

Part 2

Giriş birleşimli devre:

Dijital Sistemlerin mantık devreleri **birleşimli - HAFIZASIZ**
serali - HAFIZALI olabilir.

Birleşimli (Hafızasız) devre herhangi bir zamanda çıktıları
yalnızca mevcut giriş değerleriyle belirlenen mantık kapıllarından
oluşur.

Boole işlemleri gerçekleştirir. **Her zaman aynı girişe aynı çıkış**
değerini üretir.

Daha karmaşık anahtarlama devreleri üretmek için "birleştirilen"
mantık kapılarından oluşur. **NAND, NOR, NOT** kapıları.

Geri besleme hattı olmamalıdır.

— o —

Sıralı (Hafızalı) Devre

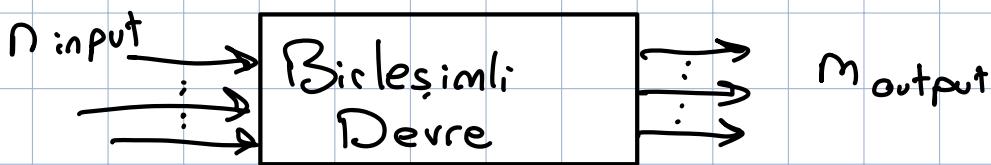
Mantık kapılalarına ek olarak depolama elementlerini da kullanır.

Girişleri, giriş fonksiyonu ve depolama elementlerinin durumları ile belirlenir.

Geri besleme bağlantıları vardır.

Yani devreler aynı giriş değerine farklı zamanlarda farklı çıktılar üretebilir.

Birleşimli (Hafızasız) Devreler.



İkili bilgisi, giriş verisinden çıkış verisine döndürür.
Hafızasız devredir.

n girişli devrenin 2^n olsı çıkışı vardır.

Her olsı girişin çıkışı her zaman tekdir ve aynı değerdedir

En önemli toplayıcılar

- Binary Adders
 - Half adder
 - Full adder
 - BCD adder
- Subtractors
 - Half Subs.
 - Full Subs.
- Coders
- Comparitors
- Decoders

Small Scale Integration < 10 logik kapı SSI \hookrightarrow NAND

Medium (MSI) 10 ~ 1000 Adder, Decoder,

Large (LSI) 1000 ~ 100.000 Mikroişlemci, bellekler

Very Large (VLSI) 100.000 ~ Million Gelişmiş tüm mikroişlemciler

Birleşimli Derle Analizi

Nasıl yapılır

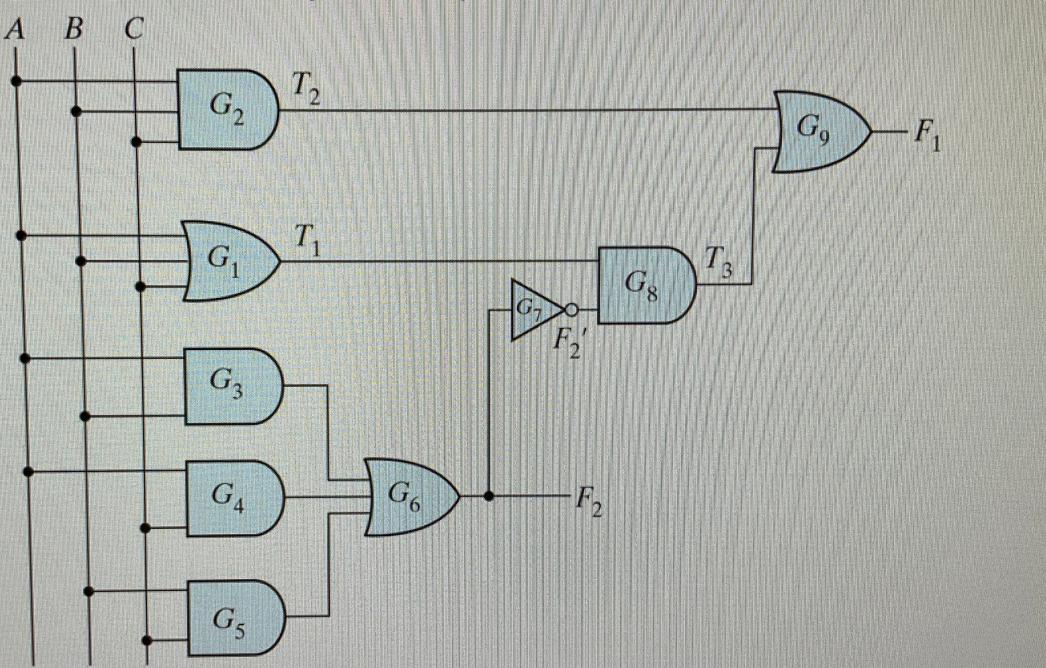
Geri besleme yolu olmamalı !

Boole fonksiyonu doğruluk tablosu oluşturabilmesi için.

Tasarım doğrulama. Devrenin çıktı, fonksiyonu bulunabilir olmalıdır.

Tasarımı (1/2)

- Problem: ifade edin (sistem spesifikasyonu)
 - Girdi ve çıktıların belirlenmesi;
 - Girdi ve çıktıya sembol ataması.
- Doğruluk tablosunu çıkarın
- Sadeleştirilmiş boole fonksiyonunu elde edin
- Matik diagramını çizin ve doğruluğunu kontrol edin.



Minterm olarak yazarsak
 $f_1(A, B, C, D) = \sum(1, 2, 4, 7)$

A	B	C	F_2	F'_2	T_1	T_2	T_3	F_1
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

Tasarım, (2/2):

- Fonksiyonel tanım:
- Boole fonksiyonu
- Donanım tasarım dili: (Verilog HDL, VHDL)
- Sematik çizim
- Mantıksal küfürltme (minimizasyon)
 - Kapıların sayısı
 - bir kapının girdi sayısı
 - iletim geçikmesi
 - bağlantı sayısı
 - kullanım (sürüç) ile ilgili kısıtlar.

Kod dönüştürücü

Bazı durumlarda bir sistemin çıkışı başka bir sistemin girdisi olarak kullanılabılır.

Bu durumlarda iki sistem arasına **dönüştürücü devre** konulmalıdır.

BCD \rightarrow Excess-3 dönüştürücü devre

- Genel tasarım giriş çıkış bacaklarına göre kabaca çizilir.
Bunun için önce bacak sayıları belirlenmelidir.
- BCD 0'dan 9'a sayıların 0 ve 1 ifadesidir.
- Giriş sayısı önceden belirlenmelidir. Bu da 6'tır. Yarı dövenin 6 giriş'i vardır.
- $2^6 = 64$ farklı değer gösterebilir. ($0, 1, 2, \dots, E, F$)
Fakat BCD kodlamada 10 farklı ($0, 1, 2, \dots, 9$) sembol vardır.

Bu yüzden 6 adası durum BCD kodlamada yoktur.

Önemsizdir tabloda X konur.

X'ler ihtiyaca göre 0 veya 1'ler ile gruplanabilir.
K-Map oluşturulmalıdır.

INPUT BCD				OUTPUT EXCESS-3 CODE					
DEC	A	B	C	D	W	X	Y	Z	DEC
0	0	0	0	0	0	0	1	1.	3
1	0	0	0	1	0	1	0	0	4
2	0	0	1	0	0	1	0	1.	5
3	0	0	1	1	0	1	1	0	6
4	0	1	0	0	0	1	1	1.	7
5	0	1	0	1	1	0	0	0	8
6	0	1	1	0	1	0	0	1.	9
7	0	1	1	1	1	0	1	0	A
8	1	0	0	0	1	0	1	1.	B
9	1	0	0	1	1	1	0	0	C
10	1	0	1	0	X	X	X	X	-
11	1	0	1	1	X	X	X	X	-
12	1	1	0	0	X	X	X	X	-
13	1	1	0	1	X	X	X	X	-
14	1	1	1	0	X	X	X	X	-
15	1	1	1	1	X	X	X	X	-

Combinational table oluştururken

- Her sütun için ayrı K-map çizilmeli
- Sadeleştirilmeli
- tasarım elde edilmeli
- Giriş çıkış sıralamasına göre birleştirilerek çizilmelidir.

$$ABCD + 3(0011)$$

eklenirse

$$W \times Y \times Z$$

sütunları bilinir.

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

$$Z = D' \quad \text{tabloda örüntü var}$$

AB \ CD	00	01	11	10
00	1	0	1	0
01	1	0	1	0
11	X	X	X	X
10	1	0	X	X

$$y = \overline{CD} + CD$$

$AB \backslash CD$	00	01	11	10
00	0	1	(1)	1
01	1	0	0	1
11	X	X	X	X
10	0	1	X	X

$$x = \overline{B}D + \overline{B}C + BD'$$

$$\overline{B}CD' + \overline{C}D + \overline{B}C$$

$AB \backslash CD$	00	01	11	10
00				
01				
11	X	X	X	X
10	1	1	X	X

$$w = A + \overline{B}D + \overline{B}C$$

A

C

B

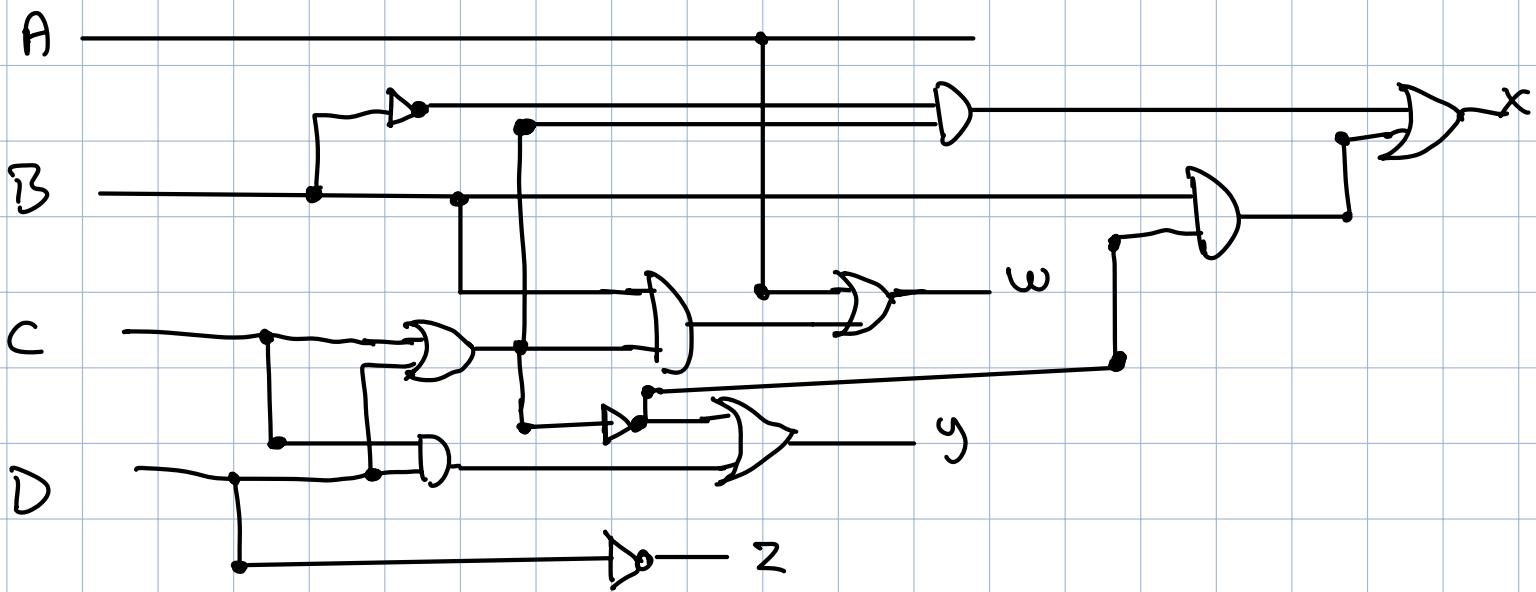
D

$$\checkmark Z = D'$$

$$\checkmark y = CD + C'D' = \checkmark CD + (C+D)'$$

$$x = B'(C+D) + B(C+D)'$$

$$\checkmark w = A + B(C+D)$$



Binary (ikili) Toplayıcılar

Half Adder

2 bit toplar

Full Adder :

2 bit + carry toplar
iki half adder bir full adder yapar.

Half Adder:

X	Y	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

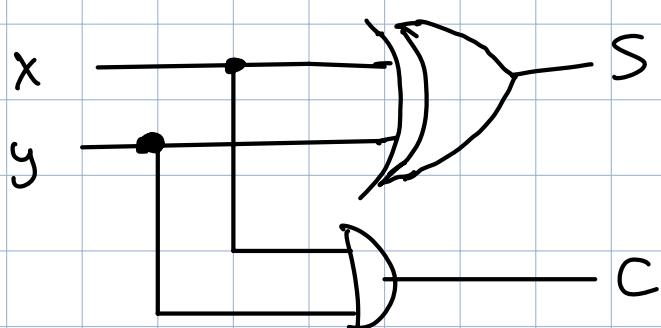
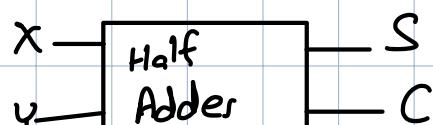
AND

XOR

C çıktısi soldaki bite karşılık gelir.
S çıktısi sağdaki

$$S = x \oplus y$$

$$C = xy$$



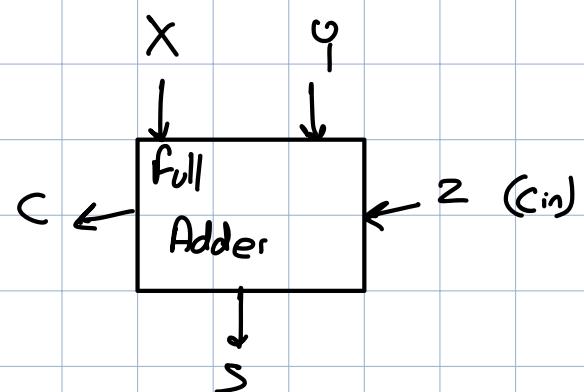
Full Adder:

3 bit toplanır

X ve y toplanacak biti

z bir önceki islemlerden gelen Carry'i tensil eder

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



Cıktının 1 olduğu yerleri SOP formunda yazalım.

1 olan yerler: SOP formunda yazalım

$$S = \overline{x}\overline{y}z + \overline{x}yz' + xy\overline{z}' + xyz$$

$$C = \overline{x}\overline{y}z + x\overline{y}z + xy\overline{z}' + xyz$$

Şimdi S ve C bool fonksiyonlarının K-Map ile soraleştirelim

$x \backslash yz$	00	01	11	10
0	m_1	1		m_2
1	m_4	1	m_3	1

Sum K-Map ile
Sadeleşmiyor.

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

$x \backslash yz$	00	01	11	10
0			1	
1	1	1	1	1

$$b) C = yz + xz + xy$$

$$S = \underline{x'y'z} + \underline{x'y'z'} + \underline{xy'z'} + \underline{xyz}$$

$$= z(x'y' + xy) + z'(x'y + xy')$$

$xNOR = A'$ over $xOR = A$ dersen

$$\Rightarrow S = zA' + z'A$$

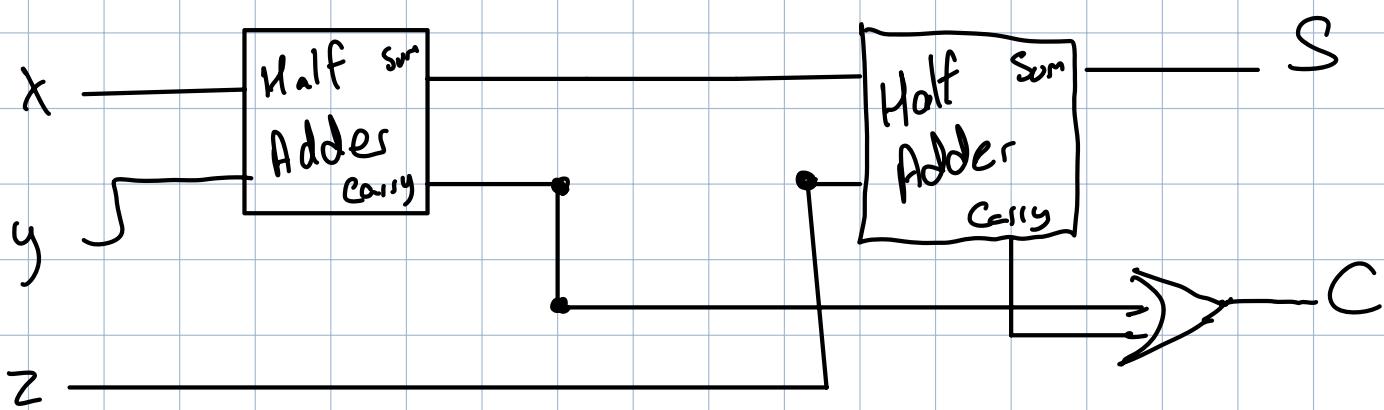
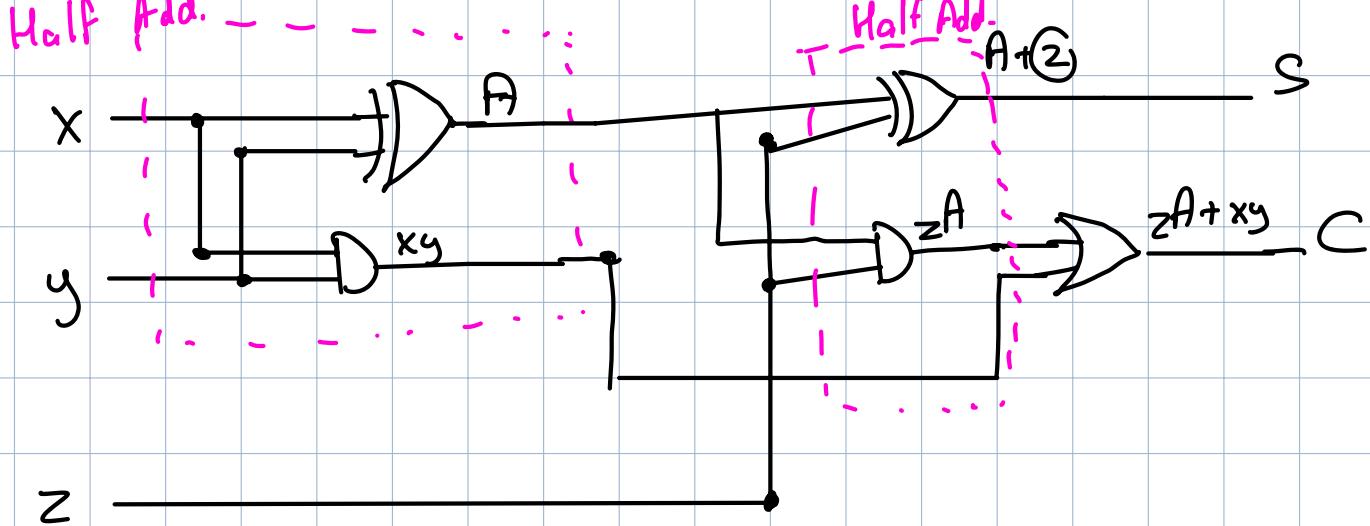
$= S = z \oplus A$
 $= z \oplus (x \oplus y)$

$$C = x'y'z + x'y'z' + \underline{xy'z'} + \underline{xyz}$$

$$= z(x'y + xy') + xy(z' + 1)$$

$C = zA + xy$

$$C = z(x \oplus y) + xy$$



Binary Adder (Parallel)

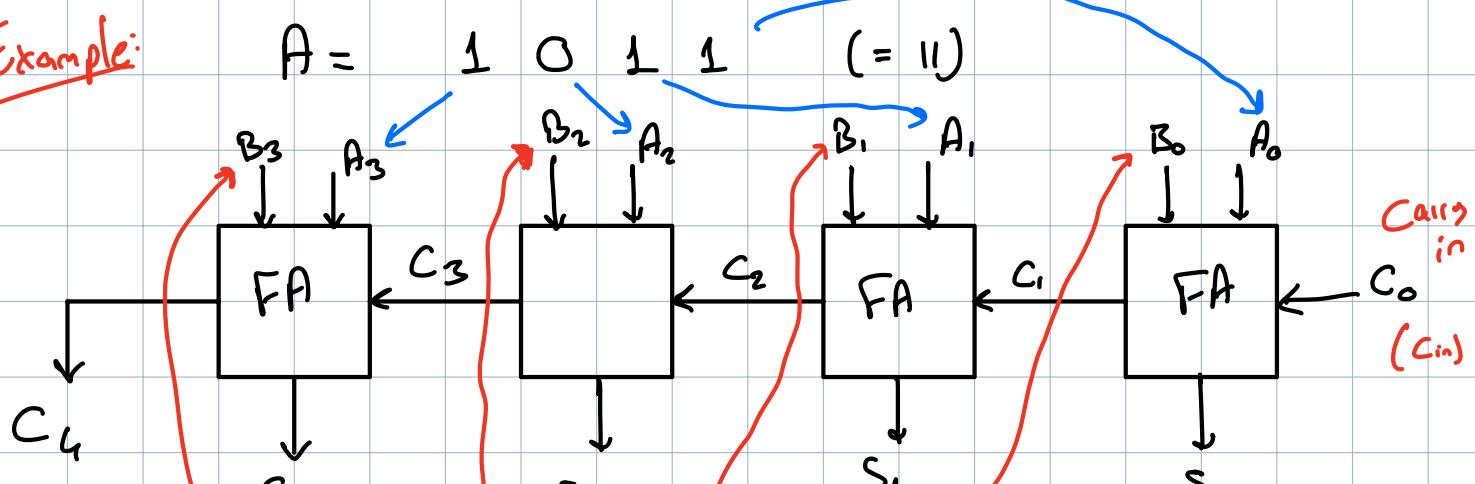
n bitlik bir sayıyı toplamak için n -adet Full Adder

yada 1 HalfAdder ve $n-1$ FA

Kullanılır

En az önemli bitin ucuna 0 verilir.
(LSB)

Example:



$$\begin{array}{r} \text{input carry} \\ \text{Augend} \\ \text{Addend} \\ \text{Sum} \\ \text{Output carry} \end{array}$$

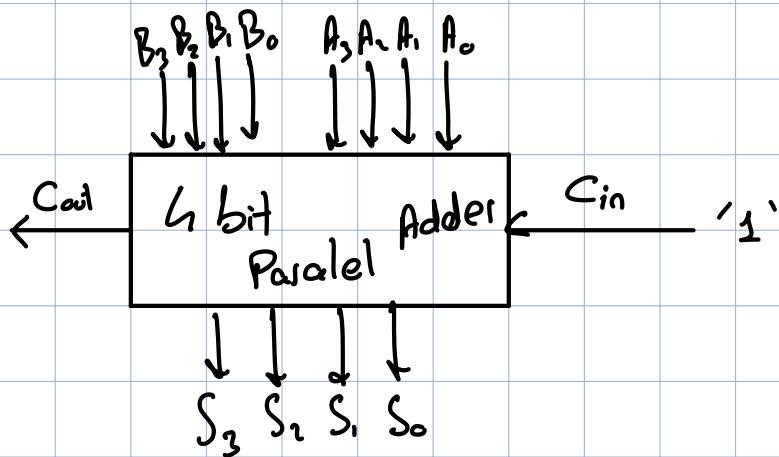
$$\begin{array}{r}
 & 0 & 1 & 1 & 0 \\
 & | & | & | & | \\
 \text{C}_i & & & & \\
 \text{A}_i & 1 & 0 & 1 & 1 \\
 \text{B}_i & 0 & 0 & 1 & 1 \\
 \hline
 \text{S}_i & 1 & 1 & 1 & 0 \\
 & | & | & | & | \\
 & 0 & 0 & 1 & 1 \\
 & & & & \text{C}_{i+1}
 \end{array}$$

Gıkarma Devresi

$$S = A - B$$

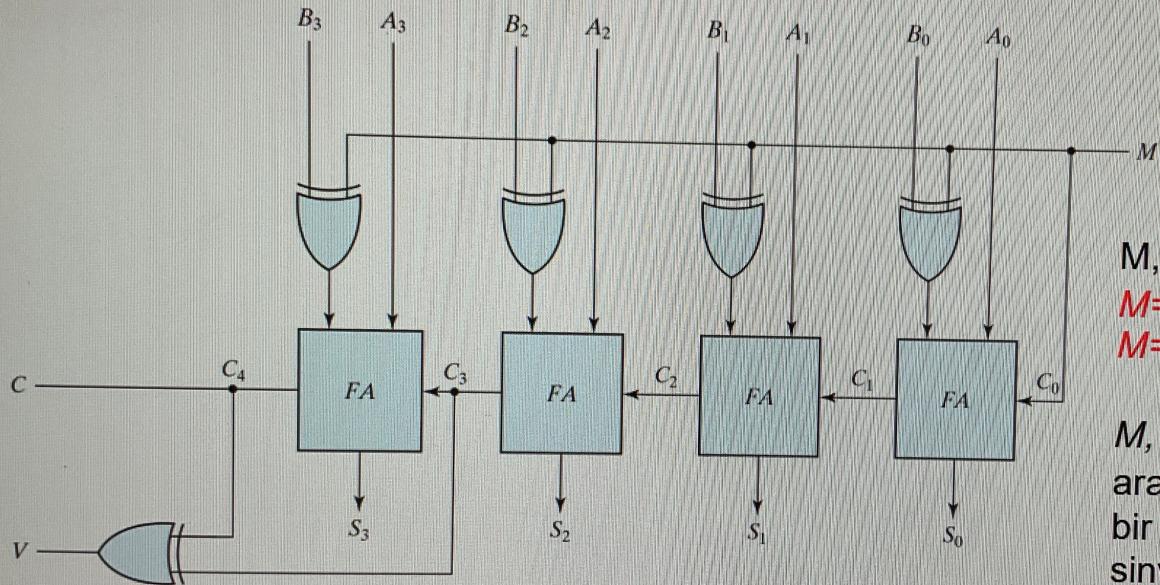
B sayısının 2^k 's complementi alınır ($B' + 1$)

$$S = A - B = A + (B' + 1)$$



4 Bit Toplayıcı ve Çıkarıcı

Four-bit adder-subtractor (with overflow detection).



M, modu belirler:
M=0 toplama ve
M=1 çıkarma

M, Topla ve Çıkar arasında değişen bir "kontrol sinyali"dir ("data sinyali" değil)

v=0 taşma (overflow) yok
v=1 taşma var

$$B \oplus M = B'M + Bm' \quad M=0 \text{ için}$$

$B' + B = 1$ Toplama

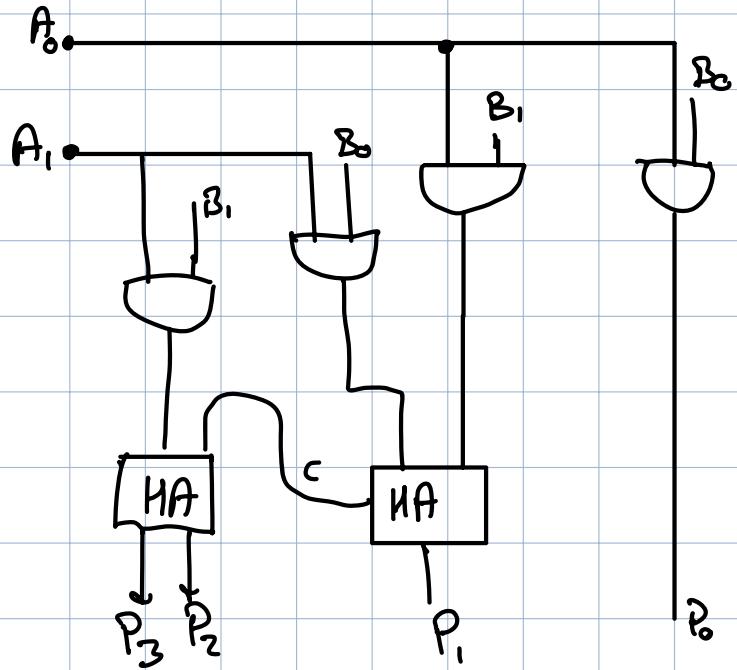
$M=1$ için

$B' + 0 = B'$ Çıkarma

Part 2:

Binary (ikili) Çarpma

$$\begin{array}{r}
 B_1 \quad B_0 \\
 A_1 \quad A_0 \\
 \hline
 A_0 B_1 \quad A_0 B_0 \\
 A_1 B_1 \quad A_1 B_0 \\
 \hline
 P_3 \quad P_2 \quad P_1 \quad P_0
 \end{array}$$

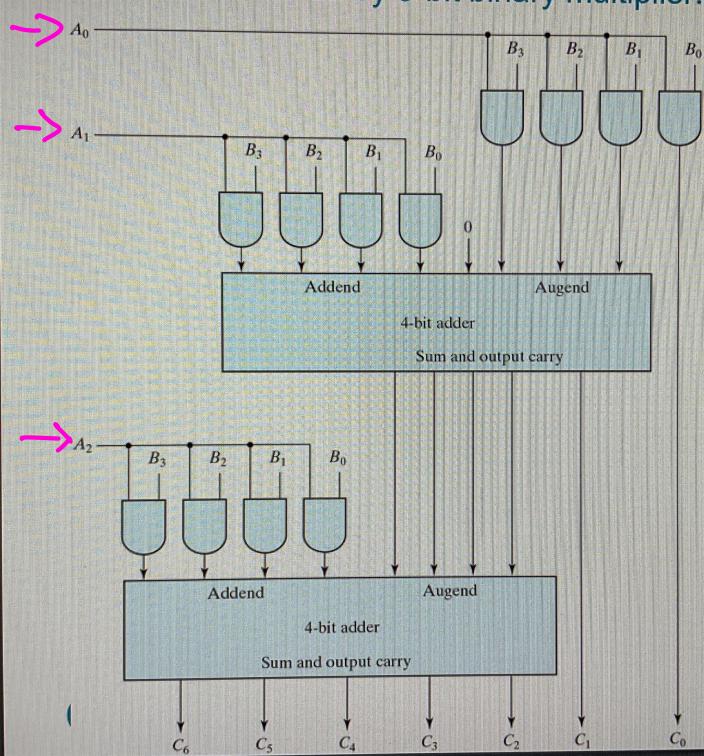


K J

4-bit \times 3-bit binary multiplier

Figure 4.16

4-bit by 3-bit binary multiplier.



(Carpan)

Sayı 1 = $A_2 A_1 A_0$

J = 3 bit

Çarpilan

Sayı 2 = $B_3 B_2 B_1 B_0$

k = 4 bit

3 bit \times 4 bit = 7 bit

(İhtiyaç listesi) $3 \times 4 = 12$ AND

$J - 1 = 2 \text{ adet } (k = 4 \text{ bit toplayıcı})$

Sonuç = 7 bit ($\frac{4+3}{k}$)

$J \rightarrow$ Çarpan bit sayısı

$K \rightarrow$ Çarpılan bit sayısı

$J \times K$ adet AND kopisi

$J-1$ adet K bitlik toplayıcı

Büyüklik Karşılaştırıcılar

$n =$ bit sayısı

4 bit büyüklik karşıtıcı

n^2 orantılı büyür

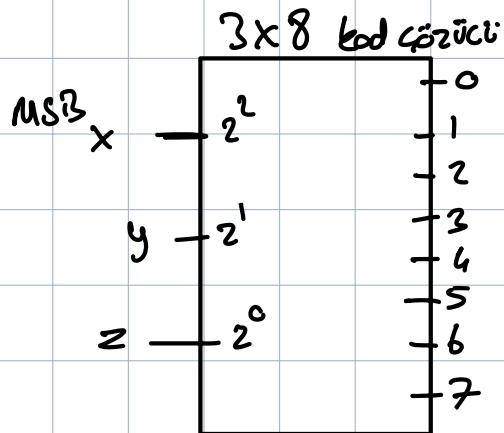
Kod Gözüçü Devre (Decoders)

• Verilen ikili girdiye göre çıktılarından birini seçer.

• n bitlik kodu 4^n çıktı yapar, yalnızca biri aktif olurak şekilde dönüştürür.

Uygulama Alanları

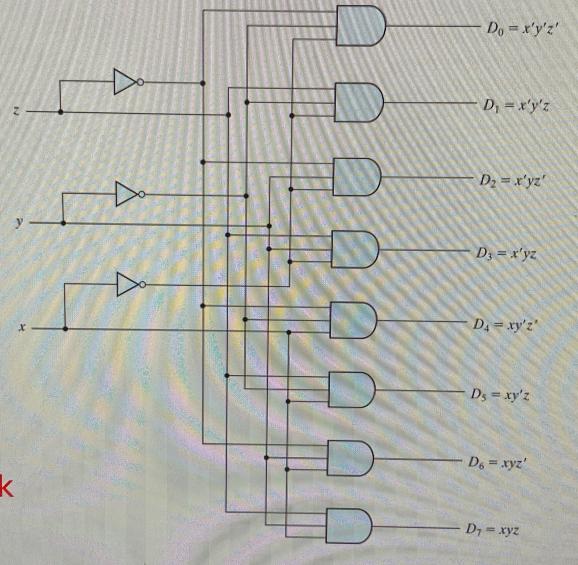
- 2^3 liden 8^3 lige kod sözme
- Belirli adresi seçimi
- Herhangi bir seçme işlemi
- Herhangi bir mantık fonksiyonunu gerçekleştirmek için kullanılabilir.



Inputs			Outputs								
	x	y	z	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

Sadece 1 çıkış aktiftir!

minitermlerin sıralanmasıdır



14

5 çıktısi ne zaman seçilir
xy'z olduğunda

Gelişmiş Kod Görüçüler Etkinleştirme Uglu Kod Görüçüler
(Advanced Decoders)

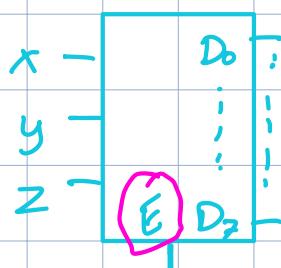
Etkinleştirme (Enabling):

Kontrol signalidir bir şeyin olmasına izin verir veya engeller.

- Durum (state) ^{söyle} tanımlanabilir.
Aktif, ON, Enabled
Pasif, OFF, Disabled

Kontrol durumunun polaritesi ise:

Aktif yüksek - Sematik simbolde balon yoktur



Aktif düşük - Balon Vardır

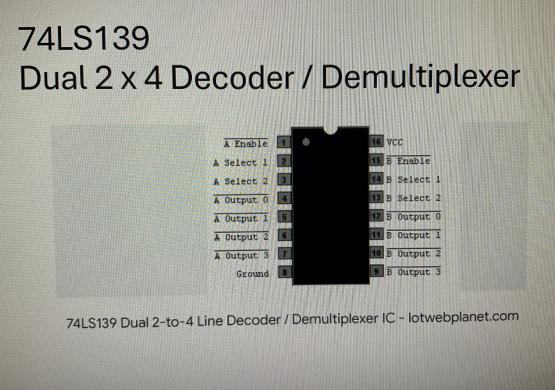
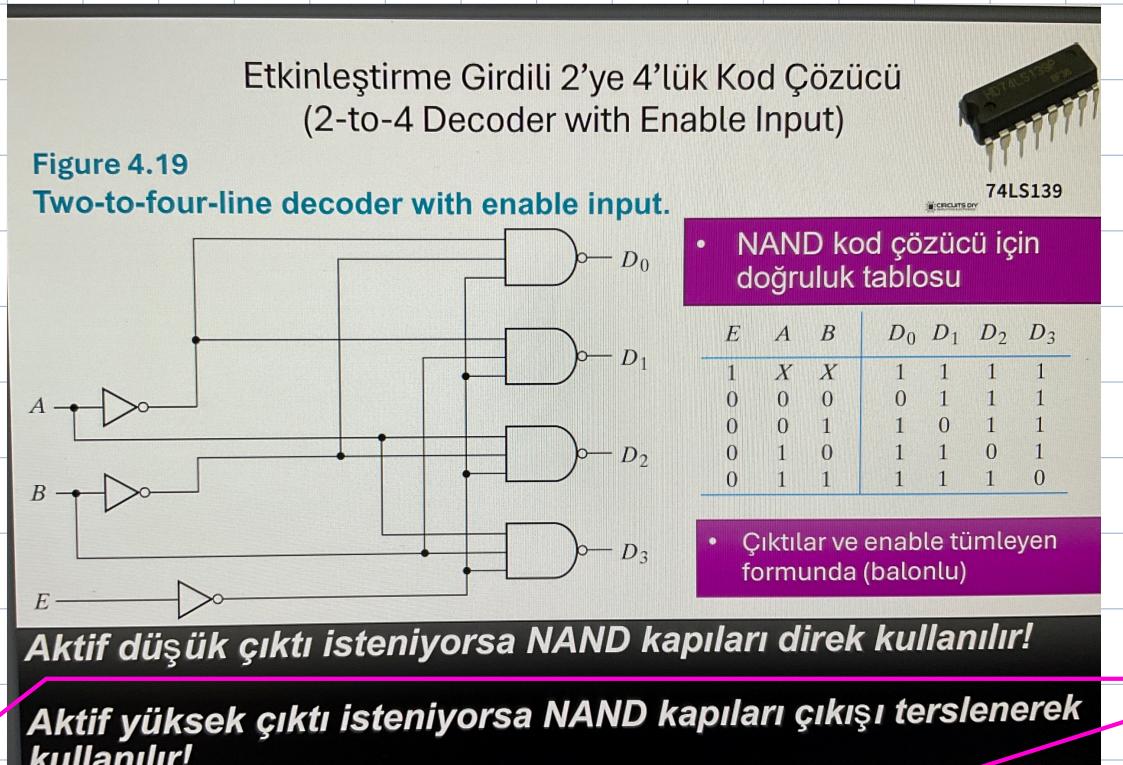
Gelişmiş Kod Çözücü

Ek özelliği Enable girdi ucunun olmasıdır.

- Devre

- yalnızca Enable seçili ise ($E=0$) çıktı üretir.
- Etkinleştirilmemis, ise ($E=1$) çıktı seçilmez.

önek: 2 ve 4'lik Enable sinyalli kod çözücü NAND kapıları kullanarak gerçekleştirir.



Daha büyük kod Gözüçüler.

Örnek 3: 3×8 kod Gözüçü ile

Tam toplayıcı Gerçeklestirme

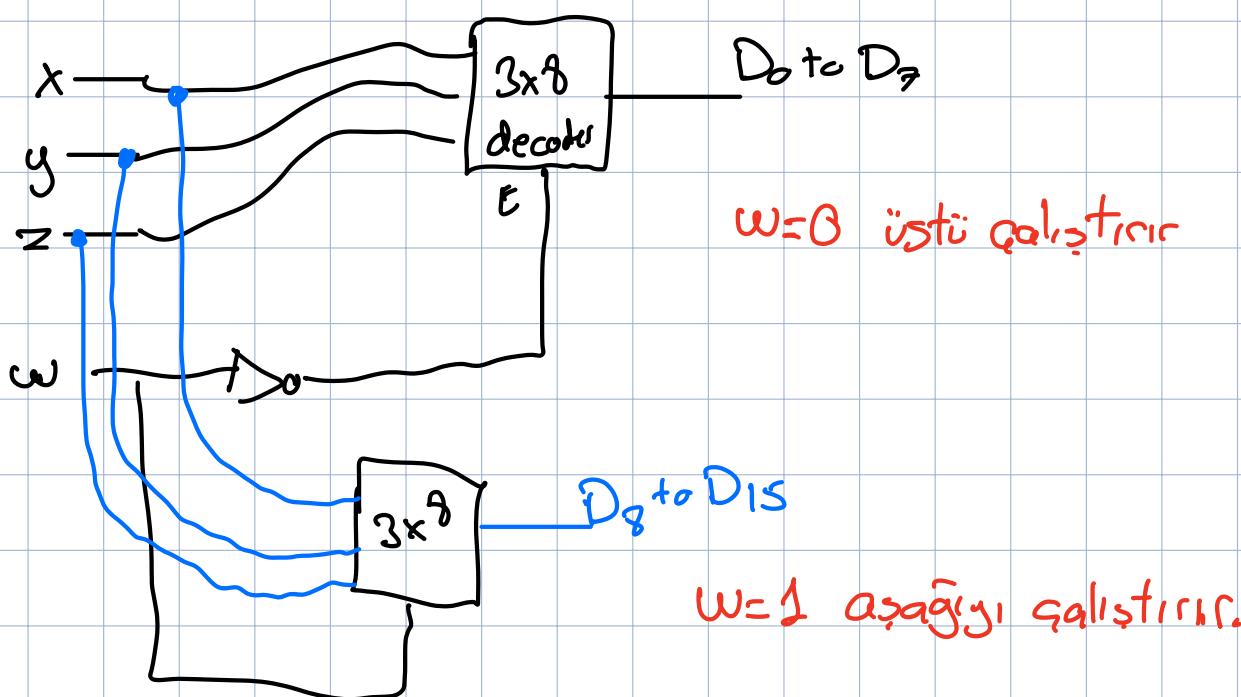
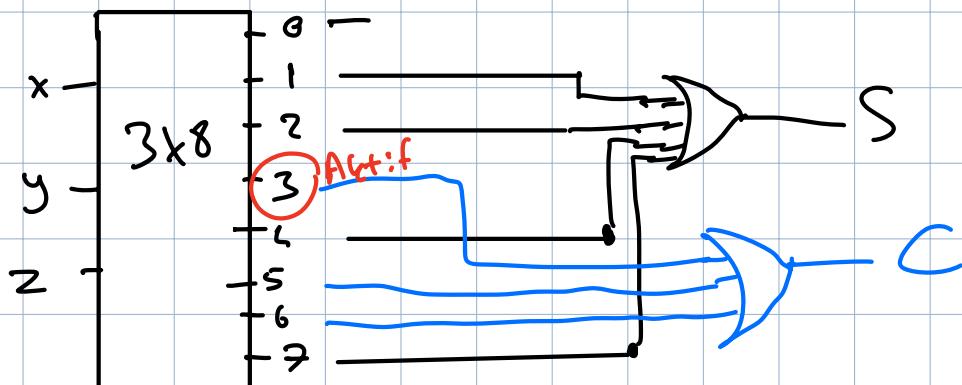
$$S(x,y,z) = \sum(1,2,4,7)$$

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

$$\text{ör: } xyz = 011 \rightarrow CS = 10$$

$$0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 3$$

aktif
olacak



Kodlayıcılar (Encoders)

⇒ Decoder'in tersi işlem yapar.

- Kodlayıcı Tasarımı

→ 2^n adet girdi hattını, n adet çıktı hattına çevirir.

Girdi: 2^n bit

Cıktı: n bit

Mangi giriş 1 ise çıktı onun ikili kodudur. *binary*

Table 4.7

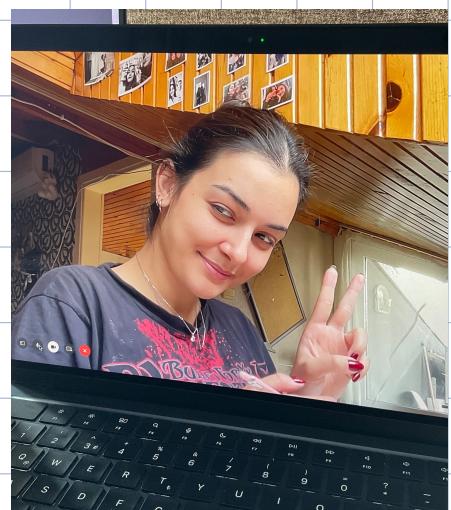
Truth Table of an Octal-to-Binary Encoder
(n=3 bit) → Input 8bit, Output 3bit

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	0	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

Ör: Girdi, bu satırın, bu noktasında 1 ise, çıktı binary değer 3 olmalıdır.

Sadece 1 girişin aktif HIGH olma kısıtı vardır.
Aynı anda birden fazla giriş 1 olamaz. Eğer olursa, çıkış tanımsız olur. !

Copyright © 2018, 2013, 2007 by Pearson Education, Inc..



Priority Encoder

D_0	D_1	D_2	D_3
0	1	1	0
1	0	1	0
1	1	0	1

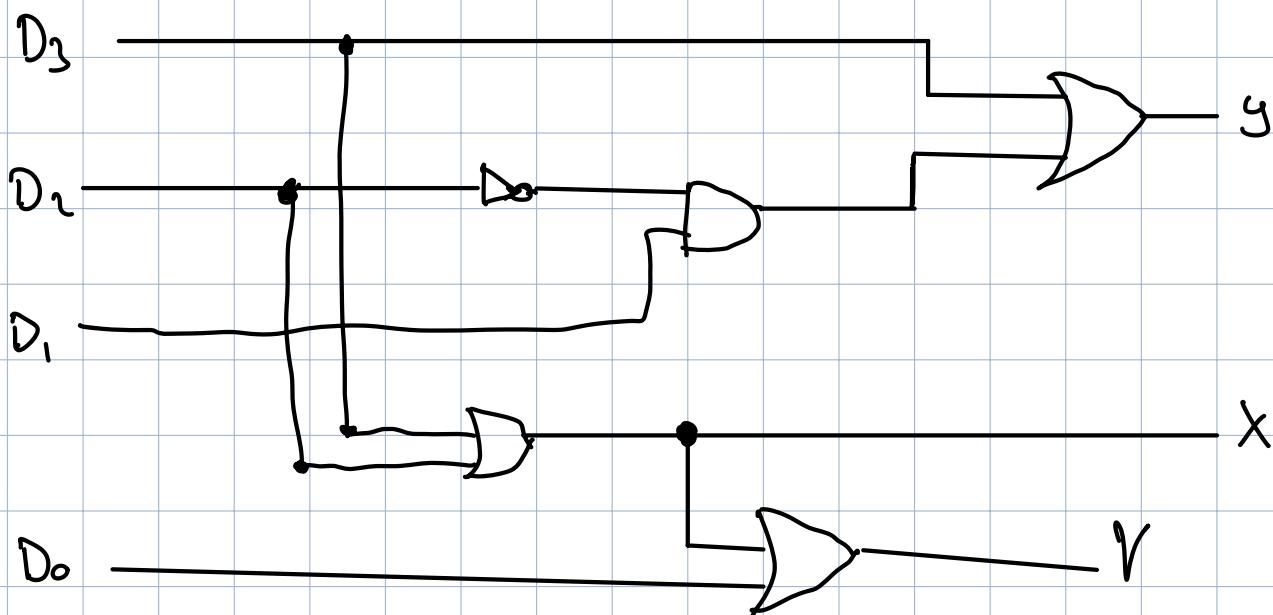
Validity

	X	Y	V
2.	1	0	1
2.	1	0	1
3.	1	1	1

anlamında bu kism

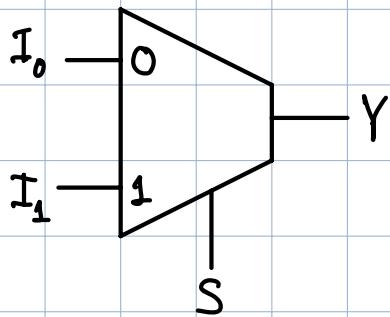
MSB alınır diğerleri 0 gibi davranışılır.

4 input priority encoder



MULTIPLEXER (MUX)

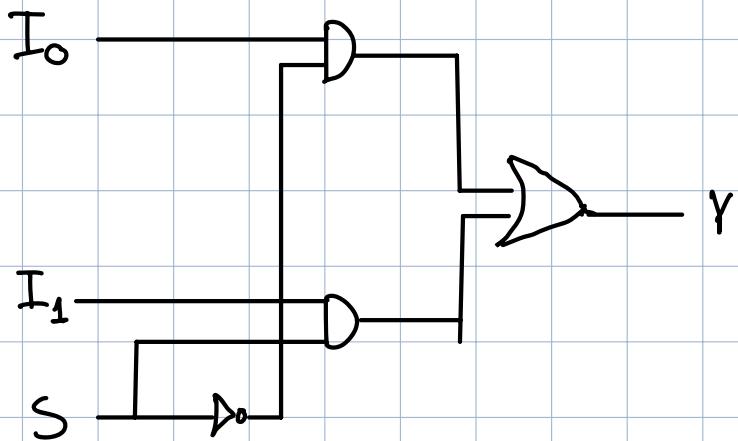
2×1



S	Y
0	I_0
1	I_1

=

4×1 MUX



4×1 MUX

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

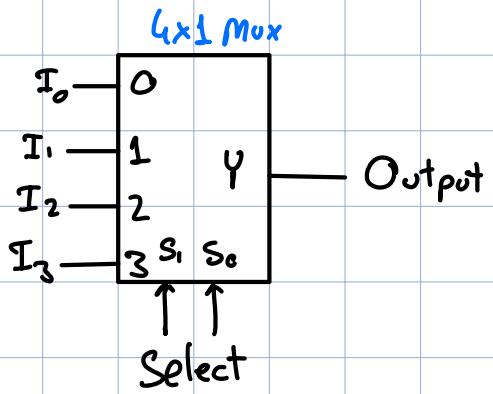
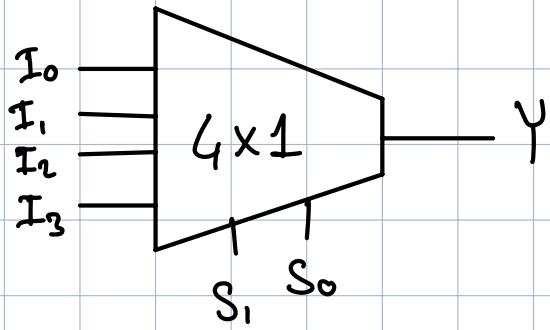
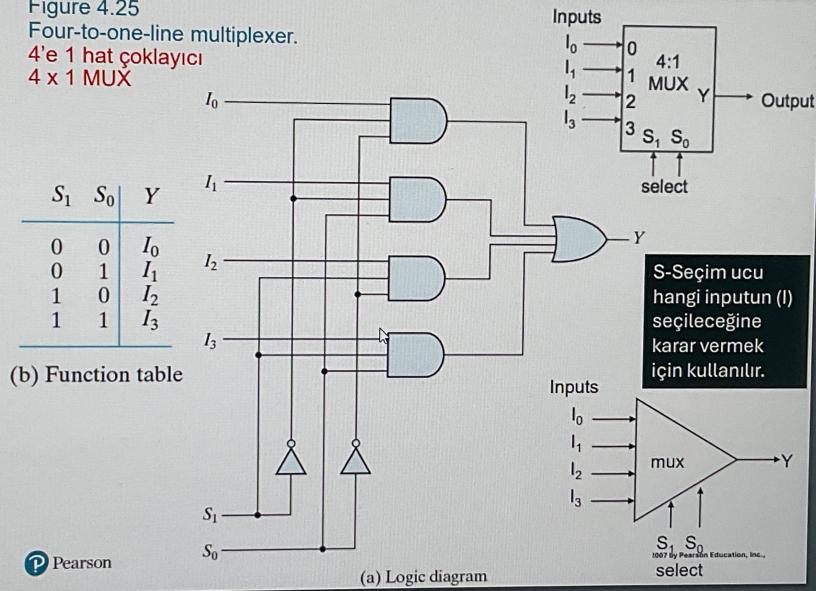


Figure 4.25

Four-to-one-line multiplexer.
4'e 1 hat çöklayıcı
 4×1 MUX



Ex

Implement the boolean function $F(x,y,z) = \sum(1,2,6,7)$ using 4×1 Mux
and necessary gates.

olduğu yerler

x	y	F	e çevirdik bu way
S_1	S_0	I_0	
0	0	I_0	
0	1	I_1	
1	0	I_2	
1	1	I_3	

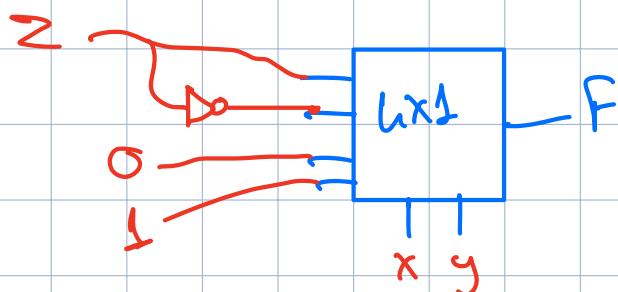
x	y	z	F
0	0	0	0
1	0	1	1
2	0	0	1
3	0	1	0
4	1	0	0
5	1	0	0
6	1	1	1
7	1	1	1

$0'$ i seçer xy
 00 (Binary)

$1'$ i seçer

$2'$ yi seçer

$3'$ ü seçer



Aida Gimer

Selection				F
A ₀	A ₁	A ₂	A ₃	
0	0	0	0	1
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

$A_2 A_3 I_0 \text{ Nand}$

$A_2 A_3 I_1$

$A_2 A_3 I_2$

$A_2 A_3 I_3$

$$\begin{aligned} & A_0' A_1' A_3' + A_1' A_2' A_3' \\ & + A_0' A_2' A_3 + A_1 A_2 A_3' \\ & + A_0 A_1 A_3 \end{aligned}$$

(XOR)

(NOR) m

OR

