CPE301 – SPRING 2019

Design Assignment 1B

Student Name: Francisco Mata Carlos

Student #: 1012593607

Student Email: matacarl@unlv.nevada.edu

Primary Github address: https://github.com/chicosisco/da\_sub.git

Directory: repository/cpe301/DesignAssignments/DA\_1B

1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

The components used for this assignment are Atmel studio-7 was used to simulate and debug the code. And Sigma calculator to calculate the sum of the registers, this calculator is found at <https://goodcalculators.com/summation-calculator/>. The calculators used for hex, decimal, and binary were online calculators called rapidtables.com and calculator.net are the online calculator named Calculator.net

1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A**

Insert initial code here

; DA\_1B.asm

;

; Created: 2/22/2019 4:22:41 PM

; Author : Francisco Mata Carlos

;

;========================================================================================

; PART 1

;

; 1. Store 99 numbers starting from the STARTADDS=0x0200 location. Populate the

; value of the memory location by adding high(STARTADDS) and low(STARTADDS).

; Use the X/Y/Z registers as pointers to fill up 99 numbers that are greater than 10

; and less than 255. The numbers can be consecutive or random numbers.

;========================================================================================

ldi xl, 0x00 ; initiliazing pointer

ldi xh, 0x02

ldi r21, 11 ; load value 100 to r20, which is the first value store to memory location 200

loop1:

st x+, r21 ; store value, then move pointer

inc r21 ; increment value on register

cpi xl, 0x63 ; compare if until 99 values have been stored

brne loop1 ; jump to label loop until 99 values have

end :

;rjmp end

;========================================================================================

; PART 2

;

; 2. Use X/Y/Z register addressing to parse through the 99 numbers, if the number is

; divisible by 3 store the number starting from memory location 0x0400, else store

; at location starting at 0x0600.

;

;========================================================================================

ldi yl, 0x00 ; initiliazing pointer starting at 0x0400

ldi yh, 0x04

ldi zl, 0x00 ; initiliazing pointer starting at 0x0600

ldi zh, 0x06

dec xl ; decrementint the lower byte of stack pointer

loop2:

ld r20, x ; loading valule from sram, memory

mov r22, r20

loop3:

subi r22, 3 ; subtract 3 from the value moved from the memery location

cpi r22, 3 ; comparing the value to see if it's divisible by 3

breq to\_400 ; branch if the value in r22 is equal to 3

cpi r22, 3 ; comparing the value in r22 if it's not equal to 3, which means is not divisible by 3

brlo to\_600 ; branch if the value in r22 is less than 3

rjmp loop3

;========================================================================================

to\_600: ; if the value is not divisible by 3, the value is store in SRAM starting at 0x0600 location address

st z+, r20 ; storing value in memory location with starting address 0x0600

dec xl ; decrement lower byte of memory location

cpi xl, 0xFF ; comparting the values to see if all the values from memory location address are zero

breq done ; branch if all memory locations have all gone through

rjmp loop2 ; jump back to loop again

to\_400: ; if the value is divisible by 3, the value is store in SRAM starting at 0x0400 location address

st y+, r20 ; storing value in memory location with starting address 0x0600

dec xl ; decrement lower byte of memory location

cpi xl, 0xFF ; comparting the values to see if all the values from memory location address are zero

breq done ; branch if all memory locations have all gone through

rjmp loop2 ; jump back to loop again

done:

; rjmp done

;========================================================================================

; PART 3

;

; 3. Use X/Y/Z register addressing to simultaneously add numbers from memory

; location 0x0400 and 0x0600 and store the sums at R16:R17 and R18:R19

; respectively. Pay attention to the carry overflow.

;========================================================================================

ldi yl, 0x00 ; initiliazing pointer

ldi yh, 0x04

ldi zl, 0x00 ; initiliazing pointer

ldi zh, 0x06

ldi r16, 0 ; registers 16 and 17 are used to store the summation from the divisible by 3 memory addresses

ldi r17, 0

loop4:

ld r24, y+ ; loading the value of current memory address, and then post incrementing the pointer

add r16, r24 ; adding the values from the divisible by 3 memory locations

adc r17, r0 ; adding the carry from the last sum if any

ld r25, z+ ; loading the value of current memory addres, and then post incrementing the pointer

add r18, r25 ; adding the values from the not divisible by 3 memory locations

adc r19, r0 ; adding the carry from the last sum if any

cpi zl, 0x42 ; comparing the lower byte z rerister from the not divisible by 3 memory locations to check all have gone through

breq finish ; branch if the lower byte register is equal to 0x042 since the decimal is 66 by counting up or by just incrementing

rjmp loop4 ; jump back to loop4 until all memory location address have gone through

finish:

rjmp finish

1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

Same as above

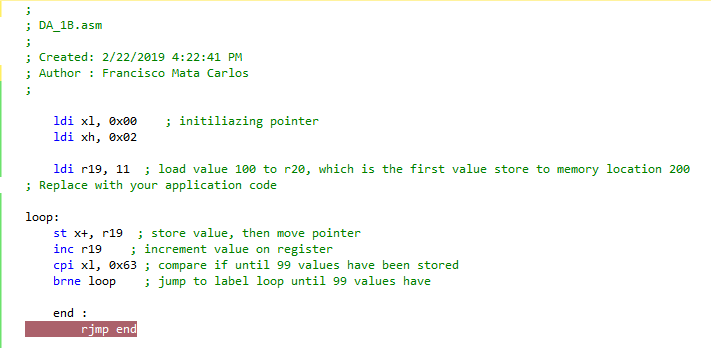
1. **SCHEMATICS**

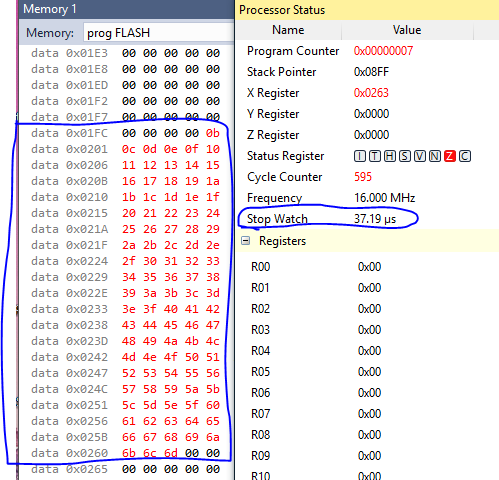
Schematics were not used for this assignment

1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

**Task 1:**

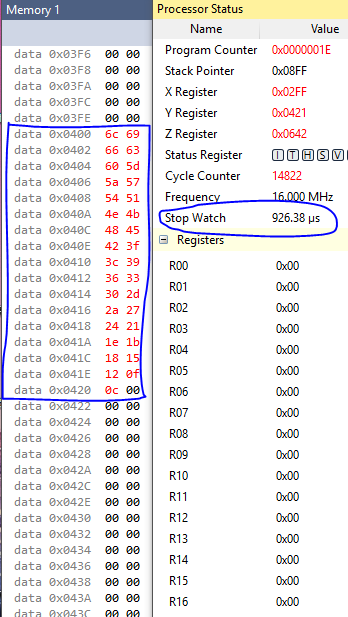
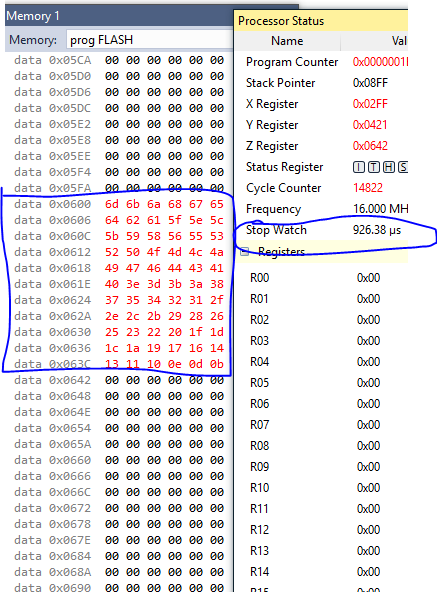
1. Store 99 numbers starting from the STARTADDS=0x0200 location. Populate the value of the memory location by adding high(STARTADDS) and low(STARTADDS). Use the X/Y/Z registers as pointers to fill up 99 numbers that are greater than 10 and less than 255. The numbers can be consecutive or random numbers.

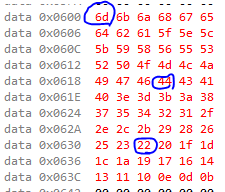


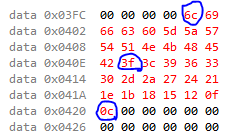


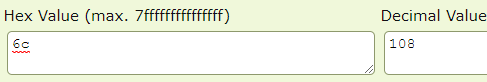
**Task 2:**

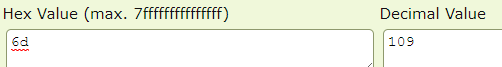
2. Use X/Y/Z register addressing to parse through the 99 numbers, if the number is divisible by 3 store the number starting from memory location 0x0400, else store at location starting at 0x0600

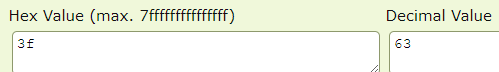


Values not divisible by 3 values divisible by 3

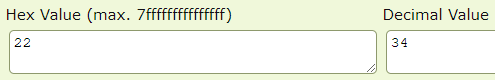
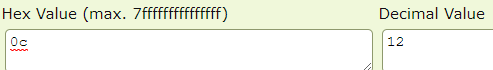






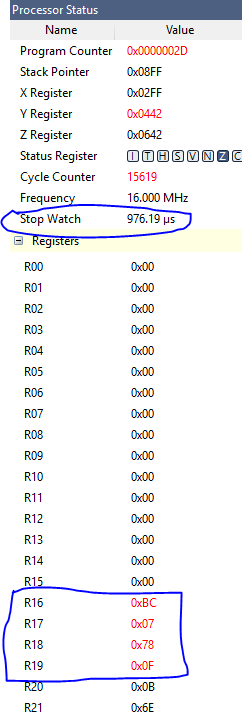


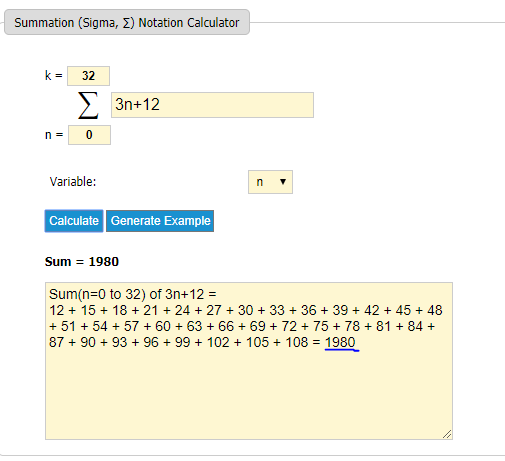
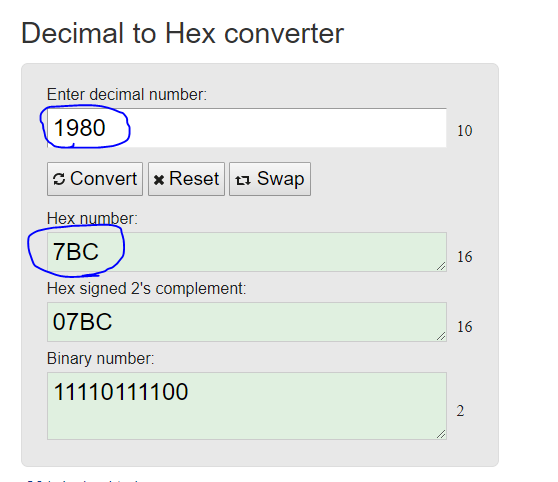


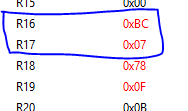


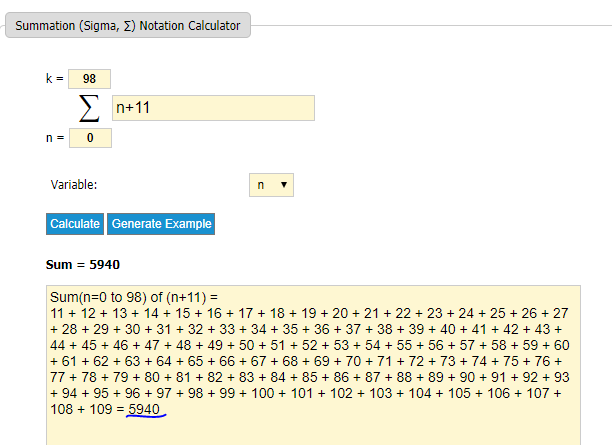
**Task 3:**

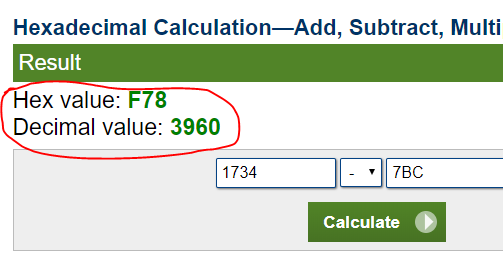
3. Use X/Y/Z register addressing to simultaneously add numbers from memory location 0x0400 and 0x0600 and store the sums at R16:R17 and R18:R19 respectively. Pay attention to the carry overflow.

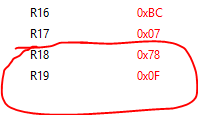












1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

Demo board was not used for this assignment

1. **VIDEO LINKS OF EACH DEMO**

Video of demo was not created for this assignment

1. **GITHUB LINK OF THIS DA**

**Student Academic Misconduct Policy**

<http://studentconduct.unlv.edu/misconduct/policy.html>

“This assignment submission is my own, original work”.

Francisco Mata Carlos