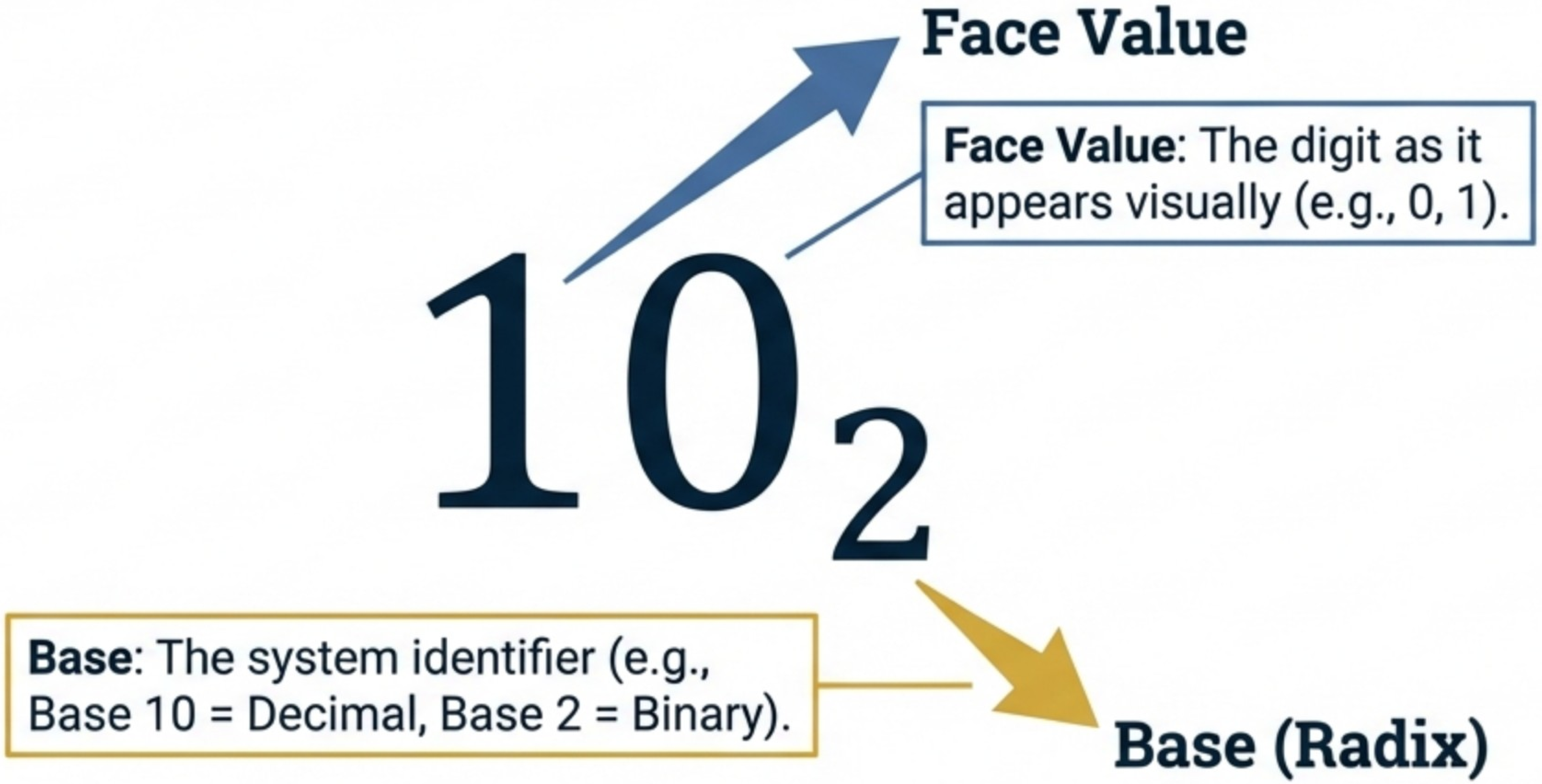
Mastering Number Systems: Conversions & Arithmetic

From Binary to Hexadecimal | Fractional Logic | Base Arithmetic & Complements

A comprehensive guide to understanding how computers process data, covering integer conversions, fractional precision, and the logic of negative numbers.

The Anatomy of a Number: Face vs. Base



$$\text{Actual Value} = \text{Face Value} \times \text{Base}^{\text{Position}}$$

Example: In 10 (Base 2), the ‘1’ is not just ‘one’; it represents $1 \times 2^1 = 2$.

The Valid Digit Rule

Rule: In any Base N, the valid digits range strictly from 0 to N-1.

Base 10 (Decimal)	Base 7	Base 2 (Binary)
0, 1, 2, 3, 4, 5, 6, 7, 8, 9	0, 1, 2, 3, 4, 5, 6	0, 1

Case Study

$\cancel{288_8}$

Impossible. The digit '8' cannot exist in Base 8. The maximum valid digit is 7.

Shortcut: Binary to Base 4

Logic: 4 = 2^2 → Group bits in pairs of 2.

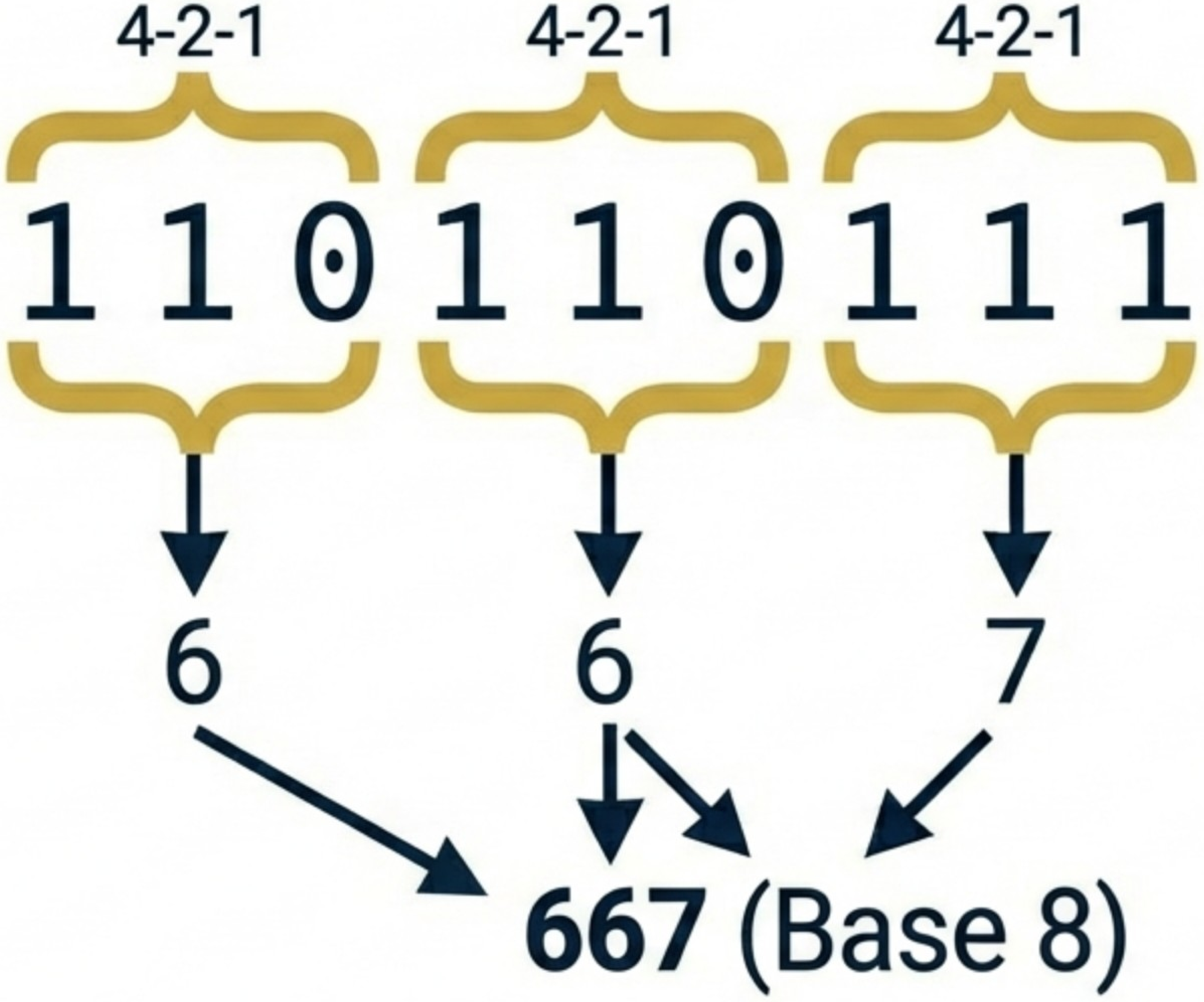
1. Start from Right (LSB).
2. Group bits into pairs.
3. Convert each pair.
3. Convert each pair.

$$\underbrace{11}_{(3)}\ \underbrace{10}_{(2)} \rightarrow 32 \text{ (Base 4)}$$

00 = 0	
01 = 1	
10 = 2	
11 = 3	

Shortcut: Binary to Octal (Base 8)

Logic: 8 = 2^3 → Group bits in sets of 3.



Apply the 4-2-1 value pattern to each group.

Shortcut: Binary to Hexadecimal (Base 16)

Logic: $16 = 2^4 \rightarrow$ Group bits in sets of 4 (Nibbles).

8 - 4 - 2 - 1

1011

$8 + 0 + 2 + 1 = 11 \longrightarrow$ B $\longrightarrow$ B (Base 16)

Alpha-Numeric Key
10 = A
11 = B
12 = C
13 = D
14 = E
15 = F

Reverse Shortcut: Hex/Octal to Binary

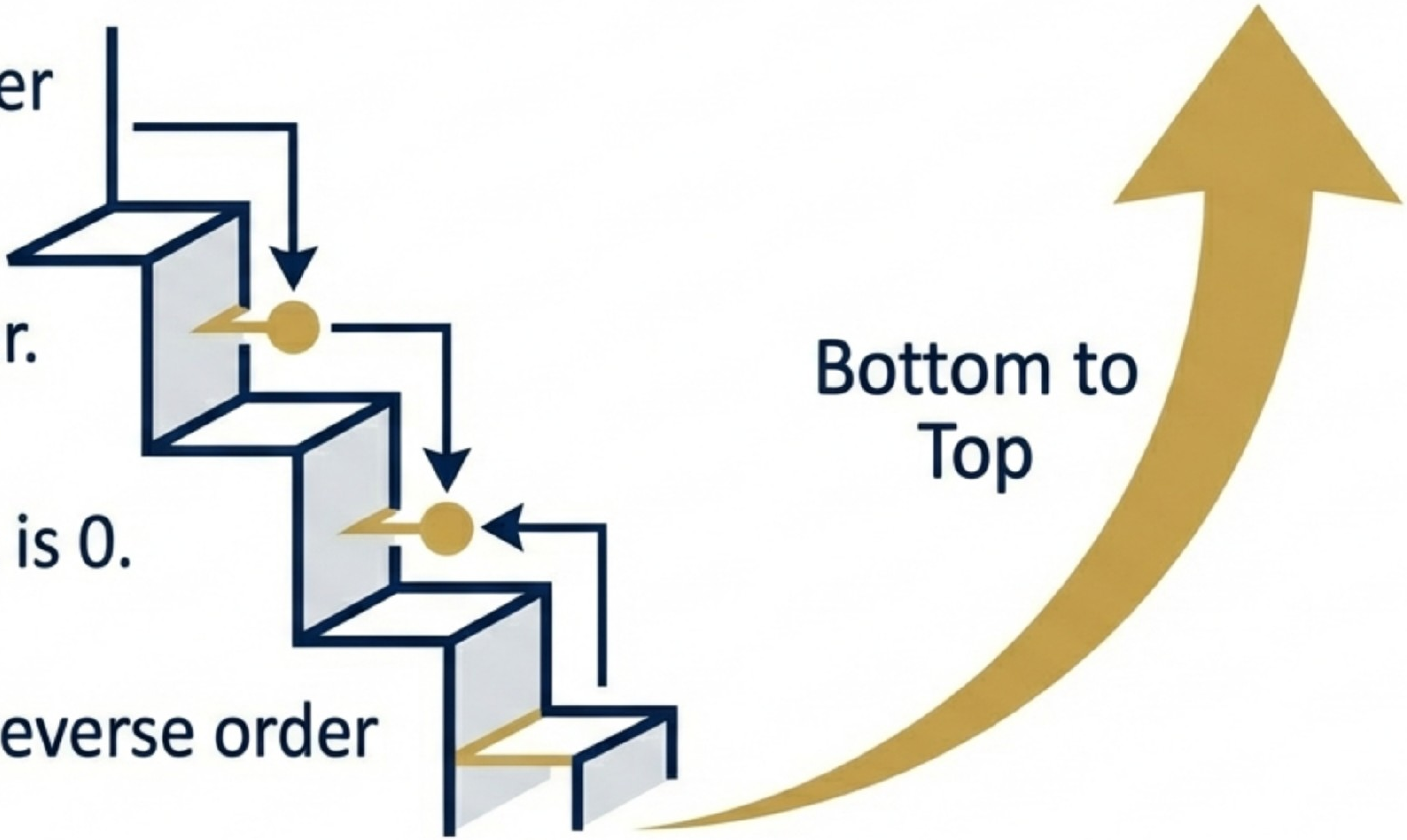
Concept: Expand the “Big Base” digits back into their **bit components.**

From Octal: Write each digit as **3 bits.**
From Hex: Write **each** digit as **4 bits.**

Decimal to Any Base: The LCM Method

Rule: Big Base (10) → Small Base

1. Divide Decimal number by the Target Base.
2. Record the Remainder.
3. Repeat until quotient is 0.
4. Write remainders in reverse order (Bottom to Top).



Example: Converting 57 (Decimal) to Binary

$57 \div 2 = 28$	1
$28 \div 2 = 14$	0
$14 \div 2 = 7$	0
$7 \div 2 = 3$	1
$3 \div 2 = 1$	1
$1 \div 2 = 0$	1

Bottom to Top

Result: 111001 (Base 2)

Any Base to Decimal: Multiplication Method

Rule: Small Base → Big Base (10)

1. Assign position indices (Right to Left).
2. Multiply digit by Base^Position.
3. Sum the results.

Sum of Position Weights

$$\begin{array}{ccc} 2 & 1 & 0 \\ 1 & 0 & 1 \end{array}$$

$$(1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$$

$$4 + 0 + 1 \rightarrow 5 \text{ (Base 10)}$$

Example: 145 (Base 8) → Decimal

Digit 5 (Position 0):	5×8^0	$= 5 \times 1$	$= 5$
Digit 4 (Position 1):	4×8^1	$= 4 \times 8$	$= 32$
Digit 1 (Position 2):	1×8^2	$= 1 \times 64$	$= 64$

Sum: $64 + 32 + 5 = 101$

145 (Base 8) = 101 (Base 10)

Binary Fractions: The Point System

0.101 (Base 2) → 0.5 + 0 + 0.125 = 0.625 (Base 10)

Decimal Fraction to Binary

Rule: Multiply the fraction by the Target Base (2).

$$0.125 \times 2 = \boxed{0}.25$$

$$0.25 \times 2 = \boxed{0}.50$$

$$0.50 \times 2 = \boxed{1}.00$$

Read Top to Bottom.

Stop when fraction reaches 0.

0.001 (Base 2)

Binary Addition Logic

0 + 0 =	0	
0 + 1 =	1	
1 + 1 =	10	Write 0, Carry 1
1 + 1 + 1 =	11	Write 1, Carry 1

$$
\begin{array}{r}
{}^{1}10 \\
10 \\
+\ 110 \\
\hline
1000
\end{array}
\quad \text{(Decimal 8)}
$$

Hexadecimal Addition & Overflow

⚠ Overflow Rule: If Sum ≥ 16, Subtract 16 and Carry 1.

```
     1
     A  2  (Base 16)
  +  4  F  (Base 16)
---------
     F  1  (Base 16)
       F1  (Base 16)
```

Column 1:

2 + F(15) = 17.

17 - 16 = 1. Write 1, Carry 1.

Column 2:

A(10) + 4 + 1(carry) = 15.

15 = F. Write F.

Octal Addition Rules

Magic Number is 8. If Sum ≥ 8: Subtract 8, Write Result, Write Result, Carry 1.

$$\begin{array}{r} 6 \text{ (Base 8)} \\ +\ 2 \text{ (Base 8)} \\ \hline \end{array}$$

$6 + 2 = 8.$

8 is not a valid digit.

8 is not a valid digit.

Action: 8 - 8 = 0. Carry 1.

1

$\boxed{1\ 0}$ (Base 8)

The 1’s Complement

The digital method of inversion.

Flip every bit.

Change 1 → 0

Change 0 → 1

Original:	1	0	1	1	0
	↓	↓	↓	↓	↓
1’s Comp:	0	1	0	0	1

The 2's Complement

Standard for computer arithmetic negative numbers.

2's Complement = 1's Complement + 1

1. Invert all bits (Find 1's Comp).
2. Add binary 1 to the result.

Example: Writing Negative 32 (-32)

Write Positive (+32)

0100000

Invert (1's Comp)

0100000

1011111

Add 1

$$\begin{array}{r} 1011111 \\ +\quad\quad 1 \\ \hline 1100000 \end{array}$$

1100000 represents -32

Summary of Rules

Conversions

- Decimal to Base N = Division (LCM).
- Base N to Decimal = Multiplication (Power).
- Fractions = Multiply by Base.

Grouping Shortcuts

- Base 4 (2 bits).
- Base 8 (3 bits).
- Base 16 (4 bits).

Arithmetic

- Binary: 1+1=10.
- Hex/Octal: Subtract Base on Overflow.

Negatives

- Use 2’s Complement (Flip bits + 1).