BSCS-3A

Assembly Language

Project Presentation

Hamza Haroon

BCS07203008

MY PROJECT - Flag of Pakistan

Introduction: What is Assembly Language

- Assembly Language was developed in 1951 by John Mauchly and John Eckert, who were working on ENIAC at the time.
- Assembly language is a low-level programming language that is used to create instructions for a computer's processor. Assembly language can be translated into machine code or object code which can then be executed by the computer.
- Assembly language is often used when the programmer wants to have direct control of the computer's processor and when they need to interact with hardware devices. It also makes the program more efficient and easier to understand for other programmers.

Project Theme: Flag of Pakistan

Symbolism

The green represents Islam as well as hope for peace between different cultures and religions. The white symbolizes honesty and integrity while the red stands for bravery. The crescent moon represents progress while its star stands for light or knowledge.

Color Code

Register – AL

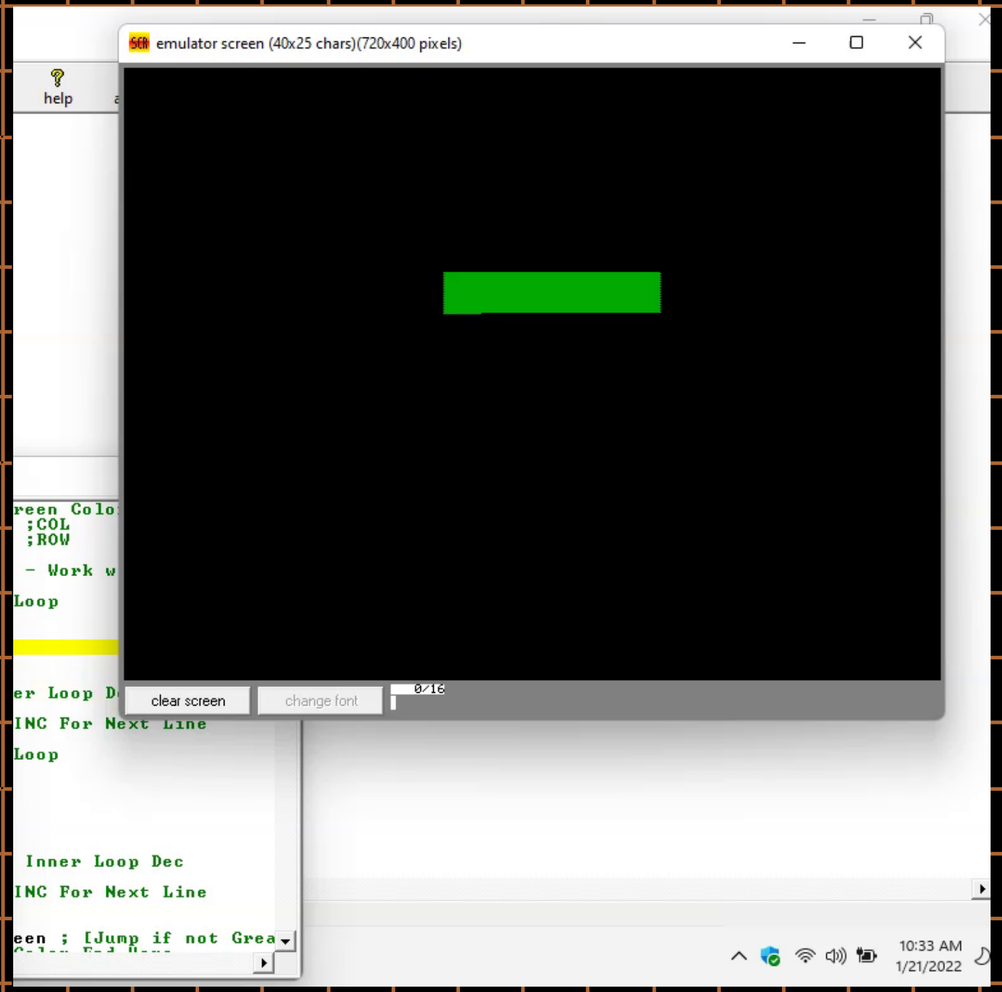
To Turn ON/OFF Pixel on screen we use Instruction AH=0CH and giving some color by placing value in AL Register. Some Color code are attahed on right side of slide.

Dec	Hex	Binary	Color
0	0	0000	Black
1	1	0001	Blue
2	2	0010	Green
3	3	0011	Cyan
4	4	0100	Red
5	5	0101	Magenta
6	6	0110	Brown

ROW AND COL

Row and Column

When we are in video mode the column and row are the main key-roll to work on screen and the value of these two is control by CX and DX Registers. The value of Column is store in CX and Value of ROWs store on DX register. we can also set manually value in these registers.

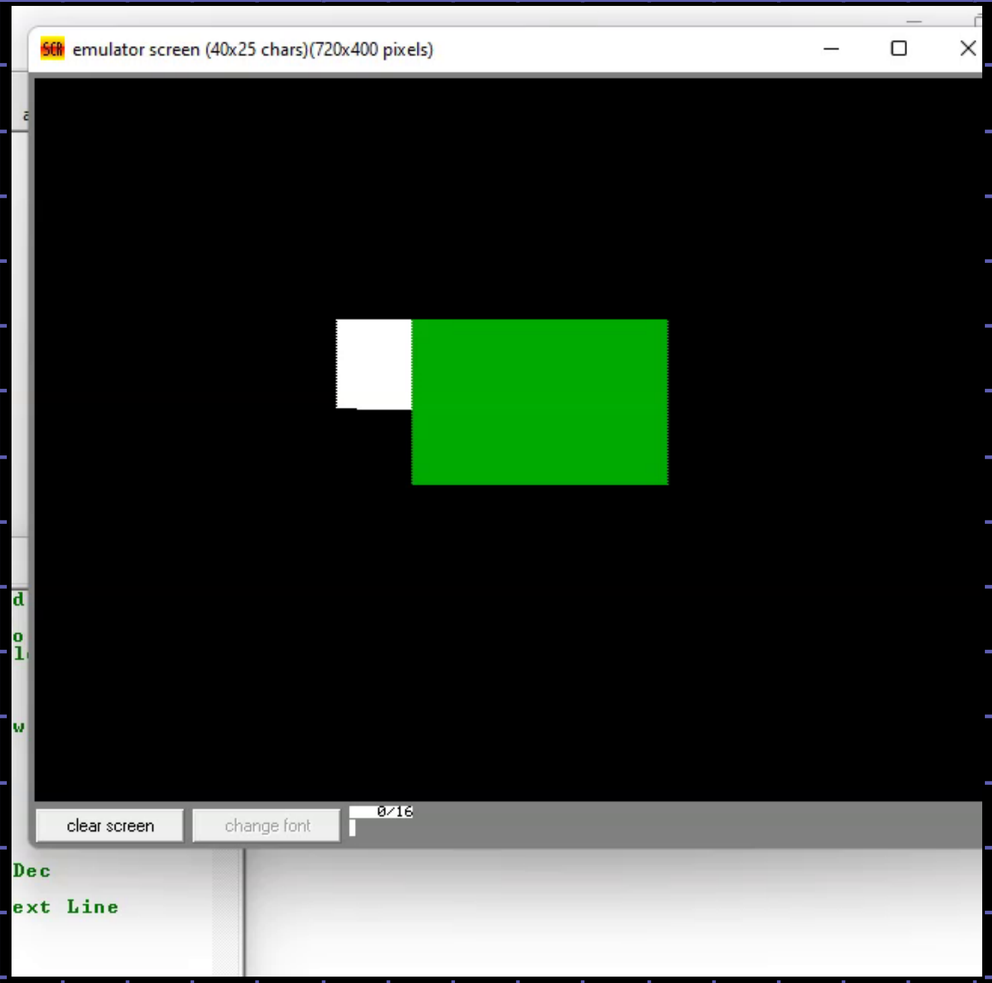


GREEN AND WHITE

LOOP LOGIC

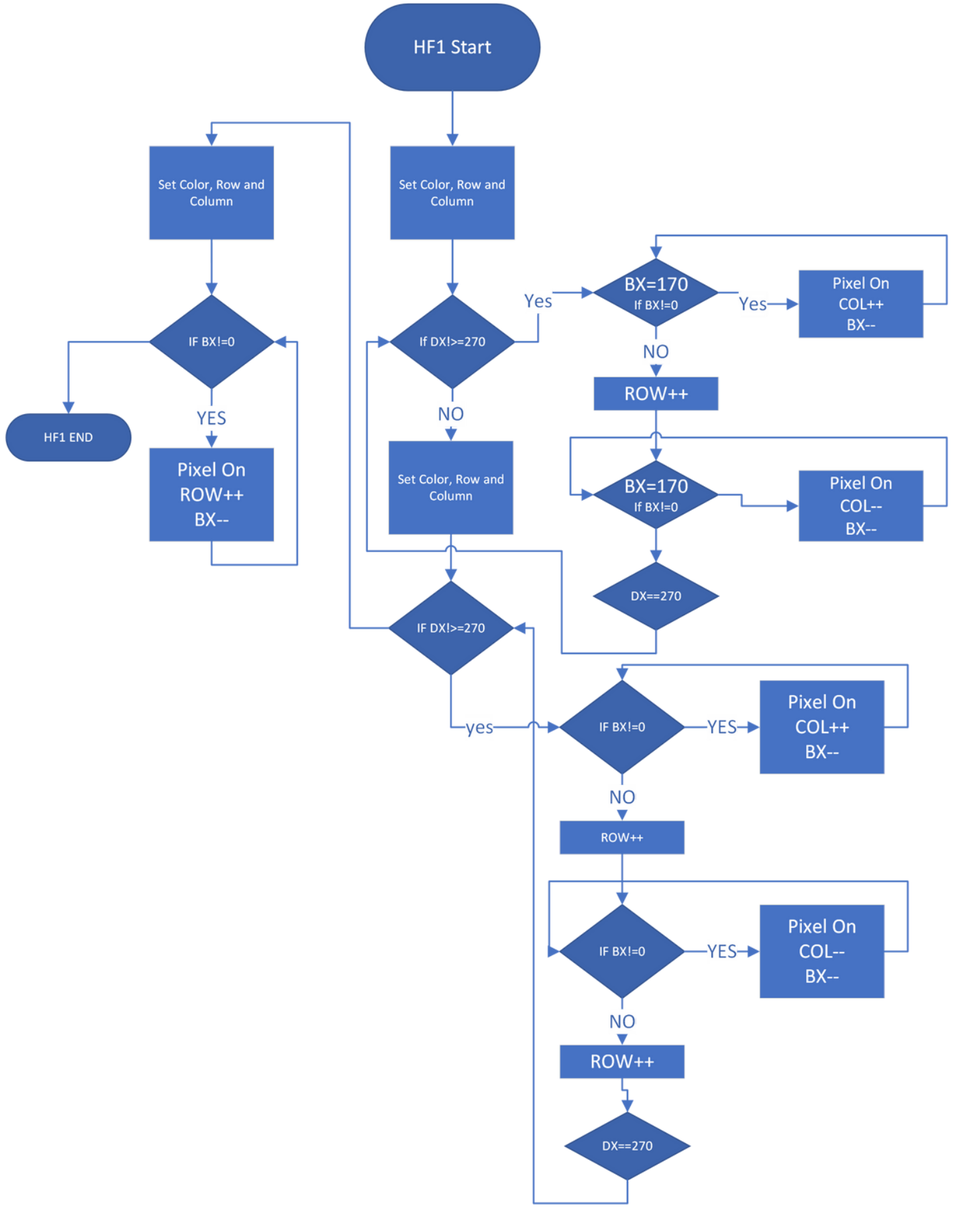
Our main goal is to print Rectangle with fill of Green and a Rectangle with White color on screen.

we are now working in pixel/ video mode, so we must turn every pixel in that space Turn on to get these done. I have built a flowchart which showing a complete guide how it can be done easily using nested loop in my code.

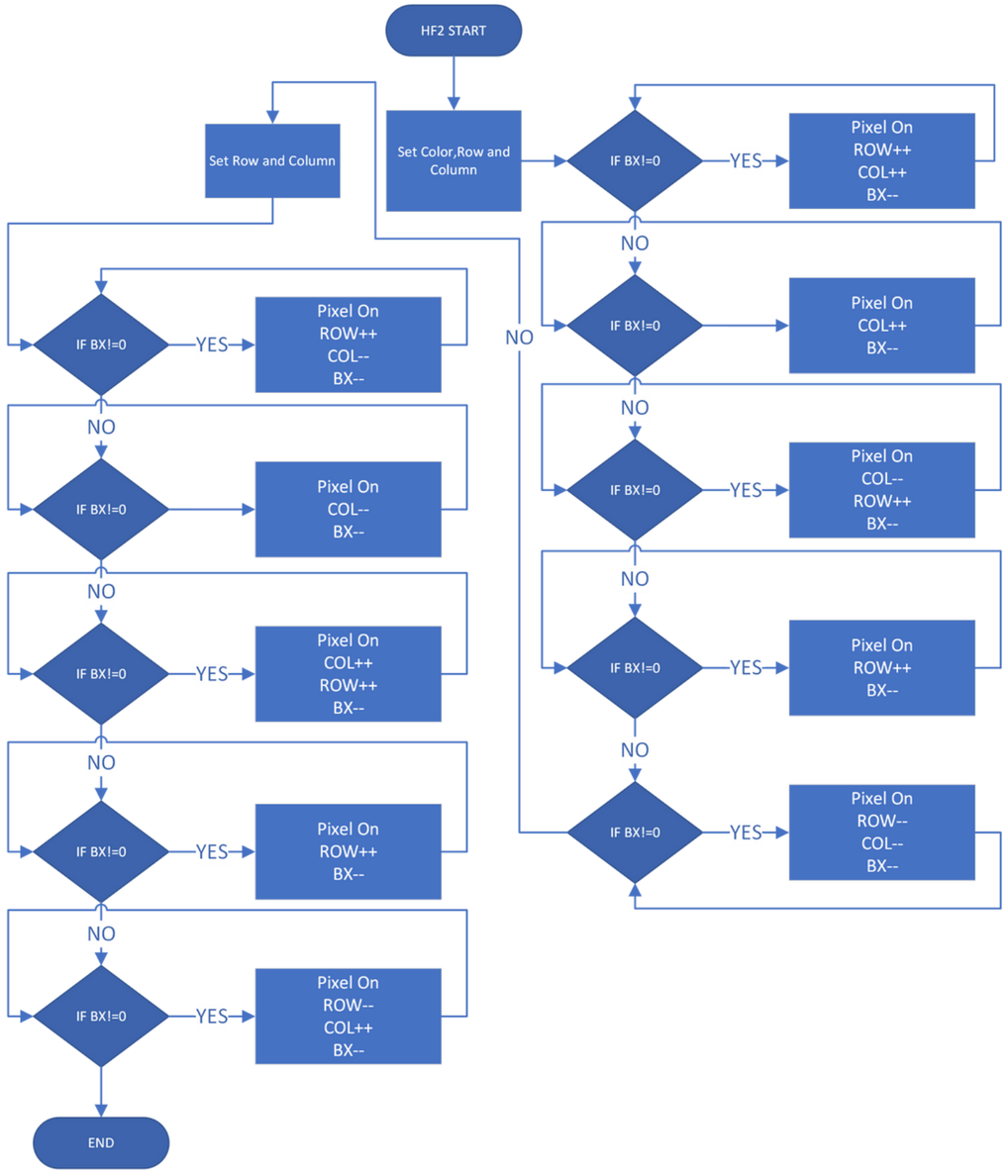


Controlling the loop counter I'm using the BX Roister as CX is using for the value of Column. We can use INC/DEC Command to change the value of any register or variable by One. and Using Labels to tell a loop to jump on a specific part of Code on a condition.

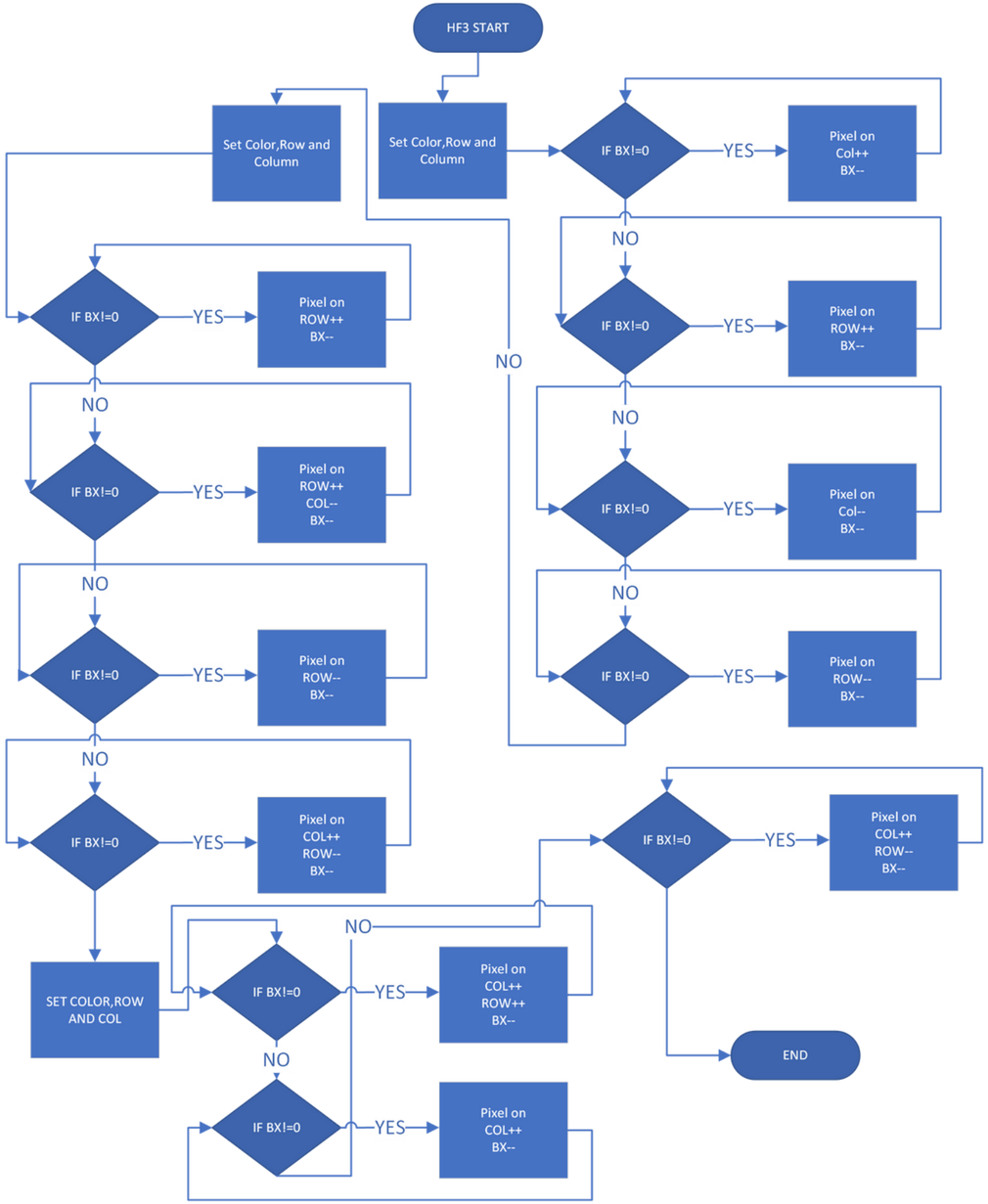
Print Green and White Color

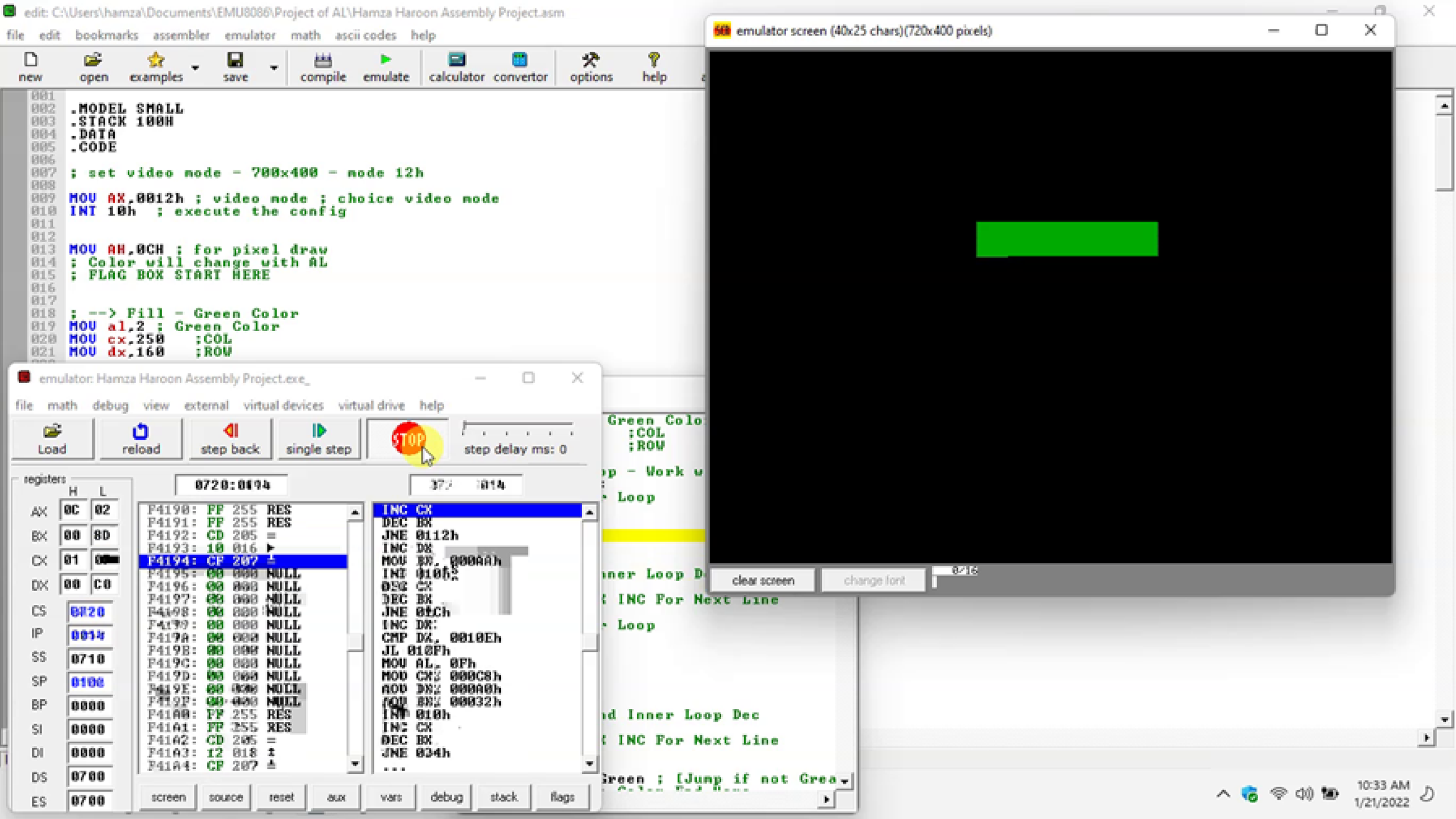


PRINT STAR



PRINT OUTLINE OF FLAG & ROD




edit: C:\Users\hamza\Documents\EMU8086\Project of AL\Hamza Haroon Assembly Project.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help
.MODEL SMALL
.STACK 100H
.DATA
.CODE
; set video mode - 700x400 - mode 12h
MOV AX,0012h ; video mode ; choice video mode
INT 10h ; execute the config
MOV AH,0CH ; for pixel draw
; Color will change with AL
; FLAG BOX START HERE
; --> Fill - Green Color
MOV al,2 ; Green Color
MOV cx,250 ;COL
MOV dx,160 ;ROW
emulator screen (40x25 chars)(720x400 pixels)
clear screen
change font
emulator: Hamza Haroon Assembly Project.exe
file math debug view external virtual devices virtual drive help
Load reload step back single step STOP step delay ms: 0
registers
H L
AX 0C 02
BX 00 8D
CX 01
DX 00 C0
CS 0720
IP 0014
SS 0710
SP 0100
BP 0000
SI 0000
DI 0000
DS 0700
ES 0700
0720:0014
INC CX
DEC BX
JNE 0112h
INC DX
INC CX
DEC BX
JNE 010Ch
INC DX
CMP DX, 0010Eh
JL 010Fh
MOV AL, 0Fh
INT 010h
INC CX
DEC BX
JNE 034h
screen source reset aux vars debug stack flags
Green Colo
;COL
;ROW
p - Work u
Loop
ner Loop D
INC For Next Line
Loop
d Inner Loop Dec
INC For Next Line
reen ; [Jump if not Grea
10:33 AM
1/21/2022

edit: C:\Users\hamza\Documents\EMU8086\Project of AL\Hamza Haroon Assembly Project.asm

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help

```
.MODEL SMALL
.STACK 100H
.DATA
.CODE

; set video mode - 700x400 - mode 12h

MOV AX,0012h ; video mode ; choice video mode
INT 10h  ; execute the config


MOV AH,0CH ; for pixel draw
; Color will change with AL
; FLAG BOX START HERE


; --> Fill - Green Color
MOV al,2 ; Green Color
MOV cx,250    ;COL
MOV dx,160    ;ROW
```

emulator screen (40x25 chars)(720x400 pixels)

clear screen change font

emulator: Hamza Haroon Assembly Project.exe_

file math debug view external virtual devices virtual drive help

Load reload step back single step STOP step delay ms: 0

registers

	H	L
AX	0C	0F
BX	00	04
CX	00	CC
DX	00	FD
CS	0720	
IP	0042	
SS	0710	
SP	0100	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0720:0042

```
07240: 49 073 I
07241: 4B 075 K
07242: 75 117 u
07243: FA 250 ·
07244: 42 066 B
07245: 81 129 ü
07246: FA 250 ·
07247: 0E 014 ♫
07248: 01 001 ☺
07249: 7C 124 |
0724A: E6 230 µ
0724B: B0 176 ░
0724C: 06 006 ♠
0724D: B9 185 ╣
0724E: FA 250 ·
0724F: 00 000 NULL
07250: BA 186 ║
07251: A0 160 á
07252: 00 000 NULL
07253: BB 187 ╗
07254: 6E 110 n
```

0720:003E

```
DEC CX
DEC BX
JNE 013Eh
INC DX
CMP DX, 0010Eh
JL 0131h
MOV AL, 06h
MOV CX, 000FAh
MOV DX, 000A0h
MOV BX, 0006Eh
INT 010h
INC DX
DEC BX
JNE 056h
MOV AL, 0Fh
MOV CX, 00154h
MOV DX, 000C8h
MOV BX, 00006h
INT 010h
INC CX
...
```

screen source reset aux vars debug stack flags

10:47 AM 1/21/2022

edit: C:\Users\hamza\Documents\EMU8086\Project of AL\Hamza Haroon Assembly Project.asm

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help

```
001
002 .MODEL SMALL
003 .STACK 100H
004 .DATA
005 .CODE
006
007 ; set video mode - 700x400 - mode 12h
008
009 MOV AX,0012h ; video mode ; choice video mode
010 INT 10h  ; execute the config
011
012
013 MOV AH,0CH ; for pixel draw
014 ; Color will change with AL
015 ; FLAG BOX START HERE
016
017
018 ; --> Fill - Green Color
019 MOV al,2 ; Green Color
020 MOV cx,250   ;COL
021 MOV dx,160   ;ROW
```

emulator screen (40x25 chars)(720x400 pixels)

clear screen change font

emulator: Hamza Haroon Assembly Project.exe_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	0C	06
BX	00	00
CX	01	A4
DX	01	0E
CS	0720	
IP	015D	
SS	0710	
SP	0100	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0720:015D 0720:015D

```
0735B: 90 144 É  NOP
0735C: 90 144 É  NOP
0735D: F4 244 ⌠  HLT
0735E: 00 000 NULL  ADD [BX + SI], AL
0735F: 00 000 NULL  ADD [BX + SI], AL
07360: 00 000 NULL  ADD [BX + SI], AL
07361: 00 000 NULL  ADD [BX + SI], AL
07362: 00 000 NULL  ADD [BX + SI], AL
07363: 00 000 NULL  ADD [BX + SI], AL
07364: 00 000 NULL  ADD [BX + SI], AL
07365: 00 000 NULL  ADD [BX + SI], AL
07366: 00 000 NULL  ADD [BX + SI], AL
07367: 00 000 NULL  ADD [BX + SI], AL
07368: 00 000 NULL  ADD [BX + SI], AL
07369: 00 000 NULL  ADD [BX + SI], AL
0736A: 00 000 NULL  ADD [BX + SI], AL
0736B: 00 000 NULL  ADD [BX + SI], AL
0736C: 00 000 NULL  ADD [BX + SI], AL
0736D: 00 000 NULL  ...
```

screen source reset aux vars debug stack flags

10:49 AM 1/21/2022