Sayısal Sistemler-H7CD2

Vize Öncesi Genel Tekrar-2

Dr. Meriç Çetin

versiyon251020

Harita Yöntemi

- Bir Boole işlevini uygulayan sayısal mantık kapılarının karmaşıklığı, doğrudan işlevin uygulandığı cebirsel ifadenin karmaşıklığı ile ilgilidir.
- Bir fonksiyonun doğruluk tablosu ile temsili benzersiz olsa da, cebirsel olarak ifade edildiğinde birçok farklı fakat eşdeğer formda görünebilir.
- Boole ifadeleri, 2. Bölümde tartışıldığı gibi cebirsel yollarla basitleştirilebilir.
- Bununla birlikte, bir sonraki adımı tahmin etmek için belirli kurallara sahip değildir.
- Burada sunulan harita yöntemi, Boole işlevlerini en aza indirgemek için basit, anlaşılır bir prosedür sağlar.
- Bu yöntem, bir doğruluk tablosunun resimli bir formu olarak kabul edilebilir.
- Harita yöntemi, **Karnaugh haritası** veya **K-haritası** olarak da bilinir.

Karnaugh Haritası

- K-haritası, karelerden oluşan bir diyagramdır ve her kare, küçültülecek fonksiyonun bir **mintermini** temsil eder.
- Herhangi bir Boole fonksiyonu mintermlerin toplamı olarak ifade edilebildiğinden, mintermleri fonksiyona dahil edilen karelerin çevrelediği alanda bir sadeleştirme yapılır.
- Harita, bir fonksiyonun standart formda ifade edilebileceği tüm olası yolların görsel bir diyagramını sunar.
- Harita tarafından üretilen basitleştirilmiş ifadeler her zaman iki standart formdan biridir: çarpımların toplamı veya toplamların çarpımı.
- **En basit cebirsel ifadenin**, minimum sayıda terim ve her terimde mümkün olan en az sayıda bilgi içeren bir cebirsel ifade olduğu varsayılacaktır.
  - Bu ifade, minimum sayıda kapı ve her kapıya minimum sayıda giriş içeren bir devre şeması oluşturur.
- Daha sonra, en basit ifadenin benzersiz olmadığını göreceğiz: Bazen minimizasyon kriterlerini karşılayan iki veya daha fazla ifade bulmak mümkündür. Bu durumda her iki çözüm de olasıdır.

2 Değişkenli K-haritası

- İki değişkenli harita Şekil 3.1 (a) 'da gösterilmektedir.
- İki değişken için dört minterm vardır; dolayısıyla harita, her minterm için bir tane olmak üzere dört kareden oluşur.
- Kareler ile x ve y değişkenleri arasındaki ilişkiyi göstermek için harita (b) 'de yeniden çizilmiştir.
- Her satırda ve sütunda işaretlenen 0 ve 1, değişkenlerin değerlerini belirtir.

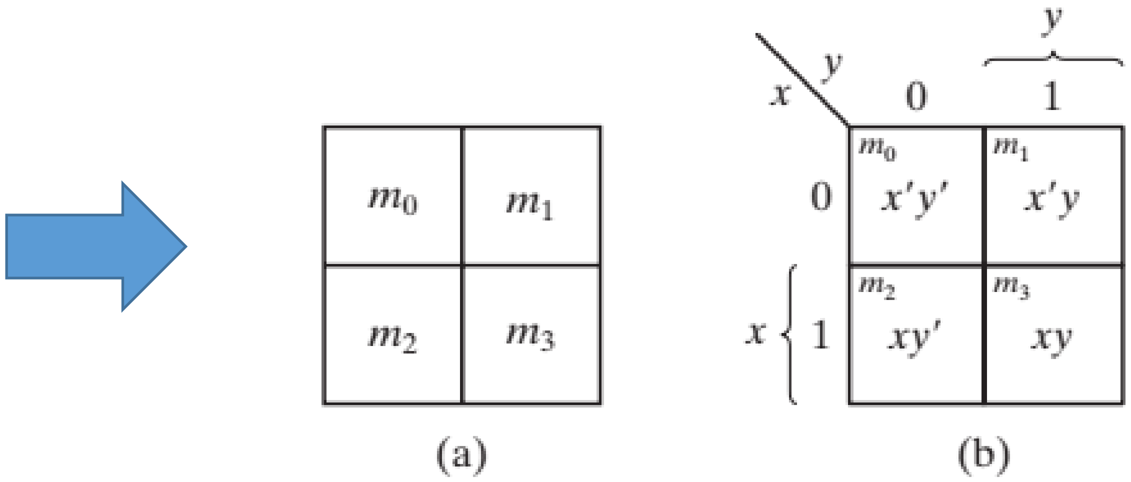
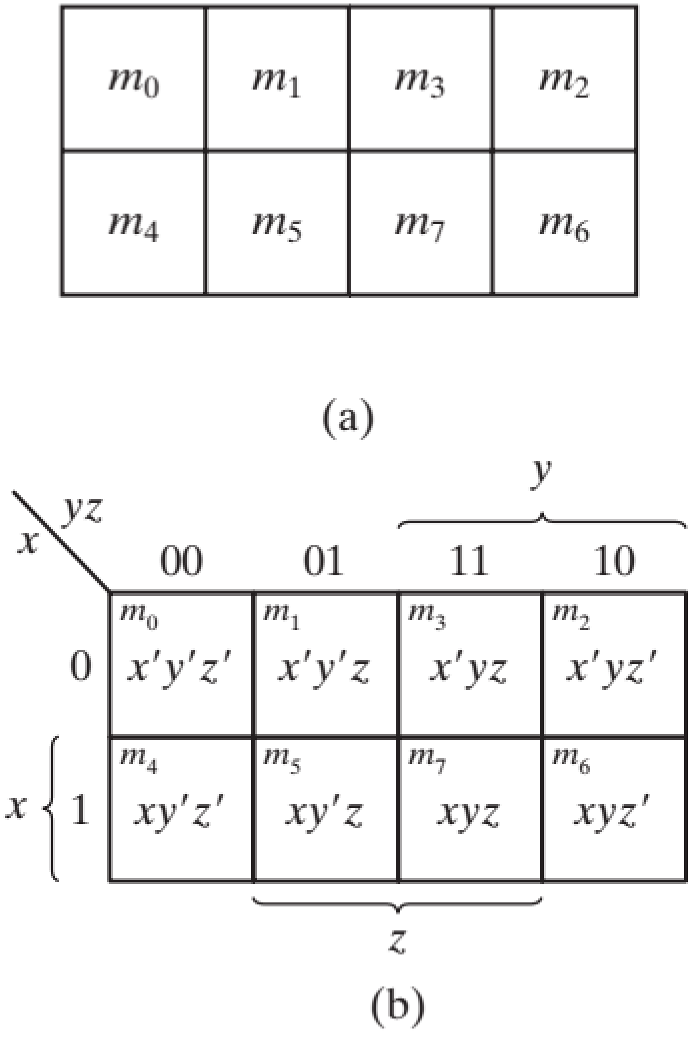


FIGURE 3.1
Two-variable K-map

3 Değişkenli K-haritası

- Üç değişkenli bir K-haritası Şekil 3.3'te gösterilmektedir.
- Üç ikili değişken için sekiz minterm vardır;
- Bu nedenle harita sekiz kareden oluşur.
- Mintermlerin ikili bir sırayla değil, Gray koduna benzer bir sırayla düzenlendiğini unutmayın.
- Boole fonksiyonlarını basitleştirmede haritanın kullanışlılığını anlamak için, bitişik karelerin sahip olduğu temel özelliği tanımalıyız



m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

(a)

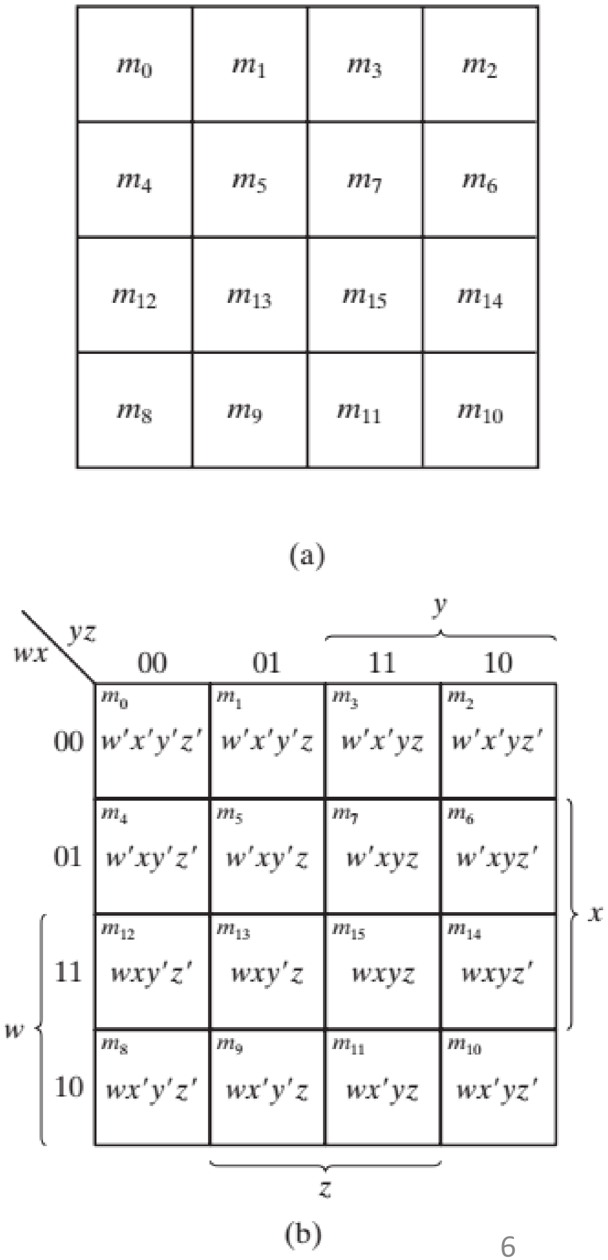
$x \backslash yz$	00	01	11	10
0	m_0 $x'y'z'$	m_1 $x'y'z$	m_3 $x'yz$	m_2 $x'yz'$
1	m_4 $xy'z'$	m_5 $xy'z$	m_7 xyz	m_6 xyz'

(b)

FIGURE 3.3
Three-variable K-map

4 Değişkenli K-haritası

- Dört ikili değişkenin (w, x, y, z) Boole fonksiyonlarının haritası Şekil 3.8'de gösterilmektedir.
- Şekil 3.8 (a) 'da 16 minterm ve her birine atanmış kareler listelenmiştir.
- Şekil 3.8 (b) 'de, kareler ve dört değişken arasındaki ilişkiyi göstermek için harita yeniden çizilmiştir.
- Sıralar ve sütunlar, iki bitişik satır veya sütun arasında yalnızca bir rakamın değiştiği bir Gri kod dizisi ile numaralandırılır.
- Her kareye karşılık gelen minterm, satır numarasının sütun numarası ile birleştirilmesinden elde edilebilir.



m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

(a)

$wx \backslash yz$	00	01	11	10
00	m_0 $w'x'y'z'$	m_1 $w'x'y'z$	m_3 $w'x'yz$	m_2 $w'x'yz'$
01	m_4 $w'xy'z'$	m_5 $w'xy'z$	m_7 $w'xyz$	m_6 $w'xyz'$
11	m_{12} $wxy'z'$	m_{13} $wxy'z$	m_{15} $wxyz$	m_{14} $wxyz'$
10	m_8 $wx'y'z'$	m_9 $wx'y'z$	m_{11} $wx'yz$	m_{10} $wx'yz'$

(b)

FIGURE 3.8
Four-variable map

2 Değişkenli K-haritası vs Boole cebri

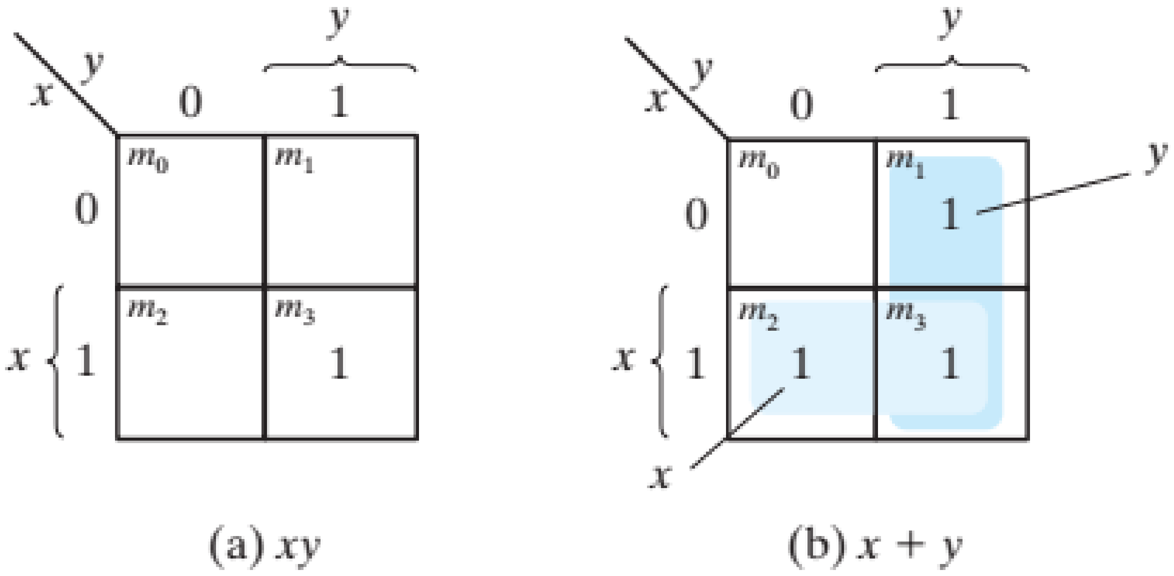
- Aşağıdaki fonksiyonu Boole cebri kurallarına göre indirgeyiniz

$$m_1 + m_2 + m_3 = x'y + xy' + xy$$

$$y\underbrace{(x+x')}_{1}+xy'=y+xy'=(y+x)\underbrace{(y+y')}_{1}=x+y$$

2 Değişkenli K-haritası

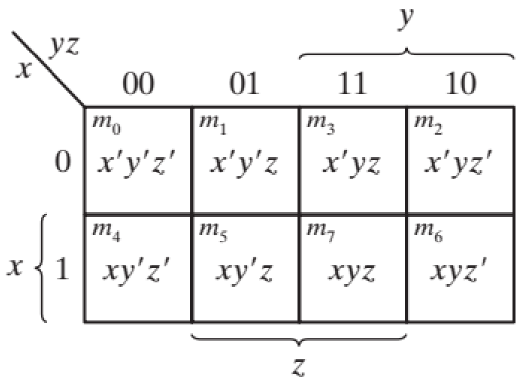
$$m_1 + m_2 + m_3 = x'y + xy' + xy = x + y$$



(a) xy (b) $x + y$

FIGURE 3.2
Representation of functions in the map

Örnek2

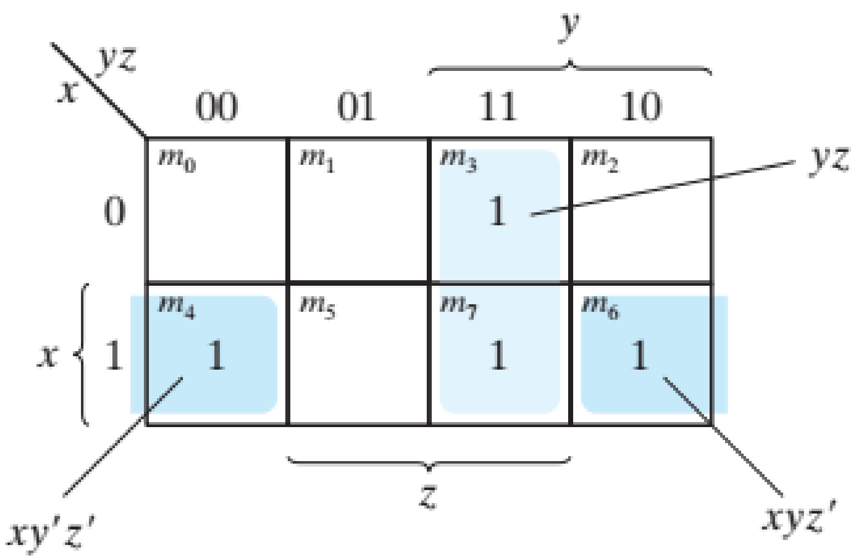


(b)

EXAMPLE 3.2

Simplify the Boolean function

$$F(x, y, z) = \Sigma(3, 4, 6, 7)$$



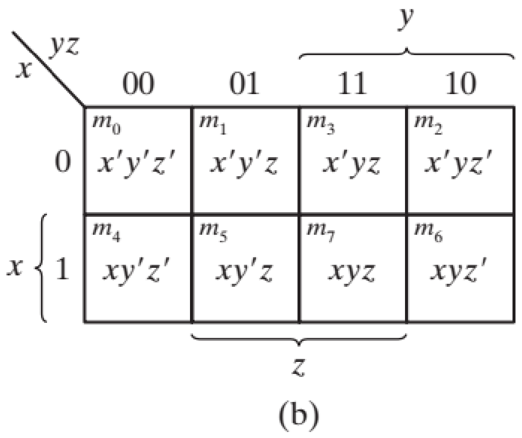
Note: $xy'z' + xyz' = xz'$

FIGURE 3.5

Map for Example 3.2, $F(x, y, z) = \Sigma(3, 4, 6, 7) = yz + xz'$

"**Digital Design**",
M. Morris Mano

Örnek3



EXAMPLE 3.3

Simplify the Boolean function

$$F(x, y, z) = \Sigma(0, 2, 4, 5, 6)$$

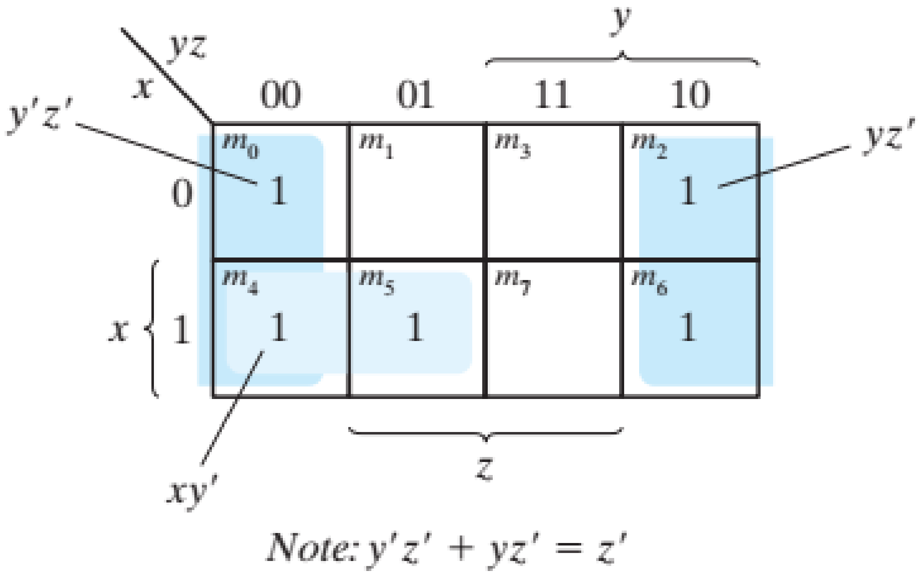


FIGURE 3.6

Map for Example 3.3, $F(x, y, z) = \Sigma(0, 2, 4, 5, 6) = z' + xy'$

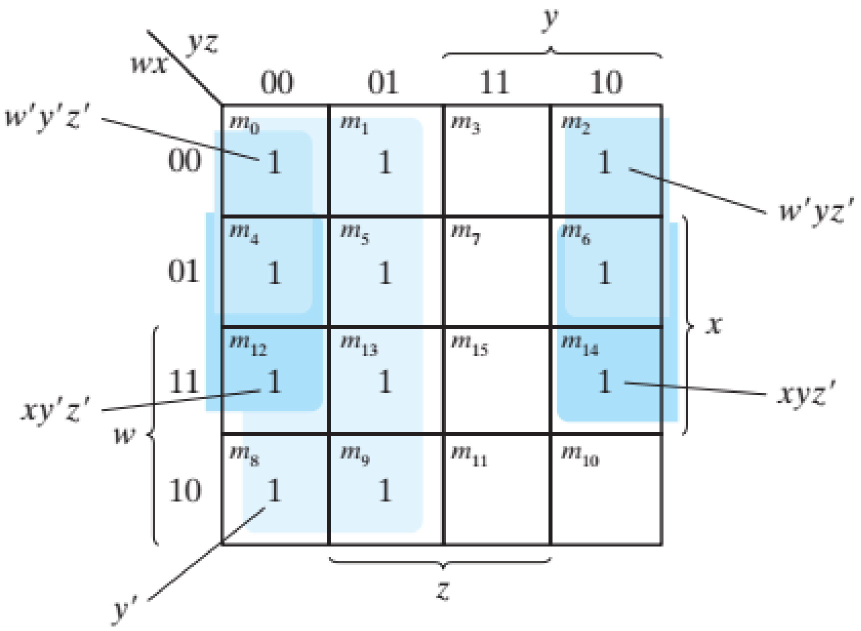
“**Digital Design**”,
M. Morris Mano

Örnek1

EXAMPLE 3.5

Simplify the Boolean function

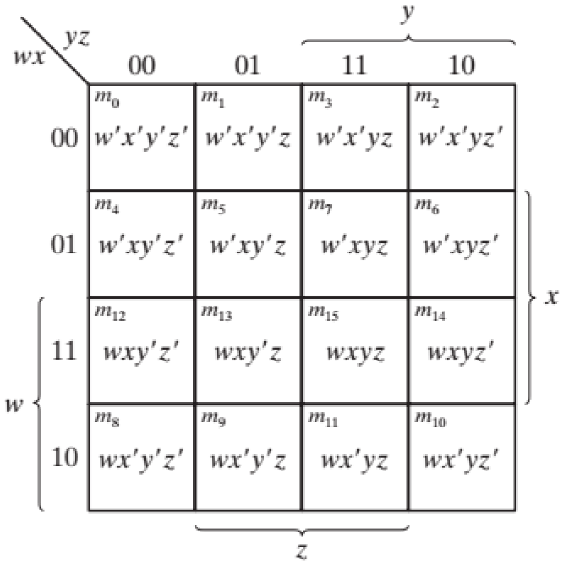
$$F(w, x, y, z) = \Sigma(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$$



wx \ yz	00	01	11	10
00	m_0 1	m_1 1	m_3	m_2 1
01	m_4 1	m_5 1	m_7	m_6 1
11	m_{12} 1	m_{13} 1	m_{15}	m_{14} 1
10	m_8 1	m_9 1	m_{11}	m_{10}

Note: $w'y'z' + w'yz' = w'z'$
$xy'z' + xyz' = xz'$

FIGURE 3.9
Map for Example 3.5, $F(w, x, y, z) = \Sigma(0,1, 2, 4, 5, 6, 8, 9, 12, 13, 14) = y' + w'z' + xz'$



wx \ yz	00	01	11	10
00	m_0 $w'x'y'z'$	m_1 $w'x'y'z$	m_3 $w'x'yz$	m_2 $w'x'yz'$
01	m_4 $w'xy'z'$	m_5 $w'xy'z$	m_7 $w'xyz$	m_6 $w'xyz'$
11	m_{12} $wxy'z'$	m_{13} $wxy'z$	m_{15} $wxyz$	m_{14} $wxyz'$
10	m_8 $wx'y'z'$	m_9 $wx'y'z$	m_{11} $wx'yz$	m_{10} $wx'yz'$

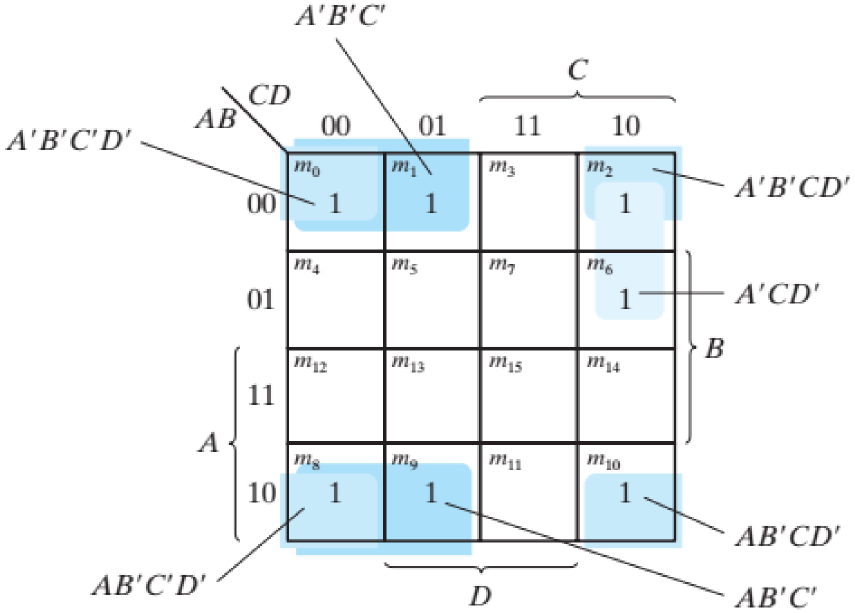
"**Digital Design**",
M. Morris Mano

Örnek2

EXAMPLE 3.6

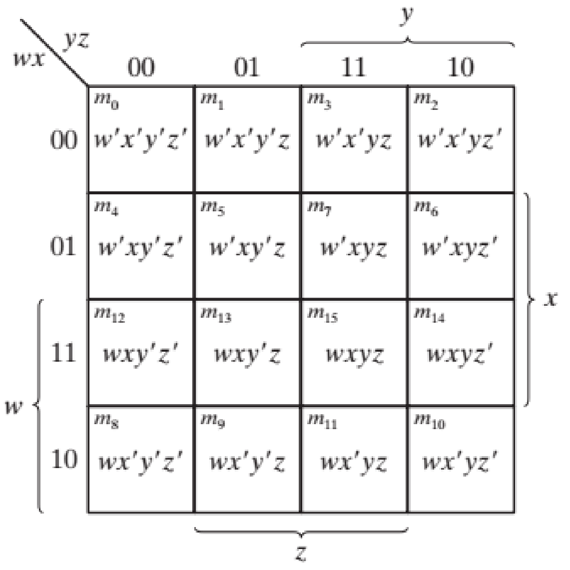
Simplify the Boolean function

$$F = A'B'C' + B'CD' + A'BCD' + AB'C'$$



Note: $A'B'C'D' + A'B'CD' = A'B'D'$
$AB'C'D' + AB'CD' = AB'D'$
$A'B'D' + AB'D' = B'D'$
$A'B'C' + AB'C' = B'C'$

FIGURE 3.10
Map for Example 3.6, $A'B'C' + B'CD' + A'BCD' + AB'C' = B'D' + B'C' + A'CD'$



wx \ yz	00	01	11	10
00	m_0 $w'x'y'z'$	m_1 $w'x'y'z$	m_3 $w'x'yz$	m_2 $w'x'yz'$
01	m_4 $w'xy'z'$	m_5 $w'xy'z$	m_7 $w'xyz$	m_6 $w'xyz'$
11	m_{12} $wxy'z'$	m_{13} $wxy'z$	m_{15} $wxyz$	m_{14} $wxyz'$
10	m_8 $wx'y'z'$	m_9 $wx'y'z$	m_{11} $wx'yz$	m_{10} $wx'yz'$

"**Digital Design**",
M. Morris Mano

Belirsiz durumlar


3.5 DON’T-CARE CONDITIONS

- Önemsiz bir minterm, mantıksal değeri belirtilmeyen değişkenlerin bir kombinasyonudur.
- Böyle bir minterm haritada 1 ile işaretlenemez, çünkü böyle bir kombinasyon fonksiyonun her zaman 1 olmasını gerektirir.
- Benzer şekilde, kareye 0 koymak fonksiyonun 0 olmasını gerektirir.
- Önemseme koşulunu 1’ler ve 0’lardan ayırmak için X kullanılır.
- Bu nedenle, haritadaki bir karenin içindeki bir X, belirli bir minterm için 0 veya 1 değerinin F'ye atanıp atanmadığını umursamadığımızı gösterir.
- Bir haritadaki işlevi basitleştirmek için bitişik kareleri seçerken, önemsiz mintermlerin 0 veya 1 olduğu varsayılabilir.
- İşlevi basitleştirirken, her önemsiz mintermi en basit ifadeyi verecek şekilde 1’ler veya 0’lar ile dahil etmeyi seçebiliriz.

Örnek

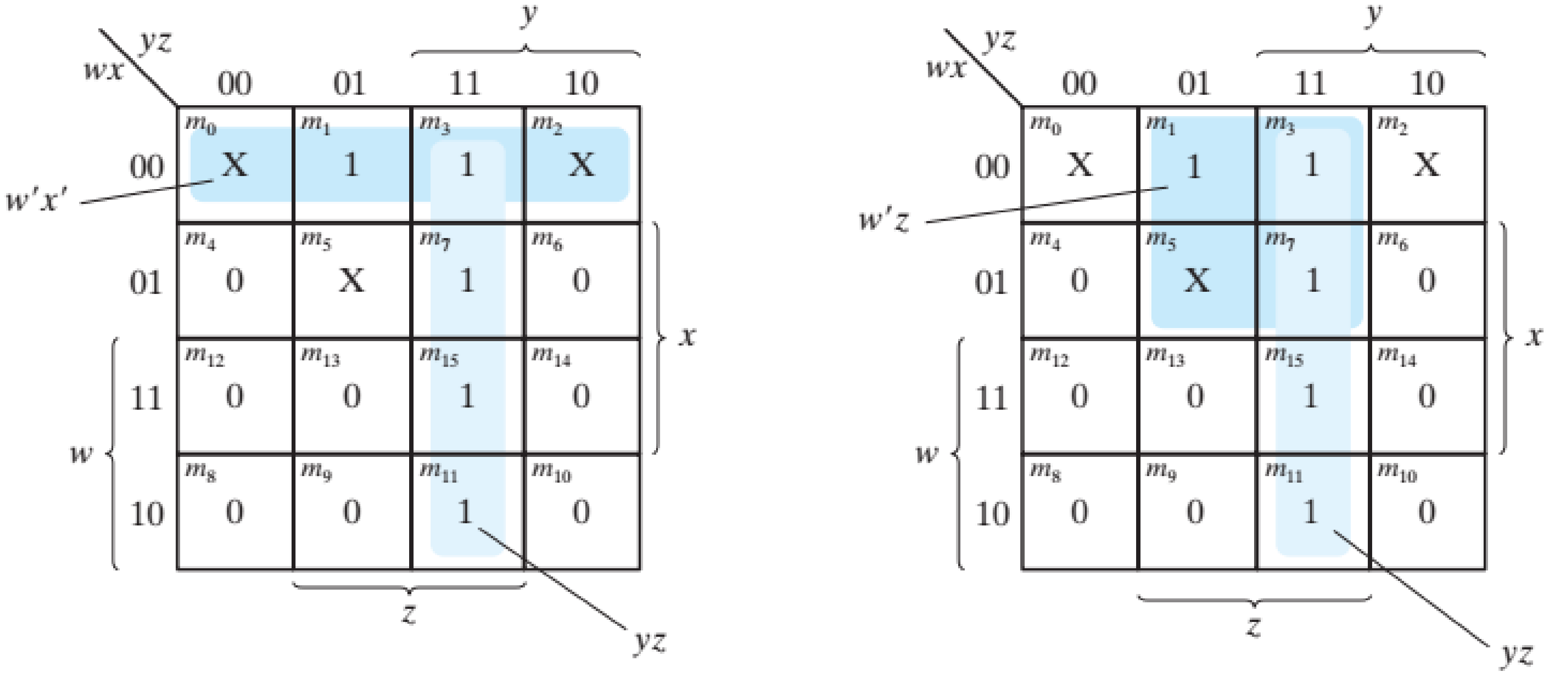
EXAMPLE 3.8

Simplify the Boolean function

$$F(w, x, y, z) = \Sigma(1, 3, 7, 11, 15)$$

which has the don't-care conditions

$$d(w, x, y, z) = \Sigma(0, 2, 5)$$



wx \ yz	00	01	11	10
00	m_0 X	m_1 1	m_3 1	m_2 X
01	m_4 0	m_5 X	m_7 1	m_6 0
11	m_{12} 0	m_{13} 0	m_{15} 1	m_{14} 0
10	m_8 0	m_9 0	m_{11} 1	m_{10} 0

(a) $F = yz + w'x'$ (b) $F = yz + w'z$

FIGURE 3.15
Example with don't-care conditions

"**Digital Design**",
M. Morris Mano

Kombinasyonel Devreler

- Sayısal sistemler için mantık devreleri, kombinasyonel veya sıralı olabilir.
- Bir birleşimsel devre, herhangi bir zamanda çıktıları yalnızca mevcut girdi kombinasyonundan belirlenen mantık kapılarından oluşur.
- Bir kombinasyonel devre, bir dizi Boole fonksiyonu tarafından mantıksal olarak belirlenebilen bir işlemi gerçekleştirir.
- Bunun aksine, sıralı devreler mantık kapılarına ek olarak depolama elemanlarını kullanır. Çıktıları, girişlerin ve depolama elemanlarının durumunun bir fonksiyonudur.
- Depolama elemanlarının durumu önceki girişlerin bir fonksiyonu olduğu için, sıralı bir devrenin çıkışları sadece mevcut giriş değerlerine değil, aynı zamanda geçmiş girişlere de bağlıdır.
- Kombinasyonel mantık kapıları, girişlerindeki sinyallerin değerlerine tepki verir ve çıkış sinyalinin değerini üretir, ikili bilgiyi verilen giriş verilerinden gerekli bir çıkış verilerine dönüştürür.

- Temel lojik kapılarından, giriş ve çıkış değişkenlerinden oluşan birleşimsel mantık devreleri; çıkışların değerinin girişlerin o anki değerlerine göre belirlenmesi mantığı ile çalışırlar. Bu devrelerde bellek ve geri besleme yoktur, bu sebeple daha önceki girişler ya da devrenin önceki şart ve çıkışları sonraki durumlarını etkileyemez. Bir birleşimsel devrenin blok diyagramı Şekil 4.1'de gösterilmektedir.

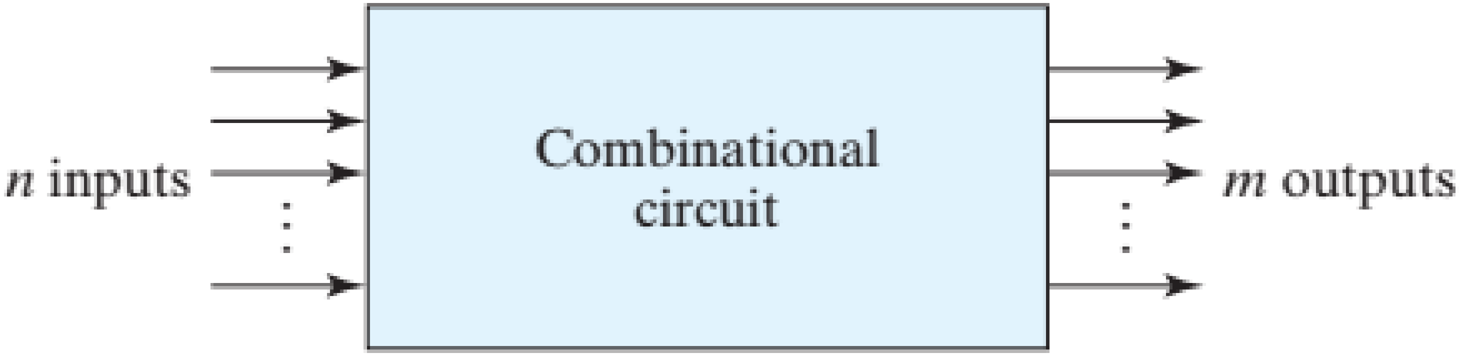


FIGURE 4.1
Block diagram of combinational circuit

Birleşimsel Mantık Devreleri;

- Uygulama alanlarına göre gruplandırılabilir:
    - Aritmetik işlem ve kıyaslama devreleri
    - Tekilleyici (Veri-bilgi seçici-multiplexer) devreler
    - Çoğullayıcı (Veri-bilgi dağıtıcı-demultiplexer) devreler
    - Kodlama devreleri

Aritmetik İşlem ve Kıyaslama Devreleri

- Bu tür devrelerin başlıcaları;

- Toplayıcı (adder) devreler
  - Yarı Toplayıcı
  - Tam Toplayıcı
- Çıkarıcı (subractor) devreler
  - Yarı Çıkarıcı
  - Tam Çıkarıcı
- Çarpıcı (multiplier) devreler
- Karşılaştırıcı (comparator) devreler

Toplayıcı Devreleri

- En temel aritmetik işlem, iki ikili rakamın toplanmasıdır.
- Bu basit ekleme, dört olası temel işlemden oluşur:
  - **0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1 ve 1 + 1 = 10.**
- İlk üç işlem bir rakamın toplamını üretir, ancak her iki toplanan bit 1'e eşit olduğunda ikili toplam (**sum**) iki basamaktan oluşur. Bu sonucun daha yüksek anlamlı bitine taşma (**carry**) denir.
- İki bitin eklenmesini gerçekleştiren bir birleşimsel devreye yarım toplayıcı denir.
- Üç bitin (iki önemli bit ve bir önceki taşma) toplamasını gerçekleştiren birleşimsel devre bir tam toplayıcıdır.
- Devrelerin adları, tam toplayıcı uygulamak için iki yarım toplayıcının kullanılabileceği gerçeğinden kaynaklanmaktadır.

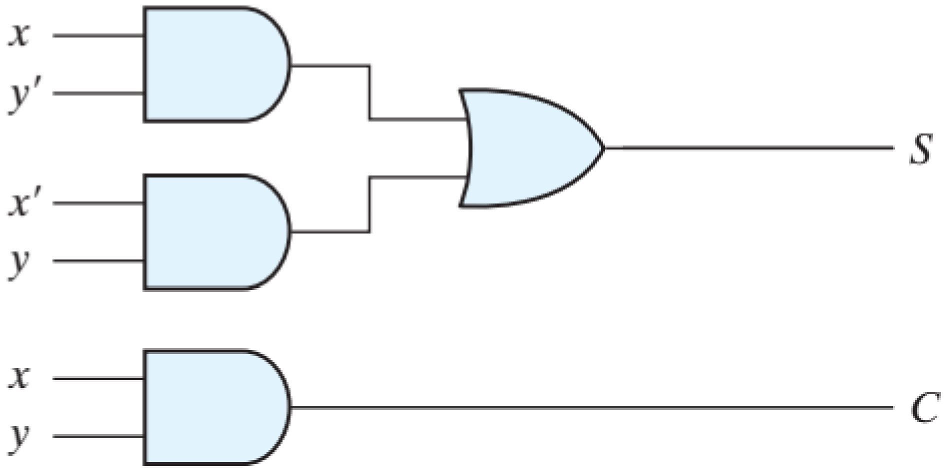
Yarı Toplayıcı Devre

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

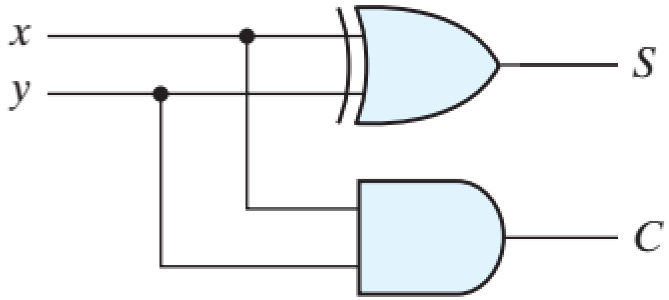
$$C = xy$$

Table 4.3
Half Adder

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



(a) $S = xy' + x'y$
$C = xy$



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

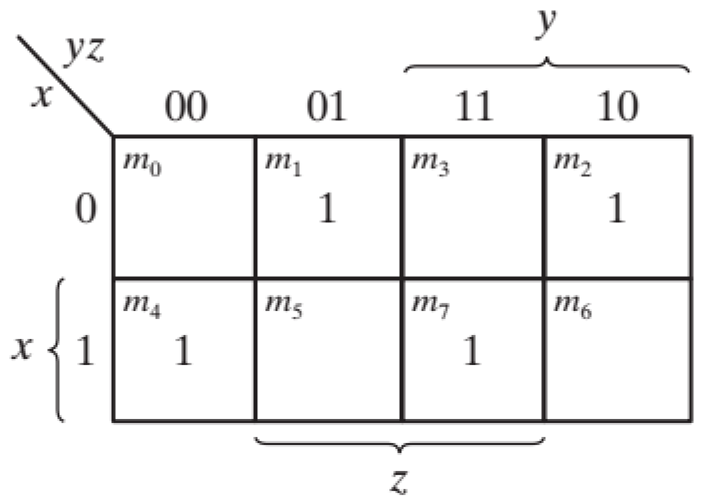
FIGURE 4.5
Implementation of half adder

Tam Toplayıcı Devre

$$S = x'y'z + x'yz' + xy'z' + xyz$$
$$C = xy + xz + yz$$

Table 4.4
Full Adder

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



(a) $S = x'y'z + x'yz' + xy'z' + xyz$

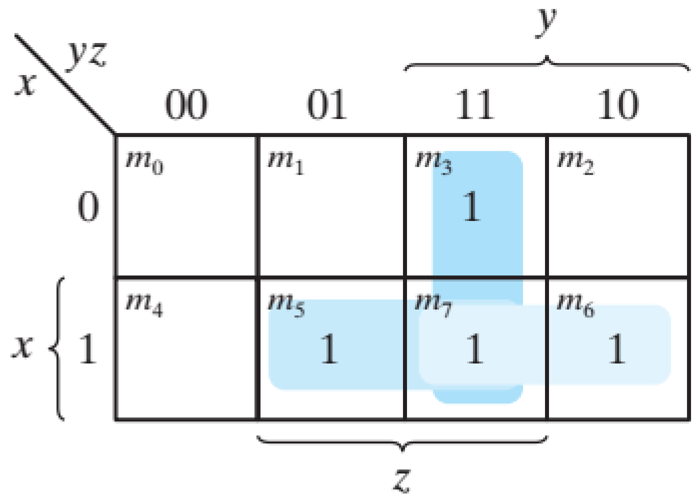


FIGURE 4.6
K-Maps for full adder (b) $C = xy + xz + yz$

Çıkarıcı Devreleri

- İki ikili sayının çıkarılması, çıkarılanın tümleyenini alıp eksiye ekleyerek gerçekleştirilebilir.
- Bu yöntemle, çıkarma işlemi, makine uygulaması için tam toplayıcılar gerektiren bir toplama işlemi haline gelir.
- Bu yöntemle, sayının her bir çıkarılan biti, bir fark biti oluşturmak için karşılık gelen önemli eksi bitinden çıkarılır.
- Yarım ve tam toplayıcılar olduğu gibi, yarı ve tam çıkarıcılar da vardır.

Çıkarıcı Devreleri

- Yarı çıkarıcı, iki biti çıkaran ve bunların farklılıklarını üreten bir birleşimsel devredir.
- Ayrıca, 1'in ödünç alınıp alınmadığını belirten bir çıktıya sahiptir.
- x - y'yi gerçekleştirmek için, x ve y'nin göreli büyüklüklerini kontrol etmeliyiz.
  - Eğer x> y ise, üç olasılığımız vardır:
    - **0 - 0 = 0, 1 - 0 = 1 ve 1 - 1 = 0.** Sonuç, fark biti olarak adlandırılır.
  - Eğer x <y ise, 0-1'e sahibiz ve bir sonraki yüksek aşamadan bir 1 ödünç almak gerekir.
- Bir yarı çıkarıcının girdi-çıktı ilişkileri için doğruluk tablosu şu şekilde türetilebilir:

Yarı Çıkarıcı Devre

$$D = x'y + xy'$$

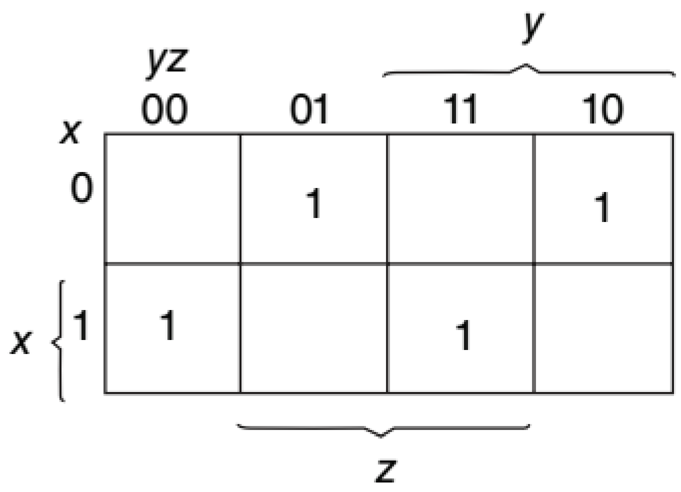
$$B = x'y$$

x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

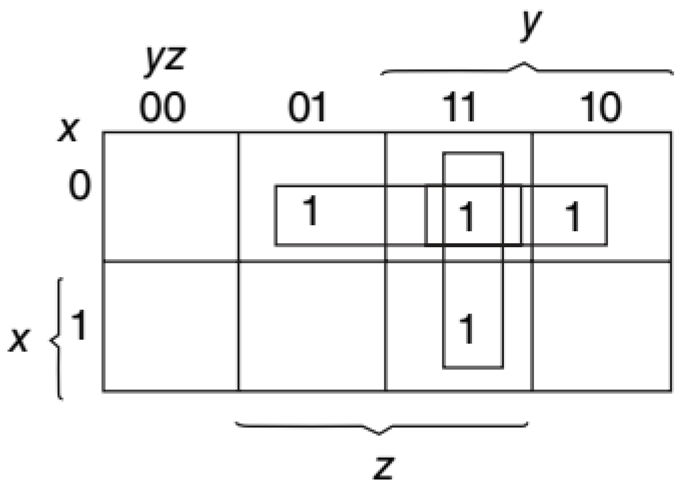
"**Digital Design**", M. Morris Mano

Tam Çıkarıcı Devre

x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



$$D = x'y'z + x'yz + x\,y'z' + x\,yz$$



$$B = x'y + x'z + yz$$

Maps for full-subtractor

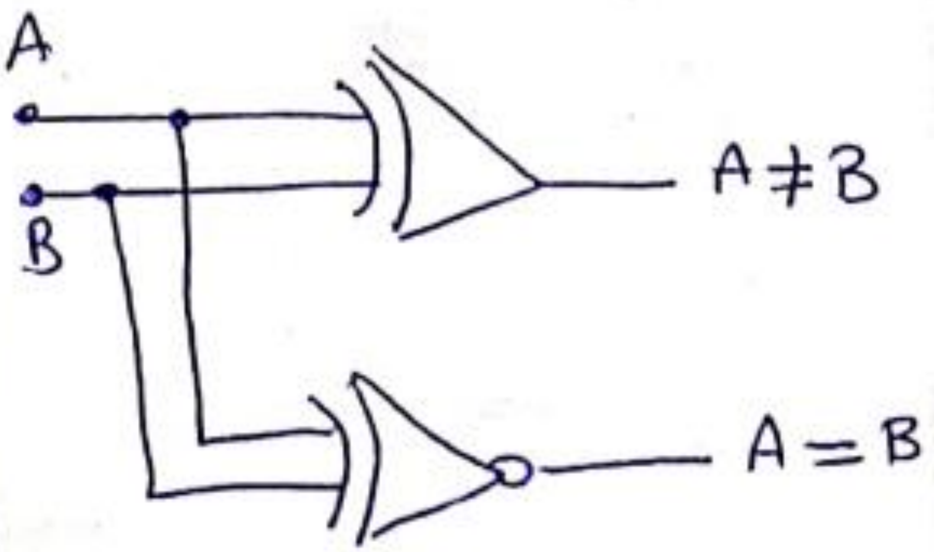
Karşılaştırıcı Devreler

- İki sayının karşılaştırılması, bir sayının diğer sayıdan büyük, küçük veya ona eşit olup olmadığını belirleyen bir işlemdir.
- Bir büyüklük karşılaştırıcısı, iki A ve B sayısını karşılaştıran ve göreli büyüklüklerini belirleyen bir birleşimsel devredir.
- Karşılaştırmanın sonucu, A > B, A = B veya A < B olup olmadığını gösteren üç ikili değişkenle belirtilir.
- A = B ve A ≠ B durumunlarını içeren karşılaştırıcı devre yarı karşılaştırıcıdır,
- A > B, A = B veya A < B olup olmadığını gösteren karşılaştırıcı devre tam karşılaştırıcıdır.

Karşılaştırıcı Devreler

Yarım karşılaştırıcı

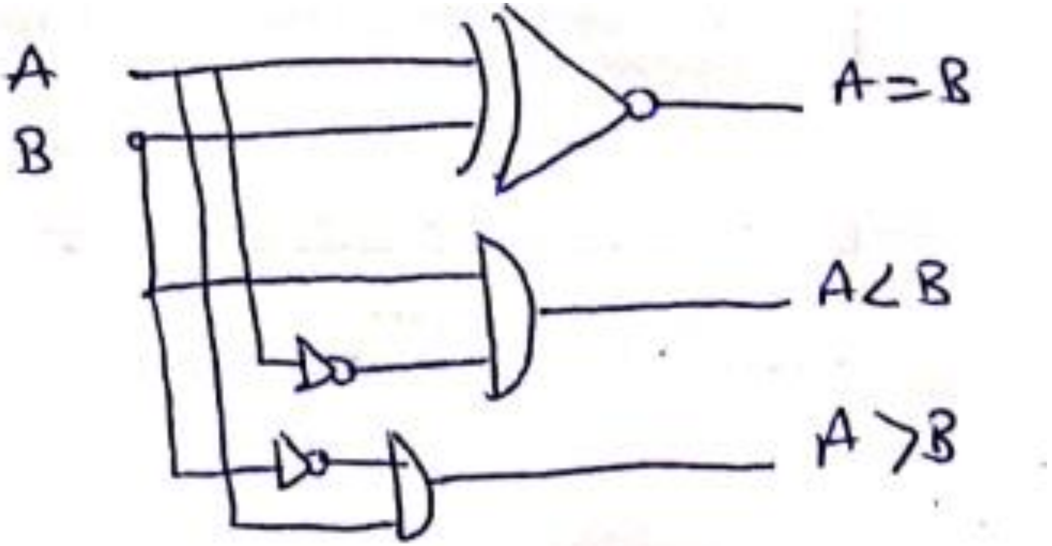
A	B	A=B	A≠B
0	0	1	0
0	1	0	1
1	0	0	1
1	1	1	0



Karşılaştırıcı Devreler

Tam Karşılaştırıcı

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



Kod çözücüler (Decoders)

- Sayısal sistemlerde ayrık bilgiler ikili kodlarla temsil edilir.
- n bitlik bir ikili kod, kodlanmış bilginin en fazla 2^n farklı elemanını temsil edebilir.
- Bir kod çözücü, ikili bilgileri n giriş hattından maksimum 2^n benzersiz çıkış hattına dönüştüren kombinasyonel bir devredir.
- n-bit kodlu bilginin kullanılmayan kombinasyonları varsa, kod çözücü 2^n'den daha az çıktıya sahip olabilir.

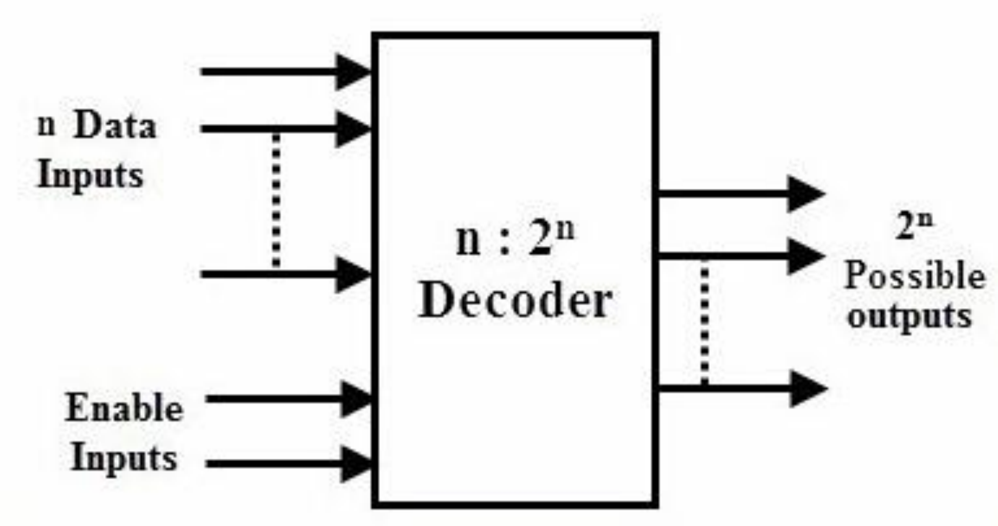
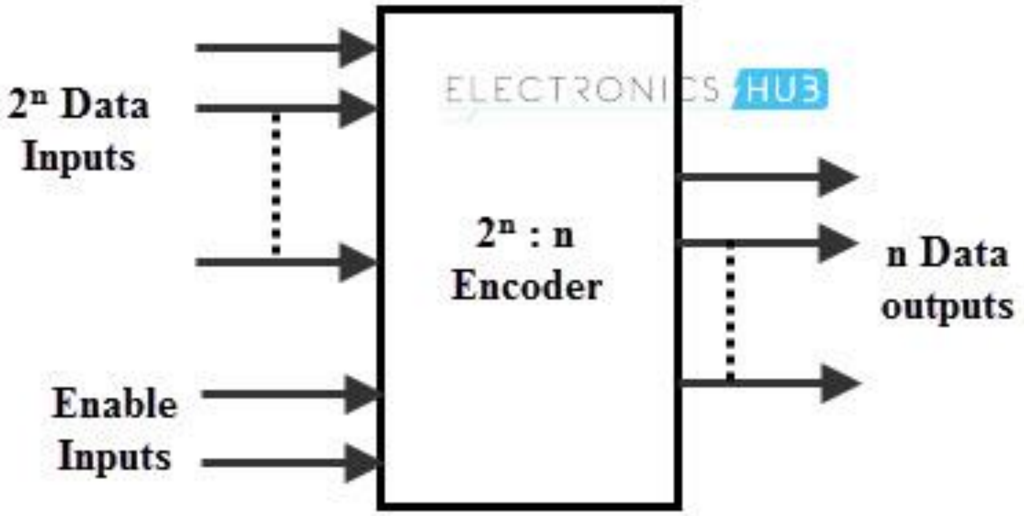


Figure: wikimedia.org

Kodlayıcılar (Encoders)

- Birleşik mantık devrelerinin en önemli uygulamalarından biri olan kodlayıcılar, bir kod çözücünün ters işlemini gerçekleştiren sayısal bir devrelerdir.
- Bir kodlayıcının 2^n (veya daha az) giriş hattı ve n çıkış hattı vardır.
- Kodlayıcılar türleri 4x2, 8x3, 16x4,.. şeklinde ifade edilir.



Tekilleyici-Veri Seçici (Multiplexer)

- Multiplexer, birçok giriş hattının birinden ikili bilgiyi seçen ve tek bir çıkış hattına yönlendiren kombinasyonel bir devredir.
- Belirli bir giriş hattının seçimi, bir dizi seçim satırı tarafından kontrol edilir.
- Normalde, hangi girişin seçildiğini bit kombinasyonları belirleyen 2^n giriş satırı ve n seçim satırı vardır.

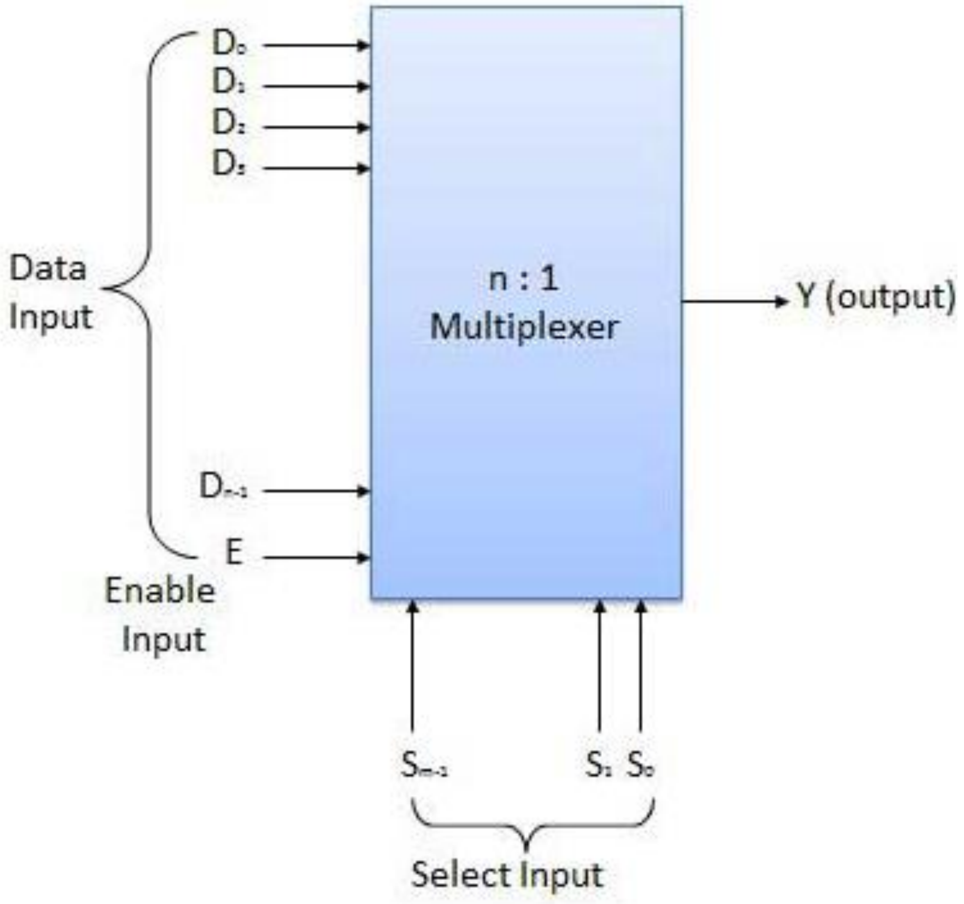
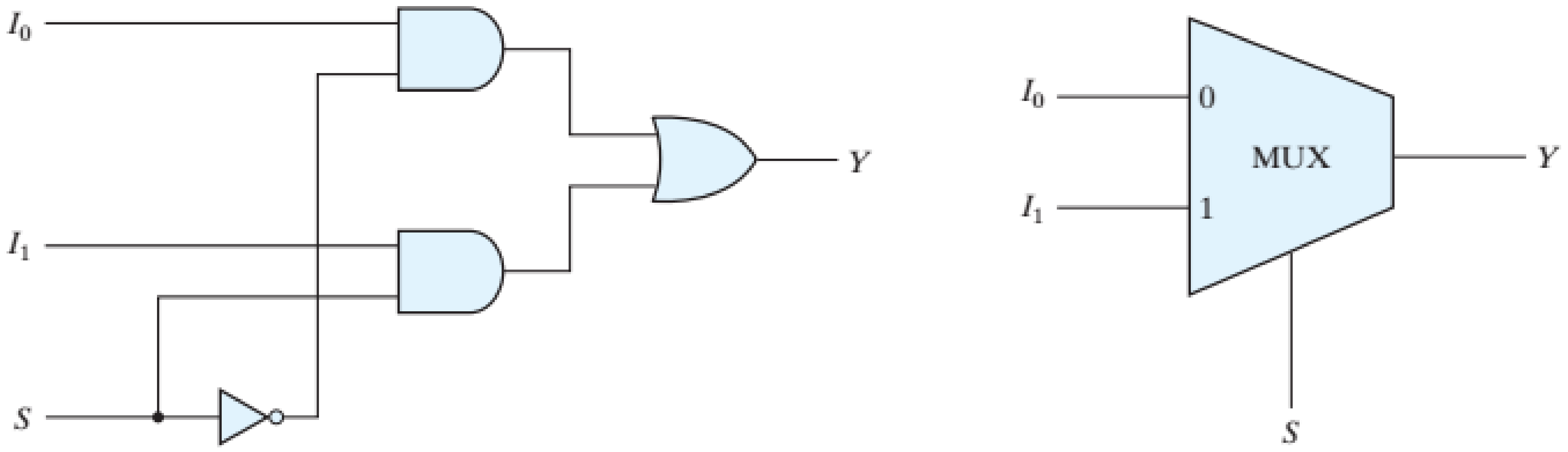


Figure: https://learn.circuitverse.org/docs/MSI/mux.html

2x1 Multiplexer



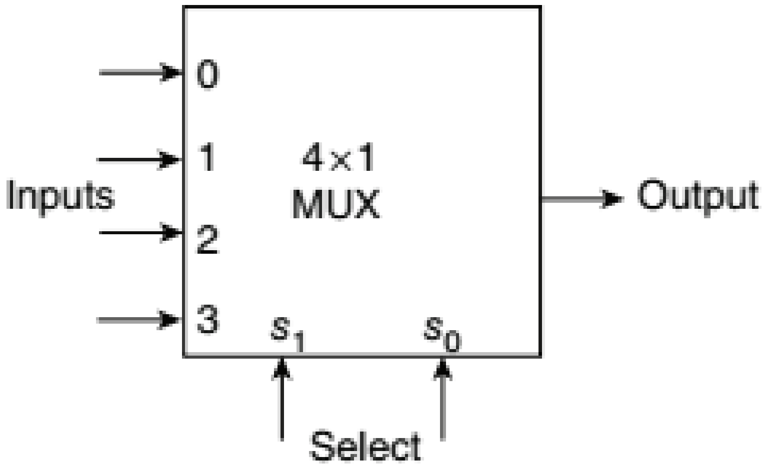
(a) Logic diagram

(b) Block diagram

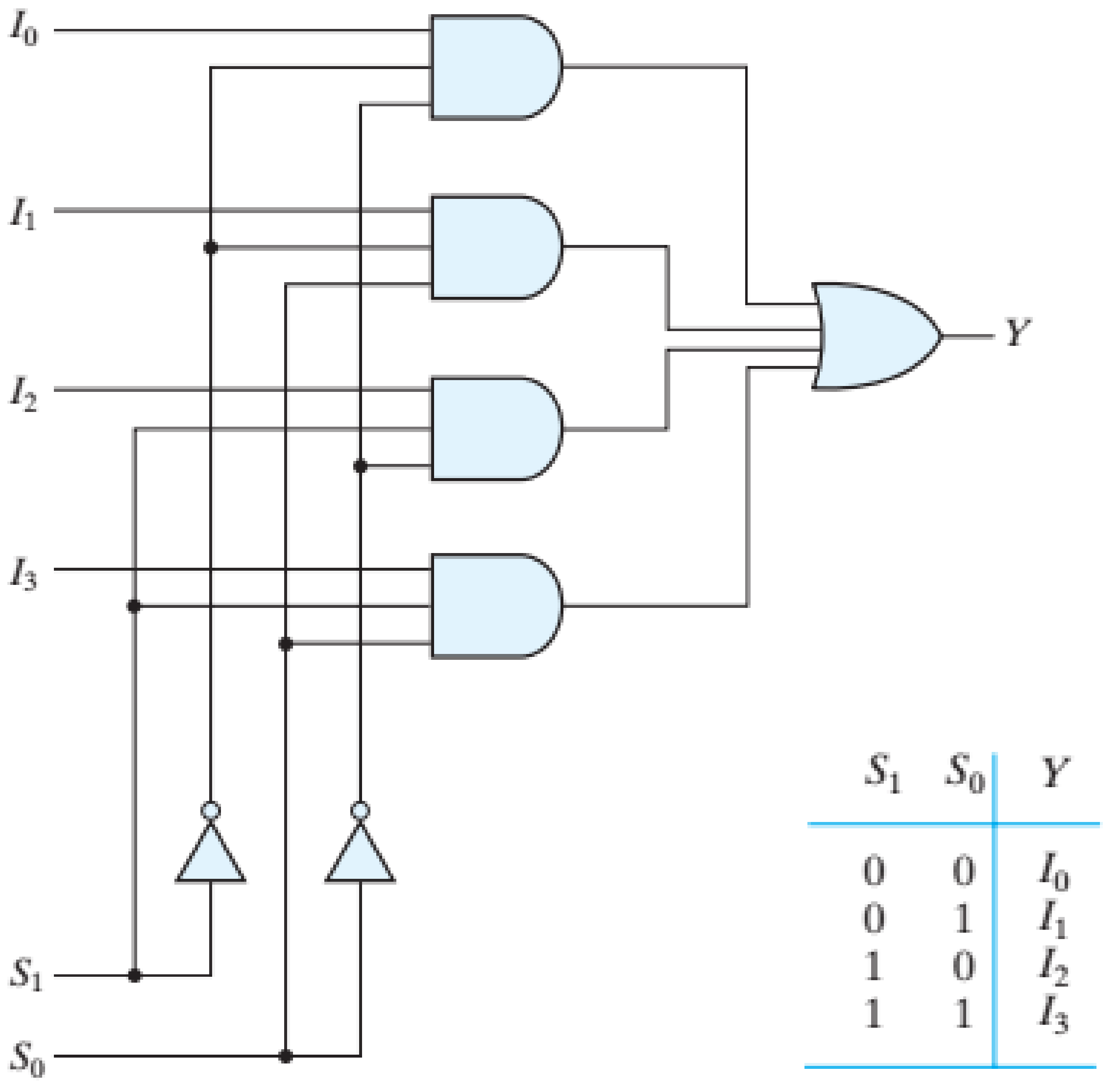
FIGURE 4.24
Two-to-one-line multiplexer

“**Digital Design**”, M. Morris Mano

4x1 Multiplexer



(a) Logic diagram



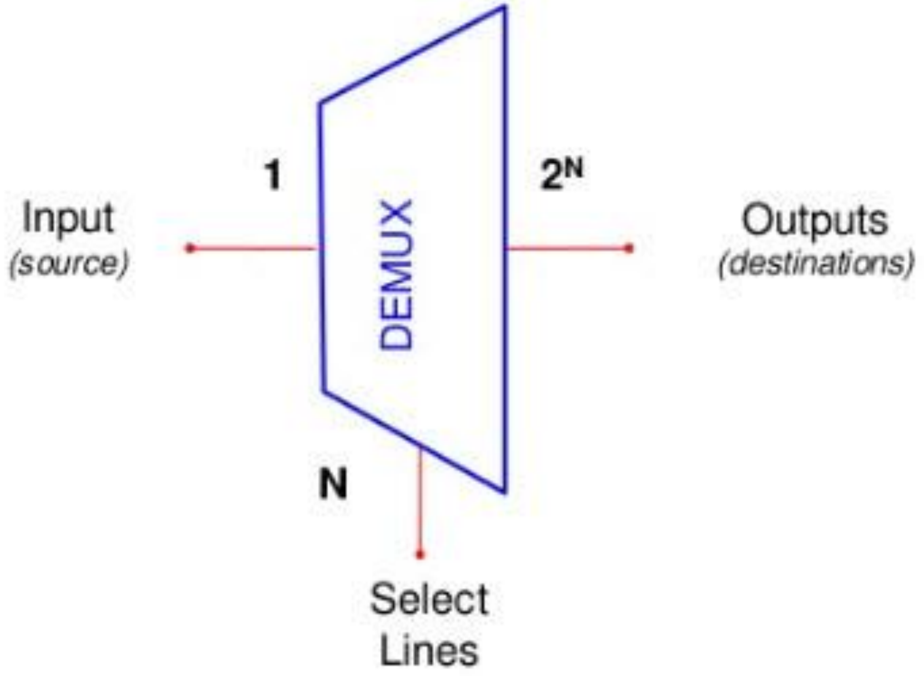
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(a) Logic diagram (b) Function table

FIGURE 4.25
Four-to-one-line multiplexer

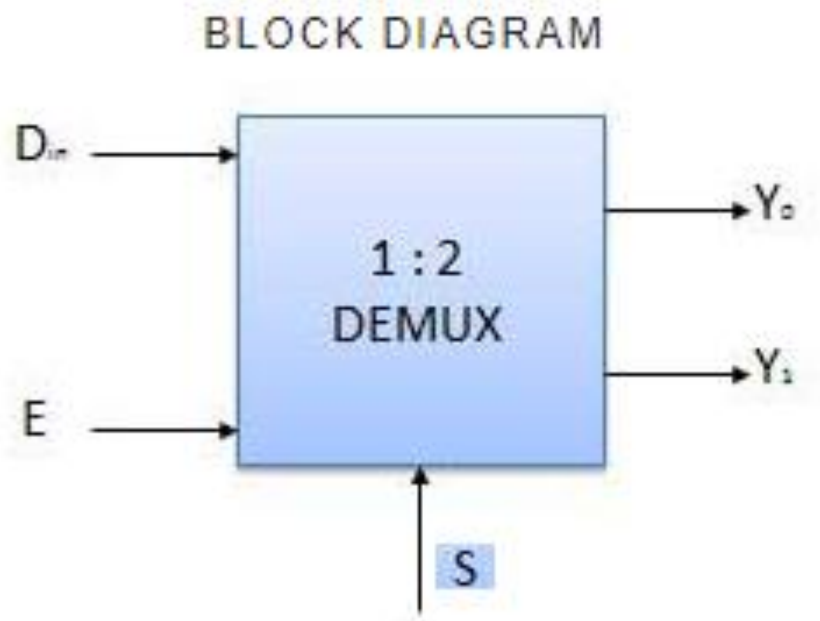
Çoğullayıcı-Veri Dağıtıcı (Demultiplexer)

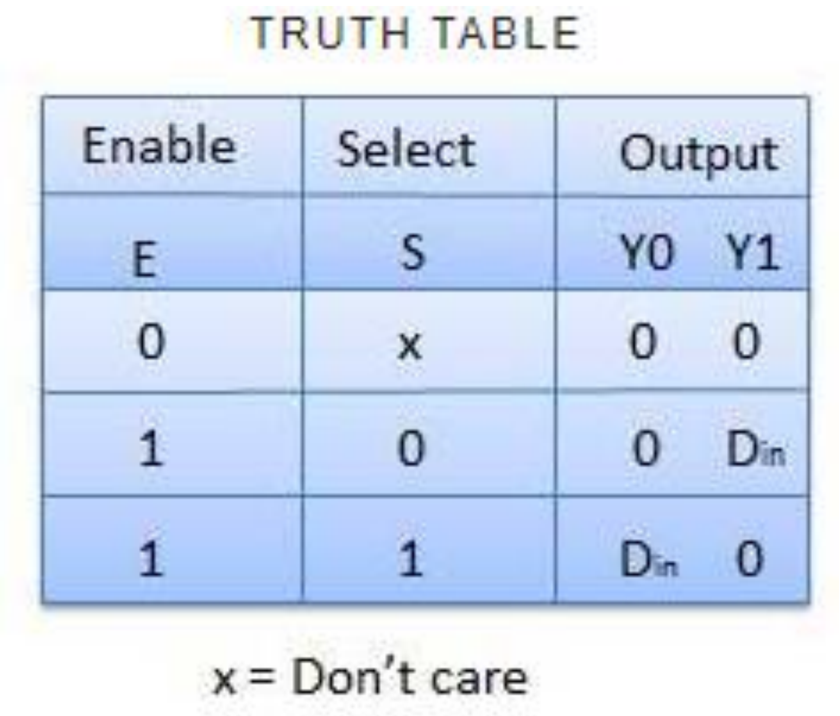
- Bir çoğullayıcı, bir tekilleyicinin ters işlemini gerçekleştirir, yani bir girişi alır ve bunu birkaç çıkışa dağıtır.
- Sadece bir giriş, n çıkış, m seçim girişi vardır.
- Bir seferde, seçilen hatlar tarafından yalnızca bir çıkış hattı seçilir ve giriş, seçilen çıkış hattına iletilir.



https://www.elprocus.com/different-types-of-demultiplexers/

1x2 Demultiplexer


BLOCK DIAGRAM

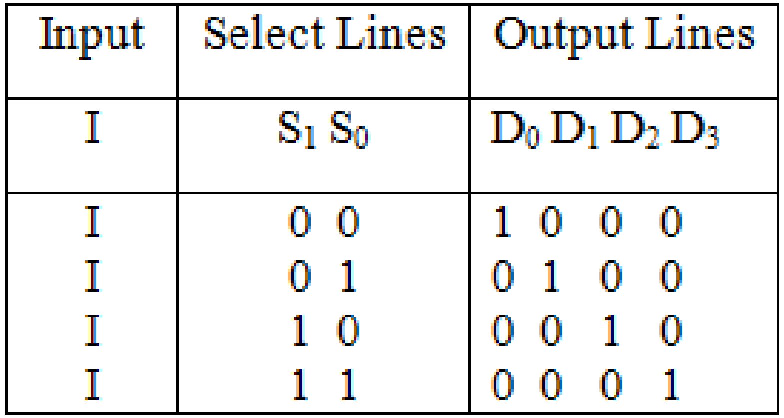

TRUTH TABLE

Enable	Select	Output	
E	S	Y0	Y1
0	x	0	0
1	0	0	D_{in}
1	1	D_{in}	0

x = Don’t care

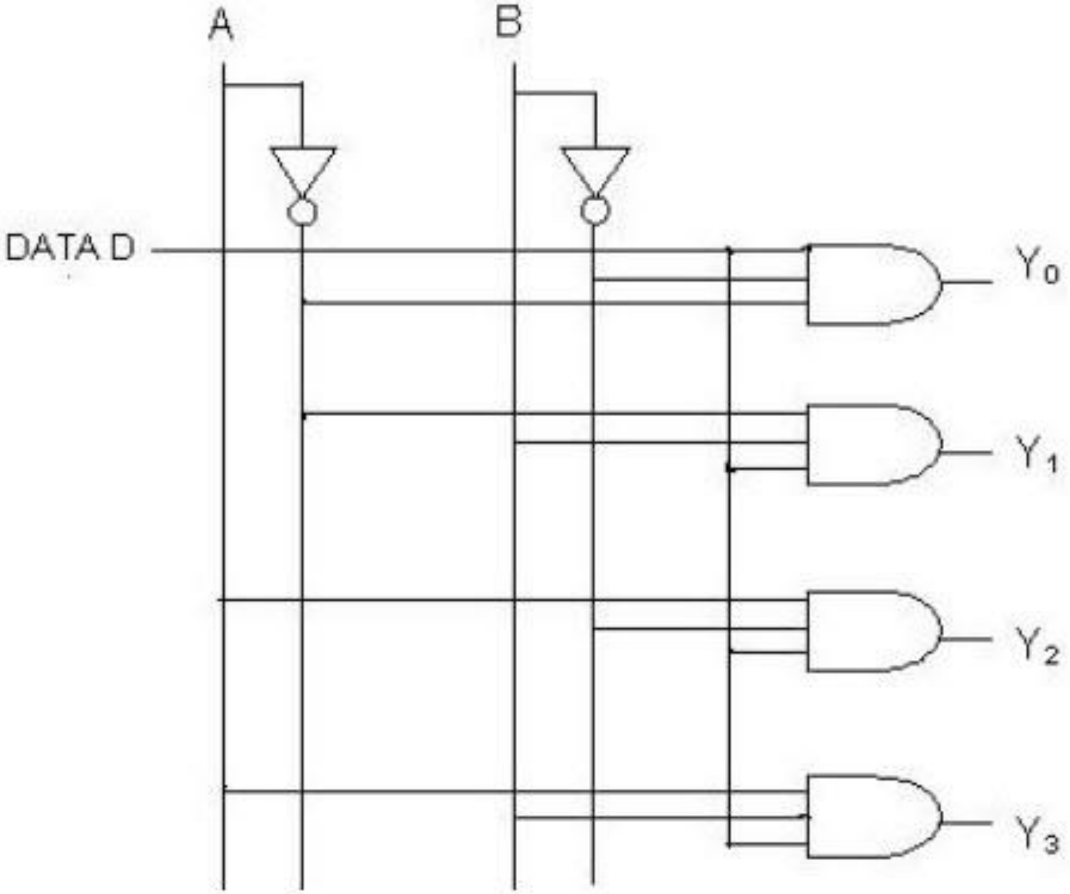
https://learn.circuitverse.org/docs/MSI/demux.html

1x4 Demultiplexer



Input	Select Lines	Output Lines
I	S_1 S_0	D_0 D_1 D_2 D_3
I	0 0	1 0 0 0
I	0 1	0 1 0 0
I	1 0	0 0 1 0
I	1 1	0 0 0 1

1 to 4 Demux Truth Table



https://www.elprocus.com/different-types-of-demultiplexers/