



# Lojik Tasarım

Ders 8

Kaynak:

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

# Kod Çözücü (Decoder)

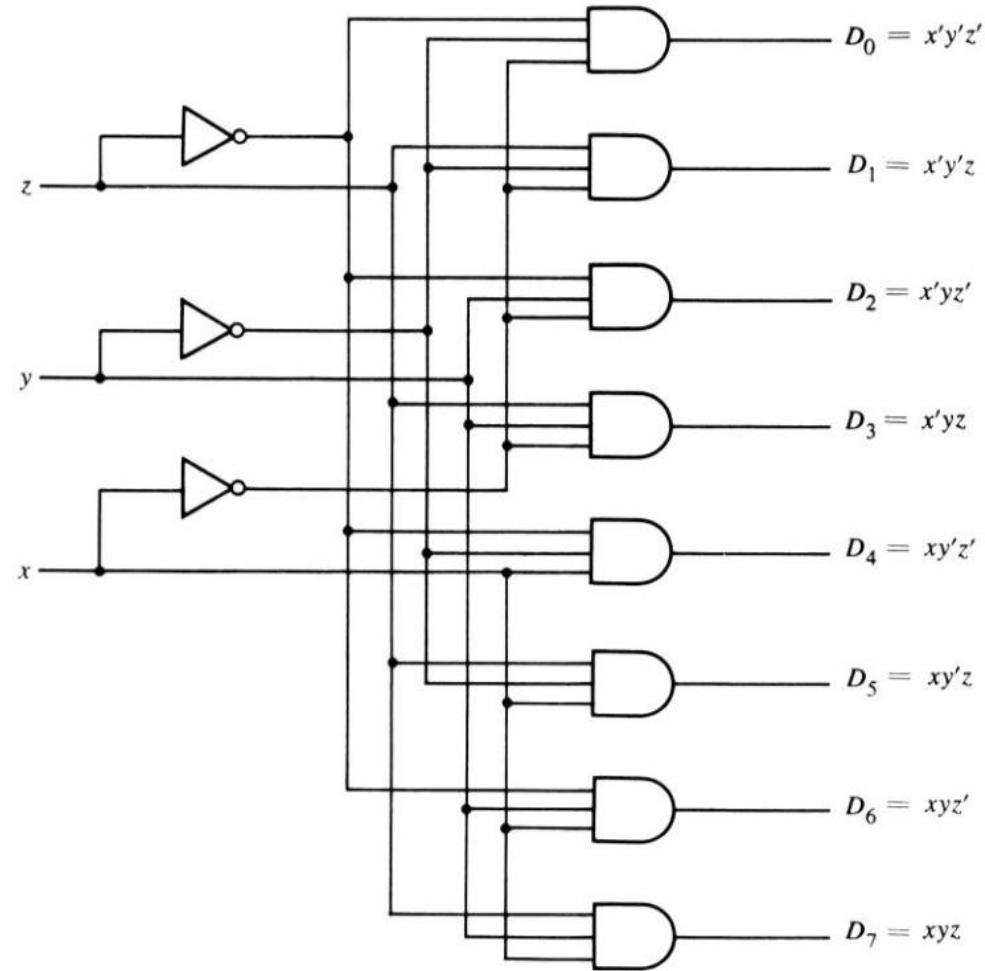
- Ayrık değerli bilgiler sayısal sistemlerde ikili kodlarla temsil edilir.  $n$  bitlik bir bilgi,  $2^n$  tane bağımsız bilgiyi temsil edebilme kapasitesine sahiptir.
- Kod çözücü  $n$  bitlik bir bilgiyi maksimum  $2^n$  çıkış hattına dönüştüren kombinezonal devredir.

# 3x8 Decoder

3'ten 8'e Kod Çözücünün Doğruluk Tablosu

Girişler			Çıkışlar							
$x$	$y$	$z$	$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$
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

# 3x8 Decoder



ŞEKİL 5-8  
3'ten 8'e bir kod çözücü

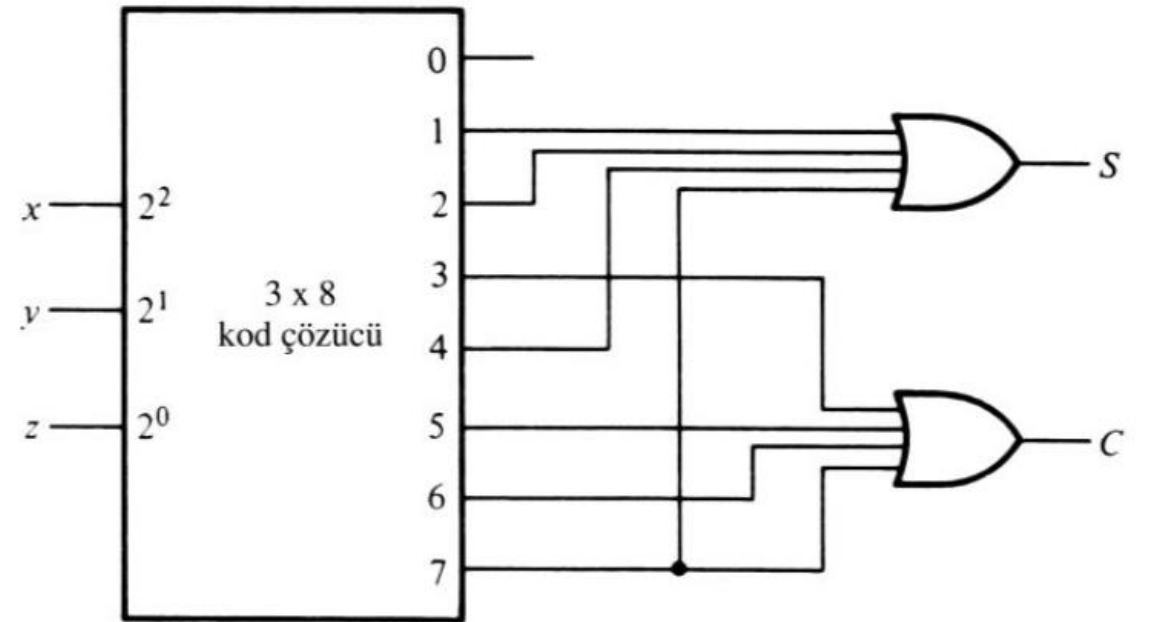
# Kombinezonal Lojik Uygulaması

Bir kod çözücü  $n$  giriş değişkeninden  $2^n$  minterim oluşturur. Herhangi bir Boole fonksiyonu kanonik formda minterimlerin toplamı biçiminde ifade edilebildiğinden, minterimleri üretmek için bir kod çözücü, toplamı biçimlendirmek için de dışardan eklenen bir VEYA kapısı kullanılabilir. Bu yolla,  $n$  giriş  $m$  çıkışlı herhangi bir kombinezonal devre,  $n$ 'den  $2^n$ 'ye kod çözücüyle  $m$  adet VEYA kapısından oluşabilir.

## Örnek: Bir tam toplayıcının kod çözücü ile tasarlanması

$$S(x, y, z) = \sum(1, 2, 4, 7)$$

$$C(x, y, z) = \sum(3, 5, 6, 7)$$



ŞEKİL 5-9

Bir tam toplayıcının kod çözücüyle gerçekleştirilmesi

# İzin Girişli 2X4 Kod Çözücü

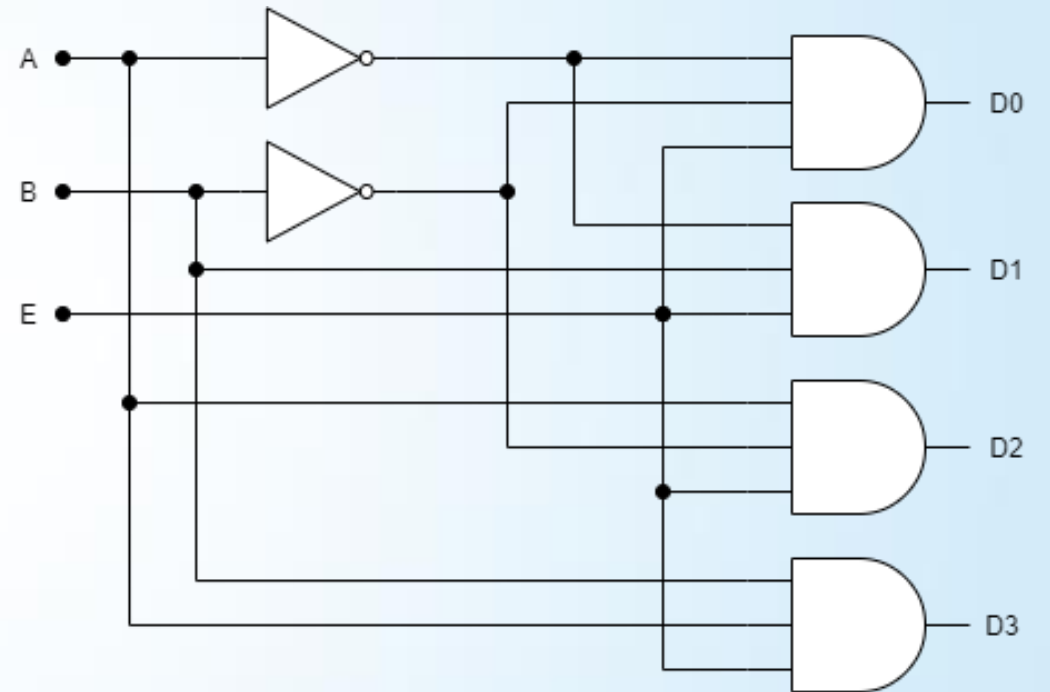
E	A	B	D0	D1	D2	D3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

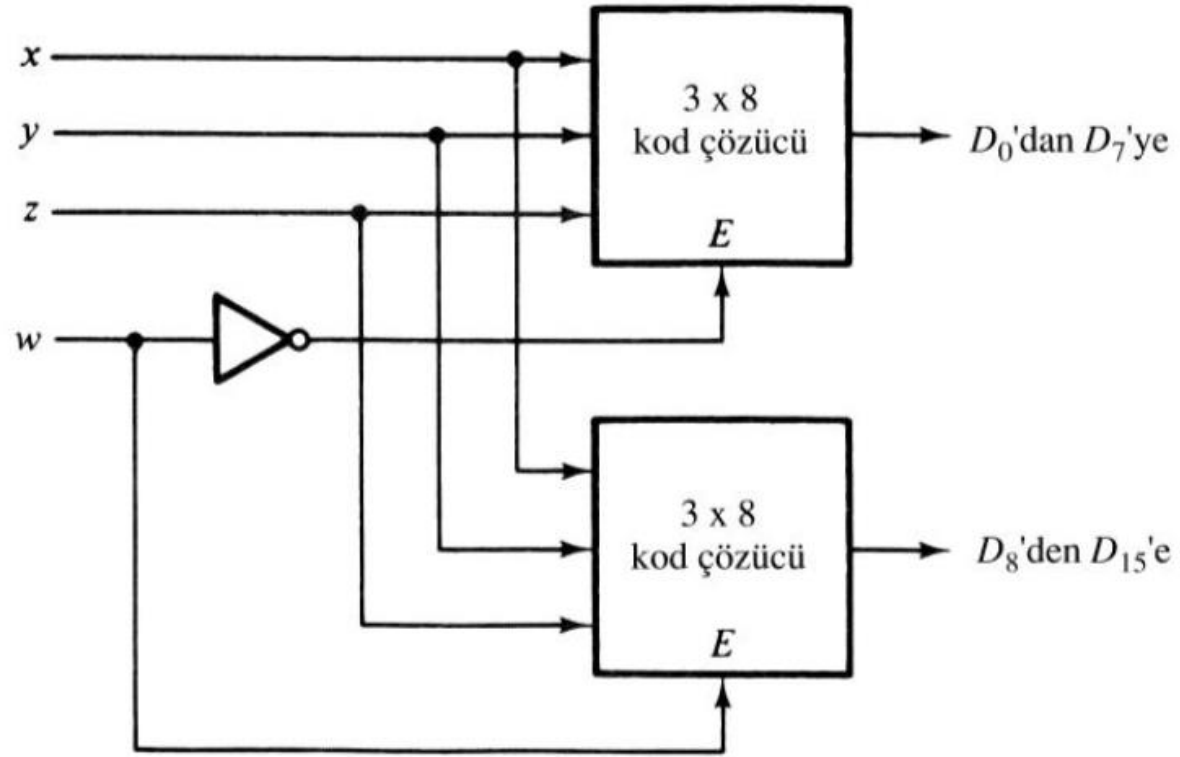
$$D0 = E A' B'$$

$$D1 = E A' B$$

$$D2 = E A B'$$

$$D3 = E A B$$





ŞEKİL 5-12

İki  $3 \times 8$  kod çözücü kullanarak bir  $4 \times 16$  kod çözücü oluşturulması

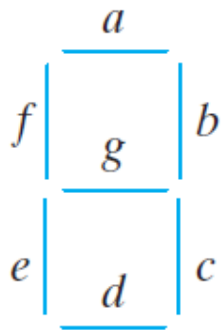


# Soru

- 4 adet  $3 \times 8$  kod çözücü kullanarak  $5 \times 32$  kod çözücü tasarlayınız

# Soru

- 7 Segment display nedir?
- BCD sayıları 7 segment displaye sürecekle 7 segment display çözücü devresini tasarlayınız.



(a) Segment designation

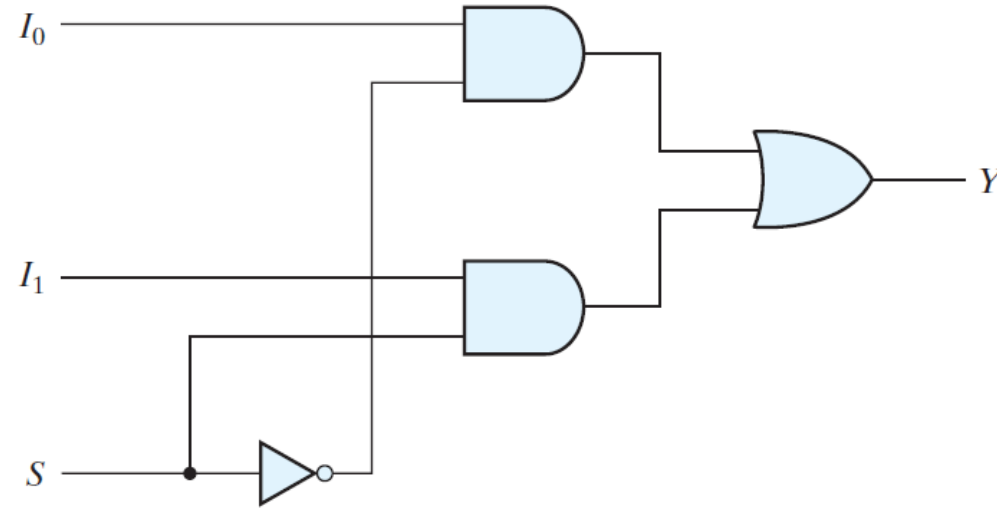


(b) Numerical designation for display

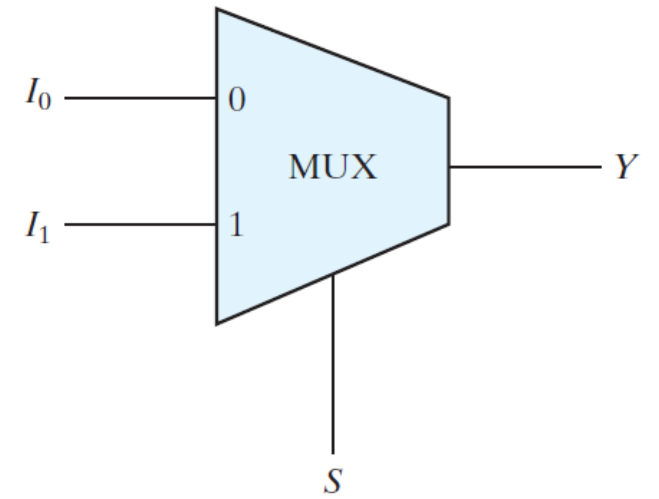
# Multiplexer (Veri Seçici)

Veri seçmenin anlamı büyük sayıda bilgi birimlerinin daha az sayıda kanallara veya yollara aktarılmasıdır. *Sayısal bir veri seçici* bir veya daha fazla giriş hattından ikili bilgiyi seçen ve bunu tek çıkış hattına bağlayan bir kombinenzonal devredir. Özel bir giriş hattının seçilmesi bir grup seçme hattıyla kontrol edilir. Normal olarak,  $2^n$  giriş hattı ve hangi girişin seçileceğini belirleyen bit kombinasyonlarını oluşturan  $n$  seçme hattı vardır.

# Multiplexer (Veri Seçici)



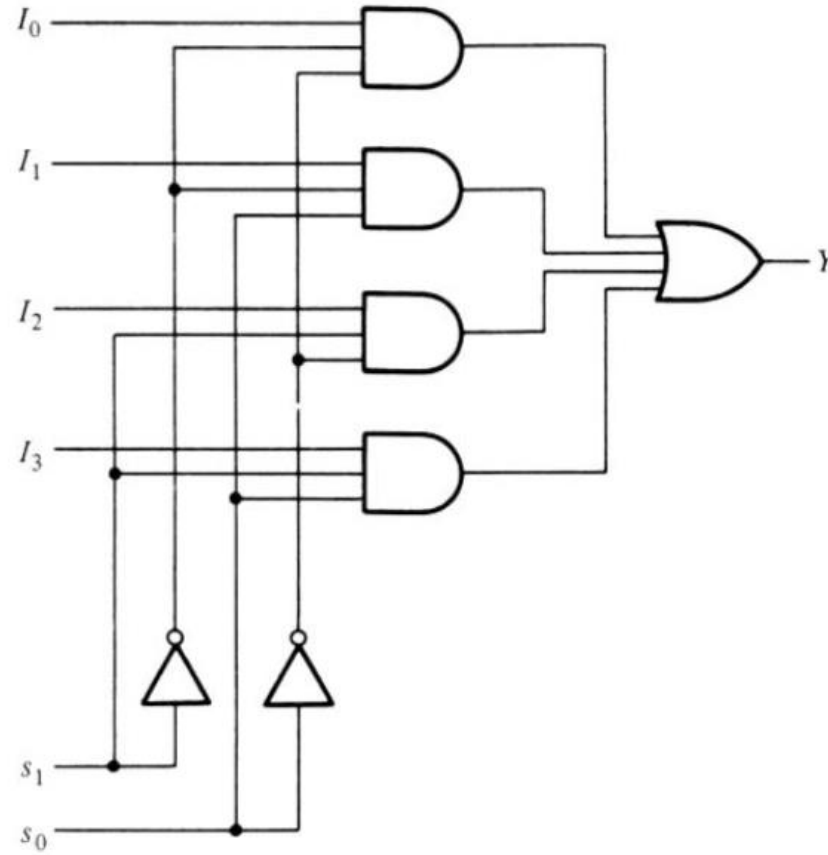
(a) Logic diagram



(b) Block diagram

**FIGURE 4.24**  
Two-to-one-line multiplexer

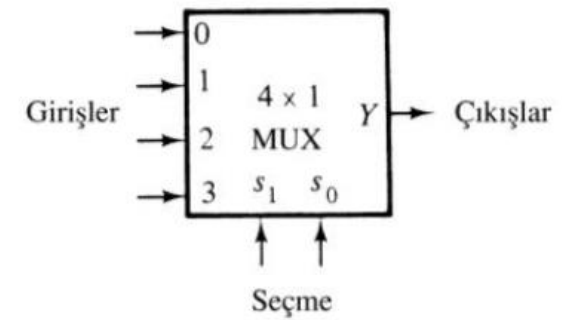
# 4 Kanallı MUX



(a) Lojik diyagram

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

(b) Fonksiyon tablosu

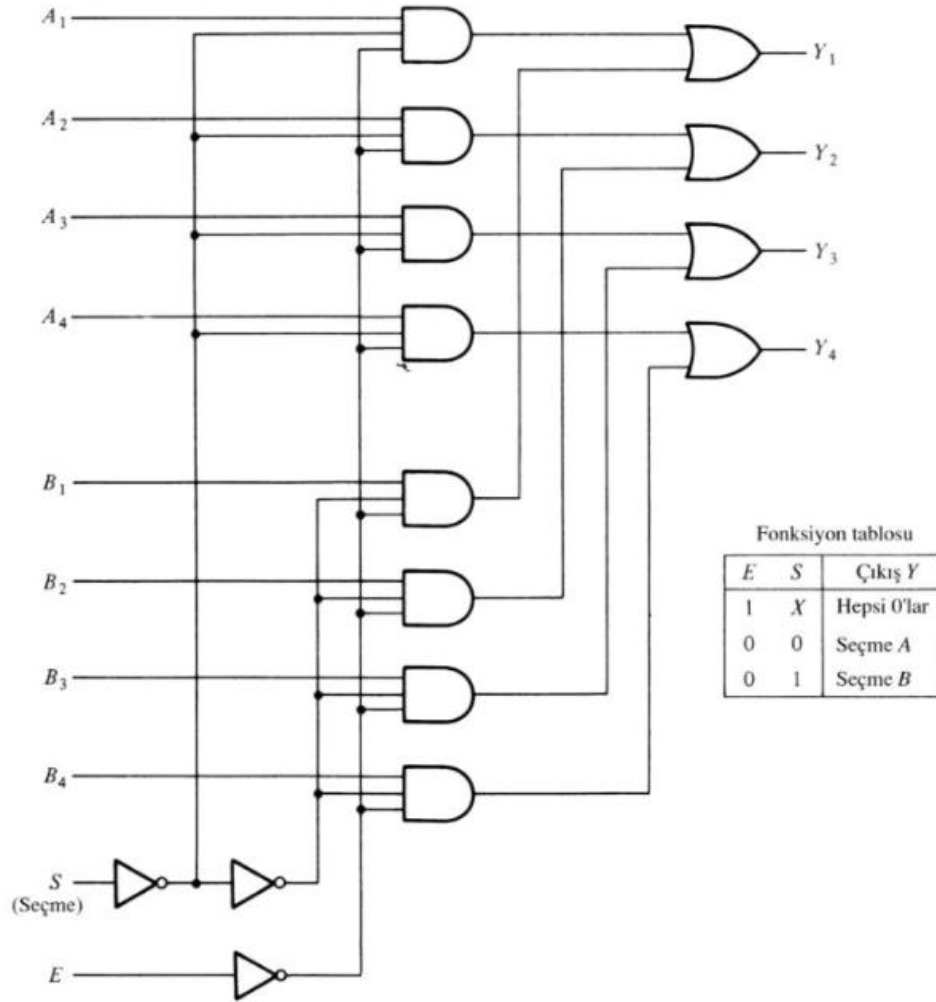


(c) Blok diyagramı

ŞEKİL 5-16

Bir 4'ten 1'e veri seçici

# Multiplexer (Veri Seçici)



ŞEKİL 5-17

Dörtlü 2'den 1'e veri seçici

# Boole Fonksiyonlarının Uygulanması

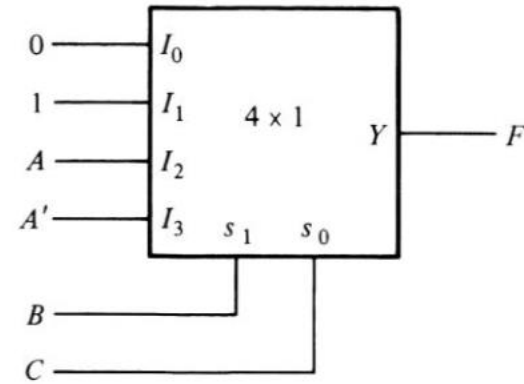
- $n$  seçicili  $2^n \times 1$  lik mux kullanarak  $n+1$  değişkenli fonksiyonlar tasarlanabilir.

## Tasarım Yöntemi

1.  $n+1$  değişkenli bir fonksiyonumuz varsa  $n$  değişkeni alınır ve seçme hatlarına bağlanır.
2. Fonksiyonun kalan tek değişkeni veri seçicinin girişine bağlanır.

# Örnek

- $f(A, B, C) = \Sigma(1, 3, 5, 6)$  fonksiyonunu MUX kullanarak gerçekleştiriniz



(a) Veri seçici gerçeklemesi

Minterim	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

(b) Doğruluk tablosu

	$I_0$	$I_1$	$I_2$	$I_3$
$A'$	0	①	2	③
A	4	⑤	⑥	7
	0	1	A	$A'$

(c) Gerçekleme tablosu

ŞEKİL 5-18

$F(A, B, C) = \Sigma(1, 3, 5, 6)$  fonksiyonunun bir veri seçiciyle gerçekleştirilmesi

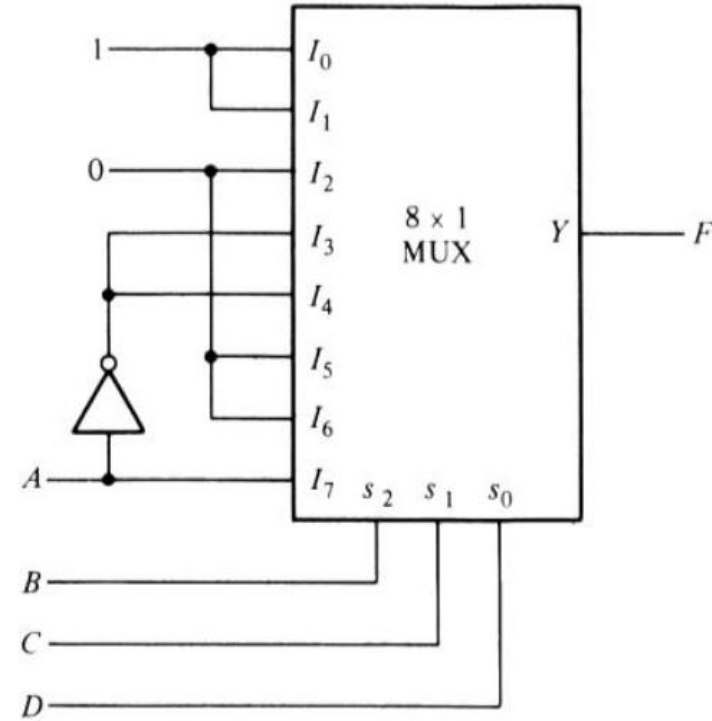


Örnek  
5-2

Aşağıdaki fonksiyonu bir veri seçici kullanarak gerçekleştirin:

$$F(A, B, C, D) = \Sigma(0, 1, 3, 4, 8, 9, 15)$$

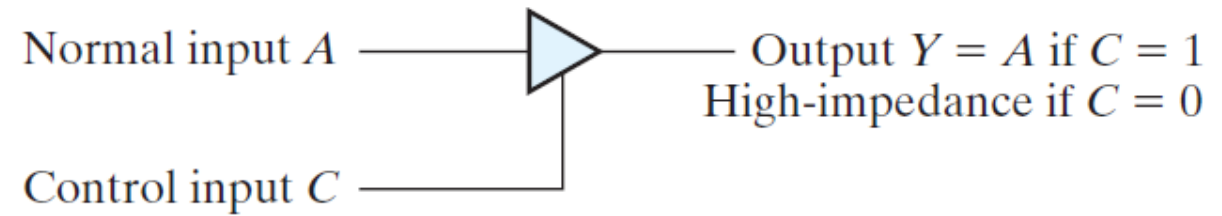
	$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$
$A'$	①	①	2	③	④	5	6	7
$A$	⑧	⑨	10	11	12	13	14	⑮
	1	1	0	$A'$	$A'$	0	0	$A$



ŞEKİL 5-20

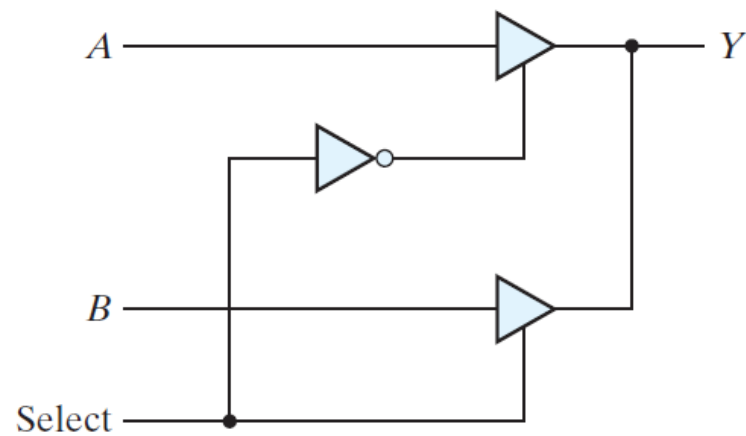
$F(A, B, C, D) = \Sigma(0, 1, 3, 4, 8, 9, 15)$  fonksiyonunun gerçekleştirilmesi

# Üç Durumlu Kapılar

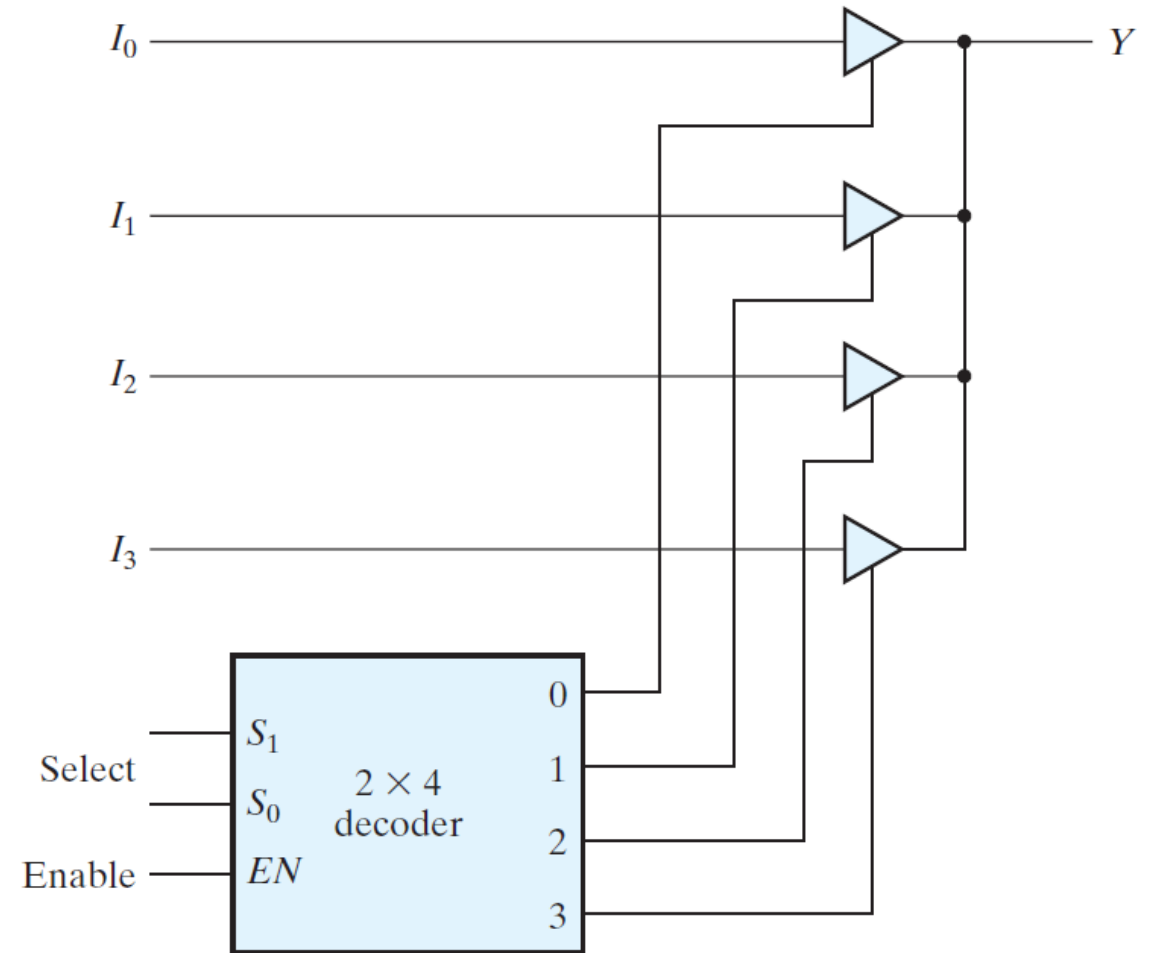


**FIGURE 4.29**

Graphic symbol for a three-state buffer



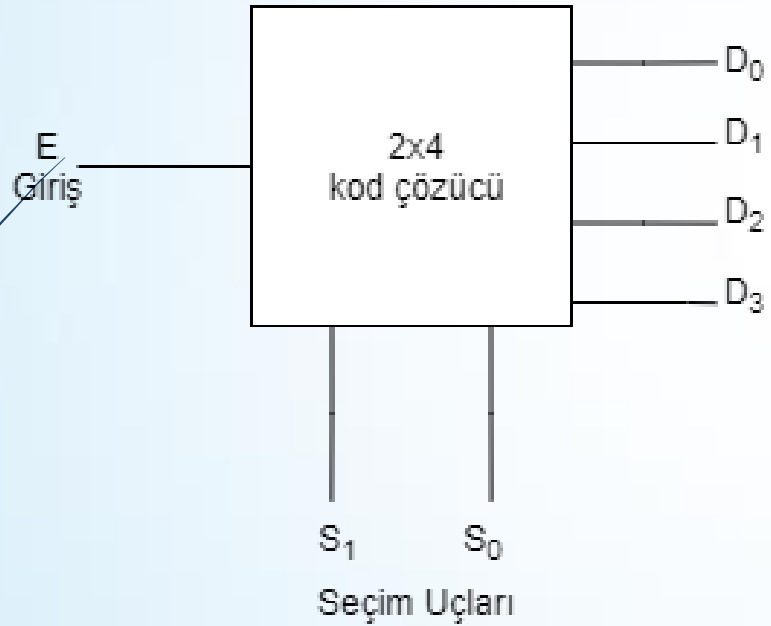
(a) 2-to-1-line mux



(b) 4-to-1-line mux

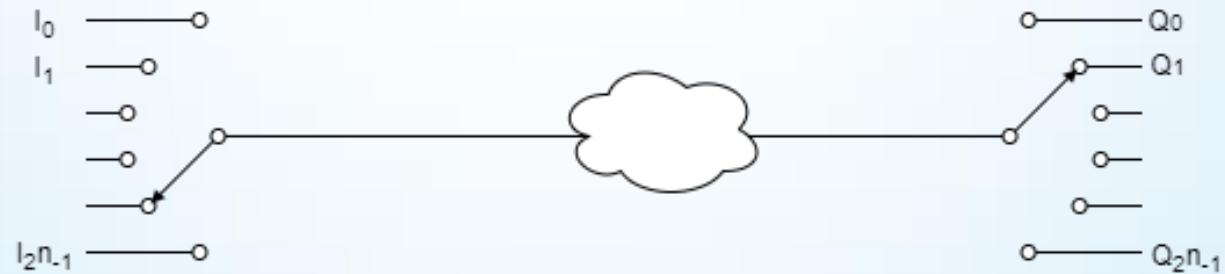
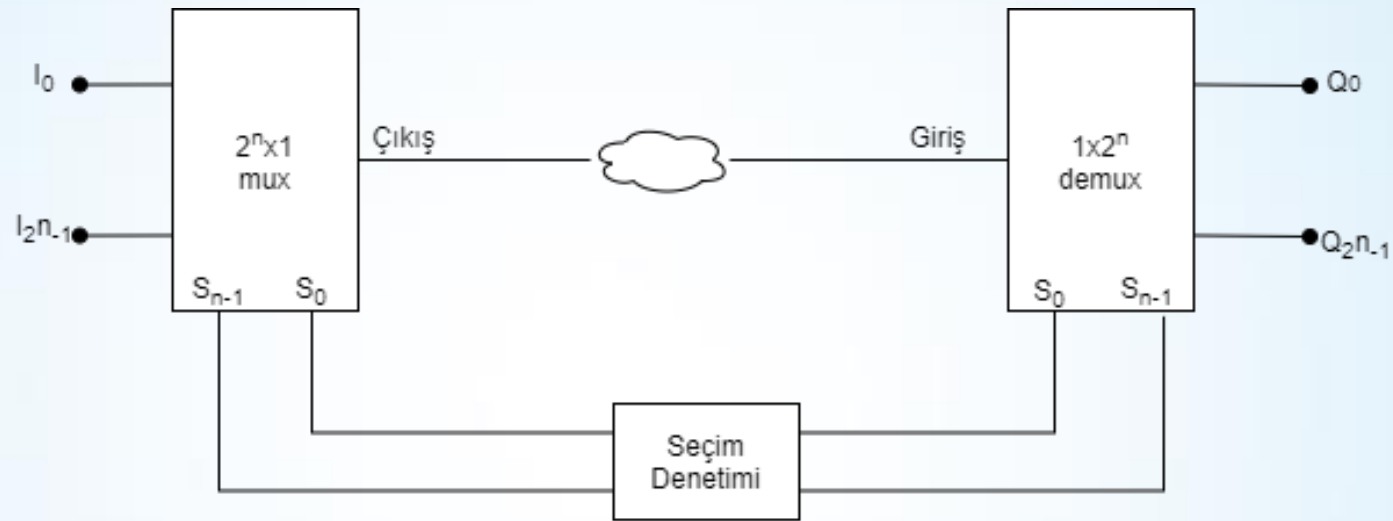
**FIGURE 4.30**  
Multiplexers with three-state gates

# DEMULTIPLEXER (Tekilleyici)

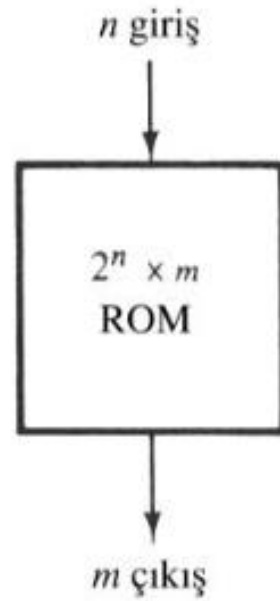


S <sub>1</sub>	S <sub>0</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	0	0	0	0	E
0	1	0	0	E	0
1	0	0	E	0	0
1	1	E	0	0	0

# Haberleşme Sistemlerinde MUX ve DEMUX kullanımı

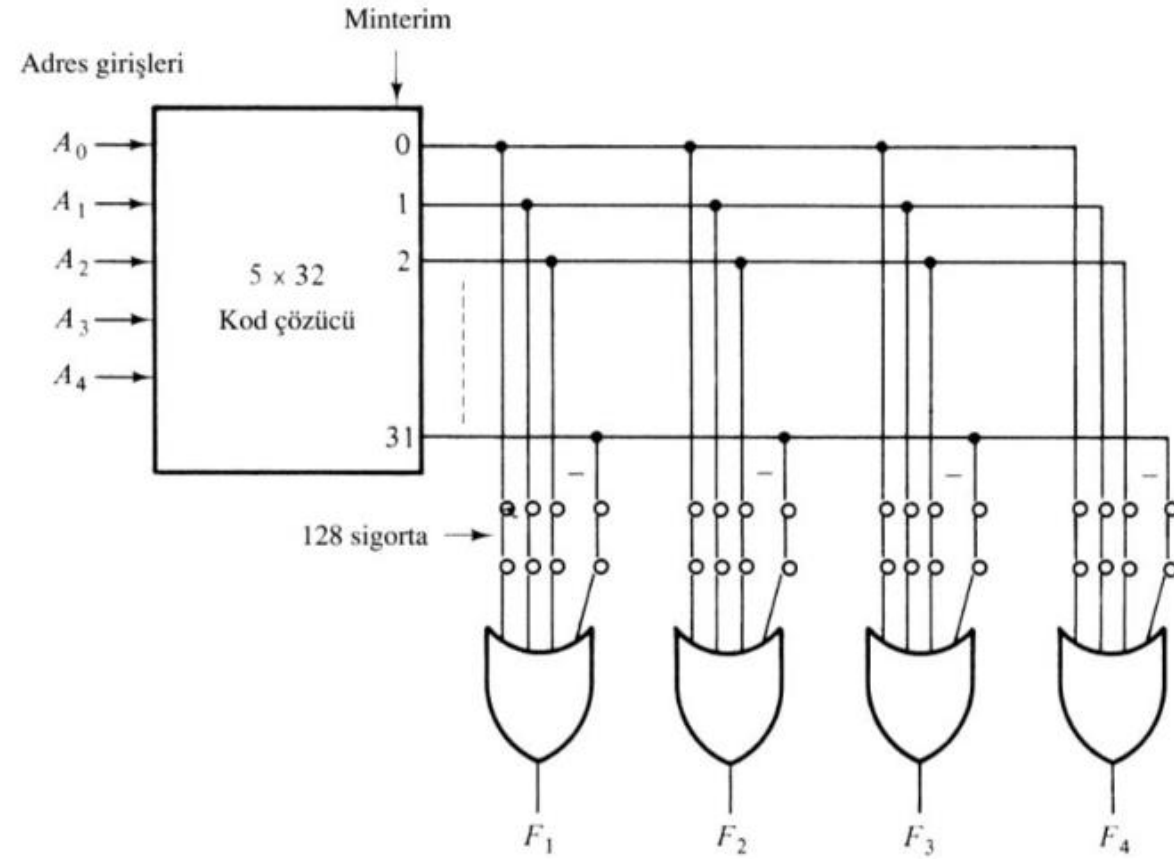


# Salt Oku Bellek (ROM)



**ŞEKİL 5-21**  
ROM'un blok diyagramı

# ROM



ŞEKİL 5-22

Bir  $32 \times 4$  ROM'un lojik yapısı

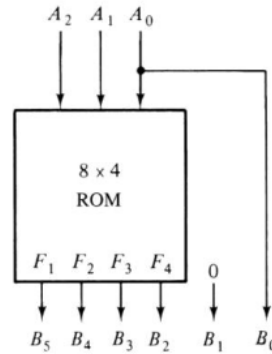
# Kombinezonal Lojik Uygulamaları

**Örnek 5-3** Girişine uygulanan 3 bitlik bir sayının karesinin ikili karşılığını çıkışında verebilecek kombinezonal devreyi bir ROM kullanarak gerçekleştirin.

İstenenleri gerçekleştirmek üzere ilk adım, kombinezonal devre için doğruluk tablosunun oluşturulmasıdır. Bu tür uygulamaların çoğunda bu iş-

**TABLO 5-5**  
Örnek 5-3'teki Devrenin Doğruluk Tablosu

Girişler			Çıkışlar							Onlu
$A_2$	$A_1$	$A_0$	$B_5$	$B_4$	$B_3$	$B_2$	$B_1$	$B_0$		
0	0	0	0	0	0	0	0	0	0	
0	0	1	0	0	0	0	0	1	1	
0	1	0	0	0	0	1	0	0	4	
0	1	1	0	0	1	0	0	1	9	
1	0	0	0	1	0	0	0	0	16	
1	0	1	0	1	1	0	0	1	25	
1	1	0	1	0	0	1	0	0	36	
1	1	1	1	1	0	0	0	1	49	



(a) Blok diyagram

$A_2$	$A_1$	$A_0$	$F_1$	$F_2$	$F_3$	$F_4$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

(b) ROM'un doğruluk tablosu

**ŞEKİL 5-24**

Örnek 5-3'teki ROM uygulaması





# Gelecek Ders

- Sıralı Lojik Devreleri
- 