

Arhitectura Sistemelor de Calcul

Lect. Dr. Șotropa Diana
diana.sotropa@ubbcluj.ro



Facultatea de Matematică și Informatică
Universitatea Babeș-Bolyai





Examen scris

Examen scris: Exercițiul 1

```
mov ah, 128 | 2  
mov bh, 90h >> 3  
sub ah, bh
```

Detaliați efectul fiecărei linii sursă exprimând fiecare dintre valorile implicate în fiecare din bazele 10 și 16, cu semn și fără semn. Precizați dacă apare conceptul de depășire, justificând cauza și efectul său.

Examen scris: Exercițiul 1 - rezolvare

`mov ah, 128 | 2`

`; 100000000b | 00000010b = 10000010b`

`; AH = 82h = -126 (c.s.) = 130 (f.s.)`

`mov bh, 90h >> 3`

`; 10010000b >> 3 = 00010010b`

`; BH = 12h = 18 (c.s.) = 18 (f.s.)`

`sub ah, bh`

`; (c.s.) AH-BH=70h=>10...0b-10...0b=00...0b=>OF=1`

`; (f.s.) AH-BH=70h=>CF=0`

Examen scris: Exercițiul 2

```
mov ebx, esp
lea eax, [et1]
push eax
mov ebx, [ss:ebx-4]
jmp ebx
```

```
mov ebx, et1
push ebx
.....
```

Scrieți în locul liniei punctate din a doua secvență **o singură instrucțiune ASM**, astfel încât efectul celor două secvențe de instrucțiuni să fie absolut identic. Explicați și justificați efectul fiecărei linii sursă din ambele secvențe date.

Examen scris: Exercițiul 2 - rezolvare

```
mov ebx, esp
; ebx = esp (pointeaza spre varful stivei)
lea eax, [et1]
; eax = offset et1
push eax
; se pune pe stiva offset et1 si ESP = ESP - 4 (new ESP = old ESP - 4)
mov ebx, [ss:ebx-4]
; ebx = valoarea din varful stivei (EBX fiind old ESP), adica offset et1
jmp ebx
; se face salt la et1

mov ebx, et1
; ebx = offset et1
push ebx
; se pune pe stiva offset et1
ret
```

Examen scris: Exercițiul 3

a db -23, not -5, 200, \$-a, \$+a, \$\$-\$, \$-x
 db -128 & 80h, -200, a+1, 1752h, 256, f7h
b dw b-2, bx
c dd c-b

Prezentați structura din memorie a segmentului de date de mai sus, justificând în context modul de obținere a fiecărei valori ce va fi memorată. Dacă sunt elemente pe care le considerați incorecte sintactic, justificați motivul și ignorați apoi acea valoare în construirea modului de reprezentare în memorie a segmentului de date.

Examen scris: Exercițiul 3 - rezolvare

a db -23, ! -5, 200, \$-a, \$+a, \$\$-\$, \$-x
 db -128 & 80h, -200, a+1, 1752h, 256, f7h
b dw b-2, bx
c dd c-b

a db \$+a; operand is not simple or relocatable

a db \$-x; symbol x undefined

a db -200; warning: byte data exceeds bounds

a db a+1; error: OBJ format can only handle 16- or 32-
byte relocations

Examen scris: Exercitiul 3 - rezolvare

a db -23, ! -5, 200, \$-a, \$+a, \$\$-\$, \$-x
 db -128 & 80h, -200, a+1, 1752h, 256, f7h
b dw b-2, bx
c dd c-b

a db 1752h; warning: byte data exceeds bounds

a db 256; warning: byte data exceeds bounds

a db f7h; symbol f7h undefined

b dw bx; operand is not simple or relocatable

Examen scris: Exercițiul 3 - rezolvare

a db -23, ! -5, 200, \$-a, \$\$-\$
db -128 & 80h, -200, 1752h, 256
b dw b-2
c dd c-b

a: E9h | 00h | C8h | 00h | 00h | 80h | 38h | 52h | 00h |

b: 07h | 00h |

c: 02h | 00h | 00h | 00h |

Examen scris: Exercițiul 4

```
cmp ax, ! -1
js et1
xor dx, dx
jmp go2
et1: mov dh, 3 | 4
mov dl, 2 << 2
and dh, dl
mov dl, -1
xor dh, -1
go2:
```

Scrieți o singură instrucțiune asm, având același efect ca secvența asm dată. Justificați răspunsul explicând efectul exact al fiecărei linii sursă din codul dat și de ce răspunsul dvs. corespunde efectului global al codului sursă de mai sus.

Examen scris: Exercițiul 4 - rezolvare

```
cmp ax, ! -1; cmp ax, 0
js et1; daca ax are bitul de semn 1 salt la et1
xor dx, dx; deci daca SF = 0 atunci dx = 0
jmp go2; salt la eticheta go2
et1: mov dh, 3 | 4; dh = 7 = 00000111b
mov dl, 2 << 2; dl = 8 = 00001000b
and dh, dl; dh = 7 and 8 = 0111b and 1000b = 0
mov dl, -1; dl = -1 = 11111111b
xor dh, -1; dh = dh xor 11111111b = 11111111b,
           ; deci daca SF=1 atunci dx = 0ffffh
go2:
```

CWD

Examen scris: Exercițiul 5

Realizați o clasificare a instrucțiunilor de mai jos într-un număr de categorii după criteriul **“efectului identic asupra registrului BX”** (care trebuie explicat și justificat):

1. `lea ebx, [ebx+6]`
2. `mov ebx, ebx+6`
3. `lea bx, [ebx+6]`
4. `mov ebx, [ebx+6]`
5. `lea bh, byte [ebx+6]`
6. `mov ebx, dword [ebx+6]`
7. `lea ebx, word [ebx+6]`
8. `lea ebx, dword [ebx+6]`
9. `add ebx, 6`

Explicați în detaliu ce face (de sine stătător) fiecare dintre cele 9 instrucțiuni și pe baza explicațiilor date realizați clasificarea cerută.

Examen scris: Exercițiul 5 - rezolvare

1. `lea ebx, [ebx+6]; ebx = old ebx + 6 => bx = old bx + 6`
2. `mov ebx, ebx+6; eroare de sintaxă`
3. `lea bx, [ebx+6]; bx = old bx + 6`
4. `mov ebx, [ebx+6]; ebx = conținutul dublucuv. de memorie de la adresa ds:[EBX+6] => bx = conținutul cuv. de memorie de la adresa ds:[EBX+6]`
5. `lea bh, byte [ebx+6]; eroare de sintaxă`
6. `mov ebx, dword [ebx+6]; ebx = conținutul dublucuv. de memorie de la adresa ds:[EBX+6] => bx = conținutul cuv. de memorie de la adresa ds:[EBX+6]`
7. `lea ebx, word [ebx+6]; eroare de sintaxă`
8. `lea ebx, dword [ebx+6]; eroare de sintaxă`
9. `add ebx, 6; ebx = old ebx + 6 => bx = old bx + 6`

Examen scris: Exercițiul 6

Se da următoarea secvență de 3 instrucțiuni ASM:

```
add ebx, 3 | 4  
mov eax, esi  
add ebx, eax
```

Scriveți o singură instrucțiune ASM care să aibă același efect ca și secvența dată și explicați/justificați de ce se obține același efect.

Examen scris: Exercițiul 6 - rezolvare

`add ebx, 3 | 4; ebx = ebx + 111b = ebx + 7`

`mov eax, esi; eax = esi`

`add ebx, eax`

`; ebx = ebx + eax = ebx(initial) + esi + 7`

`lea ebx, [ebx + esi + 7]`

Examen scris: Exercițiul 7

- Prezentați și exemplificați (prin scurte secvențe de cod sursă adecvate și sugestive) fiecare dintre instrucțiunile limbajului de asamblare (sau categorii de instrucțiuni dacă sunt similar ca tip) capabile să opereze cu numere negative (câte 1 exemplu pt fiecare instrucțiune sau categorie – explicat și justificat).

Examen scris: Exercitiul 7 - rezolvare

- Instrucțiuni ale I.a. "capabile să lucreze cu nr. negative" = instrucțiuni care țin cont de semn.
- 1). IMUL - sintaxa + efect + 1 exemplu explicat
- 2). IDIV - sintaxa + efect + 1 exemplu explicat
- 3). Conversii - cbw, cwd, cwde, movsx d,s - sintaxa + efect + 1 exemplu explicat
- 4). SAR - sintaxa + efect + 1 exemplu explicat
- 5). instr. salt condiționat ("greater than", "less than") - sintaxa + efect + 1 exemplu explicat

Examen scris: Exercițiul 8

Prezentați și justificați structura și valorile din memorie a următorului segment de date presupunând că offsetul său de început este 0:

segment data

x dw -129, 10+100h+1000b

y dw 1001h>>1001b, 128h & 128

z dw z, \$\$-z

w dd x+y-z, w-y+x

h dw 101b,101-h,11h-11b,h-11

a db \$\$-\$, h-11b

b dd a+b-0bh+2, h-b+0ah-0bh

c dd abcdefh, 'abcdefh'

k dw 1+2b+3h+a, c+0ch

m dd a+0ah, a+ah

Dacă sunt linii sau elemente pe care le considerați incorecte sintactic, justificați motivul și ignorați apoi acea linie sau element în construirea modului de reprezentare în memorie a segmentului de date. Adică, dacă TOATE elementele de pe o linie sunt incorecte sintactic se elimină/se ignoră INTREAGA linie, dacă doar UNELE elemente sunt incorecte sintactic se elimină/se ignoră DOAR ACELE elemente, celelalte elemente corecte sintactic rămânând valide și trebuind a fi reflectate în structura segmentului de date ca efect.

Examen scris: Exercițiul 8 - rezolvare

segment data

x dw -129, 10+100h+1000b

y dw 1001h>>1001b, 128h & 128

z dw z, \$\$-z

w dd x+y-z, w-y+x

h dw 101b, 101-h, 11h-11b, h-11

a db \$\$-\$, h-11b

b dd a+b-0bh+2, h-b+0ah-0bh

c dd abcdefh, 'abcdefh'

k dw 1+2b+3h+a, c+0ch

m dd a+0ah, a+ah

Examen scris: Exercițiul 8 - rezolvare

```
0      x dw -129, 10+100h+1000b;  
-129 = FF7Fh, 10+100h+1000b=274=112h  
4      y dw 1001h >> 1001b, 128h & 128  
; 1001h >> 9 = 0008h  
; 128h & 80h = 0000h  
8      z dw z, $$-z  
; z = 8 = 0008h  
; $$-z = offset început secțiune curentă = 0-8 = -8 = FFF8h  
12     w dd x+y-z, w-y+x  
; = 0+4-8 = -4 = FFFF FFFCh, 12-4+0 = 0000 0008h  
20     h dw 101b, 101-h, 11h-11b, h-11  
; 0005h, 101-h nu este o operație validă cu pointeri !! deci syntax error acest element;  
11h-11b = 17-3=14=000Eh; h-11 = 20-11 = 9 = 0009h  
26     a db $$-$, h-11b  
; $$ - $ = offset început "secțiune" curentă (0) (data segment în cazul nostru) - offset  
current (26), deci obținem 0-26 = -26 = E6h; h-11b este o operație validă cu pointeri  
(pointer – constantă, însă rezultatul = pointer, iar un offset NU încapă pe un byte – deci  
syntax error!)
```

Examen scris: Exercițiul 8 - rezolvare

27 b dd a+b-0bh+2, h-b+0ah-0bh

; syntax error – adunarea de pointeri (a+b) nu este o operație permisă; $20-27+10-11 = -8 = \text{FFFF FFF8h}$

31 c dd abcdefh, 'abcdefh' abcdefh – syntax error(Unknown identifier)

; 'a' 'b' 'c' 'd' 'e' 'f' 'h' 00

k dw 1+2b+3h+a, c+0ch

; 2b – syntax error !!

; $c+0ch = 31+12 = 43 = 002Bh$

m dd a+0ah, a+ah

; $26+10=36 = 0000 0024h$

; a+ah - expresie incorectă sintactic – un registru nu poate să apară în cadrul unei expresii de inițializare, nefiind o valoare (constantă) determinabilă la momentul asamblării !

Examen scris: Exercițiul 8 - rezolvare

x: | 7Fh | FFh | 12h | 01h |

y: | 08h | 00h | 00h | 00h | - little endian

z: | 08h | 00h | F8h | FFh |

w: | FCh | FFh | FFh | FFh | = FFFCh

| 08h | 00h | 00h | 00h | = 0008h

h: | 05h | 00h | 0Eh | 00h | 09h | 00h |

a: | E6h | = -26

b: | F8h | FFh | FFh | FFh | = -8

c: | 'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'h' | 00h | - atentie la completarea cu 00h
la final !!!!

k: | 2Bh | 00h |

m: | 24h | 00h | 00h | 00h |

Examen scris: Exercițiul 9

Răspundeți punctual la fiecare dintre întrebările de mai jos, justificând în context răspunsurile date prin explicațiile necesare care să justifice și să detalieze mecanismele implicate.

- 1). Care este numărul MINIM de biți necesari pentru reprezentarea numerelor **32** și respectiv **-132** ? Justificați și explicați. Dați valorile atât în baza 2 cât și în baza 16.
- 2). Instrucțiunea „**mov dword [esp-4], [esp]**” furnizează din păcate „syntax error”. De ce ? Scrieți o instrucțiune corectă în locul celei date care să fie echivalentă ca efect cu intenția avută în vedere. Explicați și justificați. Presupunând că instrucțiunea dată ar funcționa și ar fi admisă, există vreo diferență ca efect final complet între instrucțiunea de la care plecăm și cea scrisă de voi ?
- 3). Dați 2 exemple de instrucțiuni asm diferite care să aibă un singur operand explicit și nici unul implicit. **Explicați și exemplificați prin câte un exemplu concret de utilizare.**
- 4). Dați 2 exemple de instrucțiuni asm diferite care să aibă 2 operanzi implicați și nici unul explicit. **Explicați și exemplificați prin câte un exemplu concret de utilizare.**
- 5). Dați 2 exemple de instrucțiuni asm diferite care să aibă 2 operanzi expliți și unul implicit (deci în total 3 operanzi). **Explicați și exemplificați prin câte un exemplu concret de utilizare.**
- 6). Care sunt cele 2 valori complementare ale numărului binar **1011001b**, un număr binar reprezentat pe 7 biți ? **Explicați.**
- 7). Scrieți o instrucțiune **asm** care să seteze simultan SF și ZF la valoarea 1. **Explicați.**
- 8). Avem **x resd 1**. Este corectă instrucțiunea **xchg eax,x** ? Dacă da, explicați și exemplificați ce face și care este efectul ei. Dacă nu, justificați care este cauza.

Examen scris: Exercițiul 9 - rezolvare

- 1). Care este numărul MINIM de biți necesari pentru reprezentarea numerelor **32** și respectiv **-132** ? Justificați. Dați valorile atât în baza 2 cât și în baza 16.
- **$32 \notin [0, +31] = [0, 2^5 - 1]$, însă $32 \in [0, +63] = [0, 2^6 - 1]$**
deci 6 biți (acest interval conține $64 = 2^6$ valori, deci numerele din acest interval au nevoie de MINIM 6 biți pt a putea fi reprezentate în baza 2) **$32 = 10\ 0000b = 20h$**
- **$-132 \notin [-128, +127] = [-2^7, +2^7 - 1]$, însă $-132 \in [-256, +255] = [-2^8, +2^8 - 1]$, (acest interval conține $512 = 2^9$ valori, deci numerele din acest interval au nevoie de MINIM 9 biți pt a putea fi reprezentate în baza 2); **$-132 = +380 = 1\ 0111\ 1100b = 17Ch$****

Examen scris: Exercițiul 9 - rezolvare

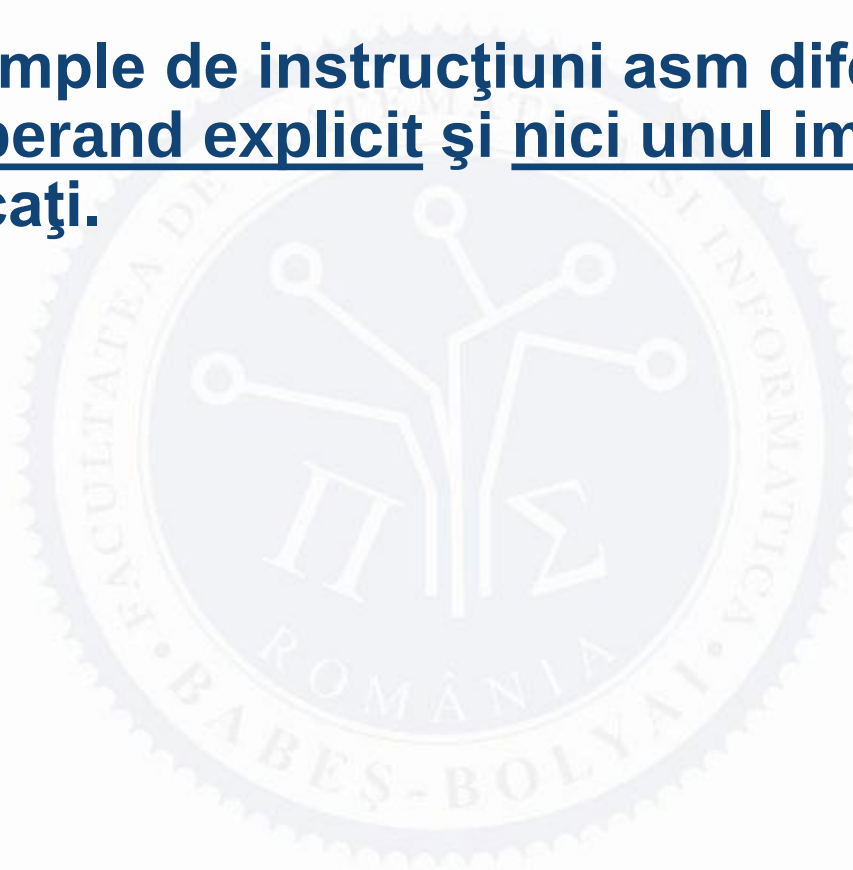
- 2) Instrucțiunea „**mov dword [esp-4], [esp]**” furnizează din păcate „syntax error”. De ce ? Scrieți o instrucțiune **corectă** în locul celei date care să fie echivalentă ca efect cu intenția avută în vedere. Există vreo diferență ca efect final complet între instrucțiunea de la care plecăm și cea scrisă de voi ?

Examen scris: Exercițiul 9 - rezolvare

- 2) Instrucțiunea „**mov dword [esp-4], [esp]**” furnizează din păcate „syntax error”. De ce ? Scrieți o instrucțiune **corectă** în locul celei date care să fie echivalentă ca efect cu intenția avută în vedere. Există vreo diferență ca efect final complet între instrucțiunea de la care plecăm și cea scrisă de voi ?
- (i). Pt că are doi operanzi expliciți din memorie, situație interzisă de sintaxă;
- (ii). **PUSH dword [ESP]**
- (iii). Da, există. Instrucțiunea de la care plecăm nu modifică valoarea din ESP, în schimb Push dword [ESP] o modifică; $ESP = ESP - 4$.

Examen scris: Exercițiul 9 - rezolvare

- 3) Dați 2 exemple de instrucțiuni asm diferite care să aibă un singur operand explicit și nici unul implicit. Explicați și exemplificați.

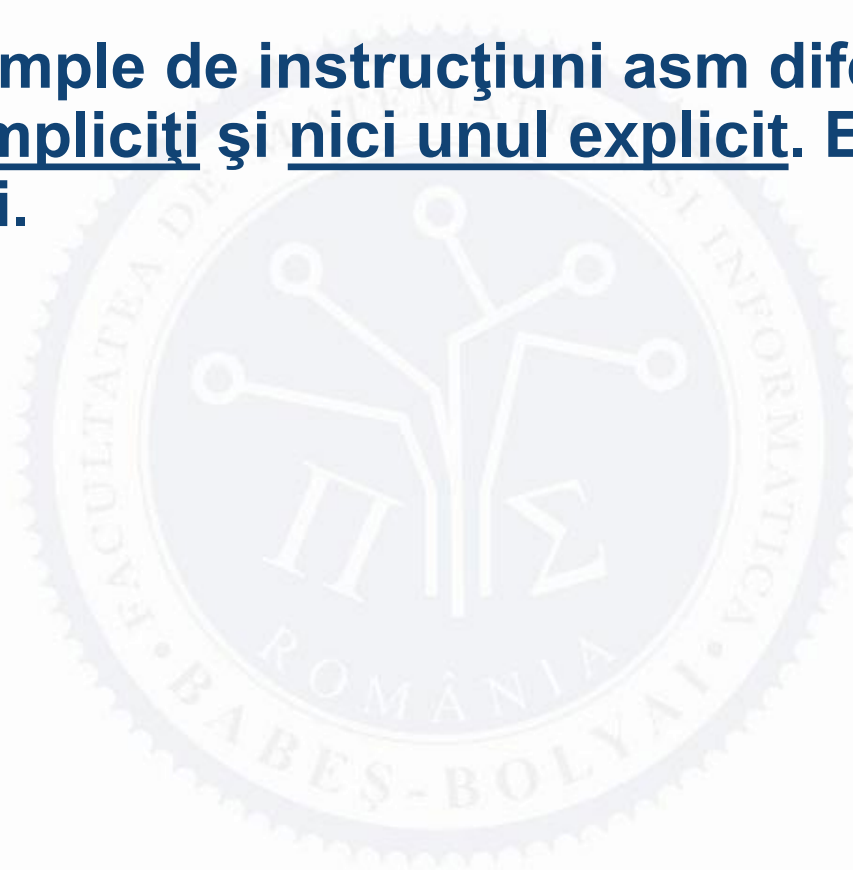


Examen scris: Exercițiul 9 - rezolvare

- 3) Dați 2 exemple de instrucțiuni asm diferite care să aibă un singur operand explicit și nici unul implicit. Explicați și exemplificați.
- JMP destinație CALL destinație RET [n]
- NEG dest (complement față de 2) $d := 0 - d$
- NOT sursa (complement față de 1) $sursa := \sim sursa$

Examen scris: Exercițiul 9 - rezolvare

- 4) Dați 2 exemple de instrucțiuni asm diferite care să aibă 2 operanzi implicați și nici unul explicit. Explicați și exemplificați.

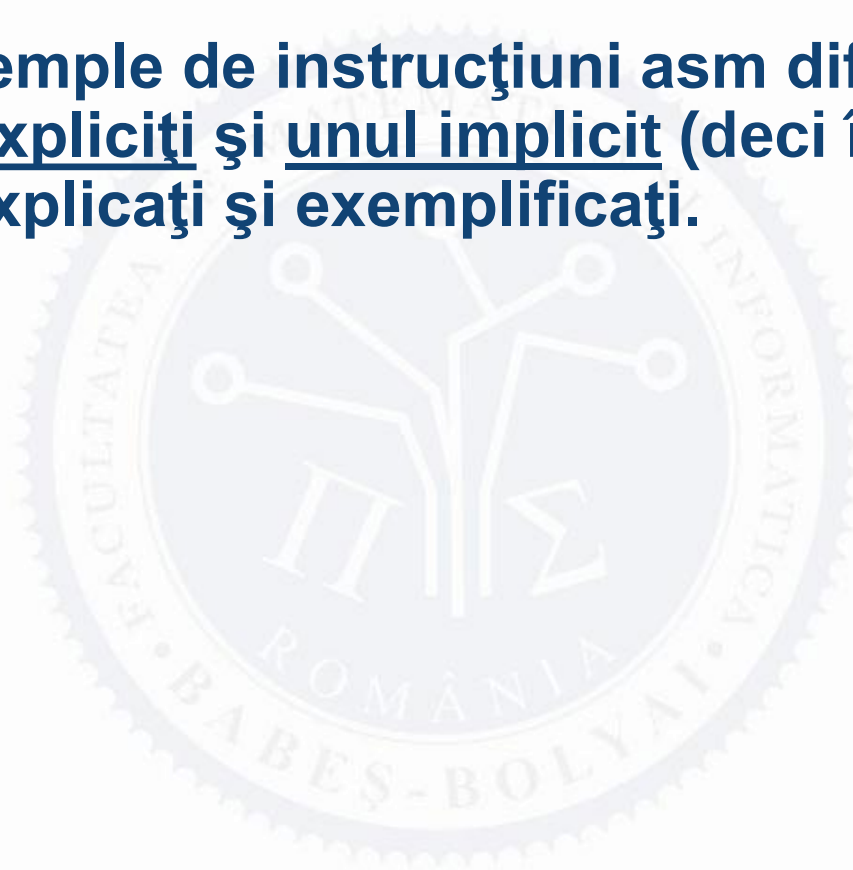


Examen scris: Exercițiul 9 - rezolvare

- 4) Dați 2 exemple de instrucțiuni asm diferite care să aibă 2 operanzi impliciti și nici unul explicit. Explicați și exemplificați.
- `cbw, cwd, cdq` `xlat` `clc, stc, cmc cld, std`

Examen scris: Exercițiul 9 - rezolvare

- 5). Dați 2 exemple de instrucțiuni asm diferite care să aibă 2 operanți expliți și unul implicit (deci în total 3 operanți). Explicați și exemplificați.

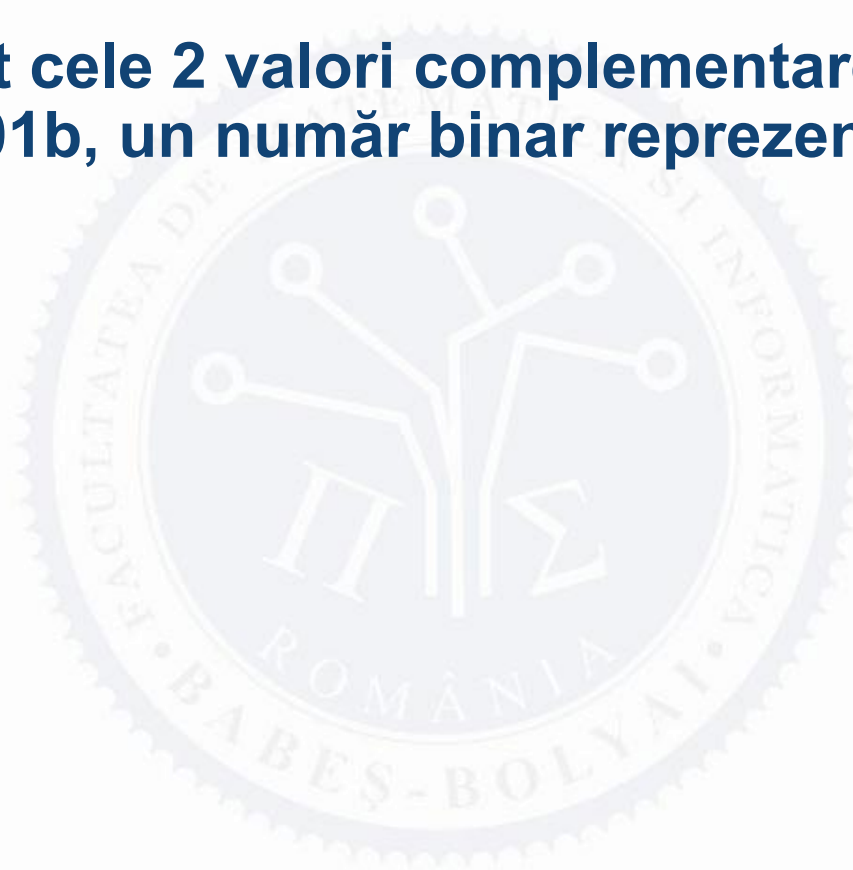


Examen scris: Exercițiul 9 - rezolvare

- 5). Dați 2 exemple de instrucțiuni asm diferite care să aibă 2 operanți expliți și unul implicit (deci în total 3 operanți). Explicați și exemplificați.
- ADC op1, op2 (op1 și op2 - expliți, CF - implicit)
- SBB op1, op2 ((op1 și op2 - expliți, CF - implicit)
- RCL sursa,cu_cat (expliți) și CF – implicit
- RCR sursa,cu_cat (expliți) și CF - implicit

Examen scris: Exercițiul 9 - rezolvare

- 6). Care sunt cele 2 valori complementare ale numărului binar $1011001b$, un număr binar reprezentat pe 7 biți ?
Explicați.



Examen scris: Exercițiul 9 - rezolvare

- **6). Care sunt cele 2 valori complementare ale numărului binar 1011001b, un număr binar reprezentat pe 7 biți ? Explicați.**
- Valoarea absolută a reprezentării pe 7 biți 1011001b este $59_{10} = 89$;
- Pe 7 biți se pot reprezenta $2^7 = 128$ numere: fie numerele fără semn din intervalul $[0..127]$, fie numerele cu semn din intervalul $[-64, +63]$;
- $89 \in [0..127]$, iar complementul numărului 1011001b este 0100111b, adică $27_{10} = 39$, în sensul că $1011001b = -39$ (verificare : suma valorilor absolute a celor două valori complementare este cardinalul mulțimii reprezentabile pe acea dimensiune – $89+39 = 128$, deci se verifică faptul că 1011001b și 0100111b sunt valori complementare)
- plecând de la aceste valori complementare în baza 2, în vorbirea curentă se admit și exprimările «89 și 39 sunt valori complementare pe 7 biți» în sensul că 89 și -39 au aceeași reprezentare în baza 2 (pe 7 biți) sau că «89 și -39 sunt valori complementare pe 7 biți »

Examen scris: Exercițiul 9 - rezolvare

- 7). Scrieți o instrucțiune care să seteze simultan SF și ZF la valoarea 1.

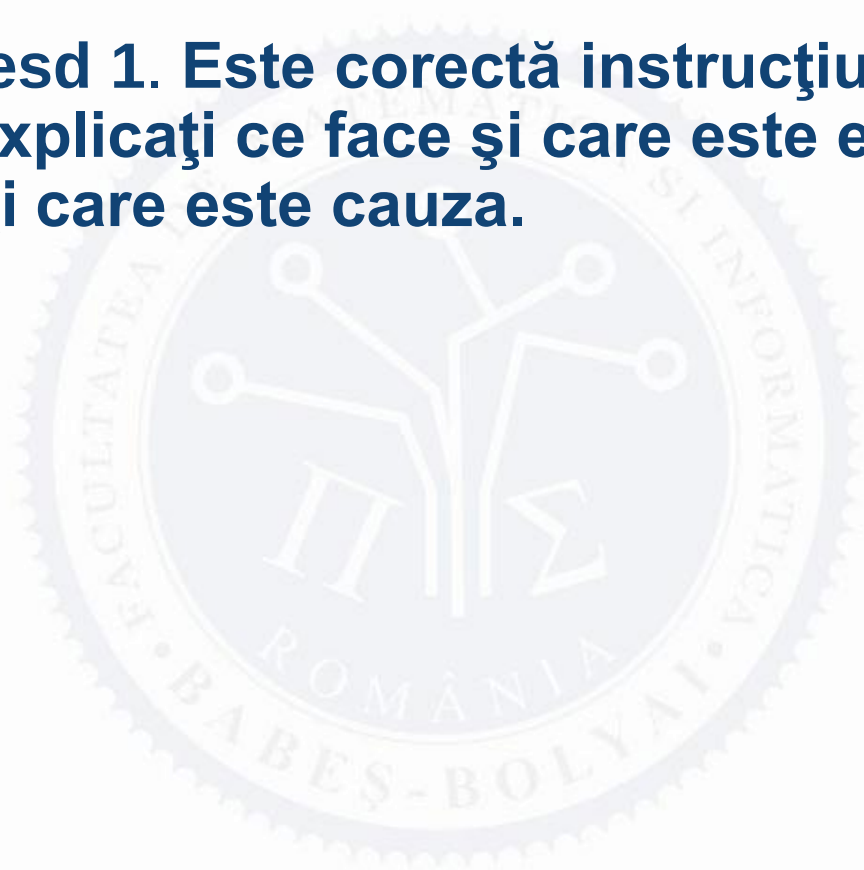


Examen scris: Exercițiul 9 - rezolvare

- 7). Scrieți o instrucțiune care să seteze simultan SF și ZF la valoarea 1. **(În afara de SAHF (care suprascrie complet registrul de flag-uri cu valorile din AH), NU EXISTA nici o alta instructiune și NU are sens, deoarece dacă $ZF=1$ înseamnă că rezultatul ultimei operații efectuate a fost 0 și atunci $SF=0$)**

Examen scris: Exercițiul 9 - rezolvare

- 8). Avem x resd 1. Este corectă instrucțiunea **xchg eax,x**? Dacă da, explicați ce face și care este efectul ei. Dacă nu, justificați care este cauza.



Examen scris: Exercițiul 9 - rezolvare

- 8). Avem $x \bmod 1$. Este corectă instrucțiunea **xchg eax,x**? Dacă da, explicați ce face și care este efectul ei. Dacă nu, justificați care este cauza.
- (Nu este corectă deoarece ambii operanzi trebuie să fie L-values – syntax error. „x” NU este o valoare ce poate fi atribuită, fiind o constantă de tip pointer, deci o valoare ce nu poate fi atribuită).



FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ UNIVERSITATEA BABEȘ-BOLYAI

Str. Mihail Kogălniceanu nr. 1
Cluj-Napoca, Cluj, România

www.cs.ubbcluj.ro