Two's Complement

What Is Two's Complement?

Two's complement is the most common way to represent signed integers in binary, allowing for positive and negative numbers. In computer science, understanding two's complement is essential for working with binary arithmetic and data representation. It builds on the foundation of binary notation but modifies it to handle negative values effectively.

Why Use Two's Complement?

Two's complement simplifies the process of arithmetic operations like addition and subtraction. Without it, computers would need separate hardware to handle positive and negative values, complicating operations and design.

Two's complement solves the issue of "double zero" in other systems (like signed magnitude) by letting the highest bit represent a negative value. It also ensures that adding positive and negative numbers together results in correct outputs without special rules.

Counting in Two's Complement

Two's complement works by "flipping" the way we think about the most significant bit (MSB). In binary, the leftmost bit normally represents the highest power of 2. However, in two's complement, this bit becomes the sign bit:

- 0: The number is positive or zero.
- 1: The number is negative.

The key to two's complement is that negative numbers are represented by the binary pattern you'd get if you subtracted the absolute value from 2^n where n is the number of bits.

How to Convert Positive Numbers

Converting positive numbers to two's complement is straightforward:

1. Write the binary representation as usual.
2. If needed, pad with leading zeros to fit the required bit width (8 bits, 16 bits, etc.)

Example: Convert +13 to an 8-bit Two's Complement

1. Write 13 in binary: 1101.
2. Pad with zeros to fit 8 bits: 00001101.

How to Represent Negative Numbers

The process for converting a negative number involves two main steps:

1. Find the binary representation of its positive counterpart.
2. Invert all the bits and add 1.

Example: Convert -13 to an 8-bit Two's Complement

1. Start with the positive version of 13: 00001101.
2. Invert the bits: 11110010.
3. Add 1 to the inverted result:

```
11110010
+        1
------------
11110011
```

So, -13 in 8-bit two's complement is 11110011.

Converting Two's Complement Back to Decimal

To convert a two's complement number back to decimal, check the MSB (sign bit).

1. If it's 0, the number is positive, and you can read it as usual.
2. If it's 1, the number is negative:
    a. Invert all the bits and add 1.
    b. Convert the result to decimal and add a negative sign.

Example: Convert 11110011 to Decimal

1. The MSB is 1, so it's negative.
2. Invert the bits: 00001100.
3. Add 1:

    00001100

    \+ 1

    00001101

4. Convert 00001101 to decimal: 13.
5. Add the negative sign: -13.