Computer Network

Data-Link Layer

Lecture : 8

Gaurav Raj

TCP/IP

TCP/IP Layer	Hardware	Software/Protocols
Application	None	HTTP, FTP, SMTP, POP3, IMAP, DNS, SSH
Transport	None	TCP, UDP
Internet	Routers	IP (IPv4/v6), ICMP, IGMP, ARP, RARP Routing(DVR(RIP), LSR(OSPF), BGP)
Data Link	Switches, Bridges, NICs	Ethernet (MAC framing), Wi-Fi (802.11 MAC), PPP, Frame Relay, HDLC
Physical	Cables (fiber, coaxial, twisted pair), Hubs, Repeaters, Connectors (RJ-45), Amplifier	ONLY physical standards (IEEE 802.3 for wiring, IEEE 802.11 PHY for Wi-Fi)

Data-Link Layer

Responsibility
Framing
Error Detection
Error Recovery
Flow Control
Access Control
Addressing
Link Management
Framing and Encapsulation

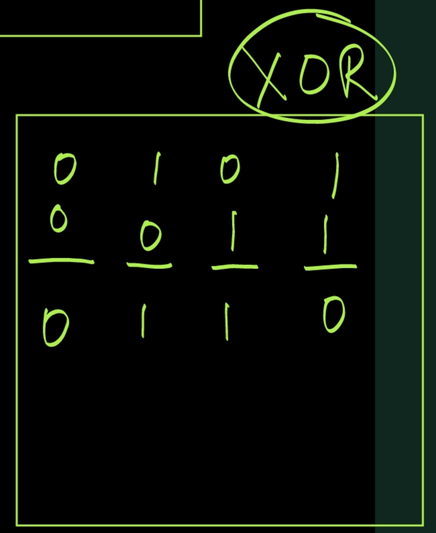
Cyclic Redundancy Check : Error Control

Term	Meaning
Data/Message	Binary string to be transmitted
Generator (G)	A predetermined binary number (like a polynomial)
Divisor	Same as generator
CRC or Remainder	The extra bits added to message for error detection
Transmitted Frame	Message + CRC

Cyclic Redundancy Check : Error Control

Conceptual Steps in CRC

1. **Append (n−1) Zeros** to the data, where n = length of the generator.
2. **Divide** the new data by the generator using **modulo-2 division** (XOR instead of subtraction).
3. The **remainder** becomes the **CRC**.
4. The sender **appends** this CRC to the original data.
5. The receiver **divides** the received data (message + CRC) by the same generator.
    1. If remainder = 0 ⇒ **No Error**
    2. Else ⇒ **Error Detected**



9 = 1001
9 = 1001

0

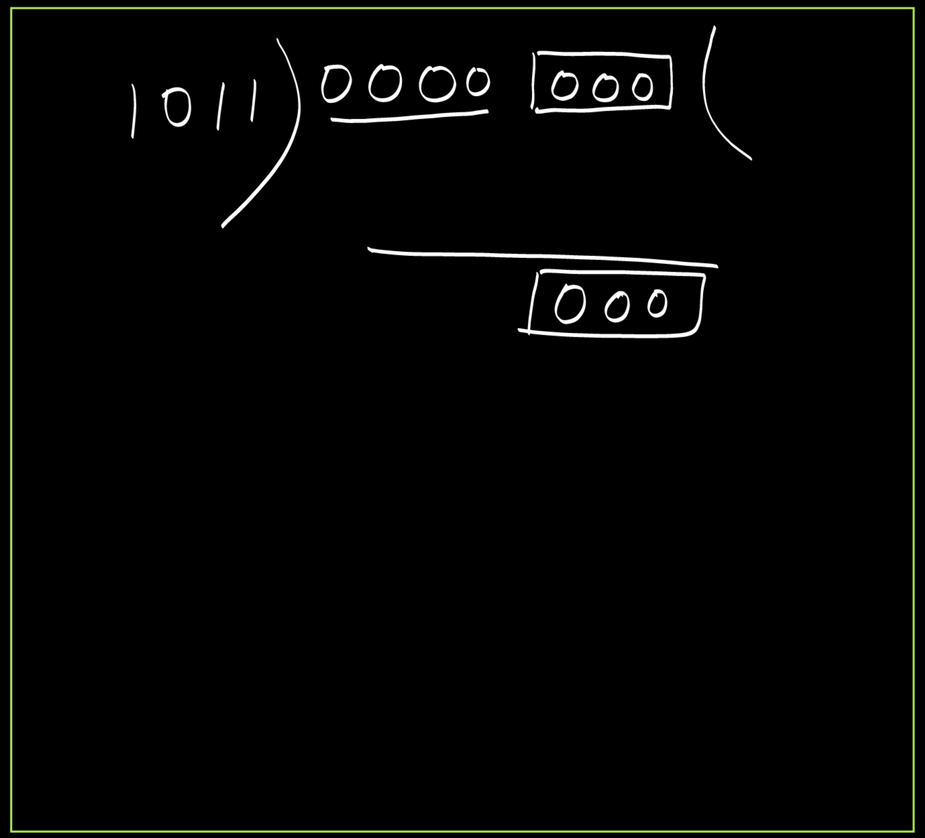
1	2	3	4	5	5	4	2

?

Cyclic Redundancy Check : Error Control

Message (M): 0000 (All 4-bit)

→ Generator: **G(x) = 1011 ($x^3 + x + 1$)**



Modulo 2

→ $x^2 + x + 1$

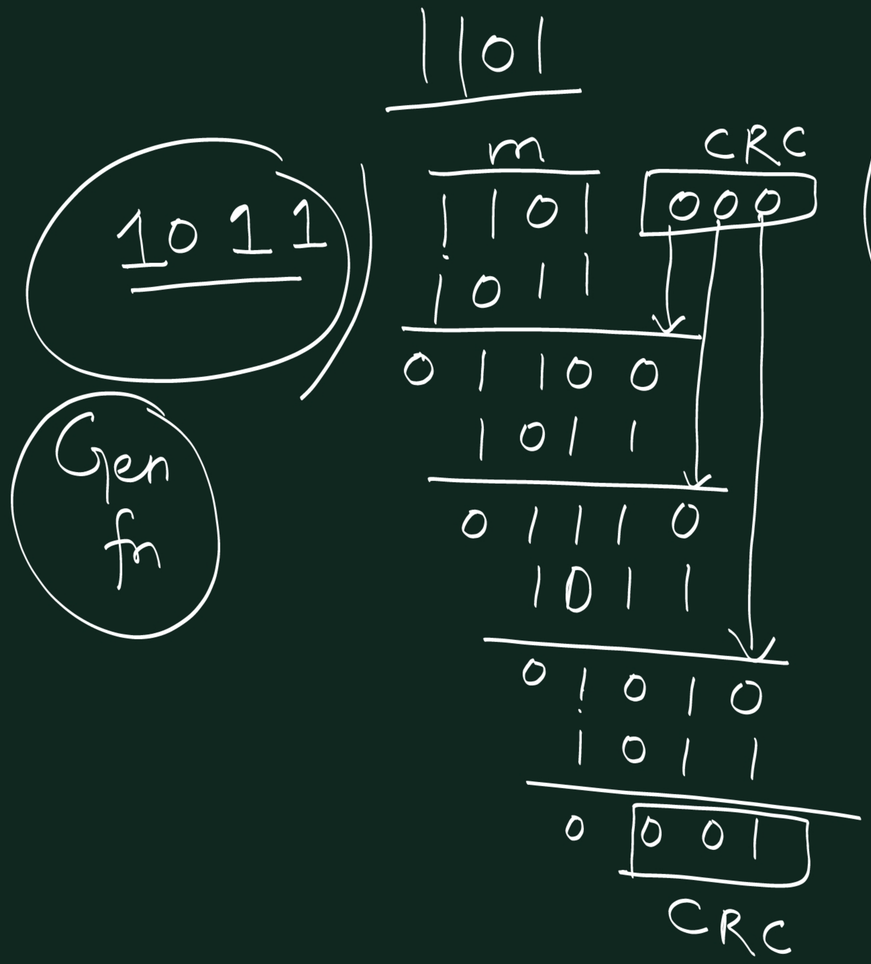
1 1 1

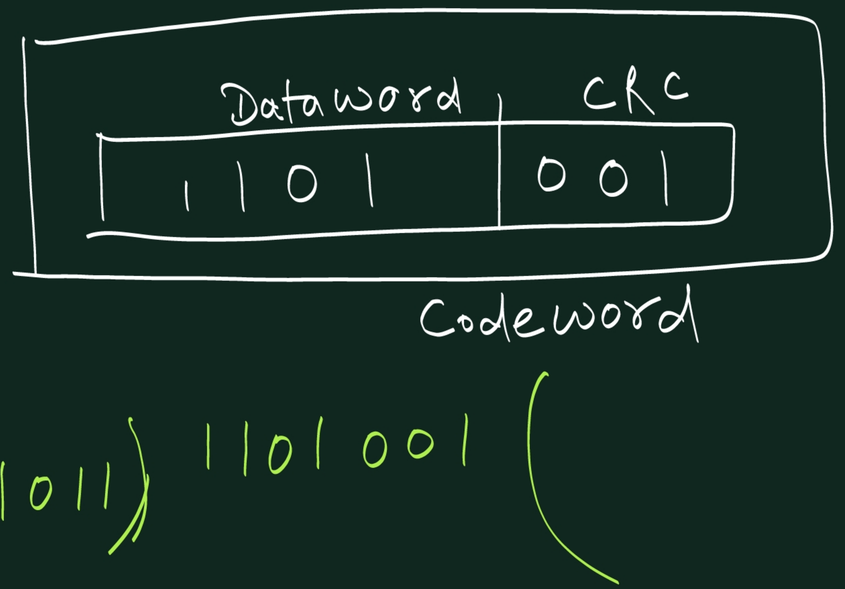
→ $x^3 + x^2 + 1$

1 1 0 1

→ $x^3 + 1$

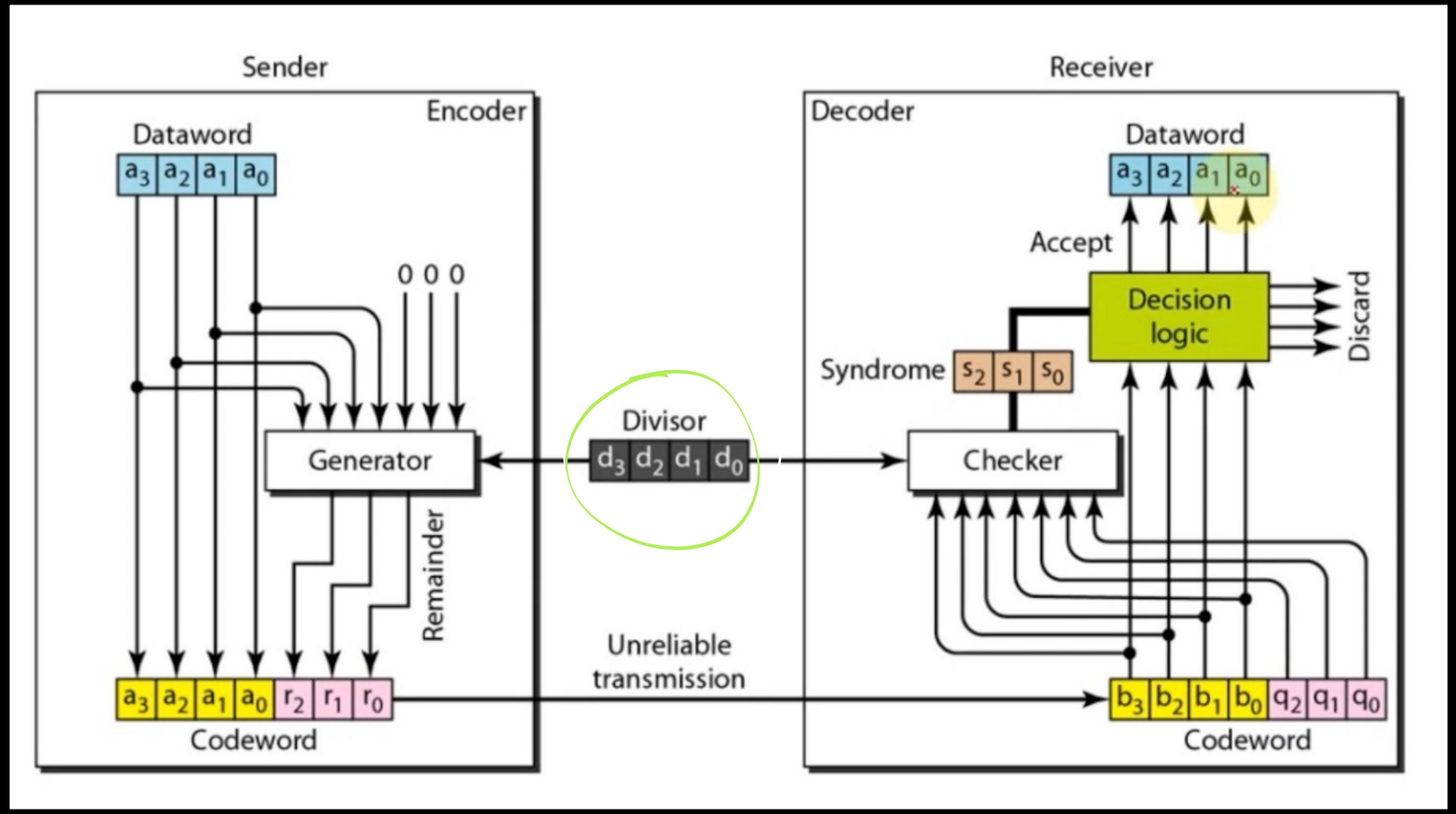
1 0 0 1





1011) 1101001 (

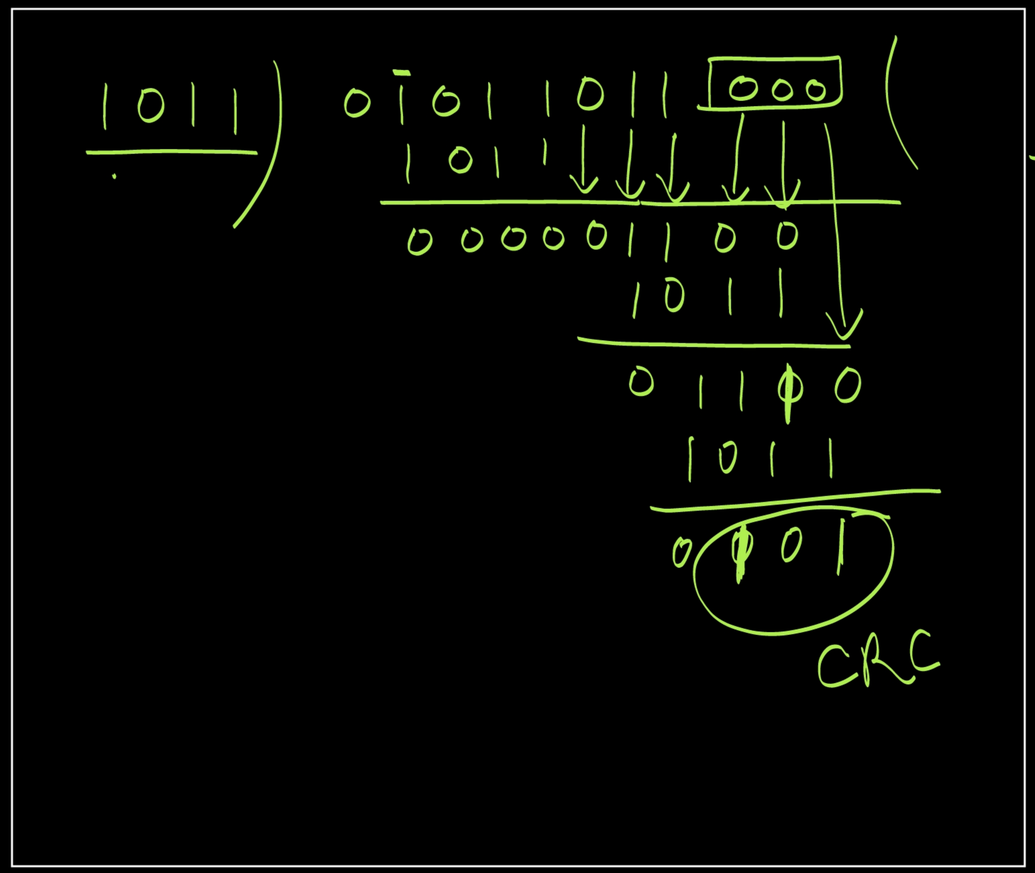
Cyclic Redundancy Check : Error Control

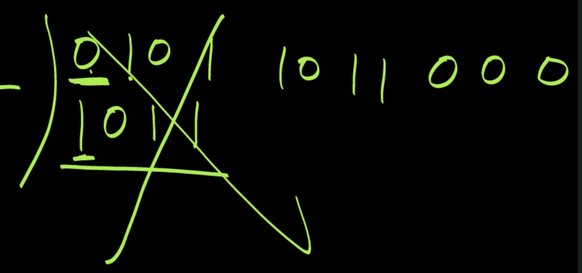


1. A computer network uses polynomials over GF(2) for error checking with 8 bits as information bits and uses x^3+x+1 as the generator polynomial to generate the check bits. In this network, the message 01011011 is transmitted as

GATE:2017

A. 01011011010
B. 01011011011
C. 01011011101
D. 01011011100





Cyclic Generator: **G(x) = 1011 ($x^3 + x + 1$)**

(4)

Dataword	Codeword	Dataword	Codeword
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111

m = 4-bit

7 7

Let's Test G(x)=x3+x+1=1011.
G(x) | (x^n+1) This is equivalent to saying:

We want to find the **smallest value of n** such that:
What is the **order** of G(x) in GF(2)?

Let's compute powers of x mod G(x)
We reduce each **x^k modulo G(x)** and see when we first get 1 again. We'll use polynomial division mod 2:

k	X^k mod G(x)
1	x
2	x^2
3	X^3 mod G(x) = x + 1
4	x(x+1) = x^2 + x
5	x(x^2 + x) = x^3 + x^2 ≡ (x + 1) + x^2= x^2 + x + 1.
6	x·(x^2+x+1) = x^3+x^2+x ≡ (x+1)+x^2+x = x^2+1
7	X^7 = x·(x^2 + 1) = x^3 + x ≡ (x+1) + x = 1
8	X^7 mod G(x) =1

(7, 4)

7
Codeword length

So, the **order of x mod G(x) is 7**

Yes, G(x)=x^3+x+1 is a **cyclic generator** for a **(7, 4)** cyclic code, because it divides x^7+1 over GF(2).

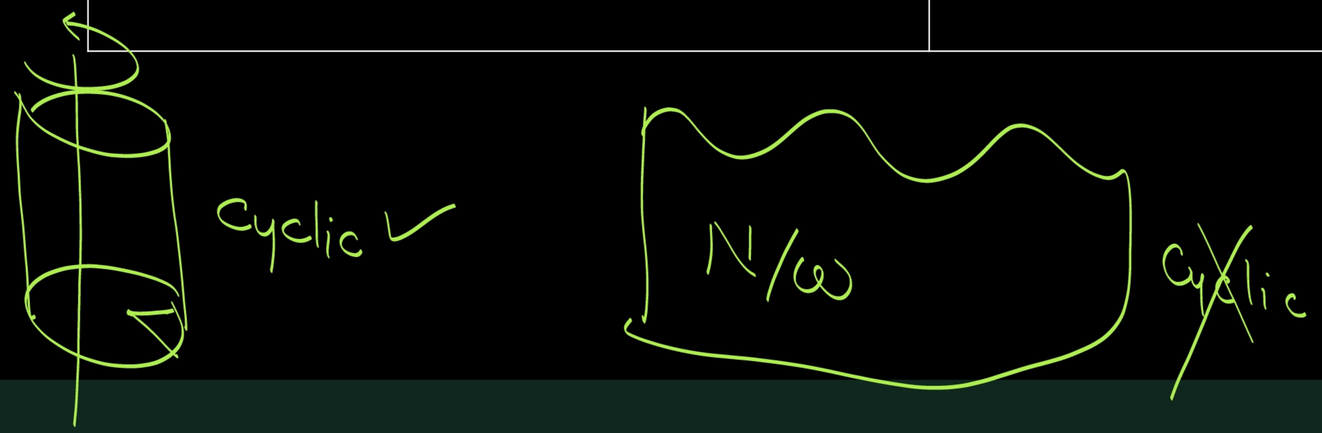
Cyclic Redundancy Check : Error Control

Type of Change	Detected by CRC?	Why?
Single-bit flip	Yes	Changes polynomial, changes remainder
Multiple random bit flips	Yes (usually)	Remainder likely changes
Burst errors (small)	Yes	Covered by degree of generator
Cyclic rotation of bits	No	Rotated version still divisible by generator
Cleverly crafted errors	No	If difference is a multiple of generator

CRCs are **not cryptographic hashes**. They're **fast error-checking tools**, not meant for security or uniqueness.

When You SHOULD Use a Cyclic Generator (i.e., Generator that makes codewords cyclic)

Use Case	Reason
You want closed behaviour under bit rotation	Good for applications where cyclic shifts might occur (like rotating disk errors).
Better error detection	Some cyclic codes (like Hamming, BCH, Reed-Solomon) have **strong error detection and correction** capabilities.



When You Should AVOID Using a Cyclic Generator (e.g., in typical CRC)

Use Case	Reason
Standard CRC for error detection in packets/files	CRC is **not meant** to be closed under rotation. In fact, cyclic shifts are often **not detectable** by CRC.
Security or anti-manipulation use	If rotation creates another valid codeword, **CRC fails to detect it**, which is dangerous.

Technique	Error Detection Capability	Complexity
Parity Bit	Detect single-bit error	Low
Checksum	Detects multiple errors but weak	Low
CRC	Detects burst errors(many bits)	Moderate

1. Consider the generator polynomial $G(x)=x^3+x+1$. A message $M(x)=11000$ (i.e., 5 bits) is to be transmitted using CRC. Determine the CRC check bits (remainder) and the final transmitted bit stream.

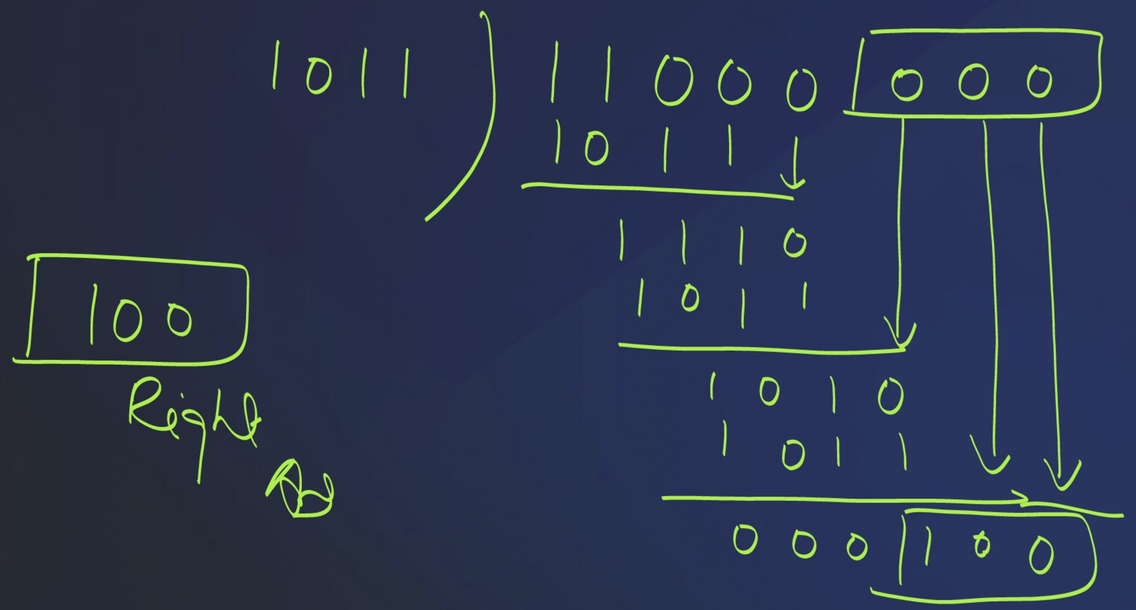
[MCQ]

A 001

B 110

C 111

D 101

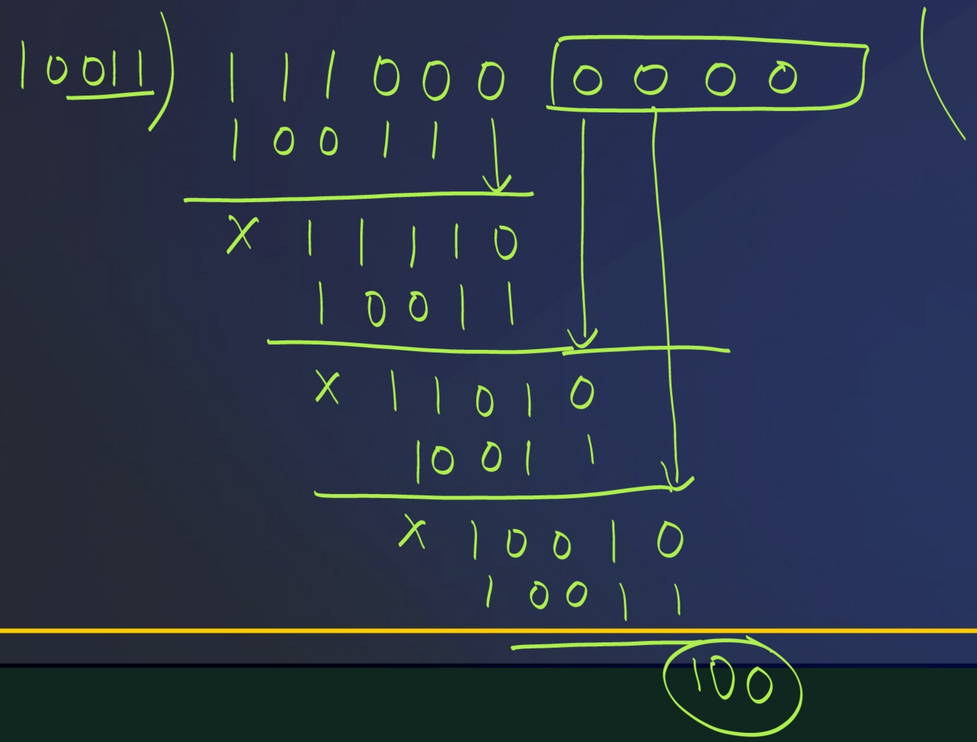


2. A CRC scheme uses generator x^4+x+1 (i.e., 10011). If the message is 111000, what are the CRC bits and the transmitted data?

[MCQ]

100 Ans

A 1001 B 1011 C 0001 D 1101



3. A sender wants to transmit the following three 8-bit binary data words using checksum-based error detection:

Word 1: 11001001
Word 2: 01101101
Word 3: 10011011

[NAT]

Compute the 8-bit checksum that should be sent by the sender.

4. At the receiver's end, the received words are:

Word 1: 11000001 **[NAT]**
Word 2: 01101101
Word 3: 10011011
Checksum: _____

Fill in the correct checksum value from QN:5 and verify if any error is detected at the receiver side.

5. A 2-D even parity scheme is used to detect errors in a data block consisting of **4 rows × 7 bits**. The sender constructs the 2-D parity matrix by first adding one **row parity bit** at the end of each row and then one **column parity bit** at the bottom of each column, including the parity bits.
The resulting **5×8 matrix** is sent over a noisy channel.
At the receiver's end, the received matrix is as follows: **[NAT]**

Row 1: 1 0 1 1 0 1 0 | 0
Row 2: 0 1 0 1 1 1 0 | 0
Row 3: 1 1 1 0 1 0 1 | 1
Row 4: 0 0 0 1 0 0 1 | 0
---
0 0 0 1 0 0 0 | 1

(a) Has any error occurred during transmission?
(b) If yes, identify the bit position (row, column) where the single-bit error occurred.

Thank You