Base 2

Addition and Subtraction

Assembly language programming
By xorpd

xorpd.net

Objectives

- We discuss the decimal representation system and some of its features.
- We study how to represent numbers using only two symbols: 0 and 1.
  - This is called **binary** representation.
- We study how to add and subtract numbers using the binary representation.

Why is this important?

- Computer parts are engineered to understand only two symbols: 0 and 1.
    - 0 means low voltage, and 1 means high voltage.

- Everything in the computer's software is eventually constructed from 0 and 1 symbols. That is – **binary** numbers.

- In order to instruct the processor to do the things you want, you must be able to speak his language.

- Understanding binary numbers is **crucial** for your understanding of anything low level in the computer, especially assembly programming

Decimal representation

- We make a daily use of the decimal representation, also known as “base 10”.
    - It is the most common numbers representation among human beings.
    - If you survived so far in our culture, you probably know how to use it.
- Decimal means 10.
- The decimal representation uses 10 different symbols (0,1,2,...,9) to represent numbers.

Decimal representation (Cont.)

- The significance of every digit is related to its location in the number - Also called **positional notation**.

- $$\begin{aligned} 12513 &= 3 \cdot 10^0 + 1 \cdot 10^1 + 5 \cdot 10^2 + 2 \cdot 10^3 + 1 \cdot 10^4 \\ &= 3 + 1 \cdot 10 + 5 \cdot 100 + 2 \cdot 1000 + 1 \cdot 10000 \end{aligned}$$

- Leading zeroes don’t change the value of the number.

Why you should care

- It is easy to underestimate the wisdom in this numeric system.
  - There were times and places in the history of human beings where people were struggling with much stranger numeric systems.

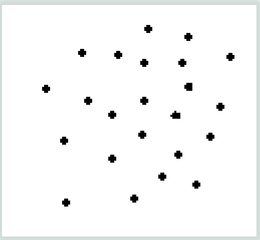
- Numeric representations are ways to keep in memory or communicate the quantity of something.

- The quantity and the representation are not the same thing.

- If you don’t know about any smart numeric system, in order to communicate a quantity to someone else you actually have to show him the quantity.
  - That is very wasteful.

Why you should care (Cont.)

- This is how 23 looks like if you don't know about any numeric system:



- What if you wanted to represent 1000000 in the same method?
- Different numeric systems invoke different thoughts and ideas about the nature of numbers.

Features of the decimal system

- The decimal system allows to represent very large numbers with not so many digits.
  - k digits are enough to generate a number as big as 10^k.

- Arithmetic is pretty easy:
  - Easy to add and subtract numbers when they are in the decimal representation.
  - Relatively easy to multiply and divide numbers in the decimal representation. (Though not as easy as adding or subtracting).

- It is easy to check divisibility of a decimally represented number by 2,4,5,10. It is also not very hard to check divisibility by 3,9 and 11.

Generalizing positional notation

- Choosing 10 as a base number is pretty arbitrary.
  - We could choose any other base number to create a new number representation system.

- For example, in base 5:
  - we will have exactly five symbols: 0,1,2,3,4.
  - $12403_5 = 3 \cdot 5^0 + 0 \cdot 5^1 + 4 \cdot 5^2 + 2 \cdot 5^3 + 1 \cdot 5^4 = 978_{10}$.

- Base 2
  - About the lowest we could get, while having a useful experience.
  - Only two symbols: 0 and 1. Every **bi**nary digi**t** is called **bit**.
  - Example: $1101_2 = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 = 13_{10}$

- Life in base 2 is mostly like in base 10.
  - maybe even better.

Addition in base 2

- Addition is done just like in the decimal case.
  - Align the two summands to the right, and add digit by digit.
  - If the sum is larger than the base, move the carry to the next position.

- Basic examples:
  - $0_2 + 1_2 = 1_2$
  - $1_2 + 1_2 = 10_2$
  - $11_2 + 1_2 = 100_2$
  - $1111_2 + 1_2 = 10000_2$

Addition in base 2 (Cont.)

- Larger example: $10110_2 + 1110_2 = ?$

$$\begin{array}{r} 10110 \\ +\ \ 1110 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ 10110 \\ +\ \ 1110 \\ \hline 00 \end{array} \qquad \begin{array}{r} 11 \\ 10110 \\ +\ \ 1110 \\ \hline 100 \end{array}$$

$$\begin{array}{r} 111 \\ 10110 \\ +\ \ 1110 \\ \hline 0100 \end{array} \qquad \begin{array}{r} 1111 \\ 10110 \\ +\ \ 1110 \\ \hline 00100 \end{array} \qquad \begin{array}{r} 1111 \\ 10110 \\ +\ \ 1110 \\ \hline 100100 \end{array}$$

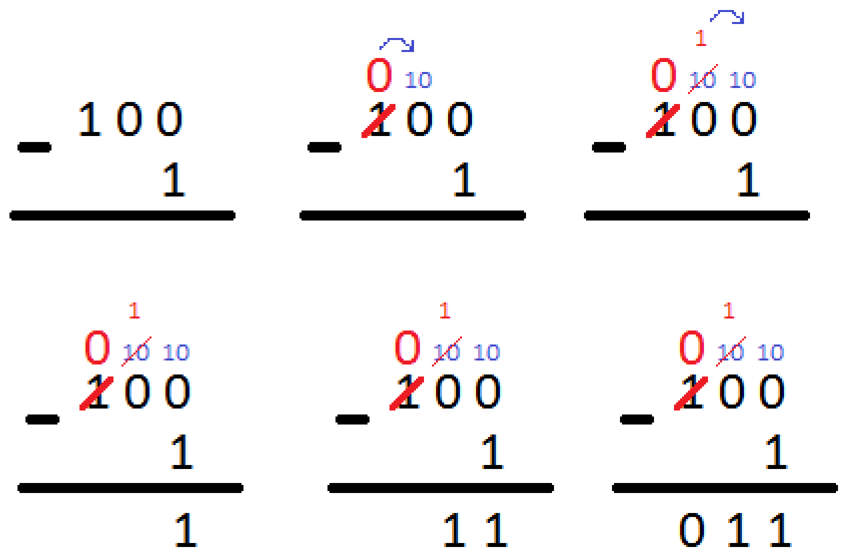
Subtraction in base 2

- Example: $1101_2 - 100_2 = ?$

$$\begin{array}{r} 1101 \\ -\ 100 \\ \hline 1 \end{array} \quad \begin{array}{r} 1101 \\ -\ 100 \\ \hline 01 \end{array} \quad \begin{array}{r} 1101 \\ -\ 100 \\ \hline 001 \end{array} \quad \begin{array}{r} 1101 \\ -\ 100 \\ \hline 1001 \end{array}$$

Subtraction in base 2 (Cont.)

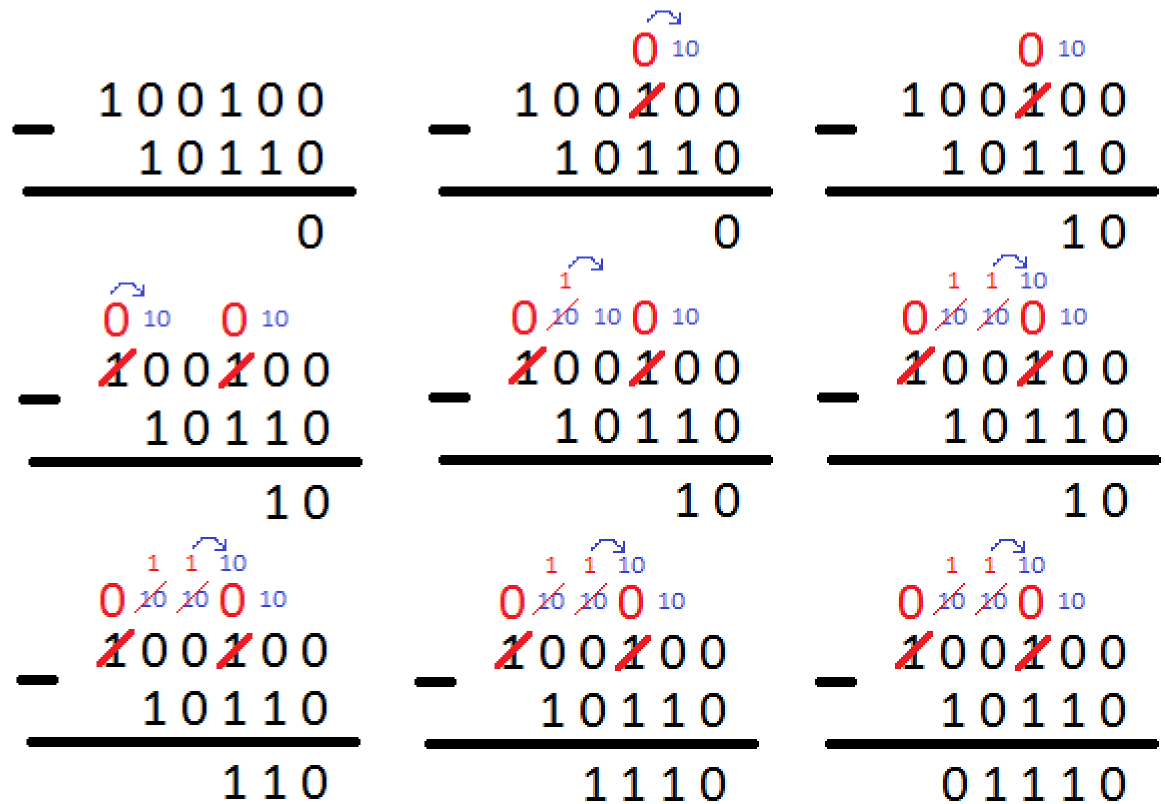
- Example: $100_2 - 1_2 = ?$



- In some cases subtraction could be less fun than addition.

Subtraction in base 2 (Cont.)

- Example: $100100_2 - 10110_2 = ?$



Exercises

- Adding numbers.
- Subtracting numbers.
- Enjoy :)