Sayısal Sistemler-H4CD2

Boolean Fonksiyonlarının Sadeleştirilmesi

Karnaugh Diyagramları

Dr. Meriç Çetin

versiyon011025

Bu derste öğreneceklerimiz

3 Gate-Level Minimization

- Kapı seviyesinde minimizasyon, bir dijital devreyi tanımlayan Boole fonksiyonlarının optimal bir kapı seviyesinde uygulamasını sağlayabilmek için kullanılan bir sadeleştirme şeklidir.
- Bu prosedürü birkaç girdiden daha fazlasına sahip durumlar için manuel yöntemlerle yürütmek zordur. Neyse ki, bilgisayar tabanlı mantık sentezi araçları, büyük bir Boole denklemleri kümesini verimli ve hızlı bir şekilde en aza indirebilir.
- Yine de, bir tasarımcının problemin altında yatan matematiksel tanımı ve çözümünü anlaması önemlidir.
- Bu bölüm, bu önemli konuyu anlamanız için bir temel teşkil eder ve sizi modern tasarım araçlarının becerikli kullanımına hazırlayarak basit devrelerin manuel tasarımını gerçekleştirmenizi sağlar.

• Harita Yöntemi

- Bir Boole işlevini uygulayan sayısal mantık kapılarının karmaşıklığı, doğrudan işlevin uygulandığı cebirsel ifadenin karmaşıklığı ile ilgilidir.
- Bir fonksiyonun doğruluk tablosu ile temsili benzersiz olsa da, cebirsel olarak ifade edildiğinde birçok farklı fakat eşdeğer formda görünebilir.
- Boole ifadeleri, 2. Bölümde tartışıldığı gibi cebirsel yollarla basitleştirilebilir.
- Bununla birlikte, bir sonraki adımı tahmin etmek için belirli kurallara sahip değildir.
- Burada sunulan harita yöntemi, Boole işlevlerini en aza indirgemek için basit, anlaşılır bir prosedür sağlar.
- Bu yöntem, bir doğruluk tablosunun resimli bir formu olarak kabul edilebilir.
- Harita yöntemi, **Karnaugh haritası** veya **K-haritası** olarak da bilinir.

• Karnaugh Haritası

- K-haritası, karelerden oluşan bir diyagramdır ve her kare, küçültülecek fonksiyonun bir **mintermini** temsil eder.
- Herhangi bir Boole fonksiyonu mintermlerin toplamı olarak ifade edilebildiğinden, mintermleri fonksiyona dahil edilen karelerin çevrelediği alanda bir sadeleştirme yapılır.
- Harita, bir fonksiyonun standart formda ifade edilebileceği tüm olası yolların görsel bir diyagramını sunar.
- Harita tarafından üretilen basitleştirilmiş ifadeler her zaman iki standart formdan biridir: çarpımların toplamı veya toplamların çarpımı.
- **En basit cebirsel ifadenin**, minimum sayıda terim ve her terimde mümkün olan en az sayıda bilgi içeren bir cebirsel ifade olduğu varsayılacaktır.
  - Bu ifade, minimum sayıda kapı ve her kapıya minimum sayıda giriş içeren bir devre şeması oluşturur.
- Daha sonra, en basit ifadenin benzersiz olmadığını göreceğiz: Bazen minimizasyon kriterlerini karşılayan iki veya daha fazla ifade bulmak mümkündür. Bu durumda her **iki çözüm de olasıdır.**

Minterm & Maxterm tablosunu hatırlayalım

Table 2.3
Minterms and Maxterms for Three Binary Variables

			Minterms		Maxterms	
x	**y**	**z**	**Term**	**Designation**	**Term**	**Designation**
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

2 Değişkenli K-haritası

- İki değişkenli harita Şekil 3.1 (a) 'da gösterilmektedir.
- **İki değişken için dört minterm** vardır; dolayısıyla harita, her minterm için bir tane olmak üzere dört kareden oluşur.
- Kareler ile x ve y değişkenleri arasındaki ilişkiyi göstermek için harita (b) 'de yeniden çizilmiştir.
- Her satırda ve sütunda işaretlenen 0 ve 1, değişkenlerin değerlerini belirtir.

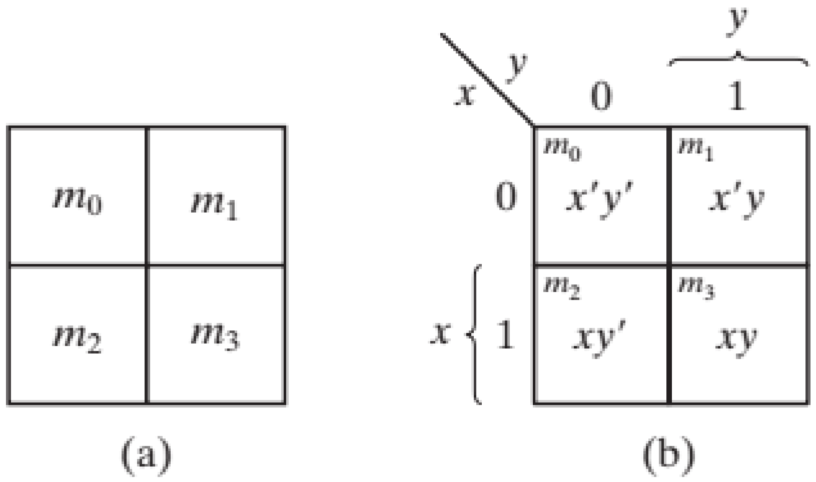
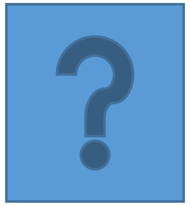


FIGURE 3.1
Two-variable K-map

2 Değişkenli K-haritası vs Boole cebri

- Aşağıdaki fonksiyonu Boole cebri kurallarına göre indirgeyiniz

$$m_1 + m_2 + m_3 = x'y + xy' + xy$$



2 Değişkenli K-haritası vs Boole cebri

- Aşağıdaki fonksiyonu Boole cebri kurallarına göre indirgeyiniz

$$m_1 + m_2 + m_3 = x'y + xy' + xy$$

$$y\underbrace{(x+x')}_{1}+xy' = y+xy' = (y+x)\underbrace{(y+y')}_{1} = x+y$$

2 Değişkenli K-haritası

$$m_1 + m_2 + m_3 = x'y + xy' + xy = x + y$$

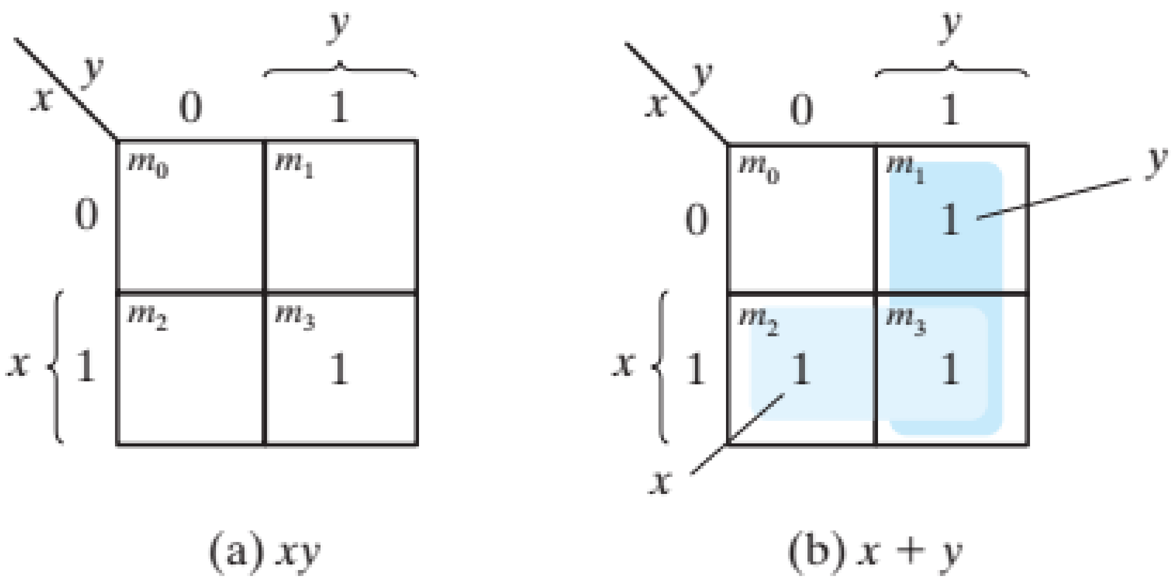
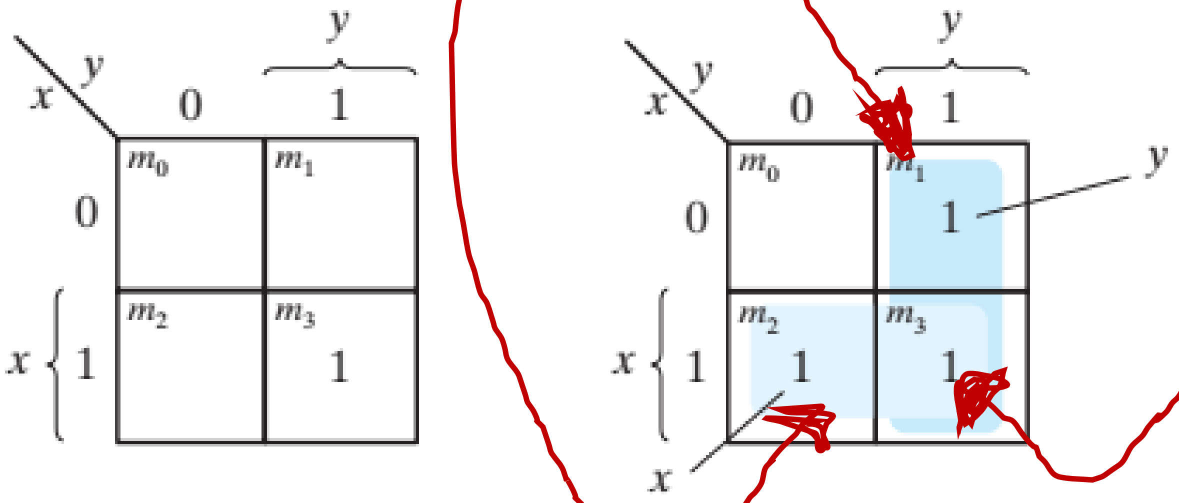


FIGURE 3.2
Representation of functions in the map

2 Değişkenli K-haritası

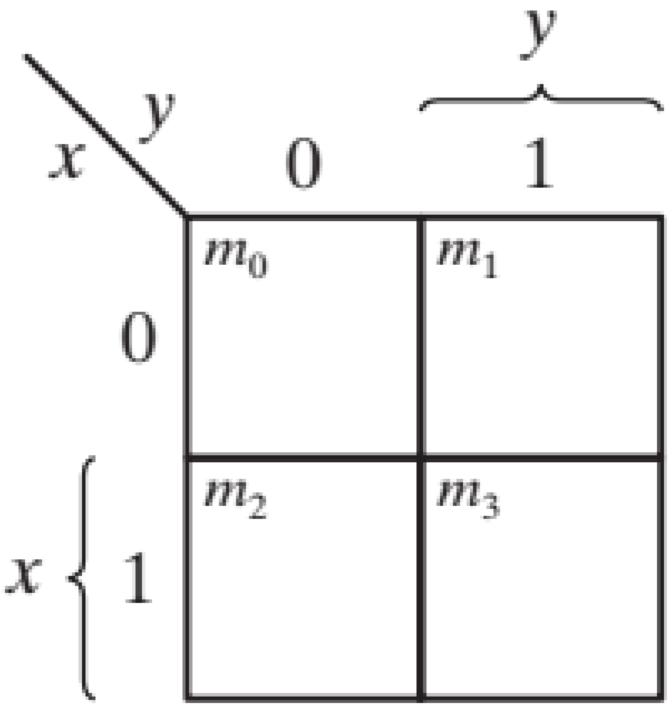
$$m_1 + m_2 + m_3 = x'y + xy' + xy = x + y$$

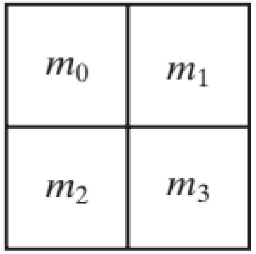


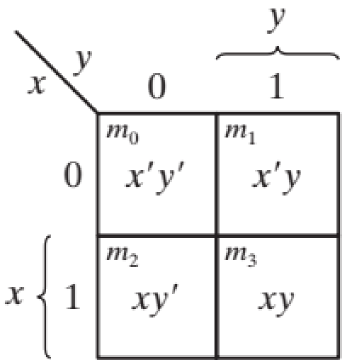
(a) xy (b) $x + y$

FIGURE 3.2
Representation of functions in the map

Örnek1

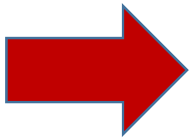


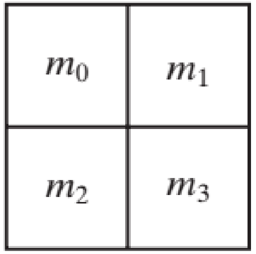


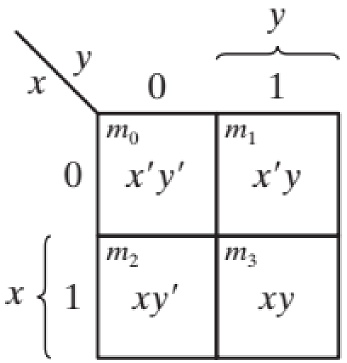


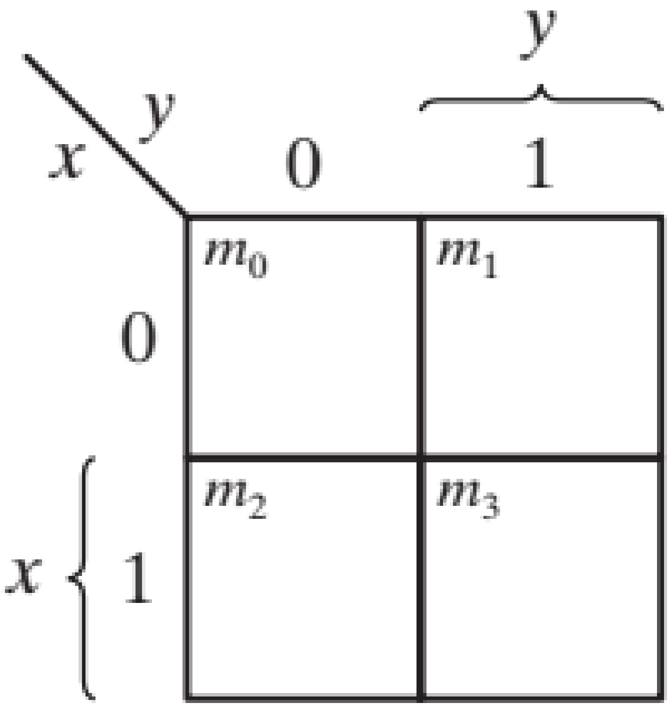
X	Y	f
0	0	0
0	1	1
1	0	0
1	1	1

Örnek2



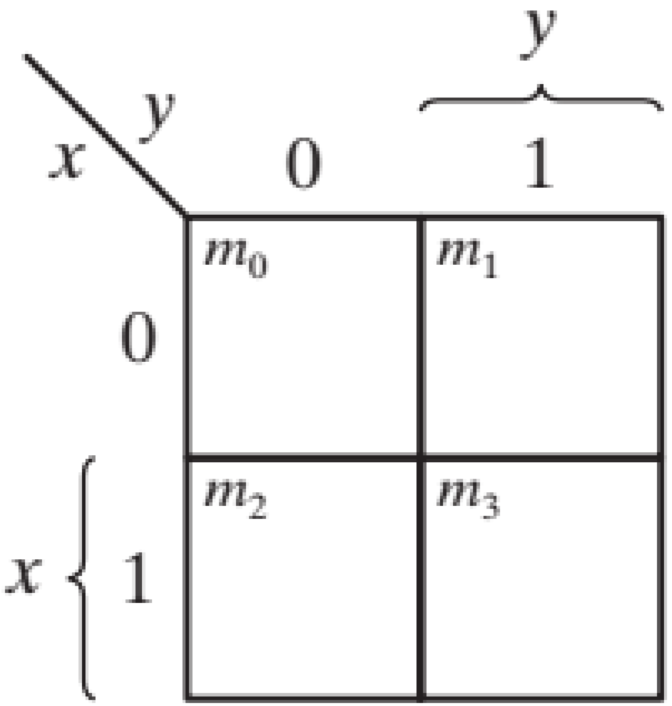


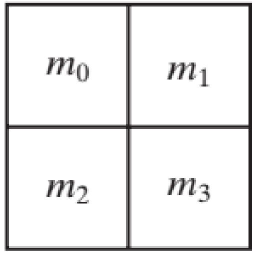


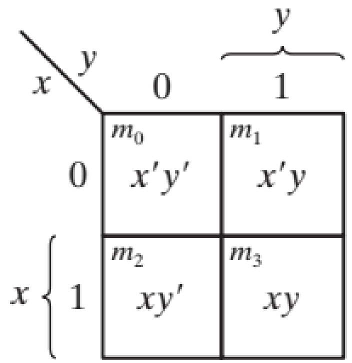


X	Y	f
0	0	1
0	1	0
1	0	0
1	1	1

Örnek3

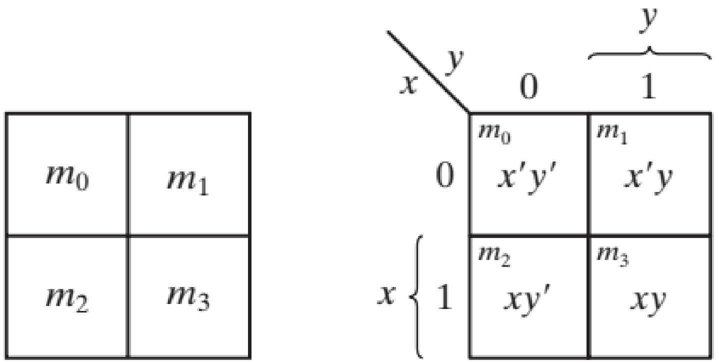


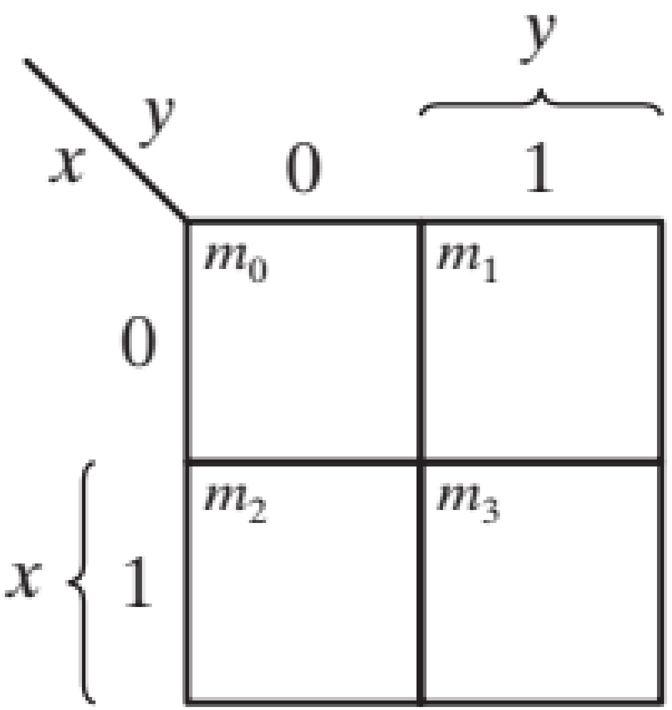




X	Y	f
0	0	1
0	1	1
1	0	0
1	1	1

Örnek4





X	Y	f
0	0	1
0	1	1
1	0	1
1	1	1

3 Değişkenli K-haritası

- Üç değişkenli bir K-haritası Şekil 3.3'te gösterilmektedir.
- **Üç ikili değişken için sekiz minterm** vardır;
- Bu nedenle harita sekiz kareden oluşur.
- Mintermlerin ikili bir sırayla değil, Gray koduna benzer bir sırayla düzenlendiğini unutmayın.
- Boole fonksiyonlarını basitleştirmede haritanın kullanışlılığını anlamak için, bitişik karelerin sahip olduğu temel özelliği tanımalıyız

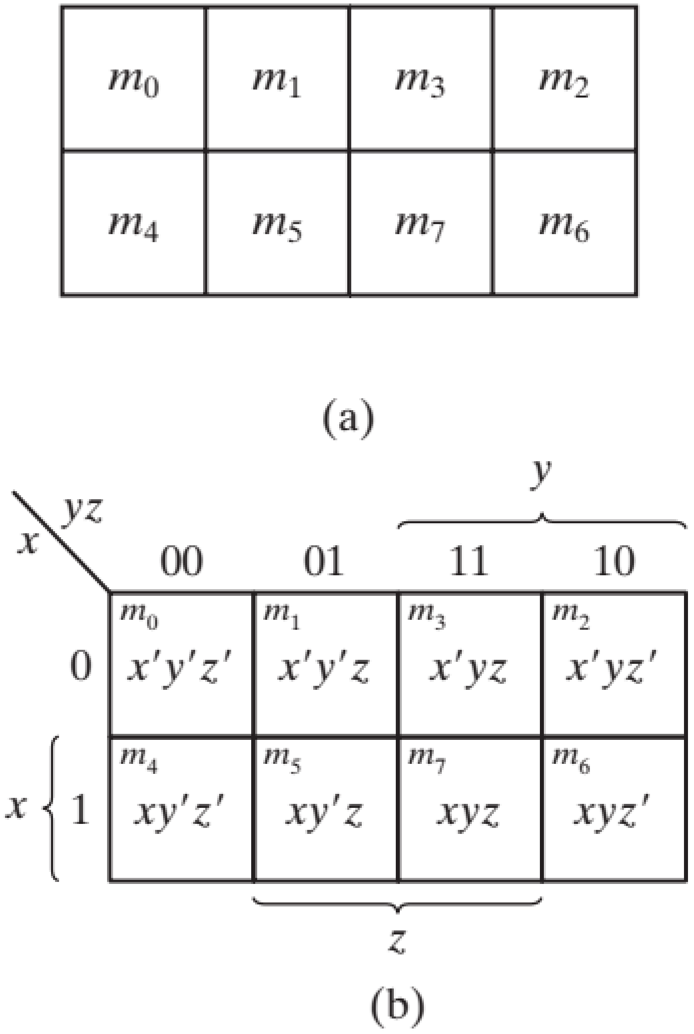


FIGURE 3.3
Three-variable K-map

3 Değişkenli K-haritası vs Boole cebri

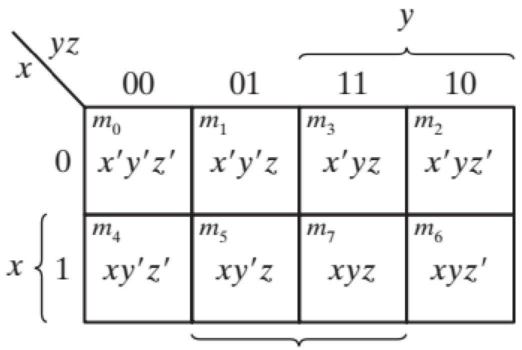
- Aşağıdaki fonksiyonu Boole cebri kurallarına göre indirgeyiniz

EXAMPLE 3.1

Simplify the Boolean function

$$F(x, y, z) = \Sigma(2, 3, 4, 5)$$

Örnek1



(b)

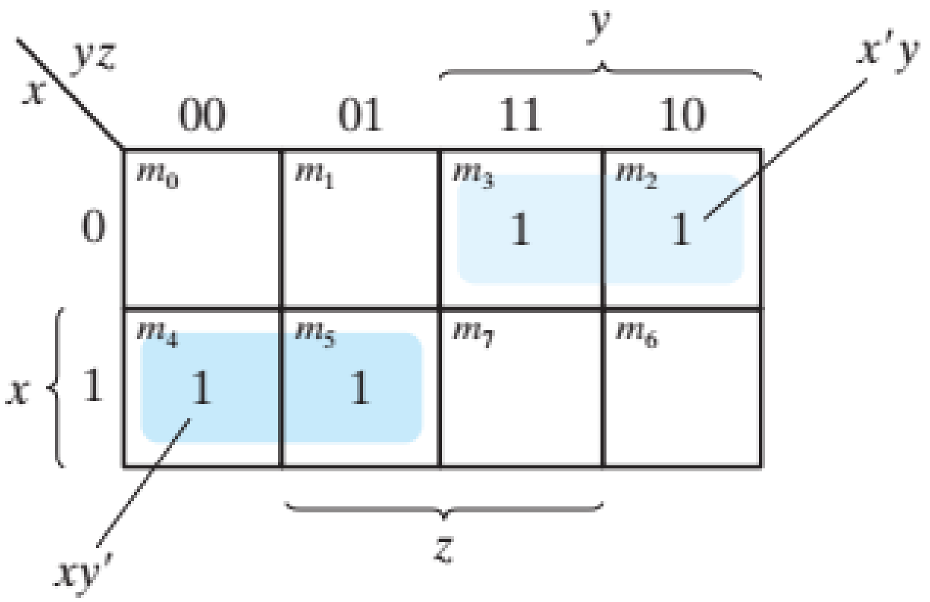
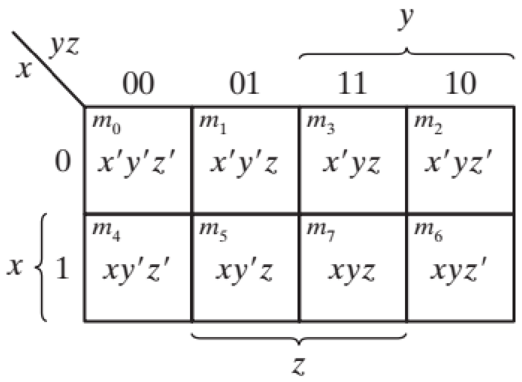


FIGURE 3.4

Map for Example 3.1, $F(x, y, z) = \Sigma(2, 3, 4, 5) = x'y + xy'$

"**Digital Design**",
M. Morris Mano

Örnek2

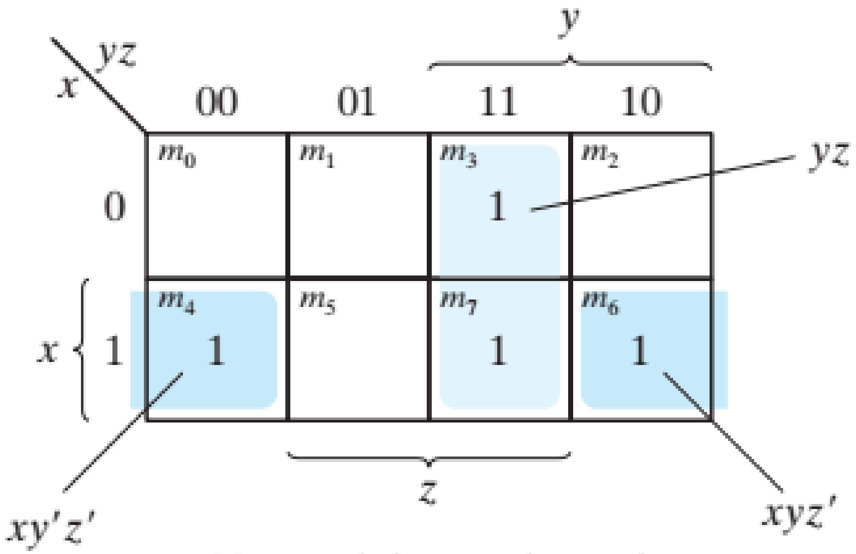


(b)

EXAMPLE 3.2

Simplify the Boolean function

$$F(x, y, z) = \Sigma(3, 4, 6, 7)$$

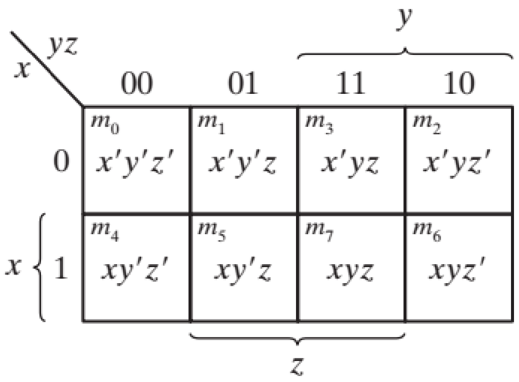


Note: $xy'z' + xyz' = xz'$

FIGURE 3.5
Map for Example 3.2, $F(x, y, z) = \Sigma(3, 4, 6, 7) = yz + xz'$

"**Digital Design**",
M. Morris Mano

Örnek3

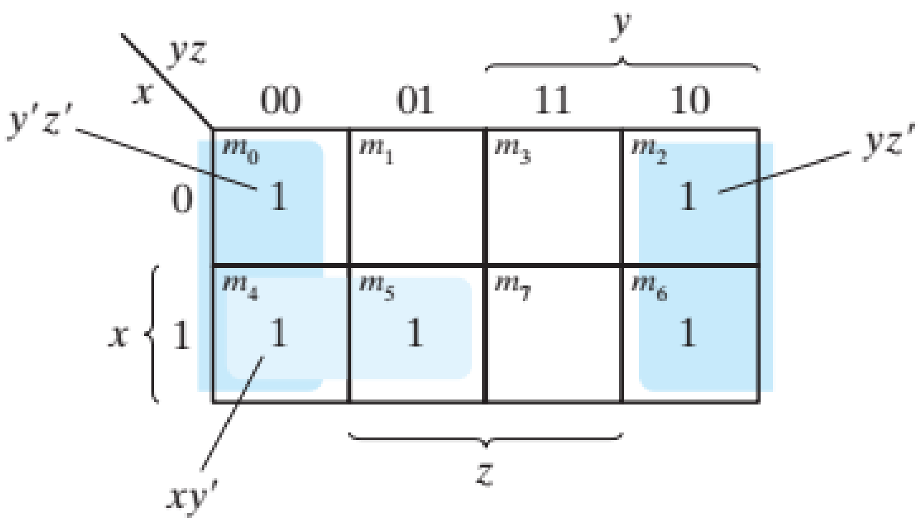


(b)

EXAMPLE 3.3

Simplify the Boolean function

$$F(x, y, z) = \Sigma(0, 2, 4, 5, 6)$$

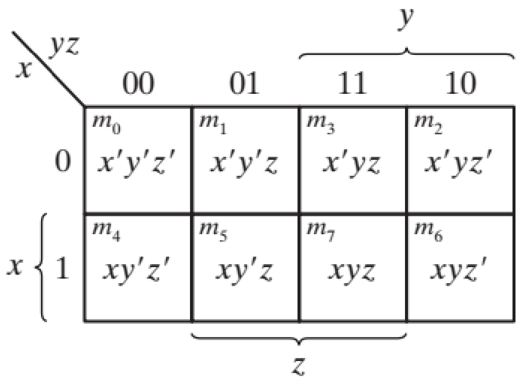


Note: $y'z' + yz' = z'$

FIGURE 3.6
Map for Example 3.3, $F(x, y, z) = \Sigma(0, 2, 4, 5, 6) = z' + xy'$

"**Digital Design**",
M. Morris Mano

Örnek4



(b)

EXAMPLE 3.4

For the Boolean function

$$F = A'C + A'B + AB'C + BC$$

$$F(A, B, C) = \Sigma(1, 2, 3, 5, 7)$$

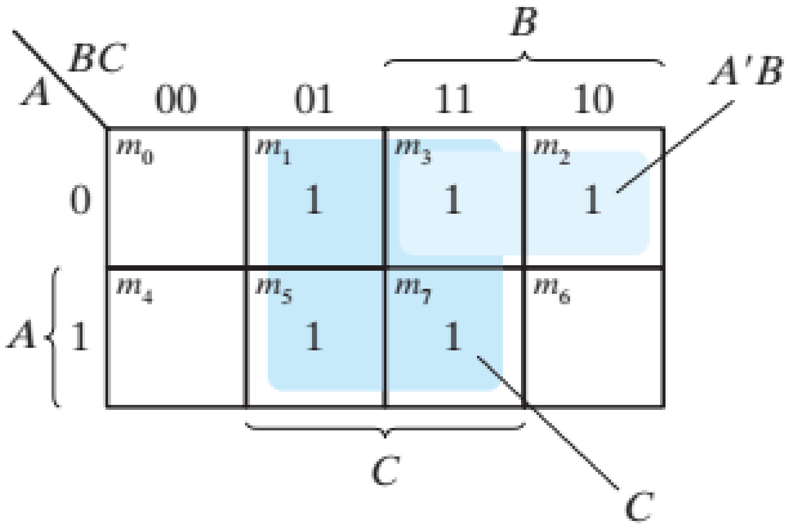


FIGURE 3.7
Map of Example 3.4, $A'C + A'B + AB'C + BC = C + A'B$

"**Digital Design**",
M. Morris Mano

4 Değişkenli K-haritası

- Dört ikili değişkenin (w, x, y, z) Boole fonksiyonlarının haritası Şekil 3.8'de gösterilmektedir.
- Şekil 3.8 (a) 'da 16 minterm ve her birine atanmış kareler listelenmiştir.
- Şekil 3.8 (b) 'de, kareler ve dört değişken arasındaki ilişkiyi göstermek için harita yeniden çizilmiştir.
- Sıralar ve sütunlar, iki bitişik satır veya sütun arasında yalnızca bir rakamın değiştiği bir Gri kod dizisi ile numaralandırılır.
- Her kareye karşılık gelen minterm, satır numarasının sütun numarası ile birleştirilmesinden elde edilebilir.

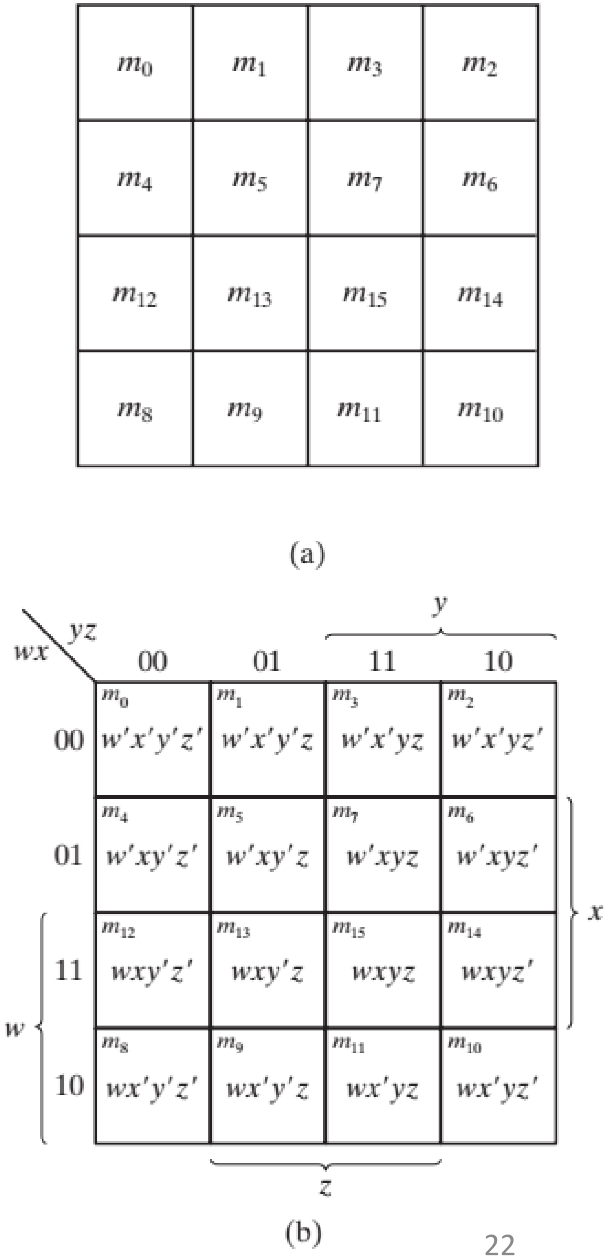


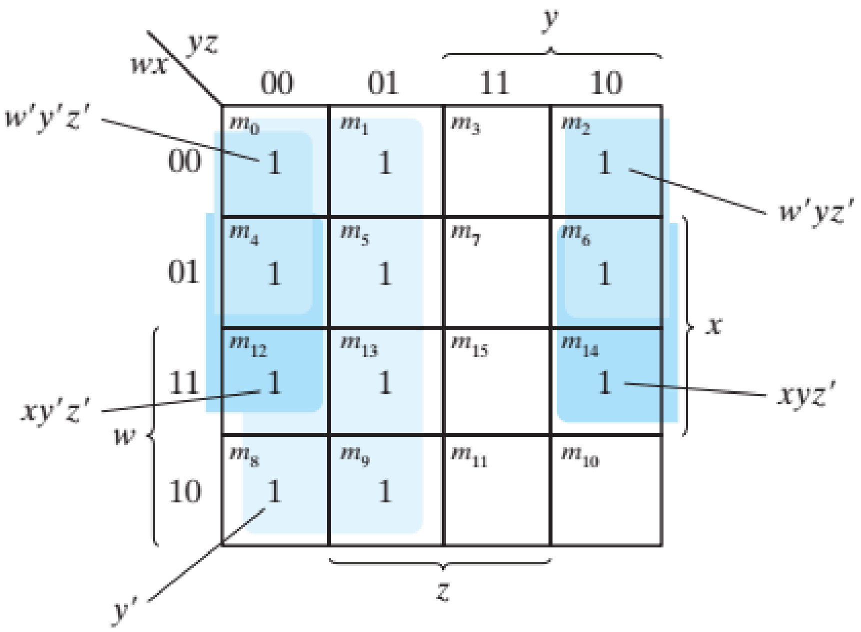
FIGURE 3.8
Four-variable map

Örnek1

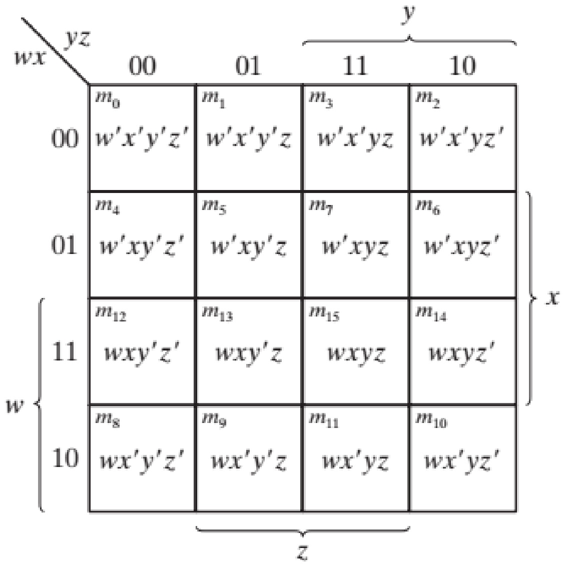
EXAMPLE 3.5

Simplify the Boolean function

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



Note: $w'y'z' + w'yz' = w'z'$
$xy'z' + xyz' = xz'$



“**Digital Design**”,
M. Morris Mano

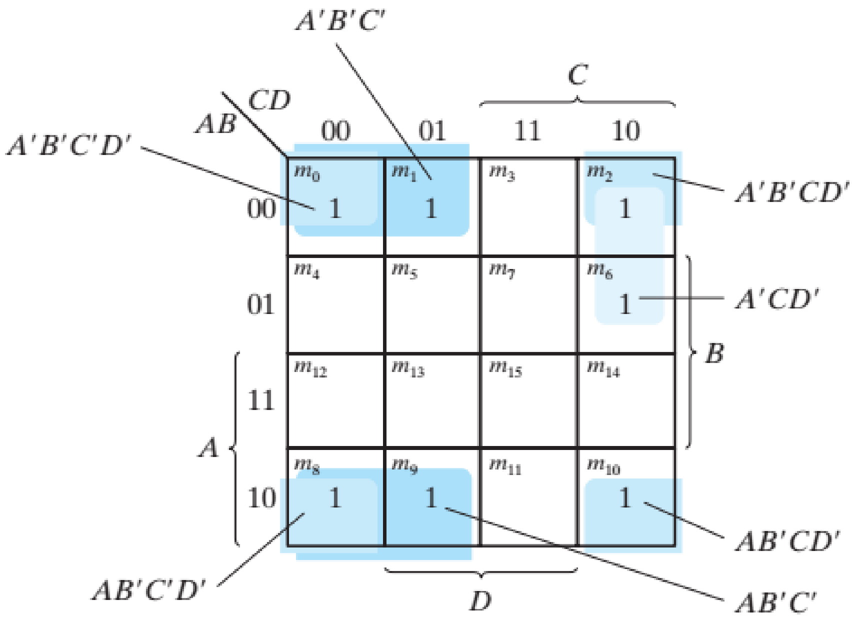
FIGURE 3.9
Map for Example 3.5, $F(w, x, y, z) = \Sigma(0,1, 2, 4, 5, 6, 8, 9, 12, 13, 14) = y' + w'z' + xz'$

Örnek2

EXAMPLE 3.6

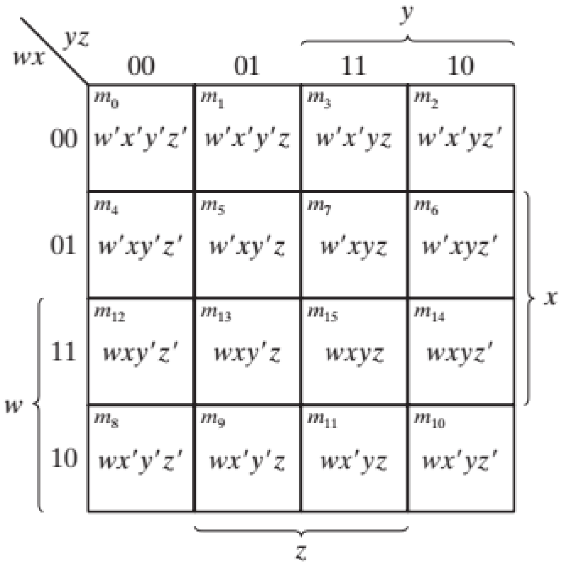
Simplify the Boolean function

$$F = A'B'C' + B'CD' + A'BCD' + AB'C'$$



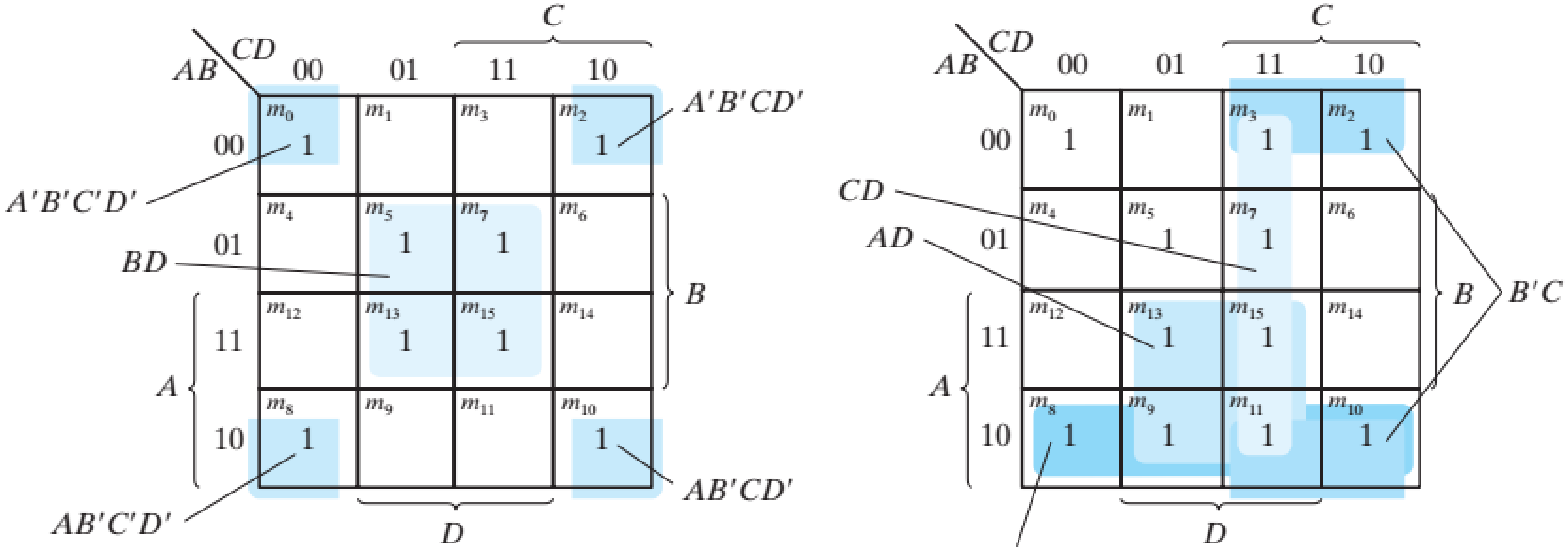
Note: $A'B'C'D' + A'B'CD' = A'B'D'$
$AB'C'D' + AB'CD' = AB'D'$
$A'B'D' + AB'D' = B'D'$
$A'B'C' + AB'C' = B'C'$

FIGURE 3.10
Map for Example 3.6, $A'B'C' + B'CD' + A'BCD' + AB'C' = B'D' + B'C' + A'CD'$



"**Digital Design**",
M. Morris Mano

Örnek3

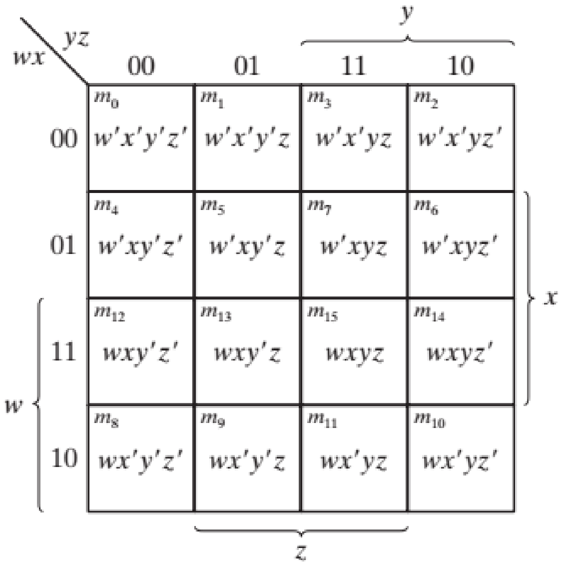


Note: $A'B'C'D' + A'B'CD' = A'B'D'$
$AB'C'D' + AB'CD' = AB'D'$
$A'B'D' + AB'D' = B'D'$

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

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

FIGURE 3.11
Simplification using prime implicants



"**Digital Design**",
M. Morris Mano

Belirsiz durumlar

3.5 DON'T-CARE CONDITIONS

- Bir Boole fonksiyonuyla ilişkili mintermlerin mantıksal toplamı, fonksiyonun 1'e eşit olduğu koşulları belirtir. Fonksiyon, mintermlerin geri kalanı için 0'a eşittir.
- Bu koşul çifti, fonksiyonun değişkenleri için değerlerin tüm kombinasyonlarının geçerli olduğunu varsayar.
- Pratikte, bazı uygulamalarda fonksiyon, değişkenlerin belirli kombinasyonları için belirtilmemiştir.
- **Bazı girdi kombinasyonları için belirtilmemiş çıktılara sahip olan fonksiyonlar**, eksik belirtilmiş fonksiyonlar olarak adlandırılır.
- Çoğu uygulamada, **belirtilmemiş mintermler için fonksiyonun hangi değeri üstlendiğini umursanmaz.**
- Bu nedenle, bir fonksiyonun belirtilmemiş mintermlerini önemsememe koşullarını çağırmak gelenekseldir.
- Bu önemsiz koşullar, Boole ifadesinin daha da basitleştirilmesini sağlamak için bir harita üzerinde kullanılabilir.

Belirsiz durumlar

3.5 DON'T-CARE CONDITIONS

- Önemsiz bir minterm, mantıksal değeri belirtilmeyen değişkenlerin bir kombinasyonudur.
- Böyle bir minterm haritada 1 ile işaretlenemez, çünkü böyle bir kombinasyon fonksiyonun her zaman 1 olmasını gerektirir.
- Benzer şekilde, kareye 0 koymak fonksiyonun 0 olmasını gerektirir.
- Önemseme koşulunu **1'ler ve 0'lardan ayırmak için X kullanılır.**
- Bu nedenle, haritadaki bir karenin içindeki bir X, belirli bir minterm için 0 veya 1 değerinin F'ye atanıp atanmadığını umursamadığımızı gösterir.
- Bir haritadaki işlevi basitleştirmek için bitişik kareleri seçerken, önemsiz mintermlerin 0 veya 1 olduğu varsayılabilir.
- **İşlevi basitleştirirken, her önemsiz mintermi en basit ifadeyi verecek şekilde 1'ler veya 0'lar ile dahil etmeyi seçebiliriz.**

Örnek

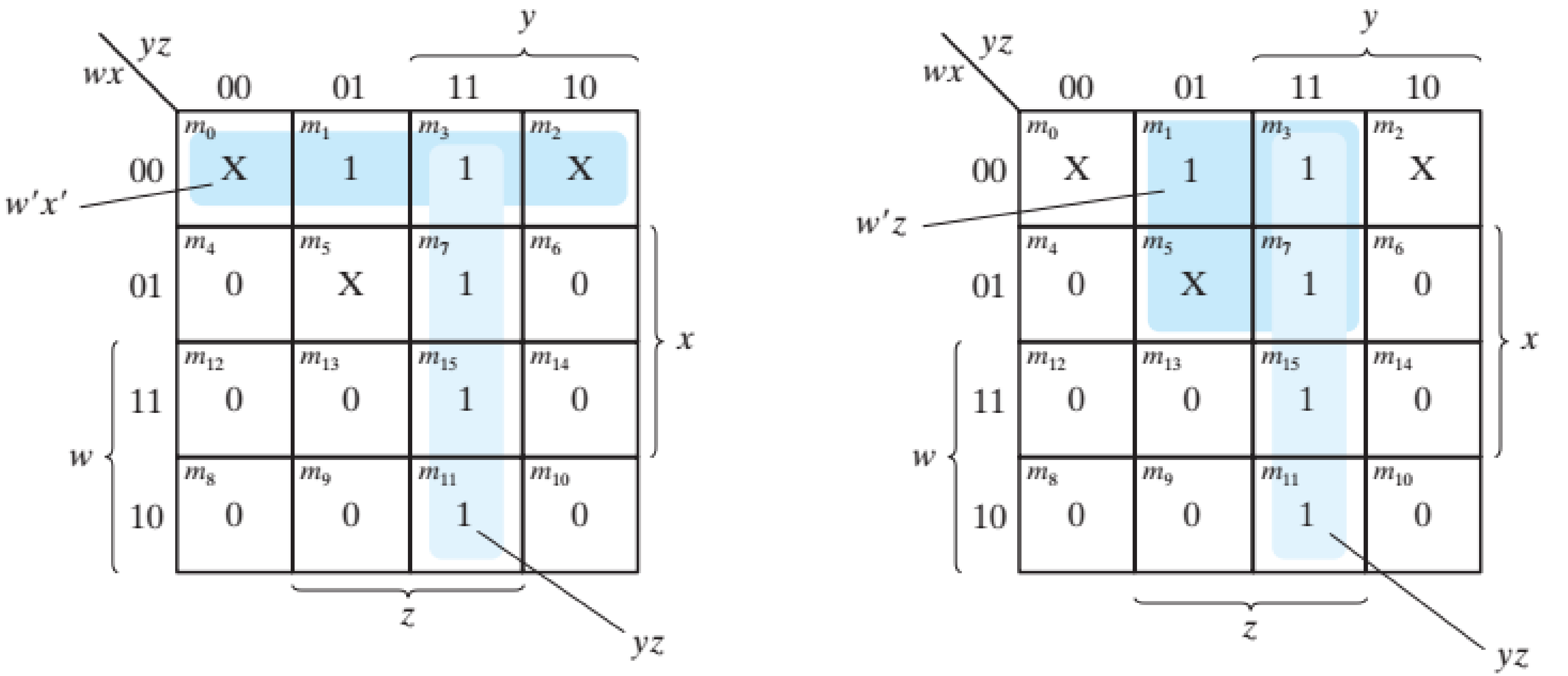
EXAMPLE 3.8

Simplify the Boolean function

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

which has the don't-care conditions

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



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

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

"**Digital Design**",
M. Morris Mano

FIGURE 3.15
Example with don't-care conditions

Nand ve Nor Uygulaması

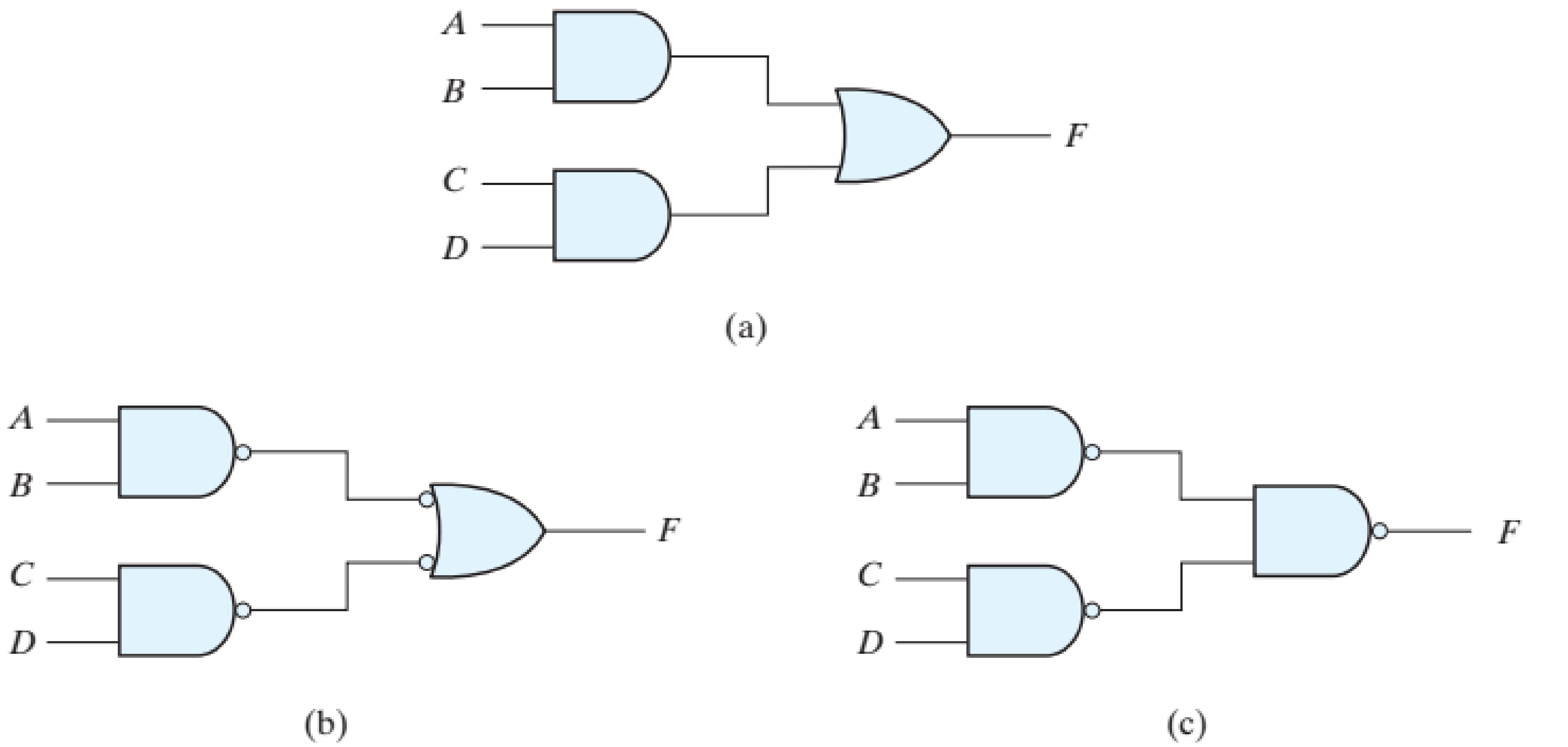


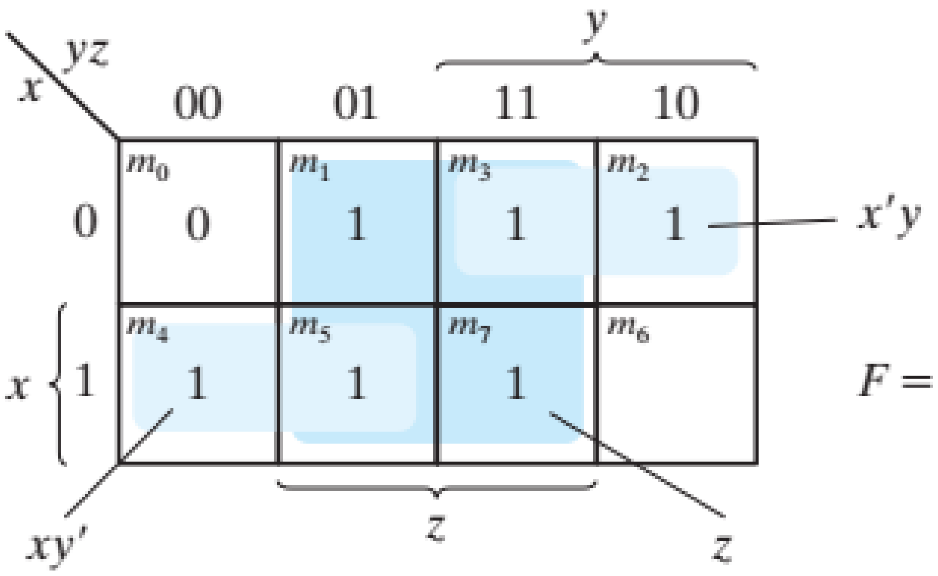
FIGURE 3.18
Three ways to implement $F = AB + CD$

Örnek

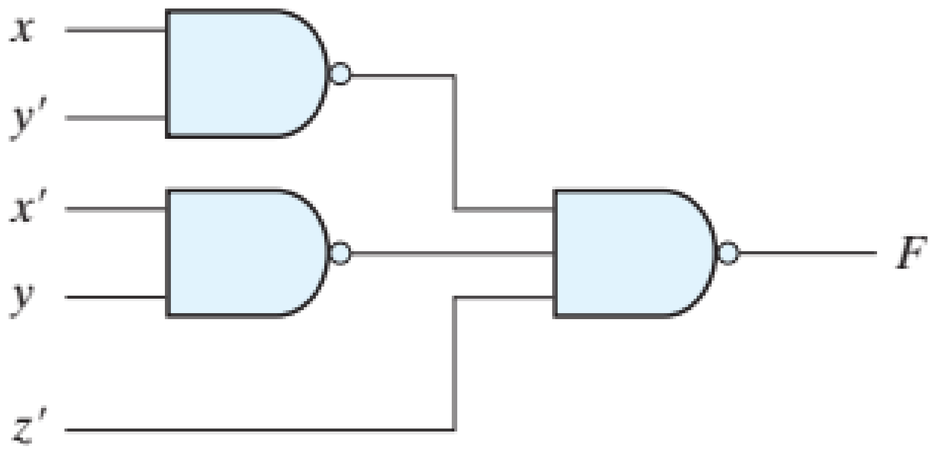
EXAMPLE 3.9

Implement the following Boolean function with NAND gates:

$$F(x, y, z) = (1, 2, 3, 4, 5, 7)$$



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



“**Digital Design**”,
M. Morris Mano

3.9 HARDWARE DESCRIPTION LANGUAGE

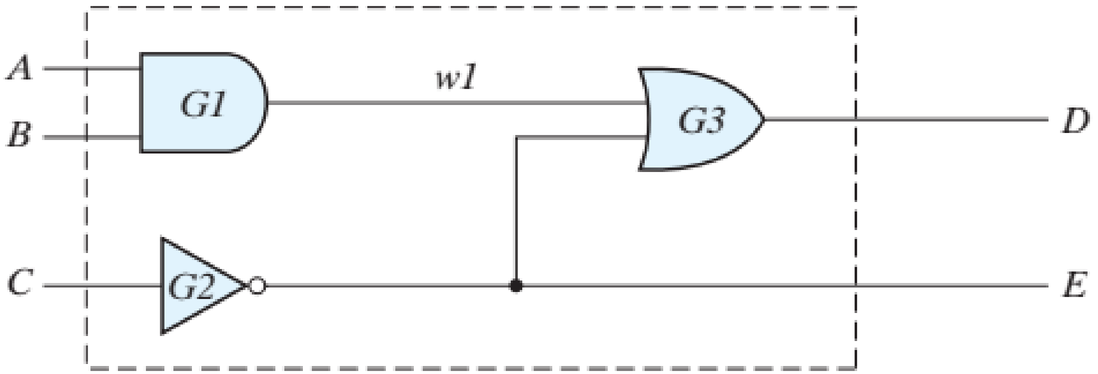


FIGURE 3.35
Circuit to demonstrate an HDL

HDL Example 3.1 (Combinational Logic Modeled with Primitives)

```
// Verilog model of circuit of Figure 3.35. IEEE 1364–1995 Syntax

module  Simple_Circuit (A, B, C, D, E);
  output        D, E;
  input         A, B, C;
  wire          w1;

  and           G1 (w1, A, B); // Optional gate instance name
  not           G2 (E, C);
  or            G3 (D, w1, E);
endmodule
```

WEB SEARCH TOPICS

Boolean minimization
Karnaugh map
Wired logic
Emitter-coupled logic
Open-collector logic
Quine McCluskey method
Expresso software
Consensus theorem
Don't-care conditions