

- **Bit manipulation and shifting instructions**

- **◆ 1. SHL (Shift Left)**

👉 Syntax:

- Shifts bits to the left, inserting 0s on the right.
- Each shift left is equivalent to multiplying by 2.
-

💡 Example:

MOV AL, 00000011b ; AL = 3

SHL AL, 1 ; Shift left by 1 → AL becomes 00000110b = 6

✅ Explanation:

00000011 → shift left → 00000110

Multiplied by 2: $3 \times 2 = 6$

- **◆ 2. SHR (Shift Right)**

👉 Syntax:

- Shifts bits to the right, inserting 0s on the left.
- Equivalent to dividing by 2 (unsigned).

💡 Example:

MOV AL, 00001000b ; AL = 8

SHR AL, 1 ; AL becomes 00000100b = 4

✅ Explanation:

00001000 → shift right → 00000100

Divided by 2: $8 \div 2 = 4$

◆ 3. ROL (Rotate Left)

👉 Syntax:

- Rotates bits to the left — the leftmost bit rotates into the rightmost position.

💡 Example:

MOV AL, 10000001b

ROL AL, 1 ; AL becomes 00000011b

✅ Explanation:

10000001 → rotate left

Leftmost 1 comes to the right → 00000011

◆ 4. ROR (Rotate Right)

👉 Syntax:

- Rotates bits to the right — the rightmost bit rotates into the leftmost position.

💡 Example:

MOV AL, 00000001b

ROR AL, 1 ; AL becomes 10000000b

✅ Explanation:

00000001 → rotate right

Rightmost 1 moves to leftmost position → 10000000

Example 1: If ZF = 1 → SHL, else → SHR

Write an Assembly program that:

- Loads the value **00001100b** into the **AL** register.
- Compares it with the same value.
- If the two values are equal, shift the bits in AL one position to the left **using SHL**.
- If not equal, shift the bits in AL one position to the right **using SHR**.
- Use **conditional jump instructions and labels**.

```
MOV AL, 00001100b  
CMP AL, 00001100b
```

```
JZ SHIFT  
SHR AL, 1  
JMP END
```

```
SHIFT:  
SHL AL, 1
```

```
END:
```

Example 2: ROR if Not Equal, ROL if Equal

Write an Assembly program that:

- Loads the value **00000011b** into **AL**.
- **Compares** it with **00000011b**.
- If the values are not equal, **rotate the bits in AL** one position to the right **using ROR**.
- If the values are equal, rotate the bits in AL one position to the left **using ROL**.

→ Use *CMP*, *JNZ*, and proper labels to manage control flow.

MOV AL, 00000011b

CMP AL, 00000011b

JNZ RR ; If ZF = 0 → ROR

ROL AL, 1 ; Else → ROL

JMP END

RR:

ROR AL, 1 ; Rotate right by 1

END:

Example 3: Compare AL and BL, then shift or rotate

Write an Assembly program that:

- Loads **2** into both **AL** and **BL**.
- Compares **AL** and **BL**.
- If they are equal, perform **SHL AL, 1**.
- If they are not equal, perform **ROR AL, 1**.

→ Use **CMP**, **JE**, **JMP**, and define separate labels for both operations.

```
MOV AL, 2
MOV BL, 2
CMP AL, BL
```

```
JE SHIFT
ROR AL, 1
JMP END
```

```
SHIFT:
    SHL AL, 1
```

```
END:
```