Lab 06 - Ones and Twos Complement

In this lab, you’ve learned about how computers store and process numbers, specifically negative numbers. In addition, you have learned how to implement a full signed integer system with the Two's Complement number representation.

# Rubric

| **Item** | **Description** | **Value** |
| --- | --- | --- |
| Summary Answers | Your writings about what you learned in this lab. | 25% |
| Question 1 | Your answers to the question | 25% |
| Question 2 | Your answers to the question | 25% |
| Question 3 | Your answers to the question | 25% |

# Lab Summary

In this lab we learned how to use our previously written adder and full adder to implement a ones and twos complement adder/subtractor. We learned how to represent numbers as negatives using 1s and 2s complement but toggling bits and adding(for two's complement).

# Lab Questions

## 1 - Explain the differences between our Half Adder from last lab and the Half Subtractor from this lab.

The half adder carries the bit to the left, making it the new MSB, the subtractor brings the bit around to the right, and adds one in the LSB. Then the carry out from that addition, if another bit is carried out of the second addition, it is not added.

## 2 - What about the end around carry of One’s Complement makes it hard to use and implement?

This requires you to recalculate the bits after you carry around at the end of the first addition. This makes you have to use another set of adders to recalculate each bit. This requires you to do twice the amount of coding in order to feed the output of the carry to the LSB.

## 3 - What is the edge case and problem with Two’s Complement number representation?

The problem with the twos complement is when an overflow occurs when you exceed the upper or lower limit of your bit amounts possible values. WHen the magnitude of the sum of addition is greater than you can represent. This would cause your number to be represented correctly. For example when adding 10 + 12 represented by 4 bit unsigned binary numbers, there is an overflow because 22 can't be represented in 4 bits.

Code Submission

Upload a .zip of all your code or a public repository on GitHub.