Number System

- A **number system** is a collection of various symbols which are called digits

- **Two types of number systems are:**
  - Non-positional Number Systems
  - Positional Number Systems

Non-positional Number System

- **Characteristics**
  - Use symbols such as I for 1, II for 2, III for 3 etc
  - Each symbol represents the same value regardless of its position in the number
  - The symbols are simply added to find out the value of a particular number

- **Difficulty**
  - It is difficult to perform arithmetic with such a number system

Positional Number System

- **Characteristics**
  - Use only a few symbols called digits
  - These symbols represent different values depending on the position they occupy in the number
  - The value of each digit is determined by:
    1. The digit itself
    2. The position of the digit in the number
    3. **The base of the number system** (Base = total number of digits in the number system)
  - The maximum value of a single digit is always equal to one less than the value of the base

Positional Number System

- There are four Positional Number Systems: Binary, Decimal, Octal and Hexadecimal

Binary	Decimal	Octal	Hexadecimal
0000	00	0	0
0001	01	1	1
0010	02	2	2
0011	03	3	3
0100	04	4	4
0101	05	5	5
0110	06	6	6
0111	07	7	7
1000	08	10	8
1001	09	11	9
1010	10	12	A
1011	11	13	B
1100	12	14	C
1101	13	15	D
1110	14	16	E
1111	15	17	F

Binary Number System

- **Characteristics:**
  - Has only 2 symbols or digits (0 and 1). Hence its base = 2
  - The maximum value of a single digit is 1 (one less than the value of the base)
  - **Each position of a digit represents a specific power of the base (2)**
  - This number system is used in computers

Example

$$10101_2 = (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) \times (1 \times 2^0)$$

$$= 16 + 0 + 4 + 0 + 1$$

$$= 21_{10}$$

Decimal Number System

- **Characteristics:**
  - Has 10 symbols or digits (0, 1, 2, 3, 4, 5, 6, 7,8, 9). Hence, its base = 10
  - The maximum value of a single digit is 9 (one less than the value of the base)
  - Each position of a digit represents a specific power of the base (10)
  - We use this number system in our day-to-day life

Example

$$2586_{10} = (2 \times 10^3) + (5 \times 10^2) + (8 \times 10^1) + (6 \times 10^0)$$

$$= 2000 + 500 + 80 + 6$$

Octal Number System

- **Characteristics:**

  - Has total 8 symbols or digits (0, 1, 2, 3, 4, 5, 6, 7). Hence, its base = 8
  - The maximum value of a single digit is 7 (one less than the value of the base)
  - Each position of a digit represents a specific power of the base (8)
  - Since there are only 8 digits, 3 bits ($2^3 = 8$) are sufficient to represent any octal number in binary

Example

$$2057_8 = (2 \times 8^3) + (0 \times 8^2) + (5 \times 8^1) + (7 \times 8^0)$$

$$= 1024 + 0 + 40 + 7$$

$$= 1071_{10}$$

Hexadecimal Number System

- **Characteristics:**
  - Has total 16 symbols or digits (0, 1, 2, 3, 4, 5, 6, 7,8, 9, A, B, C, D, E, F). Hence its base = 16
  - The symbols A, B, C, D, E and F represent the decimal values 10, 11, 12, 13, 14 and 15 respectively
  - The maximum value of a single digit is 15 (one less than the value of the base)
  - Each position of a digit represents a specific power of the base (16)
  - Since there are only 16 digits, 4 bits ($2^4 = 16$) are sufficient to represent any hexadecimal number in binary

Example

$$
\begin{aligned}
1AF_{16} &= (1 \times 16^2) + (A \times 16^1) + (F \times 16^0) \\
&= 1 \times 256 + 10 \times 16 + 15 \times 1 \\
&= 256 + 160 + 15 \\
&= 431_{10}
\end{aligned}
$$

Conversions

- **Decimal to Binary**
- Here is an example of using repeated division to convert 1792 decimal to binary:

Decimal Number	Operation	Quotient	Remainder
1792	÷ 2 =	896	0
896	÷ 2 =	448	0
448	÷ 2 =	224	0
224	÷ 2 =	112	0
112	÷ 2 =	56	0
56	÷ 2 =	28	0
28	÷ 2 =	14	0
14	÷ 2 =	7	0
7	÷ 2 =	3	1
3	÷ 2 =	1	1
1	÷ 2 =	0	1
0	done.		

- Reverse the remainders, we get 11100000000
- $(1792)_{10} = (11100000000)_2$

Conversions

- **Decimal to Octal**
- Here is an example of using repeated division to convert 1792 decimal to octal:

Decimal Number	Operation	Quotient	Remainder
1792	÷ 8 =	224	0
224	÷ 8 =	28	0
28	÷ 8 =	3	4
3	÷ 8 =	0	3
0	done.		

- Reverse the remainders, we get 3400
- $(1792)_{10} = (3400)_8$

Conversions

- **Decimal to Hexadecimal**
- Here is an example of using repeated division to convert 1792 decimal to hexadecimal:

Decimal Number	Operation	Quotient	Remainder
1792	÷ 16 =	112	0
112	÷ 16 =	7	0
7	÷ 16 =	0	7
0	done.		

- Reverse the remainders, we get 700
- $(1792)_{10} = (700)_{16}$

Conversions

- The only addition to the algorithm when converting from decimal to hexadecimal is that a table must be used to obtain the hexadecimal digit if the remainder is greater than decimal 9.

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

- For example, 590 decimal converted to hex is:

Decimal Number	Operation	Quotient	Remainder	Hexadecimal Result
590	÷ 16 =	36	14	E
36	÷ 16 =	2	4	4
2	÷ 16 =	0	2	2
0	done.			

- Reverse the remainders, we get 24E
- $(590)_{10} = (24E)_{16}$

Conversion from other to decimal number system

Example

$10101_2 = (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) \times (1 \times 2^0)$

$= 16 + 0 + 4 + 0 + 1$

$= 21_{10}$

Example

$2057_8 = (2 \times 8^3) + (0 \times 8^2) + (5 \times 8^1) + (7 \times 8^0)$

$= 1024 + 0 + 40 + 7$

$= 1071_{10}$

Example

$1AF_{16} = (1 \times 16^2) + (A \times 16^1) + (F \times 16^0)$

$= 1 \times 256 + 10 \times 16 + 15 \times 1$

$= 256 + 160 + 15$

$= 431_{10}$

Conversion from Binary to Octal

- **Step 1** – Divide the binary digits into groups of three (starting from the right).
- **Step 2** – Convert each group of three binary digits to one octal digit.
- Example

Binary Number – 10101_2

Calculating Octal Equivalent –

Step	Binary Number	Octal Number
1.	10101_2	010 101
2.	10101_2	2_8 5_8
3.	10101_2	25_8

Binary Number – 10101_2 = Octal Number – 25_8

Conversion from Octal to Binary

- **Step 1** – Convert each octal digit to a 3 digit binary number (the octal digits may be treated as decimal for this conversion).
- **Step 2** – Combine all the resulting binary groups (of 3 digits each) into a single binary number.
- Example

Binary Number – 10101_2

Calculating Octal Equivalent –

Step	Octal Number	Binary Number
1.	25_8	2_{10} 5_{10}
2.	25_8	010_2 101_2
3.	25_8	010101_2

Octal Number – 25_8 = Binary Number – 10101_2

Conversion from Binary to Hexadecimal

- **Step 1** – Divide the binary digits into groups of four (starting from the right).
- **Step 2** – Convert each group of four binary digits to one hexadecimal symbol.
- Example

Binary Number – 10101_2

Calculating Octal Equivalent –

Step	Binary Number	Hexadecimal Number
1.	10101_2	0001 0101
2.	10101_2	1_{10} 5_{10}
3.	10101_2	15_{16}

Binary Number – 10101_2 = Hexadecimal Number – 15_{16}

Conversion from Hexadecimal to Binary

- **Step 1** – Convert each hexadecimal digit to a 4 digit binary number (the hexadecimal digits may be treated as decimal for this conversion).
- **Step 2** – Combine all the resulting binary groups (of 4 digits each) into a single binary number.
- Example

Binary Number – 10101_2

Calculating Octal Equivalent –

Step	Hexadecimal Number	Binary Number
1.	15_{16}	$1_{10}\ 5_{10}$
2.	15_{16}	$0001_2\ 0101_2$
3.	15_{16}	00010101_2

Hexadecimal Number – 15_{16} = Binary Number – 10101_2

Octal to Hexadecimal

- When converting from octal to hexadecimal, it is often easier to first convert the octal number into binary and then from binary into hexadecimal.
- For example, to convert 345 octal into hex:*(from the previous example)*
- Octal =345

  Binary =011 100 101

Drop any leading zeros or pad with leading zeros to get groups of four binary digits (bits):
Binary 011100101 = 1110 0101

Then, look up the groups in a table to convert to hexadecimal digits.

- Binary =1110 0101 Hexadecimal =E5= E5 hex

Binary	Decimal	Octal	Hexadecimal
0000	00	0	0
0001	01	1	1
0010	02	2	2
0011	03	3	3
0100	04	4	4
0101	05	5	5
0110	06	6	6
0111	07	7	7
1000	08	10	8
1001	09	11	9
1010	10	12	A
1011	11	13	B
1100	12	14	C
1101	13	15	D
1110	14	16	E
1111	15	17	F

Binary	Decimal	Octal	Hexadecimal
0000	00	0	0
0001	01	1	1
0010	02	2	2
0011	03	3	3
0100	04	4	4
0101	05	5	5
0110	06	6	6
0111	07	7	7
1000	08	10	8
1001	09	11	9
1010	10	12	A
1011	11	13	B
1100	12	14	C
1101	13	15	D
1110	14	16	E
1111	15	17	F

Hexadecimal to Octal

- When converting from hexadecimal to octal, it is often easier to first convert the hexadecimal number into binary and then from binary into octal.
- For example, to convert A2DE hex into octal:
- Hexadecimal =**A 2 D E**
- **Binary** =1010 0010 1101 1110 = 1010001011011110
- **Add leading zeros or remove leading zeros to group into sets of three binary digits.**
- Binary: 1010001011011110 = 001 010 001 011 011 110
- Then, look up each group in a table:

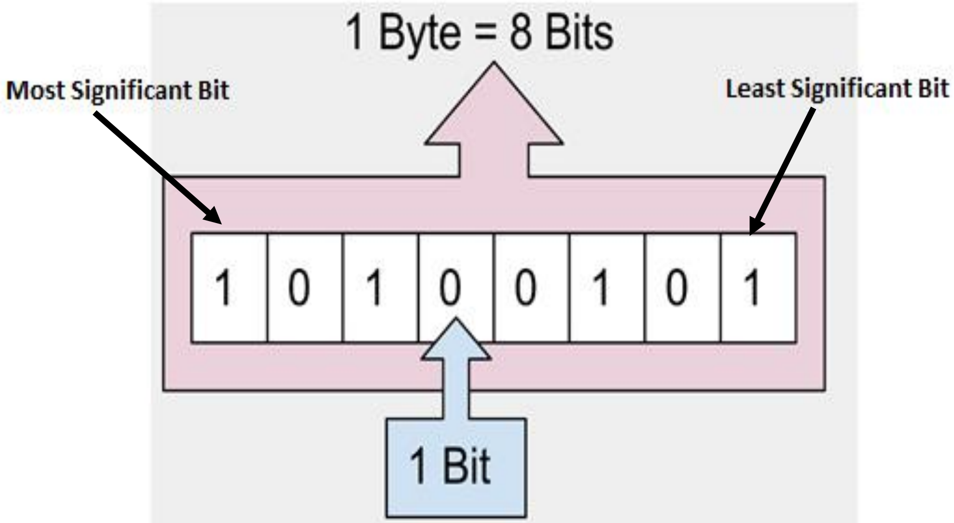
Binary	000	001	010	011	100	101	110	111
Octal	0	1	2	3	4	5	6	7

Binary =001010001011011110 Octal =121336

Data Representation

- Computer uses a fixed number of bits to represent a piece of data, which could be a number, a character, or others
- A n-bit storage location can represent up to 2^n distinct entities
- For example, a 3-bit memory location can hold one of these eight binary patterns: 000, 001, 010, 011, 100, 101, 110, or 111
- Hence, it can represent at most 8 distinct entities. You could use them to represent:
  - Numbers 0 to 7
  - Characters 'A' to 'H‘
  - 8 kinds of fruits like apple, orange, banana
  - or 8 kinds of animals like lion, tiger, etc.

Data Representation



1 byte	= 8 bits
1 kilobyte	= 1024 bytes
1 megabyte	= 1024 kilobyte
1 gigabyte	= 1024 megabyte
1 terabyte	= 1024 gigabyte

Multiples of Bytes

Unit (Symbol)	Value (SI)	Value (Binary)
Kilobyte (kB)	10^3	2^{10}
Megabyte (MB)	10^6	2^{20}
Gigabyte (GB)	10^9	2^{30}
Terabyte (TB)	10^{12}	2^{40}
Petabyte (PB)	10^{15}	2^{50}
Exabyte (EB)	10^{18}	2^{60}
Zettabyte (ZB)	10^{21}	2^{70}
Yottabyte (YB)	10^{24}	2^{80}

Memory Units

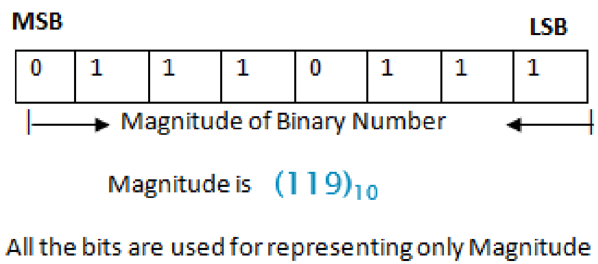
Data Representation: Signed and Unsigned

- Integers, for example, can be represented in 8-bit, 16-bit, 32-bit or 64-bit

- Besides bit-lengths, there are two representation schemes for integers:
  1. ***Unsigned Integers***: can represent zero and positive integers
  2. ***Signed Integers***: can represent zero, positive and negative integers

Data Representation: Unsigned

- Unsigned integers can represent zero and positive integers, but not negative integers.
- The value of an unsigned integer is interpreted as "the magnitude of its underlying binary pattern".



Example 1:
Suppose n=8 and

binary pattern is **0100 0001 (Binary)**

Value of this unsigned integer is:

1×2^0 + 1×2^6 = 65 (Decimal)

Example 2:
Suppose n=16 and

binary pattern is **0001 0000 0000 1000 (Binary)**

Value of this unsigned integer is:

1×2^3 + 1×2^12 = 4104 (Decimal)

Data Representation: Unsigned

- An n-bit pattern can represent 2^n distinct integers. An n-bit unsigned integer can represent integers from 0 to (2^n)-1, as tabulated below:

n	Minimum	Maximum
8	0	(2^8)-1 (=255)
16	0	(2^16)-1 (=65,535)
32	0	(2^32)-1 (=4,294,967,295)
64	0	(2^64)-1 (=18,446,744,073,709,551,615)

- An 8-bit unsigned integer has a range of 0 to 255

- Maximum value = 255

128+64+32+16+8+4+2+1 = 255

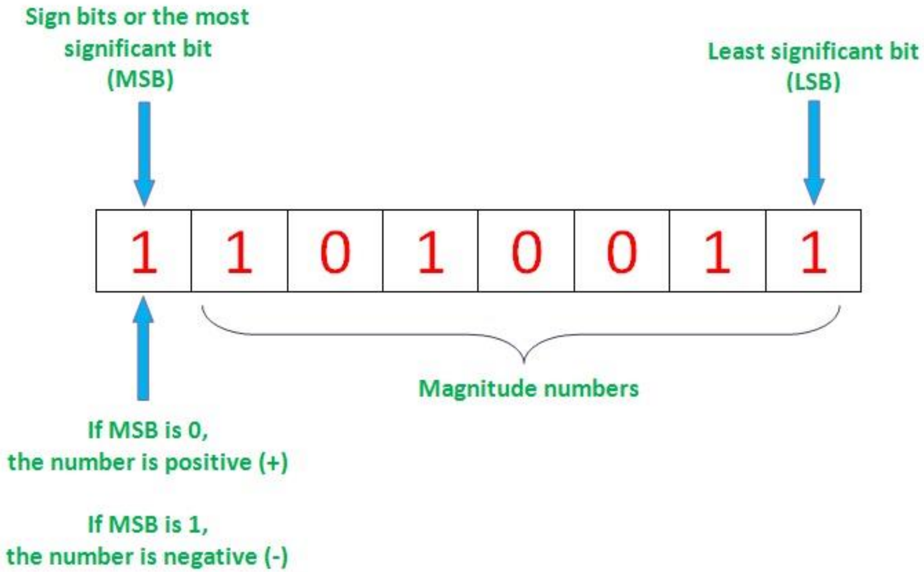
128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1

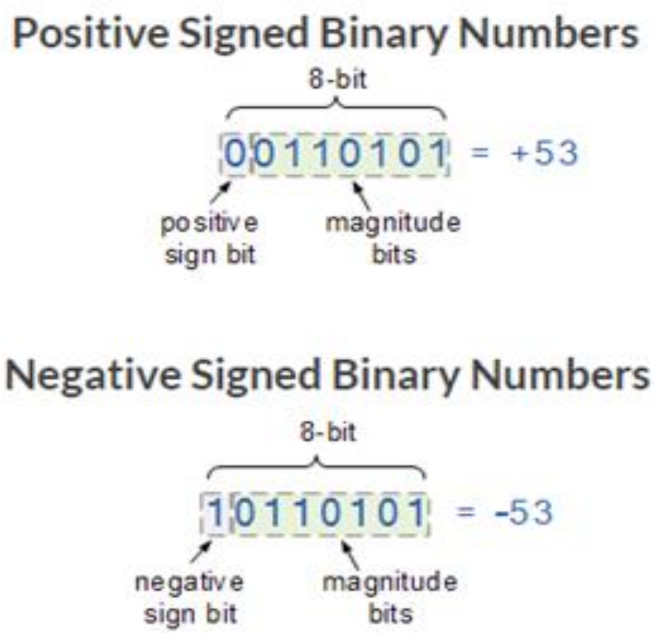
- Minimum value = 0

128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0

Data Representation: Signed

- The Most-significant Bit (MSB) is the sign bit, with value of 0 representing positive integer and 1 representing negative integer.
- The remaining n-1 bits represents the magnitude (absolute value) of the integer.
- The absolute value of the integer is interpreted as "the magnitude of the (n-1)-bit binary pattern".





Data Representation: Signed

Example 1:

Suppose n=8 and
binary representation is **0 100 0001B**

Sign bit is 0 ⇒ positive

Absolute value is 100 0001B = 65D

Hence, the integer is +65D

Example 2:

Suppose n=8 and
binary representation is **1 000 0001B**

Sign bit is 1 ⇒ negative

Absolute value is 000 0001B = 1D

Hence, the integer is -1D

Example 3:

Suppose n=8 and
binary representation is **0 000 0000B**

Sign bit is 0 ⇒ positive

Absolute value is 000 0000B = 0D

Hence, the integer is +0D

Example 4:

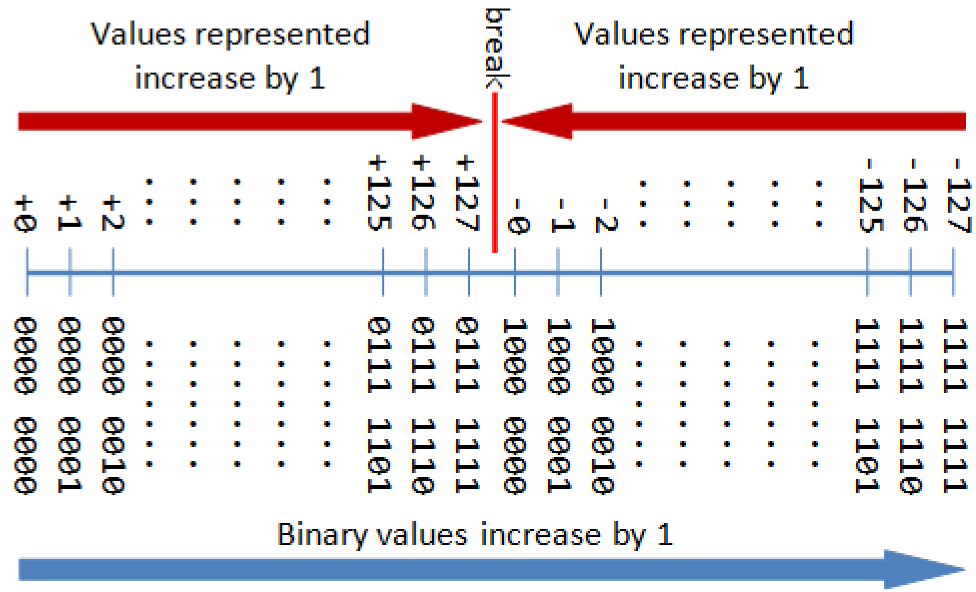
Suppose n=8 and
binary representation is **1 000 0000B**

Sign bit is 1 ⇒ negative

Absolute value is 000 0000B = 0D

Hence, the integer is -0D

Data Representation: Signed



Sign-Magnitude Representation

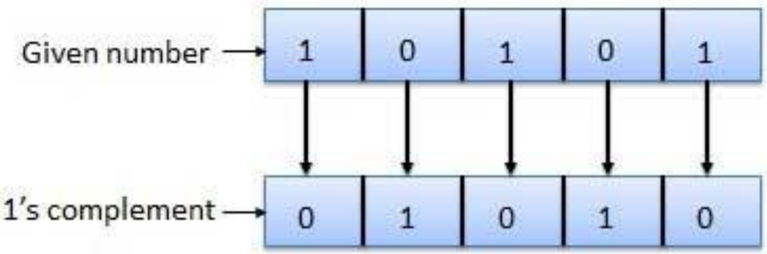
The drawbacks of sign-magnitude representation are:

1. There are two representations (0000 0000B and 1000 0000B) for the number zero, which could lead to inefficiency and confusion.

1. Positive and negative integers need to be processed separately.

Complement of a number

- Complements are used in the digital computers in order to simplify the subtraction operation and for the logical manipulations
- **Binary system complements**
  - As the binary system has base r = 2. So the two types of complements for the binary system are 2's complement and 1's complement
- **1's complement**
  - The 1's complement of a number is found by changing all 1's to 0's and all 0's to 1's. Example of 1's Complement is as follows:



Complement of a number

- **2's complement**
  - The 2's complement of binary number is obtained by adding 1 to the Least Significant Bit (LSB) of 1's complement of the number.
  - 2's complement = 1's complement + 1
  - Example of 2's Complement is as follows: