

Proyecto

CARRERA: Ingeniería en Computación

NOMBRE: *Efrain Robles Pulido*

CÓDIGO: 221350095

MATERIA: Traductores de Lenguajes I

MAESTRA: José Juan Meza Espinoza

SECCIÓN: D09

CALENDARIO: 2023A

UNIVERSIDAD DE GUADALAJARA



CUCEI



Desarrollo

Implementar el **Algoritmo de Bresenham de lineal recta** en el emu8086

Código:

org 100h

jmp start

msg: db "Introduzca coordenadas: ",24h

startX dw ?

startY dw ?

nextX dw ?

nextY dw ?

dsx dw ?

dsy dw ?

pk dw ?

cen db ?

dece db ?

uni db ?

COLOR EQU 0Fh ;La expresion COLOR es igual a 0Fh (255d), como constante numerica

macro DRAW ;La expresion DRAW se sustituirá por definicion para ejecutar el código dentro de el

mov ah,0Ch ;Escribe un punto en la pantalla en modo video, con lo que tenga en Cx y Dx, con la int 10h

mov al, COLOR ;Valor del color a pintar

int 10h

endm

start:

mov ax, @data ;Se almacena la direccion de memoria de nuestro arreglos

mov ds, ax ;Guardandola en el registro para las direcciones

mov ah,09h

mov dx,msg ;Instrucciones para la impresion de la cadena msg

int 21h

xor cx,cx

jmp entrada ;Funcion para introducir 1er numero

X0:

mov startX,di

mov ah,02h

mov dl,32 ;Instrucciones para la impresion de caracter de la cadena menu

int 21h

jmp entrada

Y0:

mov startY,di

mov ah,02h

mov dl,10 ;Instrucciones para la impresion de caracter

int 21h

mov dl,13 ;Instrucciones para la impresion de caracter

int 21h

mov ah,09h

mov dx,msg ;Instrucciones para la impresion de la cadena msg

int 21h

jmp entrada

X1:

mov nextX,di

mov ah,02h

mov dl,32 ;Instrucciones para la impresion de caracter

int 21h

jmp entrada

Y1:

mov nextY,di

jmp video

entrada:

xor ax,ax ;Limpiamos

mov ah, 01h

int 21h ;Entrada del 1ndo digito

sub al,30h ;Lo convertimos en numero desde el ASCII

mov cen,al ;Guardar en memoria

int 21h ;Entrada del 2ndo digito

sub al,30h ;Lo convertimos en numero desde el ASCII

mov dece,al

int 21h ;Entrada del 3er digito

sub al,30h ;Lo convertimos en numero desde el ASCII

mov uni,al ;Guardar en memoria

xor ax,ax ;Limpiamos

xor bx,bx

;Juntamos las cifras en el regirto Di, multiplicando por

;multiplos de 10 con su respectiva cifra para sumarse

mov bl,uni

mov di,bx

xor ax,ax

```
mov al,dece
mov bx,10
mul bx
add di,ax
```

```
xor ax,ax
mov al,cen
mov bx,100
mul bx
add di,ax ;Resultado final en Di
```

```
xor bx,bx ;Limpiamos registros
xor ax,ax
xor dx,dx
```

```
inc cx ;Incrementara nuestro contador
```

```
cmp cx, 1
je X0
cmp cx, 2
je Y0
cmp cx, 3
je X1
cmp cx, 4
je Y1
```

video:

```
mov ah, 0 ;Establece el modo video
mov al, 13h ;320 x 200 en grafico
int 10h
```

; Obtener las coordenadas de inicio y siguiente

```
mov ax, startX
mov bx, startY
mov cx, nextX
mov dx, nextY
```

; Calcular las diferencias dx y dy

```
sub cx, ax ;dx
sub dx, bx ;dy
mov dsx, cx
mov dsy, dx
```

```
mov ax, dsy
CWD
mov bx, dsx
idiv bx ;calculo de M
```

```
cmp ax, 1
jg end
```

```
mov ax, dsy
sub ax,bx ;calculo de (dy-dx) = pk
mov bx,2
mul bx
```

```
mov pk, ax
```

```
cmp ax,0
```

```
jle L1
```

```
jg L2
```

```
L1: ;Si pk es menor o igual a 0
```

```
add startX,1
```

```
mov ax, dsy
```

```
CWD
```

```
mov bx, 2
```

```
imul bx
```

```
add pk, ax
```

```
jmp draw
```

```
L2: ;Si pk es mayor a 0
```

```
add startX, 1
```

```
add startY, 1
```

```
mov ax, dsy
```

```
CWD
```

```
mov bx, 2
```

```
imul bx
```

```
mov cx, ax
```

```
mov ax, dsx
```

```
CWD
```

```
imul bx
```

```
sub cx,ax
```

```
add pk, cx
```

```
draw: ;Dibujara el punto en pantalla
```

```
mov cx, startX
```

```
mov dx, startY
```

```
DRAW
```

```
;Verificacion de que ya alcanzo los puntos
```

```
cmp cx, nextX
```

```
jg end
```

```
cmp dx, nextY
```

```
jg end
```

```
cmp pk, 0
```

```
jle L1
```

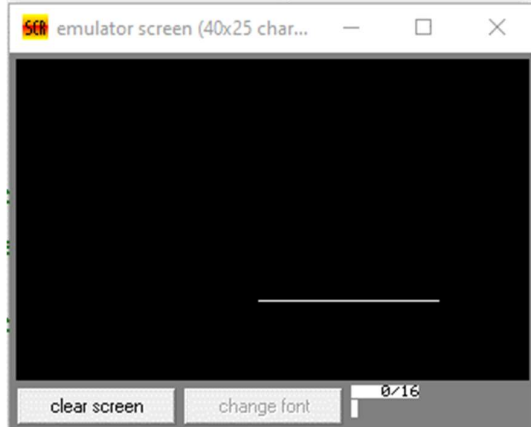
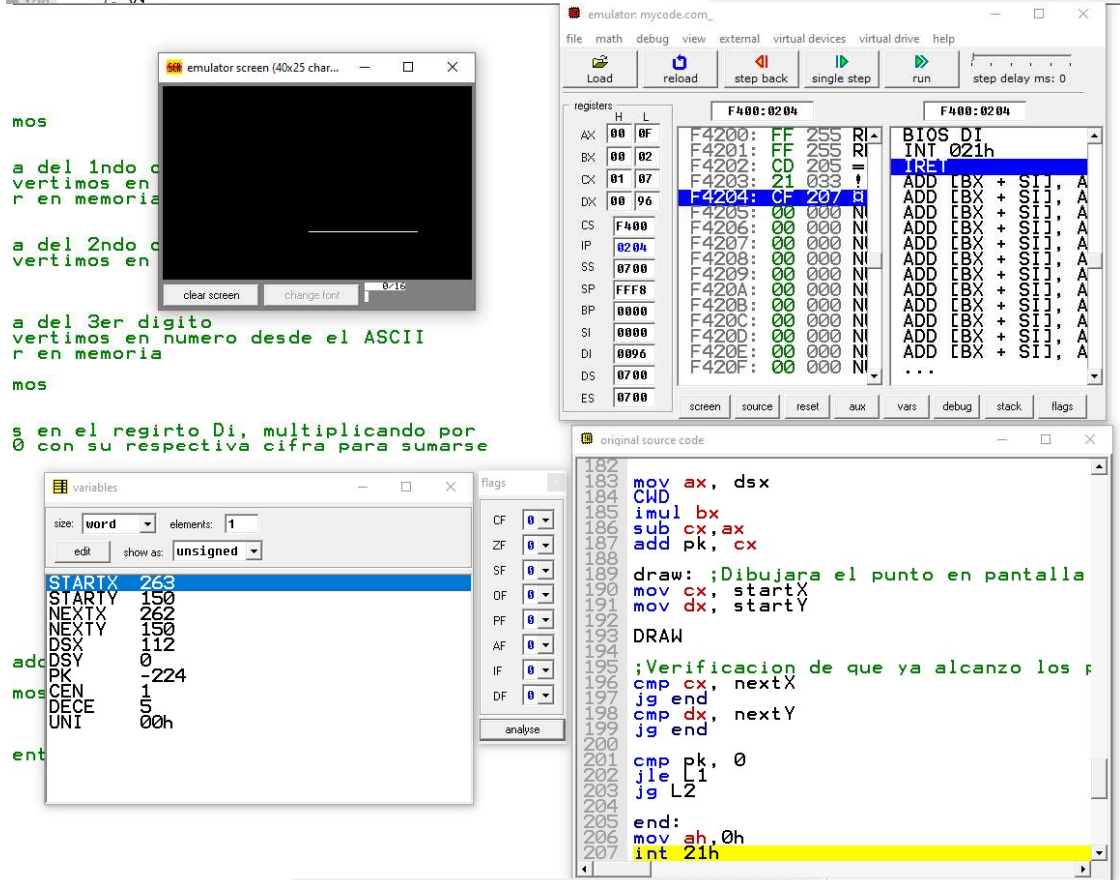
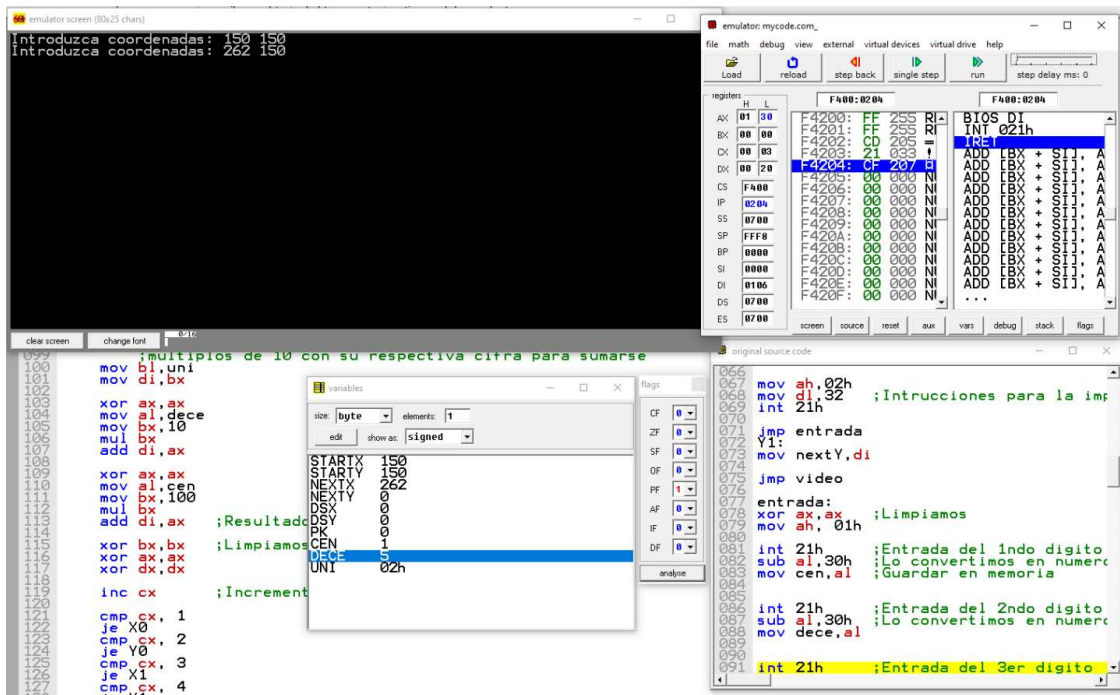
```
jg L2
```

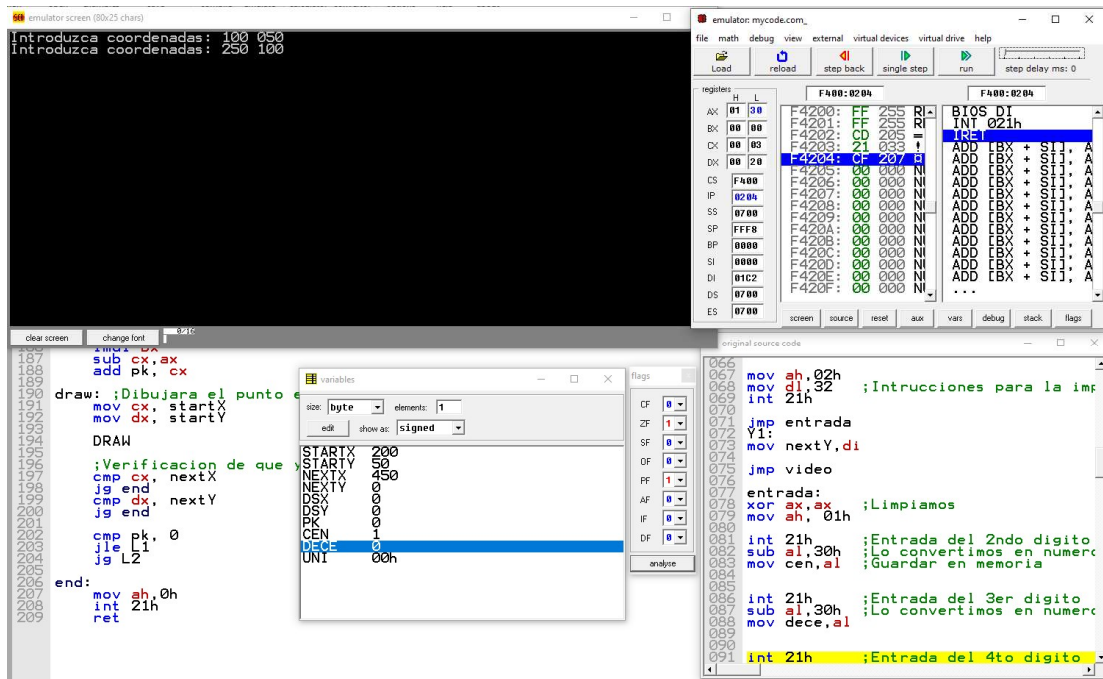
```
end:
```

```
mov ah,0h
```

```
int 21h
```

```
ret
```



nos

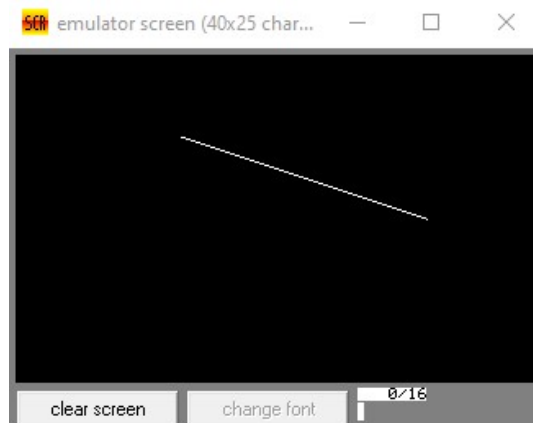
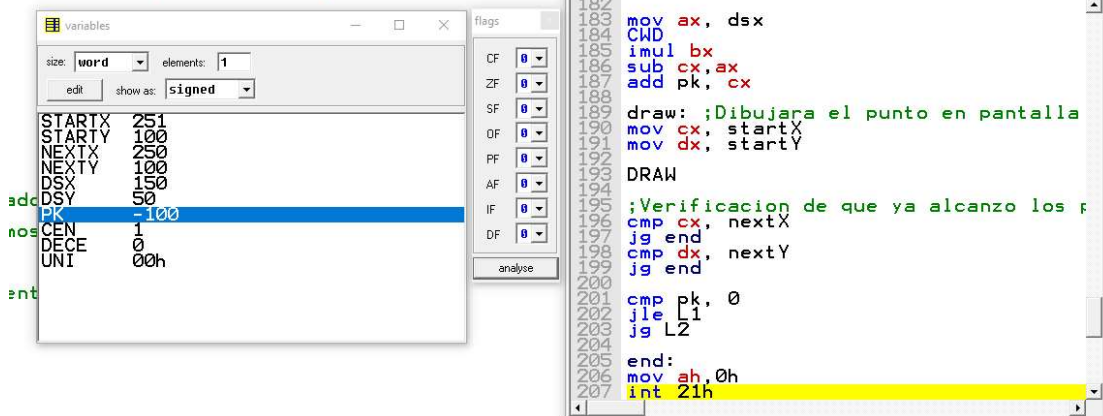
a del 1no
vertimos
en memoria

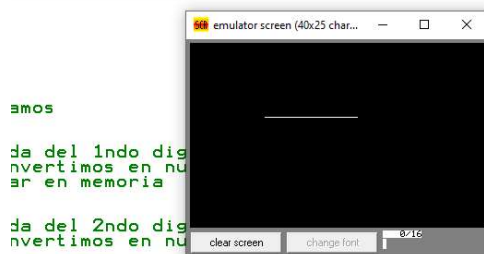
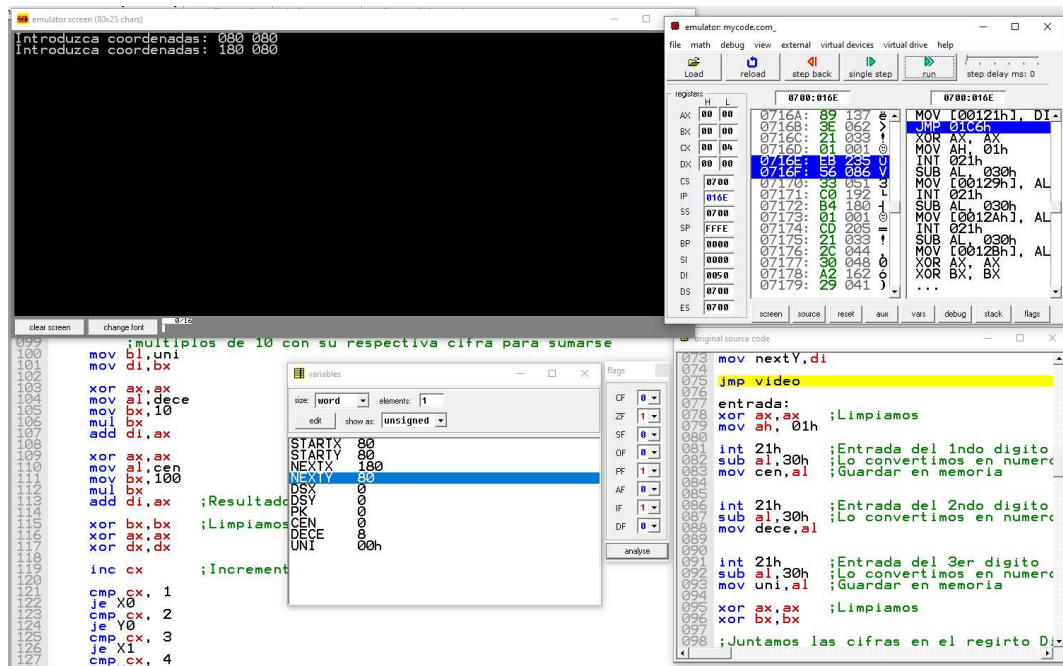
a del 2no
vertimos

a del 3er digito
vertimos en numero desde el ASCII
en memoria

nos

en el registro Di, multiplicando por
2 con su respectiva cifra para sumarse



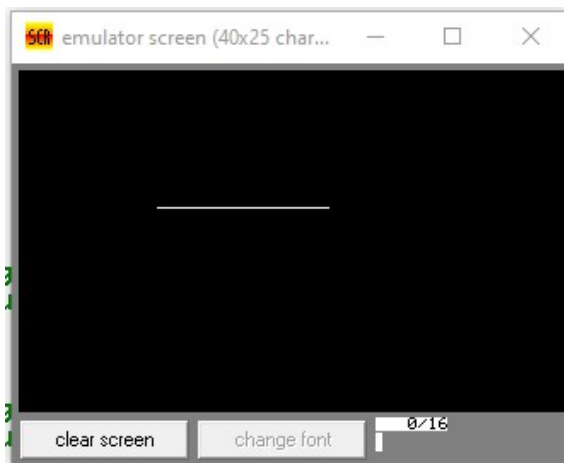
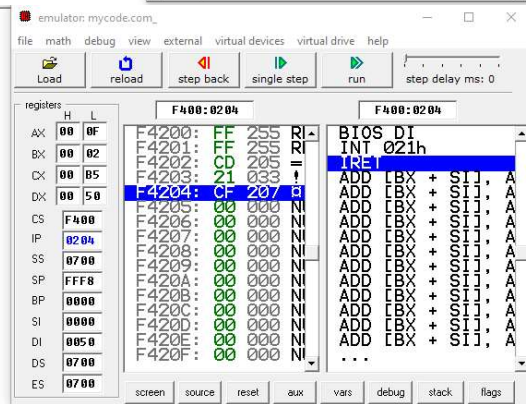
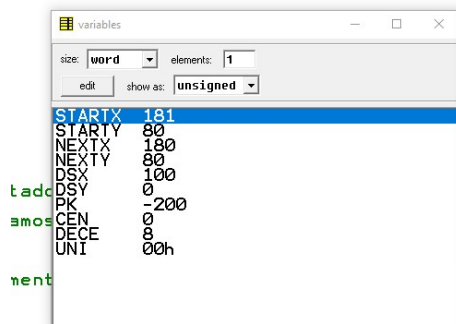


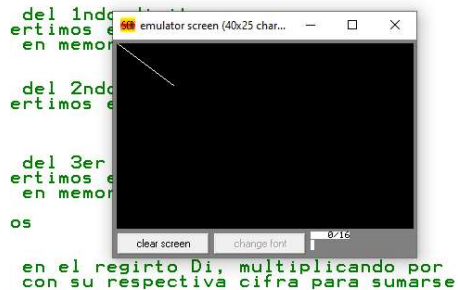
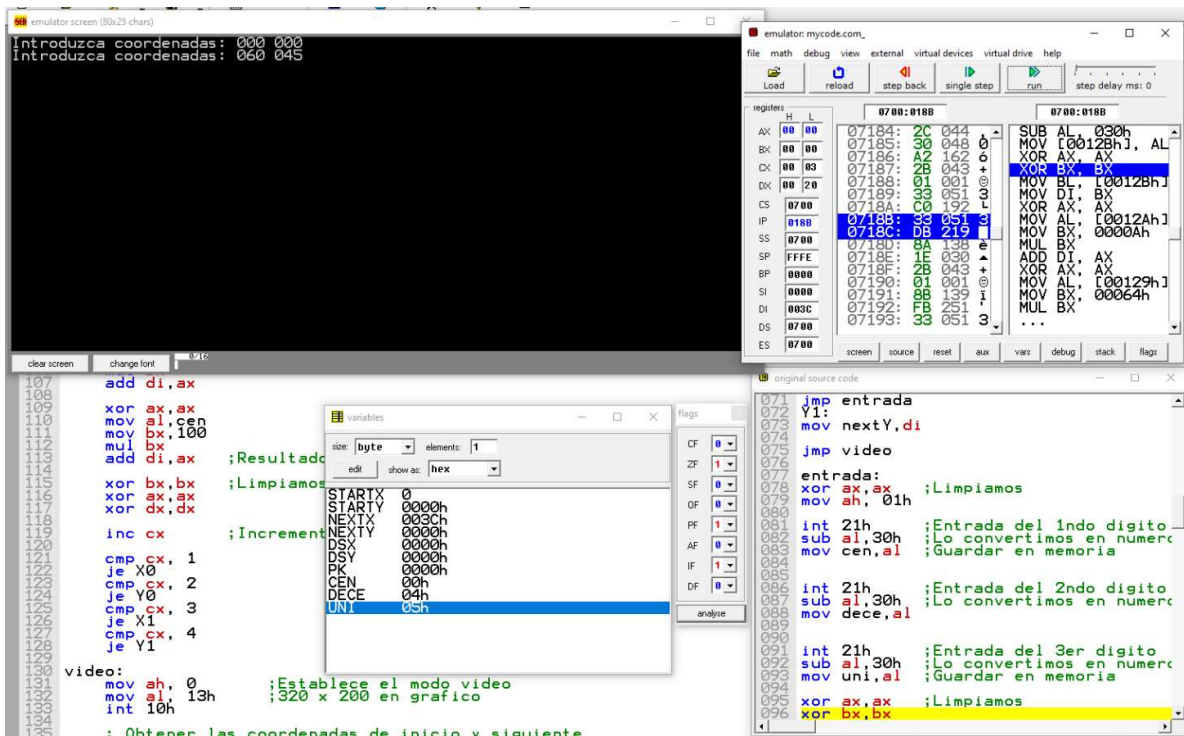
amos

da del 3er digito
nvertimos en numero desde el ASCII
ar en memoria

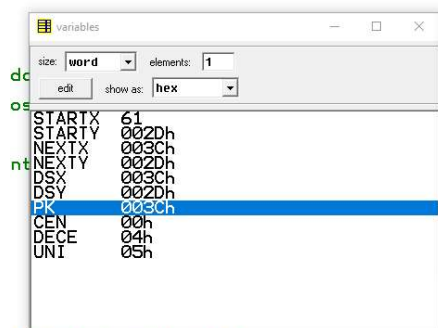
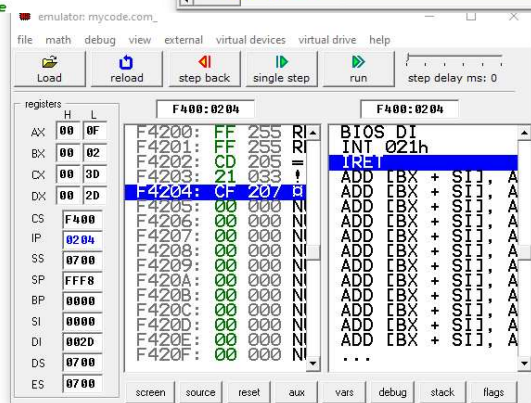
amos

as en el registro DI, multiplicando por
10 con su respectiva cifra para sumarse



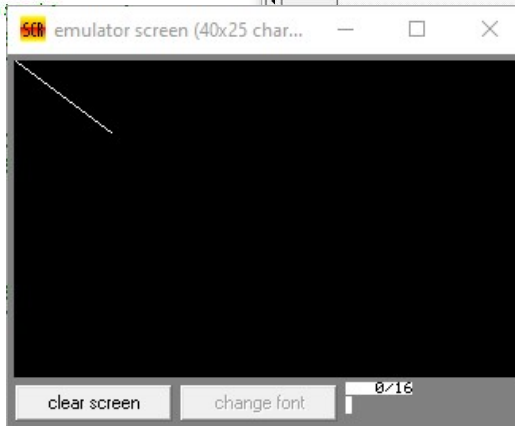


en el registro Di, multiplicando por su respectiva cifra para sumarse



establece el modo video
x 200 en grafico

coordenadas de inicio y siguiente



Conclusión

Con esta actividad me fue muy complicada de entender al principio ya que anteriormente realice esta práctica en otro lenguaje, pero es muy diferente en ensamblador, por lo que se requirió mas instrucciones de los que estaba acostumbrado para poder realizarla correctamente, además de reutilizar algunos códigos anteriores para poder introducir desde consola los valores de las coordenadas. A pesar de que me base en los videos de apoyo del profesor, me encontré con errores de implementación o se podría decir de incoherencias matemáticas, como teniendo que ser necesario que la primera coordenada este cerca del origen de la grafica para que funcione bien el código debido a los cálculos de PK y M, de acuerdo las diferencias de los ejes "x" y "y".

Bibliografía:

Brey, B. B. (2006). Microprocesadores Intel - 7 Edición (7a). Pearson Publications Company.