

There is another way to subtract (in all bases) using complements.

First we'll explore the idea of complements in the decimal system before moving on the other systems.

10's complement -> subtract each digit from 9 then add 1 to the least aight e.g. 10's complement - 99 76 of 23:

9's complement it subtract each digit from 9

e.g. 9's complement of 23 is 99

-23

76

Complements can be used to pernform Subtraction

To subtract using 10's complement, you add the 10's complement (of the # you are subtracting) to the first number and drop the first carry.

To subtract using 9's complements, you add the 9's complement (of the # you are subtracting) to the first number and then add the final carry to the least digit.

EX: Subtract using both 10's and 9's complements.

Complements in any base:

We want to avoid having to write the code for subtraction in a programming language because it requires another electrical circuit and circuits are very expensive.

To save money, computers do addition only (they use complements to subtract and work with negative numbers).

In general, there are two types of complements in every number system:

Detrue complement - formed by subtracting cach digit of the number from "the radix minus one" and then adding one to the least significant digit of the number formed

In the decimal system, the 10's complement is the true complement,

by subtracting each digit of the number from "the radix minus one"

In the decimal system, the 9's complement is the radio minus one complement.

In the binary system we have:

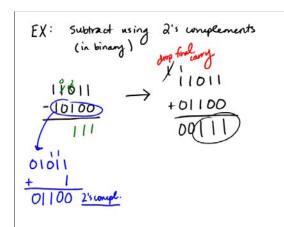
1. 2's complement (true compl.)

2. 1's complement (radix minus one longle)

Binary Complements (much easier to find)

of's complement is formed by changing every 1 to a 0, every 0 to a 1, and adding 1 to the least bit

"bit" = binary diget



EX: Subtract in binary using 2's complements.

11011

-1100

+10100

01111

10100

changing each 0 to a 1

and each 1 to a 0

EX: 18001

-10110

00010

The Octal and Hexadecimal number (base 8) (base 16)

Syptems help us condense binary numbers into numbers that are more manageable.

Decimal system isn't so helpful (it makes it hard to "see" the suitches)

Octal System (base 8)

We want to be able to convert quickly binary -> octal and octal -> binary without having to go through decimal.

| binary -> octal octal of 3 (starting at radio point and working orderards)

read each group of 3 as an octal digit

 $\frac{1}{2^2}\frac{1}{2^4}\frac{1}{2^4}$. $\frac{1}{2^4}\frac{1}{2^4}\frac{1}{2^4}$.

EX: 111110111.2 -> base 8

EX: convert 111012 to octal.