handshake I/O)
   - In this mode, Port A is used as an 8-bit bi-directional Handshake I/O Port.
   - Port A requires 5 signals from Port C for doing Bi-directional handshake.
   - Port B has the following two options:
     - Use the remaining 3 lines of Port C for handshaking so that Port B is in Mode 1. Here Port C lines will be completely used for handshaking (5 by Port A and 3 by Port B).
     - Port B works in Mode 0 as simple I/O.
       - In this case the remaining 3 lines of Port C can be used for data transfer.
       - Port A can be used for data transfer between two computers as shown.
       - The high-speed computer is known as the master and the dedicated computer is known as the slave. Handshaking process is similar to Mode 1.For Input:STB and IBF handshaking signals, INTR→ Interrupt signalFor Output:OBF and ACKhandshaking signals, INTR→ Interrupt signal.Thus the 5 signals used from Port Care: STB, IBF, INTR, OBF and ACK.

8255 PROGRAMMABLE PERIPHERAL INTERFACE





(1C4)**Fig. 3.14.1 : Mode 1 Input**


Mode 1 (Port A)
Control Word
D7 D6 D5 D4 D3 D2 D1 D0
1 0 1 1 1/0
PC4, PC5
1 = Input
0 = Output
PA7 - PA0
8
PC7
OBFA
INTE A
PC6
ACKA
PC3
INTRA
WR
2
PC4, PC5
Mode 1 (Port A)
Control Word
D7 D6 D5 D4 D3 D2 D1 D0
1 1 0
PB7 - PB0
8
PC1
OBFB
INTE B
PC2
ACKB
PC0
INTRB
WR