* Octal Number System

— The number System with base eight is known as the octal number system.

* Octal - to - Decimal Conversion

— Any octal number can be converted into its equivalent decimal number using the weights assigned to each octal digit position.

- Prob: Convert $(6327.4051)_8$ into its equivalent decimal number.

Sol:

$$(6327.4051)_8 = 6\times8^3+3\times8^2+2\times8^1+7\times8^0+$$
$$4\times8^{-1}+0\times8^{-2}+5\times8^{-3}+1\times8^{-4}$$
$$= 3072+192+16+7+\frac{4}{8}+0+\frac{5}{512}+\frac{1}{4096}$$

$\therefore (6327.4051)_8 = (3287.5100098)_{10}$

- Prob: Convert the following octal numbers to decimal numbers.

a) $(416)_8$ b) $(360.15)_8$

Sol: $(270)_{10}$ $(240.2031)_{10}$

* Decimal - to - Octal Conversion

— The conversion from decimal to octal is similar to the conversion procedure for base-10 to base-2 conversion. The only difference is that number 8 is used in place of 2 for division in the case of integers and for multiplication in the case of fractional numbers.

Prob: Convert the following decimal numbers to [illegible] numbers:

a) $(234)_{10}$

$$\begin{array}{c|l} 8 & 234 \\ \hline 8 & 29 - 2 \\ \hline & 3 - 5 \end{array}$$

$(234)_{10} = (352)_8$

b) $(2988.6875)_{10}$

Integer part:

$$\begin{array}{c|l} 8 & 2988 \\ \hline 8 & 373 - 4 \\ \hline 8 & 46 - 5 \\ \hline & 5 - 6 \end{array}$$

$(2988)_{10} = (5654)_8$

Fractional part:

Fractional No	prod	Fractional part	Int part
.6875	$.6875 \times 8 = 5.5$	.5	5 MSB
.5	$.5 \times 8 = 4.0$	.0	4

$\therefore (2988.6875)_{10} = (5654.54)_8$

c) $(3287.5100098)_{10} = (6327.4051)_8$

Int

$$\begin{array}{c|l} 8 & 3287 \\ \hline 8 & 410 - 7 \\ \hline 8 & 51 - 2 \\ \hline & 6 - 3 \end{array}$$

MSB

Frac

$.5100098 \times 8 =$	4.0800784	4	.0800784
$.0800784 \times 8 =$	0.6406272	0	.6406272
$.6406272 \times 8 =$	5.1250176	5	.1250176
$.1250176 \times 8 =$	1.0001408	1	.0001408

Note: The conversion for fractional no.s may not be exact. In general, an approximate equivalent can be determined by terminating the process at a desired p[illegible]

Octal-to-Binary Conversion

- Octal numbers can be converted into equivalent binary numbers by replacing each octal digit by its 3-bit equivalent binary.

Octal	Decimal	Binary
0	0	000
1	1	001
2	2	010
3	3	011
4	4	100
5	5	101
6	6	110
7	7	111
10	8	001 000
11	9	001 001
12	10	001 010
13	11	001 011
14	12	001 100
15	13	001 101
16	14	001 110
17	15	001 111

Prob: Convert $(736)_8$ into an equivalent binary.

Sol: $(736)_8 = (111\ 011\ 110)_2$

Prob: $(725.63)_8$ to binary

$= (111\ 010\ 101 . 110\ 011)_2$

Prob: $(364.25)_8$ to binary

$= (011\ 110\ 100 . 010\ 101)_2$

Binary-to-Octal Conversion

- Binary numbers can be converted into equivalent octal numbers by making groups of 3-bits starting from LSB and moving towards MSB for integer part of the number and then replacing each group of 3-bits by its octal representation. For fractional part, the groupings of 3-bits are made starting from the binary point.

Prob: Convert the following binary numbers to octal

a) $(101111001110 \cdot 001100)_2 = (5716 \cdot 14)_8$

b) $(\cdot 111001111)_2 = (\cdot 717)_8$

c) $(001011011110 \cdot 110010100110)_2 = (1336 \cdot 6246)_8$

* Octal Arithmetic :

Addition

— The sum of two octal digits is the same as their decimal sum, provided the decimal sum is less than 8. If the decimal sum is 8 or greater Subtract 8 to obtain the octal digit.

— A Carry of 1 is produced when the decimal sum is corrected this way.

Prob: a) $4_8 + 2_8$ b) $6_8 + 7_8$ c) $1_8 + 7_8$

Sol: $4_8 + 2_8 = 6_8$

$6_8 + 7_8 = (13-8)$

$6_8 + 7_8 = 5_8$ & Carry 1

$1_8 + 7_8 = (8-8)$

$= 0_8$ Carry

d) Add 167_8 and 325_8

Sol:

$$\begin{array}{r} {}^{1}\,{}^{1} \\ 167 \\ 325 \\ \hline (514)_8 \end{array}$$

e) Add 341_8, 125_8, 472_8 & 577_8

Sol:

$$\begin{array}{r} {}^{2}\,{}^{1} \\ 341 \\ 125 \\ 472 \\ 577 \\ \hline (1757)_8 \end{array}$$

Scanned by CamScanner

Note: Octal addition and Subtraction can also be performed by converting the numbers to binary perform addition/Subtraction and convert the result back to octal.

Eg: Add $(23)_8$ and $(67)_8$

Sol:

$$\begin{array}{r} 23 \\ +67 \\ \hline (112)_8 \end{array} \qquad \begin{array}{r} = 010\ 011 \\ 110\ 111 \\ \hline 001\ 001\ 010 \end{array} = (112)_8$$

Eg: Subtract $(37)_8$ from $(53)_8$ using 8-bit representation

Sol:

$$\begin{array}{r} 53 \\ -37 \\ \hline +14 \end{array} \qquad \begin{array}{r} 00101011 \\ +11100001 \\ \hline \boxed{1}\,00001100 \end{array} \text{(2's comp of 37)}$$

discard 014

* Subtraction

* Subtraction with 7's Complement:

— The 7's complement of an octal number is found by subtracting each digit from 7.

Prob: Find 7's complement of $(612)_8$

Sol:

$$\begin{array}{r} 7\ 7\ 7 \\ -\ 6\ 1\ 2 \\ \hline (1\ 6\ 5)_8 \end{array}$$

* Steps for Octal Subtraction using 7's comp. method : (A−B)

Step-1 : Find 7's complement of subtrahend (7's of B)

Step-2 : Add two octal numbers (first number (A) & 7's complement of the second number (B))

Step-3 : If carry is produced in the addition, add carry in the LSB of the sum ;

Step-4 : ~~Otherwise~~ If no carry, find 7's complement of the sum as a result with negative sign.

Prob: Use the 7's complement method of subtraction to compute $176_8 - 157_8$

Sol: 7's comp of 157 →

$$\begin{array}{r} 7\ 7\ 7 \\ -\ 1\ 5\ 7 \\ \hline 6\ 2\ 0 \end{array}$$

Add 176 & 620 →

$$\begin{array}{r} 1\ 7\ 6 \\ +\ 6\ 2\ 0 \\ \hline \boxed{1}\ 0\ 1\ 6 \\ +\ 1 \\ \hline (0\ 1\ 7)_8 \end{array}$$

Scanned by CamScanner

Use the 7's complement method of Subtraction (4)
to compute $153_8 - 243_8$

7's compl. of 243 →
$$\begin{array}{r} 777 \\ -\,243 \\ \hline 534 \end{array}$$

Add 153 & 534 →
$$\begin{array}{r} 153 \\ 534 \\ \hline \boxed{0}\,707 \end{array}$$

7's comp of 707 →
$$\begin{array}{r} 777 \\ -\,707 \\ \hline (-70)_8 \end{array}$$

Subtraction using 8's Complement

The 8's complement of an octal number is found by adding a 1 to the LSB of the 7's comple of an octal number.

Find the 8's complement of $(346)_8$

$$\begin{array}{r} 777 \\ -\,346 \\ \hline 431 \\ +\,1 \\ \hline (432)_8 \end{array}$$

* Steps for octal Subtraction using 8's comp

Step-1: Find 8's complement of Subtrahend

Step-2: Add two octal numbers (first number and 8's complement of the sec. number)

Step-3: If Carry is produced in the addition it is discarded;

Step-4: If no Carry, find 8's complement of the Sum as a result with negative sign.

Prob: Use the 8's complement method of subtraction to compute $516_8 - 413_8$

Sol: 8's comple of 413 →

$$\begin{array}{r} 7\ 7\ 7 \\ -\ 4\ 1\ 3 \\ \hline 3\ 6\ 4 \\ +\ 1 \\ \hline (3\ 6\ 5)_8 \end{array}$$

Add 516 & 365 →

$$\begin{array}{r} 5\ 1\ 6 \\ +\ 3\ 6\ 5 \\ \hline \boxed{1}\ 1\ 0\ 3 \end{array}$$

Discard the carry →

Scanned by CamScanner

Use the 8's complement method of subtraction to compute $316_8 - 451_8$

8's comple of 451 →

$$\begin{array}{r} 777 \\ -451 \\ \hline 326 \\ +1 \\ \hline (327)_8 \end{array}$$

- Add 316 & 328 →

$$\begin{array}{r} 316 \\ +327 \\ \hline \boxed{0}645 \end{array}$$

- 7's compl of 645 →

$$\begin{array}{r} 777 \\ -645 \\ \hline (132)_8 \\ +1 \\ \hline (-133)_8 \end{array}$$

Applications of octal Number System

— It is highly inconvenient to handle long strings of binary numbers. It may cause errors also.

Eg: The binary number 01111110 can easily be rembered as 376 (7-1's) and can be entered as 376 using keys. Since digital circuits can process only 0's & 1's, the octal numbers have to be converted into binary form using special circuits known as oct-to-bin converters before being processed by the digital circuits.

* Hexadecimal Number System

— Hexadecimal number system is very popular in computer uses. This consists of 16-distinct symbols. These are numerals 0 through 9 and alphabets A through F. Since numeric digits and alphabets both are used to represent the digits in the hexadecimal system, therefore, this is an alphanumeric number system.

Hexa	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

Table: Binary & Decimal equivalent of Hexadecimal Numbers.

* Hexadecimal-to-Decimal Conversion:

Prob: obtain decimal equivalent of $(3A\cdot 2F)_{16}$

Sol: $(3A\cdot 2F)_{16} = 3\times 16^{1} + 10\times 16^{0} + 2\times 16^{-1} + 15\times 16^{-2}$

$= 48 + 10 + \frac{2}{16} + \frac{15}{16^2}$

$(3A\cdot 2F)_{16} = (58\cdot 1836)_{10}$

Prob: $(1A82)_{16}$

$(1A82)_{16} = 1\times 16^{3} + 10\times 16^{2} + 8\times 16^{1} + 2\times 16^{0}$

$= 4096 + 2560 + 128 + 2$

$(1A82)_{16} = (6786)_{10}$

Decimal - to - Hexadecimal Conversion:

1. Convert the following decimal numbers to hexadecimal numbers:

a) $(0.625)_{10}$

$0.625 \times 16 = 10$ A

$(0.625)_{10} = (.A)_{16}$

b) $(2824.725)_{10}$

16	2824	
16	176 – 8	
	11 – 0	

$(B08)_{16}$

$.725 \times 16 = 11.6$	.6	B
$.6 \times 16 = 9.6$	.6	9
$.6 \times 16 = 9.6$	.6	9

$(B08.B99)_{16}$

Hexadecimal - to - Binary Conversion:

- Hexadecimal numbers can be converted into equivalent binary numbers by replacing each hex digit by its equivalent 4-bit binary number.

Prob: Convert the following hexadecimal numbers to binary numbers:

a) $(A4C)_{16}$

$(1010\ 0100\ 1100)_2$

b) $(3E7.DA)_{16}$

$(0011\ 1110\ 0111 . 1101\ 1010)_2$

Binary - to - Hexadecimal Conversion:

- Binary numbers can be converted into the equivalent hexadecimal numbers by making groups of 4-bits starting from LSB and moving towards MSB for integer part and then replacing each group of 4-bits by its hexadecimal representation.
- For the fractional part, the above procedure is repeated starting from the bit next to the binary point & moving toward the right

Prob: Convert the following binary numbers to hexadecimal numbers:

a) 1011 1100 1011 = $(BCB)_{16}$

b) 001 0111 0011 . 0011 0010 = $(2F3.32)_{16}$

c) 001 0110 1110 . 1100 1010 0110 = $(2DE.CA6)_{16}$

d) 0001 1111 0001 . 1001 1001 1010 = $(1F1.99A)_{16}$

* Hexadecimal-to-octal Conversion:

— Hexadecimal numbers can be converted to equivalent octal numbers by converting hexa number to equivalent binary and then to octal.

Prob: Convert the following hexadecimal number to octal number:

a) $(B7A4)_{16}$

001 011 011 110 100 100

$(133644)_8$

b) D43E.5A

001 101 010 000 111 110 . 010 110 100

$(152076.264)_8$

c) $(0.BF85)_{16}$

.101 111 111 000 010 100

$(0.577024)_8$

Octal-to-Hexadecimal Conversion:

- Octal numbers can be converted to equivalent hex numbers by converting octal to equivalent binary and then to hex.

Prob: Convert the following octal number to hex no.

a) $(744)_8$

000111100100

$(1E4)_{16}$

b) $(3472.56)_8$

011100111010 . 10111000

$(73A.B8)_{16}$

c) $(247.36)_8$

000010100111 . 01111000

$(A7.78)_{16}$

Hexadecimal Arithmetic

Addition

- The sum of two hexadecimal digits is the same as their equivalent decimal sum, provided the decimal equivalent is less than 16.
- If the decimal sum is 16 or greater, subtract 16 to obtain the hexadecimal digit.
- A carry of 1 is produced when the decimal sum is corrected this way.

Prob: a) $3_{16} + 9_{16}$ b) $9_{16} + 7_{16}$ c) $A_{16} + 8_{16}$

Sol: $3_{16} + 9_{16} = C_{16}$

$9_{16} + 7_{16} = 16 - 16$
$= 0$ Carry 1

$A_{16} + 8_{16} = 18 - 16$
$= 2$
Carry 1

Prob: Add $3F8_{16}$ and $5B3_{16}$

Sol:

$$\begin{array}{ccc} 3 & F & 8 \\ 5 & B & 3 \\ \hline (9 & A & B)_{16} \end{array}$$

* Subtraction

— Hex subtraction is best accomplished using the complement method. The 15's & 16's complements for hex numbers are used like 1's & 2's complements to perform subtraction.

* Subtraction with 15'S complement

— The 15's complement of a hexadecimal number is found by subtracting each digit from 15.

Prob: Find 15's complement of $A9B_{16}$

Sol:

$$\begin{array}{cccc} & 15 & 15 & 15 \\ - & A & 9 & B \\ \hline & 5 & 6 & 4_{16} \end{array}$$

Steps for hex Subtraction using 15's complement method:

Step-1 : Find 15's complement of subtrahend

Step-2 : Add two hex numbers (first no. & 15's comple of the second no.)

Step-3 : If carry is produced in the addition, add carry to the LSB of the sum

Step-4 : If no carry, find 15's complement of the sum as a result with a negative sign.

Prob: Use the 15's complement method of subtraction to compute $B02_{16} - 98F_{16}$

Sol: 15's compl of 98F →

$$\begin{array}{ccc} 15 & 15 & 15 \\ 9 & 8 & F \\ \hline 6 & 7 & 0 \end{array}$$

Add B02 & 670 →

$$\begin{array}{cccc} & 6 & 7 & 0 \\ + & B & 0 & 2 \\ \hline \boxed{1} & 1 & 7 & 2 \\ & & \to + & 1 \\ \hline & (1 & 7 & 3)_{16} \end{array}$$

Prob: Use the 15's complement method to comp[illegible]

$69B_{16} - C14_{16}$

Sol: 15's compl of C14 →

$$\begin{array}{ccc} 15 & 15 & 15 \\ C & 1 & 4 \\ \hline 3 & E & B \end{array}$$

Add 69B & 3EB

$$\begin{array}{cccc} & 3 & E & B \\ & 6 & 9 & B \\ \hline \boxed{0} & A & 8 & 6 \end{array}$$

15's Comple of A86 →

$$\begin{array}{ccc} 15 & 15 & 15 \\ A & 8 & 6 \\ \hline \end{array}$$

$(-579)_{16}$

* Subtraction with 16's complement:

- The 16's comple of a hex no. is found by adding a 1 to the LSB of the 15's comp of a hex number.

Prob: Find the 16's complement of $A8C_{16}$

Sol:

$$\begin{array}{cccc} & 15 & 15 & 15 \\ - & A & 8 & C \\ \hline & 5 & 7 & 3 \\ + & & & 1 \\ \hline \end{array}$$

$(574)_{16}$

Scanned by CamScanner

Steps for hex Subtraction using 16's complement method:

Step-1: Find 16's complement of Subtrahend

Step-2: Add two hex numbers (first no & 16's complement of the 2^{nd} no.)

Step-3: If Carry is produced in the addition it is discarded

Step-4: If no carry, find 16's complement of the sum as a result with negative sign.

Prob: Use the 16's complement method of subtraction to compute $CB2_{16} - 972_{16}$

Sol: 16's comp 972 ⟶

$$
\begin{array}{rrr}
15 & 15 & 15 \\
9 & 7 & 2 \\
\hline
6 & 8 & D \\
 & + & 1 \\
\hline
6 & 8 & E
\end{array}
$$

Add CB2 to 68E

$$
\begin{array}{rrrrr}
 & & 6 & 8 & E \\
+ & & C & B & 2 \\
\hline
 & & 1 & 1 & \\
 & \boxed{1} & 3 & 4 & 0 \\
\hline
 & & 3 & 4 & 0
\end{array}
$$

Carry is ignored

Prob: Use the 16's complement method of subtraction to compute $3B7_{16} - 854_{16}$

Sol: 16's compl of 854 →

$$\begin{array}{rrr} 15 & 15 & 15 \\ 8 & 5 & 4 \\ \hline 7 & A & B \\ & + & 1 \\ \hline 7 & A & C \end{array}$$

Add 3B7 to 7AC →

$$\begin{array}{rrrr} & 7 & A & C \\ + & 3 & B & 7 \\ \hline \boxed{0} & B & 6 & 3 \end{array}$$

16's comp of B63 →

$$\begin{array}{rrrr} & 15 & 15 & 15 \\ - & B & 6 & 3 \\ \hline & 4 & 9 & C \\ & & + & 1 \\ \hline \end{array}$$

$(-49D)_{16}$

* Signed Binary Numbers:

* Sign-Magnitude Representation:

- In the decimal number system a plus (+) sign is used to denote a positive number and a minus (−) sign for denoting a negative number.
- This representation of numbers is known as signed number.
- Digital circuits can understand only two symbols, 0 & 1. Therefore, the same symbols are to be used to indicate the sign of the number also.
- Normally, an additional bit is used as the sign bit and it is placed as the MSB.
- '0' used to represent a '+ve' number and
- '1' used to represent a '−ve' number.

Eg: 8-bit signed number: 01000100

$(01000100)_2 \rightarrow$ +ve number & its value (magnitude) is $(1000100) = (68)_{10}$.

$(11000100)_2 \rightarrow (-68)_{10}$

- This kind of representation for signed numbers is known as sign-magnitude representation.

Prob: Find the decimal equivalent of the following binary numbers assuming sign-magnitude representation of the binary numbers

a) 101100

Sol: Sign bit is 1, Which means the number is -ve

Magnitude = $01100 = (12)_{10}$

$\therefore (101100)_2 = (-12)_{10}$

b) $(0111)_2 = (+7)_2$

c) $(1111)_2 = (-7)_2$

* 1's Complement Representation

- In a binary number, if each 1 is replaced by 0 and each 0 by 1, the resulting number is known as the one's complement of the number.
- In fact, both the numbers are complement of each other
- (If one of these numbers is +ve, then the other number will be -ve with the same magnitude and vice-versa).

Eg: $(0101)_2 \xrightarrow{\text{represents}} (+5)_{10}$

$(1010)_2 \xrightarrow{\text{represents}} (-5)_{10}$ in this representation

- This method is used for representing Signed numbers.

Prob: Find the 1's complement of the following binary numbers.

a) 0100111001 b) 11011010

Sol: a) 1011000110 b) 00100101

Prob: Represent the following numbers in 1's complement form.

a) +7 & −7 b) +8 & −8 c) +15 & −15

Sol: In 1's complement representation

a) $+7 = (0111)_2$ & $-7 = (1000)_2$

b) $+8 = (01000)_2$ & $-8 = (10111)_2$

c) $+15 = (01111)_2$ & $-15 = (10000)_2$

Note: For an n-bit number, the maximum +ve & −ve numbers that can be represented in 1's complement representation is $+/-(2^{n-1}-1)$

* Two's complement Representation:

- If '1' is added to 1's complement of a binary number, the resulting number is known as the "two's complement of the binary number.

Eg: 0101 $\xrightarrow{2\text{'s comp}}$ 1011
$(+5)_{10}$ $(-5)_{10}$ in 2's complement representation.

- In this representation also, if the MSB is 0 the number is +ve, whereas if the MSB is 1, the number is -ve.

- Note: For an n-bit number, the maximum +ve number which can be represented in 2's complement form is $(2^{n-1}-1)$ and the maximum -ve number is -2^{n-1}

Prob: find the 2's complement of the numbers:

i) 01001110 (ii) 00110101

Sol: i) Number 01001110 $(+78)_{10}$
1's comp 10110001
Add 1 1
10110010 $(-78)_{10}$

ii) Number 00110101 $(+53)_{10}$
1's comp 11001010
Add 1 1
11001011 $(-53)_{10}$

b: Represent $(-17)_{10}$ in

(i) Sign- magnitude

(ii) One's Complement

(iii) Two's complement representation.

: The minimum number of bits required to represent $(+17)_{10}$ in signed number format is six.

$$(+17)_{10} = (010001)_2$$

$\therefore$ $(-17)_{10}$ is represented by:

(i) Sign- magnitude form : $(110001)_2$

(ii) One's complement form : $(101110)_2$

(iii) Two's complement form : $(101111)_2$

Table: Sign-magnitude, 1's and 2's complement Representation using 4-bits

Decimal Number	Binary Number		
	Sign-magnitude	One's Complement	Two's Complement
0	0000	0000	0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	0101	0101
6	0110	0110	0110
7	0111	0111	0111
−8	—	—	1000
−7	1111	1000	1001
−6	1110	1001	1010
−5	1101	1010	1011
−4	1100	1011	1100
−3	1011	1100	1101
−2	1010	1101	1110
−1	1001	1110	1111
−0	1000	1111	—

* Error Detection and Correction: Hamming 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. This means 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 sometimes correction of error in the data.
- The data along with the extra bit/bits forms 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.

* Error-Detecting codes:

- When a digital information is transmitted, it may not be received correctly by the receiver.

Eg: Consider BCD code corresponding to decimal-9 i.e 1001

Case-1:- This is transmitted and received as 1011.

- Since 1011 is an invalid BCD code, it may be detected by the receiver.

Case-2: If it is received as 0001 which is a valid BCD for decimal 1, the receiver will interpret as decimal-1 and the error is not detected.

* Parity Bit

- For detection of error an extra bit known as parity bit is attached to each code word to make the number of 1's in the code even (Even parity) or odd (Odd parity).

BCD code				BCD code with even parity					BCD code with odd parity				
D	C	B	A	P	D	C	B	A	P	D	C	B	A
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	1	1	0	0	0	1	0	0	0	0	1
0	0	1	0	1	0	0	1	0	0	0	0	1	0
0	0	1	1	0	0	0	1	1	1	0	0	1	1
0	1	0	0	1	0	1	0	0	0	0	1	0	0
0	1	0	1	0	0	1	0	1	1	0	1	0	1
0	1	1	0	0	0	1	1	0	1	0	1	1	0
0	1	1	1	1	0	1	1	1	0	0	1	1	1

- The parity bit 1 or 0 is attached to the code to be transmitted at the transmitter end and the parity of the received (n+1)-bit word is checked at the receiving end.
- If there is only one error, the erroneous code is detected at the receiving end by parity check.
- If odd number of bits are transmitted erroneously, then also the parity check will detect the incorrect code

But if there are even number of bits received incorrectly, this method will not detect the error.

: Write the ASCII code of the word "COMPUTER" Using even parity.

l:

	P	7	6	5	4	3	2	1
C	1	1	0	0	0	0	1	1
O	1	1	0	0	1	1	1	1
M	0	1	0	0	1	1	0	1
P	0	1	0	1	0	0	0	0
U	0	1	0	1	0	1	0	1
T	1	1	0	1	0	1	0	0
E	1	1	0	0	0	1	0	1
R	1	1	0	1	0	0	1	0

* Error Correcting Codes: Hamming Code

- Hamming Code is Called Error detecting and correcting code.
- The code uses number of parity bits (depending on the number of information bits) located at Certain positions in the code group.

Assigning Values to Parity bits: (For a 7-bit code

— How to determine 1 or 0 value to each parity

Assignment of P1 : Parity bit P_1 Checks bit locations
1, 3, 5 and 7 and assigns
P1 according to even or odd parity

Assignment of P2 : 2, 3, 6 and 7

Assignment of P4 : 4, 5, 6 and 7

b: Encode the binary word 1011 into 7-bit even parity hamming code.

D_7	D_6	D_5	P_4	D_3	P_2	P_1
1	0	1		1		

P_1 – ? 1 1 1 → 1

P_2 – ? 1 0 1 → 0

P_4 – ? 1 0 1 → 0

∴ The Hamming Code is : 1010101

5: Assume that the even parity hamming code ~~is~~ 0110011 is transmitted and that 0100011 is received. Determine bit location where error has occurred using received code.

Sol: Received Code:

D_7	D_6	$\boxed{D_5}$	P_4	D_3	P_2	P_1
0	1	$\boxed{0}$	0	0	1	1

P_1 — 1 0 0 0 $\longrightarrow$ 1 (LSB)

P_2 — 1 0 1 0 $\longrightarrow$ 0

P_4 — 0 0 1 0 $\longrightarrow$ 1

The resultant word is: 1 0 1 = 4+1 = 5

$\therefore$ Error is at location 5

$\therefore$ The correct code is: 0110011

6: The Hamming Code 101101101 is received. Correct it if any errors. There are four parity bits and odd parity is used.

Sol:

Received Codeword

D_9	P_8	D_7	D_6	D_5	P_4	D_3	P_2	P_1
1	0	1	1	0	1	1	0	1

$P_1 \rightarrow 11011 \rightarrow 1$ LSB

$P_2 \rightarrow 0111 \rightarrow 0$

$P_4 \rightarrow 1011 \rightarrow 0$

$P_8 \rightarrow 01 \rightarrow 0$

Error word ~~Parity Code~~ $= 0001 = (1)_{10}$

Scanned by CamScanner

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.

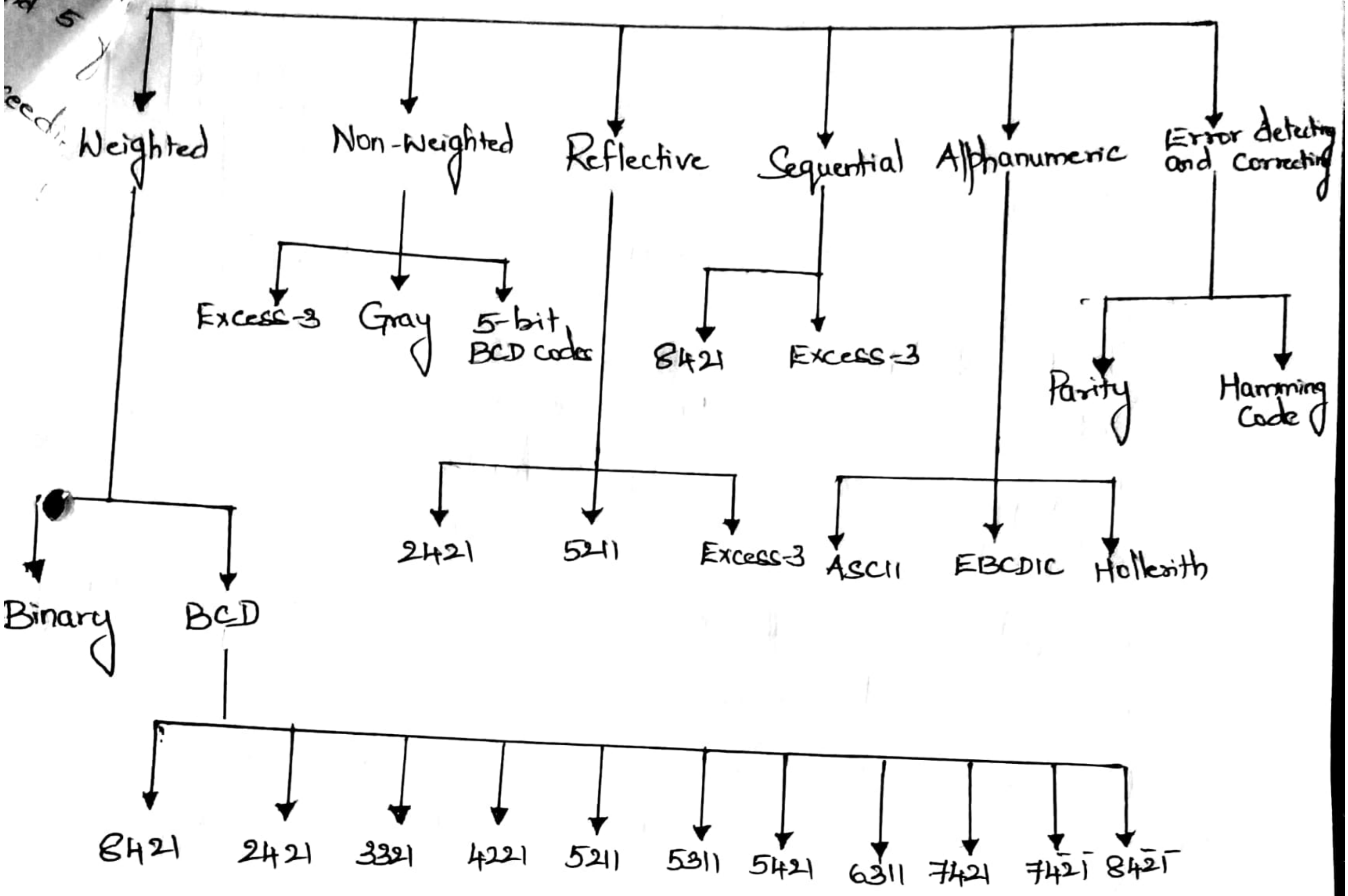

Fig: Classification of various binary codes

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

- 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 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.

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

* **Summary of BCD addition procedure:**

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)

24	0010	0100
+18	0001	1000
42	0011	1100 ← Sum>9
	+	0110
	0011 $\boxed{1}$	0010
	+ 1	
	0100	0010

(b)

48	0100	1000
+58	0101	1000
106	1001 $\boxed{1}$	0000 → Carry 1
	+ 1	0110
Sum>9 ←	1010	0110
	+ 0110	
0001	0000	0110
1	0	6

(c)

175	0001	0111	0101
+326	0011	0010	0110
501	0100	1001	1011 Sum>9 → add 6
			0110
	0100	1001 $\boxed{1}$	0001
		1	
	0100	1010	0001
		0110	
	0100 $\boxed{1}$	0000	0001
	1		
	0101	0000	0001
	5	0	1

d) 589 + 199 = 788

589		0101	1000	1001
+199		0001	1001	1001
788		0110	$\boxed{1}$0001	$\boxed{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

⑤

Eg: Regular subtraction   9's complement subtraction

a)
$$\begin{array}{r} 8 \\ -2 \\ \hline 6 \end{array} \qquad \begin{array}{r} 8 \\ +7 \\ \hline \boxed{1}\ 5 \\ +1 \\ \hline 6 \end{array}$$

b)
$$\begin{array}{r} 9 \\ -5 \\ \hline 4 \end{array} \qquad \begin{array}{r} 9 \\ +4 \\ \hline \boxed{1}\ 3 \\ +1 \\ \hline 4 \end{array}$$

c)
$$\begin{array}{r} 4 \\ -8 \\ \hline -4 \end{array} \qquad \begin{array}{r} 4 \\ +1 \\ \hline \boxed{0}\ 5 \\ = -4 \end{array}$$

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 → $\begin{array}{r} 99 \\ -26 \\ \hline 73 \end{array}$

Sol:

$$\begin{array}{r} 79 \\ -26 \\ \hline 53 \end{array}$$

$$\begin{array}{rrrl}
 & 0111 & 1001 & \\
 & 0111 & 0011 & \\
\hline
 & 1110 & 1100 & 1100>9 \text{ add } 6 \\
 & & 0110 & \\
\hline
 & 1110\ \boxed{1} & 0010 & \\
 & 1+ & & \\
\hline
 & 1111 & 0010 & \\
 & 0110 & & \text{Add } 6 \\
\hline
\boxed{1} & 0101 & 0010 & \\
 & + & 1 & \text{End around carry} \\
\hline
\end{array}$$

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

b) $\begin{array}{r} 89 \\ -54 \\ \hline 35 \end{array}$

9's comple of 54 → $\begin{array}{r} 99 \\ -54 \\ \hline 45 \end{array}$

$$\begin{array}{rrrl}
 & 1000 & 1001 & \\
+ & 0100 & 0100 & \\
\hline
 & 1101 & 1100 & 1110>9 \text{ add } 6 \\
 & & +0110 & \\
\hline
 & 1101\ \boxed{1} & 0100 & \\
 & 1+ & & \\
\hline
 & 1101 & 0100 & 1101>9 \text{ add } 6 \\
 & +0110 & & \\
\hline
\boxed{1} & 0011 & 0100 & \\
 & & +1 & \\
\hline
 & \underbrace{0011}_{3} & \underbrace{0101}_{5} & \\
\end{array}$$

* 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) $8 - 2 = 6$	$8 + 8$ (10's compl of 2) $= \cancel{1}6$
b) $9 - 5 = 4$	$9 + 5$ (10's comple of 5) $= \cancel{1}4$
c) $4 - 8 = -4$	$4 + 2 = 6$ (10's complement of the result) $\downarrow$ -4

* 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: $(12)_{10} \xrightarrow{BCD} 0001\ 0010 \xrightarrow{\text{Excess-3}} 0100\ 0101$

— 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".

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

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

a) 592

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

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

b) 403

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

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

(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

Sol: 8 + 6 = 14

```
          1 0 1 1      XS-3 for 8
        + 1 0 0 1      XS-3 for 6
        ---------
      1 [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
```

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

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 → $\begin{array}{r} 9 \\ -5 \\ \hline 4+3=7 \end{array}$

Sol: 8 − 5

```
   1 0 1 1
 + 0 1 1 1   → compl 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

$\begin{array}{r} 5 \\ -8 \\ \hline 3 \\ +3 \end{array} = 6$

(0110) = −6

```
    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
```

c) 16 + 29

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

c) 103 + 287 = 390

103	0100	0011	0110	Excess-3 for 103
+ 287	0101	1011	1010	Excess-3 for 287
390	1001	1110	$\boxed{1}$ 0000	Propagate Carry
	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	1001	0111	1000	
− 319	+ 1001	1011	0011	
326	$\boxed{1}$ 0010	$\boxed{1}$ 0010	1011	Propagate carry and add end around carry.
			+1	
	0010	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

0011 0010 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

$$\begin{array}{r} 0101\ 1001\ 0011 \\ +\ 1000\ 0101\ 1001 \\ \hline 1101\ 1110\ 1100 \end{array}$$

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

$\underbrace{0010}_{2}\ \underbrace{0001}_{1}\ \underbrace{0011}_{3}$

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

* 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.

<u>Prob</u>: Convert gray code 101011 into its binary equivalent.

<u>Sol</u>:

	MSB					
Gray code	1	0	1	0	1	1
Binary code	1	1	0	0	1	0

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

<u>Prob</u>: Convert 1110 gray to binary

<u>Sol</u>:

	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:

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

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

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

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 → BCD → 0101 0111 0110 . 1001 0101
↓ xs-3
576.95

+ 0110 0100 1000 . 1010 0111 (649.10 7)

x1011 x1011 x1111 . $\boxed{1}$0011 x1100

1011 1011 $\boxed{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 . $\boxed{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

$(-773.96)_{10}$