Benjamin Linam

[Bml42@students.uwf.edu](mailto:Bml42@students.uwf.edu)

EEL 4744L: Microprocessor Applications Laboratory

Lab 3: Writing and Testing a Simple Program

2/7/2018

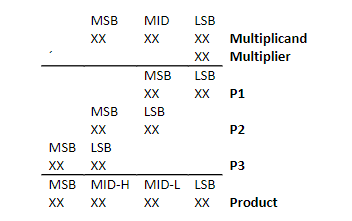
**Objective**

Introduce students to writing and testing a program in HC11 assembly language and using the BUFFALO I/O routines to display/verify the results.

**Introduction/Background/Theory**

This lab required construction of an assembly program which would compute the 4-byte product of a 3-byte number and a 1-byte number. At first glance, it appeared that a singular usage of the MUL command would provide the desired result; however, MUL is restricted by only being able to multiply two 2-byte numbers saved in both registers A and B. Since one of the numbers, the multiplicand, will be 3-bytes long, it is impossible to utilize only one MUL command to compute the product. As such, a complex program must be constructed to utilize MUL several times to multiply each individual byte of the multiplicand and arrive at the desired value.

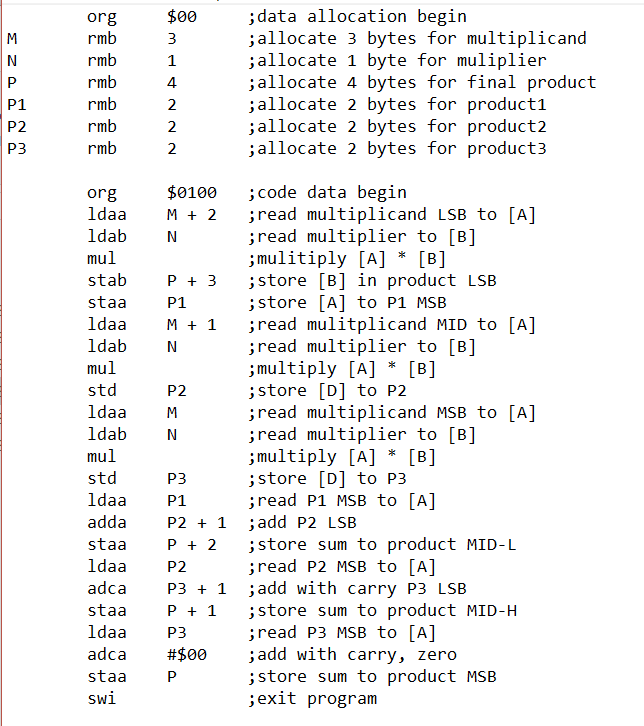
Provided for the lab, was the following algorithm which lists the operations required to compute the product:



**Figure 1**: 3-byte by 1-byte multiplication algorithm

**Procedure**

1. By using the algorithm shown in Figure 1, the following assembly language program was designed to formulate the 4-byte product ([A], [B], and [D] represent accumulators A, B, and D respectively).



**Figure 2**: Assembly language code designed to compute the product of a 3-byte number and a 1-byte number

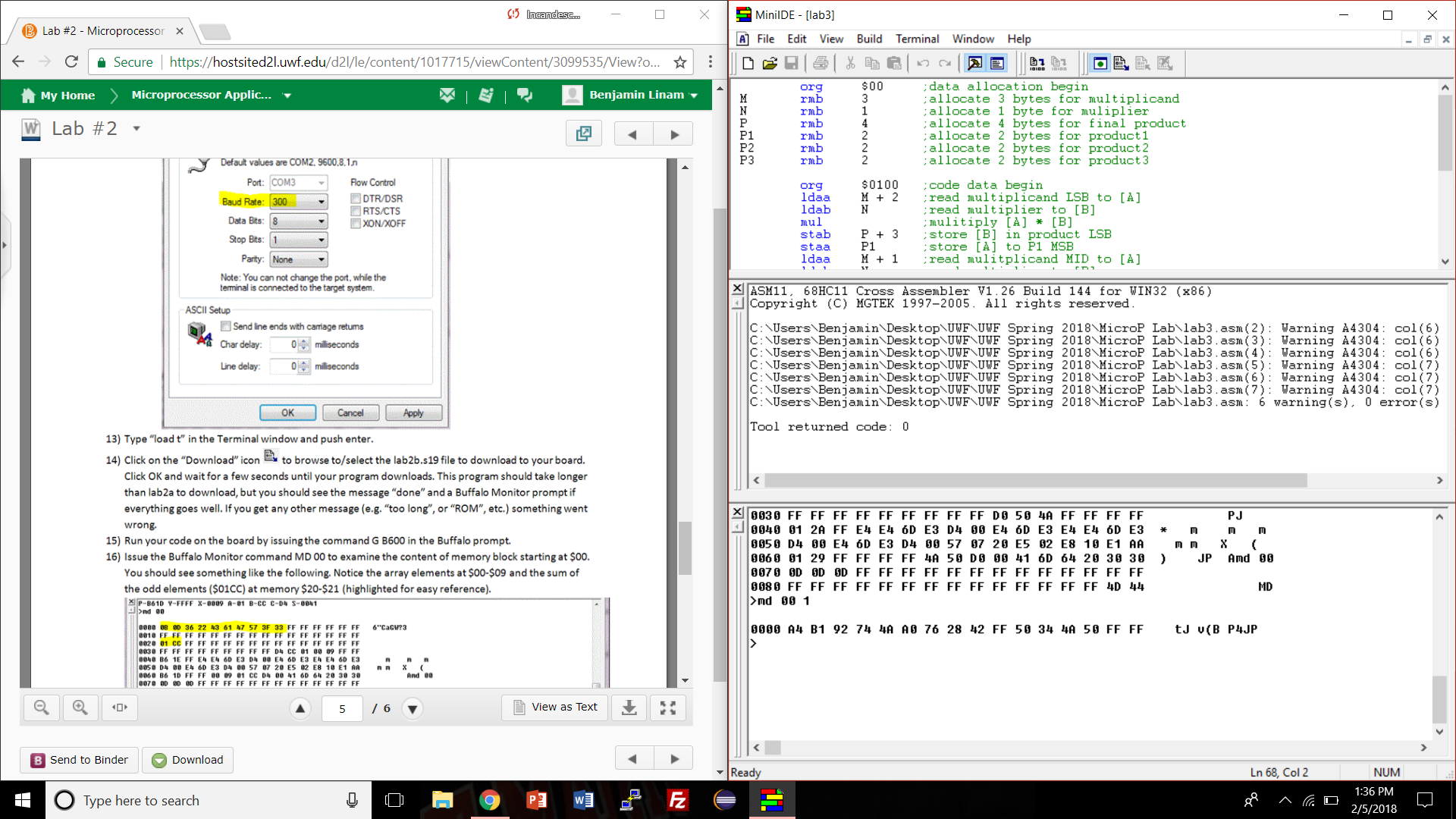
2. The program retrieves a 3-byte multiplicand from memory locations $00-$02 and a 1-byte multiplier from memory location $03. Each set of 2-byte terms shown in Figure 1 are then stored in accumulators A and B, multiplied by use of the MUL command, and finally saved at memory locations $04-$07 from most significant bit to least significant bit.

3. The following is a list of multiplicands and multipliers, of which the 4-bit product was to be determined:

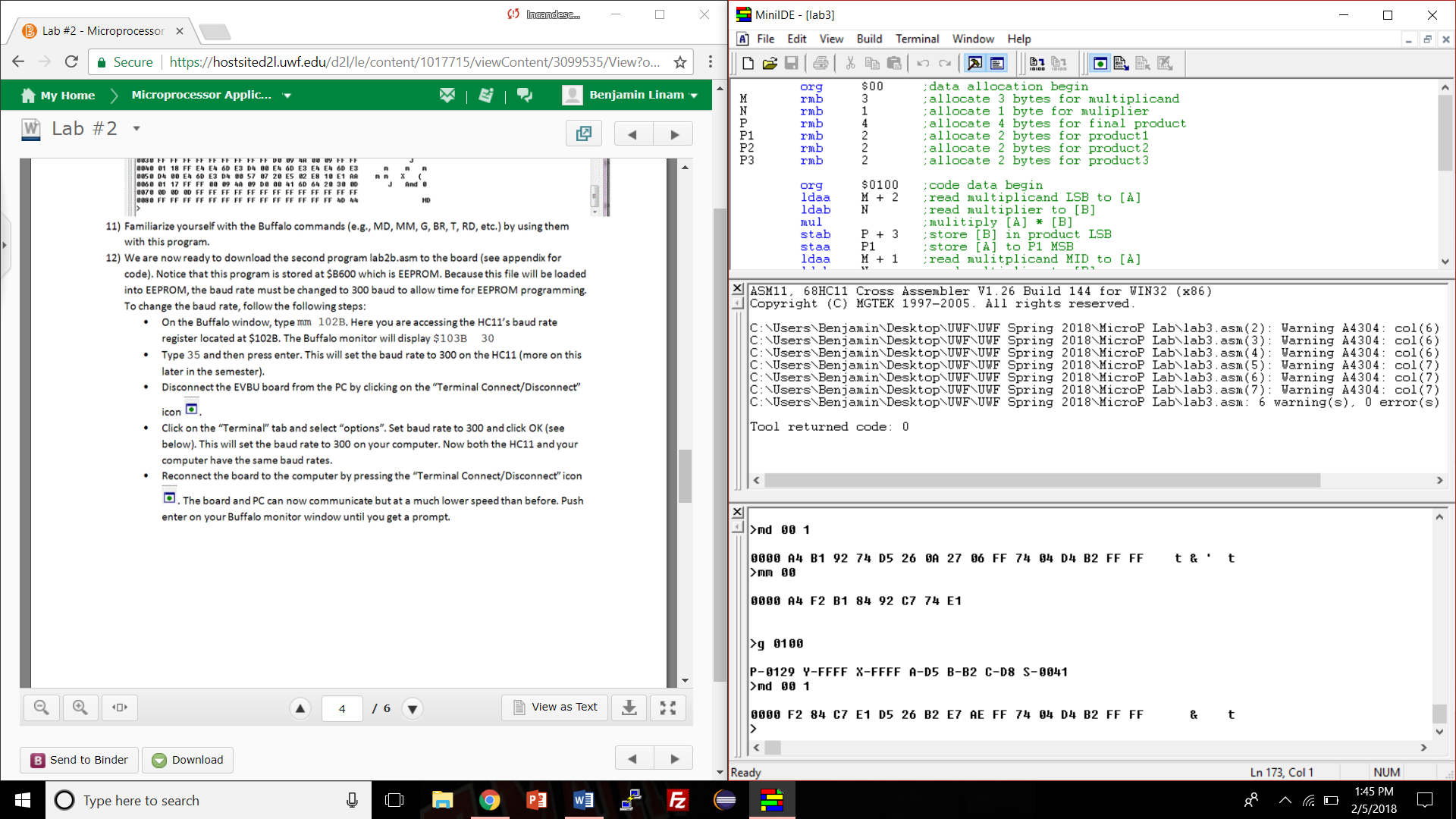
|  |  |  |
| --- | --- | --- |
| **Multiplicand** | **Multiplier** | **Product** |
| $A4 B1 92 | $74 | $4A A0 72 28 |
| $F2 84 C7 | $E1 | $D5 26 B2 E7 |
| $19 65 E9 | $F4 | $18 35 22 14 |
| $19 65 E9 | $00 | $00 00 00 00 |
| $19 65 E9 | $01 | $00 19 65 E9 |
| $19 65 E9 | $08 | $00 CB 2F 48 |
| $FF FF FF | $FF | $FE FF FF 01 |

**Table 1**: List of test multiplicands, multipliers, and their respective products.

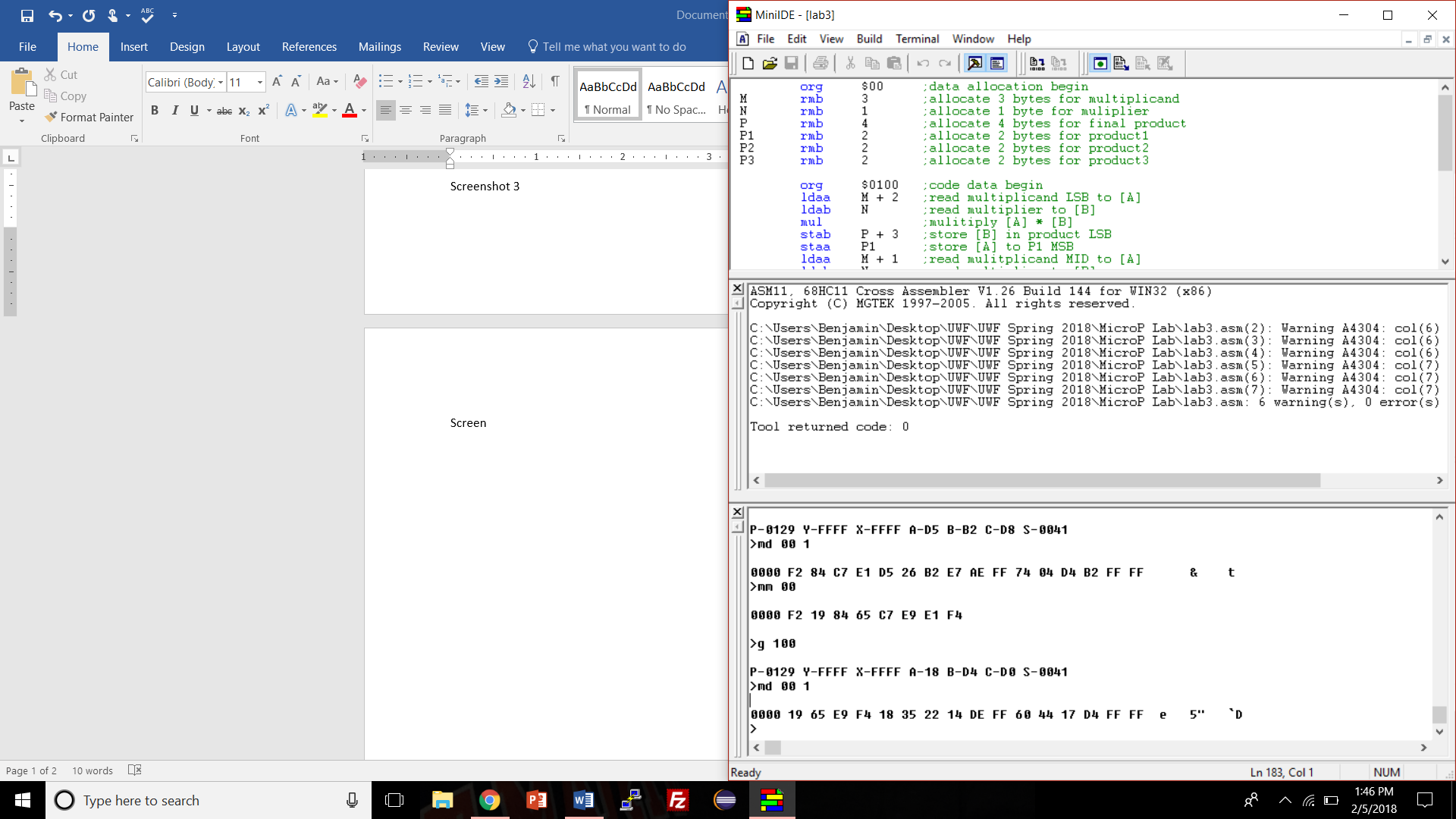
4. Once the product is calculated, that value can be displayed by using BUFFALO I/O routines to search for the specific memory location where the product was stored. The Buffalo commands used to enter the multiplicands and multipliers is “MM 00”, followed by the desired values represented in hexadecimal (0-F). That command will save the multiplicand to memory locations $00-02 and multiplier to memory location $03. After running the program using the command “G 0100” (0100 being the location of the starting byte of the program’s code), the product will be saved to memory locations $04-$07. Entering the command “MD 00 1” displays memory locations $00-$0F to enable all three values to be viewed at once.



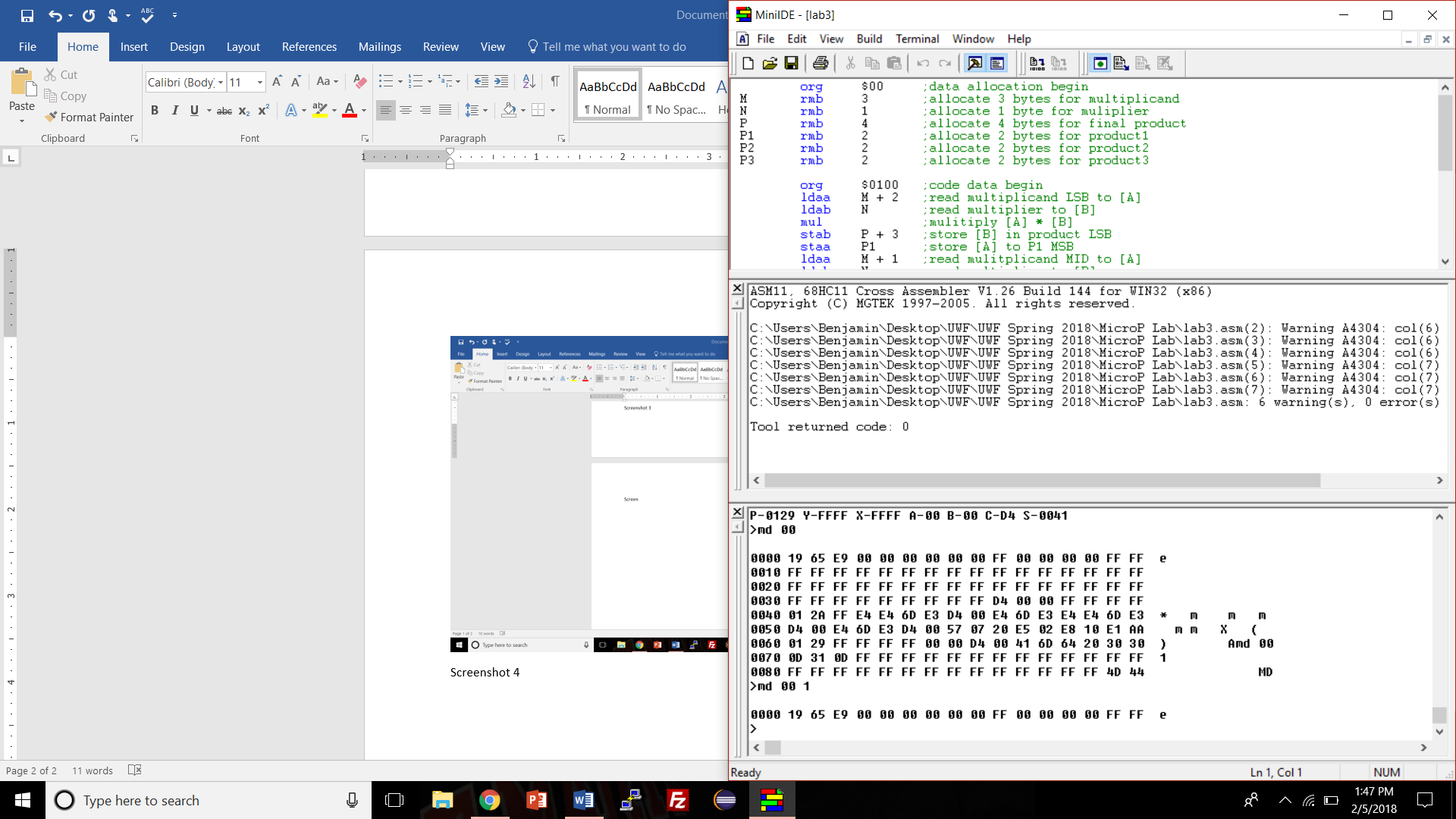
**Figure 3**: Memory locations $00-$0F after calculating first product.



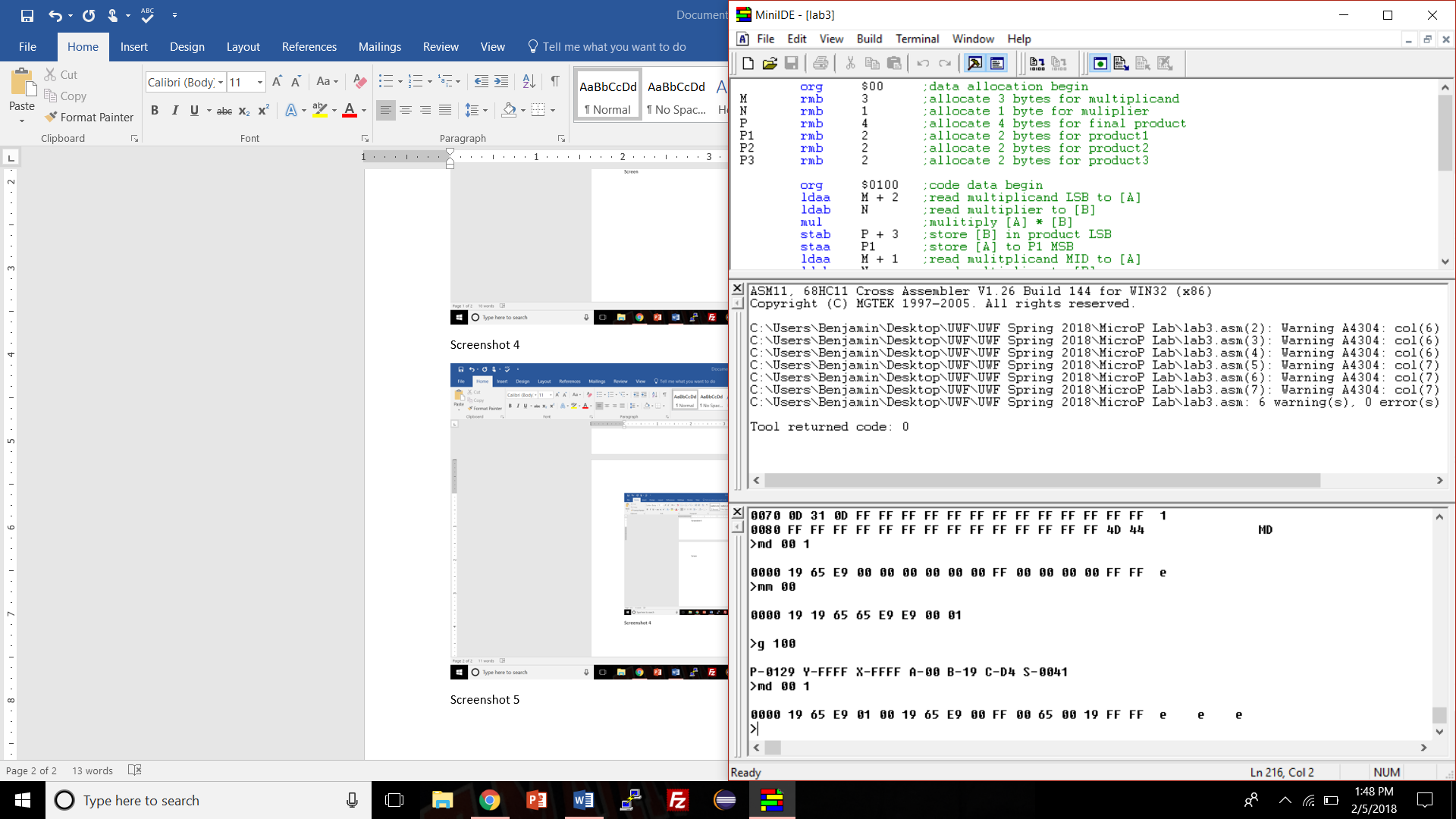
**Figure 4**: Memory locations $00-$0F after calculating second product.



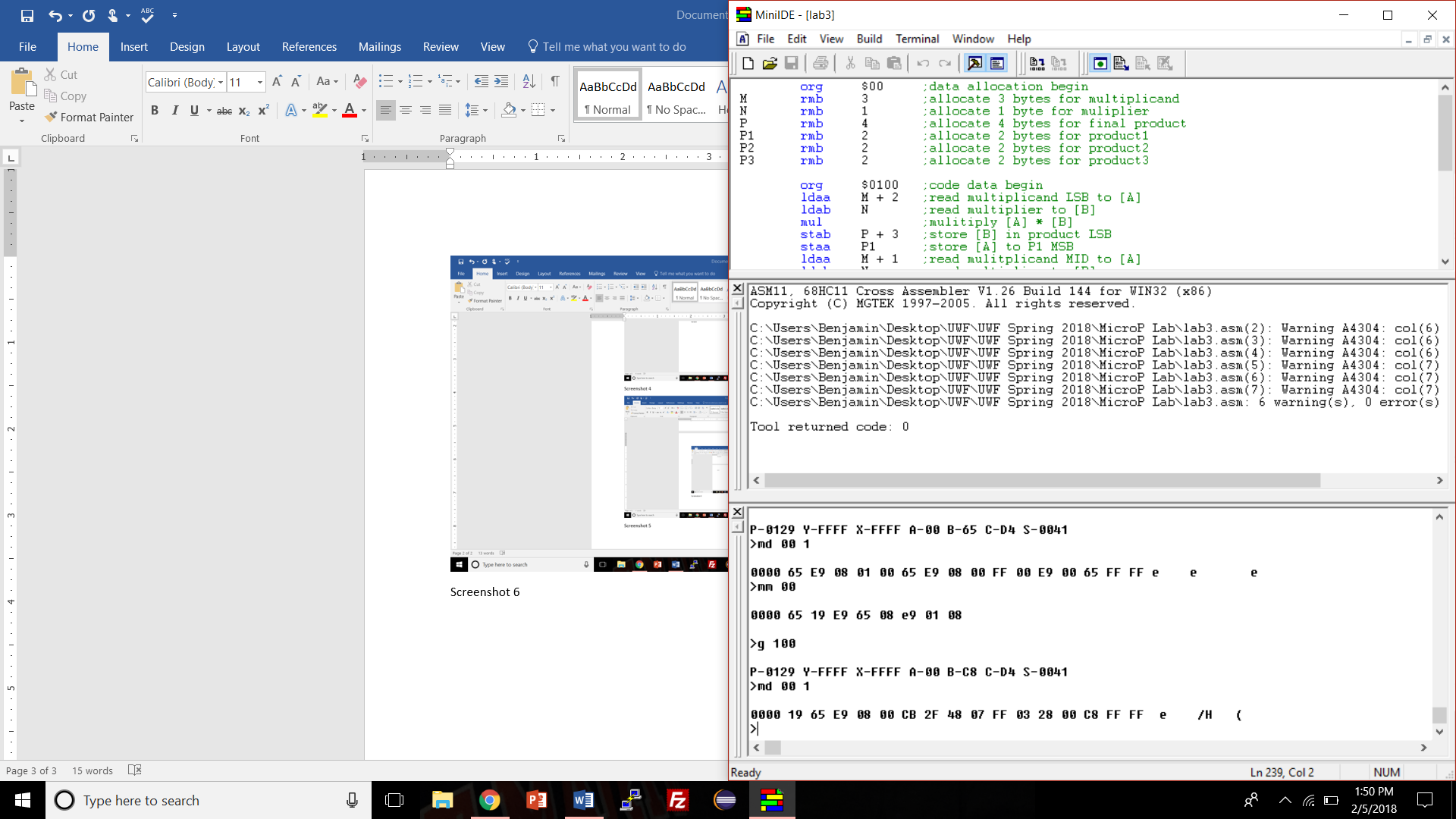
**Figure 5**: Memory locations $00-$0F after calculating third product.



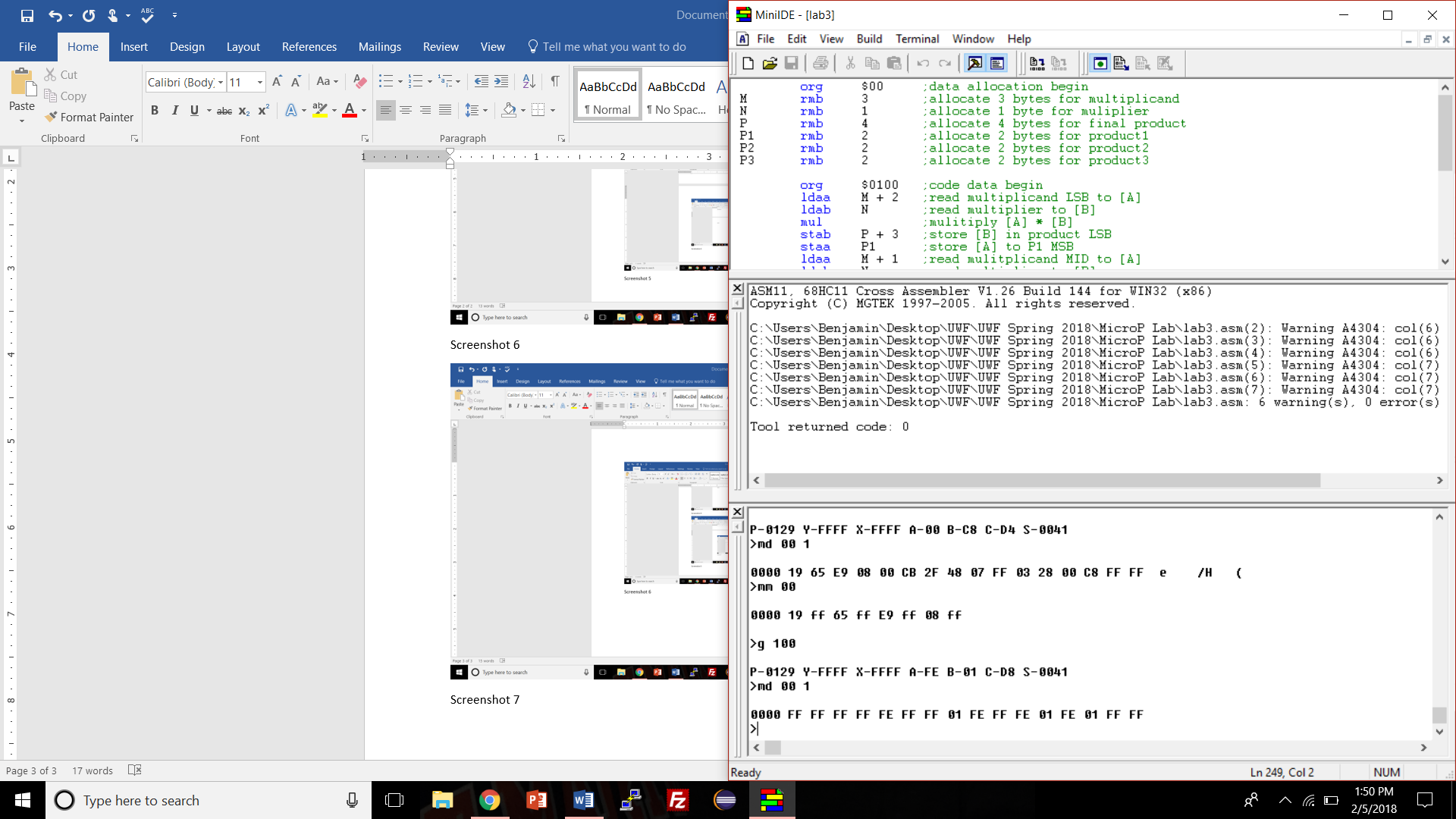
**Figure 6**: Memory locations $00-$0F after calculating fourth product.



**Figure 7**: Memory locations $00-$0F after calculating fifth product.



**Figure 8**: Memory locations $00-$0F after calculating sixth product.



**Figure 9**: Memory locations $00-$0F after calculating seventh product.

**Conclusions**

The program and lab were completed quickly and efficiently. Having the algorithm as well as the algorithm steps made it easy to decide which assembly commands were needed to complete the lab. The only problems encountered while completing the lab were that the mode jumpers were in the wrong location for testing the program and two screws had to be removed from the serial connector on the HC11.