CPE301 – SPRING 2019

Design Assignment 1

Student Name: Geovanni Portillo

Student #: 8000603824

Student Email: portig1@unlv.nevada.edu

Primary Github address: https://github.com/portig1/submissions\_E.git

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**

No components used

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

;

; Assignment1.asm

;

; Created: 2/9/2019 1:21:01 PM

; Author : gausp

;

.include<m328pbdef.inc>

.cseg

.org 0x00

Task1: ;Itterative addition to multiply a 8-bit number to a 16-bit number

LDI R22, 50 ;Setting 8-bit multipler in R2, range is from 0-255

LDI R24, 50 ;Setting lower 8-bits of 16-bit multiplicand

LDI R25, 0x00 ;Setting higher 8-bits of 16-bit multiplicand

AddLoop:

BREQ Task2 ;Checks to see if the counter (

ADD R18, R24 ;Adds multiplicand to result register, specifically the lowest 8-bits when considering that result as 24-bits from R18 to R20

ADC R19, R25 ;Adds next 8-bits of multiplicand to the register as well as adding a carry onto it, if there is one

ADC R20, R12 ;R12 remains as 0x00 and is used to add a carry with the highest 8-bits should there be one from adding R25 to R19

DEC R22 ;Uses the multiplier as a counter for how many times the multiplicand needs to be added to itself

jmp AddLoop

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

Task2: ;Verify the algorithm using a mul instruction

LDI R16, 50 ;Setting test multiplicand values

LDI R17, 50

MUL R16, R17

MOV R10, R0 ;Result stored in R10:R11

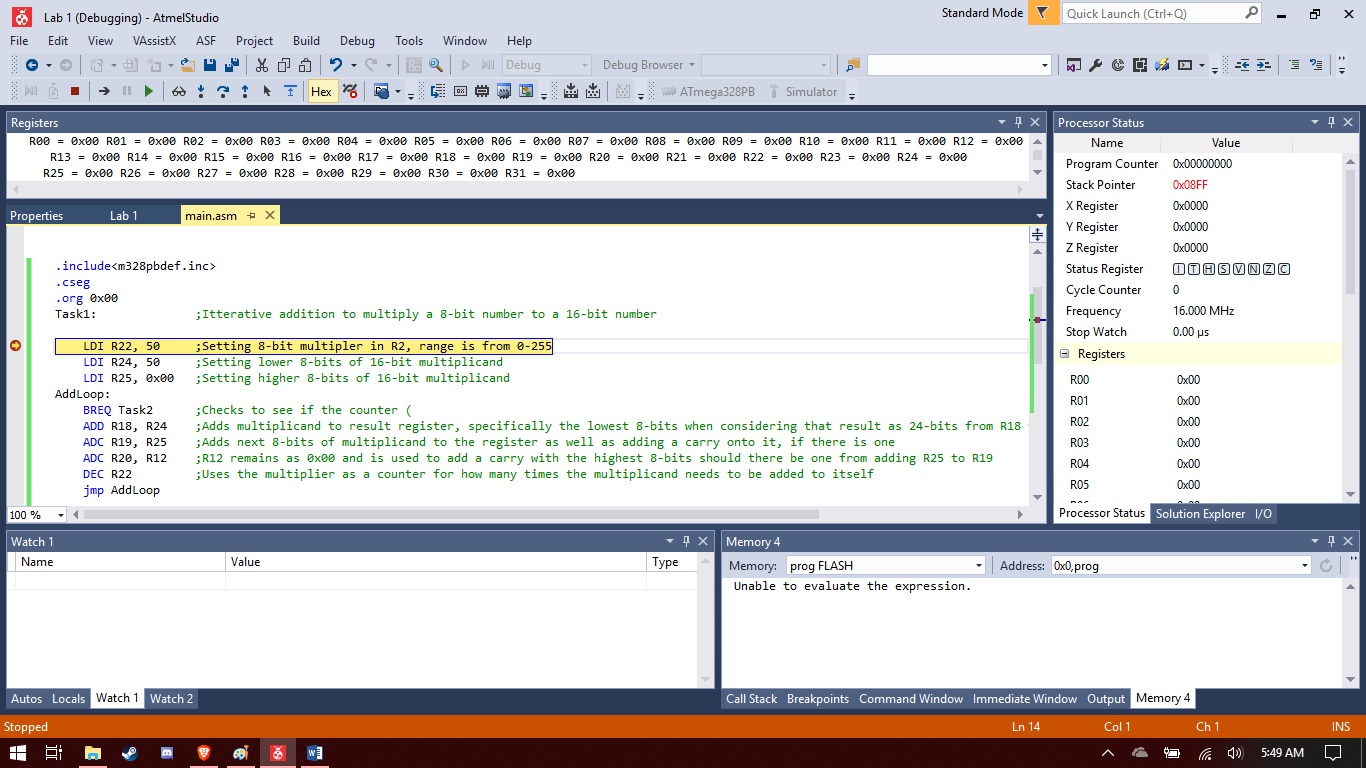
MOV R11, R1

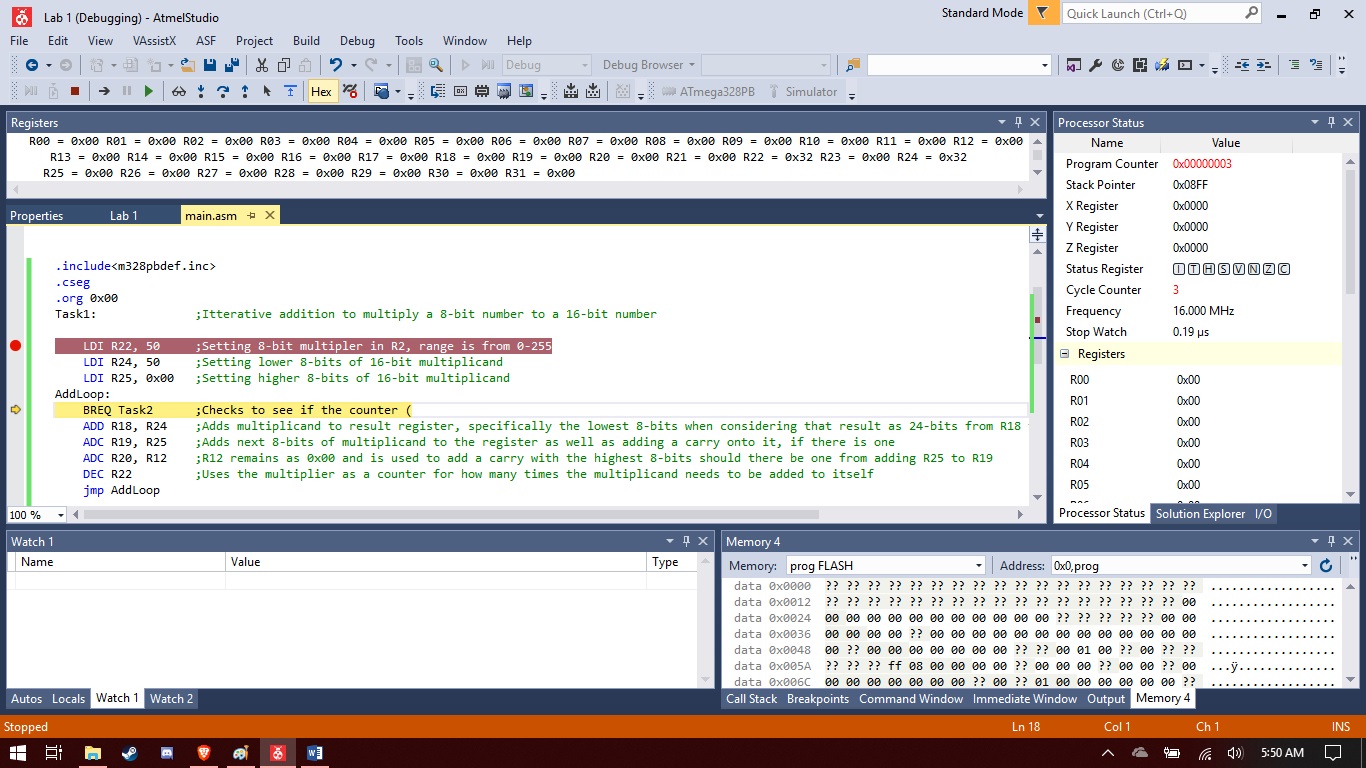
end: rjmp end

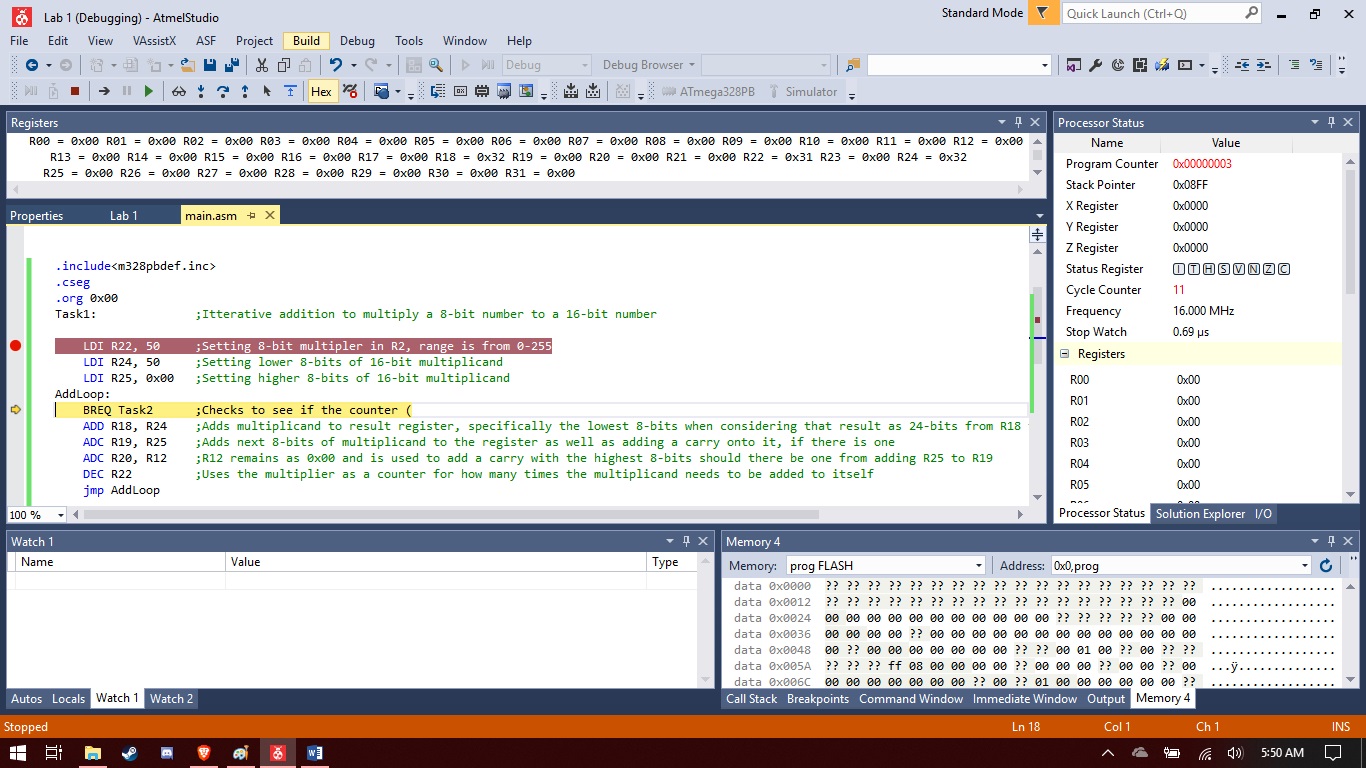
1. **SCHEMATICS**

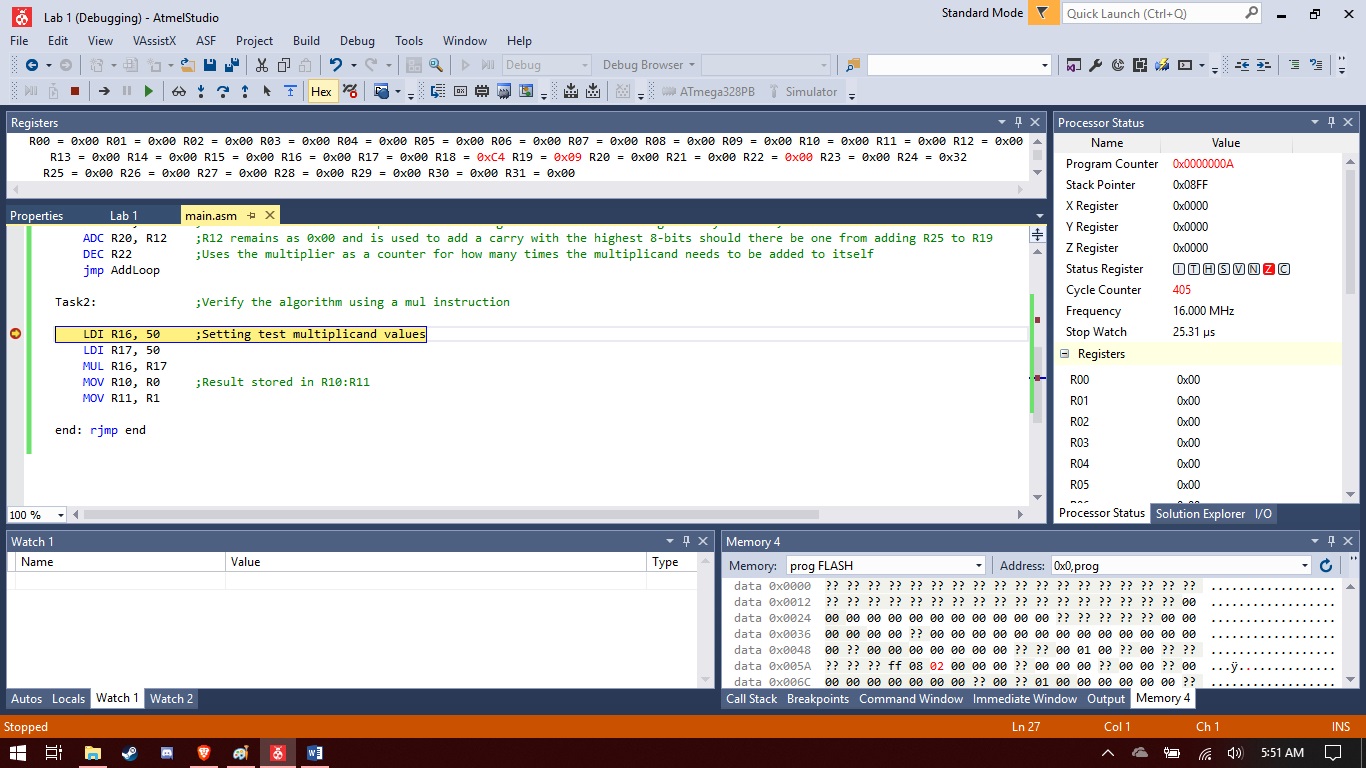
No schematics

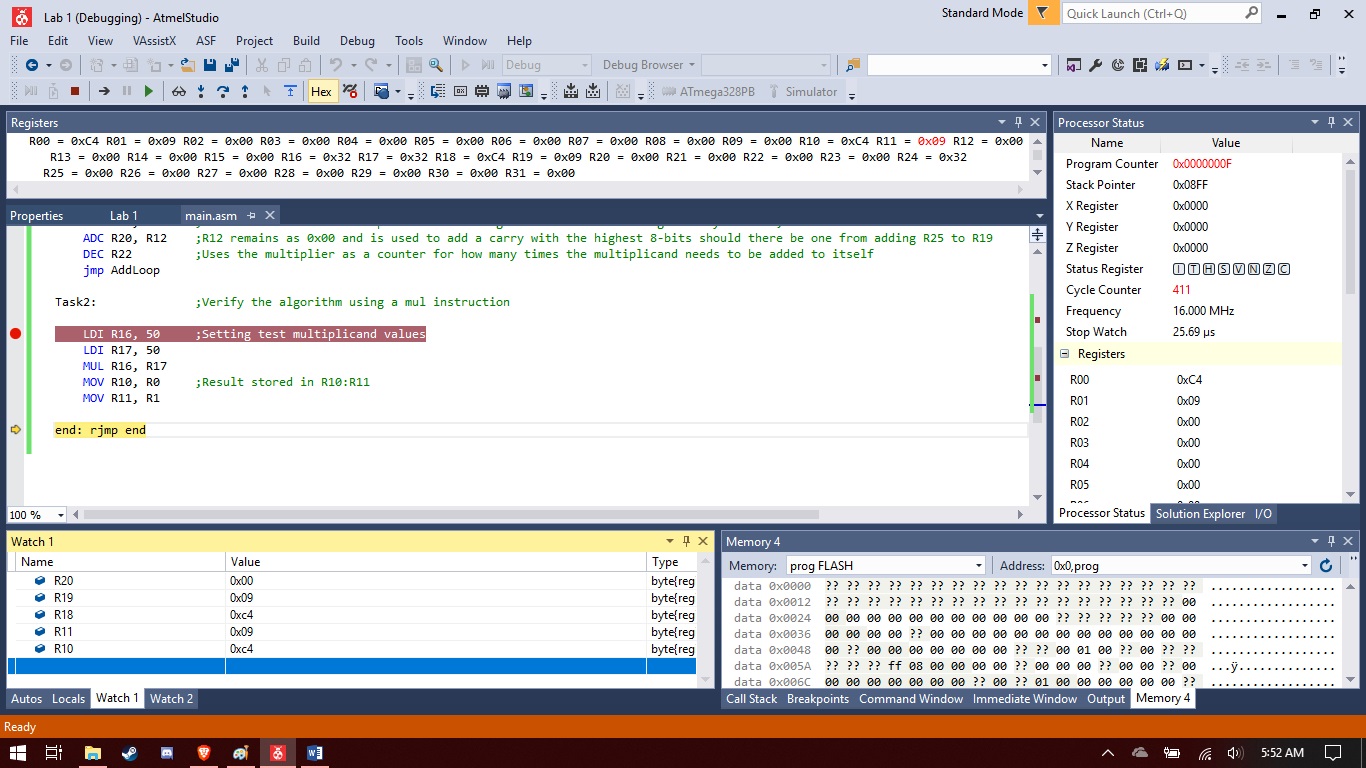
1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

Simulation at initial state

Simulation after R22, R24 and R25 have been set. Three clock cycles have passed at 16MHz and time is at 0.19us

After one iteration of the algorithm, we are at 11 cycles with 0.69us though this is including the initial setup for the registers. Taking that into account, the iteration of the algorithm itself took 8 cycles and 0.5us.

For the specific example of multiplying 50 by 50, the task took in total 405 cycles and 25.31us.



Verifying the result by using the mul instruction. Registers 18 through 20 have the result 0x0009c4 and the result from the mul instruction has a matching result of 0x09c4

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

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT