INTRODUCTION

In the modern world of electronics, the term **Digital** is generally associated with a computer because the term **Digital** is derived from the way computers perform operation, by counting digits. For many years, the application of digital electronics was only in the computer system. But now-a-days, digital electronics is used in many other applications. Following are some of the examples in which **Digital electronics** is heavily used.

* Industrial process control
* Military system
* Television
* Communication system
* Medical equipment
* Radar
* Navigation

Signal

**Signal** can be defined as a physical quantity, which contains some information. It is a function of one or more than one independent variables. Signals are of two types.

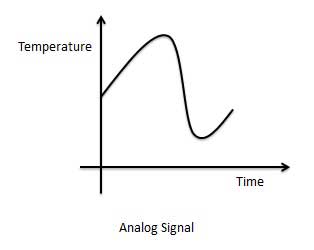
* Analog Signal
* Digital Signal

Analog Signal

An **analog signal** is defined as the signal having continuous values. Analog signal can have infinite number of different values. In real world scenario, most of the things observed in nature are analog. Examples of the analog signals are following.

* Temperature
* Pressure
* Distance
* Sound
* Voltage
* Current
* Power

Graphical representation of Analog Signal (Temperature)



The circuits that process the analog signals are called as analog circuits or system. Examples of the analog system are following.

* Filter
* Amplifiers
* Television receiver
* Motor speed controller

Disadvantage of Analog Systems

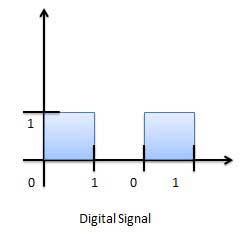
* Less accuracy
* Less versatility
* More noise effect
* More distortion
* More effect of weather

Digital Signal

A **digital signal** is defined as the signal which has only a finite number of distinct values. Digital signals are not continuous signals. In the digital electronic calculator, the input is given with the help of switches. This input is converted into electrical signal which have two discrete values or levels. One of these may be called low level and another is called high level. The signal will always be one of the two levels. This type of signal is called digital signal. Examples of the digital signal are following.

* Binary Signal
* Octal Signal
* Hexadecimal Signal

Graphical representation of the Digital Signal (Binary)



The circuits that process the digital signals are called digital systems or digital circuits. Examples of the digital systems are following.

* Registers
* Flip-flop
* Counters
* Microprocessors

Advantage of Digital Systems

* More accuracy
* More versatility
* Less distortion
* Easy communicate
* Possible storage of information

Comparison of Analog and Digital Signal

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Analog Signal** | **Digital Signal** |
| 1 | Analog signal has infinite values. | Digital signal has a finite number of values. |
| 2 | Analog signal has a continuous nature. | Digital signal has a discrete nature. |
| 3 | Analog signal is generated by transducers and signal generators. | Digital signal is generated by A to D converter. |
| 4 | Example of analog signal − sine wave, triangular waves. | Example of digital signal − binary signal. |

Decimal Number System

The number system that we use in our day-to-day life is the decimal number system. Decimal number system has base 10 as it uses 10 digits from 0 to 9. In decimal number system, the successive positions to the left of the decimal point represents units, tens, hundreds, thousands and so on.

Each position represents a specific power of the base (10). For example, the decimal number 1234 consists of the digit 4 in the units position, 3 in the tens position, 2 in the hundreds position, and 1 in the thousands position, and its value can be written as

(1×1000) + (2×100) + (3×10) + (4×l)

(1×103) + (2×102) + (3×101) + (4×l00)

1000 + 200 + 30 + 1

1234

As a computer programmer or an IT professional, you should understand the following number systems which are frequently used in computers.

|  |  |
| --- | --- |
| **S.N.** | **Number System & Description** |
| 1 | **Binary Number System**  Base 2. Digits used: 0, 1 |
| 2 | **Octal Number System**  Base 8. Digits used: 0 to 7 |
| 3 | **Hexa Decimal Number System**  Base 16. Digits used: 0 to 9, Letters used: A- F |

Binary Number System

Characteristics

* Uses two digits, 0 and 1.
* Also called base 2 number system
* Each position in a binary number represents a 0 power of the base (2). Example: 20
* Last position in a binary number represents an x power of the base (2). Example: 2x where x represents the last position - 1.

Example

Binary Number: 101012

Calculating Decimal Equivalent −

|  |  |  |
| --- | --- | --- |
| **Step** | **Binary Number** | **Decimal Number** |
| Step 1 | 101012 | ((1 × 24) + (0 × 23) + (1 × 22) + (0 × 21) + (1 × 20))10 |
| Step 2 | 101012 | (16 + 0 + 4 + 0 + 1)10 |
| Step 3 | 101012 | 2110 |

**Note:** 101012 is normally written as 10101.

Octal Number System

Characteristics

* Uses eight digits, 0,1,2,3,4,5,6,7.
* Also called base 8 number system
* Each position in an octal number represents a 0 power of the base (8). Example: 80
* Last position in an octal number represents an x power of the base (8). Example: 8x where x represents the last position - 1.

Example

Octal Number − 125708

Calculating Decimal Equivalent −

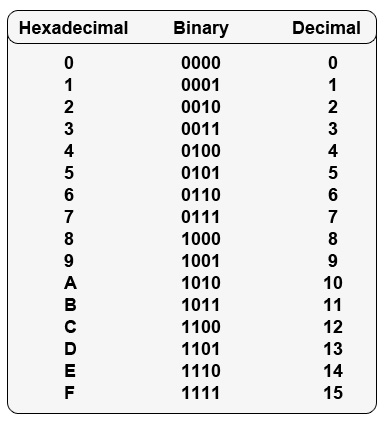
|  |  |  |
| --- | --- | --- |
| **Step** | **Octal Number** | **Decimal Number** |
| Step 1 | 125708 | ((1 × 84) + (2 × 83) + (5 × 82) + (7 × 81) + (0 × 80))10 |
| Step 2 | 125708 | (4096 + 1024 + 320 + 56 + 0)10 |
| Step 3 | 125708 | 549610 |

**Note:** 125708 is normally written as 12570.

Hexadecimal Number System

Characteristics

* Uses 10 digits and 6 letters, 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.
* Letters represents numbers starting from 10. A = 10, B = 11, C = 12, D = 13, E = 14, F = 15.
* Also called base 16 number system.
* Each position in a hexadecimal number represents a 0 power of the base (16). Example 160.
* Last position in a hexadecimal number represents an x power of the base (16). Example 16x where x represents the last position - 1.



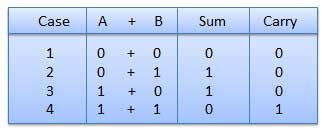
BCD code: 8421

BINARY ARTHMETIC

Binary arithmetic is essential part of all the digital computers and many other digital systems.

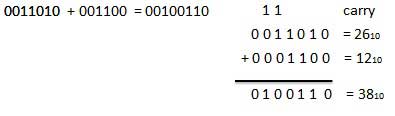
Binary Addition

It is a key for binary subtraction, multiplication, division. There are four rules of binary addition.



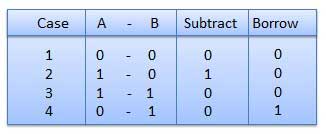
In fourth case, a binary addition is creating a sum of (1 + 1 = 10) i.e. 0 is written in the given column and a carry of 1 over to the next column.

Example − Addition

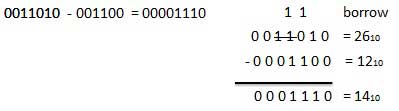


Binary Subtraction

**Subtraction and Borrow**, these two words will be used very frequently for the binary subtraction. There are four rules of binary subtraction.

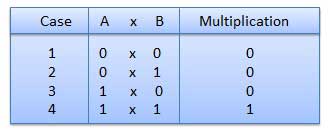


Example − Subtraction

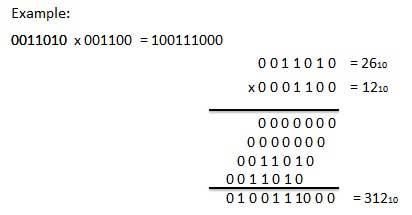


Binary Multiplication

Binary multiplication is similar to decimal multiplication. It is simpler than decimal multiplication because only 0s and 1s are involved. There are four rules of binary multiplication.



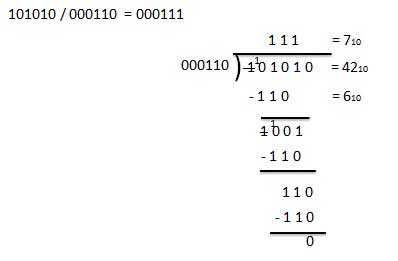
Example − Multiplication



Binary Division

Binary division is similar to decimal division. It is called as the long division procedure.

Example − Division

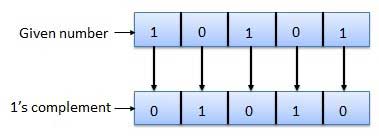


Binary system complements

As the binary system has base r = 2. So the two types of complements for the binary system are 2's complement and 1's complement.

1's complement

The 1's complement of a number is found by changing all 1's to 0's and all 0's to 1's. This is called as taking complement or 1's complement. Example of 1's Complement is as follows.

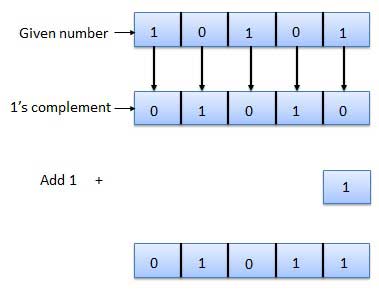


2's complement

The 2's complement of binary number is obtained by adding 1 to the Least Significant Bit (LSB) of 1's complement of the number.

2's complement = 1's complement + 1

Example of 2's Complement is as follows.



# **Addition and Subtraction using 2's complement**

In our previous section, we learned how we could perform arithmetic operations such as addition and subtraction using 1's complement. In this section, we will learn to perform these operations using 2's complement.

## Addition using 2's complement

There are three different cases possible when we add two binary numbers using 2's complement, which is as follows:

**Case 1: Addition of the positive number with a negative number when the positive number has a greater magnitude.**

Initially find the 2's complement of the given negative number. Sum up with the given positive number. If we get the end-around carry 1 then the number will be a positive number and the carry bit will be discarded and remaining bits are the final result.

**Example: 1101 and -1001**

1. First, find the 2's complement of the negative number 1001. So, for finding 2's complement, change all 0 to 1 and all 1 to 0 or find the 1's complement of the number 1001. The 1's complement of the number 1001 is 0110, and add 1 to the LSB of the result 0110. So the 2's complement of number 1001 is 0110+1=0111
2. Add both the numbers, i.e., 1101 and 0111;  
   1101+0111=1  0100
3. By adding both numbers, we get the end-around carry 1. We discard the end-around carry. So, the addition of both numbers is 0100.

**Case 2: Adding of the positive value with a negative value when the negative number has a higher magnitude.**

Initially, add a positive value with the 2's complement value of the negative number. Here, no end-around carry is found. So, we take the 2's complement of the result to get the final result.

#### **Note: The resultant is a negative value.**

**Example: 1101 and -1110**

1. First, find the 2's complement of the negative number 1110. So, for finding 2's complement, add 1 to the LSB of its 1's complement value 0001.  
   0001+1=0010
2. Add both the numbers, i.e., 1101 and 0010;  
   1101+0010= 1111
3. Find the 2's complement of the result 1111 that is the final result. So, the 2's complement of the result 1111 is 0001, and add a negative sign before the number so that we can identify that it is a negative number.

# **BCD or Binary Coded Decimal**

* It is a form of binary encoding where each digit in a decimal number is represented in the form of four bits of binary.

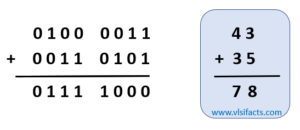
*. Convert (123)10 in BCD*

*From the truth table above,   
1 -> 0001   
2 -> 0010   
3 -> 0011   
thus, BCD becomes -> 0001 0010 0011*

**Steps of BCD Addition**

* **Step 1:** Add the two BCD numbers using the [rules for binary addition](http://www.vlsifacts.com/rules-of-binary-arithmetic/).
* **Step 2:** If a 4-bit sum is equal to or less than 9, it is a valid BCD number.
* **Step 3:** If a 4-bit sum is greater than 9 or if a carry-out of the 4-bit group is generated, it is an invalid result. Add 6 (0110) to the 4-bit sum in order to skip the six invalid BCD code words and return the code to 8421. If a carry results when 6 is added, simply add the carry to the next 4-bit group.

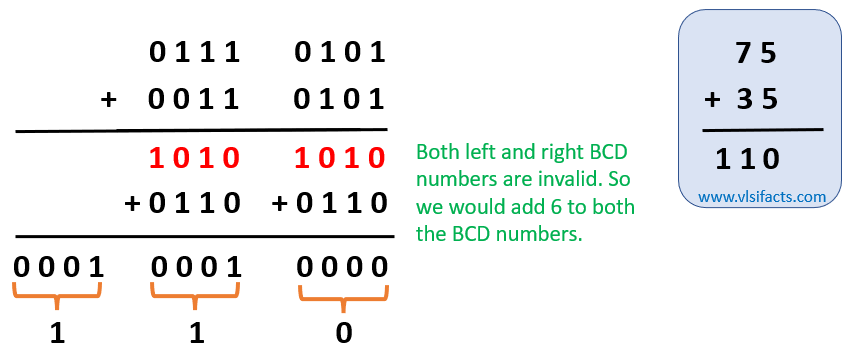
Example 1: Find the sum of the BCD numbers 01000011 + 00110101



In the above example, all the 4-bit BCD additions generate valid BCD numbers, which means less than 9. So, the final correct result is 7810 = 01111000BCD.

Example 2 : Find the sum of the BCD numbers 01110101 + 00110101

Solution:



In the above example, both the BCD code addition generated result larger than 9, so invalid. To get the correct result, we added 6 (01102) to both the invalid sum. Notice that the carry generated from the left group is forwarded to the right group. The final correct result is 11010 = 000100010000BCD.