Unit
04

BINARY ARITHMETIC

UNIT STRUCTURE

4.0 Learning Objectives :

After working through this unit, you should be able to :

- Understand Binary Coded Decimal (BCD) System.
- Know, representation of alphabets and numbers into the memory.
- Understand how negative numbers are represented in the memory.
- Get the knowledge about how computer system adds or subtracts binary numbers.

4.1 Introduction :

In our previous unit, we have seen that the machine can understand only machine language, which is often known as binary language. We have discussed how can we convert a decimal number into binary and any binary number into decimal. We have seen binary, octal decimal and hexa–decimal number systems. In this unit we will start with another type of binary representation, which is called binary coded decimal (BCD).

We will also discuss how the alphabets strings and numbers will be stored in the computer memory.

We will discuss how computer process the binary numbers and how two binary numbers are added or subtracted. Finally, we will see how computer can detect or correct the error in the binary format of data.

4.2 Representation of Data :

In the previous unit we have seen different number systems. In this unit we will focus on how data will be stored in the computer memory. From the discussion made in the previous unit, you know that the data are stored in the memory in the form of binary digits called bits. In this unit we will try to find out answer of how computer perform arithmetic operations such as addition or subtraction on these bits. Along with this we will also see how alphabets or strings will be stored inside the computer memory.

4.2.1 Binary Coded Decimals (BCD) :

The binary number system is a natural language for computers because they are made from two state electronic devices like transistors. Unfortunately, we humans do not use binary number system. We are using decimal number system for out day to day tasks. Then how computer system performs arithmetic ? One obvious answer to this question is computer takes decimal from user, converts them into binary, process it and at the time of output it again converts binary into decimal.

But is there any other method is there ? Obviously Yes. There exists another alternative method of performing computation in the form of decimals but it requires that the decimal number has to be coded in the form of binary, so that computer can perform arithmetic operations on it. This method of representing each decimal digit in binary form is called Binary Coded Decimal (BCD).

In our decimal system, a number is made from digits (symbol). In decimal number system we use 10 symbols to represent a digit. If we use 4–bits (binary) then we have total 24=16 different combinations, which are enough to represent our 10 symbols. For that reason, in BCD system we represent each decimal digit of a number using a pair of 4–bits of binary. For example, a number 125.29 can be represented as : 0001 0010 0101. 0010 1001. Consider the following table in which we have given decimal, binary and BCD codes.

Decimal	Binary	BCD	Decimal	Binary	BCD
0	0	0000	**8**	1000	1000
1	1	0001	**9**	1001	1001
2	10	0010	**10**	1010	0001 0000
3	11	0011	**11**	1011	0001 0001
4	100	0100	**12**	1100	0001 0010
5	101	0101	**13**	1101	0001 0011
6	110	0110	**14**	1110	0001 0100
7	111	0111	**15**	1111	0001 0101

BCD takes more memory as you can see in the table. 15 is represented with the help of 4–bits in binary but takes 8–bits of memory in the BCD system. But because of its ease implementation and faster processing it is mostly used in computers.

- **Check Your Progress – 1 :**

1. BCD stands for ________.

   [A] Bits Coded Decimals [B] Binary Coded Decimals

   [C] Binary Computation Decimals [D] Binary Code of Data

2. How many bits are required to represent a digit in BCD system ?

   [A] 1 bit [B] 2 bits [C] 4 bits [D] 8 bits

3. $(19)_{10}$ can be written as ________ in BCD system.

   [A] 10011 [B] 0001 1011 [C] 10001 [D] 0001 1001

4.3 Representation of Alphabets :

We can represent any decimal number using binary or BCD. We can computer binary of any number or any digit of a number by dividing it by 2. We are dividing number or digit by 2 and collect the remainder. We are doing this process till the number or digit do not turns to 0. Finally, we write the remainders in reverse order to get the binary.

But what happed to alphabets ? Can we divide alphabet 'A' by 2 ? Obviously, the answer is No. Then how can we find a binary of alphabet ?

It is true that we can find binary or BCDs only of numbers, not the alphabets. Due to that reason, each symbol (alphabets, digits and special symbols like @, #, $ etc.) available on the keyboard has its own unique number. This set where each symbol of keyboard represents a unique number is called ASCII (American Standard Code for Information Interchange). This code use 7–bits to represent 128 characters. Now an extended ASCII is using 8–bits representation for alphabets on Micro–computers.

If you store 'a' then computer is storing 97 (a binary of 97) in the memory. 97 is the ASCII value for character 'a'. Similarly, the ASCII value of 'b' is 98 and so on. The ASCII value for 'A' is 65, 'B' is 66 and that way the ASCII value of 'Z' is 90. Similar to the alphabets, we can represent colours, musical notes etc. can easily store in the computer memory.

- **Check Your Progress – 2 :**

1. ASCII stands for ________.

   [A] American Standard Code for Information Interchange

   [B] American Scientific Code for Information Interchange

   [C] American Scientific Code for Interchanging Information

   [C] American Standard Code for Interchanging Information.

2. How many bits are used in extended ASCII to represent a symbol.

   [A] 1 bit [B] 2 bits [C] 7 bits [D] 8–bits

3. What is the ASCII value of 'A' ?

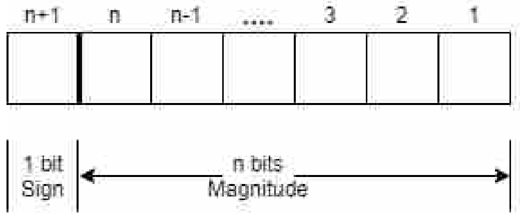
   [A] 65 [B] 97 [C] 60 [D] 90

4.4 Representation of Numeric Data :

Till now we have discussed different types of number system, BCD representation and how alphabets are represented in the computer memory. Now we will discuss how the numbers are actually stored in the memory so that computer system can perform arithmetic and scientific operations on it.

4.4.1 Representation of Integer Numbers :

To store n–bits numbers computer uses n + 1 bit of memory space. The actual number of n–bits is called Magnitude. One extra bit is used to represent sign (whether the number is positive or negative). This bit is known as a sign bit or a number. One sign bit is sufficient because a number can be positive (sign bit is 0) or negative (sign bit is 1). Representation of an integer number is shown in the figure given below :



Before going into the details that, how computer performs arithmetic operations such as addition and subtraction, let us discuss the term called 'Complements'. Complements can be used to represent negative numbers in our digital computers.

4.4.2 Complements :

For any number system of base 'k', there are two types of complements are there called k's complement and (k – 1)'s complement. For example, for the decimal system (base = 10) we have 10's complement and (10 – 1)'s = 9's complement. Let us discuss how can we find 9's and 10's complement for given number.

9's complement :

To find the 9's complement of a given number we need to subtract each digit of a number from 9 (the highest digit value). For example, if we want to find a 9's complement of decimal number 382 then :

9's complement of 382 :	9	9	9
	–3	–8	–2
	=6	1	7

10's complement :

So, we have seen that the to obtain 9's complement of a number 382, wee need to subtract 382 from 999. 617 is the 9's complement of a number 382. Now adding 1 to the 9's complement we will get 10's complement. Therefore the 10's complement of 382 will be 617 (9's complement of 382) + 1 = 618.

In binary number system, base is 2. So, we can find 2's complement and 1's complement for any given binary number.

1's complement :

1's Complement for the Binary numbers are similar to the 9's complement of Decimal number system. In 9's complement we have subtracted each digit of a decimal number from 9, similarly to find 1's complements each digit (bit) of binary number we need to subtract from 1 (the highest digit value). For example, if we want to find 1's complement of a binary number 0010 1101 then,

1's Complement of 0010 1101 :	1 1 1 1	1 1 1 1
	– 0 0 1 0	1 1 0 1
	1 1 0 1	0 0 1 0

Therefore, 1101 0010 is the complement of binary 0010 1101. If you see the 1's complement and the binary number, you can notice that 1's complement is nothing but each bit is inverted. That means for any binary number if we replace every 0 with 1 and every 1 with 0 you will get the 1's complement of that binary number.

2's complement :

2's complement of any binary number is nothing but adding 1 to the 1's complement. Recall 10's complement. In 10's complement we have added 1 to the 9's complement.

2's complement of a binary number 0010 1101 is :

```
1 1 0 1  0 0 1 0      [1's complement of binary 0010 1101]
              + 1
1 1 0 1  0 0 1 1      [2's complement of binary 0010 1101 ]
```

Is there any other direct method is there to find 2's complement directly without finding its 1's complement and then by adding 1 to it ? Obviously, the answer is Yes. To do this you need to find first 1 from the right side of the given binary number. From right to left you need to keep all bits same, and from first 1 onwards all bits have to be inverted (0 will becomes 1, and 1 will becomes 0).

To find the 2's complement directly of binary number 0010 1101, you need keep same underlined bits. In this example only one bit (last bit) 1 will remain as it is that is 1, and rest of the bit will be inverted. Therefore 2's complement will be : 1101 0011.

Example : Find 2's complement of binary number : 0110 1000.

To find 2's complement we will find first 1 from right to left. So, in this example last 4–bits 1000 will remail as it is and another 4–bits 0110 will be complemented to 1001. Therefore, 2's complement will be 1001 1000.

0110 **1**000 [From right to left all bits will remain same till first 1].
$\quad\;\;\longleftarrow$

0110 **1**000 [After first 1, all bits will be inverted to 0].
$\longleftarrow$

❑ **Check Your Progress – 3 :**

1. Find 10's complement of a decimal number 763.

   [A] 763 [B] 236 [C] 237 [D] 367

2 Find 2's complement of a binary number 0110 1010.

   [A] 0110 1011 [B] 1001 0101 [C] 1001 0110 [D] 1111 1111

3. Find 1's complement of a binary number 0110 1010

   [A] 0110 0110 [B] 1001 1011 [C] 1001 0101 [D] 0000 1111

4.5 Addition of Binary Numbers :

We know that when we input the decimal number in the computer system, the number will be stored in the form of binary into computer memory. Now one obvious question will come to your mind that how the computer will perform arithmetic operations like addition or subtraction on it ?

The answer for this question is there is some built–in circuitries are there called adder which can do the addition and subtraction of binary numbers. To understand how arithmetic addition will carried out by these circuitries memorise the following truth table of addition :

Input Bit : 1	Input Bit : 2	Sum Bit	Carry Bit
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

For example, if we want to do the addition of two 8–bit numbers 47 and 29 then,

Carry :		1	1	1	1	1	1	
Binary of 47 :	0	0	1	0	1	1	1	1
Binary of 29 :	0	0	0	1	1	1	0	1
Sum 47 + 29 :	0	1	0	0	1	1	0	0

Make sure, in binary 0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1, 1 + 1 = 10 (1 carry and sum bit 0), 1 + 1 + 1 = (10 + 1) = 11 (1 carry and sum bit is also 1).

So, the sum of (47) : 0010 1111 and (29) : 0001 1101 = 0100 1100. Now try to convert binary 0100 1100 into decimal you will get 0 + 64 + 0 + 0 + 8 + 4 + 0 + 0 = 76.

❑ **Check Your Progress – 4 :**

1. Addition of Binaries 0011 1010 and 0010 1011 is ________.

   [A] 110 0101 [B] 0110 0111 [C] 0110 0101 [D] 0110 1010

2. Addition of Binaries 0101 0100 and 0011 0110 is ________.

   [A] 1100 1010 [B] 1000 1010 [C] 1101 1011 [D] 1000 0101

4.6 Subtraction of Binary Numbers :

Subtraction of two binary numbers is quite interesting. Do you know computer use same circuitry for addition and subtraction ? Yes, computer use same circuitry to do addition and subtraction that is adder. Computer doesn't have circuit for subtraction. Now, you might think, how adder can be used to subtract two numbers ?

See, if you want to subtract 25 from 47, we write : 47–25. But computer understand this expression as : 47 + (– 25). Means computer understand –25 has to be added to number 47. Here, computer use the same circuit called adder. But now you think, how can we negate 25 ? To negate 25, computer uses 1's complement or 2's complement. For example, the binary of 25 is : 0001 1001. Then –25 can be represent by its 1's complement that is : 1110 0110 or 2's complement that is : 1110 0111.

Let us understand how computer use 1's complement and 2's complement to compute subtraction of two numbers.

4.6.1 Subtraction Using 1's Complement Method :

Let us take two numbers 25 and 30. We want to find 30–25. The computer will take this expression as : 30 + (– 25). Means binary of 30 is added to 1's complement of 25. Binary of 30 : 0001 1110 and Binary of 25 : 0001 1001

The leftmost bit for both the numbers are 0. Leftmost bit represents sign of a number (it also called sign bit). Sign bit of both the numbers are 0, that means both numbers are positive. The binary of the number –25 is 1's complement of 25.

Binary of – 25 : **1**110 0110

You may notice that the sign bit is 1. That means 1110 0110 represent negative number. Now, add 30 and –25 that means add 0001 1110 and 1110 0110.

```
Carry :    1 1 1 1  1 1
30 :       0 0 0 1  1 1 1 0
– 25 :     1 1 1 0  0 1 1 0      [1's complement of (25) 0001 1001.]
          ------------------
         1 0 0 0  0 0 1 0 0
                           1
          ------------------
5 :        0 0 0 0 0 1 0 1
```

You will get binary 0000 0100 and 1 as a carry. Add final carry to the binary 0000 0100 you will get 0000 0101. Check the binary obtained, it is of which decimal number ? Yes, it is 5. Similarly, if you want to compute 25 – 30 then, 25 – 30 = 25 + (– 30).

```
Carry :          1
25 :     0 0 0 1 1 0 0 1
– 30 :   1 1 1 0 0 0 0 1      [1's complement of (30) 0001 1110.]
        -----------------
– 5 :    1 1 1 1 1 0 1 0
```

In the result sign bit (Left most bit) is 1, that means the result negative. Once you know the result is negative, find 1's complement of the result. You will get 0000 0101. It is a binary of 5. Therefore, we can say that the result is negative and value is 5, so result is –5. Let us do (– 30) + (– 25).

```
           1 1
– 30 :     1 1 1 0 0 0 0 1      [1's complement of (30) 0001 1110.]
– 25 :     1 1 1 0 0 1 1 0      [1's complement of (25) 0001 1001.]
          -----------------
         1 1 1 0 0 0 1 1 1
                         1      [Add final carry to the result]
          -----------------
– 55 :     1 1 0 0 1 0 0 0      [Result after adding carry 1].
```

The sign bit (Leftmost bit) is 1. Therefore, the result is negative. If the result is negative the find the complement of the resultant binary, we get 0011 0111. It is a binary of 55 and we know that the result is negative so the answer is – 55.

❖ **Points to Remember :**

- To subtract to binary numbers, you need to add both binary numbers. For the first number you have to consider its binary and for second (negative) number you have to consider 1's complement.
- After addition you have to check sign bit. If sign bit is 1 that means the answer is negative in this case you need to find 1's complement of the resultant binary. Convert that 1's complement into decimal. Suppose the decimal value is X then the final value of the subtraction is – X.
- If sign bit is 0, then result of the subtraction is positive. In this case you need to find decimal value of the binary and it will be the answer of your subtraction.
- If after addition carry is produced then add that carry to the sum.

4.6.2 Subtraction Using 2's Complement Method : :

The problem with 1's complement method is, it has two representations for 0. To represent +0 you need to use binary 0000 0000, where as to represent –0 its 1's complement 1111 1111 is used. Now, we know that +0 and –0 are same. Therefore, subtraction by using 2's complement is more efficient method than of 1's complement method. In 2's complement method +0 is represented as 0000 0000, and –0 is nothing but 2's complement of it that is 0000 0000. Therefore, in 2's complement +0 and –0 will be represented only by a single binary that is 0000 0000.

In this method negative numbers will be represented as 2's complement. For example : If you want to compute 30 – 25 then we need to do : 30 + (– 25).

Binary of 30 : 0001 1110

Binary of 25 : 0001 1001

Carry :	1 1 1 1 1 1	
30 :	0 0 0 1 1 1 1 0	
– 25 :	1 1 1 0 0 1 1 1	[2's complement of (25) 0001 1001.]
5 :	**1** 0 0 0 0 0 1 0 1	

After addition 1 carry is produced, you need to discard that carry (recall in 1's complement method we have added the carry to the result). After discarding carry the resultant binary is 0000 0101. In this binary sign bit is 0, that means number is positive and if we convert this binary into decimal, we are getting 5. Therefore, the result of the subtraction is : + 5.

Now, suppose if we want to subtract 30 from 25 then : 25 – 30 = 25 + (–30).

Carry :		
25 :	0 0 0 1 1 0 0 1	
–30 :	1 1 1 0 0 0 1 0	[2's complement of (30) 0001 1110.]
	1 1 1 1 1 0 1 1	

After addition we are getting binary 1111 1011. In this binary sign bit is 1 therefore the result is negative. In this case find the 2's complement of the resultant binary. 2's complement of the binary 1111 1011 is 0000 0101 which is nothing by binary of 5. Therefore, the result is : – 5.

If we want to do (– 30) + (– 25) then,

Carry :	1 1 1 1	
– 25 :	1 1 1 0 0 1 1 1	[2's complement of (25) 0001 1001.]
– 30 :	1 1 1 0 0 0 1 0	[2's complement of (30) 0001 1110.]
	1 1 1 0 0 1 0 0 1	

After discarding carry, we get 1100 1001, in which sign bit is 1, so result is negative. If result is negative, we need to find 2's complement of the binary 1100 1001, which is 0011 0111. This is nothing but binary of 55. Therefore, the final answer of the subtraction is – 55.

- **Check Your Progress – 5 :**

1. To represent negative number in binary subtraction ________ is used.

   [A] 1's complement [B] 2's complement

   [C] Both A and B [D] None of the above

2. ________ is used in binary subtraction of two numbers.

   [A] Adder [B] Subtractor [C] Multiplicator [D] divisor

4.7 Let Us Sum Up :

In this unit, we :

- Discussed how numbers are presented in the memory
- Explained how alphabets and special symbols are presented in the memory.
- Learnt, how negative numbers are presented in the memory.
- Elaborated addition will be done by the system
- Talked about how subtraction is performed by the system.

4.8 Suggested Answers For Check Your Progress :

- **Check Your Progress 1 :**

  1. [B] 2. [C] 3. [D]

- **Check Your Progress 2 :**

  1. [A] 2. [D] 3. [A]

- **Check Your Progress 3 :**

  1. [C] 2. [C] 3. [C]

- **Check Your Progress 4 :**

  1. [C] 2. [B]

- **Check Your Progress 5 :**

  1. [C] 2. [A]

4.9 Glossary :

ASCII : American Standard Code for Information Interchange. A global standard to provide numeric value to all alphabets and special symbols.

BCD : Binary Coded Decimal. It is another way to represent number into the memory. Here instead of binary we store the binary of each digit of a number.

Complement : It is a method to represent negative numbers into the computer's memory.

4.10 Assignment :

1. What BCD ? Explain it in brief.
2. Explain 1's complement with an example.
3. Explain 2's complement with an example.

4.11 Activity :

Perform following binary additions :

[1] 28 + 35 [2] 17 + 29 [3] 45 + 19 [4] 21 + 34

Perform following binary subtraction using 1's complement method :

[1] 28 – 35 [2] 17 – 29 [3] 45 – 19 [4] 21 – 34

Perform following binary subtraction using 2's complement method :

[1] 28 – 35 [2] 17 – 29 [3] 45 – 19 [4] 21 – 34

4.12 Case Study :

- What happed if we do binary addition of 55 and 75. Are you get the correct answer ? If not try to find out the reason for this.

4.13 Further Readings :

1. Computer Fundamentals by P.K.Sinha and Priti Sinha.
2. Discovering Computers 2016 by Shelly Cashman Series. CENGAGE publications.
3. Computer Fundamentals by Pearl Software, Khanna Book Publishing.

BLOCK SUMMARY :

- Computer is a machine which can perform arithmetic and logical operations at very fast speed, it can store the data, process the data and retrieve the data as per user's requirement.
- Data is unstructured raw material and unstructured facts which provides necessary input to the computer system.
- Processed data is called Information.
- First generations machines are based on vacuum tubes. They are very costly, slow in speed consume more electricity and larger in size.
- Second generation machines are based on transistors. Transistors are two state device, which can be charged or discharged. Second generation computers are smaller in size, less–costly and more powerful than first generation machines.
- Third generation computers are more powerful and based on IC (Integrated Circuits).
- Fourth generation computers are based on VLSI technology, they are compact less–costly and more powerful machines.
- Computers can be classified into Micro–computers, Mini–Computers, Mainframes, and Super–Computers.
- Desktops and Laptops are Micro–Computers. Super–Computers are very costly, Most powerful and very large size machines.
- Machine can understand only language of 1's and 0's. It is called a binary language.
- Assembly language uses mnemonic code instead of strings of binaries.
- Assembly codes are translated in machine–code by Assembler. Assembly language is more convenient language than binary.
- C–Language, FORTRAN, COBOL etc. are High–Level languages. Which are easier to program. They use compiler to translate high–level codes into machine language.
- There are 4 number systems are there. Binary, Octal, Decimal and Hexa–Decimal.
- Binary language includes only two symbols 0 and 1. It is also known as machine language. The base of the binary number system is 2.
- Decimal number system is that which we are using in our daily life. There are 10 symbols are used in decimal number system therefore the base of the decimal number system is 10.
- The base of the Octal number system is 8 and base of Hexa–Decimal number system is 16.
- Any type of data is stored in the computer's memory is stored in the form of Binary.

- Each alphabets and special symbols are available on the keyboard has unique number called ASCII. ASCII stands for American Standard Code for Information Interchange. It used 7 bits. Extended version of ASCII uses 8–bits to represent any character.
- To represent negative numbers 1's complement or 2's complement methods are used.
- In 1's complement method there are two different representations are there for –0 and +0.
- Whereas, in 2's complement system, only one representation is there to represent +0 and –0.
- 2's complement system is more effective that 1's complement system.
- To do addition and subtraction same circuit is used called Adder.
- To do the subtraction negative numbers are considered in the form of 1's complement or 2's complement form.

BLOCK ASSIGNMENT :

❖ **Short Answer Questions :**

(1) Define term 'Computer'

(2) Differentiate data and information

(3) Give the name of the technology used in First Generation, Second Generation and Third Generation computers

(4) List names of computer classification

(5) List named of different types of number system

(6) List all symbols used in Hexa–decimal number system

(7) What is base for octal number system ? Why ?

❖ **Long Questions :**

(1) Draw and explain computer organization chart

(2) Explain computer classification in brief

(3) List and explain different types of number system in detail

(4) List and explain different types of programming languages in detail

(5) Explain how to convert decimal number into binary and binary number into decimal with example.

(6) Explain how to convert decimal number into hexa–decimal and hexa–decimal number into decimal with example.

❖ **Enrolment No. :** ☐

1. How many hours did you need for studying the units ?

Unit No.	1	2	3	4
No. of Hrs.				

2. Please give your reactions to the following items based on your reading of the block :

Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	☐	☐	☐	☐	______
Language and Style	☐	☐	☐	☐	______
Illustration used (Diagram, tables etc)	☐	☐	☐	☐	______
Conceptual Clarity	☐	☐	☐	☐	______
Check your progress Quest	☐	☐	☐	☐	______
Feed back to CYP Question	☐	☐	☐	☐	______

3. Any other Comments

....

....

....

....

....

....

....

....