David J Tinley
10/22/2023
Lab 3 Report
Computer Organization

For lab 3 I again used the macros I have been implementing since lab1.

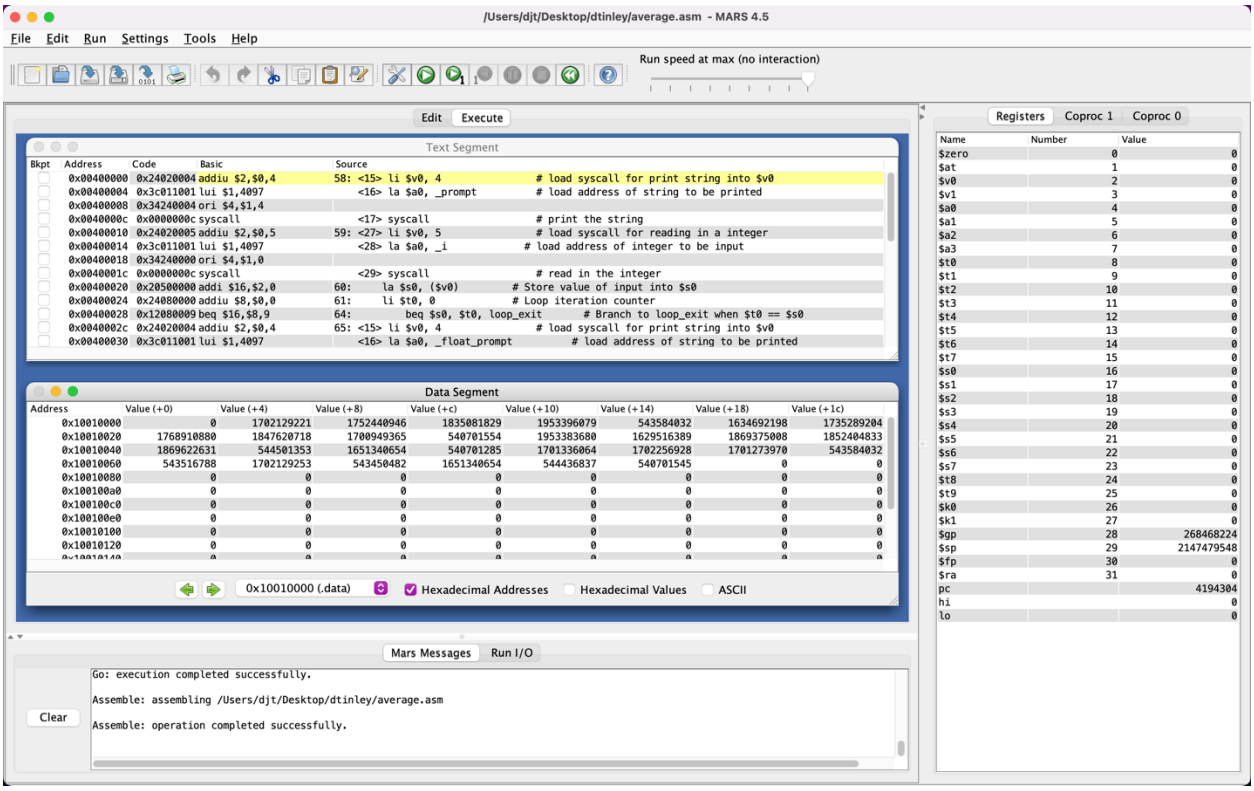
average.asm

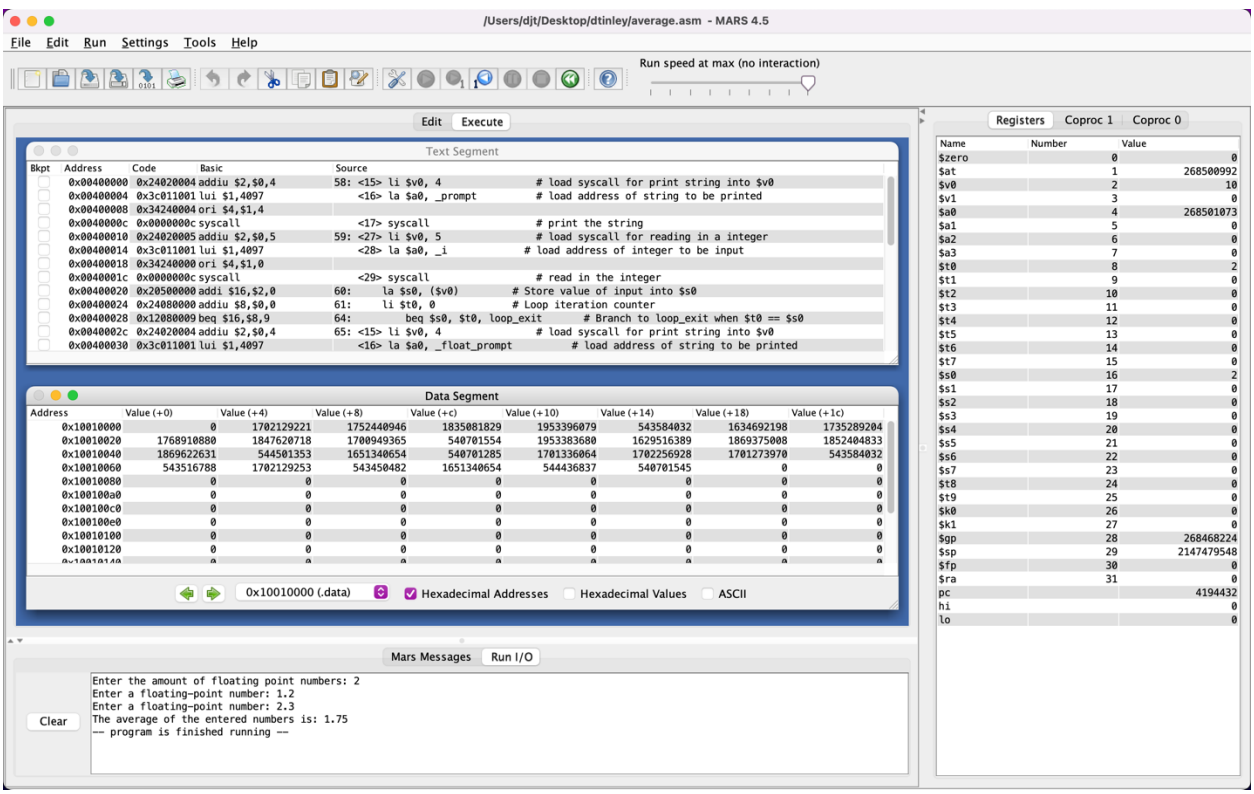
```
##################################################
# Lab 3 - Computer Organization
# David J Tinley
# 10/20/2023
#
# Write an assembly program named average.asm that
# reads a list of floating point numbers from
# the keyboard and displays the measure given above
# on the simulator's console.
##################################################


# MACRO'S ##########################################################
.macro print_string(%string) # macro for printing strings
    li $v0, 4                # load syscall for print string into $v0
    la $a0, %string          # load address of string to be printed
    syscall                  # print the string
.end_macro

.macro print_float(%float)   # macro for printing a float number
    li $v0, 2                # load syscall for printing float
    mov.s $f12, %float       # move single point float value into $f12 register
    syscall                  # print the floating point number
.end_macro

.macro input(%num)           # macro for inputting an integer
    li $v0, 5                # load syscall for reading in a integer
    la $a0, %num             # load address of integer to be input
    syscall                  # read in the integer
.end_macro

.macro f_input()             # macro for inputting a float number
    li $v0, 6                # load syscall for reading a float
    syscall                  # read the float
.end_macro

.macro end()
    li $v0, 10               # macro to end program. 10 = exit
    syscall                  # exit the program
.end_macro
#####################################################################

# DATA ##############################################################
.data
 i: .align 2 # align data to 2^2 for words
```

Next, I declare the data to be used in the program. It was just three different strings and one integer value to be used as the number of inputs to use.

```
# DATA ##########################################################################
.data
_i: .align 2 # align data to 2^2 for words
    .word  0 # _i will be the number of floats to input

_prompt:        .asciiz "Enter the amount of floating point numbers: "
_float_prompt:  .asciiz "Enter a floating-point number: "
_result_message: .asciiz "The average of the entered numbers is: "
##############################################################################

# TEXT ##########################################################################
.text
.globl main

main:
    print_string(_prompt) # Prompt the user for the number of floating-point numbers
    input(_i)             # Read the number of floating-point numbers
    la $s0, ($v0)         # Store value of input into $s0
    li $t0, 0             # Loop iteration counter

    loop:                           # Loop for inputting float numbers
        beq $s0, $t0, loop_exit     # Branch to loop_exit when $t0 == $s0
        print_string(_float_prompt) # Print prompt for inputting floats
        f_input()                   # Float is stored in $f0

        add.s $f1, $f1, $f0         # Add the floats together
        addi $t0, $t0, 1            # Increment iteration counter by 1
        j loop

    loop_exit:
        mtc1 $s0, $f2               # Store the _i integer in float register $f2
        cvt.s.w $f2, $f2            # Convert _i integer to float
        div.s $f3, $f1, $f2         # Divide total of floats ($f1) by _i ($f2). ($f3 = $f1 / $f2)
        print_string(_result_message) # Print the result message
        print_float($f3)            # Print the average of the entered floats
        end()                       # Exit the program correctly
##############################################################################

# EOF #
```

In the main section I first prompt the user for the number of floats they would like to enter and then have them enter those values. As the values are entered there are added together into a single register. Once the total numbers are input it jumps to the next section where the value for the total input numbers is converted into a float. The total of the floats is then divided by the converted number to get the average. The program then exits.





Overall, I did not experience any major difficulties. The main thing I learned was how mips differentiates between single point and double point floats and how they are stored in the co-processors.