CHAPTER 1

NUMBER SYSTEMS AND DIGITAL LOGIC FAMILIES

1.1 Number System

There are four types of popular number systems.

1. Decimal
2. Binary
3. Octal
4. Hexadecimal

Every number system has a radix, r. The radix, r is also called ***base***. The ***radix*** or ***base*** is the number of unique symbols used to represent the number system. Hence the number systems are also called ***radix number systems*** or ***based number systems***.

Table 1.1: Radix/Base and Symbols of Number Systems

Number Systems	Radix/Base	Symbols
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Binary	2	0, 1
Octal	8	0, 1, 2, 3, 4, 5, 6, 7
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Note: $A_{16} = 10_{10}$; $B_{16} = 11_{10}$; $C_{16} = 12_{10}$; $D_{16} = 13_{10}$; $E_{16} = 14_{10}$; $F_{16} = 15_{10}$

In order to differentiate different number system the base or radix is marked as suffix as shown in following examples.

Examples of decimal number: 1024.765_{10} , 79645.23_{10}

Examples of binary number: 1011.101_{2} , 11010.11_{2}

Examples of octal number: 7421.342_{8} , 12467.57_{8}

Examples of hexadecimal number: $2A6B.7C4_{16}$, $1296B.CA_{16}$

1.1.1 Number Representation

A number in general is represented in the following format.

$$\underbrace{d_{n-1} \ldots d_4\, d_3\, d_2\, d_1\, d_0}_{\text{n digit integer part}} \cdot \underbrace{d_{-1}\, d_{-2}\, d_{-3} \ldots d_{-m}}_{\text{m digit fraction part}}$$

Here, d represent the digit in a number. The dot is the seperation between integer part and fraction part.

Each symbol in the number is called ***digit***. Each digit has a ***weight*** which is given by the power of radix.

Table 1.2: Weights of Digits in Number Systems

Number Systems	Weights of Digits .... d_4 d_3 d_2 d_1 d_0 . d_{-1} d_{-2} d_{-3} ....
Decimal	.... 10^4 10^3 10^2 10^1 10^0 . 10^{-1} 10^{-2} 10^{-3} ....
Binary	.... 2^4 2^3 2^2 2^1 2^0 . 2^{-1} 2^{-2} 2^{-3} ....
Octal	.... 8^4 8^3 8^2 8^1 8^0 . 8^{-1} 8^{-2} 8^{-3} ....
Hexadecimal	.... 16^4 16^3 16^2 16^1 16^0 . 16^{-1} 16^{-2} 16^{-3} ....

The following example illustrates the relation between digits and weights of number system.

$$786.35_{10} = 7 \times 10^2 + 8 \times 10^1 + 6 \times 10^0 + 3 \times 10^{-1} + 5 \times 10^{-2} = 786.35_{10}$$

$$572.24_8 = 5 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 + 2 \times 8^{-1} + 4 \times 8^{-2} = 378.3125_{10}$$

$$110.11_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 6.75_{10}$$

$$A79.B2_{16} = A \times 16^2 + 7 \times 16^1 + 9 \times 16^0 + B \times 16^{-1} + 2 \times 16^{-2}$$

$$= \underbrace{10_{10} \times 16^2 + 7 \times 16 + 9 \times 1}_{2681_{10}} + \underbrace{11_{10} \times 16^{-1} + 2 \times 16^{-2}}_{.6953125_{10}} = 2681.6953125_{10}$$

> ***Note:*** *Use calculator to compute the above calculations. The result has 11 digits and so in 10-digit calculator the calculation has to be performed separately for integer part and fraction part.*

The above examples also shows the method to convert any number to decimal number.

1.1.2 Decimal Number System

Decimal number system is the common and widely used number system. The invention of digital systems leads to usage of binary number system. The octal and hexadecimal are useful for short-hand representation of binary numbers. Since decimal number system is the standard number system, the value of a number in any other number system is verified/cross-checked by converting to an equivalent decimal number.

The ***decimal number system*** is a radix number system in which the radix r is 10. Since the radix is 10, the number system uses 10 symbols which are chosen as 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9.

A decimal number in general is represented in the following format.

$$\underbrace{d_{n-1} \ldots d_4\, d_3\, d_2\, d_1\, d_0}_{\text{n digit integer part}} \underset{\substack{\uparrow \\ \text{Decimal} \\ \text{point}}}{.} \underbrace{d_{-1}\, d_{-2}\, d_{-3} \ldots d_{-m}}_{\text{m digit fraction part}}$$

In the above representation, d represents the digit in the decimal number. The suffix in each digit represent power in the weight of the digit.

The dot in decimal number is called ***decimal point*** and it separates the integer part and fraction part.

Each digit of a decimal number has a weight which is given by the power of radix, 10. The digits and corresponding weights of a decimal number are shown in Table 1.3.

Table 1.3: Digits and Weights of Decimal Number

Digits	d_{n-1}	....	d_4	d_3	d_2	d_1	d_0	.	d_{-1}	d_{-2}	d_{-3}	....	d_{-m}
Weights	10^{n-1}	....	10^4	10^3	10^2	10^1	10^0	.	10^{-1}	10^{-2}	10^{-3}	....	10^{-m}

The value of a decimal number can be expressed as a sum of products of decimal digits and its corresponding weights as shown below:

$$d_{n-1} \times 10^{n-1} + \ldots + d_4 \times 10^4 + d_3 \times 10^3 + d_2 \times 10^2 + d_1 \times 10^1 + d_0 \times 10^0 + d_{-1} \times 10^{-1} + d_{-2} \times 10^{-2} + d_{-3} \times 10^{-3} + \ldots + d_{-m} \times 10^{-m}$$

Example: $1932.25_{10} = 1 \times 10^3 + 9 \times 10^2 + 3 \times 10^1 + 2 \times 10^0 + 2 \times 10^{-1} + 5 \times 10^{-2}$

1.1.3 Binary Number System

The ***binary number system*** is a radix number system in which the radix r is 2. Since the radix is 2, the number system uses only two symbols 0 and 1.

A binary number in general is represented in the following format.

$$\underbrace{d_{n-1} \ldots d_4\, d_3\, d_2\, d_1\, d_0}_{\text{n digit integer}} \underset{\substack{\uparrow \\ \text{Binary} \\ \text{point}}}{.} \underbrace{d_{-1}\, d_{-2}\, d_{-3} \ldots d_{-m}}_{\text{m digit fraction}}$$

In the above representation, d represents the digit in the binary number. The suffix in each digit represent power in the weight of the digit.

The dot in binary number is called ***binary point*** and it separates the integer part and fraction part.

Each digit of a binary number has a weight which is given by the power of radix, 2. The digits and corresponding weights of a binary number are shown in Table 1.4.

Table 1.4: Digits and Weights of Binary Number

Digits	d_{n-1}	....	d_4	d_3	d_2	d_1	d_0	.	d_{-1}	d_{-2}	d_{-3}	....	d_{-m}
Weights	2^{n-1}	....	2^4	2^3	2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}	....	2^{-m}

The value of a binary number can be expressed as a sum of products of binary digits and its corresponding weights as shown ahead:

$$d_{n-1} \times 2^{n-1} + \dots + d_4 \times 2^4 + d_3 \times 2^3 + d_2 \times 2^2 + d_1 \times 2^1 + d_0 \times 2^0 + d_{-1} \times 2^{-1} + d_{-2} \times 2^{-2} + d_{-3} \times 2^{-3} + \dots + d_{-m} \times 2^{-m}$$

Example: $1011.01_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$

Table 1.5: Decimal Numbers and Equivalent Binary Numbers

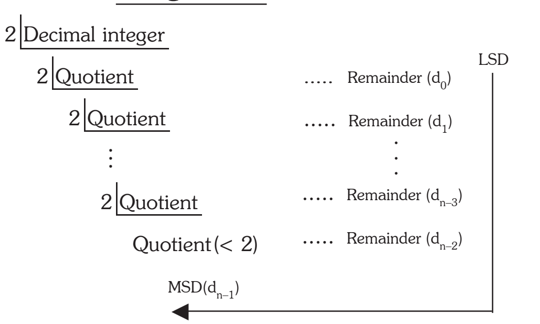
Decimal Number	Binary Number	Decimal Number	Binary Number	Decimal Number	Binary Number	Decimal Number	Binary Number
0	0	11	1011	21	10101	31	11111
1	1	12	1100	22	10110	32	100000
2	10	13	1101	23	10111	33	100001
3	11	14	1110	24	11000	34	100010
4	100	15	1111	25	11001	35	100011
5	101	16	10000	26	11010	36	100100
6	110	17	10001	27	11011	37	100101
7	111	18	10010	28	11100	38	100110
8	1000	19	10011	29	11101	39	100111
9	1001	20	10100	30	11110	40	101000
10	1010						

Decimal to Binary Conversion

Integer part: Divide the integer part of decimal number by 2 and list the quotient and remainder as shown below. Repeat division until quotient is less than 2. The remainders are arranged in the order shown below to get the binary equivalent of decimal integer part.

Fraction Part: Multiply the fraction part of decimal number by 2 as shown below and record the overflow. Repeat multiplication of fraction part alone till the result is zero. The overflows are arranged in the order shown below to get the binary equivalent of decimal fraction.

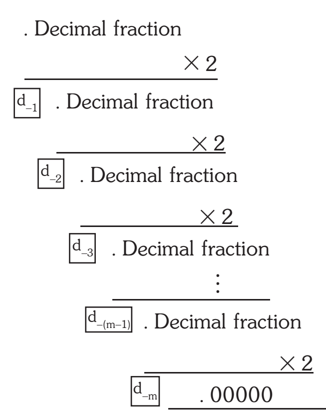
Integer Part



LSD: Least Significant Digit
MSD: Most Significant Digit

Binary integer = $(d_{n-1}\ d_{n-2}\ d_{n-3} \dots\dots\ d_2\ d_1\ d_0)_2$

Fraction Part



Binary fraction = $(.\ d_{-1}\ d_{-2}\ d_{-3} \dots\dots\ d_{-(m-1)}\ d_{-m})_2$

> ***Note:*** *In case of recurring fraction we will not get zero result in repeated multiplication and so stop multiplication when the required number of digits are obtained.*

Binary to Decimal Conversion

For binary to decimal conversion multiply each binary digit by its weight and sum up the product.

Consider the binary number,

$$\ldots\ldots d_3\ d_2\ d_1\ d_0\ .\ d_{-1}\ d_{-2}\ d_{-3}\ \ldots\ldots$$

The decimal number can be obtained from following calculation.

$$\text{Binary number} = \ldots d_3\ d_2\ d_1\ d_0\ .\ d_{-1}\ d_{-2}\ d_{-3}\ldots$$

$$= \ldots d_3 \times 2^3 + d_2 \times 2^2 + d_1 \times 2^1 + d_0 \times 2^0 + d_{-1} \times 2^{-1} + d_{-2} \times 2^{-2} + d_{-3} \times 2^{-3} + \ldots$$

= Equivalent decimal number

Example 1.1

Convert 26.375_{10} to binary and check the result by reverse conversion.

Solution

Integer Part

2 | 26
2 | 13 0
2 | 6 1
2 | 3 0
1 1

1 1 0 1 0
↑ MSD ↑ LSD

$26_{10} = 11010_2$

Fraction Part

.375 × 2 = [0].750

.750 × 2 = [1].500

.500 × 2 = 1.000

. 0 1 1
↑ MSD ↑ LSD

$.375_{10} = .011_2$

$\therefore\ 26.375_{10} = 11010.011_2$

Reverse Conversion

$$11010.011_2 = 1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0 + 0\times 2^{-1} + 1\times 2^{-2} + 1\times 2^{-3}$$

$$= 16 + 8 + 0 + 2 + 0 + 0 + .25 + .125$$

$$= 26.375_{10}$$

1.1.4 Octal Number System

The ***octal number system*** is a radix number system in which the radix r is 8. Since the radix is 8, the number system uses 8 symbols which are chosen as 0, 1, 2, 3, 4, 5, 6 and 7.

An octal number in general is represented in the following format.

$$\underbrace{d_{n-1} \ldots d_4\, d_3\, d_2\, d_1\, d_0}_{\text{n digit integer part}} \cdot \underbrace{d_{-1}\, d_{-2}\, d_{-3} \ldots d_{-m}}_{\text{m digit fraction part}}$$

In the above representation, d represents the digit in the octal number. The suffix in each digit represent power in the weight of the digit.

Each digit of an octal number has a weight which is given by the power of radix, 8. The digits and corresponding weights of an octal number are shown in Table 1.6.

Table 1.6: Digits and Weights of Octal Number

Digits	d_{n-1}	....	d_4	d_3	d_2	d_1	d_0	.	d_{-1}	d_{-2}	d_{-3}	....	d_{-m}
Weights	8^{n-1}	....	8^4	8^3	8^2	8^1	8^0	.	8^{-1}	8^{-2}	8^{-3}	....	8^{-m}

The value of an octal number can be expressed as a sum of products of octal digits and its corresponding weights as shown below:

$$d_{n-1} \times 8^{n-1} + \ldots + d_4 \times 8^4 + d_3 \times 8^3 + d_2 \times 8^2 + d_1 \times 8^1 + d_0 \times 8^0 + d_{-1} \times 8^{-1} + d_{-2} \times 8^{-2} + d_{-3} \times 8^{-3} + \ldots + d_{-m} \times 8^{-m}$$

Example: $246.34_8 = 2 \times 8^2 + 4 \times 8^1 + 6 \times 8^0 + 3 \times 8^{-1} + 4 \times 8^{-2}$

Table 1.7: Decimal Numbers and Equivalent Octal Numbers

Decimal Number	Octal Number	Decimal Number	Octal Number	Decimal Number	Octal Number	Decimal Number	Octal Number
0	0	11	13	21	25	31	37
1	1	12	14	22	26	32	40
2	2	13	15	23	27	33	41
3	3	14	16	24	30	34	42
4	4	15	17	25	31	35	43
5	5	16	20	26	32	36	44
6	6	17	21	27	33	37	45
7	7	18	22	28	34	38	46
8	10	19	23	29	35	39	47
9	11	20	24	30	36	40	50
10	12						

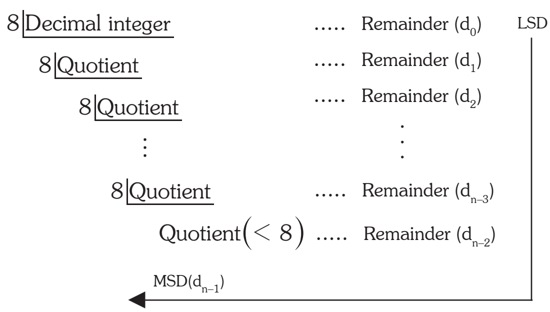
Decimal to Octal Conversion

Integer part: Divide the integer part of decimal number by 8 and list the quotient and remainder as shown below. Repeat division until quotient is less than 8. The remainders are arranged in the order shown below to get the octal equivalent of decimal integer part.

Fraction Part: Multiply the fraction part of decimal number by 8 as shown below and record the overflow. Repeat multiplication of fraction part alone till the result is zero. The overflow are arranged in the order shown below to get the octal equivalent of decimal fraction.

> ***Note:*** *In case of recurring fraction we will not get zero result in repeated multiplication and so stop multiplication when the required number of digits are obtained.*

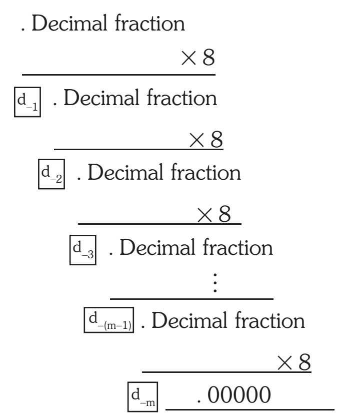
Integer Part



LSD: Least Significant Digit
MSD: Most Significant Digit

Octal integer = $(d_{n-1}\ d_{n-2}\ d_{n-3}\ ....\ d_2\ d_1\ d_0)_8$

Fraction Part



Octal fraction = $(.\ d_{-1}\ d_{-2}\ d_{-3}\ .....\ d_{-(m-1)}\ d_{-m})_8$

Octal to Decimal Conversion

For octal to decimal conversion multiply each octal digit by its weight and sum up the product.

Consider the octal number,

$$.........d_3\ d_2\ d_1\ d_0\ .\ d_{-1}\ d_{-2}\ d_{-3}\ ....$$

The decimal number can be obtained from following the calculation.

$$\text{Octal number} = ...d_3\ d_2\ d_1\ d_0\ .\ d_{-1}\ d_{-2}\ d_{-3}......$$

$$= .....d_3 \times 8^3 + d_2 \times 8^2 + d_1 \times 8^1 + d_0 \times 8^0 + d_{-1} \times 8^{-1} + d_{-2} \times 8^{-2} + d_{-3} \times 8^{-3} + ...$$

= Equivalent decimal number

Example 1.2

Convert 1012.390625_{10} to octal and check the result by reverse conversion.

Solution

Integer Part

8 | 1012
8 | 126 4
8 | 15 6
1 7

$1\ 7\ 6\ 4_8$

↑ MSD ↑ LSD

$1012_{10} = 1764_8$

Fraction Part

$.390625 \times 8 = 3.125$ → 3 (d_{-1}), .125

$.125 \times 8 = 1.000$ → 1 (d_{-2})

$.3\ 1_8$

$.390625_{10} = .31_8$

$\therefore\ 1012\ .\ 390625_{10} = 1764\ .\ 31_8$

Reverse Conversion

$$1764.31_8 = 1 \times 8^3 + 7 \times 8^2 + 6 \times 8^1 + 4 \times 8^0 + 3 \times 8^{-1} + 1 \times 8^{-2}$$

$$= 1012.390625_{10}$$

Note: *Use calculator.*

1.1.5 Hexadecimal Number System

The ***hexadecimal number system*** is a radix number system in which the radix r is 16. Since the radix is 16, the number system uses 16 symbols which are chosen as 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F. The hexadecimal in short can be called ***hexa***.

A hexadecimal number in general is represented in the following format.

$$\underbrace{d_{n-1} \,.....\, d_4\, d_3\, d_2\, d_1\, d_0}_{\text{n digit integer part}} \;.\; \underbrace{d_{-1}\, d_{-2}\, d_{-3} \,.....\, d_{-m}}_{\text{m digit fraction part}}$$

In the above representation, d repesents the digit in the hexa number. The suffix in each digit represent power in the weight of the digit.

Each digit of a hexa number has a weight which is given by the power of radix,16. The digits and corresponding weights of a hexa number are shown in Table 1.8.

Table 1.8: Digits and Weights of Hexadecimal Number

Digits	d_{n-1}	....	d_4	d_3	d_2	d_1	d_0	.	d_{-1}	d_{-2}	d_{-3}	....	d_{-m}
Weights	16^{n-1}	....	16^4	16^3	16^2	16^1	16^0	.	16^{-1}	16^{-2}	16^{-3}	....	16^{-m}

The value of a hexadecimal number can be expressed as a sum of products of hexa digits and its corresponding weights as shown below:

$$d_{n-1} \times 16^{n-1} + ... + d_4 \times 16^4 + d_3 \times 16^3 + d_2 \times 16^2 + d_1 \times 16^1 + d_0 \times 16^0 + d_{-1} \times 16^{-1} + d_{-2} \times 16^{-2} + d_{-3} \times 16^{-3} + ... + d_{-m} \times 16^{-m}$$

Example: $AC2.7B = A \times 16^2 + C \times 16^1 + 2 \times 16^0 + 7 \times 16^{-1} + B \times 16^{-2}$

Table 1.9: Decimal Number and Equivalent Hexadecimal Number

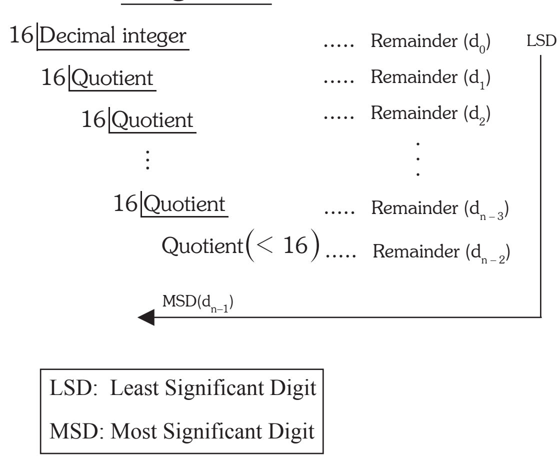
Decimal Number	Hexadecimal Number	Decimal Number	Hexadecimal Number	Decimal Number	Hexadecimal Number	Decimal Number	Hexadecimal Number
0	0	11	B	21	15	31	1F
1	1	12	C	22	16	32	20
2	2	13	D	23	17	33	21
3	3	14	E	24	18	34	22
4	4	15	F	25	19	35	23
5	5	16	10	26	1A	36	24
6	6	17	11	27	1B	37	25
7	7	18	12	28	1C	38	26
8	8	19	13	29	1D	39	27
9	9	20	14	30	1E	40	28
10	A						

Decimal to Hexadecimal Conversion

Integer part: Divide the integer part of decimal number by 16 and list the quotient and remainder as shown below. Repeat division until quotient is less than 16. The remainders are arranged in the order shown below to get the hexadecimal equivalent of decimal integer part.

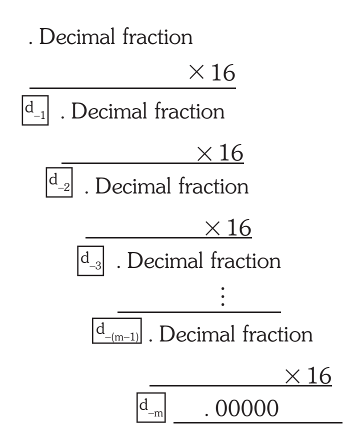
Fraction Part: Multiply the fraction part of decimal number by 16 as shown below and record the overflow. Repeat multiplication of fraction part alone till the result is zero. The overflow are arranged in the order shown below to get the hexadecimal equivalent of decimal fraction.

Integer Part



Hexa integer $= (d_{n-1}\, d_{n-2}\, d_{n-3} \,........\, d_2\, d_1\, d_0)_{16}$

Fraction Part



Hexa fraction $= (.\, d_{-1}\, d_{-2}\, d_{-3} \,.....\, d_{-(m-1)}\, d_{-m})_{16}$

> ***Note:*** *In case of recurring fraction we will not get zero result in repeated multiplication and so stop multiplication when the required number of digits are obtained.*

Hexadecimal to Decimal Conversion

For hexadecimal to decimal conversion multiply each hexa digit by its weight and sum up the product.

Consider the hexadecimal number,

$$\ldots\ldots\ d_3\ d_2\ d_1\ d_0\ .\ d_{-1}\ d_{-2}\ d_{-3}\ \ldots$$

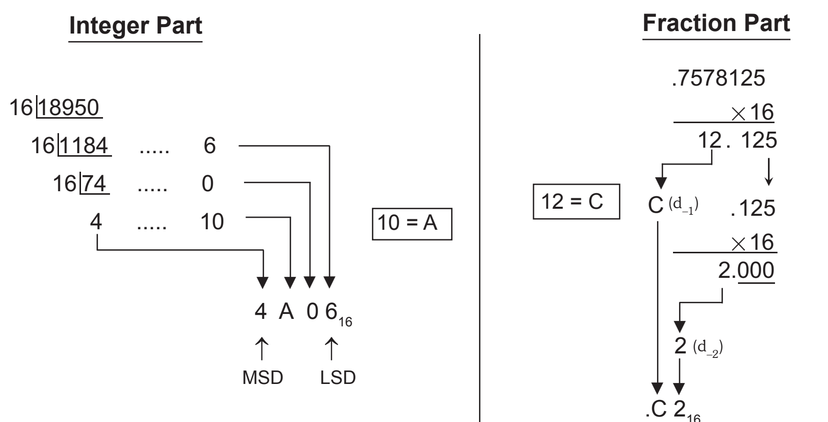
Now the decimal number can be obtained from the following calculation.

$$\begin{aligned}\text{Hexadecimal number} &= \ldots d_3\ d_2\ d_1\ d_0\ .\ d_{-1}\ d_{-2}\ d_{-3}\\ &= \ldots d_3\times16^3 + d_2\times16^2 + d_1\times16^1 + d_0\times16^0\\ &\qquad + d_{-1}\times16^{-1} + d_{-2}\times16^{-2} + d_{-3}\times16^{-3}\\ &= \text{Equivalent decimal number}\end{aligned}$$

Example 1.3

Convert 18950.7578125_{10} to hexadecimal number and cross-check result by reverse conversion.

Solution



$18950_{10} = 4A06_{16}$

$.7578125_{10} = .C2_{16}$

$\therefore\ 18950.7578125_{10} = 4A06.C2_{16}$

Reverse Conversion

$$\begin{aligned}4A06.C2_{16} &= 4\times16^3 + A\times16^2 + 0\times16^1 + 6\times16^0 + C\times16^{-1} + 2\times16^{-2}\\ &= 4\times16^3 + 10\times16^2 + 0 + 6 + 12\times16^{-1} + 2\times16^{-2}\\ &= 18950 + .7578125\\ &= 18950.7578125_{10}\end{aligned}$$

> ***Note:*** *Cannot be calculated in single step by 10-digit calculator.*

1.1.6 Relation between Binary, Octal and Hexadecimal Number Systems

The radix of octal is 8, the radix of hexa is 16.

Here, $8 = 2^3$ and $16 = 2^4$.

Therefore, a hexa digit can be represented by 4 digit binary and an octal digit by 3 digit binary. The octal and hexa decimal numbers are short hand form of binary.

An octal number can be directly converted to binary and vice-versa.

Similarly, a hexa number can be directly converted to binary and vice-versa.

Table 1.10: Binary Equivalent of Octal Digit

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

Table 1.11: Binary Equivalent of Hexa Digit

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

Example 1.4

Convert 726.35_8 to binary and cross-check binary and octal by converting to decimal.

Solution

Note: *Each octal digit is represented by three digit binary.*

$$726.35_8 = 111\ 010\ 110.011\ 101_2$$

$$= 111010110.011101_2$$

Cross-Check

$$726.35_8 = 7\times 8^2 + 2\times 8^1 + 6\times 8^0 + 3\times 8^{-1} + 5\times 8^{-2} = 470.453125_{10}$$

$$111010110.011101_2 = 1\times 2^8 + 1\times 2^7 + 1\times 2^6 + 0\times 2^5 + 1\times 2^4 + 0\times 2^3 + 1\times 2^2 + 1\times 2^1 + 0\times 2^0$$
$$+ 0\times 2^{-1} + 1\times 2^{-2} + 1\times 2^{-3} + 1\times 2^{-4} + 0\times 2^{-5} + 1\times 2^{-6}$$

$$= 2^8 + 2^7 + 2^6 + 2^4 + 2^2 + 2 + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-6}$$

$$= 470.453125_{10}$$

Example 1.5

Convert $A52.42_{16}$ to binary and cross-check binary and hexa by converting to decimal.

Solution

Note: *Each hexa digit is represented by four digit binary.*

$$A52.42_{16} = 1010\ 0101\ 0010.0100\ 0010_2$$

$$= 101001010010.0100001_2$$

Cross-Check

$$A52.42_{16} = A \times 16^2 + 5 \times 16^1 + 2 \times 16^0 + 4 \times 16^{-1} + 2 \times 16^{-2}$$

$$= 10 \times 16^2 + 5 \times 16 + 2 + 4 \times 16^{-1} + 2 \times 16^{-2}$$

$$= 2642 + 0.2578125 = 2642.2578125_{10}$$

$$101001010010.0100001_2 = 1 \times 2^{11} + 0 \times 2^{10} + 1 \times 2^9 + 0 \times 2^8 + 0 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4$$
$$+ 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3}$$
$$+ 0 \times 2^{-4} + 0 \times 2^{-5} + 0 \times 2^{-6} + 1 \times 2^{-7}$$

$$= 2^{11} + 2^9 + 2^6 + 2^4 + 2 + 2^{-2} + 2^{-7}$$

$$= 2642 + 0.2578125 = 2642.2578125_{10}$$

Note: *When using 10-digit calculator, calculate integer part and fraction part separately.*

Example 1.6

Convert 10101001010.10101101_2 to octal and hexa number and cross-check binary, octal and hexa by converting to decimal.

Solution

Separate the binary into group of three digits to find octal number. Represent each three digit binary by an equivalent octal digit.

$$10101001010.10101101_2 = 10\ 101\ 001\ 010.101\ 011\ 01_2$$

$$= 010\ 101\ 001\ 010.101\ 011\ 010_2$$

$$= 2\quad 5\quad 1\quad 2\ .\ 5\quad 3\quad 2_8$$

$$= 2512.532_8$$

Append with leading and trailing zero.

Separate the given binary into group of four digit binary to find hexa number. Represent each four digit binary by an equivalent hexa digit.

$$10101001010.10101101_2 = 101\ 0100\ 1010.1010\ 1101_2$$

$$= 0101\ 0100\ 1010.1010\ 1101_2$$

$$= 5\quad 4\quad A\ .\ A\quad D_{16}$$

$$= 54A.AD_{16}$$

Append with leading zero.

Cross-Check

$$10101001010.10101101_2 = 1 \times 2^{10} + 1 \times 2^8 + 1 \times 2^6 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^{-1}$$
$$+ 1 \times 2^{-3} + 1 \times 2^{-5} + 1 \times 2^{-6} + 1 \times 2^{-8}$$

$$= 1354.67578125_{10}$$

Note: *Weights of zeros are not shown.*

$$2512.532_8 = 2 \times 8^3 + 5 \times 8^2 + 1 \times 8^1 + 2 \times 8^0 + 5 \times 8^{-1} + 3 \times 8^{-2} + 2 \times 8^{-3}$$

$$= 1354.67578125_{10}$$

$$54A.AD_{16} = 5 \times 16^2 + 4 \times 16^1 + A \times 16^0 + A \times 16^{-1} + D \times 16^{-2}$$

$$= 5 \times 16^2 + 4 \times 16 + 10 + 10 \times 16^{-1} + 13 \times 16^{-2}$$

$$= 1354.67578125_{10}$$

Example 1.7

Convert 732.51_8 to binary and hexa and cross-check binary and hexa by converting to decimal.

Solution

$732.51_8 = 111\ 011\ 010.101\ 001_2$ — Represent each octal digit by three digit binary.

$= 111011010.101001_2$

$732.51_8 = 111\ 011\ 010.101\ 001_2$ — Rearrange the binary representation into group of four digits.

$= 1\ 1101\ 1010.1010\ 01_2$ — Append with leading and trailing zeros.

$= 0001\ 1101\ 1010.1010\ \ 0100_2$

$= \ 1 \quad D \quad A \ . \ A \quad 4_{16}$ — Represent each group of four digit binary by an equivalent hexa digit.

$= 1DA\ .\ A4_{16}$

Cross-Check

$$732.51_8 = 7\times8^2+3\times8^1+2\times8^0+5\times8^{-1}+1\times8^{-2}$$

$$= 474.640625_{10}$$

$$111011010.101001_2 = 1\times2^8+1\times2^7+1\times2^6+1\times2^4+1\times2^3+1\times2^1+1\times2^{-1}+1\times2^{-3}+1\times2^{-6}$$

$$= 474.640625_{10}$$

Note: *Weights of zeros are not shown.*

$$1DA.A4_{16} = 1\times16^2+D\times16^1+A\times16^0+A\times16^{-1}+4\times16^{-2}$$

$$= 1\times16^2+13\times16+10+10\times16^{-1}+4\times16^{-2}$$

$$= 474.640625_{10}$$

1.1.7 Unsigned and Signed Binary Number Systems

The binary number system can be unsigned binary or signed binary number system. The unsigned binary has only positive numbers. The ***signed binary*** has positive and negative numbers. There are three types of signed binary number system. They are,

1. Sign-magnitude Form
2. One's Complement Form
3. Two's Complement Form

1.1.8 Unsigned Binary

The general format of an unsigned binary number is given below:

$d_{n-1}\ .....\ d_4\, d_3\, d_2\, d_1\, d_0$

Each binary digit has a weight. The weight of digit, d_k is r^k. Here r = 2. Therefore weight is 2^k.

In a n-bit unsigned binary number the digit d_0 has the weight $2^0 = 1$, which is the lowest weight. Hence the digit d_0 is called ***LSD (Least Significant Digit)***.

In a n-bit unsigned binary number the digit d_{n-1} has the weight 2^{n-1}, which is the largest weight. Hence the digit d_{n-1} is called ***MSD (Most Significant Digit)***.

Range of Unsigned Binary Number

With n-bit binary it is possible to generate 2^n binary numbers.

When n = 1, $2^n = 2^1 = 2$, 1-bit binary numbers: 0, 1

When n = 2, $2^n = 2^2 = 4$, 2-bit binary numbers: 00, 01, 10, 11

When n = 3, $2^n = 2^3 = 8$, 3-bit binary numbers: 000, 001, 010, 011, 100, 101, 110, 111

and so on.

Therefore, the range of decimal number that can be represented by n-bit unsigned binary is 0 to $2^n - 1$.

The binary equivalent of decimal numbers 0 to 31 are listed in Table 1.12.

Sometime we have to write the binary number in a required size of 4-bit or 6-bit or 8-bit, etc. In order to express the unsigned binary to required size, append required number of zeros before MSD. The 8-bit representation of the binary numbers are also shown in Table 1.12.

Table 1.12: Decimal Number and its Equivalent Unsigned Binary Number

Decimal Number	Unsigned Binary Number	8-bit Unsigned Binary Number	Decimal Number	Unsigned Binary Number	8-bit Unsigned Binary Number
0	0	0000 0000	16	10000	0001 0000
1	1	0000 0001	17	10001	0001 0001
2	10	0000 0010	18	10010	0001 0010
3	11	0000 0011	19	10011	0001 0011
4	100	0000 0100	20	10100	0001 0100
5	101	0000 0101	21	10101	0001 0101
6	110	0000 0110	22	10110	0001 0110
7	111	0000 0111	23	10111	0001 0111
8	1000	0000 1000	24	11000	0001 1000
9	1001	0000 1001	25	11001	0001 1001
10	1010	0000 1010	26	11010	0001 1010
11	1011	0000 1011	27	11011	0001 1011
12	1100	0000 1100	28	11100	0001 1100
13	1101	0000 1101	29	11101	0001 1101
14	1110	0000 1110	30	11110	0001 1110
15	1111	0000 1111	31	11111	0001 1111

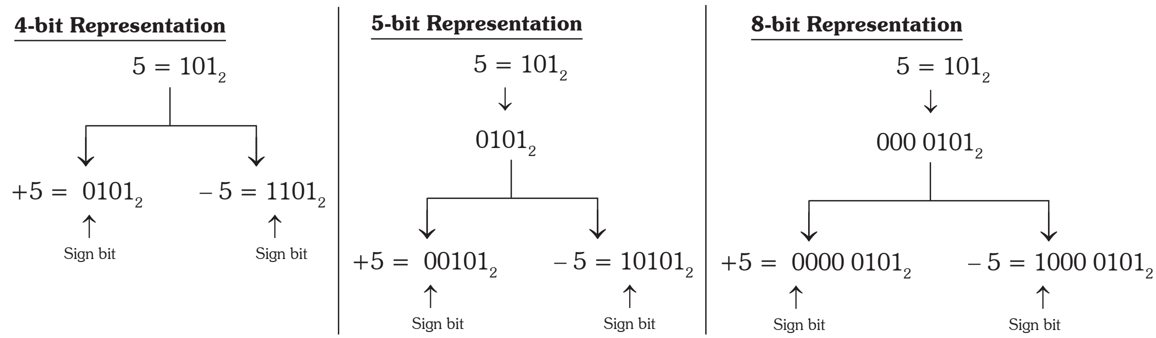
1.1.9 Sign-Magnitude Form

In n-bit binary representation of sign-magnitude form of binary, 1-bit is reserved for sign and n–1 bits are used to represent the magnitude.

For positive number the sign bit is "0" and for negative number the sign bit is "1". The magnitude bits are same for a given number with opposite signs.

In order to represent the given decimal number in n-bit sign-magnitude form, first represent the magnitude as n – 1 bits unsigned number. Then put "**0**" in MSD for positive number and put "**1**" in MSD position for negative number. The drawback in sign-magnitude representation is that there will be two zeros + 0 and – 0. (Refer Table 1.13).

An example of representing $+5_{10}$ and -5_{10} in sign-magnitude form is shown below:



Range of Sign-Magnitude Binary Number

The range of decimal numbers that can be represented in n-bit sign-magnitude form of binary number is,

$$-(2^{n-1}-1) \quad \text{to} \quad +(2^{n-1}-1)$$

The range of decimal numbers that can be represented in sign-magnitude form of binary for n = 4 and 5 are shown below:

n = 4

Range = $-(2^{4-1}-1)$ to $+(2^{4-1}-1)$

= -7_{10} to $+7_{10}$

n = 5

Range = $-(2^{5-1}-1)$ to $+(2^{5-1}-1)$

= -15_{10} to $+15_{10}$

Table 1.13: Decimal and Sign-Magnitude form of Binary Number

Decimal Number	4-bit Sign-magnitude Binary Number	Standard 8-bit Sign-magnitude Binary Number
+0	0000	0000 0000
+1	0001	0000 0001
+2	0010	0000 0010
+3	0011	0000 0011
+4	0100	0000 0100
+5	0101	0000 0101
+6	0110	0000 0110
+7	0111	0000 0111
–0	1000	1000 0000
–1	1001	1000 0001
–2	1010	1000 0010
–3	1011	1000 0011
–4	1100	1000 0100
–5	1101	1000 0101
–6	1110	1000 0110
–7	1111	1000 0111

1.1.10 Complement of Number Systems

Complement of positive numbers are used to represent negative numbers in digital system.

In radix number system, two types of complement are used.

1. r's complement or radix complement.
2. (r – 1)'s complement or diminished radix complement.

Basically, ***complement*** is the number obtained by subtracting a number from it's base number. For an n-digit number the base number is r^n. Therefore, r's complement of an n-digit number N is given by $r^n - N$ and (r – 1)'s complement is given by $(r^n - 1) - N$.

Table 1.14: Summary of Complements in Number Systems

Number Systems	r's Complement	(r – 1)'s Complement
Decimal	10's complement	9's complement
Octal	8's complement	7's complement
Hexa	16's complement	15's complement
Binary	2's complement	1's complement

Importance of Complement

The basic mathematical operation is addition. Subtraction is addition of positive and negative number.

Example: 75 – 42 = 72 + (– 42)

Multiplication is successive addition.

Example: 6 × 3 = 6 + 6 + 6 = 3 + 3 + 3 + 3 + 3 + 3

Division is successive subtraction. Subtraction is again addition of positive and negative number.

Example: 10 ÷ 3 ⇒ 10 – 3 = 7 – 3 = 4 – 3 = 1 (Remainder)

Quotient = 1 + 1 + 1 = 3 (Count of number of subtraction)

Therefore in digital system all mathematical operations are realized in terms of addition.

Therefore, negative number is represented in complement form in digital system, so that every arithmetic operation is performed by addition.

Note: *The ALU (Arithmetic Logical Unit) of a microprocesser is basically an adder capable of performing arithmetic and logical operations in terms of addition.*

1.1.11 One's Complement

In one's complement form of binary number representation, the positive numbers will be same as that of unsigned binary. The negative of a n-digit positive number N is given by $(2^n - 1) - N$. This is same as bit-by-bit complement of the positive number which is changing every 1 to 0 and 0 to 1 of the positive number. Therefore for positive numbers the MSD is "0" and for negative numbers MSD is "1". In order to express the one's complement number in the required size, sign extension technique is employed. In this technique the upper most bit is extended to the required size.