Lecture 1 - Computer Data and Numbering Systems

Computer Programming

Samia Hilal

Marianopolis College

January 30, 2022

Data in a computer

- A computer represents everything as a sequence of binary digits or *bits* (False = 0, True = 1).
- A *byte* is a group of eight bits.
- The terms kilobyte, megabyte, and gigabyte refer to collections of bytes, generally in thousands, millions, or billions.
- Actually, this is not exactly True. Data storage in the computer is measured in powers of 2.
- Everything in the computer – text, music, video, and programs – is made up of bits.
- Most computers process data in specific widths, for example 8, 16, 32, or 64 bits.

Data Storage in a computer

- We have all heard the terms: kilobyte, megabyte, and gigabyte used to refer to computer data storage.
- These terms are used *roughly* to indicate thousands, millions, or billions.
- We say *roughly* because the exact data storage values represented by these terms are slightly different:
  - 1KB $= 2^{10} =$ 1024 Bytes
  - 1MB $= 2^{20} =$ 1,048,576 Bytes
  - 1GB $= 2^{30} =$ 1,073,741,824 Bytes
  - 1TB $= 2^{40} =$ 1,099,511,627,776 Bytes

Binary

- Integers are usually represented in "base 10" or decimal.
- Each digit represents 0-9 times 10^i, where i is the position of the digit counting from the right.

$$102 = 1 \cdot 10^2 + 0 \cdot 10^1 + 2 \cdot 10^0$$

- In digital electronics, we represent numbers using only two states: *on* or *off*, or 1 and 0.
- This is called "base 2" or *binary*.
- $102_{10} = 1100110_2$

Binary, continued

- Why is $102_{10} = 1100110_2$?
- $= 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$
- $= 1 \cdot 64 + 1 \cdot 32 + 1 \cdot 4 + 1 \cdot 2$
- Computer scientists and programmers also make use of *octal* (base 8) and *hexadecimal* (base 16).
- Octal uses digits 0-7 times powers of eight.
- Hexadecimal uses digits 0-9 and A-F times powers of sixteen.
- Why is $102_{10} = 66_{16}$? $6 \cdot 16^1 + 6 \cdot 16^0$
- $102_{10} = 146_8 = 66_{16}$

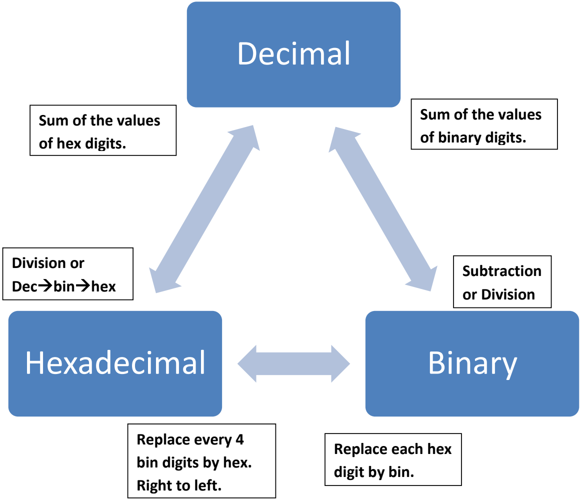
Positional Numeral Systems

- Binary, Octal, Decimal, and Hexadecimal are examples of positional numeral systems.
- They provide a representation of numbers by an ordered set of digits. The value of each digit depends on its position.
- In a given numeral system, the base is the number of unique digits, including zero, used to represent numbers.
- Our decimal system is the base-10 system. It uses 10 digits and powers of 10 for each position in a number.
- An integer can be represented exactly using any base.

Positional Numeral Systems

- We will be looking at base 2 (binary) and base 16 (hexadecimal) in addition to base 10.
- When the base of a number is something other than 10, the base is denoted by a subscript.
- Sometimes, the subscript 10 is added for emphasis but it is not required:
- $11001_2 = 25_{10} = 19_{16}$
- We will learn to convert numbers between the 3 numeral systems.
- The document: DataRepresentationExamples has several detailed examples.

Positional Numeral Systems



From Decimal to Binary

- There are 2 conversion methods: *Subtraction Remainder* and *Division Remainder*.
- The Subtraction Remainder method:
  - Intuitive and reinforces the idea behind base mathematics. It is strongly encouraged.
  - An integer is represented as the sum of powers of two starting from the largest possible power of 2 and going down.
- The Division method:
  - Mechanical and easy.
  - It uses the idea that successive division by a base is equivalent to successive subtraction by powers of the base.
- Refer to examples document provided.

From Binary To Hexadecimal

- It is difficult to read long strings of binary numbers – and even a modestly-sized decimal number becomes very long in binary.
- For example: $11010100011011_2 = 13595_{10}$
- For compactness and ease of reading, binary values are usually expressed using the hexadecimal (base-16) numbering system.
- The hexadecimal numbering system uses the digits 0 through 9 and the letters A through F.
- Example: decimal $12 = C_{16}$ and $14 = E_{16}$.
- To convert from binary to hexadecimal, group binary digits from right to left into groups of four and replace each group by its hex value.

From Binary To Hexadecimal

Decimal	4-Bit Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

From Decimal To Hexadecimal

- To convert from decimal to hex, we can convert from decimal to binary and then convert from binary to hex.
- Decimal –> Binary –> Hexadecimal
- This method is simple because we can reuse the same steps used earlier.
- Another option is the division remainder method which can be used to convert integers directly from decimal to hexadecimal.
- Refer to examples document.