John Galanza

5002981771

[galanj1@unlv.nevada.edu](mailto:galanj1@unlv.nevada.edu)

<https://github.com/JohnGalanza>

<https://github.com/JohnGalanza/supersmashjoe>

**Components:**

None

**Initial Code:**

; DesignAssignment1b.asm;

LDI XL,LOW(0x0400) ;location for numbers div by 3

LDI XH,HIGH(0X0400)

LDI YL,LOW(0x0200) ; puts 00 into lowbits of y register

LDI YH,HIGH(0x0200) ; puts 02 into highbits of y reg

LDI ZL,LOW(0x0600)

LDI ZH,HIGH(0X0600)

LDI R22,99; counter

LDI R23,01 ; increments by 1

LDI R24,12 ; storing number we start at

L1: ST Y,R24 ;store number into Y addr then inc

INC YL

ADD R24,R23 ;inc R24 by 1

DEC R22 ;dec R22 by 1

BRNE L1 ;restart loop if counter isnt 1

LDI YL,LOW(0x0200) ;initialize Y to start at address 0200 again

LDI YH,HIGH(0x0200)

LDI R22,99 ;counter again

LDI R24,0 ;reused to be a temp reg

LDI R20,3 ;divisor

L3:

LDI R19,0 ;Quotient

LD R23,Y ;loads number fromm memory and acts as numerator

LDI R21,0

LDI R18,0

ADD R21,R23 ;R21 acts as a temp register

ADD R18,R21 ;used for comparison

L2: INC R19 ;Following lines are from lec slides

SUB R21,R20

BRCC L2

DEC R19

MUL R19,R20 ;Multiplies quotient and divisor

SUB R18,R19 ;subtracts them to get remainder

BRNE not\_div3 ;compares remainder if it is 0

div\_3: ST X, R23 ;properly stores data into 0x0400

INC XL ;moves to next empty memory location

INC YL ;moves to next mem loc

DEC R22 ;decreases counter

BRNE L3 ;restarts loop again

JMP pt\_2end ;if counter is 0 then skips to next part

not\_div3:ST Z,R23 ;stores number not divisible by 3 into 0x0600

INC ZL ;moves to next empty memory loc

INC YL ;moves to next mem loc

DEC R22 ;decreases counter

BRNE L3 ;restarts loop

pt\_2end:

LDI XL,LOW(0x0400) ;next few lines re initialize pointers

LDI XH,HIGH(0x0400)

LDI ZL,LOW(0x0600)

LDI ZH,HIGH(0x0600)

LDI R20,0 ;acts as zero register

LDI R22,50; Another counter, number is random

L4: LD R21,X ;stores number here

ADD R17,R21 ;adds number to r17

ADC R16,R20 ;adds to r16 if there was overflow

INC XL;moves to the next mem loc

DEC R22 ;decreases counter

BRNE L4

LDI R22,50 ;used as counter again

L5: LD R21,Z ;stores number here

ADD R19,R21 ;adds stored number to r19

ADC R18,R0 ;acounts for carry

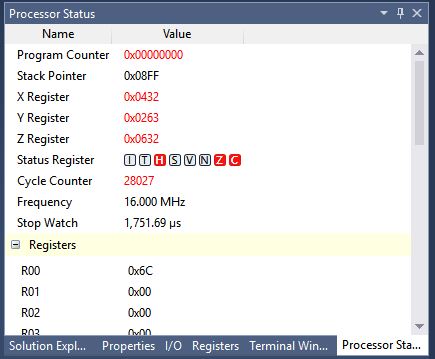
INC ZL ;moves to the next mem loc

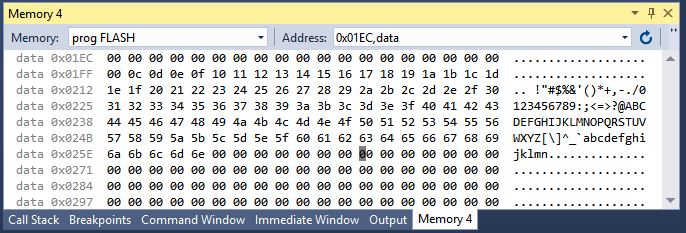
DEC R22 ;decreases counter

BRNE L5

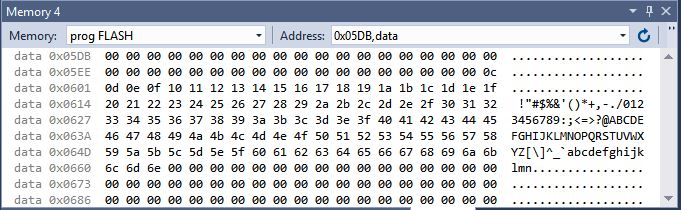
END:

Screenshots



This is a screenshot of the cycles and time

Screenshot of 0x0200 mem location



Screenshot of 0x0600 memory location

I wasnt able to get a proper screen shot of the memory locations because I couldnt get the simulation to end. Also it seemed like the modulo function I implemented didnt work properly since none of the numbers were stored in 0x0400.

“This assignment is my own, original work.”

John Galanza