

**Metro State University of Denver, Department of Mathematics and Computer Science**  
**CS 2400-001: Computer Organization and Assembly Language, Spring 2014, Dr. Weiyang Zhu**  
**Homework 6, Due Date: 10:00am 03/05/2014, Cutoff Date: 10:00am 03/10/2014**  
**Submission: TWO .s files on Blackboard**

**PLEASE ORGANIZE YOUR WORK IN THE SEQUENCE GIVEN IN THE ASSIGNMENT!!!**

1. Write an ARM program to add the two's complements of a negative **HEXADECIMAL** number,  $-a$ , and a positive hexadecimal number,  $b$ , i.e.,  $(-a) + b$ . For example,  $(-A8F) + 8E0C5$

- in the data area,
  - Declare and initialize a list of **bytes** between label A\_MSD and label A\_LSD as values of your choice (each byte is a Hex symbol in a) and a list of **bytes** between label B\_MSD and label B\_LSD as values of your choice (each byte is a Hex symbol in b). For example, the list could be 10, 8, 15 or 0xA, 0x8, 0xF for the Hex number a.
  - Reserve a **word** with a label of RESULT to be used to store  $(-a) + b$  in 2's complement
- in the main program,
  - Read symbols from memory one by one and convert  $(-a)$  into a 32-bit 2's complement, and  $b$  into a 32-bit 2's complement, respectively. Make sure there is no overflow.
  - If either  $(-a)$  or  $b$  is out of the valid range of a 32-bit 2's complement, set 0x00000000 as the value of the word labeled as RESULT in memory.
  - If any symbol of  $a$  or  $b$  is outside the range of 0 to 15, set 0x00000000 as the value of the word labeled as RESULT in memory.
  - Add the two's complements of  $(-a)$  and  $b$  together and save the result to the word labeled as RESULT in memory

(Hints: (a)  $0xA8F_{16} = ((10 \times 16 + 8) \times 16 + 15)_{10}$ ; (b)  $? \times 16 = ? \times 2^4$ , which can be implemented using Logically Shift Left by 1 bit four times).

2. Write an ARM program to

- in the data area,
  - declare and initialize two NULL-terminated strings labeled as StrOne, and StrTwo.
  - define a symbol called MAX\_LEN and equivalent it with a number like 100 or greater.
  - reserved a chunk of zeroed memory with a size of  $(MAX\_LEN + 1)$ , label this chunk of memory as MixStr. (You may assume that the sum of the lengths of StrOne and StrTwo is no greater than MAX\_LEN)
- in the main program,
  - Merge the first string labeled by StrOne and the second string labeled by StrTwo in a unit of one.
  - Store the ASCII string as a NULL-terminated string to memory labeled as MixStr
  - If one string is longer than the other one, just copy the rest of the longer one to MixStr
  - E.g., `'Hello Metro State!'` and `'I like assembly programming.'` should be mixed as

`'HIe llIiok eM eatrsroe mSbtlayt ep!rogramming.'`