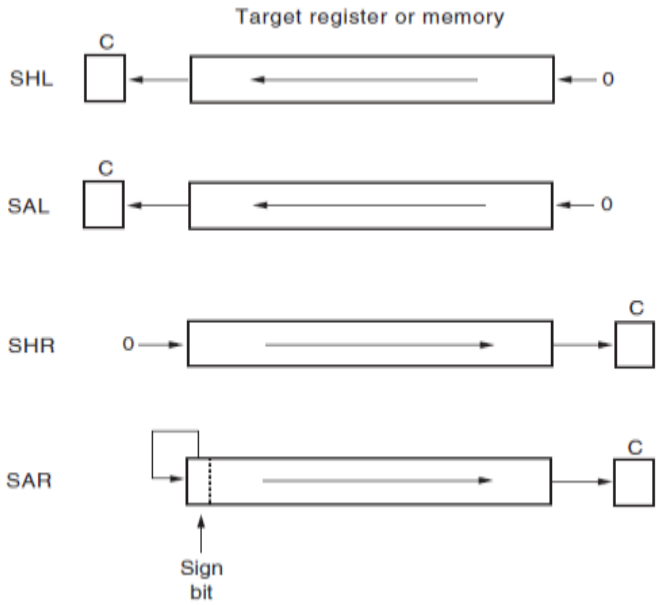
4- Shift and Rotate Instructions

Shift and rotate instructions manipulate binary numbers at the binary bit level, as did the AND, OR, Exclusive-OR, and NOT instructions. The microprocessor contains a complete complement of shift and rotate instructions that are used to shift or rotate any memory data or register.

Shift

Shift instructions position or move numbers to the left or right within a register or memory location. They also perform simple arithmetic such as **multiplication** by powers of 2^{+n} **(left shift)** and **division** by powers 2^{-n} **of (right shift).** The microprocessor's instruction set contains **four** different shift instructions: Two are **logical shifts** and two are **arithmetic shifts**. All four shift operations appear in Figure below.



Notice in above Figure there are **two right shifts** and **two left shifts**.

✓ Logical shifts

move a 0 into the rightmost bit position for a logical left shift and a 0 into the leftmost bit position for a logical right shift.

✓ **Arithmetic shifts**

The arithmetic shift left and logical left shift are identical. The arithmetic right shift and logical right shift are different **because** the arithmetic right shift copies the sign-bit through the number, whereas the logical right shift copies a 0 through the number.

Example shift instructions.

Assembly Language	Operation
SHL AX,1	AX is logically shifted left 1 place
SHR BX,12	BX is logically shifted right 12 places
SAL DATA1,CL	The contents of data segment memory location DATA1 are arithmetically shifted left the number of spaces specified by CL
SAR SI,2	SI is arithmetically shifted right 2 places

The difference between logical and arithmetic shifts

Logical shifts	Arithmetic shifts
Function with unsigned numbers	Function with signed numbers
multiply or divide unsigned data	multiply or divide signed data

✓ A **shift left** always multiplies by **2** for each bit position shifted,

✓ A **shift right** always divides by 2 for each bit position shifted.

✓ Shifting a number **two places**, to the left or right, multiplies or divides by **4**.

There are **two** different forms of shifts that allow any register (except the segment register) or memory location to be shifted.

✓ One mode uses an **immediate shift count**, and

✓ The other uses **register CL** to hold the shift count.

- ✓ **Note** that CL must hold the shift count. When CL is the shift count, it does not change when the shift instruction executes.

The following Example shows how to shift the DX register left 14 places in two different ways. The first method uses an immediate shift count of 14. The second method loads 14 into CL and then uses CL as the shift count. Both instructions shift the contents of the DX register logically to the left 14 binary bit positions or places.

Example:

```
SHL DX,14
```

or

```
MOV CL,14
SHL DX,CL
```

Suppose that the contents of AX must be multiplied by 10, as shown in the following Example . This can be done in two ways:

- ✓ By the MUL instruction or
- ✓ By shifts and additions.

When a number is doubled, and then added to the number times 8, the result is 10 times the number. The number **10 decimal is 1010 in binary (A in hex)**. A logic 1 appears in both the 2's and 8's positions. If 2 times the number is added to 8 times the number, the result is 10 times the number. Using this technique, a program can be written to multiply by any constant. **This technique often executes faster than the multiply instruction found in earlier versions of the Intel microprocessor.**

Example:

Multiply AX by 10_d (1010_b) (A_h) without using any MUL instruction

```
MOV AX,1234h    ; you can move any number for register AX
SHL AX,1        ;AX times 2
```

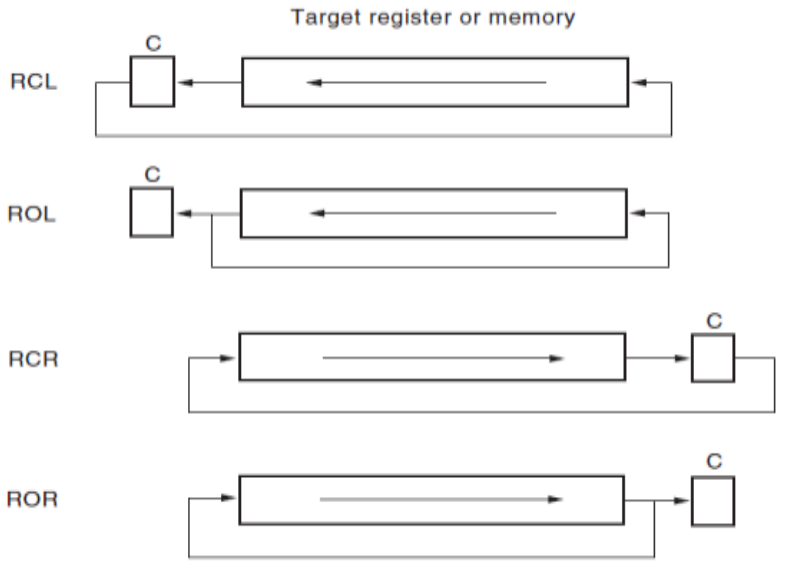
```
MOV BX,AX
SHL AX,2        ;AX times 8
ADD AX,BX       ;AX times 10
```

اذا اردت التأكد من ان هذه الايعازات تعمل نفس عمل الايعاز MUL نفذ التالي:

```
MOV AX,1234h
MOV BX,0Ah
MUL BX
```

Rotate

Rotate instructions position binary data by rotating the information in a register or memory location, either from one end to another or through the carry flag. Numbers rotate through a register or memory location, through the **C flag (carry),** or through a register or memory location only. With either type of rotate instruction, the programmer can select either a left or a right rotate. Addressing modes used with rotate are the same as those used with shifts. A rotate count can be immediate or located in register CL.



Types of rotate instructions

Example rotate instructions

Assembly Language	Operation
ROL SI,14	SI rotates left 14 places
RCL BL,6	BL rotates left through carry 6 places
RCR AH, CL	AH rotates right through carry the number of places specified by CL
ROR WORD PTR[BP],2	The word contents of the stack segment memory location addressed by BP rotate right 2 places

Exercises

1- What is the result of executing the following program :

```
MOV AX,1234h
SHL AX,1
```

2- If CL=2, AX=091Ah What is the result of executing the following instruction:

```
SHR AX,CL
```

3- If CL=1, AX=1234h What is the result of executing the following instruction:

```
SAR AX,CL
```

4- What is the result of executing the following program :

```
MOV AX,1234h
ROL AX,1
```

5- If CL=2, AX=ABCDh , CF=1 What is the result of executing the following instruction:

```
ROR AX,CL
```

6- If AX=12A9h , CF=0 What is the result of executing the following instruction:

```
RCL AX,1
```

7- If CL=3, AX=1203h , CF=1 What is the result of executing the following instruction:

RCR AX,CL