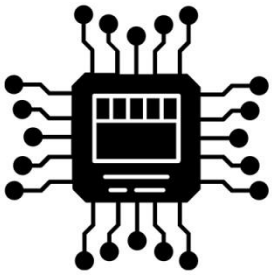


قـســـــــــم علوم الامن السيبراني

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

SUBJECT:

Microprocessor

CLASS:

SECOND

LECTURER:

Dr. Abdulkadhem A. Abdulkadhem

LECTURE: (5)

Addressing Modes

1. Introduction

In any operation performed by the CPU, the required information must be identified and accessed. In computer architecture, this information is known as the **operand**. Therefore, every instruction consists of two essential parts:

- **Opcode**: Specifies the operation to be performed
- **Operand address information**: Specifies where the data is located

The method used to specify the location of operands is called the **Addressing Mode**. Different addressing modes provide flexibility, efficiency, and support for complex data structures such as arrays and strings.

2. Addressing Modes in 8086

The Intel 8086 microprocessor supports **eight addressing modes**:

1. Immediate Addressing Mode
2. Register Addressing Mode
3. Direct Addressing Mode
4. Register Indirect Addressing Mode
5. Based Addressing Mode
6. Indexed Addressing Mode
7. Based-Indexed Addressing Mode
8. Based-Indexed with Displacement Addressing Mode

 Exam Note*:* Classification and identification of addressing modes are frequently asked questions.

2.1. Immediate Addressing Mode

In immediate addressing mode, the operand is **part of the instruction itself**. No memory access is required to fetch the data.

Characteristics:

- Operand value is directly specified
- Fast execution
- No effective address calculation

Examples:

```
MOV AL, 20
MOV AX, 44H
MOV AL, 'A'
MOV CL, 11001110B
```

 Note: If a hexadecimal number starts with a letter, it must be preceded by 0
Example:

```
MOV AL, 0F2H
```

Table (1): Examples of Immediate Addressing Mode

Instruction	Size	Operation
MOV AL, 20	8 bit	Copies 20 decimal (14H) into register AL
MOV BL, 44	8 bit	Copies 44 decimal (2CH) into register BL
MOV AX, 44H	16 bit	Copies 0044H into register AX
MOV BX, 55H	16 bit	Copies a 0055H into register BX
MOV SI, 0	16 bit	Copies 0000H into register SI
MOV CH, 100	8 bit	Copies 100 decimal (64H) into register CH
MOV AL, 'A'	8 bit	Copies ASCII of A into register AL
MOV AH, 1	8 bit	Copies 1 decimal (01H) into register AH
MOV AX, 'AB'	16 bit	Copies ASCII of AB into register AX
MOV DX, 'Ahmed'	16 bit, not allowed	Copies an ASCII Ahmed into register DX
MOV CL, 11001110B	8 bit	Copies 11001110 binary into register CL

2.2. Register Addressing Mode

In this mode, both source and destination operands are **registers**.

Characteristics:

- Fastest addressing mode
- No memory access
- No effective address computation

Rules:

- Registers must be of the same size
- CS cannot be used as a destination
- Only one operand can be a segment register

Example:

```
MOV AX, BX
MOV CL, AL
```

2.3. Direct Addressing Mode

In this type of the addressing modes, the address of the memory location that holds the operand is included in the instruction. There are two basic forms of direct data addressing:
Direct addressing: this form specifically involves transferring data between a memory location and the AL (8-bit) or AX (16-bit). The instruction directly contains the memory address of the data.
Example:

```
MOV AX, [1234H]       (Move the word from memory location 1234H into the AX register).
```

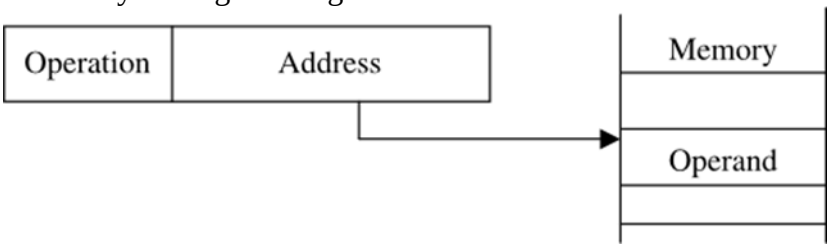
- The effective address is the displacement value.
- The segment register that is used by default is the data segment register (DS).

Displacement addressing: this is a more versatile اكثر مرونة form that can be used with almost any instruction that involves memory access. It involves a displacement value (a constant offset) that is added to the base address. The base address is typically determined by the default data segment (DS) or can be overridden by specifying another segment register (CS, SS, ES).

Example: MOV BX, [DI + 10H] (Move the word from the memory location pointed to by DI + 10H into the BX register).

- In that example, 10H is the displacement.
- DI is an index register, and by default, if an index register is used without a segment override, the DS register is used.

Example of segment override:
MOV BX, ES:[DI + 10H] (In this case, the ES segment register is used).
The effective address is the sum of the displacement and the contents of the index or base register, and then offset by the segment register.



2.4. Register Indirect Addressing Mode

In this mode, the memory address is stored in a register.

Allowed Registers:

- BX, BP, SI, DI

Default Segments:

- BX, SI, DI → DS
- BP → SS

Example:

```
MOV AL, [SI]
MOV AX, [BX]
```

Physical Address Calculation Example:

If:

- DS = 0200H
- SI = 1234H

Then:

```
Physical Address = 02000H + 1234H = 03234H
```

2.5. Based Addressing Mode

The offset address is located in the base registers (BX/BP) and deals with an 8-bit/16-bit displacement number. The effective memory address is calculated by adding the contents of a base register to a displacement value

Formula:

```
EA = Base Register + Displacement
```

Example:

```
MOV AX, [BX+4]
ADD CL, [BP+8]
```

 Displacement may be 8-bit or 16-bit.

2.6. Indexed Addressing Mode

Indexed addressing uses index registers (SI/DI) to access array elements.

Formula:

```
EA = Index Register + Displacement
```

Example:

```
MOV AL, [DI+16]
ADD [SI], 5
```

 Used extensively for one-dimensional arrays.

Q/ What is the main difference between base and index register addressing modes?
Answer
In the 8086 through the 80286, this type of addressing uses one base register (BP/BX), and one index register (DI/SI) to indirectly address memory. The base register often holds the **beginning of array** locations in memory, while the index register holds the relative **position of an element in the array**.

2.7. Based-Indexed Addressing Mode

This mode combines base and index registers.

Formula:

```
EA = Base Register + Index Register
```

Example:

```
MOV AX, [BX+SI]
ADD CX, [BP+DI]
```

 Note:

- BX/BP cannot be used together
- SI and DI cannot be used together

2.8. Based-Indexed with Displacement Addressing Mode

This is the most powerful addressing mode in the 8086.

Formula:

```
EA = Base + Index + Displacement
```

Example:

```
MOV AX, [BX+SI+20]
ADD CX, [BP+DI+16]
```

 Used for two-dimensional arrays.

3. Effective Address (EA) Calculation

The **Effective Address (EA)** is the offset used to calculate the physical address.

General Formula:

```
Physical Address = (Segment × 16) + EA
```

4. Exam-Oriented Summary

- Addressing modes define how operands are accessed
- EA calculation is central to memory access
- Segment selection rules are critical
- Understanding addressing modes is essential for arrays, strings, and procedures

Check your understanding

Easy Level (1–10)

Q1. In computer architecture, the data on which an instruction operates is called:
A) Opcode
B) Register
C) Operand
D) Address
E) Segment

Q2. Which part of an instruction specifies the operation to be performed?
A) Operand
B) Opcode
C) Address field
D) Segment field
E) Effective address

Q3. How many addressing modes are supported by the 8086 microprocessor?
A) 6
B) 7
C) 8
D) 9
E) 10

Q4. In which addressing mode is the operand part of the instruction itself?
A) Register
B) Direct
C) Immediate
D) Indexed
E) Based

Q5. Which of the following instructions uses **immediate addressing mode**?
A) MOV AX, BX
B) MOV AX, [1234H]
C) MOV AL, 20
D) MOV AX, [SI]
E) MOV AX, [BX+SI]

Q6. Which addressing mode does **not require memory access** to fetch the operand?
A) Direct
B) Register indirect
C) Indexed
D) Immediate
E) Based-indexed

Q7. Which register cannot be used as a destination operand in MOV instructions?
A) DS
B) ES
C) SS
D) CS
E) AX

Q8. In register addressing mode, operands must be:
A) Any size
B) In memory
C) Of the same size
D) Immediate values
E) Segment registers only

Q9. Which symbol indicates memory access in assembly language?
A) ( )
B) { }
C) [ ]
D) < >
E) @

Q10. In direct addressing mode, the effective address (EA) is:
A) Base + Index
B) Segment × 16
C) Displacement value
D) Offset + Segment
E) Register contents

Medium Level (11–20)

Q11. In the instruction `MOV AX, [1234H]`, which segment register is used by default?
A) CS
B) SS
C) ES
D) DS
E) FS

Q12. Which addressing mode allows the use of BX, BP, SI, or DI to point to memory?
A) Direct
B) Immediate
C) Register indirect
D) Register
E) Immediate indirect

Q13. When BP is used to access memory, the default segment register is:
A) DS
B) ES
C) CS
D) SS
E) FS

Q14. What is the effective address formula for **based addressing mode**?
A) EA = Index
B) EA = Base
C) EA = Base + Displacement
D) EA = Index + Segment
E) EA = Segment × 16

Q15. Which instruction uses **indexed addressing mode**?
A) MOV AX, BX
B) MOV AX, [1234H]
C) MOV AL, [DI+16]
D) MOV AX, [BX+SI]
E) MOV AL, 25H

Q16. Which registers are considered **index registers** in the 8086?
A) AX and BX
B) CX and DX
C) SI and DI
D) BP and SP
E) CS and DS

Q17. Which addressing mode is mainly used to access **one-dimensional arrays**?
A) Based
B) Immediate
C) Indexed
D) Direct
E) Register

Q18. What is the default segment register when SI or DI is used?
A) SS
B) CS
C) ES
D) DS
E) FS

Q19. Which instruction demonstrates **segment override** usage?
A) MOV AX, [1234H]
B) MOV AL, [SI]
C) MOV BX, ES:[DI+10H]
D) MOV AX, BX
E) MOV CL, 10H

Q20. The displacement value in addressing modes can be:
A) Only 8-bit
B) Only 16-bit
C) 4-bit only
D) 8-bit or 16-bit
E) 32-bit

Hard Level (21–30)

Q21. Which addressing mode combines a base register and an index register?
A) Based
B) Indexed
C) Register indirect

D) Based-indexed
E) Immediate

Q22. Which of the following is **NOT allowed** in 8086 based-indexed addressing?
A) BX + SI
B) BP + DI
C) BX + DI
D) BX + BP
E) BP + SI

Q23. What is the effective address formula for based-indexed with displacement mode?
A) EA = Base
B) EA = Index
C) EA = Base + Index
D) EA = Base + Index + Displacement
E) EA = Segment × 16

Q24. Which addressing mode is considered the **most powerful** in the 8086?
A) Direct
B) Immediate
C) Register
D) Based-indexed
E) Based-indexed with displacement

Q25. Which instruction is suitable for accessing **two-dimensional arrays**?
A) MOV AX, [SI]
B) MOV AX, [BX]
C) MOV AX, [BX+SI]
D) MOV AX, [BX+SI+20]
E) MOV AX, 20

Q26. In the 8086, the physical address is calculated using which formula?
A) EA × 16
B) Segment + Offset

C) (Segment × 16) + EA
D) Register + Offset
E) Index + Base

Q27. Which addressing mode requires **effective address calculation** but no memory access for the operand itself?
A) Immediate
B) Register
C) Direct
D) Indexed
E) Based

Q28. In `MOV AL, [SI]`, if DS = 0200H and SI = 1234H, what is the physical address?
A) 1234H
B) 0200H
C) 03234H
D) 1434H
E) 2234H

Q29. Which addressing mode stores the offset address inside a register?
A) Immediate
B) Direct
C) Register
D) Register indirect
E) Based-indexed

Q30. Understanding addressing modes is essential mainly for:
A) Compiler design only
B) High-level languages
C) Array, string, and memory manipulation
D) Graphics programming
E) Internet applications