

NSC - 18 feb. 2019

I a)  $\rightarrow 0/x \text{ dw } -256, 256h$

$$\begin{aligned} -256 &= C2(256) = C2(100h) = C2(0001\ 0000\ 0000) = \\ &= 1111\ 0000\ 0000 = F00h \end{aligned}$$

în memorie:  $00\ 1FF\ 156\ 102$

$\rightarrow 4/y \text{ dw } 2561-256, 256h \& 256$

$$\begin{aligned} 2561-256 &= 0001\ 0000\ 0000 \mid 1111\ 0000\ 0000 = 1111\ 1111\ 0000\ 0000 \\ &= FF00h \end{aligned}$$

$$\begin{aligned} 256h \& 256 &= 0000\ 0010\ 0101\ 0110 \& 0000\ 0001\ 0000\ 0000 = \\ &= 0000\ 0000\ 0000\ 0000 = 0h \end{aligned}$$

în memorie:  $00\ 1FF\ 100\ 100$

$\rightarrow 8/z \text{ db } \$-z, y-x$

d.b. 'y' - 'x', 'y-x'

$$\$-z = 8-8 = 0 \quad (\text{scădere de pointeri})$$

$$y-x = 4-0 = 4 \quad (\text{scădere de pointeri})$$

$$'y'-'x' = 1 \quad (\text{scădere între codurile car. ASCII})$$

în memorie:  $00\ 104\ 101\ 1'y'-'x'$

$\rightarrow 14/a \text{ db } 512 \gg 2, -512 \ll 2$

$$512 = 200h = 0010\ 0000\ 0000$$

$$512 \gg 2 = 0000\ 1000\ 0000 = 080h$$

$$-512 = C2(512) = 1110\ 0000\ 0000$$

$$-512 \ll 2 = 1000\ 0000\ 0000 = 800h$$

BRUNNEN

în memorie:  $80\ 00$



→ 16 | b dw  $\$ - a$ ,  $!(\$ - a)$

$$\$ - a = \$ - 14 = -6 \text{ (scalar)}$$

$$= C2(6) = C2(0000 \ 0110) = 1111 \ 1010 = FAh \text{ (FFFAh word)}$$

$$!(\$ - a) = !(FAh) = 0h$$

in memorie: FA | FF | 00 | 00

→ 20 | c dd  $(\$ - b) + (d - \$)$ ,  $\$ - 2 * y + 3$

$\$ - 2 * y + 3$  se va ignora, deoarece inmultirea unui pointer cu un scalar da un rezultat dincolo de limita maximă a memoriei ce poate fi accesată.

$$\underbrace{(\$ - b)}_{\text{scalar}} + \underbrace{(d - \$)}_{\text{scalar}} = (20 - 16) + (24 - 20) = 4 + 4 = 8 = 08h$$

in memorie: 08 | 00 | 00 | 00

→ 24 | d db  $-128$ ,  $128^{\wedge} (\sim 128)$

$$-128 = C2(128) = C2(0000 \ 1000 \ 0000) = 1111 \ 1000 \ 0000 = F80h$$

$$128^{\wedge} (\sim 128) = 1000 \ 0000^{\wedge} 0111 \ 1111 = 1111 \ 1111 = FFh$$

in memorie: 80 | FF

→ 26 | e times 2 resw 6

Se vor rezerva 6 worduri (resw 6) de două ori (times 2),  
deci 12 worduri, 24 de octeți de 0.

in memorie: 00 | 00 | 00 | ... | 00 | 00 | 00  
de 24 de ori

→ times 2 dd 1234h, 5678h

In memorie se vor stoca 1234h și 5678h de două ori.

34 | 12 | 00 | 00 | 78 | 56 | 1001<sup>1001</sup> | 34 | 12 | 00 | 00 | 78 | 56 | 00 | 00



b) `mov BH, 7Fh` ;  $BH = 0111\ 1111\ b$   
`cmp BH, AL` ; scăderea fictivă  $BH - AL$   
 ;  $CF = 1$  dacă  $AL > BH$   
 ;  $CF = 0$  dacă  $AL < BH$   
`rcr AH, 1` ;  $CF$  se pune pe bitul 0 al lui  $AH$ , adică  
 bitul de semn  
`sar AH, 7` ; shift aritmetic right, completează cu bitul  
 de semn  
 ;  $AH$  va avea bitul de semn al lui  $AL$

Echivalent cu `cbr` (sau `movsx AX, AL`)

III a) • `mov eax, 200`

$EAX$  va avea valoarea  $200 = 00\ 00\ 00\ c8h =$   
 $= 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 1100\ 1000\ b$ , în ambele interpretări,  
 deci  $OF = CF = 0$

• `mov ebx, 254h`

$EBX = 254h = 16^3 \cdot 4 + 16^2 \cdot 5 + 16^1 \cdot 2 = 4 + 80 + 32 = 596$  in ambele  
interpretări

$= 0000\ 0000\ 0000\ 0000\ 0000\ 0010\ 0101\ 0100\ b$

~~`OF = CF = 0`~~

• `idiv bl`

Se va efectua împărțirea cu semn dintre  $AX$  și  $BL$ ,  
 iar câtul și restul se vor reține în  $AL$ , respectiv  $AH$ .

$AX = 00c8h = 200$  (în ambele interpretări)

$$BL = 54h = 1 \cdot 4 + 16 \cdot 5 = 84 \quad (\text{în ambele interpretări})$$

$$84 \in [0, 255] \quad , \quad 84 \in [-128, 127]$$

$$AX : BL = \frac{209}{16} : 84 = 2 \text{ rest } 32$$

$$\text{Deci } AL = 2 = 02h \text{ și } AH = 32 = 20h \quad (\text{în ambele interpretări})$$

$$= 00000010b$$

Flagurile nu sunt definite la împărțire, deci nu se setează.

b). • mov ax, ~ (16h 132) ; AX = FFC9h

$$\sim (16h 132) = \sim (0000 0000 0001 0110 10000 0000 0010 0000) =$$

$$= \sim (0000 0000 0011 0110) = 1111 1111 1100 1001 = FFC9h$$

↙ -55 (în interpretarea cu semn)

↘ 65536 - 55 = 65481 (în interpretarea fără semn)

• mov dx, -1 ; DX = FFFFh = 1111 1111 1111 1111 b

DX = ↙ în interpretarea cu semn : -1

↘ în interpretarea fără semn : 65536 - 1 = 65535

• add ah, dh

Se va efectua adunarea dintre AH și DH, iar rezultatul în AH

$$AH = FFh = 1111 1111b \quad \leftarrow -1 \text{ (cu semn)}$$

255 (fără semn)

$$DH = FFh = 1111 1111b$$

$$AH + DH = \begin{array}{r} 1111 1111 \\ + 1111 1111 \\ \hline \end{array}$$

$$\begin{array}{r} 1111 1111 \\ 1111 1111 \\ \hline 1111 1110 \\ CF \quad F \quad E \end{array} = FEh = 254 \quad (\text{în interpretarea fără semn})$$

Se observă că s-a produs depășire în cazurile interpretării fără semn, întrucât rezultatul corect este 310  $\notin [0, 255]$ , deci CF = 1.

În cazurile interpretării cu semn, rezultatul este -2  $\in [-128, 127]$

$$OF = 0$$

2) • `shl bh, 8` ; se shiftază la stânga cu 8 biți ( $BH = BL$ ,  $BL = 0$ ).

`add bx, ax` ;

`mov eax, ebx` ; echivalent cu `lea eax, [ebx]`