CPE301 – SPRING 2019

Design Assignment 1B

Student Name: Benjamin Johnson

Student #: 5003284489

Student Email: johnsb18@unlv.nevada.edu

Primary Github address: https://github.com/johnsb18/ClassRepository

Directory:

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

;

; AdvancedASM.asm

;

; Created: 2/23/2019 3:09:16 PM

; Author : Benjamin Johnson

; Student #: 5003284489

; Email: johnsb18@unlv.nevada.edu

; --- STORING VALUES ---

.equ STARTADDS = 0x0200 ; set starting address

.equ THREEADDS = 0x0400 ; set address for #s divisible by 3

.equ OTHERADDS = 0x0600 ; set address for all other #s

ldi ZL, LOW(STARTADDS) ; ZL=0x00

ldi ZH, HIGH(STARTADDS) ; ZH=0x20

ldi R20, 0x0B ; load R20 with starting value to store (11)

ldi R21, 0x63 ; load R21 with counter value (99)

POPULATE\_LOOP:

st Z+, R20 ; Z=R20 then increment pointer

inc R20 ; increment value to be stored

dec R21 ; R21 -= 1

brne POPULATE\_LOOP ; loop until R21=0

; --- SEPARATING AND ADDING VALUES ---

ldi ZL, LOW(STARTADDS) ; ZL=0x00

ldi ZH, HIGH(STARTADDS) ; ZH=0x20

ldi YL, LOW(THREEADDS) ; YL=0x00

ldi YH, HIGH(THREEADDS) ; YH=0x04

ldi XL, LOW(OTHERADDS) ; XL=0x00

ldi XH, HIGH(OTHERADDS) ; XH=0x06

ldi R16, 0 ; initialize sum register to zero

ldi R17, 0 ; initialize sum register to zero

ldi R18, 0 ; initialize sum register to zero

ldi R19, 0 ; initialize sum register to zero

ldi R20, 99 ; load R0 with counter value

ldi R23, 0 ; used to add carry

PARSING\_LOOP:

ld R21, Z+ ; R21=Z then inc Z

mov R22, R21 ; R22=R1

DIV\_BY\_THREE:

subi R21, 0x03 ; R21 -= 3 to check for divisibility

breq DIVISIBLE ; branch if R21 = 0 (divisible by 3)

brsh DIV\_BY\_THREE ; branch if R21 >= 3, continue parsing

st X+, R22 ; store value not divisible by 3

add R18, R22 ; add value to running sum

adc R19, R23 ; add carry

rjmp CONT\_PARSE ; jump to continue parsing

DIVISIBLE:

st Y+, R22 ; store value divisible by 3

add R16, R22 ; add value to running sum

adc R17, R23 ; add carry

CONT\_PARSE:

dec R20 ; R20 -= 1

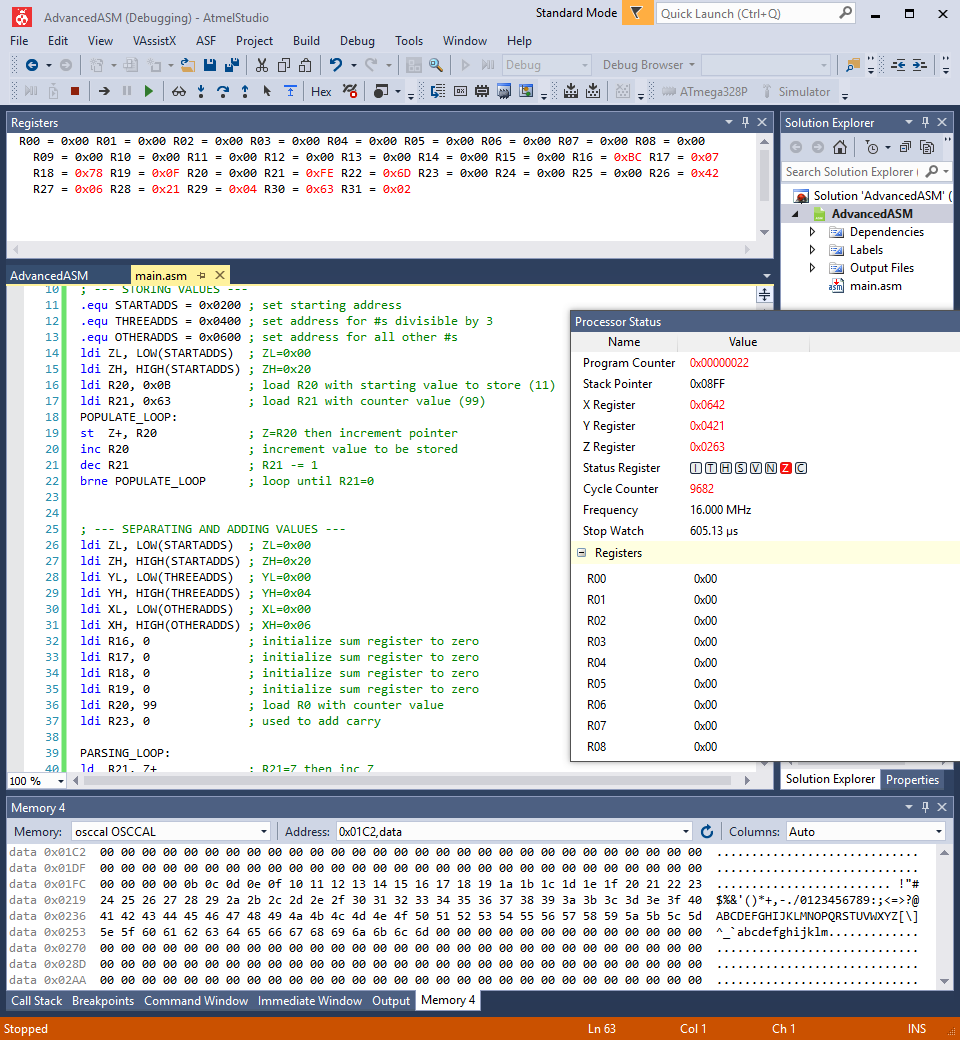
brne PARSING\_LOOP ; loop until R20=0

END: jmp END ; end of program

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

; nothing was modified from the previous assignment

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



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

https://github.com/johnsb18/ClassRepository/tree/master/DesignAssignments/DA1B

**Student Academic Misconduct Policy**

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

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

Benjamin Johnson