Data Communication

DIGITAL DATA TRANSMISSION TECHNİQUES

Yrd.Doç.Dr. Murat İSKEFİYELİ

Digital Data Transmission Techniques

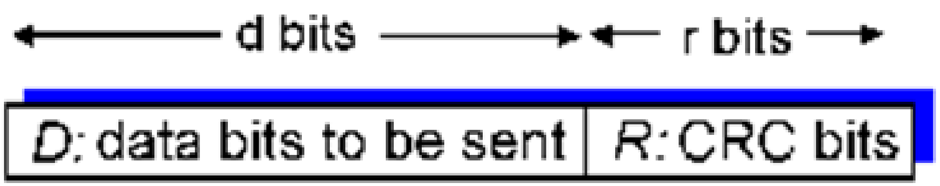
- Transmission Technique (Serial, Parallel)
- Error Detection Techniques
- Error Correction Techniques
- Connection Interface Standarts

Cyclic Redundancy Check

- one of most common and powerful checks
- for block of k bits transmitter generates an n bit frame check sequence (FCS)
- transmits $k+n$ bits which is exactly divisible by some number
- receiver divides frame by that number
  - if no remainder, assume no error

CRC

- based on r+1 bit pattern (generator) G known to tx and rx
- treat data bits D as a binary number
- obtain r-bit CRC R such that <D,R> divisible (modulo 2) by G
- receiver divides <D,R> by G. non-zero remainder implies error
- can detect all burst errors less than r+1 bits
- widely used in practice (ATM, HDCL)



bit pattern

$$D * 2^r \quad \text{XOR} \quad R$$

mathematical formula

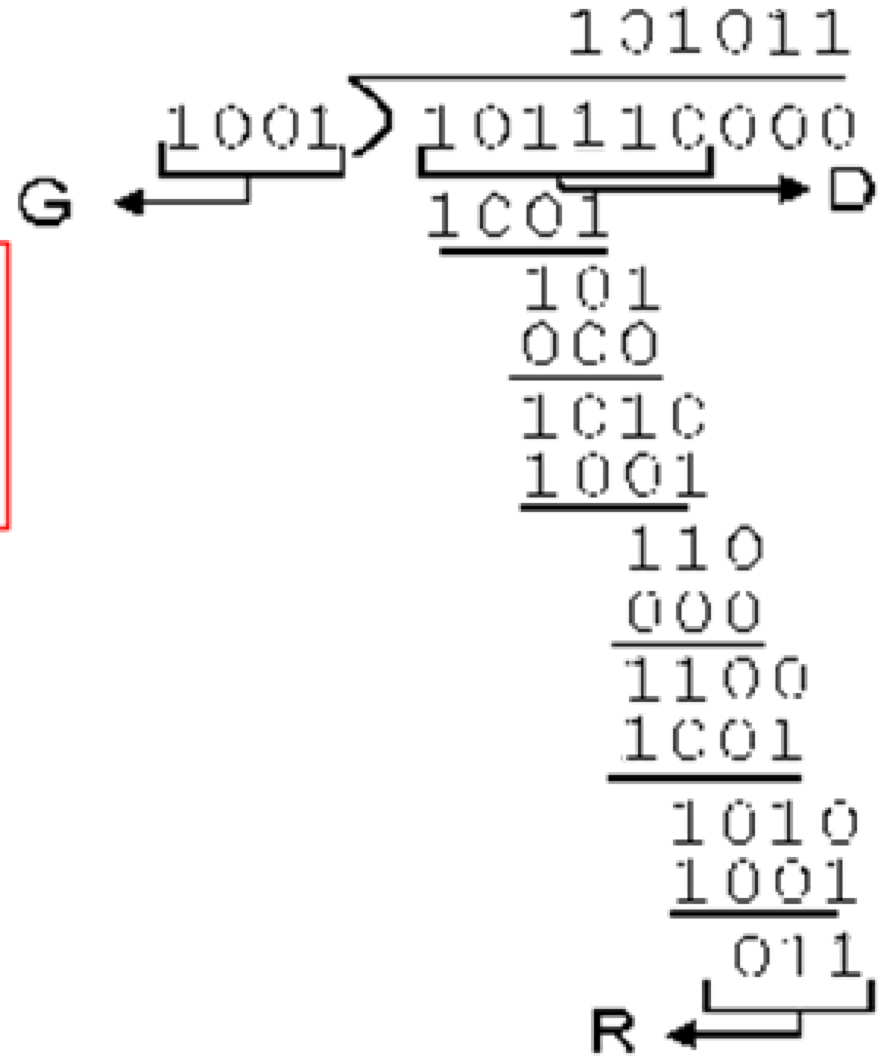
$$R = \text{remainder}\left[\frac{D \cdot 2^r}{G}\right]$$

CRC-Example

$G = 1001 \quad (r=3)$

$D = 101110$

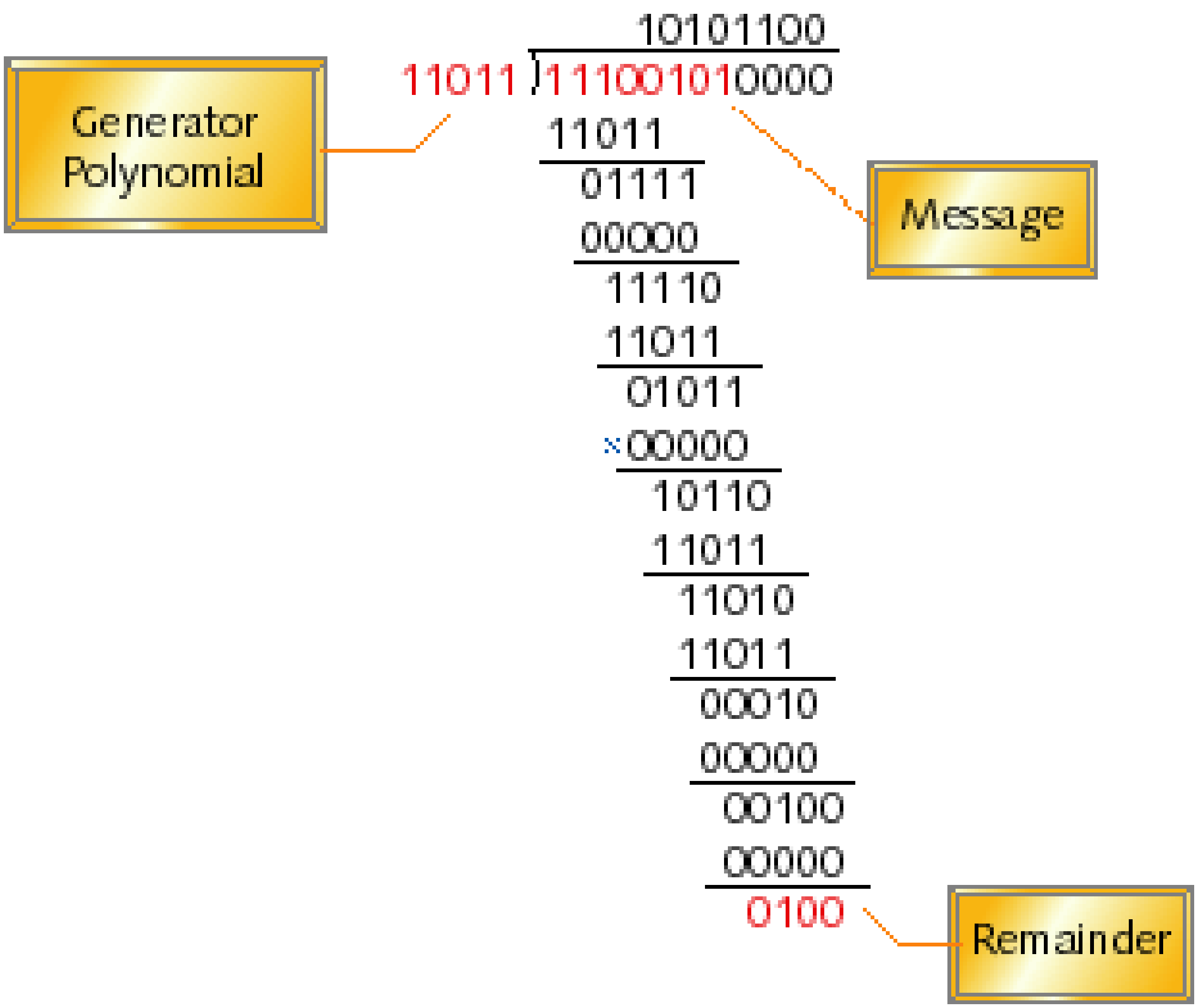
$$R = \text{remainder}\left[\frac{D \cdot 2^r}{G}\right]$$



Transmitted Frame:

101110011

CRC-Example 2

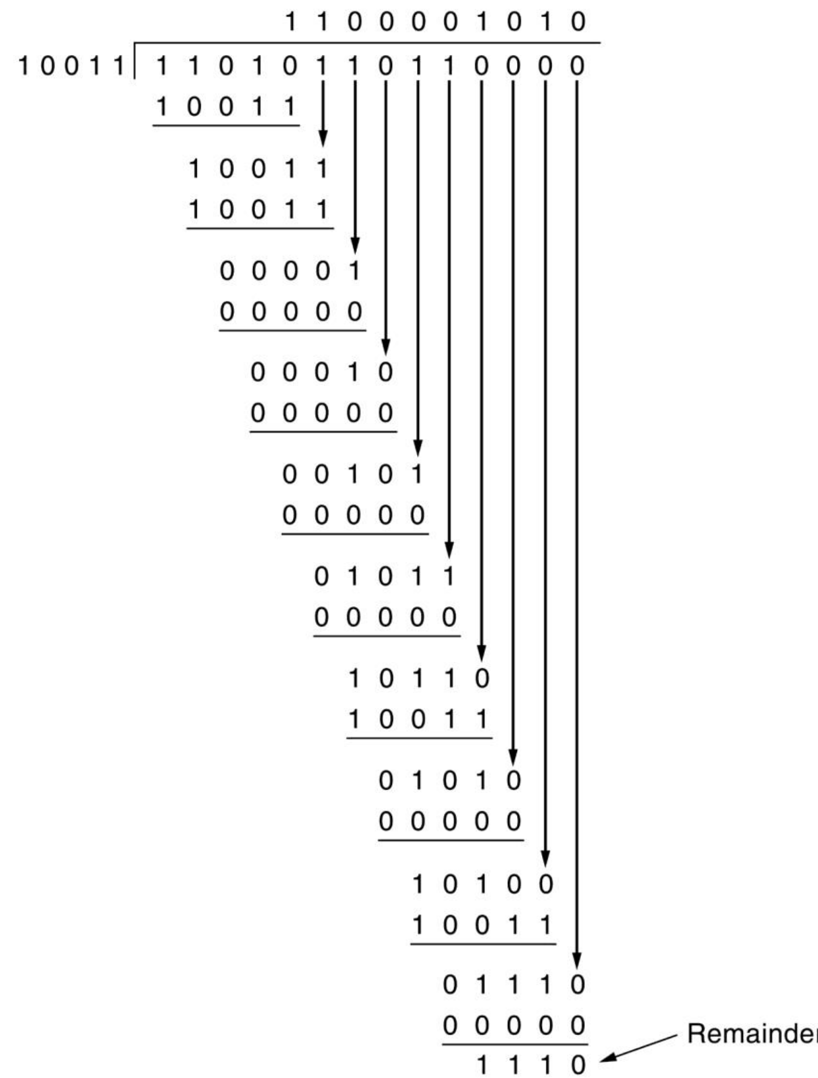


CRC-Example 3

Frame : 1 1 0 1 0 1 1 0 1 1

Generator: 1 0 0 1 1

Message after 4 zero bits are appended: 1 1 0 1 0 1 1 0 1 1 0 0 0 0



Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 1 0

CRC (Receiver Side)

- Receiver divide the received data stream by G(x) generator function.
- If the remainder is “0”, it is said that there is no error.
- Receiver crops the last ***r*** bits and send the ***d*** bits

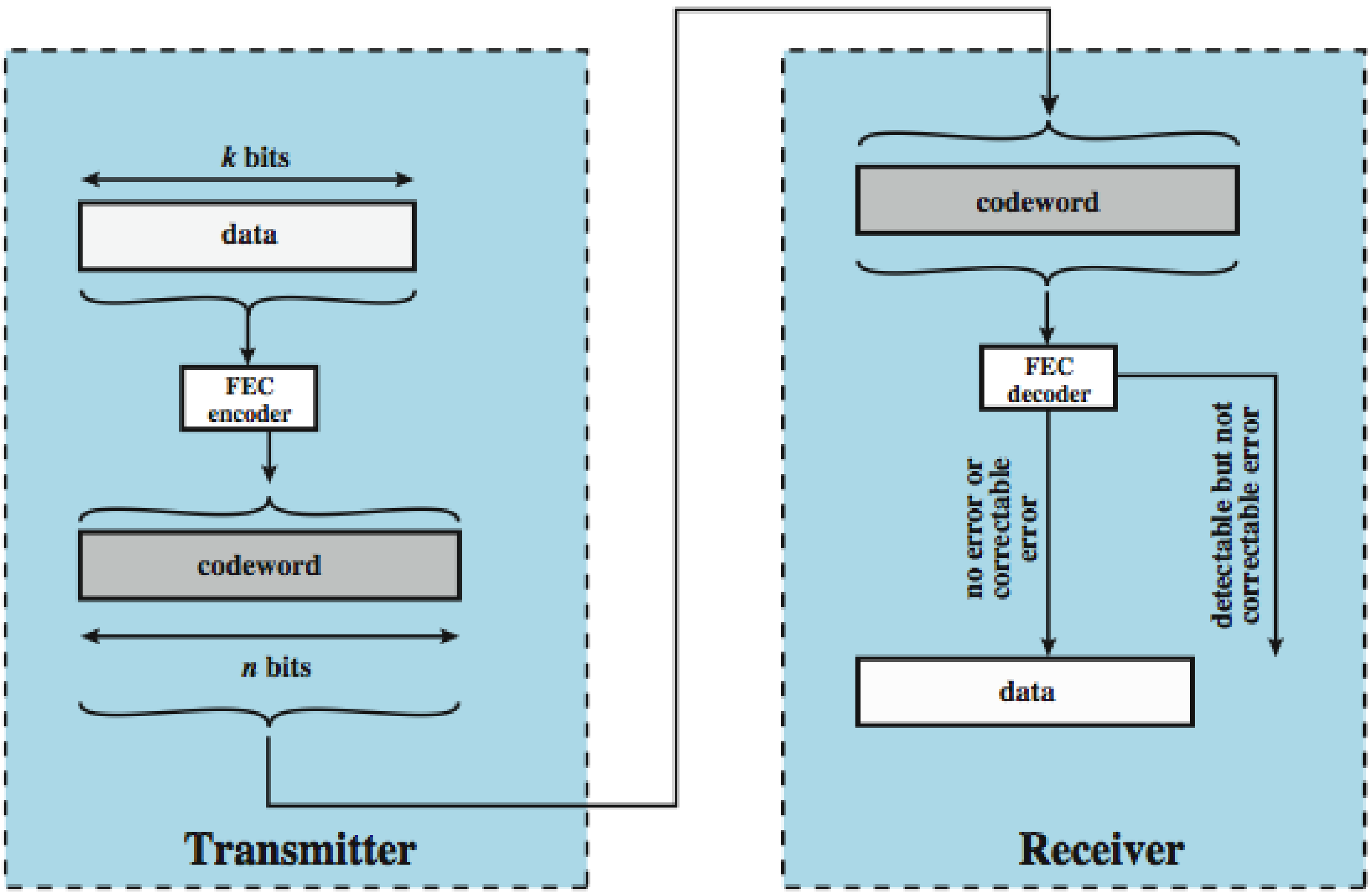
CRC Coding Standarts Example

CRC-32-IEEE 802.3	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$ (V.42, MPEG-2, PNG[16], POSIX cksum)
CRC-1	x + 1 (most hardware; also known as parity bit)
CRC-8-ATM	x8 + x2 + x + 1 (ATM HEC)
CRC-16-IBM	$x^{16} + x^{15} + x^{2} + 1$ (Bisync, Modbus, USB, ANSI X3.28, many others; also known as *CRC-16* and *CRC-16-ANSI*)

Digital Data Transmission Techniques

- Transmission Technique (Serial, Parallel)
- Error Detection Techniques
- Error Correction Techniques
- Connection Interface Standarts

Error Correction Process



How Error Correction Works

- adds redundancy to transmitted message
- can deduce original despite some errors
- eg. block error correction code
  - map *k* bit input onto an *n* bit codeword
  - each distinctly different
  - if get error assume codeword sent was closest to that received
- means have reduced effective data rate

Hamming Code Rate

As you can see, if you have m parity bits, it can cover bits from 1 up to $2^m - 1$.

If we subtract out the parity bits, we are left with $2^m - m - 1$ bits we can use for the data. As m varies, we get all the possible Hamming codes:

Parity bits	Total bits	Data bits	Name
2	3	1	Hamming(3,1) (Triple repetition code)
3	7	4	Hamming(7,4)
4	15	11	Hamming(15,11)
5	31	26	Hamming(31,26)
		...	
m	$2^m - 1$	$2^m - m - 1$	Hamming($2^m - 1, 2^m - m - 1$)

Parity bit Coverage

Bit position		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Encoded data bits		P1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8	d9	d10	d11	p16	d12	d13	d14	d15	
Parity bit coveragE	p1	X		X		X		X		X		X		X		X		X		X		
	p2		X	X			X	X			X	X			X	X			X	X		...
	p4				X	X	X	X					X	X	X	X					X	
	p8								X	X	X	X	X	X	X	X						
	P16																X	X	X	X	X	

Hamming Code

- M message bits
- C test bits
- $C1 = M1 \oplus M2 \oplus M4 \oplus M5 \oplus M7$
- $C2 = M1 \oplus M3 \oplus M4 \oplus M6 \oplus M7$
- $C3 = M2 \oplus M3 \oplus M4 \oplus M8$
- $C4 = M5 \oplus M6 \oplus M7 \oplus M8$

- http://cnx.org/content/m18663/latest/lbc_hamming-encoder-hammingcode.htm

Hamming Code Example

- Suppose that we have a data with seven bits.
  0110101
- Calculate the Hamming code.

Hamming Code Example

	p_1	p_2	d_1	p_3	d_2	d_3	d_4	p_4	d_5	d_6	d_7
Data (no parity bit):			**0**		**1**	**1**	**0**		**1**	**0**	**1**
P_1	**1**		0		1		0		1		1
P_2		**0**	0			1	0			0	1
P_3				**0**	1	1	0				
P_4								**0**	1	0	1
Data (with parity bits):	**1**	**0**	0	**0**	1	1	0	**0**	1	0	1

Hamming Code Example

- **10**0**0**110**0**101 (D1-D11) is the stream of data sent.
- **10**0**0**110**0**100 is the stream of data received. (suppose that the MSB is in error)

- Execute the Hamming process at the receiver side.

Hamming Code Example

	p_1	p_2	d_1	p_3	d_2	d_3	d_4	p_4	d_5	d_6	d_7	Parity check	Parity bit
Received Data	1	0	0	0	1	1	0	0	1	0	0		
p_1	**1**		0		1		0		1		0	**Fall**	**1**
p_2		**0**	0			1	0			0	0	**Fall**	**1**
p_3				**0**	1	1	0					Pass	0
p_4								**0**	1	0	0	**Fall**	**1**

- If parity check falls then new parity bit is "1". Else "0"
- All parity bits are in even parity.

- p4p3p2p1 = $(1011)_2 = 11$
- So error is in 11th bit.

Hamming Code Simulator

- http://candle.ctit.utwente.nl/wp5/tel-sys/exercises/datalinkp2p/hamming74demo.html