

Monday, May 6, 2024 4:11 PM

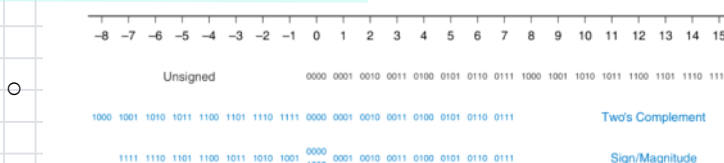
- **Bits** - the smallest unit of data in computing
  - **Eg** - a USB 3.0 can send data in 5 Gbytes/sec
- **Byte** - a group of 8 bits
- **Nibble** - a group of 4 bits or half a byte
- **Least and most significant bits and bytes**



- **Binary Numbers**
  - Decimal → Binary
    - divide by 2
    - put the remainder to the left of the binary representation
  - $2^{**10} = 1024$ ,  $2^{**20} = 10^{**6}$ ,  $2^{**30} = 10^{**9}$
  - **ESTIMATING POWERS OF TWO**
    - Split the exponent into a multiple of ten *(which can easily be solved)* and the remainder then multiply them.
  - **Adding binary numbers**
    - As in decimal addition, if the sum of two numbers is greater than what fits in a single digit, we carry a 1 into the next column.
  - Binary → Decimal

$$A = \sum_{i=0}^{n-1} a_i 2^i$$

- **Sign/Magnitude Numbers** - uses the most significant bit as the sign and the remaining  $N-1$  bits as the magnitude
  - **0** - positive
  - **1** - negative
  - $N$  bits can represent numbers ranging b/n  $-(2^{**n-1} - 1)$  up to  $2^{**n-1} - 1$
  - **shortcoming** - we cannot perform a simple addition and Num 0 has two representation +0 and -0
- **Two's Complement Numbers** - is similar to unsigned binary numbers except that the most significant bit position has a weight of  $-2^{N-1}$  instead of  $2^{N-1}$ 
  - Overcomes the shortcomings of sign/magnitude numbers
  - **Positive → negative** - invert all of the bits in the number, then add 1 to the least significant bit position.
  - $N$  bits can represent numbers ranging b/n  $-2^{**n-1}$  up to  $2^{**n-1} - 1$
  - **Overflow** - when result is too big to fit in the available number of bits, happens if the result is greater than  $2^{N-1} - 1$  or less than  $-2^{N-1}$  when adding two  $N$ -bit positive numbers or negative numbers
    - never happens when adding negative and positive
  - **Sign Extension** - Adding bits to the left to avoid overflow
    - 4-bit representation of 3 = 0011 and 8-bit sign-extended value: 0000 0011
    - 4-bit representation of -5 = 1011 8-bit sign-extended value: 1111 1011
- **Comparison b/n the three types**



★ Unless stated otherwise, assume that all signed binary numbers use two's complement representation.