CPE 325: Embedded Systems Laboratory
Laboratory Assignment #4

Assignment [100 pts]

1. Write an assembly program that counts the number of digits and number of characters (length) in the given string. Setup port P1 and port P2 to display the number of digits and length of the string respectively. Assume that a string does not include more than one line. Assume the input string is "Welcome 2 the MSP430 Assembly!" then the correct results for testing purpose are shown below (in decimal): **[50 pts]**

   Total Number of Digits: 4 **(i.e. P1OUT = 0x04)**
   Length of the String: 30 **(i.e. P2OUT = 0x1E)**

2. Write an assembly program that will verify whether the given integer variable, A, is odd or even. Allocate space in memory to store the result (char array) which can be Odd/Even. If the number is even, write "Even" otherwise write "Odd". Assume that A = 20 and address of result is 0x1000. Then the correct results for testing purpose are shown below: **[50 pts]**

   Mem[0x1000] = 'E'
   Mem[0x1001] = 'v'
   Mem[0x1002] = 'e'
   Mem[0x1003] = 'n'

   Hint: In assembly, variables can be initialized as shown below (by default memory is allocated in flash, which is read only):

   General format: Label: datatype value

   Examples:

   ```
   varA: .int 20          ; same as const int varA = 20 in C
   arr:  .cstring "Hello" ; same as const char arr[] = "Hello" in C
   ```

   Note: By default, variables are declared in the **.text** segment (or code segment) which is read only. But **.data** segment is readable and writable and hence to be able to update the variables, declare them in **.data** segment as shown below:

   ```
         .data
   varA: .int 20          ; same as int varA = 20 in C
   arr:  .cstring "Hello" ; same as char arr[] = "Hello" in C
   ```

(**Bonus: up to 10pts**) Write an assembly program that will update an input string. Each Uppercase letter in the input string should be replaced by the corresponding Lowercase letter. Assume that a string does not include more than one line. Count the number of changes and put the output in port P3 output register. Assume the input string is "WELCoME To MSP430 ASSEMBLY!" then the correct results for testing purpose are shown below:

Updated string: "welcome to msp430 assembly!"
Total Number of Changes (P3OUT): 18(12 in Hex)

3. Deliverables
    1. Lab report which includes:
        a. **Screenshots** of outputs
        b. **Flowchart for Q.1 and Q.2**
    2. Source files (.asm files)

Theory Topics

Assembler directives

Addressing modes:

- Register
- Indexed
- Symbolic
- Absolute
- Indirect
- Immediate
- Indirect with autoincrement