Introduction to Numeral systems

Agenda

01. Introduction and Decimal Numeral System

02. Binary and Octal Numeral Systems

03. Hexadecimal Numeral System and Conversion

04. BCD and Alphanumeric codes

Introduction to Numeral Systems

In this session, we will dive into the fascinating world of numeral systems, which are fundamental to how we represent numbers in computers and mathematics.

We will explore various numeral systems, including decimal, binary, octal, and hexadecimal, and understand their significance in programming and digital systems.

Number System

• Decimal	0–9
• Binary	0-1
• Octal	0-7
• Hexadecimal	0-F

Why Numeral Systems matters?

Numeral systems are foundational to various fields, especially in the realm of computing, mathematics, and technology. Here are some examples why numeral systems matter:

- Representation of Numbers: Numeral systems provide a structured way to represent and communicate numbers. Different numeral systems use distinct symbols and rules to represent numeric values. This diversity of representation is crucial for different contexts and applications.
- Digital Computing:In digital computing, all information is ultimately represented in binary (base-2) form. Understanding binary is essential for designing and programming computers. Binary representation allows electronic devices to store and manipulate data using electronic components.
- Cryptographic Algorithms: Cryptography relies on numeral systems for secure data encryption and decryption. Bit-level operations and numerical representations are at the core of cryptographic algorithms, ensuring data privacy and security.

In essence, numeral systems bridge the gap between human understanding and computer processing. They enable efficient data storage, manipulation, and communication, making them essential for a wide range of applications in technology and beyond.

Decimal Numeral System

What is decimal?

Decimal Numeral System

What is decimal?

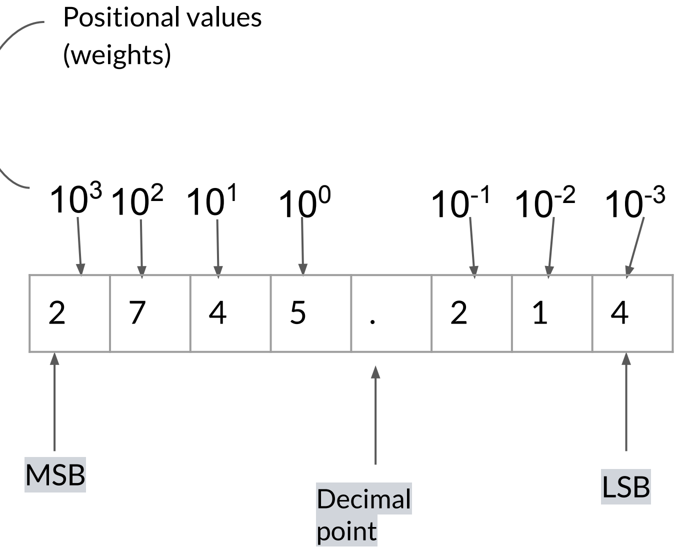
Decimal is a numbering system that uses a base-10 representation for numeric values.The decimal system is also referred to as the Hindu-Arabic system. Additionally, the term decimal is often used to refer to a fraction that is represented as a number in the decimal system, such as 19.368.

The decimal system consists of 10 single-digit numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. The number 9 is followed by 10, which is followed by 11, then 12, and so on. The number on the left is incremented by 1 each time the digit to the right goes beyond 9. For example, 20 follows 19, 30 follows 29, 100 follows 99, and 1000 follows 999.

The decimal system includes both integers and fractions. A fraction in decimal notation includes a decimal point, followed by the fractional component. For example, 5 3/4 can be represented as 5.75 in decimal notation.

Decimal Numeral System

Example: Express the decimal number 2745.214 as a sum of the values of each digit



Note:
MSB= Most significant Bit
LSB=Least significant Bit

Decimal Numeral System

Express the decimal number 2745.214 as a sum of the values of each digit

Solution:

weights	10^3	10^2	10^1	10^0		10^{-1}	10^{-3}	10^{-2}
	2	7	4	5	.	2	1	4

$$2745.214_{10} = (2x10^3) + (7x10^2) + (4x10^1) + (5x10^0) + (2x10^{-1}) + (1x10^{-2}) + (4x10^{-3})$$

Talent Accelerator powered by START

Examples of real-world scenarios where the decimal system is used.

The decimal system, also known as the base-10 numbering system, is the most commonly used numerical system in our everyday lives. It has ten digits (0-9) and is used for a wide range of applications. Here are some real-world scenarios where the decimal system is used:

- Currency and Finance: The decimal system is used worldwide for currency and financial transactions. Money is typically represented in decimal format, where dollars and cents are separated by a decimal point (e.g., $25.99).
- Cooking and Baking: Recipes often use decimal-based measurements for ingredients. For example, a recipe might call for 0.5 cups of flour or 1.25 teaspoons of salt.
- Stock Market: Stock prices are commonly quoted and tracked using decimal numbers. For instance, a stock might be valued at $150.25 per share.
- Calendars and Dates: Dates are represented using the decimal system. For example, July 4th is written as 07/04, where the numbers are in decimal format.
- Phone Numbers: Telephone numbers are usually written in decimal form. Each digit represents a numeric value from 0 to 9.

Here are some examples but there is much more...

zalando

Decimal Numeral System examples

Questions:

Express the decimal number below as a sum of the values of each digit :

A. 64
B. 452.67
C. 12.834

Break 10 minutes

Agenda

Talent Accelerator powered by START

01. Introduction and Decimal Numeral System

02. Binary and Octal Numeral Systems

03. Hexadecimal Numeral System and Conversion

04. BCD and Alphanumeric codes

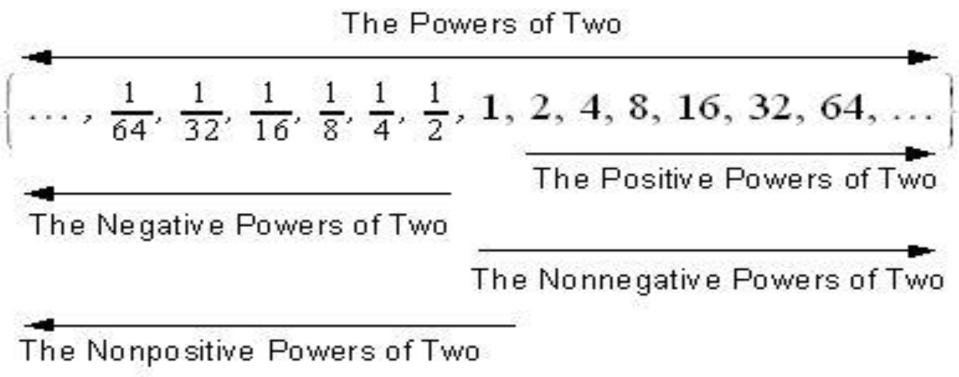
zalando

Introduction to Binary Number

The Binary Number System consist of only two digits -0 and 1.

All digital computer use this number system and convert the data input from the decimal format into its binary equivalent.The Binary number system is another way to represent quantities. There are 1(HIGH) and 0 (LOW).

The binary numbering system has base 2 with each position weighted by a factor of 2 :



Binary Numeral System

The position of each digit(bit) in a binary number can be assigned a weight.

Example: 1011.101

weights	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
	1	0	1	1	.	1	0	1

1011.101 is a binary number
1 is a digit, 0 is a digit , 1 is a digit ....

Binary Numeral System

Example:

Convert the binary whole number 1101101 to decimal:

Weight	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Binary number	1	1	0	1	1	0	1

$$\begin{aligned} 1101101 &= 2^6 + 2^5 + 2^3 + 2^2 + 2^0 \\ &= 64 + 32 + 8 + 4 + 1 \\ &= 109 \end{aligned}$$

Binary to Decimal

Example:

Weight:	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
Binary number:	1	0	1	1	.	1	0	1

$$
\begin{aligned}
1011.101_2 &= (1\text{x}2^3)+(0\text{x}2^2)+(1\text{x}2^1)+(1\text{x}2^0) \\
&\quad +(1\text{x}2^{-1})+(0\text{x}2^{-2})+(1\text{x}2^{-3}) \\
&= 8+0+2+1+0.5+0+0.125 \\
&= 11.625_{10}
\end{aligned}
$$

Binary to decimal

Questions:

Convert the binary number below to decimal:

A. 0.1101
B. 1010111
C. 101011.011

Break 10 minutes

Binary Weight

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	.	2^{-1}	2^{-2}
256	128	64	32	16	8	4	2	1	.	0.5	0.25

The left most bit is the MSB and the right most bit is LSB.

The value of a bit is determined by its position in the number.

In general, with n bits you can count up to a number equal to $2^n - 1$.

For example:
With three bits (n = 3), you can count from 0 to 7.

$=2^n - 1$

$=2^3 - 1$

$= 7$

Decimal Base-10	Binary Base-2
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

Binary Weight table

Decimal	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Decimal to Binary Conversion

There are two ways to convert a decimal number to binary number:

- Sum of weights method(also known as Weighted Sum Method):
    - This method involves representing the decimal number as a sum of power of 2.Each digit in the binary representation corresponds to a specific power of 2, and you calculate the binary digits by finding the largest power of 2 that fits within the decimal number and subtracting it.
- Repeated Division by 2:
    - This method is similar to the one mentioned above. It involves repeatedly dividing the decimal number by 2 and recording the remainders. The remainders are read in reverse order to get the binary representation.

Sum of Weight method

Decimal Number -> Binary Number

To get the binary number for a given decimal number, find the binary weight that add up to the decimal number.

Example:

The decimal number is 11. So you can expressed as the sum of binary weight as follows:

$11 = 8 + 2 + 1$ or $11 = 2^3 + 2^1 + 2^0$

Placing 1st in appropriate weight:

2^3	2^2	2^1	2^0
8	4	2	1
1	0	1	1

So,the binary number for 11 is 1011

Sum of Weight method

Questions:

Convert the decimal number below to binary using Sum of Weight method:

A. 42
B. 157
C. 73
D. 255

Repeated Division by 2 method

Decimal Number -> Binary Number

It requires repeatedly dividing the decimal number by 2 and writing down the reminder after each division until a quotient of 0 is obtained.

Example:

Convert the decimal number 12 to binary:

	Remainder	
12÷ 2 = 6	0	<- LSB
6÷2 = 3	0	
3÷2 =1	1	
1÷2= 0	1	<- MSB

Answer: 1100

Note: Stop when the whole number quotient is 0

Repeated Division by 2 method

Questions:

Convert the decimal number below to binary using Repeated Division by 2 method:

A. 25
B. 85
C. 129
D. 63

Converting Decimal fractions to Binary

There are two ways to convert a decimal number to binary number:

- Sum of weights method(also known as Weighted Sum Method)
- Repeated Multiplication by 2

Converting Decimal fractions to Binary

Converting decimal fractions to binary using Repeated multiplication by 2:

- Multiplying by 2 until the fractional product is zero or until the desired number of decimal places is reached.
- The carry digits or carries generated by the multiplications produce the binary number.
- The first carry produced in the MSB, and the last carries is the LSB.

Converting Decimal fractions to Binary

To get the binary number for a given fractional decimal, find the binary weights that up to the decimal number.

Note: Negative power of two

(Fractional number)

2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
1/2	1/4	1/8	1/16	1/32	1/64
0.5	0.25	0.125	0.0625	0.03125	0.015625

Converting Decimal fractions to Binary

2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
1/2	1/4	1/8	1/16	1/32	1/64
0.5	0.25	0.125	0.0625	0.03125	0.015625

Example:

The decimal number is 0.625. So you can expressed as the sum of binary weight as follows:

$0.625 = 0.5 + 0.125$ or $0.625 = 2^{-1} + 2^{-3}$

Placing 1st in appropriate weight:

2^{-1}	2^{-2}	2^{-3}
1	0	1

So, the binary number for 0.625 is 1.101

Converting Decimal fractions to Binary

Example:

Convert decimal numbers 0.3125 to binary :

		Carry
0.3125 x 2 = 0.625	$\Longrightarrow$	0
0.625 x 2 = 1.25	$\Longrightarrow$	1
0.25 x 2 = 0.50	$\Longrightarrow$	0
0.5 x 2 = 1.00	$\Longrightarrow$	1

MSB 0.0101 LSB

Break 10 minutes

Binary Arithmetic

Binary Arithmetic involves performing mathematical operations (addition, subtraction, multiplication and division) using binary numbers. Just like decimal arithmetic, binary arithmetic follows similar rules but with binary digits (0 and 1).

Binary Addition

The four basic rules for adding binary digits:

Case	A+B	Sum	Carry
1	0+0	0	0
2	0+1	1	0
3	1+0	1	0
4	1+1	0	1

Note: In fourth case, a binary addition is creating a sum of (1 + 1 = 10), where 0 is written in the sum column and a carry of 1 over to the next column.

Example:

0011010 + 001100 = 00100110

```
  1 1          carry
 0 0 1 1 0 1 0  = 26₁₀
+0 0 0 1 1 0 0  = 12₁₀
---------------
 0 1 0 0 1 1 0  = 38₁₀
```

Binary Addition

The four basic rules for adding binary digits:

Case	A+B	Sum	Carry
1	0+0	0	0
2	0+1	1	0
3	1+0	1	0
4	1+1	0	1

Questions:

Add the following binary numbers:

A. 11 + 11
B. 100+ 10
C. 111 + 11
D. 110 + 100

Binary Subtraction

The four basic rules for subtracting binary digits:

Case	A-B	Subtract	Borrow
1	0-0	0	0
2	1-0	1	0
3	1-1	0	0
4	0-1	0	1

Example:

0011010 - 001100 = 00001110

$$
\begin{array}{rl}
1\ 1 & \text{borrow} \\
0\ 0\ \cancel{1}\ \cancel{1}\ 0\ 1\ 0 & = 26_{10} \\
-\ 0\ 0\ 0\ 1\ 1\ 0\ 0 & = 12_{10} \\
\hline
0\ 0\ 0\ 1\ 1\ 1\ 0 & = 14_{10}
\end{array}
$$

Binary Subtraction

The four basic rules for subtracting binary digits:

Case	A-B	Subtract	Borrow
1	0-0	0	0
2	1-0	1	0
3	1-1	0	0
4	0-1	0	1

Questions:

Subtract the following binary numbers:

A. 11 - 01
B. 1001 -0111
C. 10101-00111

Binary Multiplication

The four basic rules for multiplication in binary digits:

Case	A x B	Multiplication
1	0 x 0	0
2	0 x 1	0
3	1 x 0	0
4	1 x 1	1

Example:

0011010 x 001100 = 100111000

```
       0 0 1 1 0 1 0   = 26₁₀
     x 0 0 0 1 1 0 0   = 12₁₀
  ---------------------
       0 0 0 0 0 0 0
     0 0 0 0 0 0 0
   0 0 1 1 0 1 0
 0 0 1 1 0 1 0
  ---------------------
 0 1 0 0 1 1 1 0 0 0   = 312₁₀
```

Binary Division

Division in binary follows the same procedure as division in decimal.

Example:

101010 / 000110 = 000111

```
                111       = 7₁₀
000110 ) 101010           = 42₁₀
        -110              = 6₁₀
        ----
         1001
         -110
         ----
           110
          -110
          ----
             0
```

First and Second complements of Binary Numbers

The first and second complements of a binary number are concepts used in digital computing to represent negative numbers and perform subtraction.

Finding the first (1st) Complement

The first complement of a binary number is found by changing all 1 to 0 and all 0 to 1.

For example:

1 0 1 1 0 0 1 0 (Binary number)

0 1 0 0 1 1 0 1 (1st complement)

Finding the second (2nd) Complement

The second complement of a binary number is found by adding 1 to the LSB of the first complement.

For example:

Find the second complement of 10110010:

	10110010	(Binary number)
+	01001101	(1st complement)
	1	(Add 1)
	01001110	(Second complement)

Break 10 minutes

Signed Numbers

Digital system, such as the computer, must be able to handle both positive and negative numbers. A signed binary number consists of both sign and magnitude information. The sign indicates whether a number is positive or negative and the magnitude is the value of the number.

There are three form in which signed integer (whole) numbers can be represented in binary:

1. Sign-Magnitude
2. 1st Complement
3. 2nd Complement

The Sign Bit

The left-most bit in a signed binary number is the sign bit, which tells you whether the number is positive or negative.

0 = Positive Number

1 = Negative Number

Sign-Magnitude Form

When a signed binary number is represented in sign-magnitude, the leftmost bit is the sign bit and the remaining bits are the magnitude bits.The magnitude bits are in true (uncomplemented) binary of both positive and negative numbers.

Example: Decimal number +25 is expressed as an 8-bit signed binary number using sign-magnitude form as:

Magnitude-Bit

Sign Bit → 00011001

The Sign Bit

Example:

+25 = 0 0011001

Magnitude bit

Signed bit

-25 = 1 0011001

Magnitude bit

Signed bit

In signed bit magnitude form, a negative number has same magnitude bits as corresponding positive number but the sign bit is a 1 rather than 0.

The Sign Bit First complement form

In the first complement form , negative number is the **first complement of corresponding positive number.**

Example:

Using 8-bits, decimal number for -25 is expressed as first complement of +25.

+25 = 00011001

So, in first complement form:

-25 = first complement of +25

= 11100110

The Sign Bit Second complement form

In the second complement form , negative number is the **second complement of corresponding positive number.**

Example:

Using 8-bits, decimal number for -25 is expressed as second complement of +25.

+25 = 00011001

So, in second complement form:

-25 = second complement of +25

= 11100111

The decimal value of signed numbers

Sign-Magnitude: Decimal value of positive and negative numbers in the sign-magnitude form are determined by summing up the weights in all the magnitude bit positions where there are 0.

Example:

Determine the decimal value of this signed binary number expressed in sign magnitude : 1 0 0 1 0 1 0 1

2^6 2^5 2^4 2^3 2^2 2^1 2^0

0 0 1 0 1 0 1 >> 16 + 4 + 1 = 21

The sign bit is 1 : Therefore, the decimal number is -21

The decimal value of signed numbers

First Compliment: Decimal values of positive numbers in the sign-magnitude form are determined by summing the weights in all the magnitude bit positions where there are 1s and ignoring those positions where there are 0.Decimal values of negative numbers are determined by assigning a negative value to the weight of the sign bit, summing all the weights where there are 1s, and adding 1 to the result.

Example:Determine the decimal value of this signed binary number expressed in first compliment :

A. 00010111

-2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

0 0 0 1 0 1 1 1 >> 16 + 4 + 2 + 1 = +23

B. 11101000

-2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

1 1 1 0 1 0 0 0 >> -128 + 64 + 32 + 8 = -24

Adding the 1 to the result : -24 + 1 = -23

The decimal value of signed numbers

Second complement: Decimal values of positive and negative numbers in the second complement form are determined by summing the weights in all the bit positions where there are 1 and ignoring those positions where there are 0. The weight of the sign bit in a negative number is given in a negative value.

Example:Determine the decimal value of this signed binary number expressed in second compliment :

A. 01010110

$-2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

0 1 0 1 0 1 1 0 >> 64 + 16 + 4 + 2 = +86

B. 10101010

$-2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

1 0 1 0 1 0 1 0 >> -128 + 32 + 8 + 2 = -86

Octal system

The octal system, also known as base-8, is a numeral system that uses 8 distinct symbols to represent numbers. In octal system we use the digits 0 through 7.

Each position in an octal number represents a power of 8.

Last position in an octal number represents an X power of the base 8 .

Octal digits: The digits used in the octal system are 0, 1, 2, 3, 4, 5, 6 and 7.

Octal addition

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

A

Sum

B

To use this table , simply follow the directions used in this example: Add 6_8 and 5_8 . Locate 6 in the A column then locate the 5 in the B column. The point in 'sum' area where these two columns intersect is the 'sum' of two numbers:

$$6_8 + 5_8 = 13_8$$

Example:

$456_8 + 123_8 = 601_8$

$$\begin{array}{rl} 1\,1 & \text{carry} \\ 4\,5\,6 & = 302_{10} \\ +1\,2\,3 & = 83_{10} \\ \hline 6\,0\,1 & = 385_{10} \end{array}$$

Octal subtraction

The subtraction of octal numbers follows the same rules as the subtraction of numbers in any other number system. The only variation is in borrowed number. In the decimal system, you borrow a group of 10_{10}. In the binary system, you borrow a group of 2_{10}. In the octal system you borrow a group of 8_{10}.

Example:

$456_8 - 173_8 = 333_8$

8	borrow
^{3}4 5 6	$= 302_{10}$
- 1 7 3	$= 123_{10}$
2 6 3	$= 179_{10}$

Break 10 minutes

Agenda

01. Introduction and Decimal Numeral System

02. Binary and Octal Numeral Systems

03. Hexadecimal Numeral System and Conversion

04. BCD and Alphanumeric codes

Talent Accelerator powered by START

Hexadecimal System

The hexadecimal number uses base 16. It has 16 possible digit symbols. It uses the digits 0 through 9 plus A , B , C , D , E and F as the 16 digit symbols.

The digit positions are weighted as power of 16 as shown below, rather than as power of 10 as in the decimal system.

16^4	16^3	16^2	16^1	16^0	16^{-1}	16^{-2}	16^{-3}	16^{-4}

zalando

Table

Decimal (Base 10)	Binary (Base 2)	Hexadecimal (Base 16)
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

Binary to Hexadecimal Conversion

1. Group the binary number into group of 4-bits
2. Each group is converted to its equivalent HEX digit.
3. Zero are added as needed to complete a 4-bits group

Example:

Convert 1100101001010111 to hex

1100	1010	0101	0111
C	A	5	7

Note: use the table one slide before

Hexadecimal to binary conversion

Is a reverse process from Binary to HEX conversion

Example:

Determine binary numbers for $10A4_{16}$

1	0	A	4
0001	0000	1010	0100

Note: use the table two slides before

Hexadecimal to decimal conversion

Multiply the decimal values of each hex digit by its weight and then take the sum.

Example:

Convert $1C_{16}$ to decimal

$$1C_{16} = (1 \times 16) + (C \times 1)$$

$$= (1 \times 16) + (12 \times 1)$$

$$= 28_{10}$$

Decimal (Base 10)	Binary (Base 2)	Hexadecimal (Base 16)
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

$C_{16} = 12_{10}$

Decimal to Hexadecimal conversion

Talent Accelerator powered by START

Example:

Convert 650_{10} to Hex:

$\frac{650}{16}$ =40.625 -------------> 0.625 x 16 = 10 ⇒ **A**

$\frac{40}{16}$ =2.5 -------------------------> 0.5 x 16 = 8 ⇒ **8**

$\frac{2}{16}$ =0.125 ----------------> 0.125 x 16 = 2 ⇒ **2**

Stop when whole number quotient is zero

MSB 2 8 A LSB

zalando

Agenda

Talent Accelerator
powered by START

01. Introduction and Decimal Numeral System

02. Binary and Octal Numeral Systems

03. Hexadecimal Numeral System and Conversion

04. BCD and Alphanumeric codes

zalando

Binary Coded Decimal (BCD)

BCD stands for Binary-Coded Decimal, and it is a binary encoded representation of decimal numbers. In BCD each decimal digit is represented by a fixed number of binary digits (usually 4 bits).

This encoding allows easy conversion between binary and decimal, making it suitable for applications where decimal numbers need to be processed using binary-based systems.

Binary Coded Decimal (BCD)

Example:

To convert the number 874_{10} to BDC:

8	7	4		
1000	0111	0100	=	100001110100_{BCD}

- Each decimal digit is represented using 4 bits
- Each 4-bit group can never be greater than 9.
- Reverse the process to convert BCD to decimal

Decimal Digit	BCD			
	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Binary Coded Decimal (BCD)

Designation 8421 indicate the binary weight of 4-bits. (2^3, 2^2, 2^1, 2^0)

Decimal Digit	BCD			
	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Alphanumeric codes

Alphanumeric codes are a way to represent both numbers and letters (alphanumeric characters) using binary digits or other encoding schemes. These codes are used to facilitate data communication and storage, especially when a mixture of letters, numbers, and other symbols need to be represented in a digital format.

ASCII Code

ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange):

- ASCII is a widely used character encoding standard that represents alphanumeric characters, control characters, and some special symbols using 7-bit binary codes.
- It includes codes for uppercase and lowercase letters, digits, punctuation marks, and control characters (e.g., newline, tab).
- For example, the ASCII code for uppercase 'A' is 65, and the ASCII code for lowercase 'a' is 97.

ASCII codes provide a standardized way to represent characters across different computers and communication systems. They are still widely used in programming, data transmission, and text processing, though they are limited to representing a smaller set of characters compared to more modern encodings like Unicode.

Thank you for your attention!