**Daniel Meyer – Lab 5 Report**

**Lab5.asm**

;Class:CSE 313 Machine Organization Lab

;Section: 02

;Instructor: Taline Georgiou

;Term: Summer 2019

;Name(s): Daniel Meyer

;Lab#5: Subroutines: Multiplication, Division, Modulus

;Description: This assignment focuses on using subroutines to perform multiplication,

; division, and modulus calculations. The multiplication and division functions are

; subroutines that are accessed using the JSR command. The multiplication function saves

; the R5 and R6 registers so that the original X and Y values aren't modified during

; calculations. There are also checks to see if the product is negative or positive as well

; as a check to see if the numbers are valid for division.

; Given 4 different data sets of X and Y, the program will perform X \* Y, X / Y,

; and X mod Y. X is stored at x3100 and Y is stored at x3101. The product of X \* Y is

; stored at x3102. The quotient and remainder are stored at x3103 and x3104 respectively.

; When running be sure to load the data set before the main program.

.ORIG x3000

AND R4, R4, #0 ;Clear R4

ADD R4, R4, #1 ;R4 = sign bit, initialize to 1, if prod is negative, then sign = 0

;Get X and check sign

LDI R1, X ;R1 = X

ADD R1, R1, #0 ;

BRzp POSX ;Check if X is negative

NOT R4, R4 ;

ADD R4, R4, #1 ;2's complement

NOT R1, R1 ;

ADD R1, R1, #1 ;-X = X

;Get Y and check sign

POSX LDI R2, Y ;R2 = Y

BRzp POSY ;Check if Y is negative

NOT R4, R4 ;

ADD R4, R4, #1 ;2's complement

NOT R2, R2 ;

ADD R2, R2, #1 ;-Y = Y

;Initialize and proceed to multiplication subroutine

POSY AND R3, R3, #0 ;Initialize product

ADD R5, R1, #0 ;

ADD R6, R2, #0 ;Prevents modifying data

JSR MULT ;

;Adjust product with sign bit

ADD R4, R4, #0 ;

BRp POS ;Sign = 1, positive product

NOT R3, R3 ;

ADD R3, R3, #1 ;Sign = 0, negative product

POS STI R3, PROD ;x3102 = product

AND R3, R3, #0 ;Initialize Temp / remainder

AND R4, R4, #0 ;Initialize quotient

AND R5, R5, #0 ;Clear R5

AND R6, R6, #0 ;Clear R6

ADD R5, R5, R1 ;R5 = X

BRn INVALID ;Check if X >= 0

ADD R6, R6, R2 ;R6 = Y

BRnz INVALID ;Check if Y > 0

;ADD R3, R3, R5 ;TEMP = X

NOT R6, R6 ;

ADD R6, R6, #1 ;R6 = -Y

JSR DIV ;

INVALID STI R3, MODU ;

STI R4, QUOT ;

HALT

X .FILL x3100

Y .FILL x3101

PROD .FILL x3102

QUOT .FILL x3103

MODU .FILL x3104

SaveR5 .FILL x3110

SaveR6 .FILL x3111

;Multiplication Sub-Routine

MULT ST R5, SaveR5 ;Save R5

ST R6, SaveR6 ;Save R6

LOOPM ADD R3, R3, R5 ;Add X to product

ADD R6, R6, #-1 ;Decrement counter

BRp LOOPM ;Loop until counter = 0

LD R5, SaveR5 ;Restore R5

LD R6, SaveR6 ;Restore R6

RET ;Return to main program

;Division Sub-Routine

DIV ADD R3, R3, R5 ;

LOOP2 ADD R0, R3, R6 ;

BRn RETURN ;TEMP < Y

ADD R3, R3, R6 ;temp = temp + (-Y)

ADD R4, R4, #1 ;Quotient +1

BR LOOP2 ;

RETURN

RET ;Return to main program

.END

**Data1.asm**

;Class:CSE 313 Machine Organization Lab

;Section: 02

;Instructor: Taline Georgiou

;Term: Summer 2019

;Name(s): Daniel Meyer

;Lab#5: Subroutines: Multiplication, Division, Modulus

;Description: First data file for Lab5, load before Lab5.asm

.ORIG x3100

.FILL #100 ;x3100 = 100

.FILL #17 ;x3101 = 17

.END

**Data2.asm**

;Class:CSE 313 Machine Organization Lab

;Section: 02

;Instructor: Taline Georgiou

;Term: Summer 2019

;Name(s): Daniel Meyer

;Lab#5: Subroutines: Multiplication, Division, Modulus

;Description: Second data file for Lab5, load before Lab5.asm

.ORIG x3100

.FILL #211 ;x3100

.FILL #4 ;x3101

.END

**Data3.asm**

;Class:CSE 313 Machine Organization Lab

;Section: 02

;Instructor: Taline Georgiou

;Term: Summer 2019

;Name(s): Daniel Meyer

;Lab#5: Subroutines: Multiplication, Division, Modulus

;Description: Third data file for Lab5, load before Lab5.asm

.ORIG x3100

.FILL #11 ;x3100

.FILL #-15 ;x3101

.END

**Data4.asm**

;Class:CSE 313 Machine Organization Lab

;Section: 02

;Instructor: Taline Georgiou

;Term: Summer 2019

;Name(s): Daniel Meyer

;Lab#5: Subroutines: Multiplication, Division, Modulus

;Description: Fourth data file for Lab5, load before Lab5.asm

.ORIG x3100

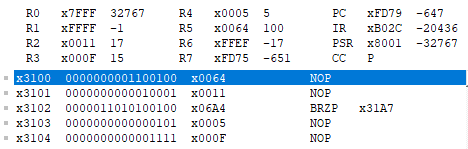
.FILL #12 ;x3100

.FILL #0 ;x3101

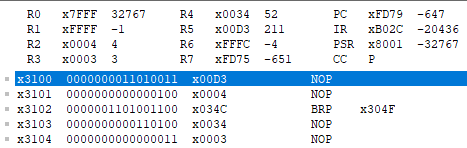
.END

**Screenshots**

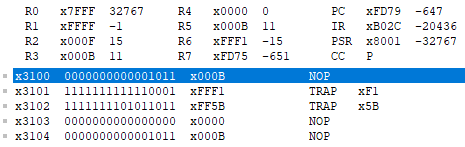
**Results for (100, 17)**



**Results for (211, 4)**



**Results for (11, -15)**



**Results for (12, 0)**

