SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with 'A' Grade by NAAC

Lecture session
SCSA1201- FUNDAMENTALS OF DIGITAL SYSTEMS
Topic: Binary codes

By
V.GEETHA
ASSISTANT PROFESSOR/EEE
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
CHENNAI-119

BINARY CODES

In the coding, when numbers, letters or words are represented by a specific group of symbols, it is said that the number, letter or word is being encoded. The group of symbols is called as a 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.

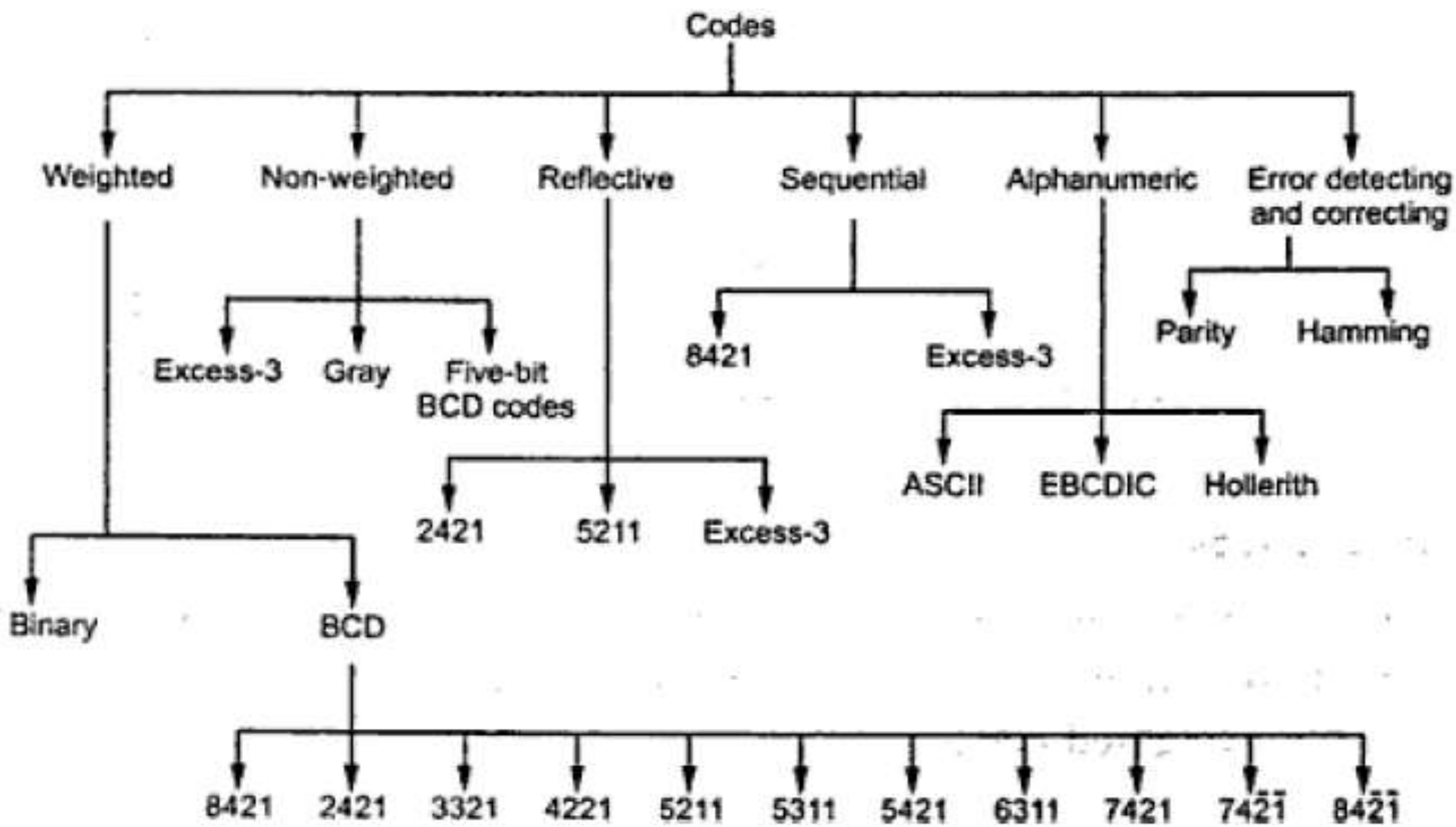
Advantages of Binary Code

Following is the list of advantages that binary code offers.

1. Binary codes are suitable for the computer applications.
2. Binary codes are suitable for the digital communications.
3.Binary codes make the analysis and designing of digital circuits if we use the binary codes.
4. Since only 0 & 1 are being used, implementation becomes easy.

Classification of binary codes

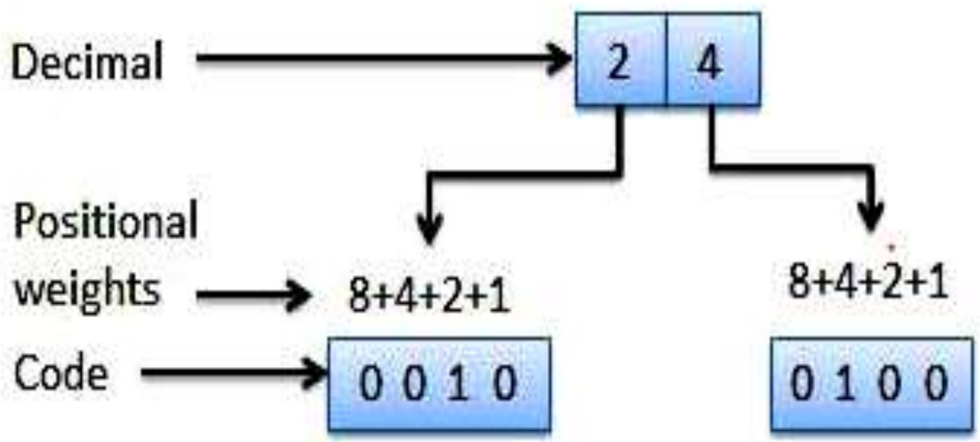
The codes are broadly categorized into following four categories.



Binary codes block diagram

Weighted Codes

Weighted binary codes are those binary codes which obey the positional weight principle. Each position of the number represents a specific weight. Several systems of the codes are used to express the decimal digits 0 through 9. In these codes each decimal digit is represented by a group of four bits.



Non-Weighted Codes

In this type of binary codes, the positional weights are not assigned. The examples of non-weighted codes are Excess-3 code and Gray code.

Excess-3 code

The Excess-3 code is also called as XS-3 code. It is non-weighted code used to express decimal numbers. The Excess-3 code words are derived from the 8421 BCD code words adding $(0011)_2$ or (3)10 to each code word in 8421. The excess-3 codes are obtained as follows −

$$\text{Decimal Number} \longrightarrow \text{8421 BCD} \xrightarrow[\text{0011}]{\text{Add}} \text{Excess-3}$$

Example

Decimal	BCD 8 4 2 1	Excess-3 BCD + 0011
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

Gray Code

It is the non-weighted code and it is not arithmetic codes. That means there are no specific weights assigned to the bit position. It has a very special feature that, only one bit will change each time the decimal number is incremented as shown in fig. As only one bit changes at a time, the gray code is called as a unit distance code. The gray code is a cyclic code. Gray code cannot be used for arithmetic operation.

Decimal	BCD	Gray
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1
3	0 0 1 1	0 0 1 0
4	0 1 0 0	0 1 1 0
5	0 1 0 1	0 1 1 1
6	0 1 1 0	0 1 0 1
7	0 1 1 1	0 1 0 0
8	1 0 0 0	1 1 0 0
9	1 0 0 1	1 1 0 1

Application of Gray code

- Gray code is popularly used in the shaft position encoders.
- A shaft position encoder produces a code word which represents the angular position of the shaft.

Binary Coded Decimal (BCD) code

In this code each decimal digit is represented by a 4-bit binary number. BCD is a way to express each of the decimal digits with a binary code. In the BCD, with four bits we can represent sixteen numbers (0000 to 1111). But in BCD code only first ten of these are used (0000 to 1001). The remaining six code combinations i.e. 1010 to 1111 are invalid in BCD.

Decimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Advantages of BCD Codes

- It is very similar to decimal system.
- We need to remember binary equivalent of decimal numbers 0 to 9 only.

Disadvantages of BCD Codes

- The addition and subtraction of BCD have different rules.
- The BCD arithmetic is little more complicated.
- BCD needs more number of bits than binary to represent the decimal number. So BCD is less efficient than binary.

2. BCD code (8421 code)

- ***Simplest form***: each decimal digit is replaced by its binary equivalent.

Example1: 937.25 is represented by

937.25

1001 0011 0111 0010 0101

(937.25)= (100100110111.00100101)$_{BCD}$

- This representation is referred to as ***"Binary-Coded-Decimal"*: BCD** or more explicitly as **8-4-2-1(8421 code).**

> **Note:**
> The result is quite different than that obtained by converting the number as a whole into binary.

Example 2:

854_{10} = $100001010100_{(BCD)}$

- BCD is ***inefficient***, e.g. to represent **999** and **999999** bits needed:
  - 10 and 20 in binary numbers
  - 12 and 24 for BCD code.

Decimal numbers	8421(BCD)	6311	642-3
0	0000	0000	0000
1	0001	0001	0101
2	0010	0011	0010
3	0011	0100	1001
4	0100	0101	0100
5	0101	0111	1011
6	0110	1000	0110
7	0111	1001	1101
8	1000	1011	1010
9	1001	1100	1111

Example 3: convert 0110100000111001(BCD) to its decimal equivalent.

Solution:

Divide the BCD number into four-bit groups and convert each to decimal:

$$\underset{6}{\underset{\downarrow}{0110}} \quad \underset{8}{\underset{\downarrow}{1000}} \quad \underset{3}{\underset{\downarrow}{0011}} \quad \underset{9}{\underset{\downarrow}{1001}}$$

0110100000111001(BCD) = 6839_{10}

- BCD is used in interfacing between a digit device and a human being, e.g. digital voltmeter (DVM).

Example 4: Convert the following decimal and binary numbers to BCD.

a) 5648_{10}
b) 10001101_2

Solution:

a) 5648_{10} =0101 0110 0100 1000
b) 10001101_2=141_{10}=0001 0100 0001

Example 5: convert the BCD number 011111000001 to its decimal equivalent.

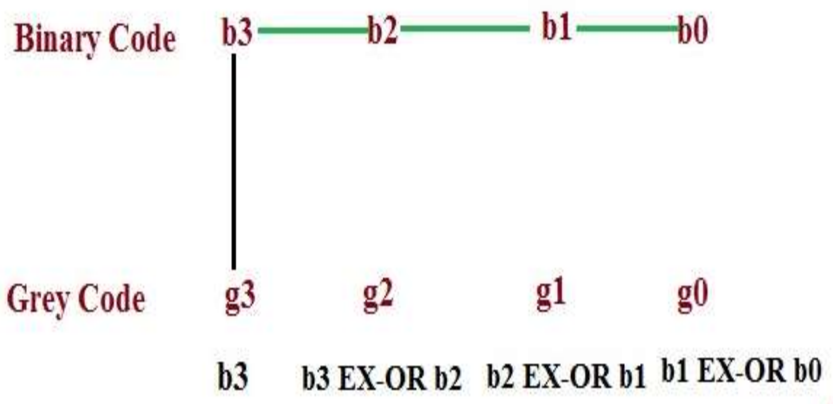
0111 1100 0001_{BCD} = error

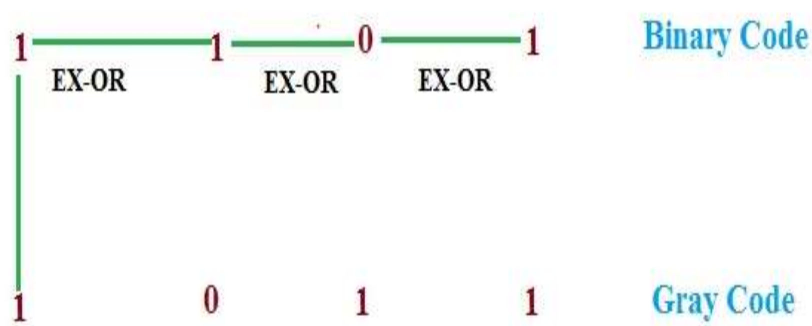
Doesn't exist in the BCD Code

3. Alphanumeric codes

- ✓ A complete alphanumeric code would include the ***26 lowercase characters, 26 uppercase characters, 10 numeric digits, etc.***
- ✓ There are ***many choices of codes sets*** to represent alphanumeric characters and several control characters.
- ✓ ***Two*** well accepted code sets are used for information coding:
  - **EBCDIC code**: ***extended binary coded decimal interchange code.***
  - **ASCII Code:** ***American standard code for information interchange:*** The ASCII code is a ***seven-bit code***, and so it has $2^7 = 128$ possible code groups.

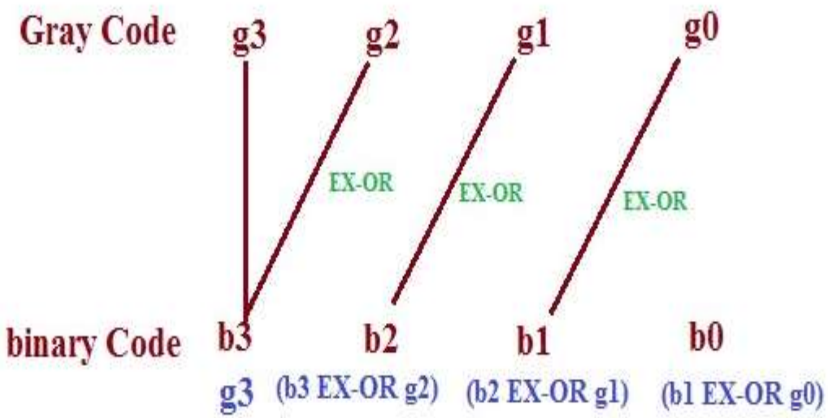
BINARY TO GRAY CONVERSION

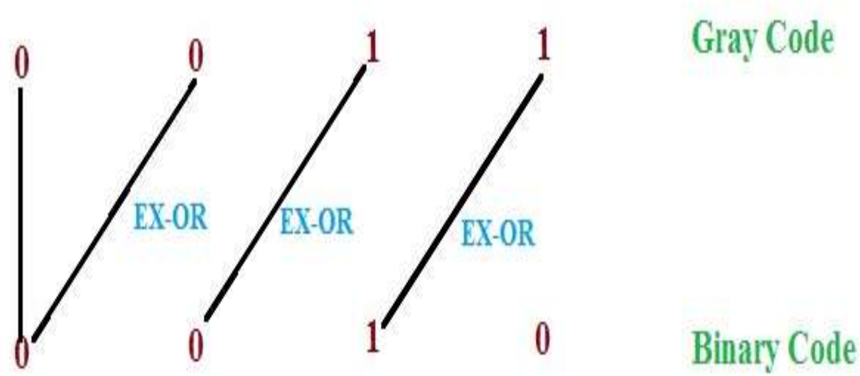




Decimal numbers	Binary code	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

GRAY TO BINARY CONVERSION





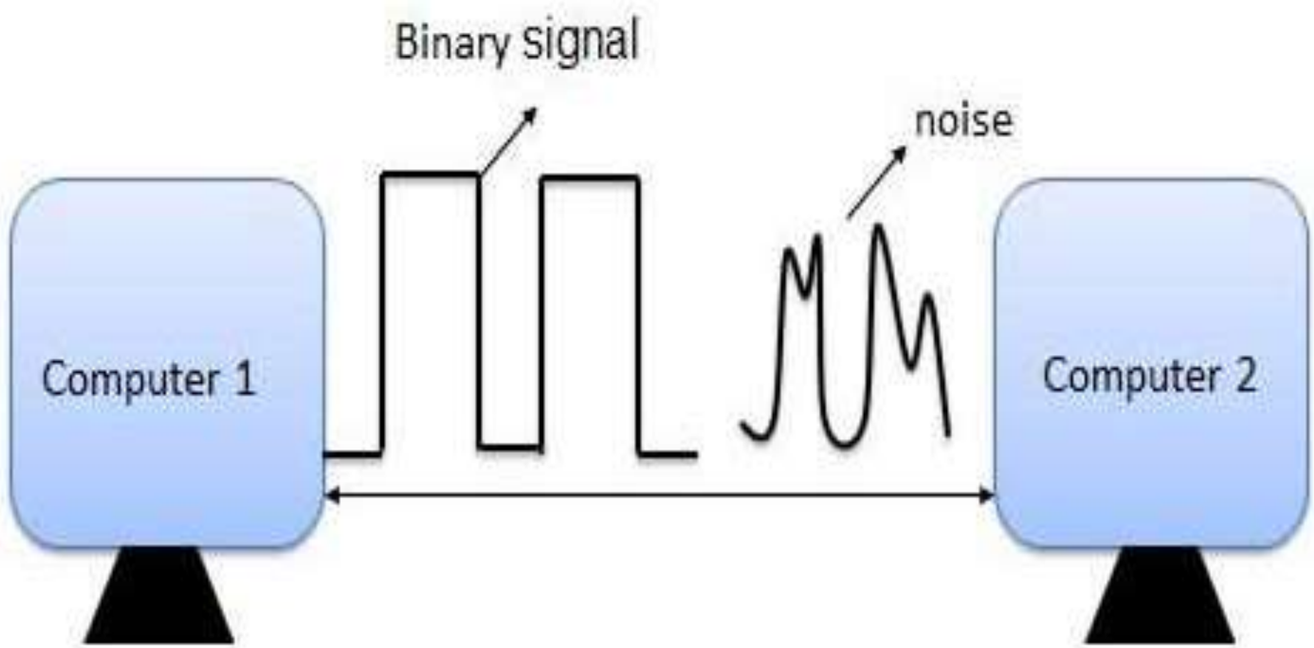
4 bit Gray Code	4 bit Binary Code
A B C D	B_4 B_3 B_2 B_1
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1
0 0 1 1	0 0 1 0
0 0 1 0	0 0 1 1
0 1 1 0	0 1 0 0
0 1 1 1	0 1 0 1
0 1 0 1	0 1 1 0
0 1 0 0	0 1 1 1
1 1 0 0	1 0 0 0
1 1 0 1	1 0 0 1
1 1 1 1	1 0 1 0
1 1 1 0	1 0 1 1
1 0 1 0	1 1 0 0
1 0 1 1	1 1 0 1
1 0 0 1	1 1 1 0
1 0 0 0	1 1 1 1

Error detection codes (Parity).

- Weighted codes and non-weighted codes are used to represent the ***decimal numbers***.
- Alphanumeric codes are used to represent the ***numeric and nonnumeric data (characters)***.
- Error detection codes are used to ***detect the errors during the data transmission.***
- Weighted codes use ***4 binary digits to represent (0-9) decimal numbers.***

What is Error?

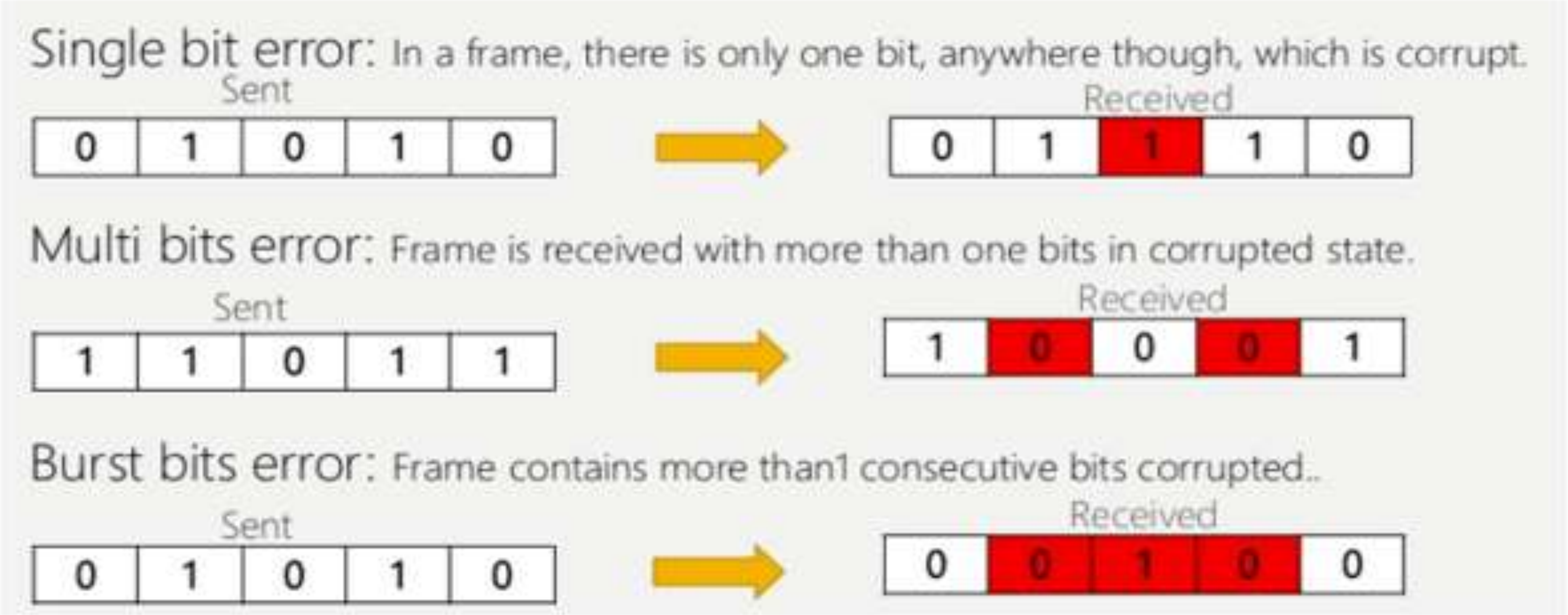
Error is a condition when the output information does not match with the input information. During transmission, digital signals suffer from noise that can introduce errors in the binary bits travelling from one system to other. That means a 0 bit may change to 1 or a 1 bit may change to 0.



Types of Errors

There are mainly three types of a bit error that occur in data transmission from the sender to the receiver.

- Single bit errors
- Multiple bit errors
- Burst errors



What is Error Detection and Correction?

In digital communication system error will be transferred from one communication system into another. If these errors are not detected and corrected, then the data will be lost. For effective communication, system data should transfer with high accuracy. This will be done by first identifying the errors and them correcting them.

Error detection is a method of detecting the errors which are present in the data transmitted from a transmitter to receiver in a data communication system.

Here, you can use redundancy codes to find these errors, by adding to the data when it is transmitted from the source. These codes is called "Error detecting codes".

Three types of error detection codes are:

Parity Checking

Cyclic Redundancy Check (CRC)

Longitudinal Redundancy Check (LRC)

Parity Check

1. Even Parity
2. Odd Parity

Example: Data Received is 10110101. Check for even parity scheme.

Check number of 1's in the Received data.

10110101

There are 5 1's present,

so the Received data is error data.

Example: Data Received is 110101011. Check for odd parity scheme.

Check number of 1's in the Received data.

10110101

There are 5 1's present,

so the Received data is correct data.

Single bit Error detection can be done by parity scheme.

Problem: Encode the binary word 1101 into 7 bit length hamming code using odd parity

Solution:

- Step1: Calculate the no of parity bits required
- Step2: Identify the position of the parity bits
- Step3: Structure the data and parity bits together
- Step4: Assign values for the parity bits (for odd/even parity)
- Step5: Write the hamming code(data+parity)

Computing parity

Step 1: No. of Parity bits depends on the length of the data

$$\mathbf{2^P \geq P+x+1}$$

where x= no. of data bits(known)

P=no .of parity bits(unknown)

P is found by trial and error method

Eg. Data=1101, where x=4,

If p=2, $\mathbf{2^2 \geq 2+4+1}$ **is false**

If p=3, $\mathbf{2^3 \geq 3+4+1}$ **is true**

Positioning of the parity bits

Step2

No of data bits = 4

No of parity bits = 3

Parity bits take positions such as $\mathbf{2^0, 2^1, 2^2, 2^3}$**…..**

Thus parity positions are $\mathbf{P_1, P_2, P_4}$

Step3: Structure of 7 bit hamming code is

D7	D6	D5	P4	D3	P2	P1
1	1	0	?	1	?	?

unknown

Step 4: Assigning values to parity bits

Bit designation	D_7	D_6	D_5	P_4	D_3	P_2	P_1
Bit location	**7**	**6**	**5**	**4**	**3**	**2**	**1**
Binary location	**111**	**110**	**101**	**100**	**011**	**010**	**001**
Data bits	1	1	0	?	1	?	?
Parity bits							

Assigning values to parity bits (odd parity)

Bit designation	D_7	D_6	D_5	P_4	D_3	P_2	P_1
Bit location	7	6	5	4	3	2	1
Binary location	111	110	101	100	011	010	001
Data bits	1	1	0		1		1
Parity bits							

Since, D3 = 1, D5 =0, D7 = 1

P1 = 1

Assigning values to parity bits

Bit designation	D_7	D_6	D_5	P_4	D_3	P_2	P_1
Bit location	7	6	5	4	3	2	1
Binary location	111	110	101	100	011	010	001
Data bits	1	1	0		1	0	
Parity bits							

Since, D3 = 1, D6 =1, D7 = 1

P2 = 0

Assigning values to parity bits

Bit designation	D_7	D_6	D_5	P_4	D_3	P_2	P_1
Bit location	7	6	5	4	3	2	1
Binary location	111	110	101	100	011	010	001
Data bits	1	1	0	1	1		
Parity bits							

Since, $D_6 = 1, D_5 = 0, D_7 = 1$

$P_4 = 1$

Assigning values to parity bits

Bit designation	D_7	D_6	D_5	P_4	D_3	P_2	P_1
Bit location	7	6	5	4	3	2	1
Binary location	111	110	101	100	011	010	001
Data bits	1	1	0		1		
Parity bits				1		0	1

	D_7	D_6	D_5	P_4	D_3	P_2	P_1
Data + Parity →	**1**	**1**	**0**	**1**	**1**	**0**	**1**

Given data 1101

Generated hamming code: 1101101

A 7 bit hamming code is received as 1011011.Assume even parity scheme and state whether the received code is correct or wrong, if wrong locate the bit in error

Given: 1.Even parity scheme.

2. Received data = 1011011

Find : Received code is correct or wrong.

If wrong which bit is error bit.

How it can be corrected.

Solution:

Step 1: Even parity scheme

No. of 1's = odd/even in received code

1011011

= odd no. of 1's

Received code is wrong.

Step 2:

7 bit hamming code

D7	D6	D5	P4	D3	P2	P1
1	0	1	1	0	1	1

Check with parity bit P1

[1,3,5,7]

[1,0,1,1]

No of 1's = odd

P1 is a wrong data,

it should be corrected **(1)**

Check with parity bit P2

[2,3,6,7]

[1,0,0,1]

No of 1's = even

P1 is a correct data,

it need not be corrected **(0)**

Check with parity bit P4

[4,5,6,7]

[1,1,0,1]

No of 1's = odd

P4 is a wrong data,

it should be corrected **(1)**

P4	P2	P1
1	0	1

Equivalent of Binary value 101 = 5

The position of the error is 5^{th} bit in the received code

1011011

7	6	5	4	3	2	1
1	0	1	1	0	1	1

0

The corrected code is 1001011 which is actually transmitted

Assignment Questions

1.Convert the following octal number to binary $(2453)_8$
2.What is the purpose of self complimentary codes?
3.Illustrate the purpose of Hamming code with an example.
4.Perform the following conversion
$(673.6)_8$ to hexadecimal
$(E7C.B)_{16}$ to octal
5.Find the 9's complement of the following decimal number. 720499.
6.Represent the decimal number 6027 in (i) BCD (ii) excess-3 code.
7. Given the two binary numbers X = 1010100 and Y = 1000011. Perform the subtraction X – Y using 2's complement.
8.Find the excess -3 code and 9's complement of the number 40310
9.Convert $(9B2.1A)_{16}$ to its decimal equivalent
10.Explain in detail about the Error Detection & Error Correction.
11.Develop the Hamming code for the message 1101 with odd parity and the received code is 1100011, correct if any error.
12.Develop the Hamming code for the message 1001 with even parity and the received code is 1010001, correct if any error.
13.Develop 2's complement subtraction for 100101 – 100100
14.Simplify

a) $A(A+\overline{A})+B = AA+A\overline{A}+B$

b) $(A+B)(\overline{A}+B)\overline{B} = (A+B)(\overline{A}\,\overline{B}+B\overline{B})$

c) $\left(\overline{\overline{A}B+\overline{B}}\right)\left(\overline{B+\overline{A}B}\right)$

15. Brief about the classifications of binary codes.

Thank you!