ECE 331 Spring 2013 Homework 7

Due Monday Feb 25 at the beginning of class.

For the problems below that require you to write and verify functional code, you should submit a printed copy of the .LST file and CLEARLY indicate at the top of the page which problem # it belongs to.

Copying someone else's code is not permitted.

NOTE: In the WinIDE program you can use the command ***mm <address> <value>*** to modify a memory value. This is very handy when testing a program; it allows you to change data values without editing and reloading the .asm file. Just remember to reset the PC before running again.

1. Write a complete ASM program that will do the following:
   a. Store the following values in memory starting at location $64A0
      $E2,$D7,$CB,$A0,$9F,$91,$88,$83,$5E,$28,$1F,$01
   b. Begin your code at $4000 by initializing accA to $11
   c. Read through the numbers above sequentially and perform the following conditional tasks
      i. If the S2C number is positive, end program
      ii. If the number is negative and odd, XOR the value with the contents of accA and save it back to the same memory location
      iii. If the number is negative and even, complement (negate) the value and store result into accA.
   d. Once the code is functioning properly, print the final .LST file and submit with your homework.
   e. To verify your code is correct, list the final contents of accA memory $64A0 - $64AF. You can write these by hand or paste a simulator image cropped to the memory block.

2. a. Complete Exercise 2-4 of PC Lab 2 by following these steps (as in the PC Lab 2 slides).
   - Download the "PCLab2-4.asm" file from class website.
   - The existing code assumes data values are stored in memory. Modify the program file to include directives to store 10 values and an initial SUM at the appropriate memory locations.
   - Modify the program so that the counter starts at 10 ($0A) and counts down to 0.
   - Simulate the program and verify it is free of syntax errors and working correctly.
   - Print the final .LST file and submit with your homework.

   b. Describe briefly what you should check/test to determine if your program is correct, and specify the final SUM value your program computed for your data.

3. Branch Set/Clear: Assuming address $50FF holds the value $33, indicate if the following branches will be taken or not taken (no branch).
   a. BRSET $50FF %00110011 ZAY
   b. BRCLR $50FF %00110011 ZAY
   c. BRCLR $50FF %11001100 ZAY
   d. BRSET $50FF %11110000 ZAY

4. Peripheral hardware concepts:
   a. What function do configuration registers hold in microcontroller operation?
   b. What are the two general I/O addressing schemes? What is the difference between them?
   c. What is a parallel I/O port?
   d. How do you determine if a parallel I/O port is an input or an output?

5. Parallel I/O Ports
   a. Write an ASM program segment that will assign the lowest five pins (4:0) of Port B to be inputs and the upper three pins (7:5) to be outputs. Use the specific MCU Register addresses for an HCS12 device; see textbook Appendix C or MC9S12DP256 user manual (under "Registers" chapter) on class website.
   b. Write an ASM program segment that will output a logic low on all output pins of Port B.

6. Using the block diagram for the MC9S12DP256, list the secondary functions (other than general purpose I/O) of each of the following ports on the HCS12. For more information, see the course textbook or MC9S12DP256 user manual on class website.
   a) Port AD0
   b) Port AD1
   c) Port A
   d) Port B
   e) Port E
   f) Port H
   g) Port P
   h) Port S
   i) Port T

7. Operation Modes:
   a. How many Modes of Operation does the HC12/S12 have?
   b. How many of these modes are useful for "normal" applications?
   c. What MCU pin (signal) determines if the chip will operate in a normal or special mode?
   d. Which HCS12 I/O ports are consumed by secondary functions in normal expanded wide mode and not available for general purpose I/O?
   e. In Normal Single-Chip mode, which ports are available for general purpose I/O?