CPE301 – SPRING 2019

Design Assignment 1A

Student Name: Bruce Moquete

Student #: 1013123266

Student Email: moquete@unlv.nevada.edu

Primary Github address:

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**

* Atmel Studio 7.0 Simulator

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

;

; DA1.asm

;

; Created: 9/14/2019 7:35:40 PM

; Author : bruce

; Description: A small assembly program that multiplies through repeated addition using a

; 2-Byte multiplicand, a 2-Byte Multiplier and a 4-byte storage for the product.

ldi r25, 0b11001000 ; Initializes the register w/ Multiplicand's Lower Byte of the number 456

ldi r24, 0b00000001 ; Initializes the register w/ Multiplicand's Higher Byte of the number 456

ldi r23, 0b01001000 ; Initializes the register w/ Multiplier's Low Byte of the number 72

ldi r22, 0b00000000 ; Initializes the register w/ Multiplier's High Byte of the number 72

ldi r20, 0 ; Initializes for Product Low byte 1

ldi r19, 0 ; Initializes for Product High Byte 1

ldi r18, 0 ; Initializes for Product Low Byte 2

ldi r17, 0 ; Initializes for Product High Byte 2

ldi r16, 0 ; Register used to hold zero value for adc instructions.

start: ; Main function of this code. Repeatedly adds to multiply until the multiplier decrements to zero.

add r20, r25 ; Adds the lower byte of the multiplicand into r20

adc r19, r24 ; Adds the higher byte of the multiplicand into r19 w/ carry

adc r18, r16 ; Adds zero to register r18 with a carry

adc r17, r16 ; Adds zero to register r17 with a carry

call decrement ; Calls the decrement function to reduce a two byte number by one.

jmp start ; jumps back to start

decrement: ; Function used to decrement a 2-Byte number used as the Multiplier.

dec r23 ; Decrements the lower byte of the multiplier.

cpi r23, 0 ; Compares the contents of R23 with the number 0.

breq end ; Branches to the End function if R23 equals 0.

sbci r22, 0 ; Subtracts the higher byte of the multiplier with an immediate of 0 and the carry.

ret ; Returns to Start function where it was called.

end: ; Ending function that keeps the code in an infinite loop of nop.

nop ; No op instruction.

jmp end ; Jumps back to the beginning of the End function.

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

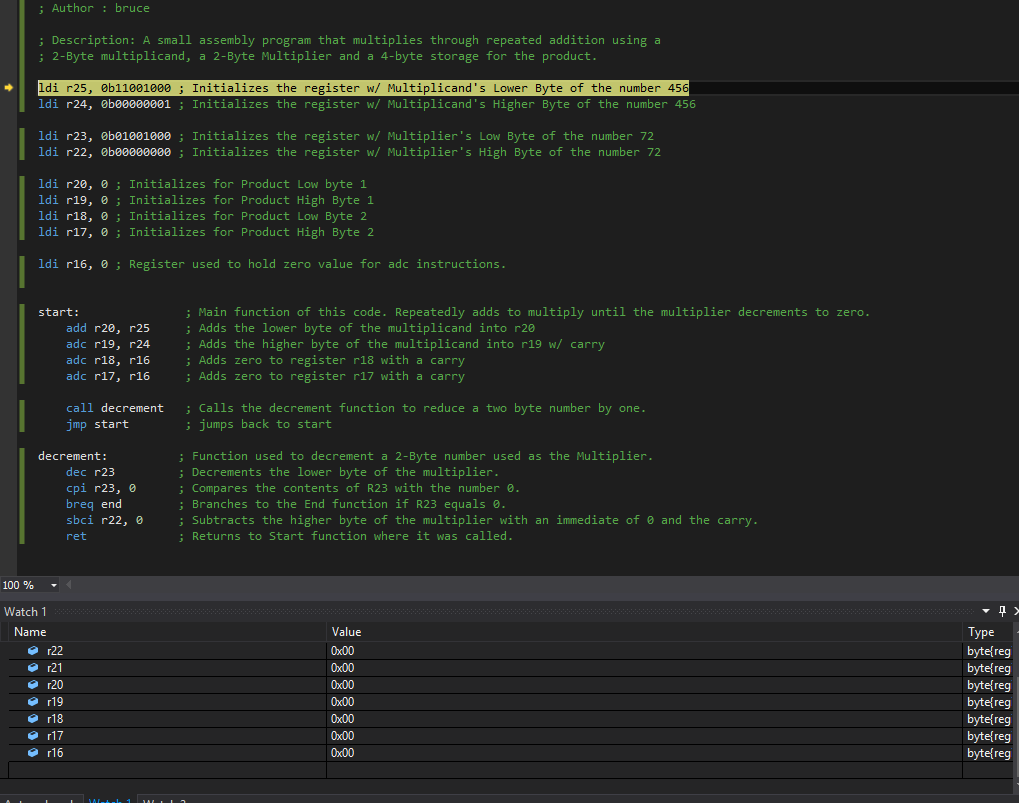
* N/A

1. **SCHEMATICS**

* N/A

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

* Beginning of Debugging:



* End Results:



* End Results Comparison From Calculator:



* Task 4: 85.63 micro Seconds

1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

* N/A

1. **VIDEO LINKS OF EACH DEMO**

* N/A

1. **GITHUB LINK OF THIS DA**

**Student Academic Misconduct Policy**

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

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

Bruce Moquete