# Lab 4: Macros, Context Saving, Pulse Width Modulation (PWM), The Capture/Compare/PWM (CCP) Module

**Name:**\_\_\_\_\_REZA SHISHEIE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ID\_\_\_\_2708062\_\_\_\_\_\_\_\_\_\_  
  
**Give brief answers to the following questions. You can edit this document and insert your answers after each question.**

**Due dates:**

**MW – Wed, Feb 28, beginning of class  
TTH – Tue, Feb 27, beginning of class**

**Circle one: MW or TTH**

1. (1 pt) What are the three RAM addresses that are shared (common) with 0x7A?  
     
   **Ans.** ????

0x80 = decimal:128

Bank0 🡪 0x7A (decimal: 122)

Bank0 🡪 0x7A + 0x80 = 0xFA (decimal: 250)

Bank1 🡪 0x7A + 0x80 + 0x80 = 0x17A (decimal: 378)

Bank2 🡪 0x7A + 0x80 + 0x80 + 0x80 = 0x1FA (decimal: 506)

1. (1 pt) What are the contents of W (in decimal and hex) after the following instruction sequence?  
     
   movlw 2  
   sublw 1  
     
   **Ans.**

2-1 =1

1. (2 pts)Write assembly code that performs the following function:

If PORTC ≠ 0, then write 100 (decimal) to PORTC

Assume that bank 0 is already selected. Use no more than four instructions. Don’t forget to comment your code.  
  
**Ans.**

If Line 3 is skipped the same value of PORTC is replaced into PORTC

Movf PORTC ; MOVF PORTC to W which changes Z  
btfss STATUS, Z ; if Z=1 (PORTC=0), skip next line

movlw D'100' ; set W to 100,

movwf PORTC ; PORTC=W=100

1. (2 pts)What are the contents of W (in binary) after the following instruction sequence?  
     
   movlw 2  
   movwf PORTC  
   comf PORTC, W  
   swapf PORTC, F  
     
   **Ans.**movlw 2 ; **W = 0000 0010**movwf PORTC ; **PORTC = 0000 0010**  
   comf PORTC, W ; **W = 1111 1101**, PORTC = 0000 0010  
   swapf PORTC, F ; W = 1111 1101, **PORTC = 0010 0000**
2. (2 pts) Write a three-instruction sequence that copies PORTB to PORTC without affecting the STATUS register. Assume that Bank 0 is already selected.   
     
   **Ans.**   
   swapf PORTB, W

Movwf PORTC  
swapf PORTC, F

1. (1 pt) Which SFRs have common memory in all four banks?  
     
   **Ans.**

STATUS, FSR, INTCON, PCL, PCLATH

1. (1 pt) Suppose W = 0000 0001. Write one instruction that will toggle bit-0 in W each time the instruction is executed.  
     
   **Ans.**

xorlw B’00000001’

1. (4 pts) The following relates to lab04a interrupts:  
   a) Approximately how long will it take after lab04a begins executing before we get the first Timer0 interrupt? (Hint: see slides)  
     
   **Ans.**Timer0 interrupts on rollover every every16 ×256 = 4096 instruction cycles or 4096 ×1.085 μs = 4.44 ms. This is the longest time that it takes to get the first interrupt. Of the potentiometer is set to any voltage less than 5 vdc, the first time timer interrupt would be less than 4.44 ms.

b) Will the first interrupt turn the PORTC LEDs on or off? Where did you find the answer?

**Ans.**   
The code starts with OnFlag set to 0. Thus it goes to the right branch. In the right branch PORTC is set to zero and TMR0 is set to 51 and thus LED would be off for 205 ticks.

clrf OnFlag ; Initialize OnFlag to 0

c) Suppose the A/D reference voltages are 0 and 5 volts and the A/D input voltage is 4.3 volts. What value will be loaded into TMR0 during the first interrupt?   
  
**Ans.**   
4.3/5\*255 = 219.3 🡪 rounded = 219 🡪 TMR0 = 219 ticks  
  
d) How long will it be between the first and second Timer0 interrupts?

**Ans.**  
Once the routine starts, TMR0 Is set to 219 ticks. Time0 counts down from 219 to 0 and during this time LEDs were all off. Then first interrupt happens. The next time TMR0 is set to 256-219 = 37 ticks, and then the second interrupt happens. Thus between the forst and second interrupt there are 37 ticks. Every 4.44 ms Time rolls over for 256 ticks. Thus the time between the first and second interrupts are:

37/256\*4.44 = .64ms

1. (1 pt) How should T1CON be configured (in binary) to obtain the longest time possible between Timer1 interrupts when using a 3.6864 MHz oscillator? Use x’s for bits that are not related to this question. What would the time between interrupts be in this case (to the nearest millisecond)?  
     
   **Ans.**T1CON = xx**11**1x01

the important bits are (T1CKPS1 T1CKPS0). If set to 11, pre-scalar is set to 8.

Rollover time = 8 \* 256 = 2048 instruction \* 1.085 us = 2.22 ms

1. (2 pts) Suppose that the subroutine MYSUB is located at program memory address 0x1B6D.  
     
   a) How many bits are required for a program memory address?  
     
   **Ans.**  
   0x1B6D = 7021 < 2^13(8192). Thus 13 bits is needed at least

b) How many bits are available in the goto instruction for the address of MYSUB?  
  
**Ans.**

The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.  
  
c) Write the opcode (in binary and hex) for the instruction goto MYSUB.  
  
**Ans.**   
10 1011 0110 1101

1. (1 pt) Assume that the oscillator frequency is 3.6864 MHz and that T1CON = 0011 0001. What should CCPR1H:CCPR1L be set to (to nearest decimal and binary) so that the CCP1 module will interrupt every 255 milliseconds?  
     
   **Ans.**   
   T1CON = 0011 0001🡪 prescale: 8

CCP1 interrupt period = (4 Tosc) (T1 Prescale) (CCPR1H:CCPR1L)

2550000us = 1.085us\*8\*CCPR1 🡪 CCPR1 = 29377 = 01110010 11000001

CCPR1H: 01110010

CCPR1L: 11000001

1. (1 pt) Suppose we want to test whether the register TEMP contains zero. What instruction could we use along with the instruction movf TEMP, F to test for zero? Explain.  
     
   **Ans.** ????

movf TEMP, F ; place TEMP to itself and then check Z

btfss STATUS, Z ; check the status of Z bit

1. (1 pt) How could the xorwf instruction be used to test whether the register TEMP contains zero? Write the instructions.  
     
   **Ans.**  
   movlw B'11111111' ; move 1s to W

xorlw TEMP, F ; XOR W with TEMP and put in F

if Z=1 it means that TEMP has no zero and they are all 1s. If Z is 0 it means that TEMP had at least 1 zero that it is not equal to FF

1. (5 pts) Demonstrate lab04b to the instructor or TA and explain what the Interrupt Service Routine does and how the RB0Int subroutine works.

The program turns the LEDs up and down with potentiometer. And if there is an interrupt from the switch, it toggles the LEDs.

Here is the explanation of what the interrupt service routine does:

* The ISR first figures out whether the interrupt is from and external source or and internal source by checking T01F (external source) or INTF (internal timer flag).
* If the source of interrupt is from T01F, ISR will redirect to Toggle label.
* If the source of interrupt is from INTF, ISR will redirect to RB0Int label. In RB0Int routine, the interrupt flag is cleared, then PORTC is cleared which kills all LEDs and finally toggles all LEDs by COMF command.

Student Name:\_\_\_\_\_\_\_\_\_\_REZA SHISHEIE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
Instructor/TA signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (10 pts) Add two instructions to the TOGGLE subroutine in the lab04b code so that as you turn the potentiometer, it makes the first four of the LEDs brighter and the other four dimmer. Demonstrate your lab to the instructor or TA and explain your code changes.

Instructor/TA signature \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_

btfsc OnFlag, 0 ; if OnFlag = 1 execute next line

movlw 0xF0

btfss OnFlag, 0 ; if OnFlag = 0 execute next line movlw 0x0F  
  
  


