CPE301 – SPRING 2019

Design Assignment 1B

Student Name: Brian West

Student #: 5003032874

Student Email: westb2@unlv.nevada.edu

Primary Github address: https://github.com/westbrian2/Spring2019

Directory: Spring2019/DesignAssignments/DA1B\_submission

Submit the following for all Labs:

1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder ESD301/DA, sub-folder named LABXX, with one document and one video link file for each lab, place modified asm/c files named as LabXX-TYY.asm/c.
3. If multiple asm/c files or other libraries are used, create a folder LabXX-TYY and place these files inside the folder.
4. The folder should have a) Word document (see template), b) source code file(s) and other include files, c) text file with youtube video links (see template).

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

Atmel studio was used as the IDE and for debugging

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

.equ startadds= 0x0200 ; The start of the numbers array

.org 0x0000 ;Start of program

ldi zl,0x00 ;Getting low part of startadds address

ldi zh,0x02 ; getting high part of startadds address

ldi r16,0 ; Storing zero to do additions

ldi r17,99; counter

addlp:

cpi r17,0 ; Exit condition for loop

breq tasktwo; Take branch with counter is 0

add r16,zh ; Adding high part of the address to zero.

lsl r16 ; shift 1

lsl r16 ; shift 2

lsl r16 ; shift 3

lsl r16 ; shift 4

add r16,zl ; Adding low part of the address to previous value.

st z+,r16 ; Storing zh+zl

dec r17 ; decrement counter

ldi r16, 0; Resetting r16

jmp addlp ; restart loop

tasktwo:

ldi zl,0x00 ; Z pointer will point at value in already created array

ldi zh,0x02 ;

ldi yl,0x00 ; Y pointer will point at the divisible by 3 array.

ldi yh,0x04 ;

ldi xl, 0x00; X pointed will point at the other array.

ldi xh, 0x06;

ldi r17,99; counter for existing array

ldi r18,0; counter for 3 array

ldi r19,0; counter for other array

ldi r23,3 ; I'm too lazy to think of anything other than subtracting until the number is 0 or negative for test.

parseLP:

cpi r17,0; checking if whole array is parsed

breq taskthree

ld r21,z+; Getting value from memory and moving pointer.

mov r22,r21; r22 will be the register that gets worked on, r21 will preserve value.

subLP:

sub r22,r23 ; sub by 3 (Moved here to make sure N flag value is set for brmi)

brmi other ; Testing for negative first to check flag

cpi r22,0; Checking if zero, must be done after negative check.

breq equalThree

jmp subLP

equalThree:

st y+,r21 ; storing value into divisible by three.

inc r18; tracking number of elements in array.

dec r17; tracking progress through original array.

jmp parseLP

other:

st x+,r21; storing value into other array

inc r19; tracking number of elements in other array.

dec r17; tracking progress through original array.

jmp parseLP

taskthree:

ldi yl,0x00 ; Y pointer will point at the divisible by 3 array.

ldi yh,0x04 ;

ldi xl, 0x00; X pointed will point at the other array.

ldi xh, 0x06;

mov r22,r18 ; moving 3array counter to r22

ldi r18,0;

mov r25,r19 ; moving other array counter to r25

ldi r19,0;

ldi r23,0 ; will be used for the high registers

sumLP:

cpi r22,0

breq loadother

ld r20,y+ ; loading value from 3 array

dec r22

add r16,r20 ; storing value into answer register for 3 array

adc r17,r23 ; taking carry and adding it to high.

loadother:

cpi r25,0

breq reset

ld r21,x+ ; loading from other array

dec r25;

add r18,r21 ; storing value into answer register for other array

adc r19,r23 ; for carry bits

reset:

mov r24,r22; Adding the two counter registers together, then check if 0.

add r24,r25;

cpi r24,0

breq done

jmp sumLP ;reset loop

done:

jmp done

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

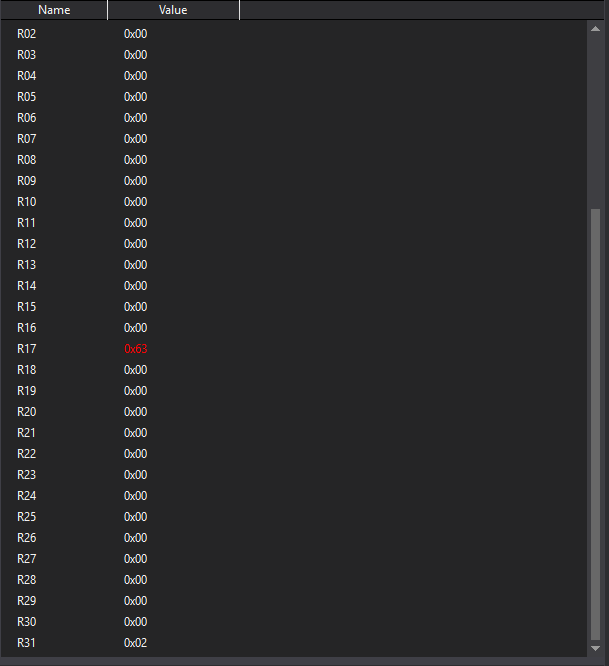
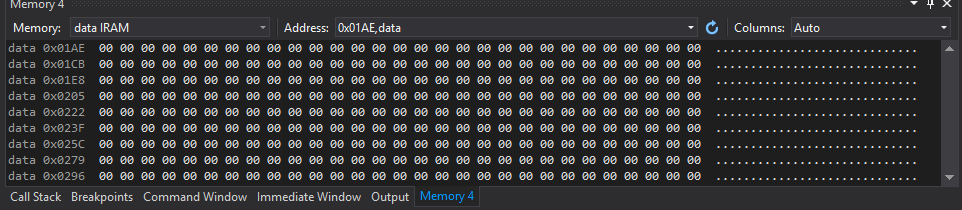
No modifications were made

1. **SCHEMATICS**

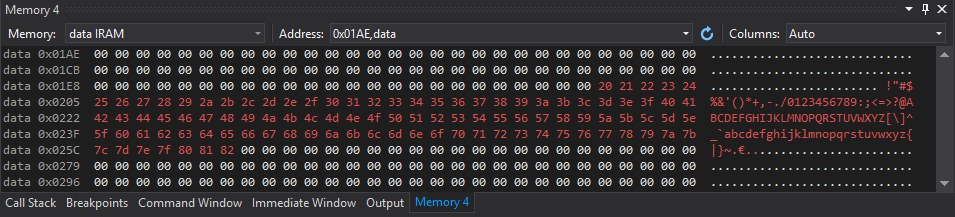
No schematics were used

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

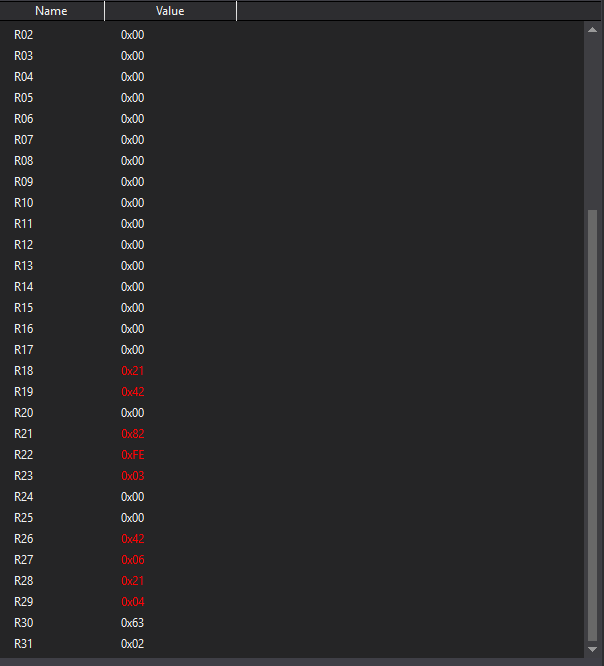
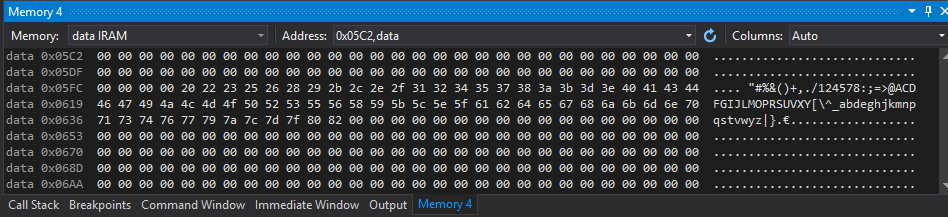
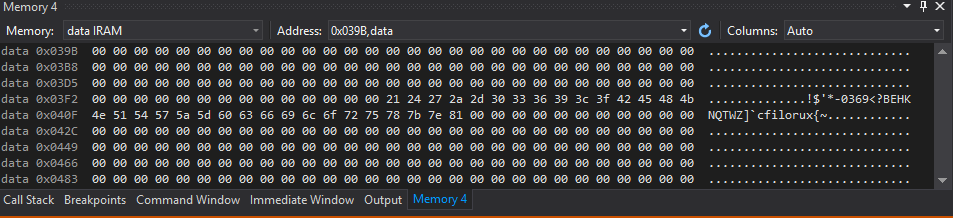
Task1 Before:



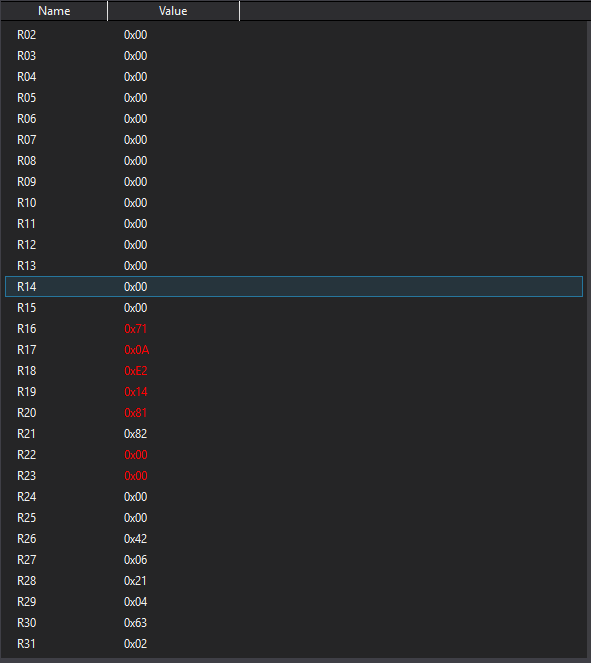
Task 1 After/Task 2 Before:



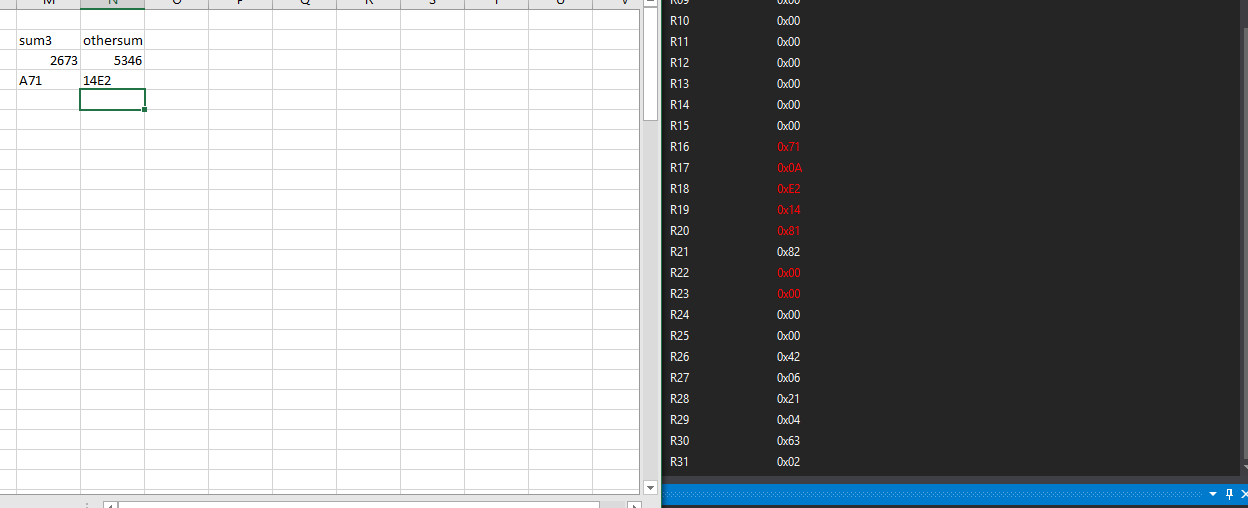
Task 2 After / Task 3 Before:



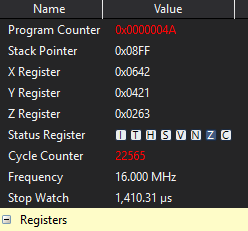
Task 3 After



Task 4 (Verifcation)



Task 5 (Execution time)



1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**
2. **VIDEO LINKS OF EACH DEMO**
3. **GITHUB LINK OF THIS DA**

https://github.com/westbrian2/Spring2019/tree/master/DesignAssignments/DA1B\_submission

**Student Academic Misconduct Policy**

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

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

Brian West