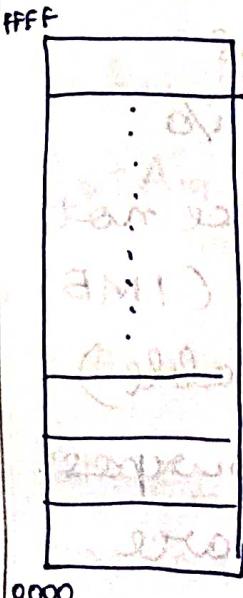


## I/O Interfacing

- I/O space is present
- 64 KB of I/O space
- Ports: I/O Space
- Address location: 8 bit



↳ If ports mapped to I/O space: I/O mapped I/O ports  
↳ If ports mapped to memory space:  
    Memory mapped I/O ports

{ E0000 - E0FFF }  $\Rightarrow$  communicating with I/O

in memory space

Byte : 1 port

Word : 2 ports

Port no: 0 - 255  $\Rightarrow$  Page 0 port numbers  
(Special ports)

{ 0000 - 00FF }  $\rightarrow$

{0100-FFFF}  $\Rightarrow$  other normal ports

Isolated I/O      Mem-mapped I/O

$\hookrightarrow$  65,536 ports  
(0000<sub>h</sub>-FFFF<sub>h</sub>)

$\hookrightarrow$  Many I/O devices

$\hookrightarrow$  only 4096 ports available  
(E000<sub>h</sub>-E0FFF<sub>h</sub>)

$\hookrightarrow$  If processor does not take a lot of I/O devices

in : taking i/p from I/O devices

out : writing / sending data to o/p devices

$\hookrightarrow$  Instruction time very less

$\hookrightarrow$  special unit for isolated I/O

$\hookrightarrow$  Entire space available for data storing

$\hookrightarrow$  Entire space not available (1MB not available)

$\hookrightarrow$  IN

OUT

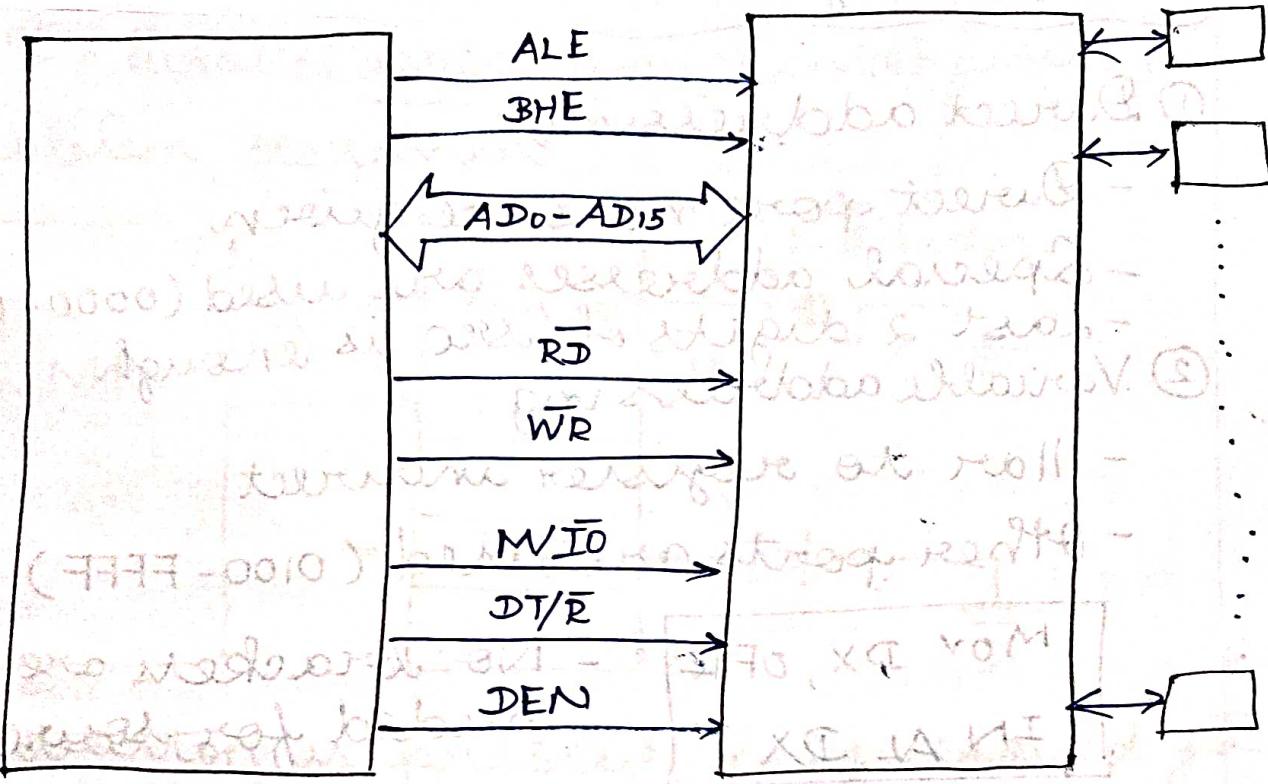
Involves only with accumulator and so, faster

$\hookrightarrow$  Other gen. purpose registers are used

Byte wide : AL

Word " : AX

Min mode



Only 16-bit address line is used

$A_{16} - A_{19}$   $\Rightarrow$  carries only status bits & port address not carried

I/O instructions:

- ① IN AL, CO
- ② IN AX, CO
- ③ OUT CQ, AL
- ④ OUT CQ, AX

Direct addressing

Similar to 'mov' instruction  
No flags affected

Destination: Always register

Source: Port address

## Addressing mode:

### ① Direct addressing:

- Direct port no.s are given
- Special addresses are used (0000-~~0FFF~~ OFFF)
- Last 2 digits of SEC is enough

### ② Variable addressing:

- Illar to register indirect
- Other ports are used (0100- FFFF)

```
Mov DX, DF12  
IN AL, DX
```

- No brackets are needed for source

NOTE: ~~Source and destination first~~

Page 0 address: Direct addressing mode  
Other "variable"

### Conditions:

- ① DX must be used for storing address
- ② Only accumulator is used for storing data

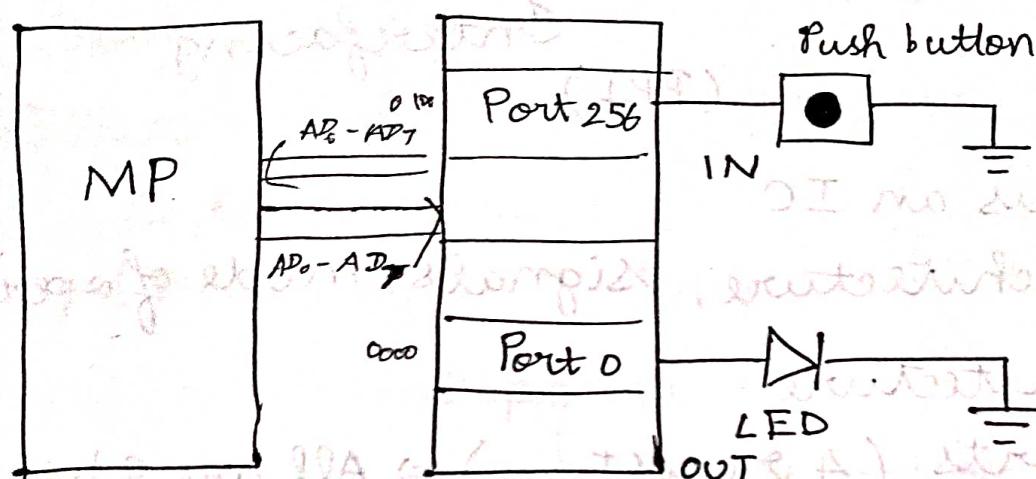
### Variable A.M:

```
MOV DX, 0101  
IN AL, DX
```

8251 - Serial processing of character &

8255 - Parallel processing → interfacing

Problem Statement:



If button is pressed, light should glow  
Else, not glow

ON - 01<sub>16</sub> } hexadecimal values  
OFF - 00<sub>16</sub>

MOV DX, 0100

IN AL, DX

OUT 00, AL

Write another program for blinking  
the LED

{ MOV AL, 00 }

OUT 01, AL }

{ Delay }

INC AL

OUT 01, AL }

Loop

2 Standards for I/O interfacing:

① 8255

## Programmable Peripheral Interfacing

multiple ports

(PPI)

- It is an IC

- Architecture, signals, mode of operation

Architecture:

① 4 ports: (4 registers)  $\Rightarrow$  All are 8 bit

3 ports: A, B, C  $\rightarrow$  C is not a register

Data collected for o/p devices from

I/P devices drops data at

- Dual in-line

$PA_0 - PA_7, PB_0 - PB_8$

- Byte wide ports : A, B  $\Rightarrow$  8 bit

Nibble wide 2 ports: C  $\Rightarrow$  4 bit + 4 bit

$PC_0 - PC_3, PC_4 - PC_7$

1 Control registers

- Controls how the ports function  
- To configure the ports

↳ All ports are bidirectional

↳ So, users can configure which ports acts as i/p or o/p  $\Rightarrow$  So, called as PPI

↳ To configure:

Control word to be written to control register

Processor sends 2 words to PPI

1) Control word:

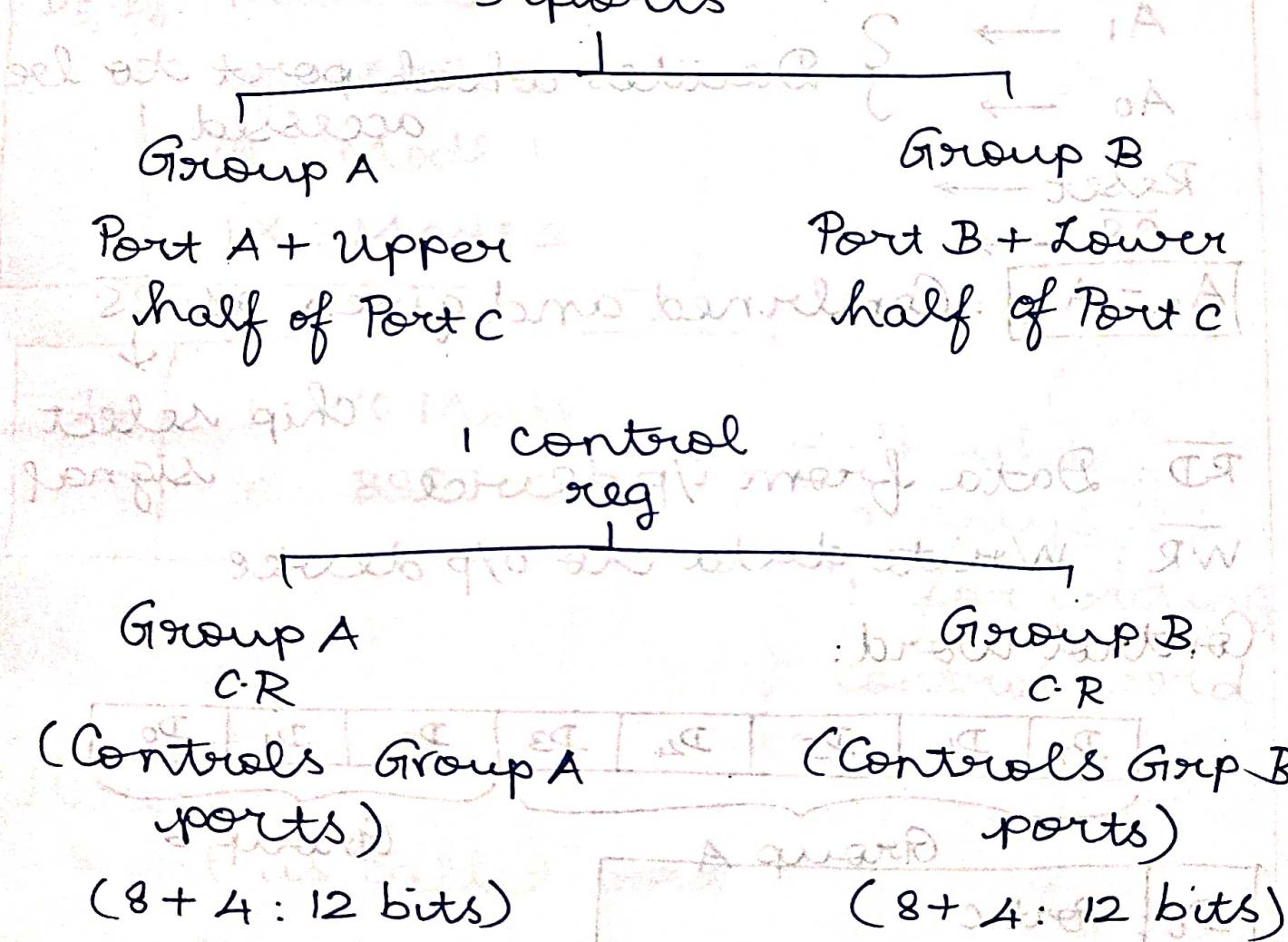
Gr to control register for configuration

2) Data word:

Bidirectional

(P → M or M → P)

3 ports



② Internal bus communicates data from and to ports and processor (Bidirectional)

③ Data Bus Buffer:

- Connected to processor
- Data gr to internal bus, which in

twin sends the data to the port  
and vice versa

### ④ Read / Write Control Logic:

- Controls read / write and for

Selecting the ports

$\overline{RD}$  →

$\overline{WR}$  →

$A_1$  →

$A_0$  →

Reset →

$\overline{CS}$  →

$A_2 - A_{16}$  : Combined and given as  $\overline{CS}$

} Decides which port to be accessed

$\overline{RD}$  : Data from i/p device

chip select signal

$\overline{WR}$  : Write data to o/p device

Control Word:

$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$
Group A				Group B			
$D_0$ : Port C (Lower: $PC_3 - PC_0$ ) I : I/P O : O/P				$D_0$ : Port A + 8 $\Rightarrow$ Group B			
$D_1$ : Port B I : I/P O : O/P							
$D_2$ : Mode Selection 0 : Mode 0 1 : Mode 1							

$D_3$	Port C (Upper) $(PC_7 - PC_4)$ 1 : I/P 0 : O/P
$D_4$	Port A 1 : I/P 0 : O/P
$D_5, D_6$	Mode Selection 00 : Mode 0 01 : Mode 1 1X : Mode 2

$\Rightarrow$  Group A

$D_7$	1 : I/O Mode 0 : BSR Mode
-------	------------------------------

$\Rightarrow$  Should always be '1' for writing control word

Here:

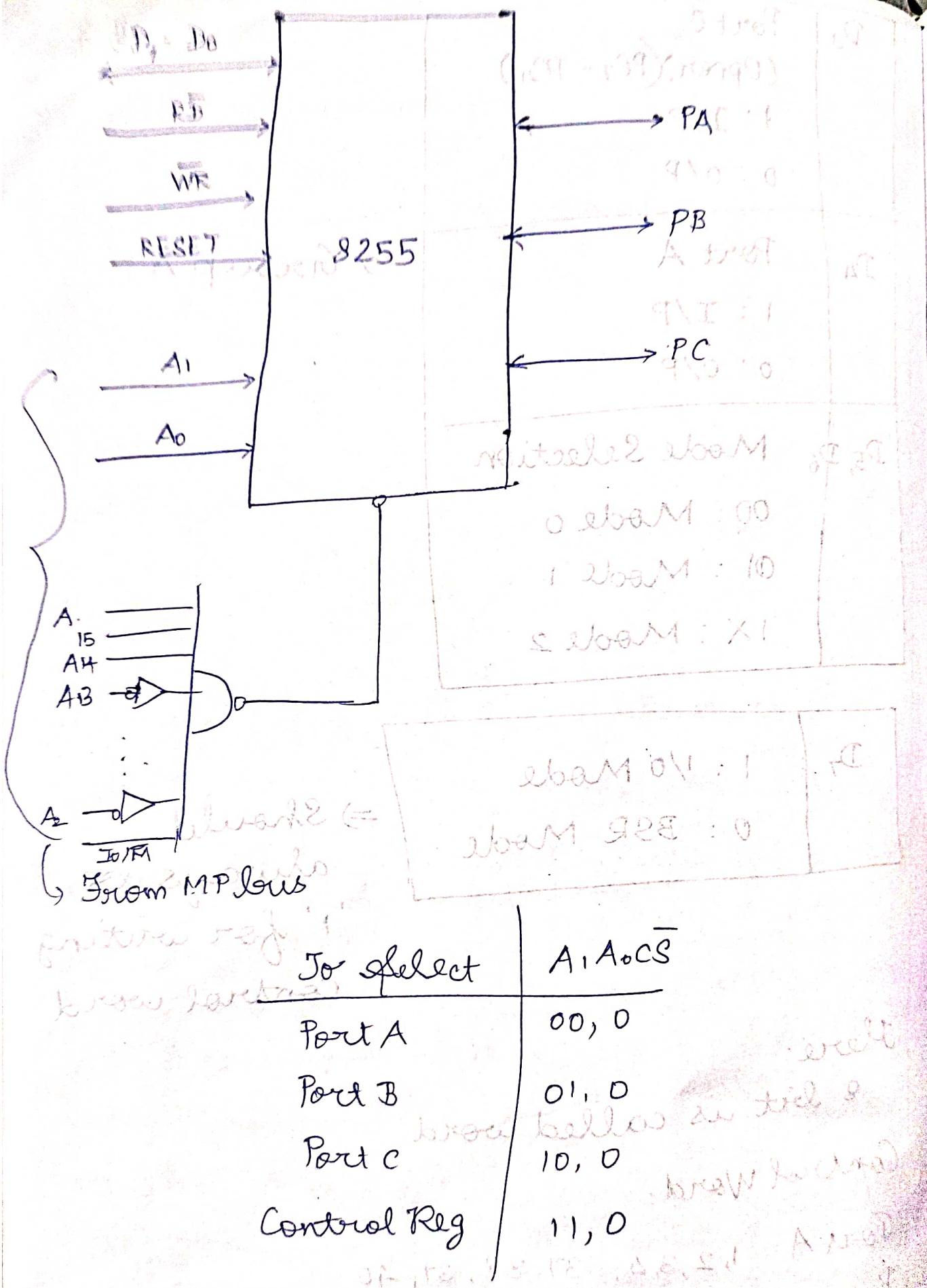
8 bit is called word

Control Word:

Port A : 1, 2, 3, 4, 37, 38, 39, 40

Port B : 18, 19, 20, 21, 22, 23, 24, 25

Port C : 10, 11, 12, 13, 14, 15, 16, 17



Ex:

1) Control word #0

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

$$80 = 80_{10}, 1 = \text{SEL}$$

I/O Mode :  $D_7 = 1$

$D_5, D_6 = 00$ : Mode 0

$D_4 = 0 \Rightarrow \text{Port A} = \text{O/P}$

$D_3 = 0 \Rightarrow \text{Port C (Upper)} = \text{O/P}$

$D_2 = 0 \Rightarrow \text{Mode 0}$

$D_1 = 0 \Rightarrow \text{Port B} = \text{O/P}$

$D_0 = 0 \Rightarrow \text{Port C (Lower)} = \text{O/P}$

All ports act as output line and act in Mode 0 operation

2) Control word #1

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

$$82 = 82_{10}, 1 = \text{SEL}$$

$D_7 = 1$  : I/O

$D_5, D_6 = 00$ : Mode 0

$D_4 = 0 \Rightarrow \text{Port A O/P}$

$D_3 = 0 \Rightarrow \text{Port C O/P}$

$D_2 = 0 \Rightarrow \text{Mode 0}$

$D_1 = 1 \Rightarrow \text{Port B} = \text{I/P}$

### 3) Control word #2

1|0|0|0|0|0|0|1

81<sub>H</sub> (All three I/O ports)

D<sub>7</sub> = 1 : I/P

D<sub>6</sub> = 1 : Lower Port C input

All other ports I/P

Mode 0

D<sub>5</sub>, D<sub>6</sub> : Mode 0

D<sub>4</sub> : Port A I/P

D<sub>3</sub> : Upper port C i/p

D<sub>2</sub> : 0 Mode 0

D<sub>1</sub> : 0 Port B i/p

### 4) Control word #3

1|0|0|0|0|0|1|1

storage 0 ebam

83<sub>H</sub>

D<sub>7</sub> = 1 : I/P

D<sub>5</sub>, D<sub>6</sub> = 00 : Mode 0

D<sub>4</sub> : Port A I/P

D<sub>3</sub> : Upper port C i/p

D<sub>2</sub> : 0 : Mode 0

D<sub>1</sub> = 1 : Port B o/p

D<sub>0</sub> = 1 : Lower port C o/p

80<sub>H</sub> : All 3 ports : Mode 0, O/P port

82<sub>H</sub> : Port A,C = O/P port, Port B = I/P port,  
mode 0

81<sub>H</sub> : All ports : I/P, mode 0

83<sub>H</sub> : Port A, Upper port C : I/P, Port B, Lower C : O/P  
mode 0

Mode 0

2 Byte wide I/O P

1 Nibble wide O/I P

Control word = ?

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	1	0	0	1

$$1 \Rightarrow [1 | 0 | 0 | 1 | 0 | 0 | 1 | X] \rightarrow 10010010 \\ 10010011$$

$$2 \Rightarrow [1 | 0 | 0 | 1 | X | 0 | 1 | 0] \rightarrow 10010010 \\ 10011010$$

Modes of operation:

Mode 0: All 3 ports handles I/O

Basic I/O

Mode 1: A, B  $\Rightarrow$  I/O port

Strobe I/O C = controls transfer thru A, B

Mode 2: A  $\Rightarrow$  I/O port

Bi-Dir Bus C  $\Rightarrow$  some lines control bidirectional data transfer

What is the mode & I/O config

FC<sub>16</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	1	1	1	1	1	1	0

D<sub>7</sub> = 1 : I/O mode

D<sub>5</sub>, D<sub>6</sub> = 1, 1 : Mode 2

D<sub>4</sub> = 1 : I/O Port A

D<sub>3</sub> = 1 : Upper C I/O

D<sub>2</sub> = 1 : Mode 1

D<sub>1</sub> = 0 : O/I Port B

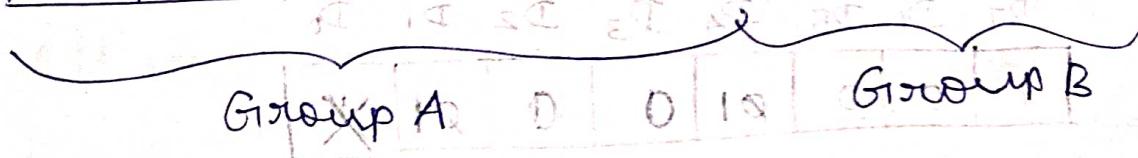
D<sub>0</sub> = 0 : Lower C O/I

# Mode 1: Strobed I/O:

I/P config

a) SIE after strobe

I/O	I/O	IBFA	INTE <sub>A</sub>	INTR <sub>A</sub>	INTE <sub>B</sub>	IBFB	INTR <sub>B</sub>
-----	-----	------	-------------------	-------------------	-------------------	------	-------------------



O/P config

b)

OBFA	INTEA	I/O	I/O	INTR <sub>A</sub>	INTE <sub>B</sub>	OBFB	INTR <sub>B</sub>
------	-------	-----	-----	-------------------	-------------------	------	-------------------



## Mode 1: status information of port C

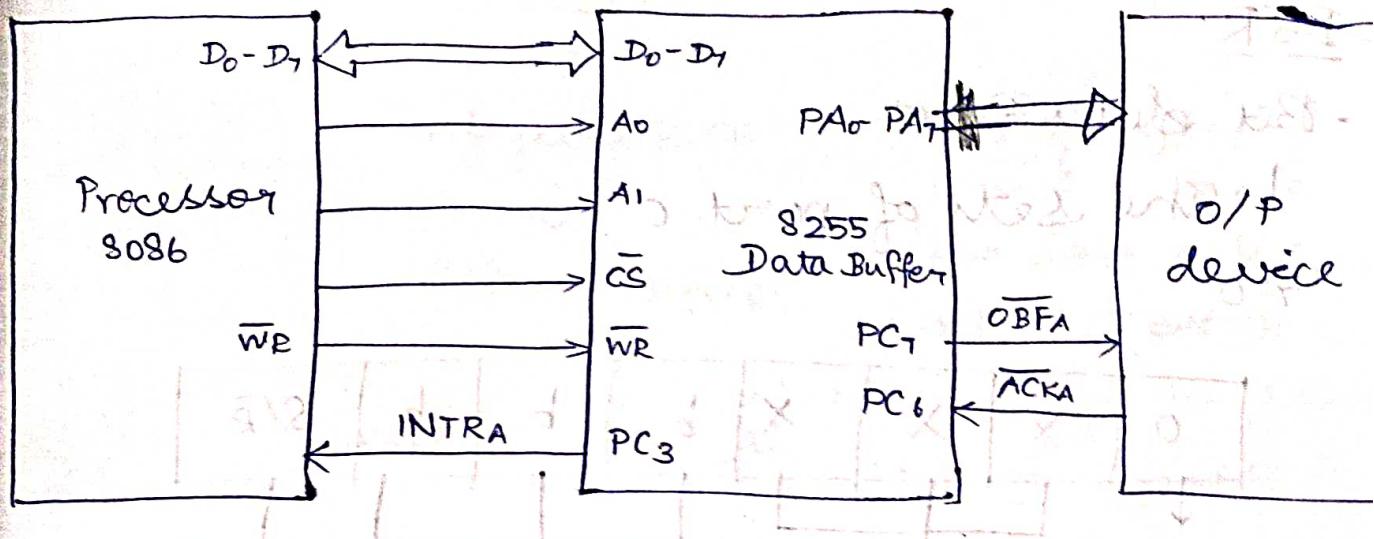
Port A	Port B	PC <sub>0</sub> - PC <sub>7</sub>	J/P=1 O/P=0
write 0 opened	write 0	b	out 1
0	req INT A	c	out 2
write 0	req INT B	d	out 3
0	free PJE & use LAN medium	a	out 4

c)

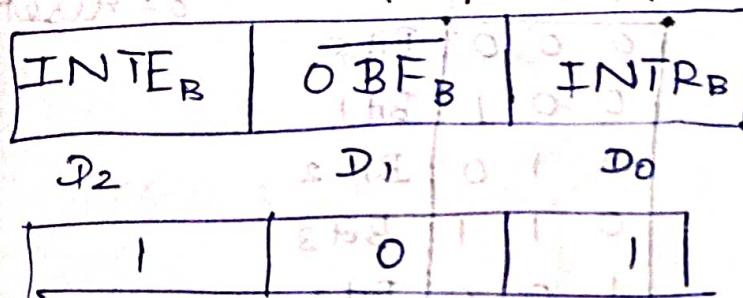
OBFA	INTE <sub>A</sub>	I/O	I/O	INTR <sub>A</sub>	INTE <sub>B</sub>	IBFB	INTR <sub>B</sub>
------	-------------------	-----	-----	-------------------	-------------------	------	-------------------

d)

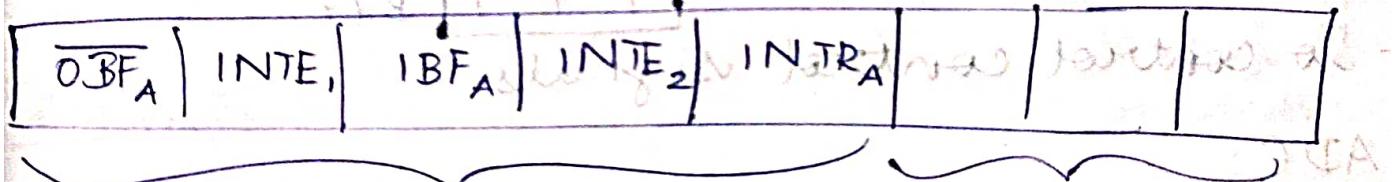
I/O	I/O	IBFA	INTE <sub>A</sub>	INTR <sub>A</sub>	INTE <sub>B</sub>	OBFB	INTR <sub>B</sub>
-----	-----	------	-------------------	-------------------	-------------------	------	-------------------



Q. What is  $PC_2$ ,  $PC_1$ ,  $PC_0$  if B should be made as O/P port?



Mode 2: Strobed Bidirectional I/O



Group A

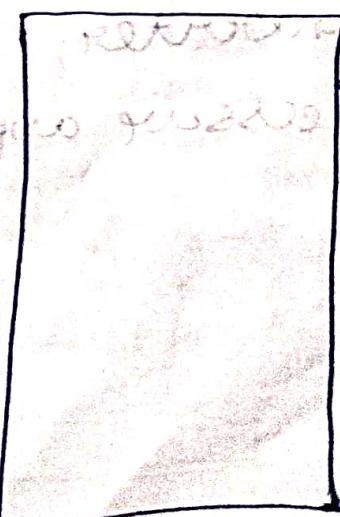
Grp B

Group A mode 2 port A I/P  $PC_0 - PC_2$

Group B mode 1 port B O/P  $PC_3 - PC_7$

1 1 1 1 1 1 0 0

Input & Output Register

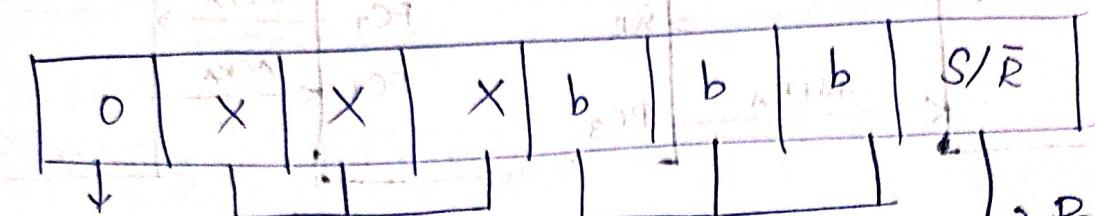


## BSR:

- Bit Set & Reset

↳ The sets of port C

$$D_7 = 0$$



BSR mode

Don't care

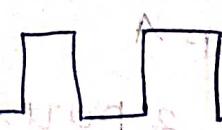
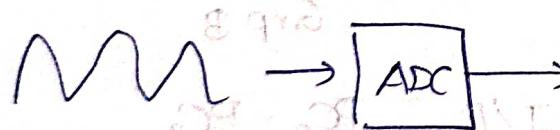
Port C select

Set  
1 = Set  
0 = reset

0	0	0	Bit 0
0	0	1	Bit 1
0	1	0	Bit 2
0	1	1	Bit 3
1	0	0	Bit 4
1	0	1	Bit 5
1	1	0	Bit 6
1	1	1	Bit 7

- To control control signals

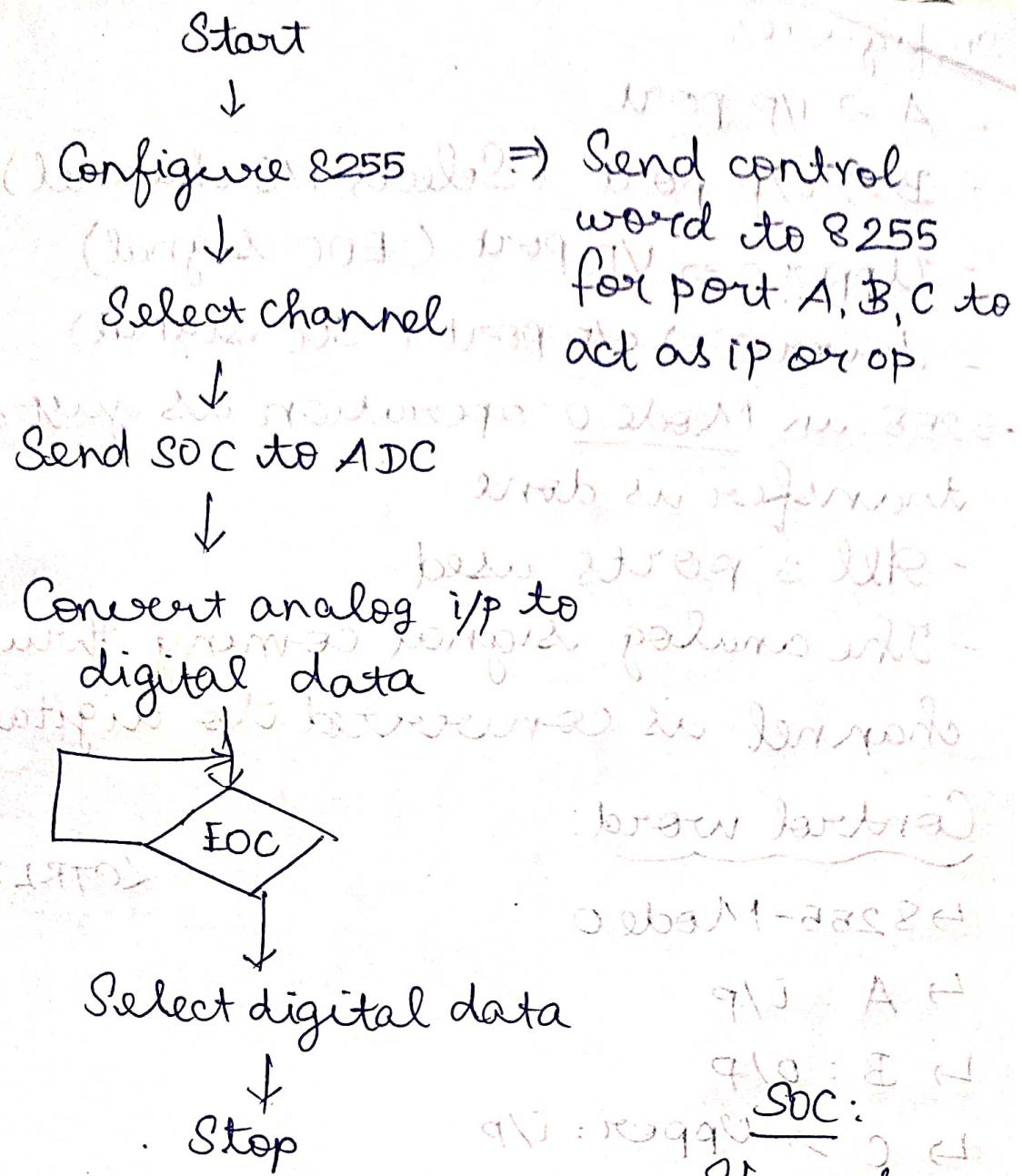
## ADC:



- ADC 0808/0809

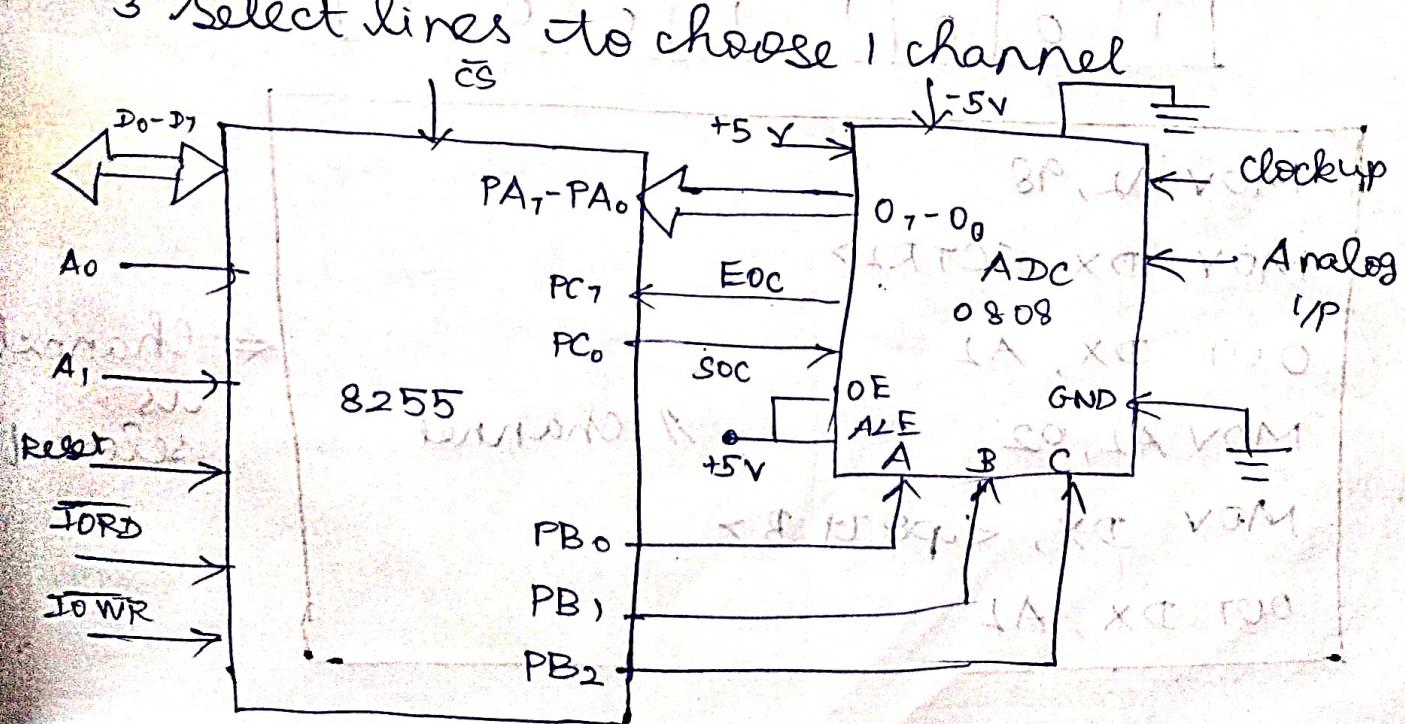
- 8 bit converter

- Uses successive approximation technique



## Channels:

- 8 channels
  - 3 Select lines to choose 1 channel



In figure:

- A  $\Rightarrow$  i/p port
- B  $\Rightarrow$  o/p port (Selects 1/8 channel)
- Upper C  $\Rightarrow$  i/p port (EOC signal)
- Lower C  $\Rightarrow$  o/p port (SOC signal)

8255 in Mode 0 operation as just data transfer is done

- All 3 ports used
- The analog signal coming thru selected channel is converted to digital signal.

Control word:

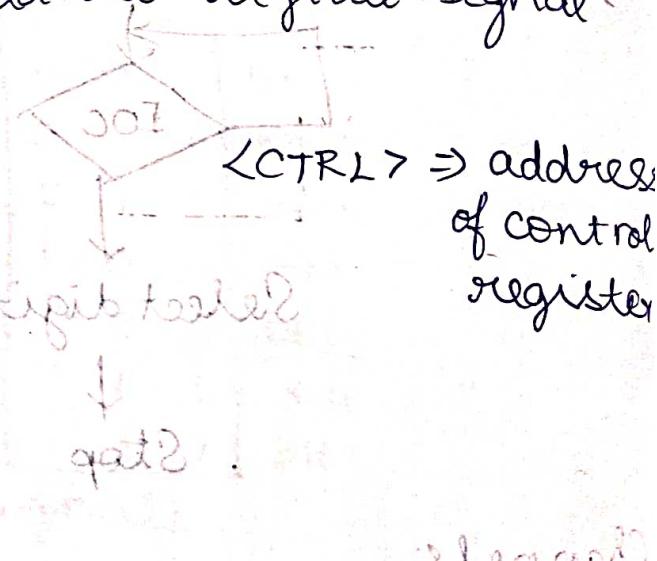
↳ 8255-Mode 0

↳ A : i/p

↳ B : o/p

↳ C  $\rightarrow$  Upper : i/p

↳ Lower : o/p



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

<CTRL>

MOV AL, 98

MOV DX, <CTRL>

OUT DX, AL

MOV AL, 02

MOV DX, <port B>

OUT DX, AL

// Channel

← Channel  
is  
selected

Step 2:  $\text{SOC} \leftrightarrow \text{PORT C}_L (\text{OUT})$

$\text{MOV AL, SOC} \quad 01$

$\text{MOV DX, } <\text{PORT C}_L>$

$\text{OUT DX, AL}$

base with pin selection logic

Step 3:  $\text{EOC} \leftrightarrow \text{PORT C}_D (\text{IN})$

$\text{MOV DX, } <\text{PORT C}_D>$

$\text{IN AL, DX}$

Step 4:  $\text{AL} \leftarrow \text{MSB} \leftarrow 1 \xrightarrow{\text{NOT}}$  waiting

$\text{MOV AL, MSB}$

$\text{CMP AL, } 01$

~~JNZ LOC A~~

~~JZ EOP LOC A~~

~~JNC LOC A~~

$\text{MOV AL, } 01$

$\text{MOV DX, } <\text{Port C}_2>$

$\text{OUT DX, AL}$

$\text{MOV DX, } <\text{Port C}_0>$

$\text{IN AL, DX } <\text{LI}>$

~~MOV SI, 2000~~

~~MOV [SI], AL~~

~~AND AL, 01~~

~~CMP AL, 10~~

~~JNZ <LI>~~

~~HLT~~

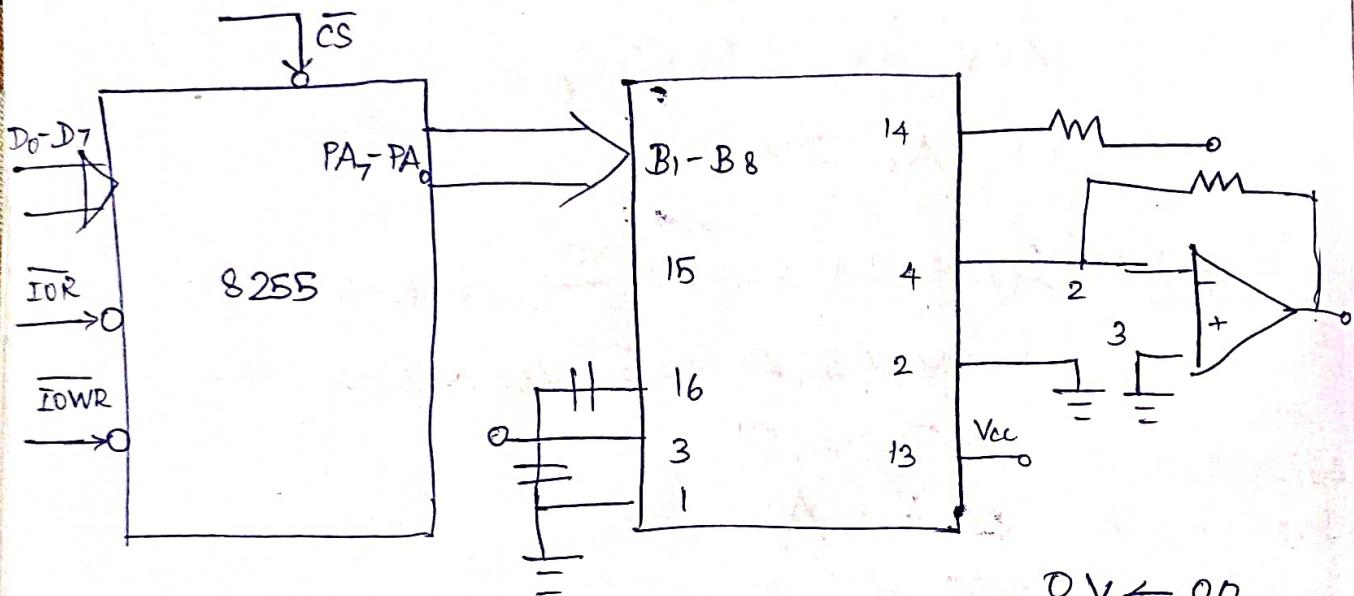
Checking EOC

MOV DX, <PortA>  
 IN AL, DX  
 HLT

Digital data

DAC interfacing thru 8255

DAC0800 - 8 bit converter



DAC:



Give 0 to AL

Give AL to PortA

Give FF to AL

OUT AL, <PortA>

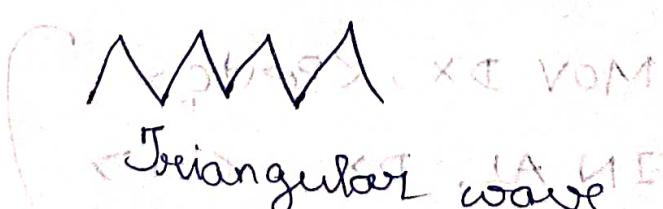
In falling edge:

Loops AL ← FF

DEC AL,

OUT AL, <PortA>

until AL = 00



0002 (2nd vrom)

1A.5E7 (1M)

0002 (2nd vrom)

1A.5E7 (1M)

0002 (2nd vrom)

1A.5E7 (1M)

Rising edge:  $AL \leftarrow 00$  } Port A  
 $AL \leftarrow FF$  }

Falling edge:  $\begin{array}{l} \text{DEC AL} \\ \text{CMP AL, 00} \\ \text{LOOP} \end{array}$  } Port A

$MOV AL, 00$   
 $MOV DX, <PORTA>$   
 $OUT DX, AL$   
 $MOV AL, FF$   
 $OUT DX, AL$

$\begin{array}{l} \text{DEC AL} \\ \text{CMP AL, 00} \end{array}$

$JNZ <A>$

$JZ <B>$

$HLT$

$<B>$

Rising edge

$<A>$

$OUT DX, AL$

Falling edge