Traffic Light Simulator

By Team 15

TEAM MEMBERS

01

KUMAR LAXMIKANT

[CH.EN.U4CSE20039]

02

ADITYA BANSAL

[CH.EN.U4CSE20002]

03

ANIKET GUPTA

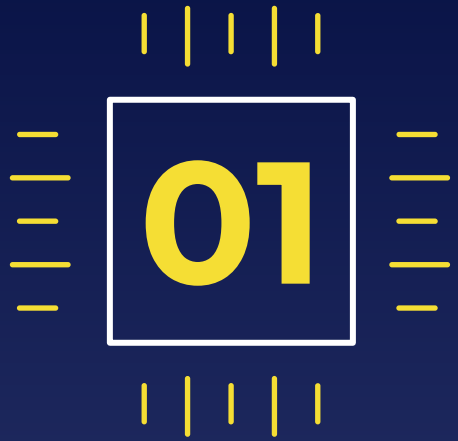
[CH.EN.U4CSE20004]

04

ANUSHKA NAMBIAR

[CH.EN.U4CSE20005]

TABLE OF CONTENTS



ABOUT PROJECT

Here we will describe and introduce the project

ABOUT PROJECT

This is a Traffic Light simulator coded in emu8086. Emu8086 is a Microprocessor Emulator with an integrated 8086 Assembler.

OBJECTIVE

- To design a working simulation of traffic-light on crossroad
- The long term goal of the project is to create a dynamic traffic light system that would avoid traffic and would ensure safety.
- It would decrease the accident level on roads due to traffic light failure.

ABSTRACT

- The efforts required in achieving the desired output can be effectively and economically be decreased by the implementation of better designs and algorithm. Traffic Lights play a crucial role in regulating traffic across street to prevent confusion and ambiguity, thus traffic light somehow also regulates our economy.

- The traffic light work on the principle of circular queue without any external intervention, thus this automated solution can help in reducing errors in operation to maximum extent.

- The significance and purpose of this project is to provide a algorithm that consumes less space and time.

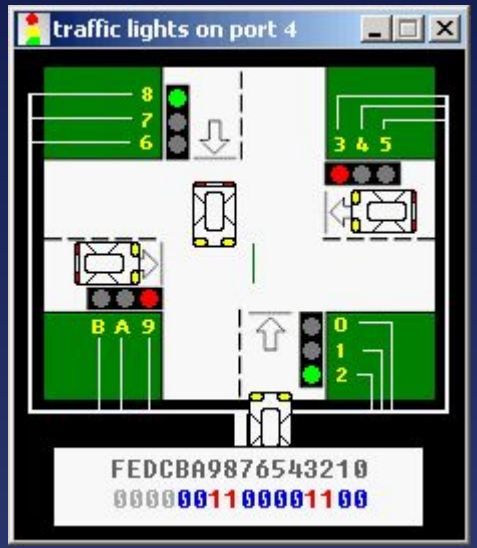
02

TECHNOLOGY USED

Here we will describe the technology used in our project

1) Emu8086

2) Traffic Light Simulator

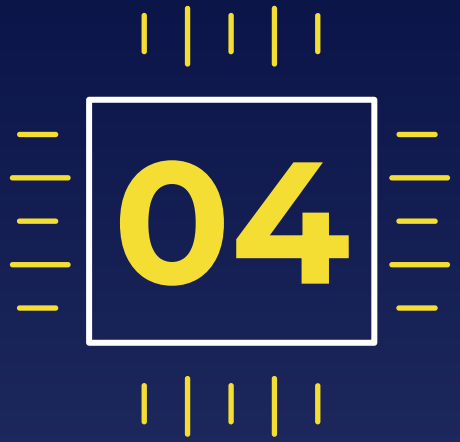


CODING

Here we will explain the coding implemented in our program

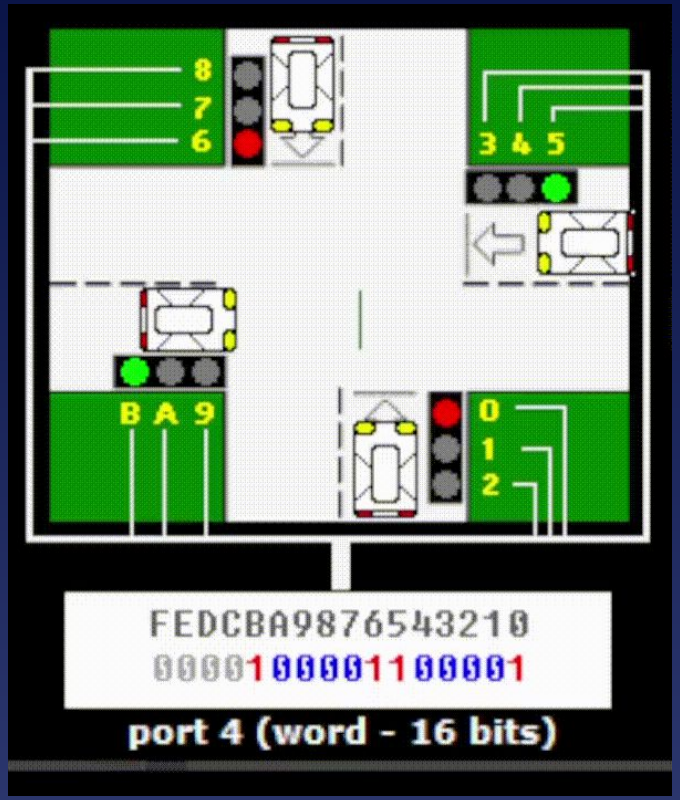
```
c:\emu8086\devices\Traffic_Lights.exe

#start=Traffic_Lights.exe#

name "traffic"


mov ax, all_red
out 4, ax


mov si, offset situation


next:
mov ax, [si]
out 4, ax

 ; wait 1 seconds (1 million microseconds)
mov     cx, 0FH
mov     dx, 4240H
mov     ah, 86h
int     15h
```

```
mov     cx, 0FH
mov     dx, 4240H
mov     ah, 86h
int     15h


add si, 2 ; next situation
cmp si, sit_end
jb  next
mov si, offset situation
jmp next


;                       FEDC_BA98_7654_3210
situation       dw      0000_0011_0000_1100b
s1              dw      0000_0110_1001_1010b
s2              dw      0000_1000_0110_0001b
s3              dw      0000_1000_0110_0001b
s4              dw      0000_0100_1101_0011b
sit_end = $


all_red         equ     0000_0010_0100_1001b
```



OUTPUTS

Here we would describe the output generated after executing the code


8
7
6
3 4 5
B A 9
0
1
2
FEDCBA9876543210
0000100001100001
port 4 (word - 16 bits)

THANK YOU