Assembly Language Project

Traffic Lights System

Group members:

EB22210106081 Najeebullah

EB22210106014 Eman Ali Abbasi

EB22210106049 Md. Hammad Fayyaz

EB22210106085 Raza Muhammad

EB22210106063 Muhammad Eshaq

Project Description:

In this project we made a traffic light system by using assembly language. In which following are the features as given below:

1. **Red Light (Stop):** The initial state of the project is to show a red screen in which we used graphic properties of assembly language. The red appears and after 1 second a text “STOP” appears inside the red screen.
2. **Countdown:** After the appearance red light and STOP text, there's a countdown displayed from 5 to 0 which take time 1 second to each count down, presumably indicating the time remaining for the next signal change.
3. **Yellow Light (Ready):** Once the countdown reaches 0, the yellow light appears with a "Ready" text, preparing for the next phase.
4. **Green Light (Go):** Finally, the green light displays, indicating "Go," allowing the traffic to proceed.

CODE:

```
.model small
.stack 100h
.data


stop        db "       00000000000    000000000000000    000000      00000000", 10
            db "     000000000000000  000000000000000   0::::::0     000000000", 10
            db "   0::0          0::0 --------------- 00::::::::00   000---0::0", 10
            db "  0::0           0::0     0:::0     0:::::00:::::0  0:0    0::0", 10
            db "  0::0           0::0     0:::0     0::::0  0::::0  0:0    0::0", 10
            db "    0::0          0::0     0:::0     0:::0    0:::0  0:0    0::0", 10
            db "      0::0                0:::0     0:::0    0:::0  0:0    0::0", 10
            db "        0::0              0:::0     0:::0 00 0:::0  0:0   0::0", 10
            db "          0::0            0:::0     0:::0 00 0:::0  0:0-0::0", 10
            db "            0::0          0:::0     0:::0    0:::0  0:0", 10
            db "  0::0         0::0       0:::0     0:::0    0:::0  0:0", 10
            db "  0::0          0::0      0:::0     0::::0  0::::0  0:0", 10
            db "  0::0           0::0     0:::0     0:::: 00 ::::0  0:0", 10
            db "   0::0          0::0     0:::0      00::::::::00   0:0", 10
            db "     000000000000000      0:::0        0::::::0     0:0", 10
            db "       00000000000        0:::0         000000      0:0", 10, "$"

ready       db "      0:0          0000000000      0       000000     0000     0000", 10
            db "    00:::00        00:::::::0    00:00     0:0::00    :::0     0:::", 10
            db "  00:::::::00      00:::::::0  00:::::00   0:0::::00  ::::0   0::::", 10
            db "  0::::000:::0     00::        0:::0::::0  0:0   0::0 0::::0 0::::0", 10
            db "  0:::0   0:::0    00::        0::0  0::0  0:0   0::0  0::::_::::0", 10
            db "  0::0     0:::0   00::        0::0  0::0  0:0   0::0   0:::0:::0", 10
            db "  0::0     0:::0   00::        0::0  0::0  0:0   0::0    0:::::0", 10
            db "  0::0     0:::0   00::        0::0  0::0  0:0   0::0     0:::0", 10
            db "  0::0 000 0:::0   00:::::::0  0::0::0::0  0:0   0::0      0:0", 10
            db "  0::0 000 0:0     00:::::::0  0::0::0::0  0:0   0::0      0:0", 10
            db "  0::0    0:::0    00::        0::0  0::0  0:0   0::0      0:0", 10
            db "  0::0     0:::0   00::        0::0  0::0  0:0   0::0      0:0", 10
            db "  0::0      0:::0  00::        0::0  0::0  0:0   0::0      0:0", 10
            db "  0::0       0:::0 00::        0::0  0::0  0:0   0::0      0:0", 10
            db "  0::0       0:::0 00:::::::0  0::0  0::0  0:0::::00       0:0", 10
            db "  0__0       0____000:::::::0  0::0  0::0  0:0::00         0:0", 10, "$"
```

```
go          db "     000000000000000               000000000     ", 10
            db "  00:::00000000000000            00:::::::::00   ", 10
            db "00::::0           0:::0         00:::::::::::::00 ", 10
            db "0:::0              0:::0       0:::::::000:::::::0", 10
            db "0:::0               0:::0      0::::::0   0::::::0", 10
            db "0::0                0:::0      0:::::0     0:::::0", 10
            db "0::0                0:::0      0:::::0     0:::::0", 10
            db "0::0                           0:::::0 000 0:::::0", 10
            db "0::0       0000000000:::0      0:::::0 000 0:::::0", 10
            db "0::0       00000000000:::0     0:::::0     0:::::0", 10
            db "0::0                0:::0      0:::::0     0:::::0", 10
            db "0::0                0:::0      0::::::0   0::::::0", 10
            db " 0::0               0:::0      0:::::::000:::::::0", 10
            db "  0::0              0:::0       00:::::::::::::00 ", 10
            db "   0::0000000000000::0          00:::::::::00   ", 10
            db "     0000000000000                000000000     ", 10, "$"


zero        db "     000000000     ", 10
            db "   00:::::::::00   ", 10
            db " 00:::::::::::::00 ", 10
            db "0:::::::000:::::::0", 10
            db "0::::::0   0::::::0", 10
            db "0:::::0     0:::::0", 10
            db "0:::::0     0:::::0", 10
            db "0:::::0 000 0:::::0", 10
            db "0:::::0 000 0:::::0", 10
            db "0:::::0     0:::::0", 10
            db "0:::::0     0:::::0", 10
            db "0::::::0   0::::::0", 10
            db "0:::::::000:::::::0", 10
            db " 00:::::::::::::00 ", 10
            db "   00:::::::::00   ", 10
            db "     000000000     ", 10, "$"

 one        db "  1111111   ", 10
            db " 1::::::1   ", 10
            db "1:::::::1   ", 10
            db "111:::::1   ", 10
            db "   1::::1   ", 10
            db "   1::::1   ", 10
            db "   1::::1   ", 10
            db "   1::::l   ", 10
            db "   1::::l   ", 10
            db "   1::::l   ", 10
```

```
            db "   1::::1   ", 10
            db "   1::::1   ", 10
            db "111::::::111", 10
            db "1::::::::::1", 10
            db "1::::::::::1", 10
            db "111111111111", 10, "$"

    two   db " 222222222222222    ", 10
            db "2::::::::::::::22  ", 10
            db "2::::::222222:::::2 ", 10
            db "2222222     2:::::2 ", 10
            db "            2:::::2 ", 10
            db "            2:::::2 ", 10
            db "         2222::::2  ", 10
            db "    22222::::::22   ", 10
            db "  22::::::::222     ", 10
            db " 2:::::22222        ", 10
            db "2:::::2             ", 10
            db "2:::::2             ", 10
            db "2:::::2       222222", 10
            db "2::::::2222222:::::2", 10
            db "2::::::::::::::::::2", 10
            db "22222222222222222222", 10, "$"

three       db " 333333333333333   ", 10
            db "3::::::::::::::33 ", 10
            db "3::::::33333::::::3", 10
            db "3333333     3:::::3", 10
            db "            3:::::3", 10
            db "            3:::::3", 10
            db "    33333333:::::3 ", 10
            db "    3:::::::::::3  ", 10
            db "    33333333:::::3 ", 10
            db "            3:::::3", 10
            db "            3:::::3", 10
            db "            3:::::3", 10
            db "3333333     3:::::3", 10
            db "3::::::33333::::::3", 10
            db "3::::::::::::::33 ", 10
            db " 333333333333333   ", 10, "$"
```

```
four       db "       444444444  ", 10
           db "      4::::::::4  ", 10
           db "     4:::::::::4  ", 10
           db "    4::::44::::4  ", 10
           db "   4::::4 4::::4  ", 10
           db "  4::::4  4::::4  ", 10
           db " 4::::4   4::::4  ", 10
           db "4::::444444::::444", 10
           db "4::::::::::::::::4", 10
           db "4444444444:::::444", 10
           db "          4::::4  ", 10
           db "          4::::4  ", 10
           db "          4::::4  ", 10
           db "        44::::::44", 10
           db "        4::::::::4", 10
           db "        4444444444", 10, "$"

five        db "555555555555555555 ", 10
           db "5::::::::::::::::5 ", 10
           db "5::::::::::::::::5 ", 10
           db "5:::::555555555555 ", 10
           db "5:::::5            ", 10
           db "5:::::5            ", 10
           db "5:::::5555555555   ", 10
           db "5::::::::::::::5  ", 10
           db "555555555555:::::5 ", 10
           db "            5:::::5", 10
           db "            5:::::5", 10
           db "5555555     5:::::5", 10
           db "5::::::55555::::::5", 10
           db " 55:::::::::::::55 ", 10
           db "   55:::::::::55   ", 10
           db "     555555555     ", 10, "$"

.code
main proc

mov ax,@data
mov ds,ax

;red light
mov ah,6
mov al,00h
mov bh,01000000b
mov ch,0
```

```
mov cl,0
mov dx,184fh
int 10h

call delay
mov cx,974
mov si,offset stop


l9:
mov dx,[si]
mov ah,2
int 21h
inc si
loop l9

call delay
; call clear_screen

;Printing 5
mov cx,320
mov si,offset five


l6:
mov dx,[si]
mov ah,2
int 21h
inc si
loop l6

call delay
; call clear_screen

;Printing 4
mov cx,304
mov si,offset four


l5:
mov dx,[si]
mov ah,2
int 21h
inc si
loop l5
```

```
call delay
; call clear_screen

;printing 3
mov cx,320
mov si,offset three


l4:
mov dx,[si]
mov ah,2
int 21h
inc si
loop l4

call delay
; call clear_screen

;printing 2
mov cx,336
mov si,offset two


l3:
mov dx,[si]
mov ah,2
int 21h
inc si
loop l3

call delay
; call clear_screen

;printing 1
mov cx,208
mov si,offset one

l2:
mov dx,[si]
mov ah,2
int 21h
inc si
loop l2
```

```
call delay
; call clear_screen

;printing 0
mov cx,320
mov si,offset zero


l1:
mov dx,[si]
mov ah,2
int 21h
inc si
loop l1




call delay
call clear_screen

;yellow light
mov ah,6
mov al,00h
mov bh,11100000b
mov ch,0
mov cl,0
mov dx, 184fh
int 10h

mov cx,1040
mov si,offset ready


l7:
mov dx,[si]
mov ah,2
int 21h
inc si
loop l7

call delay
call clear_screen

;green light
```

```
mov ah,6
mov al,00h
mov bh,00100000b
mov ch,0
mov cl,0
mov dx,184fh
int 10h
mov cx,820
mov si,offset go


l8:
mov dx,[si]
mov ah,2
int 21h
inc si
loop l8

mov ah,4ch
int 21h
main endp

delay proc
mov cx,0
outer_loop:
mov dx,0FFFFH
inner_loop:
dec dx
jnz inner_loop
inc cx
cmp cx,30
jl outer_loop
ret
delay endp


clear_screen proc
    mov ah, 0fh
    int 10h

    mov ah, 0
    int 10h

    ret
clear_screen endp
```

```
end main
```

OUTPUT: