Ministry of Higher Education and Scientific Research
العلمي البحث و العالي التعليم وزارة

BADJI MOKHTAR-ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR-ANNABA

جامعة باجي مختار-عنابة

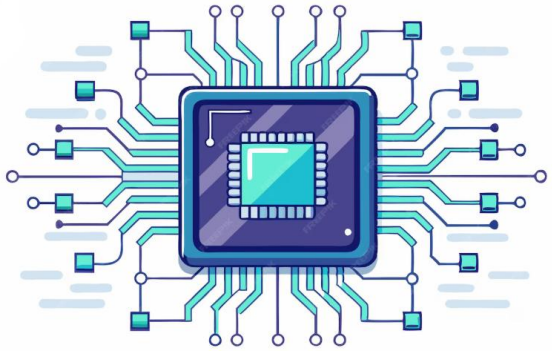
Faculty of technology
Electronics departement

Embedded Computing Systems course

Teaching method : Distance learning

Chapter 2
Instruction set intel 8085

Course 5



Teaching by
Dr. MERABTI Nardjes

EAD3/ DD3
Promotion : 2025/2026

☐ **Instruction set of 8085:** consists of one, two and three byte instructions.

The first byte is always the **opcode**; in two byte instructions the second byte is usually **data**; in three byte instructions the last two byte present **address** or **16-bit data.**

1.Example of One Byte Instruction: MOV B, C whose opcode is 41H which is one byte; This instruction copies the contents of C register in B register

2. Example of two byte instruction: MVI B, 08H.

The opcode for this instruction is 06H and is always followed by a byte data (08H in this case).

This instruction is a two byte instruction which copies immediate data into B register

3. Example of Three Byte Instruction: Opcode Operand Operand

JMP 8200H: The opcode for this instruction is **$C3_H$** and is always followed by 16 bit address (8200_H in this case).

This instruction is a **three byte** which loads 16 bit address into program counter

There are 5 categories: Data Transfer Instruction, Arithmetic Instructions, Logical Instructions, Branching Instructions, Control Instructions

Data Transfer Instructions

- **MOV rd, rs/ MOV M, Rs/ MOV Rd, M:** This instruction copies the contents of the source register into the destination register. The contents of the source register are not altered.

Example: MOV B,A or MOV M,B or MOV C,M

- **MVI R, Data(8-bit) / MVI M, Data(8-bit):** The 8-bit immediate data is stored in the destination register (R) or memory (M), R is general purpose 8 bit register such as A, B, C, D, E, H and L.

• Example: MVI B, 60H or MVI M, 40H

- **XCHG : The contents of register H are exchanged**

with the contents of register D. The contents of register L are exchanged with the contents of register E.

• Example: XCHG

❖ **LDA 16-bit address:** The contents of a memory location, specified by a 16-bit address in the operand, are copied to the accumulator (A); The contents of the source are not altered.

- Example: LDA 2000H

❖ **LDAX Register Pair:** Load accumulator (A) with the contents of memory location whose address is specified by BC or DE or register pair; The contents of either the register pair or the memory location are not altered.

- Example: LDAX D

❖ **LXI Rp, data16:** Loads a 16-bit immediate data into a register pair (rp = B, D, H, or SP).

- Example: LXI H, 2050H

❖ **STA 16-bit address:** The contents of accumulator are copied into the memory location i.e. address specified by the operand in the instruction.

- Example: STA 2000 H

❖ **STAX Register Pair:** Store the contents of accumulator (A) into the memory location whose address is specified by BC Or DE register pair.

- Example: STAX B

Arithmetic Instructions

- **ADD R/ ADD M:** The contents of register or memory are added to the contents of accumulator. The result is stored in accumulator. If the operand is memory location, its address is specified by H-L pair.

• Example: ADD C or ADD M

- **ADC R/ ADC M:** The contents of register or memory and Carry Flag (CY) are added to the contents of accumulator; The result is stored in accumulator. If the operand is memory location, its address is

specified by H-L pair. All flags are modified to reflect the result of the addition.

• Example: ADC C or ADC M

- **ADI 8-bit data :** The 8-bit data is added to the contents of accumulator; The result is stored in accumulator.

• Example: ADI 10 H

- **ACI 8-bit data:** The 8-bit data and the Carry Flag (CY) are added to the contents of accumulator; The result is stored in accumulator.

• Example: ACI 20 H

- **DAD Rp:** Adds the 16-bit register pair (rp) to the HL pair → result stored in HL (HL ← HL + Rp)
- **DAA:** (Decimal Adjust Accumulator): Used after a BCD addition to correct the result in the accumulator (A) so that it becomes a valid BCD (Binary-Coded Decimal) number.

❖ **SUB R/ SUB M**

• The contents of the register or memory location are subtracted from the contents of the accumulator; The result is stored in accumulator; If the operand is memory location, its address is specified by H-L pair.

• Example: SUB B or SUB M

❖ **SBB R/ SBB M:** The contents of the register or memory location and Borrow Flag (i.e.CY) are subtracted from the contents of the accumulator; The result is stored in accumulator. If the operand is memory location, its address is specified by H-L pair.

• Example: SBB C or SBB M

❖ **SUI 8-bit data:** OPERATION: A=A-DATA(8): The 8-bit immediate data is subtracted from the contents of the accumulator; The result is stored in accumulator.

• Example: SUI 45 H

❖ **SBI 8-bit data:** The 8-bit data and the Borrow Flag (i.e. CY) is subtracted from the contents of the accumulator; The result is stored in accumulator.

• Example: SBI 20 H

❖ **INR R/ INR M:** The contents of register or memory location are incremented by 1; The result is stored in the same place; If the operand is a memory location, its address is specified by the contents of H-L pair.

- Example: INR B or INR M

❖ **INX Rp:** This instruction increments the contents of register pair by 1.The result is stored in the same place.

- Example: INX H

❖ **DCR R/ DCR M:** The contents of register or memory location are decremented by 1; The result is stored in the same place. If the operand is a memory location, its address is specified by the contents of H-L pair.

- Example: DCR E or DCR M

❖ **DCX Rp:** This instruction decrements the contents of register pair by 1; The result is stored in the same place.

- Example: DCX D

LOGIC Instructions

- **ANA R/ ANA M:** AND specified data in register or memory with accumulator; Store the result in accumulator (A).
  - Example: ANA B, ANA M
- **ANI 8-bit data:** AND 8-bit data with accumulator (A); Store the result in accumulator (A)
  - Example: ANI 3FH
- **XRA Register (8-bit):** XOR specified register with accumulator; Store the result in accumulator.
  - Example: XRA C
- **XRA M:** XOR data in memory (memory location pointed by H-L pair) with Accumulator; Store the result in Accumulator.
  - Example: XRA M
- **XRI 8-bit data :** XOR 8-bit immediate data with accumulator (A); Store the result in accumulator.
  - Example: XRI 39H
- **ORA Register:** OR specified register with accumulator (A); Store the result in accumulator.
  - Example: ORA B
- **ORA M:** OR H-L pair (i.e. M) with accumulator (A); Store the result in accumulator.
  - Example: ORA M

❖ **ORI 8-bit data:** OR 8-bit data with accumulator (A); Store the result in accumulator.

• Example: ORI 08H

❖ **CMP Register:** CMP M: Compare specified data in register or memory with accumulator (A); Store the result in accumulator.

• Example: CMP D or CMP M

❖ **CPI 8-bit data:** Compare 8-bit immediate data with accumulator (A); Store the result in accumulator.

• Example: CPI 30H

❖ **STC:** It sets the carry flag to 1.

❖ **CMC:** It complements the carry flag.

❖ **CMA:** It complements each bit of the accumulator.

❖ **RLC: Rotate accumulator left:** Each binary bit of the accumulator is rotated left by one position; Bit 7 is placed in the position of Bit 0 as well as in the Carry flag.

CY is modified according to bit 7.

❖ **RRC: Rotate accumulator right:** Each binary bit of the accumulator is rotated right by one position; Bit 0 is placed in the position of bit 7 as well as in the Carry flag.

CY is modified according to bit 0.

Branching Instructions

The branch group instructions allows the microprocessor to change the sequence of program either conditionally or under certain test conditions. The group includes:

- (1) Jump instructions,
- (2) Call and Return instructions,
- (3) Restart instructions,

❖ JUMP address

❖ **Jump unconditionally to the address:** The instruction loads the PC with the address given within the instruction and resumes the program execution from specified location.

• Example: JMP 2000H

Conditional Jumps:

Instruction Code	Description	Condition For Jump
JC	Jump on carry	CY=1
JNC	Jump on not carry	CY=0
JP	Jump on positive	S=0
JM	Jump on minus	S=1
JPE	Jump on parity even	P=1
JPO	Jump on parity odd	P=0
JZ	Jump on zero	Z=1
JNZ	Jump on not zero	Z=0

❖ **CALL address:** Call **unconditionally** a subroutine whose starting address given within the instruction and used to transfer program control to a subprogram or subroutine.

- Example: CALL 2000H

❖ **Conditional Calls:**

Instruction Code	Description	Condition for CALL
CC	Call on carry	CY=1
CNC	Call on not carry	CY=0
CP	Call on positive	S=0
CM	Call on minus	S=1
CPE	Call on parity even	P=1
CPO	Call on parity odd	P=0
CZ	Call on zero	Z=1
CNZ	Call on not zero	Z=0

❖ **RST n:** Restart n (0 to 7): This instruction transfers the program control to a specific memory address. The processor multiplies the RST number by 8 to calculate the vector address (in hexadecimal).

- Example: RST 6

Instruction Code	Vector Address
RST 0	0*8=0000H
RST 1	1*8=0008H
RST 2	2*8=0010H
RST 3	3*8=0018H
RST 4	4*8=0020H
RST 5	5*8=0028H
RST 6	6*8=0030H
RST 7	7*8=0038H

Control Instructions

- **NOP:** No operation: No operation is performed; The instruction is fetched and decoded but no operation is executed.
- **HLT:** Halt: The CPU finishes executing the current instruction and halts any further execution; An interrupt or reset is necessary to exit from the halt state.
- **EI :Enable Interrupts:** Activates the interrupt system allows the processor to recognize maskable interrupts after the next instruction.
- **DI :Disable Interrupts :** Deactivates all maskable interrupts (RST5.5, RST6.5, RST7.5, INTR).

SIM – Set Interrupt Mask: Used to configure or mask interrupts and control serial output (SOD).

RIM – Read Interrupt Mask: Reads the status of interrupt masks, pending interrupts, and the serial input line (SID).

Stack & Input /Output instruction

- **IN 8-bit port address:** Copy data to accumulator from a port with 8-bit address. The contents of I/O port are copied into accumulator; Example: IN 80 H
- **OUT 8-bit port address:** Copy data from accumulator to a port with 8- bit address; The contents of accumulator are copied into the I/O port; Example: OUT 50 H
- **PUSH rp:** Push (store) the contents of a register pair onto the stack:
  - Decrement SP by 1 → store high-order byte of the register pair.
  - Decrement SP again → store low-order byte of the pair.
- **POP rp:** Pop (retrieve) two bytes from the stack into a register pair:
  - Read low byte from memory → store in lower register.
  - Increment SP.
  - Read high byte → store in higher register.
  - Increment SP again.
- **XTHL: Exchange Top of Stack with HL:** Exchange the top two bytes of the stack with the HL register pair: Load L ← (SP) and H ← (SP+1).
- **SPHL : Move H–L contents to Stack Pointer:** Copy the contents of the HL register pair into the Stack Pointer (SP).

Data Transfer instructions:

Instruction	Size (Bytes)	Machine Cycles
MOV R1, R2	1	1 (Opcode Fetch)
MOV M, R	1	2 (OF + Memory Write)
MOV R, M	1	2 (OF + Memory Read)
MVI R, data	2	2 (OF + Memory Read)
LDA addr	3	3 (OF + MR + MR)
STA addr	3	3 (OF + MR + MW)
LHLD addr	3	3 (OF + MR + MR)
SHLD addr	3	3 (OF + MR + MW + MW)
XCHG	1	1 (OF)

R: Register
addr: Address memory
$M=M_{(HL)}$

Arithmetic instructions:

Instruction	Size (Bytes)	Machine Cycles
ADD R ADD M ADI data	1 1 2	1 (Opcode Fetch) 2 (OF + Memory Read) 2 (OF + Memory Read)
SUB R SUB M SUI data	1 1 2	1 (OF) 2 (OF + MR) 2 (OF + MR)
INR R INR M INX Rp	1 1 1	1 (OF) 3 (OF + MR + MW) 1 (OF)
DCR R DCR M DCX Rp	1 1 1	1 (OF) 3 (OF + MR + MW) 1 (OF)
DAD Rp DAA	1 1	1 (OF) 1 (OF)

Rp: Register pair

Logical instructions:

Instruction	Size (Bytes)	Machine Cycles
ANA R	1	1 (Opcode Fetch)
ANA M	1	2 (OF + Memory Read)
XRA R	1	1 (OF)
XRA M	1	2 (OF + MR)
ORA R	1	1 (OF)
ORA M	1	1 (OF + MR)
CMP R CMP M CMA	1 1 1	1 (OF) 2 (OF + MR) 2 (OF)
RLC / RRC / RAL / RAR	1	1 (OF)
STC	1	1 (OF)

Branching instructions:

Instruction	Size (Bytes)	Machine Cycles
JMP addr	3	3 (OF + MR + MR)
JC / JNC / JZ / JNZ / JP / JM	3	3 (if taken) / 2 (if not taken)
CALL addr	3	5 (OF + MR + MR + MW + MW)
CC / CNC / CZ / CNZ / CP / CM	3	5 (if taken) / 2 (if not)
RET	1	3 (OF + MR + MR)
RC / RNC / RZ / RNZ / RP / RM	1	3 (if taken) / 1 (if not)

Control instructions:

Instruction	Size (Bytes)	Machine Cycles
NOP	1	1 (OF)
HLT	1	1 (OF)
EI/ DI	1	1 (OF)
SIM / RIM	1	1 (OF)

➢ Stack & Input /Output instructions:

Instruction	Size (Bytes)	Machine Cycles
PUSH Rp	1	3 (OF + MW + MW)
POP Rp	1	3 (OF + MR + MR)
XTHL	1	5 (OF + MR + MR + MW + MW)
SPHL	1	1 (OF)
IN Port OUT Port	2 2	3 (OF + I/O Read + I/O Read) 3 (OF + I/O Write + I/O Write)