```
  0110 1100+                    6ch + c4h =  130h
  1100 0100
1 0011 0000  =
1     3     0 h                 AF=1
```

4 bits = 1 semioctet = 1 nibble = 1 hexadecimal digit

CF (*Carry Flag*) is the transport flag. It will be set to <u>1</u> if in the LPO there was a transport digit outside the representation domain of the obtained result and set to 0 otherwise. **CF signals overflow for the case of UNSIGNED interpretation.**


```
    1001 0011 +    147+            93h +  - 109 +
    0111 0011      115            73h            115
  1 0000 0110      262          1 06h            06
        CF=1     (unsigned)      (hexa)      (signed)  OF = 0
```


byte + byte → byte   (147 + 115 = 6 !!!!! – ADC) It follows that in the above example unsigned addition does not function correctly as an arithmetic operation in assembly language and that is why the processor reacts to this situation by setting CF=1 ! The signed interpretation addition functions correctly and that is why we have OF=0.

In such situations is advisable to take into account the value from CF using further ADC instructions if needed.

word + word → word   dword + dword → dword
B+B = B (but what if it doesn't fit ??...)   B+B = W

Both CF and OF are reffering to addition and subtraction ONLY.

OF (*Overflow Flag*) flags the **<u>signed overflow</u>**. If the result of the LPO (considered in the signed interpretation) didn't fit the operands reserved space (admissible representation interval), then OF will be set to 1 and will be set to 0 otherwise.

Base 2 – 1…… → the 2 interpretations (SIGNED and UNSIGNED) will be ALWAYS DIFFERENT !!!!

Base 2 – 0…… → the 2 interpretations (SIGNED and UNSIGNED) will be IDENTICAL !!!!

a). Express the number ab in each of the 2, 10 and 16 nummeration bases, in each of the two possible interpretations (signed and unsigned)   1001 0011b, 147, -109, 93h (4 values – NOT 6 !!!!!!)

b). Express the number 73h in each of the 2, 10 and 16 nummeration bases, in each of the two possible interpretations (signed and unsigned)

c). Express the number 129 in each of the 2, 10 and 16 nummeration bases, in each of the two possible interpretations (signed and unsigned)

d). Express the number -37 in each of the 2, 10 and 16 nummeration bases, in each of the two possible interpretations (signed and unsigned)

b,c,d – also 4 values each – NOT 6 !!!!!

Base 2   →   Base 10   ←   Base 16
(representation)   INTERPRETATION   (representation)
                    of base 2 representation
                        ↙   ↘
            Unsigned   Signed
        (absolute value)   (positive and/or negative)