EECE.3170: Microprocessor Systems Design I

Fall 2016

Exam 3
December 14, 2016

Name: ____

For this exam, you may use a calculator and one 8.5" x 11" double-sided page of notes. All other electronic devices (e.g., cellular phones, laptops) are prohibited. If you have a cellular phone, please turn it off prior to the start of the exam to avoid distracting other students.

The exam contains 4 questions for a total of 100 points. Please answer the questions in the spaces provided. If you need additional space, use the back of the page on which the question is written and clearly indicate that you have done so.

Please note: Question 4 has three parts, but you are only required to complete two of them. You may complete all three parts for up to 10 points of extra credit. If you do so, **please clearly indicate which part is the extra one—I will assume it is part (c) if you mark none of them.**

You will be provided with two pages (1 double-sided sheet) of reference material for the exam that contain the PIC16F1829 instruction set. You do not have to submit this sheet when you turn in your exam.

You will have 3 hours to complete this exam.

Q1: Multiple choice	/ 20
Q2: General microcontroller programming	/ 25
Q3: PIC interrupt service routines	/ 15
Q4: PIC assembly programming	/ 40
TOTAL SCORE	/ 100
EXTRA CREDIT	/ 10

1. (20 points, 5 points per part) ***Multiple choice***
For each of the multiple choice questions below, clearly indicate your response by circling or underlining the single choice you think best answers the question.

a. You are running a PIC16F1829 program that uses a single I/O port, Port C. Which of the following TRISC values would configure this port so that half of the pins are inputs and the other half are outputs?

   i. TRISC = 0x00

   ii. TRISC = 0x04

   iii. TRISC = 0x61

   iv. TRISC = 0xB2

   v. TRISC = 0xFF

b. Under what conditions will the following code jump to the label L1?

```
lslf    x, F
btfss   STATUS, Z
btfsc   STATUS, C
goto    L1
```

   i. x = 0 (after shift) and C = 0

   ii. x = 0 (after shift) or C = 0

   iii. x = 0 (after shift) or C = 1

   iv. x ≠ 0 (after shift) or C = 1

   v. x ≠ 0 (after shift) and C = 1

1 (continued)

c. Which of the choices below accurately describes <u>all</u> of the steps that must be taken when an interrupt occurs on a PIC microcontroller?

   i. Save program counter, then jump to start of ISR

   ii. Save flags register, then jump to start of ISR

   iii. Save program counter and flags register, then jump to start of ISR

   iv. Save program counter, flags register, and all file registers, then jump to start of ISR

   v. Save the women and children first, then jump to start of ISR

d. Circle one (or more) of the choices below that you feel best "answers" this "question."

   i. "Thanks for the free points."

   ii. "I don't REALLY have to answer the last three questions, do I?"

   iii. "It's nice to finally have a test that's not at 8:00 AM."

   iv. None of the above.

2. (25 points) ***General microcontroller programming***

a. (8 points) The following code is taken from a program that implements a state machine similar to the one covered in class. Assume this program is run on a PIC16F1829 microcontroller on the development board we used in HW 8, which means the lowest four bits of Port C are wired to the LEDs on the development board. You may assume nothing is connected to the upper four bits of Port C.

The main program repeatedly calls a function "`stChange`", then uses the return value from that function to change states, as shown in the snippet from the main program:

```
stChange
  movf    PORTC, W
  andlw   0x03
  addwf   PCL, W
  retlw   B'00000110'
  retlw   B'00000111'
  retlw   B'00001001'
  retlw   B'00001010'
```

In main program (assume these two lines are repeatedly executed during entire program):

```
  call    stChange
  addwf   LATC, F
```

Assume Port C starts with the lowest four bits set to 0100. What repeated sequence of values will be written to the four LEDs? **Show your work for full credit.**

2 (continued)

b. (8 points) Assume you are running a program using the 10-bit analog-to-digital converter (ADC) on the PIC16F1829. The positive reference voltage for the ADC is 10 V; the negative reference voltage is 0 V. The result of each analog-to-digital conversion will be left justified.

Given this configuration, if the analog input voltage is 1 V, what will the value in ADRESH be when the next conversion is complete? **Show your work for full credit.**

2 (continued)

c. (9 points) You are given the following delay loop, which is slightly different than the one used in our in-class examples:

```
whygo:
    decfsz     COUNTL, F
    goto       whygo
    decfsz     COUNTH, F
    goto       once
    goto       release
once:
    decf       COUNTL, F
    goto       whygo
release:
    return
```

If COUNTL is initially 50, COUNTH is initially 21, the clock frequency is 250 kHz, and each instruction takes 4 clock cycles, how long does the whole delay loop take? **Show your work for full credit.**

3. (15 points) ***PIC interrupt service routines***

Complete the interrupt service routine below by writing the appropriate line(s) of C code into each blank space. A comment to the right of the blank describes the purpose of each section.

This interrupt service routine detects both switch and timer interrupts and works with a global variable, `count`. The ISR does the following:

- On a timer interrupt, clear the number of LEDs specified by `count` (in other words, if count = 1, set the lowest bit of Port C to 0; if count = 2, set the lowest two bits of Port C to 0, and so on) starting with the lowest LED (the one connected to the LSB of Port C), then reset count to 0.
- On a switch interrupt, increment `count` unless the increment will cause the variable to exceed the number of LEDs wired to Port C. In that case, `count` should be unchanged.

Assume the LEDs are wired to the lowest four bits of Port C (as on the board used in HW 8) and that "SWITCH" and "DOWN" are appropriately defined.

```
void interrupt ISR(void) {

   if (INTCONbits.T0IF) {              // Timer 0 interrupt

      ___________________________      // Clear flag in software

                                       // Clear the number of LEDs
                                       //  specified by count,
                                       //  then reset count to 0





   }

   if (________________________) {     // SW1 was pressed
      IOCAF = 0;                       // Clear flag in software
      __delay_ms(5);                   // Delay for debouncing
      if (SWITCH == DOWN) {            // If switch still pressed

                                       // Increment count unless
                                       //  count will exceed the
                                       //  number of LEDs wired to
                                       //  Port C after being
                                       //  incremented


      }
   }
}
```

4. (40 points, 20 points per part) ***PIC assembly programming***
For each of the following complex operations, write a sequence of PIC 16F1829 instructions—**not C code**—that performs an equivalent operation. **CHOOSE ANY TWO OF THE THREE PARTS** and fill in the space provided with appropriate code. **You may complete all three parts for up to 10 points of extra credit, but must clearly indicate which part is the extra one—I will assume it is part (c) if you mark none of them.**

Assume that 8-bit variables "TEMP" and "COUNT" have been defined for cases where you may need extra variables.

Finally, please note that you are not required to write comments describing each instruction. You may certainly do so if you feel comments will make your solution clearer to the instructor.

a. You are given a 16-bit variable, X, and an 8-bit variable, P. You can access individual bytes within X using the 8-bit variables XL and XH (XL is the least significant byte).

   Write an instruction sequence that sets P to 1 if X is a palindrome (i.e., a value in which the bits can be reversed without changing its value) and 0 otherwise. Three examples of 16-bit palindromes include 0xF00F ($1111\ 0000\ 0000\ 1111_2$), 0x1248 ($0001\ 0010\ 0100\ 1000_2$) and 0xA3C5 ($1010\ 0011\ 1100\ 0101_2$).

   Your solution should not change XL or XH.

4 (continued)

Remember, you can assume that 8-bit variables “TEMP” and “COUNT” have been defined for cases where you may need extra variables.

b. You are given a 16-bit variable, X, and an 8-bit variable, CT. You can access individual bytes within X using the 8-bit variables XL and XH (XL is the least significant byte).

   Write an instruction sequence setting CT equal to the number of nonzero bits in X. For example:

   - If X = 0x0FF0 = $0000\ 1111\ 1111\ 0000_2$, CT = 8
   - If X = 0x1234 = $0001\ 0010\ 0011\ 0100_2$, CT = 5

   Your solution should not change XL or XH.

4 (continued)

Remember, you can assume that 8-bit variables "TEMP" and "COUNT" have been defined for cases where you may need extra variables.

c. You are given two unsigned 16-bit values, X and Y. You can access individual bytes within each value—"X" contains bytes XH and XL (XL is the least-significant byte) and "Y" contains bytes YH and YL.

   Write a sequence of instructions that jumps to location "L1" if X is less than <u>or</u> equal to Y. Your solution should not change any of the bytes of X or Y. Again, assume X and Y are unsigned (i.e., non-negative).