P.R. GOVERNMENT COLLEGE (A), KAKINADA

B.Sc. II Year - Electronics – Semester – 3

PAPER – 3 [Code: EL3202]

w.e.f. 2019 - 20 ADMITTED BATCH

DIGITAL ELECTRONICS

4 Hours/Week [Total: 60 hrs] **Credits: 3**

Teaching hours: 12 hrs

Unit – I

Number system and codes:

Decimal, Binary, Hexadecimal, Octal, BCD, Conversions – Binary to Decimal vice versa – Binary to Hexa decimal vice versa, Decimal to Hexa decimal vice versa, Complements (1's and 2's,), Addition, Subtraction. Gray code & Excess-3 Code conversion of - BCD to Gray vice versa – BCD to Excess 3 Code vice versa.

Index

Introduction to digital system

A Digital system is an interconnection of digital modules and it is a system that manipulates discrete elements of information that is represented internally in the binary form.

Now a day's digital systems are used in wide variety of industrial and consumer products such as automated industrial machinery, pocket calculators, microprocessors, digital computers, digital watches, TV games and signal processing and so on.

In digital systems, the signals have two discrete values and are therefore said to be binary.

Digital Electronics represents information (0, 1) with only two discrete values.

- Ideally
  - "no voltage" (e.g., 0v) represents a 0 and
  - "full source voltage" (e.g., 5v) represents a 1
- Realistically
  - "low voltage" (e.g., < 1.5 v) represents a 0 and
  - "high voltage" (e.g., > 3.5 v) represents a 1

We achieve these discrete values by using switches.

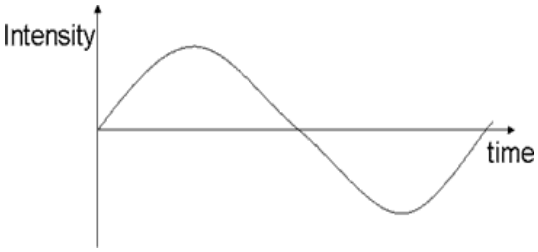
We use transistor switches, which operates at high speed, electronically, a small in size.

Analog System Vs Digital System

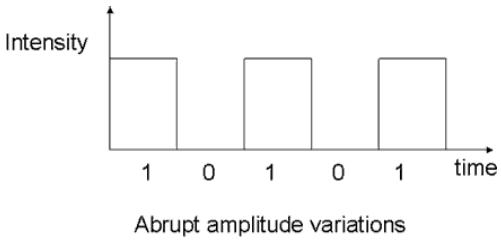
Analog systems process time-varying signals that can take on any value across a continuous range of voltages (in electrical/electronic systems).

Digital systems use digital circuits that can process digital signals which can take on only one of two discrete values of voltages (in electrical/electronic systems).

Discrete values are called 1 and 0 (ON and OFF, HIGH and LOW, TRUE and FALSE, etc.)



Analog wave form



Digital wave form

Introduction to Number system

To define any number system, we have to specify

- Base of the number system such as 2,8,10 or 16.
- The base decides the total number of digits available in that number system.
- First digit in the number system is always zero and last digit in the number system is always base-1.

Number systems

There are 4 number systems

1) Decimal number system
2) Binary number system

3) Octal number system
4) Hexa decimal number system

Decimal number system :- In this system ten digits (symbols) are used to represent a number. The digits are 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. So, the base of this system is 10.

Ex: The number 265 in decimal system should be written as $(265)_{10}$

In this system each digit in the number has a place value

The place value of the right digit is 1 unit i.e. (10^0)

The place value of the next left digit is 10 units i.e. (10^1)

The place value of the next left digit is 100 units i.e. (10^2) ……. so on.

So, the value of a number in this system is the sum of the products of the digits with its respective powers of 10.

Ex 1:- The value of $(2547)_{10}$ is $2 \times 10^3 + 5 \times 10^2 + 4 \times 10^1 + 7 \times 10^0$

The left digit or bit is called most significant bit (MSB) and the right digit or bit is called least significant bit (LSB) in a given number.

Ex 2:- The value of $(125.36)_{10}$ is $1 \times 10^2 + 2 \times 10^1 + 5 \times 10^0 + 3 \times 10^{-1} + 6 \times 10^{-2}$

This means that

$$\begin{array}{r} 100 \\ +\ 20 \\ +\ 5 \\ +\ 0.3 \\ +\ 0.06 \\ \hline 125.36 \end{array}$$

Binary number system :- In this system only two digits (symbols) are used to represent a number. The digits are 0 and 1. So, the base of this system is 2.

This system is useful to utilize in computers, machines and in electronic circuits because they can recognize only two states i.e. ON and OFF states. On means 1 and Off means 0.

The number 101 in binary system should be written as $(101)_2$

In this system also each digit in the number has a place value.

The place value of the right digit is 1 unit i.e. (2^0)

The place value of the next left digit is 2 units i.e. (2^1)

The place value of the next left digit is 4 units i.e. (2^2) ……. so on.

The following table is useful for conversion between decimal and binary systems.

Decimal	Binary		Decimal	Binary
0	0		11	1011
1	1		12	1100
2	10		13	1101
3	11		14	1110
4	100		15	1111
5	101		16	10000
6	110		17	10001
7	111		18	10010
8	1000		19	10011

9	1001		20	10100
10	1010		21	10101

The value of a number in this system is the sum of the products of the digits with its respective powers of 2.

Ex 1:- The value of $(1101)_2$ is $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

In this system also the left digit or bit is called most significant bit (MSB) and the right digit or bit is called least significant bit (LSB).

Ex 2:- The value of $(101.01)_2$ is $1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$

Octal number system :- In this system eight digits are used to represent a number. The digits are 0, 1, 2, 3, 4, 5, 6 and 7. So, the base of this system is 8.

Ex: The number 237 in octal system should be written as $(237)_8$

In this system each digit in the number has a place value

The place value of the right digit is 1 unit i.e. (8^0)

The place value of the next left digit is 8 units i.e. (8^1)

The place value of the next left digit is 64 units i.e. (8^2) ……. so on.

So, the value of a number in this system is the sum of the products of the digits with its respective powers of 8.

Ex 1:- The value of $(237)_8 = 2 \times 8^2 + 3 \times 8^1 + 7 \times 8^0 = (159)_{10}$

In this system also the left digit or bit is called most significant bit (MSB) and the right digit or bit is called least significant bit (LSB).

Hexa Decimal number system :- In this system sixteen digits (symbols) are used to represent a number. The digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 followed by the letters A, B, C, D, E and F, which represent the values 10, 11,12,13,14 and 15 respectively. So, the base of this system is 16.

Ex: The number A6E in decimal system should be written as $(A6E)_{16}$

In this system each digit in the number has a place value

The place value of the right digit is 1 unit i.e. (16^0)

The place value of the next left digit is 16 units i.e. (16^1)

The place value of the next left digit is 256 units i.e. (16^2) ……. so on.

So, the value of a number in this system is the sum of the products of the digits with its respective powers of 16.

Ex 1:- The value of $(A5E)_{16} = 10 \times 16^2 + 5 \times 16^1 + 14 \times 16^0 = (2654)_{10}$

In this system also the left digit or bit is called most significant bit (MSB) and the right digit or bit is called least significant bit (LSB).

Decimal	Binary	Octal	Hexadecimal
0	**0000**	**0**	**0**
1	**0001**	**1**	**1**
2	**0010**	**2**	**2**
3	**0011**	**3**	**3**
4	**0100**	**4**	**4**
5	**0101**	**5**	**5**
6	**0110**	**6**	**6**

7	0111	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

Conversion systems

Binary to Decimal conversion

Conversion of integer binary number into decimal number is easy.

- Take each digit in the binary number and multiply it with its place value.
- The sum of all the above products will give the equivalent decimal number.

Ex :- $(1011)_2$ to decimal number.

Sol :- $(1011)_2 = (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

$= (1 \times 8) + (0 \times 4) + (1 \times 2) + (1 \times 1)$

$= (11)_{10}$

Fractional Binary number to Decimal number

For the fraction (after the decimal point) the place value starts with negative power of 2. This negative power value increases from left to right.

The place value of the first left digit in fraction is (2^{-1})

The place value of the next right digit in fraction is (2^{-2})

The place value of the next right digit in fraction is (2^{-3}) ……. so on

Ex :- Convert the binary number $(0.1101)_2$ into decimal number.

$(0.1101)_2 = (1 \times \frac{1}{2}) + (1 \times \frac{1}{4}) + (0 \times \frac{1}{8}) + (1 \times \frac{1}{16})$

$= 1 \times 0.5 + 1 \times 0.25 + 0 \times 0.125 + 1 \times 0.0625$

$= 0.5 + 0.25 + 0 + 0.0625$

$= (0.8125)_{10}$

Ex :- Convert the binary number $(1010.0101)_2$ into decimal number.

Sol :- Integer part $(1010)_2 = (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$

$= (1 \times 8) + (0 \times 4) + (1 \times 2) + (0 \times 1)$

$= (10)_{10}$

Fractional part $(0.0101)_2 = (0 \times 2^{-1}) + (1 \times 2^{-2}) + (0 \times 2^{-3}) + (1 \times 2^{-4})$

$= (0 \times \frac{1}{2}) + (1 \times \frac{1}{4}) + (0 \times \frac{1}{8}) + (1 \times \frac{1}{16})$

$= 0 \times 0.5 + 1 \times 0.25 + 0 \times 0.125 + 1 \times 0.0625$

$= 0 + 0.25 + 0 + 0.0625$

$= (0.3125)_{10}$

Binary to Octal and Hexadecimal system

i) Binary to octal number conversion
ii) Binary to hexa decimal number conversion

The binary number:	001	010	011	000	100	101	110	111
The octal number:	1	2	3	0	4	5	6	7

The binary number:	0001	0010	0100	1000	1001	1010	1101	1111
The hexadecimal number:	1	2	5	8	9	A	D	F

Decimal to Binary conversion

The decimal number is converted in to binary number by using successive divisions method.

1. The decimal number (i.e. dividend) is divided by 2 (i.e. divisor).
2. If '1' or "0" occurs as remainder, transfer that '1' or "0" to the binary record.
3. Now take quotient as dividend and divide it by 2 and transfer the remainder to the binary record.
4. The same procedure is continued until the quotient becomes zero
5. The last remainder is taken as most significant bit (MSB).
6. The first remainder is taken as least significant bit (LSB).
7. The equivalent binary number comprises with all the remainders in successive divisions method in the order from MSB (bottom) to LSB (top).

Ex :- Convert the decimal number $(196)_{10}$ in to binary number.

Successive divisions		Remainders	
2	196		
2	98	0	LSB
2	49	0	
2	24	1	
2	12	0	
2	6	0	
2	3	0	
2	1	1	
	0	1	MSB

$\therefore (196)_{10} = (11000100)_2$

Order - From bottom(MSB) to top (LSB)

Fractional Decimal number to binary

The fractional decimal number is converted in to binary number by using successive fraction multiplications method or double-dabble method.

1. The fractional decimal number is multiplied with 2 by successive fraction multiplications method.
2. If '1' or '0' occurs in units place in the product, transfer that '1' or '0' to the binary record.
3. The multiplication is continued with the remaining fraction.
4. The same procedure is followed in each multiplication.
5. The first transferred number (1 or 0) to binary record is taken as most significant bit (MSB).
6. The last transferred number (1 or 0) to binary record is taken as least significant bit

(LSB).

7. If the multiplication does not end, it can be stopped at any of our desired level.

Ex : - Convert the fractional decimal number $(0.638)_{10}$ in to fractional binary number.

Sol :-

Successive multiplications		Binary	
0·638 × 2 = 1·276	⟶	1	MSB
0·276 × 2 = 0·552	⟶	0	
0·552 × 2 = 1·104	⟶	1	
0·104 × 2 = 0·208	⟶	0	
0·208 × 2 = 0·416	⟶	0	
0·416 × 2 = 0·832	⟶	0	
0·832 × 2 = 1·664	⟶	1	
0·664 × 2 = 1·328	⟶	1	
0·328 × 2 = 0·656	⟶	0	
0·656 × 2 = 1·312	⟶	1	LSB

So, $(0.638)_{10} = (0.1010001101)_2$

Decimal to Hexadecimal Conversion

Ex: Convert $(415)_{10}$ in to hexadecimal system

```
16 | 415
16 |  25    F  (15 decimal)
16 |   1    9
       0    1
```

Read bottom to top

Result: 415 in decimal is 19F in hexadecimal

Hexadecimal to Decimal Conversion

Ex: $5C7_{16}$ to decimal

Sol:- $(5C7)_{16} = (5x16^2) + (C\ x16^1) + (7\ x16^0)$

$= 1280 + 192 + 7$

$= (1479)_{10}$

Binary Addition

In decimal addition, if you add 8 + 2 you get ten, which you write as 10; in the sum this gives a digit 0 and a carry of 1. Something similar happens in binary addition when you add 1 and 1; the result is two (as always), but since two is written as 10 in binary, we get, after summing 1 + 1 in binary, a digit 0 and a carry of 1.

The rules of binary addition are

1) 0 + 0 = 0
2) 1 + 0 = 1
3) 0 + 1 = 1
4) 1 + 1 = 10 (it is two in binary system)

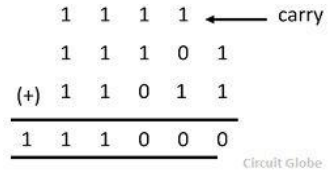
The 4th rule can be written as 1 + 1 = 0 with a carry 1

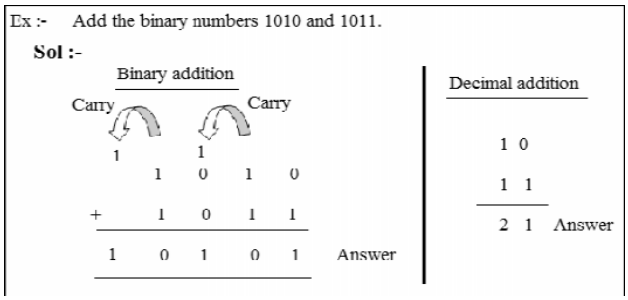
The above rules can be written as the following truth table.

Truth table

Input		Out put	
A	B	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

The above first three equations are very identical to the binary digit number. The column by column addition of binary is applied below in details. Let us consider the addition of 11101 and 11011.





Compliment of binary number

The usual practice in computers is to change the subtraction in to an addition process. So, the circuits are simplified such that only adder circuits are used for both addition and subtraction. Compliments are useful for this purpose.

Two types of compliments are there in binary numbers.

1's compliment :- The 1's compliment of a binary number is the number obtained after changing each 0 into 1 and each 1 into 0 in that binary number.

Few examples for 1'compliment to the binary numbers.

Binary	1's compliment
1100	0011
1010	0101
1110	0001
1001	0110

2's compliment :- The 2's compliment of a binary number is the number obtained by adding 1 to its 1's compliment.

For doing subtraction by using 2's compliment method, the following rules are to be followed.

1) First find the 2's compliment to the subtrahend (number to be subtracted).
2) Add this 2's compliment to the minuend (number from which subtraction is to be done).
3) Drop the carry 1 in the last (Left) position (called end around carry or EAC) in the sum.
4) This is the answer an is positive.
5) If there is no carry or EAC in the sum, then find the 2's compliment to the sum and put a (-) sign before it to get the answer.

Ex. 1 :- Subtract $(10011)_2$ from $(11010)_2$ using 2's compliment method.

Sol :- 2's compliment method

1's compliment of 10011 is 01100
+ 1
2's compliment of 10011 is 01101

Adding the given minuend (11010) and 2's compliment of subtrahend (01101)

11010
+ 01101
Carry [1] [1]00111

Since, the carry is 1, it is removed and the remaining part is the answer
So, the answer is 00111

Decimal subtraction

26
- 19
7 Answer

Ex.2 :- Subtract $(11011)_2$ from $(10010)_2$ using 2's compliment method.

Sol :- 2's compliment method

1's compliment of 11011 is 00100
+ 1
2's compliment of 11011 is 00101

Adding the given minuend (10010) and 2's compliment of subtrahend (00101)

10010
+ 00101
No carry in the sum 10111

Since there is no carry in the sum, to get the answer we have to take the 2's compliment with (-) sign.

1's compliment of the sum is 01000
+ 1
2's compliment of the sum is 01001

So, the answer is - 01001

Decimal subtraction

18
-27
-9 Answer

What is meant by code?

The digital data is represented, stored and transmitted as group of binary bits. This group is also called as binary code. The binary code is represented by the number as well as alphanumeric letter.

Codes are classified in to 2 types

- Weighted
- Non-weighted

Weighted code:

In weighted code, each digit position has a weight or value. The sum of all digits multiplied by a weight gives a total amount being represented. BCD or 8421 is a type of weighted code where each digit position is assigned a specific weight.

Non weighted code:

In non-weighted code there is no positional weight . i.e., Each position within the binary number is not assigned a prefix value. No specific positions are assigned to bit positions in non-weighted code. The non-weighted codes are (1) Gray code (2) Excess-3 code.

Code conversions

Code conversion is used to change the data present in one type of binary code to another type of binary code. Some of the codes are BCD, Gray, Excess 3, ASCII and so on.

Binary to Gray code conversion:

The Gray code is non-weighted code, as the position of bit does not contain any weight. The Gray code is a reflective digital code which has the special property that any two subsequent numbers codes differ by only one bit. This is also called a unit- distance code.

The generation of 4-bit Gray code can be calculated by using formula.

$G_1 = B_1$

$G_2 = B_1 \oplus B_2$

$G_3 = B_2 \oplus B_3$

$G_4 = B_3 \oplus B_4$

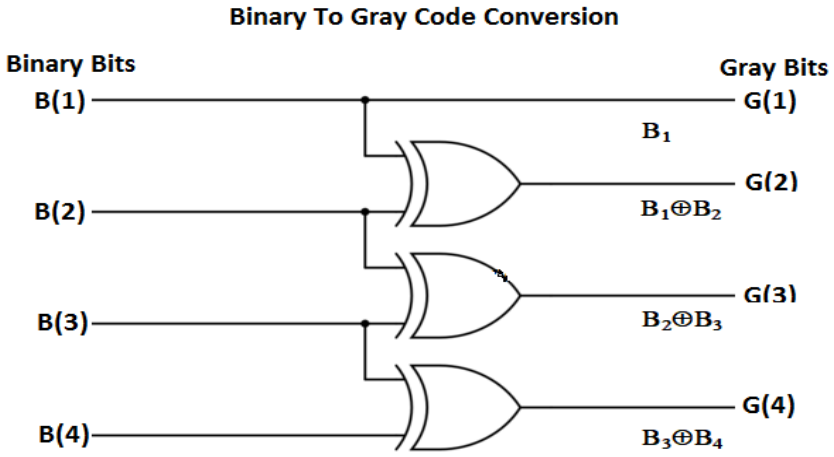
The most significant bit (MSB) of the Gray code is always equal to the MSB of the given binary code other bits of the output Gray code can be obtained by Xoring binary code bit at that index and previous index.

The binary to Gray code conversions can be done by using xoring logic gate. A four-bit binary code converter is shown above. The input is binary code and the output is equivalent Gray code.

A four-bit binary to Gray code conversion table is as shown below.

Four Bit Binary Number				Four Bit Gray Code			
B_1	B_2	B_3	B_4	G_1	G_2	G_3	G_4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1

0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0



Gray to binary code conversion:

In Gray to binary conversion, the input is Gray code and output is its equivalent binary code. The generation of four-bit binary equivalent code can be calculated by using formula.

$B_1 = G_1$

$B_2 = G_2 \oplus B_1$

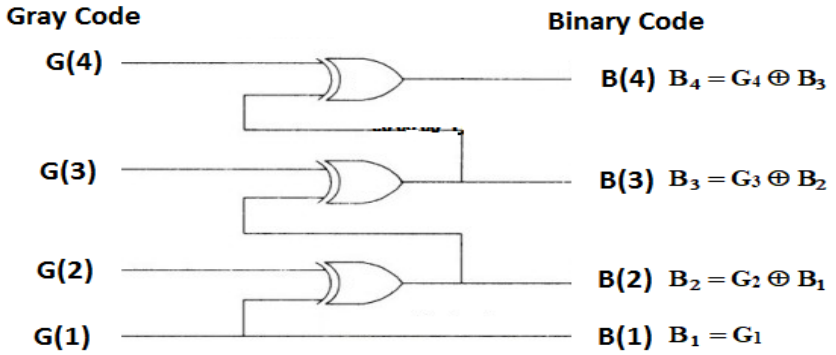
$B_3 = G_3 \oplus B_2$

$B_4 = G_4 \oplus B_3$

The MSB of binary code is similar to the MSB of Gray code. To get next bit, use the xoring operation among the MSB of binary to the next bit of the Gray code. Similarly, to get the third bit, it uses the xoring operation among the second bit to the third MSB of the Gray code and so on.

The Gray to binary conversion method can be done by using xoring logic gate. A four-bit Gray to binary code converter is as shown below.

Gray to Binary Code Conversion



A four-bit Gray to binary code conversion table is as shown below.

Four Bit Gray Code				Four Bit Binary Number			
G_1	G_2	G_3	G_4	B_1	B_2	B_3	B_4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

BCD to Excess 3 code:

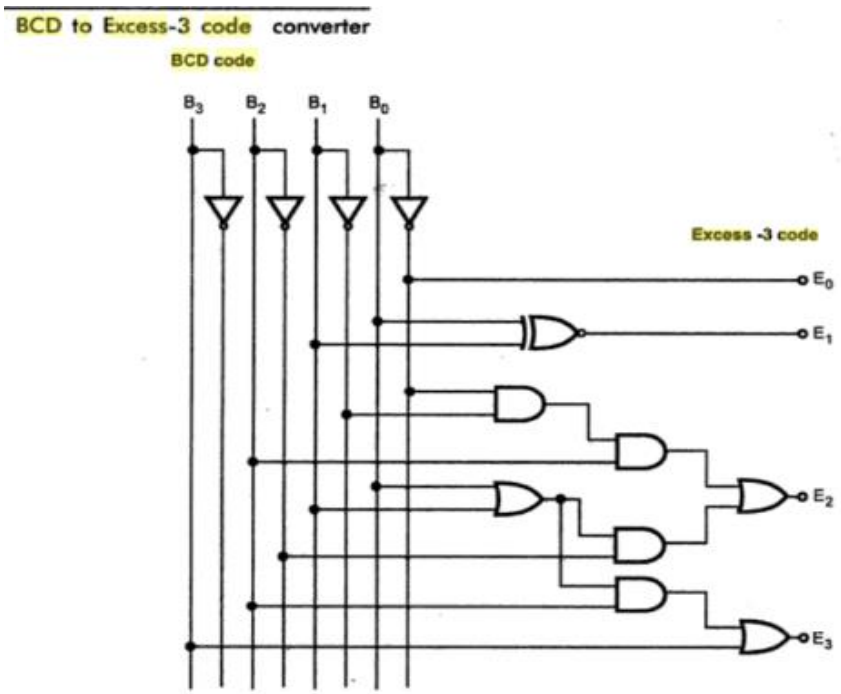
Excess-3 codes are non-weighted and can be obtained by adding 3 to each decimal digit then it can be represented by using 4-bit binary number for each digit.

To find the decimal equivalent of the given binary number. Add 0011 to each four-bit group in binary coded decimal number (BCD) to get desired excess-3 equivalent. The variables B_0, B_1, B_2, and B_3 represent the bits of the binary numbers. The variable 'B_0' represents the LSB, and the variable 'B_3' represents the MSB. The variables E_0, E_1, E_2, and E_3 represent the bits of the Excess-3 code. The variable 'E_0' represents the LSB, and the variable 'E_3' represents the MSB.

The truth table for BCD to Excess-3 code converter can be determined as shown in table below. For impossible four bit Excess-3 code we use output as Don't care conditions. The 'don't care conditions' is defined by the variable 'X'.

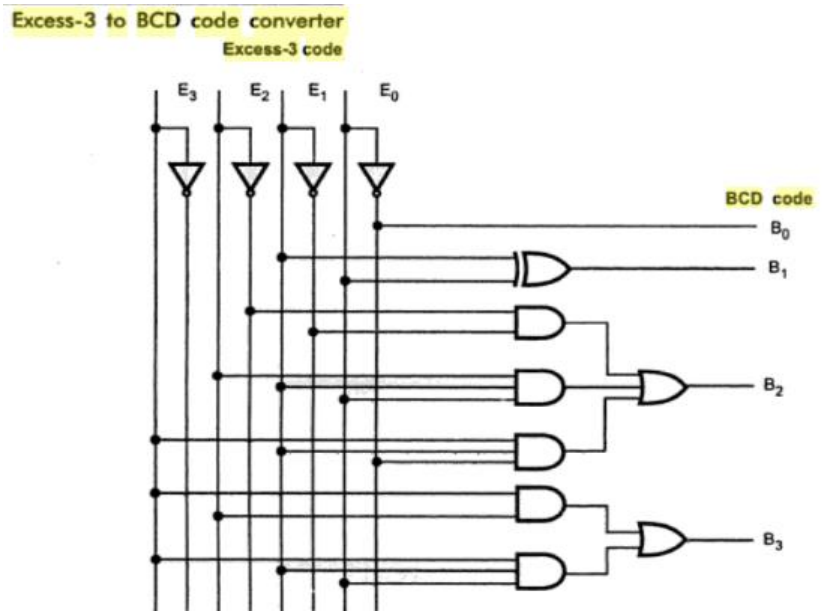
	BCD Number				Excess-3 Code Number			
Decimal	**B_3**	**B_2**	**B_1**	**B_0**	**E_3**	**E_2**	**E_1**	**E_0**
0	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**
1	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**
2	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**1**
3	**0**	**0**	**1**	**1**	**0**	**1**	**1**	**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**

The circuit diagram for BCD to Excess-3 code converter is shown below



Excess-3 to BCD converter:

The process of converting Excess-3 to BCD is opposite to the process of converting BCD to Excess-3. The BCD code can be calculated by subtracting 3, i.e., 0011 from each four-digit Excess-3 code. The variables E_0, E_1, E_2, and E_3 represent the bits of the Excess-3 code. The variable 'E_0' represents the LSB, and the variable 'E_3' represents the MSB. In the same way, the variables B_0, B_1, B_2, and B_3 represent the bits of the binary numbers. The variable 'B_0' represents the LSB, and the variable 'B_3' represents the MSB. The 'don't care conditions' is defined by the variable 'X'.



Below is the truth table for the conversion of Excess-3 code to BCD.

	Excess-3 Code Number				BCD Number			
Decimal	**E_3**	**E_2**	**E_1**	**E_0**	**B_3**	**B_2**	**B_1**	**B_0**
0	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**
1	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**
2	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**0**
3	**0**	**1**	**1**	**0**	**0**	**0**	**1**	**1**
4	**0**	**1**	**1**	**1**	**0**	**1**	**0**	**0**
5	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**1**
6	**1**	**0**	**0**	**1**	**0**	**1**	**1**	**0**
7	**1**	**0**	**1**	**0**	**0**	**1**	**1**	**1**
8	**1**	**0**	**1**	**1**	**1**	**0**	**0**	**0**
9	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**1**

QUESTION BANK

ESSAY QUESTIONS

1. Explain 1's & 2's Complement of a number in binary system with example. Explain 2's complement method of subtraction by suitable example.
2. Write Binary Addition rules. Add the following using Binary addition
   (a) $(10111)_2$ and $(10101)_2$ (b) $(10110)_2$ and $(11011)_2$.

SHORT ANSWER QUESTIONS

1. Explain the process of converting BCD to GRAY code
2. Explain the process of converting GRAY code to BCD.

PROBLEMS

1. Convert the following (1) $(11011)_2$ to (?)$_{10}$ (2) $(78)_{10}$ to (?) $_2$
2. Convert the Hexadecimal numbers (ACB) & (CAD) in to binary system.
3. Convert the following (a) (ACB)$_{16}$ → (?)$_2$ (b) (11010101)$_2$ → (?)$_{16}$
4. Convert the following (a) (1101.110)$_2$ → (?)$_{10}$ (b) (56)$_{10}$ → (?)$_2$
5. Find the equivalent Binary for the Gray (10011)