Seminar 3 ASC

Reprezentarea little endian a nr în memorie.

Operatii de salt neconditionat si conditionat.

Operatii pe siruri

1. Reprezentarea little endian a nr în memorie

A db 12h A:

12h

12h

B dw 1234h B:

34h 12h

C dd 12345678h C:

78h 56h 34h 12h

D dq 1234567887654321h D

21h 43h 65h

E dd 34567h E:

Pb1

x=a\*b+c\*d

f.s, a,b,c,d – cuv

34h 12h 78h 06h 89h 12h 01h 10h ? ? ? ?

a b c d x

|  |  |
| --- | --- |
| Data segment | Code segment |
| A dw 1234h  B dw 678h  C dw 1289h  D dw 1001h  x resd 1 | mov ax, [a]  mul word[a+2] ;mul word[b] ;DX:AX=a\*b  mov [x], dx  mov [x+2], dx  mov ax, [c]  mul word[d]  add [x], ax  adc [x+2], dx |

1. Instructiuni de salt neconditionat

-

-

-

JMP eticheta

-

-



-

eticheta

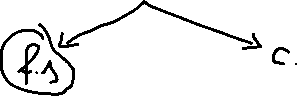
-



-

-

3) Instructiuni de salt conditionat



**J – jump; E – equal; N – not**

(f.s – fara semn) (c.s – cu semn)

A – Above; G – Greater;

B – Below; L- Less

JA et, op1>op2 <=> JNBE et JG et, op1>op2 <=> JNLE et

JAE et, >= <=> JNB et

JNA et, <= <=> JBE et

JNAE et, < <=> JB et

“IF” instructiune de comparare

CMP op1, op2



TEST op1, op2

instructiune de salt conditionat

ex. Se da cuv A. Sa se adune 1 daca A e par si 2 daca e impar

V1

----------

a dw 17

mov AX, [a]

mov DL, 2

div DL

cmp AH, 0

ja impar

;par

mov AX, [a]

inc AX

jmp et

impar:

mov AX, [a]

add AX, 2

et:

V2:

a dw 17

mov AX, [a]

Test AX, 1

ja impar

;par

inc AX

jmp et

impar:

add AX, 2

et:

**4) Siruri de octeti**

offset = baza + index\*scala + constanta

EAX 1 8bit

EBX 2 16bit

ECX 4 32bit

EDX 8



ESI

EDI

ESP -----



EBP

mov AX, [EAX]

mov AX, [a+EAX+EBX]

mov AX, [a+4+EBX\*2]

Pb 2. Se da un sir de cuvinte. Sa se adune in cuv A numerele formate de bitii 6-9 din fiecare cuv al sirului.

12h

1234h = 0001 0010 0011 0100b

A= 1000b + .....

|  |  |
| --- | --- |
| data segment | code segment |
| s dw 5, 6, 7, 8  len equ ($-s)/2  a dw 0 | mov ecx, len  mov esi , 0  Repeta:  mov ax, [s+esi]  and ax, 0000001111000000b  shr ax, 6  add [a], ax  add esi, 2  dec ecx  cmp ecx, 0 |

Pb 3 Se da un sir de cuvinte S. Sa se formeze sirul de octeti D, care contin octetii high din fiecare cuvant al lui S rotiti spre stanga cu valoarea octetilor low din cuvantul corespunzator.

1234h

Rol 12h, 34h

|  |  |
| --- | --- |
| data segment | code segment |
| s dw 5,6,7,8  len equ ($-s)/2  d times len db 0 | move ecx, len  jecxz final  mov esi, 0  repeta:  mov bx, [s+esi\*2]  push ecx  mov, cl, bl  rol bh,cl  pop ecx  mov [d+esi],bh  inc esi  loop repeta  final: |

Pb 4 Se da un sir de dublucuvinte S. Sa se formeze sirul cuvintelor din S pentru care octetul high este mai mare decat octetul low

|  |  |
| --- | --- |
| data segment | code segment |
| s dd 6, 7, 8  len equ ($-s)/2  d times len dw 0 | mov ecx, len  mov esi, s  mov edi, d  jecxz final  mov ax, [esi]  cmp ah, al  jle next  mov [edi], ax  add edi, 2  next:  add esi, 2  loop repeta  final: |