Boole Fonksiyonlarının Sadeleştirilmesi

3. Bölüm

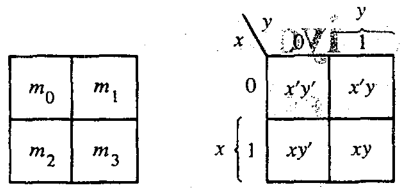
3.1 HARİTA YÖNTEMİ

Boole fonksiyonu uygulanan sayısal mantık kapılarının karmaşıklığı, doğrudan doğruya fonksiyonun uygulandığı cebirsel ifadenin karmaşıklığından kaynaklanmaktadır. Bir fonksiyonun doğruluk tablosuyla gösterilmesi benzersiz olmasına rağmen, cebirsel olarak birçok farklı şekilde ifade edilebilmektedir. 2.4. Bölümde de tartıştığımız gibi Boole fonksiyonları cebirsel yöntemlerle sadeleştirilebilir. Ne var ki bu sadeleştirme işlemi acemicedir, çünkü işlem sürecindeki birbirini izleyen her adımı tahmin etmek için kullanılan özel kurallardan yoksundur. Harita yöntemi, Boole fonksiyonlarının sadeleştirilmesi için basit ve dolaysız bir yöntem sağlar. Bu yöntem doğruluk tablosunun şematik biçimi, ya da Venn şemasının uzantısı olarak değerlendirilebilir. İlk defa Veitch (1) tarafından önerilen ve Karnaugh (2) tarafından biraz değiştirilen harita yöntemi, ayrıca "Veitch şeması" veya "Karnaugh haritası" olarak da bilinmektedir.

Harita, karelerden oluşan bir şemadır. Her bir kare bir mintermi gösterir. Boole fonksiyonları mintermlerin toplamı olarak ifade edilebildiği için fonksiyon, haritada grafiksel olarak fonksiyonun mintermleri içerdiği karelerle çevrili alanlarla gösterilebilir. Aslında harita, bir fonksiyonun standard bir formda ifade edilebileceği olası bütün yolları içeren görsel bir şemayı temsil eder. Çeşitli desenleri gören kullanıcı, aynı fonksiyon için alternatif cebirsel ifadeler geliştirebilir ve bunlardan en sadesini seçebilir. Bir çarpımlar toplamındaki veya toplamlar çarpımındaki en sade cebirsel ifadenin, minimum sayıda literali olan ifade olduğunu varsayacağız. (Bu ifadenin benzersiz olması da gerekmez.)

3.2 İKİ VE ÜÇ DEĞİŞKENLİ HARİTALAR

İki değişkenli bir harita Şekil 3.1'de verilmiştir. İki değişken için dört minterm vardır; dolayısıyla haritada her minterm için bir tane olmak üzere dört kare vardır. Kareler ve iki değişken arasındaki ilişkiyi göstermek amacıyla harita (b)'de yeniden çizilmiştir. Her satır ve sıra için işaretlenen 0'lar ve 1'ler, sırasıyla x ve y değişkenlerinin değerini gösterir. x değişkeninin, satır 0'da işaretli ve satır 1'de işaretsiz gözüktüğüne dikkat edin. Benzer bir şekilde y değişkeni, 0 sütununda işaretli, 1



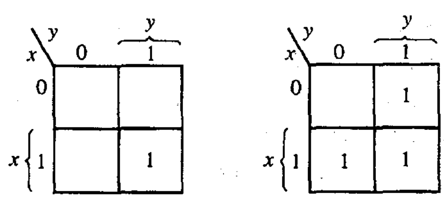
(a) (b)

Şekil 3-1 İki değişkenli harita

sütununda ise işaretsiz gözükmektedir.
Mintermleri belli bir fonksiyona ait olan kareleri işaretlersek; iki değişkenli harita, iki değişkenin 16 Boole fonksiyonundan herhangi birisini temsil etmenin kullanışlı başka bir yolunu sağlamış olur. Örnek olarak, xy fonksiyonu Şekil 3.2(a)'da verilmiştir. xy, m_3'e eşit olduğu için, m_3'e ait karenin içine 1 konmuştur. Aynı şekilde, $x + y$ fonksiyonu, Şekil 3.2(b)'deki haritada 1'le işaretli üç kareyle gösterilmiştir. Bu kareler şu fonksiyonun mintermlerinden bulunmuştur:

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

Üç kare, x değişkeninin ikinci satırdaki, y değişkeninin ise ikinci sütundaki kesişme noktasıyla da belirlenebilirdi; bu, x veya y değişkenine ait alanı kapsar.



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

Şekil 3.2 Fonksiyonun haritada gösterilmesi

Üç değişkenli bir harita Şekil 3.3'te verilmiştir. Üç ikili değişken için sekiz minterm vardır. Bu nedenle harita sekiz kareden oluşur. Mintermlerin, ikili bir sırada değil, Tablo 1.4'te verilen yansımalı koda benzer bir sırada verildiğine dikkat edin. Bu sıranın özelliği, verilen sırada sadece bir bitin 1'den 0'a veya 0'dan 1'e değişmesidir. (b) şıkkında çizilen harita, karelerle üç değişken arasındaki ilişkiyi göstermek açısından her satır ve sütundaki sayılarla işaretlidir. Örneğin m_5 işaretli kare, satır 1, sütun 01'e tekabül eder. Bu iki sayı birleştirildiği zaman ondalık 5'in eşdeğeri olan ikili sayı 101 elde edilir. m_5 = $xy'z$ karesine bakmanın başka bir yolu da, x işaretli satır ve $y'z$ sütun (sütun 01) ait olduğunu düşünmektir. Her bir

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

(a)

$x \backslash yz$	00	01	11	10
0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
1	$xy'z'$	$xy'z$	xyz	xyz'

(b)

Şekil 3.3 Üç değişkenli harita

değişkenin 1 olduğu dört kare ve 0 olduğu dört kare olduğuna dikkat edin. Değişken, 1'e eşit olduğu dört karede işaretsiz, 0'a eşit olduğu karelerde işaretli gözükmektedir. Kolaylık açısından değişkeni işaretsiz olduğu dört karenin altına harf sembolüyle yazıyoruz.

Haritanın; Boole fonksiyonlarını sadeleştirmedeki yararını anlamak için, bitişik karelerin temel özelliğini kavramamız gerekir. Haritadaki bitişik her iki kare, sadece, bir karede işaretli, diğerinde işaretsiz olan bir değişken kadar farklıdır. Örneğin m_5 ve m_7 bitişik karelerde bulunmaktadır. y değişkeni m_5'te işaretli m_7'de işaretsiz'dir, buna karşılık diğer iki değişken karelerin ikisinde de aynıdır. Boole cebirinin önermelerinden, bitişik karelerdeki iki mintermin toplamlarının, sadece iki literal içeren tek bir VE terimine sadeleştirilebileceği sonucu çıkar. Açıklık kazandırmak açısından, m_5 ve m_7 gibi iki bitişik karenin toplamını ele alalım:

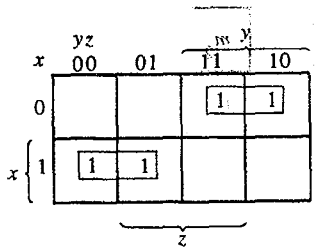
$$m_5 + m_7 = xy'z + xyz = xz\,(y' + y) = xz$$

Bu iki kare arasındaki fark, y değişkenidir; ki bu da iki mintermin toplamı alındığında iptal edilebilir. Bu nedenle bitişik karelerdeki iki mintermin VEYA'lanması, farklı olan değişkenin atılmasına neden olacaktır. Aşağıdaki örnek, bir Boole fonksiyonunun harita yardımıyla sadeleştirilmesi yöntemine açıklık getirecektir.

ÖRNEK 3.1: Aşağıdaki Boole fonksiyonunu sadeleştirin:

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

İlk önce Şekil 3.4'te gösterildiği gibi, fonksiyonu temsil etmek için ihtiyaç duyulduğu gibi her kare 1 ile işaretlenir. Bu iki yoldan yapılabilir: her bir mintermi ikili bir sayıya çevirip ilgili kareyi 1'le işaretleyerek, ya da her bir terimde değişkenlerin çakışmasını elde ederek. Örneğin $x'yz$ terimine karşılık gelen ikili sayı 011'dir ve 011 karesindeki m_3 mintermini temsil



Şekil 3.4 Örnek 3.1 için harita; $x'yz + x'yz' + xy'z' + xy'z = x'y + xy'$

eder. Kareyi anlamanın ikinci bir yolu da haritada x' literalini ilk satırdaki dört karede, y literalini sağdaki iki sütundaki dört karede ve z literalini ortadaki iki sütundaki dört karede olduğunu gözleyerek x', y ve z değişkenlerinin çakışmasını bulmaktır. Bu üç literalin üçüne birden ait olan alan, ilk satırın üçüncü sütununda bulunan tek karedir. Benzer bir şekilde F fonksiyonuna ait olan diğer üç alan, haritada 1 ile işaretlenir. Böylece fonksiyon, Şekil 3.4'teki gibi 1'le işaretli dört kareyi içeren alanla temsil edilir. Bir sonraki adım, söz konusu alanı bitişik karelere bölmektir. Bunlar haritada her biri iki adet 1 içeren iki diktörtgenle gösterilmiştir. Üst sağdaki dikdörtken, $x'y$ ile kapsanan alanı; alt sol ise xy' ile kapsanan alanı gösterir. Bu iki terimin toplamı, cevabı verir:

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

Daha sonra Şekil 3.3(a)'da m_0 ve m_2 veya Şekil 3.3(b)'de $x'y'z'$ ve $x'yz$"ile işaretli iki kareyi ele alın. Bu iki minterm de sadece y değişkeni farklıdır ve toplamları iki literalli bir ifadeye sadeleştirilebilir:

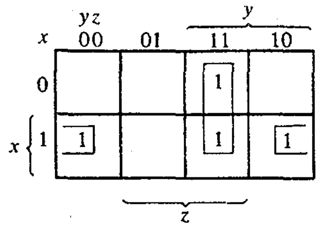
$$x'y'z' + x'yz' = x'z'$$

Sonuç olarak bitişik kareler tanımını bu ve benzeri diğer durumları kapsayacak şekilde değiştirmemiz gerekir. Bu amaçla haritanın bitişik kareler oluşturacak şekilde sağ ve sol köşeleri birbirine değecek şekilde bir yüzeye çizildiğini varsayarız.

ÖRNEK 3.2: Aşağıdaki Boole fonksiyonunu sadeleştirin:

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

Bu fonksiyonun haritası Şekil 3.5'te verilmiştir. Her biri fonksiyonun bir



Şekil 3.5 Örnek 3.2 için harita; $x'yz + xy'z' + xyz + xyz' = yz + xz'$

mintermi için olmak üzere 1'le işaretli dört kare vardır. Üçüncü sütundaki bitişik iki kare birleştirilerek iki literalli yz terimi elde edilmiştir. Yeni tanım gereği 1 içeren diğer iki kare de bitişiktir ve şemada yarım dikdörtgenle çevrili olarak gösterilmiştir. Bu iki kare birleştirildiği zaman iki literalli xz' terimi elde edilir. Sadeleştirilmiş fonksiyon şu hali alır:

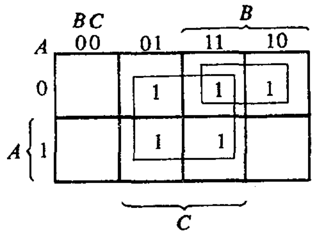
$$F = yz + xz'$$

Şimdi de üç değişkenli haritadaki dört bitişik karenin birleşimlerini ele alalım. Bu tür her birleşim, bitişik dört mintermin VEYA'lanmasını temsil eder ve sadece bir literalli bir ifadeyle sonuçlanır. Örnek olarak, m_0, m_2, m_4 ve m_6 ile gösterilen dört bitişik mintermin toplamı, aşağıda gösterildiği gibi sadece z' literaline indirgenir:

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

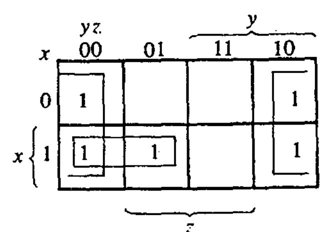
ÖRNEK 3.3: Aşağıdaki Boole fonksiyonunu sadeleştirin:

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



Şekil 3.6 Örnek 3.3 için harita; $A'C + A'B + AB'C + BC = C + A'B$

Bu fonksiyonu sadeleştirmek için kullanılan harita Şekil 3.6'da verilmiştir. Fonksiyondaki terimlerden bazıları üçten az literale sahiptir ve haritada birden fazla kareyle temsil edilmektedir. Örneğin $A'C$ terimine karşılık gelen kareleri bulmak için, A' ile (ilk satır) C literalinin (iki orta sütun) kesişmesini alır ve 001 ile 011 karelerini elde ederiz. Kareleri 1 ile işaretlerken, önceki terim nedeniyle karede zaten bir 1 olduğunu görebiliriz. Elimizdeki örnekte, $A'B$ terimi için 011 ve 010 karelerinde 1 vardır, ama 011 karesi ilk $A'C$ terimiyle ortaktır ve işaretli sadece bir adet 1 vardır. 1 ile işaretli beş karenin de gösterdiği gibi örneğimizdeki fonksiyonda beş minterm vardır. Ortadaki dört kare birleştirilerek sadeleştirme yapılır ve C literali elde edilir. 010'daki



Şekil 3-7 $f(x,y,z) = \Sigma(0,2,4,5,6) = z' + xy'$

1 işaretli tek kare, zaten kullanılmış olan bitişikteki bir kareyle bitişiktir. Bu, izin verilebilir, hatta arzu edilir bir durumdur, çünkü iki karenin birleştirilmesi $A'B$ terimini verirken, kareyle temsil edilen tek minterm, üç değişkenli $A'BC'$ terimini verir. Fonksiyon şu şekilde sadeleşir:

$$F = C + A'B$$

ÖRNEK 3.4: Aşağıdaki Boole fonksiyonunu sadeleştirin:

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

Burada mintermler ondalık sayılarıyla verilmiştir. İlgili kareler Şekil 3.7'deki gibi 1'lerle işaretlenmiştir. haritadan bulduğumuz sadeleştirilmiş fonksiyon şöyle olacaktır:

$$F = z' + xy'$$

3.3 DÖRT DEĞİŞKENLİ HARİTA

Dört ikili değişkenli Boole fonksiyonlarının haritası Şekil 3.8'de verilmiştir. (a)'da, 16 minterm ve her biri için tahsis edilen kareler yer almaktadır. (b)'de ise dört değişkenin ilişkisini göstermek için harita yeniden çizilmiştir. Satırlar ve sütunlar, bitişik iki satır veya sütun arasında sadece bir hanenin satır numarasıyla sütun

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)

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

(b)

Şekil 3.8 Dört değişkenli harita

numarası birleştirilerek elde edilebilir. Örneğin üçüncü satırın numarasıyla (11) ikinci sütunun numarası (01) birleştirildiği zaman ondalık 13'ün eşdeğeri ikili 1101 sayısı elde edilir. Bu nedenle üçüncü satır ikinci sütundaki kare, m_{13} mintermini temsil eder.

Dört değişkenli Boole fonksiyonlarının haritayla sadeleştirilmesi, üç değişkenli fonksiyonlardakine benzer. Bitişik kareler, yanyana kareler olarak tanımlanır. Buna ek olarak, haritanın hem üst ve alt kenarların, hem de sağ ve sol kenarların bitişik kareler oluşturacak şekilde birbirine dokunacak şekilde bir yüzey üzerinde bulunduğu varsayılmıştır. Örneğin m_0 ve m_2 bitişik kareler oluşturur, m_3 ve m_{11} de öyle. Sadeleştirme işleminde kullanılan bitişik kareler birleşimi, dört değişkenli haritanın incelenmesiyle kolayca belirlenir:

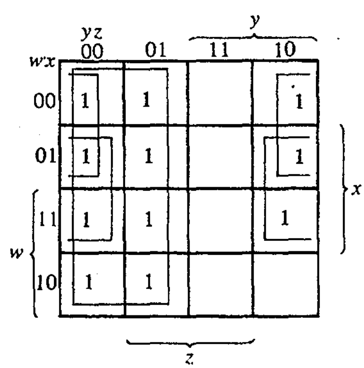
Bir kare bir mintermi temsil eder, dört literalli bir terim verir.
İki bitişik kare üç literalli bir terimi temsil eder.
Dört bitişik kare iki literalli bir terimi temsil eder.
Sekiz bitişik kare bir literalli bir terimi temsil eder.
Onaltı bitişik kare 1'e eşit fonksiyonu temsil eder.

Diğer kare kombinasyonları fonksiyonu sadeleştiremez. Aşağıdaki iki örnek, dört değişkenli Boole fonksiyonlarının sadeleştirilmesinde kullanılan işleme açıklık getirecektir:

ÖRNEK 3.5: Aşağıdaki Boole fonksiyonunu sadeleştirin:

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

Fonksiyonda dört değişken olduğu için, dört değişkenli bir haritanın kullanılması gerekir. Toplamda verilen mintermler, Şekil 3.9'daki haritada 1 ile işaretlenmiştir. 1 işaretli sekiz bitişik kare, bir literalli y' terimi elde edilecek şekilde birleştirilebilir. Sağdaki diğer üç 1, sadeleştirilmiş bir terim verecek şekilde birleştirilemez. Bunların iki veya dört bitişik kare olarak birleştirilmesi gerekir. Birleştirilen kare sayısı ne kadar fazlaysa, terimdeki literal sayısı da o kadar az olur. Örneğimizde sağdaki üst iki 1, soldaki üst iki 1'le birleştirilerek $w'z'$ terimi elde edilir. Aynı karenin birden çok kullanılabileceğine dikkat edin. Geriye üçüncü satır dördüncü sütunda 1 işaretli kare (1110 karesi) kalıyor. Bu kareyi tek başına almak yerine (ki bu dört literalli bir terim verir), bunu dört bitişik kare alanı oluşturmak için



Şekil 3.9 Örnek 3.5 için harita; $F(w,x,y,z) = \Sigma\,(0,1,2,4,5,6,8,9,12,13,14) = y' + w'z' + xz'$

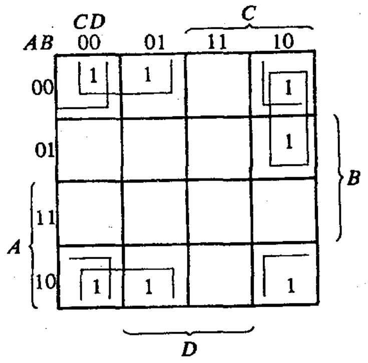
kullanılan karelerle birleştiririz. Bu kareler iki orta sıradan ve sondaki iki sütundan oluşur ve xz' terimini verir. Fonksiyonun sadeleşmiş hali şöyle olacaktır:

$$F = y' + w'z' + xz'$$

ÖRNEK 3.6: Aşağıdaki Boole fonksiyonunu sadeleştirin:

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

Haritada bu fonksiyonun kapladığı alan, Şekil 3.10'da 1'lerle işaretli olan



Şekil 3.10 Örnek 3.6 için harita; $A'B'C' + B'CD' + A'BCD' + AB'C' = B'D' + B'C' + A'CD'$

karelerden oluşur. Bu fonksiyon dört değişkenden oluşur ve ifade edildiği gibi, üçer literalli üç terimden ve dört literalli bir terimden oluşmaktadır. Üç literalli her terim, haritada iki kareyle gösterilmiştir. Örneğin $A'B'C'$, 0000 VE 0001 kareleriyle gösterilmiştir. Fonksiyon, haritada, dört köşedeki 1'ler alınarak sadeleştirilebilir; böylece $B'D'$ terimi elde edilir. Bu mümkündür, çünkü harita üst ve alt veya sol ve sağ kenarlar birbirine temas edilecek şekilde bir yüzey üzerine çizildiği zaman bu dört kare komşudur. Üst sıranın solundaki iki 1, alt sıradaki iki 1'le birleşerek, $B'C'$ terimini oluşturur. Geri kalan 1, iki kareli bir alanla birleştirilerek $A'CD'$ terimi elde edilebilir. Sadeleştirilen fonksiyon şöyle olacaktır:

$$F = B'D' + B'C' + A'CD'$$

3.4. BEŞ VE ALTI DEĞİŞKENLİ HARİTALAR

Dörtten çok değişkeni olan haritaları kullanmak kolay değildir. Karelerin sayısı aşırı ölçüde artar ve komşu (bitişik) kareleri birleştirme işi daha karmaşık hale gelir. Kare sayısı her zaman için minterm sayısına eşittir. Beş değişkenli haritalar için 32 kareye; altı değişkenli haritalar için ise 64 kareye ihtiyaç duyarız. Yedi veya daha fazla değişkenli haritalarda çok fazla kare gerekecektir. Bu tür haritaların kullanımı pratik değildir. Beş ve altı değişkenli haritalar sırasıyla Şekil 3.11 ve 3.12'de verilmiştir. Satır ve sütunlar, yansımalı kod sırasıyla verilmiştir; her bir kareye verilen minterm, bu numaralardan okunur. Bu yolla, beş değişkenli haritada üçüncü satır (11), ikinci sütundaki (001) kare, ondalık 25'in eşdeğeri olan 11001 sayısıdır. Bu nedenle bu kare m_{25} mintermini temsil eder. Yansımalı kod numarasının ilgili bit değerinin 1 olduğu yerlerde her bir değişkenin harf sembolü işaretlenmiştir.

AB \ CDE	000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4
01	8	9	11	10	14	15	13	12
11	24	25	27	26	30	31	29	28
10	16	17	19	18	22	23	21	20

C: 110, 111, 101, 100; B: 01, 11; A: 11, 10; D: 011, 010, 110, 111; E: 001, 011 and 111, 101

Şekil 3-11 5 değişkenli harita

Örneğin beş değişkenli haritada *A* değişkeni, son iki sırada 1'dir; *B* ise ortadaki iki sırada 1'dir. Sütunlardaki yansıyan sayılar, *C* değişkenini en sağdaki dört sütunda 1 ile, *D* değişkenini ise ortadaki dört sütunda 1 ile gösterir; *E* değişkenini durumunda 1'ler, fiziksel olarak bitişik olmayan, ancak ikiye bölünmüş karelerdedir. Altı değişkenli haritada değişken tahsisi benzer bir şekilde belirlenir.

ABC \ DEF	000	001	011	010	110	111	101	100
000	0	1	3	2	6	7	5	4
001	8	9	11	10	14	15	13	12
011	24	25	27	26	30	31	29	28
010	16	17	19	18	22	23	21	20
110	48	49	51	50	54	55	53	52
111	56	57	59	58	62	63	61	60
101	40	41	43	42	46	47	45	44
100	32	33	35	34	38	39	37	36

D: 110, 111, 101, 100; C: 001, 011 and 111, 101; B: 011, 010, 110, 111; A: 110, 111, 101, 100; E: 011, 010, 110, 111; F: 001, 011 and 111, 101

Şekil 3-12 6 değişkenli harita

Bazı değişkenlerin iki parçaya bölündüğü gerçeğini dikkate almak için Şekil 3.11 ve 3.12'deki haritalar için bitişik kare tanımında değişiklik yapmak gerekir. Beş değişkenli haritanın, dört değişkenli iki adet haritadan, altı değişkenli haritanın ise dört değişkenli dört haritadan oluştuğu düşünülmelidir. Bu dört değişkenli haritalardan her birisi, haritanın ortasındaki çift çizgiyle ayırdedilebilir; ayrı ayrı ele alındığında her biri daha önce tanımlanan bitişikliği korur. Buna ek olarak, ortadaki çift çizginin, bir kitabın ortası olarak değerlendirilmesi gerekir; haritanın her bir yarısı bir sayfaya karşılık gelir. Kitap kapatıldığı zaman bitişik kareler birbirinin üstüne gelecektir. Başka bir deyişle, ortadaki çift çizgi bir aynaya benzer; her bitişik kare, sadece komşu dört kareye değil, ayrıca aynadaki görüntüsüne de bitişiktir. Örneğin beş değişkenli haritadaki 31 nolu minterm, 30, 15, 29, 23 *ve* 27 nolu mintermlere bitişiktir. Altı değişkenli haritada ise aynı minterm bu mintermlerin tamamının yanısıra 63 nolu mintermle de bitişiktir.

Haritayı inceler ve bitişik kareler için verilen yeni tanımı dikkate alırsak, n değişkenli bir haritadaki $k = 0, 1, 2, ...n$ için 2^k sayısı kadar olan bitişik karelerden her birisinin, $n - k$ literalli bir terimi veren bir alanı temsil ettiği gösterilebilir. Bu ifadenin anlamlı olması için n sayısının k'dan büyük olması gerekir. $n = k$ durumunda harita alanının tamamı birleşerek birim eleman fonksiyonu verir. Tablo 3.1, bitişik kare sayısıyla terimdeki literal sayısı arasındaki ilişkiyi göstermektedir.

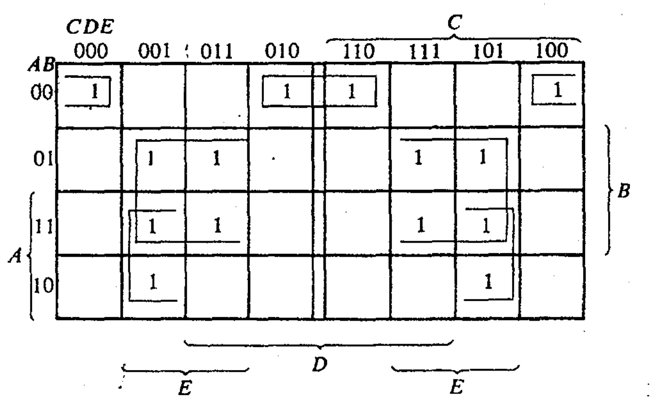
Tablo 3.1 Bitişik kare sayısıyla terimdeki literal sayısı arasındaki ilişki

	Bitişik kare sayısı	n- değişkenli haritada bir terimdeki literal sayısı					
k	2^k	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$
0	1	2	3	4	5	6	7
1	2	1	2	3	4	5	6
2	4	0	1	2	3	4	5
3	8		0	1	2	3	4
4	16			0	1	2	3
5	32				0	1	2
6	64					0	1

Örneğin beş değişkenli bir haritadaki bir alanda birleşen sekiz bitişik kare, iki literalli bir terim verir.

ÖRNEK 3.7: Aşağıdaki Boole fonksiyonunu sadeleştirin:

$$F(A,B,C,D,E) = \Sigma(0,2,4,6,9,11,13,15,17,21,25,27,29,31)$$



Şekil 3-13 Örnek 7 için harita;
$F(A,B,C,D,E) = \Sigma(0,2,4,6,9,11,13,15,17,21,25,27,29,31) = BE + AD'E + A'B'E'$

Bu fonksiyonun beş değişkenli haritası Şekil 3.13'te verilmiştir. Her minterm, eşdeğer ikili sayıya çevrilmiş ve ilgili karelerinde 1 ile işaretlenmiştir. Bu noktada, olası en büyük alanı verecek bitişik kareler kombinasyonunu bulmamız gerekiyor. Haritanın sağ yarısının ortasındaki dört kare, çift çizgi boyunca yansıtılır ve haritanın sol yarısının ortasındaki dört kareyle birleştirilir, böylece BE terimine eşdeğer olan sekiz bitişik kare elde edilir. Alt sıradaki iki adet 1, merkezdeki çift çizgi üzerinden birbirinin yansımasıdır. Bunların diğer bitişik iki kareyle birleştirilmesi, $AD'E$ terimini verir. Üst sıradaki dört adet 1 bitişiktir ve birleştirilerek $A'B'E'$ terimi elde edilir. Böylece bütün 1'ler dahil edilmiş olur. Sadeleştirilmiş fonksiyon aşağıdaki gibi olacaktır:

$$F = BE + AD'E + A'B'E'$$

3.5 TOPLAMLARIN ÇARPIMIYLA SADELEŞTİRME

Önceki bütün örneklerde, haritadan elde edilen sadeleştirilmiş Boole fonksiyonları, çarpımların toplamı formunda ifade ediliyordu. Küçük bir değişiklik yapmak suretiyle toplamların çarpımı formu elde edilebilir.

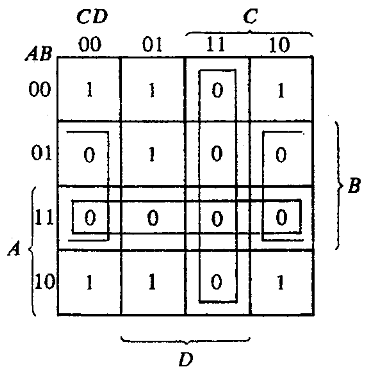
Toplamların çarpımı formunda sadeleştirilmiş (en aza indirilmiş) bir fonksiyon elde etme yöntemi, Boole fonksiyonların temel özelliklerine dayanmaktadır. haritanın karelerinde bulunan 1'ler, fonksiyonun mintermlerini temsil eder. Fonksiyona dahil edilmeyen mintermler, fonksiyonun tümleyenini gösterir. Bundan, bir fonksiyonun tümleyeninin, haritada 1 ile işaretli olmayan kareler ile temsil edildiğini görürüz. Boş kareleri 0 ile işaretleyip geçerli bitişik karelerde birleştirirsek, fonksiyonun tümleyeninin (F') sadeleştirilmiş bir ifadesini elde ederiz. F' tümleyeni bize gene orijinal F fonksiyonunu verir. Genelleştirilen DeMorgan teoremi nedeniyle bu şekilde elde edilen fonksiyon, otomatik olarak toplamların çarpımı şeklindedir. Bunu göstermenin en iyi yolu bir örnek vermektir:

ÖRNEK 3.8: Aşağıdaki Boole fonksiyonunu (a) çarpımların toplamı ve (b) toplamların çarpımı şeklinde sadeleştirin:

$$F(A,B,C,D) = \Sigma\,(0,1,2,5,8,9,10)$$

Şek. 3.14'teki haritada 1 ile işaretli kareler, fonksiyonun tüm mintermlerini temsil eder. 0 ile işaretli kareler ise F fonksiyonunda bulunmayan mintermleri ve bu nedenle F fonksiyonunun tümleyenini gösterir. 1 bulunan karelerin birleştirilmesi, sadeleştirilen fonksiyonu, çarpımların toplamı olarak verir:

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



Şekil 3.14 Örnek 3.8 için harita;
$F(A,B,C,D) = \Sigma\,(0,1,2,5,8,9,10) = B'D' + B'C' + A'C'D = (A' + B')(C' + D')(B' + D)$

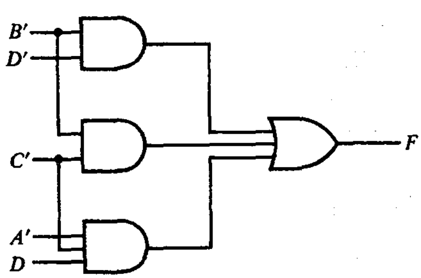
Şemada gösterildiği gibi 0 işaretli kareleri birleştirdiğimiz zaman ise, sadeleştirilmiş tümleyen fonksiyonunu elde ederiz:

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

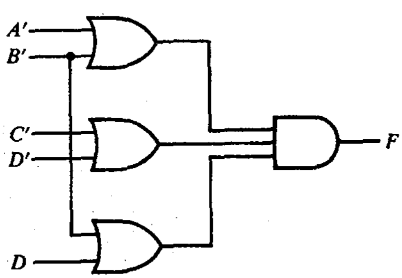
DeMorgan teoremini uygularsak (dualini alıp her literali 2.4. Bölümde anlatıldığı şekilde tümleyerek), sadeleştirilen fonksiyonu toplamların çarpımı şeklinde elde ederiz:

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

3.8. Örnekte elde edilen sadeleştirilmiş ifadelerin uygulaması Şekil 3.15'te verilmiştir. Çarpımların toplamı ifadesi her VE terimi için bir adet olmak üzere bir grup VE kapısıyla (a)'da uygulanmıştır. VE kapılarının çıkışları, tek bir VEYA kapısının girişlerine bağlanmıştır. Aynı fonksiyon, (b)'de ise her VEYA terimi için bir tane olmak üzere bir grup VEYA kapısıyla toplamların çarpımı şeklinde uygulanmıştır. VEYA kapılarının çıkışları, tek bir VE kapısının girişlerine bağlanmıştır. Her bir durumda, giriş değişkenlerinin tümleyeniyle doğrudan mevcut olduğu, bu nedenle tersleyiciye ihtiyaç olmadığı varsayılmıştır. Şekil 3.15'te verilen düzenleme deseni, standard formlardan birisiyle ifade edilerek uygulanan bir Boole fonksiyonunun genel halidir. Çarpımların toplamında, VE kapıları tek bir VEYA kapısına bağlanır; toplamların çarpımında ise, VEYA kapıları tek bir VE kapısına bağlanır. Her iki düzenlemede de da kapılar için iki düzey oluşturur. Bu nedenle bir fonksiyonun standard formda uygulanmasına, iki-düzeyli uygulama denir.



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



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

Şekil 3.15 Örnek 3.8'deki fonksiyonun kapı uygulaması

Tablo 3.2 *F* fonksiyonunun doğruluk tablosu

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

Orijinal fonksiyonun mintermler toplamı kanonik formunda ifade edildiği durumlarda, toplamların çarpımıyla sadeleştirme yöntemi, 3.8. Örnekte gösterilmiştir. Bu yöntem, orijinal fonksiyonun maksterm kanonik form çarpımı olarak ifade edildiği durumlarda da geçerlidir. Örneğin Tablo 3.2'de F fonksiyonunu tanımlayan doğruluk tablosunu ele alın. Mintermlerin toplamında bu fonksiyon şu şekilde ifade edilir:

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

Makstermler çarpımında ise şu şekilde ifade edilir:

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

Başka bir deyişle fonksiyonun 1'leri mintermleri, 0'ları ise makstermleri temsil eder. Bu fonksiyonun haritası, Şekil 3.16'da verilmiştir. Sadeleştirmeye, ilk önce fonksiyonun 1 olduğu her mintermi 1 ile işaretlemek suretiyle başlanabilir. Geri kalan kareler 0 ile işaretlenir. öte yandan başlangıçta makstermler çarpımının

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

Şekil 3-16 Tablo 3-2'nin fonksiyonu için harita

verilmiş olması halinde fonksiyonda verilen karelerdeki 0'lar işaretlenerek başlanabilir; geri kalan kareler de 1 ile işaretlenir. 1'ler ve 0'lar işaretlendikten sonra fonksiyon, standard formlardan herhangi birisiyle sadeleştirilebilir. Çarpımların toplamı için 1'leri birleştirerek aşağıdaki ifadeyi elde ederiz:

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

Toplamların çarpımı için ise 0'ları birleştirerek sadeleştirilmiş tümleyen fonksiyonu elde ederiz:

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

bu da özel VEYA fonksiyonunun, eşdeğerlik fonksiyonunun tümleyeni olduğunu gösterir (Bölüm 2.6). F' fonksiyonunun tümleyenini alırsak, sadeleştirilen fonksiyonu toplamların çarpımı olarak elde ederiz:

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

Toplamların çarpımıyla ifade edilen bir fonksiyonu haritaya girmek için, fonksiyonun tümleyenini alın ve 0'la işaretlenecek kareleri bulun. Örneğin:

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

fonksiyonu ilk önce tümleyeni alınarak haritaya girilebilir:

$$F' = ABC' + B'D'$$

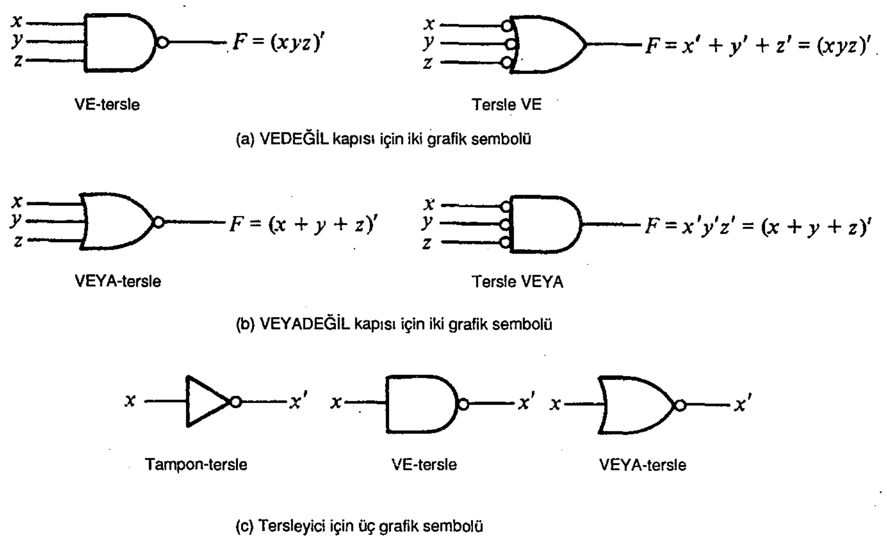
ve daha sonra F' mintermlerini temsil eden kareler 0'la işaretlenebilir. Geri kalan kareler 1'le işaretlenir.

3.6 VEDEĞİL VE VEYADEĞİL UYGULAMASI

Sayısal devreler; VEDEĞİL ya da VEYADEĞİL kapılarıyla, VE ile VEYA kapılarıyla olandan daha sık tasarlanır. VEDEĞİL ile VEYADEĞİL kapılarının elektronik bileşenlerle üretilmesi daha kolaydır ve bütün IC sayısal mantık ailelerinde kullanılan temel kapılardır. Sayısal devrelerin tasarımında VEDEĞİL ve VEYADEĞİL kapılarının taşıdığı temel önemden ötürü, VE, VEYA ve DEĞİL terimleriyle verilen Boole fonksiyonlarını eşdeğer VEDEĞİL ya da VEYADEĞİL mantık şemalarına dönüştürmek için bazı kural ve yöntemler geliştirilmiştir. Bu bölümde iki-düzeyli uygulama yöntemi anlatılacak. Çok seviyeli uygulama ise 4.7. Bölümde tartışılacaktır.

VEDEĞİL ve VEYADEĞİL mantığına çevrimi kolaylaştırmak için, bu kapılar için ek iki grafik sembolü tanımlamak yerinde olacaktır. VEDEĞİL kapısının iki eşdeğer sembolü Şekil 3.17(a)'da verilmiştir. Daha önce tanımlanmış olan VE-tersleyici sembolü, bir VE grafik sembolünden ve bunu izleyen küçük bir daireden oluşur. Bunun yerine VEDEĞİL kapısını, girişlerinin önüne küçük daireler koyarak bir VEYA grafik sembolüyle göstermek de mümkündür. VEDEĞİL kapısı için tersle-VEYA sembolü, DeMorgan teoremine dayanır ve küçük dairelerin tümleyene karşılık geldiği kuralını oluşturur.

Aynı şekilde, Şekil 3.17(b)'de de gösterildiği gibi, VEYADEĞİL kapısı için iki grafik fonksiyonu vardır. VEYA-tersle, geleneksel semboldür. Tersle-VE ise



Şekil 3-17 VE DEĞİL ve VEYA DEĞİL Kapıları için grafik semboller.

DeMorgan teoreminden yararlanan uygun bir seçenektir; bu ise girişlerdeki küçük dairelerin tümleyene karşılık geldiği yolundaki ikinci kuralımızı oluşturur.

Tek girişli bir VEDEĞİL ya da VEYADEĞİL kapısı bir tersleyici gibi davranır. Sonuçta tersleyici bir kapı Şekil 3.17(c)'de gösterildiği gibi üç farklı yoldan çizilebilir. Bütün tersleyici sembollerindeki küçük daireler, kapının mantığı değiştirilmeksizin giriş ucuna aktarılabilir.

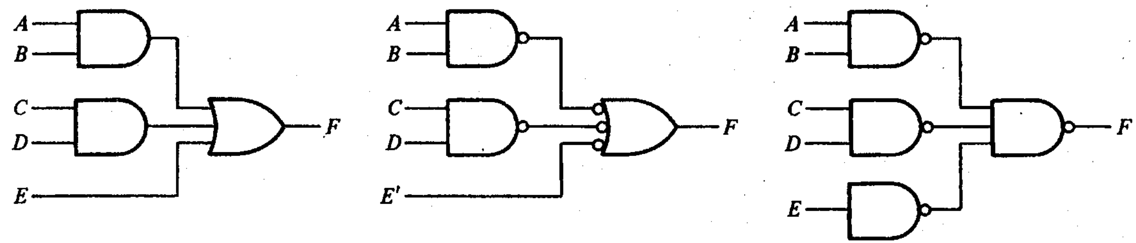
VEDEĞİL ya da VEYA kapıları için öngörülen alternatif sembollerin, bütün giriş uçlarında daire yerine üçük üçgenlerle çizilebileceğini belirtmek gerek. Küçük bir üçgen, negatif-mantık polarite göstergesidir (bkz. 2.8. Bl. ve Şekil 2.11). Giriş uçlarına küçük üçgenler konduğu zaman grafik sembolü girişlerin negatif-mantık polaritesi olduğunu gösterir, ancak kapı çıkışının (üçgensiz) pozitif bir mantık düzenlemesi olacaktır. Bu kitapta başından sonuna pozitif mantıkta kalmayı ve gerektiği durumlarda tümleyeni göstermek için küçük daire kullanmayı tercih ediyoruz.

VEDEĞİL Uygulaması

VEDEĞİL kapılı bir Boole fonksiyonunun uygulanması, fonksiyonun çarpımların toplamı şeklinde sadeleştirilmesini gerektirir. Çarpımların toplamı ifadesi ile eşdeğer VEDEĞİL uygulaması arasındaki ilişkiyi görmek için, Şekil 3.18'deki mantık şemalarını ele alın. Şemaların üçü de eşdeğerdir ve şu fonksiyonu uygular:

$$F = AB + CD + E$$

Fonksiyon, VE ve VEYA kapıları kullanılarak çarpımların toplamı halinde Şekil



(a) (b) (c)

Şekil 3-18 $F = AB + CD + E$ fonksiyonunu uygulamanın üç yolu

3.18(a)'da uygulanmıştır. Şek. 3.18(b)'de ise VE kapılarının yerine VEDEĞİL ve VEYA kapısının yerine tersle-VEYA sembollü bir VEDEĞİL kullanılmıştır. *E* değişkeninin tümleyeni alınmış ve ikinci-seviyeden-tersle-VEYA kapısına uygulanmıştır. Küçük daire sembolünün tümleyeni gösterdiğini unutmayın. Bu nedenle aynı çizgideki iki daire çifte tümleyeni gösterir ve bu nedenle her ikisi de atılabilir. *E* tümleyeni, bir değişkeni tümleyerek *E*'nin normal değerini tekrar üretir. Şekil 3.18(b)'deki küçük dairelerin atılmasıyla, (a)'daki devre elde edilir. Dolayısıyla iki şema aynı fonksiyonu uygulamaktadır ve eşdeğerdir.

VEDEĞİL kapısı Şekil 3.18(c)'de geleneksel sembolüyle tekrar çizilmiştir. Tek girişli VEDEĞİL kapısı, *E* değişkeninin tümleyenini alır. Bu tersleyiciyi iptal edip E' değerini doğrudan doğruya ikinci-seviye VEDEĞİL kapısının girişine uygulamak mümkündür. (c)'deki şema (b)'dekine, o da (a)'dakine eşdeğerdir. VE ve VEYA kapılarının yerine VEDEĞİL kapıları konmuştur, ancak tek *E* değişkeniyle ek bir VEDEĞİL kapısı daha konulmuştur. VEDEĞİL mantık şemaları çizilirken (b) veya (c)'deki devre kabul edilebilir. Ancak (b)'deki devre, uygulanan Boole ifadesiyle daha dolaysız bir ilişkiye sahiptir.

Şekil 3.18(c)'deki VEDEĞİL uygulaması cebirsel olarak doğrulanabilir. Uygulanan VEDEĞİL fonksiyonu, DeMorgan teoremi kullanılarak çarpımların toplamı formuna kolayca çevrilebilir:

$$F = [\,(AB)' \cdot (CD)' \cdot E'\,]' = AB + CD + E$$

Şekil 3.18'deki dönüştürmeden, bir Boole fonksiyonunun iki seviyeli VEDEĞİL kapılarıyla uygulanabileceği sonucunu çıkarıyoruz. Boole fonksiyonundan VEDEĞİL mantık şeması elde etme kuralı aşağıda verilmiştir:

1. Fonksiyonu sadeleştirin ve çarpımların toplamı halinde ifade edin.

2. Fonksiyonun en az iki literale sahip her çarpım terimi için bir VEDEĞİL kapısı çizin. Her bir VEDEĞİL kapısına yapılan girişler, terimin literalleridir. Bu, birinci seviye kapıları grubunu oluşturur.

3. Girişleri birinci seviye kapılarının çıkışlarından gelecek şekilde, ikinci seviyede tek bir VEDEĞİL kapısı çizin (VE-tersle ya da tersle-VEYA grafik sembolünü kullanarak).

4. Tek literalli bir terim, birinci seviyede tersleyici gerektirir, ya da tümleyeni alınarak ikinci seviye VEDEĞİL kapısına giriş olarak uygulanabilir.

Bu kuralları belli bir örneğe uygulamadan önce, Boole fonksiyonlarını VEDEĞİL kapılarıyla uygulamanın ikinci bir yolunun daha olduğunu belirtmek gerek. Bir haritadaki 0'ları birleştirdiğimiz zaman çarpımların toplamıyla ifade edilen fonksiyon tümleyeninin sadeleştirilmiş ifadesini elde ettiğimizi hatırlayın. Daha sonra fonksiyonun tümleyeni, yukarıda belirtilen kurallarla iki seviyeli VEDEĞİL kapılarıyla uygulanabilir. Normal çıkış istendiği taktirde çıkış değişkeninin gerçek değerini elde etmek için tek girişli VEDEĞİL veya tersleme kapısı eklemek gerekecektir. Tasarımcının, fonksiyonun tümleyenini elde etmek isteyeceği durumlar olacaktır; bu durumda bu ikinci yöntem tercih edilebilir.

ÖRNEK 3.9: Aşağıdaki fonksiyonu VEDEĞİL kapılarıyla uygulayın:

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

Yapılacak ilk iş, fonksiyonu çarpımların toplamı formunda sadeleştirmek olacaktır. Şekil 3.19(a)'da bu yapılmıştır. Haritada sadece iki adet 1 vardır ve birleştirilemez. Örneğimizde çarpımların toplamı halindeki sadeleştirilmiş fonksiyon şu şekilde olacaktır:

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

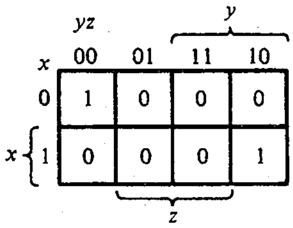
İki seviyeli VEDEĞİL uygulaması Şekil 3.19(b)'de verilmiştir. Daha sonra fonksiyonun tümleyenini, çarpımların toplamı halinde sadeleştirmeye çalışırız. Bu, haritadaki 0'lar birleştirilerek yapılır:

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

F' elde etmek için kullanılan iki seviyeli VEDEĞİL kapısı Şekil 3.19(c)'de verilmiştir. F çıkışı istendiği taktirde fonksiyonu evirmek (tersini almak) için tek girişli bir VEDEĞİL kapısı eklemek gerekecektir. Bu da üç seviyeli bir uygulama demektir. Her bir durumda, giriş değişkenlerinin hem normal hem tümleyen şeklinde mevcut olduğu varsayılmıştır. Sadece bir şekilde mevcut olmaları halinde girişlere evirici eklemek gerekecekti, bu da devrelere ek bir seviyenin daha uygulanması anlamına gelirdi. Girişin z' olarak değiştirilmesi koşuluyla tek z değişkeniyle ilgili tek girişli VEDEĞİL kapısı iptal edilebilir.

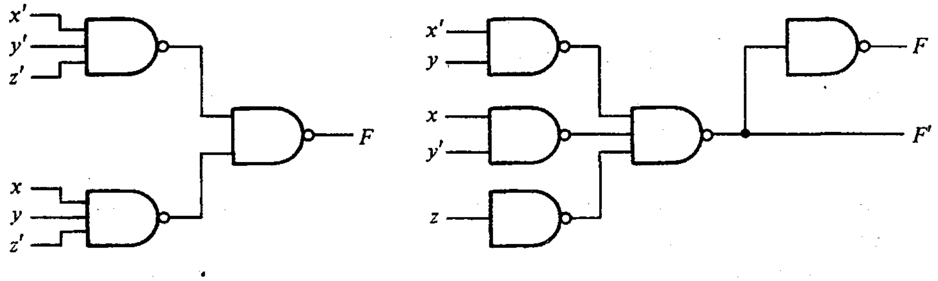
VEYADEĞİL Uygulaması

VEYADEĞİL fonksiyonu, VEDEĞİL fonksiyonunun dualidir. Bu nedenle VEYADEĞİL mantığı için gerekli bütün yöntem ve kurallar, VEDEĞİL mantığı için geliştirilen ilgili yöntem ve kuralların dualidir.



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

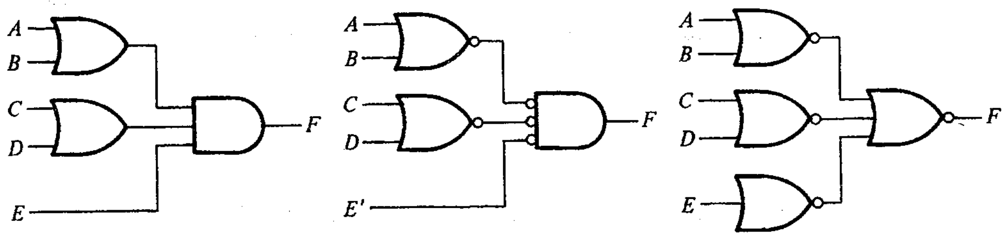
(a) Çarpımların toplamıyla harita sadeleştirme



(b) $F = x'y'z' + xyz'$ (c) $F' = x'y + xy' + z$

Şekil 3.19 3.9. Örnekteki fonksiyonun VEDEĞİL kapılarıyla uygulanması

Bir Boole fonksiyonunun VEYADEĞİL kapılarıyla uygulanması, fonksiyonun toplamların çarpımı formunda sadeleştirilmesini gerektirir. Toplamların çarpımı ifadesi, toplam terimleri için bir VEYADEĞİL kapıları grubunu, daha sonra da çarpımı oluşturan bir VE kapısını belirler. VEYA-VE'den VEYADEĞİL-VEYADEĞİL (NOR-NOR) şemasına dönüştürme işlemi Şekil 3.20'de verilmiştir. İşlem, burada toplamların çarpımı ifadesini kullanmamız dışında, daha önce tartıştığımız VEDEĞİL dönüştürmesine benzer:



(a) (b) (c)

Şekil 3.20 $F = (A + B)(C + D)E$ fonksiyonunu uygulamanın üç yolu.

$$F = (A + B)(C + D)E$$

Boole fonksiyonundan VEYADEĞİL mantık şeması elde etme kuralı bu dönüştürmeden çıkarılabilir. Bu, üç aşamalı VEDEĞİL kuralına benzer, ancak burada sadeleştirilen ifadenin toplamların çarpımı halinde olması ve birinci seviye VEYADEĞİL kapıları terimlerinin, toplam terimler olması gerekir. Tek literalli bir terim, tek girişli bir VEYADEĞİL veya evirme kapısı gerektirir; ayrıca tümleyeni alınarak doğrudan doğruya ikinci seviye VEYADEĞİL kapısına da uygulanabilir.

Bir fonksiyonu VEYADEĞİL kapısıyla uygulamanın ikinci bir yolu da, toplamların çarpımı halindeki fonksiyon tümleyeni ifadesini kullanmaktır. Bu, F' için iki seviyeli bir uygulama, normal F çıkışının istenmesi halinde ise üç seviyeli bir uygulama sağlayacaktır.

Haritadan sadeleştirilmiş toplamların çarpımını elde etmek için haritadaki 0'ları birleştirip daha sonra fonksiyonun tümleyenini almak gerekecektir. Fonksiyonun tümleyeni için sadeleştirilmiş toplamların çarpımı ifadesini elde etmek için haritadaki 1'leri birleştirip daha sonra fonksiyonun tümleyenini almak gerekir. Aşağıdaki örnek, VEYADEĞİL uygulaması yöntemine açıklık getirecektir:

ÖRNEK 3.10: Örnek 3.9'daki fonksiyonu VEYADEĞİL kapılarıyla uygulayın.

Bu fonksiyonun haritası Şekil 3.19(a)'da verilmiştir. İlk önce haritadaki 0'lar birleştirilerek:

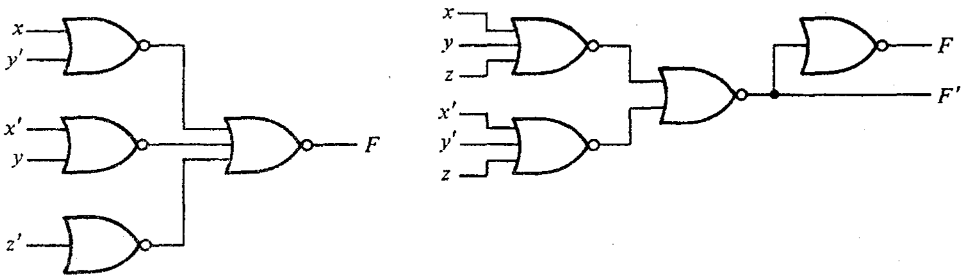
$$F' = x'y + xy' + z$$

elde edilir. Bu, çarpımların toplamı halindeki fonksiyon tümleyenidir. VEYADEĞİL uygulaması için gerekli toplamların çarpımı halindeki sadeleştirilmiş fonksiyonu elde etmek için F' fonksiyonunun tümleyenini alın:

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

VEYADEĞİL kapılı iki seviyeli uygulama Şekil 3.21(a)'da verilmiştir. Tek z' literalli terim, tek girişli bir VEYADEĞİL veya tersleme kapısı gerektirir. Bu kapı iptal edilebilir ve z girişi doğrudan doğruya ikinci seviyeden VEYADEĞİL kapısının girişine uygulanabilir.

Fonksiyonun toplamların çarpımları halindeki tümleyeniyle ikinci bir uygulama da mümkündür. Bu durumda ilk önce haritadaki 1'ler



(a) $F = (x + y')(x' + y)z'$ (b) $F' = (x + y + z)(x' + y' + z)$

Şekil 3.21 VEYADEĞİL kapılarıyla uygulama

Tablo 3.3 VEDEĞİL ve VEYADEĞİL uygulaması için kurallar

Durum	Sadeleştirilecek Fonksiyon	Kullanılacak .Standart Form	Türetilişi	Uygulanışı	F için düzey sayısı
(a)	F	Çarpımların toplamı	Haritada 1'leri birleştir	VEDEĞİL	2
(b)	F'	Çarpımların toplamı	Haritada 0'ları birleştir	VEDEĞİL	3
(c)	F	Toplamların çarpımı	(b)'deki F' tümleyeni	VEYADEĞİL	2
(d)	F'	Toplamların çarpımı	(a)'daki F tümleyeni	VEYADEĞİL	3

birleştirilerek:

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

elde edilir. Bu, sadeleştirilmiş çarpımların toplamı ifadesidir. Fonksiyonun, VEYADEĞİL uygulaması için gerekli toplamların çarpımı şeklindeki tümleyenini elde etmek için bu fonksiyonun tümleyenini alın:

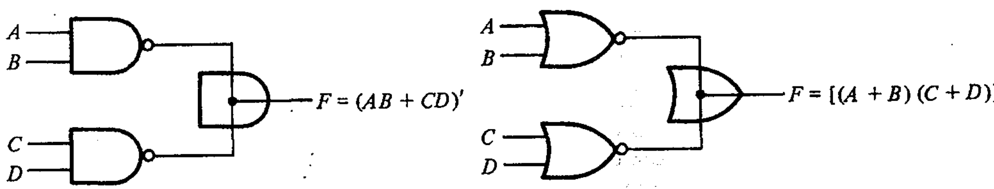
$$F' = (x + y + z)(x' + y' + z)$$

F' için iki seviyeli uygulama Şekil 3.21(b)'de verilmiştir. F çıkışı istendiği takdirde, üçüncü seviyeden bir tersleyici ile elde edilebilir.

VEDEĞİL ve VEYADEĞİL uygulama yöntemlerinin bir özeti, Tablo 3.3'te verilmiştir. Uygulamadaki kapı sayısını azaltmak amacıyla her zaman için fonksiyonu sadeleştirmeyi unutmamak gerek. Harita sadeleştirme yöntemlerinden elde edilen standard formlar, doğrudan geçerlidir ve VEYADEĞİL ve VEDEĞİL mantığıyla çalışılırken çok yararlı olmaktadır.

3.7 DİĞER İKİ SEVİYELİ UYGULAMALAR

Entegre devrelerde en sık rastlanan kapılar, VEYADEĞİL ve VEDEĞİL kapılarıdır. Bu nedenle pratik bakış açısından VEDEĞİL ve VEYADEĞİL mantık ile uygulamaları önem bakımından ilk sırayı alır. Bazı VEDEĞİL ve VEYADEĞİL kapıları (ancak hepsi değil), belli bir mantık fonksiyonu elde etmek için iki kapının çıkışı arasında bir tel bağlanmasını mümkün kılar. Bu mantık türüne *telli mantık* denir. Örneğin açık-kolektörlü TTL VEDEĞİL kapıları, birbirine bağlandıkları zaman, kablolu VE mantığı işi görürler. (Açık-kolektör TTL kapısı Bölüm 10, Şekil 10.11'de verilmiştir.) İki VEYADEĞİL kapısıyla uygulanan telli-VE mantık, Şekil 3.22(a)'da verilmiştir. Geleneksel kapılardan ayırdetmek için VE



(a) Açık-kolektörlü TTL VEDEĞİL kapılarındaki telli-VE

(VE-VEYA-TERSLE)

(b) ELC kapılarında telli-VEYA

(VEYA-VE-TERSLE)

Şekil 3.22 Telli Mantık

kapısı, kapının ortasından geçen çizgilerle çizilmiştir. Telli VE kapısı, fiziksel bir kapı değildir, sadece, gösterilen telli bağlantısıyla elde edilen fonksiyonu gösteren bir semboldür. Şekil 3.22(a)'daki devreyle uygulanan mantık fonksiyonu:

$$F = (AB)' \cdot (CD)' = (AB + CD)'$$

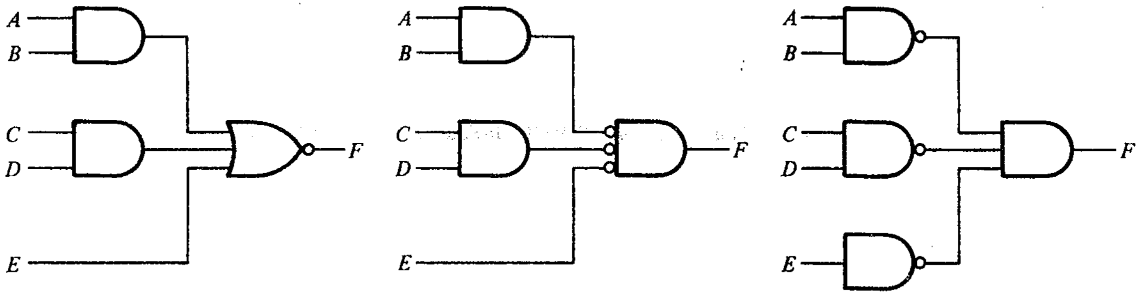
Buna VE-VEYA-TERSLE fonksiyonu denilmektedir.

Benzer bir şekilde, ECL kapılarının VEYADEĞİL çıkışları, telli-VEYA fonksiyonunu uygulamak için birbirine bağlanabilir. Şekil 3.22(b)'deki devreyle uygulanan mantık fonksiyonu:

$$F = (A + B)' + (C + D)' = [(A + B)(C + D)]'$$

Buna VEYA-VE-TERSLE fonksiyonu denilmektedir.

Telli kapı fiziksel bir ikinci seviye kapısı yaratmaz, çünkü kablo bağlantısından başka bir şey değildir. Yine de tartışma amacıyla Şekil 3.22'deki devreleri iki seviyeli uygulamalar olarak değerlendireceğiz. Birinci seviye

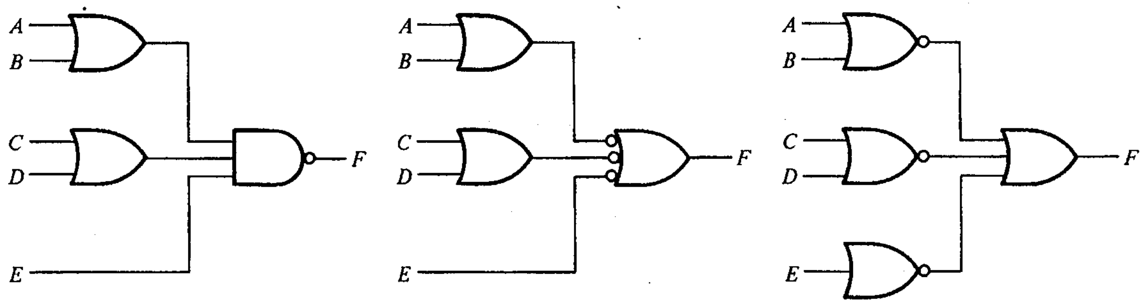


(a) VE-VEYADEĞİL

(b) VE-VEYADEĞİL

(c) VEYADEĞİL-VE

Şekil 3-23 VE-VEYA-TERSLE Devreleri; $F = (AB + CD + E)'$



(a) VEYA-VEDEĞİL (b) VEYA-VEDEĞİL (c) VEYADEĞİL-VEYA

Şekil 3-24 VEYA-VE-TERSLE Devreleri; $F = [(A+B)(C+D)E]'$

VEDEĞİL (ya da VEYADEĞİL) kapılarından oluşur, buna karşılık ikinci seviyede tek bir VE (ya da VEYA) kapısı vardır. Aşağıdaki tartışmada grafik sembolündeki tel bağlantısı gözardı edilecektir.

Bozuk Olmayan Formlar

Teorik bir bakış açısından, kaç tane iki seviyeli kapı birleşiminin mümkün olduğunu belirlemek öğretici olacaktır. Dört tip kapı inceliyoruz: VE, VEYA, VEDEĞİL ve VEYADEĞİL. Kapı tiplerinden birini bir seviyeye, diğerini ikinci seviyeye uygularsak, iki seviyeli formlar için 16 olası birleşim olduğunu görürüz. (VEDEĞİL-VEDEĞİL uygulamasında olduğu gibi, aynı kapı tipi hem birinci hem de ikinci seviyede olabilir.) Tek bir işleme indirgenmeleri nedeniyle bu birleşimlerden sekizine *bozuk f*ormlar denilmektedir. Bu, birinci ve ikinci seviyelerde VE kapısı olan bir devrede görülebilir. Devrenin çıkışı bütün giriş değişkenleri için VE fonksiyonu olmaktan öte bir şey değildir. Geri kalan sekiz adet *bozuk olmayan* form, çarpımların toplamı veya toplamların çarpımı şeklinde bir uygulama sağlar. Sekiz bozuk olmayan form şunlardır:

VE-VEYA	VEYA-VE
VEDEĞİL-VEDEĞİL	VEYADEĞİL-VEYADEĞİL
VEYADEĞİL-VEYA	VEDEĞİL-VE
VEYA-VEDEĞİL	VE-VEYADEĞİL

Listedeki her bir formun ilk kapısı, birinci seviyeden uygulama sağlar. İkinci kapı ise ikinci seviyeye konulan tek bir kapıdır. Aynı satırda verilen iki formun birbirinin duali olduğunu unutmayın.

VE-VEYA ile VEYA-VE formları, 3.5. Bölümde tartışılan temel iki seviyeli formlardır. VEDEĞİL-VEDEĞİL ve VEYADEĞİL-VEYADEĞİL ise 3.6. Bölümde anlatılmıştı. Bu bölümde ise geri kalan dört formu inceleyeceğiz.

VE-VEYA-TERSLE Uygulaması

VEDEĞİL-VE ile VE-VEYADEĞİL formları eşdeğerdir ve birlikte ele alınabilir. Her ikisi de Şekil 3.23'te gösterildiği gibi VE-VEYA-TERSLE fonksiyonunu uygular. VE-VEYADEĞİL formu, VEYADEĞİL kapısının çıkışındaki küçük daireyle yapılan terslemeli bir VE-VEYA formuna benzer. Bu, şu fonksiyonu uygular:

$$F = (AB + CD + E)'$$

VEYADEĞİL kapısı için alternatif grafik sembolünü kullanarak Şekil 3.23(b)'deki şemayı elde ederiz. Tek E değişkeninin tümleyeninin *alınmadığına* dikkat edin, çünkü yapılan tek değişiklik VEYADEĞİL kapısının grafik sembolündedir. Şimdi daireleri ikinci seviye kapısının giriş ucundan alıp birinci seviyeden kapıların çıkış uçlarına taşıyalım. Alternatif olarak, E girişinin tümleyeninin alınması koşuluyla tersleyici de iptal edilebilir. Şekil 3.23(c)'deki devre, bir VEDEĞİL-VE formudur ve VE-VEYA-TERSLE fonksiyonunu uygulaması bağlamında Şekil 3.22'de gösterilmişti.

VE-VEYA uygulaması, çarpımların toplamı şeklindeki bir ifadeyi gerektirir. Tersleme sayılmazsa, VE-VEYA-TERSLE ile gerçekleştirmede buna benzer. Dolayısıyla fonksiyonun *tümleyeni* çarpımların toplamı halinde sadeleştirildiği taktirde (haritadaki 0'ları birleştirerek), F' fonksiyonunu, fonksiyonun VE-VEYA kısmıyla uygulamak mümkün olacaktır. F' her zaman mevcut olan çıkış terslemesinden (TERSLE kısmından) geçtiği zaman fonksiyonun F çıkışını üretecektir. Daha sonra VE-VEYA-TERSLE uygulamasına bir örnek verilecektir.

VEYA-VE-TERSLE Uygulaması

VEYA-VEDEĞİL ve VEYADEĞİL-VEYA formları, VEYA-VE-EVİR fonksiyonunu uygular. Bu, Şekil 3.24'te gösterilmiştir. VEYA-VEDEĞİL formu, VEDEĞİL kapısındaki dairenin yaptığı tersleme dışında, VEYA-VE formuna benzer ve aşağıdaki fonksiyonu uygular:

$$F = [(A + B)(C + D)E]'$$

VEDEĞİL kapısı için alternatif grafik sembolünü kullanarak Şek. 3.24(b)'deki şemayı elde ederiz. (c)'deki devre, küçük daireler ikinci seviyeden kapının girişlerinden birinci seviyeden kapıların çıkışlarına taşınarak elde edilir. Şekil 3.24(c)'deki devre bir VEYADEĞİL-VEYA formudur ve VEYA-VE-TERSLE fonksiyonunu uygulanması bağlamında Şekil 3.22'de gösterilmiştir.

VE-VEYA-TERSLE uygulaması, çarpımların toplamı şeklindeki bir ifadeyi gerektirir. Fonksiyon tümleyeninin toplamların çarpımı halinde sadeleştirilmesi durumunda F' fonksiyonunu, fonksiyonun VE-VEYA kısmıyla uygulayabiliriz. F' TERSLE kısmından geçtiği zaman çıkışta F' tümleyenini, yani F fonksiyonunu elde ederiz.

Özet Tablo ve Örnek

Tablo 3.4'te, iki seviyeli dört formdan herhangi birisini kullanarak Boole fonksiyonlarını uygulama yöntemlerinin bir özeti verilmiştir. Her bir durumdaki

TERSLE kısmı nedeniyle fonksiyonun F' (tümleyen) sadeleştirmesini kullanmak yerinde olacaktır. F' bu formlardan birisiyle uygulandığı taktirde fonksiyonun tümleyenini VE-VEYA ya da VEYA-VE formunda elde ederiz. İki seviyeli dört form bu fonksiyonu evirir; bu da sonuçta F' (F tümleyeninin) tümleyeni olan bir çıkış verir. Bu ise normal F çıkışıdır.

Tablo 3.4 Diğer iki seviyeli formlarla uygulama

Eşdeğer bozuk olmayan (a)	(b)*	Uygulanan fonksiyon	F' sadeleştirme formu	Elde edilen çıkış
VE-VEYADEĞİL	VEDEĞİL-VE	VE-VEYA-TERSLE	Haritadaki 0'ları birleştirerek çarpımların toplamı	F
VEYA-VEDEĞİL	VEYADEĞİL-VEYA	VEYA-VE-TERSLE	Haritadaki 1leri birleştirip tümleyenini alarak toplamların çarpımı	F'

*(b) formu, tek literalli bir terim için tek girişli bir VEDEĞİL ya da VEYADEĞİL (evirme) kapısı gerektirir.

ÖRNEK 3.11: Şekil 3.19(a)'daki fonksiyonu Tablo 3.4'te verilen iki seviyeli dört formda uygulayın. Fonksiyonun tümleyeni, haritadaki 0'lar birleştirilerek çarpımların toplamı olarak sadeleştirilir.

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

Bu fonksiyon için normal çıkış,

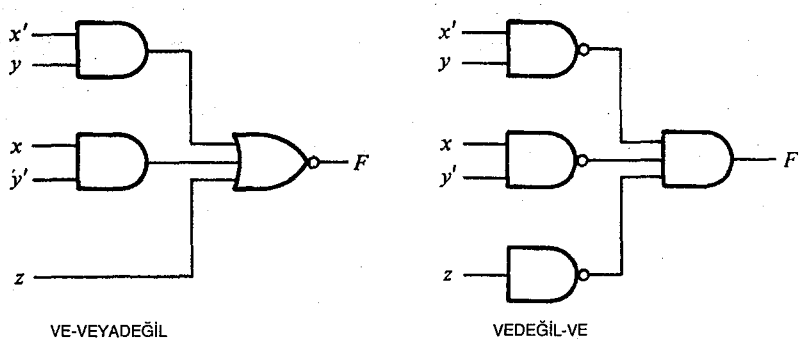
$$F = (x'y + xy' + z)'$$

olarak ifade edilebilir, ki bu da VE-VEYA-TERSLE formundadır. VE-VEYADEĞİL ve VEDEĞİL-VE uygulamaları Şekil 3.25(a)'da verilmiştir. VEDEĞİL-VE uygulaması için tek girişli bir VEDEĞİL veya tersleme kapısına ihtiyaç duyulduğuna, ancak VE-VEYADEĞİL durumunda buna gerek olmadığına dikkat edin. z yerine z' giriş değişkenini uyguladığımız taktirde tersleyici iptal edilebilir.

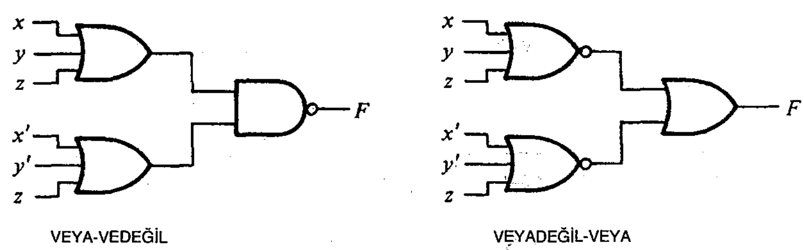
VEYA-VE-TERSLE formları, fonksiyonun tümleyeninin toplamların çarpımı halinde sadeleştirilmesini gerektirir. Bu ifadeyi elde etmek için ilk önce haritadaki 1'leri birleştirmemiz gerekir:

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

Daha sonra fonksiyonun tümleyenini alırız:



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



(b) $F = [(x + y + z)(x' + y' + z)]'$

Şekil 3.25 Diğer iki seviyeli uygulamalar

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

Böylece normal F çıkışı:

$$F = [(x + y + z)(x' + y' + z)]'$$

şeklinde ifade edilebilir, ki bu da VEYA-VE-TERSLE formundadır. Bu ifadeden, Şekil 3.25(b)'de gösterildiği gibi, fonksiyonu, VEYA-VEDEĞİL ve VEYA DEĞİL-VEYA formlarında uygulayabiliriz.

3.8 DİKKATE ALINMAYAN DURUMLAR

Haritadaki 1'ler ve 0'lar, fonksiyonu sırasıyla 1'e veya 0'a eşitleyen değişkenler kombinasyonunu gösterir. Birleşimler genellikle fonksiyonun 1 olduğu durumları veren bir doğruluk tablosundan elde edilir. Diğer bütün durumlarda fonksiyonun 0'a eşit olduğu varsayılır. Bu varsayım her zaman doğru değildir, çünkü bazı giriş değişkenleri kombinasyonları hiç bir zaman görülmez. Örneğin dört bitlik ondalık bir kodda kullanılmayan altı kombinasyon vardır. Bu kod kullanılan bir sayısal devre, sistem gerektiği gibi çalıştığı sürece kullanılmayan bu kombinasyonların hiç bir zaman olmayacağı varsayımıyla çalışır. Sonuç olarak, değişkenlerin bu birleşimlerinde fonksiyon çıkışının ne olacağına aldırış etmeyiz. Fonksiyonun daha ileri düzeyde sadeleştirilmesi için harita üzerinde bu dikkate alınmaz durumları kullanılabilir.

Bir dikkate alınmaz kombinasyonunun harita üzerinde 1 ile işaretlenemeyeceğini kavramak gerekir, çünkü bu, söz konusu giriş kombinasyonu için fonksiyonun her zaman 1 olmasını gerektirecektir. Benzer bir şekilde, kareye 0 konulması, fonksiyonun 0 olmasını gerektirir. Dikkate alınmaz durumlarını 1'lerden ve 0'lardan ayırdetmek için için X kullanılacaktır.

Haritada fonksiyonu sadeleştirmek için bitişik kareler seçilirken X 'lerin 0 veya 1 olduğu varsayılabilir. Burada önemli olan hangisinin en basit ifadeyi verdiğidir. Buna ek olarak, daha geniş bir alanın kapsanmasına katkıda bulunmuyorsa, X hiç kullanılmayabilir de. Her durumda seçim, sadece ulaşılabilecek sadeleştirmeye bağlıdır.

ÖRNEK 3.12: Aşağıdaki Boole fonksiyonunu sadeleştirin:

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

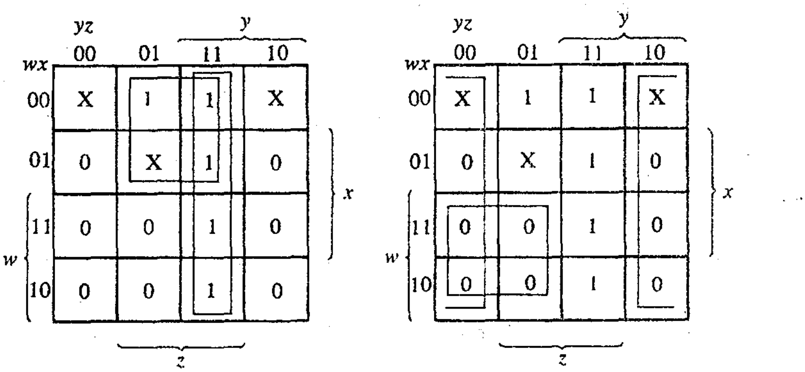
ve dikkate alınmaz durumları:

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

F mintermleri, fonksiyonu 1'e eşit yapan değişken kombinasyonlarıdır. d mintermleri ise hiç bir zaman olmayacağı bilinen dikkate alınmaz kombinasyonlarıdır. Sadeleştirme Şekil 3.26'da verilmiştir. F mintermleri 1 ile, d mintermleri X ile işaretlenmiş geri kalan kareler ise 0'la doldurulmuştur. (a)'da 1'ler ve X'ler, maksimum sayıda bitişik kareyi kapsayacak şekilde birleştirilmiştir. X'lerin tümünü veya herhangi birini dahil etmek gerekmez; sadece bir terimin sadeleştirilmesi için yararlı olanlar kullanılır. Minimum bir fonksiyon veren bir kombinasyonda bir X kullanılmış, diğer ikisi dışarıda bırakılmıştır. Bu, fonksiyonun çarpımların toplamı şeklinde sadeleştirilmiş halini verir:

$$F = w'z + yz$$

(b)'de, 0'lar, fonksiyonun tümleyenini sadeleştirmek için uygun olan X'lerle birleştirilmiştir. Gösterildiği gibi, en iyi sonuç, iki X'in kapsanması halinde



(a) 1'lerle X'lerin birleştirilmesi $F = w'z + yz$ (b) 0'larla X'lerin birleştirilmesi $F = z\,(w' + y)$

Şekil 3.26 dikkate alınmaz durumları için örnek

alınır. Tümleyen fonksiyon, aşağıdaki gibi sadeleşir:

$$F' = z' + wy'$$

Tekrar tümleyenini alırsak, sadeleştirilmiş bir toplamların çarpımı fonksiyonu elde ederiz:

$$F = z\,(w' + y)$$

3.12. örnekte elde edilen iki ifade, cebirsel olarak eşit oldukları kanıtlanabilen iki fonksiyon verir. Dikkate alınmaz durumları söz konusuysa her zaman böyle olmaz. Aslına bakılırsa 1'ler birleştirilirken X, 1 olarak; ya da 0'lar birleştirilirken 0 olarak kullanılırsa, sonuçtaki iki fonksiyon cebirsel açıdan eşit cevaplar vermeyecektir. Dikkate alınmaz durumunun ilkinde 1, ikincisinde 0 olarak seçilmesi, farklı minterm ifadelerine ve dolayısıyla farklı fonksiyonlara yol açar. 3.12. Örnekte bu görülebilir. Bu örneğin çözümünde X, 0 olarak değil, 1 olarak seçilmiştir. Şimdi Şekil 3.26(a)'da $w'z$ yerine $w'x'$ terimini seçersek, yine sadeleştirilmiş bir fonksiyon elde ederiz:

$$F = w'x' + yz$$

Ama cebirsel açıdan bu, toplamların çarpımıyla elde edilene eşit değildir, çünkü aynı X'ler ilkinde 1, ikincisinde ise 0 olarak kullanılmıştır.

Bu örnek ayrıca minimum sayıda literale sahip bir ifadenin mutlaka benzersiz (tek) olması gerekmediğini de gösterir. Tasarımcı bazen eşit sayıda literale sahip her biri sadeleştirilmiş bir ifadeyle sonuçlanan iki terim arasında bir seçim yapmak durumunda kalır.

3.9 TABLO YÖNTEMİ

Değişken sayısı beş veya altıyı geçmediği sürece haritayla sadeleştirme yöntemi uygundur. Değişken sayısı arttıkça aşırı ölçüde artan kare sayısı, makul bir bitişik kareler seçimini engeller. Haritanın açık dezavantajı, temelde insanın bazı yapıları (desenleri) farketme yeteneğine dayanan bir deneme yanılma yöntemi olmasıdır. Altı veya daha fazla değişkenli fonksiyonlarda en iyi seçimin yapıldığından emin olmak zordur.

Tablo yöntemi bu zorluğun aşılmasını sağlar. Bu, fonksiyon için sadeleştirilmiş standard formda bir ifadeyi garantileyen özel, adım adım bir işlemdir. Birçok değişkenli problemlere uygulanabilir ve bilgisayarla uygulamaya elverişlilik gibi bir avantajı vardır. Ancak oldukça yorucudur ve rutin, monoton işlemleri nedeniyle hataya açıktır. Tablo yöntemi ilk önce Quine (3) tarafından formüle edilmiş ve daha sonra McCluskey (4) tarafından geliştirilmiştir. Ayrıca Quine McCluskey yöntemi de denilmektedir.

Tabloyla sadeleştirme yöntemi iki kısımdan oluşur. İlkinde kapsamlı bir incelemeyle sadeleştirilmiş fonksiyona dahil edilebilecek tüm terimler bulunur. Bu terimlere *temel içeren* denir. İkinci adımda ise temel içerenler arasında en az sayıda literalli olan ifade seçilir.

3.10 TEMEL İÇERENLERİN BELİRLENMESİ*

Tablo yönteminin başlama noktası fonksiyonu belirleyen miniterimlerin listesini çıkarmaktır. İlk tablo işlemi ise eşleme yoluyla temel içerenlerin bulunmasıdır. Bu işlemde her minterm diğer her mintermle karşılaştırılır. İki mintermin sadece tek değişkende farklı olması halinde, o değişken atılır ve bir literal az olan bir terim bulunmuş olur. Bu işlem, kapsamlı arama tamamlanıncaya kadar her minterim için tekrarlanır. Eşleme işlemi çevrimi, bulunan yeni terimler için tekrarlanır. Bir çevrimde başkaca literal elemesi yapılamayacak bir noktaya kadar üçüncü, dördüncü, vb. çevrimlere devam edilir. Kalan terimlerle işlemde eşlenmeyen bütün terimler, temel içerenleri oluşturur. Bu tablo yöntemi ayağıdaki örnekle açıklanmıştır.

ÖRNEK 3.13: Tablo yöntemini kullanarak aşağıdaki Boole fonksiyonunu sadeleştirin.

$$F = \Sigma\,(0,1,2,8,10,11,14,15)$$

*Bu ve izleyen kısmın iptal edilmesi kitabın sürekliliğinde bir kayba yol açmaz.

1. Adım: Tablo 3.5, sütun (a)'da gösterildiği gibi, mintermlerin ikili gösterimlerini içerdikleri 1'lerin sayısına göre gruplandırın. Bu, mintermler yatay çizgilerle ayrılan beş bölüm halinde gruplandırılarak yapılmıştır. İlk bölüm, hiç 1 olmayan sayıları içerir. İkinci bölüm, sadece bir adet 1 olan sayıları içerir. Üçüncü, dördüncü ve beşinci bölümler ise sırasıyla iki, üç ve dört adet 1 bulunan ikili sayıları içerir. Tanıma amacıyla mintermlerin ondalık eşdeğerleri de yanlarına yazılmıştır.

2. Adım: Birbirinden sadece bir değişkenle farklı olan herhangi iki minterm birleştirilebilir ve eşlenmeyen değişken atılabilir. Her ikisinin de, bir bit dışında bütün pozisyonlarda aynı bit değerine sahip olması halinde, iki minterm sayısı bu kategoriye girer. Bir bölümdeki mintermler, sadece bir alt bölümdeki mintermlerle

Tablo 3-5 Örnek 3-13 için işaretli içerenlerin belirlenmesi

(a)			(b)			(c)	
	w x y z			*w x y z*			*w x y z*
0	0 0 0 0	√	0, 1	0 0 0 –		0, 2, 8, 10	– 0 – 0
			0, 2	0 0 – 0	√	0, 8, 2, 10	– 0 – 0
1	0 0 0 1	√	0, 8	– 0 0 0	√	10, 11, 14, 15	1 – 1 –
2	0 0 1 0	√				10, 14, 11, 15	1 – 1 –
8	1 0 0 0	√	2, 10	– 0 1 0	√		
			8, 10	1 0 – 0	√		
10	1 0 1 0	√					
			10, 11	1 0 1 –	√		
11	1 0 1 1	√	10, 14	1 – 1 0	√		
14	1 1 1 0	√					
			11, 15	1 – 1 1	√		
15	1 1 1 1	√	14, 15	1 1 1 –	√		

karşılaştırılır, çünkü birden fazla biti farklı olan iki terim eşlenemez. İlk bölümdeki minterm ikinci bölümdeki üç mintermden her birisiyle karşılaştırılır. Herhangi iki sayı birisi dışında her pozisyonda aynıysa, her iki mintermin sağına da kullanıldıklarını göstermek için bir kontrol (çek) işareti konur. Ondalık eşdeğerleriyle birlikte sonuçta elde edilen terim, tablonun (b) sütununda verilir. Eşleme sırasında elenen değişken, orijinal pozisyonunda bir tire ile belirtilir. Elimizdeki örnekte m_0 (0000), m_1 (0001) ile birleşerek (000-) sonucunu vermiştir. Bu kombinasyon aşağıdaki cebirsel işleme eşdeğerdedir:

$$m_0 + m_1 = w'x'y'z' + w'x'y'z = w'x'y'$$

Minterm m_0 ayrıca m_2 ile birleşerek (00-0), m_8 ile birleşerek de (-000) sonucunu verir. Bu karşılaştırmanın sonucu, (b) sütununun ilk bölümüne girilir. Daha sonra (a) sütununun ikinci ve üçüncü bölümlerindeki mintermler karşılaştırılır ve elde edilen sonuçlar (b) sütunundaki ikinci bölüme girilir. Bu kapsamlı karşılaştırma sonucunda (b) sütununda dört bölüm elde edilir.

3. Adım: (b) sütunundaki terimlerde sadece üç değişken vardır. Değişkenin altındaki bir 1, değişkenin işaretsiz, 0 işaretli, tire (-) ise değişkenin terime dahil edilmediğini gösterir. Arama ve karşılaştırma işlemi (b) sütunundaki terimler için tekrarlanarak (c) sütunundaki iki değişkenli terimler elde edilir. Yine burada da her bir bölümdeki terimlerin sadece aynı pozisyonda tire işareti olması halinde karşılaştırılması gerekir. (000-) teriminin hiç bir terimle eşlenmediğine dikkat edin. Bu nedenle sağına kontrol işareti konulmamıştır. Tanıma amacıyla her girişin sol tarafına ondalık eşdeğerleri yazılmıştır. Karşılaştırma işleminin (c) sütununda ve uygun eşleme görüldüğü sürece sonraki sütunlarda da tekrarlanması gerekir. Elimizdeki örnekte işlem üçüncü sütunda tamamlanır.

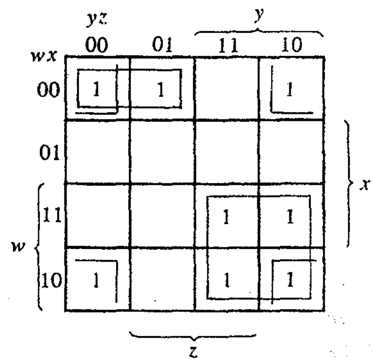
4. Adım: Tabloda işaretlenmemiş olan terimler temel içerenleri oluşturur. Örneğimizde (b) sütununda $w'x'y'$ (000-) terimi, (c) sütununda ise $x'z'$ (0-0) ve wy (1-1) terimleri bu türdendir. (c)'deki her bir terimin tabloda iki kere göründüğüne dikkat edin; terim bir temel içeren oluşturduğu sürece, aynı terimi iki kere kullanmak gereksizdir. Temel içerenlerin toplamı, fonksiyonun sadeleştirilmiş bir ifadesini verir. Bunun nedeni, tablodaki işaretli her terimin, sonraki bir sütunda daha basit bir terim girişiyle hesaba katılmış olmasıdır. Bu nedenle işaretsiz girişler (temel içerenler), fonksiyonu formüle etmek için kalan terimlerdir. Örneğimizde temel içerenler toplamı, sadeleştirilen fonksiyonu çarpımların toplamı formunda verir:

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

Bu cevabı, harita yöntemiyle elde edilenle karşılaştırmaya değer. Bu fonksiyonun harita sadeleştirmesi Şekil 3.27'de verilmiştir. Bitişik karelerin birleştirilmesi, fonksiyonun üç temel içerenini verir. Bu üç terimin toplamı, fonksiyonun çarpımların toplamı şeklindeki sadeleştirilmiş ifadesidir.

3.13. örneğin, sadeleştirilen fonksiyonu temel içerenler toplamı olarak vermek için amaçlı olarak seçildiğini belirtmekte yarar var. Diğer durumların çoğunda, temel içerenlerin mutlaka minimum sayıda terimli bir ifade vermesi gerekmez. Örnek 3.14'te bu gösterilmiştir.

Karşılaştırma ikili yerine ondalık sayılarla yapıldığı taktirde tablo yöntemi kullanılırken gösterilmesi gereken aşırı çaba azaltılabilir. Aşağıda, ikili sayıların



Şekil 3.27 Örnek 3.13'teki fonksiyon için harita; $F = w'x'y' + x'z' + wy$

karşılaştırılması ve eşlenmesi yerine ondalık sayıların çıkarılmasının kullanıldığı bir yöntem gösterilecektir. İkili bir sayıda her 1'in, 2'nin bir kuvvetiyle çarpılan bir katsayıyı temsil ettiğini biliyoruz. İki miniterim biri dışında bütün pozisyonlarda aynıysa, fazla 1'i olan miniterimin, diğer miniterimin sayısından 2'nin kuvveti kadar büyük olması gerekir. Bu nedenle İlk miniterim, bir sonraki alt bölümde bulunan ikinci ve daha büyük olan miniterimden 2'nin kuvveti kadar farklı ise, bu iki minterm birleştirilebilir. Bu işlemi 3.3. Örneği tekrarlayarak göstereceğiz.

Tablo 3.6, sütun (a)'da görüleceği üzere, mintermler, daha önceki gibi bölümler halinde düzenlenmiştir; ancak burada mintermlerin ondalık eşdeğerleri verilmiştir. Miniterim karşılaştırma işlemi şu şekilde yapılır: Tablonun bitişik bölümlerindeki her iki ondalık sayıyı inceleyin. Alt bölümdeki sayı üst bölümdeki sayıdan 2'nin kuvveti (yani, 1, 2, 4, 8, 16, vb.) kadar *büyükse*, kullanıldıklarını göstermek için her iki sayıyı da işaretleyin ve (b) sütununa yazın. (b) sütununa aktarılan sayı çiftleri, sayılar arasındaki fark olan 2'nin kuvvetini parantez içinde gösteren bir sayı içerir. Parantez içindeki sayı bize ikili gösterimdeki tirenin yerini gösterir. (a) sütunundaki bütün karşılaştırmaların sonucu (b) sütununa aktarılır.

(b) sütunundaki bitişik (komşu) bölümlerdeki karşılaştırma aynı şekilde yapılır; ancak burada sadece parantez içinde aynı sayıya sahip olanlar karşılaştırılır. Bir bölümdeki sayı çiftlerinin bir sonraki bölümdeki sayı çiftlerinden 2'nin kuvveti kadar farklı olması gerekir. Ve birleştirmenin olabilmesi için alt bölümdeki sayıların üst bölümdekinden *büyük* olması gerekir. (c) sütununda, parantez içinde tirelerin yerini gösteren iki sayıya sahip dört ondalık sayı yazılır. Tablo 3.6'daki sonuçları anlamak için 3.5 ve 3.6. Tabloları karşılaştırmak yararlı olabilir.

Temel içerenler, tabloda işaretlenmeyen terimlerdir. Bunlar, ondalık gösterimde verilmiş olmalarının dışında, öncekilerle aynıdır. Ondalıktan ikiliye çevirmek için, terimdeki bütün ondalık sayıları ikiliye çevirin ve daha sonra parantezler içindeki sayılarla gösterilen pozisyonlara bir tire koyun. Böylece 0, 1

Tablo 3.6 Örnek 3.13'teki fonksiyonun temel içerenlerin ondalık gösterimle belirlenmesi

(a)	(b)	(c)
0 √	0, 1 (1)	0, 2, 8, 10 (2, 8)
	0, 2 (2) √	0, 2, 8, 10 (2, 8)
1 √	0, 8 (8) √	
2 √		10, 11, 14, 15 (1, 4)
8 √	2, 10 (8) √	10, 11, 14, 15 (1, 4)
	8, 10 (2) √	
10 √		
	10, 11 (1) √	
11 √	10, 14 (4) √	
14 √		
	11, 15 (4) √	
15 √	14, 15 (1) √	

(1) ikiliye 0000, 0001 olarak çevrilir; sayıların ilk pozisyonuna tire konulmasıyla (000-) elde edilir. Aynı şekilde, 0, 2, 8, 10 (2, 8), ikili gösterime çevrilerek 0000, 0010, 1000 ve 1010 elde edilir ve 2 ile 8. pozisyonlara tire konarak (-0-0) elde edilir.

ÖRNEK 3.14: Aşağıdaki fonksiyonun temel içerenlerini belirleyin:

$$F(w,x,y,z) = \Sigma(1,4,6,7,8,9,10,11,15)$$

Minterm sayıları, Tablo 3.7, sütun (a)'daki gibi bölümlere ayrılmıştır. Mintermin ikili eşdeğeri 1'lerin sayısını saymak amacıyla konulmuştur. İlk bölümdeki ikili sayılarda sadece bir adet 1 vardır; ikinci bölümde iki 1 vardır, vs. Minterm sayıları ondalık yöntemle karşılaştırılır ve bir alt bölümdeki sayının üst bölümdekinden büyük olması halinde bir eşleme sağlanmış olur. Alt bölümdeki sayının üst bölümdekinden küçük olması halinde, iki sayı birbirinden 2'nin kuvveti kadar farklı olsa da eşleme olarak kaydedilmez. (a) sütunundaki kapsamlı (tüm sayıların) karşılaştırılması sonucu (b) sütunundaki terimler elde edilir; (a)'daki bütün terimler işaretlenmiştir. (b) sütununda sadece iki terim eşlemesi vardır. Her biri (c) sütununda kaydedilen aynı iki literalli terimi verir. Temel içerenler, tablodaki işaretlenmeyen bütün terimlerden oluşur. Ondalıktan ikiliye çevrim, tablonun altında verilmiştir. Bulunan temel içerenler şunlardır: $x'y'z'$, $w'xz'$, $w'xy$, xyz, wyz ve wx'.

Tablo 3.7 Örnek 3.14 için temel içerenlerin belirlenmesi

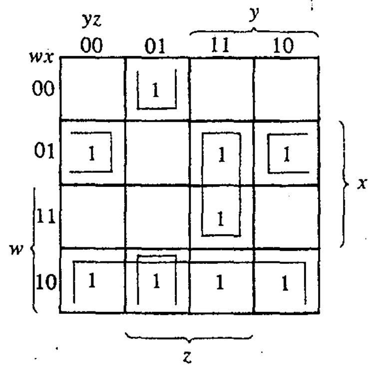
(a)			(b)			(c)
0001	1	√	1, 9	(8)		8, 9, 10, 11 (1, 2)
0100	4	√	4, 6	(2)		8, 9, 10, 11 (1, 2)
1000	8	√	8, 9	(1)	√	
			8, 10	(2)	√	
0110	6	√				
1001	9	√	6, 7	(1)		
1010	10	√	9, 11	(2)	√	
			10, 11	(1)	√	
0111	7	√				
1011	11	√	7, 15	(8)		
			11, 15	(4)		
1111	15	√				

Ondalık	İkili $w\ x\ y\ z$	Terim
1, 9 (8)	– 0 0 1	$x'y'z$
4, 6 (2)	0 1 – 0	$w'xz'$
6, 7 (1)	0 1 1 –	$w'xy$
7, 15 (8)	– 1 1 1	xyz
11, 15 (4)	1 – 1 1	wyz
8, 9, 10, 11 (1, 2)	1 0 – –	wx'

Temel içerenler toplamı fonksiyon için geçerli bir cebirsel ifade verir. Ne var ki bu ifadenin en az sayıda terimli olması gerekmez. Bu, 3.14. Örnekteki fonksiyonun haritasının incelenmesiyle görülebilir. Şekil 3.28'de de gösterildiği gibi, sadeleştirilen fonksiyon:

$$F = x'y'z + w'xz' + xyz + wx'$$

olarak gözlenmektedir; ki bu da 3.14. örnekte elde edilen altı temel içerenden dördünün toplamından oluşur. Sadeleştirilmiş fonksiyonu veren temel içerenlerin seçimi için kullanılan tablo yöntemi aşağıdaki bölümde anlatılmıştır.



Şekil 3-28 Örnek 3-14'teki fonksiyon için harita: $F = x'y'z + w'xz' + xyz + wx'$

3.11 TEMEL İÇERENLERİN SEÇİMİ

Sadeleştirilmiş fonksiyonu oluşturan temel içerenlerin seçimi, bir temel içerenler tablosundan yapılır. Bu tabloda her temel içeren bir satırda ve her minterm bir sütunda gösterilir. Temel içerenleri oluşturan mintermler kompozisyonunu göstermek için her satıra çarpı işaretleri konur. Daha sonra fonksiyondaki bütün mintermleri kapsayan minimum bir temel içerenler kümesi seçilir. Bu işlem Örnek 3.15'te açıklanmıştır.

ÖRNEK 3.15: Örnek 3.14'teki fonksiyonu sadeleştirin (minimuma indirin). Bu örneğin temel içerenler tablosu Tablo 3.8'de verilmiştir. 3.14. Örnekte bulunan her bir temel içeren için bir tane olmak üzere altı satır, ve her biri fonksiyonun bir mintermini gösteren dokuz sütun vardır. Satırdaki temel içerende bulunan mintermleri göstermek için her bir satıra çarpı işaretleri konmuştur. Örneğin ilk satırdaki iki çarpı işareti, 1 ve 9 nolu mintermlerin, $x'y'z$ temel içereninde bulunduğunu gösterir. Kapsadığı mintermi vermesi açısından, her satırdaki temel içerenin ondalık eşdeğerinin eklenmesi tavsiye edilir. Bütün çarpılar konduktan sonra *minimum* sayılı bir temel içerenin seçimine geçeriz.

Tamamlanan temel içerenler tablosu, tek bir çapraz işareti içeren sütunların belirlenmesi için incelenir. Örneğimizde sütunlarında tek çarpı bulunan dört minterm vardır: 1, 4, 8 ve 10. Minterm 1, $x'y'z$ temel içereninde bulunmaktadır, yani $x'y'z$ temel içereninin seçilmesi, minterm 1'in fonksiyona dahil edilmesini garanti eder. Aynı şekilde minterm 4, $w'xz'$ temel içerenindedir, ve 8 ile 10 mintermleri wx' temel içerenindedir. Sütunlarında tek bir çarpı bulunan mintermleri kapsayan temel içerenler, hakiki *temel içerenler* olarak adlandırılır. Bütün mintermleri içeren

Tablo 3-8 Örnek 3-15 için temel içeren tablo

		1	4	6	7	8	9	10	11	15
√ $x'y'z$	1, 9	X					X			
√ $w'xz'$	4, 6		X	X						
$w'xy$	6, 7			X	X					
xyz	7, 15				X					X
wyz	11, 15								X	X
√ wx'	8, 9, 10, 11					X	X	X	X	
		√	√	√		√	√	√	√	

sadeleştirilmiş nihai ifadeyi bulmak için, hakiki *temel içerenleri* dahil etmekten başka seçeneğimiz yoktur. Seçildiklerini göstermek amacıyla tabloda hakiki temel içerenlerin yanına bir kontrol işareti konur.

Daha sonra mintermi seçilen hakiki temel içerenlerde bulunan her bir sütunu kontrol ederiz. Örneğin seçilen $x'y'z$ temel içerenlerinin, 1 ve 9 mintermlerini içerir. Sütunların altına bir kontrol işareti konulmuştur. Aynı şekilde, $w'xz'$ temel içereni, 8, 9, 10 ve 11 mintermlerini kapsar. Temel içerenler tablosunun incelenmesi, hakiki temel içerenlerin seçiminin, 7 ve 15 dışında fonksiyonun tüm mintermlerini kapsadığını gösterir. Bir veya daha fazla temel içerenlerin seçilerek bu iki mintermin de dahil edilmesi gerekir. Örneğimizde xyz temel içerenlerinin her iki mintermi de içerdiği ve seçilmesi gereken temel içeren olduğu açıktır. Böylece toplamı istenen aşağıdaki sadeleştirilmiş fonksiyonu veren minimum temel içerenler kümesini elde etmiş oluruz:

$$F = x'y'z + w'xz' + wx' + xyz$$

Yukarıdaki örneklerden edle edilen sadeleştirilmiş ifadelerin tamamı çarpımların toplamı şeklindeydi. Tablo yöntemi, toplamların çarpımı şeklinde sadeleştirilmiş bir ifade elde etmek için uyarlanabilir. Harita yönteminde olduğu gibi, ilk önce başlangıçtaki mintermler listesi olarak 0'ları almak suretiyle fonksiyonun tümleyeniyle işe başlamamız gerekecektir. Bu liste, orijinal fonksiyonda bulunmayan ve nümerik olarak fonksiyonun makstermlerine eşit olan mintermleri içerir. Tablolama işlemi fonksiyonun 0'larıyla yapılır ve sonuçta fonksiyonun tümleyeninin çarpımlarının toplamı şeklindeki sadeleştirilmiş ifade elde edilir. Tekrar tümleyenini almak suretiyle sadeleştirilmiş toplamların çarpımı ifadesini elde ederiz.

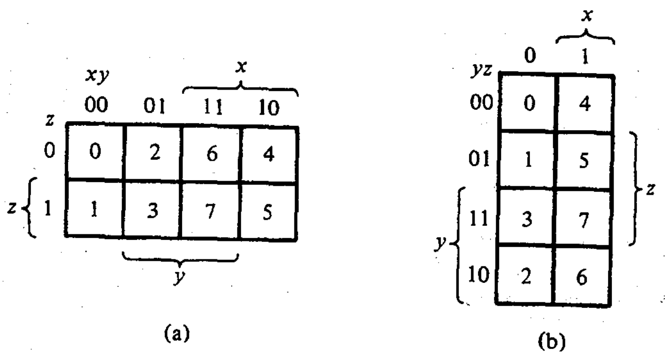
Dikkate alınmaz durumlu bir fonksiyon, küçük bir değişiklikle tablo yöntemiyle sadeleştirilebilir. dikkate alınmaz terimleri, temel içerenler belirlenirken listeye dahil edilir. Bu da en az sayıdaki literalli temel içerenlerinin belirlenmesini mümkün kılar. Temel içerenler tablosu hazırlanırken dikkate alınmaz terimler, mintermler listesine alınmaz, çünkü seçilen temel içerenlerinin, dikkate alınmaz terimlerini içermesi gerekmez.

3.12 SONUÇ

Bu bölümde Boole fonksiyonlarını sadeleştirmek için kullanılan iki yöntem anlatılmıştır. Sadeleştirme kriteri, çarpımların toplamı veya toplamların çarpımı ifadelerindeki literal sayısının en aza indirilmesidir. Sadece standard formlarda ifade edilen Boole fonksiyonlarının sadeleştirilmesinde kullanılabilmeleri nedeniyle hem harita hem de tablo yöntemlerinin sınırları vardır. Bu, metodlar için bir dezavantaj olsa da, çok belirleyici değildir. Uygulamaların çoğunda diğer formlara kıyasla standard formlar tercih edilir. Şekil 3.15'te, standard formdaki ifadelerin kapı uygulamasının, en fazla iki seviyeli kapılardan oluştuğunu görmüştük. Standard formda olmayan ifadeler ikiden fazla seviyeyle uygulanmaktadır. Mumphery (5), harita yöntemini, sadeleştirilmiş çok seviyeli ifadeler verecek şekilde genişletmiştir.

Haritalar için seçilen yansıyan kod sırasının benzersiz olmadığını kavramak gerek. Bir harita çizip burada kullanılan sıradan farklı satır ve sütunlara ikili bir yansıyan kod sırası vermek mümkündür. Seçilen ikili sıra bitişik kareler arasında sadece bir bit değişikliği yarattığı sürece, geçerli ve yararlı bir harita oluşturacaktır.

Sayısal mantık literatüründe sık sık görülen üç değişkenli haritaların iki farklı versiyonu Şekil 3.29'da verilmiştir. Referans sağlamak için minterm sayıları her



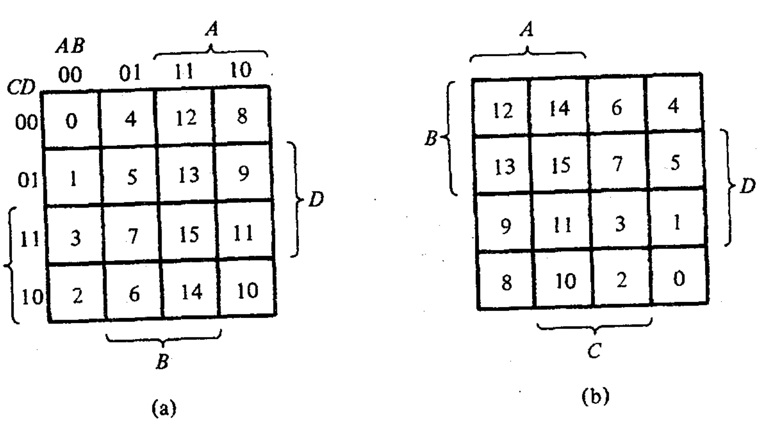
Şekil 3.29 Üç değişkenli harita çeşitleri

bir kareye yazılmıştır. (a)'daki satır ve sütunlara değişken tahsihi, bu kitapta kullanılandan farklıdır. (b)'de ise harita düşey pozisyonda döndürülmüştür. Minterm sayı tahsisleri bütün haritalarda xyz sırasında kalmaktadır. Örneğin minterm 6 karesi, sıralı değişkenlere xyz = 110 ikili sayısı verilerek bulunur. Bu mintermin karesi (a)'da xy = 11 işaretli sütundan ve z = *0* işaretli satırdan bulunur. (b)'de buna karşılık gelen kare ise x = *1* işaretli sütuna ve yz = 10 işaretli satıra aittir. Bu haritalardaki sadeleştirme işlemi bu bölümde anlatılanın aynısıdır, ancak elbette minterm ve değişken tahsisi farklılık göstermektedir.

Dört değişkenli haritanın iki farklı versiyonu ise Şekil 3.30'da verilmiştir. (a)'daki harita çok popülerdir ve literatürde sık sık kullanılmaktadır. Burada da fark küçüktür ve satırlardan sütunlara ve sütunlardan satırlara değişken tahsisi farklarında ortaya çıkar. (b)'deki harita, orijinal Veitch şemasıdır (1), ki bu daha sonra Karnaugh (2) tarafından (a)'daki gibi değiştirilmiştir. Bu kitaptakinin yerine bu haritalar kullanıldığı zaman sadeleştirme işlemleri değişmez. Ayrıca beş ve altı değişkenli harita çeşitleri de vardır. Her ne olursa olsun, bu kitapta kullanılandan farklı gözüken veya farklı bir isimle adlandırılan bir haritanın, sadece haritadaki karelere minterm tahsisinde farklılık gösteren bir harita olduğu bilinmelidir.

3.13 ve 3.14. Örneklerden de açık olduğu gibi, tablo yönteminin, uzun listelerde sayılar karşılaştırılırken kaçınılmaz olarak hataya neden olması gibi bir kusuru vardır. Harita yöntemi tercih edilebilir gibi gözükecektir, ancak beşten fazla değişkenler durumunda en iyi sadeleştirilmiş ifadenin bulunduğundan emin olamayız. Tablo yönteminin gerçek avantajı, bir cevabı garanti eden özel, adım adım işlemlerden oluşması gerçeğinde yatar. Dahası, bu formel yöntem, bilgisayar mekanizasyonu için de elverişlidir.

3.9. Bölümde, tablo yönteminin her zaman için fonksiyonun minterm listesiyle başladığını belirtmiştik. Fonksiyonun bu şekilde olmaması halinde,



Şekil 3.30 Dört değişkenli harita çeşitleri

çevrilmesi gerekir. Uygulamaların çoğunda sadeleştirilecek fonksiyon, minterm listesi kolayca çıkarılabilen bir doğruluk tablosundan alınır. Aksi taktirde, mintermlere çevirme işi önemli ölçüde çalışmayı gerektirir. Ne var ki tablo yönteminin, keyfi olarak seçilen çarpımların toplamı ifadelerinden temel içerenleri bulmak için kullanılan versiyonları da vardır. Örneğin bakınız McCluskey (7).

Bu bölümde birçok giriş değişkeni ve tek bir çıkış değişkeni olan fonksiyonların sadeleştirilmesini inceledik. Ne var ki bazı sayısal devrelerde birden çok çıkış vardır. Bu tür devreler, her bir çıkış değişkeni için bir tane olmak üzere, bir Boole fonksiyonları kümesiyle tanımlanır. Çoklu çıkışa sahip bir devrede çeşitli fonksiyonlar arasında, bazen uygulama sırasında ortak kapıların oluşturulması için kullanılabilecek ortak terimler bulunabilir. Bu da her fonksiyonun ayrı ayrı sadeleştirilmesi durumunda dikkate alınmayan ek bir sadeleştirme sağlar. Çoklu çıkışlı devreler için tablo yönteminin değişik bir türü vardır (6, 7). Ancak bu yöntem de çok özelleşmiştir ve insan kullanımı için son derece yorucudur. Sadece kullanıcının bu yönteme dayalı bir bilgisayar programı kullanabilmesi durumunda pratik bir önem taşır.

REFERANSLAR

1. Veitch, E. W., "A Chart Method for Simplifying Truth Functions." *Proc. of the ACM (May* 1952), 127-33.
2. Karnaugh, M., "A Map Method for Synthesis of Combinational Logic Circuits." *Trans. AIEE, Comm. and Electronics,* Vol. 72, Part I (November 1953), 593-99.
3. Quine, W.V., "The Problem of Simplifying Truth Functions." *Am. Math. Monthly,* Vol. 59, No. 8 (October 1952), 521-31.
4. McCluskey, E. J., Jr., "Minimization of Boolean Functions". *Bell System Tech.,* J., Vol. 35, No. 6 (November 1956), 1417-44.
5. Humphrey, W. S., Jr., *Switching Circuits with Computer Applications.* New York: McGraw-Hill Book Co., 1958, Chap. 4.
6. Hill, F., J., and G.R. Peterson, *Introduction to Switching Theory and Logical Design,* 3rd ed. New York: John Wiley & Sons, Inc., 1981.
7. McCluskey, E. J., Jr., *Introduction to the Theory of Switching Circuits.* New York: McGraw-Hill Book Co., 1965, Chap. 4.
8. Kohavi, Z., *Switching and Finite Automata* Theory. New York: McGraw-Hill Book Co., 1970.
9. Nagle, H. T. Jr., B. D. Carrol and J. D. Irwin, *An Indroduction to Computer* Logic. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1975.

PROBLEMLER

3.1. Aşağıdaki Boole fonksiyonlarının sadeleştirilmiş ifadelerini, çarpımların toplamı şeklinde elde edin:

(a) $F(x,y,z) = \Sigma(2,3,6,7)$

(b) $F(A,B,C,D) = \Sigma(7,13,14,15)$

(c) $F(A,B,C,D) = \Sigma(4,6,7,15)$

(d) $F(w,x,y,z) = \Sigma(2,3,12,13,14,15)$

3.2 Aşağıdaki Boole fonksiyonlarının sadeleştirilmiş ifadelerini, çarpımların toplamı şeklinde elde edin:

(a) $xy + x'y'z' + x'yz'$

(b) $A'B + BC' + B'C'$

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

(d) $xy'z + xyz' + x'yz + xyz$

3.3 Aşağıdaki Boole fonksiyonlarının sadeleştirilmiş ifadelerini, çarpımların toplamı şeklinde elde edin:

(a) $D(A' + B) + B'(C + AD)$

(b) $ABD + A'C'D' + A'B + A'CD' + AB'D'$

(c) $k'lm' + k'm'n + klm'n' + lmn'$

(d) $A'B'C'D' + AC'D' + B'CD' + A'BCD + BC'D$

(e) $x'z + w'xy' + w(x'y + xy')$

3.4 Aşağıdaki Boole fonksiyonlarının sadeleştirilmiş ifadelerini, çarpımların toplamı şeklinde elde edin:

(a) $F(A,B,C,D,E) = \Sigma(0,1,4,5,16,17,21,25,29)$

(b) $BDE + B'C'D + CDE + A'B'CE + A'B'C + B'C'D'E'$

(c) $A'B'CE' + A'B'C'D' + B'D'E' + B'CD' + CDE' + BDE'$

3.5. Aşağıdaki doğruluk tablosu verilmiş olsun:

x	y	z	F_1	F_2
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(a) F_1 ve F_2 fonksiyonlarını, makstermlerin çarpımı olarak ifade edin.
(b) Sadeleştirilmiş fonksiyonları, çarpımların toplamı olarak bulun.
(c) Sadeleştirilmiş fonksiyonları, toplamların çarpımı olarak bulun.

3.6 Sadeleştirilmiş ifadeleri, toplamların çarpımı olarak bulun.
(a) $F(x,y,z) = \prod(0,1,4,5)$
(b) $F(A,B,C,D) = \prod(0,1,2,3,4,10,11)$
(c) $F(w,x,y,z) = \prod(1,3,5,7,13,15)$

3.7 Sadeleştirilmiş ifadeleri, (1) çarpımların toplamı ve (2) toplamların çarpımı şeklinde bulun:
(a) $x'z' + y'z' + yz' + xyz$
(b) $(A + B' + D)(A' + B + D)(C + D)(C' + D')$
(c) $(A' + B' + D')(A + B' + C')(A' + B + D')(B + C' + D')$
(d) $(A' + B' + D)(A' + D')(A + B + D')(A + B' + C + D)$
(e) $w'yz' + vw'z' + vw'x + v'wz + v'w'y'z'$

3.8. VE ve VEYA kapıları kullanarak, 3.7. problemde elde edilen sadeleştirilmiş Boole fonksiyonlarının, kapı uygulamasını çizin.

3.9. Aşağıdaki fonksiyonları sadeleştirin ve VEDEĞİL kapıları kullanarak uygulayın. İki alternatif verin.
(a) $F_1 = AC' + ACE + ACE' + A'CD' + A'D'E'$
(b) $F_2 = (B' + D')(A' + C' + D)(A + B' + C' + D)(A' + B + C' + D')$

3.10. 3.9. problemi, VEYADEĞİL kapısıyla tekrarlayın.

3.11. Aşağıdaki fonksiyonları VEDEĞİL kapılarıyla uygulayın. Hem normal hem de tümleyen girişlerin mevcut olduğunu varsayın.
(a) $BD + BCD + AB'C'D' + A'B'CD'$
(Herbiri 3 girişe sahip ve 6'danfazla kapı olmamalı)
(b) $(AB + A'B')(CD' + C'D)$ (İki girişli kapılarla)

3.12. Aşağıdaki fonksiyonları VEYADEĞİL kapılarıyla gerçekleştirin. Hem normal hem de tümleyen girişlerin mevcut olduğunu varsayın.
(a) $AB' + C'D' + A'CD' + DC'(AB + A'B') + DB(AC' + A'C)$
(b) $AB'CD' + A'BCD' + AB'C'D + A'BC'D$

3.13. İki seviyeli sekiz bozuk formu sayın ve tek bir işleme indirgendiğini kanıtlayın. Kapıların giriş yelpazesinin genişletilmesi için iki seviyeli bozuk formların nasıl kullanılabileceğini açıklayın.

3.14. 3.9. problemdeki fonksiyonları, aşağıdaki iki seviyeli formlarla uygulayın: VEYADEĞİL-VEYA, VEDEĞİL-VE, VEYA-VEDEĞİL ve VE-VEYADEĞİL.

3.15. Aşağıdaki F Boole fonksiyonlarını, d dikkate alınmaz durumlarını kullanarak çarpımların toplamı halinde sadeleştirin:

(a) $F = y' + x'z'$
$d = yz + xy$

(b) $B'C'D' + BCD' + ABC'D$
$d = B'CD' + A'BC'D'$

3.16. Aşağıdaki F Boole fonksiyonlarını, d dikkate alınmaz durumlarını kullanarak (1) çarpımların toplamı ve (2) toplamların çarpımları şeklinde sadeleştirin:

(a) $F = A'B'D' + A'CD + A'BC$
$d = A'BC'D + ACD + AB'D'$

(b) $F = w'(x'y + x'y' + xyz) + x'z'(y + w)$
$d = w'x(y'z + yz') + wyz$

(c) $F = ACE + A'CD'E' + A'C'DE$
$d = DE' + A'D'E + AD'E'$

(d) $F = B'DE' + A'BE + B'C'E' + A'BC'D'$
$d = BDE' + CD'E'$

3.17. Dikkate alınmaz durumları kullanarak aşağıdaki fonksiyonları uygulayın. Hem normal hem de tümleyen girişlerin mevcut olduğunu varsayın.

(a) $F = A'B'C' + AB'D + A'B'CD'$ En fazla iki VEYADEĞİL kapısıyla
$d = ABC + AB'D'$

(b) $F = (A + D)(A' + B)(A' + C')$ En fazla üç VEDEĞİL kapısıyla

(c) $F = B'D + B'C + ABCD$ VEDEĞİL kapılarıyla
$d = A'BD + AB'C'D'$

3.18. Aşağıdaki fonksiyonu VEDEĞİL ya da VEYADEĞİL kapılarıyla uygulayın. Sadece dört kapı kullanın. Sadece normal çıkışlar mevcuttur.

$$F = w'xz + w'yz + x'yz' + wxy'z$$
$$d = wyz$$

3.19. $BE + B'DE'$ Boole ifadesi

$A'BE + BCDE + BC'D'E + A'B'DE' + B'C'DE'$

ifadesinin sadeleştirilmiş bir versiyonu mudur?
Dikkate alınmaz durumları var mıdır? Varsa, nelerdir?

3.20. $F = A'B'D' + AB'CD' + A'BD + ABC'D$

fonksiyonunu sekiz veya daha az literalle ifade etmenin üç muhtemel yolunu açıklayın.

3.21. f ve g terimlerinin aşağıdaki ifadeyle verili olduğu bir $F = fg$ fonksiyonunun çarpımların toplamı halindeki en basit formunu harita kullanarak belirleyin:

$$f = wxy' + y'z + w'yz' + w'yz'$$
$$g = (w + x + y' + z')(x' + y' + z)(w' + y + z')$$

İpucu: 8.2(b). probleme bakın.

3.22. Şekil 3.29(a)'da tanımlanan haritayı kullanarak 3.2(a) problemindeki Boole fonksiyonunu sadeleştirin. Şekil 3.29(b)'deki harita ile tekrarlayın.

3.23. Şekil 3.30(a)'da tanımlanan haritayı kullanarak 3.3(a) problemindeki Boole fonksiyonunu sadeleştirin. Şekil 3.30(b)'deki harita ile tekrarlayın.

3.24. Aşağıdaki Boole ifadelerini tablo yöntemini kullanarak sadeleştirin:
(a) $F(A,B,C,D,E,F,G) = \Sigma(20,28,52,60)$
(b) $F(A,B,C,D,E,F,G) = \Sigma(20,28,38,39,52,60,102,103,127)$
(c) $F(A,B,C,D,E,F) = \Sigma(6,9,13,18,19,25,27,29,41,45,57,61)$

3.25. 3.6. problemi tablo yöntemini kullanarak tekrarlayın.

3.26. 3.16(c) ve (d) şıklarını tablo yöntemini kullanarak tekrarlayın.