NUMBER SYSTEMS

CC1/L
INTRO TO COMPUTING

Learning Objectives:

- By the end of the class, students should be able to explain the fundamental concepts of different number systems, including binary, decimal, octal, and hexadecimal, and recognize their applications in various fields.
- Students should be proficient in performing arithmetic operations (addition, subtraction, multiplication, and division) using binary numbers, and be able to demonstrate this knowledge through practical exercises.
- Should be able to convert numbers seamlessly between different number systems, such as converting a decimal number to binary, an octal number to hexadecimal, or vice versa, while understanding the underlying principles of these conversions.

Introduction

Understanding number systems is essential in various fields, including mathematics, computer science, and digital electronics. This course will provide you with a solid foundation in number systems, particularly focusing on binary, decimal, octal, and hexadecimal systems, as well as teaching you how to convert between them.

Types of Number Systems

A. Binary (Base-2)

Representation: Binary uses two symbols, 0 and 1.

Sample: Binary representation of decimal 13_{10}: 1101_2

B. Decimal (Base-10)

Representation: Decimal uses ten symbols, 0 to 9.

Sample: Decimal representation of binary 101010_2: 42_{10}

C. Octal (Base-8)

Representation: Octal uses eight symbols, 0 to 7.

Sample: Octal representation of decimal 56_{10}: 70_8

D. Hexadecimal (Base-16)

Representation: Hexadecimal uses sixteen symbols, 0-9 and A-F.

Sample: Hexadecimal representation of binary 11011011_2: DB_{16}

Conversions

A. Binary to Decimal Conversion

Binary is a base-2 number system, which means it only uses two digits: 0 and 1. To convert a binary number to decimal (base-10), follow these steps:

1. Start from the rightmost digit (the least significant bit) and assign it the value 1.

2. Move to the left for each subsequent digit, doubling the value assigned in the previous step.

3. Multiply each digit in the binary number by the corresponding value calculated in step 2.

4. Sum up all the products to obtain the decimal equivalent.

Sample: Let's convert the binary number `110010` to decimal:

- 0 * 1 = 0
- 1 * 2 = 2
- 0 * 4 = 0
- 0 * 8 = 0
- 1 * 16 = 16
- 1 * 32 = 32

Now, add up all these values: 0 + 2 + 0 + 0 + 16 + 32 = 50

So, the binary number `110010` is equivalent to the decimal number 50.

B. Decimal to Binary Conversion

To convert a decimal number to binary, you can use a process called successive division by 2. Here are the steps:

1. Start with the decimal number you want to convert.

2. Divide the number by 2 and write down the remainder (0 or 1).

3. Keep dividing the quotient from the previous step by 2 and writing down remainders until the quotient becomes 0.

4. Write down the remainders in reverse order to get the binary equivalent.

Sample: Let's convert the decimal number 27 to binary:

- 27 ÷ 2 = 13 remainder 1
- 13 ÷ 2 = 6 remainder 1
- 6 ÷ 2 = 3 remainder 0
- 3 ÷ 2 = 1 remainder 1
- 1 ÷ 2 = 0 remainder 1

Now, write down the remainders in reverse order: `11011`

So, the decimal number 27 is equivalent to the binary number `11011`.

C. Binary to Octal Conversion

Octal is a base-8 number system, which means it uses eight digits: 0 to 7. To convert a binary number to octal, group the binary digits into sets of three (starting from the right) and replace each set with its octal equivalent.

Sample: Let's convert the binary number `1011011` to octal:

- Group the binary digits into sets of three: `101` and `101` and `1`.
- Replace each group with its octal equivalent: `5` `5` `1`

So, the binary number `1011011` is equivalent to the octal number `551`.

D. Octal to Binary Conversion

To convert an octal number to binary, simply replace each octal digit with its binary equivalent.

Sample: Let's convert the octal number `45` to binary:

- `4` in binary is `100`
- `5` in binary is `101`

Combine these binary representations: `100101`

So, the octal number `45` is equivalent to the binary number `100101`.

E. Binary to Hexadecimal Conversion

Hexadecimal is a base-16 number system that uses the digits 0-9 and the letters A-F to represent values from 0 to 15. To convert a binary number to hexadecimal, group the binary digits into sets of four (starting from the right) and replace each set with its hexadecimal equivalent.

Sample: Let's convert the binary number `101110110` to hexadecimal:

- Group the binary digits into sets of four: `1011` and `1011` and `0`.
- Replace each group with its hexadecimal equivalent: `B` `B` `0`

So, the binary number `101110110` is equivalent to the hexadecimal number `BB0`.

F. Hexadecimal to Binary Conversion

To convert a hexadecimal number to binary, replace each hexadecimal digit with its binary equivalent.

Sample: Let's convert the hexadecimal number `2A` to binary:

- `2` in binary is `0010`
- `A` in binary is `1010`

Combine these binary representations: `00101010`

So, the hexadecimal number `2A` is equivalent to the binary number `00101010`.

These conversion methods allow you to translate numbers between different number systems, which is particularly useful in various applications, including computer programming and digital electronics.

G. Decimal to Octal Conversion

To convert a decimal number to octal, you can use a process similar to converting to binary, but this time using base-8. Here are the steps:

1. Start with the decimal number you want to convert.

2. Divide the number by 8 and write down the remainder (0 to 7).

3. Keep dividing the quotient from the previous step by 8 and writing down remainders until the quotient becomes 0.

4. Write down the remainders in reverse order to get the octal equivalent.

Sample: Let's convert the decimal number 56 to octal:

- 56 ÷ 8 = 7 remainder 0
- 7 ÷ 8 = 0 remainder 7

Now, write down the remainders in reverse order: `70`

So, the decimal number 56 is equivalent to the octal number `70`.

H. Octal to Decimal Conversion

Converting from octal to decimal is straightforward. Each octal digit represents a power of 8, with the rightmost digit being 8^0, the next digit being 8^1, and so on. To convert an octal number to decimal, simply multiply each digit by the corresponding power of 8 and add the results.

Sample: Let's convert the octal number `133` to decimal:

- 3 * 8^0 = 3 * 1 = 3
- 3 * 8^1 = 3 * 8 = 24
- 1 * 8^2 = 1 * 64 = 64

Now, add up these values: 3 + 24 + 64 = 91

So, the octal number `133` is equivalent to the decimal number 91.

I. Decimal to Hexadecimal Conversion

To convert a decimal number to hexadecimal, you can use a process similar to converting to binary or octal, but this time using base-16. Here are the steps:

1. Start with the decimal number you want to convert.

2. Divide the number by 16 and write down the remainder (0 to 15).

3. Keep dividing the quotient from the previous step by 16 and writing down remainders until the quotient becomes 0.

4. Write down the remainders in reverse order. For values 10 to 15, use the corresponding letters A to F.

Sample: Let's convert the decimal number 273 to hexadecimal:

- 273 ÷ 16 = 17 remainder 1
- 17 ÷ 16 = 1 remainder 1
- 1 ÷ 16 = 0 remainder 1

Now, write down the remainders in reverse order: `111`

So, the decimal number 273 is equivalent to the hexadecimal number `111`.

J. Hexadecimal to Decimal Conversion

Converting from hexadecimal to decimal is straightforward. Each hexadecimal digit represents a power of 16, with the rightmost digit being 16^0, the next digit being 16^1, and so on. To convert a hexadecimal number to decimal, simply multiply each digit by the corresponding power of 16 and add the results.

Sample: Let's convert the hexadecimal number `1B6` to decimal:

- 6 * 16^0 = 6 * 1 = 6
- B (which is 11 in decimal) * 16^1 = 11 * 16 = 176
- 1 * 16^2 = 1 * 256 = 256

Now, add up these values: 6 + 176 + 256 = 438

So, the hexadecimal number `1B6` is equivalent to the decimal number 438.

Introduction to Binary Arithmetic

Binary arithmetic is the foundation of all digital computing. It involves performing arithmetic operations (addition, subtraction, multiplication, and division) using the binary number system, which is composed of only two digits: 0 and 1. In this handout, we will explore the basics of binary arithmetic and provide samples for each operation.

Binary Digits

0: Represents no quantity.
1: Represents the smallest quantity.

Place Value in Binary

Similar to the decimal system, binary numbers have place values that increase by a power of 2 from right to left:

... (8) (4) (2) (1)

For example, in the binary number **1101**:
- The rightmost digit (1) represents 2^0 = 1.
- The next digit to the left (0) represents 2^1 = 2.
- The next digit to the left (1) represents 2^2 = 4.
- The leftmost digit (1) represents 2^3 = 8.

Binary Addition

To add two binary numbers, follow these rules:

- **0 + 0 = 0**
- **0 + 1 = 1**
- **1 + 0 = 1**
- **1 + 1 = 10** (carry over 1)

Example:

```
  1101
+ 1010
-------------
```

1. Start from the rightmost digits and add them:
   - 1 + 0 = 1
   - 0 + 1 = 1
   - 1 + 0 = 1
   - 1 + 1 = 10 (Write down 0, carry over 1)

2. Move to the next digits:
   - Carry over the 1: 1 + 1 (from the previous step) = 10 (Write down 0, carry over 1)

3. Continue until all digits are added, including any carried-over values.

The result is 11011

Binary Subtraction

Subtracting binary numbers is similar to decimal subtraction, but it may involve borrowing.

0 - 0 = 0**
1 - 0 = 1**
1 - 1 = 0**
0 – 1 : Borrow 1 from the next higher place value, making it **10 - 1 = 1**, and subtract as usual.

Example:

```
  1101
- 1010
-------------
```

1. Start from the rightmost digits and subtract them:
   1 - 0 = 1
   0 - 1: Borrow 1, making it 10 - 1 = 1
   1 - 0 = 1
   1 - 1 = 0

The result is 0001

Binary Multiplication

To multiply two binary numbers, perform a series of additions based on the place values of the digits.

Example:

```
   1101
× 1010
----------------
```

1. Start by multiplying the rightmost digit of the bottom number (0) by the top number:
   0 × 1101 = 0000 (Write this under the line).

2. Move to the next digit of the bottom number (1) and shift the top number one place to the left:
   1 × 1101 = 1101 (Write this with one position shifted to the left).

3. Continue this process for each digit in the bottom number and add the results:
   1 × 1101 = 1101 (Shifted 2 positions to the left)
   0 × 1101 = 0000 (Shifted 3 positions to the left)

4. Add the results:

```
    11010
+   1101
--------------------
  1110010
```

The result is 1110010.

Binary Division

Binary division is performed similarly to decimal division, using long division.

Example:

```
    101101
÷   11
-----------------
```

1. Start by dividing the leftmost digit of the dividend by the divisor:
   1 ÷ 1 = 1

2. Multiply the divisor by the result from step 1 (1) and subtract it from the dividend:

```
   101101
 -   11
 ----------------
     11
```

3. Bring down the next digit from the dividend:

```
101101
 -   11
 -----------------
     110
```

4. Repeat steps 1 to 3 until all digits in the dividend have been processed.

The result is 1001 with a remainder of 10.

Note:
You can always convert the resulting binary values to its decimal values then perform the actual operation to check if the results are correct.

------END-----