CS311: Data Communication

Channel Coding

by

Dr. Manas Khatua
Assistant Professor
Dept. of CSE
IIT Jodhpur
E-mail: manaskhatua@iitj.ac.in
Web: http://home.iitj.ac.in/~manaskhatua
http://manaskhatua.github.io/

Coding Theory

- Coding theory is the study of the properties of codes and their respective fitness for specific applications.

- Codes are used for
  - Data compression
  - Error-detection and error-correction
  - Networking
  - Cryptography

- the purpose of coding is of designing efficient and reliable data transmission methods.

- There are four types of coding:
  - Source coding
  - **Channel coding**
  - Line coding
  - Cryptographic coding

Cont…

- Source coding
  - The aim of source coding is to take the source data and make it smaller in size.
  - e.g., Zip coding

- Channel coding
  - The purpose is to find codes which transmit quickly, contain many valid code words and can correct or at least detect many errors.
  - e.g., Reed-Solomon code, Turbo code, LDPC code, Cyclic code, Convolution code

- Line coding
  - is called digital baseband modulation technique
  - e.g., unipolar, polar, bipolar, and Manchester encoding

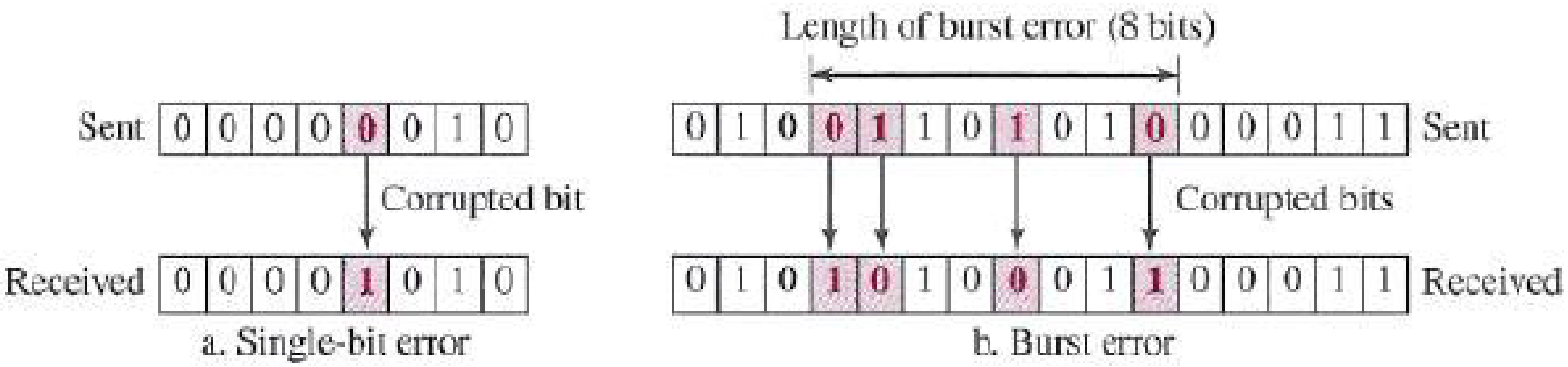
- Cryptographic coding
  - is the practice and study of techniques for secure communication in the presence of third parties
  - e.g., RSA Algorithm

Error Detection and Correction

- Objective:
  - System must guarantee that the data received are identical to the data transmitted

- Methods:
  1. If a frame is corrupted between the two nodes, it needs to be corrected
  2. Drop the frame and let the upper layer (e.g. Transport) protocol to handle it by retransmission

Types of Error



- Single bit error
- Burst error / multibit error

- Reason: noise in the channel

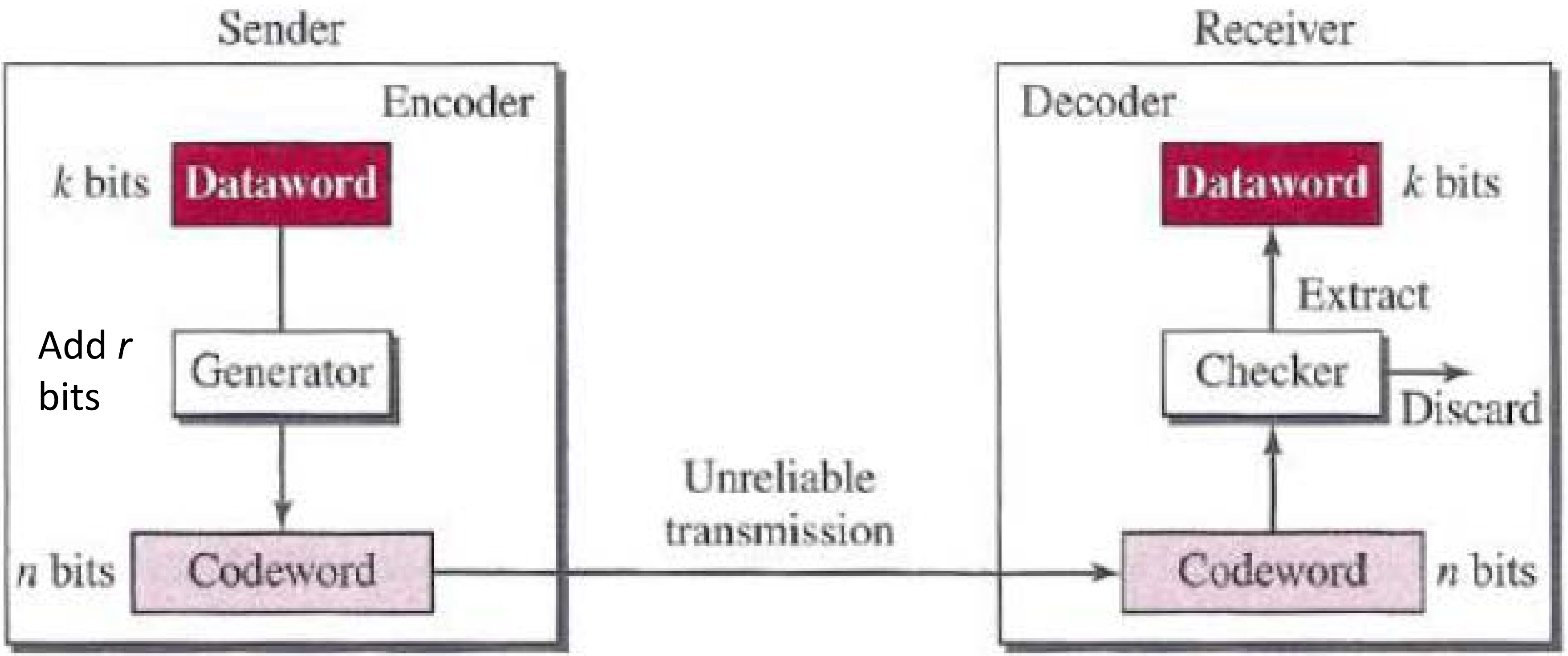
Detection and Correction

- Central idea: redundancy
  - put some extra bit with our data
  - Achieved by channel coding scheme
    - Linear block coding : the sum of any two codewords is also a code word
      - Cyclic codes (e.g., Hamming codes)
      - Repetition codes
      - Parity codes
      - Polynomial codes (e.g., BCH codes)
      - Reed–Solomon codes
      - Algebraic geometric codes
      - Reed–Muller codes
      - Perfect codes
    - Convolution coding: make every codeword symbol be the weighted sum of the various input message symbols

- Error Detection : looking to see if any error has occurred

- Error Correction: trying to recover the corruption
  - Need to know exact number of bits that are corrupted
  - Needs the position of those bits

Block Coding



- How the extra *r* bits are chosen or calculated?
- How can errors be detected?
  - Finds the existence of invalid codeword

Example

- Let us assume that $k = 2$ and $n = 3$.
- Table shows the list of datawords and codewords.

Dataword	Codeword
00	000
01	011
10	101
11	110

Assume the sender encodes the dataword 01 as 011 and sends it to the receiver.

Possible options at receiver (assume one bit corruption):
011 => correct
111 => invalid
001 => invalid
010 => invalid

Hamming Distance

- The Hamming distance between two words (of the same size) is the number of differences between the corresponding bits.

- Notation: Hamming distance between two words *x* and *y* as *d(x, y).*

- Calculation*:* apply the XOR operation on the two words and count the number of 1's in the result

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

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

Types of Block Codes

- the block coder is a *memoryless* device.
- Algebraic block code/ linear block code / cyclic block code

- Linear block codes have the property of linearity:
  – the exclusive OR (addition modulo-2) of two valid codewords creates another valid codeword.

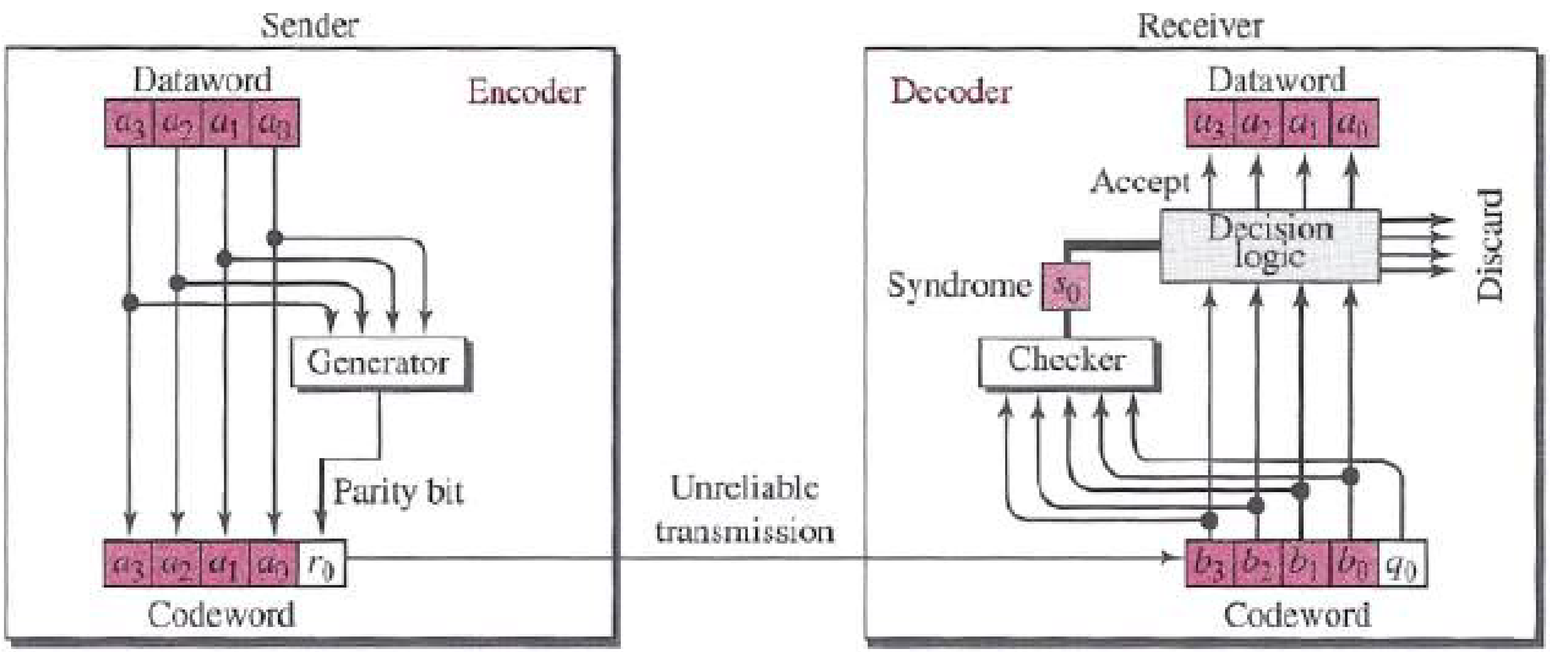
- Example:
  – Parity Check Code
  – Cyclic Redundancy Check

Dataword	Codeword
00	000
01	011
10	101
11	110

- Minimum Hamming distance: number of 1s in the nonzero valid codeword with the smallest number of 1s.

Parity Check Code

Dataword	Codeword	Dataword	Codeword
0000	00000	1000	10001
0001	00011	1001	10010
0010	00101	1010	10100
0011	00110	1011	10111



Parity Check Code

- Modulo arithmetic:

- Generator:

  $r_0 = a_3 + a_2 + a_1 + a_0$ (modulo-2)

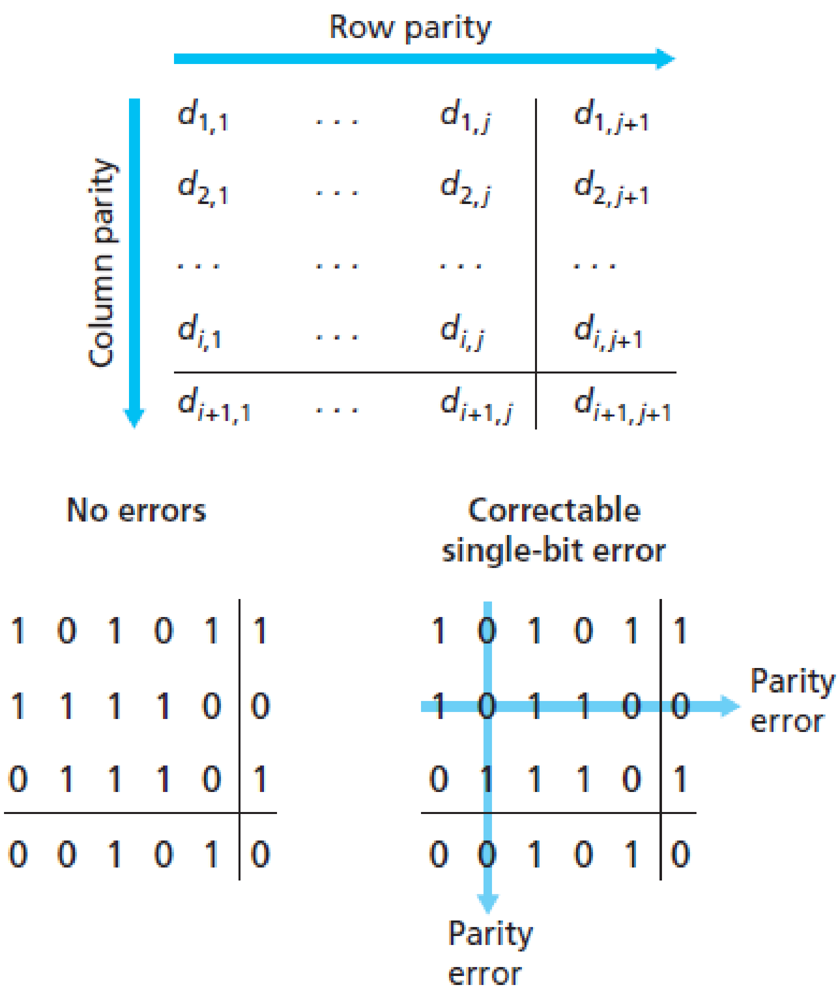
- Checker:

  $s_0 = a_3 + a_2 + a_1 + a_0 + q_0$ (modulo-2)

- A parity-check code can detect an odd number of errors.

- what happens if an even number of bit errors occur?
  - a more robust error-detection scheme is needed

Insight into Error-Correction



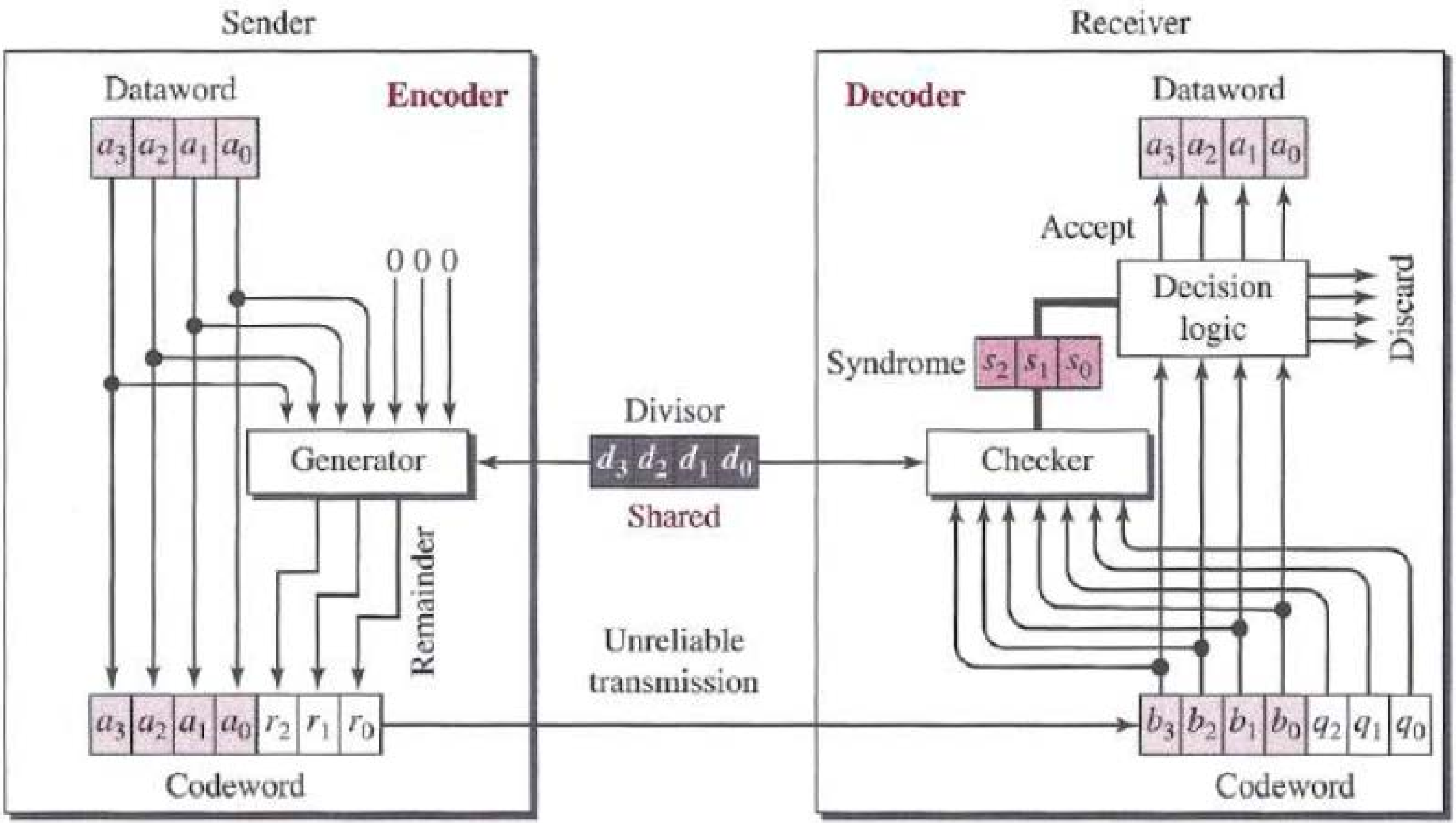
- two-dimensional parity scheme
- The receiver can thus not only *detect* but can *identify* the bit that has corrupted.
- The ability of the receiver to both detect and correct errors is known as forward error correction (FEC).

Cyclic Redundancy Check (CRC)

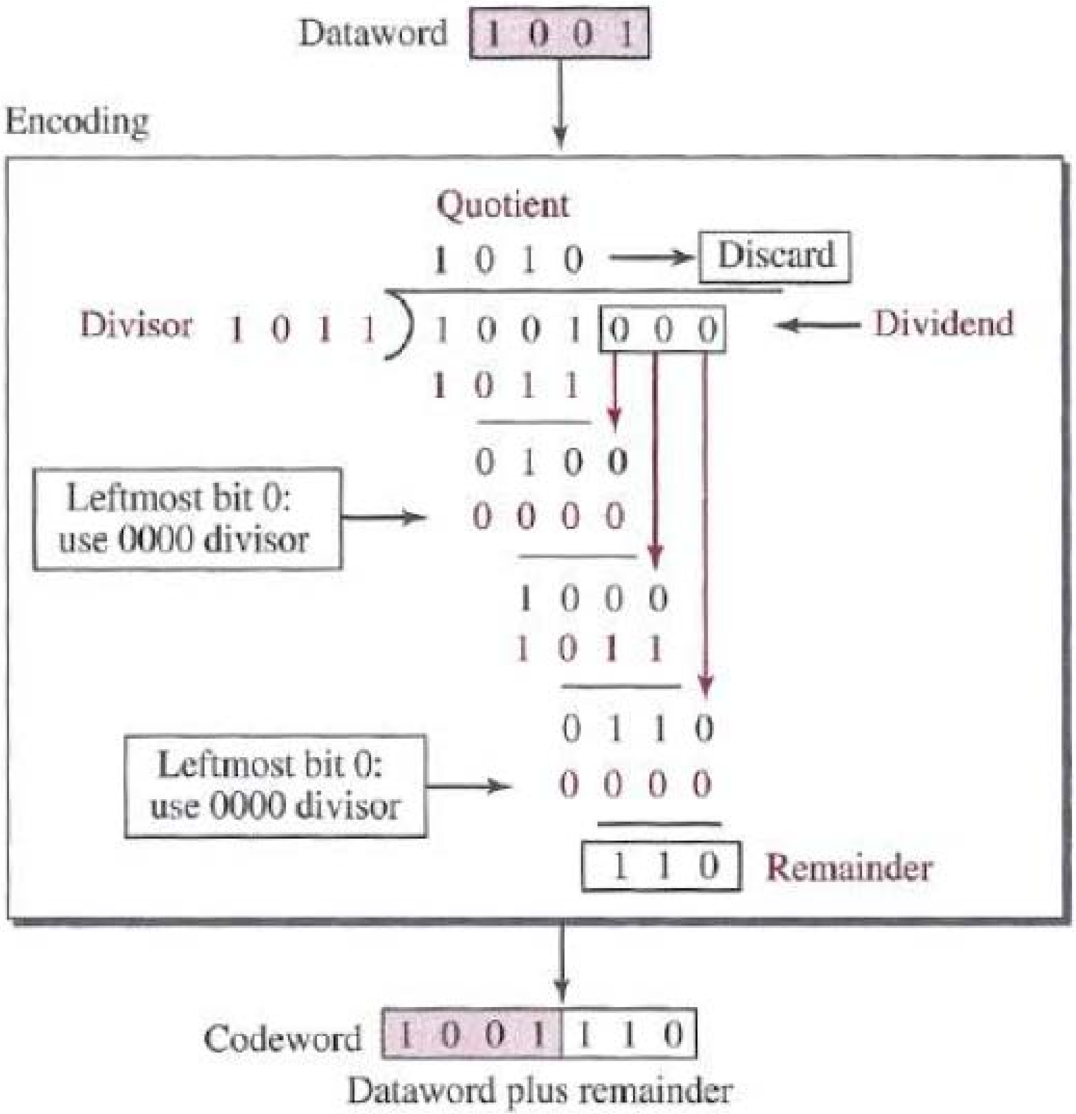
- It is an error-detection technique used widely in today's computer networks (e.g., LAN, WAN)
- developed by *W. Wesley Peterson* in 1961
- It is linear block code but cyclic in nature
- If a codeword is cyclically shifted (rotated), the result is another codeword.
  - E.g., if **1011000** is a codeword and we cyclically left-shift, then **0110001** is also a codeword.

CRC code with C(7,4)

- C(7,4) => 4 bits dataword, 7 bits codeword



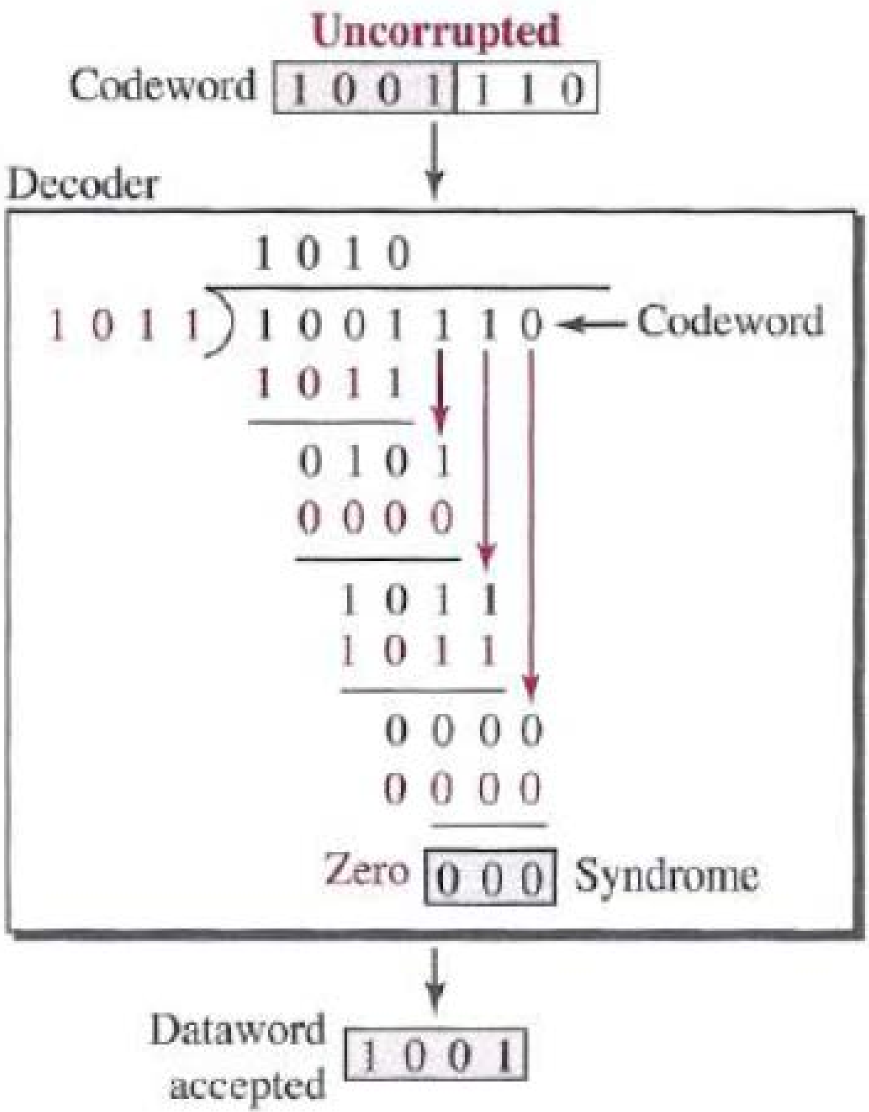
CRC Encoding



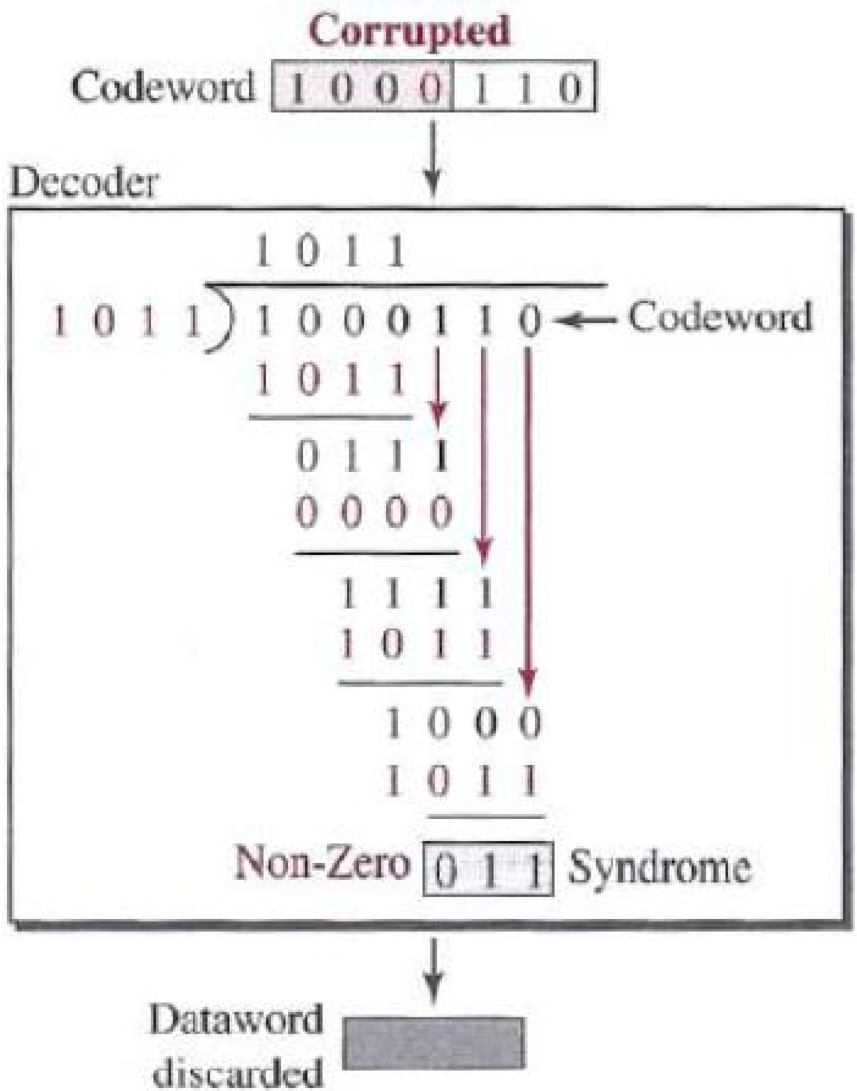
Note:

Multiply: AND
Subtract: XOR

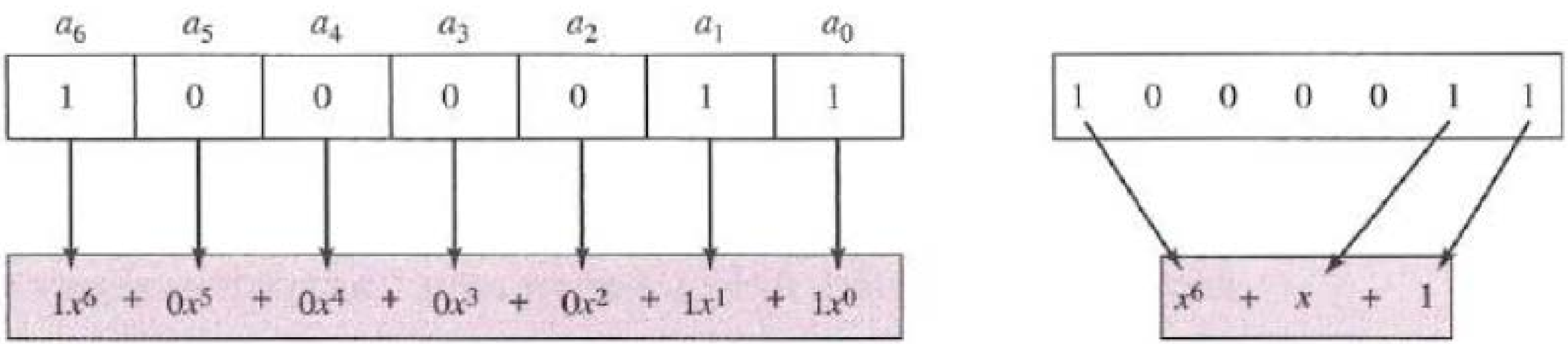
CRC Decoding



CRC Decoding



Polynomial Representation



a. Binary pattern and polynomial

b. Short form

- The power of each term shows the position of the bit
- The coefficient shows the value of the bit
- The degree of a polynomial is the highest power in the polynomial.

Polynomial Operations

- Adding and Subtracting Polynomials
  - Not same as it is performed in mathematics
  - adding or subtracting is done by combining terms and deleting pairs of identical terms
  - E.g., $(x^5+x^4+x^2) + (x^6+x^4+x^2) \Rightarrow x^6+x^5$
  - addition and subtraction are the same

- Multiplying or Dividing Terms
  - just add the powers
  - E.g., $x^4 * x^3 \Rightarrow x^7$

    $x^7/x^3 \Rightarrow x^4$

Cont…

- Multiplying Two Polynomials
  - is done term by term
  - E.g.,

$(x^5+x^3+x^2+x)(x^2+x+1)$

$\Rightarrow (x^7+x^6+x^5)+(x^5+x^4+x^3)+(x^4+x^3+x^2)+(x^3+x^2+x)$

$\Rightarrow x^7+x^6+x^3+x$

- Dividing One Polynomial by Another
  - Division of polynomials is conceptually the same as the binary division we discussed for an encoder.

Cont…

- Shifting
  - It requires to create augmented dataword

Shifting left 3 bits: 10011 becomes 10011000 $\quad x^4+x+1$ becomes $x^7+x^4+x^3$

Shifting right 3 bits: 10011 becomes 10 $\quad x^4+x+1$ becomes x

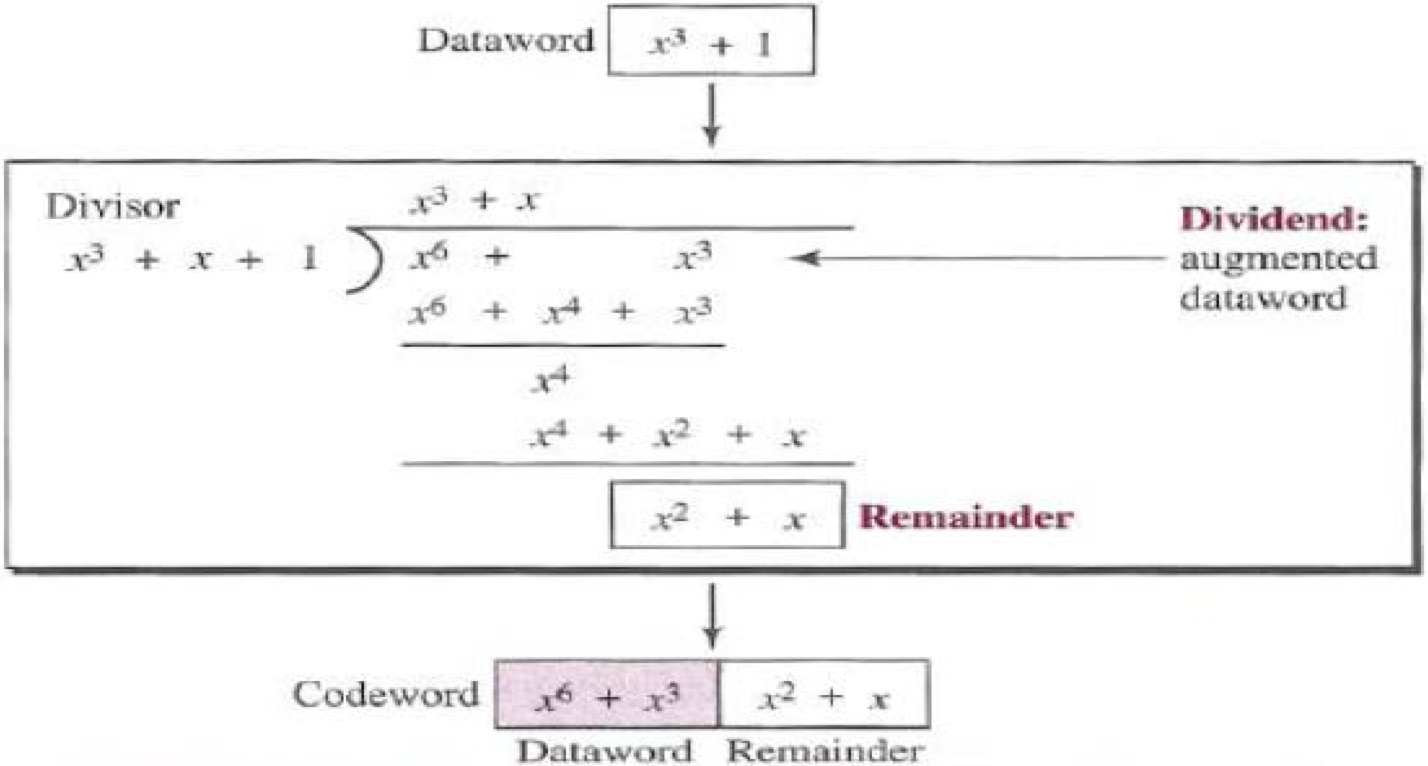
Divisor

$$
\begin{array}{r|l}
 & x^3 + x \\
\hline
x^3 + x + 1 & x^6 + x^3 \\
 & x^6 + x^4 + x^3 \\
\hline
 & x^4 \\
 & x^4 + x^2 + x \\
\hline
 & \boxed{x^2 + x}
\end{array}
$$

Dividend: augmented dataword ($x^6 + x^3$)

Remainder ($x^2 + x$)

Cyclic Code Encoder Using Polynomials



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

Standard Polynomials for CRC

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

Name	*Polynomial*	*Used in*
CRC-8	$x^8 + x^2 + x + 1$ 100000111	ATM header
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$ 11000110101	ATM AAL
CRC-16	$x^{16} + x^{12} + x^5 + 1$ 10001000000100001	HDLC
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ 100000100110000010001110110110111	LANs

Divisor Polynomial Selection

- This depends on the expectation we have from the code.
- Let,
  - Dataword: *d(x)*
  - Codeword: *c(x)*
  - Generator: *g(x)*
  - Syndrome: *s(x)*
  - Error: *e(x)*

- If *s(x) ≠ 0* --> one or more bits is corrupted
- If *s(x)* == 0 --> either no bit is corrupted or the decoder failed to detect any errors

$$\text{Received codeword} = c(x) + e(x)$$

Cont..

$$\frac{\text{Received codeword}}{g(x)} = \frac{c(x)}{g(x)} + \frac{e(x)}{g(x)}$$

- Those errors that are divisible by *g(x)* are not caught.
- A good polynomial generator needs to have the following characteristics:
  - It should have at least two terms.
  - The coefficient of the term x^0 should be 1.
  - It should not divide $x^t + 1$, for t between 2 and $n - 1$.
  - It should have the factor $x + 1$.

Thanks!

Figure and slide materials are taken from the following sources:

1. W. Stallings, (2010), Data and Computer Communications
2. NPTL lecture on Data Communication, by Prof. A. K. Pal, IIT Kharagpur
3. B. A. Forouzan, (2013), Data Communication and Networking