INFORMATION REPRESENTATION AND NUMBER BASE SYSTEMS

LEARNING OBJECTIVES

At the end of this lesson, you should be able to:

- Represent data in binary, decimal, and hexadecimal number systems.
- Convert between the four (4) common number systems.
- Use Binary Coded Decimal (BCD) to represent decimal numbers.
- Carry out basic addition and subtraction in the binary system.
- Represent negative numbers in binary.

WHY INFORMATION REPRESENTATION?

- Understanding various data formats ensures compatibility and efficient data exchange.
- Algorithm efficiency relies on data representation. Choosing the right data structures and formats enhances algorithm performance
- Computer scientists working on visual applications in Graphics and multimedia need to grasp representation intricacies.

WHY INFORMATION REPRESENTATION?

- Secure data transmission and storage rely on robust representation
- Cryptographers need to manipulate data while preserving its integrity.
- As a fundamental skill, it underpins programming, data manipulation, algorithm design, and future adaptability.
- Computer scientists need to bridge the gap between hardware and software, boolean concepts help in understanding how software instructions are executed by hardware.

NUMBER SYSTEMS

Frequently used:

- Decimal Number system
- Binary Number system
- Octal Number system
- Hexadecimal Number system

Other number systems include:

Ternary (base 3), Quaternary (base 4), Quinary (base 5), etc.

DECIMAL NUMBER SYSTEM

It is represented by 10 digits:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Each digit also has a positional value in the decimal system:

Example:	$1_{\text{(thousand)}}$	$4_{\text{(hundred)}}$	$8_{\text{(ten)}}$	$2_{\text{(unit)}}$
Weight:	10^3	10^2	10^1	10^0

BINARY NUMBER SYSTEM

It is represented by 2 digits: **0, 1**

Other numbers are represented as multiple arrangements of 0s and 1s.

Example: $1 \quad 0 \quad 1 \quad 1 \;.\; 0 \quad 1_2$

Weight: $2^3 \quad 2^2 \quad 2^1 \quad 2^0 \quad 2^{-1} \quad 2^{-2}$

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

$$8 + 0 + 2 + 1 + 0 + 0.25 = 11.25_{10}$$

OCTAL NUMBER SYSTEM

It is represented by 8 digits: **0, 1, 2, 3, 4, 5, 6, 7**

Similar to the other number systems, the octal number system possesses weights:

Example:	**5**	**6**	**0_8**
Weight:	**8^2**	**8^1**	**8^0**

$$5 * 8^2 \quad +6 * 8^1 \quad +0 * 8^0$$

$$5 * 64 \quad + \quad 6 * 8 \quad +0 * 1$$

$$320 \quad +48 \quad +0 = 368$$

HEXADECIMAL NUMBER SYSTEMS

It is represented by 9 digits and 6 letters: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F**

Example:	2	0	F	B_{16}
Weight:	16^3	16^2	16^1	16^0
	$2 * 16^3$ +	$0 * 16^2$ +	$F * 16^1$ +	$B * 16^0$
	2*4096 +	0*256 +	15 * 16 +	11 * 1
	8192 +	0 +	240 +	11 = 8443

Converting between Number Systems

Bin2Dec: 1100101_2 to decimal

Binary:	1	1	0	0	1	0	1_2
Weight:	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	64+	32+	0 +	0 +	4 +	0 +	1

Decimal = 101

Dec2Bin: 101 to binary

$$101 = (2 * 50) + 1$$
$$50 = (2 * 25) + 0$$
$$25 = (2 * 12) + 1$$
$$12 = (2 * 6) + 0$$
$$6 = (2 * 3) + 0$$
$$3 = (2 * 1) + 1$$
$$1 = (2 * 0) + 1$$

$$101 \equiv 1100101_2$$

2	**101**	
	50	r 1
	25	r 0
	12	r 1
	6	r 0
	3	r 0
	1	r 1
	0	r 1

Oct2Dec: 2506_8 to decimal

Oct:	2	5	0	6_8
Weight:	8^3	8^2	8^1	8^0
	2*512 +	5*64 +	0*8 +	6*1
	1024 +	320 +	0 +	6

Decimal = 1350

Oct2Bin:	2	5	0	6_8
	010	101	000	110

Binary = $1010100 0110_2$

Dec2Oct: 1350 to Octal

$1350 = (8 * 168) + 6$

$168 = (8 * 21) + 0$

$21 = (8 * 2) + 5$

$2 = (8 * 0) + 2$

$1350 \equiv 2506_8$

8	1350	
	168	r 6
	21	r 0
	2	r 5
	0	r 2

Hex2Dec: $2FA_{16}$ to decimal

Hex:	2	F	A_{16}
Weight:	16^2	16^1	16^0
	2*256 +	15*16 +	10*1
	512 +	240 +	10
	= 762		

Hex2Bin: 2 F A_{16}

0010 1111 1010

$= 1011111010_2$

Dec2Hex: 762 to hex

$762 = (16 * 47) + 10$

$47 = (16 * 2) + 15$

$2 = (16 * 0) + 2$

$762 \equiv 2FA_{16}$

16	762		
	47	r	10
	2	r	15
	0	r	2

Dec	Bin_2	Oct_8	Hex_{16}
1	00001	01	01
2	00010	02	02
3	00011	03	03
4	00100	04	04
5	00101	05	05
6	00110	06	06
7	00111	07	07
8	01000	10	08
9	01001	11	09

Dec	Bin_2	Oct_8	Hex_{16}
10	01010	12	0A
11	01011	13	0B
12	01100	14	0C
13	01101	15	0D
14	01110	16	0E
15	01111	17	0F
16	10000	20	10
17	10001	21	11
18	10010	22	12

QUESTION

1. Convert the following decimal numbers to their binary equivalent:

a) 68 b) 33 c) 112

2. Convert the following binary numbers to their decimal equivalent:

a) 110100_2 b) 10111_2 c) 10101011_2

3. In the binary number system, 347_8 is equivalent to?

4. Complete the table below with the corresponding values

Decimal	$Binary_2$	$Octal_8$	$Hexadecimal_{16}$
89		131	
	110101000111101		6A3D
		2500	540

DATA ENCODING

This describes how different forms of data are represented.

For example:

1. **Binary Coded Decimal (BCD) :** This system expresses each of the decimal digits with its binary equivalent, making it easier to convert between decimal and BCD. However, it is limited to 10 decimal numbers, 0 - 9.

2. **Gray Code:** The gray code is a non-arithmetic coding system, with no specific weights assigned to the bit positions. The distinguishing feature of the Gray code is that it exhibits only a single bit change from one code word to the next.

Decimal	Binary	BCD	Gray Code
00	0000	0000	0000
01	0001	0001	0001
02	0010	0010	0011
03	0011	0011	0010
04	0100	0100	0110
05	0101	0101	0100
06	0110	0110	0101
07	0111	0111	0100
08	1000	1000	1100
09	1001	1001	1101
10	1010	-	1111
11	1011	-	1110
12	1100	-	1010
13	1101	-	1011
14	1110	-	1001
15	1111	-	1000

DATA ENCODING Contd.

3. **American Standard Code for Information Interchange (ASCII):** This is a universally accepted alphanumeric code, used in most electronic equipment. ASCII has 128 characters and symbols represented by an 8-bit binary code with the most significant bit always equal to 0. It is one of the most used character encoding schemes for non-numeric data

4. **Unicode:** The unicode is another character encoding system that provides a unique code point (numeric value) for all characters used in human languages, including various scripts, symbols, and special characters. It utilizes the Universal Character Set (UCS).

Binary Arithmetic

Binary Addition: The principles of binary arithmetic are similar to those of the decimal system. The rules are as follows:

1. $0 + 0 = 0$
2. $0 + 1 = 1 + 0 = 1$
3. $1 + 1 = 0$ with a carry of "1" to the next MORE significant bit

Examples:

1. $10010110_2 + 110011_2$
2. $101110_2 + 1101_2$
3. $1100010_2 + 10101_2$

BINARY ARITHMETIC

Solution:

1. $10010110_2 + 110011_2$

$$\begin{array}{r} 10010110_2 \\ +\quad 110011_2 \\ \hline \mathbf{11001001_2} \end{array}$$

2. $101110_2 + 1101_2$

$$\begin{array}{r} 101110_2 \\ +\quad 1101_2 \\ \hline \mathbf{111011_2} \end{array}$$

3. $1100010_2 + 10101_2$

$$\begin{array}{r} 1100010_2 \\ +\quad 10101_2 \\ \hline \mathbf{1110111_2} \end{array}$$

BINARY ARITHMETIC

Binary Subtraction: Similar to the decimal subtraction, the rules of binary subtraction are as follows:

1. 0 - 0 = 0
2. 1 - 1 = 0
3. 1 - 0 = 1
4. 0 - 1 = 1 with a borrow of "1" from the next MORE significant bit

Examples:

1. $10010110_2 + 110011_2$
2. $101110_2 + 1101_2$
3. $1100010_2 + 10101_2$

BINARY ARITHMETIC

Solution:

1. $10010110_2 - 110011_2$

$$\begin{array}{r} 10010110_2 \\ -\quad 110011_2 \\ \hline \mathbf{01100011_2} \\ \hline \end{array}$$

2. $101110_2 - 1101_2$

$$\begin{array}{r} 101110_2 \\ -\quad 1101_2 \\ \hline \mathbf{100001_2} \\ \hline \end{array}$$

3. $1100010_2 + 10101_2$

$$\begin{array}{r} 1100010_2 \\ -\quad 10101_2 \\ \hline \mathbf{1001101_2} \\ \hline \end{array}$$

NEGATIVE BINARY NUMBERS

Formats used for representing and calculating negative numbers in the binary number system include:

1. **Sign-Bit Magnitude:** In sign-bit magnitude representation, the most significant bit (leftmost bit) is used to indicate the sign of the number, where 0 represents positive and 1 represents negative. The remaining bits represent the magnitude of the number.

   For example:

   +4 in binary: 0100_2 or 00000100_2 (in 8-bit representation)

   -4 in binary: 1100_2 or 10000100_2 (in 8-bit representation)

NEGATIVE BINARY NUMBERS

2. One's complement: In the 1's complement format, the positive numbers remain unchanged, while the negative numbers are obtained by inverting the binary value of the positive number.

For example:

+4 in binary: 00000100_2

-4 in binary: 11111011_2

+43 in binary: 00101011_2

-43 in binary: 11010100_2

NEGATIVE BINARY NUMBERS

3. Two's complement: The 2's complement of a binary number is found by adding 1 to the LSB of the 1's complement.
That is, 2's complement = 1's complement + 1

For example:
+43 in binary: 00101011_2

1's complement (-43): 11010100_2

2's complement (-43): $11010100_2 + 1_2 = 11010101_2$

SUMMARY AND APPLICATIONS

- There are 4 common number base systems; decimal (base 10), binary (base 2), octal (base 8), hexadecimal (base 16).
- Decimal numbers can be represented in several other formats, such as BCD, ASCII, Gray Code, as well as other number systems.
- Arithmetic operations can be carried out in binary number system.
- Negative binary numbers can be represented using one's complement and two's complement.
- One of the advantages of BCD is that it is often used to represent digits in digital clocks and displays due to its direct mapping to decimal digits.
- In cybersecurity, techniques like two's complement and error-correcting codes are used to ensure data integrity and prevent unauthorized alterations during transmission.

REFERENCES

1. Hill, F. J., Peterson, G. R. (2013). *Introduction to Switching Theory and Logic Design* (3rd ed.). John Wiley & Sons Inc.
2. Floyd, T. L. (2013). *Digital Fundamentals: A Systems Approach*. Pearson.
3. Bhaskara, B. (2012). *Switching Theory and Logic Design*. Tata McGraw Hill Publication.
4. Roth, C. H. (2004). *Fundamentals of Logic Design* (5th ed.). Cengage Learning.

Thank You