Electronics Engineering

Digital Circuits

Comprehensive Theory

with **Solved Examples** and **Practice Questions**

MADE EASY
Publications

MADE EASY Publications Pvt. Ltd.

Corporate Office: 44-A/4, Kalu Sarai (Near Hauz Khas Metro Station), New Delhi-110016
E-mail: infomep@madeeasy.in
Contact: 9021300500
Visit us at: www.madeeasypublications.org

Digital Circuits

First Edition: 2015
Second Edition: 2016
Third Edition: 2017
Fourth Edition: 2018
Fifth Edition: 2019
Sixth Edition : 2020
Seventh Edition : 2021
Eighth Edition : 2022
Ninth Edition : 2023

Contents

Digital Circuits

CHAPTER 1

Number Systems

Introduction

Electronic systems are of two types:

(*i*) Analog systems (*ii*) Digital systems

Analog systems are those systems in which voltage and current variations are continuous through the given range and they can take any value within the given specified range, whereas a digital system is one in which the voltage level assumes finite number of distinct values. In all modern digital circuits there are just two discrete voltage level.

Digital circuits are often called switching circuits, because the voltage levels in a digital circuit are assumed to be switched from one value to another instantaneously. Digital circuits are also called logic circuits, because every digital circuit obeys a certain set of logical rules.

Digital systems are extensively used in control systems, communication and measurement, computation and data processing, digital audio and video equipments, etc.

Advantages of Digital Systems

Digital systems have number of advantages over analog systems which are summarized below:

1. Ease of Design

The digital circuits having two voltage levels, OFF and ON or LOW and HIGH, are easier to design in comparison with analog circuits in which signals have numerical significance ; so their design is more complicated.

2. Greater Accuracy and Precision

Digital systems are more accurate and precise than analog systems because they can be easily expanded to handle more digits by adding more switching circuits.

3. Information Storage is Easy

There are different types of semiconductor memories having large capacity, which can store digital data.

4. Digital Systems are More Versatile

It is easy to design digital systems whose operation is controlled by a set of stored instructions called program. However in analog systems, the available options for programming is limited.

5. Digital Systems are Less Affected by Noise

The effect of noise in analog system is more. Since in analog systems the exact values of voltages are important. In digital system noise is not critical because only the range of values is important.

6. Digital Systems are More Reliable

As compared to analog systems, digital systems are more reliable.

Limitations of Digital System

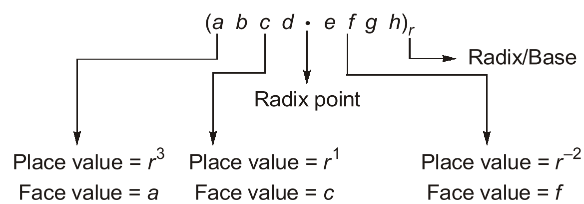
(*i*) The real world is mainly analog.

(*ii*) Human does not understand the digital data.

1.1 Digital Number Systems (Positional Weight System)

Many number systems are used in digital technology. A number system is simply a way to count. The most commonly used number systems are:

- Decimal number system
- Binary number system
- Octal number system
- Hexadecimal number system



Place value = positional weight

The digit present in greatest positional weight = Most Significant Digit (MSD)

The digit present in lowest positional weight = Least Significant Digit (LSD)

Radix(r) = Different symbols used to represent a number in a number system

(i) Decimal $\Rightarrow$ 10 (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

(ii) Binary $\Rightarrow$ 2 (0, 1)

(iii) Octal $\Rightarrow$ 8 (0, 1, 2, 3, 4, 5, 6, 7, 8, 7)

(iv) Hexadecimal $\Rightarrow$ 16 (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)

Example - 1.1 In a particular number system, 24 + 17 = 40. Find the base of the system.

Solution :

$$[(2\times r)+(4\times 1)]+[(1\times r)+(7\times 1)] = (4\times r)+(0\times 1)$$

$$3r+11 = 4r$$

$$r = 11$$

Example - 1.2 In a particular number system, $\sqrt{41}=5$. Find the base of the system.

Solution :

$$\sqrt{(4\times r)+(1\times 1)} = 5\times 1$$

$$\sqrt{(4r+1)} = 5$$

$$4r+1 = 25$$

$$r = 6$$

Example - 1.3 In a particular number system, roots of $x^2 - 11x + 22 = 0$ are 3, 6. Find the base of the system.

Solution :

For $ax^2 + bx + c = 0$, product of roots $= \dfrac{c}{a}$; Sum of roots $= -\dfrac{b}{a}$

$$(3)_r(6)_r = \frac{(22)_r}{(1)_r}$$

$$(3 \times r^0)(6 \times r^0) = \frac{(2\times r^1) + (2\times r^0)}{(1\times r^0)}$$

$$18 = \frac{2r+2}{1}$$

$$r = 8$$

Example - 1.4 Evaluate $(1.2)_4 + (2.3)_4 = (\_\_\_\_)_4$.

Solution :

$$\begin{array}{r} 1.2 \\ 2.3 \\ {}^{1} \\ \hline 10.1 \end{array} \qquad 4 \,\underline{|\, 5} \;\; 1-1 \Rightarrow (5)_{10} = (11)_4 \qquad 4\,\underline{|\, 4} \;\; 1-0 \Rightarrow (4)_{10} = (10)_4$$

$$(1.2)_4 + (2.3)_4 = (10.1)_4$$

1.1.1 Decimal Number System

- This system has **'base 10'**.
- It has 10 distinct symbols (0, 1, 2, 3, 4, 5, 6, 7, 8 and 9).
- This is a positional value system in which the value of a digit depends on its position.

  $\Rightarrow$ Let we have $(453)_{10}$ is a decimal number

  then,

  4 5 3

  $3 \times 10^0 = 3$

  $5 \times 10^1 = 50$

  $4 \times 10^2 = 400$

  Finally we get, $(453)_{10}$

$\therefore$ We can say "3" is the least significant digit(LSD) and "4" is the most significant digit(MSD).

1.1.2 Binary Number System

- It has base '2' i.e. it has two base numbers 0 and 1 and these base numbers are called "Bits".
- In this number system, group of "Four bits" is known as "Nibble" and group of "Eight bits" is known as "Byte".

  i.e. $\boxed{\text{4 bits = 1 Nibble; 8 bits = 1 Byte}}$

Binary to Decimal Conversion

A binary number is converted to decimal equivalent simply by summing together the weights of various positions in the binary number which contains **'1'**.

Example - 1.5 Find the decimal number representation of $(101101.10101)_2$.

Solution :

$$(101101.10101)_2 = 1\times2^5 + 0\times2^4 + 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} + 0\times2^{-4} + 1\times2^{-5}$$

$$= 32 + 0 + 8 + 4 + 0 + 1 + \frac{1}{2} + 0 + \frac{1}{8} + 0 + \frac{1}{32} = (45.65625)_{10}$$

Decimal to Binary Conversion

The integral decimal number is repeatedly divided by '2' and writing the remainders after each division until a quotient '0' is obtained.

Example - 1.6 Convert $(13)_{10}$ into its equivalent binary number.

Solution :

	Quotient	Remainder	
$13 \div 2$	6	1	LSB
$6 \div 2$	3	0	
$3 \div 2$	1	1	
$1 \div 2$	0	1	MSB

$\therefore \quad (13)_{10} = (1101)_2$

Remember

To convert Fractional decimal into binary, Multiply the number by '2'. After first multiplication integer digit of the product is the first digit after binary point. Later only fraction part of the first product is multiplied by 2. The integer digit of second multiplication is second digit after binary point, and so on. The multiplication by 2 only on the fraction will continue like this based on conversion accuracy or until fractional part becomes zero.

Example - 1.7 Convert $(0.65625)_{10}$ into its equivalent binary number.

Solution :

$$0.65625 \times 2 = 1.31250 \rightarrow 1$$
$$0.31250 \times 2 = 0.62500 \rightarrow 0$$
$$0.62500 \times 2 = 1.25000 \rightarrow 1$$
$$0.25000 \times 2 = 0.50000 \rightarrow 0$$
$$0.50000 \times 2 = 1.00000 \rightarrow 1$$

Thus, $(0.65625)_{10} = (0.10101)_2$

1.1.3 Octal Number System

- It is very important in digital computer because by using the octal number system, the user can simplify the task of entering or reading computer instructions and thus save time.
- It has a base of '8' and it posses 8 distinct symbols (0,1...7).
- It is a method of grouping binary numbers in group of three bits.

Octal to Decimal Conversion

An octal number can be converted to decimal equivalent by multiplying each octal digit by its positional weightage.

Example - 1.8 Convert $(6327.4051)_8$ into its equivalent decimal number.

Solution :

$$(6327.4051)_8 = 6 \times 8^3 + 3 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} + 0 \times 8^{-2} + 5 \times 8^{-3} + 1 \times 8^{-4}$$

$$= 3072 + 192 + 16 + 7 + \frac{4}{8} + 0 + \frac{5}{512} + \frac{1}{4096}$$

$$= (3287.5100098)_{10}$$

Thus, $(6327.4051)_8 = (3287.5100098)_{10}$

Decimal to Octal Conversion

- It is similar to decimal to binary conversion.
- For integral decimal, number is repeatedly divided by '8' and for fraction, number is multiplied by '8'.

Example - 1.9 Convert $(3287.5100098)_{10}$ into its equivalent octal number.

Solution :

For integral part:

	Quotient	Remainder
$3287 \div 8$	410	7
$410 \div 8$	51	2
$51 \div 8$	6	3
$6 \div 8$	0	6

$\therefore$ $(3287)_{10} = (6327)_8$

Now for fractional part:

0.5100098 × 8	0.0800784 × 8	0.6406272 × 8	0.1250176 × 8
4.0800784	0.6406272	5.1250176	1.0001408
4	0	5	1

$\therefore$ $(0.5100098)_{10} = (0.4051)_8$

Finally, $(3287.5100098)_{10} = (6327.4051)_8$

Octal-to-Binary Conversion

This conversion can be done by converting each octal digit into binary individually.

Example - 1.10 Convert $(472)_8$ into its equivalent binary number.

Solution :

$$\begin{array}{cccc} & 4 & 7 & 2 \\ & \Downarrow & \Downarrow & \Downarrow \\ (472)_8 = & (100 & 111 & 010)_2 \end{array}$$

Binary-to-Octal Conversion

In this conversion the binary bit stream are grouped into groups of three bits starting at the LSB and then each group is converted into its octal equivalent. After decimal point grouping start from left.

Example - 1.11 Convert $(1011011110.11001010011)_2$ into its equivalent octal number.

Solution :

For left-side of the radix point, we grouped the bits from LSB: $\underbrace{001}_{1}\ \underbrace{011}_{3}\ \underbrace{011}_{3}\ \underbrace{110}_{6}$

Here two 0's at MSB are added to make a complete group of 3 bits.

For right-side of the radix point, we grouped the bits from MSB: $\underset{\text{radix point}}{\bullet}\ \underbrace{110}_{6}\ \underbrace{010}_{2}\ \underbrace{100}_{4}\ \underbrace{110}_{6}$

Here a '0' at LSB is added to make a complete group of 3 bits.

Finally, $(1011011110.11001010011)_2 = (1336.6246)_8$

1.1.4 Hexadecimal Number System

- The base for this system is "16", which requires 16 distinct symbols to represent the numbers.
- It is a method of grouping 4 bits.
- This number system contains numeric digits (0, 1, 2,....9) and alphabets (*A*, *B*, *C*, *D*, *E* and *F*) both, so this is an "ALPHANUMERIC NUMBER SYSTEM".
- Microprocessor deals with instructions and data that use hexadecimal number system for programming purposes.
- To signify a hexadecimal number, a subscript 16 or letter '*H*' is used i.e. $(A7)_{16}$ or $(A7)_H$.

Hexadecimal	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Hexadecimal-to-Decimal Conversion

Example - 1.12 Convert $(3A.2F)_{16}$ into its equivalent decimal number.

Solution :

$$(3A.2F)_{16} = 3\times 16^1 + 10\times 16^{\circ} + 2\times 16^{-1} + 15\times 16^{-2}$$

$$= 48 + 10 + \frac{2}{16} + \frac{15}{16^2} = (58.1836)_{10}$$

Decimal-to-Hexadecimal Conversion

Example - 1.13 Convert $(675.625)_{10}$ into its equivalent Hexadecimal number.

Solution :

For Integral Part:

	Quotient	Remainder
$675 \div 16$	42	3
$42 \div 16$	2	$10 = A$
$2 \div 16$	0	2

$\therefore \qquad (675)_{10} = (2A3)_{16}$

For Fractional Part:

$$625 \times 16 = 10 = A$$

$\therefore \qquad (0.625)_{10} = (0.A)_{16}$

Finally, $\qquad (675.625)_{10} = (2A3.A)_{16}$

Hexadecimal-to-Binary Conversion

For this conversion replace each hexadecimal digit by its 4 bit binary equivalent.

Example - 1.14 Convert $(2F9A)_{16}$ into its equivalent binary number.

Solution :

$$\begin{array}{cccc} 2 & F & 9 & A \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 0010 & 1111 & 1001 & 1010 \end{array}$$

$\therefore \qquad (2F9A)_{16} = (0010\,1111\,1001\,1010)_2$

Binary-to-Hexadecimal Conversion

For this conversion the binary bit stream is grouped into pairs of four (starting from LSB) and hex number is written for its equivalent binary group.

Example - 1.15 Convert $(10100110101111)_2$ into its equivalent hexadecimal number.

Solution :

$$\underbrace{00\,10}_{\downarrow\,2}\;\;\underbrace{10\,01}_{\downarrow\,9}\;\;\underbrace{10\,10}_{\downarrow\,A}\;\;\underbrace{1111}_{\downarrow\,F}$$

Here two 0's at MSB are added to make a complete group of 4 bits.

$\therefore \qquad (10100110101111)_2 = (29AF)_{16}$

The number systems can also be classified as weighted binary number and unweighted binary number. Where weighted number system is a positional weighted system for example, Binary, Octal, Hexadecimal *BCD*, 2421 etc. The unweighted number systems are non-positional weightage system for example Gray code, Excess-3 code etc.

1.2 Codes

When numbers, letters or words are represented by a special group of symbols, we say that they are being encoded, and the group of symbols is called "CODE".

CODES

Non-Binary

Ex: Morse code

$\Rightarrow$ In Morse code series of dots and dashes represents the letters of the alphabet.

Binary

Ex: Gray code

BCD, Excess-3 Code, 8-4-2-1 code
Error detecting and correcting codes

Alphanumeric

Ex: Boudot's code (5 bit code)

Internal code (6 bit)
ASCII code (7 bit + 1bit parity)
EBCDIC (8 bit)

1.2.1 Binary Coded Decimal Code (BCD)

- In this code, each digit of a decimal number is represented by binary equivalent.
- It is a 4-bit binary code.
- It is also known as "8-4-2-1 code" or simply "BCD Code".
- It is very useful and convenient code for input and output operations in digital circuits.
- Also, it is a "weighted code system".

For example:

$$(943)_{decimal} \longrightarrow (........)_{BCD}$$

$$\Rightarrow \quad \begin{array}{ccc} 9 & 4 & 3 \\ \downarrow & \downarrow & \downarrow \\ 1001 & 0100 & 0011 \end{array}$$

$$\therefore \quad (943)_{10} = (100101000011)_2$$

Advantages of BCD Code

- The main advantage of the BCD code is relative ease of converting to and from decimal.
- Only 4-bit code groups for the decimal digits "0 through 9" need to be remembered.
- This case of conversion is especially important from the hardware standpoint.

  $\Rightarrow$ In 4-bit binary formats, total number of possible representation = $2^4 = 16$

  Then, Valid BCD codes = 10

  Invalid BCD codes = 6

  $\Rightarrow$ In 8-bit binary formats,

  Valid BCD codes = 100

  Invalid BCD codes = 256 – 100 = 156

1.2.2 Excess-3 Code

- It is a 4-bit code.
- It can be derived from BCD code by adding "3" to each coded number.
- It is an "unweighted code".
- It is a "self-complimenting code" i.e. the 1's compliment of an excess-3 number is the excess-3 code for the 9's compliment of corresponding decimal number.
- This code is used in arithmetic circuits because of its property of self complimenting.

Example - 1.16 Convert $(48)_{10}$ into Excess-3 code.

Solution :

$$\begin{array}{cc} 4 & 8 \\ +3 & +3 \\ \hline 7 & 11 \\ \downarrow & \downarrow \\ 0111 & 1011 \end{array}$$

$$\therefore \quad (48)_{10} = \underset{\substack{\Downarrow \\ \text{equivalent} \\ \text{4-bit binary}}}{(01111011)}$$

Example - 1.17 Represent the decimal number 6248 in

(i) BCD code (ii) Excess-3 code (iii) 2421 code

Solution :

(i) BCD code

$$\begin{array}{ccccccc} 6 & & 2 & & 4 & & 8 \\ 0110 & - & 0010 & - & 0100 & - & 1000 \end{array}$$

(ii) Excess-3 = BCD + 3

= 1001 0101 0111 1011

(iii) 2421 code

2	4	2	1	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
1	0	1	1	5
1	1	0	0	6
1	1	0	1	7
1	1	1	0	8
1	1	1	1	9

6248 = 1100 0010 0100 1110

Example - 1.18 The state of a 12-bit register is 100010010111. What is its content if it represents?

(i) Three decimal digits in BCD?

(ii) Three decimal digits in Excess-3 code?

Solution :

(i) In BCD $\Rightarrow$ $\underline{1000}$ $\underline{1001}$ $\underline{0111}$; Decimal digits = 897

(ii) In Excess-3 $\Rightarrow$ $\underset{\substack{\downarrow \\ 8-3=5}}{\underline{1000}}$ $\underset{\substack{\downarrow \\ 9-3=6}}{\underline{1001}}$ $\underset{\substack{\downarrow \\ 7-3=4}}{\underline{0111}}$; Decimal digits = 564

1.2.3 Gray Code

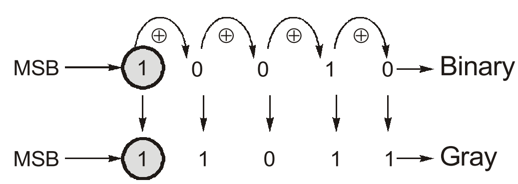
- It is a very useful code also called “minimum change codes” in which only one bit in the code group changes when going from one step to the next.
- It is also known as “Reflected code”.
- It is an unweighted code, meaning that the bit positions in the code groups do not have any specific weight assigned to them.
- This code is not well suited for arithmetic operations but it finds application in input/output devices.
- These are used in instrumentation such as shaft encoders to measures angular displacement or in linear encoders for measurement of linear displacement.

Binary-to-Gray Conversion

- ‘MSB’ in the gray code is same as corresponding digit in binary number.
- Starting from “Left to Right”, add each adjacent pair of binary bits to get next gray code bit. (Discard the carry if generated).

Example - 1.19 Convert $(10010)_2$ to gray code.

Solution :



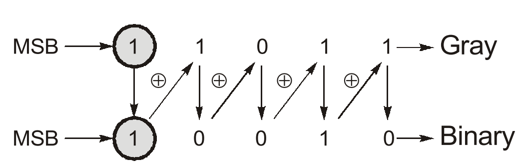
$\therefore$ $(10010)_2 = (11011)_{Gray}$

Gray-to-Binary Conversion

- “MSB” of Binary is same as that of gray code .
- Add each binary bit to the gray code bit of the next adjacent position (discard the carry if generated), to get next bit of the binary number.

Example - 1.20 Convert $(11011)_{Gray}$ to Binary code.

Solution :



$\therefore$ $(11011)_{Gray} = (10010)_2$

Various Binary Codes

Decimal Number	Binary				BCD				Excess-3				Gray			
	B_3	B_2	B_1	B_0	D	C	B	A	E_3	E_2	E_1	E_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
1	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	1
2	0	0	1	0	0	0	1	0	0	1	0	1	0	0	1	1
3	0	0	1	1	0	0	1	1	0	1	1	0	0	0	1	0
4	0	1	0	0	0	1	0	0	0	1	1	1	0	1	1	0
5	0	1	0	1	0	1	0	1	1	0	0	0	0	1	1	1
6	0	1	1	0	0	1	1	0	1	0	0	1	0	1	0	1
7	0	1	1	1	0	1	1	1	1	0	1	0	0	1	0	0
8	1	0	0	0	1	0	0	0	1	0	1	1	1	1	0	0
9	1	0	0	1	1	0	0	1	1	1	0	0	1	1	0	1
10	1	0	1	0									1	1	1	1
11	1	0	1	1									1	1	1	0
12	1	1	0	0									1	0	1	0
13	1	1	0	1									1	0	1	1
14	1	1	1	0									1	0	0	1
15	1	1	1	1									1	0	0	0

1.3 Arithmetic Operations

We are all familiar with the arithmetic operations like addition, subtraction, multiplication and division using decimal numbers. Such operations can also be performed on digital numbers.

1.3.1 Binary Addition

$$0 + 0 = 0; \quad 0 + 1 = 1$$
$$1 + 0 = 1; \quad 1 + 1 = 10$$

For example:

Add the binary numbers 110110 and 101101

Arrow indicates the carry operation

$$\Rightarrow \quad \begin{array}{r} 1\ 1\ 0\ 1\ 1\ 0 \\ +\ 1\ 0\ 1\ 1\ 0\ 1 \\ \hline 1\ 1\ 0\ 0\ 0\ 1\ 1 \end{array}$$

Carry

1.3.2 Binary Subtraction

$$0 - 0 = 0; \quad 10 - 1 = 1 \ \text{(Borrow)}$$
$$1 - 0 = 1; \quad 1 - 1 = 0$$

While subtracting a large number from a smaller number, we can subtract the smaller from the larger and change the sign.

For example:

Subtract two binary numbers 11011 and 10110.

Represents borrow

$$\Rightarrow \quad \begin{array}{r} 1\ 1\ 0\ 1\ 1 \\ -\ 1\ 0\ 1\ 1\ 0 \\ \hline 0\ 0\ 1\ 0\ 1 \end{array}$$

1.3.3 Binary Multiplication

Multiply two binary numbers 1010 and 101

$$\Rightarrow \quad \begin{array}{r} 1010 \times 101 \\ \hline 1010 \\ 000\times \\ 1010\times\times \\ \hline 110010 \end{array}$$

Example - 1.21 If $(10W1Z)_2 \times (15)_{10} = (Y01011001)_2$ then find the value of W, Y, Z.

Solution :

$$\begin{array}{r} 10W1Z \times 1111 \\ \hline 10W1Z \rightarrow 1 \\ 10W1Z \\ 10W1Z \\ 10W1Z \\ \hline 101011001 \end{array}$$

$\because (15)_{10} = (1111)_2$

Also,
$W = 1$

$\therefore W = Y = Z = 1$

Example - 1.22 $(1111)_2 \times (1111)_2 = ?$

Solution :

$$\begin{array}{r} 1111 \times 1111 \\ \hline 1111 \\ 1111 \\ 1111 \\ 1111 \\ \hline 11100001 \end{array}$$

1.3.4 Octal Addition

$0 + 0 = 0$

$0 + 2 = 2$

$1 + 6 = 7$

$1 + 7 = 0$ with carry = 1

Whenever the generated number is greater than 7 then, after decimal addition it should be converted into octal.

For example:

$7 + 2 = 9$

$$\begin{array}{r|l} 8 & 9 \\ \hline & 1 \rightarrow 1 = 11 \end{array}$$

$7 + 17 = 26$

$$\begin{array}{r|l} 8 & 14 \\ \hline & 1 \rightarrow 6 = 16 \end{array}$$

Example - 1.23 Add two octal numbers 567 and 243.

Solution :

$$\Rightarrow \quad \begin{array}{r} {}^{1\,1} \\ 567 \\ +\,243 \\ \hline 1032 \end{array}$$

Here,

$7 + 3 = 10$

$$\begin{array}{r|l} 8 & 10 \\ \hline & 1 \rightarrow 2 \end{array}$$

$10 + 1 = 11$

$$\begin{array}{r|l} 8 & 11 \\ \hline & 1 \rightarrow 3 \end{array}$$

1.3.5 Octal Subtraction

Let, the two octal numbers to be subtracted are 723 and 564.

Then,

$$\Rightarrow \quad \begin{array}{r} 7\;2\;3 \\ -\,5\;6\;4 \\ \hline 1\;3\;7 \end{array}$$

(borrows of 8) ⟶ Represents the borrow

1.3.6 Hexadecimal Addition

$$1 + 1 = 2$$
$$1 + 9 = A$$
$$1 + 15 = 0 \quad \text{with carry '1'}$$
$$A + A = 14 \qquad 16 \,|\, 20,\ 1 \rightarrow 4$$

Example - 1.24 Add two Hexadecimal numbers ADD + DAD = ?

Solution :

$$\Rightarrow \quad \begin{array}{r} {}^{1\,1} \\ A\,D\,D \\ +\,D\,A\,D \\ \hline 1\,8\,8\,A \end{array}$$

$16 \,|\, 26,\ 1 \rightarrow A$

$16 \,|\, 24,\ 1 \rightarrow 8$

$16 \,|\, 24,\ 1 \rightarrow 8$

1.3.7 Hexadecimal Subtraction

Subtract 974B to 587C

$$\Rightarrow \quad \begin{array}{r} 9\;7\;4\;B \\ -\,5\;8\;7\;C \\ \hline 3\;E\;C\;F \end{array}$$

(borrows of 16) ⟶ Represents borrow

1.3.8 BCD Addition

- Addition is the most important operation because subtraction, multiplication, and division can all be done by a series of additions or two's-compliment additions.
- The procedure for BCD addition is as follows:
  (*i*) Add the BCD numbers as regular true binary numbers.
  (*ii*) If the sum is 9(1001) or less, it is a valid BCD answer; leave it as it is.
  (*iii*) If the sum is greater than 9 or if there is a carry-out of the MSB, it is an invalid BCD number.
  (*iv*) If it is invalid, add 6 (0110) to the result to make it valid. Any carry-out of the MSB is added to the next-more-significant BCD number.
  (*v*) Repeat steps 1 to 4 for each group of BCD bits.

Example - 1.25 Convert the decimal number $(76)_{10}$ and $(94)_{10}$ in BCD and add them. Convert the result back to decimal to check the answer.

Solution :

$$\begin{array}{rcll} 76 & = & 0111 & 0110 \\ +94 & = & 1001 & 0100 \\ \hline & & \boxed{10000} & \boxed{1010} \end{array}$$

From the rule we add (0110)

$$\begin{array}{rr} 10000 & 1010 \\ +0110 & 0110 \\ \hline 1\,0111 & 0000 \end{array}$$

In BCD $\Rightarrow (1\ 7\ 0)_{10}$
Also, we have,

$$\begin{array}{r} (76)_{10} \\ +(94)_{10} \\ \hline (170)_{10} \end{array}$$

Example - 1.26 Perform the following addition and subtraction of excess-3 numbers:
(i) 0100 1000 + 0101 1000 (ii) 1100 1011 – 0100 1001
Check the results obtained, by performing the above operations in decimal format.

Solution :
(i) 0100 1000 + 0101 1000 in excess-3 format:

$$\begin{array}{rrl} 0100 & 1000 & \\ 0101 & 1000 & \\ \hline 1001\,1 & 0000 & \text{There is a carry from lower nibble, which is to be propagated} \\ 1 & & \\ \hline 1010 & 0000 & \text{Add "0011" to lower nibble} \\ (-)0011 & (+)0011 & \text{Subtract "0011" from higher nibble} \\ \hline 0111 & 0011 & \leftarrow \text{Final result in excess-3 format} \end{array}$$

Checking the above result in decimal format:

$$0100\ 1000 \xrightarrow{\text{To 8421 BCD}} 00010101 \xrightarrow{\text{To decimal}} (15)_{10}$$

$$01011000 \xrightarrow{\text{To 8421 BCD}} 0010\ 0101 \xrightarrow{\text{To decimal}} (25)_{10}$$

$$(15)_{10} + (25)_{10} = (40)_{10}$$

$$(40)_{10} \xrightarrow{\text{To 8421 BCD}} 0100\ 0000 \xrightarrow{\text{To excess-3}} 01110011$$

(ii) 1100 1011 – 0100 1001 in excess-3 format:

$$\begin{array}{rrl} 1100 & 1011 & \\ (-)\ 0100 & 1001 & \\ \hline 1000 & 0010 & \text{Add "0011" to both the nibbles} \\ (+)\ 0011 & (+)0011 & \\ \hline 1011 & 0101 & \leftarrow \text{Final result in excess-3 format} \end{array}$$

Checking the above result in decimal format:

$$1100\ 1011 \xrightarrow{\text{To 8421 BCD}} 10011000 \xrightarrow{\text{To decimal}} (98)_{10}$$

$$0100\ 1001 \xrightarrow{\text{To 8421 BCD}} 00010110 \xrightarrow{\text{To decimal}} (16)_{10}$$

$$(98)_{10} - (16)_{10} = (82)_{10}$$

$$(82)_{10} \xrightarrow{\text{To 8421 BCD}} 1000\ 0010 \xrightarrow{\text{To excess-3}} 10110101$$

Student's Assignments 1

Q.1 List out the rules for the BCD (Binary Coded Decimal) addition with corresponding examples?

Q.2 For two binary numbers X = 1010100 and Y = 1000011. Perform the subtraction.

(i) $X - Y$

(ii) $Y - X$ by using 2's compliment method.

Answer: (Conventional)

2. (i) (0010001) (ii) –(0010001)

Student's Assignments 2

Q.1 If we convert a binary sequence, $(1100101 \cdot 1011)_2$ into its octal equivalent as (X)s, the value of 'X' will be

(a) (145.13) (b) (145.54)
(c) (624.54) (d) (624.13)

Q.2 A binary $(11011)_2$ may be represented by following ways:

1. $(33)_8$
2. $(27)_{10}$
3. $(10110)_{GRAY}$
4. $(1\,B)_H$

Which of these above is/are correct representation?

(a) 1, 2 and 3 (b) 2 and 4
(c) 1, 2, 3 and 4 (d) only 2

Q.3 Consider $X = (54)_b$ where 'b' is the base of the number system. If $\sqrt{X} = 7$ then base 'b' will be

(a) 7 (b) 8
(c) 9 (d) 10

Q.4 Regarding ASCII codes, which one of the following characteristics is NOT correct?

(a) It is an Alphanumeric code.
(b) It is an 8-bit code.
(c) It has 128 characters including control characters.
(d) The minimum distance of ASCII code is '1'.

Q.5 Addition of all gray code to convert decimal (0 – 9) into gray code is

(a) 129 (b) 108
(c) 69 (d) 53

Q.6 The decimal equivalent of hexadecimal number of '2A0F' is

(a) 17670 (b) 17607
(c) 17067 (d) 10767

Q.7 A new Binary Coded Pentary (BCP) number system is proposed in which every digit of a base-5 number is represented by its corresponding 3-bit binary code. For example, the base-5 number 24 will be represented by its BCP code 010100. In this numbering system, the BCP code 100010011001 corresponds to the following number in base-5 system

(a) 423 (b) 1324
(c) 2201 (d) 4231

Answer Key :

1. (b) **2.** (c) **3.** (c) **4.** (b) **5.** (d)
6. (d) **7.** (d)

■■■■