Arithmetic overflow

Arithmetic overflow occurs when one or more numbers in code take part in an arithmetic operation whose result is too large for the data type – thus the result is incorrect. For example, arithmetic overflow occurs when two positive integers add up and the result is negative (interpreted as a complement of two), or when two added up negative numbers give a positive result. In 4-bit arithmetic, when adding two numbers 1100 and 1000 (with two's complement) we have the result 1 0100, where the most significant bit is a carry bit. The sign of 0100 is equal to 0, which indicates a positive value. Since the numbers that formed were negative, arithmetic overflow occurred. When added to two addition, arithmetic overflow can be found "If the sign bits of the two source operands are the same and different from the sign bit of the operand of the result, an arithmetic overflow occurred."