CPE301 – SPRING 2019

Design Assignment 1B

Student Name: Prachi Patel

Student #: 5002380222

Student Email: patelp3@unlv.nevada.edu

Primary Github address: https://github.com/prachi173/da\_sp18

Directory: https://github.com/prachi173/da\_sp18/tree/master/Design%20Assignments/DA1B

The following are required for successful completion of the design assignment:

* 1. a. AVR assembly code that has been assembled and working. Only the source files required.
  2. b. The assembly code should be well documented with explanation of every instruction.
  3. c. A word document that contains the assembly code along with the screenshots of the Atmel Studio 7 during debugging at the beginning and end of Task 1.
  4. d. Submit one solution folder, with doc and video/snapshot file. See assignment submission guidelines through github posted in the class website.

Insert initial code here

;

; cpe301\_1B.asm

;

; Created: 2/23/2019 2:01:32 PM

; Author : patel

;

; Replace with your application code

.CSEG

.ORG 0

LDI R23, 0X00 ;Load 0 into R23.

LDI XL, LOW(0X0200) ;Setting up the location for X pointer by putting lower bits of 0x0200 (00) into XL

LDI XH, HIGH(0X0200) ;and loading higher bits (02) into XH

LDI ZL, LOW(0x0400) ;Similar to X, Z is also intiated at 0x0400

LDI ZH, HIGH(0x0400)

LDI YL, LOW(0x0600) ;Z is intiated at 0x0600

LDI YH, HIGH(0x0600)

LDI R20, 100 ;Load decimal 100 or 0x64 into R20

L1:

DEC R20 ;Decrement R20 (this starts count at 99)

MOV R21, R20 ;Copy value of R20 into R21

ST X+, R20 ;Store the value in R20 at location X (starts at 0x0200) and then increment X (for the first loop, that would take X to location 0x0201)

CPI R20, 0 ;Compare R20 to 0

BRNE L3 ;If R20 is not 0, branch to label L3

JMP END ;If R20 is 0, Jump to label END

L3:

LDI R22, 3 ;Load 3 into R22

SUB R21, R22 ;Subtract R22 from R21

CPI R21, 3 ;Compare R21 to 3

BRGE L3 ;If R21 greater or equal to 3, branch to L3 (loop)

CPI R21, 0 ;If not R21 is less than 3, compare to 0

BREQ L4 ;If R21 = 0, branch to label L4

CPI R21, 0x03 ;Else compare to 3

BRLT L5 ;If less than 3, branch to label L5

L4:

ST Z+, R20 ;Store the value in R20 at location Z and then increment Z. This will store the first value of R20 that is also a multiple of 3 to 0x0400 and then change the location pointer Z to 0x0401

ADD R17, R20 ;Add R20 to the lower bit of R16:R17, that is R17.

ADC R16, R23 ;Add the carry to R16 and add R23. R23 is 0 here. Sum of multiples of 3 is now stored in R16:R17

JMP L1 ;Jump to label L1

L5:

ST Y+, R20 ;Store the value in R20 at location Y and then increment Y. This will store the values that are not multiples of 3.

ADD R19, R20 ;Add R20 to the lower bit of R18:R19, that is R19.

ADC R18, R23 ;Add the carry to R18 and add R23. R23 is 0. Sum of numbers that are NOT multiples of 3 is now stores in R18:R19.

JMP L1 ;Jump to label L1

END: RJMP END ;Once here, stay here forever

**Beginning of Debugging**

A screenshot of a social media post

Description automatically generated

**After Debugging**

(Answer)

A screenshot of a computer

Description automatically generated

**X Memory**

**A screenshot of a computer

Description automatically generated**

**Z Memory**

**A close up of a keyboard

Description automatically generated**

**Y Memory**

A close up of a keyboard

Description automatically generated

**Execution time, as seen in the screenshot above, is 614.19 microseconds with 9827 cycles at 16MHz.**

**Confirming Answer through C code.**A screenshot of a computer

Description automatically generated

**Student Academic Misconduct Policy**

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

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

Prachi Patel