

ENGR 270 Lab #3 Online – Finite State Machine

Objectives

Apply PICmicro Assembly language, standard header files and processor to implement Finite State Machines.

Preparation

Complete the following steps before starting to work on the experiments in this lab:

- 1) Complete Lab2 and associated report.
- 2) Review Chapter 2 & 3.
- 3) Instead of define each of Special Function Registers (SFRs) such as PORTA and PORTB, the following header that contains all the definition can be included:

```
#include p18f1220.inc ;header file
```

For this lab, you need to use a delay function. The following code provides an example of a delay function using the default RC clock oscillator ($T_{osc} = 32 \text{ usec}$) which results in Instruction Cycle ($T_{cycle} = 4 * T_{osc} = 128 \text{ usec}$).

```

-----
; FILE: main.asm
; DESC: Delay Function Example
; DATE: 6-28-20
; AUTH: Class
; DEVICE: PICmicro (PIC18f1220)
-----

list    p=18f1220    ; Set processor type
radix   hex          ; Sets the default radix for data exp.
; Disable Watchdog timer, Low V. Prog, and RA6 as a clock
config  WDT=OFF, LVP=OFF, OSC = INTIO2

#include  p18f1220.inc    ;header file

#define  dCount    0x080    ; Delay count register (GPR)

org 0x00    ; after reset the next line of code is executed
main:
; initialize all I/O ports
CLRF    PORTA    ; Initialize PORTA
CLRF    PORTB    ; Initialize PORTB
MOVLW   0x7F
MOVWF   ADCON1    ; Configure all I/O Ports as digital
BCF     TRISB,0    ; Configure RB0 to output

loop:   ; Approx. every 2 second toggle RB0
BTG     PORTB,0
MOVLW   .20
MOVWF   dCount    ; set dCount to 20
CALL    delay    ; delay for approximately 20 *1/10 = 2 seconds
BRA     loop

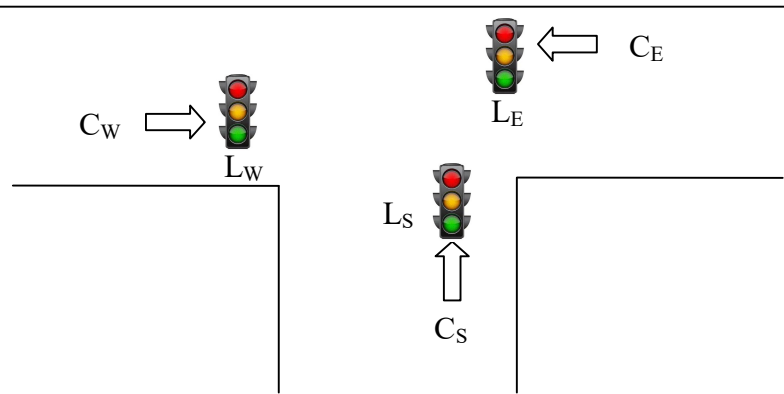
; Delay function waits for (dcount/10) seconds
; Assuming PICmicro is set to default RC Clock Oscillator
delay:
MOVLW   .195    ; WREG = 195
; requires 195*4*128=99840 usec to complete the loop or Approx. 0.1 sec.
interDelay:
DECF    WREG          ; 1 Cycle
NOP     ; 1 Cycle
BNZ     interDelay    ; 2 Cycles except the last time
DECF    dCount,1
BNZ     delay
RETURN

end      ; program end

```

Experiment 1. 3-way Traffic Light Controller

A three way intersection is in need of a traffic signal control system and you have been assigned the task to design and implement the system using PICmicro assembly. Each direction has only one lane and one set of signal control lights (green, yellow and red). Your design should only allow one go or green light at a time. Further, all transitions from green to red should go through an intermediate 2-4 seconds yellow light. Complete the following steps for this design:



Your PICmicro accept Car presence indicators (C_W , C_S and C_E) from RA_0 , RA_1 , RA_2 respectively. The output will be directed according to the following:

L_E and output will be stored in $PORTB$ using the following coding to indicate light status:

Output Code	Green	Yellow	Red
L_W	RB0	RB1	RB2
L_S	RB3	RB4	RB5
L_E	RB6	RB7	RA7

Report Requirements

This lab and associated report must be completed individually. All reports must be computer printed (formulas and diagrams may be hand drawn) and at minimum:

For each experiment include:

- Clear problem statement in your words.
- Answer to any specific experiment questions (if any)
- Pseudo code which may be written in C-like syntax
- Disassembled code available after successful assembling
- Test plan which describes the input values and expect output/memory values for a successful design.
- Simulator output which shows the stimuli, relevant memory locations values showing validation based on test plan.

For the whole report include:

- A Cover sheet with your name, class, lab and completion date.
- A Lessons Learned section which summarizes your learning from this lab.
- A New Experiment section that has description of a new experiment and the experiment's results. Experiment should be related to material covered in class but not similar to one of the experiments in this lab.