Lab Exercise Number 05

68000 Integer Arithmetic

Lab Partners:

Gabriel Stroe

Mostapha Baydoun

**Academic Honor Code:**

*"I have neither given nor received unauthorized aid in completing this work, nor have I presented someone else’s work as my own"*

Lab Dates: 2/25/16 – 3/3/16

Date Submitted: 3/17/16

# Introduction:

# The purpose of this lab is to focus on the proper usage of the easy68k integer arithmetic instructions. It made us become more familiar with the inputs and outputs functions that are used to call them. We became very familiar in making arithmetic integer formula’s to be written in Assembly language using the easy68k simulator in 68000 Assembly language. We also got familiar in the used to call the data and display and how to specify the location of the data that were processed in the code.

# Required Resources:

|  |  |
| --- | --- |
| Lab Resource Identification: | |
| Easy68k Assembler | v5.15.04 |

# Lab Description & Pre-Lab:

In this lab we designed a program that performs integer calculations. The thing to note in this lab is that while we expected a decimal result for many of the calculations but in actuality we get integer numbers as a result. Also do to the maximum number size on a word or long piece of data, sometimes the result actually comes out negative when the expected result is positive.



# Set-up and Procedure:

# We initialized the values of the variables that are used in the formulas, and we have initialized the memory location where the data is saved as specified in the prelab. We typed the formulas for each function. Each function has a their own block of code that defines the formula given in the prelab. We have used the Trap 15 command to display the results of the math operations that were written. When we ran the program, the outcome of the first formula was the same as the prelab but for the second and third the result was different and we have noticed that due to the overflow that occurred due the math operations.

# Results:

As shown in the prelab table (We just combined the two) here is our data that was returned to us.



# Conclusion:

While this lab did have a bit of arithmetic to learn and write in assembly, that was not the actual purpose of this lab. What this lab showed us is that even if we perform the correct function in the correct order, it may not give us the correct return values as there are inherent limits with the base functions. Things like not being able to return a decimal value, a number cap or even seeing a negative number when it should be positive are all things that the programmer needs to be aware of when writing their code.