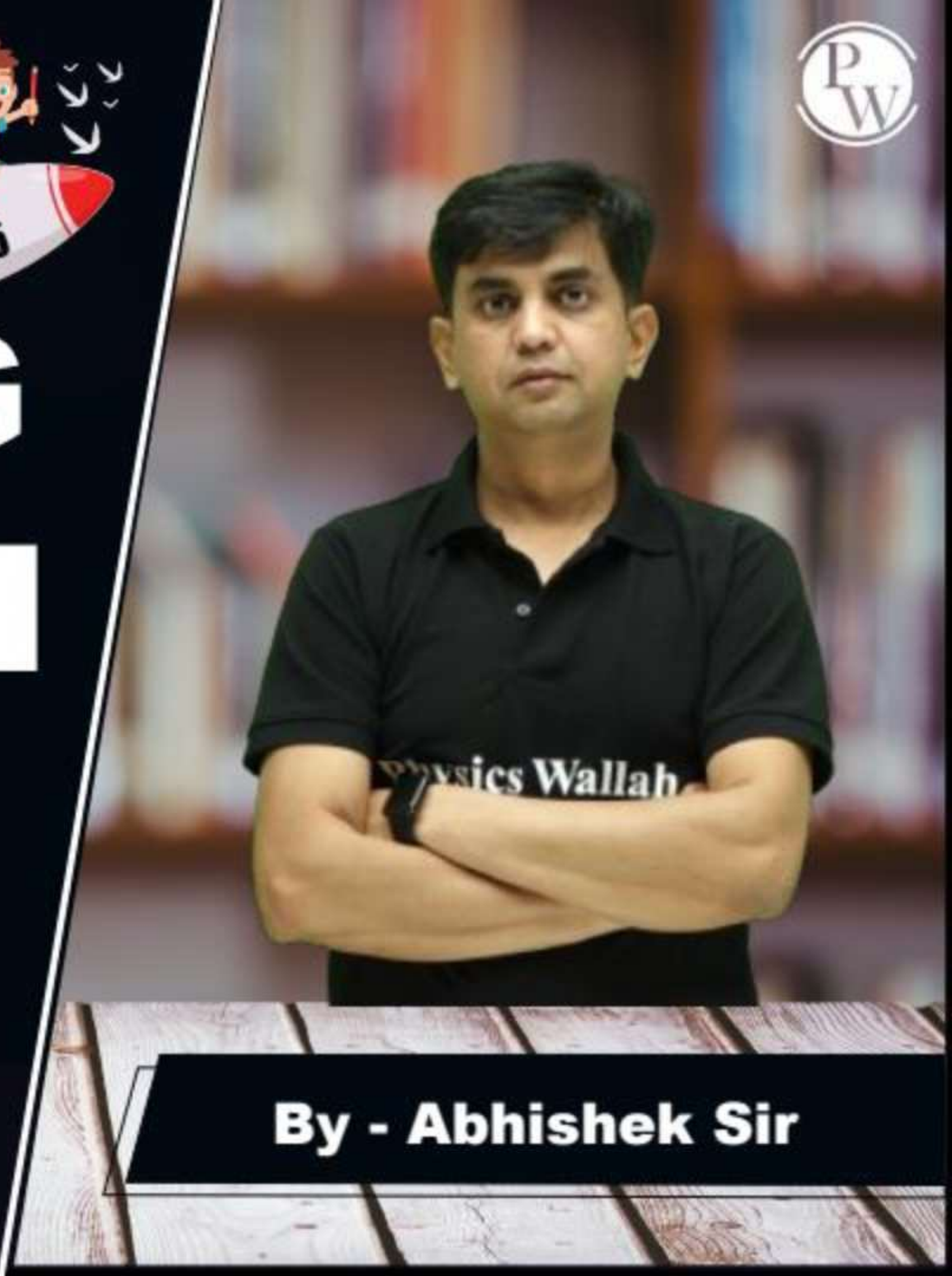
CS & IT ENGINEERING

Computer Network

Error Control

Lecture No. - 01



By - Abhishek Sir

Recap of Previous Lecture

Topic Byte Count

Topic Byte Stuffing

Topic Bit Stuffing

Topics to be Covered

Topic Error Control

Topic One-bit parity

Topic Block Code

ABOUT ME

Hello, I'm Abhishek

- GATE CS AIR - 96
- M.Tech (CS) - IIT Kharagpur
- 12 years of GATE CS teaching experience

Telegram Link : https://t.me/abhisheksirCS_PW

@ABHISHEKSIRCS_PW

#Q. A bit-stuffing based framing protocol uses an 8-bit delimiter pattern of 01111110. If the output bit-string after stuffing is 01111100101, then the input bit-string is

[GATE-2014]

Bit pattern → 01111110

Output string →

01111100101

discard

Input string →

0111110101

(A) 0111110100

(B) 0111110101

(C) 0111111101

(D) 0111111111

Ans: B

01110

#Q. In a data link protocol, the frame delimiter flag is given by 0111. Assuming that bit stuffing is employed, the transmitter sends the data sequence 01110110 as

[GATE-2004]

~~(A)~~ 01101011

~~(B)~~ 011010110

~~(C)~~ 011101100

[✓ (D)] 0110101100

Flag bits → 01110

Input (Frame) →

01110110

Output (transmitted) string →

$$01110\;\underbrace{0110101100}\;01110$$

$$\boxed{\text{Ans: D}}$$

Topic : Error Control

Receiver

(Received data)
1 0 1 1 0 0 1 0

Sender

(Transmitted data)
1 0 1 1 0 0 1 0

→ if "**Received data**" is not same as "**Transmitted data**"
then "**chance of error**"

Topic : Error

Error : Corrupted data [flipped data bits]

Types of error:

1. Single bit error
2. Burst error (Multiple bit error)

Topic : Single bit error

→ Only one bit in the received data has changed.

Transmitted data = 1 0 1 1 0 0 1 0

Received data = 1 0 1 1 1 0 1 0

Topic : Burst Error

→ Multiple bit error

→ More than one [two or more] bit in the received data have changed.

Transmitted data = 1 0 1 1 0 0 1 0

Received data = 1 1 1 1 0 1 0 0

Topic : Burst Error

→ Burst Length =

Length from first corrupted bit to the last corrupted bit [inclusive]

Transmitted data = 1 0 1 1 0 0 1 0

Received data = 1 1 1 1 0 1 0 0

6

→ In case of burst error,

total number of corrupted data bits is less than equal to Burst Length

#Q. Consider ASCII character "A" is transmitted by transmitter, but ASCII character "D" is received by receiver. Calculate burst length?

Transmitted Data = 'A' = 65 = $01000\underline{0}0\underline{1}$

Received Data = 'D' = 68 = $01000\underbrace{\underline{1}0\underline{0}}_{\textcircled{3}}$

Two bit error,

Burst Length = 3

$$\boxed{\text{Ans} = 3}$$

Topic : Error Control

→ Based on Redundant bits
[Parity bits or extra bits]

1. **Error detection** only
2. **Error detection and correction**

Topic : Error detection

→ Can only detect error(s)

→ Not able to correct

→ Retransmission of corrupted data

Two error detection technique :

1. Cyclic Redundancy Check (CRC) ✓
2. Checksum

Topic : Error detection and correction

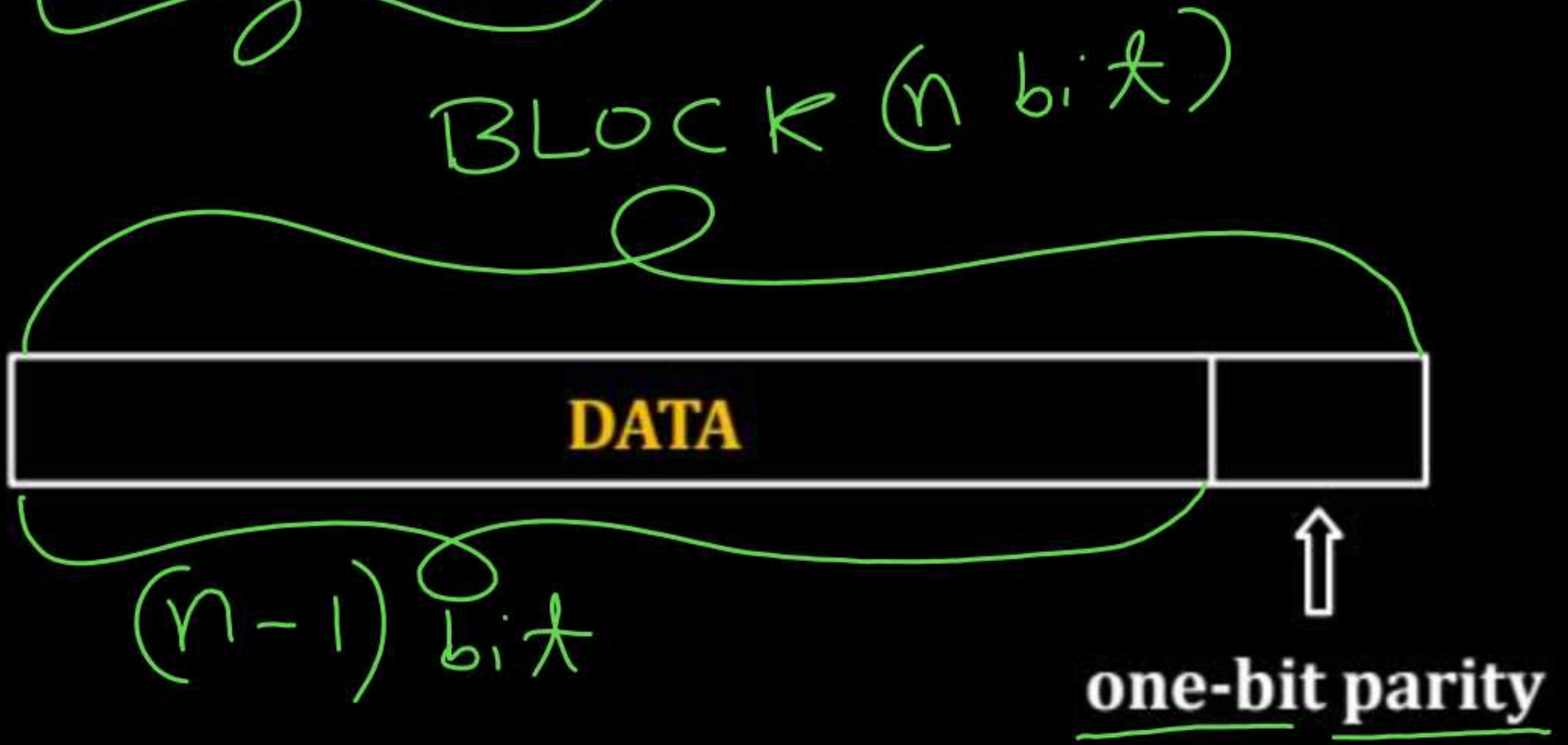
→ Can detect as well as correct error(s)

→ Forward error correction (FEC)

Two error detection and correction technique :

1. 2D Parity (IT-08)
2. Hamming Code (CS-21)

Topic : One-bit parity



DATA

one-bit parity

Topic : One-bit parity

Transmitter protocol :-

1. **Even Parity**

```
if number of one's in the data is even
    than transmitter set parity bit "zero"
else
    set parity bit "one"
```

2. **Odd Parity**

```
if number of one's in the data is odd
    than transmitter set parity bit "zero"
else
    set parity bit "one"
```

Topic : One-bit parity

Suppose "$\underbrace{\text{Even parity}}_{0}$"

$\underline{\text{DATA}}$ = "$\underbrace{1\ 0\ 1\ 1\ 1\ 0\ 1}_{0}$"

Transmitted Data = 1 0 1 1 1 0 1 $\underset{P}{\underline{1}}$

Topic : One-bit parity

Receiver protocol :-

1. **Even Parity**

```
if receiver find number of one's in the received block is even (including parity)
        then receiver concluded "no error detected"
else
        receiver concluded "error detected"
```

2. **Odd Parity**

```
if receiver find number of one's in the received block is odd
            then receiver concluded "no error detected"
else
            receiver concluded "error detected"
```

Topic : One-bit parity

Suppose "Even parity"

CASE I : No any error

DATA = "1 0 1 1 1 0 1"

Transmitted Data = 1 0 1 1 1 0 1 1

Received Data = 1 0 1 1 1 0 1 1

Receiver Concluded : No any error detected, accept the data

Topic : One-bit parity

Suppose "**Even parity**"

CASE II : One-bit error

DATA = "1 0 1 1 1 0 1"

Transmitted Data = $\boxed{1\ 0\ 1\ 1\ 1\ 0\ 1\ \underline{1}}$

Received Data = 1 0 1 0 1 0 1 1

Receiver Concluded : Error detected, reject the data

Topic : One-bit parity

Suppose "**Even parity**"

CASE III : Two-bit error

DATA = "1 0 1 1 1 0 1"

Transmitted Data = 1 0 1 1 1 0 1 1

Received Data = 1 0 0 1 1 1 1 1

Receiver Concluded : No any error detected, accept the data

Topic : One-bit parity

Suppose "**Even parity**"

CASE IV : Three-bit error

DATA = "1 0 1 1 1 0 1"

Transmitted Data = 1 0 1 1 1 0 1 **1**

Received Data = 1 0 1 0 0 1 1 1

Receiver Concluded : Error detected, reject the data

Topic : One-bit parity

→ Receiver detect "**all single bit error**" ✓

→ In case of burst error,
receiver able to detect "**all odd number of errors**"

#Q. Let suppose, even parity is used in one-bit parity error detection technique. If receiver find total 295 one's in the received block (including parity) then what receiver concluded?

~~(A)~~ No any error detected

[✓ (B)] Error detected

~~(C)~~ Unable to detect error

~~(D)~~ Data insufficient

Ans : B

Topic : One-bit parity

Suppose "Even parity"

DATA = "1 0 1 1 1 0 1"

d_1 d_2 d_3 d_4 d_5 d_6 d_7

Transmitted Data = 1 0 1 1 1 0 1 1

P

X-OR Gate

$0 \oplus 0 = 0$

$0 \oplus 1 = 1$

$1 \oplus 0 = 1$

$1 \oplus 1 = 0$

Topic : One-bit parity

Suppose "**Even parity**"

$d_1 = 1$

$d_2 = 0$

$d_3 = 1$

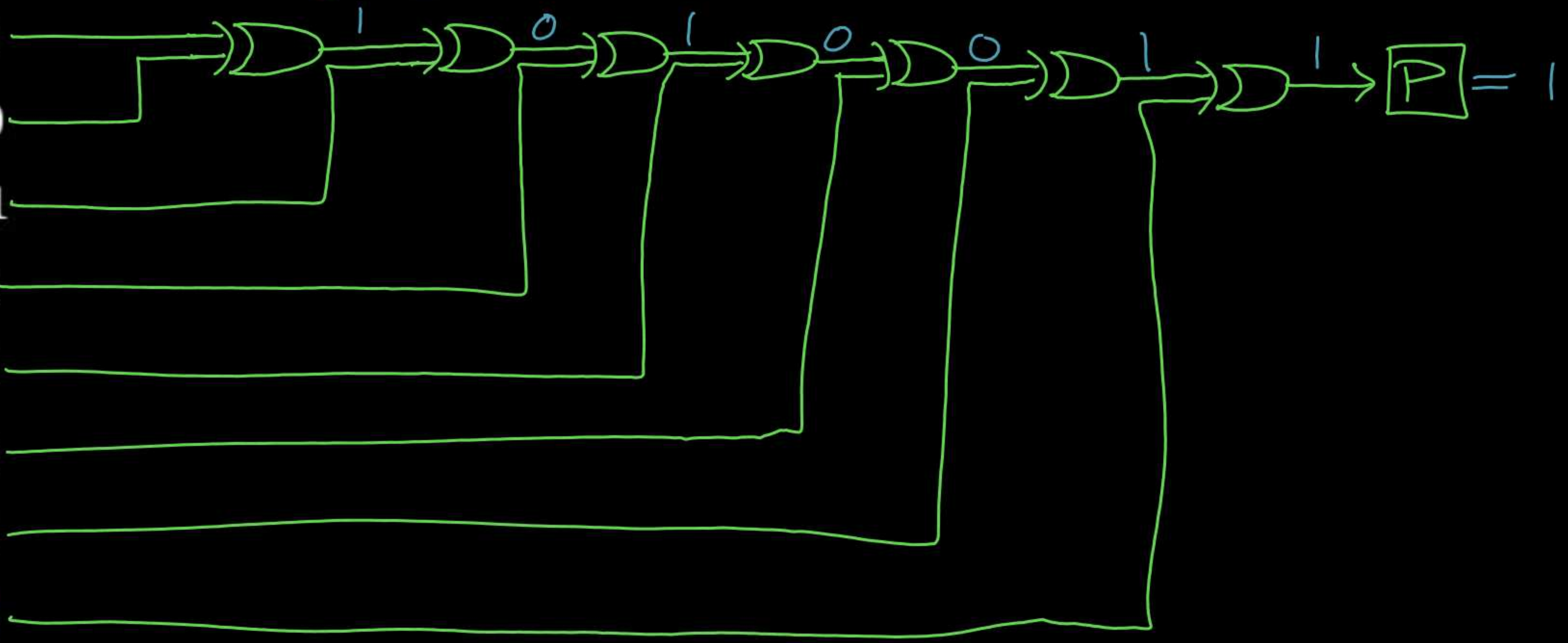
$d_4 = 1$

$d_5 = 1$

$d_6 = 0$

$d_7 = 1$

P = 0



2 mins Summary

Topic Error Control

Topic One-bit Parity

Topic ~~Block code~~

THANK - YOU