

Operații și operatori pe biți

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.

Atenție la diferența dintre operatori și instrucțiuni !!

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

Vs.

Mov ah, 01110111b

Shl ah, 3

& - operatorul SI bit cu bit

$x \text{ AND } 0 = 0$

$; x \text{ AND } x = x$

AND – instrucțiune

$x \text{ AND } 1 = x$

$; x \text{ AND } \sim x = 0$

Operație utilă pt forțarea valorii anumitor biți la 0 !!!!

| - operatorul SAU bit cu bit

$x \text{ OR } 0 = x$

$; x \text{ OR } x = x$

OR – instrucțiune

$x \text{ OR } 1 = 1$

$; x \text{ OR } \sim x = 1$

Operație utilă pt forțarea valorii anumitor biți la 1 !!!!

^ - op. SAU EXCLUSIV bit cu bit;

$x \text{ XOR } 0 = x$

$x \text{ XOR } x = 0$

XOR – instrucțiune

$x \text{ XOR } 1 = \sim x$

$x \text{ XOR } \sim x = 1$

Operație utilă pt complementarea valorii anumitor biți !!!

XOR ax, ax ; AX=0 !!!

Utilizarea operatorilor ! si ~

! Negare logică: $!X = 0$ când $X \neq 0$, altfel 1

~ Complement față de 1: `mov al, ~0 => mov AL, 0ffh`

a d?....

b d?...

`Mov eax, ![a]` ; expression syntax error pt ca... [a] NU este o constantă det la mom asamblării

`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(+7) = 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