NUMBER SYSTEM

Computers primarily use the binary number system, also known as base-2, as the fundamental numbering system for representing and processing data. In the binary system, there are only two digits: 0 and 1. These two digits are used to represent all data and instructions in a computer's memory and in its central processing unit (CPU).

Here are some key concepts related to the binary number system in computing:

Bits: The basic unit of data in a computer is called a "bit," which is short for "binary digit." A bit can represent either a 0 or a 1.

Bytes: A group of 8 bits is known as a "byte." Bytes are commonly used to represent characters, numbers, and other data. For example, a single character in a computer's memory is typically represented by one byte.

Binary Representation: In binary representation, each digit's position in a number has a power of 2 associated with it. For example, in the binary number 1101, the rightmost digit represents 2^0 (1), the next one to the left represents 2^1 (2), the next represents 2^2 (4), and the leftmost represents 2^3 (8). You can calculate the decimal equivalent of binary numbers using this positional system.

Hexadecimal: While the binary system is fundamental, it is not very convenient for humans to work with. To make it more manageable, hexadecimal (base-16) is often used in computer programming. Hexadecimal uses digits 0-9 and letters A-F to represent values from 0 to 15. Four bits can be represented by a single hexadecimal digit. For example, the binary number 1101 is equivalent to the hexadecimal number D.

Octal: Octal (base-8) is another system sometimes used in older computer systems. It uses digits 0-7 to represent values from 0 to 7. Three bits can be represented by a single octal digit.

Decimal: Decimal (base-10) While computers primarily use binary internally, they often work with decimal numbers as well. It uses digits 0-9 to represent values from 0 to 9. Conversions between binary and decimal are performed as needed for human-readable input and output.

These number systems, along with conversion methods between them, are fundamental to digital computing and are used in various aspects of computer science, including programming, computer architecture, and digital logic design.

NUMBER CONVERSION

Converting between different number systems, such as binary, decimal, hexadecimal, and octal, is a common task in computer science and digital electronics. Here's how you can perform conversions between these number systems:

Binary to Decimal Conversion:
To convert a binary number to decimal, use the positional notation, where each digit's position has a power of 2 associated with it. For example, the binary number 1101 in decimal is:

1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 8 + 4 + 0 + 1 = 13.

Decimal to Binary Conversion:
To convert a decimal number to binary, repeatedly divide the decimal number by 2 and keep track of the remainders. The binary representation is the series of remainders, read in reverse order.
Example: Convert 13 to binary:

13 ÷ 2 = 6 remainder 1
6 ÷ 2 = 3 remainder 0
3 ÷ 2 = 1 remainder 1
1 ÷ 2 = 0 remainder 1
Reading the remainders in reverse gives the binary representation: 1101.

Binary to Hexadecimal Conversion:
To convert a binary number to hexadecimal, group the binary digits into sets of four, starting from the right and pad with leading zeros if necessary. Then, convert each set of four binary digits to a single hexadecimal digit.

Example: Convert 110101101 to hexadecimal:

Pad to make it a multiple of 4: 0011 0101 1010
Convert each group to hexadecimal: 3 5 A
So, 110101101 in binary is equivalent to 35A in hexadecimal.

Hexadecimal to Binary Conversion:
To convert a hexadecimal number to binary, convert each hexadecimal digit to its equivalent 4-bit binary representation.

Example: Convert 35A to binary:

3 in hexadecimal is 0011 in binary.
5 in hexadecimal is 0101 in binary.
A in hexadecimal is 1010 in binary.
So, 35A in hexadecimal is equivalent to 001101011010 in binary.

NUMBER CONVERSION

Decimal to Hexadecimal Conversion and vice versa:

You can convert a decimal number to hexadecimal by first converting it to binary and then converting the binary number to hexadecimal as shown in the previous steps. The reverse conversion can also be done by going through binary.

These conversion methods are fundamental in computer science and are often used when working with different number systems and performing operations on digital data.

To convert an octal number to hexadecimal, you can use an intermediate step by converting the octal number to binary first and then converting the binary number to hexadecimal. Here's how you can do it:

Convert Octal to Binary:
Convert each octal digit to its equivalent 3-bit binary representation. If a digit is not three bits long, pad it with leading zeros.
For example, let's convert the octal number 753 to binary:

7 in octal is 111 in binary.
5 in octal is 101 in binary.
3 in octal is 011 in binary.
So, 753 in octal is equivalent to 111101011 in binary.

Convert Binary to Hexadecimal:
Group the binary digits into sets of four, starting from the right, and pad with leading zeros if necessary. Then, convert each set of four binary digits to a single hexadecimal digit.

For the binary number 111101011, we can group it as follows:
0011 1101 0110
Now, convert each group to hexadecimal:
0011 is 3 in hexadecimal.
1101 is D in hexadecimal.
0110 is 6 in hexadecimal.
So, 753 in octal is equivalent to 3D6 in hexadecimal.