CPE301 – SPRING 2021

Design Assignment 1B

Student Name: Brian Wolak

Student #: 2000509437

Student Email: wolak@unlv.nevada.edu

Primary Github address: <https://github.com/brianwolak/submission_da.git>

Directory: <https://github.com/brianwolak/submission_da/tree/main/DA_1B>

Video Link: <https://youtu.be/AjRJZLf4EMs>

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

Task 1:

The goal in Design Assignment 1B is to write an assembly program to store 112 consecutive numbers beginning at decimal number 33 into SRAM memory and then separate the EVEN and ODD numbers and store to another area in SRAM memory. The 112 numbers should begin to be stored at memory location 0x0400 while the odd and even numbers should be stored beginning at 0x0500 and 0x0600 respectively. All ODD numbers should be added together and stored in registers R18-R16 and EVEN numbers should be added and stored in R21-R19. X register will be used to add all numbers, Y register will be used to add OOD values, and Z register will be used to add EVEN values. Below we can see the assembly code, matching C++ code and overall outputs along with the execution time @ 16MHz as well as the total number of clock cycles.

Assembly Code:

.equ STARTADDS = 0x0400 ;address begin of all numbers

.equ ypoint = 0x0500 ;address begin of odd numbers

.equ zpoint = 0x0600 ;address begin of even numbers

ldi r16, 0 ;used to store low bits odd numbers

ldi r17, 0 ;used to store mid bits odd numbers

ldi r18, 0 ;used to store high bits odd numbers

ldi r19, 0 ;used to store low bits even numbers

ldi r20, 0 ;used to store mid bits even numbers

ldi r21, 0 ;used to store high bits even numbers

ldi r22, 0x71 ;total numbers used (112 in decimal) +1 for proper counter function

ldi r23, 0x21 ;start of numbers to add

ldi r24, 0x01 ;used to set carry flag in add/subtract

ldi xh, HIGH(STARTADDS) ;storing all numbers using xpointer

ldi xl, LOW(STARTADDS)

ldi yh, HIGH(ypoint) ;storing odd numbers using ypointer

ldi yl, LOW(ypoint)

ldi zh, HIGH(zpoint) ;storing even numbersing using zpointer

ldi zl, LOW(zpoint)

START:

cpi r22, 0 ;check to see if counter is zero

breq END ;if counter is zero end program

st x+, r23 ;store number from r23 to X register

mov r25, r23 ;temporary store of current number

sub r22, r24 ;decrement counter

ror r25 ;odd/even check

brcs ODD ;if r25 odd branch to ODD

rjmp EVEN ;if r25 even jump to EVEN

ODD:

st y+, r23 ;store odd number to Y register

add r16, r23 ;add current number to odd sum

brcs ODDOVERFLOW ;if r27 rollover branch to ODDOVERFLOW

inc r23 ;increment current number

jmp START ;return to start

ODDOVERFLOW:

add r17,r24 ;increment middle bit

brcs ODDOVERFLOW2 ;branch to ODDOVERFLOW2 if carry set

inc r23 ;increment current number

rjmp START

ODDOVERFLOW2:

add r18, r24 ;increment highest bit values for odd

inc r23 ;increment current number

rjmp START ;return to start

EVEN:

st z+, r23 ;store even number to X register

add r19, r23 ;add current number to even sum

brcs EVENOVERFLOW ;if r23

inc r23 ;increment current number

jmp START ;back to start

EVENOVERFLOW:

add r20, r24 ;increment middle even bits

brcs EVENOVERFLOW2 ;if overflow set branch to EVENOVERFLOW2

inc r23 ;increment curent number

rjmp START ;return to start

EVENOVERFLOW2:

add r21, r24 ;increment highest even bits

inc r23 ;increment current number

rjmp START ;return to start

END:

rjmp END ;program ends

C++ Code:

Graphical user interface, text, application, email

Description automatically generated

C++ Code Verification Output:

**Text

Description automatically generated**

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

**Table

Description automatically generated with medium confidence**

**Here we see the register and memory values before the start of the final debugging session**

**Table

Description automatically generated A picture containing table

Description automatically generated**

**Above on the right we can see the R18-R16 registers contain the 0x0013D1 value or 5073 in decimal form as the summation of ODD numbers, and R21-R19 contains the value 0x001378 or 4984 in decimal which is the summation of EVEN numbers. On the left we can see all numbers have been added starting from 0x0400 with only ODD and EVEN values added to 0x0500 and 0x0600 respectively.**

**This program used a total of 2072 clock cycles with a processing time of 129.5us at 16MHz**

1. **GITHUB LINK OF THIS DA**

<https://github.com/brianwolak/submission_da/tree/main/DA_1B>