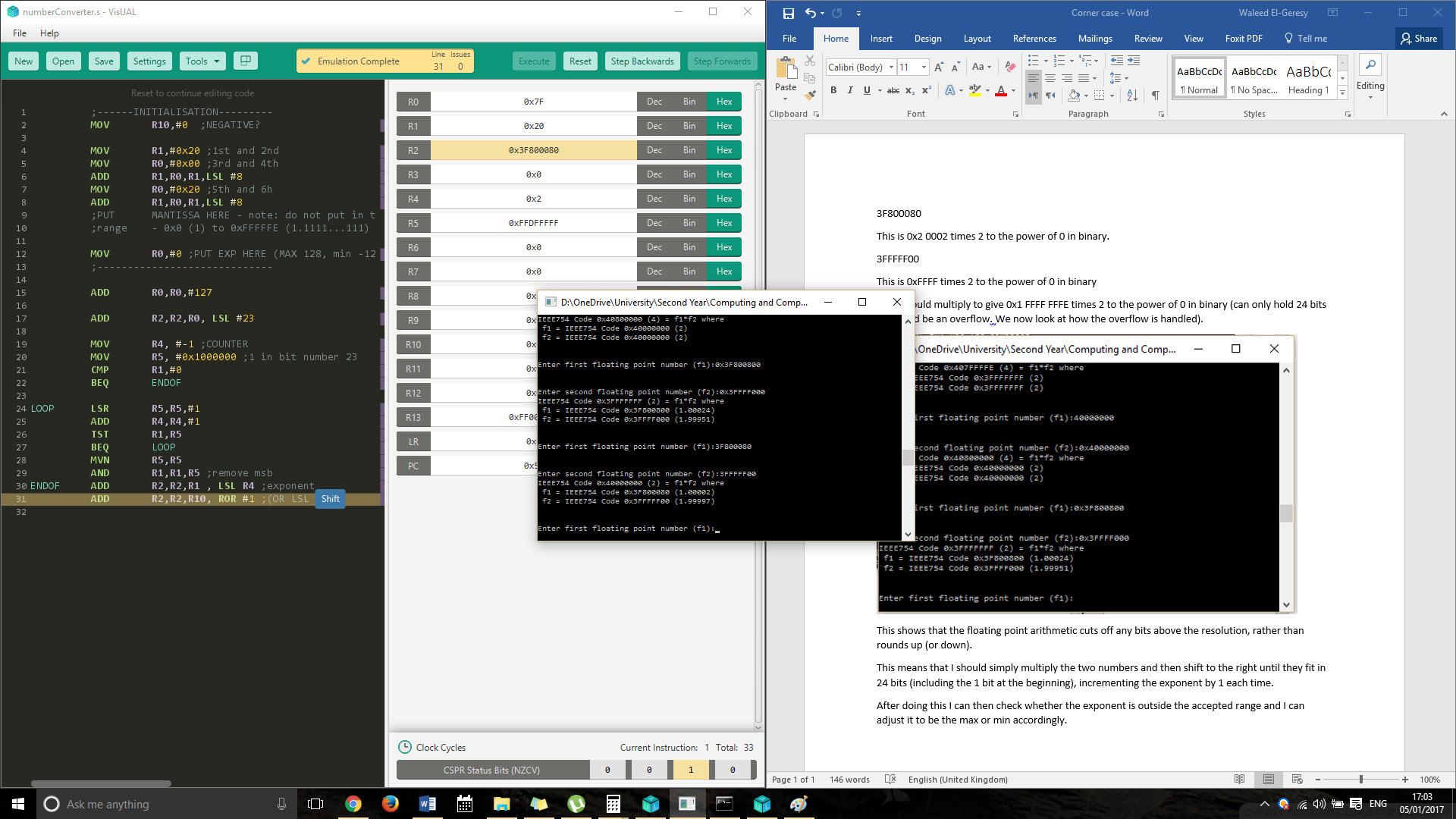
3F800080

This is 0x2 0002 times 2 to the power of 0 in binary.

3FFFFF00

This is 0xFFFF times 2 to the power of 0 in binary

They should multiply to give 0x1 FFFF FFFE times 2 to the power of 0 in binary (can only hold 24 bits so should be an overflow. We now look at how the overflow is handled).



This shows that the floating point arithmetic rounds up (or down) rather than cutting off any bits.

This means that I should simply multiply the two numbers and then shift to the right until they fit in 24 bits (including the 1 bit at the beginning), incrementing the exponent by 1 each time.

After doing this I need to also check the bit directly to the right of the new mantissa, and if it is 1, I need to add 1 to the mantissa. Then I need to check one more time and shift appropriately.

After doing this I can then check whether the exponent is outside the accepted range and I can adjust it to be the max or min accordingly.