B.Sc. T.Y

SEM - VI

PAPER - XV-A

DIGITAL AND COMMUNICATION ELECTRONICS

UNIT-I

NUMBER SYSTEMS

Page No:
Date:

A continuously varying signal (voltage or current) is called an analog signal. For example, a sinusoidal voltage is an analog signal. In an analog electronic circuit, the output voltage changes continuously according to the input voltage variations. In other words, the output voltage can have an infinite number of values. A signal (voltage or current) which can have only two discrete value is called a digital signal. For example, a square wave is a digital signal. The semiconductor devices (e.g. diodes, transistors) can be designed for two-state operation i.e saturation and cut off. In that case, the output voltage can have only two states (i.e values), either low or high. An electronic circuit that is designed for two-state operation is called a digital circuit.

The branch of electronics which deals with digital circuits is called digital electronics.

Analog Signal

A continuously varying signal (voltage or current is called an analog signal. Fig. 1 shows the analog signal. If such an analog signal is applied to the input of a transistor amplifier, the output voltage will also vary sinusoidally.

This is the analog operation i.e the output voltage can have an infinite number of values. Due to many-valued output, the analog operation is less reliable.

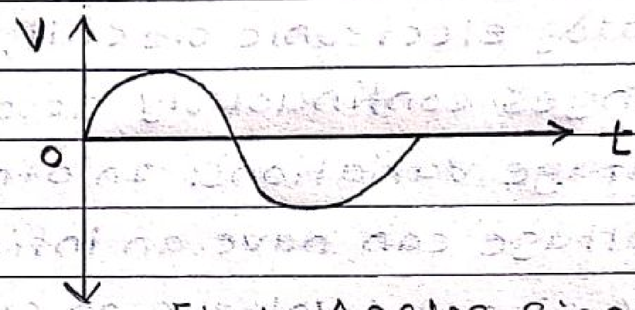

Fig. 1 Analog Signal

Digital Signal

A signal (voltage or current) that can have only two discrete value is called a digital signal. For example, a square wave is digital signal. shown in Fig. 2

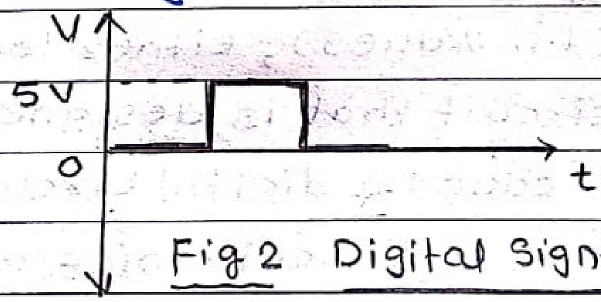

Fig 2 Digital Signal

It is because this signal has only two values ie +5 V and 0 V and no other value. These values are called High and Low. The High voltage is +5 V and the low voltage is 0 V. If proper signal is applied to the input of a transistor, the transistor can be driven between cut off and saturation. In other words, the transistor will have two-state operations i.e output is either low or high. Since digital operation has only two states (i.e ON and OFF), it is far more reliable than many-valued analog operation.

Digital Circuit

An electronic circuit that handles only a digital signal is called a digital circuit.

The output voltage of a digital circuit is either low or high and no other value. In other words digital operation is a two-state operation. These states are expressed as (High or Low) or (ON or OFF) or (1 or 0). Therefore, a digital circuit is one that expresses the values in digits 1's or 0's. Hence the name digital. The numbering concept that uses only the two digits 1 and 0 is the binary numbering system.

* Radix (Base)

The radix or base of a number system is defined as the number of different symbols which can occur in each position in the number system. The decimal number system has radix or base of 10. Thus the system has 10 different digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) any one of which

Page No.:
Date:

may be used in each position in a number

* Decimal Number System

The number system which has base or radix 10 and uses 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 basic digits is called as decimal number system. It is weighted number system, because each digit in the decimal number has distinct positional weight.

Consider the decimal number 7777. It can be written as

$$7777 = 7000 + 700 + 70 + 7$$
$$= (7\times1000) + (7\times100) + (7\times10) + (7\times1)$$
$$= 7\times10^{3} + 7\times10^{2} + 7\times10^{1} + 7\times10^{0}$$

In the above decimal number, all digits are 7 but the positional weight of each digit is different i.e positional weight of first digit is thousand, that of second digit is hundred, third digit is ten and fourth digit is one.

Consider the fractional decimal number 0.315, it can be written as

$$0.315 = 0.3 + 0.01 + 0.005$$
$$= 3\times10^{-1} + 1\times10^{-2} + 5\times10^{-3}$$

Thus, weights of different position in a decimal number system is given by

$$\leftarrow \; 10^{3} \quad 10^{2} \quad 10^{1} \quad 10^{0} \, . \, 10^{-1} \quad 10^{-2} \quad 10^{-3} \; \rightarrow$$

$\uparrow$ Decimal point

* Binary Number System

The number system which has base or radix 2 and uses 0 and 1 basic digits is called as binary number system. It is used in computer and many digital circuits. Like decimal number system, it is also weighted number system.

The weights of different positions in binary number system is given by

$$\leftarrow \; 2^3 \;\; 2^2 \;\; 2^1 \;\; 2^0 \; . \; 2^{-1} \;\; 2^{-2} \;\; 2^{-3} \; \rightarrow$$

↑
Binary point

The below table shows some decimal numbers and their binary equivalent numbers.

Decimal No.	Binary Equivalent No.	Decimal No.	Binary No.
0	0	9	1001
1	1	10	1010
2	10	11	1011
3	11	12	1100
4	100	13	1101
5	101	14	1110
6	110	15	1111
7	111		
8	1000		

Notes:-

(i) Each binary digit (0 or 1) is referred to as a bit. A string of four bits is called as a nibble and eight bits make a byte. Thus, 1001 is a nibble and 10010110 is a binary byte.

(ii) The binary number system is the most useful in digital circuits because there are only two digits (0 and 1).

* Octal number system

The number system which has base or radix 8 and uses 0, 1, 2, 3, 4, 5, 6 and 7 basic digits is called as octal number system. It is useful in computer industry and it is also weighted number system. The weights of different positions in octal number system is given by

$$\leftarrow 8^3 \; 8^2 \; 8^1 \; 8^0 \; . \; 8^{-1} \; 8^{-2} \; 8^{-3} \rightarrow$$

octual point

The below table shows some decimal numbers with their octal equivalent numbers.

Decimal No.	Octal No.	Decimal No.	Octal No.
0	0	7	7
1	1	8	10
2	2	9	11
3	3	10	12
4	4	11	13
5	5	12	14
6	6	13	15

* Hexadecimal Number System

The number system which has base or radix 16 and used 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F basic symbols is called as hexadecimal number system. It is a weighted number system. It is useful in computer industry. The weights of different positions in hexadecimal number system is given by,

$$\leftarrow 16^3 \quad 16^2 \quad 16^1 \quad 16^0 \; . \; 16^{-1} \quad 16^{-2} \quad 16^{-3} \rightarrow$$

↑ Hexadecimal point

The below table shows some decimal numbers with their hexadecimal equivalent numbers

Decimal No.	Hexadeci. No.	Decimal No.	Hexadeci. No.
0	0	10	A
1	1	11	B
2	2	12	C
3	3	13	D
4	4	14	E
5	5	15	F
6	6	16	10
7	7	17	11
8	8	18	12
9	9	19	13

* Decimal to Binary Conversion

An integer decimal number can be converted into binary using double-dabble method. It is also known as divide-by two method. In this method, progressively divide the given integer decimal no. by 2 and write down the remainder after each division and taking the remainders from bottom to top, we get the equivalent binary number.

* Convert $(21)_{10}$ to binary

			Remainders	
2	21			
2	10	→	1	↑ (LSB)
2	5	→	0	
2	2	→	1	
2	1	→	0	
	0	→	1	(MSB)

$\therefore (21)_{10} = (10101)_2$

* convert $(37)_{10}$ to binary

			Remainders	
2	37			
2	18	→	1	↑ (LSB)
2	9	→	0	
2	4	→	1	
2	2	→	0	
2	1	→	0	
	0	→	1	(MSB)

$\therefore (37)_{10} = (100101)_2$

Page No:

Date:

The fractional decimal no. can be converted into Binary using multiply by two rule ie multiply the fraction decimal no. by 2 and record the carry in the integer position. Taking the carries from top to bottom, we get equivalent binary number.

* convert $(0.3125)_{10}$ to binary

$0.3125 \times 2 = 0.625$ with a carry of 0

$0.625 \times 2 = 0.25$ with a carry of 1

$0.25 \times 2 = 0.50$ with a carry of 0

$0.50 \times 2 = 0.00$ with a carry of 1

$\therefore (0.3125)_{10} = (0.0101)_2$

* convert $(12.15)_{10}$ to binary

(a) conversion of integer part (12)

			Reminders
2	12		
2	6	→	0
2	3	→	0
2	1	→	1
	0	→	1

$\therefore (12)_{10} = (1100)_2$

(b) conversion of fractional part (0.15)

$0.15 \times 2 = 0.30$ with a carry of 0

$0.30 \times 2 = 0.60$ with a carry of 0

$0.60 \times 2 = 0.20$ with a carry of 1

$0.\cancel{2}0 \times 2 = 0.40$ with a carry of 0

$\therefore (0.15)_{10} = (0.0010)_2$

$\therefore (12.15)_{10} = (1100.0010)_2$

Page No.:
Date :

* conversion of $(57.35)_{10}$ to binary

(a) Integer part (57)

			Remainders
2	57		
2	28	$\rightarrow$	1
2	14	$\rightarrow$	0
2	7	$\rightarrow$	0
2	3	$\rightarrow$	1
2	1	$\rightarrow$	1
	0	$\rightarrow$	1

$\therefore (57)_{10} = (111001)_2$

(b) Fractional part (0.35)

$0.35 \times 2 = 0.70$ with a carry of 0

$0.70 \times 2 = 0.40$ with a carry of 1

$0.40 \times 2 = 0.80$ with a carry of 0

$0.80 \times 2 = 0.60$ with a carry of 1

$\therefore (0.35)_{10} = (0101)_2$

$\therefore (57.35)_{10} = (111001.0101)_2$

Scanned with CamSc

* Binary to Decimal Conversion

Binary numbers can be converted to equivalent decimal numbers quite easily. Let us convert the binary number 110011 to decimal. Its conversion to equivalent decimal number involves the following two steps:

(i) Place the decimal value of each position of the binary number.

(ii) Add all the decimal values to get the decimal number.

$$\begin{array}{cccccc} 1 & 1 & 0 & 0 & 1 & 1 \\ 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array}$$

Thus $(110011)_2 = 1\times 2^5 + 1\times 2^4 + 0\times 2^3 + 0\times 2^2 +$
$1\times 2^1 + 1\times 2^0$

$= 1\times 32 + 1\times 16 + 0\times 8 + 0\times 4 +$
$1\times 2 + 1\times 1$

$= 32 + 16 + 0 + 0 + 2 + 1$

$= 51$

$\therefore (110011)_2 = (51)_{10}$

* Convert $(1011011\cdot101)_2$ to decimal number

$$(1011011.101)_2 = \begin{array}{cccccccccc} 1 & 0 & 1 & 1 & 0 & 1 & 1 & \cdot & 1 & 0 & 1 \\ 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & \cdot & 2^{-1} & 2^{-2} & 2^{-3} \end{array}$$

$= 2^6 + 2^4 + 2^3 + 2^1 + 2^0 + 2^{-1} + 2^{-3}$

$= 64 + 16 + 8 + 2 + 1 + 0.5 + 0.125$

$= 91.625$

$\therefore (1011011\cdot101)_2 = (91.625)_2$

Page No.:
Date:

* Decimal to octal conversion

An integer decimal no. can be converted into octal using octal dabble method which is same as double dabble method. Instead of dividing by 2 (base of binary no. system) we divide by 8 (base of octal no. system) and writing down the remainders after each division. Taking the remainders from bottom to top we get equivalent octal number.

* convert $(359)_{10}$ to octal number.

		Remainders
8	359	
8	44	→ 7 ↑
8	5	→ 4
	0	→ 5

$\therefore (359)_{10} = (547)_8$

* convert $(163)_{10}$ to octal number

		Remainders
8	163	
8	20	3 ↑
8	2	4
	0	2

$(163)_{10} = (243)_8$

(b) conversion of fractional decimal no. into octal

With decimal fractions, multiply instead of divide, writing the carry into integer ~~part~~ position. Take the carries from top to bottom. We get equivalent octal fraction

* convert $(0.23)_{10}$ to octal

$0.23 \times 8 = 0.84$ with a carry of 1

$0.84 \times 8 = 0.72$ with a carry of 6

$0.72 \times 8 = 0.76$ with a carry of 5

$\therefore (0.23)_{10} = (0.165)_8$

* convert $(98.20)_{10}$ to octal number

(a) Integer part (98)

		Remainders
8	98	
8	12	$\rightarrow$ 2
8	1	$\rightarrow$ 4
	0	$\rightarrow$ 1

$\therefore (98)_{10} = (142)_8$

(b) Fractional part (0.20).

$0.20 \times 8 = 0.60$ with a carry of 1

$0.60 \times 8 = 0.80$ with a carry of 4

$0.80 \times 8 = 0.40$ with a carry of 6

$0.40 \times 8 = 0.20$ with a carry of 3

$\therefore (0.20)_{10} = (0.1463)_8$

$\therefore (98.20)_{10} = (142.1463)_8$

* octal to decimal conversion

To convert the given octal no. to decimal, multiply each octal digit by its position weight and add the resulting products.

* convert $(2374)_8$ to decimal number

$(2374)_8 = (2\times8^3)+(3\times8^2)+(7\times8^1)+(4\times8^0)$

$= (2\times512)+(3\times64)+(7\times8)+(4\times1)$

$= 1024 + 192 + 56 + 4$

$= 1276$

$\therefore (2374)_8 = (1276)_{10}$

* convert $(46.5)_8$ to decimal number.

$\therefore (46.5)_8 = (4\times8^1)+(6\times8^0)+(5\times8^{-1})$

$= 32 + 6 + 0.625$

$= (38.625)_{10}$

$\therefore (46.5)_8 = (38.625)_{10}$

* Octal to Binary conversion

The base of octal numbers is 8, and 8 is the third power of 2 (base of binary numbers) i.e $8 = 2^3$ so to convert octal number to binary, change each octal digit to its 3 bit binary equivalent.

* (a) convert $(432)_8$ into binary

$$\begin{array}{ccc} 4 & 3 & 2 \\ \downarrow & \downarrow & \downarrow \\ 100 & 011 & 010 \end{array}$$

$\therefore (432)_8 = (100011010)_2$

* (b) convert $(576.21)_8$ to binary.

$$\begin{array}{cccccc} 5 & 7 & 6 & . & 2 & 1 \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ 101 & 111 & 110 & & 010 & 001 \end{array}$$

$\therefore (576.21)_8 = (101111110.010001)_2$

* convert octal number $(5431)_8$ to binary.

$$\begin{array}{cccc} 5 & 4 & 3 & 1 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 101 & 100 & 011 & 001 \end{array}$$

Therefore, octal 5431 is equivalent to binary 101100001 i.e

$(5431)_8 = (101100011001)_2$

Page No.:
Date:

The advantage of octal number system is the ease with which an octal number can be converted to a binary number and vice-versa. It is because eight is third power of two, providing a direct correlation between three-bit groups in a binary number and the octal digits i.e each three-bit group of binary bits can be represented by one octal digit. Therefore, conversion from octal to binary is performed by converting each octal digit to its 3-bit binary equivalent.

Octal and Binary Equivt	
Octal Digit	Binary Bits
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

* Binary to octal Conversion

To convert the binary number to octal, make the group of three bits, starting from the binary point (0s are added to each end, if necessary) then convert each group to its octal equivalent.

* convert binary number $(100111010)_2$ to octal no.

$$(100111010)_2 = \underset{4}{\underline{100}}\ \underset{7}{\underline{111}}\ \underset{2}{\underline{010}}$$

$\therefore (100111010)_2 = (472)_8$

* convert $(11010110)_2$ to octal no.

$$(11010110)_2 = \underset{3}{\underline{011}}\ \underset{2}{\underline{010}}\ \underset{6}{\underline{110}}$$

$(11010110)_2 = (326)_8$

* convert $(10101.01)_2$ to octal no.

$$(10101.01)_2 = \underline{10}\ \underline{101}\ .\ \underline{01}$$

$$= \underset{2}{\underline{010}}\ \underset{5}{\underline{101}}\ .\ \underset{2}{\underline{010}}$$

$\therefore (10101.01)_2 = (25.2)_8$

* Decimal to hexadecimal conversion

To convert the given decimal number to hexadecimal, hex dabble method is used. In this method, divide sussessively the given decimal no. by 16, writing down the remainders and take the remainders from bottom to top to get equivalent hexadecimal number.

* convert $(650)_{10}$ to hexadecimal number.

		Remainders
16	650	
16	40	→ 10 → A
16	2	→ 8
	0	→ 2

$\therefore (650)_{10} = (28A)_{16}$

* convert $(573)_{10}$ to hexadecimal number.

		Remainders
16	573	
16	35	→ 13 → D
16	2	→ 3
	0	→ 2

$\therefore (573)_{10} = (23D)_{16}$

* Hexadecimal to Decimal Conversion

To convert the given hexadecimal number to a decimal, multiply each hexadecimal digit by its position weight and add the resulting products.

* convert $(A85)_{16}$ to decimal.

$$(A85)_{16} = (A \times 16^2) + (8 \times 16^1) + (5 \times 16^0)$$
$$= (10 \times 256) + (8 \times 16) + (5 \times 1)$$
$$= 2560 + 128 + 5$$
$$= (2693)_{10}$$

$\therefore (A85)_{16} = (2693)_{10}$

* convert $(B2F8.84)_{16}$ to decimal number.

$$(B2F8.84)_{16} = (B \times 16^3) + (2 \times 16^2) + (F \times 16^1) + (8 \times 16^0) + (8 \times 16^{-1}) + (4 \times 16^{-2})$$
$$= (11 \times 4096) + (2 \times 256) + (15 \times 16) + (8 \times 1) + \left(\frac{8}{16}\right) + \left(\frac{4}{256}\right)$$
$$= 45056 + 512 + 240 + 8 + 0.5 + 0.140625$$
$$= (45816.640625)_{10}$$

$\therefore (B2F8.84)_{16} = (45816.640625)_{10}$

* Hexadecimal to Binary Conversion

To convert a hexadecimal number to binary number, convert each hexadecimal digit to its 4-bits binary equivalent ($\because 16 = 2^4$)

* convert hex number $(9F2)_{16}$ to binary number

$$\begin{array}{ccc} 9 & F & 2 \\ \downarrow & \downarrow & \downarrow \\ 1001 & 1111 & 0010 \end{array}$$

$\therefore (9F2)_{16} = (100111110010)_2$

* convert $(AB2D.7F)_{16}$ to binary number.

$$\begin{array}{cccccc} A & B & 2 & D . & 7 & F \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 1010 & 1011 & 0010 & 1101 . & 0111 & 1111 \end{array}$$

$\therefore (AB2D.7F)_{16} = (1010101100101101.01111111)_2$

* convert $(1234.C)_{16}$ to binary number

$$\begin{array}{ccccc} 1 & 2 & 3 & 4 . & C \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 0001 & 0010 & 0011 & 0100 . & 1100 \end{array}$$

$\therefore (1234.C)_{16} = (0001001000110100.1100)_2$

Each hexadecimal digit represents a group of four binary digits.

Decimal	Hexadecimal	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
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

* Binary to hexadecimal conversion

To convert the given binary number to a hexadecimal, make the group of 4 bit, starting from the binary points (0s are added to each end, if necessary), then convert each group to its hexadecimal equivalent.

* convert binary number $(1110100110)_2$ to its equivalent hex number.

The given number is

1110100110

$\therefore$ 0011 1010 0110

$\downarrow$ $\downarrow$ $\downarrow$

3 A 6

$\therefore (1110100110)_2 = (3A6)_{16}$

* convert $(10010110.1011)_2$ to hex number.

$(10010110.1011)_2 =$ 1001 0110 . 1011

$\downarrow$ $\downarrow$ $\downarrow$

9 6 . 13 B

$\therefore (10010110.1011)_2 = (96.B)_{16}$

* convert $(10110101111.001)_2$ to hex number

$(10110101111.001)_2 =$ 101 1010 1111 . 001

$\downarrow$ $\downarrow$ $\downarrow$ $\downarrow$

$=$ 0101 1010 1111 . 0010

$\downarrow$ $\downarrow$ $\downarrow$ $\downarrow$

5 A F . 2

$\therefore (10110101111.001)_2 = (5AF.2)_{16}$

Page No.:

* Binary Coded Decimal Code (BCD code)

The code in which basic decimal digits 0 through 9 are represented by their binary equivalents using four bits is called as BCD code. It is also called as 8421 code. It is weighted code

BCD code is very convenient and useful code for input and output operations in digital circuits. It is used to represent decimal digits in systems like digital calculators, voltmeters etc.

The below table shows decimal digits and corresponding 8421 BCD code.

Decimal Digit	Binary Equivalent	8421 BCD code
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	0001 0000
11	1011	0001 0001
12	1100	0001 0010

Page No.:
Date:

* Convert $(3469)_{10}$ to BCD code

3 4 6 9

0011 0100 0110 1001

$\therefore (3469)_{10} = (0011\,0100\,0110\,1001)_{BCD}$

* What decimal number is represented by the BCD string given below?

0100 0000 0010

→ Divide the BCD ~~st~~ number into 4-bit groups and convert each to decimal.

0100	0000	0010
↓	↓	↓
4	0	2

So, the equivalent decimal number is $(402)_{10}$

Note:- To avoid confusion between BCD and true binary, a BCD string is often seperated into groups of 4 binary bits or a subscript BCD is sometimes attached to the string as below:

0100 0000 0010 or $(010000000010)_{BCD}$

Page No.:
Date:

* Gray Code

The gray code is non-weighted code. The below table shows the Gray code along with the binary numbers. Each gray code number differs from the preceding number by a single bit. For instance goining from decimal 7 to 8, the Gray number changes from 0100 to 1100, these numbers differs only in the most significant bit.

Decimal Digit	Binary code	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111

* Binary to Gray Code Conversion

To convert the given binary no. to Gray code, following procedure is adopted.

1) Write the binary number.
2) MSB of Gray code is same as in binary number.
3) Goining from left to right, add each adjescent pair of binary digits to get the next Gray code digit, discard the carries.

* convert $(101101)_2$ to Gray code

Binary	1	0	1	1	0	1	
	↓	+1	+0	+1	+1	+0	
	1	1	1	0	1	1	← Gray code

* convert $(110010)_2$ to Gray code

Binary	1	1	0	0	1	0	
	↓	+1	+1	+0	+0	+ϕ	
	1	0	1	0	1	ϕ	— Gray code

(1 0 1 0 1 1) Gray code

* Gray Code to Binary Conversion

To convert Gray code to binary, following procedure is adopted

1> Write the Gray code number.

2> The MSB of binary number is same as in Gray code number

3> Add the MSB of binary number (diagonally) into the 2nd MSB of Gray code number to get 2nd MSB of binary number. Then add this 2nd MSB of binary into the 3rd MSB of Gray code no. to get 3rd MSB of binary no. In this way this procedure is continued till we get LSB of binary no. Discard the carries.

* convert the Gray code no. (101011) to binary

Gray code no. 1 0 1 0 1 1

Binary No. → 1 1 0 0 1 0

* convert the Gray code no. (111011) to binary

Gray code no. 1 1 1 0 1 1

1 0 1 1 0 1 ← Binary number

Page No.:
Date:

* Excess-3 code

To convert any decimal number into its Excess-3 form, add 3 to each decimal digit, then convert the sum to a binary number.

* convert $(45)_{10}$ to Excess-3 code

4	5
+3	+3
7	8
↓	↓
0111	1000

$\therefore (45)_{10} = (01111000)$ Excess-3

* convert $(38)_{10}$ to Excess-3 code

3	8
+3	+3
6	11
↓	↓
0110	1011

$\therefore (38)_{10} = (01101011)$ Excess-3

After adding 3 into 8, do not carry the to the next column, convert the 11 to bina

Page No.:
Date:

* Binary Arithmetic

The binary arithmetic is much simpler to learn because binary system deals with only two digits 0 and 1.

* Binary Addition

Binary addition is performed in the same manner as decimal addition. However, since binary system has only two digits, the addition table for binary arithmetic is very simple consisting of only four entries. The complete table for binary addition is as follows:

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 0 plus a carry of 1 to next higher column. carry-overs are performed in the same manner as in decimal arithematic. since 1 is the largest digit in the binary system, any sum greater than 1 requires that a digit be carried over.

* Add the binary numbers 1010 and 0101 in both decimal and binary form

Solution

Binary	Decimal
1010	10
+ 0101	+ 5
1111	15

* Add the binary numbers 10011 and 1001 in both decimal and binary form.

Sol^n →

Binary	Decimal
11	1
10011	19
+ 1001	+ 9
11100	28

* Binary Subtraction

The principles of binary subtraction can as well as applied to subtraction of numbers in other bases. The complete table for binary subtraction is as follows :-

0 - 0 = 0

1 - 0 = 1

1 - 1 = 0

0 - 1 = 1 with a borrow from the next column.

⊛ subtract $(0101)_2$ from $(1110)_2$.

Sol^n →

Borrow

```
  01
 1110
-0101
 0001
```

* Binary Multiplication

There are four basic rules for multiplying binary numbers. Multiplication involves forming the partial products, shifting each partial product left one place and then adding all partial products.

$0 \times 0 = 0$

$0 \times 1 = 0$

$1 \times 0 = 0$

$1 \times 1 = 1$

* carry out the following multiplications. Also show the equivalent decimal multiplication

(a) $10101_2 \times 101_2$ (b) $1011_2 \times 1001_2$

→ Solution

(a)

```
      1 0 1 0 1          21₁₀
    ×     1 0 1        ×  5₁₀
    ---------          ------
      1 0 1 0 1         105₁₀
    + 0 0 0 0 0
  + 1 0 1 0 1
  -------------
  1 1 0 1 0 0 1₂
```

(b)

```
        1 0 1 1          11₁₀
      × 1 0 0 1        ×  9₁₀
      ---------        ------
        1 0 1 1          99₁₀
      0 0 0 0
    0 0 0 0
  1 0 1 1
```

Page No:
Date:

* Complement of a Number

In digital work, two types of complements of a binary number are used for complement subtraction.

(a) 1's Complement

The 1's complement of a binary number is obtained by changing its each 0 into 1 and 1 into 0. It is also called radix-minus-one complement. eg 1's complement of 100_2 is 011_2 and that of 1110_2 is 0001_2

(b) 2's Complement

The 2's complement of a binary number is obtained by adding 1 to its 1's complement.

$\therefore$ 2's complement = 1's complement + 1

It is known as true complement. suppose we have to find 2's complement of 1011_2. It's 1's complement is 0100_2. Next adding 1 to it 0101_2. Hence 2's complement of 1011_2 is 0101_2.

The complement method of subtraction reduces subtraction to an addition process. This method is popular in digital computers