Number System

What is Number System

- When we type some letters or words, the computer translates them in numbers as computers can understand only numbers.
- The number system or the numeral system is the system of naming or representing numbers.
- A number system is defined as a system of writing to express numbers.
- It is the mathematical notation for representing numbers of a given set by using digits or other symbols in a consistent manner.
- It also allows us to operate arithmetic operations like addition, subtraction, multiplication and division.
- The value of any digit in a number can be determined by:
  - The digit
  - Its position in the number
  - The base of the number system

Base, Weighting Factor

- In mathematical numeral systems, the radix or base is the number of unique digits, including zero, used to represent numbers in a positional numeral system.
- For the decimal system, the radix is ten, because it uses the ten digits from 0 through 9.
- Based on the base value and the number of allowed digits, number systems are of many types. The four common types of Number System are:
  - **Decimal System (Base 10):** 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
  - **Binary System (Base 2):** 0 and 1
  - **Octal System (Base 8):** 0, 1, 2, 3, 4, 5, 6, 7
  - **Hexadecimal System (Base 16):** 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E , F
- Weighting factor = base raised to the power of position (value)
- Weight of a respective position. In decimal form, weight of each digit increases by a factor of 10 as one moves to the left.
  - **754 = 7*100 + 5 *10 + 4**

Decimal Number System

- Number system with base value 10 is termed as Decimal number system.
- It uses 10 digits i.e. 0-9 for the creation of numbers.
- Here, each digit in the number is at a specific place with place value a product of different powers of 10.
- Here, the place value is termed from right to left as first place value called units, second to the left as Tens, so on Hundreds, Thousands, etc. Here, units has the place value as 10^0, tens has the place value as 10^1, hundreds as 10^2, thousands as 10^3, and so on.
- Example:
  - The decimal number 7235 consists of the digit 5 in the units position, 3 in the tens position, 2 in the hundreds position, and 7 in the thousands position.
  - Its value can be written as:

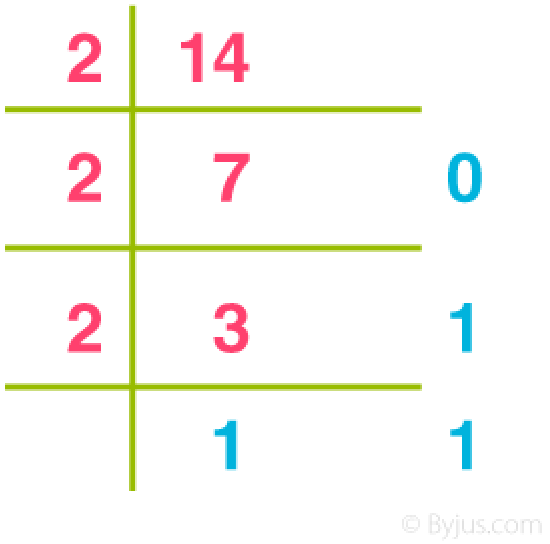
7235 = 7000 + 200 + 30 + 5

= (7 x 1000)+ (2 x 100)+ (3 x 10)+ (5 x 1)

=(7 x 10^3)+ (2 x 10^2)+ (3 x 10^1)+ (4 x 10^0)

Binary Number System

- Number System with base value 2 is termed as Binary number system.
- It uses 2 digits i.e. 0 and 1 for the creation of numbers.
- Binary number system is very useful in electronic devices and computer systems because it can be easily performed using just two states ON and OFF i.e. 0 and 1.
- Decimal Numbers 0-9 are represented in binary as: 0, 1, 10, 11, 100, 101, 110, 111, 1000, and 1001
- For example, 110101 is a binary number.
- We can convert any system into binary and vice versa.
- Example
  - Write $(14)_{10}$ as a binary number.



Octal Number System

- Octal Number System is one in which the base value is 8.
- It uses 8 digits i.e. 0-7 for creation of Octal Numbers.
- Octal Numbers can be converted to Decimal value by multiplying each digit with the place value and then adding the result. Here the place values are 8^0, 8^1, and 8^2.
- Octal Numbers are useful for the representation of UTF8 Numbers.
- Example:
  - $(135)_{10}$ can be written as $(207)_8$
  - $(215)_{10}$ can be written as $(327)_8$

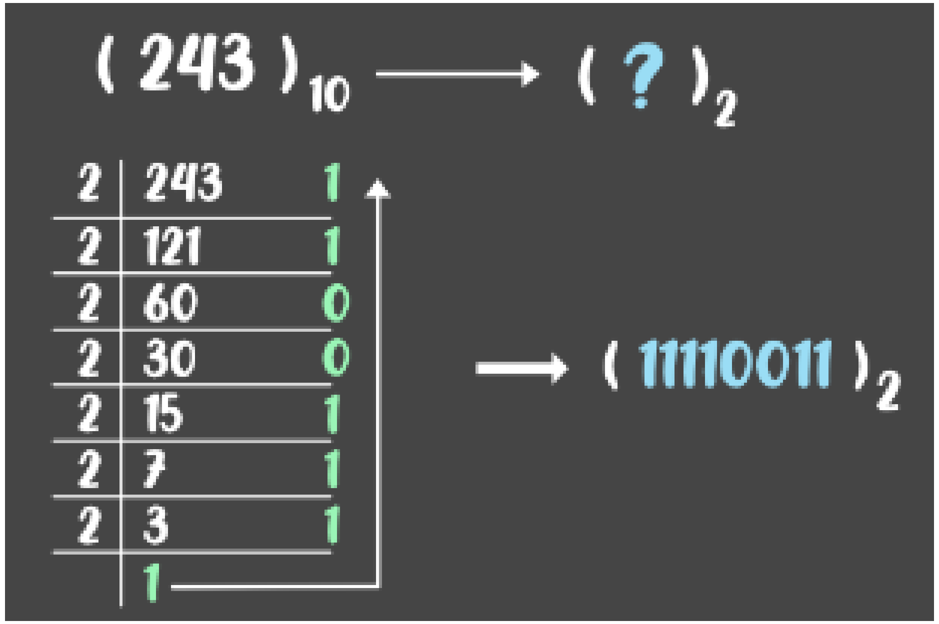
Hexadecimal Number System

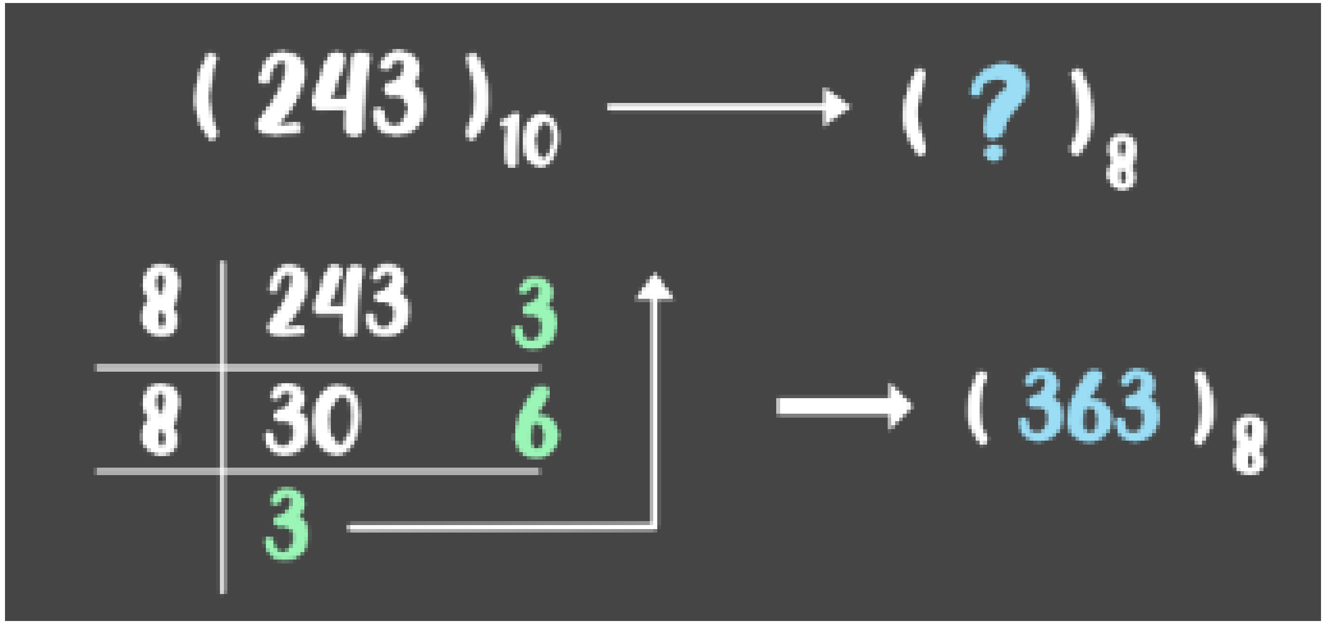
- Number System with base value 16 is termed as Hexadecimal Number System.
- It uses 16 digits for the creation of its numbers.
- Digits from 0-9 are taken like the digits in the decimal number system but the digits from 10-15 are represented as A-F i.e. 10 is represented as A, 11 as B, 12 as C, 13 as D, 14 as E, and 15 as F.
- Hexadecimal Numbers are useful for handling memory address locations.
- Examples:
  - $(255)_{10}$ can be written as $(FF)_{16}$
  - $(1096)_{10}$ can be written as $(448)_{16}$
  - $(4090)_{10}$ can be written as $(FFA)_{16}$

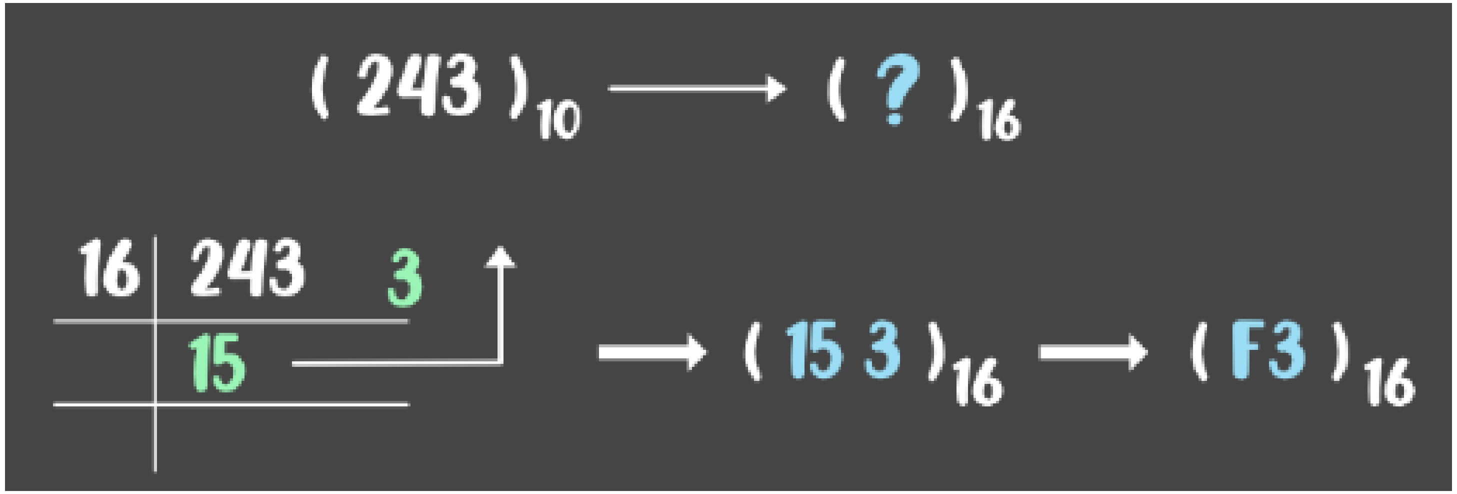
Conversion from Decimal to Other Number System

- The steps to covert decimal number system to other number system are as follows:
  - **Step 1**: Divide the Decimal Number with the base of the number system to be converted to (binary(2) / octal(8) / hexadecimal(16))
  - **Step 2**: The remainder obtained from the division will become the least significant digit of the new number.
  - **Step 3**: The quotient obtained from the division will become the next dividend and will be divided by the same base.
  - **Step 4**: The remainder obtained will become the second least significant digit i.e. it will be added in the left of the previously obtained digit.
- Now, the steps 3 and 4 are repeated until the quotient obtained becomes 0, and the remainders obtained after each iteration are added to the left of the existing digits.
- After all the iterations are over, the last obtained remainder will be termed as the most significant digit.

Conversion from Decimal to Other Number System







Conversion from Binary to Other Number System

Binary to Decimal Conversion

- The steps to convert binary number system to decimal number system are as follows:
  - **Step 1**: Multiply each digit of the Binary number with the place value of that digit, starting from right to left i.e. from LSB to MSB.
  - **Step 2**: Add the result of this multiplication and the decimal number will be formed.

$$(11101011)_2 \longrightarrow (\,?\,)_{10}$$

$$1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

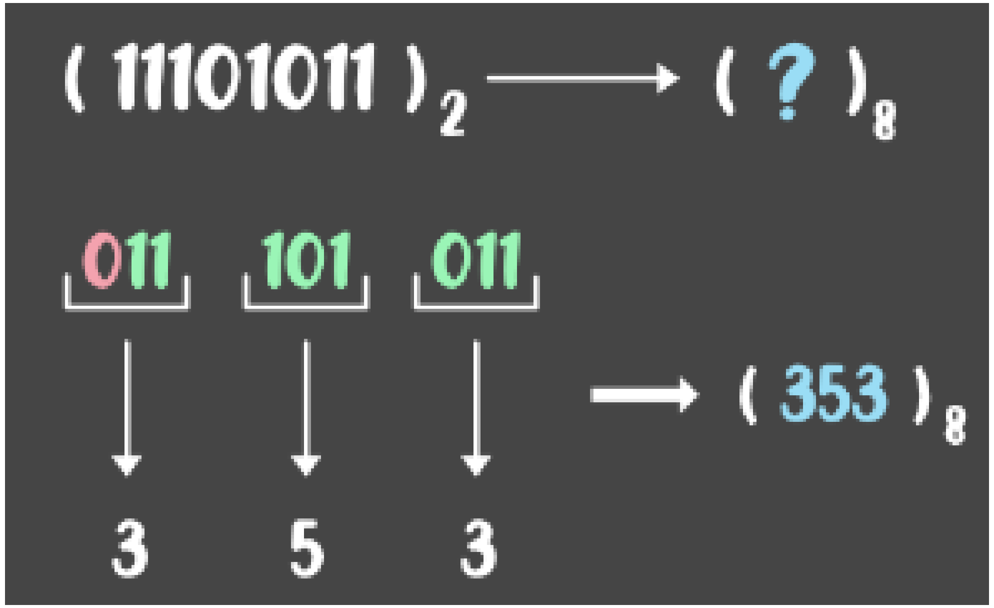
$$128 + 64 + 32 + 0 + 8 + 0 + 2 + 1$$

$$(235)_{10}$$

Conversion from Binary to Other Number System

Binary to Octal Conversion

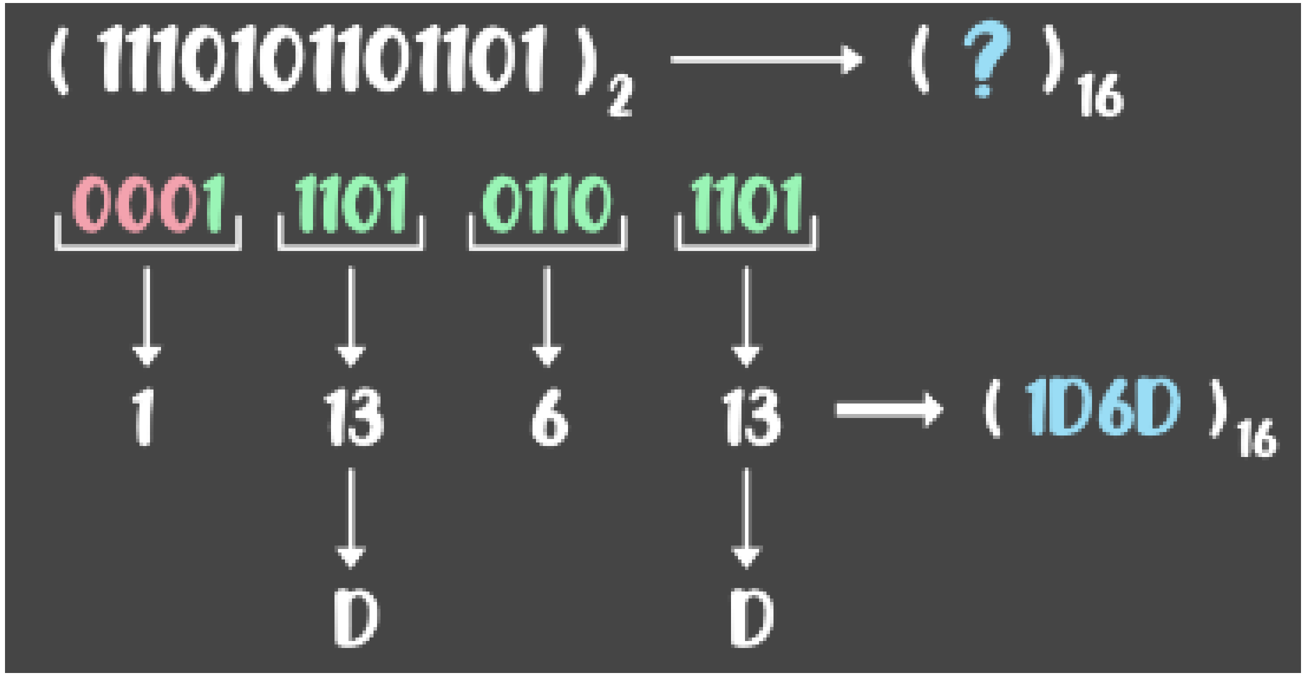
- The steps to convert binary number system to octal number system are as follows:
  - **Step 1**: Divide the binary number into groups of three digits starting from right to left i.e. from LSB to MSB.
  - **Step 2**: Convert these groups into equivalent octal digits



Conversion from Binary to Other Number System

Binary to Hexadecimal Conversion

- The steps to convert binary number system to hexadecimal number system are as follows:
  - **Step 1**: Divide the binary number into groups of four digits starting from right to left i.e., from LSB to MSB.
  - **Step 2**: Convert these groups into equivalent hex digits.



Conversion from Octal to Other Number System

Octal to Decimal Conversion

- The steps to convert octal number system to decimal number system are as follows:
  - **Step 1**: Multiply each digit of the Octal number with the place value of that digit, starting from right to left i.e. from LSB to MSB.
  - **Step 2**: Add the result of this multiplication and the decimal number will be formed.

$$(247)_8 \longrightarrow (?)_{10}$$

$$2 \times 8^2 + 4 \times 8^1 + 7 \times 8^0$$

$$2 \times 64 + 4 \times 8 + 7$$

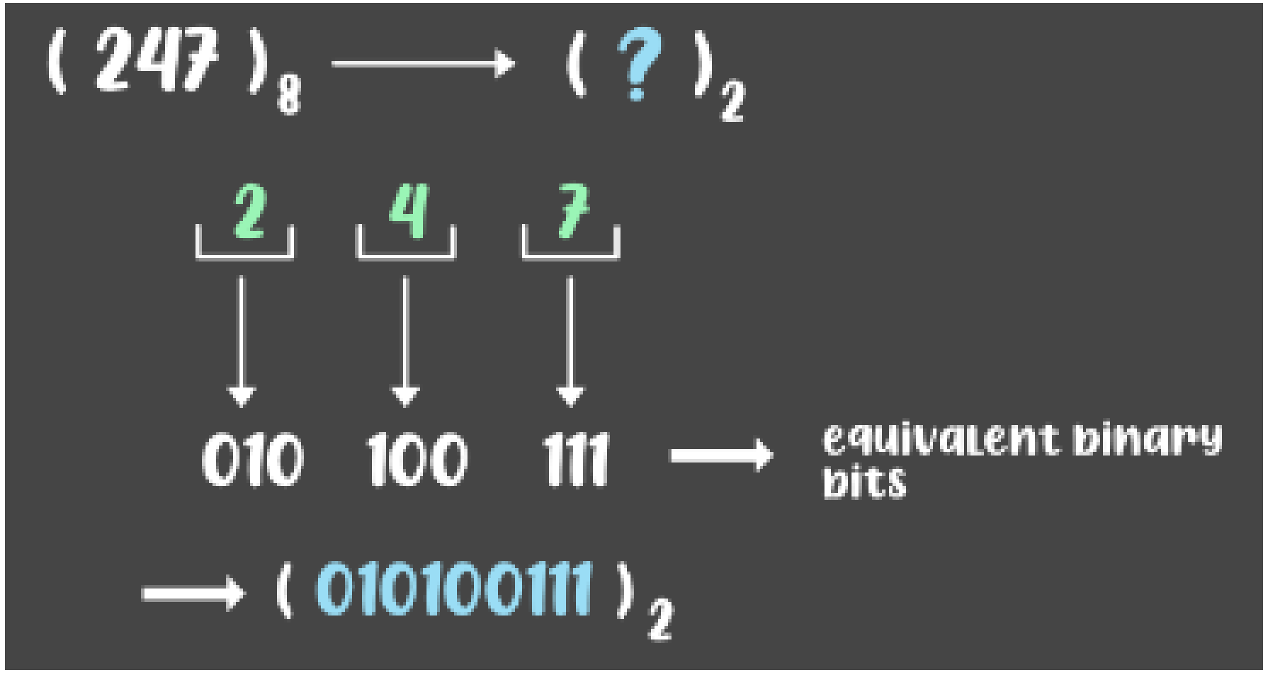
$$128 + 32 + 7$$

$$(167)_{10}$$

Conversion from Octal to Other Number System

Octal to Binary Conversion

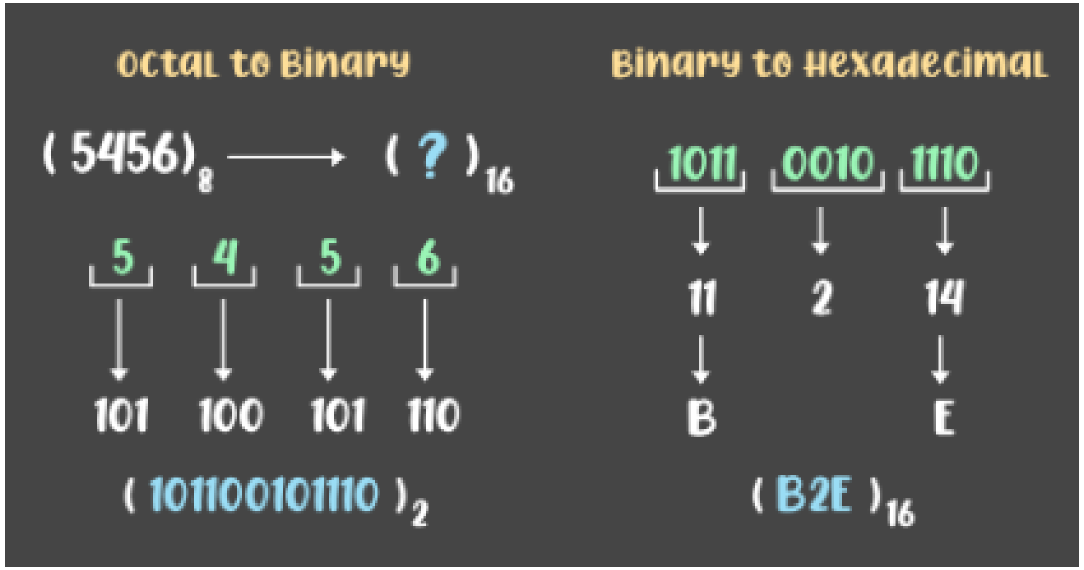
- The steps to convert octal number system to binary number system are as follows:
  - **Step 1**: Write each digit of the octal number separately.
  - **Step 2**: Convert each digit into an equivalent group of three binary digits.
  - **Step 3**: Combine these groups to form the whole binary number.



Conversion from Octal to Other Number System

Octal to Hexadecimal Conversion

- The steps to convert octal number system to hexadecimal number system are as follows:
  - **Step 1:** We need to convert the Octal number to Binary first.
  - **Step 2:** Now to convert the binary number to Hex number, divide the binary digits into groups of four digits starting from right to left i.e., from LSB to MSB.
  - **Step 3:** Add zeros prior to MSB to make it a proper group of four digits(if required)
  - **Step 4:** Now convert these groups into their relevant decimal values.
  - **Step 5:** For values from 10-15, convert it into Hex symbols i.e from A-F



Conversion from Hexadecimal to Other Number System

Hexadecimal to Decimal Conversion

- The steps to convert hexadecimal number system to decimal number system are as follows:
  - **Step 1:** Write the decimal values of the symbols used in the Hex number i.e., from A-F.
  - **Step 2:** Multiply each digit of the Hex number with its place value. starting from right to left i.e., LSB to MSB.
  - **Step 3:** Add the result of multiplications and the final sum will be the decimal number.

$$(8EB4)_{16} \longrightarrow (?)_{10}$$

$$8 \quad 14 \quad 11 \quad 4$$

$$8 \times 16^3 + 14 \times 16^2 + 11 \times 16^1 + 4 \times 16^0$$

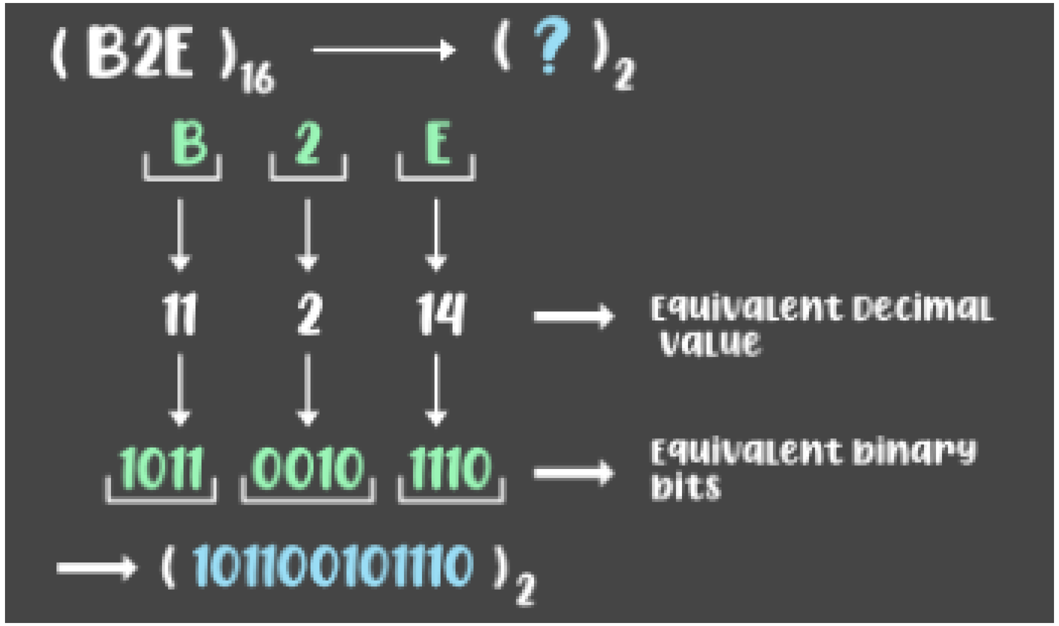
$$32768 + 3584 + 176 + 4$$

$$(36532)_{10}$$

Conversion from Hexadecimal to Other Number System

Hexadecimal to Binary Conversion

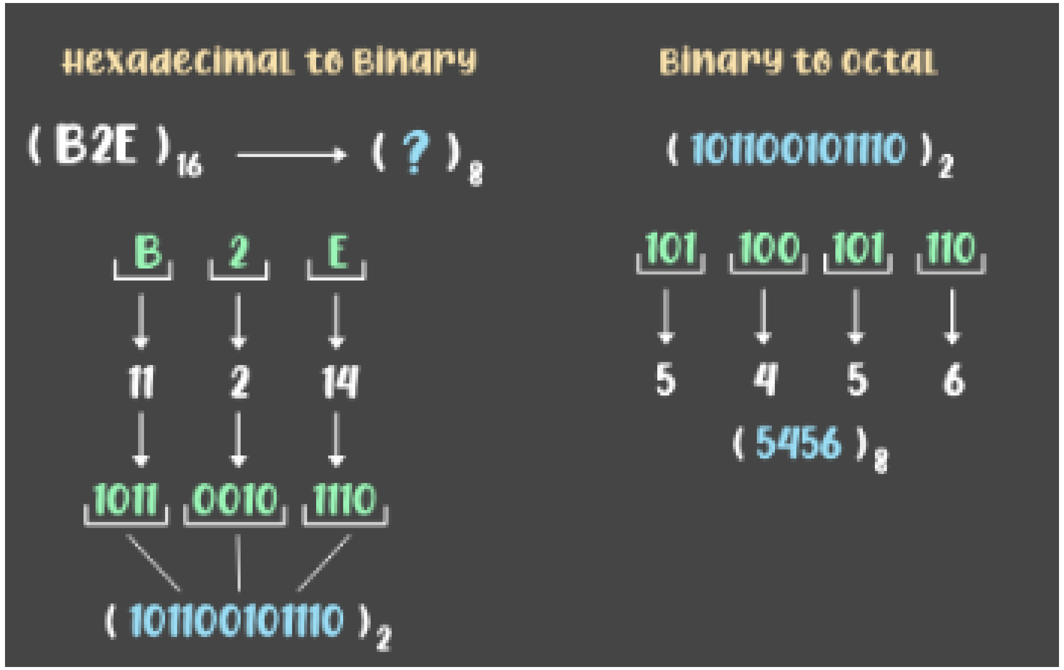
- The steps to convert hexadecimal number system to binary number system are as follows:
  - **Step 1:** Convert the Hex symbols into its equivalent decimal values.
  - **Step 2:** Write each digit of the Hexadecimal number separately.
  - **Step 3:** Convert each digit into an equivalent group of four binary digits.
  - **Step 4:** Combine these groups to form the whole binary number.



Conversion from Hexadecimal to Other Number System

Hexadecimal to Octal Conversion

- The steps to convert hexadecimal number system to octal number system are as follows:
  - **Step 1:** We need to convert the Hexadecimal number to Binary first.
  - **Step 2:** Now to convert the binary number to Octal number, divide the binary digits into groups of three digits starting from right to left i.e., from LSB to MSB.
  - **Step 3:** Add zeros prior to MSB to make it a proper group of three digits(if required).
  - **Step 4:** Now convert these groups into their relevant decimal values.

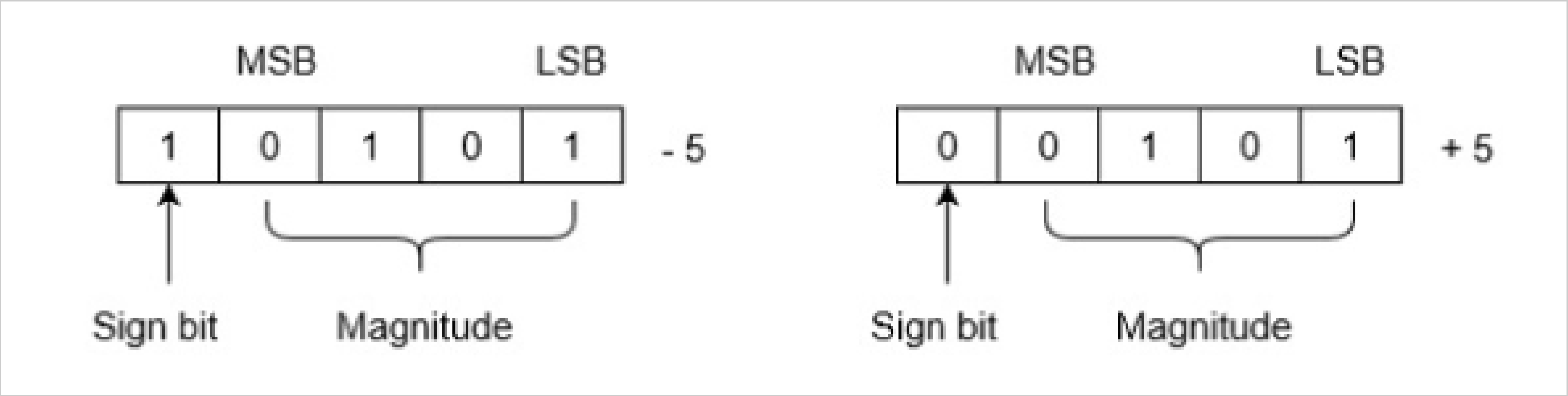


Negative Binary Numbers

- Negative numbers can be distinguishable with the help of extra bit or flag called sign bit or sign flag in Binary number representation system for signed numbers.
- The value of sign bit is 1 for negative binary numbers and 0 for positive numbers.
- When an integer binary number is positive, the sign is represented by 0 and the magnitude by a positive binary number.
- When the number is negative, the sign is represented by 1 but the rest of the number may be represented in one of three possible ways:
  - Sign-Magnitude method
  - 1’s Complement method
  - 2’s complement method

Signed Magnitude Method

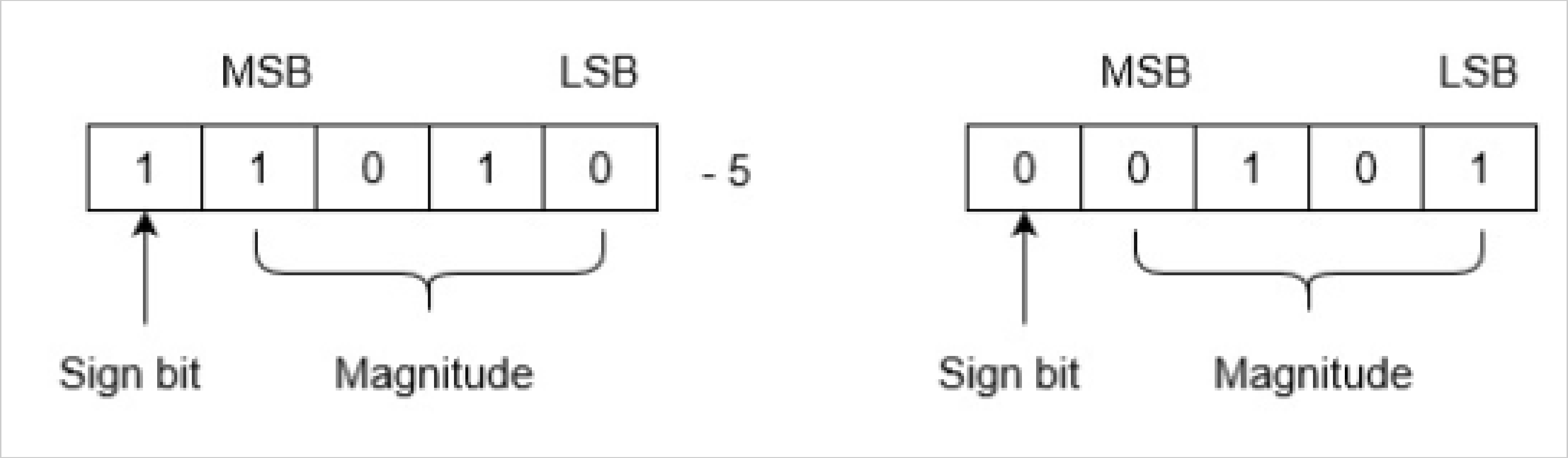
- A number is divided into two parts: Sign bit and Magnitude.
- If the number is positive, then sign bit will be 0 and if number is negative then sign bit will be 1.
- Magnitude is represented with the binary form of the number to be represented.
- **Example:** Let we are using 5 bits register. The representation of -5 and +5 will be as follows:



- **Range (for k bits register):** $-(2^{(k-1)}-1)$ to $+(2^{(k-1)}-1)$
- **Drawback:** 0 has two different representation one is -0 (e.g., 1 0000 in five-bit register) and second is +0 (e.g., 0 0000 in five-bit register).

1's Complement Method

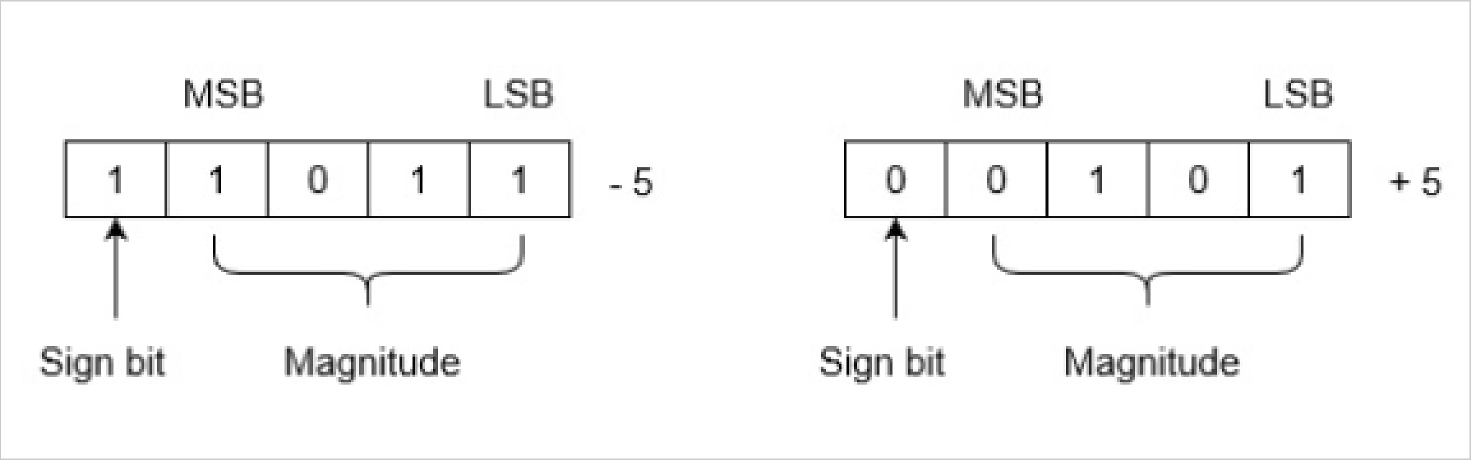
- Positive numbers are represented in the same way as they are represented in sign magnitude method.
- If the number is negative, then it is represented using 1's complement. First represent the number with positive sign and then take 1's complement of that number.
- **Example:** Let we are using 5 bits register. The representation of -5 and +5 will be as follows:



- **Range (for k bits register):** $-(2^{(k-1)}-1)$ to $+(2^{(k-1)}-1)$
- **Drawback:** 0 has two different representation one is -0 (e.g., 1 1111 in five-bit register) and second is +0 (e.g., 0 0000 in five-bit register).

2’s Complement Method

- Positive numbers are represented in the same way as they are represented in sign magnitude method.
- If the number is negative, then it is represented using 2’s complement. First represent the number with positive sign and then take 2’s complement of that number.
- **Example:** Let we are using 5 bits register. The representation of -5 and +5 will be as follows:



- **Range (for k bits register):** $-(2^{(k-1)})$ to $+(2^{(k-1)}-1)$
- **Advantage:** 0 has only one representation for -0 and +0. Zero (0) is considered as always positive (sign bit is 0) in 2’s complement representation