# Paper 7, TDC Part-3
# Chapter– 3, Number Systems and Codes
# Electronics
# Lecture - 8

**By:**

**Mayank Mausam**

**Assistant Professor (Guest Faculty)**

**Department of Electronics**

**L.S. College, BRA Bihar University,**

**Muzaffarpur, Bihar**

**Gray Code: -**

In this code system a decimal number is represented in binary form in such a way that each preceding or succeeding decimal number in Gray-code differ by a single bit, hence it is a cyclic code.

It is not a weighted code. So it can not be used in arithmetic operation.

It is a reflected code and is used extensively for shaft encoders due to its representation property.

# Number Systems and Codes

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

**(1) 4- bit straight binary code and gray code representation of decimal numbers 0 to 13**

# Number Systems and Codes

We can proceed further for more decimal numbers. Gray code can be constructed using the property as mentioned below: -

a) An n-bit (n>=2) gray code will have first $2^{n-1}$ Gray codes of (n-1)- bits written in order with a leading 0 appended. As it can be seen from table 1. For 0 to 7 we have MSB as 0.

b) Next $2^{n-1}$ Gray codes will be written with leading 1 appended of first $2^{n-1}$ Gray codes of (n-1)- bits in reverse order. As it can be seen from table 1. For 8 to 15 we have MSB as 1. Here written upto 13 only.

# Number Systems and Codes

**Conversion of binary code into gray-code / Process to write gray code from binary code**

Binary to gray code conversion can be learn from example.

Let the write the gray code for 13.

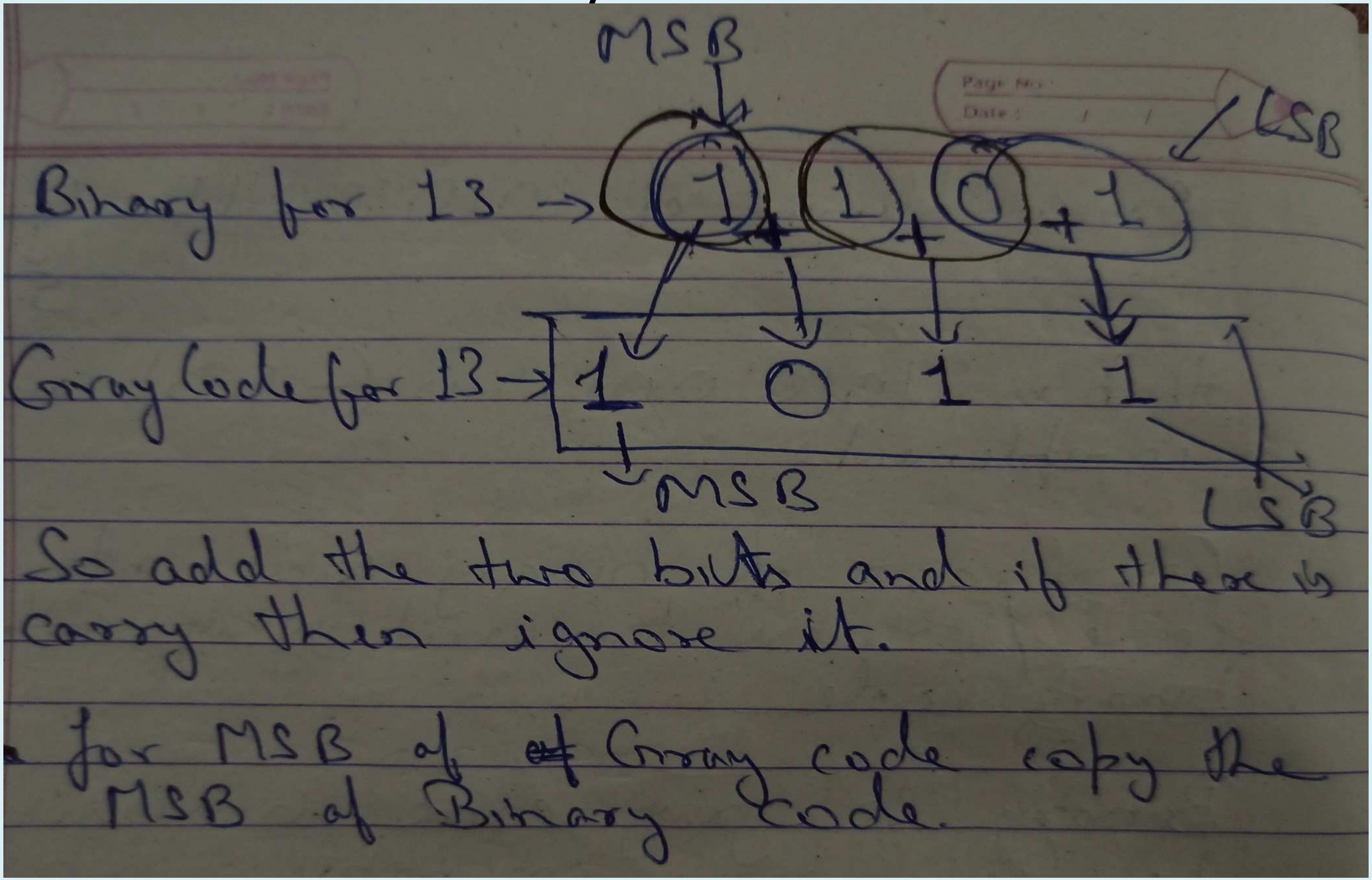
MSB → LSB

Binary Representation of 13 → 1 1 0 1

To write gray code for 13 →

To write LSB of gray code add the LSB of binary code with next higher bit of binary code.

# Number Systems and Codes



MSB

Binary for 13 → 1 + 1 + 0 + 1 LSB

Gray Code for 13 → 1 0 1 1

MSB LSB

So add the two bits and if there is carry then ignore it.

- for MSB of Gray code copy the MSB of Binary Code.

# Number Systems and Codes

Example 2) Write gray code for 6, 23, & 37.

Soln → 1) Binary for 6 → 1 + 1 + 0

Gray for 6 → 1 0 1

→ ignore carry

(2) Binary for 23 → 1 + 0 + 1 + 1 + 1

Gray for 23 → 1 1 1 0 0

Ignore Carry

So gray for 23 → 1 1 1 0 0

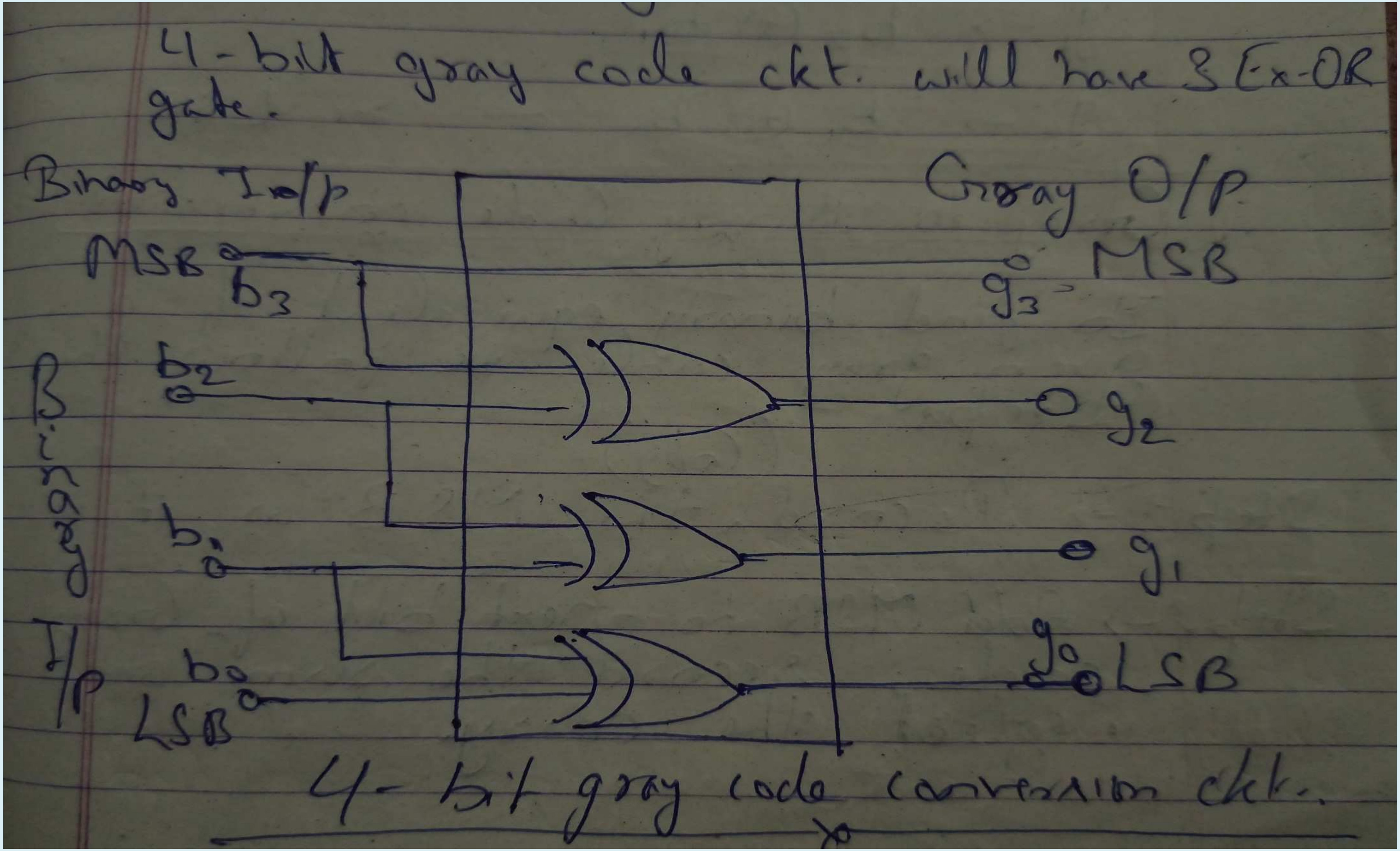
# Number Systems and Codes

(3) Binary for 37 → 1 + 0 + 0 + 1 + 0 + 1

Gray for 37 → 1 1 0 1 1 1

So we can have ckt that can convert binary code into gray code. The circuit that convert binary to gray code consist of Ex-or gate. To implement n-bit gray code we need (n-1) Ex-or gates.

# Number Systems and Codes



# Number Systems and Codes

So we can write expression for different bit position of gray code from binary code

for, 4- bit gray code

$g_3 \rightarrow b_3$ ⟶ i.e. MSB

$g_2 \rightarrow b_3 \oplus b_2$

$g_1 \rightarrow b_2 \oplus b_1$

$g_0 \rightarrow b_1 \oplus b_0$ ⟶ i.e. LSB

## Gray to Binary Code conversion

To find binary equivalent of a given gray code we follow the steps:

(Copy)

Step 1 → Record the MSB as it is.

Step 2 → Add MSB to next bit of Gray code, record the sum and neglect the carry.

# Number Systems and Codes

Step3:→ Repeat the process.

Example:→ Convert gray code of 11100 ($23_{10}$) into its equivalent binary.

Soln. Gray Code → 1 1 1 0 0

To write binary Code → 1 0 1 1 1

Copy MSB

Ignore Carry If there is carry.

$(10111)_2 = 23_{10}$

# Number Systems and Codes

So from here we can write expression also for different bit position of binary code from gray code.

For, 4-bit binary code from 4-bit gray.

$b_3 \rightarrow g_3 \longrightarrow$ i.e. MSB

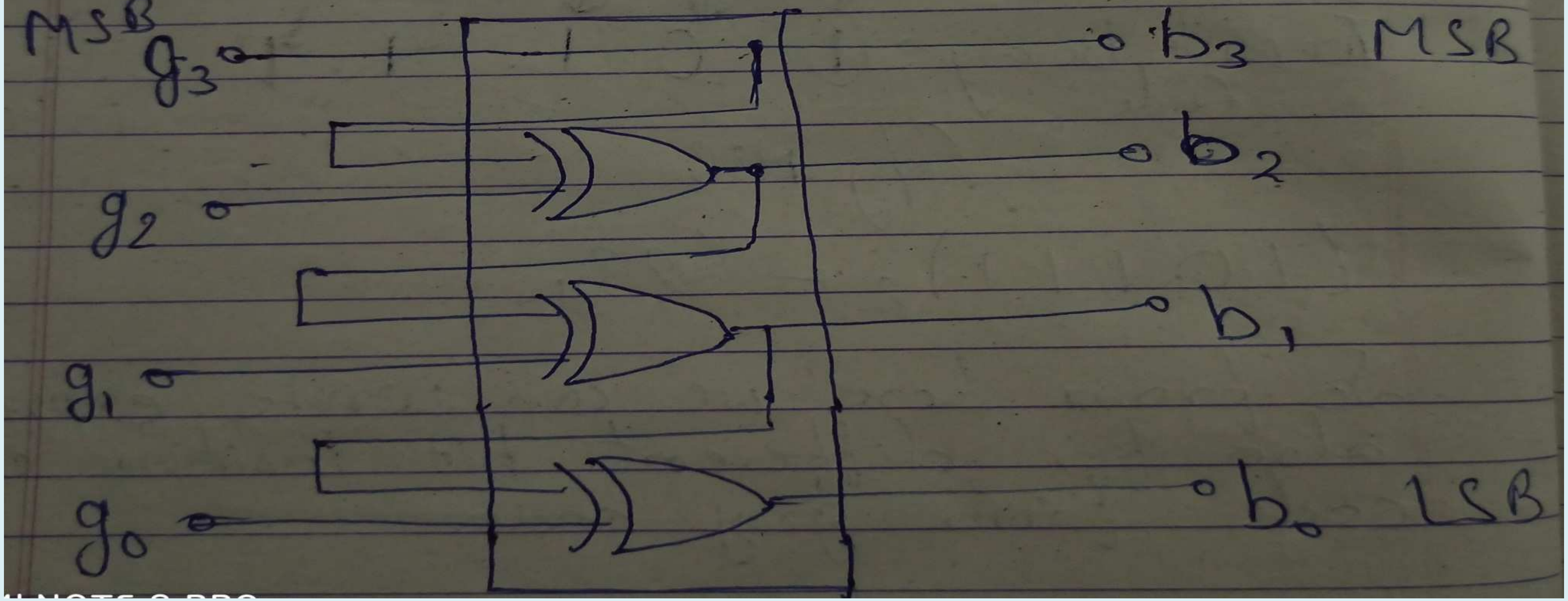
$b_2 \rightarrow b_3 \oplus g_2$

$b_1 \rightarrow b_2 \oplus g_1$

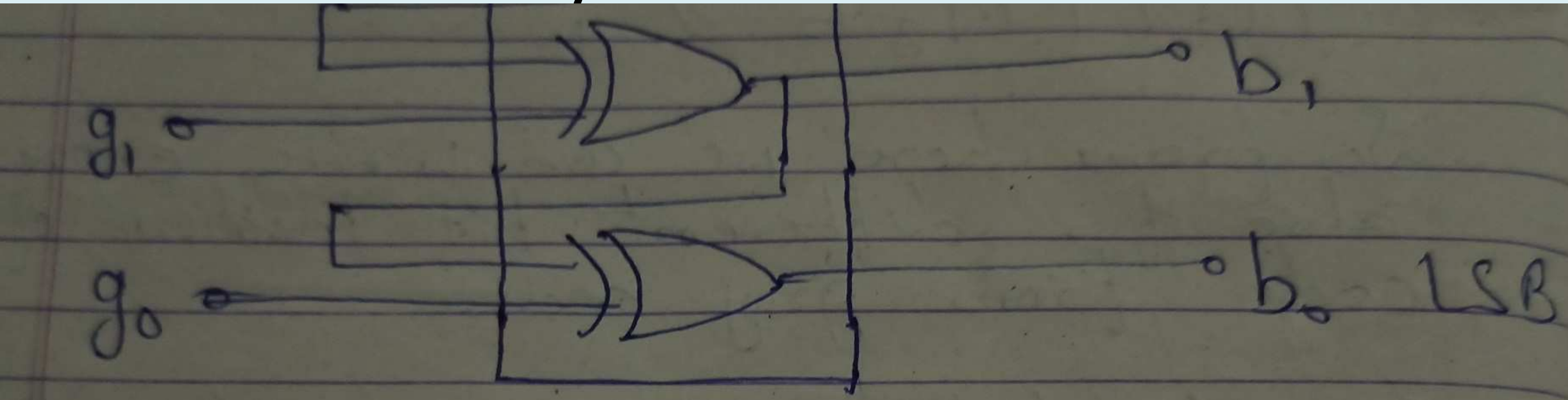
& $b_0 \rightarrow b_1 \oplus g_0 \longrightarrow$ i.e. LSB

So the ckt of gray code to binary code ~~be~~ can be construct using Ex-OR gate..

4-bit gray code to binary code need 3 Ex-OR Gates. as constructed below,



# Number Systems and Codes




Ckt for 4-bit gray code to 4-bit binary code.

Example (2)(a) Convert Gray Code (1 1 0 1 1 1) i.e. of $37_{10}$ to it's equivalent binary

(b) Convert gray code 1 0 0 0 ($15_{10}$) to it's equivalent binary.

# Number Systems and Codes

Solution:- (a) Gray Code → 1 1 0 1 1 1

Binary Code → 1 0 0 1 0 1

Ignore Carry

So Binary Code is $(100101)_2$ i.e. $37_{10}$

(b) Gray Code → 1 0 0 0

Binary Code → 1 1 1 1

So Binary Code is $(1111)_2$ i.e. $15_{10}$ for gray code (1000)

**Octal Code: -**

Octal code is nothing but the representation of octal number system with 3-bit straight binary numbers. The octal system symbol 0-7 is represent with 3-bits because 3-bits are sufficient to represent maximum symbol value 7 of octal system.

This code is used for binary inputs in digital computers, microprocessors. We have already seen octal number system in previous lectures.

Example :- Octal symbol 0 is 000, 1 is 001, 2 is 010 ……….. 6 is 110 and 7 is 111,

## Hexadecimal Code: -

Hexadecimal code is the representation of hexadecimal number system with 4-bit straight binary numbers. The hex system symbols 0-9 and A-F can be represented with 4-bits because 4-bits are sufficient to represent maximum symbol value F (15) of hexadecimal system.

This code is also used for inputs/outputs in digital computers, microprocessors etc. We have already seen hexadecimal number system in previous lectures.

Example :- Hex symbol 0 is 0000, 1 is 0001, 2 is 0010 ……….. E is 1110 and F is 1111, Number Systems and Codes

**Alphanumeric Code: -**

When we work on digital systems we need to deal with data that are numerals, letters and special symbols.

For example (a) maintaining the records of students of an university uses digital system i.e. computers. In this the name of the students, roll number, DoB, subjects, etc. are to be represented in binary form.

b) Reservation of tickets at counter also uses computer to record the data of passengers like their name, address, age, journey details, etc in binary form.

# Number Systems and Codes

There are lots of such application in which data are fed in digital systems in the form of alphanumeric codes.

Therefore it is necessary to have binary code for alphabets and symbols also. As an N-bit binary code can represent $2^n$ elements in the code.

There is need to represent more than 64 characters to represent 10 digits 0-9, 26 lowercase & 26 uppercase alphabets i.e. 52 alphabets and symbols for the transmission of digital information. Following two codes are normally used:

1. Extended BCD Interchange code (EBCDIC)
2. American Standard Code for Information Exchange (ASCII)

## Number Systems and Codes

Refer book- Modern Digital Electronics by RP Jain page number 81 to 83 for binary representation of alphanumeric characters in EBCDIC and ASCII.

***Thank You***