8085 INSTRUCTION SETS

a) **Data Transfer Instructions**
b) **Arithmetic Instructions**
c) **Logical Instructions**
d) **Rotate Instructions**
e) **Branching Instructions**
f) **Control Instructions**

a) Data Transfer instructions

Mnemonics	Description	Example
MOV Rd, Rs	• Copies the content of source register Rs into destination register Rd • Rs and Rd can be A, B, C, D, E, H, L	MOV A, B
MOV Rd, M	• Copies the content of memory location M into the destination register Rd • Rd can be A, B, C, D, E, H, L • The memory location M is specified by HL pair	MOV A, M
MOV M, Rs	• Copies the content of register Rs into memory location M • Rs can be A, B, C, D, E, H, L • The memory location M is specified by HL pair	MOV M, A
MVI Rd, 8-bit	• The 8-bit data is stored in the destination register Rd • Rd can be A, B, C, D, E, H, L	MVI A, 32H
MVI M, 8-bit	• The 8-it data is stored in memory location M • M is specified by HL pair	MVI M, 32H
LDA 16-bit	• Copies the content of memory	LDA 2015H

(Load Accumulator Direct)	location specified by 16-bit address into A	A ← [2015]
STA 16-bit (Store Accumulator Direct)	• Copies the content of A into 16-bit memory address	STA 2015H A → [2015]
LDAX Rp (Load Accumulator Inirect)	• Copies the content of memory location specified by register pair Rp into A • Rp can be B or D i.e. BC pair or DE pair	LDAX B
STAX Rp (Store Accumulator Indirect)	• Copies the content of A into 16-bit memory address specified by register pair Rp • Rp can be B or D i.e. BC pair or DE pair	STAX B
LXI Rp, 16-bit (Load Register Pair)	• Loads 16-bit data into register pair • Rp can be B, D or H i.e. BC pair, DE pair or HL pair	LXI H, 2015H L ← 15 H ← 20
IN 8-bit	• The data from i/p port specified by 8-bit address is transferred into A	IN 40H A ← [40] 40H is address of input port
OUT 8-bit	• The data of A is transferred into output port specified by 8-bit address	OUT 10H A → [10] 10H is address of output port
XCHG	• Exchange the content of HL pair with DE pair i.e. the content of H and D are exchanged whereas content of L and E are exchanged	XCHG H ↔ D L ↔ E

b) Arithmetic Instructions

Mnemonics	Description	Example
ADD R/M (add register/memory)	• The content of register /memory (R/M) is added to the A and result is stored in A • The memory M is specified by HL pair	ADD B A ← A+B ADD M A ← A+M
ADC R/M (add with carry)	• The content of register /memory (R/M) is added to the A along with carry flag CF and result is stored in A • The memory M is specified by HL pair	ADC B A ← A+B+CF ADC M A ← A+M+CF
ADI 8-bit (add immediate)	• The 8-bit data is added to A and result is stored in A	ADI 32H A ← A+32
ACI 8-bit	• The 8-bit data is added to A	ACI 32H

(add immediate with carry)	along with carry flag CF and result is stored in A	A ← A+32+CF
SUB R/M (subtract register/memory)	• The content of register /memory (R/M) is subtracted from A and result is stored in A • The memory M is specified by HL pair	SUB B A ← A-B SUB M A ← A-M
SBB R/M (subtract with borrow)	• The content of register /memory (R/M) is subtracted from A along with borrow flag BF and result is stored in A • The memory M is specified by HL pair	SBB B A ← A-B-BF SBB M A ← A-M-CF
SUI 8-bit (subtract immediate)	• The 8-bit data is subtracted from A and result is stored in A	SUI 32H A ← A-32
INR R/M (Increment Register/Memory)	• Increment the content of register/memory by 1 • Memory is specified by HL pair	INR B B ← B+1 INR M M ← M+1
DCR R/M (Decrement Register/Memory)	• Decrement the content of register/memory by 1 • Memory is specified by HL pair	DCR B B ← B-1 DCR M M ← M-1
INX Rp (Increment Register Pair)	• Increment the content of register pair Rp by 1	INX H HL ← HL+1
DCX Rp (Decrement Register Pair)	• Decrement the content of register pair Rp by 1	DCX H HL ← HL-1

c) Logical Instructions

Mnemonics	Description	Example
CMP R/M (Compare Register/Memory)	• Compares the content of register/memory with A • The result of comparison is: If A< R/M : Carry Flag CY=1 If A= R/M : Zero Flag Z=1 If A> R/M : Carry Flag CY=0	CMP B CMP M
CPI 8-bit (Compare Immediate)	• Compares 8-bit data with A • The result of comparison is: If A< 8-bit : Carry Flag CY=1 If A= 8-bit : Zero Flag Z=1 If A> 8-bit : Carry Flag CY=0	 CPI 32H

ANA R/M (logical AND register/memory)	• The content of A are logically ANDed with the content of register/memory and result is stored in A • Memory M must be specified by HL pair	ANA B $A \leftarrow A.B$ ANA M $A \leftarrow A.M$
ANI 8-bit (AND immediate)	• The content of A are logically ANDed with the 8-bit data and result is stored in A	ANI 32H $A \leftarrow A.32H$
ORA R/M (logical OR register/memory)	• The content of A are logically ORed with the content of register/memory and result is stored in A • Memory M must be specified by HL pair	ORA B $A \leftarrow A \text{ or } B$ ORA M $A \leftarrow A \text{ or } M$
ORI 8-bit (OR immediate)	• The content of A are logically ORed with the 8-bit data and result is stored in A	ORI 32H $A \leftarrow A \text{ or } 32H$
XRA R/M (logical XOR register/memory)	• The content of A are logically XORed with the content of register/memory and result is stored in A • Memory M must be specified by HL pair	XRA B $A \leftarrow A \text{ xor } B$ XRA M $A \leftarrow A \text{ xor } M$
XRI 8-bit (XOR immediate)	• The content of A are logically XORed with the 8-bit data and result is stored in A	XRI 32H $A \leftarrow A \text{ xor } 32H$

d) Rotate Instructions

Mnemonics	Description	Example
RLC (Rotate Accumulator Left)	• Each bit of A is rotated left by one bit position. • Bit D7 is placed in the position of D0.	RLC
RRC (Rotate Accumulator Right)	• Each bit of A is rotated right by one bit position. • Bit D0 is placed in the position of D7.	RRC
RAL (Rotate Accumulator Left with Carry)	• Each bit of A is rotated left by one bit position along with carry flag CY • Bit D7 is placed in CY and CY in the position of D0.	RAL
RAR (Rotate Accumulator Right with Carry)	• Each bit of A is rotated right by one bit position along with carry flag CY. • Bit D7 is placed in CY and CY in the position of D0.	RLC

e) Branching Instructions

Mnemonics	Description	Example
JMP 16-bit (Unconditional Jump)	The program sequence is transferred to the memory location specified by 16-bit address	JMP C000H
JC	Jump on Carry (CY=1)	
JNC	Jump No Carry (CY=0)	
JP	Jump on Positive (S=0)	
JM	Jump on Negative (S=1)	
JZ	Jump on Zero (Z=1)	
JNZ	Jump No Zero (Z=0)	
JPE	Jump on Parity Even (P=1)	
JPO	Jump on Parity Odd (P=0)	
CALL 16-bit	The program sequence is transferred to the subroutine at memory location specified by the 16-bit address	CALL C000H
RET	The program sequence is transferred from the subroutine program to calling program	RET

f) Control Instructions

Mnemonics	Description	Example
NOP	No operation is performed	NOP
HLT	The CPU finishes executing the current instruction and stops any further execution	HLT

8085 ADDRESSING MODES

The ways by which operands are specified in an instruction are called addressing modes. The different addressing modes of 8085 are:

1. Immediate Addressing Mode

If the data is present within the instruction itself, then it is called immediate addressing mode.

Examples: MVI A, 05H
ADI 55H
LXI H, C000H

2. Register Addressing Mode

If the data is present in the register and the register are specified in an instruction, than it is called register addressing mode.

Example: MOV A, B
ADD B
ANA C

3. Direct Addressing Mode

If the address of the data is specified in the instruction itself, than it is called direct addressing mode.

Example:
```
LDA 2000H
STA 2000H
IN 10H
OUT 01H
```

4. Register Indirect Addressing Mode

If the register pair which contains the address of the data is specified in the instruction, than it is called register indirect addressing mode.

Example:
```
LDAX B
STAX D
MOV M, A
MOV B, M
```

5. Implied Addressing Mode

If the opcode in an instruction tells about the operand, than it is called implied addressing mode.

Example:
```
RAL
RRC
```