UNIT-3
I/O INTERFACING

8255 PROGRAMMABLE PERIPHERAL INTERFACE

- The parallel input-output port chip 8255 is also called as **Programmable peripheral input-output port.**
- 8255 is mainly used for interface peripheral equipment to the microcomputer system bus.
- 8255 is programmed by the system software so that normally no external logic is necessary to interface peripheral devices.

Features of 8255A

1. The 8255A is a widely used, programmable, parallel I/O device
2. It can be programmed to transfer data under various conditions, from simple I/O to interrupt I/O
3. The Intel's 8255A is designed for use with Intel's 8-bit, 16-bit and higher capability microprocessors.
4. It is completely TTL compatible.
5. It has three 8-bit ports: Port A, Port B, and Port C, which are arranged in two groups of 12 pins. Each port has a 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: (i) Mode 0, (ii) Mode 1, and (iii) Mode 2.
   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 (PC3-PC7). Port B can be used in Mode 0 or in Mode 1.

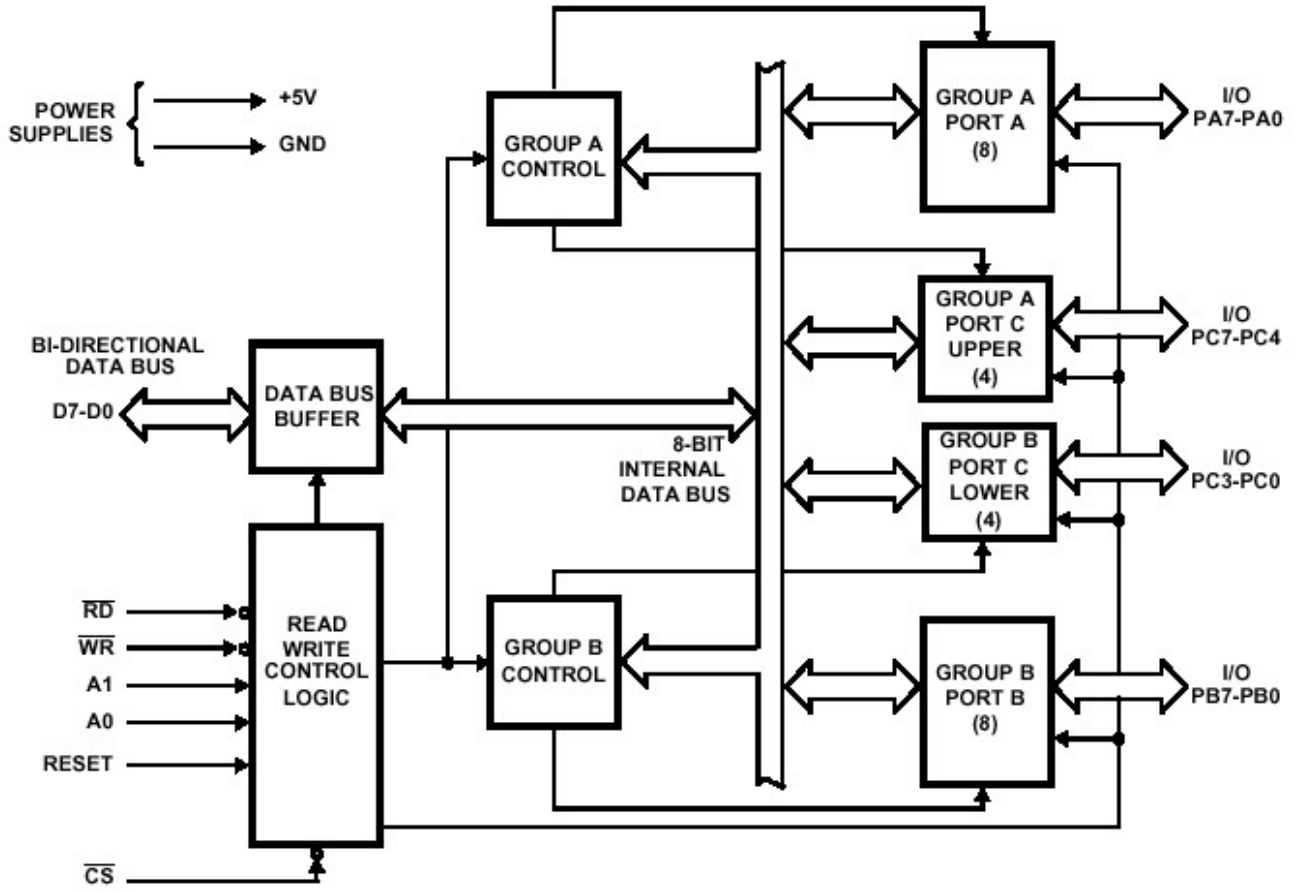
Functional block diagram of 8255 PPI

The functional block diagram of 8255 PPI contains the following blocks

1. Data bus buffer
2. Read Write control logic
3. Group A and Group B controls
4. Port A, Port B and Port C

Description

Data bus buffer: -This is a 8-bit tri-state bidirectional buffer. It is used to interface 8255 to system data bus. It receives or transmits data upon the execution of input or output instructions by the microprocessor. Control word and status information's are also transferred through this unit only. The directions of the data bus are decided by the read or write control signals. When the write signal is activated, it receives data from the system data bus and when the read signal is activated, it transmits data to the system data bus.



Read/Write control logic: -It is the heart of 8255.It manages all of the internal and external transfers of both data and status. It accepts RESET, control signals ($\overline{RD}$, $\overline{WR}$) and also inputs from address bus(A_1, A_0, $\overline{CS}$) and issues commands to individual group of control blocks (Group A, Group B). Every block of the 8255 is connected to this unit.

$\overline{RD}$ and $\overline{WR}$ are connected to $\overline{IOR}$, $\overline{IOW}$ or $\overline{MEMR}$, $\overline{MEMW}$ depending on the types of mapping used.

A_1 and A_0 are directly connected to address lines A_2 and A_1 of the system address lines. The $\overline{CS}$ is connected to chip select decoder; the selection of 8255 is enabled or disabled by $\overline{CS}$ signal.

$\overline{RD}$ and $\overline{WR}$ decide the operation to be performed, i.e. write data to the 8255 PPI or read data from the 8255 PPI.

A high on the RESET input clean the control register and all ports (A, B, C) are set to the input mode
The address lines A_1 and A_0 are used to select one of the three ports or control word register.

A_1	A_0	selected
0	0	PORT A
0	1	PORT B
1	0	PORT C
1	1	CWR

8255 basic operation

A1	A0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	INPUT OPERATION (READ)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
1	1	0	1	0	IILEGAL Operation
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
					DISABLE FUNCTION
X	X	X	X	1	Data Bus → Three-State
X	X	1	1	0	Data Bus → Three-State

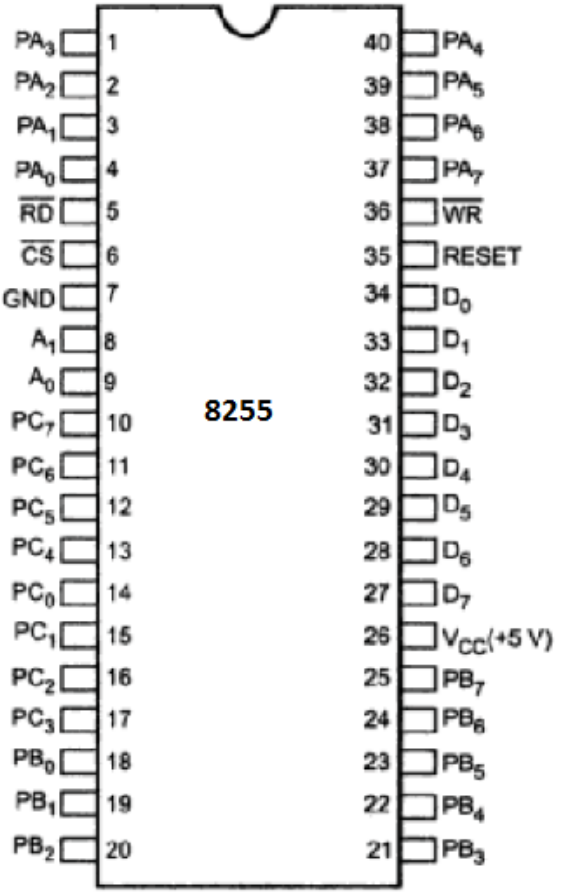
Group A and Group B controls:-
The Ports of 8255 PPI are divided into two groups Group A and Group B
Group A consists of port A and port C upper (PC_7-PC_4) and Group B consists of port B and port C lower (PC_0-PC_3). Each of the control blocks Group A and Group B accepts “commands” from the Read/Write

Control logic, receives “control words” from the internal data bus and issues proper commands to its associated ports.

Port A, Port B and Port C:-

8255 PPI contains three 8-bit ports (A, B and C). The functions of the ports are decided by the control bit pattern available in the control word register and also on the mode operation. **Port A** can be programmed in 3 modes – mode 0, mode 1 and mode 2. **Port B** can be programmed in mode 0, mode1. **Port C** can be divided into two 4 bit ports $\mathbf{PC_U}$ **(**$\mathbf{PC_7}$ **–**$\mathbf{PC_4}$**),** $\mathbf{PC_L}$ **(**$\mathbf{PC_3}$ **–**$\mathbf{PC_0}$**)**. It can be used as control signals (handshaking signals) for port A and port B and it can be programmed as I/O in mode 0.

PIN CONFIGURATION OF 8255:-



Pin Description

SYMBOL	TYPE	DESCRPTION
Vcc		Vcc: The +5V power supply pin. A 0.1µF capacitor between pins 26 and 7 is recommended for decoupling.

GND		GROUND
D0-D7	I/O	DATA BUS: The Data Bus lines are bidirectional three-state pins connected to the system data bus.
RESET	I	RESET. A high on this input clean the control register and all ports (A, B, C) are set to the input mode
$\overline{CS}$	I	CHIP SELECT: Chip select is an active low input used to enable the 8255A onto the Data Bus for CPU communications.
$\overline{RD}$	I	READ: Read is an active low input control signal used by the CPU to read status information or data via the data bus.
$\overline{WR}$	I	WRITE. Write is an active low input control signal used by the CPU to load control words and data into the 8255A
A_0-A_1	I	ADDRESS: These input signals in conjunction with the RD and WR inputs control the selection of one of the three ports or the control word register. A_0 and A_1 are normally connected to the least significant bits of the Address Bus A_1, A_2
PA0-PA7	I/O	PORT A. 8-bit input and output port.
PB0-PB7	I/O	PORT B: 8-bit input and output port.
PC0-PC7	I/O	PORT C: 8-bit input and output port.

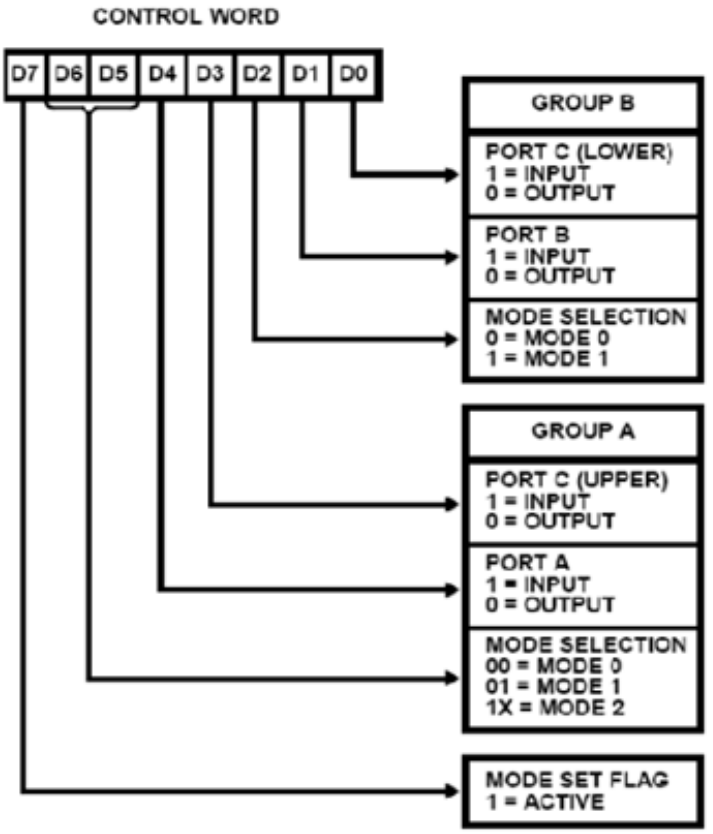
Modes of operation of 8255: -

8255 Operates in 2 basic modes.

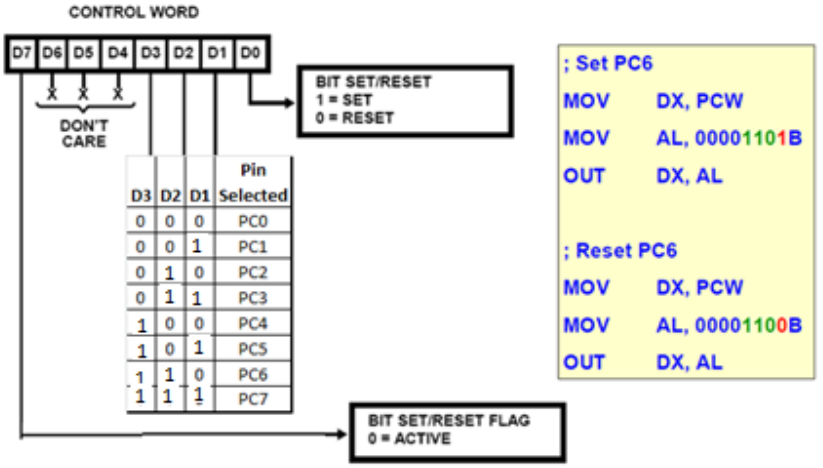
1. Parallel I/O (**I**nput **O**utput) mode
2. BSR (**B**it **S**et **R**eset) mode

The MSB of the Control Word Register of 8255 PPI decides the Mode operation of the 8255. If that bit equals to '1' then 8255 operates in I/O Mode, if the same bit equals to '0' then 8255 operates in BSR (Bit Set Reset) Mode. In I/O Mode or BSR Mode the functionality of the remaining bits CWR are shown in below figures.

1. Control Word Format in I/O Mode Operation of 8255 PPI:-



2. Control Word Format in BSR Mode Operation of 8255 PPI:-



1. BSR Mode operation of 8255: -In this mode any one of the 8-output pins of port C can be set or reset depending on D0 bit of the control word. The Port C bit that is to be set or reset is selected by bit select bits D3, D2 and D1of the CWR. The bits D6, D5 and D4 are don't care bits for BSR mode and for simplicity and compatibility with future products; we make these 3 bits of the CWR as 0's.

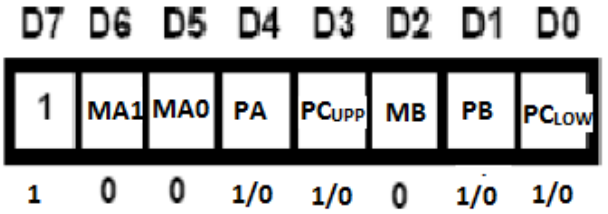
2. Parallel I/O Modes of operation: -The 8255 PPI can operate in three basic input/output modes: mode 0, mode 1 and mode 2. The mode is programmed using the control word register.

a) Mode-0(Basic I/O Mode) Operation: -This mode is also called as simple input/output mode. This mode provides simple input and output capabilities using each of the three ports. Data can be simply read from and written to the input and output ports respectively, after appropriate initialization.

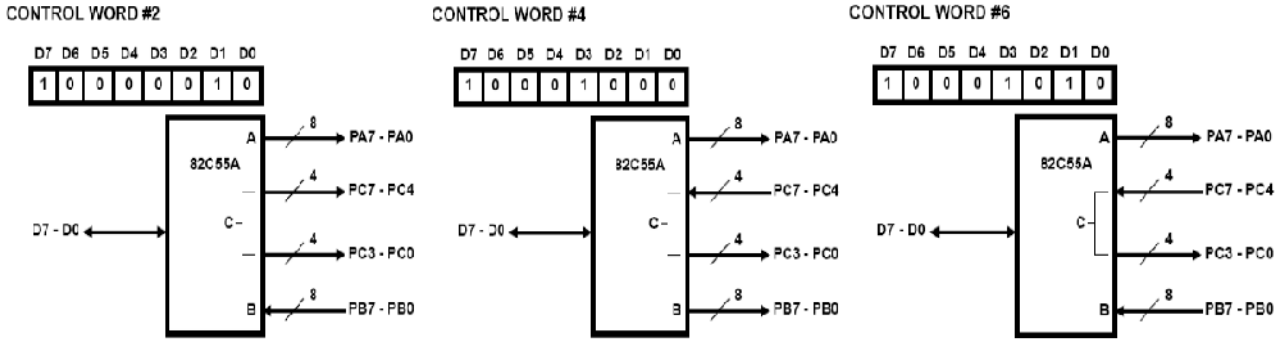
The salient features of this mode are as listed below: -

1. Two 8-bit data ports (port A and port B) and two 4-bit data ports (Port C upper and lower) are available. The two 4-bit ports can be combined and used as a third 8-bit port.
2. Any port can be used as an input or output port. All these modes can be selected by programming a register internal to 8255 known as CWR.
3. If both Port A and Port B are initialized in Mode 0 then the two halves of Port C can be initialized in Mode 0 as separate ports or together as 8-bit port.
4. The two halves of Port C are independent, so one half can be initialized as input and the other half initialized as output.
5. Total Sixteen different input/output configurations are available in this mode.
6. When port C is used as output port, the Port C lines can be individually set or reset by sending a special control word to the CWR address.

In mode 0, Ports are defined as follow:



For example:



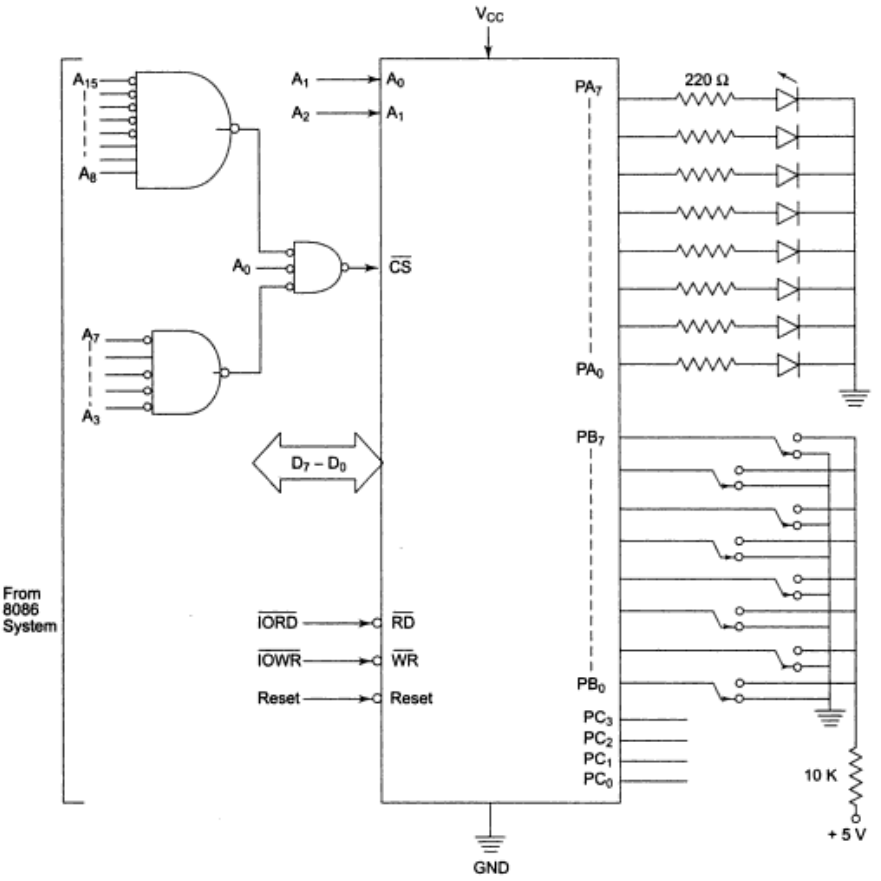
Problem-1

Interface an 8255 with 8086 to work as an I/O port. Initialize port A as output port, port B as input port and port C as output port. Port A address should be 0740H. Write a program to sense switch positions SW0-SW7 connected at port B. The sensed pattern is to be displayed on port A to which 8 LEDs are connected, while the port C lower displays number of ON switches out of the total 8 switches.

Solution The control word is decided upon as follows:

B7	B_6	B_5	B_4	B_3	B_2	B_1	B_0	Control word
1	0	0	0	0	0	1	0	= 82H
I/O	Port A		Port	Port	Port	Port	Port	
mode	in mode 0		A,o/p	C,o/p	B,mode 0	B,i/p	C,o/p	

8255 Ports	I/O Address lines A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_{09}	A_{08}	A_{07}	A_{06}	A_{05}	A_{04}	A_{03}	A_{02}	A_{01}	A_{00}	Hex. Port Addresses
PortA	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0740H
Port B	0	0	0	0	0	1	1	1	0	1	0	0	0	0	1	0	0742H
Port C	0	0	0	0	0	1	1	1	0	1	0	0	0	1	0	0	0744H
CWR	0	0	0	0	0	1	1	1	0	1	0	0	0	1	1	0	0746H



Program:

```
        MOV AL, 82H
        MOV DX, 0746H
        OUT DX, AL
AGAIN: MOV DX, 0742H
          IN AL, DX
        MOV DX, 0740H
        OUT DX, AL

        MOV AH, 00H
        MOV CX, 0008H
UP:     ROR AL, 01
        JNC SKIP
        INC AH
SKIP: LOOP UP

        MOV AL, AH
        MOV DX, 0744H
        OUT DX, AL
        JMP AGAIN
```

Problem-2

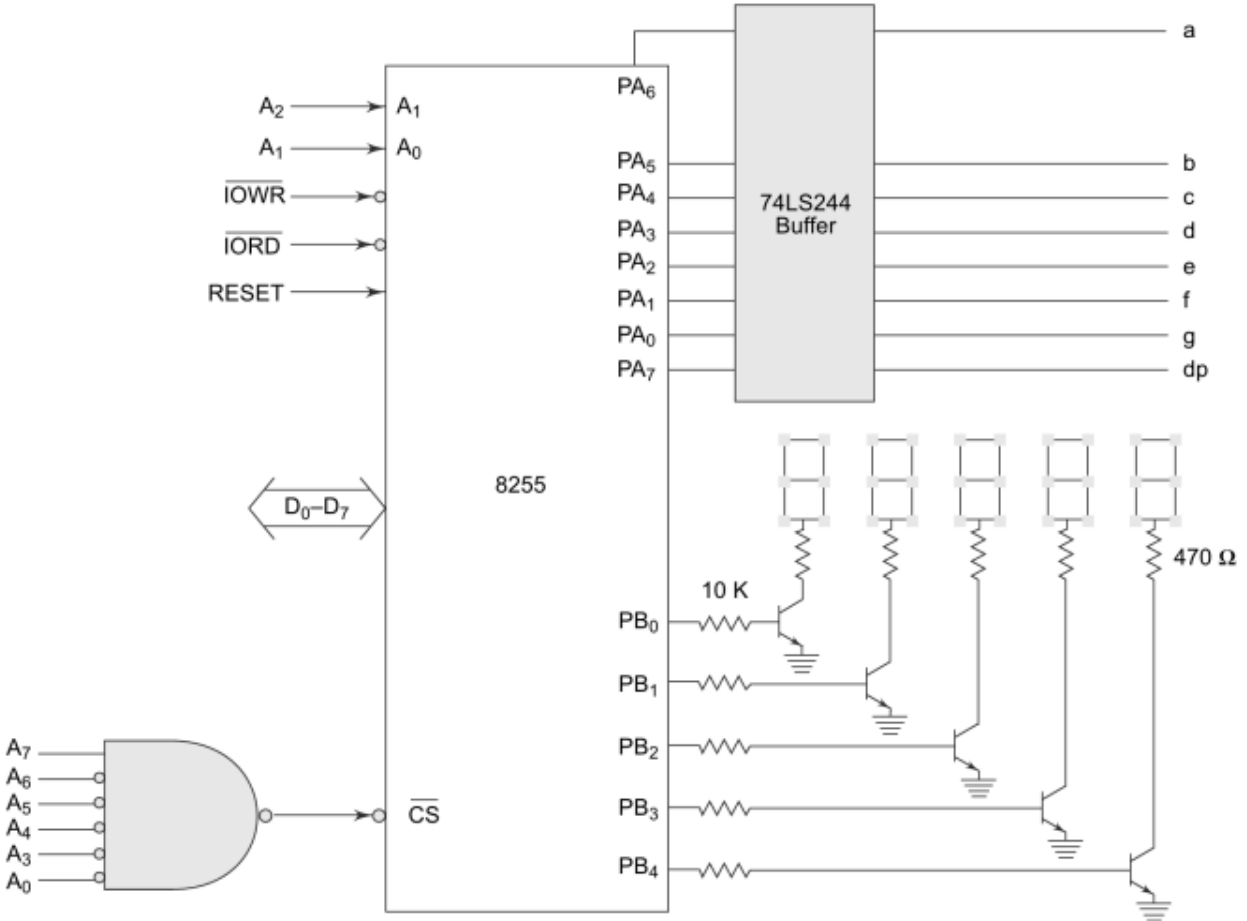
Interface an 8255 with 8086 at 80H as an I/O address of port A. Interface five 7 segment displays with the 8255. Write a sequence of instructions to display 1, 2, 3, 4 and 5 over the five displays continuously as per their positions starting with 1 at the least significant position.

Solution:

The hardware scheme for the above problem is shown in Fig. In this scheme, I/O port A is multiplexed to carry data for all the 7-segment displays. The port B selects one of the displays at a time. The displays used in the above hardware scheme are common cathode type. To glow a segment, logic 1 is applied on the corresponding line and the corresponding 7-segment display is selected by applying logic 1 on the port line that drives a transistor to ground the common cathode pin of the display. For a common cathode display, a '1', applied to a segment glows it and a '0' blanks it.

The control word is decided upon as follows:

B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0	Control word
1	0	0	0	0	0	0	0	= 80H
I/O mode	Port A in mode 0		Port A,o/p		Port B,mode 0	Port B, o/p		



Interfacing Multiplexed 7-Segment Display

PORT A ADDRESS: 80H
PORT B :82H, PORT C: 84H, CWR ADDRESS: 86H

Number to be displayed	PA_7 dp	PA_6 a	PA_5 b	PA_4 c	PA_3 d	PA_2 e	PA_1 f	PA_0 g	Code
1									
2									
3									
4									
5									

All these codes are stored in memory locations starting at DS: 2000H.

Program

```
MOV AL, 80H
OUT 86H, AL
```

```
AGAIN: MOV SI, 2000H
       MOV BL, 01H
       MOV CX, 0005H

UP:    MOV AL, [SI]
       OUT 80H, AL

       MOV AL, BL
       OUT 82H, AL

       INC SI
       ROL BL, 01
       LOOP UP

       JMP AGAIN
```

b) Mode-1: -(Strobed I/O Mode) Operation: -This mode is also called as single handshaking mode**.** In this mode handshaking control the input and output action of the specified port.

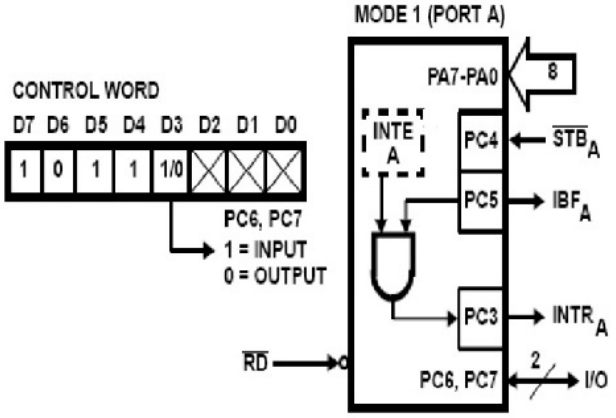
The basic functioning provided by this mode is given below: -

1. Two groups – group A and group B are available for strobe / handshake data transfer.
2. Each group contains one 8-bit I/O data port and one 4-bit control/data port.
3. The 8-bit data port can be either used as input and output port.
4. The 4-bit port is used for control and status of the 8-bit data port.
5. When Port A is to be programmed as an input port PC_3, PC_4 and PC_5 are used for control. PC_6 and PC_7 are not used and can be input or output as programmed by bit D_3 of the CWR.
6. When Port A is to be programmed as an output port PC_3, PC_6 and PC_7 are used for control. PC_4 and PC_5 are not used and can be input or output as programmed by bit D_3 of the CWR.
7. When Port B is to be programmed as an input or output port PC_0, PC_1 and PC_2 are used for control.

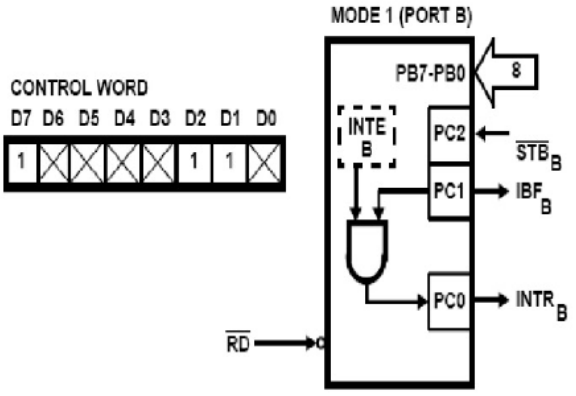
Input control signal definitions in mode 1:

1. $\overline{STB}$**(Strobe input)** – This active low input signal is activated by an input device to indicate that the data to be read is already sent on the port lines of 8255 port. If $\overline{STB}$ line falls to logic low level, the data available at 8-bit input port is loaded into input latches.
2. **IBF (Input buffer full)** – A high on this output indicates that data has been loaded into latches, i.e. it works as an acknowledgement to peripheral 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 the input device sends data on the port lines only when IBF signal is not active. IBF is set by $\overline{STB}$ input being low and is reset by the rising edge of $\overline{RD}$ input.

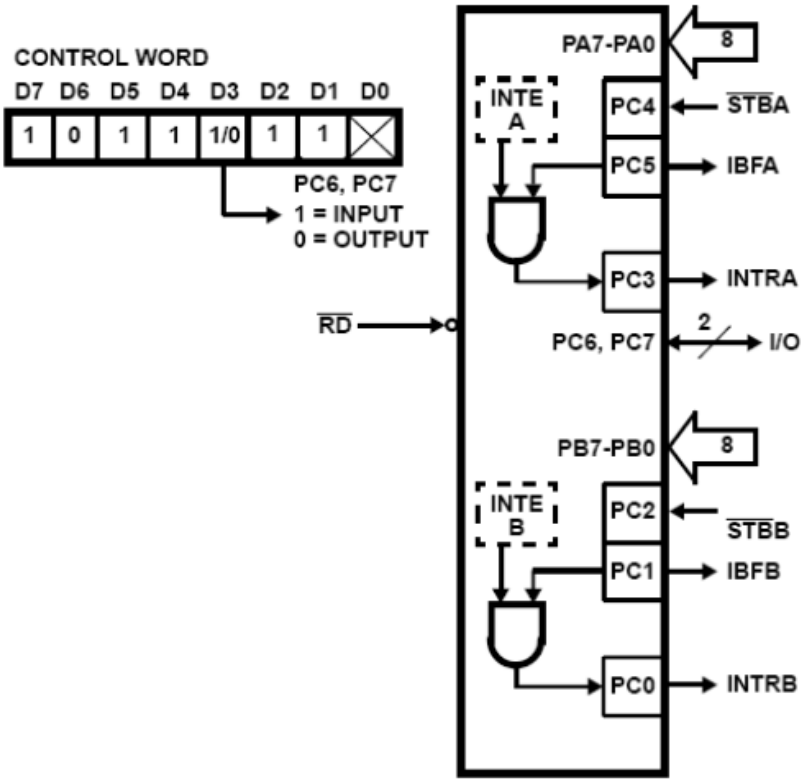
3. **INTR (Interrupt request)** – This active high output signal can be used to interrupt the CPU whenever an input device requests the service. The INTR signal is conditioned by $\overline{STB}$, IBF and INTE signals.
   INTR is set when $\overline{STB}$ =1, IBF = 1 and INTE =1 indicating CPU that the data from 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.
4. **INTE (Interrupt enable) –** This is an internal flip-flop used to enable or disable interrupt signal. These INTE flip-flops are set or reset by PC_4 bit for the port A and set or reset by PC_2 bit for the port B by using BSR (Bit set reset) mode only.



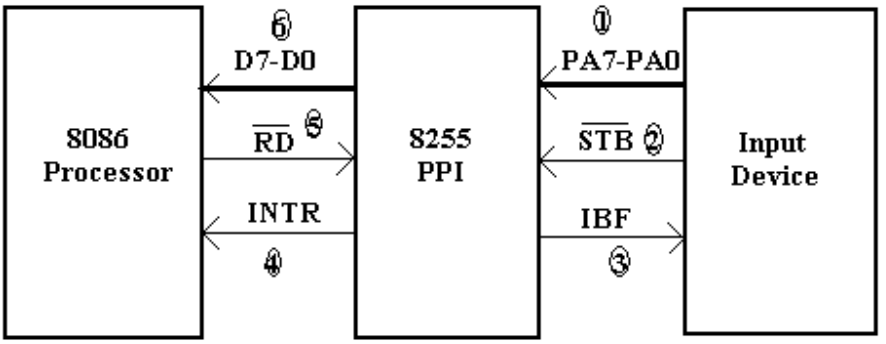
INTE A Controlled by bit set/reset of PC4.



INTE B Controlled by bit set/reset of PC2.




```
MOV     DX, PCW
MOV     AL, 10111110B
OUT     DX, AL
MOV     AL, 00001001B
OUT     DX, AL
MOV     AL, 00000101B
OUT     DX, AL
```



Data Accessed from an Input Device in Mode-1 Operation of 8255

The connected i/p device starts the process by sending out a byte of data to Port A on its eight data lines and then asserts its $\overline{STB}$ line low to tell the 8255 that a new byte of data has been sent. In response, the 8255 raises its input buffer full IBF signal on PC_5 high to tell the device that it is ready for the data. When the device detects the IBF signal at high level, it raises its $\overline{STB}$ signal high again. The raising edge of the $\overline{STB}$ signal has two effects on the 8255

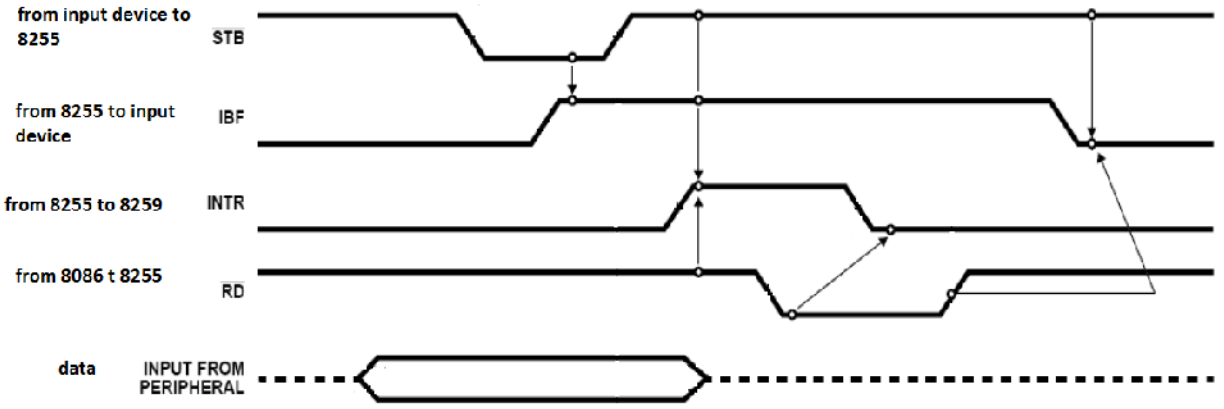
i) It first latches the data byte in the input latches of the 8255. Once the data is latched, the device can remove the data byte in preparation for sending the next data byte.

ii) Second, if the interrupt signal output has been enabled, the rising edge of $\overline{STB}$ signal will cause the 8255 to output an interrupt request signal to the microprocessor on PC_3.

The processors response to the interrupt request will be to go to an ISR which reads in the byte of data latched in Port A. when the $\overline{RD}$ signal from the Microprocessor goes LOW for this read of Port A, the 8255 automatically reset its Interrupt request signal on PC_3. This is done so that a second interrupt cannot be caused by the same data byte transfer.

When the processor raises its $\overline{RD}$ signal high again at the end of read operation, the 8255 automatically drop its IBF signal on PC_5 low again. IBF going low again is the signal to the device that the data transfer is complete and that it can send the next byte of data.

The device will not send the next byte of data until it detects that IBF signal has gone low again. The data transfer cycle will then repeat for the next data byte.



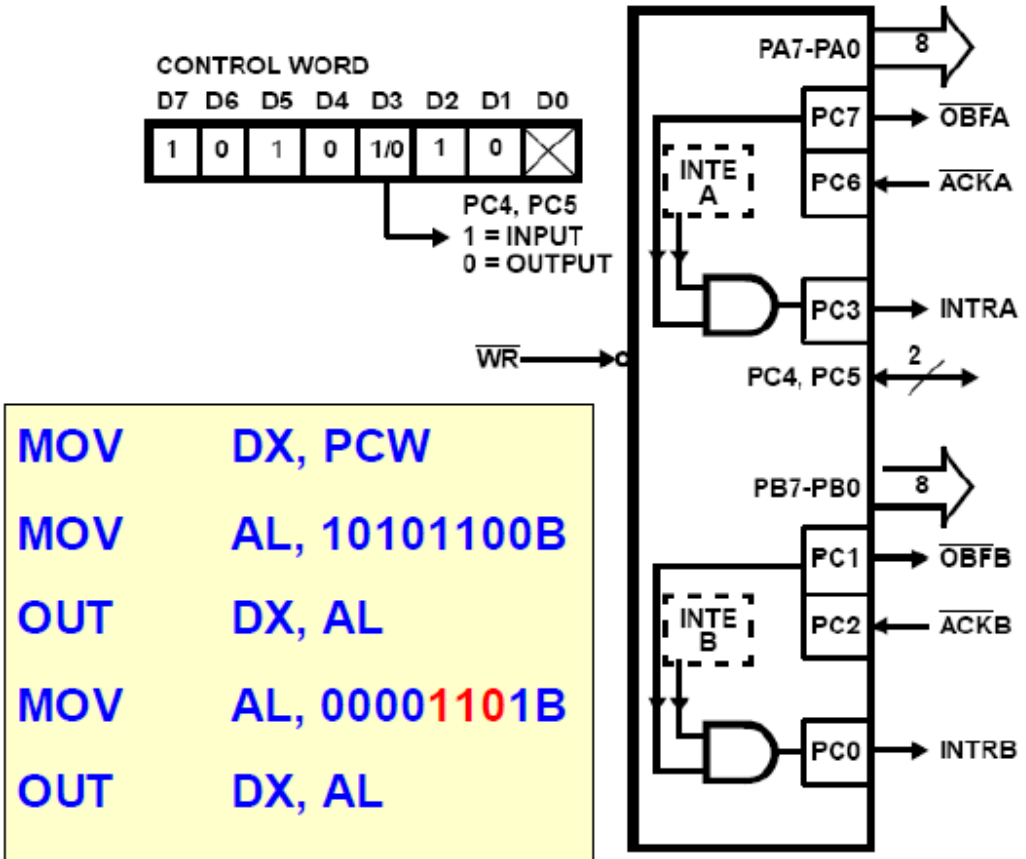
Timing waveforms for 8255 handshake data input from a input device

Output control signal definitions in mode 1:

1. $\overline{OBF}$**(Output buffer full)** – This status signal, whenever falls to low, indicates that CPU has written data to the specified output port. The OBF flip-flop will be set by a rising edge of $\overline{WR}$ signal and reset by a low going edge at the $\overline{ACK}$ input.
2. $\overline{ACK}$**(Acknowledge input)** – $\overline{ACK}$ signal acts as an acknowledgement to be given by an output device. $\overline{ACK}$ signal, whenever low indicates that the data transferred by the CPU to the output device through the port is received by the output device.
3. **INTR (Interrupt request**) – This is an active high output signal that can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. The INTR signal is conditioned by $\overline{ACK}$, $\overline{OBF}$ and INTE signals.

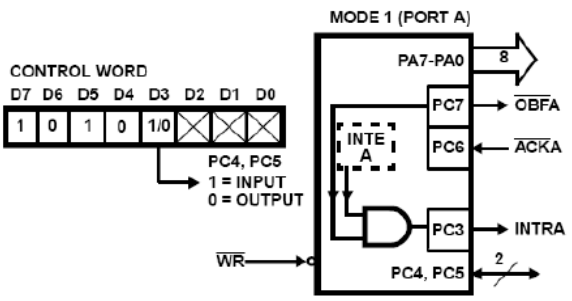
INTR is set when $\overline{OBF}$ =1, $\overline{ACK}$ = 1 and INTE =1 indicating that the output device is ready to accept next data byte. This signal is reset by the falling edge of the $\overline{WR}$ signal i.e. immediately after sending the data to the output port.

4. INTE (Interrupt enable) - This is an internal flip-flop used to enable or disable INTR signal. These INTE flip-flops are set or reset by PC_6 bit for the port A and set or reset by PC_2 bit for the port B by using BSR (bit set reset) mode only.

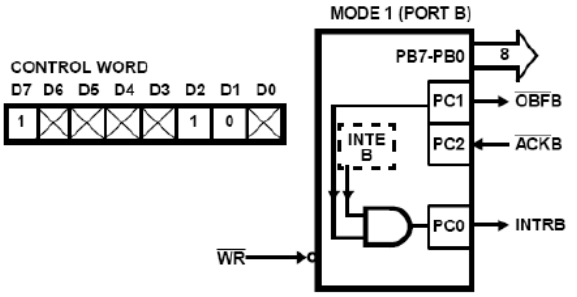



```
MOV     DX, PCW
MOV     AL, 10101100B
OUT     DX, AL
MOV     AL, 00001101B
OUT     DX, AL
MOV     AL, 00000101B
OUT     DX, AL
```

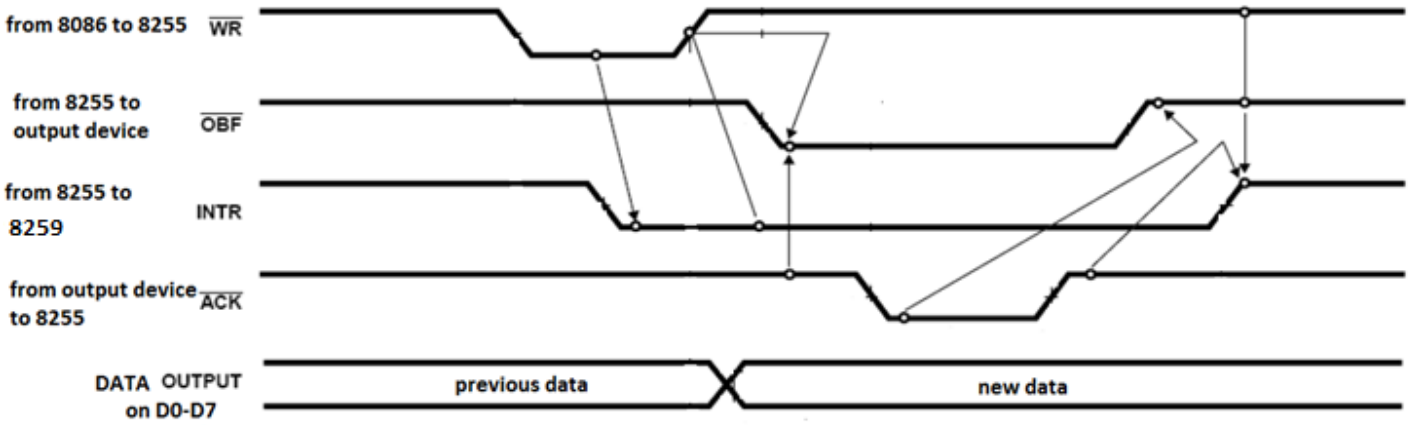
Output operation Mode-1



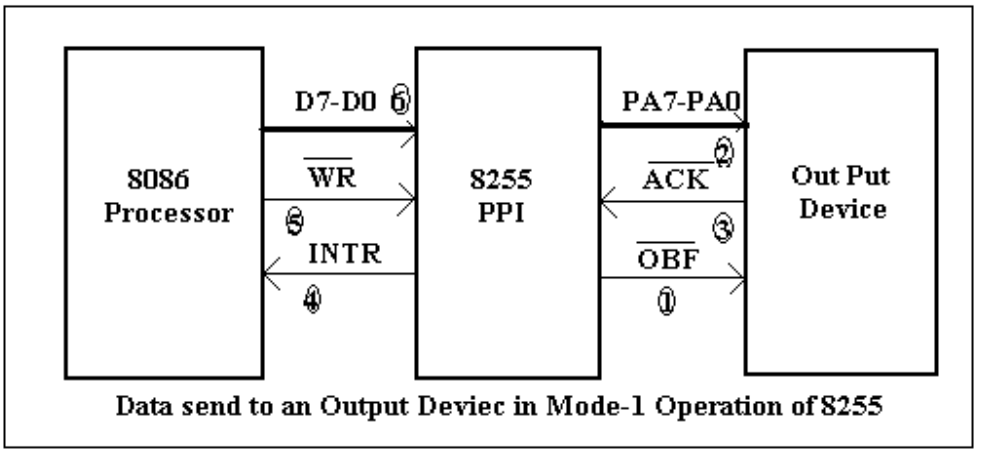
INTE A Controlled by Bit Set/Reset of PC6.



INTE B Controlled by Bit Set/Reset of PC2.

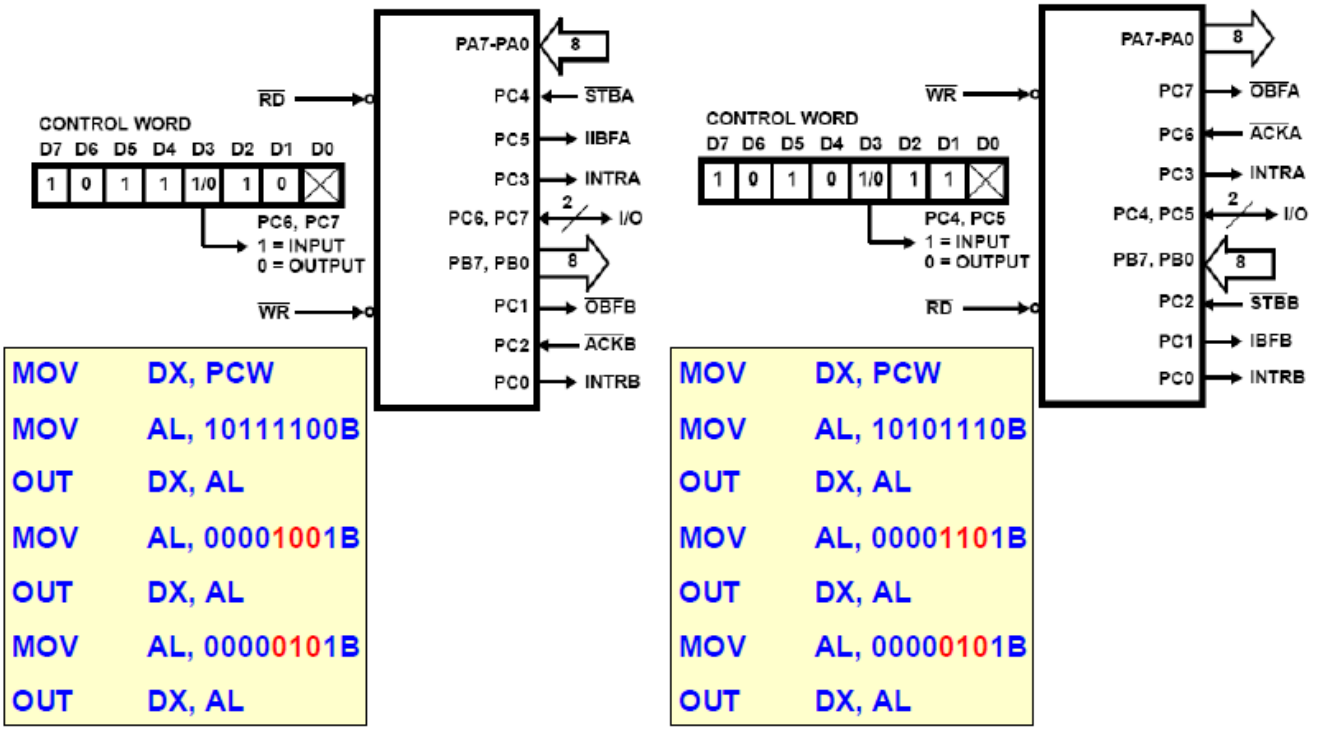


Timing waveforms for 8255 handshake data output to a output device



Data send to an Output Deviec in Mode-1 Operation of 8255

Combinations of Mode 1



c) Mode-2(Strobed Bidirectional I/O) Operation:

This mode of operation provides 8255 with additional features for communicating with a peripheral device on an 8-bit data bus for both transmitting and receiving data. Handshaking signals are provided to maintain proper data flow and synchronization between the data transmitter and receiver. The interrupt generation and other functions are similar to mode 1. The $\overline{RD}$ and $\overline{WR}$ signals decide whether the 8255 is going to operate as an input port or output port.

The basic functioning provided by this mode is given below: -

1. Used in group A only.
2. The 8-bit port (Port A) is bidirectional and additionally a 5-bit control port (port C) is available.
3. Three I/O lines are available at port C ($PC_2 – PC_0$)
4. Inputs and outputs are both latched.
5. The 5-bit control port C (PC_3-PC_7) is used for generating / accepting handshake signals for the 8-bit data transfer on port A.
6. When the 8255 is operated in Mode 2, 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, PC_0, PC_1 and PC_2 lines of Port C are used as handshaking signals.

Control signal definitions in mode 2(Strobed Bidirectional Bus I/O):

1. INTR– (Interrupt request)- As in mode 1, this control signal is active high and is used to interrupt the microprocessor to ask for transfer of the next data byte to/from it. This signal is used for input as well as output operations.

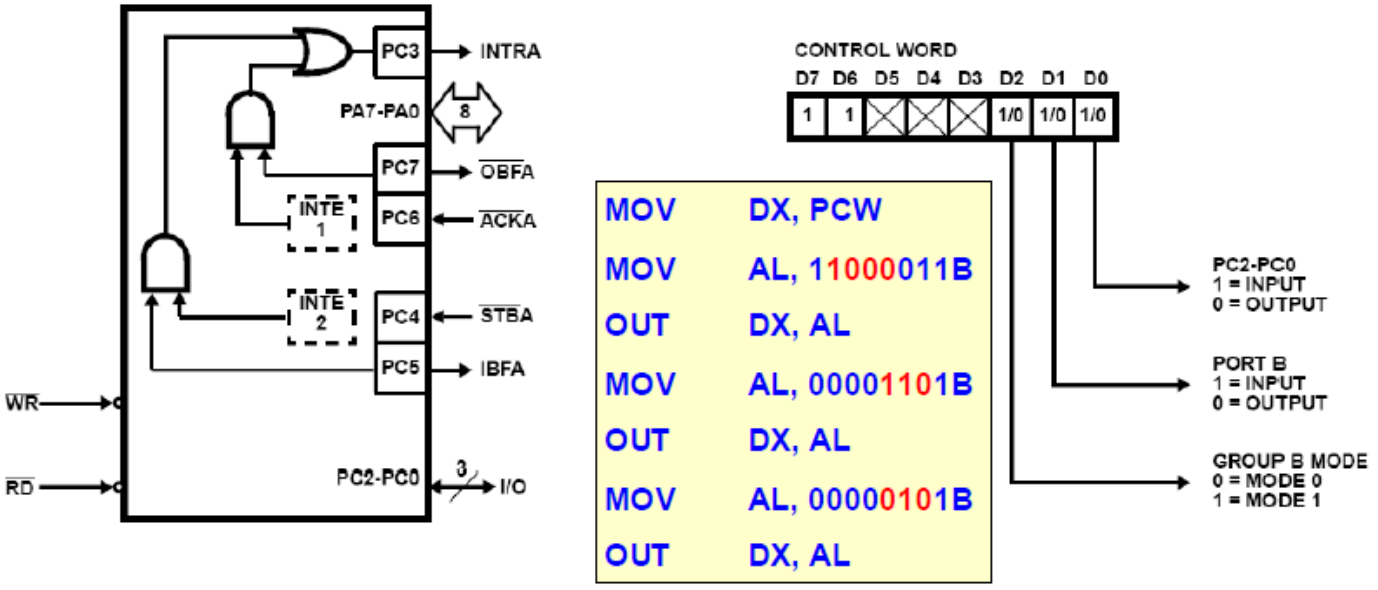
A. Control Signals for Output operations:

1. $\overline{OBF}$(Output buffer full) – This signal, when falls to low level, indicates that the CPU has written data to port A.
2. $\overline{ACK}$(Acknowledge)- This control input, when falls to logic low level, acknowledges that the previous data byte is received by the destination and next byte may be sent by the processor. This signal enables the internal tristate buffers to send the next data byte on port A.
3. $INTE_1$(A flag associated with OBF)- This can be controlled by bit set/reset mode using PC_6 bit of Port C.

B.* Control signals for input operations*:

1. $\overline{STB}$(Strobe input) - A low on this line is used to strobe in the data into the input latches of 8255.
2. IBF (Input buffer full) - When the data is loaded into input buffer, this signal rises to logic '1'. This can be used as an acknowledge that the data has been received by the receiver.
3. $INTE_2$ (A flag associated with IBF) - This can be controlled by bit set/reset mode using PC_4 bit of Port C.

Mode 2 (Strobed Bi-Directional Bus I/O)



Note:

- There are several combinations of modes possible. For any combination, some or all of the Port C lines are used for control or status. The remaining bits are either inputs or outputs as defined by a set mode command.
- During a read of Port C, the state of all the Port C lines, except the $\overline{ACK}$ and $\overline{STB}$ lines will be placed on data bus. In place of the $\overline{ACK}$ and $\overline{STB}$ lines Interrupt enable Flag Status will appear on the data bus in the PC_2, PC_4, PC_6 bit positions.
- Through a "WRITE Port C" command, only the Port C pins programmed as outputs in mode 0 group can be written. No other pins can be affected by a "Write Port C" command nor can the Interrupt enable flags be accessed.
- To write to any Port C output programmed as an output in a Mode-1 group or to change an Interrupt enable flag the "set/ reset Port C bit" Command must be used.
- Port C lines programmed as inputs including $\overline{ACK}$ and to the corresponding $\overline{STB}$ lines are not affected by a SET/RESET Port C Bit command. Writing Port C bit positions of the $\overline{ACK}$ and $\overline{STB}$ lines with the "set/ reset Port C bit" Command will affect the Group A and Group B Interrupt enable flags.

Reading Port C Status

- In modes 1 and 2 Port C generates or accepts Handshaking signals with the peripheral device.
- Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.
- There is no special instruction to read the status information from Port C, a normal read operation of Port C is executed to perform this function.

Status Word

MODE 1 STATUS WORD FORMAT

INPUT CONFIGURATION

D7	D6	D5	D4	D3	D2	D1	D0
I/O	I/O	IBFA	INTEA	INTRA	INTEB	IBFB	INTRB
GROUP A					GROUP B		

OUTPUT CONFIGURATION

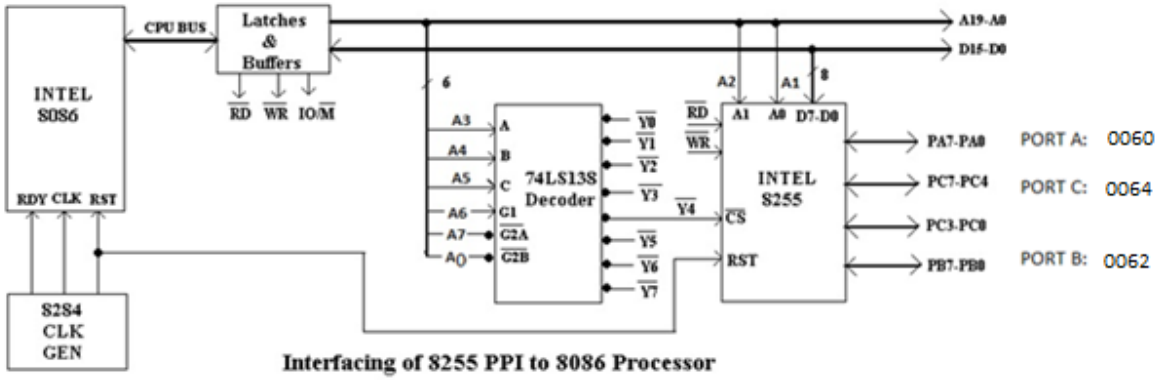
D7	D6	D5	D4	D3	D2	D1	D0
$\overline{OBFA}$	INTEA	I/O	I/O	INTRA	INTEB	$\overline{OBFB}$	INTRB
GROUP A					GROUP B		

MODE 2 STATUS WORD FORMAT

D7	D6	D5	D4	D3	D2	D1	D0
$\overline{OBFA}$	INTE1	IBFA	INTE2	INTRA	X	X	X
GROUP A					GROUP B		

(Defined by Mode 0 or Mode 1 Selection)

Interfacing of 8255 PPI to 8086 Microprocessor:-

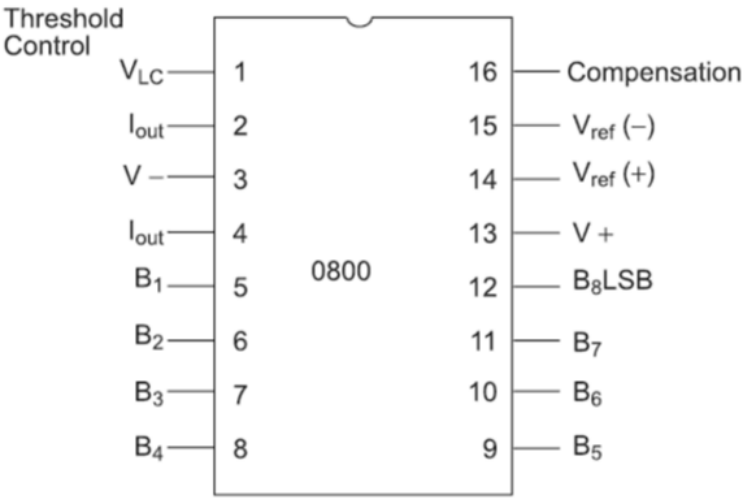


Interfacing of 8255 PPI to 8086 Processor

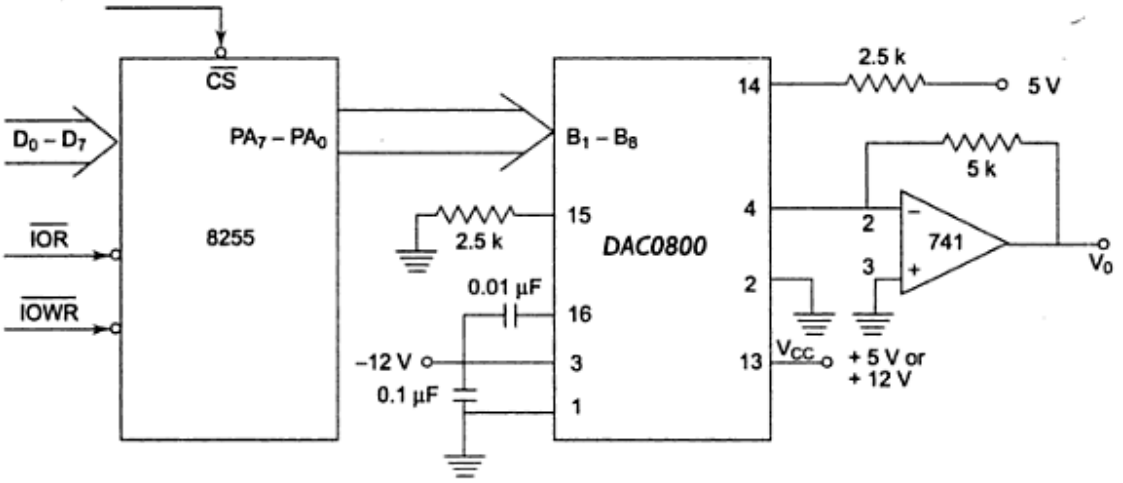
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
x	x	x	x	x	x	x	x	0	1	1	0	0	-	-	0
								Enable Signals		Y4			Ports		Enable

Mode Definition Summary

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA0	In	Out	In	Out	↔	
PA1	In	Out	In	Out	↔	
PA2	In	Out	In	Out	↔	
PA3	In	Out	In	Out	↔	
PA4	In	Out	In	Out	↔	
PA5	In	Out	In	Out	↔	
PA6	In	Out	In	Out	↔	
PA7	In	Out	In	Out	↔	
PB0	In	Out	In	Out		Mode 0 or Mode 1 Only
PB1	In	Out	In	Out		
PB2	In	Out	In	Out		
PB3	In	Out	In	Out		
PB4	In	Out	In	Out		
PB5	In	Out	In	Out		
PB6	In	Out	In	Out		
PB7	In	Out	In	Out		
PC0	In	Out	INTRB	INTRB	I/O	
PC1	In	Out	IBFB	$\overline{OBF}$B	I/O	
PC2	In	Out	$\overline{STB}$B	$\overline{ACK}$B	I/O	
PC3	In	Out	INTRA	INTRA	INTRA	
PC4	In	Out	$\overline{STB}$A	I/O	$\overline{STB}$A	
PC5	In	Out	IBFA	I/O	IBFA	
PC6	In	Out	I/O	$\overline{ACK}$A	$\overline{ACK}$A	
PC7	In	Out	I/O	$\overline{OBF}$A	$\overline{OBF}$A	



Pin Diagram of DAC 0800



Interfacing DAC0800 with 8086

Triangular wave

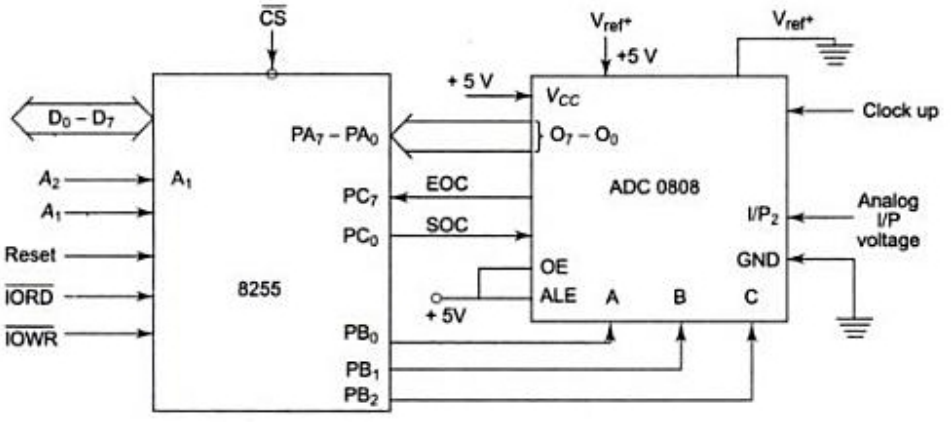
```
MOV AL, 80H
OUT CWR, AL

        MOV AL, 00H
UP1:    OUT PORT-A, AL
        INC AL
        CMP AL, FFH
        JNZ UP1

UP2:    OUT PORT-A, AL
        DEC AL
        CMP AL, 00H
        JNZ UP2
        JMP UP1
```

Square wave program

```
MOV AL, 80H
OUT CWR, AL

UP:     MOV AL, 0FFH
        OUT PORTA, AL
        CALL DELAY
        MOV AL, 00H
        OUT PORTA, AL
        CALL DELAY

        JMP UP
```

Sawtooth wave

```
MOV AL, 80H
OUT CWR, AL

        MOV AL, 00H
UP1:    OUT PORT-A, AL
        INC AL
        JMP UP1
```



Interfacing 0808 with 8086

Solution The control word is decided upon as follows:

B7	B_6	B_5	B_4	B_3	B_2	B_1	B_0	Control word
1	0	0	1	1	0	0	0	= 98H
I/O mode	Port A in mode 0		Port A,i/p	Port C,i/p	Port B,mode 0	Port B, o/p	Port C,o/p	

```
MOV AL, 98H
OUT CWR, AL

        MOV AL, 02H
        OUT PORTB, AL

        MOV AL, 00H
        OUT PORTC, AL
        MOV AL, 01H
        OUT PORTC, AL
        MOV AL, 00H
        OUT PORTC, AL
```

```
UP1:    IN AL, PORTC
        ROL AL, 01
        JNC UP1

        IN AL, PORTA
        MOV [SI], AL
        HLT
```

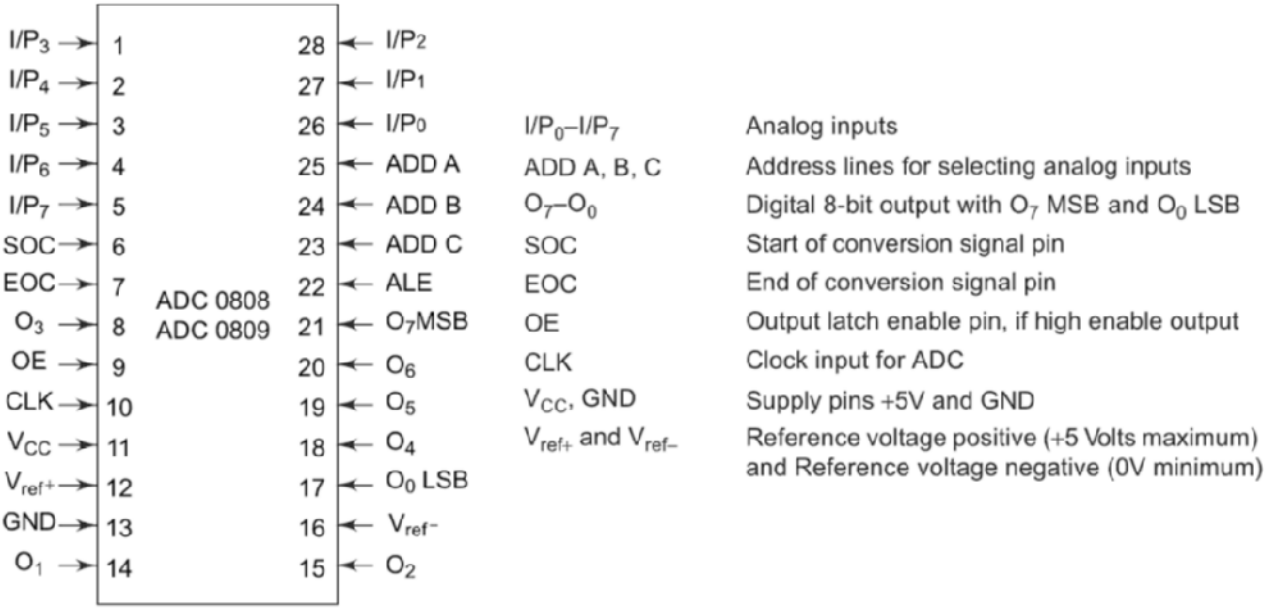


Fig. 5.37(b) *Pin Diagram of ADC 0808/0809*

8259
PROGRAMMABLE INTERRUPT CONTROLLER

Review

Non-Maskable Interrupt (NMI)

- Whenever the NMI input is activated, a type 2 interrupt occurs because NMI is internally decoded.

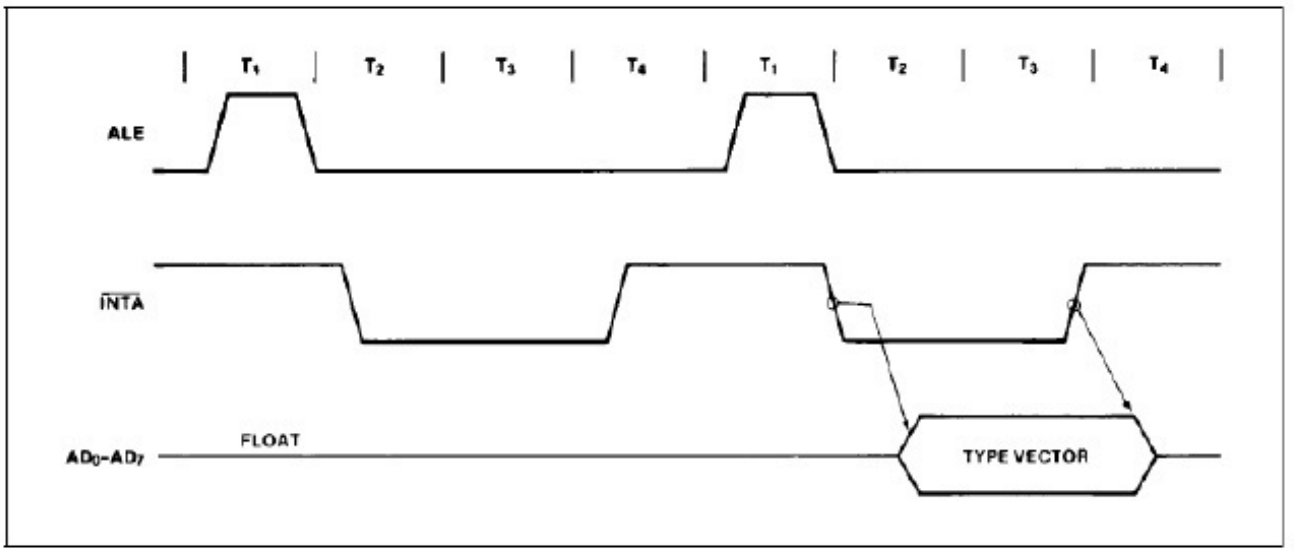
- The NMI is an edge-triggered input that requests an interrupt on the positive edge (0-to-1 transition).
- After a positive edge, the NMI pin must remain a logic 1 two T-states at minimum.
- The NMI input is often used for major system faults such as power failures. Power failures are easily detected by monitoring the AC power line and causing an NMI interrupt whenever AC power drops out.

Interrupt Input (INTR)

- The INTR input is automatically disabled once it is accepted by the microprocessor and re-enabled by the IRET instruction at the end of the Interrupt Service Routine (ISR).
- The INTR input must be externally decoded to select a vector. Any interrupt vector can be chosen for the INTR pin, but we usually use an interrupt type number between 20H and FFH.
- The microprocessor responds to the INTR input by pulsing the $\overline{\text{INTA}}$ output in anticipation of receiving an interrupt vector type on data bus connections (AD7-AD0).
- There are two $\overline{\text{INTA}}$ pulses generated by the microprocessor that are used to insert the vector type number on the data bus.
- The INTR can be disabled by a CLI instruction which clears the IF, and enabled by a STI instruction which sets the IF.

Name	Initiated by	Maskable	Trigger	Priority	Acknowledge Signal	Vector Table Address
NMI	External Hardware	No	↑edge, hold 2T states min.	2	None	00008H-0000BH
INTR	External Hardware	Yes via IF	High level until acknowledged	3	$\overline{\text{INTA}}$	n × 4
INT n	Internal via Software	No	None	1	None	n × 4
INT 3 (Breakpoint)	Internal via Software	No	None	1	None	0000CH-0000FH
INTO	Internal via Software	No	None	1	None	00010H-00013H
Divide-by-0	Internal via CPU	No	None	1	None	00000H-00003H
Single-Step	Internal via CPU	Yes via TF	None	4	None	00004H-00007H

Interrupt Acknowledge Timing



Methods to Service an I/O Device

- Programmed I/O (Polling)
- Interrupt Driven I/O

Programmed I/O

- Programmed I/O method is the first method used to check whether an I/O device needs a service.
- Programmed I/O checks the status of I/O interface periodically under the control of the software.
- It is appropriate for small systems and dedicated applications with limited number of I/O devices.
- The drawbacks of programmed I/O:
  - It wastes the CPU time in busy wait loops.
  - It is difficult to implement prioritized service.
  - It worsens the response time of a CPU to respond to a request.
  - Example:

```
START:      MOV     DX, PORTC
AGAIN:      IN      AL, DX
            TEST    AL, 00001000B
            JZ      AGAIN
;
            MOV     DX, PORTA
            IN      AL, DX
            INC     DX
            OUT     DX, AL

            JMP     START
```

Interrupt Driven I/O

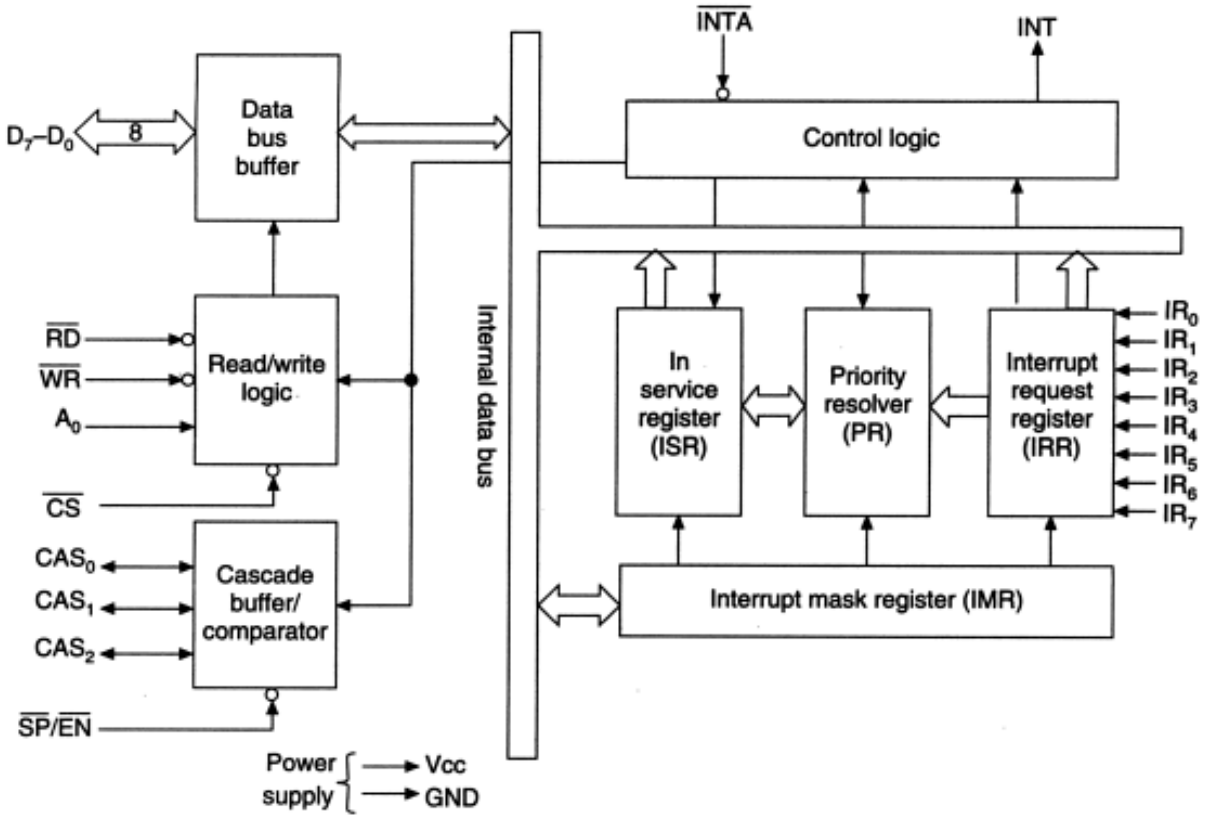
- Unlike the polling technique, interrupt processing allows the microprocessor to execute other software.
- As soon as the microprocessor gets an interrupt from I/O interface, it stops executing the running program and starts the execution of the Interrupt Service Routine (ISR) to service the I/O device.
- When the microprocessor finishes the execution of ISR, it returns to the interrupted program.

- The advantages of the interrupt driven I/O method:
  - It increases the CPU throughput since the microprocessor does not waste its time in the busy wait loop.
  - It is easy to implement prioritized service.
  - It improves the response time.

8259A Main Features

- It accepts eight interrupt requests (expandable to 64 by cascading 8 slaves PICs).
- It prioritizes the interrupt requests (selects the highest priority request for service).
- It issues a single interrupt request to the CPU.
- In response to the $\overline{\text{INTA}}$, it issues a unique type number (vector) for each interrupt request input. In addition, type numbers are programmable.
- It has variety of programmable modes of operations.
- The interrupt requests are individually maskable.
- The operating modes and masks may be dynamically changed by the software at any time during execution of program.
- The 8259 can accept requests from the peripheral, determines priority of incoming request, checks whether the incoming request has a higher priority value than the level currently being serviced, and issues an interrupt signal to the microprocessor.
- It can be used in polled as well as in vectored modes.
- It can be monitored in buffered mode which can be used for the large system.
- The starting address called vector number is programmable.
- The 8259 is static, does not require clock signal.
- It can identify the interrupting device.
- It can resolve the priority of interrupt i.e. it does not require any external priority resolver.
- It can be operated in various priority modes such as fixed priority and rotating priority modes.

Block Diagram of 8259 PIC: -



- The architecture or functional block diagram of the 8259 programmable interrupt controller is shown in Figure. It includes 8 blocks: control logic, read/write logic, data bus buffer, in-service register, priority resolver, interrupt request register, interrupt mask register, and cascade buffer/comparator.

Data Bus Buffer

- The 3-state, bidirectional 8-bit data bus is used to interface the 8259 PIC data bus with the system data bus. Control words and status information are transferred through the data bus buffer.
- It is internally connected to the data bus and its outer pins D7-D0 are connected to the system data bus directly.
- The directions of data buffer are decided by read and write control signals.
- When signal read is activated, then it transmits data to the system. When write signal is activated, then it receives data from the system data bus. This reading and writing operation is achieved by IN and OUT microprocessor instructions.

Read/Write Logic

- This block accepts inputs from system control bus and address bus. The control signals are $\overline{RD}$ and $\overline{WR}$, and address signals used are A0 and $\overline{CS}$.
- $\overline{RD}$ and $\overline{WR}$ are connected to $\overline{IOR}$, $\overline{IOW}$, or $\overline{MEMR}$, $\overline{MEMW}$ depending on the types of mapping used.
- $\overline{RD}$ and $\overline{WR}$ decide the operation is to be performed, i.e. write data to the 8259 PIC or read data fromthe 8259 PIC. A_0 is directly connected to address lines A_1 of system address lines.
- The $\overline{CS}$ is connected to the chip select decoder; the selection of the 8259 is enabled or disabled by $\overline{CS}$ signal. If logic 0 is applied on this signal, the 8259 PIC is selected, or when 1 is applied on this signal, the 8259 PIC is rejected.

Control Logic

- This block has two pins: INT (interrupt) as an output, and $\overline{INTA}$ (interrupt acknowledge) as an input.
- The INT is connected to the interrupt pin of the microprocessor.
- Whenever a valid interrupt is asserted, this signal goes high. The $\overline{INTA}$ is the interrupt acknowledgement signal from the microprocessor.

Interrupt Request Register (IRR)

- The interrupt at the IR (Interrupt Request) lines are handled by interrupt request register internally.
- The Interrupt Request Register (IRR) is used to store all the interrupt levels which are requesting service in it in order to serve them one-by-one on the priority basis.

In-Service Register (ISR)

- In-service register is used to store all the interrupt levels which are being serviced.
- Each bit of this register is set by the priority resolver and reset by the end of the interrupt command word.
- The microprocessor can read the contents of this register by issuing appropriate command word.

Priority Resolver (PR)

- Priority resolver determines the priorities of the bits set in the IRR. To make decision, the priority resolver looks at the ISR. If the highest priority bit in the ISR is set, then it ignores the new request.
- If the priority resolver finds that the new interrupt has a higher priority than the interrupt currently being serviced, then it will set the appropriate bit in the ISR and send the INT signal to the microprocessor for the new interrupt request.

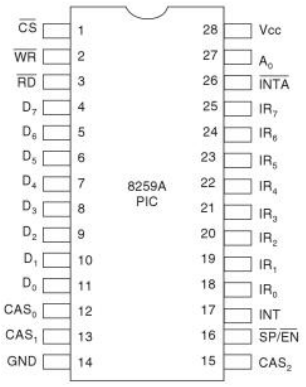
Interrupt Mask Register (IMR)

- It is a programmable register. It is used to mask unwanted interrupt request by writing appropriate control word.
- The IMR operates on the IRR. Masking of a higher priority input will not affect the interrupt request lines of lower priority.
- The microprocessor can read contents of this register without issuing any command word.

Cascade Buffer/Comparator

- This functional block stores and compares the identification numbers (IDs) of all 8259s used in the system.
- The associated three I/O pins CAS_2-CAS_0 are outputs when the 8259 is used as a master and the inputs when the 8259 is used as a slave.
- As a master, the 8259 sends the ID of the interrupting slave device onto the CAS_2-CAS_0 lines. The slave thus selected will send its programmed subroutine address onto the data bus during the next one or two consecutive $\overline{INTA}$ pulses.
- In buffered mode, it generates $\overline{EN}$ signal.
- When just one 8259 is used in a system, the cascade lines (CAS_0, CAS_1, and CAS_2) can be left open.

PIN CONFIGURATION OF 8259 PIC



Pin configuration of 8259 PIC.

Pin Description

Symbol	Pin No.	Type	Name and Function
V_{CC}	28	I	**SUPPLY:** + 5V Supply.
GND	14	I	**GROUND**
$\overline{CS}$	1	I	**CHIP SELECT:** A low on this pin enables $\overline{RD}$ and $\overline{WR}$ communication between the CPU and the 8259A. INTA functions are independent of CS.
$\overline{WR}$	2	I	**WRITE:** A low on this pin when CS is low enables the 8259A to accept command words from the CPU.
$\overline{RD}$	3	I	**READ:** A low on this pin when CS is low enables the 8259A to release status onto the data bus for the CPU.
D_7–D_0	4–11	I/O	**BIDIRECTIONAL DATA BUS:** Control, status and interrupt-vector information is transferred via this bus.
CAS_0–CAS_2	12, 13, 15	I/O	**CASCADE LINES:** The CAS lines form a private 8259A bus to control a multiple 8259A structure. These pins are outputs for a master 8259A and inputs for a slave 8259A.
$\overline{SP}/\overline{EN}$	16	I/O	**SLAVE PROGRAM/ENABLE BUFFER:** This is a dual function pin. When in the Buffered Mode it can be used as an output to control buffer transceivers (EN). When not in the buffered mode it is used as an input to designate a master (SP = 1) or slave (SP = 0).
INT	17	O	**INTERRUPT:** This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU, thus it is connected to the CPU's interrupt pin.
IR_0–IR_7	18–25	I	**INTERRUPT REQUESTS:** Asynchronous inputs. An interrupt request is executed by raising an IR input (low to high), and holding it high until it is acknowledged (Edge Triggered Mode), or just by a high level on an IR input (Level Triggered Mode).
$\overline{INTA}$	26	I	**INTERRUPT ACKNOWLEDGE:** This pin is used to enable 8259A interrupt-vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.
A_0	27	I	**AO ADDRESS LINE:** This pin acts in conjunction with the $\overline{CS}$, $\overline{WR}$, and $\overline{RD}$ pins. It is used by the 8259A to decipher various Command Words the CPU writes and status the CPU wishes to read. It is typically connected to the CPU A0 address line (A1 for 8086, 8088).

The interrupt sequence in an 8086-8259A system is described as follows:

- One or more IR lines are raised high that set corresponding IRR bits.
- 8259 resolves priority and sends an INT signal to CPU.
- The CPU acknowledges the INT and responds with an $\overline{\text{INTA}}$ pulse.
- Upon receiving an $\overline{\text{INTA}}$ signal from the CPU, the highest priority ISR bit is set and the corresponding IRR bit is reset. The 8259A does not drive data bus during this period.
- The 8086 will initiate a second $\overline{\text{INTA}}$ pulse. During this period 8259A releases an 8-bit pointer on to data bus from where it is read by the CPU.
- This completes the interrupt cycle. The ISR bit is reset at the end of the second $\overline{\text{INTA}}$ pulse if automatic end of interrupt (AEOI) mode is programmed. Otherwise ISR bit remains set until an appropriate EOI command is issued at the end of interrupt subroutine.

CONTROL/COMMAND WORDS OF 8259

- The 8259A accepts two types of command words generated by the CPU:

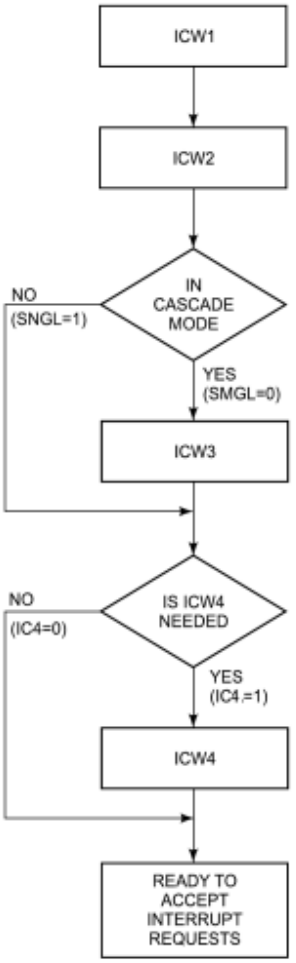
1. **Initialization Command Words (ICWs):** Before normal operation can begin, each 8259A in the system must be brought to a starting point by a sequence of 2 to 4 bytes timed by $\overline{\text{WR}}$ pulses.

2. **Operation Command Words (OCWs):** These are the command words which command the 8259A to operate in various interrupt modes. These modes are:
    - Fully nested mode
    - Rotating priority mode
    - Special mask mode
    - Polled mode

The OCWs can be written into the 8259A any time after initialization.

Initialization Command Words:

Initialization command words (ICWs) needed for various 8259A applications are given in this flowchart form. According to this flowchart, an ICW I and an ICW2 must be sent to any 8259A in the system. If the system has any slave 8259As (cascade mode), then an ICW3 must be sent to the master, and a different ICW3 must be sent to the slave. If the system is an 8086 or if we want to specify certain special conditions, then we have to send an ICW4 to the master and to each slave.



Flowchart of Initialization sequence of 8259 PIC

ICW 1

ICW 1 is used to program the basic operations of the 8259 PIC. To write the command word into the 1CW 1 register, A_0 pin should be at logic 0 i.e. ICW1 write in Port 0. (In the 8086 A_0 is replaced by A_1 because A_0 is used for the even bank)

PORT 0 (A_0 = 0)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
A_7	A_6	A_5	1	LTM	ADI	SNGL	IC4	→ 8085
X	X	X	1	LTM	X	SNGL	IC4	→ 8086

ICW 1 is used to perform the following operations.

(a) IC4 (bit D_0) ICW4: This bit indicates whether ICW4 is required or not. If IC4 = 0, then ICW4 is not required and if this bit is set, then ICW4 is required.

(b) SNGL (bit D_1) Single: This bit is used to indicate whether the 8259 PIC is cascaded or not. If SNGL = 1 (set), then single 8259 PIC is used and when SNGL = 0 (cleared), then cascaded 8259 PICs are used.

(c) ADI (bit D_2) Address interval: This bit is not used in the 8086. It is required in 8085 to define the address interval spacing between successive interrupt routines.

If ADI = 1 (set), then address interval is of 4 bytes.

(d) LTM (bit D_3): Level-triggered mode or level-triggered Interrupt mode: This bit is used to determine if the interrupt request is to be recognized in the level-triggered mode or in the edge-triggered mode.

If LTM = 1, then all IR input levels are level-triggered mode.

If LTM = 0, then all IR input levels are edge-triggered mode.

(e) A_7-A_5 (bits D_7-D_5) Vectored address: These bits are not used in the 8086 microprocessor. These bits are used for the interrupt service routine (ISR) address for the microprocessor 8085.

ICW2

To write the command word into ICW2 register, A_0 pin should be at logic 1 i.e. defined in Port 1only.

PORT 1 (A_0 = 1)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	→ 8085
T_7	T_6	T_5	T_4	T_3	X	X	X	→ 8086

- In the 8085 microprocessor, these bits are used to provide A_{15}-A_8 address bits of ISR address.
- In the 8086 microprocessor, this mode is used to program T_7 to T_3 bits of an 8-bit vector number or vector address. The lower three bits T_2-T_0 are fixed and these are provided by the 8259 PIC itself as shown in the following table:

Interrupt	**T_7**	**T_6**	**T_5**	**T_4**	**T_3**	**T_2**	**T_1**	**T_0**
IR_0	T_7	T_6	T_5	T_4	T_3	0	0	0
IR_1	T_7	T_6	T_5	T_4	T_3	0	0	1
IR_2	T_7	T_6	T_5	T_4	T_3	0	1	0
IR_3	T_7	T_6	T_5	T_4	T_3	0	1	1
IR_4	T_7	T_6	T_5	T_4	T_3	1	0	0

IR5	T7	T6	T5	T4	T3	1	0	1
IR6	T7	T6	T5	T4	T3	1	1	0
IR7	T7	T6	T5	T4	T3	1	1	1

ICW3

- ICW3 is optional and it is transferred if these are more than one 8259 PICs in the system (cascade mode), in which case SNGL = 0.
- There are two types of ICW3 used
    - master ICW3 and
    - slave ICW3.
- To write the command word into ICW3 register for the both, i.e. master and slave, A_0 pin should be at logic 1, i.e. defined in Port 1 only.

ICW3 for master:

PORT 1 (A_0 = 1)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SL_7	SL_6	SL_5	SL_4	SL_3	SL_2	SL_1	SL_0

ICW3 for slave

PORT 1 (A_0 = 1)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	ID_2	ID_1	ID_0

- In the master mode (either when $\overline{SP}$ = 1, or in buffered mode when M/$\overline{S}$ = 1 in ICW4) a "1" is set for each slave in the system. The master will enable the corresponding slave through the cascade lines.
- In the slave mode (either when $\overline{SP}$ = 0, or if BUF = 1 and M/$\overline{S}$ = 0 in ICW4) bits 2-0 identify the slave. The slave compares its cascade input with these bits and, if they are equal, an 8-bit pointer is released by it on the Data Bus.

These bits are coded as shown in table

Table Slave identification with ICW3

Master IR number	ID_2	ID_1	ID_0
IR7	1	1	1
IR6	1	1	0
IR5	1	0	1
IR4	1	0	0
IR3	0	1	1
IR2	0	1	0
IR1	0	0	1
IR0	0	0	0

ICW4

To write this command word A_0 pin should be at logic 1, i.e. it is defined in Port 1 only.

PORT 1 (A_0 = 1)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	SFNM	BUF	M/$\overline{S}$	AEOI	μPM

This command word is used to define.

a) **SFNM (bit 4) Special fully nested mode**:

If SFNM = 1, then special fully nested mode is programmed.

If SFNM = 0, then fully nested mode is programmed.

This bit is used in cascaded mode only.

b) **BUF (bit D3) Buffered**: If BUF = 1, then buffered mode is programmed. In buffered mode, $\overline{SP}/\overline{EN}$ becomes an enable output and the master/slave is determined by M/$\overline{S}$. If BUF = 0, then non-buffered mode is programmed

c) **M/$\overline{S}$ (bit D2) Master/Slave**: If Buffered mode is selected and M/$\overline{S}$ =1, it means that 8259 is programmed to be a master, and if M/$\overline{S}$ =0, it means that 8259 is programmed to be a slave. If BUF=0 then this bit has no function.

BUF	*M/$\overline{S}$*	*Operation*
0	X	Non-buffered mode
1	1	Master is buffered
1	0	Slave is buffered

d) AEOI (bit D1) Automatic End of Interrupt: it indicates whether the 8259 is operated in normal EOI mode or in auto EOI mode.

If AEOI=0, then normal EOI command mode is programmed,

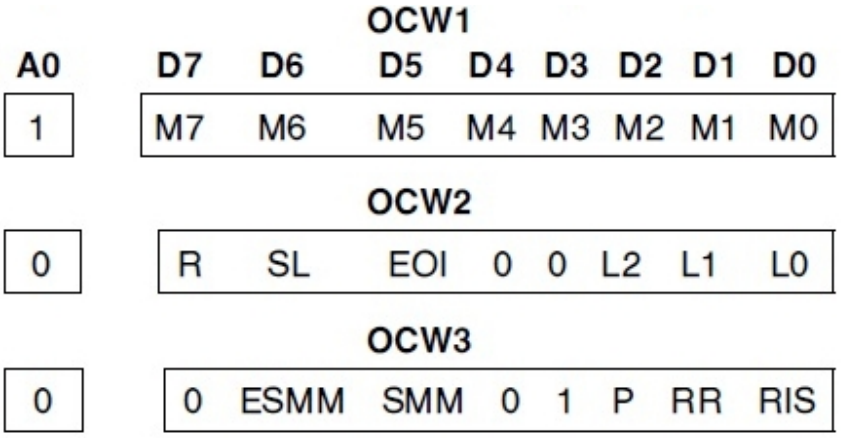
If AEOI=1, then automatic EOI command mode is programmed.

e) **µPM (bit D0) Microprocessor Mode:** It indicates whether the 8259 is operated with the 8085 or 8086 Microprocessor

If µPM =0, it sets the 8259 PIC for the Microprocessor 8085 system operation, and
If µPM =1, it sets the 8259 PIC for the Microprocessor 8086 system operation.

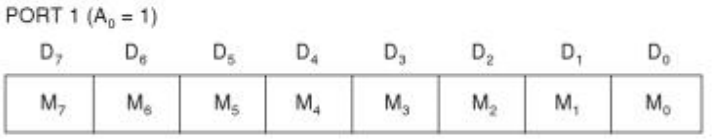
Operation Command Words:

After the Initialization Command Words (ICWs) are programmed into the 8259A, the chip is ready to accept interrupt requests at its input lines. However, during the 8259A operation, a selection of algorithms can command the 8259A to operate in various modes through the Operation Command Words (OCWs).



Operation Command Word 1 (OCW 1)

This command word is used to mask out unwanted interrupt request inputs (IR inputs).
To write this command word into interrupt mask register (IMR).
A_0 pin should be at logic 1, i.e. in Port 1.



M_7—M_0 represent the eight mask bits for the IR_7-IR_0 interrupt request levels.
M = 1 indicates Interrupt Request (IR) input is masked or disabled.
M = 0 indicates Interrupt Request (IR) input is unmasked or enabled.

Programming the 8259A

; ICW1 Initialization (IC4 Needed, Single, Edge Triggered Mode)

```
MOV DX, PIC_PORT0
MOV AL, 00010011B
OUT DX, AL
```

```
; ICW2 Initialization (Interrupt Vector Address)
INC DX
MOV AL, VECTOR
OUT DX, AL
; ICW4 Initialization (8086 Mode, Normal EOI, Non-Buffered Mode, ; Not Special Fully Nested Mode)
MOV AL, 00000001B
OUT DX, AL
; OCW1 Initialization (Unmask IR1 Only)
MOV AL, 11111101B
OUT DX, AL
```

Operation Command Word 2 (OCW 2)

This command word is used to provide end of interrupt (EOI) command which clears appropriate bits of ISR, rotate priorities in normal EOI, and auto EOI modes.

PORT 0 ($A_0 = 0$)

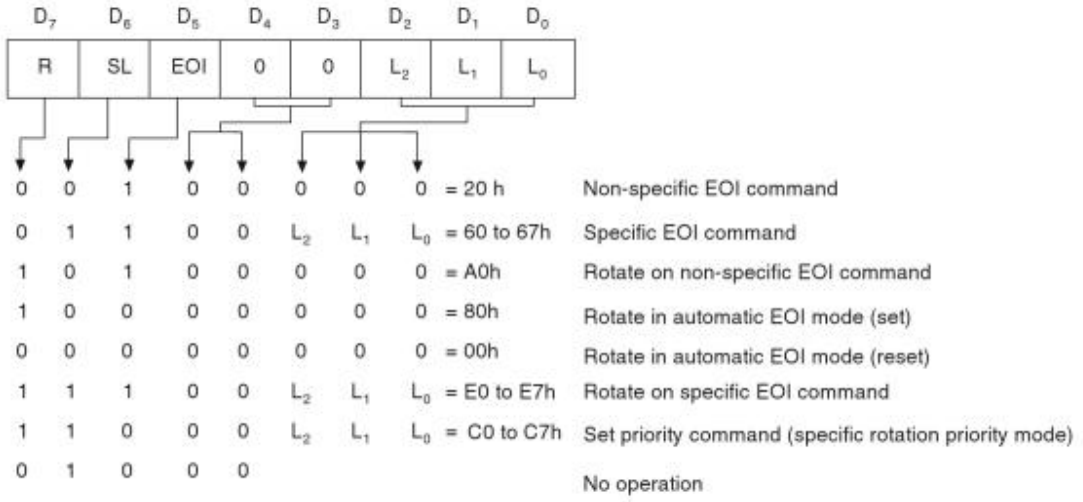
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
R	SL	EOI	0	0	L_2	L_1	L_0

To write this command word into OCW2 register, A_0 should be at logic 0, i.e. in Port 0.

R, SL and EOI three bits are used to control the rotate and end of interrupt modes, and combinations of the two.

R – Rotate; SL- Specific level; EOI: End of Interrupt

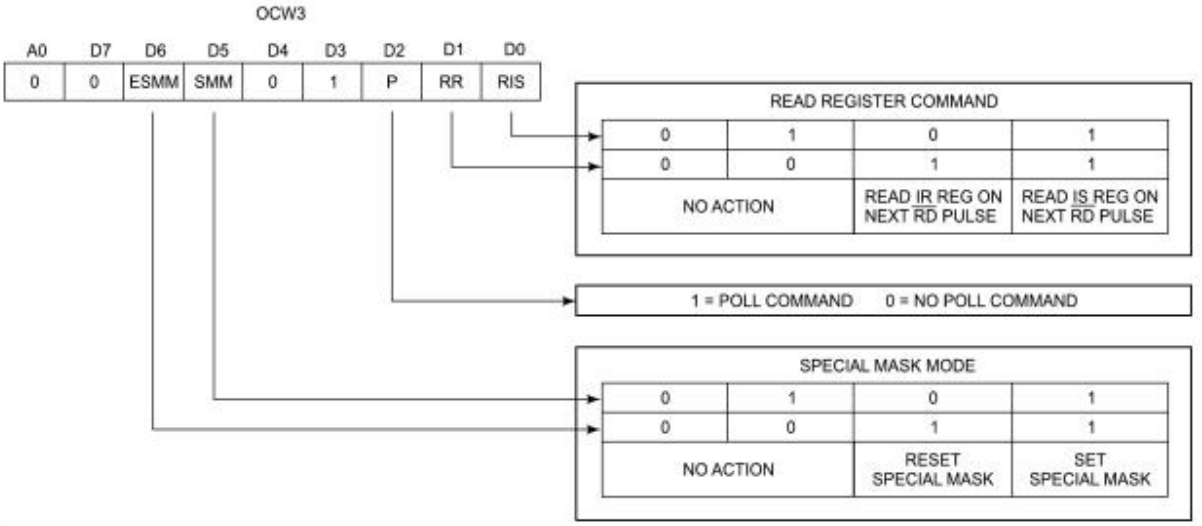
L2, L1, L0 — these bits determine the interrupt level acted upon when the SL bit is active.



Operation Command Word 3 (OCW3)

ESMM — Enable Special Mask Mode. When this bit is set to 1 it enables the SMM bit to set or reset the Special Mask Mode. When ESMM = 0 the SMM bit becomes a "don't care".

SMM — Special Mask Mode. If ESMM = 1 and SMM = 1 the 8259A will enter Special Mask Mode. If ESMM = 1 and SMM = 0 the 8259A will revert to normal mask mode. When ESMM = 0, SMM has no effect.



Reading the 8259A Status

The input status of several internal registers can be read to update the user information on the system.

The following registers can be read via OCW3 (IRR and ISR or OCW1 [IMR]).

- Interrupt Request Register (IRR): 8-bit register which contains the levels requesting an interrupt to be acknowledged. The highest request level is reset from the IRR when an interrupt is acknowledged. (Not affected by IMR.)
- In-Service Register (ISR): 8-bit register which contains the priority levels that are being serviced. The ISR is updated when an End of Interrupt Command is issued.
- Interrupt Mask Register: 8-bit register which contains the interrupt request lines which are masked.
- The IRR can be read when, prior to the $\overline{RD}$ pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 0.)
- The ISR can be read, when, prior to the $\overline{RD}$ pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 1).
- There is no need to write an OCW3 before every status read operation, as long as the status read corresponds with the previous one; i.e., the 8259A "remembers" whether the IRR or ISR has been previously selected by the OCW3. This is not true when poll is used.
- After initialization the 8259A is set to IRR.
- For reading the IMR, no OCW3 is needed. The output data bus will contain the IMR whenever $\overline{RD}$ is active and A0 = 1 (OCW1).
- Polling overrides status read when P = 1, RR = 1 in OCW3.

Program for Reading IMR, ISR, and IRR

```
; To read IMR simply:
MOV DX, PIC_PORT1
 IN      AL, DX
; To read ISR
MOV DX, PIC_PORT0
 MOV  AL, 00001011B         ; Select ISR using OCW3
OUT    DX, AL
 IN      AL, DX
; No need to write an OCW3 before every ISR read, as long as the ISR has been previously selected by OCW3
; To read IRR
MOV   DX, PIC_PORT0
MOV   AL, 00001010B         ; Select IRR using OCW3
OUT    DX, AL
IN       AL, DX
; No need to write an OCW3 before every IRR read, as long as the IRR has been previously selected by OCW3
```

8259A: Interrupt Request Register (IRR)

- The IRR is a transparent latch. Bits of IRR are frozen by the $\overline{FREEZE}$ signal starting at the beginning of first $\overline{INTA}$ pulse till the end of the second $\overline{INTA}$ pulse.
- Whatever interrupt requests arrive at the IRR will be latched transparently.
- If the interrupt request is not masked as per the Interrupt Mask Register (IMR), it will reach to the Priority Resolver (PR).
- If new request's priority is higher than the interrupt under service, INT signal is raised by the PR to make a request to the CPU.
- After some time, the CPU sends the first $\overline{INTA}$ pulse. Now the requests in IRR are frozen, and the highest priority interrupt is identified by the PR. The corresponding bit in the ISR is set, and the corresponding bit of IRR is cleared.
- During the second $\overline{INTA}$ pulse, the vector type is sent out, and then the INT signal is lowered.

8259A: In-Service Register (ISR)

- The ISR records the requests currently being serviced.
- It enables the 8259A to check whether any of the incoming new requests has higher priority than the priority of the currently in-service requests. If so, the 8259A will again raise the INT signal and wait for $\overline{\text{INTA}}$.
    - The ISR bit(s) get cleared:
- At the end of the Interrupt Service Routine, there should be an instruction to send End Of Interrupt (EOI) command to the 8259A. Then the highest priority ISR bit gets reset.
- Note: If End Of Interrupt (EOI) is not issued at the end of ISR, all interrupt requests of equal or lower priority remain blocked until the ISR bit is cleared.
- For nesting of interrupts, ISR bit is reset at the end of the second $\overline{\text{INTA}}$ pulse (It is a programmable feature called Automatic End Of Interrupt (AEOI)).

8259 Modes of operation

Fully Nested Mode

- This mode is entered after initialization unless another mode is programmed. After the initialization sequence, IR0 has the highest priority and IR7 the lowest. Priorities can be changed with the rotating priority mode.
- When an interrupt is acknowledged the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service register (IS0-IS7) is set.
- This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine, or if AEOI (Automatic End of Interrupt) bit is set, until the trailing edge of the last $\overline{\text{INTA}}$.

- While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microprocessor internal Interrupt enable flip-flop has been re-enabled through software).

8259A: Special Fully Nested Mode

- In the cascade mode, if a slave receives a higher priority interrupt request than the one which is in service (through the same slave), it won't be recognized by the master. This is because the masters ISR bit is set, ignoring all requests of equal or lower priority.
- The special fully nested mode is programmed in the master only. In this mode the master will ignore only those requests of lower priority than the set ISR bit and will respond to all requests of equal or higher priority. Thus if a slave receives a higher priority request than the one in service, it will be recognized.
- The software must determine if any other slave interrupts are still in service before issuing an EOI command to the master. This is done by resetting the appropriate slave ISR bit with an EOI and then reading its ISR. If the ISR contains all zeros, there are not any interrupts from the slave in service and an EOI command can be sent to the master. If the ISR is no all zeros, an EOI should not be sent to the master.

End of Interrupt (EOI)

- The In Service (IS) bit can be reset either automatically or by a command word that must be issued to the 8259A. An EOI command must be issued twice if in the Cascade mode, once for the master and once for the corresponding slave.
- There are two forms of EOI command: Specific and Non-Specific.
- When a Non-Specific EOI command is issued the 8259A will automatically reset the highest IS bit of those that are set. A non-specific EOI can be issued with OCW2 (EOI = 1, SL = 0, R = 0).
- When a mode is used which may disturb the fully nested structure, the 8259A may no longer be able to determine the last level acknowledged. In this case a Specific End of Interrupt must be issued which includes as part of the command the IS level to be reset. A specific EOI can be issued with OCW2 (EOI = 1, SL = 1, R = 0, and L0-L2 is the binary level of the IS bit to be reset).
- It should be noted that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the 8259A is in the Special Mask Mode.

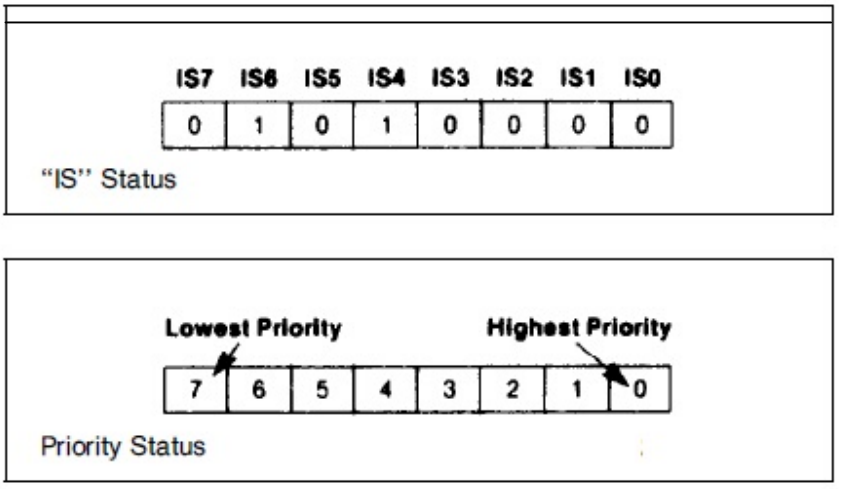
Automatic End of Interrupt (AEOI) Mode

- If AEOI = 1 in ICW4, then the 8259A will operate in AEOI mode continuously until reprogrammed by ICW4, in this mode the 8259A will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse (secondpulse in 8086).
- Note that from a system standpoint, this mode should be used only when a nested multilevel interrupt structure is not required within a single 8259A.

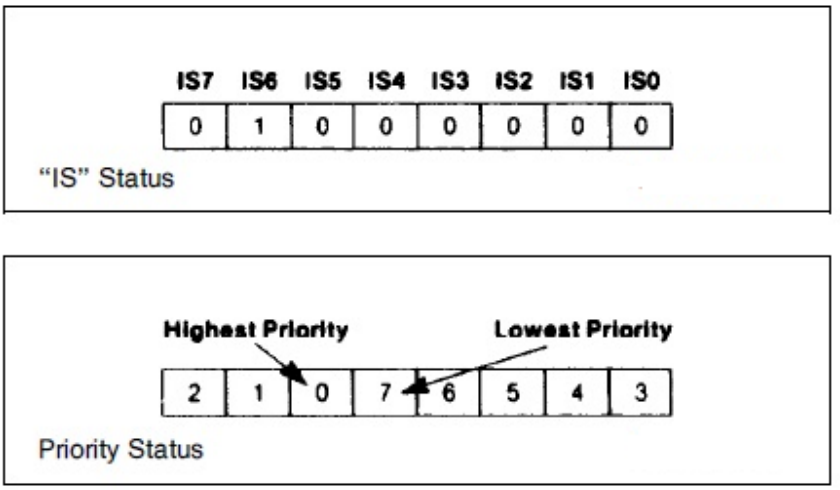
- The AEOI mode can only be used in a master 8259A and not a slave.

Automatic Rotation (Equal Priority Devices)

- In some applications there are a number of interrupting devices of equal priority. In Automatic Rotation mode a device, after being serviced, receives the lowest priority, so a device requesting an interrupt will have to wait, in the worst case until each of 7 other devices are serviced at most once. For example, if the priority and "in service" status is:
    - Before Rotate (IR4 the highest priority requiring service)



    - After Rotate (IR4 was serviced, all other priorities rotated correspondingly)



- There are two ways to accomplish Automatic Rotation using OCW2,
    1. Rotation on Non-Specific EOI Command (R = 1, SL = 0, EOI = 1) and
    2. Rotate in Automatic EOI Mode which is set by (R = 1, SL = 0, EOI = 0) and cleared by (R = 0, SL = 0, EOI = 0).

Specific Rotation (Specific Priority)

- The programmer can change priorities by programming the bottom priority and thus fixing all other priorities; i.e., if IR5 is programmed as the bottom priority device, then IR6 will have the highest one.
- The Set Priority command is issued in OCW2 where: EOI =0, R = 1, SL = 1, L0-L2 is the binary priority level code of the bottom priority device.

Buffered Mode

- The Buffered Mode is useful in large systems where buffering is required on data bus.
- In Buffered Mode, whenever the 8259A's data bus output is enabled, its $\overline{SP}/\overline{EN}$ will goes low. Therefore, this signal is used to enable the direction of (additional) bidirectional data buffers.
- In Buffered Mode, the user can program each individual 8259A as a master or a slave (ICW4).

Interrupt Masks

- Each Interrupt Request input can be masked individually by the Interrupt Mask Register (IMR) programmed through OCW1.
- Each bit in the IMR masks one interrupt channel if it is set (1).
- Bit 0 masks IR0, Bit 1 masks IR1 and so forth. Masking an IR channel does not affect the other channels operation.

Special Mask Mode

- In the special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level and enables interrupts from all other levels (lower as well as higher) that are not masked.
- Thus, any interrupts may be selectively enabled by loading the mask register.
- The special Mask Mode is set by OWC3 where: SSMM = 1, SMM = 1, and cleared where SSMM = 1, SMM = 0.

OPERATION OF 8259 PIC WITH 8086 IN CASCADED MODE

- The 8259 PIC can be easily interconnected in a system of one master with up to 8 slaves to handle up to 64 priority levels.
- CAS2—CAS0are outputs for the master PIC and inputs for the slave PICs.
- Through software initialization, the master PIC is informed which of its IR inputs are connected to the slave PICs.

- The master controls the slaves through three cascade lines CAS2—CAS0 which acts as chip select inputs for the slaves.

  For example:

  If two slaves are connected with the master using $IR_1$and IR_4, then which one is selected depends on the contents on CAS_2—CAS_0. If CAS_2—CAS_0= 100, then the device connected with the IR_4 interrupt level is selected.
- In this mode, the slave interrupt outputs (INT) are connected to the master interrupt request (IR) inputs.
- In the Non Buffered Mode, the master is designated by $\overline{SP}$=Vcc and a slave is designated by $\overline{SP}$=GND. In the Buffered Mode, this is done by software (ICW4).
- Master PIC INT is connected to the CPU's INTR.
- CPU's $\overline{INTA}$ is connected to all PICs $\overline{INTA}$.

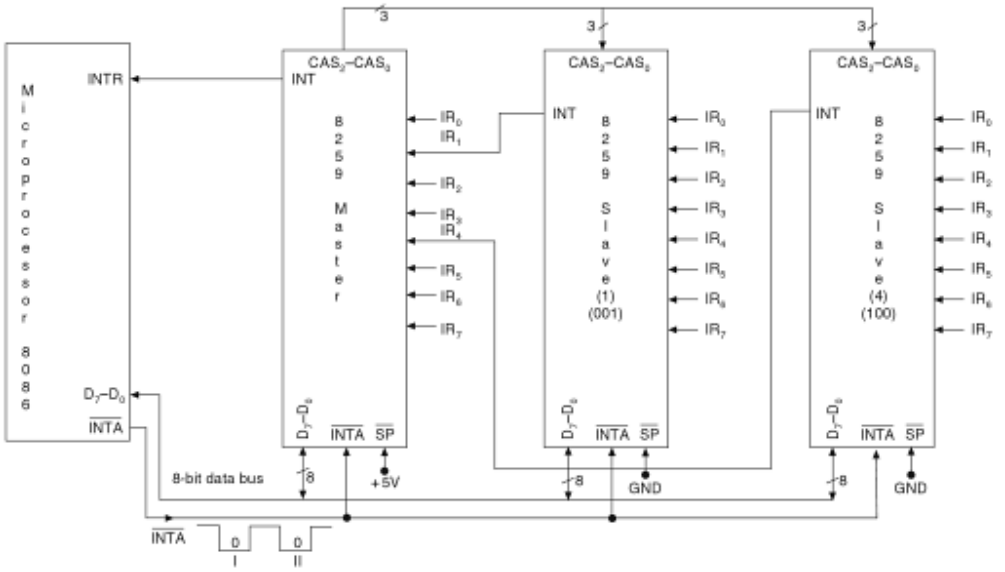


FIGURE Cascade mode of 8259 PIC with microprocessor 8086.

8259A: Operation Sequence in cascade Mode

- Slave PIC receives an interrupt request on one of its IR pins.
- Slave PIC raises INT output. Therefore, the master PIC receives an interrupt request on one of its IR pins.
- The master PIC raises the INT signal.
- The CPU gets INTR, and sends the 1st $\overline{INTA}$pulse.
- The master PIC sends cascade address (CAS2—CAS0) and sets its ISR bit.

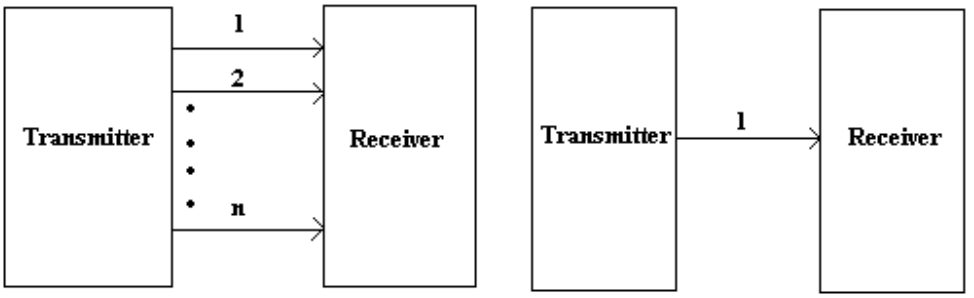
- The slave PIC with an ID matching to the address on CAS2—CAS0lines respond during the 2nd $\overline{\text{INTA}}$ pulse and sends its vector type. In addition, it sets its ISR bit.
- The master and the slave PICs de-activate their INT outputs.
- Note: ISR bit of both master and slave get set. This means two EOI commands must be issued (if not in the AEOI mode), one for the master and one for the corresponding slave.

Fundamentals of Serial Communication: - Most of Microprocessors are designed for Parallel Communication.

In Parallel Communication number of lines required to transfer the data dependent on the number of bits to be transmitted. For example to transfer a Byte of data 8- lines are required and all 8-bits are transferred simultaneously.

Transmitting data over a long distance, using Parallel Communication is impractical due to the increase in cost of cabling. Parallel Communication is also not practical for devices such as Cassette tapes or a CRT terminal. In such Situations, Serial Communication is used. In Serial Communication One bit is transferred at a time over a single line.

'Baud Rate' is the rate at which serial data is being transferred and in general measured in bit/sec



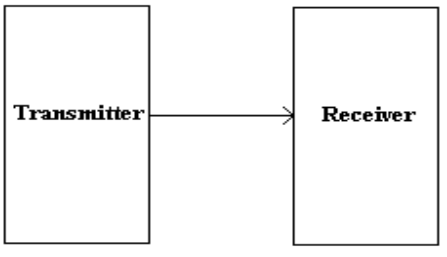
Parallel Communication Serial Communication

Classification of Serial Data Transmissions based on the direction of signal between two linked devices: -

1. Simplex Serial Communication
2. Half-Duplex Serial Communication
3. Full-Duplex Serial Communication

1. Simplex Serial Communication: - In Simplex Serial Data Transmission the data transfer takes place only in one direction. There is no possibility of data transfer in other direction.

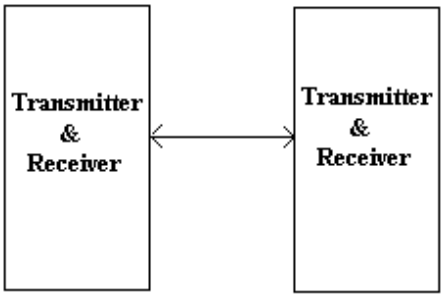
Ex: -Data Transfer from CPU to Printer or Monitor.



Simplex Serial Communication

2. **Half-Duplex Serial Communication: -** In Half Duplex Serial Data Transmission the data transfer takes place in both the directions but it is not simultaneous.

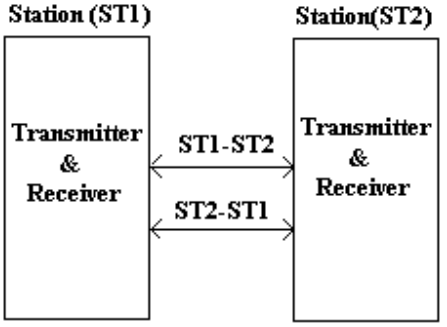
Ex: - Walkie talkie System



Half- Duplex Serial Communication

3. **Full-Duplex Serial Communication: -** In Full-Duplex Serial Data Transmission the data transfer takes place in both the directions simultaneous.

Ex: - Telephone Network

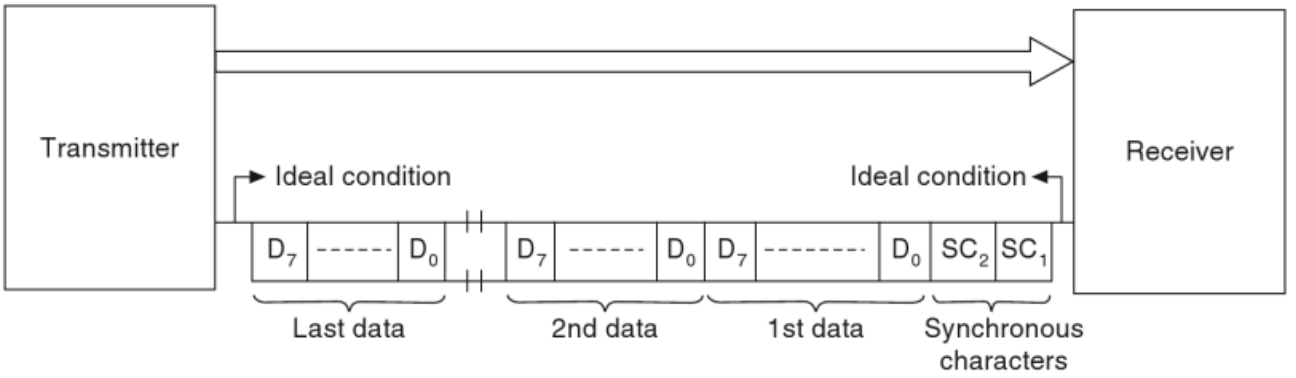


Full-Duplex Serial Communication

Serial Data Transmission Modes: -

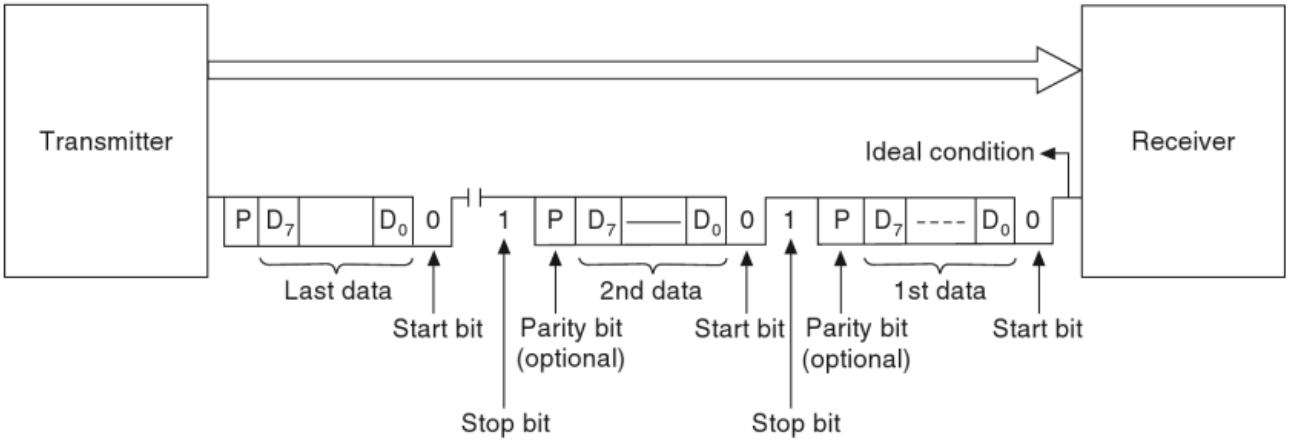
1. Synchronous Serial Data Transmission Mode
2. Asynchronous Serial Data Transmission Mode

1. **Synchronous Serial Data Transmission Mode: -** The features of this mode
   a. The data is sent in blocks at a constant rate. i.e. the frequencies of transmission and reception are same.
   b. Transmission and reception take place simultaneously.
   c. The beginning of a block is identified with specific bytes or bit pattern called Sync Character.
   d. Synchronous Data Transmission is used for high transmission speeds of more than 20Kbps.
   e. The data transmission takes place with a reference from transmitter to receiver.



Synchronous data format.

2. **Asynchronous Serial Data Transmission Mode: -** The features of this mode
   - **a.** In this mode each data character has a bit to identify its start and one or two bits to identify its end. Here each character is identified individually.
   - **b.** The character can be sent at any time, without checking the receiver.
   - **c.** Reception and transmission are not synchronized.
   - **d.** Asynchronous serial data format



Asynchronous data format.

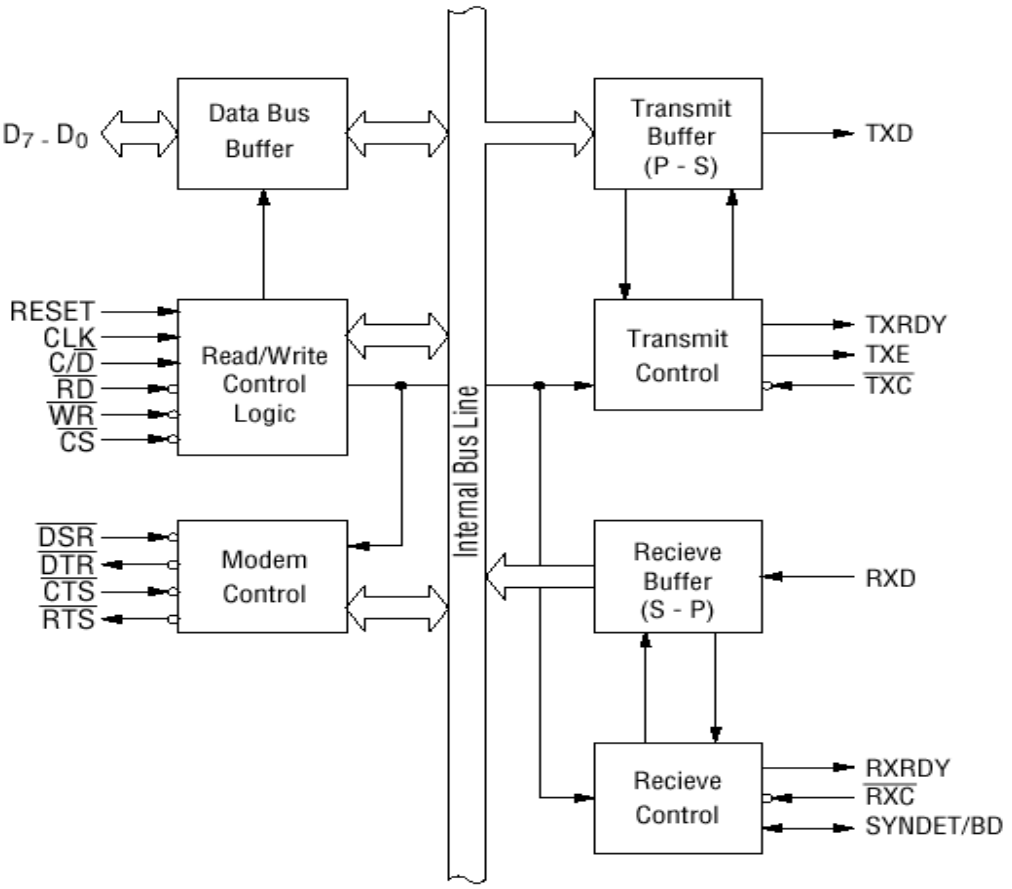
When no data is being set, the signal line is in a constant high level. The first data character is indicated by the line going low for one bit duration and is usually called, 'Start bit'. The data bits are then sent out on the line one after another. The Least Significant bit is sent out first, the data bits are followed by the optional parity bit which is used to check for errors in received data at receiving end.

After the data bits and the parity bit, the signal bit is made high for at least one bit duration to identify the end of the character. This is referred as 'Stop bit' some system use two stop bits also.

Differences between Synchronous and Asynchronous Serial Data Transfer

Synchronous data transfer	*Asynchronous data transfer*
1. It is used to transfer a group of characters at a time.	It is used to transfer one character at a time
2. Used for high data transfer rates ≥ 20 Kbps	Used for data transfer rates ≤ 20 Kbps.
3. Synchronous characters are transmitted along with the group of characters	Synchronous characters are not transmitted along with the characters
4. No start and stop bits for each character is used	Start and stop bit for each character is present which forms a frame
5. One clock is used for both transmitter and receiver	Two separate clock inputs can be used for transmitter and receiver
6. Since synchronization is involved, this can be implemented by using hardware only	No synchronization is required hence hardware and software implementation is possible.

Architecture of 8251 USART: -



The USART accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream for transmission. Simultaneously, it can receive serial data streams and convert them into parallel data characters for the CPU.

The 8251 USART supports both Synchronous and Asynchronous data transmission forms and is Programmable. It supports Full-Duplex Serial Data Transmission and Reception with variable Baud rates. Common Baud Rates are 300, 600, 1200, 2400, 4800, 9600, 19200 etc.

The functional block diagram of 825 1A consists five sections. They are:

1. Data bus buffer
2. Read/Write control logic
3. Transmitter Section
4. Receiver Section
5. Modem control.

Data Bus Buffer: - This block provides the link between the internal architecture of 8251 with the system bus, It take the Command Words from the Processor, It send the Status information to the Processor and it read or write data from or to the Processor by using data lines D7 – D0.

Read / Write Control Logic: - This block is treated as the heart of the 8251 USART, because it controls the internal and external operation of the device based on the control signals and the command words which are received from the processor. It receives the different control signals from the processor and those are, RESET, CLK, $\overline{RD}$, $\overline{WR}$, $\overline{CS}$, C/$\overline{D}$.

$\overline{CS}$	C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	
1	×	×	×	Data Bus 3-State
0	×	1	1	Data Bus 3-State
0	1	0	1	Status → CPU
0	1	1	0	Control Word ← CPU
0	0	0	1	Data → CPU
0	0	1	0	Data ← CPU

RESET (Reset)

A "high" on this input forces the 8251A into an "Idle" mode. The device will remain at "Idle" until a new set of control words is written into the 8251A to program its functional definition.

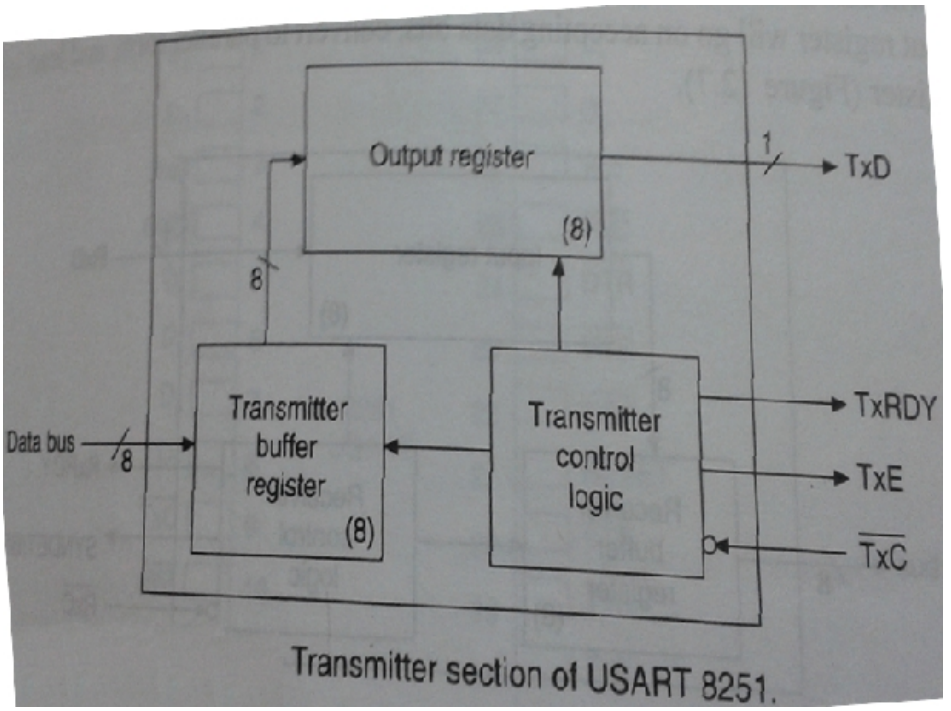
CLK (clock): CLK signal is used to generate internal device timing. CLK signal is independent of RXC or TXC. However, the frequency of CLK must be greater than 30 times the RXC and TXC at Synchronous mode and Asynchronous "x1" mode, and must be greater than 5 times at Asynchronous "x16" and "x64" mode.

Transmitter section of 8251 USART: - This section contains two different blocks, those are

i. Transmit Buffer ii. Transmit Control

By using these two blocks the 8251 send the data information to the external devices.

a. The transmitter section accepts parallel data from CPU and converts them into serial data.
b. The transmitter section is double buffered, i.e., it has a buffer register to hold an 8-bit parallel data and another register called output register to convert the parallel data into serial bits.
c. When output register is empty, the data is transferred from buffer to output register. Now the processor can again load another data in buffer register.
d. If buffer register is empty, then TxRDY is goes to high.
e. If TxRDY = I, this indicates that the transmit buffer register is empty and microprocessor is ready for data transfer to transmit buffer register
f. If output register is empty then TxEMPTY goes to high.
g. The clock signal, TxC (low) controls the rate at which the bits are transmitted by the USART.
h. The clock frequency can be 1, 16 or 64 times the baud rate.



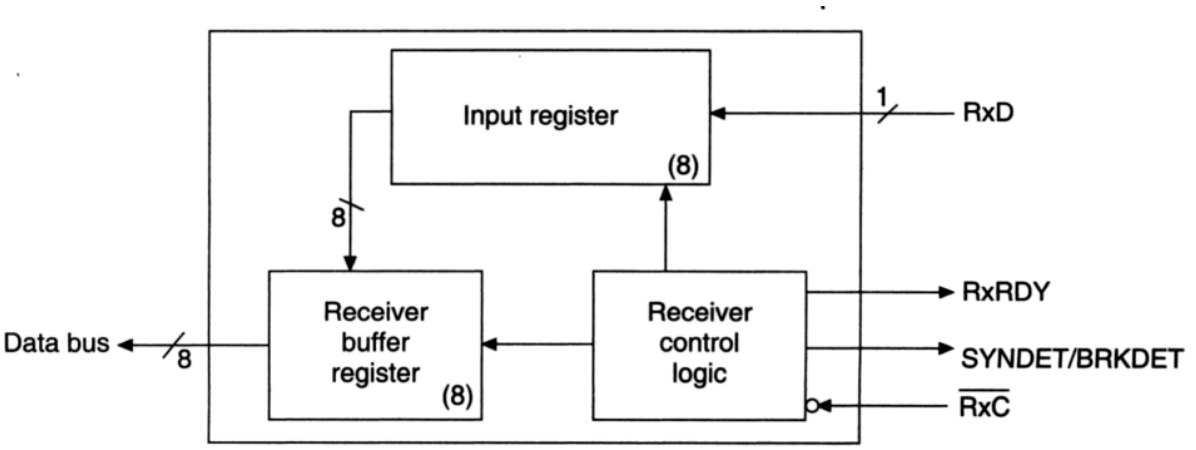
Transmitter section of USART 8251.

Receiver section of 8251 USART: - This section also contains two different blocks, those are

i. Receive Buffer ii. Receive Control

By using these two blocks the 8251 receive the data information from the external devices.

a. The receiver section accepts serial data and convert them into parallel data
b. The receiver section is double buffered, i.e., it has an input register to receive serial data and convert to parallel, and a buffer register to hold the parallel data.
c. When the RxD line goes low, the control logic assumes it as a START bit, waits for half a bit time and samples the line again.
d. If the line is still low, then the input register accepts the following bits, forms a character and loads it into the buffer register.
e. The CPU reads the parallel data from the buffer register.
f. When the input register loads a parallel data to buffer register, the RxRDY line goes high to signal microprocessor about availability of data byte to be read by microprocessor.
g. The clock signal RxC (low) controls the rate at which bits are received by the USART.
h. During asynchronous mode, the signal SYNDET/BRKDET will indicate the break in the data transmission.
i. During synchronous mode, the signal SYNDET/BRKDET will indicate the reception of synchronous character.



Receiver section of USART 8251.

Modem Control: -

For sending data over long distances the telephone lines are used. The telephone lines are analog in nature, so MODEMs (Modulator—Demodulator) are used to convert digital data to analog data. To control or communicate MODEMs with 8251, the 8251 USART provides a block called MODEM control.

The MODEM control block uses different lines such as RTS, CTS, DTR, and DSR for MODEM.

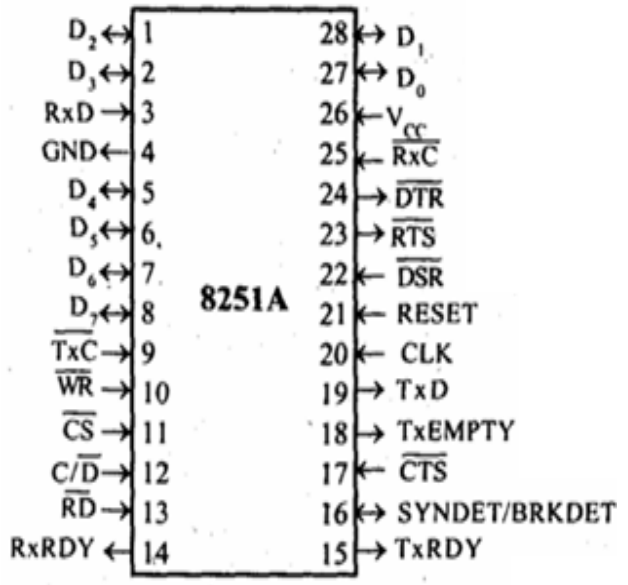
(a) **DTR (Data terminal ready):** When the terminal is made ON, it will perform different operations. When it is ready for data transmission/reception, it generates a signal DTR to indicate its readiness for data transfer.

(b) **DSR (Data set ready):** This is an input port for MODEM interface and it is an 'active low' signal. Whenever Modem has a data set it sends that information to 8251 by using this signal.

(c) **CTS (Clear to send):** When the MODEM is ready to transmit data, it assets CTS signal to terminal. The terminal upon receiving CTS sends serial data character to the MODEM.

(d) **RTS (Request to send):** When the terminal is ready to transmit data and has a data character to be transmitted, the terminal assets a signal RTS to the MODEM.

Pin Description: -



D 0 - D 7 Data Lines: - This is bidirectional data bus which receives control words and data from the CPU and sends status words and received data to CPU.

RESET: - A "High" on this input forces the 8251 into "reset status." The device waits for the writing of "mode instruction control words." The min. reset width is six clock inputs during the operating status of CLK.

CLK: - CLK signal is used to generate internal device timing. CLK signal is independent of RXC or TXC. However, the frequency of CLK must be greater than 30 times the RXC and TXC at Synchronous mode and Asynchronous "x1" mode, and must be greater than 5 times at Asynchronous "x16" and "x64" mode.

WR (Write): -This is the "active low" input terminal which receives a signal for writing transmit data and control words from the CPU into the 8251.

RD (Read): -This is the "active low" input terminal which receives a signal for reading receive data and status words from the 8251.

C/D (Command or Status / Data): -This is an input terminal which receives a signal for selecting data or command words and status words when the 8251 is accessed by the CPU. If C/D = low, data will be accessed. If C/D = high, command word or status word will be accessed.

CS (Chip Select): -This is the "active low" input terminal which selects the 8251 at low level when the CPU accesses.

TXD (Transmit Data): -This is an output terminal for transmitting data from which serial-converted data is sent out to external devices.

TXRDY (Transmit Ready): -This is an output terminal which indicates that the 8251is ready to accept a transmitted data character from CPU. In another way there is no data at data bus buffer this signal is active.

TXEMPTY (Transmit Empty): -This is an output terminal which indicates that the 8251 has transmitted all the characters and had no data character in the transmitter section it means there is no data in both Data bus buffer and the transmit buffer. If the CPU writes a data character, TXEMPTY will be reset by the leading edge of WR signal.

TXC (Transmit Clock): -This is a clock input signal which determines the transfer speed of transmitted data. In "synchronous mode," the baud rate will be the same as the frequency of TXC. In "asynchronous mode", it is possible to select the baud rate factor by mode instruction. It can be 1, 1/16 or 1/64 the TXC. The falling edge of TXC sifts the serial data out of the 8251.

RXD (Receive Data): -This is a terminal which receives serial data from External device.

RXRDY (Receive Ready): - This is a terminal which indicates that the 8251 contains a character that is ready to READ. If the CPU reads a data character, RXRDY will be reset by the leading edge of RD signal. Unless the CPU reads a data character before the next one is received completely, the preceding data will be lost. In such a case, an overrun error flag status word will be set.

RXC (Receive Clock): - This is a clock input signal which determines the transfer speed of received data. In "synchronous mode," the baud rate is the same as the frequency of RXC. In "asynchronous mode," it is possible to select the baud rate factor by mode instruction. It can be 1, 1/16, 1/64 the RXC.

SYNDET/BD (Sync Detect/ Break Detect): - This is a terminal whose function changes according to mode. In "internal synchronous mode." this terminal is at high level, if sync characters are received and synchronized. If a status word is read, the terminal will be reset. In "external synchronous mode, "this is an input terminal. A "High" on this input forces the 8251 to start receiving data characters.

In "asynchronous mode," this is an output terminal which generates "high level “output upon the detection of a "break" character if receiver data contains a "low-level" space between the stop bits of two

continuous characters. The terminal will be reset, if RXD is at high level. After Reset is active, the terminal will be output at low level.

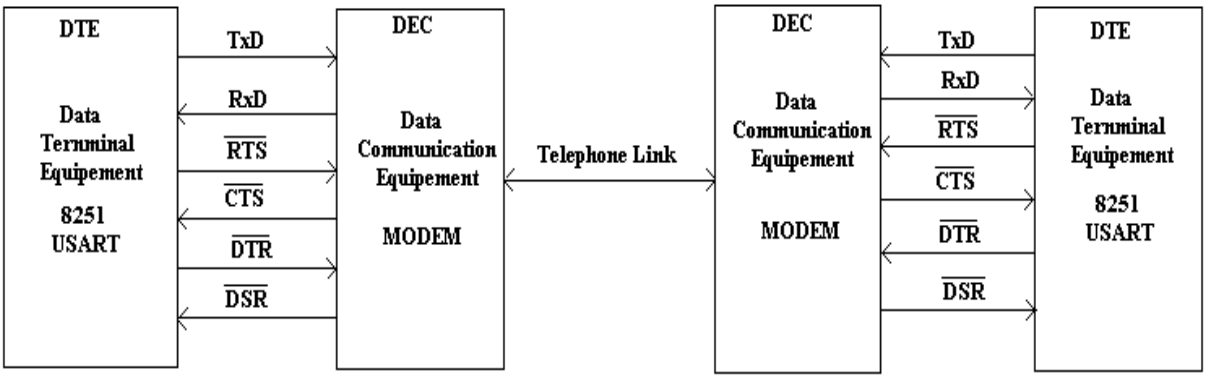
DSR (Data Set Ready): -This is an input port for MODEM interface and it is an 'active low' signal. Whenever Modem has a data set it sends that information to 8251 by using this signal.

DTR (Data Terminal Ready): -This is an output port for MODEM interface and it is also an 'active low' signal. If 8251 is ready to send or receive a data from Modem it sends its readiness information to Modem by using this signal. It is possible to set the status of DTR by a command.

CTS (Clear to Send): - This is an input terminal for MODEM interface which is used for controlling a transmit circuit. The terminal controls data transmission if the device is set in "TX Enable" status by a command. Data is transmittable if the terminal is at low level.

RTS (Request to Send): -This is an output port for MODEM interface. It is possible to set the status RTS by a command.

Digital Data Transmission Using MODEMs & Standard Telephone Links: -



Digital Data Transmission Using MODEMs & Standard Telephone Links

Control Words of 8251 USART: - There are two types of control words that are supported by 8251 USART. Those are,

1. Mode Instruction Control Word 2. Command Instruction Control Word

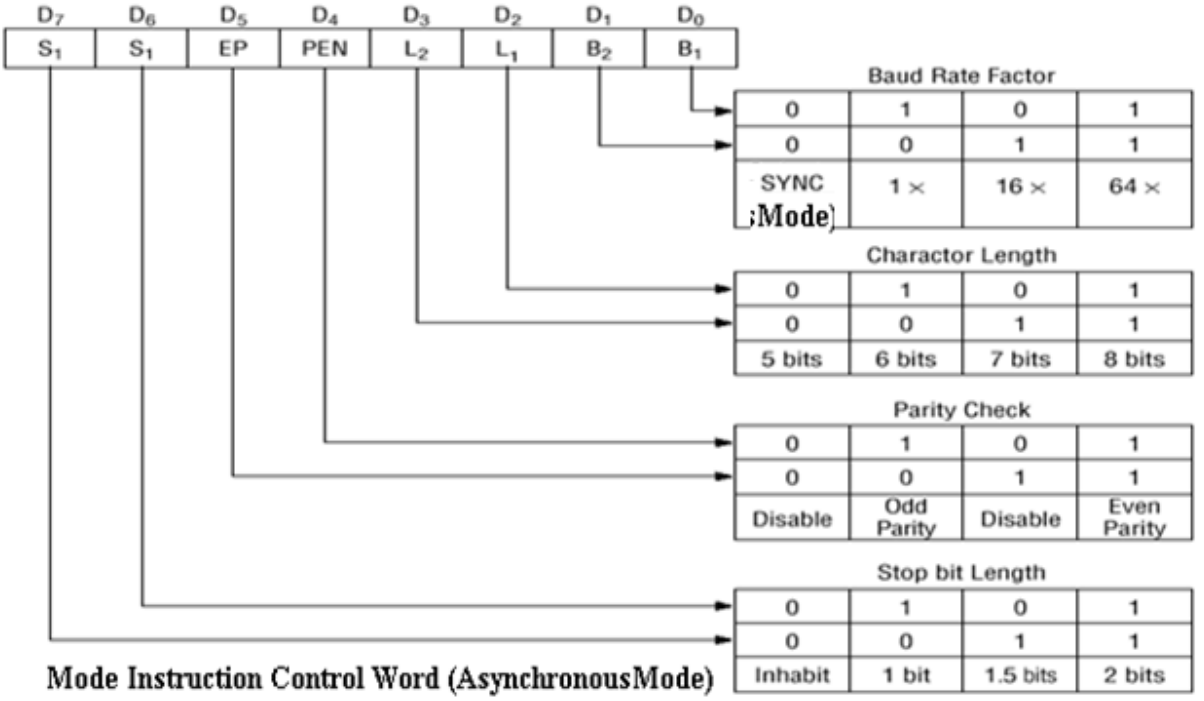
1) Mode Instruction Control Word: - Mode instruction is used for setting the function of the 8251. Mode instruction will be in "wait for write" at either internal reset or external reset. That is, the writing of a control word after resetting will be recognized as a "mode instruction."

Items set by mode instruction control word are as follows:

a. Synchronous/asynchronous mode
b. Stop bit length (asynchronous mode)
c. Character length

d. Parity bit
e. Baud rate factor (asynchronous mode)
f. Internal/external synchronization (synchronous mode)
g. Number of synchronous characters (Synchronous mode)

The bit configuration of mode instruction is shown in Figures 2 and 3. In the case of synchronous mode, it is necessary to write one-or two byte sync characters. If sync characters were written, a function will be set because the writing of sync characters constitutes part of mode instruction.



Mode Instruction Control Word (AsynchronousMode)

Synchronous Mode

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SCS	ESD	EP	PEN	L_2	L_1	0	0

ESD (External synchronous detect):

ESD	*Operation*
1	SYNDET is an input (Synch character is check by externally)
0	SYNDET is an output (Synch character is check by internally)

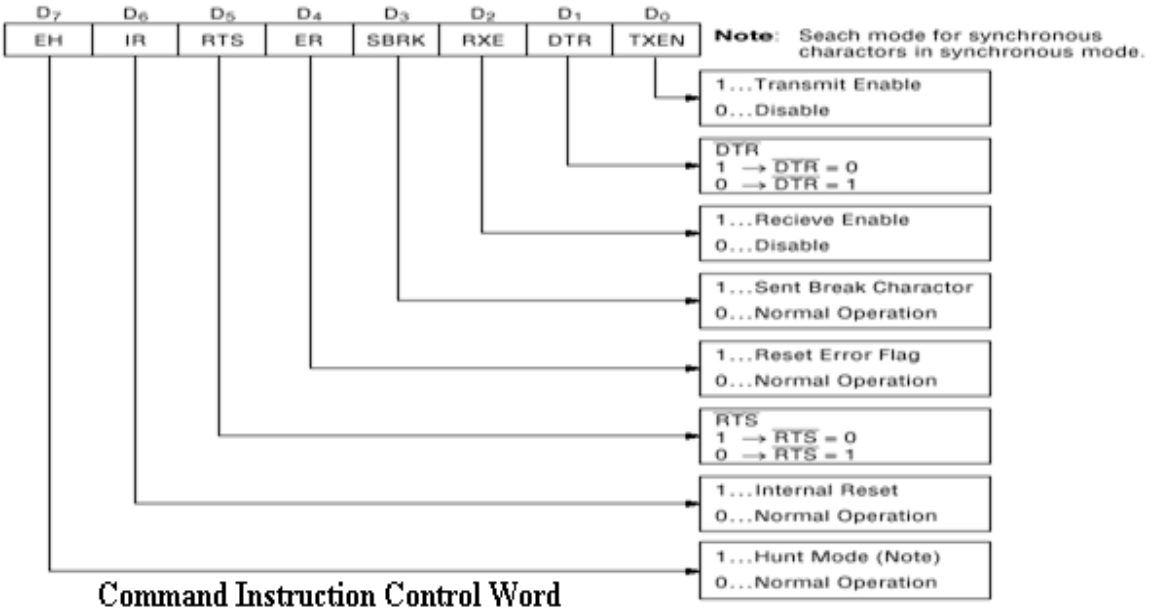
SCS (Single character synchronously):

SCS	*Operation*
1	Single Synchronous character
0	Double Synchronous character

2) Command Instruction Control word: - Command is used for setting the operation of the 8251. It is possible to write a command whenever necessary after writing a mode instruction and sync characters.

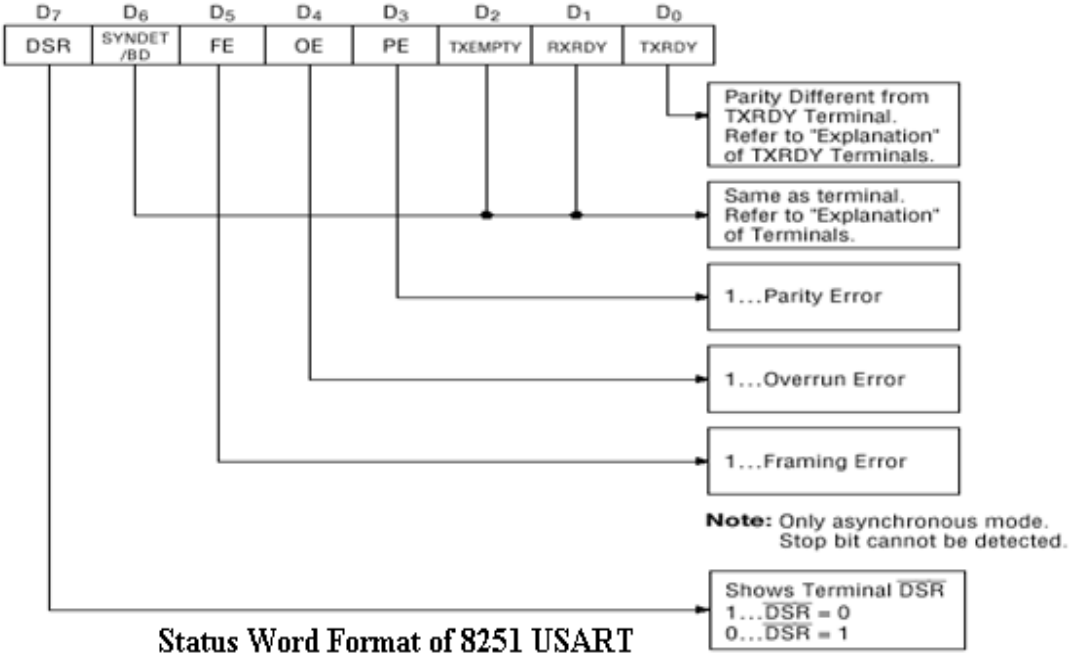
Items to be set by command are as follows:

a.Transmit Enable/Disable
b. Receive Enable/Disable
c. DTR, RTS Output of data
d. Resetting of error flag.
e. Sending to break characters
f. Internal resetting
g. Hunt mode (synchronous mode)



Command Instruction Control Word

Status Word Format: - It is possible to see the internal status of the 8251 by reading a status word. The bit configuration of status word is shown in Fig.



Status Word Format of 8251 USART

Error Definitions: - There are three types of errors are encountered I 8251 USART Serial communication operations. Those are

i. Parity Error ii. Overrun Error iii. Framing Error

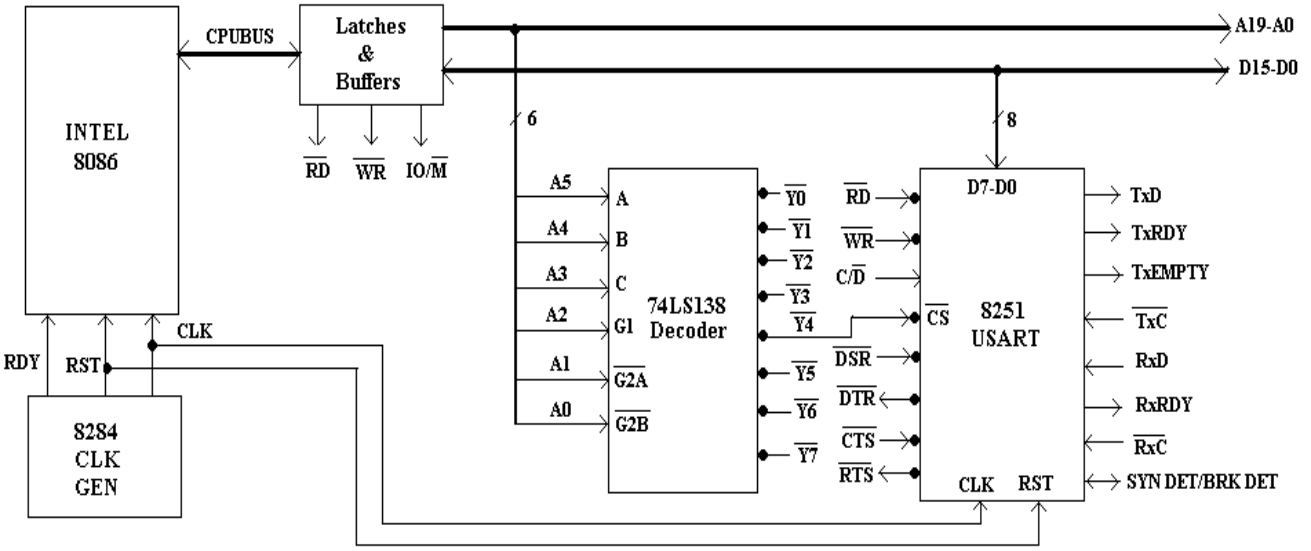
i. Parity Error: - At the time of transmission of data an even or odd parity bit is inserted in the data stream. At the receiver end, if the parity of the character does not match with the predefined parity, the parity error occurs.

ii. Overrun Error: - In the receiver section, received character is stored in the receive buffer and makes RXRDY line high. The CPU is to read this character before reception of next character. But if CPU fails in reading the character loaded in the receive buffer, the next received character replaces the previous one and the overrun error occurs.

iii. Framing Error: - If the valid Stop bit is not detected at the end of each character then Framing error occurs.

Parity and Overrun Errors both are appears in both modes of operation (In Asynchronous and in Synchronous serial communication modes). But the Framing Error only appears in only in Asynchronous Mode of operation, but it is not appears in Synchronous mode of operation.

Interfacing of 8251 USART to 8086 Processor: -



Interfacing of 8251USART to 8086 Microprocessor

Address Map: - The address of 8251 USART

A7	A6	A5	A4	A3	A2	A1	A0	Hex Address
0	0	1	0	0	1	0	0	24H