

SAYISAL TASARIM KOMBİNASYONEL DEVRELER

Hazırlayan:

Yard. Doç. Dr. Metin ARTIKLAR

KOMBİNASYONEL DEVRELER

- Bu aşamaya kadar ikili sistemler, Boolean cebiri, Boolean fonksiyonları ve K-Map'lerle sayısal devre tasarımı yapabilmek için gereken alt yapıyı oluşturduk.
- Bu aşamadan sonra ise bu alt yapıyı kullanarak sayısal devre tasarımı yapacağız.

KOMBİNASYONEL DEVRELER

- Sayısal devreleri iki ana başlık altında düşünebiliriz
 - Kombinasyonel devreler
 - Senkron (eşzamanlı) Ardışıl Devreler

Senkron (eşzamanlı) Ardışıl Devreler

- Bu devreler kombinasyonel devrelere flip floplar ve geri besleme yolları eklenerek oluşturulan devrelerdir ve bir sonraki bölümümüzün konusunu oluşturacaklardır

KOMBİNASYONEL DEVRELER

- Devre tasarımında sadece mantık kapılarının kullanıldığı dolayısıyla devrenin çıkışlarının o anki giriş değerlerine göre belirlendiği sayısal devrelere denir
- En genel haliyle bir kombinasyonel devre n tane girişi ve m tane çıkışı olan bir kutu şeklinde modellenenebilir

KOMBİNASYONEL DEVRELER

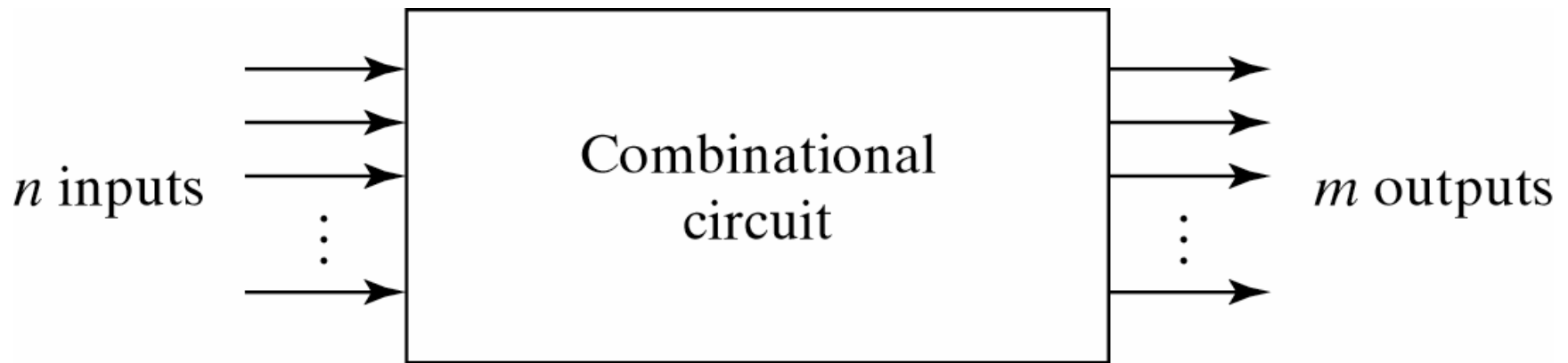


Fig. 4-1 Block Diagram of Combinational Circuit

KOMBİNASYONEL DEVRELER

- Şimdiye kadar bir tek çıkışı olan Boolean fonksiyonlarının oluşturduğu sayısal devreleri gördük. Genel olarak düşünersek bunlar $m=1$ olan kombinasyonel devrelerdir. m 'nin birden büyük olduğu kombinasyonel devreler ise aynı değişkenlere sahip birden fazla Boolean fonksiyonunun bir tek devre içerisinde gerçekleştirilmiş hali olarak düşünülebilir

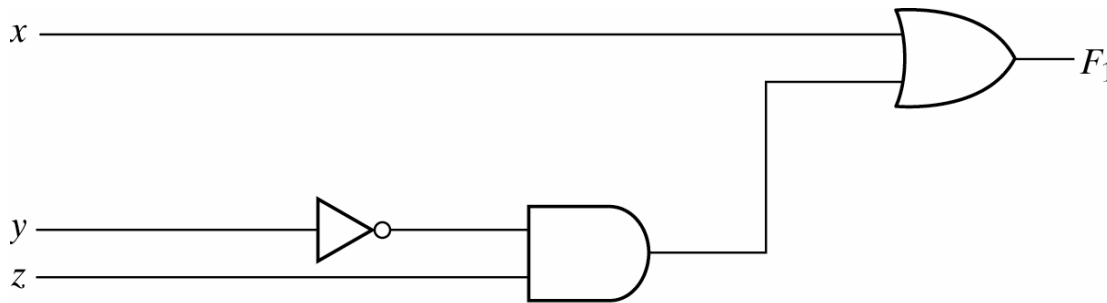
KOMBİNASYONEL DEVRELER

- Kombinasyonel devreler n giriş ve m çıkışının bir arada listelendiği doğruluk tabloları ile temsil edilir

Tek Çıkışlı (m=1 olan) Kombinasyonel Devreler

$$F_1(x,y,z) = x + y'.z$$

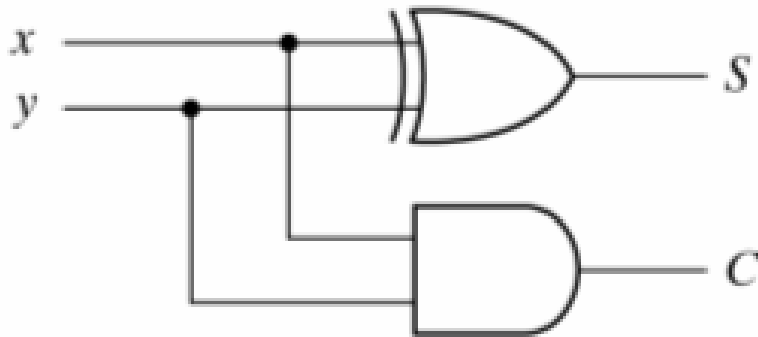
n= 3 m= 1



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

İki Çıkışlı (m=2 olan) Kombinasyonel Devreler

- $S = x \oplus y$ (n=2 m=2)
- $C = x \cdot y$



$$S = x \oplus y$$
$$C = xy$$

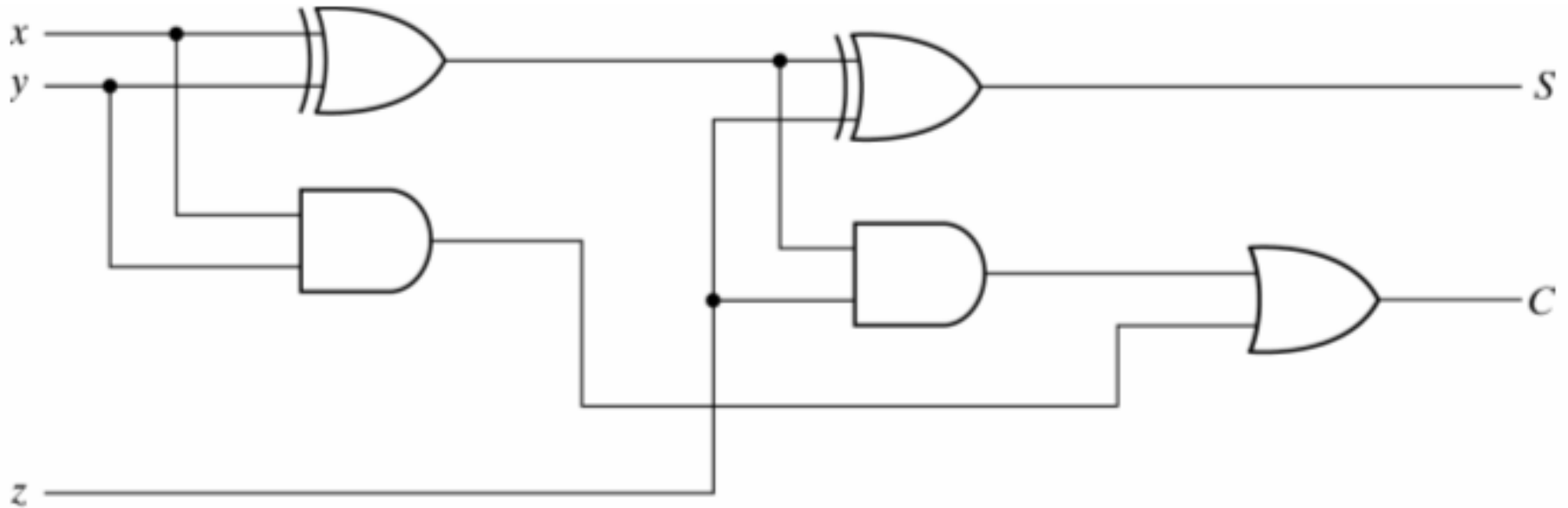
x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

İki Çıkışlı (m=2 olan) Kombinasyonel Devreler

- $S = (x \oplus y) \oplus z$ (n=3 m=2)
- $C = (x \oplus y).z + x.y$

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

İki Çıkışlı ($m=2$ olan) Kombinasyonel Devreler



Kombinasyonel Devre Analizi

- Devre şeması verilmiş olan bir kombinasyonel devrenin tüm çıkışlarını temsil eden Boolean fonksiyonlarının bulunmasına ve devrenin ne amaçla tasarlandığı hakkında fikir yürütülmesine devre analizi denir.
- Bir kombinasyonel devrenin analiz edilmesi için 3 aşamaya ihtiyaç vardır

Kombinasyonel Devre Analizi

1. m tane çıkışın her birisi için Boolean fonksiyonlarının bulunması
2. Boolean fonksiyonları yardımıyla devrenin doğruluk tablosunun oluşturulması
3. Doğruluk tablosuna bakarak devrenin çalışması veya ne amaçla tasarlanmış olduğu hakkında fikir yürütülmesi

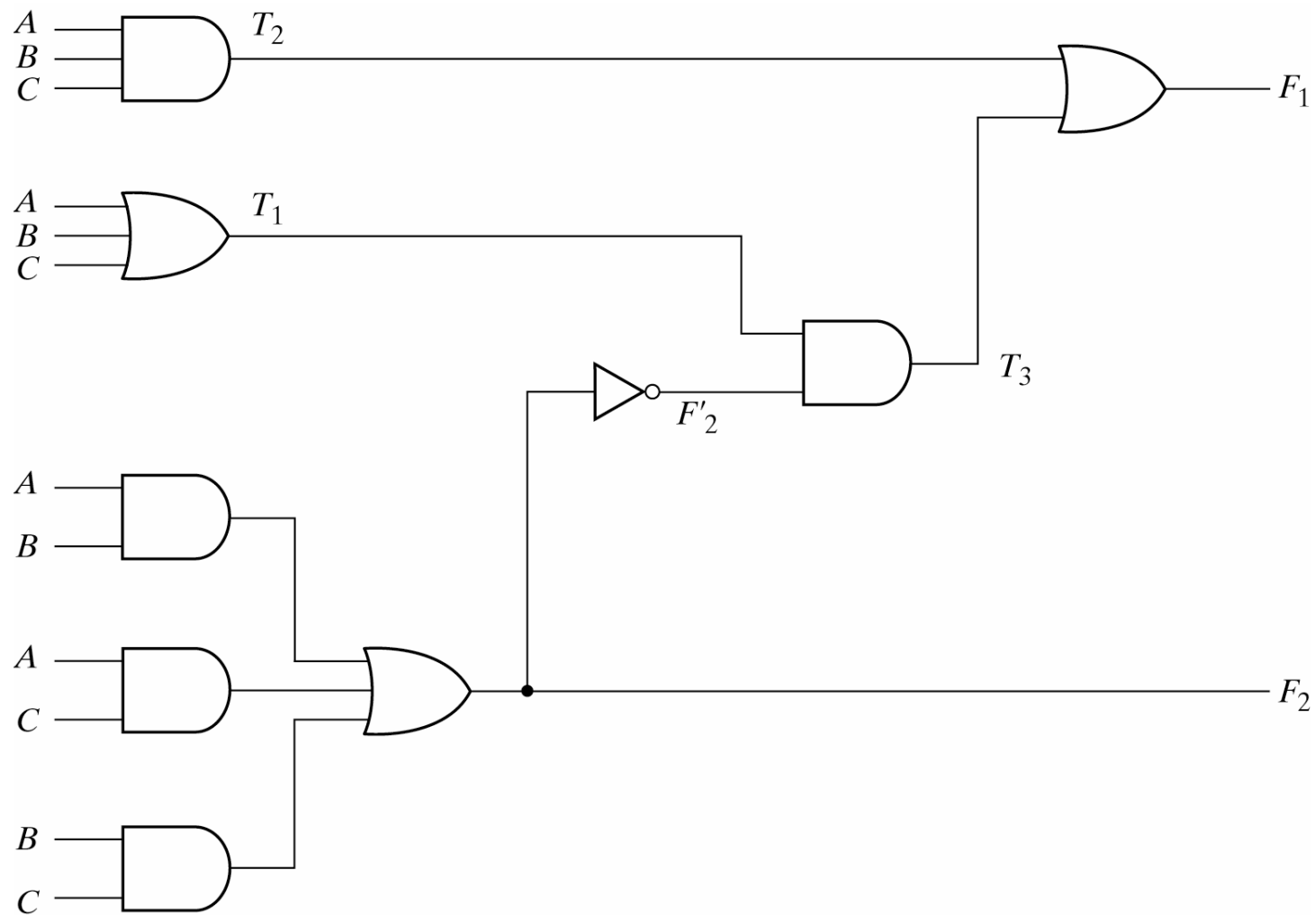


Fig. 4-2 Logic Diagram for Analysis Example

Kombinasyonel Devre Analizi

- 1. Seviye
- $T_1 = A + B + C$
- $T_2 = A.B.C$
- $F_2 = A.B + A.C + B.C$ (1. çıkış)
- 2. Seviye
- $T_3 = F_2'.T_1$
- $F_1 = T_2 + T_3 = A'BC' + A'B'C + AB'C' + A.B.C$
(2.inci çıkış)

Doğruluk tablosunu oluşturmak için şu işlemleri gerçekleştiriniz

1. Devre şemasına bakarak giriş ve çıkışların sayısını belirleyiniz ve bunları isimlendiriniz
2. Devre içerisinde 1.inci seviye ikinci seviye veya aradaki diğer yerlerden kilit noktalar seçiniz ve bu noktalardaki Boolean fonksiyonları ve doğruluk tablolarını oluşturunuz (bu noktalar genellikle seviye çıkışlarına karşılık gelir)
3. Bu noktaları kullanarak devrenin çıkışları için Boolean fonksiyonları ve doğruluk tablolarını oluşturunuz

Kombinasyonel Devre Analizi

<i>A</i>	<i>B</i>	<i>C</i>	<i>F₂</i>	<i>F₂'</i>	<i>T₁</i>	<i>T₂</i>	<i>T₃</i>	<i>F₁</i>
0	0	0	0	1	0	0	0	0
0	0	1	0	1	1	0	1	1
0	1	0	0	1	1	0	1	1
0	1	1	1	0	1	0	0	0
1	0	0	0	1	1	0	1	1
1	0	1	1	0	1	0	0	0
1	1	0	1	0	1	0	0	0
1	1	1	1	0	1	1	0	1

Sayısal Devre Dizaynı

- Tanımı yapılmış bir problemin çözümünü gerçekleyen Boolean fonksiyonlarının bulunması ve doğruluk tablosunun oluşturularak devre şemasının bulunması işlemlerinin tümüne sayısal devre dizaynı denir.
- Dolayısıyla sayısal devre dizaynı aşağıda verilen aşamaları içerir:

Sayısal Devre Dizaynı

- Problemin tanımından devrenin kaç giriş ve kaç çıkışının olduğunun belirlenmesi
- Her bir çıkış için doğruluk tablosunun bulunması ve bunların bir araya getirilmesi ile devrenin tümü için doğruluk tablosunun bulunması
- K-Map'ler kullanılarak her bir çıkışı temsil eden en sade Boolean fonksiyonlarının bulunması
- Bulunan Boolean fonksiyonlarının bir araya getirilerek devrenin şemasının çizilmesi

BCD Kodunu 3-Fazlası Koduna Çeviren Devrenin Dizaynı

1. Her iki kodda 0-9'a kadar olan sayıları 4'er bit ile kodladıklarından devrenin 4 girişi (A, B, C, D olarak adlandırılan) ve 4 çıkışı (w, x, y, z olarak adlandırılan) vardır
2. Devrenin doğruluk tablosu aşağıda verildiği gibidir
3. Her bir çıkış için Boolean fonksiyonların en sade şekli Figür 4.3'te verilmiştir
4. Devrenin şeması Figür 4.4'te verilmiştir (Devreyi yeniden çizdir)

BCD Kodunu 3-Fazlı Koduna Çeviren Devrenin Dizaynı

GİRİŞLER				ÇIKISLAR			
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

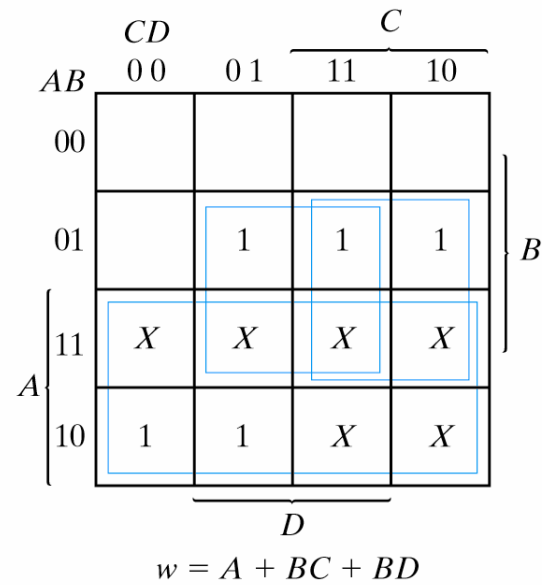
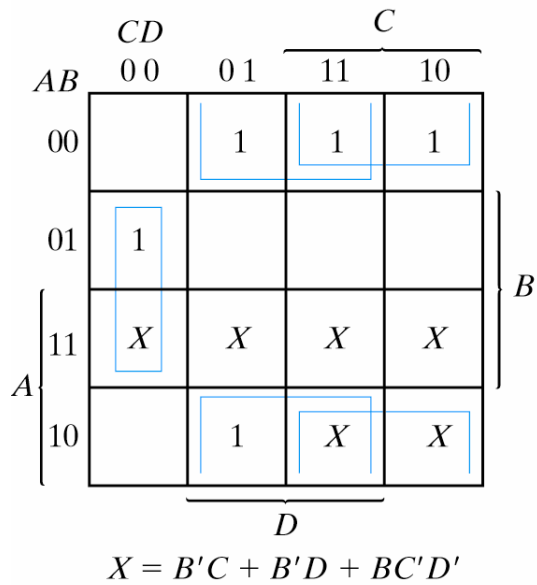
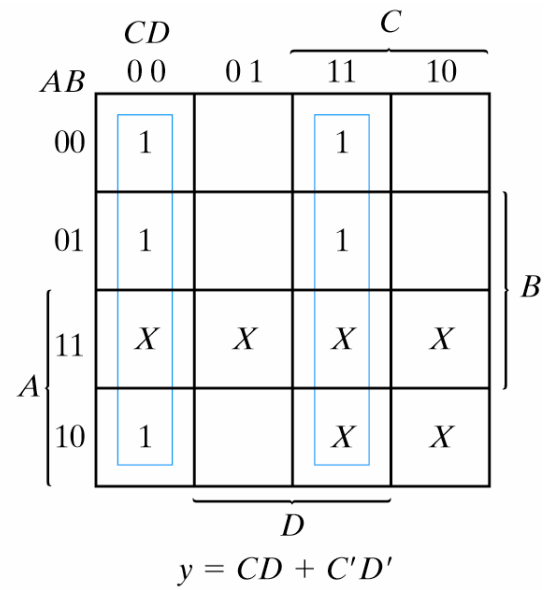
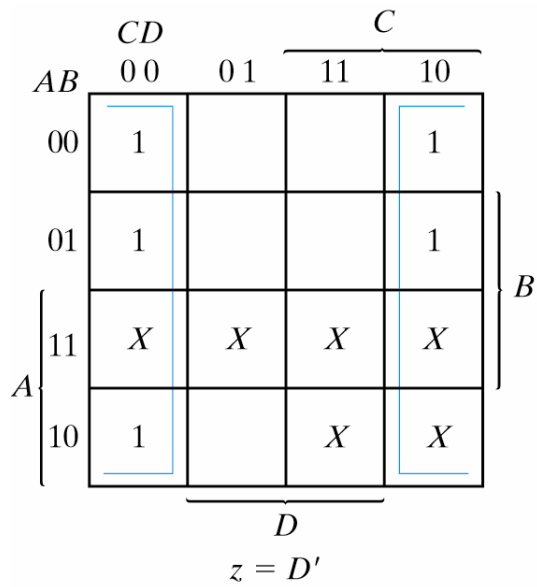


Fig. 4-3 Maps for BCD to Excess-3 Code Converter

BCD Kodunu 3-Fazlı Koduna Çeviren Devrenin Şeması

İkili Toplayıcılar ve Çıkarıcılar:

1- Yarım Toplayıcılar

- İki biti toplayabilen sayısal devreye yarım toplayıcı devresi denir. Benzer işlevleri yerine getiren 3 çeşit devrede vardır:
 - İki bit üzerinde işlem gören yarım toplayıcılar
 - Üç bit üzerinde işlem gören tam toplayıcılar
 - n bitlik sayılar üzerinde toplama ve çıkarma işlemini beraber yerine getirebilen devreler

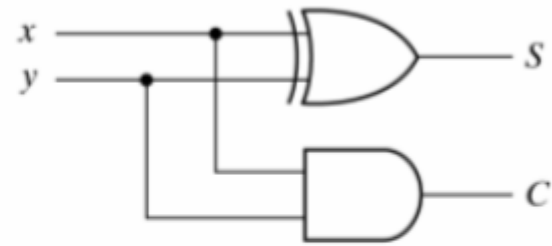
Yarım Toplayıcı Devresinin Sentezi

1. Bu devrede toplanacak bitleri temsil eden iki giriş (x , y olarak adlandırılan) ve bitlerin toplamını ve oluşabilecek eldeyi temsil eden (S , C olarak adlandırılacak) iki çıkışı vardır
2. Devrenin doğruluk tablosu bir sonraki figürde verilmiştir
3. K-Mapler kullanılarak bulunan ve çıkışları gösteren en sade Boolean fonksiyonları bir sonraki figürde verilmiştir
4. Devrenin şeması bir sonraki figürde verilmiştir

2. Doğruluk Tablosu

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

4. Devrenin Şeması



$$(b) S = x \oplus y$$

$$C = xy$$

3. Çıktıları Gösteren Boolean Fonksiyonlarının K-Map'ler İle Bulunması

	y	y
	0	1
x 0	0	0
x 1	0	1

$$C = xy$$

	y	y
	0	1
x 0	0	1
x 1	1	0

$$S = x'y + xy'$$

$$S = x \ominus y$$

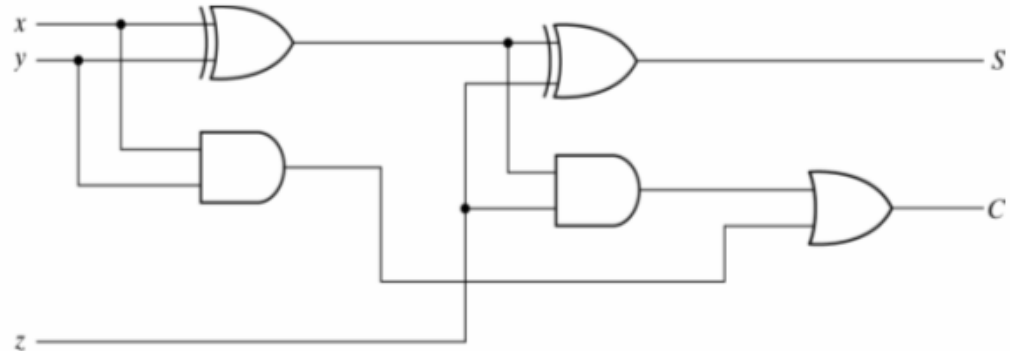
Tam Toplayıcı Devresinin Sentezi

1. Bu devrede toplanacak bitleri temsil eden üç giriş (x ve y toplanacak olan bitleri ve z 'de bir önceki basamaktan gelebilecek eldeyi gösteren) ve bitlerin toplamını ve oluşabilecek eldeyi temsil eden (S , C olarak adlandırılan) iki çıkışı vardır
2. Devrenin doğruluk tablosu bir sonraki figürde verilmiştir
3. K-Mapler kullanılarak bulunan ve çıkışları gösteren en sade Boolean fonksiyonları bir sonraki figürde verilmiştir
4. Devrenin şeması bir sonraki figürde verilmiştir

2. Doğruluk Tablosu

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

4. Devrenin Şeması



3. Çıktıları Gösteren Boolean Fonksiyonlarının K-Map'ler ile Bulunması

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

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

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

$$\begin{aligned}
 C &= xy + xy'z + x'y'z \\
 C &= xy + z(xy' + x'y) \\
 C &= xy + z(x \oplus y)
 \end{aligned}$$

n-Bitlik İkili Toplayıcılar

- İki adet n-bitlik ikili sayıyı toplayan toplayabilen sayısal devreler denir
- Aşağıda 4 bitlik iki ikili sayıyı toplayan bir toplayıcı devresi verilmiştir

İleriki Basamaklardaki Eldelerin Önceden Bulunması

- Bir önce gördüğümüz 4-bitlik toplayıcı devresinde her bir basamaktaki toplama işlevinin yerine getirebilmesi için bir önceki basamaktaki eldenin oluşması gerekmektedir.
- Böylece toplama işlemi 4 basamakta yapılabilmektedir
- Bu toplama işlemini eldeleri önceden hesaplatarak iki basamakta yapabiliriz

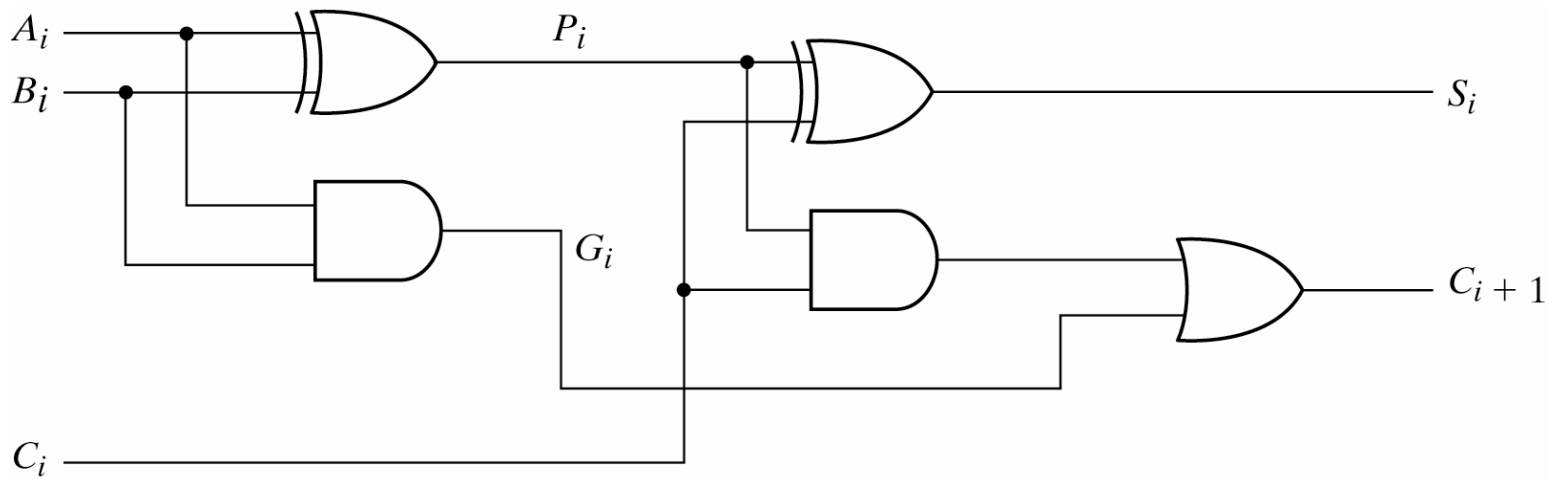


Fig. 4-10 Full Adder with P and G Shown

$$S_i = C_i \oplus (A_i \oplus B_i) \quad C_{i+1} = A_i \cdot B_i + C_i \cdot (A_i \oplus B_i)$$

İleriki Basamaklardaki Eldelerin Önceden Bulunması

- $P_i = A_i \ominus B_i$; $S_i = P_i \ominus C_i$
- $G_i = A_i . B_i$; $C_{i+1} = G_i + P_i \ominus C_i$
- C_0
- $C_1 = G_0 + P_0 \ominus C_0$
- $C_2 = G_1 + P_1 \ominus C_1 = G_1 + P_1 \ominus (G_0 + P_0 \ominus C_0)$
- $C_3 = G_2 + P_2 \ominus C_2 = G_2 + P_2 \ominus G_1 + P_2 \ominus P_1 \ominus G_0 + P_2 \ominus P_1 \ominus P_0 \ominus C_0$

Bu yöntemle toplama işlemine alınan sayıların bit sayılarına bakılmaksızın eldeler iki basamakta bulunabilir

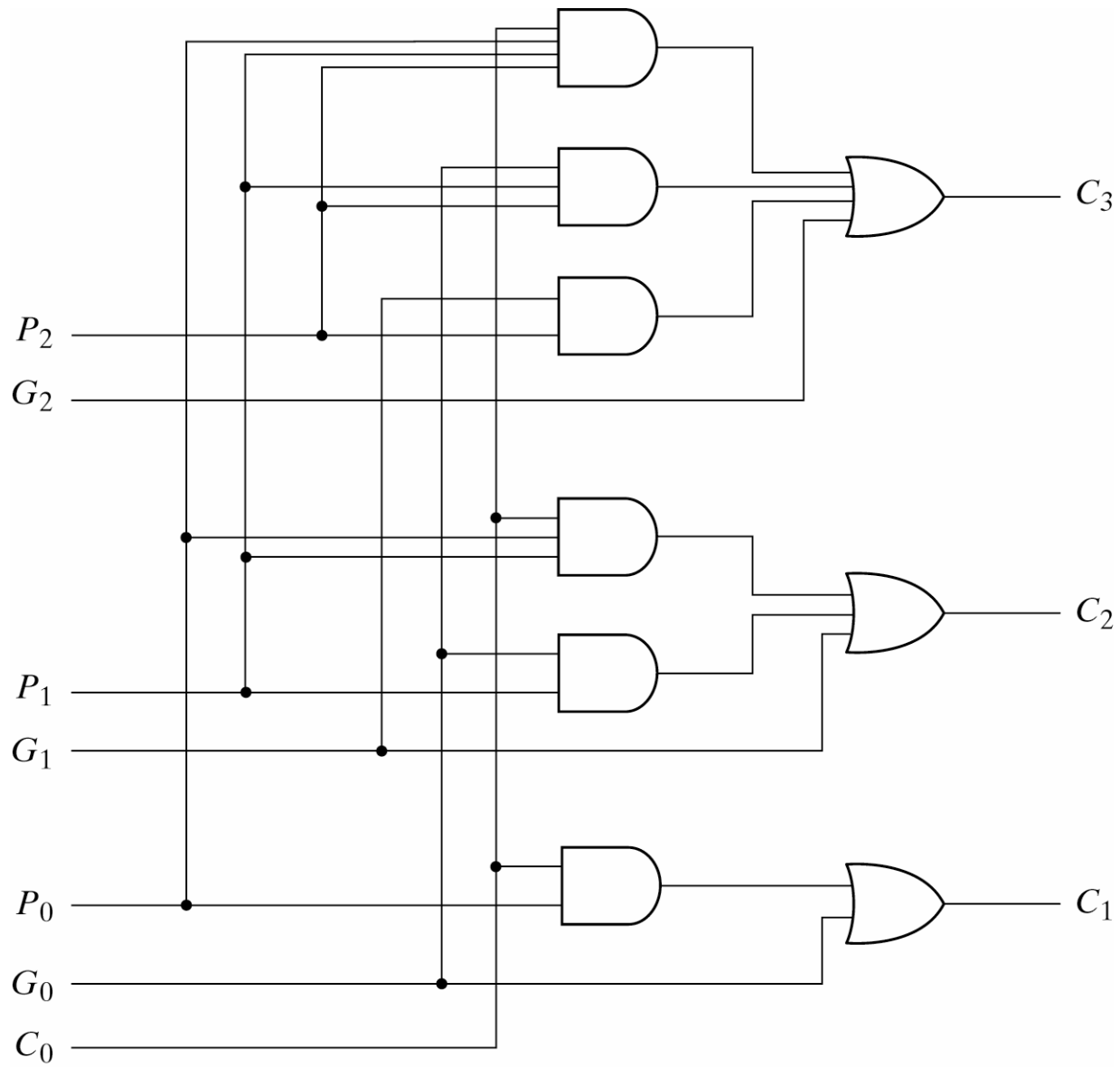


Fig. 4-11 Logic Diagram of Carry Lookahead Generator

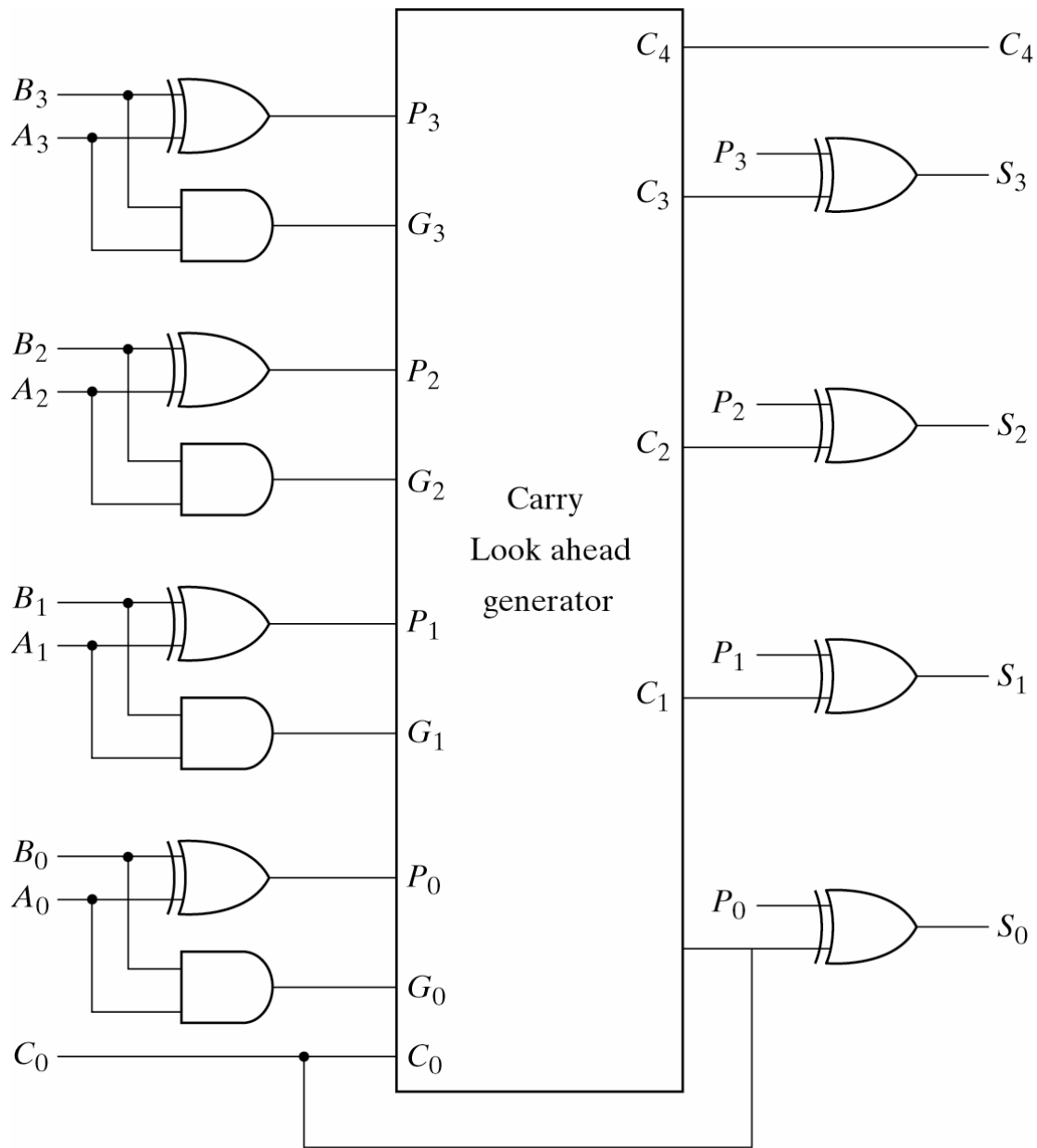


Fig. 4-12 4-Bit Adder with Carry Lookahead

İkili Çıkarıcılar

- Bir önceki kısımda görmüş olduğumuz n-bitlik ikili toplayıcı devresini, üzerinde bazı değişiklikler yaparak, hem toplama hem de çıkarma işlemi için kullanabiliriz
- Bu amaçla toplayıcı devresinin üzerine Ex-OR kapıları eklenir
- Böylelikle çıkarma işlemi yapılacağı zaman sayının ikiye tümleyenini alınır ve toplayıcıya gönderilir

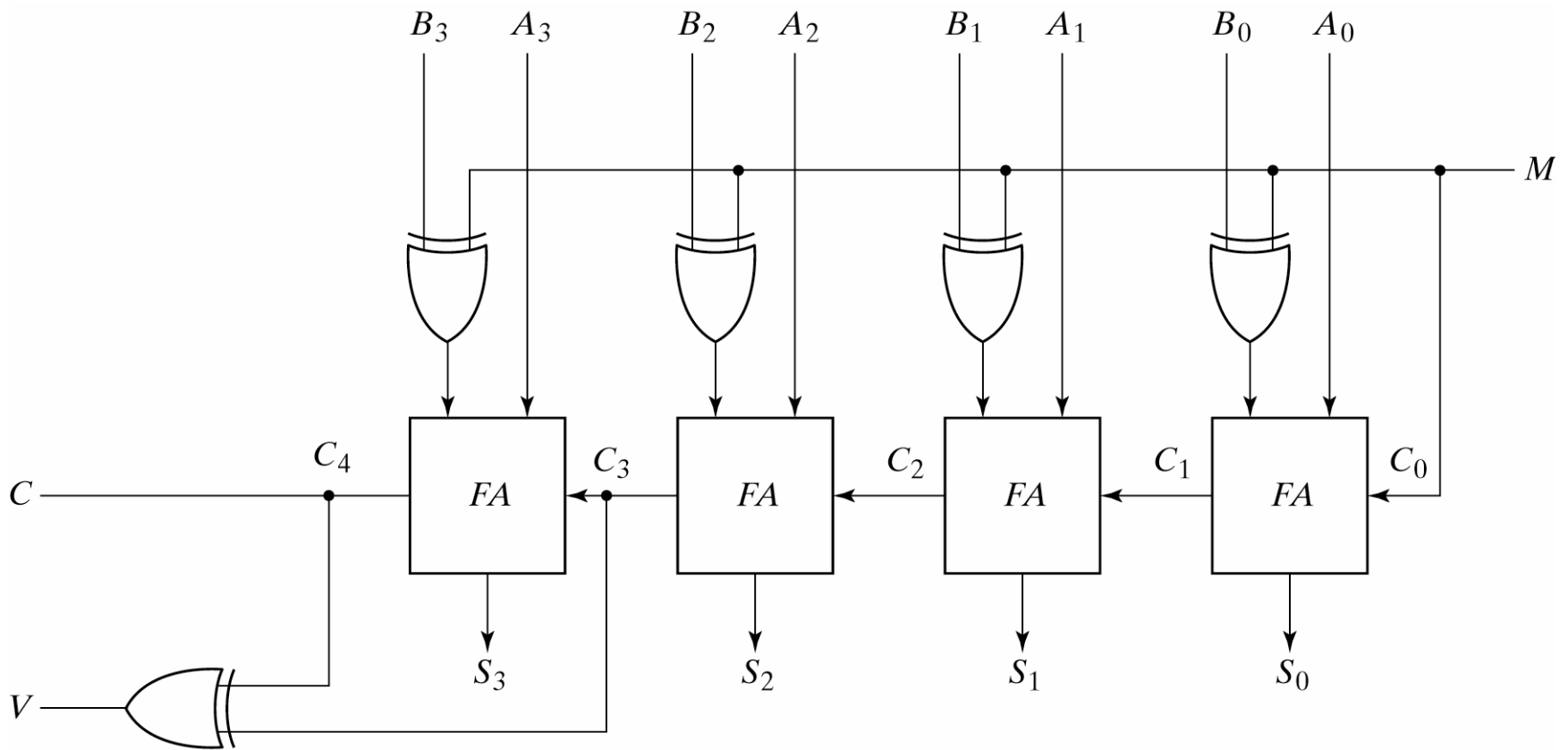


Fig. 4-13 4-Bit Adder Subtractor

Taşma Bayrağı

- n rakamlı iki sayı birbirine eklendiğinde $n+1$ basamaklı bir sayı oluşuyorsa taşma olduğu söylenir
- İşaretsiz sayılarda taşma iki adet n bitlik sayının toplanması durumunda sonucun $n+1$ bit olması durumunda gerçekleşir
- İşaretli sayılarda ise taşma iki n -bitlik sayı toplandığında sonucun n -bitin temsil ettiği sayıları aşması durumunda olur
- Örn: 2 adet 8-bitlik sayı toplandığında sonucun $+127$ 'den büyük veya -128 'den küçük ise taşma oluşmuş demektir

Taşma Bayrağı

- Taşma şu şekilde anlaşılır:
- İşaret bitine gelen elde ile işaret bitinden çıkan elde aynı ise taşma oluşmamış, farklı ise taşma oluşmuştur

		0 1	V = 1
70		0 1 0 0 0 1 1 0	
+ 80		+ 0 1 0 1 0 0 0 1	
150		1 0 0 1 0 1 1 0	

		1 0	V = 1
-70		1 0 1 1 1 0 1 0	
+ -80		+ 1 0 1 1 0 0 0 0	
-150		0 1 1 0 1 0 1 0	

Taşma Bayrağı

- Taşma bayrağı da göz önüne alındığında Figür 4.13'teki toplayıcı devresindeki C ve V çıkışları şunları belirtir:
- A ve B sayıları işaretsiz sayılar olarak düşünülürse C çıkışı sayıların toplamı sonucunda oluşabilecek eldeyi ve çıkarma işlemi sonucunda oluşabilecek borcu gösterir

Taşma Bayrağı

- A ve B sayıları işaretli sayılar olarak düşünülürse V çıkışı oluşabilecek taşmayı gösterir. Bu durumda eğer $V=0$ ise toplama veya çıkarma işlemi sonucunda taşma oluşmadığını belirtir (yani n bitlik işlemin gerçek sonucunu belirtir). Eğer $V=1$ ise taşma oluşmuş demektir. Bu durumda elde biti işaret biti olarak alınırsa işlemin gerçek sonucu elde edilmiş olunur

BCD Toplayıcılar

- Bu devreler 2 adet BCD rakamı toplamak için kullanılır
- 9 adet girişleri (1 tanesi bir önceki basamaktan gelebilecek eldeyi belirtmek için) ve 5 çıkışı bulunur
- Bu işlem bir adet 4 bitlik ikili toplayıcı ve BCD rakamlarını toplamı 9'u geçmesi durumunda toplama 6 (0110) ekleyen bir ek devreden oluşur.Devre şu şekilde çalışmaktadır:

İkilik Tabandaki Toplam						BCD Toplam						
K	Z ₈	Z ₄	Z ₂	Z ₁		C	S ₈	S ₄	S ₂	S ₁		Onluk
0	0	0	0	0		0	0	0	0	0		0
0	0	0	0	1		0	0	0	0	1		1
0	0	0	1	0		0	0	0	1	0		2
0	0	0	1	1		0	0	0	1	1		3
0	0	1	0	0		0	0	1	0	0		4
0	0	1	0	1		0	0	1	0	1		5
0	0	1	1	0		0	0	1	1	0		6
0	0	1	1	1		0	0	1	1	1		7
0	1	0	0	0		0	1	0	0	0		8
0	1	0	0	1		0	1	0	0	1		9
0	1	0	1	0		1	0	0	1	0		10
0	1	0	1	1		1	0	0	1	1		11
0	1	1	0	0		1	0	0	0	0		12
0	1	1	0	1		1	0	0	0	1		13
0	1	1	1	0		1	0	1	1	0		14
0	1	1	1	1		1	0	1	1	1		15
1	0	0	0	0		1	0	1	0	0		16
1	0	0	0	1		1	0	1	0	1		17
1	0	0	1	0		1	1	0	1	0		18
1	0	0	1	1		1	1	0	1	1		19

BCD Toplayıcılar

- 4 bitlik toplayıcı çıkısına 6 eklenmesi gereken durumlar yani rakamların toplamı sonucu oluşan sayının 9'dan büyük olma durumunu şu şekilde belirleyebiliriz:
- $K = 1$ ise toplam 15'ten büyüktür
- Toplamın 10 ile 15 arasında olduğu durumlar Z_8 VE (Z_4 VEYA Z_2) şartının 1 olduğu durumlardır. Dolayısıyla toplam sonucu oluşan sayının 9'dan büyük olma şartını şu şekilde belirleyebiliriz
- $$C = K + Z_8 \cdot (Z_4 + Z_2) = K + Z_8 \cdot Z_4 + Z_8 \cdot Z_2$$

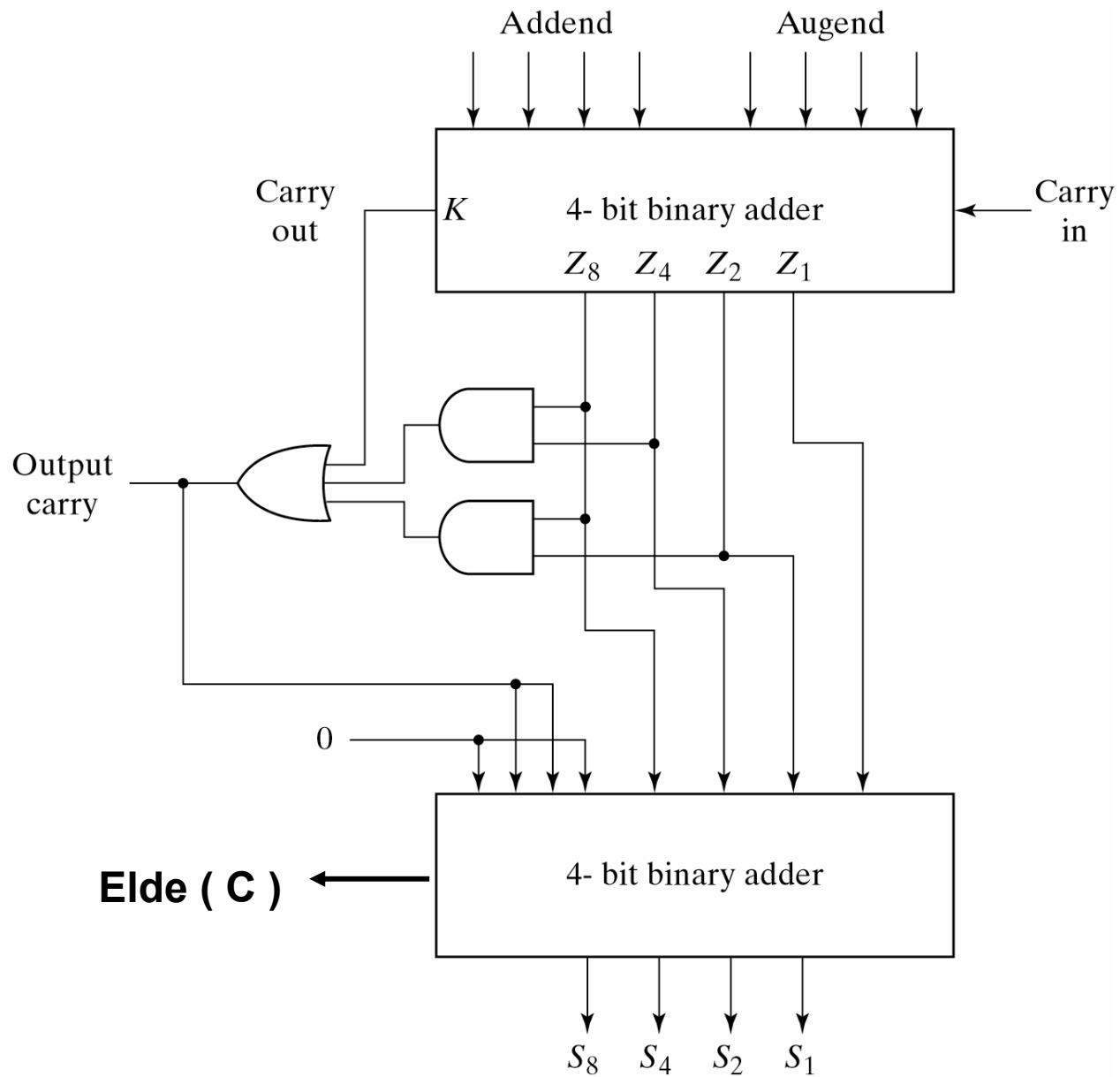


Fig. 4-14 Block Diagram of a BCD Adder

İkili Çarpıcılar

- İki bitlik iki adet sayıyı çarpan sayısal devrelere denir. Figüre 4.15'te devre şeması verilmiştir
- Figüre 4.16'da ise 4-bitlik bir sayıyı 3 bitlik bir sayı ile çarpan bir ikili çarpıcı devre verilmiştir

		B_1	B_0
	A_1	$A_1 B_1$	$A_1 B_0$
	A_0	$A_0 B_1$	$A_0 B_0$
C_3	C_2	C_1	C_0

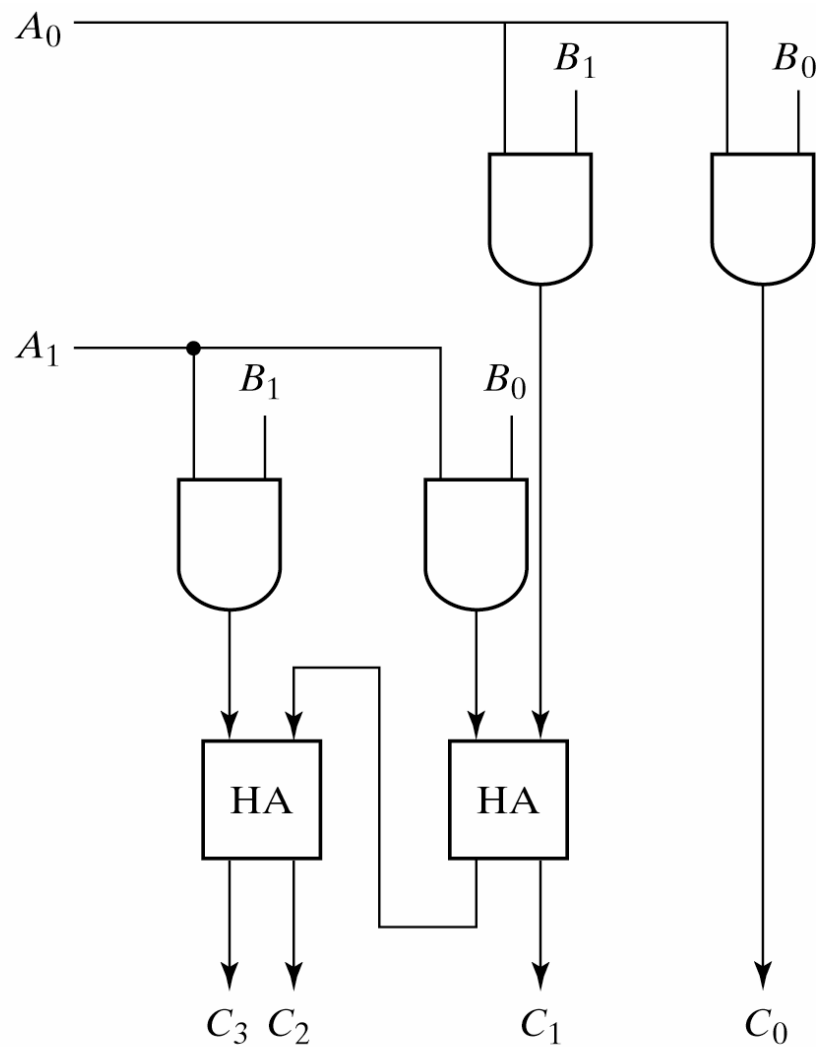


Fig. 4-15 2-Bit by 2-Bit Binary Multiplier

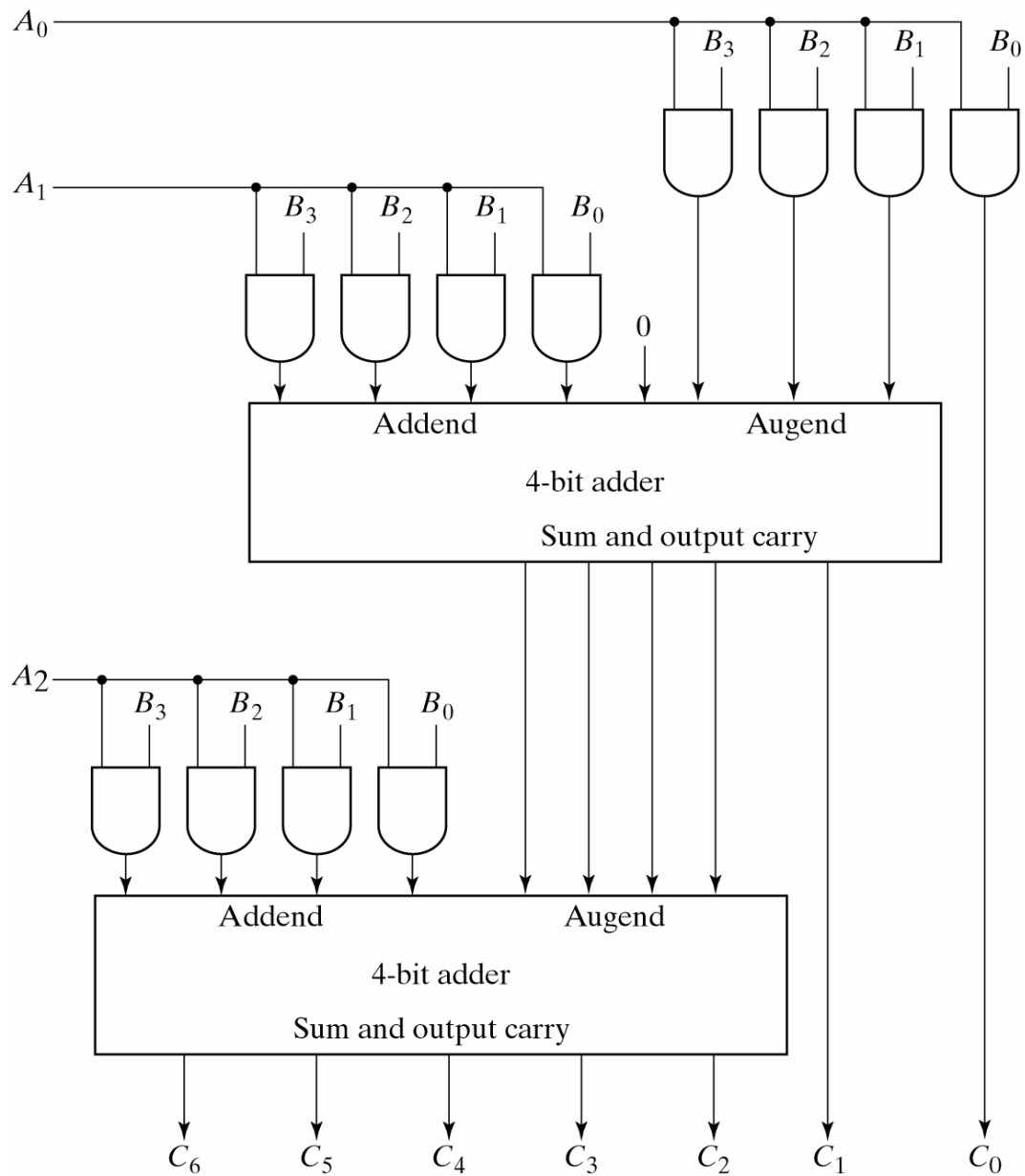


Fig. 4-16 4-Bit by 3-Bit Binary Multiplier

Genlik Karşılaştırıcı

- İki adet ikili düzendeki sayıyı birbiri ile karşılaştıran devreye karşılaştırıcı devresi denir
- Örn: $A = A_3 A_2 A_1 A_0$
 $B = B_3 B_2 B_1 B_0$

Böyle bir devrenin 9 girişi ve 3 çıkışı olacağından devre dizaynı için K-Mapler ve dolayısıyla vermiş olduğumuz 4 aşamalı dizayn prosedürü kullanılamaz

Bunun yerine karşılaştırıcı devresi Ex-NOR kapıları kullanılarak A_i ve B_i bitlerinin birbirleri ile karşılaştırılması ile gerçekleştirilebilir

Genlik Karşılaştırıcı

- $x_i = A_i \cdot B_i + A_i' \cdot B_i'$ olarak tanımlanırsa (bitler aynı olursa x_i 1 değilse sıfır olur)
- $(A == B)$ için $x_3 \cdot x_2 \cdot x_1 \cdot x_0 = 1$ olması gerekir
- $(A > B)$ için $A_3 \cdot B_3' + x_3 \cdot A_2 \cdot B_2' + x_3 \cdot x_2 \cdot A_1 \cdot B_1' + x_3 \cdot x_2 \cdot x_0 \cdot A_0 \cdot B_0' = 1$ olması gerekir
- $(A < B)$ için $A_3' \cdot B_3 + x_3 \cdot A_2' \cdot B_2 + x_3 \cdot x_2 \cdot A_1' \cdot B_1 + x_3 \cdot x_2 \cdot x_0 \cdot A_0' \cdot B_0 = 1$ olması gerekir

A ₃	A ₂	A ₁	A ₀	0	1	0	0
B ₃	B ₂	B ₁	B ₀	0	0	1	1

				X ₃	X ₂	X ₁	X ₀
				1	0	0	0

(A = B) için $x_3.x_2.x_1.x_0 = 0$

(A > B) için $A_3.B_3' + x_3.A_2.B_2' + x_3.x_2.A_1.B_1' + x_3.x_2.x_0.A_0.B_0'$
 $= 0.1 + 1.1.1 + 1.0.0.0 + 1.0.0.0.0 = 1$

(A < B) için $A_3'.B_3 + x_3.A_2'.B_2 + x_3.x_2.A_1'.B_1 + x_3.x_2.x_0.A_0'.B_0$
 $= 1.0 + 1.0.0 + 1.0.1.1 + 1.0.0.1.1 = 0$

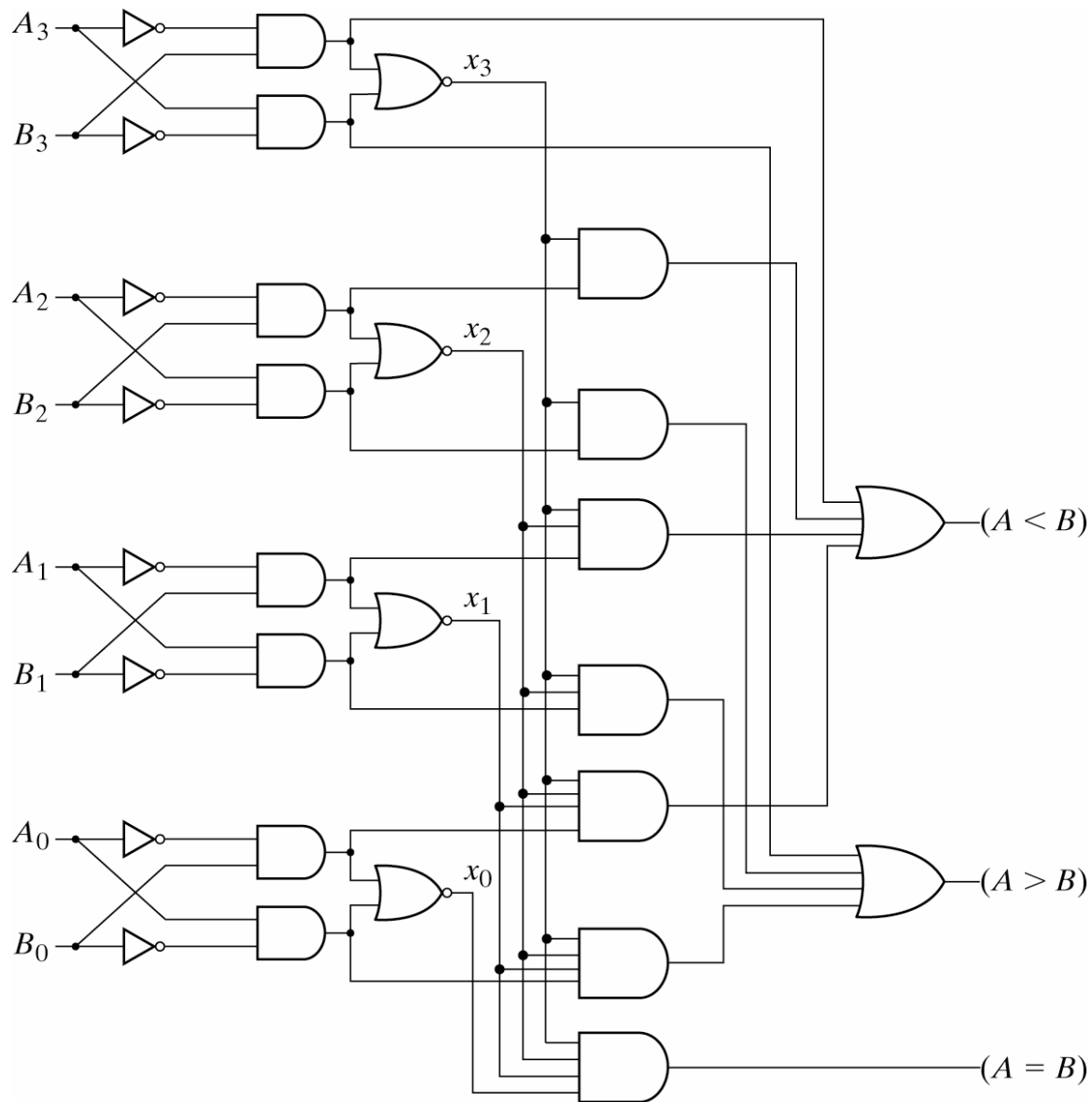


Fig. 4-17 4-Bit Magnitude Comparator

Kod Çözücüler

- n tane girişi ve 2^n tane çıkışı olan ve n -bit olarak kodlanmış bir bilgiyi 2^n tane çıkıştan birine gönderen sayısal devrelere kod çözücüler denir
- Kod çözücü dizaynı:

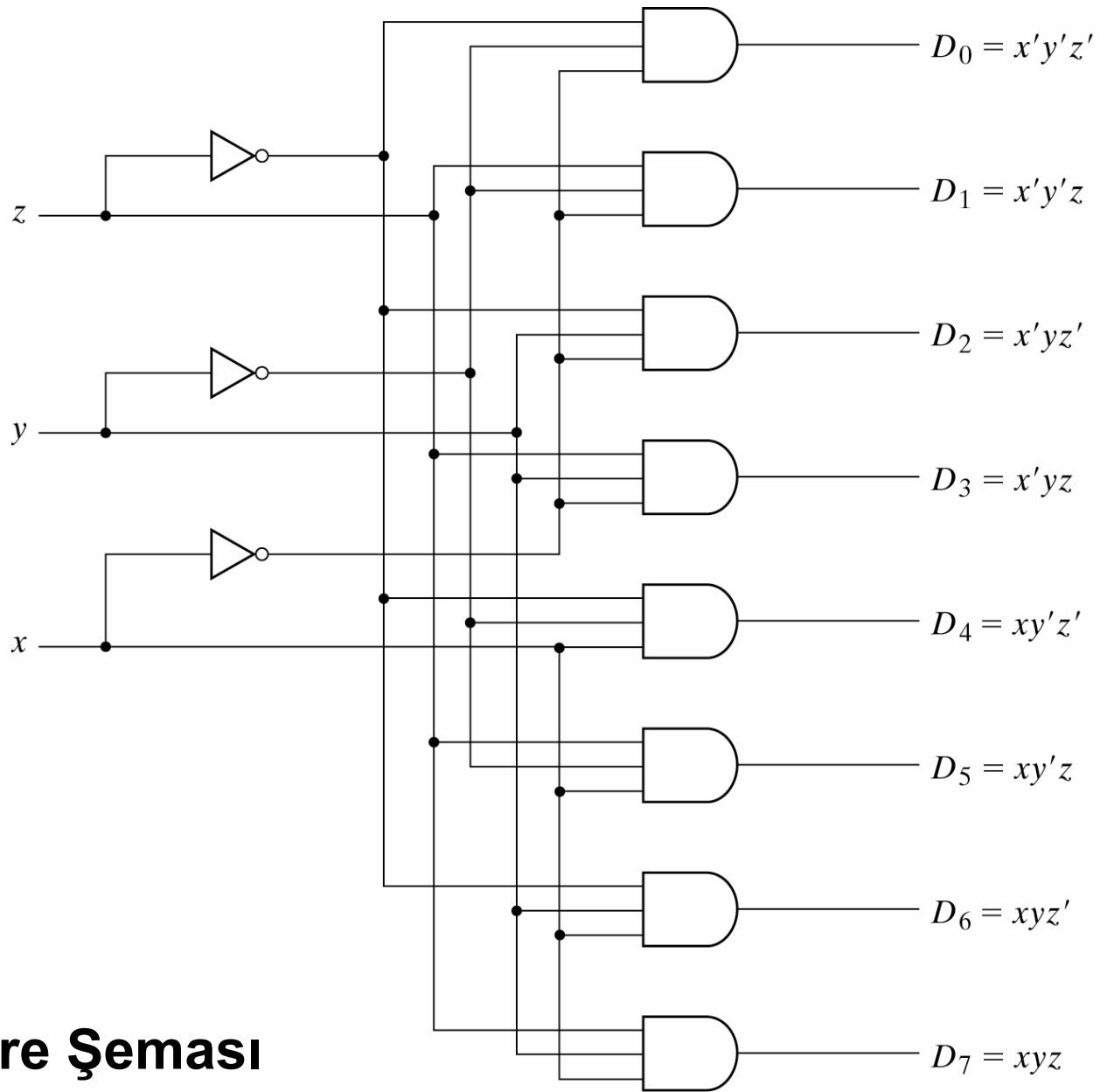
Kod Çözücüler

1. Devrenin n tane girişi ve 2^n tane çıkışı vardır
2. Devrenin doğruluk tablosu bir sonraki figürde verilmiştir
3. Doğruluk tablosuna bakıldığında K-Mapler'in kullanılmasına gerek olmadığı görülür
4. Devrenin şeması bir sonraki figürde verilmiştir

Kod Çözücüler

2. Doğruluk Tablosu

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



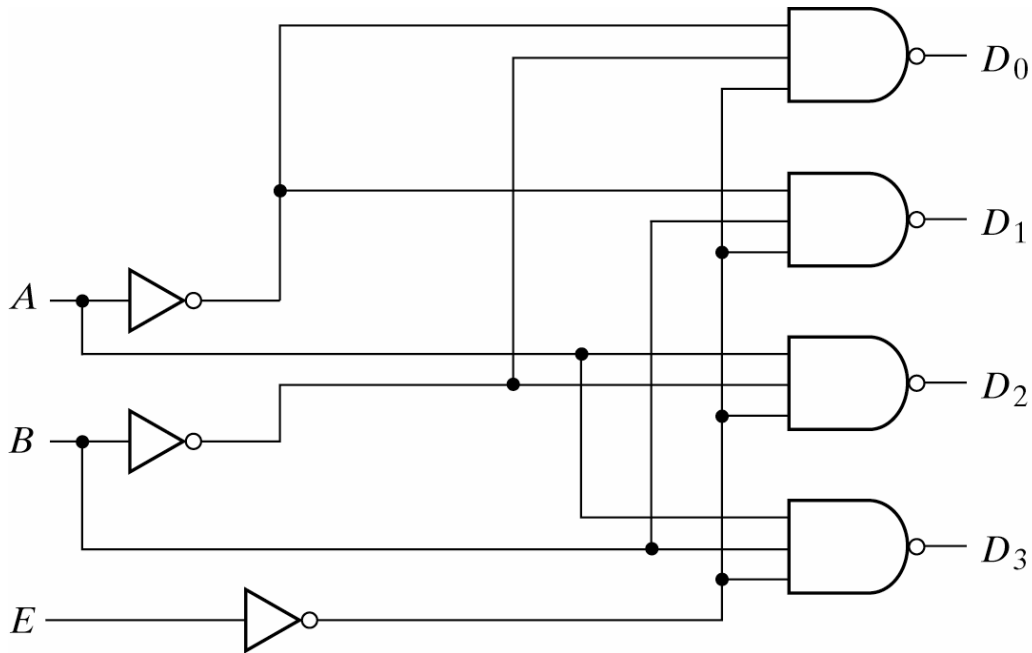
4. Devre Şeması

Fig. 4-18 3-to-8-Line Decoder

Kod Çözücüler

- Çıkışlara dikkat edilirse her bir çıkışın bir mintörmü belirttiği görülür
- Genellikle sayısal devre içerisindeki entegre devrelerin aktif yada pasif duruma getirilmesi için kullanılır
- Birden fazla kod çözücü yan yana getirilerek daha büyükleri inşa edilebilir

Kod Çözücüler



(a) Logic diagram

E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

(b) Truth table

Fig. 4-19 2-to-4-Line Decoder with Enable Input

Kod Çözücüler

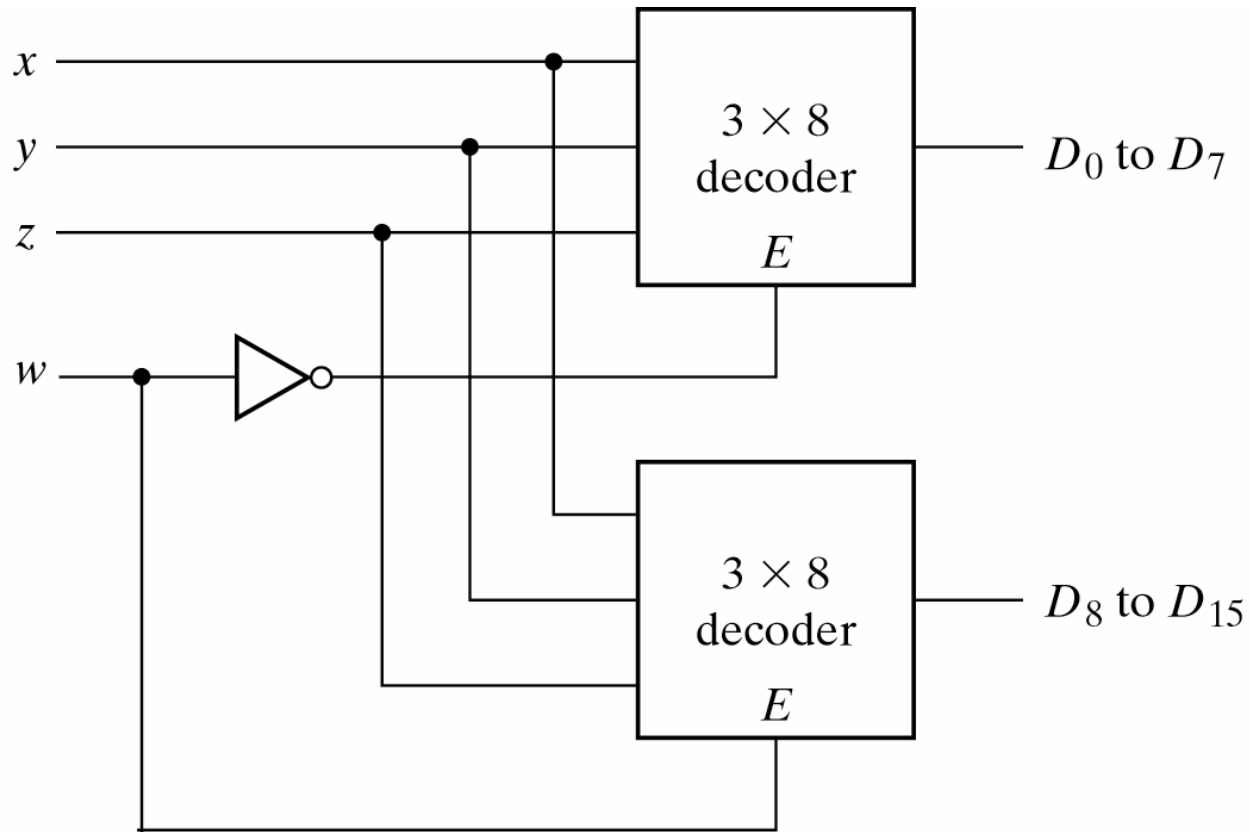


Fig. 4-20 4×16 Decoder Constructed with Two 3×8 Decoders

Kod Çözücüler

- Kod çözücüler aynı zamanda Boolean fonksiyonlarını gerçeklemek içinde kullanılır
- Bu durumda fonksiyonlar en sade şekilleri ile değil doğruluk tablolarında 1'e karşılık gelen mintörmlerinin toplamı olarak ifade edilirler

Kod Çözücüler

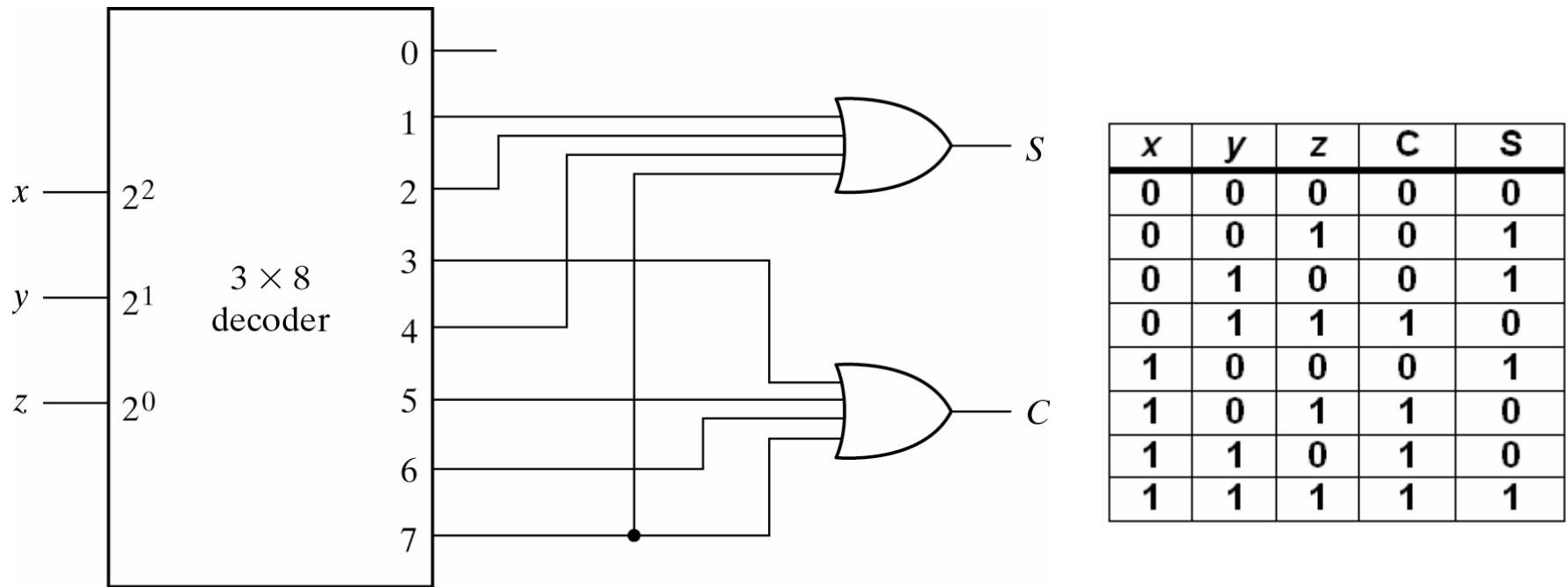


Fig. 4-21 Implementation of a Full Adder with a Decoder

Kodlayıcılar

- Kodlayıcılar kod çözücülerin yaptığı işlevin tersini gerçekleştirirler.
- Bir başka deyişle 2^n tane giriş ve n çıkışı olan devre elemanlarıdır

Kodlayıcılar

GİRİŞLER								ÇIKIŞLAR		
D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$$x = D_4 + D_5 + D_6 + D_7$$

$$y = D_2 + D_3 + D_6 + D_7$$

$$z = D_1 + D_3 + D_5 + D_7$$

Kodlayıcılar

- Yukarıdaki devrenin dizaynı için K-Mapleri kullanamadığımıza dikkat ediniz
- D_3 ve D_6 birlikte 1 olursa x, y, z ' nin üçününde çıkışları 1 olur
- Tüm girişler 0 olursa ($D_3 = 0$ 'da olduğu gibi) çıkışlar 0 0 0 olur
- Bu durumları önleyecek şekilde bir kodlayıcı dizaynı şu şekilde yapılabilir (Not: K-map'lerin kullanılabilmesi için devre dizaynı 4 girişli bir kodlayıcı için yapılacaktır)

Kodlayıcı Dizaynı

1. Devrenin D_0 , D_1 , D_2 ve D_3 adlı 4 girişi ve x , y , V olarak isimlendirilecek 3 çıkışı vardır (V çıkışı tüm girişlerin sıfır olduğu durumu algılamak için tasarlanmıştır)
2. Devrenin doğruluk tablosu bir sonraki figürde verilmiştir
3. K-Mapler kullanılarak bulunan ve çıkışları gösteren en sade Boolean fonksiyonları bir sonraki figürde verilmiştir
4. Devrenin şeması bir sonraki figürde verilmiştir

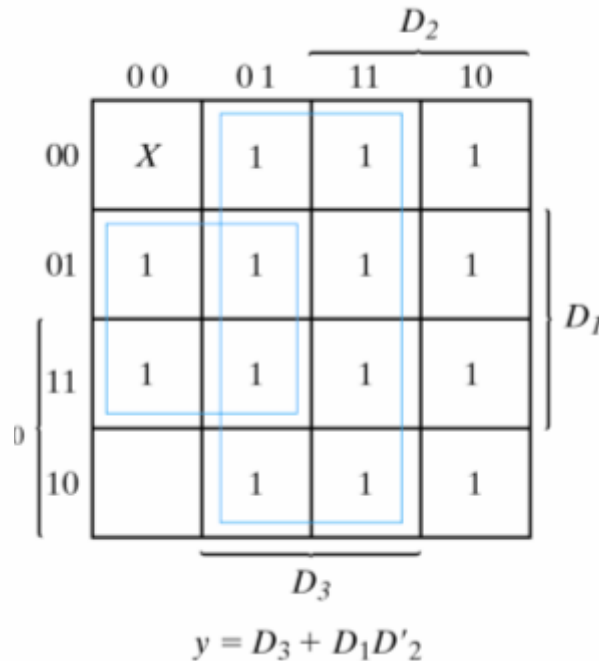
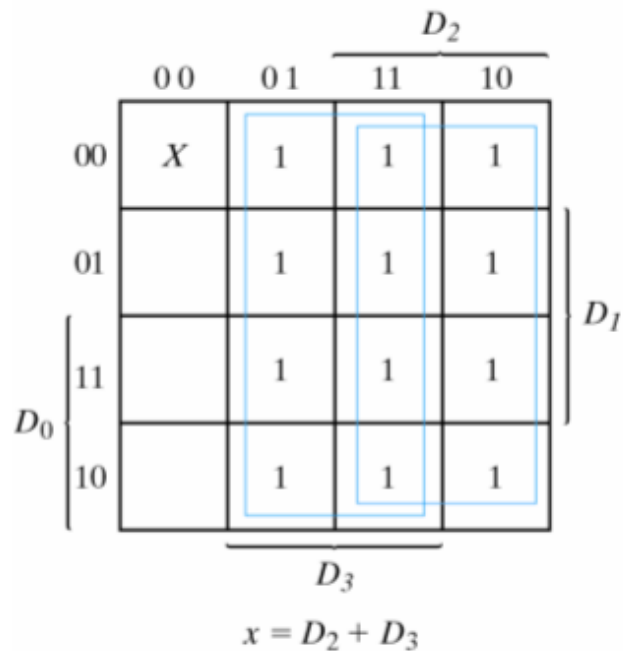
Kodlayıcı Dizaynı

2. Doğruluk Tablosu

GİRİŞLER				ÇIKIŞLAR		
D_0	D_1	D_2	D_3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

Kodlayıcı Dizaynı

3. Çıkışları Gösteren Boolean Fonksiyonlarının K-Map'ler ile Bulunması



	D_2D_3 00	D_2D_3 01	D_2D_3 11	D_2D_3 10
D_0D_1 00	0	1	1	1
D_0D_1 01	1	1	1	1
D_0D_1 11	1	1	1	1
D_0D_1 10	1	1	1	1

$$V = D_0 + D_1 + D_2 + D_3$$

Kodlayıcı Dizaynı

4- Devre Şeması

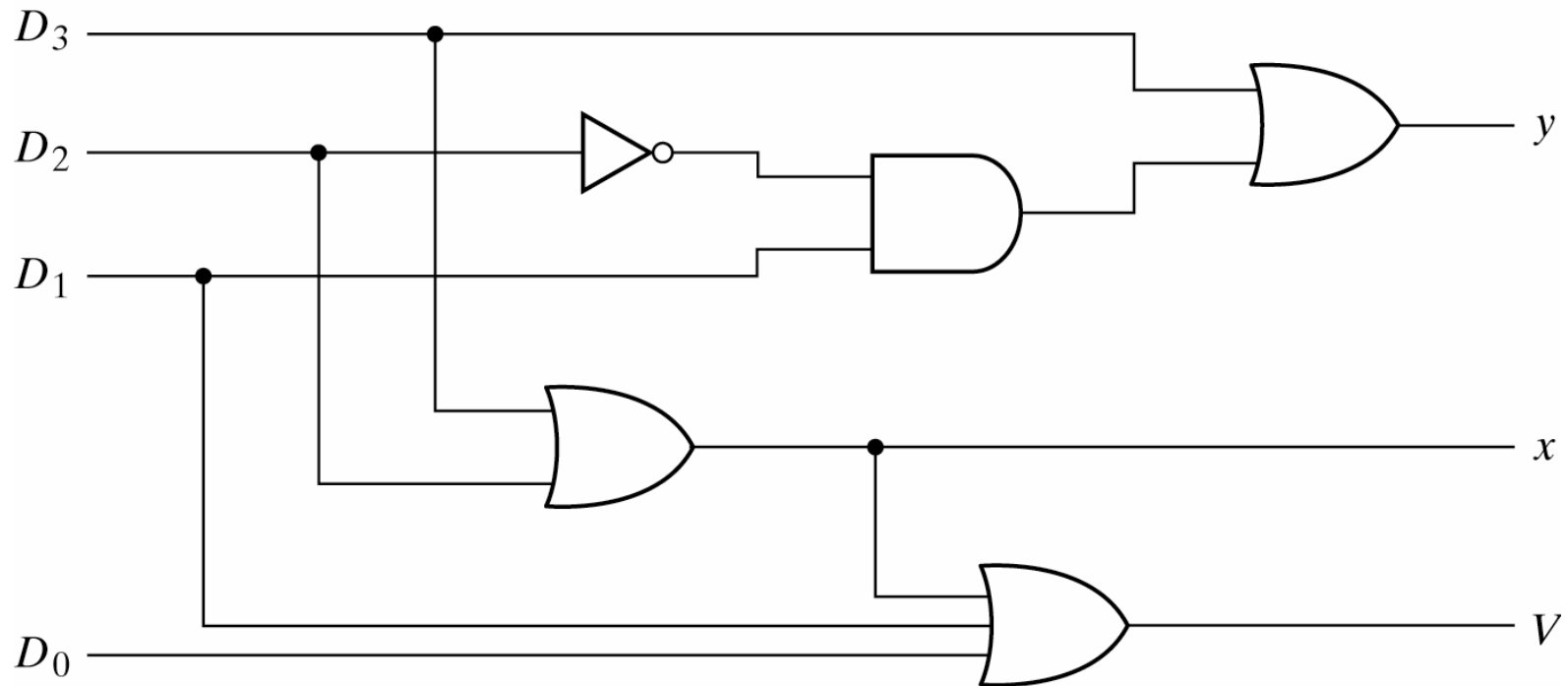
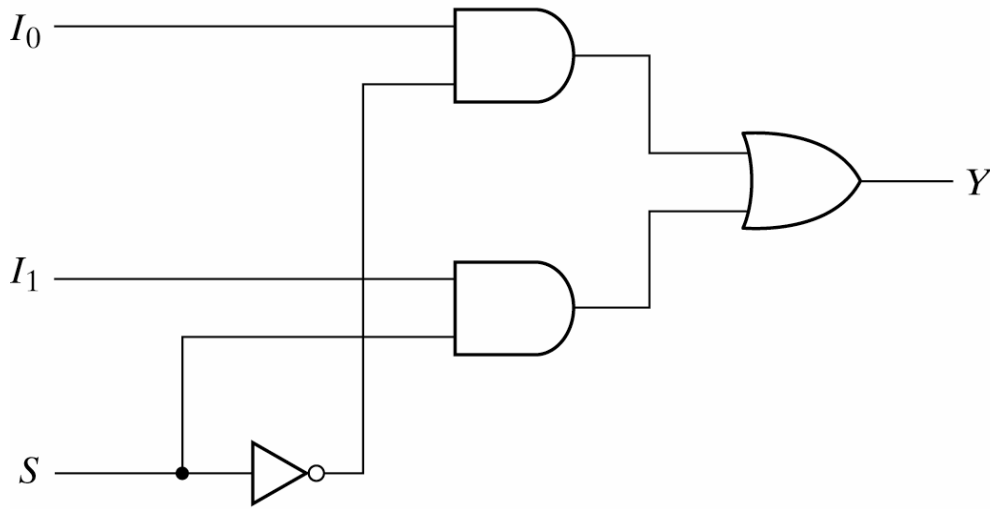


Fig. 4-23 4-Input Priority Encoder

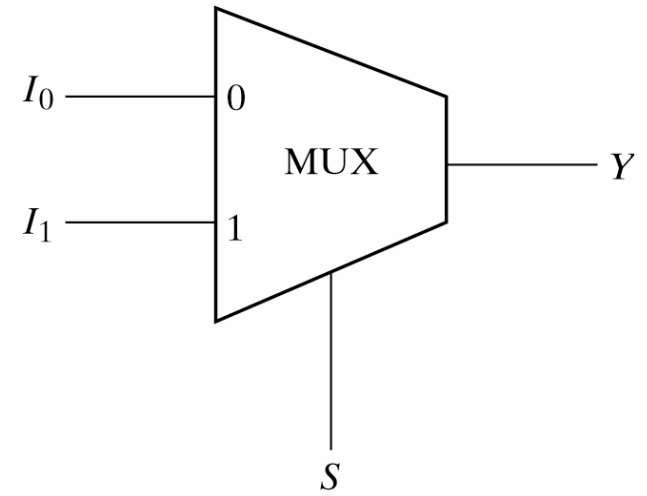
Seiciler

- Seiciler n tane seme giriři kullanarak $2n$ tane ıkıřtan 1 tanesini 1 adet ıkıřa gnderen devre elemanlarıdır.
- Seiciler giriř sayılarına gre isimlendirilirler
- rn: 2×1 , 4×1 , 8×1 , 16×1 v.s. gibi
- Kodlayıcılarda olduėu gibi seicilerinde aktif/pasif giriř olabilir

Seçiciler (2x1 Bir Seçici Devre Şeması ve Sembolü)



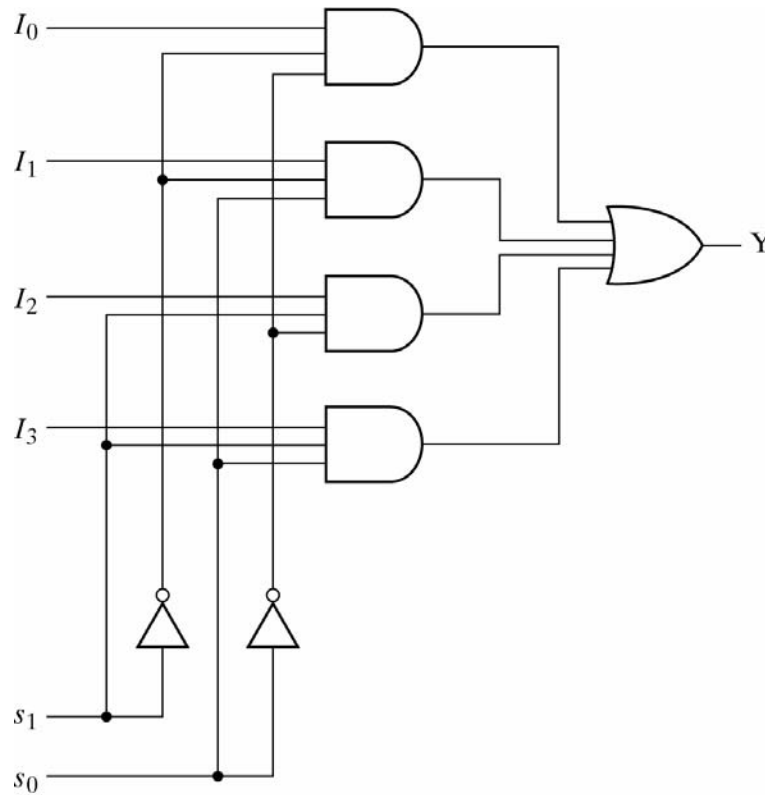
(a) Logic diagram



(b) Block diagram

Fig. 4-24 2-to-1-Line Multiplexer

Seçiciler (4x1 Bir Seçici)



(a) Logic diagram

s_1	s_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(b) Function table

Fig. 4-25 4-to-1-Line Multiplexer

4'lü 2x1 Seçici

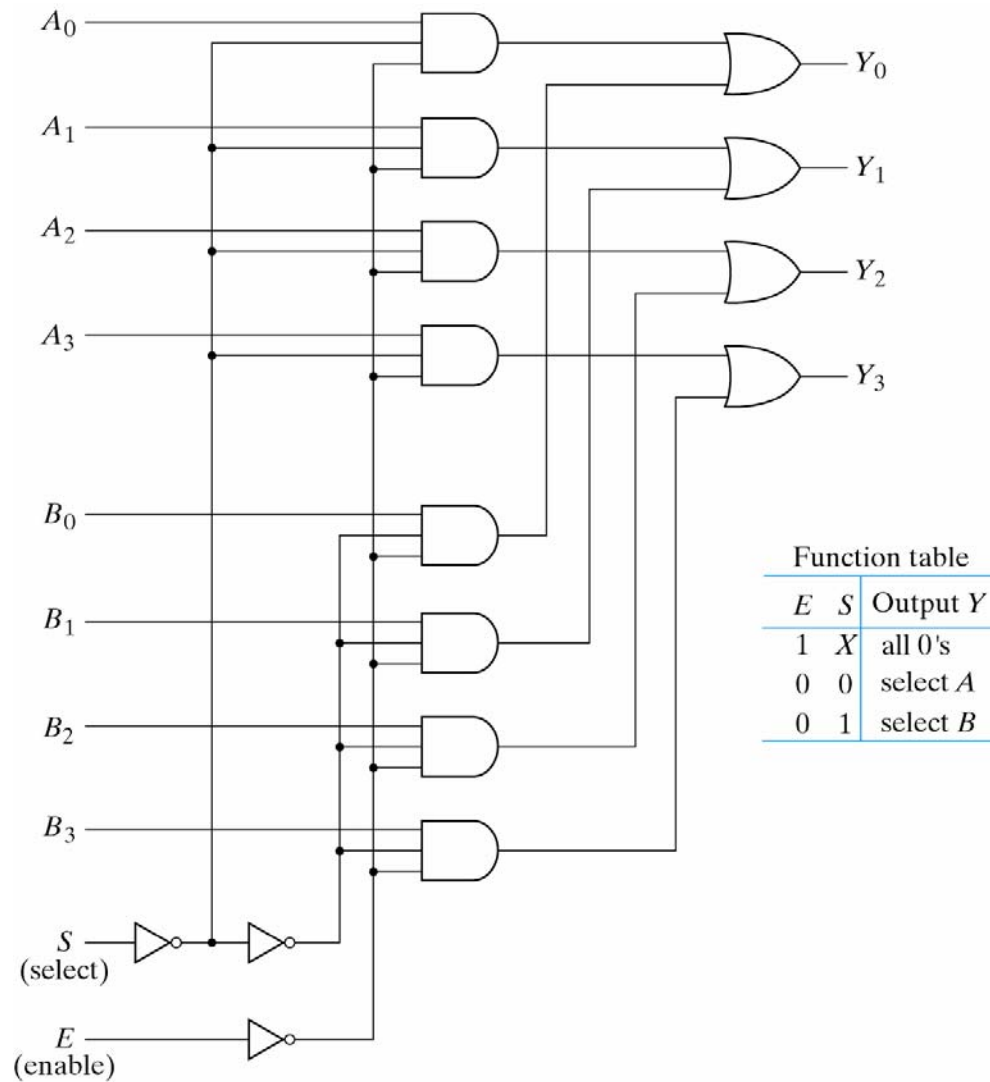


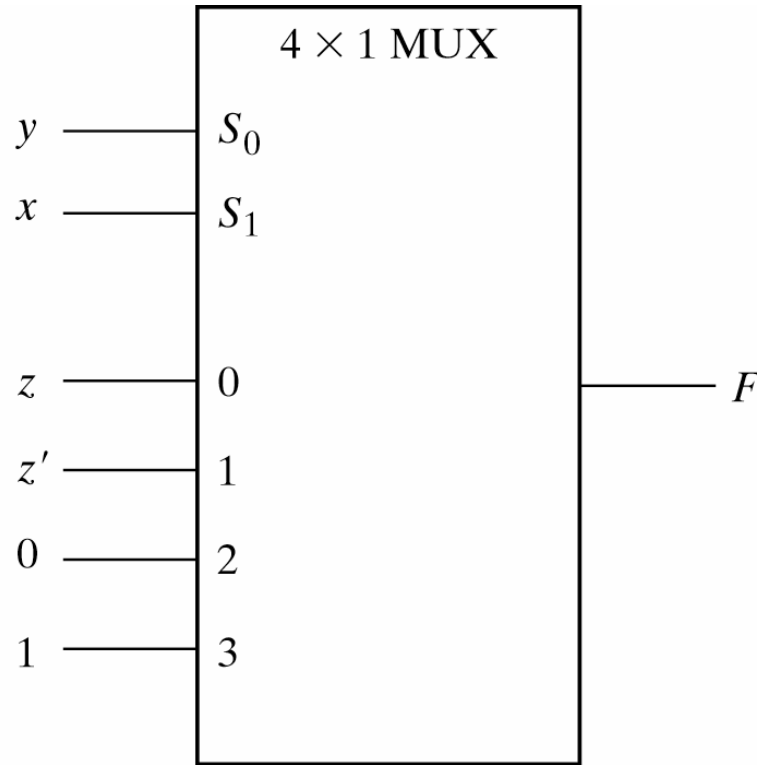
Fig. 4-26 Quadruple 2-to-1-Line Multiplexer

Boolean Fonksiyonlarının Seçiciler Kullanılarak Gerçeklenmesi

- n değişkenli bir Boolean fonksiyonu $2^{n-1} \times 1$ 'lik bir seçici ile gerçekleştirilebilir
- İşlemi gerçekleştirmek için n değişkenden $n-1$ tanesi soldan başlanarak seçme girişlerine uygulanır
- n . değişken ise (en küçük değerlikli bit) fonksiyonun doğruluk tablosunun ikişerli gruplara ayrılmasından sonra (0 ve 1 değerleri ile birlikte) kendisi veya tümleyeni olarak 2^{n-1} girişe bağlanır

x	y	z	F	
0	0	0	0	
0	0	1	1	$F = z$
0	1	0	1	
0	1	1	0	$F = z'$
1	0	0	0	
1	0	1	0	$F = 0$
1	1	0	1	
1	1	1	1	$F = 1$

(a) Truth table



(b) Multiplexer implementation

Fig. 4-27 Implementing a Boolean Function with a Multiplexer

$$F(x,y,z) = \sum (1, 2, 6, 7)$$

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>F</i>	
0	0	0	0	0	$F = D$
0	0	0	1	1	
0	0	1	0	0	$F = D$
0	0	1	1	1	
0	1	0	0	1	$F = D'$
0	1	0	1	0	
0	1	1	0	0	$F = 0$
0	1	1	1	0	
1	0	0	0	0	$F = 0$
1	0	0	1	0	
1	0	1	0	0	$F = D$
1	0	1	1	1	
1	1	0	0	1	$F = 1$
1	1	0	1	1	
1	1	1	0	1	$F = 1$
1	1	1	1	1	

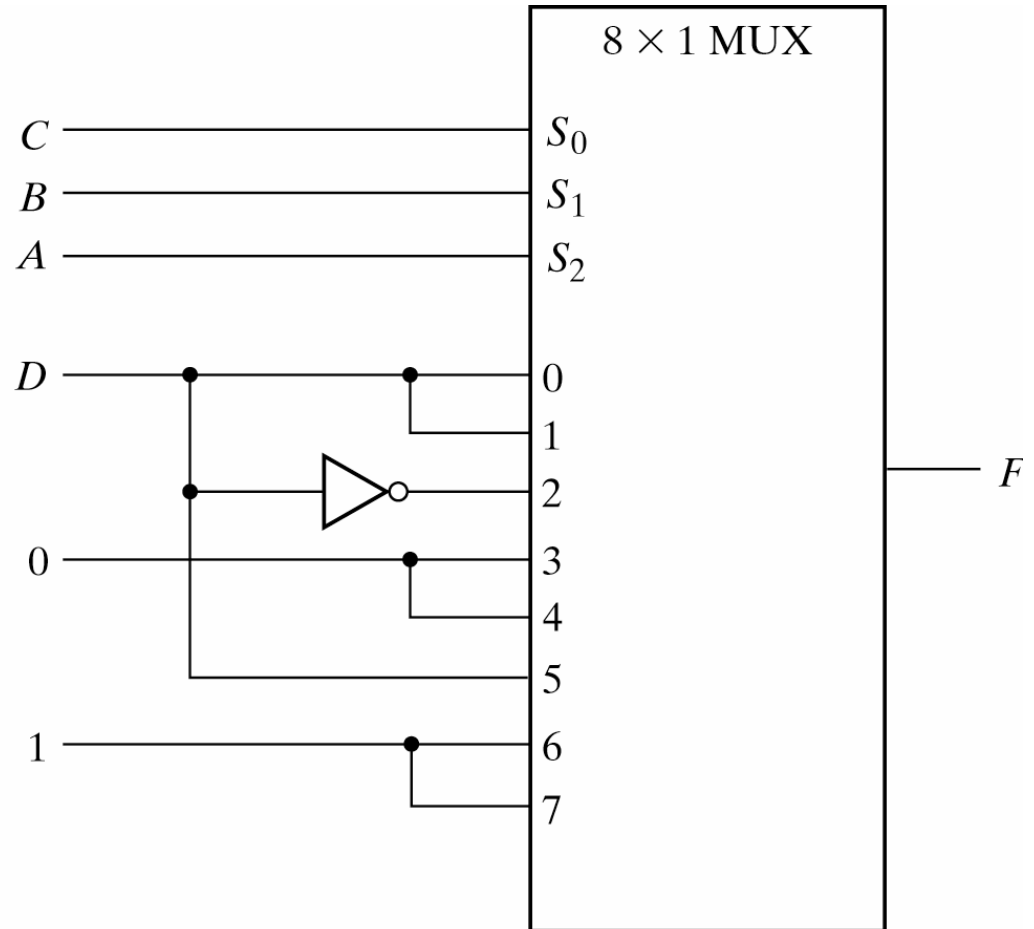


Fig. 4-28 Implementing a 4-Input Function with a Multiplexer

$$F(A,B,C,D) = \sum (1, 3, 4, 11, 12, 13, 14, 15)$$

Yüksek Empedans Çıkışları

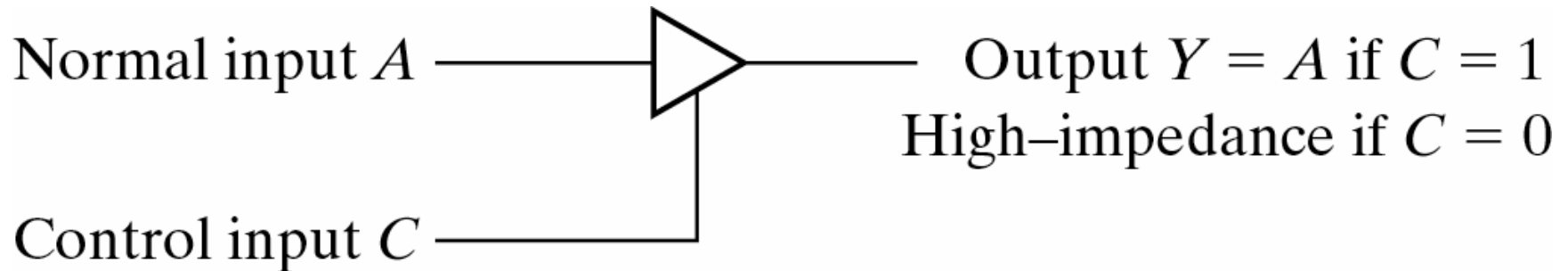
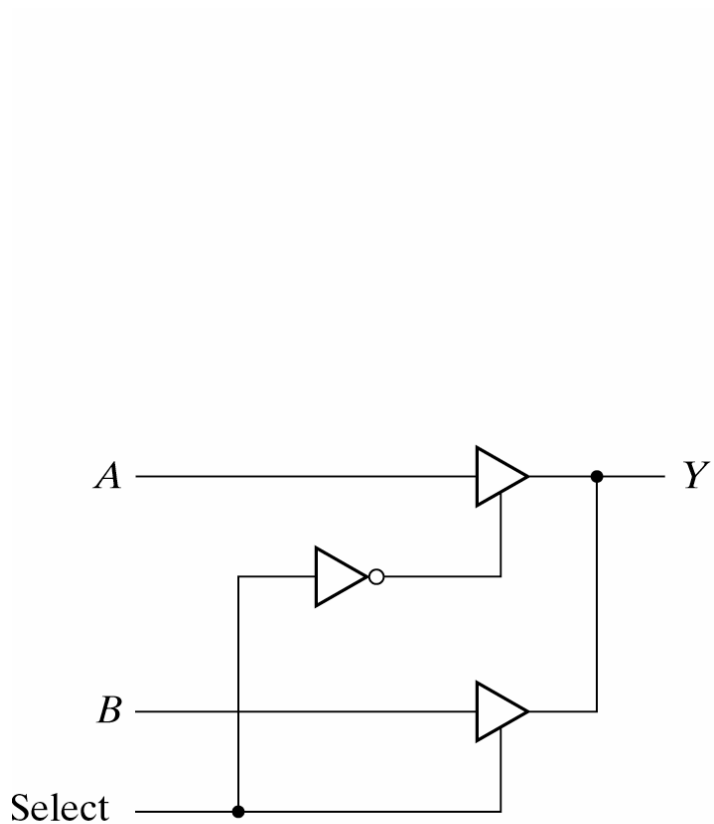
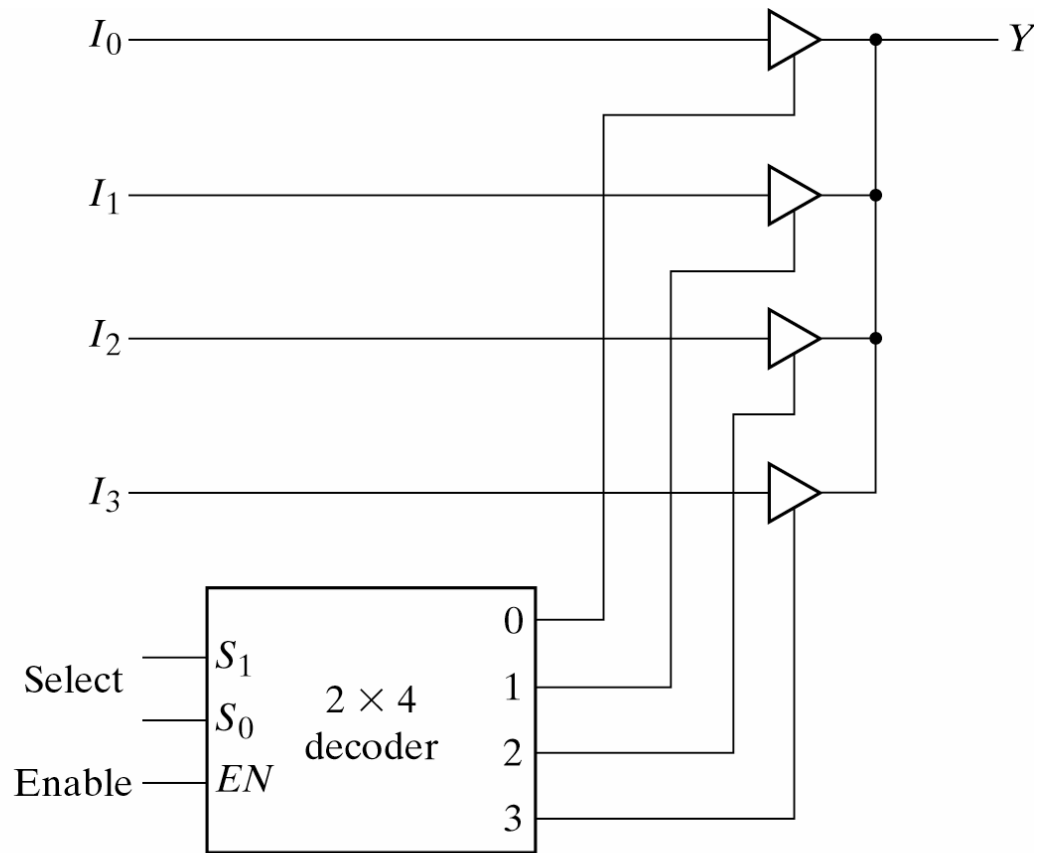


Fig. 4-29 Graphic Symbol for a Three-State Buffer



(a) 2-to-1- line mux



(b) 4 - to - 1 line mux

Fig. 4-30 Multiplexers with Three-State Gates