**Daniel Meyer – Lab 4 Report**

**Lab4.asm**

;Class:CSE 313 Machine Organization Lab

;Section: 02

;Instructor: Taline Georgiou

;Term: Summer 2019

;Name(s): Daniel Meyer

;Lab#4: Fibonacci Numbers

;Description: This program finds the total of the Fibonacci sequence up until

;a specified value. The total is then stored in x3100 and the input is stored

;at x3101. The second half of the program continues to sum the Fibonacci

;numbers until overflow occurs (highest 16-bit two's complement) and that sum is

;saved in x3103. The corresponding N-th Fibonacci number is stored in x3102.

; The input, n, is the n-th Fibonnaci number input from a data file.

;This value is stored at x3100. Fn is the sum of each Fibonacci number and is

;stored at x3101. The largest fibonacci number without overflow, FN is stored at

;x3103. The corresoponding N-th Fibonacci number is stored at x3102.

;Fibonacci Sequence

;1,1,2,3,5,8,13,...

;F1 & F2 always = 1

;Fn = Fn-1 + Fn-2

;Test with n = 15, 20

;x3100 = n

;x3101 = Fn

;x3102 = N

;x3103 = FN

.ORIG x3000

LDI R3, n;

;Fibonacci Algorithm

;Check if n<=2

AND R4, R4, #0 ;Clear R4

ADD R4, R4, #-2 ;R4 = -2

ADD R5, R5, #0 ;Clear R5

ADD R5, R4, R3 ;R5 = -2 + R3

BRzp ELSE ;neg <2, zero =2, pos >2

AND R1, R1, #0 ;Clear R1

ADD R1, R1, #1 ;R1 = 1 = Fn

STI R1, Fn ;Store Fn in x3101

;Find Fn

ELSE AND R1, R1, #0 ;Clear R1

ADD R1, R1, #1 ;R1 = 1 = F1

AND R2, R2, #0 ;Clear R2

ADD R2, R2, #1 ;R2 = 1 = F2

;Initialize the loop

ADD R3, R3, #-2 ;-2 to offset first two 1's

AND R4, R4, #0 ;Clear R4 (Hold F)

AND R5, R5, #0 ;Clear R5 (i=0)

;R1 = A, R2 = B, R4 = Fn, R5 = i

;for(i = 0; i != n; i++)

LOOP1 NOT R6, R3 ;

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

ADD R6, R6, R5 ;check if i != n

BRz STORE ;exit when i = n (n = 0)

ADD R4, R1, R2 ;Fn = B + A

ADD R1, R2, #0 ;A = B

ADD R2, R4, #0 ;B = Fn

ADD R5, R5, #1 ;i++

BR LOOP1 ;

STORE STI R4, Fn ;Store Fn in x3101

;Find FN & N

;Overflow check

;R1 = A, R2 = B, R4 = FN, R5 = i (N)

ADD R5, R5, #2 ;Remvoe offset applied earlier

LOOP2 ADD R4, R1, R2 ;Continue adding

BRn OVER ;When negative, overflow

ADD R1, R2, #0 ;A = B

ADD R2, R4, #0 ;B = FN

ADD R5, R5, #1 ;i++ (N)

BR LOOP2 ;

OVER STI R5, N ;Store N in x3102

STI R2, FN ;Store FN in x3103

HALT

n .FILL x3100

Fn .FILL x3101

N .FILL x3102

FN .FILL x3103

.END

**Data1.asm**

;Class:CSE 313 Machine Organization Lab

;Section: 02

;Instructor: Taline Georgiou

;Term: Summer 2019

;Name(s): Daniel Meyer

;Lab#4: Fibonacci Numbers

;Description: First data file for Lab4, load before Lab4.asm

.ORIG x3100

.FILL #15 ;x3100 = 15

.END

**Data2.asm**

;Class:CSE 313 Machine Organization Lab

;Section: 02

;Instructor: Taline Georgiou

;Term: Summer 2019

;Name(s): Daniel Meyer

;Lab#4: Fibonacci Numbers

;Description: Second data file for Lab4, load before Lab4.asm

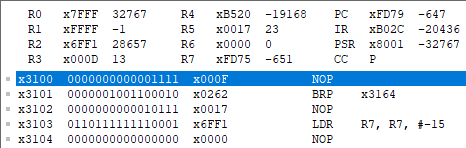
.ORIG x3100

.FILL #20 ;x3100 = 20

.END

**Screenshots**

**Results for inputs *n = 15***



**Results for inputs *n = 20***

