

# **8255A Programmable Peripheral Interface (PPI)**

By

Dr. Sikder Sunbeam Islam

Dept. of EEE. IIUC

# Introduction to 8255A

- Intel 8255A is a general purpose parallel I/O interface IC. It is also called as programmable *peripheral input-output port*.
- PPI makes an inter-relation between microprocessor and slower peripheral devices.
- It has three I/O ports :Port A, Port B and Port C.
- The **process of establishing** inter relation between slower peripheral device and microprocessor is called *handshaking*.
- 8255-based devices performs handshaking using some handshaking signals like, Strobe Input (STB), Acknowledged (ACK), Input Buffer Full (IBF), Output Buffer Full (OBF).

# Parallel Data Transfer

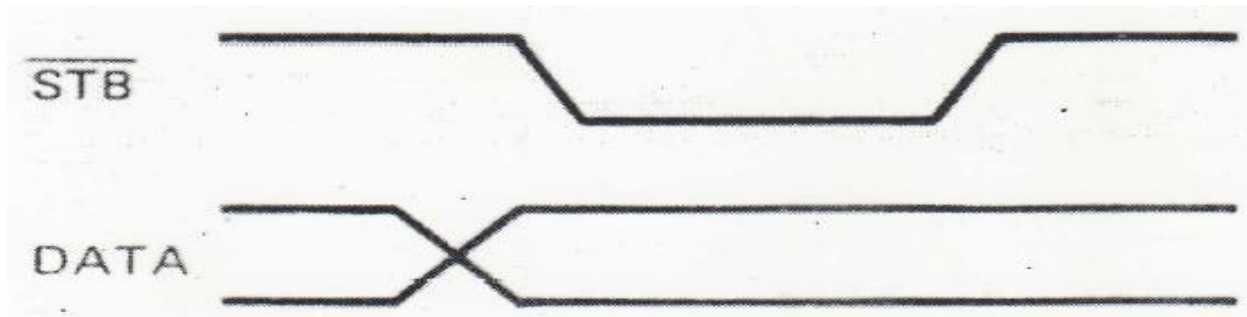
- To understand the operation of 8255A with microprocessor it would be better to know the methods of parallel data transfer . There are four parallel data transfer methods:
- ***Simple I/O:***
- This data transfer method is used when the I/O devices **need no communication before the data transfer.** Such devices are thermostat, LED.
- The crossed lines on the wave form represent the time at which a new data byte becomes valid on the output lines of the port.



simple  
(no handshake)

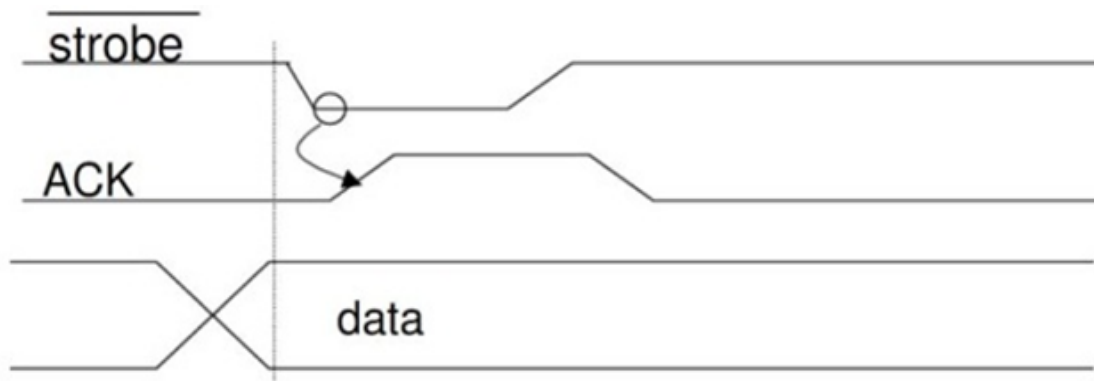
# Parallel Data Transfer(Continued)

- *Simple Strobe I/O:*
- The sending device, such as a keyboard, outputs parallel data on the data lines, and then outputs an  $\overline{STB}$  signal to let the receiving device know that valid data is present.



# Parallel Data Transfer(Continued)

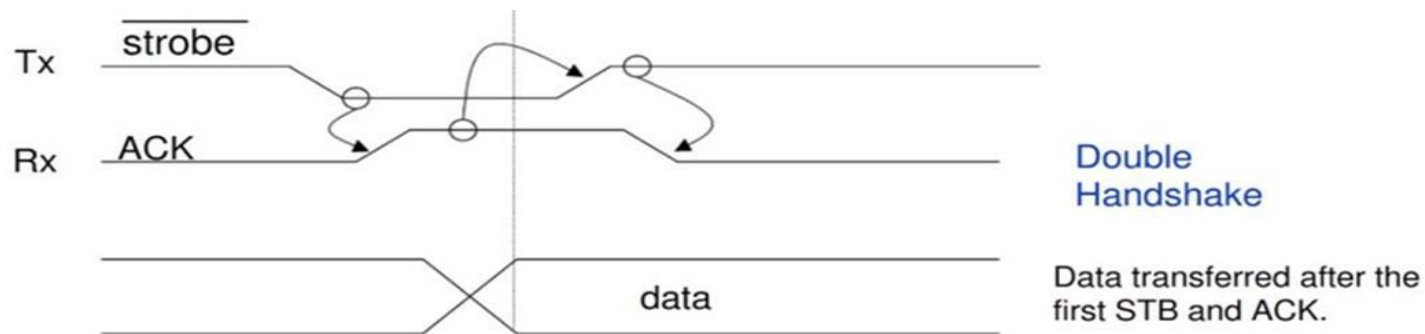
- *Single handshake Data transfer:*
- The sending device outputs some parallel data and sends an  $\overline{STB}$  signal to the receiving device.
- After detecting the  $\overline{STB}$  signal receiving device as a response reads in the data and send an acknowledge signal to indicate that the data has been read.
- The sending device is designed so that it does not send the next data byte until the receiving device indicates with an ACK signal.



Single  
Handshake

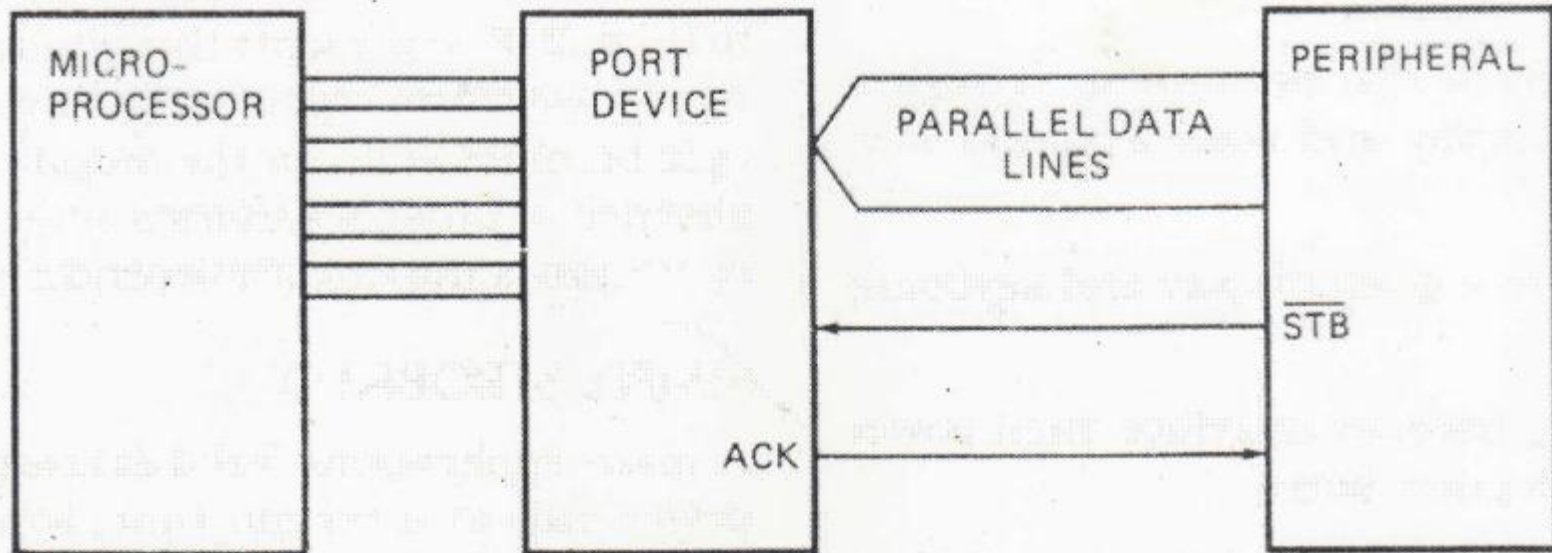
# Parallel Data Transfer(Continued)

- ***Double Handshake Data Transfer:***
- The sending device asserts its  $\overline{STB}$  line low to ask the receiving device “are you ready?”
- The receiving device raises its ACK line high to say “I am ready”.
- The peripheral device then sends the byte of data and raises its  $\overline{STB}$  line high to say “Here is some valid data for you”.
- After it has read in the data, the receiving data drops its ACK line low to say “I have the data”. The receiving device is then ready to be requested for accepting the next data byte.

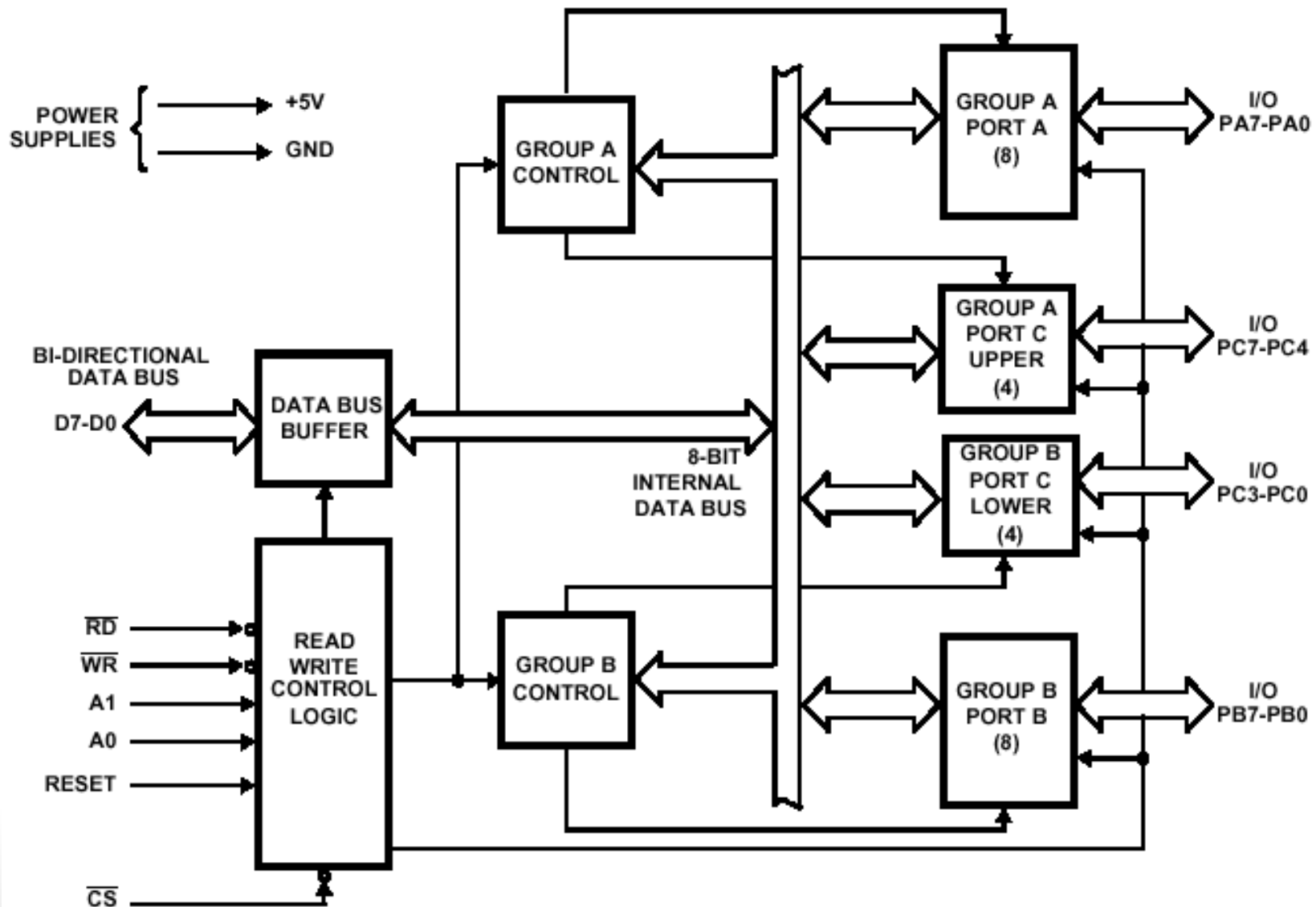


# Handshaking & Parallel Data Transfer

$\overline{STB}$  and ACK signal for these hand shake transfers can be produced on a port pin by instructions in the program. However, this method takes too much processing time. So, parallel port device(like, 8255A) is needed and is connected as Figure bellow.



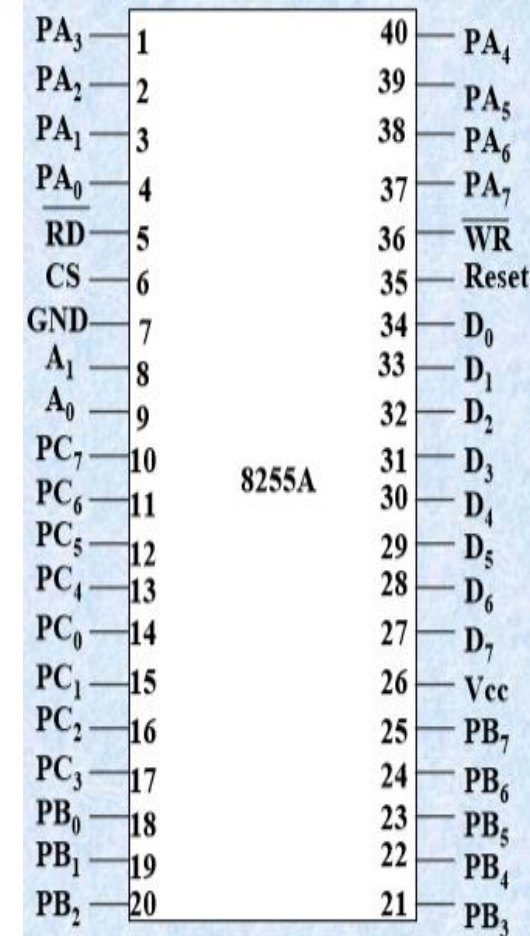
# 8255A Internal Block Diagram





# 8255A Block Diagram Description

- **I/O Port Pins:** 24 ( *Port A* = 8, *Port B* =8, *Port C* =8)
- **Control Pins:** 6 ( $\overline{\text{RD}}$ ,  $\overline{\text{WR}}$ ,  $\overline{\text{CS}}$ , RESET, A1, A0)
- **Data Lines:** 8
- **Power Supply:** 2 (VCC, GND)
- **Total 40 pins**
- **Port C:** -8 bit I/O port is divided into two 4-bit ports or to produce handshake signals for Port A and Port B.
- **Port A and upper 4 bits of Port C** are controlled by **Group A**; Port B and lower part of Port C are controlled by Group B.
- **Eight Data Lines:** Write data bytes to a port or control register and read bytes from port under control of RD and WR lines.
- **A0 and A1:** These input signals, control the selection of one of the three ports or the control word register. →
- **Data Bus Buffer:** This Bi-directional 8-bit Buffer is used to interface the 8255 to the system data bus.
- **RESET:** It is connected to system reset lines. A "high" on **this input initializes** the control register to **9Bh** and all ports are set to the input mode.



8255A Pin Configuration

A0	A1	Selection
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Control

# 8255A Operation Modes

## Mode 0

- Port A works as simple input or output without handshaking.
- Port B works as simple input or output without handshaking.
- Port C can be used together as an additional 8 bit port or they can be used individually as two 4-bit ports.
- When used as outputs, the Port C lines can be individually set or reset by sending a special control word to the control register address.

# 8255A Operation Modes (Continued)

## Mode 1

- Used for handshake input/output operation.
- Port B is initialized in mode 1 for either **input or output**, Pins PC0, PC1 and PC2 function as handshake lines.
- If port A is initialized in mode 1 as handshake **input** port, then pins PC3, PC4 and PC5 function as handshake signals. (PC6 and PC7 are available for using as input lines or output lines)
- If port A is initialized as handshake **output** port, then PC3, PC6 and PC7 function as handshake signals. (PC4 and PC5 are available for using as input or output lines)

# 8255A Operation Modes (Continued)

## Mode 2

- Only port A can be initialized in mode 2.
- In mode 2, port A can be used for “bi-directional handshake” data transfer i.e. data can be input or output on the same eight lines.
- Pins PC3, PC4, PC5, PC6, PC7 used as handshake lines for port A.
- Port B is operating in either mode 0 or mode 1.
- If port B is in **mode 0**, then PC0, PC1 and PC2 used for I/O.
- If port B is in **mode 1**, then PC0, PC1 and PC2 used as handshake lines.

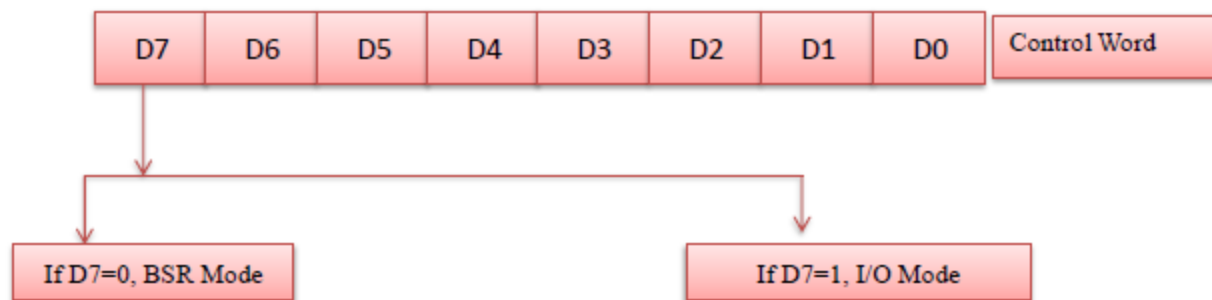
# 8255A Operation Modes (Continued)

## 8255A modes summarization

Mode	8- Bit Port A	8-Bit Port B	Handshake lines for mentioned Mode	Other Port C pins available for I/O
<b>0</b>	I/O	I/O	N/A	PC7- PC4 and PC3- PC0
<b>1</b>	Input	I/O	PC3, PC4, PC5 (Port A) PC0, PC1, PC2 (Port B)	PC6, PC7
<b>1</b>	Output	I/O	PC3, PC6, PC7 (Port A) PC0, PC1, PC2 (Port B)	PC4, PC5
<b>2</b>	Bi-Directional	N/A	PC3, PC4, PC5, PC6, PC7	PC0, PC1, PC2 (If Port B in Mode 0)

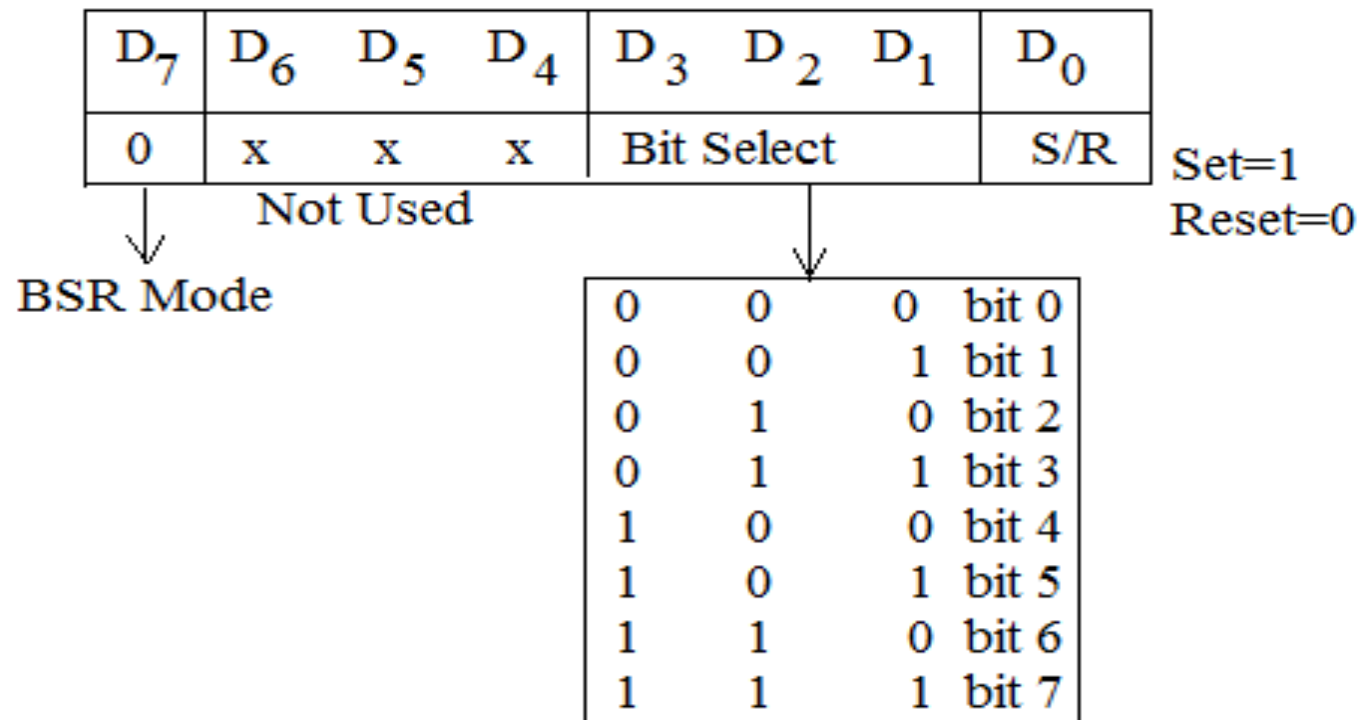
# 8255A Control Word

- There are two types of control words:
- 1) **Mode definition control word:** To tell the device what modes we want the ports to operate in.
- 2) **Bit Set/Reset control word:** When we want to set or reset the output on a pin of *Port C*.
- Or, when we want to enable interrupt output signals for handshake data transfer.
- Both control words are sent to the control register of 8255A.
- Control Word is an 8-bit data that stored in control register. Control Words are two types: (a) BSR Control Word (b) Mode definition Control Word.



# Port C Bit Set/Reset (BSR) Control Word

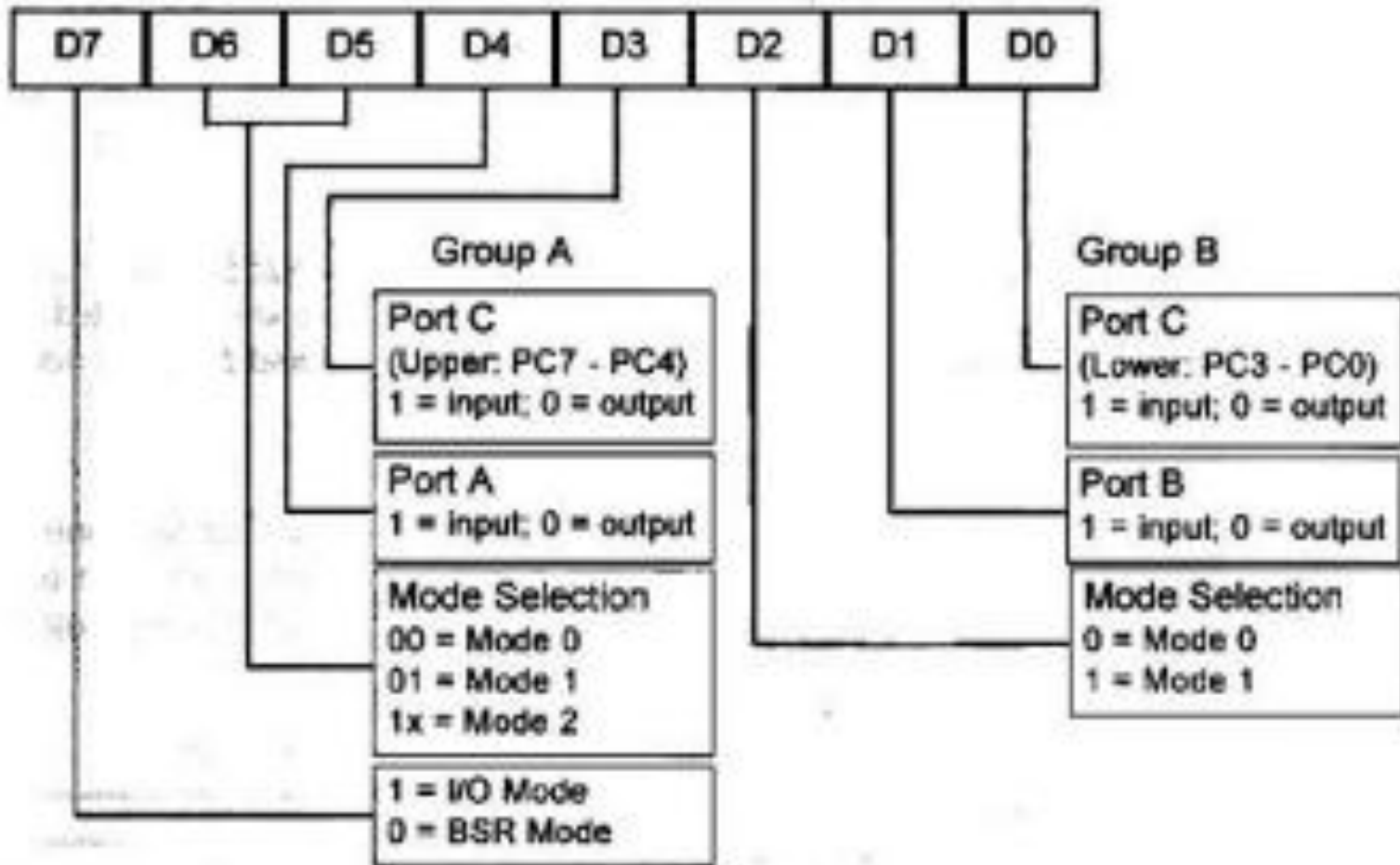
- If bit D7 of control word is set as logic 0 then 8255 will be configured as BSR (Bit Set Reset) mode.
- In this mode we can set or reset the pins of Port C by selection.



**N.B:** Don't Cares are  
Generally set as zero.

# Mode Definition Control Word

- If bit D7 of control word is set as logic 1 then 8255 will be configured as Mode Definition mode.



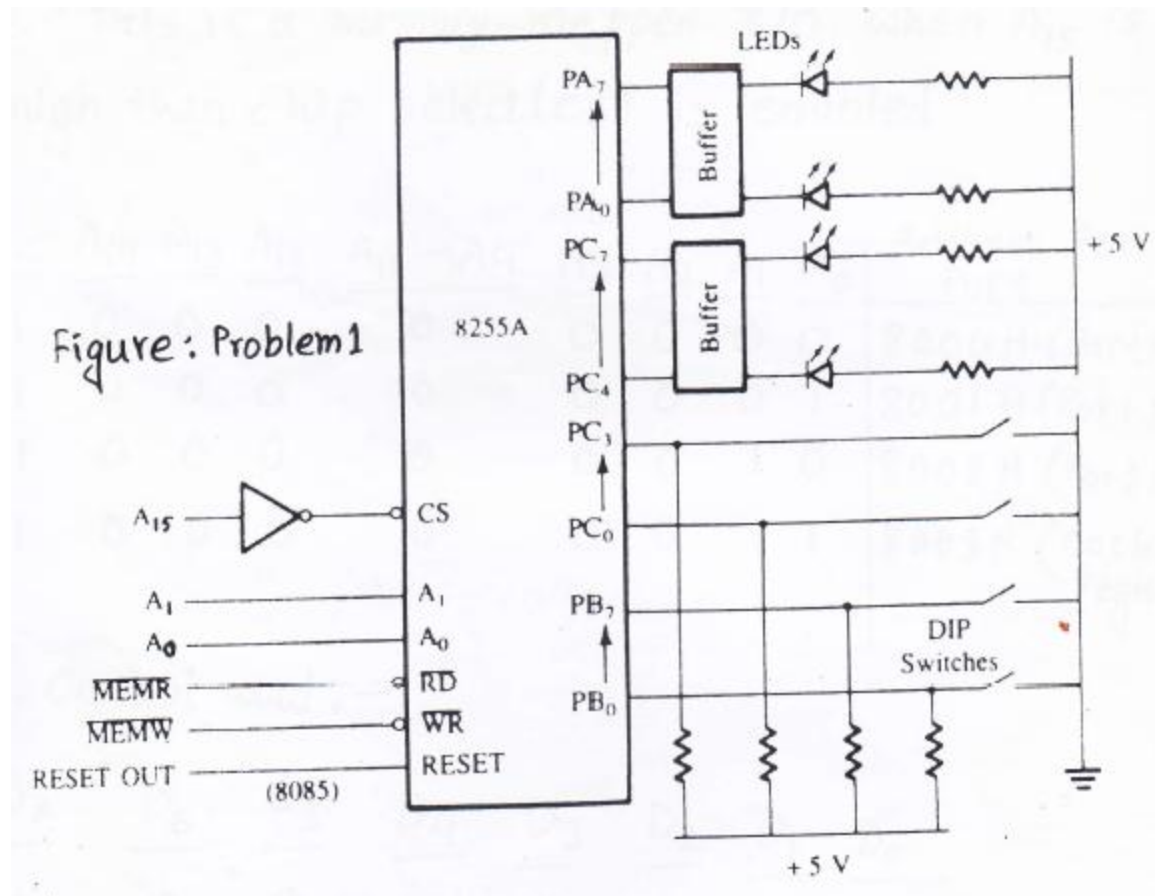


**Problem 1:** Figure shows an 8255A interfaced with 8086 microprocessor. Perform the following-

(a) Identify the Port Address.

(b) Identify the Mode 0 control word to configure Port A and Port C<sub>U</sub> as output ports and Port B and Port C<sub>L</sub> as input ports.

(c) Write a program to read the DIP switches and display the reading from Port B at Port A, and from Port C<sub>L</sub> at Port C<sub>U</sub>.

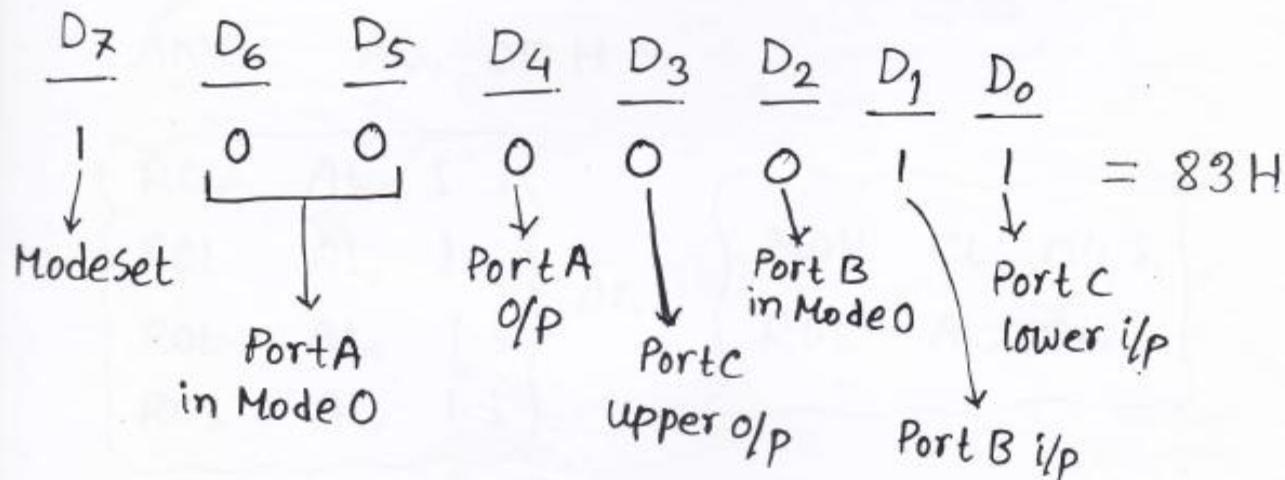


## Solution:

1/ This is a memory-mapped I/O. when  $A_{15}$  is high then chip select ( $\overline{CS}$ ) is enabled.

$A_{15}$	$A_{14}$	$A_{13}$	$A_{12}$	$A_{11} - A_4$	$A_3$	$A_2$	$A_1$	$A_0$	Address for Port
1	0	0	0	0	0	0	0	0	8000 H (Port A)
1	0	0	0	0	0	0	0	1	8001 H (Port B)
1	0	0	0	0	0	0	1	0	8002 H (Port C)
1	0	0	0	0	0	0	1	1	8003 H (Control register)

2. Control word:—



### 3/Program:-

```
PPICR    EQU    8003H ;  
PPIC      EQU    8002H ;  
PPIB      EQU    8001H ;  
PPIA      EQU    8000H ;  
MOV       AL, 83H ;  
OUT        PPICR, AL ;  
IN         AL, PPIB ;  
OUT        PPIA, AL ;  
IN         AL, PPIC ;  
AND        AL, 0FH ;
```

```
{ ROL    AL, 1 ;  
  ROL    AL, 1 ;  
  ROL    AL, 1 ;  
  ROL    AL, 1 ; }
```

or,

```
{ MOV    CL, 04 ;  
  ROL    AL, CL }
```

```
OUT PPIC, AL ;  
HLT
```

**Problem 2:** An 8086-8255 based microcomputer is required to drive an LED connected to bit 2 of Port B based on two switch inputs connected to bit 6 and 7 of port A. If both switches are either high or low, LED will turn **on**; otherwise, it will remain **OFF**. Assume base address of 60H. Write an 8086 assembly language program to accomplish this.

### Solution:

```
PORTA EQU 60H  
PORTB EQU 61H  
CNTRL EQU 63H
```

```
MOV AL, 10010000B; Configure Port A as input and Port B as output.  
OUT CNTRL, AL;
```

```
MAIN : IN AL, PORTA;  
      AND AL, 11000000B;  
      JPE LEADON; (JPE= Jump if parity even, p=1)  
      MOV AL, 00H;  
      OUT PORTB, AL;  
      JMP MAIN;
```

```
LEDON : MOV AL, 00000100B;  
      OUT PORTB, AL;  
      JMP MAIN;
```

**Problem3:** Write a BSR control word subroutine to set bit PC7 and PC3 and reset them after 10ms. Use Fig.1. Also write the delay procedure considering the processor clock at 5MHz.

Solution:

BSR Control word :—

$$\text{To set bit PC}_7 = \begin{matrix} \overline{D_7} & \overline{D_6} & \overline{D_5} & \overline{D_4} & \overline{D_3} & \overline{D_2} & \overline{D_1} & \overline{D_0} \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{matrix} = 0FH$$

$$\text{To reset PC}_7 = 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 = 0EH$$

$$\text{To set PC}_3 = 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 = 07H$$

$$\text{To reset PC}_3 = 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 = 06H$$

Port Address for control register:—

$$\begin{matrix} \overline{A_{15}} & \overline{A_{14}} & \overline{A_{13}} & \overline{A_{12}} & \overline{A_{11}-A_4} & \overline{A_3} & \overline{A_2} & \overline{A_1} & \overline{A_0} \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{matrix} = 8003H$$

$$\text{Duration of 1 clock pulse} = \frac{1}{5\text{MHz}} = 200\text{ ns.}$$

$$\text{So, for 10ms we have to count} = \frac{10\text{ms}}{200\text{ns}} = 50,000$$

Subroutine: Let us assume that PPICR=8003H,  
already declared in the main program.

```
BSR:  MOV  AL, 0FH ;  
      OUT  PPICR, AL ;  
      MOV  AL, 07H ;  
      OUT  PPICR, AL ;  
      CALL DELAY ;  
      MOV  AL, 06H ;  
      OUT  PPICR, AL ;  
      MOV  AL, 0EH ;  
      OUT  PPICR, AL ;  
      RET
```

```
DELAY PROC NEAR
```

```
      MOV  CX, 50000
```

```
      HERE: LOOP HERE
```

```
      RET
```

```
DELAY ENDP
```

# Display Interfacing

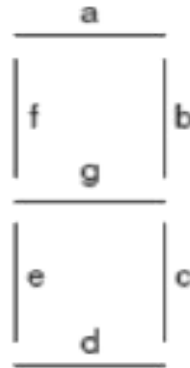
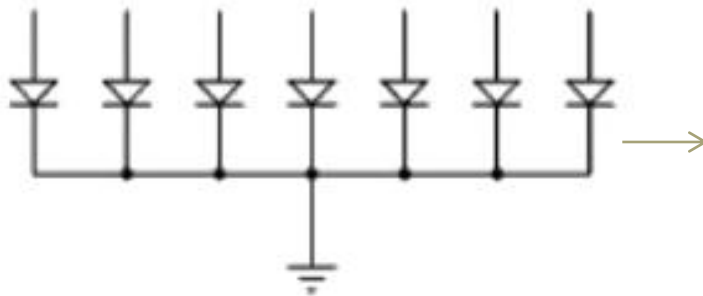


Figure : Segment names of a 7-segment display



Number	x g f e d c b a
0	0 0 1 1 1 1 1 1
1	0 0 0 0 0 1 1 0
2	0 1 0 1 1 0 1 1

Figure : Common cathode configuration



# Exercise:

- A. An 8086-8255 based system is required to drive an LED connected to bit 2 of Port B based on two switch inputs connected to bit 0 and 1 of port A. Write an 8086 assembly language program to perform logical **AND** operation on the LED using the switches. Assume base address of 60H.
- B. An 8086-8255 based system is required to operate an 7-segment display (Common-Cathode) connected to Port A based on two switch inputs connected to bit 0 and 1 of Port B as Fig.B. If both switches are either high or low, display number 0; otherwise, display number 1. Assume base address of 60H. Write an 8086 assembly language program to accomplish this.

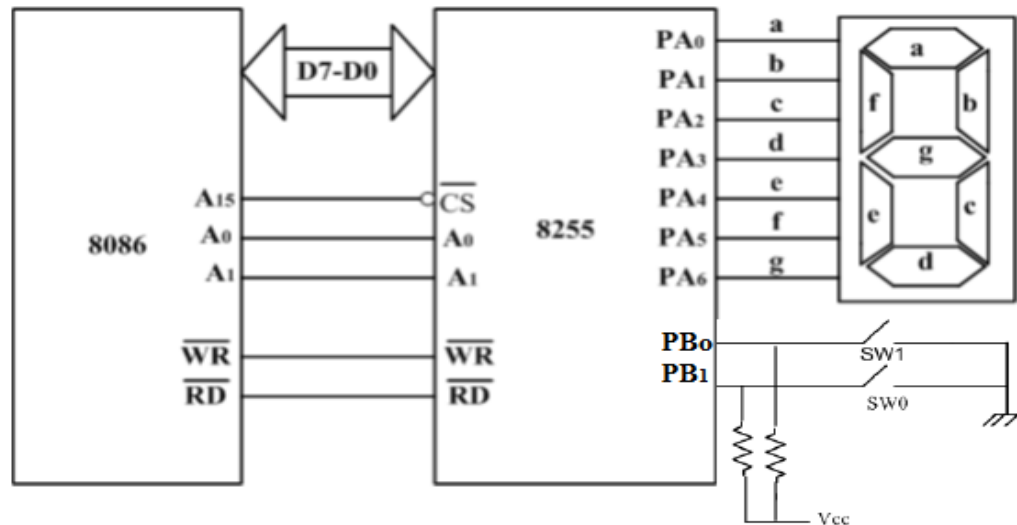


Fig.B



# Exercise:

- C. A 8086-8255A based system is given in Fig.B The Port A of 8255A is connected with a 7-segment display (**Common-Cathode**). Write an assembly language program to display the numbers 2,4,6,8 repeatedly.

# References

- Microprocessor and Interfacing Programming and Hardware; Douglas V Hall, 2<sup>nd</sup> Edition.
- Microprocessors and Microcomputer-Based System Design; Mohammed Rafiquzzaman, revised edition.
- Microprocessor & Interfacing Lecture Materials; Mohammed Abdul Kader, IIUC.
- 8255A PIO; M Krishna Kumar  
[<https://nptel.ac.in/courses/Webcourse-contents/IISc-BANG/> Accessed on: August,2018]