Operatorii efectueaza calculea cu valori constante

**OPERATORI LOGICI PE BITI**

**“~”** -> Neaga fiecare bit din numarul de dupa el

~a=1 daca a = 0 si 0 daca a=1

Exemple:

a db ~10101010b; |55| 5h=0101b a=55h

b db ~10000000b; |7F| 7h=0111b Fh=1111b b=7Fh

**“&”** ->face “si” bit cu bit dintre cele doua constante

Exemple:

a db 11111111b&0b; |00| a=00h

b db 11111111b&11110000b; |F0| Fh=1111b b=F0h

**“|”** ->face “sau” bit cu bit dintre cele doua constante

Exemple:

a db 11111111b|0b; |FF| a=FFh

b db 00001111b|11110000b; |FF| a=FFh

**“^”** ->face “sau exclusiv” bit cu bit dintre cele doua constante

Bitul resultant va avea valoarea 1 daca cei doi biti au valori diferite (unul 1 si celalalt 0)

Exemple:

a db 11110000b^11111111b; |F0| a=F0h

b db 11111111b^11111111b; |00| a=00h

**“!”**

Neaga o expresie intreaga

!a=1 daca a=0, sau 1 altfel

Exemple:

a db !10101100b; |00| a=00h

b db !0b; |01| a=01h

**OPERATORI DE DEPLASARE PE BITI**

**“<<”** -> Deplaseaza la stanga constant data ca parametru

->Valoriile ramase se completeaza cu 0

Exemple:

a db 11111111b<<4; |F0| a=11110000b

a db 10000000b<<1; |00| a=00000000b

Consideram urmatoarea secventa:

a db 0FFh

b db 0b<<8

Rezultatul final din memory layout va fi |FF|00|, deci nu se deplaseaza in stanga memoriei

**“>>”** -> Deplaseaza la dreapta constant data ca parametru

->Valoriile ramase se completeaza cu 0

Exemple:

a db 11111111b>>4; |0F| a=00001111b

a db 00001111b>>1; |07| a=00000111b

Consideram urmatoarea secventa:

a db 0b>>8

b db 0FFh

Rezultatul final in memory layout va fi |00|FF|, deci nu se deplaseaza in dreapta memoriei