Introduction to Numbering Systems

- What is the difference between a digit and a number?

Digit Examples	Number Examples
3	25
0	328
9	10458
7	5000
1	5

- Difference between a digit and a number is similar to the difference between a letter (a character) and a word.

“a” , “F” **“Book” , “Door”**

Introduction to Numbering Systems

- Please, read the following Number:

3 2 1

- Your answer:

three hundreds and twenty one

- What makes 3 has a value of hundreds while 2 has a value of tens ???

$$3\ 2\ 1 = 3 \times 100 + 2 \times 10 + 1$$

$$3\ 2\ 1 = 3 \times 10^2 + 2 \times 10^1 + 1 \times 10^0$$

- This numbering is called a **Decimal System**

Introduction to Numbering Systems

- Least Significant Digit (LSD) is the right most digit has the lowest value

- Most Significant Digit (MSD) is the left most digit has the highest value

- The most significant symbol can not be zero.

- Example:
  Show how the value of the number $(9375)_{10}$ is estimated

Introduction to Numbering Systems

❖ Example:

Show how the value of the number $(9375)_{10}$ is estimated

position	3	2	1	0	
Weight	10^3	10^2	10^1	10^0	
Digit	9	3	7	5	
Value	9x1000	3x100	7x10	5x1	9375

5 is the least significant digit (LSD)

9 is the most significant digit (MSD)

Numbering System Definition

Numbering System Definitions

- A **numbering system** is a way of representing numbers.
- Numbers are usually expressed in positional notation
- A number is represented as a string of digits, e.g., a number N with n digits represented by sequence.

$$d_{n-1}, \ldots, d_3 \; d_2 \; d_1 \; d_0$$

- Example: 7 0 5 4 3

Types of Numbering Systems

- Any **numbering system** is defined by:

1. A base or a radix N

2. N digits : 0, 1, 2, 3, 4, 5, …, N-1

Note that the number of digits is equal to the system base

System	Base or Radix	Symbols or digits	Used by humans?	Used in computers?
Binary	2	0, 1	No	Yes
	4	0, 1, 2, 3	No	No
	5	0, 1, 2, 3, 4	No	No
Octal	8	0, 1, … 7	No	No
Decimal	10	0, 1, … 9	Yes	No
Hexa-decimal	16	0, 1, … 9, A, B, … F	No	No

Types of Numbering Systems

System	Base or Radix	Valid Examples	Invalid Examples
Binary	2	11101 - 0000 101 - 10	11012 – 1110034 00021 – 943A
	4	23101- 10101 11023 - 3330	120034 – 01514 B34 - 12016
	5	40001 – 1234 100101 - 444	F11 – 500 10456 - 7
Octal	8	7512 – 0014 101112	D34 – 800 9001
Decimal	10	0142 – 111010 9871 - 1000	1E56 – 0101A
Hexa-decimal	16	1250 – 11101 A22 – 3B45	G234 – 34H2

Conversion among Different Numbering Systems

Conversion among different Numbering Systems

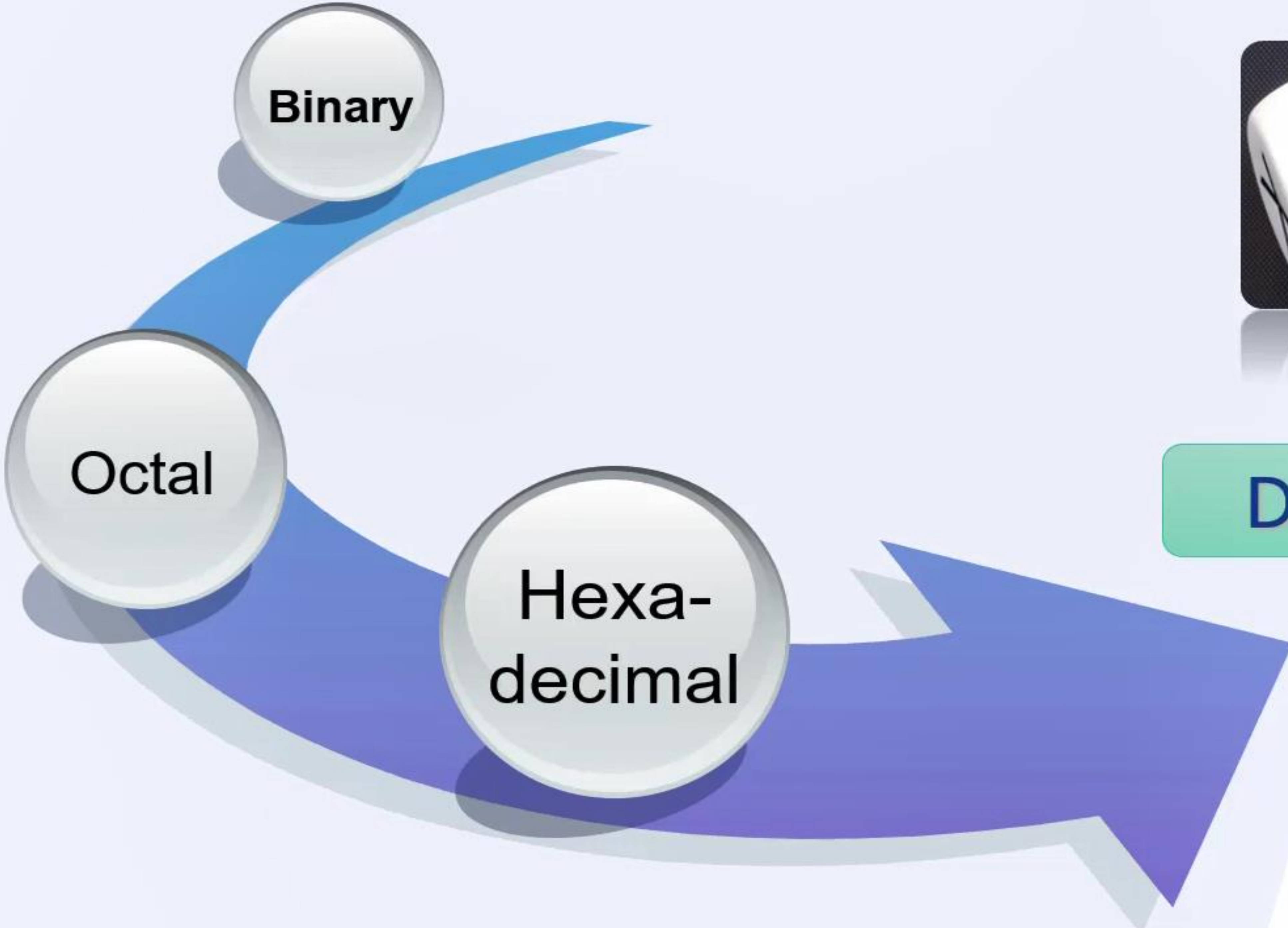
❖ Quick Example:

$$(25)_{10} = (11001)_2 = (31)_8 = (19)_{16}$$

Base

Conversion among different Numbering Systems

1. Converting from any base to decimal
2. Converting from decimal to any base
3. Converting from any base to any base
4. A special conversion case (Shortcut method to/from binary)



Decimal

Conversion from Any System to Decimal

- The general form of any number in any numbering system:

$$\ldots \quad d_3 \; d_2 \; d_1 \; d_0 \,.\, d_{-1} \; d_{-2} \; d_{-3} \quad \ldots$$

Where d_i is the coefficient

- Example: 5 4 3 . 7 5

- A number in base *r* contains *r* digits 0,1,2 ...., r-1 is expressed with a power series in r

$$d_n\, r^n + d_{n-1}\, r^{n-1} + \ldots + d_2\, r^2 + d_1\, r^1 + d_0\, r^0 + d_{-1}\, r^{-1} + d_{-2}\, r^{-2} + \ldots$$

Conversion from Decimal to Decimal

$527 = 5 \times 10^2 + 2 \times 10^1 + 7 \times 10^0$

Place	10^2	10^1	10^0
Value	100	10	1
Number	5	2	7
Evaluate	5 x 100	2 x 10	7 x1
Sum	500	20	7

Conversion from Decimal to Decimal

- Base (also called radix) = 10
  - 10 digits { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }
- Digit Position
  - Integer & fraction
- Digit Weight
  - Weight = $(Base)^{Position}$
- Magnitude
  - Sum of "*Digit* x *Weight*"
- Formal Notation

2	1	0		-1	-2
5	1	2	.	7	4
100	10	1		0.1	0.01
500	10	2		0.7	0.04

$$d_2 * B^2 + d_1 * B^1 + d_0 * B^0 + d_{-1} * B^{-1} + d_{-2} * B^{-2}$$

$$(512.74)_{10}$$

Conversion from Binary to Decimal

- Base = 2
  - 2 digits { 0, 1 }, called **binary digits or "bits"**
- Weights
  - Weight = $(Base)^{Position}$
- Magnitude
  - Sum of "*Bit* x *Weight*"
- Formal Notation
- Groups of bits
  - 4 bits = *Nibble*
  - 8 bits = *Byte*

4	2	1		1/2	1/4
1	0	1	.	0	1
2	1	0		-1	-2

$$1*2^2+0*2^1+1*2^0+0*2^{-1}+1*2^{-2}$$

$$=(5.25)_{10}$$

$$(101.01)_2$$

1 0 1 1

1 1 0 0 0 1 0 1

Conversion from Octal to Decimal

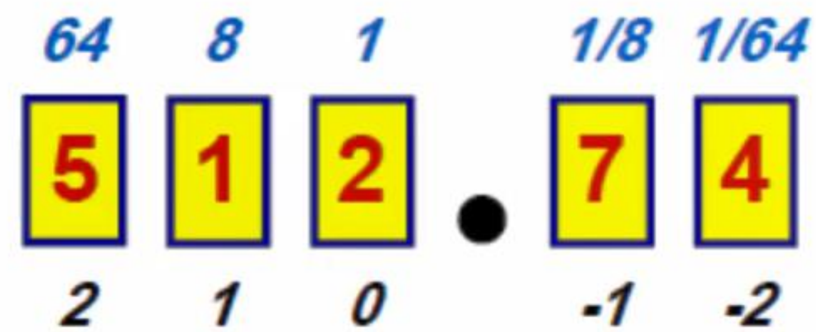
$624_8 = 404_{10}$

64's place 8's place 1's place

Place	8^2	8^1	8^0
Value	64	8	1
Number	6	2	4
Evaluate	6 x 64	2 x 8	4 x 1
Sum for Base 10	384	16	4

Conversion from Octal to Decimal

- Base = 8
  - 8 digits { 0, 1, 2, 3, 4, 5, 6, 7 }
- Weights
  - Weight = $(Base)^{Position}$
- Magnitude
  - Sum of "*Digit* x *Weight*"
- Formal Notation



$$5*8^2+1*8^1+2*8^0+7*8^{-1}+4*8^{-2}$$

$$=(330.9375)_{10}$$

$$(512.74)_8$$

Conversion from Hexadecimal to Decimal

$6{,}704_{16} = 26{,}372_{10}$

4,096's place 256's place 16's place 1's place

Place	16^3	16^2	16^1	16^0
Value	4,096	256	16	1
Number	6	7	0	4
Evaluate	6 x 4,096	7 x 256	0 x 16	4 x 1
Sum for Base 10	24,576	1,792	0	4

Conversion from Hexadecimal to Decimal

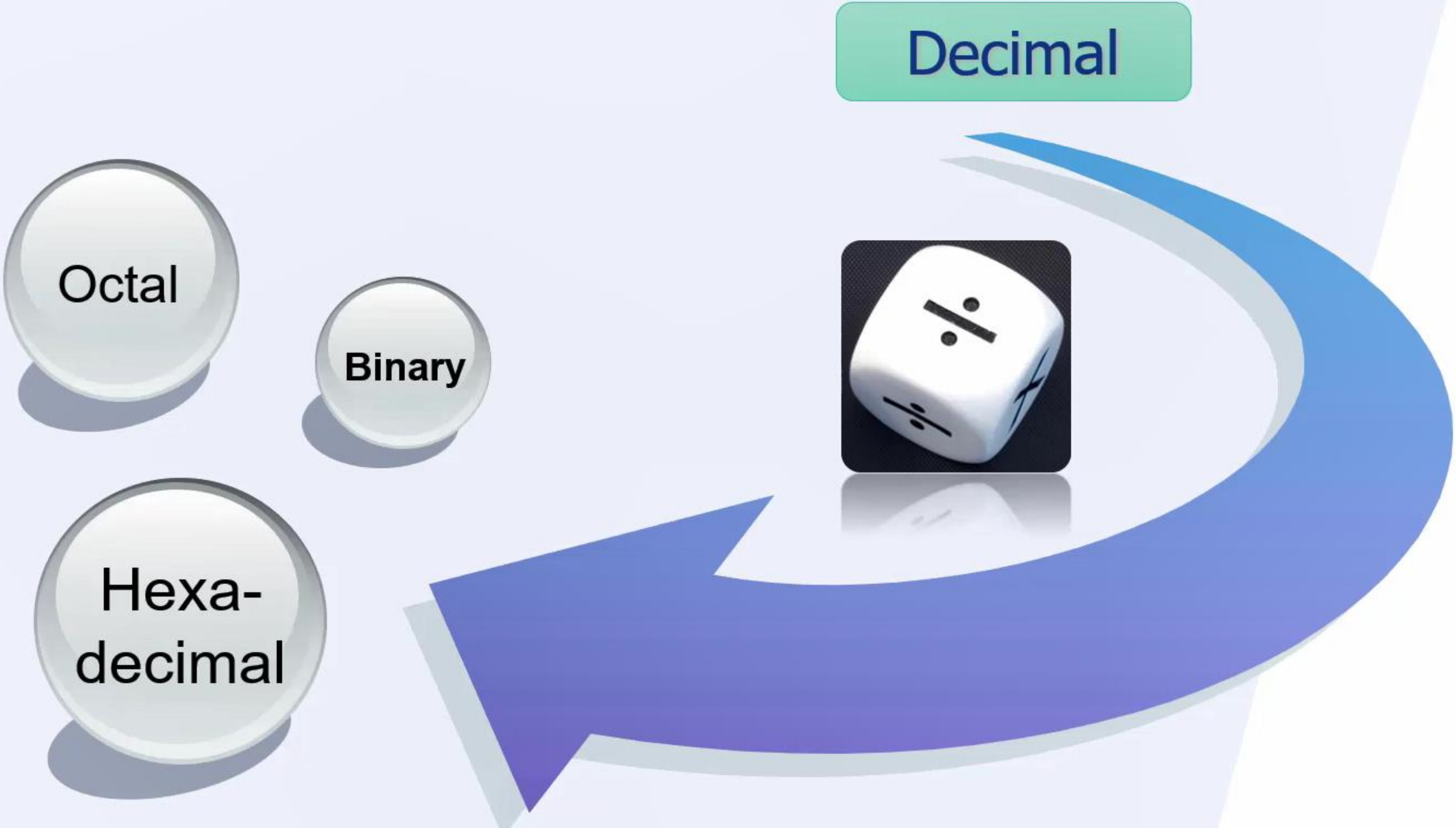
- Base = 16
  - 16 digits { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F }
- Weights
  - Weight = $(Base)^{Position}$
- Magnitude
  - Sum of "*Digit* x *Weight*"
- Formal Notation

256	16	1		1/16	1/256
1	E	5	.	7	A
2	1	0		-1	-2

$$1*16^2+14*16^1+5*16^0+7*16^1+10*16^2$$

$$=(485.4765625)_{10}$$

$$(1E5.7A)_{16}$$


Decimal
Octal
Binary
Hexa-
decimal

Conversion from Decimal to Binary

- Divide the number by the 'Base' (=2)
- Take the remainder (either 0 or 1) as a coefficient
- Take the quotient and repeat the division

Example: $(13)_{10}$

	Quotient	Remainder	Coefficient
13 / 2 =	6	1	$a_0 = 1$
6 / 2 =	3	0	$a_1 = 0$
3 / 2 =	1	1	$a_2 = 1$
1 / 2 =	0	1	$a_3 = 1$

Answer: $(13)_{10} = (a_3\, a_2\, a_1\, a_0)_2 = (1101)_2$

MSB (a_3) LSB (a_0)

Conversion from Decimal to Binary

$(125)_{10} = (?)_2$

2	125	
2	62	1
2	31	0
2	15	1
2	7	1
2	3	1
2	1	1
	0	1

$(125)_{10} = (1111101)_2$

Conversion from Decimal (Fraction) to Binary

- Multiply the number by the 'Base' (=2)
- Take the integer (either 0 or 1) as a coefficient
- Take the resultant fraction and repeat the division

Example: $(0.625)_{10}$

	Integer		Fraction	Coefficient
0.625 * 2 =	1	.	25	$a_{-1} = 1$
0.25 * 2 =	0	.	5	$a_{-2} = 0$
0.5 * 2 =	1	.	0	$a_{-3} = 1$

Answer: $(0.625)_{10} = (0.a_{-1}\, a_{-2}\, a_{-3})_2 = (0.101)_2$

MSB LSB

Conversion from Decimal to Binary

Convert 3.6 to binary with accuracy 8 bit

2*.6 = 1 + .2

2*.2 = 0 + .4

2*.4 = 0 + .8

2*.8 = 1 + .6

2*.6 = 1 + .2

2*.2 = 0 + .4

2*.4 = 0 + .8

2*.8 = 1 + .6

Answer: 11.10011001...

MSB

Conversion from Decimal to Octal

Example: $(175)_{10}$

	Quotient	Remainder	Coefficient
175 / 8 =	21	7	$a_0 = 7$
21 / 8 =	2	5	$a_1 = 5$
2 / 8 =	0	2	$a_2 = 2$

Answer: $(175)_{10} = (a_2\, a_1\, a_0)_8 = (257)_8$

Example: $(0.3125)_{10}$

	Integer		Fraction	Coefficient
0.3125 * 8 =	2	.	5	$a_{-1} = 2$
0.5 * 8 =	4	.	0	$a_{-2} = 4$

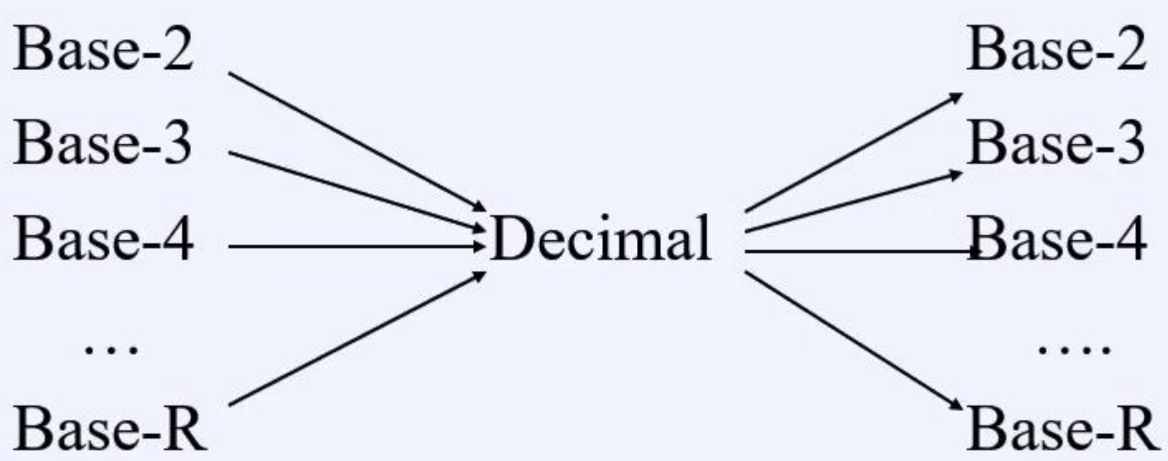
Answer: $(0.3125)_{10} = (0.a_{-1}\, a_{-2}\, a_{-3})_8 = (0.24)_8$

Conversion from Decimal to Any System

For a number that has both integral and fractional parts, Conversion is done separately for both parts, and then the result is put together with a system point in between both parts.

Conversion from Any System to Any System

- In general, conversion between bases can be done via decimal:



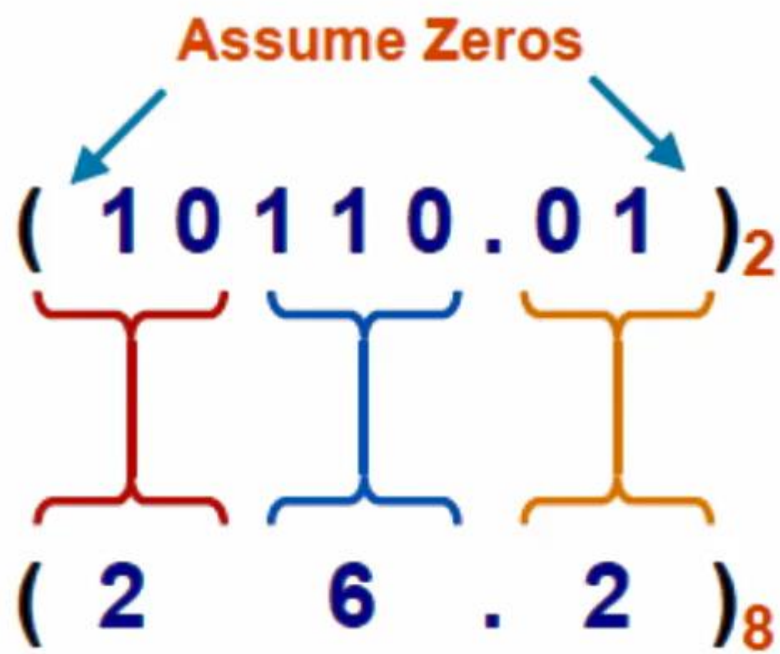
- First convert given number to decimal then convert decimal number to the new base.

Conversion from Binary to Octal System

- $8 = 2^3$
- Each group of 3 bits represents an octal digit

Octal	Binary
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

Example:



Works **both** **ways (*Binary* to *Octal* & *Octal* to *Binary*)**

Conversion from Octal to Binary System

$(705)_8 = (?)_2$

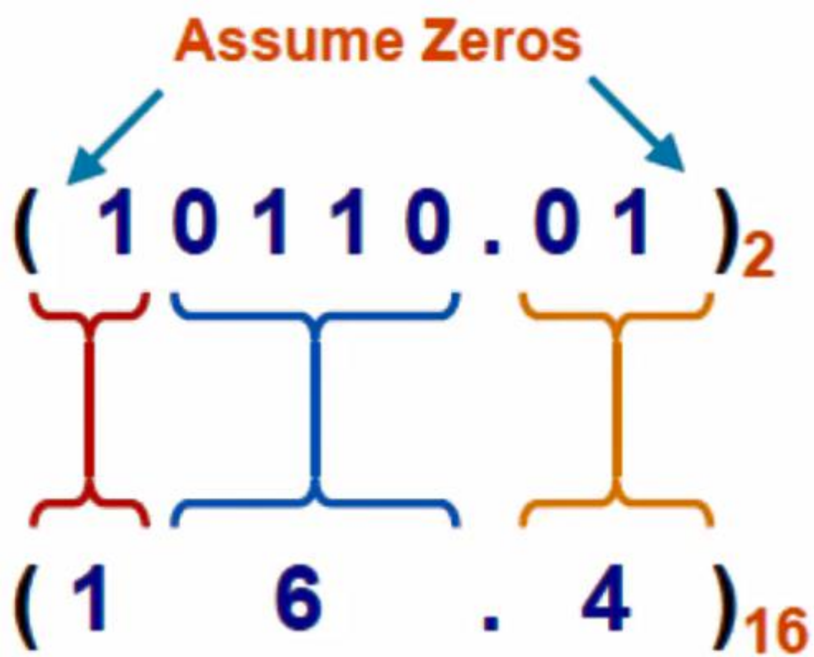
7	0	5
↓	↓	↓
111	000	101

$(705)_8 = (111000101)_2$

Conversion from Binary to Hexadecimal System

- $16 = 2^4$
- Each group of 4 bits represents a hexadecimal digit

Example:



Hex	Binary
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
A	1 0 1 0
B	1 0 1 1
C	1 1 0 0
D	1 1 0 1
E	1 1 1 0
F	1 1 1 1

Works both ways (*Binary* to *Hex* & *Hex* to *Binary*)

Conversion from Hexadecimal to Binary System

$(10AF)_{16} = (?)_2$

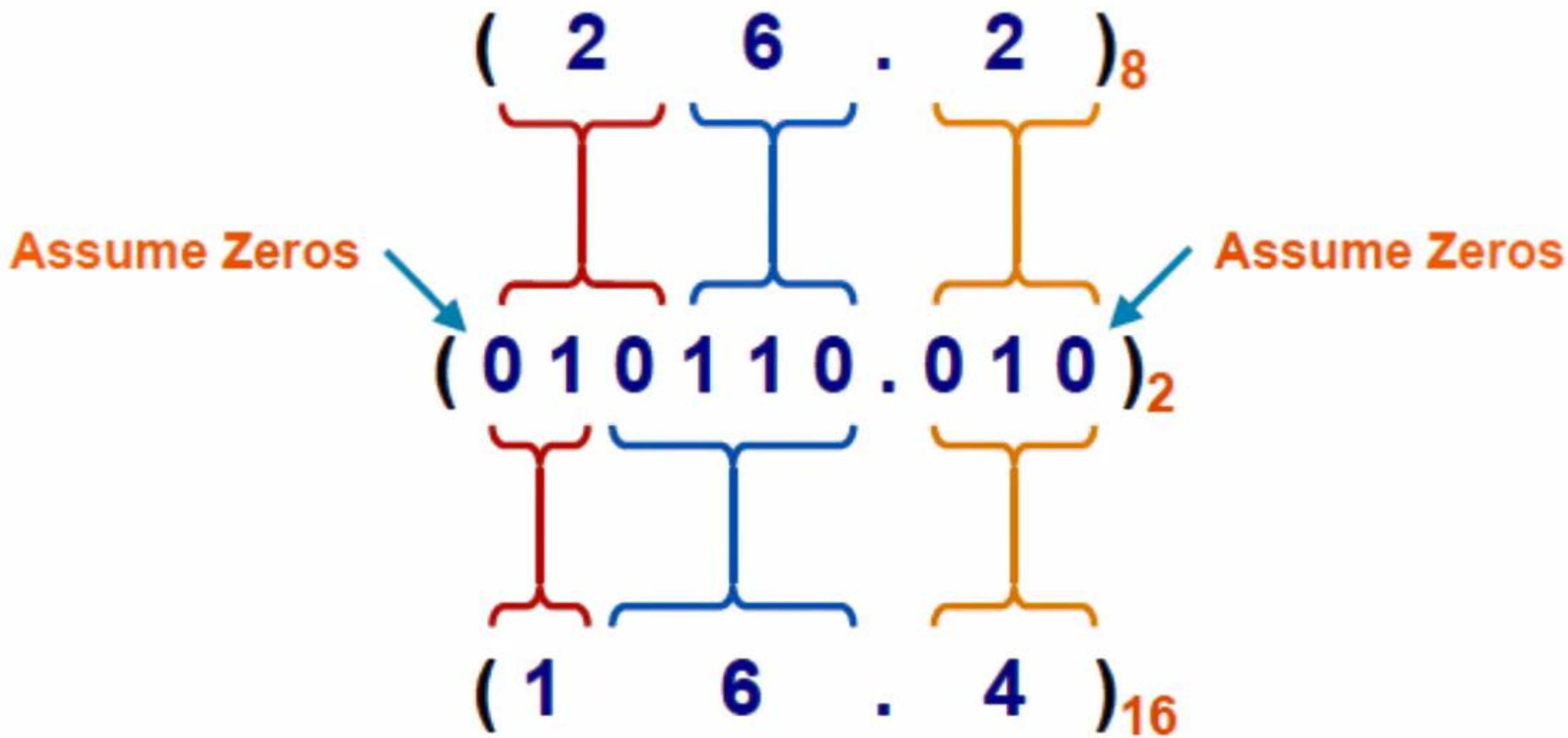
1	0	A	F
↓	↓	↓	↓
0001	0000	1010	1111

$(10AF)_{16} = (0001000010101111)_2$

Conversion from Octal to Hexadecimal System

- Convert to Binary as an intermediate step

Example:



Works both ways (*Octal* to *Hex* & *Hex* to *Octal*)

Sum of Weights Method

- To find a binary number that is equivalent to a decimal number, we can determine the set of binary weights whose sum is equal to the decimal number.

- **Example:**
- Convert the following decimal numbers to binary form: 13, 100, 65, and 189. Put your answer as eight bit numbers.

	128	64	32	16	8	4	2	1
13 =	0	0	0	0	1	1	0	1
100 =	0	1	1	0	0	1	0	0
65 =	0	1	0	0	0	0	0	1
189 =	1	0	1	1	1	1	0	1

The Powers of 2

n	2^n
0	$2^0=1$
1	$2^1=2$
2	$2^2=4$
3	$2^3=8$
4	$2^4=16$
5	$2^5=32$
6	$2^6=64$
7	$2^7=128$

n	2^n	
8	$2^8=256$	
9	$2^9=512$	
10	$2^{10}=1024$	Kilo
11	$2^{11}=2048$	
12	$2^{12}=4096$	
20	$2^{20}=1M$	Mega
30	$2^{30}=1G$	Giga
40	$2^{40}=1T$	Tera

Range of Binary Numbers

- Total combinations of a binary number consists of n-bits = 2^n different numbers in the range 0 to $(2^n – 1)$

- Examples:

- A 4-bit number can hold up to $2^4=16$ different values in the range 0 to 15 (0 to 1111).

- An 8-bit number can hold up to $2^8=256$ different values in the range 0 to 255 (0 to 11111111).

Exercise

- What is the range of values (in decimal) that can be represented by a binary number of the following number of bits: 10, 20 and 24?

- Solution:
- N=10 range = 0 to $2^{10} - 1$ = 0 to 1023
  i.e. 1024 (1K)numbers

- N=20 range =0 to $2^{20} - 1$ =0 to 1048575
  i.e. 1048576 (1M)numbers

- N=24 range =0 to $2^{24} - 1$ =0 to 16777215
  i.e. 16777216 (16M)numbers

Numbers with Different Bases

Decimal	Binary	Octal	Hex
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

You can convert between base-10, base-8 and base-16 using techniques

like the ones we just showed for converting between decimal and binary

Binary Numbers Tips

- To represent a decimal number in binary, consider the following tips:

- If the decimal number can be written as 2^n

$$1\,0\,0\,\ldots\,0\,0 \quad \underbrace{\hphantom{0\,0\,\ldots\,0\,0}}_{\text{n zeros}}$$

Decimal	Powers of 2	Binary	Decimal	Powers of 2	Binary
2	2^1	10	64	2^6	1000000
4	2^2	100	128	2^7	10000000
8	2^3	1000	256	2^8	100000000
16	2^4	10000	512	2^9	1000000000
32	2^5	100000	1024	2^{10}	10000000000

Binary Numbers Tips

- To represent a decimal number in binary, consider the following tips:

➢ If the decimal number can be written as 2^n-1

$$\underbrace{1\ 1\ 1\ 1\ \ldots\ 1\ 1}_{\text{n ones}}$$

Decimal	Powers of 2	Binary	Decimal	Powers of 2	Binary
1	$2^1 - 1$	1	63	$2^6 - 1$	111111
3	$2^2 - 1$	11	127	$2^7 - 1$	1111111
7	$2^3 - 1$	111	255	$2^8 - 1$	11111111
15	$2^4 - 1$	1111	511	$2^9 - 1$	111111111
31	$2^5 - 1$	11111	1023	$2^{10} - 1$	1111111111

Binary Numbers Tips

- In binary system, for any number such as 1011001, can we say directly "it is an odd number or an even number"?, or firstly should we convert it to decimal form, then look for is it odd or even number?

- A number is odd if and only if its binary representation ends with one.

Binary	100011	1101	10001	11111	111101
Decimal	35	13	17	31	61

- A number is even if and only if its binary representation ends with zero.

Binary	10000	10100	110	110010	1111110
Decimal	16	20	6	50	126

Binary and Octal Conversions

- Converting from octal to binary: Replace each octal digit with its equivalent 3-bit binary sequence

$$(673.12)_8 = 6 \quad 7 \quad 3 \;.\; 1 \quad 2$$
$$= 110 \quad 111 \quad 011 \;.\; 001 \quad 010$$
$$= (110111011.001010)_2$$

- Converting from binary to octal: Make groups of 3 bits, starting from the binary point. Add 0s to the ends of the number if needed. Convert each bit group to its corresponding octal digit.

$$10110100.001011_2 = 010 \quad 110 \quad 100 \;.\; 001 \quad 011_2$$
$$= 2 \quad 6 \quad 4 \;.\; 1 \quad 3_8$$

Octal	Binary
0	000
1	001
2	010
3	011

Octal	Binary
4	100
5	101
6	110
7	111

Binary and Hexadecimal Conversions

- Converting from hex to binary: Replace each hex digit with its equivalent 4-bit binary sequence

$$261.35_{16} = 2 \quad 6 \quad 1 \quad . \quad 3 \quad 5_{16}$$
$$= 0010 \quad 0110 \quad 0001 \quad . \quad 0011 \quad 0101_{2}$$

- Converting from binary to hex: Make groups of 4 bits, starting from the binary point. Add 0s to the ends of the number if needed. Convert each bit group to its corresponding hex digit

$$10110100.001011_{2} = 1011 \quad 0100 \quad . \quad 0010 \quad 1100_{2}$$
$$= B \quad 4 \quad . \quad 2 \quad C_{16}$$

Hex	Binary
0	0000
1	0001
2	0010
3	0011

Hex	Binary
4	0100
5	0101
6	0110
7	0111

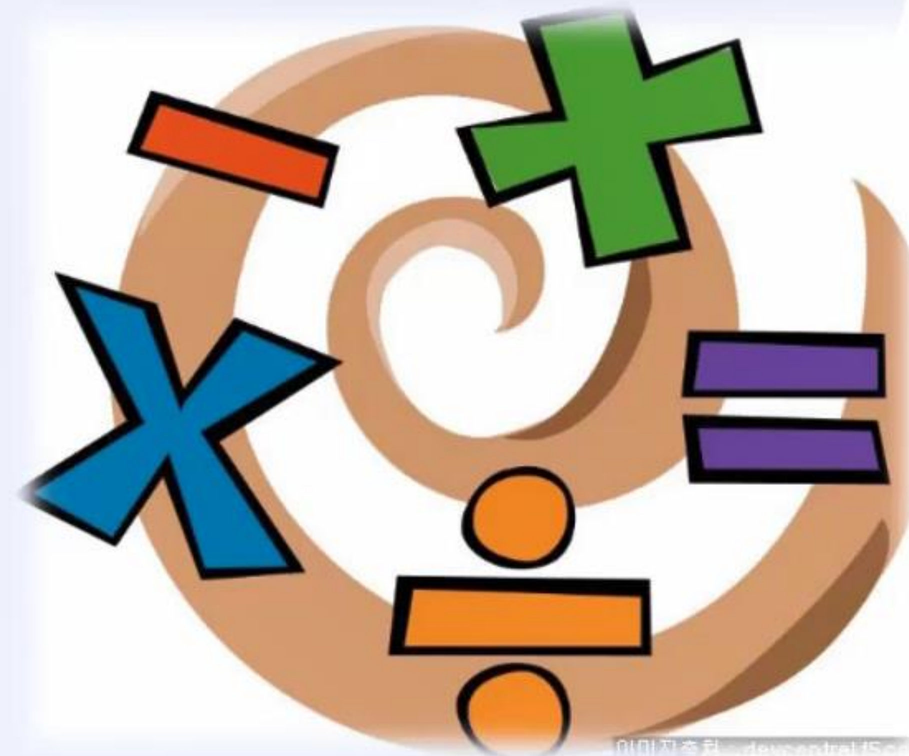
Hex	Binary
8	1000
9	1001
A	1010
B	1011

Hex	Binary
C	1100
D	1101
E	1110
F	1111

Arithmetic Operations

Arithmetic Operations on Numbering System

- Addition
- Subtraction
- Multiplication
- Division



Decimal Addition Example

Add **3758** to **4657**:

	1	1	1	
	3	**7**	**5**	**8**
+	**4**	**6**	**5**	**7**
	8	4	1	5

1) Add **8 + 7 = 15**
   Write down **5**, carry **1**

2) Add **5 + 5 + 1 = 11**
   Write down **1**, carry **1**

3) Add **7 + 6 + 1 = 14**
   Write down **4**, carry **1**

4) Add **3 + 4 + 1 = 8**
   Write down **8**

Decimal Addition Example

```
  1 1 1
   3 7 5 8
+  4 6 5 7
---------

   8 4 1 5
```

What just happened?

```
     1  1  1     (carry)

     3  7  5  8
  +  4  6  5  7
  -------------
  8  14  11  15  (sum)
    -10  10  10  (subtract the base)
     8  4  1  5
```

So when the **sum** of a column is **equal to** or **greater than** the **base**, we subtract the **base** from the **sum**, record the **difference**, and carry **one** to the next column to the left.

Binary Addition Rules

Rules:

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1 (just like in decimal)
  - 1 + 1 $= (2)_{10}$
    $= (10)_2 = 0$ with 1 to carry
  - 1 + 1 + 1 $= (3)_{10}$
    $= (11)_2 = 1$ with 1 to carry

Binary Addition Example 1

Example 1: Add binary **110111** to **11100**

1	1	1		1		
	1	1	0	1	1	1
+	0	1	1	1	0	0
1	0	1	0	0	1	1

Col 1) Add **1 + 0 = 1**
Write **1**

Col 2) Add **1 + 0 = 1**
Write **1**

Col 3) Add **1 + 1 = 2** (**10** in binary)
Write **0**, carry **1**

Col 4) Add **1+ 0 + 1 = 2**
Write **0**, carry **1**

Col 5) Add **1 + 1 + 1 = 3** (**11** in binary)
Write **1**, carry **1**

Col 6) Add **1 + 1 + 0 = 2**
Write **0**, carry **1**

Col 7) Bring down the carried 1
Write **1**

Binary Addition Explanation

What is actually happened when we carried in binary?

	1	1	1	1			
		1	1	0	1	1	1
+		0	1	1	1	0	0
		2	3	2	2		
–		2	2	2	2		
		0	1	0	0	1	1

In the first two columns, there were no carries.

In column 3, we add **1 + 1 = 2**
Since 2 is equal to the base, subtract the **base** from the **sum** and carry **1**.

In column 4, we also subtract the **base** from the **sum** and carry **1**.

In column 5, we also subtract the **base** from the **sum** and carry **1**.

In column 6, we also subtract the **base** from the **sum** and carry **1**.

Binary Addition Verification

You can always check your answer by converting the figures to decimal, doing the addition, and comparing the answers.

```
      1 1 0 1 1 1
+     0 1 1 1 0 0
---------------
    1 0 1 0 0 1 1
```

Verification

$110111_2 \rightarrow 55_{10}$

$+011100_2 \quad + 28_{10}$

83_{10}

64	32	16	8	4	2	1
1	0	1	0	0	1	1

$= 64 + 16 + 2 + 1$

$= 83_{10}$

Binary Addition Example 2

Example 2:
Add **1111** to **111010**.

```
  1 1 1 1 1
    1 1 1 0 1 0
+   0 0 1 1 1 1
---------------
  1 0 0 1 0 0 1
```

Verification

$111010_2 \rightarrow 58_{10}$

$+\underline{001111_2} \quad + \underline{15_{10}}$

73_{10}

64	32	16	8	4	2	1
1	0	0	1	0	0	1

$= 64 + 8 + 1$

$= 73_{10}$

Decimal Subtraction: Example (Borrow Method)

Subtract
4657 from **8025**:

	7	9	1	
		10	10	10
	~~8~~	~~0~~	~~2~~	5
–	4	6	5	7
	3	3	6	8

1) Try to subtract 5 – 7 → can't.
   Must borrow 10 from next column.
   Add the borrowed 10 to the original 5.
   Then subtract 15 – 7 = 8.
2) Try to subtract 1 – 5 → can't.
   Must borrow 10 from next column.
   But next column is 0, so must go to column after next to borrow.
   Add the borrowed 10 to the original 0.
   Now you can borrow 10 from this column.
   Add the borrowed 10 to the original 1..
   Then subract 11 – 5 = 6
3) Subtract 9 – 6 = 3
4) Subtract 7 – 4 = 3

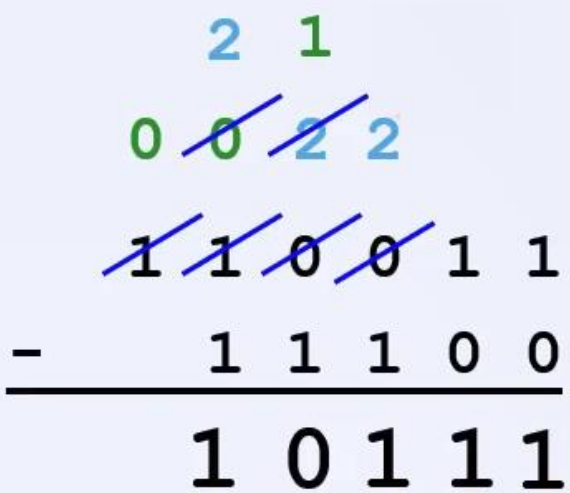
Decimal Subtraction explanation

	8	0	2	5
−	4	6	5	7
	3	3	6	8

- So when you cannot subtract, you borrow from the column to the left.
  - The amount borrowed is **1 base unit**, which in decimal is **10**.
  - The 10 is added to the original column value, so you will be able to subtract.

Binary Subtraction Example 1

Example 1: Subtract binary **11100** from **110011**



Col 1) Subtract 1 – 0 = 1

Col 2) Subtract 1 – 0 = 1

Col 3) Try to subtract 0 – 1 → can't.
Must borrow 2 from next column.
But next column is 0, so must go to column after next to borrow.
Add the borrowed 2 to the 0 on the right.
Now you can borrow from this column (leaving 1 remaining).
Add the borrowed 2 to the original 0.
Then subtract 2 – 1 = 1

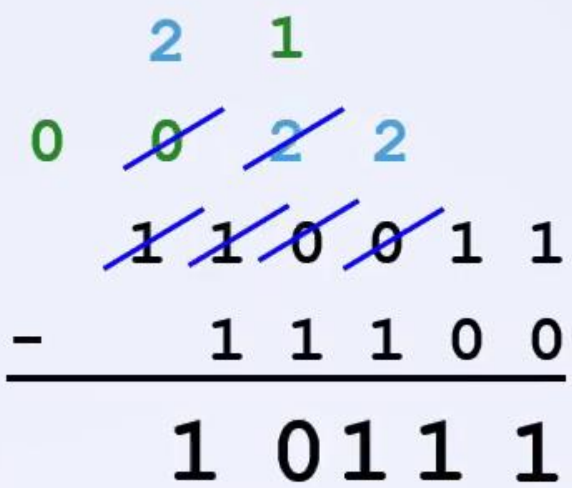
Col 4) Subtract 1 – 1 = 0

Col 5) Try to subtract 0 – 1 → can't.
Must borrow from next column.
Add the borrowed 2 to the remaining 0.
Then subtract 2 – 1 = 1

Col 6) Remaining leading 0 can be ignored.

Binary Subtraction Verification

Subtract binary
11100 from **110011**:



Verification

$110011_2 \rightarrow 51_{10}$

$- \underline{11100_2} \quad - \underline{28_{10}}$

23_{10}

64	32	16	8	4	2	1
		1	0	1	1	1

$= 16 + 4 + 2 + 1$

$= 23_{10}$

Binary Multiplication

- When performing binary multiplication, remember the following rules:
  - Copy the multiplicand when the multiplier digit is 1. Otherwise, write a row of zeros.
  - Shift your results one column to the left as you move to a new multiplier digit.
  - Add the results together using binary addition to find the product.

Binary Multiplication

- Binary multiplication uses the same technique as decimal multiplication.

- **Example:** multiplying 110_2 by 10_2.

```
    110
x    10
---------
    000
+ 110
--------
1100
```

```
  6
x 2
------
12
```

Range of Binary Numbers

- Binary, two n-bit values
  - As with decimal values
  - E.g.,

```
      1110
  ×   1011
  --------
      1110
     1110
    0000
   1110
  --------
  10011010
```

```
           1  1  0  0
        ×  1  0  1  0
        -------------
           0  0  0  0
        1  1  0  0
     0  0  0  0
  1  1  0  0
  -------------------
  1  1  1  1  0  0  0
```

Division of Binary Numbers

Division rule

- Set quotient to zero
- Repeat while dividend is greater than or equal to divisor
  - Subtract divisor from dividend
  - Add 1 to quotient
- End of repeat block
- quotient is correct, dividend is remainder
- STOP

Division of Binary Numbers

- Division as repeated subtraction
  Example in decimal 84 ÷ 21 = ??

We subtracted 21 four times,
so 84 ÷ 21 = 4

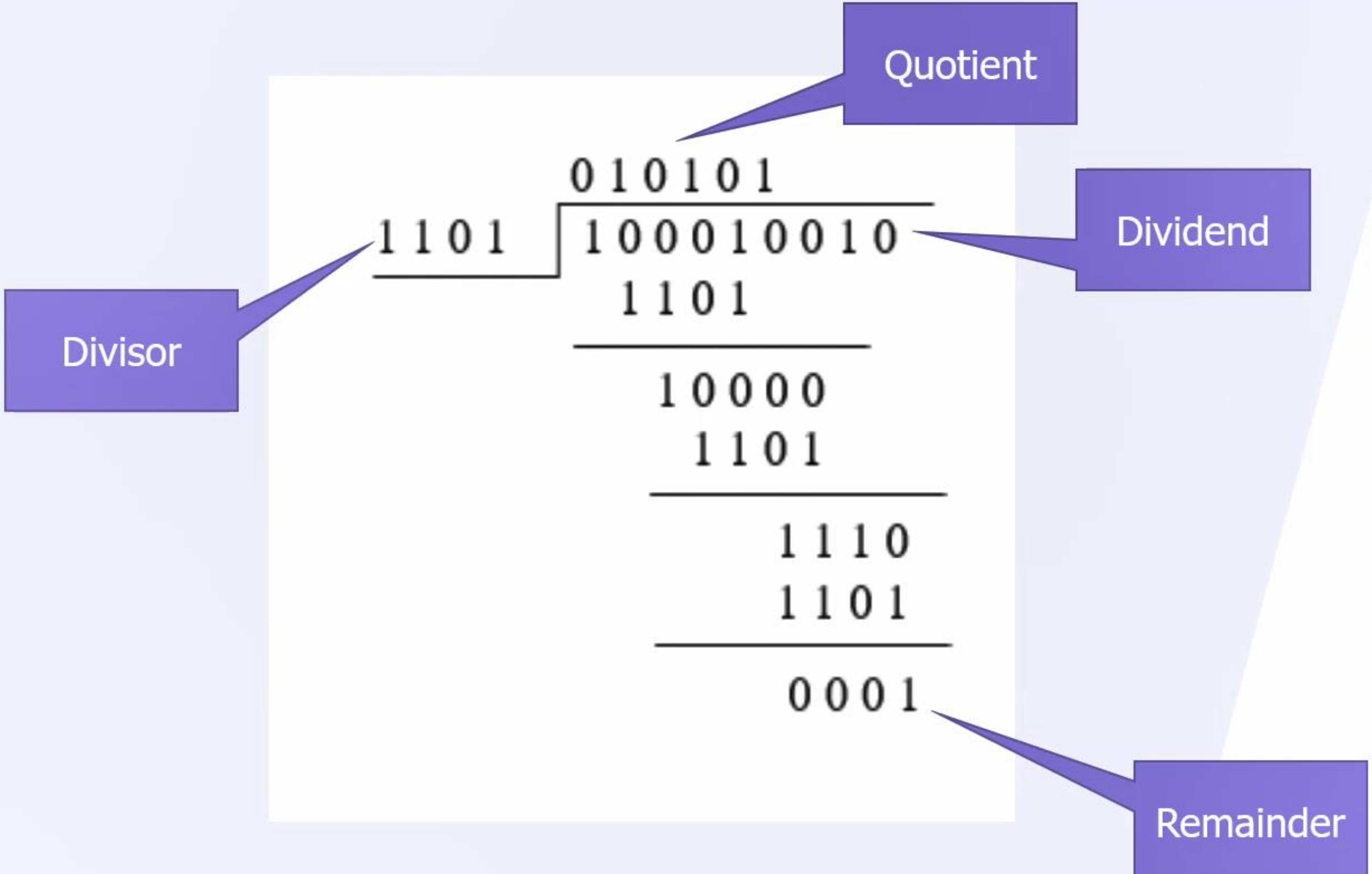
```
 84
- 21   1
------
 63
- 21   1
-------
 42
- 21   1
------
 21    1
- 21
------- ----
 0     4
```

Division of Binary Numbers

This is division by repeated subtraction.

- You subtract 21 repeatedly, or many times, till you hit zero.
- Each subtraction is forming a group of 21.
- How many groups did you form?
- How many times did you subtract?
- That is the answer to the division problem 84÷ 21=4.

Division of Binary Numbers



Division of Binary Numbers