

Computer Organization and Assembly Language

Jump Instructions

INC and DEC Instructions

- The INC (increment) and DEC (decrement) instructions, respectively, add 1 and subtract 1 from a single operand.
- The syntax is
 - INC *reg/mem*
 - DEC *reg/mem*

The INC (increment)

- Algorithm:
 - operand = operand + 1
- Example:
- Org 100h
 - *MOV AL, 4*
 - *INC AL ; AL = 5*
- RET

DEC (decrement)

- Algorithm:
 - operand = operand – 1
- Example:
 - MOV AL, 255 ; AL = 0FFh (255 or -1)
 - DEC AL ; AL = 0FEh (254 or -2)
- RET

JMP Instruction (Unconditional Jump)

- The JMP instruction causes an unconditional transfer to a destination identified by a code label.
- The syntax is
 - JMP *destination*
- The JMP instruction provides an easy way to create a loop by jumping to a label at the top of the loop.

JMP Example

- Org 100h
 - Mov al, 5
 - Mov bl, 8
- Jmp sum ; jump to sum
 - Sub bl,al
 - Mov dl,al
 - Int 21h
- Sum:
 - Add al, bl
 - Mov ah,2
 - Int 21h

CMP instruction

- **CMP: Compare.**
- Algorithm:
 - operand1 - operand2
- The result is not stored anywhere
- Flags are set (OF, SF, ZF, AF,PF, CF) according to result.

CMP Example

- Example:
- Org 100h
 - MOV AL, 5
 - MOV BL, 5
 - CMP AL, BL ; AL = 5, ZF = 1 (so equal!)
- RET

NEG Instruction

- NEG: Negative
- Makes operand negative (two's complement).
- Algorithm:
 - Invert all bits of the operand
 - Add 1 to inverted operand
- Operands used:
 - REG
 - memory

NEG-Example

- Example:
- Org 100h
- MOV AL, 5 ; AL = 05h
- NEG AL ; AL = 0FBh (-5)
- NEG AL ; AL = 05h (5)
- RET



CONDITIONAL JUMPS

JE Instruction

- Short Jump if first operand is Equal to second operand (as set by CMP instruction).
- It can be used with both Signed/Unsigned.
- Algorithm:
 - if ZF = 1 then jump

```
include 'emu8086.inc'  
ORG 100h  
MOV AL, 5  
CMP AL, 5  
JE label1  
PRINT 'AL is not equal to  
5.'  
JMP exit  
label1:  
PRINT 'AL is equal to 5.'  
exit:
```



JNE Instruction

- Short Jump if first operand is Not Equal to second operand (as set by CMP instruction).
- It can be used with both Signed/Unsigned.
- Algorithm:
 - if ZF = 0 then jump

```
include 'emu8086.inc'  
ORG 100h  
MOV AL, 2  
CMP AL, 3  
JNE label1  
PRINT 'AL = 3.'  
JMP exit  
label1:  
PRINT 'Al not equal to 3.'  
exit:  
RET
```



JG Instruction

- Short Jump if first operand is Greater than second operand (as set by CMP instruction).
- **Example:**

```
include 'emu8086.inc'  
ORG 100h  
MOV AL, 5  
CMP AL, -5  
JG label1  
PRINT 'AL is not greater -5.'  
JMP exit  
label1:  
PRINT 'AL is greater -5.'  
exit:  
RET
```



JGE Instruction

- Short Jump if first operand is Greater or Equal to second operand (as set by CMP instruction).
- Example:

```
include 'emu8086.inc'  
ORG 100h  
MOV AL, 2  
CMP AL, -5  
JGE label1  
PRINT 'AL < -5'  
JMP exit  
label1:  
PRINT 'AL >= -5'  
exit:  
RET
```



JNG Instruction

- Short Jump if first operand is Not Greater than second operand (as set by CMP instruction)
- Example

```
include 'emu8086.inc'  
ORG 100h  
MOV AL, 2  
CMP AL, -5  
JGE label1  
PRINT 'AL < -5'  
JMP exit  
label1:  
PRINT 'AL >= -5'  
exit:  
RET
```




JNGE

- Short Jump if first operand is Not Greater and Not Equal to second operand
- It is set by CMP instruction.
- Example:

```
include 'emu8086.inc'  
ORG 100h  
MOV AL, 2  
CMP AL, 3  
JNGE label1  
PRINT 'AL >= 3.'  
JMP exit  
label1:  
PRINT 'Al < 3.'  
exit:  
RET
```



JL Instruction

- Short Jump if first operand is Less than second operand (as set by CMP instruction).
- Example;

```
include 'emu8086.inc'  
ORG 100h  
MOV AL, -2  
CMP AL, 5  
JL label1  
PRINT 'AL >= 5.'  
JMP exit  
label1:  
PRINT 'AL < 5.'  
exit:  
RET
```



JLE Instruction

- Short Jump if first operand is Less or Equal to second operand.
- It is set by CMP instruction.

```
include 'emu8086.inc'  
ORG 100h  
MOV AL, -2  
CMP AL, 5  
JLE label1  
PRINT 'AL > 5.'  
JMP exit  
label1:  
PRINT 'AL <= 5.'  
exit:  
RET
```

JNL Instruction

- Short Jump if first operand is Not Less than second operand
- It is set by CMP instruction too
- Example.

```
include 'emu8086.inc'  
ORG 100h  
MOV AL, 2  
CMP AL, -3  
JNL label1  
PRINT 'AL < -3.'  
JMP exit  
label1:  
PRINT 'Al >= -3.'  
exit:  
RET
```

JNLE Instruction

- Short Jump if first operand is Not Less and Not Equal to second operand (as set by CMP instruction)
- Example;

```
include 'emu8086.inc'  
ORG 100h  
MOV AL, 2  
CMP AL, -3  
JNLE label1  
PRINT 'AL <= -3.'  
JMP exit  
label1:  
PRINT 'A1 > -3.'  
exit:  
RET
```

- End