

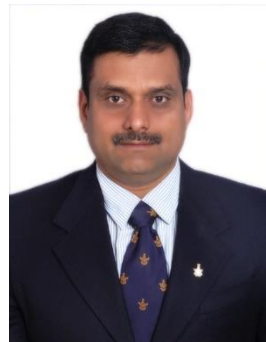


Microprocessors and Interfaces: 2021-22

Lab 10

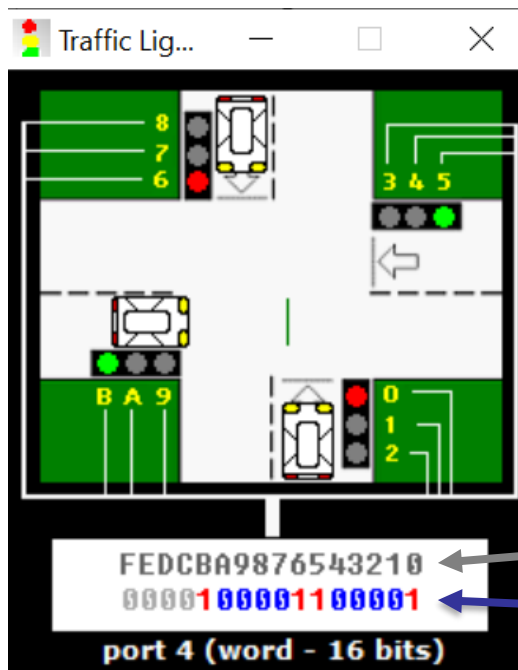
Program to Control Traffic Light System

By Dr. Sanjay Vidhyadharan



Traffic Light in Emulator

- A virtual traffic light system is available in EMU8086 with port address 4.
- It consists of 12 LEDs with an animation of car moving in direction where green LED is activated.
- Control word can be provided using 8086 to change the signal color.



State of each LED

1: LED is turned on
0: LED is turned off

Control Word Format

Current Control Word

Control Words

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
x	x	x	x	1	0	0	0	0	1	1	0	0	0	0	1
x	x	x	x	0	0	1	0	0	1	0	0	1	0	0	1
x	x	x	x	0	0	1	1	0	0	0	0	1	1	0	0
x	x	x	x	0	1	1	0	1	0	0	1	1	0	1	0
x	x	x	x	1	0	0	0	0	1	1	0	0	0	0	1
x	x	x	x	1	0	0	0	0	1	1	0	0	0	0	1
x	x	x	x	0	1	0	0	1	1	0	1	0	0	1	1

Default

All red

x: Not used bits (should be replaced by 0).

Traffic Light in Emulator

- Port Address: 04H
- Instruction to include stepper motor in emulator:
`#start=traffic_lights.exe#`

- **How to load the control word in traffic light?**

Use OUT instruction. For out instruction select port 4.
Use AX register to store control word intermediately.

- **How to maintain time gap between two control words?**

Use int 15h, AH=86h for delay

int 15h, AH=86h

INT 15h / AH = 86h - BIOS wait function. *input:*
CX:DX = interval in microseconds *return:*
CF clear if successful (wait interval elapsed),
CF set on error or when wait function is already in progress.

- What should be the value of CX-DX register for wait time of 2s between two control instructions?

Wait Time = 2 s = $2 \times 10^6 \mu\text{s}$
Hexadecimal Value = 1E8480
CX = 001Eh
DX = 8480h

ALP for Traffic Signal

```
#start=Traffic_Lights.exe#
```

Initializes the virtual traffic signal

```
name "traffic"
```

```
mov ax, all_red  
out 4, ax
```

Closes all traffic

```
mov si, offset situation  
next:  mov ax, [si]  
       out 4, ax
```

Offset calculation of CW and load in SI.

Loads the first data in AX and send it IO

```
mov cx, xxxxh
```

Use CX-DX register to provide wait instruction for 5s.

```
mov dx, xxxxh
```

```
mov ah, 86h
```

BIOS Delay Function. Unit in μ s. Content Format: CX-DX

```
int 15h
```

```
add si, 2
```

Increase SI for next data

```
cmp si, sit_end
```

```
jb next
```

```
mov si, offset situation
```

```
jmp next
```

```
situation dw xxxx_xxxx_xxxx_xxxxh
```

```
s1 dw xxxx_xxxx_xxxx_xxxxh
```

```
s2 dw xxxx_xxxx_xxxx_xxxxh
```

```
s3 dw xxxx_xxxx_xxxx_xxxxh
```

```
s4 dw xxxx_xxxx_xxxx_xxxxh
```

Control world (16-bit format)

```
sit_end = $
```

```
all_red equ 0000_0010_0100_1001b
```

Thankyou

Sanjay Vidnyadhharan