VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Computer Science & Engineering

CSE2006

Microprocessor and Interfacing

LAB ASSIGNMENT 5

Submitted to **Prof. SANJAY R**

TOPIC: ASSEMBLY LANGUAGE PROGRAMMING

NAME: PUNIT MIDDHA

REG.NO: 19BCE2060

SLOT: L43+L44

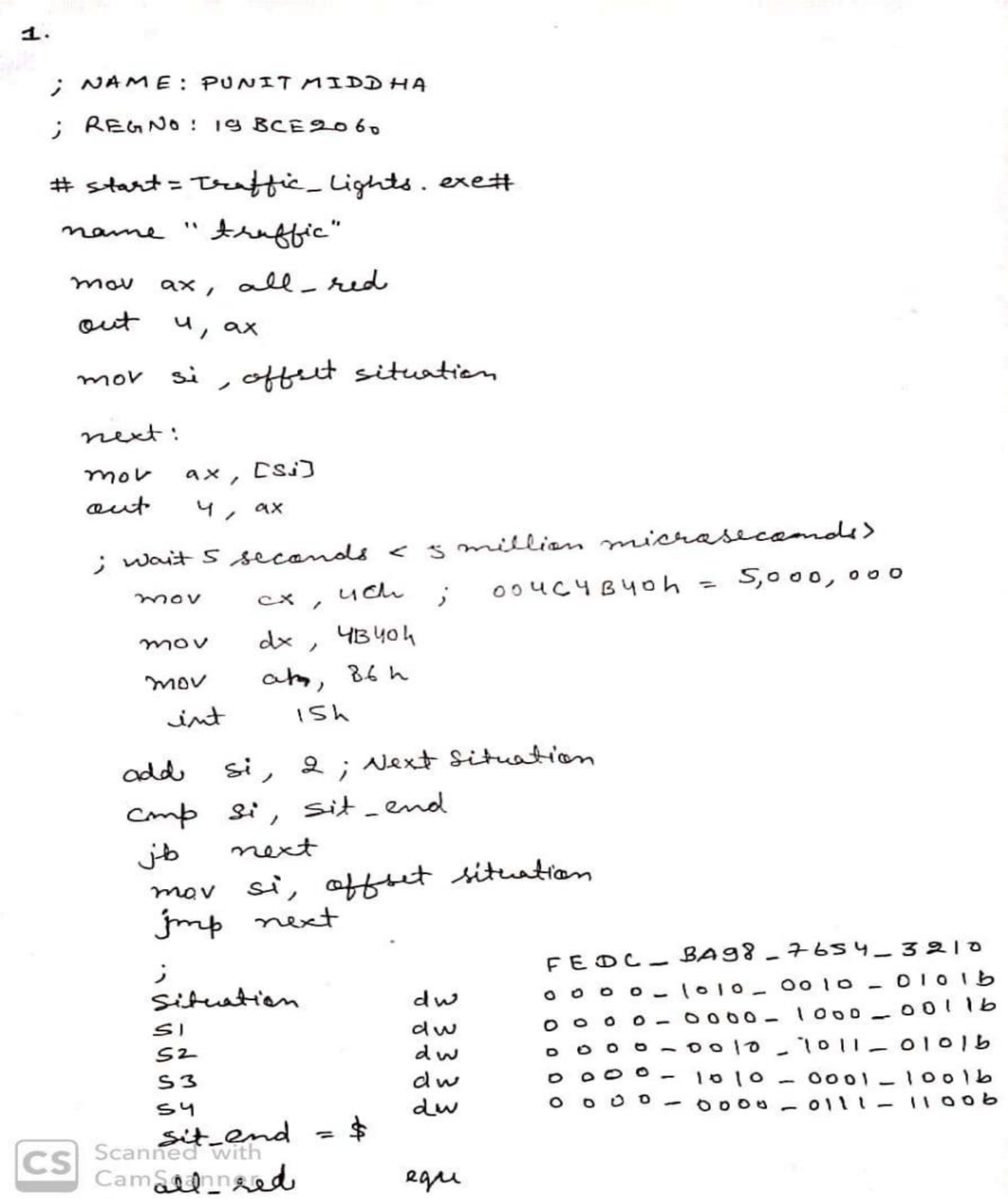
DATE: 24/11/2021

1. Traffic Light Control System

Aim:

To load different control words in traffic light system in emu8086

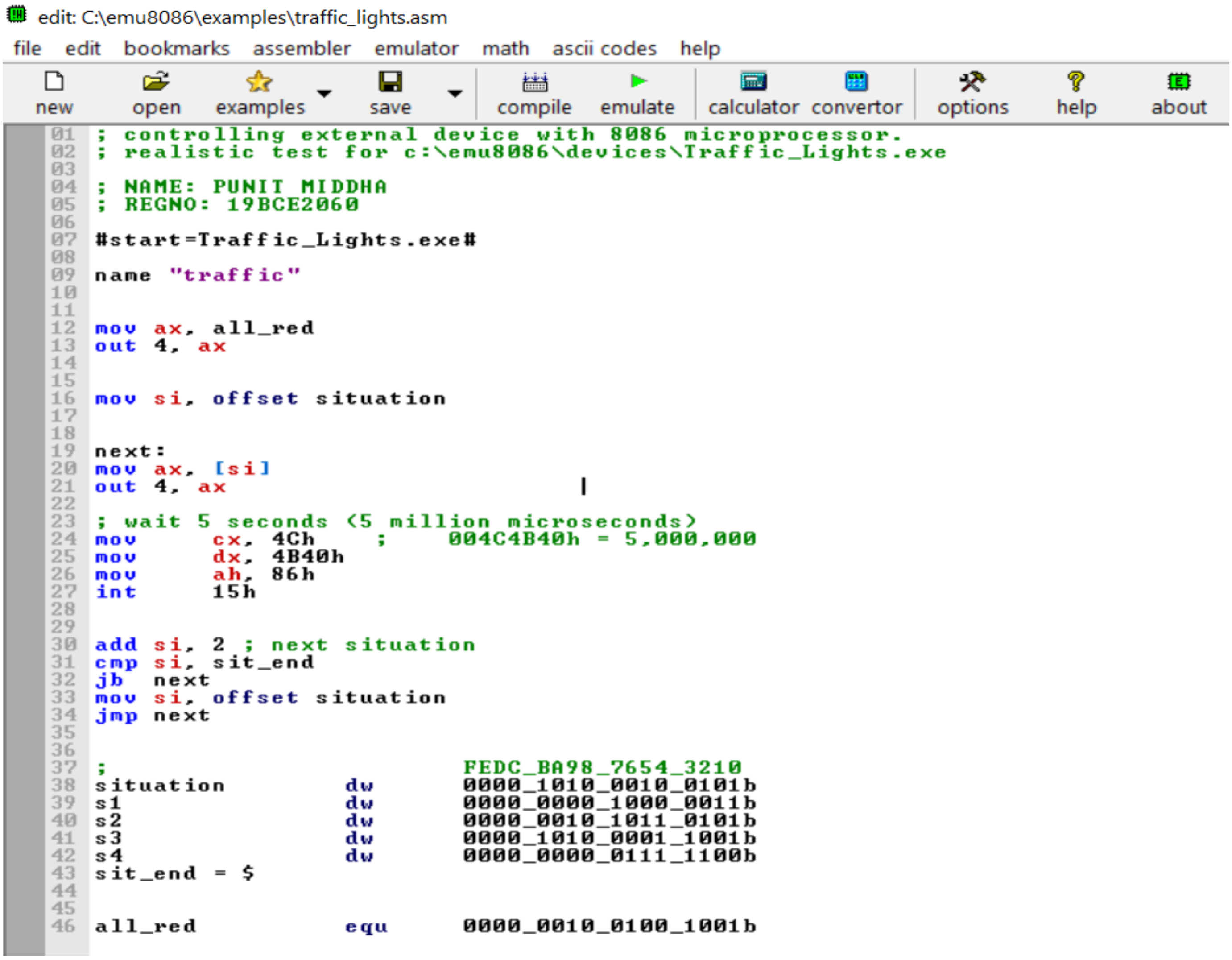
Handwritten Program:



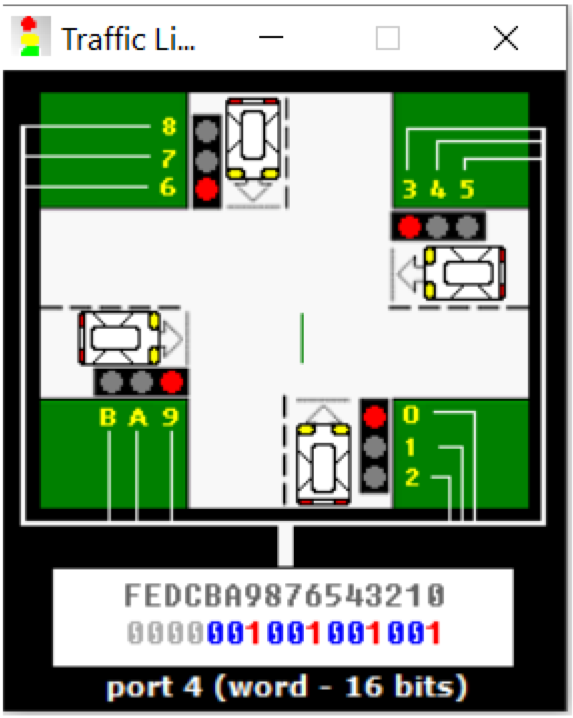
1.

```
; NAME: PUNIT MIDDHA
; REGNO: 19BCE2060

#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 5 seconds < 5 million microseconds>
   mov cx, 4Ch ; 004C4B40h = 5,000,000
   mov dx, 4B40h
   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_1010_0010_0101b
  S1                 dw                  0000_0000_1000_0011b
  S2                 dw                  0000_0010_1011_0101b
  S3                 dw                  0000_1010_0001_1001b
  S4                 dw                  0000_0000_0111_1100b
  sit_end = $
  all_red            equ
```

Snapshots of typed program and Output:

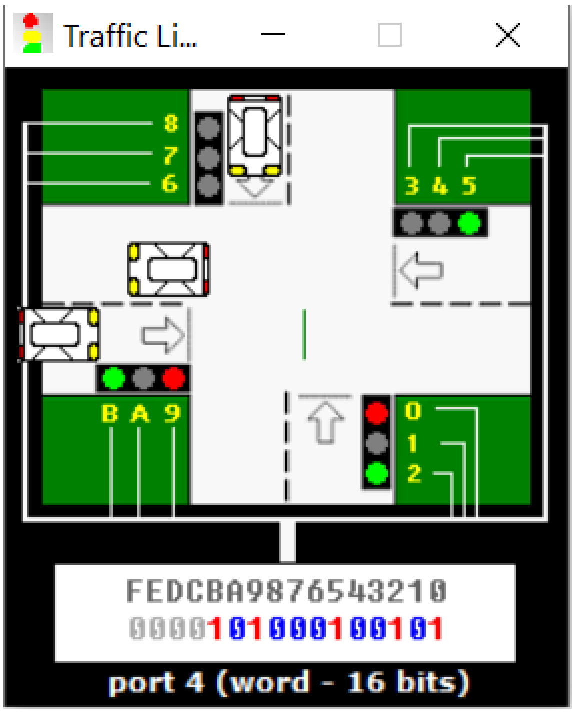


```
; controlling external device with 8086 microprocessor.
; realistic test for c:\emu8086\devices\Traffic_Lights.exe

; NAME: PUNIT MIDDHA
; REGNO: 19BCE2060

#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 5 seconds (5 million microseconds)
mov     cx, 4Ch    ;    004C4B40h = 5,000,000
mov     dx, 4B40h
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_1010_0010_0101b
s1                  dw      0000_0000_1000_0011b
s2                  dw      0000_0010_1011_0101b
s3                  dw      0000_1010_0001_1001b
s4                  dw      0000_0000_0111_1100b
sit_end = $


all_red             equ     0000_0010_0100_1001b
```

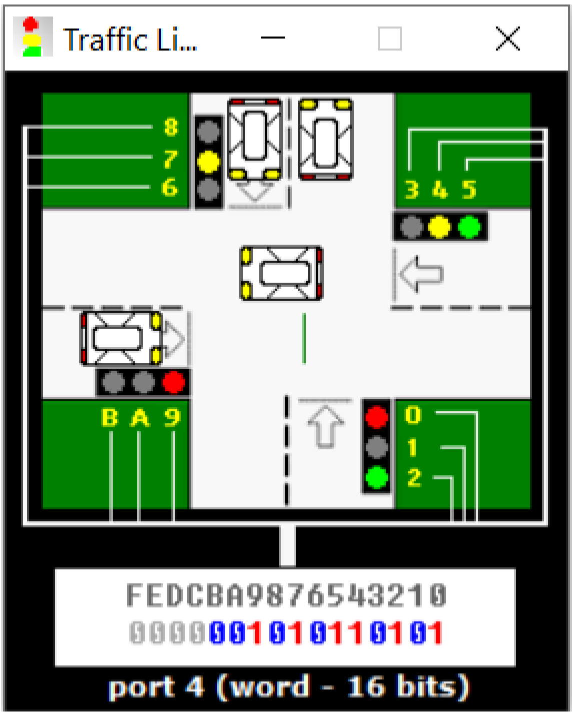


```
FEDCBA9876543210
0000001001001001
```

port 4 (word - 16 bits)



```
FEDCBA9876543210
0000101000100101
port 4 (word - 16 bits)
```



```
FEDCBA9876543210
0000001010110101
port 4 (word - 16 bits)
```

Inference:

- Output 1:
  In the first output i.e., all red, the control word is:
  0000001001001001
  001 means red light.

So, we can see in the output red light on in all traffic lights.

- Output 2:
  In the second output the control word is:
  0000101000100101
  101 means red and green light are on
  000 means no light is on
  100 means red light is on
  101 means red and green lights are on
  This can be seen in the output.

- Output 3:
  In the last output the control word is:
  0000001010110101
  001 means red light is on
  010 means yellow light is on
  110 means green and yellow lights are on
  101 means red and green lights are on
  This can be seen in the output.