

# BM 403 Veri İletişimi (Data Communications)

---

Hazırlayan: M.Ali Akcayol  
Gazi Üniversitesi  
Bilgisayar Mühendisliği Bölümü

## Ders konuları

---

- Giriş
- Blok kodlama
- Lineer blok kodlar
- Cyclic kodlar
- Checksum

## Giriş

- Ağlar iki cihaz arasında kabul edilebilir doğrulukta veri transferi yapmak zorundadır.
- Veri iletişim sırasında bozulabilir. Bazı uygulamalar bozulmaları denetleme ve düzeltme gerektirir.
- Bazı uygulamalar belirli bir hata seviyesini tolere edebilir.
- **Single-bit error**, sadece bir bitin değeri değişir.
- **Burst error**, çok sayıda bitin değişir.
- 1200 bps hızında iletişim yapılırken eğer 0,01 sn burst error (impulse noise) oluşmuşsa, toplam 12 bit bozulur.
- **Single-bit error**, genellikle seri veri iletişiminde oluşma olasılığı azdır. 1Mbps hızında iletişim yapılırken her bit 1µs gerekir. Gürültünün 1µs süreye sahip olması gerekir. Gürültü normalde daha uzun süre devam eder.

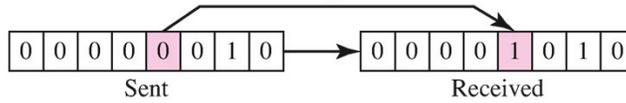
3/35

## Giriş

- Burst error, en az iki veya daha fazla bit bozulur. Bozulan bitler ardarda olmayabilir.
- Şekilde single-bit error ve burst error görülmektedir.

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

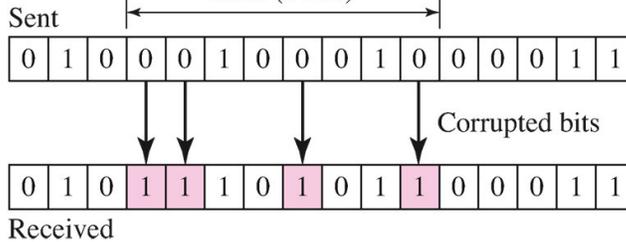
0 changed to 1



Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

Length of burst

error (8 bits)



4/35

## Giriş

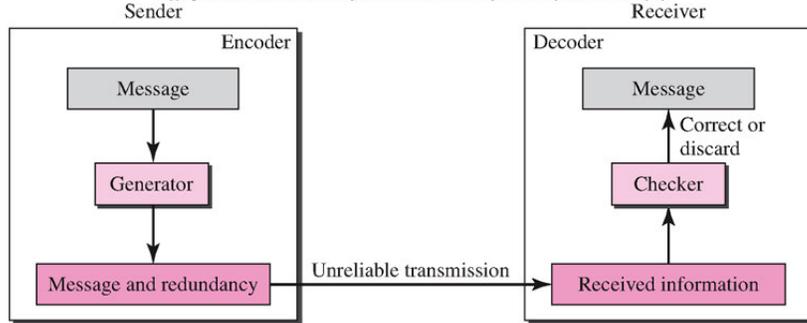
- 1kbps hızında iletişim yapılırken 1/100 sn noise oluşursa, 10 bit bozulur. 1Mbps hızında iletişim yapılırsa toplam 10.000 bit bozulur.
- Hata denetimi ve düzeltmede **redundancy** temel yaklaşımdır. Gönderilen veriyle birlikte hata denetim ve düzeltme bitleri eklenir.
- Redundant bitler gönderende eklenir, alıcıda çıkartılır.
- Hata düzeltme hata denetlemeden çok daha zordur.
- Error detection**(hata denetleme), hata olup olmadığına bakılır.
- Error correction**(hata düzeltme), hata varsa düzeltme işlemi yapılır.
- Hata denetlemede hatanın boyutuyla ilgilenilmez.
- Hata düzeltmede tam olarak kaç bitte bozulma olduğunu bilmek zorundayız.

5/36

## Giriş

- Forward error correction**, alıcı tarafından redundant bitler kullanılarak mesajın tamamı tahmin edilmeye çalışılır.
- Retransmit**, alıcı hatayı denetler ve göndericiden mesajı tekrar göndermesini ister.
- Redundancy**, farklı kodlama yöntemleri kullanılarak oluşturulur.
- Kodlama yöntemleri, iki grupta yapılır: **blok kodlama** ve **convolution kodlama**.

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.



6/36

## Giriş

- **Modüler aritmetikte**, sınırlı aralıkta tamsayı kullanılır.
- Mod-2 aritmetikte toplama ve çıkarma işlemleri aşağıdaki gibi yapılır:  
Toplama:  $0+0=0$   $0+1=1$   $1+0=1$   $1+1=0$   
Çıkarma:  $0-0=0$   $0-1=1$   $1-0=1$   $1-1=0$
- Toplama ve çıkarma aynı sonucu verir ve **XOR** işlemiyle gerçekleştirilirler.
- XOR işlemi aynı girişler için 0, farklı girişler için 1 sonucunu üretir.

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

$$0 \oplus 0 = 0 \qquad 1 \oplus 1 = 0$$

a. Two bits are the same, the result is 0.

$$0 \oplus 1 = 1 \qquad 1 \oplus 0 = 1$$

b. Two bits are different, the result is 1.

$$\begin{array}{rcccccc} & 1 & 0 & 1 & 1 & 0 \\ \oplus & 1 & 1 & 1 & 0 & 0 \\ \hline & 0 & 1 & 0 & 1 & 0 \end{array}$$

c. Result of XORing two patterns

7/35

## Ders konuları

- Giriş
- **Blok kodlama**
- Lineer blok kodlar
- Cyclic kodlar
- Checksum

8/35

## Blok kodlama

- Blok kodlamada mesaj  $k$ -bit bloklara (dataword) bölünür.
- Her bloğa  $r$ -bit redundant bit eklenir.
- Oluşan  $n=k+r$  bit **codeword** olarak adlandırılır.
- Toplam  $2^k$  dataword ve  $2^n$  codeword üretilebilir. Ancak  $2^k$  adet codeword kullanılır.

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.



$2^k$  Datawords, each of  $k$  bits



$2^n$  Codewords, each of  $n$  bits (only  $2^k$  of them are valid)

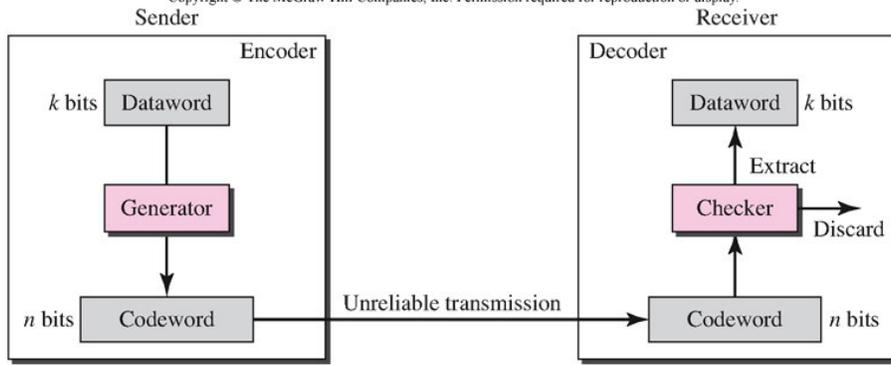
9/36

## Blok kodlama

### Hata denetimi (error detection)

- İki durumda alıcı orijinal codeword'deki hatayı algılar.
  - Alıcı geçerli codeword listesine sahiptir.
  - Orijinal codeword geçerli olmayan bir tanesi ile değişir.
- Gönderici dataword kullanarak codeword oluşturur.

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.



10/36

## Blok kodlama

### Hata denetimi (error detection) - örnek

- $k = 2$  ve  $n = 3$  için aşağıdaki tablo dataword ve codeword listesini göstermektedir.
- Alıcı 011 geldiğinde tabloda olduğu için geçerli olarak belirler.
- İletim sırasında codeword bozulur ve alıcı 111 alırsa tabloda olmadığı için geçersiz olarak belirler.
- Codeword bozulur ve alıcı 000 alırsa (sağdaki iki bit 0 olmuş) tabloda olduğundan geçerli olarak belirlenir. Aslında bozulan codeword'tür.

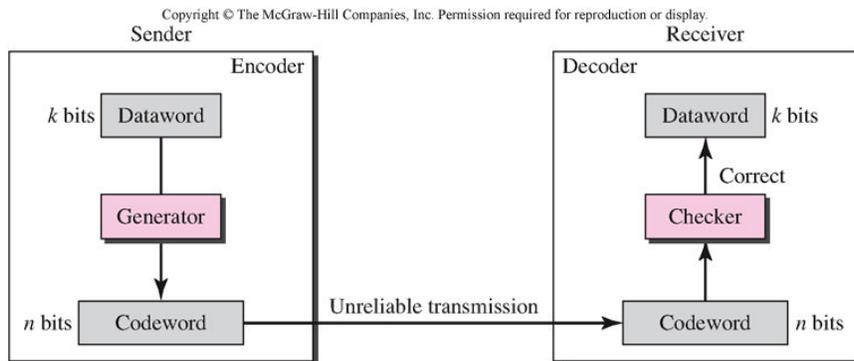
Dataword	Codeword
00	000
01	011
10	101
11	110

11/35

## Blok kodlama

### Hata düzeltme (error correction)

- Hata denetiminde alıcı sadece codeword'ün geçerli olup olmadığına bakar.
- Hata düzeltmede gereken redundant bit sayısı hata denetimine göre daha fazladır.



12/35

## Blok kodlama

### Hata düzeltme (error correction) - örnek

- Önceki tabloya iki bit redundant bit daha eklenirse aşağıdaki tablo oluşur.
- 2-bit dataword ve 3 bit redundant vardır.
- Dataword 01 ise gönderen tablodan codeword olarak 01011 oluşturur.
- Code bozulur ve alıcı 01001 alırsa önce tabloya bakılır ve olmadığı görülür.
- Alıcı 1-bit bozulma olduğunu tahmin eder ve ilk sıradan kontrol etmeye başlar. İlk sıradakiyle 2-bit fark vardır.
- 2.sıradaki hariç diğerleride 2-bit farklıdır. 2.sıradaki 1-bit farklı olduğu için gelen codeword ikinci sıradakiyle değiştirilir.

Dataword	Codeword
00	00000
01	01011
10	10101
11	11110

13/36

## Blok kodlama

### Hamming distance (hamming uzaklığı)

- Hamming distance iki word arasında farklı bit sayısını gösterir.
- x ve y word'leri arasındaki hamming distance  $d(x,y)$  şeklide gösterilir.
- Hamming distance XOR kapısıyla gerçekleştirilir. Elde edilen 1 sayısı hamming distance değerini gösterir.
- $d(000, 011) = 2$  olur ( $000 \oplus 011 = 011$ )
- $d(10101, 11110) = 3$  olur ( $10101 \oplus 11110 = 01011$ )

14/36

## Blok kodlama

### Minimum Hamming distance

- Tüm olası çiftler içinde en küçük Hamming distance değeridir.
- Aşağıdaki tablo için Hamming distance değerleri eşittir.  
 $d(000, 011) = 2$        $d(000, 101) = 2$        $d(000, 110) = 2$   
 $d(011, 101) = 2$        $d(011, 110) = 2$        $d(101, 110) = 2$
- Bir kodlama yöntemi (C) 3 parametreyle ifade edilir: **codeword boyutu** (n), **dataword boyutu** (k) ve **minimum Hamming distance** ( $d_{\min}$ ).
- İlk tablodaki kodlama C(3,2) ve  $d_{\min} = 2$ , ikinci tablodaki kodlamada C(5,2) ve  $d_{\min} = 3$  şeklinde gösterilir.
- İletişim sırasında s tane hata olursa, gönderilen ve alınan codeword'ler arasındaki Hamming distance s olur.
- Tüm durumlar için s tane hatayı algılamak için, tüm çiftler arasındaki **minimum Hamming distance** değerinin **s+1** olması gerekir.

15/36

## Blok kodlama

### Minimum Hamming distance - örnek

- Aşağıdaki tabloda **minimum Hamming distance = 2'** dir. **1-bit hata denetimi** garanti edilir.
- Eğer 3.satırdaki codeword gönderilirse ve 1-bit hata olursa, alınan codeword tablodakilerin hiçbirisiyle aynı değildir.
- Eğer 2-bit hata olursa, alınan hatalı codeword tablodaki codeword'lerden birisiyle aynı olabilir ve hata algılanamaz.

Dataword	Codeword
00	000
01	011
10	101
11	110

16/36

## Blok kodlama

### Minimum Hamming distance - örnek

- Aşağıdaki tabloda **minimum Hamming distance = 3** tür. **2-bit hata denetimi** garanti edilir.
- Eğer geçerli bir codeword gönderilirse ve 2-bit hata olursa, alınan codeword tablodakilerin hiçbirisiyle aynı olmayabilir.
- Aynı tabloda 3-bit hataların bazıları da tabloda geçerli olan bir codeword oluşturabilir.

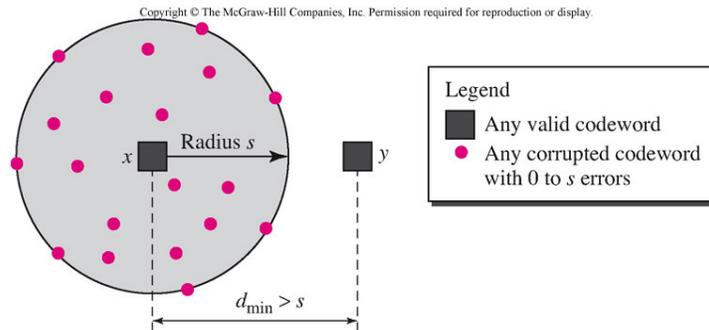
Dataword	Codeword
00	00000
01	01011
10	10101
11	11110

17/36

## Blok kodlama

### Minimum Hamming distance – hata denetleme

- Aşağıdaki şekilde bir x codeword ve etrafında s yarıçapında codeword kümesi vardır.
- Tabloda minimum Hamming distance değeri s'den büyüktür.  $d_{\min} = s + 1$
- Diğer geçerli olan tüm codeword'ler çemberin dışındadır.



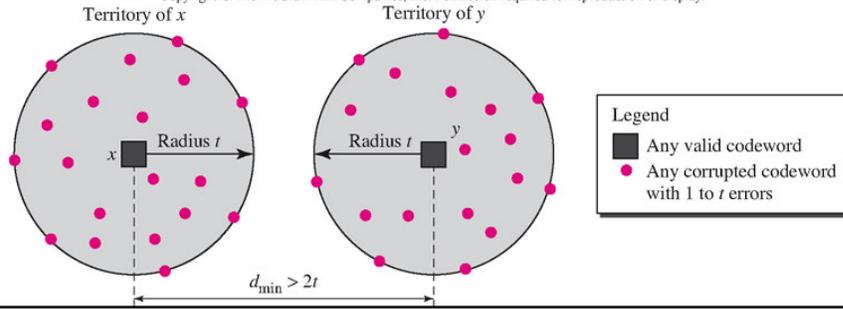
18/36

## Blok kodlama

### Minimum Hamming distance – hata düzeltme

- Alıcı aldığı codeword'ün geçersiz olduğunu algılasa, tablodan hangi codeword'ün gönderildiğini bulmaya çalışır.
- Her geçerli codeword, kendi etrafında  $t$  yarıçapında bir çember oluşturur.
- Alıcı aldığı geçersiz codeword'ün hangi geçerli codeword'e ait çemberde olduğunu belirler. Belirlenen çemberin ortasındaki codeword gönderilen gerçek codeword olarak alınır. Hata düzeltmede  $d_{\min} > 2t$  olmalıdır.

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.



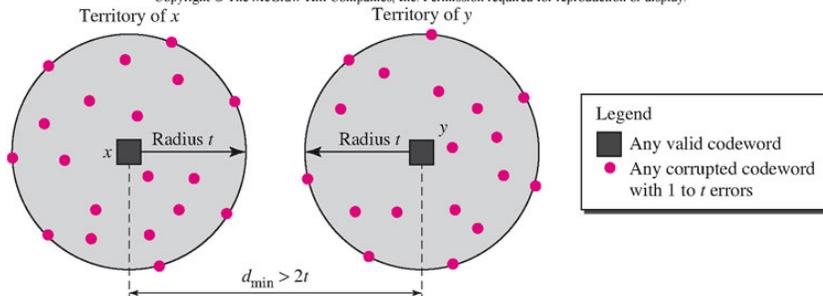
19/35

## Blok kodlama

### Minimum Hamming distance – hata düzeltme

- Alıcı aldığı codeword'ün geçersiz olduğunu algılasa, tablodan hangi codeword'ün gönderildiğini bulmaya çalışır.
- Her geçerli codeword, kendi etrafında  $t$  yarıçapında bir çember oluşturur.
- Alıcı aldığı geçersiz codeword'ün hangi geçerli codeword'e ait çemberde olduğunu belirler. Belirlenen çemberin ortasındaki codeword gönderilen gerçek codeword olarak alınır. Hata düzeltmede  $d_{\min} = 2t+1$  olmalıdır.

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.



20/35

## Blok kodlama

### Minimum Hamming distance – hata düzeltme – örnek

- Bir kodlama yönteminde  $d_{\min} = 4$  ise hata denetleme ve hata düzeltme kapasitesi nedir ?

Bu kodlamada  $s = 3$  için **3 bit hata denetleme** garanti edilir.

Ancak en fazla **1 bit hata düzeltme** yapılabilir. ( $2t+1 = 4$ ,  $t = 1,5$ )

**Hata düzeltme için kodlama yönteminin  $d_{\min}$  değeri tek sayı (t tamsayı olur) olmalıdır.**

21/35

## Ders konuları

- Giriş
- Blok kodlama
- **Lineer blok kodlar**
- Cyclic kodlar
- Checksum

22/35

## Lineer blok kodlar

- İki codeword arasındaki XOR sonucu yine geçerli bir codeword oluşturursa bu kodlar lineer blok kodlardır.
- Aşağıdaki tabloda herhangi iki codeword arasında XOR işlemi yapılırsa, yine geçerli bir codeword oluşmaktadır.
- Lineer blok kod için minimum Hamming distance değeri 0'dan farklı geçerli codeword'deki 1 sayısına eşittir.
- Soldaki tabloda 0'dan farklı codeword'lerdeki 1 sayısı 2,2 ve 2'dir. Minimum Hamming distance değeri 2 olur.
- Sağdaki tabloda 3,3 ve 4'tür.  $d_{\min} = 3$  olur.

Dataword	Codeword
00	000
01	011
10	101
11	110

Dataword	Codeword
00	00000
01	01011
10	10101
11	11110

23/35

## Lineer blok kodlar

- Simple parity-check code**, en yaygın kullanılan hata denetim kodudur.
- Bu kodlamada k-bit dataword n codeword'e dönüştürülür. ( $n = k+1$ )
- Extra bit toplam 1 sayısını çift veya tek yapmak için kullanılır.
- Bu kodlamada  $d_{\min} = 2$ 'dir ve hata düzeltme yapılamaz sadece 1-bit hata denetimi yapılır.
- Aşağıdaki tablo C(4,5) parity check kodudur.

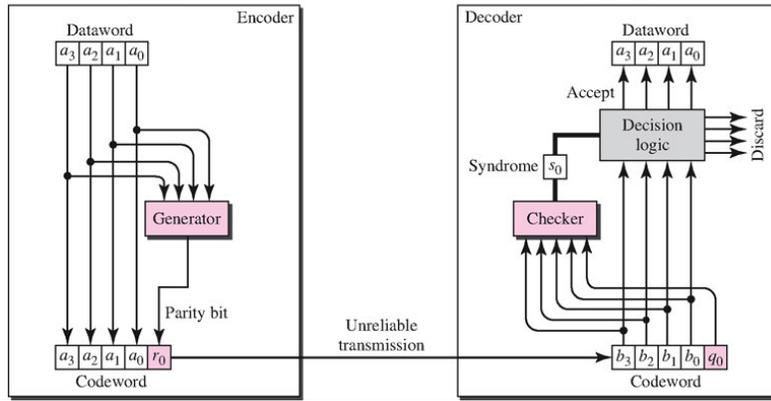
Dataword	Codeword	Dataword	Codeword
0000	00000	1000	10001
0001	00011	1001	10010
0010	00101	1010	10100
0011	00110	1011	10111
0100	01001	1100	11000
0101	01010	1101	11011
0110	01100	1110	11101
0111	01111	1111	11110

24/35

## Lineer blok kodlar

- Şekilde 4-bit dataword'e 1-bit parity biti ekleniyor. Parity bit 5-bit codeword'deki 1 sayısını çift yapmaktadır.
- Dataword'deki 4-bit mod 2 ye göre toplanır. ( $r_0 = a_3 + a_2 + a_1 + a_0$ )
- Alicıda sonuç (syndrome) 0 ise doğrudur, 1 ise hatalıdır ve atılır (discard).

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.



25/36

## Lineer blok kodlar

- Two-dimensional parity check** yaklaşımında her satır ve sütun için 1-bit parity bit eklenir.
- 3-bit hata denetimi yapılabilir.

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

1	1	0	0	1	1	1	1	1
1	0	1	1	1	0	1	1	1
0	1	1	1	0	0	1	0	1
0	1	0	1	0	0	1	1	1
0	1	0	1	0	1	0	1	1
								Column parities
0	1	0	1	0	1	0	1	1

a. Design of row and column parities

1	1	0	0	1	1	1	1	1
1	0	1	1	1	0	1	1	1
0	1	1	1	0	0	1	0	1
0	1	0	1	0	0	1	1	1
0	1	0	1	0	1	0	1	1
								Column parities
0	1	0	1	0	1	0	1	1

b. One error affects two parities

1	1	0	0	1	1	1	1	1
1	0	1	1	1	0	1	1	1
0	1	1	1	0	0	1	0	1
0	1	0	1	0	0	1	1	1
0	1	0	1	0	1	0	1	1
								Column parities
0	1	0	1	0	1	0	1	1

c. Two errors affect two parities

1	1	0	0	1	1	1	1	1
1	0	1	1	1	0	1	1	1
0	1	1	1	0	0	1	0	1
0	1	0	1	0	0	1	1	1
0	1	0	1	0	1	0	1	1
								Column parities
0	1	0	1	0	1	0	1	1

d. Three errors affect four parities

1	1	0	0	1	1	1	1	1
1	0	1	1	1	0	1	1	1
0	1	1	1	0	0	1	0	1
0	1	0	1	0	0	1	1	1
0	1	0	1	0	1	0	1	1
								Column parities
0	1	0	1	0	1	0	1	1

e. Four errors cannot be detected

## Lineer blok kodlar

### Hamming kodları

- Hamming kodları  $d_{\min}=3$  ile tasarlanır. 2-bit hatayı denetler ve 1-bit düzeltir.
- $m \geq 3$  olmak üzere  $n=2^m - 1$  ve  $k = n - m'$  dir.  $r = m$  denetlenen bit sayısıdır.
- Aşağıdaki tabloda  $m=3$  ise  $n=7$  ve  $k=4$  olur. Hamming kod  $C(7,4)$ ,  $d_{\min}=3$  olur.

Dataword	Codeword	Dataword	Codeword
0000	0000000	1000	1000110
0001	0001101	1001	1001011
0010	0010111	1010	1010001
0011	0011010	1011	1011100
0100	0100011	1100	1100101
0101	0101110	1101	1101000
0110	0110100	1110	1110011
0111	0111001	1111	1111111

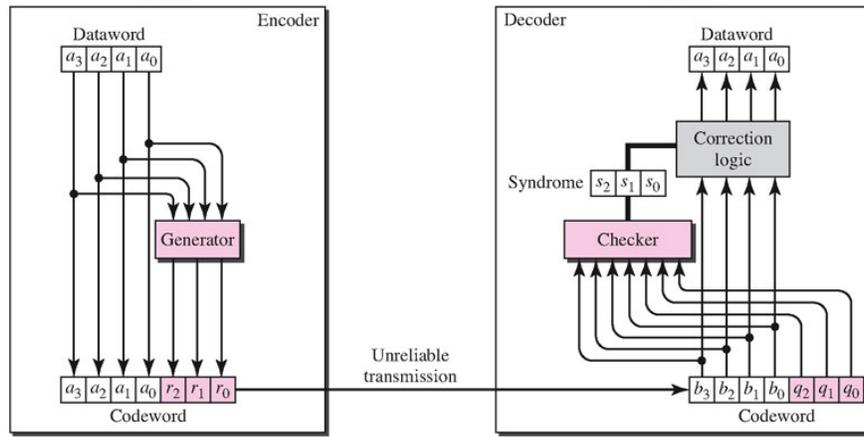
27/35

## Lineer blok kodlar

### Hamming kodları - örnek

- $C(7,4)$ ,  $d_{\min}=3$  için  $r_0=a_2+a_1+a_0$ ,  $r_1=a_3+a_2+a_1$ ,  $r_2=a_1+a_0+a_3$  ve alıcıda  $s_0=b_2+b_1+b_0+q_0$ ,  $s_1=b_3+b_2+b_1+q_1$ ,  $s_2=b_1+b_0+b_3+q_2$  ile 3 syndrome hesaplanır.

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.  
Sender Receiver



28/35

## Lineer blok kodlar

### Hamming kodları – örnek - devam

- 3 bit syndrome ile 8 farklı durum oluşturulur.
- Bu durumlar hata durumunu ve hata varsa hatalı biti gösterir.

Syndrome	000	001	010	011	100	101	110	111
Error	Yok	$q_0$	$q_1$	$b_2$	$q_2$	$b_0$	$b_3$	$b_1$

### Örnek

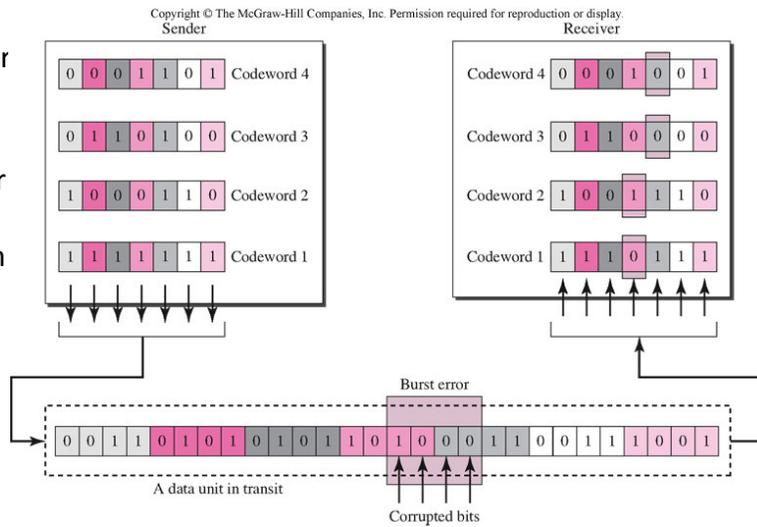
- 0100 dataword, codeword 0100011 olarak gönderiliyor. Codeword 0100011 olarak alındığında syndrome 000 olur ve hata yoktur.
- 0111 dataword, codeword 0111001 olarak gönderiliyor. Codeword 0011001 alınıyor. Syndrome 011 hesaplanır. Tabloya göre  $b_2$  hatalıdır.  $b_2$  0'dan 1'e değiştirilir ve dataword 0111 olur.
- 1101 dataword, codeword 1101000 olarak gönderiliyor. Codeword 0001000 alınıyor (2-bit hatalı). Syndrome 101 hesaplanır. Tabloya göre  $b_0$  hatalıdır.  $b_2$  1'den 0'a değiştirilir ve dataword 0000 olur. (yanlış)

29/36

## Lineer blok kodlar

### Hamming kodları - performans

- Hamming kodları 1-bit hata düzeltir ve 2-bit hata denetler.
- Ancak burst error denetleyebilir. Burst hata oluşan kısım farklı codeword'lere dağıtılır.



30/36

## Ders konuları

- Giriş
- Blok kodlama
- Lineer blok kodlar
- **Cyclic kodlar**
- Checksum

31/35

## Cyclic kodlar

- Cyclic kodlar özel lineer blok kodlardır. Her cyclic kod rotate edilebilir.
- Rotate edildikten sonra oluşan kod geçerli bir codeword'tür.
- 1011000 codeword left-shift yapılırsa codeword 0110001 olur ve geçerlidir.
- **Cyclic redundancy check (CRC)**, LAN ve WAN ağlarda kullanılır.

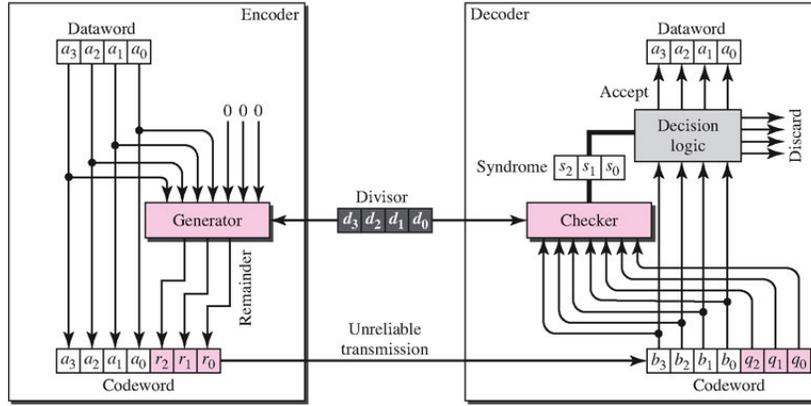
Dataword	Codeword	Dataword	Codeword
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111

32/35

## Cyclic kodlar

- Kodlayıcıda 4-bit dataword ( $k=4$ ) ve 7-bit codeword ( $n=7$ ). Dataword'ün sağ kısmına 000 eklenerek 7-bit codeword elde edilmiştir.
- Generator dataword'ü belirlenmiş olan divisor ile mod 2'ye göre böler. Bölüm atılır ve kalan dataword'e eklenerek codeword oluşturulur.

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

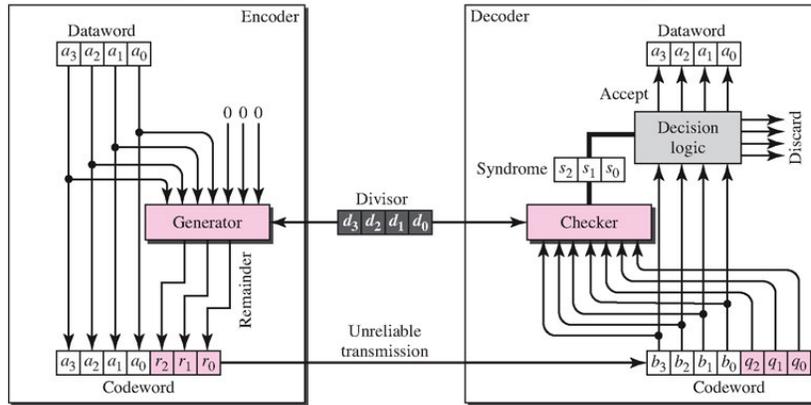


33/35

## Cyclic kodlar

- Alıcı gelen codeword'ü alır. Checker gelen bitlerin hepsini alır.
- Kalan tekrar oluşturulur ve kontrol edilerek  $s_2s_1s_0$  syndrome oluşturulur.
- Decision logic, gelen 4-bit dataword'ü alır veya iptal eder.

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

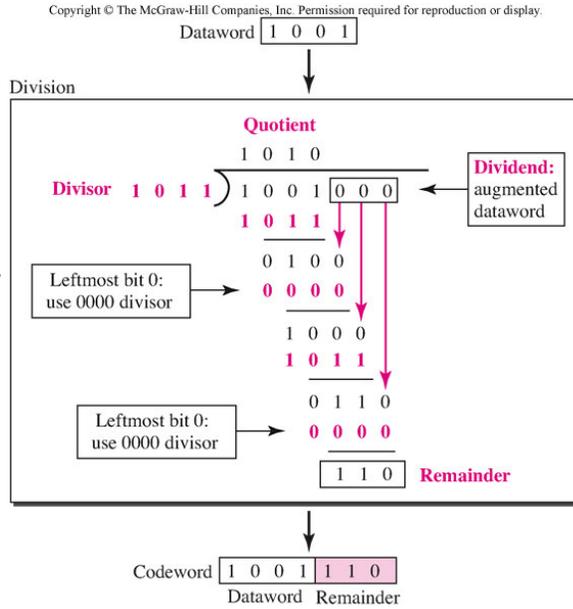


34/35

## Cyclic kodlar

### Encoder

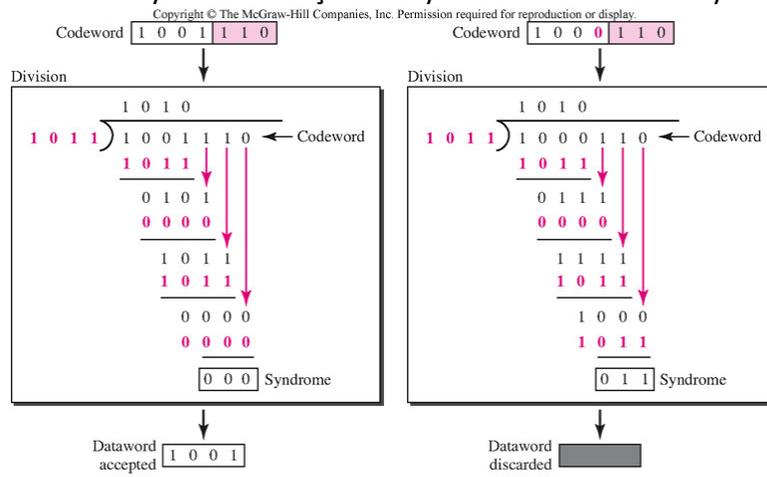
- Encoder dataword'ü alır ve n-k adet 0 ekler. Elde edilen 7-bit divisor ile bölünür.
- Toplama ve çıkarma işlemleri XOR ile yapılır.
- Her aşamada dividenden ile divisor XORlanır.
- Sonuçta kalan 3-bit dataword'e eklenir ve codeword elde edilir.



## Cyclic kodlar

### Decoder

- Decoder ile encoder'da yapılan bölme işlemi aynen yapılır.
- Bölmeden kalan syndrome oluşturur. Syndrome 000 ise hata yoktur.

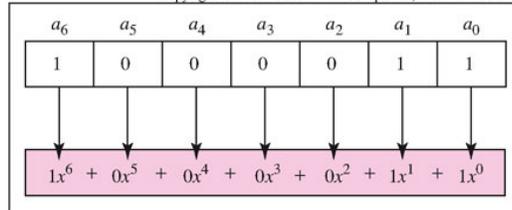


## Cyclic kodlar

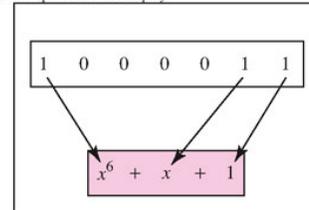
### Polinomlar

- Bir patern 0 ve 1 lerle birlikte polinom halinde gösterilebilir.
- Şekilde 7-bit patern 3 terimle gösterilebilmiştir.
- Polinomun derecesi en yüksek dereceli terimin derecesine eşittir.
- Polinomlarda toplama ve çıkarma aynı şekilde yapılır. Aynı dereceli olan çiftler silinir. Aynı dereceli 3 terim varsa ikisi silinir üçüncüsü alınır.
- Çarpma ve bölmede üstler toplanır veya çıkarılır.
- $(x^5+x^3+x^2+x)(x^2+x+1) = x^7+x^6+x^3+x$

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.



a. Binary pattern and polynomial



b. Short form

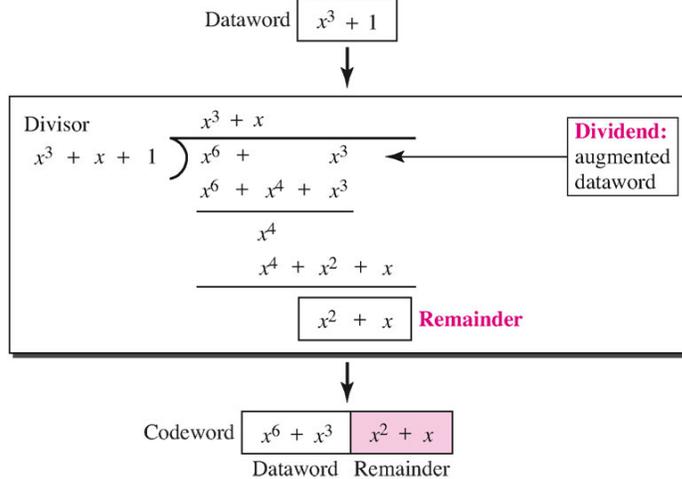
37/38

## Cyclic kodlar

### Polinomlar - devam

- Sola ve sağa shift binary shift'le aynıdır. Sola shift yapılırken sağa 0 eklenir.
- 10011 ( $x^4+x+1$ ) sola 3 shift yapılırsa 10011000 ( $x^7+x^4+x^3$ ) olur.
- 10011 ( $x^4+x+1$ ) sağa 3 shift yapılırsa 10 ( $x$ ) olur.
- Polinom işlemleri daha kısadır.
- Bölünen ilk terimi bölünen ilk terimine bölünür ( $x^6/x^3=x^3$ ) ve sonuç divisor ile çarpılır.

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.



38/38

## Cyclic kodlar

### Polinomlar - devam

- Tabloda CRC kodları ve kullanıldığı yerler verilmiştir.
- **Reed-Solomon** kodu günümüzde hata denetimi ve düzeltmesi için yaygın kullanılmaktadır.

Ad	Polinom	Uygulama
CRC-8	$X^8+X^2+X+1$	ATM başlık
CRC-10	$X^{10}+X^9+X^5+X^4+X^2+1$	ATM AAL
CRC-16	$X^{16}+X^{12}+X^5+1$	HDLC
CRC-32	$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$	LAN lar

39/36

## Ders konuları

- Giriş
- Blok kodlama
- Lineer blok kodlar
- Cyclic kodlar
- **Checksum**

40/36

## Checksum

- Checksum Internette yaygın kullanılmaktadır.
- Gönderilen sayıların toplamı alınır ve birlikte gönderilir. (7,11,12,0,6) için (7,11,12,0,6,36).
- Alıcı gelen sayıları toplar ve gelen toplamla karşılaştırır. Aynı ise data alınır değilse atılır.
- Checksum'da toplama işlemi **one's complement (birin tümleyeni)** aritmetiğiyle yapılır.

### Örnek

- 21 sayısını sadece 4 bitle gösteriniz.

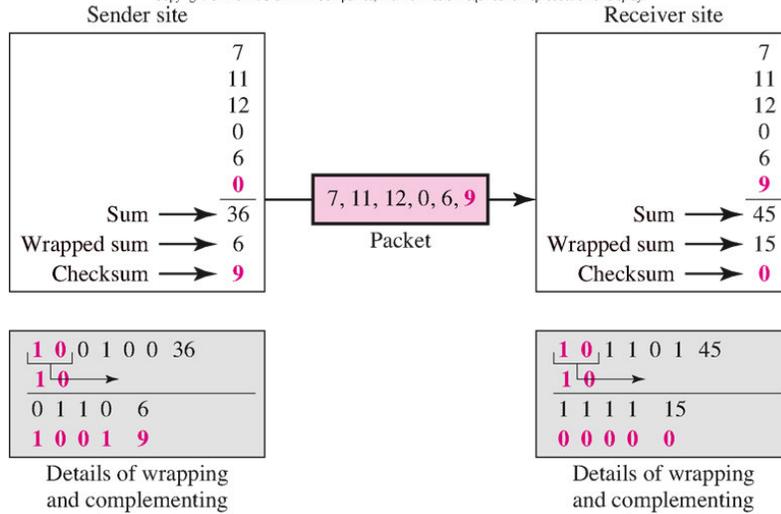
21 = 10101, en soldaki 1 sağa alınır ve toplanır. 0101 + 1 = 0110 (6) olur.

41/36

## Checksum

- Şekilde 4-bit kullanılarak gönderici ve alıcı tarafta yapılan işlemler görülmektedir.

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.



42/36

## Checksum

- İnternette 16-bit checksum kullanılır.
- Gönderen mesajı 16-bit parçalara böler ve hepsini 1 tümleyene göre toplar.
- Toplamın tümleyeni alınır ve checksum elde edilir.
- Alıcı aynı işlemleri tekrarlar. Checksum değeri 0 olursa hata yoktur.

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

<b>1</b>	<b>0</b>	<b>1</b>	<b>3</b>	Carries
4	6	6	F	<b>(Fo)</b>
7	2	6	F	<b>(ro)</b>
7	5	7	A	<b>(uz)</b>
6	1	6	E	<b>(an)</b>
0	0	0	0	Checksum (initial)
8	F	C	6	Sum (partial)
				→ 1
8	F	C	7	Sum
<b>7</b>	<b>0</b>	<b>3</b>	<b>8</b>	<b>Checksum (to send)</b>

a. Checksum at the sender site

<b>1</b>	<b>0</b>	<b>1</b>	<b>3</b>	Carries
4	6	6	F	<b>(Fo)</b>
7	2	6	F	<b>(ro)</b>
7	5	7	A	<b>(uz)</b>
6	1	6	E	<b>(an)</b>
<b>7</b>	<b>0</b>	<b>3</b>	<b>8</b>	Checksum (received)
F	F	F	E	Sum (partial)
				→ 1
F	F	F	F	Sum
<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>Checksum (new)</b>

b. Checksum at the receiver site