*In your implementation of the ALU there is no need to detect or handle overflow. However, it's a good mental exercise to think about when/how an overflow might occur. When adding numbers with different signs, overflow cannot occur because the sum must be no larger than one of the numbers. What happens when the signs of the numbers are the same? What happens in the case of subtraction? How do we detect when an overflow does occur?*

Overflow can happen when two values added together have the same signs in such as positive-positive case or negative-negative case. For instance, when inputs and an output have 4 bits at maximum, and both inputs are 1111 in binary, the output has to be 11110, the most left side bit of which cannot be expressed with the output 4 bits. This is the how overflow happens. It is also the same case in negative numbers. If an addition of two negative numbers results in generating the larger digit numbers than the original inputs and the result exceeded the capacity of output bits, overflow can happen. To clarify the pattern of overflow, I quote the statements from the article. “Overflow occurs when: Two negative numbers are added and an answer comes positive or Two positive numbers are added and an answer comes as negative” (GeeksforGeeks, 2021).

In terms of ALU development, we need to detect the condition of overflow. In order to detect overflow, MSB, so called Most Significant Bit, and the result bit of those MSBs are important. If the MSBs are both positive and the result is negative bit 0, overflow happens. Also, when MSBs are both negative and the result is positive number, overflow occurs as well. This is how to detect the overflow in ALU.

References:

*Overflow in Arithmetic Addition in Binary Number System.* GeeksforGeeks. (2021, July 15). Retrieved July 7, 2022, from https://www.geeksforgeeks.org/overflow-in-arithmetic-addition-in-binary-number-system/#:~:text=Overflow%20Detection%20%E2%80%93&text=So%20overflow%20can%20be%20detected,Addition%20of%202's%20Complement%20number.