Nadine Nam
student ID: 811151197

The Code File:

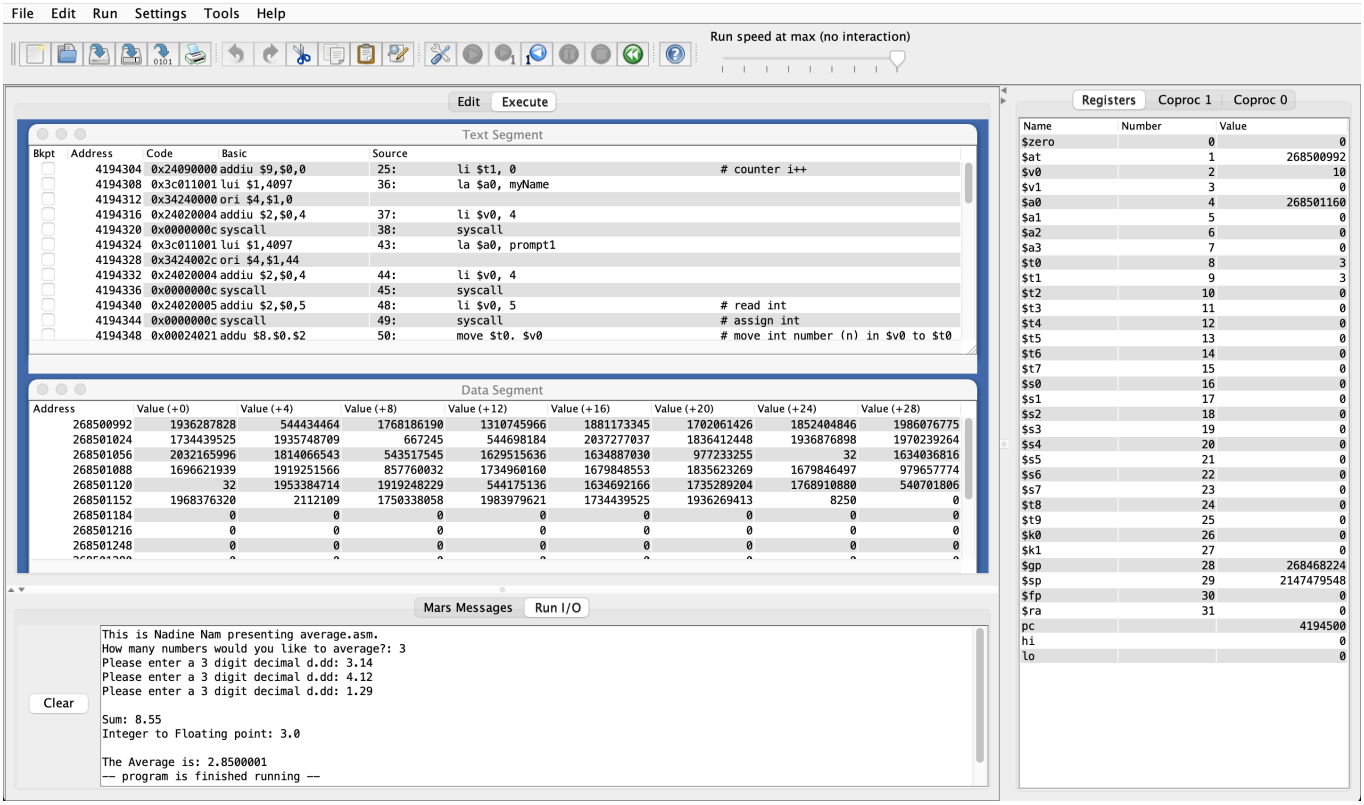
- Nam - average.asm
- Nam - average.pdf

A brief Summary of Project Implementation

Regarding the project 3 implementation, this is learning how to store and calculate floating point numbers by using instructions in MIPS. An user enters numbers to average the sum, and they input decimal numbers according to the number of average values. Then divide the sum by floating points that are converted from integer.

Results showing the working code via screen prints

ENTIRE SCREEN



RUN I/O

The Conclusion listing the lessons learned and problems faced

My project 3 has three procedures. They are main(), Loop(), and Return(). In the main(), it prints out prompt messages to input integer number.

For the Loop() procedure, it repeats looping to get decimal numbers from the user until the end of a for-statement. Once the counter i met the number of n, the process moves on to the next procedure, the Return().

For the Return(), it prints out the sum of decimals values, converts integer to the floating points, and calculates the average of decimals numbers.

Problems I faced

1. At the first time, I couldn't print out the float numbers, but learned that the float type contents should be moved to register $f12 for the output.
2. The most complicated problem I faced in this project was converting the data type from integer to floating points.

Solution:

1.
   - A) To input the decimal numbers, the user should do system call 6 for reading a float.

```
# < Read – Input Decimal Numbers from User >
        li $v0, 6                    # system call 6 for reading a float is
required in $v0
        syscall                  # the user's input is placed in register
$f0
```

1. - B) To print out the floating points, I need to move the stored contents of FP to the register $f12

```
# < Print the number as floating point on the terminal >
        mov.s $f12, $f4
        li $v0, 2
        syscall
```

2. To convert int to float in MIPS, I should use data movement and conversion instructions

```
# < CONVERT int to floating point >
        mtc1 $t0, $f3           # move int stored in $t0 to floating point
register $f3
        cvt.s.w $f4, $f3        # convert (from word to single precision)
$f3 to $f4
                                                # cvt.s.w $f12, $f12 is also
acceptable
```

Lessons

In the project 3, I learned

- How to input output floating points
- How to convert integer to the floating points in MIPS
- Using MIPS floating point instructions
- Types of FP instructions, such as Arithmetic Instructions and Data movement/conversion instructions