**Assignment 1 - Assembly Process, MSP430 Execution, MSP430 Instruction Set**

**Name:**  Mark Demore  
  
**Section:**  M1A  
  
**Documentation:** None

**Assembly Process (10 pts each)**

**What does the assembler do? Be specific.**   
  
  
Convert assembly code into object files in machine code, which make up the executable that actually executes the program.  
  
  
  
  
  
  
  
**What does the linker do? Be specific.**   
  
The linker links… the object files together into the executable that is stored in ROM and runs the program.  
  
  
  
  
  
  
Consider the following code:

mov.w #0xdfec, &0x0200 ; stores the value 0xdfec at memory location 0x0200

**What is the location of each byte of the stored word, assuming big-endian byte ordering?**   
  
  
  
0x0200 – 0xdf  
  
0x0208 – 0xec  
  
  
  
  
**What byte ordering scheme does the MSP430 use? What would be the location of each byte of the stored word?**   
  
Little-Endian is used by MSP430  
  
0x0200 – 0xec  
  
0x0208 – 0xdf

**MSP430 Execution**

**What's the purpose of the program counter? Be specific.** 

The program counter keeps track of the location of the next line of code to be executed. It gives the location of the next instruction.  
  
  
  
  
  
**Assume pc currently holds 0xc000. The current instruction is 4 bytes long. What is the value of pc the instant after this step? Don't overthink this.**   
  
  
  
0xc004  
  
  
  
  
  
  
**What happens if you attempt to access a memory address that isn't implemented in your chip? Talk about reads, writes, and execution.**   
  
  
If you attempt to read or write to the ROM section of memory for example, where the program is stored the program will crash or memory could be corrupted. Similarly, if you attempted to execute code from a register in RAM, the program would crash or memory could be corrupted. If either was attempted at an address that was nonexistent, such as 0xFFFFFFF when memory only goes up to 0xFFFFF, the program would simply crash.  
  
  
**MSP430 Instruction Set**

**MSP430 Instruction Set.**

Consider the following code:

mov.w #0xbeef, r8

swpb r8

and.w #0xff, r8

mov.w r8, &0x0200

inv.w r8

dec.w r8

**Show the contents of r8 after the execution of each instruction. (5 pts each)**

mov.w #0xbeef, r8 r8 = 0xBEEF

swpb r8 r8 = 0xEFBE

and.w #0xff, r8 r8 = 0x00BE

mov.w r8, &0x0200 r8 = 0x00BE

inv.w r8 r8 = 0xFF41

dec.w r8 r8 = 0xFF40