ELECTRICAL ENGINEERING

Digital Electronics

Comprehensive Theory
with Solved Examples *and* Practice Questions

MADE EASY PUBLICATIONS

www.madeeasypublications.org

MADE EASY Publications Pvt. Ltd.
Corporate Office: 44-A/4, Kalu Sarai (Near Hauz Khas Metro Station), New Delhi-110016 | **Ph. :** 9021300500
Email : infomep@madeeasy.in | **Web :** www.madeeasypublications.org

Digital Electronics

MADE EASY Publications Pvt. Ltd. has taken due care in collecting the data and providing the solutions, before publishing this book. Inspite of this, if any inaccuracy or printing error occurs then **MADE EASY Publications Pvt. Ltd.** owes no responsibility. We will be grateful if you could point out any such error. Your suggestions will be appreciated.

EDITIONS

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
Tenth Edition : 2024
Eleventh Edition : 2025
Twelfth Edition : 2026

CONTENTS

Digital Electronics

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 SYSTEM

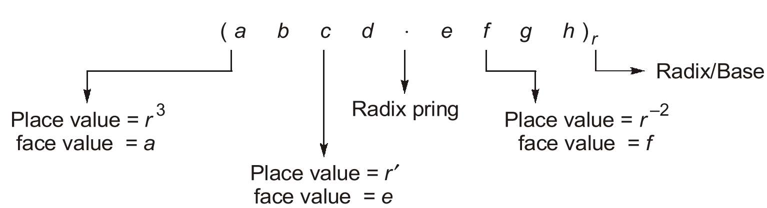
A number system is simply a way to count. The number of systems are called position weighted systems, since the weight of each digit depends on its relative position within the number. Many number systems are used in digital technology.

Base (or) Radix of System

It is defined as the number of different symbols (digits (or) character) used in that number system.

- If the base of the number system is 'r', the number of different symbols used in the system is 'r' i.e. the different symbols are '0 to $(r-1)$'.
- The largest value of digits in base 'r' system is '$(r-1)$'.

	General	Binary	Octal	Decimal	Hexadecimal
Base	r	2	8	10	16
Symbols	0, 1, 2, ... $(r-1)$,	0, 1	0, 1, 2, 3, 4, 5, 6, 7	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	0, 1, 2, 3, 4, 5, 6, 7 8, 9, *A*, *B*, *C*, *D*, *E*, *F*
Maximum digit	$(r-1)$	1	7	9	*F*



Place value = Positional weight

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

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

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 :

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

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

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,

$$\begin{array}{ccc l} 4 & 5 & 3 & \\ & & \to & 3 \times 10^0 = 3 \\ & \to & & 5 \times 10^1 = 50 \\ \to & & & 4 \times 10^2 = 400 \\ \hline \end{array}$$

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 :

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

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

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

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

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 :

$$\begin{array}{r} 0.65625 \\ \times 2 \\ \hline 1.31250 \\ \downarrow \\ 1 \end{array} \quad \begin{array}{r} 0.31250 \\ \times 2 \\ \hline 0.62500 \\ \downarrow \\ 0 \end{array} \quad \begin{array}{r} 0.62500 \\ \times 2 \\ \hline 1.25000 \\ \downarrow \\ 1 \end{array} \quad \begin{array}{r} 0.25000 \\ \times 2 \\ \hline 0.50000 \\ \downarrow \\ 0 \end{array} \quad \begin{array}{r} 0.50000 \\ \times 2 \\ \hline 1.00000 \\ \downarrow \\ 1 \end{array}$$

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:

$$0.5100098 \times 8 = 4.0800784 \rightarrow 4$$

$$0.0800784 \times 8 = 0.6406272 \rightarrow 0$$

$$0.6406272 \times 8 = 5.1250176 \rightarrow 5$$

$$0.1250176 \times 8 = 1.0001408 \rightarrow 1$$

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

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 :

$$\therefore \quad (472)_8 = (\underset{\Downarrow}{\overset{4}{100}}\ \ \underset{\Downarrow}{\overset{7}{111}}\ \ \underset{\Downarrow}{\overset{2}{010}})_2$$

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: $\underbrace{001}_{\downarrow\,1}\ \underbrace{011}_{\downarrow\,3}\ \underbrace{011}_{\downarrow\,3}\ \underbrace{110}_{\downarrow\,6}$

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: $\underset{\substack{\uparrow \\ \text{radix point}}}{\bullet}\ \underbrace{110}_{\downarrow\,6}\ \underbrace{010}_{\downarrow\,2}\ \underbrace{100}_{\downarrow\,4}\ \underbrace{110}_{\downarrow\,6}$

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 :

$$(3A.2F)_{16} = 3\times16^1 + 10\times16^{\circ} + 2\times16^{-1} + 15\times16^{-2}$$

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

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

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

For Fractional Part:

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

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

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

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 :

2	F	9	A
↓	↓	↓	↓
0010	1111	1001	1010

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

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

⇒ 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".
- The designation 8421 indicates the binary weights of the four bits (2^3, 2^2, 2^1, 2^0).
- It is very useful and convenient code for input and output operations in digital circuits.
- Also, it is a "weighted code system".

Invalid Codes

With 4-bits, 16-numbers (0000 through 1111) can be represented but, in 8421 code only ten of these are used. The six code combinations that are not used 1010, 1011, 1100, 1101, 1110 and 1111 are invalid in the 8421 BCD code.

Decimal Number	8421 *BCD* Code	2421 *BCD* Code	5421 *BCD* Code
0	0000	0000	0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	1011	1000
6	0110	1100	1001
7	0111	1101	1010
8	1000	1110	1011
9	1001	1111	1100

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

  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

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

(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 $\Rightarrow$ $\underbrace{1000}$ $\underbrace{1001}$ $\underbrace{0111}$; Decimal digits = 897

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

1.2.3 Gray Code

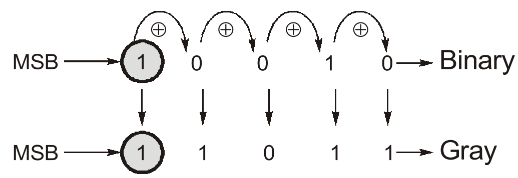
- It is a very useful code also called "minimum change codes" in which only one bit in the code group changes when going from one step to the next.
- It is also known as "Reflected code".
- It is an unweighted code, meaning that the bit positions in the code groups do not have any specific weight assigned 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 :



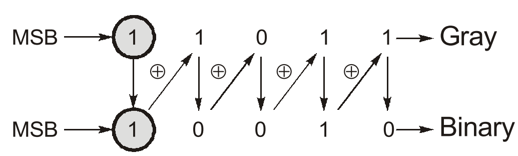
$\therefore \qquad (10010)_2 = (11011)_{Gray}$

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 :



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

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

1.2.4 Error Detection Codes

- A parity bit is an extra bit added to a string of data bits for bit error detection.
- A parity bit is attached to a group of bits to make total number of 1's in a group always even for even parity and always odd for odd parity.
- A parity bit provides for the detection of single bit error (or any odd number of errors which is unlikely) but cannot check for two errors one group.

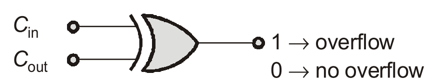
1.5 OVER FLOW CONCEPT

- If we add two same sign numbers and in the result if sign is opposite then it indicates "OVERLFLOW".
- When "OVERFLOW" occurs, then number of bits being increased by "1-bit in MSB".

1.5.1 Overflow Condition

- If X and Y are the MSB's of two number and Z is the resultant MSB after adding two numbers then overflow condition is, $\boxed{\bar{X}\,\bar{Y}\,Z + X\,Y\,\bar{Z}}$
- In 2's compliment arithmetic operation, if carry in and carry out from last bit position are different then overflow will occur.

1.5.2 EX-OR Logic Diagram for overflow



where, C_{in} = Carry into MSB

C_{out} = Carry from MSB

[Since, $A \oplus A = 0$ and $A \oplus \bar{A} = 1$]

OBJECTIVE BRAIN TEASERS

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

ANSWERS KEY

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

HINTS & EXPLANATIONS

1. (b)

Given binary sequence,

$$(1100101 \cdot 1011)_2$$

The given sequence can be written as,

$$\underset{1}{\underline{001}}\,\underset{4}{\underline{100}}\,\underset{5}{\underline{101}} \cdot \underset{5}{\underline{101}}\,\underset{4}{\underline{100}}$$

$\therefore$ The octal equivalent value of X is 145.54.

So, option (b) is correct.

2. (c)

Given binary number $(11011)_2$

1. $(33)_8 = (011011)_2$
2. $(27)_{10} =$

$$\begin{array}{c|l|c} 2 & 27 & \\ \hline 2 & 13 & 1 \\ \hline 2 & 6 & 1 \\ \hline 2 & 3 & 0 \\ \hline & 1 & 1 \end{array} \quad \therefore (11011)_2$$

3. $(10110)_{Gray}$

   Now, we need to convert Gray code to binary code.

$$\begin{array}{ccccc} 1 & 0 & 1 & 1 & 0 \\ \oplus & \oplus & \oplus & \oplus & \\ (1 & 1 & 0 & 1 & 1)_2 \end{array}$$

4. $(1\,B)_H = (1\,B)_{16} = 1 \times 16^1 + 11 \times 16^0 = (27)_{10}$

From solution of case 2: $(27)_{10} = (11011)_2$.

So, the given option (c) is correct.

3. (c)

Given, $\quad X = (54)_b$

Also given, $\sqrt{X} = 7 \Rightarrow X = 49$

But, $\quad X = 5b + 4$

$\therefore \quad 5b + 4 = 49$

$5b = 45$

$\therefore \quad b = 9$

4. (b)

ASCII code is 7-bit code.

5. (d)

Decimal	Binary code	Gray code
0 →	0 0 0 0 →	0 0 0 0 (0)
1 →	0 0 0 1 →	0 0 0 1 (1)
2 →	0 0 1 0 →	0 0 1 1 (3)
3 →	0 0 1 1 →	0 0 1 0 (2)
4 →	0 1 0 0 →	0 1 1 0 (6)
5 →	0 1 0 1 →	0 1 1 1 (7)
6 →	0 1 1 0 →	0 1 0 1 (5)
7 →	0 1 1 1 →	0 1 0 0 (4)
8 →	1 0 0 0 →	1 1 0 0 (12)
9 →	1 0 0 1 →	1 1 0 1 (13)

$\therefore$ Addition of all gray code:

$1 + 3 + 2 + 6 + 7 + 5 + 4 + 12 + 13 = 53$

Hence, option (d) is correct.

6. (d)

Given hexadecimal number, 2A0F

$(2A0F)_{16} = 2 \times 16^3 + 10 \times 16^2 + 0 \times 16^1 + 15 \times 16^0$

$= 2 \times 4096 + 10 \times 256 + 0 + 15 = (10767)_{10}$

Hence, option (d) is correct.

7. (d)

Given, Binary Coded Pentary (BCP) number

$$\left[\underset{4}{\underline{100}}\,\underset{2}{\underline{010}}\,\underset{3}{\underline{011}}\,\underset{3}{\underline{001}}\right]$$

$\therefore \quad (4231)_5$ option (d) is correct.

CONVENTIONAL BRAIN TEASERS

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

1. (Sol.)

Introduction to BCD: BCD stands for binary Encoded Digital. In BCD every decimal number is represented by four binary bits.

Ex: 190 in decimal is equivalent to 0001 1001 000 in binary encoded decimal.

0 to 9 in decimal can be represented in binary using four digits and all integers can be represented by these 10 digits.

BCD Addition: In *BCD* addition of two involve following rules:

Step-1: Maximum value of the sum for two digits
= 9(max digit) + 9(max digit 2) + 1 (Previous addition carry) = 9

Step-2: If sum of two BCD digits is less than or equal to 9(1001) without carry then the result is a correct *BCD* number.

Step-3: If the sum of two *BCD* digits is greater than or equal to 10(1010) the result is incorrect *BCD* number perform step 4 for correct *BCD* sum.

Step-4: Add 6(0110) to the result.

Example:

Perform *BCD* addition of two decimal numbers 599 and 84?

BCD	①	②	③
599	0101	1001	1001
+ 984	1001	1000	0100
Sum	1110	10001	1101

here, binary sum of (1), (2) and (3) are greater than 1010, so from step 3, 4 in the rules specified for binary addition odd correction factor 0110.

Carry 1	Carry 1	Carry 1	
Result	1110	10001	1101
+ 6	0110	0110	0110
End carry 1	$0101_{(5)}$	$1000_{(8)}$	$0011_{(3)}$

∴ Result of BCD addition is 1583.

■■■■