1

1.1 INTRODUCTION

Binary number system, hexadecimal number system and the octal number system are the three important systems used in different computer applications. With these systems, the computer can work very efficiently and effectively. The first number system is the main number system used in every type of computer application. Even the hardware of the computer is also designed for binary number system to work effectively. In this chapter, we shall study three basic number systems, binary, hexadecimal and octal number systems along with their internal relationship and conversion procedure between them.

1.2 DIFFERENT NUMBER SYSTEMS

Decimal number system: since our school education, we are familiar with decimal number system only. It consists of ten symbols i.e. 0 to 9. With the help of these symbols, i.e. with their combination we can obtain any small or big decimal number like 2345, 987987, 780906656 etc. In this system, we see that its radix or base is (10). It means that any given decimal number can be expressed in respective power of 10, as follows –

$$5637 = (5 \times 10^3) + (6 \times 10^2) + (3 \times 10^1) + (7 \times 10^0) = 5000 + 600 + 30 + 7$$

Binary number system: this system contains only two symbols: a 1 and a 0. Hence, its base or radix is (2). With their different combinations, we can obtain any small or big binary number like $(11101)_2$, $(10101000)_2$ etc. The suffix '2' indicates binary number.

In a binary number each digit is called as bit. The word bit stands for binary digit. The positions of digits are indicated with different method. Suppose we have a binary number as $(11001)_2$, then left-most bit is called MSB i.e. Most Significant Bit and the right-most bit is called LSB i.e. Least Significant Bit.

Also from left to right direction, the positions are given as MSB, second MSB, third MSB and so on. But from right to left direction, the positions are given as LSB, second LSB, third LSB and so on.

Hexadecimal number system: this system contains sixteen symbols: 0 to 9 and A, B, C, D, E & F. Hence, its base or radix is (16). With different combination, we can obtain any small or big hexadecimal (hex) number like $(A34)_{16}$, $(BA345C)_{16}$ etc. The suffix '16' indicates hexadecimal number.

1.2.1 BINARY TO DECIMAL CONVERSION

Suppose we have a binary number given as follows: 1101. To avoid confusion with decimal number as one thousand one hundred and one, we shall write this number as $(1101)_2$. Here the radix '2' shows that the number in bracket is binary number. Now to convert this binary number into equivalent decimal number, we shall use following steps –

Step 1: Split the number into separate digits –

1 1 0 1

Step 2: Write 1, 2, 4, 8 . . . from right to left, directly under the number, as shown below –

1 1 0 1

8	**4**	**2**	**1**

Step 3: Strike out the decimal digit below a binary zero –

1	**1**	**0**	**1**
8	**4**	$\cancel{\mathbf{2}}$	**1**

Step 4: Add the remaining decimal digits to get the answer –

$8 + 4 + 1 = (13)_{10}$… the radix '10' shows the number is decimal number.

1.2.2 DECIMAL TO BINARY CONVERSION

Suppose we have a decimal number $(23)_{10}$. Now we shall convert it into equivalent binary form. This method of conversion is called double-dabble method.

Step 1: Divide the number *successively* by the binary radix 2 as follows –

23 ÷ 2 = 11	**with a remainder 1**	
11 ÷ 2 = 5	**with a remainder 1**	
5 ÷ 2 = 2	**with a remainder 1**	
2 ÷ 2 = 1	**with a remainder 0**	
1 ÷ 2 = 0	**with a remainder 1**	Read

Step 2: Read the remainders in sequence from BOTTOM TO TOP and write the number as –

$(23)_{10} = (10111)_2$ … mark the number with radix 2.

EXERCISE ON THE TOPIC

1. Convert the following given binary number into its unique equivalent decimal number –
   a. $(1101)_2 = (?)_{10}$
   b. $(10110)_2 = (?)_{10}$
   c. $(1101110)_2 = (?)_{10}$
   d. $(1111111111111)_2 = (?)_{10}$
   e. $(11)_2 = (?)_{10}$
   f. $(1000001)_2 = (?)_{10}$
2. Convert the following given decimal number into its unique equivalent binary number –
   a. $(21)_{10} = (?)_2$
   b. $(11)_{10} = (?)_2$
   c. $(16)_{10} = (?)_2$
   d. $(105)_{10} = (?)_2$
   e. $(248)_{10} = (?)_2$
   f. $(014)_{10} = (?)_2$
   g. $(111)_{10} = (?)_2$
   h. $(255)_{10} = (?)_2$

1.3 HEXADECIMAL NUMBER SYSTEM

This system contains sixteen symbols: from 0 to 9 and A, B, C, D, E & F. Hence, its base or radix is (16). The relation of binary, decimal and hexadecimal numbers –

Decimal Number	Binary Number	Hexadecimal Number
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7

Decimal Number	Binary Number	Hexadecimal Number
8	1000	8
9	1001	9
10	1010	**A**
11	1011	**B**
12	1100	**C**
13	1101	**D**
14	1110	**E**
15	1111	**F**

1.3.1 DECIMAL TO HEXADECIMAL CONVERSION

Suppose we have a decimal number 420. *To avoid confusion with hex number*, we shall write it as $(420)_{10}$. Now we shall convert it into equivalent hex form. This method of conversion is called HEX-DABBLE method.

Step 1: Divide the number *successively* by the hexadecimal radix 16 as follows –

420 ÷ 16 = 26 with a remainder 4 → 4

26 ÷ 16 = 1 with a remainder 10 → A

1 ÷ 16 = 0 with a remainder 1 → 1

Read

Step 2: Read the remainders in sequence from BOTTOM TO TOP and write the number as –

$\mathbf{(420)_{10} = (1A4)_{16}}$ … the radix '16' indicates hexadecimal number.

1.3.2 HEXADECIMAL TO DECIMAL CONVERSION

Suppose we have a hex number $(2E5)_{16}$ … the radix '16' indicates hex number. We shall convert it as follows –

Step 1: Split the number into separate digits –

2 E 5

Step 2: Multiply each digit with respective radix power i.e. 16^0, 16^1, 16^2, 16^3. . . . from right to left, as shown below –

$$(2 \times 16^2) \qquad (14 \times 16^1) \qquad (5 \times 16^0)$$

Step 3: Add these figures to obtain the final answer, as follows –

$$(2 \times 16^2) + (14 \times 16^1) + (5 \times 16^0) = 512 + 224 + 5 = (741)_{10}$$

EXERCISE ON THE TOPIC

1. Convert the following given decimal number into its unique equivalent hexadecimal number –
   a. $(16)_{10} = (?)_{16}$
   b. $(420)_{10} = (?)_{16}$
   c. $(10)_{10} = (?)_{16}$
   d. $(2989)_{10} = (?)_{16}$
   e. $(64)_{10} = (?)_{16}$
2. Convert the following given hexadecimal number into its unique equivalent decimal number –
   a. $(CAD)_{16} = (?)_{10}$
   b. $(ABC)_{16} = (?)_{10}$
   c. $(162)_{16} = (?)_{10}$
   d. $(82)_{16} = (?)_{10}$
   e. $(100)_{16} = (?)_{10}$

1.4 FRACTIONAL NUMBERS

All these number systems also express their numbers in the form of integers and in the form of fractions also. These methods are given below.

1.4.1 FRACTIONAL BINARY TO FRACTIONAL DECIMAL CONVERSION

Just like a decimal number, a binary number is also represented in fractional form. Consider a fractional binary number –

$$\mathbf{(0.101)_2}$$

The point after the binary zero is called binary point and it indicates that it is fractional number. Such number can be converted into equivalent decimal number as follows –

Step 1: Split the number, into separate digits –

0	**.**	**1**	**0**	**1**

Step 2: Write 2^{-1}, 2^{-2}, 2^{-3} . . . from LEFT TO RIGHT, after binary point, below the number shown below –

0	**.**	**1**	**0**	**1**
0	**.**	**(0.5)**	**(0.25)**	**(0.125)**

Step 3: Strike out decimal fraction below a binary zero –

0	**.**	**1**	**0**	**1**
0	**.**	**(0.5)**	**~~(0.25)~~**	**(0.125)**

Step 4: Add remaining decimal fractions to get the answer –

$$0.5 + 0.125 = (0.625)_{10}$$

1.4.2 FRACTIONAL DECIMAL TO FRACTIONAL BINARY CONVERSION

Suppose we have a fractional decimal number $(0.45)_{10.}$ We shall convert it as follows –

Step 1: Multiply the number *successively* by the binary radix 2 –

0.45 x 2 = 0.9	**with a carry of 0**	Read ↓
0.9 x 2 = 1.8	**with a carry of 1**	
0.8 x 2 = 1.6	**with a carry of 1**	
0.6 x 2 = 1.2	**with a carry of 1**	
0.2 x 2 = 0.4	**with a carry of 0**	

Step 2: Read the remainders in sequence, from TOP TO BOTTOM, and write number as follows –

$$(0.45)_{10} = (0.01110)_2$$

1.4.3 FRACTIONAL DECIMAL TO FRACTIONAL HEXADECIMAL CONVERSION

Suppose we have a fractional decimal number as $(0.16)_{10.}$ We shall convert it as follows –

Step 1: Multiply the number *successively* by the radix 16 as follows –

0.16 x 16 = 2.56	**with a carry of**	**2**	Read ↓
0.56 x 16 = 8.96	**with a carry of**	**8**	
0.96 x 16 = 15.36	**with a carry of**	**15 → F**	
0.36 x 16 = 5.76	**with a carry of**	**5**	

Step 2: During this conversion, ALWAYS STOP AT 4TH STEP, for approximate answer.

Step 3: Read the remainders in sequence, from TOP TO BOTTOM, and write the number –

$$(0.16)_{10} = (0.28F5)_{16}$$

1.4.4 FRACTIONAL HEXADECIMAL TO FRACTIONAL BINARY CONVERSION

Just like decimal number, a hex number is also represented in fractional form. Consider a fractional hex number –

$$(294.\ F82C)_{16}$$

The point after the hex zero is called hexadecimal point and it indicates that it is fractional number. Such number can be converted into equivalent binary number as follows –

Step 1: Split the number into separate digits –

2 9 4 . F 8 2 C

Step 2: Write down the binary equivalent codes, below each, in nibble form –

2	**9**	**4**	.	**F**	**8**	**2**	**C**
0010	**1001**	**0100**		**1111**	**1000**	**0010**	**1100**

Step 3: Combine these codes to get final answer, as follows *(omit zeros on left & right side)* –

$$(294.F82C)_{16} = (1010010100.11111000001011)_2$$

1.4.5 FRACTIONAL BINARY TO FRACTIONAL HEXADECIMAL CONVERSION

Suppose we have a binary number as $(1011010011.\ 0010111)_2$. We shall convert it as follows –

Step 1: Split the number into separate digits –

1011010011 . 0010111

Step 2: Make groups of four bits from left to right (on right side of binary point) and from right to left (on left side of binary point) as shown below –

two zeros inserted to complete nibble → **0010 1101 0011 . 0010 1110** ← one zero inserted to complete nibble

← direction of grouping   direction of grouping →

Step 3: Write down the hex equivalent codes of binary nibbles below each group as shown below –

0010	**1101**	**0011**	.	**0010**	**1110**
2	**D**	**3**		**2**	**E**

Step 3: Combine these codes to get final answer, as follows –

$$(1011010011.\ 0010111)_2 = (2D3.2E)_{16}$$

EXERCISE ON THE TOPIC

1. Convert the following given fractional numbers into their unique equivalent fractions –
   a. $(1100.1011)_2 = (?)_{10}$
   b. $(124.36)_{10} = (?)_2$
   c. $(0.15)_{10} = (?)_2$
   d. $(56.16)_{10} = (?)_{16}$
   e. $(94DF.10F)_{16} = (?)_2$
   f. $(110.12)_{10} = (?)_{16}$
   g. $(110101011101110001.111001100011101)_2 = (?)_{16}$
   h. $(11.0101)_2 = (?)_{10}$
   i. $(13.02)_{10} = (?)_{16}$

1.5 CONCEPT OF CODES

A code is a computer utility language used in all types of its operations. Each code has specific rules and Operating System (called as OS). As given below, we shall study different codes. A code is necessary in digital circuits. The computer for example, uses binary codes, to handle numerals, alphabets and special symbols. Therefore, it is necessary to convert a given code into its equivalent binary code, when we want to process it with digital circuits. The process of converting one type of code into another is called as ENCODING PROCESS. Basically, codes are used in computers to detect errors, version type, level corrections, assembly language programming etc.

1.5.1 CONCEPT OF BCD CODE

Binary Coded Decimal (BCD) Code: It is also known as 8421BCD code. In this code the digits of given decimal number are ENCODED one at a time, into binary nibble form. Each digit in decimal number is separated and assigned to specific weight w.r.t. binary. The method of conversion is given below –

Step 1: Suppose the given decimal number is $(8392)_{10}$. Now we want to convert it into BCD code. Then split the given number into separate digits, as shown –

8 3 9 2

Step 2: Now write down the binary weightage of each decimal digit below it, in a group of four bits as follows –

8	3	9	2
1000	0011	1001	0010

Step 3: Now combine each group of binary number together to obtain BCD equivalent.

$$(8392)_{10} = (1000001110010010)_{BCD}$$

In this way, given decimal number can be converted into BCD code. Thus, the 8421–BCD code is same w.r.t. binary–for–decimal only from 0 = 0000 to 9 = 1001, but higher combinations like 10 . . . onwards are all classified into FORBIDDEN GROUP.

The basic advantage of this code is that, we have to remember binary equivalents from 0 to 9 only. Then any decimal number can be converted into equivalent BCD code, using following chart –

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

Decimal	BCD code
8	1000
9	1001
10	0001 0000
11	0001 0001
12	0001 0010
13	0001 0011
14	0001 0100
15	0001 0101

1.6 BINARY ARITHMETIC

In binary arithmetic, we can add or subtract binary numbers with the help of different methods. These methods use simple techniques for addition and subtraction of binary numbers, just like usual addition and subtraction process of decimal number system.

1.6.1 RULES OF BINARY ADDITION & SUBTRACTION

Binary addition method: In this method following important rules are used –

1) **0 + 0 = 0**
2) **0 + 1 = 1**
3) **1 + 0 = 1**
4) **1 + 1 = 10**
5) **1 + 1 + 1 = 10 + 1 = 11**

Rules 2 and 3: commutativity

Rules 4 and 5: note these rules!

Note the steps -
$(1)_2 + (1)_2 = (1)_{10} + (1)_{10} = (2)_{10} = (10)_2$
$(1)_2 + (1)_2 + (1)_2 = (1)_{10} + (1)_{10} + (1)_{10} = (3)_{10} = (11)_2$

With these rules, we can add any two binary numbers, small or large as follows. Here is an example of addition process. Just like decimal number system, it starts from rightmost side of the numbers –

1) **0 + 1 gives 1**
2) **0 + 0 gives 0**
3) **1 + 0 gives 1**
4) **1 + 1 gives 10**

$$\begin{array}{rrl} & 1100 & \rightarrow 12 \\ + & 1001 & \rightarrow 9 \\ \hline & 10101 & \rightarrow 21 \end{array}$$

Binary subtraction method: In this method following important rules are used –

1) **0 – 0 = 0**
2) **1 – 0 = 1**
3) **10 – 1 = 1**
4) **1 – 1 = 0**
5) **11 – 1 = 10**

Rules 2 and 3: note these rules!

With these rules, we can subtract *only smaller binary number from larger number.* It takes up the following steps –

1) **1 – 1 gives 0**
2) **0 – 0 gives 0**
3) **1 – 0 gives 1**
4) **1 – 1 gives 0**

$$\begin{array}{rrl} & 1101 & \rightarrow 13 \\ - & 1001 & \rightarrow 9 \\ \hline & 0100 & \rightarrow 4 \end{array}$$

Important note: in subtraction, whenever we come across a step like 0 – 1 =? In that case we take a borrow (1) for upper (0) and then subtract the lower 1 from 10. However, we can also subtract larger binary number from a smaller binary using two different methods. They are known as subtraction using 1's and 2's complement methods.

EXERCISE ON THE TOPIC

1. Add the following binary numbers and obtain their addition result in each case –
   a. $(11001)_2$ *and* $(10011)_2$
   b. $(10111)_2$ *and* $(101111)_2$
   c. $(10011110)_2$ *and* $(101110011)_2$
   d. $(110.1011)_2$ *and* $(1011.0111)_2$
   e. $(1111111)_2$ *and* $(1101000)_2$
2. Solve the following and obtain result in each case –
   a. $(11001)_2 - (1011)_2$
   b. $(110111)_2 - (100111)_2$
   c. $(1110111)_2 - (10111)_2$
   d. $(110)_2 - (101)_2$
   e. $(1100)_2 - (111)_2$

1.7 SUBTRACTION USING 1'S COMPLEMENT METHOD

The 1's complement means in a given binary number, each 1 is changed to a 0 and each 0 is changed to a 1. Thus, obtained number is called 1's complement binary number of given number. For example –

$$1000 \xrightarrow{\text{1's complement}} 0111$$

$$1001 \xrightarrow{\text{1's complement}} 0110$$

1.7.1 LARGER MUNUS SMALLER

Suppose we have to subtract 1001 from 1110, then we shall use following steps: First, obtain the 1's complement of smaller binary number. Then add 1's complement number in larger number. Do simple addition as usual. After addition, if a carry *(i.e. a 1 called as End Around Carry)* generates, remove it from its position and add it in the obtained number. Finally the number we get is the required answer of subtraction of smaller number from larger number.

```
  1110   → 14
+ 0110   → 1's complement of 1001
 -----
 10100
+ └→ 1   → remove EAC & add
 -----
  0101   → 5 (required answer)
```

1.7.2 SMALLER MUNUS LARGER

Suppose we have to subtract 1101 from 1010, then we shall use following steps: First, obtain the 1's complement of larger binary number. Add 1's complement number in smaller number. Do simple addition as usual. After addition, you will find that the End Around Carry (EAC) is absent. Note that when EAC is absent ANSWER IS NEGATIVE AND IN 1's COMPLEMENT FORM. So take 1's complement of obtained number and mark it with negative sign. This is the required answer.

```
  1010
+ 0010   → 1's complement of 1101
 -----
| 1100   → take 1's complement again
↓                    ↓
EAC is
absent            −(0011)
          Final answer is negative
```

1.8 SUBTRACTION USING 2'S COMPLEMENT METHOD

In this method, first 1's complement of given number is taken and then a 1 is added into 1's complement to get 2's complement of the given number. For example –

$$\mathbf{0001} \xrightarrow{\textbf{2's complement}} \mathbf{1110 + 1 = 1111}$$

$$\mathbf{1101} \xrightarrow{\textbf{2's complement}} \mathbf{0010 + 1 = 0011}$$

1.8.1 LARGER MUNUS SMALLER

Suppose we have to subtract 1011 from 1101, then we shall use following steps: First, obtain the 2's complement of smaller number. Add 2's complement number in larger number. Do simple addition as usual. After addition, End Around Carry generates. Ignore it. The remaining number is the required answer.

$$\begin{array}{rl} 1101 & \rightarrow 13 \\ +\ 0101 & \rightarrow \text{2's complement of 1011} \\ \hline 10010 & \\ & \rightarrow \text{ignore this EAC} \\ \hline 0010 & \rightarrow \text{2 (required answer)} \end{array}$$

1.8.2 SMALLER MUNUS LARGER

Suppose we have to subtract 1111 from 1001, then we shall use following steps: First, obtain the 2's complement of larger binary number. Add 2's complement number in smaller number. Do simple addition as usual. After addition, you will find that the End Around Carry (EAC) is absent. Note that when EAC is absent ANSWER IS NEGATIVE AND IN 2's COMPLEMENT FORM. So take 2's complement of obtained number and mark it with negative sign. This is the required answer.

$$\begin{array}{rl} 1001 & \\ +\ 0001 & \rightarrow \text{2's complement of 1111} \\ \hline 1010 & \rightarrow \text{take 2's complement again} \end{array}$$

EAC is absent

$-(0110) \rightarrow 6$

Final answer is negative

EXERCISE ON THE TOPIC

1. Solve the following using 1's complement methods –
   a. $(1001)_2 - (1101)_2$
   b. $(0000)_2 - (1111)_2$
   c. $(11011)_2 - (1100)_2$
   d. $(1110)_2 - (111111)_2$
   e. $(11000001)_2 - (1101)_2$
2. Solve the following using 2's complement methods –
   a. $(11011)_2 - (11100)_2$
   b. $(1100)_2 - (1101)_2$
   c. $(11111)_2 - (1110001)_2$
   d. $(110001111)_2 - (1111)_2$
   e. $(1001)_2 - (1101)_2$
   f. $(01111)_2 - (0111)_2$

Self Examination

Objective questions

1. Converting the given decimal number into its unique equivalent binary is called as _________ method.
2. When a given decimal number is converted into its unique equivalent hexadecimal number, this method is called as ___________ method.
3. The binary number 1110011 is equivalent to its decimal as __________.
4. The hexadecimal number 1A2 is equivalent to its binary as ______________.
5. In the rules of addition, when a 1 is added to 11, we get the answer as ________.
6. The binary number 1111 is the 1's complement of _________.
7. In 1's complement method of subtraction, when we subtract smaller binary number from larger binary number, the _________ is generated which must be added to the obtained number for getting the final answer.
8. The binary number 10000 is the 2's complement of __________.
9. The BCD code 1000 0000 is equivalent to the decimal number as _________.
10. The EBCDIC code is an _________ bit code.
11. There are 7-bits in _________ code.
12. The equivalent ASCII code of '$' sign is ________________.

Long answer questions (4 Marks)

1. Explain the conversion procedure of binary number into its unique equivalent decimal number with one example.
2. How a larger binary number is subtracted from a smaller binary number using 2's complement method? Explain with an example.
3. What is code? Explain the different types of codes with one example of each.

Conceptual study questions

1. Search the web and obtain the information on K-map, used to solve different complicated problems related to binary.
2. Why we cannot use decimal number system as the basic number system in the processing of computers? Explain with reasons.
3. What is octal number system? How many digits are there in this number system?
4. Can you define some simple rules, to remember the ASCII coding and EBCDIC coding systems? Explain.

NOTES SPACE