THE HIDDEN LANGUAGE OF COMPUTERS

Exploring Data Representation

Prabhas Reddy Muppa

08-12-2022

IIITDM Kancheepuram

The Hidden Language of Computers

Exploring Data Representation

A Research Work

Submitted by

Muppa Prabhas Reddy

CS21B1039

Department of Computer Science and Engineering

Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram

Chennai- 600127

December 2022

TABLE OF CONTENTS

1. HISTORY

One of the earliest digital systems was the dial telephone system. This system works on the principle of pulse dialing. As the dial rotates, it interrupts the circuit between the phone and the exchange creating dialing pulses. The number of pulses generated depends on the number dialed. After all the numbers had been dialed and recorded, switches were set to connect the user to the desired phone number.

The invention of the transistor in 1948 at the Bell Telephone Laboratories revolutionized the way that computers were built. Transistors are used as electrical switches that can be in the “on” or “off” state and so can be used to build digital circuits and systems. Transistors were used initially as discrete components, but with the invention of Integrated Circuit (IC) technology, their utility increased exponentially. IC technology makes it possible to build entire digital building blocks into a single chip. These chips generally have a few hundred transistors. The size of the chips has been reducing ever since their birth and even large systems are being integrated into a single chip (system on a chip).

From the 1950s on, digitization has changed the way we work, shop, bank, travel, educate, govern, manage our health, and enjoy life. The technologies of digitization enable the conversion of traditional forms of information storage such as paper and photographs into the binary code (ones and zeros) of computer storage.

2. MILESTONES

Recognizing the achievements that have led to today’s digital environment, let’s discuss few milestones in the history of digital systems.

1679	Gottfried Wilhelm Leibniz developed the modern binary system and published- Explanation of Binary Arithmetic.
1847	George Boole introduced Boolean algebra in The Mathematical Analysis of Logic, creating the field of mathematical logic, leading eventually to universal computation.
1939	Harvard University built the Harvard Mark 1, which was used to compute ballistic tables for the U.S. Navy.
1956	IBM announced RAMAC (Random Access Memory Accounting), the first computer storage system to use a random access disk drive- the 350 Disk Storage Unit.
1959	Mohamed Atalla and Dawon Kahng invented the MOSFET (Metal- Oxide Semiconductor Field Effect Transistor) at Bell Labs, which is the most common transistor in digital circuits.
1965	Maurice Wikes develops the idea of cache computer memory. Cache is the temporary memory officially termed “CPU cache memory”. This chip-based feature of the computer lets us access some information more quickly.
1985	IEEE 754 (Institute of Electrical and Electronics Engineers) standard for representing floating- point numbers in computers was officially adopted.

3. DATA REPRESENTATION IN COMPUTERS

3.1. Interpreting Binary Numbers

The term ‘binary’ means something that has only two posibble states. The two states can be viewed as ‘0’ and ‘1’ in the binary number system, ‘true’ and ‘false’ in the Boolean logic and ‘on’ and ‘off’ in the electrical circuits. Computers don’t understand words or numbers the way humans do. Every data in the computer is an electrical signal and electrical signals are much harder to measure and control precisely. It makes more sense to only distinguish between an “on” state and “off” state. Therefore, the binary system became the primary language of computing systems.

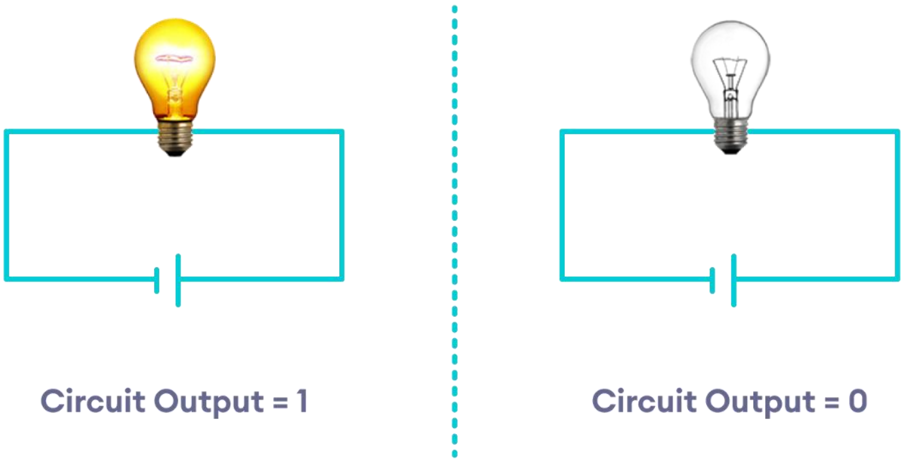


Image credit : Programiz

Computers operate on the electrical signals generated by the circuits. In order to design an efficient computer, it is better to interpret electrical signals as binary values. This broad generalization reduces the range of interpretation of each electrical signal into two distinct values, instead of an infinite range of continuous voltage values. Essentially, our computers use a series of high-voltage and low-voltage electrical signals (binary values) to represent everything from texts and numbers to images and sounds.

3.2. Understanding Boolean Logic

In the mid 1800's, an English mathematician George Boole formulated the Boolean logic. Boolean logic deals with only two possible values: **true** and **false**. True is represented by 1 and false is represented by 0. Computers use this logic to implement various instructions given by the user. Many instructions involve decision making. The computer interprets the condition and implements the corresponding instruction depending upon the truth value of the condition. This system of decision making is crucial for computers to execute instructions.

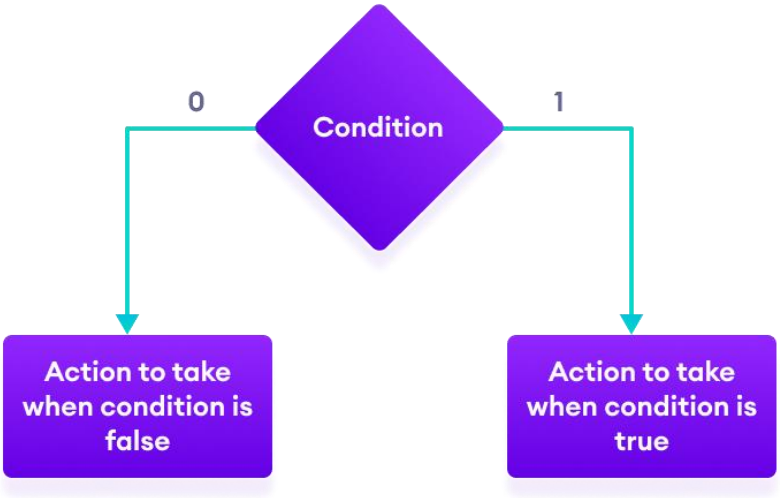


Image credit: Programiz

Logical operators such as AND, OR and NOT operators are directly taken from the AND, OR, and NOT operations from the Boolean logic. Many logical circuits can be realized from the Boolean algebra. Also, these boolean variables and logical operators are fundamental components used in implementing conditional statements and control statements in programming languages.

3.3. Number Representation

The numbers we use in everydaylife are decimal numbers. The base of the decimal number system is 10, as there are 10 different digits (from 0 to 9) called decimal digits, possible in the system. Since the computer can accept only binary values, we must represent the decimal digits by means of a code that contains 1's and 0's. Therefore, the need for a base-2 number system occurred. The base-2 number system contains only two digits: **0** and **1**. The numerical value is obtained from a binary number in the same way as that of decimal number, except that we multiply each bit with successive powers of 2. Every decimal number is accommodated a specific amount of memory (usually 4 Bytes) depending upon the configuration of the system.

We have the IEEE 754 standard for representing floating point numbers (numbers with fractions and exponents). The standard addressed many problems found in the diverse floating point implementations that made them difficult to use reliably. There are several ways to represent floating point number but IEEE 754 is the most efficient. IEEE 754 has 3 basic components:

- **Sign**
- **Exponent**
- **Mantissa**

IEEE 754 numbers are divided into two based on the above three components: single precision and double precision.

	Sign (s)	**Exponent (e)**	**Mantissa (m)**
32-bit	1-bit	8-bit	23-bit
64-bit	1-bit	11-bit	52-bit

3.4. Character Representation

Many applications of digital computers require the handling not only of numbers, but also of other characters or symbols, such as the letters of the alphabet. Therefore, it is necessary to formulate a binary code to represent alphabets, special characters and numerals at the same time. The most common systems for representing characters are **ASCII** (American Standard Code for Information Interchange) and **Unicode** (an extension of ASCII). These systems assign unique numeric values to characters and store them in binary format. The alphanumeric set contains 10 decimal digits (0- 9), 26 alphabets (A- Z) and other special characters such as *, $, & etc,. Hence, the ASCII system used 7 bits to represent a character (which was extended to 8 bits). For example, the ASCII code for character **a** is **97**, represented by **01100001.** The following hyperlink provides detailed information regarding the ASCII codes of different characters.

!"#$%&'()*+,./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRST
UVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~

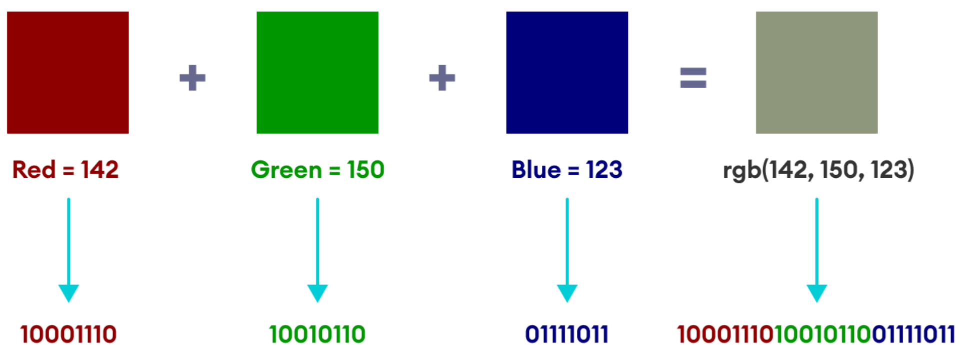
hyperlink credit: ascii-code.com

This system works well for representing English characters and symbols, but it is inadequate for representing letters and symbols from across the world. This limitation of ASCII code was fulfilled by the Unicode. Unicode is a universal character encoding standard. Unicode uses 21 bits for representing a symbol in contrast to ASCII code, which uses only 8 bits. Accommodating 21 bits greatly increases the range of values. To conclude, characters and text are assigned unique numerical values, which then converted to base- 2 format in groups of either 7 or 8 bits (ASCII) or 32 bits (UTF- 8).

3.5. Pixel and Image Representation

Even images are represented by numbers in the computer. In computer, an image is made of many, small colored squares called pixels. These pixels can be viewed as building blocks of images in computers just like puzzle blocks. The most commonly used code for encoding colors is the RGB code. RGB stands for **Red**, **Green**, **Blue**. It refers to system for representing the colors to be used on a computer display. Red, green and blue can be combined in various proportions to obtain any color in the visible spectrum. Each of the three color components is codified by a number, whose value ranges from 0 to 255. Thus, each pixel is defined by a set of three numbers. For example, an RGB code (120, 150, 200) implies that red= 120, green= 150 and blue=200.

Each of these color components are represented by their binary equivalent using 8 bits (8 bits are sufficient to represent 256 numbers i.e., 0 to 255), and then combined together. All the binary coded components are combined from left to right and the resultant RGB code is then stored in the memory.



Source: online

To conclude, images are represented by groups of tiny colored squares called pixels, each of which is coded in numbers that are eventually converted to binary.

4. REFERENCES

Mano, M. Morris, Michael, D. Ciletti, 'Digital Design: With an Introduction to the Verilog HDL', Fifth Edition, Pearson Publications, 2016.

Ferdjallah, Mohammed, 'Introduction to Digital Systems: Modeling, Synthesis and Simulation using VHDL', Wiley Publications, 2011.

Gil, Press, 'A Very Short History of Digitization', www.forbes.com

'History of Computer Memory', Old Dominion University, www.cs.odu.edu

Ayush, Sharma, Article: 'IEEE Standard 754 Floating Point Numbers', GeeksforGeeks, www.geeksforgeeks.org

Shakya, Udayan, 'Binary Numbers and the Working of Computers', Programiz Blog, www.programiz.com

'ASCII printable characters', ASCII-CODE.COM, www.ascii-code.com

https://www.ascii-code.com/characters/printable-characters?view=compact

HTML Color Codes, www.htmlcolorcodes.com