

SAYISAL TASARIM BOOLEAN FONKSİYONLARININ SADELEŞTİRİLMESİ

Hazırlayan:

Yard. Doç. Dr. Metin ARTIKLAR

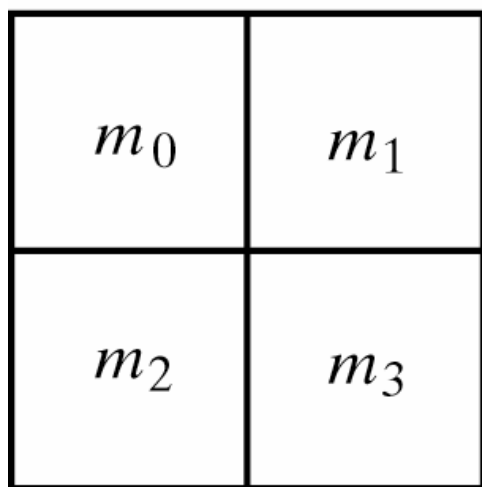
FONKSİYONLARIN SADELEŞTİRİLMESİ

- Boolean fonksiyonlarının sadeleştirilmesi için farklı yöntemler kullanılabilir. Bunlar:
 - Cebirsel işlemler
 - Karnaugh metodu (K-map Metodu)
 - Bilgisayar programları

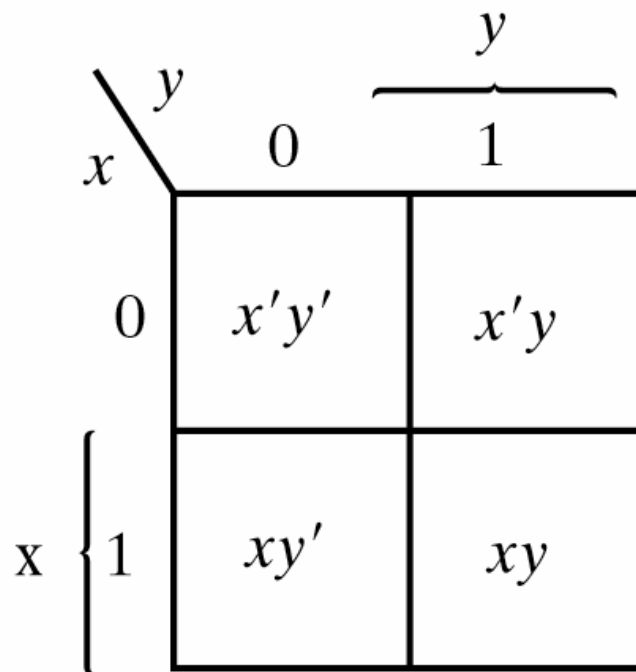
K-Map Metodu

- Bu yöntem herhangi bir fonksiyonun doğruluk tablosunda 1'e karşılık gelen mintörmelerin K-map adı verilen bir tablonun içerisine yerleştirilip gruplanması ile gerçekleştirilir

2 Değişkenli Fonksiyonlar için K-Map Metodu



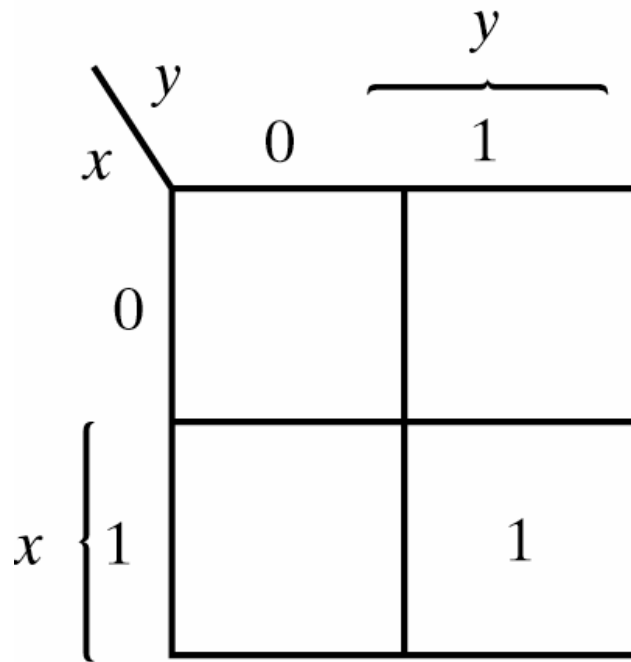
(a)



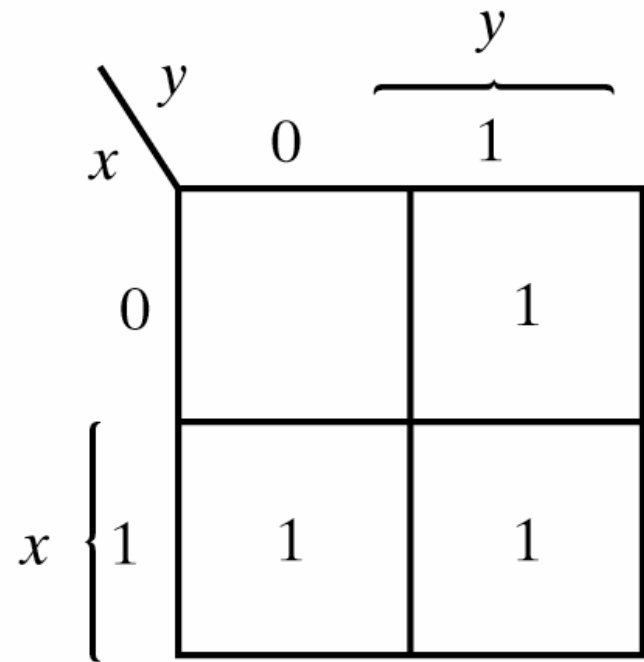
(b)

Fig. 3-1 Two-variable Map

2 Değişkenli Fonksiyonlar için K-Map Metodu



(a) xy



(b) $x + y$

Fig. 3-2 Representation of Functions in the Map

2 Değişkenli Fonksiyonlar için K-Map Metodu

- K-map içerisinde mintörmelerin sayısı kadar kutucuk bulunur.
 - Tabloda tek olarak gruplanan bir minterm iki değişkenli bir çarpım terimine
 - İkili olarak gruplanan mintermler tek değişkenli bir çarpım terimine
 - 4'lü olarak gruplanan mintermler sabit 1 sayısına karşılık gelir

3 Değişkenli K-Map'ler

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

(a)

		y			
		xz			
		0 0	0 1	1 1	1 0
x	0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
	1	$xy'z'$	$xy'z$	xyz	xyz'
		z			

(b)

Fig. 3-3 Three-variable Map

3 Değişkenli K-Map'ler

- Tablonun üstünde yz için olan sıralamanın 00 01 11 10 (gri kod) şeklinde olduğuna dikkat ediniz. Bunun sebebi bu şekliyle mintermler yanyana gelince sadeleştirme yapılabilmesidir
- $m_5 + m_7 = xy'z + xyz = xz(y' + y) = xz$

3 Değişkenli K-Map'ler

➤ m_0 'ın m_2 'ye

➤ m_4 'ün m_6 'ya

➤ m_0 ve m_4 'ün m_2 ve m_6 'ya

komşu olduğuna dikkat ediniz

- $m_0 + m_2 = x'y'z' + x'yz' = x'z'(y' + y) = x'z'$
- $m_4 + m_6 = xy'z' + xyz' = xz'(y' + y) = xz'$
- $m_0 + m_2 + m_4 + m_6 = x'y'z' + x'yz' + xy'z' + xyz' = xz' + x'z' \rightarrow z'(x + x') \rightarrow z'$

3 Değişkenli K-Map'ler

- Tablodaki:
 - Tek olarak gruplanan bir minterm üç değişkenli bir çarpım terimine
 - İkili olarak gruplanan mintermler iki değişkenli bir çarpım terimine
 - Dörtlü olarak gruplanan mintermler 1 değişkenli bir çarpım terimine
 - 8'li olarak gruplanan mintermler sabit 1 sayısına karşılık gelir

3 Değişkenli K-Map'ler

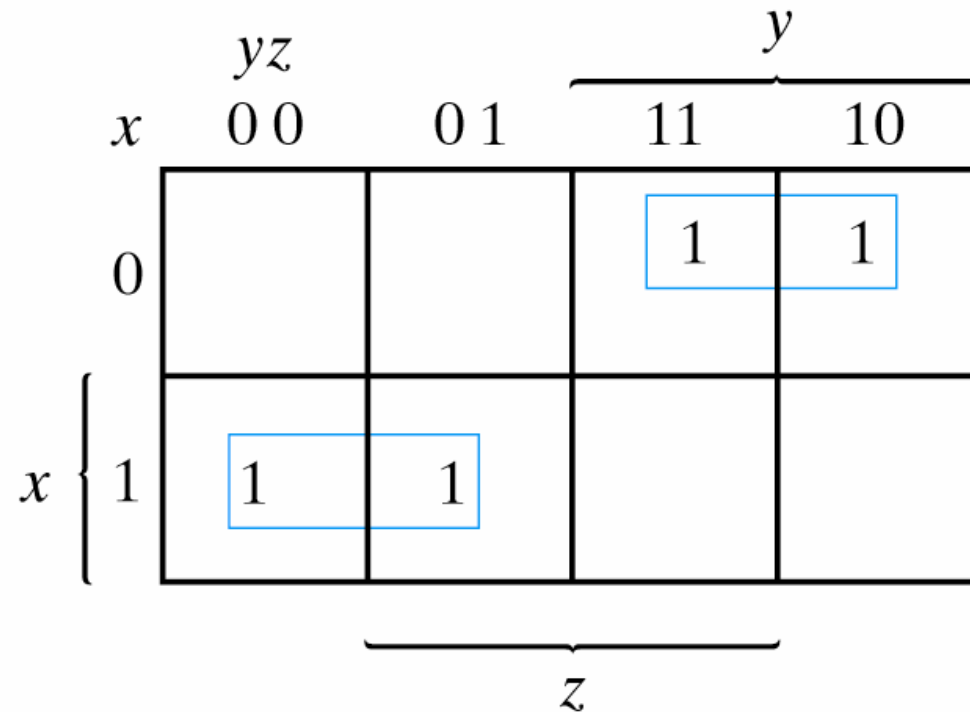


Fig. 3-4 Map for Example 3-1; $F(x, y, z) = \Sigma(2, 3, 4, 5) = x'y + xy'$

3 Değişkenli K-Map'ler

		yz		y	
x		0 0	0 1	1 1	1 0
x	0			1	
	1	1		1	1
		z			

Fig. 3-5 Map for Example 3-2; $F(x, y, z) = \Sigma(3, 4, 6, 7) = yz + xz'$

3 Değişkenli K-Map'ler

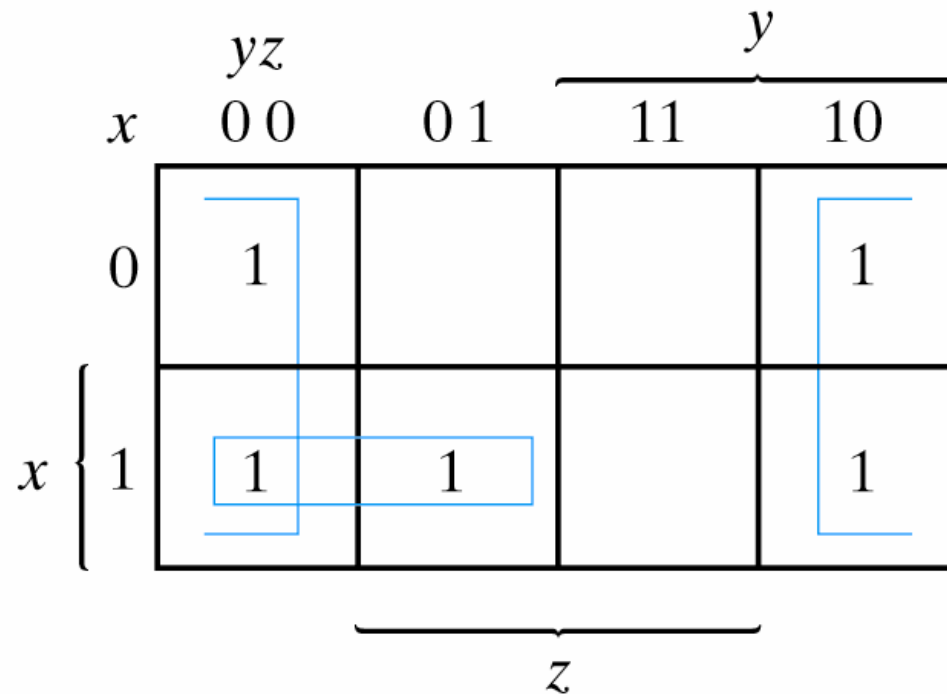


Fig. 3-6 Map for Example 3-3; $F(x, y, z) = \Sigma(0, 2, 4, 5, 6) = z' + xy'$

3 Değişkenli K-Map'ler

- $F(A,B,C) = A'C + A'B + BC + AB'C$
- Fonksiyonunun doğruluk tablosunu oluşturunuz, fonksiyonu mintörmelerin toplamı olarak ifade ediniz ve K-Map kullanarak sadeleştiriniz

3 Değişkenli K-Map'ler

A	B	C	A'C	A'B	BC	AB'C	F(A,B,C)
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1
0	1	0	0	1	0	0	1
0	1	1	1	1	1	0	1
1	0	0	0	0	0	0	0
1	0	1	0	0	0	1	1
1	1	0	0	0	0	0	0
1	1	1	0	0	1	0	1

$$F(A,B,C) = A'C + A'B + BC + AB'C$$

3 Değişkenli K-Map'ler

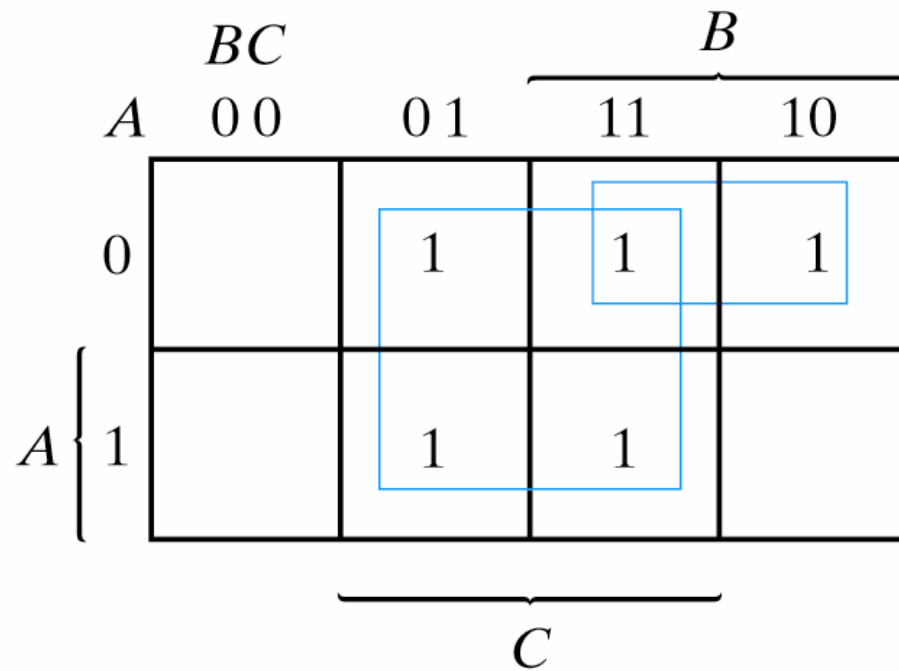


Fig. 3-7 Map for Example 3-4; $A'C + A'B + AB'C + BC = C + A'B$

4 Değişkenli K-Map'ler

- Tek olarak gruplanan bir minterm 4 değişkenli bir çarpım terimine
- İkili olarak gruplanan mintermler 3 değişkenli bir çarpım terimine
- Dörtlü olarak gruplanan mintermler 2 değişkenli bir çarpım terimine
- Sekizli olarak gruplanan mintermler 1 değişkenli bir çarpım terimine
- 16'lı olarak gruplanan mintermler sabit 1 sayısına karşılık gelir

4 Değişkenli K-Map'ler

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

(a)

		yz		y	
		00	01	11	10
w	00	$w'x'y'z'$	$w'x'y'z$	$w'x'yz$	$w'x'yz'$
	01	$w'xy'z'$	$w'xy'z$	$w'xyz$	$w'xyz'$
	11	$wxy'z'$	$wxy'z$	$wxyz$	$wxyz'$
	10	$wx'y'z'$	$wx'y'z$	$wx'yz$	$wx'yz'$
		z			

(b)

Fig. 3-8 Four-variable Map

4 Değişkenli K-Map'ler

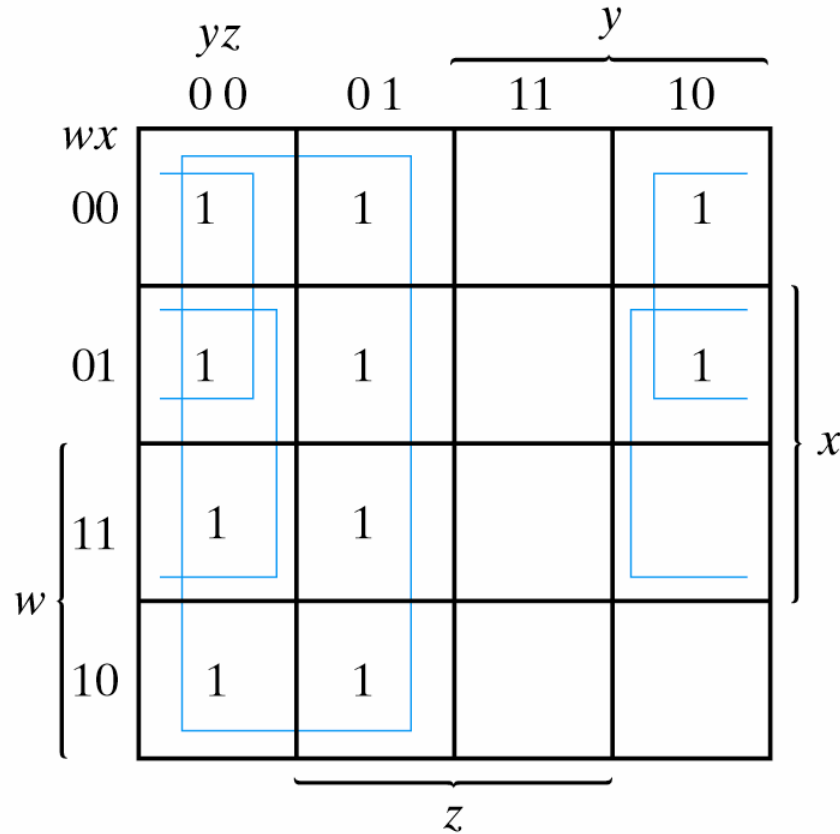


Fig. 3-9 Map for Example 3-5; $F(w, x, y, z)$

$$= \Sigma (0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14) = y' + w'z' + xz'$$

4 Değişkenli K-Map'ler

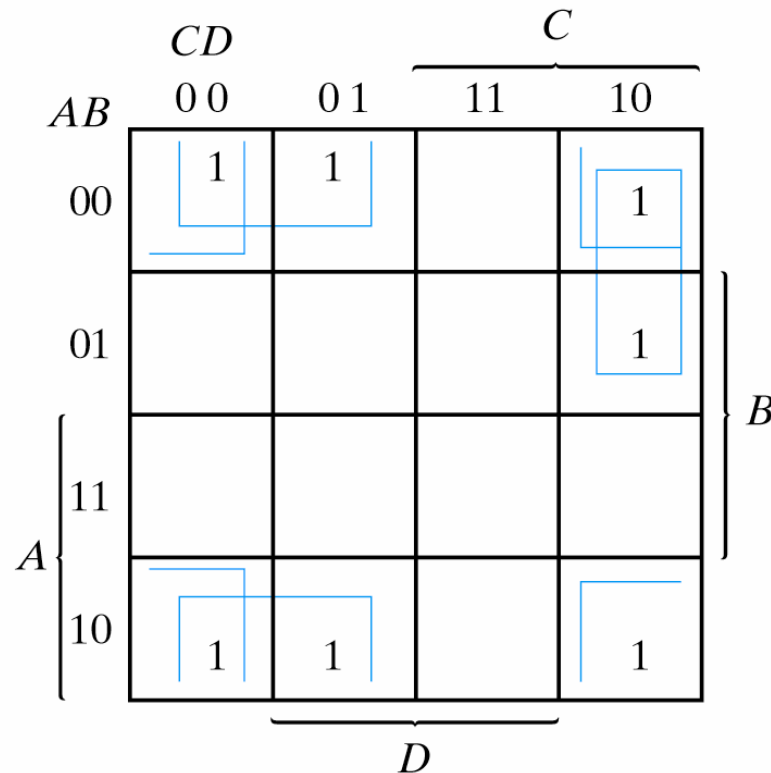


Fig.3-10 Map for Example 3-6; $A'B'C + B'CD' + A'BCD' + AB'C' = B'D' + B'C' + A'CD'$

ASAL ÇARPANLAR

- Asal çarpanlar bir tabloda maksimum sayıda mintermlerin bir araya getirilmesi ile elde edilen çarpım terimlerine denir
- Eğer herhangi bir minterm sadece bir asal çarpan içerisinde gözüküyorsa, bu asal çarpan temel asal çarpan olarak adlandırılır

ASAL ÇARPANLAR

$$F(A,B,C,D) = \sum (0,2,3,5,7,8,9,10,11,13,15)$$

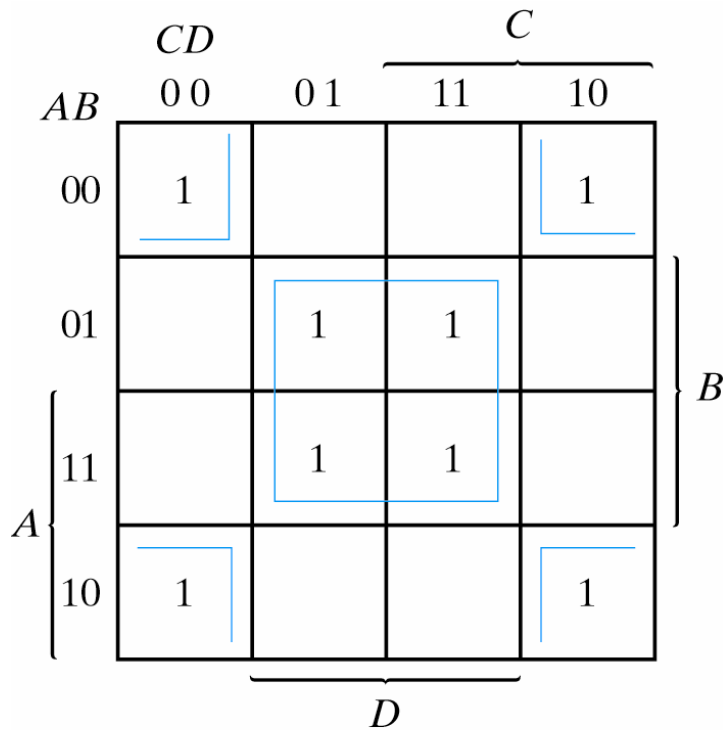
$$F = BD + B'D' + CD + AD$$

$$F = BD + B'D' + CD + AB'$$

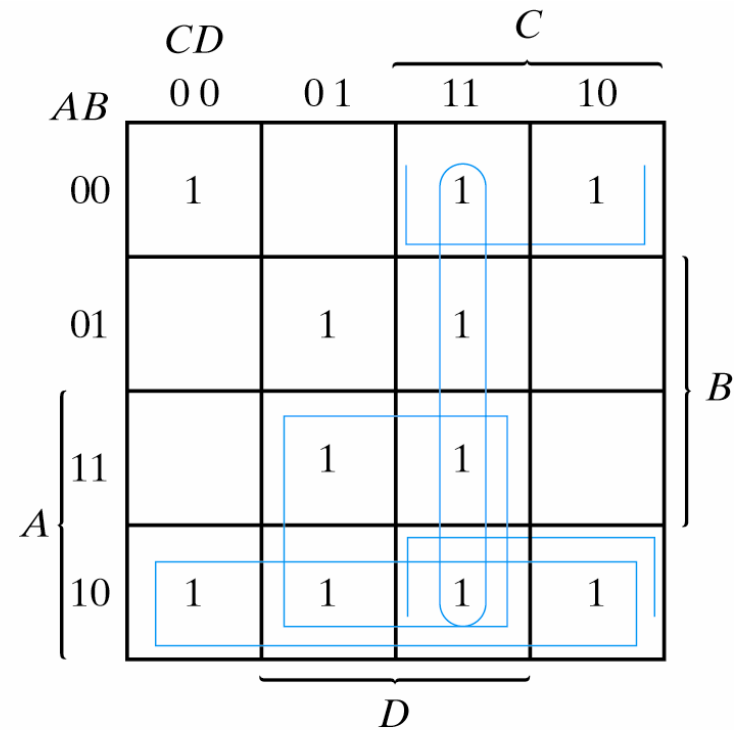
$$F = BD + B'D' + B'C + AD$$

$$F = BD + B'D' + B'C + AB'$$

ASAL ÇARPANLAR



(a) Essential prime implicants
 BD and $B'D'$



(b) Prime implicants CD , $B'C$,
 AD , and AB'

Fig. 3-11 Simplification Using Prime Implicants

Boolean Fonksiyonlarının Çarpımların Toplamların Yöntemi ile Sadeleştirilmesi

- Şimdiye kadar yapmış olduğumuz örnekler bu yöntemle birer örnekti

Boolean Fonksiyonlarının Toplamların Çarpımı Yöntemi ile Sadeleştirilmesi

- Bu yöntemde önce fonksiyonun değili, F' , bulunarak ve F' fonksiyonu için K-Map kullanılarak en sade ifade bulunur.
- Daha sonra, $F = (F')'$ olduğundan, elde edilen ifadenin değili alınır
- Böylece F fonksiyonunun toplamların çarpımı şeklinde ifade edilmiş en sade şekli bulunur

Boolean Fonksiyonlarının Toplamların Çarpımı Yöntemi ile Sadeleştirilmesi

	yz 00	yz 01	yz 11	yz 10
x 0	0	0	1	1
x 1	0	1	1	1

$$F(x,y,z) = \sum (2, 3, 5, 6, 7)$$

	yz 00	yz 01	yz 11	yz 10
x 0	1	1	0	0
x 1	1	0	0	0

$$F'(x,y,z) = \sum (0, 1, 4)$$

$$F'(x,y,z) = x'.y' + y.z$$

$$F = (F')' = (x+y).(y'+z')$$

Etkisiz (Don't Care) Koşullar

- Bu koşullar bir Boolean fonksiyonunun doğruluk tablosunda 0 veya 1 olmalarının önemsiz olduğu mintermlere verilen addır
- Doğruluk tablosunda yada K-Map'lerde bu mintermlere karşılık gelen yerlere 'x' işareti konulması ile belirtilirler
- Fonksiyonlar sadeleştirilirken K-Mapler içerisinde istenilen miktarları 1 ve diğer kısımları 0 olarak düşünülebildikleri için sadeleştirme işleminin daha az sayıda değişken ile yapılmasını sağlarlar

$$F(x,y,z) = \sum (0, 3, 5)$$

$$d(x,y,z) = \sum (1, 6, 7)$$

	yz 00	yz 01	yz 11	yz 10
x 0	1	x	1	0
x 1	0	1	x	x

	yz 00	yz 01	yz 11	yz 10
x 0	1	0	1	0
x 1	0	1	0	0

Etkisiz koşullar 0 olarak alındığında
(veya etkisiz koşulların olmadığı
düşünülünce)

$$F = x'y'z' + x'yz + xy'z$$

	yz 00	yz 01	yz 11	yz 10
x 0	1	1	1	0
x 1	0	1	1	x

Etkisiz koşullar göz önüne
alındığında

$$F = x'y' + z$$

$$F(w,x,y,z) = \sum (1, 3, 7, 11, 15)$$

$$d(w,x,y,z) = \sum (0, 2, 5)$$

		y			
		yz			
		00	01	11	10
w	wx	X	1	1	X
	01	0	X	1	0
	11	0	0	1	0
	10	0	0	1	0

z

x

$$(a) F = yz + w'x'$$

		y			
		yz			
		00	01	11	10
w	wx	X	1	1	X
	01	0	X	1	0
	11	0	0	1	0
	10	0	0	1	0

z

x

$$(a) F = yz + w'z$$

Fig. 3-17 Example with don't-care Conditions

Boolean Fonksiyonlarının NAND Kapıları İle Gerçeklenmeleri

- Sayısal devreler AND, OR ve NOT kapıları yerine genellikle NAND ve NOR kapıları ile gerçekleştirilir
- Bunun sebebi bu mantık kapılarının diğerlerine göre daha kolay ve ucuz bir şekilde üretilebilmeleridir
- AND, OR ve NOT kapıları kullanılarak gerçekleştirilen bir sayısal devre bazı kurallar uygulanarak NAND kapıları ile gerçekleştirilebilir

Boolean Fonksiyonlarının NAND Kapıları İle Gerçeklenmeleri

- Dönüştürme işleminin gerçekleştirilmesi için AND, OR ve NOT kapılarının NAND kapıları ile gerçekleştirilmesi ve devredeki eşdeğerleri ile değiştirilmeleri yeterlidir
- Bu üç kapının NAND kapıları ile gerçekleştirilmiş şekilleri aşağıda verilmiştir

AND, OR ve NOT Kapılarının NAND Kapıları ile Gerçeklenmiş Eşdeğerleri

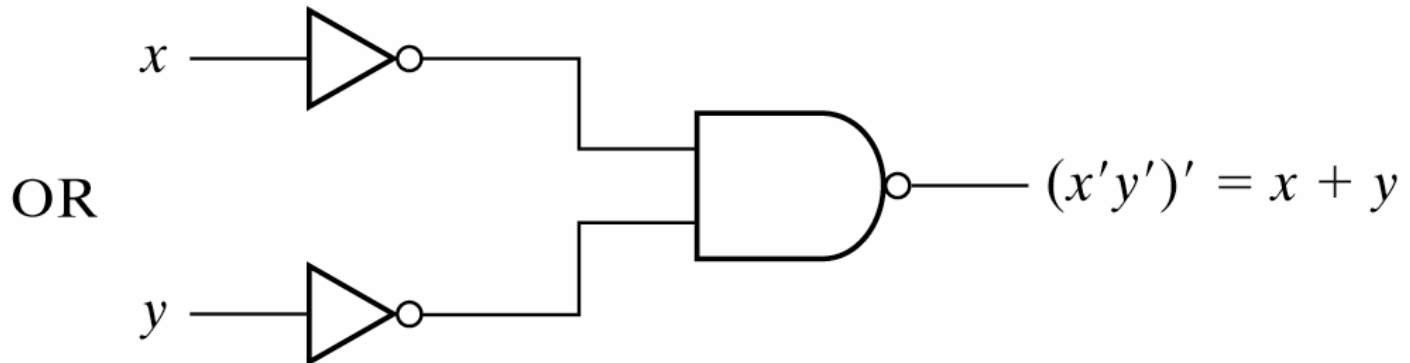
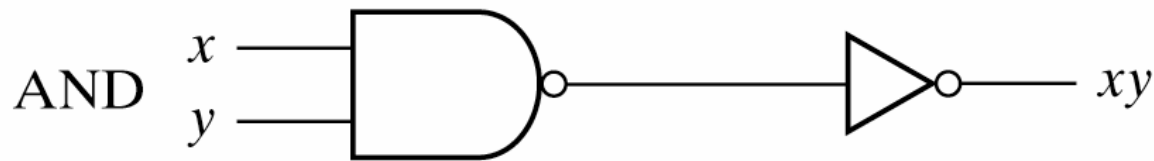
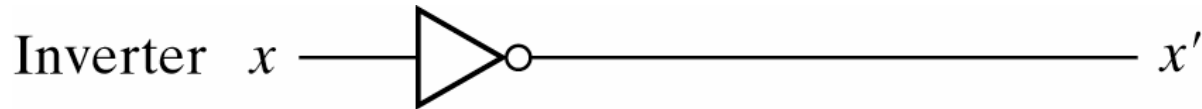
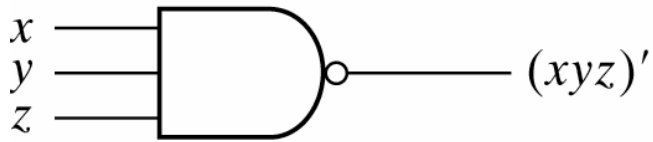
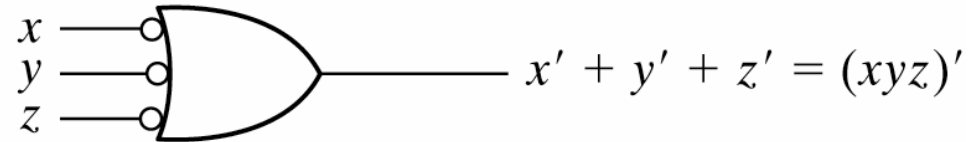


Fig. 3-18 Logic Operations with NAND Gates

NAND Kapısının 2 Farklı Gösterimi (2 Farklı Sembolü)



(a) AND–invert



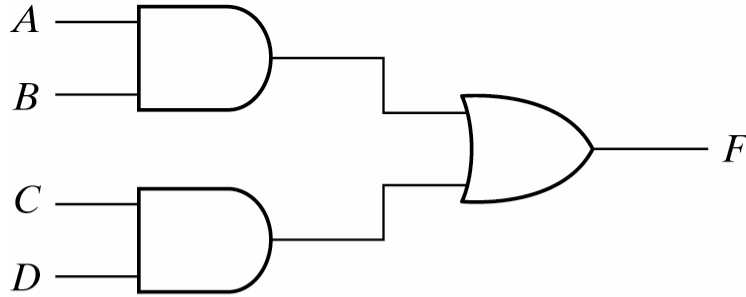
(b) Invert–OR

Fig. 3-19 Two Graphic Symbols for NAND Gate

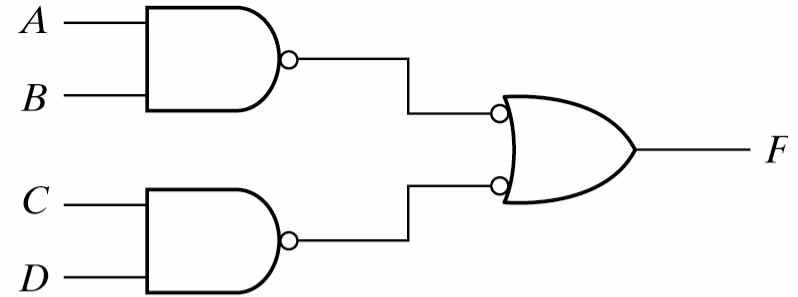
Boolean Fonksiyonlarının NAND Kapıları İle 2 Seviyeli Gerçeklenmesi

- $F = AB + CD$
- Fonksiyonunun NAND kapıları ile iki farklı gerçekleştirme yöntemi aşağıda gösterilmiştir

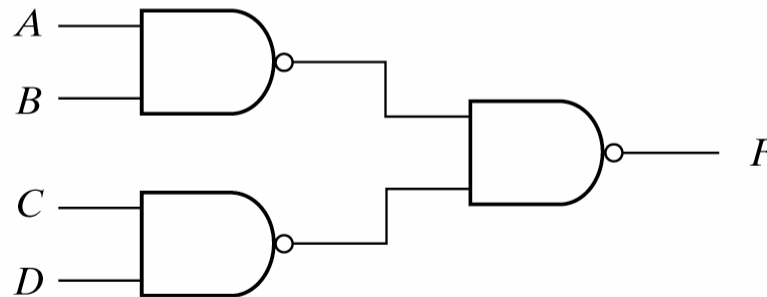
Boolean Fonksiyonlarının NAND Kapıları İle 2 Seviyeli Gerçeklenmesi



(a)



(b)



(c)

Fig. 3-20 Three Ways to Implement $F = AB + CD$

Boolean Fonksiyonlarının NAND Kapıları İle 2 Seviyeli Gerçeklenmesi

- Genel olarak söylemek gerekirse Boolean fonksiyonlarının iki seviyeli olarak NAND kapıları ile gerçekleştirilmesi için aşağıdaki 4 aşama uygulanır

Boolean Fonksiyonlarının NAND Kapıları Gerçeklenmesi

- Fonksiyon AND, OR ve NOT kapıları ile gerçekleştirilir. Daha sonra:
- Tüm AND kapıları AND-invert sembolü ile gösterilen NAND kapıları ile değiştirilir
- Tüm OR kapıları invert-OR sembolü ile gösterilen NAND kapısı ile değiştirilir
- Kapı çıkışlarındaki tüm '0' sembollerinin bir ikincisi ile dengelendiği kontrol edilir.
Dengelenmemişse araya bir NOT kapısı eklenir

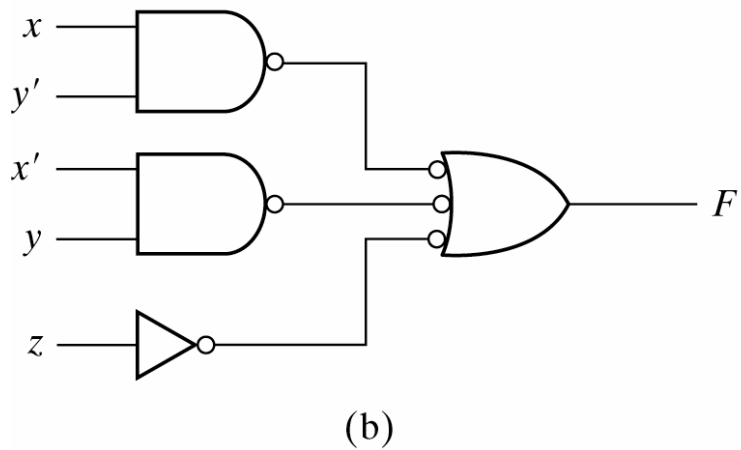
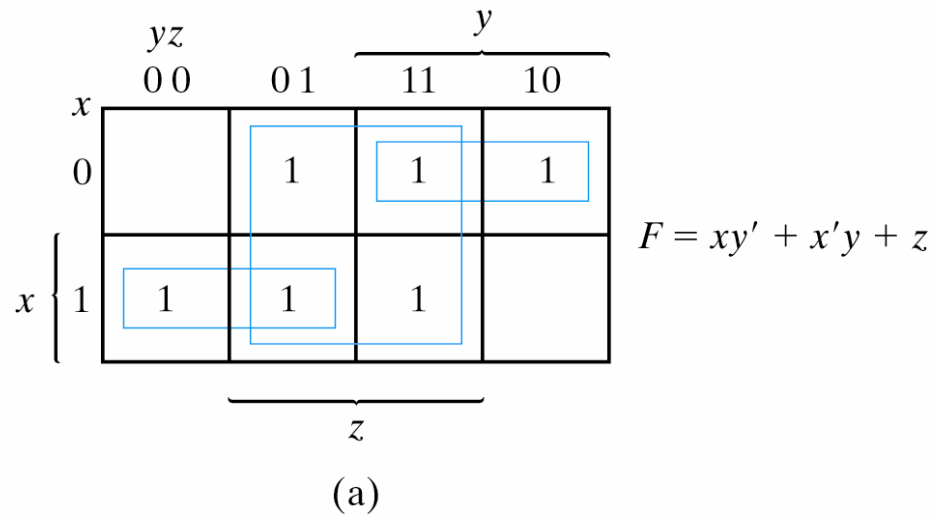
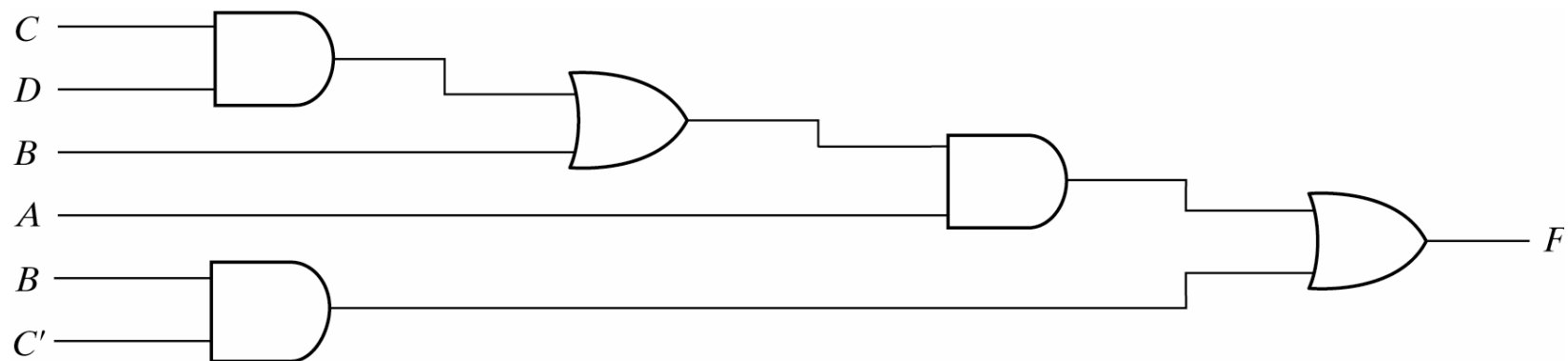
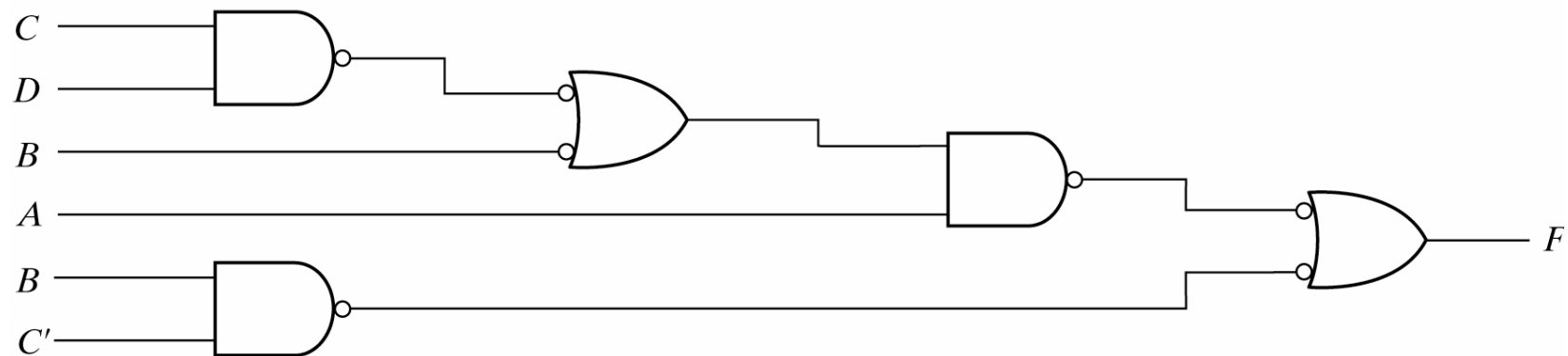


Fig. 3-21 Solution to Example 3-10

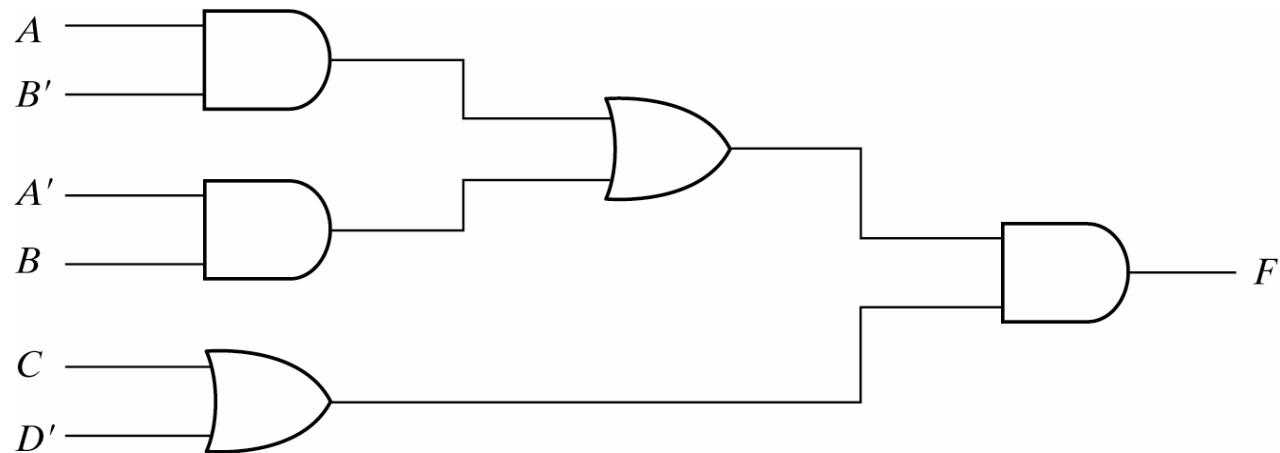


(a) AND-OR gates

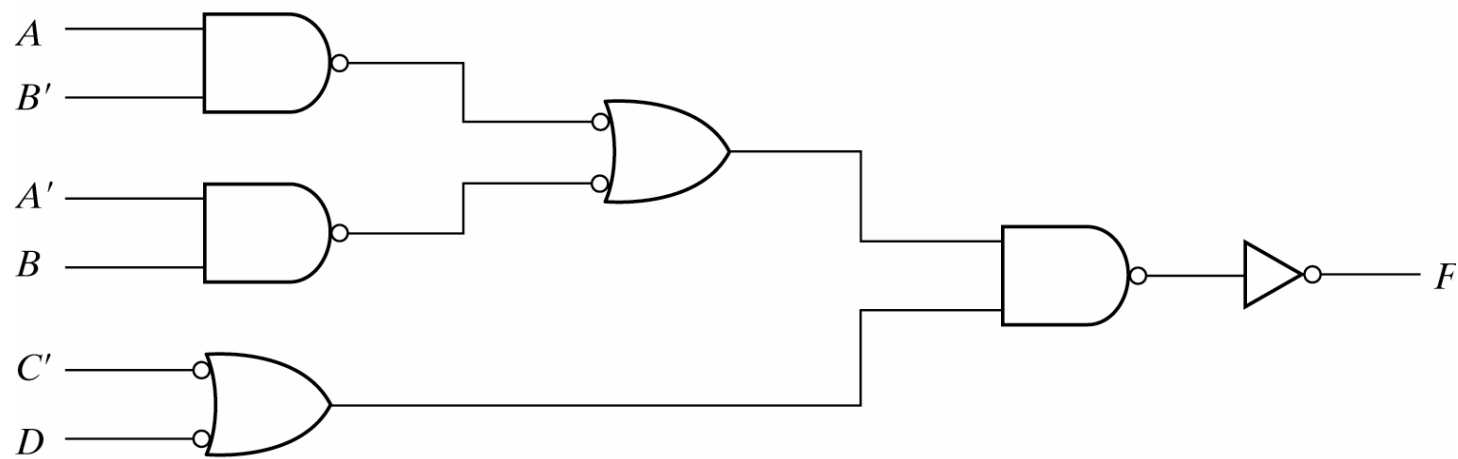


(a) NAND gates

Fig. 3-22 Implementing $F = A(CD + B) + BC$



(a) AND-OR gates



(b) NAND gates

Fig. 3-23 Implementing $F = (AB' + A'B)(C + D')$

Ex-OR ve Ex-NOR Kapıları

- Ex-OR $\rightarrow x \oplus y = xy' + x'y$
- Ex-NOR $\rightarrow (x \oplus y)' = (xy' + x'y)' = xy + x'y'$
- Ex-OR şu işlemler için kullanılır

$$x \oplus 0 = x$$

$$x \oplus 1 = x'$$

$$x \oplus x = 0$$

$$x \oplus x' = 1$$

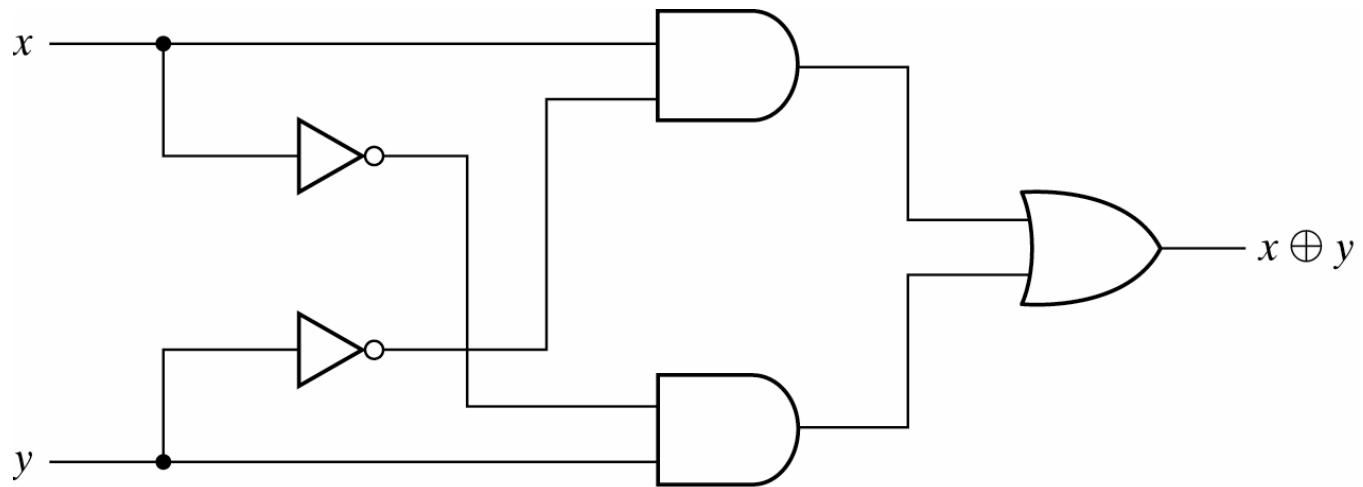
$$x \oplus y' = x' \oplus y = (x \oplus y)'$$

Ex-OR ve Ex-NOR Kapıları

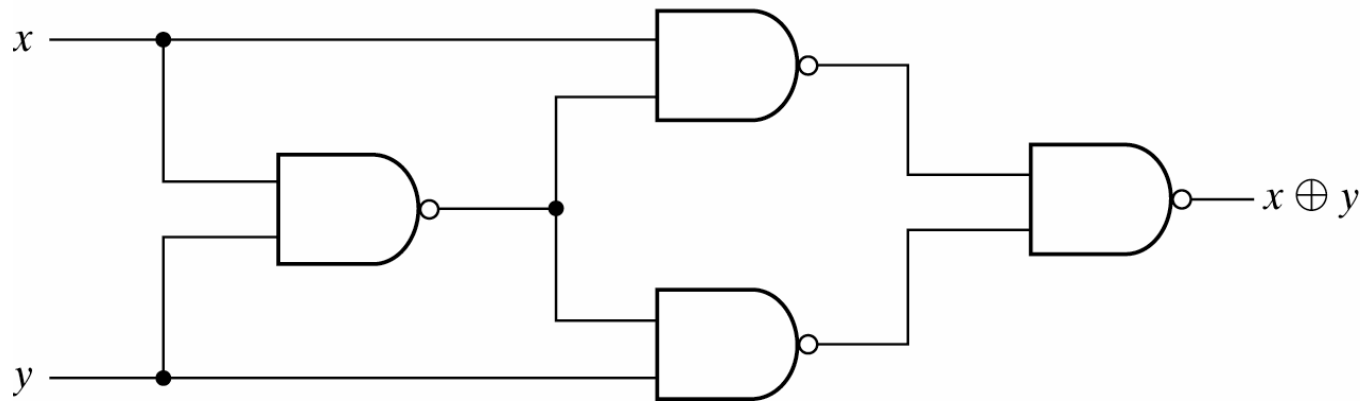
- Her iki operatörde değişme ve birleşme özelliklerinin ikisine de sahip oldukları için 2'den fazla girişe sahip mantık kapıları olarak gerçekleştirilebilirler

Ex-OR Kapısının NAND Kapısı ile Gerçeklenmesi

- 1. Seviye $(x.y)' = x' + y'$
- 2. Seviye $(x.(x' + y'))'$ ile $(y.(x' + y'))'$
- 3. Seviye $[(x.(x' + y'))' . (y.(x' + y'))']'$
$$= (x.(x' + y') + (y.(x' + y'))'$$
$$= x.y' + x'y$$
$$= x \oplus y$$



(a) With AND-OR-NOT gates



(b) With NAND gates

Fig. 3-32 Exclusive-OR Implementations

Ex-OR Kullanılarak Tek ve Çift Eşlik Yöntemi İle Veri Kodlanması

		BC		B	
		00	01	11	10
A	0		1		1
A	1	1		1	

C

(a) Odd function
 $F = A \oplus B \oplus C$

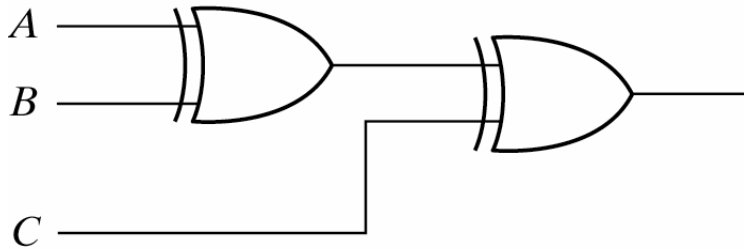
		BC		B	
		00	01	11	10
A	0	1		1	
A	1		1		1

C

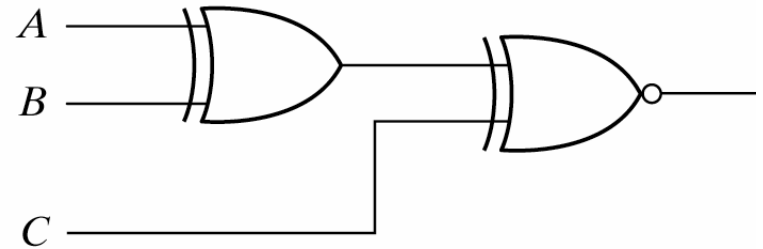
(a) Even function
 $F = (A \oplus B \oplus C)'$

Fig. 3-33 Map for a Three-variable Exclusive-OR Function

Ex-OR Kullanılarak Tek ve Çift Eşlik Yöntemi İle Veri Kodlanması



(a) 3-input odd function



(b) 3-input even function

Fig. 3-34 Logic Diagram of Odd and Even Functions

Ex-OR Kullanılarak Tek ve Çift Eşlik Yöntemi İle Veri Kodlanması

		C			
		CD		11	10
A	AB	00	01		
	00		1		1
	01	1		1	
	11		1		1
	10	1		1	

D

B

(a) Odd function

$$F = A \oplus B \oplus C \oplus D$$

		C			
		CD		11	10
A	AB	00	01		
	00	1		1	
	01		1		1
	11	1		1	
	10		1		1

D

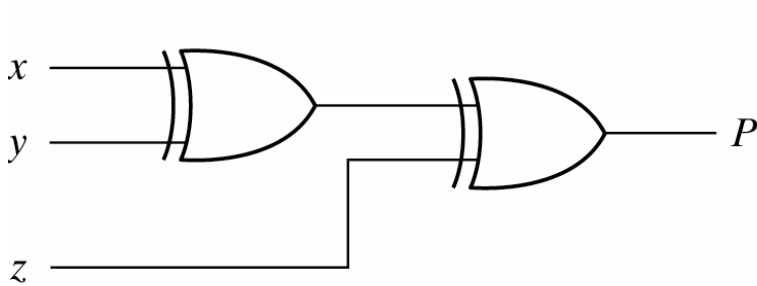
B

(b) Even function

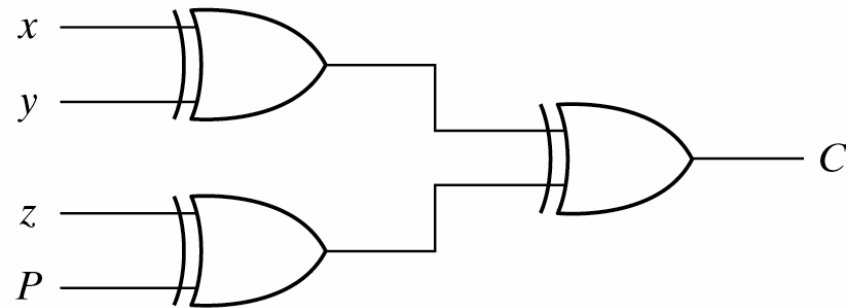
$$F = (A \oplus B \oplus C \oplus D)'$$

Fig. 3-35 Map for a Four-variable Exclusive-OR Function

Ex-OR Kullanılarak Tek ve Çift Eşlik Yöntemi İle Veri Kodlanması



(a) 3-bit even parity generator



(a) 4-bit even parity checker

Fig. 3-36 Logic Diagram of a Parity Generator and Checker