** Ministerul Educaţiei Republicii**

**Moldovei**

**Universitatea Tehnică a Moldovei**

[DEPARTAMENTUL INFORMATICĂ ŞI INGINERIA SISTEMELOR](https://utm.md/subdiviziuni-universitare/facultati/facultatea-calculatoare-informatica-si-microelectronica/catedra-calculatoare/)

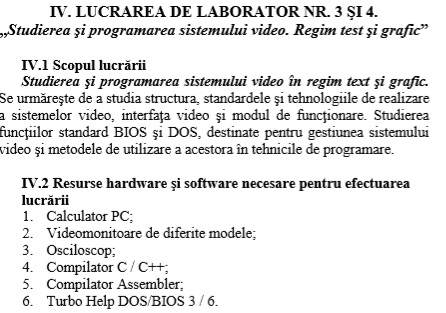
**Raport**

Lucrare de laborator nr.2-3

A efectuat: Nicolenco Eugeniu, Gr. C-162

A verificat: Pr.Univ Ababii Victor

2018



**IV.3 Consideraţii teoretice Dispozitivele de afişare video sunt alcătuite din două componente de bază: - monitorul – video display care realizează afişarea efectivă pe baza comenzilor primite de la adaptor; - adaptorul video – generează comenzile de afişare către monitor, efectuând totodată şi prelucrarea finală a informaţiilor ce vor fi afişate**

**Adaptorul video este un ansamblu de circuite care realizează procesarea informaţiilor care se afişează pe monitor şi formează semnalele de comandă pentru monitor: - culoare fiecărui pixel RGB; - semnalele de sincronizare a baleierii de sus în jos şi de la stânga la dreapta; - semnalul de sincronizare a liniilor pe orizontală HSYN; - semnalul de sincronizare a cadrelor pe verticală VSYN. Componentele de bază ale unui adaptor video sunt:**

**- memoria video;**

**- controlorul grafic;**

**- controlorul de atribute;**

**- registrele de control;**

**- convertorul numeric-analog;**

**- conectorul pentru cuplarea monitorului;**

**- conectorul de magistrală;**

**- conectorul de extensie;**

**- video BIOS.**

**Standarde IBM şi VESA. O placă video furnizează semnale în unul dintre următoarele standarde de fabricaţie:**

**- MDA – monochrome display adapter;**

**- CGA – color graphics adapter;**

**- EGA – enhanced graphics adapter;**

**- VGA – video graphics array;**

**- SVGA – super VGA;**

**- XGA – extended graphics array.**

Codul sursa pentru procesorul 8086

name "vga"

; this program draws a tiny rectangle in vga mode.

org 100h

jmp code

; dimensions of the rectangle:

; width: 10 pixels

; height: 5 pixels

w equ 100

h equ 100

; set video mode 13h - 320x200

code: mov ah, 0

mov al, 13h

int 10h

; draw upper line:

mov cx, 100+w ; column

mov dx, 20 ; row

mov al, 15 ; white

u1: mov ah, 0ch ; put pixel

int 10h

dec cx

cmp cx, 100

jae u1

; draw bottom line:

mov cx, 100+w ; column

mov dx, 20+h ; row

mov al, 15 ; white

u2: mov ah, 0ch ; put pixel

int 10h

dec cx

cmp cx, 100

ja u2

; draw left line:

mov cx, 100 ; column

mov dx, 20+h ; row

mov al, 15 ; white

u3: mov ah, 0ch ; put pixel

int 10h

dec dx

cmp dx, 20

ja u3

; draw right line:

mov cx, 100+w ; column

mov dx, 20+h ; row

mov al, 15 ; white

u4: mov ah, 0ch ; put pixel

int 10h

dec dx

cmp dx, 20

ja u4

; draw diagonal line:

mov cx, 100+w ; column

mov dx, 20+h ; row

mov al, 15 ; white

u5: mov ah, 0ch ; put pixel

int 10h

dec cx

dec dx

;sub dx,2

cmp cx, 100

ja u5

; pause the screen for dos compatibility:

;wait for keypress

mov ah,00

int 16h

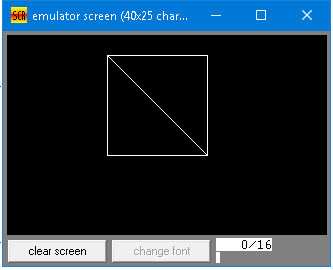
; return to text mode:

mov ah,00

mov al,03 ;text mode 3

int 10h

ret



name "keybrd"

org 100h

wait\_for\_key:

mov ah, 1

int 16h

jz wait\_for\_key

mov ah, 0

int 16h

; print the key:

mov ah, 0eh

int 10h

; press 'esc' to exit:

cmp al, 1bh

jz exit

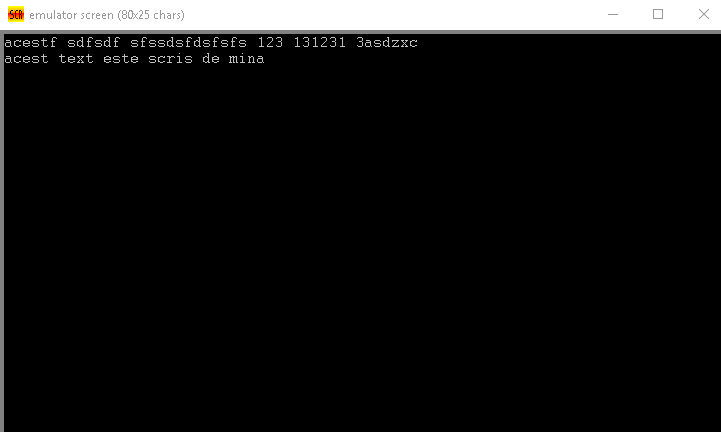
jmp wait\_for\_key

;============================

exit:

ret

end



**Concluzie:**

Regimul text/video este rulat de intreruperea int 10h si continutul registrului ah, 0- regim video; 00-regim text

Lucrarea m-a familiarizat cu limbajul masina assembler, am putut intelege un pic mai bine lucrul registrilor procesoarelor pe baza de intel 8086.

\*Nota : entru aceasta lucrare am utilizat emulatorul procesorului 8086 [emu8086.exe]