

Mantıksal Tasarım ve Uygulamaları

Dr. Burcu KIR SAVAŞ



Karnaugh Haritaları

Amaç

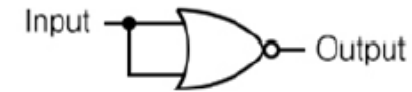
- Lojik eşitliklerin sadeleştirilmesinde kullanılan 'Karnaugh Haritası' yönteminin tanıtılması
- İki-üç-dört değişkenli 'Karnaugh Haritalarının' hücrelerin anlamlarının açıklanması
- Doğruluk tablolarından elde edilen 'minterim' veya 'maksterim' terimlerinin 'Karnaugh Haritalarına' taşınması
- Fark etmeyen durumların 'Karnaugh Haritalarında' ifade şeklinin gösterilmesi
- Lojik devrelerin tasarımında 'Karnaugh Haritalarının' kullanımının öğretilmesi

Karnaugh Haritaları

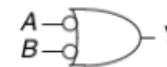
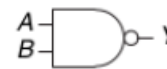
- Devre tasarımında lojik eşitlikleri oluşturmak veya oluşturulan lojik eşitlikleri grafiksel olarak sadeleştirmek için yaygın olarak kullanılan yöntemlerden birisi '**Karnaugh Haritası**' (Karnaugh Maps) yöntemidir.
- . 'Karnaugh haritası' (Karno çizelgesi), sadeleştirilecek eşitliğin bütün değerlerini sıralamak için kullanılan, eşitliğin alabileceği en basit (sade) şekli içeren, hücrelerin oluşturduğu bir yöntemdir.
- Giriş değişkenlerinin sayısı artıkça ifadelerin sadeleştirilmesinin zorlaştığı bu yöntem, giriş değişkenleri sayısının 6'ya kadar olduğu durumlarda iyi bir sonuç verir.
- Genelde kullanılan; 2, 3 ve 4 giriş değişkenli Karnaugh haritalarıdır (çizelgeleridir).

Karnaugh Haritaları

- Nand ve Nor uygulamaları: Dijital devre tasarımlarında And ve Or kapıları yerine daha çok NAND ve NOR kapıları kullanılır. Çünkü bu kapıların üretimi hem daha kolay hem de kapıların geçitleri dijital devre tasarımında büyük öneme sahiptir.



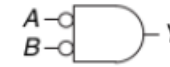
NAND



$$Y = \overline{AB} = \overline{A} + \overline{B}$$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NOR



$$Y = \overline{A+B} = \overline{A} \overline{B}$$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Karnaugh Haritaları

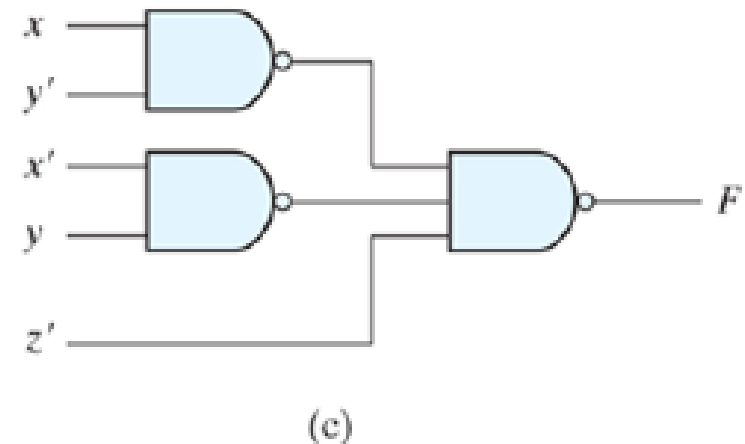
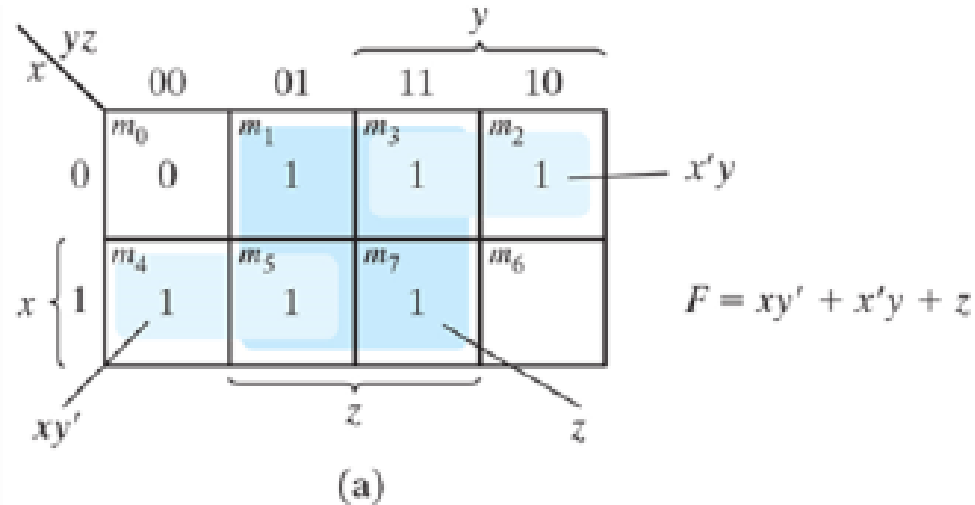
- Boole fonksiyonlarının NAND kapıları ile gerçekleştirilmesi için çarpım
- Boole fonksiyonlarının NOR kapıları ile gerçekleştirilmesi için toplama işlemleri ile ifade edilmelidir.
- Çarpımların toplamı şeklindeki fonksiyonlar iki kere değil işlemine girerse çarpımlar şekline gelir. Demorgan Kuralı: $(x+y)' = x'.y'$ Bu şekilde de NAND Kapıları ile gerçekleştirilebilir.
- Toplamların çarpımı şeklindeki fonksiyonlar iki kere değil işlemine girerse toplamlar şekline gelir. Demorgan Kuralı: $(xy)' = x'+y'$ Bu şekilde de NOR Kapıları ile gerçekleştirilebilir.

Karnaugh Haritaları

- $F(x,y,z)=(1,2,3,4,5,7)$ Boolean fonksiyonunu NAND kapıları kullanarak tasarlayınız.

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

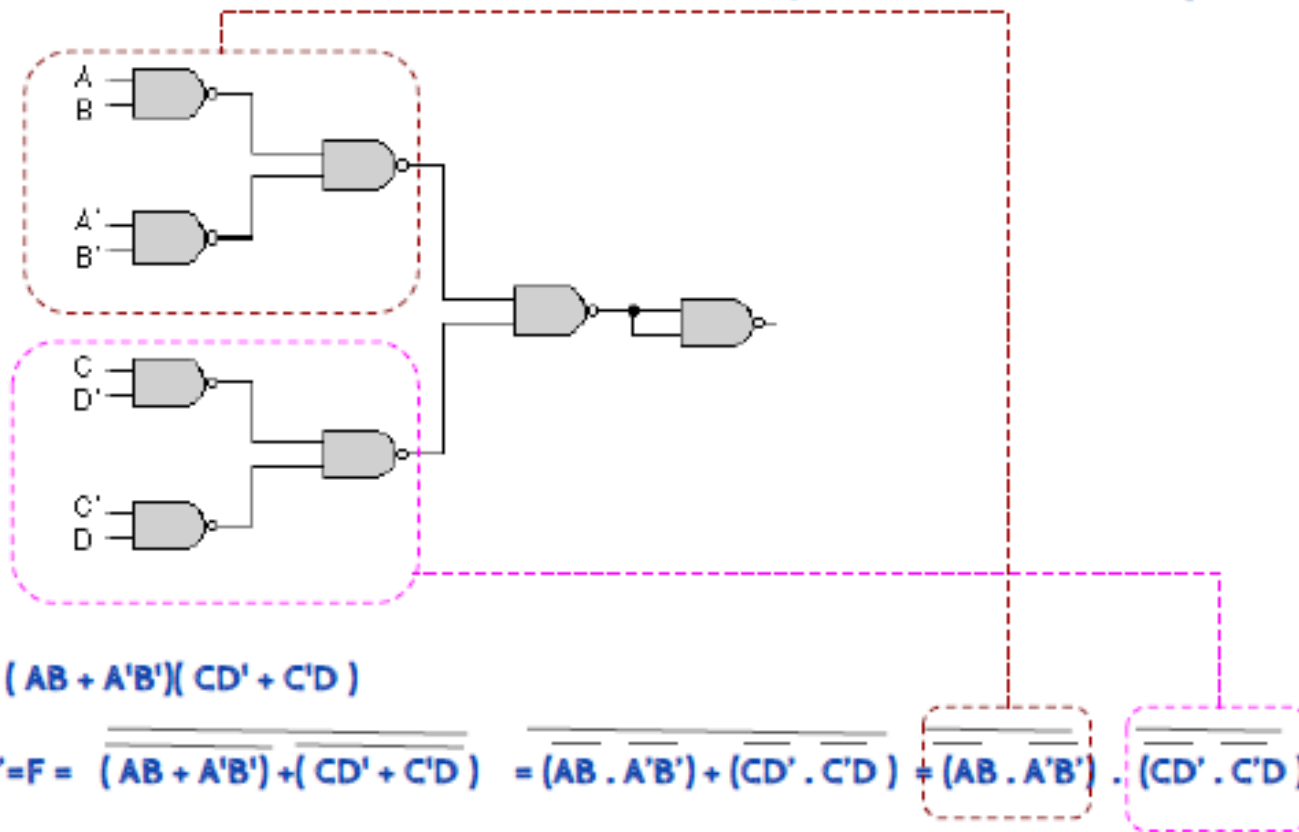
$$(F)' = F = (xy' + x'y + z)' = (xy')'(x'y)'z'$$



Demorgan kuralı $(x+y)' = x' \cdot y'$

Karnaugh Haritaları

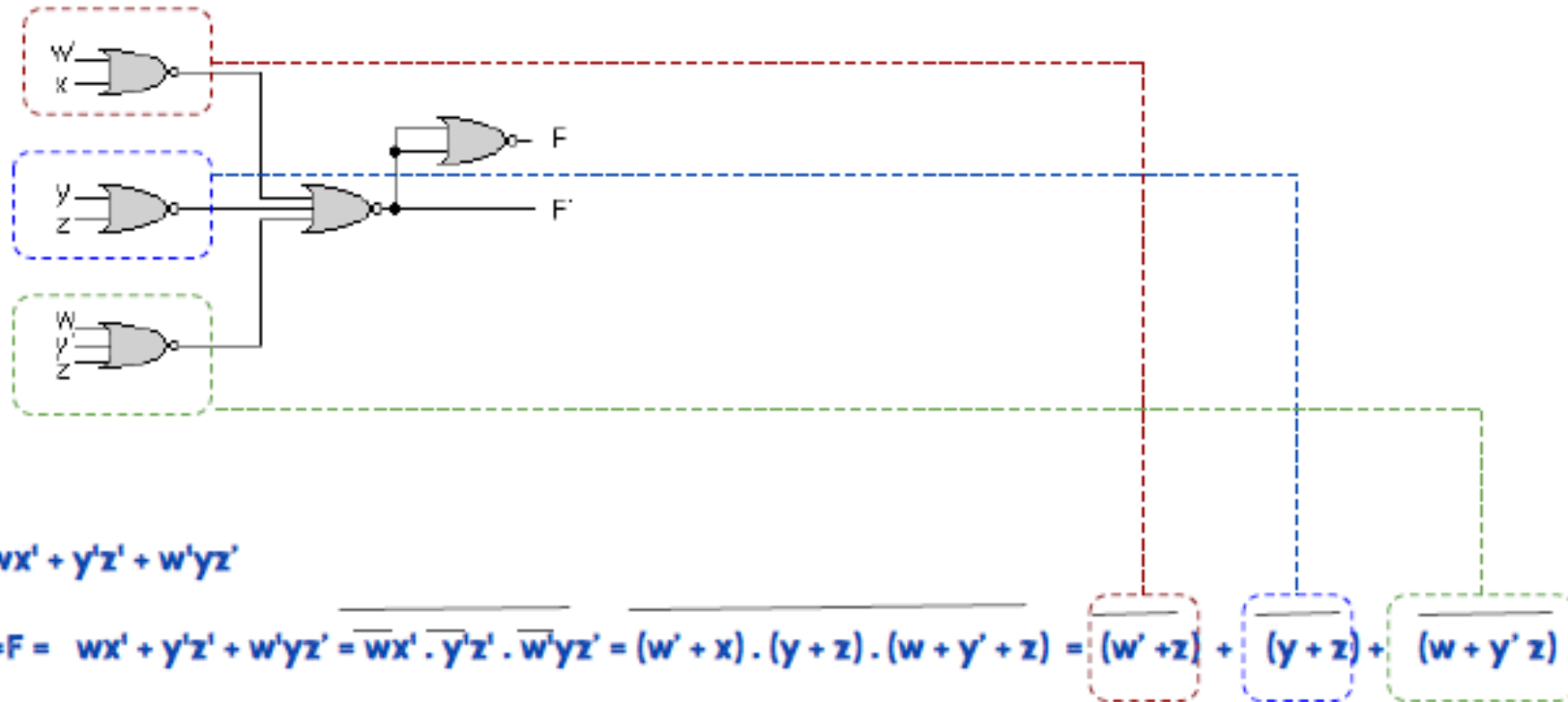
- $F = (AB + A'B')(CD' + C'D)$ ifadesini NAND kapıları kullanarak tasarlayınız



Not: Demorgan kuralı: $(x+y)' = x' \cdot y'$ && $(x \cdot y)' = x' + y'$

Karnaugh Haritaları

- $F(x,y,z,w)=wx'+y'z'+w'yz'$ Boolean fonksiyonunu NOR kapıları kullanarak tasarlayınız.

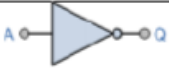



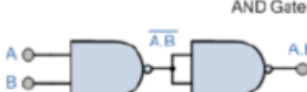
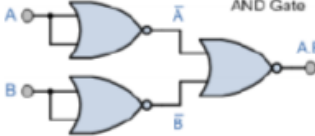

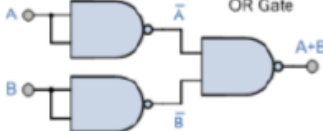
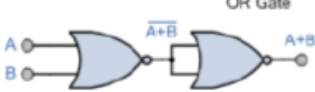


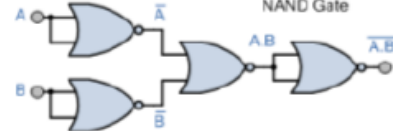
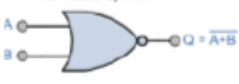
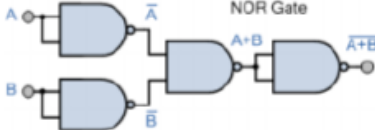
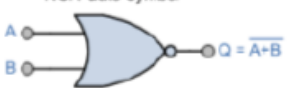

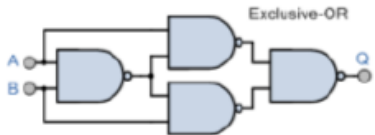
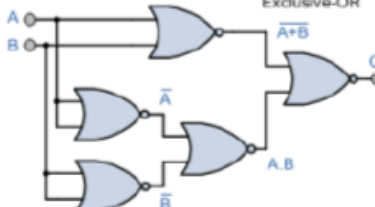

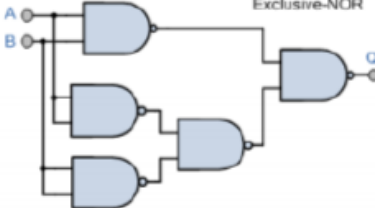
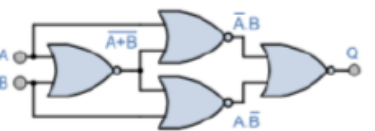


$$F = wx' + y'z' + w'yz'$$

$$(F')' = F = \overline{wx' + y'z' + w'yz'} = \overline{wx'} \cdot \overline{y'z'} \cdot \overline{w'yz'} = (w' + x) \cdot (y + z) \cdot (w + y' + z) = \overline{(w' + x)} + \overline{(y + z)} + \overline{(w + y' + z)}$$

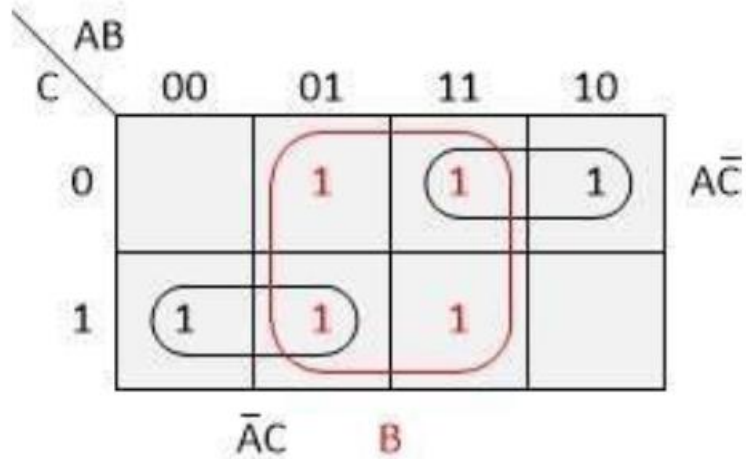
Not: Demorgan kuralı: $(x+y)' = x' \cdot y'$ && $(x \cdot y)' = x' + y'$

Karnaugh Haritaları

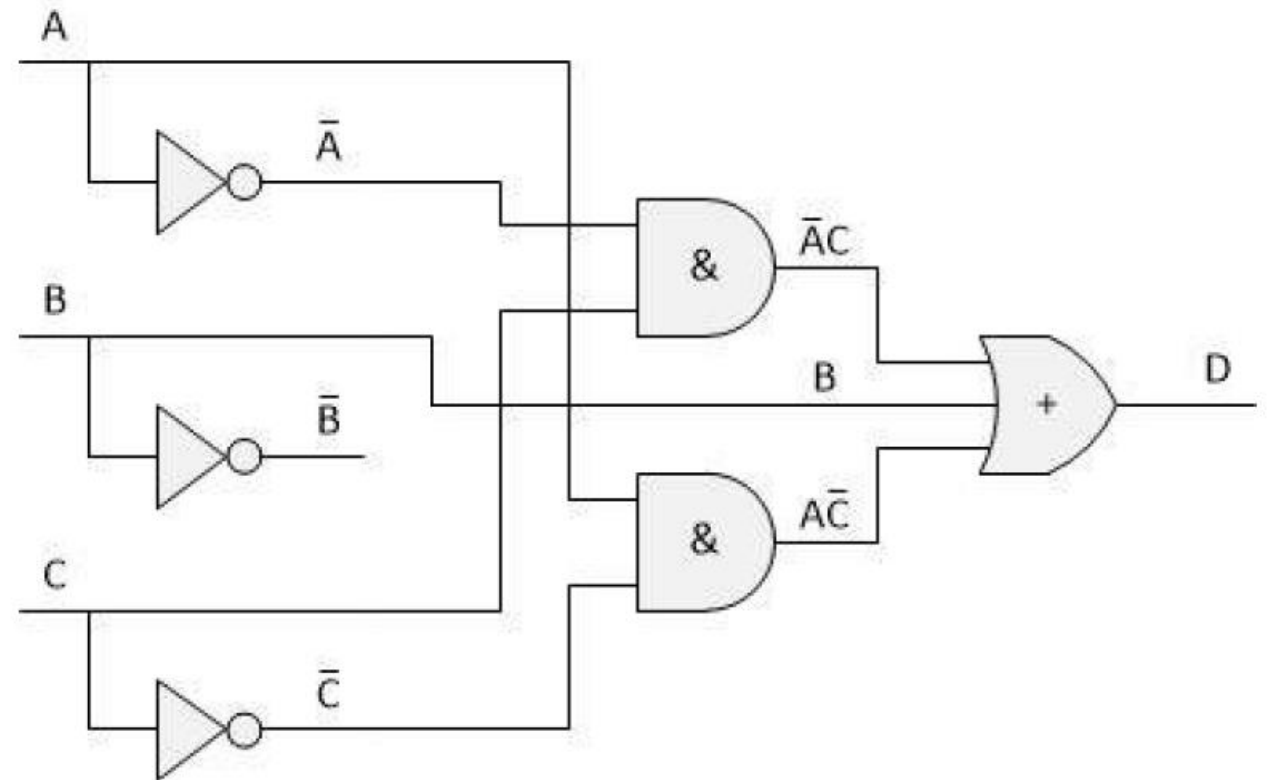
KAPI	SEMBOL	NAND Gösterimi	NOR Gösterimi
DEĞİL			
VE			
VEYA			
VE DEĞİL			
VEYA DEĞİL			
EX-OR			
EX-NOR			

Karnaugh Haritaları

- $D = A'BC + ABC' + AB'C' + A'B'C + A'BC + ABC$ ifadesini NAND kapıları kullanarak tasarlayınız.

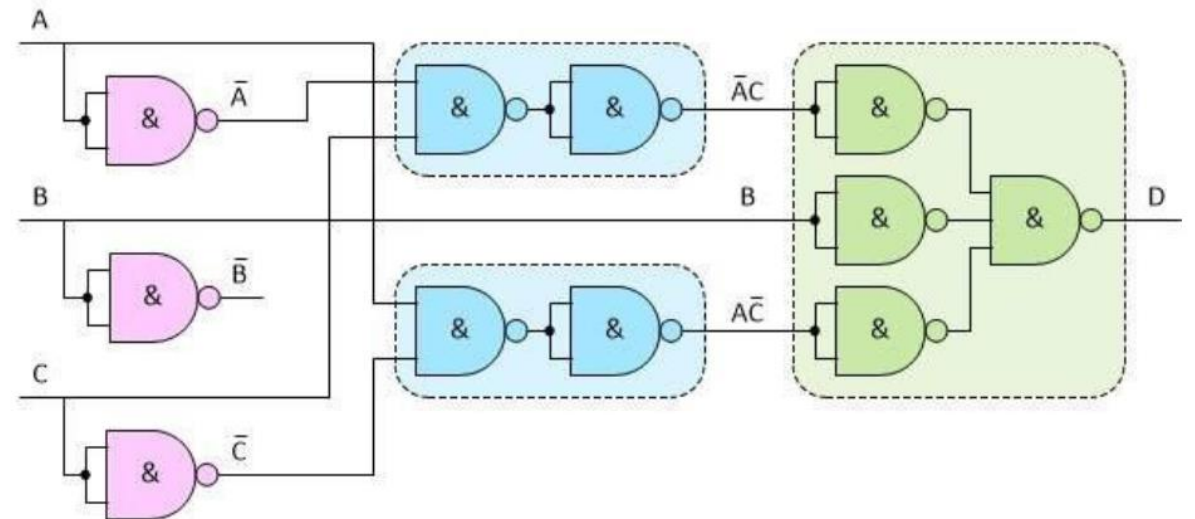
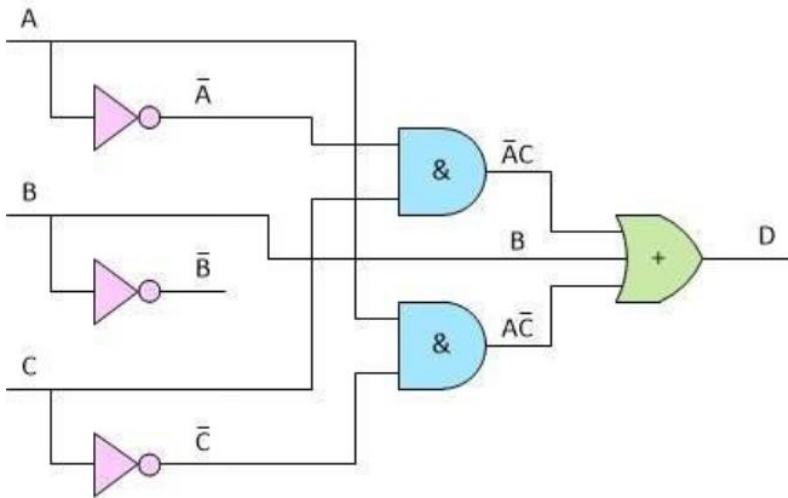


- $D = A'C + AC' + B$



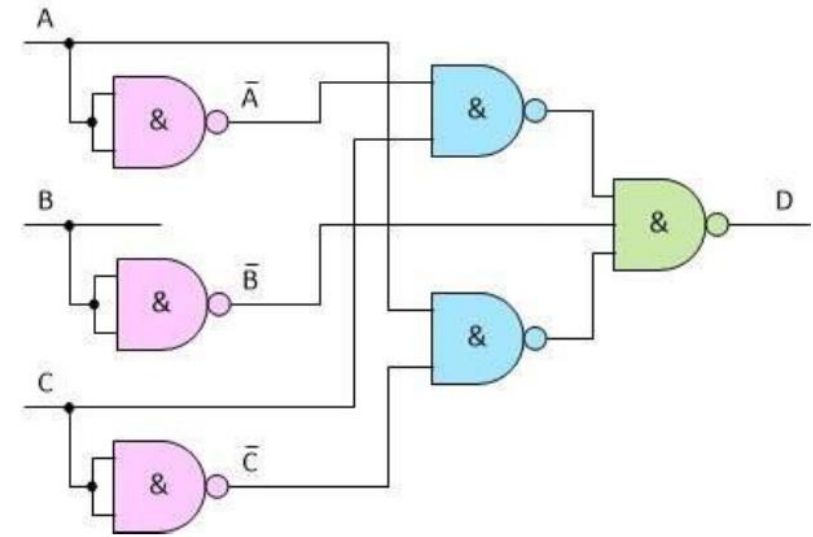
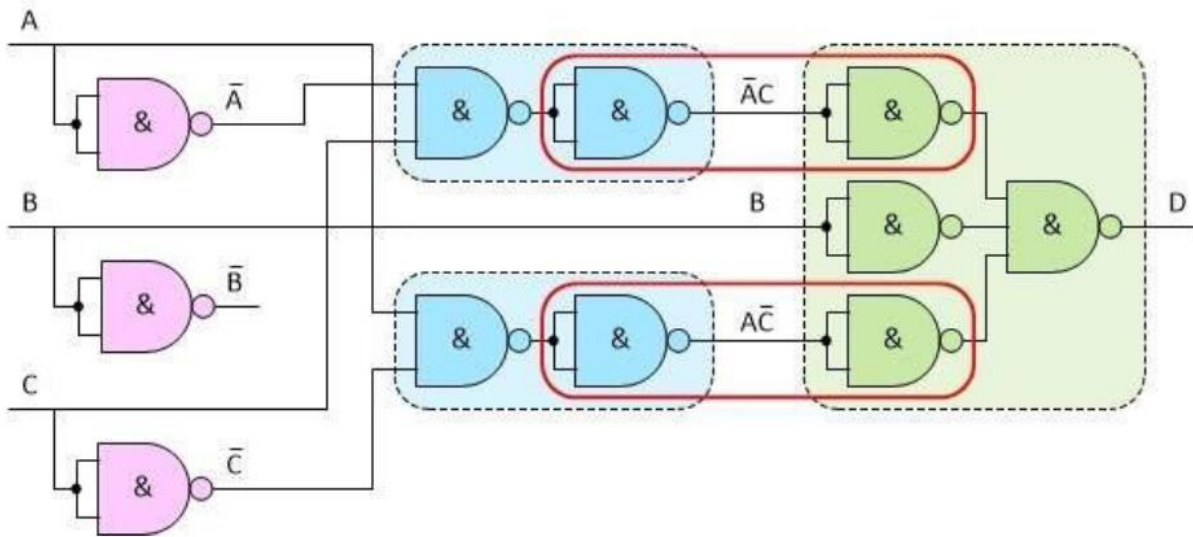
Karnaugh Haritaları

- $D = A'BC' + ABC' + AB'C' + A'B'C + A'BC + ABC$ ifadesini NAND kapıları kullanarak tasarlayınız.
- $D = A'C + AC' + B$



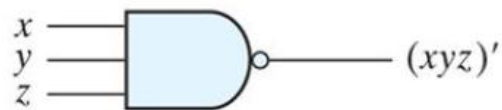
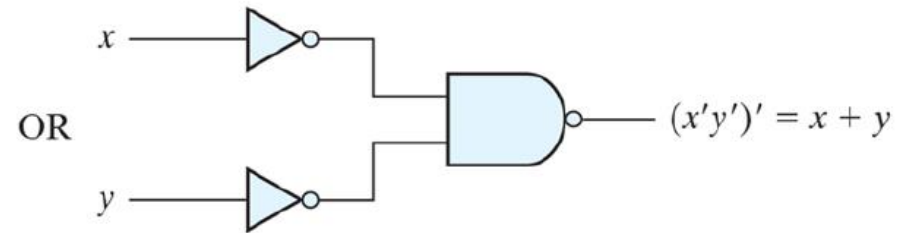
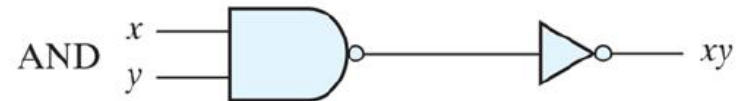
Karnaugh Haritaları

- $D = A'BC' + ABC' + AB'C' + A'B'C + A'BC + ABC$ ifadesini NAND kapıları kullanarak tasarlayınız.
- $D = A'C + AC' + B$

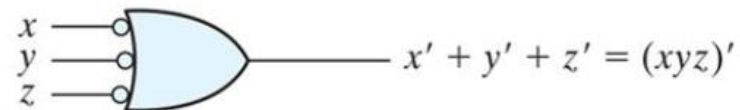


Karnaugh Haritaları

- AND ve OR gösteriminden NAND gösterimine geçiş



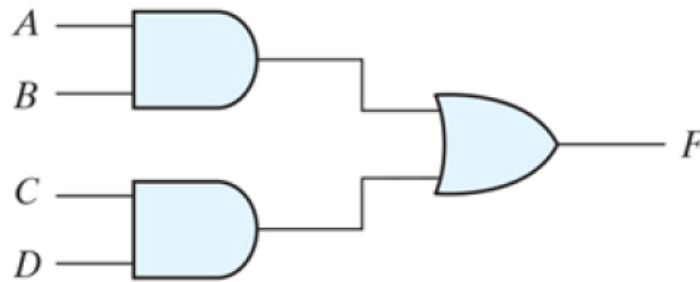
(a) AND-invert



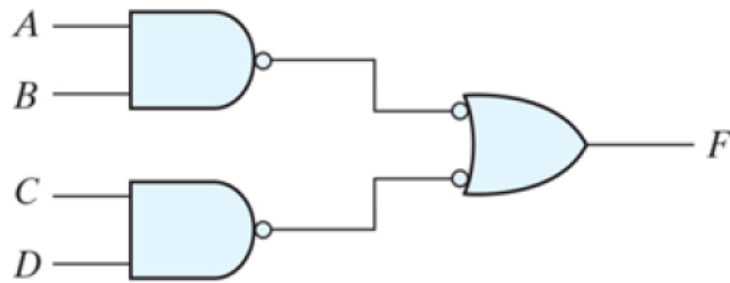
(b) Invert-OR

Karnaugh Haritaları

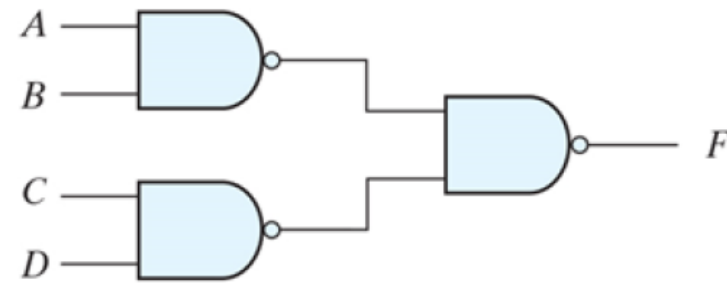
- $F=AB+CD$



(a)



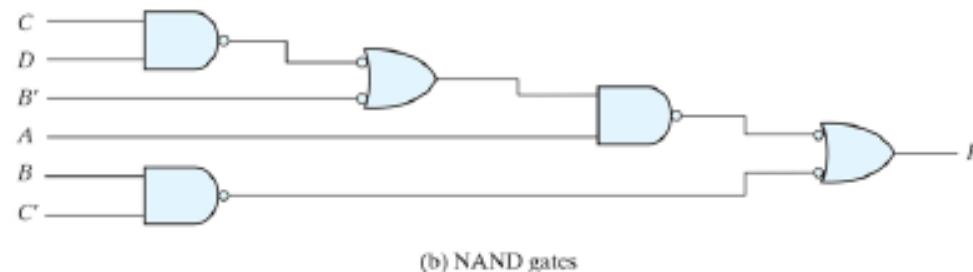
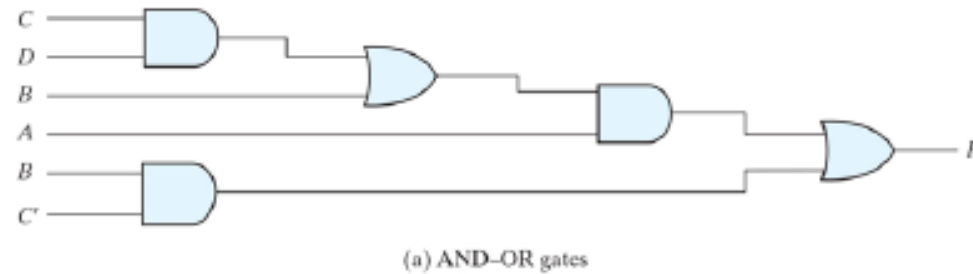
(b)



(c)

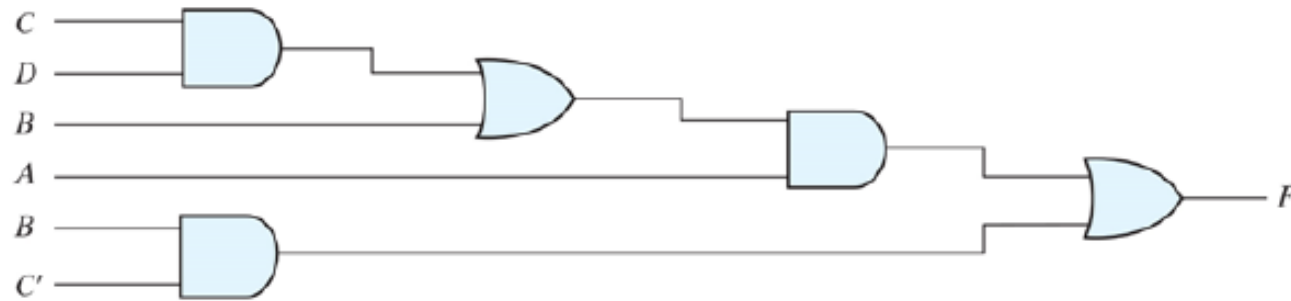
Karnaugh Haritaları

- **AND ve OR gösteriminden NAND gösterimine geçiş**
- Bütün AND kapıları AND-Tersi sembolleri kullanılarak NAND kapılarına çevrilir.
- Bütün OR kapıları Ters-OR sembolleri kullanılarak NAND kapılarına çevrilir.
- Devredeki tüm bağlantılar kontrol edilir. Aynı çizgi üzerinde başka küçük çember ile devam etmeyen her bir kablo girişi için giriş kalıbının tersi alınır

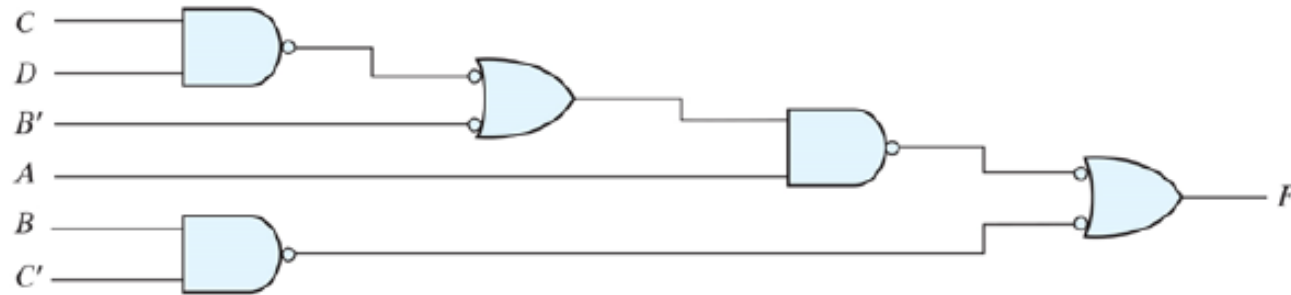


Karnaugh Haritaları

- $F = A(CD + B) + BC'$



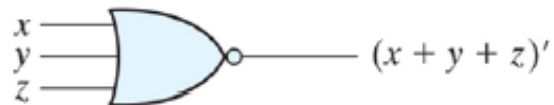
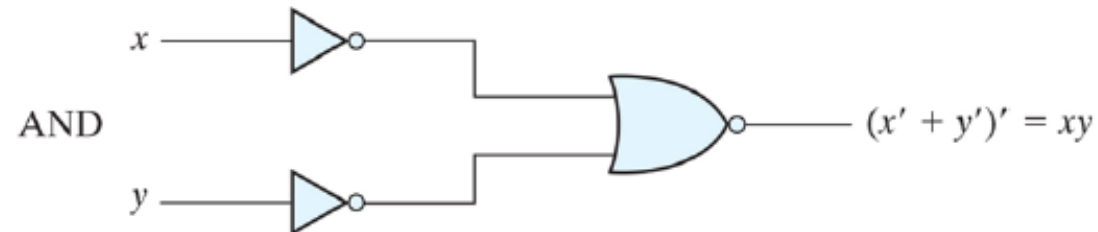
(a) AND-OR gates



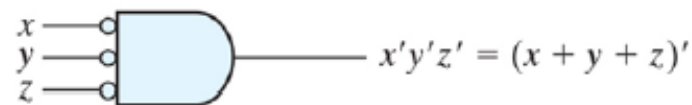
(b) NAND gates

Karnaugh Haritaları

- **AND-OR Gösteriminden NOR Gösterimine geçiş**



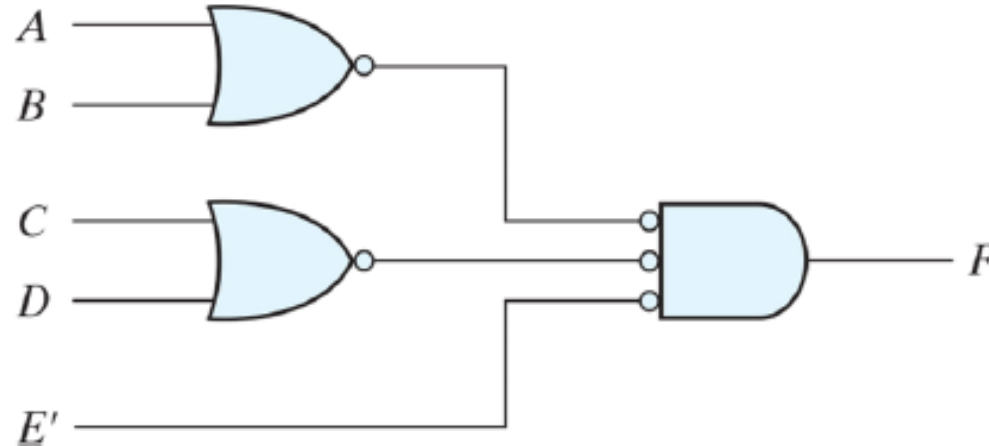
(a) OR-invert



(b) Invert-AND

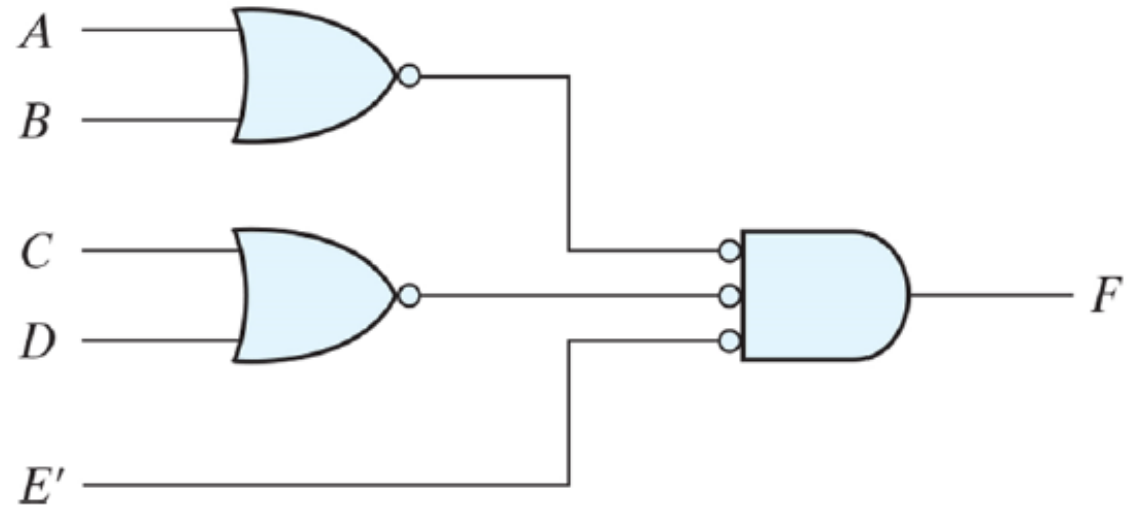
Karnaugh Haritaları

- **AND ve OR gösteriminden NOR gösterimine geçiş**
- Bütün OR kapıları OR-Tersi sembolleri kullanılarak NOR kapılarına çevrilir.
- Bütün AND kapıları ters-AND sembolleri kullanılarak NOR kapılarına çevrilir.
- Devredeki tüm bağlantılar control edilir. Aynı çizgi üzerinde başka küçük çember ile devam etmeyen her bir kablo girişi için giriş kalıbının tersi alınır



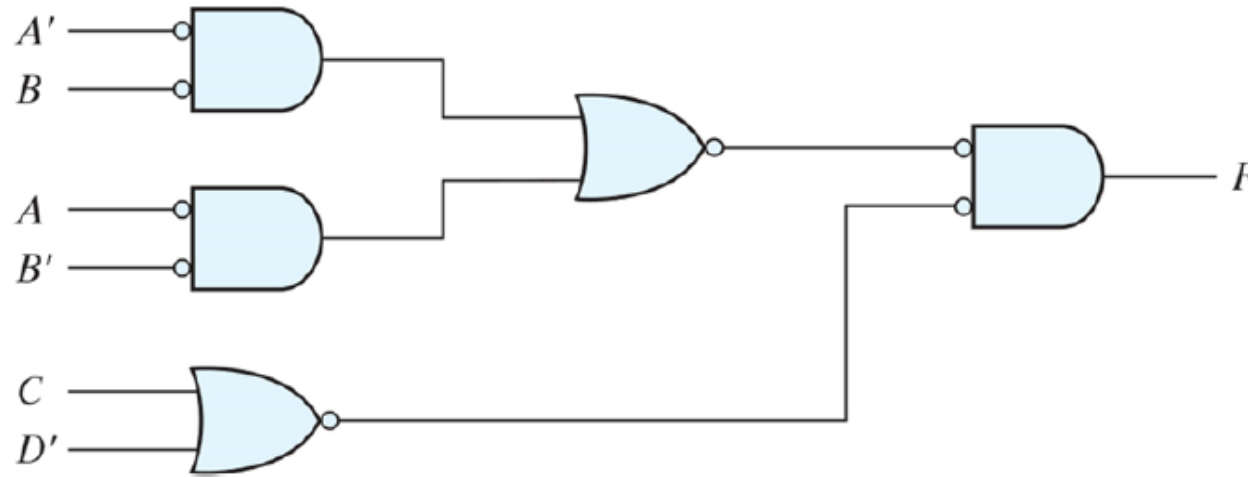
Karnaugh Haritaları

- $A = (A+B)(C+D)E$ ifadesinin gösterimi



Karnaugh Haritaları

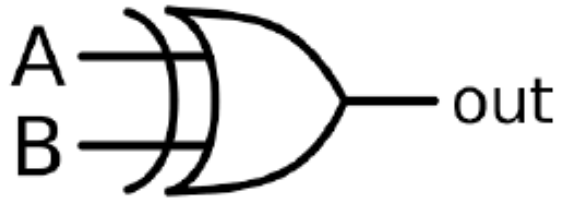
- $F=(AB'+A'B)(C+D')$ ifadesinin gösterimi



Karnaugh Haritaları

Dışlamalı Veya (Exclusive-Or; XOR) Fonksiyonu

- EX-OR(XOR) simgesiyle ifade edilir.
- $x*y=xy'+x'y$ eşitliğini gerçekler.
- Yalnızca x ya da y den herhangi biri bire eşitse çıkış 1 olur.
- İkilik bilgilerin iletimindeki hata tespitinde kullanılır.



Karnaugh Haritaları

Dışlamalı Veya (Exclusive-Or; XOR) Fonksiyonu

- XOR $x \oplus y = xy' + x'y$
 - Eğer yalnızca x 1 ise veya eğer yalnızca y 1 ise 1

- XNOR $(x \oplus y)' = xy + x'y'$
 - İki girdi de 1 veya iki girdi de 0 ise 1

- Bazı XOR eşitlikleri

$$x \oplus 0 = x$$

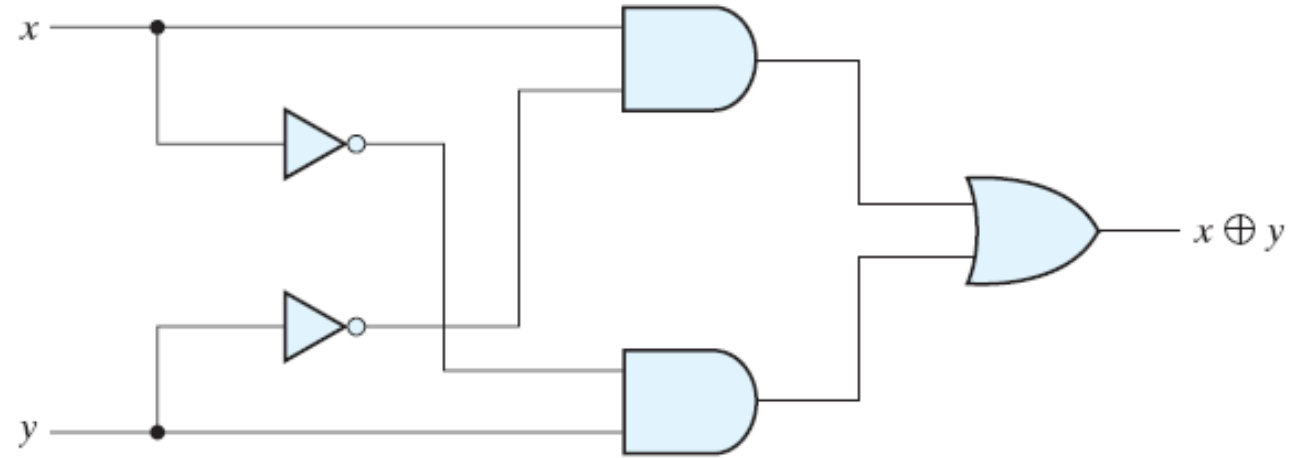
$$x \oplus 1 = x'$$

$$x \oplus x = 0$$

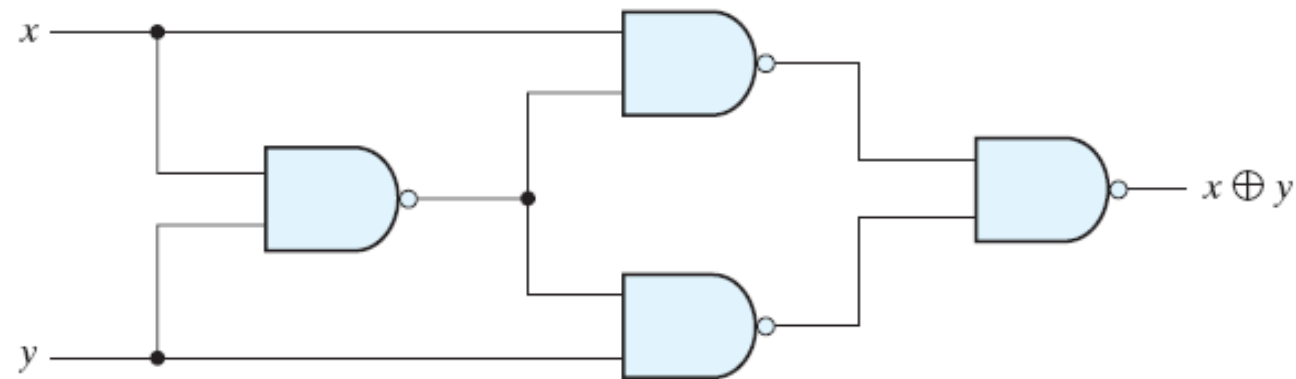
$$x \oplus x' = 1$$

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

XOR Gerçekleştirmeleri



(a) With AND-OR-NOT gates



(b) With NAND gates

Karnaugh Haritaları

- **Tek Fonksiyon**

- Aslen XOR ve XNOR kapıları sadece 2 giriş için tanımlıdır. Fakat 3 veya daha fazla değişkene genişletilebilir.
- İki den fazla giriş için XOR a “tek” fonksiyonu XNOR a “çift” fonksiyonu denir.
- 3 den fazla değişken için 3 den fazla değişken için , XOR tek fonksiyon olarak adlandırılır.
 - Değerlerine bakıldığında tek sayıda 1 içeren $2^n/2$ minterm’ün mantıksal toplamı. Eğer $n=3$ ise, 4 minterm tek sayıda 1’e sahiptir
- Değişkenlerdeki 1 lerinlerin toplam sayısı tek ise fonksiyon 1 sonucu verir.

Karnaugh Haritaları

Çift Fonksiyon

- Aslen XOR ve XNOR kapıları sadece 2 giriş için tanımlıdır. Fakat 3 veya daha fazla değişkene genişletilebilir.
- 3 den fazla değişken için 3 den fazla değişken için , XNOR çift fonksiyon olarak adlandırılır.
- Değişkenlerdeki 1 lerinlerin toplam sayısı tek ise fonksiyon 1 sonucu verir.

Karnaugh Haritaları

- $A * B * C = (AB' + A'B)C' + (AB + A'B')C = AB'C' + A'BC' + ABC + A'B'C = \Sigma(1, 2, 4, 7)$

		B			
		BC			
A	0	00	01	11	10
	1	m_4	m_5	m_7	m_6
m_0			1		1
m_1					
m_2					
m_3					

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

		B			
		BC			
A	0	00	01	11	10
	1	m_4	m_5	m_7	m_6
m_0		1		1	
m_1					
m_2					
m_3					

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

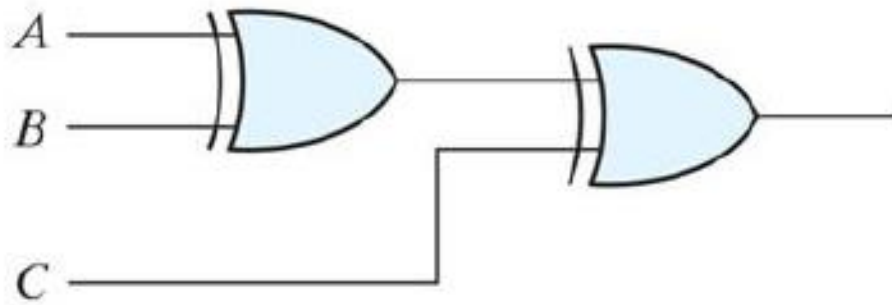
Karnaugh Haritaları

BC		B			
		00	01	11	10
A	0	m_0	m_1	m_3	m_2
	1	m_4	m_5	m_7	m_6
		1		1	

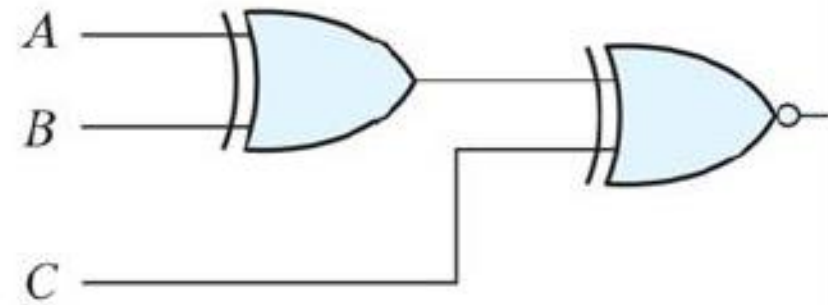
(a) Odd function $F = A \oplus B \oplus C$

BC		B			
		00	01	11	10
A	0	m_0	m_1	m_3	m_2
	1	m_4	m_5	m_7	m_6
		1			1

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



(a) 3-input odd function



(b) 3-input even function

Karnaugh Haritaları

$AB \backslash CD$		C			
		00	01	11	10
A	00	m_0	m_1 1	m_3	m_2 1
	01	m_4 1	m_5	m_7 1	m_6
	11	m_{12}	m_{13} 1	m_{15}	m_{14} 1
	10	m_8 1	m_9	m_{11} 1	m_{10}
		D			

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

$AB \backslash CD$		C			
		00	01	11	10
A	00	m_0 1	m_1	m_3 1	m_2
	01	m_4	m_5 1	m_7	m_6 1
	11	m_{12} 1	m_{13}	m_{15} 1	m_{14}
	10	m_8	m_9 1	m_{11}	m_{10} 1
		D			

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

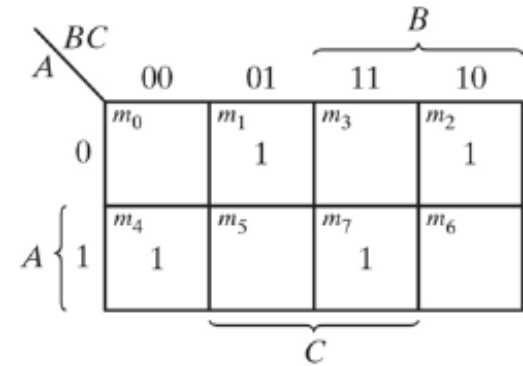
Karnaugh Haritaları

- Parity Biti- Eşlik Biti (Hata Kontrolü) : İkilik bilgilerin iletimindeki hataların tespiti için kullanılır.
- Eşlik Biti: 1'lerin sayısını tek ya da çift yapmak için kullanılan bit
- Eşlik bitini mesaja eklenir ve iletilir. Alıcı taraf hata kontrolü yapar.
- Eşlik Üretici: Eşlik biti oluşturan devreler
- Eşlik Kontrolcüler: Alıcıda eşliği kontrol eden devrelerdir.

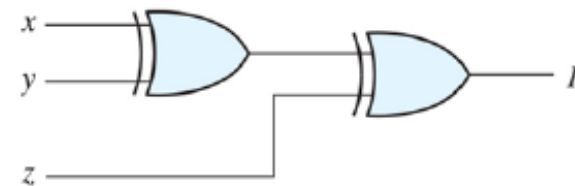
Karnaugh Haritaları

- Eşlik Üreteci: Gönderici tarafta parity biti üreten devreye *parity üreticisi (jeneratörü)* denir. Parity biti, ikili mesaja eklenen ekstra bir bittir. Eğer 1'lerin sayısı tek ise, parity bit 1; değilse 0. Parity biti XOR fonksiyonu ile üretilebilir.

Three-Bit Message			Parity Bit
x	y	z	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



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

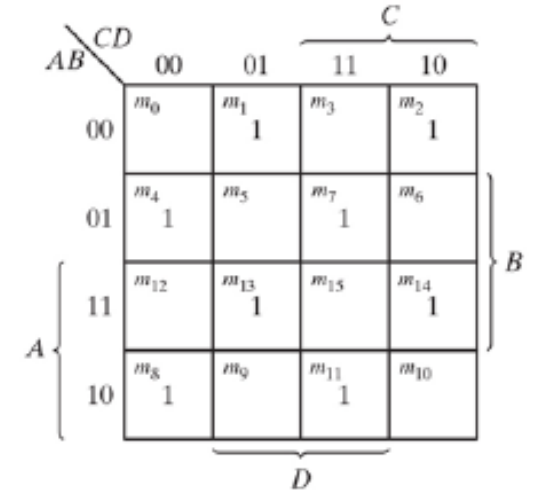


(a) 3-bit even-parity generator

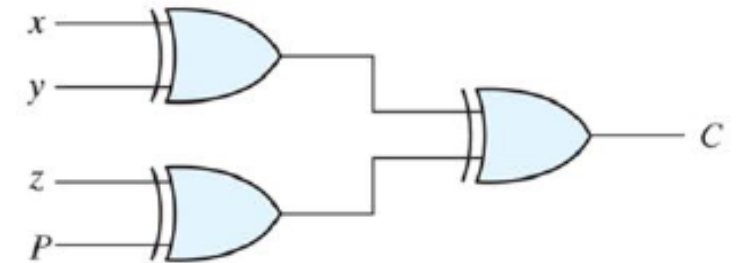
Karnaugh Haritaları

- Eşlik *denetçisi* : Bitler hedefe parity bit ile birlikte iletilir.
- Alıcı tarafta parity bit'i kontrol eden devreye *parity denetçisi* denir.
- Parity denetçisi XOR kapılarıyla gerçekleştirilebilir.

Four Bits Received				Parity Error Check
x	y	z	P	C
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0



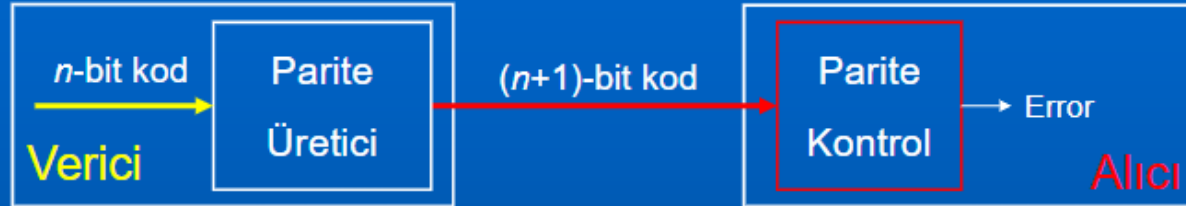
(a) Odd function $F = A \oplus B \oplus C \oplus D$



(b) 4-bit even-parity checker

Karnaugh Haritaları

Parite Üretimi ve Kontrolü



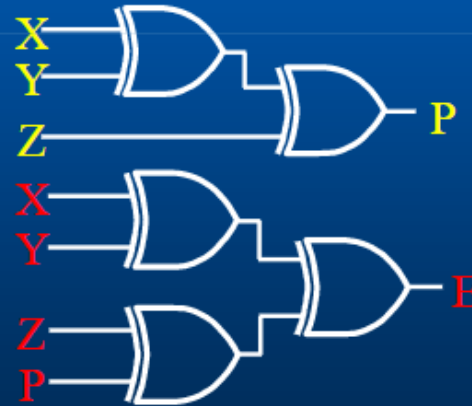
- 3-bit kod için çift parite üretim ve kontrolü gösterilmektedir

- 3-bit tek fonksiyonu çift parite bitini üretmek için kullanılır

- 4-bit tek fonksiyonu gönderilen kodlardaki parite hatasını kontrol eder

- $(X,Y,Z) = (0,0,1)$ parite üretimi sonucu $(X,Y,Z,P) = (0,0,1,1)$ ve $E = 0$

- Eğer iletim sırasında bir bit bozulursa E çıkışından 1 alınır



Karnaugh Haritaları

Cümlelerin Boolean Cebirine Dönüştürülmesi:

ÖRNEK 1: Eğer Pazartesi gecesi ise ve ödevlerini bitirmişse Meryem TV izler.

- Burada «eğer» ve «ve» bağlaçları hiçbir ifadeye dahil değildir; bu bağlaçlar ifadeler arasındaki ilişkiyi gösterir.
- Bir ifadenin doğru yada yanlış olduğunu gösteren bir değişken ile ifadeleri inceleyelim:
- $F = 1$ Eğer «Meryem TV izlerse» doğru;
- değilse, $F = 0$ («Meryem TV izlemezse»).
- $A = 1$ Eğer «Pazartesi gecesi ise» doğru; değilse, $A = 0$.
- $B = 1$ Eğer «eğer Meryem ödevini bitirmişse» doğru; değilse $B = 0$.
- F doğruyken eğer A ve B ifadelerinin ikisi de doğruysa cümle şu şekilde Boole ifadeye çevrilebilir:

$$F = A \cdot B$$

Karnaugh Haritaları

- **ÖRNEK 2:** Aşağıdaki örnek bir cümleden cebirsel bir ifadeye nasıl geçiş yapılabileceğini göstermektedir. Bir alarm devresi aşağıda tanımlanan biçimde tasarlanmak isteniyor:
- **Alarm çalacaktır; eğer ve ancak (Ancak ve Ancak) alarm anahtarı açıksa ve kapı kapalı değilse, veya saat 18:00 den sonra ise ve pencere kapalı değilse.**
- İlk adım yukarıdaki cümlelerin cebirsel karşılığını, her bir ifadeye bir değişken verilerek yazılması işlemidir.

Karnaugh Haritaları

- Z = Alarm çalacak/ çalmayacak:
- $Z=1$ Eğer alarm çalarsa, $Z'=1$ ($F=0$) Eğer alarm çalmazsa
- A = Alarm anahtarı açık/kapalı: $A=1$ Eğer alarm anahtarı açık ise, $A'=1$ ($A=0$) Eğer alarm anahtarı açık değilse (kapalı ise)
- B = Kapı açık/kapalı
- $B=1$ Eğer kapı kapalı ise, $B'=1$ ($B=0$) Eğer kapı açık ise (Kapalı değilse)
- C = Saat 18:00 den önce/sonra
- $C=1$ Eğer 18:00den sonra ise. $C'=1$ ($C=0$) 18:00den sonra değilse
- D = Pencere açık/kapalı,
- $D=1$ Eğer pencere kapalı ise, $D'=1$ ($D=0$) Eğer pencere kapalı değilse Bu tanımlardan sonra F nin Boole (lojik) ifadesi aşağıdaki gibi olur:
- **$Z=A.B'+C.D'$**

Karnaugh Haritaları

Z =Alarm çalacaktır= Eğer ve ancak (if) alarm anahtarı açık ise (A) VE Kapı kapalı değilse (B') VEYA Saat 18:00den sonra ise (C) VE pencere kapalı değilse (D').

$Z=A.B'+C.D'$ Bu lojik fonksiyonu (Boole ifadesini) gerçekleştirecek lojik devre aşağıda verilmiştir.

Karnaugh Haritaları

$$Z = A.B' + C.D'$$

