

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

Copyright ©VAYU EDUCATION OF INDIA

ISBN: 978-93-83137-89-3

First Edition: 2013

Price: ₹ 180/-

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the Author and Publisher.

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

I. NUMBER SYSTEMS AND CODES.....	I
1.1 Introduction	1
1.2 Decimal Number System.....	2
1.3 Binary Number System.....	2
1.4 Advantage of Binary Numbers	3
1.5 Binary To Decimal Conversion.....	3
1.6 Decimal To Binary Conversion.....	4
1.7 Binary Arithmetic.....	6
1.8 Binary Subtraction.....	7
1.9 Binary multiplication.....	8
1.10 Binary Division	9
1.11 Octal Number System	10
1.12 Octal to Decimal Conversion	10
1.13 Decimal to Octal Conversion	11
1.14 Octal to Binary Conversion	12
1.15 Binary to Octal Conversion	12
1.16 Octal Arithmetic.....	13
1.17 Octal Subtraction	14
1.18 Octal Multiplication.....	14
1.19 Octal Division	15
1.20 Hexadecimal Number System.....	16
1.21 Hexadecimal to Decimal Conversion.....	16
1.22 Decimal to Hex Conversion	17
1.23 Hexadecimal to Binary Conversion	19
1.24 Binary to Hexadecimal Conversion	19
1.25 Hexadecimal to octal Conversion	20
1.26 Octal to Hexadecimal Conversion	20
1.27 Hexadecimal Arithmetics.....	21
1.28 Hexadecimal Subtraction.....	22
1.29 Hexadecimal Multiplication	24
1.30 Hexadecimal Division.....	25
1.31 Complements	25
1.32 Subtraction with Complements	26
1.33 Subtraction with (r-1)'s Complement	27
1.34 Signed Binary Numbers.....	28
1.35 Addition of Signed Binary Numbers.....	29

1.36	Subtraction of Signed Binary Numbers.....	29
1.37	Codes	30
1.38	Binary Codes	30
1.39	Decimal Codes	31
1.40	Binary Coded Decimal (BCD) Code.....	32
1.41	BCD Addition.....	32
1.42	Bcd subtraction	33
1.43	Gray code	33
1.44	Binary-To-Gray Conversion	34
1.45	Gray to binary conversion.....	34
1.46	Excess-3 (XS-3) code	35
1.47	Subtraction in excess-3 code	36
1.48	Error Detecting code	36
2.	BOOLEAN ALGEBRA AND LOGIC GATES.....	40
2.1	Introduction	40
2.2	Axioms of Boolean Algebra.....	40
2.3	Laws of Boolean Algebra.....	41
2.4	Demorgan's Theorem.....	44
2.5	Implementation of Boolean Expression With Logic	46
2.6	Complement of A Function.....	47
2.7	Minterms And Maxterms	48
2.8	Standard Sum of Product Form	49
2.9	Standard Product of Sum (Pos) Form	51
2.10	Conversion Between Canonical Forms	52
2.11	Logic Gates.....	53
2.12	And Gate.....	54
2.13	Or Gate.....	54
2.14	Not Gate or Inverter.....	55
2.15	Universal Building Blocks (Universal Gates)	56
2.16	Nand Gate.....	57
2.17	Nor Gate	58
2.18	Exclusive-or (XOR) Gate.....	59
2.19	Exclusive-nor (x-nor) Gate.....	59
2.20	Summary of Gates	60
3.	SIMPLIFICATION OF BOOLEAN FUNCTIONS	64
3.1	Introduction	64
3.2	K-map of Two-three and Four Variables.....	64
3.3	Representation of Truth Table on k-map.....	65
3.4	Simplification of Boolean Functions Using k-map.....	66
3.5	Other Forms of Karnaugh maps-(veitch diagram).....	70
3.6	Don't Care Conditions	71
3.7	Five Variable k-map	72

3.8	six Variable k-map	74
3.9	Tabular Method—(Quine-mc Cluskey Method)	75
3.10	Cyclic Prime Implicant Chart	80
3.11	Example of Simplification of A Function With Don't Care	81

4. COMBINATIONAL LOGIC CIRCUITS 84

4.1	Introduction	84
4.2	Adders	84
4.3	Half Adder	85
4.4	Full Adder	86
4.5	Adder circuits	88
4.6	Subtractors	91
4.6.1	Half Subtractor	92
4.6.2	Full Subtractor	93
4.6.3	Parallel Subtractor	94
4.6.4	Subtraction Using Full Adders	94
4.7.4	Bit adder-subtractor	95
4.8	Magnitude comparator	96
4.9	Decoders	97
4.9.1	Basic Binary decoder	97
4.9.2	3-to-8 Line Decoder	98
4.9.3.	4-Line to 16-Line Decoder	99
4.10	Encoders	100
4.10.1	Octal to Binary Encoder	101
4.10.2	Decimal-to-BCD encoder	102
4.11	Multiplexers (Data Selectors)	103
4.11.1	Multiplexer Tree	105
4.11.2	Application of Multiplexers	106
4.12	Demultiplexers (Data Distributors)	106
4.12.1	De-Multiplexer Tree	107
4.13	Read Only Memories	108
4.14	Types of Roms	111
4.15	Programmable Logic Devices	112
4.16	Programmable Logic Arrays (PLA)	112
4.17	programmable Array Logic (PAL)	115

5. SEQUENTIAL LOGIC CIRCUITS 119

5.1	Introduction	119
5.2	Gate Delays and Timing Diagram	120
5.3	1-bit memory unit	122
5.4	Nor Gate Latch	123
5.5	S-R Flip flop	124
5.6	D-Flip flop	125
5.7	J-K Flip Flop	126
5.8	T-Flip-Flop	127

5.9	Master-slave Flip-flop	128
5.10	Master-slave J-k Flip-flop.....	129
5.11	Edge triggered flip-flops.....	129
5.12	Asynchronous Inputs	130
5.13	Excitation Table.....	131
5.14	Conversion of Flip-flops	132
5.15	Conversion of S-R Flip-Flop to A J-K Flip-Flop	133
5.16	Conversion of S-r Flip-flop To A J-k Flip-flop	134
5.17	conversion from S-R flip-flop to D flip-flop	135
5.18	Conversion From J-k Flip-flop To D Flip-flop	135
5.19	Conversion From D Flip-flop To J-k Flip-flop	135
5.20	Analysis of Clocked Sequential Circuits	136
5.21	Analysis With J-k and Other Flip-flops.....	138
5.22	State Assignment	140
5.23	Registers	141
5.24	4 Bit Shift Register	141
5.25	Shift Registers.....	141
5.26	Shift Right Register	144
5.27	Counters.....	148
5.28	Asynchronous (Ripple Or Serial) Counter	149
5.29	Synchronous (Parallel) Counter	151
5.30	Asynchronous Sequential Circuits.....	153
5.31	Analysis of Asynchronous Sequential Circuits	155
5.32	Flowtable.....	157
5.33	Race Conditions and Cycles	158

6. LOGIC FAMILIES161

6.1	Introduction	161
6.1.1	Bipolar Logic Families.....	161
6.1.2	MOS Logic Families or Unipolar Logic Families.....	162
6.2	Characteristics of Digital ICs	162
6.2.1	Speed of Operation	162
6.2.2	Power Dissipation	163
6.2.3	Fan-in	163
6.2.4	Fan-out	163
6.2.5	Noise Immunity.....	163
6.2.6	Operating Temperature	163
6.2.7	Current and Voltage Parameter	163
6.3	Current Sourcing and Current Sinking Logic	164
6.4	Switching Circuits	165
6.4.1	The Semiconductor Diode.....	165
6.4.2	Bipolar Junction Transistors (BJTs)	165
6.4.3	Metal Oxide Semiconductor Field Effect Transistor (MOSFETs) ...	166
6.5	Resistor Transistor Logic (RTL)	167

6.5.1	Input Output Characteristics	167
6.5.2	Fan-in and Fan-out	168
6.5.3	Noise Margins.....	168
6.5.4	Propagation Delay Time	168
6.5.5	Wired Logic.....	169
6.6	Direct Coupled Transistor LogiC (DCTL)	169
6.6.1	Input Output Characteristics	170
6.7	Diode Transistor Logic (DTL)	170
6.7.1	Input Output Characteristics	171
6.7.2	Fan-out.....	172
6.7.3	Wired Logic.....	173
6.8	High threshold logic (HTL)	173
6.8.1	Input Output Characteristics	174
6.9	Transistor-transistor Logic (TTL OR T2L).....	174
6.9.1	TTL Circuit Output Configurations.....	176
6.9.2	Input Output Characteristics	178
6.10	Emitter Coupled Logic (Ecl).....	179
6.11	Mos Circuits.....	180
6.12	Pmos Logic.....	181
6.13	Nmos Logic.....	182
6.14	Cmos Logic (Complementary Mos Logic).....	183
6.14.1	CMOS Inverter	183
6.14.2	CMOS NAND Gate	184
6.14.3	CMOS NOR Gate	185
6.15	Bi-CMOS LOGIC	186
6.15.1	Bi CMOS Inverter	186
6.15.2	Bi CMOS NAND Gate.....	186
6.15.3	Bi CMOS NOR Gate.....	187
6.16	Interfacing or Compatibility	188
6.16.1	Interfacing CMOS with TTL.....	188
6.16.2	TTL Driving CMOS.....	189
6.16.3	CMOS Driving TTL	190
6.17	comparison of Logic families	191

7. MEMORIES192

7.1	Introduction.....	192
7.2	Classification of Memories.....	193
7.2.1	Registers, Main Memory and Secondary Memory.....	193
7.2.2	Sequential Access Memory and Random Access Memory.....	193
7.2.3	Static and Dynamic Memory	193
7.2.4	Volatile and Non-volatile Memory.....	194
7.2.5	Magnetic and Semiconductor Memory.....	194
7.3	Basic Memory Operation.....	194
7.4	Random Access Memory (Ram)	195
7.4.1	Static RAM.....	196

- 7.4.2 Dynamic RAM (DRAM)200
- 7.5 Read only memory (Rom)201
 - 7.5.1 Types of ROM203
- 7.6 Programmable Logic Devices (Pld)208
 - 7.6.1 Programmable Logic Array (PLA)210
 - 7.6.2 Programmable Array Logic (PAL)211
- 7.7 Field Programmable Gate Array (Fpga).....213

- Index 215-216

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 volts, 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

$$(m_n \dots m_2 m_1 m_0 . m_{-1} m_{-2} \dots m_{-p})$$

can be written as

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

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

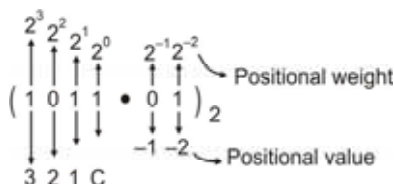
The first digit to the left of decimal point has a weight of unity (10^0). The second digit to the left of decimal point has a weight of 10^1 , the third digit has a weight of 10^2 and so on. The first digit to the right of the decimal point has a weight of 10^{-1} , the second digit has weight 10^{-2} 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

$$a_m a_{m-1} a_{m-2} \dots a_2 a_1 a_0 . a_{-1} a_{-2} \dots a_{-n}$$

Decimal equivalent of above number is

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

In general, decimal equivalent of any number

$$a_m a_{m-1} a_{m-2} \dots a_2 a_1 a_0 . a_{-1} a_{-2} \dots a_{-n} \text{ in any number system of base } p \text{ is given by}$$

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

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

$$\begin{array}{rcccccc}
 \text{Positional value} & \rightarrow & 4 & 3 & 2 & 1 & 0 \\
 & & 1 & 0 & 1 & 1 & 0 \\
 & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 \text{Positional weight} & & 2^4 & 2^3 & 2^2 & 2^1 & 2^0
 \end{array}$$

$$\begin{aligned}
 &= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\
 &= 16 + 0 + 4 + 2 + 0 \\
 &= (22)_{10}
 \end{aligned}$$

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

Solution:

$$\begin{array}{rcccccccc}
 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & 2^{-1} & 2^{-2} & 2^{-3} & \leftarrow \text{Positional weight} \\
 \text{Positional value} & \rightarrow & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 & 4 & 3 & 2 & 1 & 0 & -1 & -2 & -3
 \end{array}$$

$$\begin{aligned}
 &= (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) \\
 &\quad + (1 \times 2^0) + (1 \times 2^{-1}) + (1 \times 2^{-2}) + (0 \times 2^{-3}) \\
 &= 16 + 0 + 4 + 0 + 1 + \frac{1}{2} + \frac{1}{4} + 0 \\
 &= 16 + 0 + 4 + 0 + 1 + 50 + 25 + 0 \\
 &= (21.75)_{10}
 \end{aligned}$$

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

Sol. Binary number = 1001.0101

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

$$= 8 + 0 + 0 + 1 + 0 + \frac{1}{4} + 0 + \frac{1}{16}$$

$$= 9 + 25 + .0625$$

$$= (9.3125)_{10}$$

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

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

$$= \frac{1}{2} + \frac{1}{4} + 0 + \frac{1}{16} + 0$$

$$= .50 + .25 + .0625$$

$$= (.8125)_{10}$$

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

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

$$= 0 + 0 + 0 + 0$$

$$= (0)_{10}$$

1.6 DECIMAL TO BINARY CONVERSION

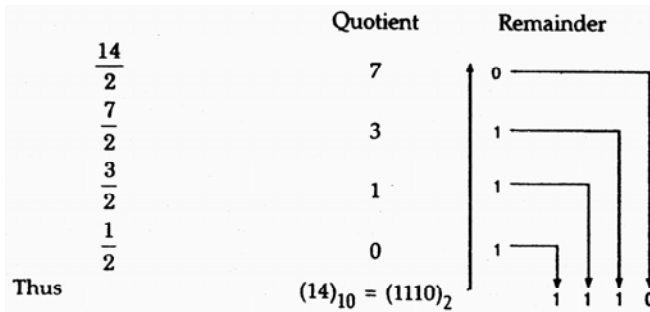
Any decimal number can be converted into its equivalent binary number by double-dabble method. For decimal integer number the conversion is obtained by continuous

division by 2 and keeping track of remainders, and the decimal fraction is converted to binary fraction by 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.

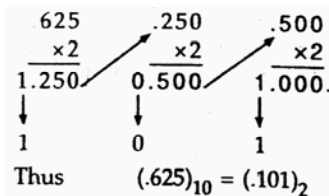
Example 1.6. Convert $(14)_{10}$ to an equivalent binary number.

Solution:



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

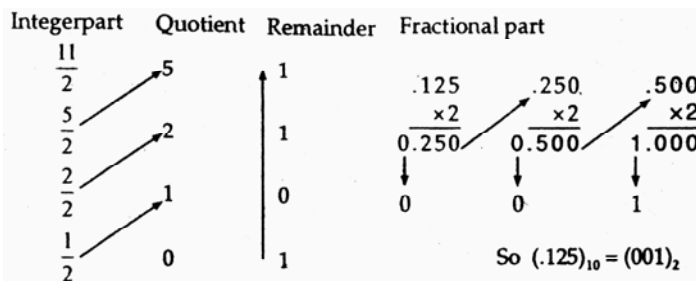
Sol.



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

(a) 11.125 (b) 25.1250

Solution:



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

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

(b) Integer Part

Quotient	Remainder
$\frac{25}{2} \rightarrow 12$	1
$\frac{12}{2} \rightarrow 6$	0
$\frac{6}{2} \rightarrow 3$	0
$\frac{3}{2} \rightarrow 1$	1
$\frac{1}{2} \rightarrow 0$	1

$$\text{So } (25)_{10} = (11001)_2$$

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

$.1250 \times 2 \rightarrow 0.2500$	$.2500 \times 2 \rightarrow 0.5000$	$.5000 \times 2 \rightarrow 1.0000$
0	0	1

$$\text{So } (.1250)_{10} = (.001)_2$$

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

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.

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

(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) $\begin{array}{r} 111 \leftarrow \text{Carry} \\ 1101 \\ + 1011 \\ \hline 11000 \\ \text{Cut} \leftarrow \end{array}$	(b) $\begin{array}{r} 111 \leftarrow \text{Carry} \\ 1011 \\ + 1111 \\ \hline 11010 \\ \text{Cut} \leftarrow \end{array}$
--	--

(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.

$$\begin{array}{r}
 \begin{array}{cccc}
 1 & 1 & 1 & 1 \\
 1 & 0 & 1 & 1 \\
 + & 1 & 1 & 0 & 1 \\
 \hline
 1 & 1 & 0 & 0 & 1
 \end{array}
 \end{array}$$

← Carry

• 110

• 101

• 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

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$0 - 1 = 1 \text{ with a borrow of } 1.$$

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:

$$\begin{array}{r}
 \begin{array}{cccc}
 1 & 0 & 0 & 0 \\
 - & 1 & 0 & 0 \\
 \hline
 0 & 1 & 0 & 0
 \end{array}
 \end{array}$$

Column 8

Column 4

Column 2

Column 1

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 $(1000.111)_2$.

Solution:

$$\begin{array}{r}
 \begin{array}{cccc}
 8 & 4 & 2 & 1 \\
 1 & 0 & 0 & 0 \\
 - & 1 & 1 & 0 \\
 \hline
 0 & 0 & 1 & 0
 \end{array}
 \end{array}$$

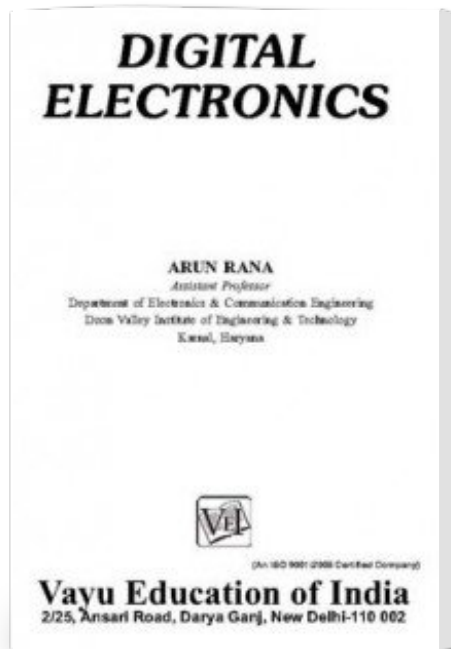
← Column numbers

← Minuend

← Subtrahend

← Difference

Digital Electronics By Arun Rana



Publisher : **Vayu Education**

ISBN : **9789383137893**

Author : **Arun Rana**

Type the URL : <http://www.kopykitab.com/product/3253>



Get this eBook