

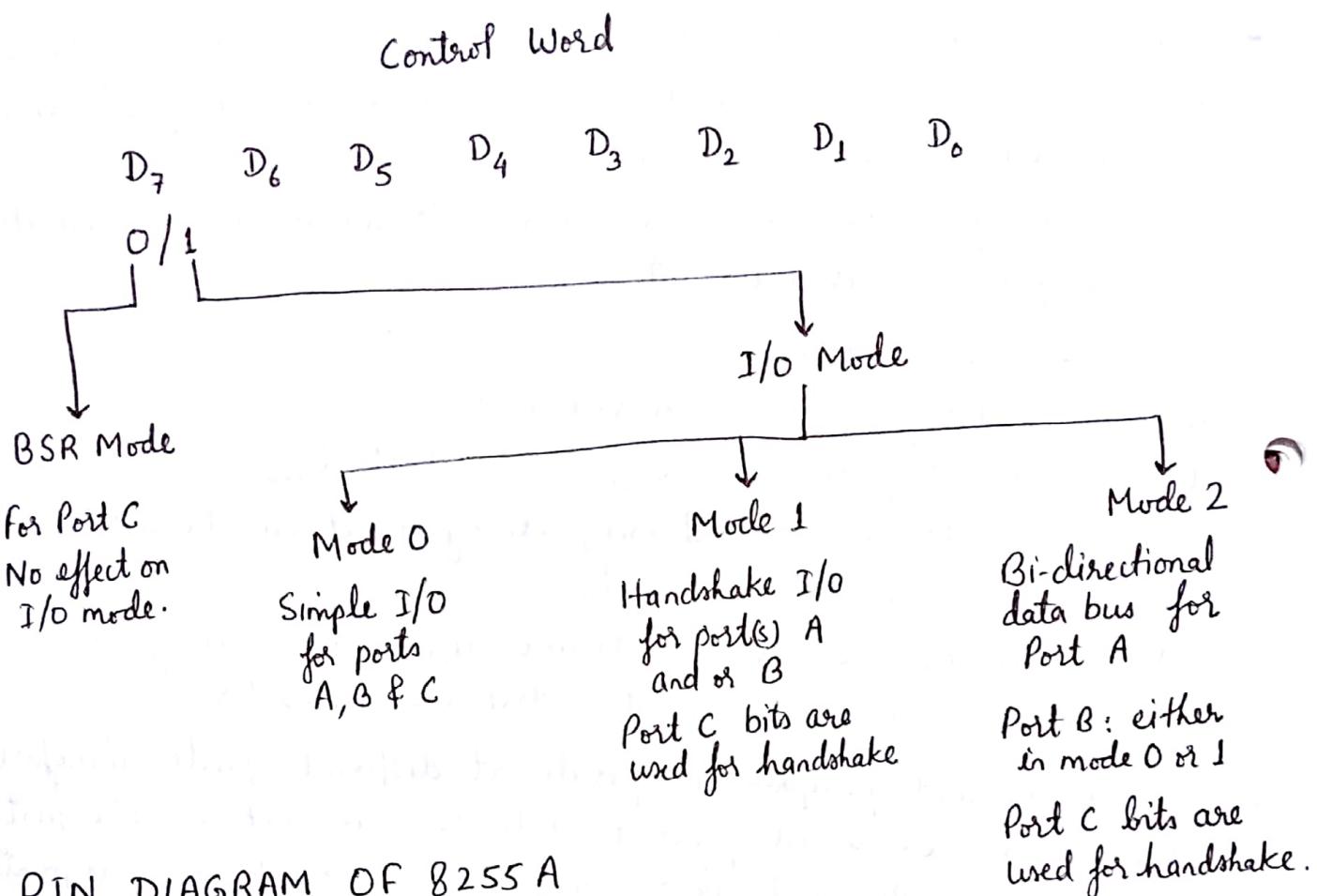
- The 8255A is a widely used, programmable, parallel I/O device. It consists of I/O ports which can be used for connecting I/O devices (peripherals) with the microprocessor.
- It can be programmed to transfer data under various conditions, from simple I/O to interrupt I/O.
- The 8255A consists of
  - a) Two 8-bit ports, Port A and Port B.
  - b) Two 4-bit ports, Port Cupper and Port Clever. These two 4-bit ports can be used independently or it can be used together as one 8-bit port.
  - c) The functions of these ports are defined by writing a control word in the 8-bit Control Word Register.

⇒ The MPU and peripherals operate at different speeds; therefore, signals are exchanged prior to data transfer between the fast responding MPU and slow-responding peripherals such as printers and data converters. These signals are called handshake signals.

The exchange of handshake signals prevents the MPU from waiting over the previous data before a peripheral has had a chance to accept it or from reading the same data before a peripheral has had a time to send the next data byte.

- When data transfer is performed using handshake signals then for Port A handshake signals are generated on Port Cupper pins, hence Port A and Port Cupper are called group A.
- Similarly, handshake signals for Port B are generated on Port Clever; so Port B and Port Clever together are called group B.
- All the functions of the 8255A can be classified according to two modes:
  - Bit Set/Reset (BSR) mode
  - I/O mode.

- The BSR mode is used to set or reset the bits in port C.
- The I/O mode is further divided into three modes: Mode 0, Mode 1 and Mode 2. The functions of 8255A are shown below



### PIN DIAGRAM OF 8255A

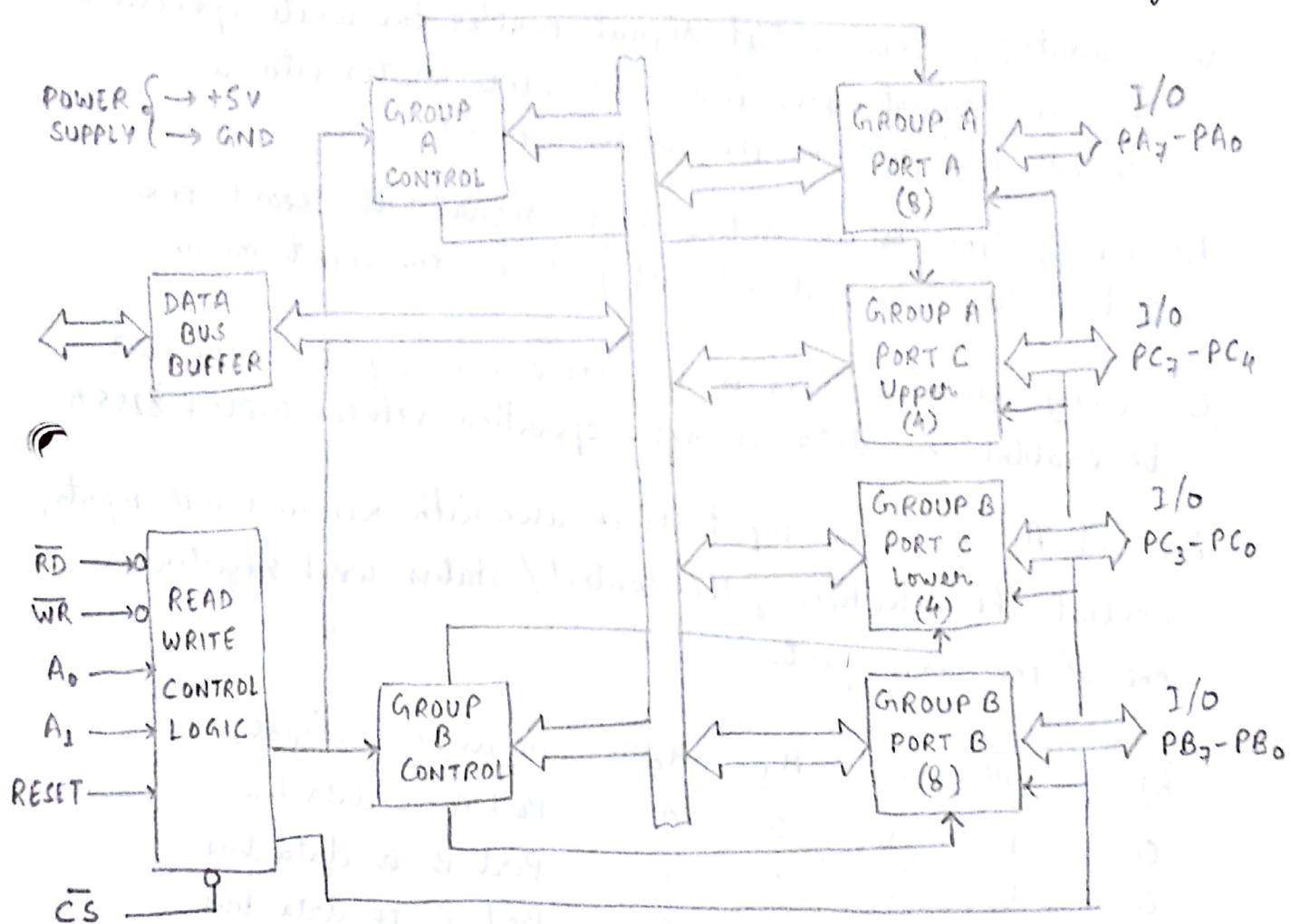
PA <sub>3</sub>	1	40	PA <sub>4</sub>
PA <sub>2</sub>	2	39	PA <sub>5</sub>
PA <sub>1</sub>	3	38	PA <sub>6</sub>
PA <sub>0</sub>	4	37	PA <sub>7</sub>
RD	5	36	WR
CS	6	35	RESET
GND	7	34	D <sub>0</sub>
A <sub>1</sub>	8	33	D <sub>1</sub>
A <sub>0</sub>	9	32	D <sub>2</sub>
PC <sub>7</sub>	10	31	D <sub>3</sub>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

#### Pin Names

D <sub>7</sub> - D <sub>0</sub>	Data bus bidirectional
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 (Bit)
PB <sub>7</sub> - PB <sub>0</sub>	Port B (Bit)
PC <sub>7</sub> - PC <sub>0</sub>	Port C (Bit)
V <sub>CC</sub>	+5 volts
GND	0 volts

## BLOCK DIAGRAM OF 8255A

The block diagram shows two 8-bit ports (A and B), two 4-bit ports ( $C_U$  and  $C_L$ ), the data bus buffer and control logic.



Data Bus Buffer - This 8-bit tri-state bi-directional buffer is used to interface the 8255 to the system data bus. Input or output instructions executed from CPU to the ports or control register and input data to the CPU from the ports or status register are all passed through the buffer.

Read/Write Control Logic :- The control logic block accepts control bus signals as well as inputs from the address bus, and issues commands to the individual group control blocks.

CONTROL LOGIC - The control section has six lines. Their functions and connections are as follows :-

RD (Read) : This control signal enables the read operation. When the signal is low, the MPU reads data from a selected I/O port of 8255 A.

WR (Write) : This control signal enables the write operation. When the signal goes low, the MPU writes into a selected I/O port or the control register.

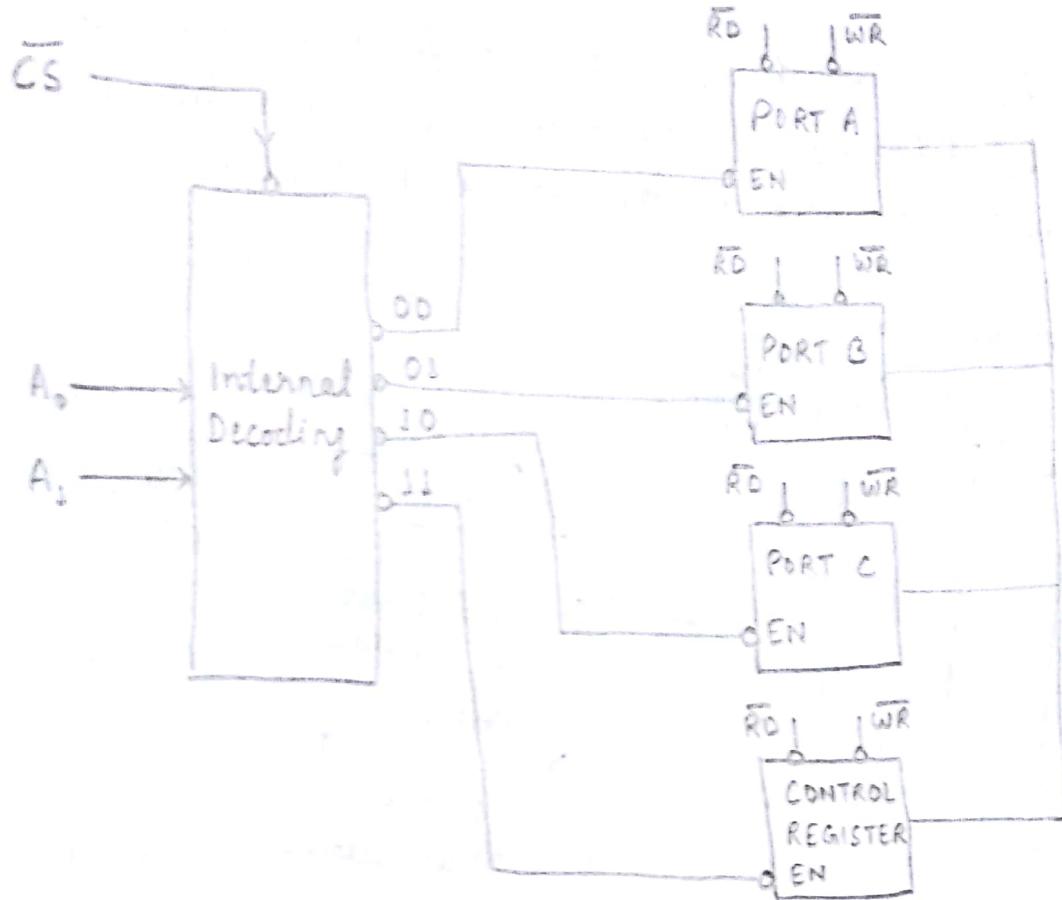
RESET : This is an active high signal; it clears the control register and sets all ports in the input mode.

CS (Chip Select) : This is an active low input which must be enabled for data transfer operation between MPU & 8255 A.

$A_0$  and  $A_1$  : - These input lines alongwith RD and WR inputs, control the selection of the control / status word register or one of the three ports.

<u>RD</u>	<u>WR</u>	<u>CS</u>	$A_1$	$A_0$	<u>Input (Read) Cycle</u>
0	1	0	0	0	Port A to data bus
0	1	0	0	1	Port B to data bus
0	1	0	1	0	Port C to data bus
0	1	0	1	1	CWR to data bus
<u>Output (Write) Cycle</u>					
1	0	0	0	0	Data bus to Port A
1	0	0	0	1	Data bus to Port B
1	0	0	1	0	Data bus to Port C
1	0	0	1	1	Data bus to CWR
X	X	1	X	X	8255 A is not selected

The address input  $A_0$  and  $A_1$  are used to select or access one of the three ports or the control word register. The figure shows an expanded version of internal structure.



Expanded version of Control Logic and I/O ports.

<u>CS</u>	<u><math>A_1</math></u>	<u><math>A_0</math></u>	<u>Selected</u>
0	0	0	Port A
0	0	1	Port B
0	1	0	Port C
0	1	1	Control Word Register
1	x	x	8255A is not selected

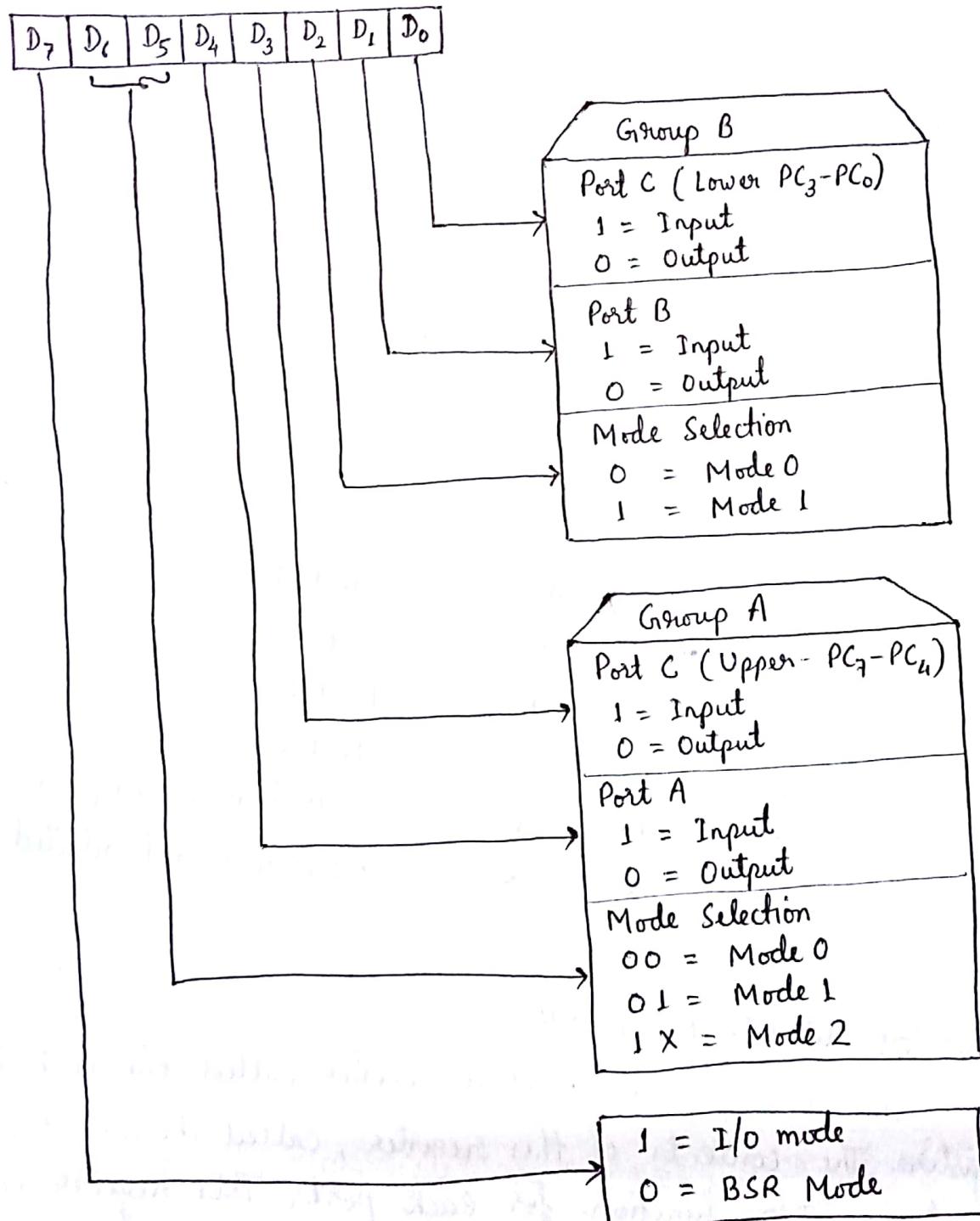
### CONTROL WORD REGISTER -

The above figure shows a register called the Control Register. The contents of this register, called the control word, specify an I/O function for each port. This register can be accessed to write a control word when  $A_0$  and  $A_1$  are at logic 1. The register is not accessible for a Read Operation.

Bit  $D_7$  of the control register specifies either the

the I/O function or the Bit Set/Reset function. If bit  $D_7 = 1$ , bits  $D_6 - D_0$  determine I/O functions in various modes. If bit  $D_7 = 0$ , port C operates in the Bit Set/Reset Mode.

### Control Word



8255A Control Word format for I/O mode

## MODES OF OPERATION OF 8255 PPI

The modes of operation of 8255 can be divided into two parts:-

- a) Parallel input/output mode
- b) BSR mode.

### Parallel I/O modes of 8255

If 8-bit data is to be transferred in parallel between I/O device and microprocessor, then 8255 will be used in parallel I/O modes. There are three types of Parallel input/output modes:

- 1. Mode 0
- 2. Mode 1
- 3. Mode 2.

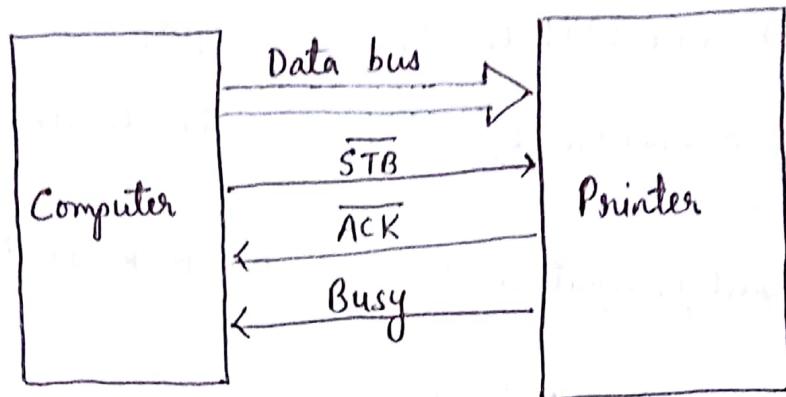
1) Mode 0 (Simple Input/Output mode): Mode 0 is used for simple input or output operations without handshaking. In this mode, ports A and B are used as two simple 8-bit I/O ports and port C as two 4-bit ports. Each port (or half-port, in case of C) can be programmed to function as simply an input port or an output port. The I/O features in mode-0 are as follows:-

- a) Outputs are latched.
- b) Inputs are buffered, not latched
- c) Ports do not have handshake or interrupt capability.

### 2) Mode 1 (Input/Output with handshake)

In this mode, input or output data transfer is controlled by handshaking signals. Handshaking signals are used to transfer data between devices whose data transfer speed are not same.

- The ports A and B function as 8-bit I/O ports. They can be configured either as input or output ports.
- Each port uses three lines from Port-C for handshake signals. The remaining two lines of port C can be used for simple I/O latched functions.
- Input and output data are latched.
- Interrupt logic is supported.



The handshaking signals are used to tell computer whether printer is ready to accept the data or not. If printer is ready to accept the data then after sending data on data bus, computer uses another handshaking signal  $\overline{STB}$  to tell printer that valid data is available on the data bus.

### Mode - 2 : Bidirectional I/O data transfer

In this mode, port A can be configured as the bidirectional port and port B either in Mode 0 or Mode 1. Port A uses five signals ( $PC_3 - PC_7$ ) from port C as handshake signals for data transfer. The remaining three signals from port C can be used either as simple I/O or as handshake for port B.

### Bit Set/Reset (BSR) Mode :-

The BSR mode is concerned only with the eight bits of port C, which can be set or reset by writing an appropriate control word in the control register. A control word with bit  $D_7 = 0$  is recognized as a BSR control word, and it does not alter any previously transmitted control word with bit  $D_7 = 0$ ; thus the I/O operations of ports A and B are not affected by a BSR control word. In the BSR mode, individual bits of port C can be used for applications such as an on/off switch.

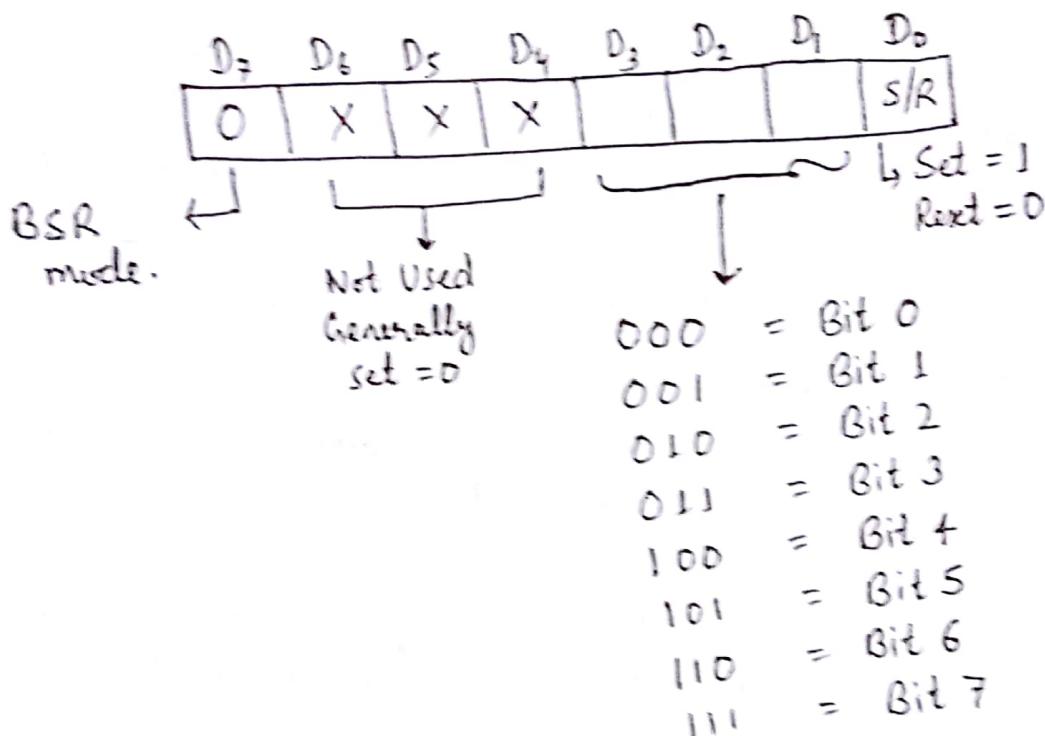
## BSR Control Word

2016-17

10

This control word, when written in the control register, sets or resets one bit at a time.

## Control Word format in the BSR Mode

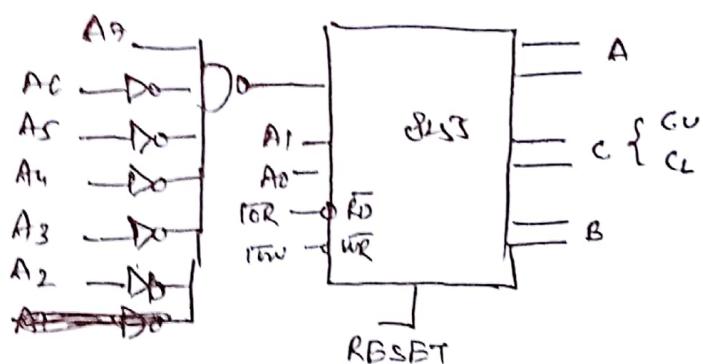


## UNIT-5

2015-16

8255 PPF

Prob.



Keyboard & 7 segment display

① Identify the port Address.

② Identify the mode 0 control word to configure A & C\_U as o/p & B and C\_L as inputs.

③ Write a program to read a DIP switches and display the reading from port B at port A, and from Port C\_L at Port C\_U.

① 80-A, 81-B, 82-C, 83-CR.

Ans. ② 83H

⊕

③ MVI A, 83H

OUT 83H

MOV IN 81H

OUT 80H

IN 82H

ANI 0FH

RLC

RLC

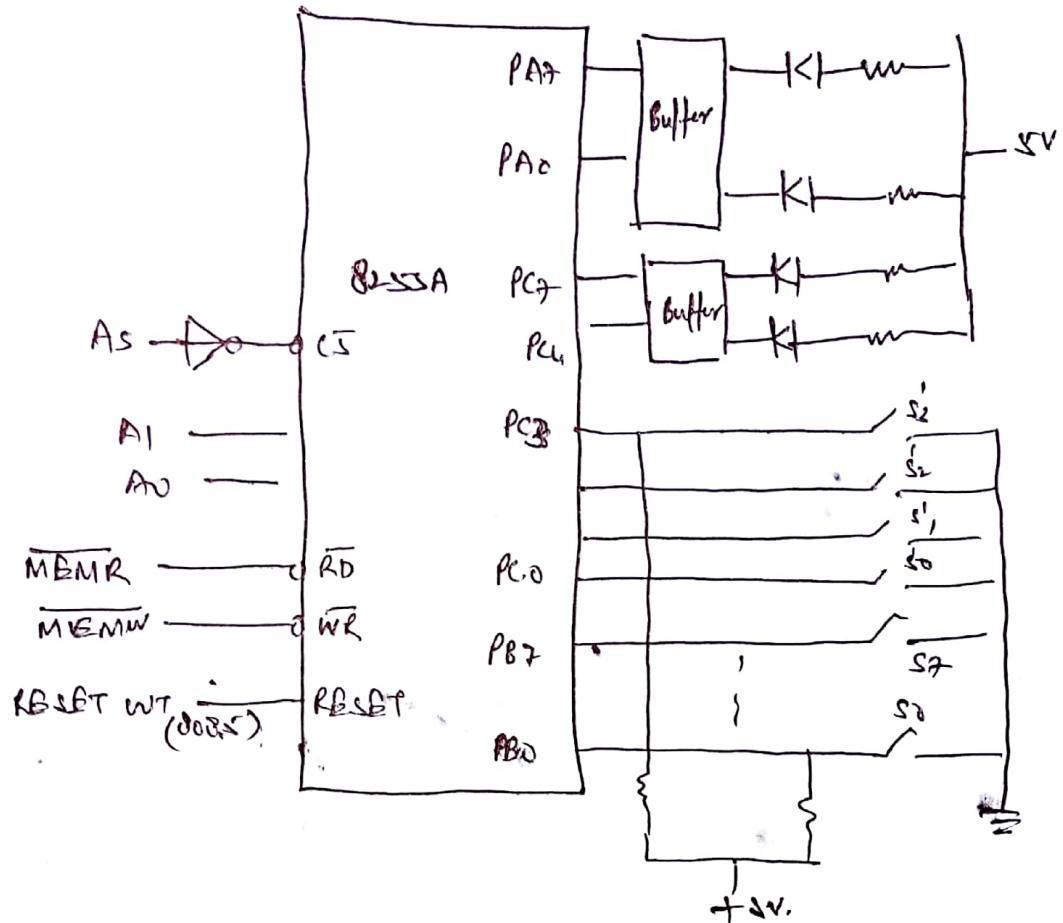
RLC

RLC

RLC  
OUT 82H

HLT.

Prob.



① Addr.

$$A = 8\pi m^2$$

B - 8001 H

C - 8002H

CR - 8003 H

$$\textcircled{2} \quad \text{cw} = 83H$$

③ MVI A, 83H

STA 8103H

IPA 8001 H

STA 91004

LDA 8103H

ANT 0611

R&C

BRU

八〇

KR C

STA 80024

HLT.

Prob. Write a BSR control word to set PC<sub>7</sub> and PC<sub>3</sub> & reset them after 10ms. Use fig. (Previous), and assume that a delay subroutine is available. 2016-17

Sol:

BSR control word -

To set PC<sub>7</sub> = 0000 1111 = 0FH

To reset PC<sub>7</sub> = 0000 1110 = 0EH

To set PC<sub>3</sub> = 0000 0111 = 07H

To reset PC<sub>3</sub> = 0000 0110 = 06H

BSR: MVI A, 0FH

OUT 83H

MVI A, 07H

OUT 83H

CALL DBLAY

MVI A, 06H

OUT 83H

MVI A, 0EH

OUT 83H

RET.

## 8254 (8253) PROGRAMMABLE INTERVAL TIMER

- The 8254 PIT is functionally similar to the software designed counters and timers. It generates accurate time delays and can be used for applications such as a real-time clock, an event counter, a digital one-shot, a square wave generator and a complex waveform generator.
- The 8254 includes three identical 16-bit counters that can operate independently in any of the six modes. It is packaged in 24-pin DIP and requires a single +5V power supply.
- To operate a counter, a 16-bit count is loaded in its register and, on command, begins to decrement the count until it reaches 0. At the end of the count, it generates a pulse that can be used to interrupt the MPU.
- The counter can count either in binary or BCD. In addition, a count can be read by the MPU while the counter is decrementing.

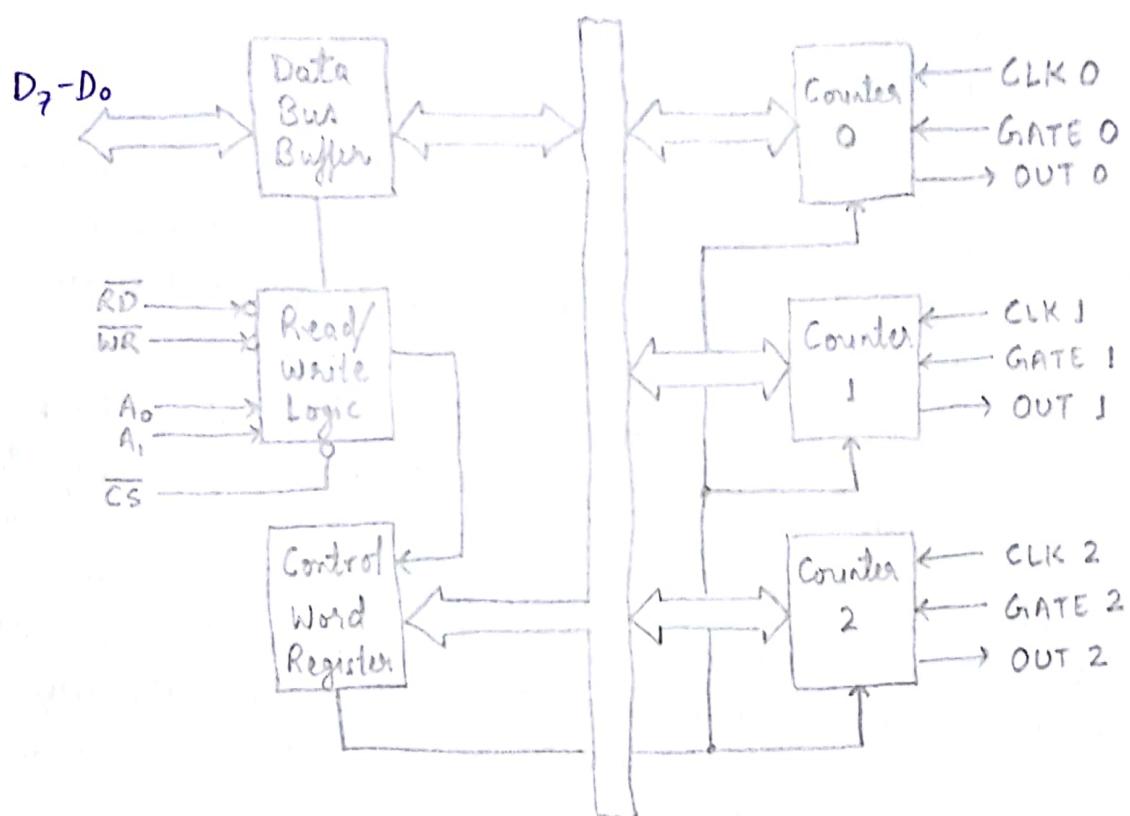
### PIN DIAGRAM OF 8254 PIT

8254		Pin Names	
D <sub>7</sub>	1	24	V <sub>cc</sub>
D <sub>6</sub>	2	23	WR
D <sub>5</sub>	3	22	RD
D <sub>4</sub>	4	21	CS
D <sub>3</sub>	5	20	A <sub>1</sub>
D <sub>2</sub>	6	19	A <sub>0</sub>
D <sub>1</sub>	7	18	CLK 2
D <sub>0</sub>	8	17	OUT 2
CLK 0	9	16	GATE 2
OUT 0	10	15	CLK 1
GATE 0	11	14	GATE 1
GND	12	13	OUT 1
		D <sub>7</sub> - D <sub>0</sub>	Data bus (8-bit)
		CLK N	Counter clock Inputs
		GATE N	Counter Gate Inputs
		OUT N	Counter Outputs
		RD	Read Counter
		WR	Write Command or Data
		CS	Chip Select
		A <sub>0</sub> - A <sub>1</sub>	Counter Select
		V <sub>cc</sub>	+ 5 Volts
		GND	Ground

## Block Diagram of 8254

2016-17, 2019-20

33



Data Bus Buffer : It is a tristate, 8-bit, bidirectional buffer which is interfaced with the 8-bit data of microprocessor. It transmits the data to the CPU and receives the data from the CPU when IN and OUT instructions are executed respectively (in case of I/O mapped I/O mode). It does the following functions :

- Programming the 8254 in various modes
- Loading the count registers
- Reading the count values.

Read/Write Control logic :- This section has five signals :  $\overline{RD}$ ,  $\overline{WR}$ ,  $\overline{CS}$ ,  $A_1$  &  $A_0$  (Address lines). In the peripheral I/O mode, the  $\overline{RD}$  and  $\overline{WR}$  signals are connected to  $\overline{IOR}$  and  $\overline{IOW}$  respectively. In memory-mapped I/O, these are connected to  $\overline{MEMR}$  and  $\overline{MEMW}$ . Address lines  $A_0$  and  $A_1$  of the MPU are usually connected to lines  $A_0$  &  $A_1$  of the 8254 and  $\overline{CS}$  is tied to the decoded address. The control word register and counters are selected according to the signals on lines  $A_0$  and  $A_1$ .

$A_1$	$A_0$	Selection
0	0	Counter-0
0	1	Counter-1
1	0	Counter-2
1	1	Control Word Register

Control Word Register - This register is accessed when lines  $A_0$  and  $A_1$  are at logic 1. It is used to write a command word which specifies the counter to be used (binary or BCD), its mode, and either a Read or a Write operation.

Counters - The 8254 includes three counters (0, 1 and 2). Each counter has two input signals CLK and GATE and one output signal OUT. Each counter to consist of a single 16-bit presetable, down counter. The counter can operate in either binary or BCD as its input, gate and output are configured by the selection of modes stored in the control word register. The counters are fully independent. The programmer can read the contents of any of three counters without disturbing the actual count in progress.

### 8254 Control Word Register format

$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

#### SC - Select Counter:

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Commands

#### M - MODE:

M2	M1	M0	
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

#### RW - Read/Write:

RW1	RW0	
0	0	Counter Latch Command
0	1	Read/Write LSB only
1	0	Read/Write MSB only
1	1	Read/Write LSB first, then MSB.

#### BCD:

0	Binary Counter 16-bits
1	BCD Counter (4 Decades)

## Gate Settings of a Counter

Modes \ Signal Status	Low or Going low	Rising	High
0	Disables Counting	-	Enables counting
1	-	(1) Initiates counting (2) Resets output after next clock	-
2	(1) Disables counting (2) Sets output immediately high	(1) Reloads counter (2) Initiates counting	Enables counting
3.	(1) Disables counting (2) Sets output immediately high	Initiates counting	Enables counting
4.	Disables counting	-	Enables counting
5.	-	Initiates counting	-

## Programming the 8254

- The 8254 can be programmed to provide various types of output through Write operations ; or to check a count while counting through Read operations.

### Write Operations

The programming procedure for 8254 is very flexible. Only two conventions need to be remembered :

1. For each counter, the control word must be written before the initial count is written.

2. The initial count must follow the count format specified in the Control word ( least significant byte only , most significant byte only , or least significant byte and then most significant byte ).

Since, the control word register and the three counters have separate addresses ( selected by the  $A_1, A_0$  inputs ), and each control word specifies the counter it applies to (  $SC_0, SC_1$  bits ); no special instruction sequence is required. With a clock and an appropriate gate signal to one of

the counters, the above steps should start the counter and provide appropriate output according to the control word.

### Read Operations

In some applications, especially in event counters, it is necessary to read the value of the count in progress. This can be done by either of two methods. One method involves reading a count after inhibiting (stopping) the counter to be read. The second method involves reading a count while the count is in progress (known as reading on the fly).

In the first method, counting is stopped by controlling the gate input or the clock input of the selected counter, and two I/O read operations are performed by the MPU. The first I/O operation reads the lower order byte and the second I/O operation reads the high-order byte.

In the second method, an appropriate control word is written into the control register to latch a count in the output latch, and two I/O read operations are performed by the MPU.

### Read-Back Command

The Read-Back command in the 8254 allows the user to read the count and the status of the counter; this command is written in the control register, and the count of the specified counter(s) can be latched if COUNT (bit  $D_5$ ) is 0. The command applies to the counters selected by setting their corresponding bits  $D_3, D_2, D_1 = 1$ .

### Read-Back Command format

$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$
1	1	COUNT	STATUS	CNT 2	CNT 1	CNT 0	0

$D_5 : 0$  = Latch Count of Selected Counter(s)

$D_4 : 0$  = Latch Status of Selected Counter(s)

$D_3 : 1$  = Select Counter 2

$D_2 : 1$  = Select Counter 1

$D_1 : 1$  = Select Counter 0

$D_0$  : Reserved for future expansion; must be 0.

$A_0, A_1 = 11$

$\overline{CS} = 0$

$\overline{RD} = 1$

$\overline{WR} = 0$

- The Read-back command may be used to latch multiple counter output latches by setting the COUNT bit  $D_5=0$  and selecting the desired counters.
- The status of the counter(s) can be read if STATUS bit ( $D_4$ ) of the Read-back command is low. The format of status byte is shown below-

$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$
OUTPUT	NULL COUNT	RW1	RW0	M2	M1	M0	BCD

$D_7 : 1$  = Out Pin is 1  
 $: 0$  = Out Pin is 0

$D_6 : 1$  = Null Count  
 $: 0$  = Count available for reading

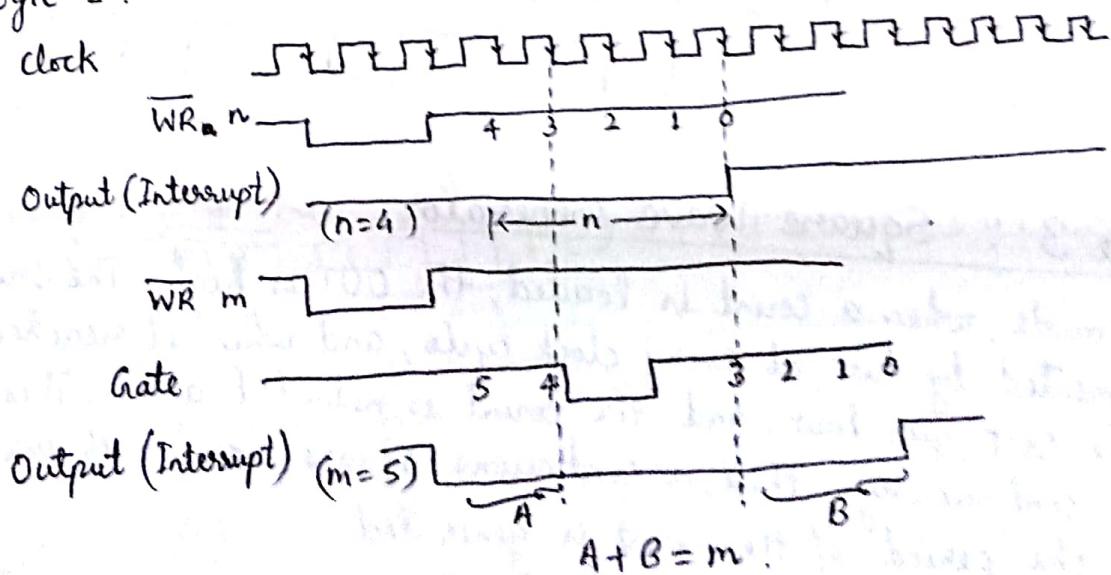
$D_5 - D_0$  : Counter Programmed mode.

## MODES OF 8254 :- 2012-13

Programmable Interval Timer operates in six modes.

### ① MODE 0 : Interrupt on Terminal Count

In this mode, initially the OUT is low. Once a count is loaded in the register, the counter is decremented every cycle, and when the count reaches zero, the OUT goes high. This can be used as an interrupt. The OUT remains high until a new count or a command word is loaded. The fig. shows that the counting ( $m=5$ ) is temporarily stopped when the Gate is disabled ( $G=0$ ), and continued again when the Gate is at logic 1.



## II Mode 1 : Hardware-Retriggerable One-Shot

In this mode, the OUT is initially high. When the Gate is triggered, the OUT goes low, and at the end of the count, the OUT goes high, thus generating a one shot pulse.

clock.

$\overline{WR} \ n$

Trigger   
Output   
 $n=4$  4 3 2 1 0

Trigger   
Output   
 $n=10$  4 3 2 4 3 2 1 0

## III Mode 2 : Rate Generator Clock

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

clock

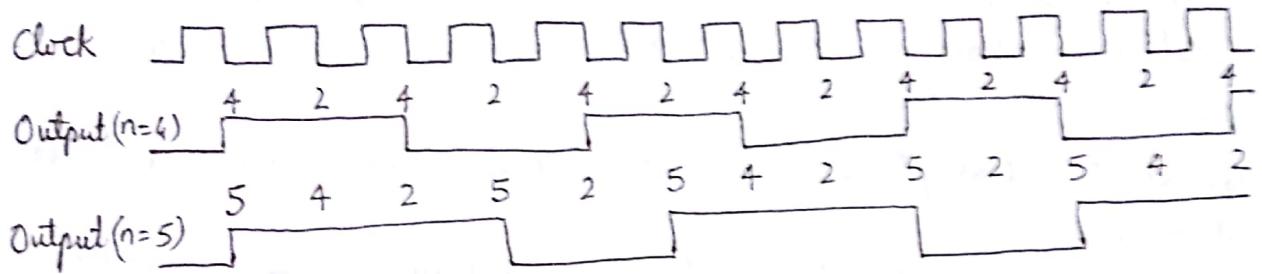
$\overline{WR} \ n$    
 $n=4$  4 3 2 1 0(4)   
 $n=3$  3 2 1 0(3)   
Output

## IV Mode 3 : Square Wave Generator with program

In this mode, when a count is loaded, the OUT is high. The count is decremented by two at every clock cycle, and when it reaches zero, the OUT goes low, and the count is reloaded again. This is repeated continuously; thus, a continuous square wave with period equal to the period of the count is generated.

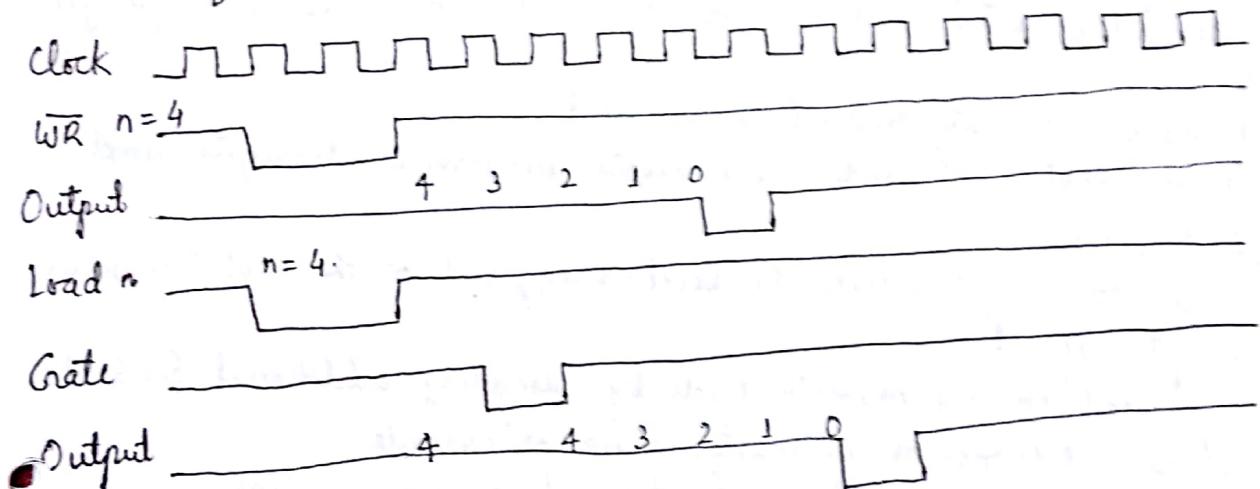
~~2016-17~~  
To generate square wave of 1 kHz

In other words, the frequency of the square wave is equal to the frequency of the clock divided by the count. If the count ( $N$ ) is odd, the pulse stays high for  $(N+1)/2$  clock cycles and stays low for  $(N-1)/2$  clock cycles.



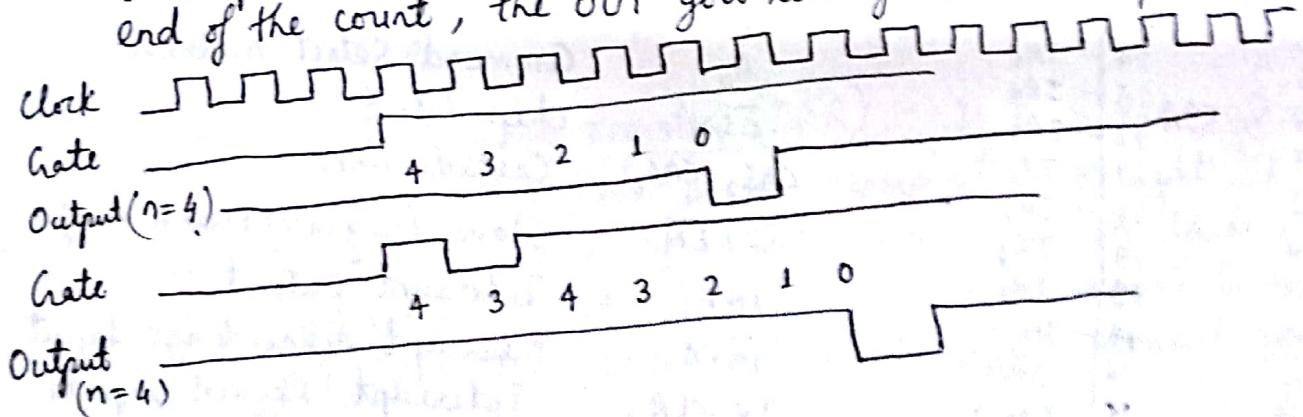
#### IV Mode 4: Software Triggered Strobe

In this mode, the 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.



#### V Mode 5: Hardware-Triggered Strobe

This mode is similar to Mode 4, except that it is triggered by the rising pulse at the gate. Initially, the OUT is low, and when the Gate pulse is triggered from low to high, the count begins. At the end of the count, the OUT goes low for one clock period.



## 7.1(b). Square wave generator

(17)

2015-16

IN 8254 can be operate in six different mode. In mode-3

IN 8254 can be used as square wave generator.

In this mode, when a count is loaded, the out is high.

The count is decremented by two at every clock cycle,

and when it reaches zero, the out goes low, and

the count is reloaded again. This is repeated continuously

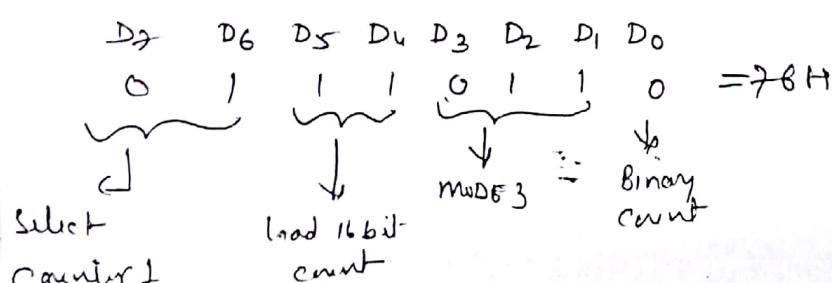
thus, a continuous square wave with period equal to  
the period of the count is generated.

Ex. Suppose we want to generate a 1 KHz square wave  
from counter 1. Assuming the gate of the counter 1  
is tied to +5V through 10k resistor. 2015-16

So, to generate square wave it should be initialize in mode-3.

Control word -

$$\left( \frac{2^{M+1}}{1 \text{ KHz}} = 2000 \right)$$



$$\text{Count} = \frac{1 \times 10^3}{0.5 \times 10^6} = 2000$$
$$= 0.5 \times 10^6 = 02D0H$$

(for N  $\rightarrow$   $N/2 \rightarrow$  upper half  
 $N/2 \rightarrow$  lower half).

Square: MVI A, 0111 0110B

OUT 83H

MVI A, D0H

OUT 81H

MVI A, 07H

OUT 81H

HLT

THE END

2-13/4

## THE 8259A PROGRAMMABLE INTERRUPT CONTROLLER

The 8259A is a programmable interrupt controller designed to work with Intel microprocessors 8085, 8086 and 8088. The 8259A interrupt controller can

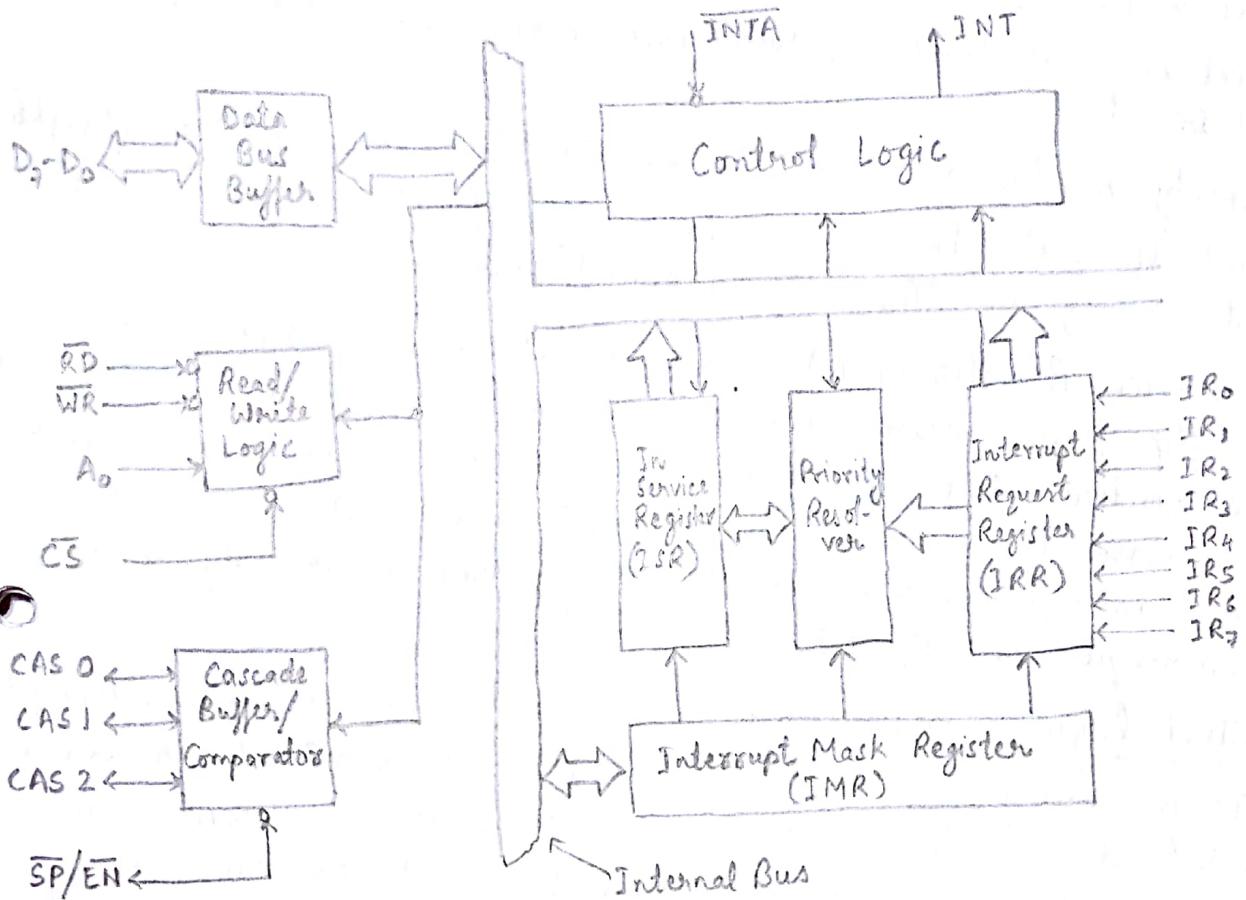
1. manage eight interrupts according to the instructions written into its control registers.
2. vector an interrupt request anywhere in the memory map. However, all eight interrupts are spaced at the interval of either four or eight locations. This eliminates the major drawback of the 8085 interrupts in which all interrupts are vectored to memory locations on page 00H.
3. generate eight levels of interrupt priorities in a variety of modes, such as fully nested mode, automatic rotation mode and specific rotation mode.
4. mark each interrupt request individually.
5. read the status of pending interrupts, in-service interrupts, and masked interrupts.
6. be set up to accept either the level-triggered or the edge-triggered interrupt request.
7. be expanded to 64 priority levels by cascading additional 8259A's, i.e. it can be configured in master slave or "cascade".
8. provides the starting address of interrupt service routine.

### PIN DIAGRAM OF 8259A

CS	1	28	Vcc
WR	2	27	A <sub>0</sub>
RD	3	26	INTA
D <sub>7</sub>	4	25	IR <sub>7</sub>
D <sub>6</sub>	5	24	IR <sub>6</sub>
D <sub>5</sub>	6	23	IR <sub>5</sub>
D <sub>4</sub>	7	22	IR <sub>4</sub>
D <sub>3</sub>	8	21	IR <sub>3</sub>
D <sub>2</sub>	9	20	IR <sub>2</sub>
D <sub>1</sub>	10	19	IR <sub>1</sub>
D <sub>0</sub>	11	18	IR <sub>0</sub>
CAS 0	12	17	INT
CAS 1	13	16	SP/EN
GND	14	15	CAS 2

Pin Names	
D <sub>7</sub> -D <sub>0</sub>	Data Bus (Bidirectional)
RD	Read Input
WR	Write Input
A <sub>0</sub>	Command Select Address
CS	Chip Select
CAS <sub>2</sub> -CAS <sub>0</sub>	Cascade lines
SP/EN	Slave Program/Enable Buffer
INT	Interrupt Output
INTA	Interrupt Acknowledge Input
IR <sub>0</sub> -IR <sub>7</sub>	Interrupt Request Inputs

# Block Diagram of 8259A



Priority Interrupt Controller 8259A can be divided into four main sections:

1. Interrupt and Control Logic Section
2. Data bus buffer
3. Read/ Write Control logic Section
4. Cascade buffer/comparator

## 1. Interrupt and Control Logic Section

(i) Interrupt request - 8259A has 8 different Interrupt request levels i.e. IR<sub>0</sub> to IR<sub>7</sub>. Eight different interrupting devices can be connected to these eight IR levels. These eight IR levels can be defined either as edge triggered or level triggered. The interrupting device will give interrupt signal to 8259A through these eight IR levels.

(ii) Interrupt Request Register (IRR) - It is 8-bit register which contains eight flip-flops  $D_0 - D_7$  corresponding to eight IR level 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. If interrupt signal is not received on any IR pin, then corresponding bit of IRR remains zero.

- (iii) Interrupt Mask Register - (IMR) - This is 8-bit register which contains 8 flip-flops corresponding to eight IR pin  $IR_0$  to  $IR_7$ . If any IR level is to be disabled (Mask) then corresponding bit of IMR should be set.
- (iv) Priority Resolver (PR) - This register 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 given lowest priority.
- (v) In-Service Register (ISR) - It is 8-bit register which contains eight flip-flops corresponding to eight IR levels  $IR_0$  to  $IR_7$ . Interrupt service routine stores the information regarding the interrupts which microprocessor is servicing, i.e., interrupts whose ISR (Interrupt Service routine) microprocessor is executing. e.g. If MPU is currently servicing  $IR_2$  interrupt then  $D_2 = 1$ , otherwise  $D_2 = 0$ .
- (vi) Control logic - Control logic block has two pins as INT (Interrupt) and INTA (interrupt acknowledge). The INT is an output pin and is connected directly to the INTR pin of 8085 MP. The INTA is an input pin and it is connected to INTA pin of 8085 MP. After receiving its interrupt request priorities, 8259A gives interrupt request to the INTR pin of 8085. After receiving the signal at INTA from MP 8085, 8259 places the opcode for CALL instruction on the systems data bus. On further receiving two additional INTA it gives address of the Interrupt service routine (ISR) on to the system data bus.

2. Data Bus Buffer - This 8-bit bidirectional buffer is used to interface the 8259 to the system data bus (8085).
3. Read/Write Control Logic - This section has Initialization command word registers (ICW registers) and the Operation Command registers (OCW). The pins related to this section are -
- CS (chip Select) - A low on this input enables the 8259. No writing or reading operation can be done unless the device is enabled.
  - WR - A low on this pin input enables the CPU to write to control words to the 8259.
  - RD - A low on this pin enables the 8259 to send status of IRR, ISR and IMR on data bus.

(iv)  $A_0$  - When the address line  $A_0$  is at logic 0, the controller is selected to write a command or read a status. The chip select logic and  $A_0$  determine the port address of the controller.

4. Cascade Buffer/Comparator - This section has three cascade signals  $CAS_0 - CAS_2$  and  $\overline{SP}/\overline{EN}$  (Slave Program/Enable buffer). This section is used to send signals from master output to slaves when multiple devices are cascaded.

### Interrupt Operation :-

- To implement interrupts, the Interrupt Enable flip-flop in the MP should be enabled by writing the EI instruction, and the B259A should be initialised by writing control words in the control register.
- The B259A requires two types of control words: Initialization Command Words (ICWs) and Operational Command Words (OCWs). The ICWs are used to set up the proper conditions and specify RST vector addresses.
- The OCWs are used to perform functions such as masking interrupts, setting up status-read operations etc.

After the B259A is initialized, the following sequence of events occur when one or more interrupt request lines go high:

1. The IRR stores the requests.
2. The priority resolver checks three registers; the IRR for interrupt requests, the IMR for masking bits, and the ISR for the interrupt request being served. It resolves the priority and sets the INT high when appropriate.
3. The MPU acknowledges the interrupt by sending INTA.
4. After INTA is received, the appropriate priority bit in the ISR is set to indicate which interrupt level is being served, and the corresponding bit in the IRR is reset to indicate that the request is accepted. Then the opcode for the CALL instruction is placed on the data bus.
5. When the MPU decodes the CALL instruction, it places two more INTA signals on the data bus.

6. When the 8259A receives the second INTA, it places the low-order byte of the CALL address on the data bus. At the third INTA, it places the high-order byte on the data bus. The CALL address is the vector memory location for the interrupt; this address is placed in the control register during initialization.
  7. During the third INTA pulse, the ISR bit is reset either automatically (Automatic-End-of-Interrupt - AEOI) or by a command word that must be issued at the end of the service routine (End-of-Interrupt - EOI). This option is determined by the initialization command word (ICW).
  8. The program sequence is transferred to the memory location specified by the CALL instruction.

## Priority Modes and Other features

Many types of priority modes are available under software control in the 8259A, and they can be changed dynamically during the program by writing appropriate command words. Commonly used priority modes are discussed below:-

## 1. Fully Nested Mode -

This is a general purpose mode in which all IRs (Interrupt Requests) are arranged from highest to lowest, with  $IR_0$  as the highest and  $IR_7$  as the lowest.

In addition, any IR can be assigned the highest priority in this mode; the priority sequence will then begin at that IR. In the example below,  $IR_4$  has the highest priority, and  $IR_3$  has the lowest priority.

$IR_0$	$IR_1$	$IR_2$	$IR_3$	$IR_4$	$IR_5$	$IR_6$	$IR_7$
4	5	6	7	0	1	2	3
↑ lowest priority				↑ Highest priority			

2. Automatic Rotation Mode - In this mode, a device, after being serviced, receives the lowest priority. Assuming that the  $IR_2$  has just been serviced, it will receive the seventh priority.

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

3. Specific Rotation Mode - This mode is similar to the automatic rotation mode, except that the user can select any IR for the lowest priority, thus fixing all other priorities.

### End of Interrupt

After the completion of an interrupt service, the corresponding ISR bit needs to be reset to update the information in ISR. This is called the End-of-Interrupt (EOI) command. It can be issued in three formats:

1. Non-specific EOI Command - When this command is sent to the 8259A, it resets the highest priority ISR bit.
2. Specific EOI command - This command specifies which ISR bit to reset.
3. Automatic EOI - In this mode, no command is necessary. During the third INTA, the ISR bit is reset. The major drawback with this mode is that the ISR does not have information on which IR is being serviced. Thus any IR can interrupt the service routine, irrespective of its priority, if the Interrupt Enable flip-flop is set.

2013-14  
2015-16  
2014-15

## DIRECT MEMORY ACCESS (DMA) AND THE

### 8237 DMA CONTROLLER

- Direct memory access is an I/O technique commonly used for high-speed data transfer; for example, data transfer between system memory and a floppy disk. In status check I/O and interrupt I/O, data transfer is relatively slow because each instruction needs to be fetched and executed.
- In DMA, the MPU releases the control of the buses to a device called a DMA controller. The controller manages data transfer between memory and a peripheral under its control, thus bypassing the MPU. Conceptually, this is an important I/O technique; it introduces two new signals available on the 8085 - HOLD & HLDA.

1) HOLD - This is an active high input signal to the 8085 from another master requesting the use of the address and data buses. After receiving the Hold request, the MPU relinquishes the buses in the following machine cycle. All buses are tri-stated and the Hold Acknowledge (HLDA) signal is sent out. The MPU regains control of the buses after HOLD goes low.

2) HLDA (Hold Acknowledge) - This is an active high output signal indicating that the MPU is relinquishing control of the buses.

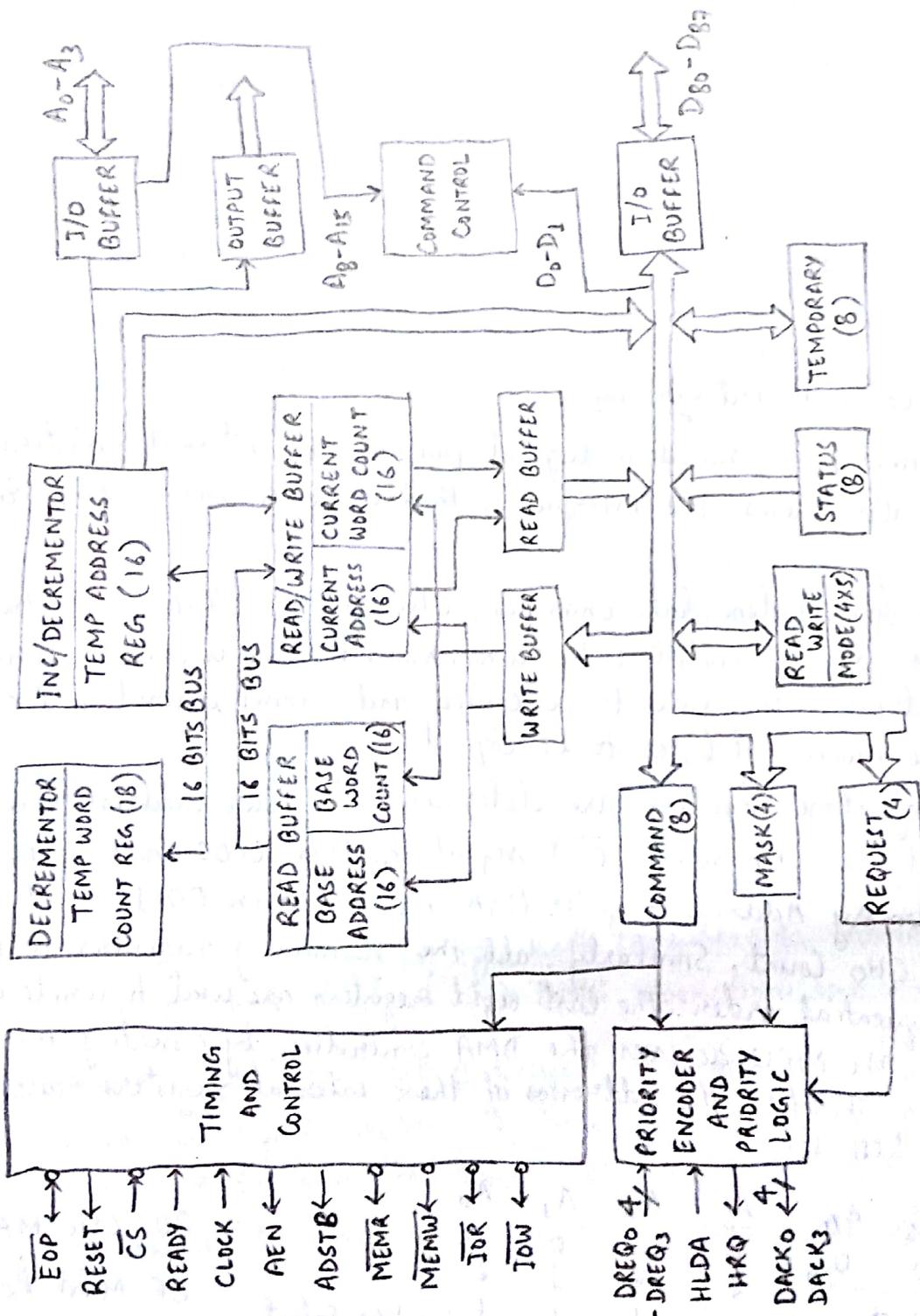
A DMA controller uses these signals as if it were a peripheral requesting the MPU for the control of the buses. The MPU communicates with the controller by using the Chip Select line, buses and control signals. However, once the controller has gained control, it plays the role of a processor for data transfer. To perform this function the DMA controller should have

- a data bus
- an address bus
- Read/Write control signals
- control signal to disable its role as a peripheral and to enable its role as a processor.

This process is called switching from the slave mode to the master mode.

## The 8237 DMA Controller

- The 8237 is a programmable Direct Memory Access controller housed in a 40-pin package. It has four independent channels with each channel capable of transferring 64 K bytes.
- It interfaces with two types of devices: the MPU and peripherals such as floppy disks. The DMA plays two roles in a given system: It is an I/O to the microprocessor (slave mode) and it is a data transfer processor to peripherals such as floppy disks (master mode).



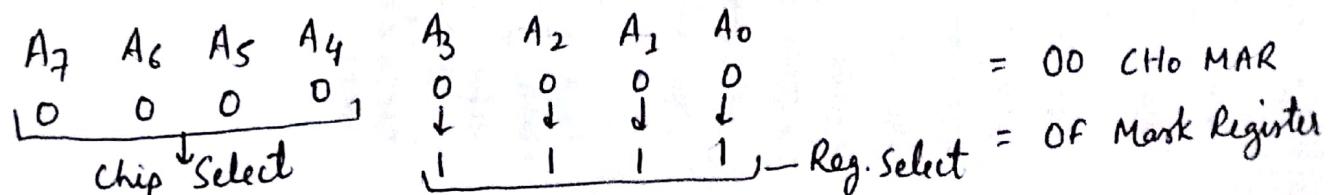
BLOCK DIAGRAM OF 8237

IOR	1	40	- A <sub>7</sub>
IOW	2	39	- A <sub>6</sub>
MEMR	3	38	- A <sub>5</sub>
MEMW	4	37	- A <sub>4</sub>
PIN5	5	36	- EOP
READY	6	35	- A <sub>3</sub>
HLDA	7	34	- A <sub>2</sub>
A0STB	8	33	- A <sub>1</sub>
AEN	9	32	- A <sub>0</sub>
HRQ	10	31	- V <sub>cc</sub> (+5V)
CS	11	30	- DB0
CLK	12	29	- DB1
RESET	13	28	- DB2
DACK2	14	27	- DB3
DACK3	15	26	- DB4
DRQ3	16	25	- DACK0
DRQ2	17	24	- DACK1
DRQ1	18	23	- DB5
DRQ0	19	22	- DB6
(GND) V <sub>SS</sub>	20	21	- DB7

PIN DIAGRAM  
OF 8237

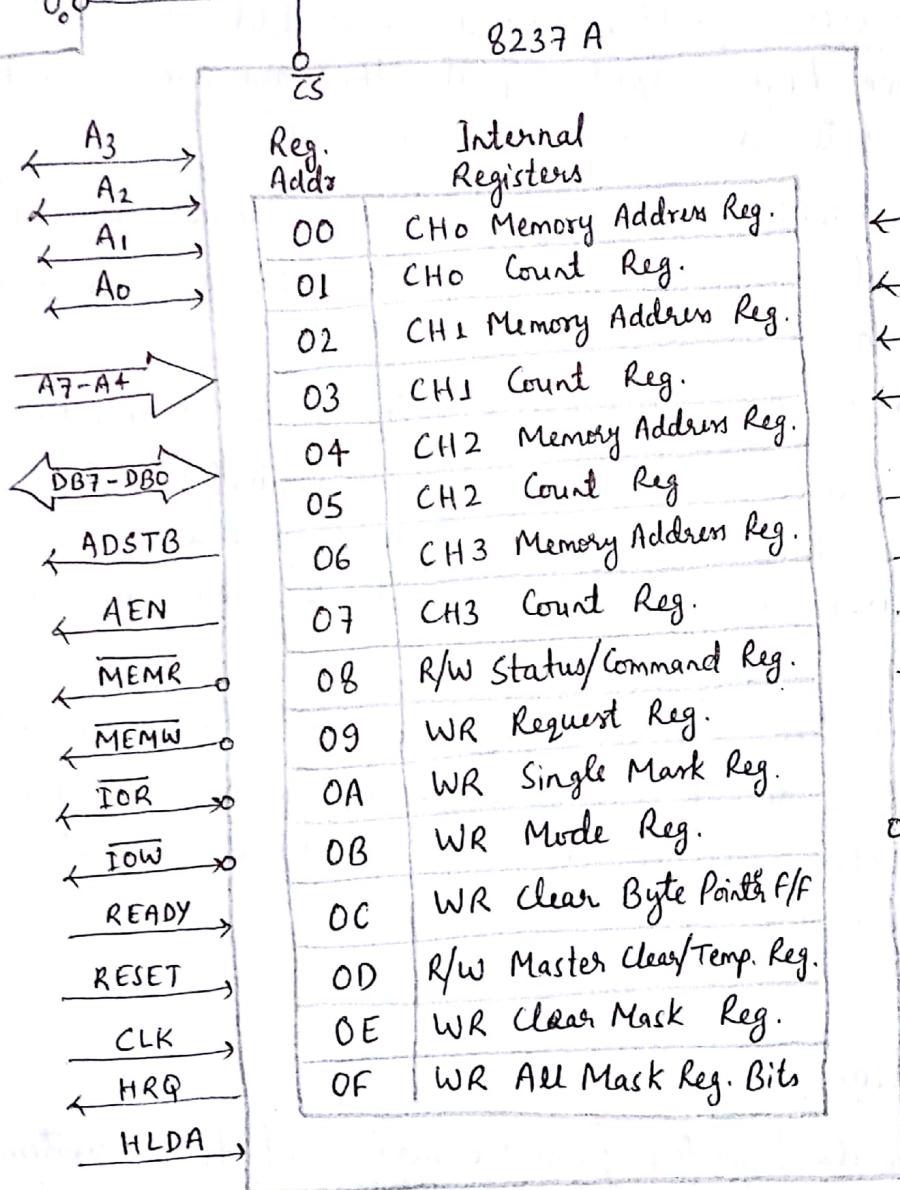
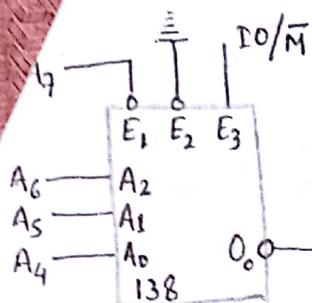
### DMA Channels and Interfacing :-

- The figure on next page shows a logical pin out and internal registers of the 8237. It also shows the interface with the 8085 using a 3-to-8 decoder.
- The 8237 has four independent channels, CH0 to CH3. Internally, two 16-bit registers are associated with each channel: One is used to load a starting address of the byte to be copied and second is used to load a count of the number of bytes to be copied.
- The addresses of these registers are determined by four address lines, A<sub>3</sub> to A<sub>0</sub>, and the Chip Select (CS) signal. Address 0000 on lines A<sub>3</sub>-A<sub>0</sub> selects CH0, Memory Address Register (MAR) and address 0001 selects the next register, CH0 Count. Similarly, all the remaining registers are selected in sequential order. The last eight registers are used to write commands or read status. The MPU accesses the DMA controller by asserting the signal Y<sub>0</sub> of the decoder. Therefore, the addresses of these internal registers range from 00 to OF H as follows:-



## DMA Controller With Internal Registers

41



### DMA Signals

- Signals are divided in two groups: (1) one group shown on left of the 8237 (above) is used for interfacing with the MPU; (2) the second group on right is for communicating with peripherals. Some signals that are necessary to understand the DMA operation are explained as follows:-

- a) DREQ 0 - DREQ 3 (DMA Request) - There are four independent, asynchronous input signals to the DMA channels from peripherals such as floppy disks and the hard disk. To obtain DMA service, a request is generated by activating the DREQ line of the channel.

- b) DACK0 - DACK3 - (DMA Acknowledge) - There are output lines to inform the individual peripherals that a DMA is granted. DREQ and DACK are equivalent to handshake signals in I/O devices.
- c) AEN and ADSTB - (Address Enable and Address Strobe) - There are active high output signals that are used to latch a high-order address byte to generate a 16-bit address.
- d) MEMR and MEMW - (Memory Read and Memory Write) - There are output signals used during the DMA cycle to write and read from memory.
- e) A<sub>3</sub> - A<sub>0</sub> and A<sub>7</sub> - A<sub>4</sub> (Address) : A<sub>3</sub> - A<sub>0</sub> are bidirectional address lines. They are used as inputs to access control registers. During the DMA cycle, these lines are used as output lines to generate a low-order address that is combined with the remaining address lines A<sub>7</sub> - A<sub>4</sub>.
- f) HRQ and HLDA - (Hold Request and Hold Acknowledge) : HRQ is an output signal used to request the MPU control of the system bus. After receiving the HRQ, the MPU completes the bus cycle in progress and issues HLDA signal.

### DMA Execution

The process of data transfer from the peripheral to the system memory under the DMA controller can be classified under two modes: the slave mode and the master mode.

Slave mode : In the slave mode, the DMA controller is treated as a peripheral, using the following steps :

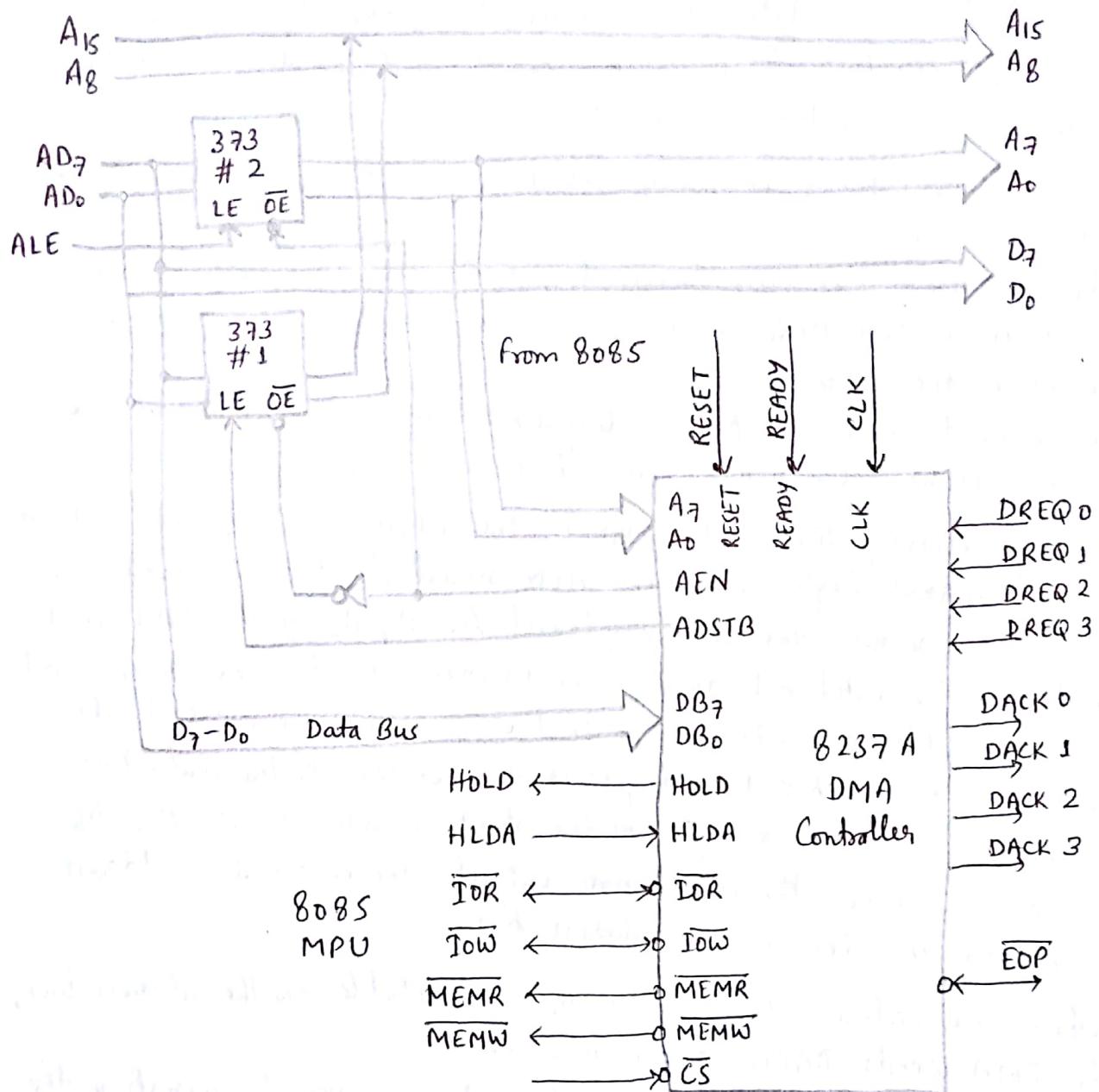
- The MPU selects the DMA controller through Chip Select.
- The MPU writes the control words in channel registers and command / status registers by using control signals I<sub>OW</sub> and I<sub>OR</sub>.

In this mode, the output signals of the 8237, such as A<sub>7</sub> - A<sub>4</sub>, MEMW and MEMR, are in tri-state.

22

Master Mode :- After the initialization, the 8237 in master mode keeps checking for a DMA request, and the steps in data transfer can be listed as follows:-

- a) When the peripheral is ready for data transfer, it sends a high signal to DRQ.
- b) When the DRQ has been received and the channel enabled, the control logic sets HRQ high. (HRQ is connected to the HOLD signal of the 8085).
- c) In the next cycle, the MPU relinquishes the buses and sends the HLDA (Hold Acknowledge) signal to the 8237.
- d) After receiving the HLDA signal, the DMA asserts AEN (Address Enable) signal high. The high AEN signal disables 373 latch #2, thus disconnecting the demultiplexed bus  $A_7-A_0$  of the MPU and enables 373 latch #1 through an inverter. Next, the DMA asserts ADSTB (Address Strobe) high that is connected to Latch Enable (LE) of 373 Latch #1 and places the contents of the data bus, which is a high-order byte of the starting address, on  $A_{15}-A_8$ . At the same time, the DMA also outputs the low order address  $A_7-A_0$  on the low-order address bus.
- e) When the entire address  $A_{15}-A_0$  is available on the address bus, the DMA sends DACK to the peripheral.
- f) The DMA controller continues the data transfer by asserting the necessary control signals ( $\overline{IOR}$ ,  $\overline{IOW}$ ,  $\overline{MEMR}$  or  $\overline{MEMW}$ ) until DACK remains high.
- g) At the end of the data transfer, the DMA asserts  $\overline{EOP}$  (End of Process) signal low that can be used to inform the peripheral that the data transfer is complete. The DMA data transfer can also be terminated by sending a low signal to  $\overline{EOP}$  from outside.



Interfacing 8237A – DMA Controller with the 8085