DATA COMMUNICATION

CSE 225/233

WEEK-10, LESSON-1 & 2

ERROR DETECTION

Background

Networks must be able to transfer data from one device to another with acceptable accuracy. For most applications, a system must guarantee that the data received are identical to the data transmitted. Any time data are transmitted from one node to the next, they can become corrupted in passage. Many factors can alter one or more bits of a message. Some applications require a mechanism for detecting and correcting errors.

Types of Error

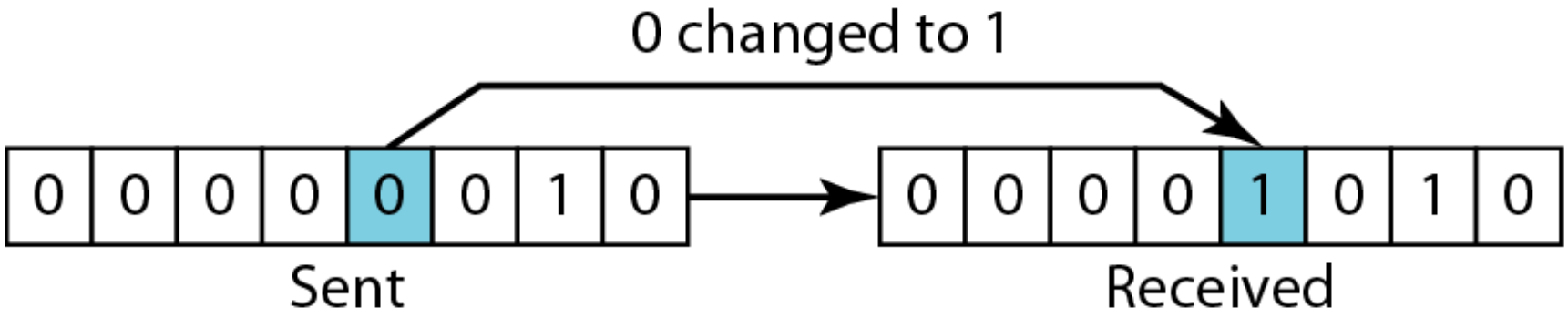
Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference. This interference can change the shape of the signal.

There are two types of Error:

- Single Bit Error
- Burst Error

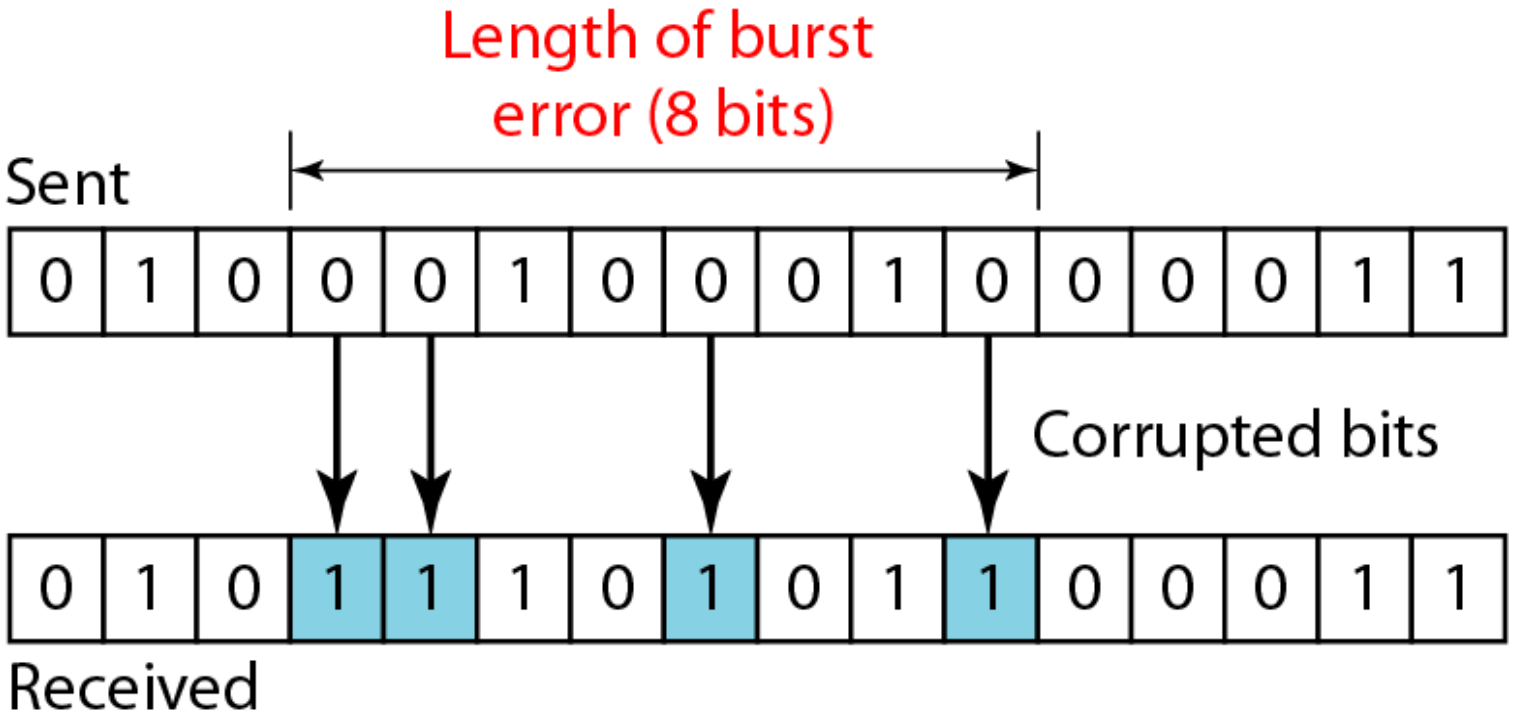
Single bit Error

The term single-bit error means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1.



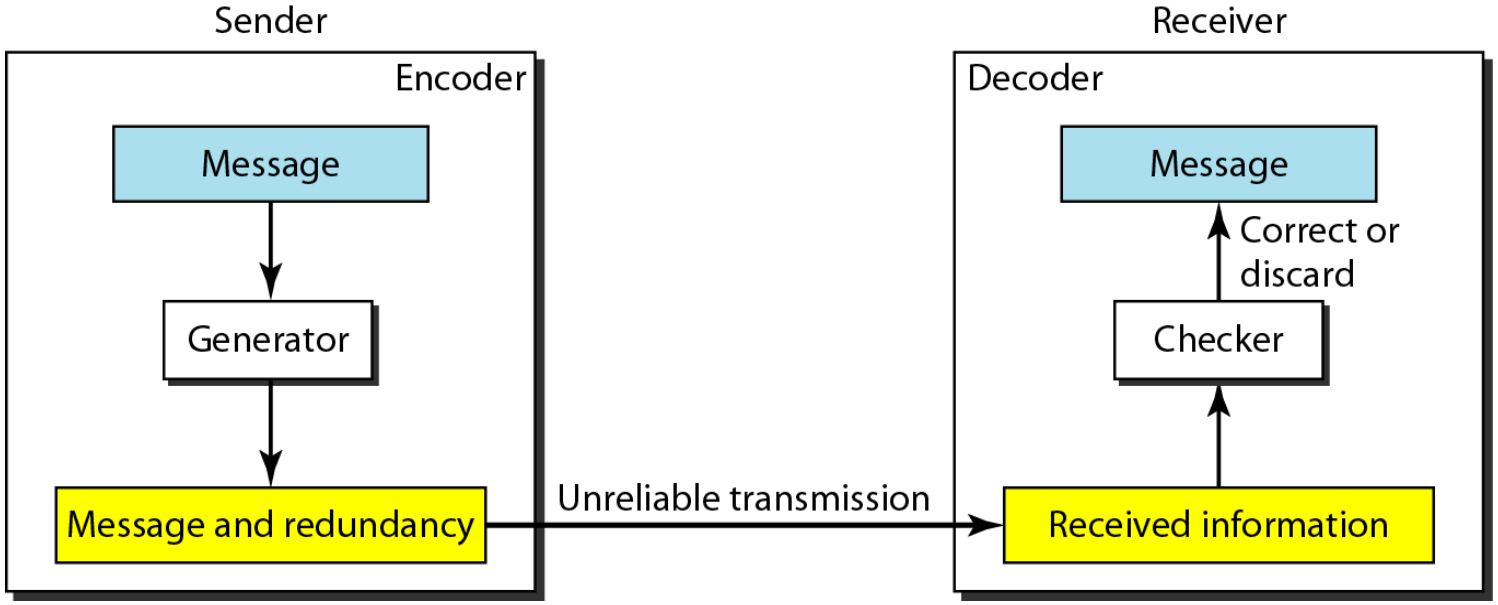
Burst Error

The term burst error means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1. Following figure shows the effect of a burst error on a data unit.



Redundancy

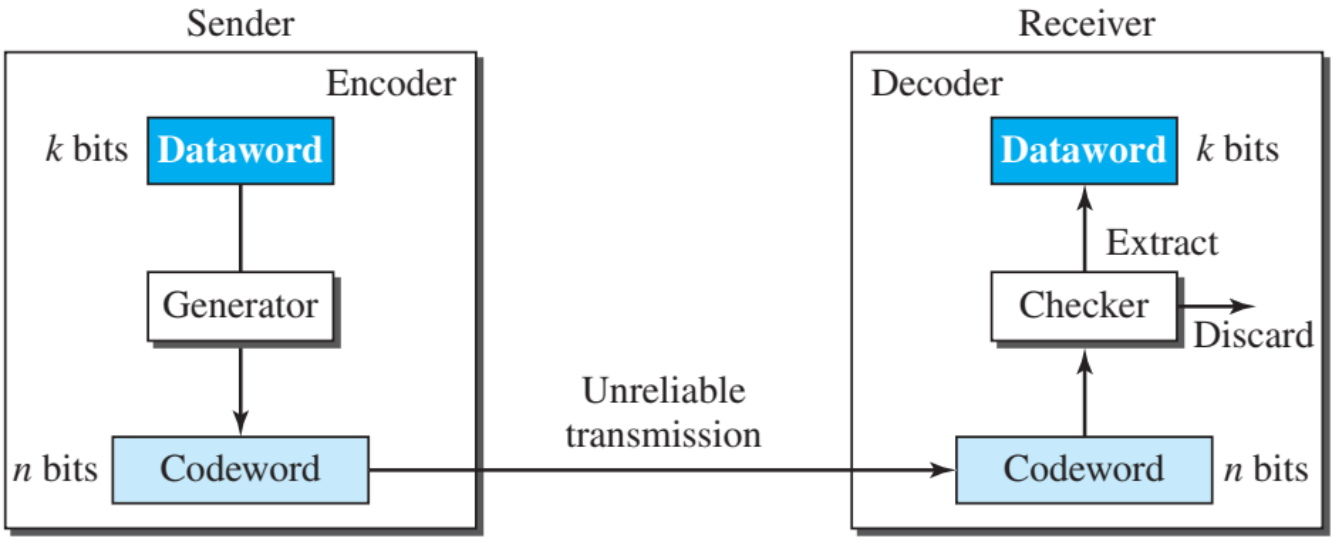
The central concept in detecting or correcting errors is redundancy. To be able to detect or correct errors, we need to send some extra bits with our data. These redundant bits are added by the sender and removed by the receiver. Their presence allows the receiver to detect or correct corrupted bits.



Error Detection

How can errors be detected by using block coding? If the following two conditions are met, the receiver can detect a change in the original codeword.

1. The receiver has (or can find) a list of valid codewords.
2. The original codeword has changed to an invalid one.



Hamming Distance

One of the central concepts in coding for error control is the idea of the Hamming distance. The Hamming distance between two words (of the same size) is the number of differences between the corresponding bits. We show the Hamming distance between two words x and y as d(x, y).

For example, if the codeword 00000 is sent and 01101 is received, 3 bits are in error and the Hamming distance between the two is d(00000, 01101) = 3. In other words, if the Hamming distance between the sent and the received codeword is not zero, the codeword has been corrupted during transmission.

The Hamming distance can easily be found if we apply the XOR operation ($\oplus$) on the two words and count the number of 1s in the result.

Example

Let us find the Hamming distance between two pairs of words.

1. The Hamming distance d(000, 011) is 2 because…

$$000 \oplus 011 \text{ is } 011 \text{ (two 1s)}$$

2. The Hamming distance d(10101, 11110) is 3 because..

$$10101 \oplus 11110 \text{ is } 01011 \text{ (three 1s)}$$

Minimum Hamming Distance

The minimum Hamming distance is the smallest Hamming distance between all possible pairs in a set of words.

To guarantee the detection of up to s errors in all cases, the minimum Hamming distance in a block code must be

$d_{min} = s+1$

Example (1)

Find the minimum Hamming distance of the following coding scheme.

We first find all Hamming distances.

$d(000, 011) = 2$	$d(000, 101) = 2$	$d(000, 110) = 2$	$d(011, 101) = 2$
$d(011, 110) = 2$	$d(101, 110) = 2$		

The d_{min} in this case is 2.

Example (2)

Find the minimum Hamming distance of the following coding scheme.

We first find all Hamming distances.

$d(00000, 01011) = 3$	$d(00000, 10101) = 3$	$d(00000, 11110) = 4$
$d(01011, 10101) = 4$	$d(01011, 11110) = 3$	$d(10101, 11110) = 3$

The d_{min} in this case is 3.

Example (3)

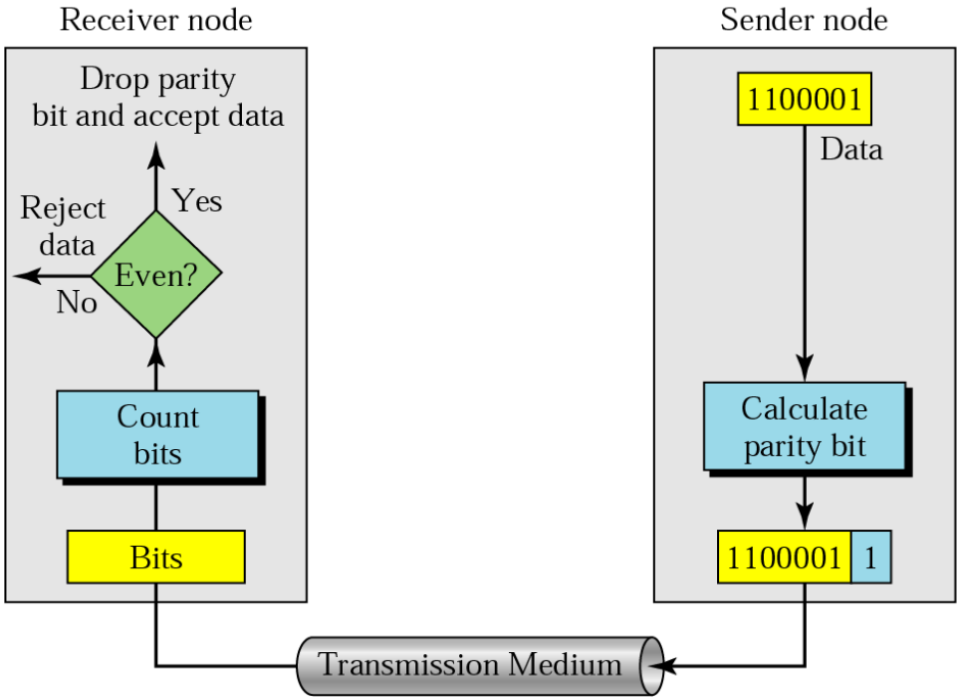
A code scheme has a Hamming distance dmin = 4. What is the error detection and correction capability of this scheme?

Solution:

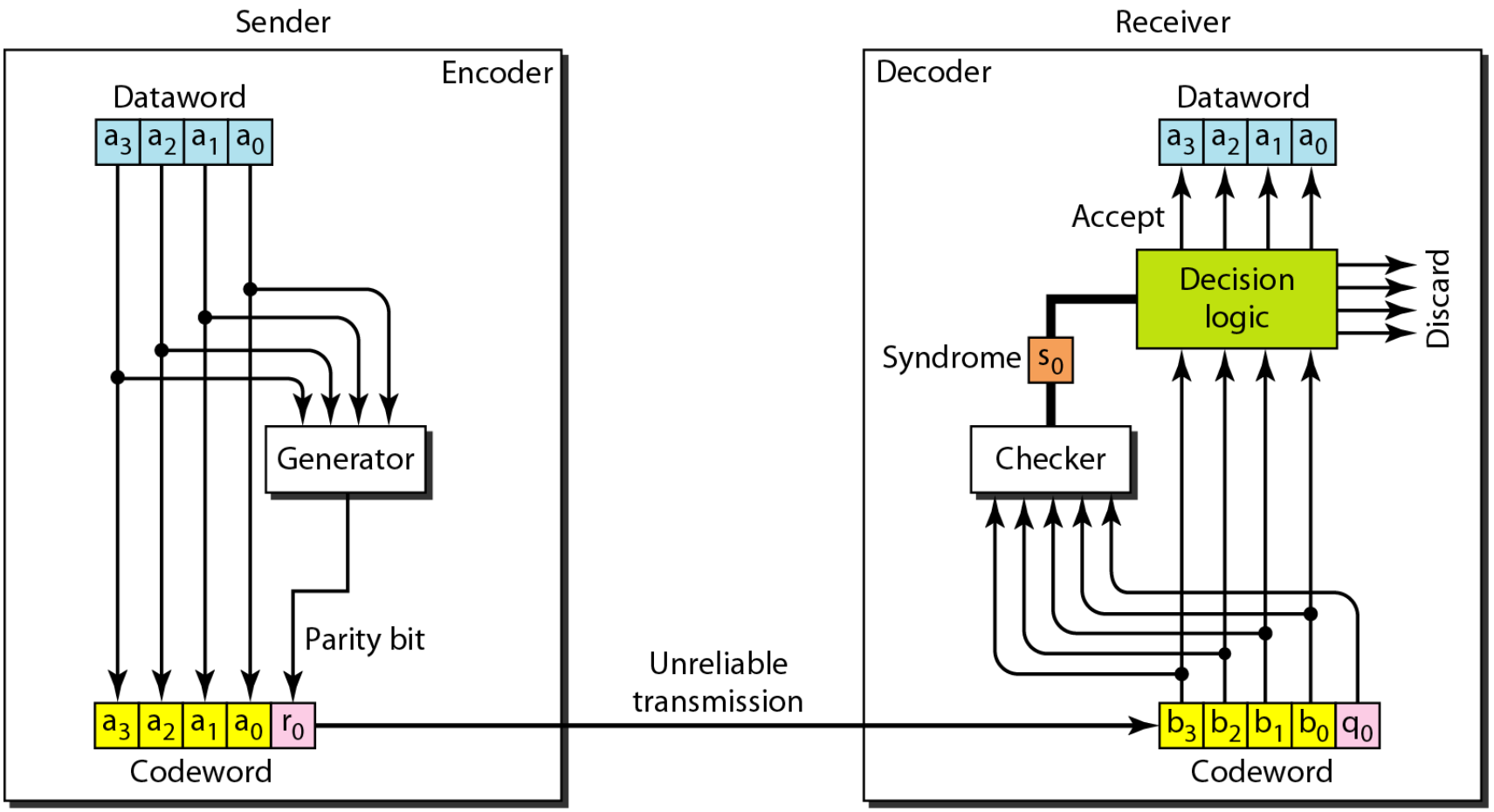
This code guarantees the detection of up to three errors (s = 3), but it can correct up to one error. In other words, if this code is used for error correction, part of its capability is wasted. Error correction codes need to have an odd minimum distance (3, 5, 7, . . .).

Simple Parity Check

- Simple or Two Dimensional
- In parity check, a parity bit is added to every data unit so that the total number of 1s is even (or odd for odd-parity).



Encoder and Decoder 0f Simple Parity Check



Example (1)

Suppose the sender wants to send the word *world*. In ASCII the five characters are coded as

1110111 1101111 1110010 1101100 1100100

The following shows the actual bytes sent

11101110 11011110 11100100 11011000 11001001

Example (1) Contd.

Now suppose the word world in Example 1 is received by the receiver without being corrupted in transmission.

11101110 11011110 11100100 11011000 11001001

The receiver counts the 1s in each character and comes up with even numbers (6, 6, 4, 4, 4). The data are accepted.

Example (1) Contd.

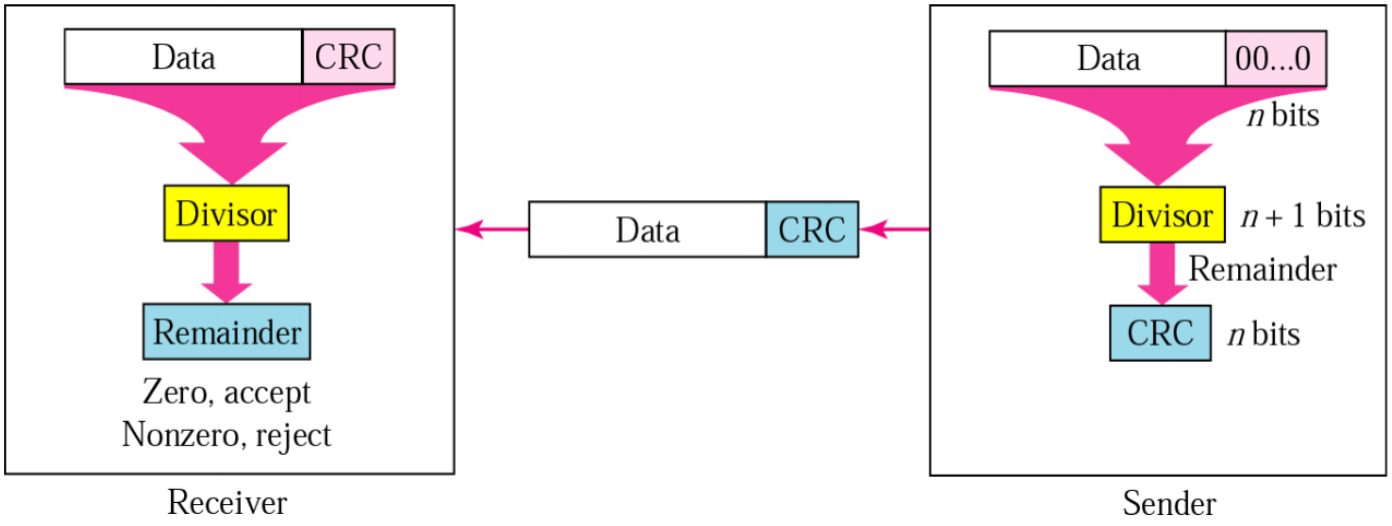
Now suppose the word world in Example 1 (Slide 16) is corrupted during transmission.

11111110 11011110 11101100 11011000 11001001

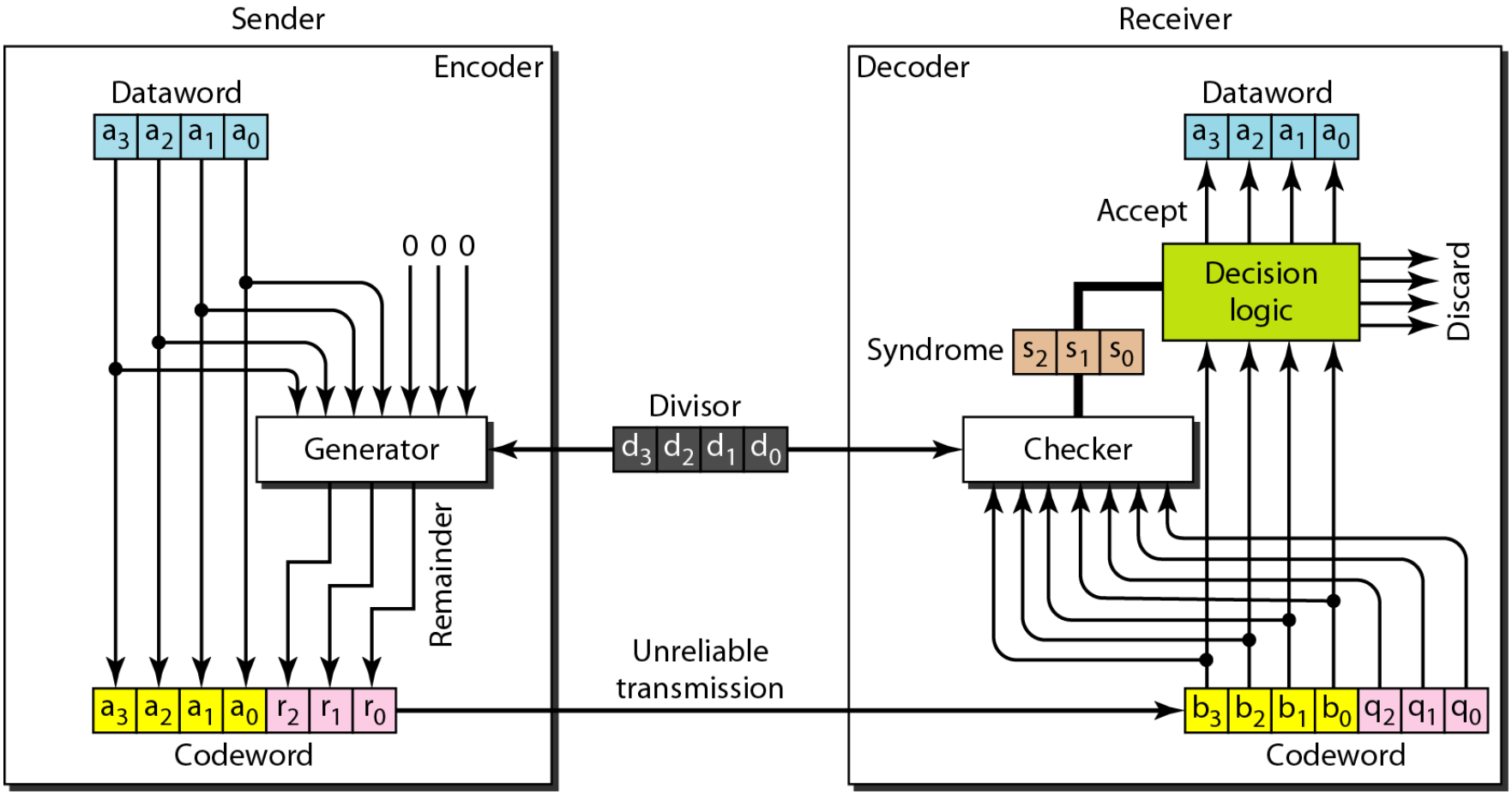
The receiver counts the 1s in each character and comes up with even and odd numbers (7, 6, 5, 4, 4). The receiver knows that the data are corrupted, discards them, and asks for retransmission.

Cyclic Redundancy Check

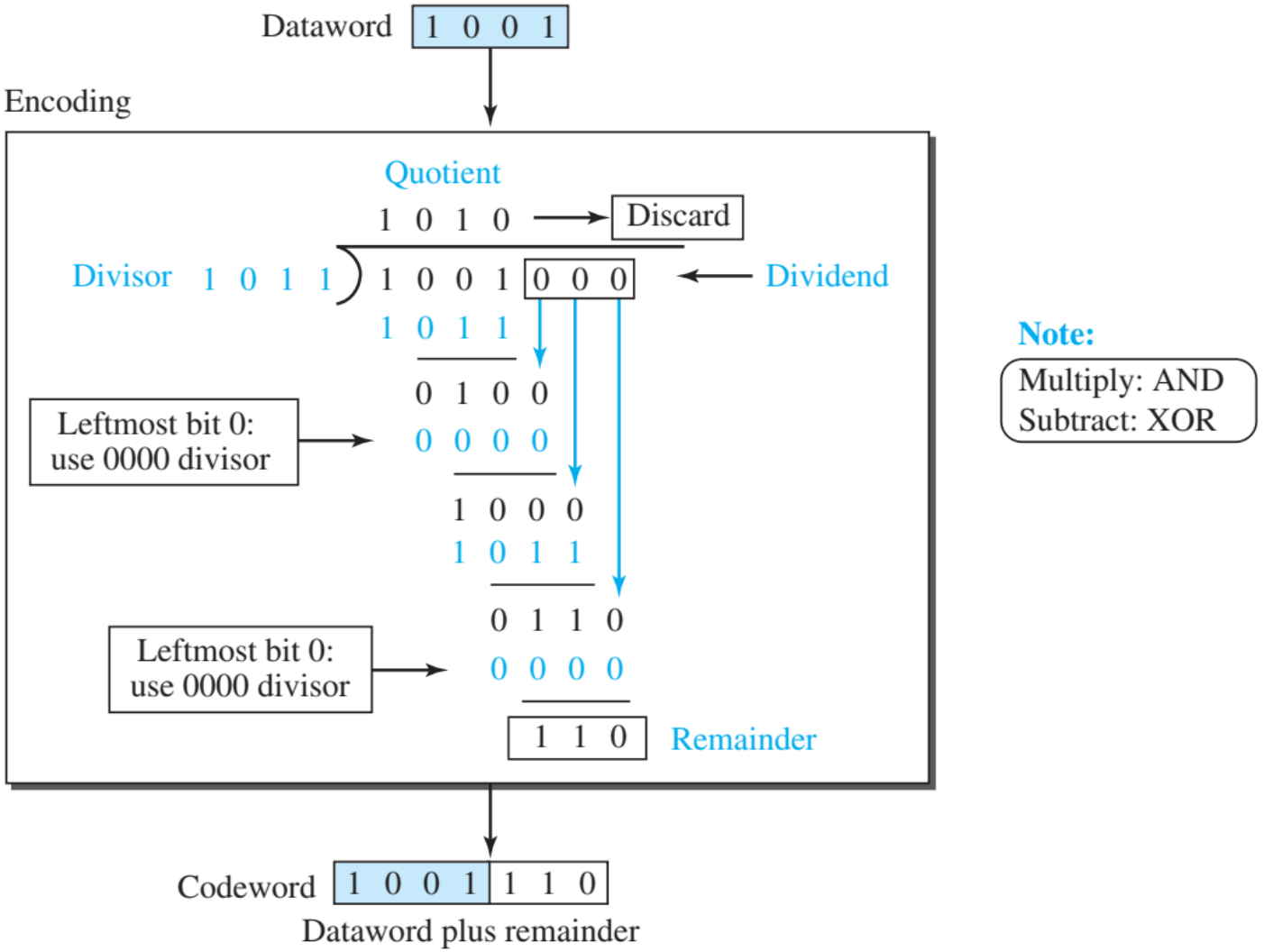
- Uses Binary division
- Add CRC remainder to data.
- Number of 0s appended is one less than the Divisor
- Appending CRC to the end of the data must make resulting bit sequence divisible by the divisor



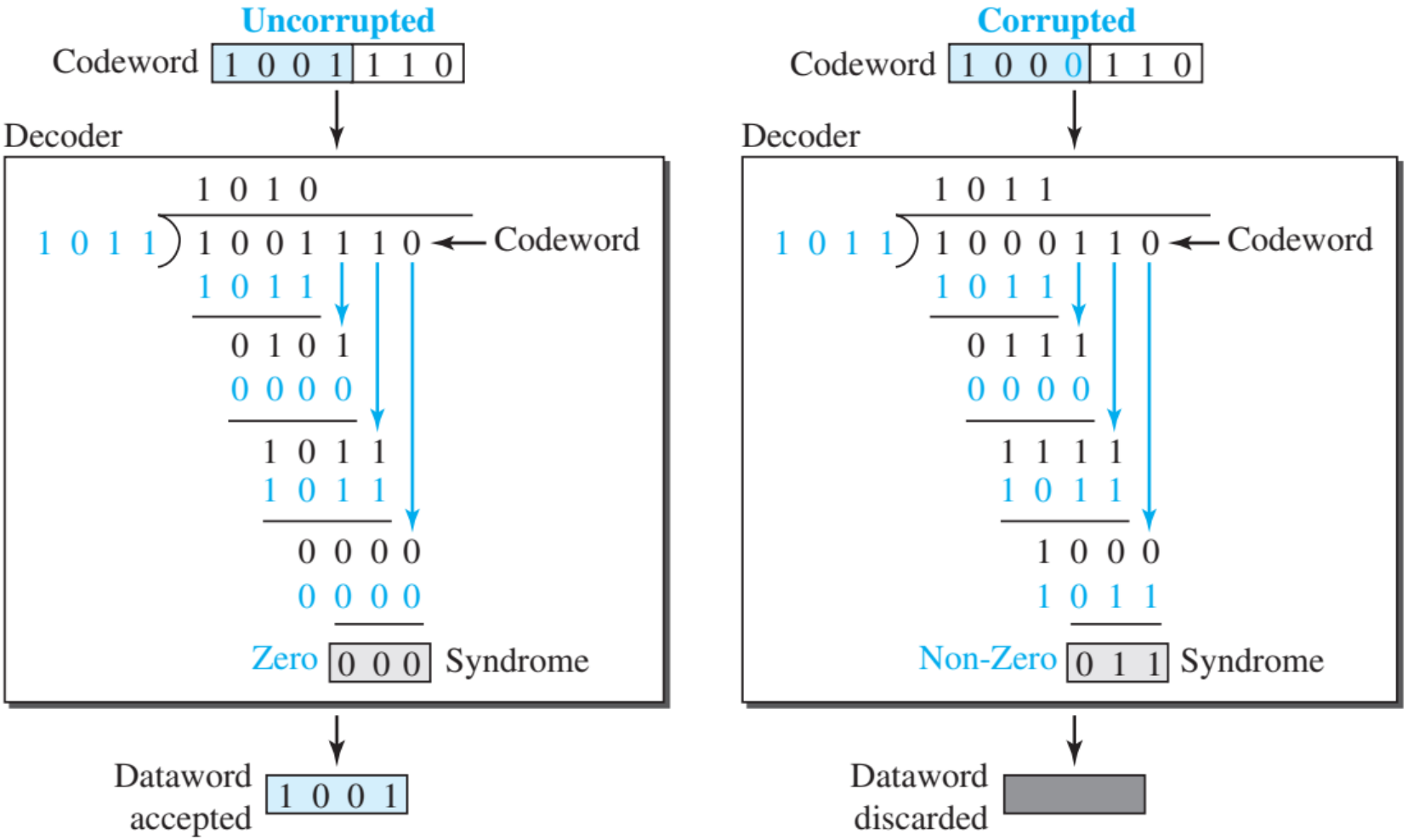
CRC Encoder and Decoder



Division in CRC Encoder



Division in CRC Encoder for Two case




Thank you!