# EE2361 – Midterm - 7/9/2012 Exam time: 1.5 hours

* **This exam is open book, open notes. Use of computers are allowed, but no internet connection is allowed outside of Moodle or your Z: drive. No communication devices allowed.**

**NAME:\_Patrick Barrett\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**1- (20 points) We have the following variable definitions:  
 x set 0x00  
 y set 0x01  
 z set 0x02  
 Write an assembly program that does the following:  
 if (y < 2\*x) {  
 z = 2\*x;  
 } else if (y == 2\*x) {  
 z = 0;  
 } else z = -1;**

|  |
| --- |
| **Did your program compile?**  **Yes** |
| **How did you test your program? Entered edge case values and ran simulator.** |
| **Paste your code here: include p18F4550.inc ;Include Rigister Names and Whatnot**  **x set 0x00**  **y set 0x01**  **z set 0x02**  **case1: movlw 0x02**  **mulwf x**  **movf PRODL, w ;assuming 2\*x is an 8 bit number**  **subwf y, w**  **bnn case2**  **movwf z**  **bra endloop**  **case2: bnz case3**  **clrf z**  **bra endloop**  **case3: clrf z**  **decf z**  **endloop:bra endloop**  **end** |

**2- (10 points) Convert the following program to machine opcodes:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Address (Hex) | Code (Hex) |  |  |  |
|  |  | x | equ | -4 |
|  |  | num | set | 0x25 |
|  |  |  | org | 0x26 |
| **26**  **28** | **EE25**  **F010** |  | lfsr | FSR2, 0x510 |
| **2A** | **0EFC** |  | movlw | x |
| **2C**  **2E** | **CFDE**  **F025** |  | movff | POSTINC2, num |
| **30** | **68DB** |  | setf | PLUSW2 |
| **32** | **0625** | loop: | decf | num |
| **34** | **6652** |  | tstfsz | num |
| **36** | **D7FD** |  | Bra | loop |
| **38**  **3A** | **EF1C**  **F000** | loopbk: | goto | loopbk |
|  |  |  | end |  |

**3- (10 points) show address calculations for the bra and goto instructions in the above problem.**

|  |
| --- |
| **You answer here.**  **Bra:**  New\_PC = (PC) + 2 + 2n  **0x32 = 0x36 + 2 + 2(n) => n=-3 = 7FD**  **Goto: PC = k << 1**  **0x38/2 = 0x1C** |

**4- (20 points) Assuming that a zero-terminated string is stored at the address pointed to by FSR1, write an assembly program to set PRODL to the number of times the letter ‘k’ appears in the string. If for example FSR1=0x245 and the string “Kiki” is stored at that address, then PRODL should be set to 1 (only the third letter is ‘k’). If “Apple” is stored, then PRODL=0.**

|  |
| --- |
| **How did you test your program? Is your program correct?**  **By using the provided test case in the simulator.**  **It appears so.** |
| **Paste you code here.  include p18F4550.inc**  **st: lfsr FSR1, 0x245**    **p4: movf INDF1, f**  **bz el**  **movlw 'k'**  **subwf POSTINC1, w**  **bnz p4**  **incf PRODL**  **bra p4**  **el: bra el**  **end** |

**5- (20 points) Write a C program (main and the isr) to generate three pulses of width 200s, each separated by a 200s separation. The pulses are output on RD0.**

|  |
| --- |
| **How did you test your program? In the simulator using the stopwatch and logic anaylser.** |
| **Paste your code here: #include <p18f4550.h>**  **#pragma config PLLDIV=2, CPUDIV=OSC1\_PLL2, USBDIV=2, IESO=ON, WDT=OFF**  **#pragma config BOR=OFF, PWRT=ON, LVP=OFF, FOSC=HSPLL\_HS, FCMEN=OFF, VREGEN=OFF**  **#pragma config MCLRE=ON, STVREN=ON, LPT1OSC=ON, PBADEN=OFF**  **//######### Function Declarations ################**  **void high\_isr(void);**  **void low\_isr(void);**  **void setup();**  **//######### Interrupts ################**  **#pragma code high\_isr\_entry=8**  **void high\_isr\_entry(void){**  **\_asm goto high\_isr \_endasm**  **}**  **#pragma interrupt high\_isr**  **void high\_isr(void){**  **static unsigned char counter = 0;**  **if(INTCONbits.TMR0IF == 1){**  **counter++;**  **TMR0H = -2400 >> 8; //Reset Counter**  **TMR0L = -2400; // -**  **INTCONbits.TMR0IF = 0; //Reset Flag**  **PORTD ^= 1; //Twiddle RD0**  **if(counter >= 7){**  **T0CONbits.TMR0ON = 0; //Turn off Timer After 3 complete pulses**  **}**  **}**  **}**  **//######### Functions ################**  **void setup(){**  **TRISB = 0x03; // set RB0, RB1 to input**  **TRISC = 0x00; //Set All to Output**  **TRISD = 0x00; //Set All to Output**  **PORTD = 0x00; //Turn off All Pins**  **INTCON2bits.TMR0IP = 1; //High Priority = 1; Low = 0**  **T0CONbits.T0CS = 0; //Clock from interal = 0, Pin Count = 1**  **T0CONbits.PSA = 1; //0 = Use; 1 = Don't Use Prescaler**  **T0CONbits.T08BIT = 0; //Use Timer 0 as 16 Bit Timer**  **INTCONbits.TMR0IE = 1; //Enable Timer 0 Interrupt**  **TMR0H = -2400 >> 8; //Preset Timer 0 Counter**  **TMR0L = -2400; // -**  **T0CONbits.TMR0ON = 1; //Turn on Timer 0**  **INTCONbits.GIE = 1; // Enable Interrups Globally**  **}**  **void main(void) {**  **setup();**  **while(1) {**  **}**  **}** |

**6- (20 points) Use polling with Timer0 to generate a delay of 2.5 minutes. Fosc=40MHz.**

|  |
| --- |
| **How did you test your program with an Fosc=40MHz? Does your program work?**  **Set simulator to 40Mhz, tested with shorter delay.** |
| **Paste you code here: #include <p18f4550.h>**  **#pragma config PLLDIV=2, CPUDIV=OSC1\_PLL2, USBDIV=2, IESO=ON, WDT=OFF**  **#pragma config BOR=OFF, PWRT=ON, LVP=OFF, FOSC=HSPLL\_HS, FCMEN=OFF, VREGEN=OFF**  **#pragma config MCLRE=ON, STVREN=ON, LPT1OSC=ON, PBADEN=OFF**  **//######### Function Declarations ################**  **void high\_isr(void);**  **void low\_isr(void);**  **void setup();**  **//######### Interrupts ################**  **#pragma code high\_isr\_entry=8**  **void high\_isr\_entry(void){**  **\_asm goto high\_isr \_endasm**  **}**  **#pragma interrupt high\_isr**  **void high\_isr(void){**  **}**  **//######### Functions ################**  **void setup(){**  **TRISB = 0x03; // set RB0, RB1 to input**  **TRISC = 0x00; //Set All to Output**  **TRISD = 0x00; //Set All to Output**  **PORTD = 0x00; //Turn off All Pins**  **INTCON2bits.TMR0IP = 1; //High Priority = 1; Low = 0**  **T0CONbits.T0CS = 0; //Clock from interal = 0, Pin Count = 1**  **T0CONbits.PSA = 0; //0 = Use; 1 = Don't Use Prescaler**  **T0CONbits.T0PS2 = 1; //Set Prescaler = 256 (Pg: 129)**  **T0CONbits.T0PS1 = 1; // -**  **T0CONbits.T0PS0 = 1; // -**  **T0CONbits.T08BIT = 0; //Use Timer 0 as 16 Bit Timer**  **INTCONbits.TMR0IE = 1; //Enable Timer 0 Interrupt**  **T0CONbits.TMR0ON = 1; //Turn on Timer 0**  **}**  **void main(void) {**  **unsigned char countdown = 89; //Round down, plenty of overhead anyway**  **setup();**  **while(1) {**  **if(INTCONbits.TMR0IF == 1){**  **countdown--;**  **INTCONbits.TMR0IF = 0; //Reset Flag**  **PORTD ^= 1; //Twiddle RD0**  **if(countdown == 0){**  **T0CONbits.TMR0ON = 0; //Turn off Timer After 3 complete pulses**  **}**  **}**  **}**  **}** |