Operatii si operatori pe biti

In computer programming, a bitwise operation operates on a bit string, a bit array or a binary numeral at the level of its individual bits. It is a fast and simple action, basic to the higher-level arithmetic operations and directly supported by the processor.

Atentie la diferenta dintre operatori si instructiuni !!

```
Mov ah, 01110111b << 3 ;     AH :=10111000b
```

Vs.

```
Mov ah, 01110111b
Shl ah, 3
```

& - operatorul SI bit cu bit	x AND 0 = 0	; x AND x = x
AND – instructiune	x AND 1 = x	; x AND ~x = 0

Operatie utila pt fortarea valorii anumitor biti la 0 !!!!

| - operatorul SAU bit cu bit	x OR 0 = x	; x OR x = x
OR – instructiune	x OR 1 = 1	; x OR ~x = 1

Operatie utila pt fortarea valorii anumitor biti la 1 !!!!

^ - op. SAU EXCLUSIV bit cu bit;	x XOR 0 = x	x XOR x = 0
XOR – instructiune	x XOR 1 = ~x	x XOR ~x = 1

Operatie utila pt complementarea valorii anumitor biti !!!

XOR ax, ax ; AX=0 !!!

<u>Utilizarea operatorilor ! si ~</u>

! Negare logică: !X = 0 când X ≠ 0, altfel 1
~ Complement faţă de 1: mov al, ~0 => mov AL, 0ffh

a d?....
b d?...

Mov eax, ![a] ; expression syntax error pt ca… [a] NU este o constanta det la mom asamblarii

Mov eax, [!a] ; ! can only be applied to SCALAR values !!!!!
A = POINTER !!!!!!!

Mov eax, !a ; ! can only be applied to SCALAR values !!!!!
A = POINTER !!!!!!!

Mov eax, !(a+7) ; ! can only be applied to SCALAR values !!!!!
A = POINTER !!!!!!!

Mov eax, !(b-a) ; OK !!!! a,b – pointers, dar b-a = SCALAR !

Mov eax, ![a+7] – expression syntax error !

Mov eax, !7 ; EAX = 0 ;

Mov AH, ~7 ; 7 = 00000111b deci ~7 = 11111000b = f8h
Mov eax, ~7 ; EAX = -8 (FFFF FFF8h)

Mov eax, !ebx ; syntax error !

Aa equ 2
Mov ah, !aa ; AH = 0 !!!! – MERGE !!!!!!

Mov AH, 17^(~17) ; AH = 11111111b = 0ffh

Mov ax, value ^ ~value ax=0ffffh