**4.2.1.3.** **Exemple şi exerciţii propuse.**

a). ………….

mov ah,0

mov al, -5 ;echivalent cu **mov al, 11111011b** deci echivalent şi cu **mov al, 251**

;echivalent şi cu **mov al, 0fbh** în hexazecimal; f = 1111b şi b=1011b

;ansamblul celor 2 instrucţiuni de mai sus este echivalent cu **mov ax,251** (echivalent în ;binar cu **mov ax, 0000000011111011b** şi în hexazecimal cu **mov ax, 00fbh**) însă nu şi ;cu **mov ax, -5** (echivalent în binar cu **mov ax, 1111111111111011b** şi în hexazecimal cu ;**mov ax, 0fffbh**) - diferenţa constă în completarea conţinutului lui AH cu 8 cifre binare 0 ;în cazul lui 251 – interpretare fără semn - şi respectiv cu 8 cifre binare 1 în cazul lui -5 ;adică în interpretarea cu semn)

mov bx,10

imul bx ;dx:ax := ax\*bx = 251 \* 10 = 2510 = 09CEh (în AX) şi DX:=0

(deşi aici prin **imul** e forţată interpretarea cu semn pentru AX, deoarece AX = 0000000011111011b , bitul de semn fiind 0, rezultă că AX = 251 în ambele interpretări)

mul bx ;idem – rezultatele sunt identice datorită observaţiei de mai sus

mov cl, -100 (= 9ch = 10011100b = 156 în interpretarea fără semn!)

idiv cl ;AX=2510; AL (câtul):= AX idiv (-100) = -25 (=e7);AH (restul):=10 (0ah)

;conţinutul lui AX este acum AX=0ae7h

imul cl ;AX:=AL\*CL = (-25)\*(-100) = 2500 (=09c4h)

add ax,10 ;AX:=AX+10 = 2500+10 = 2510 - refacerea valorii iniţiale din AX:=2510;

div cl ;AX=2510; AL (câtul) := AX div 156 = 16 (=10h); AH (restul) := 14 (0eh)

;conţinutul lui AX este acum AX=0e10h

Reluaţi discuţia, analizaţi şi justificaţi rezultatele furnizate în situaţia în care ultimele 5 instrucţiuni de mai sus devin (CL se înlocuieşte cu CX):

mov cx, -100

idiv cx

imul cx

add ax,10

div cx

b). Ce se întâmplă cu exemplul de mai sus dacă acesta devine:

mov ax, -5 ;echivalent cu mov al, -5

; cbw

mov bx,10

imul bx

Ce valoare se obţine acum ca rezultat ? Comparaţi rezultatul furnizat cu cel obţinut prin aplicarea operaţiei **mul bx** în loc de **imul bx**.

c). …………….

mov al, 251 ; 251 = 0fbh = -5 - deci instrucţiune echivalentă cu **mov al, -5**

(suma valorilor absolute ale reprezentărilor cu semn şi fără semn la nivelul unui **octet** de memorie este **256** - regulă practică !) – aici se verifică prin 256 = 5 + 251

mov cl,255

mov bl, 100

imul bl ; ax := al \* bl = -5 \* 100 = -500 = 0fe0ch = 65036

(suma valorilor absolute ale reprezentărilor cu semn şi fără semn la nivelul unui **cuvânt** de memorie este **65536** - regulă practică !) – aici se verifică prin 65536 = 500 + 65036

div cl ;div impune ca valoarea din ax să fie interpretată acum fără semn!

;al := ax div bl = 65036 div 255 = 255 = ffh ; restul în AH = 11 = 0bh

;deci conţinutul lui AX este acum AX = 0bffh

Ce se întâmplă dacă **“imul bl”** este înlocuită cu instrucţiunea **“mul bl”** ? Analizaţi şi explicaţi comparativ rezultatele furnizate.

Ce observaţi că se întâmplă dacă **“div cl”** este înlocuită cu instrucţiunea **“idiv cl”** ?

Încercaţi să (vă) explicaţi potenţiala cauză a unui astfel de comportament prin efectuarea “pe hârtie” a operaţiei “**idiv cl**” şi analizând apoi încadrarea valorilor astfel obţinute (cât şi rest) în dimensiunile de reprezentare puse la dispoziţie de o operaţie div sau idiv.

Dar dacă apoi **“div cl”** este înlocuită întâi cu **“div cx”** iar apoi cu **“idiv cx”** ? Justificaţi şi explicaţi rezultatele obţinute. De ce în acest caz nu se mai manifestă situaţia apărută în cazul “**idiv cl**” ?