CHAPTER – 1

BINARY SYSTEM

Base Conversion: A number a_n, a_{n-1} … a_2 , a_1 a_0 · a_{-1} a_{-2} a_{-3} … expressed in a base r system has coefficient multiplied by powers of r.

$$\boxed{a_n r^n + a_{n-1} r^{n-1} + a_{n-2} r^{n-2} + ... + a_1 r + a_0 + a_{-1} r^{-1} + a_{-2} r^{-2} + a_{-3} r^{-3} + ...} \quad ...(A)$$

Coefficients a_j; range from 0 to r – 1

Key Points:

To convert a number of base r to decimal is done by expanding the number in a power series as in (A) Then add all the terms.

Example 1: Convert following Binary number $(11010.11)_2$ in to decimal number.

Solution:

Base r = 2

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

$(11010.11)_2 = (26.75)_{10}$

Example 2: Convert $(4021.2)_5$ in to decimal equivalent

Solution: $4\times 5^3 + 0\times 5^2 + 2\times 5^1 + 1\times 5^0 + 2\times 5^{-1}$

$= (511.4)_{10}$

Example 3: Convert $(127.4)_8$ in to decimal equivalent.

Solution: $1\times 8^2 + 2\times 8^1 + 7\times 8^0 + 4\times 8^{-1}$

$= (87.5)_{10}$

Numbers with Different bases:

Decimal (r = 10)	Binary (r = 2)	Octal (r = 8)	Hexadecimal (r = 16)
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4

05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Example 4: Convert following hexadecimal number into decimal number: $(B65F)_{16}$

Solution:

$$11\times16^3+6\times16^2+5\times16^1+15\times16^0=(46.687)_{10}$$

Conversion of decimal number to a number in base r:

- Separate the number into an integer part and fraction part.
- Divide the number and all successive quotients by r and accumulating the remainders.
- Conversion of decimal fraction is done by multiplying the fraction and all successive fraction and integers are accumulated.

Example 1: Convert decimal number 41 to binary.

Solution:

		Integer quotient		Remainder	Coefficient
41/2	=	20	+	1	$a_0 = 1$
20/2	=	10	+	0	$a_1 = 0$
10/2	=	5	+	0	$a_2 = 0$
5/2	=	2	+	1	$a_3 = 1$
2/2	=	1	+	0	$a_4 = 0$
1/2	=	0	+	1	$a_5 = 1$

$(101001)_2$

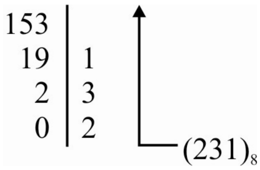
$(41)_{10} \rightarrow (101001)_2$

Example 2: Convert $(153)_{10}$ to octal.

Solution:

Required base r is 8.

153 are divided by 8 to give integer quotient of 19 and remainder 1. Then 19 are divided by 8 to give integer quotient of 2 and remainder 3. Finally 2 are divided by 8 to give quotient of 0 and remainder of 2.



Thus $(153)_{10} \rightarrow (231)_8$

Example 3: Convert $(0.6875)_{10}$ to Binary.

Solution: 0.6875 is multiplied by 2to give an integer and a fraction. The new fraction is multiplied by 2 to give a new integer and new fraction.

This process is continuing until the fraction becomes zero or until the numbers of digits have sufficient accuracy.

		Integer		**Fraction**	**Coefficient**
0.6875 × 2	=	1	+	0.3750	$a_{-1} = 1$
0.3750 × 2	=	0	+	0.7500	$a_{-2} = 0$
0.7500 × 2	=	1	+	0.5000	$a_{-3} = 1$
0.500 × 2	=	1	+	0.0000	$a_{-4} = 1$

$(0.6875)_2 \rightarrow (0.1011)_2$

Example 4: Convert (0.513)10 to octal.

Solution:

0.513 × 8	=	4	+	0.104	$a_{-1} = 4$
0.104 × 8	=	0	+	0.832	$a_{-2} = 0$
0.832 × 8	=	6	+	0.656	$a_{-3} = 6$
0.656 × 8	=	5	+	0.248	$a_{-4} = 5$
0.248 × 8	=	1	+	0.984	$a_{-5} = 1$
0.984 × 8	=	7	+	0.872	

Answer to seven significant figures is:

$(0.406517\ldots)_8$

Thus $(0.513)_{10} \rightarrow (0.406517)_8$

$(41.6875)_{10} \rightarrow (101001.1011)_2$

$(153.513)_{10} \rightarrow (231.406517)_8$

Octal and hexadecimal numbers:

Conversion from binary to octal is easily done by partitioning the binary number into groups of 3 digits each starting from binary point & proceeding to left and to the right.

The corresponding octal digit is then assigned to each group.

For conversion into hexadecimal, binary number is divided into group of 4 digits.

Example: $(2\ 6\ 1\ 5\ 3.7\ 4\ 6\ 0)_8$ to binary number

Solution:

010 110 001 101 011 · 111 100 110 000

Thus binary number is

$(010\ 110\ 001\ 101\ 011.11110011000)_2$

Example 5: Convert binary to hexadecimal number:

$(10\ 1100\ 0110\ 1011.1111\ 0010)_2$

0010 1100 0110 1011. 1111 0010

2 C 6 B F 2 = $(2C6B.F2)_{16}$

Example 6: $(673.124)_8$ to binary number:

$(673.124)_8 \equiv (110\ 111\ 011 \cdot 001\ 010\ 100)_2$

6 7 3 1 2 4

$(306.D)_{16}$ to binary number:

$(306.D)_{16} \equiv (0011\ 0000\ 0110\ .\ 1101)_2$

3 0 6 D

Note: In communication, octal or hexadecimal represented is more desirable because it can be expressed more compactly with a third or a quarter of the number of digits required for the equivalent binary number.

Complements: Complements are used in digital computer for simplifying the subtraction operations and for logic manipulation. *There are 2 types of complements for each base r system*

1. Radix complements (r's complement)
2. Diminished radix complement ((r – 1)'s complement

1. Diminished radix complement:

- Given a number N in base r having n digits, the (r – 1)'s complement of N is defined as $(r^n – 1) – N$.
- For decimal number r = 10, (r – 1)'s complement or 9's complement of N is $(10^n – 1) – N$.

9's complement: $(10^n - 1) - N$

- 10^n can be represented as single 1 followed by n 0's
- $10^n - 1$ is number represented by n 9's.
- Thus 9's complement can be obtained by subtracting each digit of number N by 9's.

Example 7: Find 9's complement of 546700

Solution:

999999 – 546700 = 453299

9's complement of 546700 is 453299

1's Complement for binary number:

- It is given as $(2^n - 1) - N$
- 2^n can be representing as binary number consist of single 1 followed by n 0's.
- $2^n - 1$ can be represented as n 1's.

Example 8: $2_4 \rightarrow 10000$

$24 - 1 \rightarrow (1111)_2$

- Thus 1's complement can be obtained as $(2^n - 1) - N$ or subtracting each digit of number from 1.

Example 9: 1's complement of 1011000.

Solution: 1111111 – 1011000 = 0100111

Note: It is similar to changing 1's to 0's and 0's to 1 or complement each digit of number is similar to taking 1's complement of the number.

Note: (r – 1)'s complement of octal or hexadecimal number is obtained by subtracting each digit from 7 and F respectively.

Example 10: Obtain 15's complement of number $(3241)_{16}$

Solution: Subtracting each digit of number from FFFF:

$$\begin{array}{r} F\,F\,F\,F \\ \underline{-\,3\,2\,4\,1} \\ C\,D\,B\,E \end{array}$$

15's complement is $(CDBE)_{16}$.

(ii) Radix Complement:

r's complement of n digit number N in base r is defined as $r^n - N$ for $N \neq 0$ & 0 for $N = 0$

It is equivalent to adding 1 to (r – 1)'s complement.

If (r – 1)'s complement is given, r's complement can be obtained.

Example: Find r's complement of 546700 if its 9's complement is 453299.

Solution: r's complement is 453299 + 1

r's complement = 453300

Example 11: 2's complement of 1010110 is:

Solution: 1's complement: complement each digit of number (1010110) $\rightarrow$ $(0101001)_2$

Thus 2's complement is 0101001 + 1

2's complement = (0101010)2

<u>Another Method to Obtain 10, 2's Complement</u>:

Leaving all least significant 0's unchanged, subtracting the first non-zero least significant digit from 10 and subtracting all higher significant digits from 9.

Example 12: Find 10's complement of 012398.

Solution:

1. Subtract 8 from 10 in the least significant position
2. Subtracting all other digits from 9.

9999910
<u>- 01239 8</u>
98760 2

Thus 10's complement of 012398 is 987602.

Example: 13 10's complement of 246700.

Solution: Leaving 2 least significant 0's unchanged, subtracting 7 from 10 and other 3 digits from 9.

999 10 00
<u>- 246 7 00</u>
753 3 00

Thus 10's complement of 246700 is 753300

Similarly 2's complement can be formed by leaving all least significant 0's and first 1 unchanged and replacing 1's with 0's and 0's with 1's in all other higher significant digits.

Example 14: 2's complement of $(1101100)_2$:

Solution: 1101 1 00

Remain unchanged

Remain unchanged

Reverse all digits

0010100

Thus 2's complement of 1101100 is $(0010100)_2$

Subtraction with complement:

1. Convert subtrahend N to r's complement.
2. Then add to the minuend M.
3. If M ≥ N, sum will produce end carry, which can be discarded, what is left is the result, M – N.
4. If M < N, sum does not produce carry and is equal to r^n – (N – M), which is same as r's complement of (N – M).
5. To take the answer in familiar form, take the r's complement of the sum and place a negative sign in front.

Example 15: Using 10's complement, subtract 72532 – 3250

Solution: M = 72532

N = 03250

10's complement of N = 96750

Sum: 72532

+ 96750

169282

Discard end carry as M > N so result: 69282

Example 16: Using 10's complement, subtract 3250 – 72532

Solution: M = 3250

N = 72532

10's complement of 72532 is

9999 10

- 7253 2

10's complement 27468

Sum: 3250

27468

Sum 30718

Since N > M so no end carry.

Therefore answer is – (10's complement of 30718) = –69282

Example 16: Subtract 1010100 – 1000011

Solution: 2's complement of N (1000011)=0111101

Sum: 1010100

+ 0111101

10010001

So result is 0010001

Example 17: Subtract: 1000011 – 1010100

Solution: 2's complement of 1010100 → 0101100

Sum: 1000011

+ 0101100

1110111

There is no end carry. Therefore, answer is – (2's complement of 1101111)

= -0010001

Note: Subtraction can also be done using (r – 1)'s complement.

Signed Binary numbers: When binary number is signed, left most bit represents the sign and rest of bits represent the number.

- If binary number is unsigned, then left most bits is the most significant bit of the number.
- Positive or Negative can be represented by (0 or 1) bit which indicate the sign.

Example 19: String of bits 01001 can be considered as 9 (unsigned binary) or +9 (signed binary) because left most bits are 0.

Example 20: String of bits 11001 represent 25 when considered as unsigned number or – 9 when considered as signed number.

Negative number representation:

(i) Signed magnitude representation: In this representation number consist of a magnitude and a symbol (+ or -) or bit (0 or 1) indicating the sign. left most bit represents sign of a number.

Eξ. : 11001 → – 9

01001 → +9

(ii) Signed complement system:

- In this system, negative number is indicated by its complement.
- It can use either 1's or 2's complement, but 2's complement is most common.

Note:

1. 2's complement of positive number remain number itself.
2. In both signed magnitude & signed complement representation, the left most significant bit of negative numbers is always 1.

Example: +9 ® 00001001

- 9 ® 11110111 (2's complement of +9)

Note: Signed complement of number can be obtained by taking 2's complement of positive number including the sign bit.

- Signed magnitude system is used in ordinary arithmetic, can not employed in computer arithmetic because of separate handling of the sign and the magnitude.
- In computer arithmetic signed complement system is used to represent negative numbers.

Decimal	Signed 2'Complement	Signed 1's complement	Signed magnitude
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
-0	-	1111	1000
-1	1111	1110	1001
-2	1110	1101	1010
-3	1101	1100	1011
-4	1100	1011	1100

Arithmetic addition:

- Addition in signed magnitude system follows rules of ordinary arithmetic.
- EX. : +25 + - 37 = -37 + 25 = -12
- Thus In this, comparison of sign and magnitude and them performing either addition or subtraction.
- But in signed complement system, only addition, it does not require comparison & subtraction.
- In signed complement system, negative numbers are represents in 2's complement form and then addition to other number including their sign bits.

Example: + 6 00000110 - 6 11111010 (2's complement)

+13	00001101	+13	00001101
+19	00010011	+ 7	100000111
+ 6	00000110	- 6	11111010
- 13	11110011	-13	11110011
- 7	111111001	-19	11101011

[Left significant bit is 1 so number is negative, number will be –(2's complement of 111111001)

= -(000000111) = – 7

Number will be: – (2's complement of 11101011) = –(00010101) = –(19)

Note: If result of sum is negative, then it is in 2's complement form.

The left most significant bit of negative numbers is always 1.

- If we use signed complement system, computer needs only one hardware circuit to handle both arithmetic (signed & unsigned), so generally signed complement system is used.

Binary Codes:

Any discrete element of information distinct among a group of quantities can be represented with a binary code.

- n bit binary code is a group of n bits that have 2n distinct combinations of 1's and 0's with each combination representing one element of the set that is being coded.

Example: With 2 bits $2^2 = 4$ elements can be coded as: 00, 01, 10, 11

With 3 bits $2^3 = 8$ elements can be coded as:

000, 001, 010, 011, 100, 101, 110, 111

- Minimum number of bits required to code 2^n distinct quantities in n.
- The bit combination of an n bit code is determined from the count in binary from 0 to $2^n - 1$.

Example: 3 bit combination

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

BCD code:

Binary coded decimal

- A number with k decimal digits require 4 K bits in BCD.
- A decimal number in BCD is same as its equivalent binary number only when number is between 0 to 9.
- BCD number needs more bits that its equivalent binary.
- Example: $(185)_{10} = (000110000101)_{BCD} = (1011101)_2$
- In BCD number, each bit is represented by its equivalent binary representation.

Note: BCD numbers are decimal numbers and not binary numbers, although they use bits in their representation.

- Decimal are written as 0,1,2,3,…,9 which BCD can be written as : 0000, 0001, 0010, 0011, …, 1001

Benefits of BCD:-

- BCD helps to do arithmetic operation directly on decimal numbers without converting them into equivalent binary numbers.

Decimal system	BCD digits	Binary equivalent
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	00010000	1010
11	00010001	1011

BCD addition:

- If binary sum is equal to or less than 1001 (without a carry), the corresponding BCD digit is correct.
- If binary sum ≥ 1010, the result is an invalid BCD digit.
- Addition of 6 = $(0110)_2$ to the binary sum converts it to the correct digit and also produces a carry as required.

Example:

4	0100	4	0100	8	1000
+5	+0101	+8	+1000	+9	+1001
9	1001	12	1100	17	10001
			+0110		0110
			10010		10111

Example: Add 184 + 576 in BCD.

Solution:	1		1	
	0001	1000	0100	184
	<u>0101</u>	<u>0111</u>	<u>0110</u>	<u>+576</u>
Binary sum	0111	10000	1010	
Add 6		<u>0110</u>	<u>0110</u>	
BCD sum	0111	0110	0000	
	7	6	0	760

The first least significant pair of BCD digits produces a BCD digits sum of 0000 and carries for the next pair of digits. The second pair of (BCD digits + carry) produces digit sum of 0110 and carry for next pair of digits. The third pair of digits plus carry produces binary sum of 0111 and does not require a correction.

- Representation of Signed decimal numbers in BCD is similar to the representation of signed number in binary.
- Sign of decimal number is represented with 4 bits :

Positive number: '0000' (0)

Negative number – '1001' (9)

Example: Do the following in BCD system:

(+375) + (-240)

Solution: +375 can be represented as 0375

-240 in 10's complement form can be represented as 9760

Put to represent negative number.

+ 0375

<u>- 9760</u>

10135 in which 1 can be discarded.

Thus result is 0135.

0 is representing that number is positive.

Other Decimal codes:

Many different codes can be formulated by arranging 4 bits in 10's distinct possible combination.

- Each code shown uses 10 bit combination out of 16 bit combinations.
- BCD & 2421 codes are example of weighted codes.

4 Different Binary codes for the decimal digits:

- The 2421 and excess 3 code are example of self complementing codes. These are having property that 9's complement can be directly obtained by changing 1's to 0's and 0's to 1's.
- BCD code is not self complementing (or reflective code).
- Self complementing codes are also called reflective code.

Example: 9's complement of 395 is 604 which can be obtained as

395

0011 1001 010 ← (BCD)

0110 1100 1000 (excess 3 code)

1001 0011 0111 (complement 1's)

(This excess 3 code of 604)

Gray code:

In this only one bit in the code group changes when going from one number to the next number.

	Gray code	Binary code
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111

- Gray code is used where normal sequence of binary numbers may produce an error during transition from one number to the next.

Example: 0111 → 1000 may produce an intermediate result of 1001 if LSB takes longer to change than other 3bits.

- In gray code, since only 1bit change during transition between 2 numbers, so no such error occurs.

Binary number can be converted to gray code as :

Binary number Gray code

$a_4\ a_3\ a_2\ a_1$ $G_4\ G_3\ G_2\ G_1$

- keep MSB a_4 same i.e. $G_4 = a_4$