Al-Mustaqbal University

College of Engineering and Technology

Department of Medical Instrumentation Techniques Engineering

Class: Third Class

Subject: Instruction sets & addressing modes .

Lecturer: M.Sc Ali Kareem

Lecture Address: Instruction sets & addressing modes .

2024 – 2025

Weeks 19

Microprocessor interfacing .

BY M.SC. ALI KAREEM AL-JUHAISHI

ali.kareem.obaid@uomus.edu.iq

INSTRUCTION SET OF 8086

Binary to Decimal Number Format 8086 also follows the binary number system similar to 8085. However, it supports both 8-bit and 16-bit operations due to its 16-bit architecture.

BINARY TO DECIMAL NUMBER FORMAT

8	4	2	1	DECIMAL
0	0	0	0	= 0
0	0	0	1	= 1
0	0	1	0	= 2
0	0	1	1	= 3
0	1	0	0	= 4
0	1	0	1	= 5
0	1	1	0	= 6
0	1	1	1	= 7

ali.kareem.obaid@uomus.edu.iq

8	4	2	1	DECIMAL
1	0	0	0	= 8
1	0	0	1	= 9
1	0	1	0	= 10 = A
1	0	1	1	= 11 = B
1	1	0	0	= 12 = C
1	1	0	1	= 13 = D
1	1	1	0	= 14 = E
1	1	1	1	= 15 = F

Examples:

- **Binary:** 0000 0001 → **Decimal:** 1
- **Binary:** 0000 1010 → **Decimal:** 10
- **Binary:** 0011 1100 → **Decimal:** 60
- **Binary:** 1001 0110 → **Decimal:** 150
- **Binary:** 1111 1111 → **Decimal:** 255

What is an Instruction? An instruction is a binary-encoded operation designed to perform a specific task inside a microprocessor. The 8086 processor has a rich set of instructions categorized based on their functionality.

- 8086 has **117 instructions.**
- Each instruction is represented by an 16-bit binary value.

Classification of Instruction Set 8086 instructions are categorized into the following types:

1. **Data Transfer Instructions**
2. **Arithmetic Instructions**
3. **Logical Instructions**
4. **Branching Instructions**
5. **Control Transfer Instructions**
6. **String Manipulation Instructions**
7. **Processor Control Instructions**

1. Data Transfer Instructions

These instructions move data between registers, memory, and I/O ports without altering the data.

- **MOV Destination, Source**: Transfers data between registers and memory.
  - Example: `MOV AX, BX`
- **PUSH Source**: Pushes the register/memory content onto the stack.
  - Example: `PUSH AX`
- **POP Destination**: Pops the top value of the stack into the destination register.
  - Example: `POP BX`

- **XCHG Destination, Source**: Swaps data between registers or memory.
    - Example: `XCHG AX, BX`

2. Arithmetic Instructions

Perform mathematical operations like addition, subtraction, multiplication, and division.

- **ADD Destination, Source**: Adds source to the destination.
    - Example: `ADD AX, BX`
- **SUB Destination, Source**: Subtracts source from the destination.
    - Example: `SUB AX, CX`
- **MUL Source**: Multiplies AX register with source operand (for unsigned multiplication).
    - Example: `MUL CX`
- **DIV Source**: Divides accumulator by source operand.
    - Example: `DIV BX`

3. Logical Instructions

These instructions perform bitwise operations like AND, OR, XOR, and NOT.

- **AND Destination, Source**: Performs bitwise AND.
    - Example: `AND AX, 0F0FH`
- **OR Destination, Source**: Performs bitwise OR.
    - Example: `OR AL, BL`
- **XOR Destination, Source**: Performs bitwise XOR.
    - Example: `XOR CX, DX`

- **NOT Destination**: Performs bitwise NOT.
  - Example: `NOT AX`

4. Branching Instructions

Used to change the flow of execution.

- **JMP Address**: Jumps to the specified address unconditionally.
  - Example: `JMP 2000H`
- **JE/JZ Address**: Jumps if equal/zero flag is set.
  - Example: `JE LOOP`
- **JNE/JNZ Address**: Jumps if not equal/zero flag is clear.
  - Example: `JNE NEXT`
- **CALL Address**: Calls a procedure (subroutine).
  - Example: `CALL FUNCTION`
- **RET**: Returns from a subroutine.
  - Example: `RET`

5. Control Transfer Instructions

These instructions control the execution of the processor.

- **HLT (Halt Instruction)**: Stops the execution of the processor. The processor enters a halt state and remains there until it is reset or interrupted.
  - Example: `HLT`
- **NOP (No Operation)**: Performs no operation but takes up a clock cycle. It is used for timing adjustments or to create delays in the execution of a program.
  - Example: `NOP`

6. String Manipulation Instructions

These instructions handle operations on strings stored in memory.

- **MOVS**: Moves string data.
  - Example: `MOVSB` (moves byte)
- **CMPS**: Compares string bytes/words.
  - Example: `CMPSB`
- **SCAS**: Scans a string for a match.
  - Example: `SCASB`

7. Processor Control Instructions

Used for system-level control operations.

- **STI**: Set interrupt flag (enable interrupts).
- **CLI**: Clear interrupt flag (disable interrupts).
- **STC**: Set carry flag.
- **CLC**: Clear carry flag.