

8086 Addressing Modes

Course Teacher:

Md. Fahmidur Rahman Sakib

Lecturer, Department of Computer Science & Engineering
Metropolitan University

Course ID: CSE 237

Course Title: Microprocessor and Interfacing

Lecture References:

▶ **Book:**

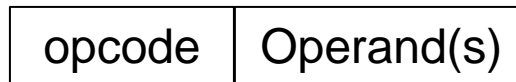
- ▶ *Microprocessors and Interfacing: Programming and Hardware, Chapter # 2, **Author:** Douglas V. Hall*
- ▶ *The 8086/8088 Family: Design, Programming, And Interfacing, Chapter # 2, **Author:** John Uffenbeck.*

Addressing Mode and Categories

- The different ways in which a microprocessor can access data are referred to as its addressing modes.
- Addressing modes of 8086 Microprocessor are categorized as:
 - *Addressing Data*
 - *Addressing Program codes in memory*
 - *Addressing Stack in memory*
 - *Addressing I/O*
 - *Implied addressing*

Things to know...

➤ Instruction format



➤ Instructions can have 1, 2 or no operands

➤ **INC AX** ; 1 operand

➤ **ADD CX, DX** ; 2 operands \longrightarrow $CX = CX + DX$
Destination source

➤ **HLT** ; no operand

➤ Instruction cannot have:

➤ **SUB [DI], [1234h]** ; memory locations as both operands

➤ **MOV 1234, AX** ; immediate data as destination operand

1. Addressing Data

- I. Immediate addressing
- II. Direct addressing
- III. Register [direct] addressing
- IV. Register indirect addressing
- V. Base-plus-index addressing
- VI. Register relative addressing
- VII. Base-relative-plus-index addressing

1. Addressing Data

I. Immediate addressing

- ▶ Data is immediately given in the instruction

MOV BL, 44

II. Direct addressing

- ▶ Data address is directly given in the instruction

MOV BX, [437AH]

1. Addressing Data

III. **Register [direct] addressing**

- ▶ Data is in a register (here BX register contains the data)

MOV AX, BX

IV. **Register [indirect] addressing**

- ▶ Register supplies the address of the required data

MOV CX, [BX]

JMP [DI]

1. Addressing Data

v. **Base-plus-index addressing**

- ▶ Base register is either BX or BP
- ▶ Index register is either DI or SI

MOV DX, [BX+DI]

vi. **Register relative addressing**

- ▶ Register can be a base (BX, BP) or an index register (DI, SI)
- ▶ Mainly suitable to address array data

MOV AX, [BX+I 000]

1. Addressing Data

vii. Base-relative-plus-index addressing

- ▶ Suitable for array addressing

MOV AX, [BX+DI+10]

2. Addressing Program Codes in Memory

- ▶ Used with JMP and CALL instructions
- ▶ 3 distinct forms:
 - ▶ Direct
 - ▶ Indirect
 - ▶ Relative

2. Addressing Program Codes in Memory

- ▶ Address is directly given in the instruction

JMP ^{CS}1000:^{IP}0000

or **JMP** doagain ; doagain is a **label** in code

CALL 1000:0000

or **CALL** doagain ; doagain is a **procedure** in code

- ▶ Often known as *far* jump or *far* call

2. Addressing Program Codes in Memory

- ▶ Address can be obtained from

- ▶ **a)** any GP registers (AX,BX,CX,DX,SP,BP,DI,SI)

JMP AX *IP = AX ; then CS : IP*

- ▶ **b)** any relative registers ([BP],[BX],[DI],[SI])

JMP [BX] *IP = what is inside the physical address of DS : BX ; then CS : IP*

- ▶ **c)** any relative register with displacement

JMP [BX + 100h]

IP = what is inside the physical address of DS : BX + 100h ; then CS : IP

3. Addressing Stack in Memory

- **PUSH** and **POP** instructions are used to move data to and from stack (in particular from stack segment).

PUSH AX

POP CX

- **CALL** also uses the stack to hold the return address for procedure.

CALL SUM ; SUM is a function name

4. Addressing Input and Output Port

- ▶ IN and OUT instructions are used to address I/O ports
- ▶ Could be *direct addressing*

IN AL, 05h ; Here 05h is a input port number

- ▶ or *indirect addressing*

OUT DX, AL ; DX contains the address of I/O port

- ▶ Only DX register can be used to point a I/O port

5. Implied Addressing

- ▶ No explicit address is given with the instruction
- ▶ implied within the instruction itself
- ▶ Examples:

CLC ; clear carry flag

HLT ; halts the program

RET ; return to DOS

Thank You !!

