CPE301 – SPRING 2019

Design Assignment 1

Student Name: Cody Jones

Student #: 5002863328

Student Email: Jonesc30@unlv.nevada.edu

Primary Github address: https://github.com/Jonesc30/Submission

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

; --- STORING VALUES ---

.equ STARTADDS = 0x0200 ; set starting address

.equ FIVEADDS = 0x0300 ; set address for #s divisible by 5

.equ OTHERADDS = 0x0500 ; set address for all other #s

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

ldi ZH, HIGH(STARTADDS) ; ZH=0x02

ldi R20, 0x0A ; load R20 with starting value to store (10)

ldi R21, 0x64 ; load R21 with counter value (100)

POPULATE\_LOOP1:

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

inc R20 ; increment value to be stored

dec R21 ; R21 -= 1

brne POPULATE\_LOOP1 ; loop until R21=0

ldi R20, 0x0A ; load R20 with starting value to store (10)

ldi R21, 0x95 ; load R21 with counter value (149)

POPULATE\_LOOP2:

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

inc R20 ; increment value to be stored

dec R21 ; R21 -= 1

brne POPULATE\_LOOP2 ; loop until R21=0

; --- SEPARATING AND ADDING VALUES ---

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

ldi ZH, HIGH(STARTADDS) ; ZH=0x02

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

ldi YH, HIGH(FIVEADDS) ; YH=0x03

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

ldi XH, HIGH(OTHERADDS) ; XH=0x05

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, 249 ; 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=R21

DIV\_BY\_FIVE:

subi R21, 0x05 ; R21 -= 5 to check for divisibility

breq DIVISIBLE ; branch if R21 = 0 (divisible by 5)

brsh DIV\_BY\_FIVE ; branch if R21 >= 5, continue parsing

st X+, R22 ; store value not divisible by 5

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 5

add R16, R22 ; add value to running sum

adc R17, R23 ; add carry

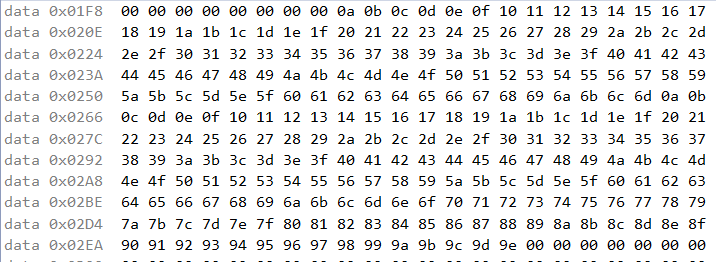
CONT\_PARSE:

dec R20 ; R20 -= 1

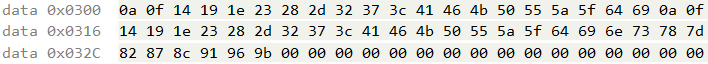
brne PARSING\_LOOP ; loop until R20=0

END: jmp END ; end of program

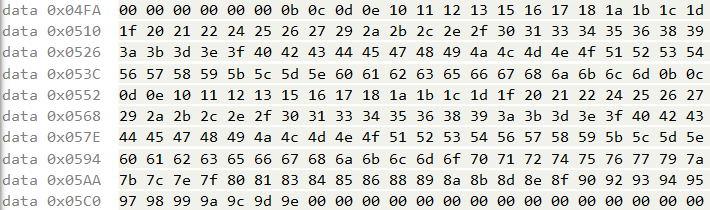
1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**
2. **SCHEMATICS**
3. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**



Data registers from 0x0200 to 0x0249 have values.



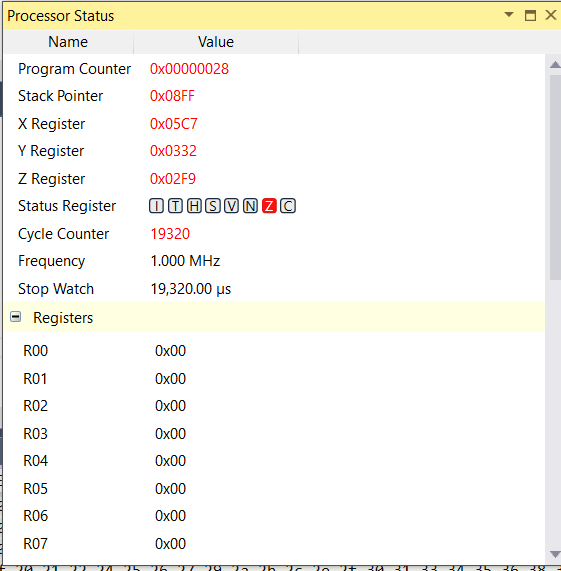
These are all the values divisible by 5 from the values above that are stored in registers 0x0300 and above.



These are the numbers that are not divisible by 5. The values are stored in the registers starting at 0x0500.



These are our sum values where the divisible sums is R17:R16 and the non-divisible are R19:R18.



This is the cycle counter and time for my code.

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

<https://github.com/Jonesc30/Submission/tree/master/DesignAssignments>

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT