**CMPE 250 Laboratory Exercise Two**

**Basic Arithmetic Operations**

Ari Sanders

Performed: September 3rd, 2015

Submitted: September 10th, 2015

Lab Section: L4

Instructor: Breton Minnehan

TA: Stephen

Ed

Lecture Section: 02

Professor: Roy Melton

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.

|  |  |
| --- | --- |
| Your Signature: |  |

**Abstract:**

The purpose of this exercise was familiarization with basic mathematical functions in assembly. The functions used were addition, subtraction, multiplication, and division. First, the target equation was checked manually and the results of each operation were recorded in hexadecimal. Then, each step was converted to assembly, with addition using the ADDS command, subtraction using SUBS, multiplication implemented as left bitshifts in combination with addition, and division as right bitshifts in combination with addition. Magic numbers were eliminated via use of the equates command (EQU).

**Procedure:**

The exercise began with manually determining the output of each operation the processor would use to calculate the value of the following target expression: -5 + 62 – (9 / 4) – (7 \* 9) + 58 + 17. This was done as follows:

R0: -5 0xFFFFFFFB

R0: -5 + 62 0x00000039

R1: 9 0x00000009

R1: 9 / 4 0x00000002

R0: 0x39 – 2 0x00000037

R1: 7 0x00000007

R2: 7 \* 8 0x00000038

R1: 0x38 + 7 0x0000003F

R0: 37 – 0x3F 0xFFFFFFF8

R0: -8 + 58 0x00000032

R0: 0x32 + 17 0x00000043

Note that 9 / 4 was calculated using integer division. More importantly, note that 7 \* 9 was calculated by multiplying 7 and 8 and then adding 9. This is because the simplest way to multiply in assembly is to bitshift left by the nearest power of two, in this case 2^3, then add the remaining times.  
  
Having determined which values to expect, the operations were converted to assembly commands, with the numbers to bitshift by placed in constants using EQU and all numbers to be added or subtracted first placed in a register.

**Results:**

The result of this exercise was the correct solution to the equation. After each command was run, the output determined in the prelab was checked against the actual output and found to match. The final values of each register are shown in Figure 1.

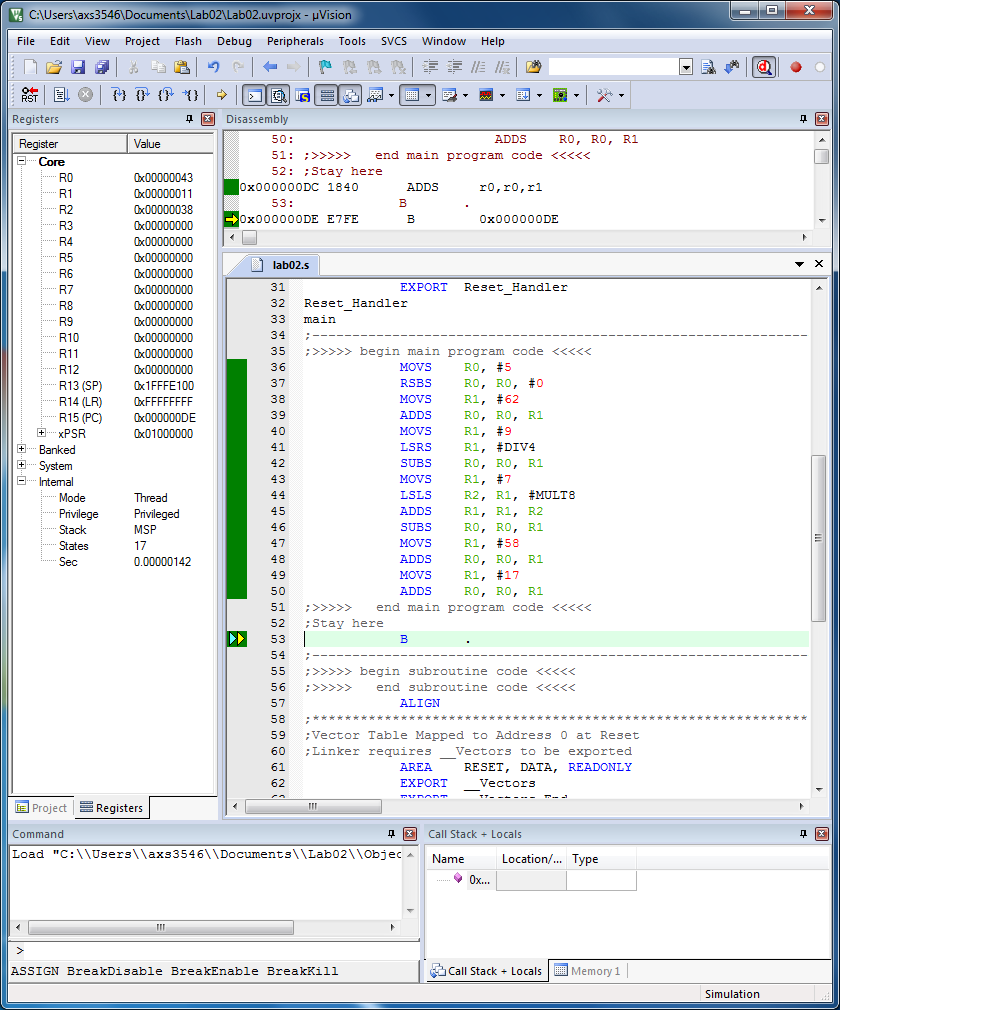


Figure 1: Register Values

**Conclusion:**

This exercise was a success.