



# Frame Error Detection

---

รศ.ดร.อนันต์ ผลเพิ่ม

Asso. Prof. Anan Phonphoem, Ph.D.

[anan.p@ku.ac.th](mailto:anan.p@ku.ac.th)

<http://www.cpe.ku.ac.th/~anan>

Computer Engineering Department

Kasetsart University, Bangkok, Thailand

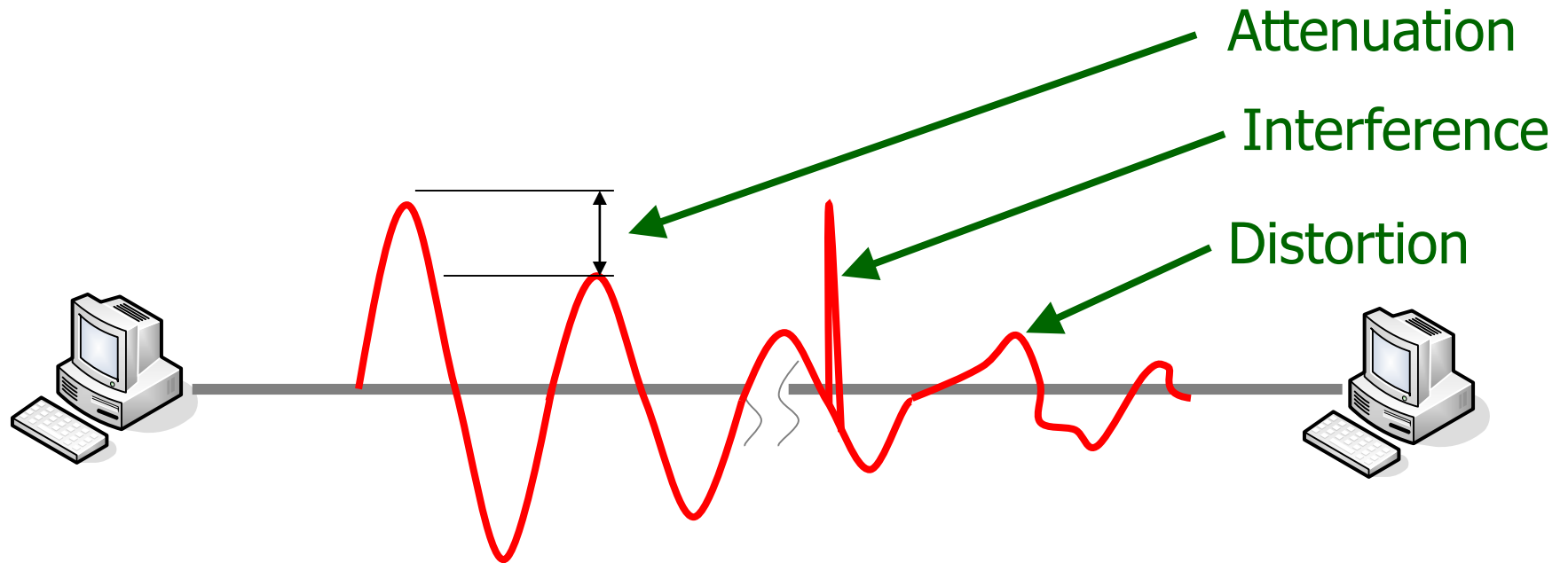


# Outline

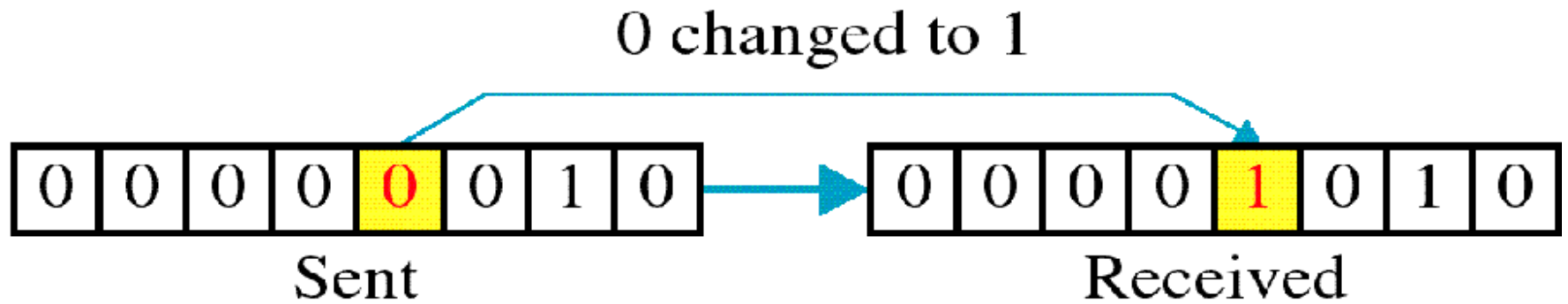
---

- Transmission Errors
- Error Detection Mechanism

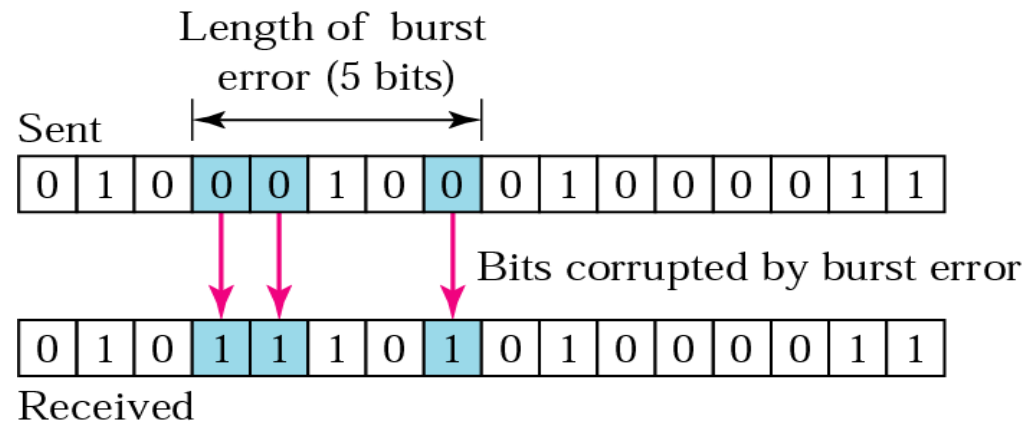
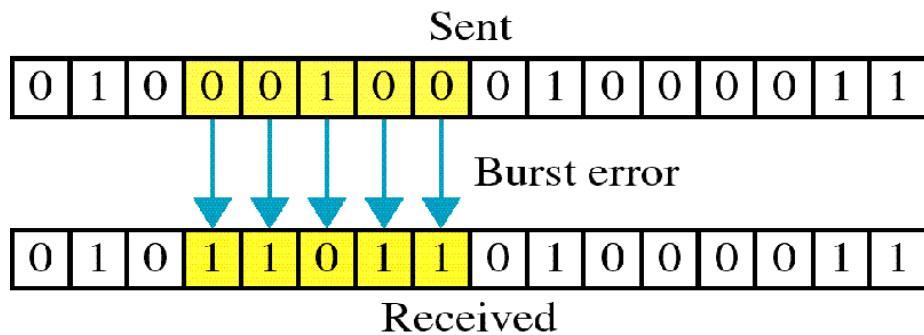
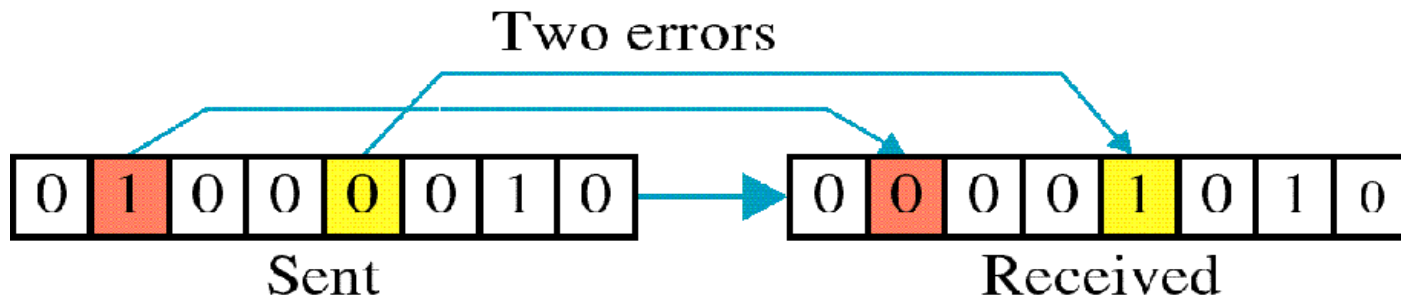
# Transmission Errors



# Single-bit error



# Burst error (Multiple-bit error)



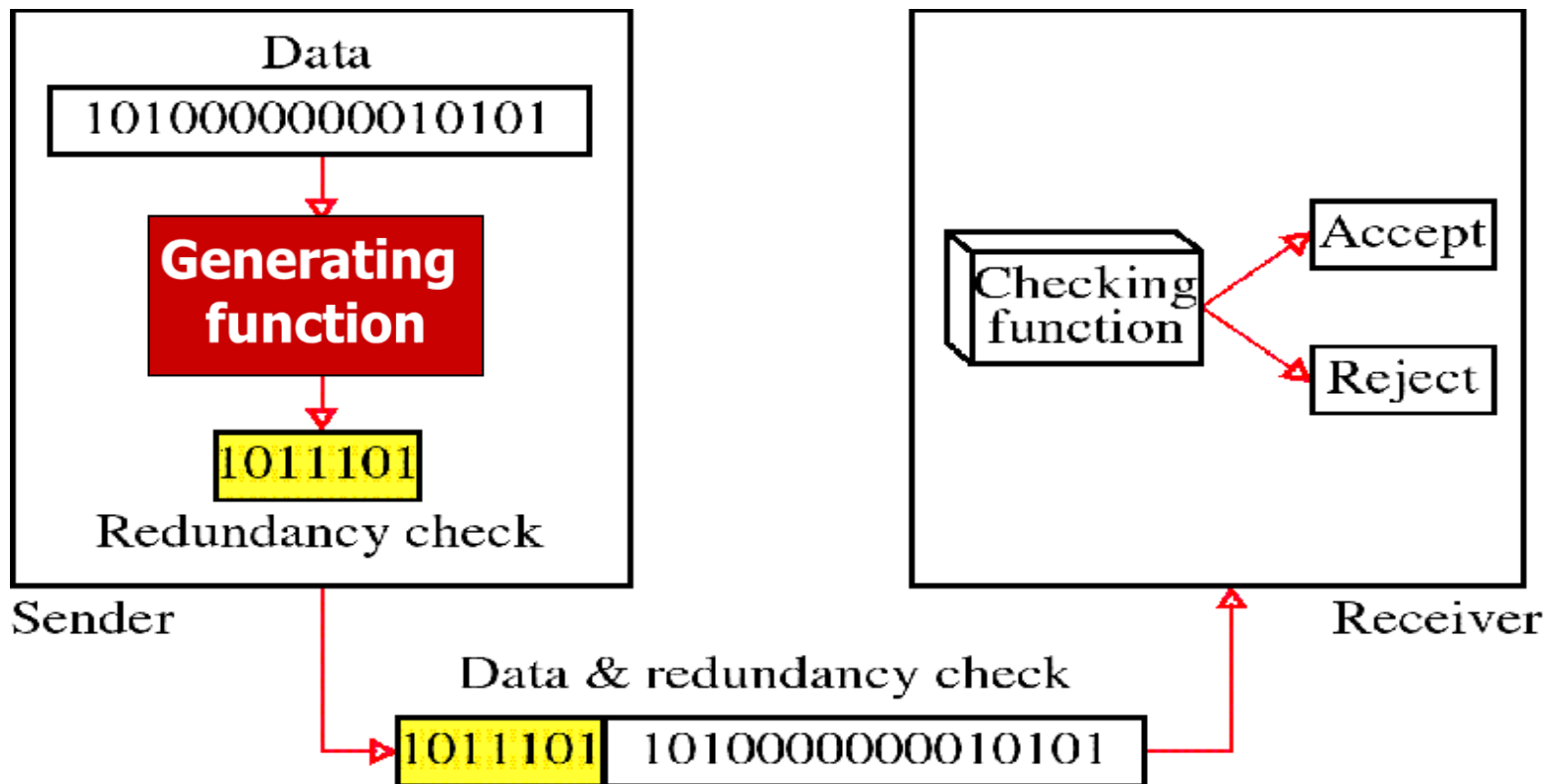


# How do we know that data is corrupted (errors occur)?

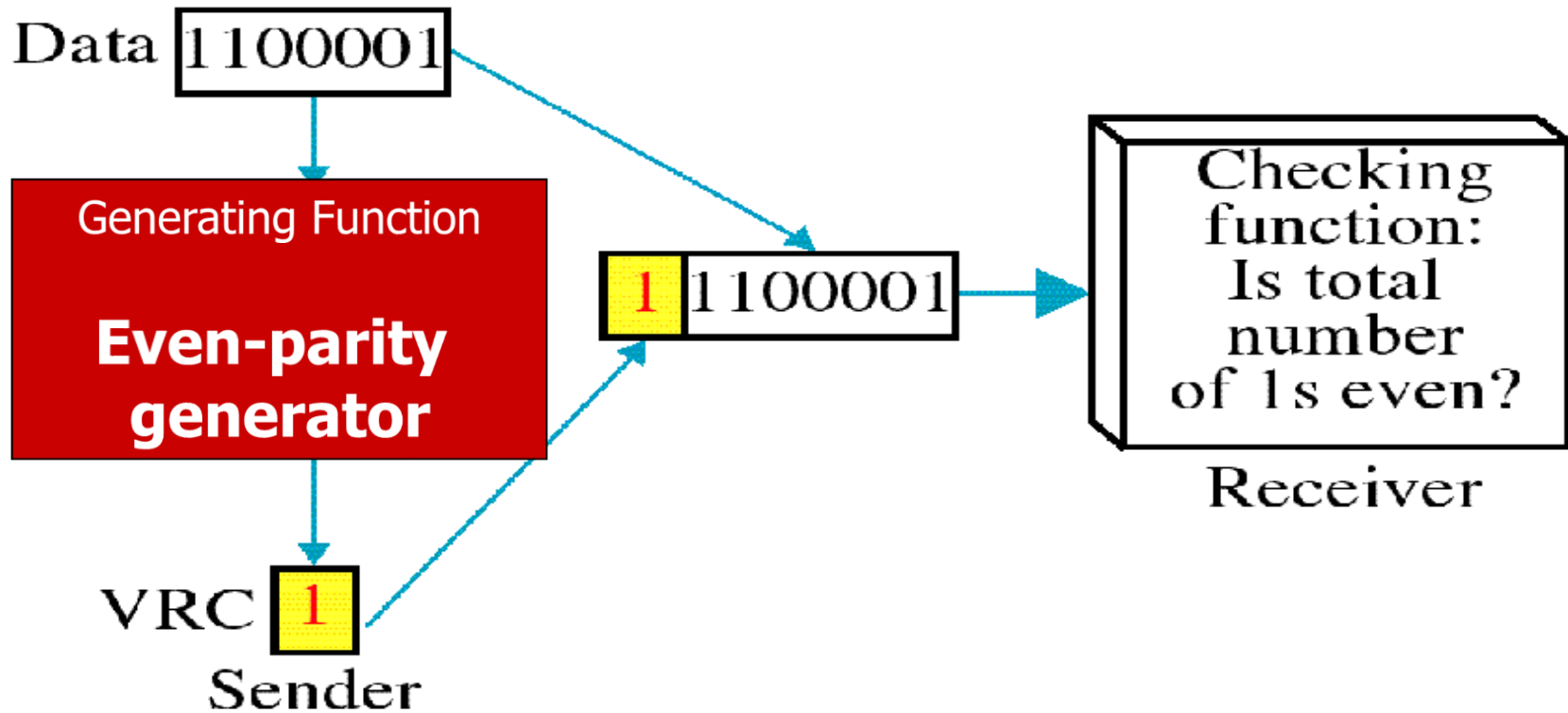
---

- → How to detect the error?
- Does not know at the signal level
- Comparison...
- → Redundancy
  - 2 copies of everything (slow, high overhead)
  - Extra info.

# Redundancy



# Parity Check







# Example

---

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

**1110111 1101111 1110010 1101100 1100100**

w                    o                    r                    l                    d

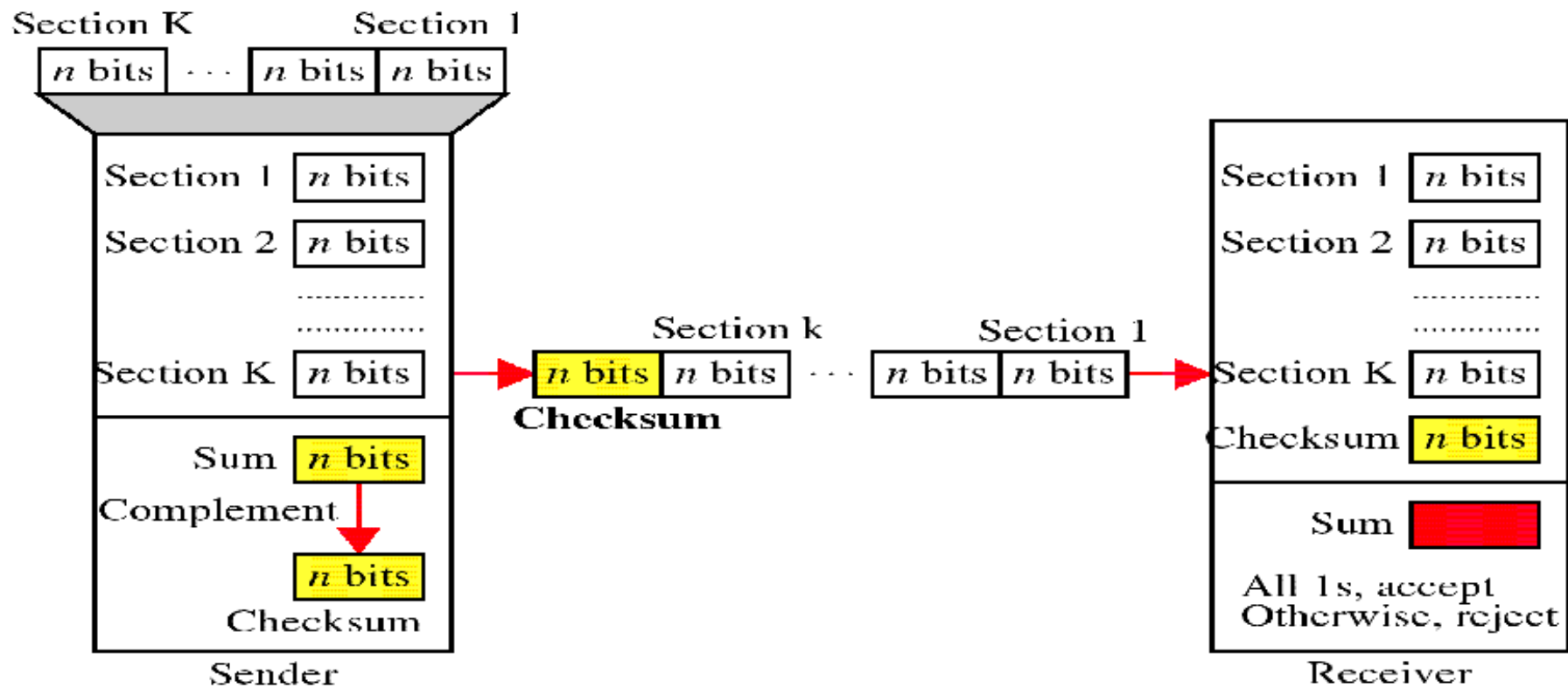
The following shows the actual bits sent

11101110 11011110 11100100 11011000 11001001

# Checksum

Notes:

- Segment
- Add using one's complement
- Sum is complemented





# 16-bit checksum example

---

H	e	l	l	o		w	o	r	l	d	.
48	65	6C	6C	6F	20	77	6F	72	6C	64	2E

$$4865 + 6C6C + 6F20 + 776F + 726C + 642E + \text{carry} = 71FC$$

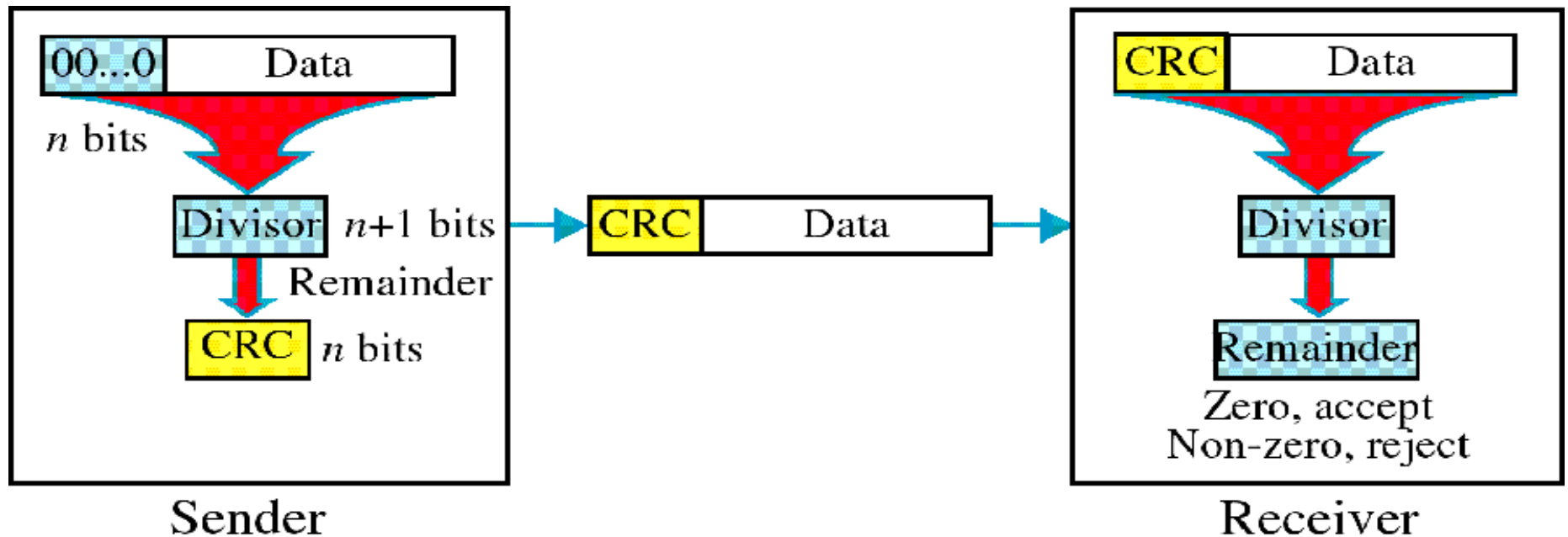


# Cannot detect (all) errors

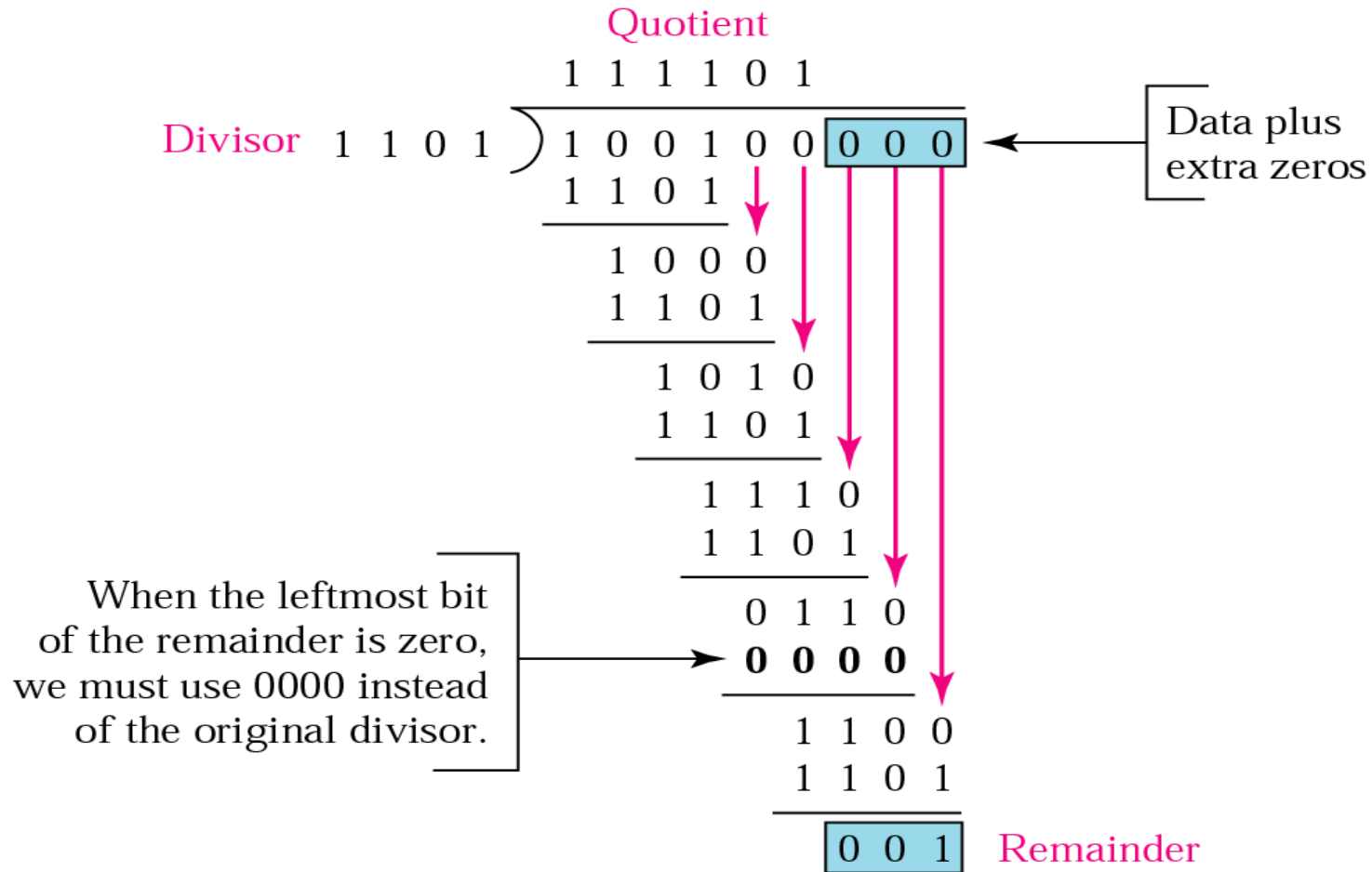
---

<b>Data Item In Binary</b>	<b>Checksum Value</b>	<b>Data Item In Binary</b>	<b>Checksum Value</b>
0001	1	0011	3
0010	2	0000	0
0011	3	0001	1
0001	1	0011	3
<b>totals</b>	<b>7</b>		<b>7</b>

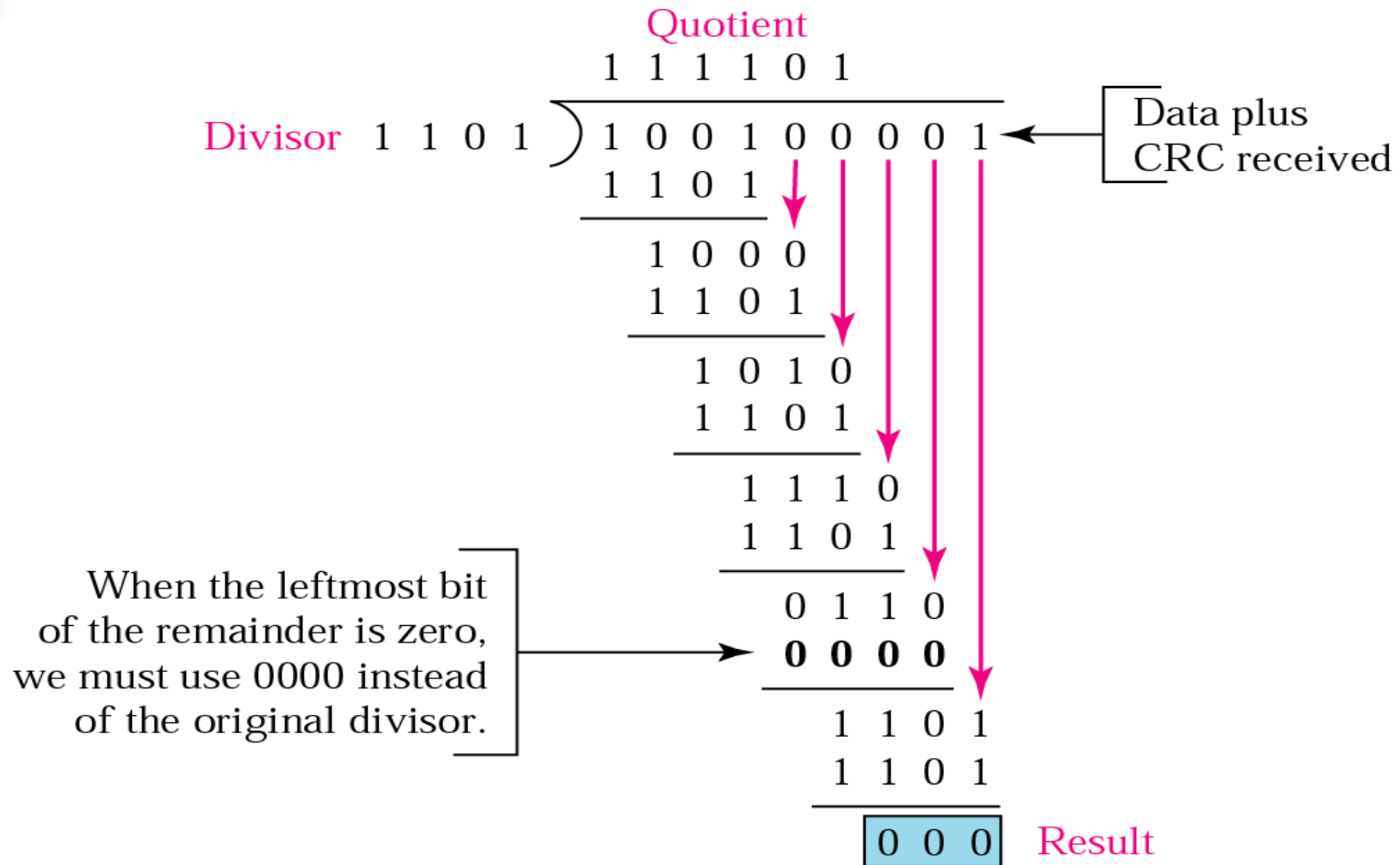
# Cyclic Redundancy Check (CRC)



# Binary Division in a CRC generator



# Binary Division in a CRC checker





# Polynomial

---

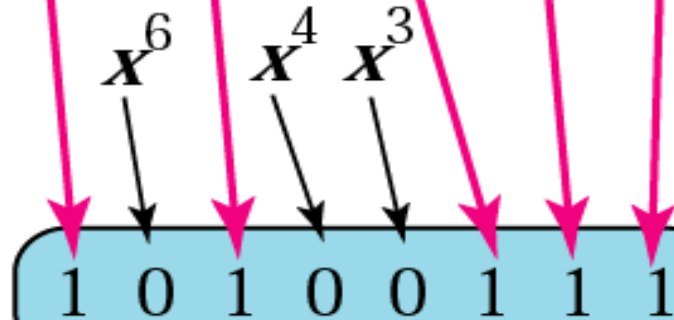
$$x^7 + x^5 + x^2 + x + 1$$



# Polynomial and Divisor

Polynomial

$$x^7 + x^5 + x^2 + x + 1$$



Divisor



# Choosing the polynomial

2 properties

- Should not be divisible by  $x$
- Should be divisible by  $(x+1)$

## Example

- Cannot choose  $x$  (binary 10) or  $x^2 + x$  (binary 110)
  - Because both are divisible by  $x$ .
- Can choose  $x + 1$  (binary 11)
  - Because it is not divisible by  $x$ , but is divisible by  $x + 1$ .
- Also choose  $x^2 + 1$  (binary 101)
  - Because it is divisible by  $x + 1$  (binary division).

# Standard Polynomials

Name	Polynomial	Application
<b>CRC-8</b>	$x^8 + x^2 + x + 1$	<b>ATM header</b>
<b>CRC-10</b>	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$	<b>ATM AAL</b>
<b>ITU-16</b>	$x^{16} + x^{12} + x^5 + 1$	<b>HDLC</b>
<b>ITU-32</b>	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	<b>LANs</b>



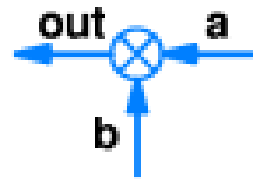
# CRC

---

- Very effective detection method
- Can detect all burst of length  $<$  degree of polynomial
- Detect other errors  $\rightarrow$  high prob.

# CRC implementation

Ex-OR

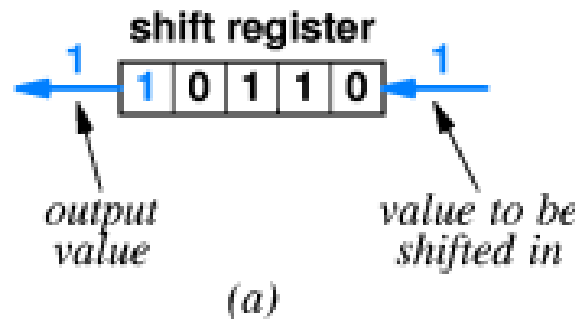


(a)

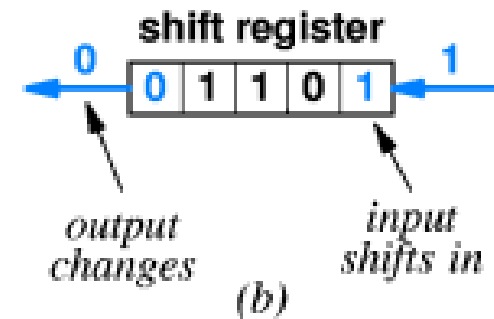
a	b	out
0	0	0
0	1	1
1	0	1
1	1	0

(b)

Shift Register

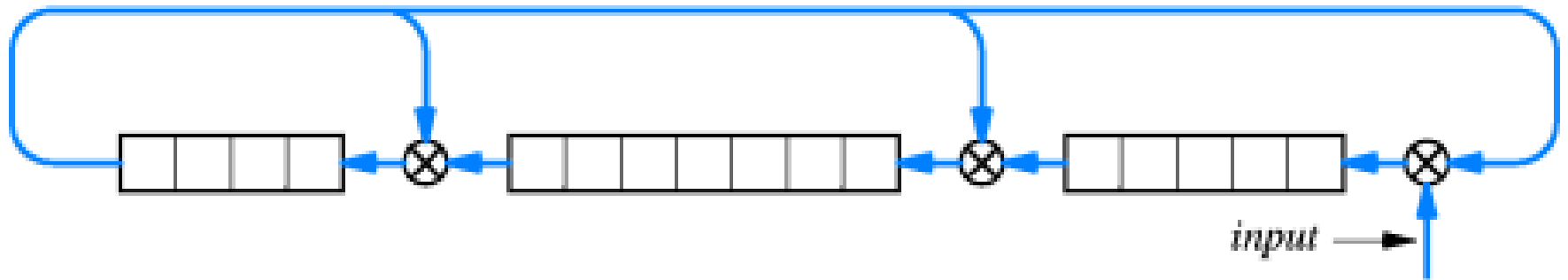


(a)



(b)

# 16-bit CRC Hardware Implementation



# Data Frame





# Summary

---

- Transmission Errors
- Error Detection Mechanism
  - Parity
  - Checksum
  - CRC