##############################
Name: Priscilla Usmani- pusmani
Lab section: 3 MW TA: Jay Roldan
Due: 5/2/13
##############################

Title:

Lab 3: Introduction to LC3

Purpose:

The purpose of this lab is to get familiarized with making programs in LC-3 assembly language. The main goal of the lab is to take an input 2 1-digit decimal numbers from the user. When the program takes the decimal numbers, it will then do couple of operations: subtraction, multiplication, and division. The results will be found in the memory locations: x3100 (substraction), 3101 (multiplication), 3102 (division) and (3103) remainder of division.

Procedure:

Program Behavior:

1. Display a greeting

(Use the command LEA and PUTS to print out the greeting)

2. Prompt the user to input a 1 digit number

(Write the greeting so that the first input number is a digit from 0-9)

3. Get user input using GETC.

(Use the command GETC and PUTC to get a user –enetered character (and put the result in R0) then echo it back right away.)

4. Prompt the user to input another 1 digit number

(Write the greeting so that the user can input another number lower than the second number from 0-9)

5. Get user input using GETC

(Repeat step 3 but use the command ADD to set the default number less than 12)

6. Perform subtraction, multiplication, and division, store the result to the correct memory locations, and output the terminal.

(The subtraction brings the values to the register, subtracts the functions, puts it into the memory location, then resets all used register to 0)

(The multiplication uses a BRp command that loops the input)

(The division also uses a BRp command that looks the input then stores the divided number to R4 and the remainder to R5.)

(After the division function, it stops the processor with the command HALT)

7. Go back to 1

Algorithms and other data:

AND- Register addressing mode when all operands are registers

LEA- computes address and saves in register (pc-relative mode)

PUTS- writes a string to the console

GETC- reads a single character (no echo)
PUTC- used with GETC, writes a single character
ST- Write data from register to memory (pc-relative mode)
LD- Read a value from a memory location(pc relative mode)
NOT- Takes a value and inverts it in the destination register
STI- Write a data from register to memory (memory-indirect mode)
BRp/BRzp- jumps to a new memory location depending on whether or not the previous instruction's result was negative zero or positive
HALT- halts the program
.FILL- Declares a memory location (variable)
.END- Tells assembler where your program source ends

What Went Wrong or What Were The Challenges:

Nothing went wrong, the challenges were understanding how to use BRp and BRzp for multiplication and division. After reading a couple times, I got it to work, but it still wouldn't give the right input. I then moved around the commands and finally got the inputs to multiply and divide.

Other Information:

How does a branch instruction in LC3 work?
Branch specifies one or more condition codes. If specified the code set, the branch is taken.
If branch is not taken, next instruction (PC+1) is executed.
Which store instruction (ST,STI,STR) did you use to store result to the correct memory location? Why did you use that particular instruction and not the other two?
I used STI, it reads from the memory location then loads and stores it to the adder. STI was the most reasonable to read from the memory location.
What is an addressing mode? What are the five LC3 addressing mode and give an example of each one?
Addressing mode is how operands are specified or how the next instruction to execute is specified.
1. Register (ADD), 2. Immediate (AND), 3. PC-RELATIVE (LD),
4. Base & Offset (LDR), 5. Memory Indirect (LDI)
Which register is used as a place to put return value for TRAP instructions?
Register 7
By stepping through your program, what does PUTS trap do? explain.
PUTS trap writes a string to the console. In the program it pints out the greeting 1 and greeting 2.

Conclusion:

In this lab I fully understood how LC3 works and how to read all the commands. I understand what the commands do and how the functions work. Due to my programming background, I understood better how to work this program. This program basically asks the user to input a digit from 0-9 and a second digit lesser than the first one and then the program starts to compile. The program then will perform subtraction, multiplication, and division and stores them in the registers.