



# Lojik Tasarım

Ders 7

Kaynak:

M.M. Mano, M.D. Ciletti, "Digital Design with An Introduction to Verilog HDL"



# KOMBİNEZONAL LOJİK DEVRELERİ (Bileşik Mantık Devreleri)

- Sayısal devreler kombinezonal (bileşik) yada ardışıl olabilir.
- Bir bileşik devrede çıkışlar, o anki giriş değerlerine bağlı olan lojik kapılardan oluşmuştur.
- Ardışıl derelerde ise bellek elemanları kullanılır ve çıkış, giriş değerleri ve bellek elemanlarında tutulan bilgilerin bir fonksiyonu olarak tanımlanır.

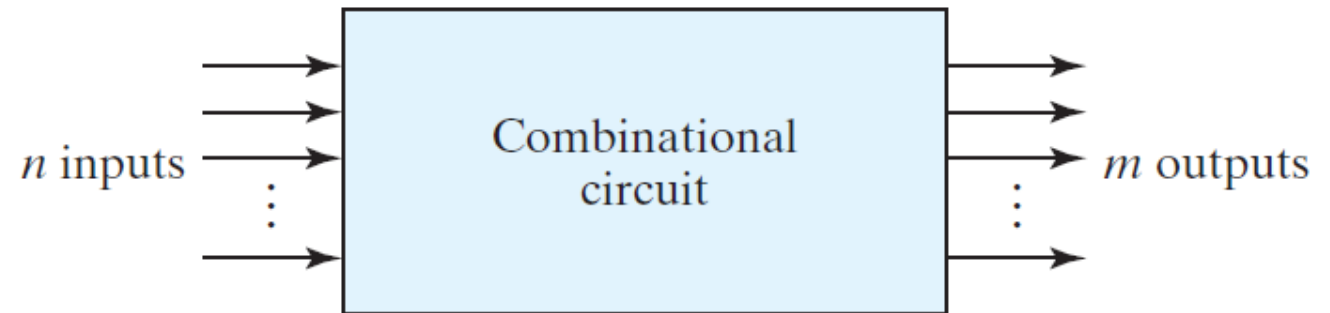
# Bileşik Mantık Devreleri

► Bileşik mantık devreleri üç kısımdan oluşur

1- Giriş değişkenleri

2- İşlem yapan lojik kapılar

3- Çıkış değişkenleri



**FIGURE 4.1**

Block diagram of combinational circuit



# Tasarım Yöntemi

1. Problem sözel olarak ifade edilir
2. Giriş ve çıkış değişkenlerinin sayısı belirlenir
3. Giriş ve çıkış değişkenlerine harf sembolleri atanır
4. Giriş ile çıkış arasındaki ilişkileri tanımlayan doğruluk tablosu oluşturulur
5. Her çıkış için basitleştirilmiş Boole fonksiyonu elde edilir
6. Lojik devre çizilir

BCD kodundan Üç Fazlalık koduna  
dönüştüren lojik devreyi tasarlayınız.

GİRİŞ					ÇIKIŞ			
A	B	C	D		W	X	Y	Z
0	0	0	0		0	0	1	1
0	0	0	1		0	1	0	0
0	0	1	0		0	1	0	1
0	0	1	1		0	1	1	0
0	1	0	0		0	1	1	1
0	1	0	1		1	0	0	0
0	1	1	0		1	0	0	1
0	1	1	1		1	0	1	0
1	0	0	0		1	0	1	1
1	0	0	1		1	1	0	0

W

•  $CD$

$AB$

	$00$	$01$	$11$	$10$
$00$				
$01$		1	1	1
$11$	$\phi$	$\phi$	$\phi$	$\phi$
$10$	1	1	$\phi$	$\phi$

$$W = A + BC + BD$$

# X

•  $CD$

$AB$

	$00$	$01$	$11$	$10$
$00$		$1$	$1$	$1$
$01$	$1$			
$11$	$\phi$	$\phi$	$\phi$	$\phi$
$10$		$1$	$\phi$	$\phi$

$$X = B'C + B'D + BC'D'$$

BCD kodundan Üç Fazlalık koduna dönüştüren lojik devreyi tasarlayınız.

GİRİŞ					ÇIKIŞ			
A	B	C	D		W	X	Y	Z
0	0	0	0		0	0	1	1
0	0	0	1		0	1	0	0
0	0	1	0		0	1	0	1
0	0	1	1		0	1	1	0
0	1	0	0		0	1	1	1
0	1	0	1		1	0	0	0
0	1	1	0		1	0	0	1
0	1	1	1		1	0	1	0
1	0	0	0		1	0	1	1
1	0	0	1		1	1	0	0

Y

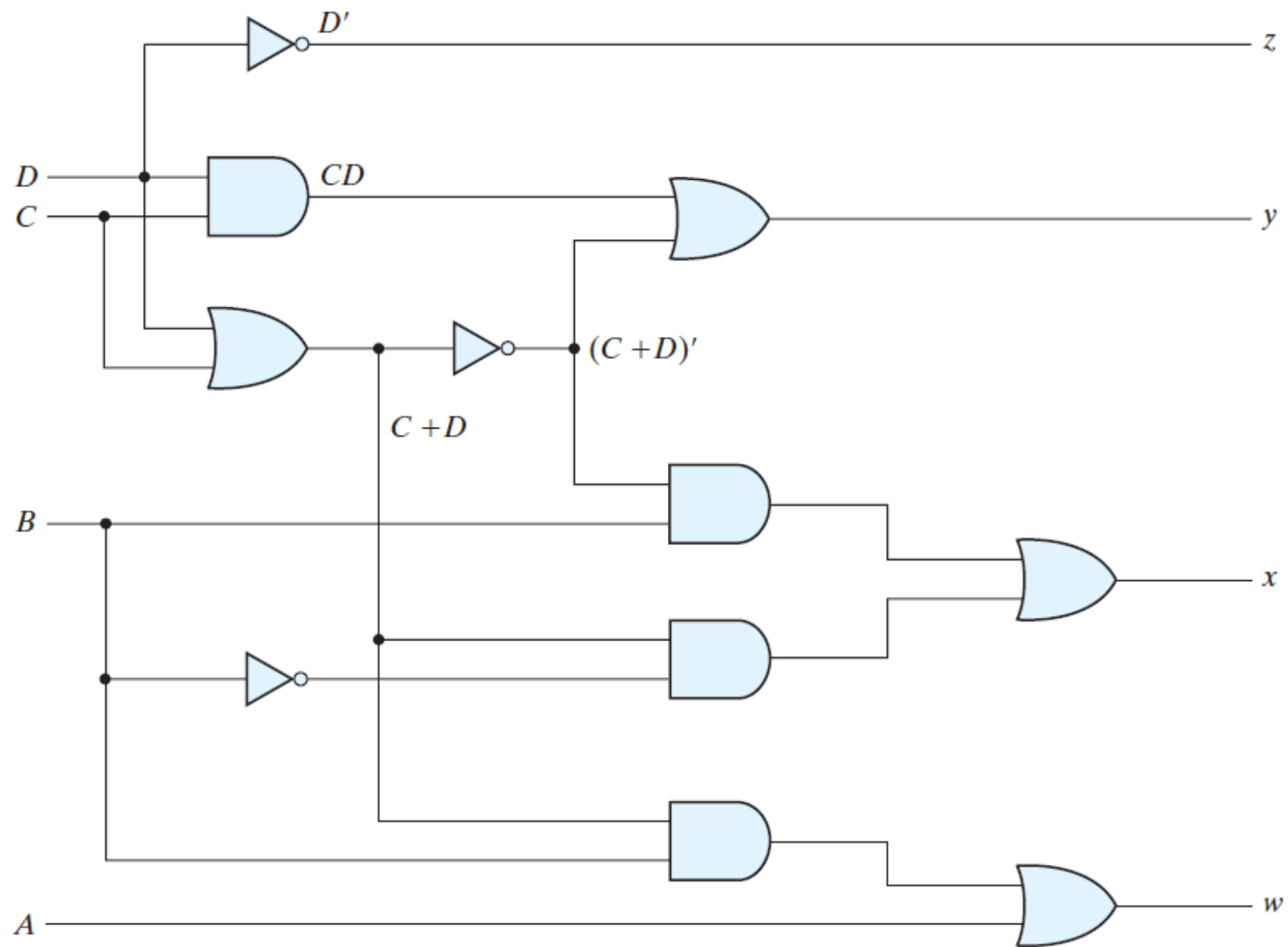
		CD			
	AB				
		00	01	11	10
00		1		1	
01		1		1	
11		$\phi$	$\phi$	$\phi$	$\phi$
10		1		$\phi$	$\phi$

$$Y = CD + C'D'$$

Z

		CD			
	AB				
		00	01	11	10
00		1			1
01		1			1
11		$\phi$	$\phi$	$\phi$	$\phi$
10		1		$\phi$	$\phi$

$$Z = D'$$





# Toplayıcı ve Çıkarıcılar





# 1-Yarı Toplayıcı

■ Soru:

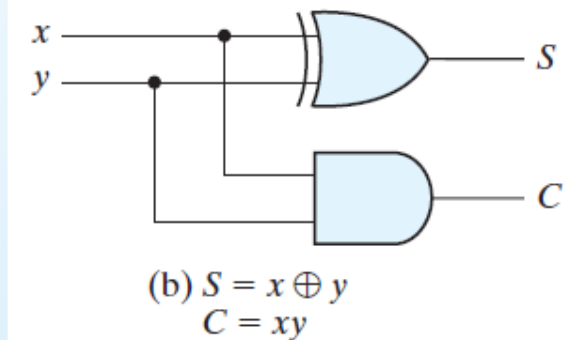
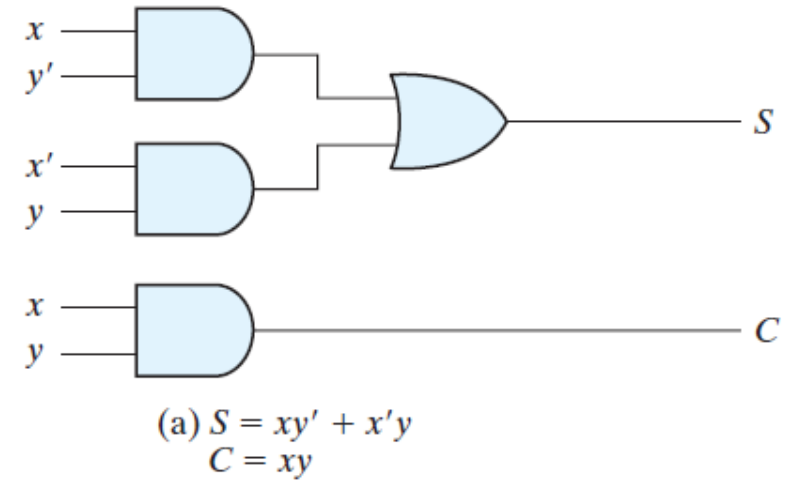
■ Bir bitlik iki sayıyı toplayacak devreyi tasarlayınız

*Half Adder*

$x$	$y$	$C$	$S$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = x'y + xy'$$

$$C = xy$$



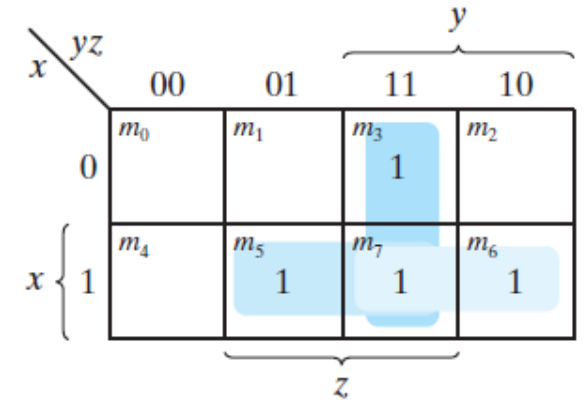
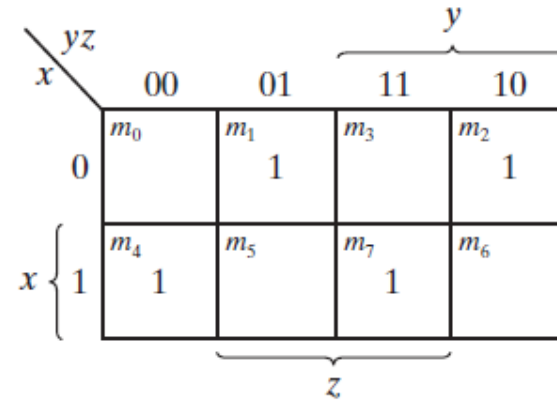
# 1- Tam Toplayıcı

► Soru:

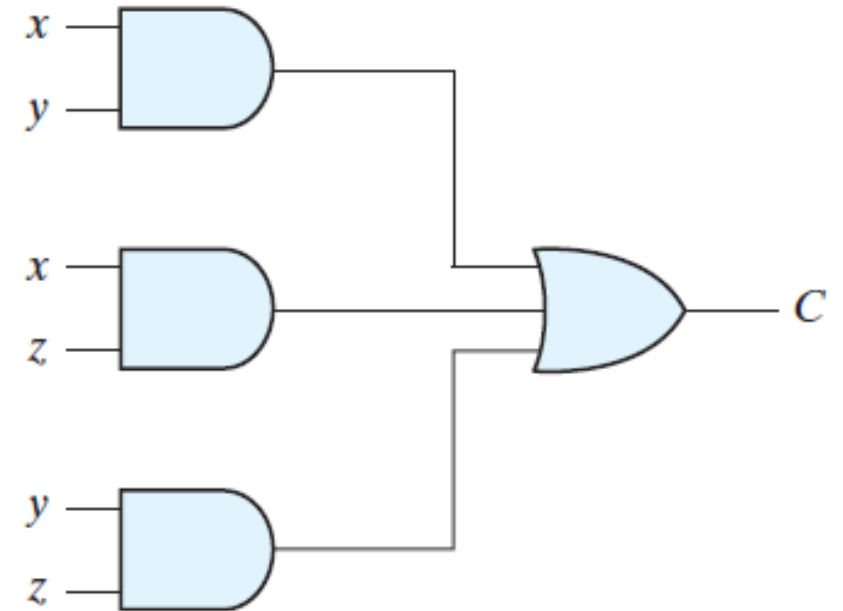
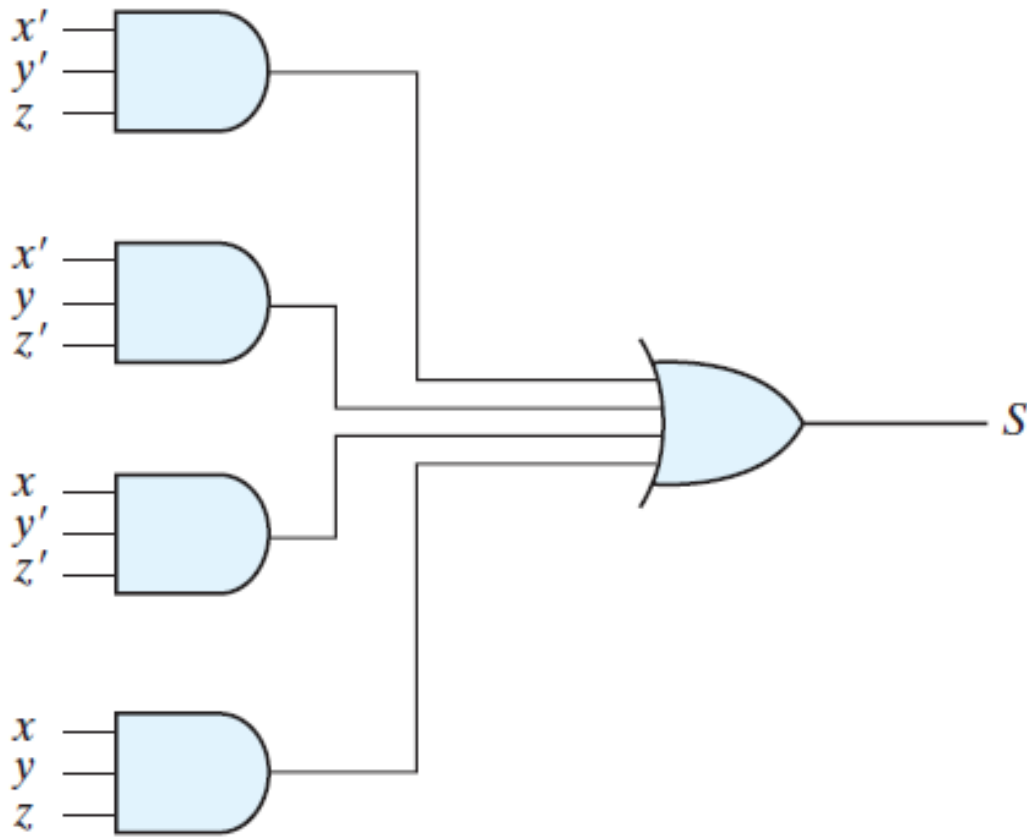
► Elde bitini de işleme katarak Bir bitlik iki sayıyı toplayacak devreyi tasarlayınız

Full Adder

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



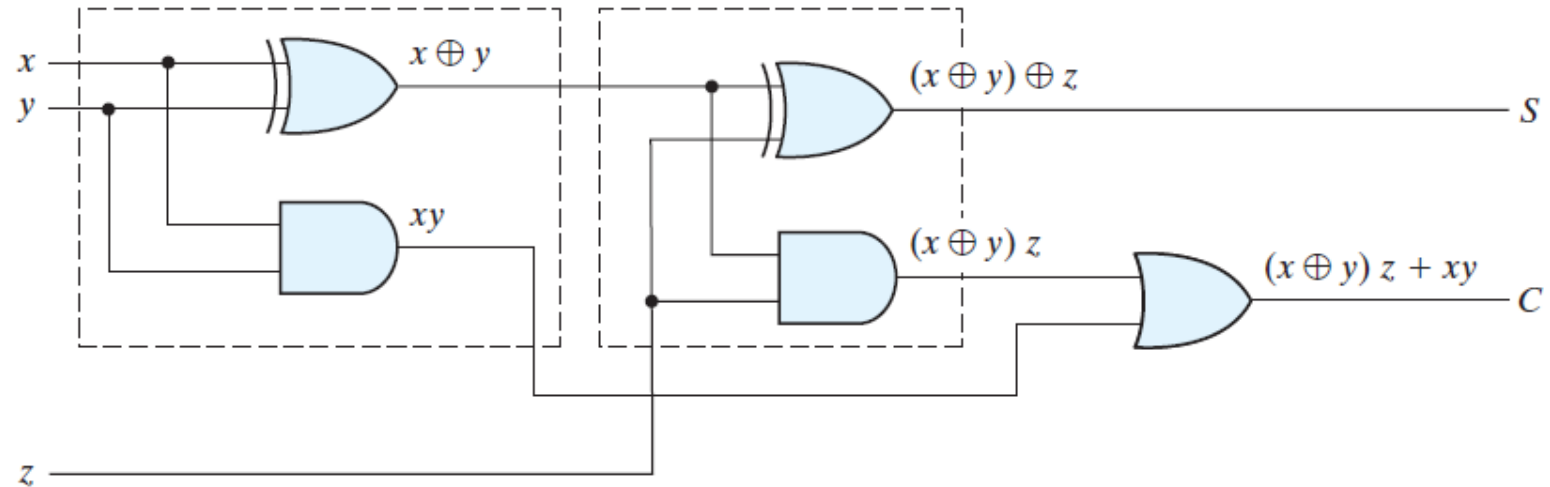
## 2- Tam Toplayıcı



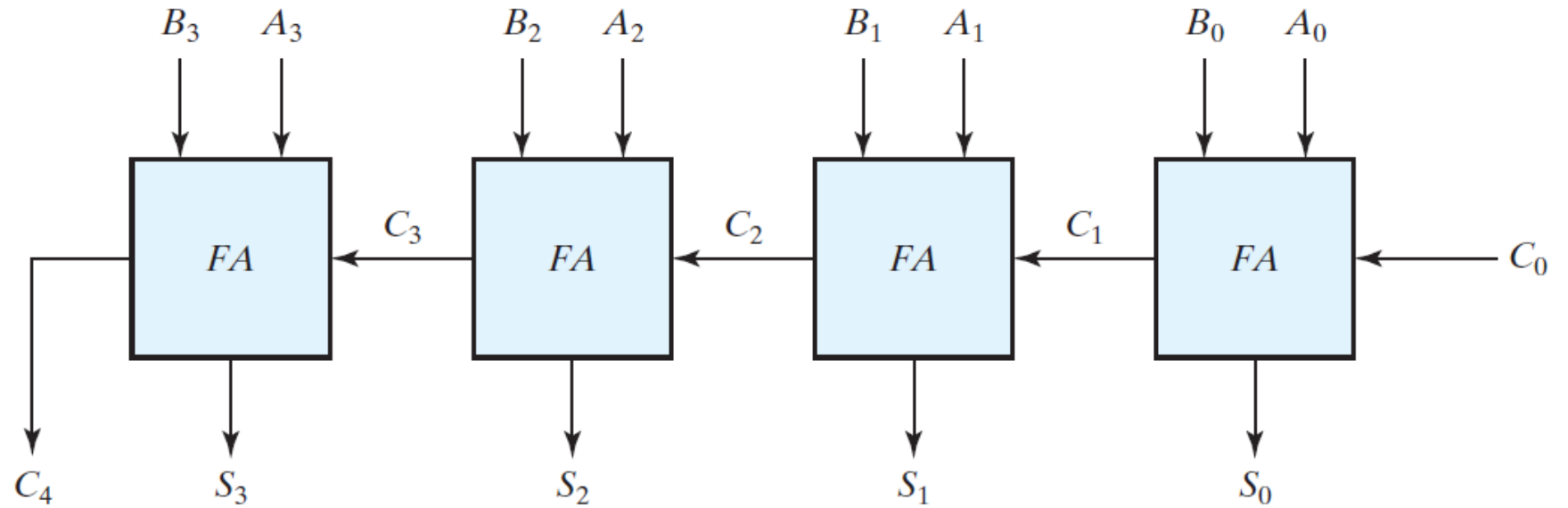
# Tam Toplayıcı

$$\begin{aligned} S &= z \oplus (x \oplus y) \\ &= z'(xy' + x'y) + z(xy' + x'y)' \\ &= z'(xy' + x'y) + z(xy + x'y') \\ &= xy'z' + x'yz' + xyz + x'y'z \end{aligned}$$

$$C = z(xy' + x'y) + xy = xy'z + x'yz + xy$$



## 4 Bitlik Toplayıcı



# Yarı Çıkarıcı

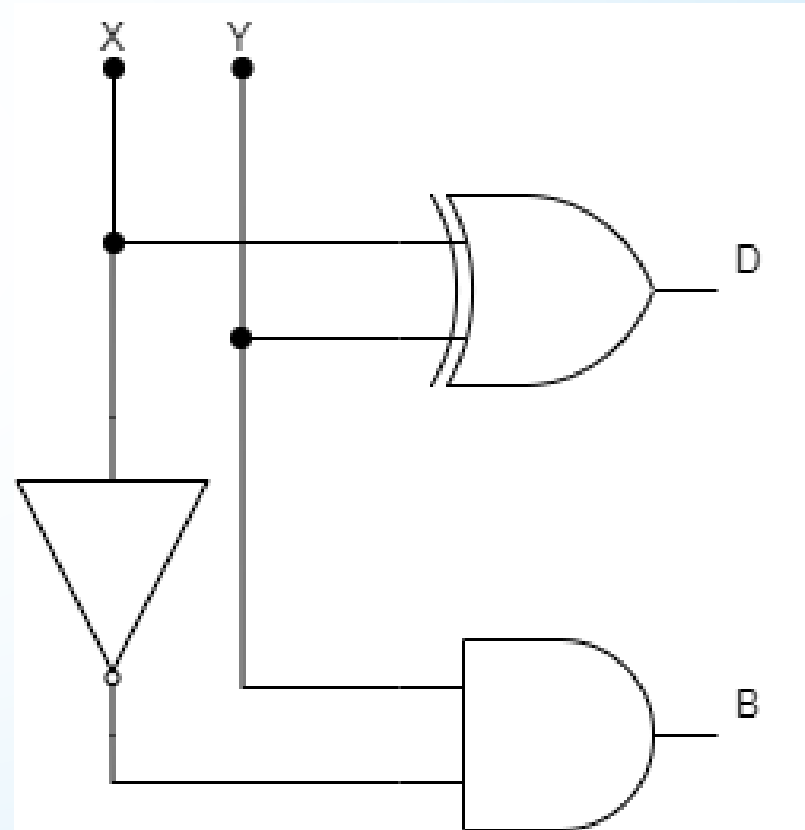
X-Y işlemini yapacak olursak;

X	Y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$$B = X'Y$$

$$D = X'Y + XY'$$

$$D = X \oplus Y$$



# Tam Çıkarıcı

X-Y-Z işlemi yapacak olursak;

x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$D = x \oplus y \oplus z$$

$$B = x'y + x'z + yz$$

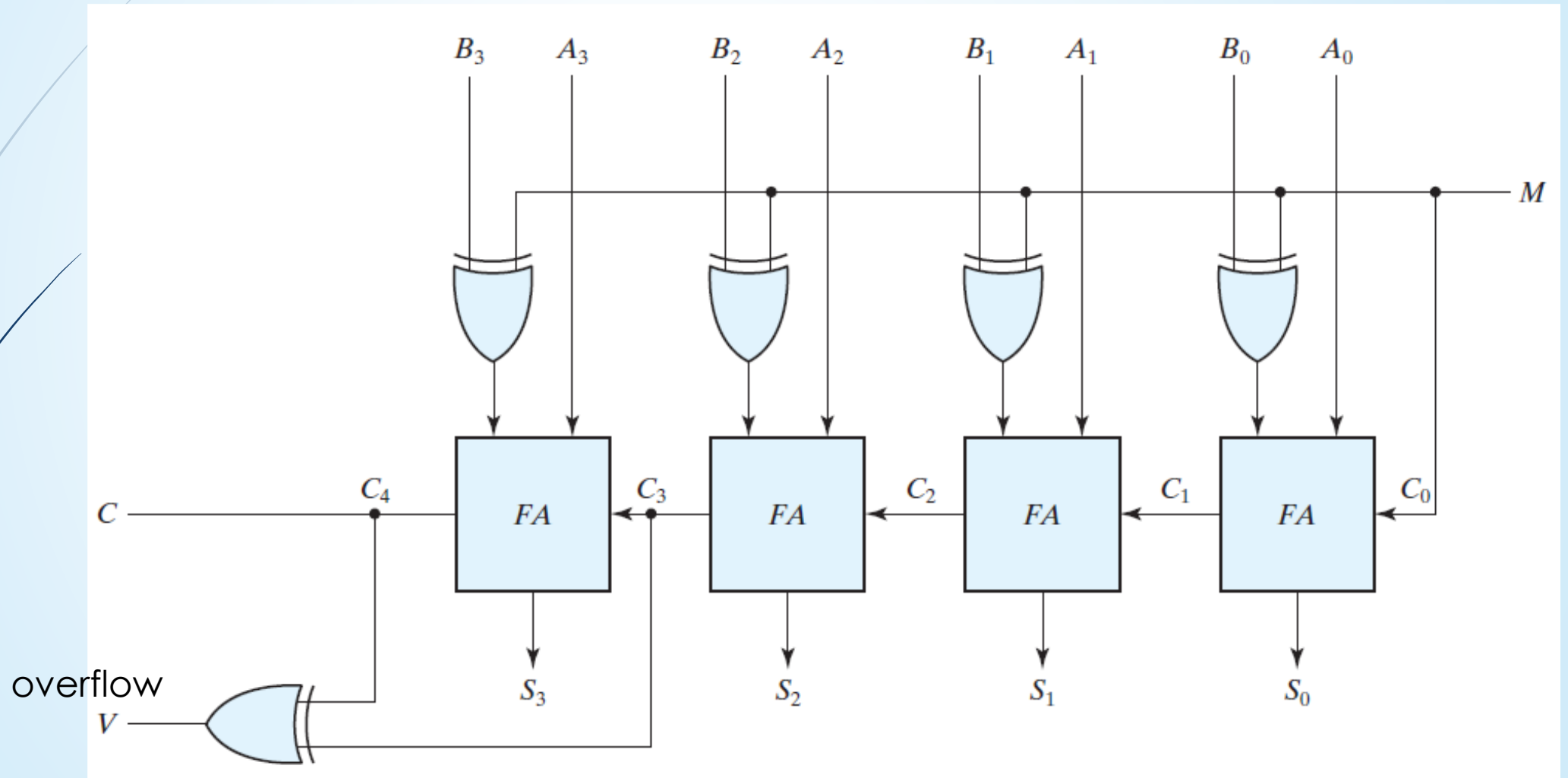
D

		$y,z$				
		$x$	00	01	11	10
0	1	0				
		1				

B

		$y,z$				
		$x$	00	01	11	10
0	1	0				
		1				


## İkinin Tümleyenine Göre Çıkarma Devresi







# KARŞILAŞTIRICILAR

- Yarı karşılaştırıcı
  - Tam karşılaştırıcı
- 

# Yarı Karşılaştırıcı

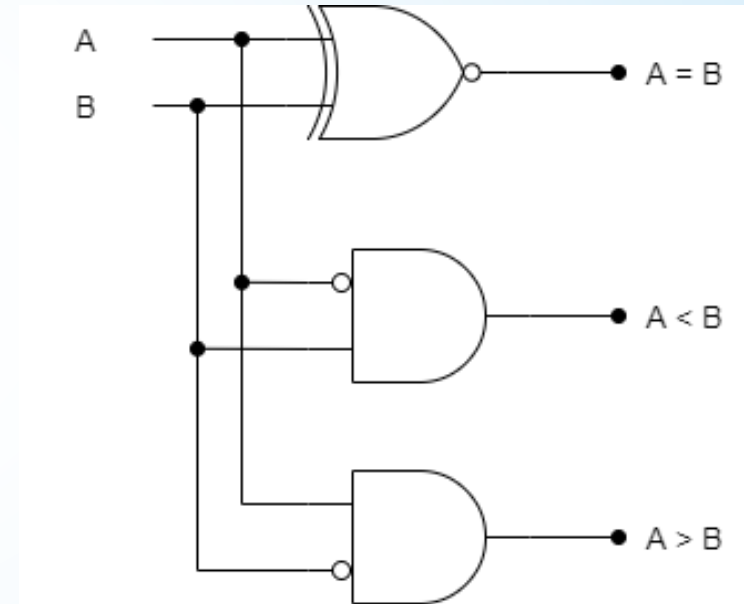
A	B	A=B	A≠B
0	0	1	0
0	1	0	1
1	0	0	1
1	1	1	0

$$A = B \text{ çıkışı} \Rightarrow = \overline{(A \oplus B)}$$

$$A \neq B \text{ çıkışı} \Rightarrow = (A \oplus B)$$

# Tam Karşılaştırıcı

A	B	A<B	A>B
0	0	0	0
0	1	1	0
1	0	0	1
1	1	0	0



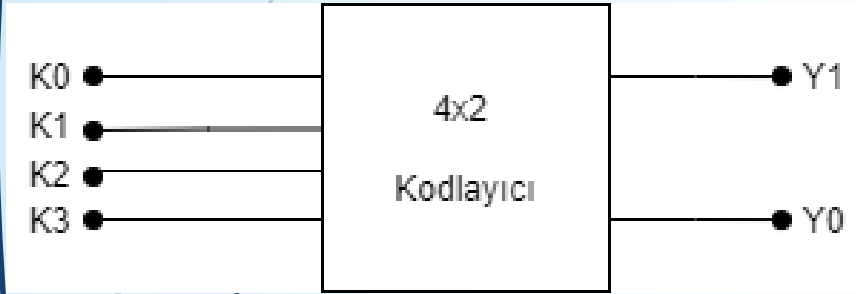


# Soru:

- 4 bitlik iki tane ikili sayının karşılaştırılmasını yapacak lojik devreyi tasarlayınız

# Kodlayıcı (Encoder)

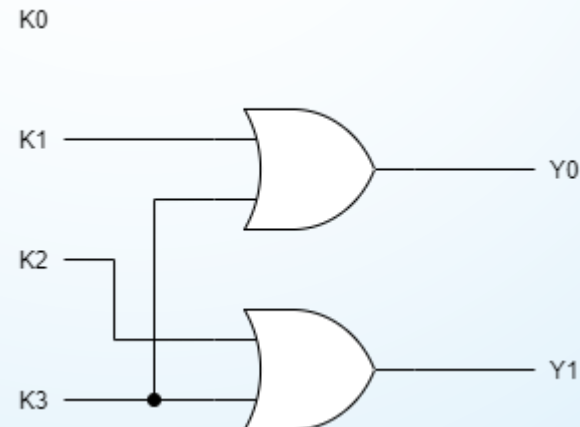
4 Tuşlu yapı



Girişler				Çıkışlar	
K3	K2	K1	K0	Y1	Y0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

K3,K2 \ K1,K0				
	00	01	11	10
00				
01				
11				
10				

$$Y0 = K1 + K3$$



K3,K2 \ K1,K0				
	00	01	11	10
00				
01				
11				
10				

$$Y1 = K2 + K3$$

# 8x3 Kodlayıcı

TABLO 5-3  
Sekizliden İkiliye Kodlayıcının Doğruluk Tablosu

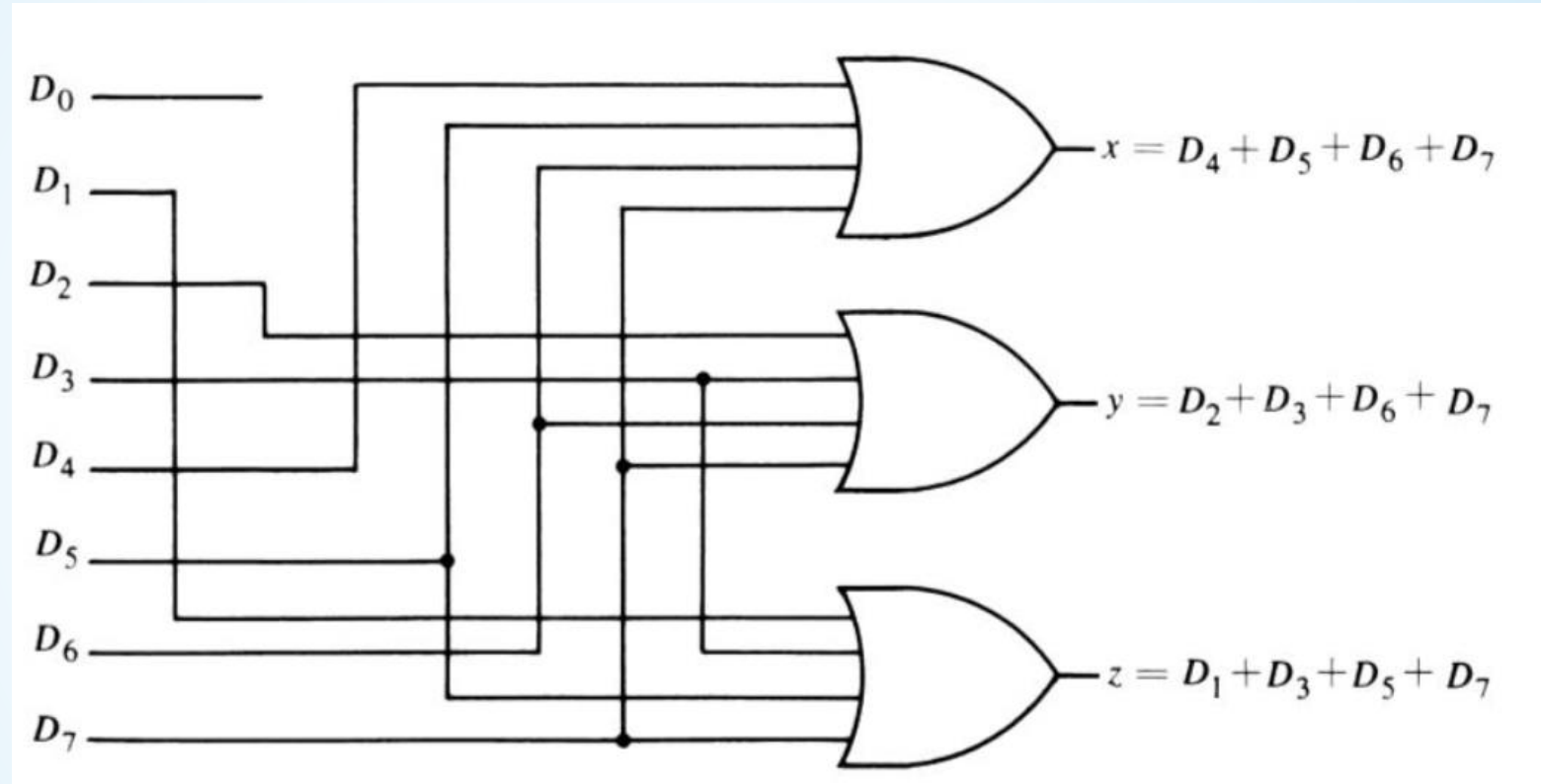
Girişler								Çıkışlar		
$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$	$x$	$y$	$z$
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$$z = D_1 + D_3 + D_5 + D_7$$

$$y = D_2 + D_3 + D_6 + D_7$$

$$x = D_4 + D_5 + D_6 + D_7$$

## 8x3 kodlayıcı

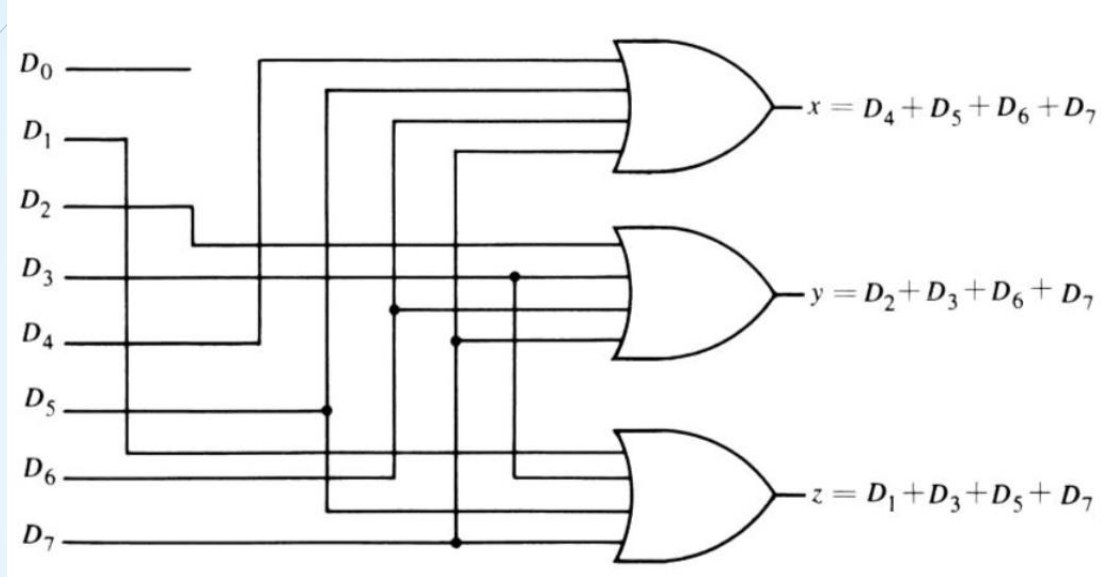


$$z = D_1 + D_3 + D_5 + D_7$$

$$y = D_2 + D_3 + D_6 + D_7$$

$$x = D_4 + D_5 + D_6 + D_7$$

# 8x3 kodlayıcı



- Böyle bir sistemde birden çok tuşa basıldığında yada birden fazla giriş 1 olduğunda (Örneğin:  $D_6$ - $D_7$ ) çıkış ne olacaktır?  
(Birine öncelik vermek gerekebilir)
- $D_0$  tuşuna basıldığını nereden bileceğiz?  
(Sisteme değer girildiğini bildiren ayrı bir çıkış eklenebilir)

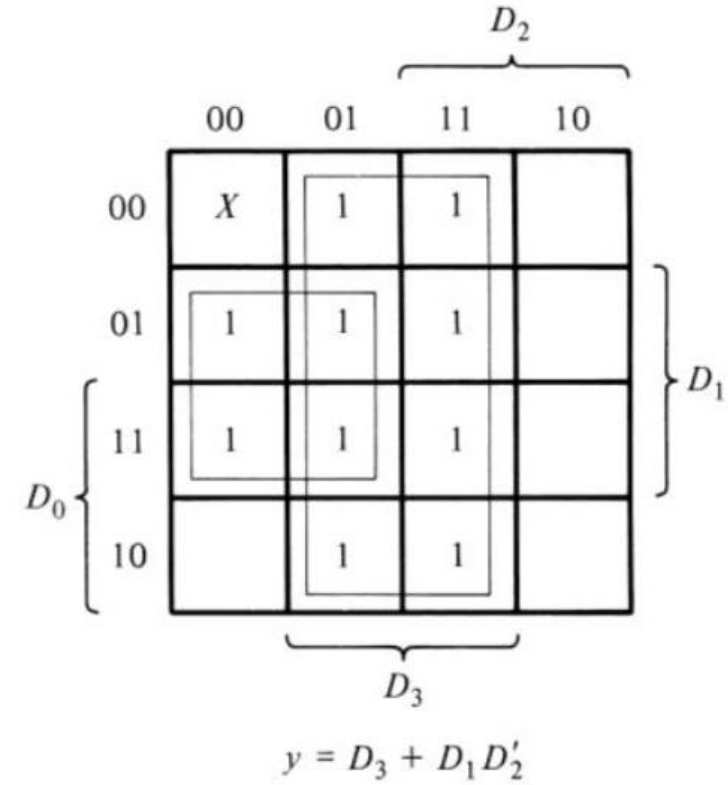
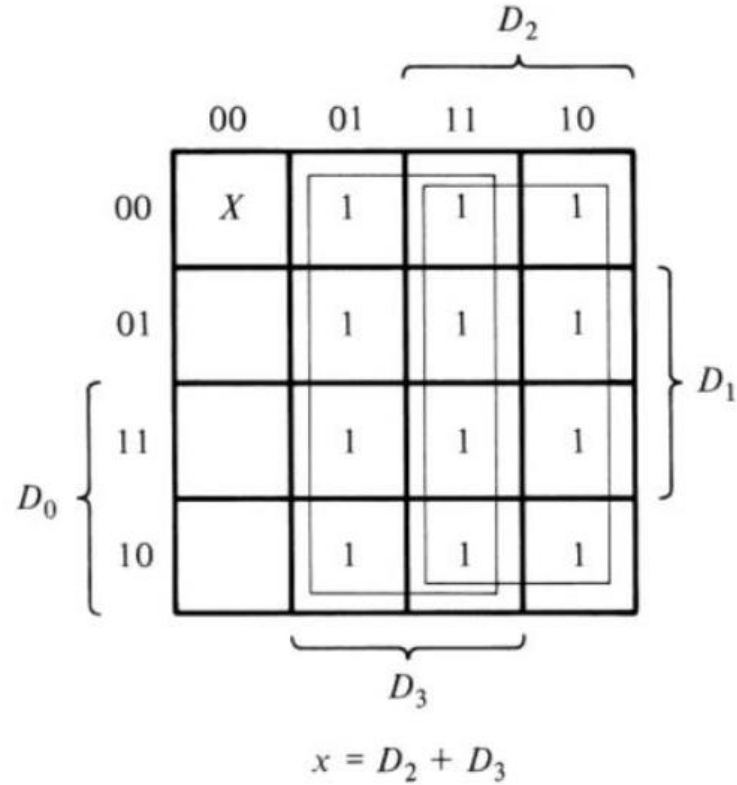


# Öncelik (Priority) Kodlayıcı

Bir Öncelik Kodlayıcısının Doğruluk Tablosu

Girişler				Çıkışlar		
$D_0$	$D_1$	$D_2$	$D_3$	$x$	$y$	$V$
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

# Öncelik (Priority) Kodlayıcı



ŞEKİL 5-14

Bir öncelik kodlayıcısı için diyagramlar

# Öncelik (Priority) Kodlayıcı

$$x = D_2 + D_3$$

$$y = D_3 + D_1 D_2'$$

$$V = D_0 + D_1 + D_2 + D_3$$

