LECTURE # 12

ARITHMETIC SHIFT

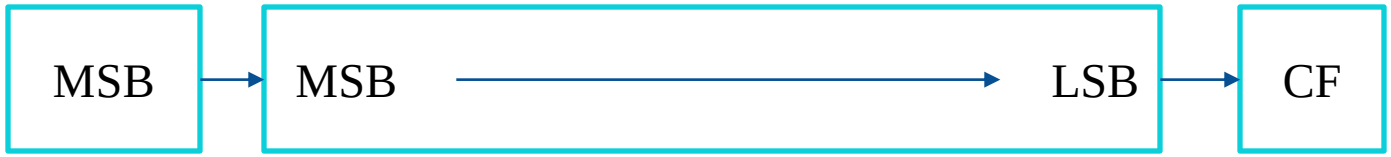
Arithmetic Shift

- As we were discussed earlier that shift instruction has two types: logical and arithmetic.
- Logical shift is used for unsigned numbers.
- The arithmetic shift is used for signed numbers.
- It is basically same as the logical shift, except that the sign bit is copied to the shifted bits.
- SAR (shift arithmetic right) and SAL (shift arithmetic left) are two instructions used for arithmetic shift.

Arithmetic Shift

SAR (shift arithmetic right)

Syntax: SAR destination, count



As the bits of the destination are shifted to the right into CF, the empty bits are filled with the sign bit.

Example:

```
MOV  AL, -10        ; AL = -10 = F6H = 1111 0110
SAR  AL,1           ; AL = 1111 1011 (CF = 0)
```

Arithmetic Shift

SAL (shift arithmetic left)

- SAL and SHL do exactly the same thing.
- It is basically the same instruction with mnemonics.
- As far as signed numbers are concerned, there is no need for SAL.

SIGNED NUMBER COMPARISON

Signed Number Comparison

Syntax:

CMP destination, source

- CMP instruction is same for both signed and unsigned numbers, the jump instruction used to make a decision for the signed numbers is different from that used for the unsigned numbers.
- In unsigned numbers comparisons CF and ZF are checked for condition of larger, equal and smaller.
- In signed numbers comparison, OF, ZF and SF are checked:

destination > source	OF = SF or ZF = 0
destination = source	ZF = 1
destination < source	OF = negation of SF

Signed Number Comparison

- The mnemonics used to detect the conditions above are as follows:

JG	Jump greater	if OF=SF or ZF=0
JGE	Jump greater or equal	if OF=SF
JL	Jump less	if OF=~SF
JLE	Jump less or equal	if OF=~SF or ZF=1
JE	Jump if equal	if ZF=1

Signed Number Comparison

Program: Find the lowest temperature as follows:

+13, -10, +19, +14, -18, -9, +12, -19, +16

```
        .MODEL SMALL
        .STACK 64
        .DATA
TEMP    DB      +13, -10, +19, +14, -18, -9, +12, -19, +16
        ORG     10H
LOWEST DB       ?
        .CODE
MAIN    PROC FAR
        MOV     AX, @DATA
        MOV     DS, AX
        MOV CX, 8
        MOV SI, OFFSET TEMP
        MOV AL, [SI]
BACK:   INC SI
        CMP AL, [SI]
        JLE SEARCH
        MOV AL, [SI]
SEARCH: LOOP BACK
        MOV LOWEST, AL
        MOV AH, 4CH
        INT 21H
MAIN    ENDP
        END MAIN
```