Workshop 3

In project 140_load_store2, we only addressed the load and store of 32-bit numbers. In this workshop, we extend these functions to cover 8- and 16-bit numbers. Note that there are two versions of the load instruction for both the 8- and 16-bit numbers—the signed version and the unsigned version. Note also that we have learned earlier that the MCU does not care if the numbers are signed or unsigned, but we care—hence it's our programmers' responsibility to use the correct version to have the correct sign extension or zero filling.

Specifically, you are given a new C file (workshop3_main.c) with 6 tasks implemented in C and the calling of 6 assembly functions defined in an assembly file (workshop3_asm_functions_prob.s). Among the 6 assembly functions, you are given the implementation of 3 and you are supposed to finish the remaining three. You can refer to the example codes in 140_load_store2 as the functionalities of this workshop and those of that project are very similar.

When the workshop is done, you will have the results for Tasks 10 to 15 shown in Figures 1 to 7.

You are supposed to do the assignment in teams of two members. You are supposed to finish the coding for Tasks 11, 13, and 15. Below is the point distribution of the assignment.

- 25 points each for correct programming of Tasks 11, 13, and 15.
- 25 points for following the assignment/submission requirements:
    - You need to submit a renamed version of the `main.c` file of the project. Use this convention: `cec320_sec_x_ws3_lastname1_firstname1__lastname2_firstname2.c`, where `x` is your section number.
    - You need to submit a renamed version of the `asm_functions.s` file of the project. Use this convention: `cec320_sec_x_ws3_asm_lastname1_firstname1__lastname2_firstname2.s`.
    - You need also submit a pdf file which contains the screenshots of your assembly functions for the three tasks you are programming and the screenshots of the running results like those given in Figures 1 to 7. Use the same naming convention as above `c` file but with a suffix of `pdf` for this file.

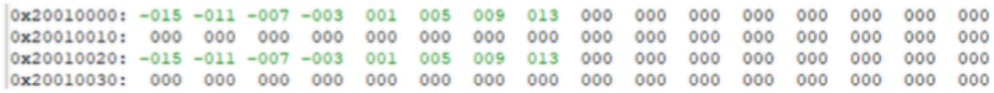

Fig. 1. Results for Task 10. Display parameters: decimal, signed, char.

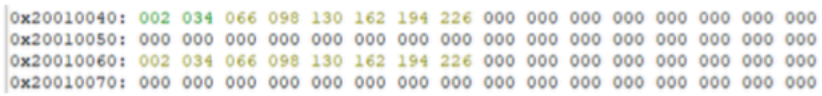

Fig. 2. Results for Task 11. Display parameters: decimal, unsigned, char.

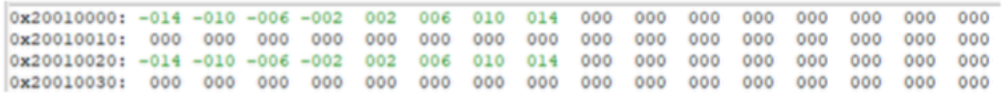

Fig. 3. Results for Task 12. Display parameters: decimal, signed, char. Note that the results here are the increment results of Fig. 1

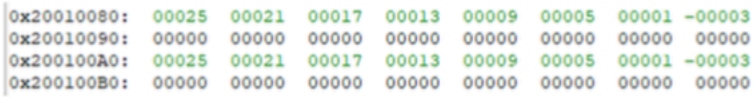

Fig. 4. Results for Task 12. Display parameters: decimal, signed, short.

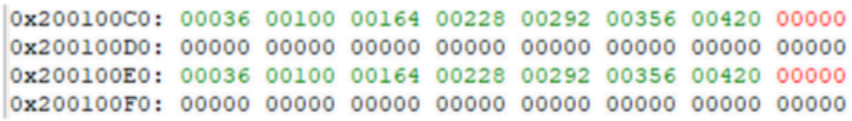

Fig. 5. Results for Task 13. Display parameters: decimal, unsigned, short.

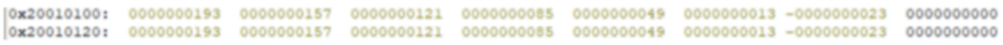

Fig. 6. Results for Task 14. Display parameters: decimal, signed, int.

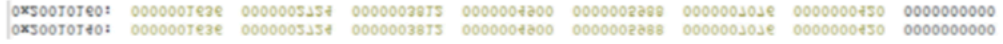

Fig. 7. Results for Task 15. Display parameters: decimal, unsigned, int.