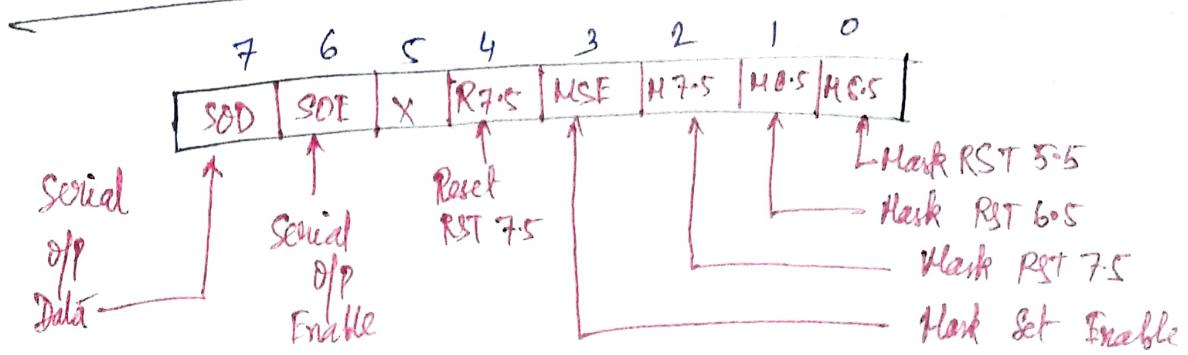


Masking Interrupts:

SIM (Set Interrupt Mask) Function

- 1) Set Mask for RST 7.5, 6.5 and 5.5.
- 2) Reset RST 7.5
- 3) Supplement Serial I/O.

Accumulator content for SIM



Bit 0-2 mask the interrupts RST 7.5, RST 6.5, RST 5.5 if the corresponding bit is set to 1. If the bit is set to 0, corresponding bit is enabled.

Bit 3 is set to 1 to make the bits 0-2 effective. (MSF).

Bit 4 is additional control for RST 7.5. If it is set to 1, the flop flop for RST 7.5 is reset, irrespective of the fact whether the bit 2 of the same instruction is 0 or 1.

Bits 6 and 7 are used for serial data output. If SIM instruction is executed, then content of 9th bit of accumulator comes as the output on the SOP line of JP.

Q) Enable all the interrupts of Intel 8085.

Ans:- EI

MOV A, 08H.

SSM

Q) Enable RST 6.5 disable RST 2.5 and RST 5.5.

0000110101

Q) PENDING INTERRUPTS :-

RIM (Read Interrupt Mask) :-

- i) This instruction loads the accumulator indicating the current status of the interrupt mask, identifying pending interrupt.
- ii) Identifies pending interrupt.
- iii) Received serial data.

o Accumulator after execution of RIM :-

7	6	5	4	3	2	1	0
C	I	S	E	S	S	I	H

↓ These bits are set to 1 if serial input data corresponding interrupt are pending.

↓ These bits are set to 1 if interrupt is enabled.

↓ These bits are set to 1 if the interrupt are masked.

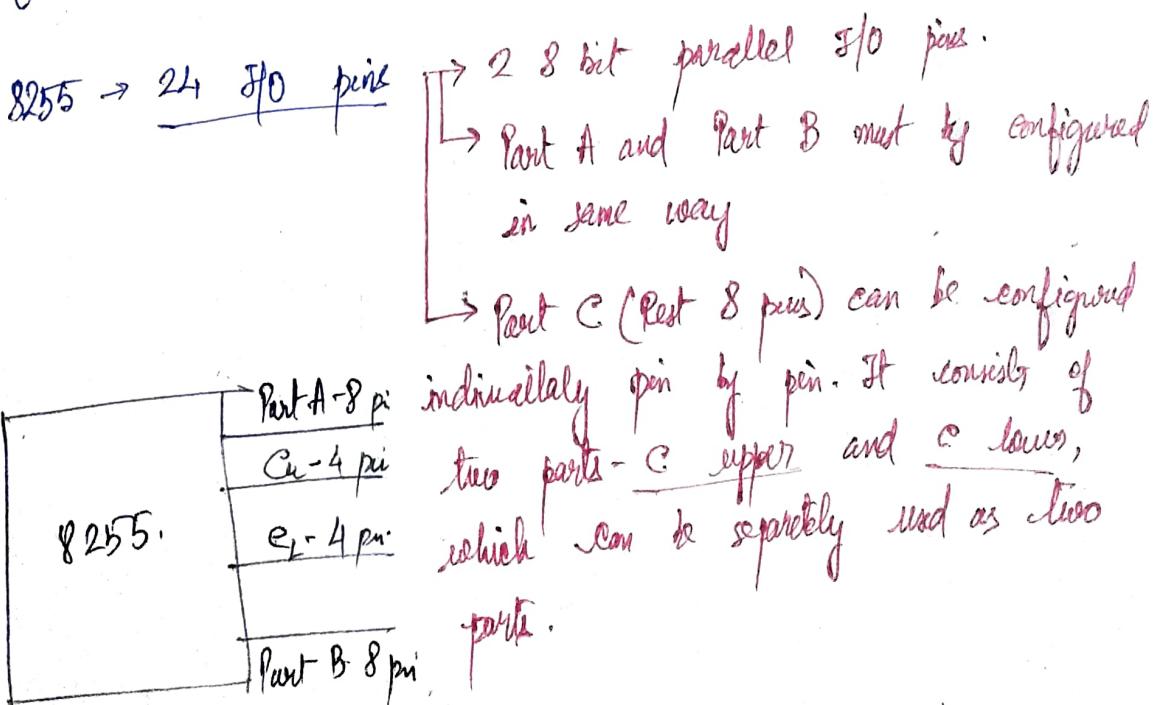
RJM instruction is used to read serial data on the SDA pin of processor. Data on SDA pin is stored with MSB of Accumulator.

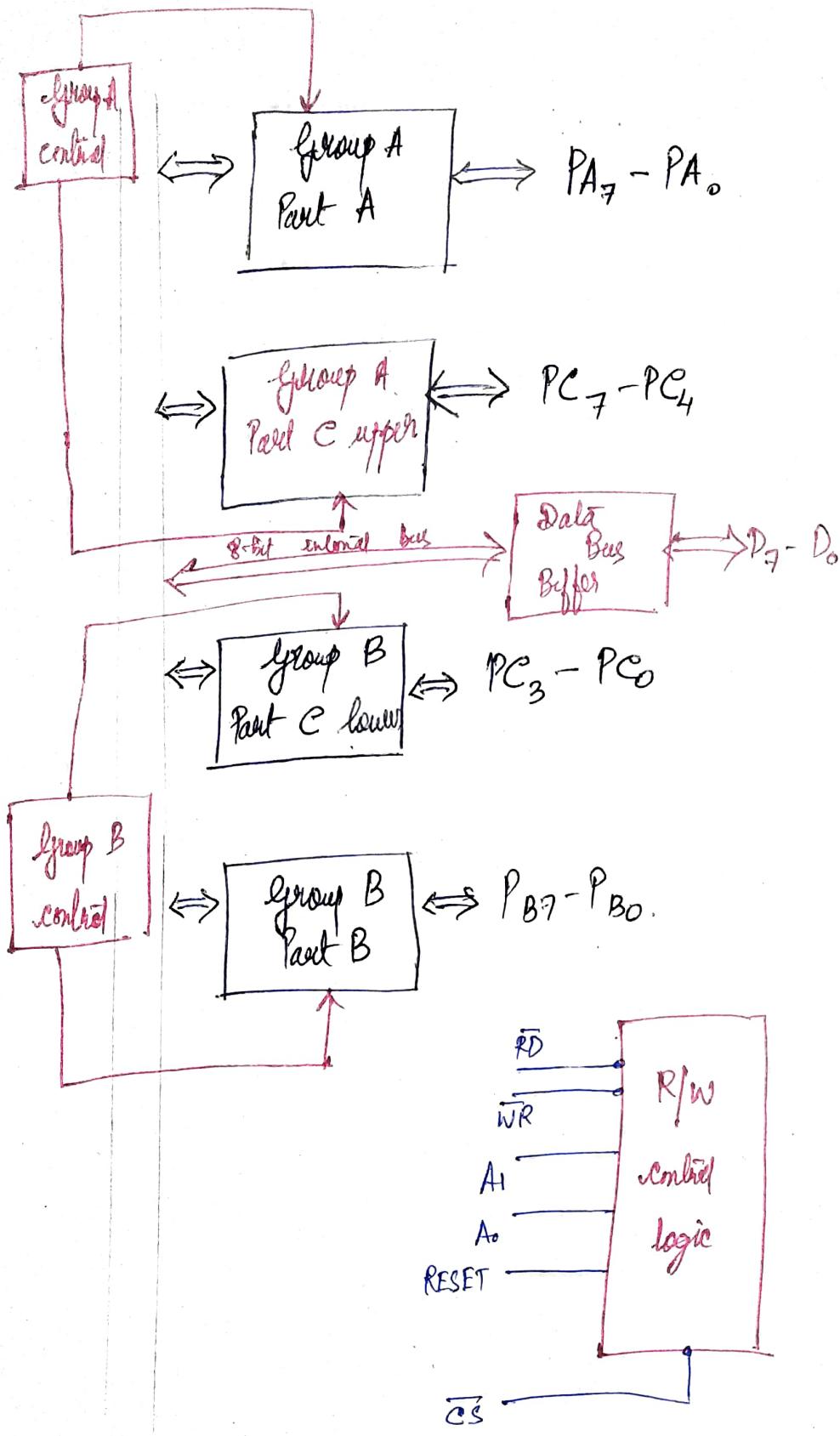
Q) Assuming the processor is completing an RST 7.5 interrupt request. Check to see if RST 6.5 is pending. If it is pending, enable RST 6.5 without affecting any other interrupt otherwise return to main program.

PERIPHERALS

8255A Programmable Peripheral Interface

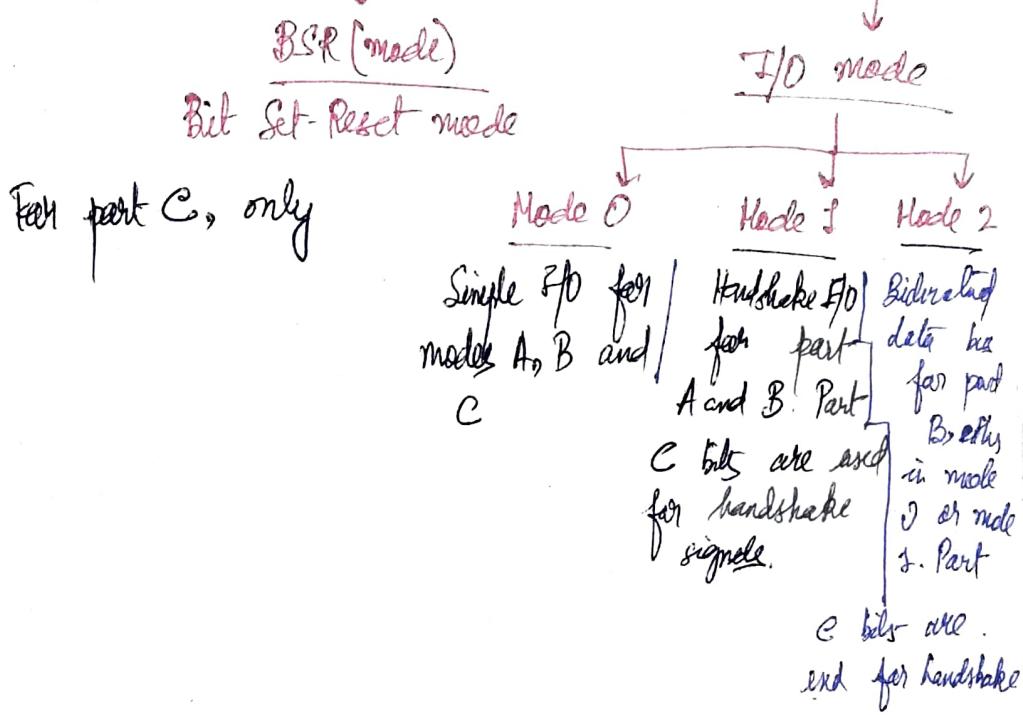
It is a general purpose, programmable, versatile, economic parallel I/O device. It is mainly used for several types of data transfer. Parallel I/O operations are possible.





Along with this, there is a control register which is connected to all the parts.

• Modes of operation :-



Mode 0 operation - A part can be operated as a simple Input-output part.

Mode 1 - This mode is called shielded input/output mode. The handshaking signals control the I/O section of the specified part. Part C lines $PC_0 - PC_2$ provide handshake signals for part B. Part C lines $PC_3 - PC_5$ provide handshake signals for part A. PC_6 and PC_7 may be used for independent data transfer.

Mode 2 - It is a shielded bidirectional mode of operation. In this mode, only part A can be programmed to operate as a bidirectional part controlled by part C lines $PC_3 - PC_7$. Part B can be used either in mode 1 or in mode 0.

R/W to Control logic:

RD \rightarrow The MPU reads data from selected I/O port.

WR \rightarrow writes

Reset \rightarrow When this signal goes high, it clears the control register and configures all ports in input mode.

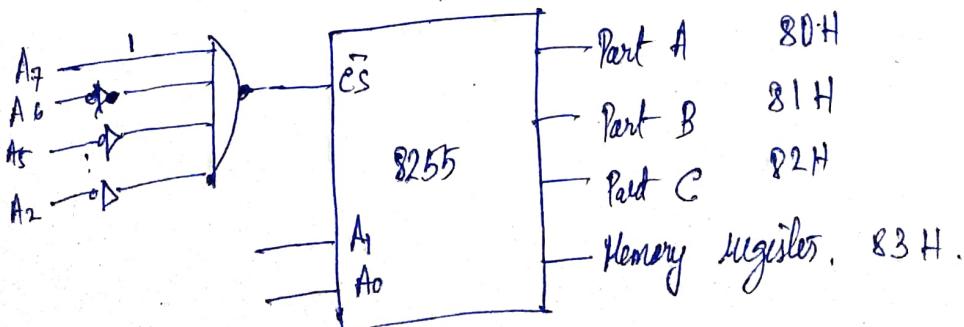
A₀, A₁ are connected to A₀ and A₁ of processor.

CS is connected to interfacing hardware.

CS, A₀, A₁ form a group.

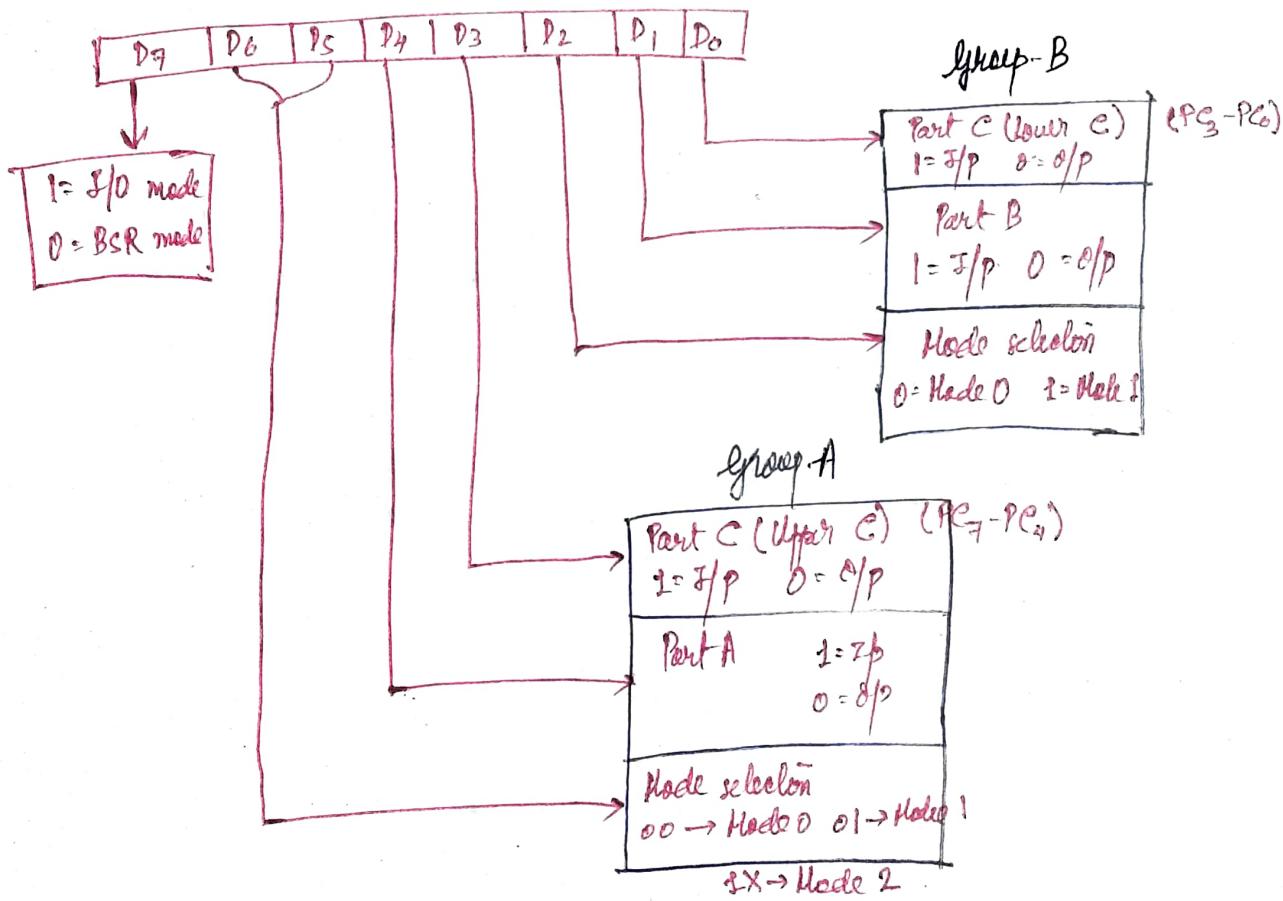
CS	A ₁	A ₀	Selected
0	0	0	Part A
0	0	1	Part B
0	1	0	Part C
0	1	1	Control register
1	x	x	8255 is not selected.

Chip select

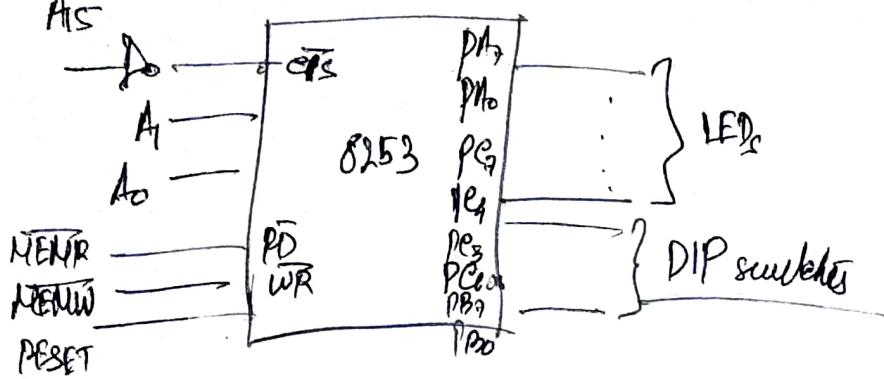


Control word

A control word is formed in a specific format which determines the mode and function of the parts. The CPU transfers the control word to 8255 which accepts the commands from R/W control logic. It uses internal data bus to issue the proper commands to associated parts through the words written in control registers.



Q) Suppose.



Identify the part address. Identify the Mode word to configure Part A and Part C upper as off parts and part B and part C lower as I/O parts. Write a program to read the DIP switches and display the reading from part B and part A and from part C lower and part C upper.

Mode 0 \Rightarrow Part A
~~A15~~ A1 A0
~~A15~~ 0 0

8000H \rightarrow Part A

8001H \rightarrow Part B

8002H \rightarrow Part C

8003H \rightarrow Control register.

Part A \Rightarrow 0 Part B \Rightarrow 1
 Part C₁ \Rightarrow 0 Part C₂ \Rightarrow 1
 00

~~128 64 32 16 8 4 2 1~~
~~1 00 00 01 1~~
~~8 3 H~~
 ~~~~~  
 control word

## Program

MVJ A,83H

STA 8903 H

LDA 8001 H.

STA 8000H

LDA 8002H

ANJ OF H

P10

四

41c

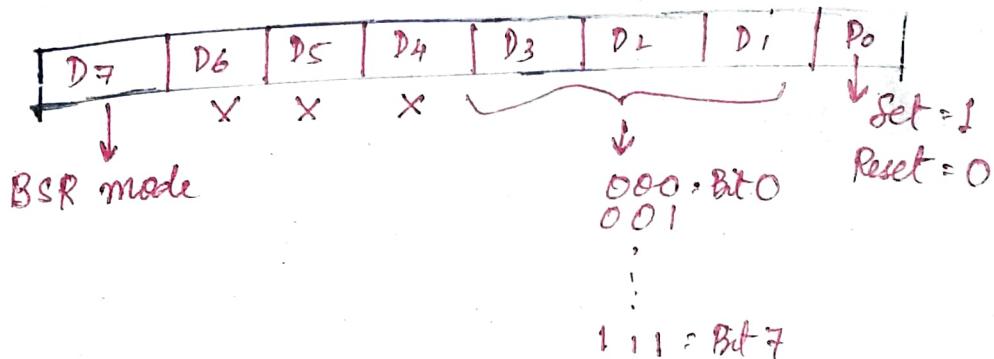
40

STA 8002H

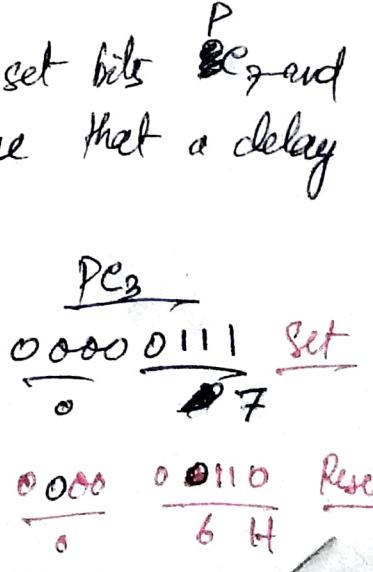
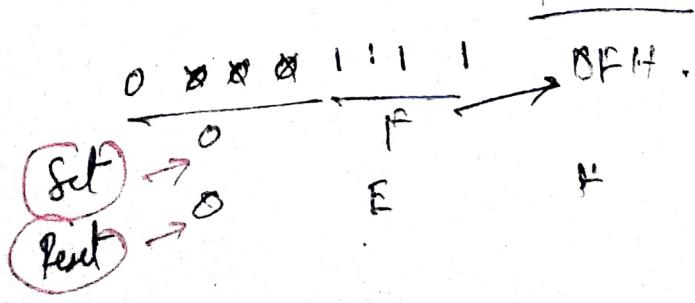
HLT

### BSR Mode - Bit Set Reset Mode

Control word format in BSR mode.



Q) Write a ISR control word subroutine to set bits  $\text{PC}_7$  and  $\text{PB}_3$  and reset them after 10ms. Assume that a delay subroutine is available.



Program

MVI A, 0F8H

OUT 83H

MVI A, 07H

OUT 83H

CALL DELAY

MVI A, 0FH

OUT 83H

MVI A, 0EH

OUT 83H

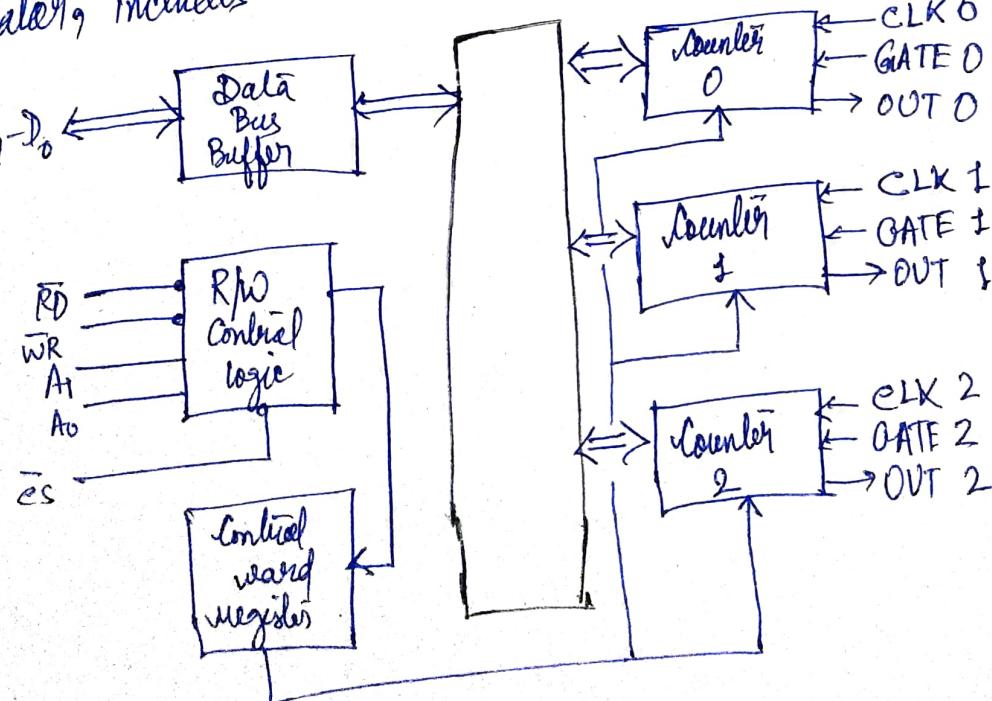
RETURN.

### 8254 Programmable Internal Timer :-

This is basically an software program counter.

#### Applications :-

It generates acute time delay that can be used for real time clock, even counter, digital one shot (single pulse), square wave generation, complex waveform generator, includes 3 16-bit counters.

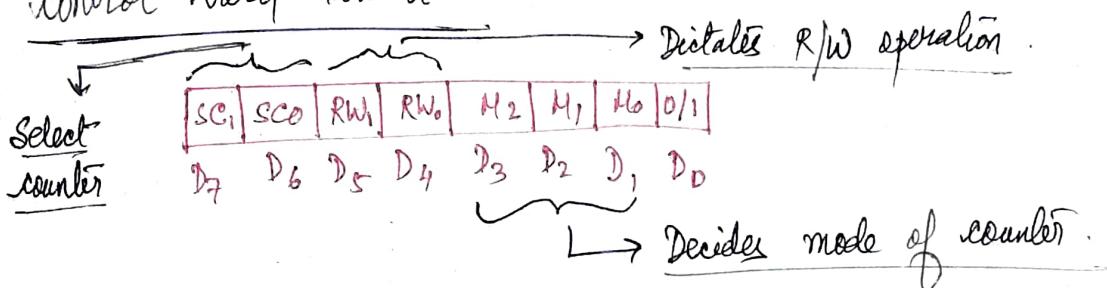


## Control logic

It has 5 signals -  $\bar{RD}$ ,  $\bar{WR}$ ,  $A_1$ ,  $A_0$  and chip select ( $\bar{CS}$ ).  
 The control word register and counters are selected according to signals  $A_1$  and  $A_0$ .

| $A_1$ , $A_0$ | Selection              |
|---------------|------------------------|
| 0, 0          | Counter 0              |
| 0, 1          | Counter 1              |
| 1, 0          | Counter 2              |
| 1, 1          | Control word register. |

## Control Word Format



$D_0 \rightarrow 0 \rightarrow$  gives binary count  
 $\rightarrow 1 \rightarrow$  gives BCD count.

| $D_5$           | $D_4$                  |
|-----------------|------------------------|
| RW <sub>1</sub> | RW <sub>0</sub>        |
| 0               | 0 → Counter batch      |
| 0               | 1 R/W LSB only command |
| 1               | 0 R/W MSB only         |
| 1               | 1 R/W LSB, then MSB.   |

| $D_3$          | $D_2$          | $D_1$          |
|----------------|----------------|----------------|
| M <sub>2</sub> | M <sub>1</sub> | M <sub>0</sub> |
| 0              | 0              | 0 → Mode 0     |
| 0              | 0              | 1 → Mode 1     |
| 0/X            | 1              | 0 → Mode 2     |
| 0/X            | 1              | 1 → Mode 3     |
| 1              | 0              | 0 → Mode 4     |
| 1              | 0              | 1 → Mode 5.    |

| $D_7$           | $D_6$                    |
|-----------------|--------------------------|
| SC <sub>1</sub> | SC <sub>0</sub>          |
| 0               | 0 → Counter 0            |
| 0               | 1 → Counter 1            |
| 1               | 0 → Counter 2            |
| 1               | 1 → Read back operation. |

## Modes of operation

1) Mode 0 - Interrupt / terminal count :- Initially output is low once a count is loaded, counter is decremented every cycle and when count reaches 0, output goes HIGH. It remains high until a new count or command word is loaded.

2) Mode 1 - Hardware Retriggerable one shot :-

OP is initially high, when gate is triggered, output goes low. At the end of count, output goes high again, thus generating a one-shot pulse.

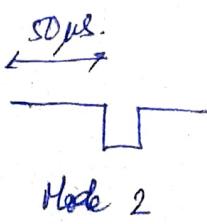
3) Mode 2 - Rate generator :-

Used to generate a pulse equal to the clock period, at a given interval. When a count is loaded, OUTPUT stays high, until count reaches 1 and then OUT goes low for 1 clock period. The count is reloaded automatically and pulse is generated continuously.

Q) Write instruction to generate a pulse every ~~50~~ <sup>50 μs</sup> from counter 0, considering 8254 is operating at 24Hz.

$$\text{Mode 2} \rightarrow \text{Time period} = \frac{1}{2} \times 10^{-6} = 0.5 \mu\text{s}$$

$$\text{Counter} = \frac{50 \times 10^{-6}}{0.5 \times 10^{-6}} = 100 \approx 64H.$$



| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|-------|-------|-------|-------|-------|-------|-------|-------|
| 0     | 0     | 0     | 1     | 0     | 1     | 0     | 0     |

14 H.

MVI A, 14H

// 83H control word  
register

OUT 83H

MVI A, 64H

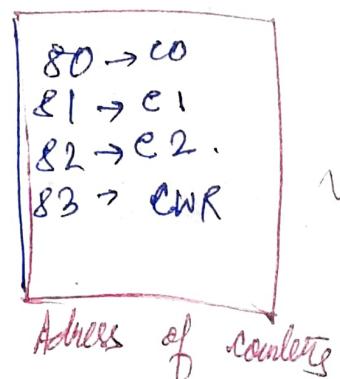
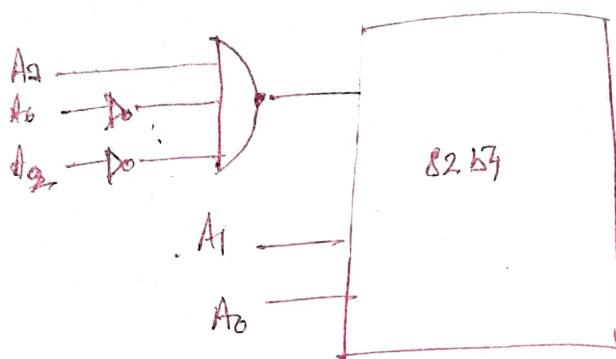
OUT 80H

HLT  $\hookrightarrow$  Address of counter D.

considering 8255 is peripheral mapped I/O.

4) Mode 3 - Square wave generation - In this mode, 8254 can be used as square wave generator. When the count N is loaded is even, then for half of the count the output remains high and for remaining half, it remains low. If the count loaded is odd, the first clock pulse decrements the count by 1, resulting in an even count value holding the output high. Then output remains high for half of the new count and goes low for remaining half. This procedure is repeated continuously, resulting in generation of square wave. If the loaded count value is odd, then for  $(N+1)/2$  pulses, output remains high and for  $(N-1)/2$  pulses, the output remains low.

Q) Let us suppose the interfacing circuit as shown below:-



Q) Which instruction of 8085 generates square wave generated from counter? Assume gate of counter 1 is tied to +5V. Explain why significance. 8259 is working at 2MHz.

~~X1 gate signal → Level triggered~~

Time period = 1ms.

8259 time period  $\rightarrow 0.5\mu s$ .

$$\therefore \text{Count} = \frac{10 \times 10^3}{0.5} = 2000 \\ = 07D0 \text{ H.}$$

$$\begin{array}{r} 16 \\ | \\ 2000 \rightarrow 0 \\ 16 \\ | \\ 125 \rightarrow 0 \\ 16 \\ | \\ 7 \rightarrow 7 \end{array}$$

Mode 3  $\rightarrow 011$

$$\begin{array}{r} 0111 \\ \hline 7 \end{array}$$

$$\begin{array}{r} 0110 \\ \hline 6 \end{array}$$

Address  $\rightarrow 01$

Control word  $\rightarrow 76H$

~~UVI~~ UVI A, 76H  
OUT 83H  
UVI A, 30H  
OUT 81H

UVI

~~A, 07H~~ A, 07H  
OUT 81H  
HLT

5) Mode 4 - Software Triggered Shibe - In this mode OUT is initially high, it goes low for one clock period at the end of the count. The count must be reloaded for subsequent outputs. The pulse can be used as a shibe routine interfacing the processor with other peripherals. Here, gate signal is level triggered.

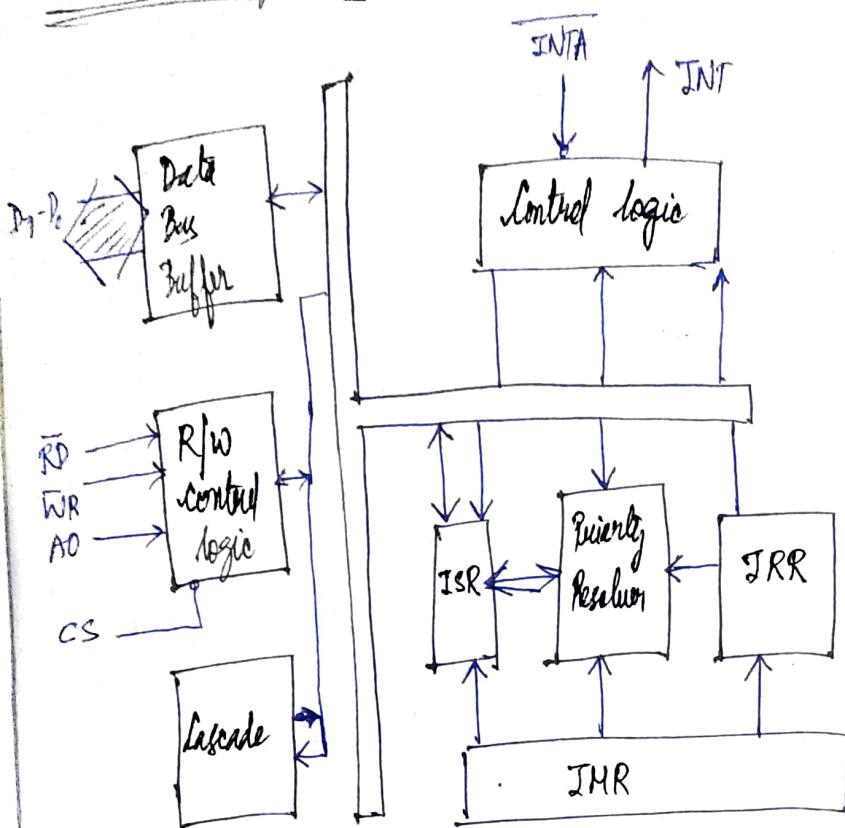
5) Mode 5 - Hardware Triggered Shibe - This mode is similar to mode 4 except that it is triggered by rising pulse of the gate. Initially output is ~~high~~, when gate pulse is triggered from low to high, the count begins. At the end of count, output goes low for one clock period.

## 8259 Programmable Interrupt Controller :- → 8085

8259A → 8086/8088

- ① 8259 Interrupt controller can manage 8 interrupts according to instructions written into its control register. This is equivalent to providing 8 interrupt pins on the processor in place of 1 INTR pin.
- ② Vector and interrupt request anywhere on the memory map.
- ③ Resolve 8 levels of interrupt priority in variety of modes.
- ④ Mask each interrupt individually.
- ⑤ Read the status of pending interrupt/in-service interrupt and masked interrupt.
- ⑥ It can be expanded using cascaded additional 8259.

## Block diagram :-



## R/W control logic :-

When AO is at logic 0, the controller is selected to write a command or read a status. The chip-select logic and AO determines the port address of the controller.

WR → A low on this pin enables 8259 to accept command from MPU.

RD → A low on this pin to enables 8259 to release status (pending, in-service, masked) on the data bus for the CPU.

## Cascade Buffer :-

It is used to expand the number of interrupt levels by cascading two or more 8259.

## • Interrupt register and priority resolver :-

⇒ In-service register (ISR) :- It keeps track of which interrupt inputs are currently being serviced. For each input that is being serviced, the corresponding bit of ISR will be set.

During the service of an interrupt request, if any other higher priority interrupt becomes active, it will be acknowledged and control will be transferred from lower priority ISS to higher priority ISS.

⇒ Interrupt Request Register (IRR) :- This stores all interrupt inputs that are requesting service. It is also a 8-bit register and has 8 input lines ( $IR_0 - IR_7$ ). Basically, it keeps track of which interrupt is asking for service. If an interrupt input is unmasked and it has interrupt request in it, then the corresponding bit of IRR will be set. The contents of this register give a list of pending interrupts.

⇒ Interrupt Mask Register (IMR) :- It is used to disable or enable individual interrupt request. It is also a 8-bit register and works on IRR. To unmask any interrupt, the corresponding bit is zero.

⇒ Priority Resolver:- This logic block determines the priority of the interrupt set in the IRR. It takes the information from IRR, IMR and ISR to determine whether the next interrupt request is having highest priority or not. If the interrupt is having highest priority that is being selected and processed. during interrupt acknowledge machine cycle. the corresponding bit of ISR will be set

### • Interrupt Operation

8259 requires two types of control words.

ICW - Initialization Control Word  
OCW - Operational Control Word.

There are 4 ICWs which indicate proper set-up conditions, like no of ICWs, vectorized addresses, cascade and single status etc.

OCWs set different mode of operation and perform functions such as masking interrupt, setting up states, read operations etc.

The sequence of events that occur even more than one interrupt request line goes high, are as follows:-

- i) SRR stores the request.
- ii) Priority resolver checks the three registers (IRR, IMR, ISR) and sets if resolving the priority, sets the INT high

when appropriate request is done.

- v) The MPV acknowledges the interrupt by sending INTA.
- vi) After the INTA is received, the appropriate priority pin in the Register ISR is set to indicate which interrupt level is being served and corresponding bit in the IRR is reset to indicate that the request is accepted.
- vii) Then the output for CALL instruction is placed on the data bus.
- viii) When the MPV decodes the CALL instruction it places two more INTA on the data bus.
- ix) During 2nd and 3rd INTA the CALL address is placed on the data bus.
- x) The program sequence is transferred to location specified by CALL instruction.

Types of priorities :-

- i) Fully Nested Mode:- several purpose mode in which all interrupt requests are arranged from highest to lowest bit. JRO is highest and JR7 is lowest one. In addition, any IR can be

assigned to highest priority in this mode. The priority sequence will thus begin at that IR.

### ii) Automatic Rotation mode:-

In this mode, device after being serviced receives the lowest priority.

iii) Specific Rotation mode:- Similar to automatic rotation mode except that user ~~can~~ can set at any IR for lowest priority.

### • End of interrupt (EOI) :-

- Non-specific EOI command:- It resets the highest priority ISR list.
- Specific EOI command:- It specifies which ISR bit is to be reset.
- Automatic EOI command:- In this mode, no command is necessary. During the INTA, ISR bit is reset. The major drawback of the mode is that ISR does not have information in which IR is being serviced. Thus any IR can interrupt the service routine irrespective of the priority if the interrupt flip flop is set.