## Number Systems and Binary Arithmetic

A.Y. 2020/2021 Prof. Alessandro Ortis

ortis@dmi.unict.it

#### **Introduction to Numbering Systems**

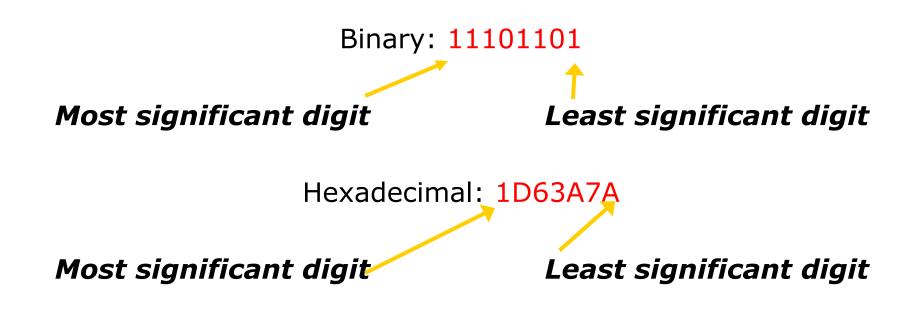
- We are all familiar with the decimal number system (Base 10). Some other number systems that we will work with are:
  - Binary → Base 2
    Octal → Base 8
    Hexadecimal → Base 16

#### **Characteristics of Numbering Systems**

- 1) The digits are consecutive.
- 2) The number of digits is equal to the size of the base.
- 3) Zero is always the first digit.
- 4) The base number is never a digit.
- 5) When 1 is added to the largest digit, a sum of zero and a carry of one results.
- 6) Numeric values are determined by the implicit positional values of the digits.



## **Significant Digits**



# **Binary Number System**

- Also called the "Base 2 system"
- The binary number system is used to model the series of electrical signals computers use to represent information
- O represents the no voltage or an off state
- 1 represents the presence of voltage or an on state

# **Binary Numbering Scale**

<u>Base 2</u> Number	<u>Base 10</u> Equivalent	Power	<u>Positional</u> <u>Value</u>
000	0	<b>2</b> <sup>0</sup>	1
001	1	<b>2</b> <sup>1</sup>	2
010	2	<b>2</b> <sup>2</sup>	4
011	3	<b>2</b> <sup>3</sup>	8
100	4	<b>2</b> <sup>4</sup>	16
101	5	<b>2</b> <sup>5</sup>	32
110	6	<b>2</b> <sup>6</sup>	64
111	7	27	128

# **Decimal to Binary Conversion**

- The easiest way to convert a decimal number to its binary equivalent is to use the *Division Algorithm*
- This method repeatedly divides a decimal number by 2 and records the quotient and remainder
  - The remainder digits (a sequence of zeros and ones) form the binary equivalent in least significant to most significant digit sequence

## **Division Algorithm**

#### **Convert 67 to its binary equivalent:**

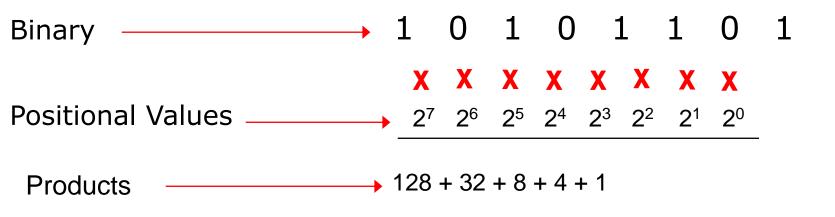
 $67_{10} = X_2$ Step 1: 67 / 2 = 33 R 1 Divide 67 by 2. Record quotient in next row Step 2: 33 / 2 = 16 R 1 Again divide by 2; record quotient in next row Step 3: 16 / 2 = 8 R 0 Repeat again Step 4: 8 / 2 = 4 R 0. Repeat again Step 5: 4 / 2 = 2 R 0 Repeat again Step 6: 2 / 2 = 1 R 0 Repeat again STQP when quotient equals 0 Step 7: 1 / 2 = 0 R 1  $1000011_{2}$ 

# **Binary to Decimal Conversion**

- The easiest method for converting a binary number to its decimal equivalent is to use the *Multiplication Algorithm*
- Multiply the binary digits by increasing powers of two, starting from the right
- Then, to find the decimal number equivalent, sum those products

## **Multiplication Algorithm**

#### **Convert (10101101)**<sub>2</sub> to its decimal equivalent:



**173**<sub>10</sub>

## **Octal Number System**

- Also known as the Base 8 System
- Uses digits 0 7
- Readily converts to binary
- Groups of three (binary) digits can be used to represent each octal digit
- Also uses multiplication and division algorithms for conversion to and from base 10

# **Hexadecimal Number System**

- Base 16 system
- Uses digits 0-9 & letters A,B,C,D,E,F
- Groups of four bits represent each base 16 digit

Decimal	Hexadecimal	
0	0	
1	1	
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	
9	9	
10	A	
11	В	
12	С	
13	D	
14	E	
15	F	

#### **Decimal to Hexadecimal Conversion**

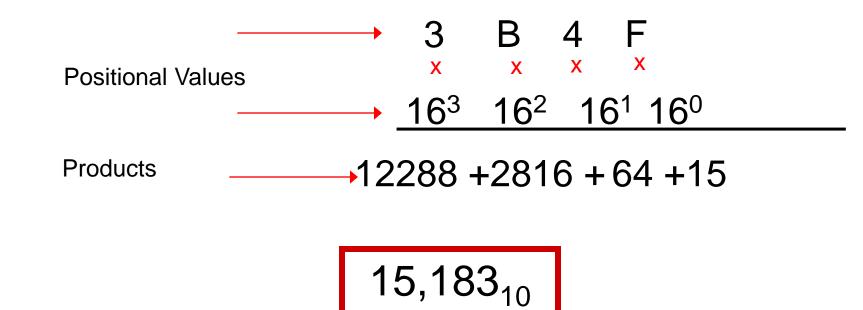
Convert 830<sub>10</sub> to its hexadecimal equivalent:

$$830 / 16 = 51 R 14 - = E in Hex$$
  
 $51 / 16 = 3 R 3$   
 $3 / 16 = 0 R 3$   
 $33E_{16}$ 

### **Hexadecimal to Decimal Conversion**

Convert 3B4F16 to its decimal equivalent:

Hex Digits



### **Binary to Hexadecimal Conversion**

- The easiest method for converting binary to hexadecimal is to use a substitution code
- Each hex number converts to 4 binary digits

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

