**ELEMENTS OF COMPUTING SYSTEM [19AIE101]**

**S1 B. TECH CSE (AIE)**

Project Report

**ROLL NUMBER NAME**

AM.EN.U4AIE21138 Karthik G Nair

AM.EN.U4AIE21151 R Aravind

AM.EN.U4AIE21167 Vishnu Shaji

AM.EN.U4AIE21108 Akhilesh

AM.EN.U4AIE21181 Mamidi Sowji Krishna

**PSEUDO CODE:-**

**LOOP:**

If input in KBD is A (65) then goto DRAW

If not goto BLANK

**DRAW:**

i=16384

h=16384

R0=32

for(int j=0;j<64;j++0)

{

for(int k=0;k<8;k++)

{

//draw 16 black pixels

i++;

}

i=h+32;

h=h+32;

}

if(i==18432)//change the values of starting point for TOP RIGHT Rectangle and gotoDRAW

{

i=16408;

h=16408;

}

else if (i==18456)//change the values of starting point for BOTTOM LEFT Rectangle and gotoDRAW

{

i=22528;

h=22528;

}

else if(i==24576)//change the values of starting point for BOTTOM RIGHT Rectangle and gotoDRAW

{

i=22552;

h=22552;

}

Goto LOOP

**BLANK:**

i=16384

n=24576

z=i-n

if(z=0){

@i //draws 16 pixels of blank space

i++

}

Else

Goto LOOP

**HACK ASSEMBLY CODE:-**

(LOOP) //this will check whether to draw or not

@65 //ASCII value of A '65'

D=A

@24576

D=D-M

@DRAW //goto DRAW

D;JEQ

@BLANK //goto BLANK

D;JNE

(DRAW)

@16384

D=A

@i

M=D

@16384

D=A

@h

M=D

@32

D=A

@0

M=D

@i //Starting point of TOP LEFT rectangle

D=M

@h

(RECTANGLE) //Drawing a rectangle of 64 rows and 128 columns of pixels of

@0

D=D+M

@1

M=D

@j //using pointer for 64 rows

M=0

@64

D=A

@n

M=D

(CP)

@j

D=M

@n

D=D-M

@TR

D;JEQ

@k //using pointer for column

M=0

@8

D=A

@m

M=D

(RP)

@k

D=M

@m

D=D-M

@CRP

D;JEQ

@i

D=M

@k

A=D+M

M=-1 //drawing 16 black pixels

@k

M=M+1

@RP //goto column pointer

0;JMP

(CRP)

@1

D=M

@h

M=D

@h

D=M

@i

M=D

@i

D=M

@h

D=M

@0

D=D+M

@1

M=D

@j

M=M+1

@CP //goto row pointer

0;JMP

(TR)

@18432

D=A

@i

D=D-M

@BL

D;JNE

@16408 //Starting point of TOP RIGHT Rectangle

D=A

@i

M=D

@h

M=D

@j

M=0

@n

M=0

@RECTANGLE //goto rectangle

0;JMP

(BL)

@18456

D=A

@i

D=D-M

@BR

D;JNE

@22528 //Starting point of BOTTOM LEFT Rectangle

D=A

@i

M=D

@h

M=D

@j

M=0

@n

M=0

@RECTANGLE

0;JMP

(BR)

@24576

D=A

@i

D=D-M

@LOOP

D;JNE

@22552 //Starting point of BOTTOM RIGHT Rectangle

D=A

@i

M=D

@h

M=D

@j

M=0

@n

M=0

@RECTANGLE

0;JMP

(BLANK) //If KBD input is not A then it will blank the screen

@16384

D=A

@i

M=D

@24576

D=A

@n

M=D

(B)

@i

D=M

@n

D=D-M

@STOP

D;JEQ

@i

A=M

M=0

@i

M=M+1

@B

0;JMP

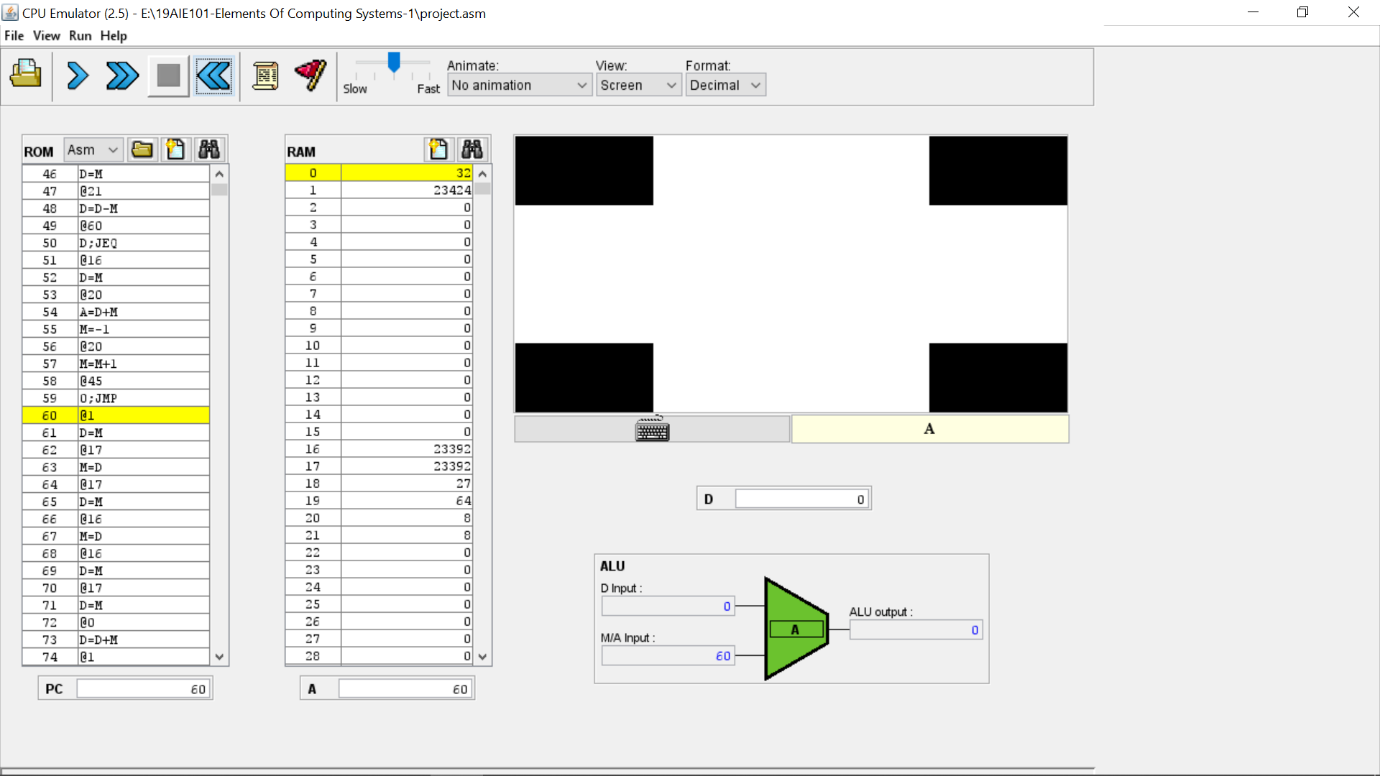
(STOP)

@LOOP //goto LOOP

0;JMP

**Output Snapshot:-**

When A is given as input in KEYBOARD



When A is NOT given as input in KEYBOARD

A picture containing graphical user interface

Description automatically generated

**INSIGHTS ABOUT HACK ASSEMBLY CODE:**

* HACK ASSEMBLY CODE is a low level programming for handling the screen according to several loops.
* We implemented address registers ,data registers and memory registers.
* For several calculations performed in loops we used variables as a reference to the data stored in each RAM location.
* We used keyboard as input and with using that input we handled the screen output.
* We used labels for jump statements and variables to reference of data stored in each RAM location.

**TEAM CONTRIBUTION**

The entire team has contributed equally. We had kept a meeting for almost 3 days and then we completed the whole project together. Everyone was actively participated in the meeting.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*