Nathan Hanuscin

CPE301 – SPRING 2018

Design Assignment X

**DO NOT REMOVE THIS PAGE DURING SUBMISSION:**

The student understands that all required components should be submitted in complete for grading of this assignment.

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SUBMISSION ITEM** | **COMPLETED (Y/N)** | **MARKS**  **(/MAX)** |
| 1 | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |
| 2. | INITIAL CODE OF TASK 1/A |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E |  |  |
| 4. | SCHEMATICS |  |  |
| 5. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 5. | SCREENSHOT OF EACH DEMO |  |  |
| 6. | VIDEO LINKS OF EACH DEMO |  |  |
| 7. | GOOGLECODE LINK OF THE DA |  |  |
|  |  |  |  |
|  |  |  |  |

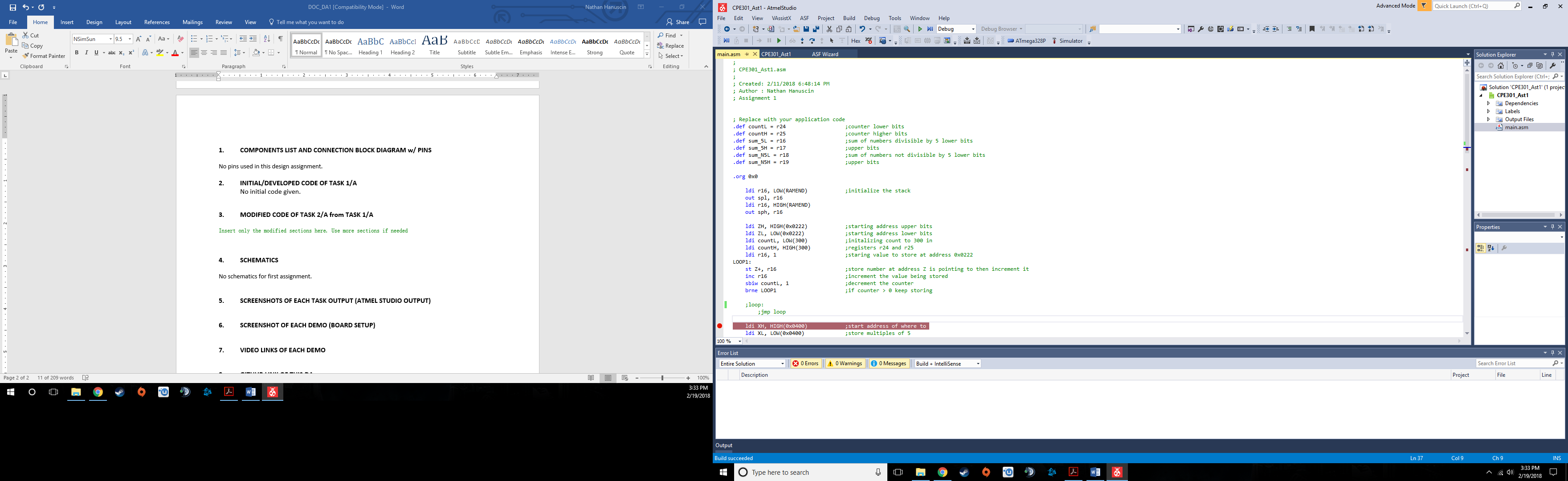
1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

No pins used in this design assignment.

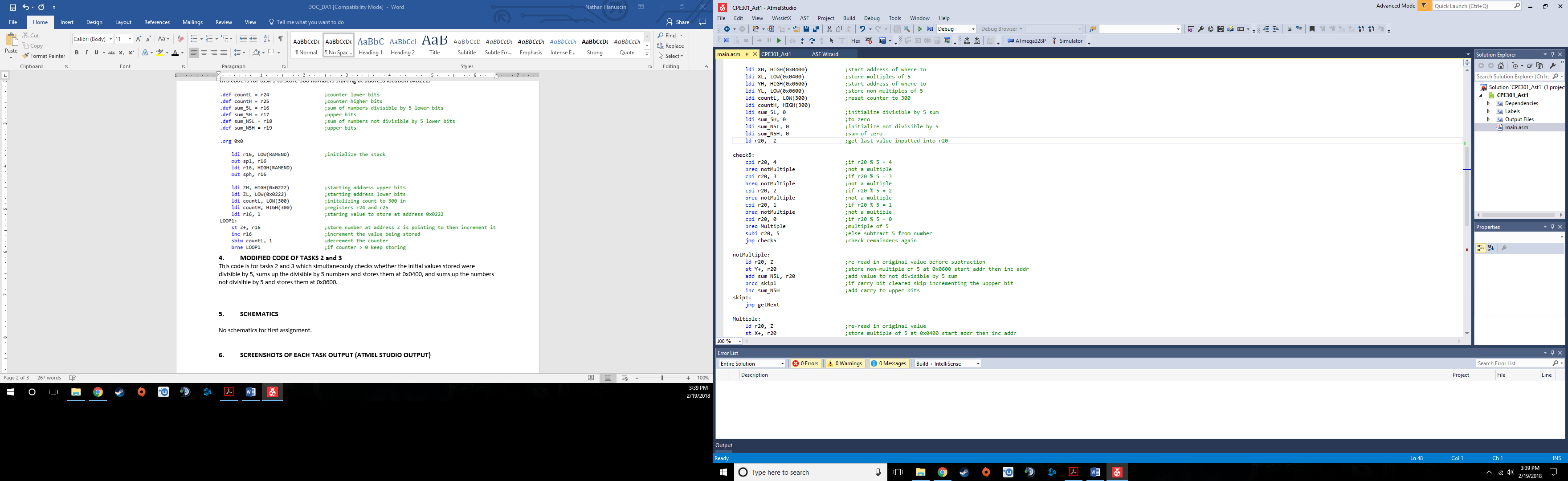
1. **INITIAL/DEVELOPED CODE OF TASK 1/A**

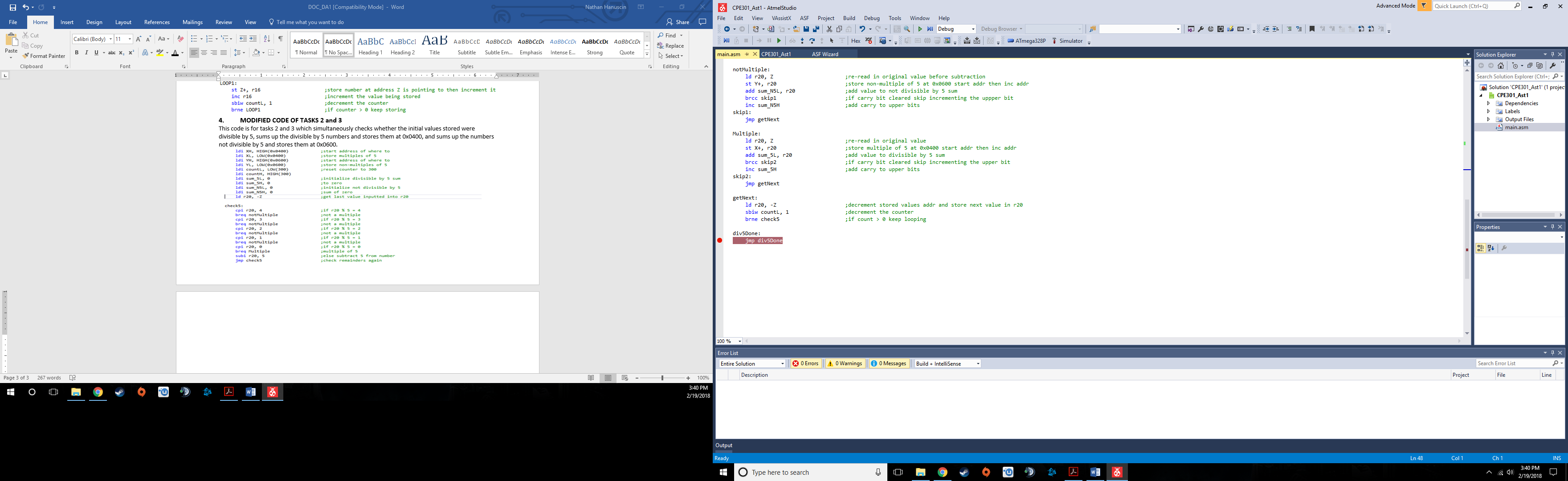
No initial code given.

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

This code is for task 1 to store 300 numbers starting at address location 0x0222.

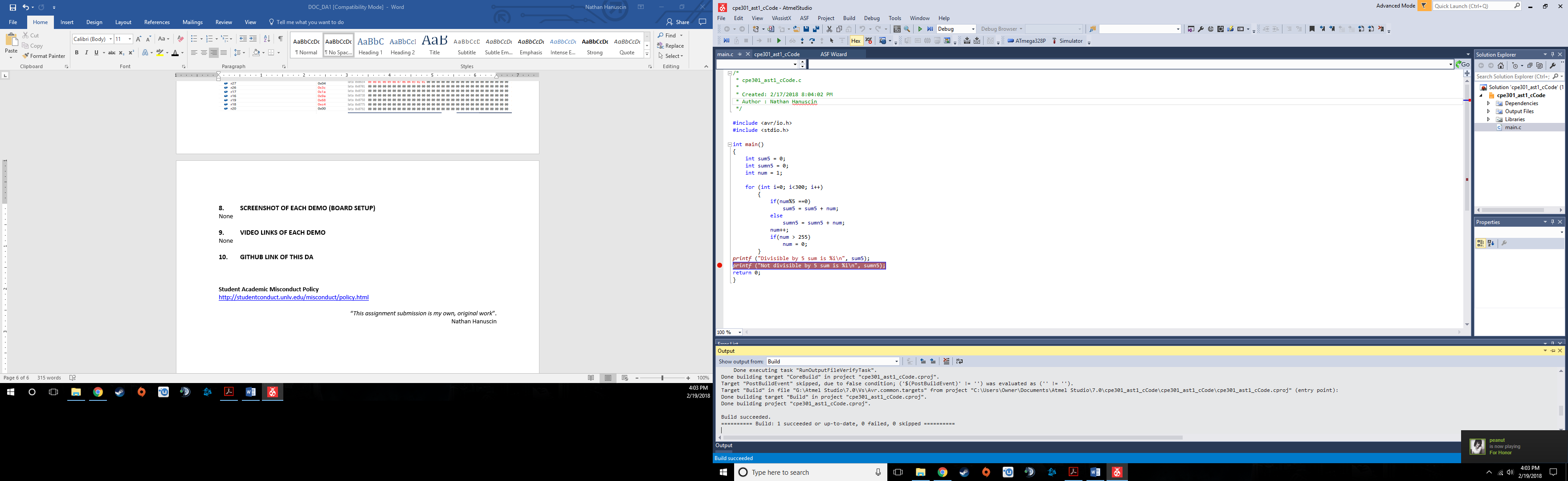
**4. MODIFIED CODE OF TASKS 2 and 3/B**

This code is for tasks 2 and 3 which simultaneously checks whether the initial values stored were divisible by 5, sums up the divisible by 5 numbers and stores them at 0x0400, and sums up the numbers not divisible by 5 and stores them at 0x0600.



**5. MODIFIED CODE FOR TASK 4/D**

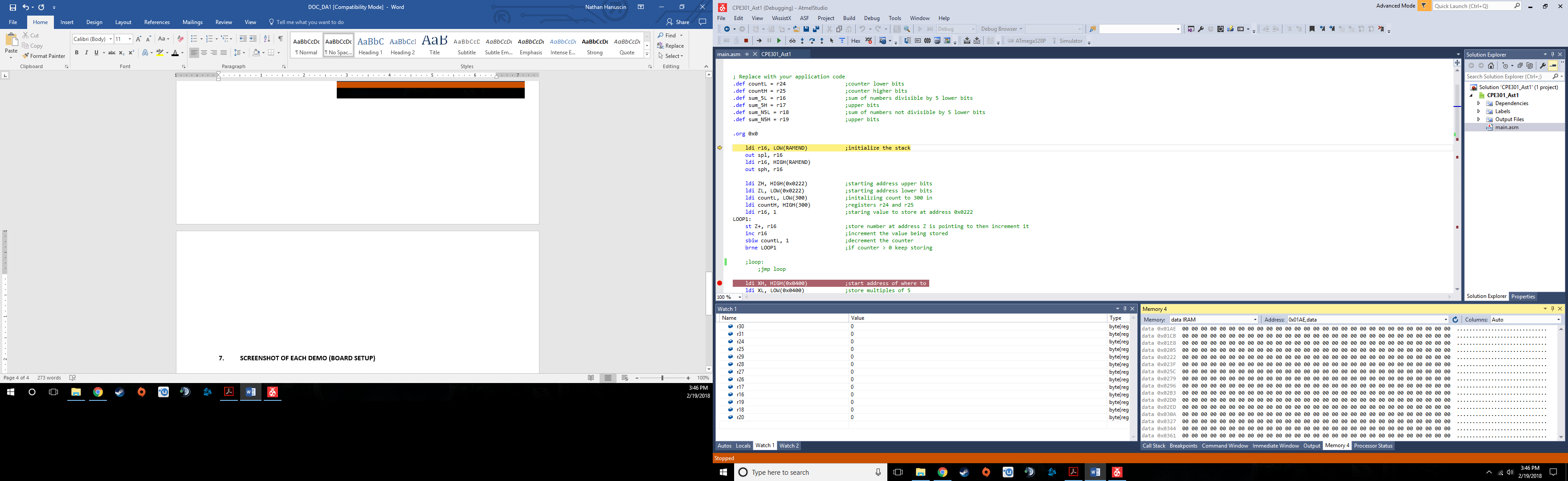
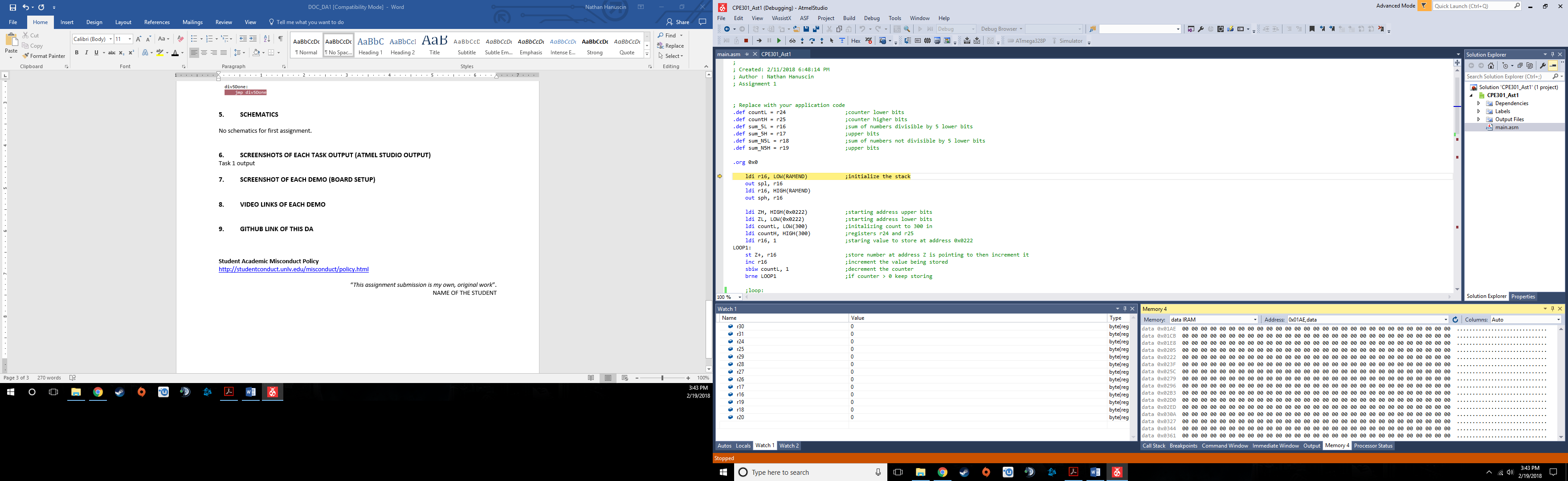
This c code checks to make sure the sums were calculated properly form the assembly code.

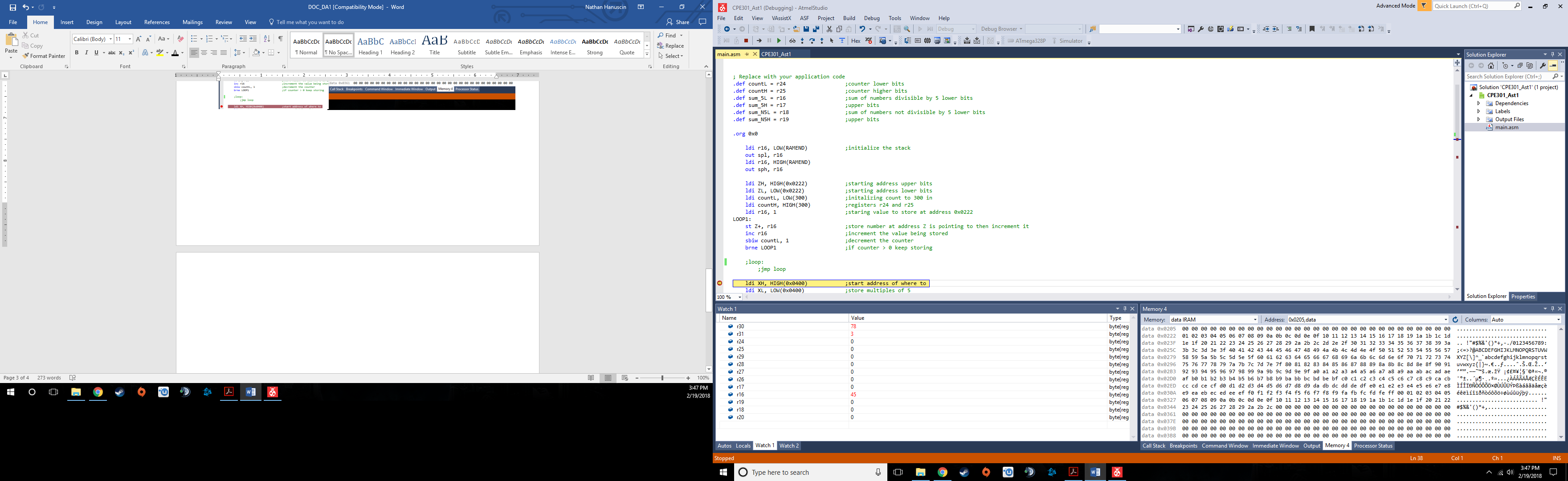
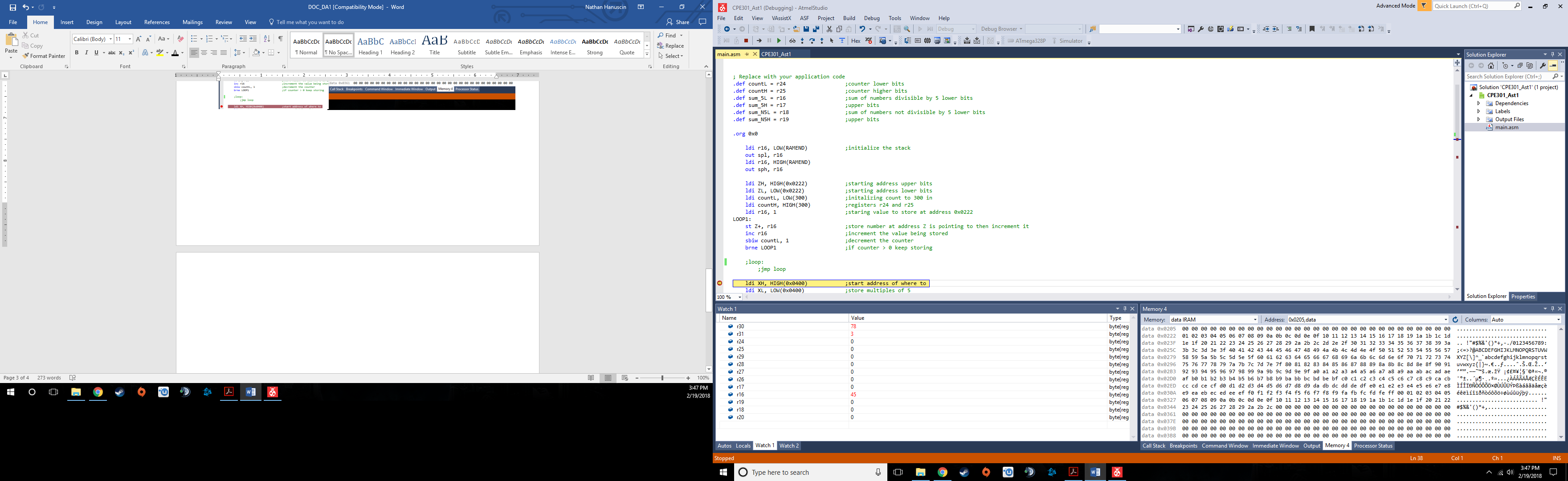


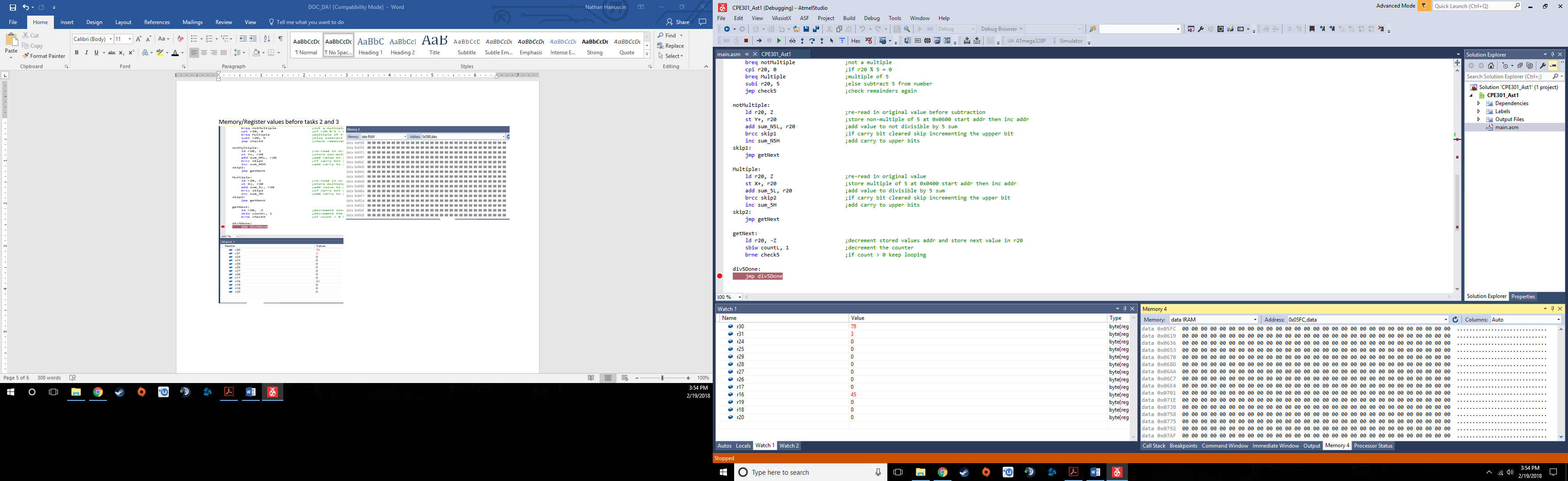
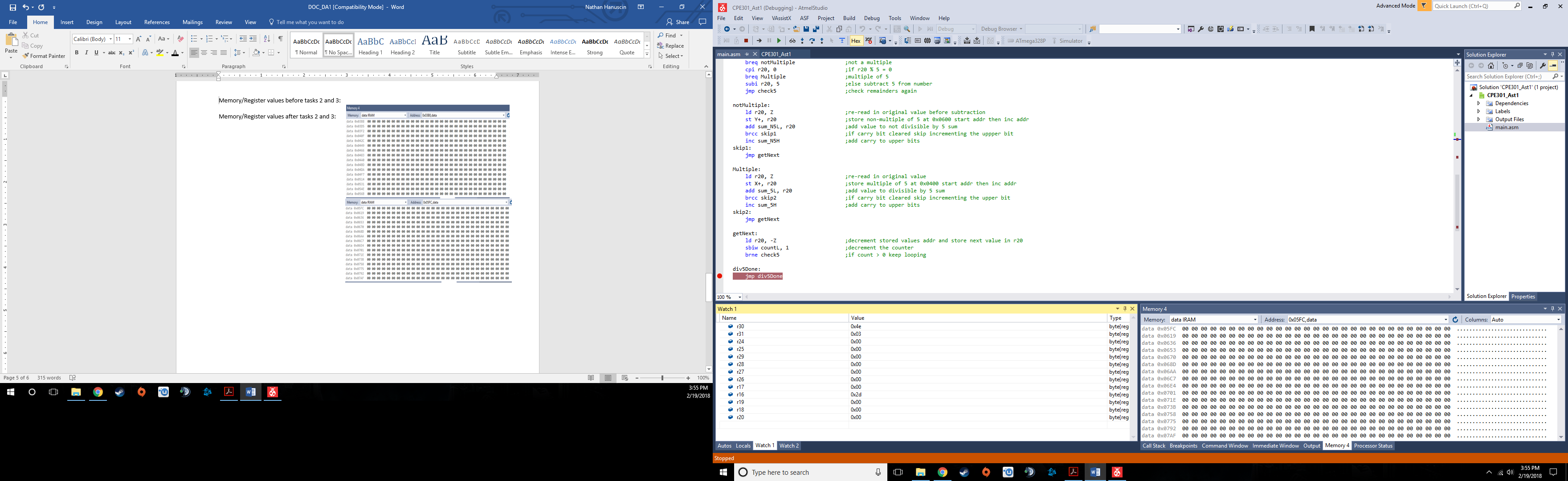
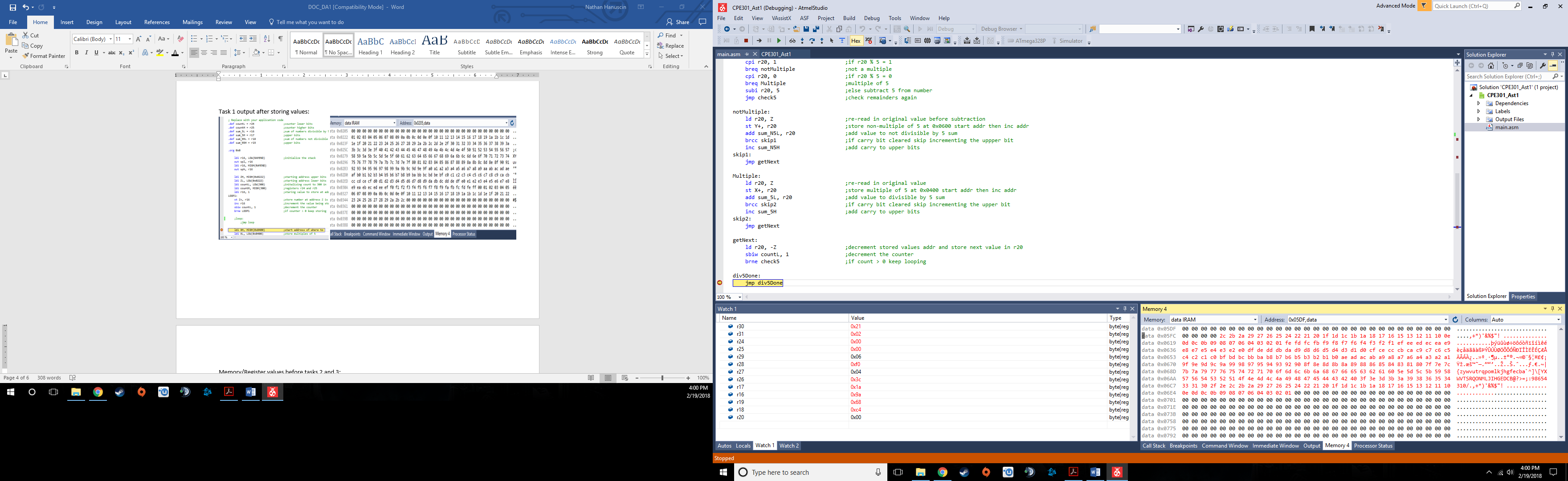
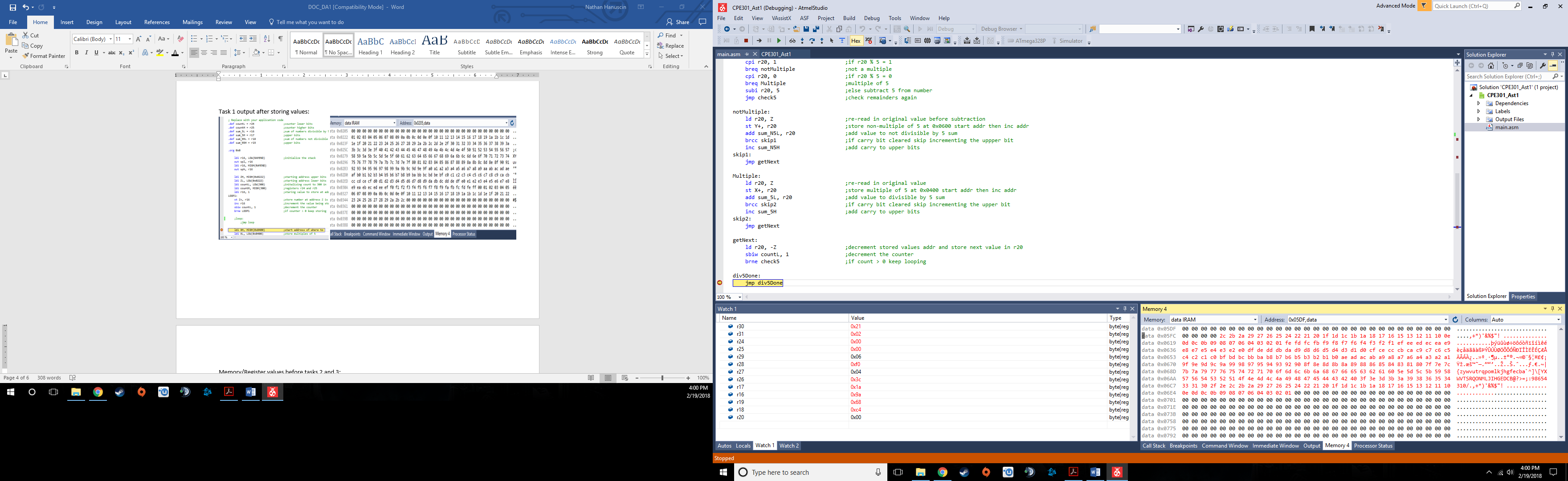
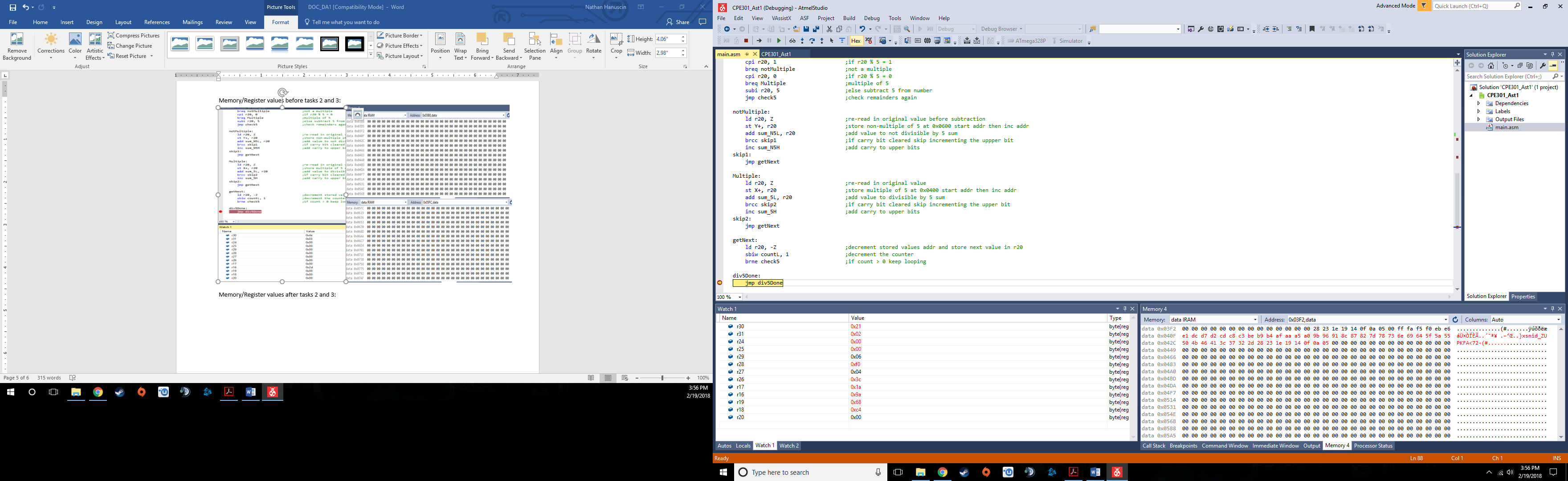
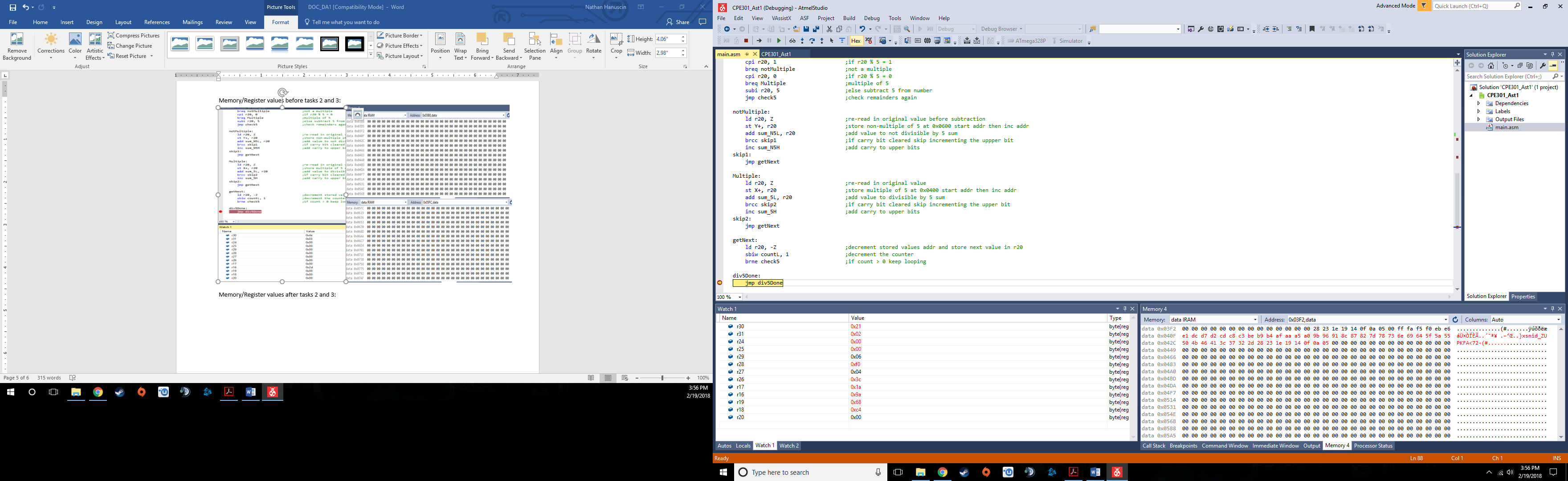
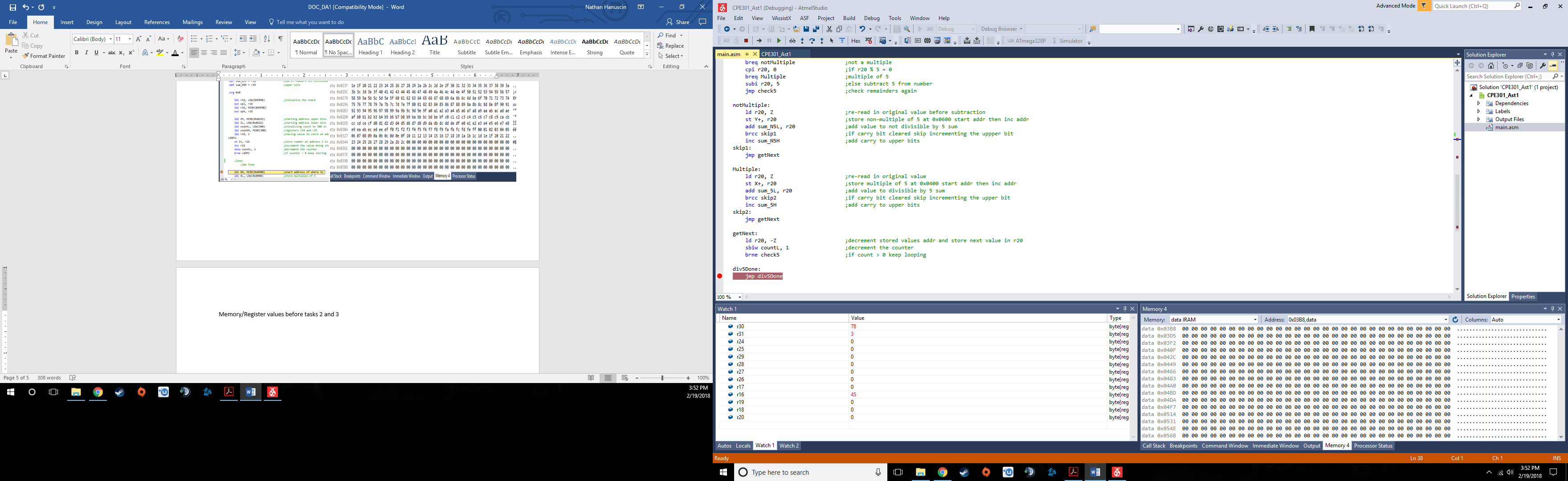
1. **SCHEMATICS**

No schematics for first assignment.

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

Task 1/A output before storing values:

Task 1/A output after storing values:

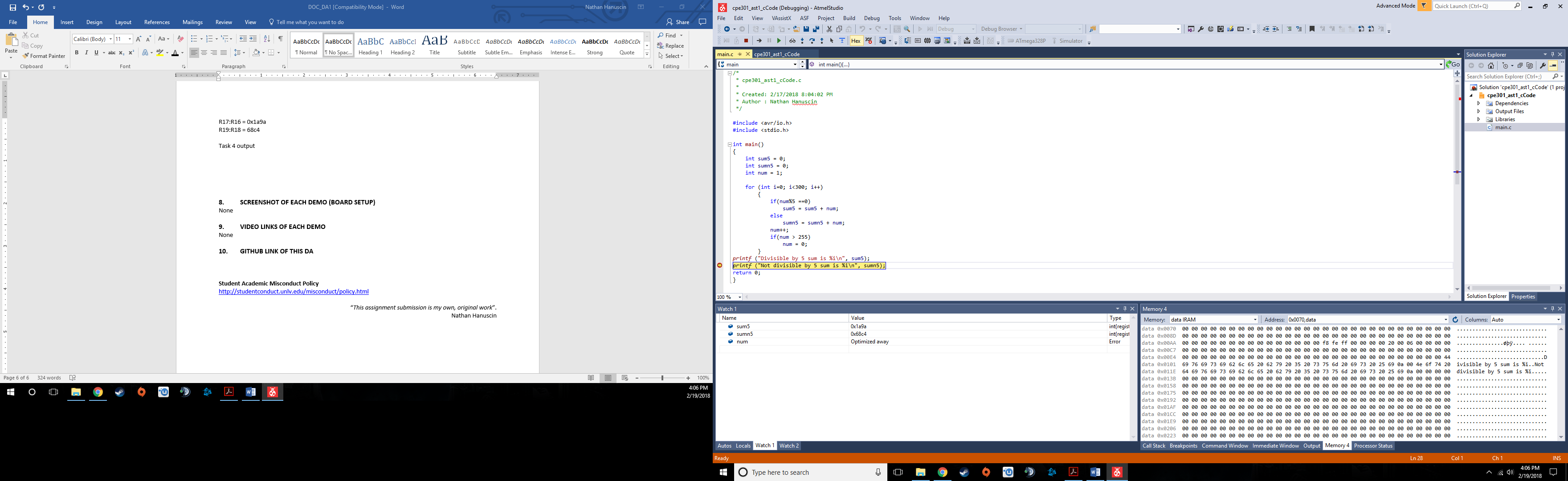
Memory/Register values before tasks 2 and 3/B:

Memory/Register values after tasks 2 and 3/B:

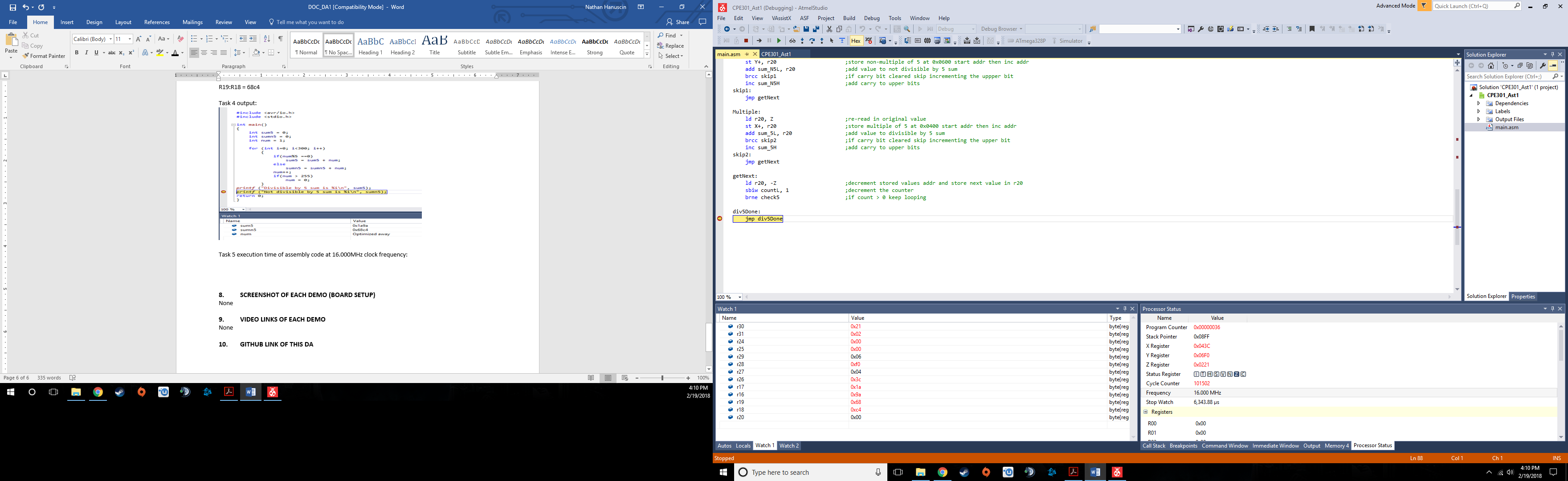
R17:R16 = 0x1a9a

R19:R18 = 68c4

Task 4/C output:



Task 5/E execution time of assembly code at 16.000MHz clock frequency:



At 16MHz, the algorithm takes 6,343.88 micro seconds.

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

None

1. **VIDEO LINKS OF EACH DEMO**

None

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”.

Nathan Hanuscin