# 8086 Microprocessor Laboratory Experiments

## Experiment 3: Addressing Modes

Murtadha Hssayeni, Ph.D.

m.hssayeni@uobabylon.edu.iq


# Outlines

- Logical address: summary
- Addressing modes
  - Register
  - Immediate
  - Direct
  - Register indirect
  - Based relative
  - Indexed relative
  - Based indexed addressing
- Procedure
- Discussion

# Logical Address: Summary

- The **logical address** consists of a segment value and an offset address in the form **segment:offset**
- **To access a program in memory:**
  - The **CS** and **IP** are used to define the logical address
- **To access memory data:**
  - Valid segment registers to define segment.
  - Valid general-purpose register to define offset: **BX, SI, DI, and BP**.
- The 8086 microprocessor provides various **addressing modes**.
  - The addressing mode is a method of specifying an operand:

1. Register
2. Immediate
3. Direct
4. Register indirect
5. Based relative
6. Indexed relative
7. Based indexed addressing

| Addressing Mode | Operand | Default Segment |
|---|---|---|
| Register | reg | none |
| Immediate | data | none |
| Direct | [offset] | DS |
| Register indirect | [BX] | DS |
| | [SI] | DS |
| | [DI] | DS |
| Based relative | [BX]+disp | DS |
| | [BP]+disp | SS |
| Indexed relative | [DI]+disp | DS |
| | [SI]+disp | DS |
| Based indexed relative | [BX][SI]+disp | DS |
| | [BX][DI]+disp | DS |
| | [BP][SI]+ disp | SS |
| | [BP][DI]+ disp | SS |

# Register addressing mode

- The register addressing mode involves the use of registers to hold the data to be manipulated.
- Memory is not accessed when this addressing mode is executed

MOV REG, REG

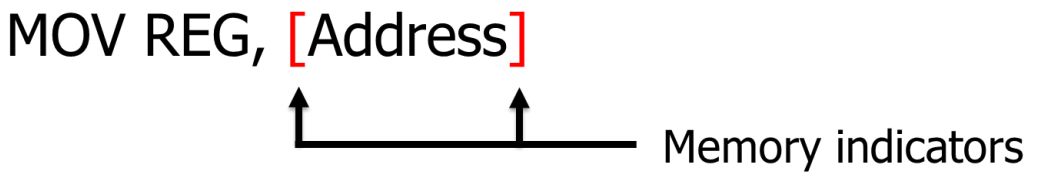
- Example:
  - MOV BX, DX
  - MOV DS, AX

# Immediate Addressing Mode

- The source operand is constant.
- The operand comes immediately after the opcode.
- Immediate addressing mode can be used to load information into any of the registers **except the segment registers and flag registers.**
- Example:
  - Mov AX, 2550h
  - Mov CX, 625h
  - Mov Bl, 40h

# Direct Addressing Mode

- a 16-bit memory address (offset or **displacement**) is directly specified in the instruction as the location of the operand.




| BX | SI | + disp |
|---|---|---|
| BP | DI | |

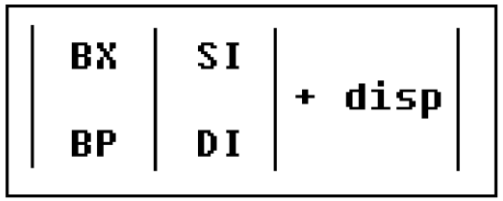
- The physical address is calculated using **DS** and the offset
  - **Physical address = DS *10h + displacement**

- Example:
  - MOV AX, 1500H
  - MOV DS, AX
  - MOV CL, 33H
  - **MOV [3518H], CL** ; moving one byte to memory**, physical address = 1500*10+3518**
  - **MOV AX, [5000H]** ; moving two bytes, logical address DS:5000H

# Register Indirect Addressing Mode

- The registers (**BX, SI or DI**) are directly specified in the instruction as the memory location of the operand.

MOV REG, [BX or SI or DI]




- The physical address is calculated using **DS** and (BX, SI or DI)
  - **Physical address = DS *10h + offset**

- Example:
  - MOV AX, 1500H
  - MOV DS, AX
  - MOV CL, 33H
  - MOV BX, 3518H
  - MOV [BX], CL ; moving one byte to memory**, physical address = 1500*10+3518**
  - MOV SI, 5000H
  - MOV AX, [SI] ; moving two bytes, logical address DS:SI

# Based Relative Addressing Mode

- The registers (**BX, and BP**) and a displacement are used to calculate the effective address of the operand.

MOV REG, [BX or BP] + Disp

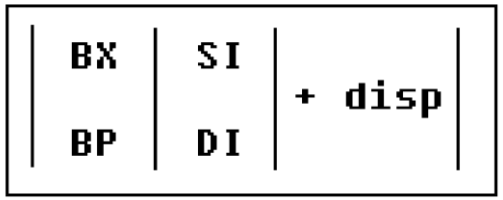
| BX | SI | + disp |
|---|---|---|
| BP | DI | |

- The physical address is calculated
  1. using **DS and BX**
     - **Physical address = DS *10h + BX + Disp**
  2. using **SS and BP**
     - **Physical address = SS *10h + BP + Disp**
- Example:
  - MOV AX, 0B820H
  - MOV DS, AX
  - MOV CL, 'H'
  - MOV BX, 14CH
  - MOV [BX], CL ; moving one byte to memory**, physical address = 0B820*10+014C**
  - MOV [BX+2], 'i' ; moving one byte to memory**, physical address = 0B820*10+014C + 2**
  - MOV [BX+4], '!' ; moving one byte to memory**, physical address = 0B820*10+014C + 4**

# Indexed Relative Addressing Mode

- The registers (**SI, DI**) and a displacement are used to calculate the effective address of the operand.

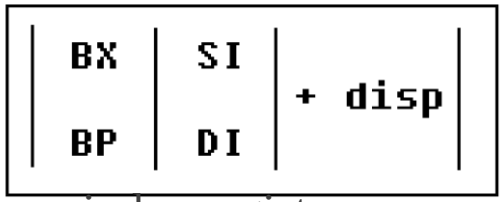
MOV REG, [SI or DI] + Disp




- The physical address is calculated
  1. using **DS and DI or SI**
     - **Physical address = DS *10h + offset + Disp**
- Example:
  - MOV AX, 0B820H
  - MOV DS, AX
  - MOV CL, 'H'
  - MOV SI, 14CH
  - MOV [SI], CL ; moving one byte to memory**, physical address = 0B820*10+014C**
  - MOV [SI +2], 'i' ; moving one byte to memory**, physical address = 0B820*10+014C + 2**
  - MOV [SI +4], '!' ; moving one byte to memory**, physical address = 0B820*10+014C + 4**

# Based Indexed Addressing Mode

- The registers (**BX or BP, and SI or DI**) and a displacement are used to calculate the effective address of the operand.




  - Use one base register and one index register
- The physical address is calculated
  1. using **DS and BX, SI, and DI**
     - **Physical address = DS *10h + BX + Disp**
  2. using **SS and BP, SI, and DI**
     - **Physical address = SS *10h + BP + Disp**
- Example:
  - MOV AH, [BP+SI+29] ; physical address = SS*10+BP+SI+29
  - MOV CH, [BX+SI] + 20 ; physical address = DS*10+BX+SI+20
  - Note: MOV AX, [SI+DI] is illegal

# Procedure

1. Write an assembly program to use DS = 4500, SS = 2000, BX = 2100, SI = 1486, DI = 8500, BP =7814, and AX = 2512. All values are in hex.

2. Find the addressing mode and the exact physical memory location where AX is stored in each of the following:
   - MOV [BX]+20H, AX
   - MOV [BX+SI]+10H, AX
   - MOV [BP]+12H, AX

3. Save the data (A123H) to the physical memory (46300H) if DS = 4500H.

# Discussion

1. Write a program to store (10101010B) to the memory location 47511H. Using based relative addressing mode.

2. Find the addressing modes used in the following program:

*MOV AX, 3000H*
*MOV DS, AX*
*MOV AX, 4000H*
*MOV SS, AX*
*MOV BX, 0600H*
*MOV BP, 0700H*
*MOV DX, 0A6E2H*
*MOV SI, 0040H*
*MOV DI, 00E0H*
*MOV [BX]+SI, AH*
*MOV [BX]+DI, DH*
*MOV [BP]+DI, DL*
*MOV [BX]-20H, DH*
*MOV [DI]+4H, DL*
*MOV [BP]+100H, AL*

3. Save the data (0A6E2H) to the physical memory (4EA00H) if DS = 4500H.