Codes

* **Binary codes**

— Usually, the digital data is represented, stored and transmitted as groups of binary digits (bits). The group of bits, also known as binary codes, represent both numbers and letters of the alphabets as well as many special characters and control functions.

— They are classified as numeric or alphanumeric.

* **Classification of Binary Codes**:

— The different binary codes can be classified as:

1) Weighted codes
2) Non-weighted codes
3) Reflective codes
4) Sequential codes
5) Alphanumeric codes
6) Error detecting & correcting codes.

1) **Weighted codes**: In weighted codes, each digit position of the number represents a specific weight. eg: In weighted binary codes each digit has a weight 8, 4, 2, 1.

2) **Non-weighted Codes**: Non-weighted codes are not assigned with any weight to each digit position i.e., each digit position within the number is not assigned fixed value.

3) Reflective Codes: A code is said to be reflective when the code for 9 is the complement for the code for 0, 8 for 1, 7 for 2, 6 for 3 and 5

4) Sequential Codes: In sequential codes each succeeding code is one binary number greater than its preceeding code.

5) Alphanumeric codes: The codes which consists of both numbers and alphabetic characters are called alphanumeric codes.

6) Error detecting and Correcting codes: When the digital information in the binary form is transmitted from one circuit or system to another circuit or system an error may occur i.e a signal corresponding to 0 may change to 1 or vice-versa due to presence of noise.

- To maintain the data integrity between transmitter and receiver, extra bit or more than one bit are added in the data. These extra bits allow the detection and some times correction of error in the data. The data along with the extra bit/bits form the code.

- Codes which allow only error detection are called error detecting codes and codes which allow error detection and correction are called error detecting and correcting codes.

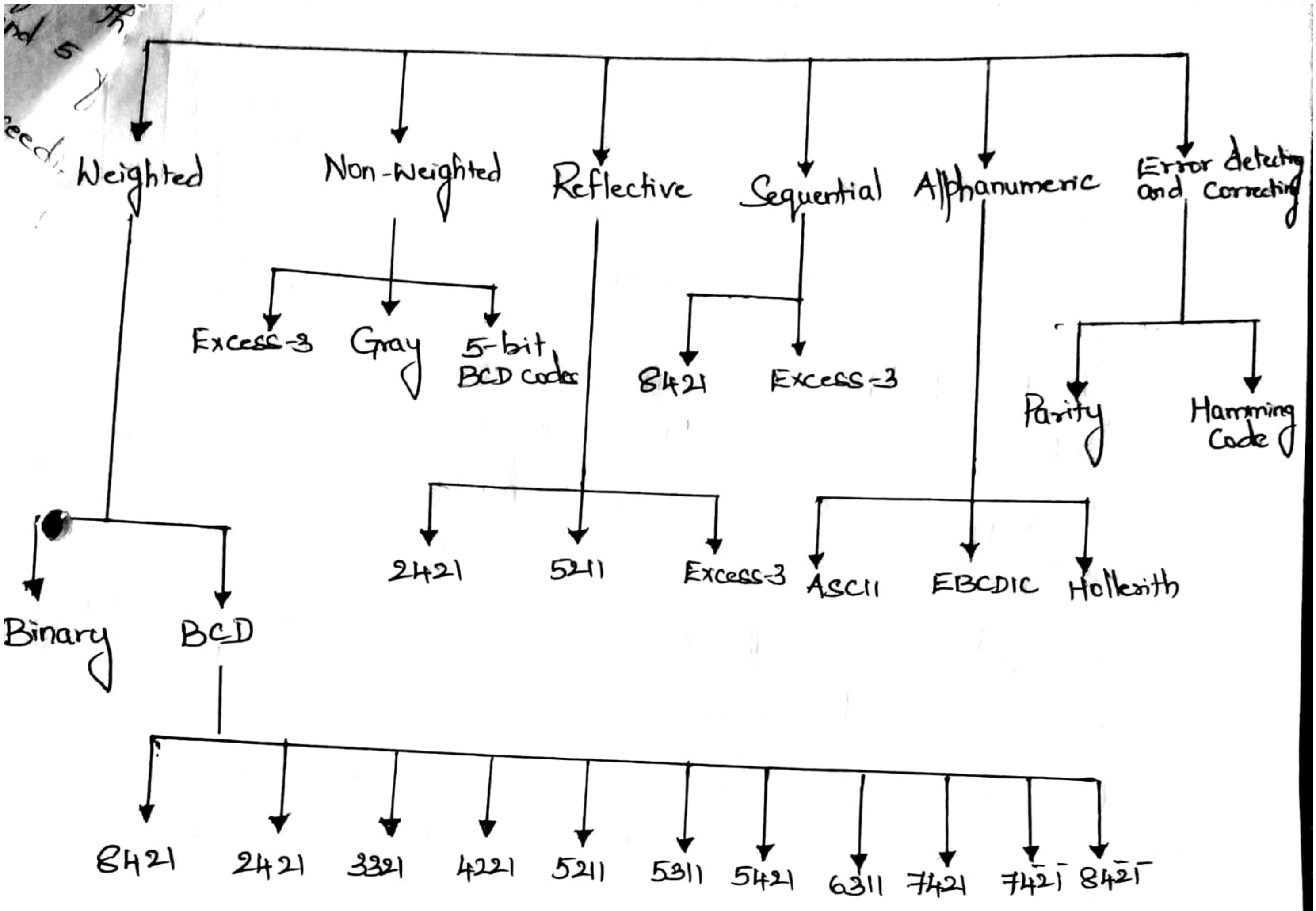

Fig: Classification of various binary codes

* BCD (8-4-2-1)

- BCD Stands for Binary-coded-decimal. BCD is a numeric code in which each digit of a decimal number is represented by a separate group of 4bits binary number.
- The most Common BCD code is 8-4-21 BCD Since bit-3 has weight 8, bit-2 — 4, bit-1-2 and bit-0 — 1

Decimal digit	BCD code 8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Table: 8-4-2-1 BCD code

- In multidigit coding, each decimal digit is individually coded with 8-4-2-1 BCD code.

Eg: $(58)_{10}$ in BCD code:

5 8

0101 1000

- Total 8-bits are required to encode $(58)_{10}$ in 8-4-2-1 BCD whereas in binary, it requires only 6-digits. Hence BCD is less efficient than binary number system.
- The advantage of a BCD code is that it is easy to convert ~~to~~ decimal.
- One more disadvantage is that arithmetic operati[ons] are more complex than they are in binary.

* BCD Addition:

- Three cases that can occur during addition two BCD numbers are as follows:

Case-1: Sum equals 9 or less with carry

Case-2: Sum equals 9 or less with carry

Case-3: Sum ~~equals~~ greater than 9 with carry 0

Case 1: Sum equals 9 or less with carry 0

Eg. Add 3 and 6 in BCD

Sol:

$$\begin{array}{r} 6 \\ +3 \\ \hline 9 \end{array} \qquad \begin{array}{r} 0110 \\ 0011 \\ \hline 1001 \end{array}$$

- The addition is carried out as in normal binary addition and the sum is 1001.

Case 2: Sum equals 9 or less with carry 1

Eg. Add 8 and 9 in BCD

Sol:

$$\begin{array}{r} 8 \\ +9 \\ \hline 17 \end{array} \qquad \begin{array}{r} 1000 \\ 1001 \\ \hline \underbrace{0001}_{1}\ \underbrace{0001}_{1} \end{array}$$

- In this case, the result is incorrect. To get the correct BCD result correction factor of 6 has to be added to the least significant digit of the sum.

$$\begin{array}{r} 1000 \\ +\ 1001 \\ \hline 0001\ 0001 \\ 0000\ 0110 \\ \hline \underbrace{0001}_{1}\ \underbrace{0111}_{7} \end{array}$$

Case - 3: Sum greater than 9 with carry 0

Eg: Add 6 and 8 in BCD

$$\begin{array}{r} 6 \\ +8 \\ \hline 14 \end{array} \qquad \begin{array}{r} 0110 \\ 1000 \\ \hline 1110 \end{array}$$

- The sum is an invalid BCD number since it is of 2-digits that exceed 9.

Case-1: Sum equals 9 or less with carry 0

Eg. Add 3 and 6 in BCD

Sol:

```
  6      0110
 +3      0011
 ---     ----
  9      1001
```

- The addition is carried out as in normal binary addition and the sum is 1001

Case-2: Sum equals 9 or less with carry 1

Eg. Add 8 and 9 in BCD

Sol:

```
  8      1000
 +9      1001
 ---     ---------
 17      0001 0001
```

$\underbrace{0001}_{1}\ \underbrace{0001}$

- In this case, the result is incorrect. To get the correct BCD result correction factor of 6 has to be added to the least significant digit of the sum.

```
        1000
      + 1001
  ---------
  0001 0001
  0000 0110
  ---------
  0001 0111
```

$\underbrace{0001}_{1}\ \underbrace{0111}_{7}$

Case-3: Sum greater than 9 with carry 0

Eg: Add 6 and 8 in BCD

```
  6      0110
 +8      1000
 ---     ----
 14      1110
```

- The sum is an invalid BCD number since is of 2-digits that exceed 9.

- The Sum has to be corrected by the addition of 6 (0110) to the invalid BCD.

$$
\begin{array}{r}
6 \\
+\,8 \\
\hline
14
\end{array}
\qquad
\begin{array}{r}
0110 \\
1000 \\
\hline
1110 \\
0110 \\
\hline
\underbrace{0001}_{1}\ \underbrace{0100}_{4}
\end{array}
$$

* <u>Summary of BCD addition procedure:</u>

1) Add two BCD numbers using ordinary binary addition.
2) If 4-bit Sum is equal to or less than 9, no correction is needed. The Sum is in proper BCD form.
3) If the 4-bit Sum is equal to or < 9, with carry = 1 or is greater than 9 ~~or if a carry is generated from the four bit Sum~~, the Sum is invalid.
4) To correct the invalid Sum, add 0110_2 to the 4-bit Sum. If a carry results from this addition, add it to the next higher-order BCD digit

Prob: Perform each of the following decimal additions in 8-4-2-1 BCD. (4)

a) $24 + 18$ b) $48 + 58$ c) $175 + 326$ d) $589 + 199$

Sol:

(a)

$$\begin{array}{r} 24 \\ +\,18 \\ \hline 42 \end{array} \qquad \begin{array}{rr} 0010 & 0100 \\ 0001 & 1000 \\ \hline 0011 & 1100 \leftarrow \text{Sum}>9 \\ + & 0110 \\ \hline 0011 & \boxed{1}0010 \\ +\;\;1 & \\ \hline 0100 & 0010 \end{array}$$

(b)

$$\begin{array}{r} 48 \\ +\,58 \\ \hline 106 \end{array} \qquad \begin{array}{rr} 0100 & 1000 \\ 0101 & 1000 \\ \hline 1001 & \boxed{1}0000 \rightarrow \text{Carry } 1 \\ +\;\;1 & 0110 \\ \hline \text{Sum}>9 \leftarrow 1010 & 0110 \\ +\,0110 & \\ \hline \underbrace{0001}_{1}\;\underbrace{0000}_{0} & \underbrace{0110}_{6} \end{array}$$

(c)

$$\begin{array}{r} 175 \\ +\,326 \\ \hline 501 \end{array} \qquad \begin{array}{rrr} 0001 & 0111 & 0101 \\ 0011 & 0010 & 0110 \\ \hline 0100 & 1001 & 1011 \;\text{Sum}>9 \rightarrow \text{add } 6 \\ & & 0110 \\ \hline 0100 & 1001 & \boxed{1}0001 \\ & 1 & \\ \hline 0100 & 1010 & 0001 \\ & 0110 & \\ \hline 0100 & \boxed{1}0000 & 0001 \\ 1 & & \\ \hline \underbrace{0101}_{5} & \underbrace{0000}_{0} & \underbrace{0001}_{1} \end{array}$$

d) 589 + 199 = 788

```
  589        0101  1000  1001
+ 199        0001  1001  1001
-----        ------------------
  788        0110 1 0001 1 0010
             ------------------
             0111  0010  0010
                   0110  0110
             ------------------
             0111  1000  1000
              7     8     8
```

* BCD Subtraction

- Addition of signed BCD numbers can be performed by using 9's or 10's complement methods.
- A negative BCD number can be expressed by taking the 9's or 10's complement.

* Subtraction using 9's complement:

- The 9's complement of a decimal number is found by subtracting each digit in the number from 9.
- In 9's complement subtraction when 9's complement of smaller number is added to the larger number carry is generated.
- The carry is added to the result.
- When larger number is subtracted from smaller one, there is no carry, and the result is in 9's complement form and negative.

Digit	9's Complement
0	9
1	8
2	7
3	6
4	5
5	4
6	3
7	2
8	1
9	0

Eg: Regular subtraction | 9's complement subtraction

a)
$$\begin{array}{r} 8 \\ -2 \\ \hline 6 \end{array}$$

$$\begin{array}{r} 8 \\ +7 \\ \hline \boxed{1}\ 5 \\ \hookrightarrow +1 \\ \hline 6 \end{array}$$

b)
$$\begin{array}{r} 9 \\ -5 \\ \hline 4 \end{array}$$

$$\begin{array}{r} 9 \\ +4 \\ \hline \boxed{1}\ 3 \\ \hookrightarrow +1 \\ \hline 4 \end{array}$$

c)
$$\begin{array}{r} 4 \\ -8 \\ \hline -4 \end{array}$$

$$\begin{array}{r} 4 \\ +1 \\ \hline \boxed{0}\ 5 \\ \downarrow \\ -4 \end{array}$$

9's complement of the result

* **Summary of BCD Subtraction procedure:**

1) Find the 9's complement of a negative number
2) Add two numbers using BCD addition
3) If carry is generated, add carry to the result. The carry is called as end around carry.
4) If no carry, then the result is -ve hence find the 9's complement of the result.

Prob: Perform each of the following decimal Subtractions in 8-4-2-1 BCD using 9's complement method.

a) 79 − 26 b) 89 − 54

a) 79 − 26

9's comple of 26 → 99 − 26 = 73

Sol: 79 − 26 = 53

```
  0111   1001
  0111   0011
  -------------
  1110   1100    1100>9 add 6
         0110
  -------------
  1110 [1] 0010
     1+
  -------------
  1111   0010
  0110           Add 6
  -------------
[1] 0101   0010
 └──────→ +  1   End around carry
  -------------
```

BCD for 53 = $\underbrace{0101}_{5}$ $\underbrace{0011}_{3}$

b) 89 − 54 = 35

9's comple of 54 → 99 − 54 = 45

```
   1000   1001
 + 0100   0100
  -------------
   1101   1100    1100>9 add 6
        + 0110
  -------------
   1101 [1] 0100
      1+
  -------------
   1101   0100    1101>9 add 6
 + 0110
  -------------
[1] 0011   0100
 └──────→    +1
  -------------
```

$\underbrace{0011}_{3}$ $\underbrace{0101}_{5}$

* Subtraction using 10's complement

– The 10's complement of a decimal number is equal to the 9's complement plus 1.

* Steps for 10's complement BCD Subtraction:

1) Find 10's complement of a negative number
2) Add two numbers using BCD addition
3) If carry is generated, discard the carry.
4) If carry is not generated find the 10's complement of the result.

Regular Subtraction	10's complement Subtraction
a) $\begin{array}{r} 8 \\ -2 \\ \hline 6 \end{array}$	$\begin{array}{r} 8 \\ +8 \\ \hline \cancel{1}6 \end{array}$ 10's compl of 2
b) $\begin{array}{r} 9 \\ -5 \\ \hline 4 \end{array}$	$\begin{array}{r} 9 \\ +5 \\ \hline \cancel{1}4 \end{array}$ 10's comple of 5
c) $\begin{array}{r} 4 \\ -8 \\ \hline -4 \end{array}$	$\begin{array}{r} 4 \\ +2 \\ \hline 6 \\ \downarrow \\ -4 \end{array}$ (10's complement of the result)

* Excess-3 code

— Excess-3 code is a modified form of a BCD number.

— The Excess-3 code can be derived from the natural BCD code by adding 3 to each Coded number.

Eg: $(12)_{10} \xrightarrow{BCD}$ 0001 0010 $\xrightarrow{Excess\text{-}3}$ 0100 0100

Decimal	Excess-3 code			
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

— In Excess-3 code, we get 9's complement of a number by just complementing each bit.

— Due to this excess-3 code is called "Self-Complementing code".

Prob: Find the excess-3 code and its 9's complement for the following decimal numbers.

a) 592

$592_{10} \xrightarrow{Excess\text{-}3}$ 1000 1100 0101

* $592_{10} \xrightarrow{9's\ comp}$ 0111 0011 1010

b) 403

$403_{10} \xrightarrow{Excess\text{-}3}$ 0111 0011 0110

$403_{10} \xrightarrow{9's\ comple}$ 1000 1100 1001

(Note:

* Excess-3 Addition

1) Add two Excess-3 numbers
2) If carry = 1 → add 3 to the sum of two digits
   = 0 → Subtract 3

Prob: Perform the excess-3 addition of
a) 8, 6 b) 1, 2

8 $\xrightarrow{xs\text{-}3}$ 11
6 $\xrightarrow{xs\text{-}3}$ 9
14 $\xrightarrow{xs\text{-}6}$ 20
∴ 14+3 = 17
∴ add 3

Sol: 8 + 6 = 14

```
           1 0 1 1        XS-3 for 8
        +  1 0 0 1        XS-3 for 6
        ----------
        [1] 0 1 0 0
Add 3  0 0 1 1  0 0 1 1
       ----------------
       0 1 0 0  0 1 1 1   → Excess-3 for 14
```

b) 1 + 2

```
         0 1 0 0      XS-3 for 1
       + 0 1 0 1      XS-3 for 2
       ---------
         1 0 0 1
Sub 3  - 0 0 1 1
       ---------
         0 1 1 0   → Excess-3 for 3
```

* Excess-3 Subtraction

1) Complement the subtrahend.
2) Add complemented subtrahend to minuend
3) If carry = 1, Result is +ve. Add 3 & end around carry
4) If carry = 0, Result is −ve. Subtract 3.

Prob: Perform the excess-3 subtraction of

a) 8 5, b) 5 8

compl of 5 → $\frac{9}{\frac{-5}{4+3=7}}$

Sol: 8 − 5

```
    1 0 1 1
  + 0 1 1 1   → comple of 5 in excess-3
  ---------
  [1] 0 0 1 0
Add-3   0 0 1 1
  ---------
        0 1 0 1
      →+      1
  ---------
        0 1 1 0 → Excess-3 for +3
```

b) 5 − 8

```
    1 0 0 0
  + 0 1 0 0 → Comple of 8 in excess-3
  ---------
    1 1 0 0
  − 0 0 1 1
  ---------
    1 0 0 1 → Excess-3 for −3
```

$\frac{5}{\frac{-8}{\frac{3}{+3}}}=6$

(0110) = −6

c) 16 + 29

```
      0100   1001      Exs-3 for 16
      0101   1100      XS-3 for 29
      ------------
      1001 [1] 0101    Propagate carry
          ←
      ------------
      1010   0101
    +        0011      Add -3 to correct 010
      ------------
Sub-3 ←  1010   1000   Subtract 3 to 1010
    − 0011
      ------------
      0111   1000      Excess-3 for 45
```

Scanned by CamScanner

c) 103 + 287 = 390

	0100	0011	0110	Excess-3 for 103
	0101	1011	1010	Excess 3 for 287
	1001	1110	$\boxed{1}$ 0000	Propagate Carry
		+1		
	1001	1111	0000	Add 3 to Correct 0000
			+ 0011	
	1001	1111	0011	Subtract 3 to correct 1001 & 1111
−	0011	0011		
	0110	1100	0011	Excess-3 for 390

Prob: Perform the subtraction $645_{10} - 319_{10}$ in excess-3 using the 9's complement method.

Sol: Excess-3 for 645 : 1001 0111 1000

Excess-3 for 319 : 0110 0100 1100

9's complement of 319 : 1001 1011 0011

645 − 319 = 326

	1001	0111	1000	
+	1001	1011	0011	
	$\boxed{1}$ 0010	$\boxed{1}$ 0010	1011	Propagate Carry and add end around Carry.
		+1	+1	
	0011	0010	1100	
+	0011	0011		
	0110	0101	1100	Subtract 3 to correct 1100
			− 0011	
	0110	0101	1001	→ Excess-3 for 326

*. Excess-3 Subtraction using 9's complement

Steps:

1) Take 9's complement of subtrahend
2) Add excess-3 of minuend and excess-3 of Complemented subtrahend.
3) If carry = 1 ; Result is positive. Add end around carry. If any digit is not a valid BCD, subtract 6 from that digit.
4) If carry = 0 ; Result is negative. Ignore carry. Take 1's complement to get true result. If any digit is not valid BCD, subtract 6 from that digit.

Prob: Perform the subtraction $(645)_{10} - (319)_{10}$ in excess-3 using the 9's complement method.

Sol: 9's comple of 319 → 680

Excess-3 of 645 → 1001 0111 1000

Excess-3 of 680 → + 1001 1011 0011

(9 11 3)

$\boxed{1}$ 0010 $\boxed{1}$ 0010 1011

→ + 1

0011 0010 1100 ← Invalid BCD

− 0110

$\underbrace{0011}_{3}$ $\underbrace{0010}_{2}$ $\underbrace{0110}_{6}$

$\therefore (645)_{10} - (319)_{10} = (326)_{10}$

Prob: Perform the subtraction $(526)_{10} - (739)_{10}$ in excess-3 using the 9's complement method

Sol: 9's comple of 739 → 260

260 in excess-3 → 0101 1001 0011

526 in excess-3 → + 1000 0101 1001

$$\begin{array}{r} 0101\ 1001\ 0011 \\ +\ 1000\ 0101\ 1001 \\ \hline 1101\ 1110\ 1100 \\ \underbrace{0010}_{2}\ \underbrace{0001}_{1}\ \underbrace{0011}_{3} \end{array}$$

No carry, so result is negative. Complement the result.

$\therefore (526)_{10} - (739)_{10} = -(213)_{10}$

* Gray Code :

- Gray code is a special case of unit-distance code.
- In unit-distance code, bit patterns for two consecutive numbers differ in only one bit position.
- These codes are also called cyclic codes.
- For Gray code any two adjacent code groups differ only in one bit position.
- The gray code is also called reflected code.

(The two LSBs for 4_{10} through 7_{10} are the mirror images of those for 0_{10} through 3_{10})
Similarly, the three LSBs for 8_{10} through 15_{10} are the mirror images of those for 0_{10} through 7_{10}.

- In general, the n-least significant bits for 2^n through $2^{n+1}-1$ are the mirror images of those for 0 through $2^n - 1$.

Decimal Code	Gray code
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101
10	1111
11	1110
12	1010
13	1011
14	1001
15	1000

* Gray to Binary Conversion:

The gray to binary code conversion can be achieved using the following steps:

1) The MSB of the binary number is the same as the most significant bit of the gray code number.
2) To obtain the next binary digit, perform an exclusive-OR-operation between the bit just written down and the next gray code bit.
3) Repeat step-2 until all gray code bits have been XORed with binary digits. The sequence of bits that have been written are the binary equivalent of the gray-code.

Prob: Convert gray code 101011 into its binary equivalent.

Sol:

MSB

Gray code 1 0 1 0 1 1

Binary code 1 1 0 0 1 0

$\therefore (101011)_2 = (110010)_2$

Prob: Convert 1110 gray to binary

Sol:

MSB

Gray code 1 1 1 0

Binary code 1 0 1 1

$\therefore (1110)_2 = (1011)_2$

* Binary to Gray Conversion:

Let us represent a binary number as:

$B_1\ B_2\ B_3\ B_4 \cdots B_n$ and its equivalent gray code as $G_1\ G_2\ G_3 \cdots G_n$

With this representation gray code bits are obtained from the binary bits as follows

$$\begin{array}{ccccccccc} B_1 & \xrightarrow{\oplus} & B_2 & \xrightarrow{\oplus} & B_3 & \xrightarrow{\oplus} & B_4 & \xrightarrow{\oplus} & B_5 \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ G_1 & & G_2 & & G_3 & & G_4 & & G_5 \end{array}$$

Prob: Convert 10111011 in binary into its equivalent gray code.

Sol:

$$\begin{array}{lccccccccccccccc} \text{Binary code} & 1 & \xrightarrow{\oplus} & 0 & \xrightarrow{\oplus} & 1 & \xrightarrow{\oplus} & 1 & \xrightarrow{\oplus} & 1 & \xrightarrow{\oplus} & 0 & \xrightarrow{\oplus} & 1 & \xrightarrow{\oplus} & 1 \\ & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ \text{Gray code} & 1 & & 1 & & 1 & & 0 & & 0 & & 1 & & 1 & & 0 \end{array}$$

$\therefore (10111011)_2 = (11100110)_2$

Prob: Convert binary 1011 to gray.

Sol:

$$\begin{array}{lccccccc} \text{Binary code} & 1 & \xrightarrow{\oplus} & 0 & \xrightarrow{\oplus} & 1 & \xrightarrow{\oplus} & 1 \\ & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ \text{Gray code} & 1 & & 1 & & 1 & & 0 \end{array}$$

$\therefore (1011)_2 = (1110)_2$

* · **ASCII Code** (American Standard Code for Information Interchange):

— It is a universally accepted alphanumeric (character) code

— It is a 7-bit code in which the decimal digits are represented by the BCD code preceded by 011.

— Since it is a 7-bit code, it represents $2^7 = 128$ symbols.

Prob: Obtain the ASCII Code for COMPUTER ENGINEERING

Sol:

$\overset{C}{\underbrace{100\ 0011}}$ $\overset{O}{\underbrace{100\ 1111}}$, $\overset{M}{\underbrace{100\ 1101}}$, $\overset{P}{\underbrace{101\ 0000}}$, $\overset{U}{\underbrace{101\ 0101}}$, $\overset{T}{\underbrace{101\ 0100}}$, $\overset{E}{\underbrace{100\ 0101}}$, $\overset{R}{\underbrace{101\ 0010}}$

$\underbrace{100\ 0101}_{E}$ $\underbrace{100\ 1110}_{N}$, $\underbrace{100\ 0111}_{G}$ $\underbrace{100\ 1001}_{I}$, $\underbrace{100\ 1110}_{N}$, $\underbrace{100\ 0101}_{E}$, $\underbrace{100\ 0101}_{E}$ $\underbrace{101\ 0010}_{R}$, $\underbrace{100\ 1001}_{I}$,

$\underbrace{100\ 1110}_{N}$, $\underbrace{100\ 0111}_{G}$

Excess-3 Subtraction using 9's comple

Prob: 243·62
− 684·25

0110 1000 0100 · 00100101

9's comp of 684·25 → 999·99
− 684·25
315·74 $\xrightarrow{XS-3}$ 648·107

243·62 $\xrightarrow{\text{BCD}}$ 0101 0111 0110 · 1001 0101

↓ XS-3

576·95

0101 0111 0110 · 1001 0101
+ 0110 0100 1000 · 1010 0111 (649·10 7)

x1011 x1011 x1111 · [1]0011 x1100

1011 1011 [1]0000 · 0011 1100
+

1011 1100 0000 · 0011 1100
+ 0011

1011 1100 0000 · 0110 1100
−3 −0011 −0011 −0011 −0011

0101 0111 0110 · 1001 0101
+ 0110 0100 1000 · 1010 0111

1011 1011 1110 · [1]0011 x1100
+1

1011 1011 1111 · 0011 1100
− 0011 − 0011 − 0011 + 0011 − 0011

1000 1000 1100 0110 1001

9's comp 0111 0111 0011 1001 0110

$(-773·96)_{10}$