



西北工业大学

EXPERIMENT REPORT OF ASSEMBLY LANGUAGE

Project 2- Traffic Light Control

NAME : ABID ALI

STUDENT ID :2019380141

DATE : 07/10/2021

SUBMITTED TO :PROFESSOR Yin LU

Problem Description:

We will try to simulate a traffic light control system. The hardware schematic design is given to us. The 8254.Timer0 is used to provide timing function, and the incoming clock rate is 1KHz. When time is up, the output signal out1 will be caught by 8259.IR0 and send to 8086 as a interrupt request. In the interrupt service routine, traffic light is changed.

The PIT 8259 is set to be working single piece, normal EOI, edge triggered, none buffered mode, and the interrupt number for IR0 is 40H.

It is desired that the red light lights for 45 seconds, yellow lights for 5 seconds, and green lights for 60 seconds.

The template given is given below:

.MODEL SMALL

.STACK 32

.DATA

PORT_LIGHT EQU 68H

PORT_8254 EQU 048H

PORT_8259LOW EQU 60H

PORT_8259HIGH EQU 62H

IR1_INT_NUMBER EQU 41H

```

        PATTERN_CODE DB 01H

.CODE
MAIN PROC FAR

        MOV    AX, @DATA
        MOV    DS, AX
        CLI

;TODO1: regist ISR
        PUSH DS

; retister ISR for IR1
        POP DS

;TODO2: initialize 8259 ICW1,2,4, OCW1
        ;

;TODO3: de-light all trafic lights
        ;

;TODO4: initialize 8254
        ;

; and wait for interrupt request comes

FIX_BUG:
        MOV DX,60H
        MOV AL,41H           ;dump 41H, which is the int number, to data bus
        OUT DX,AL
        ;TODO5: enable interrupt, start to work
        STI
        JMP FIX_BUG
        ;quit to DOS
        MOV AX, 4C00H
        INT 21H

MAIN ENDP

;;=====

;SubrOUTine: MY_ISR
MY_ISR    PROC FAR

        CLI                 ;close interrupt service

```

;TODO1: change the traffic light

;TODO2: reset 8254.Timer0 by sending initial value

;TODO3: save next pattern word

MOV pattern_code, AL

;TODO4: send OCW2 which will be regarded as EOI command

;TODO5: open interrupter service and return back from ISR

MY_ISR ENDP

END MAIN

Goal:

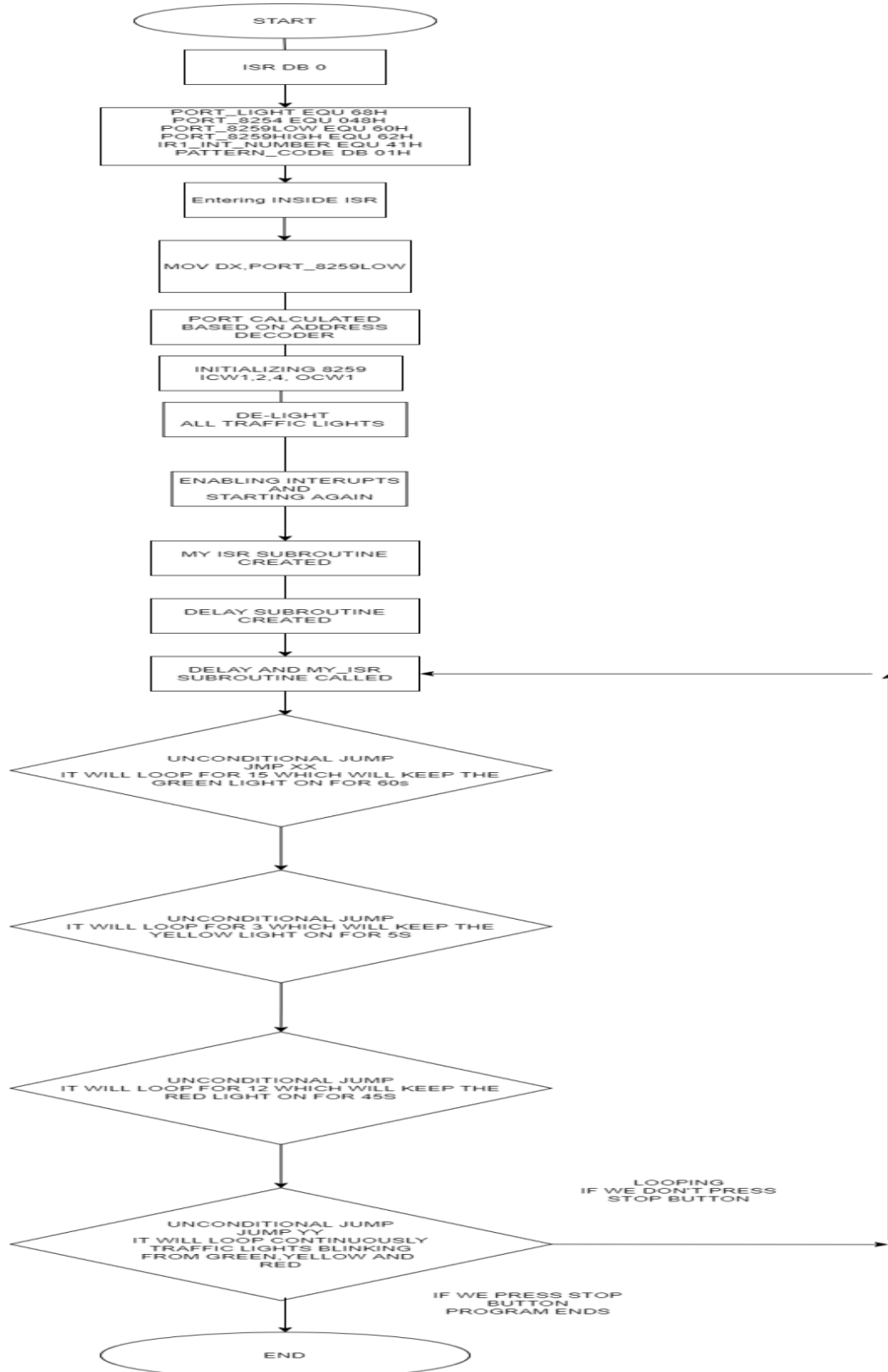
We are given 3 LED lights(Green,Red,Blue).Those are lights of traffic lights.

First, Red LED light will blink for 45 seconds.

Second ,Yellow LED light will blink for 5 seconds.

Third ,Green LED light will blink for 60 seconds.

Flow-chart:



Code:

=====

;Description: Program of Project 2 Traffic Lights

; It is one of the optional project designed for

; the online final examination of spring semester 2021.

;Name:ABID ALI ID:2019380141

;Date:07/10/2021

=====

;This is the program for Project 2 Traffic Lights

; making use of a 8254 and 8259 to control traffic lights

; red = 45s, yellow = 5s, green = 60s;

; and incoming clock = 1KHz

.MODEL SMALL

.STACK 32

.DATA

ISR DB 0

PORT_LIGHT EQU 68H

PORT_8254 EQU 048H

PORT_8259LOW EQU 60H

PORT_8259HIGH EQU 62H

IR1_INT_NUMBER EQU 41H

PATTERN_CODE DB 01H

.CODE

MAIN PROC FAR

;THIS IS THE PROGRAM ENTRY POINT

MOV AX, @DATA

;LOAD THE DATA SEGMENT ADDRESS

MOV DS, AX

;ASSIGN VALUE TO DATA SEGMENT

REGISTER

CLI

;TODO1: regist ISR

PUSH DS

MOV AX,0

MOV DS,AX

MOV SI,40h*4

MOV BX,OFFSET ISR

MOV BX,[SI]

MOV BX,SEG ISR

MOV BX,[SI+2]

; register ISR for IR1

POP DS

;TODO2: initialize 8259 ICW1,2,4, OCW1

MOV DX,PORT_8259LOW

MOV AL,00010011B ;ICW 1

OUT DX,AL

JMP YY

YY:

JMP XX

XX:

MOV DX,PORT_LIGHT ;CONTROL GREEN LIGHT PORT

MOV AL,04H ;Initial count 4 sent to counter A, every 5 Times, send 4

OUT DX,AL

;CREATING A LOOP FOR GREEN LIGHT.SO,THAT IT WORKS IN CONTINUOUS
LIGHT

;LOOPING UNTIL 60SECONDS THEN BREAKS THE LOOP

loopyA:loop loopyA

loopyB:loop loopyB

loopyC:loop loopyC

loopyD:loop loopyD

loopyE:loop loopyE

loopyF:loop loopyF

loopyG:loop loopyG

loopyH:loop loopyH

loopyI:loop loopyI

loopyJ:loop loopyJ

loopyK:loop loopyK

loopyL:loop loopyL

loopyM:loop loopyM

loopyN:loop loopyN

loopyO:loop loopyO

loopyQ:loop loopyQ

loopyP:loop loopyP

loopyR:loop loopyR

loopyS:loop loopyS

loopyT:loop loopyT

loopyU:loop loopyU

loopyV:loop loopyV

loopyW:loop loopyW

loopyX:loop loopyX

```
loopyY:loop loopyY
loopyZ:loop loopyZ
loopyA1:loop loopyA1
loopyB1:loop loopyB1
loopyC1:loop loopyC1
loopyD1:loop loopyD1
loopyE1:loop loopyE1
loopyF1:loop loopyF1
loopyG1:loop loopyG1
loopyH1:loop loopyH1
loopyI1:loop loopyI1
loopyJ1:loop loopyJ1
loopyK1:loop loopyK1
loopyL1:loop loopyL1
loopyM1:loop loopyM1
loopyN1:loop loopyN1
loopyO1:loop loopyO1
;JMP XX
```

```
MOV DX,PORT_8259HIGH
```

```
MOV AL,40H ;ICW 2
```

```
OUT DX,AL
```


CALL DELAY

MOV AL,00000001B ;ICW 4

OUT DX,AL

CALL DELAY

OCW1 EQU 200H

;TODO3: de-light all traffic lights

MOV DX,PORT_8254

MOV AL, PATTERN_CODE

OUT DX,AL

;TODO4: initialize 8254

MOV DX,PORT_8254 ;CONTROL ALL LIGHT PORT

MOV AL,40H ;Mode control word 00010000

OUT DX,AL

MOV DX,PORT_LIGHT ;CONTROL GREEN LIGHT PORT

MOV AL,04H ;Initial count 4 sent to counter A, every5Times,
send4

OUT DX,AL

CALL DELAY

```
CALL DELAY                ;CONTROL YELLOW LIGHT
MOV AL,10001010B          ;ICW 4
OUT DX,AL
CALL DELAY
;CREATING A LOOP FOR YELLOW LIGHT.SO,THAT IT WORKS IN CONTINUOUS
LOOP
```

```
;LOOPING UNTIL 5 SECONDS THEN BREAKS THE LOOP
```

```
    loopy11:loop loopy11
    loopy61:loop loopy61
    loopy71:loop loopy71
    loopy81:loop loopy81
    CALL DELAY
MOV AL,10001010B          ;ICW 4
OUT DX,AL
CALL DELAY
```

```
    CALL DELAY                ;CONTROL YELLOW LIGHT
MOV AL,10000001B          ;ICW 4
OUT DX,AL
CALL DELAY
    loopy9:loop loopy9
```

```

MOV DX,PORT_LIGHT          ;CONTROL GREEN LIGHT
MOV AL,04H                  ;Initial count 4 sent to counter A, every5Times,
send4
OUT DX,AL
CALL DELAY

```

```

CALL DELAY                  ;Red light
MOV AL,01000001B           ;ICW 4
OUT DX,AL
CALL DELAY
;CREATING A LOOP FOR RED LIGHT.SO,THAT IT WORKS IN CONTINUOUS
LIGHT

```

```

;LOOPING UNTIL 45 SECONDS THEN BREAKS THE LOOP

```

```

    loopy1A:loop loopy1A

```

```

    loopy2A:loop loopy2A

```

```

    loopy3A:loop loopy3A

```

```

    loopy4A:loop loopy4A

```

```

    loopy5A:loop loopy5A

```

```

    loopy6A:loop loopy6A

```

```

    loopy7A:loop loopy7A

```

```

    loopy8A:loop loopy8A

```

```

    loopy9A:loop loopy9A

```

```

    loopy10A:loop loopy10A

```

```

    loopy11A:loop loopy11A

```

```

    loopy12A:loop loopy12A

```

```

    loopy13A:loop loopy13A

```

```

    loopy14A:loop loopy14A

```

```

    loopy15A:loop loopy15A

```

```
loopy16A:loop loopy16A
loopy17A:loop loopy17A
loopy18A:loop loopy18A
loopy19A:loop loopy19A
loopy20A:loop loopy20A
loopy21A:loop loopy21A
loopy22A:loop loopy22A
loopy23A:loop loopy23A
loopy24A:loop loopy24A
loopy25A:loop loopy25A
loopy26A:loop loopy26A
loopy27A:loop loopy27A
loopy28A:loop loopy28A
loopy29A:loop loopy29A
loopy30A:loop loopy30A
```

```
CALL DELAY
MOV AL,10000100B
OUT DX,AL
CALL DELAY
```

```
MOV DX,PORT_LIGHT          ;CONTROL GREEN LIGHT
MOV AL,04H                  ;Initial count 4 sent to counter A, every5Times,
send4
```

OUT DX,AL

CALL DELAY

;CALL DELAY SUBPROCEDURE

loopy12:loop loopy12

MOV DX,PORT_LIGHT

;control green light

MOV AL,04H

;Initial count 4 sent to counter A, every5Times,

send4

OUT DX,AL

CALL DELAY

;CALL DELAY SUBPROCEDURE

JMP YY

; and wait for interrupt request comes

STI

FIX_BUG:

MOV DX,60H

MOV AL,41H

;dump 41H, which is the int number, to

data bus

OUT DX,AL

;TODO5: enable interrupt, start to work

XOR AX, AX

MOV ES, AX

;first interrupt for timer

MOV AL, 40H

MOV AH, 4

MUL AH

MOV BX, AX

LEA AX, MY_ISR

MOV WORD PTR ES:[BX], AX

MOV AX, CS

MOV WORD PTR ES:[BX+2], AX

XOR AX, AX

MOV AL, 08H

MOV BX, AX

LEA AX, MY_ISR

MOV WORD PTR ES:[BX], AX

MOV AX, CS

MOV WORD PTR ES:[BX+2], AX

MOV AL, PORT_8259HIGH

MOV DX, OCW1

OUT DX, AL

STI

```

        JMP FIX_BUG

        ;quit to DOS

        MOV AX, 4C00H

        INT 21H

MAIN ENDP

;;=====

;SubrOUTine: MY_ISR

MY_ISR    PROC FAR

        CLI                                ;close interrupt service

        ;The flag may be set or cleared using the CLI (Clear Interrupts)

;TODO1: change the traffic light

        PUSH AX                            ;PUSH AX FROM THE STACK

        PUSH DX                            ;PUSH DX FROM THE STACK

        MOV AL,pattern_code                ;pattern_code are put on AL register

        ROL AL,1

        MOV DX,PORT_LIGHT;

        OUT DX,AL

;TODO2: reset 8254.Timer0 by sending initial value

        MOV DX,PORT_8254                    ;control all light

        MOV AL,40H                          ;Mode control word 00010000

        OUT DX,AL

;TODO3: save next pattern word

        MOV AL,10000100B                    ;ICW 4

        OUT DX,AL

;TODO4: send OCW2 which will be regarded as EOI command

        OUT DX, AL                          ;WRITE OCW2

```

MOV AL, 6AH

;TODO5: open interrupter service and return back from ISR

MOV pattern_code,AL

MOV DX,PORT_8259LOW

MOV AL,20H

OUT DX,AL

POP DX ;POP DX FROM THE STACK

POP AX ;POP AX FROM THE STACK

sti ;The STI instruction sets the IF flag, but interrupts are not checked
for until after the next instruction which in this case would be the CLI which takes effect
immediately.

IRET

MY_ISR ENDP

;;=====

;SUBROUTINE: DELAY

;DELAY CAN BE SEEN FOR FEW SECOND

DELAY PROC NEAR

PUSH BX ;PUSH BX IN THE STACK

PUSH CX ;PUSH CX IN THE STACK

MOV BX,0AH

Loop_OUT: MOV CX, 03H ;LABEL OF OUTERLOOP

Loop_Inner: LOOP Loop_Inner ;LABEL OF INNERLOOP

DEC BX

JNZ Loop_OUT

POP CX ;POP CX FROM THE STACK


```

                POP BX                ;POP BX FROM THE STACK
                RET
DELAY          ENDP

                ;THIS IS THE PROGRAM EXIT POINT

END MAIN

```

Debugging:

This assembly language and also IDE for this program was very new to me. I gradually become familiar with them. At first, I was making silly mistakes like syntax error and illegal instructions used in my program. I couldn't correctly put the port address in code. After a lot of trial and error, I finally managed to get the idea and that idea was implemented. In this way, I solved few problems. When I have any problem, I discussed with my classmates. Do they know how to solve that problem if they couldn't solve then I posted questions in different forms. After reading different articles, videos, teacher lecture notes, I managed to solve it.

Attachment:

- 1) TrafficLight.docx
- 2) TrafficLight.pdf
- 3) TrafficLight.asm
- 4) TrafficLights(Flowchart).png
- 5) TrafficLight.mkv

Acknowledgement:

I complete this assignment by myself by using online videos and taking help from online resources. The most useful help was for me, that was my previous practical experiments. The experience I gained through those practicals, I implemented in this practical.

This project was very challenging for me. I had great fun solving this problem. It took me 10-12 days to solve this problem. I feel delighted finally to solve this problem.

I learned through this project if I discuss with some classmates or look some articles,video.It's possible to solve different complex problem.This experience will help me in future in solving real world complex problems.