

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.

```

0x20010000: -015 -011 -007 -003 001 005 009 013 000 000 000 000 000 000 000 000
0x20010010: 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
0x20010020: -015 -011 -007 -003 001 005 009 013 000 000 000 000 000 000 000 000
0x20010030: 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000

```

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

```

0x20010040: 002 034 066 098 130 162 194 226 000 000 000 000 000 000 000 000
0x20010050: 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
0x20010060: 002 034 066 098 130 162 194 226 000 000 000 000 000 000 000 000
0x20010070: 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000

```

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

```

0x20010000: -014 -010 -006 -002 002 006 010 014 000 000 000 000 000 000 000 000
0x20010010: 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
0x20010020: -014 -010 -006 -002 002 006 010 014 000 000 000 000 000 000 000 000
0x20010030: 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000

```

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

```

0x20010080: 00025 00021 00017 00013 00009 00005 00001 -00003
0x20010090: 00000 00000 00000 00000 00000 00000 00000 00000
0x200100A0: 00025 00021 00017 00013 00009 00005 00001 -00003
0x200100B0: 00000 00000 00000 00000 00000 00000 00000 00000

```

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

```

0x200100C0: 00036 00100 00164 00228 00292 00356 00420 00000
0x200100D0: 00000 00000 00000 00000 00000 00000 00000 00000
0x200100E0: 00036 00100 00164 00228 00292 00356 00420 00000
0x200100F0: 00000 00000 00000 00000 00000 00000 00000 00000

```

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

```

0x20010100: 0000000193 0000000157 0000000121 0000000085 0000000049 0000000013 -0000000023 0000000000
0x20010120: 0000000193 0000000157 0000000121 0000000085 0000000049 0000000013 -0000000023 0000000000

```

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

```

0x200101E0: 0000007e3e 0000003154 0000002873 0000004200 0000002288 000000101e 0000000430 0000000000
0x200101f0: 0000007e3e 0000003154 0000002873 0000004200 0000002288 000000101e 0000000430 0000000000

```

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