BACS1024 INTRODUCTION TO COMPUTER SYSTEMS

Chapter 2: Numerical Data Representation

0. Overview

1. Alphanumeric character data representation
2. Numbering Systems
3. Arithmetic conversions
4. Arithmetic operations

1. Alphanumeric Character Data Representation

1. <u>Alphanumeric Character Data Representation</u>

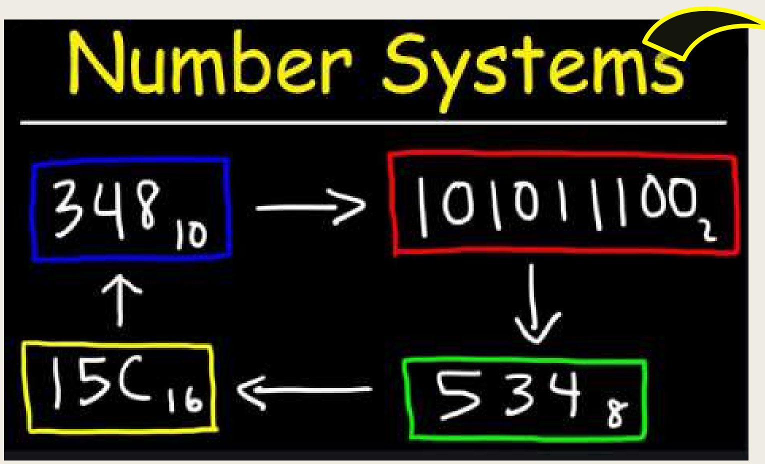
- In the computing environment, it is easier for the computer to process numbers than text. Therefore, the characters used are encoded.
- Computer uses alphanumeric character data code / representation to map the data to strings of binary digits.
- The 2 major types of alphanumeric & character data representation used are:

	EBCDIC	ASCII
	Extended Binary Coded Decimal Interchange Code	American Standard Code for Info Interchange
Founder	IBM	Other hardware manufacturer
Character set	More	Less
Letter order	Random	Linear
Compatibility	Low	High

2. Numbering Systems

2. Numbering Systems

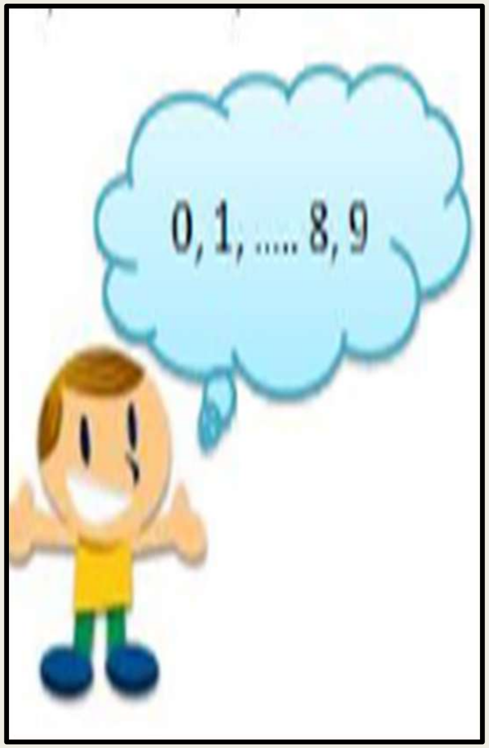
- Numbering system is essential to illustrate how numbers work, the nature of counting and the arithmetic computations performed
- Human count and perform arithmetic using **decimal** , or base 10 number while computer uses **binary** system.
- To represent a binary-coded values in computing & digital electronics, **octal** number and **hexadecimal** numbers are used.



	Binary	Octal	Decimal	Hexadecimal
Base	2	8	10	16
Data range	0 - 1	0 – 7	0 – 9	0 – 9 & A – F
Usage	Computer	Programmer	User	Programmer

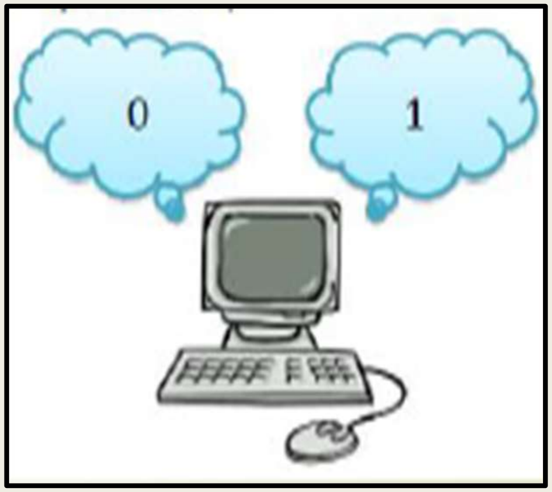
2. Numbering Systems

- **Decimal number**
  - ❑ Base **10**
  - ❑ Data range: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**.
  - ❑ Each number is multiplied by 10 raised to a power corresponding to that digit's position.
  - ❑ E.g.:
    - ❖ 78 = $(7 \times \mathbf{10}^{1}) + (8 \times \mathbf{10}^{0})$
    - ❖ 1024 = $(1 \times \mathbf{10}^{3}) + (0 \times \mathbf{10}^{2}) + (2 \times \mathbf{10}^{1}) + (4 \times \mathbf{10}^{0})$



2. Numbering Systems

- **Binary number**
  - Base **2**
  - Data range: **0** & **1**
    - 1 = ON / TRUE
    - 0 = OFF / FALSE
  - Bit: The fundamental building block of computer storage
  - Byte: A combination of 8 bits



0	1	0	0	0	0	0	1

= "**A**"

MSB **LSB**

- Most significant bit (MSB) & Least Significant Bit (LSB)

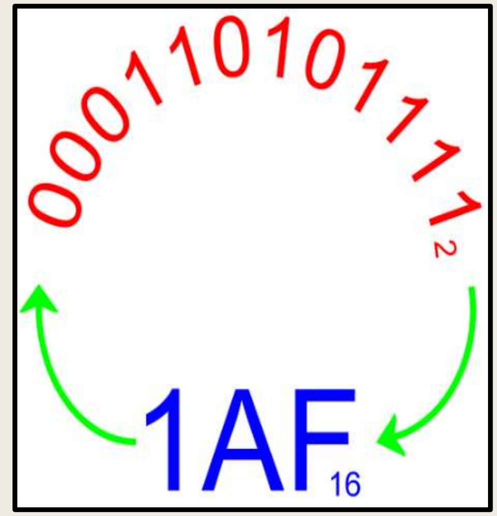
2. Numbering Systems

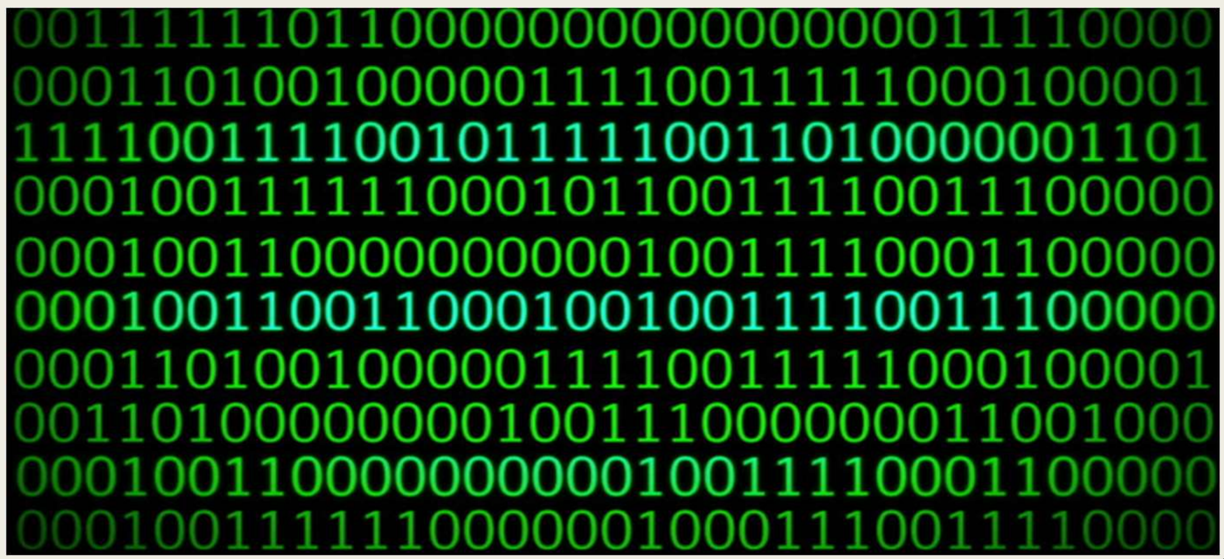
- **Octal number**
  - ❑ Base **8**
  - ❑ Data range: **0, 1, 2, 3, 4, 5, 6, 7**

Decimal	Octal	Binary
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111

2. <u>Numbering Systems</u>

- **Hexadecimal number**
  - Base **16**
  - Rata range: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F**
  - Computer works with binary numbering system
  - Example:





2. Numbering Systems

- **Hexadecimal number**

Hex	Dec	Oct	Bin
00	00	00	00000
01	01	01	00001
02	02	02	00010
03	03	03	00011
04	04	04	00100
05	05	05	00101
06	06	06	00110
07	07	07	00111
08	08	10	01000
09	09	11	01001
0A	10	12	01010
0B	11	13	01011
0C	12	14	01100
0D	13	15	01101
0E	14	16	01110
0F	15	17	01111
10	16	20	10000

- A set of <u>**4 bits**</u> in binary is used to represent a digit in hexadecimal number
- Why Hex is used?
  - ❑ For conversion of Binary & Hexadecimal is easier
  - ❑ To solve the problem of numbers written in binary which is tend to be long and difficult to express.
- E.g.:
  - ❑ 100111000001_2
    - $= 1001\ 1100\ 0001_2$
    - $= 9\ \ C\ \ 1_{16}$

3. Arithmetic Conversions

3. Arithmetic Conversion

- E.g.: Base **2** → Base **10**

$111_2 = (1 \times \mathbf{2}^2) + (1 \times \mathbf{2}^1) + (1 \times \mathbf{2}^0)$

$= 4 + 2 + 1$

$= 7_{10}$

$11001_2 = (1 \times \mathbf{2}^4) + (1 \times \mathbf{2}^3) + (0 \times \mathbf{2}^2) + (0 \times \mathbf{2}^1) + (1 \times \mathbf{2}^0)$

$= 16 + 8 + 1$

$= 25_{10}$

3. Arithmetic Conversion

- E.g.: Base **8** → Base **10**

$17_{\mathbf{8}} = (1 \times \mathbf{8}^{1}) + (7 \times \mathbf{8}^{0})$

$= 8 + 7$

$= 15_{\mathbf{10}}$

$7263_{\mathbf{8}} = (7 \times \mathbf{8}^{3}) + (2 \times \mathbf{8}^{2}) + (6 \times \mathbf{8}^{1}) + (3 \times \mathbf{8}^{0})$

$= 3584 + 128 + 48 + 3$

$= 3763_{\mathbf{10}}$

3. Arithmetic Conversion

- E.g.: Base **16** → Base **10**

$6704_{16} = (6 \times \mathbf{16}^3) + (7 \times \mathbf{16}^2) + (0 \times \mathbf{16}^1) + (4 \times \mathbf{16}^0)$

$= 24576 + 1792 + 0 + 4$

$= 26372_{10}$

$2C_{16} = (2_{16} \times \mathbf{16}^1) + (C_{16} \times \mathbf{16}^0)$

$= (2_{10} \times \mathbf{16}^1) + (12_{10} \times \mathbf{16}^0)$

$= 44_{16}$

3. Arithmetic Conversion

- E.g.: Base **2** → Base **16**

$111_{\mathbf{2}} = 0111_{\mathbf{2}}$
$= 7_{\mathbf{16}}$

$101\ 1100_{\mathbf{2}} = 0101\ 1100_{\mathbf{2}}$
$= 5C_{\mathbf{16}}$

3. Arithmetic Conversion

- E.g.: Base **2** → Base **8**

$111_2 = 7_8$

$101\ 1100_2 = 01\ 011\ 100_2$

$= 134_8$

3. Arithmetic Conversion

- E.g.: Base **16** → Base **2**

$16_{16} = 0001\ 0110_2$

$CAD_{16} = 1100\ 1010\ 1101_2$

3. Arithmetic Conversion

- E.g.: Base **8** → Base **2**

$16_8 = 001\ 110_2$

$725_8 = 111\ 010\ 101_2$

3. Arithmetic Conversion

- E.g.: Base **10** → Base **2**

35_{10}

2) <u>35</u>
2) <u>17</u> – 1
2) <u> 8</u> – 1
2) <u> 4</u> – 0
2) <u> 2</u> – 0
      1 – 0

$35_{10} = 100011_2$

3. Arithmetic Conversion

- E.g.: Base **10** → Base **8**

135_{10}

8) 135

8) 16 – 7

2 – 0

$135_{10} = 207_8$

3. Arithmetic Conversion

- E.g.: Base **10** → Base **16**

8151_{10}

16) 8151
16)   509 – 7
16)   31 – 13 (D)
            1 – 15 (F) $8151_{10} = 1FD7_{16}$

4. <u>Arithmetic Operations</u>

4. Arithmetic Operations

- **Binary Addition**

Binary addition Table

+	0	1
0	0	1
1	1	10

$$
\begin{array}{rrrrrr}
 & & & & & 1_2 \\
 & & & & 1_2 & 1_2 \\
0_2 & 0_2 & 1_2 & 1_2 & 1_2 & 1_2 \\
\underline{+\ 0_2} & \underline{+\ 1_2} & \underline{+\ 0_2} & \underline{+\ 1_2} & \underline{+\ 1_2} & \underline{+\ 1_2} \\
\underline{0_2} & \underline{1_2} & \underline{1_2} & \underline{10_2} & \underline{11_2} & \underline{100_2}
\end{array}
$$

4. Arithmetic Operations

- **Octal Addition**

$$
\begin{array}{rrrr}
 & & & 2_8 \\
 & & 2_8 & 5_8 \\
0_8 & 2_8 & 5_8 & 7_8 \\
\underline{+\ 7_8} & \underline{+\ 7_8} & \underline{+\ 7_8} & \underline{+\ 3_8} \\
\underline{0_8} & \underline{11_8} & \underline{16_8} & \underline{21_8}
\end{array}
$$

4. Arithmetic Operations

- **Hexadecimal Addition**

			B_{16}
		2_{16}	A_{16}
1_{16}	7_{16}	A_{16}	C_{16}
$+ C_{16}$	$+ 7_{16}$	$+ 7_{16}$	$+ 1_{16}$
D_{16}	E_{16}	13_{16}	22_{16}

4. Arithmetic Operations

- **Binary Subtraction**

Remaining value after borrow

Borrowed

	$\boxed{0}\ \boxed{2}$	$0\ 2$
11_2	$\cancel{1}0_2$	$\cancel{1}01_2$
$-\ 1_2$	$-\ 1_2$	$-\ 11_2$
10_2	1_2	10_2

4. Arithmetic Operations

- **Octal Subtraction**

Remaining value after borrow

Borrowed: 8 + 2 = 10

$$
\begin{array}{r}
7_8 \\
-\ \underline{1_8} \\
\underline{6_8}
\end{array}
\qquad
\begin{array}{r}
\boxed{0}\,\boxed{8} \\
\cancel{1}2_8 \\
-\ \underline{7_8} \\
\underline{3_8}
\end{array}
\qquad
\begin{array}{r}
705_8 \\
-\ \underline{213_8} \\
\underline{472_8}
\end{array}
$$

4. Arithmetic Operations

- **Hexadecimal Subtraction**

Remaining value after borrow

Borrowed: 16 + 7 = 23

	$\boxed{0}\ \boxed{16}$	
$1F_{16}$	$\cancel{1}7_{16}$	1024_{16}
$-\ C_{16}$	$-\ C_{16}$	$-\ BAC_{16}$
13_{16}	B_{16}	478_{16}

4. Arithmetic Operations

- **Binary Multiplication**

$$
\begin{array}{r}
10_2 \\
\times\quad 1_2 \\
\hline
10_2 \\
\hline
\end{array}
\qquad\qquad
\begin{array}{r}
1011_2 \\
\times\quad 111_2 \\
\hline
1011 \\
1011 \\
1011 \\
\hline
1001101_2
\end{array}
\qquad\qquad
\begin{array}{r}
1011_2 \\
\times\quad 101_2 \\
\hline
1011 \\
1011 \\
\hline
110111_2
\end{array}
$$

4. Arithmetic Operations

- Octal Multiplication

$$\begin{array}{r} 7_8 \\ \times \quad \underline{5_8} \\ \underline{43_8} \end{array} \qquad \begin{array}{r} \mathbf{1}2_8 \\ \times \quad \underline{5_8} \\ \underline{5_1\ 10}_{10} \\ =\mathbf{6}\ \ \mathbf{2}_8 \end{array} \qquad \begin{array}{r} 315_8 \\ \times \quad \underline{22_8} \\ 6\ \ 2\ \ 10_{\mathbf{10}} \\ \underline{6\ 2\ 10}_{\mathbf{10}}\quad \\ 6_1 8_1 12_1 10_{\mathbf{10}} \\ = 7\ \ 1\ \ 5\ \ 2_8 \end{array}$$

4. Arithmetic Operations

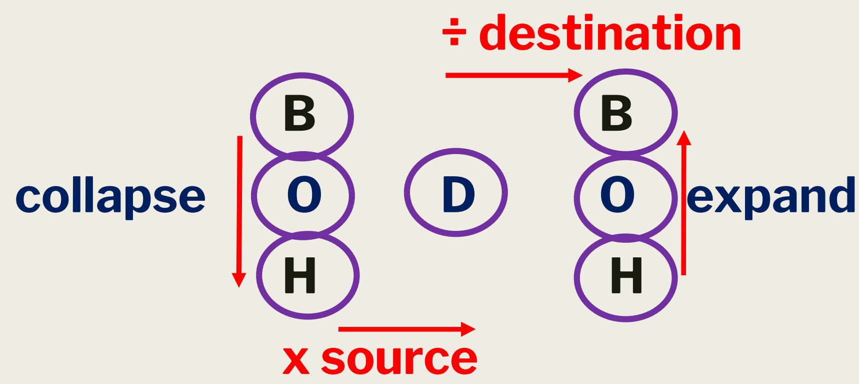
- **Hexadecimal Multiplication**

$$\begin{array}{r} 12_{16} \\ \underline{\times \quad 5_{16}} \\ \underline{5A_{16}} \end{array} \qquad \begin{array}{r} 17_{16} \\ \underline{\times \quad C_{16}} \\ \underline{114_{16}} \end{array} \qquad \begin{array}{r} 17_{16} \\ \underline{\times \quad C_{16}} \\ \underline{12_{5}\ \ 84_{10}} \\ 1\ \ 1\ \ \ 4_{16} \end{array}$$

Chapter Review

Chapter Review

1. **Alphanumeric character data representation**
   - ❑ EBCIDIC Codes
   - ❑ ASCII Codes
2. **Numbering Systems**
   - ❑ Binary
   - ❑ Octal
   - ❑ Decimal
   - ❑ Hexadecimal
3. **Arithmetic conversions**



4. **Arithmetic operations**
   - ❑ Addition
   - ❑ Subtraction
   - ❑ Multiplication