DAY 10 - 111 DAYS VERIFICATION CHALLENGE

Topic: Number System & Digital Codes

Skill: Digital Electronics

DAY 10 CHALLENGE:

1. How many types of number system are there? Describe each in detail.

There are several types of number systems used in mathematics and computing, each with its unique properties and applications. The most common ones include:

1. **Decimal Number System (Base 10)**
2. **Binary Number System (Base 2)**
3. **Octal Number System (Base 8)**
4. **Hexadecimal Number System (Base 16)**

1. Decimal Number System (Base 10)

Description:

- The decimal number system is the most widely used number system in everyday life.
- It consists of 10 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9.
- Each digit's position represents a power of 10. For example, in the number 345, the digit 3 represents 3$10^2$ *(or 300), 4 represents 4*10^1 (or 40), and 5 represents 5*10^0 (or 5).

Example:

- 345 = 3*10^2 + 4*10^1 + 5*10^0

2. Binary Number System (Base 2)

Description:

- The binary number system is used internally by almost all modern computers and computer-based devices.
- It consists of only 2 digits: 0 and 1.

- Each digit's position represents a power of 2. For example, in the binary number 101, the digit 1 represents 1$2^2$ *(or 4), 0 represents 0*2^1 (or 0), and 1 represents 1*2^0 (or 1).

Example:

- $101_2 = 1*2^2 + 0*2^1 + 1*2^0 = 4 + 0 + 1 = 5_{10}$

3. Octal Number System (Base 8)

Description:

- The octal number system is used in some computing applications, particularly in older systems.
- It consists of 8 digits: 0, 1, 2, 3, 4, 5, 6, and 7.
- Each digit's position represents a power of 8. For example, in the octal number 345, the digit 3 represents 3$8^2$ *(or 192), 4 represents 4*8^1 (or 32), and 5 represents 5*8^0 (or 5).

Example:

- $345_8 = 3*8^2 + 4*8^1 + 5*8^0 = 192 + 32 + 5 = 229_{10}$

4. Hexadecimal Number System (Base 16)

Description:

- The hexadecimal number system is widely used in computer science and digital electronics as a more human-friendly representation of binary-coded values.
- It consists of 16 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F (where A=10, B=11, C=12, D=13, E=14, and F=15).
- Each digit's position represents a power of 16. For example, in the hexadecimal number 1A3, the digit 1 represents 1$16^2$ *(or 256), A represents 10*16^1 (or 160), and 3 represents 3*16^0 (or 3).

Example:

- $1A3_{16} = 1*16^2 + 10*16^1 + 3*16^0 = 256 + 160 + 3 = 419_{10}$

2. Explain below codes:

- **BCD Code (Binary-Coded Decimal)**

Description: Binary-Coded Decimal (BCD) is a class of binary encodings of decimal numbers where each digit is represented by its own binary sequence. In

the most common variant, each of the decimal digits (0-9) is represented by a 4-bit binary number.

Details:

- In BCD, each decimal digit is independently converted to its 4-bit binary equivalent.
- This encoding is straightforward and easy to understand but is less efficient in terms of storage compared to pure binary representation.

Example: The decimal number 259 in BCD is represented as follows:

259 = 0010 0101 1001

Advantages:

- Simplifies the design of hardware for decimal arithmetic operations.
- Easier to convert between decimal and binary for human readability.

Disadvantages:

- Less efficient in terms of storage since 4 bits are used for each decimal digit, even though some combinations (like 1010 to 1111) are unused.

➢ Gray Code

Description: Gray code, also known as reflected binary code, is a binary numeral system where two successive values differ in only one bit. It is useful in error correction in digital communications and in minimizing errors in analog to digital conversions.

Details:

- In Gray code, two consecutive numbers differ by only one bit.
- This property helps reduce errors during the transition between numbers because only one bit changes at a time.

Example: To convert binary numbers to Gray code:

1. Start with the most significant bit (MSB) which remains the same.
2. Each subsequent bit in Gray code is found by XORing the current binary bit with the previous binary bit.

For example, converting binary 1011 to Gray code:

- MSB remains the same: 1

- Next bit: 0 XOR 1 = 1
- Next bit: 1 XOR 0 = 1
- Last bit: 1 XOR 1 = 0

So, 1011 (binary) = 1110 (Gray code)

Advantages:

- Reduces the possibility of errors during the transition between two values.
- Particularly useful in analog to digital conversion (ADCs) where the input may change slowly, leading to fewer errors due to the one-bit change property.

Disadvantages:

- More complex to convert between Gray code and binary compared to pure binary systems.
- Not as commonly used for general-purpose computations as binary systems.

3. What are the applications of:

I. Weighted Code:

- Weighted codes are commonly used for representing decimal numbers in binary form. They assign different weights to each bit position, making them suitable for arithmetic operations.
- Applications: Weighted codes find use in digital signal processing, analog-to-digital conversion, and error detection.

II. Non-weighted Code:

- Non-weighted codes do not use positional weights. They are employed in special applications where binary weights are not necessary.
- Examples: Excess-3 code and Gray code.
- Applications: Excess-3 code enhances arithmetic tasks in early computing systems, while Gray code is useful in rotary encoders and error detection.

III. Excess-3 Code:

- Excess-3 (or XS-3) is a non-weighted binary coded decimal (BCD) code.
- It represents decimal digits by adding 3 to their corresponding 4-bit BCD representation.

- Applications: Excess-3 code improves arithmetic operations in early computers and miniaturized devices[12].

IV. BCD Code:

- Binary Coded Decimal (BCD) represents decimal numbers using 4-bit binary codes.
- Applications: BCD is used in digital displays, calculators, and financial systems.

V. Gray Code:

- Gray code is a non-weighted code where adjacent numbers differ by only one bit.
- Applications: Gray code is used in rotary encoders, error detection, and minimizing errors during analog-to-digital conversion.

VI. ASCII Code:

- The American Standard Code for Information Interchange (ASCII) assigns numeric values to characters.
- Applications: ASCII is fundamental for character encoding in computers, communication protocols, and text-based data exchange.

VII. Octal Number System:

- Octal is a base-8 number system.
- Applications: Octal is used in computer programming (especially when dealing with permissions), file permissions, and hardware addressing.

VIII. Hexadecimal Number System:

- Hexadecimal (hex) is a base-16 number system.
- Applications: Hexadecimal is widely used in programming, memory addressing, and representing colors in HTML and CSS.

4. State the disadvantage of an 8-4-2-1 code?

- **Inefficient Use of Bits:**
  - The 8-4-2-1 BCD code uses 4 bits to represent each decimal digit. Since a 4-bit binary number can represent values from 0 to 15, but the BCD code only uses values from 0 to 9, there is a wastage of 6 combinations (1010 to 1111 are unused).

- **More Complex Arithmetic Operations:**
  - Performing arithmetic operations like addition and subtraction is more complex compared to pure binary representations. Special correction rules are needed when the sum of two BCD digits exceeds 9, leading to additional logic circuitry in hardware.

- **Inefficient for Large Numbers:**

  - Representing large numbers in BCD format can be inefficient compared to binary representation. For example, a 4-digit decimal number requires 16 bits in BCD (4 digits * 4 bits) compared to a maximum of 14 bits in pure binary.

- **Increased Storage Requirements:**

  - Due to the inefficiency in bit usage, storing numbers in BCD format requires more memory space compared to binary representation, especially for large data sets.

- **Slower Processing:**
  - Arithmetic operations in BCD can be slower due to the need for additional steps to handle the decimal adjustments, impacting the overall speed of calculations in systems using this code.

- **Limited Use Cases:**
  - BCD is primarily useful in applications requiring precise decimal representation, such as financial and commercial applications. In most other contexts, the advantages of binary representation outweigh the benefits of BCD, limiting its widespread use.

- **Complicated Conversion:**
  - Converting between BCD and binary or other number systems can be more complicated and require additional computational steps, which can be cumbersome in certain applications.

5. What are different ways to represent a negative number?

There are several way to represent negative number but here we discussed main 3 ways

1. **Sign-Magnitude Method:**

   - In this method, we add an extra sign bit to recognize negative and positive numbers.
   - The sign bit is 1 for negative numbers and 0 for positive numbers.

- Range of Numbers: For an n-bit register, the MSB (most significant bit) is the sign bit, and the remaining (n-1) bits represent the magnitude. The negative lowest number that can be stored is -(2^(k-1) - 1), and the positive largest number is (2^(k-1) - 1).
- Note: This representation has an ambiguous representation of zero, as it can be either -0 or +0.

2. **1's Complement Method:**

- The MSB is always the sign bit (1 for negative numbers).
- To represent negative numbers, we take the 1's complement (invert all bits) of the positive number.
- Range of Numbers: Similar to sign-magnitude, the negative lowest number is -(2^(n-1) - 1), and the positive largest number is (2^(n-1) - 1). Again, there's an ambiguous representation of zero.

3. **2's Complement Method:**

- Like the previous methods, the MSB is the sign bit (1 for negative numbers).
- To represent negative numbers, we take the 2's complement (invert all bits and add 1) of the positive number.
- Advantage: There's only one representation of +0 and -0, making it better than the other two methods.
- Range of Numbers: For an n-bit register, the negative lowest number is -(2^(n-1)), and the positive largest number is (2^(n-1) - 1).