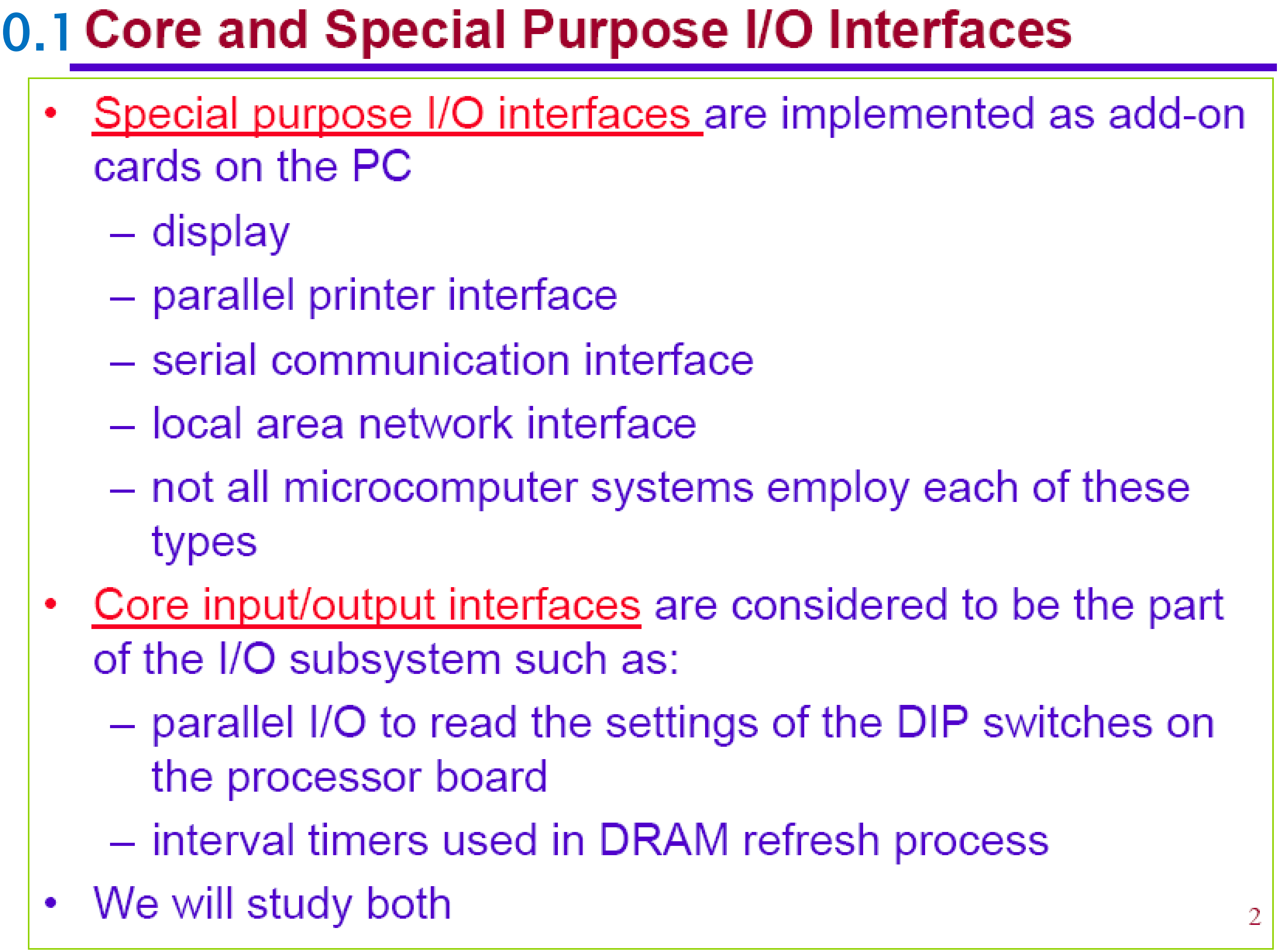
Microprocessor Systems

Chapter 10

Input/Output Interface Circuits and LSI Peripheral Device

Lecture Outline

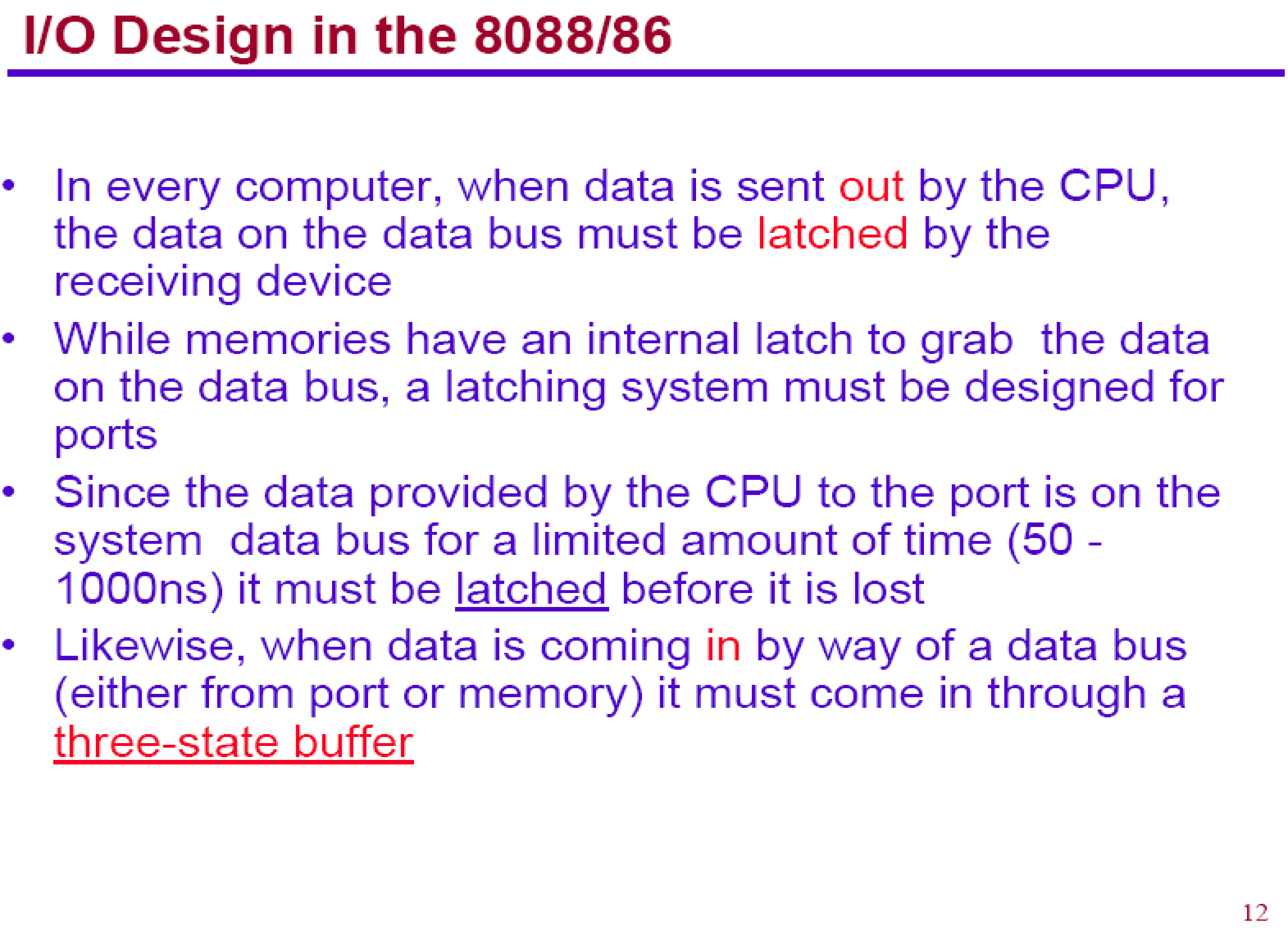
- 10.1 Core and special-purpose I/O interfaces
- 10.2 Byte-Wide output ports using isolated I/O
- 10.3 Byte-Wide input ports using isolated I/O
- 10.4 Input/Output handshaking and parallel printer interface
- 10.5 the 8255 Programmable Peripheral Interface



10.1 Core and Special Purpose I/O Interfaces

- Special purpose I/O interfaces are implemented as add-on cards on the PC
  - display
  - parallel printer interface
  - serial communication interface
  - local area network interface
  - not all microcomputer systems employ each of these types
- Core input/output interfaces are considered to be the part of the I/O subsystem such as:
  - parallel I/O to read the settings of the DIP switches on the processor board
  - interval timers used in DRAM refresh process
- We will study both

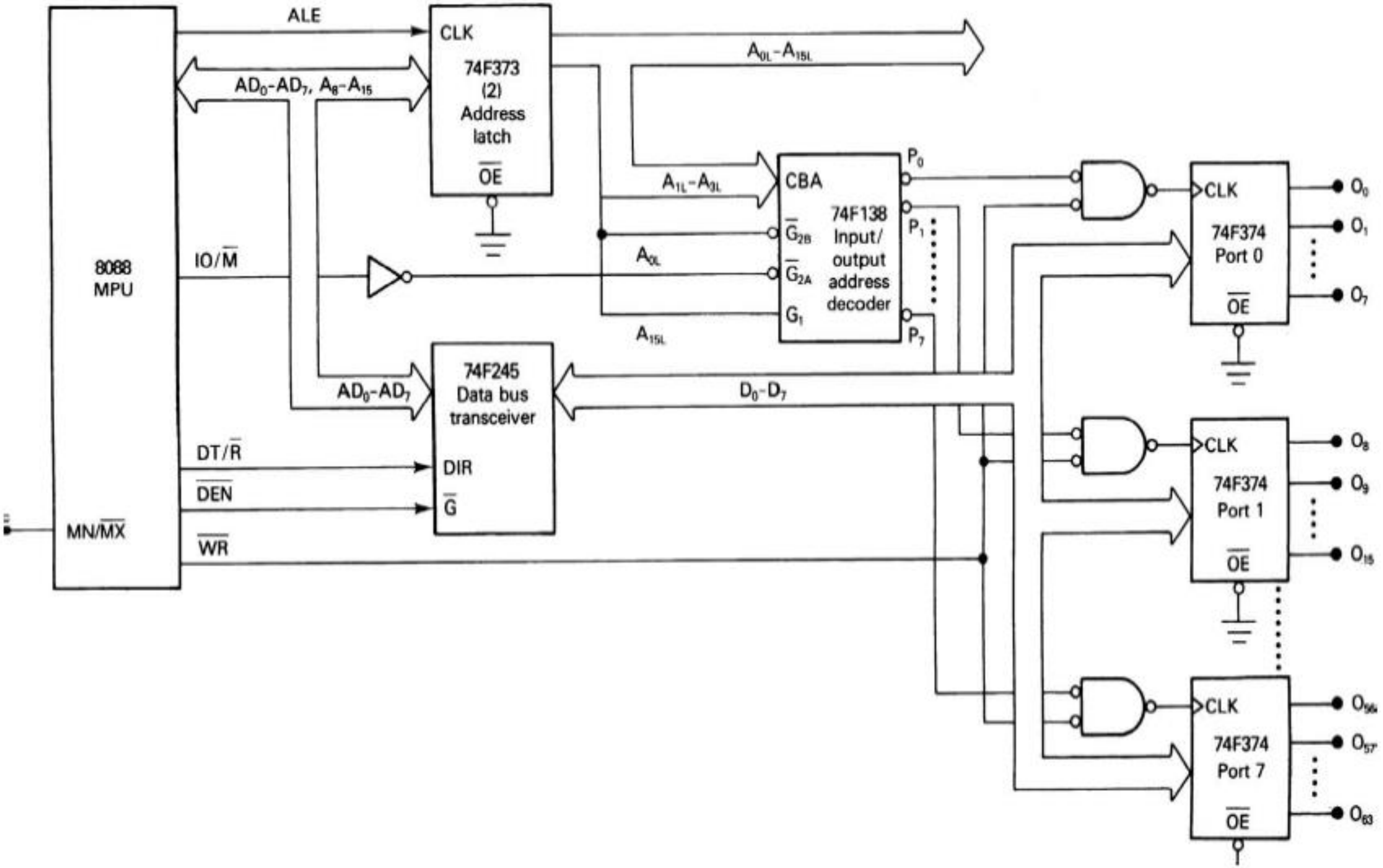
10.2



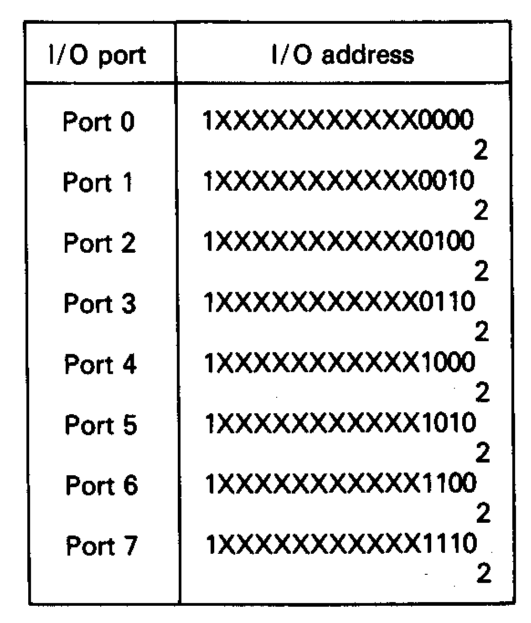
I/O Design in the 8088/86

- In every computer, when data is sent out by the CPU, the data on the data bus must be latched by the receiving device
- While memories have an internal latch to grab the data on the data bus, a latching system must be designed for ports
- Since the data provided by the CPU to the port is on the system data bus for a limited amount of time (50 - 1000ns) it must be latched before it is lost
- Likewise, when data is coming in by way of a data bus (either from port or memory) it must come in through a three-state buffer

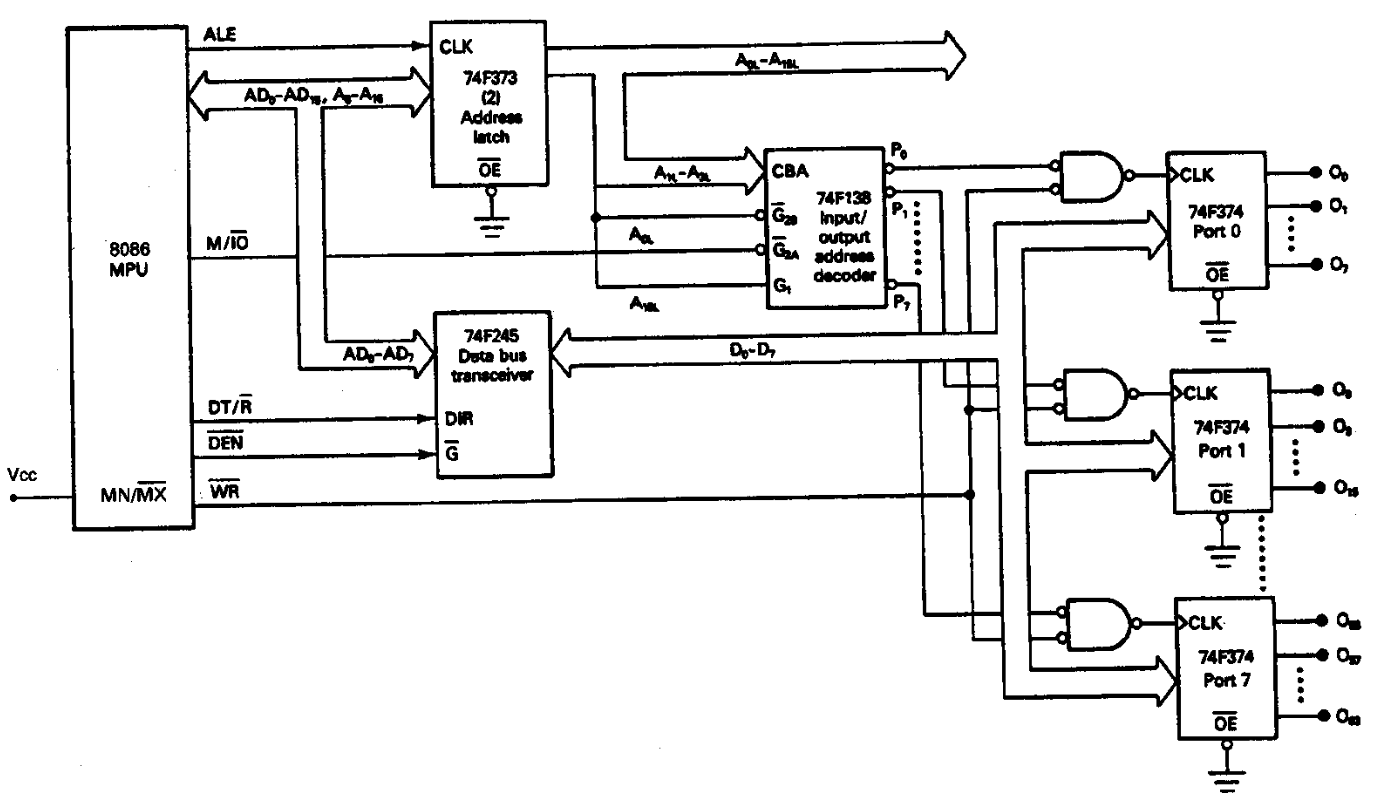
Example - 64 line parallel output circuit - 8088



I/O decoding and the 8086 64-line parallel port



I/O port	I/O address
Port 0	$1XXXXXXXXXXX0000_2$
Port 1	$1XXXXXXXXXXX0010_2$
Port 2	$1XXXXXXXXXXX0100_2$
Port 3	$1XXXXXXXXXXX0110_2$
Port 4	$1XXXXXXXXXXX1000_2$
Port 5	$1XXXXXXXXXXX1010_2$
Port 6	$1XXXXXXXXXXX1100_2$
Port 7	$1XXXXXXXXXXX1110_2$



Examples

- To which output port in the previous figure are data written when the address put on the bus during an output bus cycle is 8002h?
  - A15 .. A0 = 1000 0000 0000 0010b
  - A15L = 1
  - A0L = 0
  - A3L A2L A1L = 001
  - $\overline{P1}$ = 0

- Write a sequence of instructions that output the byte contents of the memory address DATA to output port 0 in the previous figure

```
MOV DX, 8000h
MOV AL,DATA
OUT DX, AL
```

Time Delay Loop and Blinking a LED at an Output

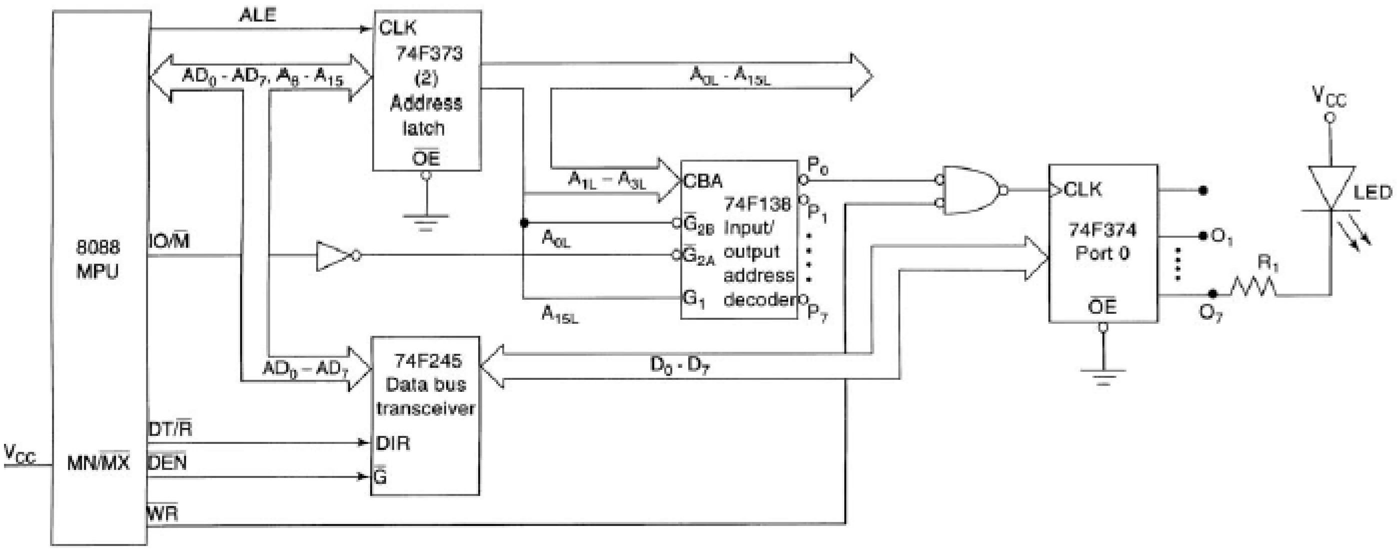


Figure 10–2 Driving an LED connected to an output port.

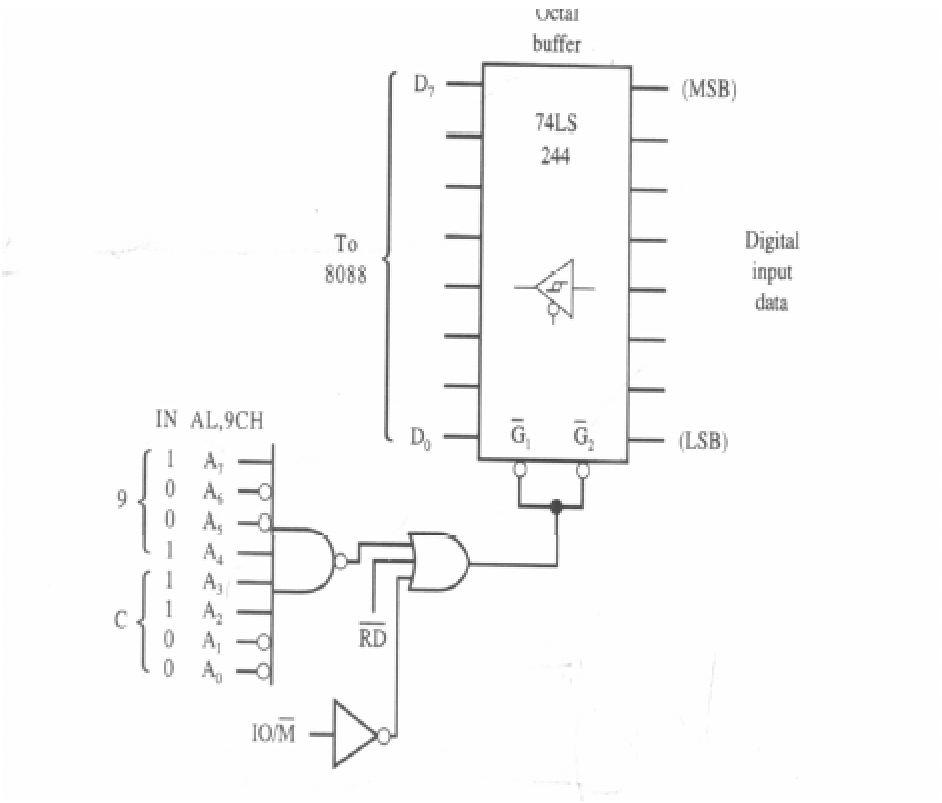
```
         MOV DX, 8000h      ;initialize address of port 0
         MOV AL, 00h        ; load data with bit 7 as logic 0
ON_OFF:  OUT DX,AL          ; turned on
         MOV CX,0FFFFh      ; load delay count of FFFFh
HERE:    LOOP HERE
         XOR  AL,80h        ; complement bit 7
         JMP  ON_OFF
```

Aprox.

17 T states * 64K * Frequency

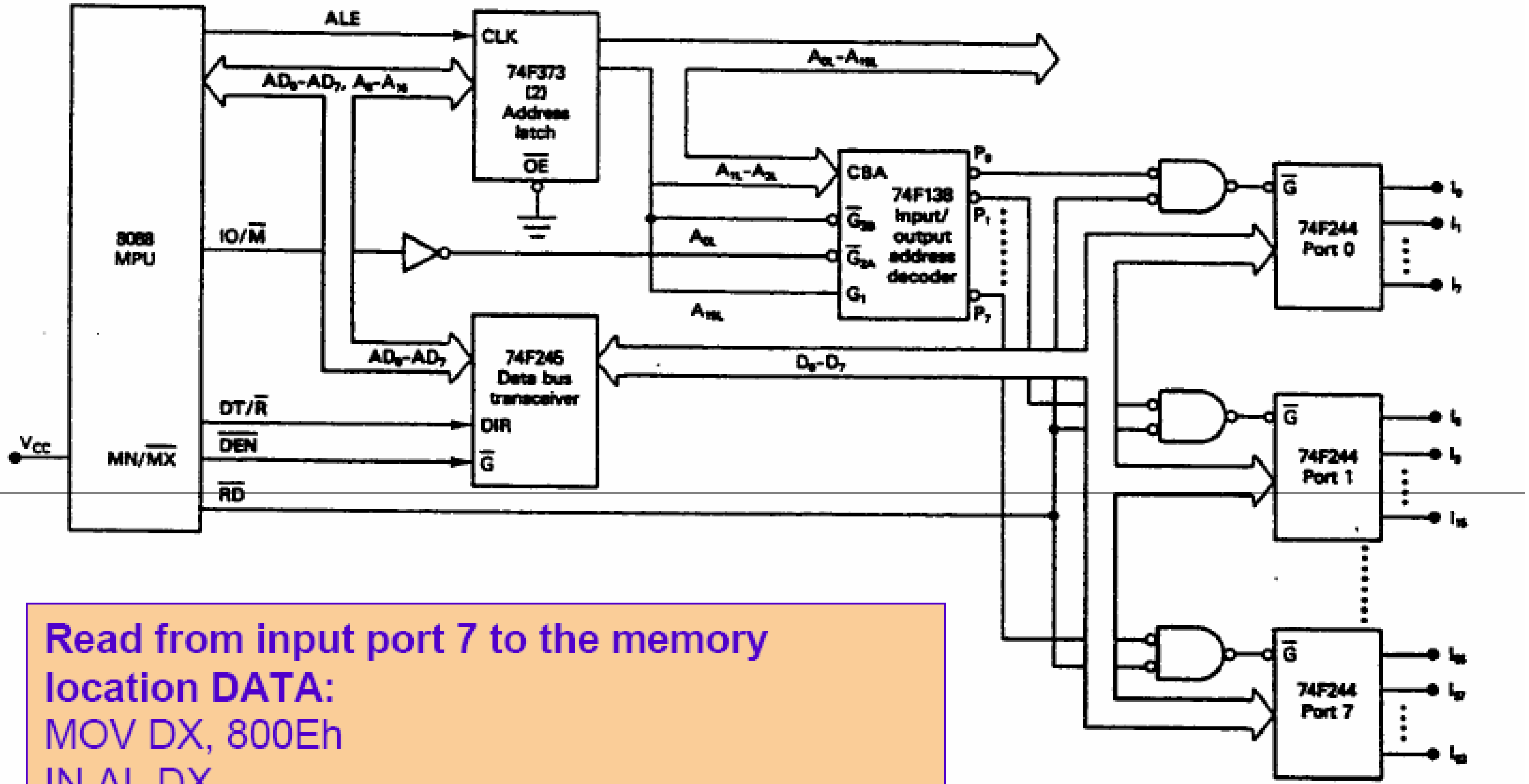
10.3 IN port design using the 74LS244

- Design for IN AL,9CH



- In order to prevent any unwanted data (garbage) to come into the system (global) data bus, all input devices must be isolated through the tri-state buffer. The 74LS244 not only plays this role but also provides the incoming signals sufficient strength (driving capability) to travel all the way to the CPU
- It must be emphasized that every device (memory, peripheral) connected to the global data bus must have a latch or a tri-state buffer. In some devices such as memory, they are internal but must be present.

Example - 64 line parallel input circuit

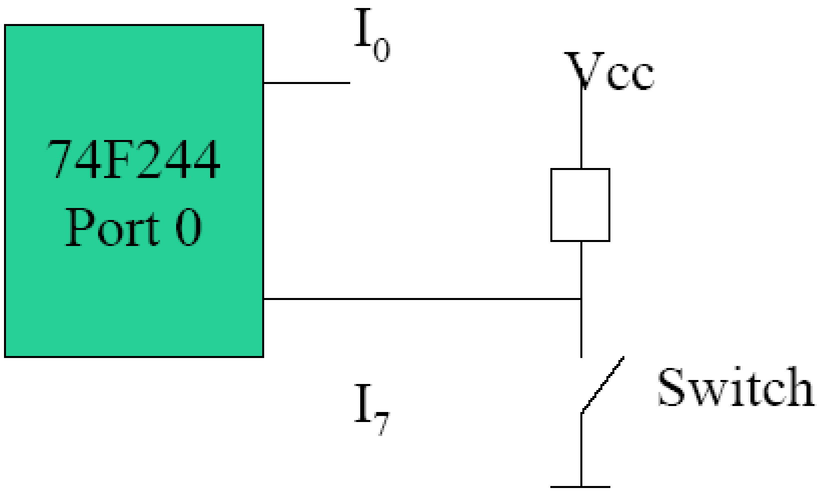


Read from input port 7 to the memory location DATA:

```
MOV DX, 800Eh
IN AL,DX
MOV DATA, AL
```

Example

- In practical applications, it is sometimes necessary within an I/O service routine to repeatedly read the value at an input line and test this value for a specific logic level.



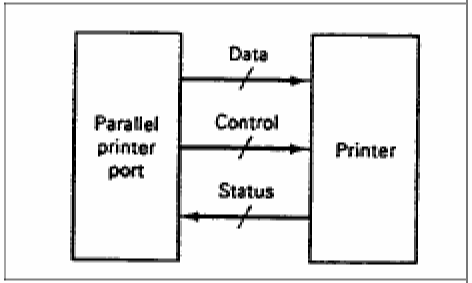
Poll the switch waiting for it to close

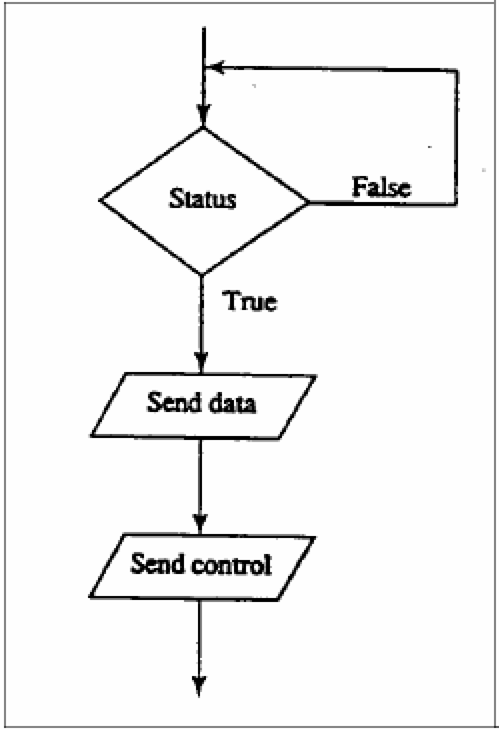
```
        MOV DX,8000h
POLL:   IN AL,DX
        SHL AL,1
        JC POLL
```

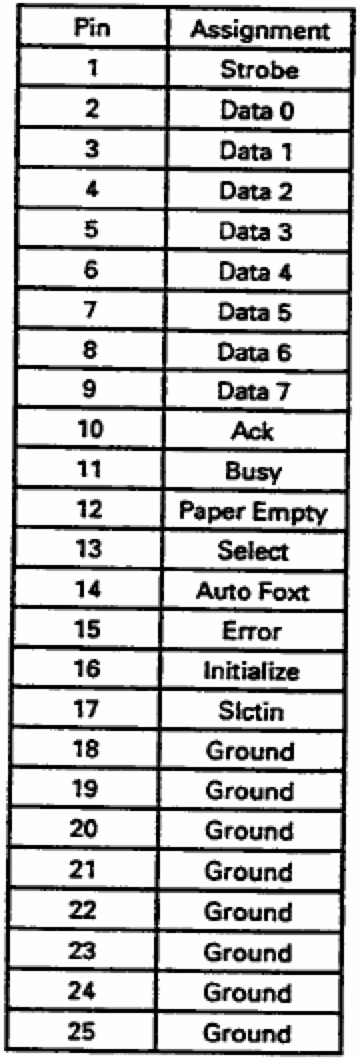
10.4 Input Output Handshaking

- The I/O ports of a computer typically operate at different data rates
- A hard disk drive, for example, might require the computer to input data at 10Mbps → 100Mbps
- CD-ROM drives operate at 300-600 Kbps
- However when inputting keystrokes from the operator, the data rate may fall to only one or two characters per sec.
- If the processor is to operate efficiently, one needs to develop a strategy to control or synchronize the flow of data between the processor and the widely varying rates of its I/O devices
- This type of synchronization is achieved by implementing what is known as handshaking as part of the input/output interface
- Printers typically have buffers that can be filled by the computer at high speed
- Once full the computer must wait while the data in the buffer is printed
- Most printer manufacturers have settled on a standard set of data and control signals Centronics Parallel Printer Interface

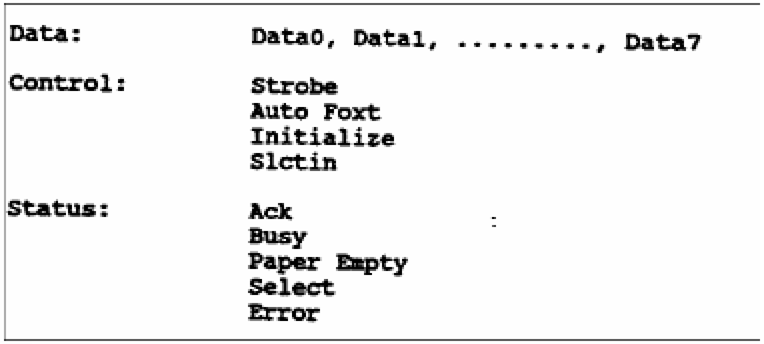
Parallel Printer Interface







Pin	Assignment
1	Strobe
2	Data 0
3	Data 1
4	Data 2
5	Data 3
6	Data 4
7	Data 5
8	Data 6
9	Data 7
10	Ack
11	Busy
12	Paper Empty
13	Select
14	Auto Foxt
15	Error
16	Initialize
17	Slctin
18	Ground
19	Ground
20	Ground
21	Ground
22	Ground
23	Ground
24	Ground
25	Ground



Data:	Data0, Data1, ..., Data7
Control:	Strobe Auto Foxt Initialize Slctin
Status:	Ack Busy Paper Empty Select Error

ACK is used by printer to acknowledge receipt of data and can accept a new character.

BUSY high if printer is not ready to accept a new character

SELECT when printer is turned on

ERROR goes low when there are conditions such as paper jam, out of paper, offline

STROBE when PC presents a character

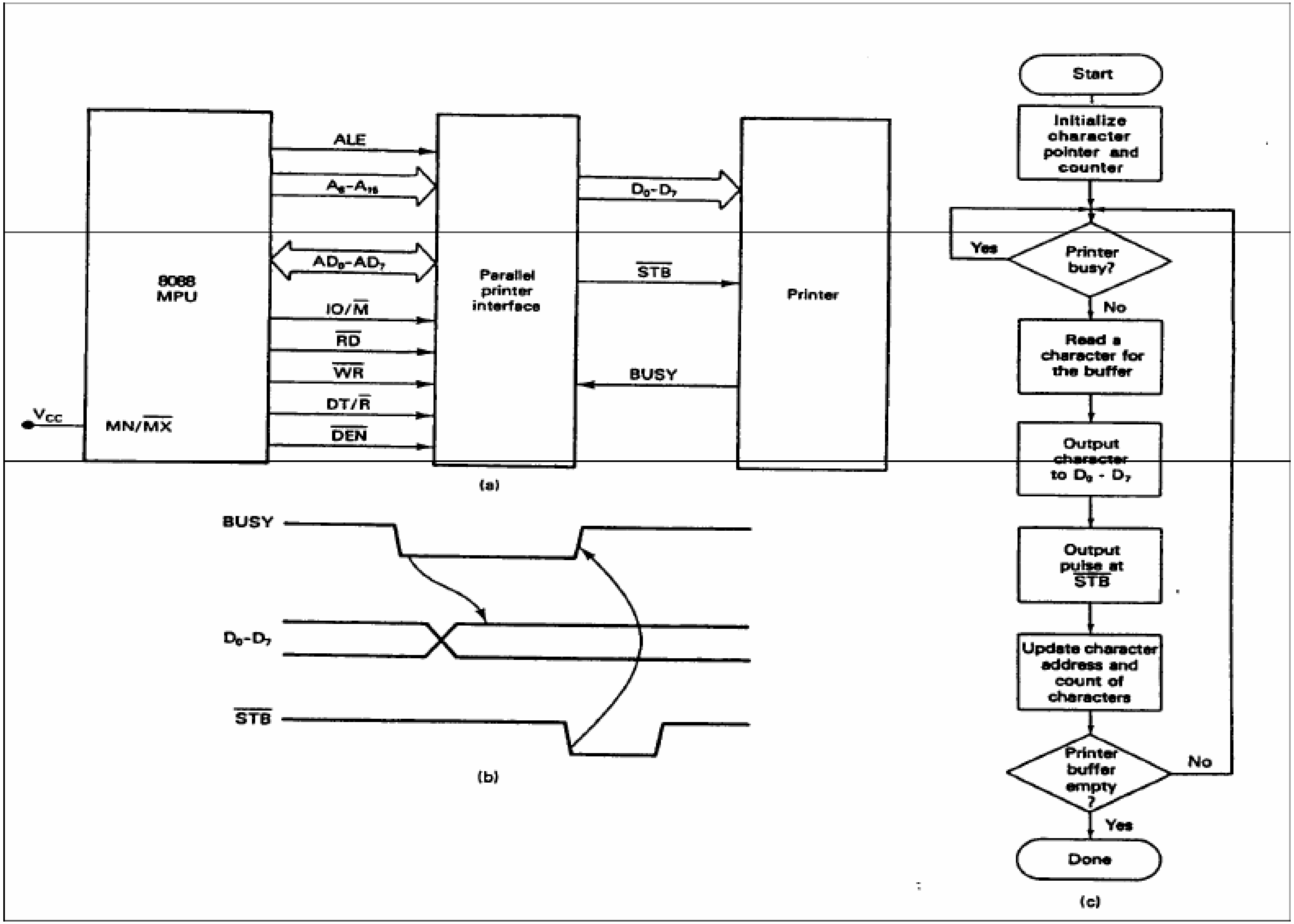
INITIALIZE Clear Printer Buffer and reset control

Operational Principle - Parallel Printer Port

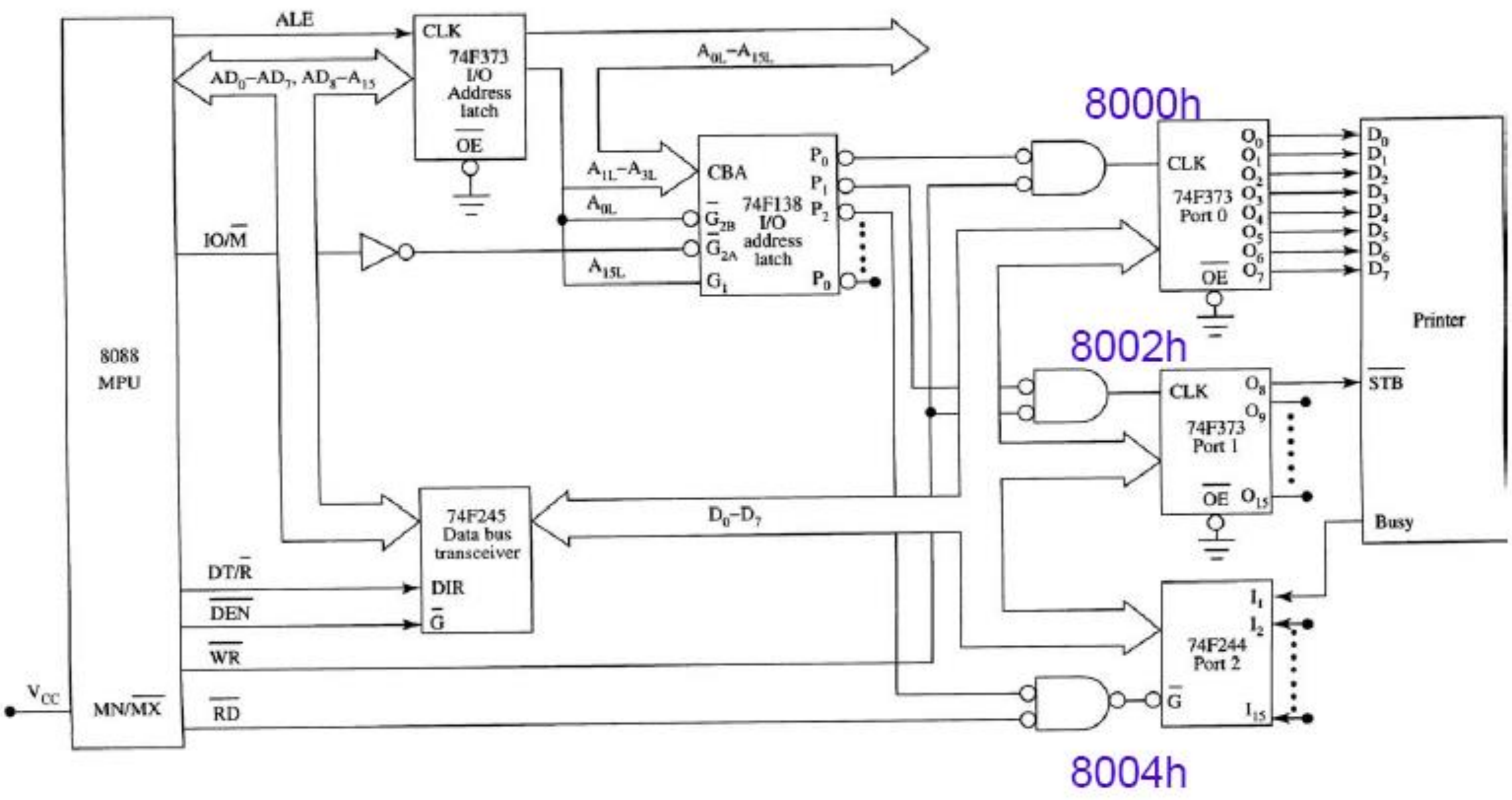
- The computer checks the BUSY signal from the printer, if not BUSY then
- When the PC presents a character to the data pins of the printer, it activates the STROBE pin, telling it that there is a byte sitting at the data pins. Prior to asserting STROBE pin, the data must be at at the printer's data pins for at least 0.5 microsec. (data setup time)
- The STROBE must stay for 0.5 microsec

- The printer asserts BUSY pin indicating the computer to wait
- When the printer picks up the data, it sends back the ACK signal, keeps ACK low for 5 microsec.
- As the ACK signal is going high, the printer makes the BUSY pin low to indicate that it is ready to accept the next byte
- The CPU can use ACK or BUSY signals from the printer to initiate the process of sending another byte

Handshaking



Printer Interface Circuit



Example

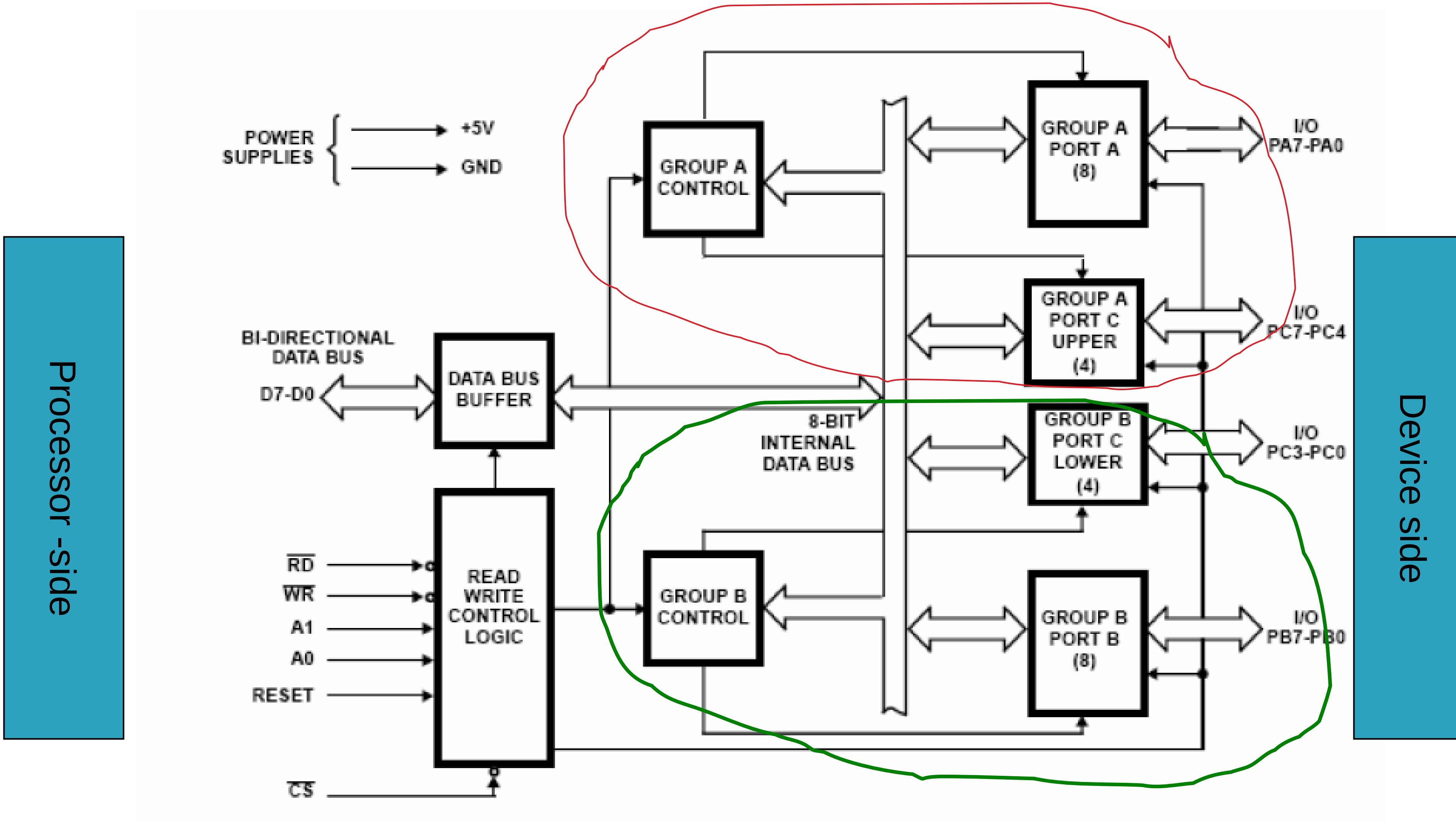
- Write a program that implements the flowchart. Character data is held in memory starting at address PRNT_BUFF, the number of characters held in the buffer is identified by the count address CHAR_COUNT.

```
                MOV CL, CHAR_COUNT
                MOV SI,  OFFSET PRNT_BUFF

POLL_BUSY:      MOV DX,8004h
                IN AL,DX
                AND AL,01h                          BUSY input checked
                JNZ POLL_BUSY

                MOV AL, [SI]
                MOV DX,8000h                       Character is output
                OUT DX,AL

                MOV AL, 00h            ;STB = 0
                MOV DX,8002h
                OUT DX,AL                            So as the strobe
                MOV BX,0Fh             ; delay for STB = 0
                STROBE: DEC BX
                JNZ STROBE
                MOV AL,01h
                OUT DX,AL              ; STB bar = 1

                 INC SI
                 DEC CL
                JNZ POLL_BUSY
```

10.5 The 8255 Programmable Peripheral Interface

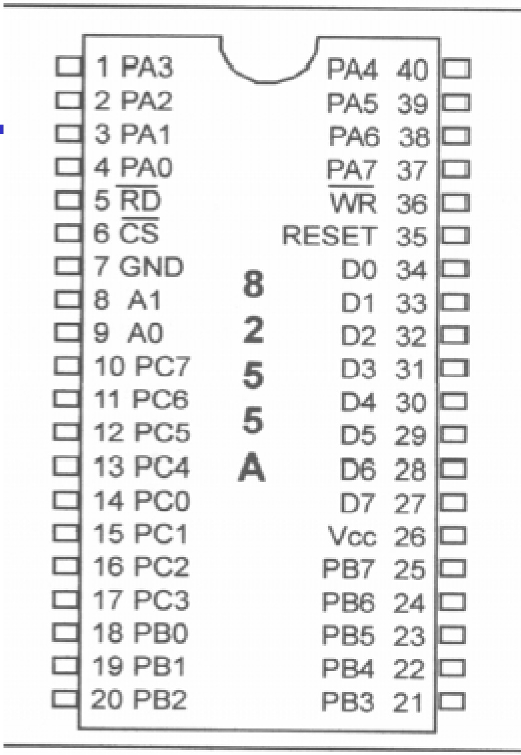
- Intel has developed several peripheral controller chips designed to support the 80x86 processor family. The intent is to provide a complete I/O interface in one chip.
- 8255 PPI provides three 8 bit input ports in one 40 pin package making it more economical than 74LS373 and 74LS244
- The chip interfaces directly to the data bus of the processor, allowing its functions to be programmed; that is in one application a port may appear as an output, but in another, by reprogramming it as an input. This is in contrast with the 74LS373 and 74LS244 which are hardwired and fixed
- Other peripheral controller chips include the 8259 Programmable Interrupt Controller (PIC), the 8253/54 Programmable Interval Timer (PIT) and the 8237 DMA controller

8255A internal

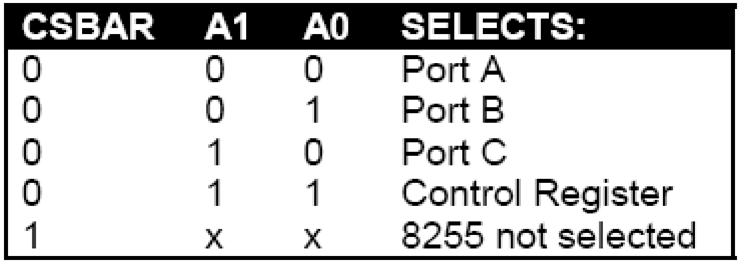


8255 Pins

- PA0 - PA7: input, output, or bi-directional port
- PB0 - PB7: input or output
- PC0 - PC7: This 8 bit port can be all input or output. It can also be split into two parts, CU (PC4 - PC7) and CL (PC0 - PC3). Each can be used for input and output.
- $\overline{\text{RD}}$ or $\overline{\text{WR}}$
  - $\overline{\text{IOR}}$ and $\overline{\text{IOW}}$ of the system are connected
- RESET
- A0, A1, and $\overline{\text{CS}}$
  - $\overline{\text{CS}}$ selects the entire chip whereas A0 and A1 select the specific port (A, B, or C)

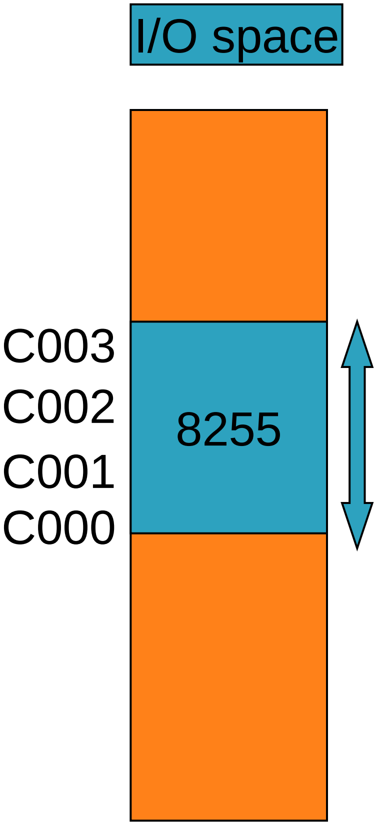


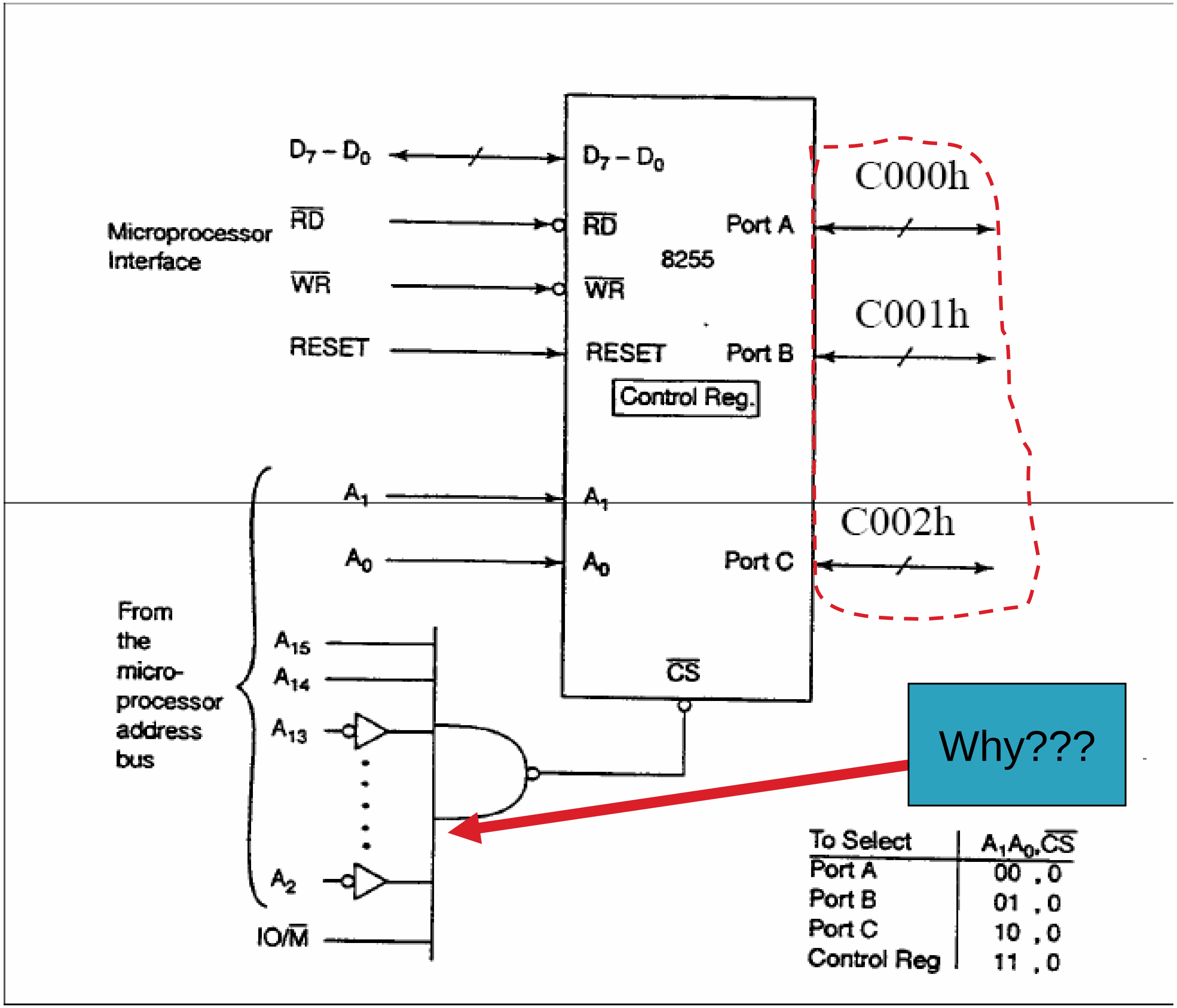
ure 11-11 8255 PPI Chip



CSBAR	A1	A0	SELECTS:
0	0	0	Port A
0	0	1	Port B
0	1	0	Port C
0	1	1	Control Register
1	x	x	8255 not selected

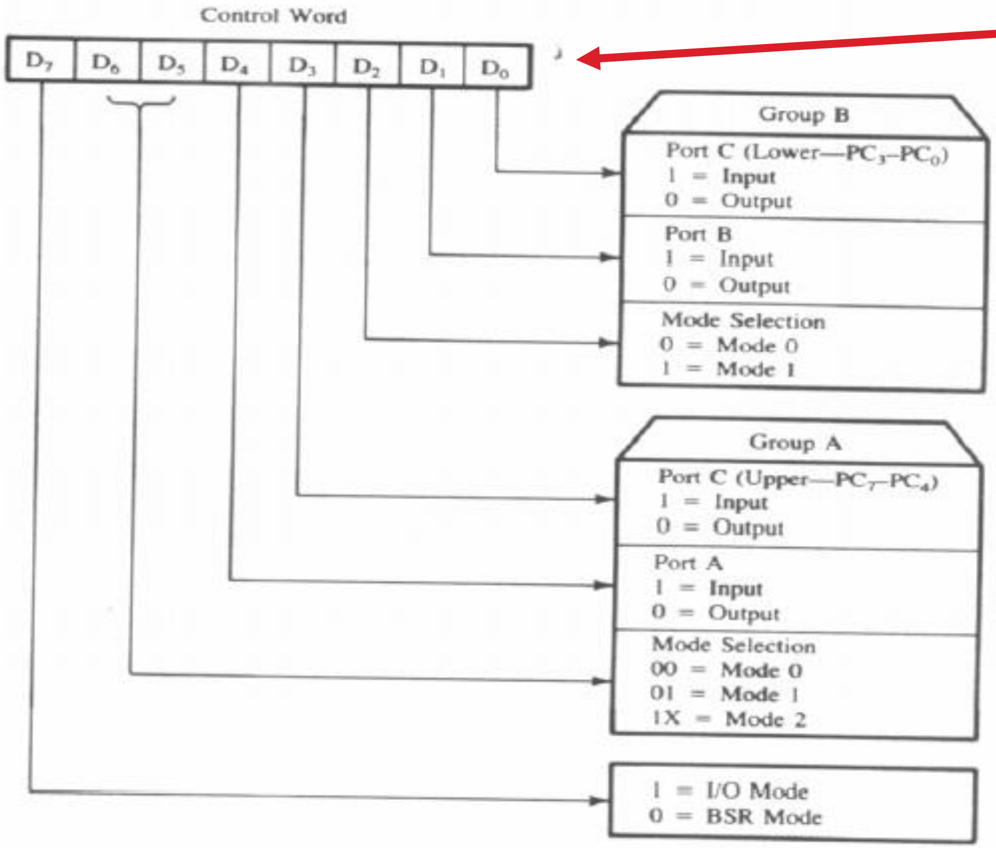
Addressing an 8255





To Select	$A_1A_0,\overline{CS}$
Port A	00 , 0
Port B	01 , 0
Port C	10 , 0
Control Reg	11 , 0

8255 Control Word Format

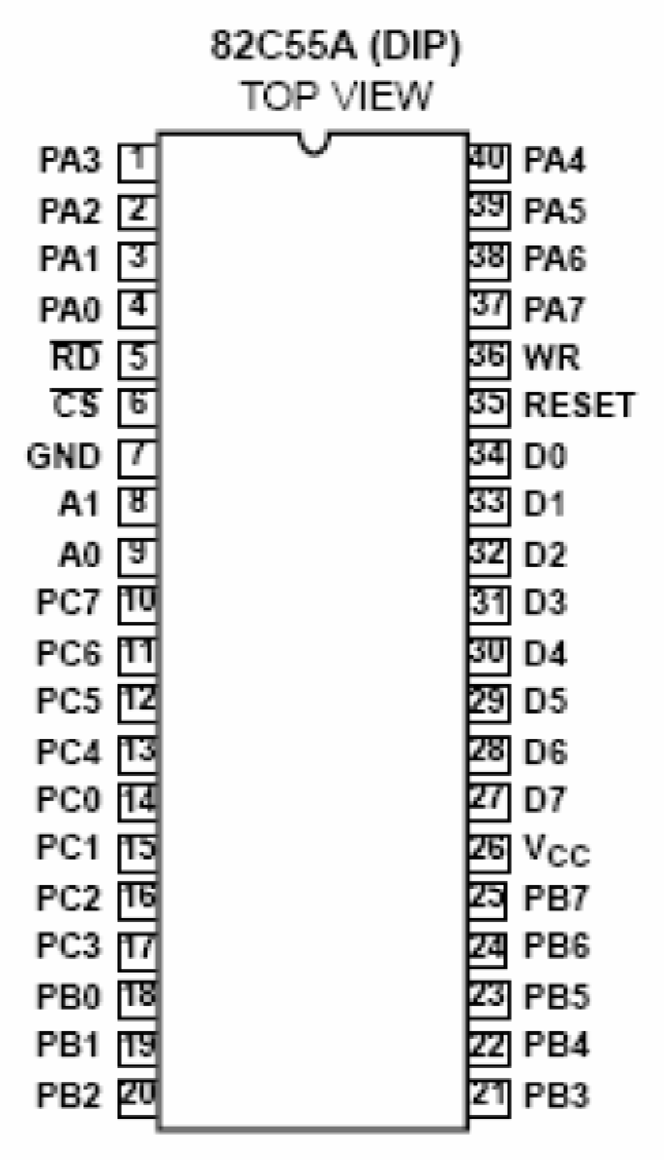


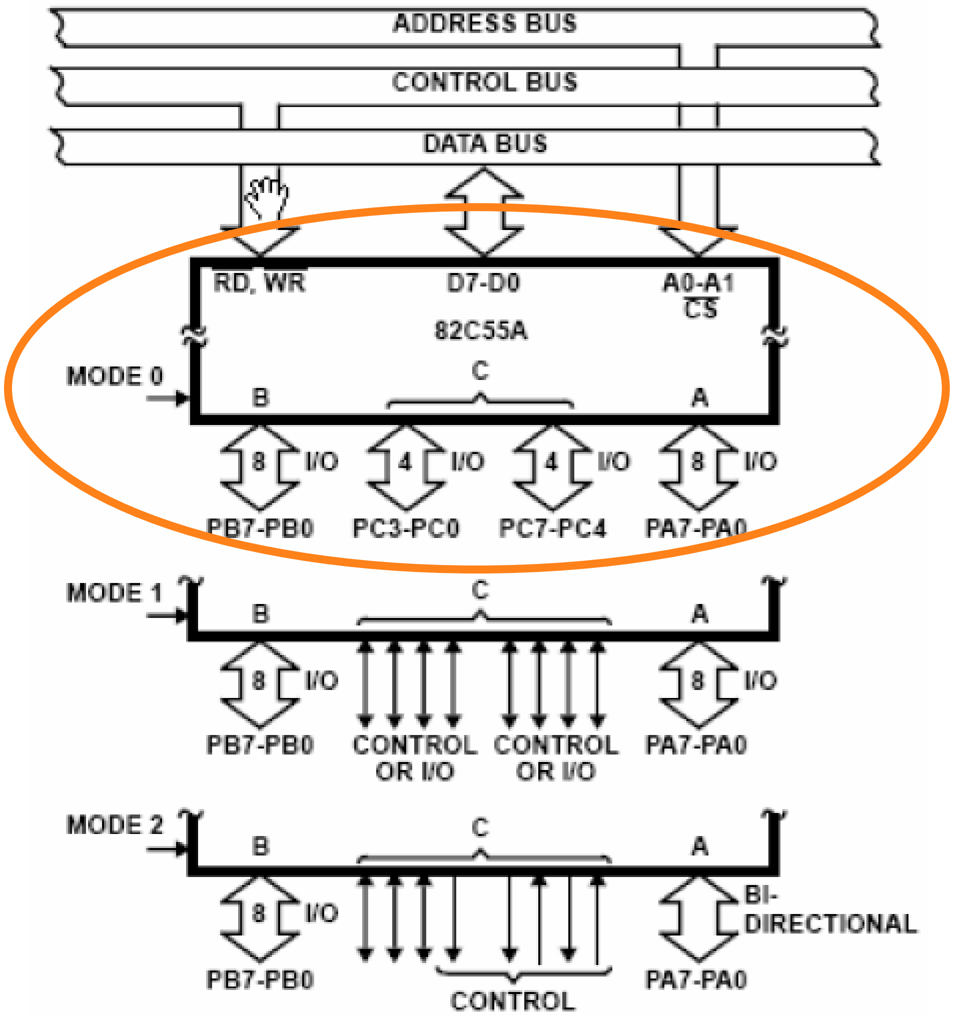
Control register

FIGURE 15.4

8255A Control Word Format for I/O Mode

SOURCE: Adapted from Intel Corporation, *Peripheral Components* (Santa Clara, Calif.: Author, 1993), p. 3–

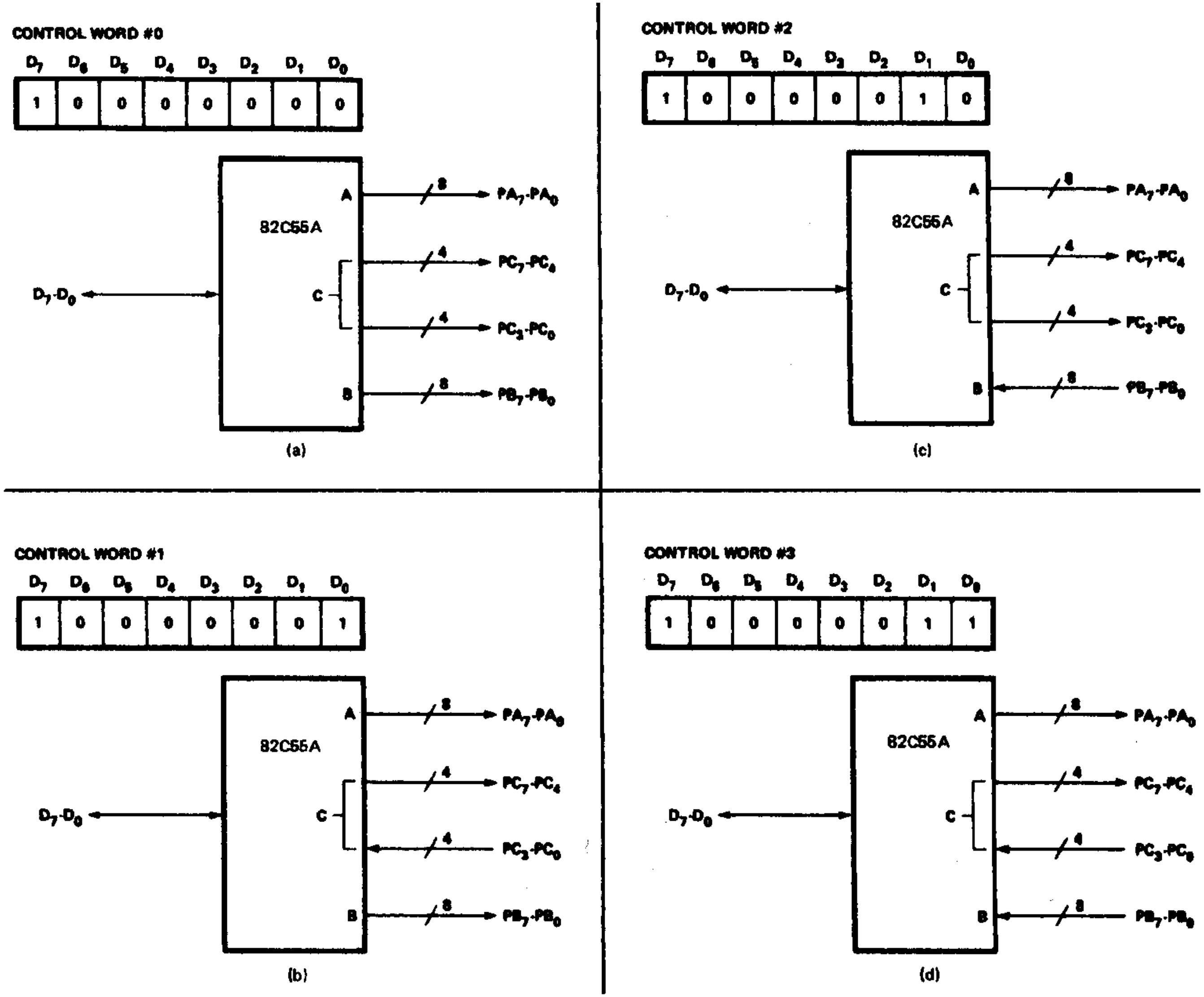

82C55A (DIP)
TOP VIEW
PA3 1
PA2 2
PA1 3
PA0 4
RD 5
CS 6
GND 7
A1 8
A0 9
PC7 10
PC6 11
PC5 12
PC4 13
PC0 14
PC1 15
PC2 16
PC3 17
PB0 18
PB1 19
PB2 20
40 PA4
39 PA5
38 PA6
37 PA7
36 WR
35 RESET
34 D0
33 D1
32 D2
31 D3
30 D4
29 D5
28 D6
27 D7
26 VCC
25 PB7
24 PB6
23 PB5
22 PB4
21 PB3


ADDRESS BUS
CONTROL BUS
DATA BUS
RD, WR
D7-D0
A0-A1
CS
82C55A
MODE 0
B
C
A
8
I/O
4
I/O
4
I/O
8
I/O
PB7-PB0
PC3-PC0
PC7-PC4
PA7-PA0
MODE 1
B
C
A
8
I/O
8
I/O
PB7-PB0
CONTROL OR I/O
CONTROL OR I/O
PA7-PA0
MODE 2
B
C
A
8
I/O
BI-DIRECTIONAL
PB7-PB0
CONTROL
PA7-PA0

Mode 0 - Simple input/output

- Simple I/O mode: any of the ports A, B, CL, and CU can be programmed as input or output.
- Example: Configure port A as input, B as output, and all the bits of port C as output assuming a base address of 50h
- Control word should be 1001 0000b = 90h

```
PORTA  EQU 50h
PORTB  EQU 51h
PORTC  EQU 52h
CNTREG EQU 53h
MOV AL, 90h
OUT CNTREG,AL
IN AL, PORTA
OUT PORTB, AL
OUT PORTC, AL
```



CONTROL WORD #0

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0

(a)

CONTROL WORD #1

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	1

(b)

CONTROL WORD #2

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	1	0

(c)

CONTROL WORD #3

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	1	1

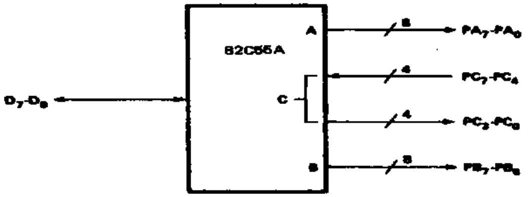
(d)

Mode 0 control words (I/O)

(16 cases)

CONTROL WORD #4

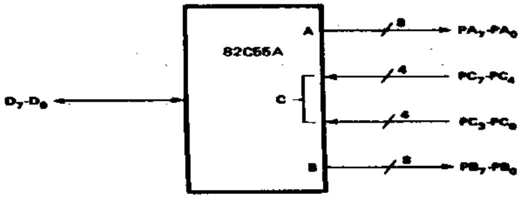
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	0



(e)

CONTROL WORD #5

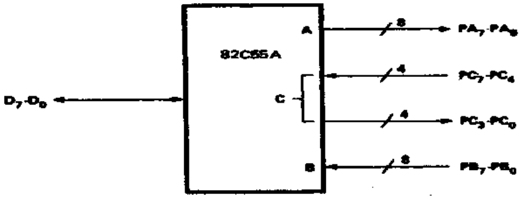
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	1



(f)

CONTROL WORD #6

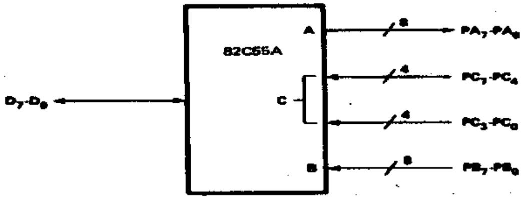
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	0



(g)

CONTROL WORD #7

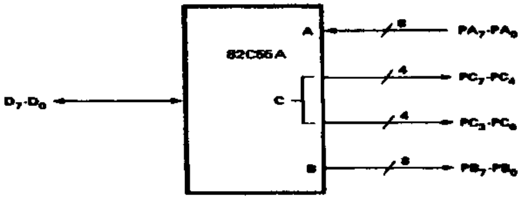
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	1



(h)

CONTROL WORD #8

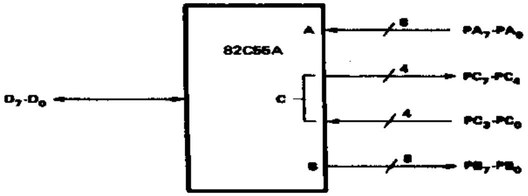
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	0



(i)

CONTROL WORD #9

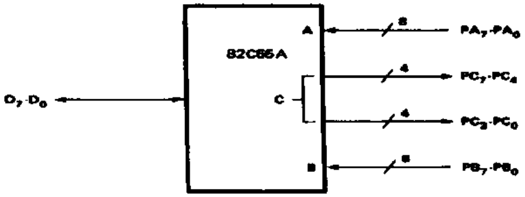
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	1



(j)

CONTROL WORD #10

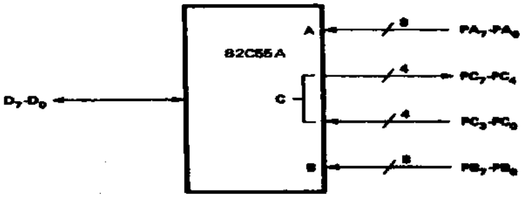
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	0



(k)

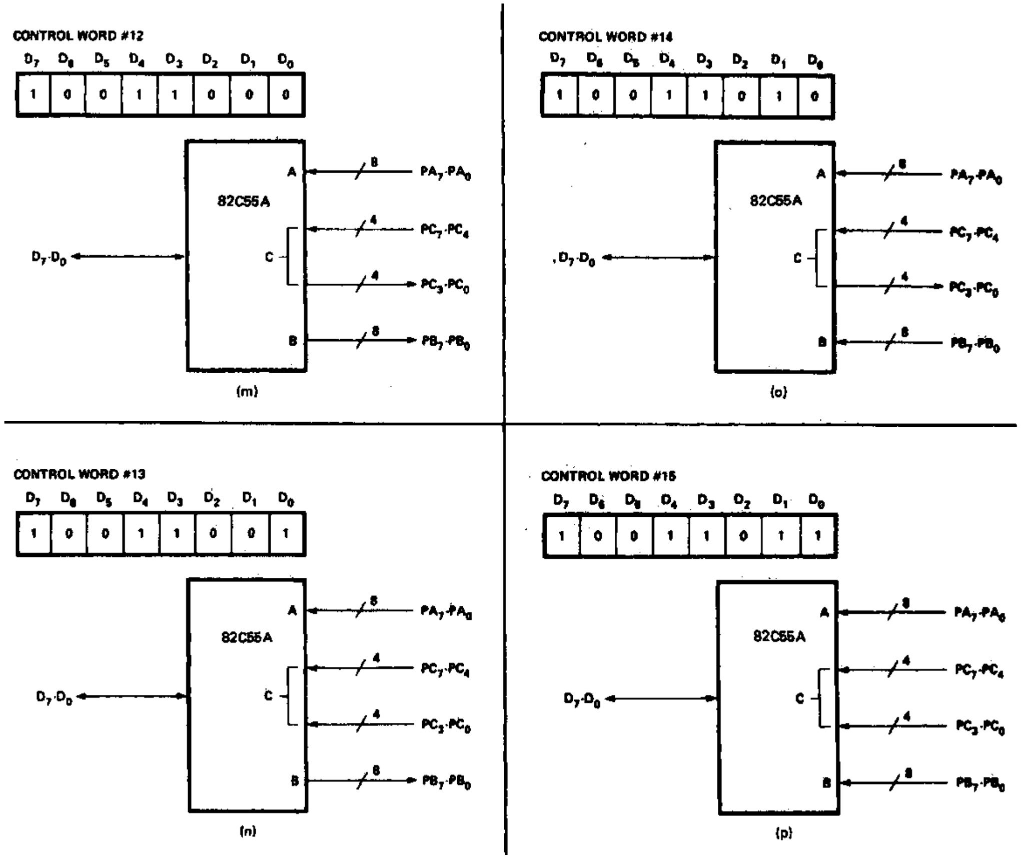
CONTROL WORD #11

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	1



(l)

Mode 0 control words (I/O)



CONTROL WORD #12

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	0	0

(m)

CONTROL WORD #14

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	1	0

(o)

CONTROL WORD #13

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	0	1

(n)

CONTROL WORD #15

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	1	1	0	1	1

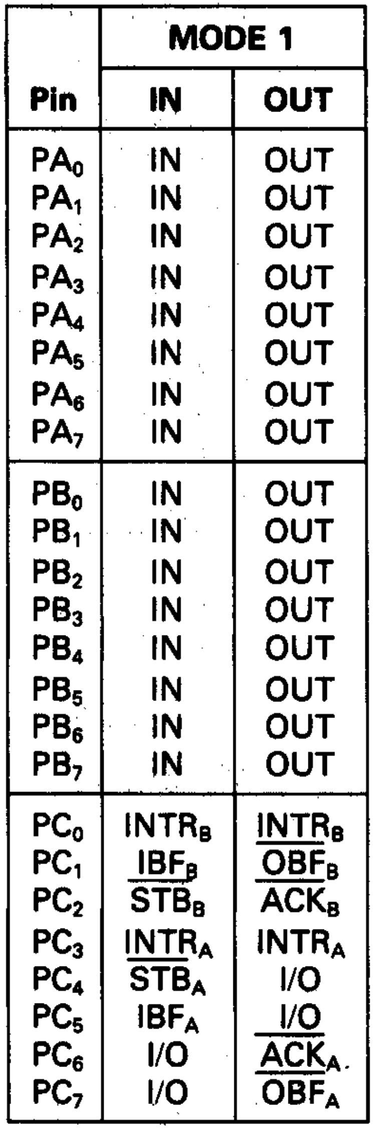
(p)

Mode 0 control words (I/O)

Mode 1: I/O with Handshaking Capability

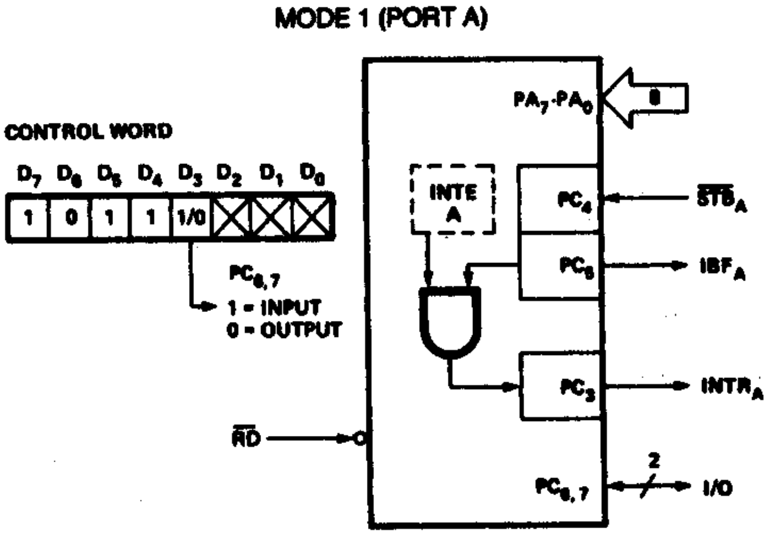
- Handshaking refers to the process of communicating back and forth between two intelligent devices
- Example: Process of communicating with a printer
  - a byte of data is presented to the data bus of the printer
  - the printer is informed of the presence of a byte of data to be printed by activating its strobe signal
  - whenever the printer receives the data it informs the sender by activating an output signal called ACK
  - the ACK signal initiates the process of providing another byte of data to the printer
- 8255 in mode 1 is equipped with resources to handle handshaking signals

Setup of Mode 1

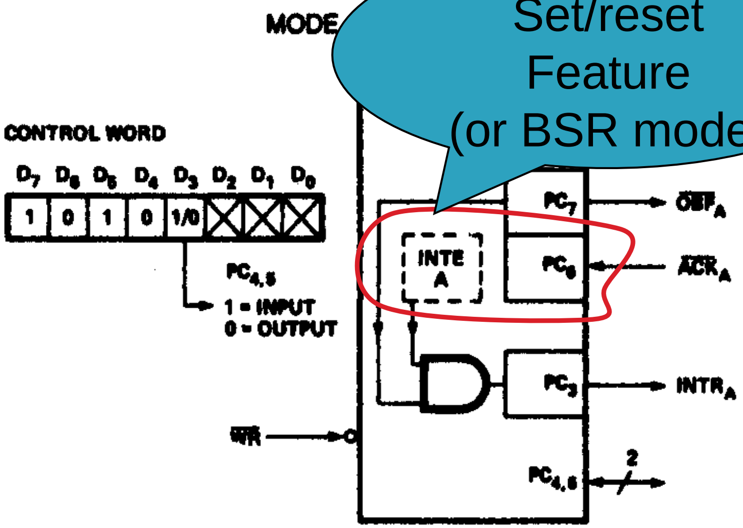


Pin	MODE 1	
	IN	OUT
PA_0	IN	OUT
PA_1	IN	OUT
PA_2	IN	OUT
PA_3	IN	OUT
PA_4	IN	OUT
PA_5	IN	OUT
PA_6	IN	OUT
PA_7	IN	OUT
PB_0	IN	OUT
PB_1	IN	OUT
PB_2	IN	OUT
PB_3	IN	OUT
PB_4	IN	OUT
PB_5	IN	OUT
PB_6	IN	OUT
PB_7	IN	OUT
PC_0	$INTR_B$	$INTR_B$
PC_1	IBF_B	$\overline{OBF}_B$
PC_2	$\overline{STB}_B$	$\overline{ACK}_B$
PC_3	$INTR_A$	$INTR_A$
PC_4	$\overline{STB}_A$	I/O
PC_5	IBF_A	I/O
PC_6	I/O	$\overline{ACK}_A$
PC_7	I/O	$\overline{OBF}_A$

Mode 1 (configuration of port A)



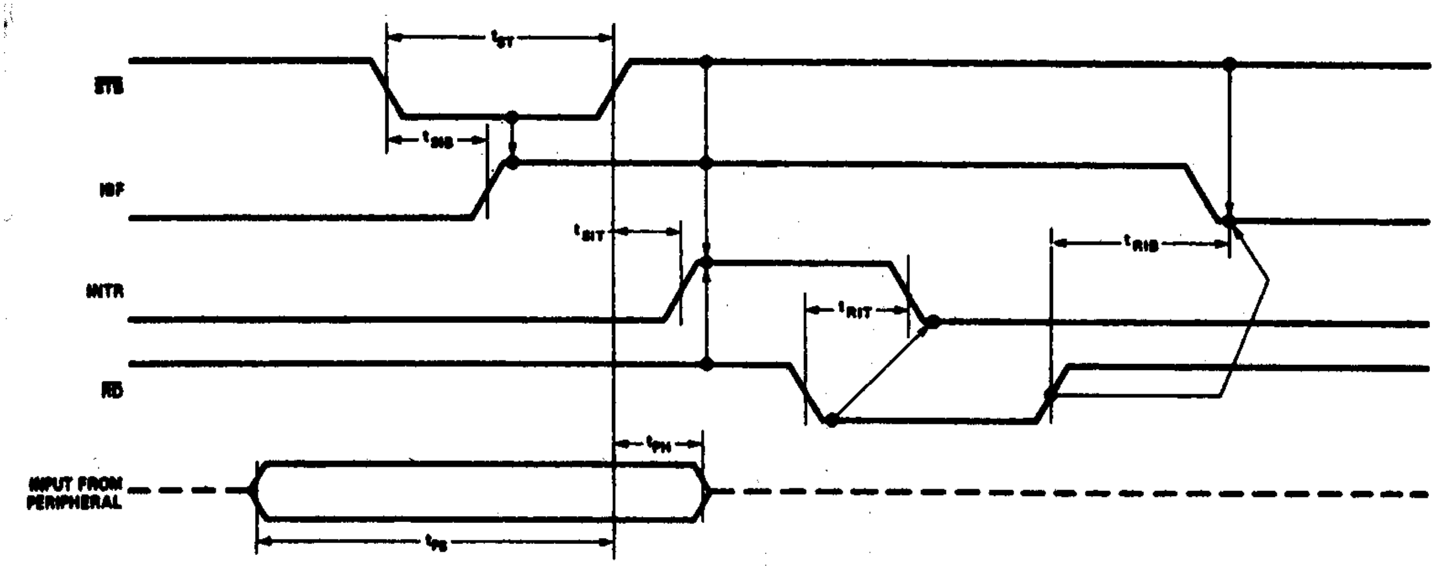
input



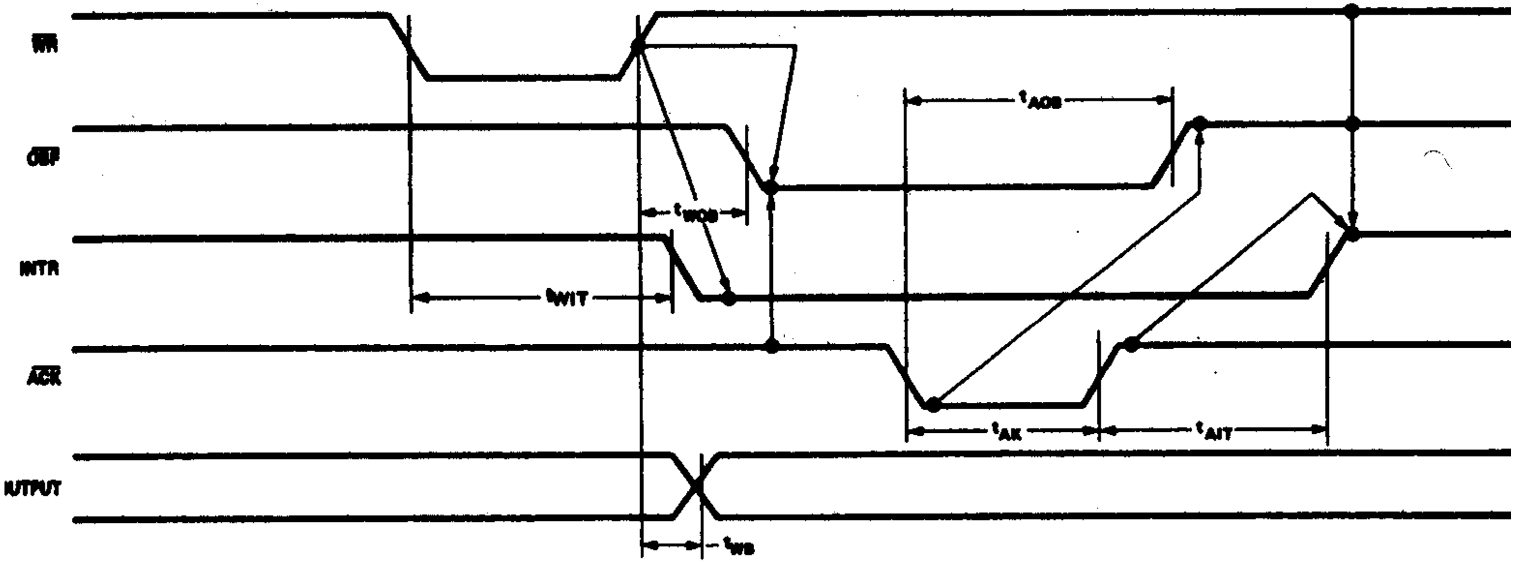
Set/reset Feature (or BSR mode)

output

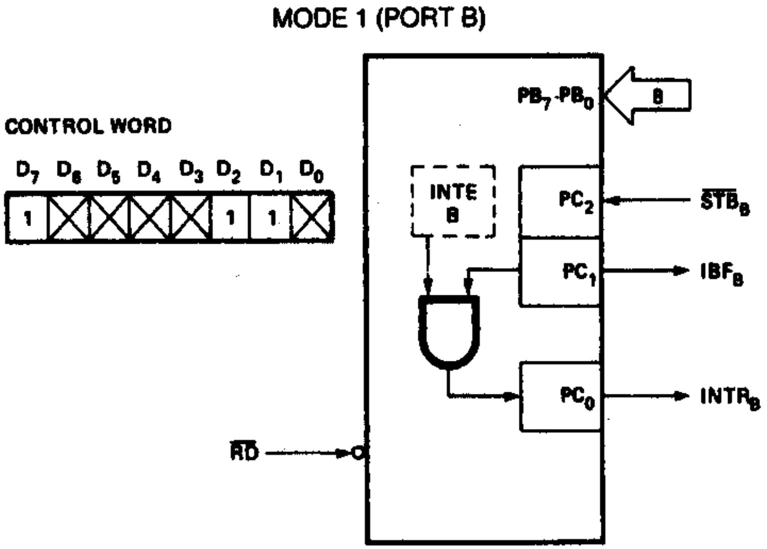
Timing diagram of port A (input)



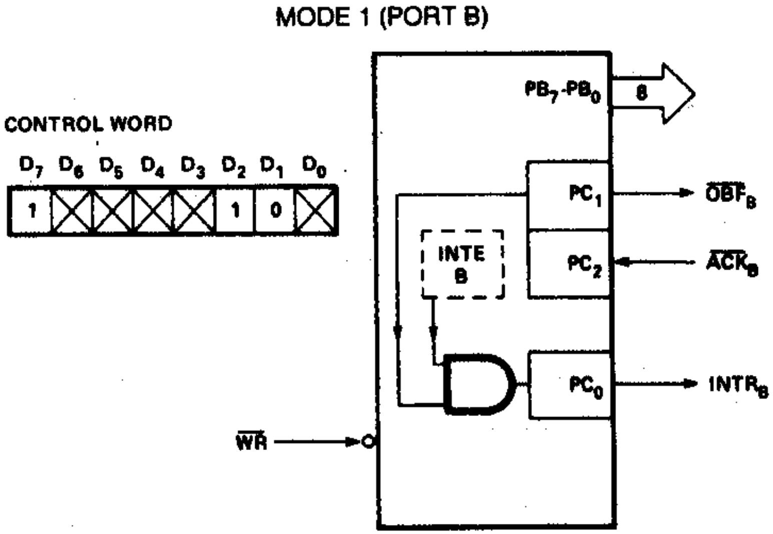
Timing diagram of port A (output)



Mode 1 (configuration of port B)

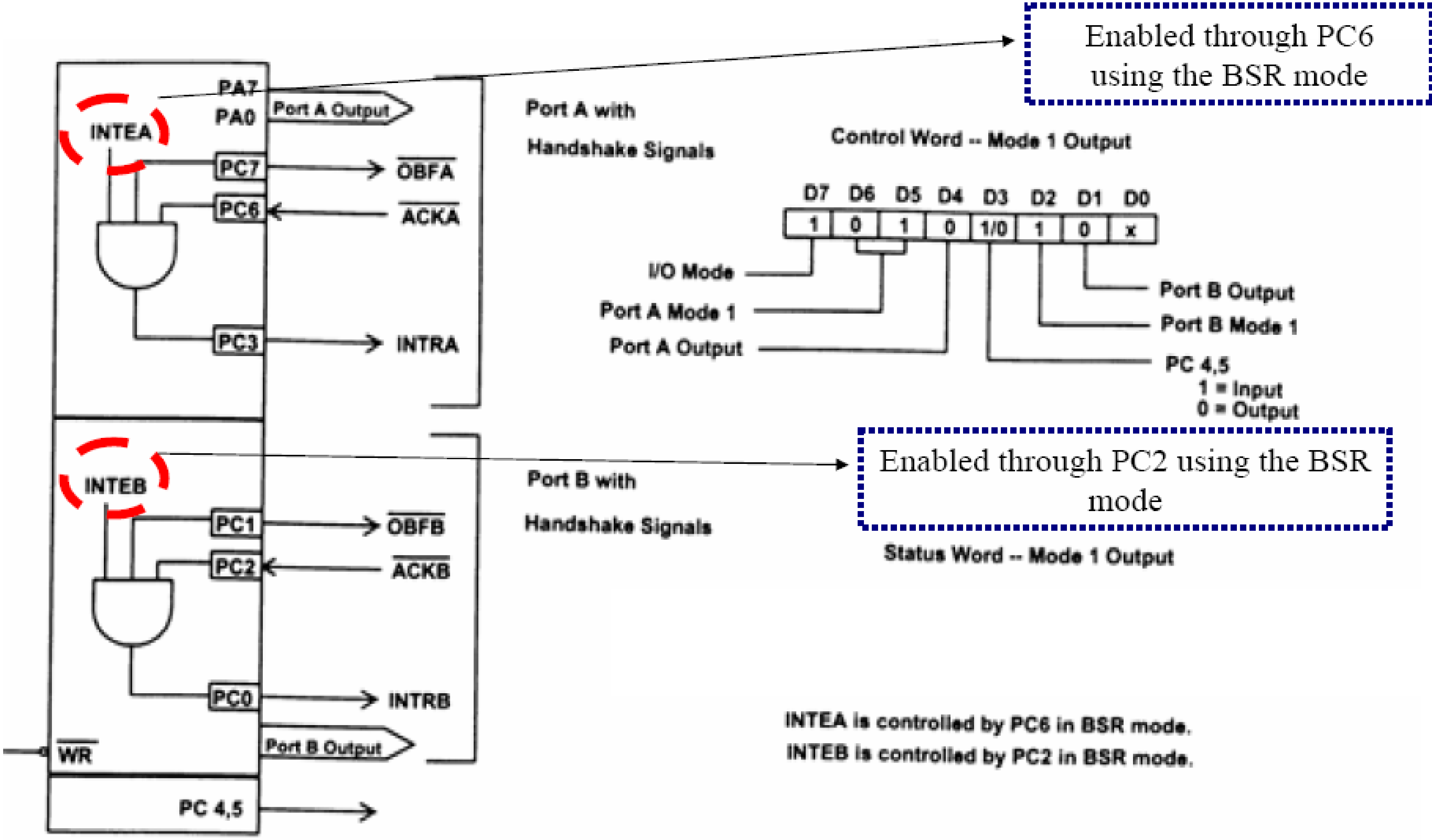


input



output

Mode 1 Strobed Output



Enabled through PC6 using the BSR mode

Control Word -- Mode 1 Output

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	0	1/0	1	0	x

Enabled through PC2 using the BSR mode

INTEA is controlled by PC6 in BSR mode.
INTEB is controlled by PC2 in BSR mode.

Mode 1 Strobed Output Signals

- $\overline{\text{OBFa}}$ (output buffer full for port A)
  - indicates that the CPU has written a byte of data into port A
  - must be connected to the STROBE of the receiving equipment
  - Goes back high again after ACK'ed by the peripheral.
- $\overline{\text{ACKa}}$ (acknowledge for port A)
  - through ACK, 8255 knows that data at port A has been picked up by the receiving device
  - 8255 then makes $\overline{\text{OBFa}}$ high to indicate that the data is old now. $\overline{\text{OBFa}}$ will not go low until the CPU writes a new byte of data to port A.
- INTRa (interrupt request for port A)
  - it is the rising edge of ACK that activates INTRa by making it high. INTRa is used to get the attention of the microprocessor.
  - it is important that INTRa is high only if INTEa, OBFa, ACKa are all high
  - it is reset to zero when the CPU writes a byte to port A
- The 8255 enables the monitoring the status signals INTR, OBF, and INTE for both ports A and B. This is done by **reading port C** into the accumulator and testing the bits. This feature allows the implementation of polling

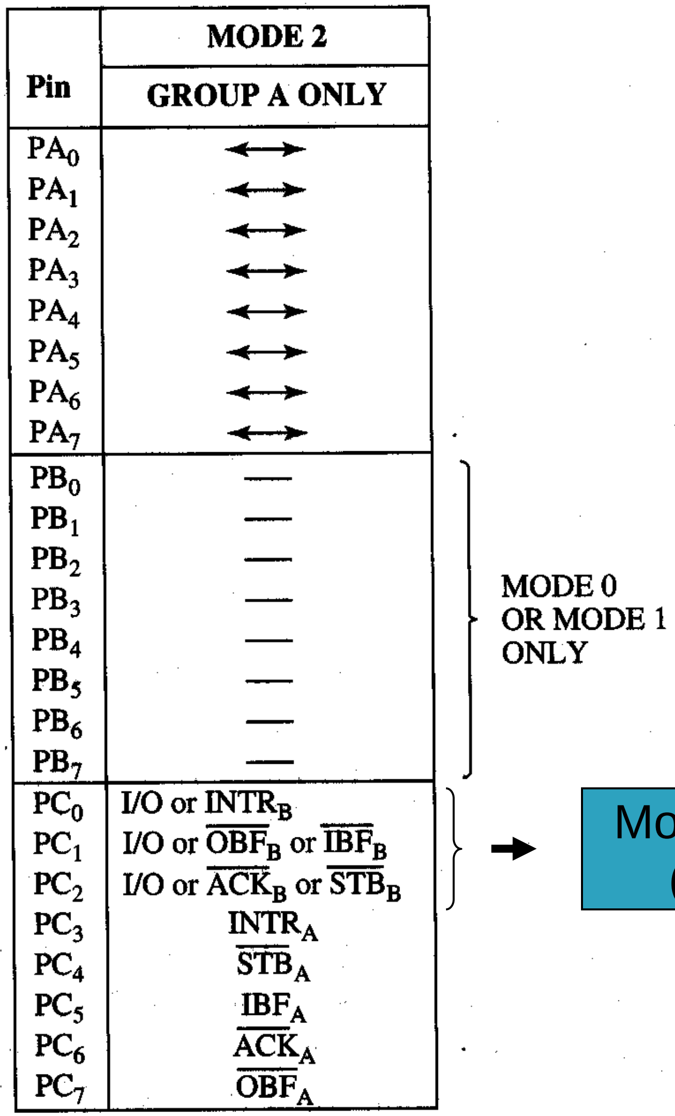
Mode 1 Input Ports with Handshaking Signals

- $\overline{\text{STB}}$
  - When an external peripheral device provides a byte of data to an input port, it informs the 8255 through the STB pin. STB is of limited duration
- IBF (Input Buffer Full)
  - In response to STB, the 8255 latches into its internal register the data present at PA0-PA7 or PB0-PB7.
  - Through IBF it indicates that it has latched the data but it has not been read by the CPU yet
  - To get the attention of the CPU, IBF activates INTR
- INTR
  - Falling edge of RD makes INTR low
  - The RD signal from the CPU is of limited duration and when it goes high the 8255 in turn makes IBF inactive by setting it low
  - IBF in this way lets the peripheral know that the byte of data was latched by the 8255 and read into the CPU as well.
- The two flip flops INTEA and INTB are set/reset using the BSR mode. The INTEA is enabled or disabled through PC6 and INTEB is enabled or disabled through PC2.

Mode 2 Strobed Bidirectional I/O

- In this Mode Port A is A bidirectional I/O port
- Port C Provide the control functions for both directions
- To select this Mode D7 and D8 of the control register should be 1’s (i.e. 11XXXXXX).

Setup of Mode 2

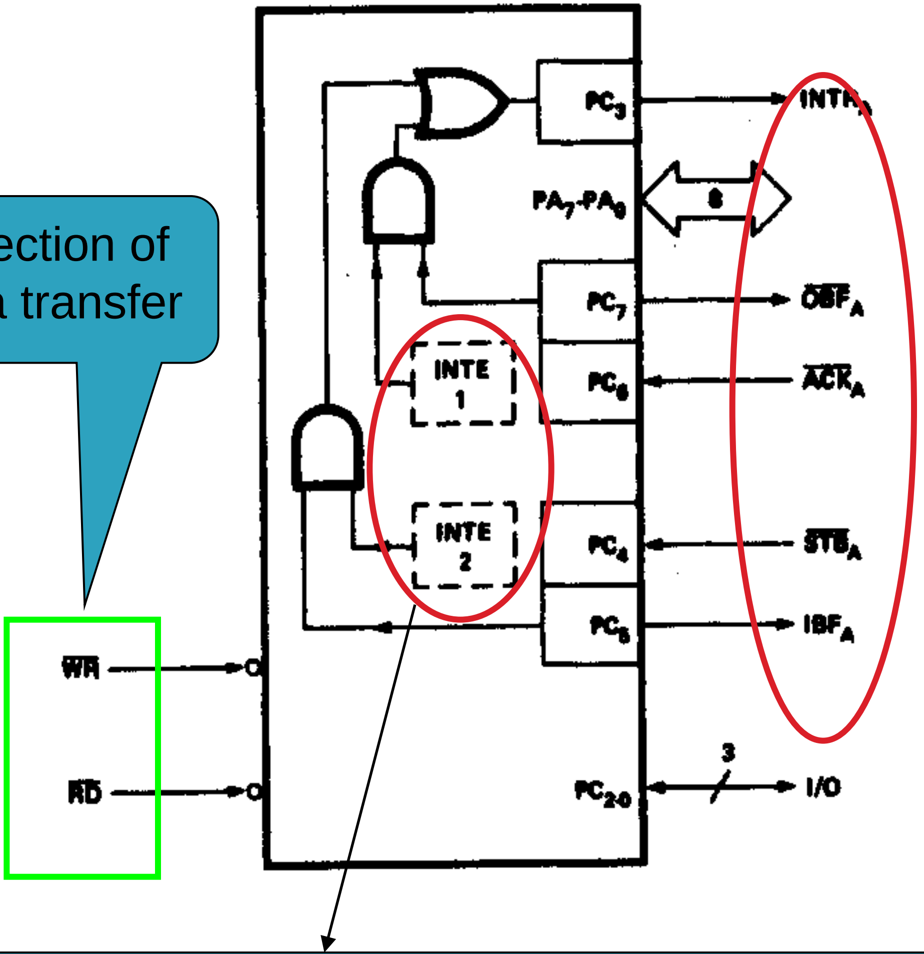


Pin	MODE 2
	GROUP A ONLY
PA_0	↔
PA_1	↔
PA_2	↔
PA_3	↔
PA_4	↔
PA_5	↔
PA_6	↔
PA_7	↔
PB_0	—
PB_1	—
PB_2	—
PB_3	—
PB_4	—
PB_5	—
PB_6	—
PB_7	—
PC_0	I/O or $INTR_B$
PC_1	I/O or $\overline{OBF}_B$ or $\overline{IBF}_B$
PC_2	I/O or $\overline{ACK}_B$ or $\overline{STB}_B$
PC_3	$INTR_A$
PC_4	$\overline{STB}_A$
PC_5	IBF_A
PC_6	$\overline{ACK}_A$
PC_7	$\overline{OBF}_A$

MODE 0
OR MODE 1
ONLY

Mode 0 or 1
(port B)

Mode 2 (configuration of port A)



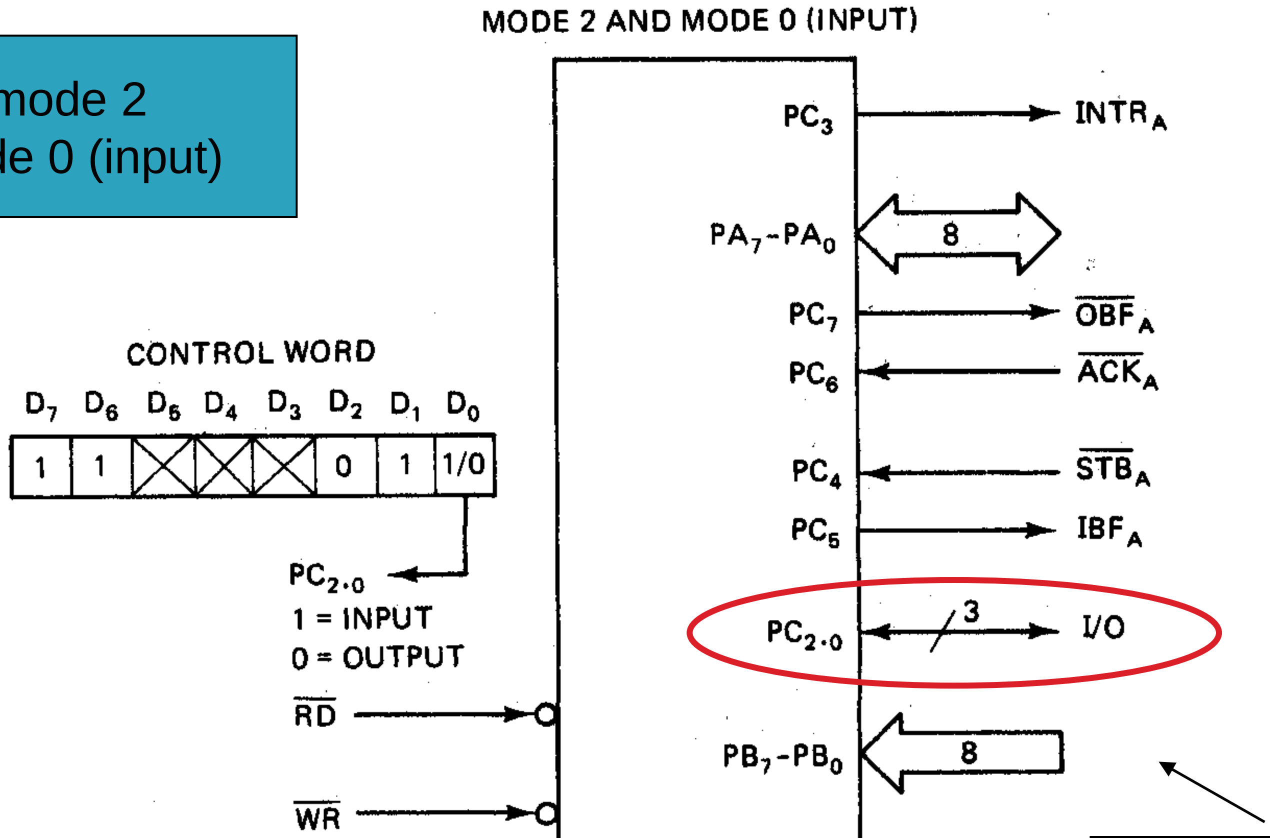
Direction of data transfer

Port A (mode 2) control signals

Port A-Input: INTE1=1 by setting PC4
Port A-Output: INTE2=1 by setting PC6

Combined Modes

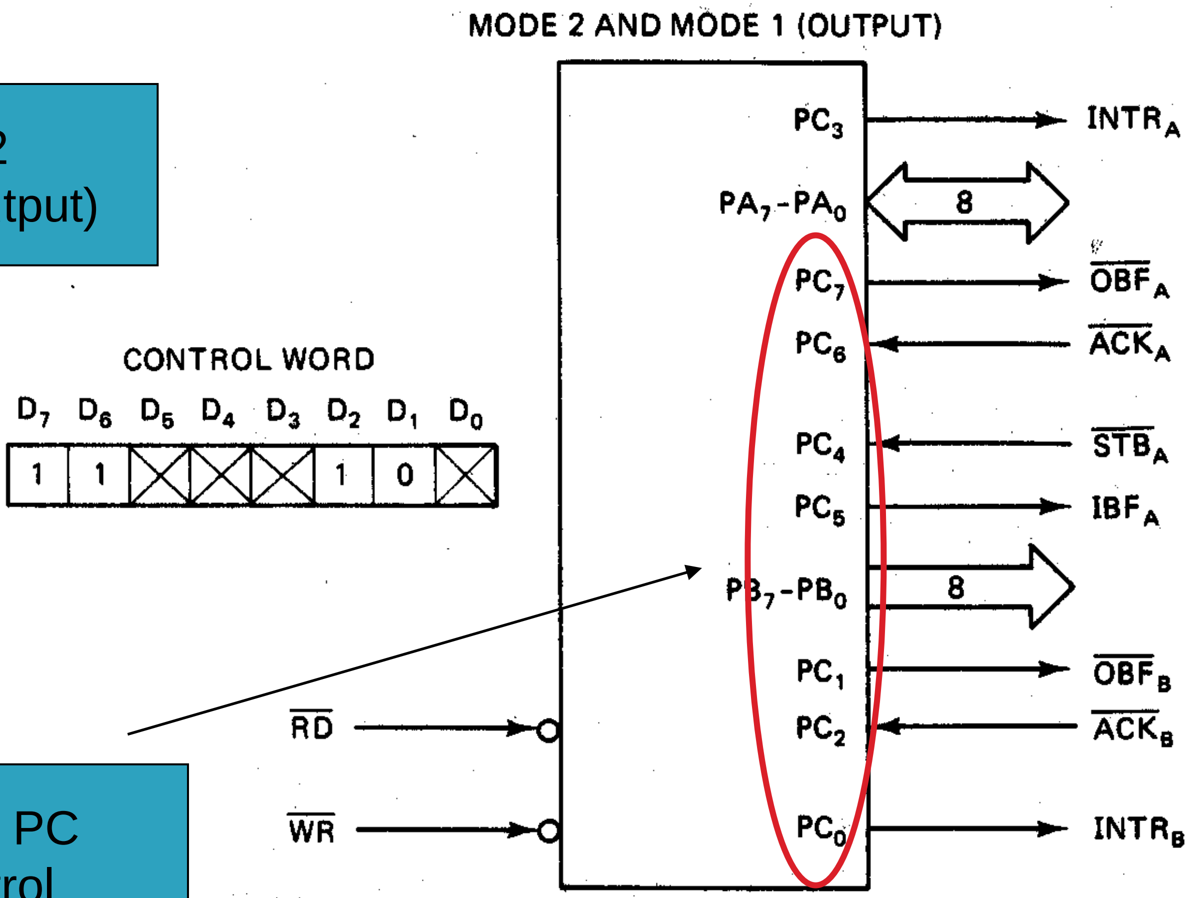
Port A-mode 2
Port B-mode 0 (input)



Unused bits of port PC; they can be used for general I/O of single or group of signals

Combined Modes (cont'd)

Port A-mode 2
Port B-mode 1 (output)

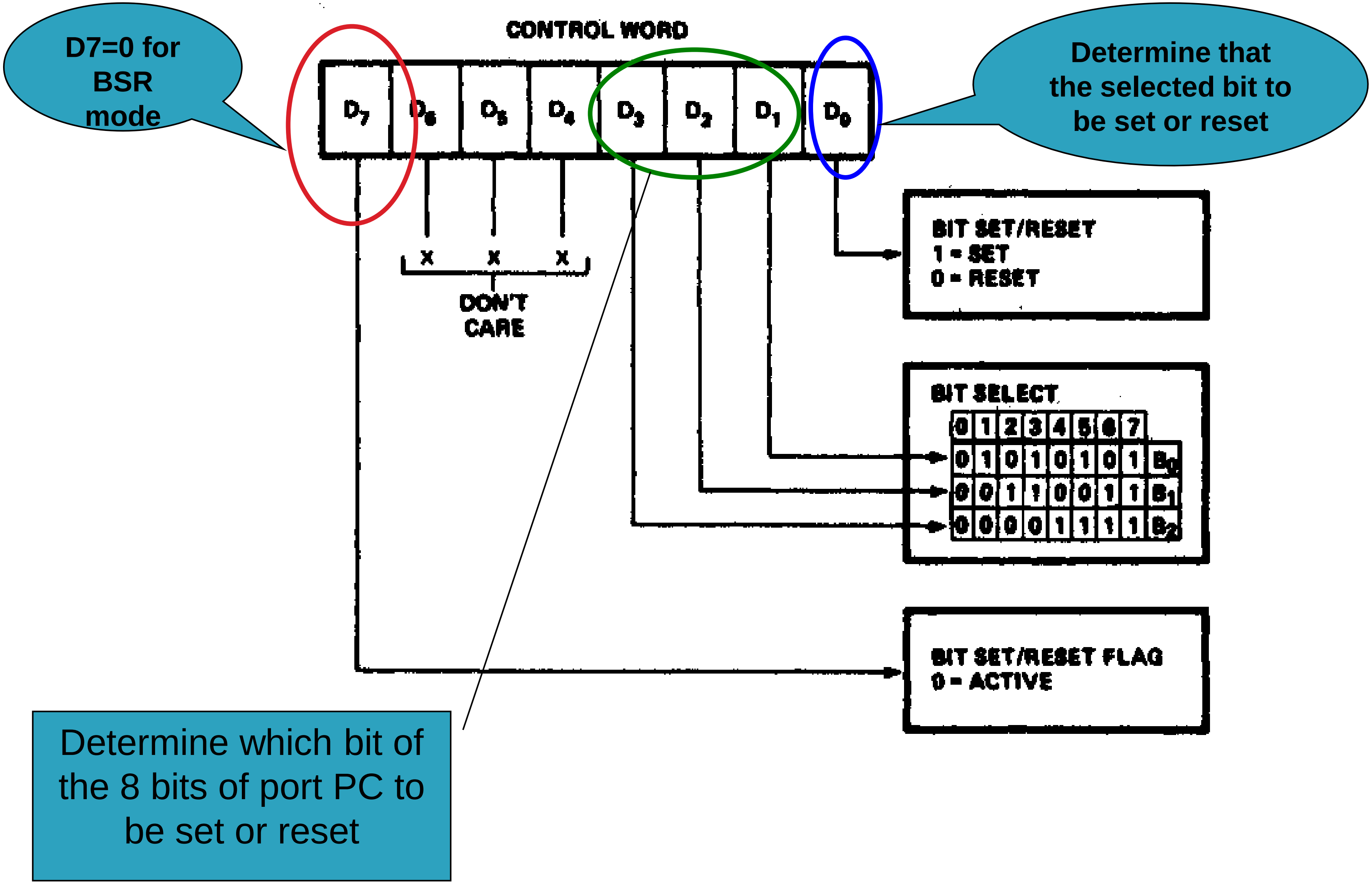


CONTROL WORD

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	☒	☒	☒	1	0	☒

Note that all bits of PC
are used for control

Bit set/reset Mode (BSR mode)



D7=0 for BSR mode

Determine that the selected bit to be set or reset

BIT SET/RESET
1 = SET
0 = RESET

BIT SELECT

0	1	2	3	4	5	6	7	
0	1	0	1	0	1	0	1	B_0
0	0	1	1	0	0	1	1	B_1
0	0	0	0	1	1	1	1	B_2

BIT SET/RESET FLAG
0 = ACTIVE

Determine which bit of the 8 bits of port PC to be set or reset

Example 1: BSR mode

The interrupt control flag INTEA for port A is controlled by PC6. Using te BSR mode of 8255A. What configuration code must be written to the control register to set it to enable this control flag?

Solution:

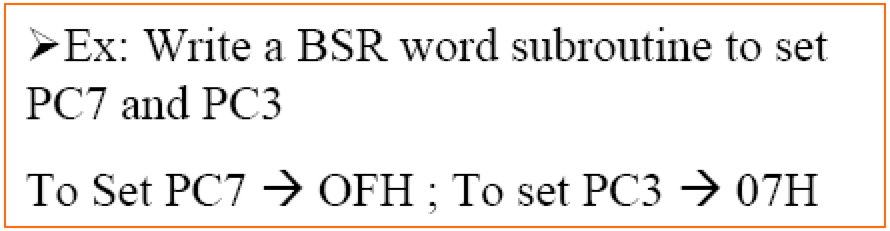
1. D7=0
2. INTEA is to be set, hence, D0=1
3. To select PC6 then D3 D2 D1=110.
4. The remaining bits are don't care.

So…..

Control register=0XXX1101 or 00001101.

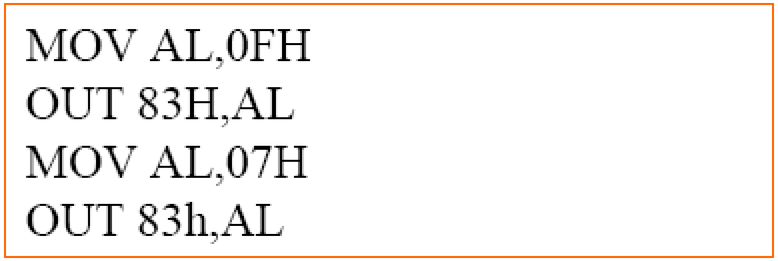
Example 2: BSR mode

Assume that the 8255 is mapped to the address 0080H in the I/O space:



➢Ex: Write a BSR word subroutine to set PC7 and PC3

To Set PC7 → OFH ; To set PC3 → 07H



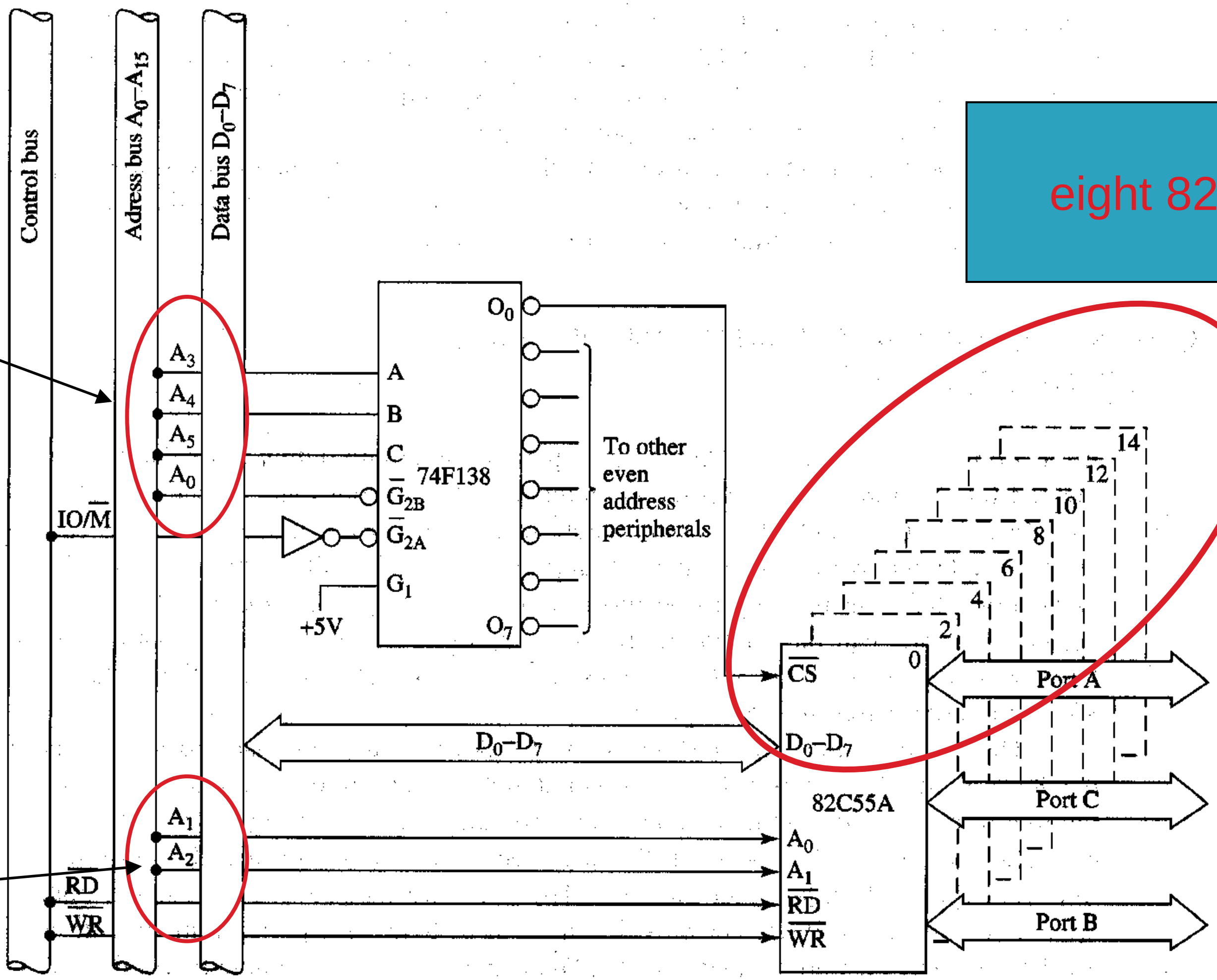
```
MOV AL,0FH
OUT 83H,AL
MOV AL,07H
OUT 83h,AL
```

Note that we sent two different control words to the control register to set, respectively, PC7 and PC3.

8255 implementation of parallel I/O ports

To select one 8255-PPI

eight 8255-PPI



select one port in 8255-PPI

- 8255 parallel I/O ports in 8088 based Microcomputer.

8255 implementation of parallel I/O port

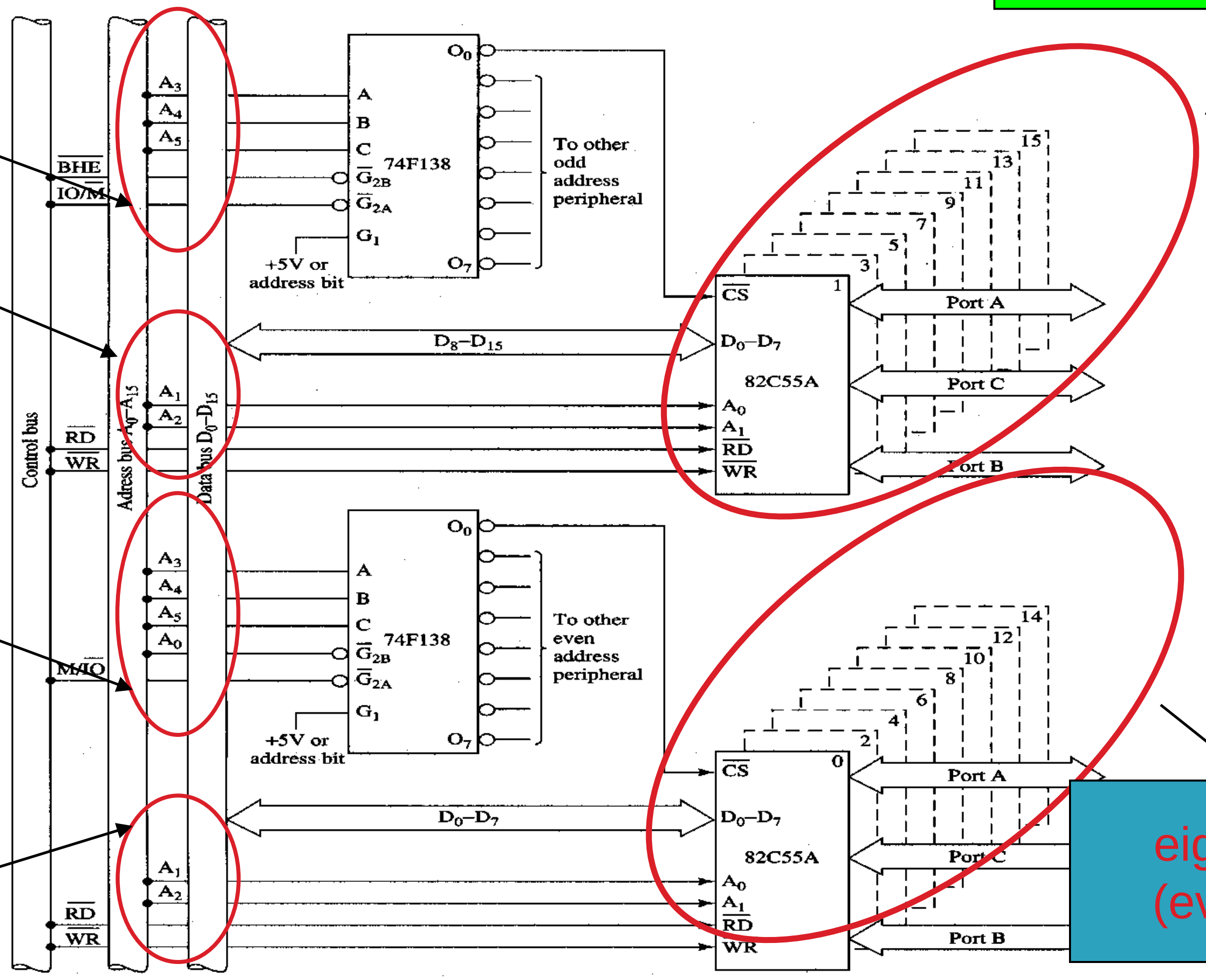
eight 8255-PPI
(odd address)

To select one
8255-PPI

To select one
8255-PPI
(odd address)

To select one
8255-PPI

To select one
8255-PPI
(even address)



eight 8255-PPI
(even address)

- 8255 parallel I/O ports in 8086 based Microcomputer.

Example 1:

- What must be the address bus inputs of the circuit shown in Fig 10.21 if port C of PPI 14 is to be accessed?

- Answer:

1. To enable PPI 14, the decoder 74138 must be enabled and O7 must be 0 (active), G2B=0 and CBA=111.
2. A_0=0 to enable decoder (74138) and $A_5A_4A_3$=111
3. Port C of PPI is selected A1A0=10 or (A2A1=10 from the bus)
4. The rest of addresses are don’t care

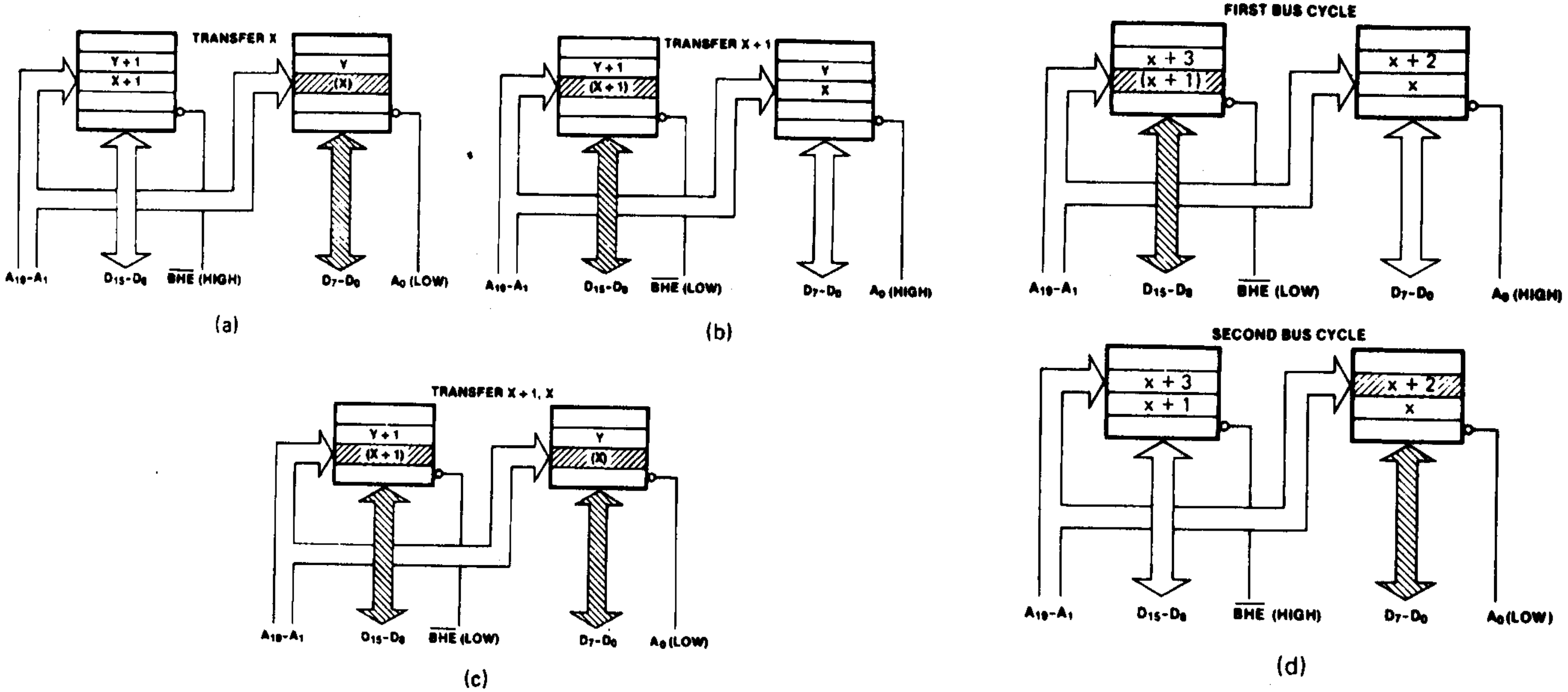
Example 2:

- Assume that PPI 14 of Fig 10.21 is configured so that Port A is output and C and B are inputs. All in mode 0. Write a program that input data from ports C and B and then find the difference between C and B (PC-PB) and then output it to port A ?

- Answer:

1. To enable PPI 14, the decoder 74138 must be enabled and

Memory-processor data transfer



(a)

(b)

(c)

(d)

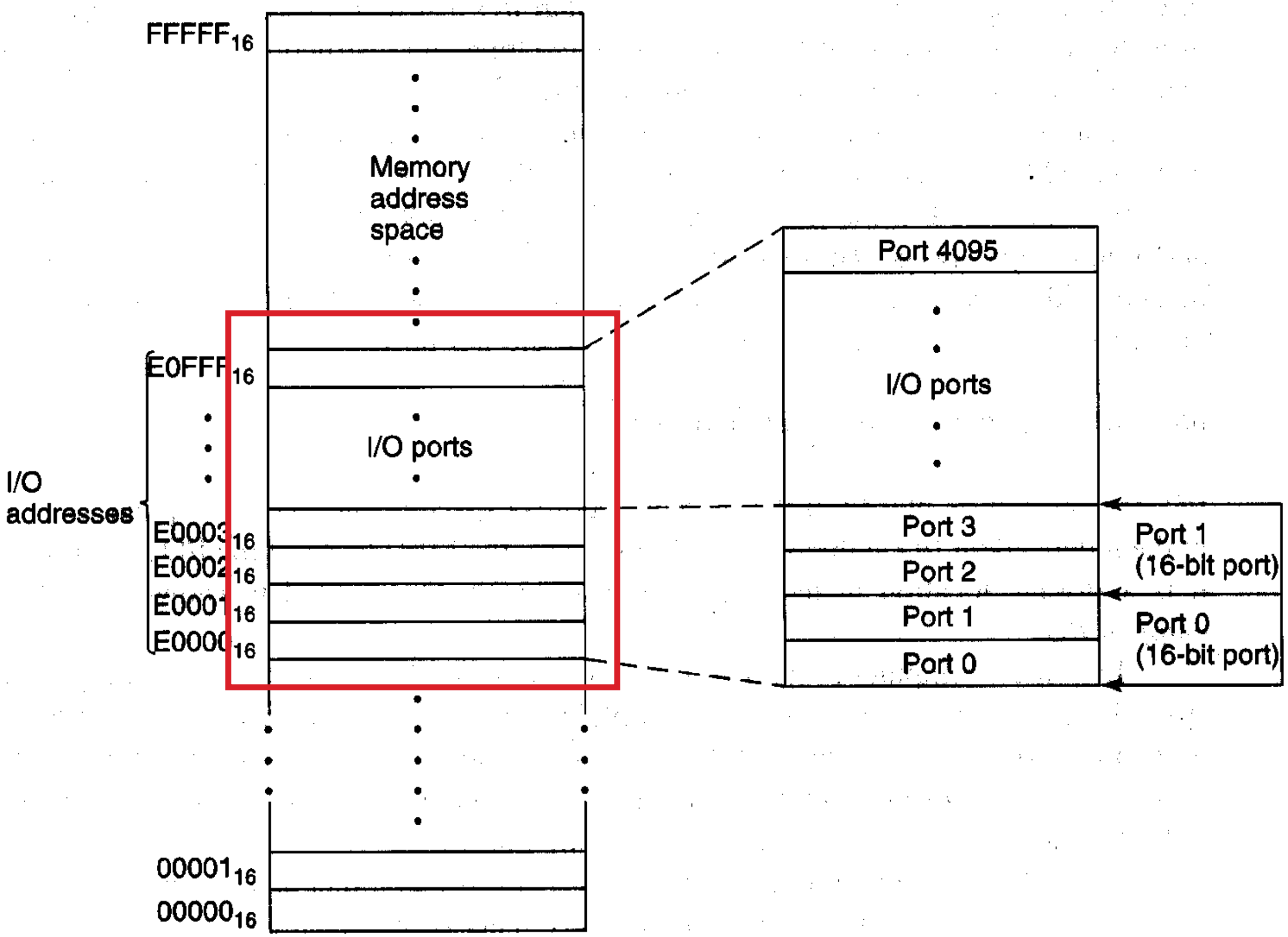
Misaligned word

Memory mapped I/O

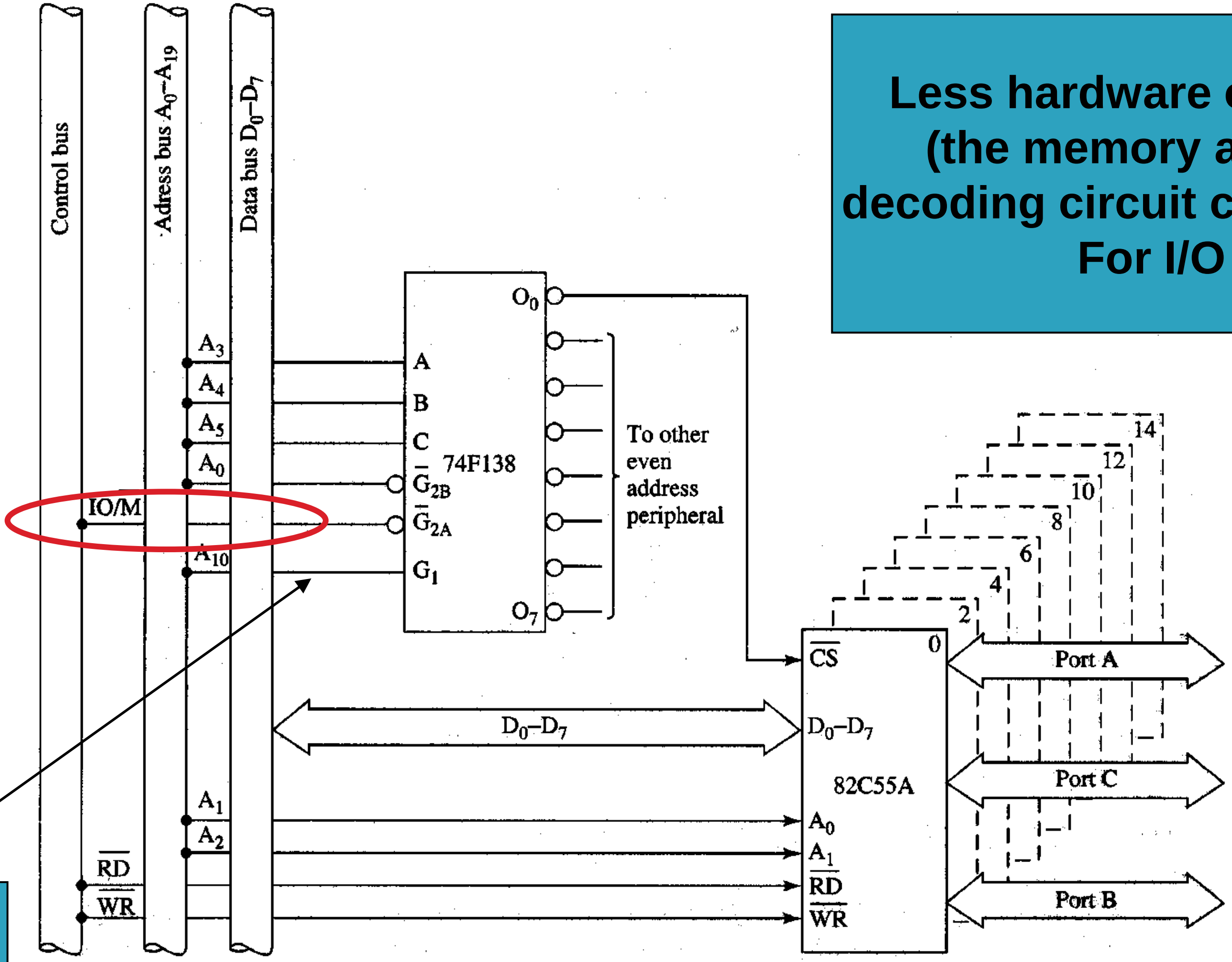
- I/O devices can be mapped to the memory address space.
- MPU looks at the I/O port as a storage location in memory.
- In micro-computer with M-M I/O, some memory addresses are dedicated for I/O port.

- Advantages:
  - Instruction that affect data in memory are used for I/O instead of special I/O instruction.
  - Hence, much more instructions and addressing modes are available for I/O operation.
  - For example data transfer can be performed not only with AL and AX but also with the other internal registers.
- Disadvantages:
  - Slower operations than those specially designed for I/O
  - Part of memory space is lost.

Memory-mapped I/O



Memory-mapped I/O



Less hardware overhead (the memory address decoding circuit can be used For I/O

Not inverted

Why????

Why A10 not Vcc?

Example 1 for M-M I/O:

- Which I/O port in Fig 10.23 is selected for operation when Memory address bus contains 00402H

- Answer:

1. Convert to binary: 00000000010000000010
2. Then, we find, A10=1 and A0=0 and IO/Mprime=0 (memory operation)
3. Now, A5A4A3=000, then PPI 0 is selected
4. Moreover, A2A1=01, then Port B of PPI 0 is selected.

Example 2 for M-M I/O:

- Write a sequence of instruction to initialize the control register of PPI 0 in Fig 10.23 so Port A is output and ports B and C are inputs, all ports in mode 0.

- Answer:

1. Control register=10001011=8BH
2. Memory address of PPI0=00000000010000000110=00406
3. Code:

```
MOV AX, 0
MOV DS, AX
MOV AL, 8BH
MOV [406H], AL
```

Example 3 for M-M I/O:

- Assume the same configuration of Example 2. Write a sequence of instruction so that the content of B and C are ANDED then output the result to port A.

- Answer:

1. Address of the ports: address port A=00400, address port B=00402, and address port C=00404.
2. Code:

```
MOV AX, 0
MOV DS, AX
MOV AL, 8BH
MOV [406H], AL
MOV BL, [402H]
MOV AL, [404H]
AND AL, BL
MOV [400H], AL
```