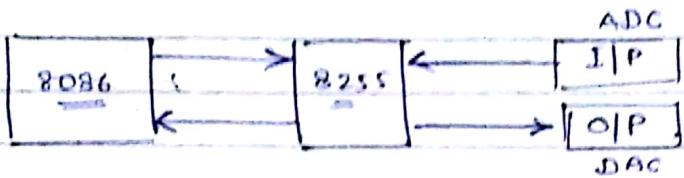


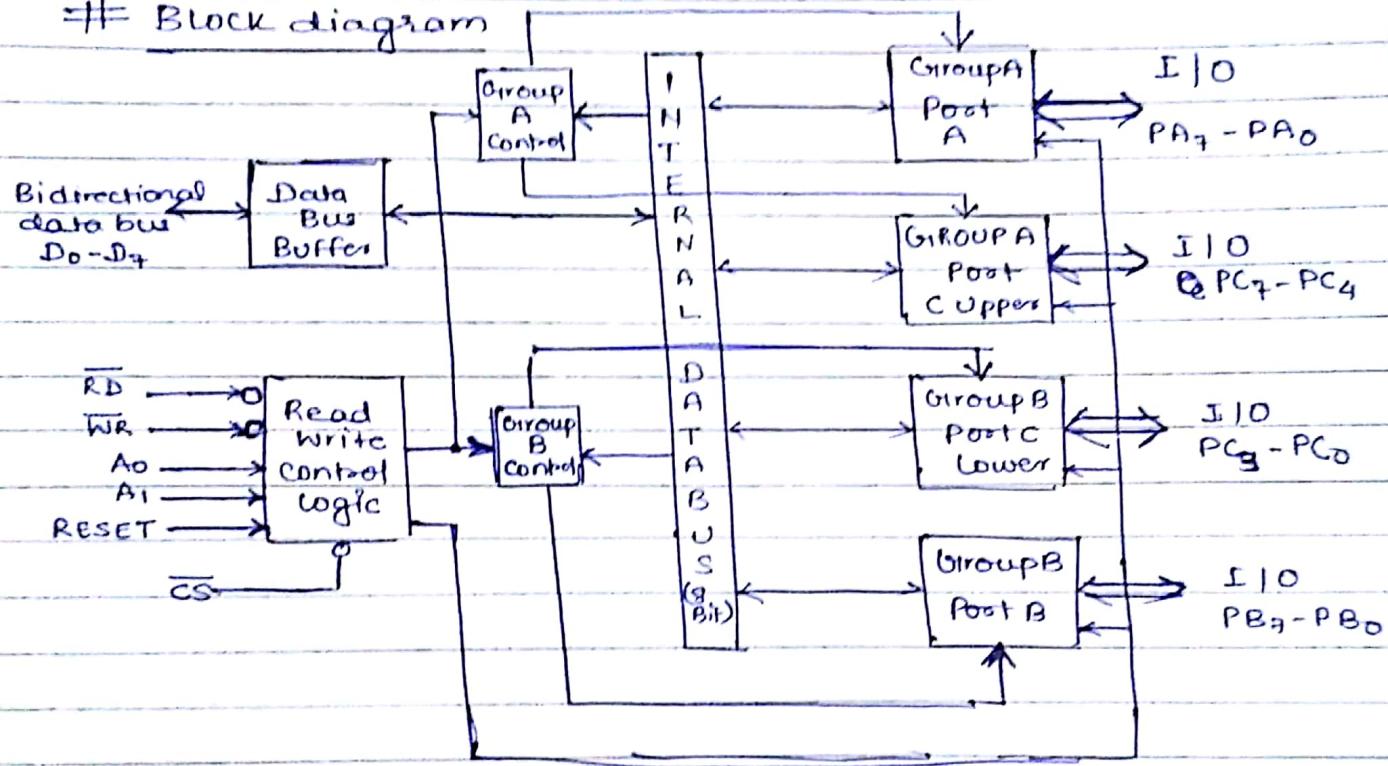
## Unit 3

# Basic block diagram of interfacing



8255 - Programmable Peripheral Interface (PPI)

# Block diagram



8255 is a general purpose programmable I/O designed to transfer the data from I/O to interrupt I/O under certain cond<sup>n</sup> as required.

Q1

It consists of I/O ports. These I/O ports can be used for connecting input/output devices (Peripherals) with microprocessor.

It is used for parallel data transfer.

It is a 40 pin IC and have 3 ports A, B & C 8 bit each

It consists of three 8-bit bidirectional I/O ports (24 ports) which can be configured as per requirement.

### Ports of 8255

- (i) PORT A - Contains one 8-bit output buffer and one 8-bit input buffer.
- (ii) PORT B - similar to PORT A
- (iii) PORT C - split into two parts i.e. PORT C LOWER (PC<sub>0</sub>-PC<sub>3</sub>) and PORT C UPPER (PC<sub>4</sub>-PC<sub>7</sub>)

These three ports are further divided into two groups:-  
i.e GROUP A → PORT A and PORT C UPPER  
GROUP B → PORT B and PORT C LOWER

### # Pin diagram

PA <sub>3</sub>	1	40	PA <sub>4</sub>
PA <sub>2</sub>	2	39	PA <sub>5</sub>
PA <sub>1</sub>	3	38	PA <sub>6</sub>
PA <sub>0</sub>	4	37	PA <sub>7</sub>
A <sub>1</sub>	5	36	WR
CS	6	35	Reset
RD	7	34	D <sub>0</sub>
A <sub>0</sub>	8	33	D <sub>1</sub>
AO	9	32	D <sub>2</sub>
PC <sub>9</sub>	10	31	D <sub>3</sub>
PC <sub>8</sub>	11	30	D <sub>4</sub>
PC <sub>7</sub>	12	29	D <sub>5</sub>
PC <sub>6</sub>	13	28	D <sub>6</sub>
PC <sub>5</sub>	14	27	D <sub>7</sub>
PC <sub>4</sub>	15	26	Vcc
PC <sub>3</sub>	16	25	PB <sub>9</sub>
PC <sub>2</sub>	17	24	PB <sub>8</sub>
PC <sub>1</sub>	18	23	PB <sub>7</sub>
PC <sub>0</sub>	19	22	PB <sub>6</sub>
PB <sub>2</sub>	20	21	PB <sub>5</sub>

D<sub>7</sub>-D<sub>0</sub> → databus  
Reset → Reset Input  
CS → Chip select  
RD → Read Input  
WR → Write Input  
A<sub>0</sub>-A<sub>1</sub> → Port Address  
PA<sub>7</sub>-PA<sub>0</sub> → Port A  
PB<sub>9</sub>-PB<sub>0</sub> → Port B  
PC<sub>9</sub>-PC<sub>0</sub> → Port C  
VCC → +5VOLT  
GND → 0VOLT

## Data bus buffers

It is a tri-state bidirectional buffer, which is used to interface the 8255 to the system data bus. Data is transmitted or received by the buffer as per instruction by the CPU.

## Read | Write Control logic

responsible for controlling the internal/external transfer of data | control | status word. It accepts inputs from CPU address and control buses.

### CS (chip select)

- enables the communication between 8255 and CPU.  
It is connected to the decoded address and A<sub>0</sub> & A<sub>1</sub> are connected to microprocessor address lines.

CS	A <sub>1</sub>	A <sub>0</sub>	Result
0	0	0	PORTA
0	0	1	PORTB
0	1	0	PORTC
0	1	1	<u>Control Word Register</u>
1	x	x	8255 is not selected.

WR - enables the write operation. When this signal goes low, the microprocessor writes into selected I/O port.

RD - enables the read operation. When low, the microprocessor reads data from selected I/O ports of 8255.

Reset - Clears the control register and sets all ports in the input mode.

A<sub>0</sub> & A<sub>1</sub> — These input signals along with RD and WR inputs control the selection of control/status word register or one of the 3 ports.

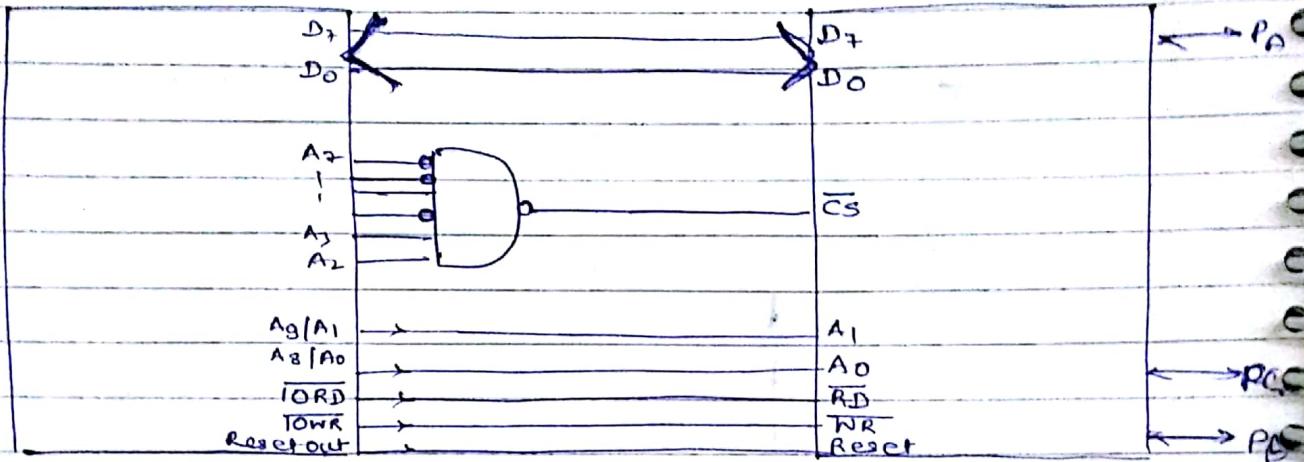
A <sub>1</sub>	A <sub>0</sub>	RD	WR	CS	Result
0	0	0	1		
0	1	0	1		
1	0	0	1		
0	0	0	0		
0	1	1			
1	0	1			
1	1	1			

A <sub>1</sub>	A <sub>0</sub>	RD	WR	CS	Result
0	0	0	1	0	PORT A → databus
0	1	0	1	0	PORT B → databus
1	0	0	1	0	PORT C → databus
1	1	0	1	0	CWR to databus
0	0	1	0	0	databus → PORT A
0	1	1	0	0	databus → PORT B
1	0	1	0	0	databus → PORT C
1	1	1	0	0	databus → CWR

### Features of 8255

- consists of 3 8 bit I/O ports i.e PA, PB, PC.
- Address/databus must be externally demuxed.
- TTL compatible (transistor-to-transistor logic)
- Improved DC driving capability.

## Interfacing of 8255 PPI with 8085



CS						A <sub>1</sub>	A <sub>0</sub>	Hex Address	Selected
A <sub>3</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>				
0	0	1	0	1	1	0	0	2C	PA
0	0	1	0	1	1	0	1	2D	PB
0	0	1	0	1	1	1	0	2E	PC
0	0	1	0	1	1	1	1	2F	CWR

### # Modes of Operation \*

- divided into 2 parts

- parallel I/O mode - (mode 0, mode 1, mode 2)
- BSR mode (Bit Set Reset)

#### (i) Parallel I/O mode

If 8-bit data is to be transferred in parallel b/w I/O device and microprocessor then 8255 will be used in parallel I/O modes.

### Mode 0 - (simple Input/Output mode)

It is used for simple I/O operations without handshaking. PORT A and PORT B can be initialized in Mode 0. PORT C can be used either as a single 8-bit port or it can be used as two 4-bit ports.

The two halfs of Port C are independent so one half can be initialized as input and other half as output.

### Mode 1 (strobed Input/Output)

Port A or Port B can be used for a handshake operation. When port A or port B is operating in handshake mode few lines of port C can be used as handshake signal lines. Each port uses 3 lines from port C as handshake signal lines.

ex when Port A is initialized for handshake input port the pins PC<sub>3</sub>, PC<sub>4</sub>, PC<sub>5</sub> functions as handshake signal and PC<sub>6</sub> & PC<sub>7</sub> are available for use as I/O lines.

If port A is initialized as a handshake output port then port C pins PC<sub>3</sub>, PC<sub>6</sub> & PC<sub>7</sub> function as handshake signal. PC<sub>4</sub> & PC<sub>5</sub> are for use as I/O lines.

8255 is mostly used in Mode 1

### Mode 2 (strobed bidirectional bus Input/Output)

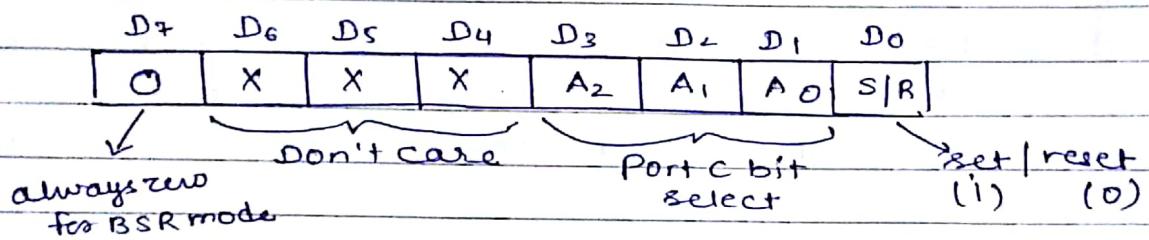
Only port A is initialized in this mode. PORT A can be configured as bidirectional port and Port B either in Mode 0 or Mode 1.

Port A uses 5 signals from PORT C i.e. PC<sub>3</sub>, PC<sub>4</sub>, PC<sub>5</sub>, PC<sub>6</sub> & PC<sub>7</sub> as handshake lines.

## (ii) Bit Set/Reset Mode (BSR mode) IMP

Only PORT C is used in BSR mode. In this mode we can transfer either logic '1' (Set) or logic '0' (reset) on any pin of PORT C without changing the status of rest of the seven pins.

CWR format for BSR mode



- D<sub>7</sub> is always zero.
- A<sub>2</sub>, A<sub>1</sub>, A<sub>0</sub> bits are used to indicate three 3-bit address of the corresponding port C pin

A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	PORt C pin selected
0	0	0	P <sub>C0</sub>
0	0	1	P <sub>C1</sub>
0	1	0	P <sub>C2</sub>
0	1	1	P <sub>C3</sub>
1	0	0	P <sub>C4</sub>
1	0	1	P <sub>C5</sub>
1	1	0	P <sub>C6</sub>
1	1	1	P <sub>C7</sub>

- LSB D<sub>0</sub> bit is transferred on the selected port C pin

D<sub>0</sub> → 1 ⇒ particular bit is set

D<sub>0</sub> → 0 ⇒ particular bit is reset

Ex

Write a BSR-CWR to set PC<sub>7</sub>, PC<sub>3</sub> and clear PC<sub>5</sub>

PC<sub>7</sub>  $\Rightarrow$ 

0	X	X	X	1	1	1	1
---	---	---	---	---	---	---	---

 $\Rightarrow$  OF

PC<sub>3</sub>  $\Rightarrow$ 

0	X	X	X	0	1	1	1
---	---	---	---	---	---	---	---

 $\Rightarrow$  O<sub>7</sub>

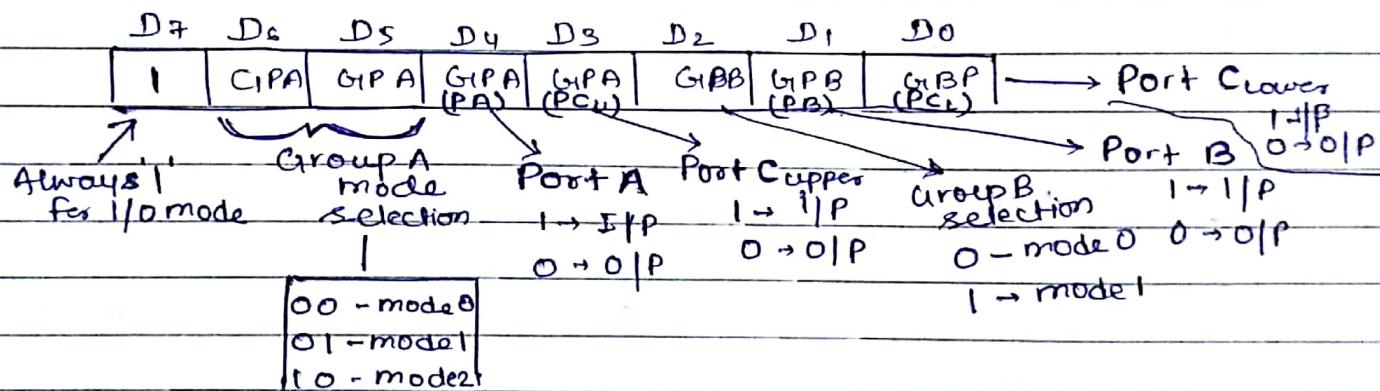
PC<sub>5</sub>  $\Rightarrow$ 

0	X	X	X	1	0	1	0
---	---	---	---	---	---	---	---

 $\Rightarrow$  OA

Imp

# I/O mode CWR (for mode 0, mode 1, mode 2)



Ex Make a CWR where Port def<sup>n</sup> as follows

- PORT A as input port in mode 0
- PORT B as output port in Mode 0
- PORT Cu as input port
- PORT CL as output port

1	0	0	1	1	0	0	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---

Lx 98H

Ex Write inst. to define port A as input port in mode 0,  
Port B as output port in mode 1 and PORT C upper  
as input, 8255 starts from 60H.

→ 

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	1	0	1	0	01

 $\Rightarrow 9C/9DH$   
 ↓ undefined

### Program

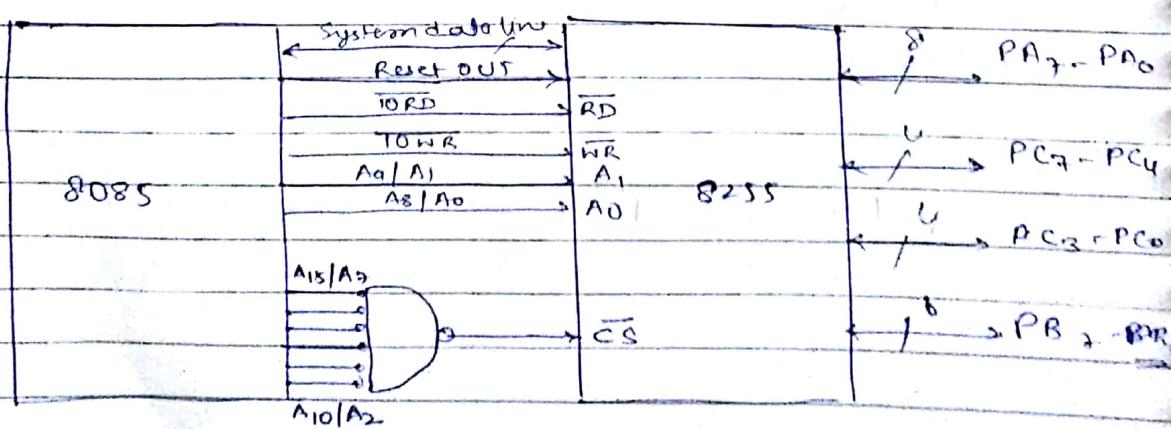
MVI A, 9CH → Load 8bit no. in Accumulator  
 OUT 63H → Load content of Acc. in CWR.

Ex Interface 8255 PPI with microprocessor 8085

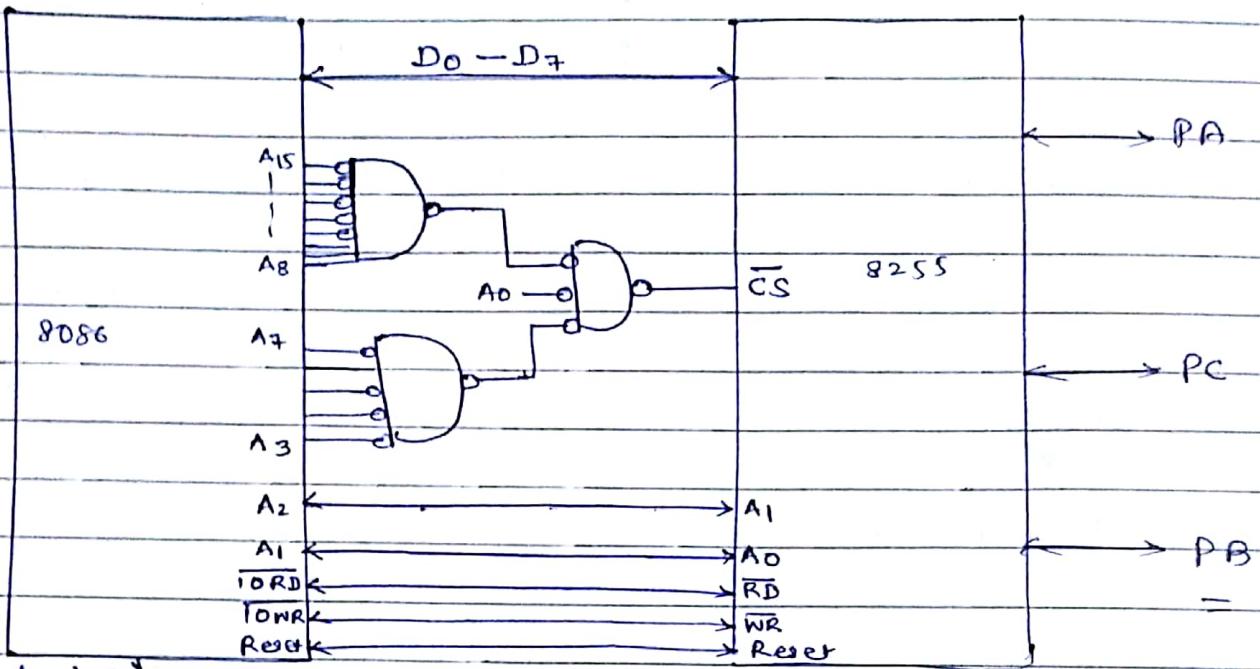
→ let 8 bit address of Port A be 00H

SC	PORT	Hex Address	Binary Address															
			A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>
8255 PPI	Port A	00H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Port B	01H	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
	Port C	02H	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
	CWR	03H	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1

← to generate CS of decoder      → To A, A<sub>0</sub> pin of internal decoder



## 8255 Interfacing with 8086 microprocessor



(LED interfacing)

Ex Interface an 8255 chip with 8086 to work as an I/O port. Initialize Port A as output port, Port B as input port and Port C as O/P port. PA address should be 0740H. Write an ALP to sense switch position SW0-SW7 connected at Port B. The sensed pattern is to be displayed on Port A, to which 8 LED's are connected, while Port C lower displays no. of on switches out of total 8 switches.

→ CWR format (Step 1)

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	Control word
1	0	0	0	0	0	1	0	82H

Step 2 Memory mapping (always take even address)

Port	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Address
Port A	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0740H
Port B	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	1	0742H
Port C	0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	0	0744H
CWR	0	0	0	0	0	0	1	1	1	0	1	0	0	0	1	0	0746H

## programming

MOV DX, 0746H; } initialize CWR with control word  
MOV AL, 82H

OUT DX, AL;

SUB DX, 04; → get Add. of Port B in DX

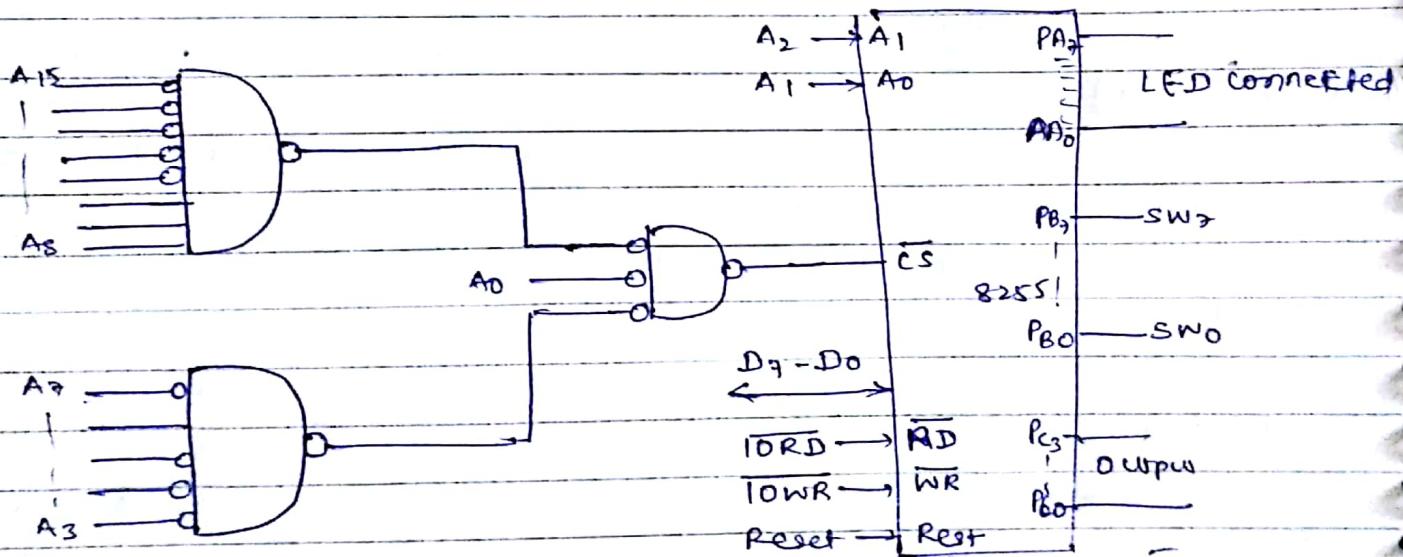
IN AL, DX; → Read Port B for switch

SUB DX, 02; → position into AL and get port A address

OUT DX, AL

HLT ;

## Circuit (interfaced)



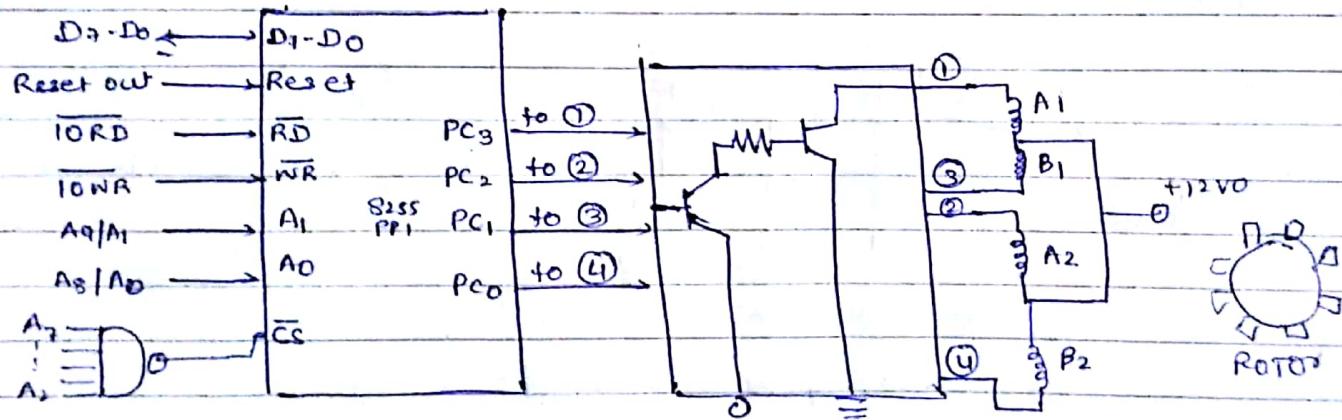
## # Some devices

### \* Stepper Motor Control using Microprocessor

Q Design the hardware and software to rotate stepper motor clockwise at a speed of 200 rpm.

Soln for hardware //

IC	Port	Hex address	Binary Address							
			A15	A14	A13	A12	A11	A10	A9	A8
	P00		A7	A6	A5	A4	A3	A2	A0	A1
8255 PPI	P00+A	20H	0	0	1	0	0	0	0	0
	P00+B	21H	0	0	1	0	0	0	0	1
	P00+C	22H	0	0	1	0	0	0	1	0
	CWR	23H	0	0	1	0	0	0	1	1



for dir<sup>n</sup> control

Steps	sequence of grounding				dir <sup>n</sup>
	PC <sub>3</sub>	PC <sub>2</sub>	PC <sub>1</sub>	PC <sub>0</sub>	
1	0	0	1	1	clock wise
2	1	0	0	1	
3	1	1	0	0	
4	0	1	1	0	
5	0	0	1	1	

Anticlockwise

## Speed control

Speed of stepper motor depend on time delay b/w two steps.

If speed  $N = 200$  rounds per min. So, time req. for one round

$$T = \frac{1}{N} = \frac{1}{200} \text{ min} = 0.3 \text{ sec.}$$

If one round = 200 steps

$$\text{then time of 1 step} = \frac{0.3}{200} = 1.5 \text{ msec}$$

$$t_d = 1.5 \text{ msec}$$

## Software

LXI SP, 0000 H

MVI A, 80 H

OUT 23 H

MVI A, 33 H

} post defined

L1: OUT 22 H

call Delay '1.5 sec'

RRC

JMP L1

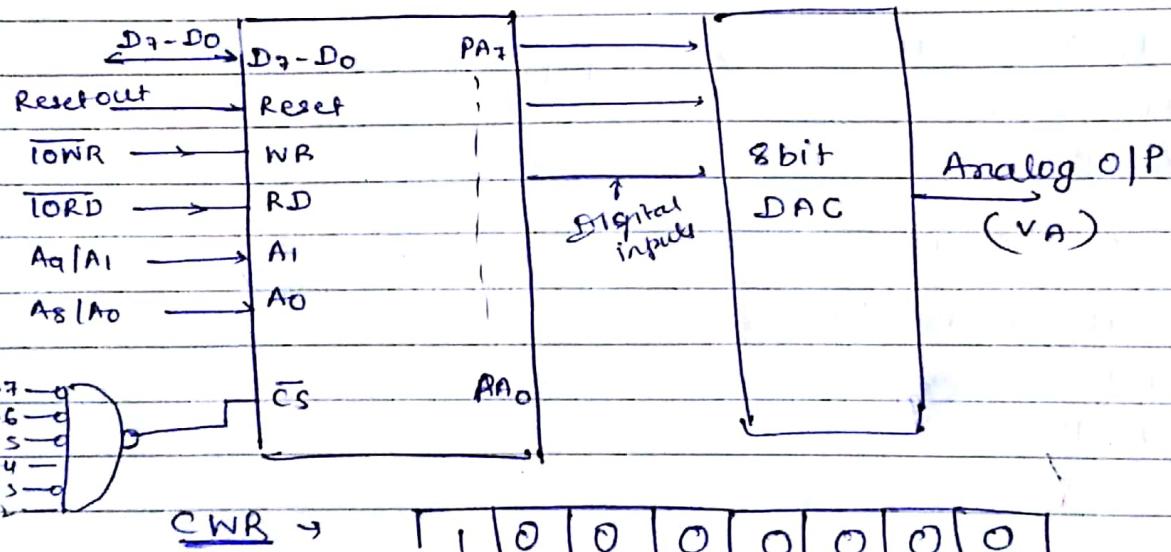
HLR

IMP

## \* Interfacing of Digital to Analog converter (DAC) with 8085 / 8086 (both are same)

→ Address decoding table

IC	Port	Hex Address	Binary Address								
			A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	
		A <sub>7</sub>	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>				
8255 PPI	Port A	14H	0	0	0	1	0	1	0	0	
	Port B	15H	0	0	0	1	0	1	0	1	
	Port C	16H	0	0	0	1	0	1	1	0	
	WR	17H	0	0	0	1	0	1	1	1	

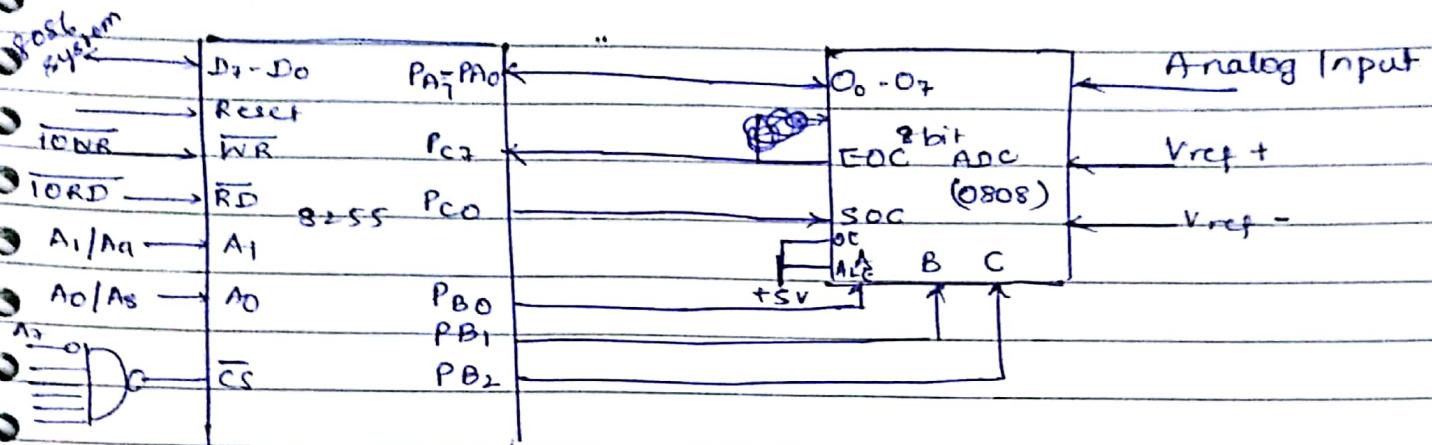


### Software

MVI A , 80H	— Load Accumulator with 80H
OUT 17H	— Transfer to CWR
XRA A	— Clear Accumulator
L1: OUT 14H	— Output port A
INR A	— Increment Accumulator
JMP L1	— Jump to location L1
HLT	

IMP

## \* Interfacing of ADC (Analog to Digital) With 8086



### Combination

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	1	1	0	0	0

= 98H

### Program

```
MOV AL, 98H  
OUT CNR, 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
```

```
WAIT: IN AL, PORTC  
RCR  
JNC WAIT  
IN AL, PORTA  
HLT;
```

## 8253 | 8254 - Programmable Interval timer (PIT)

- 8253 & 8254 are PIT designed for microprocessors to perform timing and counting functions using 3 16-bit registers / counters.
- Each counter has 2 input pins i.e. clock & gate and 1 pin for "OUT" output.

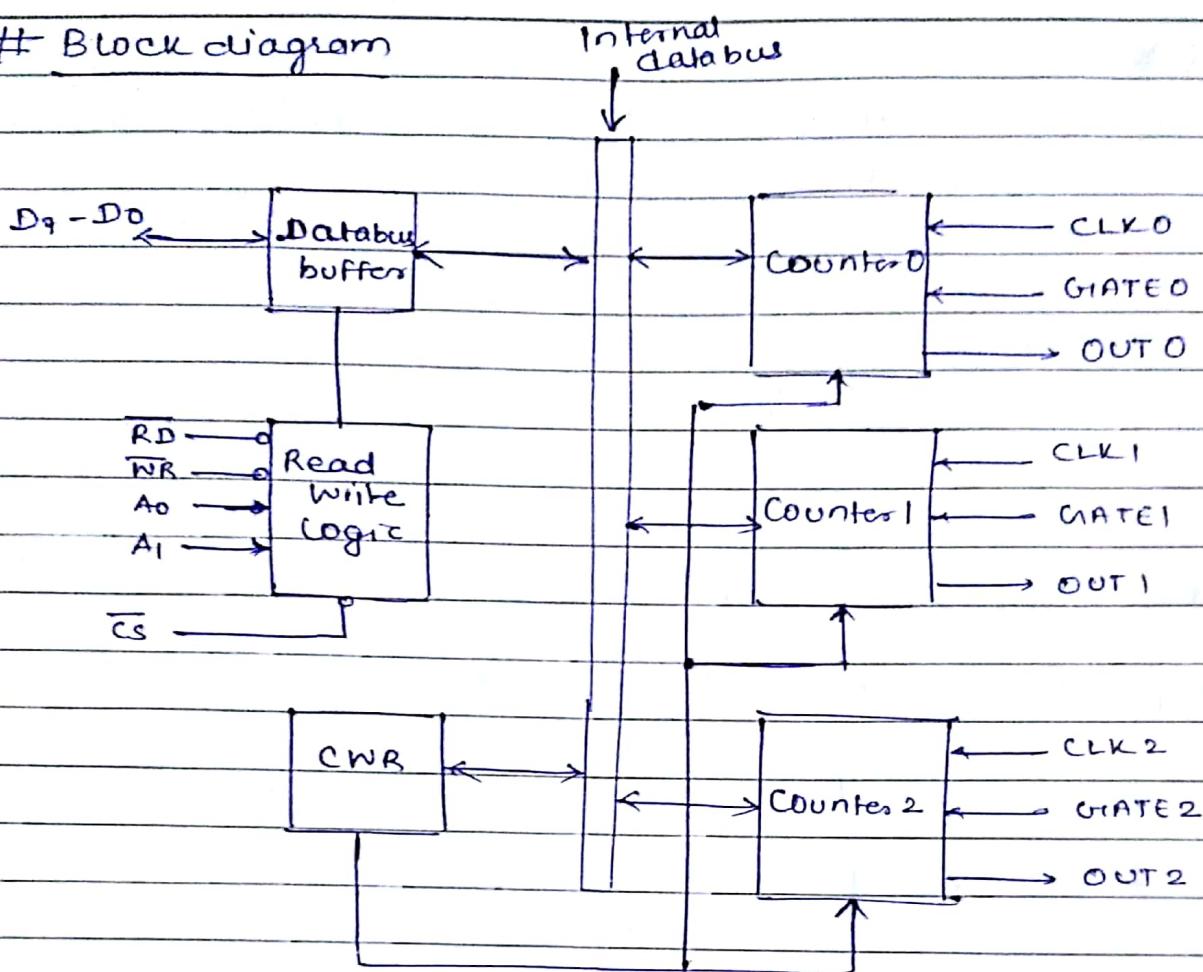
### diff. b/w 8253 & 8254

8253	8254
- operating freq. is 0-2.6MHz	- Operating frequency is 0-10MHz
- uses NMOS tech.	- uses H-MOS tech
- Read Back command is not available	- Read-Back command is available.
- R/W of the same counter can't be interleaved	- R/W of the same counter can be interleaved.

### features

- has 3 independent 16-bit down counters.
- these 3 counters can be programmed for either binary or BCD count.
- compatible with all microprocessors.

## # Block diagram



## pindiagram

<b>D<sub>9</sub></b>	1	24	V <sub>SS</sub> (+5V)
1	2	23	WR
1	3	22	RD
1	4	21	CS
1	5	20	A <sub>1</sub>
1	6	19	A <sub>0</sub>
1	7	18	CLK 2
<b>D<sub>0</sub></b>	8	17	OUT 2
CLK 0	9	16	GATE 2
OUT 0	10	15	CLK 1
GATE 0	11	14	GATE OUT 1
V <sub>SS</sub> (0V)	12	13	GATE GATE 1

## Data bus buffer

→ tri-state 8 bit bidirectional buffer which is used to interface 8253 to system database.

### functions

- loading the count registers
- reading the count registers

## Read | write logic

includes 5 signals,  $\overline{RD}$ ,  $\overline{WR}$ ,  $\overline{CS}$ ,  $A_0$ ,  $A_1$

In the peripheral I/O mode,  $\overline{RD}$  and  $\overline{WR}$  are connected with  $\overline{IOR}$  and  $\overline{IOW}$  respectively.

In the memory mapped I/O, these are connected to  $\overline{MEMR}$  &  $\overline{MEMW}$

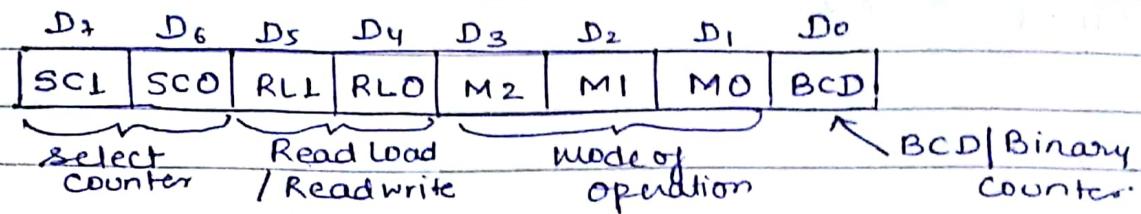
$A_1$	$A_0$	<u>Selected Counter</u>
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control word register

## Control word register

accessed when lines  $A_0$  &  $A_1$  are at logic 1. used to write a command word, which specifies the counter to be used, its mode and either a read/write operation.

$A_1$	$A_0$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	<u>Result</u>
0	0	1	0	0	Write Counter 0
0	1	1	0	0	" " 1
1	0	1	0	0	" " 2
1	1	1	0	0	Write CWR
0	0	0	1	0	Read Counter 0
0	1	0	1	0	" " 1
1	0	0	1	0	" " 2
1	1	0	1	0	No operation
X	X	1	1	1	No operat

## # Control Word Register of 8253/54



SCL    SCO    counter initialized

0    0    Counter 0

0    1    Counter 1

1    0    Counter 2

1    1    Read Back command for 8254 | Invalid for 8253

~~RLI~~    RLO    [LSB|MSB of Counter register selected]

0    0    Read on fly operation

0    1    Read & Load only LSB of Counter Register

1    0    Read & Load only MSB of Counter register

1    1    Read & Load LSB first and MSB next

~~M<sub>2</sub>~~    M<sub>1</sub>    M<sub>0</sub>    Mode defined

0    0    0    Mode 0

0    0    1    Mode 1

X    1    0    Mode 2

X    1    1    Mode 3

1    0    0    Mode 4

1    0    1    Mode 5

D<sub>0</sub> = 1 → BCD Counter (4 Decades)

D<sub>0</sub> = 0 → Binary Counter (16 bits)

## # Modes of operations

### ① Mode 0 (Interrupt on Terminal Count)

- used to generate an interrupt to the microprocessor after a certain interval.
- Initially the OUT pin is low after mode is set. Output remains low after the count value is loaded into the counters.
- If gate input = 1, then at each negative edge CLK input the counter decrements by one and after terminal count 0, OUT pin becomes high.

### ② Mode 1 :- (Programmable One shot)

- used as a monostable multivibrator.
- gate input is used as a trigger input
- the output remains high until the count is loaded and trigger is applied.

### ③ Mode 2 (Rate generator)

- OUT pin becomes high (logic 1).
- whenever count becomes zero, another low pulse is generated at the output and the counter will be reloaded.

### ④ Mode 3 (Square wave generator)

- similar to mode 2 except that output remains low for half of the timer period and high for the other half of the period.

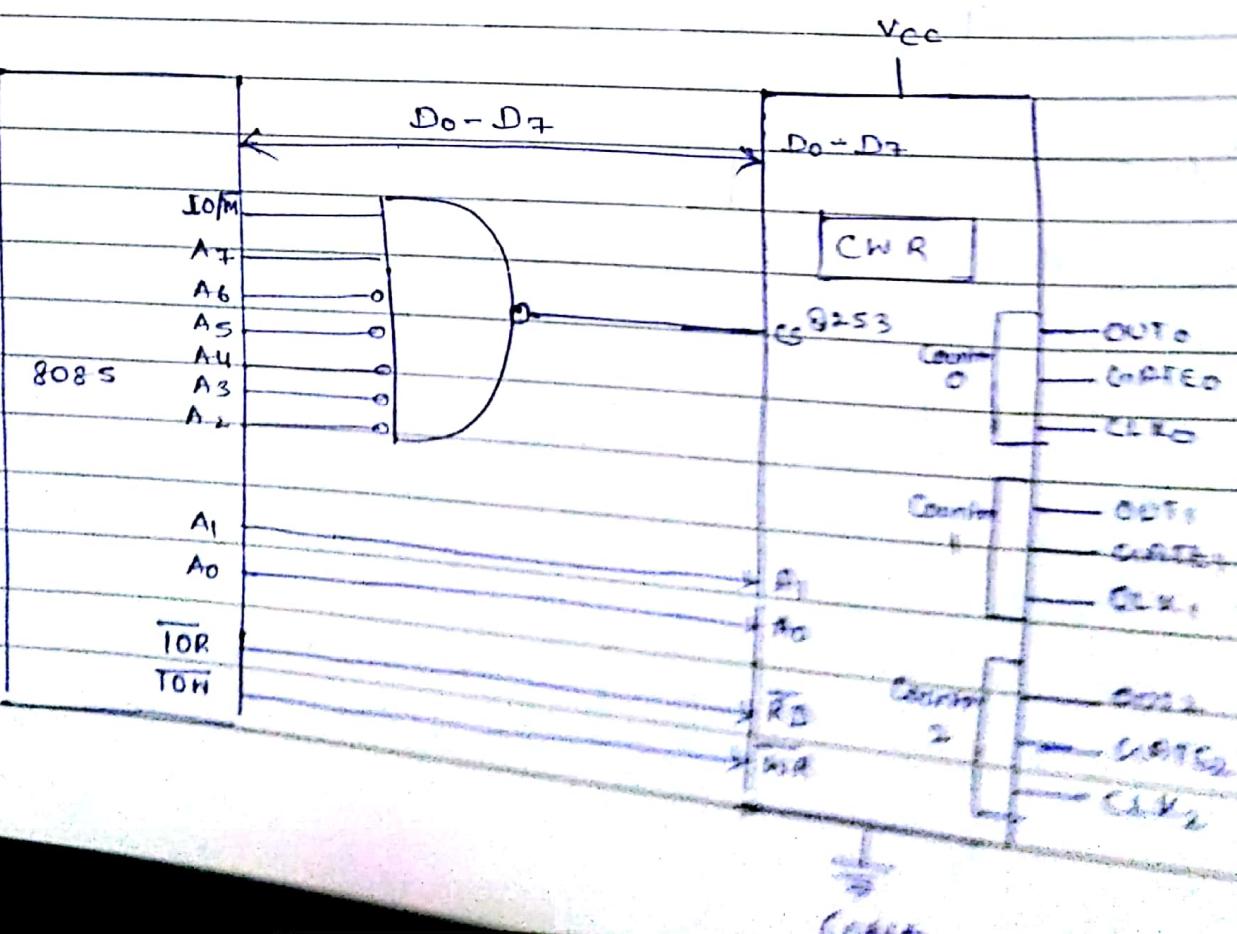
## ⑤ Mode 4 (SW triggered mode)

- output will remain high until the timer has counted to zero.
- count is latched when the INT signal goes low.
- on terminal count, the output goes low for one clock cycle and then goes high. This low pulse can be used as a strobe.

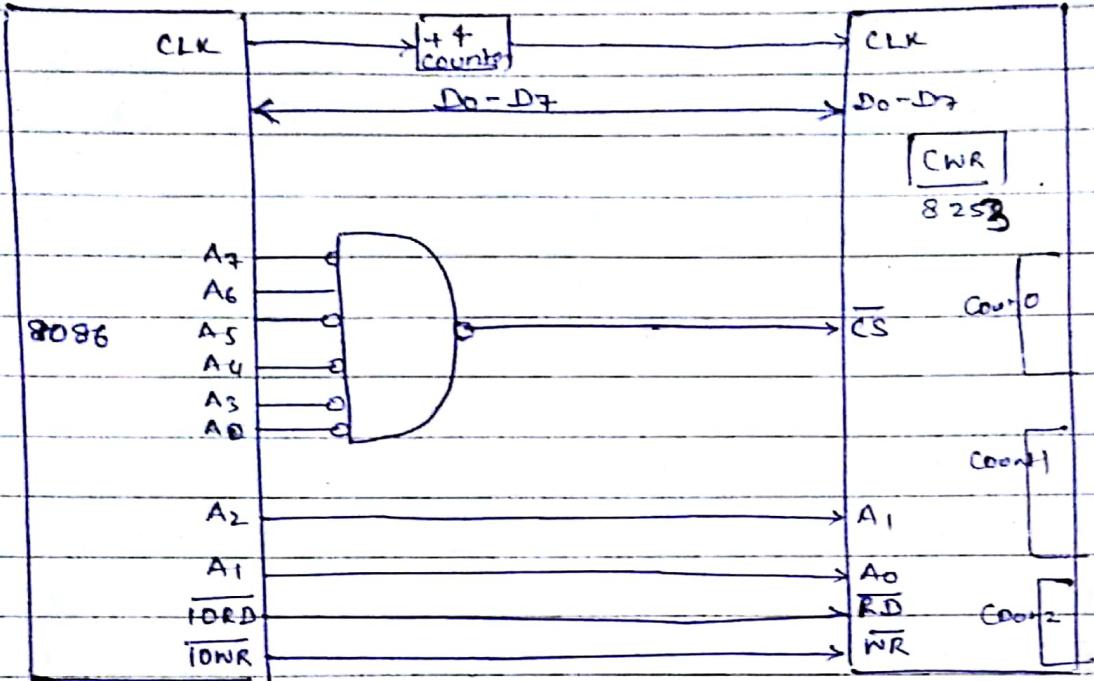
## ⑥ Mode 5 (Hardware triggered)

- mode generates a strobe in response to an externally generated signal.
- similar to mode 4 except that counting is initiated by a signal at the gate input, which means it is hardware triggered instead of software triggered.
- After it is initialized, output goes high.

## # Interfacing of 8253 with 8085 microprocessor

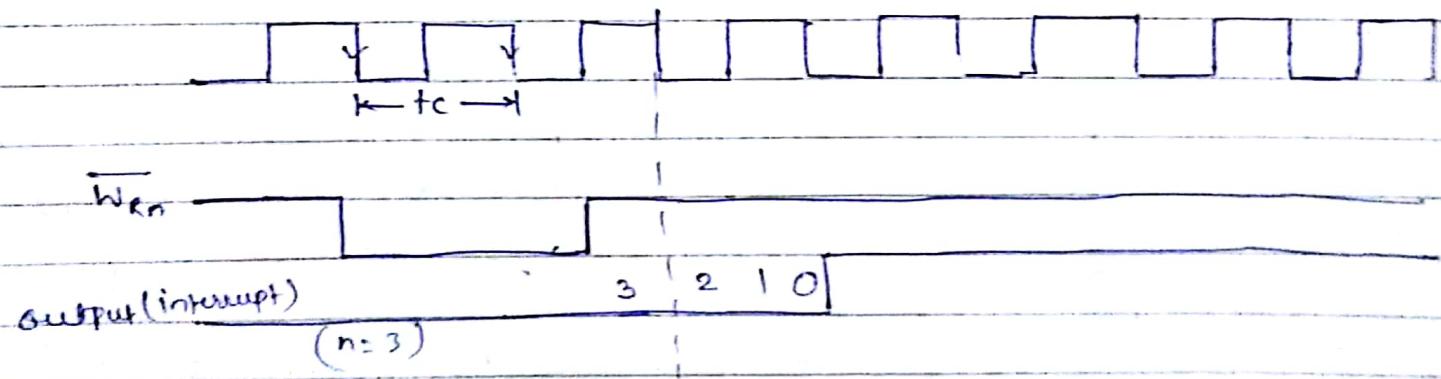


## # 8253 Interfacing with 8086 microprocessor

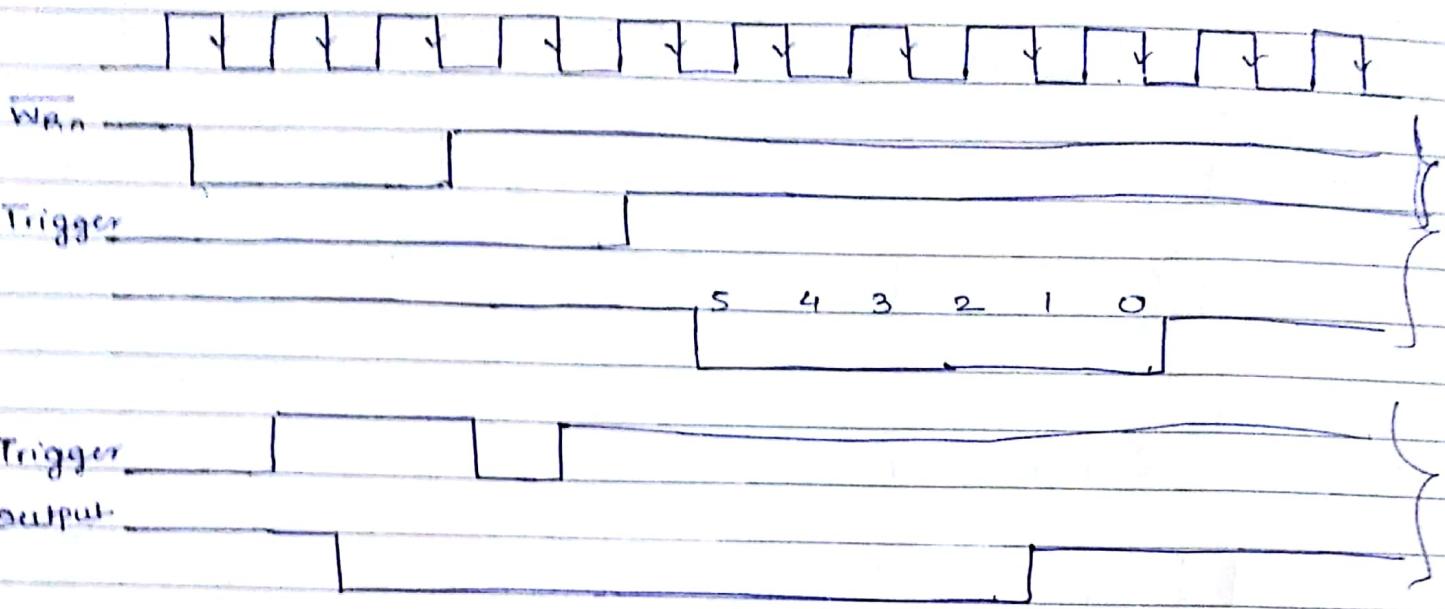


\* Waveforms of mode of operation of 8253/8254

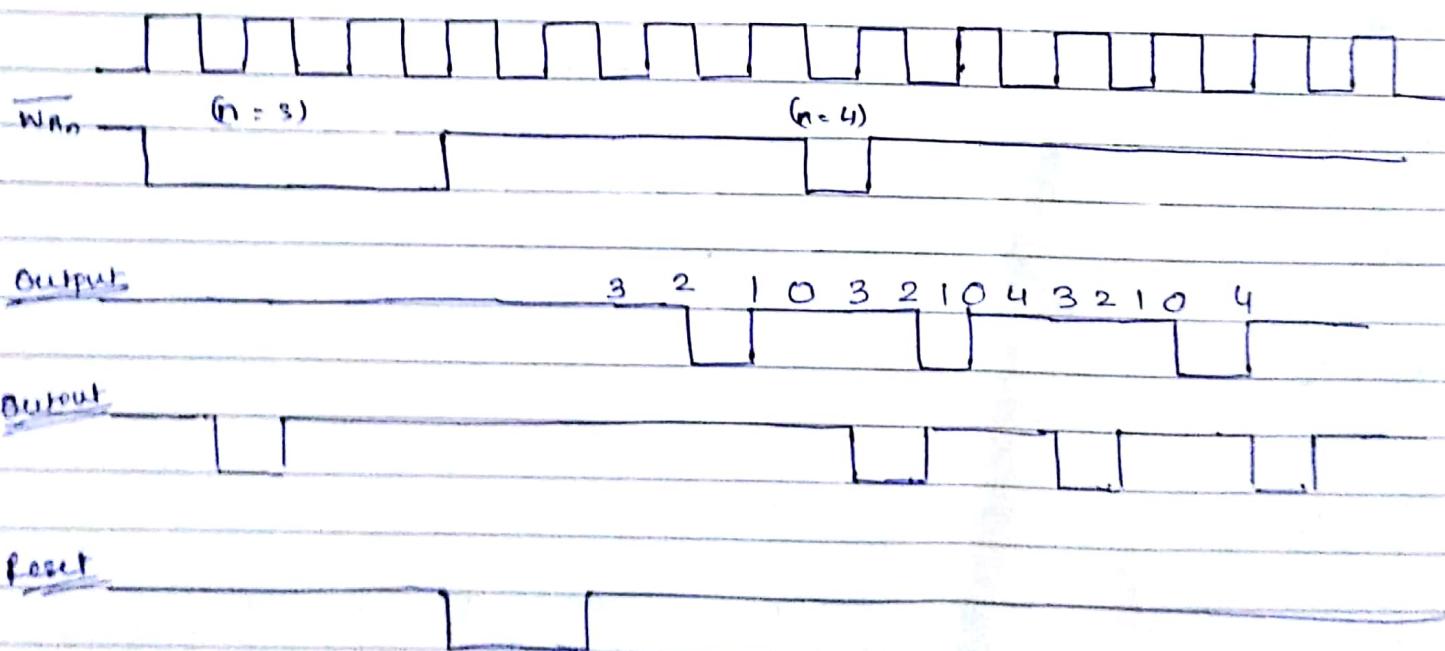
① Mode 0 (Interrupt on terminal count)



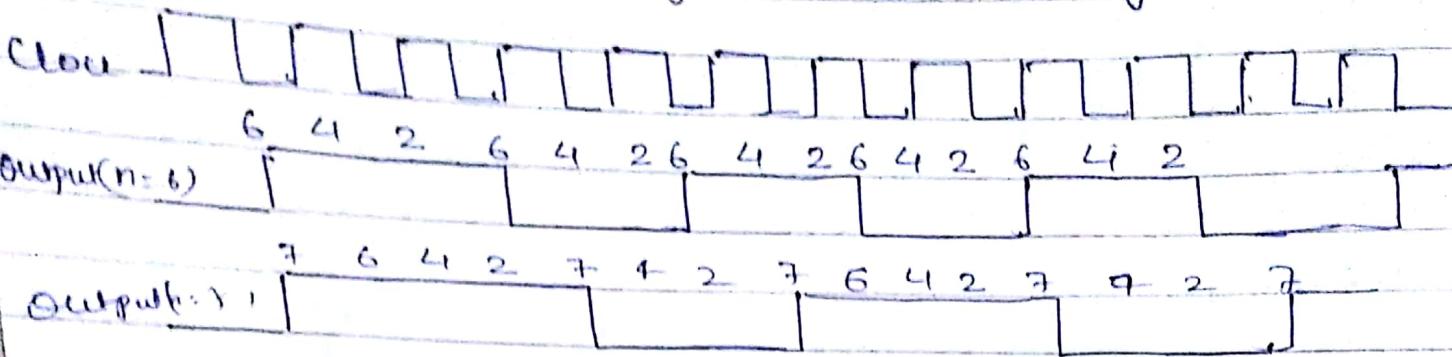
② mode 1 (Programmable Retriggable monostat) / one shot.



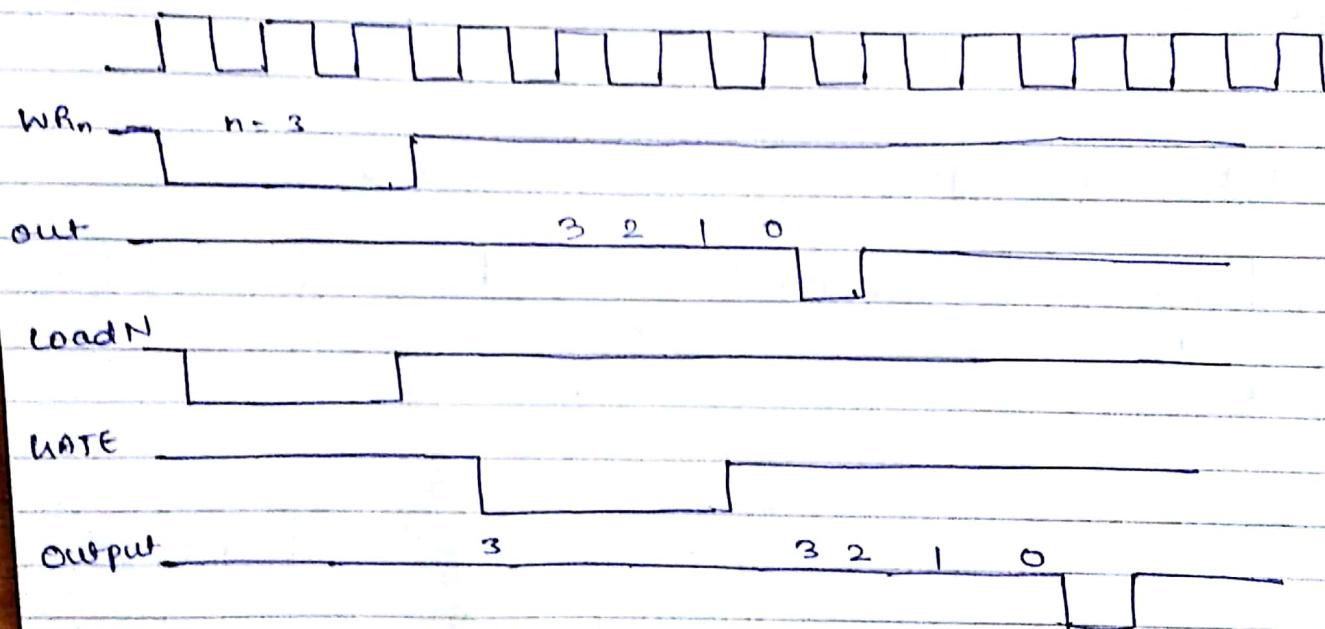
③ mode 2 (Rate generator)



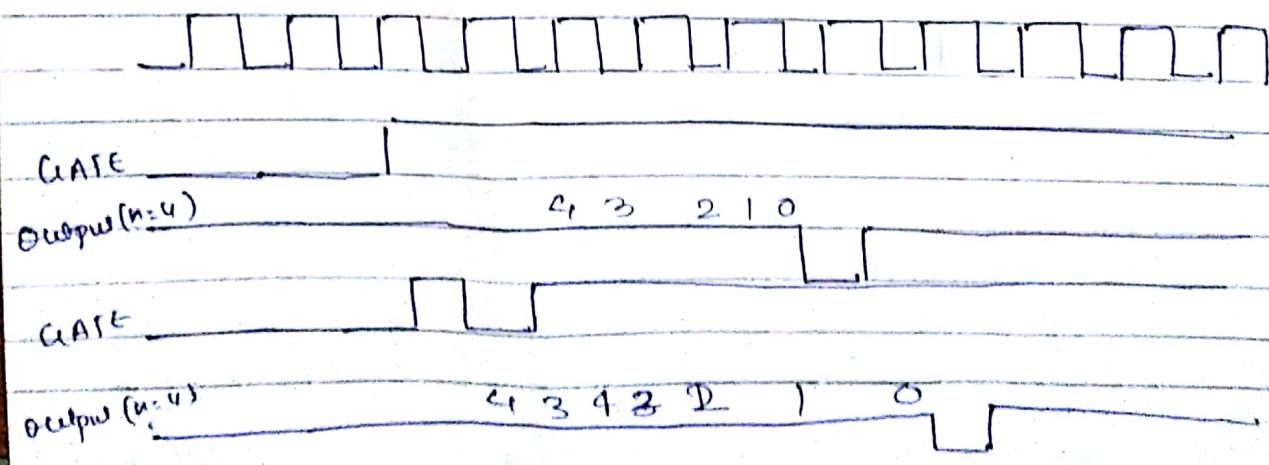
④ mode 3 ( 8 square wave generator or divide by N counter)



⑤ Mode 4 (Software triggered)



⑥ mode 5 (Hardware triggered)



## #8259 (programmable Interrupt Controller)

- PIC works as an overall manager in an interrupt driven data transfer scheme.
- It accepts requests from peripheral equipments, determine which of the incoming requests is of the highest priority. The highest priority interrupt is then executed
- It is 28 pin IC package which uses NMOS technology

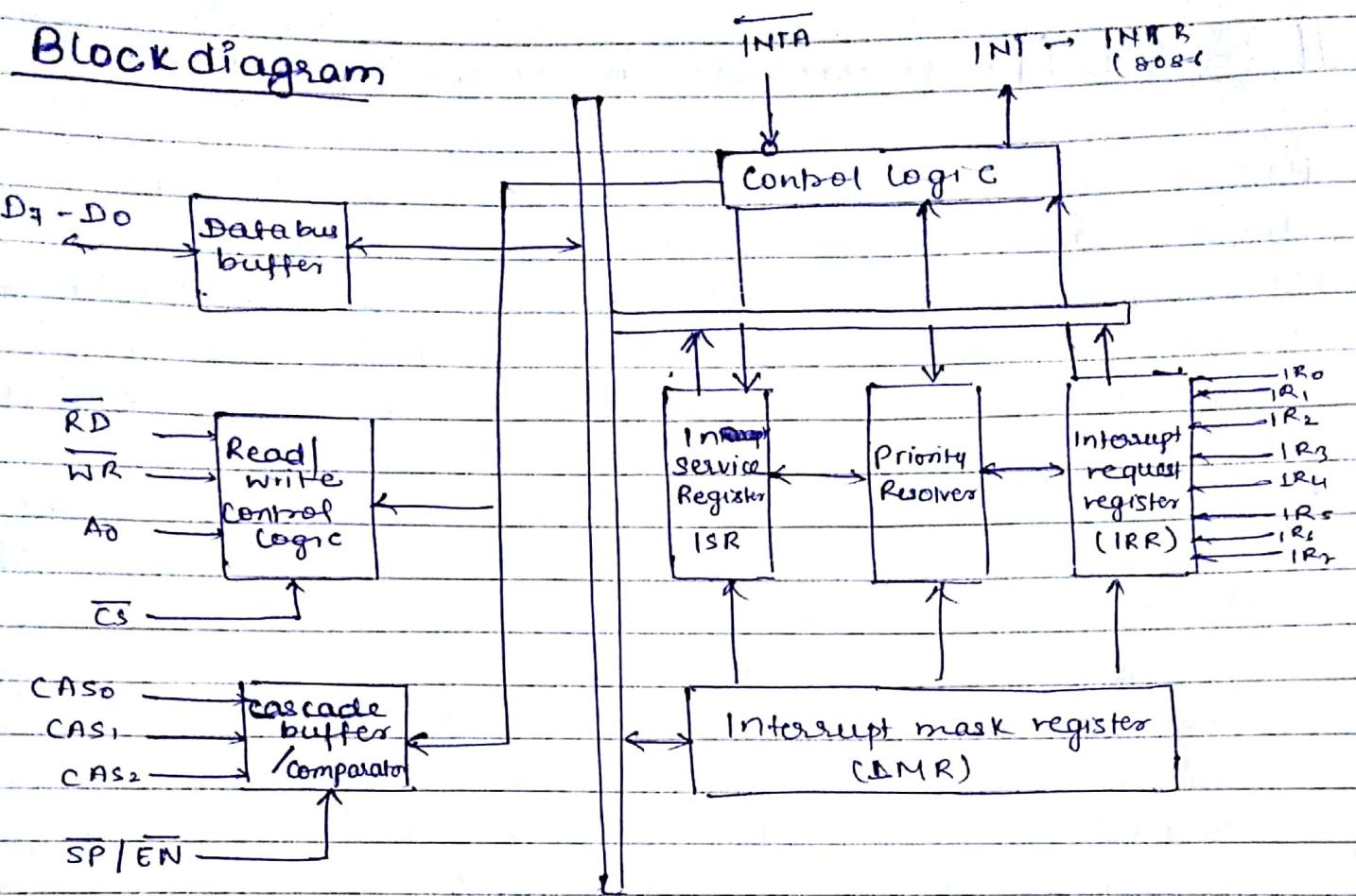
### features

- manages 8 level of interrupt
- Compatible with 8055, 8088 and 8086 mp.
- can be configured in master slave or cascade mode to handle upto 64 interrupt.
- provides on chip resolving ckt.
- can be implemented in polled as well as vector mode.
- provides starting address of interrupt service routine
- can be used in large systems in buffered mode.

~~REB000~~ -

## Interrupt & Control Logic section

### Block diagram



### Components

#### ① Interrupt and control logic section

##### (i) Interrupt Request

8259 has 8 diff. interrupt request levels, i.e IR<sub>0</sub> to IR<sub>7</sub>.  
eight diff. interrupting devices can be connected to these eight IR levels.

These 8 IR levels defined ~~as~~ either as <sup>edge</sup> triggered or level triggered.

##### (ii) Interrupt Request Register (IRR)

- 8 bit register which contains 8 flip flops D<sub>0</sub>-D<sub>7</sub> corresponding to 8 IR levels IR<sub>0</sub> to IR<sub>7</sub>.

When any device gives interrupt signal on IR level, then corresponding bit of IRR is set.

### (iii) Interrupt Mask Register (IMR)

- 8 bit registers which contains 8 flip-flop corresponding to 8 IR levels  $IR_0$  to  $IR_7$ .
- If any IR level is to be disable (mask) then corresponding bit of IMR should be set.

Ex to enable/disable  $IR_0$  the programmer has to make  $Do = 0/1$  respectively.

In case of default all eight IR levels  $IR_0$  to  $IR_7$  is enable.

### (iv) Priority Resolver

- Stores the priorities of eight different IR levels  $IR_0$  to  $IR_7$ .  
In case of default,  $IR_0$  is assigned highest priority and  $IR_7$  is lowest priority.

### (v) INR (Interrupt Service Register)

- 8 bit register contains 8 flip flop corresponding to Eight IR levels  $IR_0$  to  $IR_7$
- tells which interrupt is executing.
- set the bit of interrupt which is executing and reset other bits.

### (vi) Control Logic

- function is to accept output commands from CPU.
- contains the Initialization Command Word Register and open command register which store various control formats for device open.
- has two pins INT (Interrupt) and INTA (Interrupt acknowledge). INT is a output pin connected to INTR pin of 8085/8086.

INTA is a input pin connected to INTA pin of microprocessor.

## ② Database buffer

- 8 bit bidirectional buffer is used to ~~store~~ interface 8259 to system database.

## ③ Read/write control logic

- has initialization CWR and open" command register.

(i)  $\overline{CS}$  (chipselect) - low on this input enables 8259

(ii)  $\overline{WR}$  - write control words

(iii)  $\overline{RD}$  - read control words

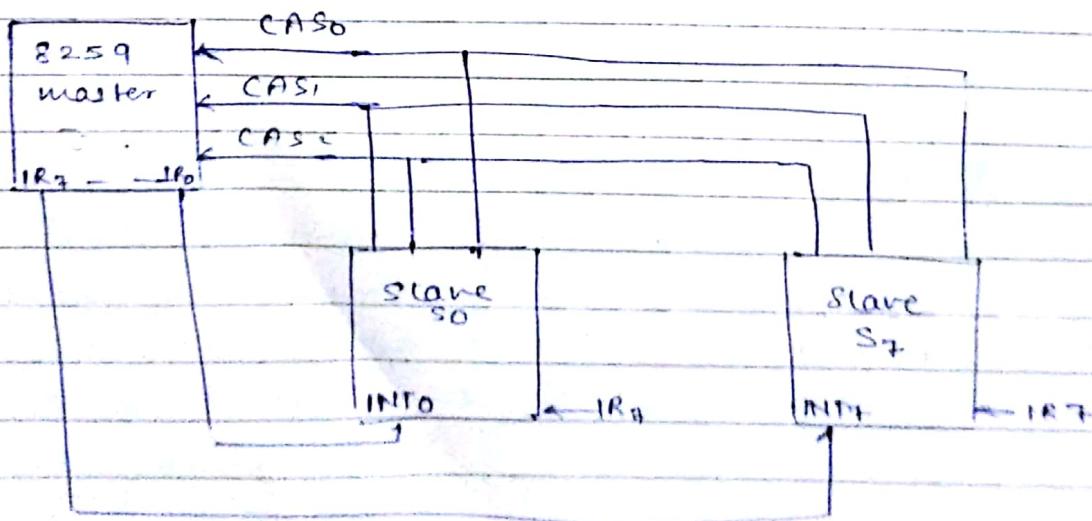
## ④ Cascade buffer comparator

has three cascade signals.

CAS<sub>0</sub>, CAS<sub>1</sub> and CAS<sub>2</sub> and  $\overline{SP} | EN$  (Slave program enable buffer)

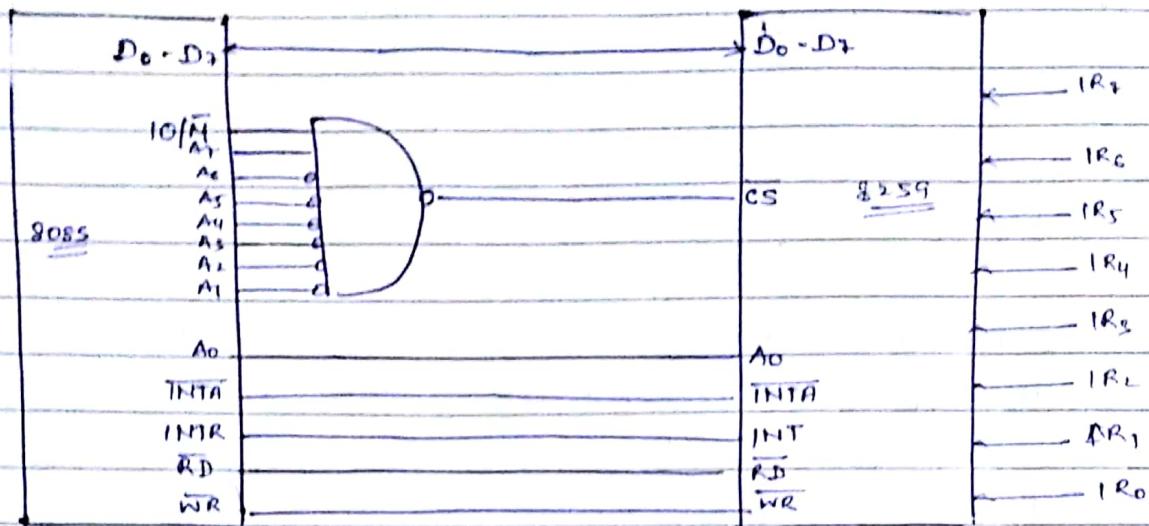
- used to send signals from master output to slaves when multiple devices are cascaded.

Block diagram

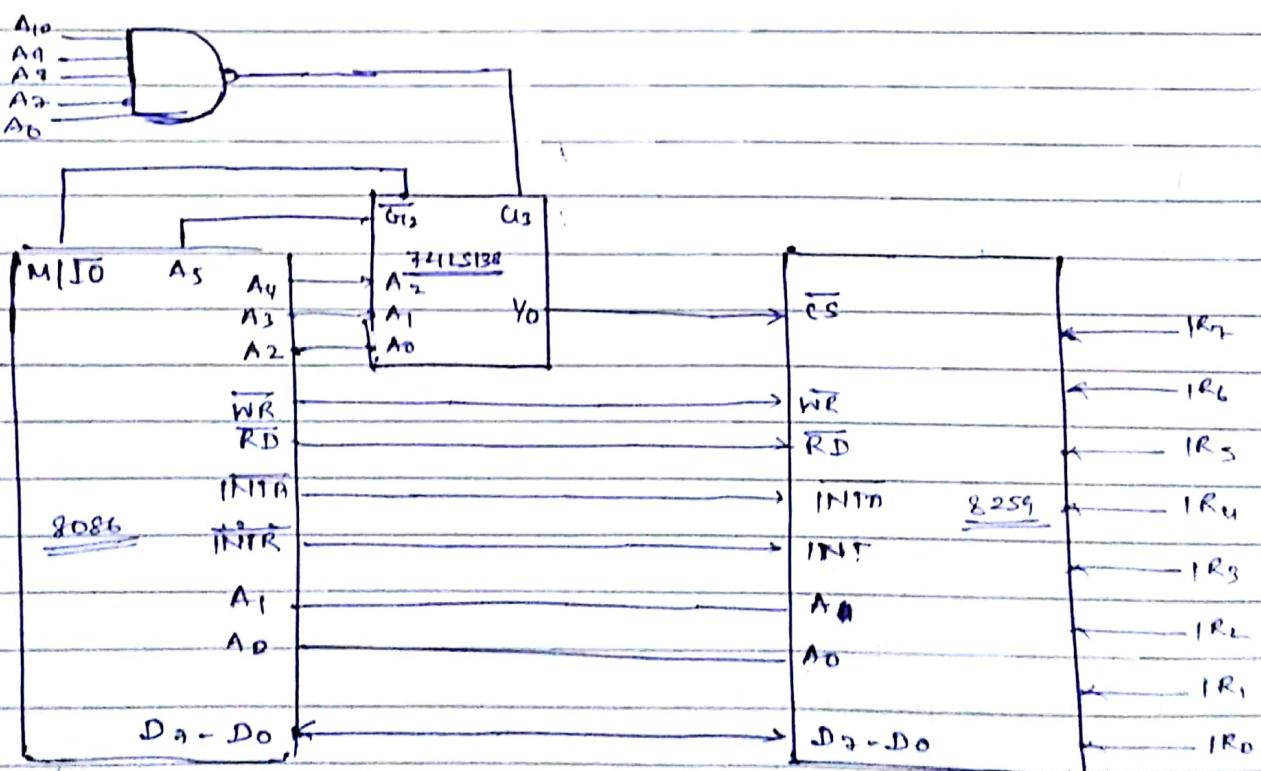


If  $\overline{SP} = 1 \rightarrow$  as master  
 $\overline{EN} = 0 \rightarrow$  as slave

## # Interfacing of 8259 PIC with 8085



## # Interfacing of 8259 with 8086



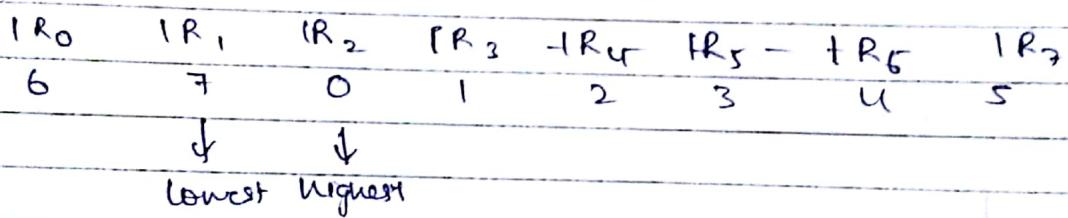
## # Modes of defining priorities of IR level

### ① fully nested mode | Default mod:



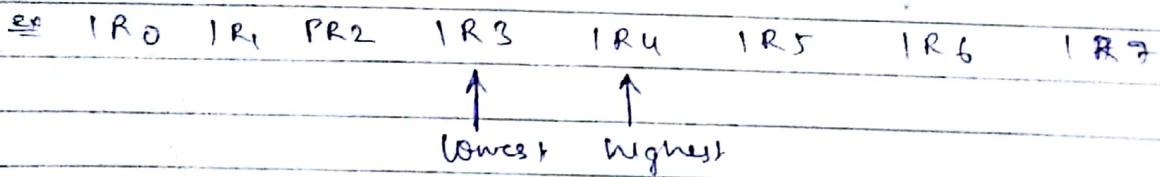
### ② Specific rotation mode

- any one IR level can be ~~can't~~ specified for assigning the lowest priority hence priorities of other IR levels gets fixed in cyclic order.



### ③ Automatic rotation mode

- interrupt which is executed is having the lowest priority.



### ④ Special fully nested mode

assigned priority to the slave.

By default, Slave 0  $\rightarrow$  highest

Slave 7  $\rightarrow$  lowest.

## # Control command words of 8259 programming

two types of control words

- (a) ICW's (Initialization control word)
- (b) OCW's (Operational control word)

### (a) ICW's (Initialization control word)

8259 can be used with four ICW's, ICW<sub>1</sub>, ICW<sub>2</sub>, ICW<sub>3</sub> & ICW<sub>4</sub>.

Each 8259 in the system must first be initialized by loading a set of these ICWs in a sequence.

Once initialized, 8259 can be set up to operate in various modes by using 8 different OCW8.

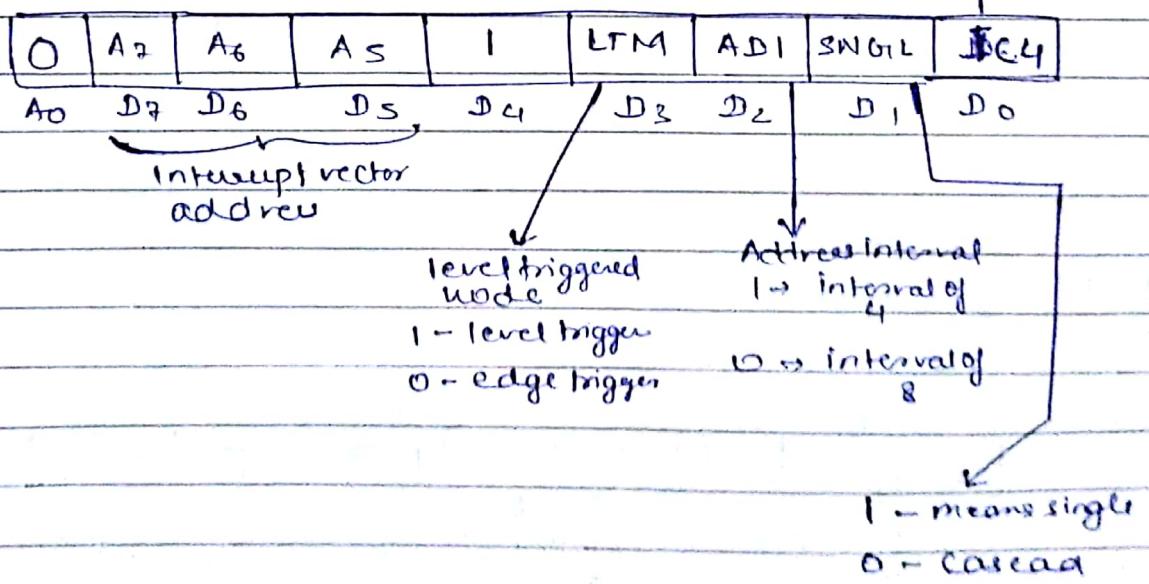
ICW<sub>1</sub> & ICW<sub>2</sub> are compulsory. Other two are optional.  
ICW<sub>3</sub> used only in cascade mode.

ICW<sub>4</sub> used for defining special oper<sup>n</sup>.

#### ① ICW<sub>1</sub>

- write command issued to 8259 with A<sub>0</sub> = 0 and D<sub>4</sub> = 1.
- ICW<sub>1</sub> is transferred to port 0.

ICF=1, ICW<sub>4</sub> <sup>is needed</sup>  
ICF=0, not needed



② ICW<sub>2</sub>

Write command following ICW<sub>1</sub> with A<sub>0</sub>=1 is interpreted as ICW<sub>2</sub> is transferred to port 1.

<u>A<sub>0</sub></u>	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub>	A <sub>8</sub>	0001
A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	

③ ICW<sub>3</sub> — transferred to port 1 of master as well as slave.  
— operates only in cascade mode.

(a) ICW<sub>3</sub> for master

— indicates slave is connected or not.

<u>I</u>	SL7	SL6	SL5	SL4	SL3	SL2	SL1	SL0
A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>

(b) ICW<sub>3</sub> for slave

ID<sub>2</sub>, ID<sub>1</sub>, ID<sub>0</sub> is 3 bit id of slave 8259.  
these Id corresponds to three bit address of I<sub>R</sub> level of master 8259.

<u>I</u>	0	0	0	0	0	ID <sub>2</sub>	ID <sub>1</sub>	ID <sub>0</sub>
A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>

④ ICW<sub>4</sub> — transferred to port 1 of 8259

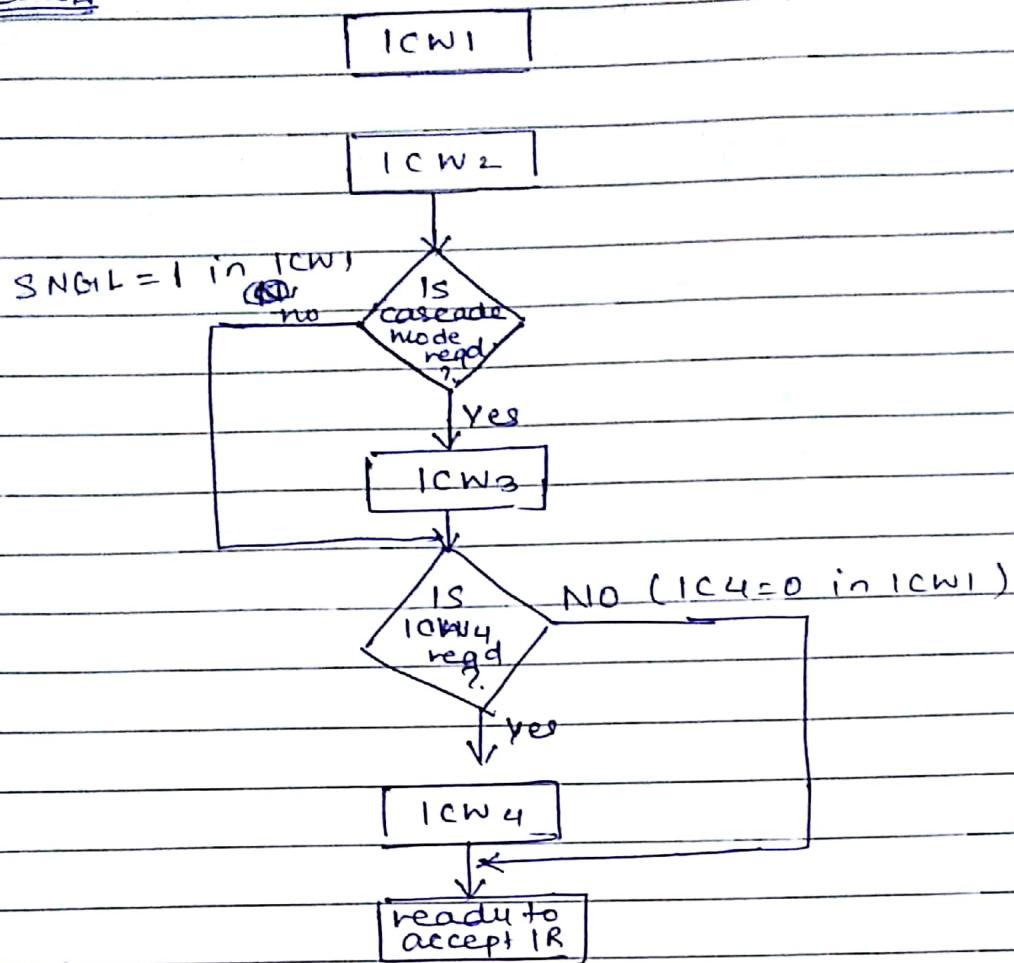
A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	
1	0	0	0	0	GFNM	BUF	N/S	AE03	01

↓  
special  
fully nested  
mode

↓  
Buffered  
System

↓  
Master-Slave

### flowchart



### (b) OCW's (Operational Control Word)

① OCW<sub>1</sub> - used for transfer 8 bit data into IMR. If IR<sub>7</sub> is to be masked then M<sub>7</sub> = 1 and if IR<sub>7</sub> is to be kept enabled, M<sub>7</sub> is transferred as 0.

- port 1

M <sub>7</sub>	M <sub>6</sub>	M <sub>5</sub>	M <sub>4</sub>	M <sub>3</sub>	M <sub>2</sub>	M <sub>1</sub>	M <sub>0</sub>
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

② OCW<sub>2</sub> - transferred to port 0

R	SL	EOI	O	O	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>
---	----	-----	---	---	----------------	----------------	----------------

used for defining different priority modes and EOI commands.

③ OCW<sub>3</sub> - transferred to port O but for OCW<sub>3</sub>, D<sub>4</sub>D<sub>3</sub> = 0.

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
O	ESMM	SMM	O	I	P	PR	RIS

↓  
Special  
mask  
mode

ESMM SMM

O	X
I	I
I	O

8259 is set in SMM

8259 is reset in SMM.

## 8251 USART (Universal Synchronous and Asynchronous Receiver and Transmitter)

### Synchronous data transmission

- same clock pulse is applied to transmitter and receiver simultaneously
- only hardware required to implement this
- synchronous pulses are required
- used for high speed transmission

### Asynchronous data transmission

- different clock pulse are applied to transmitter & receiver separately
- both hardware & software required
- not required but uses start & stop bits
- low speed transmission

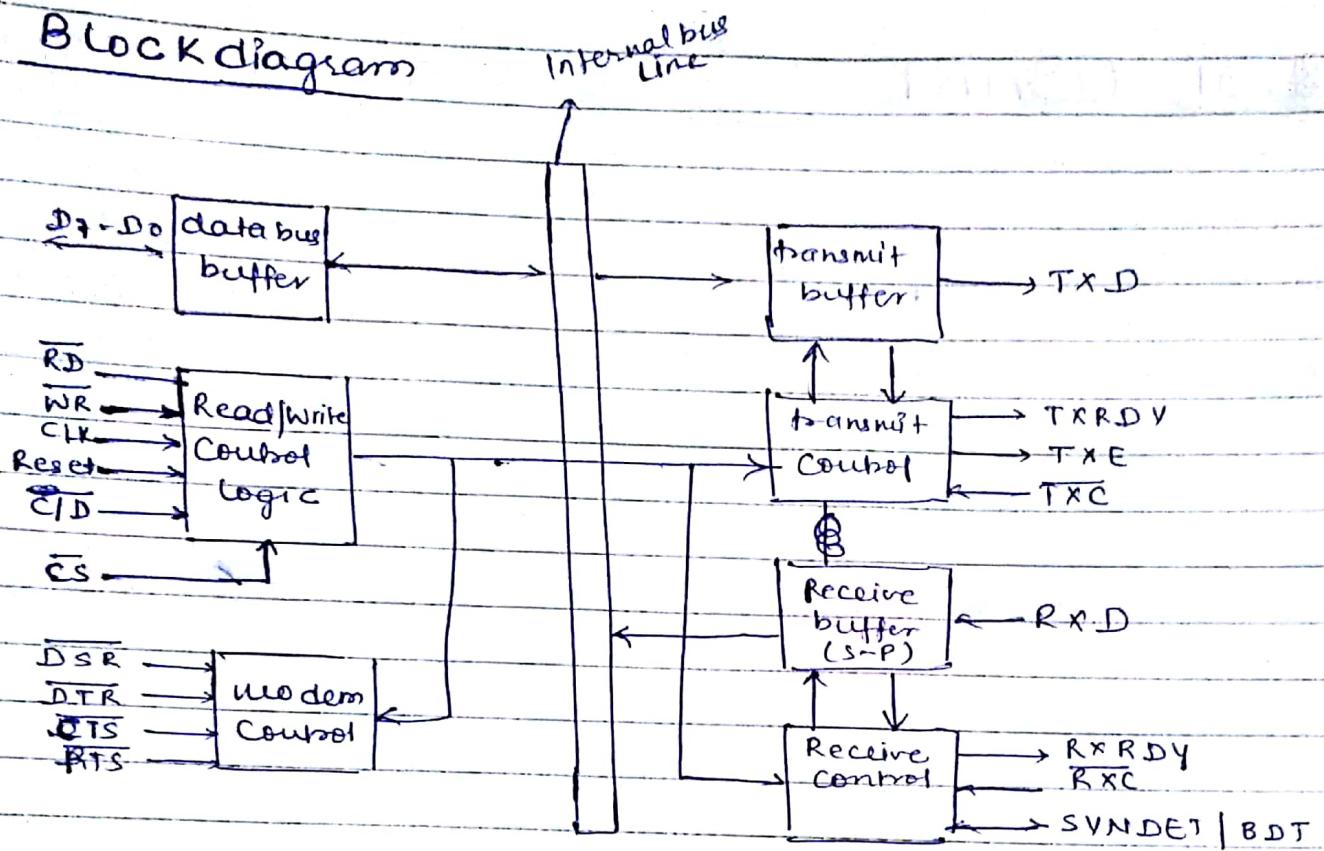
8251 USART can transmit as well as receive serial data in synchronous mode as well as asynchronous mode.

The data transfer b/w microprocessor and 8251 will be performed in parallel.

It is designed for the data communication with 8088, 8085, 8086 microprocessors.

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

## Block diagram



### ① Data bus buffer

- bidirectional buffer used to interface the 8251 to the system data bus.

### ② R/W control logic

Reset - high on this input forces 8251 into an idle mode

CK - used to generate internal device timing and is normally connected to phase 2 (TTL) output of clock generator.

WR - writing data

RD → reading data

C/D → (Control/ data)

informs 8251 that word on databus is either a data character, control word or status info.

1 = control word

0 = data

③ Modem Control :- has a set of control inputs & outputs that can be used to simplify the interface to almost any modem.

(i) DSR - (data set ready) → test modem condn  
(input signal) such as Data set ready

(ii) DTR - (data terminal ready) → output signal.  
used for modem control

(iii) CTS - (clear to send) (input signal) enables 8251  
to transmit serial data

(iv) RTS - (request to send) (output signal)

C/D	RD	WR	CS	
0	0	0	0	data port to data bus
0	1	1	0	data bus to data port
1	0	0	0	status word to data bus
1	1	1	0	data bus to control register
x	1	1	0	data bus is tri-state
x	x	x	1	data bus is tri-state

#### ④ Transmitter section

##### (a) Transmitter buffer

accepts parallel data from data bus buffer, converts it to a serial bit stream, inserts the appropriate characters and outputs a composite serial stream of data on TxD output pin.

The transmitter will begin transmission upon being enabled if  $\overline{CTS} = 0$ .

## (b) Transmission Control

manages all activities associated with transmission of serial data.

It accepts and issues signal both externally and internally to accomplish this function.

### (i) TXRDY (Transmitter Ready)

- indicates transmitter is ready to accept a data character.
- It can be used as an interrupt to the system since it is marked by TX enable.

### (ii) TXE (Transmitter empty)

- when no character to send, the transmitter empty will go high.
- used to indicate the end of transmission mode

### (iii) TxC (<sup>transmitter clock</sup> ~~transmitter~~)

- control the rate at which character is to be transmitted.

In Synchronous mode, the Baud rate is equal to TxC frequency.

In Asynchronous mode, the baud rate is a fraction of actual TxC frequency.

## ⑤ Receiver Section

### (a) Receiver buffer

accepts serial data, convert the serial input to parallel format checker for bits or char that are unique to the communication technique.

Serial data is input to RxD pin.

### (b) Receiver Control

manages all receiver-related activities

#### (i) RXRDY (Receiver Ready)

- indicates that character is ready to be input to CPU.
- can be connected to interrupt structure of the CPU

#### (ii) RXC (Receiver Clock)

- controls the rate at which character is to be received.

In sync. mod, Baud rate is equal to actual frequency of RXC

In Async. mode, Baud rate is a fraction of actual RXC frequency

#### (iii) SYNDET (SYNC detect / BRKDET Break detect)

## Control Word Register

8251 has 16 bit CWR, out of which 8 LSB's are called mode word register and 8 MSB's are called command word register.

### ① mode word for Asynchronous mode

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
S <sub>2</sub>	S <sub>1</sub>	EP	PEN	L <sub>2</sub>	L <sub>1</sub>	B <sub>2</sub>	B <sub>1</sub>

B<sub>2</sub> & B<sub>1</sub> are band rate factor

B <sub>2</sub>	B <sub>1</sub>	CLK frequency
0	0	for sync. mode only
0	1	CLK frequency = Baud rate
1	0	CLK freq = 16 Baud rate
1	1	CLK freq = 64 Baud rate

L<sub>2</sub> & L<sub>1</sub> defines length of character

L <sub>2</sub>	L <sub>1</sub>	length
0	0	five bits
0	1	six bits
1	0	seven bits
1	1	eight bits

EP & PEN defines parity of data to be received or transmitted

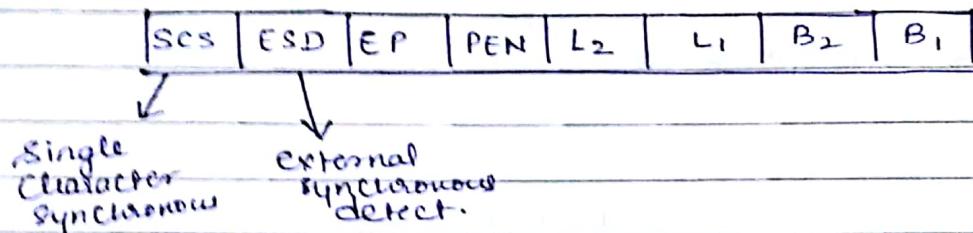
EP (Even parity)	PEN (Parity enable)	Result
0	0	not defined
0	1	Odd parity received
1	1	Even parity received

$s_2$  &  $s_1$  bits are used to define no. of stop bits

$s_2$	$s_1$	no. of stop bits
0	0	Invalid
0	1	one stop bit
1	0	one & half stop bit
1	1	two stop bit

## ② Mode word for Synchronous mode

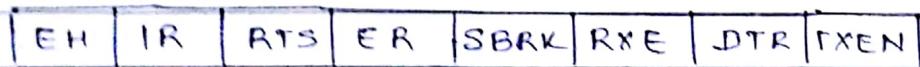
Six LSB's are same as in Async. mode.



If only one/two sync. characters are to be transferred then SCS should be 1/0 respectively.

If sync. char are to be detected internally/externally, then ESD should be 1/0 respectively.

## ③ Command word for 8251 USART



$TXEN = 1/0$ , then transmitter is enable/disable

$DTR = 1/0$ , then 8251 will make DTR pin Active/inactive

$RXE = 1/0$ , receiver is enable/disable

$SBRK$  (send break character) = 1/0, 8251 will make TXD pin low i.e. char. is broken in async. mode.

$ER$  (error reset) = 1, reset all three error flags i.e framing error, parity error, overrun error

$IR$  (interrupt reset) = 1,

$EH$  (Enter hunt mode) = 1 - will start searching for Sync. character

#### ④ Status Word Register (SWR)

DSR	SYNDET	FE	OE	PE	TXE	RXDY	TXRDY
-----	--------	----	----	----	-----	------	-------

### # 8279 (Programmable Keyboard and Display Controller)

8279 is a general purpose programmable keyboard and display I/O interface device.

It is a 40 pin TTL IC which is used to interface push button matrix keyboard and multiplexed seven segment display with 8085.

It has to perform two opns i.e. Keyboard scanning and displaying characters.

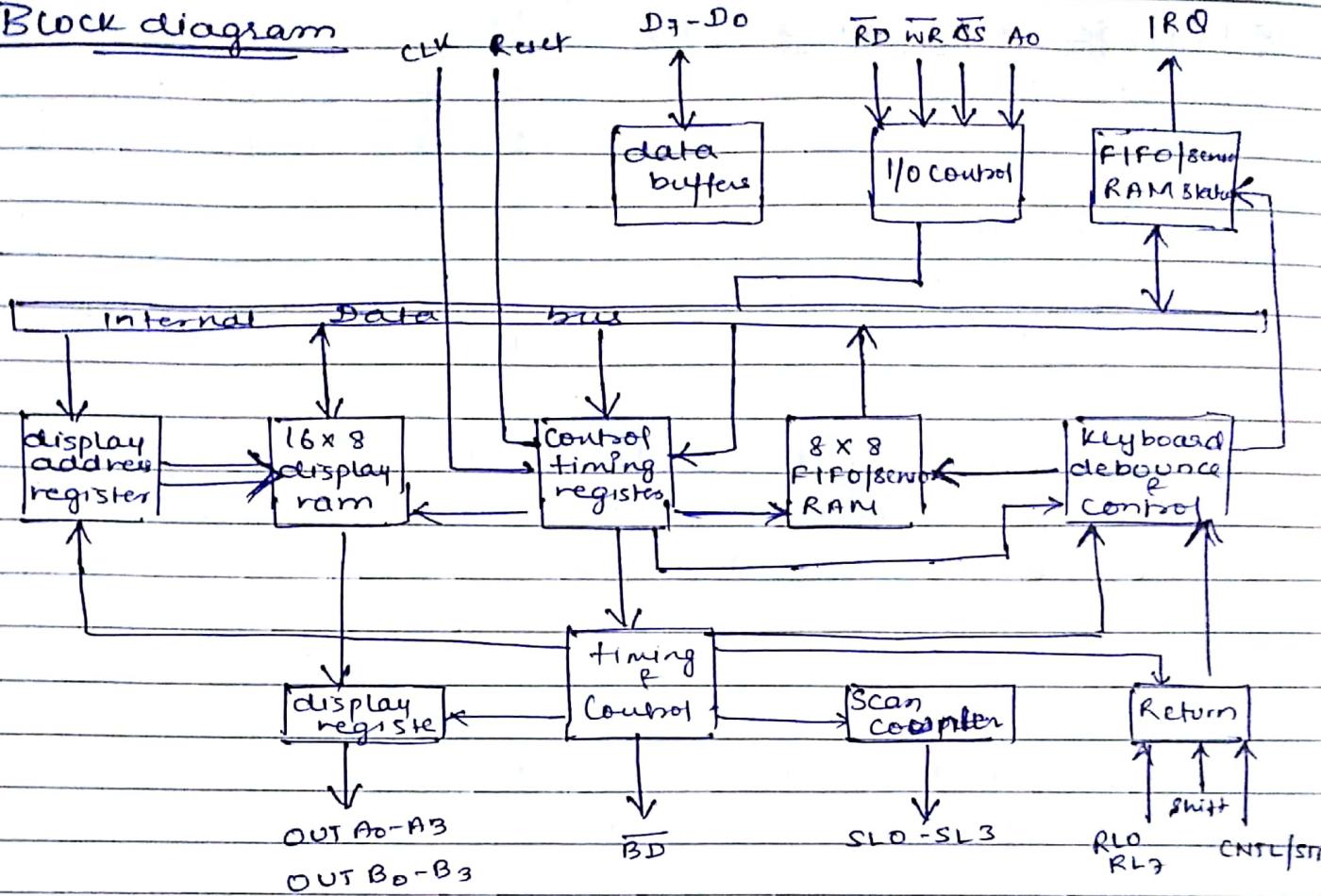
Keyboard scanning is performed to detect status of key. It is a continuous process becoz programmer can press the key ~~at~~ anytime at any instant.

Similarly Seven Segment display is connected in multiplexed mode, hence displaying characters is also a continuous opn.

8279 provides

- (i) A set of 4 scan lines & 8 lines for interfacing keyboards
- (ii) A set of 8 output lines for interfacing display.

### Block diagram



### Control & timing register

Store Keyboard & display modes & other operating cond<sup>n</sup> programmed by CPU.

written with A<sub>0</sub> = 1 and WR = 0

CLK - used to generate internal device timings & is connected to clock generator output.

Reset - high on this input forces 8279 to an idle state.

BD - (blank display) - used to blank the display during digit switching by a blanking command.

## FIFO | Sensor RAM ~~and~~ and status logic

- Status logic generates an interrupt request after each FIFO read opn till the FIFO is empty.

IRQ - the interrupt output line goes high when there is a data in the FIFO sensor RAM

It goes low with each FIFO read opn.

## Display Address registers & display RAM

- Hold address of the word currently being written/read by CPU to form the display RAM.

- Contents of registers are automatically updated by 8279 to accept next data entry by CPU.

- 16 byte RAM contains 16 bytes of data to be displayed on the sixteen 7-segment displays in encoded scan mode

## OUTA<sub>0</sub>-OUTA<sub>3</sub> & OUTB<sub>0</sub>-OUTB<sub>3</sub>

output ports of two 16x4 internal display refresh register.

## Scan mode

scan counter has two modes to refresh key matrix and refresh the display.

- In encoded mode, counter provides binary count that is to be externally decoded to provide the scan line for keyboard & display.

- In decoded mode, the counter internally decodes the least significant 2 bits to provide a decoded 1 out of 4 scan SL<sub>0</sub>-SL<sub>3</sub>.

## Return Buffers

- scans key closure row-wise. If a key closure is detected, the Keyboard debounce unit debounces the key entry i.e. wait for 10ms.

After debounce period, if the key continues to be pressed, the code of the key is directly transferred to screen RAM.

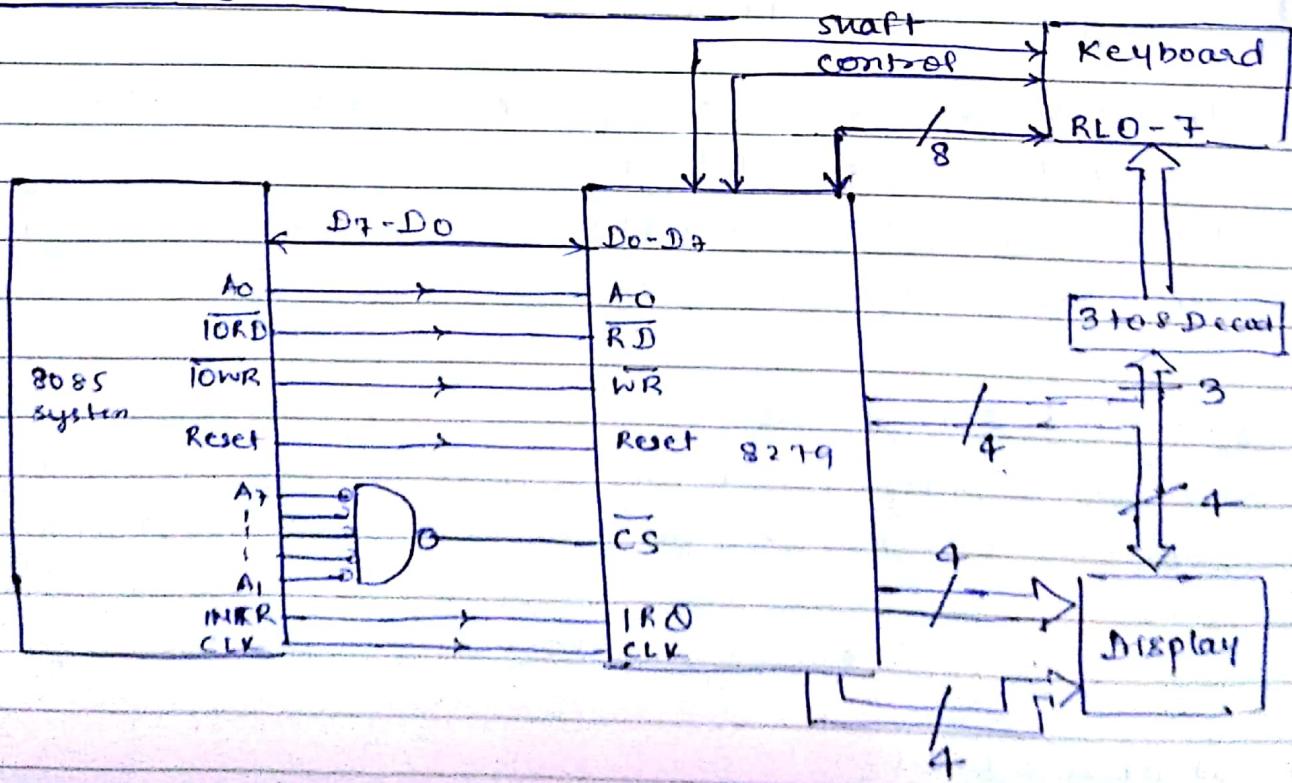
## RLO - RL<sub>f</sub> (return lines)

Shift - status of shift input line stored along with each key code in FIFO in scanned keyboard mode

## CNTL / STB - controlled / strobed I/O mode.

- used as control input and stored in FIFO in a key closure.
- enters data into FIFO RAM, in strobed I/O mode

## # interfacing of 8279 with 8085



## Operational modes of 8279

Imp

two modes of operation on 8279 - Input mode & Output mode

### Input mode

This mode deals with the input given by the keyboard and this is further classified into 3 modes.

#### (i) Scanned Keyboard mode

In this mode, key matrix can be interfaced with using either encoded or decoded scans.

In encoded scans, an 8x8 Keyboard or in decoded scan 4x8 Keyboard can be interfaced.

The code of key pressed with SHIFT & CONTROL status is stored into FIFO RAM.

Keys are automatically debounced with 2-key lockout or N key roll over.

In 2 key lockout if 2 keys are pressed simultaneously only first key is recognized.

In N-key rollover simultaneous keys are recognized and stored in internal buffer.

#### (ii) Scanned Sensor matrix

A sensor array will interface with 8279 with encoded (8x8 matrix) or decoder (4x8 matrix) scan lines.

Key status are stored in RAM addressable by CPU.

#### (iii) Strobed Input

In this mode, when control line is set to 0, the data on the return lines stored in FIFO byte by byte.

### Output mode

This mode deals with display related operations.

#### (i) Display Scans

Allows 8/16 character multiplexed displays to be organized as dual 4-bit/8 single 8-bit display unit.

#### (ii) Display entry

allows data to be entered for display either from right/left side.

### # Status word of 8279

Contains FIFO status, error & display Unavailable signals.

This word is read by CPU when  $A_0 = 1$  and  $\overline{CS}$  &  $\overline{RD}$  are low

