



ZONE TECH
Best Institute For Assistant & Junior Engineer

RPSC-AEn Mains

Work Book Solution

for

MicroProcessor



Visit Our Website
www.zonetech.in



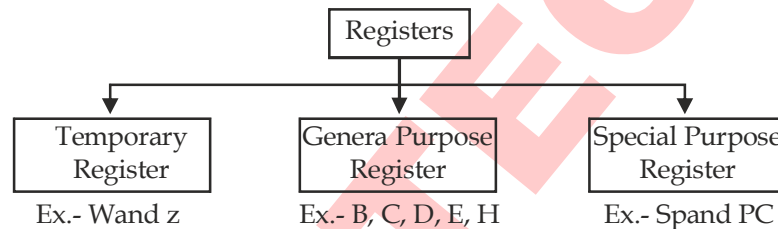
DOWNLOAD NOW

<https://zonetech.in/downloads>

PART - A

2 Marks Questions

1.
Sol.
 - It is group of lines used to transfer Data between the microprocessor and peripherals and memory.
 - Data bus always bidirectional.
2.
Sol. A program that translates an assembly language program from mnemonics to the binary machine code of a computer and these machine codes are called object programme.
3.
Sol. A medium of communication with a computer in which programs are written in mnemonics. An assembly language is specific to a given computer.
4.
Sol. The register array can be categorized as



5.
Sol. The Zero flag is set when the ALU operations results in 0, and the flag is reset if the result is non zero. This flag is affected by the results in Accumulator as well as in the other registers.
6.
Sol. When an instruction fetched into processor from memory. It is loaded in the instruction register. This register is only used for storing the opcode of instruction. It is invisible for users.
7.
Sol. The register in the microprocessor is a temporary storage location in CPU. The register stores the data and addresses temporary for operation.
8.
Sol. The stack is data structure in the RAM area which is worked last in first out. It is used for operation between two registers like swap, add, etc.
9.
Sol. The polling method used for checking, the state of preparedness of external devices connection.
10.
Sol. The Opcode stands for operation codes in the microprocessor i.e. addition, multiplication, etc operation.
The operand contains the data or memory location in the register.
11.
Sol. The macro in the microprocessor is a set of instruction which is group into a single unit. The macro works less than ten instruction set.
12.
Sol. The 16-bit microprocessor has a 16-bit data width, 20-bit address width, and 1MB memory storage. This has a powerful instruction set to do many operations. 8086 is an example of a 16-bit microprocessor.
13.
Sol. The 8-bit microprocessor has an 8-bit data width, a 16-bit address width. The 8085 is an example of an 8-bit microprocessor.

14.
Sol. The memory mapping is used to transfer the logical address space into physical memory but sometimes physical memory is a smaller size. The microprocessor can access external memory. The memory mapping is used for increase access to physical memory.
15.
Sol. Mnemonics are instructions or commands to perform a particular operation given by user to microprocessor e.g. MOV, ADD and SUB.
16.
Sol. The analog signal converted into a digital signal helping by the ADC in the microcontroller. It is an 8-bit chip which has eight-channel for conversion. Analog data has binary converted into digital using logical operation.
17.
Sol. The ALE is an acronym of Address Enable Latch. If pulse goes high i.e. $ALE=1$, it means address bus enable and pulse goes low i.e. $ALE=0$, it means data bus enable. The ALE controls the signal when the pulse goes high because the new task started.
18.
Sol. The dual-core is CPU with two cores or processors in an integrated circuit. Each processor has own cache and controller. The dual-processor has two separate modules and they linked together into a single chip.
19.
Sol. The 8086 has a 16-bit microprocessor which is 40 pins available. 16 data bus, 20 address bus, 2 ground pin, 1 reset pin, and 1 supply pin.
20.
Sol. The main reason to use crystal is high stability, large O and accurate frequency, which remains constant all the time.
21.
Sol. Program counter holds a location in memory of the next step to be performed or the 's location next step's location in a set of steps in case of multistep instruction. In all cases, the counter is increased by one as steps are progressed. Also, the program register keeps the address of the next instruction.
22.
Sol. 8085 has Von Neumann architecture. It is named after the mathematician John Von Neumann. It is based on the concept of stored program control. In this architecture both the data and the program is stored in the same memory.
23.
Sol. Accumulator is an 8 bit register which stores data and performs arithmetic and logical operations. The result of the operation is stored in the accumulator. It is designated by the letter 'A'.
24.
Sol. Stack pointer is of 16 bits length and is used to point to the value at top of the stack for the currently executed instruction.
25.
Sol. SIM is Set Interrupt Mask, which is used to mask the hardware interrupts. RIM is Read Interrupt Mask which is used to check whether the interrupt is masked or not.

PART - B

5 Marks Questions

26.
Sol. **Memory Interfacing**
During the execution of a program the microprocessor needs to access memory quite frequently to read the instruction codes and enable that access. The primary function of memory interfacing is that the microprocessor should be able to read from and write into a given register of memory chip. To perform these operations, the microprocessor should be able to

1. Select chip
2. Identify the register
3. Enable the appropriate buffer

A memory chip select decoder is used to provide chip select signal for each memory device. This will decide the address range that is allotted for each memory IC. The decoding task can be performed by a decoder, a comparator, a bipolar PROM or PLA.

The application of 74LS138, a 3 to 8 line decoder is illustrated in Figure. Here G_1 , $\overline{G_2A}$ and $\overline{G_2B}$ are enable signals. G_1 is an active high signal where as $\overline{G_2A}$ and $\overline{G_2B}$ are active low signals. Thus, to enable the decoder G_1 should be high and $\overline{G_2A}$ and $\overline{G_2B}$ should be low. A, B and C are the select lines and Y_0 to Y_7 are output lines. By applying the proper logic to these lines any one of the outputs can be selected.

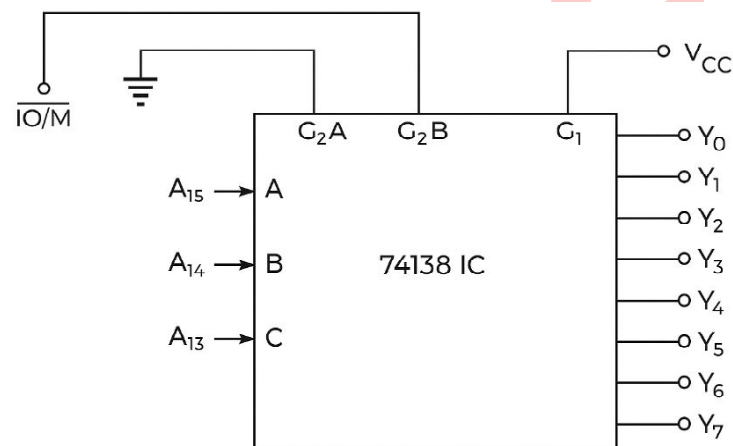


Fig.:- Interfacing of memory device using 74LS138

The Intel 8085 uses 16-bit address bus for addressing memory chips and I/O devices, thus can access $2^{16} = 64$ kB memory and I/O devices. The entire memory address has been divided into 8 zones (0000 H to FFFF H). Address lines A_{15} , A_{14} and A_{13} have been applied to the select line A, B and C of 74LS138. The logic applied to these lines selects a particular memory device (EPROM or RAM). Rest of the address lines (A_{12} - A_0) directly connected to the memory chip thus decide the address of memory location. $\overline{IO/M}$ is connected to $\overline{G_2B}$, it goes low for memory read/write operations. G_1 is connected to V_{CC} and $\overline{G_2A}$ is grounded.

27.

Sol. In many applications the capacity $2^n \times M$ of the single available memory chip is not sufficient. The process of increasing the capacity of memory chip is known as memory expansion.

The memory expansion can be categorized as

- (i) expansion of word size (increasing M)

If we have to interface 8 K × 8 chip and the available chip is 8 K × 4; then this conversion will require expansion of word size from 4 to 8.

- (ii) expansion of word capacity (increasing n)

If the available chips are 16 K × 4 and we want to have 32 K × 4 memory then, the word capacity expansion is required. Thus this conversion requires expansion of word capacity from 16 to 32.

28.

Sol.

Characteristic	Memory-mapped I/O	I/O Mapped I/O
• Device Address	16-bit	8-bit
• Control signals for Input/Output	MEMR/MEMW	IOR / IOW
• Instruction available	Memory-related instructions such as LDA; STA; LDAX; STAX; MOV M, R; ADD M, SUB M; ANA M; etc.	IN and OUT
• Data Transfer	Between any register and I/O	Only between I/O and the accumulator.
• Maximum number of input/output devices possible	The memory map (64 K) is shared between I/O devices and system memory.	The I/O map is independent of the memory map; 256 input devices and 256 output devices can be connected.
• Execution speed	13 T-states (STA, LDA) 7 T-states (MOV M.R)	10 T-states
• Hardware requirements	More hardware is needed to decode 16-bit address.	Less hardware is needed to decode 8-bit address.
• Other features	Arithmetic or logical operations can be directly performed with I/O data.	Not available

29.

Sol. **Intructions Set Based on Addressing modes****Direct Addressing Mode Instruction Set**

In this instruction set, the direct address contains address of operand within instruction itself.

e.g.: LDA 2600H
IN 05H
OUT 07H
STA 2700H

Register Addressing Mode Instruction Set

In this type of instruction set, operands will be present in the microprocessor registers.

e.g.: MOV A, B
ADD B
SUB C

Register Indirect Addressing Mode Instruction Set

In this type of instruction, the operands will be present in memory (M) and address of memory is present in register pair (e.g. HL pair).

It is also called "Indirect addressing mode type instruction set".

e.g.: MOV A, M Memory address in HL register pair
ADD M
LDAX B

Immediate Addressing Mode Instruction Set

In this type, the operand and data is specified within the instruction itself.

In other words we can say, this type of instructions has an operand field rather than an address field.

e.g.: MVI A, 25H
ADI 36H

Implied/Implicit Addressing Mode Instruction Set

If operations of microprocessor are performed by accumulator alone or machine control instructions, then it contains implied addressing type.

e.g.: CMA, RAL, DAA etc.



30.

Sol. LDAX Rp: (Load accumulator indirect)Symbolic form: $[A] \leftarrow [Rp]$

Example: LDAX B; it means that load the content of memory location, whose address is in the BC register pair, into the accumulator.

1-byte instruction

Register indirect addressing mode

Machine cycles (2)-1 opcode fetch + 1 MEMR

4T + 3T = 7 T-states

No flags are affected.

31.

Sol. DAA: (Decimal adjust accumulator after addition)

This instruction "DAA" is used in the program after ADD, ADI, ACI, ADC etc. instructions.

The "DAA" instruction operates on the accumulator result and gives the final result in valid BCD code.

It uses "CY" and "AC" for decimal adjustment.

6 i.e. (0110) is added to 4 LSBs of the content of accumulator, if their values lies between 'A' and 'F' or the "AC flag" is set to '1'.

Also, 6 (0110) is added to 4 MSBs of the content of the accumulator if their value lies in between 'A' and 'F' or "CS flag" is set to '1'.

1-byte instruction

Implied/Implicit addressing mode

Machine cycle(1)-opcode fetch

4T-states

All flags are affected

32.

Sol. CMA: (Complement the accumulator)Symbolic form: $[A] \rightarrow [\bar{A}]$

1-byte instruction

Implied/Implicit addressing mode

1 opcode fetch

Machine cycle(1)-opcode fetch

4T-states

No flags are affected

33.

Sol. RAL: (Rotate accumulator left through carry)Symbolic form: $[A_{n+1}] \leftarrow [A_n]$ $[CS] \leftarrow [A_7]$ $[A_0] \leftarrow [CS]$

The content of the accumulator is rotated left one bit through carry. The seventh bit of the accumulator is moved to carry, and the carry bit is moved to the zero bit of the accumulator.

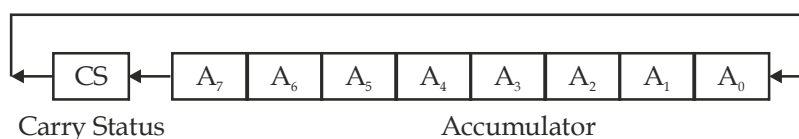


Fig.:- Schematic diagram for RAL

34.

Sol. CALL Instruction

- It is used in the main program to call a subroutine.
- When a subroutine is called, the contents of PC, which is the address of the instruction following the "CALL" instruction, is stored on the stack and the program execution is transferred to the subroutine address.

Unconditional CALL instruction:

When it is executed, microprocessor will store address of next instruction in STACK and PC is located with subroutine address

3-byte instruction

Immediate/Register indirect mode.

Machine cycle(5)-1 opcode fetch + 2 MEMR + 2 MEMW

$6T + (3T \times 2) + (3T \times 2) = 18T$ -states.

No flags are affected

Conditional CALL instruction:

In this case, the program is transferred to the subroutine if the condition is met and at this time it requires 5-MC and 18T-states.

- It condition is not met then the main program is continued and it requires 2-MC and 9T-states.
- In this instruction all flags are used except "AC" flag.
- Conditional CALL:

35.

Sol. RIM: (Read Interrupt Mask)

- Because there are several interrupt lines, when one interrupt request is being served, other interrupt requests may occur and remain pending. The 8085 has an additional instruction called RIM (Read Interrupt Mask) to sense these pending interrupts.
- It is used to read status of masked interrupts, pending interrupts and SID (Serial I/P data).
- This instruction loads the accumulator with eight bits indicating the current status of the interrupt masks, the interrupt enable, pending interrupts, and serial input data as shown in the figure below.

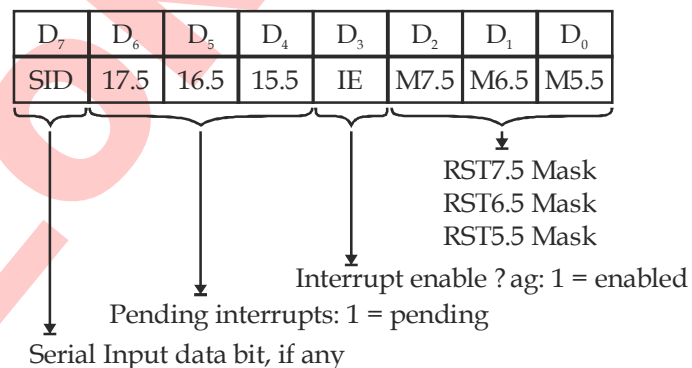


Fig.:- Interpretation of the accumulator bit pattern for RIM

Followings are the common features for instructions NOP, DI, EI, SIM and RIM:

1-byte instruction

Implied/Implicit addressing mode

Machine cycle(1)-1 opcode fetch

4T-states

No flags are affected

36.

Sol. Intel 8253: Programmable Interval Timer-

- The programmable counter/interval time is used in real time application of timing and counting such as BCD/binary counting, generation of time delay etc.
- The Intel 8253 is a 24 pin IC and operates at 5 V (dc) in the frequency range of dc to 2.6 MHz.
- It contains three independent 16-bit counters.
- The 8253 uses NMOS technology and operates any of the following six modes:
 1. Mode 0 $\Rightarrow$ Interrupt on terminal count
 2. Mode 1 $\Rightarrow$ Programmable one shot
 3. Mode 2 $\Rightarrow$ Rate generator
 4. Mode 3 $\Rightarrow$ Square wave generator
 5. Mode 4 $\Rightarrow$ Software trigger strobe
 6. Mode 5 $\Rightarrow$ Hardware trigger strobe

37.

Sol. Intel 8251: Programmable Communication Interface

- Intel 8251 is also known as universal synchronous/asynchronous receiver/transmitter (USART) used to transmit serial data.
- It is compatible with 8085, 8086 and 8088 system.
- It accepts data in parallel format from the microprocessor and converts it into serial data for transmission. It also receives data in serial format and converts it into parallel and sends it in parallel format to the CPU.

PART - C

20 Marks Questions

38.

Sol. Figure shows a simplified but formal structure of a micro-processor-based system or a product. Since a microcomputer is one among many microprocessor-based systems, it will have the same structure as shown in Fig. It includes three components: microprocessor, I/O (input/output) and memory (read/write memory and read-only memory). These components are organized around a common communication path called a bus. The entire group of components is also referred to as a system. At the outset, it is necessary to differentiate between the terms microprocessor and microcomputer because of the common misuse of these terms in popular literature. The microprocessor is one component of the microcomputer. On the other hand, the microcomputer is a complete computer similar to any other computer, except that CPU functions of the microcomputer are performed by the microprocessor. Similarly, the term peripheral is used for input/output devices. The various components of a microprocessor-based product or a microcomputer are shown in Fig. and their functions are described in the forthcoming

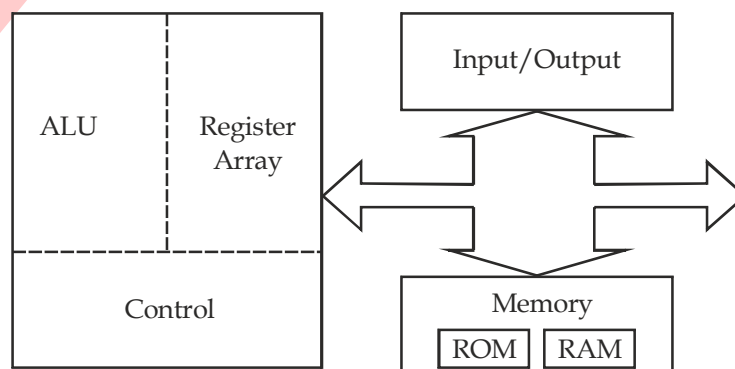


Fig.:- Microprocessor-Based System With Bus Architecture

MICROPROCESSOR

The microprocessor is a semiconductor device consisting of electronic logic circuits manufactured by using either a large scale integration (LSI) or very large scale integration (VLSI) technique. The microprocessor is capable of performing various computing functions and making decisions to change the sequence of program execution in large computers, a CPU performs these computing functions and it is implemented on one or more circuit boards. The microprocessor is in many ways similar to the CPU, but includes all the logic circuitry, including the control unit, on one chip. The microprocessor can be divided into three segments for the sake of clarity, as shown in Fig. arithmetic logic unit (ALU), register array, and control unit.

Arithmetic Logic Unit- This is the area of the microprocessor where various computing functions are performed on data. The ALU performs arithmetic operations such as addition and subtraction, and logic operations such as AND, OR, and exclusive-OR. Results are stored either in registers or in memory.

Register Array- This area of the microprocessor consists of various registers. These registers are primarily used to store data temporarily during the execution of a program. Some of the registers are accessible to the user through instructions.

Control Unit- The control unit provides the necessary timing and control signals to all the operations in the microcomputer. It controls the flow of data between the microprocessor and memory and peripherals.

Memory- Memory stores such binary information as instructions and data, and provides that information to the microprocessor whenever necessary. To execute programs, the microprocessor reads instructions and data from memory and performs the computing operations in its ALU section. Results are either transferred to the output section for display or stored in memory for later uses. The memory block shown in Fig. has two sections: read-only memory (ROM) and Read/Write memory (R/WM), popularly known as Random-Access Memory (RAM).

39.

Sol. Address Bus- The address bus is a group of 16 lines generally identified as A_0 to A_{15} . The address bus is unidirectional: bits flow in one direction only—from the MPU to peripheral device identifying a peripheral or a memory location (step 1). The address lines A_0 to A_7 are shared by data bus. Thus, the data and part of the address are transmitted over a set of shared lines. This is known as multiplexing and lines are known as AD_0 to AD_7 .

In the computer system, each peripheral or memory location is identified by a binary number, called an address, and the address bus is used to carry a 16-bit address. This is similar to the postal address of a house. A house can be identified by various number schemes. For example, the thirty-fifth house in a lane can be identified by the two-digit number 35 or by the four-digit number 0035. The two-digit numbering scheme can identify only a hundred houses, from 00 to 99. On the other hand, the four-digit scheme can identify ten thousand houses from 0000 to 9999. Similarly, the number of address lines of the MPU determines its capacity of identifying different memory locations (or peripherals). The 8085 MPU with its 16 address lines is capable of addressing $2^{16} = 65,536$ (generally known as 64 K) memory locations. The 1K memory is determined by rounding off 1024 to the nearest thousand; similarly 65,536 is rounded off to 64,000 as multiple of 1K.

Most 8-bit microprocessors have 16 address lines. This may explain why most microcomputer systems based on 8-bit microprocessors have 64K memory. However, not every microcomputer system has 64K memory. In fact, most single-board microcomputers have, memory less than 2K, even if the MPU is capable of addressing 64K memory. The number of address lines is arbitrary; it is determined by the designer of a microprocessor based on such considerations as availability of pins and intended applications of the processor. For example, the Motorola 68000 has 24 address lines.

Data Bus- The data bus is a group of eight lines used for data flow. These lines are bidirectional—data flow in both directions between the MPU and memory and peripheral devices. The MPU uses the data bus to perform the second function: transferring data (Step 2).

The eight data lines enable the MPU to manipulate 8-bit data ranging from 00 to FF ($2^8 = 256$ numbers). The largest number that can appear on the data bus is 11111111 (255_{10}). The 8085 microprocessor is known as an 8-bit microprocessor. Microprocessors such as the Intel 8086, Zilog Z8000, and Motorola 68000 have 16 data lines; thus they are known as 16-bit microprocessors and the Intel 80386/486 have 32 data lines; thus they are classified as 32-bit microprocessors.

Control Bus- The control bus is comprised of various single lines that carry synchronization signals. The bus is bidirectional but the individual line is unidirectional. The MPU uses such lines to perform the third function: providing timing signals (Step 3).

The term bus, in relation to the control signals, is somewhat confusing. These are not groups of lines like address or data buses, but individual lines that provide a pulse to indicate an MPU operation. The MPU generates specific control signals for every operation (such as Memory Read or I/O Write) it performs. These signals are used to identify a device type with which the MPU intends to communicate.

To communicate with a memory—for example, to read an instruction from a memory location—the MPU places the 16-bit address on the address bus. The address on the bus is decoded by an external logic circuit, and the memory location is identified. The MPU sends a pulse called Memory Read as the control signal. The pulse activates the memory chip, and the contents of the memory location (8-bit data) are placed on the data bus and brought inside the microprocessor.

What happens to the data byte brought into the MPU depends on the internal architecture of the microprocessor, which we will describe in the next section.

40.

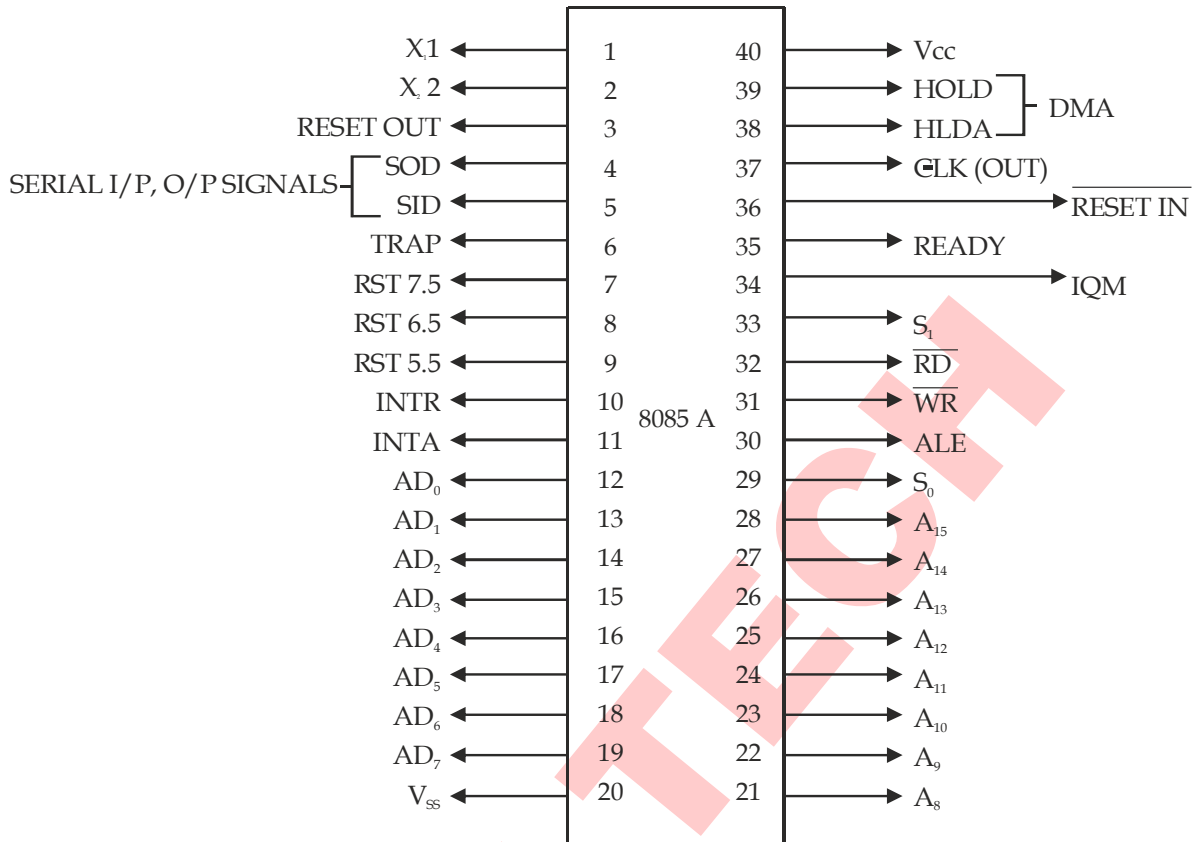
Sol. Pin Description of 8085- 8085 is a 40 pin IC, DIP package. The signals from the pins can be grouped as follows:

1. Power supply and clock signals.
2. Address bus.
3. Data bus.
4. Control and status signals.
5. Serial I/O ports.
6. Interrupts and externally initiated signals.
7. Direct Memory Access.

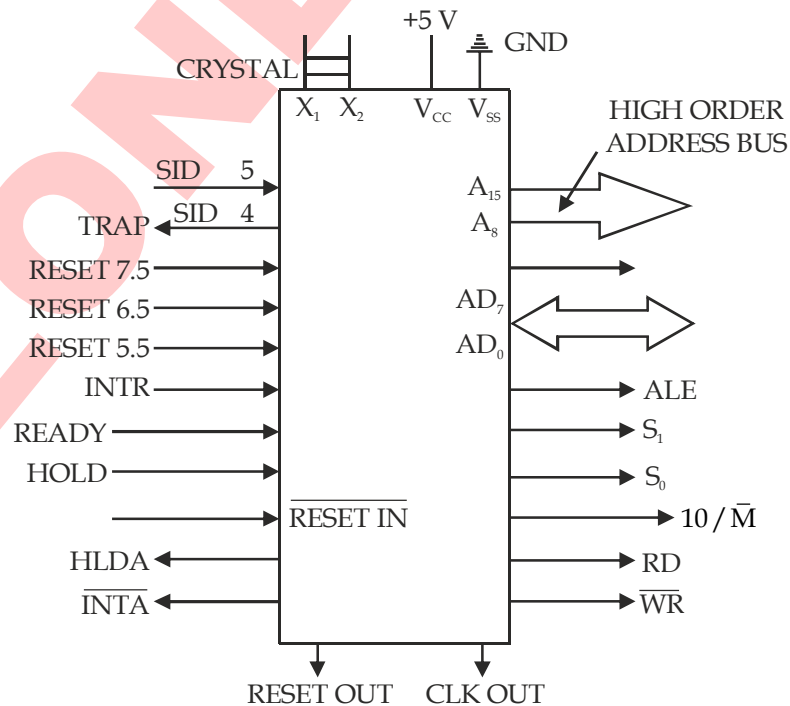
1. **Power supply and Clock frequency signals:**

- Vcc + 5 volt power supply
- Vss Ground

- **X1, X2:** A Crystal or LC network is connected on these pins to set the frequency of internal clock generator. Since the basic operating timing frequency is 3MHz, a 6MHz crystal is connected externally as the frequency is internally divided by two.



(a) Pin Diagram of 8085



(b) Logical Schematic of Pin Diagram

- **CLK (Out)-** Clock Out is used as the system clock for peripheral and devices interfaced with the microprocessor.
- 2. **Address Bus:**
 - **A8-A15** (output; 3-state): It carries the most significant 8 bits of the memory address or the 8 bits of the I/O address.
- 3. **Multiplexed Address/Data Bus:**
 - **AD0-AD7** (input/output; 3-state): These multiplexed set of lines used to carry the lower order 8 bit address as well as data bus.
 - (i) During the opcode fetch operation, in the first clock cycle, the lines deliver the lower order address A_0-A_7 .
 - (ii) In the subsequent IO/memory, read/write clock cycle the lines are used as data bus.
 - (iii) The CPU may read or write out data through these lines.
- 4. **Control and Status Signals:**
 - **ALE** (output)-Address Latch Enable: This signal helps to capture the lower order address presented on the multiplexed address/data bus.
 - **RD** (output 3-state, active low)- Read memory or I/O device: This indicates that the selected memory location or I/O device is to be read and that the data bus is ready for accepting data from the memory or I/O device.
 - **WR** (output 3-state, active low)-Write memory or I/O device: This indicates that the data on the data bus is to be written into the selected memory location or I/O device.
 - **IO/M** (output)- Select memory or an I/O device: This status signal indicates that the read/write operation relates to whether the memory or I/O device,
 - (i) It goes high to indicate an I/O operation.
 - (ii) It goes low for memory operations.
 - **S₀ and S₁:** Status Signals are used to know the type of current operation of the microprocessor as given below in Table.

IO/M (Active Low)	S ₁	S ₂	Data Bus Status (Output)
0	0	0	Halt
0	0	1	Memory WRITE
0	1	0	Memory READ
1	0	1	IO WRITE
1	1	0	IO READ
0	1	1	Opcode fetch
1	1	1	Interrupt acknowledge

- 5. **Single Bit Serial I/O ports:**
 - **SID-** Serial Input Data Line: The data on this line is loaded into accumulator bit 7 whenever a RIM instruction is executed.
 - **SOD-** Serial Output Data Line: The SIM instruction loads the value of bit 7 of the accumulator into SOD latch if bit 6 (SOE) of the accumulator is 1.
- 6. **Interrupts and Externally Initiated Operations:** The meaning of 'interrupt' is to break the sequence of operation. While the CPU is executing a program, on 'interrupt' breaks the normal sequence of execution of instructions, diverts its execution to some other program called Interrupt Service Routine (ISR). After executing ISR, the control is transferred back again to the main program. Interrupts are used when interfacing I/O devices is required that provides or require data at relatively low data transfer rate.

There are two types of interrupts namely, Hardware Interrupts (External Interrupts) for 8085 these are TRAP, RST7.5, RST6.5, RST5.5, INTR and Software Interrupts (Internal Interrupts and Instructions) for 8085 these are RST0, RST1, RST2, RST3, RST4, RST5, RST6, RST7. The hardware interrupts are presented below in the order of their priority (from lowest to highest):



INTR is maskable, 8080A compatible interrupt. When the interrupt occurs the processor fetches from the bus one instruction. **INTR** is level triggered interrupt.

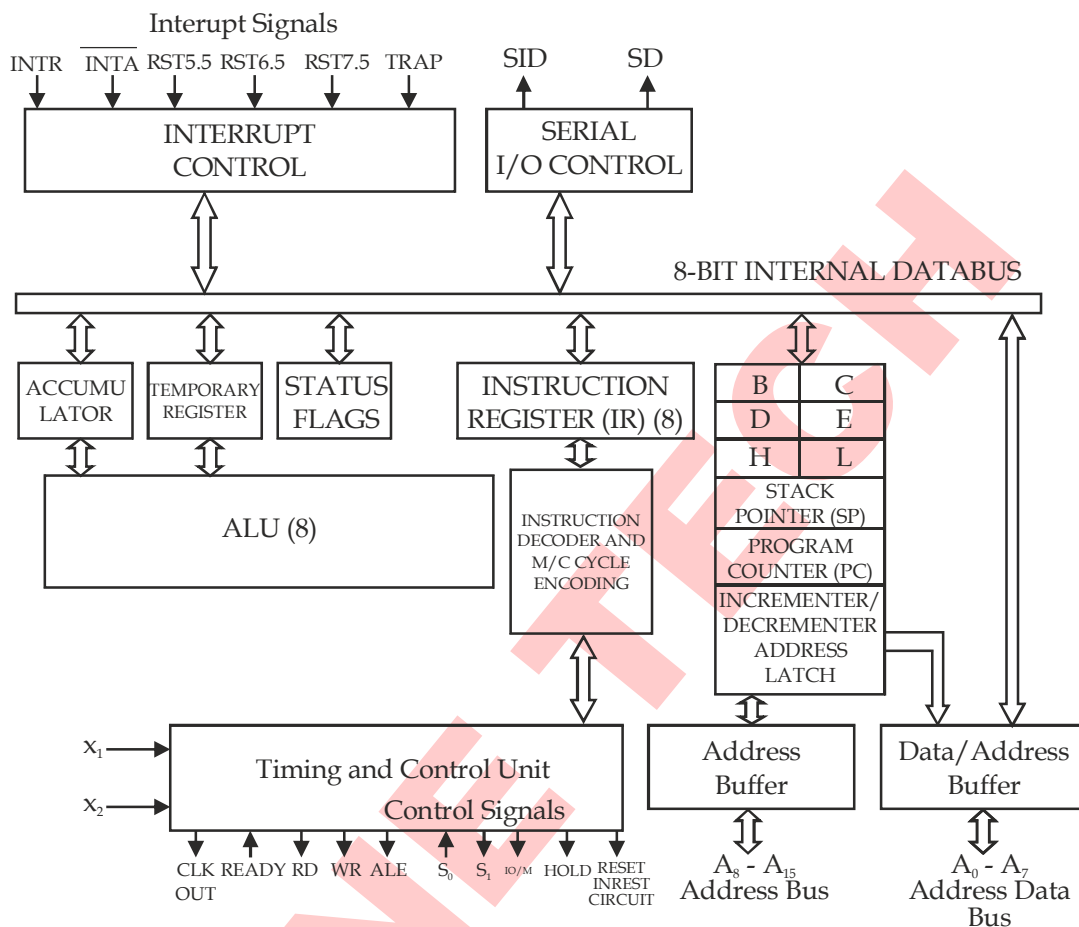


Fig.- Block Diagram of 8085 Microprocessor

RST5.5 is a maskable interrupt. When this interrupt is received the processor saves the contents of the PC register into stack and branches to 2CH (hexadecimal) address.

RST5.5 is level triggering interrupts.

RST6.5 is a maskable interrupt. When this interrupt is received the processor saves the contents of the PC register into stack and branches to 34H (hexadecimal) address.

RST6.5 is level triggering interrupts.

RST7.5 is a maskable interrupt. When this interrupt is received the processor saves the contents of the PC register into stack and branches to 3CH (hexadecimal) address.

RST7.5 is a raising edge triggering interrupt.

Trap (4.5) is a non-maskable interrupt. When this interrupt is received the processor saves the contents of the PC register into stack and branches to 24H (hexadecimal) address.

RST4.5 is raising edge level triggering interrupt.

All maskable interrupts can be enabled or disabled using EI and DI instructions. RST 5.5, RST6.5 and RST7.5 interrupts can be enabled or disabled individually using SIM instruction.

- **Reset In** (input, active low)
 - (i) This signal is used to reset the microprocessor.
 - (ii) The program counter inside the microprocessor is set to zero.
 - (iii) The buses are tri-stated.
- **Reset Out** (Output)
 - (i) It indicates CPU is being reset.
 - (ii) Used to reset all the connected devices when the microprocessor is reset.
- 7. **Direct Memory Access (DMA):** The CPU controls the data transfer operation between memory and I/O device. Direct Memory Access operation is used for large volume data transfer between memory and an I/O device directly. The CPU is disabled by tri-stating its buses and the transfer is effected directly by external control circuits.
- **HOLD:** Indicates that another master is requesting the use of the address and data buses.
 - (i) The CPU, upon receiving the hold request, will relinquish the use of the bus as soon as the completion of the current bus transfer. Internal processing can continue.
 - (ii) The processor can regain the bus only after the HOLD is removed.
 - (iii) When the HOLD is acknowledged, the Address, Data RD, WR and IO/M lines are 3-stated.
- **HLDA:** Hold Acknowledge: Indicates that the CPU has received the HOLD request and that it will relinquish the bus in the next clock cycle. HLDA goes low after the Hold request is removed. The CPU takes the bus one half-clock cycle after HLDA goes low.
- **READY (input):** This signal synchronizes the fast CPU and the slow memory, peripherals. Memory and I/O devices will have slower response compared to microprocessors and before completing the present job such a slow peripheral may not be able to handle further data or control signal from CPU.
 - (i) The processor sets the READY signal after completing the present job to access the data.
 - (ii) The microprocessor enters into WAIT state while the READY pin is disabled.

41.

Sol. The internal architecture of the 8085 microprocessor determines how and what operations can be performed with the data. These operations are

1. Store 8-bit data.
2. Perform arithmetic and logic operations.
3. Test for conditions.
4. Sequence the execution of instructions.
5. Store data temporarily during execution in the defined R/W memory locations called the stack.

To perform these operations, the microprocessor requires registers, an arithmetic logic unit (ALU) and control logic, and internal buses (paths for information flow). Fig. is a simplified representation of the 8085 internal architecture; it shows only those registers that are programmable, meaning those registers that can be used for data manipulation by writing instructions. These registers are described in reference to the five operations previously listed.

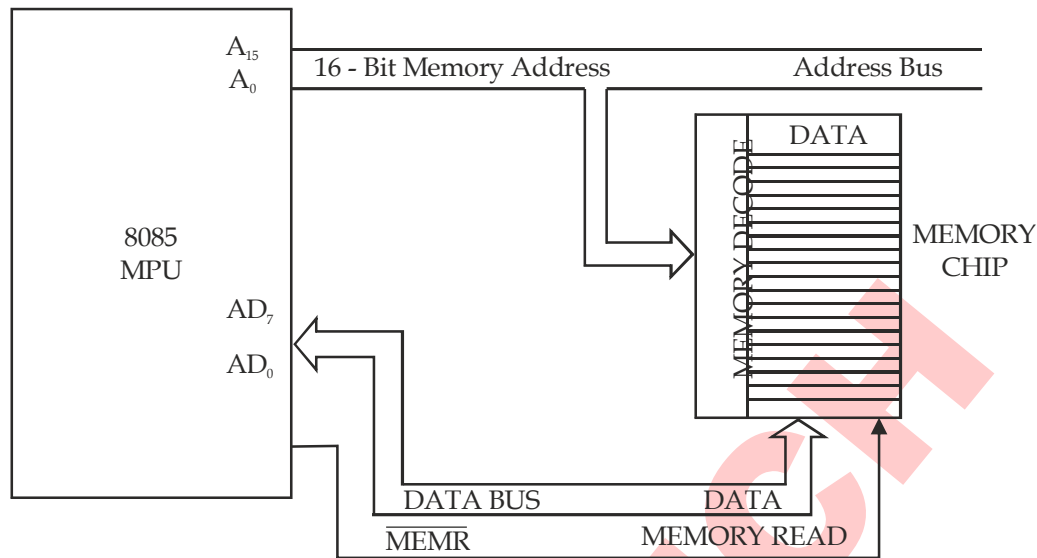


Fig.- Memory Read Operation

1. **Registers-** The registers used in this processor are classified as given:

(a) **General Purpose Registers-** These are also called scratch pad registers as data can be stored in them and to read or write data from these registers bus access is not required.

Registers- The 8085 has six general purpose registers to perform the first operation listed above; that is, to store 8-bit data during a program execution. These registers are identified as B, C, D, E, H, and L, as shown in Fig. They can be combined as register pairs-BC, DE, and HL-to perform some 16-bit operations.

These registers are programmable, meaning that a programmer can use them to load or transfer data from the registers by using instructions. For example, the instruction MOV B,C copies the data from register C to register B. Conceptually, the registers can be viewed as memory locations, except they are built inside the microprocessor and identified by specific letters for user convenience. Some microprocessors do not have these types of registers; instead, they use memory space as their registers.

(b) **Temporary Registers-** These registers cannot be accessed by programmer and are used internally by ALU. There are two types of this registers:

(i) **Temporary Data Register-** This is an 8 bit operand register because it provides operand to ALU. ALU stores intermediate result in this register during the operation.

(ii) **w and z registers-** These are 8 bit registers to provide input to ALU and are used by control section to hold data during an arithmetic or logical operation.

Accumulator- The accumulator is an 8-bit register that is part of the arithmetic logic unit (ALU). This register is used to store 8-bit data and in performing arithmetic and logical operations. The result of an operation is stored in the accumulator. The accumulator is also identified as register A.

Flags- The ALU includes five flip-flops that are set or reset according to data conditions in the accumulator and other registers. The microprocessor uses them to perform the third operation, namely, testing for data conditions.

For example, after an addition of two numbers, if the sum in the accumulator is larger than eight bits, the flip-flop that is used to indicate a carry, called the carry (CY) flag, is set to one. When an arithmetic operation results in zero, the flip-flop called the zero (Z) flag is set to one. The 8085 has five flags to indicate five different types of data conditions. They are Zero (Z), Carry (CY), Sign (S), Parity (P), and Auxiliary Carry (AC) flags. The most commonly used flags are Sign, Zero, and Carry.

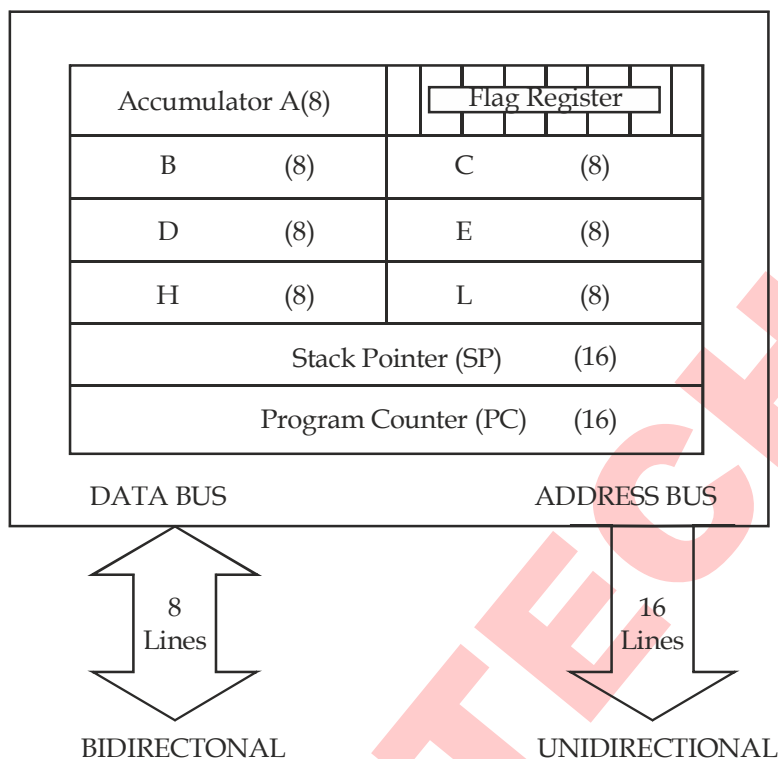


Fig.- The 8085 Programmable Registers

Figure shows an 8-bit register, called the flag register, adjacent to the accumulator. It is not used as an 8-bit register, five bit positions, out of eight, are used to store the outputs of the five flip-flops. The flags are stored in the 8-bit register so that the programmer can examine these flags (data conditions) by accessing the register through an instruction. In the instruction set, the term PSW (Program Status Word) refers to the accumulator and the flag register.

Machine Cycle	Status				Controls	
	IO / $\overline{M}$	S_1	S_0	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
Opcode Fetch (OF)	0	1	1	0	1	1
Memory Read	0	1	0	0	1	1
Memory Write	0	0	1	1	0	1
I/O Read (I/OR)	1	1	0	0	1	1
I/O Write (I/OW)	1	0	1	1	0	1
Acknowledge of INTR (INTA)	1	1	1	1	1	0
BUS Idle (BI) : DAD	0	1	0	1	1	1
ACK of RST, TRAP	1	1	1	1	1	1
HALT	Z	0	0	Z	Z	1
HOLD	Z	X	X	Z	Z	1
X- Unspecified, and Z- High impedance state						

These flags have critical importance in decision-making process of the microprocessor. The conditions (set or reset) of the flags are tested through software instructions. For example, the instruction JC (Jump On Carry) is implemented to change the sequence of a program when the CY flag is set. The importance of the flags cannot be emphasized enough.



Program Counter (PC)- This 16-bit register deals with the fourth operation, sequencing the execution of instructions. This register is a memory pointer. Memory locations have 16-bit addresses, and that is why this is a 16-bit register.

The microprocessor uses this register to sequence the execution of 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)- The stack pointer is also a 16-bit register used as a memory pointer; initially it will be called the stack pointer register to emphasize that it is a register that points to a memory location in R/W memory, called the stack. The beginning of the stack is defined by loading a 16-bit address in the stack pointer (register).

Several other registers are used by 8085 only for its internal operation. They cannot be accessed by the programmer.

2. **Arithmetic Logic Unit (ALU)-** This unit performs arithmetic operations of addition and subtraction and logical operations of AND, OR, EX-OR, complement etc., on 8 bit variables.
3. **Address Buffer-** This is an 8 bit unidirectional buffer to drive external high order address bus ($A_{15}-A_8$).
4. **Address/Data Buffer-** This is an 8 bit bidirectional buffer to drive external low order address bus (A_0-A_7) and data bus (D_0-D_7).
5. **Instruction Decoder-** It decodes the opcode sent by instruction register and accordingly gives the timing and control signals for controlling the registers, ALU, data buffers and external peripherals.
6. **Incrementer/Decrementer Address Latch-** It is a 16 bit register for incrementing or decrementing the contents of PC and SP.
7. **Timing and Control Unit-** The operations in 8085 are synchronized with clock signal. The control circuitry controls all the operations of processor like fetching and decoding instructions along with generating signals to interface with peripheral devices.
8. **Serial I/O Control-** It provides two signals named as SID (Serial input data) and SOD (Serial output data) for serial data transfer.
9. **Interrupt Control-** It has already been discussed in Art 22.9.1.

Peripheral or Externally Initiated Operations- External devices (or signals) can initiate the following operations, for which individual pins on the microprocessor chip are assigned: Reset, Interrupt, Ready, Hold.

Reset- When the reset is activated, all internal operations are suspended and the program counter is cleared (it holds 0000H). Now the program execution can again begin at the zero memory address.

Interrupt- The microprocessor can be interrupted from the normal execution of instructions and asked to execute some other instructions called service routine (for example, emergency procedures). The microprocessor resumes its operation after completing the service routine.

Ready- The 8085 has a pin called READY. If the signal at this READY pin is low, the microprocessor enters into a Wait state. This signal is used primarily to synchronize slower peripherals with the microprocessor.

Hold- When the HOLD pin is activated by an external signal, the microprocessor relinquishes control of buses and allows the external peripheral to use them. For example, the HOLD signal is used in Direct Memory Access (DMA) data transfer.

22.10.1. Instruction Cycle, Machine Cycle and T-state. Instruction Cycle- The essential steps required the processor to fetch and execute an instruction is called as instruction cycle. The time required to execute an instruction (IC) is the sum of time required for fetch cycle (FC) and execution cycle (EC).

Machine Cycle- It is the time taken by processor to complete the operation like accessing memory and I/O device, fetching opcode etc.

T-State- Microprocessor performs any operation in a specified time period in terms of clock cycle. Each clock cycle is called as a T-State and the number / T-States required to perform an operation is called a machine cycle.



22.10.2. Instruction Format- 8085 instruction set can be classified into three types as per the size of instruction:

- One Byte Instruction. In this instruction only opcode is used. Example MOV A, B instruction copies the content of B register in Accumulator.
- Two Byte Instruction. In this instruction the first byte is opcode and the second byte is operand. Example MVI B, 04H instruction loads 04H into B register.
- Three Byte Instruction. In this instruction the first byte is opcode and the other two bytes represent address or 16 bit data. Example JMP 2102H instruction loads 2102H into PC.

22.10.3. Instruction Set- Instructions are also classified as per their function:

1. Data Transfer group,
2. Arithmetic group,
3. Logical group,
4. Branching group,
5. Stack and machine control group.

The notations used in instructions are given here in tabular form.

Notations	Meaning
M	Memory location pointed by HL pair
R	8 bit register
RS	Source register
RD	16 bit register pair
RP	Destination register
Addr	16 bit address
Data 16 bit	16 bit data
Data	8 bit data

Addressing Modes in 8085- There are 5 addressing modes in 8085:

- Register Addressing Mode-** In this mode the source operand and destination operand are the registers with the microprocessor. Example-MOV A, B loads data of register B into accumulator.
- Immediate Addressing Mode-** In this mode 8 or 16 bit data can be specified as a part of instruction. Example-MVI A, 40H instruction will load accumulator with 40H.
- Direct Addressing Mode-** In this mode 16 bit address of operand is specified. Example- JMP 2100H instruction will load 16 bit address 2100H into PC.
- Indirect Addressing Mode-** In this mode register pair is used to specify the memory address where the operand is located. Example-LDAX B will load accumulator with the contents of memory location pointed by BC register pair.
- Implied Addressing Mode-** In this mode the opcode itself specifies address of the operand. Example-STC instruction sets carry flag and similarly CMA instruction complements contents of accumulator.

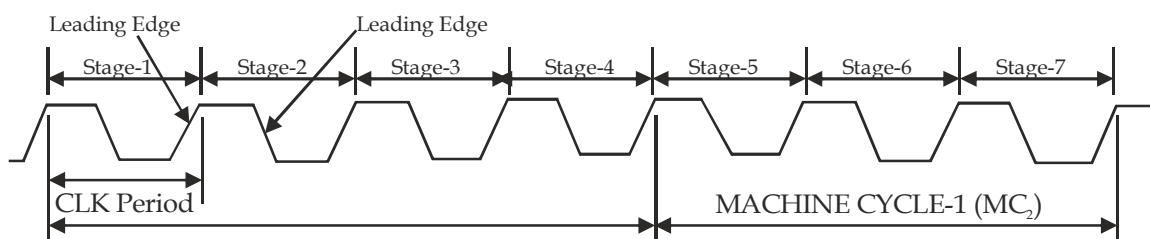


Fig.- Machine Cycle Showing Clock Period

42.

- Sol. T-State-** Microprocessor performs any operation in a specified time period in terms of clock cycle. Each clock cycle is called as a T-State and the number of T-States required to perform an operation is called a machine cycle.

The clock cycle shown in Fig. has two edges (leading and trailing or lagging). T-State is defined as the time interval between 2-trailing or leading edges of the clock.

Machine Cycle- It is the time taken by processor to complete the operation like accessing memory and I/O device, fetching opcode etc.

Instruction Cycle- The function of the microprocessor is divided into fetch and execute cycle of any instruction of a program. The program is nothing but number of instructions stored in the memory in sequence. In the normal process of operation, the microprocessor fetches (receives or reads) and executes one instruction at a time in the sequence until it executes the halt (HLT) instruction.

The time taken by the processor in performing the fetch and execute operations are called fetch and execute cycle. Thus, sum of the fetch and execute cycle is called the instruction cycle as indicated in Fig. 22.10(a).

Instruction Cycle (IC) = Fetch Cycle (FC) + Execute Cycle (EC)

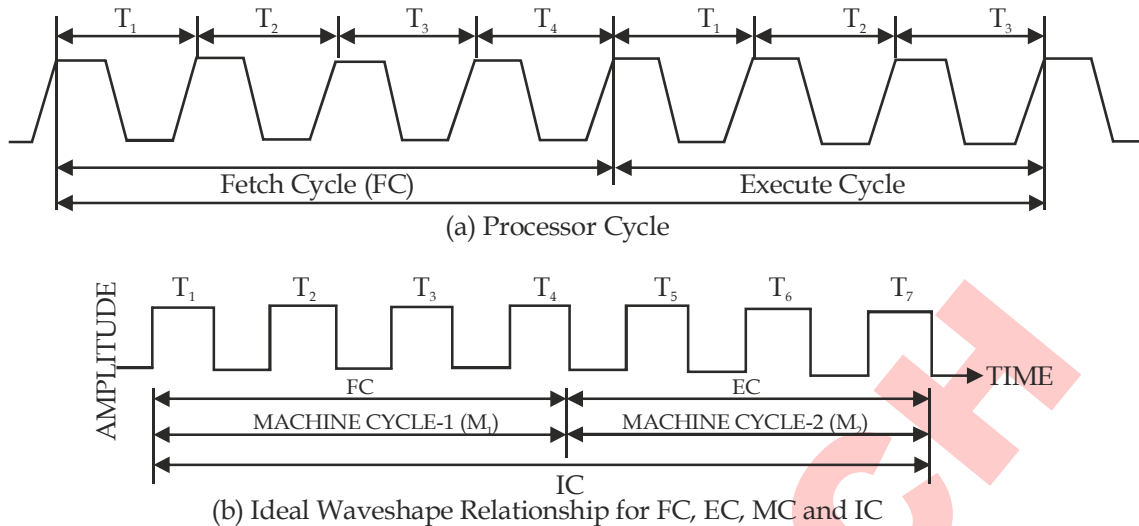
These cycles have been illustrated in Fig. 22.10(a) and (b). Each read or write operation constitutes a machine cycle. The instructions of 8085 require 1-5 machine cycles containing 3- 6 states (clocks). The 1st machine cycle of any instruction is always an Opcode fetch cycle in which the processor decides the nature of instruction. It is of at least 4-states. It may go up to 6-states.

Opcode Fetch- A microprocessor either reads or writes to the memory or I/O devices. The time taken to read or write for any instruction must be known in terms of the processor clock. The 1st step in communicating between the microprocessor and memory is reading from the memory. This reading process is called opcode fetch. The process of opcode fetch operation requires minimum 4-clock cycles T_1 , T_2 , T_3 , and T_4 and is the 1st machine cycle (M_1) of every instruction.

In order to differentiate between the data byte pertaining to an opcode or an address, the machine cycle takes help of the status signal $IO/\overline{M}$, S_1 , and S_0 . The $IO/\overline{M} = 0$ indicates memory operation and $S_1, S_0 = 1$ indicates Opcode fetch operation.

The opcode fetch machine cycle M_1 consists of 4-states (T_1 , T_2 , T_3 , and T_4). The 1st 3-states are used for fetching (transferring) the byte from the memory and the 4th-state is used to decode it. Thus, thorough understanding about the communication between memory and microprocessor can be achieved only after knowing the processes involved in reading or writing into the memory by the microprocessor and time taken w.r.t. its clock period.

Timing Diagram- For fetching a byte, the microprocessor must find out the memory location where it is stored. Then provide condition (control) for data flow from memory to the microprocessor. The microprocessor fetches opcode of the instruction from the memory as per the sequence shown in Fig.

**T1 State**

- (i) The μP sends a low on $IO/\overline{M}$ means microprocessor wants to communicate with memory.
- (ii) The μP sends a high on status signal S_1 and S_0 indicating fetch operation.
- (iii) The μP sends 16-bit address. AD bus has address in 1st clock of the 1st machine cycle, T_1 .
- (iv) The μP sends a high on ALE so that AD, to ADO address is latched in the external latch. AD bus now can carry data.

T2 State

- (i) The RD control signal becomes low to enable the memory for read operation.
- (ii) The memory places opcode on the AD bus.
- (iii) The data is placed in the data register (DR) and then it is transferred to IR.

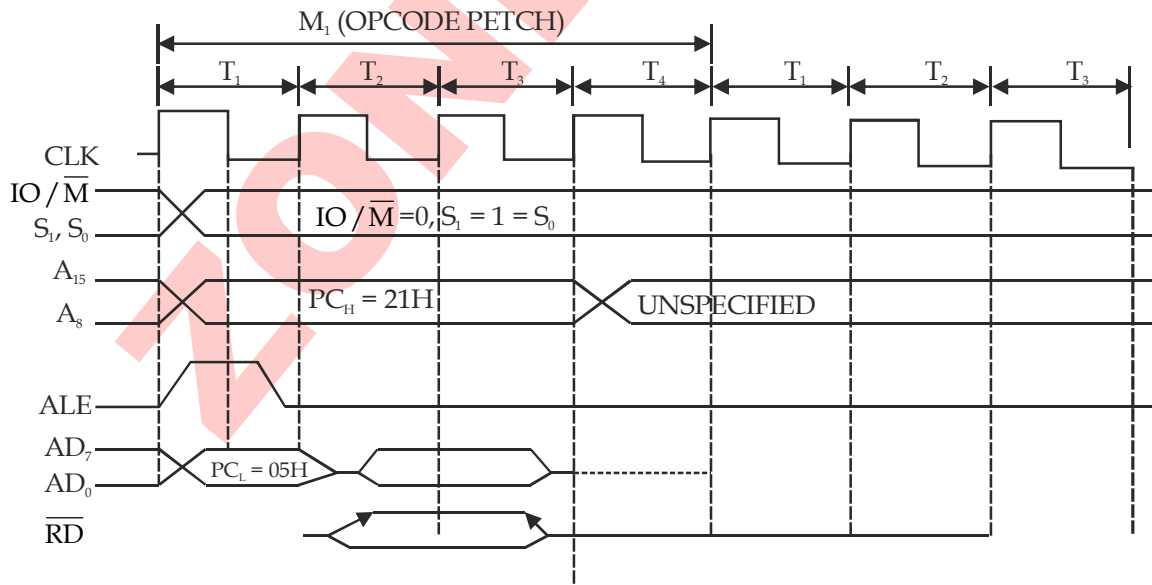
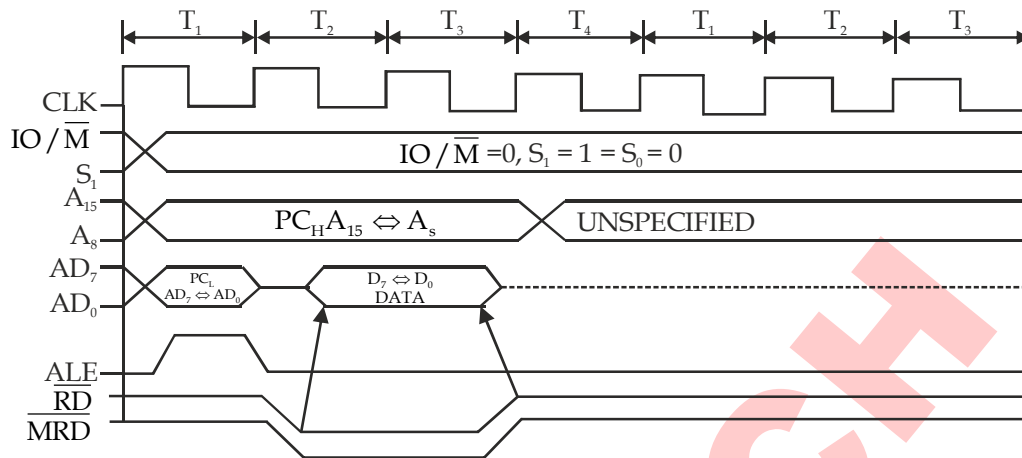
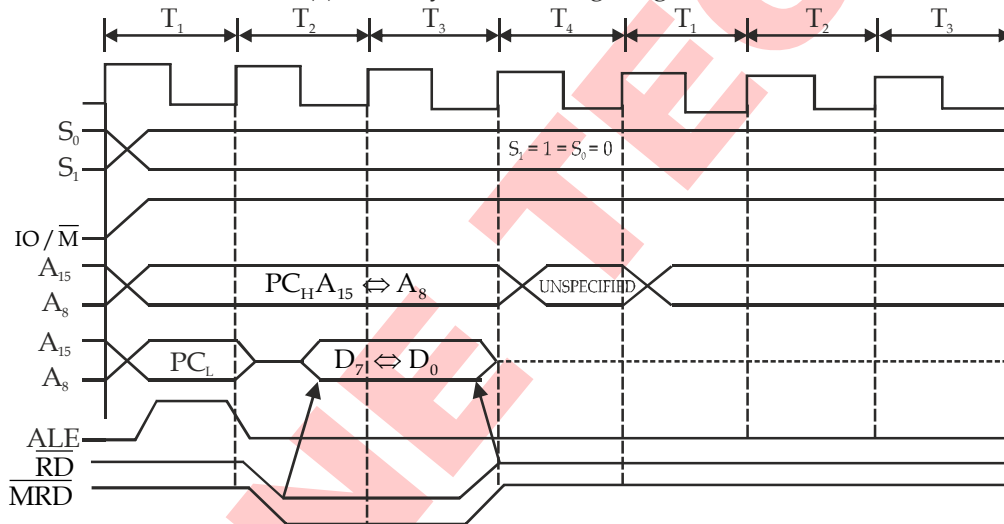


Fig.:- Opcode Fetch



(a) Memory Read Timing Diagram



(b) I/O Read Timing diagram

During T_3 state the RD signal becomes high and memory is disabled.

During T_4 state the opcode is sent for decoding and decoded in T_1 . The execution is also completed in T_4 , if the instruction is single byte. More machine cycles are essential for 2- or 3-byte instructions. The 1st machine cycle M_1 , is meant for fetching the opcode. The machine cycles M_2 , and M_3 , are required either to read/write data or address from the memory or I/O devices.

Read Cycle- The high order address (A_{15} - A_8) and low order address (AD_7 - AD_0) are asserted on 1st low going transition of the clock pulse. The timing diagram for IO/M read are shown in Fig. (a) and (b). The A_{15} - A_8 remains valid in T_1 , T_2 , and T_3 i.e., duration of the bus cycle, but AD_7 - AD_0 remains valid only in T_1 . Since it has to remain valid for the whole bus cycle, it must be saved for its use in the T_2 and T_3 .

ALE is asserted at the beginning of T_1 , of each bus cycle and is negated towards the end of T_1 . ALE is active during T_1 , only and is used as the clock pulse to latch the address (AD_7 - AD_0) during T_1 . The RD is asserted near the beginning of T_2 . It ends at the end of T_3 . As soon as the RD becomes active, it forces the the memory or I/O port to assert data. RD becomes inactive towards the end of T_3 , causing the port or memory to terminate the data.

Write Cycle- Immediately after the termination of the low order address, at the beginning of the T_2 , data is asserted on the address/data bus by the processor. WR control is activated near the start of T_2 and becomes inactive at the end of T_3 . The processor maintains valid data until after WR is terminated. This ensures that the memory or port has valid data while WR is active.

It is clear from Fig. (a) and (b) that for READ bus cycle, the data appears on the bus as a result of activating RD and for the WR bus cycle, the time the valid data is on the bus overlaps the time that the WR is active.

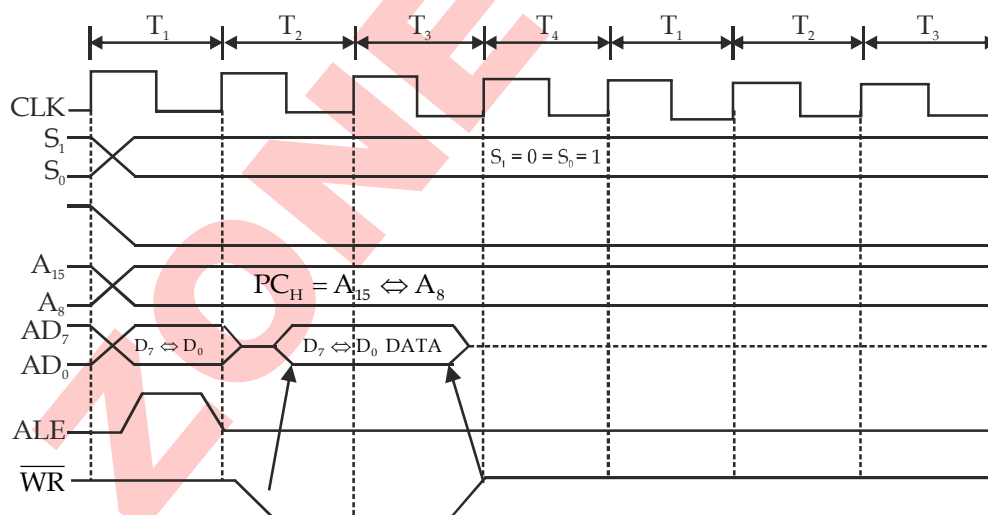
43.

Sol. The name random access memory is an artifact of hardware evolution. Random access means that the stored data can be accessed in any order, which is in contrast to the more restricted access provided by other memory systems, such as tape and disk drive. The access time to any piece of data stored on RAM is essentially the same.

RAM is normally used in computer systems for main memory or primary storage. This is where running programs and the data they use are stored. Moving data from primary storage to the processor requires only a few cycles, although retrieving data from a hard drive can take considerable longer. For this reason, modern operating systems run primarily in RAM, and as they load and run additional applications, they move these programs and their data into RAM for faster processing.

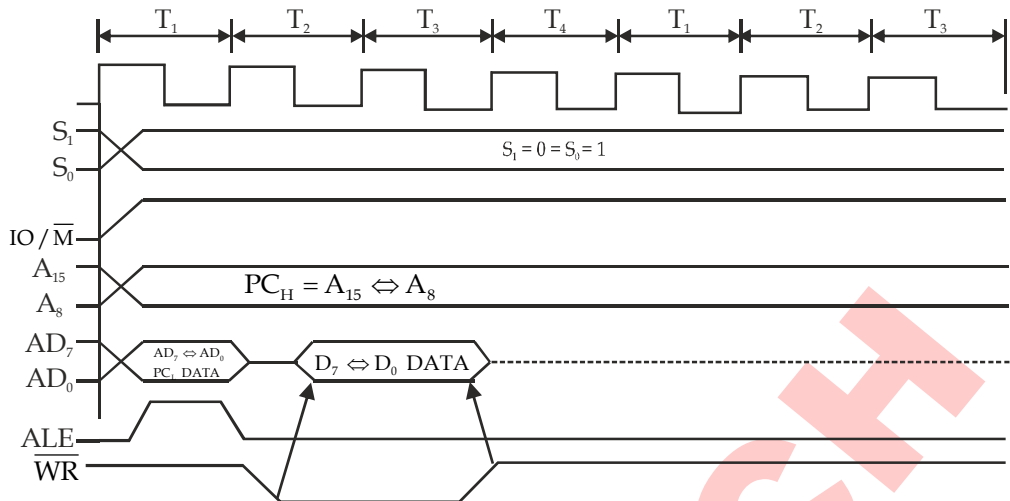
RAM can be categorized as volatile or non-volatile. Volatile means that all data is lost when the chip is powered down. Historically, non-volatile RAM was a tiny part of the market, although consumer electronics have changed that situation. When referring to computer systems, most RAM remains volatile.

Dynamic RAM (DRAM)- Most computers incorporate two types of volatile RAM: static and dynamic. Although both types require constant electrical current to function, they have some important differences. Dynamic RAM is less expensive, and therefore it is the most widely used.



(a) Memory Write Timing Diagram





(b) I/O Write Timing diagram

DRAM uses memory cells consisting of one capacitor and one transistor to store each bit. This is the cheapest and highest in density, so it is used for the main memory in computers. However the electric charge that stores the data in the memory cells slowly leaks off, so the memory cells must be periodically refreshed (rewritten), requiring additional circuitry. The refresh process is automatic and transparent to the user. This refresh requirement is what makes this technology dynamic. DRAM also suffers periodic access limitations, because it cannot be read during the refresh cycle.

Static RAM (SRAM)- Static RAM (SRAM) has the advantage of being faster than DRAM, although the disadvantage is that it is more expensive. SRAM is static in the sense that it doesn't require constant electrical refreshes; however, it still requires constant current to maintain the voltage differentials. SRAM generally requires less power than DRAM, although its power requirements vary depending on clock speed. At higher clock speeds, it can use as much power as DRAM; however, at more moderate speeds, it requires only a fraction of what DRAM uses. When idle, SRAM power requirements are low.

Each bit in a SRAM chip requires a cell of six transistors, forming a digital flip-flop to store each bit, although DRAM needs only one transistor and one capacitor. This means that SRAM cannot achieve the storage densities of the DRAM family. As with DRAM, SRAM chips are mostly large arrays of these cells of transistors.

The two primary applications of SRAM are embedded use and in computers. Embedded use refers to SRAM use in automotive and consumer electronics, industrial equipment and almost all appliances or toys with an electronic user interface. Devices, such as cellphones and music synthesizers, can incorporate several megabytes of SRAM.

SRAM in computer systems is usually delegated to roles where a small amount of high speed memory is required, such as processor caches and I/O buffers.

The two basic operations performed by a memory chip are "read", in which the data contents of a memory word is read out (nondestructively), and "write" in which data is stored in a memory word, replacing any data that was previously stored there.

44.

Sol. Memory Arrays- SRAM memory arrays are arranged in rows and columns of memory cells called wordlines and bitlines, respectively. In SRAMS, the wordlines are made from polysilicon while the bitlines are metal. Each memory cell has a unique location or address defined by the intersection of a row and column. Each address is linked to a particular data input/output pin. The number of arrays on a memory chip is determined by the total size of the memory, the speed at which the memory must operate, layout and testing requirements, and the number of data I/Os on the chip.

Memory Cell- An SRAM memory cell is a bistable flip-flop made up of four to six transistors. The flip-flop may be in either of two states that can be interpreted by the support circuitry to be a 1 or a 0. Many of the SRAMS on the market use a four transistor cell with a polysilicon load. Suitable for medium to high performance, this design has a relatively high leakage current, and consequently high standby current. Four transistor designs may also be more susceptible to various types of radiation induced soft errors. Generally SRAMS are six transistor memory cell (also called a six-device cell) that is highly stable, relatively impervious to soft errors, and has low leakage and standby currents.

Support Circuitry- The memory chip's support circuitry allows the user to read the data stored in the memory's cells, and write data to the cells. This circuitry generally includes:

- Address logic to select rows and columns.
- Translation logic that "reads" the data in a cell and sends that data to the data I/O,
- Write logic that takes the user data applied at the input and stores it in a memory cell.
- Output enable logic to prevent data from appearing at the outputs unless specifically desired.
- Internal counters and registers to keep track of burst address sequences, pipelined data, and other control functions on the chip.
- Clock circuitry to control the timing of the read and write operations and all of their variations.

SRAM Control Signals- Synchronous SRAM operation is relatively straightforward. All operations are controlled by one or more clock signals. These clock signals are generated externally using clock chips or circuits. To operate properly, each of the control signals must be valid when the appropriate clock edge occurs.

The following signals control an SRAM's operation. Other signal pins, used to implement special features such as sleep mode and specific burst addressing sequences, or to satisfy testing requirements, may also be present on a particular SRAM.

45.

Sol. 8085 instruction set can be classified into three types as per the size of instruction:

- One byte Instruction. In this instruction only opcode is used. Example MOV A,B instruction copies the content of B register in Accumulator.
 - Two byte Instruction. In this instruction the first byte is opcode and the second byte is operand. Example MVI B, 04H instruction loads 04H into B register.
 - Three byte Instruction. In this instruction the first byte is opcode and the other two bytes represent address or 16 bit data. Example JMP 2102H instruction loads. 2102H into PC. Instructions are also classified as per their function:
1. Data Transfer group,
 2. Arithmetic group,
 3. Logical group,
 4. Branching group,
 5. Stack and machine control group.

The following notations are used in instructions:

Notations	Meaning
M	Memory location pointed by HL pair
R	8-bit register
RS	Source register
RD	16-bit register pair
RP	Destination register
Addr.	16-bit address
Data 16-bit	16-bit data
Data	8-bit data



ADDRESSING MODES IN 8085

There are 5 addressing modes in 8085:

- (a) Register addressing mode. In this mode the source operand and destination operand are the registers within the microprocessor. Example - MOV A, B loads data of register B into accumulator.
- (b) Immediate addressing mode. In this mode 8 or 16 bit data can be specified as a part of instruction. Example - MVI A, 40H instruction will load accumulator with 40H.
- (c) Direct addressing mode. In this mode 16 bit address of operand is specified. Example - JMP 2100H instruction will load 16 bit address 2100H into PC. (d) Indirect addressing mode. In this mode register pair is used to specify the memory address where the operand is located. Example - LDAX B will load accumulator with the contents of memory location pointed by BC register pair.
- (e) Implied addressing mode. In this mode the opcode itself specifies address of the operand. Example - STC instruction sets carry flag and similarly CMA instruction complements contents of accumulator.