ECE2049 Homework #2 – The MSP430 Architecture & Basic Digital IO

(DUE Friday 9/11/15 At the BEGINNING of class)

Your homework should be neat and professional looking. You will loose points if your HW is not properly submitted (by properly I mean stapled with a cover sheet including your ECE box number)!

- Always review any reading assignments *before* attempting the homework.
- Show all of your work. Clearly indicate results (underline, circle or box).
- **Always write neatly. Code should be typed and properly indented for clarity.**
- *The grader can not be expected to GUESS what you meant to do!*

Reading -- Davies Ch 3, 4 & 7, MSP430x5xx User's Guide Ch 6.1, 6.3,12,

MSP430F5529 datasheet, MSP-EXP430F5529 User's Guide
(see Useful Links page on class website)

1) The MSP430 is a small, 16-bit microcontroller. The device's forte is that it capable of remarkably low power operation. It is not a processor suited to computationally heavy applications. It is a fixed point (i.e. integer) processor. Although the Code Composer Studio compiler provides full support for floating point operations the MSP430 does not have an FPU (floating point unit). That is, there is no circuit inside an MSP430 that performs arithmetic operations on floating point numbers. All floating point math is emulated in software.

i. Add a comment to each line of code saying what it does.

ii. Run these 2 simple programs in Code Composer Studio. Take a screen capture of each being run in the debugger.

iii. Comment on the efficiency (both space and run-time) of the integer math program vs the floating point program. Many (most) of the functions in the C math library <*math.h*> are defined for double precision floating point input and output variables. What does this simple example mean for including math functions like *cos()* or *pow()* in your code fore the MSP430?

Hint: The Console window in the debugger gives the code size or you can look at the size of the equivalent assembly code produced for each program in the Disassembly window. You may assume that it takes an average of 4-6 instruction cycles per assembly instruction to execute. (10 pts)

a)

```c
#include "msp430.h"
#include <stdlib.h>

int main(void)
{     // give size of these variable
      int  arr[100], i;         //
      long unsigned int sq_sum;  //

      WDTCTL = WDTPW + WDTHOLD;   // Stop watchdog timer

      for (i=0; i<100; i++)
      {
         arr[i] = (rand() % 500)-150; //
         sq_sum += arr[i]*arr[i];   //
      }
}
```

b)

```c
#include "msp430.h"
#include <math.h>
#include <stdlib.h>

void main(void)
{     // give size of these variable
      float  arr[100];       //
      int  i;         //
      float sq_sum;  //

      WDTCTL = WDTPW + WDTHOLD;   // Stop watchdog timer

      i = 0;
      while (i<100)
      {
         arr[i] = (rand() % 500)-150.5; //
         sq_sum += pow(arr[i],2.0);   //
         i++;   //
      }
}
```

2) In lab we are using the MSP430F5529 instead of the MSP430F2013 that your text book discusses. Using the MSP430x5xx Users Guide and MSP430F5529 datasheet (under Useful Links on the class website), *fully* answer the following questions about the memory map of the MSP430F5529. (15 pts)

a) Why does the MSP430F5529 have a 20-bit address bus when its word size is 16 bits? Also why are registers R0 (i.e. Program Counter) and R1 (the Stack Pointer) 20 bit registers instead of 16 bit?

b) The MSP430F5529 memory space contains both RAM and FLASH. Why? What is the difference between RAM and FLASH? Why not use just RAM or just FLASH?

c) Flash serves a number of purposes in the MSP430. How much Flash memory does the MSP430F5529 have? Into how many banks is code memory divided? How many bytes per bank? Which bank of code memory contains the address 0x0DF4A?

d) In lab, when CCS downloads a program to the MSP430F5529, where is the executable code stored? Beginning at what address?

e) When you declare a local variable inside your `main()` where in memory is that value stored? What if you declared a global variable, where in memory are globals stored?

3) Refer to the data sheet for the MSP430F5529 (under the class Useful Links page) and msp430h5529.h to *fully* answer the questions below. (10 pts)

a) The MSP430 uses a von Neumann achitecture. What is the main difference between the Harvard architecture and the VonNeumann architecture? Sketch both architectures showing CPU, memory, peripherals and all buses. Name one strength and one weakness for each architecture

b) What is the *address* for Port 6 direction register, the Port 3 input register, and the Port 7 select register. (See Datasheet Tables 19-33).

c) What MSP430 peripheral device uses addresses 00700h-0073Fh? What *package* pins numbers are used for Port 8 digital IO?

d) Most of the MSP430F5529 package pins have multiple functions that are selected using a select register. Why are all these pins multiplexed? What functionality is multiplexed with the Port 4.4-4.7 digital I/O? With Port 1.2 and 2.3? What package pin is V_{cc} and which package pin is GND (V_{ss})?

4) The MSP-EXP430F5529 board we use in lab has 8 LEDs, 2 push buttons and a scroll wheel potentiometer on it. (20 pts)

(a) Referring to resources like the MSP-EXP430F5529 Experimenter Board User's Guide and demo5529.c, fill out a table like the one below listing all the ports, pins, and package pins to which each of these devices is connected. This will be very useful to have in labs. (5 pts)

Device	I/O Port and Pins	Package Pins
Touch Pad LEDs		
LEDs 1-3		
2 Push Buttons		
Scroll Wheel		

(b) Are push buttons 1 & 2 inputs or output? What about the scroll wheel? How do you know? Draw out the schematic showing how button 2 is connected. What logic value results when button 2 is pressed? Fully explain your answer. *Hint*: Refer to the schematics at the end of the MSP-EXP430F5529 Experimenter Board User's Guide (on Useful Links page). (5 pts)

(c) Describe how each of these ports and pins are being used in the code below. Are they being used for digital IO? Are they inputs or outputs, how do you know? What is the value output to P3 if `P1IN` = 0x7E? (10 pts)

```
P3SEL &= ~(BIT7|BIT5|BIT3|BIT1);
P1SEL &= ~(BIT6|BIT4|BIT2|BIT0);

P3DIR = P3DIR | (BIT7|BIT5|BIT3|BIT1);
P1DIR &= ~(BIT6|BIT4|BIT2|BIT0);

myInput = 0;
myInput |= P1IN;
P3OUT = (myInput << 1);
```

5) Write the following functions using CCS C to complete the code framework below. You do not have to compile this code but your code must be typed to be graded. (15 pts)

(a) A function `setupP6()` that selects `P6.3-0` for digital IO with those bits set as inputs and selects `P6.7-4` for digital IO with those bits set as outputs. None of the settings for `P6` should be altered.

(b) A function `P6inOut()` that reads in the bits from `P6.3-0` and outputs the complements on `P6.7-4` , such that P6.4 outputs ~P6.0, P6.7 outputs ~P6.3, etc.

(c) Write a simple `main()` that sets up any variables and shows how your functions would be used. What value would result in P6OUT in your `P6inOut()` function if P6IN =0xFA? If P6IN =0x55?

```
/* Compiler directives  (includes and defines) */
#include "msp430x44x.h"
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* Function prototypes */
void setupP6();
void P6inOut();


/*** Implement your functions here ***/

   . . .



/* Write your main() here */
void main()
{




}
```

6) Assume that 4 slide switches like the one shown below are to be connected to the pins of Port 5.5-2 such that when the switch is slid to the right it connects that pin to GND (0V) and when slid to the left it connects to V_{cc} (3.3V). (20 pts)

(a) Would these switches be digital inputs or outputs? How would they work (i.e. what switch operation would correspond to digital 1 or digital 0)?

3.3V P5.*x* GND

(b) Describe an example application for these switches. How might they be used? How is their functionality different from the push buttons (i.e. Why would you use slid switches instead of buttons)?

(c) Write 2 functions in Code Composed Studio C to configure and use the switches and a simple main which uses your functions. The function `switchIO()` should return a value between 0-F hex which corresponds to the switch settings assuming P5.2 is the LSB and P5.5 is the MSB. You should leave Drive Strength in its default configuration and disable pull up/ pull down resistors. You do not have to compile this code but it does need to be typed.

```
/* Function prototypes */
void switchConfig();
char switchIO();
```

(d) Assume your program has properly configured the 4 slide switches for digital IO. How would your functions interpret the following settings of the Port 5 input and output registers P5IN = 0xB5 and P5OUT = 0x08, and P4IN = 0x6C and P5OUT = 0x00? What would the function `switchIO()` return in each case?

ECE2049 Homework #2

Submitted by:_____

ECE Box #:_____

Date:_____

Question	Grade
1-- 10	
2-- 10	
3-- 15	
4-- 20	
5-- 15	
6-- 20	
Total: 90	