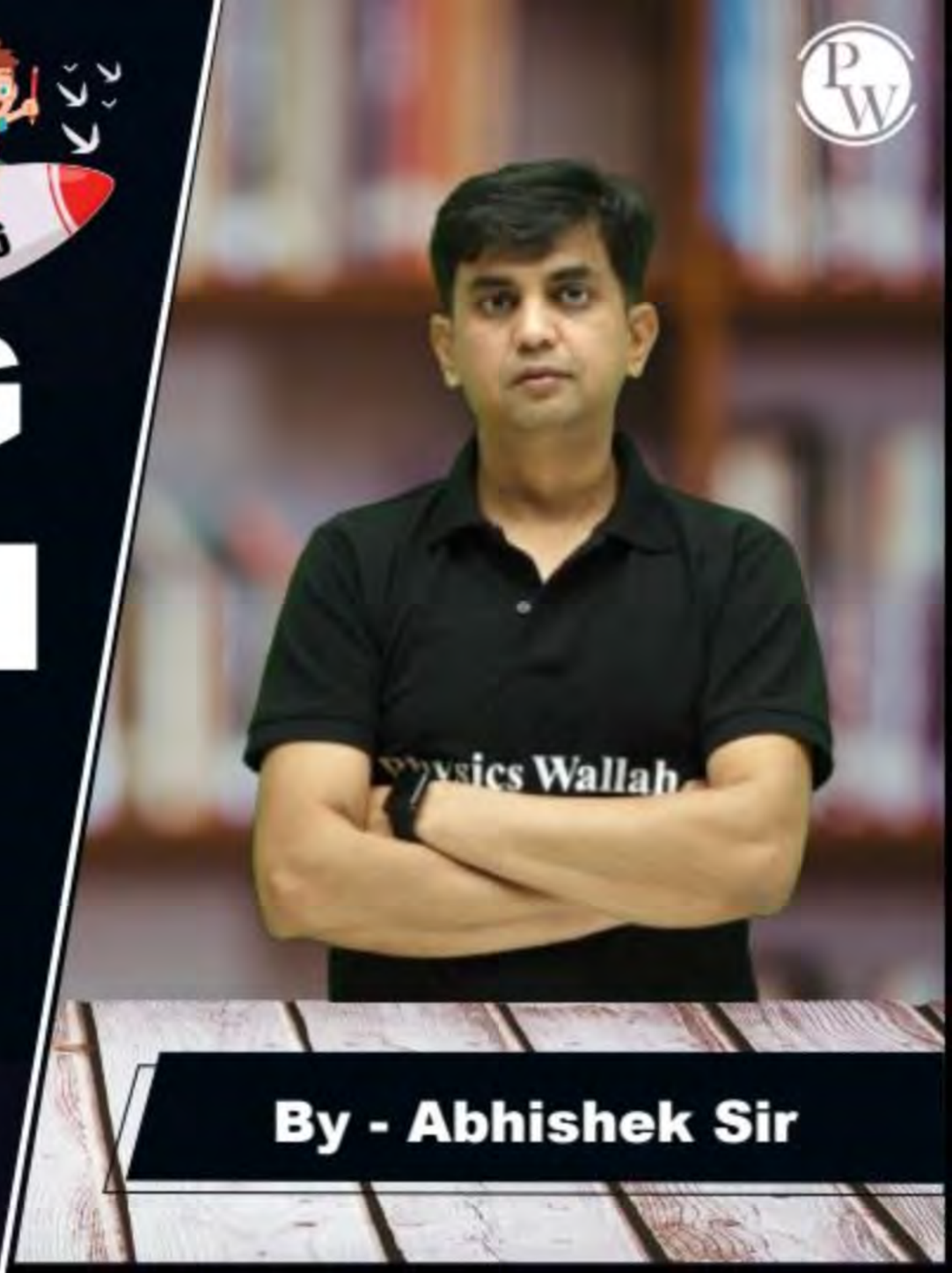
CS & IT ENGINEERING

Computer Network

Error Control

Lecture No. - 02



By - Abhishek Sir

Recap of Previous Lecture

Topic Packet Switching

Topic Error Control

Topics to be Covered

- Topic: One-bit parity
- Topic: Valid & Invalid Codewords
- Topic: CRC

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

#Q. Which one of the following statements is FALSE?

[GATE 2004, 1-Mark]

(A) Packet switching leads to better utilization of bandwidth resources than circuit switching TRUE

(B) Packet switching results in less variation in delay than circuit switching FALSE

(C) Packet switching requires more per-packet processing than circuit switching TRUE

(D) Packet switching can lead to reordering unlike in circuit switching TRUE

Ans: B

Topic : One-bit parity

⇒ Single bit parity

⇒ VRC

(N bit) Block/code

(N-1) bit

one-bit parity

*Even 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 "Even parity"

DATA = "1 0 1 1 1 0 1" (7 bit Data)

Transmitted Data = 1 0 1 1 1 0 1 1

parity

Topic : One-bit parity

Receiver protocol :-

1. **Even Parity**

   if receiver find number of one's in the received block is even

   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 ⟶ Always Detect.

DATA = "1 0 1 1 1 0 1"

Transmitted Data	= 1 0 1 1 1 0 1 1
Received Data	= 1 0 1 0 1 0 1 1

even ⟶ odd

6 one's ⟶ one bit error
- a) 5 one's
- b) 7 one's

Receiver Concluded : Error detected, reject the data

Topic : One-bit parity

Suppose "**Even parity**"

CASE III : Two-bit error ⟶ Never detected.

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

even ⟶ even

6 one's ⟶ Two bit error

a) 4 one's
b) 6 one's
c) 8 one's

Receiver Concluded : No any error detected, accept the data

Topic : One-bit parity

Suppose "Even parity"

CASE IV : Three-bit error ⟶ Always detected.

DATA = "1 0 1 1 1 0 1"

even ⟶ odd

Transmitted Data	= 1 0 1 1 1 0 1 1
Received Data •	= 1 0 1 0 0 1 1 1

6 one's ⟶ 3 bit errors
a) 3 one's
b)

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**"

[Count of Erroneous bits is odd]

#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 : Modulo-2 Arithmetic

*bit wise XOR

→ Binary arithmetic or Boolean algebra

→ Addition and subtraction are the same operation

→ Operations :

Addition / Subtraction :

$0 + 0 = 0 \bmod 2 = 0$

$0 + 1 = 1 \bmod 2 = 1$

$1 + 0 = 1 \bmod 2 = 1$

$1 + 1 = 2 \bmod 2 = 0$

$0 \oplus 0 = 0$

$0 \oplus 1 = 1$

$1 \oplus 0 = 1$

$1 \oplus 1 = 0$

$0 - 0 = 0$

$0 - 1 = 1$

$1 - 0 = 1$

$1 - 1 = 0$

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

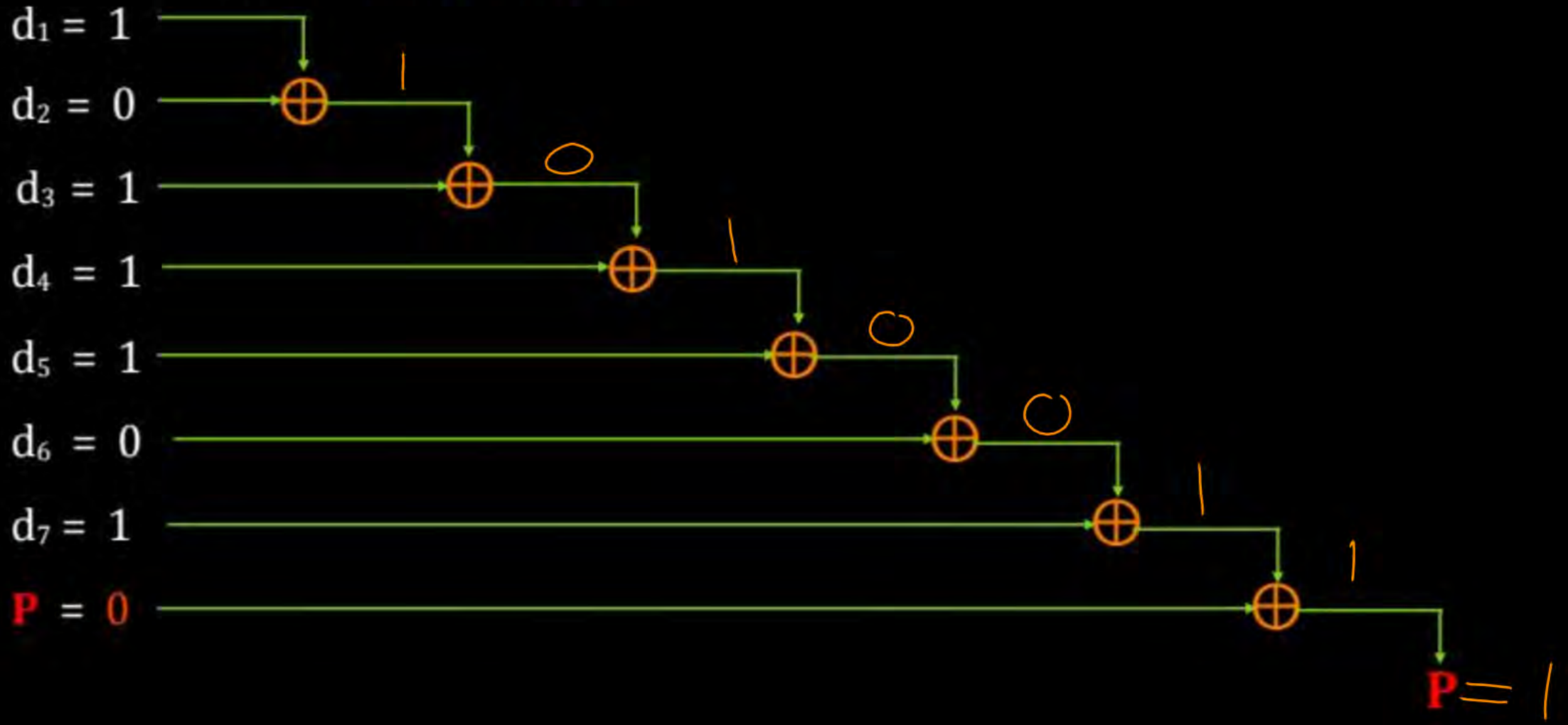
Transmitted Data = 1 0 1 1 1 0 1 **1**

P

Topic : One-bit parity

⊕ → X-OR gate

Suppose "**Even parity**"



(Even Parity)

AT Sender (Transmitter)

Input = " 1 0 1 1 1 0 1 **0** "

d_1 d_2 d_3 d_4 d_5 d_6 d_7 **P**

0	$d_1 = 1$
$d_1 = 1$	$d_2 = 0$
1	$d_3 = 1$
0	$d_4 = 1$
1	$d_5 = 1$
0	$d_6 = 0$
0	$d_7 = 1$
1	**P** = **0**

P = 1

$a \oplus 0 = a$

(Even parity)

AT Receiver

Input = " 1 0 1 1 1 0 1 **1** "

d_1 d_2 d_3 d_4 d_5 d_6 d_7 **P**

0	$d_1 = 1$
$d_1 = 1$	$d_2 = 0$
1	$d_3 = 1$
0	$d_4 = 1$
1	$d_5 = 1$
0	$d_6 = 0$
0	$d_7 = 1$
1	**P** = **1**

Result = 0

if **Result** == **ZERO** :
then Receiver concluded
"**No any error detected**"
else
Receiver concluded
"**Error detected**"

Topic : Block Code

Data (m bits) → Codeword (N bits)

Codeword = [Data bits | Parity bits]

Number of Parity bits = (N – m)

→ 2^m codewords of length N.

Topic : Block Code

$m = 3$, $N = 4$

One-bit parity (with even parity) and 3 data bits

Data	-->	Codeword
$d_1 d_2 d_3$	-->	$d_1 d_2 d_3 P$
0 0 0	-->	0 0 0 0
0 0 1	-->	0 0 1 1
0 1 0	-->	0 1 0 1
0 1 1	-->	0 1 1 0
1 0 0	-->	1 0 0 1
1 0 1	-->	1 0 1 0
1 1 0	-->	1 1 0 0
1 1 1	-->	1 1 1 1

$(2^m = 2^3)$

2^3 codewords (Valid)

Topic : Valid Codewords vs Invalid Codewords [Even Parity]

→ Valid codewords :

0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111

50%

→ Invalid codewords :

0001, 0010, 0100, 0111, 1000, 1011, 1101, 1110

50%

Topic : Valid Codewords vs Invalid Codewords

→ Transmitter always transmits valid codewords in the channel.

→ A codeword received by receiver from the channel :

if received codeword is a valid codeword

then receiver concluded "No any error detected"

else [received codeword is a invalid codeword]

then receiver concluded "Error detected"

Topic : Linear Code

→ Linear combination of codewords

C_i : 0 1 0 1 +
C_j : 0 1 1 0
C_k : 0 0 1 1

if C_i and C_j are codewords
then C_k is also be a codeword
where $C_k = C_i + C_j$

⇒ one-bit parity (in case of even parity) always produces "Linear Code".

Topic : Cyclic Code

* one bit parity always produces cyclic code.

→ Cyclic combination of linear codewords

if C is codeword
then bit-wise cyclic left or right shift on C is also be a codeword

$C = \boxed{d_1 d_2 d_3 P} \quad \boxed{P d_1 d_2 d_3}$

$\boxed{d_2 d_3 P d_1}$

Topic : CRC

→ Cyclic Redundancy Check (CRC)

→ Error detection technique

Topic : Polynomial Function

→ Polynomial function, where coefficients are either zero or one.

Example :

Function = $X^5 + X^2 + 1$

$= 1*X^5 + 0*X^4 + 0*X^3 + 1*X^2 + 0*X^1 + 1*X^0$

Topic : Generator Polynomial

G(X) : Generator Polynomial function

Divisor : binary string

Example 1 :-

$G(X) = X^3 + X^2 + 1$

$= 1*X^3 + 1*X^2 + 0*X^1 + 1*X^0$

Divisor = 1 1 0 1

$G(x) = x^5 + x^3 + 1$

Divisor = 101001

degree(G(x)) = n

No. of bits in divisor = (n+1)

Example 1 :-

$G(X) = X^3 + X^2 + 1$

degree[G(x)] = 3

DATA = 10011010

CRC = ? = 101

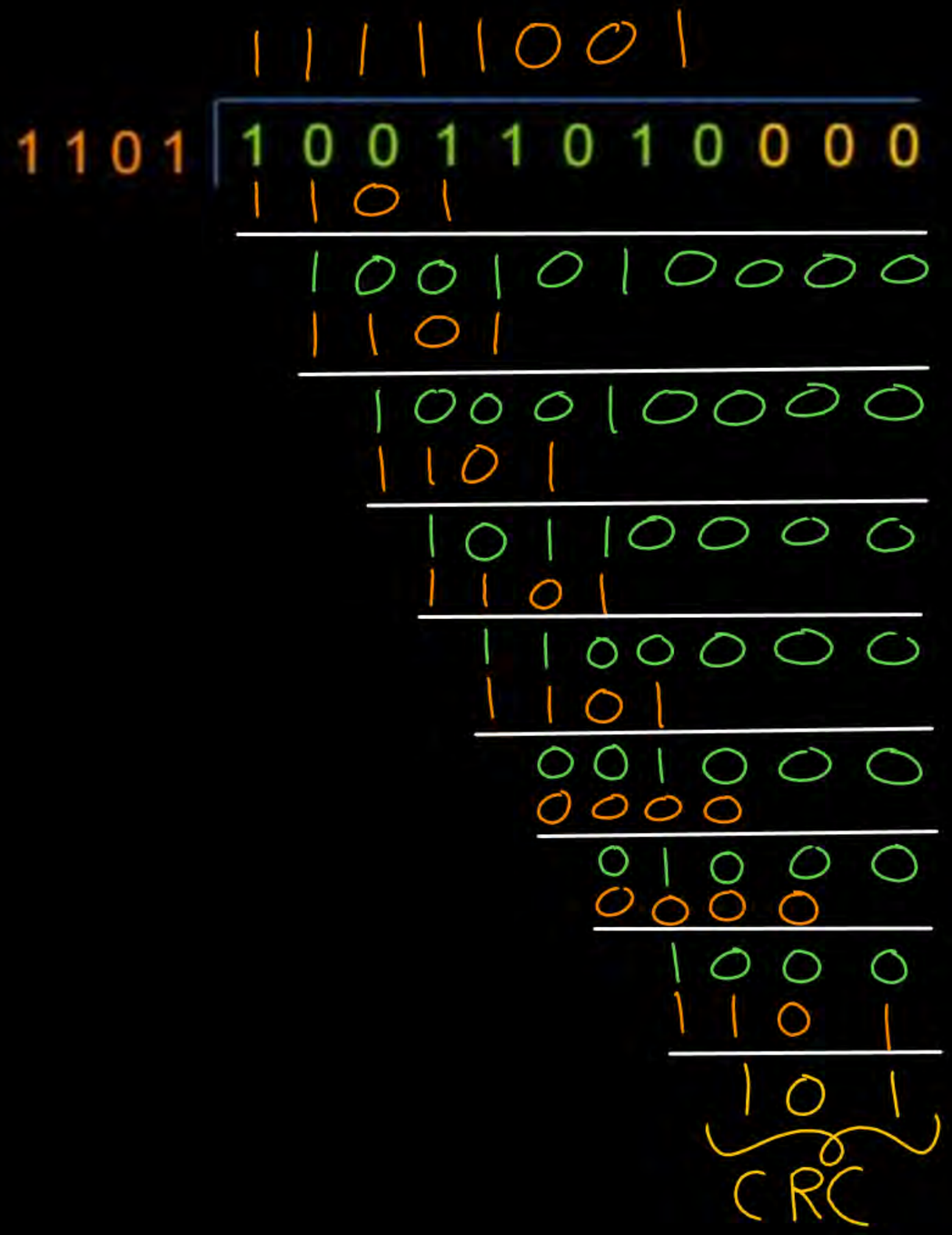
Solution :

DIVISOR = 1101

DIVIDENT = 10011010000

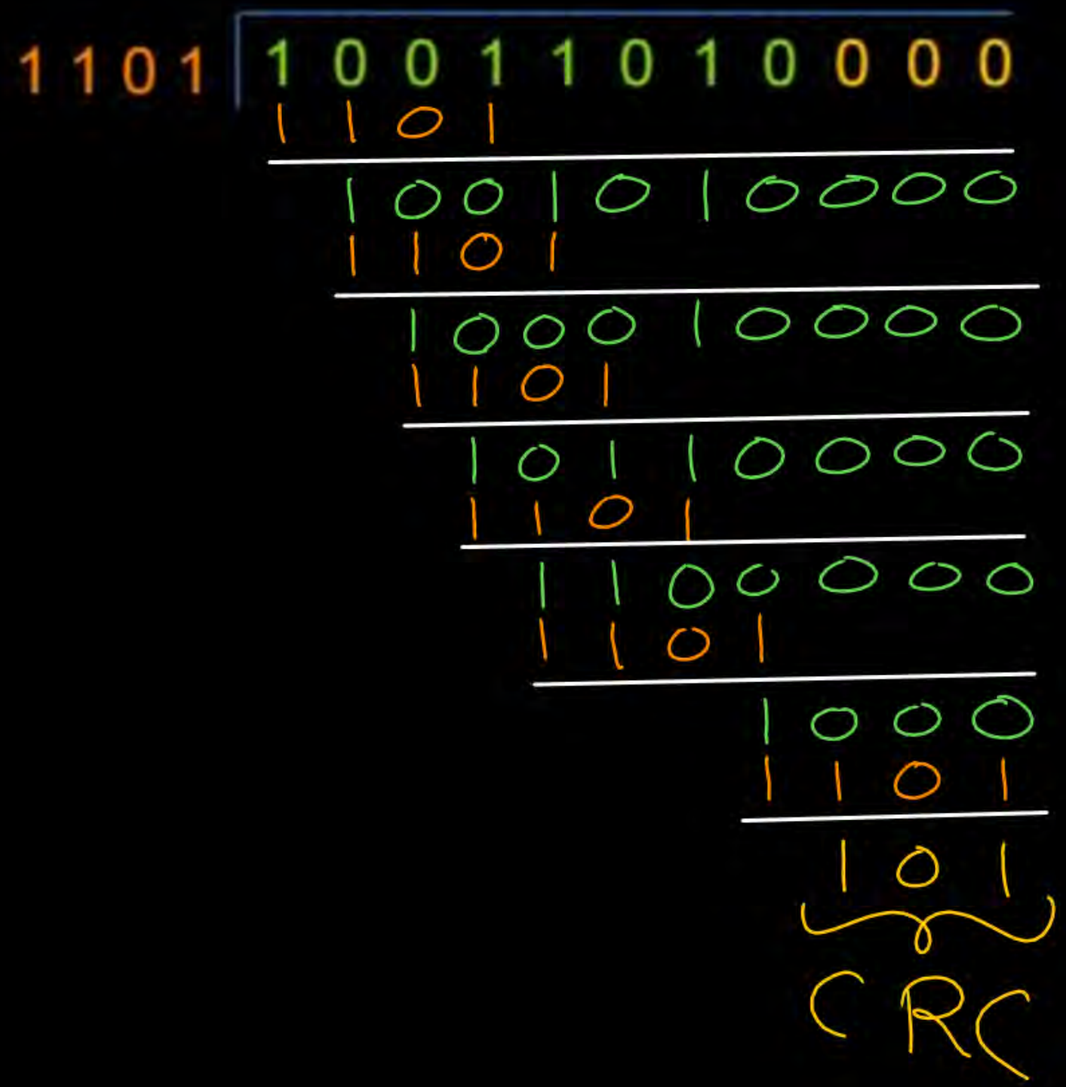
DATA (8bit) 3 zeros

Modulo 2 Division
[bit-wise X-OR]



Modulo 2 Division
[bit-wise X-OR]

Modulo 2 Division
[bit-wise X-OR]



Modulo 2 Division
[bit-wise X-OR]

Example 2 :- (H.W.)

$G(X) = X^3 + X + 1$

DATA = 1 0 0 1 1 1 0 1

CRC = ?

Solution :

DIVISOR =

DIVIDEND = 1 0 0 1 1 1 0 1

2 mins Summary

Topic One-bit parity

Topic Valid & Invalid Codewords

Topic CRC

THANK - YOU