

little – endian

bytes - bytes

In data segment	In memorie - in baza 16
<code>a db 10 ; 0ah</code>	0A - byte a+0 – adresa
<code>b dw 10 ; 000Ah</code>	0A 00 conf little-endian B+0 b+1
<code>c dd 10 ; 0000000Ah</code>	0a 00 00 00 C+0 c+1 c+2 c+3
<code>d dq 10 ; 000000000000000Ah (16 cifre hexa)</code>	0a 00 00 00 00 00 00 00 D+0 +1 +2 +3 +4 +5 +6 +7

a si b in data segm

`a dq 1122334455667788h` (16 cifre hexa)

`b dq 1000000200000000h`

a+b=>adunare in hexa

a in mem: 88 77 66 55 44 33 22 11

a+0 1 2 3 4 5 6 7

`mov edx, dword[a+4]` ; trasferam din mem de la adresa a+4 1 doubleword (adica 4 bytes) in edx

`mov eax, dword[a+0]` ; trasferam din mem de la adresa a+0 1 doubleword (adica 4 bytes) in eax

edx      eax  
11223344 55667788+

ecx      ebx  
10000000 20000000

---

212233447 5667788 h

b in mem: 00 00 00 20 00 00 00 10  
            b+0 1 2 3 4 5 6 7

mov ecx, dword[b+4]

mov ebx, dword[b+0]

add eax, ebx

adc edx, ecx ; adc- add with carry edx=edx+ecx+CF

scadere cu carry SBB

sub eax, ebx

sbb edx, ecx ; edx=edx-ecx-CF

+ posibil transport (se salveaza in CARRY Flag)

edx      eax  
11223344 55667788+

ecx      ebx  
10000000 20000000

---

212233447 5667788 h

add eax, ebx /sub eax, ebx

adc edx, ecx / sbb edx, ecx

x dq 11223344 55667788h

X in mem

88 77 66 55 44 33 22 11

x+ 0 1 2 3 4 5 6 7

declararea variab in data segment: D –define urmata de tipul de data (B-byte (8 biti), w-Word (16 biti), d-Doubleword (32 biti), Q-quadword( 64))

nr de cifre hexazecimale pentru fiec tip de data

1 cifra hexa = 4 cifre binare sau 4 biti

1 byte = 2 cif hexa

1 word = 2 bytes = 4 cifre hexa

1 doubleword = 4 bytes = 8 cifre hexa

1 qaudword = 8 bytes = 16 cifre hexa

--	--

b dd 12345678h

cx : bx

b in mem:

78 56 34 12

0 1 2 3