CHAPTER – 2

NUMBER SYSTEM AND CODES

Number System

Computers communicate and operate in binary digits 0 and 1; on the other hand, human beings generally use the decimal system with ten digits, from 0 to 9. Other number systems are also used, such as octal with eight digits, from 0 to 7, and hexadecimal (Hex) with digits from 0 to 15. In the hexadecimal system, digits 10 through 15 are designated as A through F, respectively, to avoid confusion with the decimal numbers 10 to 15.

A positional scheme is usually used to represent a number in any of the number systems. This means that each digit will have its value according to its position in a number. The number of digits in a position is also referred to as base or radix. For example, the binary system has base 2, the decimal system has base 10, and the hexadecimal system has base 16.

Different Number Systems

1. **Decimal number system (base or radix 10)**

   It uses 10 digits i.e., 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. This means it uses 10 different symbols to represent the number.

   $$\overset{3}{6}\,\overset{2}{5}\,\overset{1}{2}\,\overset{0}{3} = 6\times10^3 + 5\times10^2 + 2\times10^1 + 3\times10^0$$

   $$\overset{1}{6}\,\overset{0}{5}\,.\,\overset{1}{2}\,\overset{-2}{3} = 6\times10^1 + 5\times10^0 + 2\times10^{-1} + 3\times10^{-2}$$

2. **Binary number system (radix 2)**

   It uses only 2 different symbols to represent number i.e., 0 and 1.

   $$\overset{6}{1}\,\overset{3}{1}\,\overset{4}{0}\,\overset{3}{1}\,\overset{2}{1}\,\overset{1}{1}\,\overset{0}{1} = 1\times2^6 + 1\times2^5 + 0\times2^4 + 0\times2^4 + 1\times2^2 + 1\times2^0$$

   $$\overset{3}{1}\,\overset{2}{1}\,\overset{1}{0}\,\overset{0}{0}\,.\,\overset{-1}{1}\,\overset{-2}{1}\,\overset{-3}{1} = 1\times2^3 + 1\times2^2 + 0\times2^2 + 0\times2^1 + 1\times2^{-1} + 1\times2^{-2} + 1\times2^{-3}$$

3. **Octal number system (base 8)**

   This system uses 8 different symbols i.e., 0, 1, 2, 3, 4, 5, 6, 7

   $$\overset{3}{5}\,\overset{2}{7}\,\overset{1}{3}\,\overset{0}{2} = 5\times8^3 + 7\times8^2 + 3\times8^1 + 2\times8^0$$

4. **Hexadecimal number system (base 16)**

This system uses 16 different symbols. The first 10 digits is same as in decimal number system and the latter A, B, C, D, E, and F are used for digits 10, 11, 12, 13, 14, and 15 respectively.

$$\overset{3}{C}\,\overset{2}{2}\,\overset{1}{7}\,\overset{0}{6} = 12 \times 16^3 + 2 \times 16^2 + 7 \times 16^1 + 6 \times 16^0$$

Decimal	Binary	Octal	Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

2.1 Binary to Decimal and Decimal to Binary Conversions

- **Binary to Decimal Conversion**

$$(\overset{4}{1}\,\overset{3}{1}\,\overset{2}{0}\,\overset{1}{0}\,\overset{0}{1})_2 = 1\times2^4 + 1\times2^3 + 0\times2^1 + 1\times2^0$$

$$= 16 + 8 + 1$$

$$= (25)_{10}$$

Decimal to Binary Conversion

For conversion of $(13)_{10}$ to binary,

Quotient	Remainder
$13 \div 2 = 6$	1
$6 \div 2 = 3$	0
$3 \div 2 = 1$	1
$1 \div 2 = 0$	1
$\therefore (13)_{10} = (1101)_2$	

In case of a number with both integer and fraction part, for integer, repeated division by 2 is carried out whereas for fraction, repeated multiplication by 2 is carried out.

For example, to convert $(25.625)_{10}$ into its binary equivalents, following operations are performed.

i. **For integer part,**

Quotient	Remainder
$25 \div 2 = 12$	1
$12 \div 2 = 6$	0
$6 \div 2 = 3$	0
$3 \div 2 = 1$	1
$1 \div 2 = 0$	1
$\therefore (25)_{10} = (11001)_2$	

ii. **For fractional part,**

	Carry
$0.625 \times 2 = 1.25$	1
$0.25 \times 2 = 0.5$	0
$0.5 \times 2 = 1.0$	1
$\therefore (0.625)_{10} = (11001.101)_2$	

$\therefore (25.625)_{10} = (11001.101)_2$

2.2 Octal, Hexadecimal Number System and Conversion

- **Decimal to Octal Conversion**

  To convert a number in decimal to a number in octal, we have to divide the decimal number by 8 unless zero is obtained.

  For conversion of $(426)_{10}$ to $(?)_8$,

Quotient	**Remainder**
$426 \div 8 = 53$	2
$53 \div 8 = 6$	5
$6 \div 8 = 0$	6
$\therefore (426)_{10} = (652)_8$	

  For a number with both integer and fraction part, for integer, repeated division by 8 is carried out whereas for fraction, repeated multiplication by 8 is carried out.

  For $(0.96)_{10} \rightarrow (?)_8$,

$0.96\times8 = 7.68$	7	
$0.68\times8 = 5.44$	5	
$0.44\times8 = 3.52$	3	
$0.52\times8 = 4.16$	4	
$\therefore (0.96)_{10} = (0.7534)_8$		

- **Octal to Decimal Conversion**

$$\overset{2}{3}\,\overset{1}{4}\,\overset{0}{7}$$

$$(347)_8 = 3\times8^2 + 4\times8^1 + 7\times8^0$$
$$= 3\times64 + 4\times8 + 7$$
$$= (231)_{10}$$

- **Hexadecimal to Decimal Conversion**

$$\overset{3}{8}\,\overset{2}{6}\,\overset{1}{5}\,\overset{0}{F}$$

$$(865F)_{16} = 8\times16^3 + 6\times16^2 + 5\times16^1 + F\times16^0$$
$$= 8\times16^3 + 6\times16^2 + 5\times16^1 + 15\times16^0$$
$$= (46687)_{10}$$

- **Decimal to Hexadecimal Conversion**

  For a decimal number with only integer part, we have to divide the decimal number by 16 repeatedly until zero is obtained.

For conversion of $(348)_{10}$ to $(?)_{16}$,

$348+16 = 21$	12 (C)
$21+16 = 1$	5
$1 \div 16 = 0$	1
$\therefore (348)_{10} = (15C)_{16}$	

For a decimal number with only fraction part, repeated multiplication of the number by 16 is carried out until certain level of accuracy is achieved as shown in the following example.

For $(0.37)_{10} \rightarrow (?)_{16}$,

	Carry
$0.37\times16 = 5.92$	5
$0.92\times16 = 14.72$	14 (E)
$0.72\times16 = 11.52$	11 (B)
$0.32\times16 = 3.32$	3
$0.32\times16 = 5.12$	5
$0.12\times16 = 1.92$	1
$\therefore (0.37)_{10} = (0.5EB851)_{16}$	

- **Hexadecimal to Binary Conversion**

$(2BF.29B)_{16} \rightarrow (\ )_2$

The conversion from hexadecimal to binary number is achieved by replacing each hexadecimal digit with its 4-bit binary equivalent.

2	B	F	2	9	B
↓	↓	↓	↓	↓	↓
0010	1011	1111	0010	1001	1011

$= (00101011\ 1111.00\ 10100011011)_2$

$= (1010\ 111111.0010100\ 11011)_2$

$(1CD.2A)_{16} \rightarrow (\ )_2$

1	C	D	2	A
↓	↓	↓	↓	↓
0001	1100	1101	0010	1010

$= (000111\ 001101.00101010)_2$

$= (111\ 00\ 1101.\ 001\ 0101)_2$

- **Binary to Hexadecimal Conversion**

  The conversion is accomplished by grouping the binary number into groups of 4-bits each, starting from the binary point and proceeding to the right as well as to the left. Each group is then replaced by its hexadecimal equivalent.

  $(110\ 110111.\ 1011110)_2 \rightarrow (\ )_{16}$

0001	1011	0111	1011	1100
↓	↓	↓	↓	↓
1	B	7	B	C

  $= (1B7.BC)_{16}$

- **Octal to Binary Conversion**

  $(1745.\ 246)_8 \rightarrow (\ )_2$

1	7	4	5	2	4	6
↓	↓	↓	↓	↓	↓	↓
001	111	100	101	010	100	110

  $= (001111100101.010100110)_2$

  $= (1111100101.01010011)_2$

- **Binary to octal**

  $(10110001101011.\ 111100\ 00011)_2 \rightarrow (\ )_8$

010	110	001	101	011	1111	100	000	110
↓	↓	↓	↓	↓	↓	↓	↓	↓
2	6	1	5	3	7	4	0	6

  $= (26153.7406)_8$

- **Octal to Hexadecimal Conversion**

  $(1745.\ 246)_8 \rightarrow (\ )_{16}$

  Ans: $(3E5.530)_{16}$

  Hint: First change to binary system & then to hexadecimal system

- **Hexadecimal to Octal Conversion**

  $(3E5.530)_{16} \rightarrow (\ )_8$

  Ans: $(1745.246)_8$

  Hint: First change to binary system and then to octal system

2.3 Binary Arithmetic 1's Complement and 9's Complement

Before we can understand how a computer performs arithmetic, we have to learn how to add binary numbers because computer circuits process binary numbers. Binary addition is the key to binary subtraction, multiplication, and division.

Binary Addition

The four basic cases of binary addition are:

$0 + 0 = 0$; Sum = 0, Carry = 0

$0 + 1 = 1$; Sum = 1, Carry = 0

$1 + 0 = 1$; Sum = 1, Carry = 0

$1 + 1 = 10$; Sum = 0, Carry = 1

Addition format:

$$\begin{array}{r} \text{Augend} \\ +\text{Addend} \\ \hline \text{Sum} \end{array}$$

EXAMPLE:

1. Add two 8-bit numbers 01010111 and 00110101.

⇒

carry	1	1	1		1	1	1	
	0	1	0	1	0	1	1	1
+	0	0	1	1	0	1	0	1
	1	0	0	0	1	1	0	0

Binary Subtraction

The four cases of binary subtraction are:

$0 - 0 = 0$; difference = 0, borrow = 0

$0 - 1 = 1$; difference = 1, borrow = 1

$1 - 0 = 1$; difference = 1, borrow = 0

$1 - 1 = 0$; difference = 0, borrow = 0

Subtraction format:

$$\begin{array}{r} \text{Minuend} \\ -\ \text{Subtrahend} \\ \hline \textbf{Difference} \end{array}$$

If subtrahend is greater than minuend, then we subtract the minuend from subtrahend and put the negative sign in the result.

EXAMPLE:

1. **Subtract two 8-bit numbers: 11001000 – 01111101**

$$
\begin{array}{r}
1100\ \ 1000 \\
0111\ \ 1101 \\
\hline
0100\ \ 1011
\end{array}
$$

Unsigned Binary Numbers

The binary number which only represents the magnitude is an unsigned binary number. In this representation we forget '+' and '–' sign and concentrate on magnitude (absolute value) of a number. All the bits are used to represent magnitude.

For 8-bit number, the range is

$(0000\ 0000)_2$ to $(1111\ 1111)_2$

or, $(00)_H$ to $(FF)_H$

or, $(0)_{10}$ to $(255)_{10}$

For 16-bit number, total range is

$(0000\ 0000\ 0000\ 0000)_2$ to $(1111\ 1111\ 1111\ 1111)_2$

or, $(0000)_H$ to $(FFFF)_H$

or, $(0)_{10}$ to $(65535)_{10}$

Signed Magnitude Numbers

What to do when data have positive and negative values? In this case, we use signed magnitude number. A signed binary number consists of both sign and the magnitude information. In signed binary number representation, the leftmost bit is sign bit and the remaining is magnitude bits. For positive number, sign bit is 0 and for negative number, sign bit is 1.

Here are some examples of converting sign-magnitude numbers.

$+7 \rightarrow 0000\ 0111$

$-16 \rightarrow 1001\ 0000$

$+25 \rightarrow 0000\ 0000\ 0001\ 1001$

$-128 \rightarrow 1000\ 0000\ 1000\ 0000$

For 8-bit number, the negative numbers are

1000 0001 (–1) to 1111 1111 (–127)

Remember that for 8-bit number, positive numbers are

0000 0001 (+1) to 0111 1111 (+127)

So, for 8-bit arithmetic, the range is –127 to +127.

The main advantage of sign-magnitude numbers is their simplicity. Negative numbers are identical to positive numbers, except for the sign bit. However, sign-magnitude numbers have limited use because they require complicated arithmetic circuits.

Complements

Complements are used in digital computers to simplify the subtraction operation and for logical manipulation. Simplifying operations leads to simpler, less expensive circuits to implement the operations. For base-r system, there are two types of complement.

1. The radix complement (r's complement)
2. The diminished radix complement ((r-1)'s complement)

1. The r's Complement (Radix Complement)

r's complement of $N = r^n - N$

where n = no. of digits, N = Number, r = radix

2. The (r–1)'s Complement (Diminished Radix Complement)

(r–1)'s complement of $N = r^n - r^{-m} - N$ whereN = number, r = radix, n = an integer of n digits, m = a fractional part of m digits.

EXAMPLE:

1. Find 10's complement of $(52520)_{10}$.

⇒ Here, number of digits n = 5

10's complement of $(52520)_{10} = 10^5 – 52520$

$= 47480$

2. Find r's complement of $(0.252)_{10}$.

⇒ As there is no integer part, n = 0

10's complement of $(0.252)_{10} = 10^n – N$

$= 10^0 – 0.252$

$= 1 – 0.252$

$= 0.748$

3. Find 2's complement of $(10101)_2$.

⇒ 2's complement of $(10101)_2 = 2^5 – 10101$

$= (100000) – (10101)_2$

$= 1011$

4. **Find 2's complement of $(0.0110)_2$**

⇒ Here, n = 0

2's complement of $(0.0110)_2$ $= 2^0 - 0.0110$

$= 1 - 0.0110$

$= 0.1010$

5. **Find 9's complement of $(25.639)_{10}$**

⇒ 9's complement of $(25.639)_{10} = 10^2 - 10^{-3} - 25.639$

$= 100 - 0.001 - 25.639$

$= 73.36$

6. **Find 1's complement of $(101100)_2$**

⇒ 1's complement of $(101100)_2 = 2^6 - 2^0 - 101100$

$= (2^6 - 1) - 101100$

$= 111111 - 101100$

$= 010011$

7. **Find 1's complement of $(0.0110)_2$**

⇒ 1's complement of $(0.0110)_2 = (2^0 - 2^{-4}) - 0.0110$

$= (1 - 2^{-4}) - 0.0110$

$= 0.1111 - 0.0110$

$= 0.1001$

Note: $(1 - 2^{-4}) = 0.9375$ and was converted into binary in the above problem.

Subtraction using 9's Complement Method

Steps:

1. Find 9's complement of the subtrahend.
2. Then add it to the minuend.
3. If we get carry adding both numbers, add 1 to the LSB of the result. If we don't get carry, then 9's complement of the sum is the final result and it is negative.

2.4 Gray Code

Gray code is unweighted and is not an arithmetic code. In gray code, the bits are arranged in such a way that it changes by only one bit as it sequences from one number to next. One of the many applications of gray code is in rotational or shaft encoder, for converting some angular position of a shaft to a digital format.

Binary	Gray Code
000	000
001	001
010	011
011	010
100	110

i. Binary to Gray Code Conversion

1. Keep MSB not gray code same as MSB of binary code.
2. Going from left to right, add adjacent pairs of binary code to get next gray code bit.
3. Discard any carries.

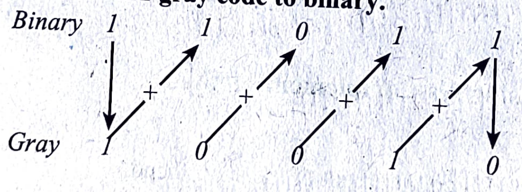
E.g., Convert 10110 to gray code

1 + 0 + 1 + 1 + 0
↓ ↓ ↓ ↓ ↓
1 1 1 0 1

ii. Gray to Binary Code Conversion

1. Keep MSB of binary code same as of gray code.
2. Add each binary code generated to the gray code bit in the next adjacent.
3. Discard carries.

E.g., Convert 110111 gray code to binary.



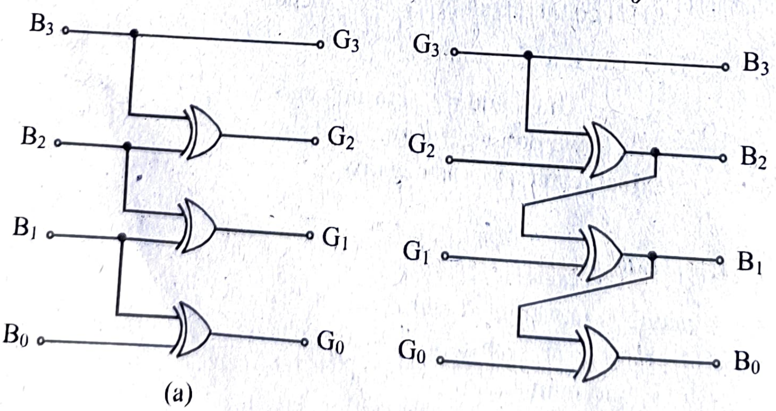


Fig.: (a) Binary to gray converter (b) Gray to binary converter

2.5 Instruction Codes

An instruction code is a group of bits that instructs the computer or processor to perform a specific function. It consists of operation codes also known as opcodes and operation part specifying the address of operand and result.

Example:

MVI A,20H

MOV C,A

ADD B.

Opcode	Operand	Remarks
MVI	A,20H	Loads the accumulator with byte 2O H
MOV	B,A	Copies content of A to B.
ADD	B	Adds content of A and B and stores result in the accumulator.

2.6 Alphanumeric Characters

The binary code that is used to represents the 26 alphabets (A-Z), 10 numerals (decimal digits 0-9) and other special characters (such as #, >,< etc.) is known as alphanumeric code.

If we consider only 26 upper case/lower case letters, 10 decimal digits, and other special characters; we need 6 bits. If we consider both upper case and lower case letters, 10 decimal digits, and other special characters; we need 7 bits.

ASCII Code

ASCII is the abbreviation for American Standard Code for Information Interchange. ASCII is a universally accepted alpha-numerical code used in most computers and other electronic equipment. Most computer keyboards are standardized with the ASCII code. When we enter a letter, a number or control command, the ASCII code goes into the computer. ASCII code is a 7-bit code; so, it can represent 128 characters.

$$'a' = 97_{10} = 1100001\ ,\ '1' = 0110001\ ,\ 'A' = 65_{10} = 1000001$$

EBCDIC as Alphanumeric Code

EBCDIC stands for Extended Binary Coded Decimal Interchange Code. It is an eight-bit code and primarily used in IBM made devices. The bit assignments of EBCDIC are different from the ASCII but the character symbols are the same.

2.7 Modulo 2 System and 2's Complement

1's Complement

The 1's complement of a binary number is the number that results when we complement each bit. That is, 1's complement of a binary number is found by changing all 1s to 0s and all 0s to 1s.

2's Complement

The 2's complement is the binary number that results when we add 1 to the 1's complement. That is,

2's complement = 1's complement + 1

Subtraction is done using 2's complement method widely. This leads to the simplest logic circuits for performing arithmetic.

Negative numbers are represented by 2's complement. Since negative number is the 2's complement of positive number, in order to represent –7,

$+7 \rightarrow 00000111$

$-7 \rightarrow 11111000 + 1 = 11111001$

EXAMPLE:

1. Find the 2's complement of 11010011

⇒

1 1 0 1 0 0 1 1	Binary number
0 0 1 0 1 1 0 0	1's complement
+1	
0 0 1 0 1 1 0 1	2's complement

2. Express – 39 as an 8–bit sign number in sign magnitude form, 1's complement, and 2's complement form.

⇒ In binary, +39 = 00100111

i) Sign magnitude form of –39 = 10100111

ii) 1's complement of –39 = 11011000

iii) 2's complement of –39 = 11011001

Subtraction using 2's Complement Method

Steps:

1. Find the 2's complement of the subtrahend.
2. Add the complement number with the minuend.
3. If we get the carry by adding both the numbers, then we discard this carry and the result is positive. If we don't get carry, take 2's complement of the result which will be negative.

i. Larger positive and smaller negative: Add + 125 and –68

+ 125	→	0 1 1 1 1 1 0 1	⟶	0 1 1 1 1 1 0 1
– 68	→	– 0 1 0 0 0 1 0 0	$\xrightarrow{\text{2's complement}}$	+ 1 0 1 1 1 1 0 0
57				(1) 0 0 1 1 1 0 0 1

For 8-bit arithmetic, discard the end carry, the result is a positive number.

Result = 00111001= +57

ii. Smaller positive and larger negative: Add +37 and –115

+37	→	0 0 1 0 0 1 0 1	
+(–115)	→	1 0 0 0 1 1 0 1	**(2's complement)**
–(78)		0 1 0 1 1 0 0 1 0	i.e., sum is negative

So, the result is in 2's complement form.

Take the 2's complement of it and put a –ve sign infront of it.

$10110010 \xrightarrow{\text{2's complement}} 01001101 + 1$

$= 01001110$

$= -78$

iii. Both negative: Add – 43 and –78

(–43)	→	1 1 0 1 0 1 0 1
+(–78)	→	1 0 1 1 0 0 1 0
–(121)		1 1 0 0 0 0 1 1 1

Sum is negative, so discard end carry. Take the 2's complement of remaining values and put minus sign in front to get the final result.

$10000111 \xrightarrow{\text{2's complement}} 01111000 + 1$

$= 01111001$

$= -121$

2.8 Binary-Coded Decimal (BCD)

Binary codes for decimal digits require a minimum of 4 bits. It is a straight assignment of binary equivalent to each digit. For example, 95 when converted into binary is equal to 1011111. But, when same number is represented in BCD code, each decimal digit is represented by 4-bits as: 1001 for 9 and 0101 for 5. Thus, the BCD equivalent will be 1001 0101.

There are only ten code groups in the BCD system i.e., 000, 0001, 0010, 0011, 0100, 0101, 0110, 0110, 0111, 1000, 1001.

Decimal	BCD
0	0000
1	0001
9	1001
10	0001 0000
10	0001 0000
11	0001 0001
99	10011001
100	0001 0000 0000

Merits:

a. It is easy to encode and decode decimals into BCD and vice versa.

b. It is also simple to implement a hardware algorithm for the BCD converter.

c. It is very useful in digital systems whenever decimal information is given either as inputs or displayed as outputs.

Demerits:

a. For integers greater than 9, BCD occupies more space than the corresponding binary. For example,

Decimal → 13

Binary → 1101

BCD code → 0001 0011

b. The arithmetic operations using BCD code require a complex design of arithmetic and logic unit (ALU) than the straight binary number system.

Excess-3 Code

It is an important 4-bit code sometimes used with binary-coded decimal (BCD) numbers. To convert any decimal number into its excess-3 form, we have to add 3 to each decimal digit, and then convert the sum to a BCD number.

Decimal to Excess-3 (Xs-3 or X-3)

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

Self complementing

Decimal to 2421 Code

Decimal	2 4 2 1 code
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	1 0 1 1
6	1 1 0 0
7	1 1 0 1
8	1 1 1 0
9	1 1 1 1

Decimal to 84–2–1 Code

Decimal	84 – 2 – 1 code
0	0 0 0 0
1	0 1 1 1
2	0 1 1 0
3	0 1 0 1
4	0 1 0 0
5	1 0 1 1
6	1 0 1 0
7	1 0 0 1
8	1 0 0 0
9	1 1 1 1

Note: Excess–3 code, 2421 code, 84–2–1 code are self complementing codes.

	BCD	Biquinary			Excess-3 code
Decimal	8 4 2 1	84–2–1	2 4 2 1	5 0 4 3 2 1 0	
0	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 0 0 0 1	0 0 1 1
1	0 0 0 1	0 1 1 1	0 0 0 1	0 1 0 0 0 1 0	0 1 0 0
2	0 0 1 0	0 1 1 0	0 0 1 0	0 1 0 0 1 0 0	0 1 0 1
3	0 0 1 1	0 1 0 1	0 0 1 1	0 1 0 1 0 0 0	0 1 1 0
4	0 1 0 0	0 1 0 0	0 1 0 0	0 1 1 0 0 0 0	0 1 1 1
5	0 1 0 1	1 0 1 1	1 0 1 1	1 0 0 0 0 0 1	1 0 0 0
6	0 1 1 0	1 0 1 0	1 1 0 0	1 0 0 0 0 1 0	1 0 0 1
7	0 1 1 1	1 0 0 1	1 1 0 1	1 0 0 0 1 0 0	1 0 1 0
8	1 0 0 0	1 0 0 0	1 1 1 0	1 0 0 1 0 0 0	1 0 1 1
9	1 0 0 1	1 1 1 1	1 1 1 1	1 0 1 0 0 0 0	1 1 0 0

EXAMPLE: Convert $(29)_{10}$ to an excess-3 code.

$$\Rightarrow \quad \begin{array}{r} 2 \\ +3 \\ \hline 5 \\ \downarrow \\ 0101 \end{array} \qquad \begin{array}{r} 9 \\ +3 \\ \hline 12 \\ \downarrow \\ 1100 \end{array}$$

After adding 9 and 3, do not carry the 1 into the next column; instead leave the result intact as 12, and then convert. Therefore, 01011100 in excess-3 code stands for decimal 29.

2.9 Parity Method for Error Detection

Parity

A parity bit is an extra bit included with a message to make the total number of 1's either odd or even.

Types of Parity

1. **Even parity:** In even parity, the number of 1's in the total information including parity bit is even.

   E.g., Message: 1101

   Even parity bit: 1

   Transmitted bit: 11011

   The system has a receiver to receive even message. If the transmitted bits and received bits are both even, then the message is accepted else rejected.

2. **Odd parity:** In odd parity, the number of 1's in the total information including parity bit is odd.

E.g., Message: 1101

Odd parity: 0

Transmitted bit: 11010

0 0 0 0 0 0 0 1

128 64 32 16 8 4 2 1

The system has a receiver to receive odd message. If the transmitted bits and received bits are both odd, then the message is accepted else rejected.

Message		Odd parity bit	Even parity bit
A	B		
0	0	1	0
0	1	0	1
1	0	0	1
1	1	1	0

SOLUTION TO IMPORTANT AND EXAM QUESTIONS.

1. **Using (r – 1)'s complement perform subtraction: $(1010100)_2 - (1000100)_2$** [2021 Fall]

(can't we do like this.

A B ⇒ 2's complement - 0111011

Now add with A

1010100
+0111011

10001111

Solution:

Let $M = (1010100)_2$

$N = (1000100)_2$

approximate method.

Taking 1's complement {(r-1)'s complement} of N, we get,

$r^n - r^{-m} - N$; where $r = 2$, $n = 7$; $m = 0$

$N = (1000100)_2$

$= 2^7 - 2^{-0} - (1000100)_2$

$= 128 - 1 - (1000100)_2$

$= (1111111)_2 - (1000100)_2$

$= (0111011)_2$

Add this one at down.

0001111
+1

$(10000)_2$ //

Now, adding this result to M, we get

$$\begin{array}{r} 1010100 \\ +0111011 \\ \hline \boxed{1}0001111 \end{array}$$

End carry

1001.001
Here m=3

Since, there is end carry, we add 1 to LSB of result to get actual result.

$$\begin{array}{r} 0001111 \\ +1 \\ \hline 0010000 \end{array} = (10000)_2$$

2. What is weighted and non-weighted code? Why is gray code called reflected code? [Fall 2020]

Solution:

Weighted codes: The number system which is positionally weighted or the digit position of any number represents a specific weight is known as weighted codes. For example, decimal, binary, octal, hexadecimal, etc.

Non-weighted codes: The number system which is not positionally weighted or the digit position of any numbers does not represent the specific weight is known as non-weighted codes. For example, excess-3 code, gray code etc.

Gray code: Gray code is the kind of code in which the bits are arranged in such a way that it changes by only one bit as it sequences from one number to another. The normal sequence of binary numbers generated by the hardware may cause error or ambiguity during the transition from one number to the next. But, the gray code eliminates this problem since only one bit changes its value during any transition between two numbers.

Gray code is also known as reflected binary code or reflected code because the first (n/2) values compare with those of the last (n/2) values, but in reverse order

For example,

For n = 4 (i.e. 4 values)

Decimal	Gray code	
0	0 0	Mirror last 1 bit
1	0 1	
2	1 1	
3	1 0	

For n = 8 (i.e. 8 values)

Decimal	Gray code
0	0 0 0
1	0 0 1
2	0 1 1
3	0 1 0
4	1 1 0
5	1 1 1
6	1 0 1
7	1 0 0

Mirror last 2 bits

Since the bit values are reflected (as in mirror), gray code is also known as reflected code.

3. Perform the following subtraction using (r – 1)'s complement.

i. $(111001)_2 - (11011)_2$

ii. $(2321)_{10} - (8301)_{10}$ **[Fall 2020]**

Solution:

i. $(111001)_2 - (11011)_2$

We have,

(r–1)'s complement of $(011011)_2$ is

$$= r^n - r^{-m} - N$$

where, r = radix = 2

N = number = (11011)

n = integer of n digit = 6

m = fractional part of m digit = 0

$\therefore$ (r–1)'s complement of (11011) is

$$= 2^6 - 2^0 - 11011$$

$$= (2^6 - 1) - (11011)$$

$$= (111111)_2 - (011011)_2$$

$$= (100100)_2$$

Now,

$$\begin{array}{r} 111001 \\ +\,100100 \\ \hline \text{Carry} \rightarrow \boxed{1}011101 \end{array}$$

Now, adding carry (1) to LSB of result, actual result

$$\begin{array}{r} 011101 \\ +1 \\ \hline (011110)_2 \end{array}$$

ii. $(2321)_{10} - (8301)_{10}$

We have,

For $(8301)_{10}$,

r = radix = 10

N = number = $(8301)_{10}$

n = integer of n digit = 4

m = fractional part of m digit = 0

$\therefore$ $(r-1)$'s complement of $(8301)_{10}$ is,

$= r^n - r^{-m} - N$

$= 10^4 - 10^{-0} - 8301$

$= (10000 - 1) - 8301$

$= 9999 - 8301 = (1698)_{10}$

Now, adding we get,

$$\begin{array}{r} 2321 \\ +1698 \\ \hline \underline{0}4019 \end{array}$$

No carry → 0

Since, there is no end carry, the actual result is $(r-1)$'s complement of the result with negative sign.

$\therefore$ The result is, $r^n - r^{-m} - N$

$= -(10^4 - r^0 - 4019)$

$= -(1000 - 1 - 4019)$

$= -(9999 - 4019)$

$= -(5980)_{10}$

4. Perform the conversion as indicated.

i. $(543)_6 = (?)_{\text{Excess-3}}$

ii. $(708)_{10} = (?)_{2421}$

iii. $(BBA)_{16} = (?)_2$ [Spring 2019]

Solution:

i. **$(543)_6 = (?)_{\text{Excess-3}}$**

First, we convert given number to decimal.

$\therefore$ $543 = 5 \times 6^2 + 4 \times 6^1 + 3 \times 6^0$

$= 5 \times 36 + 4 \times 6 + 3 \times 1$

$= 180 + 24 + 3$

$= (207)_{10}$

Now, converting into excess-3 code,

$$\begin{array}{ccc} 2 & 0 & 7 \\ +3 & +3 & +3 \\ \hline 5 & 3 & 10 \\ \downarrow & \downarrow & \downarrow \\ 0101 & 0011 & 1010 \end{array}$$

$\therefore$ $(543)_6 = (010100111010)$ excess-3

ii. **$(708)_{10} = (?)_{2421}$**

First, we convert decimal to binary,

2	708	→ 0
2	354	→ 0
2	177	→ 1
2	88	→ 0
2	44	→ 0
2	22	→ 0
2	11	→ 1
2	5	→ 1
2	2	→ 0
	1	

$\therefore$ $(708)_{10} = (1011000100)_2$

Now, grouping binary numbers to find equivalent 2421 codes

$$\begin{array}{ccc} 0010 & 1100 & 0100 \\ \downarrow & \downarrow & \downarrow \\ 2 & 6 & 4 \end{array}$$ 2421 equivalent

$\therefore$ $(708)_{10} = (264)_{2421}$

iii. **$(BBA)_{16} = (?)_2$**

$$\begin{array}{ccc} B & B & A \\ \downarrow & \downarrow & \downarrow \\ 1011 & 1011 & 1010 \end{array}$$

$\therefore$ $(BBA)_{16} = (1011\ 1011\ 1010)_2$

A = 10
B = 11
C = 12
D = 13
E = 14
F = 15
→ In decimal system

5. Use 2's complement to subtract the following

i. $(1011)_2 - (10100)_2$

ii. $(952)10 - (873)_{10}$

iii. $(368)_{BCD} - (256)_{BCD}$ [Spring 2019]

Solution:

i. $(1011)_2 - (10100)_2$

Let, $M = (01011)_2$

$N = (10100)_2$

Now, taking 2's complement of N, we get

$(10100)_2$

2's complement

$$\begin{array}{r} 01011 \\ +1 \\ \hline (01100)_2 \end{array}$$

Now, adding this result with M,

$$\begin{array}{r} 01011 \\ +01100 \\ \hline 10111 \end{array}$$

Since, there is no end carry, the answer is 2's complement of result with negative sign.

$\therefore$ Actual result $= -(\text{2's complement of } (10111)_2)$

$= -(01000 + 1)$

$= -(01001)_2$

ii. $(952)_{10} - (873)_{10}$

Binary equivalent of $(952)_{10} = (1110111000)_2 = M$

Binary equivalent of $(873)_{10} = (1101101001)_2 = N$

Now, taking 2's complement of N,

$$\begin{array}{r} 0010010110 \\ +1 \\ \hline (0010010111)_2 \end{array}$$

Adding this to M, we get

$$\begin{array}{r} 0010010111 \\ +1110111000 \\ \hline \text{End carry} \rightarrow 1\,0001001111 \end{array}$$

Since there is carry, the answer is positive number and we ignore carry.

Thus, actual result $= (0001001111)_2$

iii. $(368)_{BCD} - (256)_{BCD}$

Let, $M = (368)_{BCD} = (101110000)_2$

$N = (256)_{BCD} = (100000000)_2$

Taking 2's complement of N, we get,

$(011111111 + 1)_2 = (100000000)_2$

Adding it to M, we get,

$$\begin{array}{r} 101110000 \\ +100000000 \\ \hline \text{Carry} \rightarrow 1\,001110000 \end{array}$$

Since, there is carry, the answer is positive and we ignore the carry.

So, the result is

$= (1110000)_2$

$= (112)_{10}$

$= (000100010010)_{BCD}$

$= (112)_{BCD}$

6. **Perform the conversion as indicated**

i. $(123)_4 = (?)_{BCD}$

ii. $(4FC)_{16} = (?)_8$

iii. $(1011011)_{gray} = (?)_2$

iv. $(45)_{10} - (99)_{10}$, using r's complement. [Fall 2019]

Solution:

i. $(123)_4 = (?)_{BCD}$

Firstly, converting $(123)_4$ to decimal $(123)_4$

$= 1 \times 4^2 + 2 \times 4^1 + 3 \times 4^0$

$= 16 + 8 + 3$

$= (27)_{10}$

Now, converting decimal to BCD,

2	7
↓	↓
0010	0111

$\therefore\ (123)_4 = (00100111)_{BCD}$

ii. $(4FC)_{16} = (?)_8$

Firstly converting $(4FC)_{16}$ to binary,

4	F	C
↓	↓	↓
0100	1111	1100

$\therefore\ (4FC)_{16} = (010011111100)_2$

Now, converting to octal,

010	011	111	100
↓	↓	↓	↓
2	3	7	4

$\therefore\ (4FC)_{16} = (2374)_8$

iii. $(1011011)_{gray} = (?)_2$

Gray → 1 0 1 1 0 1 1

Binary → 1 1 0 1 1 0 1

$\therefore\ (1011011)_{gray} = (1101101)_2$

iv. $(45)_{10} - (99)_{10}$**, using r's complement.**

Let $M = (45)_{10}$

$N = (99)_{10}$

Now,

r's complement of N is

$= r^n - N$ where, $r = 10$ (radix); $n = 2$, $N = 99$

$= 10^2 - 99$

$= 100 - 99 = (1)_{10}$

Now, adding the result to M, we get,

$$\begin{array}{r} 45 \\ +\ 1 \\ \hline 46 \end{array}$$

Since, there is no end carry, answer is 10's complement of result with negative sign.

$\therefore$ Actual result $= -(\text{r's complement of } (46)_{10})$

$= -(10^2 - 46)$

$= -(100 - 46)$

$= -(54)_{10}$

7. **Perform the conversion as indicated.**

i. $(235)_6 = (?)_{excess\text{-}3}$

ii. $(369)_{10} = (?)_{2421}$

iii. $(BCA)_{16} = (?)_2$ [Spring 2018]

Solution:

Firstly converting $(235)_6$ to decimal, $(235)_6$

$= 2\times6^2 + 3\times6^1 + 5\times6^0$

$= 2\times36 + 3\times6 + 5\times1$

$= 72 + 18 + 5$

$= (95)_{10}$

Now, converting into excess-3,

9	5
+3	+3
12	8
↓	↓
1100	1000

$\therefore\ (235)_6 = (11001000)_{excess\text{-}3}$

ii. $(369)_{10} = (?)_{2421}$

Decimal		**2421**
3	→	0011
6	→	1100
9	→	1111

$\therefore\ (369)_{10} = (001111001111)_{2421}$

Decimal		2421
0	→	0000
1	→	0001
2	→	0010
3	→	0011
4	→	0100
5	→	1011
6	→	1100
7	→	1101
8	→	1110
9	→	1111

iii. $(BCA)_{16} = (?)_2$

B	C	A
↓	↓	↓
1011	1100	1010

$\therefore$ $(BCA)_{16} = (101111001010)_2$

8. Use 2's complement to subtract the following:

i. $(1010)_2 - (10100)_2$

ii. $(957)_{10} - (876)_{10}$

iii. $(378)_{BCD} - (256)_{BCD}$ [Spring 2018]

Solution:

i. **$(1010)_2 - (10100)_2$**

Let $M = (1010)_2 = (01010)_2$

$N = (10100)_2$

Now, taking 2's complement of $(10100)_2$, we get

$$\begin{array}{r} 0\,1\,0\,1\,1 \\ +1 \\ \hline (0\,1\,1\,0\,0)_2 \end{array}$$

Adding this result to M,

$$\begin{array}{r} 0\,1\,1\,0\,0 \\ +0\,1\,0\,1\,0 \\ \hline (1\,0\,1\,1\,0)_2 \end{array}$$

Since there is no carry, actual result is 2's complement of the result with –ve sign.

$\therefore$ Actual result $= -(\text{2's complement of } (10110)_2)$

$= -(01001 + 1)$

$= -(01010)_2$

ii. **$(957)_{10} - (876)_{10}$**

Converting into binary,

2	957	→ 1
2	478	→ 0
2	239	→ 1
2	119	→ 1
2	59	→ 1
2	29	→ 1
2	14	→ 0
2	7	→ 1
2	3	→ 1
	1	

$\therefore$ $(957)_{10} = (1110111101)_2 = M$ (suppose)

2	876	→ 0
2	438	→ 0
2	219	→ 1
2	109	→ 1
2	54	→ 0
2	27	→ 1
2	13	→ 1
2	6	→ 0
2	3	→ 1
	1	

$\therefore$ $(876)_{10} = (1101101100)_2 = N$ (suppose)

Now, taking 2's complement of N, we get,

$$\begin{array}{r} 0\,0\,1\,0\,0\,1\,0\,0\,1\,1 \\ +1 \\ \hline (0\,0\,1\,0\,0\,1\,0\,1\,0\,0)_2 \end{array}$$

Adding this result to M, we get,

$$
\begin{array}{r}
1110111101 \\
+0010010100 \\
\hline
\text{Carry} \rightarrow \underline{1}0001010001
\end{array}
$$

Since, there is carry, the result is positive.

So, we ignore the carry and the result is

$= (1010001)_2$

$= (81)_{10}$

iii. **$(378)_{BCD} - (256)_{BCD}$**

Converting BCD number to binary,

Let, $M = (378)_{BCD} = (101111010)_2$

$N = (256)_{BCD} = (100000000)_2$

Taking 2's complement of N, we get,

$(011111111 + 1)_2 = (100000000)_2$

Adding it to M, we get,

$$
\begin{array}{r}
101111010 \\
+100000000 \\
\hline
\text{Carry} \rightarrow \underline{1}001111010
\end{array}
$$

Since, there is carry, we ignore carry and the result is positive.

$\therefore$ Actual result $= (1111010)_2$

$= (122)_{10}$

$= (000100100010)_{BCD}$

9. Convert the following conversions:

i. $(101001.101)_2 = (?)_{10}$

ii. $(ABD)_{16} = (?)_8$

iii. $(10101101)_2 = (?)_{gray}$

iv. $(175.351)_8 = (?)_{16}$ [Fall 2018]

Solution:

i. ***$(101001.101)_2 = (?)_{10}$***

$(101001.101)_2$

$= (1\times2^5+0\times2^4+1\times2^3+0\times2^2+0\times2^1+1\times2^0).(1\times2^{-1}+0\times2^{-2}+1\times2^{-3})$

$= (32 + 8 + 1). (0.5 + 0.125)$

$= (41.625)_{10}$

ii. **$(ABD)_{16} = (?)_8$**

Firstly converting $(ABD)_{16}$ into binary,

A	B	D
↓	↓	↓
1010	1011	1101

$= (101010111101)_2$

Now, converting binary to octal,

101	010	111	101
↓	↓	↓	↓
5	2	7	5

$= (5275)_8$

iii. **$(10101101)_2 = (?)_{gray}$**

binary→	1	+	0	+	1	+	0	+	1	+	1	+	0	+	1
	↓		↓		↓		↓		↓		↓		↓		↓
gray→	1		1		1		1		1		0		1		1

$\therefore$ $(10101101)_2 = (11111011)_{gray}$

iv. **$(175.351)_8 = (?)_{16}$**

Firstly converting octal to binary,

1	7	5	3	5	1
↓	↓	↓	↓	↓	↓
001	111	101	011	101	001

$= (001111101.011101001)_2$

Now, converting into hexadecimal,

0111	1101	0111	0100	1000
↓	↓	↓	↓	↓
7	D	7	4	8

$= (7D.748)_{16}$

10. Perform the following subtraction using 2's compliment method.

i. $(1000100)_2 - (1010100)_2$

ii. $(11010)_2 - (10000)_2$ [Spring 2017]

Solution:

i. **$(1000100)_2 - (1010100)_2$**

Let, $M = (1000100)_2$

$N = (1010100)_2$

Taking 2's compliment of N, we get

$$\begin{array}{r} 0101011 \\ +1 \\ \hline (0101100)_2 \end{array}$$

Now, adding the result with M,

$$\begin{array}{r} 0101100 \\ +1000100 \\ \hline (1110000)_2 \end{array}$$

Since there is no carry, the actual result is 2's compliment of the result with negative sign.

$\therefore$ Actual result $= -(2\text{'s compliment of } (1110000)_2)$

$= -(0001111 + 1)_2$

$= -(10000)_2$

ii. **$(11010)_2 - (10000)_2$**

Let, $M = (11010)_2$

$N = (10000)_2$

Taking 2's compliment of N, we get

$$\begin{array}{r} 01111 \\ +1 \\ \hline (10000)_2 \end{array}$$

Adding the result to M, we get

$$\begin{array}{r} 11010 \\ +10000 \\ \hline \text{Carry} \rightarrow \boxed{1}01010 \end{array}$$

Since, there is carry, the result is positive and we ignore carry.

Thus, actual result $= (1010)_2$

11. **Determine the value of base x if $(211)_x = (152)_8$** [Fall 2017]

Solution:

$(211)_x = (152)_8$

or, $2 \times x^2 + 1 \times x^1 + 1 \times x^0 = 1 \times 8^2 + 5 \times 8^1 + 2 \times 8^0$

or, $2x^2 + x + 1 = 64 + 40 + 2$

or, $2x^2 + x - 105 = 0$

or, $2x^2 - 14x + 15x - 105 = 0$

or, $2x(x - 7) + 15(x - 7) = 0$

or, $(2x + 15)(x - 7) = 0$

Either $(2x + 15) = 0$

$\Rightarrow x = -\frac{15}{2}$ (ignored)

or, $x = 7$

$\therefore$ The value of base $x = 7$

12. **Convert the following numbers from the given base to the other bases as indicated.**

i. **$(11001.11)_2 = (?)_8$**

ii. **$(5FE.DD)_{16} = (?)_{10}$**

iii. **$(777)_8 = (?)_{16}$** [Spring 2016]

Solution:

i. **$(11001.11)_2 = (?)_8$**

$$\begin{array}{ccc} 011 & 001 & 110 \\ \downarrow & \downarrow & \downarrow \\ 3 & 1 & 6 \end{array}$$

$\therefore (11001.11)_2 = (31.6)_8$

ii. **$(5FE.DD)_{16} = (?)_{10}$**

$(5FE.DD)_{16}$

$= (5 \times 16^2 + 15 \times 16^1 + 14 \times 16^0).(13 \times 16^{-1} + 13 \times 16^{-2})$

$= (1280 + 240 + 14).(0.8125 + 0.05078)$

$= (1534).(0.8633)$

$= (1534.8633)_{10}$

iii. **$(777)_8 = (?)_{16}$**

Firstly converting octal to binary,

$$\begin{array}{ccc} 7 & 7 & 7 \\ \downarrow & \downarrow & \downarrow \\ 111 & 111 & 111 \end{array}$$

$\therefore (777)_8 = (111111111)_2$

Now, converting binary to hexadecimal,

0001	1111	1111
↓	↓	↓
1	F	F

∴ $(777)_8 = (1FF)_{16}$

13. Perform the subtraction with the following binary number using 1's complement

i. $(1010100)_2 - (1000100)_2$

ii. $(1000100)_2 - (1010100)_2$ [Spring 2016]

Solution:

i. $(1010100)_2 - (1000100)_2$

Let, $M = (1010100)_2$

$N = (1000100)_2$

Taking 1's complement of N, we get

$(0111011)_2$

Now, adding it to M, we get

$$\begin{array}{r} 1010100 \\ +0111011 \\ \hline 1\,0001111 \end{array}$$

Carry → 1

Since, there is carry, the answer is positive and we add 1 to LSB of the result

∴ Result = $(0001111 + 1) = (10000)_2$

ii. $(1000100)_2 - (1010100)_2$

Let, $M = (1000100)_2$

$N = (1010100)_2$

Taking 1's complement of N, we get

$(0101011)_2$

Adding it to M, we get

$$\begin{array}{r} 1000100 \\ +0101011 \\ \hline 1101111 \end{array}$$

Since, there is no carry, the answer is 1's compliment of the result with negative sign

∴ Actual result = $-(\text{1's complement of } (1101111)_2)$

$= -(0010000)_2$

14. Perform the conversion as indicated

i. $(243)_6 = (?)_{excess\text{-}3}$

ii. $(816)_{10} = (?)_{2421}$

iii. $(BE)_{16} = (?)_2$ [Fall 2016]

Solution:

i. $(243)_6 = (?)_{excess\text{-}3}$

Firstly converting $(243)_6$ into decimal,

$2 \times 6^2 + 4 \times 6^1 + 3 \times 6^0$

$= 2 \times 36 + 4 \times 6 + 3 \times 1$

$= 72 + 24 + 3$

$= (99)_{10}$

Now, converting into Excess-3,

9	9
+3	+3
12	12
↓	↓
1100	1100

∴ $(243)_6 = (11001100)_{excess\text{-}3}$

ii. $(816)_{10} = (?)_{2421}$

Decimal		2421
8	→	(1110)
1	→	(0001)
6	→	(1100)

∴ $(816)_{10} = (1110\ 0001\ 1100)_2$

iii. $(BE)_{16} = (?)_2$

B	E
↓	↓
1011	1110

∴ $(BE)_{16} = (10111110)_2$

15. **Perform the following conversion.**

i. $(573)_{10} = (\;)_{excess\text{-}3}$

ii. $(842)_{10} = (\;)_{2421}$

iii. $(10101111)_{gray} = (\;)_2$

iv. $(FAB)_{16} = (\;)_2$ [Spring 2015]

Solution:

i. $(573)_{10} = (\;)_{excess\text{-}3}$

5	7	3
+3	+3	+3
8	10	9
↓	↓	↓
1000	1010	1001

$\therefore$ $(573)_{10} = (100010101001)_{excess\text{-}3}$

ii. $(842)_{10} = (\;)_{2421}$

Decimal		2421 equivalent
8	→	1110
4	→	0100
2	→	0010

$\therefore$ $(842)_{10} = (111001000010)_{2421}$

iii. $(10101111)_{gray} = (\;)_2$

Gray → 1 0 1 0 1 1 1 1

Binary → 1 1 0 0 1 0 1 0

$\therefore$ $(10101111)_{gray} = (11001010)_2$

iv. $(FAB)_{16} = (\;)_2$

F	A	B
↓	↓	↓
1111	1010	1011

$\therefore$ $(FAB)_{16} = (111110101011)_2$

16. **Perform the conversion as indicated.**

$(243)_6 = (\;)_{excess\text{-}3}$ [Fall 2015]

Solution:

Firstly converting $(243)_6$ to decimal,

$2 \times 6^2 + 4 \times 6^1 + 3 \times 6^0$

$= 2 \times 36 + 4 \times 6 + 3 \times 1$

$= 73 + 24 + 3$

$= (99)_{10}$

Now, converting to excess-3,

9	9
+3	+3
12	12
↓	↓
1100	1100

$\therefore$ $(243)_6 = (11001100)_{excess\text{-}3}$

17. **Use 2's complement to subtract the following**

i. $(101)_2 - (10100)_2$

ii. $(3950)_{10} - (876)_{10}$ [Fall 2015]

Solution:

Let, $M = (00101)_2$

$N = (10100)_2$

Taking 2's complement of N, we get

$$\begin{array}{r} 0\,1\,0\,1\,1 \\ +1 \\ \hline (0\,1\,1\,0\,0)_2 \end{array}$$

Adding result to M, we get

$$\begin{array}{r} 0\,0\,1\,0\,1 \\ +0\,1\,1\,0\,0 \\ \hline 1\,0\,0\,0\,1 \end{array}$$

Since, there is no end carry, answer is negative and we take 2's complement of the result.

$\therefore$ Actual result $= -(\text{2's complement of } (10001)_2)$

$= -(01110 + 1)_2$

$= -(1111)_2$

ii. $(3950)_{10} - (876)_{10}$

Let, $M = (3950)_{10} = (111101101110)_2$

$N = (876)_{10}$

$= (1101101100)_2$

$= (001101101100)_2$

Now, taking 2's complement of N, we get,

$$\begin{array}{r} 110010010011 \\ +1 \\ \hline (110010010100)_2 \end{array}$$

Adding the result to M, we get,

$$\begin{array}{r} 111101101110 \\ +110010010100 \\ \hline \underline{1}\,110000000010 \end{array}$$

Carry

Since, there is carry, the answer is positive and we ignore the carr
So, the actual result is,

$(110000000010)_2 = (3074)_{10}$

18. Find the value of x:

i. $(563)_7 = (x)_3$

ii. $(100111)_{gray} = (x)_{binary}$ [Spring 201

Solution:

i. $(563)_7 = (x)_3$

Firstly converting to decimal,

$5\times7^2 + 6\times7^1 + 3$

$= 5 \times 49 + 42 + 3$

$= (290)_{10}$

Now, converting decimal to base-3

$$\begin{array}{r|l} 3 & 290 \rightarrow 2 \\ 3 & 96 \rightarrow 0 \\ 3 & 32 \rightarrow 2 \\ 3 & 10 \rightarrow 1 \\ 3 & 3 \rightarrow 0 \\ & 1 \end{array}$$

$\therefore$ $(563)_7 = (101202)_3$

ii. $(100111)_{gray} = (x)_{binary}$

Gray → 1 0 0 1 1 1

Binary → 1 1 1 0 1 0

$\therefore$ $(100111)_{gray} = (111010)_{binary}$

19. Add and multiply the following number in given base without converting to decimal.

i. $(1230)_4$ **and** $(23)_4$

ii. $(135.4)_6$ **and** $(43.2)_6$ [Fall 2014]

Solution:

i. $(1230)_4$ **and** $(23)_4$

Addition

$$\begin{array}{r} 1 \\ (1\,2\,3\,0)_4 \\ +\,(2\,3)_4 \\ \hline (1\,3\,1\,3)_4 \end{array}$$

$4\underline{|5} \rightarrow 1$ (Carry)

1

Base 4 → 4 digits

0, 1, 2, 3

0, 1, 2, 3, 10, 11, 12, 13

Multiplication

$$\begin{array}{r} (1\,2\,3\,0)_4 \\ +\,(2\,3)_4 \\ \hline 1\,1\,0\,1\,0 \\ +\,3\,1\,2\,0\,\times \\ \hline (1\,0\,2\,2\,1\,0)_4 \end{array}$$

ii. $(135.4)_6$ and $(43.2)_6$

Addition

$$\begin{array}{r} (1\,3\,5\,.\,4)_6 \\ +\,(4\,3\,.\,2)_6 \\ \hline (2\,2\,3\,.\,0)_6 \end{array}$$

Multiplication

$$
\begin{array}{r}
(1\,3\,5\,.\,4)_6 \\
+\,(4\,3\,.\,2)_6 \\
\hline
3\,1\,5\,2 \\
4\,5\,5\,0\,\times \\
+\,1\,0\,3\,4\,4\,\times\,\times \\
\hline
1\,1\,3\,1\,4.5\,2
\end{array}
$$

$= (11314.52)_6$

20. Perform the following conversion [Fall 2014

i. $(5849)_{10} = (?)_{excess\text{-}3}$

ii. $(8412)_{10} = ()_{2421}$

iii. $(10101111)_{gray} = (?)_2$

Solution:

i. $(5849)_{10} = (?)_{excess\text{-}3}$

5	8	4	9
+3	+3	+3	+3
8	11	7	12
↓	↓	↓	↓
1000	1011	0111	1100

$\therefore$ $(5849)_{10} = (1000101101111100)_{excess\text{-}3}$

ii. $(8412)_{10} = ()_{2421}$

Decimal		2421 equivalent
8	→	1110
4	→	0100
1	→	00010
2	→	0010

$\therefore$ $(8412)_{10} = (1110010000010010)_{2421}$

iii. $(10101111)_{gray} = (?)_2$

Gray → 1 0 1 0 1 1 1 1

1 1 0 0 1 0 1 0

$\therefore$ $(10101111)_{gray} = (11001010)_2$

21. Perform the following subtraction $(34)_8 - (27)_8$. Convert the given number into binary and perform subtraction using 2's complement. [Spring 2013]

Solution:

$$
\begin{array}{r}
(3\,4)_8 \\
-\,(2\,7)_8 \\
\hline
(5)_8
\end{array}
$$

$(34)_8$

3	4
↓	↓
011	100

$= (011100)_2 = M$(suppose)

$(27)_8$

2	7
↓	↓
010	111

$= (010111)_2 = N$(suppose)

Now, taking 2's complement of N, we get

$$
\begin{array}{r}
1\,0\,1\,0\,0\,0 \\
+\,1 \\
\hline
(1\,0\,1\,0\,0\,1)_2
\end{array}
$$

Adding this result to M, we get

$$
\begin{array}{r}
0\,1\,1\,1\,0\,0 \\
+\,1\,0\,1\,0\,0\,1 \\
\hline
\boxed{1}\,0\,0\,0\,1\,0\,1
\end{array}
$$

↑
Carry

Since, there is carry, the number is positive. So, we ignore end carry and the actual result is $= (101)_2 = (5)_8$

22. Subtract $(100101)_2$ from $(1111101)_2$ using 2's complement method. [Fall 2013]

Solution:

Let, $M = (1111101)_2$ and $N = (0100101)_2$

Taking 2's complement of N, we get

$1011010 + 1 = (1011011)_2$

Adding it to M, we get

$$\begin{array}{r} 1111101 \\ +1011011 \\ \hline \underline{1}1011000 \end{array}$$

↑
Carry

Since, there is carry, the number is +ve. We ignore carry. So, actual result is $(1011000)_2$.

23. Subtract $(7729)_{10} - (842.4)_{10}$ using 9's complement. [Fall 2013]

Solution:

Let, $M = (7729)_{10}$

$N = (0842.4)_{10}$

Now, taking 9's complement of N, we get

$r^n - r^{-m} - N$

$= 10^4 - 10^{-1} - 842.4$ $(n = 4, m = 2, r = 10, N = 842.4)$

$= (9157.5)_{10}$

Now, adding it to M, we get

$$\begin{array}{r} 7729.0 \\ +9157.5 \\ \hline \underline{1}6886.5 \end{array}$$

↑
Carry

Since, there is end carry, the result is +ve and we add it to the LSB of result.

$\therefore$ Actual result $= (6886.5 + 0.1)_{10}$

$= (6886.6)_{10}$

24. Perform the conversions as indicated

i. **$(556)_4 = (?)_{Excess\text{-}3}$**

ii. **$(786)_{10} = (?)_{BCD}$**

iii. **$(437.126)_8 = (?)_2$** [2021 Fall]

Solution:

i. Firstly converting $(556)_4$ into decimal, we get,

$5 \times 4^2 + 5 \times 4^1 + 6 \times 4^0$

$= 5 \times 16 + 5 \times 4 + 6 \times 1$

$= 80 + 20 + 6$

$= (106)_{10}$

Now, converting into excess-3 code,

$$\begin{array}{ccc} 1 & 0 & 6 \\ +3 & +3 & +3 \\ \hline 4 & 3 & 9 \\ \downarrow & \downarrow & \downarrow \\ 0100 & 0011 & 1001 \end{array} \leftarrow \text{Equivalent BCD numbers}$$

$\therefore$ $(556)_4 = (010000111001)_{BCD}$

ii.

$$\begin{array}{ccc} 7 & 8 & 6 \\ \downarrow & \downarrow & \downarrow \\ 0111 & 1000 & 0110 \end{array} \leftarrow \text{Equivalent BCD numbers}$$

$\therefore$ $(786)_{10} = (011110000110)_{BCD}$

iii.

$$\begin{array}{cccccc} 4 & 3 & 7 . & 1 & 2 & 6 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 100 & 011 & 111 & 001 & 010 & 110 \end{array} \leftarrow \text{Equivalent binary}$$

$\therefore$ $(437.126) = (100011111.001010110)_2$