

## Bölüm 6. Kombinasyonel Devreler (Devamı)

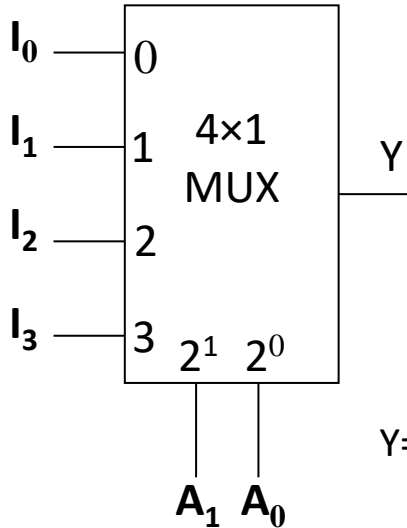
---

- ❖ **Çoğullayıcılar (Veri Seçiciler-Multiplexers-MUX)**
- ❖ **Veri Dağıtıcılar (Demultiplexers-DEMUX)**
- ❖ **Yedi Segment Display ve Kod Çözücü Devresi**

# Çoğullayıcılar (Veri Seçiciler-Multiplexers-MUX)

Çoğullayıcılar, n adet veri seçim ucuna,  $2^n$  adet veri giriş ucuna ve 1 adet de çıkışa sahip kombinasyonel devrelerdir. Veri seçici uçların değerine göre, veri girişlerinden birini çıkışa aktarırlar.

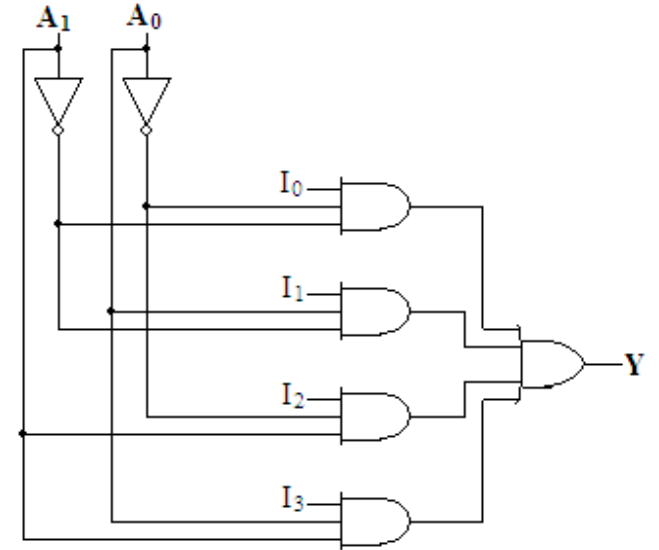
4×1'lik bir multiplexer'ın (MUX) gösterimi ve doğruluk tablosu aşağıdaki gibidir;



4×1 MUX'un gösterimi

$A_1$	$A_0$	Y
0	0	$I_0$
0	1	$I_1$
1	0	$I_2$
1	1	$I_3$

$$Y = I_0 \cdot A_1' \cdot A_0' + I_1 \cdot A_1' \cdot A_0 + I_2 \cdot A_1 \cdot A_0' + I_3 \cdot A_1 \cdot A_0$$



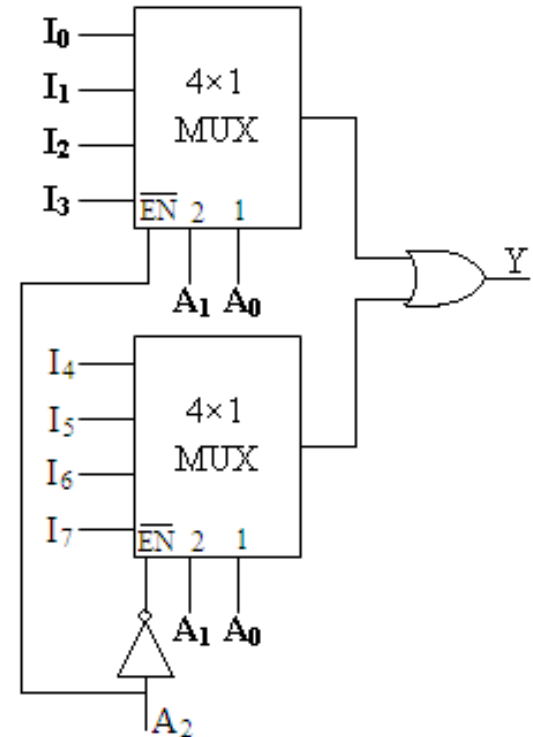
4×1 MUX'un lojik devresi

# Multiplexers

➤ Yetki (Enable) girişli multiplexer'lar kullanılarak, daha fazla veri girişine sahip çoğullayıcı devreler oluşturulabilir.

**Örnek:** Yetki girişine sahip iki adet  $4 \times 1$ 'lik multiplexer kullanarak  $8 \times 1$ 'lik bir multiplexer oluşturalım.

	$A_2$	$A_1$	$A_0$	Y
Üstteki MUX aktif	0	0	0	$I_0$
	0	0	1	$I_1$
	0	1	0	$I_2$
	0	1	1	$I_3$
Alttaki MUX aktif	1	0	0	$I_4$
	1	0	1	$I_5$
	1	1	0	$I_6$
	1	1	1	$I_7$

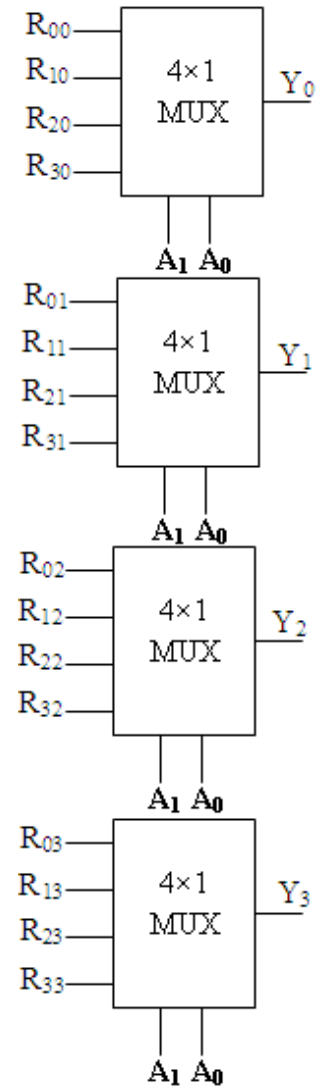


# Multiplexers

➤ Multiplexer'lar ile ortak veri yolu oluşturulabilir.

**Örnek:** 4 bitlik bir veri yoluna, 4 kaydediciyi (register) bağlamak istersek, yandaki devre kullanılabilir. Burada  $R_0$ ,  $R_1$ ,  $R_2$  ve  $R_3$  dörder bitlik kaydedicilerdir.

Şayet  $A_1A_0 = 00$  ise,  $R_0$  ( $R_{00}$ ,  $R_{01}$ ,  $R_{02}$ ,  $R_{03}$ ) kaydedicisinin içeriği yola aktarılacaktır.



# Multiplexers

➤ Multiplexer'lar lojik ifadeleri gerçekleştirmek için de kullanılabilirler. n adet girişe ve 1 çıkışa sahip bir kombinasyonel devre, (n-1) adet veri seçici uca ve  $2^{n-1}$  adet de veri giriş ucuna sahip  $2^{n-1} \times 1$  multiplexer kullanılarak gerçekleştirilebilir.  $\rightarrow 2^S$

**Örnek:**  $f(a,b,c) = abc + a'b' + bc'$  fonksiyonunu, veri seçim uçlarını a ve b seçerek  $4 \times 1$  MUX ile gerçekleştirelim.

Veri seçim uçları olarak a ve b değişkenleri seçildiğinden, tüm terimleri bu değişkenleri içerecek şekilde genişletmek gerekir;

$$f(a,b,c) = abc + a'b' + (a+a')bc' = \underline{abc} + a'b' + \underline{abc'} + a'bc' = ab + a'b' + a'bc'$$

Multiplexer'in tanım bağıntısından;

$$Y = I_0.a'.b' + I_1.a'.b + I_2.a.b' + I_3.a.b$$

O halde  $I_0 = 1$ ,  $I_1 = c'$ ,  $I_2 = 0$ ,  $I_3 = 1$  seçilmelidir.

## Örnek: (devamı)

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

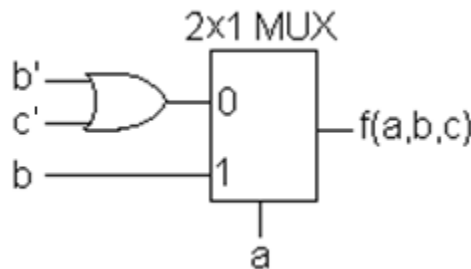
Aynı lojik ifadeyi veri seçim ucunu  $a$  seçerek  $2 \times 1$  MUX ile gerçekleştirmek istersek, ekstra kapılar kullanmamız gerekecektir.

$f$  fonksiyonunu  $a$  ve  $a'$  cinsinden yazarsak;

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

Multiplexer'in tanım bağıntısından;  $Y = I_0.a' + I_1.a$

O halde  $I_0 = b' + c'$  ve  $I_1 = b$  seçersek lojik ifadeyi gerçekleştirmiş oluruz.



**Dikkat** edilecek olursa ekstradan **VEYA** kapısını kullanmamız gerekmiştir.

# Multiplexers

➤ Standart çarpımlar biçiminde verilen bir lojik ifadeyi, uygun sayıda veri seçim ucuna sahip multiplexer kullanarak gerçekleştirmek için sistematik bir yaklaşım geliştirilmiştir.

**Örnek:** 3 girişe ve 1 çıkışa sahip bir kombinasyonel devrenin lojik ifadesi  $f(a,b,c) = \Sigma(1,3,4,7)$  olsun. Bu lojik ifadeyi, seçim uçları b ve c olarak seçilen,  $4 \times 1$  MUX ile gerçekleştirmek istediğimizi düşünelim.

$\begin{matrix} bc \\ a \end{matrix}$	00	01	10	11
0	$m_0$	$m_1$	$m_2$	$m_3$
1	$m_4$	$m_5$	$m_6$	$m_7$
Bağlantı	a	a'	0	1
Mux Veri Girişleri	$I_0$	$I_1$	$I_2$	$I_3$

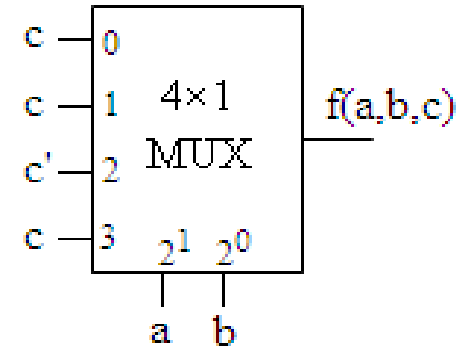
İşaretlenmiş olan minterler şayet a değişkeninin 1 olduğu satırdaysa bağlantı değeri a, 0 olduğu satırdaysa a', her ikisini de içeriyorsa 1 ve hiç birini içermiyorsa 0 seçilerek MUX'un  $I_0, I_1, I_2, I_3$  girişleri oluşturulmuştur.

## Örnek: (devamı)

$$f(a,b,c)=\Sigma(1,3,4,7)$$

Aynı lojik ifadeyi veri seçim uçlarını a ve b seçerek gerçekleştirmek isteseydik;

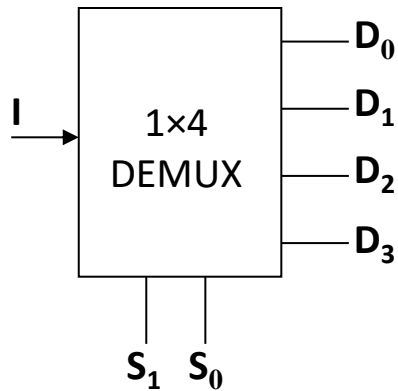
<b>ab</b> <b>c</b>	<b>00</b>	<b>01</b>	<b>10</b>	<b>11</b>
<b>0</b>	$m_0$	$m_2$	$m_4$	$m_6$
<b>1</b>	$m_1$	$m_3$	$m_5$	$m_7$
<b>Bağlantı</b>	<b>c</b>	<b>c</b>	<b>c'</b>	<b>c</b>
<b>Mux Veri Girişleri</b>	<b>I<sub>0</sub></b>	<b>I<sub>1</sub></b>	<b>I<sub>2</sub></b>	<b>I<sub>3</sub></b>





# Veri Dağıtıcılar (Demultiplexers-DEMUX)

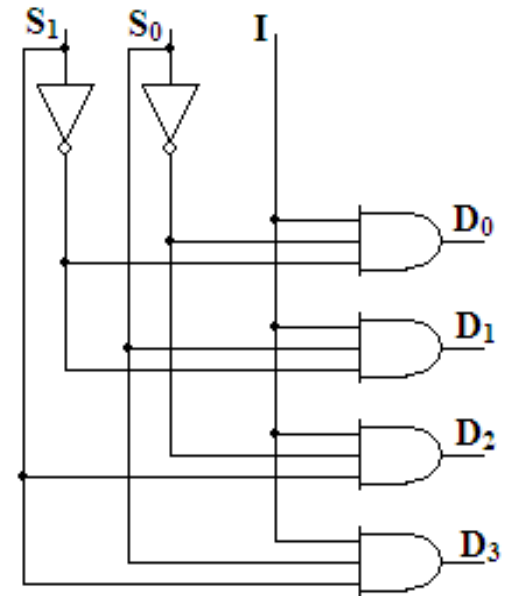
n adet seçici ucu, 1 tane girişi,  $2^n$  adet de çıkışı bulunan kombinyasyonel devrelerdir. Seçici uçların oluşturduğu sayı, girişin hangi çıkışa yönlendirileceğini gösterir. Aşağıda  $1 \times 4$  DEMUX'un gösterimi, doğruluk tablosu ve lojik devresi verilmiştir.



$1 \times 4$  DEMUX'un gösterimi

$S_1$	$S_0$	$D_0$	$D_1$	$D_2$	$D_3$
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I

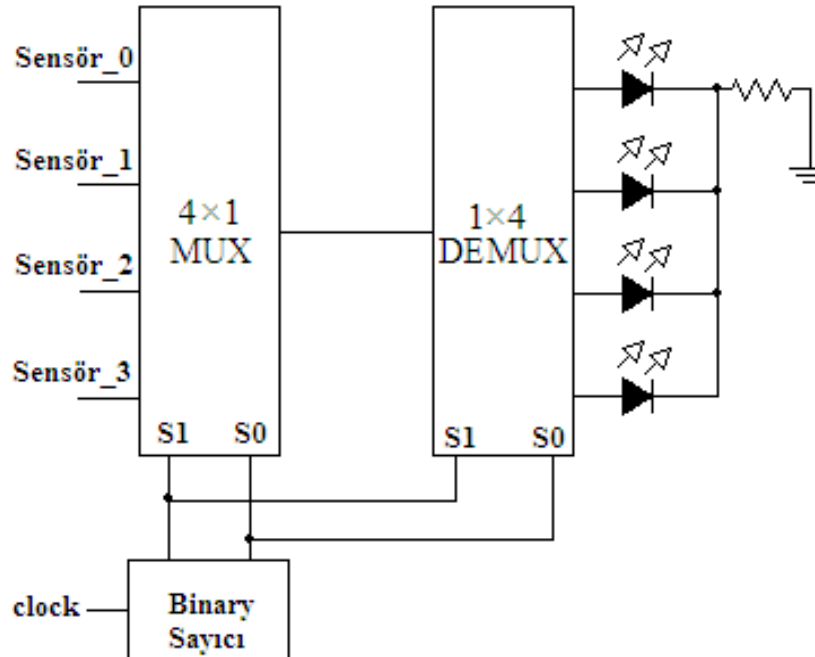
$1 \times 4$  DEMUX'un  
doğruluk tablosu



$1 \times 4$  DEMUX'un lojik devresi

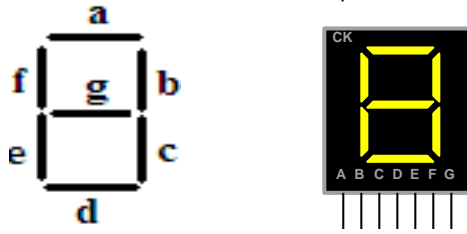
# Demultiplexers

**Örnek:** Özellikle verilerin uzak yerlere taşınması için MUX ve DEMUX'lar birlikte kullanılabilir. Örneğin 4 sensörden gelen bilgi, uzakta bulunan bir gösterge panelinden izlenebilir. Şayet sensörlerden 1 bilgisi gelirse, DEMUX çıkışında bulunan LED'ler yanıp söner. Kullanılan sayıcı, MUX girişlerindeki bilgilerin sırayla iletilmesini sağlar. Sayıcının frekansı aynı zamanda LED'lerin yanıp sönme aralığını belirler.

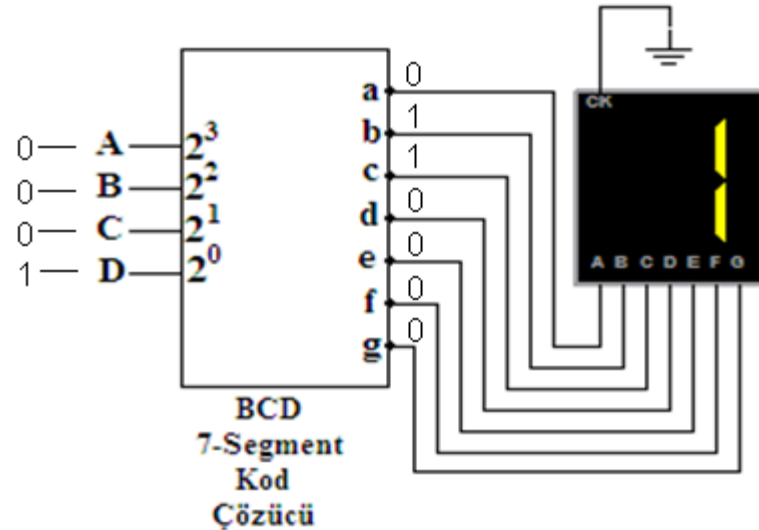


# Yedi Segment Display ve Kod Çözücü Devresi

Yedi segment display'ler lojik devrelerde özellikle decimal sayıları göstermek için yoğun olarak kullanılırlar. 7 adet LED bir araya getirilerek oluşturulur. Ortak anot veya ortak katot olarak üretilirler. Bu display'i sürmek için bir kod çözücüye ihtiyaç vardır. Bu kısımda BCD'den 7-segment display'e bir kod çözücü tasarlanacaktır. Bu kod çözücü entegre devre (IC) olarak temin edilebilir.



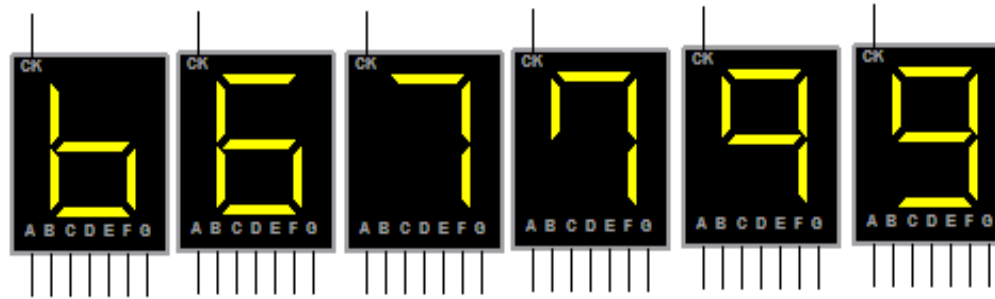
Yedi segment display



Yedi segment display için kod çözücü devre

# Yedi Segment Display İçin Kod Çözücü Devre Tasarımı

Kod çözücünün tasarımını yapmak için, doğruluk tablosunu oluşturabilir ya da Karnaugh haritasına çıkışları direkt olarak taşıyabiliriz. 6,7 ve 9'un iki türlü gösteriminin olabileceğini düşünerek, çıkışları en sade haline getirebilmek için, uygun olan gösterim, tasarım aşamasında seçilecektir.



**a**

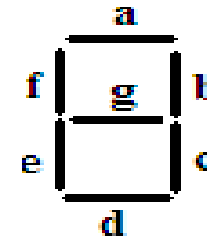
AB \ CD	00	01	11	10
00	1		x	1
01		1	x	1
11	1	1	x	x
10	1	x	x	x

$$a = A + C + BD + B'D'$$

**b**

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

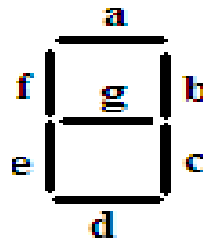
$$b = B' + CD + C'D'$$



# Yedi Segment Display İçin Kod Çözücü Devre Tasarımı

c					d				
AB \ CD	00	01	11	10	AB \ CD	00	01	11	10
00	1	1	x	1	00	1		x	1
01	1	1	x	1	01		1	x	x
11	1	1	x	x	11	1		x	x
10		1	x	x	10	1	1	x	x

$c = B + C' + D$ 
 $d = B'D' + CD' + B'C + BC'D$



0 1 2 3 4 5 6 7 8 9

# Yedi Segment Display İçin Kod Çözücü Devre Tasarımı

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

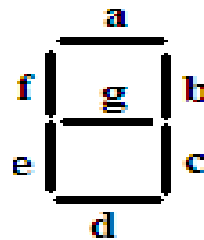
$$e = B'D' + CD'$$

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

$$f = A + B + C'D'$$

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

$$g = A + B'C + CD' + BC'$$



0 1 2 3 4 5 6 7 8 9

# Yedi Segment Display İçin Kod Çözücü Devre Tasarımı

