# DIGITAL ELECTRONICS

**ARUN RANA**
*Assistant Professor*
Department of Electronics & Communication Engineering
Doon Valley Institute of Engineering & Technology
Karnal, Haryana

(An ISO 9001:2008 Certified Company)

**Vayu Education of India**
**2/25, Ansari Road, Darya Ganj, New Delhi-110 002**

# DIGITAL ELECTRONICS



**ISBN:** 978-93-83137-89-3

**First Edition:** 2013

**Price:** ₹ 180/-



*Printed & bound in India*

*Published by:*

**(An ISO 9001:2008 Certified Company)**

**VAYU EDUCATION OF INDIA**

2/25, Ansari Road, Darya Ganj, New Delhi-110 002
Ph.: 91-11-43526600, 41564445
Fax: 91-11-41564440
E-mail: vei@veiindia.com
Web: www.veiindia.com

# CONTENTS

# Chapter-1

# NUMBER SYSTEMS AND CODES

## 1.1 INTRODUCTION

Digital Systems are used extensively in computation and data processing, control systems, communications, and measurement. Because digital systems are capable of greater accuracy and reliability than analog systems, many tasks formerly done by analog systems are now being performed digitally. In a digital system, the physical quantities or signals can assume only discrete values, while in analog systems the physical quantities or signals may vary continuously over a specified range For example, the output voltage of a digital system might be constrained to take on only two values such as 0 volts and 5 volfs, while the output voltage from an analog system might be allowed to assume any value in the range –10 volts to +10 volts.

The design of digital systems may be divided roughly into three parts-system design, logic design, and circuit design. System design involves breaking the overall system into subsystems and specifying the characteristics of each subsystem. For example, system design of a digital computer could involve specifying the number and type of memory units, arithmetic units, and input output devices as well as the interconnection and control of these subsystems. Logic design involves determining how to interconnect basic logic building blocks to perform a specific function. Circuit design involves specifying the interconnection of specific components such as resistors diodes, and transistors to form a gate, Flip-Flop or other logic building block.

Starting from a given problem statement, the first step in designing a combinational logic network is to derive a table or algebraic logic equations which describe the network outputs as a function of the network inputs. In order to design an economical network to realize these output functions, the logic equations which describe the network outputs generally must be simplified. Algebraic method for the simplification and other simplification methods (K-map and Tabular method) are used.

The switching devices used in digital system are generally two-state devices that is, the output assume only two different discrete values. Examples of switching devices are relays, diodes and transistors. A relay can assume two states - closed or open - depending on whether power is applied to the coil or not. A diode can be in a conducting

state or a non-conducting state. A transistor can be in a cut-off or saturated state, with a corresponding high or low output voltage. Because the outputs of most switching devices assume only two different values, it is natural to use binary numbers internally in digital systems A digital computer manipulates discrete elements of information and that these elements are represented in the binary form operands used for calculations may be expressed in the binary number system. Other discrete elements, including the decimal digits, are represented in binary codes. Data processing is carried out by means of binary logic element using binary signals. For this reason various number systems will be discussed first before proceeding with the design of switching networks.

## 1.2 DECIMAL NUMBER SYSTEM

It is more familiar number system, has a base of 10 implies that it contains ten unique symbols (or digits) 0, 1,2, 3, 4, 5, 6, 7, 8, 9. It is a position-value system, means that the value attached to a symbol (or digit) depends on its position with respect to the decimal point. In this number system any number can be represented by the use of these ten distinct symbols. These distinct symbols are called as digits.

Consider a decimal number 5623 using digits 5, 6, 2 and 3.

Hence $5623 = 5 \times 10^3 + 6 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$.

The left most digit (5) in above number having the greatest positional weight out of all digits present in the number is called most significant digit (MSD) and the right most digit having the least positional weight is called as Least significant digit (LSD). The numbers right to the decimal point (fractional numbers) have weights which are negative powers of base 10.

Generally the value of any mixed decimal number

$$(m_n....m_2m_1m_0.\ m_{-1}\ m_{-2}\ ...m_{-p})$$

can be written as

$$(m_n \times 10^n) + (m_{n-1} \times 10^{n-1}) +....+(m_1 \times 10^1) + (m_0 \times 10^0).(m_{-1} \times 10^{-1}) + (m_{-2} \times 10^{-2}) + ....+(m_{-p} \times 10^{-p})$$

For example consider a mixed decimal number 6523.46. It can be written as $6523.46 = 6 \times 10^3 + 5 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 6 \times 10^{-2}$.

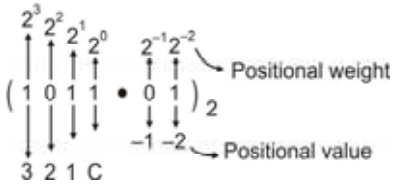
The first digit to the left of decimal point has a weight of unity ($10^0$). The second digit to the left of decimal point has a weight of $10^1$, the third digit has a weight of $10^2$ and so on. The first digit to the right of the decimal point has a weight of $10^{-1}$, the second digit has weight $10^{-2}$ and so on. For expressing quantities exceeding Nine, two or more digits are used/and the position of each digit within the number indicates the magnitude it represents.

## 1.3 BINARY NUMBER SYSTEM

It is a positional weighted system having base 2. It is less complicated than the decimal number system because it is composed of only two digits. A binary digit is called a bit. A binary number consists of a sequence of bits each of which is either 0 or 1. The first

bit to the left of the binary point has a weight of $2^0$ and that column is called the unit column. The second bit to the left has a weight of $2^1$ and so on. The weight of each bit position is one power of 2 greater than the weight of the position to its immediate left. The first bit to the right of binary point has a weight of $2^{-1}$ and it is said to be 1/2 column and so on.

For example, the number $(1011.01)_2$ is a mixed binary number.




In general, a binary number with an integer part of $(m + 1)$ bits and a fractional part of $n$ bits can be written as

$a_m\ a_{m-1}\ a_{m-2}\ ...a_2a_1a_0.a_1a_{-2}\ ...a_{-n}$.

Decimal equivalent of above number is

$(a_m \times 2^m)$ ....+ $(a_2 \times 2^2) + (a_1 \times 2^1) + (a_0 \times 2^0) + (a_{-1} \times 2^{-1}) + (a_{-2} \times 2^{-2}) + ...(a_{-n} \times 2^{-n})$

In general, decimal equivalent of any number

$a_m\ a_{m-1}\ a_{m-2}\ ...\ a_2a_1a_0.\ a_{-1}\ a_{-2}\ ...\ a_{-n}$ in any number system of base $p$ is given by

$(a_m \times p^m)\ ...\ +\ ...\ +\ (a_1 \times p^1) + (a_0 \times p^0) + (a_{-1} \times p^{-1}) + (a_{-2} \times p^{-2}) + ...(a_{-n} \times p^{-n})$

Some terms like bit, nibble and byte are used in binary number system. Bit is used for single binary digit, A binary number with four bits is called nibble. When binary number has eight bits then it is called a byte.

## 1.4 ADVANTAGE OF BINARY NUMBERS

1. Binary numbers have numerous advantages in digital computers, because the switching circuits used in these computers use two state devices, so these two states can be represented by the symbol 0 and 1 respectively.

2. A card with punched holes can also be used to represent the binary numbers. Binary data and instructions, are stored in the card with the help of a card punching machine which punches the card in a pre-arranged code.

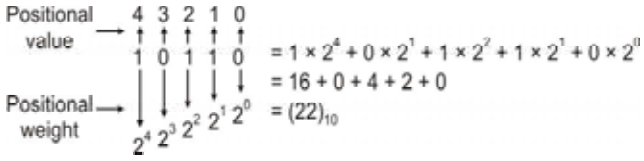
3. A magnetic tape may also be used as a two state device. Some points of magnetic tape are magnetised while leaving other points unmagnetised. A row of the points represent data or instructions. So a magnetic tape reel can be used to store thousands of binary data and instructions.

## 1.5 BINARY TO DECIMAL CONVERSION

In binary number system each binary digit (bit) carries a certain weight based on its position relative to the least significant bit. Any binary number can be converted to their decimal equivalents simply by adding the products of each bit and its positional weight.

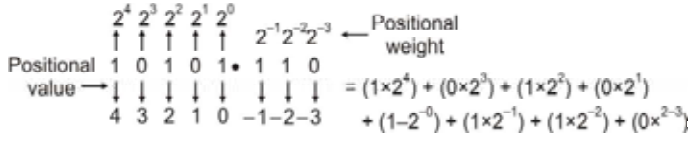
Example 1.1. Convert $(10110)_2$ to decimal.

Solution: Given binary number is




Example 1.2. Convert $(10101.110)_2$ to decimal.

Solution:




$= 16 + 0 + 4 + 0 + 1 + ½ + ¼ + 0$

$= 16 + 0 + 4 + 0 + 1 + 50 + 25 + 0$

$= (21.75)_{10}$

Example 1.3. Convert $(1001.0101)_2$ to decimal.

Sol. Binary number = 1001.0101

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

$= 8 . 0 . 0 . 1 . 0 . 1/4 . 0 . \frac{1}{16}$

$= 9 + 25 + .0625$

$= (9.3125)_{10}$

Example 1.4. Convert $(0.11010)_2$ to decimal.

Sol. $0.11010 = 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5}$

$= ½ + ¼ + 0 + \frac{1}{16} . 0$

$= .50 + 25 + .0625$

$= (.8125)_{10}$

Example 1.5. Convert $(0000)_2$ to decimal.

Sol. $(0000)_2 = 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$

$= 0 + 0 + 0 + 0$

$= (0)_{10}$

## 1.6 DECIMAL TO BINARY CONVERSION

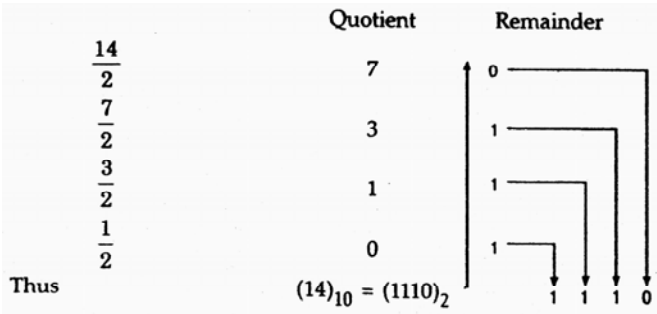
Any decimal number can be converted into its equivalent binary number by double-dabble method. For decimal integer number the conversion is obtained by continuous

division by 2 and keeping track of remainders, and the decimal fraction is converted to binary fraction y successive multiplication by 2. In the successive division by-2 method, the given decimal integer is divided by 2 till the quotient is zero. The last remainder is the MSB. The remainders read from bottom to top give the equivalent binary number. In the successive multiplication by –2 method, the given decimal fraction and the subsequent decimal fractions are successively multiplied by 2 till the fraction part of the product is zero, or being repeated itself or till the desired accuracy is obtained. Integers obtained, read from top to bottom give the equivalent binary fraction.

In general this method can be used for converting a decimal number to an equivalent number in any base system.

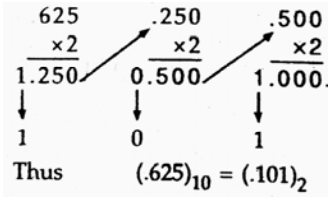
Example 1.6. Convert $(14)_{10}$ to an equivalent binary number.

Solution:




Example 1.7. Convert $(.625)_{10}$ to binary number.

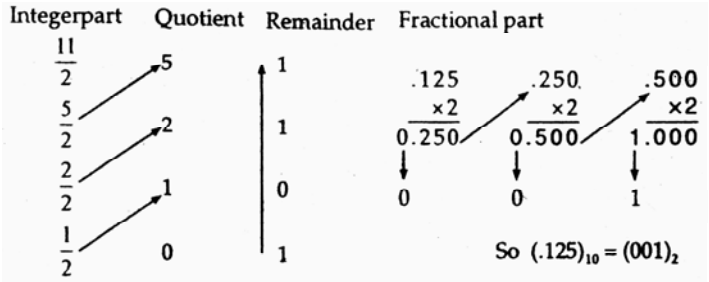
Sol.




Example 1.8. Express the following decimal numbers in the binary form.

(*a*) 11.125 (*b*) 25.1250

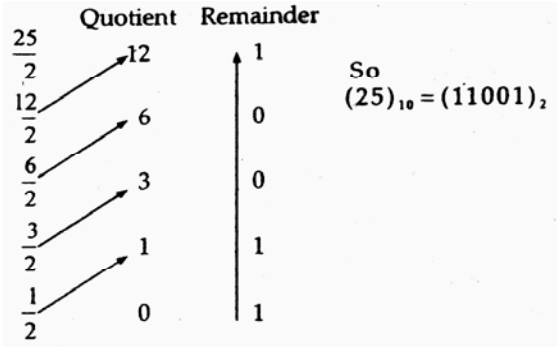
Solution:



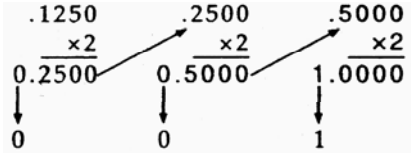
So $(11)_{10} = (1011)_2$

Therefore $(11.25)_{10} = (1011.0010)_2$

(*b*) Integer Part




fractional part $(.1250)_{10}$




So $(.1250)_{10} = (.001)2$

Therefore $(25.1250)_{10} = (11001.001)_2$.

## 1.7 BINARY ARITHMETIC

We are familiar with the arithmetic operations such as addition, division subtraction and multiplication. Similar operations can be performed on binary numbers. Binary arithmetic is simpler than decimal arithmetic due to involving only two digits 0 and 1.

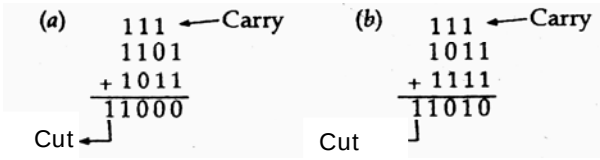
Binary Addition: The binary numbers can be added by following rules.

$$\begin{array}{cccc} 0 & 0 & 1 & 1 \\ .0 & .1 & .0 & .1 \\ \hline 0 & 1 & 1 & 10 \end{array}$$ (that is sum = 0 with a carry 1)

(that is sum = 0 with a carry 1)

Example 1.9. Add the binary numbers (*a*) 1101 and 1011 (*b*) 1011 and 1111.

Sol.




(*a*) In the first column 1 + 1 = 0 with a carry of 1 to the next column.

In the second column 0 + 1 + 1 = 0 with a carry of 1.

The the third column 1 + 0 + 1 = 0 with a carry of 1.

In the fourth column 1 + 1 + 1 = 1 with a carry of 1.

Example 1.10. Add the binary numbers 1011.110 and 1101.101.

Sol.

```
      1111  ←——Carry
      1011 • 110
    + 1101 • 101
    ------------
     11001 • 011
Cut ↵
```

From these examples we observe that

(*i*)If the number of 1's to be added in a column is even then the sum bit is 0 and if the number of 1's to be added in a column is odd then the sum bit is 1.

(*ii*)Every pair of 1's in a column produces a carry 1 to be added to the next higher column.

## 1.8. BINARY SUBTRACTION

Binary number can be subtracted in a manner to that in decimal subtraction. The rules of binary subtraction are given as

$0 - 0 = 0$

$1 - 0 = 1$

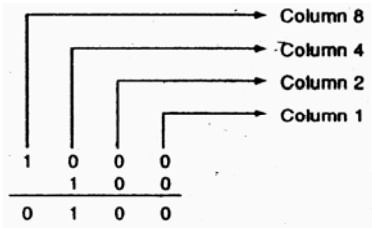
$1 - 1 = 0$

$0 - 1 = 1$ with a borrow of 1.

When the borrow is equal to 1, the number is to be subtracted from the next higher binary bit as it is done in decimal subtraction.

Example 1.11. Subtract $(100)_2$ from $(1000)_2$.

Solution:



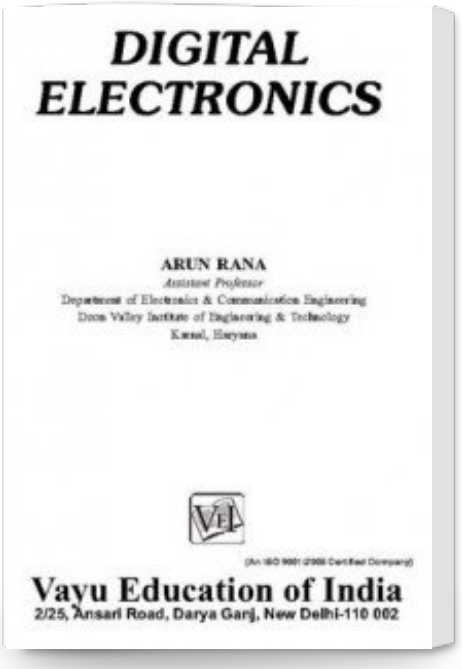

In the 4's column 1 cannot be subtracted from 0, so a 1 must be borrowed from the 8's column making 8's column 0. The 1 borrowed from the 8's column becomes $(10)_2$ in the 4's column. So 10 – 1 = 1 in the 4's column.

Example 1.12. Subtract $(110.101)_2$ from (1000.111).

Solution:

```
  8 4 2 1      2^-1 2^-2 2^-3  ← Column numbers
  1 0 0 0  •   1 1 1          → Minuend
- 1 1 0  •   1 0 1            → Subtrahend
  ---------------
  0 0 1 0  •   0 1 0          → Difference
```

# Digital Electronics By Arun Rana




Publisher : Vayu Education ISBN : 9789383137893 Author : Arun Rana