#cec32xNotes, cec32x_lab_36–Lab7_efficient_load_n_store_in_asm

Lab 7. Efficient Load and Store in Assembly

Introduction

We have learned in Module 7 about how to use load and store instructions in various forms to load and save 8-, 16-, and 32-bit signed and unsigned numbers from and to the memory. `140_load_store2.zip` provides many examples of assembly coding using load and store instructions, and WS 7 was assigned to practice these instructions.

In this lab, we do something similar to WS 7—translating the C tasks to assembly tasks. In WS 7, we used global variables to pass variables to assembly functions. In this lab, we will pass the arguments to assembly functions.

We use the same base code provided for WS 7.

To pass arguments to (assembly) functions, we need to modify the prototypes of the (assembly) functions first. These are given below:

```
extern void task10(int8_t *pInt8, int8_t num);
extern void task11(uint8_t *pUint8, int8_t num);
extern void task12(int8_t *pInt8, int16_t *pInt16, int8_t num);
extern void task13(uint8_t *pUint8, uint16_t *pUint16, int8_t num);
extern void task14(int16_t *pInt16, int32_t *pInt32, int8_t num);
extern void task15(uint16_t *pUint16, uint32_t *pUint32,
                   int8_t num);
```

To give you some help, the code for Tasks 10, 12, and 14 is provided in the enclosed .zip file

Unzip the enclosed file, you should be able to build and run the project. The running results should contain something like:

```
Task10:
Value from C: -15; value from asm: -15
Value from C: -11; value from asm: -11
Value from C: -7;  value from asm: -7
Value from C: -3;  value from asm: -3
Value from C: 1;   value from asm: 1
Value from C: 5;   value from asm: 5
Value from C: 9;   value from asm: 9
Value from C: 13;  value from asm: 13
```

```
Task12:
Value from C: 25; value from asm: 25
Value from C: 21; value from asm: 21
Value from C: 17; value from asm: 17
Value from C: 13; value from asm: 13
Value from C: 9;  value from asm: 9
Value from C: 5;  value from asm: 5
Value from C: 1;  value from asm: 1
Value from C: -3; value from asm: -3
```

Tasks of the Lab

Your work for this lab is to finish the revision of the other 3 tasks listed above, with each having 30 points.

After you are done with all the talks, the running results should show the matching results between C code and assembly code.

Lab Report

10 points will be given for the report format and quality, which should include the following:

- Code snippets you have changed/modified together with your comments.
- Screenshot of the results for each of the three tasks. Note that you need to print your name after each task. For example, for Task 12, the first line should read `Results of Task 12 by firstname lastname:` with `firstname` and `lastname` being replaced with your first and last name, respectively.