Lab 6. Efficient Load and Store in Assembly

1. Tasks of the lab

We have practiced in Workshop 3 about how to use load/store instructions in various forms to load and save 8-, 16-, and 32-bit signed and unsigned data from and to the memrry. In that project, we used global vareables to pass parameters in order to have a better understanding of the pseudo LDR instruction. In this lab, we modify the assembly functions of Workshop 3 so that we no not need to rely on global variables to pass the parameters. To this end, we we need to redefine the prototype of the assembly functions as

```
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 an example of the revision of the assembly code, the modified code for Task 12 is provided below (see the encloced code for Workshop 3 for the original code)

```
task12  PROC
        PUSH {r4-r5, lr}
        ; r0 = gPtrArray10a
        ; r1 = gPtrArray12a
        ; r2 = gVar1
        MOV  r3, #0
task12_loop
        CMP  r3, r2
        BGE  task12_end
        LDRSB  r4, [r0, r3]
        LDR  r5, =10
        SUB  r5, r4
        STRH r5, [r1, r3, LSL #1]
        ADD  r4, #1
        STRB r4, [r0, r3]
        ADD  r3, #1
        B    task12_loop
task12_end
        POP  {r4-r5, pc}
        ENDP
```

Your work for this lab is to finish the revision of the other 5 tasks of Workshop 3, the points for each task is listed below:

- Task 10: 10 points since the original code is given.

- Task 11: 20 points since the original code is missing from the provided code.
- Task 13: 30 points since the original code is missing from the provided code.
- Task 14: 10 points since the original code is given.
- Task 15: 30 points since the original code is missing from the provided code.

2. Lab report

Like Lab 5, this lab report is special as well—the results of Tasks 11, 13, and 15 will be used to assess an ABET program outcome. The grading will be following the rubrics in the enclosed pdf file. Please read it carefully before doing your programming.

You need to add appropriate comments to all your code. Copy all the code snippets you changed/modified together with your comments to your report. Please be reminded that you need to perform your programming as independently from the TA as possible as seeking too much help will lead to unsatisfactory of some Attributes. Note that each of Tasks 11, 13, and 15 will be assessed using Attributes a, b, c, and d.

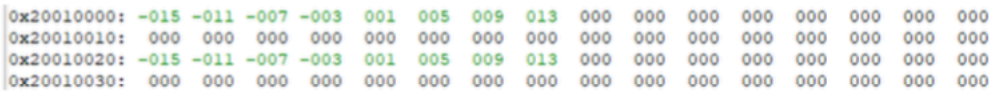

Fig. 1. Results for Task 10. Display parameters: decimal, signed, char.

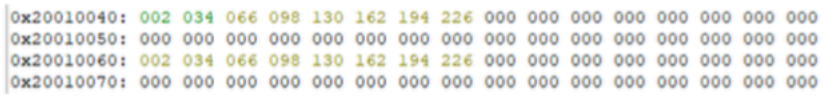

Fig. 2. Results for Task 11. Display parameters: decimal, unsigned, char.

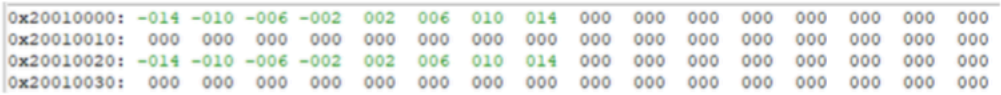

Fig. 3. Results for Task 12. Display parameters: decimal, signed, char. Note that the results here are the increment results of Fig. 1

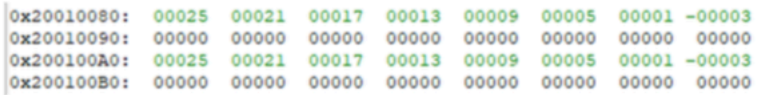

Fig. 4. Results for Task 12. Display parameters: decimal, signed, short.

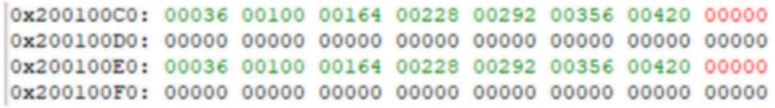

Fig. 5. Results for Task 13. Display parameters: decimal, unsigned, short.

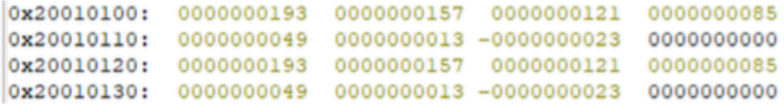

Fig. 6. Results for Task 14. Display parameters: decimal, signed, int.

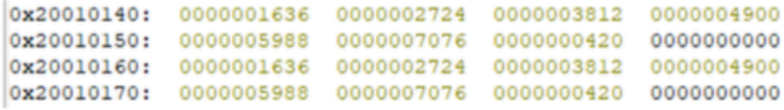

Fig. 7. Results for Task 15. Display parameters: decimal, unsigned, int.