EEL 4742C: Embedded Systems
Homework 1

QUESTION 1. (10 points)

Part a) A memory is byte addressable and has a 17-bit address. All the addresses are valid. What is the total size of the memory?

Part b) A memory is byte addressable and has a total size of 18,432 bytes (18 KB). What is the smallest address size that can be used for this memory?

QUESTION 2. (10 points)

Part a) A computer maintains memory alignment. At what addresses can we store a byte variable? What about a 16-bit variable?

Part b) A computer maintains memory alignment. Show how the variables below are stored in the memory if they have to be stored in the order they are declared (x, f, y, g, z), starting at address 500. Show the value at each address (including empty spots).

```
unsigned char x;       // 8-bit variable
short int f;           // 16-bit variable
unsigned char y;
short int g;
unsigned char z;
```

Part c) Repeat the question above knowing that the memory should be aligned but the variables can be stored in any order.

QUESTION 3. (10 points)

Part a) Explain the Big Endian and the Little Endian configurations.
Part b) Show how the data (0x12EF) is stored at address 500 in either of the configurations.
Part c) Which configuration is used in the MSP430?

QUESTION 4. (10 points)

Part a) A microcontroller's memory map allocates the FLASH code space to the address range [0x0400 to 0x0BFF]. What is the code size, in bytes, that is supported by this microcontroller?

Part b) The vector table contains memory addresses (a vector is a memory address). In a certain MSP430 device, the vector table is in the range [0xFFC0 to 0xFFFF]. The memory address is 16-bit. How many vectors does this vector table support?

QUESTION 5. **(10 points)**

Part a) Two clock technologies used in microcontrollers are the crystal oscillator and the RC oscillator. For each of them, comment on:

- Startup speed (after the clock has been disabled for a while)
- Accuracy
- Stability with respect to temperature and voltage variations
- Price

QUESTION 6. **(10 points)**

Part a) Why does embedded programming use 'extensions' to C?
Part b) What is the downside of extensions regarding code portability?
Part c) What is an intrinsic function?
Part d) In MSP430, interrupts are enabled/disabled by writing to the bit called GIE in the status register (SR: R2). Why do we use functions like below? Can't we write the code directly in C?

```
_enable_interrupts();
_disable_interrupts();
```

QUESTION 7. **(10 points)**

Part a) How many bits is an integer (int) in the C language?
Part b) Why does embedded programming use data type like (uint_8t, int_8t, uint_16t, int_16t, …) ?
Part c) A programmer wrote a software delay loop that counts the variable (unsigned int counter) from 0 up to 45,000 to create a small delay. If the user wishes to double the delay, can they simply increase the upperbound to 90,000?
Part d) If the code contains a delay loop and we noticed that no delay is being created at run-time. What should we suspect during debugging?

QUESTION 8. **(10 points)**

For the questions below, write the code using the masks that are pre-defined in the header file. (E.g: BIT0:0000 0001, BIT1:0000 0010, …, BIT7:1000 0000).

Perform the operations below on the 8-bit variable (uint_8t data).

Part a) Write code that performs the three operations below. Perform each operation independently of the others.

- Set bit 6.

- Clear bit 6.
- Invert bit 6.

Part b) Write code that performs the three operations below. Perform each operation independently of the others.

- Set bits 4 and 5.
- Clear bits 4 and 5.
- Invert bits 4 and 5.
- Set bit 4 and clear bit 5.

Part c) Write an if-condition line for each of the cases below. Perform each operation independently of the others.

- Check if bit 2 is 1.
- Check if bit 2 is 0.
- Check if bits 3,4 are 1,1.
- Check if bit 3 is 0 and bit 4 is 1.
- Check if bits 3, 4 are 0,0.

QUESTION 9. **(10 points)**

A module on the microcontroller is configured using a control register called CTL that has the format shown below.

SLP	CLK	CAP	IE
2-	3-	2-	1-

- SLP: selects sleep mode; value between 0 and 3
- CLK: selects clock speed; value between 0 and 7
- CAP: selects built-in capacitor value; choice between 0 and 3
- IE: interrupt enable bit (1: enable/ 0: disable)

To support programming the device, the environment has declared the symbolic constants:

SLP_3: $\texttt{\underline{11}00 0000}$
SLP_2: $\texttt{\underline{10}00 0000}$
SLP_1: $\texttt{\underline{01}00 0000}$
SLP_0: $\texttt{\underline{00}00 0000}$

CLK_7: $\texttt{00\underline{11 1}000}$
CLK_6: $\texttt{00\underline{11 0}000}$
…
CLK_0: $\texttt{00\underline{00 0}000}$

CAP_3: $\texttt{0000 0\underline{11}0}$

```
…
CAP_0:     0000 0000

IE:        0000 0001
```

Perform all the operations below using the masks defined above.

Part a) Write a line of code that configures the module as the following:

```
  (Sleep mode 2)(Clock speed 6)(Capacitor value 1)(Interrupts enabled)
```

Part b) For the operation above, show the masks used and the final value of CTL in binary.

Part c) Write a piece of code that changes SLP to 1. The current value of SLP is unknown.

Part d) Write a piece of code that changes CLK to 5. The current value of CLK is unknown.

Part e) Write an if-condition line that checks if SLP=1.

Part f) Write an if-condition line that checks if CLK=5.

Part g) Write an if-condition that checks if the current value of CLK is either of (0, 2, 4, 6).
