Week 5: Data Link Layer (Cont’d)

EE3017/IM2003 Computer Communications

School of Electrical and Electronic Engineering

Dr. SHAO Xuguang, Michelle

Email: XGSHAO@ntu.edu.sg

Room: S1-B1a-10

Phone: 6513-7648

Topic Outline

Introduction to Computer Communications 01

Data Communications Fundamentals 02

Data Link Layer 03

Overview, Framing and Stuffing, Flow Control, **Error Control**

Application

Transport

Network

Link

Physical

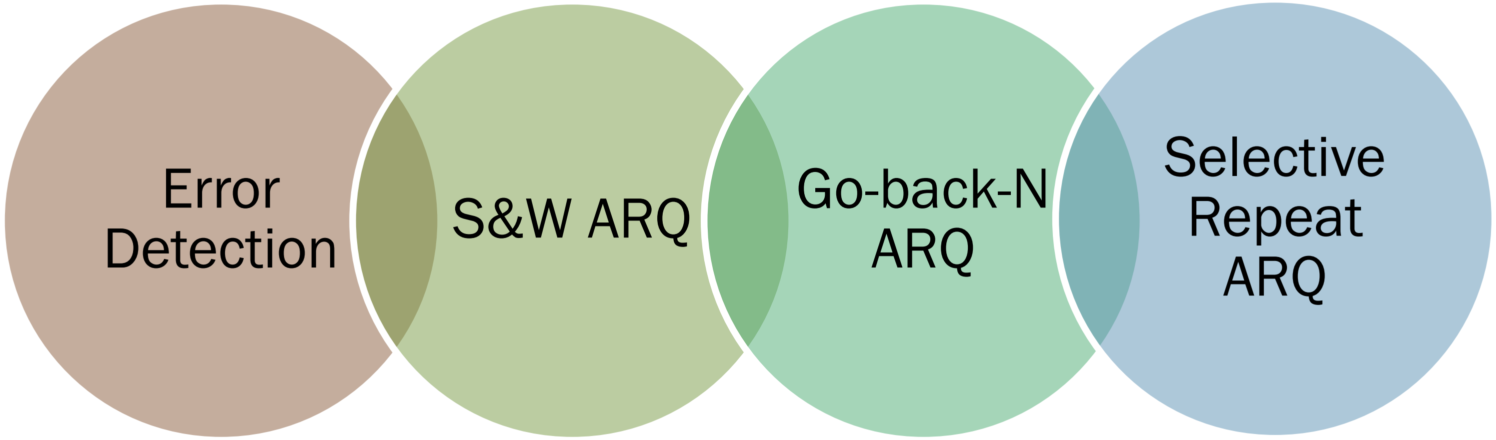
Learning Objectives

By the end of this topic, you should be able to:

- Explain the need for error control in general.
- Explain the difference between error detection and error correction.
- Explain the error detection principles and the roles that are played by the transmitter and receiver in error detection.
- Apply cyclic redundancy check (CRC) technique in doing error checking.
- Evaluate various generator polynomials.

Error Control Overview

Error Control Overview



Data Link Layer Overview

- Data Link operates over wire-like, directly-connected systems.
- Frames can be corrupted or lost, but arrive in order.
- Data link performs error-checking and retransmission.

- Detection:
    - Lost frame or Damaged frame

- Automatic Repeat Request (ARQ) function is based on some or all of the following ingredients:
    - Error detection
    - Positive acknowledgment
    - Retransmission after timeout
    - Negative acknowledgement and retransmission

Types of Errors

An error occurs when a bit is altered between its transmission and reception.

E.g. send 0/receive 1 or send 1/receive 0.

Single bit errors

- Randomly selected bits are altered.
- Bit errors are **independent events**, i.e. error in a bit does not affect the probability of any other bit being in error.
- Mostly caused by White Noise.

Burst errors

- Length *B.*
- Contiguous sequence of *B* bits in which the first bit, the last bit and any number of intermediate bits are in error.
- Caused by Impulse Noise and/or Fading in wireless channels.
- Effect is greater at higher data rates.

Error Detection VS. Error Correction

Correction of errors is more difficult than the detection.

Error Detection

- Check if any error has occurred (Yes/No).
- No need to know how many bits are in error or where the bits are.

Error Correction (out of scope)

- Need to correct the error or reproduce the packet immediately.
- Must know the exact number of bits that are corrupted and their locations.
- Used when error detection and retransmission of corrupted and lost packets is not useful.
- Real-time multimedia transmission.

Error Detection

Problem:

Detect bit errors in frames/packets/segments.

Solution:

- Transmitter adds extra (redundant) error detecting code (checkbits) to data bits to form the codeword for transmission.
- Receiver recalculates based on received codeword.

Goals:

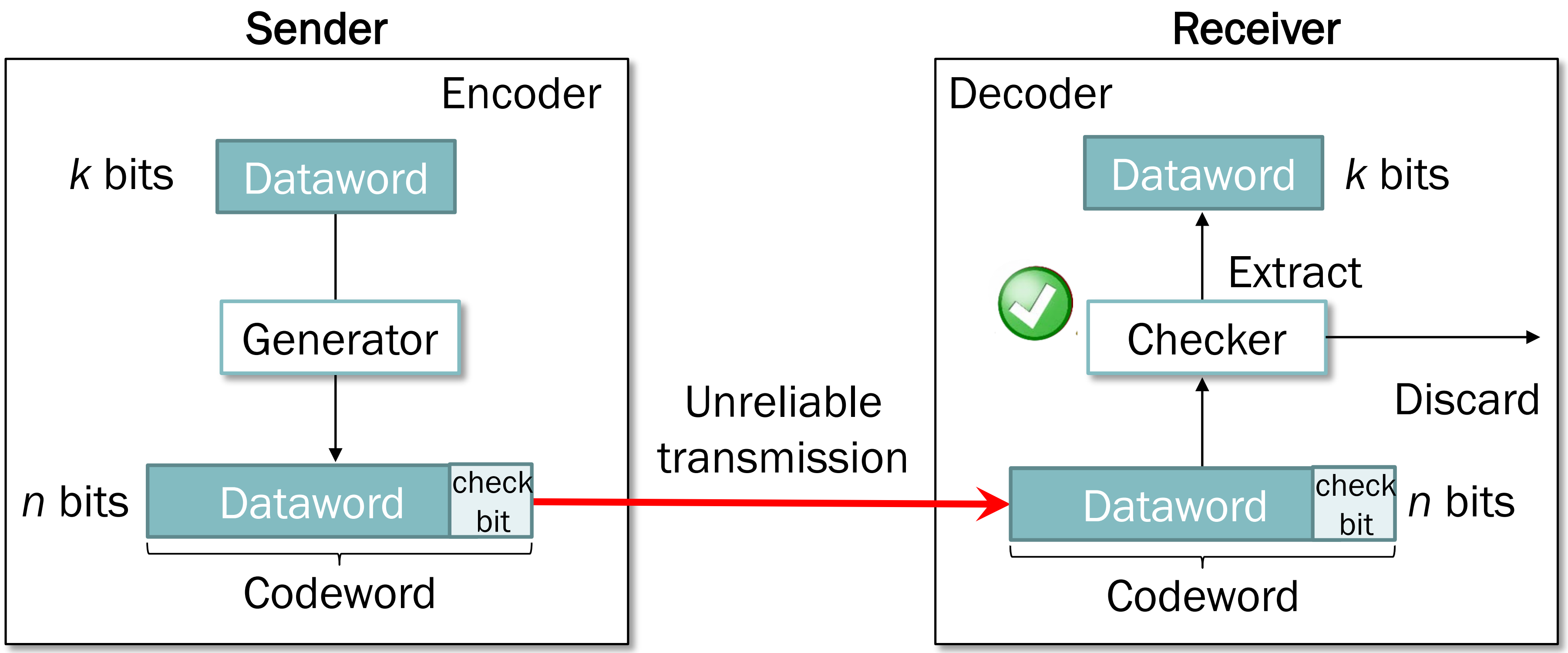
- Reduce overhead, i.e., reduce the number of redundant bits.
- Increase the number and the type of bit error patterns that can be detected.

Examples:

- Parity Checking
- Internet Checksum
- Cyclic Redundancy Check (CRC)

Block Coding and Error Detection

- Divide a message into blocks with the same bit lengths called Dataword or Information (k bits).
- Generate Codeword (n bits) by adding (appending) redundancy or checkbits (r = n - k) to each block.



Not all the error can be detected!

If the following two conditions are met, the receiver **CAN** detect a change in the original codeword:

- The receiver can find a list of valid codewords.
- The original codeword has changed to an invalid one.

Original message 11 10 01 00

❑ Transmit with redundancy

1111 1010 0101 0000

e.g. checkbits are duplicates of the dataword

1011 1010 1101 0000

Invalid patterns indicate errors, **CAN** detect

If an error changes a valid codeword to <u>another</u> valid codeword, then this error **CANNOT** be detected.

1111 1010 0101 1010

An error **CANNOT** be detected

Cyclic Redundancy Check (CRC)

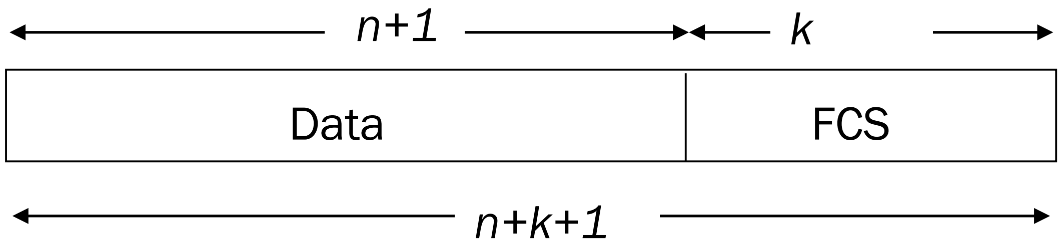
Polynomial Codes

- Considering not only the value but also the order.
- Polynomials for codewords.
- Polynomial arithmetic.
- Also called cyclic redundancy check (CRC) codes.
- Most data communications standards use polynomial codes for error detection.
- Polynomial codes are also the basis for powerful error-correction methods.

Cyclic Redundancy Check (CRC) Code

- Widely used in networks such as local area networks and wide area networks.

- For a block of $n + 1$ data bits, one generates an $n + k + 1$ bit sequence, called 'codeword' (or 'frame'), for transmission.



- Additional k bits appended to the data block as a frame check sequence (FCS). These k bits are generated in such a way that the binary polynomial $T(x)$ corresponding to the $n + k + 1$ bit codeword is exactly divisible by a specially chosen polynomial $C(x)$.

- Receiver divides the binary polynomial for the incoming frame (i.e. the received frame) by the same $C(x)$.
  - If no remainder, then it assumes "No Error in Received Frame".
  - Otherwise there would be "One or more errors somewhere in the frame".

Binary Polynomial Operations

Binary Polynomial Arithmetic

Binary vectors map to polynomials

$$(i_{k-1}, i_{k-2}, \dots, i_2, i_1, i_0) \rightarrow i_{k-1}x^{k-1} + i_{k-2}x^{k-2} + \dots + i_2x^2 + i_1x + i_0$$

Addition:

$$(x^7 + x^6 + 1) + (x^6 + x^5) = x^7 + x^6 + x^6 + x^5 + 1$$

Addition = subtraction = XOR

$$= x^7 + (\cancel{1+1})x^6 + x^5 + 1 \quad \text{since } (1+1)\text{mod}2 = 0$$

$$= x^7 + x^5 + 1$$

Multiplication:

$$(x + 1)(x^2 + x + 1) = x(x^2 + x + 1) + 1(x^2 + x + 1)$$

$$= (x^3 + \cancel{x^2} + \cancel{x}) + (\cancel{x^2} + \cancel{x} + 1)$$

$$= x^3 + 1$$

Binary Polynomial Division

Division with Decimal Numbers

$$
\begin{array}{rrl}
 & 34 & \leftarrow \text{quotient} \\
\text{divisor} \rightarrow 35\ ) & 1222 & \leftarrow \text{dividend} \\
 & \underline{105} & \\
 & 172 & \\
 & \underline{140} & \\
 & 32 & \quad \text{remainder}
\end{array}
$$

$\boxed{\text{dividend = quotient x divisor + remainder}}$

1222 = 34 x 35 + 32

Polynomial Division

$$
\begin{array}{rl}
 & x^3 + x^2 + x \qquad = A(x) \text{ quotient} \\
x^3 + x + 1\ ) & \cancel{x^6} + x^5 \\
 & \underline{\cancel{x^6} + \quad x^4 + x^3} \\
 & \quad \cancel{x^5} + x^4 + \cancel{x^3} \\
 & \quad \underline{\cancel{x^5} + \quad \cancel{x^3} + x^2} \\
 & \qquad \cancel{x^4} + \quad \cancel{x^2} \\
 & \qquad \underline{\cancel{x^4} + \quad \cancel{x^2} + x} \\
 & \qquad\qquad \boxed{x = R(x) \text{ remainder}}
\end{array}
$$

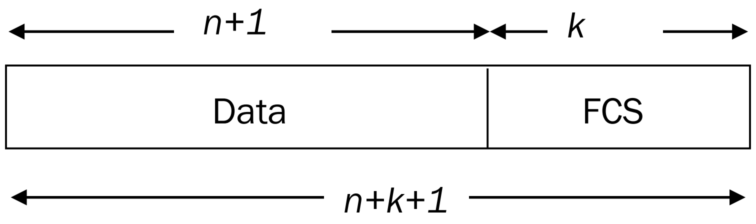
Divisor

Dividend

Note: Degree of R(x) is less than degree of divisor

CRC

- Represent an (n + 1) -bit message as an n-degree polynomial *M(x)*
  - e.g. 10101101



$\rightarrow 1.x^7 + 0{\cdot}x^6 + 1.x^5 + 0{\cdot}x^4 + 1.x^3 + 1.x^2 + 0{\cdot}x + 1.x^0 = x^7 + x^5 + x^3 + x^2 + x^0$

- Choose a divisor *k-degree polynomial C(x)* **Generator polynomial**

- Compute remainder *R(x)* from $\boxed{M(x) * x^k \,/\, C(x)}$

$$M(x) \times x^k = A(x) \times C(x) + R(x)$$

where $\boxed{Degree(R(x)) < k}$

- Let

$$\boxed{T(x) = M(x) \times x^k - R(x) = A(x) \times C(x)}$$

- Then
  - Codeword *T(x)* is divisible by *C(x)*.
  - First *n + 1* bits of *T* (binary representation of *T(x)*) represent *M*, last *k* bits represent the CRC.

CRC

$T(x)$: transmitted codeword

$T'(x)$: received codeword

Sender	Receiver
• Compute and send $T(x)$, i.e., the coefficients of $T(x)$.	• Let $T'(x)$ be the *(n + k)*-degree polynomial generated from the received message. • $C(x)$ divides $T'(x)$? ▫ No ⟶ error detected ▫ Yes ⟶ no errors (or has undetectable errors)

Note:
All computations are binary modulo 2 operation or binary polynomial operation.

Binary Modulo 2 Operations

Binary Modulo 2 Arithmetic

- Like binary arithmetic but without borrowing/carrying from/to adjacent bits.
- Both Modulo 2 addition and subtraction are the same as XOR.

Addition = subtraction = XOR

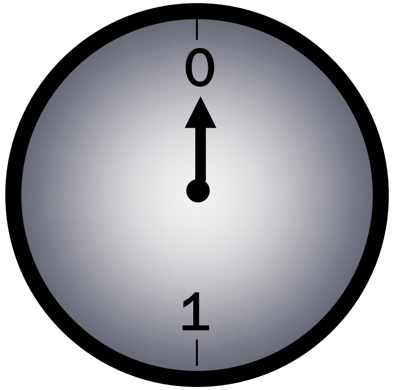
$0 \pm 0 = 0 \bmod 2 = 0$

$0 \pm 1 = \pm 1 \bmod 2 = 1$

$1 \pm 0 = 1 \bmod 2 = 1$

$1 \pm 1 = 2{,}0 \bmod 2 = 0$

INPUT A	INPUT B	OUTPUT A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



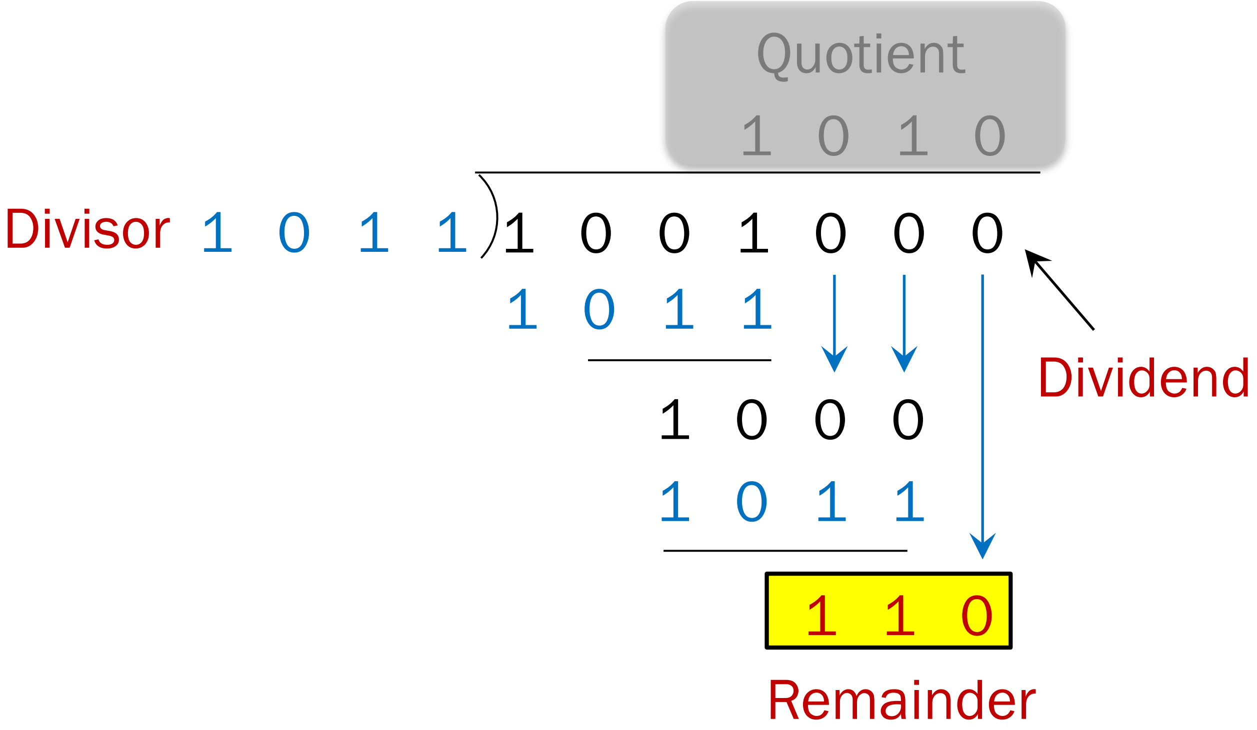
Example: Binary Modulo 2 Arithmetic

Note:

Multiply: AND

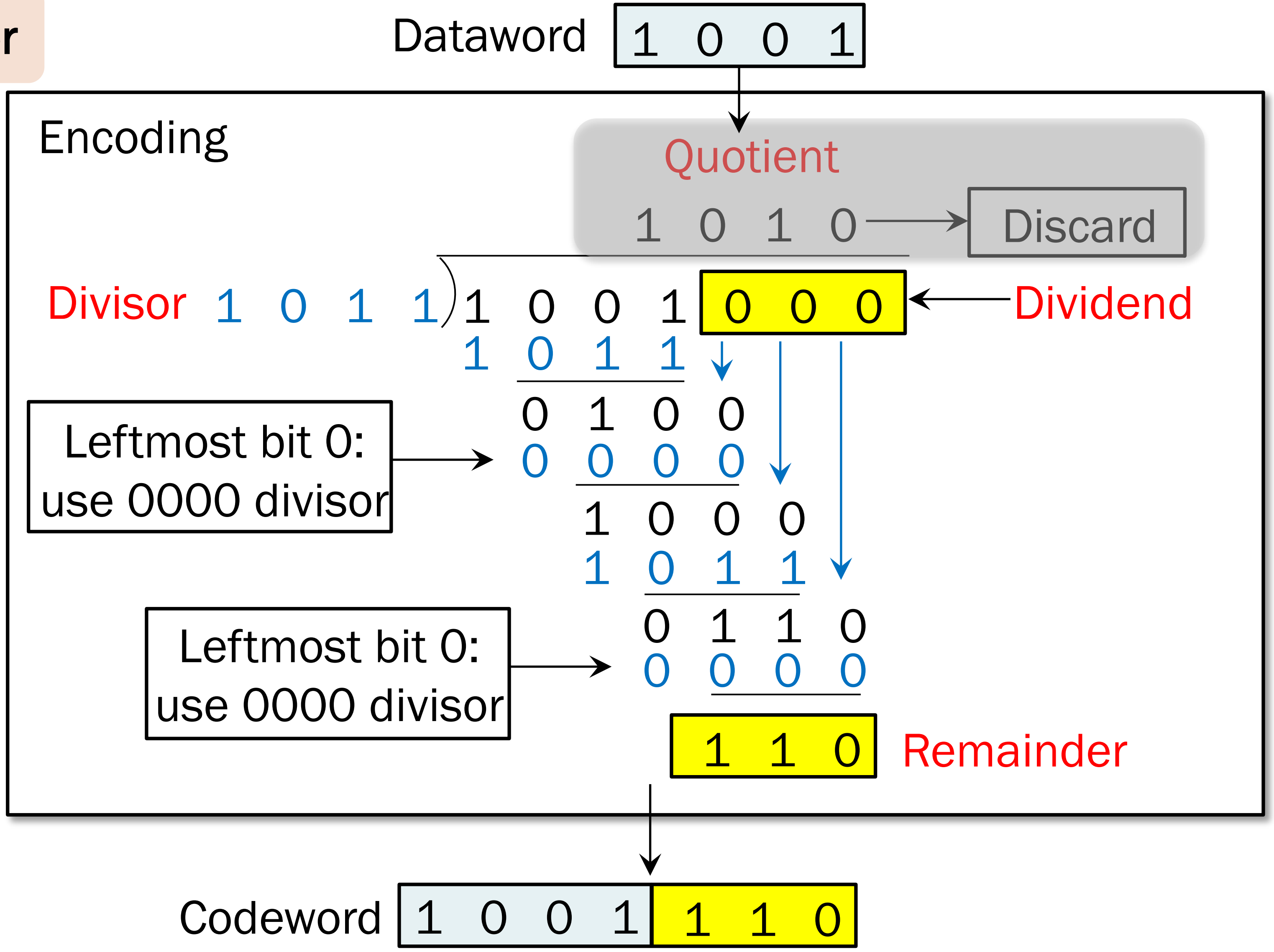
Subtract: XOR

INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



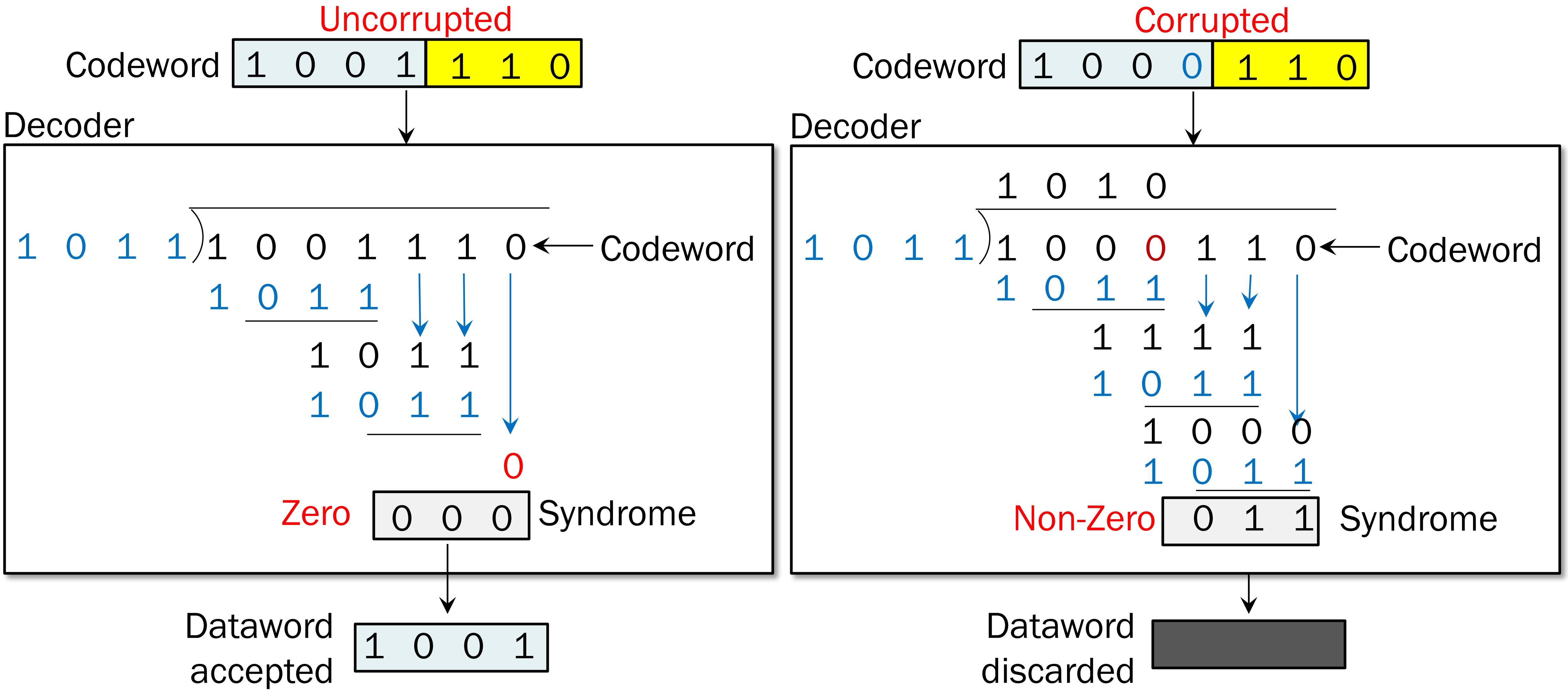
Example: CRC using Binary Modulo 2

At Encoder in sender



Note:
Multiply: AND
Subtract: XOR

CRC at Decoder in Receiver



Exercise 1:

Generator polynomial: $C(x) = x^3 + x + 1$ *In binary: 1011*

Information: (1, 1, 0, 0) $i(x) = x^3 + x^2$

Encoding: $x^3 i(x) = x^6 + x^5$

$$
\begin{array}{r|l}
 & x^3 + x^2 + x \\
\hline
x^3 + x + 1 & x^6 + x^5 \\
 & x^6 + x^4 + x^3 \\
\hline
 & x^5 + x^4 + x^3 \\
 & x^5 + x^3 + x^2 \\
\hline
 & x^4 + x^2 \\
 & x^4 + x^2 + x \\
\hline
 & \boxed{x}
\end{array}
$$

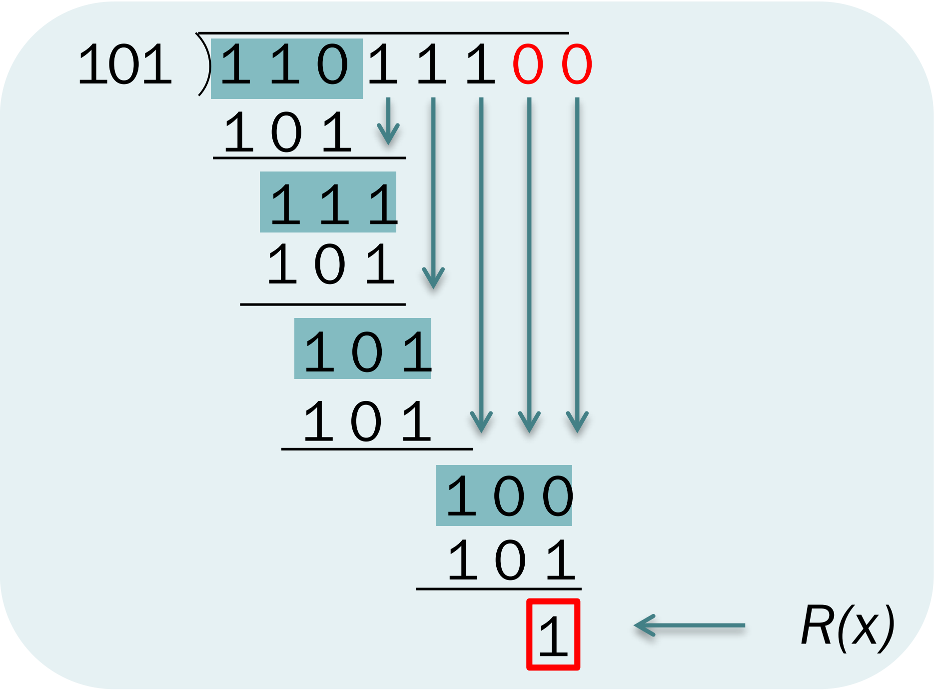
$$
\begin{array}{r|l}
 & \cancel{1110} \\
\hline
1011 & 1100000 \\
 & 1011 \\
\hline
 & 1110 \\
 & 1011 \\
\hline
 & 1010 \\
 & 1011 \\
\hline
 & \boxed{10}
\end{array}
$$

Transmitted codeword:

$T(x) = x^6 + x^5 + x \Rightarrow T = (1, 1, 0, 0, \boxed{0, 1, 0})$

Exercise 2: (Sender Operation)

- Send packet 110111, choose $C(x) = 101$
  - o $k = 2$, $M(x)*x^k \rightarrow$ 11011100
- Compute the remainder $R(x)$ of $M(x)*x^k/C(x)$

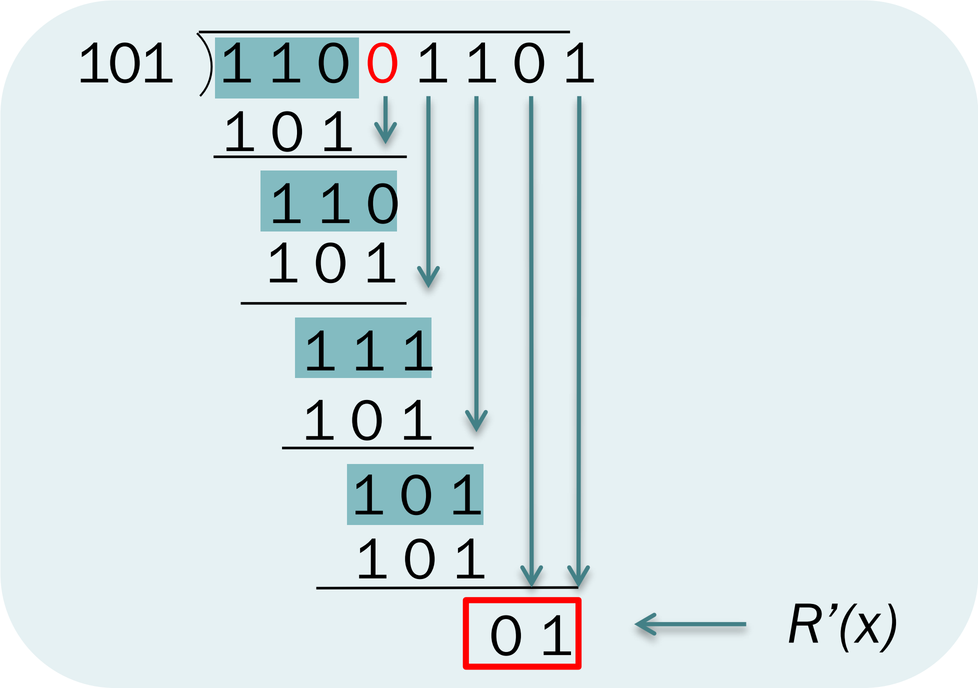


- Compute $T(x) = M(x)*x^k - R(x) \rightarrow$ 11011100 XOR 1 = 11011101
- Send $T(x)$

Exercise 2: (Receiver Operation)

- Assume $T(x) = 11011101$
  - o $C(x)$ divides $T'(x) \longrightarrow$ no errors

- Assume $T'(x) = 11001101$
  - o Remainder $R'(x) = 1 \longrightarrow$ error!



Error Detection

Error Detection Principle

- Consider a situation where the codeword transmitted is $T(x)$ and the receiver receives $T^*(x)$, then the error pattern is

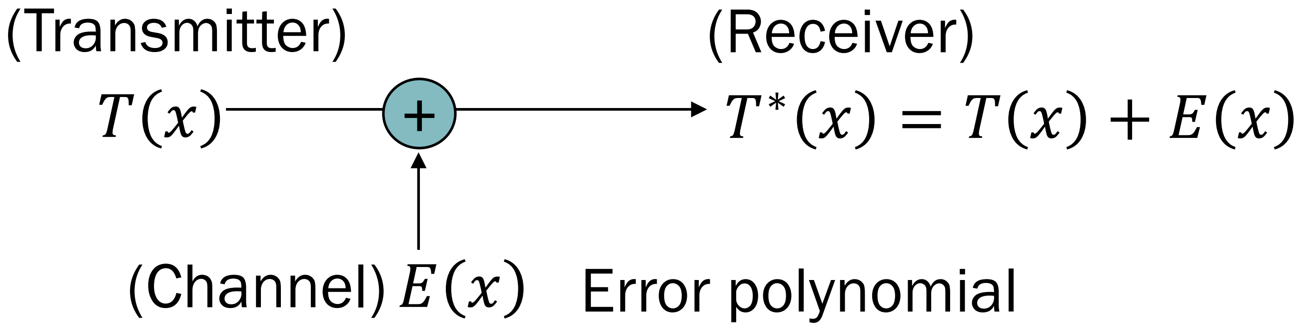
$$E(x) = T^*(x) - T(x) = T(x) - T^*(x)$$

Bit-wise XOR

  - Every term of $E(x)$ refers to a bit of the frame that has been inverted during transmission.

- The error is not detectable if $C(x)$ divides both $T(x)$ and $T^*(x)$. In other words, $C(x)$ divides $E(x)$.

- On the other hand, if $C(x)$ does not divide $E(x)$, the error is detectable.



Remainder of a Power of x

Example:

Please show that $E(x)=x^8+x^2$ is NOT detectable

$$rem\left[\frac{x^8}{g(x)}\right] + rem\left[\frac{x^2}{g(x)}\right] = x^2 + x^2 = 0$$

Periodic pattern appears as the power of x increases!

Divisor: $g(x) = x^4 + x^2 + 1$

Dividend		Remainder	
Bin	Poly	$rem\, x^i/g(x)$	Bin
1	x^0	x^0	1
10	x^1	x^1	10
100	x^2	$\boxed{x^2}$	100
1000	x^3	x^3	1000
10000	x^4	x^2+1	101
100000	x^5	x^3+x	1010
1000000	x^6	x^0	1
10000000	x^7	x^1	10
100000000	x^8	$\boxed{x^2}$	100
1000000000	x^9	x^3	1000
10000000000	x^{10}	x^2+1	101

Primitive Polynomial

Divisor: $g(x) = x^3+x+1$

Dividend		Remainder	
Bin	Poly	$rem\, x^i/g(x)$	Bin
1	x^0	x^0	1
10	x^1	x^1	10
100	x^2	x^2	100
1000	x^3	$x+1$	11
10000	x^4	x^2+x	110
100000	x^5	x^2+x+1	111
1000000	x^6	x^2+1	101
10000000	x^7	x^0	1
100000000	x^8	x^1	10
1000000000	x^9	x^2	100
10000000000	x^{10}	$x+1$	11

For given divisor g(x) with degree of m, the remainder table of a power of x has a repeating pattern with period $L \leq 2^m-1$

$$rem\left[\frac{x^i}{g(x)}\right] = rem\left[\frac{x^{i+L}}{g(x)}\right]$$

A primitive polynomial p(x) gives largest $L=2^m-1$, where m is the degree of p(x).

For example x^3+x+1, degree m=3, the remainder of a power of x has a repeating pattern with period $2^3-1=7$, so [x^3+x+1] is [primitive polynomial]

Standard Generator Polynomials

CRC-8	$= x^8 + x^2 + x + 1$	ATM
CRC-16	$= x^{16} + x^{15} + x^2 + 1$ $= (x+1)(x^{15} + x + 1)$	Bisync
CCITT-16	$= x^{16} + x^{12} + x^5 + 1$	HDLC
CCITT-32	$= x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11}$ $+ x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	IEEE 802

Summary

Summary

Key points discussed in this topic:

- One of the services offered by data link is error control.
- An error occurs when a bit is altered between its transmission and reception the frame.
- Block coding divides a message into blocks with the same bit lengths called Dataword or Information and generates a Codeword by adding redundancy or checkbits to each block.
- Cyclic redundancy check (CRC) codes are widely used in networks such as local area networks and wide area networks for error detecting.
- The calculation for CRC can be based on:
  - Binary Polynomial Operations
  - Binary Modulo 2 Operations