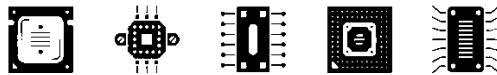


CSE 2006

MICROPROCESSOR AND INTERFACING



Lab FAT Viva

L11+L12 | SJT516

FALL SEMESTER 2021-22

by

SHARADINDU ADHIKARI

19BCE2105

VIVA QUESTION 5 MARKS (Marks will be added in LabFat)

Due tomorrow at 11:59 PM

Points
No points

Instructions

Write Program to Control Traffic Light System (ALP)

My work

 Attach  New

Aim: To write an assembly language program to control traffic light system

Algorithm:

1. First, we initialize the virtual traffic signal
2. Then using `mov x, all_red`, we close all traffic
3. Followed by Offset calculation of CW and load in SI
4. Then we load the first data in AX and sent it IO
5. After that, we use CX-DX register to provide wait instruction for 5s
6. Then it's the time for BIOS Delay Function. Its unit is in μ s. And content format: CX-DX
7. We gotta increase SI for next data now
8. And check if all situational data are emulated or not. If yes, then restart. Else, complete all the situational control words
9. Finally enter the Control world (in 16-bit format)

Code:

```
; To write an assembly language program to control traffic light system  
; by Sharadindu Adhikari, 19BCE2105
```

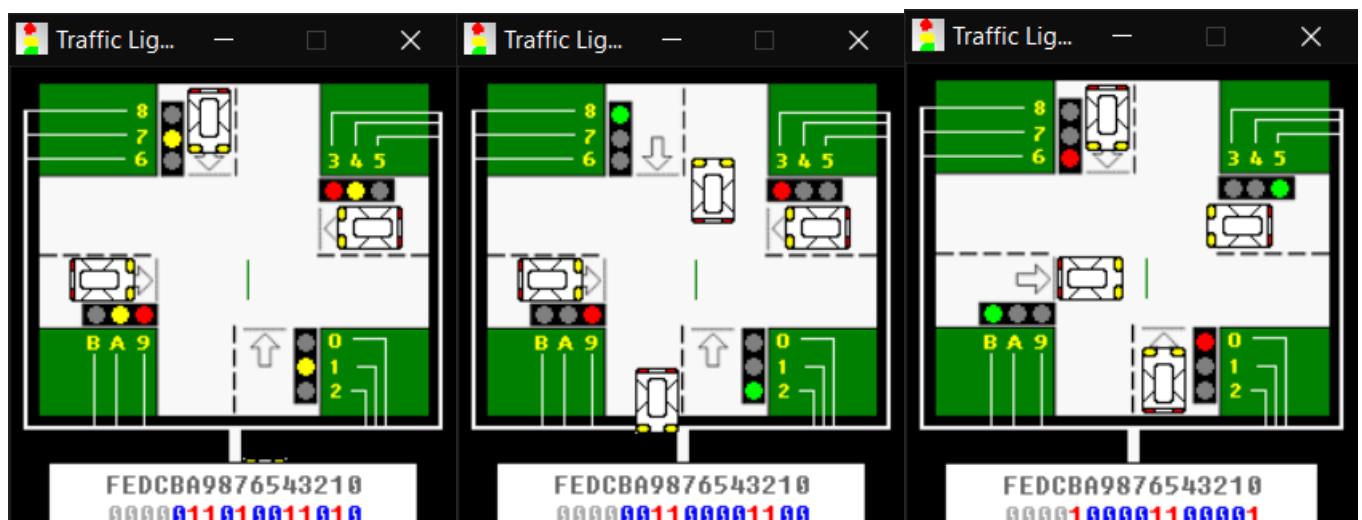
```
#start=Traffic_Lights.exe#  
name "traffic"  
mov ax, all_red  
out 4, ax  
mov si, offset situation  
next: mov ax, [si]  
out 4, ax  
mov cx, 004Ch  
mov dx, 4B40h  
mov ah, 86h  
int 15h  
add si, 2  
cmp si, sit_end  
jb next  
mov si, offset situation  
jmp next  
situation dw 100001100001b  
s1 dw 001100001100b
```

```
s2 dw 011010011010b
s3 dw 100001100001b
s4 dw 010011010011b
sit_end = $
all_red equ 0000_0010_0100_1001b
```

Screenshot:

```
edit: C:\emu8086\MySource\LabFATViva.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
; To write an assembly language program to control traffic light system
; by Sharadindu Adhikari, 19BCE2105
01
02
03
04
05
06
07 #start=Traffic_Lights.exe#
08 name "traffic"
09 mov ax, all_red
10 out 4, ax
11 mov si, offset situation
12 next: mov ax, [si]
13 out 4, ax
14 mov cx, 004Ch
15 mov dx, 4B40h
16 mov ah, 86h
17 int 15h
18 add si, 2
19 cmp si, sit_end
20 jb next
21 mov si, offset situation
22 jmp next
23 situation dw 100001100001b
24 s1 dw 001100001100b
25 s2 dw 011010011010b
26 s3 dw 100001100001b
27 s4 dw 010011010011b
28 sit_end = $
29 all_red equ 0000_0010_0100_1001b
```

Output:



Windows desktop environment showing a traffic light simulator and its assembly code.

Traffic Lig... (Simulator):

- Port 4 (word - 16 bits): FEDCBA9876543210
- Assembly code:


```

18 add si, 2
19 cmp si, sit_end
20 jb next
21 mov si, offset situation
22 jmp next
23 situation dw 100001100001b
      
```
- Registers:

Register	Value
AX	86 9A
BX	00 00
CX	00 4C
DX	4B 4B
SI	F400
DI	0100
ES	0100
- Flags:

Flag	Value
CF	1
ZF	0
SF	1
OF	0
PF	1
AF	1
F	0
DF	0
- Stack:

Address	Value
0100:FFFE	0297
0100:FFFC	0100
0100:FFFA	0100
0100:FFF8	0000
0100:FFF6	0000
0100:FFF4	0000
0100:FFF2	0000
0100:FFF0	0000
0100:FFEE	0000
0100:FFEC	0000
0100:FFEA	0000
0100:FFE8	0000
0100:FFE6	0000
0100:FFE4	0000
0100:FFE2	0000
0100:FFE0	0000
0100:FFDE	0000
0100:FFDC	0000
0100:FFDA	0000
0100:FFD8	0000
0100:FFD6	0000
0100:FFD4	0000
0100:FFD2	0000
0100:FFD0	0000
0100:FFCE	0000
0100:FFCC	0000
0100:FFCA	0000

original source code:

```

01 #start=Traffic_Lights.exe#
02 name "Traffic"
03 mov ax, all_red
04 out 4, ax
05 mov si, offset situation
06 next: mov ax, [si]
07 out 4, ax
08 mov cx, 004Ch
09 mov dx, 4B40h
10 mov ah, 86h
11 int 15h
12 add si, 2
13 cmp si, sit_end
14 jb next
15 mov si, offset situation
16 jmp next
17 situation dw 100001100001b
18 si dw 001100001100b
19 s2 dw 011010011010b
20 s3 dw 100001100001b
21 s4 dw 010011010011b
22 sit_end = $
23 all_red equ 0000_0010_0100_1001b
24
25
      
```

Windows desktop environment showing a traffic light simulator and its assembly code.

Traffic Lig... (Simulator):

- Port 4 (word - 16 bits): FEDCBA9876543210
- Assembly code:


```

18 add si, 2
19 cmp si, sit_end
20 jb next
21 mov si, offset situation
22 jmp next
23 situation dw 100001100001b
      
```
- Registers:

Register	Value
AX	86 0C
BX	00 00
CX	00 4C
DX	4B 4B
SI	F400
DI	0100
ES	0100
- Flags:

Flag	Value
CF	1
ZF	0
SF	1
OF	0
PF	1
AF	1
F	0
DF	0
- Stack:

Address	Value
0100:FFFE	0293
0100:FFFC	0100
0100:FFFA	0100
0100:FFF8	0000
0100:FFF6	0000
0100:FFF4	0000
0100:FFF2	0000
0100:FFF0	0000
0100:FFEE	0000
0100:FFEC	0000
0100:FFEA	0000
0100:FFE8	0000
0100:FFE6	0000
0100:FFE4	0000
0100:FFE2	0000
0100:FFE0	0000
0100:FFDE	0000
0100:FFDC	0000
0100:FFDA	0000
0100:FFD8	0000
0100:FFD6	0000
0100:FFD4	0000
0100:FFD2	0000
0100:FFD0	0000
0100:FFCE	0000
0100:FFCC	0000
0100:FFCA	0000

original source code:

```

01 #start=Traffic_Lights.exe#
02 name "Traffic"
03 mov ax, all_red
04 out 4, ax
05 mov si, offset situation
06 next: mov ax, [si]
07 out 4, ax
08 mov cx, 004Ch
09 mov dx, 4B40h
10 mov ah, 86h
11 int 15h
12 add si, 2
13 cmp si, sit_end
14 jb next
15 mov si, offset situation
16 jmp next
17 situation dw 100001100001b
18 si dw 001100001100b
19 s2 dw 011010011010b
20 s3 dw 100001100001b
21 s4 dw 010011010011b
22 sit_end = $
23 all_red equ 0000_0010_0100_1001b
24
25
      
```