278 Mini project 1

Floating-Point Half Precision, Mu-law and Fixed-Point adder and multiplier

Author – Deekshith Krishnegowda

ID - 012417080

San Jose State University

EE278

1. Table of Contents

[1. Floating point Half Precision-Adder/Multiplier 2](#_Toc525926517)

[1.1 Results 2](#_Toc525926518)

[1.2 Screenshots 2](#_Toc525926519)

[2. Mu-law -Adder/ Multiplier 3](#_Toc525926520)

[2.1 Results 3](#_Toc525926521)

[2.2 Screenshots 4](#_Toc525926522)

[3. Fixed point – Parameterizable Adder/Multiplier 4](#_Toc525926523)

[3.1 Results 4](#_Toc525926524)

[3.2 Screenshots 5](#_Toc525926525)

[4. Python code 6](#_Toc525926526)

[4.1 Floating point Half Precision 6](#_Toc525926527)

[4.2 Mu-law 8](#_Toc525926528)

[4.3 Fixed Point parameterized 12](#_Toc525926529)

# Floating point Half Precision-Adder/Multiplier

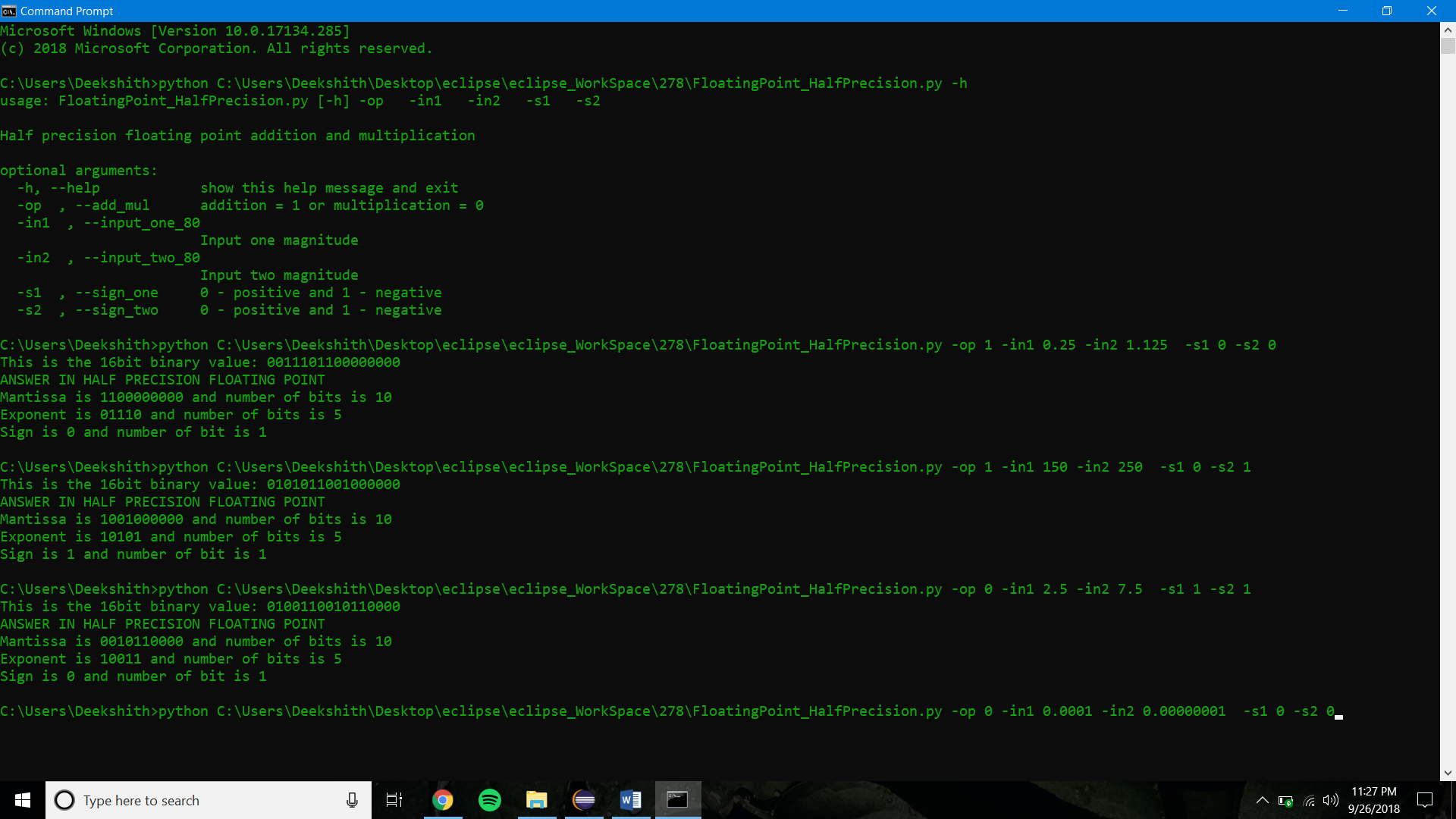
## Results

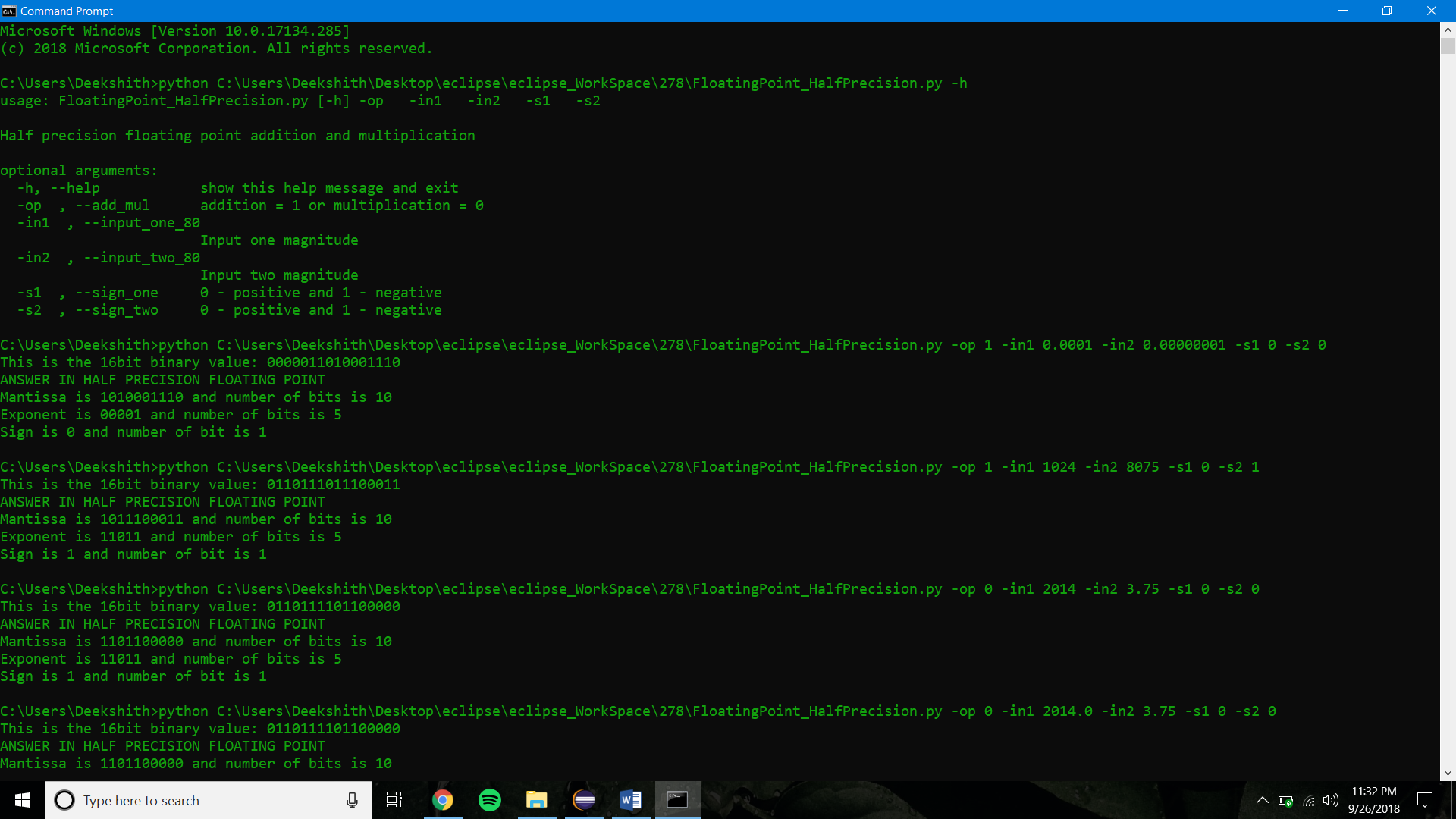
The Mantissa is 10 bits wide, exponent is 5 bits wide and the sign is 1 bit. The sign bit is 1 if the number is negative and the sign bit is 0 if the number is positive. The inputs to this module is need two numbers (only magnitude), the signs bits of two numbers. The module also requires opcode to be entered. This is 1 for addition and 0 for multiplication.

Sample commandline for Eg1: python file\_location/FloatingPoint\_HalfPrecision.py -op 1 -in1 0.25 -in2 1.125 -s0 0 -s1 0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Decimal | Sign (1) | Mantissa (10) | Exponent (5) |
| 1 | 0.25 + 1.125 | 0 | 1100000000 | 01110 |
| 2 | 150 -250 | 1 | 1001000000 | 10101 |
| 3 | -2.5 \* -7.5 | 0 | 0010110000 | 10011 |
| 4 | 0.0001 + 0.00000001 | 0 | 1010001110 | 00001 |
| 5 | 1024-8075 | 1 | 1011100011 | 11011 |
| 6 | 2014 \* 3.75 | 0 | 1101100000 | 11011 |

## Screenshots





# Mu-law -Adder/ Multiplier

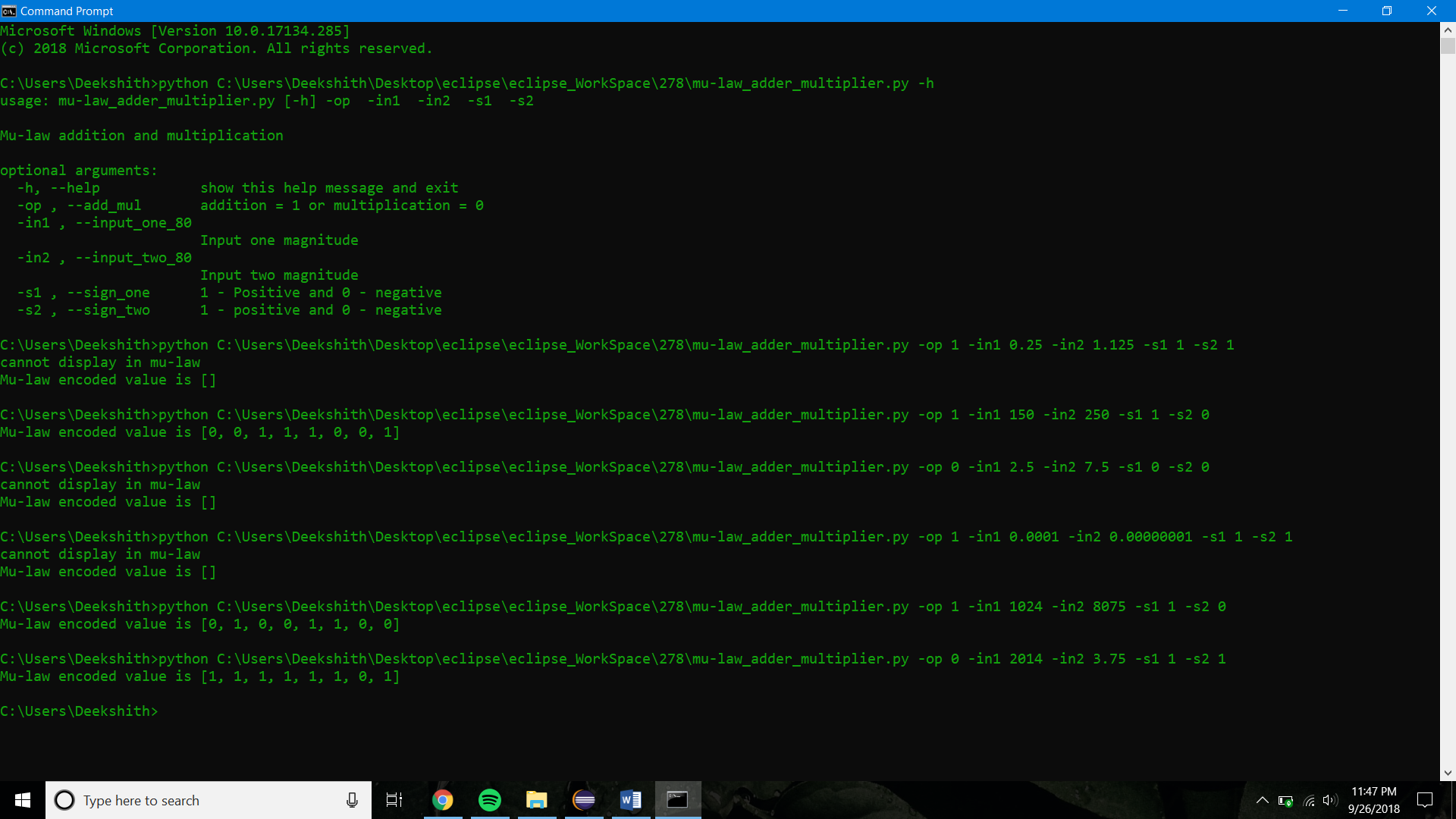
## Results

The following encoding table is used to calculate the mu-law encoded value. If the number is greater than 1 then it is divided by 8192. If the number is less than 1, then it not divided.

The number which are not is range of +1/8192 and -1/8192 cannot be displayed with this format.

|  |  |  |
| --- | --- | --- |
| No. | Decimal | Mu-law encoded |
| 1 | 0.25 + 1.125 | - |
| 2 | 150 -250 | 00111001 |
| 3 | -2.5 \* -7.5 | - |
| 4 | 0.0001 + 0.00000001 | - |
| 5 | 1024-8075 | 01001100 |
| 6 | 2014 \* 3.75 | 11111101 |

## Screenshots



# Fixed point – Parameterizable Adder/Multiplier

## Results

The floating-point format assumed in this model is 1-bit sign, 3 bit whole part and bit fraction part. Since the whole part is 3 bits wide, the maximum number which can be indicated is 2^3 which is 7. So, numbers which are not in that range will be not be displayed and an error messaged will be put out.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Decimal | Sign (1) | Whole part (3) | Fraction part (4) |
| 1 | 0.25 + 1.125 | 0 | 001 | 01100 |
| 2 | 150 -250 | - | - | - |
| 3 | -2.5 \* -7.5 | - | - | - |
| 4 | 0.0001 + 0.00000001 | 0 | 000 | 0000 |
| 5 | 1024-8075 | - | - | - |
| 6 | 2014 \* 3.75 | - | - | - |

The results within +7 and -8 will be displayed. Numbers greater than this range cannot be accommodated in this format.

## Screenshots

