lecture 3 Data Link Layer - Intro and Error Detection and Correction

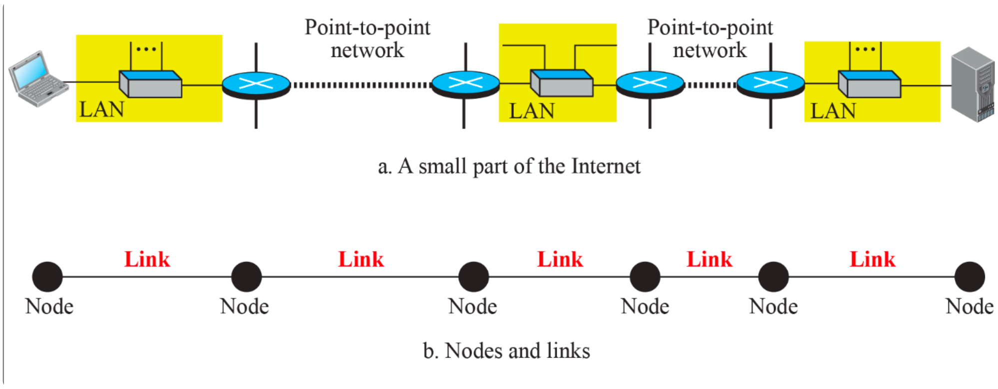
Objectives

- the major functions at link layer
- the idea of block coding and Hamming distance for error detection and correction.
- three commonly used error detection
- forward error correction Hamming codes for error correction

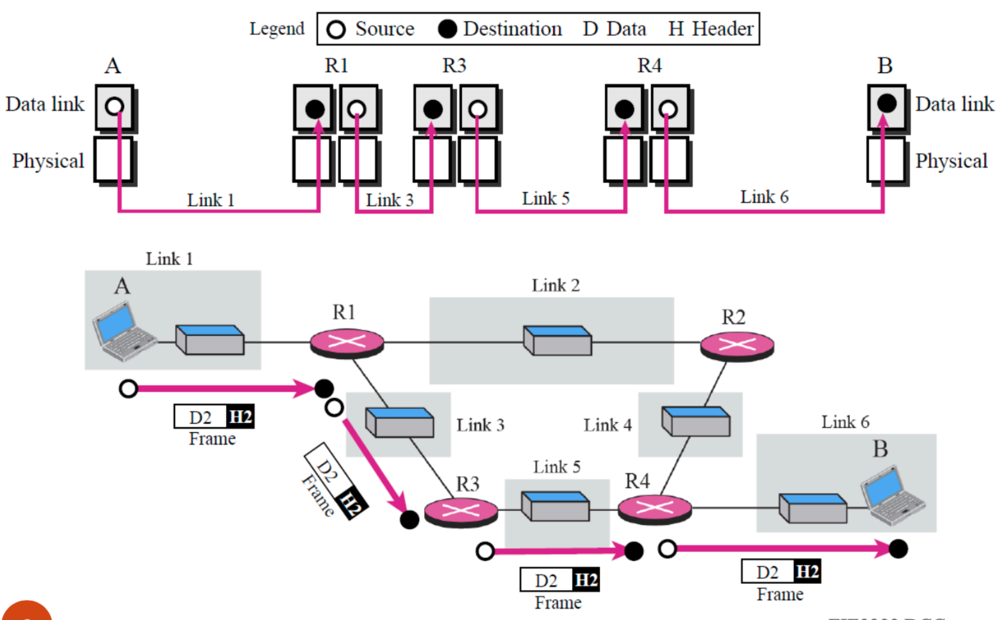
---

Introduction

Nodes and Links

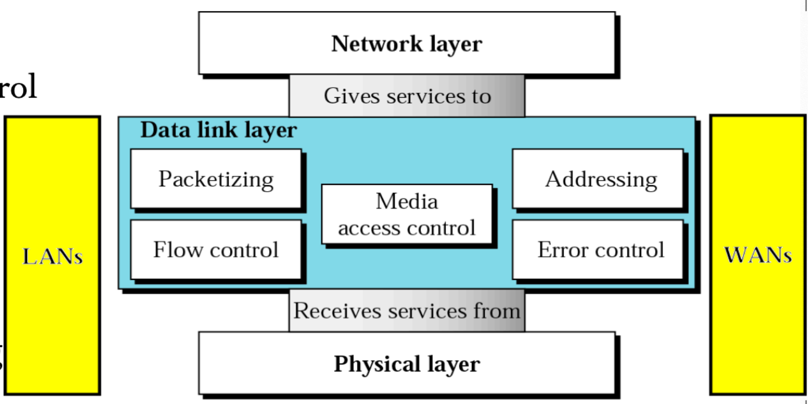


Communications at Link Layer



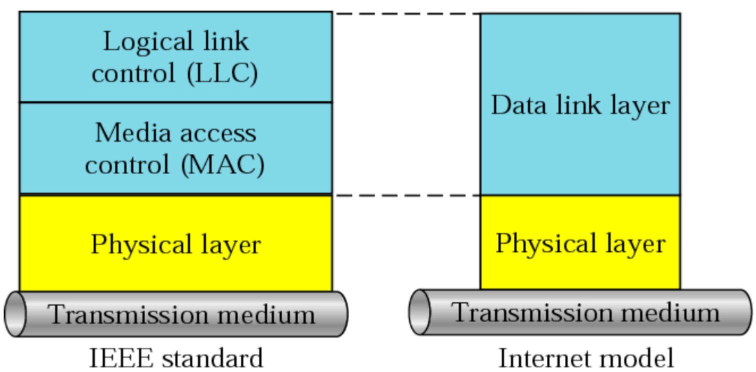
Functions of Data Link Layer

- Framing
- Flow Control
- Error Control
- Access Control
- Addressing



LLC and MAC Sublayers

- Logical Link Control (LLC): Interoperability for different LANs
- Media Access Control (MAC): Operation of the access methods



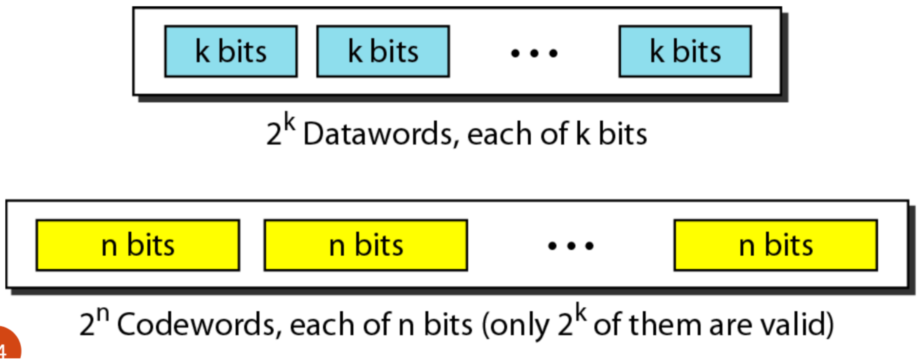
Error Detection and Correction

Type of Errors

- single-Bit error
- Burst Error

Block Coding

- Message is divided into k-bit blocks
  - Known as ***datawords***
- r redundant bits are added
  - Blocks become $n=k+r$ bits
  - Known as ***codewords***



Minimum Hamming Distance

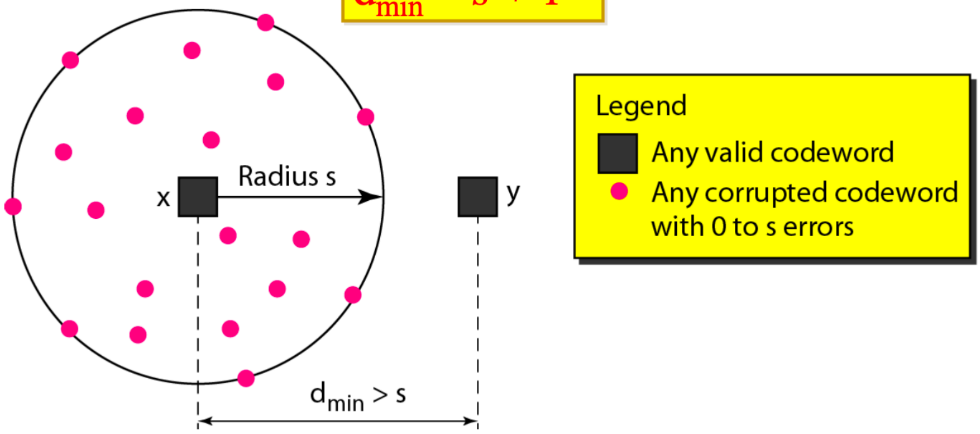
The minimum Hamming distance is the smallest Hamming distance between all possible pairs of codewords in a codebook

- Find the minimum Hamming Distance of the following codebook

00000
01011
10101
11110

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

$$d_{min} = s + 1$$

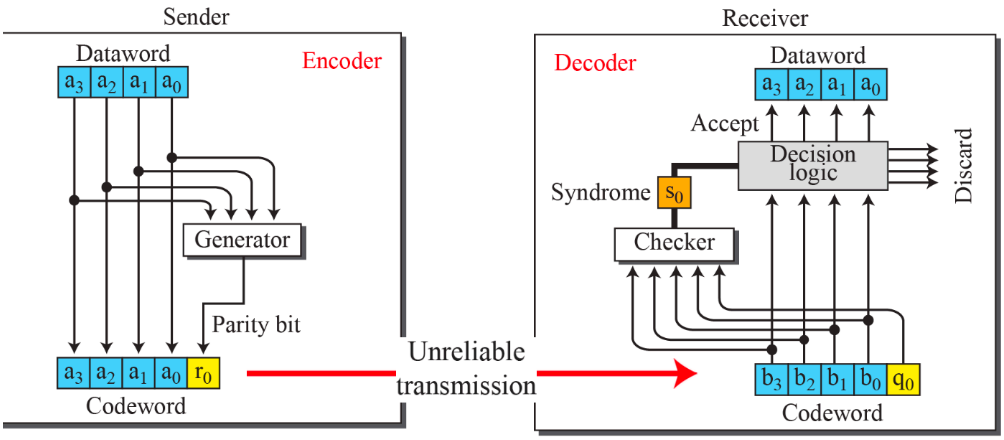


Error Detection and Correction - Parity Check

- A parity bit is added to every data unit so that the total number of 1s is even (or odd for odd-parity)

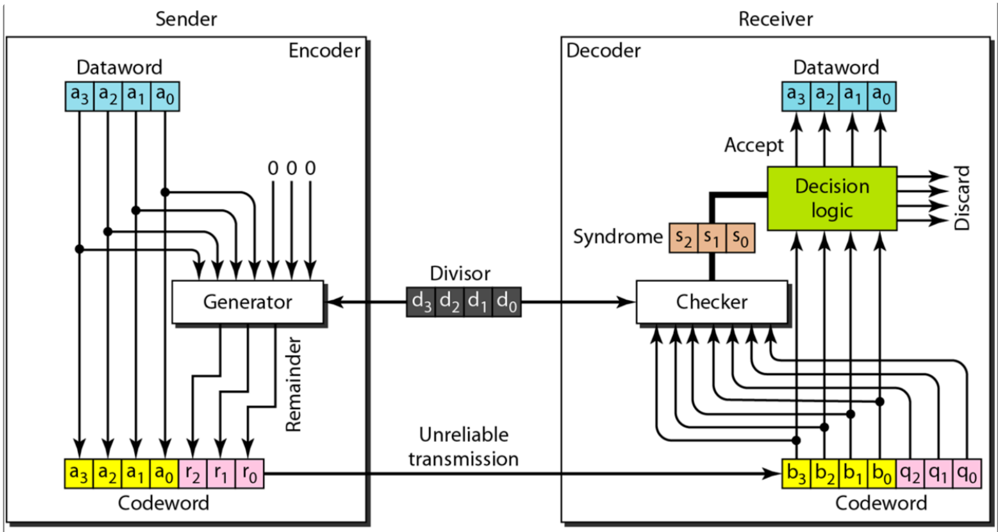
- The parity bit is obtained by adding the k bits of the codewords (modulo-2), for example, for a 4-bit block

$$r_0 = a_3 \oplus a_2 \oplus a_1 \oplus a_0 \quad \text{(modulo-2)}$$



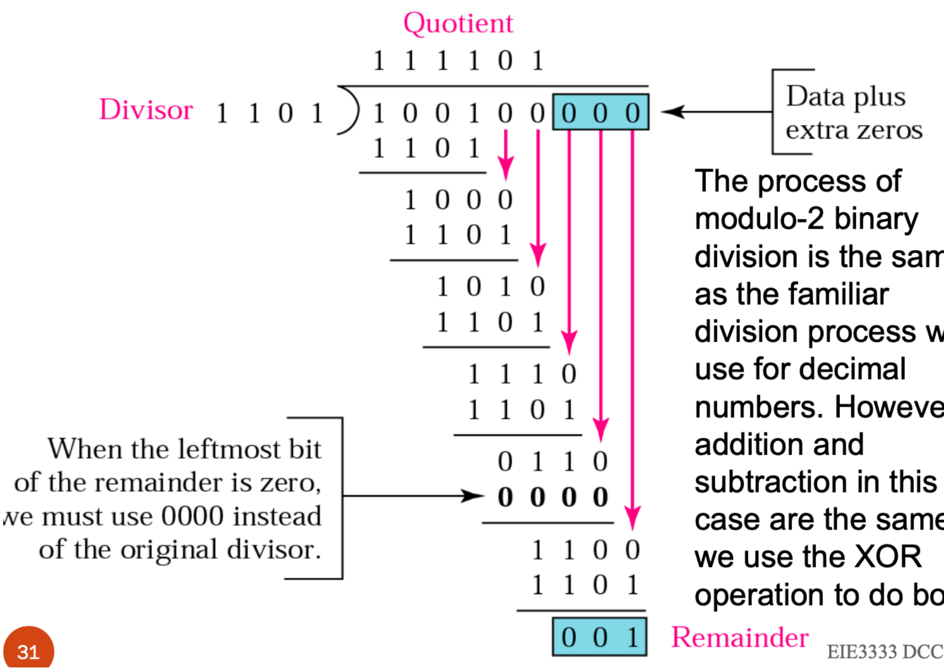
Error Detection- Cyclic Redundancy Check

- based on binary division
- a sequence of redundant bits so that the resulting data unit becomes exactly divisible by a second predetermined binary number



The remainder is always one bit smaller than the divisor

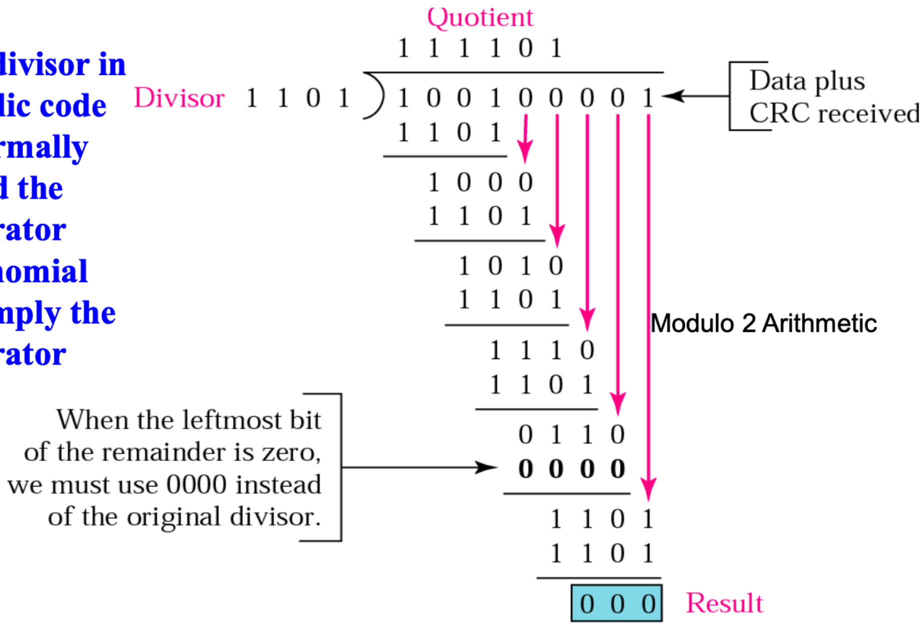
CRC Generator



The process of modulo-2 binary division is the same as the familiar division process we use for decimal numbers. However, addition and subtraction in this case are the same; we use the XOR operation to do both.

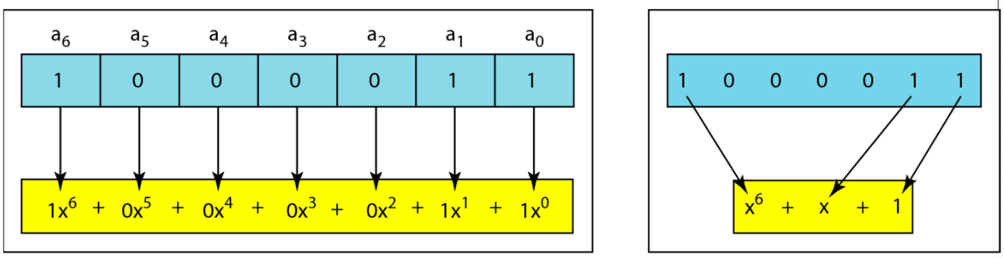
CRC Checker

The divisor in a cyclic code is normally called the generator polynomial or simply the generator



Polynomial Representation

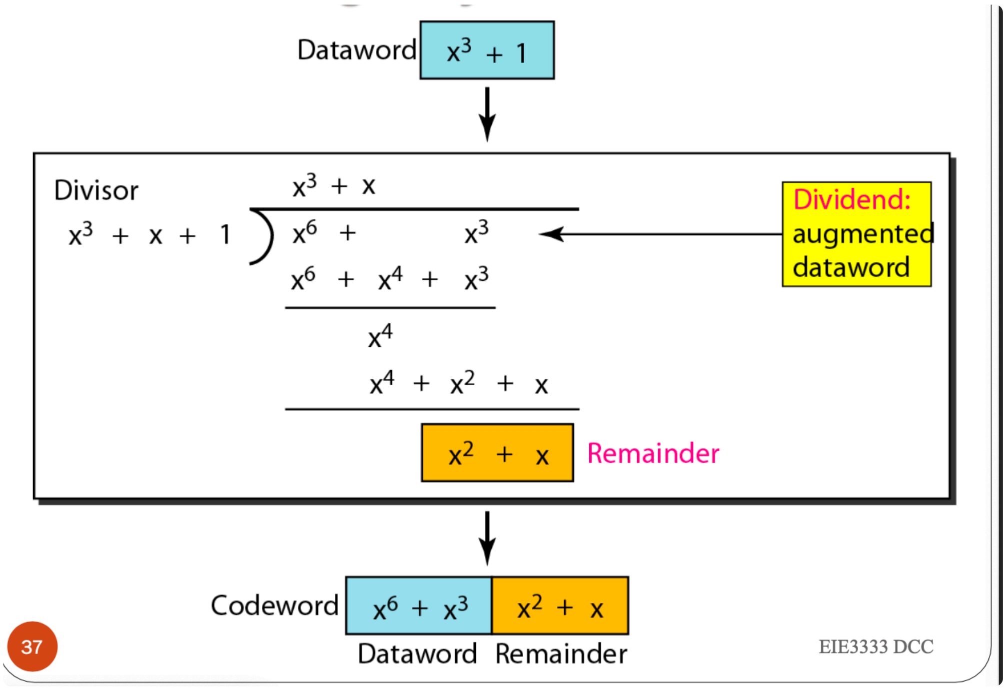
- More common representation than binary form
- Easy to analyze
- Divisor is commonly called ***generator polynomial***



a. Binary pattern and polynomial

b. Short form

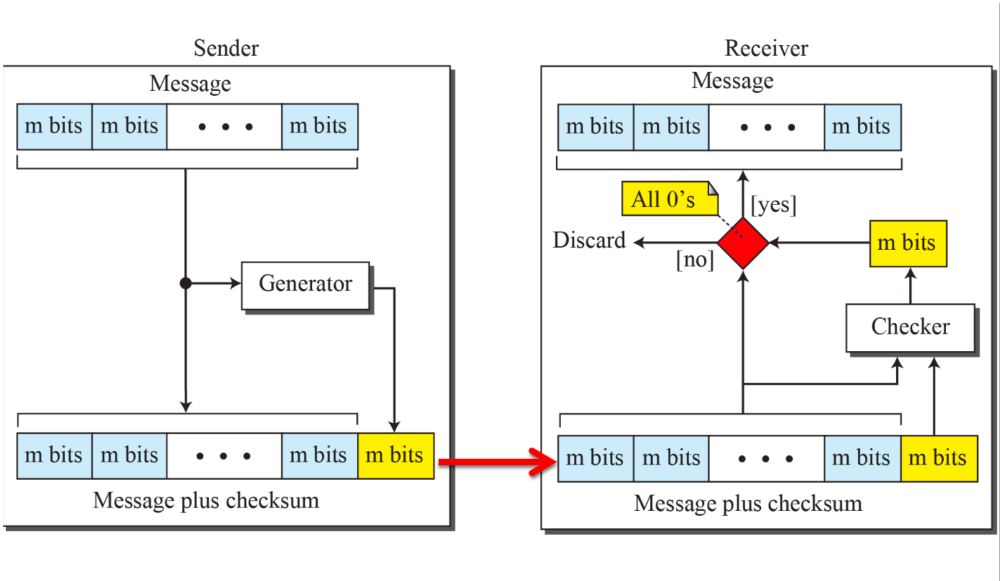
Division Using Polynomial



Error Detection - Checksum

- Checksum generator subdivides the data unit into equal segments of n bits
- These segments are added in one's complement arithmetic so that the total is n bits long
- That total is then complemented and appended to the end of the original data unit as redundancy bits
- The sum of data segment is T, the checksum will be –T

If the number has more than m bits, the extra leftmost bits need to be added to the m rightmost bits (wrapping).



Example

Suppose the following block of 16 bits is to be sent using a checksum of 8 bits

10101001 00111001

The numbers are added using one's complement

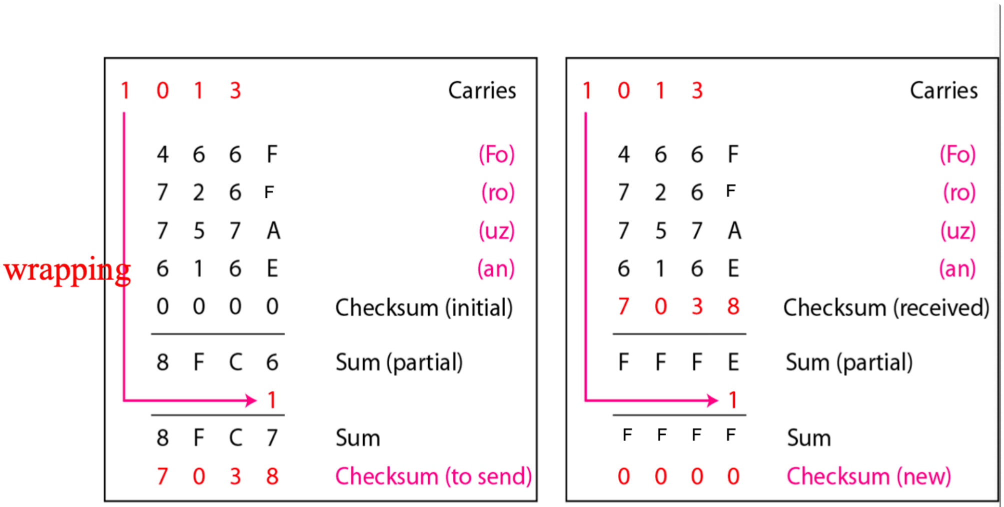
10101001

00111001

Sum 11100010

Checksum **00011101**

The pattern sent is 10101001 00111001 **00011101**



a. Checksum at the sender site

a. Checksum at the receiver site

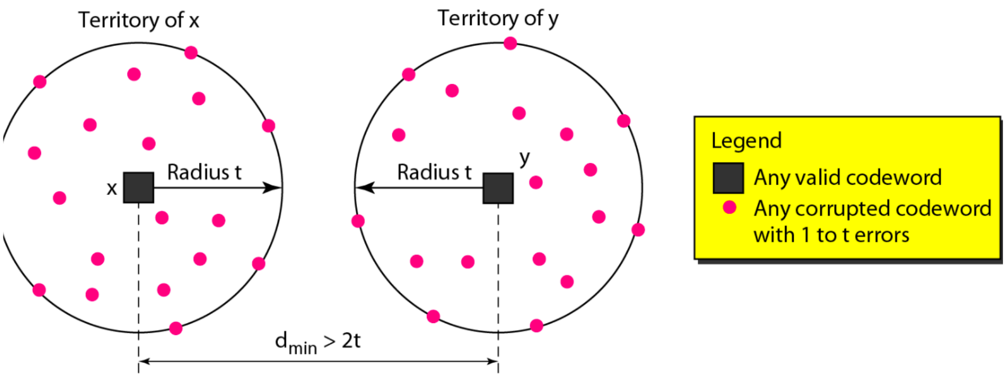
Error Correction

- Forward Error Correction
  - The receiver can use an error-correcting code, which automatically corrects certain errors

Correction Capability of Code

- To guarantee the correction of up to t-bit errors, the minimum Hamming distance in a block code must be

$$\boxed{d_{min} = 2t + 1}$$



Forward Error Correction for 1-Bit Error

- Consider only a single-bit error in k bits of data
  - k possibilities for an error
  - One possibility for no error
  - No. of possibilities $= k + 1$
- Add r redundant bits to distinguish these possibilities; we need

$$2^r \geq k+1$$

- But the r bits are also transmitted along with data; hence

$$2^r \geq k+r+1$$

Number of Redundant Bits

Number of Redundant Bits

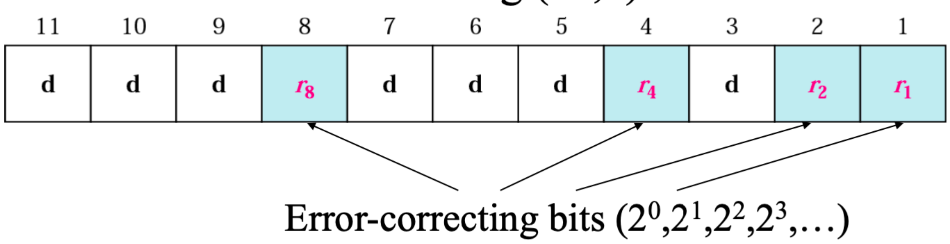
Number of data bits k	Number of redundancy bits r	Total bits $k + r$
1	2	3
2	3	5
3	3	6
4	3	7
5	4	9
6	4	10
7	4	11

Hamming Code

- Adopts parity concept, but have more than one parity bit
- It can detect up to two-bit errors or correct one-bit errors without detection of uncorrected errors

Hamming Code

Hamming (11,7)

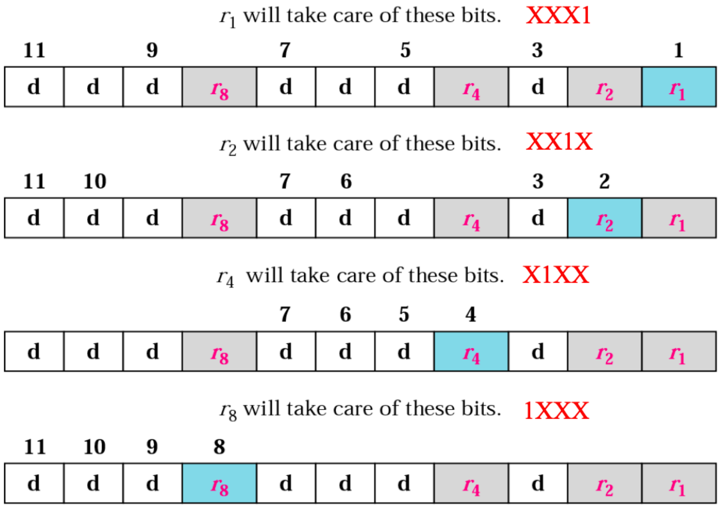


1. Calculate the number of redundancy bits required. Since number of data bits is 7, the value of r is calculated as $2^r > m + r + 1$ --> $2^4 > 7 + 4 + 1$

Therefore number of redundancy bits = 4.

2. Determining the positions of bits and redundancy bits. The EC bits are placed at the positions corresponding to the powers of 2 i.e. 1, 2, 4, 8,…

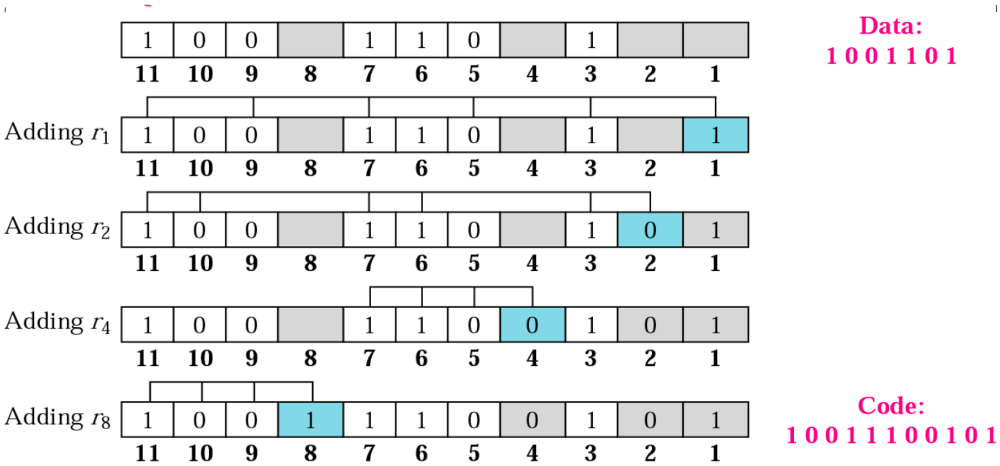
Redundant Bit Calculation



Parity bit 1 covers all bit positions whose binary representation includes a 1 in the rightmost position, 3, 5, 7, 9, etc.

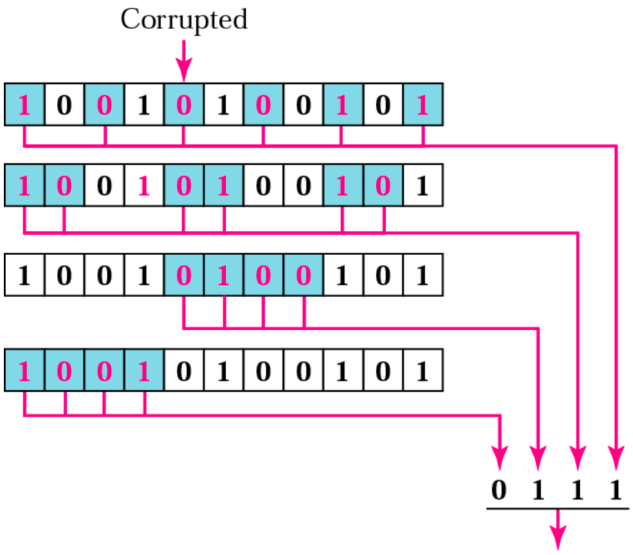
Parity bit 2 covers all bit positions whose binary representation includes a 1 in the second position, 3, 5, 7, 9, etc., …..

Example



The choice of the parity, even or odd, is irrelevant but the same choice must be used for both encoding and decoding (Even parity here)

- Receiver receives 10010100101



The bit in position 7 is in error.

The receiver does the recalculation using the same set of bits plus the relevant parity (r) bit for each set; it then assembles the new parity values into a binary number in order of r position (r_8, r_4, r_2, r_1).