**LECTURE 01**

**Introduction to Analog And Digital signals and systems**

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. These signals continuously vary with time. Examples of the analog signals are following.

* Temperature, Pressure, Voltage, Current, Sine wave etc.

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

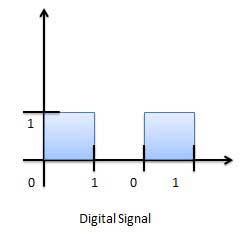
* Less accuracy
* Less versatility
* More noise effect
* More distortion
* More effect of weather

**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. 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

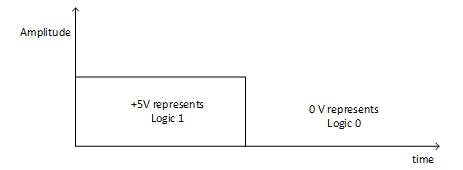
Comparison of Analog and Digital Signal

|  |  |  |
| --- | --- | --- |
| **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. |

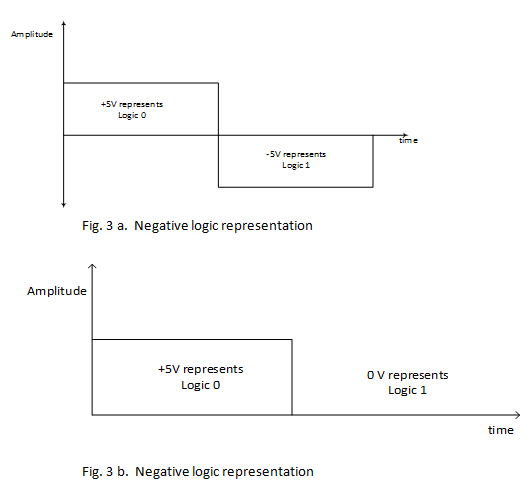
**Positive and Negative Logic**

There are two types of representations used in digital systems, the positive logic and the negative logic representations.

In positive logic representation Bit 1 represents Logic high and Bit 0 represent a Logic low as shown. High is represented by +5 Volts and low is represented by -5 Volts or 0 Volts.



In Negative logic representation Bit 1 represents logic low and Bit 0 represents logic high as shown in Fig. In terms of voltage level, bit 1 can be represented as +5V and bit 0 can be represented as 0 V or -5 Volts



**Number System**

A number system can be formed when few symbols called digits are placed side by side and these symbols represent different values depending on the position they occupy in the number. A number have two parts one is integer part and other fractional part and both parts are separated by a point called radix point. It is written as

**(N)b = an an-1……. a3 a2 a1 a0 . a-1 a-2 a-3 a-4 a-5 ……a-m**

N = number

B = base of the system

an = MSB

a0 = LSB

A value of each digit in a number can be determined using

* The digit
* The position of the digit in the number
* The base of the number system (where base is defined as the total number of digits available in the number system).

**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

|  |  |
| --- | --- |
| **.** |  |
| 1 | **Binary Number System**  Base 2. Digits used: 0,1. No of bits used = 2  No of combination of 0 and 1 = 2n = 22 = 4  Decimal. Binary  0 0 0  1 0 1  2 1 0  3 1 1  **Bit-** the smallest unit of data is defined as a single bit.  **Nibble-** combination of four bits.  **Byte-** combination of eight bits. |
| 2 | **Octal Number System**  Base 8. Digits used: 0 to 7. No of bits used = 3  No of combination of 0 and 1 = 2n = 23 = 8  Octal Number System | Electrical4U |
| 3 | **Hexa Decimal Number System ( alphanumeric number system)**  Base 16. Digits used: 0 to 9, Letters used: A- F, no of bits used = 4  No of combination of 0 and 1 = 2n = 24 = 16  Binary Numbering System to Hexadecimal Conversion Table |

**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.

Example −

Hexadecimal Number: 19FDE16

Calculating Decimal Equivalent −

|  |  |  |
| --- | --- | --- |
| **Step** | **Hexadecimal Number** | **Decimal Number** |
| Step 1 | 19FDE16 | ((1 × 164) + (9 × 163) + (F × 162) + (D × 161) + (E × 160))10 |
| Step 2 | 19FDE16 | ((1 × 164) + (9 × 163) + (15 × 162) + (13 × 161) + (14 × 160))10 |
| Step 3 | 19FDE16 | (65536 + 36864 + 3840 + 208 + 14)10 |
| Step 4 | 19FDE16 | 10646210 |

**Note −** 19FDE16 is normally written as 19FDE.

There are many methods or techniques which can be used to convert numbers from one base to another. We'll demonstrate here the following −

* Decimal to Binary System
* Binary System to Decimal
* Other Base System to Non-Decimal
* Shortcut method − Binary to Octal
* Shortcut method − Octal to Binary
* Shortcut method − Binary to Hexadecimal
* Shortcut method − Hexadecimal to Binary

**Decimal to Other Binary System**

Steps

* **Step 1** − Divide the decimal number to be converted by the value of the new base.
* **Step 2** − Get the remainder from Step 1 as the rightmost digit (least significant digit) of new base number.
* **Step 3** − Divide the quotient of the previous divide by the new base.
* **Step 4** − Record the remainder from Step 3 as the next digit (to the left) of the new base number.

Repeat Steps 3 and 4, getting remainders from right to left, until the quotient becomes zero in Step 3.

The last remainder thus obtained will be the Most Significant Digit (MSD) of the new base number.

**Example −**

**Decimal Number: 2910 to binary by repeated division method**

Calculating Binary Equivalent −

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Operation** | **Result** | **Remainder** |
| Step 1 | 29 / 2 | 14 | 1 |
| Step 2 | 14 / 2 | 7 | 0 |
| Step 3 | 7 / 2 | 3 | 1 |
| Step 4 | 3 / 2 | 1 | 1 |
| Step 5 | 1 / 2 | 0 | 1 |

The first remainder becomes the Least Significant Digit (LSD) and the last remainder becomes the Most Significant Digit (MSD).

Decimal Number − 2910 = Binary Number − 111012.

Other Base System to Decimal System

Steps

* **Step 1** − Determine the column (positional) value of each digit (this depends on the position of the digit and the base of the number system).
* **Step 2** − Multiply the obtained column values (in Step 1) by the digits in the corresponding columns.
* **Step 3** − Sum the products calculated in Step 2. The total is the equivalent value in decimal.

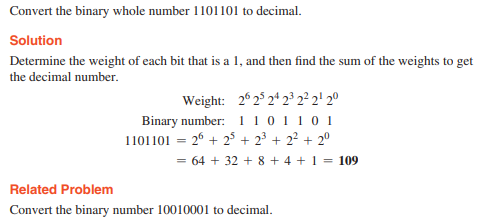
Example

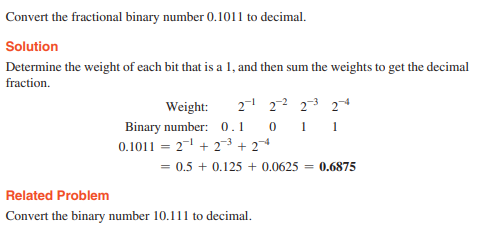
Binary Number − 111012

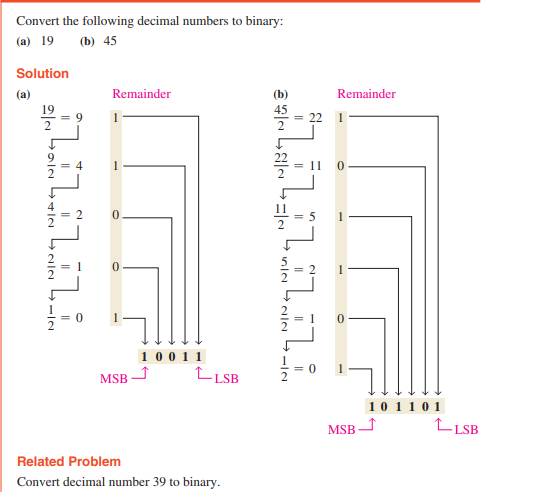
Calculating Decimal Equivalent −

|  |  |  |
| --- | --- | --- |
| **Step** | **Binary Number** | **Decimal Number** |
| Step 1 | 111012 | ((1 × 24) + (1 × 23) + (1 × 22) + (0 × 21) + (1 × 20))10 |
| Step 2 | 111012 | (16 + 8 + 4 + 0 + 1)10 |
| Step 3 | 111012 | 2910 |

Binary Number − 111012 = Decimal Number − 2910







**(96.24)10= ( )2 direct method ( )8**

**9 = 1001 96.24 = (1001 0110 . 0010 0100)2**

**6 = 0110 first change to binary**

**Form the group of 3 bits**

**010 010 110 = (226)8**

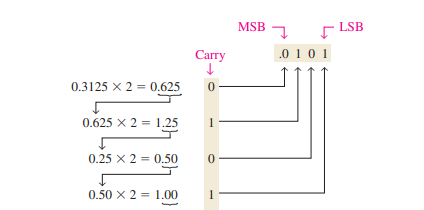
**. 001 0 01 000 = (.110)8 = (226.110)8 = (96.24)10 =( )16**

**2 = 0010**

**4 = 0100**

**Convection of fractional decimal to binary**

Repeated multiplication by 2



**Octal Number − 258 to decimal Equivalent −**

Step 1 − Convert to Decimal

|  |  |  |
| --- | --- | --- |
| **Step** | **Octal Number** | **Decimal Number** |
| Step 1 | 258 | ((2 × 81) + (5 × 80))10 |
| Step 2 | 258 | (16 + 5 )10 |
| Step 3 | 258 | 2110 |

Octal Number − 258 = Decimal Number − 2110

Step 2 − Convert Decimal to Binary

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Operation** | **Result** | **Remainder** |
| Step 1 | 21 / 2 | 10 | 1 |
| Step 2 | 10 / 2 | 5 | 0 |
| Step 3 | 5 / 2 | 2 | 1 |
| Step 4 | 2 / 2 | 1 | 0 |
| Step 5 | 1 / 2 | 0 | 1 |

Decimal Number − 2110 = Binary Number − 101012

Octal Number − 258 = Binary Number − 101012

Shortcut method - Binary to Octal

Steps

* **Step 1** − Divide the binary digits into groups of three (starting from the right).
* **Step 2** − Convert each group of three binary digits to one octal digit.

Example

Binary Number − 101012

Calculating Octal Equivalent −

|  |  |  |
| --- | --- | --- |
| **Step** | **Binary Number** | **Octal Number** |
| Step 1 | 101012 | 010 101 |
| Step 2 | 101012 | 28 58 |
| Step 3 | 101012 | 258 |

Binary Number − 101012 = Octal Number − 258

Shortcut method - Octal to Binary

Steps

* **Step 1** − Convert each octal digit to a 3 digit binary number (the octal digits may be treated as decimal for this conversion).
* **Step 2** − Combine all the resulting binary groups (of 3 digits each) into a single binary number.

Example

Octal Number − 258

Calculating Binary Equivalent −

|  |  |  |
| --- | --- | --- |
| **Step** | **Octal Number** | **Binary Number** |
| Step 1 | 258 | 210 510 |
| Step 2 | 258 | 0102 1012 |
| Step 3 | 258 | 0101012 |

Octal Number − 258 = Binary Number − 101012

Shortcut method - Binary to Hexadecimal

Steps

* **Step 1** − Divide the binary digits into groups of four (starting from the right).
* **Step 2** − Convert each group of four binary digits to one hexadecimal symbol.

Example

Binary Number − 101012

Calculating hexadecimal Equivalent −

|  |  |  |
| --- | --- | --- |
| **Step** | **Binary Number** | **Hexadecimal Number** |
| Step 1 | 101012 | 0001 0101 |
| Step 2 | 101012 | 110 510 |
| Step 3 | 101012 | 1516 |

Binary Number − 101012 = Hexadecimal Number − 1516

Shortcut method - Hexadecimal to Binary

Steps

* **Step 1** − Convert each hexadecimal digit to a 4 digit binary number (the hexadecimal digits may be treated as decimal for this conversion).
* **Step 2** − Combine all the resulting binary groups (of 4 digits each) into a single binary number.

Example

Hexadecimal Number − 1516

Calculating Binary Equivalent −

|  |  |  |
| --- | --- | --- |
| **Step** | **Hexadecimal Number** | **Binary Number** |
| Step 1 | 1516 | 110 510 |
| Step 2 | 1516 | 00012 01012 |
| Step 3 | 1516 | 000101012 |

Hexadecimal Number − 1516 = Binary Number − 101012

**Summary**

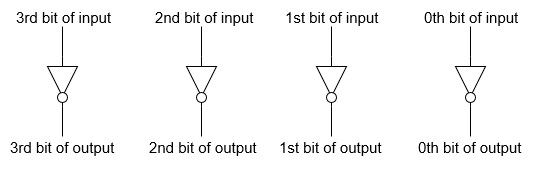
* 1. **(111101100)2 = ( )8**
  2. **(725.67)8 = ( )2**
  3. **(1100010011011)2 = ( )16**
  4. **(615)8 = ( )16**
  5. **(25B)16 = ( )8**

**Complements**

**In a digital system the subtraction operation is performed by using complement method**

* 1. **1’s complement**
  2. **2’s complement**
* **1’s Complement of a Binary Number:**

It is a simple algorithm to convert a binary number into 1’s complement. To get 1’s complement of a binary number, simply invert the given number.  You can simply implement logic circuit using only NOT gate for each bit of Binary number input.



**Example-1:** Find 1’s complement of binary number **10101110**.

Simply invert each bit of given binary number, so 1’s complement of given number will be 01010001.

**Example-2:** Find 1’s complement of binary number 10001.001.

Simply invert each bit of given binary number, so 1’s complement of given number will be 01110.110.

**Subtractions by 1’s Complement:**

The algorithm to subtract two binary number using 1’s complement is explained as following below:

* Take 1’s complement of the subtrahend
* Add with minuend
* If the result of above addition has carry bit 1, then add it to the least significant bit (LSB) of given result
* If there is no carry bit 1, then take 1’s complement of the result which will be negative

Note that subtrahend is number that to be subtracted from the another number, i.e., minuend.

**Example (Case-1: When Carry bit 1):** Evaluate 10101 - 00101

Take 1’s complement of subtrahend 00101, which will be 11010, then add both of these.

So, 10101 + 11010 =1 01111 . Since, there is carry bit 1, so add this to the LSB of given result, i.e., 01111+1=10000 which is the answer.

## 1011 -0100 = ( 0111)2

## 4 – 9 = 0100 – 1001= 0100 + 0110 = 1010 = - 0101= -5

## 9-4 = 1001 -0100 = 1001+1011=1 0100 = 0100 +1=0101

## 2’s Complement of a Binary Number

There is a simple algorithm to convert a binary number into 2’s complement. To get 2’s complement of a binary number, simply invert the given number and add 1 to the least significant bit (LSB) of given result.

**Example-1** − Find 2’s complement of binary number 10101110.

Simply invert each bit of given binary number, which will be 01010001. Then add 1 to the LSB of this result, i.e., 01010001+1=01010010 which is answer.

**Example-2** − Find 2’s complement of binary number 10001.001.

Simply invert each bit of given binary number, which will be 01110.110 Then add 1 to the LSB of this result, i.e., 01110.110+1=01110.111 which is answer.

# **Subtraction by 2’s Complement**

Top of Form

Bottom of Form

With the help of subtraction by 2’s complement method we can easily subtract two binary numbers.

**The operation is carried out by means of the following steps:**

(i) At first, 2’s complement of the subtrahend is found.

(ii) Then it is added to the minuend.

(iii) If the final carry over of the sum is 1, it is dropped and the result is positive.

(iv) If there is no carry over, the two’s complement of the sum will be the result and it is negative.

**Evaluate:**

**(i) 110110 - 10110**

**Solution:**

The numbers of bits in the subtrahend is 5 while that of minuend is 6. We make the number of bits in the subtrahend equal to that of minuend by taking a `0’ in the sixth place of the subtrahend.

Now, 2’s complement of 010110 is (101101 + 1) i.e.101010. Adding this with the minuend.

                          1     1 0 1 1 0      Minuend  
  
                          1     0 1 0 1 0      2’s complement of subtrahend  
  
   Carry over 1       1     0 0 0 0 0      Result of addition

After dropping the carry over we get the result of subtraction to be 100000.

**(ii) 10110 – 11010**

**Solution:**

2’s complement of 11010 is (00101 + 1) i.e. 00110. Hence

                                  Minued -          1 0 1 1 0  
  
   2’s complement of subtrahend -          0 0 1 1 0  
  
                    Result of addition -          1 1 1 0 0

As there is no carry over, the result of subtraction is negative and is obtained by writing the 2’s complement of 11100 i.e.(00011 + 1) or 00100.

Hence the difference is – 100.

**(iii) 1010.11 – 1001.01**

**Solution:**

2’s complement of 1001.01 is 0110.11. Hence

                                Minued -          1 0 1 0 . 1 1  
  
2’s complement of subtrahend -           0 1 1 0 . 1 1  
  
                           Carry over      1     0 0 0 1 . 1 0

After dropping the carry over we get the result of subtraction as 1.10.

**(iv) 10100.01 – 11011.10**

**Solution:**

2’s complement of 11011.10 is 00100.10. Hence

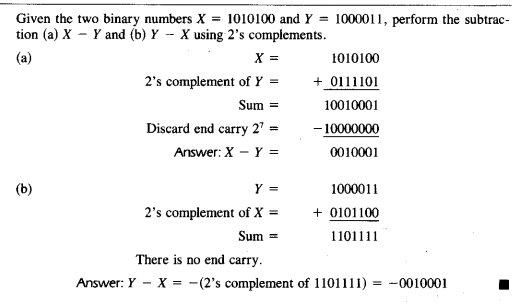
                                  Minued -          1 0 1 0 0 . 0 1  
  
   2’s complement of subtrahend -          0 1 1 0 0 . 1 0  
  
                    Result of addition -          1 1 0 0 0 . 1 1

As there is no carry over the result of subtraction is negative and is obtained by writing the 2’s complement of 11000.11.

Hence the required result is – 00111.01.

4 – 9 =0100 – 1001 =0100 +(0110+1) =0100 + 0111 = 01011

Take 2’ complement of 1011 = 0101

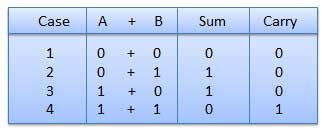


# **Binary Arithmetic**

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

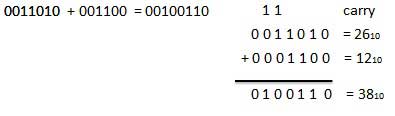
## Binary Addition

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



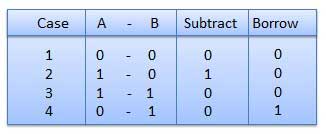
In fourth case, a binary addition is creating a sum of (1 + 1 = 10) i.e. 0 is written in the given column and a carry of 1 over to the next column.

### Example − Addition

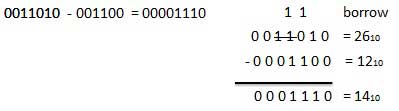


## Binary Subtraction

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

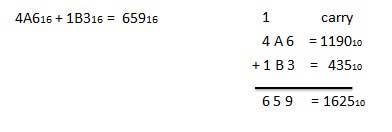


### Example − Subtraction



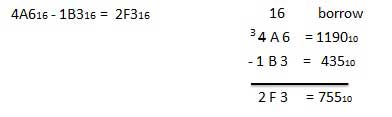
## Hexadecimal Addition

### Example − Addition



## Hexadecimal Subtraction

### Example - Subtraction



**Assignment**

1. Convert the binary number to decimal, octal and hexadecimal number.

**(11011011.100101)2**

1. Convert **(2AC5.D)16** to decimal, octal and binary.
2. Using 2’s complement, perform **(42)10 – (68)10**
3. Add **(DF)16 + (AC)16 and (C3)16 – (0B)16**