Ollscoil na Gaillimhe
University of Galway

CT101 Computing Systems

Dr. Bharathi Raja Chakravarthi
Lecturer-above-the-bar
Email: bharathi.raja@universityofgalway.ie

UniversityofGalway.ie

The Process of Computational Problem Solving

- Computational problem solving does not simply involve the act of computer programming. It is a process , with programming being only one of the steps.
- Before a program is written, a design for the program must be developed.
- Before a design can be developed, the problem to be solved must be well understood.
- Once written, the program must be thoroughly tested.

Analyze Problem

- Clearly understand the problem
- Know the constitutes a solution

Describes Data & Algorithm

- Determine what type of data is needed
- Determine how data is to be structured
- Find and/or design appropriate algorithmss

Implement Program

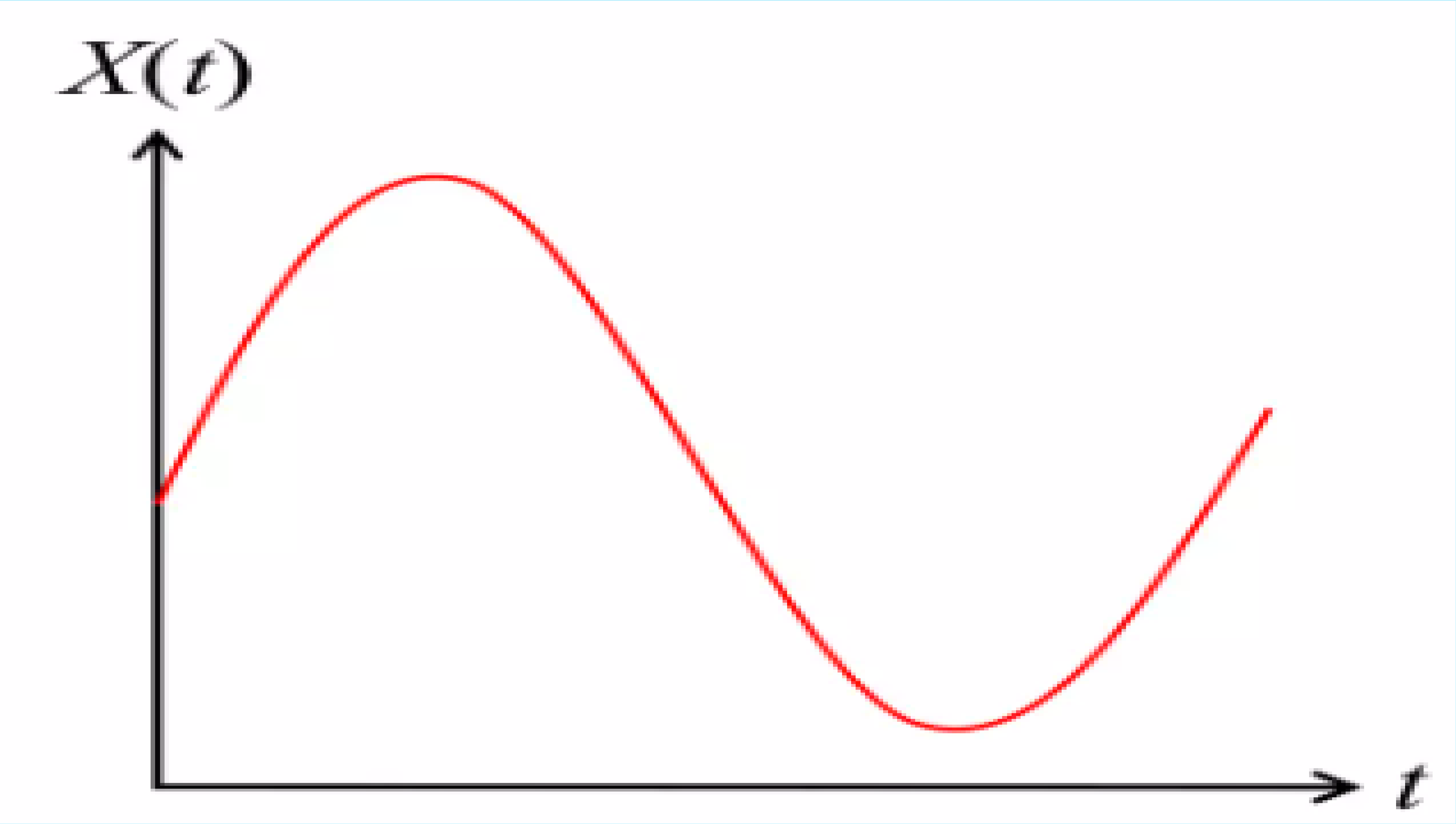
- Represent data within programming language
- Implement algorithms in programming language

Test and Debug

- Test the program on a selected set of problem instances
- Correct and understand the causes of any errors found

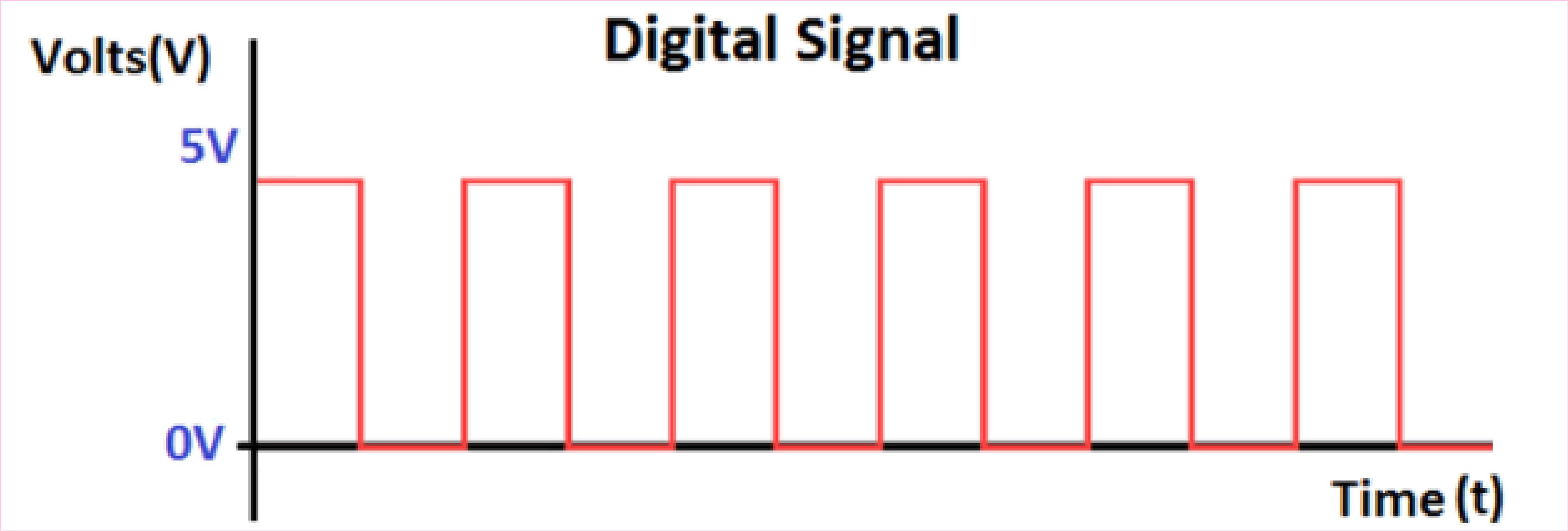
Digital System and Binary Numbers

- **Digital age**: Digital system have such a prominent role in everyday life that we refer to the present technological period
- **Digital system:**
  - Telephone switching exchanges
  - Digital camera
  - Electronic calculators
  - Digital TV
- **Discrete information processing systems**
  - Manipulate discrete elements of information
  - For example, {1, 2, 3, ...} and {A, B, C...}



Analog system

The physical quantities or signals may vary continuously over a specified range.



Digital system

The physical quantities or signals can assume only discrete values and greater accuracy

Why Digital System Important?

- ✓ It is well suited for numerical and non-numerical information processing.
- ✓ Information processing can use a general-purpose system (**computer**).
- ✓ Finite number of values in a digital signal is represented by a vector of signals with just 2 vlaues (**Binary Signals**).

Why Digital System Important?

✓ It is well suited for numerical and non-numerical information processing.

✓ Information processing can use a general-purpose system (**computer**).

✓ Finite number of values in a digital signal is represented by a vector of signals with just 2 vlaues (**Binary Signals**).

Digit	Vector
0	0000
1	0001
2	0010
3	0011

Digit	Vector
4	0100

Digit	Vector
5	0101
6	0110
7	0111
8	1000

Digital Systems

➔ Digital signals are quite insensitive to variations of component variable values.

➔ Numerical digital systems can be made more accurate by increasing the number of digits used in the representation.

➔ Complex digital systems are built as integrated circuits composed of a large number of very simple devices.

➔ It is possible to select among different implementations of systems that trade off speed and amount of hardware.

Binary Numbers

❖ 7392 represents a quantity that is equal to

$$7 * 10^3 + 3 * 10^2 + 9 * 10^1 + 2 * 10^0$$

❖ In a binary system, possible values are 0 and 1 and each digit is multiplied by $\mathbf{2^i}$

Example: 11010.11 is

$$1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0 + 1 * 2^{-1} + 1 * 2^{-2}$$

11010.11 = 26.75

Binary numbers and others

System
Binary
Octal
Decimal
Hexadecimal

Binary numbers and others

System	Radix
Binary	2
Octal	8
Decimal	10
Hexadecimal	16

Binary numbers and others

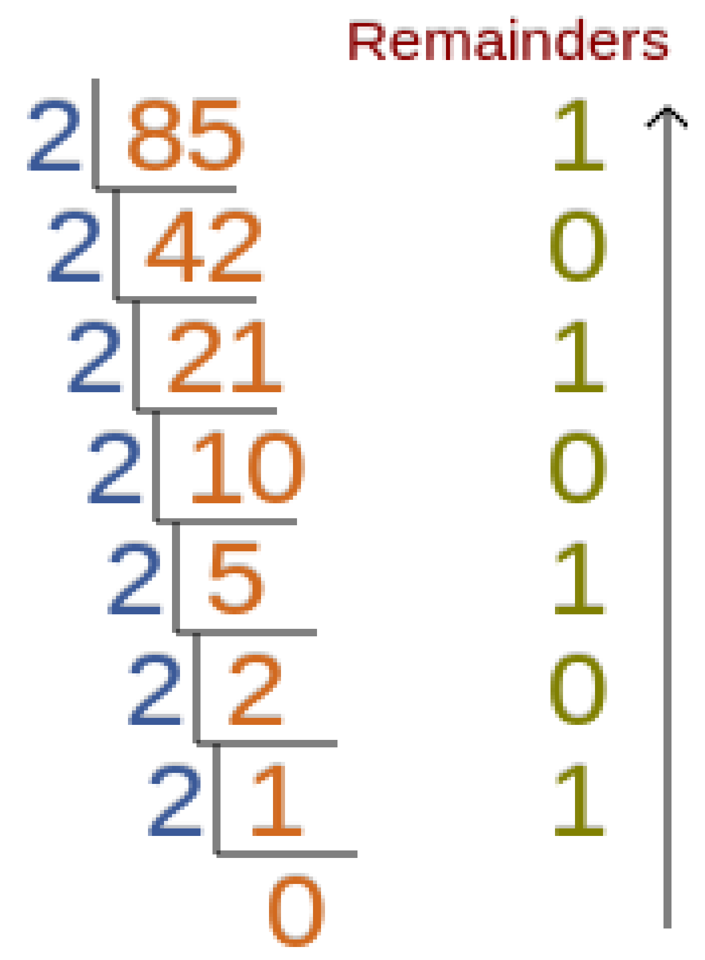
System	Radix	Allowable Digits
Binary	2	0 and 1
Octal	8	0, 1, 2, 3, 4, 5, 6, 7
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Hexadecimal	16	0 to 9 A, B, C, D, E, F

Decimal to Binary

STEP 1

WHOLE NUMBER PART

- We convert the whole number and fractional parts separately and then combine the results.
- The whole number part of **85.375** is **85**. Divide this number repeatedly by **2** until the quotient becomes **0**.



Write the remainders **from bottom to top**.

$(85)_{10} = (1010101)_2$

FRACTIONAL PART

STEP 2

The fractional part of **85.375** is **0.375**. Multiply the fractional part repeatedly by **2** until it becomes **0**.

- $0.375 \times 2 = 0.750$
- $0.750 \times 2 = 1.500$
- $0.500 \times 2 = 1.000$

STEP 3

From top to bottom, write the integer parts of the results to the fractional part of the number in base **2**.

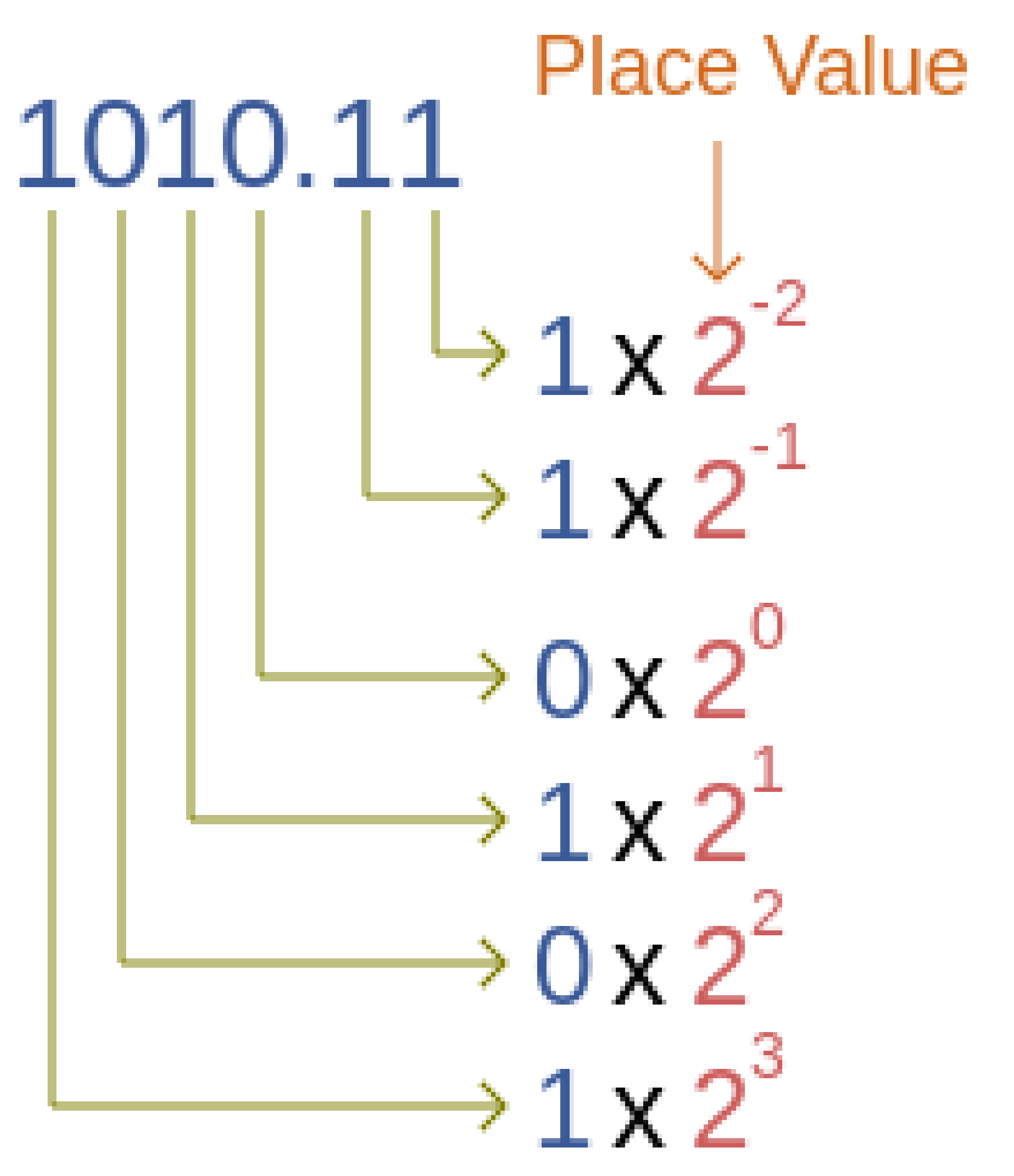
$$(0.375)_{10} = (0.011)_2$$

Combine the whole number and fractional parts to obtain the overall result.

$$(85.375)_{10} = (1010101)_2 + (0.011)_2 = (1010101.011)_2$$

Binary to Decimal

$(1010.11)_2$



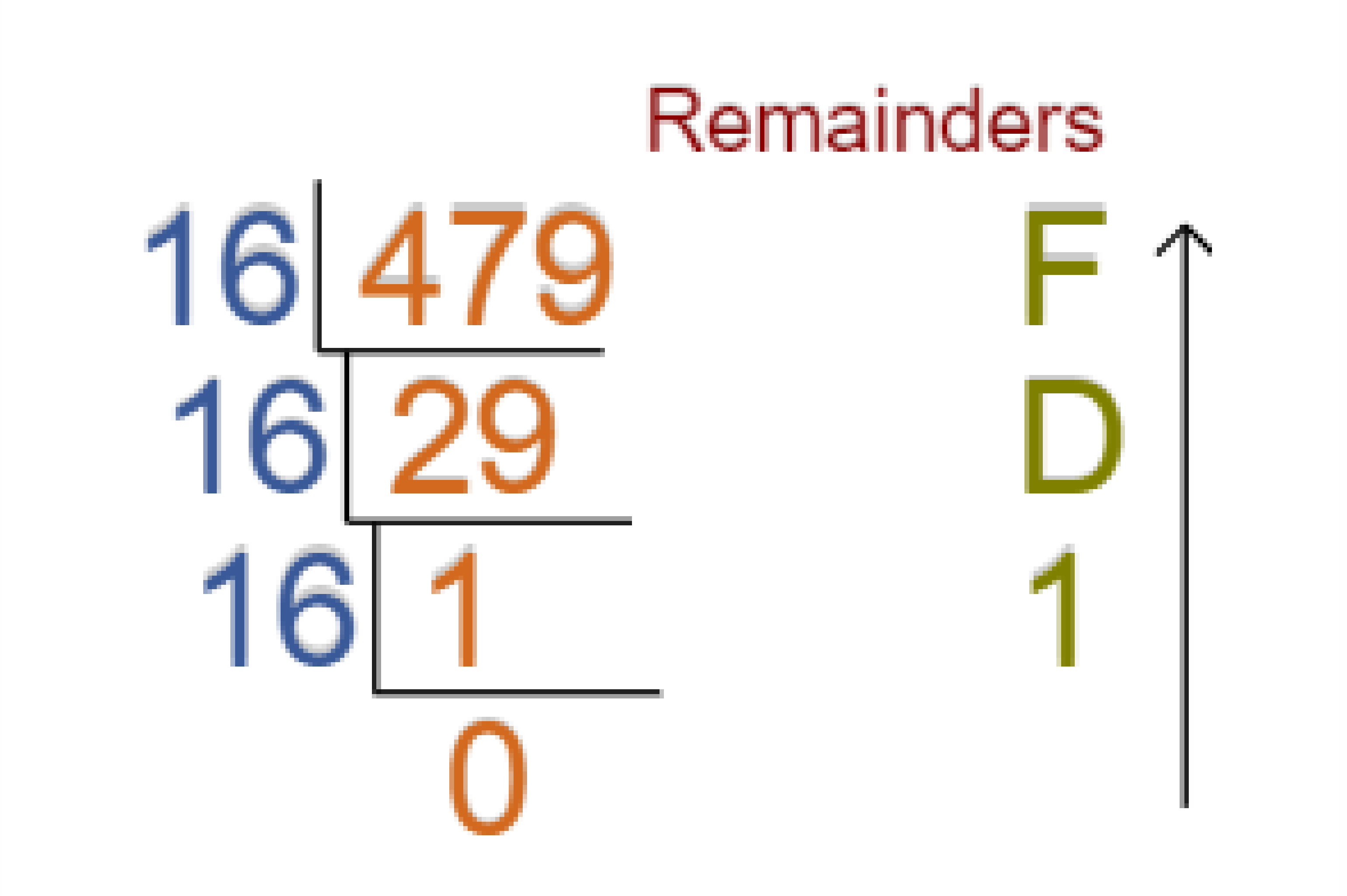
We multiply each binary digit with its place value and add the products.

$$(1010.11)_2 = (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) + (1 \times 2^{-1}) + (1 \times 2^{-2})$$

$$= 8 + 2 + \frac{1}{2} + \frac{1}{4}$$

$$= (10.75)_{10}$$

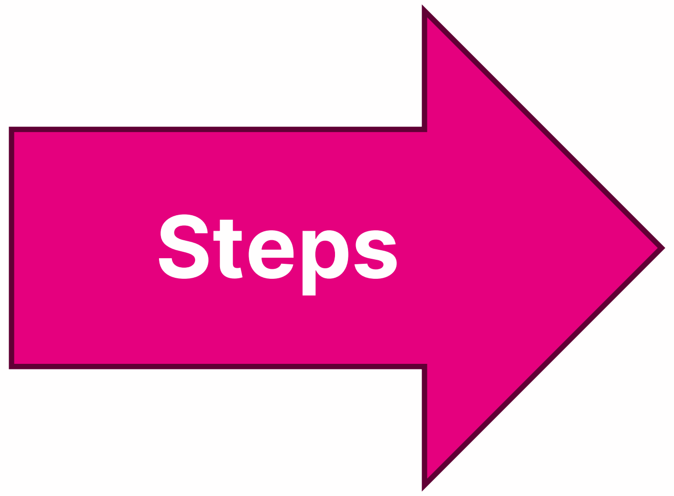
Decimal to Hexadecimal

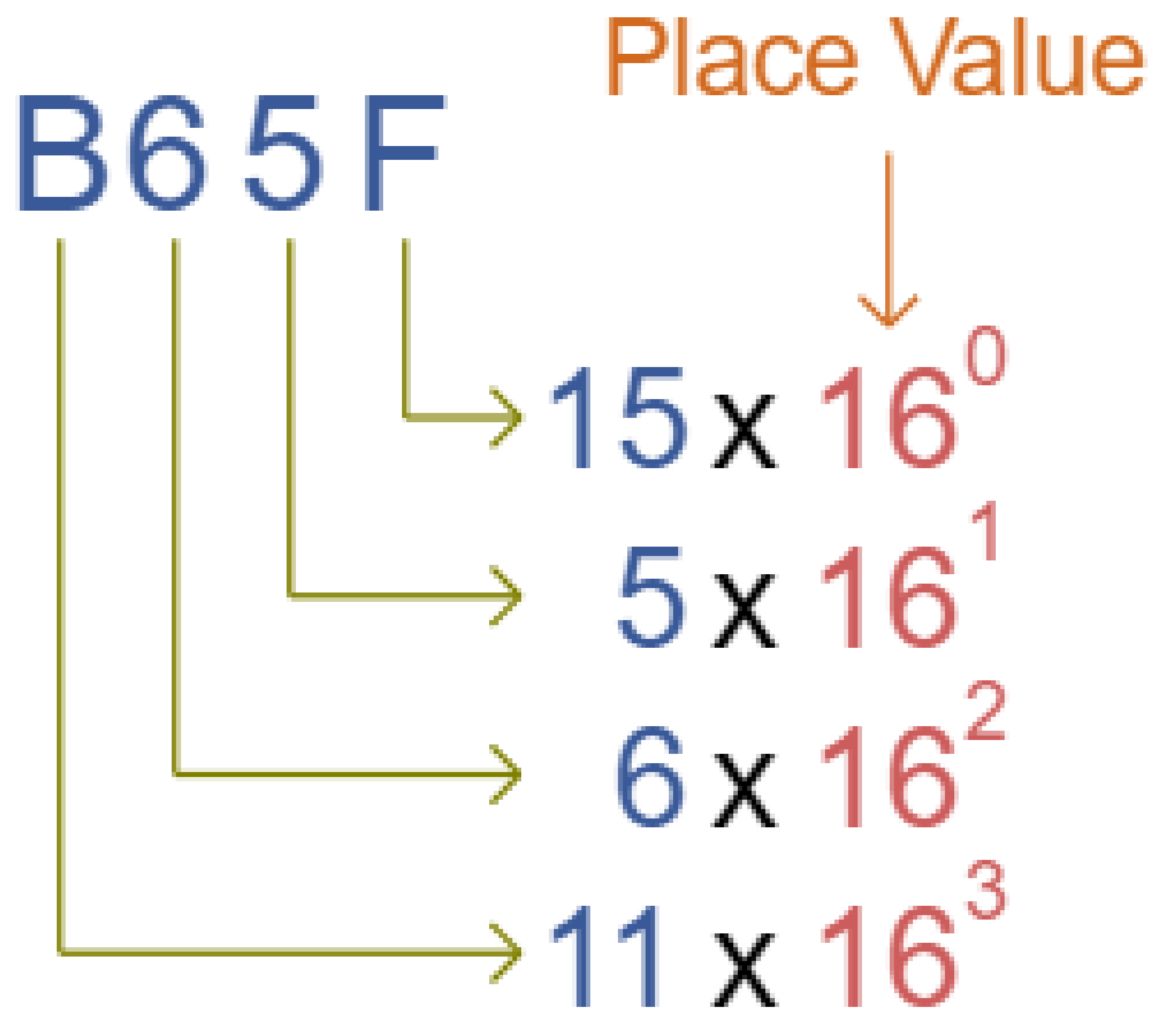


Write the remainders from **bottom to top.**

$$(479)_{10} = (1DF)_{16}$$

Hexadecimal to Decimal





$$(B65F)_{16} = (11 \times 16^3) + (6 \times 16^2) + (5 \times 16^1) + (15 \times 16^0)$$

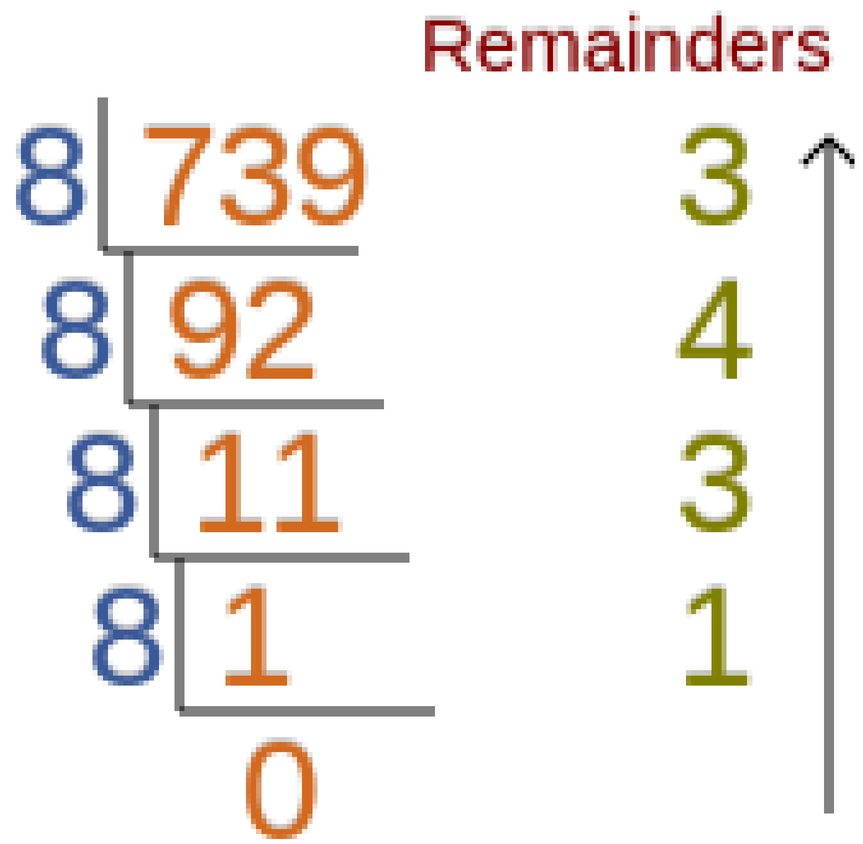
$$= 45056 + 1536 + 80 + 15$$

$$= (46687)_{10}$$

Decimal to Octal

Divide the number repeatedly by 8 until the quotient becomes 0.

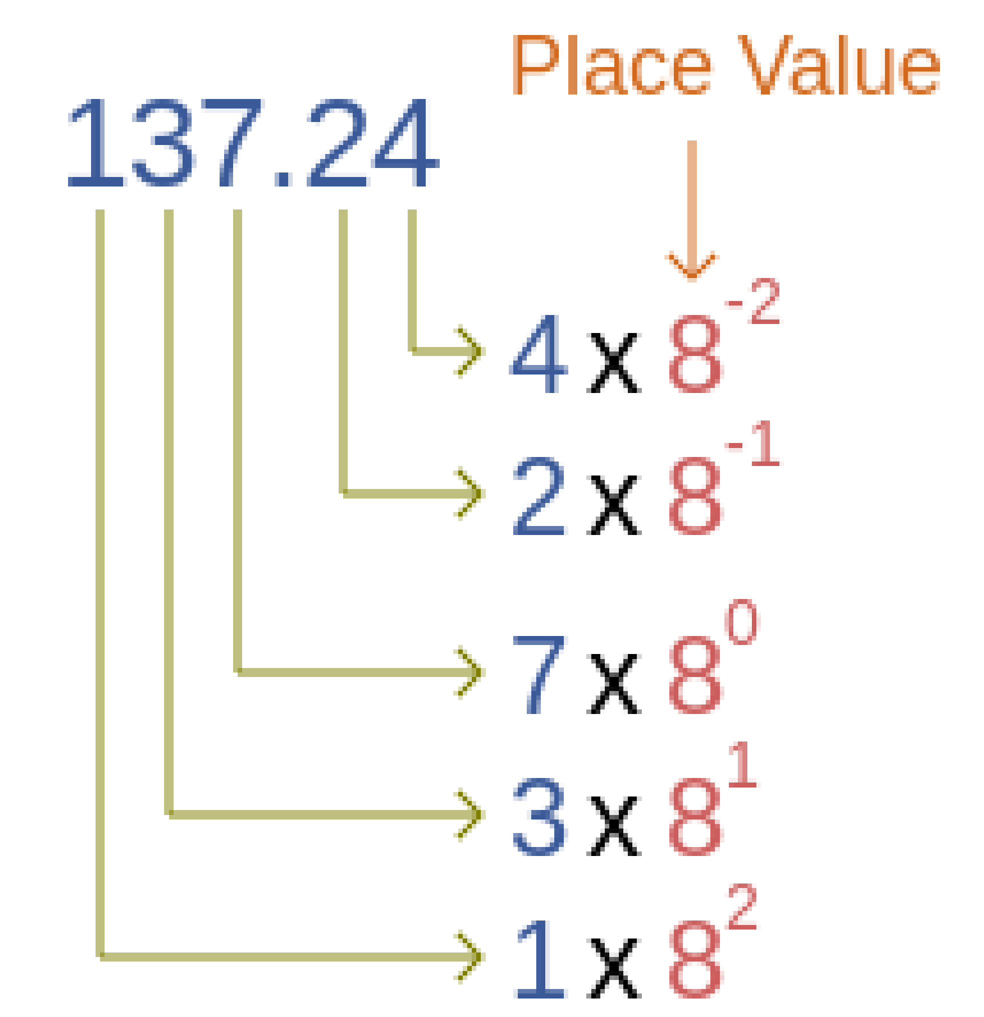
$(739)_{10}$



$(739)_{10} = (1343)_8$

Octal to Decimal

$(137.24)_8$



We multiply each digit with its place value and add the products.

$$(137.24)_8 = (1 \times 8^2) + (3 \times 8^1) + (7 \times 8^0) + (2 \times 8^{-1}) + (4 \times 8^{-2})$$

$$= 64 + 24 + 7 + \frac{2}{8} + \frac{4}{64}$$

$$= (95.3125)_{10}$$

$$(137.24)_8 = (95.3125)_{10}$$

Hexadecimal to Octal

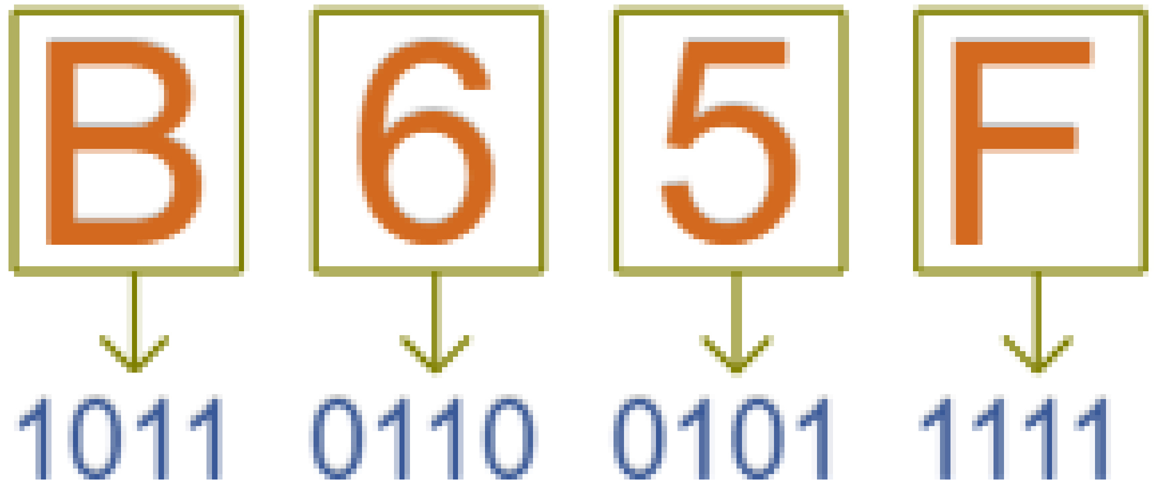
Convert each hex digit to 4 binary digits and then convert each 3 binary digits to octal digits.

Example, we can take $(B65F)_{16}$

STEP 1

Hexadecimal to Binary

In the first step, we convert the hexadecimal number to binary.



STEP 2

Binary to Octal

In the second step, we convert the binary number to octal.

001	011	011	001	011	111
1	3	3	1	3	7

Last STEP

Combining Results

Using the equalities we obtained in steps 1 and 2, we reach the following result.

$$(B65F)_{16} = (133137)_8$$

Hexadecimal to Binary

$(A46.09)_2$

To convert a hexadecimal number to binary, we write 4 bit binary equivalent of each hexadecimal digit in the same order.

A	4	6	.	0	9
1010	0100	0110		0000	1001

$$(A46.09)_2 = (101001000110.00001001)_8$$

Homework

❑ Decimal to Hexadecimal

Example: Convert $(5386)_{10}$ to a hexadecimal $(?)_{16}$ number.

Number (Division)	Quotient	Remainder
5386 / 16	336	10 = A
336 / 16	21	0
21 / 16	1	5
1 / 16	0	1

Decimal Value ⟶ Hexadecimal Value

$(5386)_{10} \longrightarrow (150\,A)_{16}$

Binary to Octal

$(1100.11011)_2$

Starting from the binary point, we partition the binary number into groups of three bits.

$$\underset{1}{\boxed{001}}\ \underset{4}{\boxed{100}}\ .\ \underset{6}{\boxed{110}}\ \underset{6}{\boxed{110}}$$

In the **integer part**, we proceed to the left. To complete the leftmost group of bits, we append **two zeros** to the left.

In the **fractional part**, we proceed to the right. To complete the rightmost group of bits, we append **a zero** to the right.

$(001)_2 = (1)_8$

$(100)_2 = (4)_8$

$(110)_2 = (6)_8$

$(110)_2 = (6)_8$

We convert each group of binary numbers to octal and write them in the same order.

$(1100.11011)_2 = (14.66)_8$

Octal to Binary

$(1743)_8$

To convert an octal number to binary, we write 3 bit binary equivalent of each octal digit in the same order.

1	7	4	3
↓	↓	↓	↓
001	111	100	011

$$(1743)_8 = (001111100011)_2$$

Octal to Hexadecimal

$(46.1)_8$

We can convert an octal number to hexadecimal in two steps.

STEP 1

In the first step, we convert the octal number to binary.

$$\boxed{4}\,\boxed{6}\;.\;\boxed{1}$$

$$\begin{array}{ccccc} 4 & 6 & . & 1 \\ \downarrow & \downarrow & & \downarrow \\ 100 & 110 & & 001 \end{array}$$

To convert an octal number to binary, we write 3 bit binary equivalent of each octal digit in the same order.

$$(46.1)_8 = (100110.001)_2$$

STEP 2

In the second step, we convert the binary number to hexadecimal.

$$\underbrace{\boxed{0010}}_{2}\ \underbrace{\boxed{0110}}_{6}\ .\ \underbrace{\boxed{0010}}_{2}$$

Starting from the binary point, we partition the binary number into groups of 4 bits. In the whole number part, we proceed to the left and in the fractional part, we proceed to the right.

$$(100110.001)_2 = (26.2)_{16}$$

STEP 3

COMBINING RESULTS

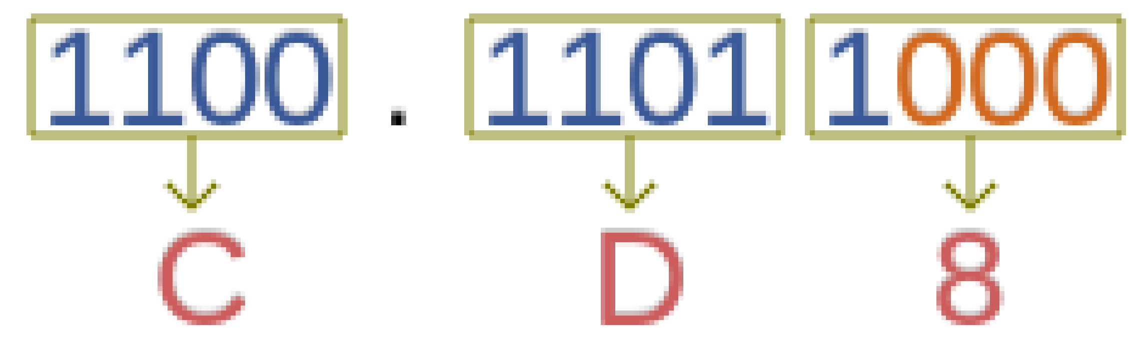
Using the equalities we obtained in steps 1 and 2, we reach the following result.

$$(46.1)_8 = (26.2)_{16}$$

Binary to Hexadecimal

$(1100.11011)_2$

Starting from the binary point, we partition the binary number into groups of four bits.



In the **integer part**, we proceed to the left.

In the **fractional part**, we proceed to the right. To complete the rightmost group of bits, we append **three zeros** to the right.

$(1100)_2 = (C)_{16}$

$(1101)_2 = (D)_{16}$

$(1000)_2 = (8)_{16}$

In the **integer part**, we proceed to the left.

In the **fractional part**, we proceed to the right. To complete the rightmost group of bits, we append **three zeros** to the right.

$$(1100)_2 = (C)_{16}$$

$$(1101)_2 = (D)_{16}$$

$$(1000)_2 = (8)_{16}$$

We convert each group of binary numbers to octal and write them in the same order.

$$(1100.11011)_2 = (C.D8)_{16}$$

Binary Numbers

Operations work similarly in all bases.

Augend: 101101
Addend: +100111

1010100

Minuend: 101101
Subtrahend: -100111

000110

Multiplicand: 1011
Multiplier: x 101

1011
0000
1011

Product: 110111

References

- Computer Organization and Architecture Designing for Performance Tenth Edition by William Stallings
- Digital Design With an Introduction to the Verilog HDL FIFTH EDITION by M Morris, M. and Michael, D., 2013.

Ollscoil na Gaillimhe
University of Galway

Thank *you*