**Lab 9:**

**Debouncing, EEPROM Data Memory, Flash EEPROM Program Memory, and Indirect Addressing**

**Name (Print):** \_\_\_\_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, Apr 18, beginning of class  
TTH – Tue, Apr 17, beginning of class**

**Circle one: MW or TTH**

1. (1 pt) Which register and which bit (bit number and mnemonic) indicates that a write operation to program memory was interrupted by a reset?  
     
   **Ans. DONE**

**EECON1<3> = WRERR. The WRERR bit is used to indicate when the PIC16F87X device has been reset during a write operation. WRERR should be cleared after Power-on Reset.**

1. (1 pt) According to Chapter 15 in the data sheet, what does the parameter D133 (TPEW) signify? What are its typical and maximum values?  
     
   **Ans.**

**That is program FLASH Memory Erase/Write cycle time. It is the time needed to erase and write on FLASH. The maximum value is 8 ms**

1. (1 pt) According to Chapter 15 in the data sheet, how many EEPROM read / write cycles can the PIC endure? How many Flash EEPROM read / write cycles can the PIC endure?  
     
   **Ans.**

**EEPROM read / write cycle: 100K**

**Flash EEPROM read / write cycle: 1000**

1. (1.5 pts) Answer the following:  
     
   a) Which program memory addresses can be written during program execution if the configuration word 2007h = 0x1013?

**Ans. DONE**

**1 0000 0001 0011**

**CP1:CP0 = 10 —> 1F00h to 1FFFh code protected**

**CP1:CP0 = 01 —> no identical to the first one —>code protection does not work 🡪**

**NONE. No code protection**

**WRT: 0 —> Unprotected program memory may not be written to by EECON control**

b) Which program memory addresses can be written during program execution if the configuration word is 0x1213?   
  
**Ans.** **DONE**  
**1 0010 0001 0011**

**CP1:CP0 = 10 —> 1F00h to 1FFFh code protected**

**CP1:CP0 = 01 —> no identical to the first one —> code protection does not work 🡪**

**NONE. No code protection**

**WRT: 1 —> Unprotected program memory may be written to by EECON control**

c) Which program memory addresses can be written during program execution if the configuration word is 0x1223?  
  
**Ans. DONE  
1 0010 0010 0011**

**CP1:CP0 = 10 —> 1F00h to 1FFFh code protected**

**CP1:CP0 = 10 —> CP pairs are identical**

**WRT: 1 —> Unprotected program memory may be written to by EECON control**

**Thus only the non-protected part can be written**

**Unprotected: 0000h to 1EFFh**

**Protected: 1F00h to 1FFFh**

1. (1 pt) In lab09, an RB0 interrupt can occur when the button is pressed or when the button is released. Which register and which bit (bit number and mnemonic) determines this option? (Hint: See PORTB in Chapter 3.)  
     
   **Ans. DONE**

**It is clear on code 🡪 Interrupt on the falling edge of RB0 🡪 when you push**

**External interrupt on the RB0/INT pin is edge triggered, either rising, if bit INTEDG (OPTION\_REG<6>) is set, or falling, if the INTEDG bit is clear.**

1. (1 pt) Which register and which bit (bit number and mnemonic) determines whether an RB0 interrupt event will wake the PIC from sleep?  
     
   **Ans.**   
   **External interrupt on the RB0/INT pin is edge triggered. When a valid edge appears on the RB0/INT pin, flag bit INTF (INTCON<1>) is set. The INT interrupt can wake-up the processor from SLEEP, if bit INTE was set prior to going into SLEEP.**
2. (1 pt) Select the **Window→EEPROM** menu item in MPLAB. What is the value of each byte of data EEPROM? How many bytes of data EEPROM are available in the PIC16F877?   
   **Ans.**

**They are all FF. There are 16x16 bytes = 256 available bytes**

1. (2 pts) What value (in hex) does EEDATAH contain after the following code executes?  
     
    bsf STATUS, IRP ; IRP = 1 —> bank 2 and 3   
    movlw EEDATA ; W = EEDATA  
    movwf FSR ; FSR=W (address of EEDATA in FSR)  
    incf FSR, F ; FSR = FSR + 1  
    incf FSR, F ; FSR = FSR + 2   
    movwf INDF ; W = FSR

**Ans. DONE**

**W still contains the address of EEDATA which is 0x10C and the address in FSR is incremented twice to the address of EEDATAH. Thus EEDATAH has the address of the EEDATA which is 0x10C (1 0000 1100). However movwf only transfers 7 bits. And thus only 000 1100 (0xc) is transferred.**

1. (1 pt) Suppose the address 0x6F contains the value 0x2D. Write a three-instruction sequence that multiples the contents of address 0x6F by 4.  
     
   **Ans. DONE**

**movlw 0x6F**

**movwf FSR**

**movlw INDF**

**rlf W, 0**

**rlf W, 0**

**movwl INDF**

1. (1 pt) Since address 0x6F is an 8-bit register, the previous question assumes that it originally contains a number less than 64 decimal. What will the code in the answer to the previous question write to 0x6F if it originally contains 64 decimal? What will the code write to 0x6F if it originally contains 128 decimal? Answer in binary.  
   **Ans.**

**64 decimal = 0100 000 🡪 shift left 🡪 1000 000 = 128**

**128 decimal = 1000 000 🡪 shift left 🡪 0000 000 = 0 with a carry bit set**

1. (1 pt) Write a goto instruction that skips one instruction without using any labels. (Hint: look at the code in the “EEPROM Data Memory” section of the lecture slides.)  
   **Ans.**

**goto $+2**

1. (4 pts) Run the lab09.asm code. Sometimes the RB0 button bounces, that is, the PIC skips an LED when you press the button. How many LED skips do you see every 20 times that you press the RB0 button?  
     
   Remove the semicolons in lab09.asm so that the RB0 button is debounced. Since debouncing is in effect, you should not see any LEDs skip even after 20 or more button presses.  
     
   Demonstrate the bouncing and debounced operation to the instructor or TA and explain how the debouncing code works.  
     
   **Student Name:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
     
   **Instructor/TA Signature \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_**
2. (8 pts) With debouncing implemented, each time you press the button, write the value of PORTC to program EEPROM at address 0x019B. After a power-on reset, retrieve the saved PORTC value from program memory and write it to PORTC. Your program should therefore rotate the PORTC LEDs each time the user presses the button, and then when the PIC is power-cycled, the saved PORTC value should be restored from EEPROM and turn on the LED that was on when power was removed. Demonstrate your program to the instructor or TA and explain the relevant code.  
     
   **Instructor/TA Signature \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_**