# Digital Electronic

## Unit 1 - fundamental of digital system & logic families.

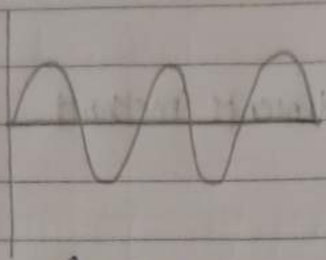
* Topics:
- advantages of digital
- Number system
- Binary
  Octal
  Decimal
  Hexadecimal

* Compliments method
- 1's
- 2's
- 9's
- 10's.

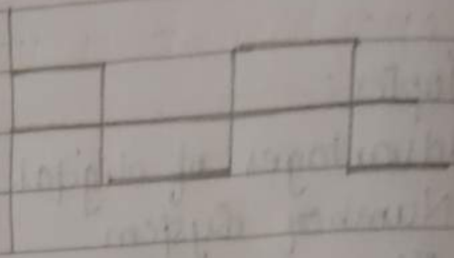
* Error Correcting & error detecting Codes.
- Logic gates
- Basic gates
- exclusive gates
- universal gates.

* Logic families
1 TTL, RTL, DTL, CMOS, Logic families } IC Gates

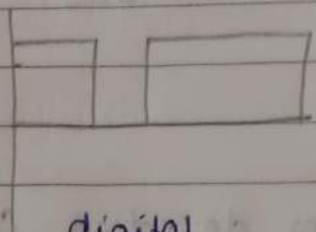
* Digital Signal:



Analog

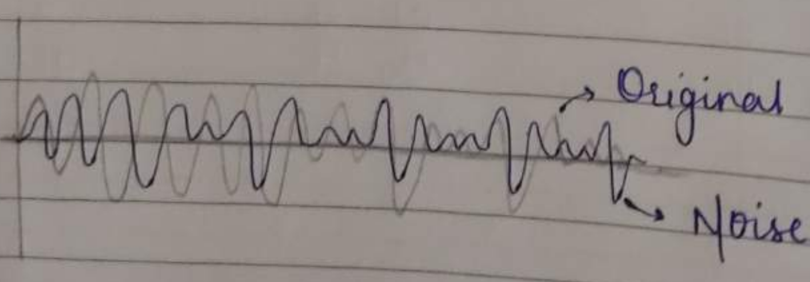


Analog

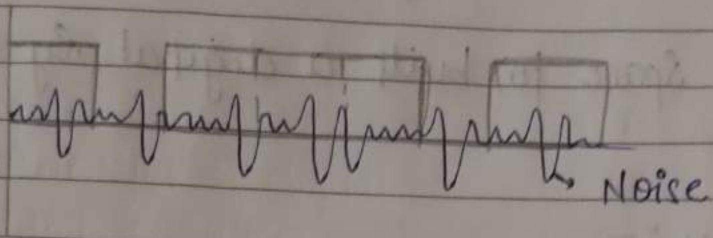


digital

- When Signal starts from -ve to +ve then it is Analog
- When Signal have Only +ve Value i.e 0 & 1 i.e digital
- filter - To remove unwanted Signal.



Transmission Convert → digital




Here we can easily filter it out.

0 → - → +
No Fluctuation.

② Storage :-

- Made up of Semiconductor
- i.e any SC device can store binary No. for long period of time
Ex :- hardisk

③ Easy to design :-

- In analog we need amplifier, filter, Convertor or many more Components are used to design the analog But In Digital this many Components are not Required to design.

④ It is Versatile :-
i.e it has Capability to adapt anything

(5) Accuracy & precision is very high.

(6) It Required less space to built a digital system

## * Number System:-

(1) Binary Number System :- 0, 1

(2) Octal number system :- 0 to 7

(3) Decimal Number System :- 0 to 9

4. Hexadecimal number System :- 0 to 9, A to F } 0 to 15

## * Conversions:

D → B - 2
O - 8
H - 16

O → B - 2
D - 10
H - 16

B → D - 10
O - 8
H - 16

H → B - 2
O - 8
D - 10

| No. sr. | Binary | Octal | Decimal | Hexadecimal |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |
| 2 | | 2 | 2 | 2 |
| 3 | | 3 | 3 | 3 |
| 4 | | 4 | 4 | 4 |
| 5 | | 5 | 5 | 5 |
| 6 | | 6 | 6 | 6 |
| 7 | | 7 | 7 | 7 |
| 8 | | 8 | 8 | 8 |
| 9 | | 9 | 9 | 9 |
| 10 | | | | A |
| 11 | | | | B |
| 12 | | | | C |
| 13 | | | | D |
| 14 | | | | E |
| 15 | | | | F |

Binary - $2^n$.

- Binary - 2 1
Octal - 4 2 1
Hexa - 8 4 2 1

• Table :-

| Deci | Binary | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 |
| 10 A | 1 | 0 | 1 | 0 |
| 11 B | 1 | 0 | 1 | 1 |
| 12 C | 1 | 1 | 0 | 0 |
| 13 D | 1 | 1 | 0 | 1 |
| 14 E | 1 | 1 | 1 | 0 |
| 15 F | 1 | 1 | 1 | 1 |

$(\_)_{16}$ $(\_)_{2}$ $(\_)_{8}$

Radix or base

# Conversion of Number Systems :-

## Binary to decimal :-

$(10101)_2$

→ $1\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0$

$16+0+4+0+1$

$(21)_{10}$

Method 2:-

| | | | | |
|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 |
| 16 | 8 | 4 | 2 | 1 |

add that no. which have one

$\therefore 16+4+1=21$

2. $(1001011)_2 \rightarrow (\ )_{10}$

→

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 64 | 32 | 16 | 8 | 4 | 2 | 1 |

$64+8+2+1$

$75$

② Binary to Octal :-

1. $(1001011)_2 \rightarrow (\ )_8$

Here we have to make 3-pair.

→

$\underbrace{0\ 0\ 1}_{1}, \underbrace{0\ 0\ 1}_{1}, \underbrace{0\ 1\ 1}_{3}$

$(113)_8$

2. (001011101)

→

$\underbrace{0\ 0\ 1}_{1}, \underbrace{0\ 1\ 1}_{3}, \underbrace{1\ 0\ 1}_{5}$

$(135)_8$

3 $(001011.101)_2$

→

$\underbrace{0\ 0\ 1}_{1}, \underbrace{0\ 1\ 1}_{3} . \underbrace{1\ 0\ 1}_{5}$

13.5

③ Binary to Hexadecimal :-

① $(01011010101)_2$

$$\begin{array}{cccccccccccc} 8&4&2&1&8&4&2&1&8&4&2&1 \end{array}$$

→ $\underbrace{0010}_{1}\,\underbrace{1101}_{6}\,\underbrace{0101}_{10\,=\,A}$

16A.

2. $(11101111111)_2$

$\underbrace{0111}_{7}\,\underbrace{0111}_{7}\,\underbrace{1111}_{15}$

77F.

3 $(11101.111111)$.

$\underbrace{0001}_{1}\,\underbrace{1101}_{D}\,.\,\underbrace{1111}\,\underbrace{1100}$

1D. F3

## ④ Octal to Binary

① $(35)_8$

→

3 5

↓ ↓

011 101

$\therefore (011101)_2$

2. $(367.52)_8$

→

3 6 7 . 5 2

↓ ↓ ↓ ↓ ↓

011 110 111 101 010

$\therefore (011110111.101010)_2$

Date
Page

⑤ Octal to decimal

① (4057.06)

→

| 4 | 0 | 5 | 7. | 0 | 6 |
|---|---|---|---|---|---|
| ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| $8^3$ | $8^2$ | $8^1$ | $8^0$ | $8^{-1}$ | $8^{-2}$ |

= 2048 + 40 + 7.093

2093.093

⑥ Octal to Hexadecimal.

① $(796.603)_8$

→

② $(514.033)_8$

→

| 5 | 1 | 4. | 0 | 3 | 3 |
|---|---|---|---|---|---|
| ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| $\underbrace{0001}_{1}$ | $\underbrace{0101}_{4}$ | $\underbrace{0100}_{12}$ | $\underbrace{0000}_{0}$ | $\underbrace{1101}_{11}$ | $\underbrace{1000}_{1}$ |

14C.0B

* Decimal to Binary

1 $(52)_{10}$

→

$$\begin{array}{r|ll} 2 & 52 & 0 \\ 2 & 26 & 0 \\ 2 & 13 & 1 \\ 2 & 6 & 0 \\ 2 & 3 & 1 \\ & 1 & 1 \end{array}$$

$(110100)_2$

2. $(163.875)_{10}$

→

$$\begin{array}{r|ll} 2 & 163 & 1 \\ 2 & 81 & 1 \\ 2 & 40 & 0 \\ 2 & 20 & 0 \\ 2 & 10 & 0 \\ 2 & 5 & 1 \\ & 2 & 0 \\ & 1 & 1 \end{array}$$

$0.875 \times 2 = 1.75$

$0.75 \times 2 = 1.5$

$0.5 \times 2 = 1$

$(10100011.111)_2$

* Decimal to Octal

· $(378.93)_{10}$

→

| | | |
|---|---|---|
| 8 | 378 | 2 |
| 8 | 47 | 7 |
| | 5 | 5 |

$0.93 \times 8 = 7.44$
$0.44 \times 8 = 3.5$
$0.52 \times 8 = 4.16$
$0.16 \times 8 = 1.28$

$(572.7341)_8$.

Extra Question

· $(59)_{10}$

→

| | | |
|---|---|---|
| 2 | 59 | 1 |
| 2 | 29 | 1 |
| 2 | 14 | 0 |
| 2 | 7 | 1 |
| 2 | 3 | 1 |
| | 1 | 1 |

$(111011)_2$.

· $(11011101)_2$

→ $\underbrace{011}_{3},\ \underbrace{011}_{3},\ \underbrace{101}_{5}$

$(335)_8$

* Decimal to Hexadecimal

1. $(2598)_{10}$

→

| | | |
|---|---|---|
| 16 | 2596 | 6 |
| 16 | 162 | 2 |
| 16 | 10 | 10 = A |
| | 0 | |

$(A26)_{16}$.

2. $(3430)_{10}$

→

| | | |
|---|---|---|
| 16 | 3430 | 6 |
| 16 | 214 | 6 |
| | 13 | 13 = D |

$(D66)_{16}$

3. $(3430.675)_{10}$

↓

→ D66

$0.675 \times 16 = 10.8$

$0.8 \times 16 = 12.8$

$0.8 \times 16 = 12.8$

$(D66.ACC)_{16}$

* Hexadecimal to Binary [pairing of 0 & 1 as 4]

- $(AB)_{16}$

→ A B

↓ ↓

1010 1000

$(10101000)_2$

- $(4BAC)_{16}$

→ 4 B A C

↓ ↓ ↓ ↓

0100 1011 1010 1100

$(0100101110101100)_2$

- $(3A9E.B0D)_{16}$

→ 3 A 9 E . B 0 D

↓ ↓ ↓ ↓ ↓ ↓ ↓

0011 1010 1001 1110 1011 0000 1101

$(0011101010011110.1011000001101)_2$

+ Hexadecimal to Octal. [Do binary of Hexa make pair of three then convert].

① $(B9F)_{16}$

→

B 9 F

↓ ↓ ↓

$\underbrace{101}_{5}\underbrace{110}_{6}\underbrace{011}_{3}\underbrace{111}_{7}$

$(5637)_8$

2. $(F5AD.B68)_{16}$

→

F 5 A D . B 6 8

↓ ↓ ↓ ↓ ↓ ↓ ↓

$\underbrace{001}_{1}\underbrace{111}_{7}\underbrace{010}_{2}\underbrace{110}_{6}\underbrace{101}_{5}\underbrace{101}_{5} . \underbrace{101}_{5}\underbrace{101}_{5}\underbrace{101}_{5}\underbrace{000}_{0}$

$(172655.555)_8$.

- Hexadecimal → Decimal [multiplying the Hex NO. by 16]

- $(5C7)_{16}$

→ $5 \times 16^2 + 12 \times 16^1 + 7 \times 16^0$

$1280 + 192 + 7$

$\underline{1479}$

- $(A0F9)_{16}$

→ $10 \times 16^3 + 0 \times 16^2 + 15 \times 16^1 + 9 \times 16^0$

$40960 + 0 + 245 + 9$

$\underline{41209}$

- $(A0F9.0EB)_{16}$

↓

→ $41209$ $\quad .01411 \times 16 = 2.25$

$0.25 \times 16 = 4.0$

$(41209.42)_{10}$

# * Compliments Functions.

① 1's Compliments } Conversion of binary
2 2's Compliments }
3 9's Compliments } conversion of decimal.
4 10's Compliments }

1 1's Compliment → 1 → 0
1 → 1

①
1 0 1 0 1 1
↓ ↓ ↓ ↓ ↓ ↓
0 1 0 1 0 0

| | Binary | 1's Compliment | |
|---|---|---|---|
| 0 | 000 | 111 | 7 |
| 1 | 001 | 110 | 6 |
| 2 | 010 | 101 | 5 |
| 3 | 011 | 100 | 4 |
| 4 | 100 | 011 | 3 |
| 5 | 101 | 010 | 2 |
| 6 | 110 | 001 | 1 |
| 7 | 111 | 000 | 0 |

If there are +ve & -ve no. then.

| 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 |

Signed bit

1 will Represent -ve Sign or -ve Number

| 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 |

0 will Represent +ve Sign or +ve No.

+5 → 0 0 1 0 1

1's → ~~3~~ 1 1 0 1

MSB → most Significant bit

LSB → Least Significant bit

∴ 1 1 0 1 0

-ve MSB (1)    +ve LSB (0)

* Subtraction 1's Compliment :-

1. Take 1's Compliment of Subahend.

2. Add with minuend

3. If the Result of above add Carried bit 1 then add it to the LSB (given result).

4. If there is NO carried bit then takes 1's Complim[ent] as a result which will give Answer.

Ex :- 10101 − 00101

10101 → minuend

00101 → Subhend

10101 + 11010 (1's Compliment)

$$\therefore \begin{array}{r} 10101 \\ 11010 \\ \hline 1\,01111 \end{array}$$

LSB

Add to LSB

$$\therefore \begin{array}{r} 01111 \\ 1 \\ \hline 10000 \end{array}$$

2) 11110 with 1110

S ↓ (11110), m ↓ (1110)

00001 + 1110

$$\begin{array}{r} 00001 \\ 1110 \\ \hline 01111 \end{array}$$

* Addition of 1's Compliment

(1) If Number is +ve or -ve then.

Condⁿ (1)

if +ve > -ve (then steps are different)

then do 1's Compliment of -ve No. & then add that given +ve No.

If Carried bit is Generated then add that Carried bit to LSB

(1) $\underset{14}{1110}$ & $-\underset{13}{1101}$

→ ∴ 1110 + 0010 (1's comp)

$$\begin{array}{r} 1110 \\ 0010 \\ \hline 1\,0000 \end{array}$$

0000 → LSB

└→ Add to LSB

$$\begin{array}{r} 0000 \\ 1 \\ \hline 0001 \end{array}$$

2) 1101 & -0101

13 > -5

$\downarrow$

1010 [1's Compliment]

```
 1101
 1010
------
10111 → LSB
```

→ Add to LSB

```
 0111
    1
-----
 1000
```

Condⁿ 2: +ve & -ve where -ve is greater

-ve > +ve

i) 0101 & -1101

5 < 13

$\downarrow$

0010

```
0010
0101
----
0111
```

$\xrightarrow[\text{Compliment again}]{\text{1's}}$ 1000

# * 2's Compliment

Step 1 ⟶ 1's Compliment
step 2 ⟶ Add to 1

① 0101
↓
1010
+ 1
1011

② 1010
↓
0101
+ 1
0110

# * Binary Additione :-

- properties

① 0+0=0
② 1+1=0
③ 0+1=1
1+0=1

Ex ① 1101
+ 111
10100

② 1010
111
10001

③ 11101.101
111.011
10101.000

* Binary Subtraction

① 
$$\left.\begin{array}{r} {}^{1}0 \\ -1 \\ \hline 1 \end{array}\right\} \rightarrow \text{Borrow function}$$

1-1=0
0-1=1
1-0=1
0-0=0

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

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

③
$$\begin{array}{r} 1\,0\,1\,0\,.\,0\,1\,0 \\ 1\,1\,1\,.\,1\,1\,1 \\ \hline 0\,0\,1\,1\,.\,0\,1\,1 \end{array}$$

* Binary Multiplication

① 1100 x 1001

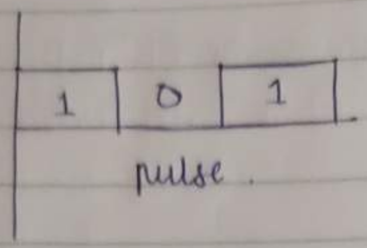
$$\begin{array}{r} 1100 \\ 1001 \\ \hline 1100 \\ 0000* \\ 0000** \\ 1100*** \\ \hline 1101100 \end{array}$$

② 1101 x 110

$$\begin{array}{r} 1101 \\ 110 \\ \hline 0000 \\ 1101* \\ 1101** \\ \hline 1001110 \end{array}$$

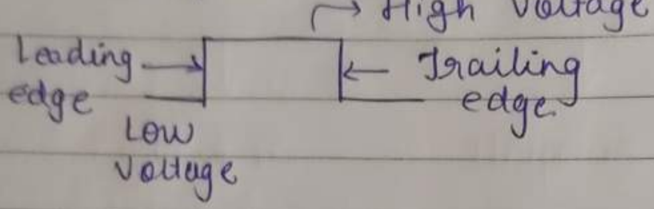
* Binary division

## * Digital Signal :- Binary Signal
0 & 1.




1 = High voltage
0 = low voltage

+ve pulse = 1




It is used to represent data in a Sequence of discrete value.

It is also called as binary Signal or logic signal.

It has two voltage level 0 & 1.
where 0 = low voltage & 1 = high voltage.

Low voltage level = 0 = Negative logic system.
= 0 volt.

High voltage level = 1 = positive logic system
= +5 volt.

* 0V, 0.5V, 0.8V } low voltage = 0 2V, 5V } High voltage = 1

- Range of voltage b/w 0.8V to 2V it is called as Indetermine Range.

Ex: 3.8 - High = 1

* Digital Circuit:
It is Circuit which as large no. of logic gates & packed with IC.

① Monolithic IC. - Single piece of SC i.e silicon

→ low cost
→ low power
→ smaller in size
→ High Reliable.

- SSI :- Small Scale Integration
  - less than 12 gates are Integrated.

- MSI :- Medium Scale Integration
  - more than 12 to 99 gate are used.

- LSI :- Large Scale Integration
  - 100 to 1000 logic gates

- VLSI – Very large Scale Integration
  10,000 to 1 lac logic gates integrated

- ULSI :- Ultra large Scale Integration
  more than 1 lac logic gates are integrated

*

IC
↓ Bipolar IC → BJT Transistor
↓ Unipolar IC → MOSFET

MOSFET
Metal oxide Semiconductor field effect Transistor

# * Binary Code:-

Numeric code (0 & 1)

Alphanumeric code

- ASCII Code → American Standard code for Information Interchange.
- EBCDIC → decimal Number are coded.

Alleast 4 binary bit

# * BCD Code:-

i.e Binary Coded Decimal.

| deci.digit | 8 4 2 1 |
|---|---|
| 0 | 0 0 0 0 |
| 1 | 0 0 0 1 |
| 2 | 0 0 1 0 |
| 3 | 0 0 1 1 |
| 4 | 0 1 0 0 |
| 5 | 0 1 0 1 |
| 6 | 0 1 1 0 |
| 7 | 0 1 1 1 |
| 8 | 1 0 0 0 |
| 9 | 1 0 0 1 |

BCD

Weighted Code

Non-weighted Code.

① 8421
② 2421
③ 5211.

* Weighted Code.

①

| digit | 2 4 2 1 |
|---|---|
| 0 | 0 0 0 0 |
| 1 | 0 0 0 1 |
| 2 | 0 0 1 0 |
| 3 | 0 0 1 1 |
| 4 | 0 1 0 0 |
| 5 | 1 0 1 1 |
| 6 | 1 1 0 0 |
| 7 | 1 1 0 1 |
| 8 | 1 1 1 0 |
| 9 | 1 1 1 1 |

Page

## Non-weighted Code:

- Xs-3 (excess-3) code
- Grary Code.

It do not assign with any weight to each digit position

| digit | Xs-3 | Xs-3 in dec |
|---|---|---|
| 0 | 0011 | 3 |
| 1 | 0100 | 4 |
| 2 | 0101 | 5 |
| 3 | 0110 | 6 |
| 4 | 0111 | 7 |
| 5 | 1000 | 8 |
| 6 | 1001 | 9 |
| 7 | 1010 | 10 = A |
| 8 | 1011 | 11 = B |
| 9 | 1100 | 12 = C |

Xs-3

0+3=3
1+3=4
2+3=5
3+3=6
4+3=7
5+3=8
6+3=9
7+3=10
8+3=11
9+3=12

0
1
2
13
14
15.

Unused decimal Number

0000
0001
0010
1101
1110
1111

Unused Bit in Xs-3 code.

* Error Detecting & Error Correcting Code

(1) While converting analog signal into digital sequence of bit & if any single bit change it's position then it results is major error (catastrophic error).

* Hamming Code error detection:

- It was developed by R.W Hamming. It is easy to implement.

Ex:- 7 bits is mostly used.

7 bits hamming code is used mostly which's data bits = 4

parity bit = 3 → $2^n$ where n = 0, 1.

| | | | | | | |
|---|---|---|---|---|---|---|
| $D_7$ | $P_6$ | $P_5$ | $P_4$ | $P_3$ | $P_2$ | $P_1$ |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 |

7 bits hamming code

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

$P_1 = P_3 P_5 P_7$

$P_2 = P_3 P_6 P_7$

$P_4 = P_5 D_6 D_7$

Receiver ← 0110001 — Transmitter

Even Parity

| 0 | 1 | 1 | 0 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|
| $P_7$ | $P_6$ | $P_5$ | $P_4$ | $P_3$ | $P_2$ | $P_1$ |

$P_1$ = [ | 0 1 0 ] even number (2)
↓ ↳ data ∴ no error
parity

$P_2$ = [ 0 | 010 ]
↓ ↳ data odd number 1's
parity ∴ no errors

$P_4$ : $P_5$ $P_6$ $P_7$
[0] [1 1 0]
↳ even
∴ no error

* Terminologies :-

① Data bit :-
The data which can be transmitted

② parity data :-
The bit send along with data bit to detect the error in transmission

Even parity → 1101010110
Odd parity → 1101010100

* Calculate Number of parity bit

3 bits

| P | P | P |
|---|---|---|
| 3 | 2 | 1 |

| Parity | data | Total |
|---|---|---|
| 2 | 1 | 3 |
| 3 | 4 | 7 |
| 4 | 11 | 15 |
| 5 | 20 | 31 |

Error detection in 7 bit hamming Code

| 1 | 0 | 0 | P | 1 | P | P |
|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 |

* What is parity ?

→ parity is a bit in form 0 & 1 use for detect the error in Signal, it Only detect Signal Single bit error not more than Single bit.

Parity bit is a Single bit or extra bit are Send with the Original data. & it tells total no. of 1's present in Signal.

It also tells Total numbers are present in Signal is
1. even
2. odd

+ Transmitter tell to pass the msg to receiver that data is even or odd

1. Even parity → $\underbrace{0\,0\,1\,0}_{\text{Original Signal.}}\ \boxed{1}$ → parity bit

   $\underbrace{0\,1\,1\,0}_{\text{Original Signal}}\ \boxed{0}$ → parity bit

2. Odd parity → $\underbrace{0\,0\,1\,0}_{\text{Original Signal}}\ \boxed{0}$ → odd parity

   $\underbrace{0\,1\,1\,0}_{\text{Original Signal.}}\ \boxed{1}$ → odd parity

* Error detecting Code & error Connecting Code.

→ If single bit a change it will damage System.

→ Error :-
when we Convert analog to digital Code their some changes in bit which will damage & our System is called error.

→ Types of error :-

① Bit error :-
when data is in Sequence & when 1 changed 0 & 0 changed 1 it is called as bit error.

1) Single bit error
2) multiple bit error
3) Burst error.

1) Single bit error.

1 0 1 1 0
↓ changed
1 0 1 0 0

Single bit error Occurs in parallel Communication System because the data is transferred bit wise

2) multiple error

more bit changes is called multiple errors.

ex:- 1 0 1 1 0

1 1 1 0 0 — Two bit changed.

3) Burst error

1 0 1 1 0

10101

When there is two or more bits get exchanged with itself & its position is burst error.

* Error detecting :- It is the process of detecting error which are present in data in a communication system their are some redundancy code to detect this error.

* Types of error:-

1) parity error
2) CRC → Cyclic redundency Code.
4) LRC → longitudnal Redundency Code.
5) Check Sum

i) parity check :

8 bit → Information

Even parity → even 1's
odd 1's

odd parity → even 1's
odd 1's.

:) Even no. of 1 = add parity bit = 0
odd no of 1 = add parity bit = 1