Bing

The overflow flag is a bit in the processor status register that indicates when an arithmetic operation has resulted in a signed overflow. A signed overflow occurs when the sign of the result does not match the sign of the operands, meaning that the result is too large or too small to fit in the destination. For example, if you add two positive numbers and get a negative result, or if you subtract two negative numbers and get a positive result, the overflow flag is set. [1] [2]

The overflow flag is useful for detecting errors in signed arithmetic, such as when adding or subtracting two's complement numbers. The overflow flag is not affected by unsigned arithmetic, such as when adding or subtracting binary numbers. For unsigned arithmetic, the carry flag is used to indicate when an overflow has occurred. [3] [4]

Here are some examples of when the overflow flag is set or cleared in assembly language, assuming a 4-bit processor:

- `ADD R0, R1` ; R0 = 0100, R1 = 0100, result = 1000, overflow flag is set
- `SUB R0, R1` ; R0 = 0000, R1 = 0001, result = 1111, overflow flag is set
- `ADD R0, R1` ; R0 = 0010, R1 = 0001, result = 0011, overflow flag is cleared
- `SUB R0, R1` ; R0 = 1000, R1 = 0001, result = 0111, overflow flag is cleared

I hope this helps you understand the overflow flag better. If you have any more questions, feel free to ask me. 😊