ELEMENTS OF COMPUTER SYSTEMS 1

END SEMESTER PROJECT

Submitted by

TEAM 4

ROLL NUMBER	NAME
AM.EN.U4AIE21008	Akarsh S Nair
AM.EN.U4AIE21011	Alfy Alex
AM.EN.U4AIE21048	Nayan M.K
AM.EN.U4AIE21060	Shyamdev Krishnan J
AM.EN.U4AIE21042	Santhosh Mamidisetti

TABLE OF CONTENTS

Pseudo Code:

Pseudo code

```
KEYBOARD:
    adress = SCREEN + 3840
    length = 16
    column = 0
    nill   = 8192
    if KBD =! 0 , go to MID
    goto RESET
MID:                          // To draw rectangle at middle
    RAM [address] = -1
    length = length - 1
    if length = 0 , go to UPDATE
    address = address + 32
    go to MID,
UPDATE:
    column = column + 1
    if column = 32, go to RECT 1
    address = SCREEN + column + 3840
    length = 16
    go to MID
```

```
RECT 1 :                    // To draw rectangle at top left corner
    address = SCREEN
    length = 60
    column = 0

LOOP 1 :
    RAM[address] = -1
    length = length - 1
    if length = 0, go to VPLOOP 1
    address = address + 32
    go to LOOP 1.

VPLOOP 1 :
    column = column + 1
    if column = 8, goto RECT 2
    address = SCREEN + column
    length = 60
    go to LOOP 1

RECT 2 :                    // To draw rectangle at bottom right corner.
    address = SCREEN + 6296
    length = 60
    column = 0
```

```
LOOP 2 :
    RAM [address] = -1
    length = length - 1
    if length = 0 , goto   VPLOOP 2
    address = address + 32
    goto  LOOP 2

VPLOOP 2 :
    column = column + 1
    if column = 8 , go-to  KEYBOARD
    address = ~~addres~~ SCREEN + column + 6296
    length = 60
    go to LOOP 2

RESET :
    address = SCREEN

BLANK :
    RAM [address] = 0
    address = address + 1
    nill = nill - 1
    if nill > 0 , goto BLANK
    go to KEYBOARD
```

Hack Assembly Code :

project - Notepad

File Edit Format View Help

```
(KEYBOARD)
    @SCREEN
    D = A
    @3840
    D = D + A
    @address
    M = D          //address = SCREEN + 3840

    @16
    D = A
    @length
    M = D         //length = 16

    @column
    M = 0        //column = 0

    @8192
    D = A
    @nill
    M = D        //nill= 8192

    @KBD
    D = M
    @MID
    D; JNE       //if KBD != 0 , go to MID

    @RESET
    0; JMP       //goto RESET

(MID)
    @address
    A = M
    M = -1        //RAM[address]= -1

    @length
    MD = M - 1
    @UPDATE
    D; JEQ

    @32
    D = A
    @address
```

```
    M = M + D      // address =address +32

    @MID
    0; JMP

(UPDATE)
    @column
    M = M + 1     //column= column + 1

    @32
    D = A
    @column
    D = D - M
    @RECT1
    D; JEQ       //if column = 32 ,go to RECT1

    @column
    D = M
    @SCREEN
    D = D + A
    @3840
    D = D + A
    @address
    M = D        //address = column + SCREEN + 3840

    @16
    D = A
    @length
    M = D       // length=16

    @MID
    0; JMP

(RECT1)           //To draw rectangle at top rigth corner
    @SCREEN
    D = A
    @address
    M = D          //address = screen

    @60
    D = A
    @length
```

```
    M = D         // length = 60

    @column
    M = 0

(LOOP1)
    @address
    A = M
    M = -1      //RAM[address]=-1

    @length
    MD = M - 1
    @UPLOOP1
    D; JEQ    //if length = 0 , goto UPLOOP

    @32
    D = A
    @address
    M = M + D    //address=address + 32

    @LOOP1
    0; JMP

(UPLOOP1)
    @column
    M = M + 1     //column = column + 32

    @8
    D = A
    @column
    D = D - M
    @RECT2
    D; JEQ        //if column = 8 , goto RECT2

    @column
    D = M
    @SCREEN
    D = D + A
    @address
    M = D       //address= SCREEN + column
```

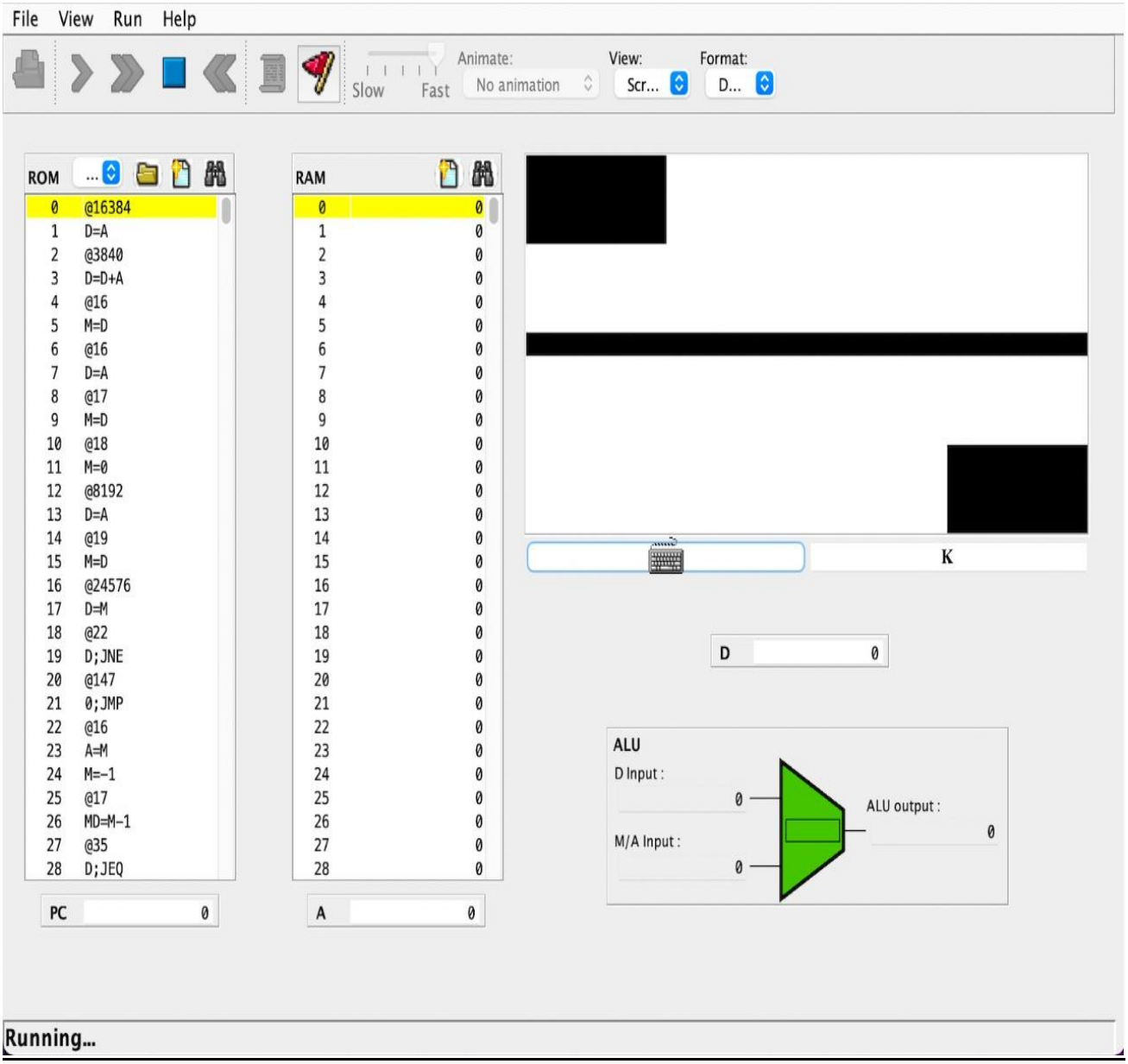
```
    @60
    D = A
    @length
    M = D     //length = 60

    @LOOP1
    0; JMP

(RECT2)           //To draw rectangle at bottom left corner
    @SCREEN
    D = A
    @6296
    D = D + A
    @address
    M = D         //adress = SCREEN + 6296

    @60
    D = A
    @length
    M = D

    @column
    M = 0


(LOOP2)
    @address
    A = M
    M = -1        //RAM[address]= -1

    @length
    MD = M - 1
    @UPLOOP2
    D; JEQ      // if length = 0 , goto UPLOOP2

    @32
    D = A
    @address
    M = M + D    //address= address + 32

    @LOOP2
    0; JMP
```

```
(UPLOOP2)
    @column
    M = M + 1     //coloumn = coloumn + 1

    @8
    D = A
    @column
    D = D - M
    @KEYBOARD
    D; JEQ      //if column = 8 , goto KEYBOARD

    @column
    D = M
    @SCREEN
    D = D + A
    @6296
    D = D + A
    @address
    M = D      //address= SCREEN + column + 6296

    @60
    D = A
    @length
    M = D     //length = 60

    @LOOP2
    0; JMP

(RESET)
    @SCREEN
    D = A
    @address
    M = D       //address = SCREEN


(BLANK)
    @address
    A = M
    M = 0      //RAM[address] = 0

    @address
    M = M + 1   //address= address + 1
```

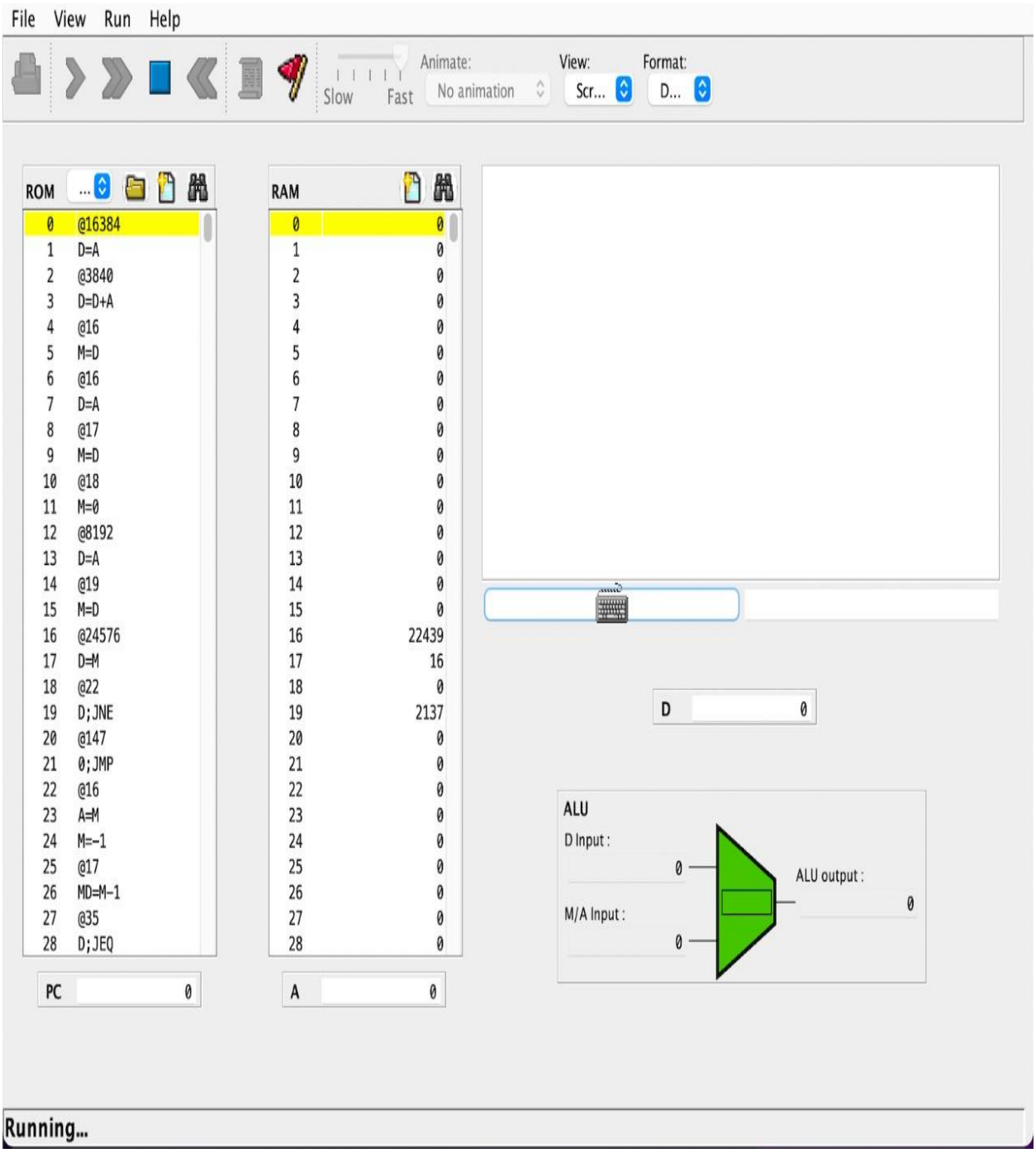
```
    @nill
    MD = M - 1   //nill = nill - 1
    @BLANK
    D; JGT       //if nill > 0 , goto BLANK

    @KEYBOARD
    0; JMP
```

Output snapshots of CPU Emulator :

When keyboard is pressed:

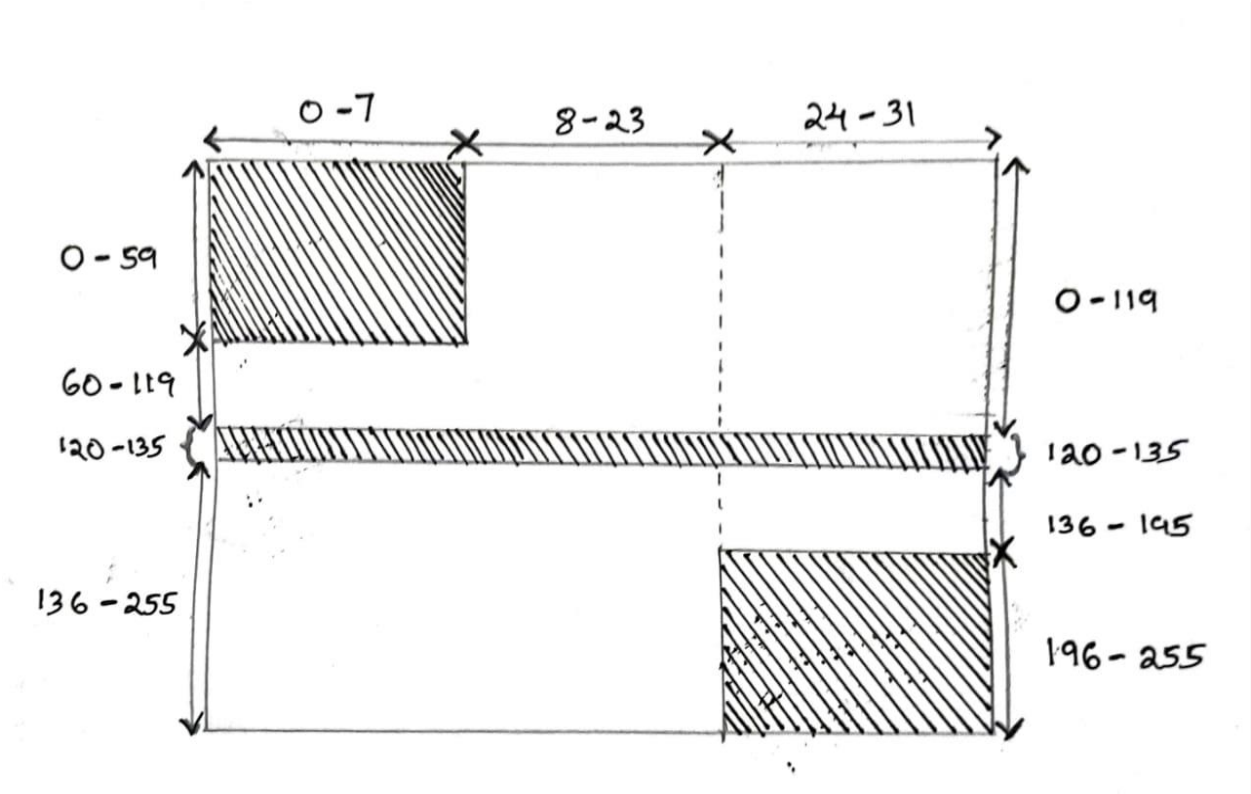


When keyboard is not pressed:



Insights learned about Hack Assembly code while implementing the project

- Came to know how to address each registers.
- Understood how to write to each pixel.
- Applying various conditional and unconditional loops to jump to different parts of the code where ever required.
- Got an idea about how to access the keyboard memory mapping.
- Came to know that for turning ON a pixel we write a '1' to the Bit and to turn it OFF we write a '0' to the Bit.
- Studied more about A and C instructions.
- We got a clear insight on using pointers to address the screen memory map.
- Accessing all the pixels on the screen using the screen memory map.
- Learned how to **on** specified pixels in the screen.
- Understood that variables are allocated to RAM[16] onward
- We came to know that it is very good practice to write pseudo code before writing the asm file .Since it helped us to reduce the chance of getting error in the codes .
- Using labels in the code had made the code more readable and easier to change the program flow.


0 - 7
8 - 23
24 - 31
0 - 59
60 - 119
120 - 135
136 - 255
0 - 119
120 - 135
136 - 195
196 - 255

****