

# Number Bases and Binary Operations

---

**Author: Shekh Mostofa Abedin**

## Understanding Number Bases

### 1. Decimal (Base 10)

- **Digits:** 0-9.
- **Usage:** The standard system for counting and arithmetic.
- **Example:** The number 345 in decimal is written as  $345_{10}$ .

### 2. Binary (Base 2)

- **Digits:** 0, 1.
- **Usage:** The fundamental language of computers.
- **Example:** The binary number  $1011_2$  equals  $11_{10}$  in decimal.

### 3. Hexadecimal (Base 16)

- **Digits:** 0-9, A-F (where A=10, B=11, ..., F=15).
- **Usage:** Compact representation of binary numbers, commonly used in computing.
- **Example:** The hexadecimal number  $2F3_{16}$  equals  $755_{10}$  in decimal.

## Use of Letters in Number Bases

- **Base 2 to Base 10:** Only digits 0-9 are used. There are no letters, and after 9, the next number is  $10_{10}$ .
- **Base 11 and Above:** Letters are introduced to represent values greater than 9. The use of letters continues until the base limit is reached. After the highest letter (which corresponds to the base minus 1), the numbering transitions to  $10_{10}$ ,  $11_{10}$ , and so on.
  - **Examples:**
    - **Base 11:** Uses digits 0-9 and the letter A (where A=10). After A, the next number is  $10_{10}$ .
    - **Base 12:** Uses digits 0-9 and the letters A (10) and B (11). After B, the next number is  $10_{10}$ .
    - **Base 16 (Hexadecimal):** Uses digits 0-9 and the letters A (10), B (11), C (12), D (13), E (14), and F (15). After F, the next number is  $10_{10}$ .
    - **Base 17:** Uses digits 0-9 and the letters A (10), B (11), C (12), D (13), E (14), F (15), and G (16). After G, the next number is  $10_{10}$ .
- **General Rule:** In bases greater than 10, after the digits 0-9 are exhausted, letters are used starting with A=10. The sequence continues with these letters until reaching the base limit. After the highest letter, the numbering restarts from  $10_{10}$  (which represents the base value), followed by  $11_{10}$ ,  $12_{10}$ , etc.
- **Base 8 (Octal):** Uses only digits 0-7. No letters are used, and after 7, the next number is  $10_{10}$ .

---

## Converting Between Number Bases

### 1. Binary to Decimal

- **Steps:**
  1. Multiply each binary digit by 2 raised to the power of its position (starting from 0 on the right).
  2. Sum all the results.
- **Example:** Convert  $1101_2$  to decimal.
  - Calculation:  $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1 = 13_{10}$

### 2. Decimal to Binary

- **Steps:**
  1. Divide the decimal number by 2.
  2. Record the remainder.
  3. Update the quotient and repeat until the quotient is 0.
  4. The binary number is the remainders read in reverse order.
- **Example:** Convert  $23_{10}$  to binary.

- Division steps:
  - $23 \div 2 = 11$ , remainder = 1
  - $11 \div 2 = 5$ , remainder = 1
  - $5 \div 2 = 2$ , remainder = 1
  - $2 \div 2 = 1$ , remainder = 0
  - $1 \div 2 = 0$ , remainder = 1
- Result:  $10111_2$

3. Hexadecimal to Decimal

- **Steps:**
  1. Multiply each hex digit by 16 raised to the power of its position (starting from 0 on the right).
  2. Sum all the results.
- **Example:** Convert  $2F3_{16}$  to decimal.
  - Calculation:  $2 \times 16^2 + F \times 16^1 + 3 \times 16^0 = 512 + 240 + 3 = 755_{10}$

4. Decimal to Hexadecimal

- **Steps:**
  1. Divide the decimal number by 16.
  2. Record the remainder.
  3. Update the quotient and repeat until the quotient is 0.
  4. The hexadecimal number is the remainders read in reverse order.
- **Example:** Convert  $438_{10}$  to hexadecimal.
  - Division steps:
    - $438 \div 16 = 27$ , remainder = 6
    - $27 \div 16 = 1$ , remainder = 11 (B)
    - $1 \div 16 = 0$ , remainder = 1
  - Result:  $1B6_{16}$

5. Binary to Hexadecimal

- **Steps:**
  1. Group binary digits into sets of four (starting from the right).
  2. Convert each group to its hexadecimal equivalent.

• **Conversion Table:**

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

- **Example:** Convert  $101110111101_2$  to hexadecimal.
  - Grouping:  $1011\ 1011\ 1101$
  - Conversion:  $BBD_{16}$

6. Hexadecimal to Binary

- **Steps:**
  1. Convert each hexadecimal digit to its 4-bit binary equivalent.
- **Example:** Convert  $A4F_{16}$  to binary.
  - Conversion:  $A=1010$ ,  $4=0100$ ,  $F=1111$
  - Result:  $101001001111_2$

7. Decimal to Other Bases

- **Steps:**
  1. Divide the decimal number by the target base.
  2. Record the remainder.
  3. Update the quotient and repeat until the quotient is 0.
  4. The number in the target base is the remainders read in reverse order.
- **Example:** Convert  $345_{10}$  to base 5.

- Division steps:
  - $345 \div 5 = 69$ , remainder = 0
  - $69 \div 5 = 13$ , remainder = 4
  - $13 \div 5 = 2$ , remainder = 3
  - $2 \div 5 = 0$ , remainder = 2
- Result:  $2340_5$

## 8. Any Base to Decimal

- **Steps:**
  1. Multiply each digit by its base power (starting from 0 on the right).
  2. Sum the results.
- **Example:** Convert  $2340_5$  to decimal.
  - Calculation:  $2 \times 5^3 + 3 \times 5^2 + 4 \times 5^1 + 0 \times 5^0 = 250 + 75 + 20 + 0 = 345_{10}$

---

## Bit Shifting

### 1. Left Shift (<<)

- **Operation:** Shifts bits left, multiplying by 2 for each shift.
- **Example:**  $1011 \ll 2$  shifts  $1011$  to  $101100$ .
  - $1011$  in decimal = 11
  - $101100$  in decimal = 44

### 2. Right Shift (>>)

- **Operation:** Shifts bits right, dividing by 2 for each shift.
- **Example:**  $1011 \gg 2$  shifts  $1011$  to  $10$ .
  - $1011$  in decimal = 11
  - $10$  in decimal = 2

---

## Binary Arithmetic

### 1. Binary Addition

#### Basic Rules:

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 10$  (0 with a carry of 1)

#### Adding Binary Numbers of Different Sizes:

- **Align the numbers by their rightmost digits.**
- **If the numbers are of different lengths, prepend zeros to the smaller number** to make both numbers the same size.
- **Example:** Add  $1011$  and  $110$ :
  - Align:

```
  1011
+ 0110
-----
 10001
```

- Result:  $10001_2$

### 2. Binary Subtraction

#### Basic Rules:

- $0 - 0 = 0$
- $1 - 0 = 1$
- $1 - 1 = 0$
- $0 - 1 = 1$  (borrow 1 from the next higher bit)

### Subtracting Binary Numbers of Different Sizes:

- **Align the numbers by their rightmost digits.**
- **If the numbers are of different lengths, prepend zeros to the smaller number** to make both numbers the same size.
- **Example:** Subtract  $110$  from  $1011$ :
  - Align:

```
  1011
- 0110
-----
  0101
```

- Result:  $0101_2$  or simply  $101_2$

### Prefixes for Number Bases

- **Hexadecimal:**  $0x$  or  $0X$  (e.g.,  $0x1A3F$ )
- **Binary:**  $0b$  or  $0B$  (e.g.,  $0b1101$ )
- **Octal:**  $0$  (leading zero) (e.g.,  $0754$ )
- **Decimal:** No prefix needed (e.g.,  $1234$ )

---

### Summary of Conversion Steps

- **To Decimal:** Multiply each digit by its base power, sum the results.
- **From Decimal:** Divide by the target base, record remainders, read them in reverse order.
- **Binary ↔ Hex/Oct:** Group binary digits (3 for octal, 4 for hex), convert each group.
- **Any Base ↔ Any Base:** Convert to decimal first, then to the target base.