



Lojik Tasarım

Ders 6

Kaynak:

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

Beş Değişkenli Karnough Diyagramı

Beş değişkenli çözümlemede $2^5=32$ durum bulunmaktadır.

$a=0$

d, e b, c		d, e			
		00	01	11	10
b, c	00	0	1	3	2
	01	4	5	7	6
	11	12	13	15	14
	10	8	9	11	10

$a=1$

d, e b, c					
		00	01	11	10
b, c	00	16	17	19	18
	01	20	21	23	22
	11	28	29	31	30
	10	24	25	27	26

Beş Değişkenli Karnough Diyagramı

$f(a, b, c, d, e) = \sum(0,2,4,6,9,13,21,23,25,29,31)$ lojik ifadesini Karnough haritası yöntemi ile sadeleştiriniz.

a=0				
d, e b, c				
	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

a=1				
d, e b, c				
	00	01	11	10
00	16	17	19	18
01	20	21	23	22
11	28	29	31	30
10	24	25	27	26

$$f = a'b'e' + ace + bd'e$$

Fazla değişkenli lojik fonksiyonlar

- 6 değişkenli bir fonksiyonda $2^6=64$ değişik durum bulunmaktadır
- 6 ve daha fazla değişkene sahip lojik fonksiyonların basitleştirilmesinde Karnaugh haritası yöntemini kullanmak zor ve karmaşık olabilir.
- Bu nedenle bu tür fonksiyonların sadeleştirilmesinde farklı yöntemler tercih edilir.
- İlerleyen derslerde farklı yöntemlere değinilecektir.

Farketmez (etkisiz) Koşullar (Don't Care)

Aşağıda verilen doğruluk tablosu verilen lojik fonksiyonu sadeleştiriniz

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>f</i>
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0

$$f = bd' + b'd + cd$$

<i>a, b</i> \ <i>c, d</i>					
		00	01	11	10
<i>a, b</i>	00				
	01				
	11				
	10				

Farketmez (etkisiz) Koşullar (Don't Care)

Verilen her iki lojik ifade aynıdır

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

$$f(w, x, y, z) = \sum_m(1,3,7,11,15) + \sum_d(0,2,5)$$

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

y, z w, x				
	00	01	11	10
00				
01				
11				
10				

$$f = yz + w'x'$$

Örnek:

- En anlamlı biti a en az anlamlı biti d olan ve girişleri abcd olarak isimlendirilmiş bir lojik sisteme BCD sayılar uygulanmaktadır. Sisteme Uygulanan sayı 4'den küçükse 0, diğer durumlarda ise 1 üretmektedir. İlgili lojik sistemin
 - a) Doğruluk tablosunu oluşturunuz
 - b) Karnough haritası yöntemi ile sadeleştiriniz
 - c) Sadeleştirilmiş ifadenin lojik devresini çizin

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>f</i>
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	

<i>a, b</i> \ <i>c, d</i>					
		<i>00</i>	<i>01</i>	<i>11</i>	<i>10</i>
<i>a, b</i>	<i>00</i>				
	<i>01</i>				
	<i>11</i>				
	<i>10</i>				

Faktörizasyon

- Bir lojik ifadede bir değişkenin, değişken grubunun ve bir grubun kısmi parantez dışına alınarak lojik ifadenin tamamının ya da bir kısmının çarpanlarına ayrılması işlemidir.
- Faktörizasyon genellikle minimumlaştırmadan sonra uygulanır.
- Bazı fonksiyonlar minimumlaştırılmaz haldeyken bile faktörizasyon uygulanabilir.

$$f = ab'c' + a'bc' + a'b'c + abc$$

$$f = a(b'c' + bc) + a'(bc' + b'c)$$

$$f = a(b \oplus c)' + a'(b \oplus c)$$

$$ax' + a'x = a \oplus x$$

$$f = a \oplus (b \oplus c)$$

Toplamların çarpımı ve Çarpımların toplamları biçimindeki ifadelerin sadeleştirilmesi

$f = b'd' + b'c' + a'c'd$ lojik ifadesini doğruluk tablosunu oluşturunuz

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>f</i>
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Toplamların çarpımı ve Çarpımların toplamları biçimindeki ifadelerin sadeleştirilmesi

$$f = b'd' + b'c' + a'c'd$$

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>f</i>
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

İfadenin değili yada tümleyeni (f') doğruluk tablosundaki 0 konumları değerlendirilerek bulunabilir

<i>c, d</i> <i>a, b</i>		00	01	11	10
00				0	
01	0			0	0
11	0	0	0	0	0
10				0	

$$f' = ab + cd + bd'$$

Çarpımların toplamları biçimindeki ifadelerin sadeleştirilmesi

$$f' = ab + cd + bd'$$

Lojik ifadesinin tümleyenini hesaplırsak fonksiyonun kendisi elde edilir

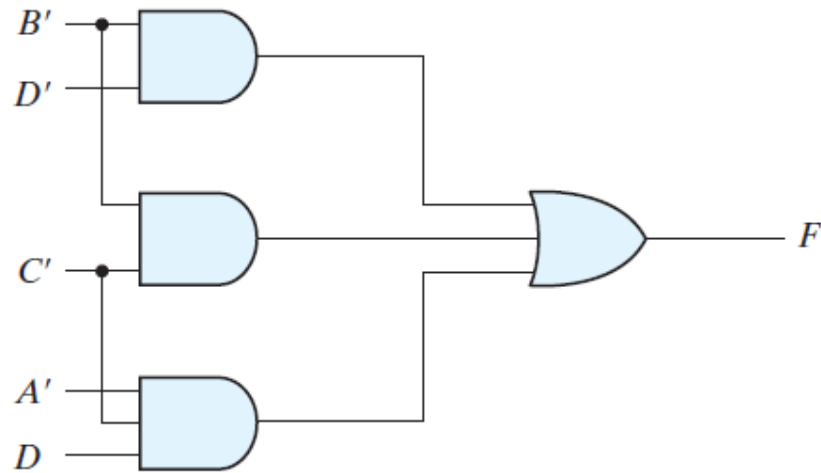
DeMorgan kuralı uygulanırsa;

$$f = (a' + b')(c' + d')(b' + d)$$

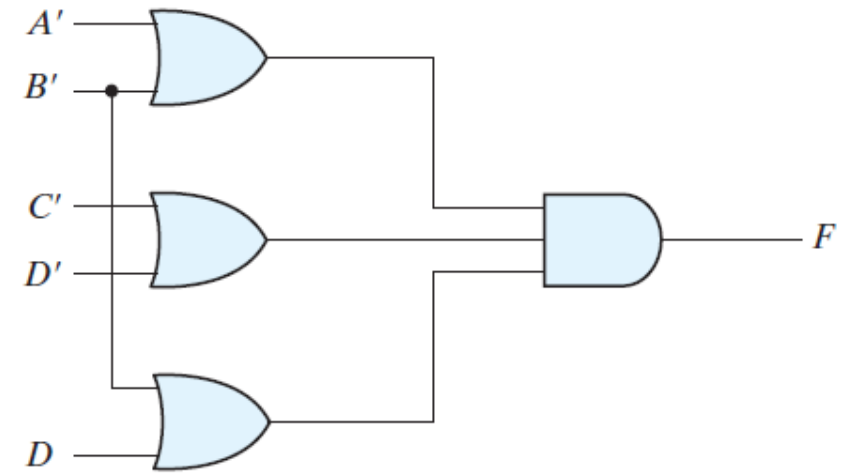
şeklinde elde edilir

c, d a, b		00	01	11	10
00				0	
01	0			0	0
11	0	0	0	0	0
10				0	

Toplamların çarpımı ve Çarpımların toplamları biçimindeki ifadelerin sadeleştirilmesi



(a) $F = B'D' + B'C' + A'C'D$



(b) $F = (A' + B')(C' + D')(B' + D)$

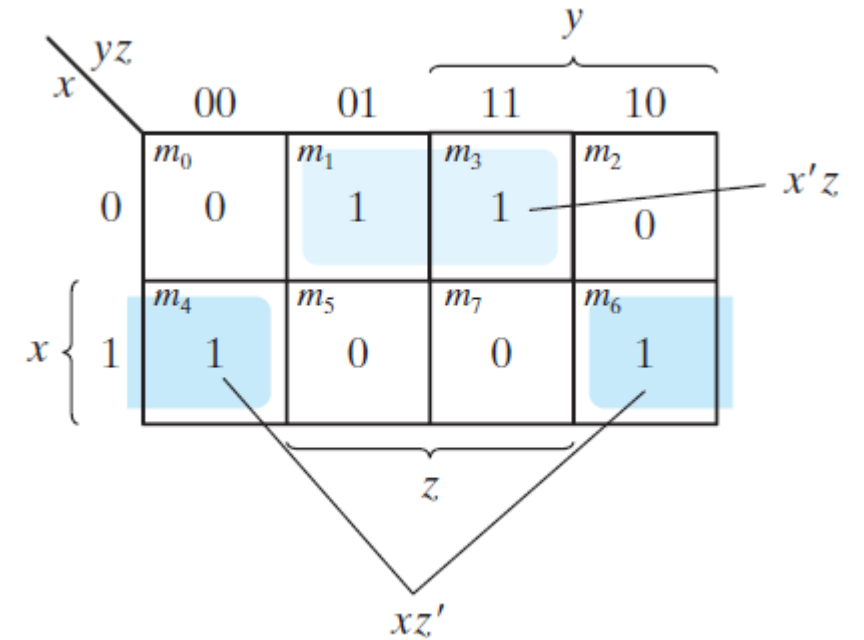
Toplamların çarpımı ve
Çarpımların toplamları biçimindeki ifadelerin sadeleştirilmesi

$$F(x, y, z) = \Sigma(1, 3, 4, 6)$$

$$F(x, y, z) = \Pi(0, 2, 5, 7)$$

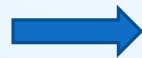
Table 3.1
Truth Table of Function F

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0



$$F = x'z + xz'$$

$$F' = xz + x'z'$$



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

Lojik Devrelerin NAND ve NOR Kapıları ile Gerçeklenmesi (NAND – NOR Lojik Teknikleri)

- Sayısal devreler VE ve VEYA kapılarından çok VEDEĞİL (NAND) ve VEYADEĞİL (NOR) kapıları ile gerçekleştirilir.
- NAND ve NOR kapılarının elektronik elemanlarla üretilmesi daha kolay ve ucuzdur.
- NAND ve NOR kapılarının üstünlüğünden dolayı sayısal devre tasarımında AND, OR ve NOT cinsinden verilen Boole fonksiyonlarının eşdeğer NAND ve NOR lojik fonksiyonlarına çevirmek için teknikler geliştirilmiştir.

NAND Lojik Tekniği - Eşdeğer Devreler

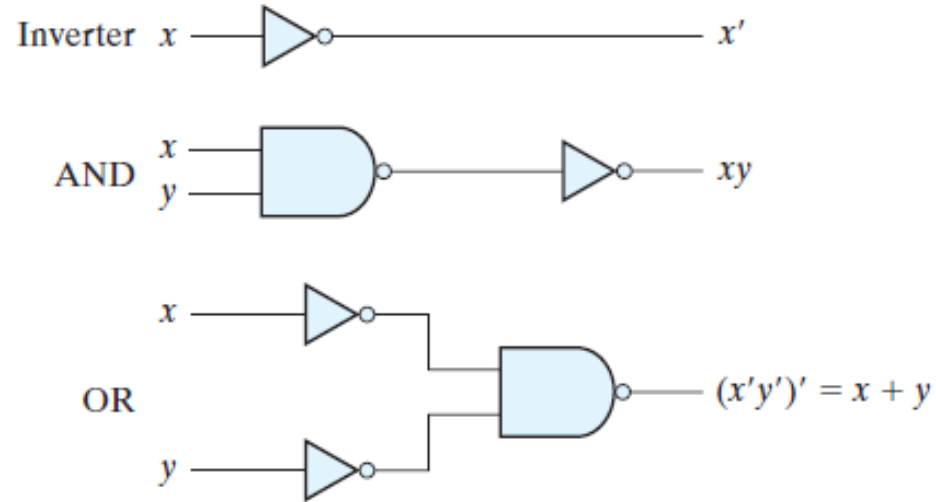


FIGURE 3.16
Logic operations with NAND gates

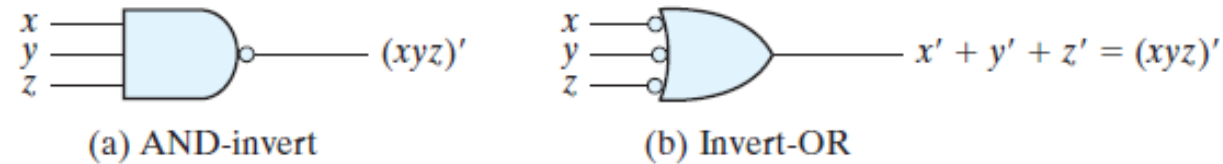
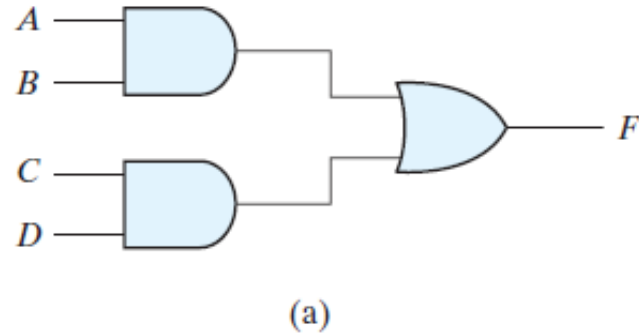


FIGURE 3.17
Two graphic symbols for a three-input NAND gate

NAND Lojik Tekniği

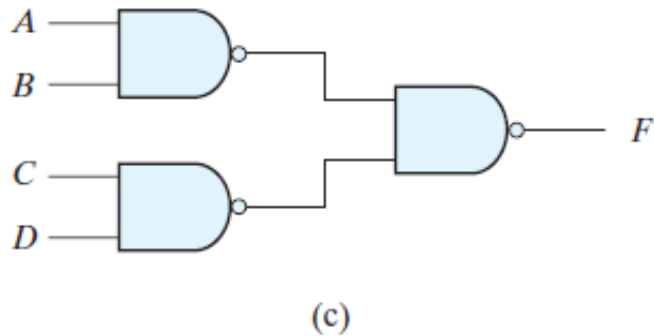
$$f = AB + CD$$



Çarpımların toplamı şeklindeki ifadenin iki kez tümleyenini alırsak ifadenin kendisini elde edebiliriz

$$f = AB + CD$$

$$f = \overline{\overline{AB + CD}}$$



$$f = \overline{(\overline{AB}) \cdot (\overline{CD})}$$

NOR Lojik Tekniği - Eşdeğer Devreler

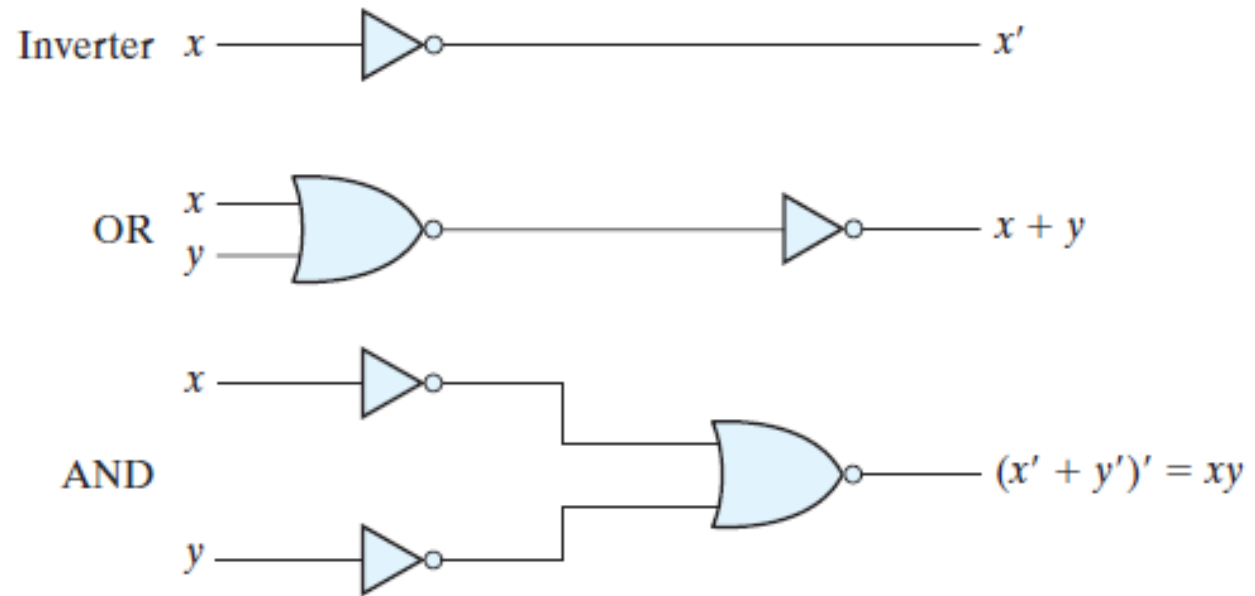
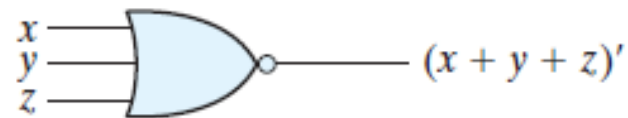
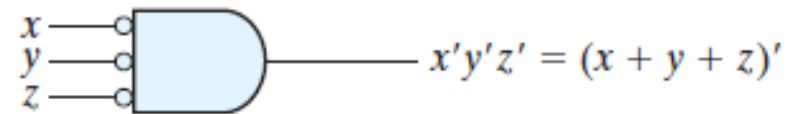


FIGURE 3.22
Logic operations with NOR gates



(a) OR-invert



(b) Invert-AND

NOR Lojik Tekniği

Toplamların çarpımı şeklindeki ifadenin iki kez tümleyenini alırsak ifadenin kendisini elde edebiliriz

$$f = (A + B) \cdot (C + D)$$

$$f = \overline{\overline{(A + B) \cdot (C + D)}}$$

$$f = \overline{\overline{(A + B)} + \overline{(C + D)}}$$