Unit 4: Data Representation

Introduction:

The data stored in the computer may be of different kinds, as follows—

- Numeric data (0, 1, 2, …, 9)
- Alphabetic data (A, B, C, …, Z)
- Alphanumeric data—Combination of any of the symbols— (A, B, C… Z), (0, 1… 9), or special characters (+, −, Blank), etc.

All kinds of data, be it alphabets, numbers, symbols, sound data or video data, is represented in terms of 0s and 1s, in the computer. Each symbol is represented as a unique combination of 0s and 1s.

Number System:

A number system in base r or radix r uses unique symbols for r digits. One or more digits are combined to get a number. The base of the number decides the valid digits that are used to make a number. In a number, the position of digit starts from the right-hand side of the number. The rightmost digit has position 0, the next digit on its left has position 1, and so on. The digits of a number have two kinds of values—

- Face value, and
- Position value.

The face value of a digit is the digit located at that position. For example, in decimal number 52, face value at position 0 is 2 and face value at position 1 is 5. The position value of a digit is (base position). For example, in decimal number 52, the position value of digit 2 is 100 and the position value of digit 5 is 101. Decimal numbers have a base of 10.

The number is calculated as the sum of, face value * base position, of each of the digits. For decimal number 52, the number is $5*10^1 + 2*10^0 = 50 + 2 = 52$.

In computers, we are concerned with four kinds of number systems, as follows—

- Decimal Number System —Base 10
- Binary Number System —Base 2
- Octal Number System —Base 8
- Hexadecimal Number System—Base 16

The numbers given as input to computer and the numbers given as output from the computer, are generally in decimal number system, and are most easily understood by humans. However, computer understands the binary number system, i.e., numbers in terms of 0s and 1s. The binary data is also represented, internally, as octal numbers and hexadecimal numbers due to their ease of use.

A number in a particular base is written as $(number)_{base}$ of number. For example, $(23)_{10}$ means that the number 23 is a decimal number, and $(345)_8$ shows that 345 is an octal number.

Decimal Number System

- It consists of 10 digits—0, 1, 2, 3, 4, 5, 6, 7, 8 and 9.

- All numbers in this number system are represented as combination of digits 0—9. For example, 34, 5965 and 867321.
- The position value and quantity of a digit at different positions in a number are as follows:

Position:	3	2	1	0	.	–1	–2	–3
Position Value:	10^3	10^2	10^1	10^0		10^{-1}	10^{-2}	10^{-3}
Quantity:	1000	100	10	1		1/10	1/100	1/1000

Binary Number System

- The binary number system consists of two digits—0 and 1.
- All binary numbers are formed using combination of 0 and 1. For example, 1001, 11000011 and 10110101.
- The position value and quantity of a digit at different positions in a number are as follows:

Position:	3	2	1	0	.	–1	–2	–3
Position Value:	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
Quantity:	8	4	2	1		1/2	1/4	1/8

Octal Number System

- The octal number system consists of eight digits—0 to 7.
- All octal numbers are represented using these eight digits. For example, 273, 103, 2375, etc.
- The position value and quantity of a digit at different positions in a number are as follows:

Position:	3	2	1	0	.	–1	–2	–3
Position Value:	8^3	8^2	8^1	8^0		8^{-1}	8^{-2}	8^{-3}
Quantity:	512	64	8	1		1/8	1/64	1/512

Hexadecimal Number System

- The hexadecimal number system consists of sixteen digits—0 to 9, A, B, C, D, E, F, where (A is for 10, B is for 11, C-12, D-13, E-14, F-15).
- All hexadecimal numbers are represented using these 16 digits. For example, 3FA, 87B, 113, etc.
- The position value and quantity of a digit at different positions in a number are as follows:

Position:	3	2	1	0	.	–1	–2	–3
Position Value:	16^3	16^2	16^1	16^0		16^{-1}	16^{-2}	16^{-3}
Quantity:	4096	256	16	1		1/16	1/256	1/4096

The summary of number system is as follows:

	Base	Digits	Largest Digit
Decimal	10	0–9	9
Binary	2	0,1	1
Octal	8	0–7	7
Hexadecimal	16	0–9, A, B, C, D, E, F	F (15)

Fig: Summary of number system

Decimal	Binary	Octal	Hexadecimal
0	0000	000	0
1	0001	001	1
2	0010	002	2
3	0011	003	3
4	0100	004	4
5	0101	005	5
6	0110	006	6
7	0111	007	7
8	1000	010	8
9	1001	011	9
10	1010	012	A
11	1011	013	B
12	1100	014	C
13	1101	015	D
14	1110	016	E
15	1111	017	F
16	10000	020	10

Fig: Decimal, binary, octal and hexadecimal equivalents

Conversion from Decimal to Binary, Octal, Hexadecimal:

A decimal number has two parts—integer part and fraction part. For example, in the decimal number 23.0786, 23 is the integer part and .0786 is the fraction part. The method used for the conversion of the integer part of a decimal number is different from the one used for the fraction part.

<u>Conversion from Decimal integer to Binary:</u>

A decimal integer is converted to any other base, by using the division operation.
To convert a decimal integer to:

- binary-divide by 2,
- octal-divide by 8, and,
- hexadecimal-divide by 16.

<u>Example 1:</u> Convert 25 from Base 10 to Base 2.

Divide the number with toBase. After each division, write the remainder on right-side column and quotient in the next line in the middle column. Continue dividing till the quotient is 0.

to Base	Number (Quotient)	Remainder
2	25	
2	12	1
2	6	0
2	3	0
2	1	1
	0	1

Write the digits in remainder column starting from downwards to upwards,

to Base	Number (Quotient)	Remainder
2	25	
2	12	1
2	6	0
2	3	0
2	1	1
	0	1

The binary equivalent of number $(25)_{10}$ is $(11001)_2$

Example 2: Convert 23 from Base 10 to Base 2, 8, 16.

to Base	Number (Quotient)	Remainder
2	23	
2	11	1
2	5	1
2	2	1
2	1	0
	0	1

The binary equivalent of $(23)_{10}$ is $(10111)_2$

to Base	Number (Quotient)	Remainder
8	23	
8	2	7
	0	2

The octal equivalent of $(23)_{10}$ is $(27)_8$

to Base	Number (Quotient)	Remainder
16	23	
16	1	7
	0	1

The hexadecimal equivalent of $(23)_{10}$ is $(17)_{16}$

Example 3: Convert 147 from Base 10 to Base 2, 8 and 16.

to Base	Number (Quotient)	Remainder
2	147	
2	73	1
2	36	1
2	18	0
2	9	0
2	4	1
2	2	0
2	1	0
	0	1

The binary equivalent of $(147)_{10}$ is $(10010011)_2$

to Base	Number (Quotient)	Remainder
8	147	
8	18	3
8	2	2
	0	2

The octal equivalent of $(147)_{10}$ is $(223)_8$

to Base	Number (Quotient)	Remainder
16	147	
16	9	3
	0	9

The hexadecimal equivalent of $(147)_{10}$ is $(93)_{16}$

Example 4: Convert 94 from Base 10 to Base 2, 8 and 16.

to Base	Number (Quotient)	Remainder
2	94	
2	47	0
2	23	1
2	11	1
2	5	1
2	2	1
2	1	0
	0	1

The binary equivalent of $(94)_{10}$ is $(1011110)_2$

to Base	Number (Quotient)	Remainder
8	94	
8	11	6
8	1	3
	0	1

The octal equivalent of $(94)_{10}$ is $(136)_8$

to Base	Number (Quotient)	Remainder
16	94	
16	5	14
	0	5

The number 14 in hexadecimal is E.

The hexadecimal equivalent of $(94)_{10}$ is $(5E)_{16}$

Converting Decimal Fraction to Binary, Octal, Hexadecimal

A fractional number is a number less than 1. It may be .5, .00453, .564, etc. We use the multiplication operation to convert decimal fraction to any other base.

To convert a decimal fraction to—

- binary: multiply by 2,
- octal: multiply by 8, and,
- hexadecimal: multiply by 16.

Steps for conversion of a decimal fraction to any other base are:

1. Multiply the fractional number with the toBase, to get a resulting number.
2. The resulting number has two parts, non-fractional part and fractional part.
3. Record the non-fractional part of the resulting number.
4. Repeat the above steps at least four times.
5. Write the digits in the non-fractional part starting from upwards to downwards.

Example: Convert 0.2345 from Base 10 to Base 2.

0.2345
x 2
0.4690

.4690
x 2
0.9380

.9380
x 2
1.8760

.8760
x 2
1.7520

.7520
x 2
1.5040

.5040
x 2
1.0080

The binary equivalent of $(0.2345)_{10}$ is $(0.001111)_2$

Example: Convert 0.865 from Base 10 to Base 2, 8 and 16.

Base 2	Base 8	Base 16
0.865	0.865	0.865
x 2	x 8	x 16
1.730	6.920	5190
x 2	x 8	865 x
1.460	7.360	13.840
x 2	x 8	x 16
0.920	2.880	5040
x 2	x 8	840 x
1.840	7.040	13.440
x 2		x 16
1.680		2640
x 2		440 x
1.360		7.040

The binary equivalent of $(.865)_{10}$ is $(.110111)_2$

The octal equivalent of $(0.865)_{10}$ is $(.6727)_8$

The number 13 in hexadecimal is D.

The hexadecimal equivalent of $(0.865)_{10}$ is $(.DD7)_{16}$

Converting Decimal Integer.Fraction to Binary, Octal, Hexadecimal

A decimal integer.fraction number has both integer part and fraction part. The steps for conversion of a decimal integer.fraction to any other base are:

- Convert decimal integer part to the desired base
- Convert decimal fraction part to the desired base
- The integer and fraction part in the desired base is combined to get integer.fraction.

Example: Convert 34.4674 from Base 10 to Base 2.

to Base	Number (Quotient)	Remainder
2	34	
2	17	0
2	8	1
2	4	0
2	2	0
2	1	0
	0	1

The binary equivalent of $(34)_{10}$ is $(100010)_2$

0.4674
x 2
0.9348
x 2
1.8696
x 2
1.7392
x 2
1.4784
x 2
0.9568
x 2
1.8136

The binary equivalent of $(0.4674)_{10}$ is $(.011101)_2$

The binary equivalent of $(34.4674)_{10}$ is $(100010.011101)_2$

Example: Convert 34.4674 from Base 10 to Base 8.

to Base	Number (Quotient)	Remainder
8	34	
8	4	2
	0	4

The octal equivalent of $(34)_{10}$ is $(42)_8$

```
 0.4674
    x 8
 3.7392
    x 8
 5.9136
    x 8
 7.3088
    x 8
 2.4704
```

The octal equivalent of $(0.4674)_{10}$ is $(.3572)_8$

The octal equivalent of $(34.4674)_{10}$ is $(42.3572)_8$

Example: Convert 34.4674 from Base 10 to Base 16.

to Base	Number (Quotient)	Remainder
16	34	
16	4	2
	0	2

The hexadecimal equivalent of $(34)_{10}$ is $(22)_{16}$

```
  0.4674
    x 16
  28044
  4674x
  9.4784
    x 16
  28704
   4784x
  7.6544
    x 16
  39264
  6544x
 10.4904
    x 16
  29424
  4904x
  7.8464
```

The hexadecimal equivalent of $(0.4674)_{10}$ is $(.97A7)_{16}$

The hexadecimal equivalent of $(34.4674)_{10}$ is $(22.97A7)_{16}$

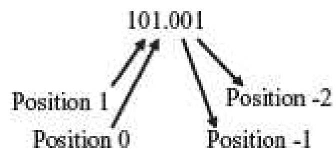
Conversion of Binary, Octal, Hexadecimal to Decimal:

A binary, octal or hexadecimal number has two parts—integer part and fraction part. For example, a binary number could be 10011, 0.011001 or 10011.0111. The numbers 45, .362 or 245.362 are octal numbers. A hexadecimal number could be A2, .4C2 or A1.34.

The method used for the conversion of integer part and fraction part of binary, octal or hexadecimal number to decimal number is the same; multiplication operation is used for the conversion. The conversion mechanism uses the face value and position value of digits. The steps for conversion are as follows:

1. Find the sum of the Face Value * $(\text{fromBase})^{\text{position}}$ for each digit in the number.
    - In a non-fractional number, the rightmost digit has position 0 and the position increases as we go towards the left.

- In a fractional number, the first digit to the left of decimal point has position 0 and the position increases as we go towards the left. The first digit to the right of the decimal point has position –1 and it decreases as we go towards the right (−2, −3, etc.)



Example: Convert 1011 from Base 2 to Base 10.
Convert 62 from Base 8 to Base 10.
Convert C15 from Base 16 to Base 10.

1011 fromBase 2 toBase 10	62 fromBase 8 toBase 10	C15 fromBase 16 toBase 10
$1011 = 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0$ $= 1*8 + 0*4 + 1*2 + 1*1$ $= 8 + 0 + 2 + 1$ $= 11$ The decimal equivalent of $(1011)_2$ is 11.	$62 = 6*8^1 + 2*8^0$ $= 6*8 + 2*1$ $= 48 + 2$ $= 50$ The decimal equivalent of $(62)_8$ is 50.	$C15 = C*16^2 + 1*16^1 + 5*16^0$ $= 12*256 + 1*16 + 5*1$ $= 3072 + 16 + 5$ $= 3093$ The decimal equivalent of $(C15)_{16}$ is 3093

Example: Convert .1101 from Base 2 to Base 10.
Convert .345 from Base 8 to Base 10.
Convert .15 from Base 16 to Base 10.

.1101 fromBase 2 toBase 10	.345 fromBase 8 toBase 10	.15 fromBase 16 toBase 10
$.1101 = 1*2^{-1} + 1*2^{-2} + 0*2^{-3} + 1*2^{-4}$ $= 1/2 + 1/4 + 0 + 1/16$ $= 13/16$ $= .8125$ The decimal equivalent of $(.1101)_2$ is .8125	$.345 = 3*8^{-1} + 4*8^{-2} + 5*8^{-3}$ $= 3/8 + 4/64 + 5/512$ $= 229/512$ $= .447$ The decimal equivalent of $(.345)_8$ is .447	$.15 = 1*16^{-1} + 5*16^{-2}$ $= 1/16 + 5/256$ $= 21/256$ $= .082$ The decimal equivalent of $(.15)_{16}$ is .082

Example: Convert 1011.1001 from Base 2 to Base 10.
Convert 24.36 from Base 8 to Base 10.
Convert 4D.21 from Base 16 to Base 10.

1011.1001 fromBase 2 toBase 10	24.36 fromBase 8 toBase 10	4D.21 fromBase 16 toBase 10
$1011.1001 = 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 + 1*2^{-1} + 0*2^{-2} + 0*2^{-3} + 1*2^{-4}$ $= 8 + 0 + 2 + 1 + 1/2 + 0 + 0 + 1/16$ $= 11 + 9/16$ $= 11.5625$ The decimal equivalent of $(1011.1001)_2$ is 11.5625	$24.36 = 2*8^1 + 4*8^0 + 3*8^{-1} + 6*8^{-2}$ $= 16 + 4 + 3/8 + 6/64$ $= 20 + 30/64$ $= 20.4687$ The decimal equivalent of $(24.36)_8$ is 20.4687	$4D.21 = 4*16^1 + D*16^0 + 2*16^{-1} + 1*16^{-2}$ $= 64 + 13 + 2/16 + 1/256$ $= 77 + 33/256$ $= 77.1289$ The decimal equivalent of $(4D.21)_{16}$ is 77.1289

Conversion of Binary to Octal, Hexadecimal:

A binary number can be converted into octal or hexadecimal number using a shortcut method. The shortcut method is based on the following information—

- An octal digit from 0 to 7 can be represented as a combination of 3 bits, since $2^3 = 8$.

- A hexadecimal digit from 0 to 15 can be represented as a combination of 4 bits, since $2^4 = 16$.

The Steps for Binary to Octal Conversion are—

- Partition the binary number in groups of three bits, starting from the right-most side.
- For each group of three bits, find its octal number.
- The result is the number formed by the combination of the octal numbers.

The Steps for Binary to Hexadecimal Conversion are—

- Partition the binary number in groups of four bits, starting from the right-most side.
- For each group of four bits, find its hexadecimal number.
- The result is the number formed by the combination of the hexadecimal numbers.

Example: Convert the binary number 1110101100110 to octal.
Given binary number 1110101100110

- Partition binary number in groups of three bits, starting from the right-most side.

  1 110 101 100 110

- For each group find its octal number.

  1 110 101 100 110
  1 6 5 4 6

- The octal number is 16546.

Example: Convert the binary number 1110101100110 to hexadecimal.
Given binary number 1110101100110

- Partition binary number in groups of four bits, starting from the right-most side.

  1 1101 0110 0110

- For each group find its hexadecimal number.

  1 1101 0110 0110
  1 D 6 6

- The hexadecimal number is 1D66.

Conversion of Octal, Hexadecimal to Binary:

The conversion of a number from octal and hexadecimal to binary uses the inverse of the steps defined for the conversion of binary to octal and hexadecimal.

The Steps for Octal to Binary Conversion are—

- Convert each octal number into a three-digit binary number.
- The result is the number formed by the combination of all the bits.

The Steps for Hexadecimal to Binary Conversion are—

- Convert each hexadecimal number into a four-digit binary number.
- The result is the number formed by the combination of all the bits.

Example: Convert the hexadecimal number 2BA3 to binary.

- Given number is 2BA3
- Convert each hexadecimal digit into four-digit binary number.

  2 B A 3
  0010 1011 1010 0011

- Combine all the bits to get the result 0010101110100011.

Example 15: Convert the octal number 473 to binary.

- Given number is 473

- Convert each octal digit into three-digit binary number.

4	7	3
100	111	011

- Combine all the bits to get the result 100111011.

Binary Arithmetic:

The arithmetic operations—addition, subtraction, multiplication and division, performed on the binary numbers is called binary arithmetic. In computer, the basic arithmetic operations performed on the binary numbers is—

- Binary addition, and
- Binary subtraction.

Binary Addition

Binary addition involves addition of two or more binary numbers. The binary addition rules are used while performing the binary addition. The rules are as follows:

Input 1	Input 2		Sum	Carry
0	0	→	0	No carry
0	1	→	1	No carry
1	0	→	1	No carry
1	1	→	0	1

Table: Binary addition rules

Binary addition of three inputs follows the rule shown below:

Input 1	Input 2	Input 3		Sum	Carry
0	0	0	→	0	No Carry
0	0	1	→	1	No Carry
0	1	0	→	1	No Carry
0	1	1	→	0	1
1	0	0	→	1	No Carry
1	0	1	→	0	1
1	1	0	→	0	1
1	1	1	→	1	1

Table: Binary addition of three inputs

Addition of the binary numbers involves the following steps—

1. Start addition by adding the bits in unit column (the right-most column). Use the rules of binary addition.
2. The result of adding bits of a column is a sum with or without a carry.
3. Write the sum in the result of that column.
4. If a carry is present, the carry is carried-over to the addition of the next left column.
5. Repeat steps 2–4 for each column, i.e., the tens column, hundreds column and so on.

Example 1: Add 10 and 01. Verify the answer with the help of decimal addition.

	Binary Addition	Decimal Addition
	1 0	2
	+ 0 1	+ 1
Result	1 1	3

$11_2 = 3_{10}$

Example 2: Add 01 and 11. Verify the answer with the help of decimal addition.

Binary Addition	Decimal Addition
1 1 ← Carry	
0 1	1
+ 1 1	+ 3
Result 1 0 0	4
$100_2 = 4_{10}$	

Example 3: Add 1101 and 1111. Verify the answer with the help of decimal addition.

Binary Addition	Decimal Addition
1 1 1 1 ← Carry	
1 0 0 1	9
+ 1 1 1 1	+ 1 5
1 1 0 0 0	2 4
$11000_2 = 24_{10}$	

Example 4: Add 10111, 11100 and 111. Verify the answer with the help of decimal addition.

Binary Addition	Decimal Addition
1 1 1 1 1 ← Carry	
1 0 1 1 1	2 3
+ 1 1 0 0 0	+ 2 4
1 1 1	7
1 1 0 1 1 0	5 4
$110110_2 = 54_{10}$	

Binary Subtraction

Binary subtraction involves subtracting of two binary numbers. The binary subtraction rules are used while performing the binary subtraction. The binary subtraction rules are shown in Table below, where "Input 2" is subtracted from "Input 1."

Input 1	Input 2		Difference	Borrow
0	0	→	0	No borrow
0	1	→	1	1
1	0	→	1	No borrow
1	1	→	0	No borrow

Table: Binary Subtraction Rules

The steps for performing subtraction of the binary numbers are as follows—

1. Start subtraction by subtracting the bit in the lower row from the upper row, in the unit column.
2. Use the binary subtraction rules. If the bit in the upper row is less than lower row, borrow 1 from the upper row of the next column (on the left side). The result of subtracting two bits is the difference.
3. Write the difference in the result of that column.
4. Repeat steps 2 and 3 for each column, i.e., the tens column, hundreds column and so on.

Example 1: Subtract 01 from 11. Verify the answer with the help of decimal subtraction.

Binary Subtraction	Decimal Subtraction
1 1	3
− 0 1	− 1
Result 1 0	2
$10_2 = 2_{10}$	

Example 2: Subtract 01 from 10. Verify the answer with the help of decimal subtraction.

Binary Subtraction	Decimal Subtraction
0 10	
+ 0	2
− 0 1	− 1
0 1	1
$01_2 = 1_{10}$	

Example 3: Subtract 0111 from 1110. Verify the answer with the help of decimal subtraction.

Binary Subtraction	Decimal Subtraction
0 10 / 0 10 / 0 10 } Borrow	
+ + + 0	1 4
− 0 1 1 1	− 0 7
0 1 1 1	7
$0111_2 = 7_{10}$	

Example 4: Subtract 100110 from 110001. Verify the answer with the help of decimal subtraction.

Binary Subtraction	Decimal Subtraction
1 1	
10 10 10 Borrow	
1 + 0 0 0 1	4 9
1 0 0 1 1 0	− 3 8
0 0 1 0 1 1	1 1
$001011_2 = 11_{10}$	

Signed and Unsigned Numbers

A binary number may be positive or negative. Generally, we use the symbol "+" and "−" to represent positive and negative numbers, respectively. The sign of a binary number has to be represented using 0 and 1, in the computer. An n-bit signed binary number consists of two parts—sign bit and magnitude. The left most bit, also called the Most Significant Bit (MSB) is the sign bit. The remaining n-1 bits denote the magnitude of the number.

In signed binary numbers, the sign bit is 0 for a positive number and 1 for a negative number. For example, 01100011 is a positive number since its sign bit is 0, and, 11001011 is a negative number since its sign bit is 1. An 8–bit signed number can represent data in the range −128 to +127

In an n-bit unsigned binary number, the magnitude of the number n is stored in n bits. An 8–bit unsigned number can represent data in the range 0 to 255 (2^8 = 256).

Complement of Binary Numbers

Complements are used in computer for the simplification of the subtraction operation. For any number in base r, there exist two complements

- r's complement and
- r-1 's complement.

Number System	Base	Complements possible
Binary	2	1's complement and 2's complement
Octal	8	7's complement and 8's complement
Decimal	10	9's complement and 10's complement
Hexadecimal	16	15's complement and 16's complement

There are two types of complements for the binary number system—1's complement and 2's complement.

1's Complement of Binary Number is computed by changing the bits 1 to 0 and the bits 0 to 1.

For example,

1's complement of 101 is 010

1's complement of 1011 is 0100

1's complement of 1101100 is 0010011

2's Complement of Binary Number is computed by adding 1 to the 1's complement of the binary number.

For example,

2's complement of 101 is 010 + 1 = 011

2's complement of 1011 is 0100 + 1 = 0101

2's complement of 1101100 is 0010011 + 1 = 0010100

Subtraction using 1's Complement:

The steps to be followed in subtraction by 1's complement is:

i) Write down 1's complement of the subtrahend.

ii) Add this with the minuend.

iii) If the result of addition has a carryover, then it is added back where a 1 is added in the last bit.

iv) If there is no carry over, then 1's complement of the result of addition is obtained to get the final result and it is negative.

Example 1: 110101 – 100101

1's complement of 100101 is 011010. Hence

Minuend		1 1 0 1 0 1
1's complement of subtrahend		0 1 1 0 1 0
Carry over -	1	0 0 1 1 1 1
		1
		0 1 0 0 0 0

The difference is 10000.

Example 2: 101011 – 111001

1's complement of 111001 is 000110.

Hence,

Minuend -	1 0 1 0 1 1
1's complement of subtrahend -	0 0 0 1 1 0
	1 1 0 0 0 1

Since no carry occurs, take 1's complement of the result.

Hence the difference is 0 0 1 1 1 0.

Example 3: 1011.001 – 110.10

1's complement of 0110.100 is 1001.011 Hence

Minuend -	1 0 1 1 . 0 0 1
1's complement of subtrahend -	1 0 0 1 . 0 1 1
Carry over - 1	0 1 0 0 . 1 0 0
	1
	0 1 0 0 . 1 0 1

Hence the required difference is 100.101

Example 4: 10110.01 – 11010.10

1's complement of 11010.10 is 00101.01

Minuend:	1 0 1 1 0 . 0 1
1's complement of subtrahend:	0 0 1 0 1 . 0 1
	1 1 0 1 1 . 1 0

Since no carry occurs, take 1's complement of the result.

Hence the required difference is: 00100.01

Subtraction by 2's Complement:

The operation is carried out by means of the following steps:

(i) At first, 2's complement of the subtrahend is found.

(ii) Then it is added to the minuend.

(iii) If the final carry over of the sum is 1, it is dropped and the result is positive.

(iv) If there is no carry over, the two's complement of the sum will be the result and it is negative.

Example 1: 110110 - 10110

Solution:

The numbers of bits in the subtrahend is 5 while that of minuend is 6. We make the number of bits in the subtrahend equal to that of minuend by taking a `0' in the sixth place of the subtrahend.

Now, 2's complement of 010110 is (101001 + 1) i.e.101010. Adding this with the minuend.

	1	1 0 1 1 0	Minuend
	1	0 1 0 1 0	2's complement of subtrahend
Carry over 1	1	0 0 0 0 0	Result of addition

After dropping the carryover, we get the result of subtraction to be 100000, which is positive.

Example 2: 10110 – 11010

Solution:

2's complement of 11010 is (00101 + 1) i.e. 00110. Hence

Minuend -	1 0 1 1 0
2's complement of subtrahend -	0 0 1 1 0
Result of addition -	1 1 1 0 0

As there is no carry over, the result of subtraction is negative and is obtained by writing the 2's complement of 11100 i.e. (00011 + 1) or 00100.

Hence the difference is 100.

Example 3: 1010.11 – 1001.01

Solution:

2's complement of 1001.01 is 0110.11. Hence

Minued -		1 0 1 0 . 1 1
2's complement of subtrahend -		0 1 1 0 . 1 1
Carry over	1	0 0 0 1 . 1 0

After dropping the carry-over, we get the result of subtraction as 1.10

Example 4: 10100.01 – 11011.10

Solution:

2's complement of 11011.10 is 00100.10. Hence

Minuend -	1 0 1 0 0 . 0 1
2's complement of subtrahend -	0 0 1 0 0 . 1 0
Result of addition -	1 1 0 0 0 . 1 1

As there is no carry over the result of subtraction is negative and is obtained by writing the 2's complement of 11000.11.

Hence the required result is 00111.01.