CPE301 – SPRING 2019

Design Assignment 1A

Student Name: Armon Latifi

Student #: 2000698173

Student Email: latifa1@unlv.nevada.edu

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

Directory: https://github.com/armonlatifi/sub\_da/tree/master/DA1A

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**

List of Components used:

* + Assembler
  + Simulator
  + Debugger

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

There was no initial code provided, only used a standard assembly file template on Atmel Studio 7.

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

Assembly code:

;

; DA1.asm

; PART A

; Created: 2/16/2019 12:32:58 PM

; Author : Armon Latifi

;

.org 0x0000

start:

   LDI R25, 0x11 ;in the high part of the 16-bit multiplicand, store 17

LDI R24, 0x00 ;in the low part of the 16-bit multiplicand, store 0

LDI R22, 0x10 ;8-bit multiplier is loaded with value 16

;at this point, 16-bit multiplicand value --> 4352

;likewise, 8-bit multiplier value --> 16

;theoretically, result should be 4352 x 16 = 69632

LDI R26, 0  ;initialize R26 to zero as the carry

MOV R17, R22  ;iteration is stored

INC R17  ;increment then compare to breq

LDI R18, 0  ;initialize R18

LDI R19, 0  ;initialize R19

LDI R20, 0  ;initialize R20

CPI R22, 0  ;compare R22 to zero, because if it is zero, then the answer would be zero,

 ;which is already loaded into R20:R19:R18

BREQ done  ;if R22 equals zero, operation is finished

multiply:

DEC R17  ;decrement the iteration

BREQ DONE  ;if equal to zero, jump to done

ADD R18, R24  ;add the low part of the 16-bit multiplicand with R18

ADC R19, R25  ;add the high part of the 16-bit multiplicand with R19, carry if needed

ADC R20, R26  ;add the second potential carry bit to R20

JMP multiply  ;loop back to multiply

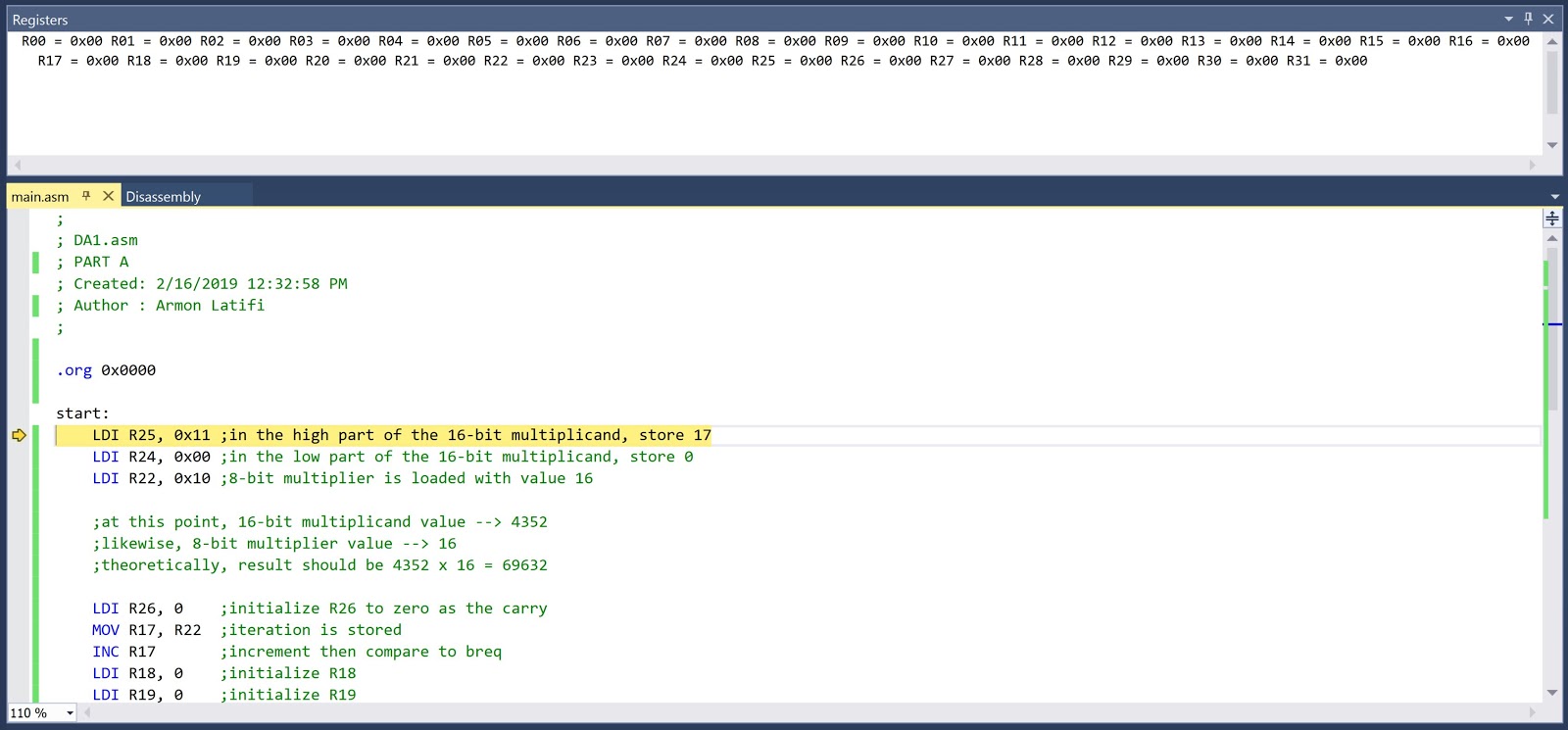
done:

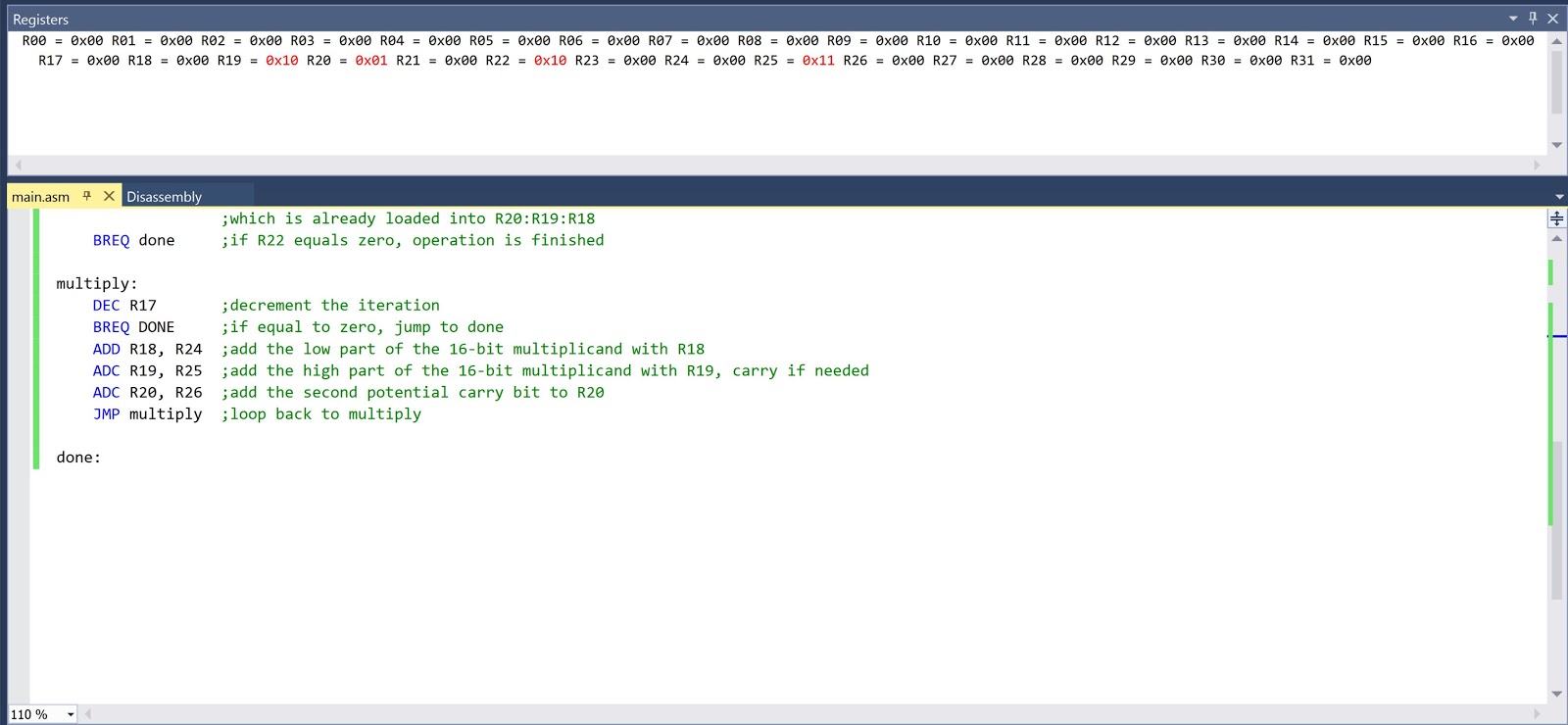
1. **SCHEMATICS**

N/A

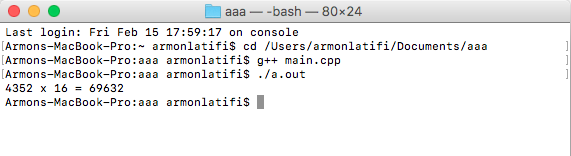
1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

In debugging mode, I set a breakpoint at the end of the program to view the final values of the registers.





Here, the final result (stored in R20:R19:R18) is equal to 0x011000. This, converted to decimal, equals 69632. The multiplier and the multiplicand were assigned as 16 and 4352, respectively. Multiplying 4352 and 16 in a C++ program yielded 69632.



For this reason, the assembly program, using iterative addition, is a valid, verified method of multiplication.

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

N/A

1. **VIDEO LINKS OF EACH DEMO**

N/A

1. **GITHUB LINK OF THIS DA**

https://github.com/armonlatifi/sub\_da/tree/master/DA1A

**Student Academic Misconduct Policy**

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

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

Armon Latifi