CMPE-250 Laboratory Exercise 4
Iteration and Subroutines

By submitting this report, I attest that its contents are wholly my individual writing about this exercise and that they reflect the submitted code. I further acknowledge that permitted collaboration for this exercise consists only of discussions of concepts with course staff and fellow students; however, other than code provided by the instructor for this exercise, all code was developed by me.

Chris Larson
Performed 18 September 2018
Submitted 25 September 2018

Lab Section 2
Instructor: Muhammad Shaaban
TA: Sebastian Echeverria
Anthony Bacchetta
Sahil Gogna

Lecture Section 01
Professor: Alessandro Sarra

Abstract

This exercise investigated the use of iteration and subroutines in a Cortex-M0+ assembly language program. The objective of the exercise was to work with using conditional flags for conditional branching to iterate and run through subroutines without side effects. A program was made and simulated in Keil MDK-ARM to implement and test an integer division subroutine.

Procedure

A new folder and Keil MDK-ARM project was created for this exercise. A properly commented and formatted Cortex-M0+ assembly language subroutine was written to compute unsigned integral division, and a program was written to test the DIVU subroutine. The DIVU subroutine sets the C flag on return if R0 = 0, otherwise the C flag was cleared. The program was then assembled and built with the Keil ARM assembler to create a listing and a map file. Using the simulator, the program was debugged and verified that the results were correct.

Results

The results from the ARM Cortex-M0+ assembly language program were verified to be correct. This shows that the program was written correctly and functioned well, as shown in Figure 1. The memory tab shows the answer to the division equations that were ran through the program.

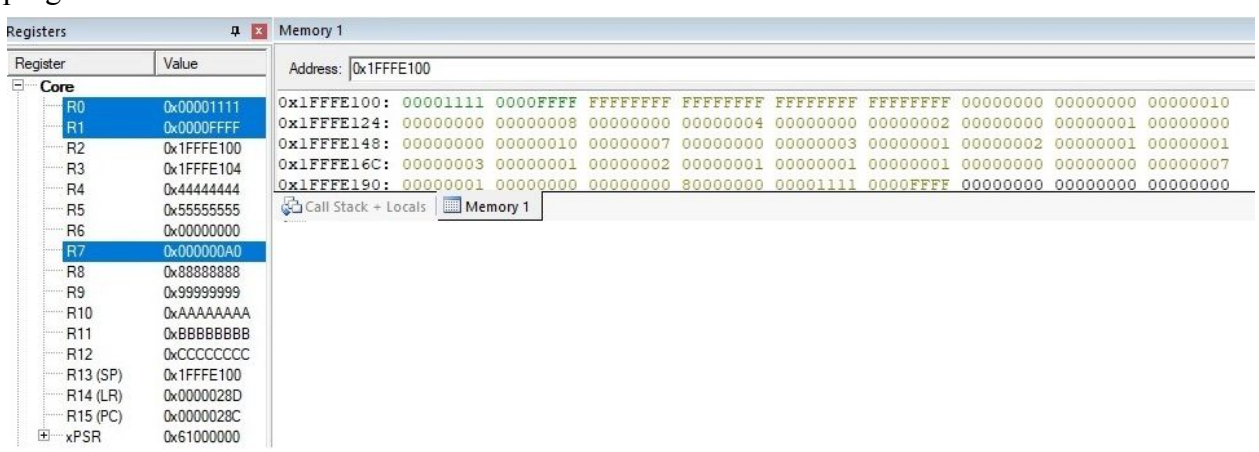

Figure 2 shows that the final value of R6 is 0 so there were zero errors

7)The memory map produced by the program shows where all of the code is being stored in memory. The code for the program was in the memory range 0x00000264 - 0x00000310. Exercise04_Lib.lib library code was 0x000000c0 - 0x00000174. The RAM used for variables was 0x00000100 and for stacks it was 0x0000003a.

Conclusion

The exercise was useful for learning the basics of subroutines and iteration, the push/pop methods, and the use of proc{} to preserve the registers from being changed by the subroutine in

an ARM Cortex-M0+ assembly language program. The program and subroutine performed correctly and gave the results that were expected.