P.R. GOVERNMENT COLLEGE (A), KAKINADA

B.Sc. III Year - Electronics – Semester – 5

PAPER – 5 [Code: EL5202]

w.e.f. 2018 - 19 ADMITTED BATCH

DIGITAL ELECTRONICS

3 Hours/Week [Total: 45 hrs] **Credits: 3**

Teaching hours: 10 hrs

UNIT- I:

Architecture of 8085 Microprocessor:

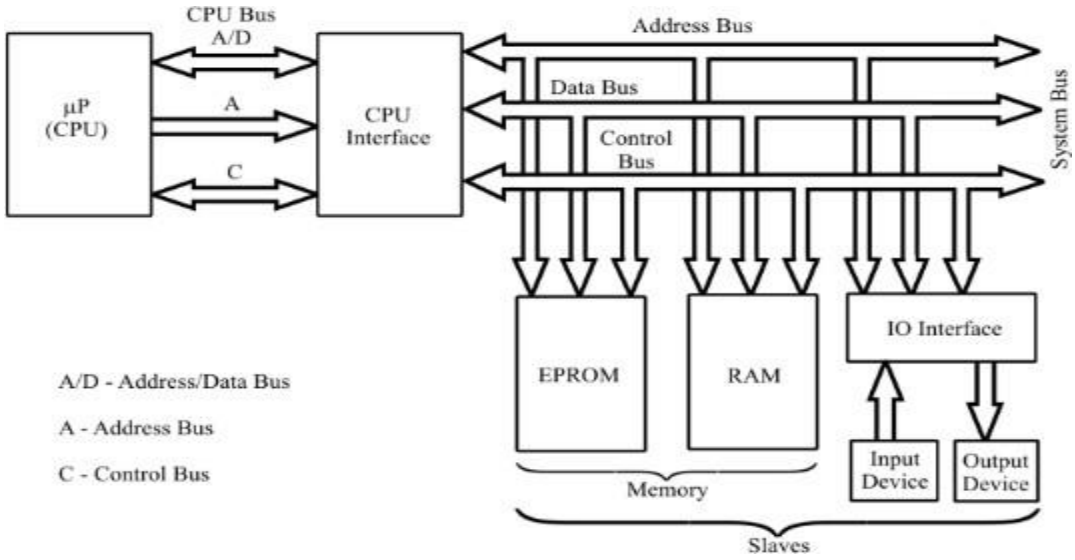
Block diagram of Intel 8085-Register structure- multiplexing &Demultiplexing of address / data bus - Control Signal Generation and status signals - 8085 pin-out diagram & functions - Interrupts

Instruction set of 8085 -Instruction set classification - addressing modes

Index

Introduction to microprocessor and microcomputer architecture:

A *microprocessor* is a semiconductor device consisting of electronic logic circuit manufactured by using very large-scale integration. It is capable of performing various computing function and making decision to change the sequence of program execution. The microprocessor is in many ways similar to the CPU including the control unit and logic circuitry on one chip.



In this system, the microprocessor is the master and all other peripherals are slaves. The master controls all peripherals and initiates all operations. The buses are group of lines that carry data, address or control signals. The CPU interface is provided to demultiplex the multiplexed lines, to generate the chip select signals and additional control signals. The system bus has separate lines for each signal.

All the slaves in the system are connected to the same system bus. At any instant of time communication takes place between the master and one of the slaves. All the slaves have tristate logic and hence normally remain in high impedance state. The processor selects a slave by sending an address. When a slave is selected, it comes to the normal logic and communicates with the processor.

The EPROM memory is used to store permanent programs and data. The RAM memory is used to store temporary programs and data. The input device is used to enter program, data and to operate system. The output device is also used for examining the results. Since the speed of IO devices does not match with speed of microprocessor, an interface device is provided between system bus and IO device.

- **Bit**: A bit is a single binary digit.
- **Word**: A word refers to the basic data size or bit size that can be processed by the arithmetic and logic unit of the processor. A 16-bit binary number is called a word in a 16-bit processor.
- **Bus**: A bus is a group of wires/lines that carry similar information.
- **System Bus**: The system bus is a group of wires/lines used for communication between the microprocessor and peripherals.
- **Memory Word**: The number of bits that can be stored in a register or memory element

is called a memory word.

- **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. For example, if an Address bus have 16 bits. This means that Microprocessor can transfer maximum 16-bit address which means it can address 65,536 different memory locations. This bus is multiplexed with 8-bit data bus. So, the most significant bits (MSB) of address goes through Address bus (A_7-A_{15}) and LSB goes through multiplexed data bus (AD_0-AD_7).
- **Data Bus**: The data bus is used to transfer data between memory and processor or between I/O device and processor. For example, an 8-bit processor will generally have an 8-bit data bus and a 16-bit processor will have 16-bit data bus.
- **Control Bus**: The control bus carry control signals, which consists of signals for selection of memory or I/O device from the given address, direction of data transfer and synchronization of data transfer in case of slow devices.
- **Central Processing Unit**: The CPU consists of ALU (Arithmetic and Logic Unit), Register unit and control unit. The CPU retrieves stored instructions and data word from memory; it also deposits processed data in memory.
- **ALU** (Arithmetic and Logic Unit): This section performs computing functions on data. These functions are arithmetic operations such as additions subtraction and logical operation such as AND, OR rotate etc. Result are stored either in registers or in memory or sent to output devices.
- **Register Unit**: It contains various register. The registers are used primarily to store data temporarily during the execution of a program. Some of the registers are accessible to the uses through instructions.
- **Control Unit**: It provides necessary timing & control signals necessary to all the operations in the microcomputer. It controls the flow of data between the µp and peripherals (input, output & memory). The control unit gets a clock which determines the speed of the µp.

Classification of Microprocessors: Based on their specification, application and architecture microprocessors are classified.

Based on size of data bus:

- 4-bit microprocessor
- 8-bit microprocessor
- 16-bit microprocessor
- 32-bit microprocessor

Microprocessor Initiated operations (or) 8085 Bus organisation:

The MPU (microprocessor unit) performs primarily four operations

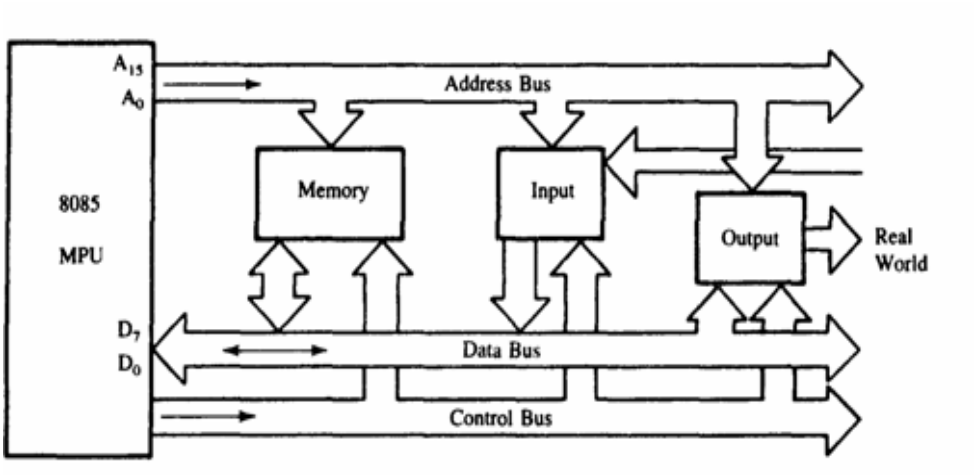
- Memory Read: Reads data (or instruction) from memory.
- Memory Write: Writes data (or instruction) into memory.
- I/O Read: Accepts data from input device.
- I/O Write: Sends data to output device.

All the operations are path of the communication process between the MPU and peripheral devices. To communicate with a peripheral, the MPU needs to perform the following steps.

Step 1: Identify the peripheral location.

Step 2: Transfer data.

Step 3: Providing timing signals.



The 8085 MPU performs these functions using three sets of communication lines called busses.

1. The address bus
2. The data bus
3. The control bus

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 from MPU to peripheral devices. The MPU uses the address bus to identify a peripheral or a memory location (In a computer system, each peripheral or memory location is identified by a binary number called as address). The address is used to carry 16-bit address. The number of address lines of MPU determines its capacity to identify different memory locations or peripherals. The 8085 MPU with its 16 address lines is capable of addressing 2^{16} memory locations.

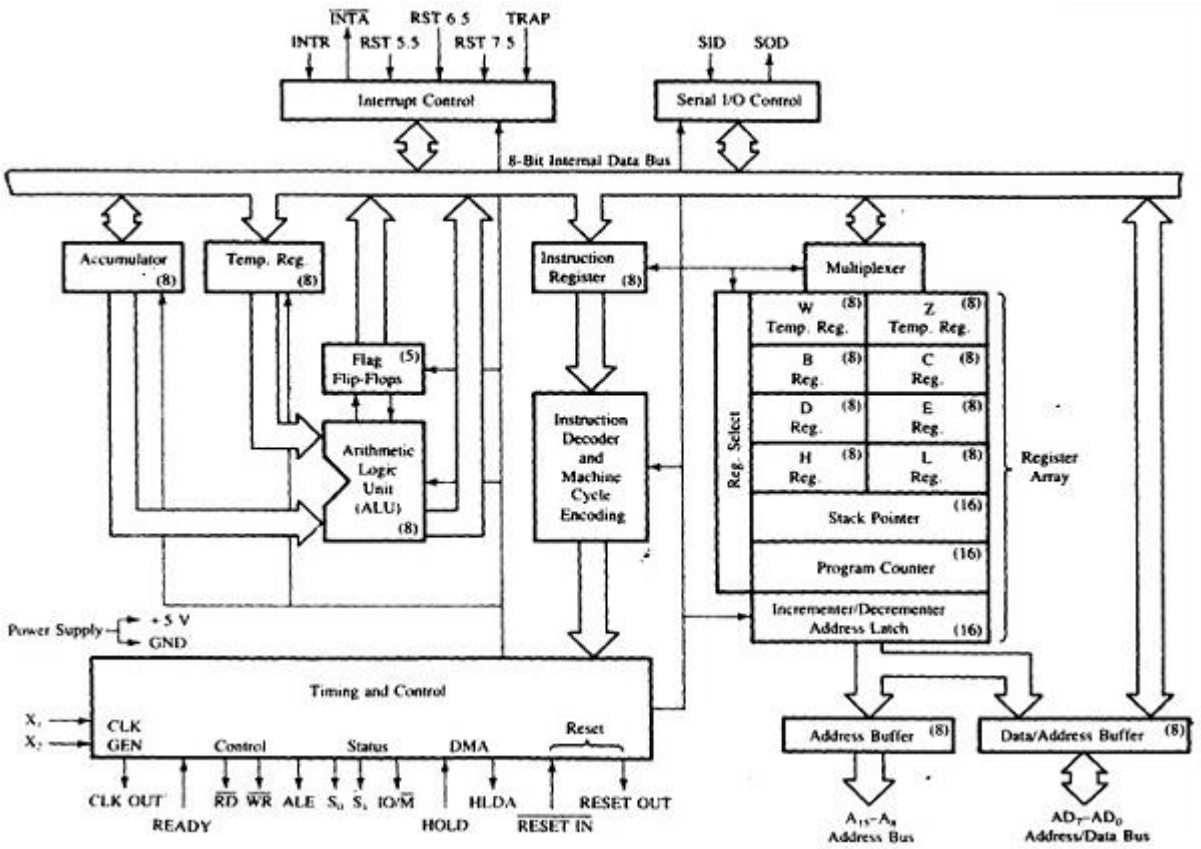
The 16-bit address bus and are generally identified as A_0 - A_{15}. The higher order address lines (A_8 – A_{15}) are unidirectional and the lower order lines (A_0 – A_7) are multiplexed (time-shared) with the eight data bits (D_0 – D_7) and hence, they are bidirectional.

Data bus: The MPU uses the data bus to transfer data. The data bus is group of 8 lines used for data flow. The data refers to any binary information that may include an instruction, an address or number. The lines are bidirectional. i.e., data flow in both directions between the MPU and memory and peripheral devices. the 8-bit data lines enable the MPU to manipulate 8-bit arranging from 00 to FF ($2^8 = 256$). The longest number on data bus is 1111 1111.

Control Bus: The control bus is compressed of various single lines that carry synchronisation signal. The MPU uses such lines to perform the providing timing signals. These are not group of lines like address or data bus, but individual lines that provide a pulse to indicate an MPU operation. The MPU generates specific control signals for memory read of I/O read write instructions.

To read an instruction from a memory location, the MPU places the 16-bit address as the address on the address bus to communicate with memory. 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.

8085 Microprocessor Architecture:



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

- Store 8-bit data.
- Perform arithmetic and logical operations
- Test for conditions.
- Sequence the execution of instructions.
- Store data temporarily in read write memory called stack.

To perform these operations, the microprocessor requires registers, an ALU, control logic and internal bus.

Registers: The 8085 has 6 general purpose registers to store 8-bit data during program execution. These 6 registers are identified as B, C, D, E, H and L. They can be combined as register pairs BC, DE and HL to perform some 16-bit operations. These registers are programmable. It can use to load or transfer data from the registers by using instructions.

Accumulator: The accumulator (A) is an 8-bit register that is part of ALU. It is used to store 8-bit data and to perform ALU operations. The result of an operation is stored in Accumulator. The Accumulator is identified as register A. The data on which operations is to be performed is operand. One of the operands must be Accumulator.

Program Counter (PC): This 16-bit register deals with sequencing the execution of instructions. This register is a memory pointer. Memory locations have 16-bit address. 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 is being fetched, the program counter is automatically 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. It points to a memory location in R/W memory, called stack. The beginning of the stack is defined by loading 16- bit address in the stack pointer.

Timing and control unit: This unit synchronizes all the microprocessor operations with the clock and generates the control signal necessary for communication between the microprocessor and peripherals. The control signals are similar to a sync pulse in an oscilloscope. The $\overline{RD}$ and $\overline{WR}$ signals are sync pulses indicating the availability of data on the data bus.

Instruction register and Decoder: The instruction register and decoder are part of the ALU. When an instruction is fetched from the memory it is loaded in to the instruction register. The decoder decodes the instruction and establishes the sequence of events to follow. The instruction register is not programmable and cannot be accessed by any instruction.

Register Array: There are two additional registers called temporary registers W and Z, which are included in the register array along with programmable registers namely B, C, D, E, H, L, SP and PC. These registers are used to hold 8-bit data during the execution of instructions. However, they are used internally by microprocessor, they are available to the program.

MUX / DEMUX unit: This unit is used to select a register out of all the available registers. This unit behaves as Multiplexer (MUX) when data going from the register to the internal data bus. It behaves as Demultiplexer(DEMUX) when data is coming to a register from the internal data bus of the processor. The register select will behave as the function of selection lines at the Mux / Demux.

Address Buffer: This is an 8-bit unidirectional buffer. It is used to drive external high order address bus ($A_{15} – A_8$). It is also used to tri-state the high order address bus under certain conditions such as reset, hold, halt and when address lines are not in use.

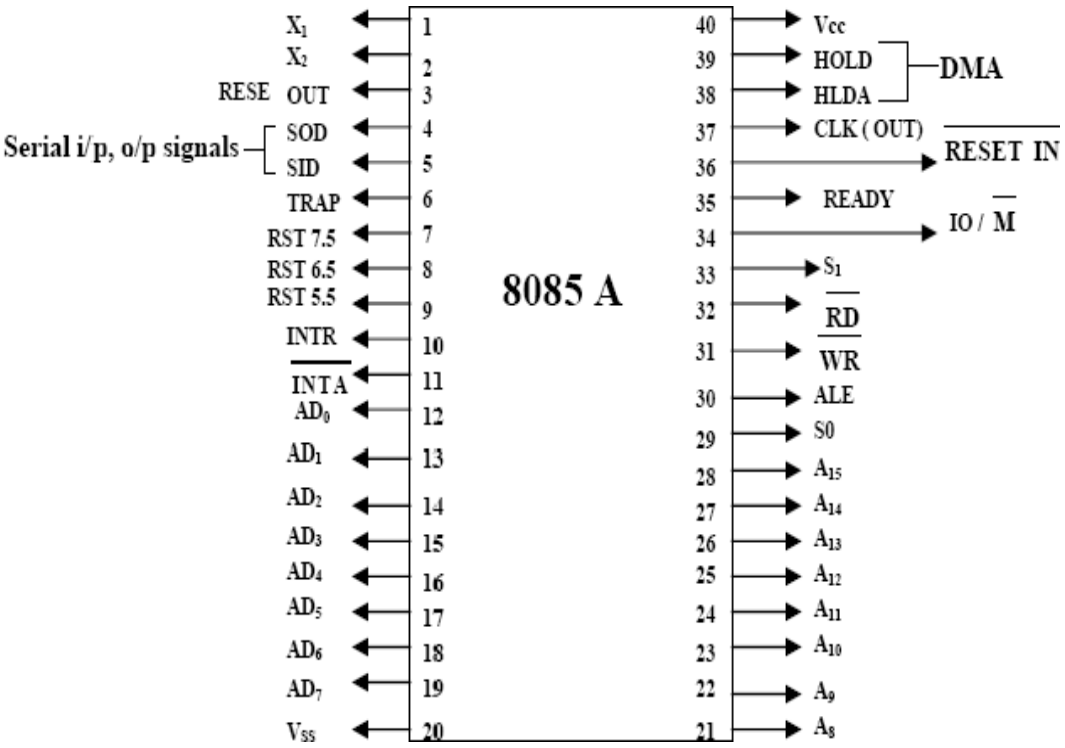
Address/Data Buffer: This is an 8-bit bi-directional buffer. It is used to drive multiplexed address/data bus. i.e., low order address bus (A_7-A_0) and data bus (D_7-D_0). It is also to tri-state the multiplexed address/data bus under certain conditions such as reset, hold, halt and when the bus is not in use.

Incrementer / Decrementer address latch: This 16-bit register is used to increment or decrement the contents of program counter or stack pointer as part of execution of instructions related to them.

Flag register: The ALU includes five flip-flops, which are set (or) reset after an operation according to data condition 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 test data conditions.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
S	**Z**		**AC**		**P**		**CY**

8085 pin diagram:



The 8085 is an 8-bit general purpose microprocessor capable of addressing 64 k of memory. This device has 40 pins requires a +5 V power supply and can operate with a 3 MHz single phase clock. The above figure shows the logic pin-out of the 8085 microprocessor.

All the signals are classified into 7 groups:

- Address bus
- Data bus
- Control & status signals
- Power supply and frequency signals
- Externally initiated signals
- Serial I/O signals
- Interrupt signals

1. Address Bus: The 8085 has 8 signal lines from A_8 – A_{15} which are unidirectional and used as higher order address bus.

2. Multiplexed Address & Data bus: The signal lines from AD_7 – AD_0 are bidirectional, they serve for a dual purpose. They are used as the lower order address bus as well as the data bus.

3. Control and Status signals: This group of signals includes two control signals $\overline{RD}$ and $\overline{WR}$, three status signal IO/$\overline{M}$, S_1 and S_0. To identify the nature of operation one special signal ALE is used to indicate the beginning of the program.
   - $\overline{RD}$ (Read): This is a read control signal (active low). This signal indicates that the selected IO or memory device is to be read and are available on the data bus.
   - $\overline{WR}$(write): This is a write control signal (active low). This signal indicates that the data on the data bus are to be written into a selected memory of IO location.
   - IO/$\overline{M}$, S_1 and S_0: This is a status signal used to differentiate between IO and memory operations. When it is high it indicates an IO operation, when it is low it indicates a memory operation. This signal is combined with read and write to generate IO and memory control signals. S_1 and S_0 indicate the type of machine cycle in progress.
   - ALE(Address Latch Enable): It is an output signal used to give information of AD_0-AD_7 contents. This is a positive going pulse generated any time the 8085 begins its operation. When pulse goes high it indicates that AD_0-AD_7 lines are address. When it is low it indicates that the contents are data. This signal is used primarily to latch the lower order address from the multiplexed bus and generate a separate set of 8-bit address lines.

4. Power supply and clock frequency:
   - Vcc: +5 V DC power supply
   - Vss: Ground reference
   - X1, X2: A crystal is connected at these two pins having frequency of 6 MHz. The frequency is internally divided by 2.
   - CLK Out: Clock output signal can be used as the system clock for the other devices.

5. Externally initiated signals:
   - $\overline{\text{RESET IN}}$: When the signal on this pin is low, the PC is set to 0, the buses are tri-stated and the processor is reset.
   - RESET OUT: This signal indicates that the processor is being reset. The signal can be used to reset other devices.
   - READY: When this signal is low, the processor waits for an integral number of clock cycles until it goes high.
   - HOLD: This signal indicates that a peripheral like DMA (direct memory access) controller is requesting the use of address and data bus.
   - HLDA: This signal acknowledges the HOLD request.

6. Serial I/O signals: The 8085 has two signals to implement the serial transmission.
   - SID (Serial Input Data): This input signal is used to accept serial data bit by bit from the external device by using RIM instruction.
   - SOD (Serial Output Data): This is an output signal which enable the transmission of serial data bit by bit to the external device by using SIM instruction.

7. Interrupt signals: The 8085 has 5 interrupt signals: $\overline{\text{INTA}}$, RST 7.5, RST 6.5, RST 5.5

& TRAP. They can be used to interrupt a program execution.

- INTR: Interrupt request is a general-purpose interrupt.
- $\overline{\text{INTA}}$: This signal is used to acknowledge an interrupt by the processor.
- RST 7.5, RST 6.5, RST 5.5 (Restart interrupt): These are vectored interrupts and have highest priority than INTR interrupt.
- TRAP: This is a non-maskable interrupt and has the highest priority.

Instruction Classification:

An instruction is a binary pattern designed inside a microprocessor to perform a specific function. The entire group of instructions that a microprocessor supports is called instruction set. The 8085 microprocessor includes the instruction set.

Every instruction is having two parts (1) Opcode (2) Operand.

Opcode specifies which operation to be done and operand specifies that data on which operation is performed.

Ex: ADD B

In this example ADD is opcode and B is operand.

Instructions can be classified in to 5 functional categories.

1. Data transfer operations
2. Arithmetic operations
3. Logical operations
4. Branch operations
5. Machine control operations

Data transfer operations: This group of instructions copies data from a location called a source to another location called destination without modifying the contents of the source. The term data transfer is used for copying function. The data transfer can be placed between

- Any two registers.
- Register and memory location.
- Specific 8-bit data to register.
- 8-bit data to memory location.
- Any 16 bit to register pair in unique direction.
- Memory location to register.
- Accumulator and IO device data can transfer.

Arithmetic operations: The instructions perform arithmetic operations such as addition, subtraction, increment / decrement etc.

- Addition: Any 8-bit data or contents of a register or contents of a memory location can be added to the contents of the accumulator and the sum (result) is stored in to accumulator. No two other 8-bit registers can be added directly. It adds 16-bit data directly in register pairs.
- Subtraction: Any 8-bit data or the contents of a register or contents of a memory location can be subtracted from the contents of the accumulator. The subtraction is performed in 2's compliment method. No two other register can be subtracted directly.
- Increment / Decrement: The 8-bit contents of a register or a memory location can be incremented or decremented by 1. The 16-bit contents of a register pair can also be incremented or decremented by 1. The increment or decrement operations differs from addition and subtraction.

Logical operations: The instructions perform various logical operations with the contents of the accumulator.

- AND, OR, XOR: Any 8-bit data or contents of a register or memory location can be logically ANDed, ORed, XORed with the contents of the accumulator. The results are stored in accumulator.
- Rotate: Each 8 bit in the accumulator can be shifted either left or right to the next position.
- Compare: Any 8 bit or contents of register or memory location can be comparing for equality, greater than or less than with the contents of the accumulator.
- Compliment: The contents of the accumulator can be complimented. i.e., all the zeros are replaced by 1's and all 1's are replaced by 0's.

Branch operations: The group of instructions alters the sequence of program execution either conditionally or unconditionally.

- Jump: Conditional jumps are the decision-making process in programming. These instruction tests for a certain condition and altered the program sequence, when the condition is meet. In addition, the instruction set includes and instruction called unconditional jump.
- Call, Return and Restart: These instructions change the sequence of a program either by calling a subroutine or returning from a subroutine. The conditional call and return instructions also can test conditional flags.

Machine control operations: The instructions in this group control execution of other instructions and control operations like interrupt, halt etc.

Addressing Modes: Each instruction specifies an operation to be performed on certain data. Every instruction consists of two parts Opcode and Operand. There are different techniques by which the address of data to be operated upon, may be specified. The techniques are known as addressing modes. In 8085 the following 5 addressing modes are used.

- Direct addressing mode
- Register addressing mode
- Register indirect addressing mode
- Immediate addressing mode
- Implicit addressing mode

Direct addressing mode: In this mode the address of the operand is directly specified within the instruction itself. It is a 3-byte instruction. The only exception being IN and OUT instructions.
Ex: LDA 4050 H – Load the contents of memory location 4050 in to accumulator.
STA 5000 H – Store the contents of accumulator in to memory location 5000 H.
IN 01 H – Read the data from port whose address is 01

Register addressing mode: In this mode the operand is one of the general-purpose registers. In this mode a copy of a byte or word from source register to destination register. Only the registers need to be specified as the address of the operand. These are 1-byte instructions.
Ex: MOV A, B – Move the contents of register B to Accumulator.

ADD B – Add the contents of register B to Accumulator.

Register Indirect Addressing Mode: The 16-bit address location of the operand stored in a register pair is given in the instruction. The address of the operand is given in an indirect way with the help of a register pair hence it is called register indirect addressing mode. The mode of instructions transfers a byte or word between a register and a memory location. These are 1-byte instructions.
Ex: MOV A, M – Move the contents of memory location pointed by the HL pair to Accumulator.

Immediate Addressing Mode: The 8 bit or 16-bit data is specified in the instruction itself as one of the operand. i.e., the operand is specified within the data. If the data is 8-bit, the instruction will be two bytes. If the immediate data is 16-bit then the instruction will be 3 bytes.
Ex: MVI A, Data 8 – Move 8-bit data immediately to the accumulator.
LXI B, Data 16 – Load the BC register pair immediate with the 16-bit data.

Implicit Addressing Mode: There are certain instructions that operate only on operand. Such instructions assume that the operand is in the accumulator and therefore need not specify any address.
Ex: CMA – Compliment the contents of Accumulator
Ex: RLC – Rotate accumulator left by 1.

Instruction Formats: An instruction is a command to the microprocessor to perform a given task on specified data. Each instruction has two parts i.e., Opcode and Operand. The 8085-instruction set is classified in to three groups according to the word size.

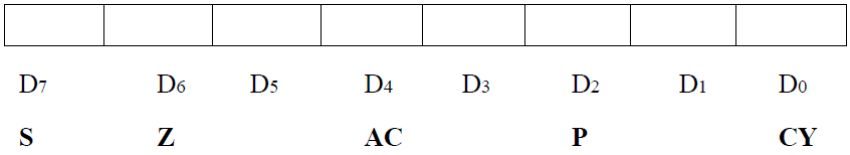
- One-byte instruction.
- Two-byte instruction.
- Three-byte instruction.

Instructions are commonly record in terms of bytes rather than words.

One-byte instruction: A one-byte instruction includes the opcode and the operand in the same byte. The instructions are stored in 8-bit binary format in memory.
Ex: ADD B, CMP A.
Two-byte instruction: In a two-byte instruction, byte 1 contains opcode and byte 2 contains operand.
Ex: MVI A, Data 8.
Three-byte instruction: In a three-byte instruction, byte 1 specifies the opcode and byte 2 & byte 3 represents the operand. The byte 2 is lower order address and byte 3 is a higher order address.
Ex: JMP 4025 – JMP is meant for byte 1, 25 is meant for byte 2 & 40 is meant for byte 3.

Flag Register in 8085: The ALU performs the computing functions, it includes the accumulator, the temporary register, the arithmetic and logic circuits, and five flags. The temporary register is used to hold data during an arithmetic/logic operation. The result is stored in the accumulator, and the flags are set or reset according to the result of the operation. The flags are affected by the arithmetic and logic operations in the ALU. The ALU includes five

flip-flops that are set or reset according to data conditions in the accumulator and other registers. The bit positions in the flag register are as follows.



The conditions of the flags are as follows.

1. Sign Flag(S): This is used in sign magnitude representation. After any operation if MSB of the result is 1, then the number is treated as negative number. If MSB of the result is 0, then the number is treated as positive number. If MSB=1 the sign flag is set, if MSB=0 the sign flag is reset.
2. Zero Flag(Z): If the result of ALU is zero the zero flag is set otherwise it is reset. This flag is modified by the results in the accumulator as well as in the other registers.
3. Auxiliary Carry Flag(AC): This flag is used in BCD number system (0-9). For any operation of ALU, if a carry is generated to D_3 (lower nibble) and forwarded to D_4 (upper nibble) then auxiliary carry flag is set otherwise it is reset. This is the only flag register which is not accessible by the programmer.
4. Parity Flag(P): If result consists of odd number of 1's, then it is said to be odd parity. If the result consists of even number of 1's, then it is said to be even parity. If the result is even parity then the flag is set, if the result is odd parity then the flag is reset.
5. Carry Flag(CY): Carry flag is also called borrow flag. If an operation performed in ALU generates the carry from MSB to next stage then CY flag is set, otherwise it is reset. During subtraction (A-B), if A > B the flag is reset and if A < B the flag is set.

8085 Interrupts:

Interrupt is the mechanism by which the processor is made to transfer control from its current program execution to another program having higher priority. In the microprocessor-based system the interrupts are used for data transfer between the peripheral and the microprocessor.

The program or the routine that is executed upon interrupt is called interrupt service routine (ISR). After execution of ISR, the processor must return to the interrupted program.
In general interrupts can be classified in to 3 types.

- Hardware and Software interrupts
- Vectored and Non-vectored interrupts.
- Maskable and Non-maskable interrupts.

Hardware Interrupts: Interrupts initiated by external hardware by sending an appropriate signal to the interrupt pin of the processor is called hardware interrupt. The 8085 processor has five interrupt pins TRAP, RST 7.5, RST 6.5, RST 5.5 and INTR.

Software Interrupts: Software interrupts are program instructions. These instructions are inserted at desired locations in a program. While running a program, if a software interrupt instruction is encountered then the process initiates an interrupt. The 8085 processor has 8 software interrupts namely RST 0, RST 1, RST 2, RST 3, RST 4, RST 5, RST 6 & RST 7. These are vectored interrupts. Software interrupts cannot be masked or disabled.

Vectored Interrupts: When an interrupt signal is accepted by the processor, and the program control automatically branches to a specified address called vector address and the interrupt is called vectored interrupt.
Non-vectored Interrupts: In non-vectored interrupts the interrupting device should supply the address of the ISR (interrupt service subroutine) to be executed in response to the interrupt.
Maskable Interrupts: The processors have the facility for accepting or rejecting hardware interrupts. The interrupts whose request can be accept or rejected by the processor is called as maskable interrupts.
Non-maskable Interrupts: The interrupts whose request has to be definitely accepted (it cannot be rejected) by the processor are called non-maskable interrupts.

The 8085 MPU has five interrupt pins namely TRAP, RST7.5, RST6.5, RST5.5 and INTR. If the signals on these interrupt pins go to HIGH simultaneously, then TRAP will be services first followed by RST7.5, RST6.5, RST5.5 and INTR. Here once an interrupt is serviced, all the interrupts except TRAP is disabled. They can also be enabled or disabled simultaneously by executing the EI or DI instruction respectively.

TRAP: TRAP is a nonmaskable interrupt. It cannot be enabled or disabled by any instruction. The 8085 responds to TRAP interrupt when a sustained HIGH level with a low to high transition. If this condition occurs, then the 8085 completes execution of the current instruction, pushes the program counter on to the stack. TRAP interrupt is cleared by the falling edge of the signal on the pin.

RST7.5: RST 7.5 is a maskable interrupt. It can be enabled or disabled using the SIM or EI/DI instruction. The 8085 responds to RST7.5 interrupt for a low to high transition. If this condition occurs, then the 8085 completes execution of the current instruction, pushes the program counter on to the stack.

RST6.5: RST 6.5 is a maskable interrupt. It can be enabled or disabled using the SIM or EI/DI instruction. RST6.5 is HIGH level sensitive. If this condition occurs, then the 8085 completes execution of the current instruction, pushes the program counter on to the stack.

RST5.5: RST 5.5 is a maskable interrupt. It can be enabled or disabled using the SIM or EI/DI instruction. RST5.5 is HIGH level sensitive. If this condition occurs, then the 8085 completes execution of the current instruction, pushes the program counter on to the stack.

INTR: INTR is a maskable interrupt having the lowest priority among all interrupts. It can be enabled or disabled by EI or DI instruction. This is also called handshake interrupt. INTR is HIGH level sensitive when no other interrupts are active and the signal on the INTR pin is HIGH, the 8085 completes execution of the current instruction, and generates an interrupt acknowledge, $\overline{\text{INTA}}$, LOW pulse on the control bus. When it is active, the program counter (PC) will stop incremented and an interrupt acknowledge signal is issued by the processor.

$\overline{\text{INTA}}$: This signal is generated by microprocessor in response to the INTR. When microprocessor accepts the INTR, it executes an INTA machine cycle.

Instruction set of 8085:

Instruction	Flags	Meaning	Opcode	Operand	Example	Addressing Mode	Byte
Data transfer instructions							
MOV r_1,r_2	None	Copy the contents of register r_2 to register r_1	MOV	r_1, r_2	MOV B,C	Register addressing	1
MOV r,M	None	Copy the contents of memory(HL registers) to register	MOV	r, M	MOV B,M	Register indirect add	1
MOV M,r	None	Copy the contents of register to memory(HL registers)	MOV	M, r	MOV M,B	Register indirect add	1
MVI r, data 8	None	Store 8-bit data in destination register	MVI	r, data 8	MVI A,32	Immediate	2
MVI M, data 8	None	Store 8-bit data in a memory location	MVI	M, data 8	MVI M,32	Immediate	2
LXI Rp,data 16	None	Loads 16-bit data in to the register pair	LXI	Rp,data 16	LXI H, 2020	Immediate	3
LDA 16-bit Addr	None	The contents of 16-bit address are copied in to accumulator	LDA	16-bit Addr	LDA 8000	Direct Addr	3
STA 16-bit Addr	None	The contents of accumulator are copied in to the specified 16-bit address location	STA	16-bit Addr	STA 8050	Direct Addr	3
LHLD 16-bit Addr	None	Copies the contents of pointed 16-bit memory address in to register L and copies the content of the next memory address in to register H.	LHLD	16-bit Addr	LHLD 8000	Direct Addr	3
SHLD 16-bit Addr	None	The contents of register L are stored into the specified 16-bit memory address and the content of register H are stored in to the next memory location	SHLD	16-bit Addr	SHLD 8050	Direct Addr	3
LDAX Rp	None	Copies the contents of designated register pair in to Accumulator	LDAX	Rp	LDAX B	Register indirect	1
STAX Rp	None	Copies the contents of Accumulator in to designated register pair.	STAX	Rp	STAX B	Register indirect	1

XCHG	None	The 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	XCHG		XCHG	Register Addr	1
Arithmetic group of instructions							
ADD r	All	Add the contents of specified register to the contents of accumulator and store result in the accumulator.	Add	r	Add B	Register addressing	1
ADD M	All	Add the contents of memory location specified by HL register pair to the contents of accumulator.	Add	M	Add M	Register indirect addressing	1
ADI data 8	All	This instruction adds the given 8-bit data to the contents of accumulator.	ADI	data 8	ADI 01	Immediate addressing	2
ADC r	All	The contents of specified register and the carry flag are added to the contents of accumulator.	ADC	r	ADC r	Register addressing	1
ADC M	All	The contents of specified memory location specified by HL register pair and the carry flag are added to the contents of accumulator.	ADC	M	ADC M	Register indirect addressing	1
ACI data 8	All	The 8-bit data and the carry flag are added to the contents of the accumulator.	ACI	data 8	ACI 05	Immediate addressing	2
SUB r	All	The contents of the register are subtracted from the contents of the accumulator.	SUB	r	SUB B	Register addressing	1
SUB M	All	The contents of the memory location pointed by HL register pair are subtracted from the contents of the accumulator.	SUB	M	SUB M	Register indirect addressing	1
SUI data 8	All	The 8-bit data is subtracted from the contents of the accumulator.	SUI	data 8	SUI 45	Immediate addressing	2
SBB r	All	The contents of register and the Borrow flag are subtracted from the contents of the accumulator.	SBB	r	SBB B	Register addressing	1

SBB M	All	The contents of memory location specified by HL register and the borrow flag are subtracted from the contents of the accumulator.	SBB	M	SBB M	Register indirect addressing	1
SBI data 8	All	The 8-bit data and the borrow flag are subtracted from the contents of the accumulator.	SBI	data 8	SBI data 8	Immediate addressing	2
DAD Rp	Carry	The 16-bit contents of the specified register pair are added to the contents of the HL register pair and the sum is stored in the HL register	DAD	Rp	DAD B	Register Addressing	1
Increment / Decrement group of operations							
INR r	Except carry	The contents of the specified register are incremented by 1 and the result is stored in the same place.	INR	r	INR B	Register addressing	1
INR M	Except carry	The contents of memory location specified by HL register are incremented by 1	INR	M	INR M	Register indirect	1
INX Rp	None	The contents of the specified register pair are incremented by 1.	INX	R	INX R	Register addressing	1
DCR r	Except carry	The contents of the specified register are decremented by 1 and the result is stored in the same place.	DCR	r	DCR B	Register addressing	1
DCR M	Except carry	The contents of memory location specified by HL register are decremented by 1.	DCR	M	DCR M	Register indirect	1
DCX r	None	The contents of the specified register pair are decremented by 1	DCX	r	DCX B	Register addressing	1
Logical Operations							
ANA r	All	The contents of accumulator are logically ANDed with the contents of the register	ANA	r	ANA B	Register addressing	1

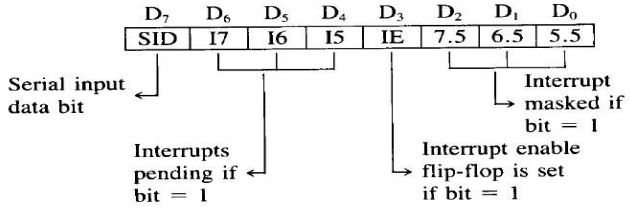
ANA M	All	The contents of memory location specified by HL register are logically ANDed with the contents of the Accumulator.	ANA	M	ANA M	Register indirect addressing	1
ANI data 8	All	The contents of the accumulator are logically ANDed with 8-bit data. And the results is placed in the accumulator	ANI	Data 8	ANI 45	Immediate addressing	2
ORA r	All	The contents of accumulator are logically ORed with the contents of the register	ORA	r	ORA B	Register addressing	1
ORA M	All	The contents of memory location specified by HL register are logically ORed with the contents of the Accumulator.	ORA	M	ORA M	Register indirect addressing	1
ORI data 8	All	The contents of the accumulator are logically ORed with 8-bit data. And the results are placed in the accumulator	ORI	Data 8	ORI 45	Immediate addressing	2
XRA r	All	The contents of accumulator are logically Exclusive ORed with the contents of the register	XRA	r	XRA B	Register addressing	1
XRA M	All	The contents of memory location specified by HL register are logically Exclusive ORed with the contents of the Accumulator.	XRA	M	XRA M	Register indirect addressing	1
XRI data 8	All	The contents of the accumulator are logically Exclusive ORed with 8-bit data. And the results is placed in the accumulator	XRI	Data 8	XRI 45	Immediate addressing	2
CMA	None	The contents of accumulator are complimented	CMA		CMA	Implicit	1
CMC	Carry	The carry flag is complimented	CMC		CMC	Implicit	1
STC	Carry	The carry flag is set to 1	STC		STC	Implicit	1
RLC	Carry	Each binary bit of the accumulator is rotated left by one position. Bit D_7 is placed in the position of	RLC		RLC	Implicit	1

		D_0 as well as in the carry flag. CY is modified according to bit D_7					
RAL	Carry	Each binary bit of the accumulator is rotated left by one position through the carry flag. Bit D_7 is placed in the carry flag and the carry flag is placed in the least significant position D_0. CY is modified according to bit D_7	RAL		RAL	Implicit	1
RRC	Carry	Each binary bit of the accumulator is rotated right by one position. Bit D_0 is placed in the position of D_7 as well as in the carry flag. CY is modified according to bit D_0.	RRC		RRC	Implicit	1
RAR	Carry	Each binary bit of the accumulator is rotated right by one position through the carry flag. Bit D_0 is placed in the carry flag and the carry flag is placed in the most significant position D_7. CY is modified according to bit D_0.	RAR		RAR	Implicit	1
CMP r	All	The contents of the register are compared with the contents of the accumulator. Both contents are preserved.	CMP	r	CMP B	Register addressing	1
CMP M	All	The contents of the memory location are compared with the contents of the accumulator. Both contents are preserved.	CMP	M	CMP M	Register indirect	1
CPI data 8	All	The 8 bit data is compared with the contents of the accumulator. The values being compared remain unchanged.	CPI	Data 8	CPI 50	Immediate addressing	2
Note: If (A) < (reg / mem / data 8): carry flag is set; If (A) = (reg / mem / data 8): Zero flag is set; If (A) > (reg / mem / data 8): carry and zero flags are reset.							
Brach group of instructions							

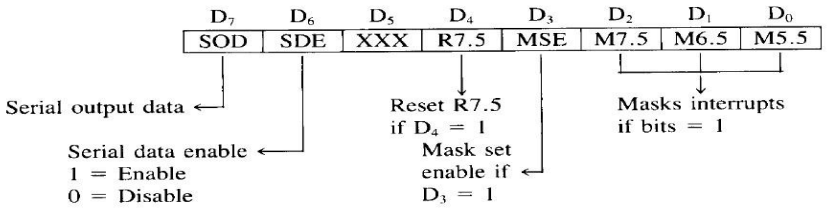
Jmp Addr	None	Transfers the program sequence to the described memory address. It is an unconditional jump. This instruction Jump to the specified address	JMP	Addr	JMP 2050	Immediate	3
JC Addr	Carry	Transfers the program sequence to the described memory address only if the condition is satisfied. It is a conditional jump. These instruction Jumps to the address if carry flag is 1	JC	Addr	JC 2060	Immediate	3
JNC Addr	Carry	Jumps to the address if carry flag is 0	JNC	Addr	JNC 2060	Immediate	3
JZ Addr	Zero	Jumps to the address if zero flag is 1	JZ	Addr	JZ 2055	Immediate	3
JNZ Addr	Zero	Jumps to the address if zero flag is 0	JNZ	Addr	JNZ 2050	Immediate	3
JPE Addr	Parity	Jumps to the address if parity flag is 1	JPE	Addr	JPE 2050	Immediate	3
JPO Addr	Parity	Jumps to the address if parity flag is 0	JPO	Addr	JPO 2060	Immediate	3
JM Addr	Sign	Jumps to the address if sign (Minus) flag is 1	JM	Addr	JM 2020	Immediate	3
JP Addr	Sign	Jumps to the address if sign (Plus) flag is 0	JP	Addr	JP 2030	Immediate	3
CALL Addr	None	The program sequence is transferred to the memory location specified by the 16-bit address. Before transferring, the address of the next instruction after CALL is pushed onto the stack.	CALL	Addr	CALL 2050	Immediate	3
RET Addr	None	The return instruction transfers the program sequence unconditionally from the subroutine to the calling program.	RET	Addr	RET 2050	Immediate	3

CONTROL INSTRUCTIONS

Opcode	Operand	Description
No operation NOP	none	No operation is performed. The instruction is fetched and decoded. However no operation is executed. Example: NOP
Halt and enter wait state HLT	none	The CPU finishes executing the current instruction and halts any further execution. An interrupt or reset is necessary to exit from the halt state. Example: HLT
Disable interrupts DI	none	The interrupt enable flip-flop is reset and all the interrupts except the TRAP are disabled. No flags are affected. Example: DI
Enable interrupts EI	none	The interrupt enable flip-flop is set and all interrupts are enabled. No flags are affected. After a system reset or the acknowledgement of an interrupt, the interrupt enable flip-flop is reset, thus disabling the interrupts. This instruction is necessary to reenable the interrupts (except TRAP). Example: EI
Read interrupt mask RIM	none	This is a multipurpose instruction used to read the status of interrupts 7.5, 6.5, 5.5 and read serial data input bit. The instruction loads eight bits in the accumulator with the following interpretations. Example: RIM



Set interrupt mask SIM	none	This is a multipurpose instruction and used to implement the 8085 interrupts 7.5, 6.5, 5.5, and serial data output. The instruction interprets the accumulator contents as follows. Example: SIM



- SOD — Serial Output Data: Bit D_7 of the accumulator is latched into the SOD output line and made available to a serial peripheral if bit $D_6 = 1$.
- SDE — Serial Data Enable: If this bit = 1, it enables the serial output. To implement serial output, this bit needs to be enabled.
- XXX — Don't Care
- R7.5 — Reset RST 7.5: If this bit = 1, RST 7.5 flip-flop is reset. This is an additional control to reset RST 7.5.
- MSE — Mask Set Enable: If this bit is high, it enables the functions of bits D_2, D_1, D_0. This is a master control over all the interrupt masking bits. If this bit is low, bits D_2, D_1, and D_0 do not have any effect on the masks.
- M7.5 — D_2 = 0, RST 7.5 is enabled.
  = 1, RST 7.5 is masked or disabled.
- M6.5 — D_1 = 0, RST 6.5 is enabled.
  = 1, RST 6.5 is masked or disabled.
- M5.5 — D_0 = 0, RST 5.5 is enabled.
  = 1, RST 5.5 is masked or disabled.

QUESTION BANK

ESSAY ANSWER TYPE QUESTIONS

1. Draw the pin diagram of 8085 & explain each pin functioning.
2. Draw the Block diagram of Intel 8085 and explain each block.
3. Write in detail about Instruction set classification of 8085.
4. Write about the addressing modes of 8085 with give examples.

SHORT ANSWER TYPE QUESTIONS

5. Explain 8085 bus organization.
6. Write a note on software, hardware, maskable and non-maskable interrupts in 8085.
7. Write a note on hardware interrupts.
8. Write a short note on 8085 flag register.
9. Write a short note on Instruction formats.