



Kayseri Üniversitesi
Mühendislik-Mimarlık ve Tasarım Fakültesi
Elektrik-Elektronik Mühendisliği Bölümü

EEM 201 LOJİK DEVRELERE GİRİŞ

Dr. Öğr. Üyesi Ahmet GANİ

DERS İÇERİĞİ

Dersin içeriği	
1-	Sayısal Sistemlere Genel Bakış
2-	Sayı Sistemleri ve Dönüşümler
3-	Sayısal Kodlama
4-	Lojik Kapılar
5-	Boole Cebri Aksiyom ve <u>Demorgan</u> Teoremleri
6-	Lojik Fonksiyonların Sadeleştirilmesi ve Uygulamaları
7-	Dijital Entegre Lojik Aileler
8-	Lojik Devre Katalog Bilgileri
9-	<u>Kombinezonsal</u> Devrelere Giriş
10-	Ara Sınav Haftası
11-	<u>Kombinezonsal</u> Lojik Toplayıcı ve Çıkarıcı Devreler
12-	<u>Coğullayıcılar</u> , Kod Çözücüler, Kodlayıcılar
13-	Ardışıl Devrelere Genel Bir <u>Bakış</u> , <u>Tutucular</u> ve <u>Flip-Flop</u> Devreleri
14-	Ardışıl Lojik Devrelerin Analizi, Durum Diyagramları ve Tablolarının Türetilmesi
15-	Ardışıl Lojik Devrelerin Tasarımı
16-	Yarıyıl Sonu Sınavı

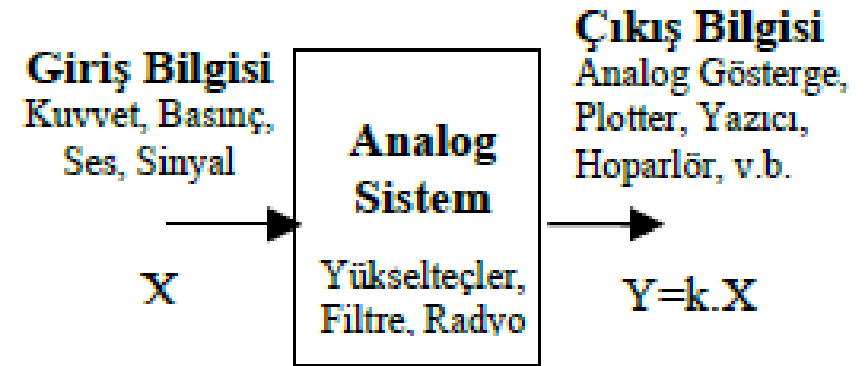
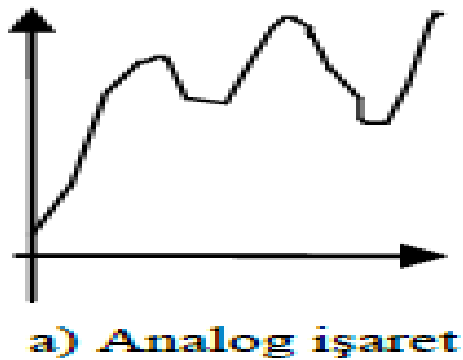
GİRİŞ

Bilim, teknoloji, ticaret ve benzeri bir çok alan büyüklükler ile ilgilenmektedir. Bu alanların ilgilendiği büyüklükler; ölçülebilme, görüntülenebilme, kaydedilebilme, aritmetik olarak hesaplanabilme, vb. özelliklere sahiptir. Büyüklükler ile işlem yapıldığı zaman onların sahip oldukları değerleri etkin ve güvenli olarak ifade etmek büyük bir önem taşır. Büyüklüklerin sayısal değerlerini ifade etmede, analog ve sayısal olarak isimlendirilen iki yöntem kullanılır.

İfade edilen büyüklüklerin, taşınabilir fiziksel büyüklüklere, örneğin bir gerilim veya akım şekline dönüştürülmeleri gerekebilir. Fiziksel bir olayın/büyüklüğün elektriksel olarak gösterilmesi, 'işaret' olarak adlandırılır. Gerek fiziksel büyüklükleri dönüştürme işleminde, gerekse de bilginin işlenmesi / iletilmesinde temel olarak analog ve sayısal (dijital) işaretlerden faydalanılır. Analog ve sayısal işaretler özelliklerine uygun devrelerde / sistemlerde işlemlere tabi tutulduktan sonra, çıkış birimi olarak isimlendirilen göstergeler yardımıyla insanlar için anlamlı hale getirilir. Yukarıda anlatılanlar ışığı altında karşımıza analog ve sayısal kavramları ve her bir kavram ile birlikte büyüklük, işaret, sistem ve gösterge terimleri ortaya çıkmaktadır.

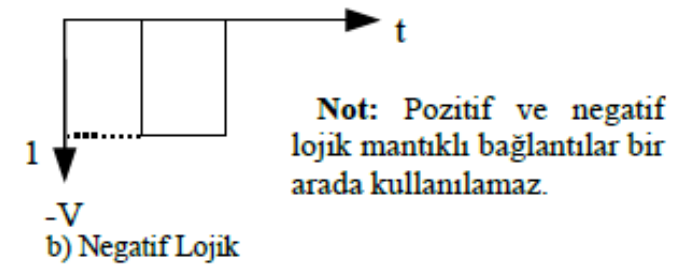
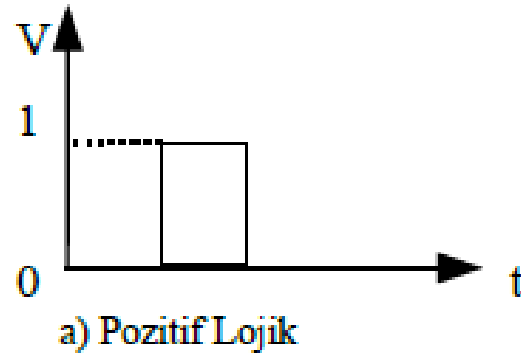
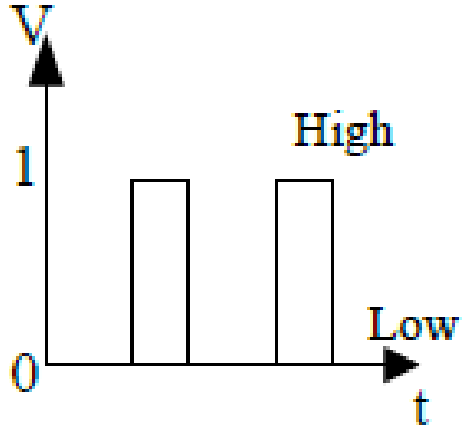
Analog Büyüklük, Analog İşaret ve Analog Sistem

Kesintisiz olarak sürekli değerler alan ve sahip oldukları değerler belirli sınırlar içerisinde devamlı olarak değişen büyüklük, 'analog büyüklük' olarak isimlendirilir. Fiziksel bir büyüklük (analog özelliğe sahip) bilgi şekline dönüştürülürken, bilgiyi temsil eden işaret doğrudan doğruya fiziksel büyüklüğün benzeri ise oluşan işaret 'analog işaret' olarak adlandırılır (Şekil 1.1.a). Giriş ve çıkış işaretleri şekil olarak benzeyen elektronik devreye / sisteme, 'analog (doğrusal) devre' veya 'analog sistem' denir (Şekil 1.1.b).



Sayısal Büyüklük, Sayısal İşaret ve Sayısal Sistem

Yalnızca iki değer alabilen (var-yok, açık-kapalı, vb.) büyüklük, 'sayısal büyüklük' olarak isimlendirilir. İki değerlikli büyüklük, işaret şekline dönüştürülürken yalnızca iki değere sahip işaret şeklinde gösterilir. Sayısal büyüklüğü göstermek için kullanılan ve '0', '1' gibi iki değer alabilen işaret, 'sayısal işaret' olarak adlandırılır (Şekil 1.2.a). Sayısal işaretin aldığı değerler adım adım değişir. Sayısal işaretle 0'dan 1'e ani değişim pozitif yönde ise 'pozitif mantık', ani değişim negatif yönde ise 'negatif mantık' olarak tanımlanır (Şekil 1.3).



Sayısal Büyüklük, Sayısal İşaret ve Sayısal Sistem

Sayısal işaretlerin aldıkları değerleri göstermek için 0-1, L-H (Low-High) sembolleri kullanılır. Sayısal teknikte kullanılan bu sembollerin çeşitli fiziksel anlamları olabilir. Sembollerin ifade ettiği anlamlardan birkaçı Tablo 1.1’de sıralanmaktadır.

‘0’ , ‘L’	‘1’ , ‘H’
Gerilim yok	Gerilim Var
Yanlış	Doğru
Kontak açık (röle)	Kontak kapalı (röle)
Hayır	Evet
İşaret yok	İşaret var
OFF	ON
Sıfır gerilim	Negatif veya pozitif gerilim
Transistör yalıtkan	Transistör iletken
1. Frekans	2.Frekans
Devre çalışmıyor	Devre çalışıyor

Sayısal Sistemler

Sayısal sistemler yaptıkları işlemlere göre üç genel grup altında incelenebilir:

1- Bileşik (Combinational) Sayısal Sistemler : Devrenin çıkışı, girişlerin o anki durumu ile doğrudan ilgili olan lojik devrelerdir. Temel lojik kapılarla yapılan tasarımlar ve toplayıcı / çıkarıcı devreleri bileşik devrelere örnek olarak gösterilebilir.

2- Ardışıl (Sequential) Sayısal Sistemler : Sistemin, daha önceden sahip olduğu konum ve hali hazırdaki giriş değişkenlerinin durumlarına bağlı olarak çıkış üreten sistemlerdir. Ardışıl devrelere örnek olarak; sayıcılar, kaydediciler, v.b. devreler verilebilir.

3- Bellek (Storage) Sistemleri : Bilgilerin veya Ardışıl lojiğin belirli bir durumunun saklanması amacıyla kullanılan lojik devrelerdir.

Sayı Sistemlerinin İncelenmesi

Sayı sistemlerini incelerken göz önünde bulundurmamız gereken ilk kavram; sayı sistemlerinde kullanılan rakam, işaret, karakter veya harfleri ve bunların temsil ettikleri anlamları açıklamaktır. Sayı sistemlerinde kullanılan rakamın / harfin / karakterin, sayı içerisinde bulunduğu yere (basamağa) bağlı olarak temsil ettiği anlamı değişir. Anlam değişikliğini belirleyen unsur, bulunan basamağın sayı sistemine bağlı olarak taşıdığı kök / taban değeridir. Bu durumda sayı sistemine bağlı olarak değişen ikinci kavram; sayı sistemlerinde kullanılan taban değeridir.

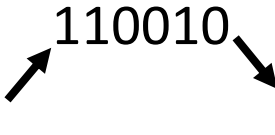
1.1. Onlu (Decimal) Sayı Sistemi

Günlük hayatımızda en çok kullandığımız onluk sayı sisteminde on değişik rakam vardır ve bunlar sırasıyla; 0, 1, 2, 3, 4, 5, 6, 7, 8, 9'dur. en sağdaki basamak en düşük ve en soldaki en yüksek anlamlı basamak olarak; 1985 sayısı, $1985 = 1 \cdot 10^3 + 9 \cdot 10^2 + 8 \cdot 10^1 + 5 \cdot 10^0$ şeklinde yazılabilir.

1.2. İkili (Binary-Dual) Sayı Sistemi

'0' ve '1' rakamları ile temsil edilen, taban değeri '2' olan ve iki olasılıklı durumları ifade etmek amacıyla kullanılan sayı sistemi 'İkili' veya 'Binary' sayı sistemi olarak adlandırılır. İkili sayı sisteminde her bir basamak 'BIT' olarak (Binary DigiT) adlandırılır. En sağdaki basamağa en 'En Düşük Değerli Bit' (Least Significant Bit - LSB), en soldaki basamağa

'En Yüksek Değerli Bit' (Most Significant Bit - MSB) denir.

En Yüksek Değerli Bit (MSB)  110010  En Düşük Değerli Bit (LSB)

İkili sayı sistemi bilgisayarlarda aşağıdaki amaçlar için kullanılmaktadır:

Gerçek sayısal değeri ifade etmek için,

ii. Veri ile ilgili bellekteki adresi belirtmek için,

iii. Komut kodu olarak,

iv. Alfabetik ve sayısal olmayan karakterleri temsil etmek için bir kod olarak,

v. Bilgisayarda dahili ve harici olarak bulunan devrelerin durumlarını belirlemesi için bir sayı grubu olarak.

1.3. Sekizli (Octal) Sayı Sistemi

İkili sayı sistemindeki sayıların daha kolay gösterilmesini sağlayan sayı sistemlerinden birisi, sekizli (octal) sayı sistemidir. Sekizli sayı sisteminde taban '8' ve kullanılan sayılar; 0, 1, 2, 3, 4, 5, 6, 7'dir.

1.4. Onaltılık (Hexadecimal) Sayı Sistemi

İkili sayı sisteminin daha kolay gösterilmesini sağlayan ve günümüz bilgisayarlarında yaygın olarak kullanılan sayı sistemi onaltılık (hexadecimal) sayı sistemidir. Onaltılı sayı sisteminde 0 ile 9 arasındaki rakamlar ile A, B, C, D, E, F harfleri kullanılır. Tablo 2.1'de 0-20 arasındaki onlu sayıların ikili, sekizli, onaltılı sayı sistemlerindeki karşılıkları gösterilmektedir. Buraya kadar sayı sistemlerini açıkladı. Şimdi bu sayı sistemlerinin birbirlerine dönüşümlerini açıklayalım.

1.4. Onaltılık (Hexadecimal) Sayı Sistemi

Onlu	İkili	Sekizli	Onaltılı
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

Tablo 2.1. 0-20 arası sayıların ikili, sekizli ve onaltılı sistemlerdeki karşılıkları

2. Sayı Sistemlerinin Birbirlerine Dönüştürülmeleri

• 2.1. Onlu Sayıların İkili, Sekizli ve Onaltılı Sayılara Dönüşümü

Onlu bir sayı başka bir sayıya dönüştürülecekse; onlu sayı, yeni oluşacak olan sayı sisteminin taban değerine sürekli bölünür. Bölüm sonucunda elde kalanların tersten sıralanmasıyla yeni sayı sistemindeki sayı bulunur.

Onlu Sayıların İkili Sayılara Dönüşümü:

Örnek 1: $(39)_{10}$ sayısını ikili sayı sistemine çevirelim.

<u>Bölünen</u>	<u>Bölüm</u>	<u>Kalan</u>	
39/2	19	+	1
19/2	9	+	1
9/2	4	+	1
4/2	2	+	0
2/2	1	+	0
			100111

MSB: En yüksek değerlikli bit.
(Most Significant Bit)
LSB: En düşük değerlikli bit.
(Least Significant Bit)

Sonuç olarak;

$$(39)_{10} = (100111)_2$$

eşitliği bulunur.

2. Sayı Sistemlerinin Birbirlerine Dönüştürülmeleri

Onlu Sayıların İkili Sayılara Dönüşümü:

Kesirli onlu sayılar ikili sayılara dönüştürülürken kesir kısmı 2 ile çarpılır. Çarpım sonucunda elde edilen sayının tam kısmı kaydedilerek, kesirli kısım 2 ile yeniden çarpılır. Bu işleme kesirli kısım '0' değerine (veya 0'a çok yakın bir değere) ulaşincaya kadar devam edilir.

Örnek 3: $(0.65)_{10}$ sayısını ikili sayı sistemine çevirelim.

	Tam Kısım	
$0.65 * 2 = 1.30$	1	a^{-1}
$0.30 * 2 = 0.60$	0	a^{-2}
$0.60 * 2 = 1.20$	1	a^{-3}
0.20		

Sıralama
yönü

Sonuç;

$$(0.65)_{10} \cong (0.101)_2$$

olarak bulunur. Bu örnekte görüldüğü gibi kesirli kısım 0 değerine varmayabilir. Bu gibi durumlarda işlem sonlandırılarak yuvarlatma yapılabilir.

2. Sayı Sistemlerinin Birbirlerine Dönüştürülmeleri

Onlu Sayıların Sekizli Sayılara Dönüştürülmesi :

Onlu sayı sistemindeki bir sayıyı, sekizli sisteme dönüştürmek için yukarıda açıklanan yöntemler kullanılır.

Örnek 5: $(153)_{10}$ sayısını sekizli sisteme çevirelim.

Verilen sayının devamlı 8 ile bölünmesi ve kalanın yazılması şeklinde işlem yapılır:

<u>İşlem</u>	<u>Bölüm</u>	<u>Kalan</u>
153 / 8	19	1
19 / 8	2	3
2		2

İşlemler sonucunda,

$$(153)_{10} = (231)_8$$

eşitliği bulunur.

Tam sayı ve kesirli kısmı bulunan onlu sayıları 8'li sayılara dönüştürme işleminde; tam sayı ve kesir kısımları ayrı ayrı dönüştürülür ve bulunan sonuçlar birlikte yazılır.

2. Sayı Sistemlerinin Birbirlerine Dönüştürülmeleri

Onlu Sistemdeki Sayıların Onaltılı Sayılara Dönüştürülmesi :

Onlu sistemdeki bir sayıyı onaltılık sisteme dönüştürmek için, onluk sistemin ikili ve sekizli sisteme çevrilmesindeki yöntem uygulanır. Ancak onaltılık sistemde taban '16' olduğundan, 16'ya bölme ve kalanı yazma şeklinde işlem yapılır.

Örnek 7: $(214)_{10}$ sayısını onaltılık sayı sistemine çevirelim.

Verilen sayının devamlı 16'ya bölünmesi ve kalanının yazılması şeklinde işlem yapılır:

<u>İşlem</u>	<u>Bölüm</u>	<u>Kalan</u>	
214 / 16	13	6	→ 6
13 / 16	0	13	→ D ↑

Sonuç olarak;

$$(214)_{10} = (D6)_{16}$$

eşitliği yazılabilir.

2. Sayı Sistemlerinin Birbirlerine Dönüştürülmeleri

2.2.İkili Sayı Sistemindeki Sayıların Onlu, Sekizli ve Onaltılı Sayı Sistemlerine Dönüştürülmesi:

İkili sistemdeki bir sayı, her basamağının ağırlık katsayısı ile çarpılıp, bulunan değerlerin toplanması ile ilgili sayı sistemine dönüştürülür.

İkili Sayıların Onlu Sayılara Dönüştürülmesi:

İkili sistemdeki bir sayı, her basamağının ağırlık katsayısı ile çarpılıp, bulunan değerlerin toplanması ile Onlu sayı sistemine dönüştürülür.

Örnek 11: $(11001)_2$ sayısının onluk sayı sistemindeki karşılığını bulalım.

Her bir basamakta bulunan sayı basamak değeri ile çarpılır ve bulunan sayılar toplanırsa;

$$1 \ 1 \ 0 \ 0 \ 1 \longrightarrow 1x2^4 + 1x2^3 + 0x2^2 + 0x2^1 + 1x2^0 = 16 + 8 + 0 + 0 + 1$$

olur. Bu durumda;

$$(11001)_2 = (25)_{10} = 25$$

eşitliği yazılabilir.

2. Sayı Sistemlerinin Birbirlerine Dönüştürülmeleri

İkili Sayıların Onlu Sayılara Dönüştürülmesi:

Kesirli ikili sayının onluk sayı sistemine dönüştürülmesi; kesirli kısmın soldan sağa doğru ikinin negatif kuvvetleri şeklinde yazılıp, bu sayıların basamaklarda bulunan sayılarla çarpılması ve bulunan çarpımların toplanması şeklinde gerçekleştirilir.

Örnek 12: $(100.01)_2$ sayısını onluk sayı sistemine dönüştürelim.

Tamsayı ve kesirli kısmın basamak değerleri ile basamaklarda bulunan sayılar çarpılırsa;

$$\begin{aligned} 100.01 &= 1.2^2 + 0.2^1 + 0.2^2, 0.2^{-1} + 1.2^{-2} \\ &= 1.4 + 0.2 + 0.1, 0.1/2 + 1.1/4 \\ &= 4 + 0 + 0, 0 + 1/4 \\ &= (4.25)_{10} \end{aligned}$$

sayısı bulunur. Bu durumda;

$$(100.01)_2 = (4.25)_{10}$$

eşitliği elde edilir.

2. Sayı Sistemlerinin Birbirlerine Dönüştürülmeleri

İkili Sayıların Sekizli Sayılara Dönüştürülmesi :

İkili sistemdeki bir sayıyı sekizli sistemde ifade etmek için, ikili sistemdeki sayılar sağdan sola doğru üçerli kümeler halinde ayrılır ve en sondaki kümedeki bitlerin sayısı üçten az ise sola doğru '0' eklenerek üçe tamamlanır.

Örnek 14: $(11001111011101)_2$ sayısını sekizli sayı sistemine dönüştürelim.

Üçerli kümelere ayırma ve eksik bitleri tamamlama sonucunda,

011 001 111 011 101

kümeleri elde edilir. Her kümedeki sayının onluk karşılığı yazılırsa;

$$(011\ 001\ 111\ 011\ 101)_2 = (3\ 1\ 7\ 3\ 5)_8$$

şeklinde sekizli sistemdeki sayı bulunur. Bu durumda,

$$(11001111011101)_2 = (31735)_8$$

eşitliği yazılabilir.

2. Sayı Sistemlerinin Birbirlerine Dönüştürülmeleri

İkili Sayıların Onaltılı Sayılara Dönüştürülmesi::

İkili sayı sisteminden onaltılık sayı sistemine dönüştürme işlemi, ikili sistemdeki sayının dörderli gruplara ayrılıp, her bir gruptaki sayıların karşılıklarının yazılması şeklinde gerçekleştirilir. Gruplama işlemine sağdan başlanır ve en sondaki grup '0' eklenerek dört bite tamamlanır. Gruplardaki sayıların karşılıkları olan sayılar yazılınca, onaltılık sistemdeki sayı elde edilir.

Örnek 17: $(10111101110000111101)_2$ sayısını onaltılık sayı sistemine dönüştürelim.

Verilen sayı dört bitlik gruplar halinde yazılırsa; 1011 1101 1100 0011 1101 şeklini alır. Bu gruplardaki sayıların onaltılık sistemdeki karşılıkları yazılırsa;

1011 1101 1100 0011 1101

B D C 3 D

sayıları elde edilir.

Sonuç olarak;

$$(10111101110000111101)_2 = (BDC3D)_{16}$$

eşitliği bulunur.

2. Sayı Sistemlerinin Birbirlerine Dönüştürülmeleri

İkili Sayıların Onaltılı Sayılara Dönüştürülmesi:

Örnek 18: $(10110001101011.11110010)_2$ sayısını onaltılık sayı sistemine çevirelim.

Çevirme işlemi için önce sayının tam sayı ve kesirli kısımları 4'erli gruplara ayrılır. Herbir grubun onaltılı sistemde karşılığı olan sayı yazılır.

0010	1100	0110	1011	.	1111	0010
2	C	6	B		F	2

Grupların karşılıkları olan sayılar sırası ile yazılınca; onaltılık sistemdeki sayı;

$$(10110001101011.11110010)_2 = (2C6B.F2)_{16}$$

olarak elde edilir.

Örnek 19: $(1100000110.1101100)_2 = (?)_{16}$ dönüşümünü yapalım.

Gruplandırma yapıp, herbir gruptaki sayıların karşılığı yazılırsa;

0011	0000	0110	.	1101	100	=	(306.D8)	₁₆
3	0	6		D	8			

sonucu elde edilir.

2. Sayı Sistemlerinin Birbirlerine Dönüştürülmeleri

2.3.Sekizli Sistemdeki Sayıların İkili, Onlu Ve Onaltılı Sistemlere Dönüştürülmesi

Sekizli Sayıların İkili Sayılara Dönüştürülmesi:

Sekizli sistemdeki bir sayıyı ikili sayı sistemine dönüştürmek için, her bir basamaktaki sayının karşılığı olan ikili sayı 3 bitlik gruplar şeklinde yazılır. Gruplar halinde yazılan ikili sayıların karşılığı olan sayıların bir araya getirilmesi ile ikili sistemdeki sayı ortaya çıkar.

Örnek 20: $(673.124)_8$ sayısını ikili sayı sistemine çevirelim.

Önce her bir sayının karşılığı olan ikili sayı 3 bit olarak yazılır:

$$6=110, 7=111, 3=011, 1=001, 2=010, 4=100.$$

Yazılan sayılar bir araya getirilirse;

$$(673.124)_8 = (110111011.001010100)_2$$

eşitliği bulunur.

2. Sayı Sistemlerinin Birbirlerine Dönüştürülmeleri

2.3.Sekizli Sistemdeki Sayıların İkili, Onlu Ve Onaltılı Sistemlere Dönüştürülmesi

Sekizli Sayıların Onlu Sayılara Dönüştürülmesi :

Sekizli sayılar, her bir basamaktaki rakamın basamak ağırlığıyla çarpılması ve daha sonra çarpımların toplanması yoluyla onluk sayı sistemine dönüştürülür.

Örnek 21: $(372)_8$ sayısını onluk sayı sistemine çevirelim.

Herbir basamaktaki sayı basamak değerleriyle çarpılıp, bulunan sayılar toplanırsa;

$$\begin{aligned}(372)_8 &= 3 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 \\ &= 3 \times 64 + 7 \times 8 + 2 \times 1 \\ &= 250\end{aligned}$$

sayısı bulunur. Bu durumda;

$$(372)_8 = (250)_{10}$$

eşitliği elde edilir.

Örnek 22: $(24.6)_8 = (?)_{10}$ dönüşümünü gerçekleştirelim.

Basamaklardaki sayılar basamak değerleriyle çarpılır:

$$(24.6)_8 = 2 \times 8^1 + 4 \times 8^0 + 6 \times 8^{-1}.$$

Çarpımından bulunan değerler toplanırsa:

$$= 16 + 4 + 0.75 = 20.75$$

sayısı bulunur. Sonuçta;

$$(24.6)_8 = (20.75)_{10}$$

eşitliği oluşur.

2. Sayı Sistemlerinin Birbirlerine Dönüştürülmeleri

2.4.Onaltılık Sistemdeki Sayıların, İkili, Sekizli ve Onlu Sayı Sistemlerine Dönüştürülmesi

Onaltılı Sayıların İkili Sayılara Dönüştürülmesi:

Onaltılı sistemdeki bir sayıyı ikili sayı sistemine dönüştürmek için; her basamaktaki sayının karşılığı olan ikili sayı 4 bit şeklinde yazılır. 4 bitlik gruplar bir araya getirilerek ikili sayı bulunur.

Örnek 24: $(5D1D69)_{16}$ sayısını ikili sisteme çevirelim.

Herbir basamaktaki onaltılık sayının karşılığı olan ikili sayı yazılırsa;

$$5=0101, D=1101, 1=0001, D=1101, 6=0110, 9=1001$$

değerleri elde edilir.

Yazılan ikili sayıların bir araya getirilmesi ile, sonuç olarak;

$$(5D1D69)_{16} = (010111010001110101101001)_2$$

eşitliği bulunur.

Örnek 25: $(E70F.CA)_{16}$ sayısını ikilik sayıya çevirelim. Her bir basamaktaki sayının karşılığı olan ikili sayı 4 bit olarak yazılırsa;

$$1110 \ 0111 \ 0000 \ 1111 \ . \ 1100 \ 1010$$

sayıları bulunur. Bu durumda;

$$(E70F.CA)_{16} = (1110011100001111.11001010)_2$$

eşitliği elde edilir.

2. Sayı Sistemlerinin Birbirlerine Dönüştürülmeleri

2.4. Onaltılık Sistemdeki Sayıların, İkili, Sekizli ve Onlu Sayı Sistemlerine Dönüştürülmesi

Onaltılı Sayıların Onlu Sayılara Dönüştürülmesi:

Onaltılı sayıyı onlu sisteme çevirmek için, her basamaktaki değer ile basamak ağırlığı çarpılır. Bulunan değerlerin toplanması ile onaltılı sistemden onlu sayı sistemine dönüşüm yapılmış olur.

Örnek 26: $(E70FCA)_{16}$ sayısını onlu sisteme dönüştürelim.

Herbir basamaktaki sayıyı basamak değerleriyle çarpıp, bulunan sayıların toplanması ile;

$$\begin{aligned} E70FCA &= E \times 16^5 + 7 \times 16^4 + 0 \times 16^3 + F \times 16^2 + C \times 16^1 + A \times 16^0 \\ &= 1844719 + 458752 + 0 + 3840 + 192 + 10 \\ &= (2307513)_{10} \end{aligned}$$

sayısı bulunur. Sonuçta;

$$(E70FCA)_{16} = (2307513)_{10}$$

Örnek 27: $(5D1.D9)_{16} = (?)_{10}$ dönüşümünü yapalım.

Basamak değerlerinin basamaklardaki sayılarla çarpılıp, bulunan sayıların toplanması ile;

$$\begin{aligned} 5D1.D9 &= 5 \times 16^2 + 13 \times 16^1 + 1 \times 16^0 + 13 \times 1/16 + 9 \times 1/256 \\ &= 1280 + 208 + 16 + 13/16 + 9/256 \\ &= (1504.8476)_{10} \end{aligned}$$

sayısı bulunur. Bu durumda,

$$(5D1.D9)_{16} = (1504.8476)_{10}$$

eşitliği yazılabilir.

2. Sayı Sistemlerinin Birbirlerine Dönüştürülmeleri

2.4.Onaltılık Sistemdeki Sayıların, İkili, Sekizli ve Onlu Sayı Sistemlerine Dönüştürülmesi

Onaltılı Sayıların Sekizli Sayılara Dönüştürülmesi :

Onaltılık sayıyı sekizli sisteme çevirmek için en pratik yöntem; onaltılık sayının ikili sisteme ve daha sonra ikili sistemdeki sayının sekizli sisteme çevrilmesidir.

Örnek 28 : $(E0CA)_{16}$ sayısını sekizli sisteme çevirelim.

Önce onaltılı sayı ikili sisteme çevrilir. Onaltılı sistemdeki sayının ikili sisteme çevrilmesi için, her bir basamaktaki sayının ikili karşılığı dört bitlik olarak yazılırsa;

$$E = 1110, \quad 0 = 0000, \quad C = 1100, \quad A = 1010$$

sayıları bulunur. Bulunan sayılar birleştirilirse;

$$(E0CA)_{16} = (1110000011001010)_2$$

sayısı elde edilir. Elde edilen ikili sayı, her grubun karşılığı olan sekizli sayının üçerli gruplar halinde yazılması şeklinde sekizli sayıya dönüştürülürse;

$$(E0CA)_{16} = (1110000011001010)_2 = (160312)_8$$

eşitliği bulunur.

ÖDEVLER

$(1471)_{10} = (?)_2$ işlemini yapınız.

$(571,571)_{10} = (?)_2$ dönüşümünü yapınız.

$(346,125)_{10} = (?)_8$ işlemini yapınız.

$(145,135)_{10} = (?)_{16}$ işlemini yapınız.

$(1453,1451)_{10} = (?)_{16}$ işlemini yapınız.

$(101101011)_2 = (?)_8$ ve $(1101101.101101)_2 = (?)_8$ dönüşümlerini yapınız.

3. Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi

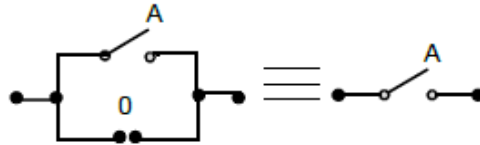
3.1. Önemli Boolean Kuralları

Temel Özellikler:

Boolean cebirindeki temel özellikler : etkisiz eleman, birim eleman, yutan eleman, ters eleman şeklinde sıralanabilir.

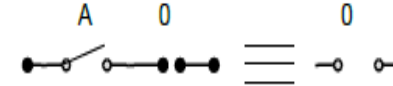
1a : Toplamada Etkisiz Eleman (0) :

$$\begin{aligned}A + 0 &= A \\0 + 0 &= 0 \\1 + 0 &= 1\end{aligned}$$



1d : Çarpmada Yutan eleman:

$$\begin{aligned}A \cdot 0 &= 0 \\0 \cdot 0 &= 0 \\1 \cdot 0 &= 0\end{aligned}$$



1b : Çarpmada Etkisiz Eleman (1) :

$$\begin{aligned}A \cdot 1 &= A \\0 \cdot 1 &= 0 \\1 \cdot 1 &= 1\end{aligned}$$



1e : Ters eleman :

Bir değişken '0' ise değil (barı, tersi vb.) '1', değişken '1' ise değil '0' olarak alınır. Bir değişkenin değil, değişken üzerine konan çizgi veya kesme işareti ile belirtilir.

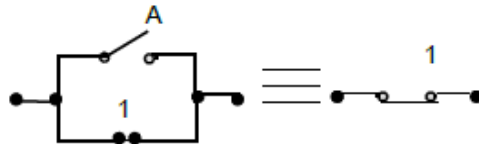
$$A = 0 \Rightarrow A' = 1, \quad A = 1 \Rightarrow A' = 0$$

Bir değişkenin değilinin değil (tersinin tersi) kendisine eşittir :

$$(A'' = A).$$

1c : Toplamada Birim Eleman :

$$\begin{aligned}A + 1 &= 1 \\0 + 1 &= 1 \\1 + 1 &= 1\end{aligned}$$



3. Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi

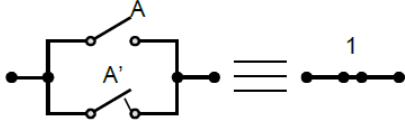
3.1. Önemli Boolean Kuralları

Temel Özellikler:

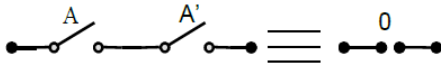
1f : Toplama ve Çarpma İşlemleri :

Boolean matematiğinde, 'VEYA' işlemi toplama (+) ve 'VE' işlemi çarpma (.) işlemlerine karşılık gelir. Boolean matematiğinde geçerli olan toplama ve çarpma işlemleri aşağıdaki şekilde özetlenebilir.

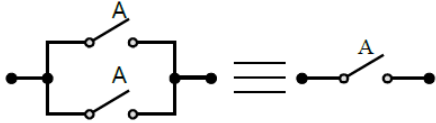
$$\begin{aligned}A + A' &= 1 \\0 + 1 &= 1 \\1 + 0 &= 1\end{aligned}$$



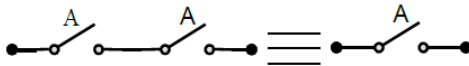
$$\begin{aligned}A \cdot A' &= 0 \\0 \cdot 1 &= 0 \\1 \cdot 0 &= 0\end{aligned}$$



$$\begin{aligned}A + A &= A \\0 + 0 &= 0 \\1 + 1 &= 1\end{aligned}$$



$$\begin{aligned}A \cdot A &= A \\0 \cdot 0 &= 0 \\1 \cdot 1 &= 1\end{aligned}$$



2- Sabit kuvvetlilik :

Boolean matematiğinde normal aritmetik işlemlerdeki toplama ve çarpma işlemlerinden farklı olarak kullanılan kurallardan birisi; sabit kuvvetlilik.

$$\text{a) } A + A = A \quad (A + A + A + \dots + A = A), \quad \text{b) } A \cdot A = A \quad (A \cdot A \cdot A \cdot \dots \cdot A = A)$$

3- Değişim Kanunu (Comutative Law) :

Toplama ve Çarpma işlemlerinde geçerli olan değişim kanunu aynı şekli ile Boolean matematiğinde de geçerlidir.

$$\text{a) } A + B = B + A$$

$$\text{b) } A \cdot B = B \cdot A$$

3. Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi

3.1. Önemli Boolean Kuralları

Temel Özellikler:

4- Birleşme Kanunu (Associative Law) :

Toplama ve Çarpma işlemlerinde geçerli olan değişim kanunu aynı şekli ile Boolean matematiğinde de geçerlidir.

$$a) (A + B) + C = A + (B + C) = A+B+C$$

$$b) (A \cdot B) \cdot C = A \cdot (B \cdot C) = A \cdot B \cdot C$$

5- Dağılma Kanunu (Distributive Law) :

Gerek ‘toplamanın çarpma’ üzerindeki gereksede ‘çarpmanın toplama’ üzerindeki dağılma özellikleri olarak tanımlanan kanunlar, aynı şekli ile boolean matematiğinde kullanılmaktadır.

$$a) A \cdot (B+C) = (A \cdot B) + (A \cdot C)$$

$$b) (A+B) \cdot (A+C) = A + (B \cdot C)$$

6- Yutma Kanunu (Absorption Law) :

Yalnızca Boolean cebirinde geçerli olan kurallardan bir diğeri; yutma kanunudur.

$$a) A + A \cdot B = A$$

$$b) A \cdot (A+B) = A$$

7- Basitleştirme Kanunu (Minimisation Law) :

Toplama ve Çarpma işlemlerinde boolean matematiğinde geçerli olan bir diğeri; basitleştirme ve sadeleştirme kuralıdır.

$$a) A + A' \cdot B = A + B$$

$$b) A \cdot (A'+B) = A \cdot B$$

8- De Morgan Kanunları :

‘VEYADEĞİL’ ve ‘VEDEĞİL’ işlemlerinden faydalanarak uygulanan ve lojik işlemlerde kolaylıklar sağlayan kurallar, ‘**De Morgan Kanunları / Kuralları**’ olarak isimlendirilir

$$a) \overline{A \cdot B} = A' + B'$$

$$b) \overline{A + B} = A' \cdot B'$$

3. Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi

3.2. Boolean Kurallarını Kullanarak Lojik Eşitliklerin Sadeleştirilmesi

Boolean kurallarının lojik ifadelerin basitleştirilmesinde kullanılmasına örnek olması bakımından yukarıdaki bazı eşitlikleri ispatlayalım ve fonksiyon basitleştirme işlemleri yapalım. İşlemlerde değişkenlerin değilini ifade etmek için ' ' işareti kullanılırken, birleşik ifadelerin değilini '—' sembolü kullanılacaktır.

3. Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi

3.2. Boolean Kurallarını Kullanarak Lojik Eşitliklerin Sadeleştirilmesi

Örnek 1: 5 b'nin ispatını yapalım.

$$\begin{aligned}(A+B) \cdot (A+C) &= \underline{A} \cdot A + A \cdot C + B \cdot A + B \cdot C = \underline{A} + \underline{A} \cdot C + \underline{A} \cdot B + B \cdot C \\ &= A \cdot \underline{(1+C+B)} + B \cdot C \\ &= A \cdot 1 + B \cdot C = A + B \cdot C\end{aligned}$$

Örnek 2: 6 a'nın ispatını yapalım.

$$A + A \cdot B = A \cdot \underline{(1+B)} = A \cdot 1 = A \quad (1+B = 1)$$

Örnek 3: 6 b'nin ispatını yapalım.

$$A \cdot (A+B) = \underline{A} \cdot A + A \cdot B = A + A \cdot B = A \cdot \underline{(1+B)} = A \cdot 1 = A$$

Örnek 4: 7 a'nın ispatını yapalım.

$$\begin{aligned}A + A' \cdot B &= \underline{A} + A' \cdot B = A' \cdot (\underline{A} + B) = A' \cdot (\underline{A} + B') \\ &= A' \cdot (\underline{A+B'}) = \underline{A' \cdot A} + A' \cdot B' = A' \cdot B' = A + B\end{aligned}$$

Örnek 5: 7 b'nin ispatını yapalım.

$$A \cdot (A+B) = \underline{A} \cdot A + A \cdot B = 0 + A \cdot B = A \cdot B$$

Örnek 6 : $A+B+C = A' \cdot B' \cdot C'$ 'nin ispatını yapalım.

$$\begin{aligned}\overline{A+B+C} &= \overline{A+X} = A' \cdot X' = A' \cdot (\overline{B+C}) = A' \cdot (B' \cdot C') = A' \cdot B' \cdot C' \\ &\quad \downarrow (B+C=X) \text{ olarak varsayalım.}\end{aligned}$$

Örnek 7 : $F = A' \cdot B + A + A \cdot B$ ifadesini sadeleştirelim.

$$\begin{aligned}A' \cdot B + A + A \cdot B &= A \cdot \underline{(1+B)} + A' \cdot B = \underline{A} + A' \cdot B = A' \cdot (\underline{A+B}) = A' \cdot (\underline{A''+B''}) \\ &= \underline{A' \cdot A''} + A' \cdot B' = \underline{A' \cdot A} + A' \cdot B' \\ &= A' \cdot B' = \underline{A+B}\end{aligned}$$

Sadeleştirme işlemi, 'B' parantezine alınarak yapılır;

$$A'B + A + AB = B \cdot (A + A') + A = B \cdot 1 + A = A+B$$

olarak bulunur. Bu sonuç, aynı sonuca farklı şekillerde ulaşılabacağına iyi bir örnektir.

Örnek 8: $A' \cdot B' \cdot C + A' \cdot B \cdot C + A \cdot B'$ ifadesini sadeleştirelim.

$$A' \cdot B' \cdot C + A' \cdot B \cdot C + A \cdot B' = A' \cdot C \cdot (\underline{B+B'}) + A \cdot B' = A' \cdot C + A \cdot B'$$

3. Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi

3.2. Boolean Kurallarını Kullanarak Lojik Eşitliklerin Sadeleştirilmesi

Örnek 9: $A.B + A'.C + B.C$ ifadesini sadeleştirelim

$$A.B + A'.C + B.C = A.B + A'.C + B.C.(A+A') = \underline{A.B} + \underline{A'.C} + \underline{A.B.C} + \underline{A'.B.C}$$

($A.1=A$ ve $A+A'=1$ olduğundan sonuç değişmez)

$$= A.B.(1+C) + A'.C.(1+B)$$

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

Örnek 10: $A'B'C' + A'B'C + ABC' + AB'C'$ ifadesini sadeleştirelim

$$\underline{A'B'C'} + \underline{A'B'C} + \underline{\underline{ABC'}} + \underline{\underline{AB'C'}} = A'B'.(\underline{C+C'}) + AC'.(\underline{B+B'})$$

1

1

$$= A'B' + AC'$$

3. Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi

3.3. Doğruluk Tablosu

Lojik devrelerde, giriş değişkenlerinin alabilecekleri sayısal değerleri (kombinasyonları) ve sayısal değerlere göre çıkışların durumunu gösteren tablolar, '**doğruluk tablosu**' olarak isimlendirilir. Doğruluk tabloları oluşturulurken, giriş değişken sayısına göre durum ifadesi ortaya çıkar. 'n' tane değişken için 2^n değişik durum oluşur. Örneğin; 2 değişkenli bir ifade için $2^2 = 4$ değişik durum, 3 değişkenli bir ifade için $2^3 = 8$ değişik durum elde edilir.

Örnek 11 : Giriş değişkenlerinin A ve B olduğu bir sistemde $A+B$ işlemi gerçekleştirildiğine göre; A ve B' nin alacağı değerler ile çıkışta oluşacak değerleri tablo halinde gösterelim.

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

Örnek 12 : A ve B giriş değişkenlerine sahip lojik devrenin çıkışı $f=A.B$ eşitliği ile gösterilmektedir. Giriş ve çıkışta oluşabilecek değerleri tablo halinde gösterelim.

A	B	A . B
0	0	0
0	1	0
1	0	0
1	1	1

3. Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi

3.3. Doğruluk Tablosu

Örnek 14 : $\overline{A+B} = A' \cdot B'$ De Morgan teoremini doğruluk tablosu ile ispatlayalım.

Eşitliğin iki tarafındaki işlemleri temsil eden sütunların aynı değerlere sahip olması, doğruluk tablosu yardımı ile eşitliğin doğru olduğunu ispatlar.

A	A'	B	B'	A+B	$\overline{A+B}$	A'·B'
0	1	0	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	1	0	0

↑ ↑

Örnek 15 : $\overline{A \cdot B} = A' + B'$ eşitliğini doğruluk tablosu ile ispatlayalım.

A	A'	B	B'	A·B	$\overline{A \cdot B}$	A'+B'
0	1	0	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	0	1	0	1	0	0

↑ ↑

Örnek 16 : $F = A + A \cdot B = A$ olduğunu ispatlayalım.

A	B	A·B	A + A·B
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

↑ ↑

3. Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi

3.4. Temel Açılımlar ve Standart İfadeler

Daha önceki konularda bahsedildiği üzere, bir binary değişkeni, ya kendi normal formu olan A olarak veya değil olan A' formu ile ifade edilebilir. Bu formlarla ifade edilebilen değişkenler fonksiyon halini aldığı zaman; 'canonical form' (kanun-kaide) olarak adlandırılan 'minterm' (çarpımların toplamı) veya 'maxterm' (toplamların çarpımı) modellerinden biri ile gösterilirler.

3. Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi

3.4. Temel Açılımlar ve Standart İfadeler

Değişken			Mintermler		Maxtermler	
A	B	C	Terim	İsim	Terim	İsim
0	0	0	$A'B'C'$	m_0	$A+B+C$	M_0
0	0	1	$A'B'C$	m_1	$A+B+C'$	M_1
0	1	0	$A'BC'$	m_2	$A+B'+C$	M_2
0	1	1	$A'BC$	m_3	$A+B'+C'$	M_3
1	0	0	$AB'C'$	m_4	$A'+B+C$	M_4
1	0	1	$AB'C$	m_5	$A'+B+C'$	M_5
1	1	0	ABC'	m_6	$A'+B'+C$	M_6
1	1	1	ABC	m_7	$A'+B'+C'$	M_7

Tablo 4.1. Üç değişkenli bir sistemde oluşabilecek minterm ve maxterm terimleri.

3. Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi

3.4. Temel Açılımlar ve Standart İfadeler

Bir boolean ifadede bulunan değişkenlerin sahip olduğu veya oluşturabileceği kombinasyonların 'VE' (çarpım) işlemi sonucunda 1 olacak şekilde uyarlanması (değişkenin değeri 1 ise olduğu gibi alınıp, 0 ise değil ile ifade edilerek), 'minterm' denir. Aynı yolla, değişkenlerin kombinasyonlarının 'VEYA' (toplama) işlemi sonucunda 0 değerini almasını sağlayacak şekilde değişkenlerin şekillendirilmesine 'maxterm' denir. Tablo 4.1'de üç değişkenli bir sistemde değişkenlerin oluşturabileceği kombinasyonlar ve bu kombinasyonlarda elde edilecek minterm ve maxterm terimleri verilmiştir. Bir Boolean fonksiyonu Tablo 4.1'deki doğruluk tablosundan belirli kombinasyonların seçilmesi, seçilen kombinasyonların sonuç olacak şekilde formlandırılması ve formlandırılan kombinasyonların toplanması ('VEYA' işlemine tabi tutulması) şeklinde tanımlanabilir.

3. Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi

3.4. Temel Açılımlar ve Standart İfadeler

Örnek 19 : Tablo 4.2'deki doğruluk tablosunda f_1 ve f_2 fonksiyonlarını minterm formu ile tanımlayalım.

A	B	C	f_1	f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	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

Tablo 4.2. Fonksiyonlardaki minterm oluşturacak kombinasyonların seçilmesi.

3. Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi

3.4. Temel Açılımlar ve Standart İfadeler

'f₁' fonksiyonunda '1' olarak tanımlanan kombinasyonlardaki değişken değerleri; 001, 100 ve 111 olduğundan, bu kombinasyonları temsil eden değişkenler fonksiyon olarak,

$$f_1 = m_1 + m_4 + m_7 = A'B'C + AB'C' + ABC$$

şeklinde ifade edilir.

Aynı şekilde f₂ fonksiyonu;

$$f_2 = m_3 + m_5 + m_6 + m_7 = A'BC + AB'C + ABC' + ABC$$

olarak tanımlanır.

Bu örnek bir Boolean fonksiyonunun mintermlerin toplanması şeklinde tanımlanabileceği özelliğini gösterir. Bu örnekte çıkıştaki '1' değerleri referans olarak alınmıştır.

'f₁' fonksiyonunda '0' olarak tanımlanan kombinasyonlar referans olarak alınır ve kombinasyonlardaki değişkenlerin toplamı '0' olacak şekilde kombinasyonlar formlandırılırsa, Boolean cebirinin diğer bir özelliği ortaya çıkar. Bu özellik; Boolean fonksiyonunun maxtermlerin çarpımı (AND işlemine tabi tutulması) şeklinde ifade edilebilirliği özelliğidir.

Bu özelliği gösterecek şekilde F₁ ve F₂ fonksiyonları yazılabilir. Bu durumda F₁ fonksiyonu;

$$\begin{aligned} F_1 &= M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6 \\ &= (A+B+C) \cdot (A+B'+C) \cdot (A+B'+C') \cdot (A'+B+C') \cdot (A'+B'+C) \end{aligned}$$

şeklinde, F₂ fonksiyonu ise;

$$\begin{aligned} F_2 &= M_0 \cdot M_1 \cdot M_2 \cdot M_4 \\ &= (A+B+C) \cdot (A+B+C') \cdot (A+B'+C) \cdot (A'+B+C) \end{aligned}$$

şeklinde tanımlanır.

3. Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi

3.5. Maxterm ve Minterm İfadelerin Birbirlerine Dönüştürülmesi

‘Minterm’ ve ‘Maxterm’ ifadelerin elde ediliş şekilleri göz önünde tutulursa, minterm ve maxterm ifadelerin birbirlerinin tersi (komplementi-tümleyen) olduğu bulunabilir. Çünkü mintermleri oluşturmak için fonksiyonlardaki '1' değerleri alınırken, maxtermmleri oluşturmak için '0' değerleri alınmaktadır. Örnek olarak;

$$\begin{aligned}f_{(A,B,C)} &= \sum(1,4,5,6,7) \\ &= m_1 + m_4 + m_5 + m_6 + m_7\end{aligned}$$

fonksiyonu incelenirse; bu fonksiyonun tümleyen;

$$\begin{aligned}f'_{(A,B,C)} &= \sum(0,2,3) \\ &= m_0 + m_2 + m_3\end{aligned}$$

3. Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi

3.5. Maxterm ve Minterm İfadelerin Birbirlerine Dönüştürülmesi

olarak oluşur. Bu fonksiyondan f' 'in kendi karşılığı De Morgan kurallarını kullanarak elde edilirse,

$$\begin{aligned} f' = m_0 + m_2 + m_3 &\Rightarrow f = \overline{m_0 + m_2 + m_3} = m_0' . m_2' . m_3' \\ &= M_0 . M_2 . M_3 \\ &= \prod (0, 2, 3) \end{aligned}$$

ifadesi bulunur. $m_i' = M_i$ olduğu doğruluk tablosundan görülebilir. Yine doğruluk tablosundan $M_i' = m_i$ olduğu çıkarılabilir.

Bu açıklamalardan minterm ve maxterm terimleri arasındaki dönüşüm için gerekli işlemleri; ' $\sum$ ' sembolü $\prod$ sembolü ile değiştirilirken, ifade edilen sayılarda bulunan sayılar bulunmayan sayılarla yer değiştirir' şeklinde özetlenebilir.

Yapılan özetlemeyi,

$$F(A,B,C) = \prod(0,2,4,5)$$

fonksiyonuna uygularsak, bu maxterm ifadenin mintermlerle ifadesi;

$$f(A,B,C) = \sum(1,3,6,7)$$

şeklinde oluşur.

3. Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi

3.5. Maxterm ve Minterm İfadelerin Birbirlerine Dönüştürülmesi

Örnek 22 : Aşağıda verilen doğruluk tablosuna göre F fonksiyonunu minterm ve maxterm yöntemlerini kullanarak sadeleştirelim.

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

i- Önce minterm ifadeler yazılır ve yazılan ifadeler sadeleştirilir;

3.5. Maxterm ve Minterm İfadelerin Birbirlerine Dönüştürülmesi

Sadeleştirme sonucunda;

$$\begin{aligned} F(A,B,C) &= \sum (0,3,4,5) = A'B'C' + A'BC + AB'C' + AB'C \\ &= B'C' \cdot (\underbrace{A+A'}_1) + A'BC + AB'C = \underline{B'C'} + A'BC + A\underline{B'C} \\ &= B' \cdot (C' + AC) + A'BC \quad \Rightarrow \quad C' + AC = C' + A \\ &= B'C' + B'A + A'BC \end{aligned}$$

eşitliği elde edilir.

ii- Fonksiyon maxterm şeklinde yazılırsa;

$$F(A,B,C) = \prod (1,2,6,7) = (A+B+C').(A+B'+C).(A'+B'+C).(A'+B'+C')$$

$$= (\underbrace{AA}_{A} + \underbrace{AB'}_{A+AB'=A} + \underbrace{AC}_{A+AC=A} + \underbrace{BA}_{A+BA=A} + \underbrace{BB'}_{A+BC+C'B'} + \underbrace{BC}_{A+BC+C'B'} + \underbrace{C'A}_{A+C'A} + \underbrace{C'B'}_{A+BC+C'B'} + \underbrace{C'C}_{A+BC+C'B'}) \cdot (\underbrace{A'A'}_{A'+A'B'=A'} + \underbrace{A'B'}_{A'+A'B'=A'} + \underbrace{A'C'}_{A'+A'C'=A'} + \underbrace{A'B'}_{A'+A'B'=A'} + \underbrace{B'B'}_{A'+A'B'=A'} + \underbrace{B'C'}_{A'+A'B'=A'} + \underbrace{A'C}_{A'+A'C'=A'} + \underbrace{B'C}_{A'+A'B'=A'} + \underbrace{CC'}_{A'+A'B'=A'})$$

Sadeleştirilmelerin yapılması ile ;

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

ifadesi elde edilir.

ÖDEVLER

$F_{(A,B,C)} = \sum(0,3,5,7)$ minterm ifadesini doğruluk tablosunda gösteriniz ve ifadenin fonksiyonunu yazınız.

$F_{(A,B,C)} = \sum(1,2,6,7)$ minterm ifadesinin fonksiyonunu yazınız.

$F_{(A,B,C)} = \prod(1,4,5)$ maxterm ifadesinin doğruluk tablosunu oluşturun ve maxterm ifadesini değişkenlerle yazınız.

$F_{(A,B,C)} = \prod(2,5,6,7)$ maxterm ifadesinin fonksiyonunu yazınız.

$F_{(A,B,C)} = \sum(2,4,6,7)$ ifadesinin fonksiyonunu yazarak sadeleştiriniz.

$F_{(A,B,C)} = \sum(3,5,7)$ ifadesinin fonksiyonunu yazarak sadeleştiriniz.

$F_{(A,B,C)} = \prod(1,4,5,6)$ ifadesinin fonksiyonunu yazarak sadeleştiriniz.

$F_{(A,B,C)} = \prod(0,5,6)$ ifadesinin fonksiyonunu yazarak sadeleştiriniz.

$F_{(x,y)} = x.y + x'.y + y'$ lojik eşitliğini sadeleştiriniz.

$F_{(A,B,C,D)} = B.C + B.D + A.C + A.D$ lojik eşitliğini sadeleştiriniz.

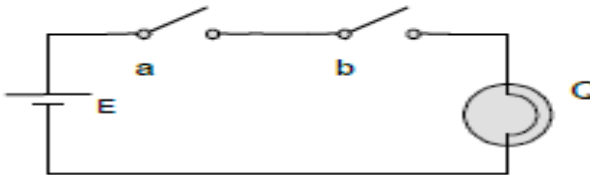
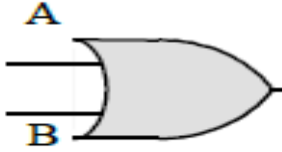
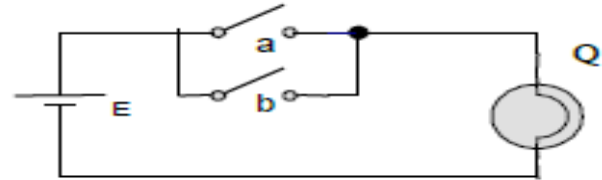
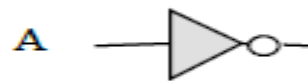
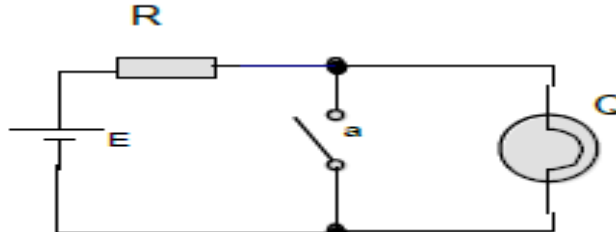
$F_{(x,y)} = x.y + \bar{x}.y + \bar{x}.\bar{y} + x.\bar{y}$ lojik eşitliğini sadeleştiriniz.

$F_{(A,B)} = A.B + \overline{A.B}$ lojik eşitliğini sadeleştiriniz.

4. Lojik Kapılar ve Lojik Devreler

4.1. Lojik Kapılar

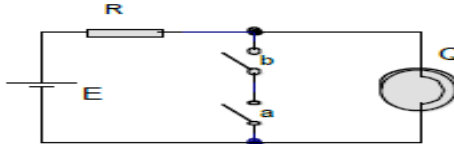
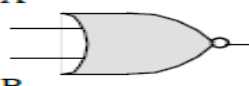
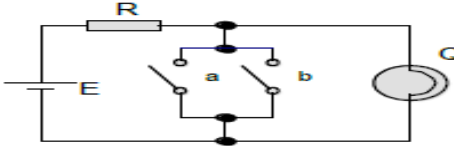
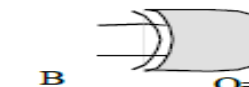
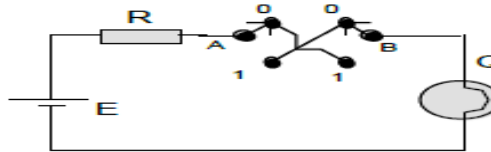
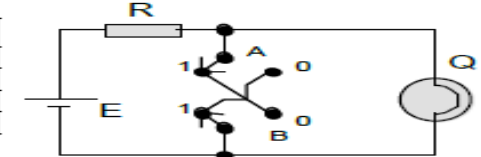
VE (AND), VEYA (OR), DEĞİL (NOT), VEDEĞİL (NAND), VEYADEĞİL (NOR) kapılarıdır ve bu kapılar 'temel lojik kapılar' olarak isimlendirilir.

İşlemin Adı	Sembolü	Yaptığı İşlem	Doğruluk Tablosu	Elektriksel Eşdeğeri															
VE (AND) KAPISI		$Q=A.B$	<table><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Q	0	0	0	0	1	0	1	0	0	1	1	1	
A	B	Q																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
VEYA (OR) KAPISI		$Q=A+B$	<table><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Q	0	0	0	0	1	1	1	0	1	1	1	1	
A	B	Q																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
DEĞİL (NOT) KAPISI		$Q=A'$	<table><tr><th>A</th><th>Q</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	Q	0	1	1	0										
A	Q																		
0	1																		
1	0																		

4. Lojik Kapılar ve Lojik Devreler

4.1. Lojik Kapılar

VE (AND), VEYA (OR), DEĞİL (NOT), VEDEĞİL (NAND), VEYADEĞİL (NOR) kapılarıdır ve bu kapılar 'temel lojik kapılar' olarak isimlendirilir.

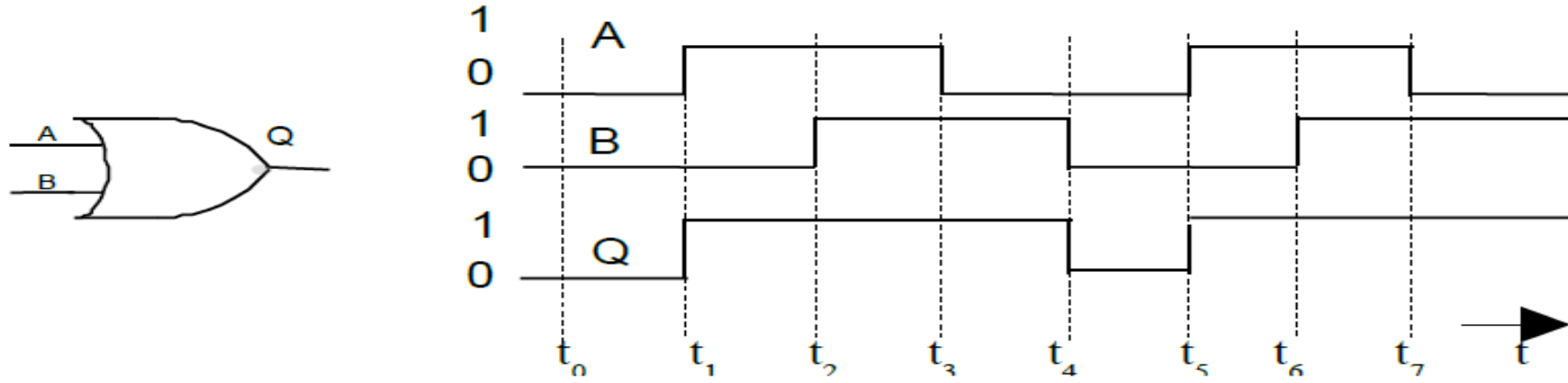
İşlemin Adı	Sembolü	Yaptığı İşlem	Doğruluk Tablosu	Elektriksel Eşdeğeri															
VEDEĞİL (NAND) KAPISI		$Q = A \cdot B$	<table><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Q	0	0	1	0	1	1	1	0	1	1	1	0	
A	B	Q																	
0	0	1																	
0	1	1																	
1	0	1																	
1	1	0																	
VEYADEĞİL (NOR) KAPISI		$Q = A + B$	<table><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Q	0	0	1	0	1	0	1	0	0	1	1	0	
A	B	Q																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	0																	
SÜRÜCÜ (BUFFER) KAPISI		$Q = A$	<table><tr><th>A</th><th>Q</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	A	Q	0	0	1	1										
A	Q																		
0	0																		
1	1																		
ÖZELVEYA (EXOR) KAPISI		$Q = A' \cdot B + A \cdot B'$ $Q = A \oplus B$	<table><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Q	0	0	0	0	1	1	1	0	1	1	1	0	
A	B	Q																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	
ÖZELVEYA DEĞİL (EXNOR) KAPISI		$Q = A \cdot B + A' \cdot B'$ $Q = A \oplus B$	<table><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Q	0	0	1	0	1	0	1	0	0	1	1	1	
A	B	Q																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	1																	

4. Lojik Kapılar ve Lojik Devreler

4.1. Lojik Kapılar

Örnek 2: Şekil 5.4'te verilen A ve B dalga şekilleri iki girişli 'VEYA' kapısına uygulandığında, Q çıkışında oluşacak dalga şeklini çizelim.

'VEYA' kapısında girişlerden birisi '1' oluncaya kadar çıkış '0' olarak kalır (t_1). Girişlerden birisi '1' olan 'VEYA' kapısı çıkışı '1' olur. Girişlerden birinin '1' olması, çıkışın '1' olması için yeterli olduğundan, girişlerden yalnızca birinin '1' olarak kalması çıkışın '1' olarak devam etmesi için yeterli olur. Bu durumda t_1 anında '1' durumuna geçen çıkış dalga şekli t_4 anına kadar '1' olarak devam eder. t_4 - t_5 arasında her iki girişin '0' olması nedeniyle çıkış '0' değerini alır. t_5 anından sonra girişlerden birinin '1' olması nedeniyle çıkış '1' olur ve '1' olarak devam eder.



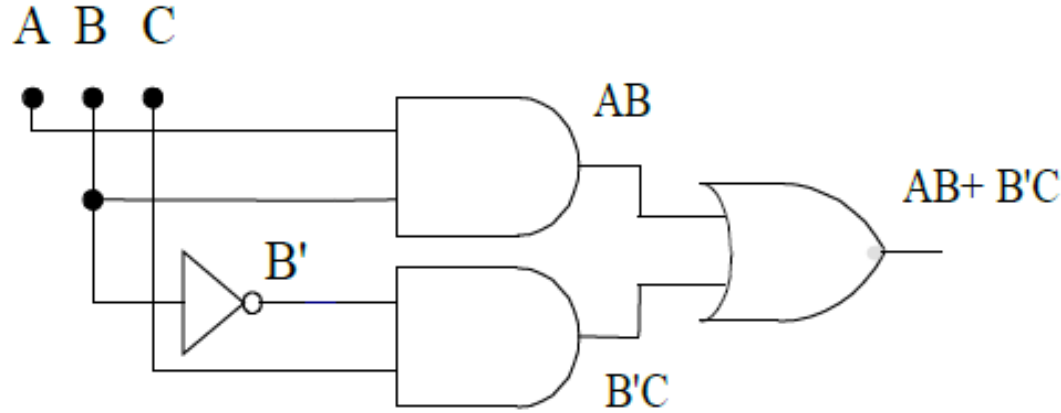
Şekil 5.4. İki girişli 'VEYA' kapısı örnek uygulaması.

4. Lojik Kapılar ve Lojik Devreler

4.2. Lojik İfadelerin Lojik Elemanlar İle Gerçekleştirilmesi

Örnek 15 : $F=AB+B'C$ lojik ifadesini gerçekleştirecek devreyi lojik kapılar ile oluşturalım.

Verilen eşitlikteki işlemlerin gerçekleştirilmesine, çarpma işlemi ile başlanır. Ancak çarpılan değişkenlerden 'DEĞİL' olan varsa, önce 'DEĞİL' kapısı kullanılarak bu işlem gerçekleştirilir. Çarpma işlemi 'VE' kapısı ile gerçekleştirilebileceğinden, çarpılacak ifadeler 'VE' kapısına uygulanır (Şekil 5.30).



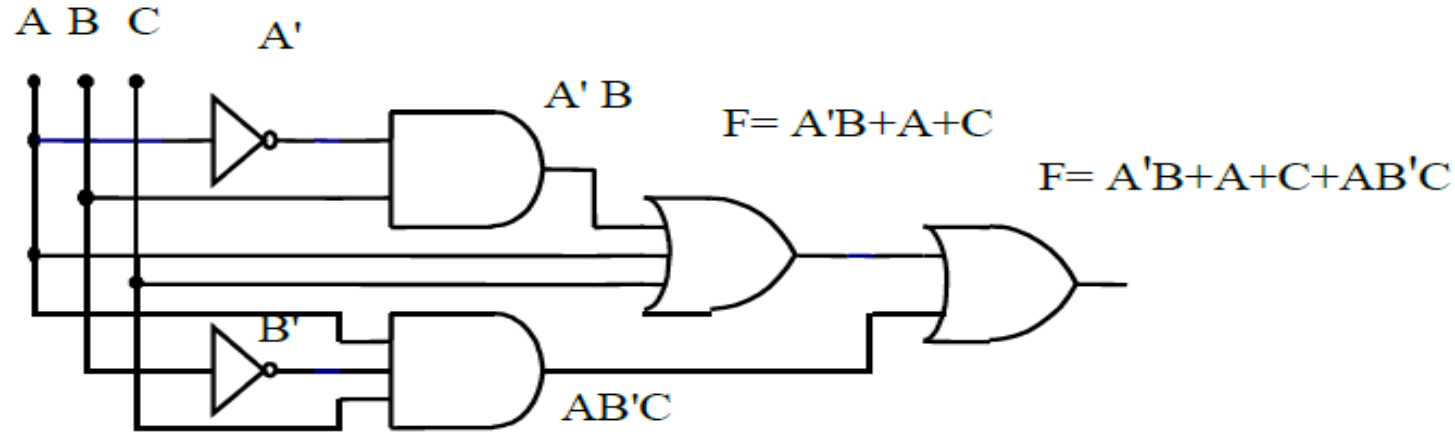
Şekil 5.30. $AB+B'C$ ifadesinin lojik kapılarla oluşturulması.

4. Lojik Kapılar ve Lojik Devreler

4.2. Lojik İfadelerin Lojik Elemanlar İle Gerçekleştirilmesi

Örnek 17 : $F = A'B + A + C + AB'C$ lojik ifadesini kapı devreleri ile gerçekleştirilelim.

Verilen ifadede dört adet minterm ifadesinin toplanması gerektiğinden, toplama işlemleri dört girişli 'VEYA' kapısı kullanılması veya daha az sayıda girişli 'VEYA' kapılarının birlikte kullanılması ile gerçekleştirilir.



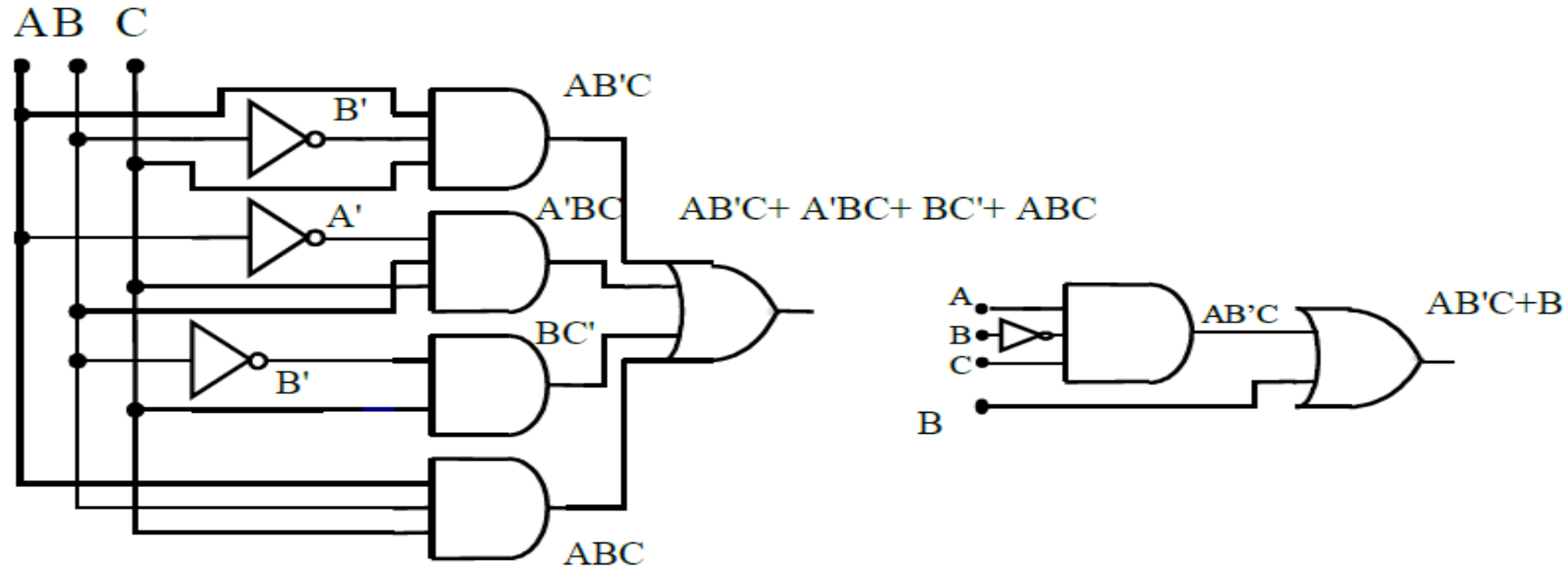
Şekil 5.32. $F = A'B + A + C + AB'C$ ifadesinin kapı devreleriyle gerçekleştirilmesi.

'DEĞİL' kapıları ile tersi alınan değişkenler, 'VE' kapıları ile çarpma işlemine tabi tutulur. Elimizde 2 ve 3 girişli 'VEYA' kapıları bulunduğunu varsayarsak; verilen eşitliği gerçekleştirecek Şekil 5.32'deki lojik devre oluşur.

4. Lojik Kapılar ve Lojik Devreler

4.2. Lojik İfadelerin Lojik Elemanlar İle Gerçekleştirilmesi

Örnek 20 : $F=AB'C+A'BC+BC'+ABC$ ifadesini normal şekli ve Boolean kurallarını kullanarak sadeleştirdikten sonra lojik kapılarla gerçekleştirelim.



Şekil 5.35. Bir lojik eşitliğin sadeleştirilmeden önce ve sadeleştirildikten sonra lojik kapılarla oluşturulması.

4. Lojik Kapılar ve Lojik Devreler

4.2. Lojik İfadelerin Lojik Elemanlar İle Gerçekleştirilmesi

$$\begin{aligned} F &= AB'C + A'BC + BC' + ABC = AB'C + BC(\underline{A' + A}) + BC' \\ &= AB'C + BC + BC' = AB'C + B(\underline{C' + C}) \quad 1 \\ &= AB'C + B \quad 1 \end{aligned}$$

Kapı devreleri ile gerçekleştirilmiş lojik bir devrenin lojik fonksiyonunun çıkarılması ve elde edilen fonksiyonun basitleştirilmesi istenebilir. Bu durumda yapılacak işlem; lojik devreden fonksiyonu yazmak, bulunan fonksiyonu sadeleştirmek ve sadeleştirme sonucunda bulunacak yeni ifadeyi gerçekleştirmek şeklinde olur (Şekil 5.35).

4. Lojik Kapılar ve Lojik Devreler

4.3. Lojik Devrelerin Tasarlanması ve Lojik Elemanlar Kullanılarak Gerçekleştirilmesi

- Lojik devrelerin tasarımında, yapılacak işleme karar verildikten sonra, yapılacak işlemin lojik prensipler ve eşitlikler kullanılarak lojik fonksiyon şekline dönüştürülmesi ve daha sonra kapılar ile gerçekleştirilmesi sırası takip edilir.
- Lojik devre tasarımında yapılacak işlemleri sıralarsak, aşağıdaki işlem sırası oluşur;
- 1. Yapılmak istenen işlem ayrıntıları ile açıklanır.
- 2. Lojik işlemin detayları belirlenir ve doğruluk tablosu haline dönüştürülür.
- 3. Doğruluk tablosu, lojik eşitlik (fonksiyon) şeklinde yazılır.
- 4. Eşitlik, mümkünse sadeleştirme işlemine tabi tutulur.
- 5. Sadeleştirilen lojik ifadeyi gerçekleştirecek lojik devre oluşturulur.

4. Lojik Kapılar ve Lojik Devreler

4.3. Lojik Devrelerin Tasarlanması ve Lojik Elemanlar Kullanılarak Gerçekleştirilmesi

Örnek 24 : K (Kırmızı), Y (Yeşil), S (Sarı) bir trafik işaretindeki lambalar olsun. Bu trafik işaretinde hatalı bileşenleri sezen bir lojik devre tasarlayalım.

Oluşturulacak sistemde aynı anda yalnızca bir lambanın yanması öngörülmektedir. Lamba yanmaması durumu veya aynı anda birden fazla lambanın yanması durumu hata olarak algılanmaktadır.

Hatalı bileşenleri gösteren durumlar tespit edilerek, bu durumları temsil eden bileşenler doğruluk tablosu yardımıyla yazılır. Yazılan fonksiyon kapı devreleri ile gerçekleştirilir (Şekil 5.39).

Doğruluk tablosundan elde edilen kombinasyonların fonksiyon halinde yazılması ile;

$$F = K'.S'.Y' + K'.S.Y + K.S'.Y + K.S.Y' + K.S.Y$$

eşitliği oluşur. Oluşan eşitliğin sadeleştirilmesi ile;

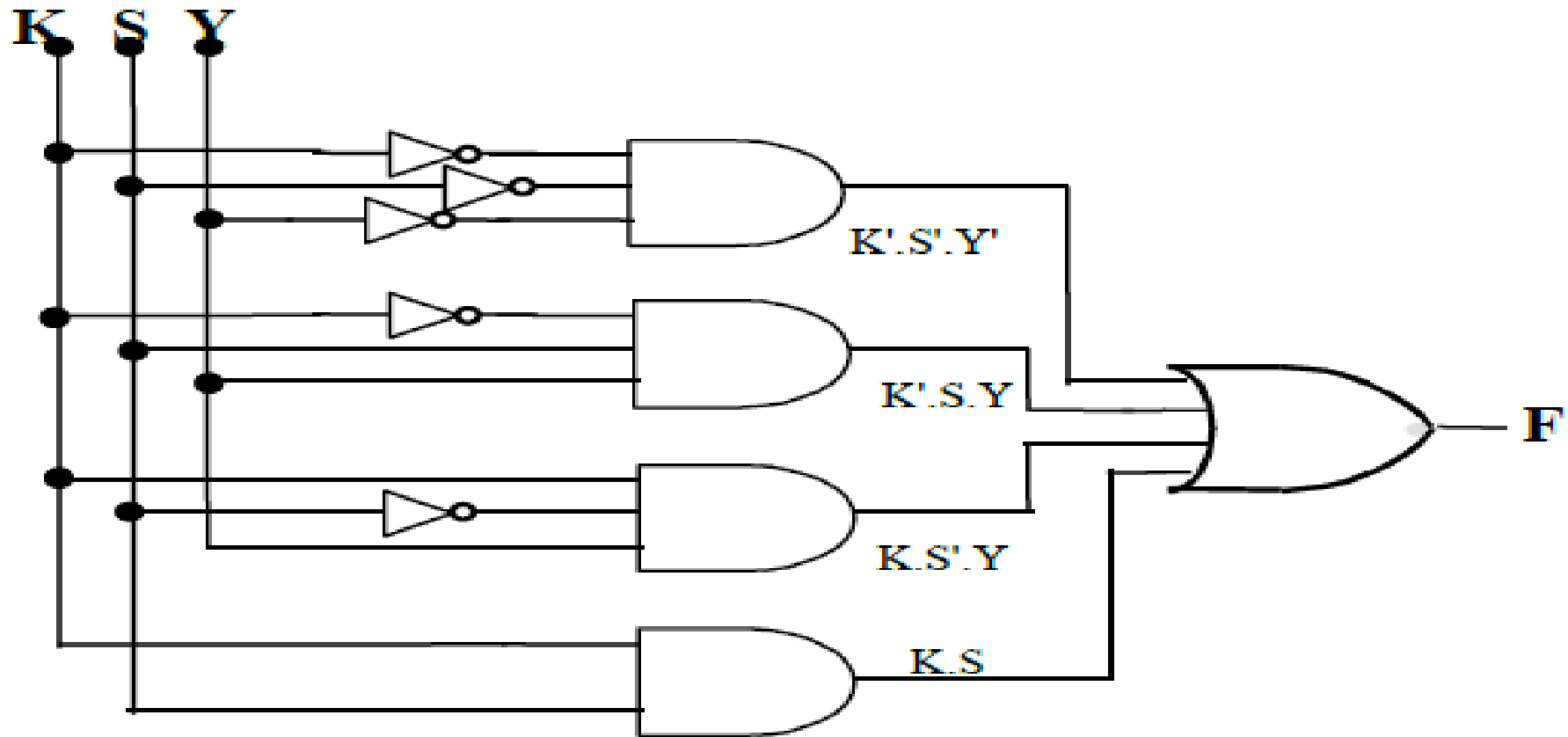
$$F = K'.S'.Y' + K'.S.Y + K.S'.Y + K.S.Y' + K.S.Y = K'.S'.Y' + K'.S.Y + K.S'.Y + K.S$$

$$\underbrace{KS(Y+Y')}_1$$

sonucu bulunur.

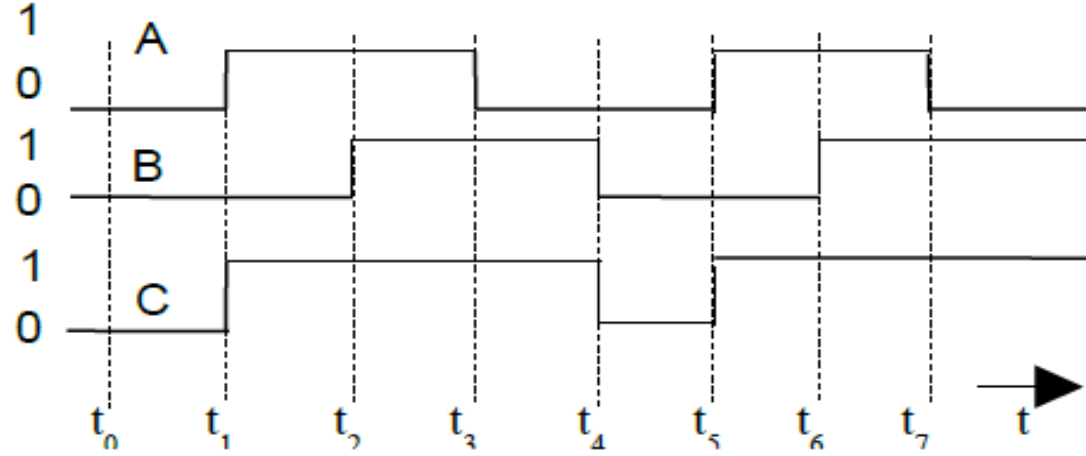
4. Lojik Kapılar ve Lojik Devreler

4.3. Lojik Devrelerin Tasarlanması ve Lojik Elemanlar Kullanılarak Gerçekleştirilmesi



ÖDEVLER

Şekildeki sinyallerin 3 girişli 'VEDEĞİL' kapısına uygulanması durumunda oluşacak dalga şeklini çiziniz.



$F = ABCD' + A'BC'D + A'B'CD + A'BCD + AB'C'D'$ fonksiyonunu kapı devreleri ile gerçekleştiriniz.

$F = A'B'C'D' + A'BC'D + A'B'CD + A'BC'D$ fonksiyonunu kapı devreleri ile gerçekleştiriniz.

$F = (A+B'+C').(A'+B'+C).(A'+B+C)$ ifadesini lojik kapılar ile gerçekleştiriniz.

5. Karnough Haritaları

Lojik ifadelerin sadeleştirilmesinde Boolean Matematiği ve Karnaugh Haritaları kullanılır.

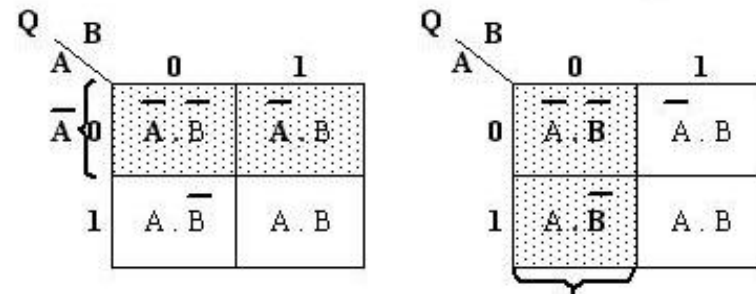
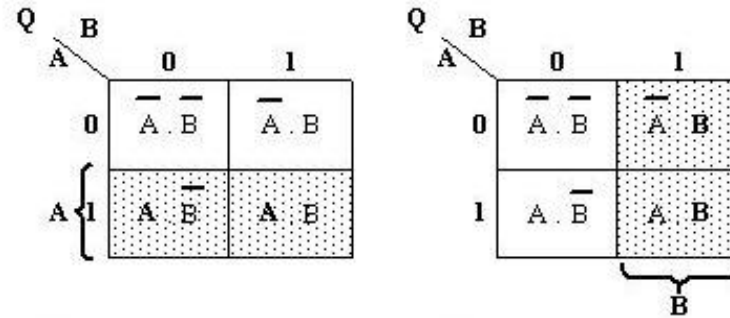
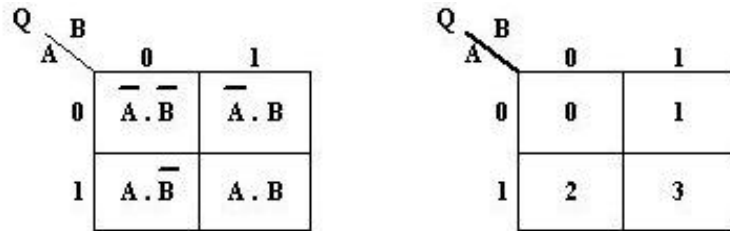
5.1. Karnough Haritalarının Kuralları:

- 1- Karnaugh Haritaları giriş değişkeni sayısına bağlı olarak standart sayıda kutudan oluşur. n =giriş değişkeni sayısı olmak üzere 2^n formülüyle kutu sayısı belirlenir. 2,4,8,16... olmak üzere 2'ye katlanarak devam eder.
- 2- Karnaugh Haritalarında hedef en çok "1" i gruplamaktır. Kutuların içindeki "1" ler dikkate alınır. Boş olan kutu "0" demektir, dikkate alınmaz.
- 3- Gruplamalardaki kutu sayısı 1,2,4,8,16.... Şeklinde olmalıdır.
- 4- Her bir grup çıkış ifadesinde giriş değişkenleri çarpım (AND) şeklinde ifade edilir. Birden fazla gruba sahip Karnaugh Haritasının çıkış ifadesinde gruplar toplama (OR) işlemine tabi tutulur.
- 5- Karnaugh Haritasında tüm kutular "1" ise çıkış "1" , tüm kutular "0" ise çıkış "0" dir.

5. Karnaugh Haritaları

5.1. İKİ DEĞİŞKENLİ KARNAUGH HARİTALARI:

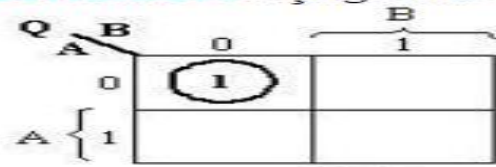
İki değişkenli Karnaugh Haritasında kutu sayısı $2^n = 2^2 = 4$ tür.



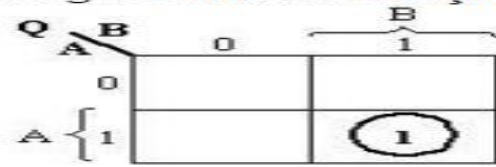
5. Karnough Haritaları

5.1. İKİ DEĞİŞKENLİ KARNAUGH HARİTALARI:

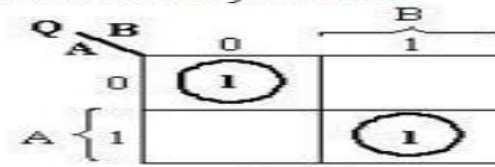
Örnekler: Aşağıdaki Karnough Haritalarının çıkış ifadelerini yazınız.



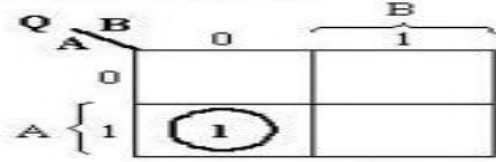
$$Q = \bar{A} \cdot \bar{B}$$



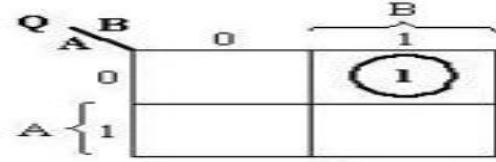
$$Q = A \cdot B$$



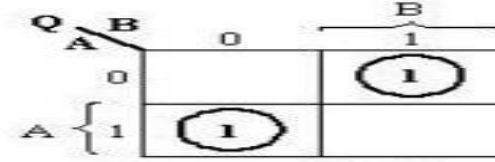
$$Q = \bar{A} \cdot \bar{B} + A \cdot B$$



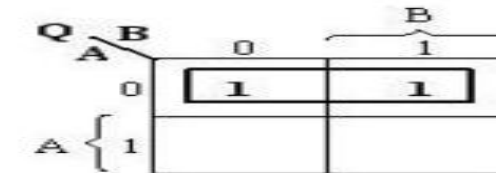
$$Q = A \cdot \bar{B}$$



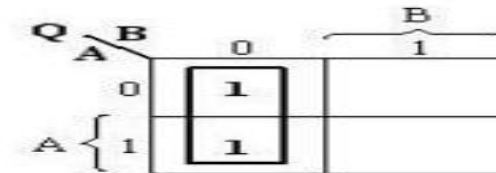
$$Q = \bar{A} \cdot B$$



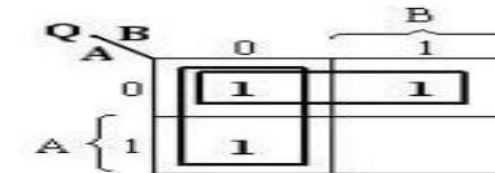
$$Q = A \cdot \bar{B} + \bar{A} \cdot B$$



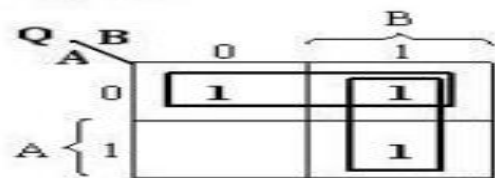
$$Q = \bar{A}$$



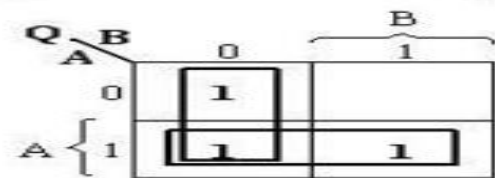
$$Q = \bar{B}$$



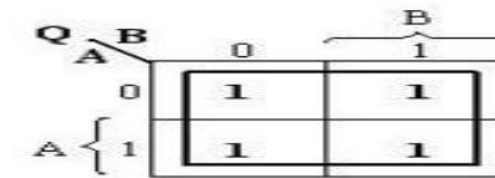
$$Q = \bar{A} + \bar{B}$$



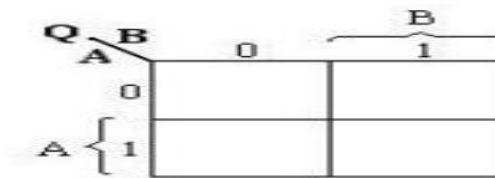
$$Q = \bar{A} + B$$



$$Q = A + \bar{B}$$



$$Q = 1$$



$$Q = 0$$

5. Karnough Haritaları

5.2. ÜÇ DEĞİŞKENLİ KARNAUGH HARİTALARI:

Üç değişkenli Karnaugh Haritasında kutu sayısı $2^n=2^3=8$ tür.

Q A	BC			
	00	01	11	10
0	$\overline{A}\overline{B}\overline{C}$	$\overline{A}\overline{B}C$	$\overline{A}B\overline{C}$	$\overline{A}BC$
1	$A\overline{B}\overline{C}$	$A\overline{B}C$	$AB\overline{C}$	ABC

Q A	BC			
	00	01	11	10
0	0	1	3	2
1	4	5	7	6

Q A	BC			
	00	01	11	10
0	$\overline{A}\overline{B}\overline{C}$	$\overline{A}\overline{B}C$	$\overline{A}B\overline{C}$	$\overline{A}BC$
1	$A\overline{B}\overline{C}$	$A\overline{B}C$	$AB\overline{C}$	ABC

Q A	BC			
	00	01	11	10
0	$\overline{A}\overline{B}\overline{C}$	$\overline{A}\overline{B}C$	$\overline{A}B\overline{C}$	$\overline{A}BC$
1	$A\overline{B}\overline{C}$	$A\overline{B}C$	$AB\overline{C}$	ABC

Q A	BC			
	00	01	11	10
0	$\overline{A}\overline{B}\overline{C}$	$\overline{A}\overline{B}C$	$\overline{A}B\overline{C}$	$\overline{A}BC$
1	$A\overline{B}\overline{C}$	$A\overline{B}C$	$AB\overline{C}$	ABC

B

Q A	BC			
	00	01	11	10
0	$\overline{A}\overline{B}\overline{C}$	$\overline{A}\overline{B}C$	$\overline{A}B\overline{C}$	$\overline{A}BC$
1	$A\overline{B}\overline{C}$	$A\overline{B}C$	$AB\overline{C}$	ABC

B

Q A	BC			
	00	01	11	10
0	$\overline{A}\overline{B}\overline{C}$	$\overline{A}\overline{B}C$	$\overline{A}B\overline{C}$	$\overline{A}BC$
1	$A\overline{B}\overline{C}$	$A\overline{B}C$	$AB\overline{C}$	ABC

C

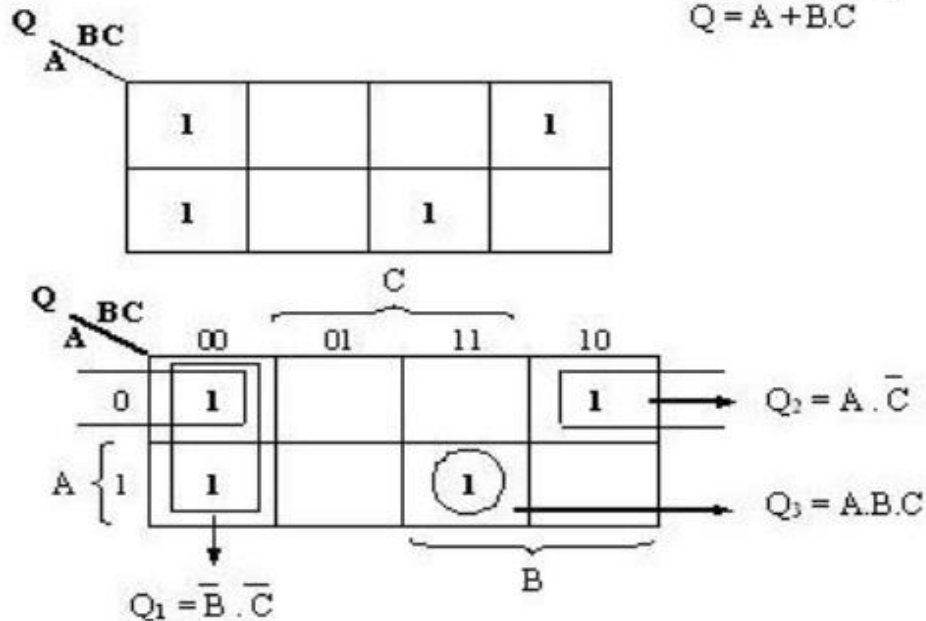
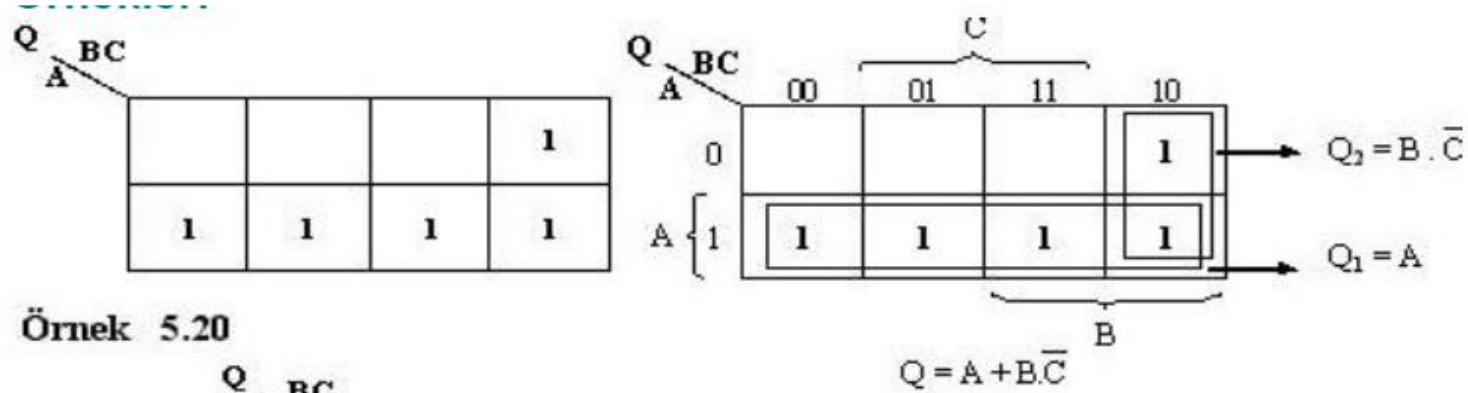
Q A	BC			
	00	01	11	10
0	$\overline{A}\overline{B}\overline{C}$	$\overline{A}\overline{B}C$	$\overline{A}B\overline{C}$	$\overline{A}BC$
1	$A\overline{B}\overline{C}$	$A\overline{B}C$	$AB\overline{C}$	ABC

C

5. Karnaugh Haritaları

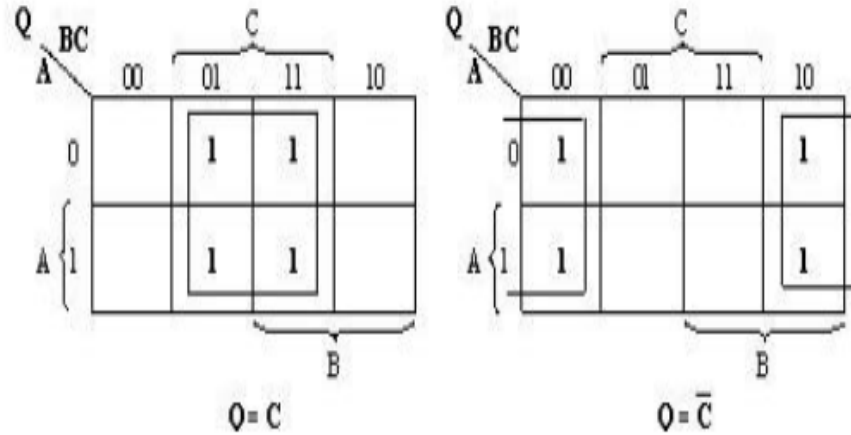
5.2. ÜÇ DEĞİŞKENLİ KARNAUGH HARİTALARI:

Üç değişkenli Karnaugh Haritasında kutu sayısı $2^n=2^3=8$ tür.



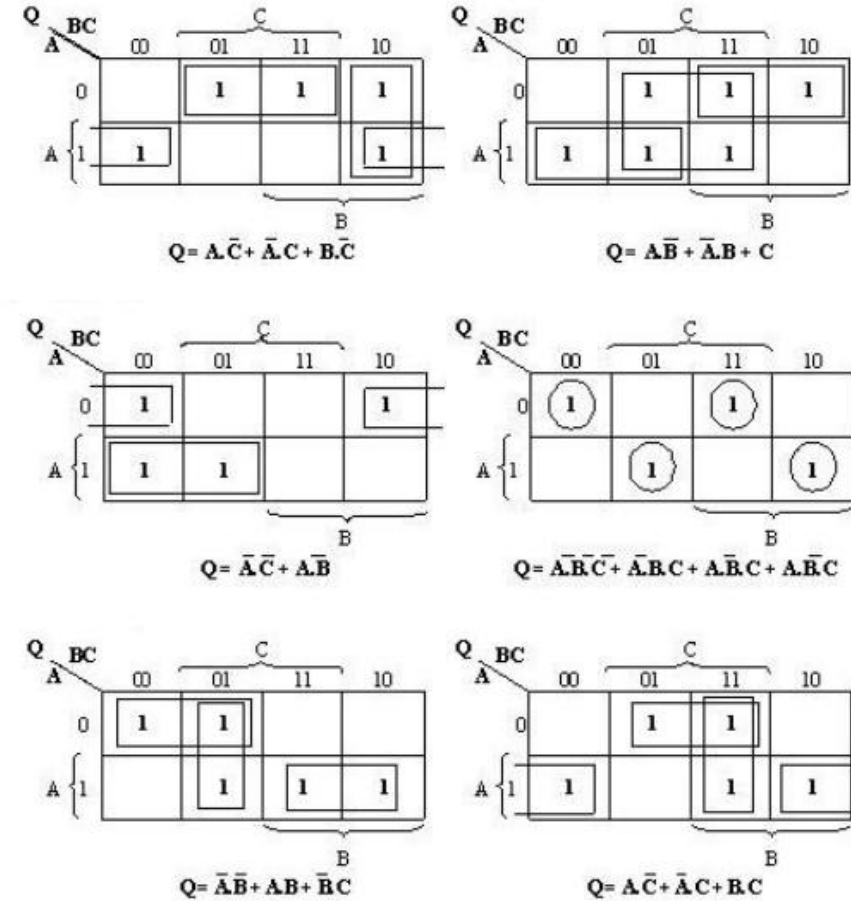
5. Karnough Haritaları

5.2. ÜÇ DEĞİŞKENLİ KARNAUGH HARİTALARI:



Yukardaki örnekte A bölgesi grup içindeki "1" lerin sadece ilkini kapsıyor, B bölgesi de aynı şekilde sadece ikisini kapsıyor ama C bölgesi bütün hepsini kapsadığı için çıkış ifadesinde C yer alır.

Bu örnekte ise A bölgesi grup içindeki "1" lerin sadece ilkini kapsıyor, B bölgesi de aynı şekilde sadece ikisini kapsıyor ama $\bar{C}$ bölgesi bütün hepsini kapsadığı için çıkış ifadesinde $\bar{C}$ yer alır.



5. Karnaugh Haritaları

5.3. DÖRT DEĞİŞKENLİ KARNAUGH HARİTALARI:

Dört değişkenli Karnaugh Haritasında kutu sayısı $2^n = 2^4 = 16$ dir.

Q A B		CD			
		00	01	11	10
00	00	$\bar{A}.\bar{B}.\bar{C}.\bar{D}$	$\bar{A}.\bar{B}.\bar{C}.D$	$\bar{A}.\bar{B}.C.\bar{D}$	$\bar{A}.\bar{B}.C.D$
	01	$\bar{A}.B.\bar{C}.\bar{D}$	$\bar{A}.B.\bar{C}.D$	$\bar{A}.B.C.\bar{D}$	$\bar{A}.B.C.D$
	11	$A.\bar{B}.\bar{C}.\bar{D}$	$A.\bar{B}.\bar{C}.D$	$A.\bar{B}.C.\bar{D}$	$A.\bar{B}.C.D$
	10	$A.B.\bar{C}.\bar{D}$	$A.B.\bar{C}.D$	$A.B.C.\bar{D}$	$A.B.C.D$

Q A B		CD			
		00	01	11	10
00	00	0	1	3	2
	01	4	5	7	6
	11	12	13	15	14
	10	8	9	11	10

5. Karnough Haritaları

5.3. DÖRT DEĞİŞKENLİ KARNAUGH HARİTALARI:

Q A B		CD			
		00	01	11	10
Q	00	$\overline{A}.\overline{B}.\overline{C}.\overline{D}$	$\overline{A}.\overline{B}.C.\overline{D}$	$\overline{A}.B.\overline{C}.\overline{D}$	$\overline{A}.B.C.\overline{D}$
	01	$\overline{A}.B.\overline{C}.\overline{D}$	$\overline{A}.B.C.\overline{D}$	$\overline{A}.\overline{B}.C.D$	$\overline{A}.\overline{B}.C.D$
	11	$A.\overline{B}.\overline{C}.\overline{D}$	$A.\overline{B}.C.\overline{D}$	$A.B.\overline{C}.\overline{D}$	$A.B.C.\overline{D}$
	10	$A.\overline{B}.\overline{C}.D$	$A.\overline{B}.C.D$	$A.B.\overline{C}.D$	$A.B.C.D$

Q A B		CD			
		00	01	11	10
00	0	1	3	2	
01	4	5	7	6	
11	12	13	15	14	
10	8	9	11	10	

5.3. DÖRT DEĞİŞKENLİ KARNAUGH HARİTALARI:

Q A B		C D			
		00	01	11	10
Q A B	00	$\overline{A} \overline{B} \overline{C} \overline{D}$	$\overline{A} \overline{B} \overline{C} D$	$\overline{A} \overline{B} C D$	$\overline{A} \overline{B} C \overline{D}$
	01	$\overline{A} \overline{B} \overline{C} \overline{D}$	$\overline{A} \overline{B} \overline{C} D$	$\overline{A} \overline{B} C D$	$\overline{A} \overline{B} C \overline{D}$
	11	$\overline{A} \overline{B} \overline{C} \overline{D}$	$\overline{A} \overline{B} \overline{C} D$	$\overline{A} \overline{B} C D$	$\overline{A} \overline{B} C \overline{D}$
	10	$\overline{A} \overline{B} \overline{C} \overline{D}$	$\overline{A} \overline{B} \overline{C} D$	$\overline{A} \overline{B} C D$	$\overline{A} \overline{B} C \overline{D}$
		$\overline{C}$			

5. Karnough Haritaları

5.3. DÖRT DEĞİŞKENLİ KARNAUGH HARİTALARI:

Q

CD

AB

1		1	1
	1		
	1		
1		1	1

Q

CD

AB

00	01	11	10
00	1		1
01		1	
11		1	
10	1		1

$Q_1 = \bar{B} \cdot \bar{D}$

$Q_2 = \bar{B} \cdot C$

$Q_3 = B \cdot \bar{C} \cdot D$

$Q = \bar{B} \cdot \bar{D} + \bar{B} \cdot D + B \cdot \bar{C} \cdot \bar{D}$

Q

CD

AB

00	01	11	10
00	1		1
01		1	
11		1	
10	1		1

$Q = \bar{B} \cdot \bar{D} + B \cdot C \cdot D$

Q

CD

AB

00	01	11	10
00	1		1
01	1	1	
11	1	1	
10	1		1

$Q = \bar{C} \cdot \bar{D} + B \cdot \bar{C} + \bar{B} \cdot C \cdot D$

Q

CD

AB

00	01	11	10
00		1	
01	1		1
11	1	1	1
10		1	

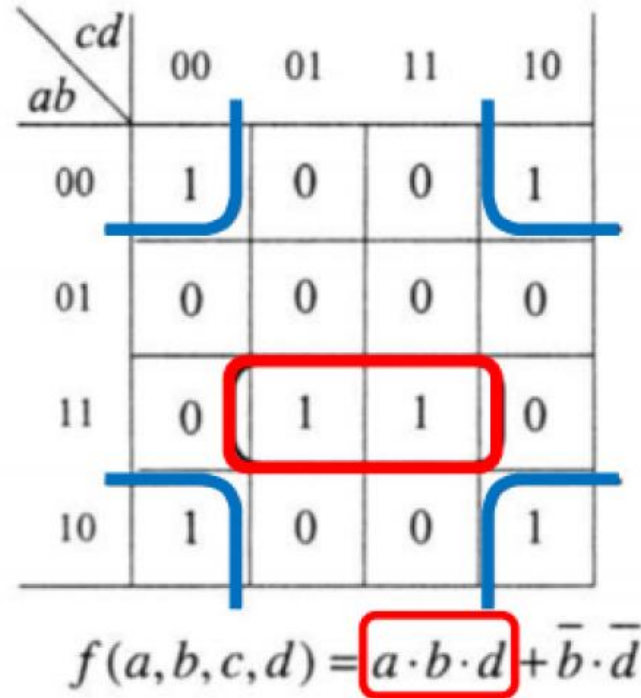
$Q = B \cdot C + B \cdot \bar{D} + \bar{A} \cdot C \cdot D + \bar{B} \cdot \bar{C} \cdot D$

5. Karnough Haritaları

5.4.MİNTERM BİÇİMİ İNDİRGEME:

$$f(a,b,c,d) = \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot \bar{b} \cdot c \cdot \bar{d} + a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} + a \cdot \bar{b} \cdot c \cdot \bar{d} + a \cdot b \cdot \bar{c} \cdot d + a \cdot b \cdot c \cdot d$$

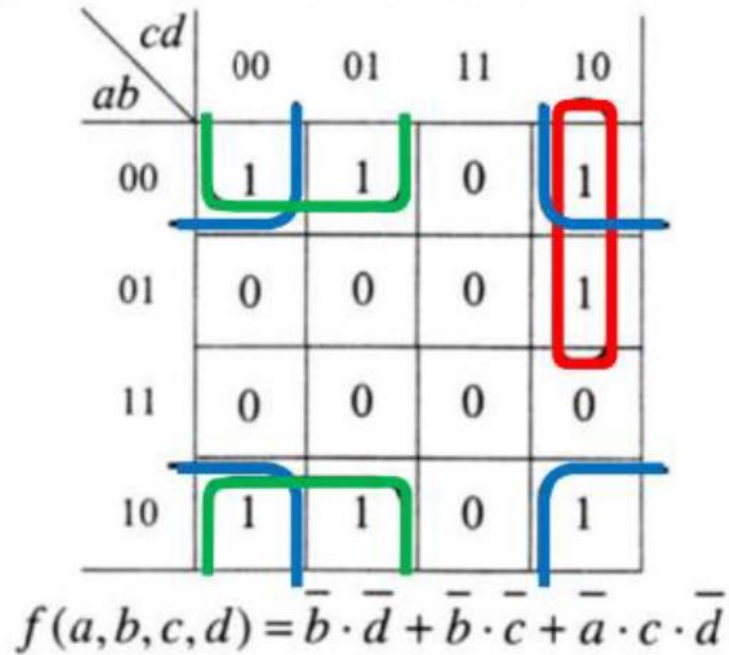
<i>m</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>f(a,b,c,d)</i>
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	1



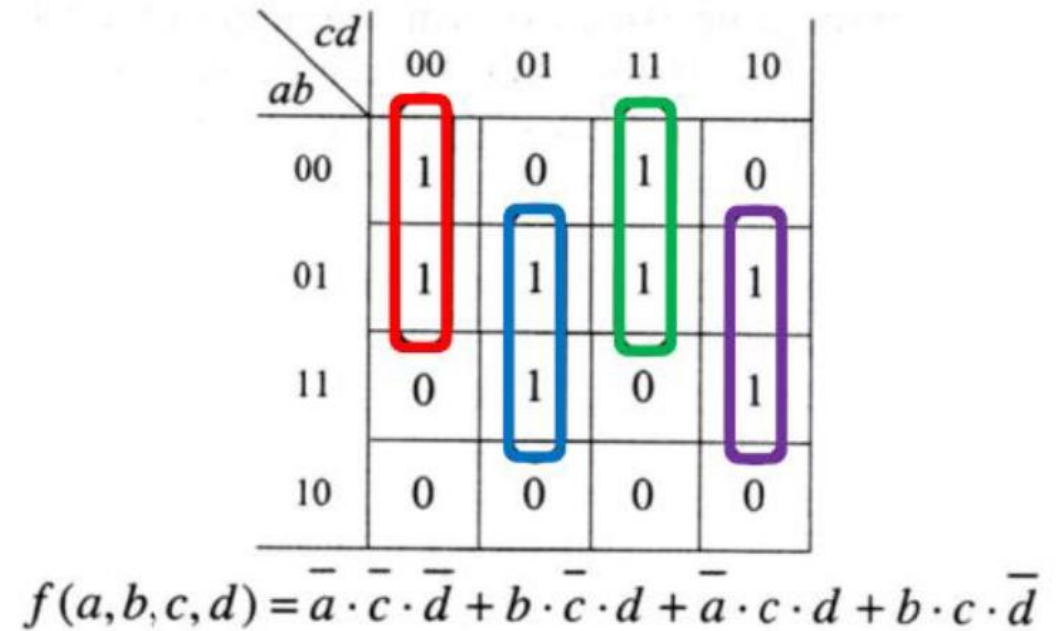
5. Karnough Haritaları

5.4.MİNTERM BİÇİMİ İNDİRGE:

$$f(a,b,c,d) = \bar{a} \cdot \bar{b} \cdot \bar{c} + \bar{b} \cdot c \cdot \bar{d} + \bar{a} \cdot b \cdot c \cdot \bar{d} + a \cdot \bar{b} \cdot \bar{c}$$



$$f(a,b,c,d) = \sum (0,3,4,5,6,7,13,14)$$



5. Karnough Haritaları

5.5.MAXTERM BİÇİMİ İNDİRGE:

Maksterm biçiminde indirgeme

$c+d$ $c+d$ $\bar{c}+\bar{d}$ $\bar{c}+\bar{d}$

c

$ab \backslash cd$	00	01	11	10
$a+b$ 00	1	1	0	1
$a+b$ 01	0	0	0	1
$\bar{a}+\bar{b}$ 11	0	0	0	0
$\bar{a}+b$ 10	1	1	0	1

$f(a,b,c,d) = [\bar{c} + \bar{d}] \cdot [\bar{a} + \bar{b}] \cdot [c + \bar{b}]$

5. Karnough Haritaları

w	x	y	z	$F(w,x,y,z)$
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Örnek: Yandaki doğruluk tablosuna ait Boole fonksiyonunu a) Çarpımlar toplamı biçiminde, b) Toplamlar çarpımı biçiminde olacak şekilde minimal fonksiyonu bulunuz.

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

$$f(x,y,z) = \bar{x}.\bar{z} + y.\bar{z} + \bar{w}.x.\bar{y}.z$$

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

$$f(x,y,z) = (\bar{y} + \bar{z}).(\bar{w} + \bar{z}).(x + \bar{z}).(\bar{x} + y + z)$$

5. Karnough Haritaları

5.4.KEYFİ ÇIKIŞLAR :

$$f(a,b,c,d) = \sum (3,7,11,12,15) + \sum_{\phi} (0,10,13,14)$$

$ab \backslash cd$	00	01	11	10
00	$\phi=0$	0	1	0
01	0	0	1	0
11	1	$\phi=1$	1	$\phi=1$
10	0	0	1	$\phi=0$

$ab \backslash cd$	00	01	11	10
00	$\phi=0$	0	1	0
01	0	0	1	0
11	1	$\phi=1$	1	$\phi=1$
10	0	0	1	$\phi=0$

$$f(a,b,c,d) = c \cdot d + a \cdot b$$

5. Karnaugh Haritaları

ÖRNEK:

$f(a,b,c,d) = \prod M(0, 1, 2, 5, 8, 10, 13, 15) \cdot \prod \phi(6, 7, 9)$ fonksiyonunun

- a) Doğruluk tablosunu yazınız.
- b) Karnaugh diyagramını çiziniz.
- c) Çarpımlar toplamı şeklinde minimal fonksiyonunu bulunuz ve elde ettiğiniz minimal fonksiyonu ilişkin lojik devreyi lojik kapıları kullanarak çiziniz.
- d) Toplamlar çarpımı şeklinde minimal fonksiyonunu bulunuz ve devresini çiziniz.

5. Karnough Haritaları

ÖRNEK:

a)

a	b	c	d	F(a,b,c,d)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	φ
0	1	1	1	φ
1	0	0	0	0
1	0	0	1	φ
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

$$f(a,b,c,d) = \prod M(0, 1, 2, 5, 8, 10, 13, 15) \cdot \prod \phi(6, 7, 9)$$

b)

ab \ cd	00	01	11	10
00	0	0	1	0
01	1	0	φ	φ
11	1	0	0	1
10	0	φ	1	0

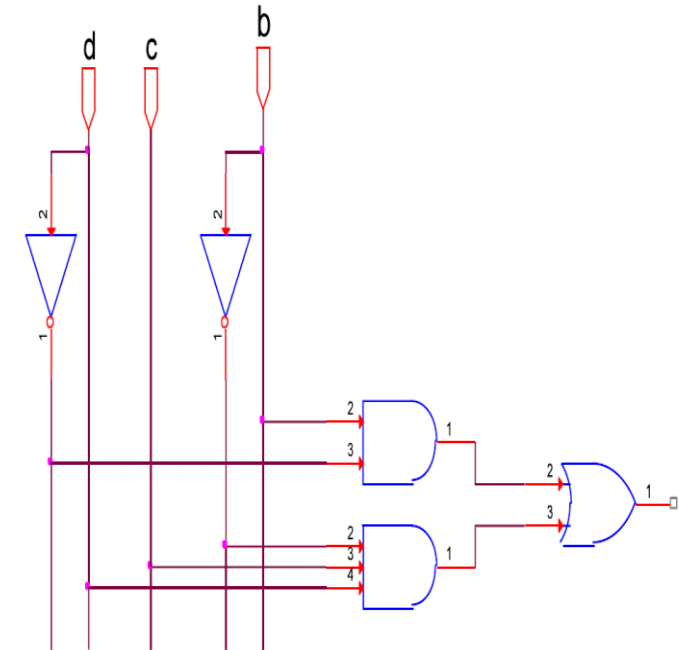
c)

ab \ cd	00	01	11	10
00	0	0	1	0
01	1	0	φ	φ
11	1	0	0	1
10	0	φ	1	0

$$f = b \cdot \bar{d} + \bar{b} \cdot c \cdot d$$

$$f = b \cdot \bar{d} + \bar{b} \cdot c \cdot d$$

c)

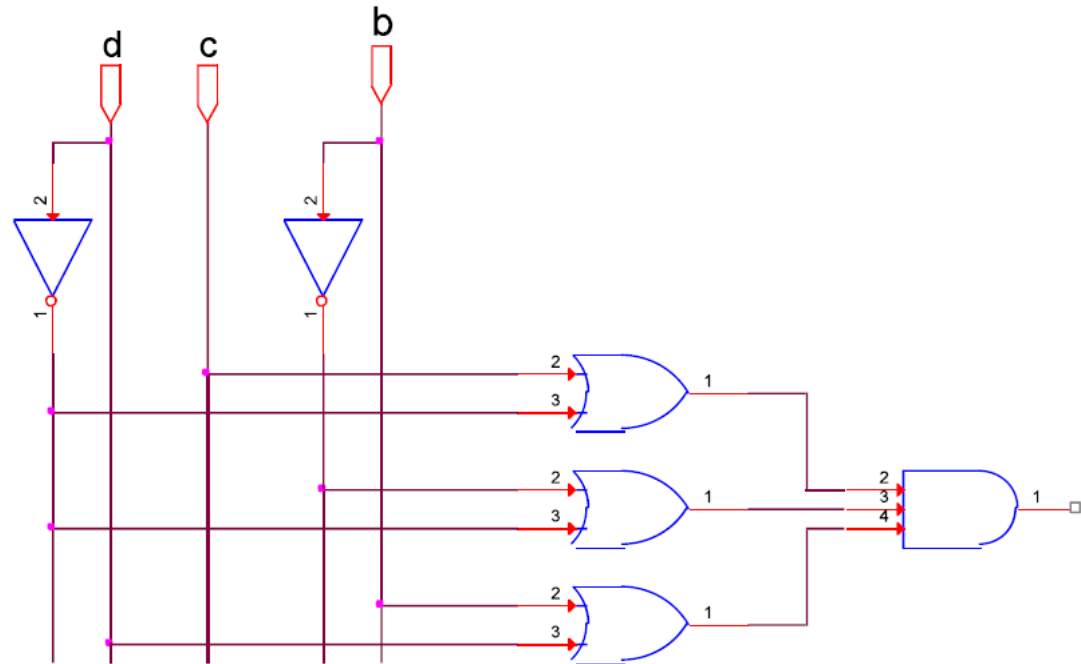


5. Karnough Haritaları

d)

<i>ab</i> \ <i>cd</i>	00	01	11	10
00	0	0	1	0
01	1	0	φ	φ
11	1	0	0	1
10	0	φ	1	0

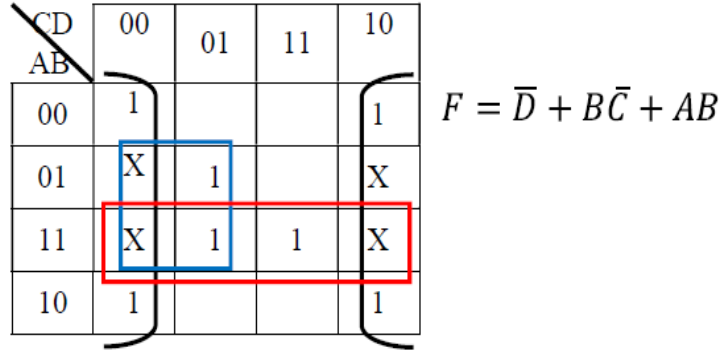
$$f = (c + \bar{d}).(\bar{b} + \bar{d}).(b + d)$$



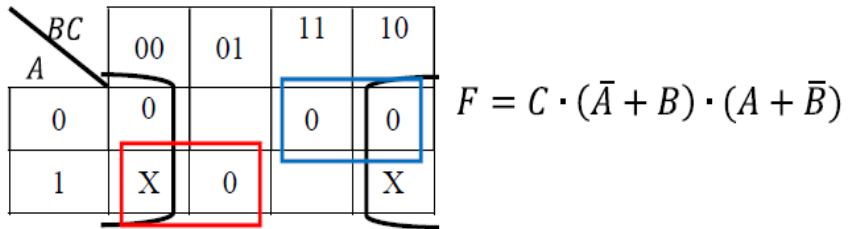
5. Karnough Haritaları

ÖRNEKLER

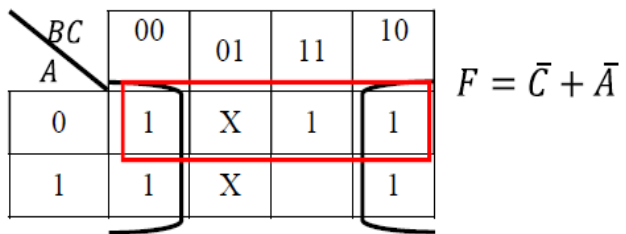
•



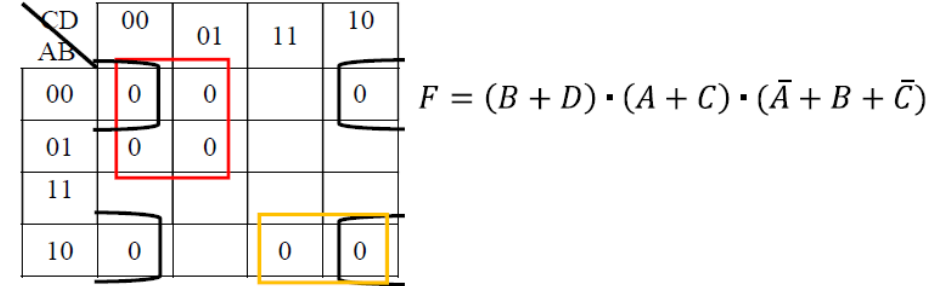
•



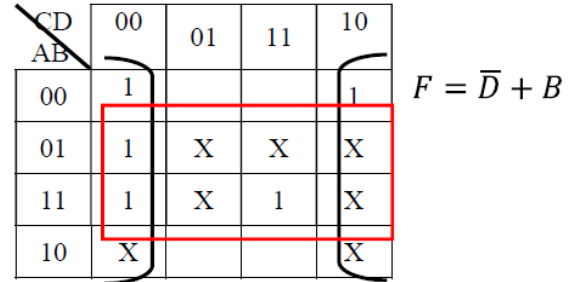
•



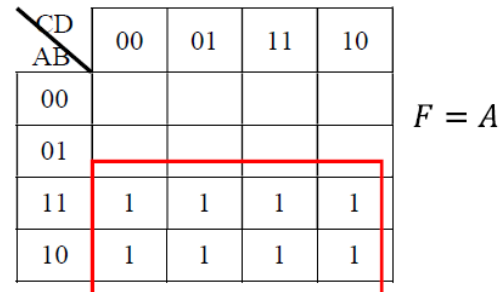
•



•



•



5. Karnough Haritaları

ÖRNEKLER

CD \ AB	00	01	11	10
00	1	1	X	1
01	1		1	
11		X	X	
10	1	X	1	

$$F(A, B, C, D) = \sum(0, 1, 2, 4, 7, 8, 11) + \Phi(3, 9, 13, 15)$$

$$F(A, B, C, D) = \bar{A}\bar{B} + CD + \bar{A}\bar{C}\bar{D} + \bar{B}\bar{C}$$

CD \ AB	00	01	11	10
00	1	1		
01	1	1	1	1
11	1	1	1	1
10	1	1		

$$F(A, B, C, D) = \bar{C} + B$$

CD \ AB	00	01	11	10
00	X	0	X	0
01	0		0	
11				0
10	0	X	0	X

$$F(A, B, C, D) = \prod(1, 2, 4, 7, 8, 11, 14) + \Phi(0, 3, 9, 10)$$

$$F = B \cdot (A + C + D) \cdot (A + \bar{C} + \bar{D}) \cdot (\bar{A} + \bar{C} + D)$$

CD \ AB	00	01	11	10
00	0	0		0
01	0	0	0	X
11		X	X	
10	0	X	0	

$$F(A, B, C, D) = \prod(0, 1, 2, 4, 5, 7, 8, 11) + \Phi(6, 9, 13, 15)$$

$$F = (B + C)(A + \bar{B})(\bar{A} + \bar{D})(A + D)$$

CD \ AB	00	01	11	10
00	1	1	X	1
01	1		1	
11	X			X
10	X	1		

$$F(A, B, C, D) = \sum(1, 2, 4, 7, 9) + \Phi(3, 8, 12, 14)$$

$$F = \bar{A}\bar{B} + \bar{C}\bar{D} + \bar{B}\bar{C} + \bar{A}CD$$

5. Kombinasyonel (Bileşik) Devreler

Temel lojik kapılardan oluşan ve devrelerin çıkışları doğrudan girişlerin o anki durumlarına göre belirlenen devreler, ‘bileşik mantık devreleri’ olarak adlandırılır. Bileşik devreler bazen, ‘birleşik mantık devreleri’ olarak da isimlendirilir. Bir bileşik devre; giriş değişkenleri, lojik kapılar ve çıkış değişkenlerinden oluşur (Şekil 8.1). Lojik kapı, giriş değişkenlerini alır, bunları işler ve çıkış için bilgi (değişkenler) üretir. Yapılan işlem, ikili giriş verilerin işlenmesi ve uygun çıkış verileri şekline dönüştürülmesidir.

5. Kombinyasyonel Devreler



Şekil 8.1. Bileşik mantık devresi blok şeması

Giriş verileri; bir harici kaynaktan gelen 'n' sayıda ikili giriş değişkenlerini, çıkış verileri; bir harici devreye doğru yönelmiş 'm' sayıda çıkış değişkenlerini içerir (Şekil 8.1). Giriş değişkenlerinin değeri, 2^n sayıda farklı ikili giriş kombinasyonundan birisi olabilir ve her bir giriş kombinasyonu için yalnızca bir çıkış kombinasyonu mevcuttur.

5. Kombinyasyonel Devreler

- Çok farklı uygulama alanları bulunan bileşik mantık devreleri, dört farklı grup altında
- incelenebilir:
- i- **Kodlama ile İlgili Lojik Devreler:** Kodlayıcı (Encoder), Kod cozucu (decoder),
- ii- **Veri seçiciler** (Multiplexer).
- iii- **Azlayıcı Devreler** (Demultiplexer).
- iv- **Kıyaslama ve Aritmetik İşlemler ile İlgili Devreler:** Karşılaştırmacı (comparator),
- Toplayıcı (adder), Cıkarıcı (subtractor), Carpıcı (multiplier).
- Farklı alanlarda farklı amaclar icin kullanılan bileşik devrelerin tasarımında aynı işlem sırası takip edilir. Bu nedenle bileşik devreleri detaylandırmadan once, bileşik devrelerin tasarımında kullanılan prensipleri inceleyelim.

5.Kombinasyonel Devreler

5.1. Bileşik Devre Tasarım Esasları:

Bir bileşik devrenin tasarımı, problemlerin ifade edilmesiyle başlayıp, lojik devrenin çizilmesi ile biter. Lojik tasarımın içerdiği işlem basamakları aşağıdaki şekilde özetlenebilir:

- i- Problem belirlenir.
- ii- Giriş değişkenlerinin sayısı ve gerekli çıkış değişkenleri tespit edilir.
- iii- Giriş ve çıkış olarak kullanılacak değişkenlere isim verilir.
- iv- Giriş ve çıkış değişkenleri arasındaki ilişkiyi belirleyen doğruluk tablosu oluşturulur.
- v- Her bir çıkış için uygun Boolean fonksiyonu yazılır.
- vi- Elde edilen Boolean fonksiyonları sadeleştirilir.
- vii- Lojik devre çizilir.

5.Kombinasyonel Devreler

5.2. Aritmetik İşlem Devreleri:

Toplama, çıkarma, çarpma, bölme işlemlerini yapan devrelere, ‘Aritmetik İşlem Devreleri’ denir. Bilgisayarlarda ve hesap makinalarında, temel işlemler toplama ve çıkartma işlemleridir. Çarpma işlemi; toplama işleminin tekrarlanması, bölme işlemi ise; çıkartma işleminin tekrarlanması ile yapılır. Bu nedenle toplayıcı ve çıkarıcı devrelerini detaylı olarak inceleyeceğiz.

5.2.1. Toplayıcı Devreleri:

Bilgisayarlar ve hesap makinaları, her biri çok sayıda bite sahip iki adet ikili sayıyı toplama işlemini gerçekleştirirler. En basit toplama işlemi dört olası temel işlemi içerir.

$$0+0=0,$$

$$0+1=1,$$

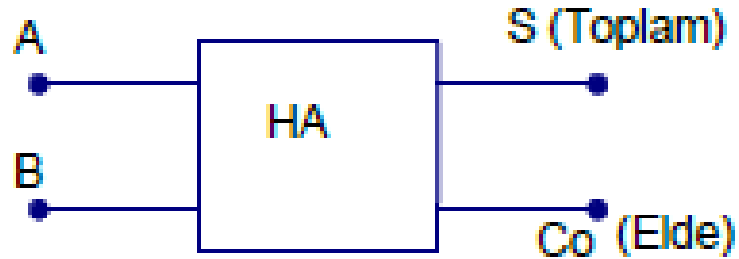
$$1+0=1,$$

$$1+1=10, (\text{Elde } 1, \text{ Toplam} = 0)$$

5.Kombinasyonel Devreler

5.2.1.i. Yarım Toplayıcı Devreleri:

Girişine uygulanan iki biti toplayıp, sonucu toplam (sum) ve elde (carry) şeklinde veren toplayıcı devresi, ‘yarım toplayıcı’ olarak isimlendirilir (Şekil 8.51). Yarım toplayıcı devresi, doğruluk tablosundan elde edilen fonksiyonların lojik devresinin çizilmesi ile oluşturulur. Oluşan devrede, ‘Toplam’ ve ‘Elde’ değerlerini temsil eden iki çıkış bulunur (Şekil 8.52).



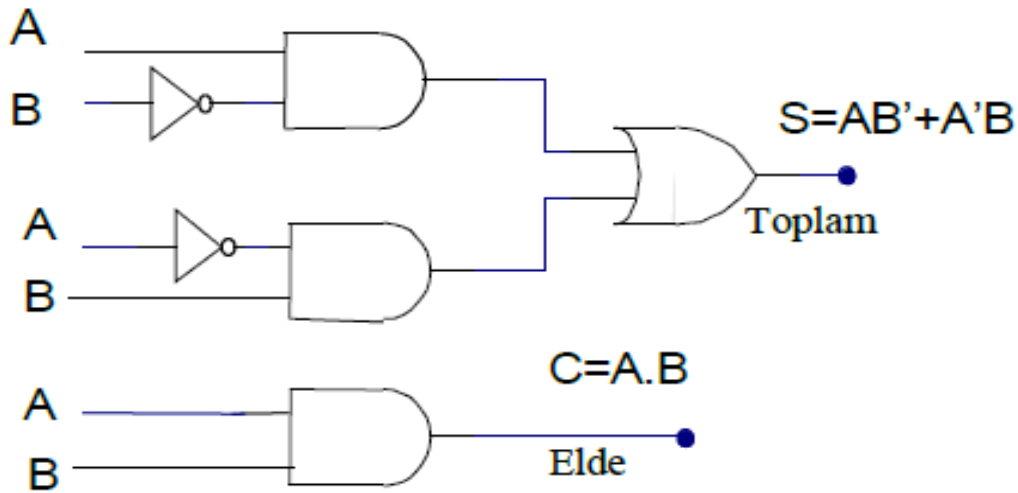
A	B	S	C _o
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Şekil 8.51. Yarım toplayıcı sembolü ve doğruluk tablosu.

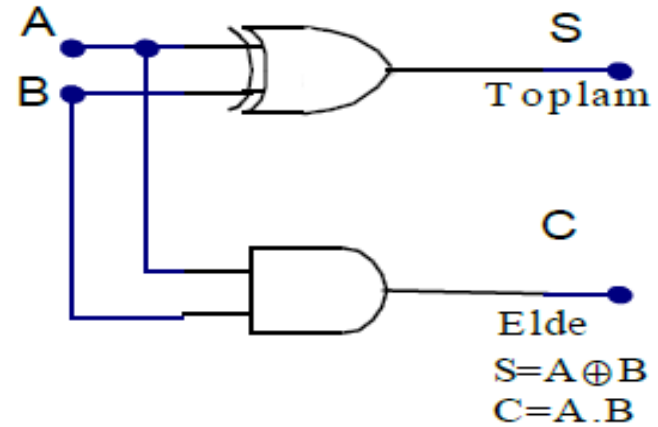
5.Kombinasyonel Devreler

5.2.1.i. Yarım Toplayıcı Devreleri:

Yarım toplayıcı çıkışlarındaki sadeleştirilmiş fonksiyonlar, $S = A'B + AB'$ ve $C = AB$ şeklinde elde edilir (Şekil 8.52.a). Girişlerin A ve B, çıkışların S ve C değişkenleri ile ifade edildiği yarım toplayıcı devresi, bir 'Özel-VEYA' (EXOR) ve bir 'VE' kapısıyla oluşturulabilir (Şekil 8.52.b).



(a)



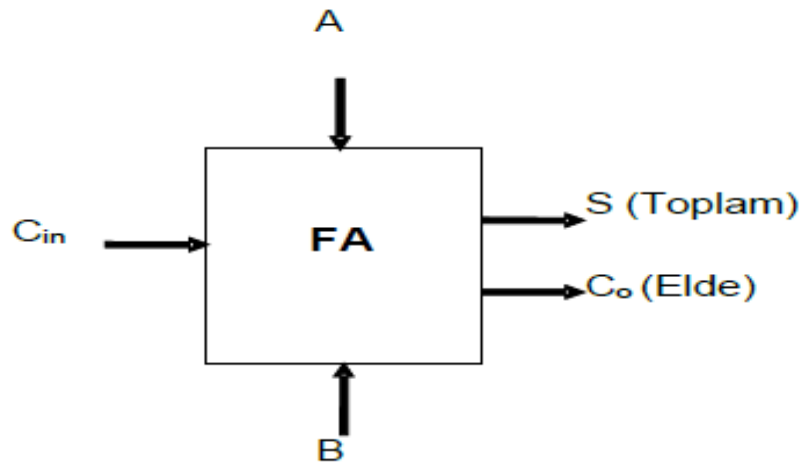
(b)

Şekil 8.52. Yarım toplayıcı devresi lojik şemaları.

5.Kombinasyonel Devreler

5.2.1.ii. Tam Toplayıcı Devreleri:

Bir bitlik üç adet sayının toplamını gerçekleştiren ve sonucu S ve C olarak isimlendirilen iki çıkış hattında gösteren düzenek, ‘Tam Toplayıcı’ olarak isimlendirilir (Şekil 8.53.a.). Girişlerden ikisi toplanacak bitleri gösterirken, üçüncü giriş bir önceki düşük değerlikli basamaktan gelen eldeyi (carry) ifade etmek için kullanılır. Tam toplayıcı devresi tasarlamak için Şekil 8.53.b’deki doğruluk tablosundan faydalanılabilir.



(a)

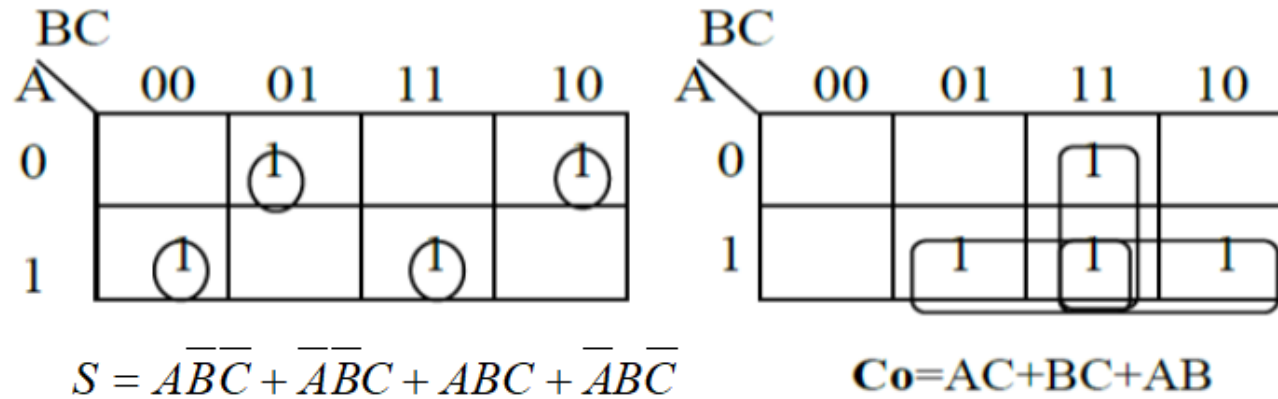
A	B	C	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(b)

5.Kombinasyonel Devreler

5.2.1.ii. Tam Toplayıcı Devreleri:

Toplayıcı tasarımı, doğruluk tablosunda giriş değişkenlerinin alabileceği değerler sıralandıktan sonra, kombinasyonlarda bulunan '1' değerleri Boolean kurallarına göre toplanıp, sonuçlar S ve Co kolonlarına yazılır. Devrede iki çıkış bulunduğundan, her bir çıkış için uygun olan değerleri içeren sütunlar oluşturulur ve sütunlardaki değerler Karnaugh haritalarına taşınır. Karnaugh haritalarından lojik eşitlikler elde edilir ve elde edilen eşitliklerin lojik şemaları çizilir.

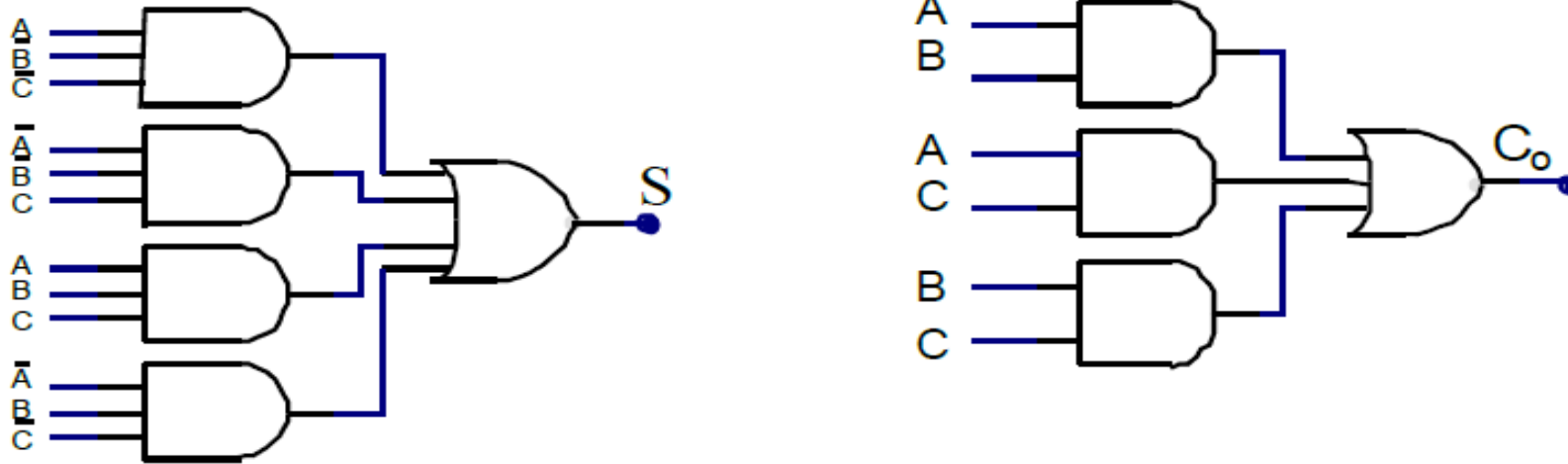


Şekil 8.54. Tam toplayıcı tasarımı için Karnaugh haritalarının oluşturulması.

5.Kombinasyonel Devreler

5.2.1.ii. Tam Toplayıcı Devreleri:

Şekil 8.53'deki doğruluk tablosundaki değerlerin Karnaugh haritalarına taşınması ile Şekil 8.54'deki haritalar elde edilir. Haritalardan elde edilen S ve Co eşitliklerine ait devrelerin çizilmesi ile, Şekil 8.55'teki lojik devreler oluşur.



Şekil 8.55. Tam toplayıcı lojik devresi.

5.Kombinasyonel Devreler

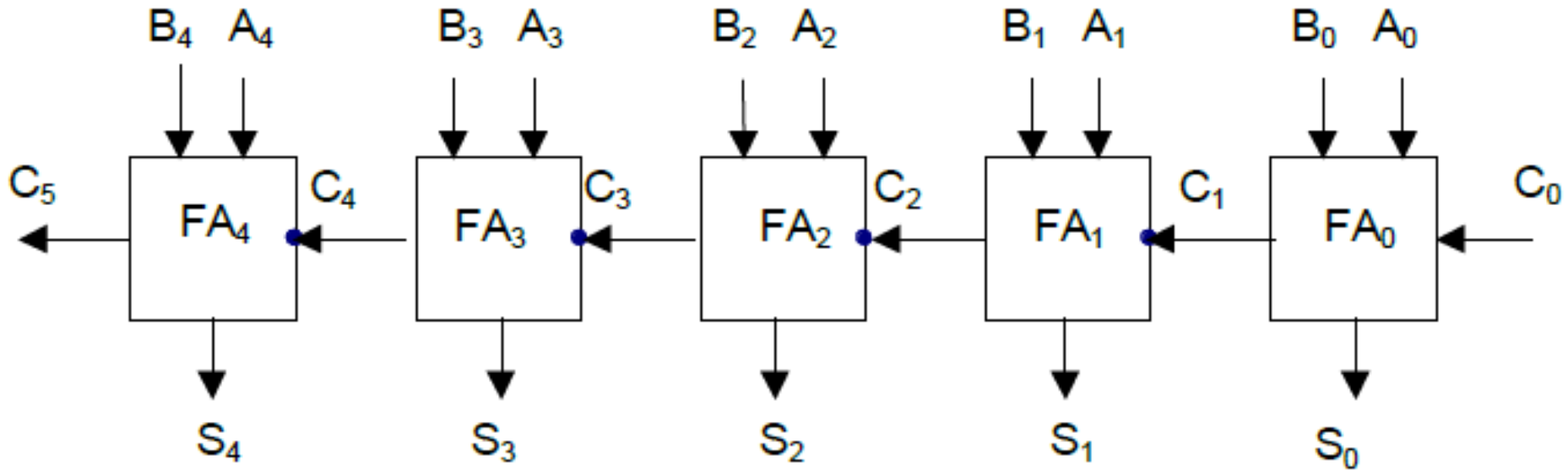
5.2.1.iii. Paralel Toplayıcı Devreleri:

Yarım ve tam toplayıcı işlemlerinde, tek bitlik sayıların toplamı işlemi açıklandı. Bununla beraber, her biri çok sayıda ikili basamak içeren iki sayının toplanması işlemini aynı anda yapan devrelere ihtiyaç vardır. Bilgisayarlarda ve hesap makinalarında çok sayıda bite sahip iki sayıyı aynı anda toplayan devreler ‘paralel toplayıcı’ olarak isimlendirilir. Şekil 8.57’de, her biri beş bitlik iki sayıyı toplayan paralel toplayıcının blok şeması görülmektedir. Bu devrede toplama işlemi, en düşük basamaklı bilgilerin toplanması ile başlar.

5.Kombinasyonel Devreler

5.2.1.iii. Paralel Toplayıcı Devreleri:

En düşük değeri basamakta C_0 biti '0' olduğundan; A_0 ve B_0 değerleri toplanarak S_0 ve C_0 çıkışlarına gönderilir. Bunun dışındaki basamakları toplamak için, A_i , B_i , C_i bitler toplanarak ilgili S_i ve C_i çıkışlarında gösterilir. C_i çıkışındaki bilgi, bir sonraki yüksek basamak değerlikli bitlerin toplandığı FA_i 'nın C_i girişine uygulanır.



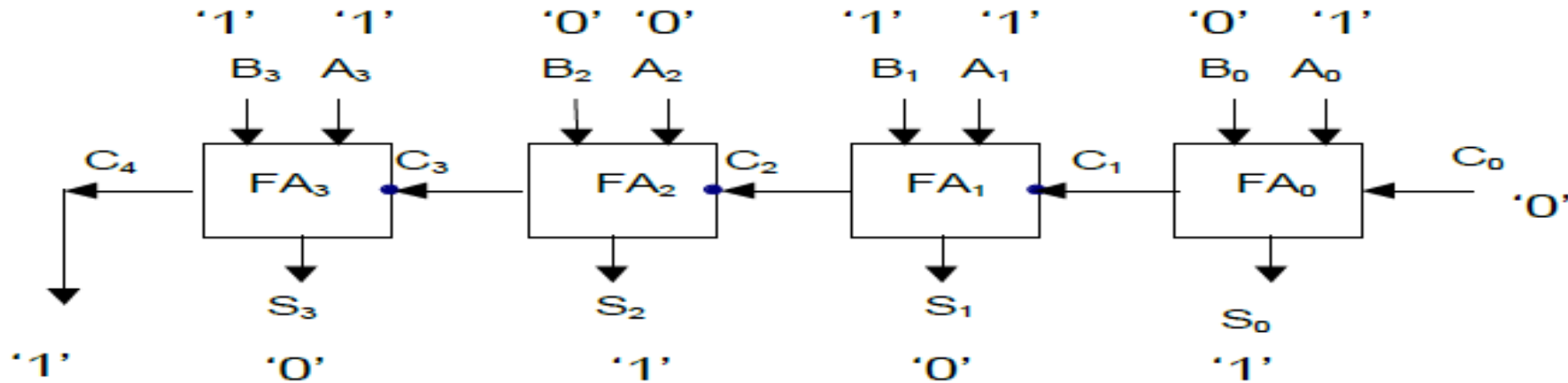
Şekil 8.57. Beş bitlik iki sayının paralel toplayıcı ile toplanması.

5.Kombinasyonel Devreler

5.2.1.iii. Paralel Toplayıcı Devreleri:

Ör: Dört bitlik paralel toplayıcı devresi ile, 1011 ve 1010 ikili sayılarını toplama işlemini yapalım.

Herbir tam toplayıcıya uygulanan sayılar ile, elde girişlerinden gelen değer toplanır. Toplama sonucu ilgili toplam (S) ve elde (C) çıkışlarına gönderilir. Tüm tam toplayıcılarda oluşan değerlerin yazılması ile Şekil 8.59'daki değerler oluşur.



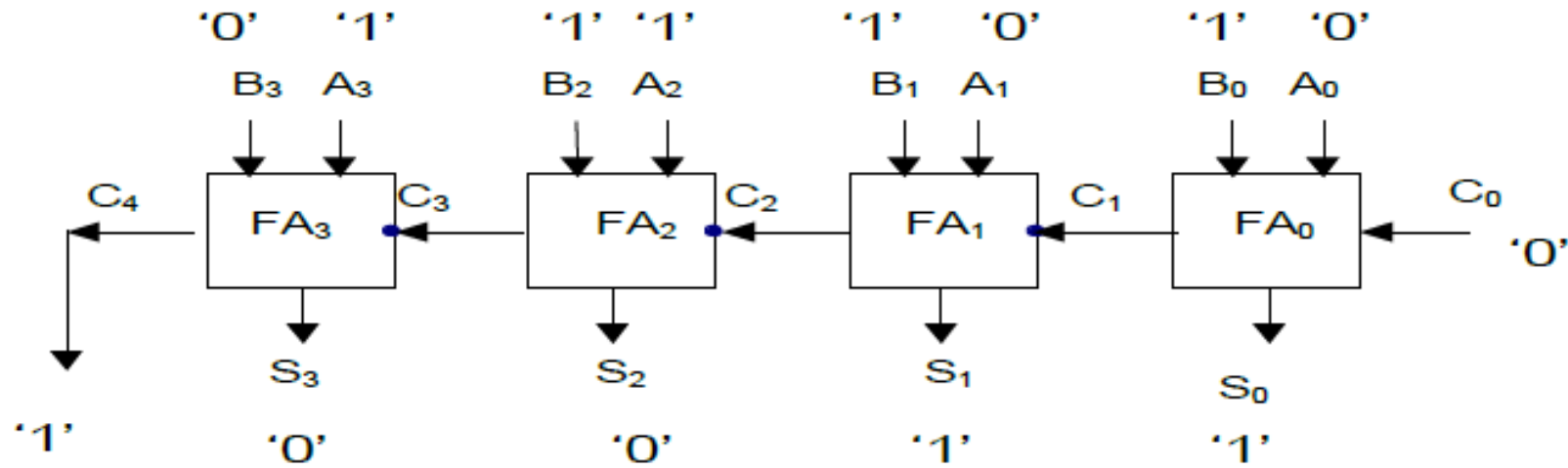
Şekil 8.59. Dört bitlik paralel toplayıcı uygulaması.

5.Kombinasyonel Devreler

5.2.1.iii. Paralel Toplayıcı Devreleri:

Ör: 0111 + 1100 işlemini dört bitlik paralel toplayıcı ile yapmak için gerekli devreyi çizerek, işlem sonuçlarını gösterelim.

Toplanacak sayılar, tam toplayıcıların girişlerine uygulanarak çıkışları yazılırsa Şekil 8.60'daki değerler bulunur.



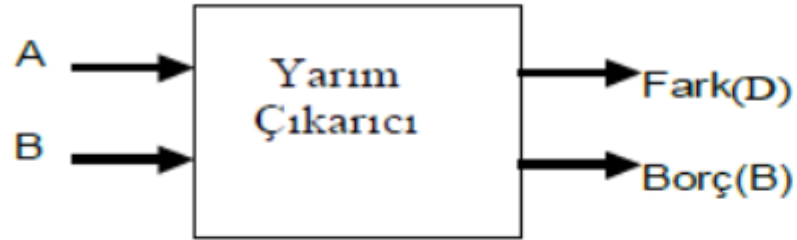
Şekil 8.60. Paralel toplayıcı ile örnek toplama işlemi.

5.Kombinasyonel Devreler

5.2.2. Çıkarıcı Devreler:

5.2.2.i. Yarım Çıkarıcı Devreler:

Lojik devrelerde yapılan ikinci temel işlem çıkarmadır. İki bitin çıkarmasını yapan devreye ‘yarım çıkarıcı’, üç bitin çıkarmasını yapan devreye ise ‘tam çıkarıcı’ devresi denir.



$$D = A'B + AB' = A \oplus B$$

$$B = A'B$$

(a)

Girişler		Çıkışlar	
A	B	Fark A-B	Borç (B)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

(b)

Şekil 8.61. Yarım çıkarıcı sembolü ve doğruluk tablosu.

5.Kombinasyonel Devreler

5.2.2.i. Yarım Çıkarıcı Devreler:

İki bitin çıkarması işlemini yapan çıkarıcı devresinde, iki giriş ve iki çıkış bulunur (Şekil 8.61.a). Çıkışlardan birisi sayının farkını (difference-D), diğeri borç bitini (borrow-B) gösterir. İki bitin çıkarılması işleminde dört farklı durum oluşur:

$$0 - 0 = 0$$

$$1 - 0 = 1$$

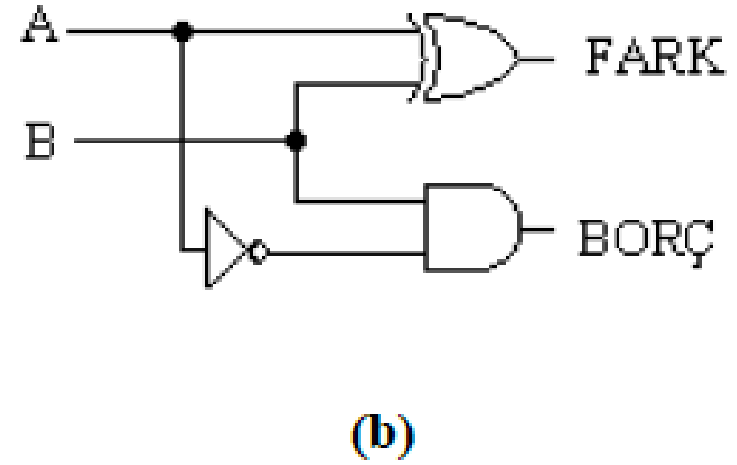
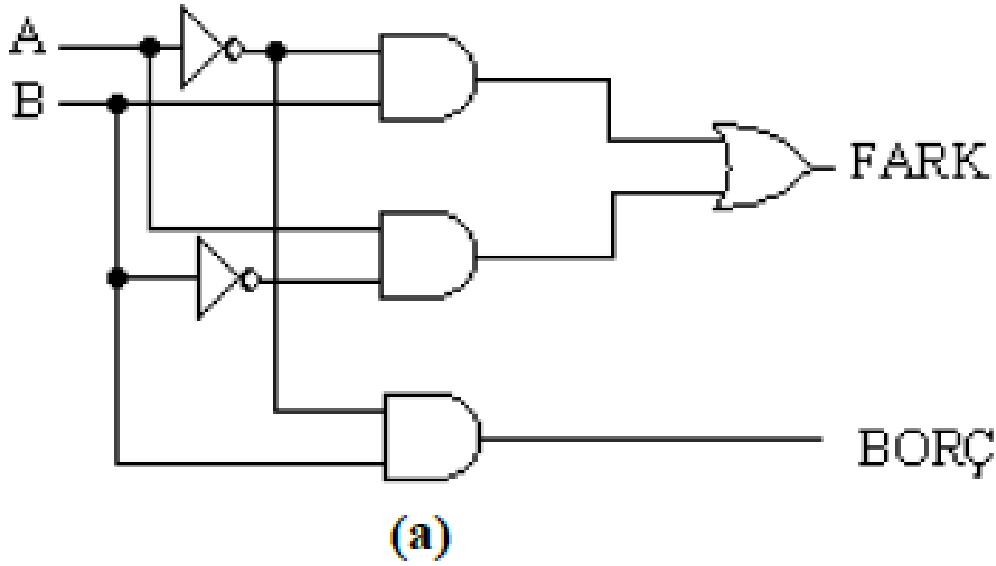
$$1 - 1 = 0$$

$$0 - 1 = 1 \text{ (Borç 1)}$$

A-B işleminde $A < B$ olduğunu zaman ‘0–1’ işlemi oluşur ve bu durumda bir yüksek değerli basamaktan ‘1’ borç alınır. Borç çıkışı, doğruluk tablosunda ayrı bir sütun olarak gösterilir. Yarım çıkarıcı devresinde oluşan işlemlerin doğruluk tablosu ve doğruluk tablosuna göre oluşan fonksiyonlar Şekil 8.61’de gösterilmektedir.

5.Kombinasyonel Devreler

5.2.2.i. Yarım Çıkarıcı Devreler:



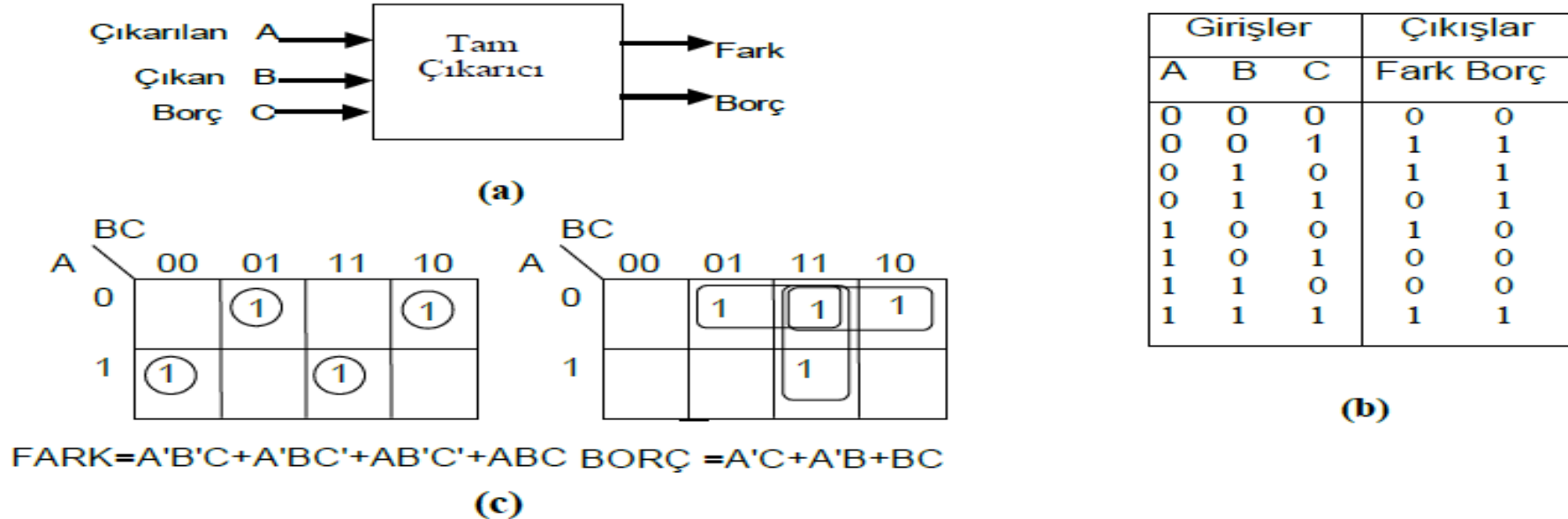
Şekil 8.62. Yarım çıkarıcı lojik devreleri.

Doğruluk tablosundan elde edilen fonksiyonları gerçekleştirecek devrenin çizilmesi ile, Şekil 8.62.a'daki devre elde edilir. Devreden elde edilen 'Fark' çıkışındaki eşitliği 'Özel-VEYA' kapısı ile gerçekleştirerek devre çizilirse, Şekil 8.62.b'deki lojik devre oluşur.

5.Kombinasyonel Devreler

5.2.2.ii. Tam Çıkarıcı Devreler:

Daha düşük değerli basamak tarafından '1' borç alınmış olabileceğini dikkate alarak iki biti birbirinden çıkaran bileşik devre, 'tam çıkarıcı' olarak isimlendirilir. Üç giriş ve iki çıkışa sahip tam çıkarıcı devresinde girişler; çıkarılan, çıkan ve borcu gösterirken, çıkışlardan biri farkı diğeri borcu gösterir (Şekil 8.64.a).

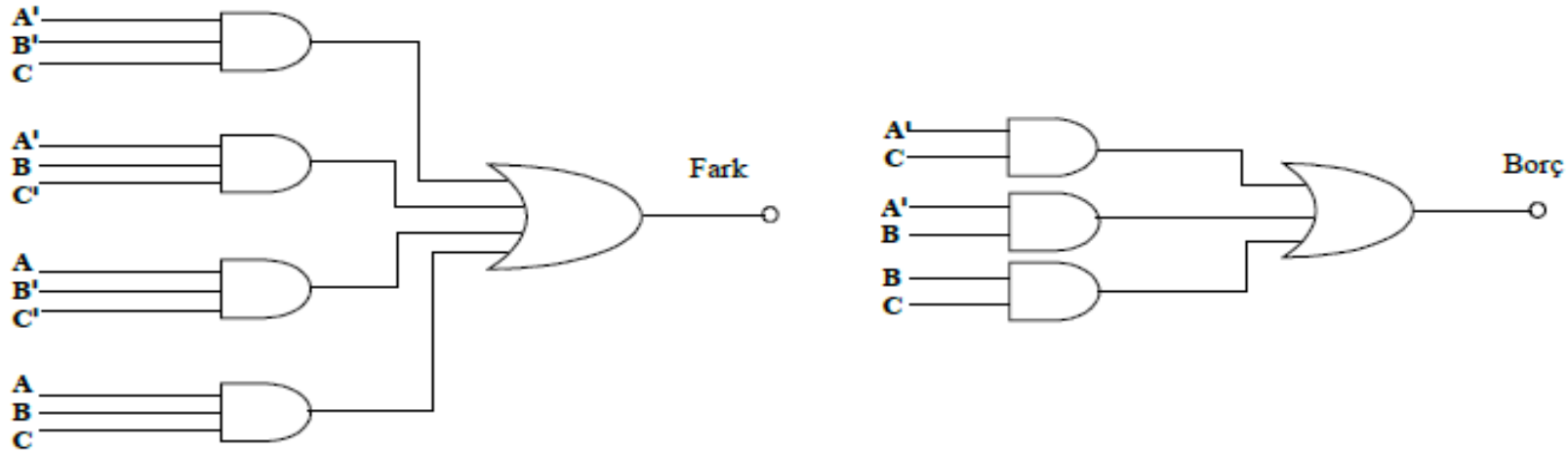


Şekil 8.64. Tam çıkarıcı devresi sembolü, doğruluk tablosu ve Karnaugh şemaları.

5.Kombinasyonel Devreler

5.2.2.ii. Tam Çıkarıcı Devreler:

Girişin üç adet olması nedeniyle, doğruluk tablosunda 8 farklı durum bulunur (Şekil 8.64.b) Doğruluk tablosundaki $A=B=0$, $C=1$ durumu, C bitinin daha önce düşük basamak değerli kademedan '1' borç alındığını gösterir. Doğruluk tablosu çıkış sütunlarındaki değerlerin Karnaugh haritasına taşınması ile Şekil 8.64.c'deki eşitlikler elde edilir. Elde edilen eşitliklerden, 'Fark' sadeleştirilemese de, 'Borç' ifadesi sadeleştirilebilir.

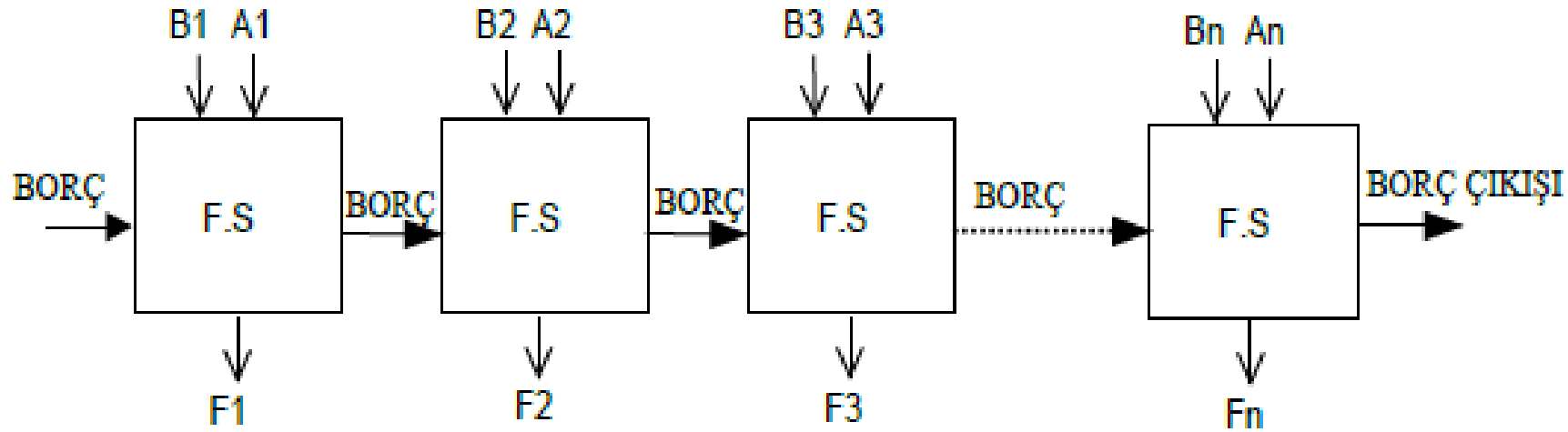


Şekil 8.65. Tam çıkarıcı devresinde 'Fark' ve 'Borç' çıkışları.

5.Kombinasyonel Devreler

5.2.2.iii. Paralel Çıkarıcı Devreler:

‘n’ bitlik iki adet ikili sayıyı çıkaran paralel çıkarıcı devresinde, paralel toplayıcılarda olduğu gibi ‘n’ sayıda tam çıkarıcı (F.S.) devresi kullanılır (Şekil 8.71). Blok şema olarak gösterilen paralel çıkarıcılarda en sondaki borç çıkışı ‘1’ ise; çıkarmanın sunucunun pozitif, ‘0’ ise sonucun negatif olduğunu gösterir.



Şekil 8.71. Paralel çıkarıcı devresi blok şeması.

5.Kombinasyonel Devreler

5.2.3. Çarpıcı Devreler:

İkili sayılarda çarpma işlemi, onlu sayılardaki çarpma işlemi gibi yapılır. Bunun yanında, çarpma işleminin tekrarlanan toplama işlemleri ile yapılması mümkündür.

Ör: Çarpma işlemi yapan lojik devre tasarımına örnek olması amacıyla ikili bir sayıyı kendisi ile çarpan yani karesini alan devreyi tasarlayalım.

Tasarlanacak devre, iki bitlik sayıların çarpımını yapacağından iki girişe sahip olur. Çıkışında ise $2^2 = 4$ değişik durum bulunur. Çıkış bit sayısını bulmanın en kolay yolu, giriş değişkenlerinden en büyük değerli olanını alıp, bu sayının karesini bulmaktır. Bulunan sayının en az kaç bit ile temsil edilebileceği tespit edilir. Böylece çıkış bit sayısı bulunur.

5.Kombinasyonel Devreler

5.2.3. Çarpıcı Devreler:

Yapılan örnekte en büyük sayı, $(11)_2$ ikili sayısı ile temsil edilen $(3)_{10}$ sayısıdır. Bu sayının karesi olan $(9)_{10}$ sayısı en az dört bit ile yazılacağından, devrenin çıkışı 4 adet olarak bulunur. Her bir çıkış, ikili olarak farklı basamak değerini temsil eder. Girişlere bağımlı olarak '1' değerini alacak çıkışlara '1' yazılırsa, Şekil 8.76'daki doğruluk tablosu ve çıkış ifadeleri oluşur.

Girişler		Çıkışlar			
A	B	F ₃	F ₂	F ₁	F ₀
0	0	0	0	0	0
0	1	0	0	0	1
1	0	0	1	0	0
1	1	1	0	0	1

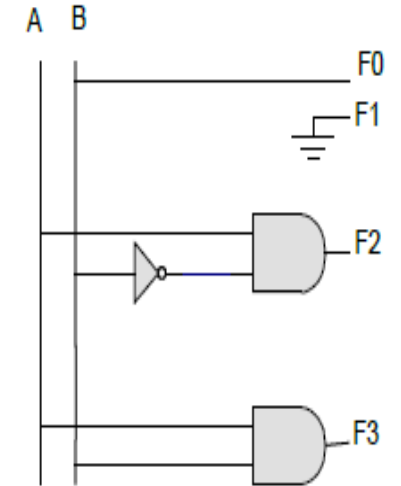
$$F_0 = A'B + AB = B(A' + A)$$

$$F_1 = 0$$

$$F_2 = AB'$$

$$F_3 = AB$$

GİRİŞLER		ÇIKIŞLAR			
A	B	F ₃	F ₂	F ₁	F ₀
0	0	0	0	0	0
0	1	0	0	0	1
1	0	0	1	0	0
1	1	1	0	0	1



Şekil 8.76. İki bitlik sayının karesini alan lojik devrenin doğruluk tablosu ve çıkış eşitlikleri.

Doğruluk tablosundan elde edilen fonksiyonlar sadeleştirilerek, Şekil 8.77'deki lojik devre elde edilir.

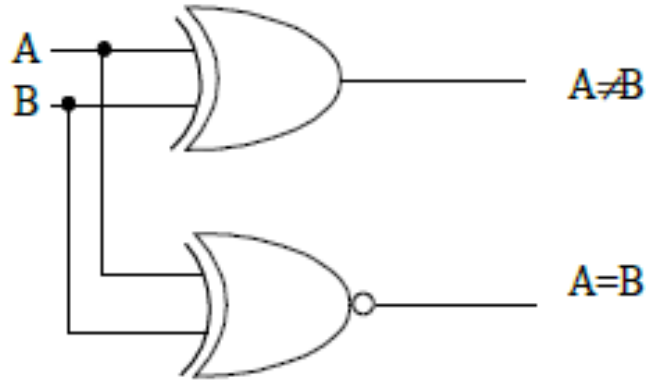
Şekil 8.77. İki bitlik ikili sayının karesini alan lojik devre şeması.

5.Kombinasyonel Devreler

5.3. Karşılaştırıcılar:

İki sayıyı karşılaştıran ve büyüklüklerini belirleyen bileşik devreler, ‘büyüklük karşılaştırıcı’ (magnitude comparator) olarak isimlendirilir. Karşılaştırma sonucu; $A > B$, $A = B$ veya $A < B$ ’yi belirleyen üç konum ile belirlenir. En yaygın kullanım yerleri Aritmetik Lojik devrelerdir. Karşılaştırıcı devreleri, girişleri aynı veya farklı iken çıkış veren kontrol devrelerinde ve ikili karşılaştırmaların kullanıldığı adres bulma devrelerinde kullanılır. En basit karşılaştırıcı devresi, tek bitlik A ve B sayılarının eşitlik durumunu karşılaştıran karşılaştırıcı devresidir. Bu devrede $A = B$ durumunda çıkışlardan birisi ‘1’ olurken, $A \neq B$ durumunda diğeri ‘1’ olur (Şekil 8.45).

Şekil 8.45.
bitlik iki
sayının



Bir

Girişler		Çıkışlar	
A	B	$A \neq B$	$A = B$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

karşılaştırması.

5.Kombinasyonel Devreler

5.3. Karşılaştırıcılar:

ÖR: İki bitlik bilgiyi karşılaştıran ve $A=B$, $A>B$ ve $A<B$ çıkışlarını üreten devreyi tasarlayalım.

Devrenin doğruluk tablosu oluşturulur ve çıkışı temsil eden fonksiyonlar yazılırsa, Şekil 8.46.a'daki eşitlikler elde edilir. Elde edilen eşitlikleri temsil eden devrenin çizilmesi ile Şekil 8.46.b'deki lojik devre oluşur.

A	B	$A>B$	$A=B$	$A<B$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

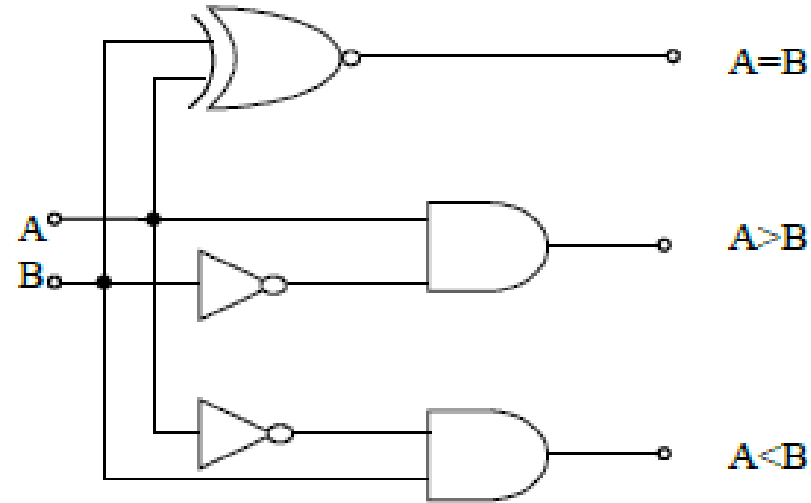
$$A>B=AB'$$

$$A=B=A'B'+AB$$

$$=A\odot B$$

$$A<B=A'B$$

(a)



(b)

Şekil 8.46. Bir bitlik iki sayıyı karşılaştıran lojik devre tasarımı.

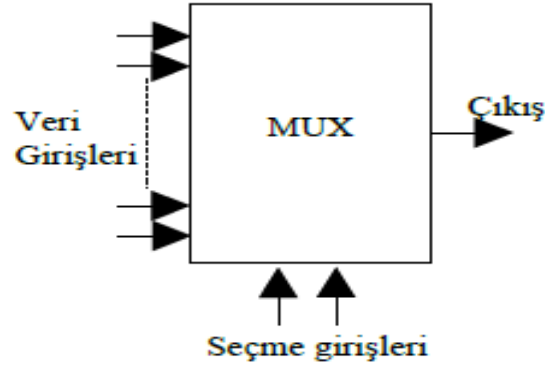
5.Kombinasyonel Devreler

5.4. oğullayıcı/Seçici(MUX) Devreleri:

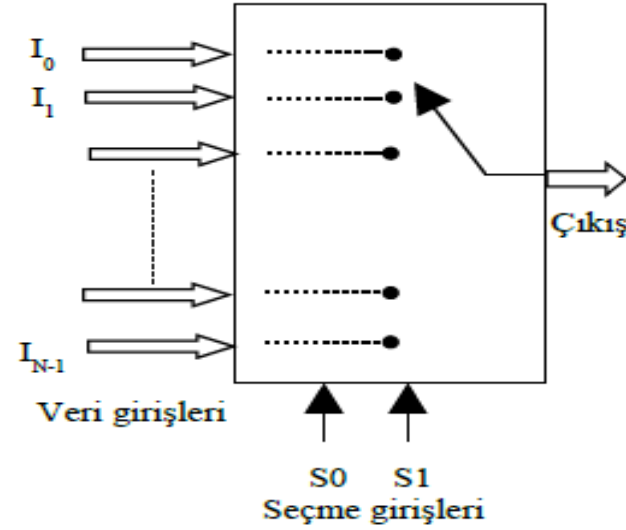
Çok sayıdaki giriş bilgisinin zaman paylaşımı olarak sırayla çıkışa aktarılması olayı,‘multiplexing - veri seçme / çoklama’ olarak tanımlanır.Bir çok giriş hattından gelen bilgilerden birisini seçerek uygun çıkış hattına yönlendirilmesini sağlayan bileşik devrelere ‘çoklayıcı / veri seçici devreler’ (multiplexer) denir ve OĞ (MUX) sembolü ile gösterilir. Birçok veri transferi, zaman paylaşım tekniğı kullanılarak multiplekser devreleri yardımıyla gerçekleştirilir. Çoklayıcılar, orijinal isminden hareketle çoğı kere ‘multiplexer-multipleksır’ olarak adlandırılır.

5.Kombinasyonel Devreler

5.4. Çokgüllayıcı/Seçici(MUX) Devreleri:



(a)



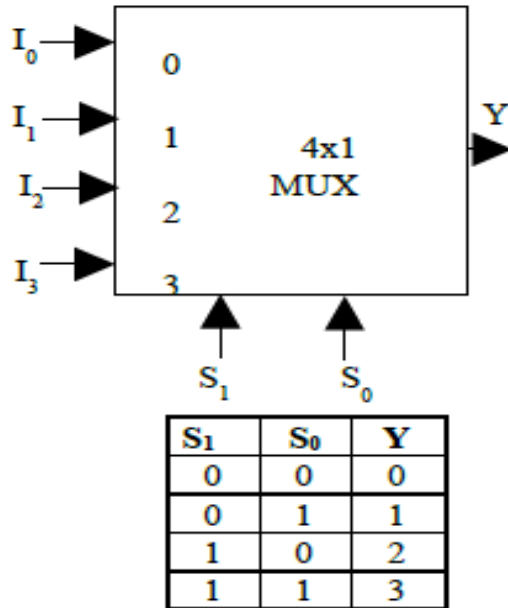
b)

Şekil 8.21.a'da sembolü ve Şekil 8.21.b'de fonksiyon şeması görünen veri seçici devresinde girişteki bilgilerden uygun olanının seçilmesi işlemi seçme girişleri (select inputs) ile yapılır. Veri seçicilerde, 2^n sayıdaki giriş hattından uygun olanı seçmek için 'n' sayıda seçme hattına ihtiyaç vardır. Dijital olarak kontrol edilebilen çok pozisyonlu anahtar gibi işlem yapan veri seçiciler, seçme hattının girişlerindeki değere göre çıkışa aktarılacak giriş hattına karar verir (Şekil 8.21.b).

5.Kombinasyonel Devreler

5.4. Çoğullayıcı/Seçici(MUX) Devreleri:

Şekil 8.23’de blok şeması ve lojik devresi verilen dört girişli multiplexer devresinde; S_0 , S_1 girişlerinin kombinasyonuna göre girişlerden birisi çıkışa aktarılır. Diğer bir değişle, her farklı seçme kombinasyonunda bir giriş çıkışta gözükür. Örneğin, I_0 girişi $S_0=0$, $S_1=0$ kombinasyonu sonucu çıkışa aktarılırken, I_2 girişi $S_0=0$, $S_1=1$ kombinasyonu sonucu çıkışta görülür. 4x1 MUX olarak isimlendirilen bu devreye benzer şekilde iki, sekiz ve onaltı girişli multiplexer devreleri TTL ve CMOS entegre olarak piyasada bulunmaktadır.



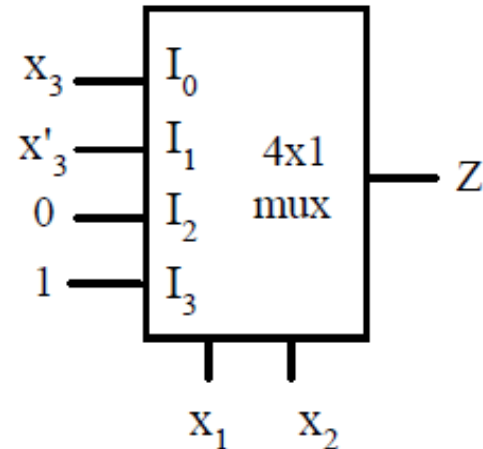
5.Kombinasyonel Devreler

5.4.1. Boolean Fonksiyonlarının ve Bileşik Devrelerin Gerçekleştirilmesi:

Yukarıda açıklananndan bir adım daha ileri giderek, $(n+1)$ değişkenli bir fonksiyon n seçicili $2^n \times 1$ lik bir mux la gerçekleştirilebilir. Bunun için n değişken seçiciler olarak alınıp, girişlere de 0, 1, x_n veya x'_n den biri uygulanır. Bir örnek üzerinden gerçekleştirmenin nasıl yapılacağını görelim. Şekil 5.13 te bir örnek fonksiyon ve devresi verilmiştir. Fonksiyon üç değişkenli $f = \Sigma 1, 2, 6, 7$ olduğuna göre 4×1 lik iki seçicili bir mux a gereksinim vardır. x_1, x_2 değişkenlerini seçiciler olarak seçelim. Doğruluk tablosundan görüldüğü gibi x_1, x_2

değişkenlerinin bir değer permütasyonu için x_3 in aldığı 0 ve 1 olmak üzere iki değer vardır. Bu değerler için fonksiyonun aldığı değerler 0,1, x_3 veya x'_3 olabilir. Buna göre tabloyu tekrar düzenlersek (gölgelendirilerek belirtildi).

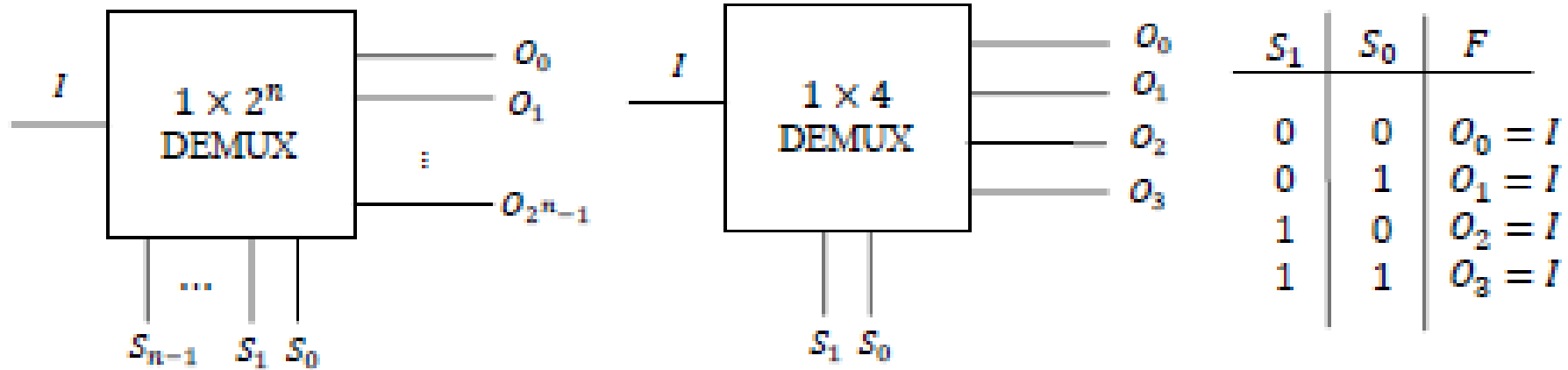
minter n	x_1	x_2	x_3	f	
0	0	0	0	0	
1	0	0	1	1	$f = x_3$
2	0	1	0	1	$f =$
3	0	1	1	0	x'_3
4	1	0	0	0	$f = 0$
5	1	0	1	0	
6	1	1	0	1	$f = 1$
7	1	1	1	1	



5.Kombinasyonel Devreler

5.4. Azlayıcılar - Veri Dağıtıcılar (DEMUX) Devreleri:

Tek bir girişten aldığı bilgileri, her bir çeşit giriş bilgisi farklı çıkışta olacak şekilde dağıtım yapan devrelere, ‘Azlayıcı / Veri dağıtıcı devreler’ (Demultiplexer) ismi verilir. Multiplexer’ın yaptığı işlemin tersini yapan bu devrede seçici girişlerin değeri, giriş verilerinin hangi çıkışa gönderileceğini belirler. Özet olarak; ‘demultiplexer devresi, tek bir kaynaktan gelen bilgileri seçme girişleri yardımıyla ayırarak, N çıkış hattından birisine gönderen çok konumlu bir anahtardır’ denebilir.



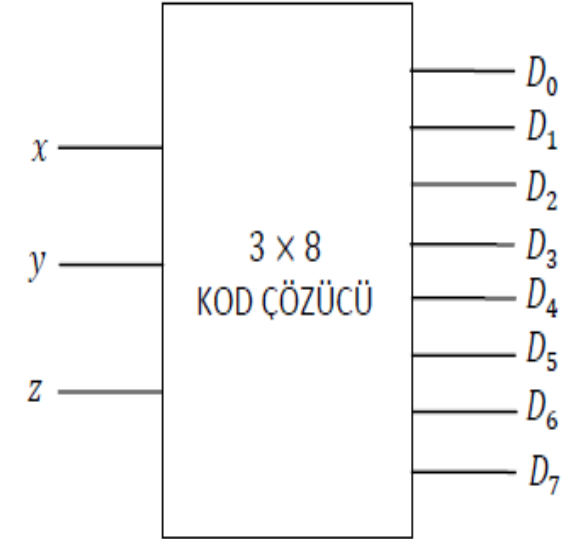
Şekil 9.8. Demultiplexer

5.Kombinasyonel Devreler

5.5. Kod Çözücüler (Decoders) Devreleri:

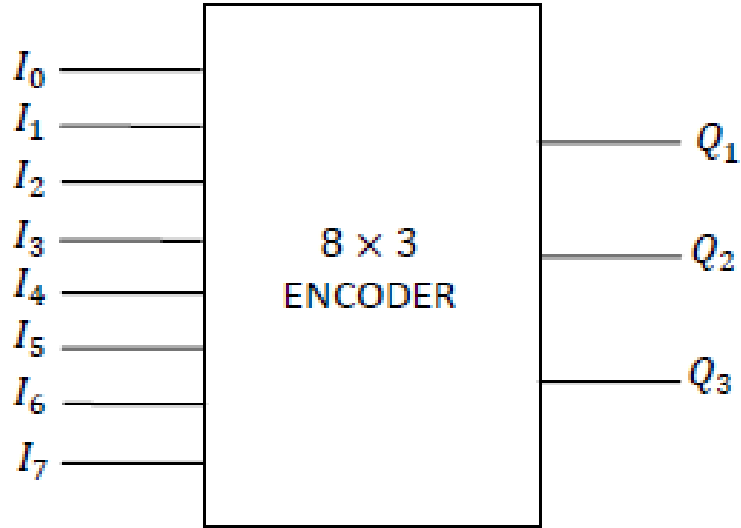
Kod çözücü n bitlik bir sözcüğün kodunu çözüp olası en çok 2^n çıkış yolundan sadece birini aktif hale getiren bir kombinasyonel devredir. 3 girişli, $2^3 = 8$ çıkışlı bir kod çözücüye ait doğruluk tablosu ve devre aşağıdaki gibidir.

GİRİŞLER			ÇIKIŞLAR							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



5.Kombinasyonel Devreler

5.6. Kodlayıcılar (Encoders) Devreleri:



Kod çözücünün tersi işlevi yerine getiren devrelere kodlayıcı (encoder) denir. Bu devrelerin giriş sayısı 2^n , çıkış sayısı ise n dir. Girişlerden biri 1 iken, diğerleri 0 olmak zorundadır. Hangi girişe (1) gelmişse on girişin iki tabanındaki karşılığı çıkışta görülür. Örneğin 6. girişe 1 gelmişse, çıkış 110 dir. Böyle bir devreyi bir uygulama olarak gerçekleyiniz.

ÖDEVLER:

- 1-Üç bitlik ikili sayının karesini alan devreyi tasarlayınız.
- 2-Üç bitlik ikili sayının karesini alan devreye 110 girişleri uygulanması durumunda oluşacak çıkış değerlerini bulunuz.
- 3-10110+11011 işlemini paralel toplayıcı kullanarak yapınız.
- 4- $F(A,B,C,D) = \Sigma (1,7,8,9,13,15)$ fonksiyonunu 8x1 MUX ile gerçekleştiriniz.
- 5- $F(A,B,C,D) = \Sigma (0,2,6,9,11,13)$ fonksiyonunu 4x1 MUX ile gerçekleştiriniz.

KAYNAKLAR

Prof. Dr. Hüseyin EKİZ, MANTIK DEVRELERİ, DERS NOTLARI, SAKARYA ÜNİVERSİTESİ.

Doç. Dr. Umut Engin AYTEN, LOJİK DEVRELER DERS NOTLARI

Prof. Dr. Ertuğrul ERİŞ LOJİK DEVRELER DERS NOTLARI