

BSM206 Mantıksal Devre Tasarımı

8. Hafta – Kodlayıcılar, Demultiplexer ve Multiplexer

Dr. Öğr. Üyesi Onur ÇAKIRGÖZ
onurcakirgoz@bartin.edu.tr

ANAHAT

- Kodlayıcılar (Encoders)
- Öncelik Kodlayıcısı (Priority Encoder)
- Demultiplexer
- Multiplexer
- Multiplexer ile Boole Fonksiyonlarının Uygulanması

Kodlayıcılar (Encoders)

- Bir kodlayıcı (encoder), kod çözücünün tersi işlemini gerçekleştirir.
- Bir kodlayıcı 2^n (veya daha az) sayıda giriş ve n adet çıkışa sahiptir.
- Çıkış hatları, giriş değerine karşılık gelen *ikili kodu* üretir.
- Sekizli (Oktal) sayı sisteminden ikili sayı sistemine kodlama yapan devre, kodlayıcı bir devre örneğidir ve doğruluk tablosu aşağıda verilmiştir:

Inputs								Outputs		
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

Kodlayıcılar (Encoders)

- Herhangi bir anda, bu sekiz girişten yalnızca bir tanesi 1 değerine sahiptir.
- Birden fazla giriş 1 olursa, devrenin **anlamsız** bir çıkışa sahip olduğu varsayılır.
- Bir kodlayıcı, girişleri doğruluk tablosundan elde edilen VEYA kapılarıyla gerçekleştirilebilir.
- Her bir çıkışa ilişkin Boole fonksiyonu şöyledir:

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

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

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

Inputs								Outputs		
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

Öncelik Kodlayıcısı (Priority Encoder)

- Kodlayıcının aynı anda birden fazla girişi aktif olursa, çıkışta **tanımsız** bir kombinasyon üretilir.
- Örneğin, D_3 ve D_6 aynı anda 1 ise, üç çıkışın hepsi 1'e eşit olur ve kodlayıcı 111 durumunda olur.
- 111, ne ikili 3'ü ne de ikili 6'yı temsil eder.
- Bu sorunu çözmek için, kodlayıcı devrelerde yalnızca bir girişin kodlandığından emin olunacak şekilde bir **öncelik** oluşturulur.
- Büyük indisli sayılara daha yüksek öncelik verilirse, D_3 ve D_6 'nın aynı anda 1 olması durumunda, D_6 , D_3 'ten daha yüksek öncelikli olduğundan çıkış 110 olacaktır.
- Diğer başka bir sorun da tüm girişler 0 olduğunda tüm çıkışların 0 olmasıdır. (Normalde, D_0 1 olduğunda böyle olmalıdır.) Girişlerin hiçbirinin aktif olmadığını belirleyen **ilave bir çıkış** kullanmak, bu sorunu çözebilir.

Öncelik Kodlayıcısı (Priority Encoder)

- Öncelik kodlayıcısı, öncelik fonksiyonunun (durumlarının) da yer aldığı bir kodlayıcı devredir.
- Bu devre, iki veya daha fazla giriş aynı anda 1 ise, en yüksek öncelikli girişi dikkate alır.
- Dört girişli öncelikli bir kodlayıcının doğruluk tablosu aşağıdadır:
- D_3 girişi en yüksek öncelikli, D_0 girişi ise en düşük önceliklidir.
- V çıkışı, geçerli çıkış göstergesidir ve yalnızca bir veya daha fazla giriş 1'e eşit olduğunda 1 olur.

Inputs				Outputs		
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

Öncelik Kodlayıcısı (Priority Encoder)

- x ve y çıkışlarının basitleştirilmesi amacıyla oluşturulan diyagramlar aşağıda verilmiştir.
- Doğruluk tablosunda 5 satır olmasına rağmen, etkisiz koşullar da dikkate alınarak tüm 16 olası giriş kombinasyonu üretilebilir.

D_0D_1		D_2D_3			
		00	01	11	10
D_0	00	m_0 X	m_1 1	m_3 1	m_2 1
	01	m_4	m_5 1	m_7 1	m_6 1
	11	m_{12}	m_{13} 1	m_{15} 1	m_{14} 1
	10	m_8	m_9 1	m_{11} 1	m_{10} X

$x = D_2 + D_3$

D_0D_1		D_2D_3			
		00	01	11	10
D_0	00	m_0 X	m_1 1	m_3 1	m_2
	01	m_4 1	m_5 1	m_7 1	m_6
	11	m_{12} 1	m_{13} 1	m_{15} 1	m_{14}
	10	m_8	m_9 1	m_{11} 1	m_{10}

$y = D_3 + D_1D'_2$

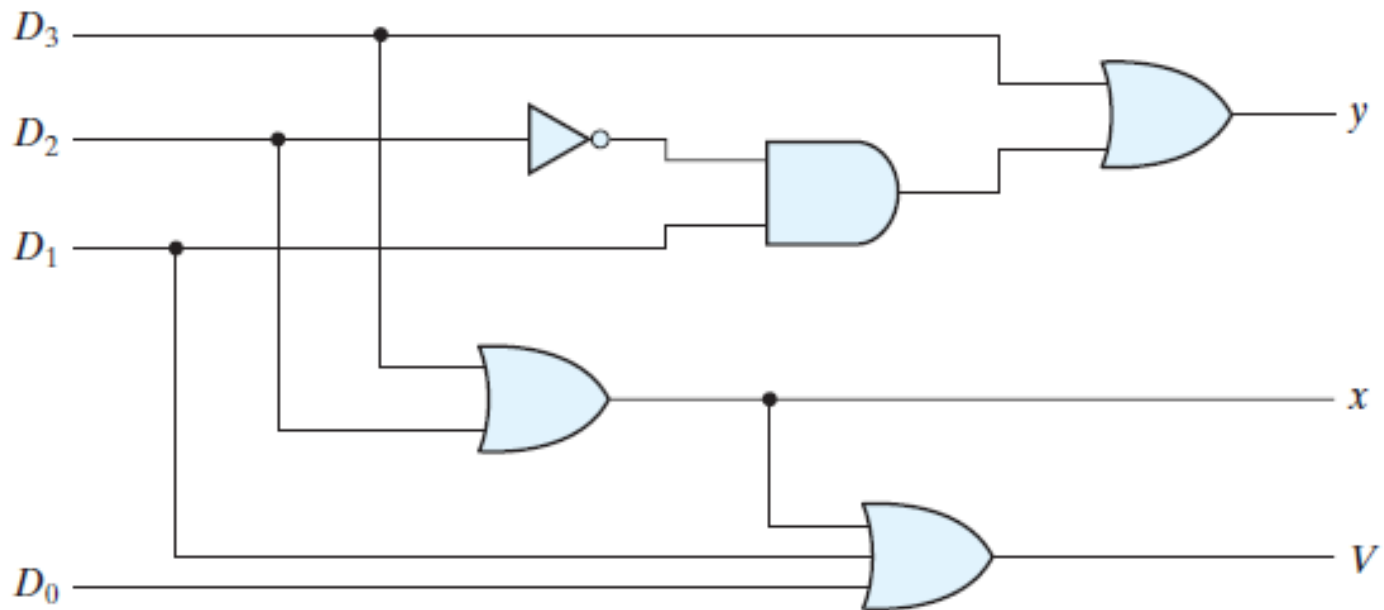
Öncelik Kodlayıcısı (Priority Encoder)

- Basitleştirilmiş Boole ifadeleri diyagramdan aşağıdaki gibi elde edilir ve öncelik kodlayıcısının lojik devresi çizilir.
- Geçerli çıkış göstergesi olan V çıkışı, tüm girişlerin bir VEYA fonksiyonudur.

$$x = D_2 + D_3$$

$$y = D_3 + D_1 D_2'$$

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

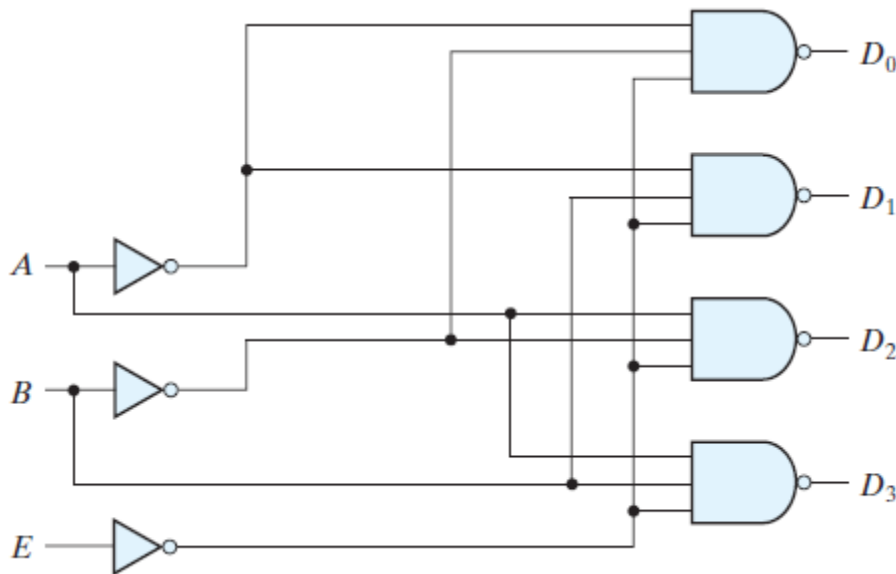


Demultiplexer

- Demultiplexer; ayırıcı, ayrıştırıcı, tekilleyici olarak da adlandırılır.
- Demultiplexer, giriş hattındaki bir bilgiyi, olası 2^n çıkış hattından birine aktaran kombinezonal bir devredir.
- İzin girişli bir kod çözücü bir demultiplexer fonksiyonu gösterebilir.
- Belirli bir çıkış hattının seçilmesi n adet seçme hattıyla kontrol edilir.

Demultiplexer

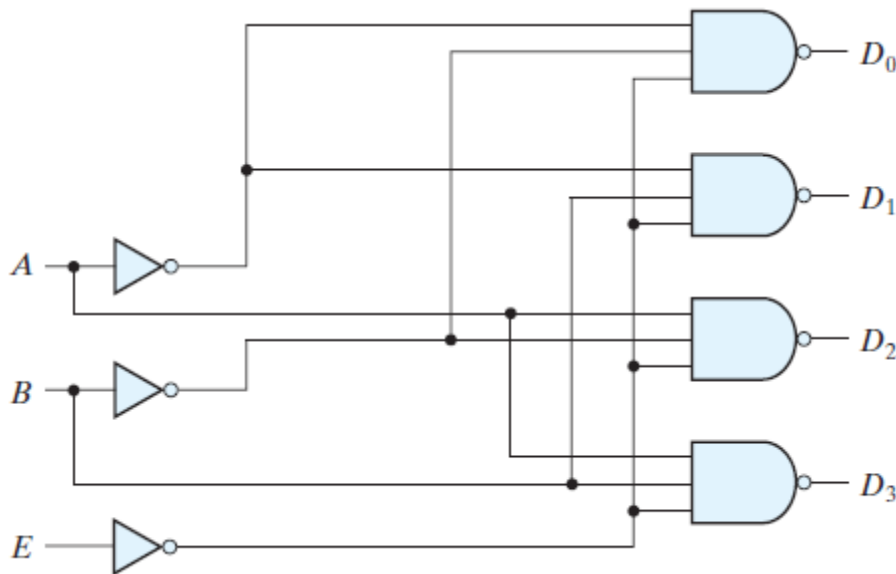
- Aşağıdaki kod çözücü, E bir veri giriş hattı, A ve B de seçme hatları olarak gözönüne alındığında bir Demultiplexer fonksiyonunu gerçekleştirebilir.
- E giriş değişkenindeki bilgi, çıkış hatlarından sadece birine yönlendirilir. (İki seçme hattı A ve B'nin ikili kombinasyonu tarafından belirlenen çıkış hattına)



<i>E</i>	<i>A</i>	<i>B</i>	<i>D</i> ₀	<i>D</i> ₁	<i>D</i> ₂	<i>D</i> ₃
1	<i>X</i>	<i>X</i>	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

Demultiplexer

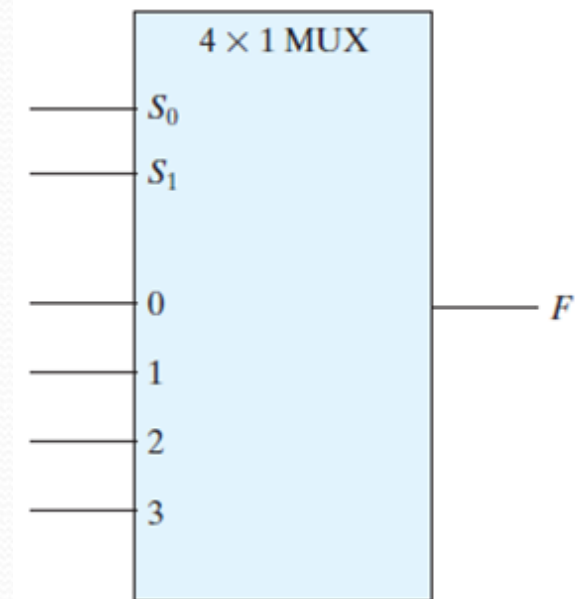
- Aşağıdaki kod çözücü yada (Demultiplexer), **tümlenmiş** çıkışlıdır yani VEDEĞİL kapıları ile oluşturulmuştur.
- Benzer şekilde, VE kapıları ile oluşturulmuş izin girişli bir kod çözücü de Demultiplexer olarak kullanılabilir ve bu formdaki Demultiplexer **normal** çıkışlıdır. (Seçilen çıkış 1 olur, diğerleri ise 0)



<i>E</i>	<i>A</i>	<i>B</i>	<i>D</i> ₀	<i>D</i> ₁	<i>D</i> ₂	<i>D</i> ₃
1	<i>X</i>	<i>X</i>	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

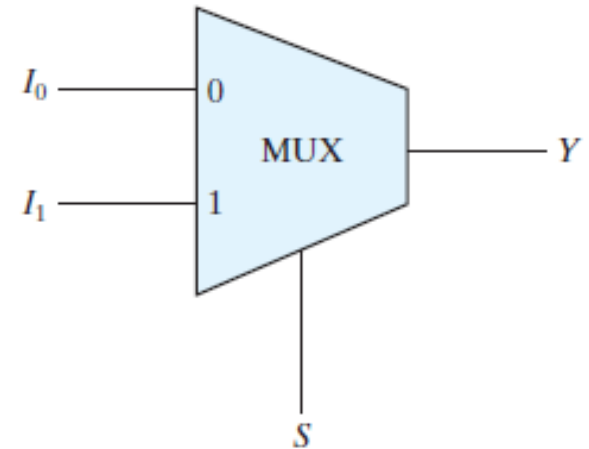
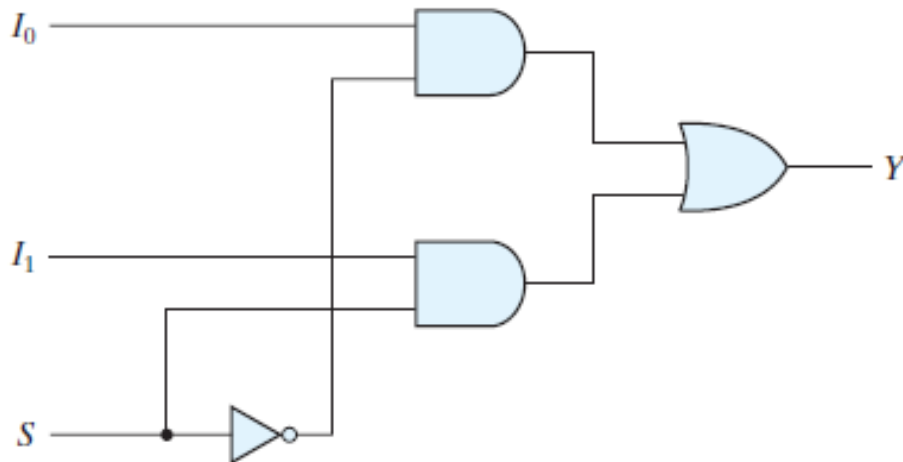
Multiplexer

- Multiplexer; "giriş yolu seçici" yada "veri seçici" olarak da adlandırılırlar.
- Bir multiplexer, bir veya daha fazla giriş hattından ikili bilgiyi **seçen** ve bunu tek çıkış hattına bağlayan bir kombinezonal devredir.
- Herhangi bir giriş hattının seçilmesi bir grup seçme hattıyla kontrol edilir.
- Genel olarak, **2^n giriş hattı** ve hangi girişin seçileceğini belirleyen **n seçme hattı** vardır.
- Yanda bir multiplexer'ın blok diyagramı görülmektedir.



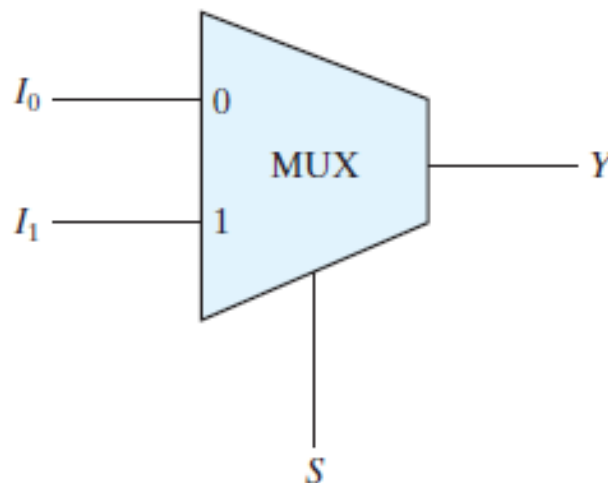
Multiplexer

- İki 1-bitlik kaynaktan birini ortak bir çıkışa bağlayan 2'den 1'e Multiplexer aşağıda gözükmemektedir.
- Devrede, iki tane veri girişi, bir çıkış ve bir tane seçim hattı (S) vardır.
- $S = 0$ olduğunda yukarıdaki VE kapısına geçit verilir (etkinleştirilir) ve I_0 girişi çıkışa aktarılır.
- $S = 1$ olduğunda aşağıdaki VE kapısına geçit verilir (etkinleştirilir) ve I_1 girişi çıkışa aktarılır.



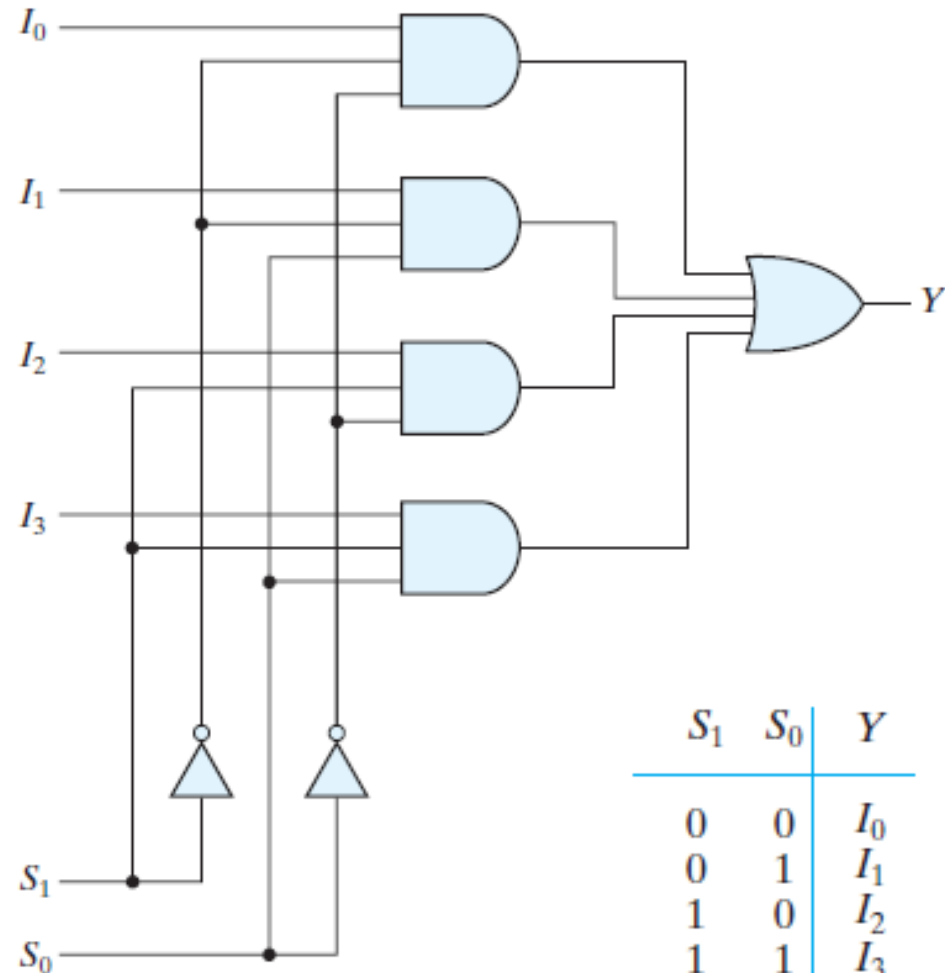
Multiplexer

- Multiplexer'lerin blok diyagramı için bazen aşağıdaki gösterim kullanılır.
- Bu blok diyagram, Multiplexer'in çok sayıdaki girişten seçilmiş olanı tek çıkışa yönlendirdiğini temsil eden anlamlı bir şekildir.
- Multiplexer'ler, aşağıda da olduğu gibi, sıklıkla blok diyagramda "MUX" ifadesiyle etiketlenir.



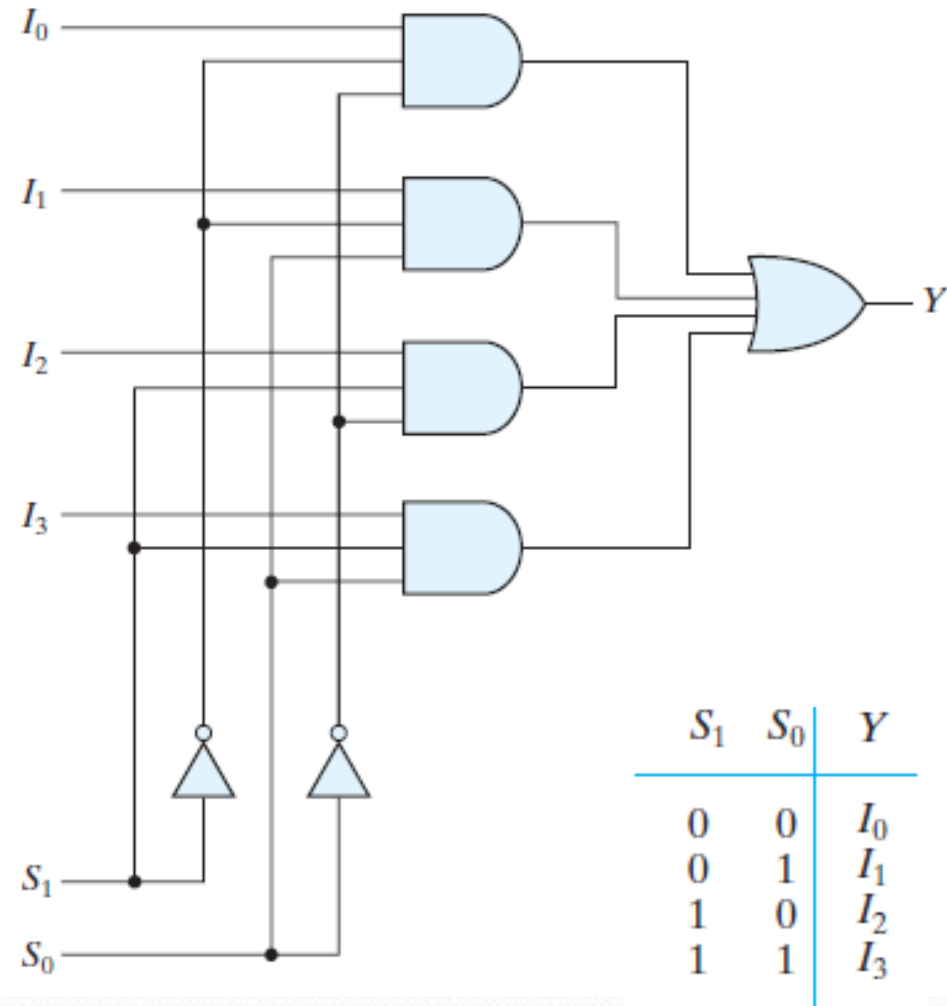
Multiplexer

- Bir 4'ten 1'e Multiplexer yanda gösterilmiştir.
- I_0 'dan I_3 'e kadar olan dört giriş hattından her biri ayrı bir VE kapısına uygulanır.
- S_1 ve S_0 seçme girişleri belirli bir VE kapısının seçilmesi için kod çözme işlemini gerçekleştirir. (Kod çözücü ?)
- Köşedeki fonksiyon tablosu, seçme hatlarının olası tüm kombinasyonları için girişten çıkışa olan bağlantıyı belirler.



Multiplexer

- Yandaki Multiplexer devresinin çalışmasını göstermek için $S_1S_0 = 01$ durumunu düşünelim.
- I_1 girişinin uygulandığı VE kapısının S_1 ve S_0 dan gelen diğer iki girişi 1 olduğundan, kapının çıkışı I_1 'e eşit olur.
- Diğer VE kapılarının en azından bir girişi 0 olduğundan, VEYA kapısının çıkışı I_1 'in değerine eşit olacaktır.

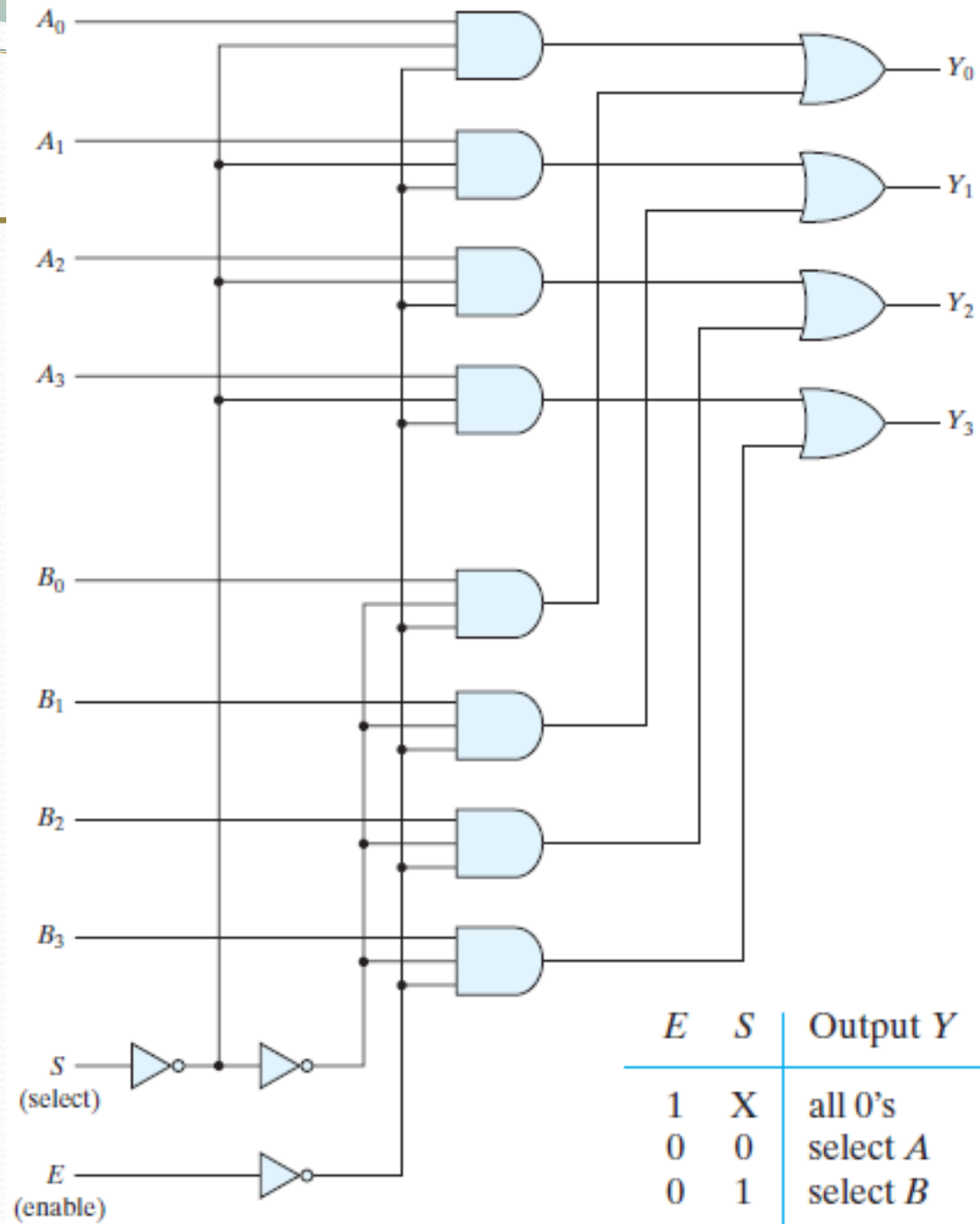


Multiplexer

- Kod çözücünde olduğu gibi Multiplexer tümdevresinde de elemanın çalışmasını kontrol etmek için bir **izin girişi** olabilir.
- **İzin girişi** belirli bir durumdayken çıkışlar **yasaklanır**.
- **İzin girişi** daha büyük boyutlu bir Multiplexer için iki veya daha fazla sayıda Multiplexer tümdevresinin birbirine *bağlanmasında* kullanılır.
- Bazı durumlarda iki veya daha fazla Multiplexer, bir tümdevre kılıfı içinde yer alabilir.
- Bu tür bir tümdevrede yer alan tüm Multiplexer'ler için seçme ve izin girişleri ortaktır.

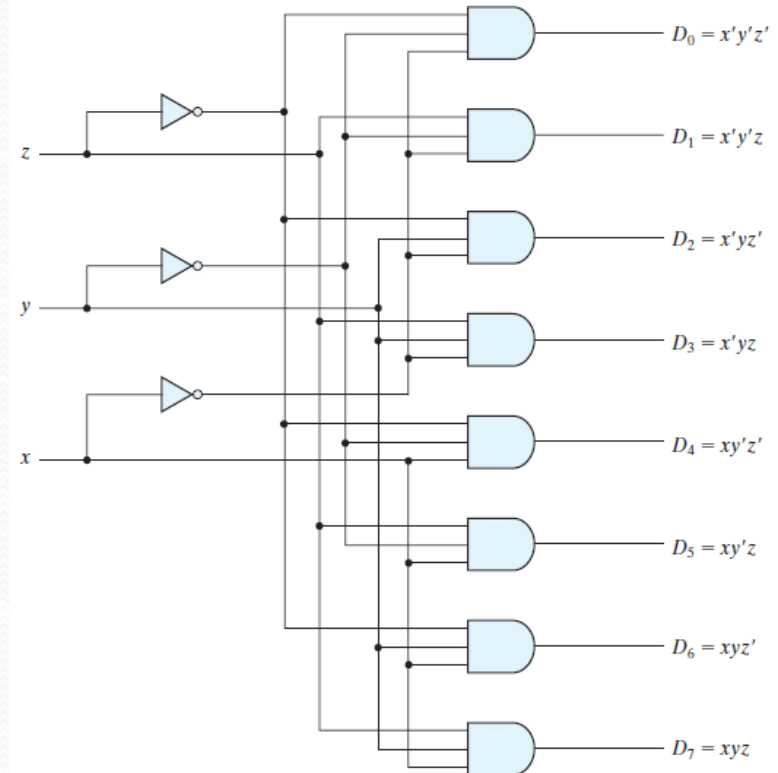
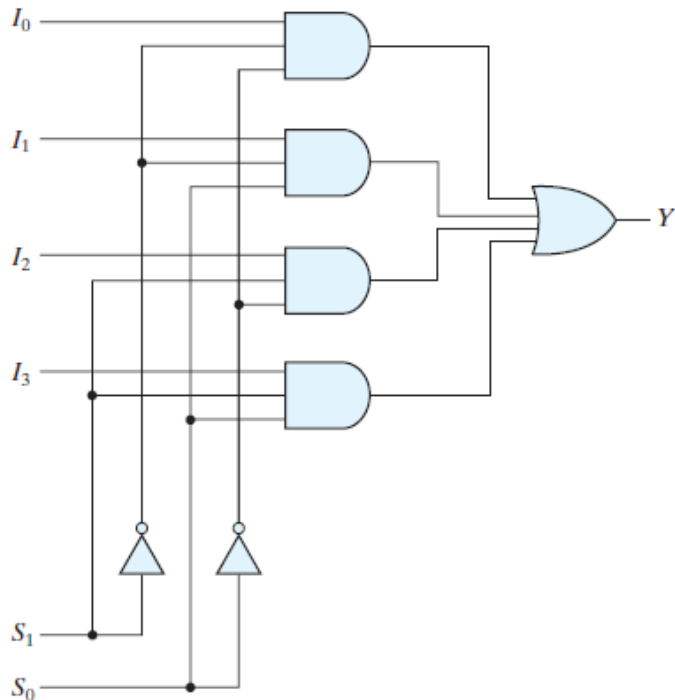
Multiplexer

- Dörtlü 2'den 1'e bir multiplexer devresi yanda gözükmemektedir.
- Devrede dört multiplexer ve dört çıkış vardır.
- Dört multiplexer'ın her biri için; iki giriş hattından (A_i ve B_i) sadece biri seçilebilir.
- Örn, Y_1 çıkışı ya A_1 yada B_1 'e eşit olabilir.
- Bir giriş hattı bulunan seçme (S), tüm dört veri seçicide iki hattan birinin seçilmesi için yeterlidir.
- İzin girişi E , 0 durumunda iken devre aktif olur, 1 durumunda iken devre yasaklanır.



Multiplexer'ler ile Boole Fonksiyonlarının Uygulanması

- Bilindiği gibi, bir kod çözücü (decoder), bir Boole fonksiyonunu dışarıdan bağlanan bir VEYA kapısı kullanarak gerçekleştirebilir.
- Multiplexer devresi ise, VEYA kapılı bir kod çözücüye benzemektedir. Solda 4'ten 1'e Multiplexer, sağda ise 3'ten 8'e bir kod çözücü yer almaktadır.



Multiplexer'ler ile Boole Fonksiyonlarının Uygulanması

- n değişkenli herhangi bir Boole fonksiyonunu, 2^{n-1} 'den 1'e bir Multiplexer kullanarak gerçeklemek için yöntem şöyledir:
 - Boole fonksiyonunun doğruluk tablosu oluşturulur.
 - Tablodaki ilk $n - 1$ değişken alınır ve Multiplexer'in seçme hatlarına (S) bağlanır.
 - Seçim değişkenlerinin her bir kombinasyonu, kalan tek değişken açısından tablo üzerinde değerlendirilir.
 - Bu tek kalan değişken C olsun, o durumda Multiplexer'in girişleri C , C' , 1 veya 0 olabilir.
 - Bu dört değer Multiplexer'in girişlerine uygun şekilde bağlanır.

Multiplexer'ler ile Boole Fonksiyonlarının Uygulanması - Örnek

- Aşağıda verilen Boole fonksiyonunu uygun bir Multiplexer ile gerçekleyin.

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

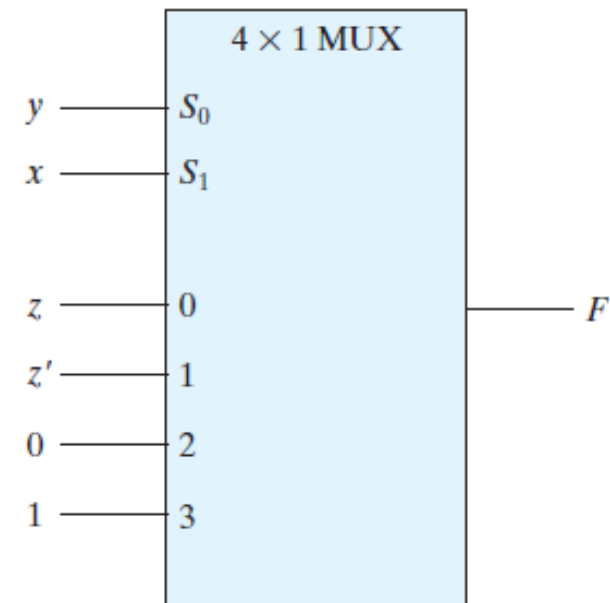
- Çözüm:** Üç değişkenli bu fonksiyon 4'ten 1'e Multiplexer ile gerçekleştirilebilir.
- x ve y değişkenleri seçim hatlarına şu sırada uygulanır: x S_1 girişine, y ise S_0 girişine bağlanır.
- Veri girişleri için uygulanması gereken değerler fonksiyonun doğruluk tablosundan bulunur.
- xy = 00 iken F çıkışı z'ye eşit olur. z, 0'ıncı veri girişine uygulanmalıdır.
- xy = 01 iken F çıkışı z'nin tümleyenine eşit olur. z', 1'inci veri girişine uygulanmalıdır.
- Benzer mantıkla, diğer veri girişlerinin girdileri de bulunur.

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

Multiplexer'ler ile Boole Fonksiyonlarının Uygulanması – Örnek (Devam)

- Bu soruda Multiplexer'in çalışmasını bir örnekle açıklayacak olursak; $xy = 00$ iken 0'ıncı veri girişi seçilmiş olur ve 0'ıncı veri girişi çıkışa aktarılır. Yani, F çıkışı z 'ye eşit olur. $z = 0$ ise çıkış 0, $z = 1$ ise çıkış 1 olur.
- Benzer şekilde, diğer seçim kombinasyonları için uygun veri girişleri çıkışa aktarılır.

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



Multiplexer'ler ile Boole Fonksiyonlarının Uygulanması - Örnek

- Aşağıda verilen Boole fonksiyonunu uygun bir Multiplexer ile gerçekleyin.

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

- Çözüm:** Dört değişkenli bu fonksiyon 8'den 1'e Multiplexer ile gerçekleştirilebilir.
- A, B ve C değişkenleri seçim hatlarına şu sırada uygulanır: A S_2 girişine, B S_1 girişine, C ise S_0 girişine bağlanır.
- Multiplexer'in veri girişlerine uygulanacak değerler fonksiyonun doğruluk tablosundan bulunur.
- Örneğin, ABC = 000 iken F çıkışı D'ye eşit olur. D, 0'ıncı veri girişine uygulanmalıdır.
- Örneğin, ABC = 011 iken F çıkışı 0'a eşit olmalıdır. 0 değeri, 3'üncü veri girişine uygulanmalıdır.

A	B	C	D	F	
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	

Multiplexer'ler ile Boole Fonksiyonlarının Uygulanması – Örnek (Devam)

- Bu soruda Multiplexer'in çalışmasını bir örnekle açıklayacak olursak; $ABC = 110$ iken 6'ncı veri girişi seçilmiş olur ve 6'ncı veri girişi çıkışa aktarılır. Yani, F çıkışı, D 'nin değerinden bağımsız olarak 1'e eşit olur.

<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	

