

# Introduction to Microprocessor

## Introduction

A Microprocessor is a multipurpose programmable, clock driven, register based electronic device that reads binary instructions from a storage device called memory, accepts binary data as input, processes data according to the instructions and provide results as output. The micro processor operates in binary 0 and 1 known as bits are represent at the terms of electrical voltages where 0 represent low voltage and 1 represent high voltage. Microprocessor recognized and processes a group of bits called words and micro processor are classified with their word length such as 8 bit microprocessor with 8 word and 32 bit microprocessor with 32 word.

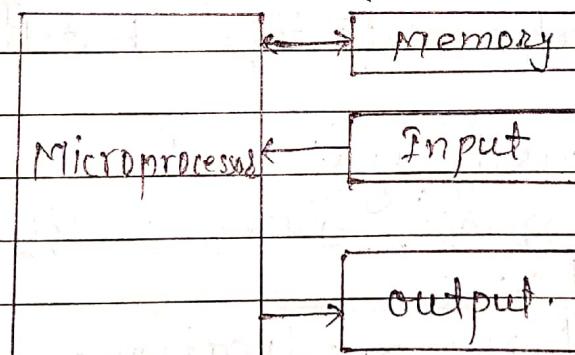


fig: programmable machine.

## Application of Microprocessors

- Microcomputer :- Microprocessor is the CPU of the micro computer.
- Embedded system :- Used in microcontrollers.
- Measurements and testing equipment :-  
Used in signal generators, oscilloscopes, counters, fal voltmeters, x-ray analyzer, blood group analyzers baby incubator, frequency synthesizers, Data acquisition systems, spectrum analyzers etc.
  - washing machine.
  - microwave oven.
  - Scientific and Engineering research.
  - Industry :-  
Used in data monitoring system, automatic weighting, batching system
  - Security systems : smart cameras, CCTV, smart door, traffic light control etc.
  - Military application.

There are two types of digital computer architecture that functionality and implementation of computer system.

1. Von Neumann Architecture.
2. Harvard Architecture.

### 1. Von Neumann Architecture

Storing the data and instruction in a same memory is called as stored program concept. This approach concept was first adopted by John von Neumann and such architecture is named as Von Neumann Architecture.

The Von Neumann architecture consist of three distinct components :-

- central processing unit.
- memory unit.
- input/output (I/O) interfaces.

CPU consist of control unit and arithmetic and logic unit (ALU). ALU is responsible for carrying out all arithmetic and logical operators on data whereas control unit determines the order of flow of instructions that need to be executed in programs by issuing control signals to the hardware.

The memory unit consist of RAM (Read/Write memory), which is the main memory used to

store programming data and instructions. The I/O interfaces allows the users to communicate with the outside world such as storage devices.

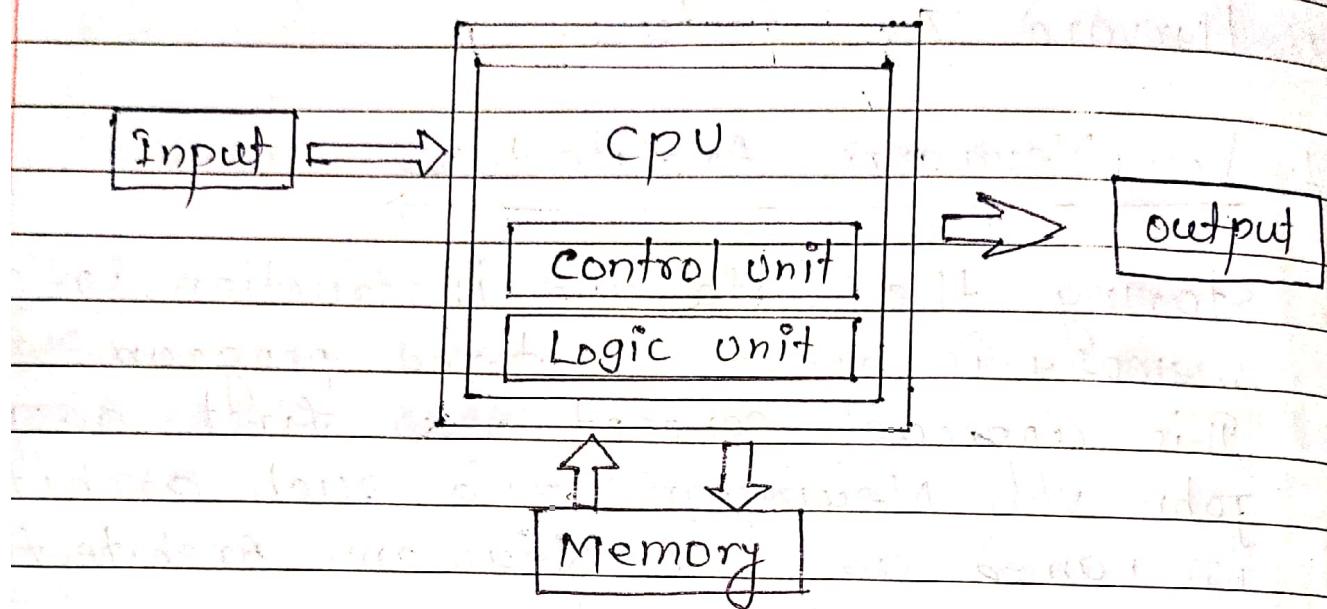


fig: Von Neumann Architecture.

## Harvard Architecture

In von-Neumann architecture same memory is used for storing instructions and data. similarly, a single bus called data bus or address bus is used for reading data and instructions from or writing to memory. It is also limited the processing speed for computers.

The Harvard architecture based computer consist of separate memory space for the programs (instructions) and data. Each space has its own address and

Data buses: so instructions and data can be fetched from memory concurrently and provides significance processing speed improvement.

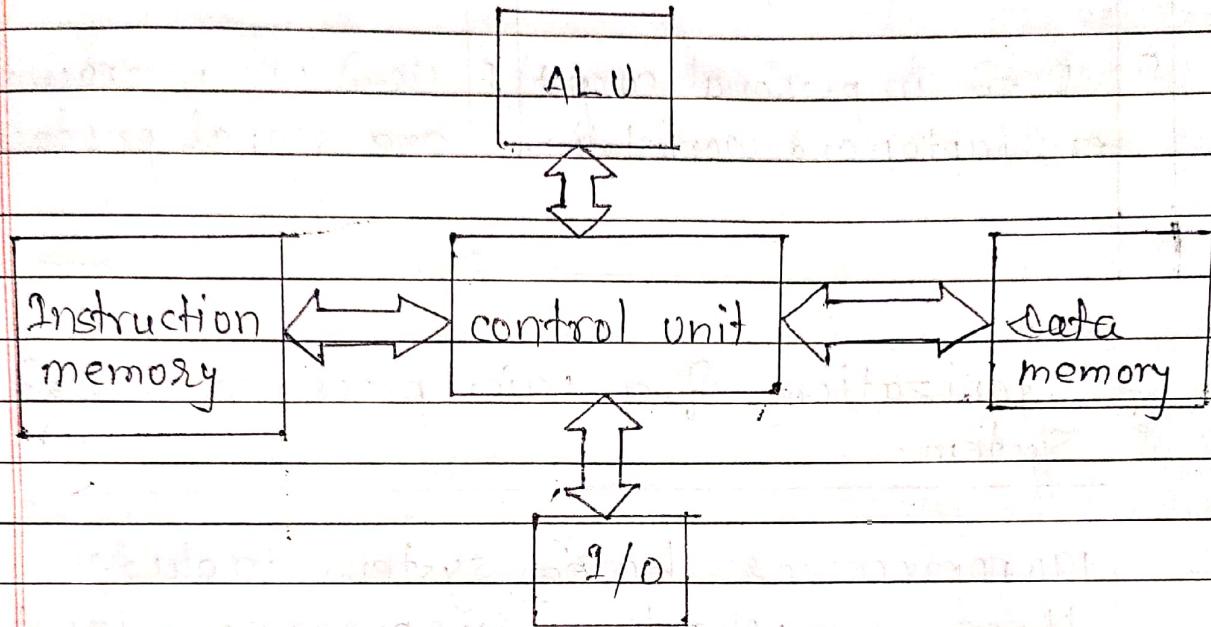


fig: Harvard architecture.

Difference between von Neumann and Harvard Architecture.

<u>s/n</u>	von Neumann Architecture	<u>s/n</u>	Harvard Architecture.
1.	It is a theoretical design based on stored program concept.	1.	It is a modern computer architecture based on the Harvard mark I computer model.
2.	It uses same physical memory address for instruction and data.	2.	It uses separate memory address for instruction and data.
3.	processor needs two clock cycles to execute an instruction.	3.	processor needs one cycle to complete an instruction.

4. Data transfers and instruction fetches cannot be performed simultaneously.

5. Used in personal computer, laptop and workstation.

5. Data and transfer and instruction fetch can be performed at the same time.

5. Used in microcontroller and signal processing.

organization of a microprocessor based system.

Microprocessor based system includes three components microprocessor, input/ output and memory. These components are organized around a common communication path called a bus.

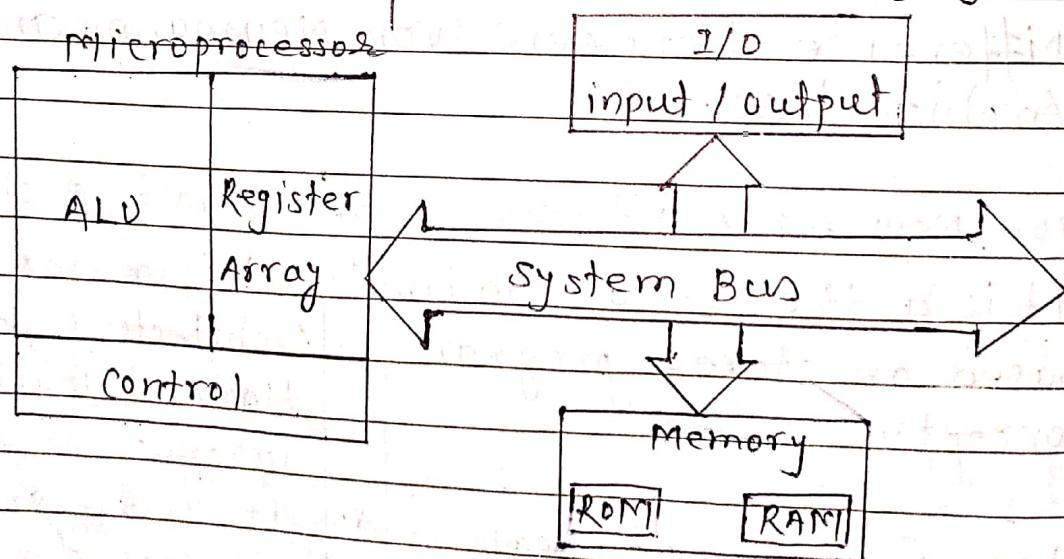
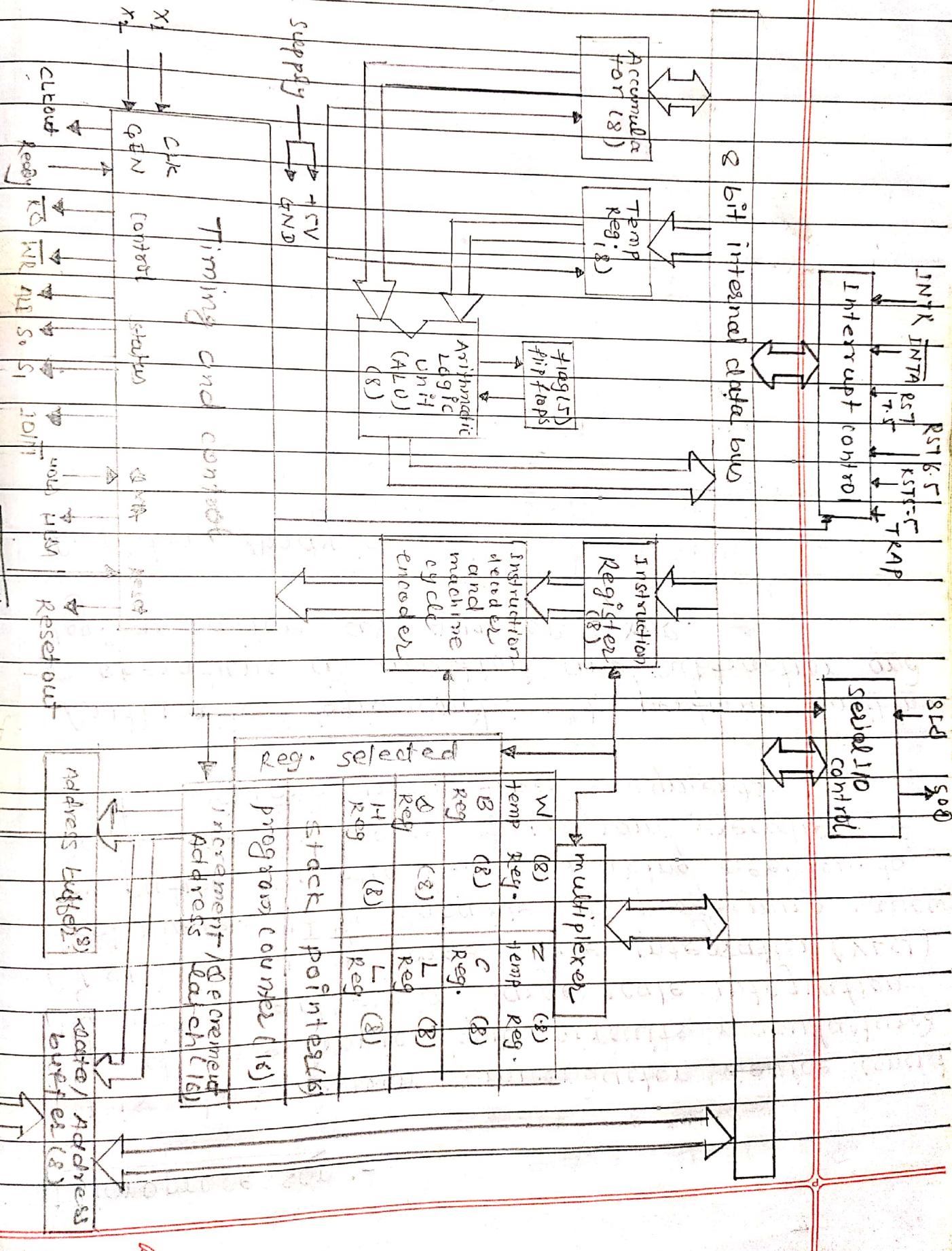


fig: microprocessor Based system with Bus architecture.

## Microprocessor :-

It is clock driven semiconductor device consisting of electronic logic circuits manufactured by using either a large scale integration (LSI) or very large scale integration (VLSI) technique. It is capable of performing various computing functions and making decision to change the sequence of program execution. It is divided into three segments.

- A. Arithmetic / logic Unit : It performs arithmetic operations as addition and subtraction and logic operation as AND, OR & XOR.
- B. Register Array :-



## 8085 Microprocessor Architecture and Operations

1. ALU:- Arithmetic and logic unit performs the competing functions. It includes the temporary register, accumulator, the arithmetic and logic circuit and five flags. The temporary register is used to hold data during arithmetic and logic operation. The result is stored in accumulator. The flag are set or reset according to the result of operation.
2. Accumulator:- It is 8 bit register that is part of ALU. The register is used to store the 8 bit data and to perform arithmetic and logic operations and 8085 micro processor is called accumulator based micro processor. When the data read from input port. It is first moved to accumulator and when the data is sent to output port it must be first
3. Temporary register (W/Z):- They are 8 bit register not accessible to the programmer during program execution, 8085 A place the data into it for brief period.
4. Instruction register (IR):- It is 8 bit register not accessible by programmer. It receives operation codes so that of instruction form internal data bus and passes to the instruction decoder which decodes so that microprocessor knows type of operation is to be performed.

Register array :- It is 8 bit register available accessible to the programmers. Data can be stored upon it during program execution. These can be used individually as 8-bit registers or in pair BC, DE as 16 bit registers. The data can be transferred from one to another. Their contents may be increment decremented and combine logically with content of accumulator.

Register H & L :- They are 8 bit register that can be used in same manner as scratch pad register.

Stack pointer (SP) :- It is a 16-bit register used as a memory pointer. It points to a memory location R/w memory called the stack. The beginning of the stack is defined by loading a 16 bit address in the stack pointer.

Program counter :- It is used to sequence the execution of the instruction. The function of PC is to point the memory address from which the next byte is to be fetched. When a byte is begin being fetch, the PC is incremented by one to point to the next memory location.

6. flag register :- Register consist of five flip flop, each holding status of different states separating is known as flag register and each flip flop are called flags. 8085A can be set or reset one or more of the flags and are sign (S), zero (Z), Auxiliary (AC) and parity (P) and carry (CY). The states of flags indicate the result of arithmetic and logical operations, which is used to decision making process.

- \* carry (CY) :- If the last operation generates a carry its state will 1 otherwise 0.
- \* zero (Z) :- If the result of last operation is zero its state will 1 otherwise zero (0).
- \* sign (S) :- If the MSB of the result of the last operation is 1 (negative) then its state will be 1 otherwise zero (0).
- \* parity (P) :- If the result of last operation has been number 1's (even parity) then its state will be 1 otherwise 0.
- \* Auxiliary carry (AC) :- If the last operation generate a carry from the lower half word It state will be 1 otherwise 0.

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	-	AC	-	P	-	CY

Timing and control unit: This unit synchronizes all the microprocessor operations with the clock and generates the control signals for the communication between the microprocessor and peripherals. The RD and WR signals are the sync pulses indicating the availability of data on the data bus.

Interrupt control:- Various interrupt control signals (INTR, RST 5.5, RST 6.5, RST 7.5 and TRAP) are used to interrupt microprocessor.

serial I/O controls:- Two serial I/O control signals (S<sub>CO</sub> and S<sub>IO</sub>) are used to implement serial data transmission.

# PIN configuration of 8085

X <sub>1</sub>	1	90	VCC
X <sub>2</sub>	2	89	HOLD
RESET OUT	3	38	HLDA
SOD	4	37	CLK(OUT)
SIO	5	36	RESET IN
TRAP	6	35	READY
RST 7.5	7	34	10/M
RST 6.5	8	33	S <sub>1</sub>
RST 5.5	9	32	RQ
INTR	10	31	WR
INTA	11	30	ALE
A0 <sub>0</sub>	12	29	S0
A0 <sub>1</sub>	13	28	A15
A0 <sub>2</sub>	14	27	A14
A0 <sub>3</sub>	15	26	A13
A0 <sub>4</sub>	16	25	A12
A0 <sub>5</sub>	17	24	A11
A0 <sub>6</sub>	18	23	A10
A0 <sub>7</sub>	19	22	A9
GND	20	21	A8

fig: pin configuration of 8085

The intel 8085 is a next generation, complete 8-bit parallel CPU. The 8085 uses a multiple address bus. The address is split between the 8-bit address and 8-bit data bus.

x X<sub>1</sub>, X<sub>2</sub> (Input)

A crystal (or RC, Lc networks) is connected to these two points. The frequency is divided

two therefore to operate a system at 3 MHz, the crystal should have a frequency of 6 MHz.

#### \* Reset output :- ( RST out )

This signal indicates that the microprocessor is being reset. It is also used to reset other devices.

#### \* TRAP (input) :-

It is a non-maskable interrupt and has the highest priority of any interrupt.

RST 5.5 / RST 6.5 / RST 7.5 (input) : OR (RESTART interrupts) :- These are the vector interrupt that transfer the program control to the specific memory locations.

RST 7.5      Highest priority

RST 6.5

RST 5.5      lowest priority

#### \* INTR (input)

INTERRUPT ACK REQUEST :- This is used for the general purpose interrupt.

#### \* INTA (output)

INTERRUPT ACKNOWLEDGE :- This is used to acknowledge an interrupt.

A<sub>00</sub> - A<sub>07</sub> :- Multiplexed Address / data bus

Memory address appears on the bus during the first clock cycle of a machine state.

It then becomes the data bus between the second and third clock cycle.

\* A8 - A15 (output - 3 state)

Address bus: The most significant 8-bit of the memory address or the 8-bit of I/O address.

\* S0, S1 (output)

These signals similar to J0/M1 can identify various operations, but they are rarely used in small systems.

S1      S0      Function

0      0      HALT

0      1      WRITE

1      0      READ

1      1      FETCH

\* ALE (Output) (Address Latch Enable):

It indicates that the bits on A8<sub>7</sub> - A8<sub>0</sub> acts as lower 8-bit address bus (A<sub>7</sub> - A<sub>0</sub>) when logic high (1). IF ALE = 0

It acts as a data bus (D<sub>7</sub> - D<sub>0</sub>)

\* WR (output 3 state)

WRITE, indicates the data in the data bus is to be written in to the selected memory or I/O location.

\* RD (output 3-state)

READ, indicates the selected memory or I/O

device is to be read and that the data bus is available for the data transfer.

\*  $I/O / \bar{M}$  (output) :-

This is a status signal used to differentiate between I/O and memory operation when it is high (1). It indicates an I/O operation when it is low (0). It indicates memory operations.

\* READY (input) :- If ready is high (1) during a read or write cycle, it indicates that the memory or peripheral is ready to send or receive data. If ready is low (0) the CPU will wait for READY to go high (1) before completing the ready or write cycle.

\* CLK(OUT) :-

This signal can be used as the system clock for other devices.

\* HLD : - It indicates that the CPU has received the HOLD request and acknowledged that request.

\* HOLD (Input) :-

This signal indicates that peripheral such as DMA controller is requesting the use of address and data bus.

VCC :- +5 volt supply.

## Addressing Mode of 8085 Microprocessor

Different ways in which a processor can access data are referred as its addressing modes. Various addressing mode, are

1. Register addressing mode:-

It is the most common form of data addressing

Transfers a copy of a byte/word from source register to destination register

Instruction	source	destination
MOV A, B	Register B	Register A

\* It carried out with 8 bit registers A, B, C, D, E, H & L

\* It is important to use register of same size.

Example:- MOV A, B : copy B in to A

MOV SP, H : copy H pair in to SP

Immediate

2. Intermediate addressing mode:-

The term intermediate implies that the data immediately follows the hexadecimal opcode in the memory.

It transfers the source immediate byte/ word of data in destination register or memory location.

instruction	source	destination
MVI C, 3AH	data 3 AH	Register C

example:- MOV A, 90 : copy 90 into A

LXI H, 1234H : copy 1234H in H.

## 3. Direct addressing mode :-

In this scheme, the address of the data is defined in the instruction itself.

instruction	source	destination
LOAD 2000H	memory location 2000H	Register A

Example :-

LHLD :- copies the content of 1000H address memory to L and 1000L H memory to H.

## 4. Register indirect addressing mode :-

- \* Register indirect addressing mode allows to be addressed at any memory location through an address held in any of the H pair, B pair and D pair register.
- \* It transfers byte/word between a register and a memory location addressed.

instruction	source	destination
MOV C,M	Assume HL = 1000H and M is the content of 1000 H address	Register

Example :-

MOV C,H :- copies the word contents of the memory location address HL pair into C.

STAX B :- copies A into the memory location address by B pair

JMP 4000H

## 5. implied Addressing mode:-

- \* The addressing mode of certain instructions is implied by the instruction's functions.

Instruction	source	destination
STC		carry flag

STC :- set carry flag

CMC :- complementary carry flag.

DAA :- decimal adjust accumulator content.

## # Interrupts in 8085

Interrupts are the signals generated by the external devices to request the microprocessor to perform a task. There are 5 interrupts signals, i.e. TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR. Interrupts are classified into following.

\* Vector interrupts:- In this type of interrupt, the interrupt address is known to the processor for example:- RST 7.5, RST 6.5, RST 5.5, TRAP

\* Non-vector interrupts:- In this type of interrupt, the interrupt address is not known to the processor so, the interrupt address needs to be sent externally by the device to perform interrupt for example: INTR.

Maskable interrupt :- In this type of interrupts, we can disable the interrupt by writing some instructions in to the program : for example: RST 7.5, RST 6.5 and RST 5.5

NON-Maskable interrupt :- In this type of interrupt, we can't disable the interrupt by writing some instructions in to the program : for example TRAP.

Software interrupt :- In this type of interrupt the programmer has to add the instruction in to the program to execute the interrupt. There are 8 software in 8085, i.e RST0, RST1, RST2, RST3, RST4, RST5, RST6, and RST7.

Hardware interrupt :- There are 5 interrupt pins in 8085 used as hardware interrupts i.e TRAP, RST 7.5, RST 6.5, RST 5.5, INTA.

Note:- INTA is not a interrupt : It is used by microprocessor for sending acknowledgement. TRAP has the highest priority, then RST 7.5 and so on.

## Instruction cycle

Machine cycle :- It is defined as the time required to complete one operation of accessing memory I/p, O/p or acknowledging and external request. This cycle consist of 3 to 6 T states.

### Machine cycle of 8085

The 8085 microprocessor has 5 basic machine cycle. They are

- \* opcode fetch cycle (4T)
  - The microprocessor use this cycle to take the op-code of an instruction from the memory location to processor.
  - The op-code is taken from memory and transferred to instruction register for decoding and execution.
  - The time required to complete this cycle is 4 to 6 T state.

### Timing diagram of op-code fetch cycle.

The signal involves during machine cycle are CLK, A<sub>15</sub> - A<sub>8</sub>, A<sub>D7</sub> - A<sub>D0</sub>, I<sub>O/m</sub>, R<sub>O</sub>, WR and S<sub>1</sub>, S<sub>0</sub>

$I_0/M$

$S_1$

$S_0$

operation

0	1	1	opcode fetch (OF)
0	1	0	Memory read
0	0	1	Memory write
1	1	0	I/O read (I/0R)
1	0	1	I/O write (I/0W)
1	1	1	Acknowledge of INTR (INTA)

signal

$T_1$        $T_2$        $T_3$        $T_4$

CLK

A<sub>15</sub>

A<sub>8</sub>

A<sub>7</sub>

A<sub>6</sub>

ALE

I<sub>0/M</sub>

R/W

Higher order memory address

unspecified

Local  
order

OP code

status  $I_0/M = 0, S_0 = 1, S_1 = L \cdot OP$

Fig:- Timing diagram of opcode fetch cycle.

The op-code fetch timing diagram can be explained as below.

- i) The MP places the 16-bit memory address from the program counter on address bus. At time period  $T_1$  the higher order memory address is placed on the address line  $A_{15} - A_8$ . When ALE is high the lower address is placed on the bus  $A_{07} - A_0$ . The status signal  $I/O/\bar{M}$  goes low indicating the memory operation and two signals  $S_1 = 1$ ,  $S_0 = 1$  to indicate op-code fetch operation.
- ii) At time period  $T_1$ , the MP sends  $\overline{RD}$  control line to enable the memory read. When memory is enable with  $\overline{RD}$  signal, the opcode value from the addressed memory location is placed on the data bus with ALE low.
- iii) The opcode value is reached at processor register clearing  $T_2$  time reached period when data (op-code value) is arrived, the  $\overline{RD}$  signal goes high. It causes the bus to go into high importance state.
- iv) The op-code byte is placed in instruction decoder of MP and the op-code is decoded and executed. This happens during time period  $T_4$ .

## Memory Read cycle:-

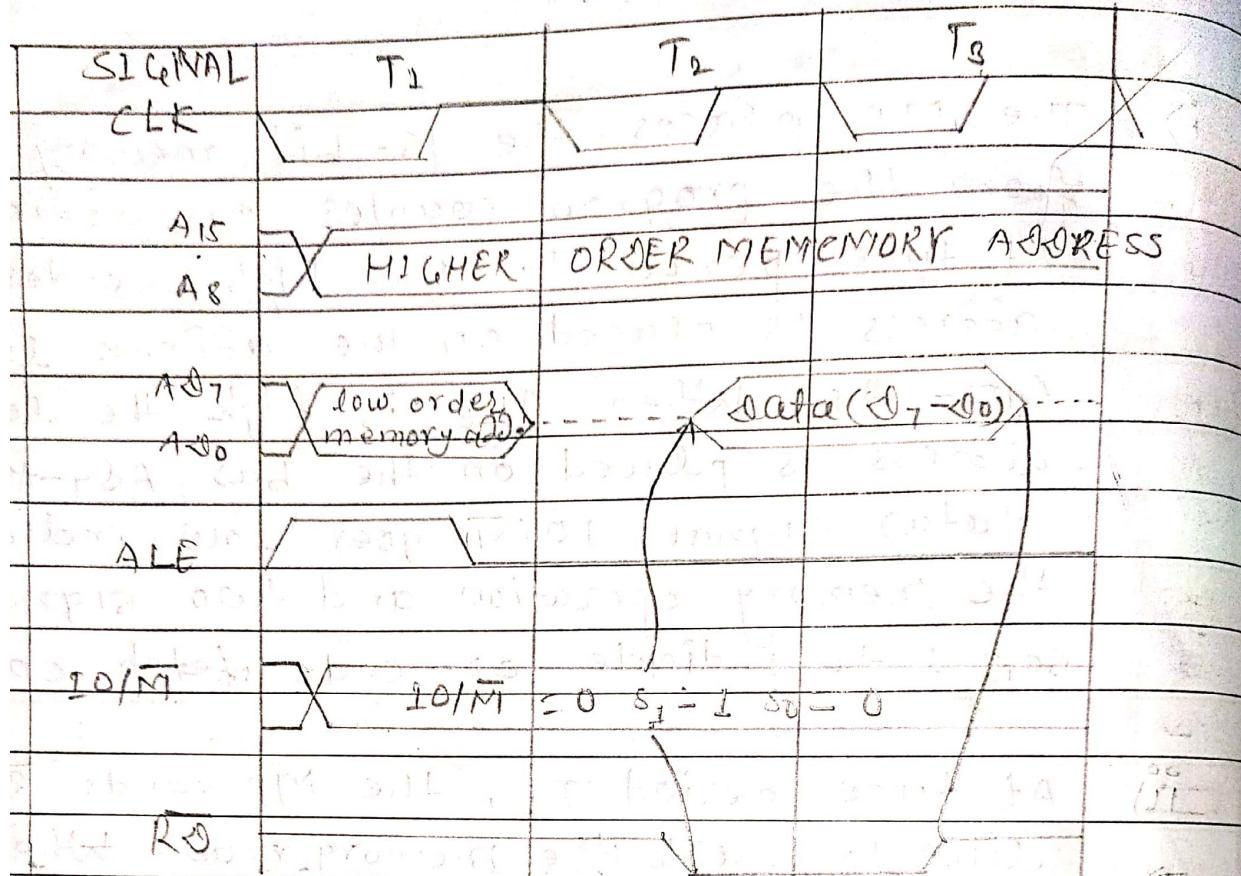


Fig:- Timing diagram for memory read cycle.

The memory read timing diagram can be explained as below.

The MP places the 16-bit memory address from the program counter on address bus. At time period  $T_1$ , the higher order address is placed on the address line A15-A8. When ALE is high, the lower address lines placed on the bus A8-A1.

The status signal  $IO/M$  goes low indicating the memory operation and two status signals  $S_1 = 1$ ,  $S_0 = 0$  to indicate memory read operation.

- (ii) At time period  $T_2$ , the MP sends RD from the addressed memory location is placed on the data bus with ALE low.
- (iii) The data is reached at processor register during  $T_3$  state when data is readed and the RD become high. It cause high impedance state.

# Timing diagram for memory write cycle.

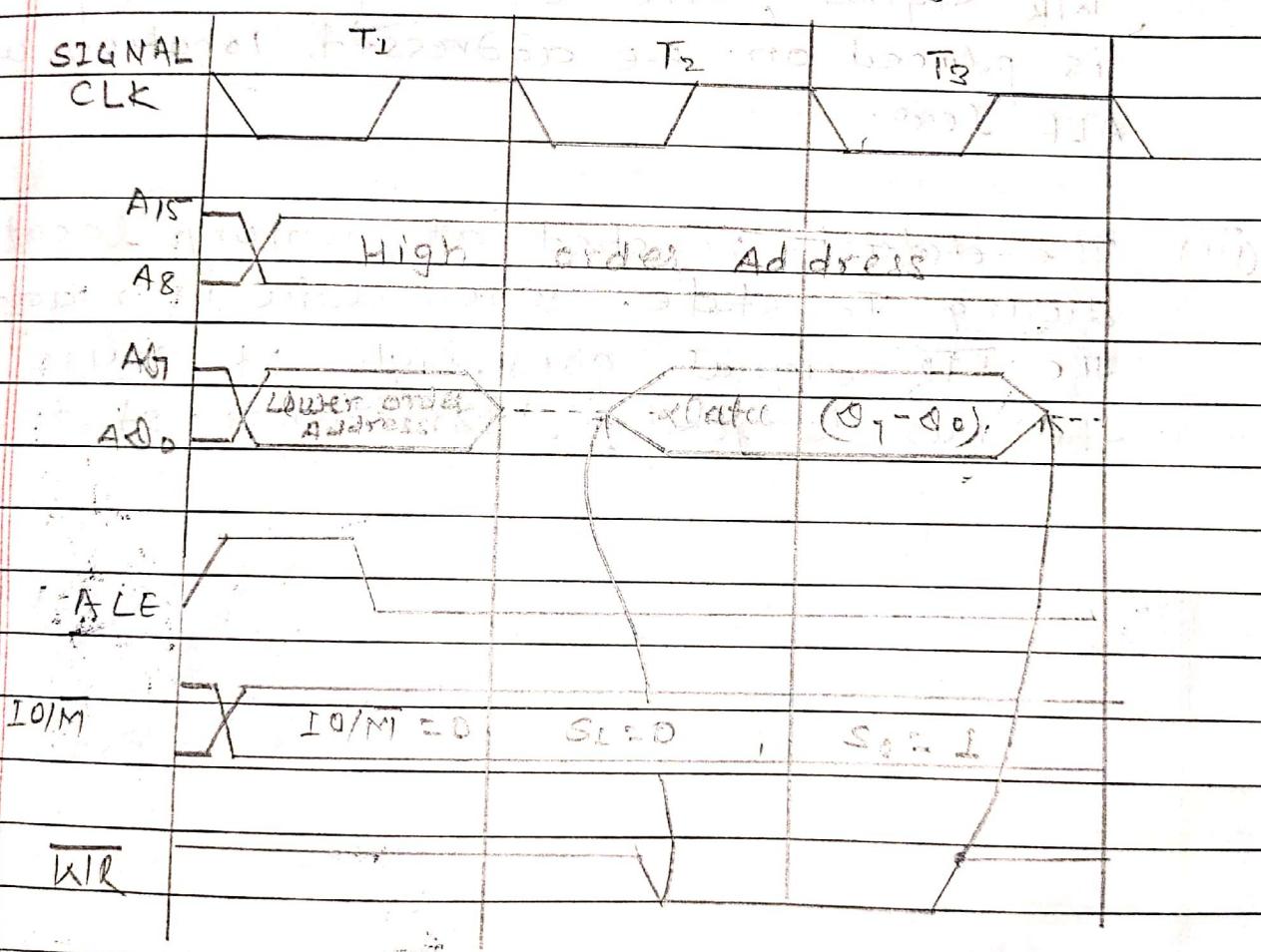


fig: Timing diagram of Memory read cycle.

- i) The MP places 16-bit memory address from the program counter on address bus. At time period  $T_1$ , the higher order memory address is placed on the address lines.

A<sub>15</sub>-A<sub>8</sub>. When ALE is high the lower address is placed on the bus A<sub>0</sub>-A<sub>3</sub>.

The status signal I<sub>O/M</sub> goes low indicating the memory operation and two signals S<sub>1</sub>=0, S<sub>0</sub>=1 to indicate memory write operation.

- (ii) At time period T<sub>2</sub>, the MP sends W/R control line to enable the memory write. When memory is enable with W/R signal, the data from the processor is placed on the addressed location with ALE low.
- (iii) The data is reached at memory location during T<sub>3</sub> state. When data is reached the W/R signal goes high. It causes the bus to go high impedance state.

## # I/O Read and I/O write cycles

Timing diagram for I/O read cycle.

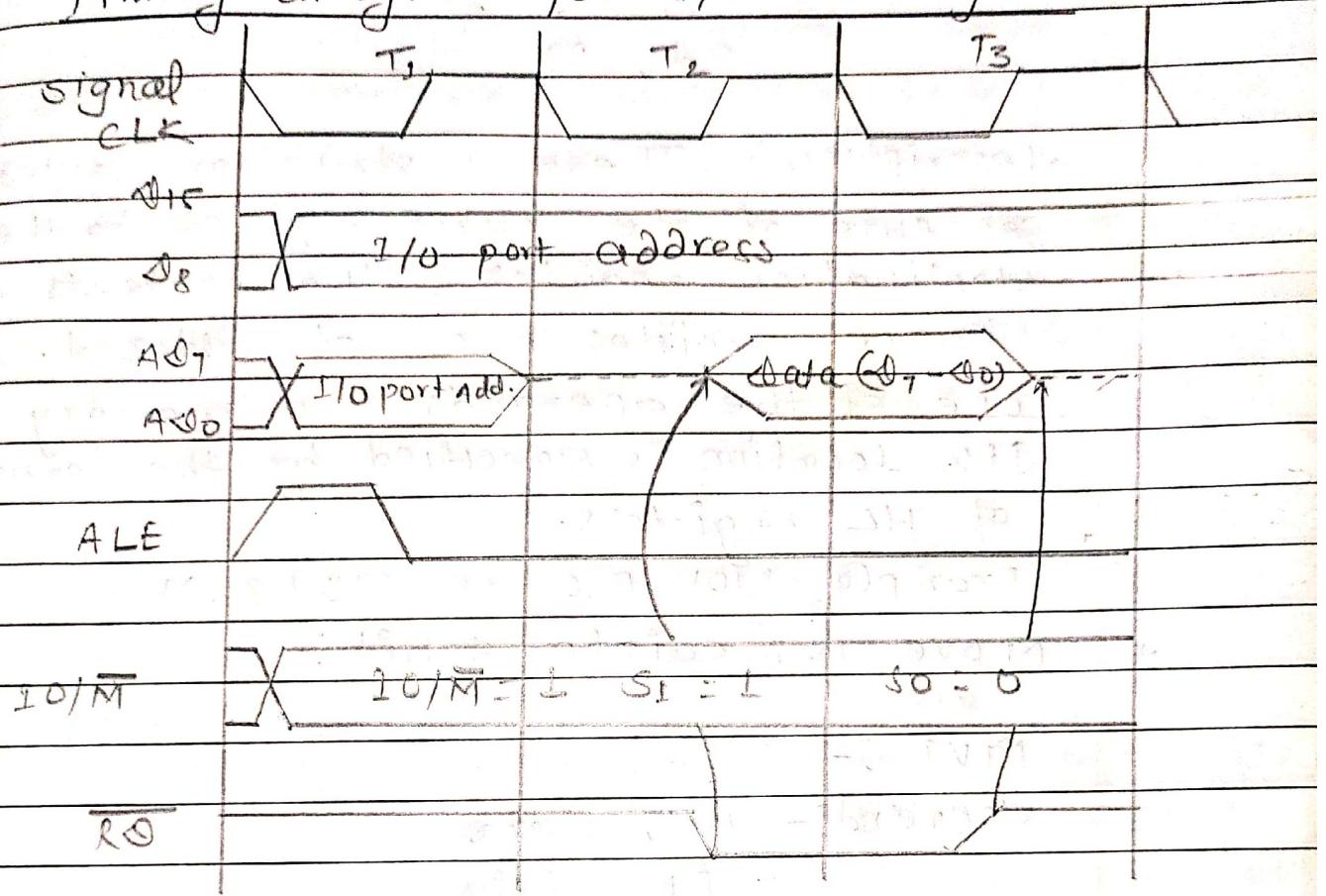
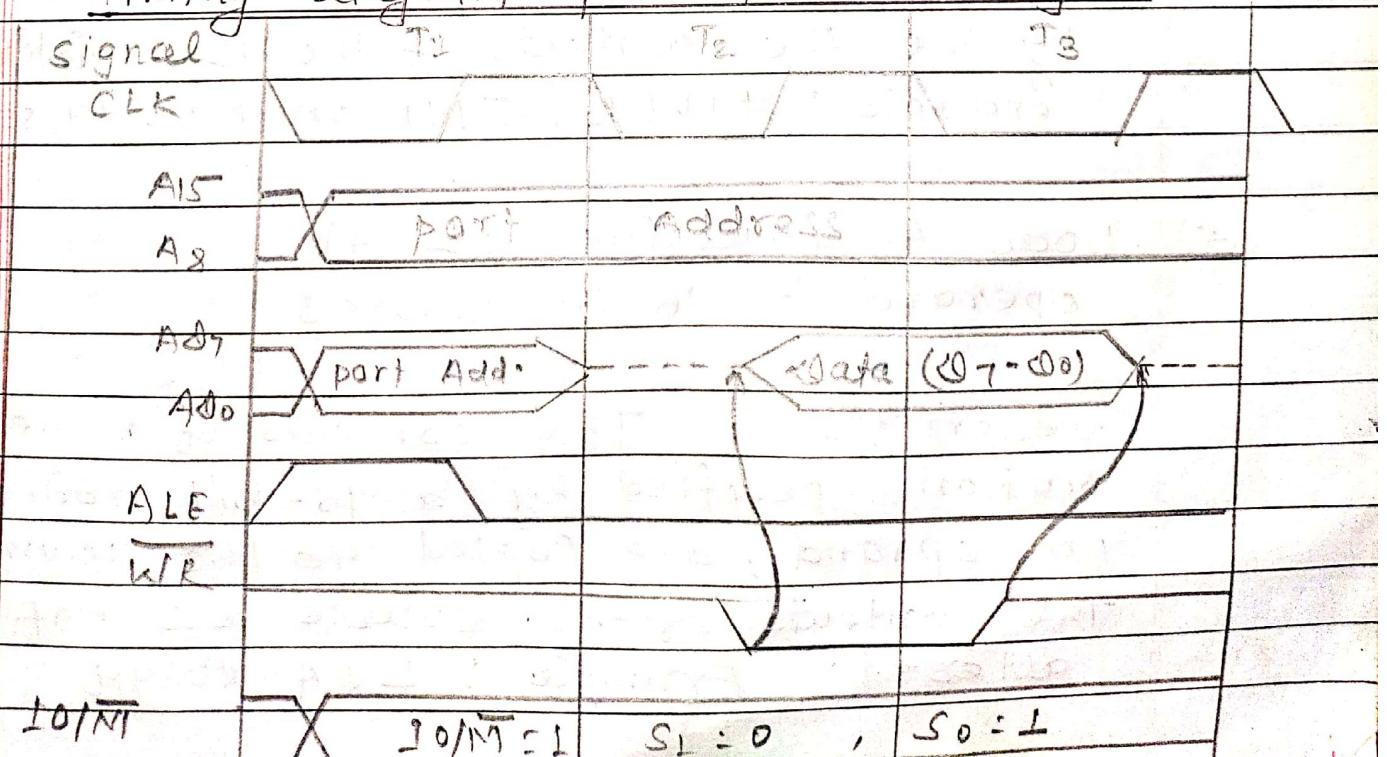


fig:- Timing diagram for I/O Read machine cycle.

Timing diagram for I/O write cycle.

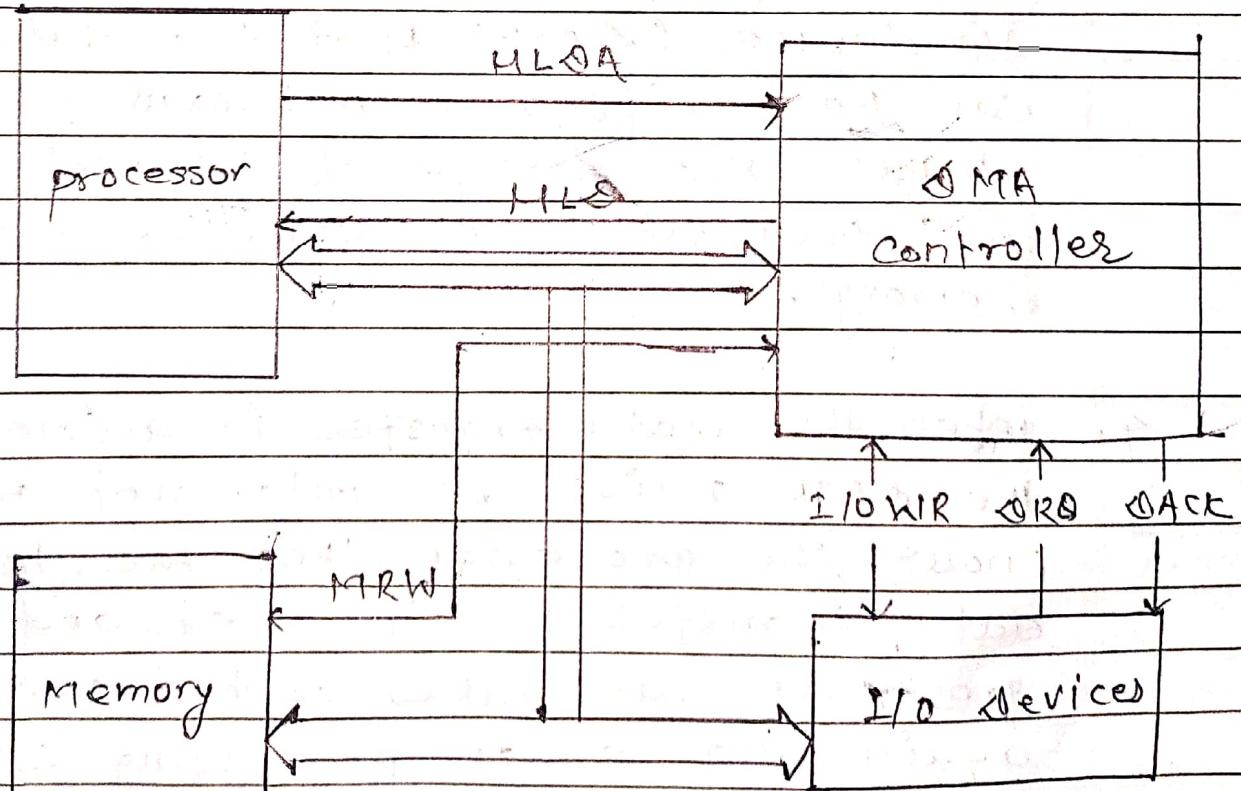


## chapter 5

पाठ्याला  
Date \_\_\_\_\_  
Page \_\_\_\_\_

### Direct Memory Access (DMA)

It is a feature of computer system or technique that allows the hardware subsystems to access main memory without involving the CPU. By using DMA large amount of data can be transferred between memory and the peripheral without severely impacting CPU performance. In the during the DMA transfer the bus CPU has no control on memory buses. A DMA controller takes over the buses to manage the transfer directly between the I/O devices.



DMA controller data transfer.

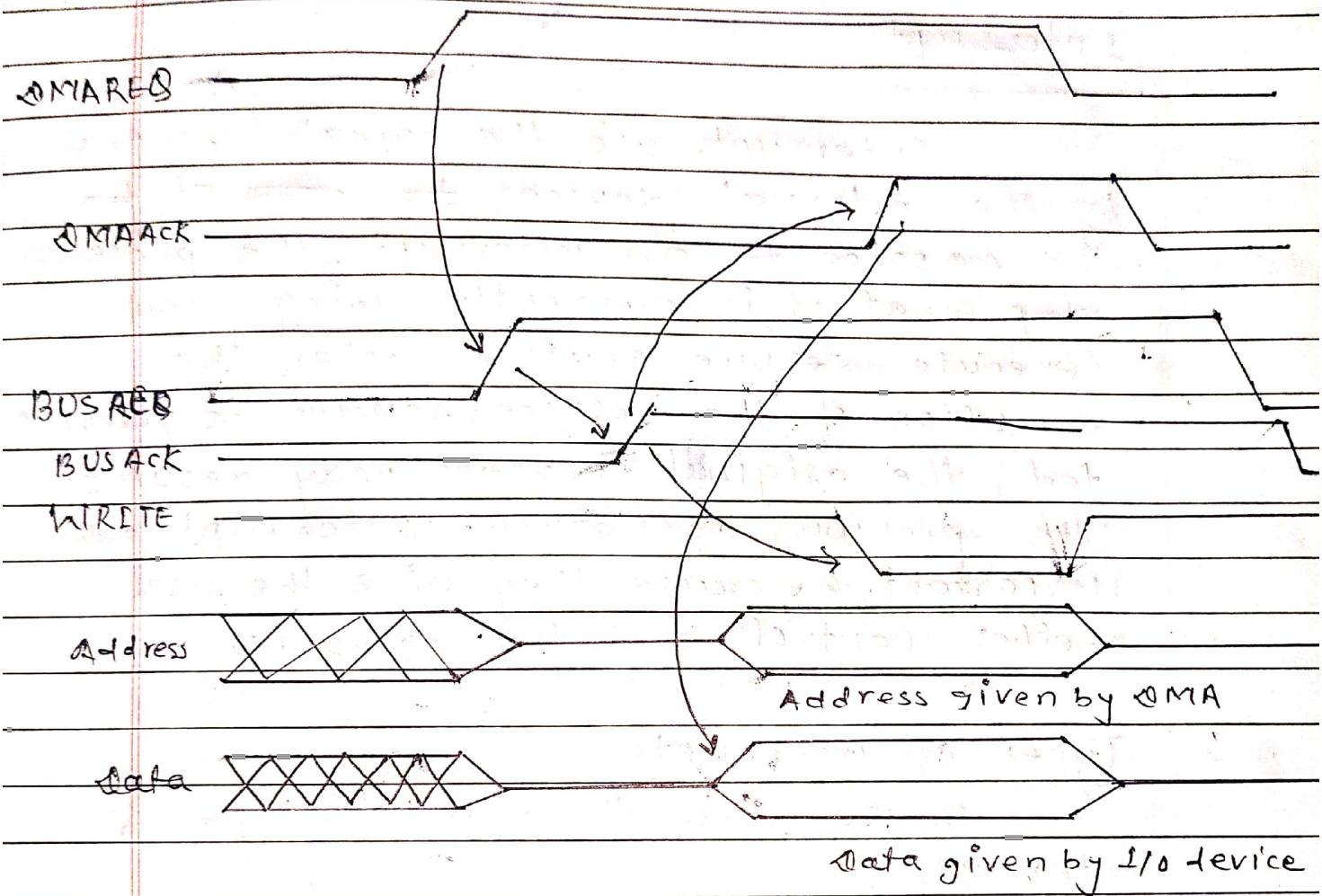
Whenever I/O device wants to transfer the data to or from memory, it sends DMA controller request (DRQ) to the DMA controller. DMA controller accepts this DRQ and asks the CPU to hold for a few clock cycles by sending it the hold request (HLD).

CPU receives the Hold request (HLD) from DMA controller and relinquishes the bus and sends the Hold acknowledgement (HLDACK) to DMA controller.

After receiving the Hold acknowledgement (HLDACK), DMA controller acknowledges I/O devices (DACK) that the data transfer can be performed and DMA controller takes the charge of the system bus and transfers the data to or from memory.

When the data transfer is accomplished the DMA raise an interrupt to let know the processor that the task of data transfer is finished and the processor can take control over bus again and start processing when it has left.

## DMA Timing diagram



### \* USE of DMA :-

1. Many hardware systems use DMA, including disk drive controllers, graphics cards, network cards, network card and sound card.
2. DMA are also use for intra-chip data transfer in multi-core processors.
3. computer that have DMA channels can transfer data to and from devices with



Much less CPU overhead than computer without DMA channels.

## Interrupt

Interrupts are the signals generated by the external devices to request the computer. In response to an interrupt, the process stops what it is currently doing and execute service routine when the execution of the service routine is finished, the original process may resume its previous operation. Interrupts are important because they give the user better control over the computer.

### Types of interrupts

There are three major types of interrupts

- 1) External interrupts
- 2) Internal interrupts
- 3) Software interrupts.

1) External interrupt :- External interrupt are initiated via the microprocessor's interrupt pins. There are two types of external interrupts

- a) Maskable interrupt :- In simple way Maskable interrupt is a those type of interrupt which we can disable by writing some instruction into the program.



b) Non-maskable interrupt :- It is a interrupt we can't disable or ignore by writing instruction in to the program.

2) Internal interrupt :- An interrupt is a type of interrupt that results from a specific event within the processor such as the occurrence of an error due to the division by zero which produces an internal interrupt called divide by a zero interrupt.

3. Software interrupt :- Software interrupt is initiated by executing an instruction. It can be used by the programmer to initiate a interrupt procedure at any desired point. The most common use of software is associated with a supervisor call instruction. In 8085 RST0, RST1---- etc are software interrupt.

Interrupt priority :-

Interrupt is a signal or condition that causes to stop the current execution ( save states of execution ), and do the function call like do the service the signal or condition. The signal and condition is come from external devices and internal device send the signal to the processor to stop the current execution. The data transfer between the CPU and peripherals are initiated by the CPU. CPU can not transfer the data unless the peripherals are ready.

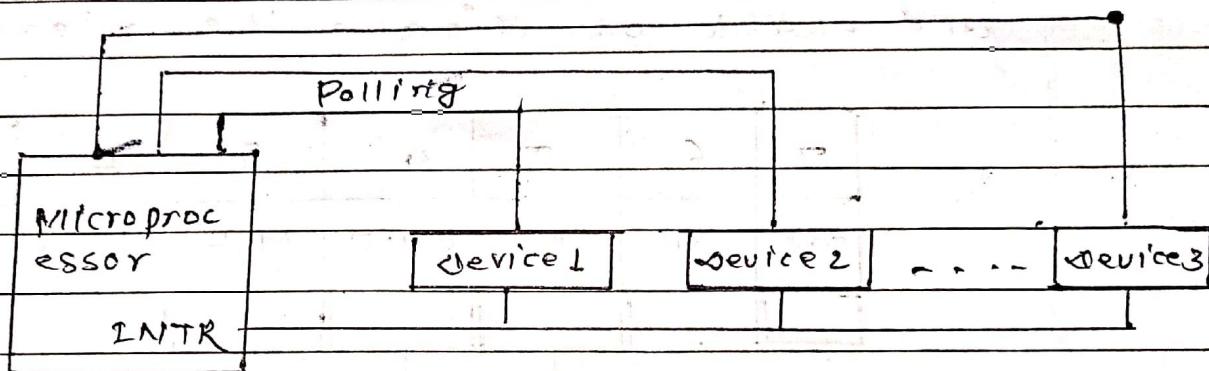
Main job of the interrupt system is to identify the source of interrupt. There is a possibility to several devices send request for the communication simultaneously. Then the system must also decide which device to serve first. High priority interrupt are serve first then the low priority interrupt. Device with high transfer speed such as magnetic disk are given high priority and slower device such as keyboard receive low priority.

There are mainly two ways of servicing multiple interrupt

(i) interrupt based I/O is efficient compare to polled I/O.) justify.

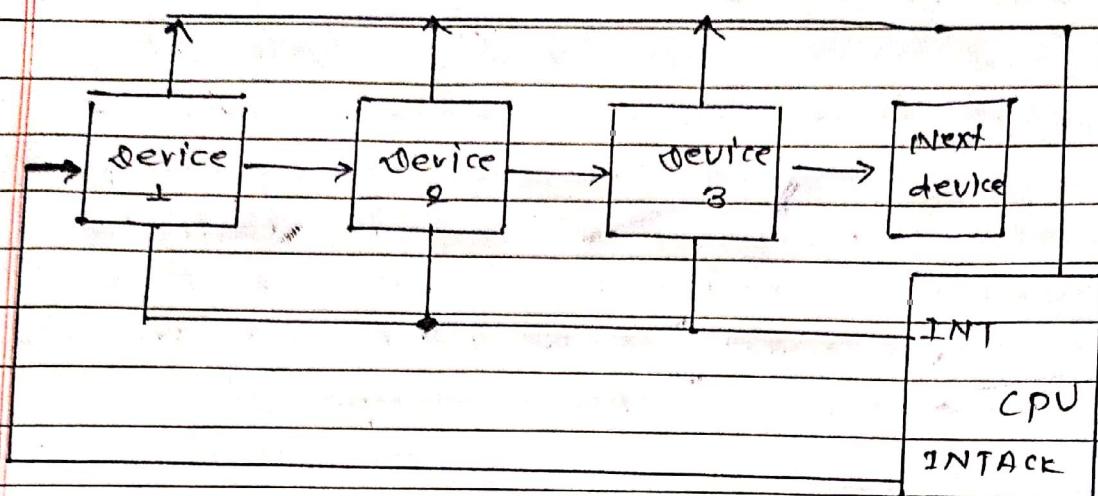
④ chained (vectored) interrupt.

⑤ polled interrupt: polled interrupts are handle by using software which is slower than hardware. The processor generate the common interrupt service routine for all the devices.



several device are connected to a single line INTR of MP. when INTR goes up. the processor saves the contents of PC and registers and then branches to an address defined by the manufacturer of the processor. the user can right write a program at this address to find the source of the interrupt; by starting the polled from highest priority device.

## ⑧ Chained interrupt :-



Chained interrupt is used for the handling the priority interrupt. In this method the device which is highest priority is placed first position followed by lower priority devices. All the devices connected in serial method.

In this method when the I/O devices generate Interrupt then INT become high (1), then CPU doesn't know which device generate the Interrupt so CPU make INTACK high (1). Then if the device of highest priority generate this Interrupt then it consume this INTACK signal and pass (0) for other devices. Then vectored address provide by this device for checking the ISR. If highest priority device doesn't generate the interrupt then it check second highest priority so on.

## पाठ्याला Date \_\_\_\_\_ Page \_\_\_\_\_

# SET interrupt mask (SIM) instruction

1 byte instruction  
can be used for three different functions

one function is set mask for RST 7.5, 6.5 and 5.5 interrupts. This instruction reads the contents of the accumulator and enable or disable the interrupt.

	7	6	5	4	3	2	1	0
508	SSE	X	RST 7.5	MSE	MTS	M6.5	M5.5	

**RST 7.5**

if bit 7 = 1 then ignore serial output data. RST 6.5

if bit 6 = 1 then ignore serial output data. RST 5.5

if bit 5 = 1 then ignore serial output data if bit 0 = 0 available

if bit 4 = 1 then ignore serial output data if bit 0 = 1 masked

all other bits of mask enable serial output data. Most set enable

if bit 3 = 1 then reset RST 7.5, 0 & 1

if bit 2 = 1 then reset RST 6.5, 0 & 1

if bit 1 = 1 then reset RST 5.5, 0 & 1

if bit 0 = 1 then disable serial output data

If bit 7 is output to serial output data.

serial output data, ignored if bit 6 = 0

fig: SIM instruction.

## Pending interrupt :-

When one interrupt request is being served, other interrupt may occur resulting in a pending request. When more than one interrupt occur simultaneously, the interrupt having high priority is served and the interrupt with low priority remain pending. The 8085 has an instruction RIM using which the programmer can know the current status of pending interrupts. This instruction gives the current status of only maskable interrupts.

## Read instruction mask (RIM)

- ④ Read interrupt Mask.
- ④ 1 byte interrupt instruction.
- ④ Can be used for followings.
- a) To read the interrupt mask. This instruction loads the accumulator with 8-bit indicating the current status of the interrupts
- b) To identify the pending interrupts. Bits D4, D5 and D6 identify the pending interrupts

7	6	5	4	3	2	1	0
S10	I7·5	I6·5	I5·5	IE	M7·5	M6·5	M5·5

	<u>pending interrupt</u> ↓ = pending	<u>Interrupts masks</u> ↓ = masked
serial input data		interrupt enable

## # difference between vector and non vector interrupt.

S.N.	vector interrupt	S.N.	NON vector interrupt
1.	In this interrupt the Address of the service routine is known to the processor.	1.	In this interrupt the Address of the service routine is provide externally.
2.	In 8085 mp: RST7.5, RST5.5, RST6.5, trap are vectored.	2.	In 8085 mp:, INTR non vector interrupt
3.			

## I/O interface

parallel and serial communication

The data transfer between two digital devices. Data is transferred from one device to another in analog and digital formate. Data is transfer in the form of bits between two digital devices. There are two method of data transfer i.e Serial communication / transmission parallel Transmission.

### Serial communication

This is common method of transmitting data between a computer and a peripheral. This method is used when data transfer rates are very low or the data must be transferred over long distance and cost of cable and synchronization difficulties makes parallel communication impractical.

### Serial data transmission modes

When data is transmitted between two pieces of equipment, three communication modes of operation can be used.

**Simplex:** In a simple connection, data is transmitted in one direction only.

for example from a computer to printer that can't send status signals back to the computer.

Half duplex : In a half-duplex connection, two-way data transfer of data is possible, but only in one direction at a time.

full duplex : In a full-duplex configuration, both ends can send and receive data simultaneously. which technique is common in our PCs.

### Serial data Transfer schemes

Like any data transfer methods, serial communication also requires coordination between the sender and receiver. For example when to start the transmission and end to end it when one or particular bit ends and another begins.

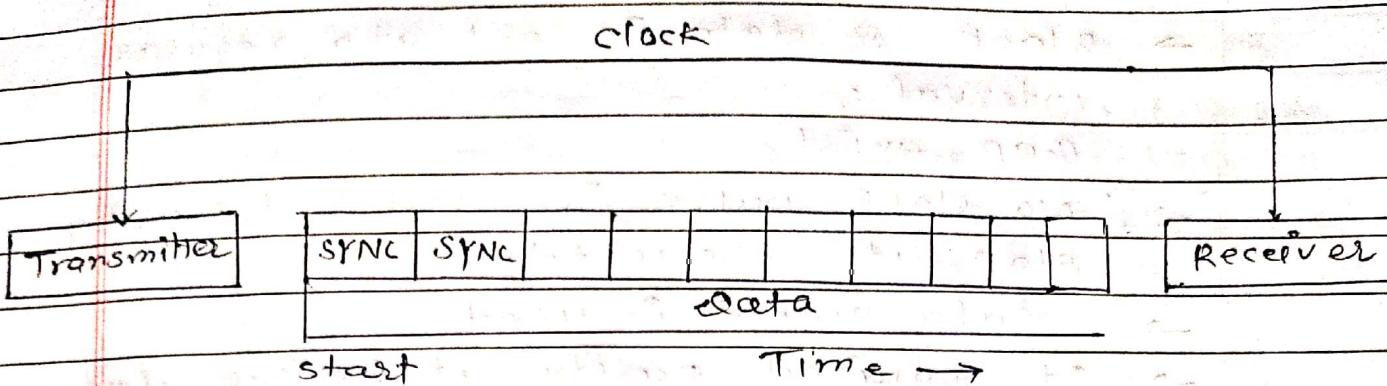
There are two ways to synchronize the two ends of the communication.

Synchronous data transmission

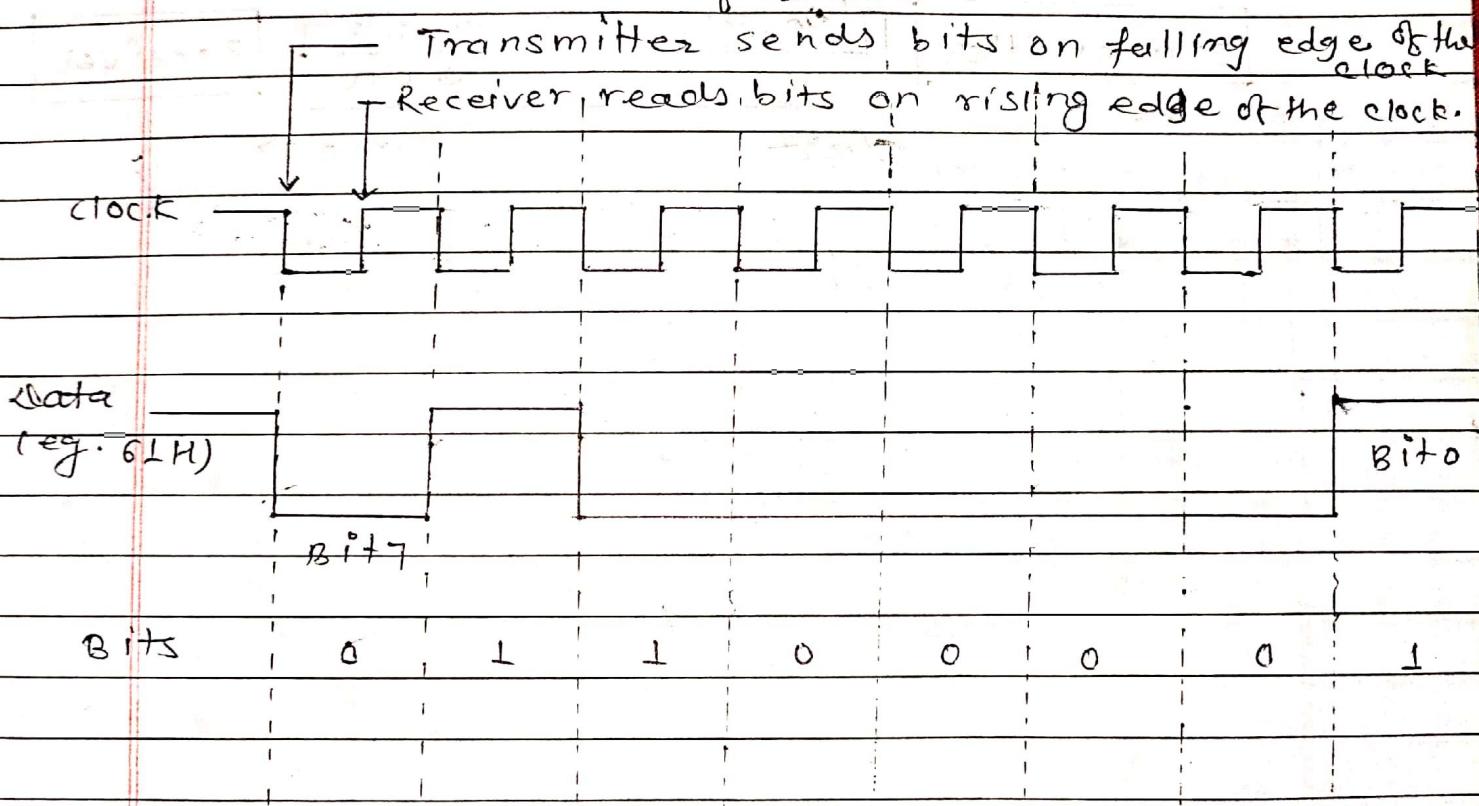
Asynchronous data transmission.

### Synchronous data transmission :-

In synchronous transmission, the stream of data to be transferred is encoded and sent on one line and periodic pulses of voltage which is often called the "clock" is put on another line that tells the receiver about the beginning and ending of each bit.



(a) Synchronous serial transmission format.



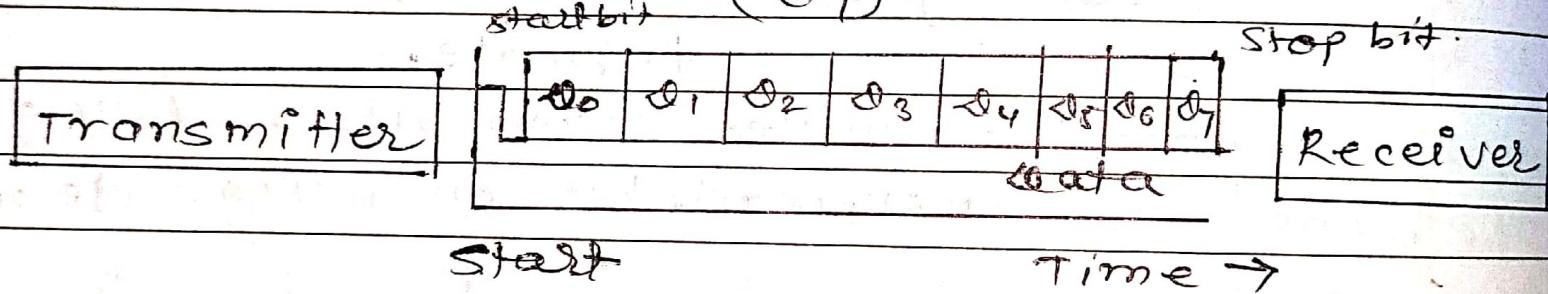
Note: Many synchronous protocols send MSB first.

**Advantages:-** The only advantage of synchronous data transfer is the lower overhead and thus, greater throughput, compared to synchronous one.

**Disadvantages:-** It is slightly more complex. Hardware is more expensive.

## Asynchronous data transmission

- Block of data is sent at a regular interval.
- Gap exist
- No. clock pulse is used but start & stop bits are used.
- Data signal is used.
- It required parity bit before data transmission (x-7)

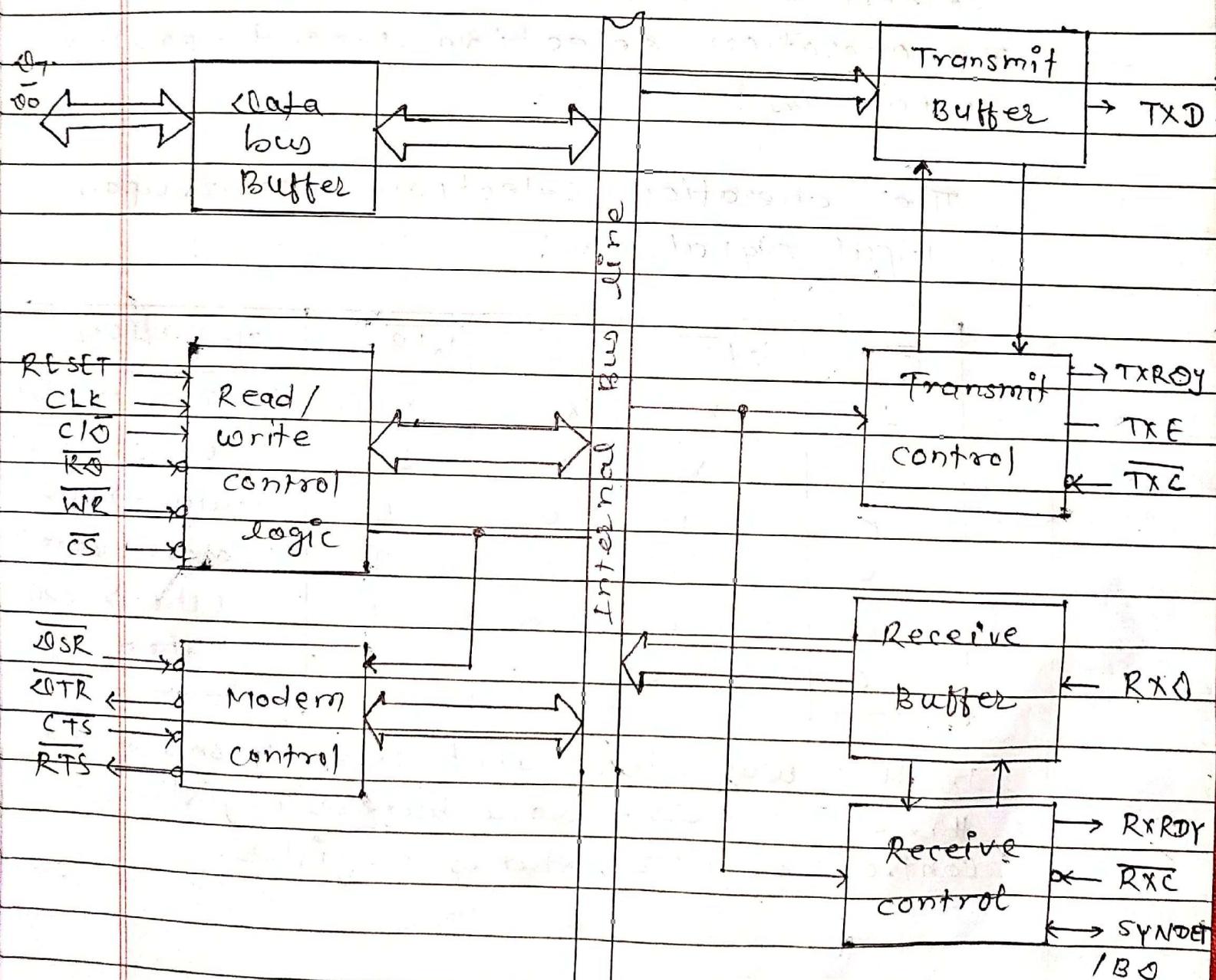


# पाठ्यालू

Date \_\_\_\_\_  
Page \_\_\_\_\_

## Intel 8251 (USART - Universal synchronous Asynchronous Receiver transmitter)

The 8251 is a USART for serial data communication. As a peripheral device of microcomputer system, the 8251 receives parallel data from the CPU and transmits serial data after conversion. This device also receives serial data from the outside and transmits parallel data to the CPU after conversion.



It contains the following blocks

Data bus buffer -

This block helps in interfacing internal data bus of 8251 to the system data bus. The data transmission is possible between 8251 and CPU by the data bus buffer.

Read / Write control logic:-

It is a control block for overall device.

It controls the overall working by selecting the operations to be done.

The operation selection depends upon input signals as :

The operation selection depends upon input signals as :

CS	C10	R0	WR	operation
1	x	x	x	Data bus 3-state
0	x	1	1	Data bus 3-state
0	1	0	1	Status $\rightarrow$ CPU
0	1	1	0	Control word
0	0	0	1	Data $\rightarrow$ CPU
0	0	1	0	Data $\leftarrow$ CPU

In this way, this unit selects one of the three registers - Data buffer register, Control register, Status register.

## Modem control (Modulator/demodulator)

A device converts analog signals to digital signal vice versa and helps the computers to communicate over telephone line or cable wires. The following are active-low pins of Modem.

- **DSR**: Data set Ready is an input signal.
- **CTS**: It is an input signal which controls the data transmit circuit. (*clear to send*)
- **DTR**: It is a output signal (data terminal ready)
- **RTS** :- It is an output signal which is used to set the status RTS. (*Request to send*)

② **Transmit buffer**:- This block is used for parallel to serial converter that receives a parallel byte for conversion into serial signal and further transmission on to the common channel.

- **TxD** :- It is an output signal, if its value is one (1), means transmitter will transmit the data.

③ **Transmit control** :- This block is used to control the data transmission with the help of followings pins

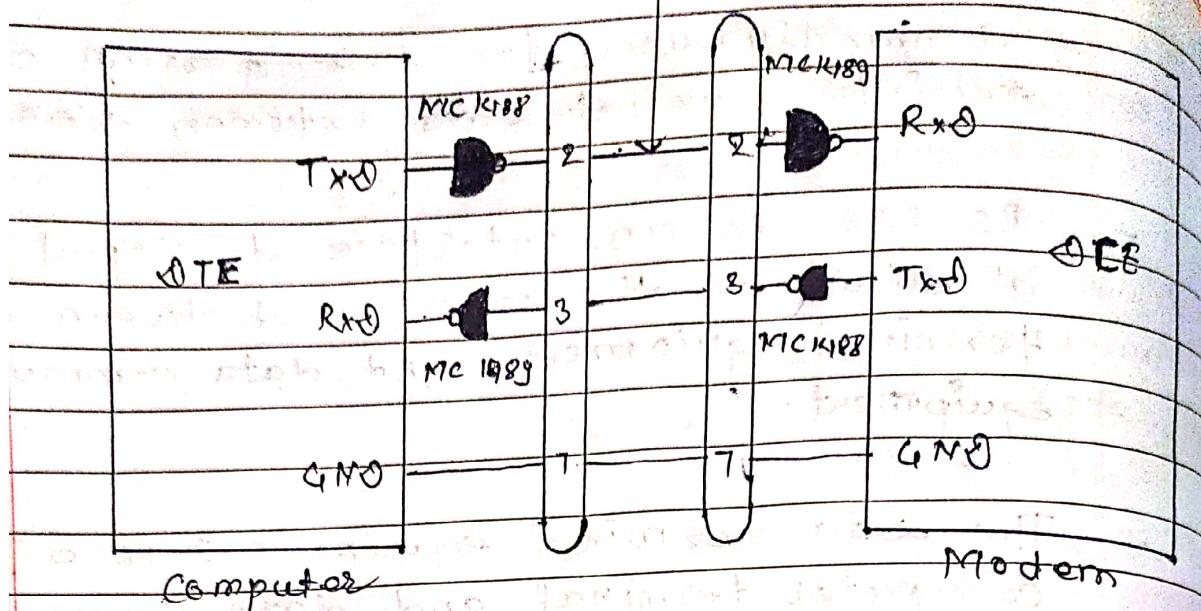
- **TXRDY** : It means transmitter is ready to transmit data character.

## RS-232

- RS 232 is a serial communication protocol commonly used for transferring and receiving serial data between two devices.
- RS 232 is an interface developed to standardized the interface between data terminal equipment and data communication equipment.
- The data terminal equipment is a computer terminal and data communication equipment is modems or controllers etc.
- It supports both synchronization and asynchronous data transmission.
- Many devices in the industries environment are still using RS-232 communication cable.
- It uses 25 pins or 9 pins where 9 pins standard doesn't use all signals i.e. data, control, timing and ground.
- It describes the voltage level, impedance levels, rise and fall time, maximum bit rate and maximum capacitance of the signals.
- It can send 20KBd for a distance of 50ft. voltage level are
  - logic + , 3V to -15V
  - logic low +3V to +15V
- Normally  $\pm 12V$  are used.

RS 232 cable.

पाठ्यालो  
Date \_\_\_\_\_  
Page \_\_\_\_\_



RS 232 pin configuration for 9 pin.

signal	description
RXD	Received data
TXD	Transmit data
CD	carrier detect
DTR	Data terminal ready
DSR	Data set ready.
RTS	Request to send.
CTS	clear to send
RI	Ringing
GND	Ground.

## Method of parallel data transfer

### 1) simple I/O:

For simple I/O, the buffer switch and latch switches, LEDs are always connected to the input output port. The device are always ready to send or receive data.

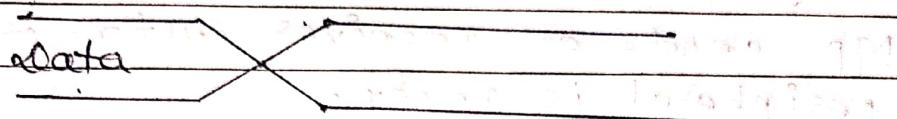


fig. Timing diagram of simple I/O .

### 2) strobe I/O

In this technique, microprocessor need to wait until the device is ready for the operation.

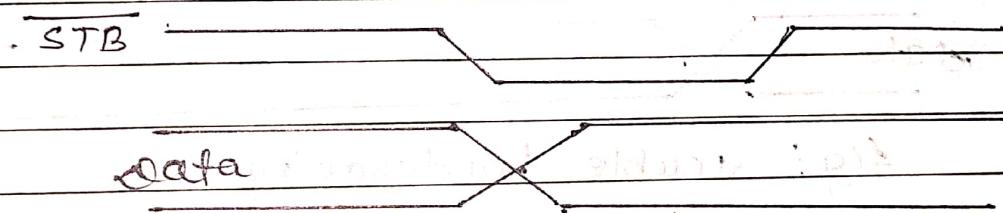
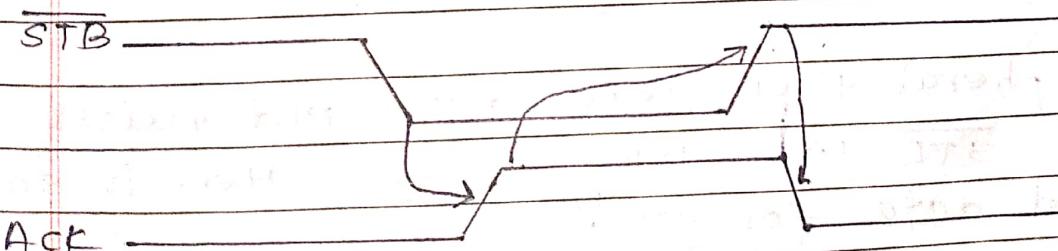


fig. Timing diagram of strobe I/O

### 3) single handshaking I/O (data transfer)



- The peripheral output some data and send STB signal to MP. "here is the data for you!".
- MP detects asserted STB signal, read the data and sends an acknowledge signal (ACK) to indicate data has been read and peripheral can send next data. "I got that one, send me another".
- MP sends or receives data when peripheral is ready..

#### 4. Double Handshaking :-

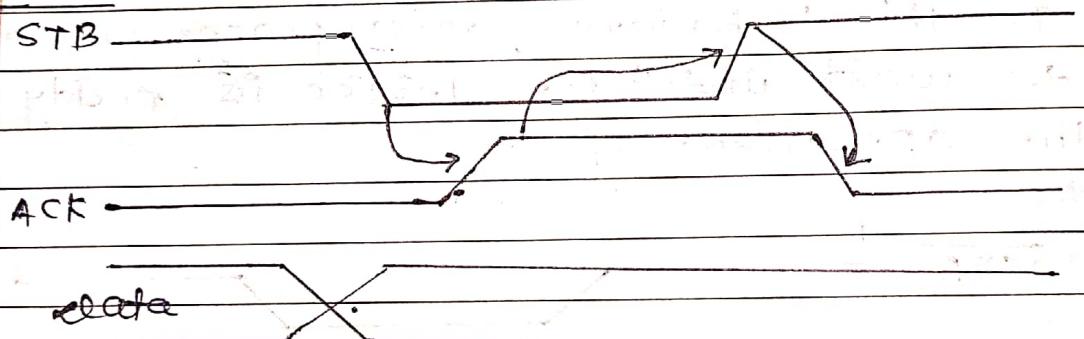


fig: double handshaking

- The peripheral asserts its STB line low to ask MP " Are you ready ?".
- The MP raises its ACK line high to say " I am ready".
- peripheral then send data and raises its STB line low to say " Here is some valid data for you!".
- MP then reads the data and drops its ACK line to say, " I have the data,