ASSEMBLY for 8086

Slide courtesy of Md. Ayaz Masud
Lecturer, Dept of EEE, BUET

Types of Programming Language

- **Machine Language**
- **Assembly Language**
- **High-Level Language**

Assembly vs HLL

- Advantages of HLL
  - Easier to write
  - Easier to understand
  - Better control
  - Doesn’t depend on particular machine

Advantages of Assembly

- Closer to Machine code
- Access to specific memory location
- Helps understand how computer actually works

ASSEMBLY language program is machine specific

- ASSEMBLER converts it into machine code
- Not case-sensitive

Syntax

Each line of an ASSEMBLY program
contains a statement

- A statement can either be an instruction or an assembler directive
- Instruction: MOV, SHL, JMP etc
- Assembler Directive: ORG, ASSUME, END etc (Pseudo Op-Code)

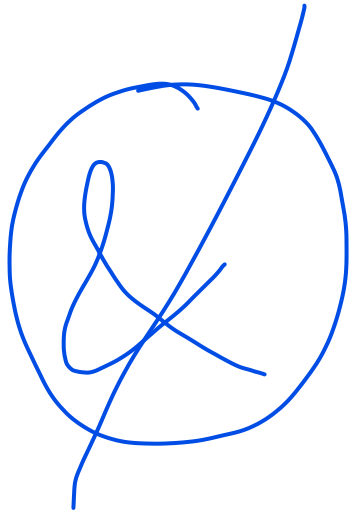
- General syntax for a statement is
  Name: Instruction Operand(s) ; Comment
  Example:
  MYLABEL : MOV AX,BX ; Moving data

Name Field

- Used for instruction labels, procedure names and variable names
- 1 to 31 characters long
- Can contain letters, digits and following special characters

“?”, “.”, “@”, “_”, “$”, “%”

- Period must be used at the beginning

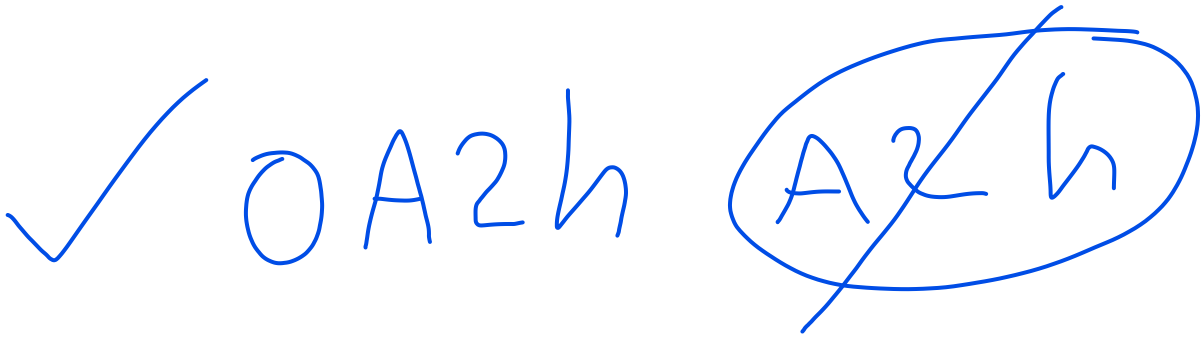


Syntax

Operand : There can be 1,2 or no operand at all. For two operands, generally, they are Destination, Source

- Comment : Anything after (;)
- One of the operands can be a data

MOV AX,10	MOV AX,10D
MOV AX, 1010B	MOV AX,0AH



- Hex number can't start with an alphabet
  - +/- sign is allowed
  - String or character is allowed as data

  MOV AX, 'ABCD'

  MOV AX,0ABCDH

(1) segment registers cannot go together!

(2) CS cannot be modified directly (use far JMP or CALL)

(3) cannot use segment register with an immediate value

MOV

- MOV destination, source

Source Operand	Destination Operand		
	Segment Register	Memory Location	Constant
General Register	not for CS (only) rest is OK		X
Segment Register	X		X
Memory Location	not for CS (only) rest is OK	X	X
Constant	X		X

XCHG destination, source

Source Operand	Destination Operand
	Memory Location
General Register	
Memory Location	X

ADD destination, source

Source Operand	Destination Operand
	Memory Location
General Register	
Memory Location	X
Constant	

INC

- Single operand instruction
- INC destination
- Destination is either a register or a memory location

  INC AX ; AX++ same as, ADD AX, 1

NEG

- Single operand instruction
- NEG destination
- Destination is either a register or a memory location.

Type Agreement of Operands

- The operands of preceding two operand instructions must be of the same type

```
MOV AH,  41H  ; ah=41H
MOV AX,  41H  ; ax =0041H
MOV  AH,  1234H ; illegal
```

but,
(1) wrong parameters: MOV bx,al
(1) operands do not match: 16 bit and 8 bit register

but, mam considers ADD CX, BL to be legal

- Find the illegal operation and comment why?

X i. MOV AL, BX size of Destination(AL->8 bit) < Source(BX->16 bit)

ii. MOV AX,[1120]

X iii. ADD [1200], [2000] cannot add from 2 memory locations

X iv. MOV AX, AE12H hex numbers cannot start with letters

v. SUB BX, 5H

X vi) MOV CS, DS segment registers cannot go together!

X vii) INC CX, 1 INC is single operand

viii) NEG [1001]

ix) NEG CX

X x) MOV 5, AL Destination cannot be a constant

Translation of High – Level Language to Assembly Language:

1. A = 5 – A

```
MOV AX, 5    ; AX = 5
SUB AX, A    ; AX = AX – A = 5 – A
MOV A, AX    ; A = AX = 5 – A
```

2. A = B – 2A

```
MOV AX, B    ; AX = B
SUB AX, A  ; AX = AX - A = B - A
SUB AX, A    ; AX = AX – A = B –A – A = B – 2A
MOV B, AX   ; B = AX = B – 2A
```