❖ Numbers Systems

The main factor that specifies the type of a system is the base of that system, for example, the binary number has a base equal to (*2*), ternary number has base equal to (*3*), quaternary has base equal to (*4*)… etc. The main systems that will be studied in this subject are:

1- Decimal Numbers

The digits of this system are [*0,1,2,3,4,5,6,7,8*, and *9*], and the base of it is (*10*), to express any number in this system, multiply all the coefficients of this number by the base of a positive powers for the real numbers and negative powers for the fractional numbers. The following examples illustrate how to express the numbers in decimal system.

Ex1/

$340_{10} = 0\times10^0 + 4\times10^1 + 3\times10^2$

$0 + 40 + 300 = 340$

Ex2/

$0.623_{10} = 6\times10^{-1} + 2\times10^{-2} + 3\times10^{-3}$

$0.6 \quad 0.02 \quad 0.003 = 0.623$

Ex3/

23.053_{10}

The real part $23 = 3\times10^0 + 2\times10^1$

$3 + 20 = 23$

The fractional part $0.023 = 0\times10^{-1} + 5\times10^{-2} + 3\times10^{-3}$

$0 \quad 0.05 \quad 0.003 = 0.053$

This means that the number is (23.053)

2- Binary Numbers

This system consists of two digits only, which are (*0 & 1*), the base of it is (*2*). To represent the numbers of this system, the digits or coefficients of the number must be multiplied by (*2*) with positive power for real part and negative power for fractional part. There are two important abbreviations that must be considered, which are [Least Significant Bit (LSB) and Most Significant Bit (MSB)]

11001110011

MSB LSB

Ex4/

$011_2 = 0\times2^2+1\times2^1+1\times2^0$

$0 + 2 + 1 = 3$

Ex5/

$0.01011_2 = 0\times2^{-1}+1\times2^{-2}+0\times2^{-3}+1\times2^{-4}+1\times2^{-5}$

$= 0 + 0.25 + 0 + 0.0625 + 0.03125$

$= 0.34375$

Ex6/

110.01_2

Real part $110 = 1\times2^2+1\times2^1+0\times2^0$

$4 + 2 + 0 = 6$

Fractional part $0.01 = 0\times2^{-1}+1\times2^{-2}$

$0 + 0.25 = 0.25$

This means that the number is $(6.25)_{10}$

It is found that the results are decimal numbers, and the same results are obtained from all systems, if the coefficients of numbers are multiplied by their bases with positive power for real part and negative power for fractional part.

3- Octal number

The digits of this system are [*0, 1, 2, 3,4,5,6,* and *7*], and the base of it is (*8*). The following examples illustrate how the numbers in this system can be expressed.

Ex7/

$$347_8 = 3\times 8^2 + 4\times 8^1 + 7\times 8^0$$
$$192 + 32 + 7 = 231$$

Ex8/

$$0.406_8 = 4\times 8^{-1} + 0\times 8^{-2} + 6\times 8^{-3}$$
$$0.5 + 0 + 0.01171875 \cong 0.512$$

Ex9/

456.157_8

Real part $456 = 4\times 8^2 + 5\times 8^1 + 6\times 8^0$

$$256 + 40 + 6 = 302$$

Fractional part $0.157 = 1\times 8^{-1} + 5\times 8^{-2} + 7\times 8^{-3}$

$$0.125 + 0.078125 + 0.013671875 \cong 0.217$$

4- Hexadecimal Number

The digits of this system are [*0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,* and *F*], and the base of it is (*16*). To represent the numbers of Hexadecimal system, the coefficients of the Hexadecimal number are multiplied by

(16) with positive power for real part and negative power for fractional part. Note that (A =10, B =11, C=12, D=13, E = 14, F=15).

Ex10/

$FA7_{16} = F\times16^2+A\times16^1+7\times16^0$

$3840 + 160 + 7 = 4007$

Ex11/

$0.41_{16} = 4\times16^{-1}+1\times8^{-2}$

$0.25 + 0.015625 \cong 0.266$

Ex12/ 4B6.2E $_{16}$

Real part $4B6_{16} = 4\times16^2+B\times16^1+6\times16^0$

$1024 + 176 + 6 = 1206$

Fractional part $0.2E_{16} = 4\times16^{-1}+E\times16^{-2}$

$0.25 + 0.0546875 \cong 0.305$

❖ ***Conversion between systems***

1- Binary to Decimal and Vice Versa

To understand the conversion of the numbers from binary system to the decimal and from decimal to binary see the following examples:

Ex13/

$110111001_2 = (\ ?\ )_{10}$

$\longrightarrow 1\times2^8+1\times2^7+0\times2^6+1\times2^5+1\times2^4+1\times2^3+0\times2^2+0\times2^1+1\times2^0 = 441_{10}$

There is another method to perform this conversion, which illustrated below 1 1 0 1 1 1 0 0 1

$2^8+2^7 + 2^6+ 2^5+ 2^4+ 2^3+ 2^2 + 2^1+ 2^0 = 441_{10}$

Or 1 1 0 1 1 1 0 0 1

$256 +128+ 64 + 32+ 16+ 8 + 4 + 2+ 1 = 441_{10}$

The above conversion is performed by putting the following sequence (2^N …….. 2^3 2^2 $2^1 + 2^0$) under the binary number, where N equal to the MSB of the binary nos.

Ex14/

$34_{10} = (?)_2$

Decimal number	Operation	Results	Remainder
34	34/2	17	0
17	17/2	8	1
8	8/2	4	0
4	4/2	2	0
2	2/2	1	0
1	1/2	0	1

LSB

Read in the following direction

MSB

This mean that the binary number is $(100010)_2$

Another method can be used to convert number from decimal to binary system by writing the following sequence:

(….. *256* *128* *64* *32* *16* *8* *4* *2* *1*), it must be noted that the largest number of this sequence is less than the given decimal number and this sequence results from (2^N …….. 2^3 2^2 $2^1 + 2^0$), for example $(34)_{10}$ the number in the previous sequence which is smaller than it is (*32*) therefore the conversion sequence will be

32 16 8 4 2 1

$(1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0)_2$

Which is the same as the result was obtained from the first method. For fractional decimal numbers, the number is multiplied by (2) and the result digit after the separator is the first binary digit after the separator in binary system and repeat the multiplication again and until the digits after the separator equal to zero if this results does not occur we continue for some digits after separator of the binary number.

Ex15/ convert the following decimal number $(0.875)_{10}$ and (26.024) to the binary system.

Sol:

$0.875 \times 2 = 1.75$

$0.75 \times 2 = 1.5$ $(0.1\ 1\ 1)_2$

$0.5 \times 2 = 1.0$

For (26.024)

The real part is convert as follow

16 8 4 2 1

$(1\ \ 1\ \ 0\ \ 1\ \ 0)_2$

For fractional part

$0.24 \times 2 = 0.48$

$0.48 \times 2 = 0.96$

$0.96 \times 2 = 1.92 \quad \cong \ (0.001\ 1\ 1)_2$

$0.92 \times 2 = 1.84$

$0.84 \times 2 = 1.68$ And so on

2- Octal to decimal and vice versa

The second conversion that must be considered is that octal to decimal and vice versa. The following two examples illustrate the conversion between these two systems.

Ex16/

$89_{10} = (?)_8$

Decimal number	Operation	Results	Remainder
89	89/8	11	1
11	11/8	1	3
1	1/8	0	1

LSB

Read in the following direction

MSB

Then the number in octal system is $(131)_8$.

Ex17/ Convert $(0.167)_{10}$ to $(?)_8$.

<u>Sol:</u>

$0.167 \times 8 = 1.336$

$0.336 \times 8 = 2.688$

$0.688 \times 8 = 5.504 \qquad \cong (0.12540)_8$

$0.504 \times 8 = 4.032$

$0.032 \times 8 = 0.256$ And so on

Ex18/ Convert $(655.325)_8$ to $(?)_{10}$

Sol: Solving with weights method

655.325

Real part

$(6 \quad 5 \quad 5)_8$

$\rightarrow (6\times 8^2 +5\times 8^2 +5\times 8^2) = (429)_{10}$

Fractional part

$(3 \times 8^{-1} +2 \times 8^{-2}+ 5\times 8^{-3} \cong (0.41602)_{10}$

3- Hexadecimal to decimal and vice versa

To convert from hexadecimal to decimal the real coefficients of hexadecimal number are multiplied by (*16*) with positive power and fractional coefficients are multiplied by (*16*) with negative power. The conversion from decimal to hexadecimal we use either division method or weights method for real numbers and multiplying by (*16*) for fractional numbers.

Ex19/ Convert $(9810)_{10}$ to $(?)_{16}$

Sol:

Decimal number	Operation	Results	Remainder
9810	9810/16	613	2
613	613/16	38	5
38	38/16	2	6
2	2/16	0	2

LSB

Read in the following direction

MSB

Ex20/ Convert $(E768.EF6)_{16}$ to $(?)_{10}$

Sol:

$E \times 16^3 + 7 \times 16^2 + 6 \times 16^1 + 8 \times 16^0.\ E \times 16^{-1} + F \times 16^{-2} + 6 \times 16^{-2} =$ (59240.93506).

Ex21/ Convert $(4076.1796875)_{10}$ to (?)

Real part (4076) Fractional part (0.1796875)

256 16 1 $0.1796875 \times 16 = 2.875$

$(F \quad E \quad C)_{16}$ $0.875 \times 16 = 14$ (2E)

4- Binary to Octal and vice versa

To convert any number from binary to octal system, each three digits are taken together and find the equivalent number in octal system according to the following table:

Octal digits	Binary numbers	Octal digits	Binary numbers
0	000	4	100
1	001	5	101
2	010	6	110
3	011	7	111

Ex22/ Convert $(1101111)_2$ to (?)

Sol:

It must be started from the (LSB) toward (MSB) as follow

(001 101 111)

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

$(1\ \ 5\ \ 7)_8$

Ex23/ Convert $(4037)_8$ to $(?)_2$

Sol:

(4 0 3 7)

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

$(100\ \ 000\ \ 011\ \ 111)_2$

5- Hexadecimal to Binary and vice versa

To convert any number from binary to Hexadecimal system, each four digits are taken together and find the equivalent number in Hexadecimal system according to the following table:

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

Ex24/ Convert $(10011011010010)_2$ to $(?)_{16}$

Sol:

0010	0110	1101	0010
↓	↓	↓	↓
(2	6	D	$2)_{16}$

Ex25/ Convert $(E767)_{16}$ to $(?)_2$

Sol:

E	7	6	7
↓	↓	↓	↓
(1110	0111	0110	$0111)_2$

❖ ***Operation on systems***

1- Addition

The following examples illustrate the addition operation for all systems. For binary system the following rules must be noted

($0+0 = 0,\ 1+0 = 1,\ 0+1 = 1,\ 1+1 = 10$)

Ex26/ Solve the following

$$\begin{array}{r} 110101 \\ +110110 \\ \hline 1101011 \end{array} \qquad \begin{array}{r} 110110 \\ 101101 \\ +111001 \\ \hline 10011100 \end{array} \qquad \begin{array}{r} 111101 \\ 001010 \\ 111000 \\ +000100 \\ \hline 10000011 \end{array}$$

Ex27/ Solve the following in octal system

$$\begin{array}{r} 357 \\ +\ 276 \\ \hline 655 \end{array} \qquad \begin{array}{r} 42667 \\ +\ 65777 \\ \hline 130666 \end{array} \qquad \begin{array}{r} 355677 \\ +\ 76746 \\ \hline 454645 \end{array}$$

It must be noted that the addition in this system is the same as decimal system except that the carry in octal system is (*8*).

Ex28/ Add the following hexadecimal numbers

AB2F	FE17
+12EC	+DBC
BE1B	10BD3

The carry in this system is (*16*).

There is important note for the addition of octal system, which state that when the result of adding two digits is equal or larger than to (*8*) then (*8*) is subtracted from the result of adding and the result of subtraction will represent the final result and the carry is (*8*) which equivalent to (*one*) in the next order as in decimal system, for example:

$$\begin{array}{r} 4577 \\ +5776 \\ \hline 12575 \end{array}$$

The previous note can be applied to the hexadecimal system except that the base of system will be (*16*) for example

$$\begin{array}{r} EF \\ +FFA \\ \hline 10E9 \end{array}$$

2- Subtraction

For binary system the following rules must be considered {0-0 = 0, 1-1 = 0, 1- 0 = 1, 0-1 = not possible except when another digit (*1*) be taken from neighboring}.

Ex29/ subtract the following numbers

```
  11011       111101      1000000
 -10110      -110001     -  110111
 ------      -------     ---------
  00101       001100       001001
```

Ex30/ Subtract the following numbers

```
  750       711        4321
 - 76      - 377      - 2567
 -----     ------     ------
  652       312        1532
```

Ex31/Subtract the following hexadecimal numbers

```
   AB21          32FB
 -  2EF        -   FEF
 -------       -------
   A832          230C
```

3- Multiplication

```
Ex32/  111            110101             000000
    ×  110          × 110101            110101
   -------          --------          + 110101
       000            110101          --------
      111            000000           101011111001
  +  111            110101
   ------
   101010
```

The multiplication in octal system is illustrated in the following example.

Ex32/

```
   57
×  23
-----
  205
+137
-----
 1575
```

1. Multiply (3) by (7) = 21 and subtract (8×2) from (21) = 5, (2) is chosen to be the larger number multiplied by (8) lead to number equal or smaller than (21), now (5) is the result and (2) is the carry.
2. Multiply (2) by (7) plus (2) = 16 and subtract (8×2) from (16) = 0, (0) is the result and (2) is the carry.
3. The result will be (205) of the first multiplication.
4. Repeat the first three steps to obtain the second result (137).
5. Add them by push the second result one space to obtain the result of multiplication operation.

Ex33/

```
   762
×   73
------
  2726

------
 71306
```

Follow the same points to perform the multiplication in hexadecimal system. To understand this operation see the following example.

Ex34/

```
   B2            3EC
×  F3         ×  A2D
-----         ------
  216           32FC
+A6E            7D8
-----         + 2738
 A8F6         ------
              27E87C
```

4- Division

An important remark (if the result of division operation is real and has not carry, then there is no problem but if there is a carry then we put a (,) as in decimal system.

Ex35/

```
        10001
  101 | 1010101
      - 101
        0000101
      -     101
        0000000
```

```
        100100.01
   11 | 1101101
      - 11
        00011
      -    11
        000000100
      -        11
                1
```

Ex36/ Solve the following in octal system

```
        10
   10 | 100
      - 10
        000
```

```
        53.05
   13 | 732
      - 67
        42
      - 41
        100
      - 67
        110
      - 102
        006      and so on
```

Ex37/ Solve

```
        9                      8C
   32 | 1C2             138 | ABC1
      - 1C2                  - 9C0
      -----                  -----
        000                    FC1
                             - EA0
                             -----
                              1210
                            - 1110
                            ------
                              0010
                             - 39
                             -----
                               100
```

❖ ***Complements of Numbers systems***

Each system has two complements according to (base-1)'s complement and base's complement of that system.

1- Binary Numbers

One's Complement for binary system can be obtained by inverting zero to one and one to zero. Two's complement represents (*one's complement + 1*)

2- Octal Numbers

In this system 7'S complement is obtained by subtracting the number from(*7, 77, 777, …, etc.*) While 8'S complement is obtained by adding one to the *7'S* complement.

3- Decimal Numbers

There are *9'S* and *10'S* complements in the decimal system, *9'S* complement is obtained by subtract decimal number from (9*, 99, 99, …, etc*), while *10'S* complement is (*1+9'S complement*).

4- Hexadecimal numbers

This system has *15'S* & *16'S* complements which can be obtain as in the previous systems.In general to find the (*2'S, 8'S, 10'S, and 16'S*) can be found from the following equation:

$$(N)_{r,c} = r^n - N$$

Where n is number of digits in the integer portion of N, and r is the radix of the number.

Ex38/ Find (1'S, 2'S, 7'S, 8'S, 9'S, 10'S, 15'S, 16'S) complements for $(10011101)_2$.

Sol/ For the number $(10011101)_2$

1'S = (01100010), 2'S = 1'S + 1 $\Longrightarrow$ 2'S = (01100011)

Since $(10011101)_2 = (157)_{10}$ then 10'S = $10^3 - 157 = (843)_{10}$

& 9'S = $843 - 1 = (842)_{10}$

$(10011101)_2 = (235)_8$ $\Longrightarrow$ 8'S = $8^3 - (235) = (543)_8$

And 7'S = $543 - 1 = (542)_8$

$(10011101)_2 = (9D)_{16}$ $\Longrightarrow$ 16'S = $16^2 - 9D = (63)_{16}$

In addition, 15'S = $(62)_{16}$.

An important not $8^3 = 512$ must be converted to octal system (*1000*) then subtract (*235*) from (*1000*) in octal system, and $16^2 = 256$ must be converted to hexadecimal system (*100*) then subtract (*9D*) from (*100*) in hexadecimal.

Ex39/ Subtract the following numbers using the principle of complement.

1-)$_2$ 111011 , 2-)$_8$ 3457 3-)$_{10}$ 349 4-)$_{16}$ DEF2

- 100011 - 456 - 67 - FE4

Sol:- 1-)

111011 → 111011 or → 111011

-100011 1'S → + 011100 (+ 1) 2'S → + 011101

011000 1010111 (1)011000

\+ 1 neglected

011000 011000

2-) 3457 → 3457 → 3457

\- 456 7'S → + 321 (+ 1) 8'S → + 322

3001 (1+3)4000 (1+3)4001

1 neglected

3001 3001

3-) 349 → 349 → 349

\- 67 9'S → + 32 (+ 1) 10'S → + 33

282 (1+2)381 (1+2)382

\+ 1 neglected

282 282

```
4-) DEF2   ------->   DEF2          ------->  DEF2
   - FE4   ------->  + 1B (+ 1)     ------->  + 1C
   -----             ----------               -----
   CF0E             (1+C)DE0D                (1+C) DF0E
                     +        1   [neglected]
                     ----------               -----
                       CF0E                   CF0E
```

❖ ***Codes***

Generally, there are two types of codes which are:

1- Weighted

There are numerous weighted codes. Among these are *2421* and *84-2-1* code, by using weighted code the corresponding decimal digits easily determined by adding the weights associated with 1 in the code group.

Decimal digits	Bcd codes 8421	2421	84-2-1	7421	6311
0	0000	0000	0000	0000	0000
1	0001	0001	0111	0001	0001
2	0010	0010	0110	0010	0011
3	0011	0011	0101	0011	0100
4	0100	0100	0100	0100	0110
5	0101	1011	1011	0101	0111
6	0110	1100	1010	0110	1000
7	0111	1101	1001	1000	1001
8	1000	1110	1000	1001	1011
9	1001	1111	1111	1010	1100

2- Non-Weighted

An example of a non-weighted code is the excess-3 code where digit codes is obtained from heir binary equivalent after adding *3*. Thus the code of a decimal *0* is *011*, that of *6* is *001*, etc.

❖ ***Binary Coded Decimal (BCD)***

In this code, four-bit binary number represents each decimal digit. BCD is a way to express each of the decimal digits with a binary code. In the BCD, with four bits we can represent sixteen numbers (*0000* to *1111*), but in BCD code only first ten of these are used (*0000* to *1001*). The remaining six code combinations i.e. (*1010* to *1111*) are invalid in BCD.

The advantages of this code are that it is very similar to decimal system. In addition, it needs to remember binary equivalent of decimal numbers (*0* to *9*) only. On the other hand, there are some disadvantages of this code which are the addition and subtraction of BCD have different rules. The BCD arithmetic is little more complicated. BCD needs more number of bits than binary to represent the decimal number. Therefore, BCD is less efficient than binary.

❖ ***Operation of BCD codecs***

1- Addition

When add any numbers in BCD system, it must be noted that the result must not be larger than (*1001*).

Ex40/ Add the following $(45)_{10} + (67)_{10}$ as BCD

Sol : 45 $\longrightarrow$ 0100 0101, 67 $\longrightarrow$ 0110 0111

```
  0100 0101
+ 0110 0111
  ---------
  1010 1100
+ 0110 0110
  ---------
1 0001 0010
```

When the result of the BCD is larger than 1001 then add 0110 to that result and find the final result

2- Subtraction

As in binary system, the rules of subtraction must be applied to obtain the results.

Ex41/ Find $(678)_{10} + (569)_{10}$, $(678)_{10} - (569)_{10}$ as BCD

Sol: 1-) $(678)_{10}$ $\longrightarrow$ 0110 0111 1000, $(569)_{10}$ $\longrightarrow$ 0101 01101001

```
  0110 0111 1000
+ 0101 0110 1001
  --------------
  1011 1110 0001
+ 0110 0110 0110
  --------------
1 0010 0100 0111
```

When carry occurs from digit to another then add 0110 to that result and find the result.

2-) $(678)_{10} - (569)_{10}$

```
  0110 0111 1000
- 0101 0110 1001
  --------------
  0001 0000 1111
-           0110
  --------------
            1001
```

When the one number take one from its neighbor then subtract 0110 from it to obtain the final result.

❖ Excess-3 codes

The Ex-3 code is also called as XS-3 code. It is non-weighted code used to express decimal numbers. The Excess-3 code words are derived from the *8421* BCD code words adding $(0011)_2$ or $(3)_{10}$ to each code word. The following table convert the decimal digits to EX-3

Decimal digits	Excess-3 codes
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

❖ Operation of Ex-3 codes

1- Addition

Ex42/ Solve the following $(87)_{10} + (45)_{10}$ as Ex-3 code

Sol: (87) $\xrightarrow{\text{to BCD}}$ (1000 0111) $\xrightarrow{\text{to Ex- 3}}$ (1011 1010)

(45) $\xrightarrow{\text{to BCD}}$ (0100 0101) $\xrightarrow{\text{to Ex- 3}}$ (0111 1000)

$$\begin{array}{r} 1011\ 1010 \\ +\ 0111\ 1000 \\ \hline 1\ 0011\ 0010 \end{array} \quad \Rightarrow \quad \begin{array}{r} 1\ 0011\ 0010 \\ +\ 0011\ 0011 \\ \hline 1\ 0110\ 0101 \end{array}$$

Each number produce carry must be added to (0011) to obtain the result.

Ex43/ Solve the following $(43)_{10} + (12)_{10}$ as Ex-3 code

(43) $\xrightarrow{\text{to BCD}}$ (0100 0011) $\xrightarrow{\text{to Ex- 3}}$ (0111 0110)

(12) $\xrightarrow{\text{to BCD}}$ (0001 0010) $\xrightarrow{\text{to Ex- 3}}$ (0100 0101)

0111 0110		1011 1011
+ 0100 0101	⇒	- 0011 0011
1011 1011		1000 1000

Each number does not produce carry then (0011) must be subtracted from it to obtain the result.

1- Subtraction

To understand the subtraction as Ex- 3 code consider the following example

Ex44/ Find $(68)_{10} - (34)_{10}$

(68) $\xrightarrow{\text{to BCD}}$ (0110 1000) $\xrightarrow{\text{to Ex- 3}}$ (1001 1011)

(39) $\xrightarrow{\text{to BCD}}$ (0011 1001) $\xrightarrow{\text{to Ex- 3}}$ (0110 1100)

1001 1011		0010 1111
- 0110 1100	⇒	+0011- 0011
0010 1111		0101 1100

Each number take one from neighbor then (0011) must be subtracted from it and (0011) is added to the number that does not take one from neighbor .

❖ ***Gray codes***

An n-bit Gray code, also called the reflected binary code. Today, Gray codes are widely used to facilitate error correction in digital communications such as digital terrestrial television and some cable TV systems.

❖ ***Conversion from Binary to Gray and vice versa***

The first gray digits is the same as the first binary digits, other bits of gray codes is obtained by adding each pairs of binary bits with each other. While when convert from gray to binary the first digit in binary is the same as the first digits in gray and other digits of binary number can be obtained by adding each binary digits with the next gray digits.

Ex45/ Convert $(1\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1)_B$ to $(?)_G$

Sol: + + + + + ++ + +

$(1\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1)_B$

$(1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1)_G$

Ex46/ Convert $(1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1)_G$ to $(?)_B$

Sol:

$(1\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1\ \ 1)_G$

+ + + + + + + + +

$(1\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 0\ \ 1)_2$

❖ ***Error detecting/correcting codes***

Interferences can change the timing and shape of the signal. If the signal is carrying binary encoded data, such changes can alter the meaning of the data. These errors can be divided into two types: Single-bit error and Burst error.

1- Single-bit Error

The term single-bit error means that only one bit of given data unit (such as a byte, character, or data unit) is changed from *1* to *0* or from *0* to*1*.

1	0	1	1	0	0	0	1	1	Sent
1	0	1	1	1	0	0	1	1	Received

Only five bit changes from *0* to *1*.

2- Burst Error

The term burst error means that two or more bits in the data unit have changed from *0* to *1* or vice-versa. The length of the burst error is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not be corrupted.

1	0	1	1	0	0	0	1	1	Sent
↓	↓	↓	↓	↓	↓	↓	↓	↓	
1	0	1	0	1	0	0	0	1	Received

⟵ Length of burst ⟶

The simplest way to correct this error is simple parity check in its two types (even and odd) parity,

1- Odd parity

One convention, called odd parity, specifies that the parity bit will be added to the incorrect received bits so that the total number of *1* bits (including the parity bit) is odd. For example

Data 1 1 0 1 0 1 1 1

Data + Odd parity 1 1 1 0 1 0 1 1 1

2- Even parity

An alternate convention, called even parity, sets the parity bit so that the total number of 1 bits (including the parity bit) is even. For example

Data 1 1 0 1 0 0 0 1

Data + Even parity 0 1 1 0 1 0 0 0 1