CPE301 – SPRING 2019

Design Assignment 1

Student Name: Matija Tomljenvic

Student #: 1012254710

Student Email: tomljeno@unlv.nevada.edu

Primary Github address: https://github.com/matcroatia

Directory:

;

; AssemblerApplication2.asm

;

; Created: 2/9/2019 4:03:33 PM

; Author : Matija Tomljenovic

;

.org 0x000

ldi r25,0x05 //register to store 16bit number

ldi r24,0xDC //register to store 16bit number

ldi r20,0 //register used to store the 24bit result

ldi r19,0 //register used to store the 16bit multiplicand

ldi r18,0 //register used to store the 16bit multiplicand

ldi r22,5 //register used to store the multiplier and counter to decriment

ldi r27,0 //used to compare the decrimented value for loop branching

start: //start loop used for addition

add r18,r24 //addition performed for the 16bit multiplicand and stored into r18

adc r19,r25 //carry addition performed for the 16bit multiplicand and stored into r19

adc r20,r27 //carry addition performed to sore the 24bit result

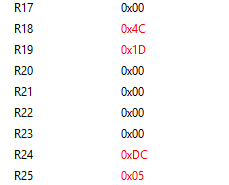
dec r22 //decriment used to control loop

cp r22,r27 //compare used for comparing the decrimented multiplier and 0 stored in r27

brne start //branch used to repeat loop untill r22 and r27 are equal.

done: //used to extit loop once decriment reaches 0

jmp done



/\*

\* AssemblerApplication1.asm

\*

\* Created: 2/9/2019 3:40:36 PM

\* Author: Matija Tomljenovic

\*/

ldi r18,0x77 //higher end byte of multiplicand

ldi r19,0x5b //lower end byte of multiplicand

ldi r20,0x7b //multiplier to be used

ldi r22,0 //register that will store lower end byte of anwser

ldi r23,0 //register used to store the middle byte of anwser

ldi r24,0 //register used to store higher end byte of anwser

mul r19,r20 //multiplication to multiply lower end bytes

mov r26,r0 //copy the lower end byte of result into r26

mov r27,r1 //copy the lower end byte of result into r27

mul r18,r20 //multiplication to multiply higher end byte

mov r28,r0 //copy the higher end byte of result into r28

mov r29,r1 //copy the higher end byte of result into r29

add r22,r26 //add the lower bytes of the result and store into r22

add r23,r27 //add the middle bytes of the result and store into r23

add r23,r28 //add the high bytes of the result and store into r23

add r24,r29 //add the last byte of result and sore into r24

done:

break

