

Chapter-10

Error Detection and Correction

Prepared by-

Shyla Afroge & S. M. Mahedy Hasan

Assistant Professor, Dept. of CSE

RUET, Rajshahi-6204, Bangladesh

Error

- ❖ Data are transmitted in the network.
- ❖ Data can be corrupted during transmission.
- ❖ This is called transmission error.
- ❖ For reliable communications, errors must be detected and corrected.

Types of Errors

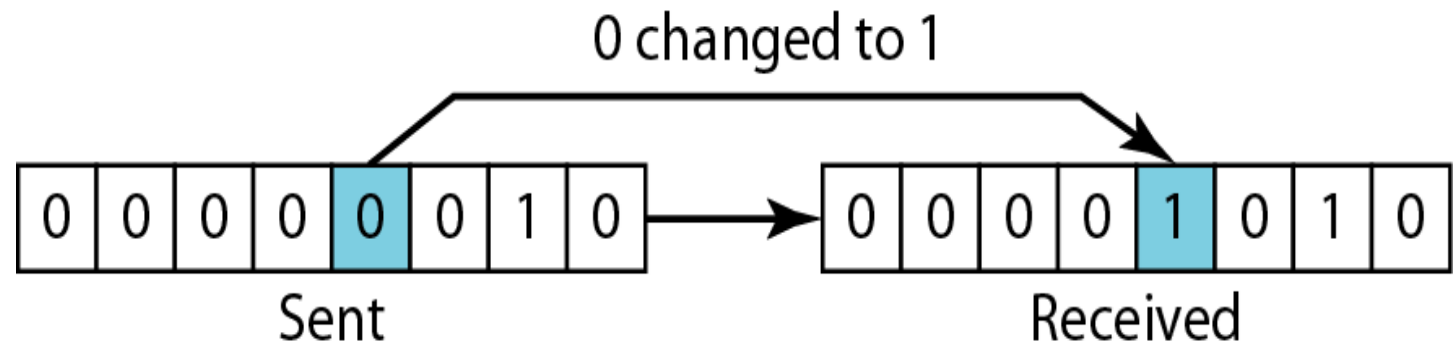
1. Bit Error
2. Burst Error

How To Detect Errors?

- ❖ Error detection means to decide whether the received data is correct or not without having a copy of the original message.
- ❖ To detect or correct errors, we need to send extra (redundant) bits with data.
- ❖ These extra bits are called **redundant bits**.

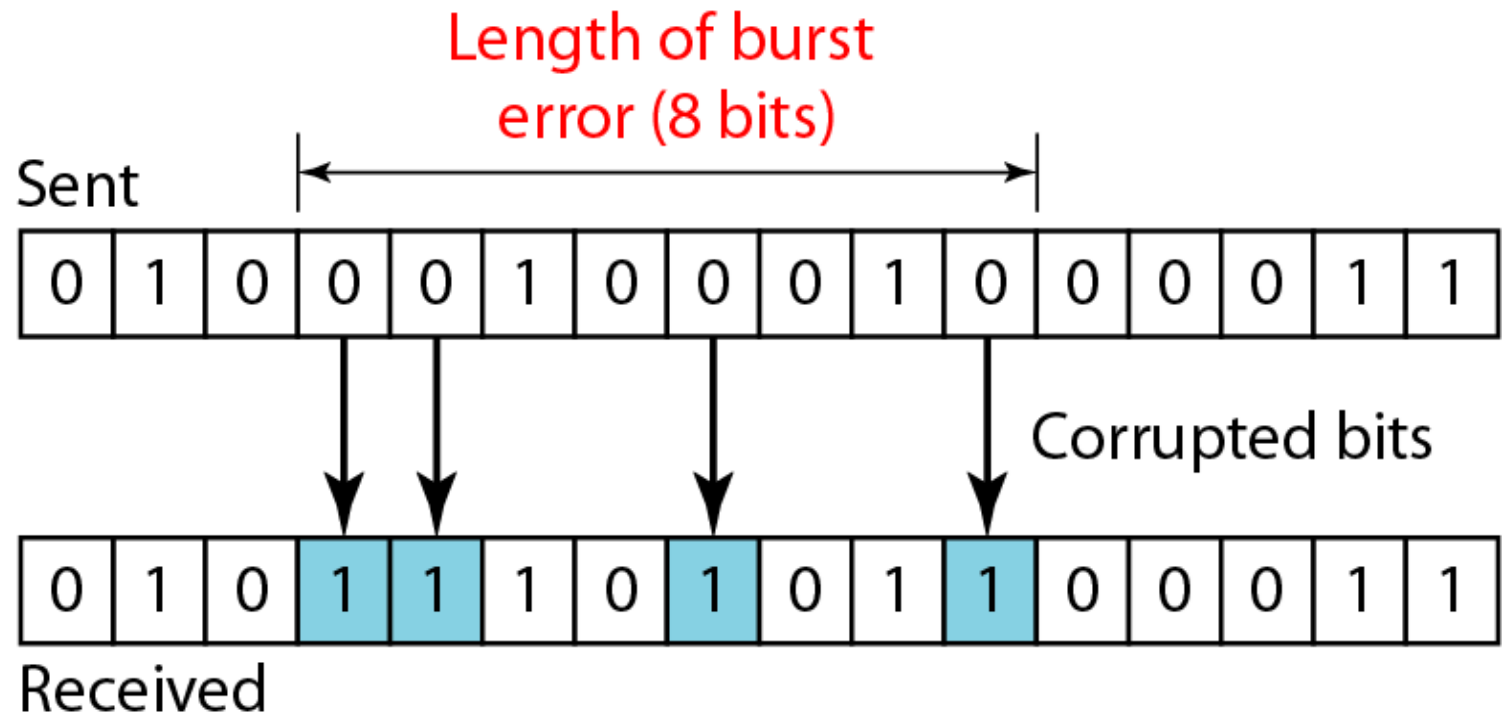
Bit Error

- ❖ In a single-bit error, only 1 bit in the data unit has changed.



Burst Error

- ❖ A burst error means that 2 or more bits in the data unit have changed.



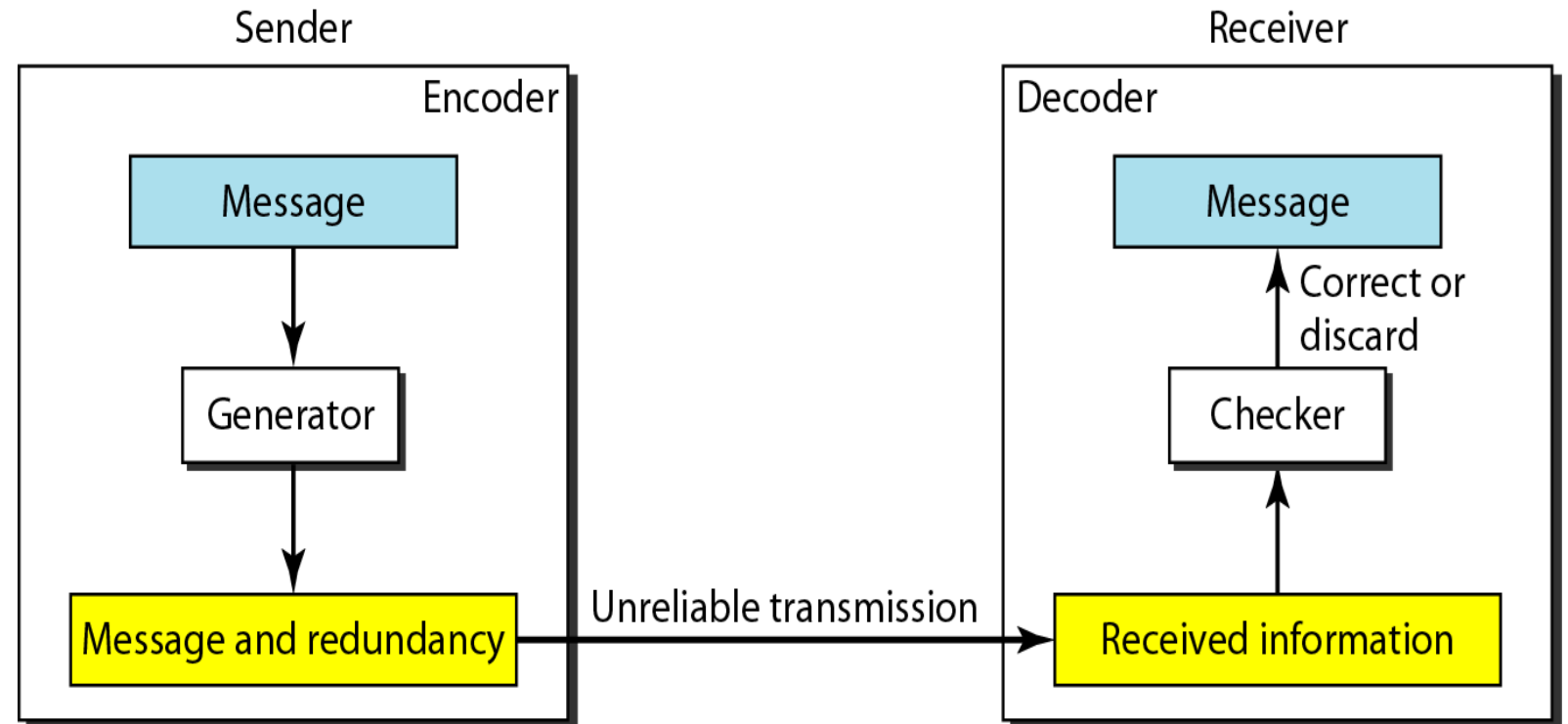
Error Correction

❖ It can be handled in two ways:

1. Forward error correction is the process in which the receiver tries to guess the message by using redundant bits.
2. Correction by retransmission is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message.

Coding

- ❖ The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits.
- ❖ The receiver checks the relationships between the two sets of bits to detect or correct the errors.



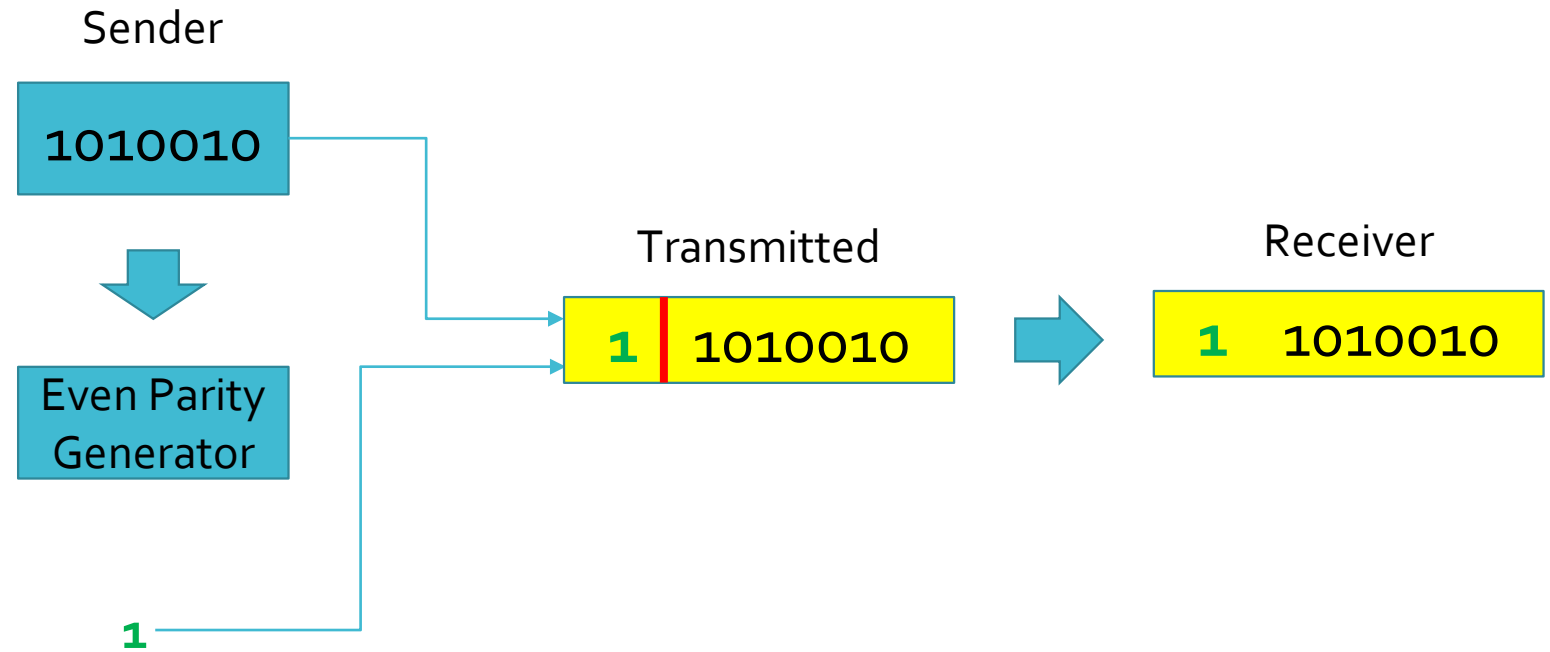
Error Detection Techniques

❖ **There are four error detection techniques:**

1. Vertical Redundancy Check (VRC)
2. Longitudinal Redundancy Check (LRC)
3. Checksum
4. Cyclic Redundancy Check(CRC)
5. Two Dimensional Parity Check

Vertical Redundancy Check (VRC)

❖ It is also known as parity check.



Performance of VRC

- ❖ It can detect single bit error.
- ❖ It can detect burst error if the number of errors are odd.

Sender 1 10110 Transmission error 1 1**1**110 Receiver rejects data

Sender 1 10110 Transmission error 1 1**1**1**0**0 Receiver accepts data

Longitudinal Redundancy Check (LRC)

- ❖ A block of bit is divided into rows and columns.
- ❖ It is known as two dimensional parity check.
- ❖ The parity bit is calculated for each column and sent along with the data.
- ❖ The block of parity acts as the redundant bits.

❖ **11001 00110 11110 00010 00001 10100 11111**

1	1	0	0	1
0	0	1	1	0
1	1	1	1	0
0	0	0	1	0
1	0	1	0	0
1	1	1	1	1
0	1	0	0	0

01000	11111	10100	00010	11110	00110	11001
-------	-------	-------	-------	-------	-------	-------

Performance of LRC

- ❖ If two bits of in one data units are damaged and two bits in exactly the same positions in another unit are also damaged, the LRC will not detect any error.

1	1	0	0	1
0	0	1	1	0
1	1->0	1	1	0
0	0->1	0	1	0
1	0	1	0	0
1	1	1	1	1
0	1	0	0	0

Checksum

- ✓ Checksum = Check + Sum
- ✓ Sender side → Checksum creation
- ✓ Receiver side → Checksum validation

Checksum

❑ Checksum Sender Side

- ✓ Break the original message into 'k' number of blocks with 'n' bits in each block.
- ✓ Sum all the 'k' data blocks.
- ✓ Add the carry to the sum, if any.
- ✓ Do 1's complement to the sum ☐ checksum

Checksum

Original Message: 10011001111000100010010010000100

10011001	11100010	00100100	10000100
----------	----------	----------	----------

				1	0	0	1	1	0	0	1
				1	1	1	0	0	0	1	0
				0	0	1	0	0	1	0	0
				1	0	0	0	0	1	0	0
Carry	1	0	Sum	0	0	1	0	0	0	1	1
										1	0
Sum				0	0	1	0	0	1	0	1
Checksum(1's complement)				1	1	0	1	1	0	1	0

Transmitted Message:

Checksum	Original message			
11011010	10011001	11100010	00100100	10000100

Checksum

❑ Checksum Receiver Side

- ✓ Collect all the data blocks including the checksum.
- ✓ Sum all the data blocks and the checksum.
- ✓ If the result is all 1's accept; else reject.

Checksum

Received Message

11011010	10011001	11100010	00100100	10000100
----------	----------	----------	----------	----------

				1	0	0	1	1	0	0	1
				1	1	1	0	0	0	1	0
				0	0	1	0	0	1	0	0
				1	0	0	0	0	1	0	0
				1	1	0	1	1	0	1	0
Carry	1	0	Sum	1	1	1	1	1	1	0	1
										1	0
Sum				1	1	1	1	1	1	1	1

Performance of Checksum

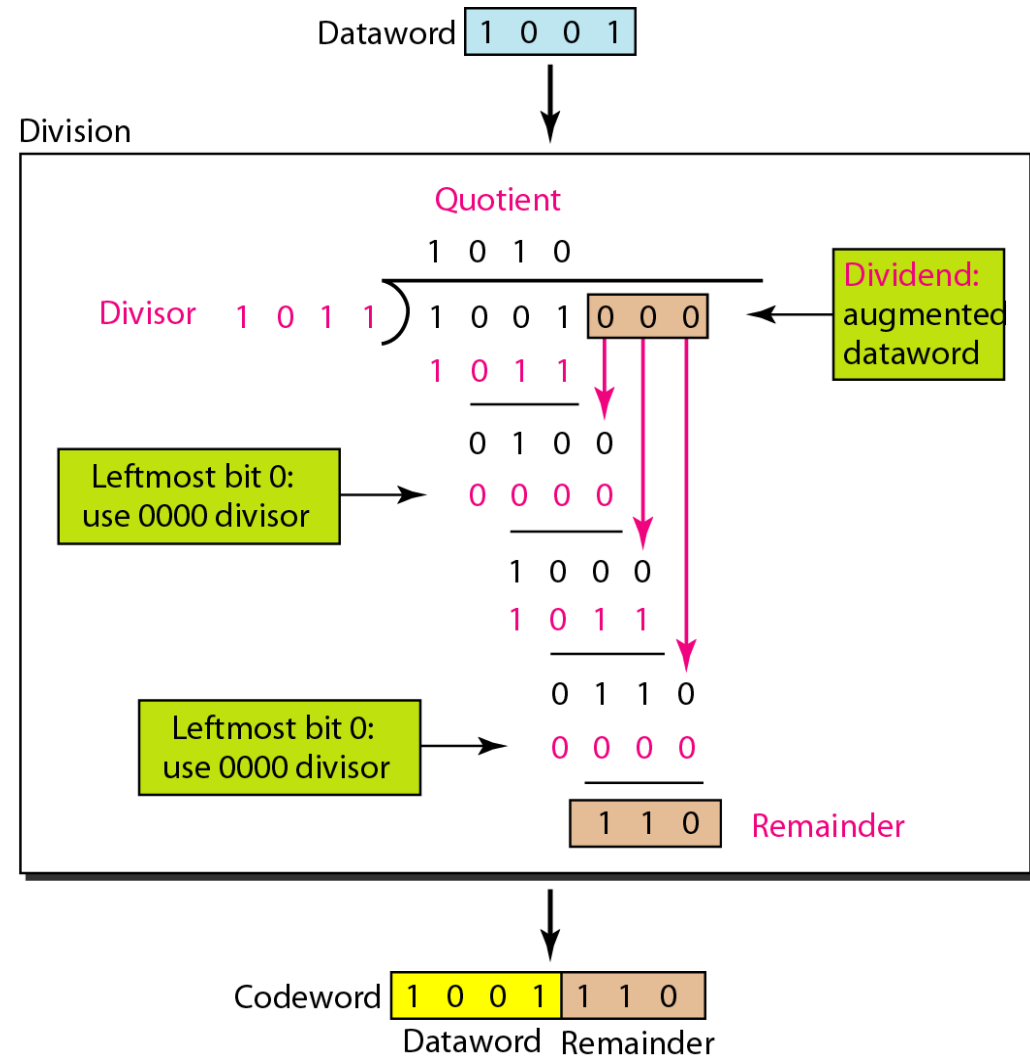
- ✓ It can check both the single bit and burst errors.
- ✓ If one more bits of a segment are damaged and the corresponding bit or bits value in a second segment are also damaged, the sum of those columns will not change and the receiver will not detect the errors.

Cyclic Redundancy Check (CRC)

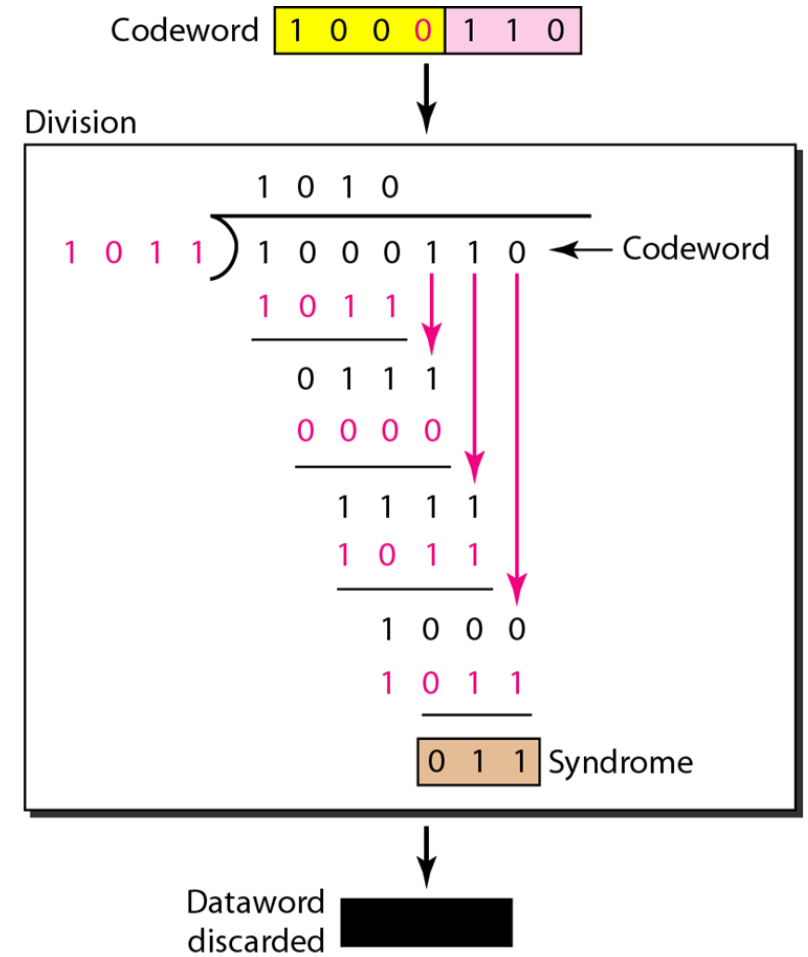
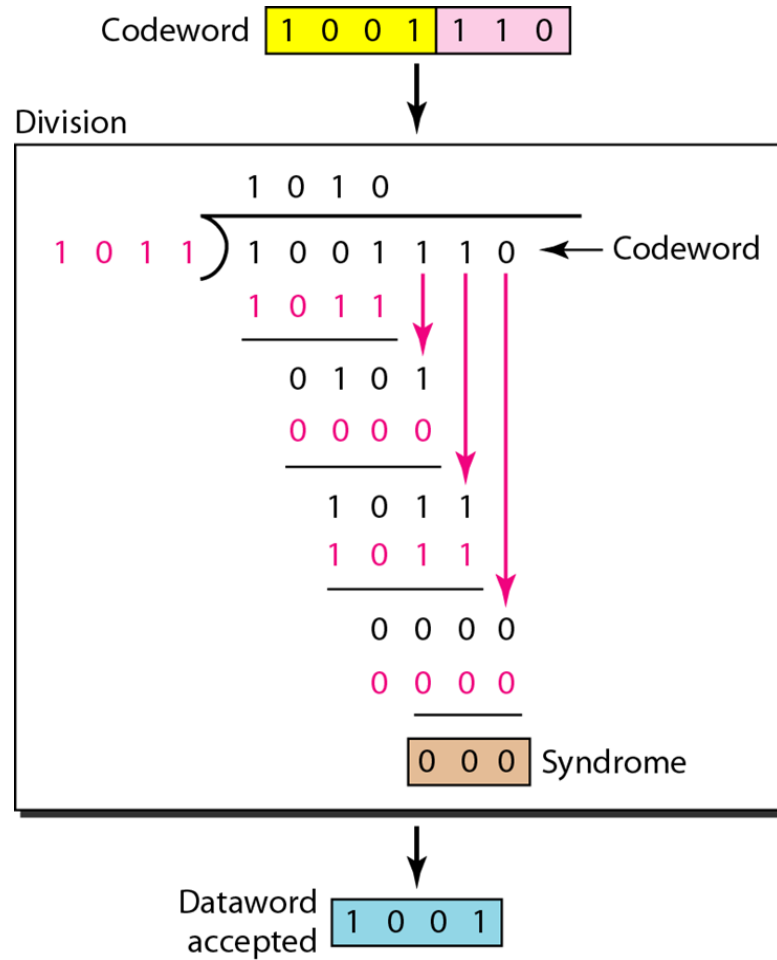
Steps of CRC operation at sender side:

1. Find the length of the divisor 'L'.
2. Append 'L-1' bits (0's) to the original message.
3. Perform binary division operation.
4. Remainder of the division = CRC.

Cyclic Redundancy Check (CRC)



Cyclic Redundancy Check (CRC)



Two Dimensional Parity Check

1	1	0	0	1	1	1	1	Row parities
1	0	1	1	1	0	1	1	
0	1	1	1	0	0	1	0	
0	1	0	1	0	0	1	1	
Column parities								
0	1	0	1	0	1	0	1	

a. Design of row and column parities

Two Dimensional Parity Check

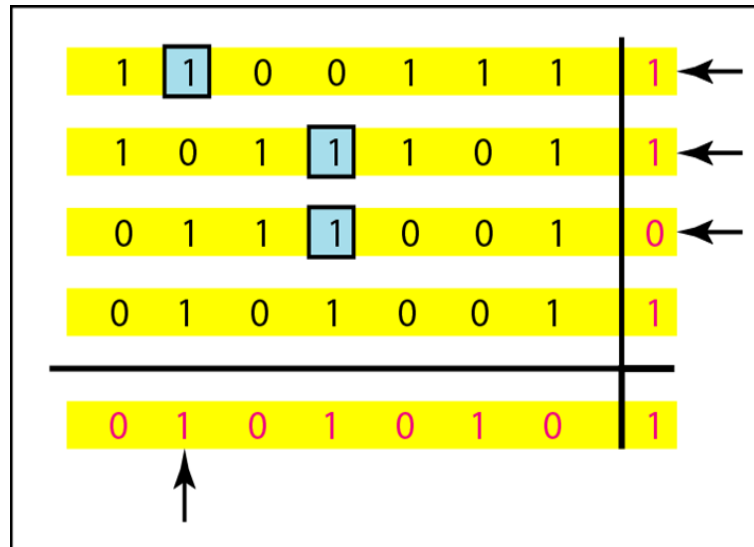
1	1	0	0	1	1	1	1
1	0	1	1	1	0	1	1
0	1	1	1	0	0	1	0
0	1	0	1	0	0	1	1
0	1	0	1	0	1	0	1

b. One error affects two parities

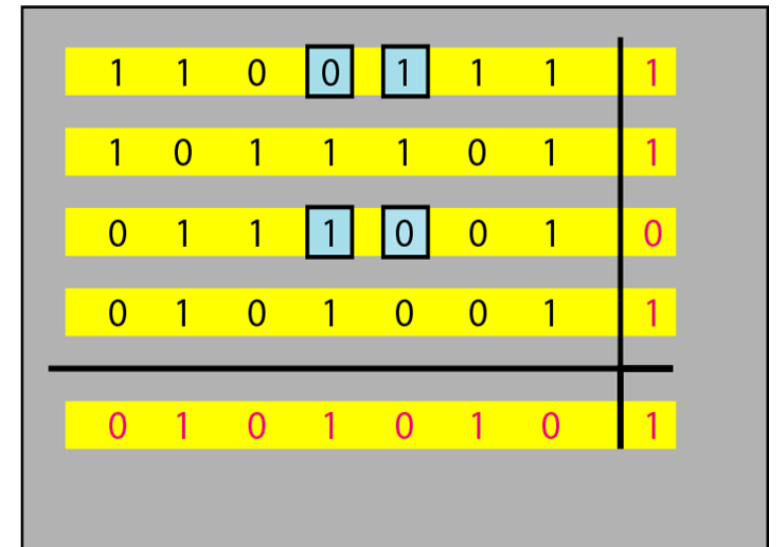
1	1	0	0	1	1	1	1
1	0	1	1	1	0	1	1
0	1	1	1	0	0	1	0
0	1	0	1	0	0	1	1
0	1	0	1	0	1	0	1

c. Two errors affect two parities

Two Dimensional Parity Check



d. Three errors affect four parities



e. Four errors cannot be detected



Thank You 😊