Number system and codes

Chapter : 2

2 state Device

In digital devices, there are only two states: on and off. Using only these two states, devices can communicate a great deal of data and control various other devices. In binary, these states are represented as a 1 or 0. Binary 1 is typically considered a logic high, and 0 is a logic low.

Number System

In a digital system, the system can understand only the optional number system. In these systems, digits symbols are used to represent different values, depending on the index from which it settled in the number system.

In simple terms, for representing the information, we use the number system in the digital system.

The digit value in the number system is calculated using:

1. The digit
2. The index, where the digit is present in the number.
3. Finally, the base numbers, the total number of digits available in the number system.

Types of Number System

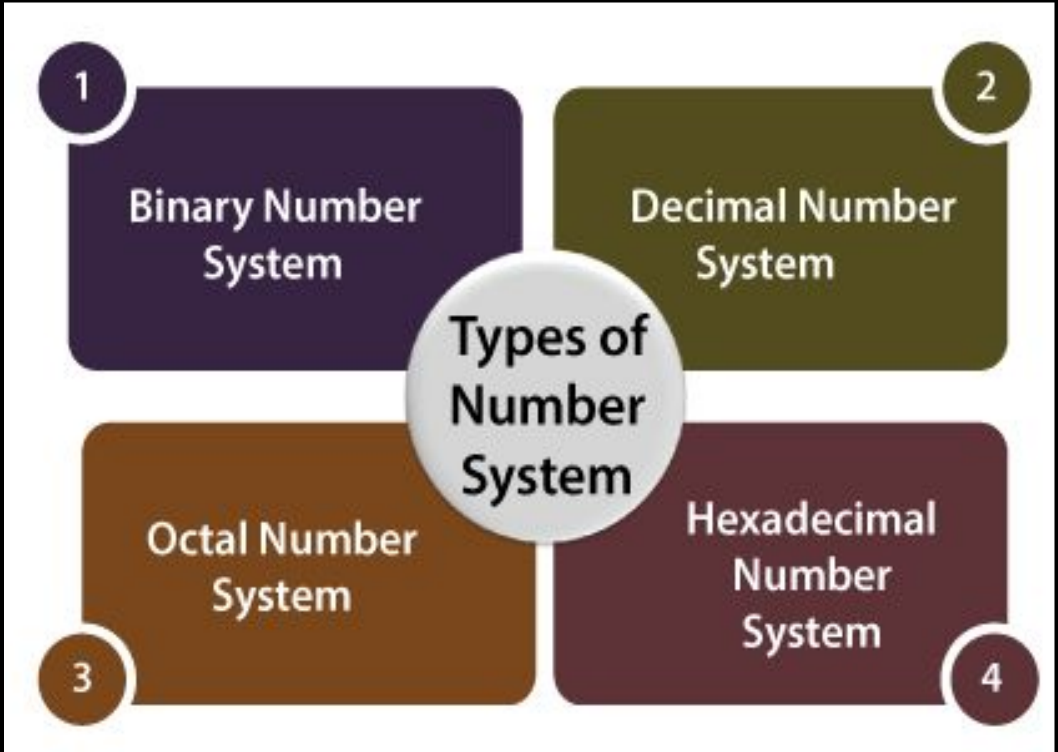
In the digital computer, there are various types of number systems used for

Number System

Types of Number System

In the digital computer, there are various types of number systems used for representing information.

1. Binary Number System
2. Decimal Number System
3. Hexadecimal Number System
4. Octal Number System



Binary Number System

The binary number system, used in digital computers, consists of only two digits: **0** and **1**. These digits represent the absence (**0**) or presence (**1**) of electronic pulses. Each digit is called a **bit**, a group of four bits is a **nibble**, and a group of eight bits is a **byte**.

The position of a digit in a binary number indicates a specific power of the base **2**, with the rightmost position being 202^020 and increasing powers of 2 to the left. Key characteristics of the binary system are:

- It uses only two values, 0 and 1.
- It is also known as the base-2 number system.

Examples:

Binary numbers include 10100_2,11011_2,11001_2,000101_2,011010_2

Decimal Number System

The decimal number system, commonly used in daily life, consists of **ten digits (0 to 9)** with a base of **10**. Each position to the left of the decimal point represents powers of 10, such as units (10^0), tens ($10^{\wedge 1}$), hundreds ($10^{\wedge 2}$), thousands ($10^{\wedge 3}$), and so on.

The digit values range from **0** (minimum) to **9** (maximum). For example, in the decimal number 25**41**, 2 in the thousand position($10^{\wedge 3}$), 5 in the hundreds position($10^{\wedge 2}$),4 in the tens position ($10^{\wedge 1}$), 1 in the units position ($10^{\wedge 0}$)

The total value is calculated accordingly.

$$(2\times1000) + (5\times100) + (4\times10) + (1\times1)$$

$$(2\times10^3) + (5\times10^2) + (4\times10^1) + (1\times10^0)$$

$$2000 + 500 + 40 + 1$$

$$2541$$

Octal Number System

The octal number system has base 8(means it has only eight digits from 0 to 7). There are only eight possible digit values to represent a number. With the help of only three bits, an octal number is represented. Each set of bits has a distinct value between 0 and 7.

Below, we have described certain characteristics of the octal number system:

Characteristics:

1. An octal number system carries eight digits starting from 0, 1, 2, 3, 4, 5, 6, and 7.
2. It is also known as the base 8 number system.
3. The position of a digit represents the 0 power of the base(8). Example: 80
4. The position of the last digit represents the x power of the base(8). Example: 8x, where x represents the last position, i.e., 1

Examples:

(273)8, (5644)8, (0.5365)8, (1123)8, (1223)8.

Hexadecimal Number System

The number system has a base of 16 means there are total 16 symbols(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) used for representing a number. The single-bit representation of decimal values 10, 11, 12, 13, 14, and 15 are represented by A, B, C, D, E, and F. Only 4 bits are required for representing a number in a hexadecimal number. Each set of bits has a distinct value between 0 and 15. There are the following characteristics of the hexadecimal number system:

Characteristics:

1. It has ten digits from 0 to 9 and 6 letters from A to F.
2. The letters from A to F defines numbers from 10 to 15.
3. It is also known as the base 16 number system.
4. In hexadecimal number, the position of a digit represents the 0 power of the base(16). Example: 160
5. In hexadecimal number, the position of the last digit represents the x power of the base(16). Example: 16x, where x represents the last position, i.e., 1

Examples:

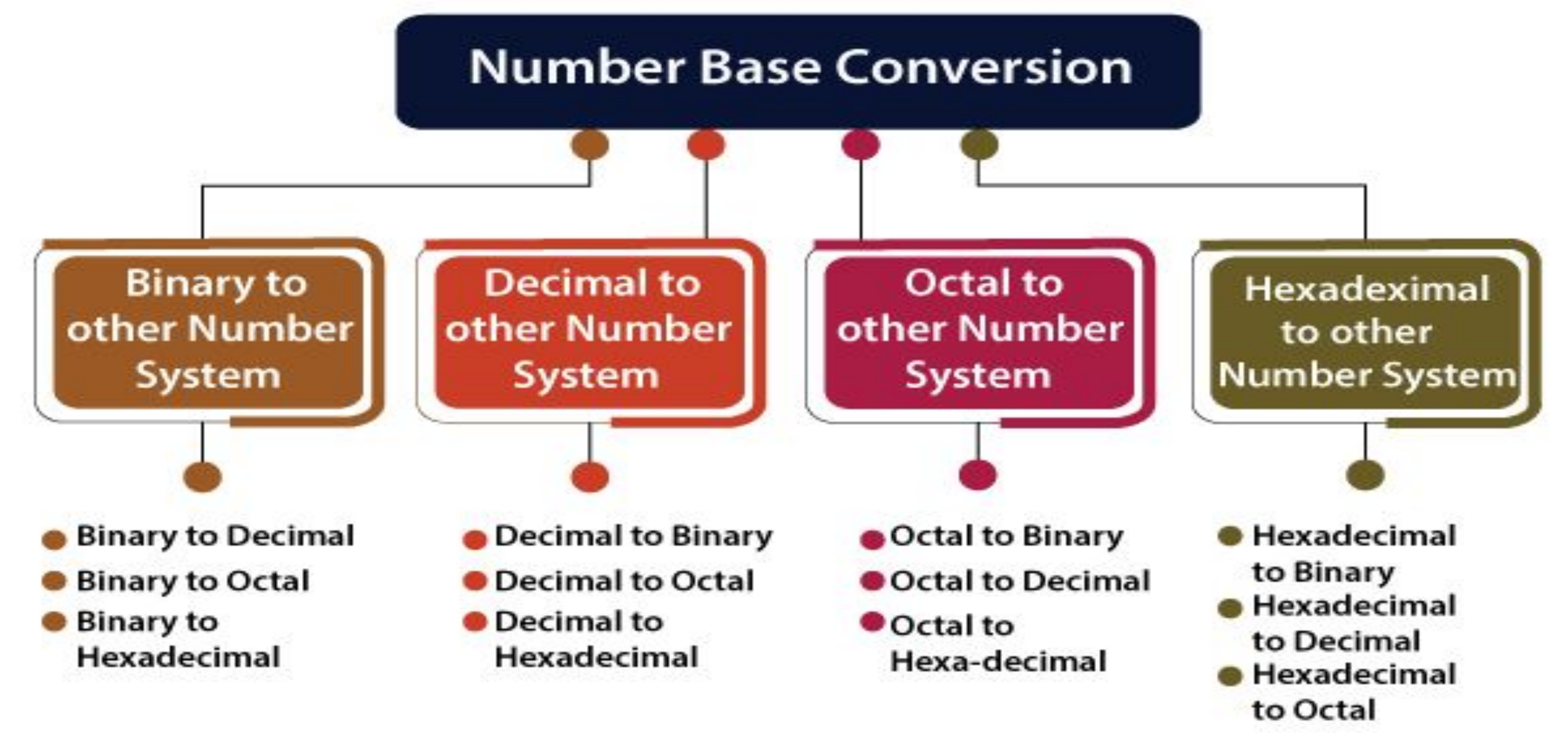
$(FAC2)_{16}$, $(564)_{16}$, $(0ABD5)_{16}$, $(1123)_{16}$, $(11F3)_{16}$.

Conversion among different Number system

As, we have four types of number systems so each one can be converted into the remaining three systems. There are the following conversions possible in Number System

1. Binary to other Number Systems.
2. Decimal to other Number Systems.
3. Octal to other Number Systems.
4. Hexadecimal to other Number Systems.

Conversion among different Number system



Binary to other number system conversion

There are three conversions possible for binary number, i.e., binary to decimal, binary to octal, and binary to hexadecimal. The conversion process of a binary number to decimal differs from the remaining others. Let's take a detailed discussion on Binary Number System conversion.

Binary to Decimal Conversion

The process of converting binary to decimal is quite simple. The process starts from multiplying the bits of binary number with its corresponding positional weights. And lastly, we add all those products.
Let's take an example to understand how the conversion is done from binary to decimal.
Example 1: (10110.001)2

We multiplied each bit of (10110.001)2 with its respective positional weight, and last we add the products of all the bits with its weight.

$$(10110.001)_2=(1\times2^4)+(0\times2^3)+(1\times2^2)+(1\times2^1)+(0\times2^0)+(0\times2^{-1})+(0\times2^{-2})+(1\times2^{-3})$$

$$(10110.001)_2=(1\times16)+(0\times8)+(1\times4)+(1\times2)+(0\times1)+(0\times\tfrac{1}{2})+(0\times\tfrac{1}{4})+(1\times\tfrac{1}{8})$$

$$(10110.001)_2=16+0+4+2+0+0+0+0.125$$

$$(10110.001)_2=(22.125\ )10$$

Binary to Octal Conversion

The base numbers of binary and octal are 2 and 8, respectively. In a binary number, the pair of three bits is equal to one octal digit. There are only two steps to convert a binary number into an octal number which are as follows:

1. In the first step, we have to make the pairs of three bits on both sides of the binary point. If there will be one or two bits left in a pair of three bits pair, we add the required number of zeros on extreme sides.
2. In the second step, we write the octal digits corresponding to each pair.

Example 1: $(111110101011.0011)_2$

1. Firstly, we make pairs of three bits on both sides of the binary point.

    111 110 101 011.001 1

On the right side of the binary point, the last pair has only one bit. To make it a complete pair of three bits, we added two zeros on the extreme side.

    111 110 101 011.001 100

2. Then, we wrote the octal digits, which correspond to each pair.

$(111110101011.0011)_2$=(7653.14)8

Binary to hexadecimal Conversion

The base numbers of binary and hexadecimal are 2 and 16, respectively. In a binary number, the pair of four bits is equal to one hexadecimal digit. There are also only two steps to convert a binary number into a hexadecimal number which are as follows:

1. In the first step, we have to make the pairs of four bits on both sides of the binary point. If there will be one, two, or three bits left in a pair of four bits pair, we add the required number of zeros on extreme sides.
2. In the second step, we write the hexadecimal digits corresponding to each pair.

Example 1: $(10110101011.0011)_2$

1. Firstly, we make pairs of four bits on both sides of the binary point.

111 1010 1011.0011

On the left side of the binary point, the first pair has three bits. To make it a complete pair of four bits, add one zero on the extreme side.

0111 1010 1011.0011

2. Then, we write the hexadecimal digits, which correspond to each pair.

$(011110101011.0011)_2=(7AB.3)_{16}$

Decimal to Other Conversion

A decimal number can be an integer or a floating-point number. To convert it to another base (**r**), the integer and fractional parts are handled separately:

1. **For the Integer Part**:
   - Divide the number by the base (**r**) repeatedly until the quotient is zero.
   - Note the remainders and write them in reverse order to get the new base equivalent.
2. **For the Fractional Part**:
   - Multiply the fraction by the base (**r**) repeatedly.
   - Record the whole numbers (carries) from each step until the result is zero or the desired precision is reached.
   - Write the carries in normal order for the fractional equivalent.

This process converts both parts into the target base.

Decimal to Binary Conversion

For converting decimal to binary, there are two steps required to perform, which are as follows:

1. In the first step, we perform the division operation on the integer and the successive quotient with the base of binary(2).
2. Next, we perform the multiplication on the integer and the successive quotient with the base of binary(2).

Example 1: $(152.25)_{10}$

Step 1:Divide the number 152 and its successive quotients with base 2.

$(152.25)_{10}=(10011000)_2$

Step 2: Now, perform the multiplication of 0.25 and successive fraction with base 2.

(0.25)10=(.01)2

Decimal to Octal Conversion

For converting decimal to octal, there are two steps required to perform, which are as follows:

1. In the first step, we perform the division operation on the integer and the successive quotient with the base of octal(8).
2. Next, we perform the multiplication on the integer and the successive quotient with the base of octal(8).

Example 1: (152.25)10

Step 1: Divide the number 152 and its successive quotients with base 8.

(152)10=(230)8

Step 2: Now perform the multiplication of 0.25 and successive fraction with base 8.

(0.25)10=(2)8

So, the octal number of the decimal number 152.25 is 230.2

Decimal to Hexadecimal Conversion

For converting decimal to hexadecimal, there are two steps required to perform, which are as follows:

1. In the first step, we perform the division operation on the integer and the successive quotient with the base of hexadecimal (16).
2. Next, we perform the multiplication on the integer and the successive quotient with the base of hexadecimal (16).

Example 1: $(152.25)_{10}$

Step 1: Divide the number 152 and its successive quotients with base 8.

$(152)_{10}=(98)_{16}$

Step 2: Now perform the multiplication of 0.25 and successive fraction with base 16.

$(0.25)_{10}=(4)_{16}$

So, the hexadecimal number of the decimal number 152.25 is 98.4.

Octal to Other Conversion

Like binary and decimal, the octal number can also be converted into other number systems. The process of converting octal to decimal differs from the remaining one. Let's start understanding how conversion is done.

Octal to Decimal Conversion

The process of converting octal to decimal is the same as binary to decimal. The process starts from multiplying the digits of octal numbers with its corresponding positional weights. And lastly, we add all those products.

Let's take an example to understand how the conversion is done from octal to decimal.

Example 1: $(152.25)_8$

Step 1: We multiply each digit of 152.25 with its respective positional weight, and last we add the products of all the bits with its weight.

$(152.25)8=(1\times8^2)+(5\times8^1)+(2\times8^0)+(2\times8^{-1})+(5\times8^{-2})$
$(152.25)8=64+40+2+(2\times{}^{1}/_{8})+(5\times{}^{1}/_{64})$
$(152.25)8=64+40+2+0.25+0.078125$
$(152.25)8=106.328125$

So, the decimal number of the octal number 152.25 is 106.328125

Octal to Binary Conversion

The process of converting octal to binary is the reverse process of binary to octal. We write the three bits binary code of each octal number digit.

Example 1: $(152.25)_8$

We write the three-bit binary digit for 1, 5, 2, and 5.

$(152.25)_8=(001101010.010101)_2$

So, the binary number of the octal number 152.25 is $(001101010.010101)_2$

Octal to Hexadecimal Conversion

For converting octal to hexadecimal, there are two steps required to perform, which are as follows:

1. In the first step, we will find the binary equivalent of number 25.
2. Next, we have to make the pairs of four bits on both sides of the binary point. If there will be one, two, or three bits left in a pair of four bits pair, we add the required number of zeros on extreme sides and write the hexadecimal digits corresponding to each pair.

Example 1: $(152.25)_8$

Step 1:We write the three-bit binary digit for 1, 5, 2, and 5.

$(152.25)_8=(001101010.010101)_2$

So, the binary number of the octal number 152.25 is $(001101010.010101)_2$

Step 2:

1. Now, we make pairs of four bits on both sides of the binary point.

0 0110 1010.0101 01

On the left side of the binary point, the first pair has only one digit, and on the right side, the last pair has only two-digit. To make them complete pairs of four bits, add zeros on extreme sides.

0000 0110 1010.0101 0100

2. Now, we write the hexadecimal digits, which correspond to each pair.

$(0000\ 0110\ 1010.0101\ 0100)2=(6A.54)_{16}$

Hexadecimal to other Conversion

Like binary, decimal, and octal, hexadecimal numbers can also be converted into other number systems. The process of converting hexadecimal to decimal differs from the remaining one. Let's start understanding how conversion is done.

Hexadecimal to Decimal Conversion

The process of converting hexadecimal to decimal is the same as binary to decimal. The process starts from multiplying the digits of hexadecimal numbers with its corresponding positional weights. And lastly, we add all those products.

Let's take an example to understand how the conversion is done from hexadecimal to decimal.

Example 1: $(152A.25)_{16}$

Step 1:

We multiply each digit of 152A.25 with its respective positional weight, and last we add the products of all the bits with its weight.

$$(152A.25)_{16}=(1\times16^3)+(5\times16^2)+(2\times16^1)+(A\times16^0)+(2\times16^{-1})+(5\times16^{-2})$$
$$(152A.25)_{16}=(1\times4096)+(5\times256)+(2\times16)+(10\times1)+(2\times16^{-1})+(5\times16^{-2})$$
$$(152A.25)_{16}=4096+1280+32+10+(2\times{}^{1}\!/\!_{16})+(5\times16^{1/256})$$
$$(152A.25)_{16}=5418+0.125+0.125$$
$$(152A.25)_{16}=5418.14453125$$

So, the decimal number of the hexadecimal number 152A.25 is 5418.14453125

Hexadecimal to Binary Conversion

The process of converting hexadecimal to binary is the reverse process of binary to hexadecimal. We write the four bits binary code of each hexadecimal number digit.

Example 1: $(152A.25)_{16}$

We write the four-bit binary digit for 1, 5, A, 2, and 5.

$(152A.25)16=(0001\ 0101\ 0010\ 1010.0010\ 0101)_2$

So, the binary number of the hexadecimal number 152.25 is $(1010100101010.00100101)_2$

Hexadecimal to Octal Conversion

For converting hexadecimal to octal, there are two steps required to perform, which are as follows:

1. In the first step, we will find the binary equivalent of the hexadecimal number.
2. Next, we have to make the pairs of three bits on both sides of the binary point. If there will be one or two bits left in a pair of three bits pair, we add the required number of zeros on extreme sides and write the octal digits corresponding to each pair.

Example 1: (152A.25)16

Step 1: We write the four-bit binary digit for 1, 5, 2, A, and 5.

(152A.25)16=(0001 0101 0010 1010.0010 0101)2

So, the binary number of hexadecimal number 152A.25 is (0011010101010.010101)2

Step 2:3. Then, we make pairs of three bits on both sides of the binary point.

001 010 100 101 010.001 001 010

4. Then, we write the octal digit, which corresponds to each pair.

(001010100101010.001001010)2=(12452.112)8

So, the octal number of the hexadecimal number 152A.25 is 12452.112

BCD code

BCD (Binary-Coded Decimal) is a method of representing decimal numbers using binary digits. Each decimal digit (0 to 9) is represented as a separate 4-bit binary value. BCD coding is widely used in digital systems where numeric data is displayed or processed in decimal form.

Features of BCD Code:

1. **4-Bit Representation**: Each decimal digit is encoded using 4 bits.
   - Example: 0=0000, 1=0001, 2=0010, ..., 9=1001.
2. **Decimal Digits Only**: BCD represents decimal digits only and does not include binary representations of numbers beyond 9.
3. **Human-Readable**: Makes it easier to interface with devices like calculators, digital clocks, and displays.
4. **Redundancy**: Not all 4-bit combinations (e.g., 1010 to 1111) are valid in BCD, as only 0000 to 1001 are used.

BCD code example

Represent the decimal number 23 in BCD:

Separate the decimal digits: 2 and 3.

Convert each digit to 4-bit binary:

2=0010,

3=0011.

Combine the BCD representation:

$23_{10} = (0010\ \ 0011)_{BCD}$

1. **Example: Binary to BCD Conversion**

Let's convert the binary number $\mathbf{11001_2}$ to its BCD representation.

Step 1: Convert Binary to Decimal

- $11001_2 = 1\times2^4+1\times2^3+0\times2^2+0\times2^1+1\times2^0$

  $=16+8+0+0+1$

  $=25_{10}$

BCD code example

Step 2: Write Each Decimal Digit in BCD

- Decimal 2=0010 (BCD representation of 2)
- Decimal 5=0101 (BCD representation of 5)

Step 3: Combine the BCD Representation

- Combine the BCD for each digit:
  25_{10}=0010 0101 $_{BCD}$

Final Result

The BCD representation of the binary number 11001_2 is **0010 0101** $_{\mathbf{BCD}}$

Gray code

Gray Code, also known as **Reflected Binary Code**, is a binary numeral system where two successive values differ by only **one bit**. It is often used in error correction, digital communication, and positional encoding in rotary encoders, as it minimizes errors when transitioning between values.

Features of Gray Code:

1. **Single Bit Change**: Only one bit changes between consecutive numbers, reducing the chance of errors during transitions.
2. **Binary-Like Representation**: Uses binary digits but follows a different pattern than traditional binary.
3. **Cyclic**: The sequence wraps around, so the last value in the sequence transitions back to the first with a single bit change.

Gray Code Conversion:

From Binary to Gray:

- The most significant bit (MSB) of the Gray code is the same as the binary MSB.
- Each subsequent Gray code bit is found by XORing the corresponding binary bit with the previous binary bit.
  Formula: $G_n = B_n \oplus B_{n+1}$, where B is binary and G is Gray.

From Gray to Binary:

- The MSB of binary is the same as the Gray code MSB.
- Each subsequent binary bit is found by XORing the previous binary bit with the current Gray code bit.
  Formula: $B_n = G_n \oplus B_{n-1}$.

Gray Code Conversion:

1. **Example: Binary to Gray Code Conversion**

   Convert 1011_2 to Gray Code:

   1. Take the MSB as it is: $G_1 = B_1 = 1$.
   2. XOR subsequent bits:
      - $G_2 = B_1 \oplus B_2 = 1 \oplus 0 = 1$,
      - $G_3 = B_2 \oplus B_3 = 0 \oplus 1 = 1$,
      - $G_4 = B_3 \oplus B_4 = 1 \oplus 1 = 0$.
   3. Result: 1011_2 in Gray code is 1110.
2. Example: Gray Code to Binary Conversion

   Let's convert the **Gray code 1110** to its binary equivalent.

   $B_1 = G_1 = 1$ (MSB remains the same)

   $B_2 = G_2 \oplus B_1 = 1 \oplus 1 = 0$

   $B_3 = G_3 \oplus B_2 = 1 \oplus 0 = 1$

   $B_4 = G_4 \oplus B_3 = 0 \oplus 1 = 1$

   Gray code 1110 is equivalent to binary 1011.