

Instructions and Flags

MOV and XCHG - no flags are changed

ADD and SUB - all flags affected

INC and DEC - all except CF

NEG - all flags affected

CMP - all flags affected (<https://courses.cs.washington.edu/courses/cse374/16wi/lectures/condition-codes.pdf>)

Status Flags - Review

- The Zero flag is set when the result of an operation equals zero.
- The Carry flag is set when an instruction generates a result that is too large (or too small) for the destination operand.
- The Sign flag is set if the destination operand is negative, and it is clear if the destination operand is positive.
- The Overflow flag is set when an instruction generates an invalid signed result.
- The Parity flag is set when an instruction generates an even number of 1 bits in the low byte of the destination operand.
- The Auxiliary Carry flag is set when an operation produces a carry out from bit 3 to bit 4

Test

It always clears the Overflow and Carry flags. It modifies the Sign, Zero, and Parity flags in the same way as the AND instruction.

Example:

mov al, 00100101 OR mov al, 00100100

test al, 00001001b; test bits 0 and 3

```

  0 0 1 0 0 1 0 1
  0 0 0 0 1 0 0 1
  -----
  0 0 0 0 0 0 0 1  -> ZF = 0

```

CMP

The CMP instruction changes the Overflow, Sign, Zero, Carry, Auxiliary Carry, and Parity flags according to the value the destination operand would have had if actual subtraction had taken place.

Unsigned operands:

CMP Results	ZF	CF
Destination < source	0	1
Destination > source	0	0
Destination = source	1	0

Signed operands:

CMP Results	Flags
Destination < source	SF \neq OF
Destination > source	SF = OF
Destination = source	ZF = 1

Example:

1. mov ax, 5

cmp ax, 10 ; ZF = \emptyset CF = 1 $5 - 10 = -5$

2. mov ax, 1000

mov bx, 1000

cmp bx, ax ; ZF = 1 CF = \emptyset $1000 - 1000 = 0$

3. mov si, 105

cm psi, 0 ; ZF = \emptyset CF = \emptyset $105 - 0 = 105$

Jcond Instruction.

Mnemonic	Description	Flags
JZ	Jump if zero	ZF = 1
JNZ	Jump if not zero	ZF = 0
JC	Jump if carry	CF = 1
JNC	Jump if not carry	CF = 0
JO	Jump if overflow	OF = 1
JNO	Jump if not overflow	OF = 0
JS	Jump if signed	SF = 1
JNS	Jump if not signed	SF = 0
JP	Jump if parity (even)	PF = 1
JNP	Jump if not parity (odd)	PF = 0

Mnemonic	Description
JE	Jump if equal (<i>leftOp</i> = <i>rightOp</i>)
JNE	Jump if not equal (<i>leftOp</i> ≠ <i>rightOp</i>)
JCXZ	Jump if CX = 0
JECXZ	Jump if ECX = 0

Unsigned integers

Mnemonic	Description
JA	Jump if above (if $leftOp > rightOp$)
JNBE	Jump if not below or equal (same as JA)
JAЕ	Jump if above or equal (if $leftOp \geq rightOp$)
JNB	Jump if not below (same as JAЕ)
JB	Jump if below (if $leftOp < rightOp$)
JNAЕ	Jump if not above or equal (same as JB)
JBE	Jump if below or equal (if $leftOp \leq rightOp$)
JNA	Jump if not above (same as JBE)

Signed integers

Mnemonic	Description
JG	Jump if greater (if $leftOp > rightOp$)
JNLE	Jump if not less than or equal (same as JG)
JGE	Jump if greater than or equal (if $leftOp \geq rightOp$)
JNL	Jump if not less (same as JGE)
JL	Jump if less (if $leftOp < rightOp$)
JNGE	Jump if not greater than or equal (same as JL)
JLE	Jump if less than or equal (if $leftOp \leq rightOp$)
JNG	Jump if not greater (same as JLE)