

**Laboratory Report**

**Course ID: CPS 2390**

**Lab 3: machine/assembly code programs**

**Student: Reuben Hernandez**

**Instructor: Dr. Jing-Chiou Liou**

**Date: 11/16/17**

**Description**

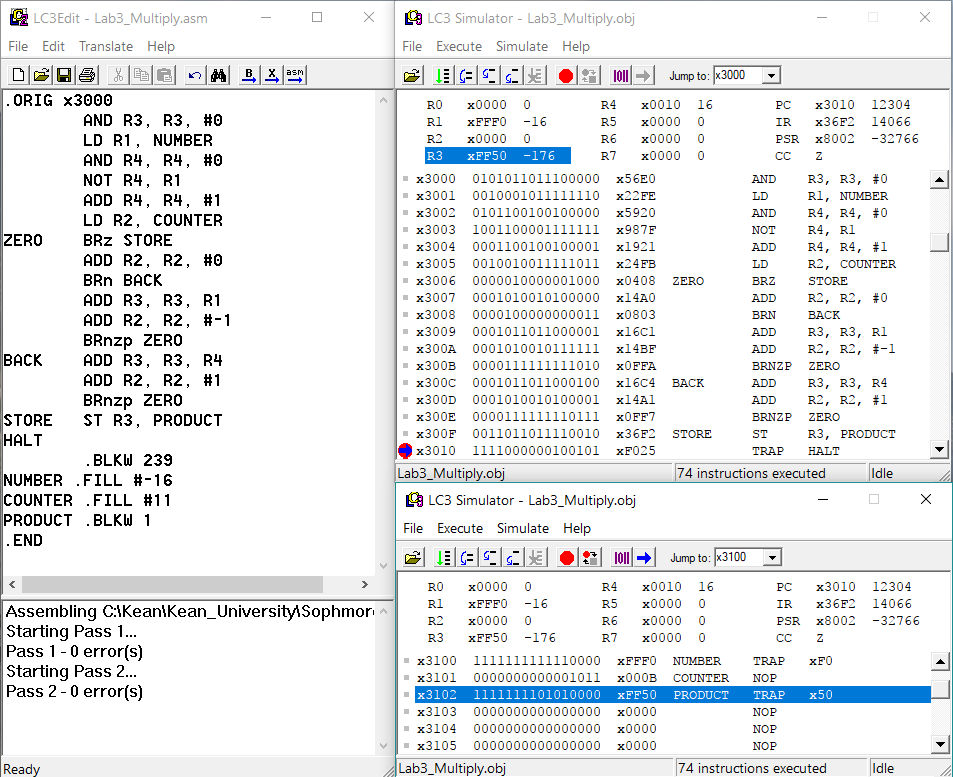
**This lab asks to create an assembly code program that will multiply and divide two integers.**

**Procedure and Notes**

1. **How to implement multiply and divided only using the ADD, NOT instructions?**
2. **You can multiply by adding the first number n times. N = the second number.**
3. **How to get a negative answer when multiplying a negative number and a positive number?**
4. **How to get a positive answer when multiplying two negative numbers?**
5. **Division can be performed by subtracting the dividend by the divisor.**

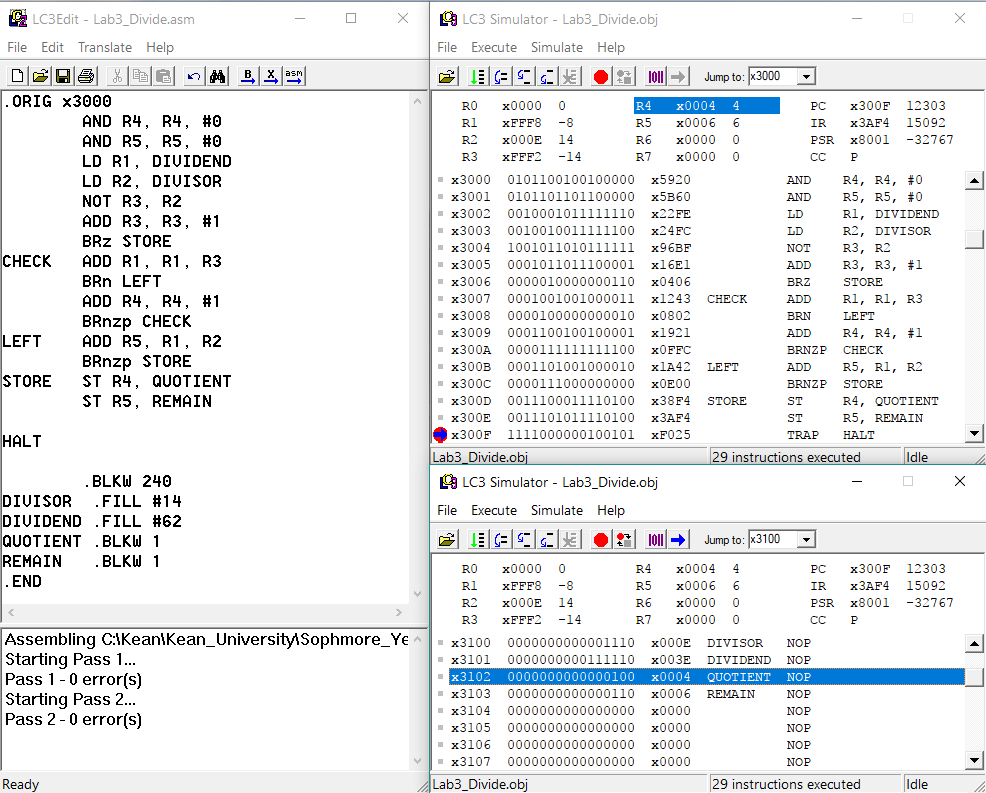
**Results and Reports**

**Multiply**

****

1. **This program loads data from mem[x3100] to R1 and mem[x3101] to R2(counter)**
2. **R4 will contain the 2’s compliment of R1. R3 will hold the result.**
3. **The product checks if the counter if 0 if it is then it will return 0. If not then it will check if the counter is negative.**
4. **If negative then add R3 with R4 and increment the counter; if not then add R3 with R1 and decrement the counter.**
5. **Finally it will store the result back into mem[x3102]**
6. **In this case: -16x11 = -176, the programs stores -176 into memory.**

**Cases it will work under: If both integers are positive. | If one integer is negative and one integer is positive. | If both integers are negative. | If one integer is zero.**

**Divide**

1. **Clears R4 (Quotient) and R5 (Remainder). Loads dividend from memory into R1 and divisor from memory into R2. R3 will contain the 2’s compliment of R2**
2. **It checks if R3 is zero. If yes then it will store 0 back into memory. If not then it will add R1 with R3 and check if the result is negative.**
3. **If the result is negative then it will add R1 with R2 and store into R5 (the remainder).**
4. **If the result is not negative then it will add #1 to R4 (the quotient).**
5. **Once the program is done with the division, it stores the quotient into mem[x3102] and the remainder into mem[x3103].**
6. **In this case: 64/14 = 4 Remainder 6. R4 = 4 and R5 = 6.**

**Cases it will work under: Both numbers must be positive | Dividend > divisor**

**Reference and Acknowledgement**

1. Patt, Yale N., and Sanjay J. Patel. *Introduction to Computing Systems: from Bits and Gates to C and Beyond*. McGraw-Hill Higher Education, 2004.
2. PowerPoint slides from lecture.