Lab 3

3 fara semn pe un byte = 00000011 = (03)h -> word (0003)h

3 cu semn:

+3 = 00000011 = (03)h -> word (0003)h

B dw 3

B in memory

Little endian- octetii in ordine inversa

03 00

-3 = codul complementar (complement fata de 2) = 11111101 = (FD)h -> word (FFFD)hs

11111100+

00000001

11111101

- 3 in memorie:

FD FF a dw -3

-1 = FF pe un byte -> -1 pe un doubleword (FFFFFFFF)h

C dd 12345678h -> dx:Ax

Dx=1234h si AX= 5678h

C in memory

78 56 34 12

C+0 c+1 c+2 c+3

C / word

c-> dx:ax

c+0 si c+1 -> ax

c+ 2 si c+3 ->dx

mov ax, word [c+0]

mov dx, word[c+2]

d dq 1122334455667788h

d -> edx:eax edx = 11223344h eax=55667788h

d in memory

88 77 66 55 44 33 22 11

D 0 1 2 3 4 5 6 7

Mov eax, dword [d+0]

Mov edx, dword [d+4]

|  |  |
| --- | --- |
| Imul  -4 \* c, c db -2  Mov al, -4  Imul byte [c] ; ax = -4\*-2  B\*a, a byte -5  B word 6  aux resd 1  ; aux dd 0  Mov al, [a]  Cbw ; ax=[a]  Imul word [b] ; dx:Ax = a\*b  ; dx:ax - > aux  Met 1 stiva  Push dx  Push ax  Pop eax,  Mov [aux], eax  ;Met 2  Mov word[aux+0], ax  Mov word[aux+2], dx | Idiv  a/b a byte b doubleword  a db 23  b dd -3  mov al, [a]  cbw  cwde ; eax=[a]  cdq edx:eax = [a]  idiv dword[b] ; eax cat a/b, edx – restul  expresie  c/(d-(a\*2))  a word 2  d byte 6  c quad -8  doi equ 2 (in ds)  mov ax,[a]  mov bx, doi ; in cod segment  imul bx ; dx:ax = a\*2  push dx  push ax  pop ebx ;ebx=a\*2  mov al, [d]  cbw ; ax=[d]  cwde ; eax = [d]  sub eax, ebx ; eax= (d-(a\*2))  ~~;sbb eax, ebx ; eax=eax=ebx-cf~~  mov ebx, eax ; ebx= (d-(a\*2))  mov eax, dword[c+0]  mov edx, dowrd[c+4]  ; edx:eax= c  Idiv ebx ; edx:eax/ebx ; **eax** –cat si in edx rest  ; adunarea a doua quadwords  A dq  B dq  A+b  Mov eax, dword [a+0]  Mov edx, dword[a+4]  ;edx:eax = a  Mov ebx, dword [a+0]  Mov ecx, dword[a+4]  ;ecx:ebx = b  tr  Edx:eax+  Ecx:ebx  Clc ; clear carry flag  Add eax, ebx ;dc exista tr se seteaza cf, adica cf=1  Adc edx, ecx ; edx=edx+ecx+CF  ; scaderea a doua quad  M –n, m n de tip quad  Mov eax, dword [m+0]  Mov edx, dword[m+4]  ;edx:eax = m  Mov ebx, dword [n+0]  Mov ecx, dword[n+4]  ;ecx:ebx = n  m-n  im  edx:eax-  ecx:ebx  clc ; pune 0 in CF  sub eax, ebx  ; daca exista imprmut se pune 1 in Cf  Sbb edx, ecx ; edx=edx-ecx-CF |