How CMP (Compare) Works in Assembly Language

The CMP (Compare) instruction in **8086 assembly language** is used to compare two values. It **subtracts** the second operand from the first operand but **does not store** the result. Instead, it **affects the CPU flags** based on the result of the subtraction.

Syntax of CMP

CMP operand1, operand2

- It performs: {operand1} - {operand2}
- **But it does not store the result in operand1**.
- Instead, it updates the **status flags** in the **FLAGS register**.
- Typically, CMP is followed by a conditional **jump (JMP) instruction**.

Effects on FLAGS Register

The result of CMP affects **three main flags**:

Flag	Meaning	When Set (1)
ZF (Zero Flag)	Indicates equality	If operand1 == operand2
SF (Sign Flag)	Indicates negative result	If operand1 - operand2 < 0
CF (Carry Flag)	Indicates unsigned borrow	If operand1 < operand2

Additionally, **OF (Overflow Flag) and PF (Parity Flag)** may also be affected.

CMP with Conditional Jumps

After CMP, we use **jump instructions** to make decisions:

Instruction	Meaning	Condition
JE / JZ	Jump if Equal / Zero	ZF = 1
JNE / JNZ	Jump if Not Equal / Not Zero	ZF = 0
JG / JNLE	Jump if Greater (signed)	ZF = 0 and SF = OF
JL / JNGE	Jump if Less (signed)	SF ≠ OF
JA / JNBE	Jump if Above (unsigned)	CF = 0 and ZF = 0
JB / JNAE	Jump if Below (unsigned)	CF = 1

Examples

Example 1: Comparing Two Numbers

```
MOV AL, 5      ; Load AL with 5
MOV BL, 3      ; Load BL with 3
CMP AL, BL     ; Compare AL (5) with BL (3)
JG GREATER     ; Jump to GREATER if AL > BL
JL LESS        ; Jump to LESS if AL < BL
JE EQUAL       ; Jump to EQUAL if AL == BL
```

◆ Since **5 > 3**, JG (Jump if Greater) will execute.

Example 2: Checking for Equality

```
MOV AL, 4
MOV BL, 4
CMP AL, BL     ; Compare AL (4) with BL (4)
JE IS_EQUAL    ; Since AL == BL, Zero Flag (ZF) is set, so JE is taken
IS_EQUAL:
   ; Code to execute if numbers are equal
```

◆ Since **AL = BL**, the **Zero Flag (ZF) is set**, and JE (Jump if Equal) is taken.

Example 3: Signed Comparison

```
MOV AL, -5
MOV BL, 2
CMP AL, BL     ; Compare AL (-5) with BL (2)
JL LESS        ; Since -5 < 2, Jump if Less (Signed)
```

◆ Since **-5 < 2**, JL (Jump if Less) is taken.

Conclusion

- CMP **subtracts** the second operand from the first **without storing the result**.
- It **sets flags** that determine the relationship between the two values.
- CMP is always **used with conditional jump instructions** (JE, JG, JL, etc.) to implement decision-making logic.