GUJARAT TECHNOLOGICAL UNIVERSITY

Mahatma Gandhi Institute Of Technical Education & Research Center, Navsari.

COMPUTER ENGINEERING DEPARTMENT

GTU

Subject Name : Microprocessor & Interfacing(3160712) **Faculty Name: Ami Patel**

Year: III **Semester: VI**

QUESTION BANK

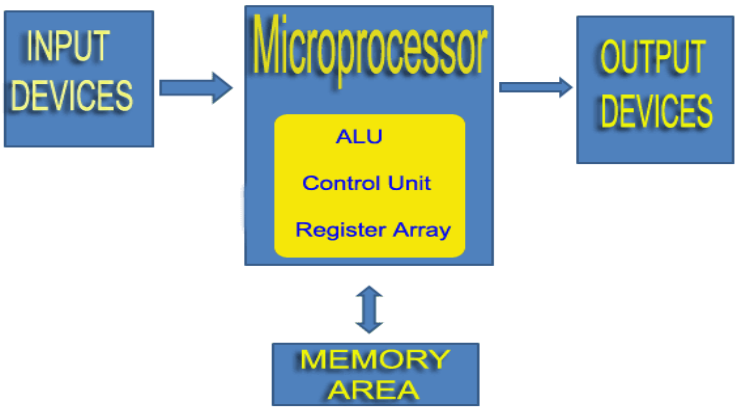
1. Explain block diagram of microprocessor

Computer's Central Processing Unit (CPU) built on a **single Integrated Circuit (IC)** is called a **microprocessor**.

Microprocessor is a hardware component of computer, and it works as brain of the computer system as well as **use in computer** because without using microprocessor, Computer like as plastic box.

A digital computer with one microprocessor which acts as a CPU is called microcomputer.

It is a programmable, multipurpose, clock -driven, register-based electronic device that reads binary instructions from a storage device called memory, accepts binary data as input and processes data according to those instructions and provides results as output.



A microprocessor consists of an ALU, control unit and register array. Where **ALU** performs arithmetic and logical operations on the data received from an input device or memory. It performs arithmetic operations such as addition, subtraction, and logical operations such as OR,AND, and Exclusive-OR .

Control unit controls the instructions and flow of data within the computer. It controls and executes the flow of data between the microprocessor, memory and peripherals.

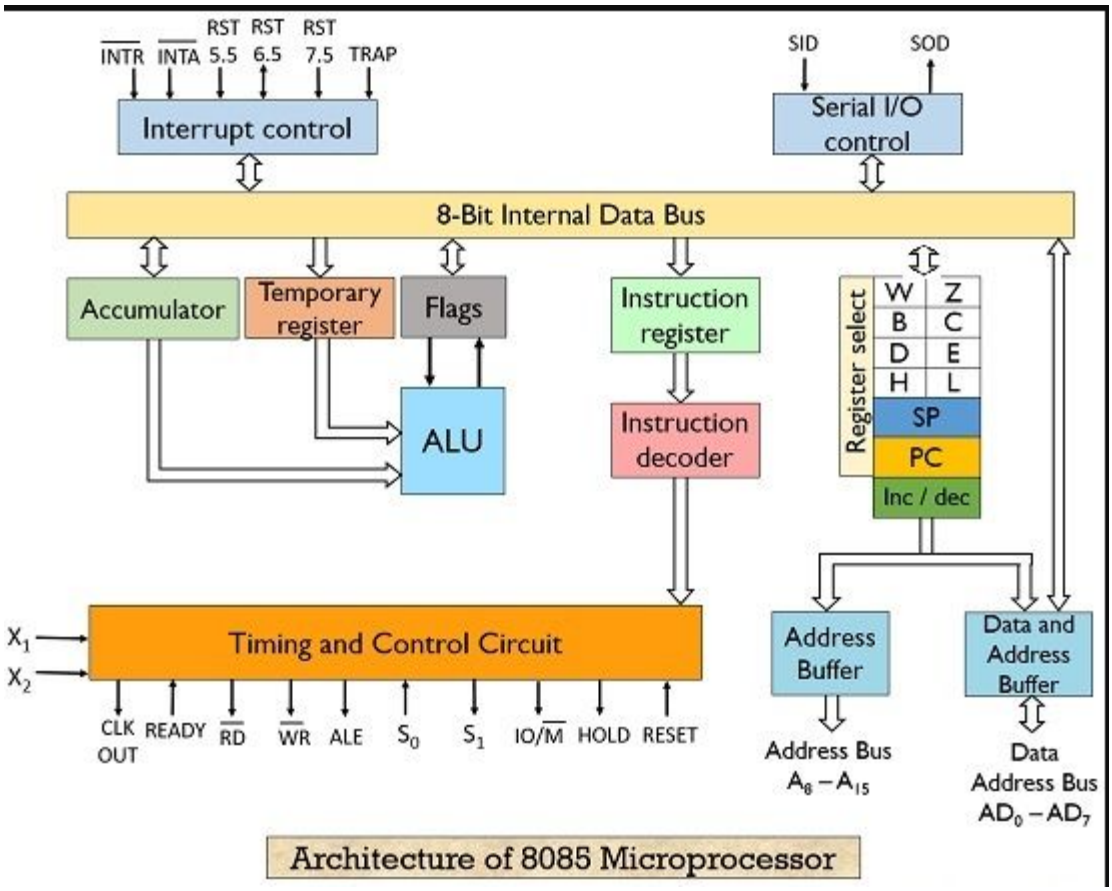
Registers are the small additional memory location which are used to store and transfer data and programs that are currently being executed Registers are identified by letters like B, C, D, E, H, L, and accumulator.

Working of Microprocessor

The microprocessor follows a sequence to execute the instruction: Fetch, Decode, and then Execute.

Initially, the instructions are stored in the storage memory of the computer in sequential order. The microprocessor fetches those instructions from the stored area (memory), then decodes it and executes those instructions till STOP instruction is met. Then, it sends the result in binary form to the output port. Between these processes, the register stores the temporary data and ALU (Arithmetic and Logic Unit) performs the computing functions.

2. Explain Architecture of 8085 microprocessor



Architecture of 8085 Microprocessor

Register Unit

General Purpose Data Register

8085 has six general purpose data registers to store 8-bit data.

These registers are named as B, C, D, E, H and L as shown in fig. 1.

The user can use these registers to store or copy a data temporarily during the execution of a program by using data transfer instructions.

These registers are of 8 bits but whenever the microprocessor has to handle 16-bit data, these registers can be combined as register pairs – BC, DE and HL.

There are two internal registers – W and X. These registers are only for internal operation like execution of CALL and XCHG instructions and not available to the user.

Program Counter (PC)

- 16-bit register deals with sequencing the execution of instructions.
- This register is a memory pointer.
- Memory locations have 16-bit addresses which are why this is a 16-bit register.
- The microprocessor uses this register to sequence the execution of the instructions.
- The function of the program counter is to point to the memory address from which the next byte is to be fetched.

When a byte (machine code) is being fetched, the program counter is incremented by one to point to the next memory location

Stack Pointer (SP)

- SP is also a 16-bit register used as a memory pointer.
- It points to a memory location in R/W memory, called the stack.
- The beginning of the stack is defined by loading 16-bit address in the stack pointer.

MUX/DEMUX unit

- This unit is used to select a register out of all the available registers.
- This unit behaves as a MUX when data is going from the register to the internal data bus.
- It behaves as a DEMUX when data is coming to a register from the internal data bus of the microprocessor.
- The register select will behave as the function selection lines of the MUX/DEMUX.

Address Buffer Register & Data/Address Buffer Register

- These registers hold the address/data, received from PC/internal data bus and then load the external address and data buses.

These registers actually behave as the buffer stage between the microprocessor and external system buses

Control Unit:

- The control unit generates signals within microprocessor to carry out the instruction, which has been decoded.
- In reality it causes connections between blocks of the microprocessor to be opened or closed, so that the data goes where it is required and the ALU operations occur.
- The control unit itself consists of three parts; the instruction registers (IR), instruction decoder and machine cycle encoder and timing and control unit.

Instruction Register

- This register holds the machine code of the instruction.
- When microprocessor executes a program it reads the opcode from the memory, this opcode is stored in the instruction register.

Instruction Decoder & Machine Cycle Encoder

- The IR sends the machine code to this unit.
- This unit, as its name suggests, decodes the opcode and finds out what is to be done in response of the coming opcode and how many machine cycles are required to execute this instruction.

Timing & Control unit

- The control unit generates signals within microprocessor to carry out the instruction, which has been decoded.
- In reality, it causes certain connections between blocks of the microprocessor to be opened or closed, so that the data goes where it is required and the ALU operations occur.

Arithmetic & Logical Unit:

- The ALU performs the actual numerical and logical operation such as ‘add’,

'subtract', 'AND', 'OR', etc.

- ALU uses data from memory and from accumulator to perform the arithmetic operations and always stores the result of the operation in accumulator.
- ALU consists of accumulator, flag register and temporary register.

Accumulator

- The accumulator is an 8-bit register that is a part of ALU.
- This register is used to store 8-bit data and perform arithmetical and logical operations.
- The result of an operation is stored in the accumulator.
- It is also identified as register A.

Flags register

- Flag register includes five flip-flops, which are set or reset after an operation according to the data conditions of the result in the accumulator and other registers.
- They are called zero (Z), carry (CY), sign (S), parity (P) and auxiliary carry (AC) flags; their bit positions in the flag register are shown in fig.
- The microprocessor uses these flags to set and test data conditions.

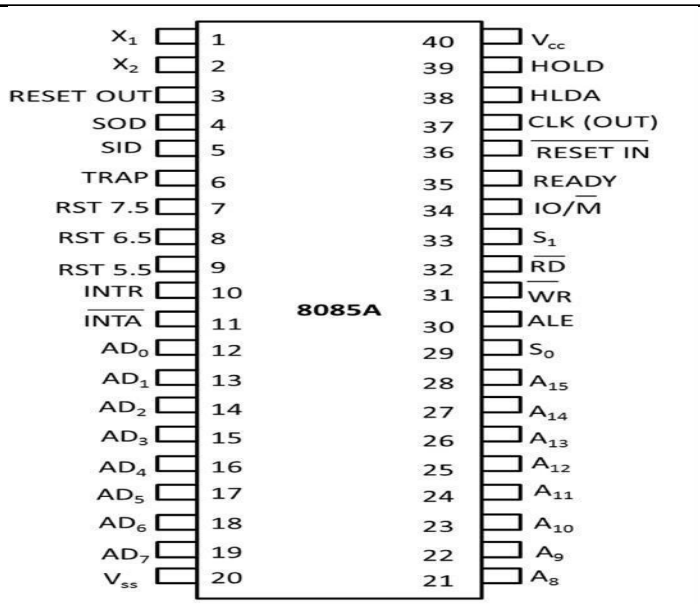
Interrupt Control

- The interrupt control unit has 5 interrupt inputs TRAP, RST 7.5, RST 6.5, RST 5.5 & INTR and one acknowledge signal INTA.
- It controls the interrupt activity of 8085 microprocessor.

Serial IO control

- 8085 serial IO control provides two lines, SOD and SID for serial communication.
- The serial output data (SOD) line is used to send data serially and serial input data line (SID) is used to receive data serially.

3. Explain pin diagram of 8058 microprocessor



- All signals can be classified into six groups:
  1. Address Bus
  2. Data Bus
  3. Control & Status Signals
  4. Power Supply & Frequency signals
  5. Externally initiated signals
  6. Serial I/O Ports

1) Address Bus (pin 12 to 28)

- A_{15} - A_8 are unidirectional and are used to carry high-order address of 16-bit address.
- AD_7 - AD_0 are used for dual purpose. Data and address

2) Data Bus/ Multiplexed Address (pin 12 to 19)

- Signal lines AD7-AD0 are bidirectional and serve dual purpose.
- They are used as low-order address bus as well as data bus.

The low order address bus can be separate from these signals by using a latch(ALE)

3.) Control & Status Signals

Two Control Signals

1) RD' (Read-pin 32)

✔ This is a read control signal (active low)

✔ This signal indicates that the selected I/O or Memory device is to be read & data are available on data bus.

2) WR' (Write-pin 31)

✔ This is a write control signal (active low)

✔ This signal indicates that the selected I/O or Memory device is to be write.

Three Status Signals

1) S1(pin33)

2) S0(pin29)

S_1 and S_0 status signals can identify various operations, but they are rarely used in small systems.

S_1	**S_0**	**Mode**
0	0	HLT
0	1	WRITE
1	0	READ
1	1	OPCODE FETCH

3) IO/M' (pin 34)

✔ This is a status signal used to differentiate I/O and memory operation

✔ When it is high, it indicates an I/O operation

✔ When it is low, it indicates a memory operation

- ALE (Address Latch Enable-Pin 30)
- This is positive going pulse generated every time the 8085 It indicates that the bits on AD7-AD0 are address bits

[4]Power Supply & Frequency Signal

- V_{cc}-->Pin no. 40, +5V Supply
- V_{ss} --> Pin no.20, Ground Reference
- X1, X2 --> Pin no.1 & 2, Crystal Oscillator is connected at these two pins. The frequency is internally divided by two;
- CLK (OUT) --> Clock output. Pin No.37: This signal can be used as the system clock .

[5]Externally Initiated Signals including Interrupts

- Pin no 10:INTR (Input) --> Interrupt Request. It is used as general purpose interrupt
- Pin no 11: INTA' (Output) --> Interrupt Acknowledge. It is used to acknowledge an interrupt.
- Pin no 7,8,9 : RST7.5, RST6.5, RST5.5 (Input) --> Restart Interrupts.
  - They have higher priorities than INTR interrupt.
  - Among these 3 interrupts, the priority order is RST7.5, RST6.5, RST5.5
- Pin no 6:TRAP (Input) --> This is a non maskable interrupt & has the highest priority.
- Pin 39:HOLD (Input) --> This signal indicates that a peripheral such as DMA Controller is requesting the use of address & data buses

Pin no 38: HLDA (Output) --> Hold Acknowledge. This signal acknowledges the HOLD request

- Pin no 35:READY (Input) --> This signal is used to delay the microprocessor read or write cycles until as low- responding peripheral is ready to send or accept data.
- Pin no 36:RESET IN' (Input) --> When the signal on this pin goes low, the Program Counter is set to zero.
- Pin no 3: RESET OUT (Output) --> this signal indicates that microprocessor

is being reset.

[6]Serial I/O Ports

- Two pins for serial transmission

  1) SID (Serial Input Data-pin 5)

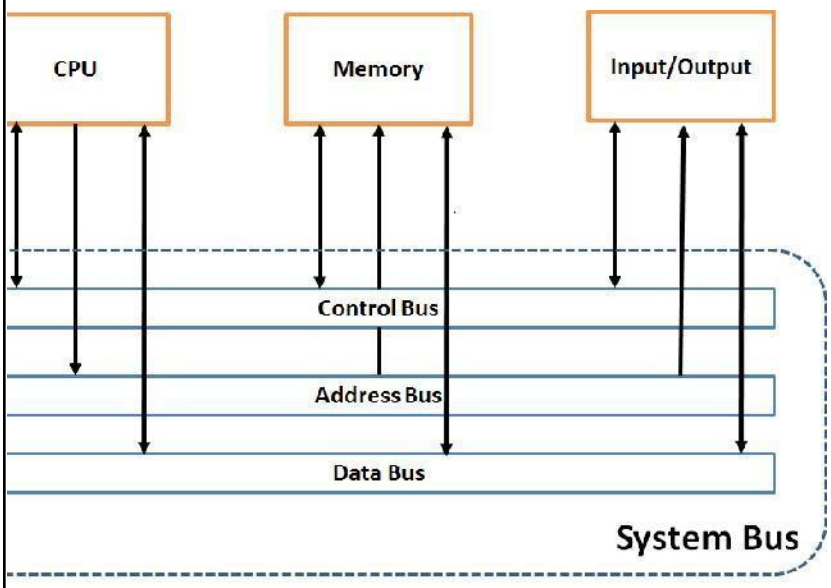
  2) SOD (Serial Output Data-pin 4)

- In serial transmission, data bits are sent over a single line, one bit at a time.

4. Explain bus organization in 8085 microprocessor

System bus (data, address and control bus).

- This network of wires or electronic pathways is called the 'Bus'.
- A system bus is a single computer bus that connects the major components of a computer system.
- It combines the functions of a data bus to carry information, an address bus to determine where it should be sent, and a control bus to determine its operation.



The technique was developed to reduce costs and improve modularity

Address Bus

- It is a group of wires or lines that are used to transfer the addresses of Memory or I/O devices.
- It is unidirectional.
- The width of the address bus corresponds to the maximum addressing capacity of the bus.
- The maximum address capacity is equal to two to the power of the number of lines present (2^lines).

Data Bus

- It is used to transfer data within Microprocessor and Memory/Input or Output devices.
- It is bidirectional as Microprocessor requires to send or receive data.
- Each wire is used for the transfer of signals corresponding to a single bit of binary data.

As such, a greater width allows greater amounts of data to be transferred at the same time

Control Bus

- Microprocessor uses control bus to process data, i.e. what to do with the selected memory location.
- Some control signals are Read, Write and Opcode fetch etc.
- Various operations are performed by microprocessor with the help of control bus.

5. Explain opcode fetch cycle of 8085 and its timing diagram

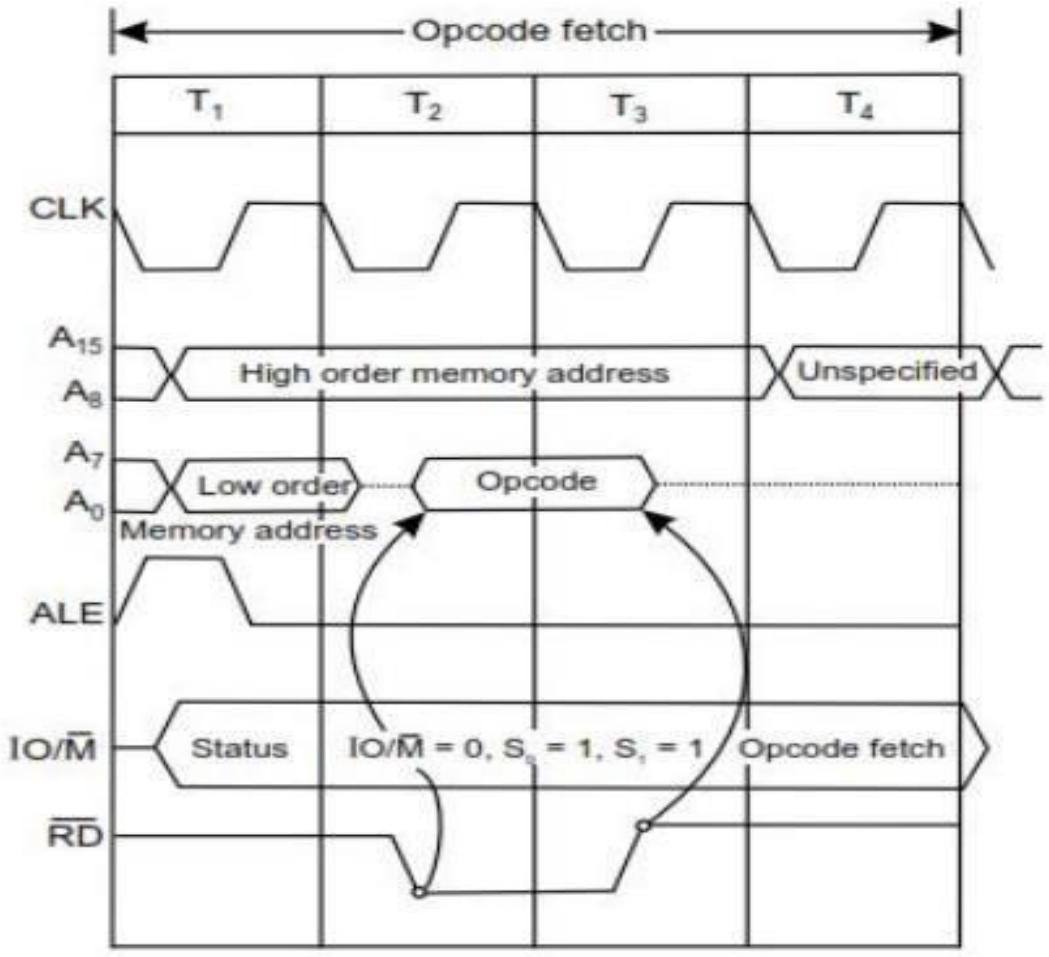


Fig 1.8 Opcode fetch machine cycle

Each instruction of the processor has one byte opcode.

The opcodes are stored in memory. So, the processor executes the opcode fetch machine cycle to fetch the opcode from memory.

Hence, every instruction starts with opcode fetch machine cycle.

The time taken by the processor to execute the opcode fetch cycle is 4T.

In this time, the first, 3 T-states are used for fetching the opcode from memory and the remaining T-states are used for internal operations by the processor.

RD (low active) – This is an active low signal and simply tells us whether it is a read operation when it is low..

WR (low active) – This is an active low signal and simply tells us whether it is a write operation when it is low.

ALE stands for Address Latch Enable. This output signal given by the microprocessor tells us whether the AD0-AD7 is carrying the address or is available for data transfer.
ALE = 0 (AD0-AD7 is available for data transfer)

ALE = 1 (AD0-AD7 is carrying the lower eight bits of the address)

In the T1 state, the microprocessor send the low byte address on AD0-AD7 lines and high byte address on A8 to A15 lines. ALE is send high to enable the address latch. The other control signals IO/ $\overline{M}$=0,S0=1, S1=1

2nd T stat

ALE signal goes low during T2. By this time, 8085 expects that the lower address bits are latched, and AD0-AD7 is free to be used as a data bus.

- At the beginning of the second T state, RD goes low, indicating that the read process has started. Meanwhile, higher address bits are present in A8-A15, and lower address bits are expected to be latched.
- As RD goes low, the opcode (eight bits) is loaded into the data bus AD0-AD7.

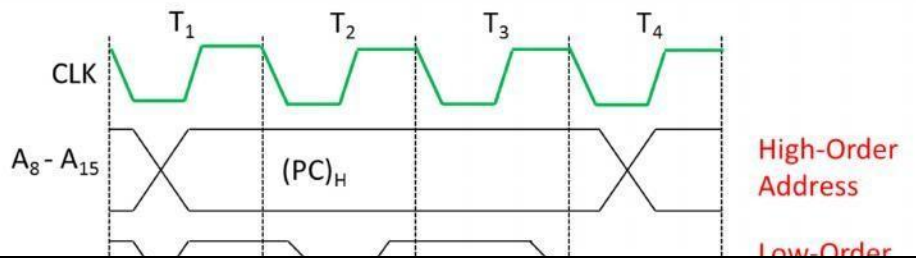
3rd T state

- The opcode loaded on the data bus is present there until the middle of the third T state.
- During the third T state, RD goes up, indicating that the read operation is completed and 'the opcode is fetched' and placed in the instruction register.
- The data on the data bus and the higher address bits on A8-A15 exist until the middle of this T state.

In T_4 state, the microprocessor decodes the opcode and based on the instruction it decides whether to T_0 state T_5 or to enter state T_1 of the next machine cycle (M_2).

6. Explain memory read and Write cycle of 8085 and its timing diagram

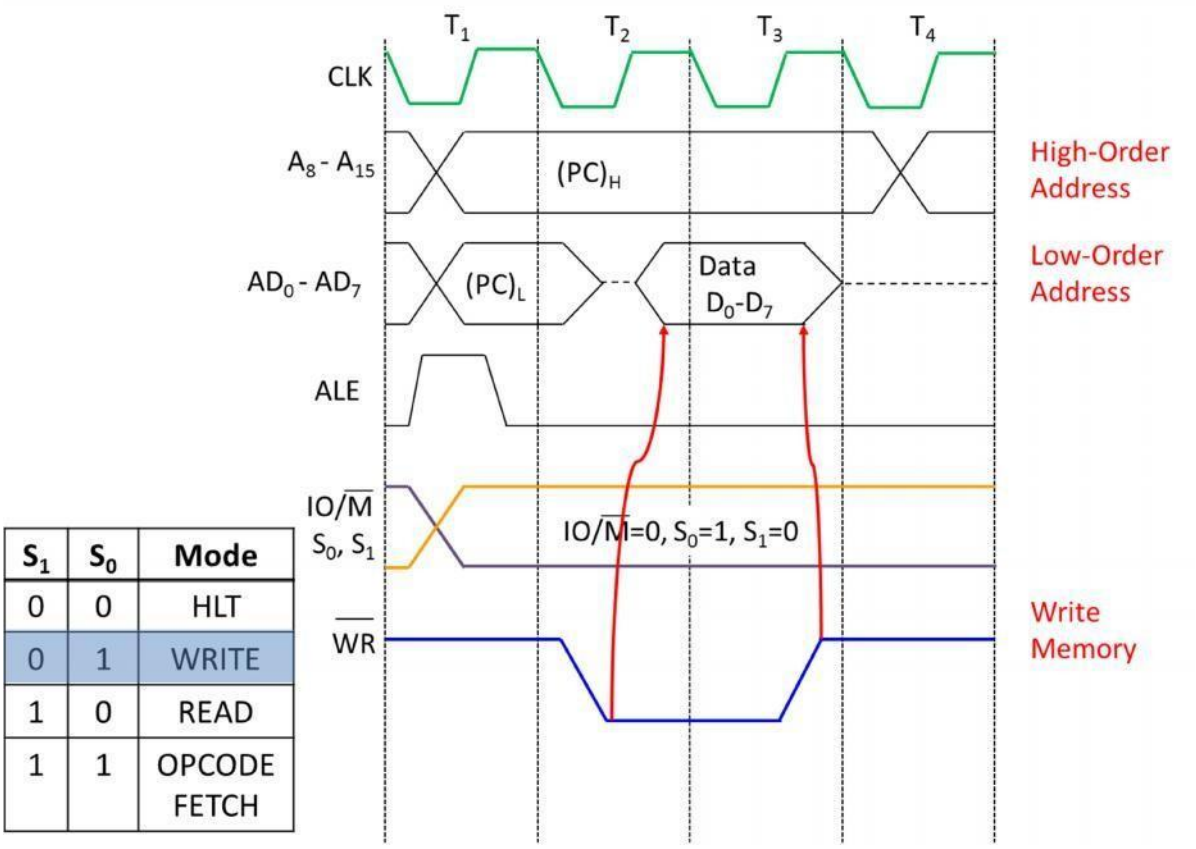
Memory Read Cycle



- It is used to fetch one byte from the memory.
- It requires 3 T-States.
- It can be used to fetch operand or data from the memory.
- During T1, A8-A15 contains higher byte of address. At the same time ALE is high. Therefore Lower byte of address A0-A7 is selected from AD0-AD7.
- Since it is memory ready operation, IO/M (bar) goes low.
- During T2 ALE goes low, RD (bar) goes low. Address is removed from AD0-AD7 and data D0-D7 appears on AD0-AD7.

During T3, Data remains on AD0-AD7 till RD (bar) is at low signal

Memory Write Cycle

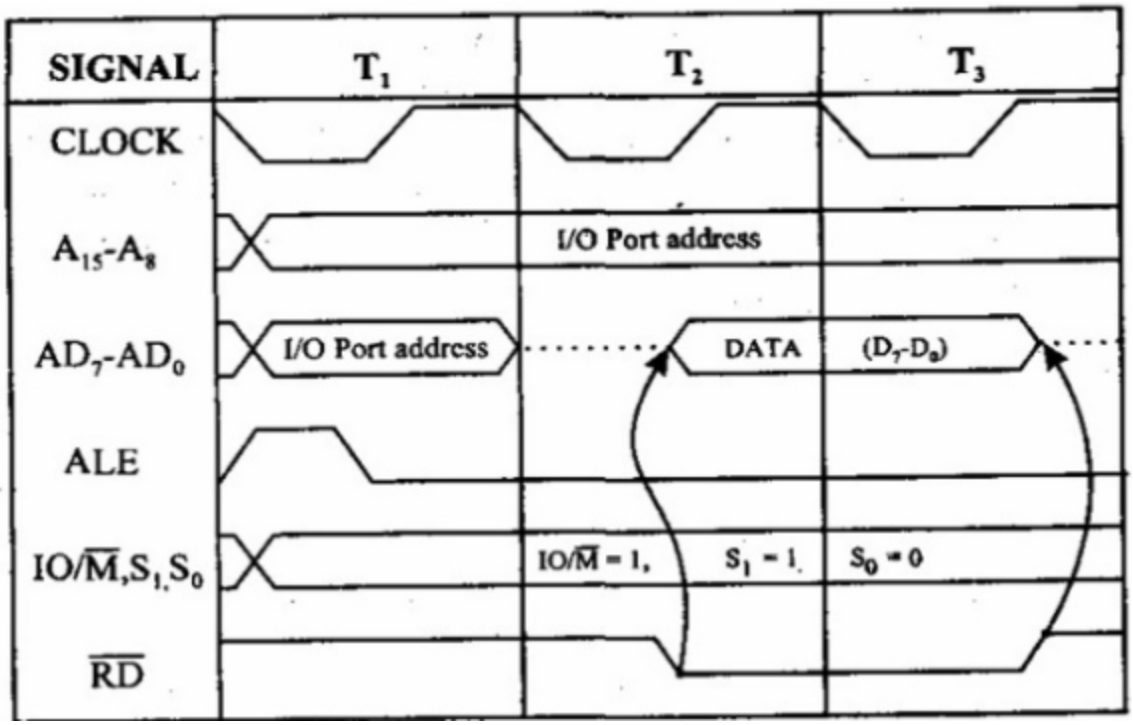


S_1	S_0	Mode
0	0	HLT
0	1	WRITE
1	0	READ
1	1	OPCODE FETCH

- It is used to send one byte into memory.
- It requires 3 T-States.
- During T1, ALE is high and contains lower address A0-A7 from AD0-AD7.
- A8-A15 contains higher byte of address.
- As it is memory operation, IO/M (bar) goes low.
- During T2, ALE goes low, WR (bar) goes low and Address is removed from AD0-AD7 and then data appears on AD0-AD7.

- Data remains on AD0-AD7 till WR (bar) is low.

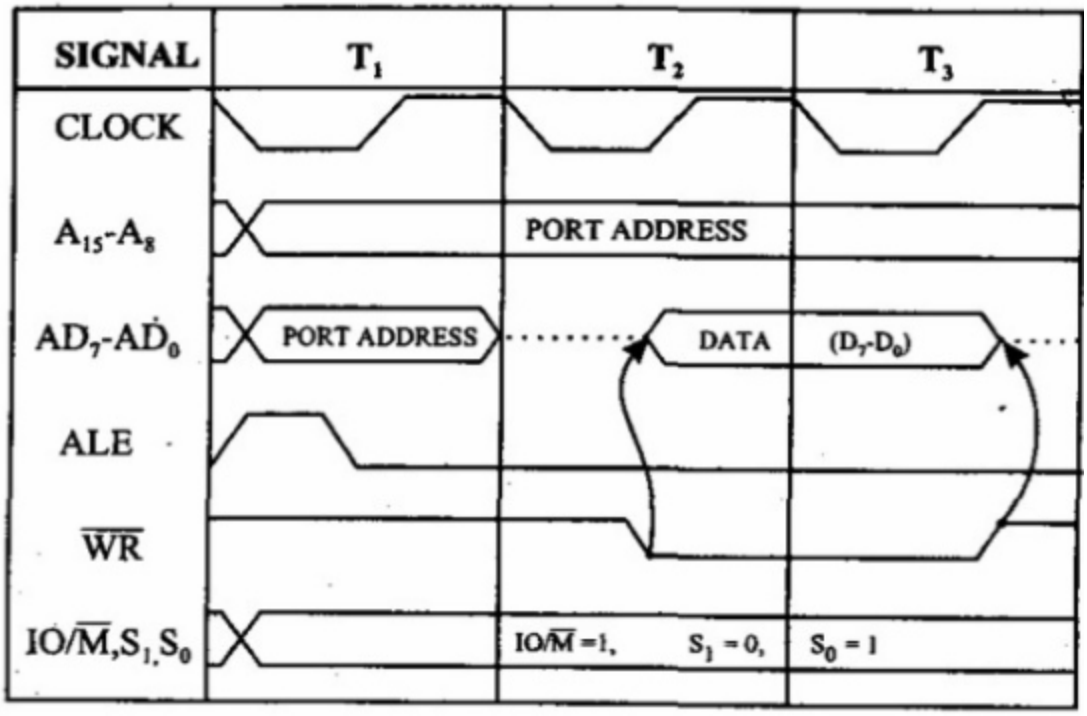
7. Explain i/o read and write cycle of 8085 and its timing diagram



It is used to fetch one byte from an IO port.
It requires 3 T-States.
During T1, The Lower Byte of IO address is duplicated into higher order address bus A8-A15.

ALE is high and AD0-AD7 contains address of IO device.
IO/M (bar) goes high as it is an IO operation.
During T2, ALE goes low, RD (bar) goes low and data appears on AD0-AD7 as input from IO device.
During T3 Data remains on AD0-AD7 till RD(bar) is low.

IO Write Operation:

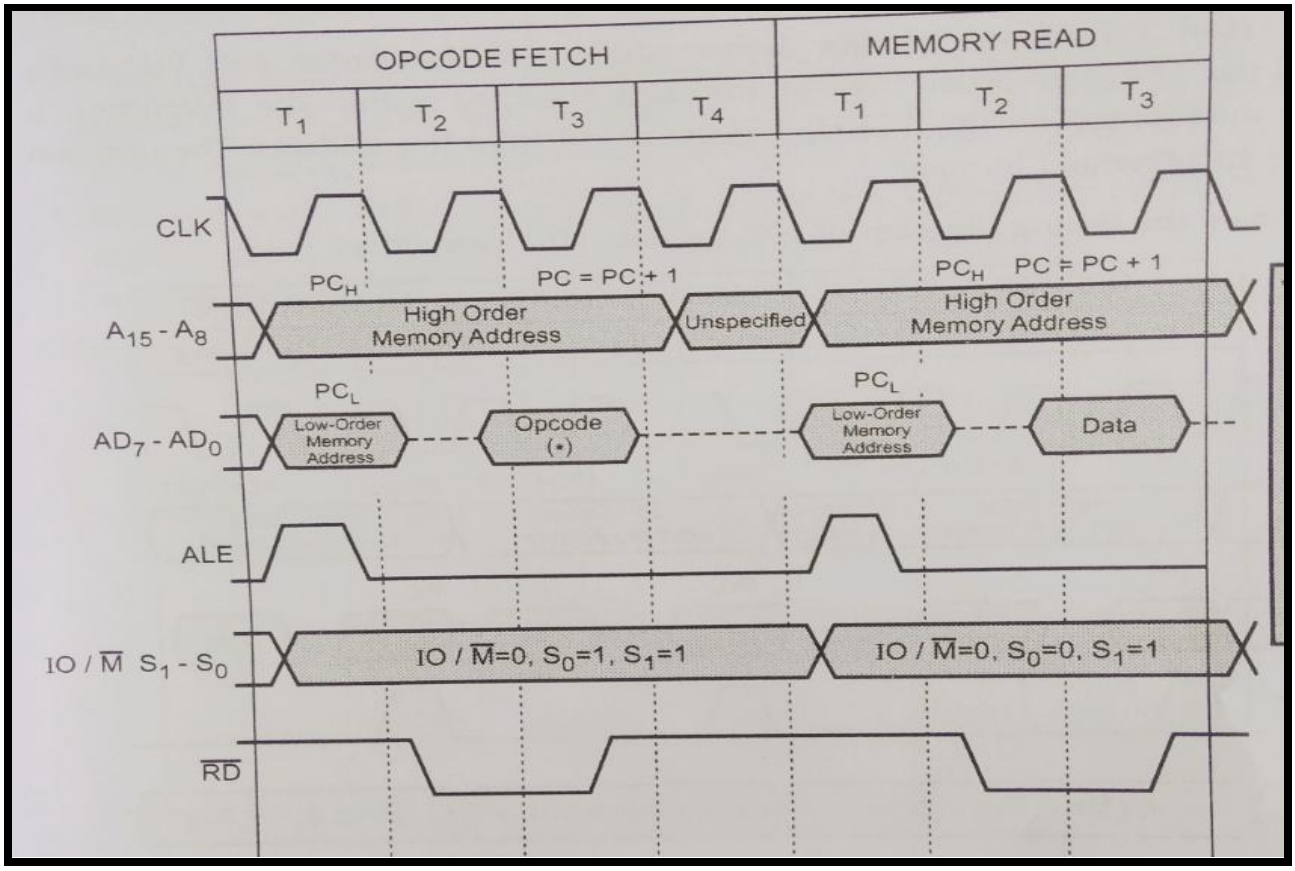


It is used to writ one byte into IO device.
It requires 3 T-States.
During T1, the lower byte of address is duplicated into higher order address bus A8-A15.
ALE is high and A0-A7 address is selected from AD0-AD7.
As it is an IO operation IO/M (bar) goes low.
During T2, ALE goes low, WR (bar) goes low and data appears on AD0-AD7 to write data into IO device.
During T3, Data remains on AD0-AD7 till WR(bar) is low.

8. Draw a timing diagram for MVI A, data instruction.

These instructions directly load a specified register with a data byte specified within the instruction. They require the following machine cycles.
It requires 7 T-States 4 for opcode fetch and 3 for memory read

1. Opcode fetch : Program counter gives the memory address on low-order and high-order address bus. This machine cycle is required for reading the opcode into the microprocessor and decode it. Program counter is incremented by one.
2. 2. Memory read : This machine cycle reads the data from addressed memory location into specified register of the microprocessor.



9. Draw a timing diagram for MOV A, B instruction

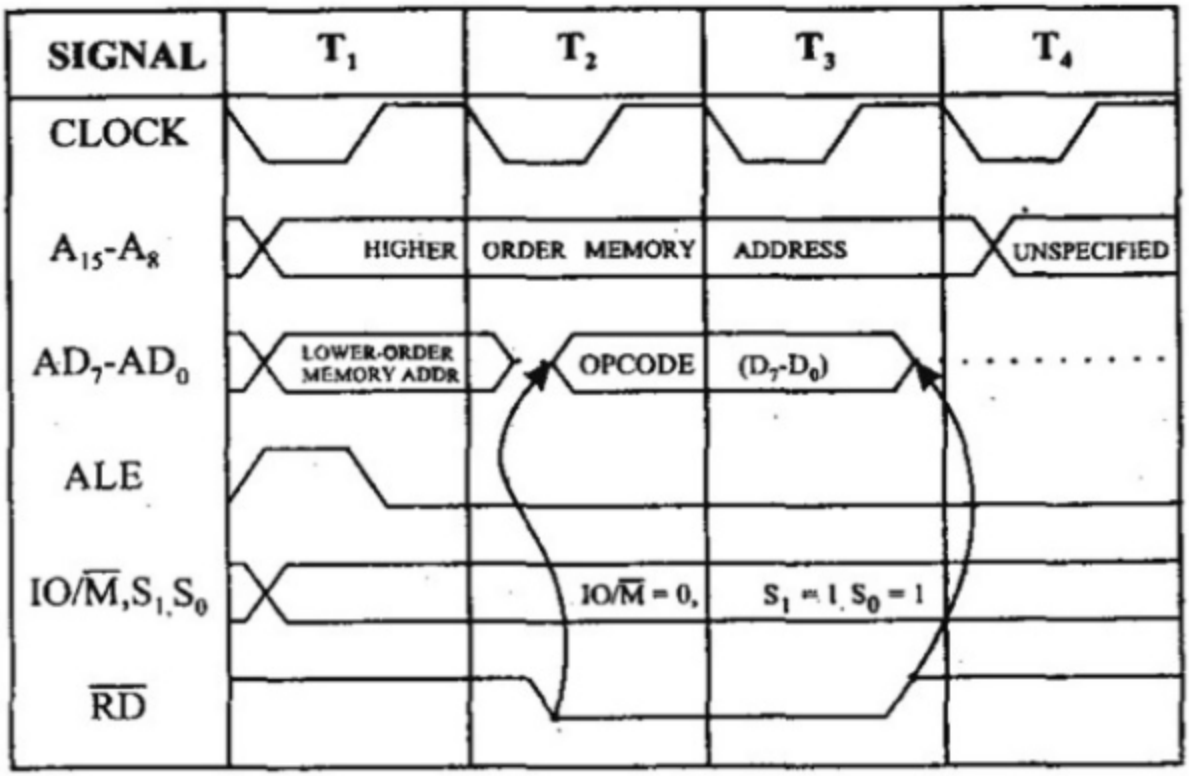
It is same like opcode fetch
In this instruction copies data from source register Rs, to the

destination register Rd

The source register Rs, and destination register Rd, can be any general purpose register like A, B, C, D, E, H or L. The contents of source register remain unchanged.

Example MOV B,C This instruction will copy the contents of register C to register B. The contents of register C remain unchanged.

Suppose B = 20H, C = 10H and the instruction MOV B, C is executed. After the execution B = 10H and C = 10H



10. Explain data transfer group of instruction set in 8085

1.MOV Rd,Rs
2. MOV R,M
3.MOV M,R
4. MVI R,data
5.MVI M, data
6.LXI Rp,16 bit data
7.LDA
8.STA add
9.HLDA address
10. SHLDA address
11. LDAX Rp
12. XCHG

1.MOV Rd,Rs

In this instruction copies data from source register Rs, to the destination register RdThe source register Rs, and destination register Rd, can be any general purpose register like A, B, C, D, E, H or L. The contents of source register remain unchanged.

Example MOV B,C This instruction will copy the contents of register C to register B. The contents of register C remain unchanged.

Suppose B = 20H, C = 10H and the instruction MOV B, C is executed. After the execution B = 10H and C = 10H

2. MOV R,M

This instruction copies data from memory M to register R. The term M specifies the HL memory pointer. The contents of HL register pair are used as the address of the memory contents of that memory location the content of memory location are transferred to the specified register R can be any general purpose register like A, B, C, D, E, register R. H or L.

Example MOV C, M.

• This instruction will copy the data from the memory location pointed by HL register pair to C register.

• Let the contents of the HL register pair be F000H, register C = 20H. At the address F000H : 10H is stored. The HL register pair contents are used as address. i.e. HL = F000H.

. The contents of memory location FOOOH are copied to the C register. So contents of C register will change from 20H to 10H.

MOV M,R

This instruction will copy the data from the register to memory M.

The specified register may be any general purpose register A, B,C,D, E, H or L.

The contents of the specified register remains unchanged.

Example MOV M,C

Let the contents of HL pair are E200 H, register C = 20H, at addressE200 : 10H is stored. On the instruction MOV M, C the data is transferred from C register to memory.The contents of the register C are copied to memory location E200 H, so contents of memory location C200 H will change from 10H to 20H.

4. MVI R,data

This instruction moves the 8 bit immediate data to the specified register. The data is specified within the instruction.

It is a two byte instruction, so first byte of instruction will be OPCODE and second byte will be 8 bit data. The register R may be any general purpose register like A, B, C, D, E,H or L.

Example: MVI D,07 H

This instruction will load 07 H in register D.

Let the contents of register D = 10H. Then after the execution of instruction MVI D, 07H the contents of register D will be change form 10H to 07H.

5. MVI M, data

his instruction moves immediate data to memory. The HL register pair is used as memory

Example : MVI H, 10H

MVI L,00H

MVI M,20H

10H is transferred to H register.

00H is transferred to L register.

20H is transferred to memoryH

When the instruction MVI M, 20H is executed the data 20H will be stored in the memory location addressed by the HL register pair i.e. 1000H. Fig. 6.6.1 shows the MVI M, 20H instruction.

6.LXI Rp,16 bit data

This instruction loads 16-bit data specified within the instruction to the specified register pair or stack pointer. In the instruction only high order register is specified for register pair. i.e. if HL pair is to be loaded only H register will be specified in the instruction. The register pair R, can be BC, DE, HL register pairs or the stack pointer SP.

Example: LXI H,2030H

LXI SP, 7FFFH

Load HL pair with 2030 H. 20H will be loaded in the H register and 30H in the L register.

This instruction will load stack pointer SP with 7FFF H.

7.LDA

Load accumulator direct from memory. This instruction copies the contents of the memory location whose address is specified in the instruction to the accumulator.

The contents of the memory location remain unchanged.

Example :LDA 5820H

This instruction will load the accumulator with the contents of memory location 5820H.

Let initially A = F0 H, contents of memory location 5820H = 15H. Then after the execution of instruction LDA 5820H, the accumulator will be loaded with 15H. The contents of accumulator will change from F0H to 15H

8.STA add

Store accumulator direct to memory. This instruction will store the contents of the accumulator to the memory location specified in the instruction. The contents of the memory location remain unchanged It is a 3 byte instruction. The first byte is the opcode, second byte is lower order address and the third byte is higher order address

STA 5820

This instruction will store the contents of accumulator at location 5820H

9.LHLD address

Load HL pair directly from memory.

This instruction loads the contents of the memory location to the H and L registers. The address of memory is specified along with 0 the instruction.

The contents of memory location whose address is specified in the instruction are transferred to L register and the contents of the next memory location i.e. (address + 1) to the H register. This instruction is used to load the H and L registers from memory

It is a 3 byte instruction. The first byte is the opcode, second byte is the lower order address and the third byte is the higher order address

Example: LHLD 4000H

Load HL pair from memory locations 4000H and 4001H.

Let H = 05 H, L = 04 H, at memory locations 4000 H and 4001 H the data 20H, 30H is stored. The instruction LHLD will load the contents of memory location

4000 H to the L register, and the contents of memory location 4001 H to the H register.

So the contents of register L will change from 04 to 20 H and contents of register H will change from 05 H to 30 H.

10. SHLD address

Store HL pair direct in memory.

This instruction copies the contents of registers H and I to the memory locations. The address of memory location is specified along with the instruction.The contents of L register are stored at the memory location whose address is specified and the contents of the H register to the (address + 1) location.

It is a three byte instruction. The first byte is the opcode, second byte is the lower order address and the third byte is the higher order address.

Example: SHLD 4000H

Store HL pair to memory locations 4000 and 4001. Let H = 05H, L = 04H, at memory locations 4000H and 4001H the

data 20H and 30H is stored and the instruction SHLD 4000 H is executed the contents of register L are copied to memory location 4000 H and contents of register H are copied memory location 4001 H.

11. LDAX Rp

This instruction copies the contents of the memory location to the accumulator The address of memory location is given by R, register pair specified along with the instruction. The register pair R, can be BC or DE only.The contents of the memory location remain unchanged.

Example:LDAX B

This instruction will load accumulator with the contents of memory location whose address is given by the BC register pair Let A = 1F H, B = 20H, C = 25H, at memory location 2025 : 56H is

stored Then after the execution of instruction LDAX B, the accumulator will be loaded with the contents of memory location 2025 i.e. 56 H

12. STAX Rp

This instruction copies the contents of accumulator to memory Location

The address of the memory location is given by the Rregister pair specified in the instruction The register pair R, can be a valid register pair like BC or DE only. The contents of accumulator remain unchanged.

Example:STAX B

This instruction will store the contents of accumulator to the memory location, whose address is given by the BC register pair Let A = 1 F H, B = 26H, C = 08H, at memory location 2608 10 is = stored.

13. XCHG: it exchange the content of H with D and L with E
Example: XCHG
Let h =12H; L=11H, D=30H and the instruction XCHG is executed

11. Explain Arithmetic group of instruction set in 8085

ADD R	This instruction adds the contents of register R and accumulatorThe result is stored in the accumulator. Let A = 47 H,C = 51 H and instruction ADD C is executed. A 0100 0111 = 47 C 0101 0001 = 51 1001 1000 = 98
ADD M	Add memory location contents to accumulator contents of memory location addressed by HL pair are added to the contents of the accumulator. The result is stored in the accumulator The contents of memory location remain unchanged. Let A = 15 H, H COH, L 05 H, at memory location C005:02 H is stored.
ADC R	This instruction adds the contents of the register R to the contents of accumulator with carry. The result is stored in accumulator Let A= 3 H, H = 20 H, CY = 1 and the instruction. ACD H A=A+H+Cy 3+20+1
ADC M	The contents of the register or memory & M the Carry flag are added to the contents of the accumulator and the result is stored in the accumulator.

ADI data	The 8-bit data is added to the contents of the accumulator and the result is stored in the accumulator.
ACI data	The 8-bit data and the Carry flag are added to the contents of the accumulator and the result is stored in the accumulator
DAD Rp	The 16-bit data of the specified register pair are added to the contents of the HL register.
SUB R	The contents of the register are subtracted from the contents of the accumulator, and the result is stored in the accumulator.
SUB M	The contents of the memory are subtracted from the contents of the accumulator, and the result is stored in the accumulator
SBB M	The contents of the memory & M the Borrow flag are subtracted from the contents of the accumulator and the result is placed in the accumulator.
SUI data	The 8-bit data is subtracted from the contents of the accumulator & the result is stored in the accumulator.
SBI data	he contents of register H are exchanged with the contents of register D, and the contents of register L are exchanged with the contents of register E.
DAA	The contents of the accumulator are changed from a binary value to two 4-bit BCD digits. If the value of the low-order 4-bits in the accumulator is greater than 9 or if AC flag is set, the instruction adds 6 to the low-order four bits. If the value of the high-order 4-bits in the accumulator is greater than 9 or if the Carry flag is set, the instruction adds 6 to the high-order four bits. **Example** − DAA
INR R	The contents of the designated register or the memory are incremented by 1 and their result is stored at the same place. Example − INR

INR M	The contents of the designated register or memory are decremented by 1 and their result is stored at the same place. **Example** − DCR K
DCR R	It decrement register content by 1and result is store in same register
DCR M	It decrement content of memory location by 1. And result is store in same memory location.
INX Rp	It increment content of register pair by 1 and result is store in same register pair
DCX Rp	It decrement content of register pair by 1 and result is store in same register pair

12. Explain Logical group of instruction set in 8085

CMP R	The contents of the register compared with the contents of the accumulator.
CMP M	The contents of the memory compared with the contents of the accumulator
CPI	The second byte data is compared with the contents of the accumulator.
ANA R	The contents of the accumulator are logically AND with M the contents of the register or memory, and the result is placed in the accumulator
ANA M	The contents of the accumulator are logically AND with M the contents of the register or memory, and the result is placed in the accumulator
ANAI	The contents of the accumulator are logically AND with the 8-bit data and the result is placed in the accumulator.
XRA	The contents of the accumulator are Exclusive OR with M the contents of the register or memory, and the result is placed in the accumulator.
	The contents of the accumulator are Exclusive OR with M the contents of the register or memory, and the result is placed in the accumulator

XRI	The contents of the accumulator are Exclusive OR with the 8-bit data and the result is placed in the accumulator.
ORA	The contents of the accumulator are logically OR with M the contents of the register or memory, and result is placed in the accumulator.
ORI	The contents of the accumulator are logically OR with the 8-bit data and the result is placed in the accumulator
RLC	Each binary bit of the accumulator is rotated left by one position. Bit D7 is placed in the position of D0 as well as in the Carry flag. CY is modified according to bit D7
RRC	Each binary bit of the accumulator is rotated right by one position. Bit D0 is placed in the position of D7 as well as in the Carry flag. CY is modified according to bit D0.
RAL	Each binary bit of the accumulator is rotated left by one position through the Carry flag. Bit D7 is placed in the Carry flag, and the Carry flag is placed in the least significant position D0. CY is modified according to bit D7
RAR	Each binary bit of the accumulator is rotated right by one position through the Carry flag. Bit D0 is placed in the Carry flag, and the Carry flag is placed in the most significant position D7. CY is modified according to bit D0.
CMA	The contents of the accumulator are complemented. No flags are affected.
CMC	The Carry flag is complemented. No other flags are affected.
STC	Set Carry

13. Explain Branch operation group of instruction set in 8085

JMP
Jumps to the address
JMP 2050

Conditional Jump Instructions: Transfers the program sequence to the described memory address only if the condition in satisfied

OPCODE	EXPLANATION	EXAMPLE
JC	Jumps to the address if carry flag is 1	JC 2050
JNC	Jumps to the address if carry flag is 0	JNC 2050
JZ	Jumps to the address if zero flag is 1	JZ 2050
JNZ	Jumps to the address if zero flag is 0	JNZ 2050
JPE	Jumps to the address if parity flag is 1	JPE 2050
JPO	Jumps to the address if parity flag is 0	JPO 2050
JM	Jumps to the address if sign flag is 1	JM 2050
JP	Jumps to the address if sign flag 0	JP 2050

PCHL: content of H and L register transferred to program counter

CALL **Unconditionally calls** **CALL 2050**

Conditional Call Instructions: Only if the condition is satisfied, the instructions executes.

OPCODE	**EXPLANATION**	**EXAMPLE**
CC	Call if carry flag is 1	CC 2050
CNC	Call if carry flag is 0	CNC 2050
CZ	Calls if zero flag is 1	CZ 2050
CNZ	Calls if zero flag is 0	CNZ 2050
CPE	Calls if parity flag is 1	CPE 2050
CPO	Calls if parity flag is 0	CPO 2050

CM	Calls if sign flag is 1	CM 2050
CP	Calls if sign flag is 0	CP 2050

Return Instructions – The return instruction transfers the program sequence from the subroutine to the calling program. Return instructions are 2 types: Unconditional Jump Instructions and Conditional Jump Instructions**.**

Unconditional Return Instruction: The program sequence is transferred unconditionally from the subroutine to the calling program

RET Return from the subroutine unconditionally: RET

Conditional Return Instruction: The program sequence is transferred unconditionally from the subroutine to the calling program only is the condition is satisfied.

OPCODE	OPERAND	EXPLANATION	EXAMPLE
RC		Return from the subroutine if carry flag is 1	RC
RNC		Return from the subroutine if carry flag is 0	RNC
RZ		Return from the subroutine if zero flag is 1	RZ
RNZ		Return from the subroutine if zero flag is 0	RNZ
RPE		Return from the subroutine if parity flag is 1	RPE
RPO		Return from the subroutine if parity flag is 0	RPO
RM		Returns from the subroutine if sign flag is 1	RM
RP		Returns from the subroutine if sign flag is 0	RP

RSTN N

Software Interrupts: 8085 instruction set includes eight software interrupt instructions called Restart (RST) instructions. These are one byte instructions that make the processor execute a subroutine at predefined locations. Instructions and their vector addresses are given in Table

Instruction	Machine hex code	Interrupt Vector Address
RST 0	C7	0000H
RST 1	CF	0008H
RST 2	D7	0010H
RST 3	DF	0018H
RST 4	E7	0020H
RST 5	EF	0028H
RST 6	F7	0030H
RST 7	FF	0032H

The software interrupts can be treated as CALL instructions with default call locations. The concept of priority does not apply to software interrupts as they are inserted into the program as instructions by the programmer and executed by the processor when the respective program lines are read

14. Explain stack instruction set in 8085

PUSH Rp

Push the contents of register pair on to the stack. This instruction is used to write 16 bit data on the stack. the contents of the specified register pair are copied on the stack in the following sequence.
a) The stack pointer is decremented by one and the contents of higher order register copied to the memory location pointed the stack pointer (b) The stack pointer is again decremented by 1 and the contents of lower order of register pair are copied to memory location pointed by the stack pointer. The register pair Rp can be any 16 bit register pair like BC, DE, HL

EXAMPLE: PUSH B
Let BC 3010H, and the stack pointer is initialized at FFFFH. Now if the instruction PUSH B is executed, then initially the stack pointer will decrement by 1 i.e. (SP = FFFE H). The contents of register B (30 H) will be copied to the memory location FFFE H. The stack pointer is then again decremented by one (SP = FFFD H). The contents of register C (10 H) will be copied to this memory location.

POP Rp

When this instruction is executed the contents of the memory location pointed by the stack pointer (SP) register are copied to the low order byte of register pair
The SP is incremented by one and the contents of that memory location are copied to the higher order byte of register pair The SP is again incremented by 1 The process of POP is exactly opposite to PUSH operation. So The contents stored by PUSH are taken back using POP instructions The examples of Rp are BC, DE, HL
EXAMPLE: POP B
Let BC = 3010 H, stack pointer = FFFD H at memory location FFFD H: 10 H is stored and at memory location FFFE H : 30 H is stored and the instruction POP B is executed. The contents of stack location FFFDH are popped off to the C register. The stack pointer will increment by one. The contents of this location i.e. FFFEH are popped off the B register. Then the stack pointer will again increment by 1 (i.e. SP = FFFF H).

SPHL

Load stack pointer with HL register pair contents. When this instruction is executed, the contents of HL register pair are transferred to the stack pointer. The contents of H register

are copied to higher order byte of stack pointer and contents of L register are copied to the lower byte of stack pointer Let HL = ABCD H, Stack Pointer = FFFE H and the instruction

Example SPHL

SPHL is executed.

The contents of stack pointer after the execution of this instruction will be SP = ABCD H

XTHL

Exchange HL with top of stack

When this instruction is executed the contents of L register are exchanged with the stack location pointed by the stack pointer. The contents of H register are exchanged with the next stack location The contents of the stack pointer register are not altered,

EXAMPLE:XTHL

Let H = 01 H, L = 20 H, SP = FFFD H and at memory location FFFD H: 05 H is stored At memory location FFFEH : 06 H is stored. After the execution of instruction XTHL, the contents of L register (20) H will be exchanged with contents of memory location FFFDH and the contents of H register (01H) will be exchanged with the contents of memory location FFFEH.

LXI SP, data

This instruction initializes the stack pointer with 16 bit address. The stack point can be initialized by two method:

(i) Direct method: LXI SP, data (16 bit).

(ii) Indirect method : LXI H, data (16 bit).

SPHL

Generally the stack pointer is initialized by direct method. If the user wants to set the stack pointer to a value that is found out from the program, then in such cases the indirect method is used.

INX SP This instruction increments the Stack Pointer by 1.

DAD SP This instruction adds the contents of Stack Pointer with the contents of HL register pair. The result of addition is store into the HL register pair.

DCX SP This instruction decrements the stack pointer by

15.Explain Input/output group of instruction set in 8085

IN

Description : This instruction copies the data at the port whose address is specified in the instruction into the accumulator.

Example : Port address = 80H, data stored at port address 80H, (80H) = 10H IN 80H ; This instruction will copy the data stored at address 80H, i.e. data 10H in the accumulator

OUT

This instruction sends the contents of accumulator to the output port whose address is specified within the instruction

Example: A 40H

OUT 50H ; This instruction will send the contents of accumulator(40H) to the output address is 50H

16. Explain machine control group of instruction set in 8085

EI

When this instruction is executed the interrupt enable flip-flop is set so that all maskable interrupts are enabled After an interrupt is acknowledged, the interrupt enable flip-flop is reset to reenable the interrupts

DI

Description This instruction resets the interrupt enable flip-flop to disable interrupts. This instruction disables all interrupts except TRAP since TRAP is non-maskable interrupt (carnot be disabled. It is always enabled)

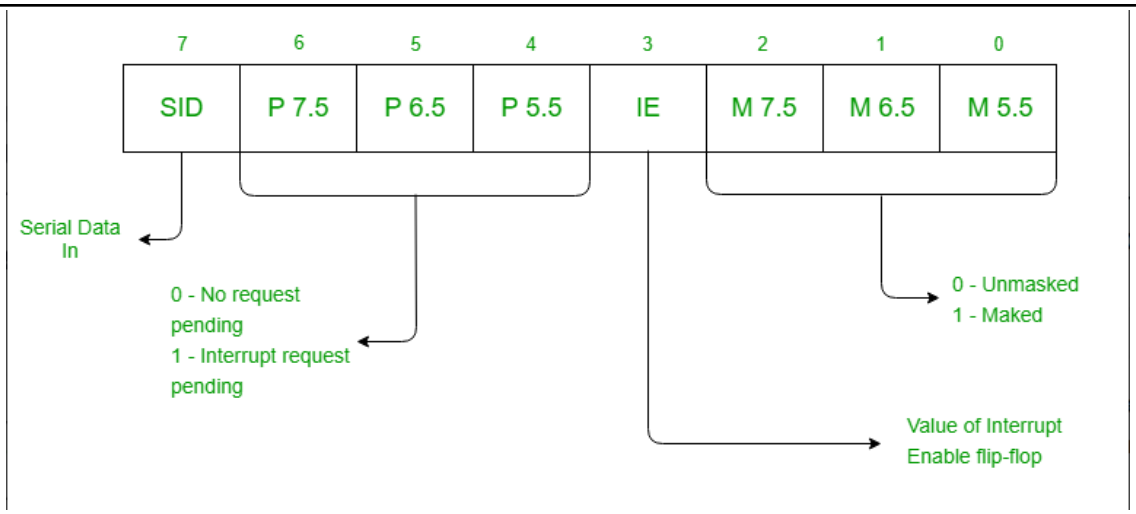
NOP: no operation is performed

HLT: IT halt the processor. it can be restared by a valid interrupt or by applying a RESET signal

RIM : When this instruction is executed the status of the interrupts is copied into the accumulator It also reads serial data through the SID pin.

(a) RIM instruction is executed and accumulator bit pattern is 00100000. The bit pattern indicates that RST 6.5 is pending interrupt.

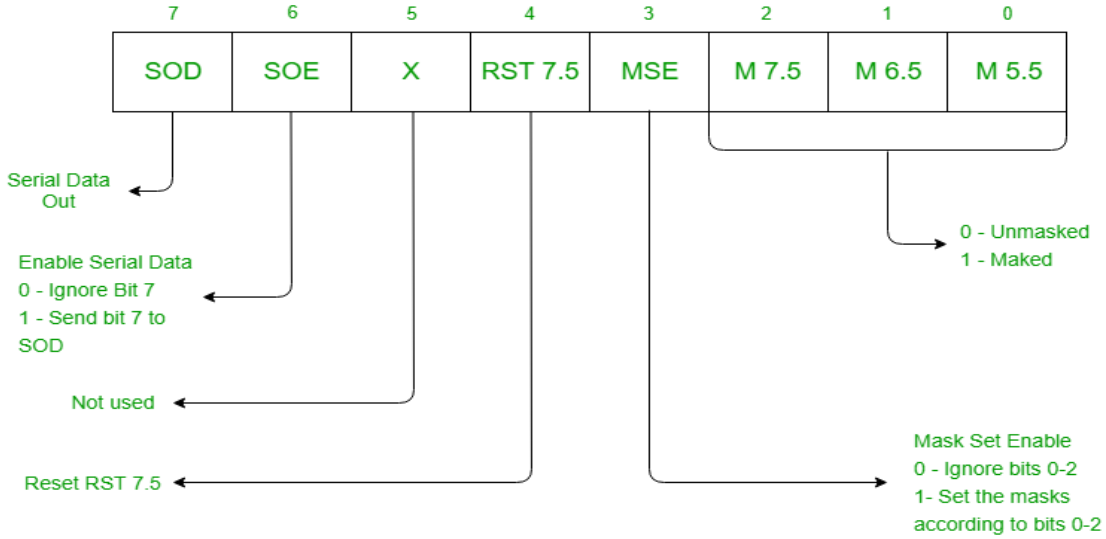
(b) RIM instruction is executed and accumulator bit pattern is 0000 1010. The bit pattern indicates interrupt enable flip flop is set,RST 6.5 is masked and RST 7.5 and RST 5.5 are unmasked.



It is used for the following purposes –

- To check whether RST7.5, RST6.5, and RST5.5 are masked or not.
- To check whether interrupts are enabled or not.
- To check whether RST7.5, RST6.5, or RST5.5 interrupts are pending or not.
- To perform serial input of data.

SIM: When this instruction is executed the interrupts are masked or kept pending as specified in the accumulator contents.
It also sends data on SOD pin.



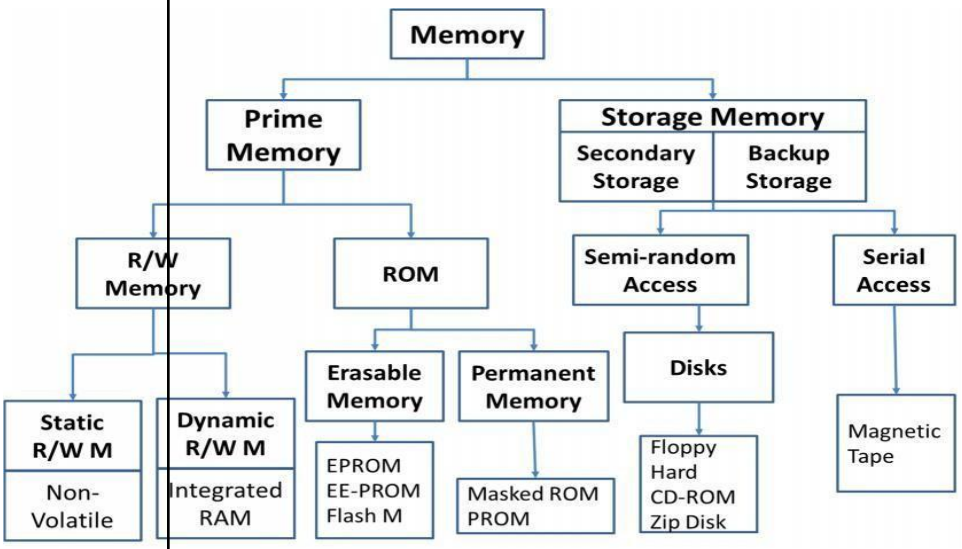
The main uses of SIM instruction are –

- Masking/unmasking of RST7.5, RST6.5, and RST5.5
- Reset to 0 RST7.5 flip-flop.
- Perform serial output of data

Difference between SIM and RIM instructions in 8085 microprocessor

Sr. No.	Sim Instruction	Rim Instruction
1	SIM stands for Set Interrupt Mask.	RIM stands for Read Interrupt Mask.
2	It is responsible for masking/unmasking of RST 7.5, RST 6.5 and RST 5.5.	It checks whether RST 7.5, RST 6.5, RST 5.5 are masked or not.
3	It resets to 0 RST 7.5 flip flop.	It checks whether interrupts are enabled or not and to check whether RST 7.5, RST 6.5 or RST 5.5 interrupts are pending or not.
4	The content of the Accumulator decides the action to be taken. So before executing the SIM instruction, it is mandatory to initialize Accumulator with the required value.	The contents of the Accumulator after the execution of the RIM instruction provide this information.Thus, it is essential to look into the Accumulator contents after the RIM instruction is executed.
5	SIM instruction can be used for serial output of data.	RIM instruction can be used for serial input of data.
6	Its opcode(in Hex) is 30.	Its opcode(in Hex) is 20.

17. Explain classification of memory in detail.



ROM (Read Only Memory):

The first classification of memory is ROM. The data in this memory can only be read, no writing is allowed. It is used to store permanent programs. It is a nonvolatile type of memory.

The classification of ROM memory is as follows:

1. **Masked ROM**: the program or data are permanently installed at the time of manufacturing as per requirement. The data cannot be altered. The process of permanent recording is expensive but economic for large quantities.

2. **PROM (Programmable Read Only Memory):** The basic function is same as that of masked ROM. but in PROM, we have fuse links. Depending upon the bit pattern, the fuse can be burnt or kept intact. This job is performed by PROM programmer.

To do this, it uses high current pulse between two lines. Because of high current, the fuse will get burnt; effectively making two lines open. Once a PROM is programmed we cannot change connections, only a facility provided over masked ROM is, the user can load his program in it. The disadvantage is a chance of re-growing of the fuse and changes the programmed data because of aging

3. EPROM (Erasable Programmable Read Only Memory): the EPROM is programmable by the user. It uses MOS circuitry to store data. They store 1's and 0's in form of charge. The information stored can be erased by exposing the memory to ultraviolet light which erases the data stored in all memory locations. For ultraviolet light, a quartz window is provided which is covered during normal operation. Upon erasing it can be reprogrammed by using EPROM programmer. This type of memory is used in a project developed and for experiment use. The advantage is it can be programmed erased and reprogrammed. The disadvantage is all the data get erased even if you want to change single data bit.

4. EEPROM: EEPROM stands for electrically erasable programmable read only memory. This is similar to EPROM except that the erasing is done by electrical signals instead of ultraviolet light. The main advantage is the memory location can be selectively erased and reprogrammed. But the manufacturing process is complex and expensive so do not commonly used.

R/W Memory (Read/Write Memory):

The RAM is also called as read/write memory. The RAM is a volatile type of memory. It allows the programmer to read or write data. If the user wants to check the execution of any program, user feeds the program in RAM memory and executes it. The result of execution is then checked by either reading memory location contents or by register contents

SRAM (Static RAM): SRAM consists of the flip-flop; using either transistor or MOS. for each bit we require one flip-flop. Bit status will remain as it is; unless and until you perform next write operation or power supply is switched off.

Advantages of SRAM:

Fast memory (less access time)

Refreshing circuit is not required.

Disadvantages of SRAM:

Low package density

Costly

DRAM (Dynamic RAM): In this type of memory a data is stored in form of charge in capacitors. When data is 1, the capacitor will be charged and if data is 0, the capacitor will not be charged. Because of capacitor leakage currents, the data will not be held by these cells. So the DRAMs require refreshing of memory cells. It is a process in which

same data is read and written after a fixed interval

Advantages of DRAM:

High package density

Low cost

Disadvantages of DRAM:

Required refreshing circuit to maintain or refresh charge on the capacitor, every after few milliseconds

Secondary Memory

Magnetic Disk: The Magnetic Disk is Flat, circular platter with metallic coating that is rotated beneath read/write heads. It is a Random access device; read/write head can be moved to any location on the platter

Floppy Disk: These are small removable disks that are plastic coated with magnetic recording material. Floppy disks are typically 3.5″ in size (diameter) and can hold 1.44 MB of data. This portable storage device is a rewritable media and can be reused a number of times. Floppy disks are commonly used to move files between different computers. The main disadvantage of floppy disks is that they can be damaged easily and, therefore, are not very reliable. The following figure shows an example of the floppy disk. Figure 3 shows a picture of the floppy disk.

Hard Disk: Another form of auxiliary storage is a hard disk. A hard disk consists of one or more rigid metal plates coated with a metal oxide material that allows data to be magnetically recorded on the surface of the platters. The hard disk platters spin at 5 a high rate of speed, typically 5400 to 7200 revolutions per minute (RPM).Storage capacities of hard disks for personal computers range from 10 GB to 120 GB (one billion bytes are called a gigabyte).

Optical Disks: Optical Mass Storage Devices Store bit values as variations in light reflection. They have higher area density & longer data life than magnetic storage. They are also standardized and relatively inexpensive. Their Uses: read-only storage with low performance requirements, applications with high capacity requirements & where portability in a standardized format is needed

18. Write assembly language program to find 1's complement in 8085 and 2's complement in 8085

program
program for 2's complement

```
MVI A,02H      // Get the number store in accumulator
CMA            //Complement the number
ADI, 01H       // Add one in the number
STA 2300H  // Store the result
HLT            // Terminate program execution
```

program for 1's complement

```
MVI ,02H       // Get the number
CMA            //Complement the number
STA 2300H  // Store the result
HLT            // Terminate program execution
```

19. Write assembly language program to add two 8bit number in 8085

```
MVI A ,02H          //  number store in accumulator
MVI B,03H           //get second no in reg B number
ADD B               //add value of reg B with value of accumulator
STA 2300H        // Store the result
HLT                 // Terminate program execution
```

20. Draw timing diagram for [1.]OUT [2.] STA instruction

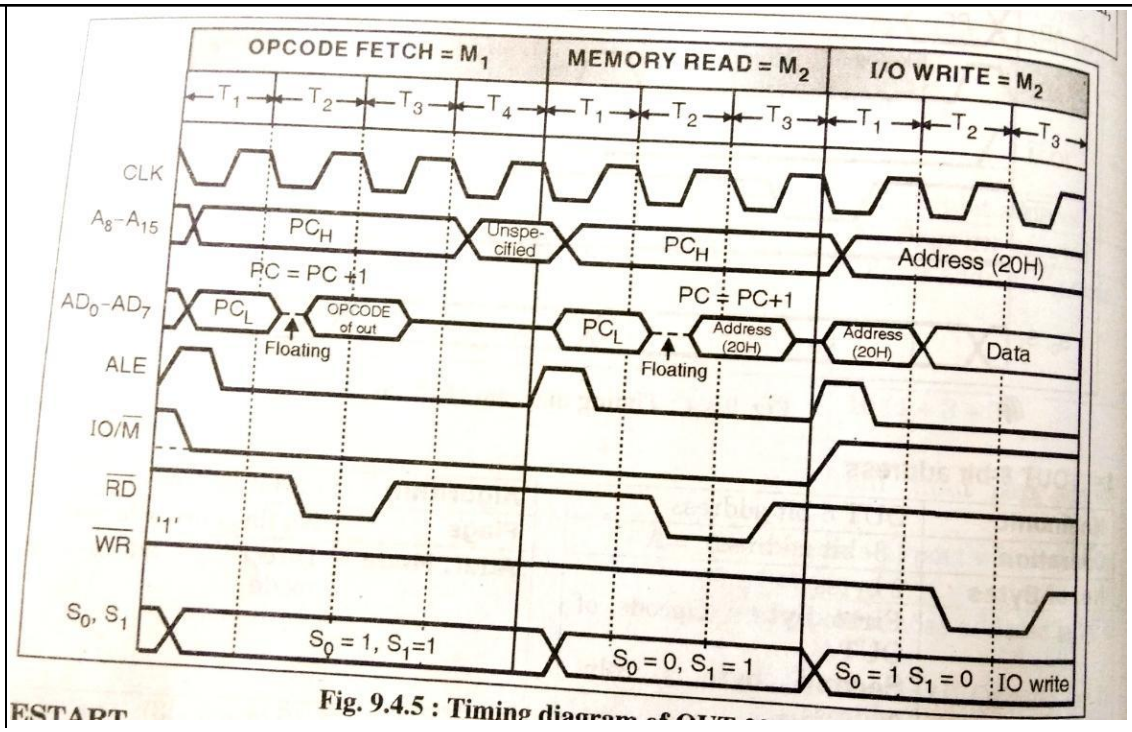


Fig. 9.4.5 : Timing diagram of OUT

This instruction is used to copy the contents of accumulator to the output port whose address is specified into the instruction.

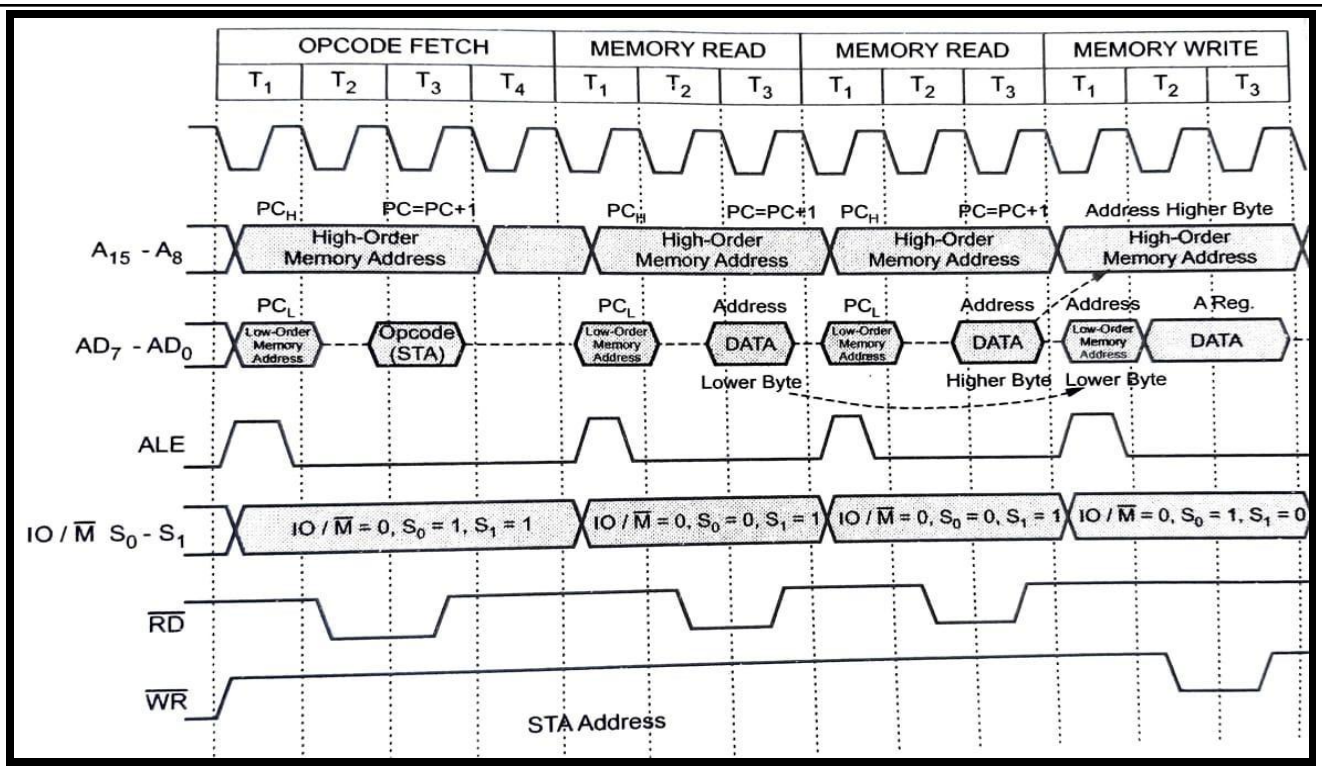
This instruction will copy the contents of accumulator to the output port whose address is 20 H.

The first machine cycle is a opcode fetch which takes opcode from memory decodes it then it comes to know that the next byte after

next machine cycle will be operand fetch to take port address i.e. memory read. This address is stored in temporary register and microprocessor

next machine cycle i.e. I/O write. In this I/O write machine cycle the address is given by temporary The only difference between I/0 read and I/O write is control signal, all other points are same.

STA add



STA add

Store accumulator direct to memory. This instruction will store the contents of the accumulator to the memory location specified in the instruction. The contents of the memory location remain unchanged It is a 3 byte instruction. The first byte is the opcode, second byte is lower order address and the third byte is higher order address

STA 5820

This instruction will store the contents of accumulator at location 5820H